WEB 2025

The Thirteenth International Conference on Building and Exploring Web Based Environments

ISBN: 978-1-68558-243-2

March 9th –13th, 2025

Lisbon, Portugal

WEB 2025 Editors

Petre DINI, IARIA, USA/EU

WEB 2025

Foreword

The Thirteenth International Conference on Building and Exploring Web Based Environments (WEB 2025), held between March 9 - 13, 2025, continued the inaugural conference on web-related theoretical and practical aspects, focusing on identifying challenges for building web-based useful services and applications, and for effectively extracting and integrating knowledge from the Web, enterprise data, and social media.

The Web has changed the way we share knowledge, the way we design distributed services and applications, the way we access large volumes of data, and the way we position ourselves with our peers.

Successful exploitation of web-based concepts by web communities lies on the integration of traditional data management techniques and semantic information into web-based frameworks and systems.

We take here the opportunity to warmly thank all the members of the WEB 2025 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to WEB 2025. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the WEB 2025 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that WEB 2025 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of Web-based environments.

We are convinced that the participants found the event useful and communications very open. We also hope that Nice provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

WEB 2025 Chairs:

WEB 2025 Steering Committee
Michel Jourlin, Jean Monnet University, Saint-Etienne, France
Daniela Marghitu, Auburn University, USA
Erich Schweighofer, University of Vienna - Centre for Computers and Law, Austria
Mariusz Trzaska, Polish-Japanese Academy of Information Technology, Poland
Taketoshi Ushiama, Kyushu University, Japan

WEB 2025 Publicity Chairs
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

WEB 2025

Committee

WEB 2025 Steering Committee
Michel Jourlin, Jean Monnet University, Saint-Etienne, France
Daniela Marghitu, Auburn University, USA
Erich Schweighofer, University of Vienna - Centre for Computers and Law, Austria
Mariusz Trzaska, Polish-Japanese Academy of Information Technology, Poland
Taketoshi Ushiama, Kyushu University, Japan

WEB 2025 Publicity Chairs
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

WEB 2025 Technical Program Committee

Muhammad Abulaish, South Asian University, New Delhi, India
Ali A. Alwan Al-juboori, Ramapo College of New Jersey, USA
Jesús M. Almendros-Jiménez, University of Almería, Spain
Leonidas Anthopoulos, University of Thessaly, Greece
Leandro Antonelli, Lifia | Universidad Nacional de La Plata (UNLP), Argentina
Hammache Arezki, Mouloud Mammeri University, Algeria
Vahid Ashrafimoghari, Stevens Institute of Technology, USA
Ismail Badache, LIS INSPE Aix-Marseille University, France
Maxim Bakaev, Novosibirsk State Technical University, Russia
Efthimios Bothos, Institute of Communications and Computer Systems (ICCS), Athens, Greece
Christos J. Bouras, University of Patras, Greece
Tharrenos Bratitsis, University of Western Macedonia, Greece
Erik Buchmann, Universität Leipzig, Germany
Rodrigo Capobianco Guido, São Paulo State University (UNESP), Brazil
Danilo Cavaliere, Università degli Studi di Salerno, Italy
Dickson Chiu, The University of Hong Kong, Hong Kong
Stefano Cirillo, University of Salerno, Italy
Toon De Pessemier, Ghent University, Belgium
Mohd Fazil, Madanapalle Institute of Technology & Science, India
Annamaria Ficara, University of Messina, Italy
Giacomo Fiumara, Università degli Studi di Messina, Italy
Raffaella Folgieri, Università degli Studi di Milano, Italy
Piero Fraternali, Politecnico di Milano, Italy
Marco Furini, University of Modena and Reggio Emilia, Italy
Jose Garcia-Alonso, University of Extremadura, Spain
Dion H. Goh, Nanyang Technological University, Singapore
Ramesh Gorantla, Arizona State University, USA

Table of Contents

ORKGEx: Leveraging Language and Vision Models with Knowledge Graphs for Research Contribution Annotation

Hassan Hussein[1], Fahad Ahmed[2], Allard Oelen[1], Ralph Ewerth[2], Sören Auer[1]
[1]Data Science & Digital Libraries Research Group, [2]Visual Analytics Research Group
TIB Leibniz Information Centre for Science and Technology
Hannover, Germany
{hassan.hussein, fahad.ahmed, allard.oelen, ralph.ewerth, auer}@tib.eu

***Abstract*—A major challenge in scholarly information retrieval is the semantic description of research contributions. While Generative Artificial Intelligence (AI) can assist in this regard, we need minimally invasive approaches for engaging users in the process. We introduce an innovative approach to annotating research articles directly within the browser and seamlessly integrate this approach with the Open Research Knowledge Graph (ORKG). This approach combines human intelligence with advanced neural and symbolic AI techniques to extract semantic research contribution descriptions and integrates the resulting AI-driven annotation tool within a web browser environment. Thus, we aim to facilitate user interaction and improve the creation and curation of scholarly knowledge. We evaluate the effectiveness of our neuro-symbolic approach through a comprehensive user study measuring the quality of the AI-assisted annotation process. Additionally, we illustrate the model's applicability and effectiveness with two use cases in sports analytics and environmental science.**

***Keywords*-Human; Multimodal AI; Knowledge Graphs; Human-Machine Collaboration.**

I. Introduction

One of the main challenges in research is accurately describing scientific contributions in a way that captures their subtle meanings, contexts, and implications. It is necessary to ensure that these descriptions can be used across different domains and with various information retrieval systems. Achieving this level of precision and consistency is crucial for improving the discoverability, accessibility, and usability of research outputs, promoting collaboration among scientists, and advancing knowledge coherently and systematically.

Previous studies have highlighted these challenges in scholarly annotations. Nantke and Schlupkothen [1] provided a comprehensive analysis of the complexity of annotation practices in scholarly editions, while Howlett and Turner [2] demonstrated how inconsistent terminology across disciplines affects research discoverability. Larivière et al. [3] and Devriendt et al. [4] revealed significant challenges researchers face in standardizing contribution descriptions and receiving proper recognition across different platforms and domains. These challenges are further compounded by the increasing volume and complexity of scientific literature, as demonstrated by Chu and Evans [5] in their analysis of scientific field progression.

Scientific articles contain not only text but figures as well. However, figures often present complex knowledge that can be challenging to comprehend, requiring cognitive and perceptual efforts. Extracting and comprehending data from these figures is challenging due to their diverse formats and contexts. Compared to photos, chart figures offer other challenges as they incorporate text and visual elements, such as lines, bars etc., to show relationships, making accurate data extraction challenging. Meanwhile, traditional Optical Character Recognition (OCR) techniques are not effective for charts since they can only extract text. However, combining OCR techniques with image processing and computer vision techniques can improve data extraction, utlizing object detection techniques, however, they have shown less generalizability and support only a few chart types [6] . Cutting-edge Vision-Language Models (VLMs) and Multimodal Large Language Models (MLLMs) enhance conventional models by integrating vision-language learning, expanding their abilities to image data.

Utilizing Generative AI can streamline semantic description tasks by automating metadata generation and research findings summarization. However, it is crucial to develop user-friendly interfaces and tools that allow researchers to engage with AI systems actively. This collaboration enables researchers to provide feedback, corrections, and additional context where needed, ensuring the precision and relevance of semantic descriptions. In addition, it also establishes trust and transparency in AI-generated outputs. Furthermore, empowering users to participate in the semantic annotation process promises deeper comprehension of their research contributions and promotes knowledge exchange within the scientific community.

We introduce a novel approach to annotate research articles directly within the browser, seamlessly integrating with the Open Research Knowledge Graph (ORKG [7]) to improve the efficiency and effectiveness of knowledge curation and discovery within the research community. Researchers often face challenges in marking key concepts, relationships, and contributions in scholarly documents, which limits the semantic richness and machine-readability of research metadata. Our approach combines human intelligence with advanced AI techniques, utilizing web-based annotation tool to effortlessly annotate research articles. The approach includes multimodal AI systems leveraging language and vision-language models to provide comprehensive annotations. It features human-in-the-loop interaction, allowing researchers to highlight text and receive AI suggestions for relevant property names and data types. In addition, the approach automates the metadata extraction to reduce manual entry and improve accuracy. These annotations automatically integrate with the ORKG, enriching

 ISBN: 978-1-68558-243-2

its content with structured, machine-readable metadata and promoting greater interoperability and collaboration within the research community, ultimately advancing scientific knowledge in a more accessible and interconnected form. To evaluate the effectiveness of our hybrid neuro-symbolic approach in enhancing scholarly knowledge creation and curation, we are conducting a user study measuring the impact of AI assistance on annotation speed, accuracy, efficiency, trustworthiness, and user satisfaction. By analyzing the benefits and challenges of incorporating AI-driven tools into the research workflow, we aim to refine our methods to better meet the needs of the academic community and create a smooth, user-friendly experience that helps researchers produce and share high-quality, and well-annotated scientific knowledge.

The article is structured as follows: Firstly, we review the related work in Section II. In Section III we discuss our approach. Furthermore, in Section IV, we explain our implementation through two use cases. Moreover, in Section V we discuss the users' feedback of our implementation. Additionally, in Section VI, we examine the methodological limitations and challenges in evaluating annotation systems, particularly focusing on the complexities of measuring annotation efficiency and system accuracy. Finally, in Section VII, we conclude and discuss the future work

II. Related Work

Over the last decade, Knowledge Graphs (KG) have emerged as a standard solution and a prominent research trend in academia and industry [8]. KGs [9] organize data in a structured and interconnected manner, simplifying access and analysis of complex datasets. Despite ongoing efforts to enhance the data quality of KGs, significant gaps in entities and relationships persist [9]. In our earlier work [10][11], we focused on improving reproducibility and addressing data inconsistencies, but these approaches often still require substantial manual intervention. Furthermore, the scholarly ecosystem remains predominantly document-centric [12], which we have previously argued hinders reproducibility and complicates the peer review process.

Current document processing approaches struggle with capturing complex document structures [13], while the integration of multimodal content, such as figures and tables, remains a significant challenge [14]. These limitations underscore the need for more comprehensive approaches that can handle both structural and multimodal aspects of scholarly documents while maintaining semantic relationships. This highlights the urgent need for new methods to manage and review the large volume [15] of published articles.

Consequently, the task of semantic description and extraction of scholarly knowledge became a primary focus of research, especially with the rise of advanced AI techniques. Current tools like SciBERT [16] and BERT [17] for scholarly texts have shown promise in text annotation. However, they often struggle with multimodal content integration, particularly in combining textual and visual information for comprehensive knowledge extraction. In this context, neural-symbolic approaches have emerged, integrating neural networks with the symbolic reasoning capabilities of KGs to enhance data quality [18]. These approaches foster comprehensive Knowledge Graph Reasoning (KGR) throughout the life cycle of KGs, offering improvements in scalability and interpretability. For instance, Knowledge Enhanced Graph Neural Networks (KeGNN) incorporate domain knowledge into graph neural networks through knowledge enhancement layers [19]. Additionally, a neural-symbolic system for KG entailment has been proposed, using abstract, generic symbols to discover entailment patterns [20]. However, neural-symbolic integration encounters challenges in effectively integrating robust learning and expressive reasoning under uncertainty [21].

Human-in-the-loop (HITL) computing is essential for developing effective AI systems by integrating human and machine intelligence [22][23]. This synergy leads to accurate results, fosters transparency, builds trust and demonstrates human control over AI [22][24][23]. Designing HITL systems enhances user experience and fosters new interactions between humans and AI [22][25]. HITL can also improve the interpretation of multimodal data, such as images and text, by enhancing collaboration and reducing measurement uncertainty [26]. Studies show that multimodal approaches integrating multiple modalities and involving human interaction have significantly enhanced multimedia comprehension [27]. To address these challenges, there is a pressing need to design intuitive user interfaces for effective human-AI collaboration.

TABLE I. ANALYZING KEY ANNOTATION TOOLS: INSIGHTS FROM AN ORKG COMPARISON

Tool Name	Focus	Type	Limitations	AI Utilization
LightTag	Text	Web-based	Limited features	No
GATE Teamware	NLP oriented	Web-based	Complex interface	Limited
VIA	Audio, Image, Video	Desktop	Limited to media files	Limited
UAM	Text	Desktop	Limited functionality	No
Anafora	Text	Web-based	Requires user configuration	No
YEDDA	Text	Desktop	Limited features	No

As shown in Table I, based on our comparative analysis [28], the available tools lack effective HITL integration. They have various weaknesses, including limited features, and complex interfaces, and require significant Natural Language Processing (NLP) expertise. Additionally, these tools often fail to fully utilize AI capabilities and do not support simultaneous annotation of text, figures, and other media types. This limitation hinders comprehensive data annotation and interpretation. In addition, information in the scholarly articles is mostly derived from their text. However, figures within these articles contain significant data that can be extracted and utilized [29].

Conventionally, OCR has been widely used to extract text from images, including scanned documents [30]. Nevertheless, when applied to charts and non-textual figures, OCR falls short due to the complexity and non-textual nature of the content [30][31]. Charts and plots often contain non-textual elements like data points, lines, bars, etc. Hybrid-OCR models also fall short to perform well on chart figures [32][33]. To solve the challenges of information extraction from figures, state-of-the-art Vision Transformers (ViT) [34], Vision Language Models (VLMs) such as UniChart [35], and LLaVa-NeXT [36] (LLaVa 1.6) have shown great results on the figure classification, chart-to-table and figure summarization tasks respectively.

In conclusion, while knowledge graphs maintain immense promise for organizing scholarly data, they demand significant effort to create and maintain. Neural-symbolic approaches offer significant advancements but also encounter specific challenges. Therefore, incorporating human expertise with multi-modal AI systems presents a viable pathway to bridge these gaps, enabling more effective and efficient scholarly knowledge management.

III. AI-Driven Multimodal Annotation Approach for Scholarly Knowledge Graphs

We propose a novel approach that integrates AI multimodality (language and vision-language models) with neuro-symbolic methods, involving human-in-the-loop interaction to semi-automatically create and curate scholarly knowledge graphs with a minimally invasive user experience.

Unlike existing annotation tools that typically focus on either text-only annotation or require extensive manual intervention, our approach uniquely combines multimodal AI capabilities to handle both textual and visual content simultaneously. This integration, coupled with automatic metadata extraction and research field detection, significantly reduces the annotation workload compared to traditional tools while maintaining high accuracy through human verification.

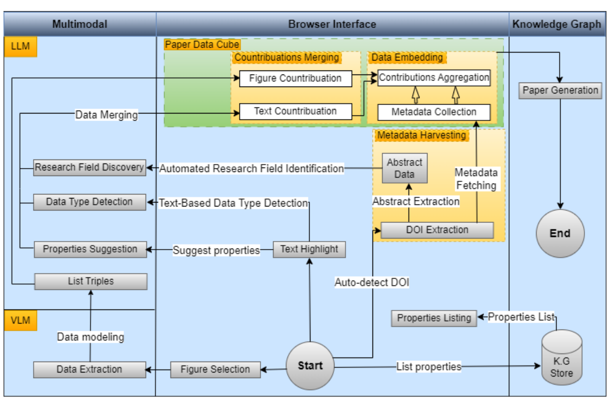


Figure 1. Illustration of the synergistic workflow integrating a multi-model AI system, knowledge graph, and human intelligence in a browser-based annotation system.

Figure 1 demonstrates our minimally invasive approach, meaning it requires minimal user intervention, to reduce user workload. We automatically extract the DOI and metadata from the annotated article, eliminating manual entry. Furthermore, we use AI Multimodal (LLM, and VLM) and knowledge graph capabilities to analyze the content of the article, particularly the abstract keywords, to automatically detect the research field and match it with predefined research fields in ORKG. When users highlight text, the LLM(For the text annotation, we use the gpt-3.5-turbo, as it the most capable model for such task) suggests relevant property names (e.g., "method" or "results") and automatically detects the data type (e.g., text, resource, integer). Meanwhile, we automatically list some most used properties for the user from the ORKG.

Our approach combines vision-language models to provide comprehensive annotations through prompt engineering by integrating multi-modal capabilities (explained in subsection III-A). Users can direct the LLM to process and annotate text, and the VLM for figures within research articles, identifying key concepts, methods, results, and discussions. In addition, the VLM analyzes visual figures within the article, to label significant elements and data points. To ensure coherent interpretation, our system creates connections between the extracted visual data and the surrounding textual content while enabling user verification. This is achieved by presenting the extracted data alongside the original figures, allowing users to cross-match and verify the accuracy of the interpretation

This approach allows the AI multimodal to interpret textual and visual data, thereby providing a holistic annotation of the research article. By integrating these advanced components, our approach furnishes a powerful and minimally invasive tool for researchers to efficiently annotate their articles, leveraging both AI multimodal and KG for accurate and context-aware annotations across multiple modalities.

A. Knowledge Extraction from Figures

We have developed an end-to-end pipeline which utilizes a fine-tuned ViT model and pretrained VLMs for figure classiication and chart comprehension tasks such as summarization and chart-to-text. We fine-tuned the ViT [34] model on the DocFigure [37] dataset to classify scientific figures. The fine-tuned model shows approx 50% accuracy on DocFigure [37]. The UniChart [35] is instruction-tuned on a variety of chart comprehension tasks and has shown improved results compared to other SOTA models. LLaVa-Next [36] has shown promising results for general figure summarization task.

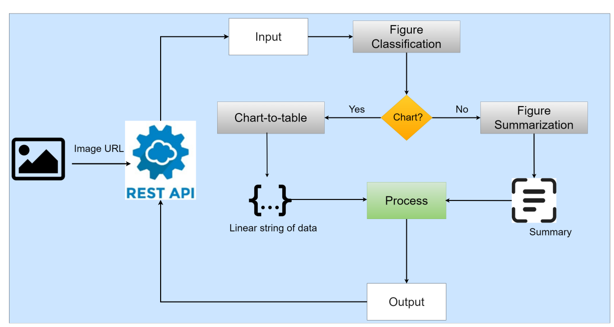


Figure 2. End-to-end information extraction from figures pipeline and REST API.

Figure 2 shows an end-to-end inference pipeline that can be accessed through a REST API. The plugin sends the URL of the selected figure to the REST API. The image is preprocessed before it is passed to the figure classification model. The classification output is then used to determine whether the figure is a chart/plot or some other type of figure. In case of chart figures, it is used to extract the data table using the UniChart while for non-chart/plot figures the LLaVa-NeXT is used to summarize what the figure actually is about. The outputs are then processed and returned to the REST API caller in the form of a JSON response.

