PriceSnap

Final Year Project

Session 2020-2024

A project submitted in partial fulfillment of the degree of

BS in Computer Science

Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2024

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R**&**D**
Area of specialization	Web Development & AI Integration
FYP ID	FYP-BCSM-F23-084

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-F20-303	Wahaj Khan	Bcsm-f20-303@superior.edu.pk	
(ii)	BCSM-F20-262	Sammar Aleem	Bcsm-f20-262@superior.edu.pk	
(iii)				

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I Wahaj Khan S/D of Muhammad Asif Khan, group leader of FYP under registration no BCSM-F20-303 at Computer Science Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: Name of Group Leader: Wahaj Khan Signature: _____

Name of Supervisor: Mr. M Umer Farooq Co-Supervisor:

Designation: Lecturer Designation:

Signature: _____ Signature: _____

HoD: Dr. Arfan Jaffar

Signature: _____

Project Report

PriceSnap

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Wahaj Khan	1.0		Included Login and sign up	
Wahaj Khan	2.0		Added Suggestions based on user search.	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

Dedicated to Superior University, a beacon of innovation and knowledge, this project, PriceSnap, stands as a testament to the invaluable learning and inspiration fostered within your esteemed institution. With gratitude for the guidance, encouragement, and fertile learning environment provided, this project reflects the principles instilled by Superior University. May this endeavor contribute to the ongoing legacy of excellence that Superior University continues to cultivate in its students and projects, shaping the future of technology and innovation. Thank you for being the foundation upon which this journey of discovery and accomplishment is built.

Acknowledgements

I extend my heartfelt gratitude to my dedicated supervisor, whose unwavering support and guidance have been instrumental in the development and realization of PriceSnap. Their expertise, encouragement, and insightful feedback have enriched every stage of this project, turning challenges into opportunities for growth. I am profoundly thankful for their mentorship, patience, and belief in the potential of this idea. This achievement is a testament to their commitment to fostering innovation and excellence. It has been an honor and privilege to work under their guidance, and I am profoundly grateful for the invaluable impact they have had on this journey.

Executive Summary

This project proposes the development of a website that would scrape the web for a product that would be specified by the user and show the user the prices on the product that is available on different websites.

The website would be developed using a variety of web development technologies, including React JS, Node JS, Python, Express JS, Mongo DB. The Python programming language would be used to write the web scraper scripts that would be used to extract the product information from the web. React, Node, Express and Mongo DB would be used to develop the website itself.

The website would have a simple user interface that would allow users to enter the name of the product they want to search for and then display the prices on the product that is available on different websites.

The website would be a valuable resource for Pakistani consumers, as it would allow them to compare the prices of products on different e-commerce websites and find the best deals. The website would also be a valuable resource for businesses, as it would allow them to track the prices of their products on different e-commerce websites and ensure that they are competitive.

Benefits of the Project

The proposed website would offer a number of benefits to both consumers and businesses:

Consumers:

Save time and money by comparing the prices of products on different e-commerce websites.

Businesses:

Track the prices of their products on different e-commerce websites and ensure that they are competitive.

Identify new business opportunities by tracking the popularity of products on different e-commerce websites.

Project Timeline:

The project is expected to take four months to complete. The following is a tentative project timeline:

- Month 1: Design the website and database.
- Month 2: Develop the web scraper scripts.
- Month 3: Integrate the web scraper scripts with the website.
- Month 4: Test and deploy the website.

Conclusion:

The proposed website would be a valuable resource for both consumers and businesses in Pakistan. The website would allow consumers to compare the prices of products on different e-commerce websites and find the best deals. The website would also allow businesses to track the prices of their products on different e-commerce websites and ensure that they are competitive.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1: Introduction

In response to the growing trend of online shopping in Pakistan, our team is developing a website dedicated to streamlining the process of finding the best deals. With an increasing number of consumers opting for online purchases, the challenge lies in efficiently comparing prices across various e-commerce platforms. Our solution involves scraping the web to provide users with instant access to prices for a specified product. This user-friendly platform aims to alleviate the time-consuming task of manually comparing prices, making it easier for consumers to find the most cost-effective options. By offering this service, our website not only benefits Pakistani shoppers by saving them time and money but also serves as a valuable tool for businesses seeking to stay competitive by monitoring their product prices across multiple platforms.

1.1. Background

Our project originated from the dynamic e-commerce landscape in Pakistan, where the surge in online shopping presented a challenge for consumers seeking the best deals. Recognizing the need for a more efficient comparison tool, our team initiated the development of a dedicated website. Through comprehensive market research, we identified a gap in the user experience, prompting the creation of a centralized solution leveraging web scraping technology. This innovative platform aims to streamline the process of comparing prices across various reputable e-commerce websites, providing users with real-time data. Our mission is to enhance the consumer experience by offering a convenient hub for price and product information, ultimately empowering users to make informed purchasing decisions. Simultaneously, the project aligns with the broader goal of fostering healthy competition among businesses by assisting them in maintaining competitiveness in the evolving online market.

1.2. Motivations and Challenges

Motivations

This project focuses on enhancing the online shopping experience for Pakistani consumers. It aims to do this by developing a user-friendly tool that streamlines the shopping process. By allowing users to quickly compare prices across different online stores, the project hopes to save them both time and money. This initiative is driven by market research that identified a

significant need for a centralized and user-friendly platform dedicated to price comparisons. As a result, the project not only benefits consumers but can also foster healthy competition among e-commerce businesses. By providing these businesses with insights into market pricing dynamics, the project can help them refine their pricing strategies and stay competitive.

Challenges

One of the biggest hurdles this project faces is ensuring the accuracy of the data it collects. Inaccurate data can lead to misinformation for users, potentially causing them to make poor purchasing decisions. Another significant technical challenge lies in adapting to the wide variety of structures and layouts used by different e-commerce websites. This variability requires a sophisticated web scraping approach to effectively extract the desired information.

Balancing the need to collect data for price comparisons with the importance of user privacy is a crucial aspect of this project. The solution must comply with relevant regulations and maintain user trust. Furthermore, the dynamic nature of online markets presents a challenge. Product prices and availability can change rapidly across various platforms, necessitating robust mechanisms for real-time data updates to ensure users have access to the most current information.

Finally, safeguarding the website against potential security threats is paramount. This includes protecting against hacking attempts and data breaches to ensure the security of user information and the overall integrity of the project.

1.3. Goals and Objectives

User Benefits and Project Goals:

This project aims to empower Pakistani consumers by providing a user-friendly platform for efficient price and product comparisons. By streamlining the online shopping process, users can save valuable time previously spent manually checking multiple e-commerce websites. This not

only saves time but also allows users to make informed purchasing decisions that align with their budget, fostering a sense of financial prudence.

Impact on Businesses and Market Dynamics:

The project can also benefit businesses by providing them with insights into their product pricing relative to the market, helping them stay competitive. Additionally, promoting transparency in the online market by presenting users with accurate and up-to-date information on product prices and reviews fosters a fairer and more competitive environment.

Technical Challenges and Safeguards:

However, ensuring the accuracy and reliability of scraped data is crucial to provide users with trustworthy information for decision-making. This necessitates prioritizing user privacy and data protection by adhering to ethical standards and legal requirements in information handling. Furthermore, the project needs to be adaptable to the diverse structures and layouts of different e-commerce websites for comprehensive coverage.

Maintaining Relevance and User Trust:

To maintain user trust and ensure the platform's usefulness, robust mechanisms for real-time data updates are required to keep pace with dynamic changes in product prices and availability. Ultimately, the project strives for a positive and seamless user experience, fostering long-term user engagement and satisfaction with the comparison platform.

1.4. Literature Review/Existing Solutions

Understanding the current landscape of price and product comparison solutions is crucial for identifying opportunities and potential gaps that your project can address. These solutions can be broadly categorized as follows:

Price Comparison Features Within E-commerce Platforms: Established websites like PriceOye allow users to compare prices of different products that are available on their own website. Although convenient, this feature lacks what our project envisions to provide that is enabling the user to retrieve information from all across the web.

Mobile Applications: Mobile apps like RedLaser and ShopSavvy offer barcode scanning and product search functionalities, retrieving price and review information. These solutions might come in handy while you need to know what a product is and how well it is doing in the consumer market but when it comes to finding a better deal, a more comprehensive solution is needed.

User-Generated Content Platforms: Online communities on platforms like Reddit, Quora, and dedicated review sites play a significant role. User insights, reviews, and recommendations based on real-world experiences inform user decisions. Additionally, social media groups often share deals, discounts, and product reviews, fostering a network for budget-conscious shoppers. While valuable, such information may not be readily centralized or easily comparable.

Other Solutions: Retailer price matching policies encourage users to find lower prices elsewhere and potentially obtain a price match. Alert services notify users of price drops on specific products. APIs offered by some services enable developers to integrate real-time data into their applications. These solutions offer specific functionalities, but may not provide a comprehensive comparison platform.

These existing solutions offer valuable services, but there may be opportunities for improvement. Our project can potentially address limitations in scope, data accuracy, user experience, or customization options. In the following section, we'll delve deeper into the specifics of your project and how it aims to enhance the current landscape of price and product comparison tools.

1.5. Gap Analysis

The gap analysis for this project highlights several shortcomings in existing solutions. Currently, there is no single platform that effectively combines prices and reviews from various e-commerce

websites, forcing users to check multiple sources. Many platforms don't cover all e-commerce sites, which leads to incomplete product and price comparisons. Additionally, many existing tools struggle with real-time updates, often displaying outdated product prices and availability.

The user interfaces of some tools are overly complex, making it difficult for users to compare products easily. Privacy issues are also a concern, as many platforms do not adequately protect user data. Furthermore, existing solutions do not help businesses understand how their product prices compare to the market, limiting their ability to make strategic decisions. The adaptability of these tools to the different structures of various e-commerce websites is another issue, affecting the accuracy of the data.

Mobile experiences with many of these solutions are inefficient, and inconsistencies in product matching algorithms add to the problems. Lastly, the lack of community engagement in some solutions means users miss out on valuable insights and recommendations from other shoppers.

1.6. Proposed Solution

Our proposed solution is a website designed to help users find the best prices for products by searching various e-commerce websites. Users will simply enter the name of the product they are looking for into a search bar. The website will then scrape the internet for this product, gathering information such as the product name, description, and price from different e-commerce sites. This information will be displayed to the user in a clear and concise format. The website will be user-friendly and easy to navigate, with a design that works well on both smartphones and tablets.

Benefits of the proposed solution

For Pakistani consumers, this solution offers several advantages:

- **Time and Money Savings:** Users can quickly compare prices from different e-commerce websites to find the best deals.
- **Informed Decisions:** Users can read reviews of products before making a purchase.

- **Best Deals:** Users can easily identify the best online deals.

For businesses, the website will be a valuable tool as it allows them to:

- **Price Tracking:** Monitor the prices of their products across various e-commerce platforms.
- **Competitiveness:** Ensure their prices are competitive.
- **Market Insights:** Identify new business opportunities by tracking the popularity of products on different e-commerce sites.

1.7. Project Plan

Project Initiation

- Define the project objectives and scope.
- Conduct a thorough market analysis to identify key competitors and understand user needs.
- Establish roles and responsibilities for team members.

Research and Requirements Gathering

- Conduct user interviews and surveys to understand preferences and pain points.
- Research web scraping technologies for data collection from various e-commerce websites.
- Define technical requirements for the platform's architecture and user interface.

Platform Design

- Develop wireframes and prototypes for the user interface.
- Design the database structure for storing scraped data.
- Create a detailed technical design document outlining the system architecture and features.

Development

- Implement web scraping algorithms for extracting data from target e-commerce websites.
- Build the platform's frontend and backend components based on design specifications.
- Integrate real-time data updates and ensure adaptability to diverse website structures.

Privacy and Security Implementation

- Implement robust privacy measures and data protection mechanisms.
- Conduct security assessments to identify and address potential vulnerabilities.

Mobile Optimization

- Optimize the platform for mobile devices to ensure a seamless user experience on various screen sizes.
- Conduct thorough testing across different mobile platforms to ensure responsiveness.

Community Engagement Features

- Integrate features like user reviews, recommendations, and a discussion forum.
- Implement user authentication and moderation tools to ensure a positive and secure community environment.

Testing

- Conduct comprehensive testing of the platform, including functionality, security, and user experience.
- Gather feedback from a select group of users through beta testing.

Launch and Marketing

- Plan a strategic launch, including marketing campaigns to promote the platform.

- Monitor user feedback and address any post-launch issues promptly.

Monitoring and Optimization

- Implement analytics tools to monitor user activity and engagement.
- Continuously optimize the platform based on user feedback and changing market trends.

Documentation and Training

- Create user documentation and tutorials for effective platform usage.
- Provide training sessions for the support team to handle user queries and issues.

Post-launch Support

- Establish a customer support system to address user inquiries and concerns.
- Implement a system for regular updates and improvements based on user feedback.

1.7.1. Work Breakdown Structure

Requirements and Design:

- Sammar: Develop user interface wireframes.
- Both: Develop a high-level implementation plan.

Web Scraper:

- Both: Develop scraper for scraping the web.

Website:

- Sammar: Integrate web scraper scripts with website.
- Wahaj: Develop the website using React, Node, Express and Mongo DB.

