

**DEDAN KIMATHI UNIVERSITY OF TECHNOLOGY**

**SCHOOL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**PROJECT DOCUMENTATION FOR FINAL YEAR IN:**

**BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY**

**BY:**

**KIMANI HEZEKIAH NYUTU**

**REGISTRATION NUMBER: C025-01-0975/2021**

**PROJECT TITLE:**

**SHOPWISEMATE**

**Project proposal submitted to the Department of Information Technology in the school of computer science and Information Technology in partial fulfillment of the requirements for the award of the degree of BSc. in Information Technology at Dedan Kimathi University of Technology.**

**JANUARY, 2024**

# DECLARATION

I declare that this is my original work and has not been presented in any University for a degree or for any consideration of any certification.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Reg No: | Sign | Date |
| Kimani Hezekiah Nyutu | C025-01-0975/2021 |  |  |

This proposal document has been submitted with my approval as the university supervisor

|  |  |  |
| --- | --- | --- |
| Name | Sign | Date |
| Mr. Samwel Muchina |  |  |

# ABSTRACT

The ShopWiseMate project aims to develop a user-friendly web application for comparing prices across multiple ecommerce websites. This proposal provides a brief overview of the problem, objectives, methodology, data processing, key findings, general impact, and major recommendations. The project utilizes web scraping techniques to ensure real-time and accurate pricing information.

# TABLE OF CONTENT

Contents

[DECLARATION 2](#_Toc169074812)

[ABSTRACT 3](#_Toc169074813)

[TABLE OF CONTENT 4](#_Toc169074814)

[LIST OF TABLES 7](#_Toc169074815)

[LIST OF FIGURES 8](#_Toc169074816)

[CHAPTER ONE: INTRODUCTION 9](#_Toc169074817)

[1.1 Background 9](#_Toc169074818)

[1.2 Statement of the Problem 10](#_Toc169074819)

[1.3 Objectives 10](#_Toc169074820)

[1.4 Research Questions 10](#_Toc169074821)

[1.5 Justification 11](#_Toc169074822)

[1.6 Scope 11](#_Toc169074823)

[1.7 Limitation 12](#_Toc169074824)

[CHAPTER TWO: LITERATURE REVIEW 13](#_Toc169074825)

[2.1 Introduction 13](#_Toc169074826)

[2.2 Case Studies 13](#_Toc169074827)

[2.2.1 Case Study 1: Price Comparison Tools 13](#_Toc169074828)

[2.2.2 Case Study 2: Web Scraping Techniques 13](#_Toc169074829)

[2.2.3 Case Study 3: User Interface Design Principles 13](#_Toc169074830)

[2.2.4 Case Study 4: Multi-Site Price Scraping 13](#_Toc169074831)

[2.3 Summary 14](#_Toc169074832)

[2.4 Research Gap 14](#_Toc169074833)

[2.5 Proposed Methodology 14](#_Toc169074834)

[CHAPTER THREE: METHODOLOGY 15](#_Toc169074835)

[3.1 Introduction 15](#_Toc169074836)

[3.2 Planning and Requirements Gathering 15](#_Toc169074837)

[3.3 Software design - Software development procedures. 16](#_Toc169074838)

[3.3.1 Agile Model 16](#_Toc169074839)

[3.4 Data Collection and Preparation 18](#_Toc169074840)

[3.5 Web Scraping 19](#_Toc169074841)

[3.6 Database Design and Management 20](#_Toc169074842)

[3.7 Front-end Development 20](#_Toc169074843)

[3.8 Back-end Development and Selenium Integration 21](#_Toc169074844)

[3.9 Testing and Quality Assurance 22](#_Toc169074845)

[3.10 Deployment and Maintenance 22](#_Toc169074846)

[3.11 User Training and Support 23](#_Toc169074847)

[CHAPTER 4: ANALYSIS AND INTERPRETATION OF RESULTS 24](#_Toc169074848)

[4.1 Introduction 24](#_Toc169074849)

[4.2 Requirement Analysis 24](#_Toc169074850)

[4.2.1 Functional Requirements 24](#_Toc169074851)

[4.2.2 Non-Functional Requirements 25](#_Toc169074852)

[4.2.3 Users of the System 25](#_Toc169074853)

[4.2.4 Inputs of the System 26](#_Toc169074854)

[4.2.5 Outputs of the System 26](#_Toc169074855)

[4.9 Data Collection 26](#_Toc169074856)

[4.9.1 Survey Overview 26](#_Toc169074857)

[4.9.2 Survey Questions 27](#_Toc169074858)

[4.9.3 Survey Responses 27](#_Toc169074859)

[4.9.4 Analysis of Survey Results 27](#_Toc169074860)

[4.4 System Analysis 34](#_Toc169074861)

[REFFERENCES 41](#_Toc169074862)

[APPENDICES 42](#_Toc169074863)

[GANTT CHART 42](#_Toc169074864)

# LIST OF TABLES

[Table 1: Gantt chart 42](#_Toc169074881)

# LIST OF FIGURES

[Figure 1 Agile Model 18](#_Toc169074865)

[Figure 2 age Distribution 28](#_Toc169074866)

[Figure 3 GenderDistribution 28](#_Toc169074867)

[Figure 4 Frequency of online Shopping 29](#_Toc169074868)

[Figure 5 Factors That InfluencePurchase 29](#_Toc169074869)

[Figure 6Search Methods 30](#_Toc169074870)

[Figure 7 Importance of Price 31](#_Toc169074871)

[Figure 8 Experience with Price Comparison Tools 31](#_Toc169074872)

[Figure 9 Challenges Faced 32](#_Toc169074873)

[Figure 10 Misleading Pricing Information 33](#_Toc169074874)

[Figure 11 Desired Improvements 34](#_Toc169074875)

[Figure 12 Context Diagram 36](#_Toc169074876)

[Figure 13 FlowChart 37](#_Toc169074877)

[Figure 14 Level 0DFD 38](#_Toc169074878)

[Figure 15Level1 Dfd 38](#_Toc169074879)

[Figure 16 Sequence Diagram 39](#_Toc169074880)

# CHAPTER ONE: INTRODUCTION

## 1.1 Background

The advent of e-commerce has revolutionized the way people shop, offering convenience and accessibility to a vast array of products. However, with the proliferation of online marketplaces, consumers are faced with the challenge of navigating through numerous platforms to find the best prices for their desired products. This project ShopWiseMate addresses this challenge by introducing an innovative solution for efficient price comparison across multiple e-commerce websites.