In summary, our approach combines AI Multimodal language and visual models with neuro-symbolic techniques and human-in-the-loop interaction to facilitate the creation and curation of scholarly knowledge graphs with minimal invasive effort. The approach aims to automate the extraction of DOIs and metadata, reducing manual data entry. Using large foundation models and knowledge graph techniques, the approach facilitates analyzing article content to identify and categorize research fields and matches them to predefined ORKG fields.

IV. A Browser Based Interface Implementation:

Our focus is on creating a browser-based extension aims to streamline the workflow of researchers. The extension promotes collaboration between humans and AI Multimodal, to reduce the time and effort researchers spend on selecting properties and identifying relevant research fields. Additionally, it allows researchers to extract data from figures with just few clicks, making the annotation process more efficient for both figures and text. In this section, we showcase two use cases to illustrate our approach. Use case IV-A demonstrates text annotation, while use case IV-B focuses on annotating figures.

A. Sports Analytics Domain Use Case:

To demonstrate our extension's text annotation capabilities, we analyzed a recent paper by Settembre et al. [38]. This work was selected for its dense scientific terminology, making it an ideal test case for evaluating our AI's effectiveness in identifying data types and suggesting appropriate property names. Figure 3 demonstrates our streamlined text annotation pipeline, which consists of four key steps.:

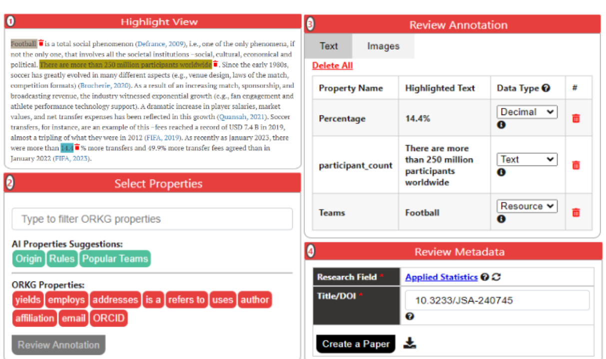


Figure 3. Streamlined Text Annotation Pipeline: From Highlight text to Property Selection, Annotation Review, Datatype Detection, and Metadata Evaluation.

- **1. Highlight View**: This view shows the standard browser interface, permitting researchers to read their selected paper. Users can highlight the text and then click the browser extension to display a list of properties as explained in the next step.
- **2. Select Properties**: In this interface, AI-suggested properties are highlighted in green. Using a predefined prompt template 1, the LLM recommends properties based on highlighted text. Additionally, top properties from the ORKG are listed as red-colored buttons. Users can further search for properties by entering names in a designated search box. Once the highlighted text is associated with a property, the "Review Annotation" button becomes active, allowing users to proceed to the next step.

> **AI Prompt 1:** Act as an ORKG researcher. Given the selected text: (Highlighted Text), identify the most suitable property names. Provide a JSON array with a maximum of four concise property names. Format the response as follows: ['property1', 'property2', 'property3', 'property4']. 1

- **3. Review Annotation**: In this step, the extension sends automatically prompt 2 to the LLM to determine if the highlighted text can be classified as a resource, aligning with the concept of linked data [39]. The annotated data is displayed in a tabular format, with the data type decided by the LLM's findings and accompanied by an explanation, viewable by hovering over the info icon. Users can fully control annotations, including deleting any or all annotated text.

> **AI Prompt 2:** Act as an ORKG researcher. Given the selected text: (Highlighted Text), decide if it should be treated as a resource. Provide your decision and explain why the text should or should not be treated as a resource. 2

- **4. Review Metadata**: The paper's DOI is extracted using a regular expression, typically the first DOI at the top of the paper. Then, metadata is harvested through the Semantic Scholar API service [40]. Concurrently, the LLM analyzes the paper's abstract using a prompt 3 to identify top three relevant research fields. Users can toggle the LLM results by clicking the refresh icon for accurate research field suggestions.

> **AI Prompt 3:** Act as an ORKG researcher. Given the following abstract: (Abstract), identify the three most relevant research fields. Provide a JSON array of concise research field names without values or additional information. Format the response as follows: ["research field1", "research field2", "research field3"]. 3

The extension compiles all metadata, research fields, and highlighted text into a single object for submission to the ORKG to create the paper entry, combining all highlighted text as a single contribution. Additionally, users can download the paper object by clicking the icon next to the "Create a Paper" button, which promotes data interoperability.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-243-2

B. Environmental Science Domain Use Case:

For our second use case, we evaluated our visual model's annotation capabilities using an Environmental Science paper by Darian-Smith [41]. Specifically, we analyzed Figure 2 from their work, which presents "The Economist: Democracy Index 2023: Age of Conflict" (page 3), as demonstrated in Figure 4. This figure was selected for its complex line chart visualization, which provides an excellent test case for our multimodal annotation approach.:

1) **Selecting Figure and Reviewing the Annotation:** The process starts with the user selecting the figure they want to annotate. When they activate the Chrome extension, an interface showing the selected images will appear. The user can then click on the image annotation view, which will open in a new window.
2) **Image Annotation Review**: After the separate window opens, a request is sent to the VLM to extract key data from the figure. The extracted data is then sent to the LLM using a prompt 4 to format it into triples. Once formatted, the data is displayed to the user, as illustrated in step two. The user can edit, add, or remove the triples as needed.

> **AI Prompt 4:** Act as a researcher and transform the following JSON object (Image Annotation Data) into subject-predicate-object form. Return the data in a JSON object, strictly in the following format: "data": ["subject": "subject value", "predicate": "predicate name", "object": "object value"] Ensure that each entry includes both the subject value and the predicate name. 4

3) **Review Metadata** This step is similar to step 4 in use case IV-A. The key discrepancy here is that each figure annotation is considered a new contribution when added to the ORKG.

Figure 4. Automatic extraction of key data points from a line chart is enabled by a Visual Language Model (VLM) integrated within a Chrome extension.

In our study, we demonstrated the capabilities of our extension in two different fields: sports analytics and environmental science. In sports analytics IV-A, our extension efficiently annotated text using LLM for data type detection, property suggestion, and research field discovery. It integrated with ORKG to streamline the process. In the field of environmental science IV-B, our extension adeptly annotated figures such as line charts using advanced visual and language models. It was able to extract and format the data into triples. Both use cases highlighted the extension's versatility, accuracy, and its ability to assist researchers in annotating various types of research artifacts. It is important to highlight that, the extension permits users to annotate text and figures simultaneously in a single batch process. However, we demonstrated these capabilities separately to emphasize their individual functionalities and benefits.

V. User Evaluation

The evaluation of user feedback serves to systematically validate the performance and effectiveness of our Chrome extension across several key dimensions. Our evaluation is presented in five subsections, structured as follows:

A. Evaluation Design

We conducted a comprehensive user study with 11 participants recruited through professional networks, including postdoctoral researchers, PhD students, and software developers. Of these, 82% had prior ORKG experience, providing valuable feedback on enhancing the extension based on their familiarity with the ORKG pain points. Since the extension is not yet available on the Chrome Web Store, we asked the participants to install TeamViewer software to access our local machine and evaluate the extension remotely. After a brief demo of the extensions' capabilities, participants independently evaluated it. They then filled out a feedback form for analysis

B. Statistical Analysis Methods

All evaluations used a 5-point Likert scale (>3 indicating agreement). The core statistical measures are including sample mean and standard deviation [42], and 95% confidence intervals via t-distribution [43]

C. Results and Analysis

TABLE II. SPEED AND EFFICIENCY METRICS

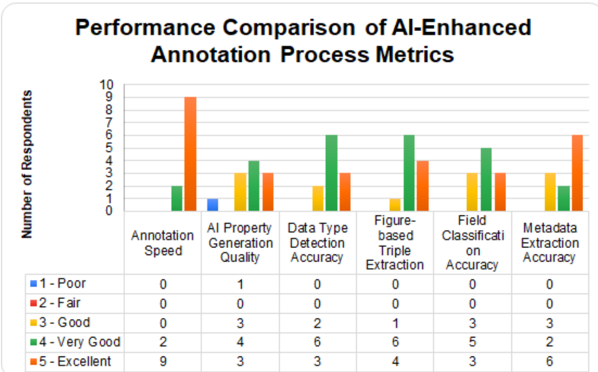
Feature	Mean $\pm$ SD	95% CI	Key Finding
Figure Triples	4.27 $\pm$ 0.65	(3.83, 4.71)	Highly efficient
Property Suggestions	3.73 $\pm$ 1.14	(2.97, 4.49)	Variable performance
Overall Speed	4.82 $\pm$ 0.39	(4.56, 5.08)	Significant gain

TABLE III. RELIABILITY AND PERFORMANCE ANALYSIS

Metric	Mean $\pm$ SD	Distribution	95% CI
Data Type Detection	4.09 $\pm$ 0.67	5*:27.3% 4*:54.5% 3*:18.2%	(3.64, 4.54)
Field Classification	4.00 $\pm$ 0.74	5*:27.3% 4*:45.5% 3*:27.3%	(3.50, 4.50)
Metadata Extraction	4.27 $\pm$ 0.86	5*:54.5% 4*:18.2% 3*:27.3%	(3.69, 4.85)

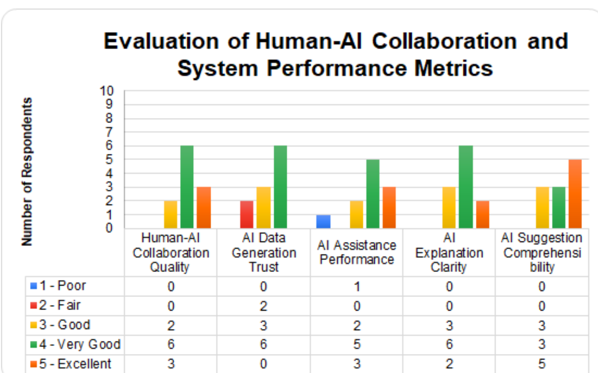
D. Visual Analysis

Figure 5 demonstrates that 82% of participants reported accelerated annotation compared to traditional interface, with 90% effectiveness in figure-based triple extraction and 73% accuracy in field detection and metadata harvesting. The human-AI collaboration metrics (Figure 6) reveal 82% user trust in AI-generated content and 73% accuracy perception, with 82% rating the AI content as transparent and explainable. The user experience assessment (Figure 7) indicates high satisfaction levels with 73% reporting adequate control, 90% ease of learning, and 90% positive overall experience.



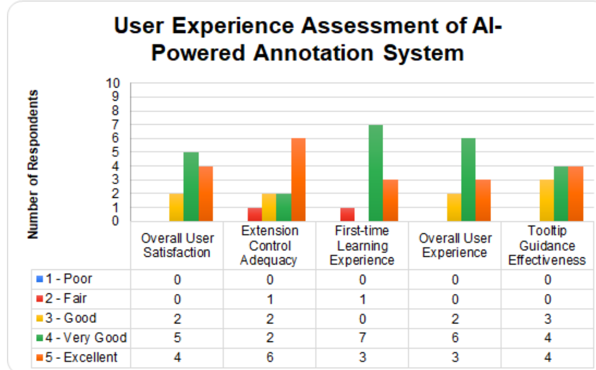
	Annotation Speed	AI Property Generation Quality	Data Type Detection Accuracy	Figure-based Triple Extraction	Field Classification Accuracy	Metadata Extraction Accuracy
1 - Poor	0	1	0	0	0	0
2 - Fair	0	0	0	0	0	0
3 - Good	0	3	2	1	3	3
4 - Very Good	2	4	6	6	5	2
5 - Excellent	9	3	3	4	3	6

Figure 5. AI-enhanced annotation process metrics showing effectiveness across tasks (Likert scale 1-5). Notable results: figure-based triple extraction (90% positive) and metadata harvesting (73% accuracy).



	Human-AI Collaboration Quality	AI Data Generation Trust	AI Assistance Performance	AI Explanation Clarity	AI Suggestion Comprehensibility
1 - Poor	0	0	1	0	0
2 - Fair	0	2	0	0	0
3 - Good	2	3	2	3	3
4 - Very Good	6	6	5	6	3
5 - Excellent	3	0	3	2	5

Figure 6. Human-AI interaction quality metrics showing strong user trust (82%) and AI suggestion comprehensibility (73%).



	Overall User Satisfaction	Extension Control Adequacy	First-time Learning Experience	Overall User Experience	Tooltip Guidance Effectiveness
1 - Poor	0	0	0	0	0
2 - Fair	0	1	1	0	0
3 - Good	2	2	0	2	3
4 - Very Good	5	2	7	6	4
5 - Excellent	4	6	3	3	4

Figure 7. User experience metrics showing strong control (73%), ease of use (90%), and overall satisfaction (90%).

E. Qualitative Analysis

The ORKGEx extension received positive feedback for its ease of use, AI-generated suggestions, and ability to save time during the annotation process. Participants found the tool intuitive and efficient, with many noting that it significantly improved their workflow compared to the other annotaion tools. The AI-generated property suggestions were particularly appreciated, though some users suggested providing more context (e.g., surrounding paragraphs) to enhance relevance. Tooltips were deemed helpful but could be improved with better visibility, such as larger fonts and more noticeable colors. Overall, the extension was seen as a valuable addition to the research community, with potential for further refinement. A summary of the key findings, along with supporting quotes from participants, is provided in Table IV.

The evaluation demonstrates the extension's effectiveness in enhancing annotation processes. Key findings include:

- **Strong user satisfaction:** with the extension, as evidenced by a high mean rating for overall satisfaction (4.18 ± 0.75)
- **Robust performance:** in AI-assisted features, particularly in metadata extraction (mean: 4.27 ± 0.86) and data type detection (mean: 4.09 ± 0.67).
- **Positive user feedback:** highlighting the extension's ease of use, efficiency, and potential to streamline the annotation process.

VI. Limitation

When considering our evaluation methodology, we identified several limitations despite the positive results. While our user study provided valuable perception metrics and validation data, there remain inherent challenges in conducting a fully comprehensive evaluation:

1) **Performance measurement limitations**: Although we obtained significant user-reported speed improvements (4.82 ± 0.39), establishing absolute performance baselines remains challenging due to:
   - Annotator's expertise and familiarity with the domain
   - Paper length and structural complexity
   - Research field and complexity of the scientific content
   - Number and complexity of figures
2) **System evaluation challenges**: We need to do a more comprehensive evaluation to:
   - Larger-scale comparison between manual and automated annotations
   - Controlled environment for quantitative time measurements
   - Standardized test sets with varying complexity levels
3) **Sample diversity**: Although our evaluation with 11 participants (82% with ORKG experience) provided statistically significant results, it is important to note that the small sample size may limit the generalizability of our findings.

TABLE IV. SUMMARY OF KEY FINDINGS WITH SUPPORTING QUOTES

Aspect	Positive Feedback	Constructive Feedback	Supporting Quote
Ease of Use	Intuitive and easy to learn.	Some technical issues and limited testing time affected usability.	*"The tool efficiently harnesses AI capabilities to suggest relevant scientific data in relation to the given user input."* (Masters Student)
AI Suggestions	Helpful in simplifying the annotation process.	Needs more context (e.g., surrounding paragraphs) for better relevance.	*"AI-generated property suggestions needed more context about the selected text."* (Frontend Developer)
Tooltips	Provide helpful information.	Need better visibility (e.g., larger fonts, catchy colors).	*"Tooltips provide helpful information, therefore they have to be represented in a more obvious way (customizing tooltips with a bigger font and a catchy color would be a great addition)."* (Backend Developer)
Metadata Extraction	Generally accurate.	Some participants did not closely examine this feature.	*"I didn't have the option to skip some questions in the evaluations because I didn't closely examine aspects like metadata extraction."* (Frontend Developer)
Overall Satisfaction	High satisfaction with the extension's ability to improve annotation efficiency.	Some participants found it difficult to assess certain features due to limited testing.	*"Overall, I see a huge potential in having such an extension to bring annotated data to the knowledge graph. Keep it up!"* (Frontend Developer)

Broader validation across different expertise levels and larger sample sizes would strengthen our results.

These limitations highlight the complexity of evaluating annotation tools in real-world scenarios, where controlled experiments must be balanced against practical usability and varied user needs.

VII. Conclusion and Future Work

In this paper, we proposed a novel approach to enhance the semantic description of research articles with web-based annotation integrated with the Open Research Knowledge Graph. Our evaluation demonstrated strong performance across multiple metrics, with notably high user satisfaction (90%) and effective AI-assisted features. However, it is important to note that the 90% user satisfaction is based on a sample size of 11 participants, and while statistically significant, it may not fully represent the broader user base.

In the future, we plan to explore advanced techniques in computer vision and natural language processing, such as transformer-based models for figure extraction and fine-tuned language models for research describtion, to improve the extraction and interpretation of complex figures within scholarly articles. To ensure real-world applicability, we will investigate scalability challenges specific to browser-based annotation, such as optimizing the extension's performance with large documents, and managing the synchronization of annotations across multiple users and sessions.

Additionally, we plan to address critical deployment challenges including: ensuring consistent annotation behavior across various browser versions, and maintaining responsive performance while processing complex figures and metadata.

While our current evaluation provides statistically significant insights into user experience and measured benefits, future work will expand evaluation scope with larger, more diverse research communities. To mitigate potential evaluation biases, we will employ systematic sampling methods to ensure balanced representation across different academic disciplines, experience levels, and institutional backgrounds. This expanded evaluation will include both ORKG experts and novices, enabling us to understand how the extension performs across different expertise levels and varied real-world academic workflows. Additionally, we will focus on expanding support for complex data types and interactive elements, such interactive tables, to further enhance the extension's utility across diverse use cases.

The source code of the implementation is available under: https://github.com/Webo1980/ORKGEx

References

[1] J. Nantke and F. Schlupkothen, *"Annotations in Scholarly Editions and Research: Functions, Differentiation, Systematization"*. Berlin: De Gruyter, 2020. DOI: 10.1515/9783110689112.

[2] K. Howlett and E. Turner, "Research on the benefits of nature to people: How much overlap is there in citations and terms for 'nature' across disciplines?", *People and Nature*, Dec. 2023.

[3] V. Larivière, D. Pontille, and C. Sugimoto, "Investigating the division of scientific labor using the Contributor Roles Taxonomy (CRediT)", *MIT Press - Journals*, Jan. 2021. DOI: 10.1162/qss_a_00097.

[4] T. Devriendt, P. Borry, and M. Shabani, "Credit and recognition for contributions to data-sharing platforms among cohort holders and platform developers in Europe : Interview study", *JMIR Publications Inc.*, Jan. 2022. DOI: 10.2196/25983.

