



Tribhuvan University

Faculty of Humanities and Social Science

## **DigiMart: Smart Platform For Electronic Products**

A PROJECT REPORT

Submitted to  
Department of Computer Application  
D.A.V. College

In partial fulfillment of the requirements for the Bachelors of Computer Application

Submitted by  
Himal Subedi  
[6-2-469-161-2020]  
November, 2025

Under the Supervision of  
Er. Ishan Pokhrel



**Tribhuvan University**  
**Faculty of Humanities and Social Sciences**  
**D.A.V. College**

**Supervisor's Recommendation**

I hereby recommend that this project be prepared under my supervision by **HIMAL SUBEDI** entitled “**DIGIMART: SMART PLATFORM FOR ELECTRONIC PRODUCTS**” in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

.....

Er. Ishan Pokhrel

**SUPERVISOR**

Lecturer

Faculty of Humanities and Social Sciences

D.A.V. College



**Tribhuvan University**  
**Faculty of Humanities and Social Sciences**  
**D.A.V. College**

**LETTER OF APPROVAL**

This is to certify that this project prepared by **HIMAL SUBEDI** entitled “**DIGIMART: SMART PLATFORM FOR ELECTRONIC PRODUCTS**” in partial fulfillment of the requirements for the degree of Bachelor of Computer Application has been evaluated. In our opinion, it is satisfactory in the scope and quality of a project for the required degree.

|                                                                                                  |                                                                                                |
|--------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <hr/> <p>Er. Ishan Pokhrel</p> <p>Lecturer</p> <p>D.A.V. College</p> <p>Jawalakhel, Lalitpur</p> | <hr/> <p>Mr. Sudip Adhikari</p> <p>H.O.D</p> <p>D.A.V. College</p> <p>Jawalakhel, Lalitpur</p> |
| <hr/> <p><b>Internal Examiner</b></p>                                                            | <hr/> <p><b>External Examiner</b></p>                                                          |

## ACKNOWLEDGMENT

I express my deepest gratitude to my supervisor, **Er. Ishan Pokhrel**, for his constant support and expert guidance throughout this project. His valuable advice and encouragement helped me overcome challenges and stay motivated. I truly appreciate the time and effort he invested in providing feedback and helping me improve the quality of this work.

I would also like to thank **Sudip Adhikari**, the Head of Department, for his continuous support and for creating a positive environment that allowed me to focus on my project. I am also grateful to our Coordinator, **Shashi Bhushan Chaturvedi**, whose encouragement and thoughtful suggestions made this journey smoother and more enjoyable. Their combined efforts played a big role in shaping the outcome of my project.

Lastly, I would like to extend my heartfelt thanks to **D.A.V. College** for providing me with the resources and opportunity to work on this project. I am also grateful to my family and friends for their constant support and encouragement and to everyone who directly or indirectly helped me complete this project successfully.

**With Regards**

Himal Subedi

TU Reg No: [6-2-469-161-2020]

# ABSTRACT

This project focuses on developing a comprehensive intelligent system for DigiMart that aims to improve both good user experience and operational efficiency. The system integrates a Content-Based Recommendation System and a Time Series Demand Forecasting using ARIMA Model to provide personalized product suggestions and accurate sales predictions. The recommendation system uses product attributes such as category, brand, price, and textual descriptions, combined with user interaction data including past purchases, viewed products, and cart behavior, to generate tailored recommendations that reflect individual preferences.

Simultaneously, the forecasting component employs historical sales data to identify trends, enabling more informed decisions. The implementation emphasizes modular and maintainable code structures, designed to ensure scalability and facilitate future enhancements. Extensive testing and validation are performed to assess recommendation relevance and to confirm that forecasting outputs correspond closely with actual sales patterns. Overall, this project provides a practical framework for leveraging data-driven techniques in DigiMart, aiming to enhance customer satisfaction, optimize operations, and support strategic business decision-making.

**Keywords:** *DigiMart, Content-Based Filtering, Product Recommendation, ARIMA, Time Series Forecasting, Demand Prediction*

## Table of Contents

|                                                                     |     |
|---------------------------------------------------------------------|-----|
| ACKNOWLEDGMENT . . . . .                                            | iii |
| ABSTRACT . . . . .                                                  | iv  |
| LIST OF FIGURES . . . . .                                           | vi  |
| LIST OF TABLES . . . . .                                            | vii |
| LIST OF ABBREVIATIONS . . . . .                                     | ix  |
| CHAPTER 1: INTRODUCTION . . . . .                                   | 1   |
| 1.1 Introduction . . . . .                                          | 1   |
| 1.2 Problem Statement . . . . .                                     | 1   |
| 1.3 Objectives . . . . .                                            | 1   |
| 1.4 Scope and Limitations . . . . .                                 | 2   |
| 1.5 Development Methodology . . . . .                               | 2   |
| 1.6 Report Organization . . . . .                                   | 3   |
| CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW . . . . .         | 5   |
| 2.1 Background Study . . . . .                                      | 5   |
| 2.2 Literature Review . . . . .                                     | 5   |
| CHAPTER 3: SYSTEM ANALYSIS AND DESIGN . . . . .                     | 7   |
| 3.1 System Analysis . . . . .                                       | 7   |
| 3.1.1 Requirement Analysis . . . . .                                | 7   |
| 3.2 Feasibility Analysis . . . . .                                  | 8   |
| 3.3 Object Modelling Using Class Diagram . . . . .                  | 11  |
| 3.3.1 Dynamic Modelling using State and Sequence Diagrams . . . . . | 12  |
| 3.3.2 Process Modeling using Activity Diagram . . . . .             | 13  |
| 3.3.3 Database Schema Diagram . . . . .                             | 14  |
| 3.3.4 Algorithm Details . . . . .                                   | 15  |