Global Perspective

In the global context, the trend of online shopping has seen exponential growth, with a significant increase in the number of consumers preferring the convenience of digital transactions. As a result, the need for tools that simplify the process of comparing product prices across various online platforms has become increasingly evident.

Local Scenario

In the local context, the rise of e-commerce in our region mirrors the global trend. However, the lack of a centralized tool for comprehensive price comparison leaves consumers grappling with the task of manually navigating through different websites to find the best deals. ShopWiseMate emerges as a solution tailored to the needs of our local consumers, providing them with a streamlined approach to make informed purchasing decisions.

Target Group

The target group for the ShopWiseMate project encompasses a diverse range of online shoppers, including individuals seeking the best deals on consumer goods, electronics, and other products. By catering to the needs of this demographic, the project aims to enhance the overall online shopping experience and contribute to the growth of the e-commerce sector in our region.

## 1.2 Statement of the Problem

Despite the convenience offered by e-commerce, the lack of a centralized platform for comparing prices across multiple websites poses a significant challenge to online shoppers. The absence of a dedicated tool results in time-consuming and often frustrating experiences for consumers who aim to find the most cost-effective options for their desired products. To address this problem effectively, the ShopWiseMate project seeks to implement advanced web scraping techniques to aggregate and present real-time pricing information from various e-commerce platforms. By doing so, the project aims to alleviate the challenges faced by online shoppers and empower them with the ability to make informed and economical purchasing decisions.

## 1.3 Objectives

***General Objective***

The general objective of the ShopWiseMate project is to develop an ecommerce price comparison website utilizing web scraping techniques.

Specific Objectives

1. To develop a machine learning model for advanced web scraping techniques to effectively aggregate product prices.
2. To train the machine learning model to ensure real-time updates of product prices and support multi-site price scraping from various ecommerce platforms.
3. To create an interface for efficient product search and comparison.
4. To ensure real-time updates of product prices.
5. To support multi-site price scraping from various ecommerce platforms.

## 1.4 Research Questions

1. How can a machine learning model be developed for advanced web scraping techniques to effectively aggregate product prices from various ecommerce platforms?
2. How can the machine learning model be trained to ensure real-time updates of product prices and support multi-site price scraping?
3. How can an interface be created to facilitate efficient product search and comparison?
4. What methods can be employed to ensure real-time updates of product prices?
5. How can multi-site price scraping be supported from various ecommerce platforms?

## 1.5 Justification

The ShopWiseMate project is justified by the growing need for a dedicated platform that simplifies the process of comparing prices across multiple e-commerce websites. This tool will not only enhance the online shopping experience for consumers but also contribute to fostering a more competitive and transparent e-commerce market.

By providing users with real-time pricing information, "ShopWiseMate" aims to empower consumers with the knowledge needed to make informed decisions, thereby promoting fair competition among online retailers.

## 1.6 Scope

General Scope:

The ShopWiseMate project aims to develop an ecommerce price comparison website utilizing web scraping techniques. It will focus on providing a user-friendly platform for online shoppers to compare prices across multiple ecommerce websites.

Specific Scope:

1. Geographical Focus: The initial focus will be on local and regional online marketplaces.
2. Target Group: The target group includes individuals seeking the best deals on consumer goods, electronics, and other products.
3. Functionality: The project will develop an interface for efficient product search and comparison, ensuring real-time updates of product prices and supporting multi-site price scraping from various ecommerce platforms.
4. Technology: Python will be used for web scraping, HTML/CSS for front-end development, and JavaScript for dynamic functionality.
5. Limitations: Potential limitations may include challenges in web scraping due to site changes or restrictions, and the scope may be limited by the availability of data from certain websites.

## 1.7 Limitation

While this proposal outlines the ambitious goals of the "ShopWiseMate" project, it is important to acknowledge potential limitations. These may include challenges in web scraping due to site changes or restrictions, and the scope may be limited by the availability of data from certain websites. Such limitations will be addressed and documented in the final project report.

# CHAPTER TWO: LITERATURE REVIEW

## 2.1 Introduction

The literature review provides a critical analysis of existing studies relevant to the development of ShopWiseMate. It explores research on price comparison tools, web scraping techniques, user interface design principles, and multi-site price scraping strategies in the context of e-commerce.

## 2.2 Case Studies

### 2.2.1 Case Study 1: Price Comparison Tools

This case study compares and contrasts existing price comparison tools, highlighting their methodologies and effectiveness. It identifies gaps in current tools, such as lack of real-time updates and user-friendly interfaces, which ShopWiseMate aims to address.

### 2.2.2 Case Study 2: Web Scraping Techniques

Here, we analyze different web scraping techniques used for aggregating and updating product prices. By understanding these techniques, ShopWiseMate can implement the most efficient method for price scraping to ensure accuracy and timeliness of information.

### 2.2.3 Case Study 3: User Interface Design Principles

This case study examines user interface design principles for e-commerce applications. It discusses the importance of a user-friendly interface in enhancing the shopping experience and attracting and retaining users.

### 2.2.4 Case Study 4: Multi-Site Price Scraping

In this case study, we explore strategies for scraping prices from multiple e-commerce websites. By analyzing these strategies, ShopWiseMate can develop a comprehensive approach to price scraping, providing users with a wide range of pricing information.

## 2.3 Summary

The literature review highlights weaknesses in existing price comparison tools, such as outdated information and complex interfaces. ShopWiseMate aims to address these weaknesses by providing real-time updates and a user-friendly interface, thereby enhancing the overall shopping experience.