[5] J. Chu and J. Evans, "Slowed canonical progress in large fields of science", *Proceedings of the National Academy of Sciences*, Dec. 2022. DOI: 10.1073/pnas.2021636118.

[6] J. Luo *et al.*, "Chartocr: Data extraction from charts images via a deep hybrid framework", in *Proceedings of the IEEE/CVF*

winter conference on applications of computer vision, 2021, pp. 1917–1925. DOI: 10.1109/WACV48630.2021.00196.
[7] S. Auer *et al.*, *"open research knowledge graph"*, https://orkg.org, Accessed: 2025.02.02.
[8] Y. Kong *et al.*, "Bolt defect classification algorithm based on knowledge graph and feature fusion", *Energy Reports*, vol. 8, pp. 856–863, 2022.
[9] C. Peng *et al.*, "Knowledge graphs: Opportunities and challenges", *Artificial Intelligence Review*, vol. 56, pp. 13 071–13 102, 2023. DOI: 10.1007/s10462-023-10465-9.
[10] H. Hussein *et al.*, "Increasing reproducibility in science by interlinking semantic artifact descriptions in a knowledge graph", in *Leveraging Generative Intelligence in Digital Libraries: Towards Human-Machine Collaboration*, D. H. Goh, S.-J. Chen, and S. Tuarob, Eds., Singapore: Springer Nature Singapore, 2023, pp. 220–229, ISBN: 978-981-99-8088-8. DOI: https://doi.org/10.1007/978-981-99-8088-8_19.
[11] H. Hussein *et al.*, "KGMM - a maturity model for scholarly knowledge graphs based on intertwined human-machine collaboration", in *From Born-Physical to Born-Virtual: Augmenting Intelligence in Digital Libraries*, Y.-H. Tseng, M. Katsurai, and H. N. Nguyen, Eds., Cham: Springer International Publishing, 2022, pp. 253–269, ISBN: 978-3-031-21756-2.
[12] M. Stocker *et al.*, "Fair scientific information with the open research knowledge graph", *FAIR Connect*, 2023.
[13] D. Wang *et al.*, "Docgraphlm: Documental graph language model for information extraction", *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 14 137–14 151, 2023. DOI: https://doi.org/10.1145/3539618.3591975.
[14] L. Wang *et al.*, "Improving the accessibility of scientific documents: Current state, user needs, and a system solution to enhance scientific pdf accessibility for blind and low vision users", *ArXiv*, vol. abs/2105.00076, 2021.
[15] J. Greenberg *et al.*, "Knowledge organization systems: A network for AI with helping interdisciplinary vocabulary engineering", *Cataloging & Classification Quarterly*, vol. 59, no. 8, pp. 720–739, Nov. 17, 2021, ISSN: 0163-9374, 1544-4554. DOI: 10.1080/01639374.2021.1995918.
[16] L. Beltagy, K. Lo, and A. Cohan, "Scibert: Pretrained contextualized embeddings for scientific text", *CoRR*, vol. abs/1903.10676, 2019. arXiv: 1903.10676.
[17] S. Bano and S. Khalid, "Bert-based extractive text summarization of scholarly articles: A novel architecture", in *2022 International Conference on Artificial Intelligence of Things (ICAIoT)*, 2022, pp. 1–5. DOI: 10.1109/ICAIoT57170.2022.10121826.
[18] H. Chen *et al.*, "Neural symbolic reasoning with knowledge graphs: Knowledge extraction, relational reasoning, and inconsistency checking", *Elsevier BV*, Sep. 2021. DOI: 10.1016/j.fmre.2021.08.013.
[19] L. Werner *et al.*, "Knowledge Enhanced Graph Neural Networks for Graph Completion", *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, Mar. 2023.
[20] M. Ebrahimi *et al.*, "Neuro-Symbolic Deductive Reasoning for Cross-Knowledge Graph Entailment", in *AAAI Spring Symposium Combining Machine Learning with Knowledge Engineering*, Oct. 2021.
[21] R. Manhaeve *et al.*, "Neuro-symbolic ai = neural + logical + probabilistic ai", in *Neuro-Symbolic Artificial Intelligence*, Jan. 2021. DOI: 10.3233/FAIA210354.
[22] C. Emmanouilidis and S. Waschull, "Human in the Loop of AI Systems in Manufacturing", *NOW PUBLISHERS INC*, Nov. 2021. DOI: 10.1561/9781680838770.ch9.
[23] J. Ostheimer, "Human-in-the-loop Computing : Design Principles for Machine Learning Algorithms of Hybrid Intelligence", *Linnéuniversitetet, Institutionen för informatik (IK)*, Jan. 2019.
[24] C. Emmanouilidis *et al.*, "Human in the AI Loop in Production Environments", *Springer Science and Business Media LLC*, Aug. 2021. DOI: 10.1007/978-3-030-85910-7_35.
[25] K. Tsiakas and D. Murray-Rust, "Using human-in-the-loop and explainable AI to envisage new future work practices", *Association for Computing Machinery (ACM)*, Jan. 2022. DOI: 10.1145/3529190.3534779.
[26] J. Effendi *et al.*, "Multimodal Chain: Cross-Modal Collaboration Through Listening, Speaking, and Visualizing", *Institute of Electrical and Electronics Engineers (IEEE)*, vol. 9, pp. 70 286–70 299, 2021. DOI: 10.1109/ACCESS.2021.3077886.
[27] T. Gong *et al.*, "Multimodal-gpt: A vision and language model for dialogue with humans", *arXiv*, 2023.
[28] H. Hussein *et al.*, *"comparative analysis of annotation tools: Focus, methods, and user suitability"*, https://orkg.org/comparison/R1354448, Accessed: 2025.02.02, DOI: 10.48366/R1354448.
[29] G. Park, J. Rayz, and L. Pouchard, "Figure descriptive text extraction using ontological representation", *eprint arXiv:2208.06040*, Aug. 2020. DOI: 10.48550/arXiv.2208.06040.
[30] J. Ghorpade-Aher *et al.*, "Text retrieval from natural and scanned images", *International Journal of Computer Applications*, vol. 133, no. 8, pp. 10–12, Jan. 2016, ISSN: 0975-8887. DOI: 10.5120/ijca2016907840.
[31] G. Skitalinskaya and N. Düpont, "Ocr report", *Deutsche Forschungsgemeinschaft (DFG)*, 2021. DOI: 10.26092/elib/1517.
[32] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer", *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.
[33] M. Zhou *et al.*, "Enhanced chart understanding in vision and language task via cross-modal pre-training on plot table pairs", *arXiv preprint arXiv:2305.18641*, 2023.
[34] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", *arXiv preprint arXiv:2010.11929*, 2020.
[35] A. Masry *et al.*, "Unichart: A universal vision-language pretrained model for chart comprehension and reasoning", *arXiv preprint arXiv:2305.14761*, 2023.
[36] H. Liu *et al.*, *Llava-next: Improved reasoning, ocr, and world knowledge*, https://llava-vl.github.io/blog/2024-01-30-llava-next/, Accessed: 2025.02.02, 2024.
[37] K. Jobin, A. Mondal, and C. Jawahar, "Docfigure: A dataset for scientific document figure classification", in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, IEEE, vol. 1, 2019, pp. 74–79. DOI: 10.1109/ICDARW.2019.00018.
[38] M. Settembre *et al.*, "Factors associated with match outcomes in elite european football – insights from machine learning models", *IOS Press*, 2024. DOI: 10.3233/JSA-240745.
[39] W. W. W. C. (W3C), *Linked data*, https://www.w3.org/wiki/LinkedData, Accessed: 2025.02.02.
[40] Allen Institute for AI, *Semantic Scholar API*, https://www.semanticscholar.org/product/api, Academic search and discovery platform, Accessed: 2025.02.02.
[41] E. Darian-Smith, "The challenge of political will, global democracy and environmentalism", *Environmental Policy and Law*, vol. 54, pp. 117–126, Jan. 2024. DOI: 10.3233/EPL-239023.
[42] S. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press, 2014.
[43] Student, "The probable error of a mean", *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908, ISSN: 00063444, 14643510.

How to Interpret an Economic Index? Generating Reports with Topic Sentiment Analysis

Kazuhiro Seki
Faculty of Intelligence and Informatics
8-9-1 Okamoto, Higashinada, Kobe
Hyogo, Japan
e-mail: seki@konan-u.ac.jp

***Abstract*—This paper presents our ongoing work on generating a monthly economic report to enhance the interpretability of a news-based business sentiment index. In our previous work, we developed a framework to provide a timely and quantitative measure of economic confidence. However, it lacked clarity on the underlying factors influencing the index values. To tackle the issue, we propose a topic-centered summarization approach which explains key events and trends during a specified period. Preliminary experiments show a promising result, producing a concise and comprehensive summary of the economic situation.**

Keywords-topic analysis; summarization; large language models; sentiment analysis.

I. INTRODUCTION

A Business Sentiment Index (BSI) provides a quantitative measure of business confidence in the current/future economic climate. Traditionally, BSIs are derived from surveys conducted across various sectors. However, these survey-based methods often suffer from high costs and delays, limiting their utility for real-time economic analysis.

To address these challenges, our previous work [1] developed a framework that automatically nowcasts a monthly BSI using online news articles. While this approach showed strong correlation with a traditional survey-based index and offered significant improvements in timeliness and cost-efficiency, it lacked interpretability. Specifically, understanding why a particular index value, such as “BSI=0.26,” was obtained remained unclear.

To bridge this gap, we propose a framework that generates an economic report to explain the factors behind the BSI values. Our method leverages topic and sentiment analyses and a two-step text summarization to analyze and synthesize relevant economic events from news articles. A preliminary experiment on October 2024’s news articles demonstrated promising results, producing a concise and comprehensive summary of the economic situation for the given period.

The remainder of this paper is organized as follows: Section II describes our proposed approach, while Section III presents the empirical results. Finally, Section IV concludes with a summary and future directions.

II. METHODS

For generating monthly economic reports, we take advantage of the news articles we have been collecting on the Web to compute the business sentiment index. As a pilot study, we propose the following two-stage summarization framework.

```
The following news sentences belong to the same topic
according to the results of topic analysis. The number at the
end of each sentence indicates its impact on the economy.
Please generate an appropriate and concise label for each
topic, and then summarize approximately three key points that
are particularly important from the perspective of economic
impact.

Output in JSON format following the structure below.

### Output Format
{
  "label" : "Topic Label",
  "points" : [
    "A sentence describing a key point",
    "A sentence describing a key point",
    "A sentence describing a key point",
  ]
}

### News Sentences
- 1st news sentence (sentiment score)
- 2nd news sentence (sentiment score)
- ...
- n-th  news sentence (sentiment score)
```

Figure 1. First prompt to generate a topic summary.

1) Data Preparation: Compile the news fragments (sentences) for a selected month, each assigned a business sentiment score using a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) regressor [1]. The news articles were collected in advance from the Sankei Newspaper website.
2) Topic Identification: Apply the neural topic model, FASTopic [2], to identify k latent topics in the news data.
3) Topic-Based Summarization: For each topic,
   a) Identify n sentences with the highest absolute topic-sentiment scores, computed as $s_i \cdot p(t|d_i)$, where s_i is the sentiment score for sentence d_i, t is a topic, and $p(t|d_i)$ is the sentence-topic distribution from FASTopic.
   b) Generate its descriptive label and three-point summary based on the n sentences, using a Large Language Model (LLM). Specifically, we utilize Generative Pretrained Transformer 4-omni (GPT-4o) mini through the OpenAI Application Programming Interface (API). Figure 1 shows the English-translated prompt.
4) Overall Summarization: Combine the k topic labels and summaries to generate an overall monthly summary using GPT-4o mini with a prompt shown in Figure 2.

```
The following summarizes recent events for each of the k
topics that have a significant impact on the economy. Each
topic is assigned an appropriate label. Generate a summary of
approximately 300 characters for the entire content.

The output should be in JSON format, following the structure
below.

### Output Format

{
  "summary": "Summary of the entire content"
}

### Topic-Specific Report (in JSON format)
[k outputs from the previous step]
```

Figure 2. Second prompt to generate an overall summary.

III. Experiments

A. Settings

We set $k = 5$ and $n = 100$ for the number of topics and selected sentences per topic, respectively. For the document embedding model used by the FASTopic model, we adopted "paraphrase-multilingual-mpnet-base-v2". Sankei news articles (in Japanese) from October 2024 were used as the input document set, with prompts (Figures 1 and 2) modified to generate English output. For reference, this month's news-based BSI is -0.17.

B. Results

The proposed framework identified the five topics with 100 news sentences per topic. The results were concatenated to the prompt in Figure 1 and were passed to the LLM. The following shows the generated labels for the topics, where the numbers in the parentheses indicate the topic-sentiment distribution, defined as the expected value of topic-sentiment $s \cdot p(t|d)$ over all documents D, i.e., $\mathbb{E}_{d_i}[s_i \cdot p(t|d_i)] = \frac{1}{|D|}\sum_i s_i \cdot p(t|d_i)$. The topic-sentiment distribution indicates the overall business sentiment of each topic. Note that the prompt also generated a three-point summary for each topic but omitted here due to the page limitation.

- Crime and Security Issues (-0.64)
- Economic Performance Overview (-0.19)
- Despair and Isolation (-0.41)
- Energy and Infrastructure Development (-0.20)
- Security and Geopolitical Tension (-0.58)

These outputs were then concatenated to the second prompt in Figure 2 and were given to the LLM, generating the following summary explaining the underlying factors of the month's BSI:

Recent events have revealed a troubling intersection of crime, economic performance, and geopolitical tensions affecting public sentiment. Rising violent crime and failures in police oversight have heightened societal fears, while economic data shows strong sales juxtaposed with declining consumer confidence due to rising costs. Alongside this, feelings of despair and isolation are on the rise, impacting mental health and productivity. In the face of mounting energy demands driven by digital advancements, there's a pressing need for sustainable infrastructure, notably nuclear power, despite the historical challenges faced. Finally, Japan's complex security environment calls for robust foreign policies as geopolitical tensions escalate, particularly due to the implications of Russia's actions and China's assertiveness, threatening both regional and global stability.

For comparison, we devised a simpler baseline which sent m news sentences with the highest absolute sentiment scores to the LLM without topic sentiment analysis, where we set $m = 500$ such that the same number of sentences as our framework were considered. The baseline summary is as follows:

The news articles present a mixed picture regarding economic impacts. On one hand, some reports indicate a significant increase in consumer activity and investment, such as in advertising and sales of certain products, along with a strong recovery in tourism from abroad. Conversely, there are alarming signs of economic decline, with a dramatic rise in bankruptcies, particularly among small businesses, and growing concerns over rising costs and declining consumer confidence. Overall, while some sectors appear to thrive, many are struggling under pressures of financial instability and societal challenges.

The baseline produced a concise but narrower economic overview, focusing on immediate trends. In contrast, our approach by design offered a more comprehensive analysis, incorporating societal, geopolitical, and infrastructural factors. Together with the three-point summary for each topic (not shown), our report would be particularly valuable for stakeholders seeking a holistic perspective on economic conditions.

IV. Conclusions and Future Work

This paper presented a framework for generating concise yet comprehensive economic reports to enhance the interpretability of a news-based business sentiment index [1]. By integrating topic sentiment analysis with LLM-based text generation, our two-step summarization approach produced more comprehensive summaries compared to a simpler, one-step method. Preliminary experiments demonstrated the potential of this approach in capturing diverse factors influencing economic sentiment, offering valuable insights for policymakers and analysts.

In future work, the proposed framework will be integrated into our web-based business sentiment nowcasting system.[1]

References

[1] K. Seki, Y. Ikuta, and Y. Matsubayashi, "News-based business sentiment and its properties as an economic index," *Information Processing & Management*, vol. 59, no. 2, Article 102795, 2022, Article No. 456.

[2] X. Wu, T. T. Nguyen, D. C. Zhang, W. Y. Wang, and A. T. Luu, "FASTopic: Pretrained transformer is a fast, adaptive, stable, and transferable topic model," in *Proceedings of the Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[1] https://sapir.apir.or.jp

An Architectural Framework for Consistent UI in Android App Development

Abdul-Rahman Mawlood-Yunis
Department of Computer Science and Physics
Wilfrid Laurier University
Waterloo, Canada
amawloodyunis@wlu.ca

***Abstract*—The User Interface (UI) is an essential component in the development of interactive apps and web applications. In this study, we present an architectural framework designed to simplify the creation of a consistent UI for Android apps. The approach is both straightforward and innovative, utilizing well-established object-oriented programming concepts, such as abstraction and inheritance, to enable the development of flexible and scalable apps. Apps built using this framework are developed by extending abstract and generic concepts to ensure uniformity across the entire interface. We outline the key components of this architecture, provide sample code for implementation, and present an app developed using this framework to highlight its practical benefits. This paper offers two main contributions: accelerating app development and enabling the creation of consistent user interface designs that enhance both visual appeal and overall user experience. While the framework is initially designed for Android app development, its applicability extends to web development and can be used by tools focused on minimizing code complexity while supporting multiplatform compatibility, including web applications.**

Keywords-Software design and architecture; app development; design pattern; software engineering; software reuse; Android; Web development

I. INTRODUCTION

The user interface (UI) holds significant importance in interactive apps and application development, particularly in mobile apps. An attractive and user-friendly visual interface becomes increasingly crucial and plays an essential role in determining their success [1]–[3]. Extensive research in software engineering [4]–[8], design patterns [9]–[13], software architecture [14]–[16], human interaction [17]–[19], and related fields have been dedicated to the proper UI design for interactive apps and applications. Leading APIs [20] offer a range of components and classes to facilitate this goal [21]–[23]. For instance, the Android API provides the Fragment component [24], which enhances modular app development and UI flexibility. The Fragment is a reusable, self-contained portion of an activity's or UI screen that can be added, removed, or replaced dynamically, allowing for a more modular approach to app design. By utilizing the Android Fragment component, developers can design and develop each page or view of the app separately, promoting efficient reuse and customization. This is because Fragments allow developers to break down the UI into smaller, independent sections that can be reused across different parts of the app. For example, a Fragment developed for one screen (e.g., a login form) can be reused in multiple screens that require similar functionality, reducing redundancy and speeding up development. Additionally, Fragments can be customized to display different content or behavior based on the context, such as device orientation or user interaction, providing greater flexibility. Fragments achieve this modularity by serving as containers that can have UI elements and logic, which can be embedded into different parts of the app's layout. They allow developers to partition the app screen into multiple independent areas, each capable of hosting its own content or functionality. These Fragments can be attached to an activity's view and positioned to occupy one or more sections of the screen. By reusing these Fragments across multiple views, developers can maintain flexibility while reducing redundancy in the UI design, which leads to more efficient development and maintenance. This modular design also makes it easier to update or change individual parts of the UI without affecting the entire screen, further streamlining the development process.