|                                                           |    |
|-----------------------------------------------------------|----|
| CHAPTER 4: IMPLEMENTATION AND TESTING . . . . .           | 21 |
| 4.1 Implementation . . . . .                              | 21 |
| 4.1.1 Tools Used . . . . .                                | 21 |
| 4.1.2 Implementation Details of Modules . . . . .         | 21 |
| 4.2 Testing . . . . .                                     | 24 |
| 4.2.1 Test Cases for Unit Testing . . . . .               | 24 |
| 4.2.2 Test Cases for System Testing . . . . .             | 27 |
| 4.3 Result Analysis . . . . .                             | 27 |
| CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION . . . . . | 30 |
| 5.1 Conclusion . . . . .                                  | 30 |
| 5.2 Future Recommendations . . . . .                      | 30 |
| REFERENCES . . . . .                                      | 30 |

## LIST OF FIGURES

|                                                              |    |
|--------------------------------------------------------------|----|
| Figure 1.1: Iterative WaterFall Model . . . . .              | 3  |
| Figure 3.1: Use Case Diagram of DigiMart . . . . .           | 8  |
| Figure 3.2: Gantt Chart . . . . .                            | 10 |
| Figure 3.3: Class Diagram of DigiMart . . . . .              | 11 |
| Figure 3.4: Sequential Diagram of DigiMart . . . . .         | 12 |
| Figure 3.5: Activity Diagram of DigiMart . . . . .           | 13 |
| Figure 3.6: Database Schema Diagram of DigiMart . . . . .    | 14 |
| Figure 3.7: Content Based Recommendation Flowchart . . . . . | 17 |
| Figure 3.8: ARIMA Based Forecasting Flowchart . . . . .      | 20 |

## LIST OF TABLES

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| Table 4.1: Login and Registration Testing . . . . .                           | 24 |
| Table 4.2: Admin Modules Testing . . . . .                                    | 24 |
| Table 4.3: User Testing . . . . .                                             | 25 |
| Table 4.4: Forecasting Module Testing . . . . .                               | 25 |
| Table 4.5: Recommendation Testing . . . . .                                   | 26 |
| Table 4.6: System Integration Testing . . . . .                               | 27 |
| Table 4.7: ARIMA Model Evaluation Summary . . . . .                           | 28 |
| Table 4.8: Home Page Recommendation Results . . . . .                         | 28 |
| Table 4.9: Product Page Recommendation Results (Context: iPhone 13) . . . . . | 29 |

## **LIST OF ABBREVIATIONS**

|         |                                          |
|---------|------------------------------------------|
| AI      | Artificial Intelligence                  |
| API     | Application Programming Interface        |
| ARIMA   | Autoregressive Integrated Moving Average |
| GUI     | Graphical User Interface                 |
| ML      | Machine Learning                         |
| NPM     | Node Package Manager                     |
| VS Code | Visual Studio Code                       |

# **CHAPTER 1: INTRODUCTION**

## **1.1 Introduction**

DigiMart is a dynamic web-based application integrated with machine learning features. It includes a content based recommendation system that analyzes product attributes to suggest items similar to those a user has viewed, ordered, or added to their cart. In addition, a time series forecasting module powered by the ARIMA model predicts future product demand, enabling better inventory planning and supply chain optimization. The platform offers an intuitive user interface where customers can browse products and receive personalized recommendations in real time. On the administrative side, the system provides tools to manage products, customers, and orders, as well as to view demand forecasts. These insights enable administrators to make data driven decisions regarding stock levels, product offerings, and overall business strategy. By combining intelligent recommendations with accurate ARIMA based demand forecasting and user friendly admin controls, DigiMart improves both the shopping experience and operational efficiency.

## **1.2 Problem Statement**

The DigiMart web application is designed to enhance the online shopping experience by integrating machine learning-based intelligence into a standard digital retail platform. It features a content based recommendation system that suggests products similar to those a user has viewed or purchased, and an ARIMA based forecasting module that predicts future sales trends for better inventory and business planning. The system provides an intuitive interface for customers to explore products and receive personalized recommendations, while administrators can manage products, orders, and view data driven forecasts. DigiMart serves as both a practical retail solution and an educational demonstration of how recommendation and forecasting algorithms can be applied in realworld online shopping scenarios.

## **1.3 Objectives**

The main objective of this project is:

- To develop DigiMart Web App for buying electronics, with admin managing products, orders, stock, and users.

- To implement a content based recommendation system and an ARIMA-based demand forecasting model.