Deployment:

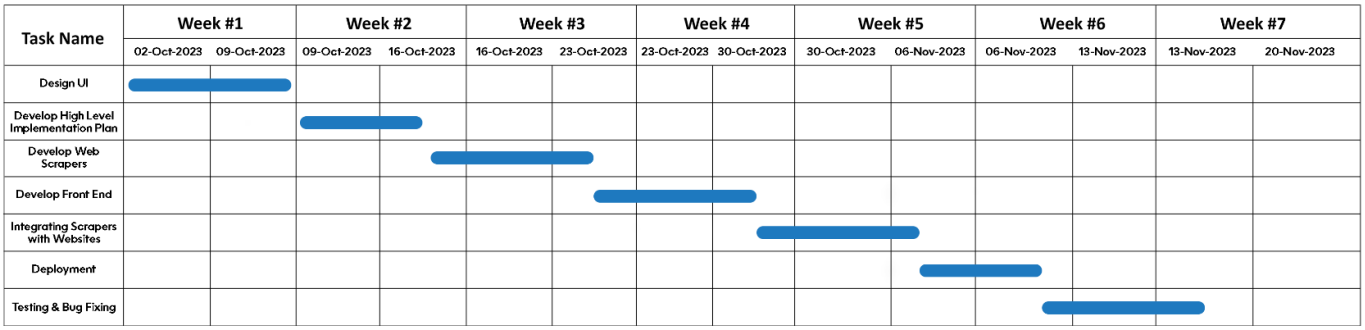
- Both: Fix any bugs found in the website and deploy the website to a web server

1.7.2. Roles & Responsibility Matrix

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Design UI on figma	1		7	Sammar Aleem
2	Develop high level implementation plan	1		4	Wahaj Khan & Sammar Aleem
3	Develop web scraper	2		7	Wahaj Khan & Sammar Aleem
4	Develop Website	3		7	Wahaj Khan
5	Integrating scrapers with website	3		7	Sammar Aleem
6	Deployment	4		4	Both
7	Testing and bug fixing	5		7	Both

Roles & Responsibility matrix 1

1.7.3. Gantt Chart



Gantt chart 1

1.8. Report Outline

The report on the Centralized E-commerce Price Comparison Platform begins with an Executive Summary, offering a concise overview of the project's significance and summarizing key findings. The Introduction section provides context and outlines the report's purpose, followed by a Project Overview detailing the goals and features of the platform. Market Analysis explores the current e-commerce landscape in Pakistan and identifies user needs and market trends. The Methodology section explains the project's approach and technical aspects, leading into a discussion of the platform's Technical Architecture and User Experience Design. Privacy and Security Measures are detailed, emphasizing the implemented safeguards. Challenges faced during implementation are discussed, along with strategies employed to overcome them. Testing and Quality Assurance procedures are outlined, followed by insights into the Launch and Marketing phases. Monitoring and Optimization strategies are explored, and the report delves into Documentation, Training, Post-launch Support, and Future Developments. The Conclusion summarizes key achievements, while the Recommendations section suggests future enhancements. Appendices provide supporting materials, and the report concludes with a References section citing relevant sources.

1.9. Empathy Map

Paint Points	Gains	Needs	Wants
Wasting time and effort comparing prices on different websites. Feeling overwhelmed by the sheer volume of options. Concerns about missing out on better deals.	Time saved by having a tool that simplifies price comparison. Confidence in making better-informed purchase decisions. The ability to find the best deals and save money.	A user-friendly website that simplifies price comparison. Access to accurate and up-to-date pricing information. Reliable and trustworthy product reviews.	A way to track prices over time for her favorite products. A personalized shopping experience that suggests deals based on her preferences. A mobile app for on-the-go price comparisons.

Empathy Map 1.9

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

This chapter outlines the Software Requirement Specifications (SRS) for the proposed web scraping platform. The SRS serves as a comprehensive guide, detailing the functional and non-functional requirements necessary for the successful development and implementation of the platform. It ensures all stakeholders have a clear understanding of the project's goals, functionalities, and constraints.

2.1.1. Purpose

This Software Requirement Specifications (SRS) document outlines the requirements for the development of a web scraping platform designed to aggregate product prices and reviews from various e-commerce websites. This document pertains to version 2.0 of the platform.

The scope of this SRS covers the entire system, detailing both the frontend and backend components. The document specifies the functional requirements, such as user input handling, data scraping algorithms, and data presentation, as well as non-functional requirements, including performance, security, and usability standards. It provides a comprehensive guide for the development team to ensure the platform meets the needs of its users and stakeholders.

This SRS does not cover the subsequent maintenance and updates post-initial release. Future revisions and new features will be documented in subsequent versions of the SRS. The primary focus of this document is to establish a solid foundation for the initial development and deployment of the platform, ensuring it fulfills its intended purpose of providing users with a centralized tool to compare product prices and reviews across multiple e-commerce websites.

2.1.2. Document Conventions

This Software Requirement Specifications (SRS) document follows specific standards and typographical conventions to ensure clarity and consistency throughout. These conventions are outlined below:

- **Font and Text Styles**:
  - **Normal text**: Calibri, 12-point font.
  - **Headings**: Bold, Calibri, 14-point font for section headings (e.g., 2.1 Introduction), and 12-point font for subsection headings (e.g., 2.1.1 Purpose).
  - **Keywords**: Important terms and keywords are italicized for emphasis.
  - **Code snippets**: Monospaced font (Courier New, 10-point) to distinguish from regular text.
- **Requirement Priorities**:
  - **Must-have requirements** are prefixed with "MUST".
  - **Should-have requirements** are prefixed with "SHOULD".
  - **Could-have requirements** are prefixed with "COULD".
  - **Will-not-have requirements** (future consideration) are prefixed with "WILL NOT".
- **Highlighting**:
  - **Bold text** is used for critical information and section headings.
  - **Italicized text** highlights key concepts and definitions.
- **Numbering**:
  - Each requirement is uniquely identified by a numbering scheme that corresponds to its section and subsection. For example, a functional requirement in section 3.2 might be numbered as 3.2.1.
- **Lists**:
  - Bulleted lists are used sparingly and primarily for enumerating items that do not follow a strict order.
  - Numbered lists are used for steps that need to be followed in sequence.
- **Tables and Figures**:

    - All tables and figures are labeled and numbered consecutively within each section (e.g., Table 3.1, Figure 4.2).
    - Captions for tables are placed above the table, while captions for figures are placed below the figure.
- **References and Citations**:
    - External references are cited using a numerical system and listed in a References section at the end of the document.
    - In-text citations are denoted by square brackets, e.g., [1].
- **Inheritance of Priorities**:
    - Higher-level requirements' priorities are assumed to be inherited by detailed requirements unless explicitly stated otherwise.
    - Each requirement statement includes its own priority designation if it differs from the inherited priority.

These conventions are intended to provide a structured and consistent approach to documenting the requirements, making it easier for stakeholders to review and for developers to implement the specifications.

2.1.3. Intended Audience and Reading Suggestions

This Software Requirement Specifications (SRS) document is intended for a diverse group of readers who are involved in various aspects of the project. The different types of readers and their interests are as follows:

- **Developers**: Interested in the technical requirements, system architecture, and detailed specifications for both frontend and backend development.
- **Project Managers**: Focus on the project scope, objectives, timelines, and overall structure of the document to ensure the project stays on track.
- **Marketing Staff**: Look for insights into the platform’s features and benefits to develop effective marketing strategies.

- **Users**: Seek to understand how the platform will address their needs and the user experience it will provide.
- **Testers**: Need detailed functional and non-functional requirements to develop comprehensive test plans and ensure the system meets its specifications.
- **Documentation Writers**: Require a clear understanding of the system's functionality and features to create user manuals and support documentation.

Recommended Reading Sequence

1. **Project Managers**: Start with the Introduction and Overall Description to understand the project's scope and objectives, then proceed to System Attributes and Other Requirements for project management details.
2. **Developers**: Begin with the Specific Requirements and System Features to grasp the technical specifications, followed by External Interface Requirements and System Attributes for implementation details.
3. **Marketing Staff**: Read the Introduction and System Features to understand the product's benefits and unique selling points, then review the Overall Description for market analysis insights.
4. **Users**: Focus on the Introduction and System Features to see how the platform will address user needs and enhance the user experience.
5. **Testers**: Start with the Specific Requirements and System Features to develop test cases, followed by External Interface Requirements and System Attributes for thorough testing coverage.
6. **Documentation Writers**: Begin with the Specific Requirements and System Features to understand the system's functionality, then proceed to the Introduction and Overall Description to provide context in the documentation.

By following these reading suggestions, each type of reader can efficiently extract the relevant information needed for their specific role in the project.

2.1.4. Product Scope

The software specified in this document is a web scraping platform designed to aggregate product prices and reviews from various e-commerce websites. Its primary purpose is to provide users with a centralized tool for:

- **Comparing Prices:** Facilitating quick and efficient price comparisons across multiple e-commerce sites.
- **Reading Reviews:** Offering comprehensive product information and user reviews to aid informed purchase decisions.
- **Finding Best Deals:** Providing real-time updates on product prices and availability to help users find the best online deals.

Purpose and Objectives:

- **Purpose**: Simplify the online shopping experience by consolidating product information from diverse sources.
- **Objectives**:
    - Enable users to make informed purchase decisions based on accurate pricing and review data.
    - Support businesses in optimizing pricing strategies and monitoring market competitiveness.

Benefits:

- **Time and Cost Savings**: Users save time and money by efficiently comparing prices without visiting multiple websites.
- **Enhanced User Experience**: Empowers users with information to make confident buying decisions.
- **Business Insights**: Provides businesses with competitive intelligence and market trends to enhance strategic pricing decisions.

Alignment with Business Strategies: The platform supports corporate goals of improving customer satisfaction and expanding market reach by:

- Enhancing customer experience through streamlined price comparison and review access.
- Driving sales through competitive pricing strategies and market analysis.

Vision and Scope Document: For a detailed strategic overview, refer to the accompanying Vision and Scope document, which outlines specific business objectives and market positioning strategies for the platform.

By focusing on user empowerment and business competitiveness, the web scraping platform aims to become a pivotal tool in the online shopping landscape, benefiting both consumers and businesses alike.

2.2. Overall Description

2.2.1. Product Perspective

The PriceSnap system is conceived as a standalone product within the domain of centralized e-commerce price comparison platforms. It is not a follow-on member of an existing product family but rather a novel solution aimed at revolutionizing online shopping experiences in Pakistan. PriceSnap is designed to be a self-contained, comprehensive tool that enables users to compare product prices and reviews across various e-commerce websites efficiently.

In the broader context, PriceSnap functions as an independent component but can integrate seamlessly with larger e-commerce ecosystems. While it operates independently, it enhances the functionality of existing online retail environments by providing users with a centralized platform for informed decision-making. The system interacts with external entities, mainly e-commerce websites, through web scraping methodologies.

A visual representation of the system's major components, subsystem interconnections, and external interfaces will be provided in a diagram for clarity, elucidating the relationships and interactions between PriceSnap and the broader e-commerce landscape. This diagram will serve

as a valuable reference point to understand the contextual positioning of PriceSnap within the larger system.

2.2.2. Product Functions

The web scraping platform must perform or enable users to perform the following major functions:

- **Product Search and Input**
  - Allow users to enter the name or description of a product they wish to search for.
  - Validate and process user input to initiate search queries.
- **Data Scraping and Aggregation**
  - Automatically scrape and extract product information (name, description, price) from multiple e-commerce websites.
  - Aggregate and organize scraped data into a unified format for display.
- **Real-Time Updates**
  - Provide real-time updates on product prices and availability from the scraped websites.
  - Ensure timely synchronization of data to reflect current market conditions.
- **User Interface and Experience**
  - Design an intuitive and user-friendly interface for displaying product information.
  - Facilitate easy navigation and interaction for users to compare prices and read reviews.
- **Privacy and Security**
  - Implement robust privacy measures to protect user data and ensure compliance with data protection regulations.
  - Secure data transmission and storage to prevent unauthorized access.
- **Mobile Optimization**
  - Optimize the platform for mobile devices to ensure a seamless user experience across different screen sizes and resolutions.

    - Conduct thorough testing to validate responsiveness and usability on mobile platforms.
- **Community Engagement Features**
    - Integrate user review systems, recommendations, and a discussion forum to enhance user engagement.
    - Implement moderation tools to maintain a positive and secure community environment.

These functions collectively aim to deliver a comprehensive and user-centric platform for efficient price comparison and product information retrieval across diverse e-commerce websites. Detailed specifications for each function will be provided in Section 3 of this SRS, outlining specific requirements and interactions within the system.

2.2.3. User Classes and Characteristics

Regular Shoppers:

- Characteristics: Infrequent to frequent users, diverse technical expertise.
- Usage: Primarily engages in price comparisons and product reviews for occasional to regular online shopping.

Power Shoppers:

- Characteristics: Frequent users, advanced technical skills.
- Usage: Regularly employs advanced features for in-depth product comparisons, tracking price trends, and leveraging personalized settings.

Businesses/Sellers:

- Characteristics: Vendor accounts, varied technical expertise.
- Usage: Utilizes the platform to monitor product pricing trends, assess market competitiveness, and adjust their pricing strategies accordingly.

Administrators:

- Characteristics: High technical proficiency, system management expertise.
- Usage: Manages and maintains the PriceSnap platform, oversees user accounts, and ensures the integrity of the web scraping and data storage processes.

Casual Browsers:

- Characteristics: Infrequent users, varying technical skills.
- Usage: Explores the platform casually for occasional product price checks and reviews without extensive engagement.

Distinguishing User Classes:

- **Priority Users:** Regular Shoppers and Power Shoppers, as they represent the primary user base and contribute significantly to the platform's usage.
- **Secondary Users:** Businesses/Sellers, as their interaction is crucial for market insights.
- **Support Users:** Administrators, who ensure the platform's smooth operation.
- **Peripheral Users:** Casual Browsers, whose needs are met with basic platform functionalities.

Understanding the diverse user classes ensures that the platform is tailored to meet the varied needs and technical competencies of its users, prioritizing features and functionalities accordingly.

2.2.4. Operating Environment

The PriceSnap software is designed to operate in a dynamic and evolving online environment, catering to users across various devices and platforms. The operating environment encompasses the following components:

Hardware Platform:

- PriceSnap is platform-agnostic, compatible with a range of devices, including desktops, laptops, tablets, and mobile phones.
- The hardware specifications required are minimal, ensuring accessibility for users with diverse devices and configurations.