While Fragment work focuses on designing a portion of the page or view to be reused, this work takes a broader approach by focusing on reusing the whole or important components of the page as the user navigates between different screens of an app. In other words, this work develops an architectural framework that enables persistent UI across app screens, ensuring a cohesive user experience as users move from one screen to another. This framework can be utilized throughout the app development process, allowing developers to create a consistent and dynamic interface that adapts to various contexts and navigation flows.

The suggested architecture does not serve as a replacement for the use of Fragment in Android app development, as Fragments offer more functionality beyond screen reuse. However, our proposed architecture surpasses Fragments when it comes to screen reuse. Here, the focus isn't solely on creating a sizeable component for reusing across app pages. Instead, we describe a framework for reusing as many components as needed across app pages and enable each component to function differently on various pages. This will be accomplished by developing a generic and abstract component containing the components to be reused and operating differently across app pages. The framework will accelerate the app development process and facilitate the creation of persistent UI creation, thereby improving the overall appearance and user experience of the apps. Moreover, the framework's applicability is not confined to Android-enabled devices. Although initially designed for Android app development, it can also be adapted for web development and tools aimed at reducing code complexity while ensuring compatibility across multiple platforms,

 ISBN: 978-1-68558-243-2

including both mobile and web applications. Additionally, the knowledge and principles derived from this framework are transferable to a wide range of application development contexts, thereby contributing to the broader advancement of software engineering practices.

The paper is organized as follows. Section II provides an overview of the proposed architecture. Section III includes implementation examples for all components of the architecture. In Section IV, we present an app developed using this architecture and discuss the testing conducted to validate its effectiveness. Finally, Section V concludes the paper and highlights potential directions for future research.

II. Overview of The Proposed Architecture

In this section, we describe the building blocks of the proposed architecture and their main responsibilities.

A. Base View

The first step is to create a base class (also called a superclass) in Object-Oriented Programming (OOP) that acts as the foundational template for the app's screens. This base class provides a structure that defines the common components and functionality that all the screens in the app will share. However, the base class itself is abstract, meaning that it is not intended to be directly used or displayed. Instead, it serves as a container for shared elements, providing a common foundation upon which all individual screens will be built.

Each screen in the app extends this base class, meaning that the screen inherits the structure and components from the base class. However, each screen can customize or define the inherited components to suit the specific needs of that screen. This allows for consistency across screens (through shared components), while still allowing flexibility and customization in terms of design and functionality.

To define the generic view and shared behavior for all screens, the base class includes two key methods:

1) An abstract method: This method has no implementation in the base class itself, but must be implemented by each subclass (screen). It acts as a placeholder for screen-specific functionality.
2) A regular method with an empty body: This method is defined in the base class, but does not perform any actions initially. Subclasses can choose to override this method or leave it as is, depending on their needs.

Additionally, the base class defines a layout that includes UI components (such as buttons, text fields, or navigation elements) that need to appear across all the app's screens. By placing these components in the base class, every screen that extends it will automatically inherit these shared elements. This ensures that there is a consistent look and feel throughout the app while still allowing each screen to define its own unique content.

By using this approach, developers can create custom screens (subclasses) that fit their specific needs, while maintaining a persistent look (consistent UI elements) and shared functionality (common methods and components) across the app. This promotes reusability, reduces redundancy, and simplifies the overall app development process.

B. Derived Views

Derived views are concrete implementations or materialized views that extend the abstract base view. In other words, these views represent actual screens or pages in the app, whereas the base view acts as a template or blueprint. To create a new screen (view) for the app, a new page needs to be designed. This new page inherits components, methods, and basic layout properties from the base view, but it also adds custom components and functionality to create a unique screen tailored to its specific needs. The process of implementing a derived view involves the following steps:

1) Reusing the base layout container: The first step is to reuse the layout container defined in the base view. This container holds only the components that are common and need to exist on every page. These are shared elements, such as headers, footers, or navigation bars, that appear across multiple screens. By reusing this container, you ensure that these consistent elements are present on every screen, helping to maintain a uniform look and feel across the app.
2) Creating a new layout for the derived view: Next, you create a custom layout for the derived view. This layout will contain the specific components and content for this screen. You then place this new layout into the space inherited from the base layout container. The inherited space ensures that your derived layout fits into the structure and design already established by the base view, preserving the overall app's consistency while allowing for unique content on each screen.
3) Implementing abstract methods: The base view has abstract methods—these are methods that have no implementation in the base view itself but must be implemented in the derived view. These methods act as placeholders, requiring the derived view to define specific behavior. For example, an abstract method in the base class returns the content view, and the derived view provides the logic for how that should be done for its specific screen.
4) Completing empty methods inherited from the base view: In addition to abstract methods, the base view has regular methods with an empty body. These methods are already given a structure but do not perform any actions in the base view. The derived view must complete the implementation of these empty methods by adding the necessary logic. For example, the base class might define a method to initialize UI elements, and the derived view would add specific code to populate those elements with data or behavior relevant to the screen it represents.

C. Customizing Views Behaviour

While every page of an app serves distinct purposes and boasts unique features, reusing components from the base view ensures a consistent appearance and functionality. A decision must be made regarding reusable components, as each page's requirements must be developed accordingly. For instance, a toolbar component can be placed in the base view and inherited across all app pages, yet the action functionalities of each toolbar item may vary from one page to another. In other words, when the toolbar items are clicked, the actions executed on page one will differ from those performed on pages two, three, and beyond.

Figure 1 illustrates the proposed architecture, including an example of a component intended for reuse. The figure displays the following components:

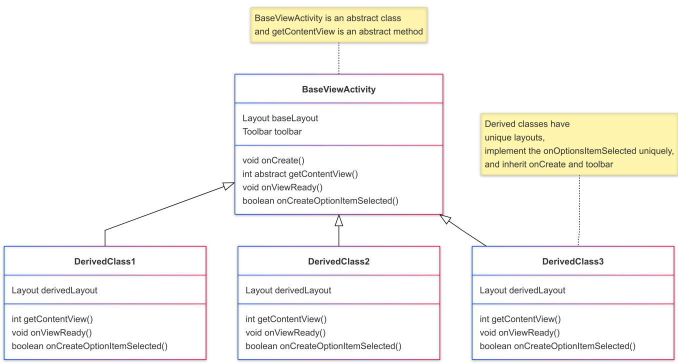


Figure 1. Class diagram for the proposed framework.

1) An abstract superclass featuring two properties: baseLayout, of type Layout, and toolbar, of type Toolbar. This class also encompasses four methods: onCreate(), onViewReady(), getView(), and onOptionMenuSelected().
2) Derived classes override the onViewReady(), getView(), and onOptionMenuSelection() methods, each providing its own implementation. Additionally, each derived class boasts a distinctive layout property, using getView() to access and initialize the local layout properties.

III. The Implementation of The Proposed Architecture

In this section, we will list the architecture components, explain their roles and usage in the overall design in more detail, and provide sample code to demonstrate how they can be implemented.

A. The Base View Implementation

As previously mentioned, a key element of this architecture is the base view. In Android development, this base view can be represented by an abstract class like BaseViewActivity, which extends AppCompatActivity. AppCompatActivity serves as a foundational class in the Android API, providing various built-in features utilized by screens. Consequently, it is extended and reused when new app screens are created.

The **BaseViewActivity** class is abstract and defines essential properties that are common to all apps, such as the layout and necessary components. The list 1 provides a template for defining the BaseViewActivity class. Additional details about other crucial methods and the app's layout are discussed in the following subsections. The ellipses (...) in Listing 1 represent the parts of the code that need to be implemented, which will depend on the specific requirements of the app.

```
public abstract class BaseViewActivity extends
     AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_base);
    ...
    onViewReady(savedInstanceState, getIntent());
  }
  // To be used by child activities.
  protected void onViewReady(Bundle savedInstanceState,
      Intent intent) {}
  protected abstract int getContentView ();
  @Override
  public boolean onOptionsItemSelected(MenuItem menuItem) {
   ...
 }
}
```

Listing 1. Base class template

More details and an overview of the base implementation are provided in the following.

1) OnViewReady Method: Within the BaseViewActivity class, we've established a method named onViewReady(), which is invoked within the onCreate() method subsequent to calling setContentView(). The method currently contains no code, leaving its implementation to be completed by the derived class. This enables derived classes or screens to append new widgets or modify the layout of the base class. The complete implementation of this method will be demonstrated when we define the derived classes in part B, item 2 of this section.

2) getContentView Method: We have defined another method called getContentView(). This method is an abstract method that needs to be implemented by the Derived classes. This method is a helper method and it will be used by the onViewReady() method to update or make changes to the layout of the base class.

3) Base Layout or Base Content: A layout for the BaseViewActvity class or its content is created and is shown in Listing 2. The layout is an XML file that will be transformed into the programming code and integrated with the rest of the app code at run time. The code statement below inside the onCreate method of the BaseViewActvity class will take care of this step, i.e., setting the layout for the base view by calling this method. setContentView(R.layout.activity_base);

The layout has two mandatory parts. A component(s) that will be reused across all the screens of the app, in this case, is a toolbar, and an inner layout. The inner layout is empty and, in this case of type linear layout. In other words, no widgets or components are attached to the inner layout and when items are attached to this layout, they will be arranged sequentially from left to right top down and take the whole screen of the base view. These properties allow the derived screen, the classes that inherit this layout, to update its content and attach widgets to it. The layout is shown in Listing 2 where the toolbar and inner layouts are shown.

 ISBN: 978-1-68558-243-2

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/
android"
    xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".BaseViewActvity ">
    <androidx.appcompat.widget.Toolbar
android:id="@+id/in_base_my_toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize" />
    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/
            android"
        xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:id="@+id/toolbarIDInbase">
    </LinearLayout>
</LinearLayout>
```

Listing 2. The layout for the Base class

4) Reused Components: An important part of this architecture which aims to enable persistent looks and feel is component reuse. One or more components can be included in the base view and be reused in various screens of the app. To demonstrate this concept one component, a toolbar, is added to the base view layout and the statement below will add or set its corresponding object code to the definition of the BaseViewActvity class.

```
setSupportActionBar(toolbar);
```

In addition to reusing components, a crucial concept in this architecture is the ability to redefine the behavior of objects defined in the base view. An example of object behaviour definition for the toolbar is shown in Listing 3. This implementation serves as a default behaivour and it will be overridden in the derived class to behave differently and to become more relevant to the new context while using the same toolbar buttons and menu items defined in the base view.

```
public boolean onOptionsItemSelected(MenuItem menuItem) {
  int id = menuItem.getItemId();
  switch (id) {
    case R.id.action_one:
      Intent intent = new Intent(BaseViewActvity.this,
           ChildActivity.class);
      startActivity(intent);
      break;
    case R.id.action_two:
      snackbar.setText("You are at home").show();
      break;
    case R.id.action_three:
      Uri webpage = Uri.parse("https://www.canada.ca/");
      intent = new Intent(Intent.ACTION_VIEW, webpage);
      if (intent.resolveActivity(getPackageManager()) !=
           null) {
        startActivity(intent);
      }
      break;
    case R.id.action_about:
      Toast
          .makeText(this,
              "Version 1.0,"
                  + "developer_information",
              Toast.LENGTH_LONG).show();
  }
  Return true;
}
```

Listing 3. A behaviour implementation of a component in the base view that will be overridden in the derived classes

5) A Complete code for Base View: Combining all the code snippets provided above creates a comprehensive template for the base view class, featuring a single component, the toolbar, intended for reuse across the app's screens. Listing 4 illustrates this template.

```
public abstract class BaseViewActvity extends
     AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_base);
    Toolbar myToolbar = findViewById(R.id.
         in_base_my_toolbar);
    setSupportActionBar(myToolbar);
    onViewReady(savedInstanceState, getIntent());
  }
  protected void onViewReady(Bundle savedInstanceState,
       Intent intent) {
    // To be used by child activities.
  }
  protected abstract int getContentView();
  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    // invoked automatically by activity
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.main_activity_actions, menu);
    return true;
  }
  public boolean onOptionsItemSelected(MenuItem menuItem) {
    int id = enuItem.getItemId();
    switch (id) {
      case R.id.action_one:
        Intent intent = new Intent(BaseViewActvity.this,
             ChildActivity.class);
        startActivity(intent);
        break;
      case R.id.action_two:
        snackbar.setText("You are at home").show();
        break;
      case R.id.action_three:
        Uri webpage = Uri.parse("a_url");
        intent = new Intent(Intent.ACTION_VIEW, webpage);
        if (intent.resolveActivity(getPackageManager()) !=
             null) {
          startActivity(intent);
        }
        break;
      case R.id.action_about:
        Toast
            .makeText(this,
                "Version 1.0,"
                    + "developer_name",
                Toast.LENGTH_LONG)
            .show();
    }
    return true;
  }\\ end of the method
}\\ end of the class
```

Listing 4. A template for the Base View

B. Implementations of The Derived Views

Every useful app is made of more than one screen. Following the proposed architecture, as many screens as needed can be created with similar feels and looks by extending the base view and thus reusing the inherited components. To create new screens using the existing base view the following need to be done.

- Defining a new class(s) by extending the base class.
- Define layout(s) for the derived classes or screens to replace the empty inner layout from the base view.
- Complete the definition of the inherited methods.
- Define new behaviors for the inherited component.

These steps are described in the following subsection along with sample codes on how to implement them

1) Defining New Classes: To develop a complete app, you'll need to define one or more classes that utilize the base view, BaseViewActivity, as their superclass. The number of classes depends on the required screens. An example of such a derived class, DerivedClass1, is provided in listing 5.

DerivedClass1 inherits the onCreate() method from the base class instead of implementing it directly. This inheritance allows DerivedClass1 to replace the base view layout with its own. This substitution occurs because the onViewReady() method is invoked within the onCreate() method, which is triggered when the screen is being loaded. Refer to the BaseViewActivity class for the definition of the onCreate() method and the location of where onViewReady() is called.

It's worth emphasizing that loading the screen entails calling the inherited and hidden onCreate() method within the derived class, followed by invoking the onViewReady() method. This process ultimately leads to displaying a personalized layout for the newly created screen.

```
public class DerivedClass1 extends BaseViewActvity  {
  LinearLayout linearLayout;
    ...
}
```

Listing 5. A derived view class definition header

2) Define new layouts for derived classes: The code snippet presented in Listing 6 depicts the layout for a derived class. This layout constitutes the content of the newly created screen and differs from the layout defined in the base view. It's important to note that this serves as merely an example of a newly created screen, and the actual screen will vary depending on the specific requirements of the app. The provision of a placeholder for the layout in the main or base view allows each newly created screen to utilize this space and construct its desired screen while also benefiting from the reusable components at the same time.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/
        android"
    xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:layout_marginTop="40dp">
    <ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/ic_favorite_border_black_24dp"/>
</LinearLayout>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
```

Listing 6. A layout example for a derived class

3) Overriding The onViewReady Method: The onCreate method isn't explicitly defined in derived classes; however, it's inherited from the base class and is automatically invoked upon opening the derived screens, i.e., executing the derived class code. As the onViewReady method is invoked within the inherited onCreate method, this provides an opportunity to override it in a manner that allows for setting a distinct layout for the derived screens. In other words, the onViewReady method takes on the responsibility of the onCreate method inside the derived classes. This explains why the onViewReady method definition is empty in the base class, and code which typically placed inside the onCreate method is now placed inside the onViewReady method.

An implementation of onViewReady is illustrated in Listing 7, where a new layout is loaded by onViewReady to create a custom view for a derived screen.

```
@Override
protected void onViewReady(Bundle savedInstanceState,
     Intent intent) {
  setTitle("First Screen");
  super.onViewReady(savedInstanceState, intent);
  linearLayout = findViewById(R.id.baseLayout);
  LayoutInflater layoutInflater = LayoutInflater.from(
       DerivedClass1.this);
  layoutInflater.inflate(getContentView(), linearLayout,
       true);
}
```

Listing 7. onViewReady implementation by a derived class

The code in the Listing 7. Does the following:

a. Get the container, the linear layout container, that has been defined in the layout for the base class. It is an empty container, and we can add views to it. This is achieved using this code statement:
findViewById(R.id.baseLayout);
b. The retrieved base linear layout is used to initialize a local liner layout instance variable of the derived classes. This is achieved using this code statement:
linearLayout = findViewById(R.id.baseLayout);
c. The retrieved layout from the previous step is put inside the empty linear layout container from the base class. This is achieved using the following lines of code:

```
LayoutInflater layoutInflater =
 LayoutInflater.from(DerivedClass1.this);
layoutInflater.inflate(getContentView(),
   linearLayout, true);
```

4) Overriding the getView Method: The getContentVeiw is an accessor method used by the onViewReady method to retrieve the locally defined layout for a derived class. The method definition is presented in Listing 8 and its usage is shown in Listing 7.

```
@Override
protected int getContentView() {
 // layout for a derived class
 return R.layout.activity_main;
}
```

Listing 8. getContentView implementation

5) Override the behavior of shared components: In the example under consideration here, we opted to utilize the toolbar consistently across the derived views while altering its functionality by overriding the onOptionItemSelected() method. Essentially, this means that when the toolbar items are clicked, the actions performed on each derived screen can be different from the default behavior and each other. The code snippet in Listing 9 illustrates such an implementation, where the code contained within each switch case is different from

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-243-2

what has been included within the switch statement for the base classes and each derived class. It's important to mention that the onOptionItemSelected method can be implemented within the base view. This implementation will act as a default behavior inherited by derived classes. Alternatively, it can be left empty, like the layout in the base view. In either scenario, the method can be overridden in the derived classes to offer customized functionality. In this example, we've chosen to provide a default implementation in the base view class.

```
public boolean onOptionsItemSelected
  (MenuItem menuItem) {
  int id =menuItem.getItemId();
  String phoneNumber = "613 000 0000";
  switch (id) {
  case R.id.action_one:
      Intent intent = new Intent(Intent.ACTION_DIAL);
      intent.setData(Uri.parse("tel:" + phoneNumber));
     if (intent.resolveActivity(getPackageManager()) !=
         null) {
           startActivity(intent);
       }
   break;

 case R.id.action_two:
 intent = new Intent(ChildActivity.this,
                  NewMainActivity.class);
   startActivity(intent);
  break;

 case R.id.action_three:
  Uri webpage = Uri.parse( "a_url");
  Intent i = new Intent(Intent.ACTION_VIEW);
  i.setData(webpage);
   startActivity(i);
  break;

case R.id.action_about:
     Toast.makeText(this,
      "Version 1.0," + " Developeer_information",
       Toast.LENGTH_LONG).show();
  }
   return true;
}
```

Listing 9. Toolbar behaviour implementation for a derived class

6) An Example of a Derived Class Implementation: The code provided in Listing 10 serves as an instance of a fully developed derived class, encompassing the steps described from 1 to 5.