## 2.4 Research Gap

While existing research has focused on price comparison tools, there is a gap in the literature regarding tools that combine advanced web scraping techniques, user-friendly interfaces, and multi-site price scraping. ShopWiseMate aims to fill this gap by providing a comprehensive solution for online shoppers, thereby improving the efficiency and convenience of online shopping.

## 2.5 Proposed Methodology

Based on the literature review, ShopWiseMate will utilize Python for web scraping, HTML/CSS for front-end development, and JavaScript for dynamic functionality. These technologies are well-suited for developing a robust and user-friendly price comparison tool that meets the needs of modern online shoppers.

# CHAPTER THREE: METHODOLOGY

## 3.1 Introduction

The evolution of e-commerce has dramatically transformed the retail landscape, offering consumers unparalleled convenience and access to a vast array of products. With the proliferation of online marketplaces, however, the task of finding the best prices for desired products has become increasingly challenging. Consumers are often faced with the daunting prospect of navigating through numerous websites, comparing prices, and ensuring they are getting the best value for their money. In response to this challenge, ShopWiseMate emerges as an innovative solution, designed to streamline the process of price comparison across multiple e-commerce platforms.

In the global context, the trend of online shopping has witnessed exponential growth, with an ever-increasing number of consumers opting for the convenience of digital transactions. This shift in consumer behavior has underscored the need for tools that simplify the process of comparing product prices across various online platforms. Similarly, in the local context, the rise of e-commerce reflects the global trend, albeit with its own unique challenges. The lack of a centralized tool for comprehensive price comparison leaves local consumers grappling with the task of manually scouring through different websites to find the best deals. ShopWiseMate aims to address this gap, providing a tailored solution that meets the needs of local consumers and enhances their overall online shopping experience.

## 3.2 Planning and Requirements Gathering

The initial phase of developing ShopWiseMate involved meticulous planning and gathering of requirements to ensure the project's success. This phase focused on defining the project's objectives, identifying the target audience, and outlining the features and functionalities required for the price comparison tool.

The planning stage began with a comprehensive analysis of the target audience, including their preferences, habits, and pain points when it comes to online shopping. This analysis helped in defining the key features that would make ShopWiseMate a valuable tool for users. Additionally, the technical requirements for the project were identified, including the choice of programming languages, frameworks, and tools that would be used for development.

Furthermore, the planning phase included defining the scope of the project, setting realistic timelines, and allocating resources accordingly. This ensured that the project remained on track and met its objectives within the specified timeframe. Overall, the planning and requirements gathering phase laid the foundation for the successful development and implementation of ShopWiseMate.

## 3.3 Software design - Software development procedures.

Choosing the appropriate system development methodology is vital for the effective creation of ShopWiseMate Web application. The selection of the methodology hinges on various factors such as the project's scale, team makeup, complexity, and desired pace of development.

### 3.3.1 Agile Model

Adopting agile development methodologies allows for iterative and incremental development, enabling the project to respond quickly to changing requirements. The development process is divided into sprints, with regular demonstrations and reviews to ensure alignment with project goals and user expectations

Steps in Agile Methodology:

1. **Initiate**: Define project goals and requirements.
2. **Design:** In the design phase of agile development, the team collaborates to outline the features and functionality of the product. This involves creating user stories, wireframes, and mockups to define the scope and requirements.
3. **Development:** During development, the team works in short iterations, or sprints, to build and implement features incrementally. Developers write code based on the requirements outlined in user stories, focusing on delivering working functionality at the end of each sprint.
4. **Testing:** Testing occurs throughout the development process, with testers focusing on verifying that the product meets the specified requirements and functions as expected. This includes unit testing, integration testing, and acceptance testing to identify and address any defects or issues.
5. **Deployment:** Deployment involves releasing the product to users or customers after it has been thoroughly tested and approved. In agile development, deployment is often done incrementally, with new features and updates rolled out regularly to provide value to users quickly.
6. **Review:** The review phase involves evaluating the product's performance and gathering feedback from stakeholders, users, and team members. This feedback is used to identify areas for improvement and inform future iterations of the product. Reviews may also involve assessing the team's processes and identifying opportunities for optimization.

These steps are iterative and cyclical, allowing for continuous improvement and adaptation based on feedback and changing requirements throughout the development process.

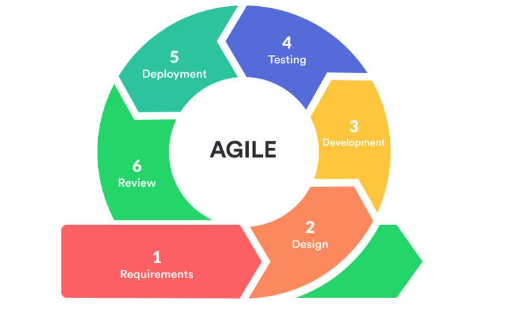


Figure Agile Model

l

## 3.4 Data Collection and Preparation

After defining the project's objectives and requirements, the next step was to collect and prepare the data needed for price comparison on ShopWiseMate. This involved identifying the e-commerce websites to be scraped for product information and devising a strategy to extract, process, and organize this data efficiently.

To begin, a list of target e-commerce websites was compiled, including popular platforms like Kilimall and Jumia. Each website's structure was analyzed to determine the best approach for scraping product information such as prices, descriptions, and images. Python was chosen as the primary programming language for web scraping, given its flexibility and robust libraries for data extraction.

Once the websites were selected and the scraping methodology was established, the data collection process began. Product information was extracted using libraries like BeautifulSoup and Scrapy, which allowed for the extraction of structured data from the websites' HTML code. Special attention was paid to ensure the accuracy and completeness of the extracted data, as any errors or inconsistencies could impact the reliability of the price comparison tool.

After collecting the data, it was processed and organized into a structured format suitable for analysis and comparison. This involved cleaning the data to remove any duplicates or irrelevant information and structuring it into a database for easy access and retrieval. Overall, the data collection and preparation phase was crucial in laying the groundwork for the development of ShopWiseMate's price comparison functionality.

## 3.5 Web Scraping

Web scraping was a pivotal component of the ShopWiseMate project, enabling the extraction of pricing data from various e-commerce websites. Python, with its rich ecosystem of libraries, was chosen for its suitability for web scraping tasks.