## **1.4 Scope and Limitations**

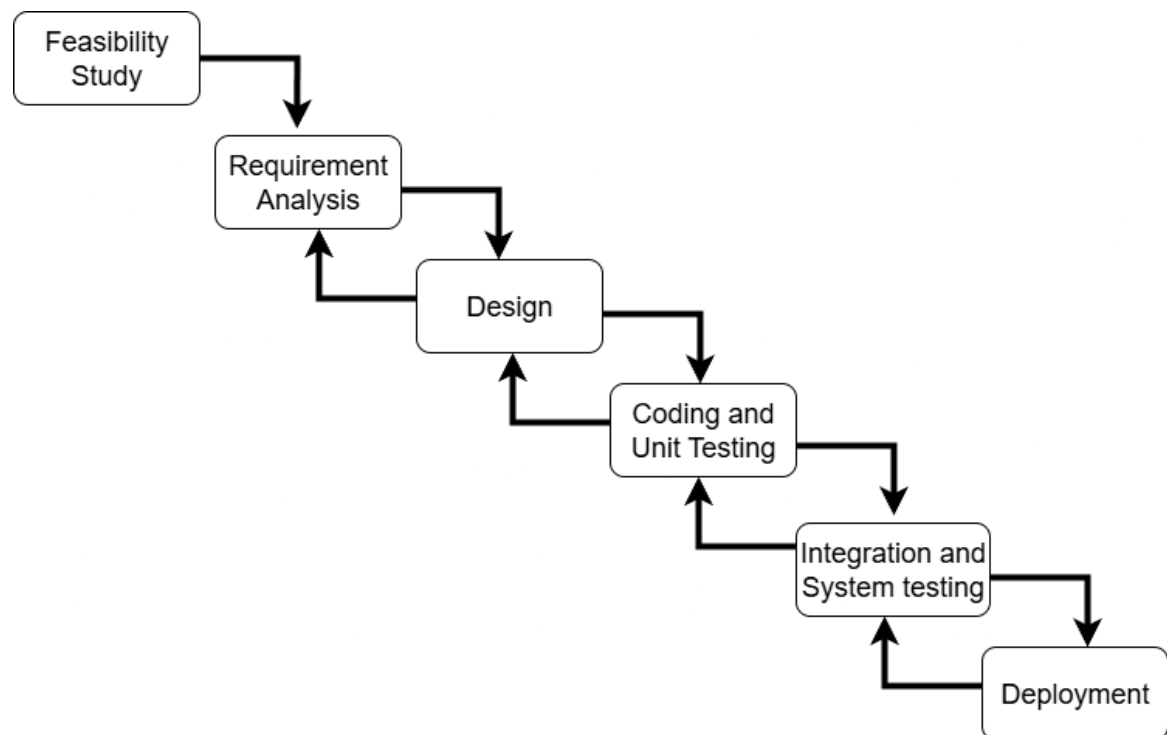
The DigiMart web application is designed to enhance the online shopping experience by integrating machine learning based intelligence into a digital retail platform. It features a content based recommendation system that suggests products similar to those a user has viewed or purchased, and an ARIMA based forecasting module that predicts future sales trends for improved inventory and business planning. The system provides an intuitive interface for customers to explore products and receive personalized recommendations, while administrators can manage products, orders, and access data driven forecasts. DigiMart serves as both a practical retail solution and an educational demonstration of how recommendation and forecasting algorithms can be applied in real world online shopping scenarios.

### **Limitations**

- Forecasting accuracy depends on the quantity and quality of available sales data.
- Recommendations may be limited to product similarity and not yet optimized for complex behavioral personalization.

## **1.5 Development Methodology**

The Iterative Waterfall model is chosen for its structured and phased development approach, which aligns well with the systematic planning and execution required for the DigiMart platform. Each phase requirements gathering, design, implementation, testing, and deployment is carried out in a defined sequence, with the flexibility to revisit earlier stages based on feedback. This approach ensures thorough documentation, clearly defined milestones, and the gradual integration of machine learning components. As a result, it becomes easier to track progress, maintain quality standards, and deliver a reliable, well tested product.



**Figure 1.1: Iterative WaterFall Model**

## **1.6 Report Organization**

- **Chapter 1: Introduction:** Provides a comprehensive overview of the project, including the background and motivation for developing the DigiMart system, the problem statement, and clearly defined objectives. It also outlines the methodology adopted. This chapter sets the foundation for understanding the context and goals of the system.
- **Chapter 2: Background Study and Literature Review:** Presents an in depth study of existing recommendation systems, demand forecasting techniques, and relevant technologies. This includes a review of classical approaches such as content based filtering and ARIMA based forecasting, along with modern advancements in personalized recommendations and time series analysis. Comparisons of various methods from previous research are discussed, highlighting their strengths and limitations to justify the chosen approach for this project.
- **Chapter 3: System Analysis and Design:** Details the functional and non-functional requirements of the DigiMart system, supported by requirement analysis diagrams such as use case diagrams, flowcharts, and sequence diagrams. The chapter also includes feasibility studies, architectural design, database schema, and algorithmic

design, providing a blueprint for the development process.