IV. RUNNING AND TESTING THE PROPOSED ARCHITECTURE

To evaluate the proposed architecture, we developed an app that can be downloaded at [25]. The app was created in Java using Android Studio. Using this app, two sets of tests were conducted.

In the first set, multiple screens were created. Consistent with the proposed architecture, none of these screens directly utilized the 'onCreate()' method, which serves a role similar to the constructor method in object-oriented programming and is essential for object instantiation. Instead, all screens inherited the 'onCreate()' method from an abstract base view and implemented the 'onViewReady()' method locally to instantiate three distinct screens. These screens are depicted in Figs. 2-4, and the concept of attaching a customized page to the base page is illustrated in Figure 5.

```
public class ChildActivity extends BaseViewActvity {
    LinearLayout linearLayout;

    @Override
    protected int getContentView() {
        return R.layout.activity_child;
    }

    @Override
    protected void onViewReady(Bundle savedInstanceState,
         Intent intent) {
        super.onViewReady(savedInstanceState, intent);
        linearLayout = findViewById(R.id.baseLayout);
        LayoutInflater layoutInflater = LayoutInflater.from
             (ChildActivity.this);
        layoutInflater.inflate(getContentView(),
             linearLayout, true);
    }

    public boolean onOptionsItemSelected(MenuItem menuItem)
          {
        int id = menuItem.getItemId();
        String phoneNumber = "613 000 0000";
        switch (id) {
        case R.id.action_one:
            Uri webpage = Uri.parse("a_url");
            Intent intent = new Intent(Intent.ACTION_VIEW,
                 webpage);
            if (intent.resolveActivity(getPackageManager())
                 != null) {
                startActivity(intent);
            }
            break;
        case R.id.action_two:
            intent = new Intent(ChildActivity.this,
                 NewMainActivity.class);
            startActivity(intent);
            break;
        case R.id.action_three:
            intent = new Intent(Intent.ACTION_DIAL);
            intent.setData(Uri.parse("tel:" + phoneNumber))
                 ;
            if (intent.resolveActivity(getPackageManager())
                 != null) {
                startActivity(intent);
            }
            break;
        case R.id.action_about:
            Toast.makeText(this, "Version 1.0," + "
                 Developer_information", Toast.LENGTH_LONG).
                 show();
        }
        return true;
    }
}
```

Listing 10. An instance of a fully developed derived class

Second, following the approach used for screen creation, the proposed architecture was employed to develop a reusable component. This component can be inherited and integrated across all screens, ensuring a consistent and persistent user interface throughout the application. As demonstrated in the app screens, each screen features a uniform toolbar, depicted in Figure 6.

Although the toolbars appear identical across screens, their functionality varies significantly, adapting to the specific requirements of each page. For example, on the first screen, selecting toolbar items 1, 2, 3, and 4 (as shown in Figure 6) triggers the following actions:

1. Opens a second page, illustrated in Figure 4.
2. Displays a message indicating that you are on the home page.

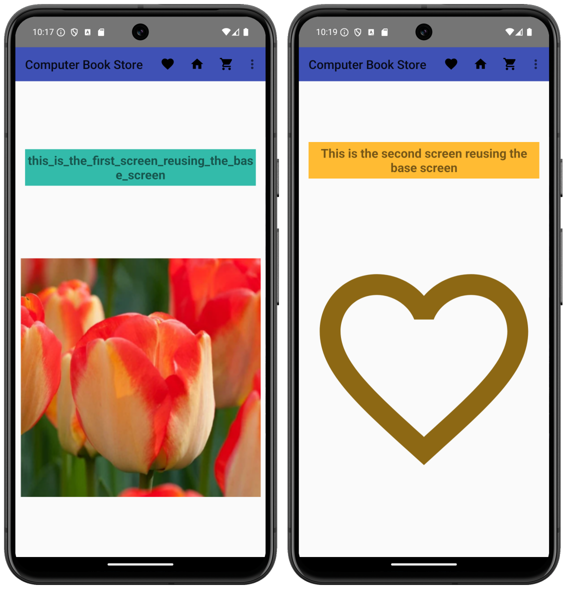


Figure 2. and 3, examples of two pages created using base page

Figure 4. Another example of custom page using the base page

3. Launches a web page on the internet; an example is shown in Figure 7.
4. Opens an ”About” window, providing information such as details about the app’s author.

On the second and third screens, the functionality of these toolbar items differs. For instance, on the second page, clicking:

1. Opens a new app page. This demonstrates the consistent use of the first toolbar item to open a new app page.
2. Navigates back to the home page or the previous page.
3. Launches a phone-dialing interface, as shown in Figure 8, displaying how the same toolbar item can trigger different functionalities depending on the page.

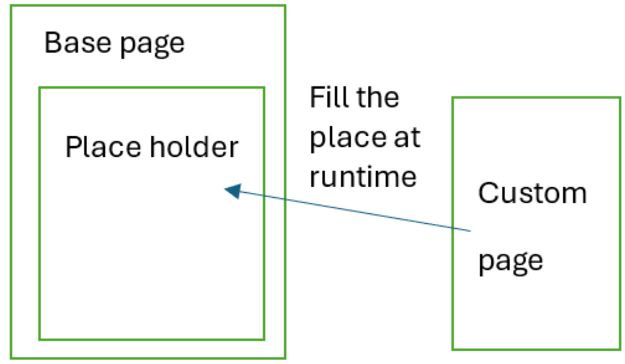


Figure 5. Attaching a customized page into the base page

Figure 6. Page toolbar and its items

4. Opens an ”About” window, presenting the same content presented on the first screen. This illustrates the option to assign a common functionality to a toolbar item across all screens.

The navigation flow described above is summarized in Figure 9. The ability to maintain a consistent user interface across all screens while enabling contextual behavior through the extension of a generic screen template highlights the feasibility and flexibility of the proposed architecture for app and application development.

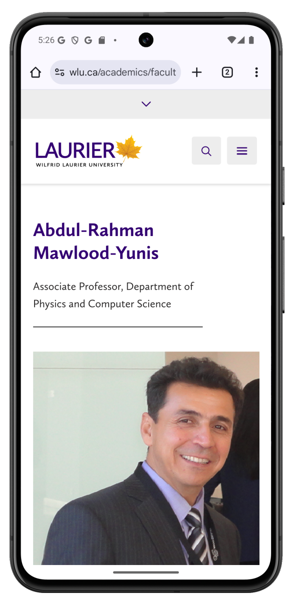


Figure 7. An example of web page lunched by clicking third item on the first page

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-243-2

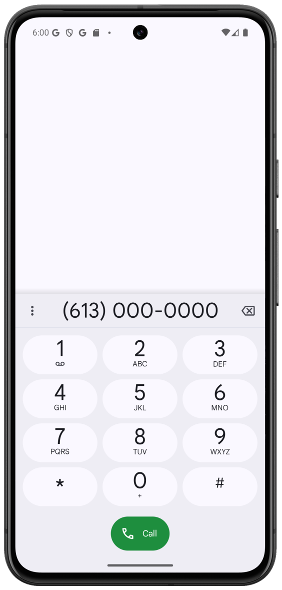


Figure 8. Example of phone-dialing interface

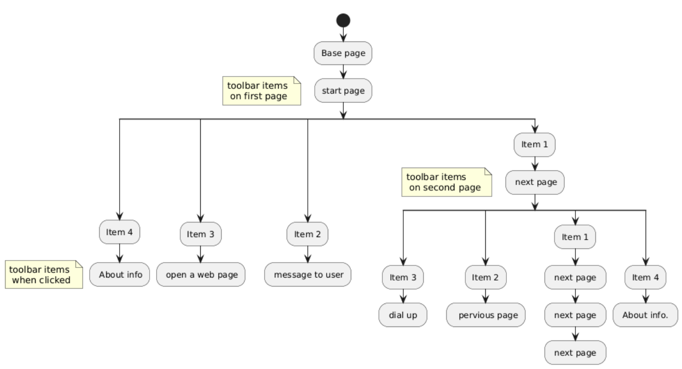


Figure 9. The app structure and component functionalities

V. Conclusion and Future Works

Drawing upon techniques from object-oriented programming, we introduced an app architecture aimed at facilitating persistent UI development in Android apps, thus enhancing the overall appearance and user experience of apps, an essential factor for their success. Detailed descriptions of the architecture components, along with an example of how to implement each component, have been provided. Furthermore, we tested the effectiveness of the architecture by building an app, highlighting its advantages and feasibility.

It is important to note that the applicability of the framework is not limited to devices enabled with Android. Although initially designed for Android app development, it can also be adapted for web development and used by tools and frameworks that aim to reduce code base and support multiple platform development, including mobile and web applications. Additionally, the knowledge and principles derived from this framework are transferable to a wide range of application development contexts, thereby contributing to the broader advancement of software engineering practices.

This work can be expanded in various ways. Much like the way we extended the base view using a single hierarchy, each derived class can undergo further extension, using multilevel class inheritance. This approach enables the development of complex and sophisticated applications with fewer codes and a persistent user interface.

We chose a toolbar to demonstrate the ability to define a component once and reuse it across different pages, each page featuring unique functionalities for the component. In future iterations, one of the toolbar items could help navigate between different app pages. Clicking on a toolbar item might activate a pop-up menu that displays all navigation options. We consider this to be viable future work, as smooth navigation is crucial for any app's success.

We believe that this work lays the groundwork for further utilization of object-oriented programming features in Android app development, enabling efficient app design and implementation. By fostering consistency, scalability, and reusability, this architecture not only accelerates development, but also paves the way for innovative advancements in both mobile and web application development.

References

[1] C. Zarmer and J. Johnson, "User interface tools: Past, present, and future trends," Hewlett-Packard Laboratories, 1990.

[2] I. Qasim, F. Azam, M. W. Anwar, H. Tufail, and T. Qasim, "Mobile User Interface Development Techniques: A Systematic Literature Review," in 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2018, pp. 1029–1034, doi: 10.1109/IEMCON.2018.8614764.

[3] H. Hu, et al., "Automated Mapping of Adaptive App GUIs from Phones to TVs," ACM Transactions on Software Engineering and Methodology, vol. 33, no. 2, pp. 1–31, 2024, doi: 10.1145/3459610.

[4] H.-W. Six and J. Voss, "A software engineering perspective to the design of a user interface framework," in [1992] Proceedings. The Sixteenth Annual International Computer Software and Applications Conference, Chicago, IL, USA, 1992, pp. 128–134, doi: 10.1109/CMPSAC.1992.217591.

[5] C. E. Wills, "User interface design for the engineer," in Proceedings of ELECTRO '94, Boston, MA, USA, 1994, pp. 415–419, doi: 10.1109/ELECTR.1994.472682.

[6] V. Chernikov, "Approach to Rapid Software Design of Mobile Applications' User Interface," in 2018 23rd Conference of Open Innovations Association (FRUCT), 2018, pp. 1–7, doi: 10.23919/FRUCT.2018.8588030.

[7] M. J. Tsai and D. J. Chen, "Generating user interface for mobile phone devices using template-based approach and generic software framework," Journal of Information Science and Engineering, vol. 23, no. 4, pp. 1189–1211, 2007.

[8] A. Marcus, "User interface design and culture," in Usability and Internationalization of Information Technology, vol. 3, pp. 51–78, 2005.

[9] J. Arifin, E. M. Fasha, and M. A. Ayu, "User Interface Design Patterns for Marketplace Mobile Application," in IEEE 7th International Conference on Computing, Engineering and Design (ICCED), 2021, pp. 1–6, doi: 10.1109/ICCED53389.2021.9664843.

[10] E. G. Nilsson, "Design patterns for user interface for mobile applications," Advances in Engineering Software, vol. 40, no. 12, pp. 1318–1328, 2009.

[11] S. Hoober and E. Berkman, "Designing mobile interfaces: Patterns for interaction design," O'Reilly Media, Inc., 2011.

[12] T. Neil, "Mobile design pattern gallery: UI patterns for smartphone apps," O'Reilly Media, Inc., 2014.

[13] I. C. Morgado and A. C. Paiva, "The impact tool: Testing UI patterns on mobile applications," in 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2015, pp. 876–881.

[14] P.H.J. Chong, P.L. So, P. Shum, X.J. Li and D. Goyal, “Design and implementation of user interface for mobile devices,” IEEE Transactions on Consumer Electronics, vol. 50, no. 4, pp. 1156–1161, 2004, doi: 10.1109/TCE.2004.1362513.
[15] M. J. Tsai and D. J. Chen, “Generating user interface for mobile phone devices using template-based approach and generic software framework,” Journal of Information Science and Engineering, vol. 23, no. 4, pp. 1189–1211, 2007.
[16] S. Wendler and D. Streitferdt, “The Impact of User Interface Patterns on Software Architecture Quality,” in The Ninth International Conference on Software Engineering Advances (ICSEA 14) IARIA, 2014, pp. 134–143.
[17] B. Shneiderman and C. Plaisant, “Designing the user interface: Strategies for effective human-computer interaction,” Pearson Education India, 2010.
[18] T. Mandel, “User/System Interface Design,” Encyclopedia of Information Systems, vol. 1, pp. 1–4, 2002.
[19] G. Briones-Villafuerte, A. Naula-Bone, M. Vaca-Cardenas, and L. Vaca-Cardenas, “User Interfaces Promoting Appropriate HCI: Systematic Literature Review,” Revista Ibérica de Sistemas e Tecnologias de Informação, no. E47, pp. 61–76, 2022.
[20] I. O. Suzanti, N. Fitriani, A. Jauhari, and A. Khozaimi, “REST API implementation on Android-based monitoring application,” in Journal of Physics: Conference Series, vol. 1569, no. 2, p. 022088, July 2020.
[21] S. Gao, L. Liu, Y. Liu, H. Liu, and Y. Wang, “API recommendation for the development of Android App features based on the knowledge mined from App stores,” Science of Computer Programming, vol. 202, p. 102556, 2021.
[22] Y. Wang, H. Liu, S. Gao, and X. Tang, “Animation2API: API recommendation for the implementation of Android UI animations,” IEEE Transactions on Software Engineering, vol. 49, no. 9, pp. 4411-4428, 2023.
[23] Google, “User Interface - Android Developers,” retrieved: [February, 2025]. Available: https://developer.android.com/develop/ui.
[24] Google, ”Fragment - Android Developers,” retrieved: [February, 2025]. Available: https://developer.android.com/reference/android/app/Fragment.
[25] ”Architectural Framework Based App” retrieved: [February, 2025]. Available: http://bohr.wlu.ca/amawloodyunis/framework/ArchitecturalFramework.zip.

A Web Service Tool for Real Estate Price Estimation Powered by Machine Learning

Youssef Roman*, Abdul-Rahman Mawlood-Yunis†
Department of Computer Science and Physics, Wilfred Laurier University, Waterloo, Canada
e-mail: roma0130@mylaurier.ca*, amawloodyunis@wlu.ca†

***Abstract*—Accurate housing price estimation is critical for informed decision-making in the real estate industry. This study develops a Flask-based web platform that integrates machine learning (ML) algorithms to predict housing prices using a dataset of 7,000+ detached home transactions from Ontario's Halton Region (2022–2023). The system applies advanced ML techniques and feature engineering, incorporating economic indicators such as prime rates to enhance prediction accuracy. It enables real-time data exploration, visualization of transaction patterns, and analysis of market shifts driven by interest rate fluctuations. Linear Regression, Random Forest, and XGBoost were evaluated, achieving R^2 values between 0.93 and 0.997, with Random Forest demonstrating the best balance of predictive performance and overfitting resistance. The models are deployed via a user-friendly web application, allowing users to estimate home prices across the Greater Toronto Area (GTA) based on key property features. By leveraging ML, the tool enhances transparency and efficiency in the real estate market, providing homebuyers, sellers, and investors with a reliable, accessible, and data-driven valuation solution.**

***Keywords- Machine learning*; *Web services*; ; *Feature engineering*; *Real estate*; *House price estimation*.**

I. Introduction

Accurate house price estimation is essential for stakeholders, such as homeowners, developers, investors, and appraisers, as property values are closely tied to economic conditions. Reliable forecasts not only facilitate informed decision-making in real estate transactions but are also critical for ensuring market stability and guiding investment planning.

Previous research has explored various machine learning (ML) techniques, including Linear Regression [1]–[3], Random Forest [2], [4], [5], and Recurrent Neural Networks (RNNs) [6], to forecast housing prices. These studies underscore the importance of feature engineering and the use of diverse algorithms for improved accuracy [7]. For example, approaches like Linear Regression and ensemble methods such as XGBoost have demonstrated promise by incorporating factors like LSTAT (Lower Status of the Population) scores and crime rates per capita [5]. Deep learning methods—such as Logistic Regression, Convolutional Neural Networks, and Long Short-Term Memory (LSTM) networks—have also been employed to predict housing prices using both property characteristics and time-series data [8]. Additionally, time-dependent factors have been analyzed using Auto-Regressive and Moving Average (ARMA) models [8], further highlighting the importance of temporal patterns in price forecasting.

Despite these advances, much of the existing literature does not fully account for external market conditions, such as changes in *prime rate* which are critical for understanding housing price fluctuations. This study aims to address that gap by integrating external economic factors into the price forecasting model, providing a more comprehensive approach. By leveraging both property-specific attributes and broader economic indicators—such as the Bank of Canada's *prime rate*, *localized price per square Foot metrics*, and *walk score* ratings—our model enhances the accuracy and relevance of housing price predictions for the Canadian market.

This research focuses on developing a web-based application using Flask, a Python web framework, to provide real estate information for the Greater Toronto Area (GTA). The key aspects of this application are:

1) Flask Framework: The application is built using Flask, which is lightweight and flexible, making it ideal for creating web services quickly and efficiently.
2) Purpose: The web service aims to provide users with easy access to real estate insights, helping them make informed decisions about property purchases or investments.
3) Data Source: The application utilizes data from various regions within the GTA, ensuring comprehensive coverage of the local real estate market.
4) Functionality:
    - Users can input specific property details. The application provides real-time price estimates based on the input
    - It presents three comparable properties to aid in decision-making
5) Real-time Insights: By connecting property characteristics with current market trends, the application offers valuable, up-to-date information to users.
6) User-Friendly Interface: The platform is designed to be intuitive and easy to use, making it accessible to various real estate stakeholders.
7) Practical Application: The tool bridges the gap between research findings and practical real estate decision-making, allowing users to apply insights directly to their property-related choices.