The process began by identifying the specific data needed for price comparison, such as product prices, descriptions, and images. Using libraries like BeautifulSoup and Scrapy, the team developed scripts to extract this data from the target websites. Careful attention was paid to ensure the scraping process was efficient, accurate, and respectful of the websites' terms of service.

Scalability and performance were key considerations, especially given the large volume of data involved. The team implemented strategies to manage and distribute the scraping workload effectively, ensuring that data was collected in a timely manner and without overloading the target websites' servers.

Throughout the web scraping phase, regular monitoring and testing were conducted to verify the accuracy and consistency of the extracted data. Any issues or discrepancies were addressed promptly to maintain the integrity of the price comparison tool. Overall, web scraping played a critical role in collecting the data needed to power ShopWiseMate's price comparison functionality.

## 3.6 Database Design and Management

In developing ShopWiseMate, a hybrid approach to database design was adopted, leveraging both MySQL and MongoDB to meet the diverse requirements of the project.

MySQL was utilized for storing structured data, such as product information (prices, descriptions, and images), user data, and website metadata. Its relational model and ACID compliance ensured data integrity and consistency, crucial for transactional operations and complex queries.

On the other hand, MongoDB was used to store unstructured or semi-structured data, such as web scraping results and temporary data caches. MongoDB's flexible document-oriented structure allowed for the storage of data in a format that closely matched the extracted data's original structure, making it ideal for storing and querying web-scraped data.

The decision to use both MySQL and MongoDB was driven by the need to balance the requirements for structured and unstructured data storage efficiently. By leveraging the strengths of each database system, ShopWiseMate was able to effectively manage the diverse data requirements of the price comparison tool, ensuring optimal performance and scalability.

## 3.7 Front-end Development

Front-end development for ShopWiseMate focused on creating a user-friendly interface that enables users to easily search for products, compare prices, and make informed purchasing decisions. The development process involved using HTML, CSS, and JavaScript to design and implement the front-end components of the application.

HTML was used to structure the content of the web pages, defining the layout and organization of the various elements on the page. CSS was then used to style the HTML elements, ensuring a visually appealing and consistent design across the application. JavaScript was used to add interactivity to the interface, such as dynamic updates of product information and seamless navigation between different sections of the website.

The front-end design of ShopWiseMate prioritized simplicity and ease of use, with a clean and intuitive layout that allows users to quickly find the information they need. The interface was designed to be responsive, adapting to different screen sizes and devices to provide a consistent user experience across desktop and mobile platforms. Overall, the front-end development of ShopWiseMate focused on creating a visually appealing, user-friendly interface that enhances the overall shopping experience for users.

## 3.8 Back-end Development and Selenium Integration

The back-end development of ShopWiseMate was crucial for handling user requests, processing data, and ensuring the overall functionality of the application. Python and the Flask framework were used to develop the back-end, providing a robust and scalable foundation for the application. Python was chosen for its versatility and ease of use, making it ideal for handling the complex logic required for web scraping and data processing. Flask, a lightweight and modular framework, was used to create the web application and manage the communication between the front-end and back-end components.

One of the key functionalities of the back-end was to retrieve pricing information from the database and deliver it to the front-end for display. This involved writing scripts to query the database, extract the relevant data, and format it for presentation on the website. Additionally, the back-end was responsible for managing user sessions, authentication, and other security features to ensure the integrity and confidentiality of user data.

Selenium was utilized in the back-end development phase to automate the web scraping process. Selenium is a powerful tool for browser automation and was used to interact with the e-commerce websites, simulate user actions, and extract pricing data. By integrating Selenium into the back-end, ShopWiseMate could ensure the timely and accurate scraping of product prices from various websites.

The use of Selenium enhanced the efficiency of the web scraping process, allowing ShopWiseMate to retrieve pricing information in a reliable and automated manner. This contributed to the overall effectiveness of the price comparison tool, providing users with up-to-date and comprehensive pricing data for informed decision-making.

## 3.9 Testing and Quality Assurance

Testing and quality assurance (QA) were integral parts of the ShopWiseMate development process to ensure the reliability and performance of the price comparison tool. Several types of testing were conducted throughout the development lifecycle to identify and address any issues or bugs.

Unit testing was performed to test individual components or functions of the application to ensure they worked correctly. Integration testing was conducted to test how different parts of the system worked together. This was important to ensure that the front-end and back-end components interacted seamlessly and that data was passed correctly between them.

User acceptance testing (UAT) was also conducted to ensure that the application met the needs and expectations of its users. This involved real users testing the application and providing feedback on its usability and functionality. Additionally, performance testing was carried out to assess how the application performed under different conditions, such as high traffic or heavy load. This helped identify any bottlenecks or areas for optimization.

Throughout the testing process, any issues or bugs that were identified were logged and addressed by the development team. Once testing was complete and all issues were resolved, the application was deemed ready for deployment. Overall, testing and QA were essential to ensure that ShopWiseMate was a reliable, high-performance tool that provided users with accurate and up-to-date pricing information for their online shopping needs.

## 3.10 Deployment and Maintenance

Deployment and maintenance of ShopWiseMate were crucial steps in ensuring the availability and reliability of the price comparison tool. Once development and testing were completed, the application was deployed to a live environment where it could be accessed by users.

Deployment involved transferring the application from the development environment to a production environment. This process required careful planning to minimize downtime and ensure a smooth transition. Configuration settings, database connections, and other dependencies were configured to ensure the application functioned correctly in the production environment. Regular maintenance was essential to keep ShopWiseMate running smoothly and efficiently. This involved monitoring the application for performance issues, security vulnerabilities, and other potential issues. Updates and patches were applied as needed to address any issues and improve the functionality of the application.

User feedback was also monitored to identify areas for improvement and new features that could be added to enhance the user experience. Regular updates and improvements were made to ShopWiseMate to ensure it remained a valuable tool for users.Overall, deployment and maintenance were ongoing processes that ensured ShopWiseMate continued to meet the needs of its users and remained a reliable and efficient price comparison tool.