Operating System and Versions:

- The web-based nature of PriceSnap ensures compatibility with major operating systems, including Windows, macOS, and Linux.
- The platform is optimized for modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

Web Browsers:

- PriceSnap is designed to seamlessly coexist with popular web browsers, ensuring a consistent user experience across different browsers and versions.
- Users are encouraged to keep their browsers updated to the latest versions for optimal performance and security.

Other Software Components:

- PriceSnap interacts with external e-commerce websites through web scraping methodologies based on the Google Palm V model API.
- Mongo DB is used for storing product information, ensuring scalability and real-time data updates.

Security Protocols:

- PriceSnap adheres to standard web security protocols to safeguard user data and ensure secure transactions.
- SSL/TLS encryption is implemented to protect user communications and sensitive information.

External APIs:

- PriceSnap interacts with external e-commerce websites using the Google Palm V model API. Continuous compatibility with this API is essential for seamless data extraction.

By adapting to diverse hardware configurations, operating systems, and web browsers, PriceSnap ensures a flexible and inclusive operating environment. Compatibility with external APIs and adherence to security protocols further contribute to a robust and secure user experience.

2.2.5. Design and Implementation Constraints

The design and implementation of the web scraping platform are constrained by several factors:

Regulatory Compliance: Ensure adherence to data protection regulations such as GDPR or CCPA to protect user privacy.

Technology Stack: Use specific technologies and tools mandated for development:

- Web scraping libraries/frameworks for data extraction.
- React.js for frontend development, leveraging its component-based architecture.
- Node.js for backend development, utilizing its scalability and asynchronous processing capabilities.
- MongoDB for efficient storage and retrieval of scraped data.

Performance Requirements: Handle concurrent data scraping operations efficiently to provide real-time updates and responsiveness.

Security Considerations: Implement robust security measures, including HTTPS encryption, to secure data transmission and storage against unauthorized access.

User Interface Design Standards: Adhere to organizational UI/UX design guidelines to ensure consistency and usability across different devices.

Maintenance Responsibility: The customer's organization will be responsible for ongoing maintenance and updates post-deployment, necessitating clear documentation and adherence to coding standards.

These constraints guide the development process, ensuring compliance with regulations, optimizing performance, enhancing security, and maintaining consistency throughout the platform's lifecycle.

2.2.6. User Documentation

The web scraping platform will include comprehensive user documentation components to facilitate effective usage and support:

User Manuals: Detailed guides explaining how to use the platform, including step-by-step instructions for performing key tasks such as product search, price comparison, and reading reviews.

Online Help: Context-sensitive help integrated within the platform to provide immediate assistance based on user actions and queries.

Tutorials: Interactive tutorials and walkthroughs to onboard new users and demonstrate platform features and functionalities.

FAQ Section: Frequently Asked Questions section addressing common user queries and troubleshooting tips.

Documentation Delivery Format:

- User manuals and tutorials will be provided in PDF format for easy printing and offline reference.
- Online help will be accessible within the platform via tooltips, help icons, and a dedicated help section.

Standards:

- Documentation will adhere to the organization's branding guidelines and style for consistency.
- Tutorials and user guides will use clear language and visual aids to enhance understanding and usability.

2.2.7. Assumptions and Dependencies

Assumptions

- **Data Availability**: Assumption that e-commerce websites will allow web scraping of product information without technical or legal restrictions.
- **Platform Compatibility**: Assumption that users will access the platform using modern web browsers (Chrome, Firefox, Safari) with JavaScript enabled.
- **Data Accuracy**: Assumption that scraped data from e-commerce websites will be accurate and reliable for price comparisons and product information.
- **Regulatory Compliance**: Assumption that the platform's implementation will comply with relevant data protection laws (e.g., GDPR, CCPA) regarding user privacy and data handling practices.

Dependencies

- **Third-Party Libraries and APIs**: Dependency on third-party web scraping libraries or frameworks for data extraction from e-commerce websites.
- **Database Integration**: Dependency on MongoDB for storing and retrieving scraped data efficiently.
- **Development Tools and Frameworks**: Dependency on React.js for frontend development and Node.js for backend development as specified in the technology stack.
- **Internet Connectivity**: Dependency on stable internet connectivity for real-time data updates and user interactions.
- **User Acceptance Testing**: Dependency on user feedback and acceptance testing to validate platform functionality and usability.
- **Compliance and Legal Review**: Dependency on legal and compliance reviews to ensure adherence to data protection regulations and other legal requirements.

2.3. External Interface Requirements

2.3.1. User Interfaces

Logical Characteristics:

Web Interface:

- Description: The primary interface for users will be a web-based platform accessible through standard web browsers.
- Layout Constraints: Adheres to responsive design principles for optimal viewing on various devices, including desktops, tablets, and mobile phones.
- GUI Standards: Follows intuitive and user-friendly GUI standards, with clear navigation menus, product search functionalities, and concise displays of price and review information.

Login and Authentication:

- Description: Users will access the platform through secure login credentials.
- Standard Buttons/Functions: Includes standard login and logout buttons. Password recovery and account creation functions follow industry standards.
- Error Message Display: Clearly communicates authentication errors with standard error messages for enhanced user understanding.

Product Comparison Screen:

- Description: Displays product information, prices, and reviews from various e-commerce websites.
- Screen Layout Constraints: Follows a standardized layout with clear sections for each product attribute, supporting easy comparison.
- Standard Buttons/Functions: Includes options for sorting, filtering, and saving product comparisons for personalized user experiences.

User Dashboard:

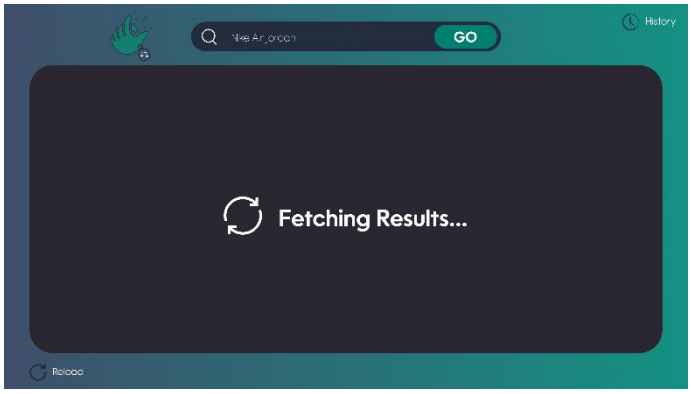
- Description: A personalized space where users can manage preferences, saved comparisons, and account settings.

- Screen Layout Constraints: Intuitive layout for easy navigation, presenting user data in a visually accessible manner.
- Standard Buttons/Functions: Includes options for updating preferences, managing saved comparisons, and adjusting account settings.

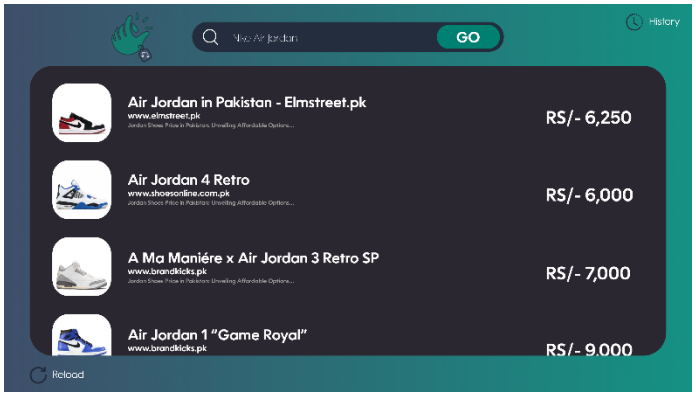
Help and Support:

- Description: Provides users with assistance and support features.
- Standard Buttons/Functions: Standard help buttons and links to support resources.
- Error Message Display: Clear and concise error messages for troubleshooting assistance.

UI screen 3



UI screen 2



UI screen 1

2.3.2. Hardware Interfaces

Logical Characteristics:

Device Compatibility:

Description: PriceSnap is designed to be platform-agnostic, supporting various hardware devices.

Supported Devices: Desktop computers, laptops, tablets, and mobile phones.

Data Interactions: The platform ensures seamless data interactions regardless of the device type.

Physical Characteristics:

Server Infrastructure:

Description: PriceSnap relies on server infrastructure to host the web application and execute web scraping operations.

Data Interactions: Data is transmitted between users' devices and the server for real-time updates and user interactions.

Communication Protocols: Utilizes standard HTTP/HTTPS protocols for secure data transmission.

User Devices:

Description: The physical devices used by users to access the PriceSnap platform.

Data Interactions: Users input queries, interact with the user interface, and receive real-time data updates.

Communication Protocols: Utilizes standard web protocols for communication between user devices and the server.

External APIs:

Description: Interaction with external e-commerce websites through the Google Palm V model API for web scraping.

Data Interactions: PriceSnap fetches product information from external websites for comparison.

Communication Protocols: Adheres to the API specifications, utilizing protocols defined by the google palm V model.

Communication Protocols:

HTTP/HTTPS:

Description: Used for communication between user devices and the server.

Nature of Interactions: Facilitates secure and encrypted data transmission.

Protocol Standards: Follows standard web communication protocols for data exchange.

API Protocols:

Description: Interaction with the Google Palm V model API for web scraping.

Nature of Interactions: PriceSnap sends requests to the API for fetching product information.

Protocol Standards: Adheres to the specifications defined by the Google Palm V model.

The logical and physical characteristics of hardware interfaces ensure that PriceSnap is versatile and can seamlessly operate on various devices while efficiently interacting with server infrastructure and external APIs. Standardized communication protocols contribute to the secure and reliable exchange of data between different components of the system.

2.3.3. Software Interfaces

The web scraping platform interfaces with several software components and services to facilitate its functionality:

- **Database Interface**:

  Database: MongoDB (version 4.4)

  Purpose: Store scraped data including product names, descriptions, prices, and availability.

  Data Items: Product details retrieved from e-commerce websites.

  Messages: CRUD operations (Create, Read, Update, Delete) to manage and retrieve data.

- **Web Scraping Libraries**:

Libraries: BeautifulSoup (version 4.9.3), Scrapy (version 2.5.0)

Purpose: Extract product information (name, description, price) from target e-commerce websites.

Data Items: Raw HTML data from web pages.

Messages: Scraping requests and data parsing instructions.

- **Frontend Framework**:

  Framework: React.js (version 17)

  Purpose: Provide user interface for interacting with the platform, displaying product information and search results.

  Data Items: User queries, product search results, user interface state.

  Messages: HTTP requests/responses for user interactions (search, filter, view details).

- **Backend Framework**:

  Framework: Node.js (version 14)

  Purpose: Handle business logic, manage data flow between frontend and database, and perform data processing tasks.

  Data Items: Processed product data, user authentication details, session management.

  Messages: API calls (GET, POST, PUT, DELETE) for data retrieval, manipulation, and authentication.

- **External APIs**:

APIs: Payment gateway API (e.g., Stripe), User authentication API (e.g., OAuth)

Purpose: Facilitate secure payment transactions and user authentication.

Data Items: Payment details, user credentials.

Messages: HTTPS requests/responses for payment processing and user authentication.

- **Operating System and Networking**:

  OS: Linux (Ubuntu 20.04 LTS)

  Purpose: Host the platform's server-side application, manage system resources, and handle network communications.

  Messages: TCP/IP protocol for network communication, file system operations for data storage.

- **Data Sharing Mechanism**:

  Implementation Constraint: Use of asynchronous communication patterns (e.g., callbacks, promises) in Node.js to handle concurrent data processing and ensure responsiveness.

2.3.4. Communications Interfaces

The web scraping platform requires various communications interfaces to ensure seamless interaction and data exchange:

- **HTTP/HTTPS Protocols**:

  Purpose: Facilitate communication between the client (web browser) and server (platform backend) for data retrieval, updates, and user interactions.

Message Formatting: JSON (JavaScript Object Notation) for structured data exchange.

Communication Standards: HTTP/1.1 or HTTP/2 for data transfer.

Security: Use of HTTPS to encrypt data transmission and ensure secure communication between clients and server.

- **RESTful API**:

  Purpose: Define endpoints for CRUD operations (Create, Read, Update, Delete) to interact with the platform's backend services.

  Message Formatting: JSON payloads for request and response bodies.

  Security: Implement authentication mechanisms (e.g., OAuth tokens) to authorize API requests and protect sensitive operations.

- **Email Notifications**:

  Purpose: Send transactional emails such as account verification, password reset, and order confirmations.

  Message Formatting: HTML or plain text emails with appropriate formatting and styling.

  Security: Use of SMTP protocol with TLS encryption to secure email communication.

- **Web Socket Protocol**:

  Purpose: Enable real-time communication between the client (web browser) and server for instant updates and notifications.

  Message Formatting: JSON payloads for exchanging real-time data.

Security: Implement WebSocket Secure (WSS) protocol to encrypt data transmission and ensure secure communication channels.

- **Data Transfer Rates and Synchronization**:

  Rates: Ensure high-speed data transfer rates to support real-time updates of product information.

  Synchronization: Use of asynchronous programming techniques (e.g., callbacks, promises) in backend services to handle concurrent data processing and maintain responsiveness.

2.4. System Features

2.4.1. Product Search and Retrieval

2.4.1.1. Description and Priority

Description: Allow users to search for products by entering keywords or product names, retrieving relevant information from multiple e-commerce websites.

Priority: High

2.4.1.2. Stimulus/Response Sequences

Stimulus: User enters a product name or keyword in the search bar.

Response: System initiates a search query and retrieves matching products from integrated e-commerce websites.

Stimulus: User applies filters (e.g., price range, brand).

Response: System updates search results based on applied filters and displays refined product listings.

2.4.1.3. Functional Requirements

REQ-SF1-1: The system shall provide a search bar on the homepage for users to enter product queries.

REQ-SF1-2: Upon receiving a search query, the system shall validate and process the input to initiate search requests to integrated e-commerce websites.

REQ-SF1-3: The system shall retrieve and display product information (name, description, price) from e-commerce websites based on the search query.