This application demonstrates the practical application of data science and web development in the real estate sector, providing a valuable resource for both professionals and individuals interested in the GTA property market.

The remainder of the paper is organized as follows: In Section II, we discuss the Data Acquisition and Preprocessing. Section III presents the feature engineering process. Section IV details the ML models's results and evaluation, highlighting their predictive performance. Section V details the web-based application implementation, emphasizing its usability and design. SectionVI details the exploratory data analysis and insights. Finally, Section VII concludes the study and outlines potential future work.

II. Data Acquisition and Prepossessing

The dataset employed in this study is derived from the REALM MLS Software [9], specifically targeting sold detached homes within the Halton Region for the years 2022 and 2023 [9]. The Halton Region, encompassing the cities of Oakville, Burlington, Milton, and Halton Hills, serves as the focal geographical area for this analysis. The dataset contains over 7,000 records, providing a substantial basis for a comprehensive examination of the real estate market trends and dynamics in this region during the specified period. Table I presents a brief description of the dataset features. Each row in the table represents a home feature and its corresponding description obtained from the REALM MLS records. The additional Engineered Features such as Canadian Prime Rates and Price per square foot (PPSQFT) are mentioned in section III.

TABLE I
Housing Features and their Descriptions

Feature	Description
Address	This column represents the street address or location of the detached home within the Halton Region.
Beds	Indicates the number of bedrooms in the detached home.
Washrooms	Specifies the number of bathrooms (including full and half) in the detached home.
Property Type	Describes the type or style of the detached home, such as bungalow, two-story, or split-level.
Sold Price	Reflects the final selling price of the detached home at the time of sale.
SqFt	Represents the total square footage of the detached home, indicating its size or living area.
MLS#	Stands for Multiple Listing Service number, a unique identifier for the property within the REALM MLS Software system.
Sold Date	Indicates the date when the detached home was sold.
Approx Age	Represents the approximate age or year of construction of the detached home, providing insight into its construction period.

III. Feature Engineering

This section outlines the enhancements made to the real estate dataset to improve insights into property pricing and market trends while optimizing its applicability for machine learning models. These enhancements include:

A. Feature Additions

The following new features are added to the dataset.

- Price Per Square Foot (psqft): Calculated by dividing the total property price by its square footage. Offers valuable insights into property pricing trends across different sizes and locations.
- Prime Rates: Reflects the percentage of the Central Bank of Canada's prime rate at the time of sale. Provides crucial context about the economic environment and interest rate conditions during each transaction.
- Walk Score: Added using the Redfin Walk Score API [10]. Measures property location walkability on a scale from 0 to 100. Higher scores indicate greater accessibility to amenities like shops, schools, parks, and public transportation. Categorized as follows:
  1) Car-Dependent: 0-50
  2) Somewhat Walkable: 50-70
  3) Very Walkable: 70-89
  4) Walker's Paradise: 90-100

B. Feature Transformation

Beyond incorporating new features into the dataset, the following data transformations and feature engineering techniques have been applied:

- Label Encoding: Applied to the 'House Type' and 'City' features to support the integration of categorical data into machine learning algorithms.
- City Name Extraction: A parsing algorithm is used on the 'Address' column to extract city names (Oakville, Milton, Burlington, Halton Hills) and store them in a new 'City' column.
- The 'Sqft Numeric' column is created by converting square footage ranges into average values. Additionally, for properties with missing square footage values, the mean square footage of the dataset is used as an imputed value to ensure completeness.
- Age Feature Transformation: Age ranges are converted into average age values, with 'New' properties assigned a value of 0.

These enhancements collectively contribute to a more robust and informative dataset, enabling more accurate and contextually informed analysis of the real estate market. Table II summarizes the enriched and engineered features, offering a comprehensive overview of the dataset improvements.

TABLE II
Enriched And Engineered Features

Feature	Description
Age Numeric	Represents the approximate age or year of construction of the detached home, provided as a numeric value for analysis.
Bedrooms	Total number of bedrooms in the detached home, including basement bedrooms, represented as a numeric value.
ppsqft	Price per square foot, calculated as the selling price of the detached home divided by its total square footage, providing insight into pricing dynamics based on size.
Canada's Prime Rates	Percentage of the Central Bank of Canada's Prime rate at the time of sale, offering context on economic conditions during the sale period.
Walk Score	A score from 0 to 100 indicating the walkability of the property's location, influencing its attractiveness and value.

IV. Price Prediction and Model Evaluation

During the ML model development phase, multiple models were built to predict housing prices, each with unique advan-

 ISBN: 978-1-68558-243-2

tages and trade-offs. The models developed included Linear Regression, Lasso Regression, Decision Tree, Random Forest, XGBoost, and Support Vector Machine (SVM). These models represent a diverse range of machine learning techniques, from simple linear models to more complex ensemble and kernel-based methods. Model evaluations were based on a 20% test split, using Root Mean Squared Error (RMSE) and R-squared (R^2) as key metrics.

Root Mean Squared Error (RMSE), measures the average magnitude of the prediction errors. It is the square root of the average squared differences between predicted and actual values. A lower RMSE indicates a model with smaller prediction errors. In this context, RMSE provides insight into the typical size of the errors made by the model in predicting housing prices.

R-squared (R^2), on the other hand, measures the proportion of the variance in the target variable (housing prices) that is explained by the model. An R^2 value closer to 1 indicates a better fit, meaning the model explains most of the variance in the data. For example, an R^2 value of 0.93 suggests that the model explains 93% of the variation in housing prices.

A. Correlation Between Housing Features and Prices

The evaluation process began with an analysis of feature correlation to identify key factors influencing housing prices. Figure 1 presents the correlation matrix heatmap visually represents relationships between different features, using colors to indicate the strength and direction of correlations. It shows the four features most strongly associated with sold price: home square footage (0.84), number of washrooms (0.54), total bedrooms (0.33), and location within the Halton region (0.27).

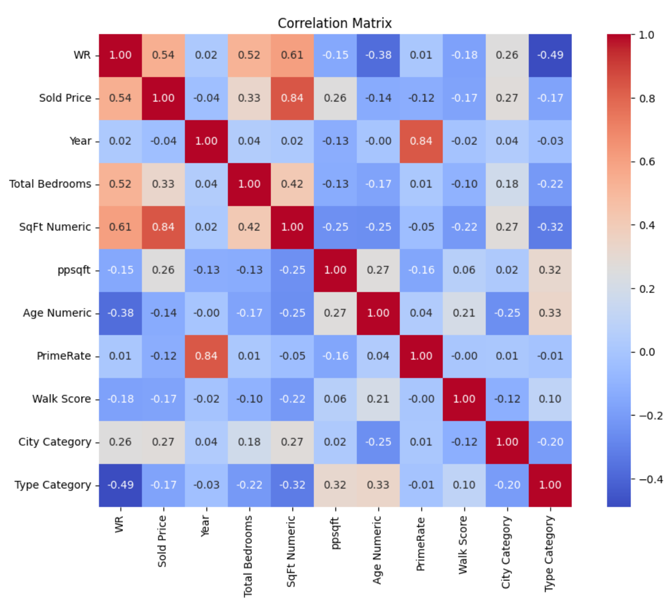


Figure 1. Correlation Matrix for Various Property Features

To understand more the results, a correlation matrix is a table that displays pairwise correlations, showing how strongly two features are related. A heatmap enhances this representation by using colors to illustrate correlation values, making it easier to identify patterns where rows and columns represent the same set of features. For example, "WR" correlates with "WR," "Sold Price" correlates with "Sold Price," and so on. The color bar on the right of figure 1 indicates the strength and direction of the correlation:

- Dark Red (1.0): **Perfect positive** correlation, meaning two features increase together.
- Light Red/Pink (closer to 0.0): **Weak positive** correlation.
- White (0.0): **No correlation**.
- Light Blue/Cyan (closer to 0.0): **Weak negative** correlation.
- Dark Blue (-1.0): **Perfect negative** correlation, meaning one feature decreases as the other increases.

Other findings include a moderately strong positive correlation (0.54) exists between the number of washrooms and sold price, suggesting that homes with more washrooms tend to have higher prices. Home square footage and sold price exhibit a very strong positive correlation (0.84), indicating that larger homes are generally more expensive. Conversely, a weak negative correlation (-0.17) is observed between sold price and Type Category, implying that certain property types may have slightly lower prices. Similarly, the weak negative correlation (-0.25) between home square footage and price per square foot suggests that larger homes tend to have a lower cost per square foot.

This correlation analysis provides valuable insights for identifying influential features and refining predictive models.

B. Model Performance Analysis

Multiple models were built to predict housing prices, including linear regression, ensemble techniques, and tree-based algorithms.

Linear Regression and Lasso Regression models were used to establish a benchmark for model performance, achieving strong R^2 scores of approximately 0.93. While these models are simple and interpretable, they may not effectively capture complex non-linear relationships in the data.

Decision Tree, Random Forest, and XGBoost models were subsequently explored to leverage more sophisticated modeling techniques. The Decision Tree model demonstrated exceptional performance with an R^2 value nearing 0.99, suggesting an impressive ability to capture intricate patterns within the dataset. However, it's important to note that decision trees can be subject to overfitting, especially with deeper trees.

The Support Vector Machine (SVM) exhibited some predictive capabilities, albeit with a lower R-squared value of 0.0877 compared to the higher values of 0.93 and 0.99 achieved by other models such as Decision Trees and Linear Regression.

Random Forest, an ensemble learning method, delivered competitive results with a notable R^2 value of approximately 0.997. This model combines multiple decision trees to mitigate overfitting and enhance predictive accuracy, making it a popular choice for regression tasks.

XGBoost, a gradient boosting algorithm, displayed strong performance with an R² score of around 0.95. XGBoost excels in optimizing predictive performance by iteratively improving upon weak learners, thus providing superior predictive accuracy.

Table III provides an overview of *model performance*, using Root Mean Squared Error (RMSE) and R-squared (R²) metrics. These metrics are shown in Figures 2 and 3, illustrating the distribution of RMSE and R² across different models.

TABLE III
MODEL EVALUATION METRICS

Model Name	Root Mean Squared Error	R-squared
Linear Regression	205345	0.930948
Lasso Regression	205345	0.930948
Decision Tree	71050	0.991733
Support Vector Machine	746374	0.087736
Random Forest	42916	0.996984
XGBoost	202231	0.950595

C. Model for an Online Tool

Based on the results obtained, Random Forest was chosen as the most suitable and accurate model for the next phase of this research, which involves developing an online tool using a Flask web application. The tool enables users to input property features—such as Home Type, Number of Bedrooms, Number of Washrooms, Square Footage, Prime Rate, and Walk Score—to estimate their home price, leveraging the robust predictive capabilities of the Random Forest model.

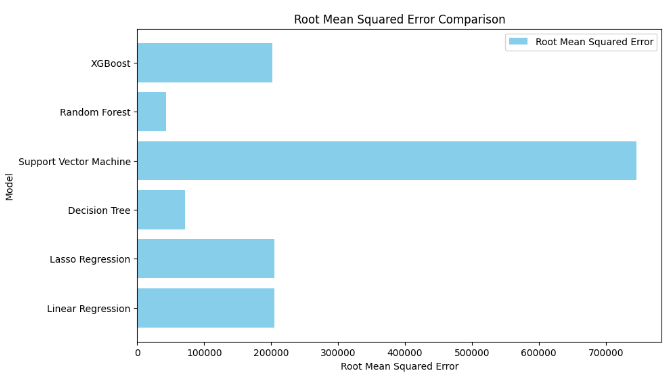


Figure 2. Root Mean Square Error Comparison

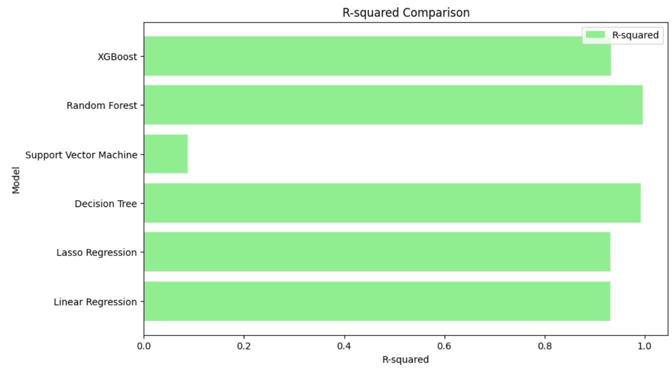


Figure 3. R squared Comparison

V. HOME PRICE ESTIMATION TOOL

The home price estimation tool is a web-based application designed to provide users with real-time predictions of house prices in the GTA, encompassing regions such as Halton, Peel, Toronto, Durham, and Hamilton. This tool is built on a machine learning framework, which dynamically predicts house prices based on user-provided property features, leveraging models tailored for each region to reflect the unique market dynamics and characteristics of that area.

The model development process begins with the collection and preprocessing of historical housing transaction data for each region. Feature engineering is applied to extract relevant attributes, including Home Type, Number of Bedrooms, Number of Washrooms, Square Feet, Prime Rate, and Walk Score, although the exact feature set varies by region. For instance, the Prime Rate is included in areas where interest rate fluctuations heavily influence house prices, while Walk Score may be excluded in regions where it is not a significant factor.

Each region's housing data is used to train a Random Forest model, implemented with the `scikit-learn` library. These models are optimized using metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), ensuring high accuracy in predictions. Once the best-performing models are identified, they are serialized using the `joblib` library and saved as '.pkl' files. This serialization step enables the models to be stored and loaded efficiently within the web application, without the need for retraining.

The web application is built using the Flask framework, providing a user-friendly interface where users can input specific property features to estimate home prices. The process begins with users selecting their region and city, followed by entering property details such as Home Type, Number of Bedrooms, Number of Washrooms, Square Feet, Prime Rate, and Walk Score. When the form is submitted, the application loads the corresponding region-specific model from its serialized '.pkl' file. The user inputs are then processed, and the model predicts the house price in real time.

Once the price is predicted, it is displayed on the web interface along with up to three comparable property listings from the same city. These comparables are selected based on their similarity to the user-provided property details, using Euclidean distance for proximity, considering factors like location, home type, and number of bedrooms to provide additional context for the predicted price. Figure 4 shows an example of the tool's output for a home price prediction in Milton, located in the Halton Region, alongside comparable listings.

By leveraging pre-trained machine learning models, serialized in '.pkl' format, the home price estimation tool ensures fast, accurate predictions while minimizing computational

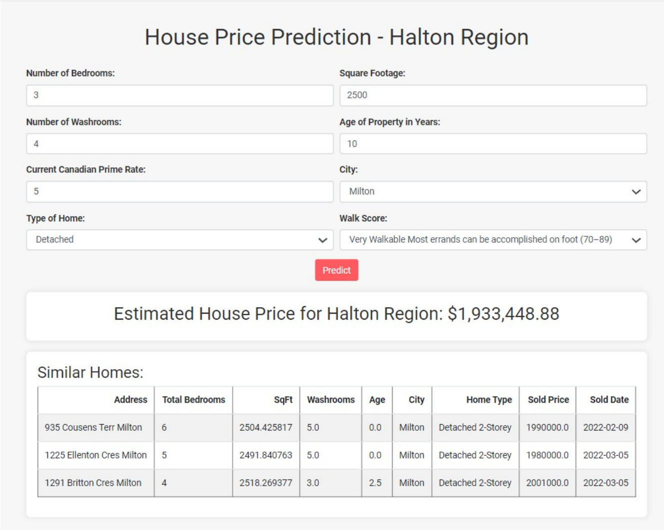


Figure 4. Home Price Tool UI

overhead. This integration of machine learning into a web-based environment offers a practical solution for real estate price estimation, making advanced predictive models accessible to users in real-time. The ML powered Architecture diagram can be shown in figure 5 below. The entire project, including data preprocessing, model training, and the Flask web application, is available on GitHub at https://github.com/Y-Roman/HousePricePredictionModel/tree/master for further exploration and use.

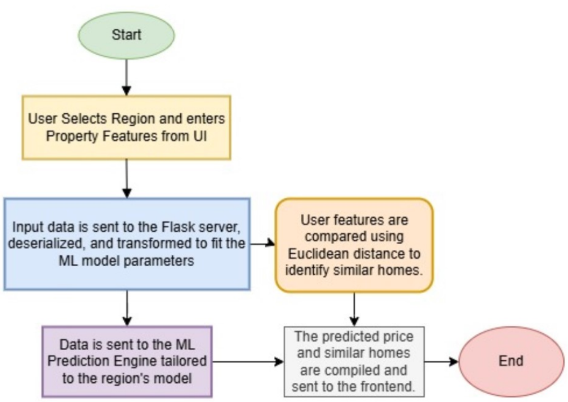


Figure 5. Activity Diagram for the ML Powered Web application

VI. Exploratory Data Analysis

The quantities of homes sold in 2022 and 2023 are illustrated in the bar graphs shown in Fig.6. Notably, the overall trend regarding the number of homes sold remains consistent across both years, with a substantial portion of transactions concentrated in the first quarter, particularly in February, March, and April. However, it is essential to highlight a significant observation regarding the shift in market dynamics between the two years. With the increase in Prime Rates observed from 2022 to 2023, coupled with interest rates surpassing 6.45% in 2023, there is no noticeable decline in the number of transactions. Specifically, there was only about a 3% decrease in the number of transactions recorded in 2023 compared to 2022. This observation suggests that there is no significant correlation between changes in interest rates and real estate market activity, indicating that financial factors, such as interest rates, may not be a key driver influencing buyer behavior and overall market dynamics during this period for this area.

Furthermore, Fig.7 portrays the percentage change of quantity of homes sold in the 2022 months versus the 2023 months, indicating a spike hike in the months of June and October. A 40% increase of homes sold in June and a 30% increase in October is observed, likely due to Prime Rates not increasing significantly in these periods, with only a 0.25% increase in interest rates.