## 3.11 User Training and Support

User training and support were integral components of the ShopWiseMate project to ensure that users could effectively utilize the price comparison tool and maximize its benefits.

Training materials were developed to provide users with guidance on how to navigate the website, perform product searches, compare prices, and make informed purchasing decisions. These materials included user manuals, tutorials, and FAQs to address common user queries and issues. A support system was established to provide users with assistance and troubleshooting services. This system included a helpdesk or customer support team that could be contacted via email, phone, or chat for personalized assistance. The support team was trained to address a wide range of user queries and issues promptly and effectively.

Additionally, feedback mechanisms were implemented to gather user feedback on the tool's usability, functionality, and overall experience. This feedback was used to identify areas for improvement and inform future updates and enhancements to ShopWiseMate. By providing comprehensive user training and support, ShopWiseMate aimed to ensure that users could easily access and utilize the price comparison tool, ultimately enhancing their online shopping experience.

# CHAPTER 4: ANALYSIS AND INTERPRETATION OF RESULTS

## 4.1 Introduction

This chapter presents the results and analysis of the ShopWiseMate project, which aimed to develop an ecommerce price comparison website. The project focused on aggregating and comparing product prices from various e-commerce websites to help consumers make informed purchasing decisions. This chapter begins by providing an overview of the project objectives and methodology to contextualize the results presented.

ShopWiseMate was designed to enhance the online shopping experience by simplifying the process of comparing product prices across different e-commerce platforms. The tool aimed to address the challenge faced by consumers in navigating numerous websites to find the best deals. The chapter discusses how the implementation of ShopWiseMate progressed, including the development of the web scraping techniques, database design, front-end and back-end development, testing, and deployment. The methodology encompassed several key stages, each crucial in ensuring the successful implementation and functionality of the price comparison tool.

## 4.2 Requirement Analysis

### 4.2.1 Functional Requirements

Functional requirements are essential for ensuring that the system meets its intended purpose. For ShopWiseMate, the functional requirements were identified based on the needs of the users and the objectives of the project.

1. **Price Comparison**: The system must compare product prices from multiple e-commerce websites and display the results to the user.
2. **Search Functionality**: Users should be able to search for products by name, category, or brand.
3. **Real-Time Data Updates**: The system should provide real-time updates on product prices to ensure accuracy.
4. **User Accounts**: Users should be able to create accounts, log in, and save their preferences and search history.
5. **Notifications**: The system should notify users about price drops or special deals on products they are interested in.
6. **User-Friendly Interface**: The interface must be intuitive and easy to navigate, ensuring a positive user experience.
7. **Multi-Site Scraping**: The system should scrape prices from multiple e-commerce websites efficiently and accurately.

### 4.2.2 Non-Functional Requirements

Non-functional requirements define the system's operational attributes. These requirements ensure that the system performs well and provides a good user experience.

1. **Performance**: The system should load search results within a reasonable time frame (e.g., under 3 seconds).
2. **Scalability**: The system should handle a large number of users and data without significant performance degradation.
3. **Security**: User data must be securely stored and protected against unauthorized access.
4. **Reliability**: The system should be reliable with minimal downtime.
5. **Maintainability**: The system should be easy to maintain and update, with clear documentation for future developers.
6. **Compatibility**: The system should be compatible with various devices and browsers, providing a consistent experience across platforms.

### 4.2.3 Users of the System

1. **Admin**: Responsible for managing the system, monitoring performance, and handling administrative tasks.
2. **Consumers**: Individuals using the tool to compare prices and make informed purchasing decisions.
3. **Retailers**: E-commerce websites providing product data for comparison.

### 4.2.4 Inputs of the System

1. **Product Information**: Prices, descriptions, images, and other relevant details from e-commerce websites.
2. **User Data**: User profiles, preferences, search history, and saved items.
3. **Website Metadata**: Information about the e-commerce platforms being scraped.
4. **Search Queries**: Keywords and filters used by users to find products.
5. **Feedback and Reviews**: User feedback on product accuracy and overall experience.
6. **Financial Data**: Subscription payments, advertising fees, and other revenue-related information.

### 4.2.5 Outputs of the System

1. **Price Comparisons**: Lists of products with their prices from different e-commerce websites.
2. **Product Details**: Detailed information about products, including descriptions, images, and reviews.
3. **Notifications**: Alerts about price drops, special deals, and other relevant updates.
4. **User Dashboard**: Personalized information for users, including saved searches and purchase history.
5. **Analytics Reports**: Insights for retailers about product performance and user engagement.
6. **Error Logs**: Records of any issues encountered during web scraping or data processing.

## 4.9 Data Collection

### 4.9.1 Survey Overview

To gather user insights and feedback on the ShopWiseMate price comparison tool, a survey was conducted among a diverse group of online shoppers. The survey aimed to understand user behaviors, preferences, and challenges related to online shopping and price comparison tools. The insights gained from the survey were essential for refining ShopWiseMate and ensuring it met the needs and expectations of its users.

### 4.9.2 Survey Questions

The survey consisted of multiple-choice and open-ended questions designed to capture comprehensive user information and feedback. These questions covered various aspects of online shopping behavior, factors influencing purchase decisions, experiences with price comparison tools, and suggestions for improving online shopping experiences.

### 4.9.3 Survey Responses

The survey received 14 responses, providing valuable data on the target user base for ShopWiseMate. Key demographic information included age, gender, and online shopping frequency. Responses also detailed factors influencing purchase decisions, typical methods of searching for products online, the importance of price compared to other factors, usage of price comparison tools, and challenges faced while shopping online.