- **Chapter 4: System Implementation And Testing:** Describes the actual development process of the DigiMart system. It specifies the tools, programming languages, libraries, and frameworks used. The chapter explains how the content based recommendation and ARIMA forecasting algorithms are implemented, how the user interface is developed, and how all components are integrated to form the complete system. It also outlines the testing procedures used to verify the accuracy and performance of the DigiMart system. Unit, integration, and system testing were conducted to ensure functional reliability.
- **Chapter 5: Conclusion and Future Work:** Summarizes the overall achievements of the project and reflects on its contributions. Limitations encountered during development are discussed, and recommendations for potential future improvements.

## **CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW**

### **2.1 Background Study**

To develop an effective and user friendly DigiMart platform, it is essential to address the needs of both customers and business administrators. Understanding user behavior, shopping trends, and the shortcomings of traditional recommendation and sales forecasting systems is crucial for designing intelligent, machine learning based features.

Modern users expect personalized experiences, real time recommendations, and seamless navigation, while businesses seek reliable tools that enable accurate sales forecasting and informed decision making. However, traditional online shopping systems often fail to provide relevant product suggestions or precise demand predictions. This can result in issues such as product overstocking, stockouts, or lost sales opportunities.

The background study highlights the growing demand for advanced, data-driven digital retail solutions. By integrating machine learning methods such as a content based recommendation system for dynamic personalized product suggestions and the ARIMA time series model for sales forecasting, the DigiMart platform effectively addresses these gaps. This integration enhances customer satisfaction, provides smarter business insights, and creates a more efficient and engaging shopping experience for both users and administrators.

### **2.2 Literature Review**

Hukut Store, an emerging online retail platform in Nepal, focuses on selling electronic gadgets such as mobile phones, laptops, and smart accessories. The platform aims to provide a seamless shopping experience with features like nationwide delivery, competitive pricing, and a user friendly interface [1]. The study highlights the growing importance of online retail in Nepal and demonstrates that platforms like Hukut Store can effectively cater to tech savvy consumers by combining digital convenience with occasional physical store presence. The authors emphasize that such platforms leverage technology to enhance customer engagement, streamline inventory management, and provide a reliable marketplace for electronic products.

Recommendation as Classification: Using Social and Content Based Information in Recommendation [2] present an inductive learning approach to recommendation that integrates both social/collaborative information and content based features to predict user preferences. Unlike traditional social filtering methods that rely solely on ratings from similar users, their hybrid system incorporates descriptive information about each item, such as genres and attributes, and frames recommendation as a classification problem—predicting whether a user will like or dislike an item. Experiments on a dataset of over 45,000 movie ratings demonstrated that combining collaborative and content features significantly improves precision while maintaining recall compared to purely collaborative approaches. This study highlights the effectiveness of leveraging multiple sources of information and provides insights into designing recommendation systems that focus on both accuracy and relevance in presenting items to users.

The study on Foreva, a Portuguese footwear retailer, presents a comparative analysis of state space models and ARIMA models for forecasting sales across five categories of women's footwear: Boots, Booties, Flats, Sandals, and Shoes [3]. The authors highlight the challenges posed by retail sales time series, which typically exhibit trend and seasonal patterns, and demonstrate that both ETS (state space) and ARIMA models achieve comparable forecasting accuracy in terms of RMSE, MAE, and MAPE for one step and multi step forecasts. Their findings emphasize the importance of selecting models based on both in sample fit and out of sample performance, and they show that minor improvements in prediction accuracy can be obtained for specific product categories by choosing the appropriate model. This study underscores the value of systematically evaluating multiple forecasting methodologies and motivates the use of hybrid or adaptive approaches in retail demand forecasting, suggesting strategies that could be adapted to enhance prediction performance in similar multi category online or physical retail platforms.

## CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

### 3.1 System Analysis

System analysis involves a structured examination of the DigiMart application to evaluate its architecture, core functionalities, and performance attributes with the aim of ensuring efficiency, usability, and scalability. This process identifies both the operational capabilities and the quality constraints that define the system, enabling its alignment with the intended objectives and user expectations.