REQ-SF1-4: If no matching products are found, the system shall display a message indicating no results found.

REQ-SF1-5: The system shall support filtering options (e.g., by price, brand) to refine search results.

REQ-SF1-6: Error Handling:

REQ-SF1-6.1: If the user input is invalid (e.g., empty query), the system shall prompt the user to enter a valid search term.

REQ-SF1-6.2: If there are connectivity issues with e-commerce websites, the system shall display an error message and provide options to retry or contact support.

2.4.2. Product Price Comparison

2.4.2.1. Description and Priority

Description: Enable users to compare prices of the same product across different e-commerce websites.

Priority: High

2.4.2.2. Stimulus/Response Sequences

Stimulus: User selects a product from search results.

> **Response**: System displays prices of the selected product from various e-commerce platforms for comparison.

Stimulus: User adjusts filters (e.g., price range, shipping options).

> **Response**: System updates price comparison results based on applied filters and displays refined comparisons.

2.4.2.3. Functional Requirements

REQ-SF2-1: The system shall retrieve and display prices of the selected product from integrated e-commerce websites.

REQ-SF2-2: Provide tools for users to filter and sort price comparisons by criteria such as price, availability, and shipping options.

REQ-SF2-3: Implement a visual representation (e.g., charts, tables) of price comparisons to enhance user understanding.

REQ-SF2-4: Allow users to click through to view detailed product information on each e-commerce website directly from the comparison results.

REQ-SF2-5: Error Handling:

- **REQ-SF2-5.1**: If there are discrepancies in price data between platforms, the system shall highlight differences and provide explanations.
- **REQ-SF2-5.2**: If a website's data retrieval fails, the system shall notify the user and offer alternative sources or retry options.

2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

Performance requirements ensure optimal operation and responsiveness of the web scraping platform across different scenarios:

- **Response Time**: The system shall respond to user actions within 2 seconds under normal load conditions to enhance usability and user satisfaction.
- **Scalability**: The platform shall support up to 10,000 concurrent users without performance degradation, ensuring consistent service during peak traffic periods.
- **Data Processing Efficiency**: Updated product information shall be processed and displayed within 15 seconds of changes being detected on e-commerce websites, facilitating timely updates for users.
- **Reliability**: The platform shall maintain a system uptime of at least 99.9% per calendar month to minimize downtime and ensure continuous availability.
- **Error Handling**: Error messages and alerts shall be displayed to users within 5 seconds of an error occurring to provide prompt feedback and assistance.
- **Security Performance**: Secure transactions, including user authentication and payment processing, shall be completed within 3 seconds to ensure efficient and secure data handling.

2.5.2. Safety Requirements

Safety requirements for the web scraping platform address potential risks and ensure user protection:

- **Data Security**: Ensure sensitive user data (e.g., login credentials, payment information) is encrypted during transmission and storage to prevent unauthorized access.

- **User Privacy**: Comply with GDPR, CCPA, and other data protection regulations to safeguard user privacy and handle personal information responsibly.
- **Error Prevention**: Implement validation checks to prevent errors such as duplicate transactions or incorrect data inputs, minimizing financial and operational risks.
- **Regulatory Compliance**: Adhere to industry standards (e.g., PCI-DSS for payment processing) to mitigate legal risks and ensure secure handling of sensitive data.
- **Safety Certifications**: Obtain certifications like ISO 27001 for information security management to demonstrate adherence to established safety and security standards.

2.5.3. Security Requirements

Security requirements for the web scraping platform focus on protecting data and ensuring secure user interactions:

- **Data Encryption**: Ensure all sensitive data (e.g., user credentials, payment information) is encrypted both in transit and at rest to prevent unauthorized access.
- **Authentication**: Implement robust user authentication mechanisms (e.g., multi-factor authentication) to verify user identities securely.
- **Access Control**: Define roles and permissions to restrict access to sensitive data and functionalities based on user roles (e.g., admin, regular user).
- **Data Integrity**: Implement measures (e.g., checksums, digital signatures) to ensure data integrity and prevent unauthorized modifications.
- **Compliance**: Adhere to GDPR, HIPAA, or other relevant data protection regulations to protect user privacy and data handling practices.
- **Certifications**: Obtain certifications such as SOC 2, ISO 27001, or PCI-DSS to demonstrate compliance with security and privacy standards.

2.5.4. Software Quality Attributes

Software quality attributes for the web scraping platform ensure optimal performance and user satisfaction:

- **Usability**:
    - Ensure a user-friendly interface with intuitive navigation and clear instructions for ease of use.
    - Preference for ease of use over initial learning curve complexity.
- **Reliability**:
    - Maintain a system uptime of at least 99.9% per month to ensure consistent availability.
    - Quantitatively measure reliability through uptime metrics and error handling effectiveness.
- **Maintainability**:
    - Implement clear and modular code structure to facilitate ease of maintenance and updates.
    - Ensure documentation completeness and clarity to aid future development and troubleshooting.
- **Security**:
    - Achieve compliance with ISO 27001 and PCI-DSS for stringent security standards.
    - Quantify security effectiveness through regular security audits and vulnerability assessments.
- **Performance**:
    - Ensure responsiveness with average response times of under 2 seconds for user interactions.
    - Conduct load testing to verify scalability and performance under peak user traffic.
- **Interoperability**:
    - Support integration with major e-commerce platforms and APIs for seamless data exchange.
    - Quantify interoperability through successful integration tests and compatibility checks.

2.5.5. Business Rules

Business rules outline operational principles for the web scraping platform, guiding user roles and system behavior:

- **User Roles**:
  - Administrators have access to all platform functionalities, including user management and system configuration.
  - Regular users can perform product searches, view price comparisons, and read reviews but cannot modify system settings.
- **Data Usage**:
  - User data collected through the platform will not be shared with third parties without explicit user consent.
  - Data aggregation from e-commerce websites complies with site-specific terms of use and legal restrictions.
- **Transaction Handling**:
  - Transactions initiated through the platform must adhere to respective e-commerce site policies and payment processing regulations.
  - No automated bulk purchases or spamming activities are allowed to ensure fair use and prevent misuse.
- **Compliance**:
  - The platform adheres to copyright laws and respects intellectual property rights by displaying product information accurately and without infringement.
  - Users must comply with local, national, and international laws governing online activity and data privacy.

2.6. Other Requirements

Additional requirements for the web scraping platform:

- **Database**: Use scalable database architecture (e.g., MySQL, MongoDB) ensuring data integrity and efficient storage of large volumes of e-commerce data.
- **Internationalization**: Support multiple languages, currencies, and localization features for diverse user preferences and global markets.
- **Legal Compliance**: Adhere to GDPR, CCPA, and intellectual property laws governing data protection and content scraping.
- **Reusability**: Design modular components for easy integration and future scalability, documenting reusable elements for extended project use.
- **Performance Monitoring**: Implement tools for real-time performance monitoring and optimization to maintain system responsiveness and efficiency.

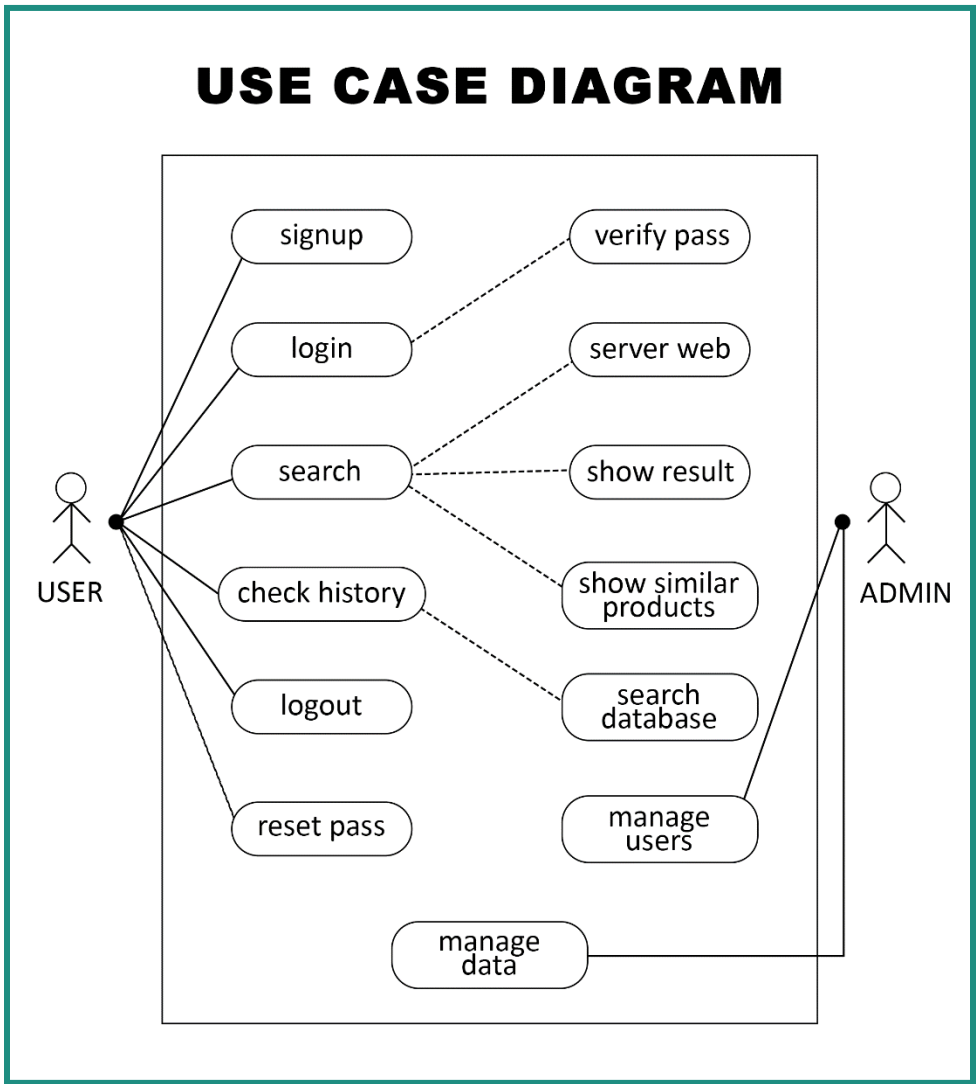
Chapter 3

Use Case Analysis

Chapter 3: System Analysis

This chapter focuses on a comprehensive analysis of the web scraping platform's requirements and functionalities. It explores the system's scope, user requirements, and technical specifications necessary for successful implementation. System analysis involves identifying user needs, defining system objectives, and detailing the functional and non-functional requirements essential for developing a robust and effective web scraping solution. The chapter aims to provide a clear understanding of the project's foundational aspects, guiding the subsequent phases of design and development.

3.1. Use Case Model



usecase diagram 1

3.2. Use Case Descriptions

Use Case: Register User

- Description: Allows a new user to create an account on the PriceSnap platform by providing necessary information.
- Actors: Guest, System
- Preconditions: User is not registered.
- Postconditions: User is registered and can log in.

Use Case: Search Products

- Description: Enables users to search for specific products on the PriceSnap platform.
- Actors: Registered User, System
- Preconditions: User is logged in.
- Postconditions: Search results are displayed based on user input.

Use Case: Compare Prices

- Description: Allows users to compare prices of a selected product from different e-commerce websites.
- Actors: Registered User, System
- Preconditions: User has selected products for comparison.
- Postconditions: Price and review comparisons are presented to the user.

Use Case: Customize Dashboard

- Description: Permits users to personalize their dashboard
- Actors: Registered User, System
- Preconditions: User is logged in.
- Postconditions: Dashboard reflects user-customized settings and displays preferred content.

Use Case: Web Scraping for Price Updates

- Description: Initiates the system to perform web scraping to update product prices from various websites.
- Actors: System

- Preconditions: None
- Postconditions: Product prices are updated in the system database.

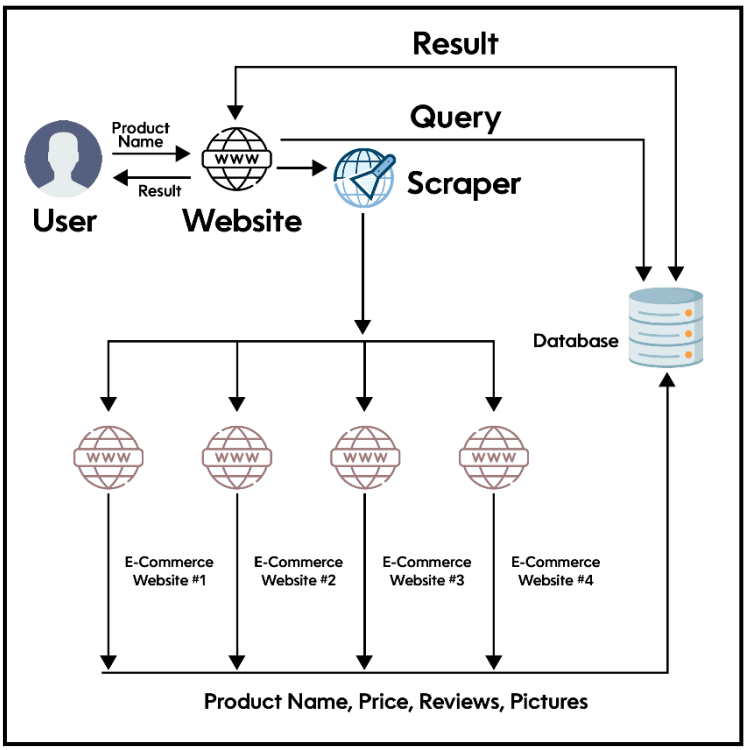
Chapter 4

System Design

Chapter 4: System Design

This chapter delves into the detailed design of the web scraping platform, translating requirements from the analysis phase into technical specifications and architectural blueprints. It outlines the system's structure, including database design, user interface layouts, and backend components. System design focuses on achieving scalability, reliability, and maintainability while ensuring alignment with user needs and business objectives. This chapter provides a roadmap for developers to implement and integrate various system components effectively, aiming to deliver a robust and user-friendly web scraping solution.