### 4.9.4 Analysis of Survey Results

#### Demographics

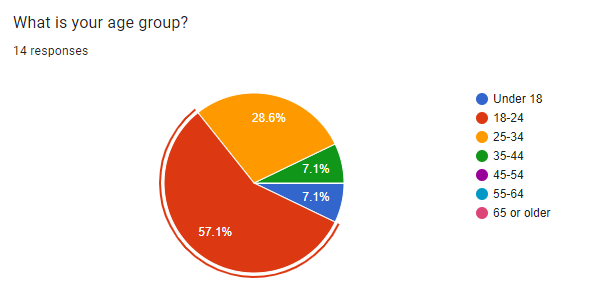
* **Age Distribution**: The majority of respondents were aged 18-24 (57.1%), followed by those aged 25-34 (28.6%). This indicates that younger adults are more likely to participate in online shopping and use price comparison tools.
* ****

Figure age Distribution

* **Gender Distribution**: Most respondents were male (71.4%), with females making up the remaining 28.6%. This suggests a higher engagement from male shoppers in this survey.

#### 

Figure GenderDistribution

#### Online Shopping Behavior

* **Frequency of Online Shopping**: The survey revealed that respondents frequently shop online, with significant numbers shopping 2-3 times a month (28.6%) or once a month (28.6%). This high frequency indicates a substantial market for price comparison tools like ShopWiseMate.

#### 

Figure Frequency of online Shopping

#### Factors Influencing Purchases

* **Primary Factors**: Price (21.4%) and product reviews (35.7%) were the most influential factors in making online purchases. This highlights the importance of providing accurate and comprehensive pricing and review information on ShopWiseMate.

#### 

Figure Factors That InfluencePurchase

#### Search Methods

* **Popular Methods**: Respondents commonly used search engines (42.9%) and e-commerce websites (21.4%) to find products online. This indicates that integrating ShopWiseMate with popular search engines and e-commerce platforms could enhance its usability and reach.

#### 

Figure Search Methods

#### Importance of Price

* **Price Importance**: A significant portion of respondents (35.7%) considered price to be very important when searching for products, while others (35.7%) remained neutral. This underscores the critical role of effective price comparison in influencing purchase decisions.

#### 

Figure Importance of Price

#### Experience with Price Comparison Tools

* **Usage and Experience**: The majority of respondents (64.3%) had used price comparison tools, with most describing their experience as positive (57.1%) or very positive (14.3%). This suggests a general familiarity and satisfaction with price comparison tools among the target audience.

#### 

Figure Experience with Price Comparison Tools

#### Challenges Faced

* **Common Challenges**: Key challenges included difficulty in finding reliable price comparison websites (35.7%) and limited availability of products on comparison websites (50%). Addressing these issues could significantly improve user satisfaction with ShopWiseMate.

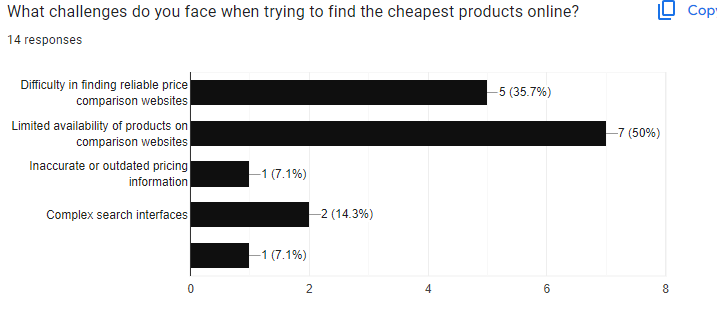


Figure Challenges Faced

#### Misleading Pricing Information

* **Encountered Issues**: A substantial number of respondents (85.7%) had encountered misleading pricing information while shopping online, indicating a strong need for reliable and transparent price comparison tools.

#### 

Figure Misleading Pricing Information

#### Desired Improvements

* **Suggested Enhancements**: Respondents expressed a desire for more reliable and up-to-date price comparison websites (57.1%), improved search filters and sorting options (21.4%), and clearer pricing information (28.6%). These insights provide valuable guidance for future developments and enhancements to ShopWiseMate.



Figure Desired Improvements

### 4.4 System Analysis

In this section, we conduct a comprehensive analysis of the ShopWiseMate system, focusing on its architecture, data flow, user interactions, and performance metrics. This analysis is essential to understand how the system achieves its goal of providing accurate and up-to-date price comparisons for online shoppers.

#### 4.4.1 System Architecture

The architecture of ShopWiseMate is designed to ensure efficiency, scalability, and reliability. It consists of three main layers:

1. **Presentation Layer**: This layer includes the user interface (UI) through which users interact with the system. Built using HTML, CSS, and JavaScript, the UI is designed to be intuitive and responsive, providing a seamless user experience. The presentation layer handles user inputs and displays processed data from the application layer.
2. **Application Layer**: The core functionality of ShopWiseMate resides in this layer. Developed using Python and Flask, this layer includes modules for web scraping, data processing, and application logic. It manages requests, processes data, and orchestrates the interactions between the presentation and data layers.
3. **Data Layer**: This layer is responsible for data storage and management. It uses a MySQL database to store product information, prices, user data, and other relevant details. The data layer ensures efficient data retrieval and storage, supporting the overall performance of the system.

#### 4.4.2 Data Flow

The data flow in ShopWiseMate is structured to ensure timely and accurate data processing and presentation:

1. **Data Collection**: Web scraping scripts collect product prices and details from various e-commerce websites at regular intervals. The data is then cleaned and formatted before being stored in the MySQL database.
2. **Data Processing**: The application layer processes the stored data, applying business logic to aggregate and compare prices. This includes filtering products based on user preferences, calculating price differences, and identifying the best deals.
3. **Data Presentation**: The processed data is sent to the presentation layer, where it is displayed to users. Users can search for products, compare prices, and view detailed product information through a user-friendly interface.