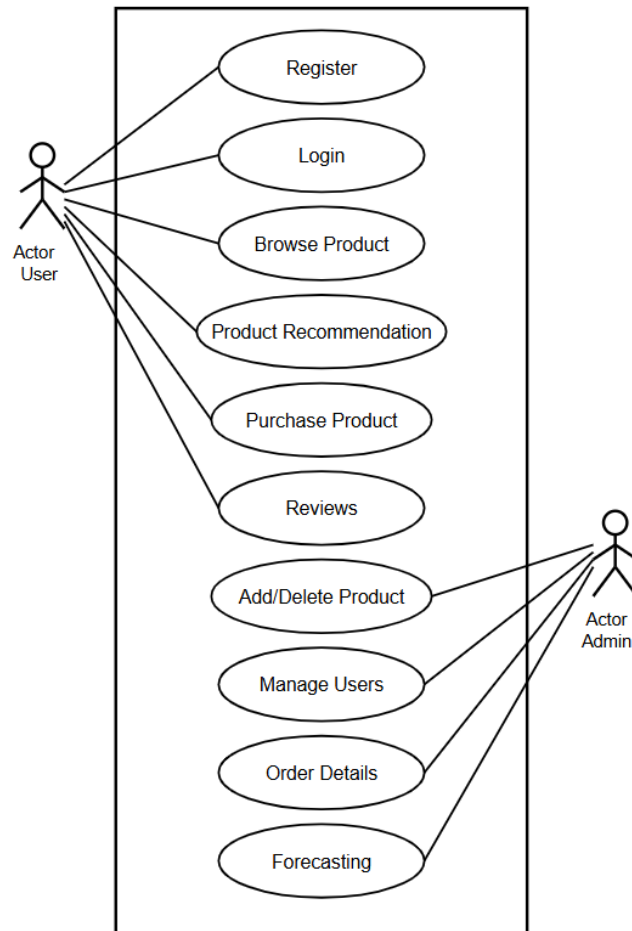
#### 3.1.1 Requirement Analysis

For this DigiMart application, the requirements are classified as follows:

##### i. Functional Requirements:

Functional requirements describe the essential capabilities and services that the system must provide to deliver its intended functionality. The key functional requirements include:

- **Admin Login:** To validate user and to track their activities conveniently.
- **Product Recommendation:** To provide realtime product suggestions using a content based recommendation system, based on the user's browsing history, past orders, and cart contents.
- **Sales Prediction:** To forecast future product sales using the ARIMA timeseries model, enabling better business decisions and planning.
- **Product, Order and User Management (Admin):** Allow administrators to add, update, and remove products, customers as well as view and manage orders.



**Figure 3.1: Use Case Diagram of DigiMart**

## ii. Non-Functional Requirements:

Non-functional requirements define the quality standards, performance benchmarks, and operational constraints of the system. These include:

- **Usability:** The system should have a user friendly interface that is intuitive and easy to navigate, allowing users to interact with the application conveniently.
- **Reliability:** The system should always be reliable and available for use, minimizing any downtime or disruptions in service.
- **Performance:** The system should be responsive and load pages in a reasonable time frame.

## 3.2 Feasibility Analysis

### 1. Technical Feasibility

The DigiMart platform is technically feasible as it uses reliable technologies for both

frontend and backend operations. React powers the dynamic user interface, while Node.js handles recommendation logic on the server side. Forecasting is implemented in Python, ensuring accurate sales predictions and personalized recommendations.

## **2. Operational Feasibility**

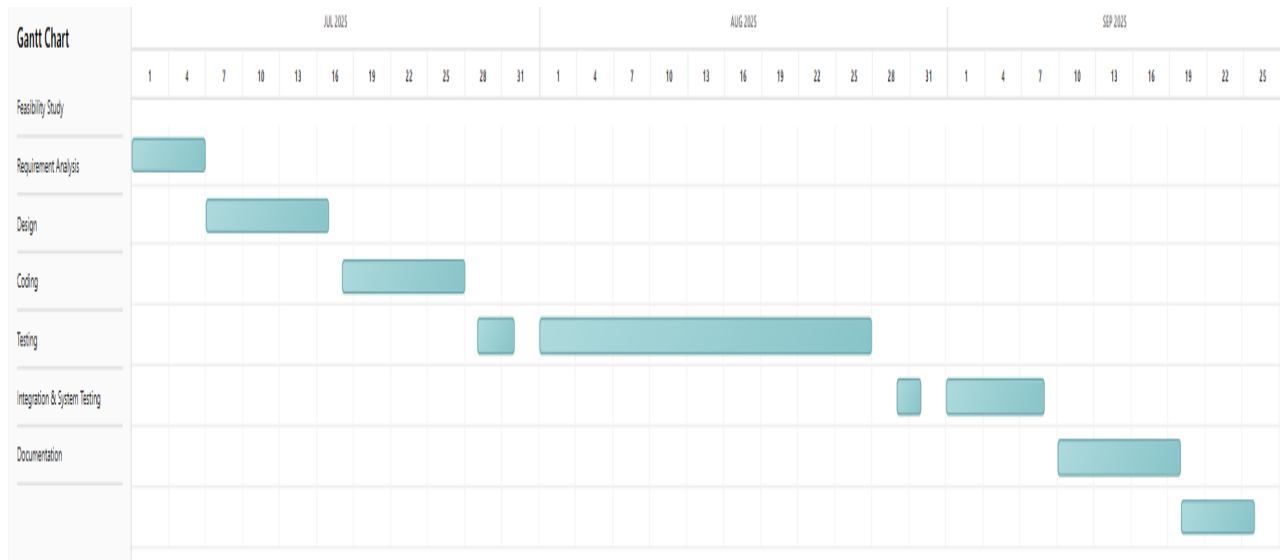
DigiMart offers significant economic benefits for both customers and administrators. Personalized product recommendations help users find relevant items quickly, boosting sales, while the ARIMA based sales prediction allows administrators to anticipate demand and reduce lost opportunities. The web based platform is accessible anytime, supporting convenient shopping and remote management. Overall, DigiMart enhances customer engagement and operational efficiency, making it a cost effective and valuable solution for modern online retail.