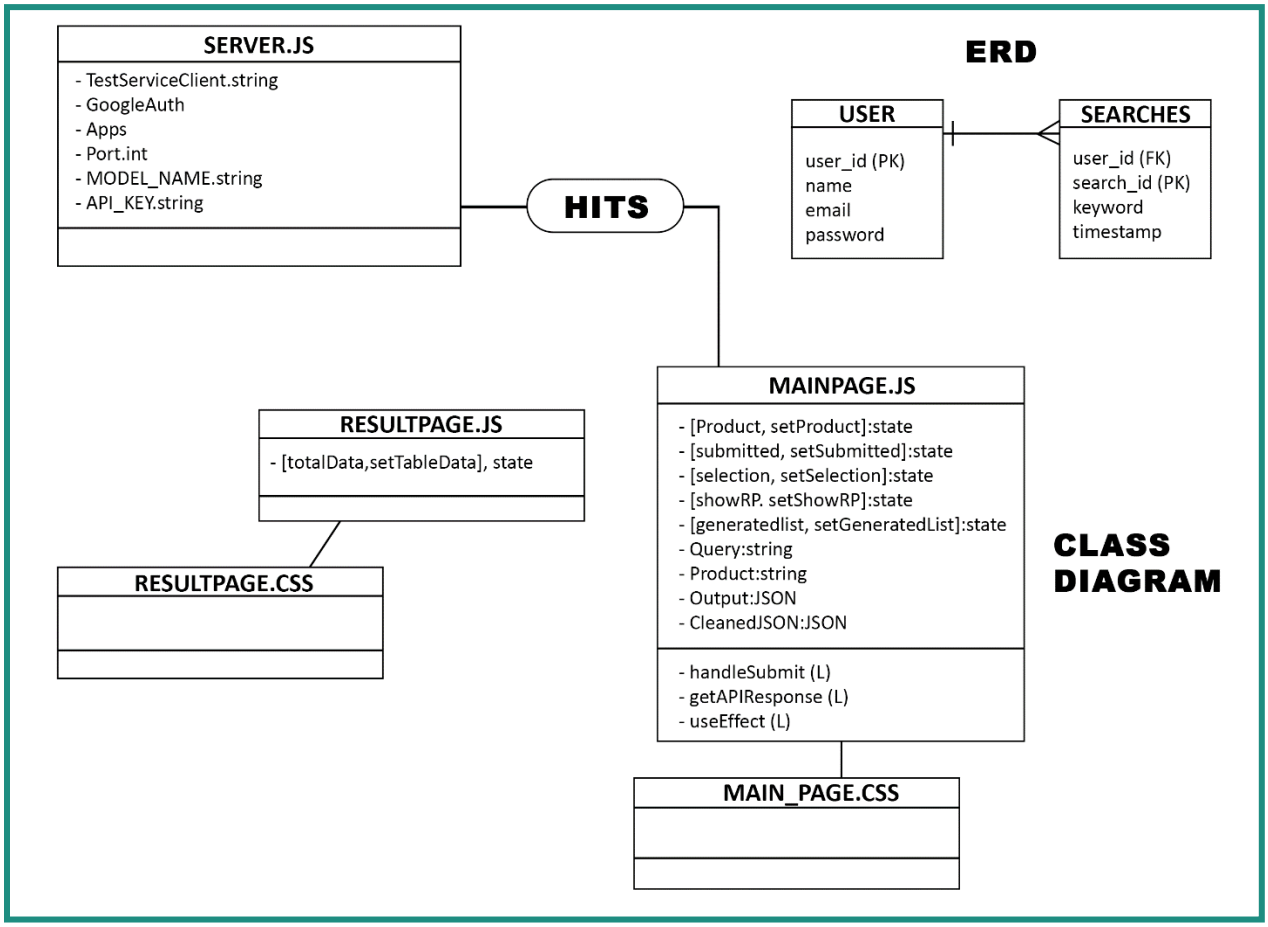
4.1. Architecture Diagram



architechture diagram 1

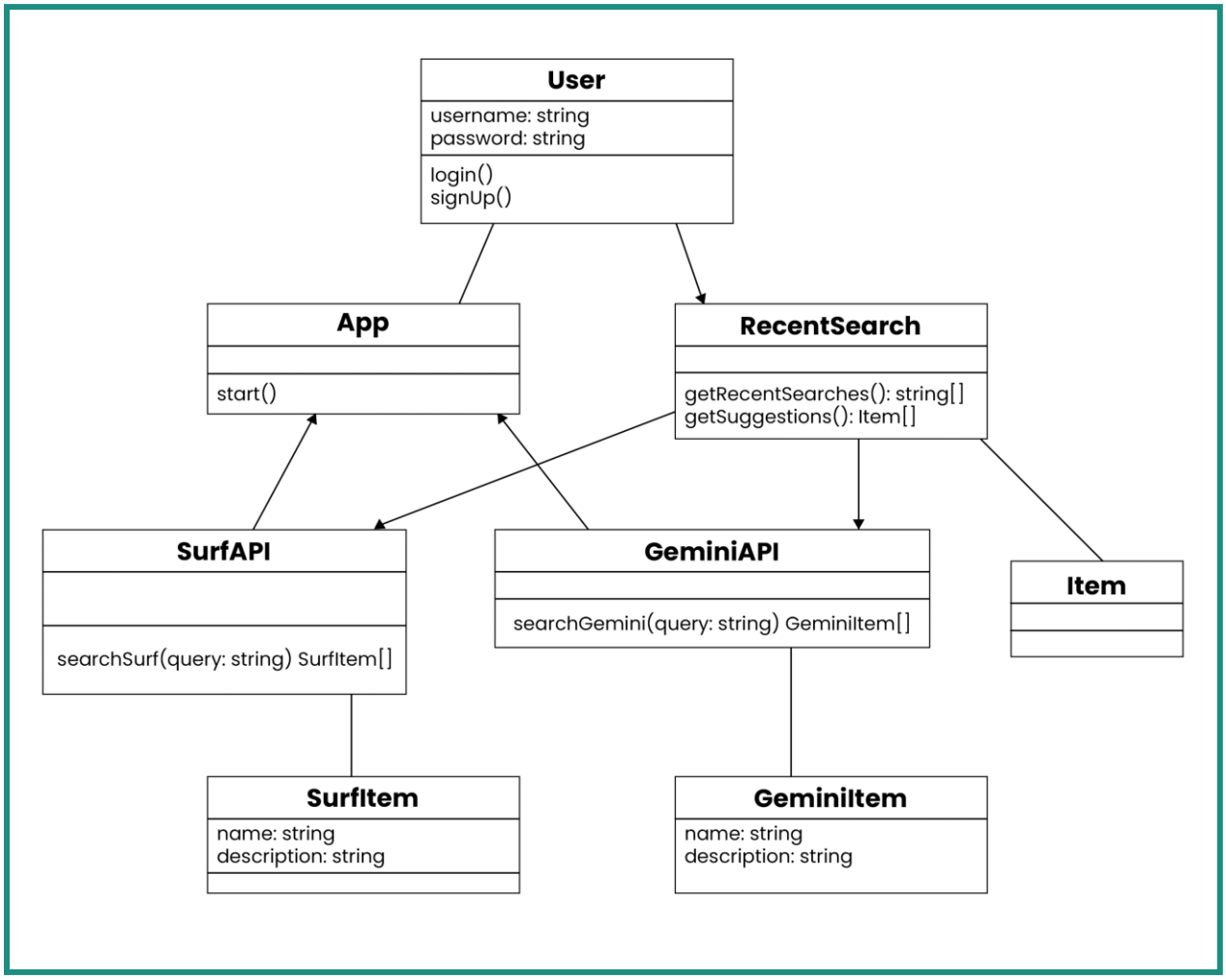
4.2. Entity Relationship Diagram

with data dictionary



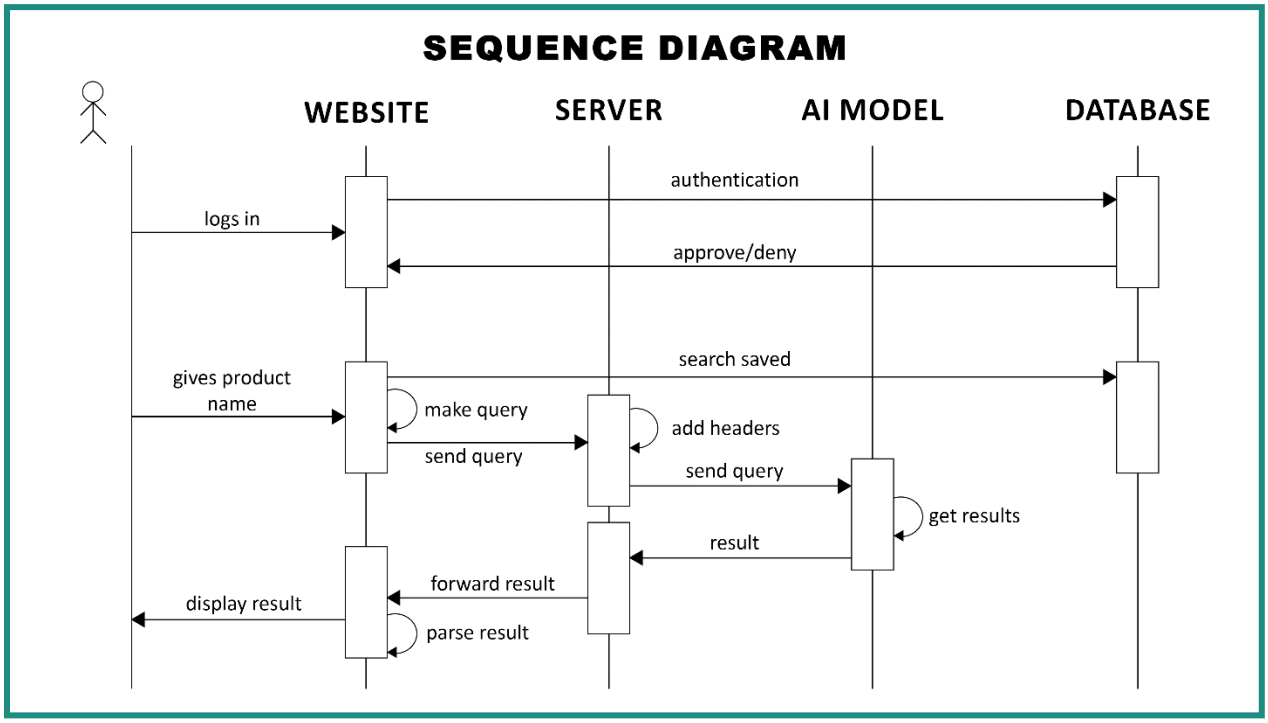
Entity relation diagram 1

4.3. Class Diagram



Class diagram 1

4.4. Sequence / Collaboration Diagram



sequence diagram 1

4.5. Operation contracts

1- contract: **SignUp**

pre-condition: user is on the login/signup page

post-condition: new user is created in the database

new user can now login

2- contract: **Login**

pre-condition: user is on the login/signup page

user has already signed up

post-condition: user will be allowed access to the website after their credentials are verified with the database.

user can now search for products.

3- contract: **Search**

pre-condition: user has already logged in

user has a stable internet connection

post-condition: user will be provided with price comparsions of the product they searched for from different websites accross the web

the product they search for will be stored in the database against their account.

the user will be shown other products similar to the one they searched for so they know their options.