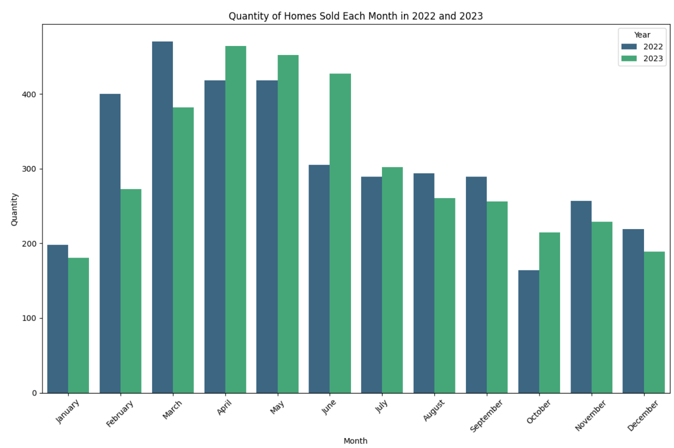


Figure 6. Quantity of Monthly Homes Sold in 2022 and 2023

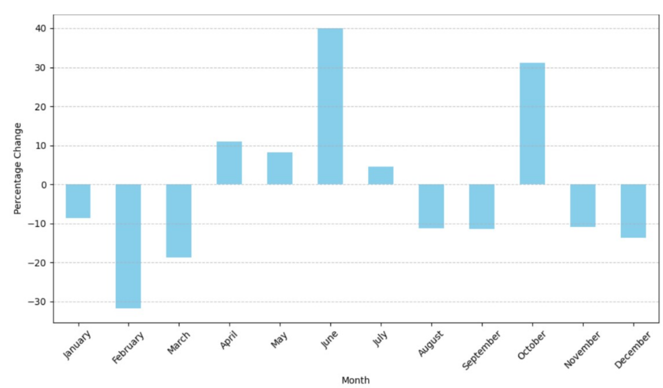


Figure 7. Monthly Percentage Change in Real Estate Transactions between 2022 and 2023

Moreover, the analysis extends to explore the broader trends in the real estate market through Fig. 8. The histogram in Fig.

8 illustrates an overarching downward trajectory in the number of homes sold, displaying a decline of over 50% from January to December of 2022. Concurrently, the accompanying line plot in Figure 9 illustrates a noticeable negative correlation between the escalation of prime rates, starting at 2.5% at the beginning of 2022 and reaching 6.5% by the year's end, and a corresponding 13.75% decrease in average home prices. These observations highlights the complex interaction between macroeconomic factors and the real estate market, emphasizing the subtle relationship between interest rates, market sentiment, and property values.

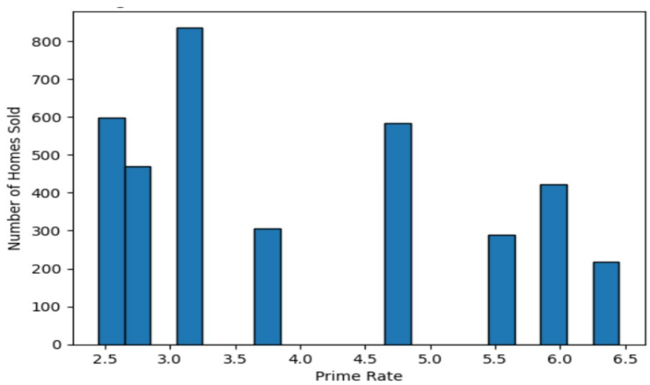


Figure 8. Number of Homes Sold VS. Prime Rate in 2022

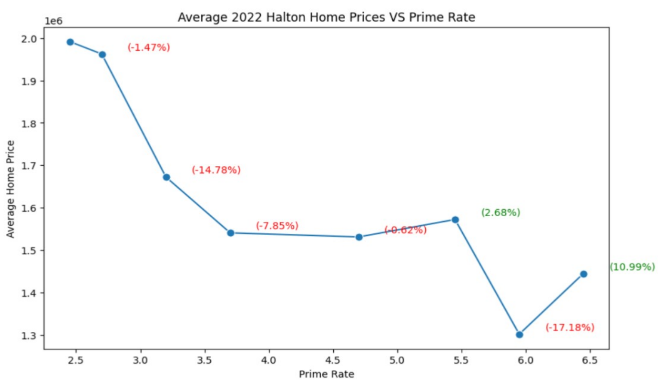


Figure 9. Percentage Trends of Number of Homes Sold VS. Prime Rate in 2022

Figure 10 illustrates the Average Home Sold Price of 2022 and 2023, revealing a modest decrease of 3.94%. In tandem, Figure 11 displays the Average Sold Price Per Square Foot of 2022 and 2023, displaying a reduction of -6.39%. These observations suggest that while the increase in interest rates may have exerted some minor influence on home prices, it appears to have been mitigated by other factors. Notably, Canada's low supply levels and the escalating construction costs attributed to heightened inflation and supply chain disruptions likely played significant roles in stabilizing home prices. As a result, despite the uptick in Prime Rates, the impact on the prices of detached homes in the Halton Region remained relatively subdued.

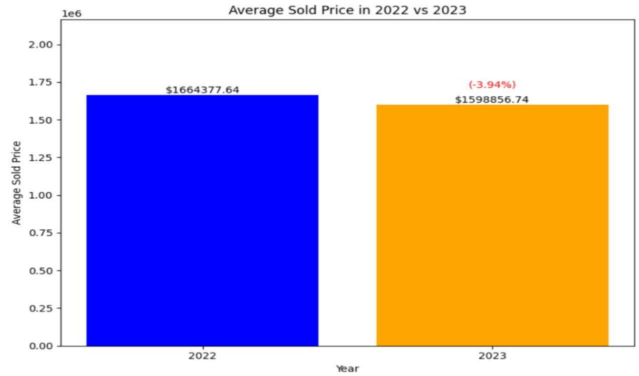


Figure 10. Average Home Price 2022 VS. 2023

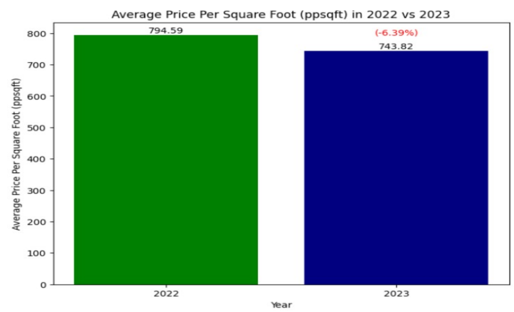


Figure 11. Average Price Per Square Foot 2022 vs. 2023

VII. Conclusion and Future Work

This study highlights the development of a Flask-based web application for home price estimation, which transforms complex machine learning predictions into an accessible, user-friendly tool. This web application allows users to input property details and receive precise price predictions, along with information on three comparable properties. It serves as a valuable resource for homebuyers, sellers, real estate agents, and investors, effectively bridging the gap between sophisticated machine learning models and practical real-world applications.

The integration of property-specific attributes with external economic indicators, such as prime rates, adds depth to the predictive capabilities of the model. This approach addresses existing research gaps by incorporating broader market conditions, resulting in more comprehensive and reliable price forecasts.

Looking ahead, future work could expand the tool's capabilities to include features allowing users to upload and analyze their own datasets, facilitating direct model training and tailored predictions through the web interface. These enhancements would provide users with greater flexibility and further broaden the tool's application to meet diverse needs in the real estate market.

References

[1] X. Li, "Prediction and analysis of housing price based on the generalized linear regression model", *Journal of Housing*

Research, 2022, Received 14 July 2022; Revised 22 August 2022; Accepted 30 August 2022; Published 29 September 2022.

[2] U. Agarwal, S. K. Gupta, and M. Goyal, “House price prediction using linear regression in ml”, 2022.

[3] M. J. Chowhaan, D. Nitish, G. Akash, N. Sreevidya, and S. Shaik, “Machine learning approach for house price prediction”, *Asian Journal of Research in Computer Science*, vol. 16, no. 23, 2023. DOI: 10.9734/AJRCOS/2023/v16i2339.

[4] P. Furia and A. Khandare, “Real estate price prediction using machine learning algorithms”, *Journal of Real Estate Research*, 2022.

[5] R. Konwar, A. Kakati, B. Das, D. B. Shah, and M. K. Muchahari, “House price prediction using machine learning”, *Journal of Real Estate Prediction and Analysis*, vol. 9, 2021.

[6] Authors, “Real-estate price prediction with deep neural network and principal component analysis”, 2022, Received: April 18, 2022; Accepted: November 10, 2022. DOI: 10.2478/otmcj-2022-0016.

[7] A.-R. Mawlood-Yunis and S. Yu, “Applying machine learning and optimization algorithms to perform feature selection”, in *Proceedings of the 7th International Conference on the Dynamics of Information Systems (DIS 2024)*, 2024.

[8] S. Lu, Z. Li, Z. Qin, X. Yang, and R. S. M. Goh, “A hybrid regression technique for house prices prediction”, *Institute of High Performance Computing (IHPC), Agency for Science Technology and Research (A*STAR)*, 2020.

[9] R. MLS, *Realm mls software package*, https://thenewrealm.ca/, Accessed: 2025-02-01.

[10] Redfin Corporation, *Redfin data center*, Accessed: 2024-11-30, 2024.

Profound Increasing UX based on Geometry Shapes and Coloring

Behzad Khosravani Majd
Master Student in UPM
Software Engineering
Serdang, Malaysia
e-mail: behzadmajd@gmail.com

Dr. Azrina Kamaruddin
Senior Lecturer at UPM
Serdang, Malaysia
e-mail: azrina@upm.edu.my

Behnoosh Khosravani Majd
Master of Intrior Archtecture
Khavaran Institute, Mashhad, Iran
e-mail: khosravanimajdbehnoosh@gmail.com

***Abstract*—The interplay between visible objects and human perception is a critical element in design, yet the role of geometric shapes and color in enhancing user experience (UX) remains underexplored. This challenge is significant because effective UX design profoundly impacts usability and aesthetics, influencing user engagement and satisfaction in web development. While prior research highlights the importance of design elements, many studies fail to systematically address the cognitive and emotional impacts of shapes and colors. This paper bridges that gap by leveraging both the Honeycomb UX model and Google's HEART framework to analyze the influence of geometric shapes and color schemes in web design. The proposed framework emphasizes their complementary strengths: the Honeycomb model's qualitative insights and the HEART framework's quantitative metrics. The findings demonstrate that integrating these models enables designers to craft interfaces that balance usability, emotional engagement, and functional efficiency. Readers will gain actionable insights into how geometric shapes and colors, when applied strategically, can transform UX into a visually compelling and highly engaging experience.**

Keywords-text; UX, Design; Geometric Shapes; Color.

I. Introduction

User experience (UX) plays a fundamental role in web design, directly impacting user satisfaction, engagement, and overall platform effectiveness. While much attention has been given to functionality and usability, the potential of geometric shapes and color schemes to enhance cognitive and emotional engagement remains underexplored. Visual elements such as shapes and colors influence how users perceive and interact with interfaces, guiding their attention and evoking emotional responses. For example, rectangles convey stability, while dynamic colors enhance visual interest and improve task focus. Despite these insights, many existing studies fail to systematically integrate the cognitive and emotional impacts of these elements into cohesive frameworks. This paper addresses this gap by proposing a structured approach that combines the qualitative strengths of the UX Honeycomb model with the quantitative metrics of the Google HEART framework. By focusing on how geometric shapes and colors can strategically transform user experiences, this paper offers actionable insights for designers aiming to create visually compelling and emotionally engaging digital platforms. The scope of this study encompasses theoretical and empirical investigations into the role of visual elements in UX design. Specifically, it seeks to:

- Section 1: What role do color schemes play in enhancing usability and engagement?
- Section 2: How do geometric shapes influence the cognitive and emotional aspects of UX?
- Section 4 & 5: How can established UX models be adapted to incorporate shape and color strategies?

The structure of this paper is organized as follows:

- Section 1: explores the theoretical and practical aspects of geometric shapes, categorizing them into organic, abstract, and geometric types, and examining their cognitive and emotional impacts.
- Section 2: delves into coloring techniques, including hue, saturation, and value, highlighting their roles in readability, mood setting, and accessibility.
- Section 3: introduces the Honeycomb and Google HEART models, analyzing their applicability to UX improvements involving shapes and colors.
- Section 4: presents a comparative analysis of these models, identifying their strengths, limitations, and complementary aspects.
- Section 5: concludes with practical recommendations for integrating these design elements into web development workflows and suggests directions for future research.

II. Analysis

The existing body of research reveals a diverse range of perspectives on the role of geometric shapes and color in UX design. Academic literature emphasizes the cognitive and emotional dimensions of these elements, highlighting their impact on user decision-making, satisfaction, and engagement [2][3]. For instance, empirical findings demonstrate that high-contrast color schemes significantly improve task performance and accessibility [17], while shapes such as circles and triangles elicit specific emotional responses that enhance usability [1]. From an industry standpoint, frameworks like the Honeycomb model and Google's HEART metrics have been instrumental in guiding design practices. However, a recurring limitation in industry applications is the tendency to prioritize usability over emotional resonance, resulting in designs that may function well but lack emotional depth. This discrepancy underscores the need for a more integrated approach that incorporates both academic insights and practical tools. By considering these sources, the current study identifies critical opportunities for innovation. For example, while academic research provides a strong theoretical foundation, it often lacks actionable guidelines for implementation. Conversely, industry practices excel in delivering measurable outcomes but may overlook nuanced qualitative aspects of user experience. Bridging these gaps requires a holistic methodology that leverages the strengths of both domains, ensuring that designs are not only efficient but also emotionally engaging and user-centered.

By systematically addressing these objectives, this paper offers a comprehensive roadmap for leveraging geometric shapes and colors in UX design. It underscores the need for collaboration between academic experts and industry practitioners to ensure that theoretical advancements translate into meaningful, real-world applications. Ultimately, this study seeks to enhance our understanding of the interplay between emotional and cognitive design elements while fostering innovation in both research and practice.

III. Coloring & Geometric-Shapes

This section explores the transformative role of color in UX design, highlighting how strategic color choices can enhance user interaction, evoke emotional responses, and reinforce brand identity.

A. Coloring

Colors hold transformative potential in UX design, influencing both the emotional and functional aspects of user interaction. They are not merely aesthetic choices but powerful tools for guiding user attention. For instance, complementary colors like blue and orange create a visual contrast that ensures critical elements, such as calls to action or buttons, stand out. This strategic use of color enhances clarity, making interfaces more engaging and intuitive [3]. Neutral colors, including white, beige, and gray, are equally impactful. These tones serve as versatile backdrops, allowing dynamic elements to take center stage without overwhelming the user. Such designs exude a clean, professional appeal and establish a sense of trustworthiness. By balancing neutral tones with vibrant accents, designers can effectively highlight critical components while maintaining harmony in the layout [18]. Advanced color techniques, such as desaturation or monochromatic schemes, further enrich the user experience. Desaturation softens vibrant palettes, creating a timeless and sophisticated appearance that resonates with audiences. Meanwhile, monochromatic schemes unify design elements, delivering consistency and emotional depth that reinforce brand identity and enhance user trust. These choices transform designs into cohesive visual experiences that captivate and guide users seamlessly [17]. Figure 1 illustrates the impact of different color choices on design outcomes. It is organized into four quadrants divided by two axes. The horizontal axis represents a spectrum from muted tones on the left to vibrant tones on the right, while the vertical axis ranges from neutral colors at the bottom to complementary colors at the top. The top-left quadrant, labeled "Sophisticated Monochromatic Scheme" highlights the use of muted and complementary colors. This approach conveys a sense of refinement and professionalism, making it ideal for designs that require subtle elegance and a cohesive aesthetic. In contrast, the top-right quadrant, titled "Striking Call-to-Action Buttons" focuses on vibrant and complementary colors. These combinations are highly attention-grabbing, making them suitable for elements like buttons or links that need to stand out and encourage user interaction. The bottom-left quadrant, titled "Clean Modern Aesthetics" features muted and neutral tones. This combination creates minimalist and sleek designs, often associated with modern, uncluttered aesthetics that prioritize simplicity and clarity. Similarly, the bottom-right quadrant, labeled "Energetic Branding" emphasizes vibrant and neutral tones. These colors evoke a sense of energy and excitement, making them effective for branding that seeks to convey dynamism and enthusiasm. This framework serves as a guide for designers to strategically select color schemes that align with specific design objectives or brand identities, balancing tonal relationships to achieve desired outcomes.

Figure 1. Coloring diagram

Techniques such as desaturation and monochromatic color schemes add depth and focus to designs. Desaturation reduces vibrancy to establish an understated elegance, making it ideal for evoking nostalgia

 ISBN: 978-1-68558-243-2

or directing focus to specific areas. Monochromatic schemes, on the other hand, use varying shades and tints of a single color, creating consistency and emotional resonance. These techniques not only enhance visual appeal but also strengthen brand identity, ensuring a lasting impression on users.

The following section delves into the foundational role of geometric shapes in UX design, illustrating how their aesthetic and functional attributes contribute to user engagement, navigation, and emotional impact.

B. Geometric-Shapes

Geometric shapes are fundamental in UX design, offering both aesthetic and functional benefits. Rectangles and squares convey a sense of stability and order, making them ideal for structuring interfaces such as buttons, grids, or content containers. These shapes promote feelings of security and reliability, which are critical for building user confidence in any application [6]. Circles, on the other hand, evoke feelings of unity, simplicity, and completion. These qualities make them suitable for highlighting important elements such as avatars or notification badges. Their inherent softness naturally draws users' attention, fostering a sense of inclusivity and approachability in design [11]. Triangles add a dynamic and directional quality to UX design. Often associated with energy and focus, they are effective tools for emphasizing key information or guiding user interactions. By blending sharp geometries with softer, rounded forms, designers can create interfaces that balance precision and warmth. This strategic application of shapes not only enhances usability but also deepens emotional engagement, leaving users with a memorable experience [12]. Triangles serve as effective visual cues, guiding users' focus and signifying movement or urgency. They are often used for tooltips, callouts, or directional indicators (see Fig. 2). The contrast between sharper and softer geometric forms allows designers to balance precision with approachability, tailoring the experience to users' emotional and practical needs. When employed thoughtfully, geometric shapes transform designs into visually striking and highly usable interfaces. Figure 2 presents a hierarchical diagram illustrating the role of geometric shapes in design and how different shapes contribute to various outcomes. The diagram is divided into three main categories: rectangles and squares, circles, and triangles, each leading to specific design applications and impacts. The left branch focuses on rectangles and squares, which are primarily used for grids and layouts. These shapes provide structure and order, enhancing navigation by organizing content in an easy-to-understand and follow manner. The center branch highlights circles, which are associated with interactive elements. Circles are often used for buttons, icons, or other user interaction points, ultimately improving interaction by drawing attention to key features or actions.