## **3. Economic Feasibility**

The DigiMart project is economically viable due to the use of open source technologies and frameworks, minimizing development costs. There are no licensing fees, and all necessary libraries and tools are freely available. With the required technical expertise available in house, development can be completed efficiently.

## **4. Scheduling Feasibility**

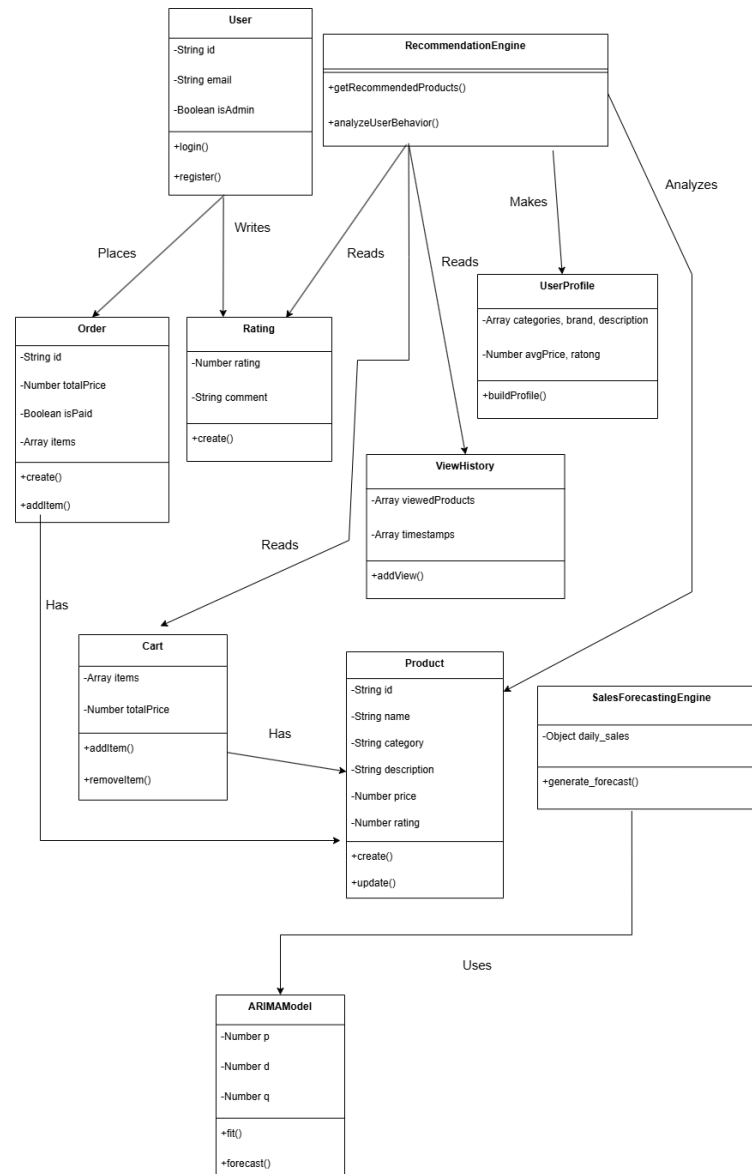
The project is feasible within a defined timeline, structured into clear phases. Each phase is divided into manageable tasks with specific milestones and deadlines to ensure timely progress and delivery.



**Figure 3.2: Gantt Chart**

The project follows a structured prototype development process over a span of 2.5 months, beginning on July 9, 2025. The workflow is divided into multiple phases, including documentation, planning, research, iterative design, coding, and testing. Each development cycle is refined across three iterations to ensure continuous improvement. The final stage includes full system implementation and testing, ensuring a functional and stable system delivery. The documentation process spans the entire timeline to maintain accurate records and support ongoing development.