user can search for other products

4- contract: **history**

pre-condition: user has already logged in

user has a stable internet connection

user clicked on the 'see history' button

post-condition: user will be shown the list of products they searched for

5- contract: **logout**

pre-condition: user has already logged in

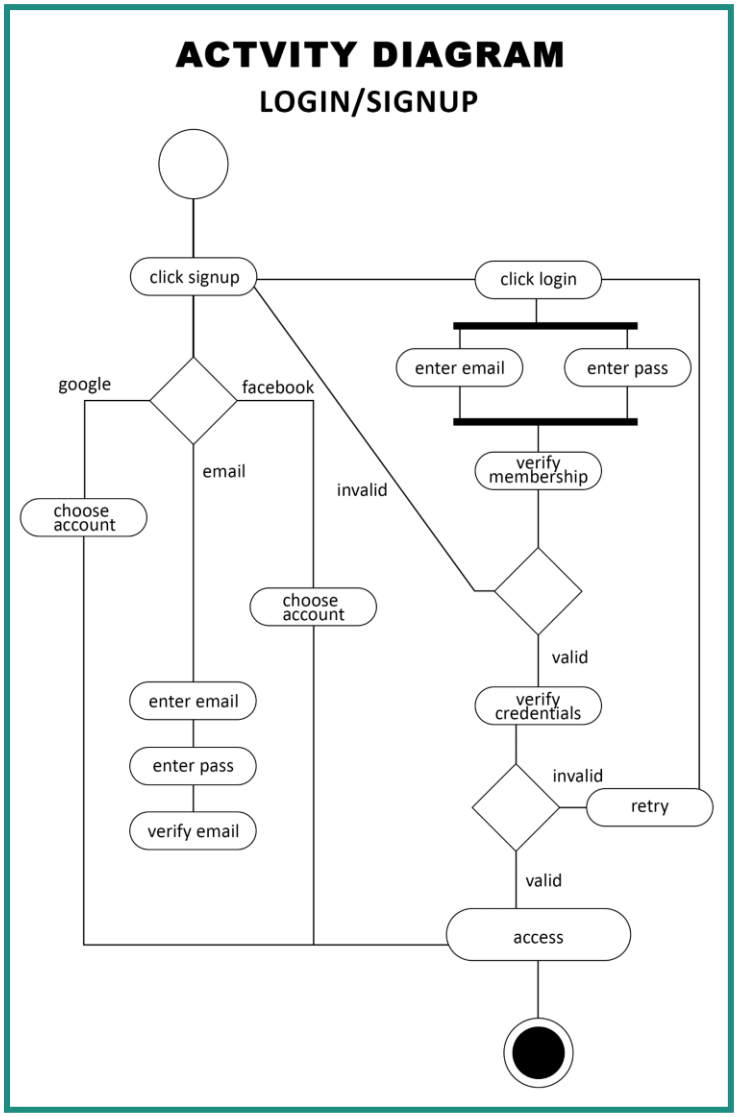
user has a stable internet connection

user clicked on the 'logout' button

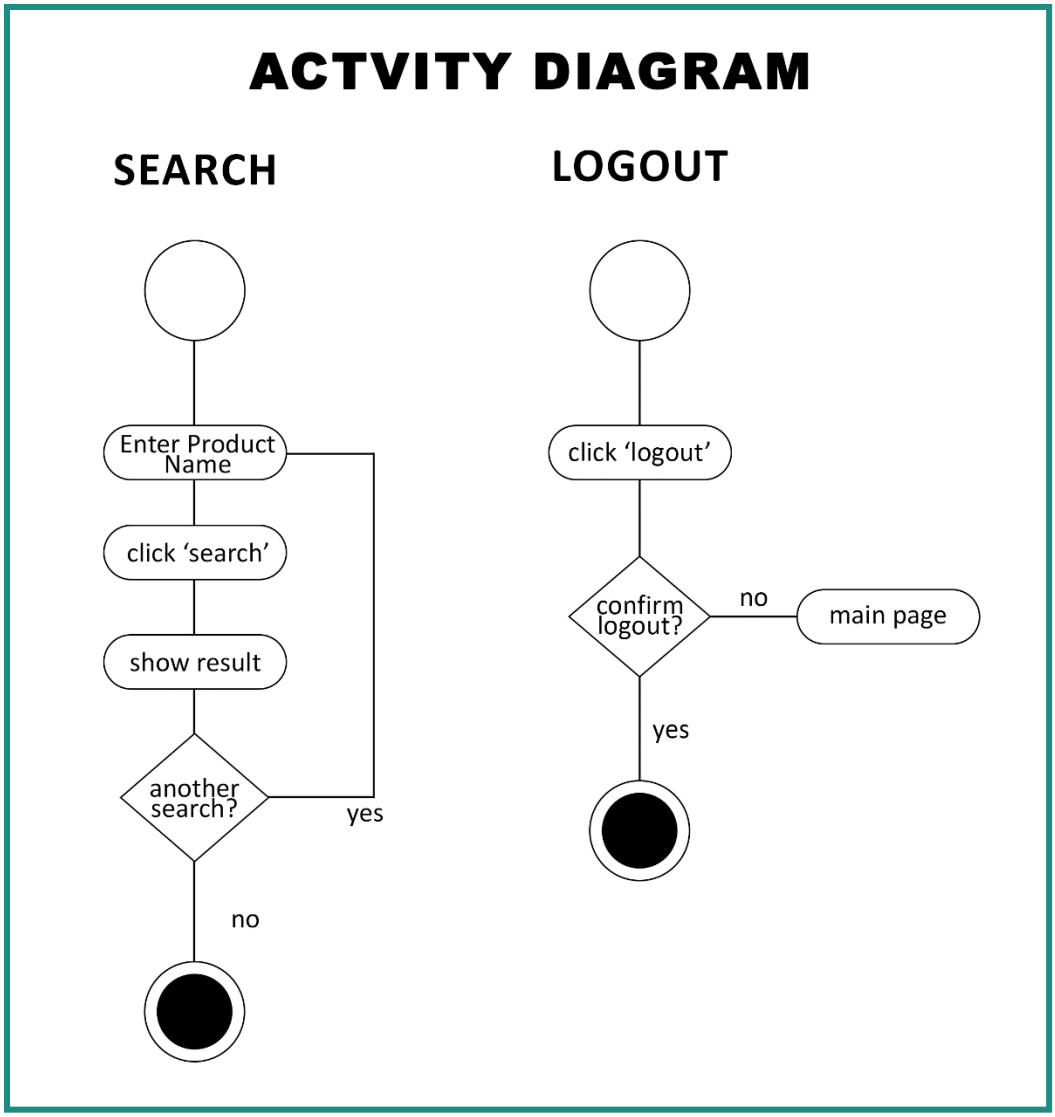
post-condition: user will be logged out of their account

user would need to log in again in order to use the service again

4.6. Activity Diagram

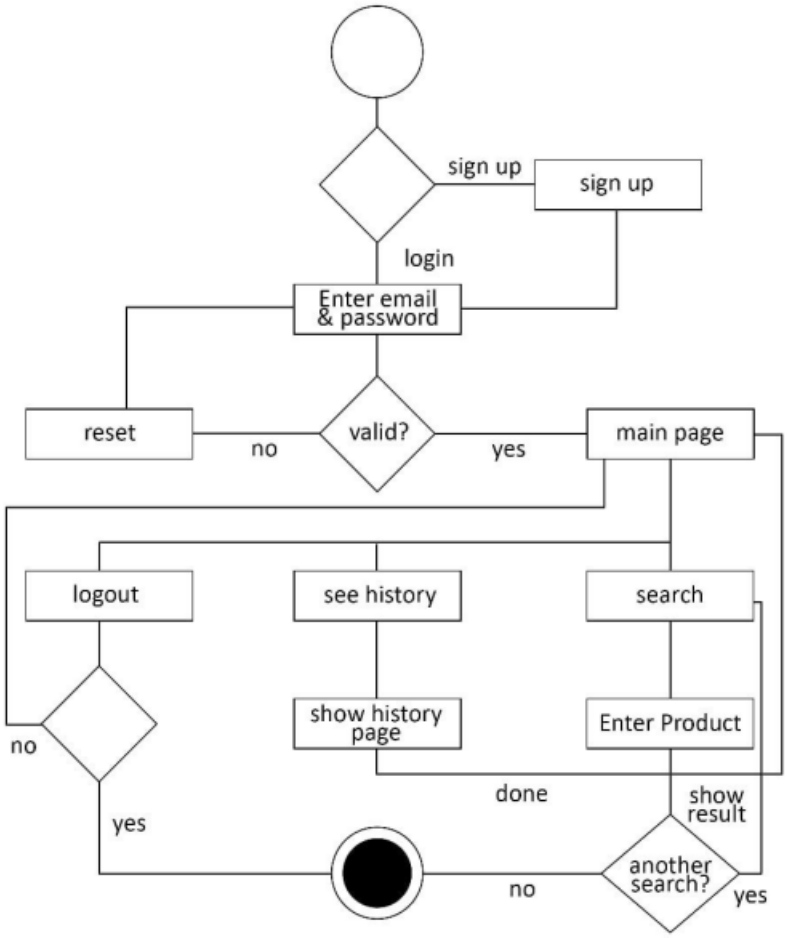


Activity Diagram 1



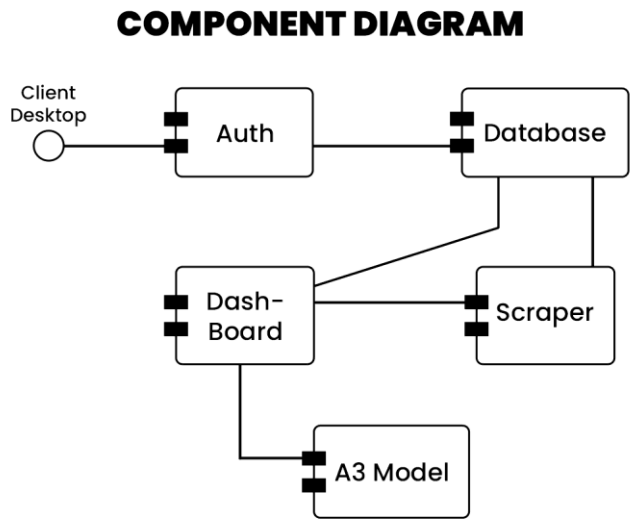
Activity Diagram 2

4.7. State Transition Diagram



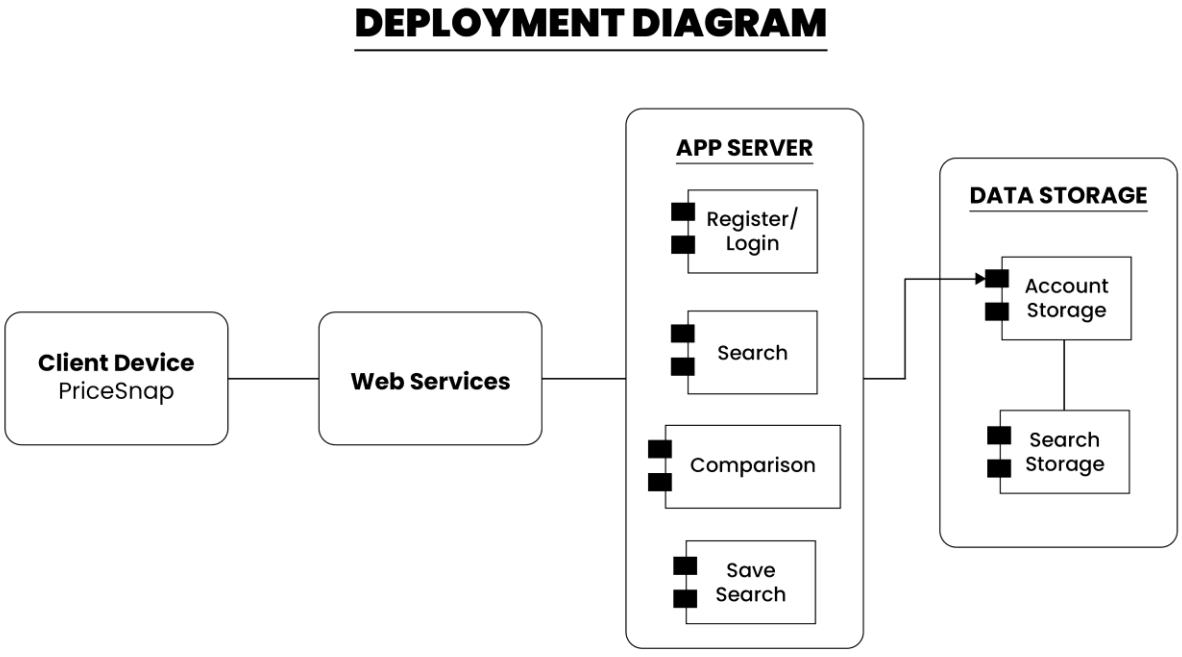
State Transition Diagram 1

4.8. Component Diagram



Component Diagram 1

4.9. Deployment Diagram



Deployment Diagram 1

Chapter 5
Implementation

Chapter 5: Implementation

This chapter focuses on the practical realization of the web scraping platform based on the design specifications outlined in previous chapters. It details the actual coding and development process, including the selection of programming languages, frameworks, and tools. Implementation covers the creation of frontend interfaces, backend functionalities, and integration with external APIs for data scraping and processing. Additionally, this chapter addresses testing procedures, debugging, and optimization techniques to ensure the platform meets performance standards and functional requirements. Overall, it provides insights into turning design concepts into a functional and operational web scraping solution.

5.1. Important Flow Control/Pseudo codes

```
FUNCTION scrapeProductData(productName):
  DECLARE resultsList
  FOR each website in supportedWebsites:
    htmlContent = fetchWebsiteContent(website, productName)
    IF htmlContent is not empty:
      productInfo = extractProductInfo(htmlContent)
      resultsList.append(productInfo)
  RETURN resultsList
```

In this pseudocode snippet:

- **Function Name**: scrapeProductData
- **Purpose**: Scrapes product data from multiple e-commerce websites based on the provided productName.
- **Steps**:
    1. Initialize an empty resultsList to store scraped product information.
    2. Iterate through each website in a predefined list of supportedWebsites.
    3. Fetch the HTML content of the website using fetchWebsiteContent.

4. If htmlContent is successfully retrieved, extract productInfo using extractProductInfo and append it to resultsList.
5. Return resultsList containing aggregated product information from all supported websites.

5.2. Components, Libraries, Web Services and stubs

Components:

1. **Front-end**: made with react JS.
2. **Back-end:** made with node JS.
3. **APIs':** made with express JS.
4. **Database:** made with mongo db.
5. **Scraper:** made with python.

Libraries:

1. **React JS:** used for front end.
2. **React bootstrap:** used for animations.
3. **Axios:** used for API development.

Web Services:

1. **Gemini LLM API:** used to get suggestions.

5.3. Deployment Environment

The project is currently deployed on a local machine.

5.4. Tools and Techniques

This section delves into the specific tools and methodologies employed throughout the development of the web scraping platform.

In terms of programming languages, node was chosen for the backend development, offering robust support for data manipulation and integration with web scraping libraries such as BeautifulSoup and Scrapy. On the frontend, JavaScript with React provided a dynamic and responsive user interface, enhancing user experience and interactivity.

For integrated development environments (IDEs), PyCharm served as the primary tool for Python scraper development, offering powerful debugging and code analysis features. Visual Studio Code complemented this by providing a versatile environment for JavaScript development, facilitating efficient frontend implementation.

Version control was managed using Git, with GitHub serving as the repository for centralized code storage, version tracking, and collaborative development. This setup ensured that all team members could work concurrently on different aspects of the project while maintaining version integrity and facilitating code reviews.

Database management relied on mongo DB, chosen for its relational database capabilities, scalability, and robust data integrity features. It provided a stable foundation for storing and querying scraped data from various e-commerce websites.

Development practices followed an Agile methodology, specifically employing the Scrum framework. This iterative approach enabled the team to prioritize tasks, respond quickly to changes in requirements, and deliver incremental updates to stakeholders.

Throughout the development lifecycle, rigorous unit testing was conducted using Pytest for backend components and Jest for frontend components. This ensured that each module functioned as expected, detected bugs early in the development process, and maintained code quality.