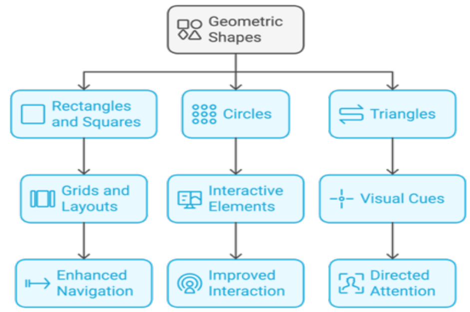


Figure 2. Geometric Shapes diagram

The right branch emphasizes triangles, which serve as visual cues. Triangles direct attention by pointing or creating a sense of movement, thereby guiding users toward important elements in a design. This diagram serves as a guide for designers, demonstrating how the strategic use of geometric shapes can enhance functionality, usability, and visual hierarchy in design projects.

IV. COLORING TECHNIQUES

Coloring techniques, encompassing hue, saturation, and value, play a pivotal role in creating visually compelling and accessible designs. Hue refers to the type of color, such as red, blue, or yellow, and is instrumental in setting the overall mood and theme of a design. Warm hues like red and orange evoke energy and urgency, while cool hues like blue and green convey calmness and reliability. By carefully selecting hues, designers can forge emotional connections and effectively communicate brand identity [7]. Saturation, the intensity or purity of a color, significantly influences readability and visual hierarchy. Highly saturated colors are excellent for drawing attention to important elements, such as call-to-action buttons or notifications. Conversely, desaturated tones are subtle and soothing, making them ideal for backgrounds and secondary content. Balancing saturation levels ensures that designs are neither overwhelming nor dull, contributing to a harmonious user experience [8]. Value, the lightness or darkness of a color, is critical for ensuring accessibility and clarity. High-contrast color schemes achieved by pairing light and dark values enhance readability and make designs more inclusive, especially for users with visual impairments. Thoughtful use of value not only enhances aesthetic appeal but also ensures that designs meet accessibility standards, making them usable and engaging for a wider audience. Together, hue, saturation, and value empower designers to create effective, inclusive, and visually impactful user experiences [9].

V. HONEYCOMB AND GOOGLE HEART MODELS

The Honeycomb and Google HEART models are foundational frameworks for evaluating and improving user experience (UX), especially when incorporating shapes and colors. The Honeycomb model, introduced by Peter Morville, emphasizes seven facets of UX: useful, usable, desirable, findable, accessible, credible, and valuable. Each facet offers insights into how visual elements, like shapes and colors influence user interaction. For instance, using geometric shapes strategically can enhance usability by simplifying navigation, while an effective color palette can boost desirability by evoking positive emotional responses [10]. The Google HEART framework focuses on measuring five key metrics: Happiness, Engagement, Adoption, Retention, and Task Success. This model is particularly relevant for assessing the impact of shapes and colors in UX. For example, well-defined shapes like buttons and icons, combined with a thoughtful color scheme, improve task success by guiding users intuitively through an interface. Moreover, consistent use of appealing colors fosters user happiness, while engaging shapes, such as dynamic animations, increase interaction and retention rates [4]. Together, these models provide a robust methodology for UX designers to evaluate the effectiveness of their visual choices. By leveraging the Honeycomb model's focus on user-centered qualities and the Google HEART framework's data-driven metrics, designers can optimize interfaces to maximize both functionality and emotional impact. The strategic application of shapes and colors, guided by these frameworks, ensures a balanced approach to creating intuitive, attractive, and user-friendly designs [2].

VI. COMPARATIVE ANALYSIS OF HONEYCOMB AND GOOGLE HEART MODELS

The Honeycomb and Google HEART models serve as complementary frameworks for enhancing UX, each offering unique strengths and addressing specific needs. The Honeycomb model's strength lies in its qualitative focus, emphasizing user-centered facets such as credibility, accessibility, and desirability. This approach allows designers to evaluate the emotional and ethical impact of shapes and colors. For instance, using warm color tones to enhance desirability or geometric shapes to improve accessibility aligns seamlessly with the model's principles. However, the Honeycomb model lacks a structured method for measuring these impacts quantitatively, which may limit its applicability in data-driven design environments [10]. Conversely, the Google HEART framework excels in providing measurable outcomes for UX improvements. Its metrics, such as engagement and retention, help designers assess the practical impact of visual choices and color schemes or shape dynamics. For example, tracking how interactive shapes and vibrant colors influence task success and engagement offers actionable insights for iterative design. However, the HEART model's reliance on metrics can sometimes overlook the nuanced, subjective experiences that the Honeycomb model captures such as emotional resonance and perceived credibility [4]. Figure 3 compares two UX frameworks, the Honeycomb Model and the Google HEART Model, providing guidance for selecting the most suitable framework based on design goals. The central visual features (as a designer standing at the base of two staircases), contemplating which path to take, symbolizing the decision-making process in choosing a UX approach. On the left, the Honeycomb Model is highlighted.

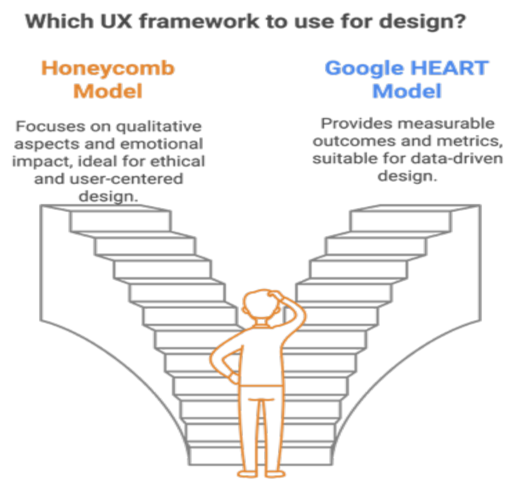


Figure 3. UX Framework options

This framework focuses on qualitative aspects of user experience, emphasizing the emotional impact of design. It is ideal for ethical, user-centered design approaches where the goal is to ensure a meaningful and positive experience for users. On the right, the Google HEART Model is presented. This model provides measurable outcomes and metrics, making it suitable for data-driven design. It prioritizes performance, usability, and efficiency, helping designers track and optimize user interactions based on tangible data. Figure 3 illustrates the contrast between qualitative and quantitative UX methodologies, assisting designers in deciding whether to prioritize emotional and ethical considerations or focus on metrics and measurable performance in their design processes. Table I clearly expresses that both models provide valuable perspectives; however, a hybrid approach that leverages the qualitative depth of the Honeycomb model and the measurable impact of the HEART model offers a well-rounded framework for UX design. Integrating both approaches will enhance designers' efforts to create digital experiences that are emotionally resonant, user-friendly, and data-driven.

VII. PRACTICAL RECOMMENDATIONS AND FUTURE RESEARCH DIRECTIONS

To effectively integrate the principles of shapes and colors, guided by the Honeycomb and Google HEART models, into web development workflows, designers and developers should adopt a systematic approach. First, they should begin with a clear definition of project objectives and user needs, leveraging the Honeycomb model to identify the most relevant facets, such as accessibility and desirability. For example, prioritizing high-contrast color palettes and accessible shapes can promote inclusive design, ensuring that all users can engage effectively.

TABLE I. COMPARATIVE EVALUATION OF UX FRAMEWORKS

Aspect	UX Honeycomb Model	Google HEART Model
Strengths	Provides a qualitative framework focusing on usability, accessibility, and desirability. Addresses cognitive and emotional engagement in UX. Encourages human-centered and ethical design.	Offers a quantitative, data-driven approach. Uses metrics (Happiness, Engagement, Adoption, Retention, Task Success) to track UX success. Helps optimize long-term user behavior and product performance.
Limitations	Lacks a structured quantitative measurement system. Relies on subjective interpretation, leading to potential inconsistencies.	Focuses heavily on user behavior analytics, sometimes overlooking cognitive and emotional factors. May miss psychological and cultural influences on UX.
Supporting Elements	Ensures that UX designs are usable, accessible, and engaging, while HEART measures their real-world impact. Adds an emotional and cognitive layer to HEART's functional and behavioral analysis. HEART's task success and engagement metrics validate the effectiveness of UX improvements guided by the Honeycomb model.	Quantifies the success of emotionally-driven design decisions suggested by the Honeycomb model. Helps UX teams track whether user-friendly and accessible designs enhance actual engagement and retention. Combines qualitative insights with empirical validation to create well-rounded UX strategies.

Concurrently, designers should utilize HEART model metrics, such as task success and engagement, to quantitatively measure the impact of design decisions throughout the iterative development stages [4][6]. Developers should incorporate tools and techniques to streamline this integration. Design systems and style guides can ensure the consistent application of color theory and geometric principles across projects. Wireframes and prototypes can help test the effectiveness of shapes and colors in achieving usability and engagement goals early in the workflow. Additionally, implementing analytics tools can capture HEART metrics, enabling real-time assessment of how shapes and colors influence user interaction and satisfaction. This feedback loop enhances decision-making and supports agile development methodologies [12]. Furthermore, expanding the discussion of cultural and psychological aspects of UX design is crucial for enhancing the theoretical depth of the proposed framework and making it adaptable to diverse user demographics. Recent research underscores that cultural values and psychological preferences significantly influence users' perceptions of design elements, such as color schemes, shapes, and layouts. For example, studies show that users from different cultural backgrounds interpret colors differently; red may symbolize danger in one culture and celebration in another [1]. Similarly, geometric shapes evoke varying emotional responses depending on cultural and psychological contexts, which directly impacts user engagement and satisfaction [13]. Incorporating cultural and psychological insights into the framework ensures that it transcends a universal approach and embraces user diversity. By understanding and integrating these aspects, designers can create interfaces that resonate emotionally and cognitively with users across different demographics. Recent empirical studies, such as those exploring how cultural influences shape UX preferences, provide actionable insights for tailoring design elements to meet specific user expectations [14]. Furthermore, psychological principles, including Gestalt theories and emotional design, offer robust methods for predicting user behavior and enhancing engagement [15]. Adapting the proposed framework to include these elements not only bridges theoretical gaps but also aligns with global design standards, making it more inclusive and effective. Future research should prioritize exploring how cultural dimensions such as individualism versus collectivism or high-context versus low-context communication styles affect UX design strategies. By integrating such research-driven insights, the framework can evolve into a universally adaptable tool that balances cultural sensitivities with psychological principles, ensuring its relevance and utility in diverse contexts. Future research should explore advanced topics, such as the role of cultural and psychological factors in shape and color perception. Investigating how different demographics respond to specific color schemes and geometric layouts can guide the development of more personalized and globally applicable designs. Moreover, integrating AI-driven tools to analyze user behavior and predict the effectiveness of shapes and colors presents a promising direction. These insights can further refine UX frameworks like Honeycomb and HEART, enabling designers to create more effective, data-driven web applications that resonate deeply with diverse audiences [13]. Figure 4 illustrates a conceptual framework for integrating design elements, such as shapes and colors, into the broader scope of web design, guided by the principles of the Honeycomb and Google HEART models.

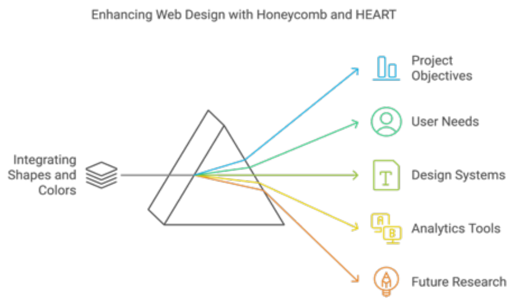


Figure 4. UX Frameworks Enhancement

The central visual is a triangular prism representing the fusion of shapes and colors, which branches out into five key aspects of the design process:

- Project Objectives: Represented by an upward arrow, this branch signifies the importance of aligning design choices with overall project goals, ensuring that all elements contribute to achieving the intended outcomes.
- User Needs: Depicted with a user icon, this aspect focuses on understanding and addressing the requirements and preferences of the target audience, ensuring a user-centered approach.
- Design Systems: Illustrated with a text icon, this branch emphasizes the role of cohesive and structured design systems in creating consistent and scalable user experiences.
- Analytics Tools: Symbolized by a bar graph, this component highlights the use of tools and metrics to analyze user behavior and evaluate the effectiveness of design choices in line with data-driven principles.
- Future Research: Represented by a light bulb icon, this aspect underscores the importance of ongoing exploration and innovation in web design, fostering continual improvement and adaptation to emerging trends.

Figure 4 encapsulates how the integration of visual design elements, coupled with strategic frameworks like Honeycomb and HEART, supports a holistic approach to web design that balances user satisfaction, measurable success, and forward-thinking innovation.

VIII. New Enhancement Approach

The integration of Google's HEART framework and Peter Morville's UX Honeycomb model presents an innovative approach to enhancing user experience by combining their complementary strengths. While the HEART model provides quantitative metrics, such as happiness, engagement, and task success, the UX Honeycomb offers qualitative insights into usability, desirability, and credibility. By merging these two models, this approach creates a comprehensive framework that addresses both the functional and emotional dimensions of user experience, ensuring a holistic evaluation and design process. This integration distinguishes itself from existing models by introducing a dual evaluation system that simultaneously considers emotional resonance and functional effectiveness. Traditional frameworks often focus on either qualitative or quantitative aspects, leaving a gap between emotional engagement and measurable outcomes. By blending the Honeycomb's emphasis on emotional and cognitive factors with HEART's focus on actionable metrics, the combined framework ensures that user experiences are both meaningful and data-driven. For instance, while HEART measures task success, incorporating the Honeycomb's principles of usability and accessibility ensures that these tasks are intuitive and inclusive. The inclusion of visual elements, such as geometric shapes and color theory, further distinguishes this approach. Rather than treating aesthetics as secondary, this framework integrates visual design strategies directly into its evaluation process. By employing high-contrast colors to enhance readability and geometric shapes to direct user attention, it creates a clear link between aesthetics and measurable user outcomes. This strategy not only improves user engagement and satisfaction but also provides actionable insights for designers to create interfaces that are both emotionally resonant and functionally efficient. Ultimately, this transformative combination changes the way user experience is evaluated and designed. By addressing the limitations of existing models and bridging the gap between qualitative and quantitative perspectives, this framework establishes a robust methodology that is adaptable to diverse design contexts. The research connects theoretical models with empirical findings, demonstrating how geometric shapes, organic layouts, and color schemes influence user experience. By incorporating cultural and psychological insights into UX frameworks, designers can create adaptable, inclusive, and effective digital platforms for diverse audiences (Index, report). This approach not only enhances the practical application of UX principles but also sets a new standard for creating interfaces that balance emotional depth with functional efficiency.

IX. Conclusion

This study explored the impact of geometric shapes and color schemes on user experience (UX) in web design, emphasizing their cognitive and emotional effects. By integrating the qualitative insights of the Honeycomb UX model with the quantitative metrics of Google's HEART framework, the research provided a comprehensive approach to evaluating and improving UX design. The findings highlight that the strategic use of geometric shapes enhances usability, guides attention, and fosters engagement, while color schemes influence readability, emotional responses, and brand identity. The comparative analysis of UX models demonstrated the benefits of combining emotional resonance with data-driven metrics to create designs that are both functionally effective and emotionally engaging. Practical recommendations underscored the importance of balancing aesthetics with usability through structured design methodologies, supported by empirical research and industry practices. For future research, further exploration of cultural and psychological influences on shape and color perception is essential to refine UX frameworks for diverse user demographics. Also, integrating AI-driven analytics to assess user behavior in real time can enhance the predictive power of UX models. By continuously evolving UX design strategies through interdisciplinary research and technological advancements, designers can develop more inclusive, adaptive, and engaging digital experiences.

REFERENCE

[1] N. Chen, Advanced Design Psychology: Shape and Form in Digital Interfaces. Cognitive Design Studies, 2023.

[2] A. Marcus, "The Impact of Color and Shape in UX Measurement Frameworks," UX Insights Quarterly, vol. 15, no. 4, pp. 12–19, 2022.

[3] D. Zhang, et al., "Modern Color Theory in Digital Design: A Comprehensive Analysis," Journal of Digital Design, 2023.

[4] K. Rodden, H. Hutchinson, and X. Fu, "Measuring the User Experience on a Large Scale: User-Centered Metrics for Web Applications," in Proc. SIGCHI Conf. Human Factors in Computing Systems, 2010, pp. 2395–2398.

[5] L. Wang, et al., "Modern Applications of the HEART Framework in Digital Products," Human-Computer Interaction Studies, 2023.

[6] T. Anderson, "Evolution of UX Design Models: From Honeycomb to Modern Frameworks," UX Research Review, 2022.

[7] J. Brown, et al., "Color Psychology in Digital Interfaces: A 2023 Perspective," Digital Design Journal, 2023.

[8] M. Thompson, "Neutral Colors in Modern Web Design: Impact and Applications," Web Design Quarterly, 2022.

[9] H. Park, "Contemporary Color Desaturation Techniques in UX," Visual Design Studies, 2024.

[10] R. Wilson, et al., "Universal Design Principles for the Digital Age," Design Science Journal, 2023.

[11] A. Garcia, Modern Elements of User Experience Design. UX Research Today, 2022.

[12] J. Nielsen, "Iterative Design for Usability in Web Development," UX Research Journal, vol. 10, no. 3, pp. 29–35, 2019.

[13] M. Huang and J. G. Lee, "Cultural Influences on Color and Shape Preferences in UX Design," International Journal of Human-Computer Interaction, vol. 35, no. 4, pp. 298–312, 2020.

[14] UXmatters, "Understanding Cultural Norms to Design Inclusive User Experiences," UXmatters, 2024. [Online]. Available: https://www.uxmatters.com/mt/archives/2024/09/understanding-cultural-norms-to-design-inclusive-user-experiences.php.

[15] Maze, "10 Gestalt Principles You Should Know: The Psychology of Design," Maze, 2024. [Online]. Available: https://maze.co/blog/gestalt-principles/.

[16] M. Huang and J. G. Lee, "Cultural Influences on Color and Shape Preferences in UX Design," International Journal of Human-Computer Interaction, vol. 35, no. 4, pp. 298–312, 2020.

[17] S. Kim, et al., "Accessibility Standards in Digital Design: New Perspectives on Value Contrast," Digital Accessibility Quarterly, 2023.

[18] R. Martinez and K. Chen, "Impact of Color Psychology in Modern UX: A Meta-Analysis," International Journal of User Experience, 2022.

INDEX

Report Generated from Empirical Research Findings: https://drive.google.com/drive/folders/1ZrkbQVOd14x2n3CKd3oeKxWXsYSIBJ1Q?usp=sharing

 ISBN: 978-1-68558-243-2