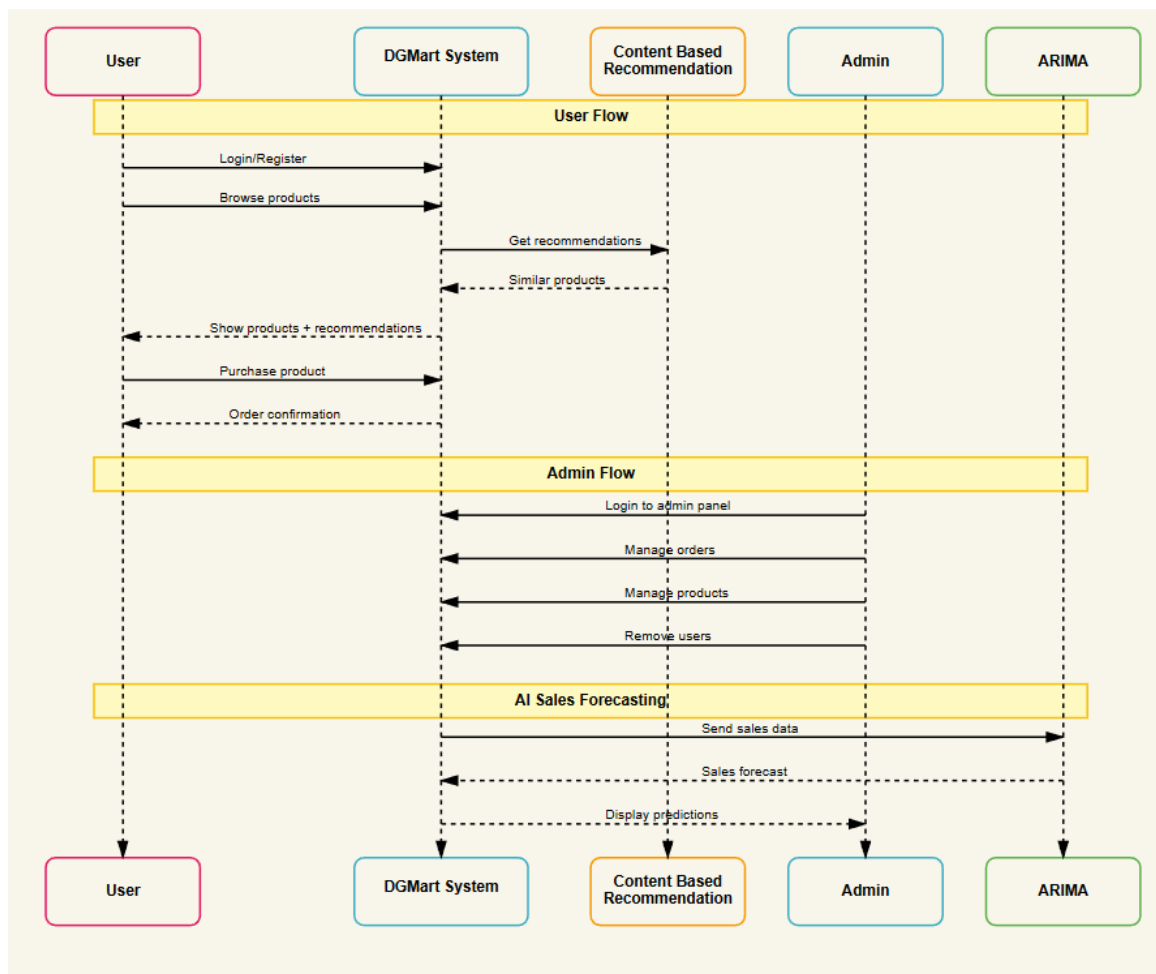
### 3.3 Object Modelling Using Class Diagram



**Figure 3.3: Class Diagram of DigiMart**

The DigiMart platform begins by tracking user interactions, such as product views, cart activity, and purchase history, to build personalized profiles. Based on these profiles, the recommendation engine analyzes behavior and suggests relevant products, while the sales forecasting engine uses historical sales data to predict future trends. Products, orders, and reviews feed into this system, providing continuous feedback to refine recommendations and forecasts. This flow ensures an intelligent, data driven shopping experience that adapts to user preferences and market trends.

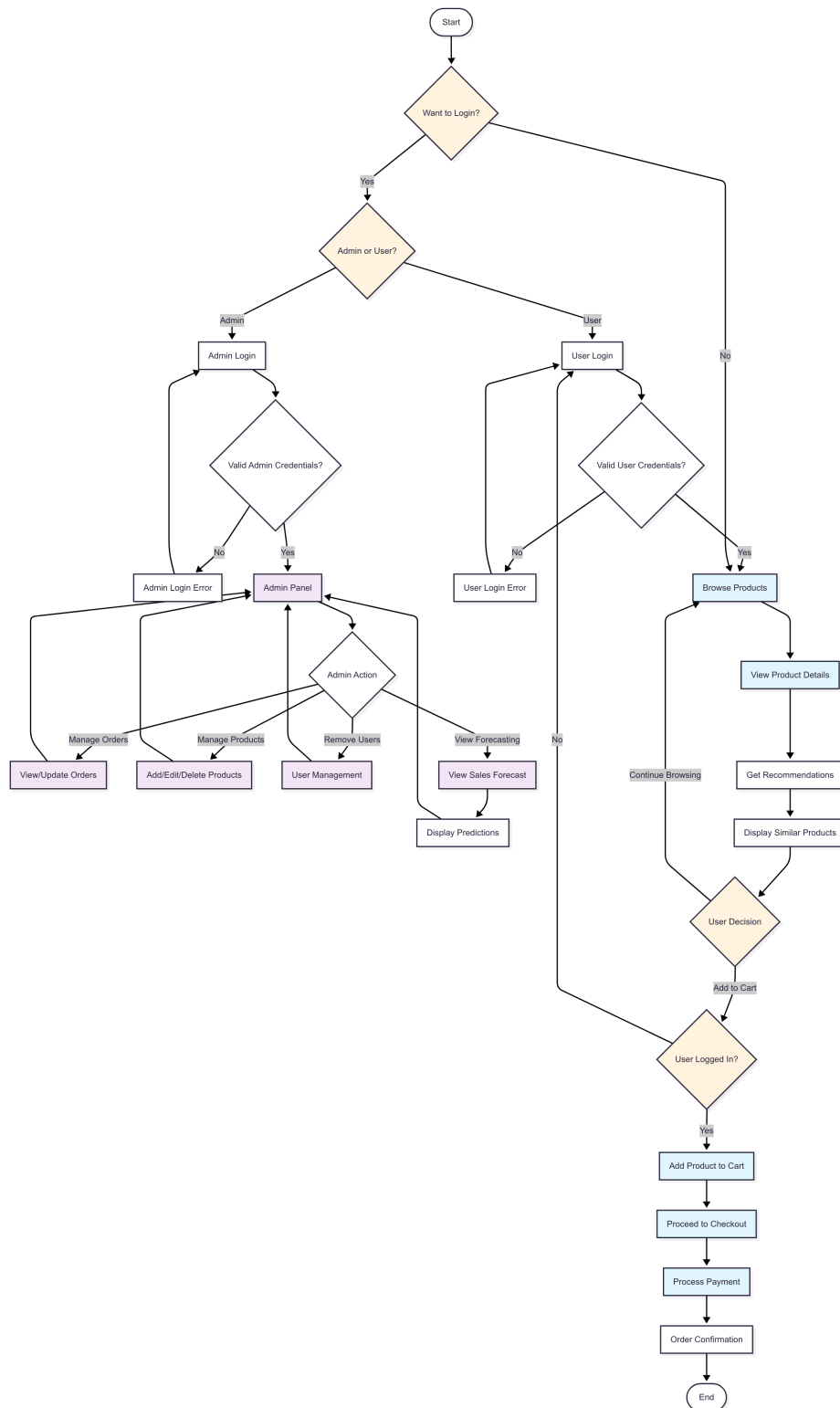
### 3.3.1 Dynamic Modelling using State and Sequence Diagrams