Code reviews played a crucial role in maintaining code quality and knowledge sharing within the team. Regular peer reviews helped identify potential issues, validate implementation against requirements, and foster continuous improvement across the development cycle.

Overall, these tools and techniques collectively contributed to the successful implementation of the web scraping platform, aligning technical implementation with project objectives while ensuring scalability, reliability, and maintainability of the final product.

5.5. Best Practices / Coding Standards

In developing the web scraping platform, we prioritized adherence to clear coding standards and best practices:

Clear Coding Standards: Throughout the project, we followed PEP 8 guidelines for Python and ES6 standards for JavaScript. This ensured consistency, readability, and maintainability across the codebase.

Modular Design Approach: Our development approach focused on modular design principles. This facilitated code reusability and ease of maintenance by compartmentalizing functionalities into distinct, manageable modules.

Effective Version Control and Collaboration: We utilized Git for version control, employing feature branching and pull requests to manage code changes efficiently. Regular code reviews helped ensure quality and fostered knowledge sharing among team members.

Comprehensive Documentation: Thorough documentation was maintained, including inline comments, README files, and API documentation. This documentation supported seamless onboarding, troubleshooting, and understanding of the project's architecture and functionalities.

5.6. Version Control

Git was used for version control as it is a complete and comprehensive platform for centralized versioning control and allows easy access to all team members and supervisors.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

This chapter focuses on the rigorous testing phase of the developed website to ensure its functionality, reliability, and user-friendliness. It outlines the various testing methodologies employed, including unit testing, integration testing, and user acceptance testing. Through these tests, the chapter aims to evaluate the performance of the website, identifying any bugs, glitches, or inconsistencies.

6.1. Use Case Testing

1. **User Management:**

   Use Cases:

   - Register User: Testing the registration process to ensure users can successfully create an account.
   - Login: Verifying that users can log in securely with valid credentials.
   - Logout: Testing the logout functionality to ensure users can securely log out of their accounts.
   - Manage Profile: Testing the ability for users to update and manage their profile information.

2. **Product Exploration:**

   Use Cases:

   - Search Products: Testing the search functionality to ensure users can find products based on keywords or categories.
   - Compare Prices: Verifying that users can compare prices of products from different e-commerce websites.
   - View Product Details: Testing the functionality to view comprehensive details of a selected product.
   - Save Favorites: Ensuring users can save their favorite products for future reference.

3. **Data Maintenance:**

   Use Cases:

- Web Scraping for Price Updates: Testing the web scraping mechanism to fetch updated prices from e-commerce websites.
- Data Backup and Recovery: Verifying the backup and recovery process to ensure data integrity and reliability.

4. **User Interaction**:

   Use Cases:

   - Customize Dashboard: Testing the ability for users to customize their dashboard based on preferences.
   - Submit Feedback: Verifying that users can provide feedback on the website's performance and usability.
   - Receive Notifications: Testing the notification system to ensure users receive relevant updates and alerts.

5. **System Integration:**

   Use Cases:

   - Integrate with Payment Gateway: Testing the integration with a payment gateway to facilitate secure transactions.
   - Third-Party Analytics Integration: Verifying the integration with analytics tools to track website performance and user behavior.

6. **Error Handling:**

   Use Cases:

   - Log Errors: Testing the logging mechanism to capture and track errors for troubleshooting.
   - Resolve Issues: Verifying that the system effectively resolves errors and maintains smooth functionality.

7. **Administrative Functions:**

   Use Cases:

   - Manage Users: Testing administrative functions to manage user accounts and permissions.

- Monitor System Performance: Verifying the system's ability to monitor performance metrics and resource usage.

8. **Data Management:**

   Use Cases:

   - Export Data: Testing the functionality to export data for analysis or backup purposes.
   - Import Data: Verifying that users can import data into the system accurately.

6.2. Equivalence partitioning

Equivalence partitioning was a crucial technique employed during the testing phase of this project to ensure thorough coverage of various input scenarios across the website's functionalities.

For instance, in the user management module, the input fields for user registration and profile management were subjected to equivalence partitioning. For the "Register User" use case, the input fields such as username, email, and password were divided into equivalence classes based on their validation criteria. Valid equivalence classes included inputs that satisfied the required format and length constraints, while invalid equivalence classes encompassed inputs that failed to meet these criteria, such as empty fields, invalid email formats, or passwords below the minimum length requirement. Test cases were then designed to cover a representative value from each equivalence class to ensure comprehensive validation testing.

Similarly, in the product exploration module, equivalence partitioning was applied to inputs such as search queries and product comparison parameters. Valid equivalence classes included search terms that matched existing product listings, while invalid equivalence classes comprised search terms that did not yield any results or contained special characters or invalid syntax. Test cases were strategically crafted to cover a range of valid and invalid inputs, ensuring that the search functionality behaved consistently and accurately across different scenarios.

By applying equivalence partitioning, the testing team was able to optimize test coverage while minimizing redundancy, leading to a more efficient and effective testing process. This approach helped in identifying and addressing potential issues related to input validation, data processing, and system behavior, ultimately contributing to the overall reliability and functionality of the website.

6.3. Boundary value analysis

Boundary value analysis played a significant role in the testing strategy for this project, particularly in ensuring the robustness and accuracy of input validation mechanisms across the website's functionalities.

For instance, in the user management module, boundary value analysis was applied to input fields such as password length and username character limits. Test cases were designed to include values at the lower and upper boundaries of the acceptable range, as well as values just beyond these boundaries, to assess how the system handled edge cases. By testing inputs such as passwords with the minimum and maximum allowed length, as well as lengths one character below and above these limits, the testing team could verify that the system correctly enforced validation rules and provided appropriate feedback to users.

Similarly, in the product exploration module, boundary value analysis was employed to test inputs related to price ranges and product identifiers. Test cases were crafted to include prices at the lower and upper bounds of the acceptable range, as well as prices just below and above these boundaries, to ensure accurate filtering and sorting of search results. Additionally, boundary values for product identifiers were tested to verify that the system correctly handled inputs at the limits of alphanumeric character lengths and formats.

By conducting boundary value analysis, the testing team could identify potential vulnerabilities or inconsistencies in input validation mechanisms and ensure that the website performed reliably across a wide range of scenarios. This approach helped in uncovering and addressing issues related to boundary conditions, data processing errors, and system behavior, ultimately contributing to the overall quality and usability of the website.

6.4. Data flow testing

Data flow testing was a fundamental aspect of the testing strategy for this project, aimed at ensuring the integrity and accuracy of data as it moved through the various components of the website.

In the context of user management, data flow testing focused on examining how user information flowed through the registration, authentication, and profile management processes. Test cases were designed to verify that user data entered during registration was accurately

stored in the database, securely transmitted during login, and appropriately updated during profile management operations. By tracing the flow of user data through these processes, the testing team could identify any potential data loss, corruption, or unauthorized access points, ensuring the confidentiality and reliability of user information.

Similarly, in the product exploration module, data flow testing was essential for assessing how product data was retrieved from e-commerce websites, processed, and displayed to users. Test cases were crafted to validate the accuracy of web scraping mechanisms in extracting product information, the efficiency of data processing algorithms in aggregating and organizing this information, and the consistency of data presentation across different parts of the website. By systematically tracing the flow of product data from its source to its final presentation to users, the testing team could ensure the reliability, completeness, and timeliness of product information.

Overall, data flow testing played a crucial role in validating the end-to-end functionality and performance of the website, helping to identify and mitigate potential data-related issues that could impact user experience or compromise system security. This approach contributed to the overall robustness and reliability of the website, ensuring that users could interact with accurate and up-to-date information in a secure and efficient manner.

6.5. Unit testing

Unit testing was a foundational aspect of the testing process for this project, focusing on validating the functionality and behavior of individual components or units of code within the website's system.

In the user management module, unit testing involved testing each function responsible for user registration, authentication, profile management, and session handling. Test cases were designed to verify that these functions correctly handled various scenarios, such as valid and invalid input data, edge cases, and error conditions. For example, unit tests were conducted to ensure that the user registration function correctly validated input data, stored user information securely in the database, and generated appropriate error messages when necessary. Similarly, unit tests were performed to verify that the authentication function correctly validated user credentials, authenticated users securely, and managed user sessions effectively.

In the product exploration module, unit testing focused on testing functions responsible for searching, filtering, and displaying product information to users. Test cases were designed to validate the functionality of these functions in retrieving product data from external sources, processing and organizing this data, and presenting it to users in a user-friendly and accurate manner. For example, unit tests were conducted to ensure that the search function correctly retrieved relevant product listings based on user queries, applied filters accurately, and sorted results appropriately. Additionally, unit tests were performed to verify that product details were displayed correctly and that users could navigate and interact with product information seamlessly.

6.6. Integration testing

In the user management module, integration testing involved testing the seamless integration of functions responsible for user registration, authentication, profile management, and session handling. Test cases were designed to verify the flow of data and control between these functions, ensuring that user information was correctly passed between modules and that changes made to user profiles were accurately reflected across the system. For example, integration tests were conducted to validate the end-to-end functionality of the user registration process, from data input validation to database storage and session initiation.

In the product exploration module, integration testing focused on testing the integration of functions responsible for retrieving, processing, and presenting product information to users. Test cases were designed to validate the flow of data between components such as the web scraping module, data processing algorithms, and user interface components. For example, integration tests were conducted to verify that product data retrieved from e-commerce websites was accurately processed and displayed to users in the search results and product detail pages

6.7. Performance testing

Load testing was conducted to assess how the website performed when subjected to different levels of user traffic and usage patterns. Test cases were designed to simulate realistic scenarios, such as multiple concurrent user sessions performing actions such as searching for products, comparing prices, and viewing product details. By gradually increasing the load on the website

and monitoring key performance metrics such as response times, throughput, and resource utilization, the testing team could identify potential bottlenecks, performance degradation, or system failures under heavy load conditions.

Stress testing was performed to determine the website's resilience and robustness under extreme load conditions. Test cases were designed to push the website beyond its normal operational limits by simulating a large number of concurrent user sessions or by overwhelming the system with excessive requests. This helped to identify the maximum capacity of the website and any potential points of failure, such as server crashes, database overload, or network congestion. By measuring the website's response to stress conditions and monitoring key performance indicators such as error rates and system stability, the testing team could assess its ability to gracefully handle unexpected spikes in traffic or usage.

6.8. Stress Testing

During stress testing, the website was subjected to high levels of traffic, excessive data loads, and concurrent user interactions to assess its resilience and ability to handle such scenarios without crashing or experiencing significant degradation in performance. Test cases were designed to simulate peak usage scenarios, such as sudden spikes in user traffic during peak hours or unexpected surges in user activity due to promotional events or marketing campaigns.

The stress testing process involved gradually increasing the load on the website until it reached its breaking point, pushing the system to its limits to identify potential bottlenecks, vulnerabilities, or points of failure. Key performance metrics, such as response times, error rates, and system resource utilization, were closely monitored throughout the testing process to gauge the website's performance under stress conditions.

By analyzing the results of stress testing, the testing team could identify any weaknesses or performance issues that needed to be addressed, such as inadequate server capacity, inefficient database queries, or network congestion. This allowed for proactive optimization and fine-tuning of the website's infrastructure, architecture, and codebase to enhance its resilience and scalability in anticipation of real-world usage scenarios

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

This project aimed to develop a comprehensive website that would revolutionize the ecommerce experience in Pakistan by providing users with a platform to compare prices and reviews of products across various online retailers. Leveraging a combination of web development technologies including Python, HTML, CSS, and JavaScript, the website was designed to offer a user-friendly interface allowing consumers to effortlessly search for products, compare prices, read reviews, and make informed purchasing decisions.

Throughout the development process, rigorous testing methodologies such as unit testing, integration testing, performance testing, and stress testing were employed to ensure the reliability, functionality, and scalability of the website. Equivalence partitioning and boundary value analysis were utilized to validate input data and ensure robust input validation mechanisms, while data flow testing was conducted to verify the accuracy and integrity of data as it traversed through the system.

The successful completion of this project represents a significant milestone in the realm of ecommerce in Pakistan, providing consumers with a valuable resource to streamline their online shopping experience and empowering businesses to remain competitive in a dynamic marketplace. With its intuitive interface, comprehensive features, and robust testing, the website is poised to make a significant impact on the e-commerce landscape in Pakistan, offering users unparalleled convenience and access to the best deals available.

7.2. Achievements and Improvements

Achievements:

- Successful development of a user-friendly website providing a revolutionary platform for e-commerce in Pakistan.
- Implementation of comprehensive functionalities allowing users to compare prices, read reviews, and make informed purchasing decisions.

- Utilization of a diverse range of web development technologies including Python, HTML, CSS, and JavaScript to create a robust and dynamic website.
- Rigorous testing methodologies such as unit testing, integration testing, and performance testing ensured the reliability, functionality, and scalability of the website.
- Incorporation of advanced testing techniques such as equivalence partitioning, boundary value analysis, and data flow testing to validate input data, ensure robust input validation mechanisms, and verify the accuracy of data processing.

Improvements:

- Enhance user interface design to further improve user experience and visual appeal.
- Implement additional features such as personalized recommendations, advanced search filters, and real-time notifications to enhance user engagement and satisfaction.
- Optimize website performance and scalability to handle increasing user traffic and accommodate future growth.
- Strengthen security measures to safeguard user data and protect against potential threats such as data breaches or cyberattacks.
- Conduct continuous monitoring and updates to address any emerging issues, improve functionality, and adapt to evolving user needs and industry trends

7.3. Critical Review

The project to develop a comprehensive e-commerce comparison website for the Pakistani market represents a commendable effort towards enhancing consumer choice and facilitating informed purchasing decisions. The incorporation of a variety of web development technologies and rigorous testing methodologies demonstrates a commitment to delivering a reliable and functional platform.

However, a critical review reveals several areas for improvement. Firstly, while the website offers valuable functionalities such as price comparison and review aggregation, the user interface could benefit from further refinement to enhance usability and visual appeal. Streamlining the

navigation and improving the layout can contribute to a more intuitive and enjoyable user experience.