To illustrate the data flow within ShopWiseMate, the following diagrams are provided:

##### Content Diagram

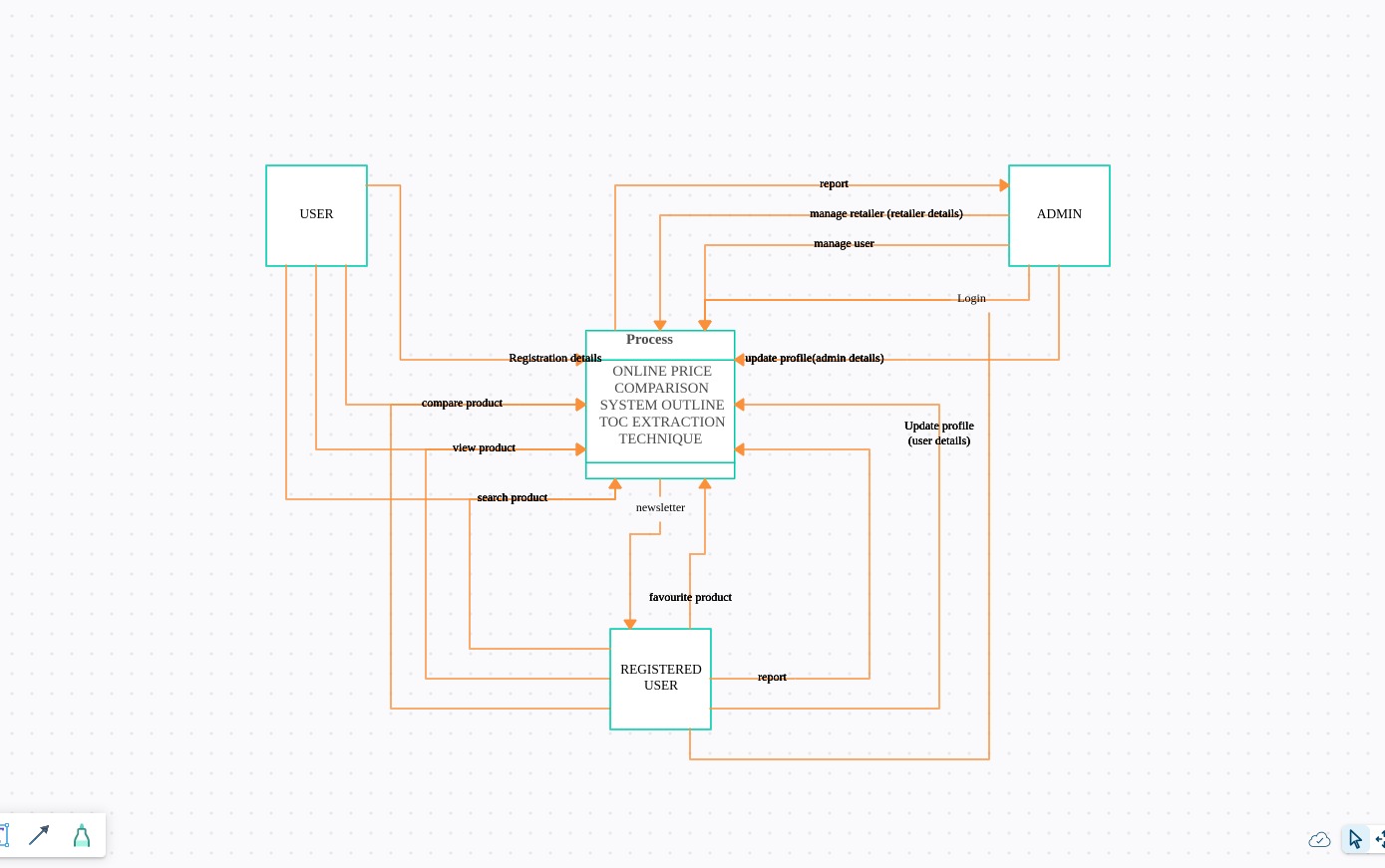


Figure Context Diagram

.

##### Flowchart

##### 

Figure FlowChart

##### Data Flow Diagrams (DFDs)

Level 0 and Level 1 Data Flow Diagrams (DFDs) provide a more detailed view of the data processing within the system.

* **Level 0 DFD**:

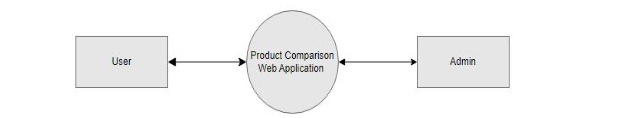


Figure Level 0DFD

* **Level 1 DFD**

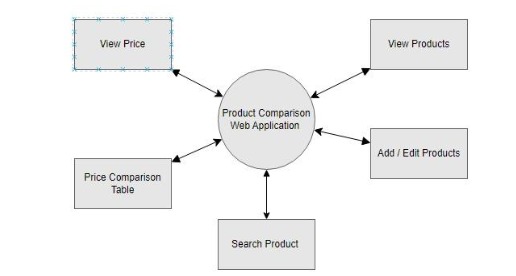


Figure Level1 Dfd

##### Sequence Diagram

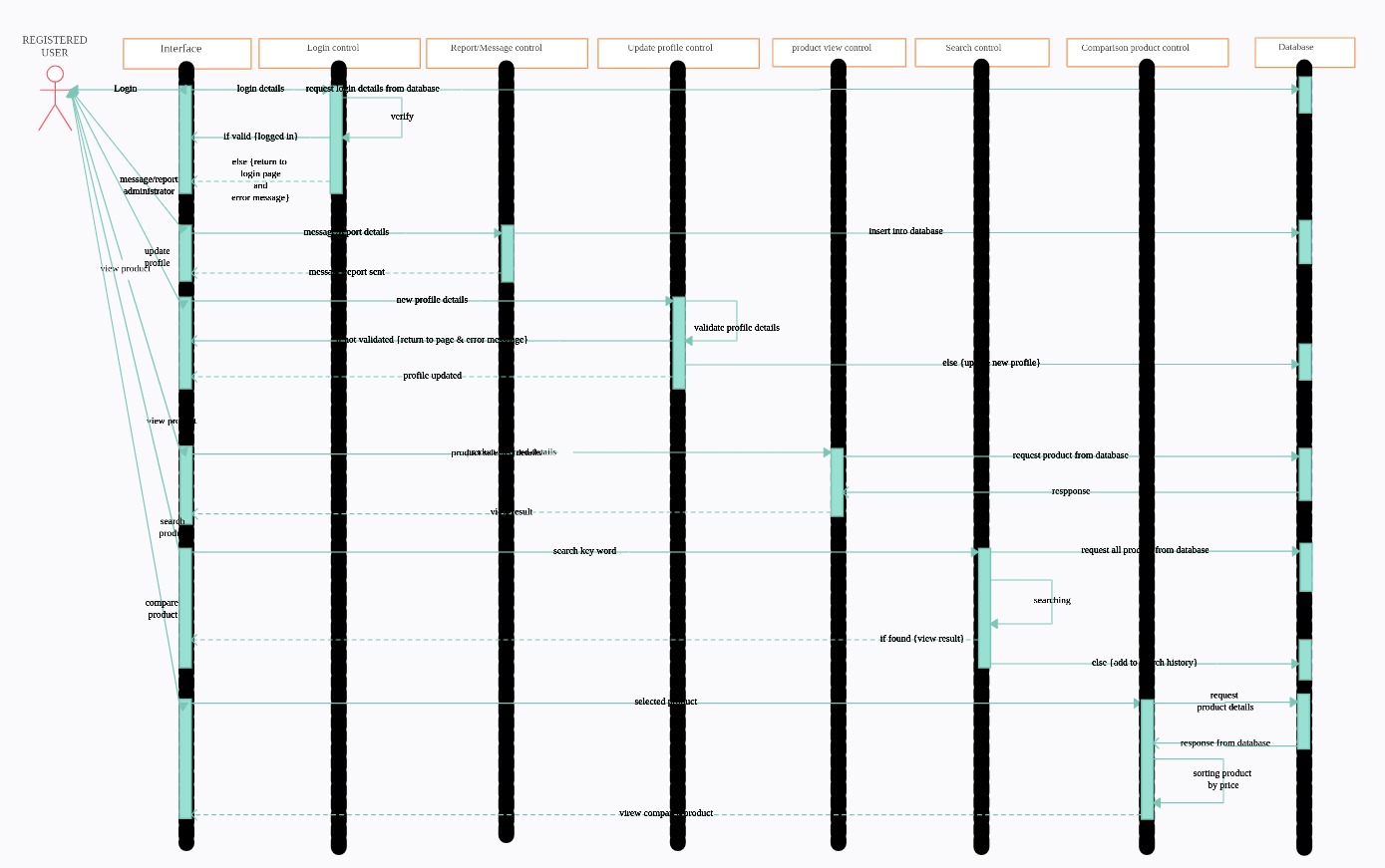


Figure Sequence Diagram

#### 4.4.3 User Interactions

ShopWiseMate supports various user interactions, primarily involving two user roles: Admin and Consumers.

1. **Admin**: Admin users manage the system configurations, user accounts, and performance monitoring. They ensure the system runs smoothly and address any issues that arise.
2. **Consumers**: Consumers use the application to search for products, compare prices, and make informed purchasing decisions. Their interactions involve entering search queries, selecting filters, and viewing detailed product information.

#### 4.4.4 Performance Metrics

To evaluate the efficiency and reliability of ShopWiseMate, several performance metrics are monitored:

1. **Response Time**: The time taken by the system to process user requests and display results. ShopWiseMate aims to keep response times minimal to enhance user experience.
2. **Data Accuracy**: The accuracy of the price and product information displayed to users. Regular updates and validation checks are in place to ensure data accuracy.
3. **System Uptime**: The availability of the application. ShopWiseMate is designed for high availability, with minimal downtime to ensure consistent access for users.
4. **User Satisfaction**: Feedback from users on their experience with the application. Surveys and feedback mechanisms help in assessing user satisfaction and identifying areas for improvement.

#### 4.4.5 Challenges and Mitigations

During the development and deployment of ShopWiseMate, several challenges were encountered. The key challenges and their mitigations are as follows:

1. **Data Inconsistency**: Ensuring consistent and accurate data from multiple sources. This was mitigated by implementing robust data validation and error-checking mechanisms.
2. **Scalability**: Handling increasing amounts of data and user requests. The system architecture was designed to be scalable, with efficient data processing and storage solutions.
3. **User Engagement**: Providing a user-friendly interface and valuable features. Continuous user feedback and iterative improvements were employed to enhance user engagement.

# REFFERENCES

1. *Price Comparison Website for Online Shopping Project*. (2019, January 28). Nevon Projects. <https://nevonprojects.com/price-comparison-website-for-online-shopping-project/>
2. Sharma, S. (2023, May 19). *smartyrad/Price-comparison-website-using-web-scraping-in-python*. GitHub. <https://github.com/smartyrad/Price-comparison-website-using-web-scraping-in-python>
3. *Document a Flight Comparison*. (n.d.). Fingate.stanford.edu. Retrieved February 28, 2024, from https://fingate.stanford.edu/business-travel-expenses/how-to/document-flight-comparison
4. *Web Scraping based Product Comparison Model for E-Commerce Websites | IEEE Conference Publication | IEEE Xplore*. (n.d.). Ieeexplore.ieee.org. Retrieved February 28, 2024, from <https://ieeexplore.ieee.org/document/9915892/>
5. *Intelligent Online Shopping using ML-based Product Comparison Engine | IEEE Conference Publication | IEEE Xplore*. (n.d.). Ieeexplore.ieee.org. Retrieved February 28, 2024, from https://ieeexplore.ieee.org/abstract/document/10134401/
6. Thomas, D. M., & Mathur, S. (2019). Data Analysis by Web Scraping using Python. *2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. <https://doi.org/10.1109/iceca.2019.8822022>
7. Karuna, M., Madhurima, P., Prasanna, J., Pittam, S., Reddy, B., & Student. (n.d.). *DATA ANALYSIS BY WEB SCRAPING USING PYTHON*. Retrieved February 28, 2024, from http://www.journal-iiie-india.com/1\_apr\_23/19.4.pdf

# APPENDICES

## GANTT CHART

Table 1: Gantt chart

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project Tasks/Weeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Idea & Title Proposal |  |  |  |  |  |  |  |
| Introduction |  |  |  |  |  |  |  |
| Aims & Objectives |  |  |  |  |  |  |  |
| Methodology |  |  |  |  |  |  |  |
| Proposal Defense (Presentation) |  |  |  |  |  |  |  |