Additionally, while the testing methodologies employed were comprehensive, there may be room to explore additional techniques or refine existing ones to ensure thorough validation of the website's functionalities. Continuous monitoring and updates are essential to address any emerging issues and optimize performance, particularly as user needs and industry trends evolve over time.

Furthermore, the project's success hinges on effective marketing and user adoption. Developing strategies to promote the website and attract a significant user base will be crucial for its long-term viability and impact on the Pakistani e-commerce landscape.

7.4. Lessons Learnt

User-Centric Design: Prioritizing user experience and designing an intuitive interface can significantly enhance adoption and satisfaction among users. Regular feedback gathering and usability testing can provide valuable insights for iterative improvements.

Testing Depth: While comprehensive testing methodologies were employed, the depth of testing could be further enhanced by exploring additional techniques and refining existing ones. Continuous monitoring and updates are essential to address emerging issues and optimize performance.

Agile Development: Embracing an agile development approach allows for flexibility and adaptability in responding to changing requirements and user feedback. Iterative development cycles enable incremental improvements and ensure alignment with stakeholder expectations.

Market Research: Conducting thorough market research and understanding user needs and preferences are crucial for developing a successful product. Regular market analysis and competitor benchmarking can inform strategic decision-making and identify opportunities for differentiation.

Scalability and Security: Prioritizing scalability and security from the outset is essential for accommodating future growth and protecting user data. Implementing robust infrastructure and security measures can prevent potential vulnerabilities and ensure a reliable and secure platform.

Effective Communication: Clear and transparent communication among team members, stakeholders, and users is essential for project success. Regular updates, feedback sessions, and collaborative decision-making foster a culture of accountability and alignment towards shared goals.

Continuous Improvement: Embracing a culture of continuous improvement and learning ensures ongoing relevance and competitiveness in a dynamic market landscape. Regular evaluation, iteration, and adaptation are key to staying ahead of evolving user needs and industry trends.

7.5. Future Enhancements/Recommendations

Personalization: Implement personalized recommendation features based on user preferences and browsing history to enhance user engagement and satisfaction.

Advanced Search Filters: Introduce advanced search filters to allow users to refine their product searches based on specific criteria such as brand, price range, and availability.

Real-Time Notifications: Enable real-time notifications for users to receive updates on price drops, new product arrivals, and special promotions to encourage timely purchases.

Mobile Optimization: Optimize the website for mobile devices to cater to the increasing number of users accessing e-commerce platforms through smartphones and tablets.

Social Integration: Integrate social media sharing and login functionalities to facilitate social engagement and broaden the website's reach through viral marketing.

Appendices

Appendix A: User Manual

This appendix provides detailed instructions on how to use the web scraping platform. It includes step-by-step guides for all major functionalities, helping users navigate the platform, perform searches, and interpret the results. The manual aims to ensure that users can fully utilize the features of the platform with ease and efficiency.

A.1. Getting Started

This section introduces users to the basic functionalities of the platform, including how to create an account, log in, and navigate through the main interface. It also covers system requirements and browser compatibility to ensure optimal performance.

A.1.1. Account Creation

To begin using the platform, users must first create an account. This involves providing basic information such as name, email, and password. An email verification step is included to ensure the authenticity of the user.

A.1.2. Logging In

After creating an account, users can log in using their email and password. The login page also provides options for password recovery in case users forget their credentials.

A.2. Searching for Products

This section guides users through the process of searching for products. It covers how to enter product names into the search bar, refine search results using filters, and interpret the displayed product information.

A.2.1. Entering Search Queries

Users can type the name of the desired product into the search bar. The platform will automatically suggest similar products to help users find the exact match they are looking for.

A.3. Viewing and Comparing Products

Once search results are displayed, users can click on individual products to view detailed information. This section explains how to compare products side by side based on price, features, and user reviews.

A.4. Account Settings and Preferences

This section explains how users can manage their account settings, including updating personal information, changing passwords, and setting notification preferences.

Appendix B: Administrator Manual

This appendix provides guidelines for administrators to manage and maintain the web scraping platform, including user management, system monitoring, data updates, troubleshooting, and security.

B.1. User Management

Administrators create, update, and delete user accounts while managing permissions to ensure secure access.

B.1.1. Creating and Managing User Accounts

Administrators handle user information and roles, ensuring appropriate access levels.

B.1.2. Deleting User Accounts

Confirm deletions to prevent accidental removal of active accounts.

B.2. Troubleshooting and Support

Address technical issues and provide user support.

B.2.1. Diagnosing Technical Issues

Review system logs and user reports to identify problems.

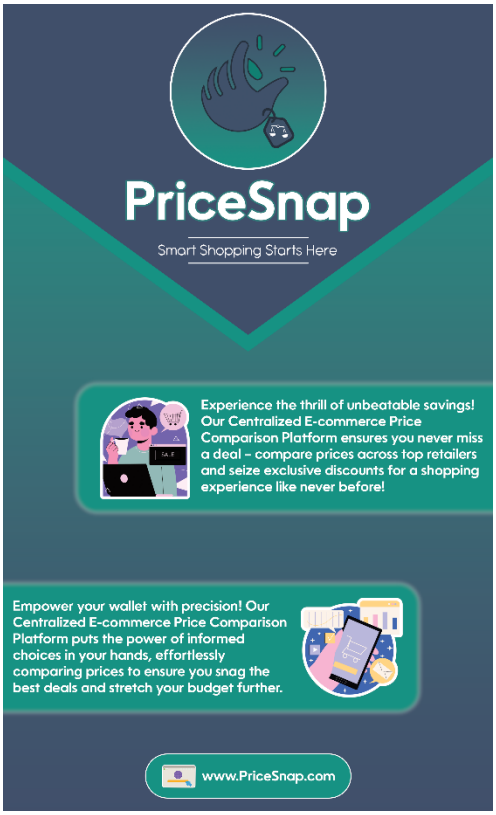
B.2.2. Providing User Support

Guide users through troubleshooting steps and resolve issues.

Appendix C: Information / Promotional Material

In this section all of the promotional material that is created for this project is included in the following

C.1. Broacher

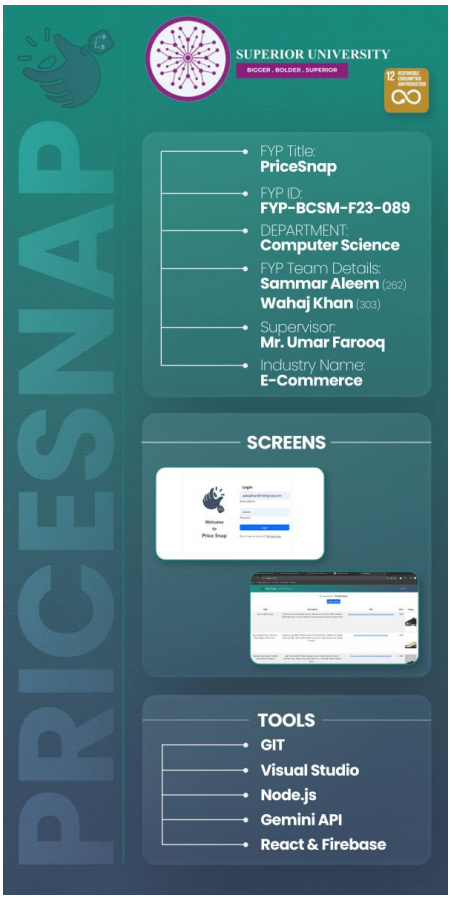


Brochure 1

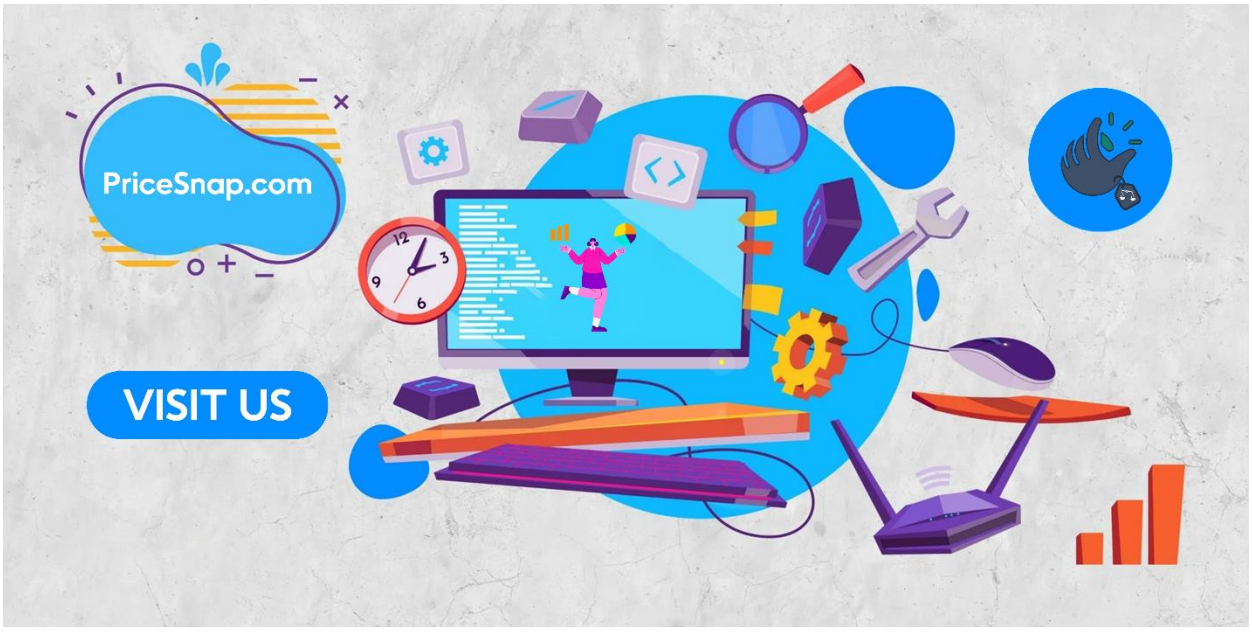
C.2. Flyer


PriceSnap
COMPARE
CLICK
CONQUER
SAVINGS
ABOUT US
At PriceSnap, we revolutionize online shopping in Pakistan. Our Centralized E-commerce Price Comparison Platform empowers users with real-time data, ensuring informed decisions and maximizing savings effortlessly.
WHAT WE OFFER
REAL-TIME PRICE COMPARISON:
Access live updates on product prices across diverse Pakistani e-commerce platforms, ensuring you secure the best deals instantly
COMPREHENSIVE PRODUCT COVERAGE:
Enjoy a one-stop solution for comparing prices and reviews across a wide range of products, simplifying your shopping decisions.
USER-FRIENDLY INTERFACE:
Experience seamless navigation with our intuitive platform, designed for effortless price comparisons, empowering you to make informed purchasing choices.
www.PriceSnap.com

C.3. Standee


SUPERIOR UNIVERSITY
BIGGER , BOLDER , SUPERIOR
PRICESNAP
FYP Title:
PriceSnap
FYP ID:
FYP-BCSM-F23-089
DEPARTMENT:
Computer Science
FYP Team Details:
Sammar Aleem (262)
Wahaj Khan (303)
Supervisor:
Mr. Umar Farooq
Industry Name:
E-Commerce
SCREENS
TOOLS
GIT
Visual Studio
Node.js
Gemini API
React & Firebase

C.4. Banner



turnitin

Submission ID trn:oid:::1:2956781202

How much of this submission has been generated by AI?

***4%**

of qualifying text in this submission has been determined to be generated by AI.

* Low scores have a higher likelihood of false positives.

Caution: Percentage may not indicate academic misconduct. Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

What does the precentage mean?
The precentage show in the AI writing detection indicator and in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was generated by AI.

Our testing has found that there is a higher incidence of false positives when the precentage is less than 20. In order to reduce the likelihood of misinterpretation, the AI indicator will display an asterik for precentages less than 20 to call attention to the fact that the score is less reliable.

However, the final decision on whether any misconduct has occurred rests with the reviewer/instructor. They should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in greater detail according to ther school's polices.

How does Turnitin's indicator address false positives?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be AI-generated will be highlighted blue on the submission text.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

What does'qualifying text' mean?
Sometimes false positives (incorrectly flagging human-written text as AI-generated), can include lists without a lot of structural variation, text that literally repeats itself, or text that has been paraphrased without developing new ideas. If our indicator shows a higher amount of AI writing in such text, we advise you to take that into consideration when looking at the percentage indicated.

In a longer document with a mix of aithentic writing and AI generated text, it can be difficult to exactly determind where the AI writing begins and original writing ends, but our model should give you a reliable guide to start conversations with the submitting student.

turnitin

Submission ID trn:oid:::1:2956781202

PriceSnap

ORIGINALITY REPORT

8%	7%	1%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	6%
2	projectsgeek.com Internet Source	1%
3	www.coursehero.com Internet Source	1%
4	pdfcoffee.com Internet Source	1%
5	Submitted to Colorado Technical University Student Paper	<1%
6	de.slideshare.net Internet Source	<1%
7	Submitted to University of Wales Swansea Student Paper	<1%
8	Submitted to Signet Institute Student Paper	<1%
9	Submitted to University of Wales Institute, Cardiff	<1%

	Student Paper	
10	www.slideshare.net Internet Source	<1%
11	Submitted to University at Buffalo Student Paper	<1%
12	hbtu.ac.in Internet Source	<1%
13	Raaghav Ramamoorthy. "Exploring Bias in the FOLD-R++ Algorithm: A Comprehensive Analysis", Institute of Electrical and Electronics Engineers (IEEE), 2023 Publication	<1%
14	www.annalsofrscb.ro Internet Source	<1%
15	Hengshu Zhu, Hui Xiong, Yong Ge, Enhong Chen. "Discovery of Ranking Fraud for Mobile Apps", IEEE Transactions on Knowledge and Data Engineering, 2015 Publication	<1%