**Figure 3.4: Sequential Diagram of DigiMart**

The process begins when a user logs in or browses products on DigiMart. The system retrieves product data and sends it to the Content Based Recommendation engine, which analyzes user behavior, past purchases, and viewed items to generate personalized suggestions. Recommendations and product details are displayed to the user. When a purchase is made, the order is processed and confirmed. Meanwhile, the Admin can manage orders, products, and users. For sales forecasting, historical sales data is sent to the ARIMA engine, which predicts future trends for display on dashboards, supporting data driven decisions.

### 3.3.2 Process Modeling using Activity Diagram

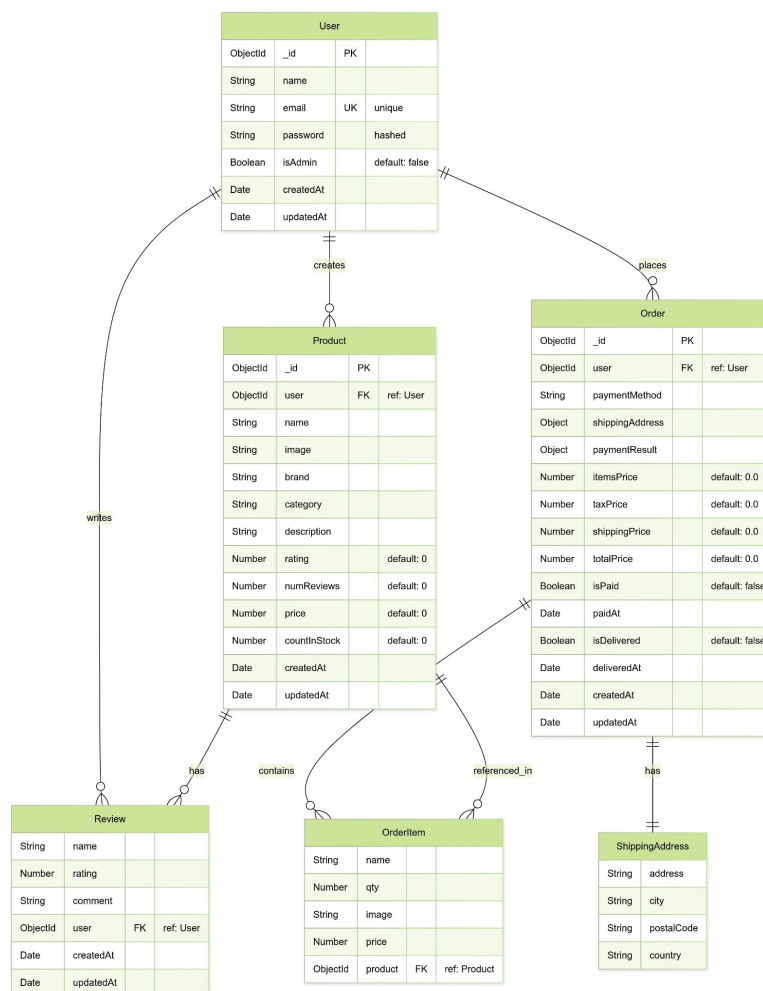


**Figure 3.5: Activity Diagram of DigiMart**

The activity diagram illustrates the operational flow of the DigiMart platform from start to finish. The process begins when a user or admin initiates the login activity. Following

successful authentication, admins proceed to the Admin Panel, where they can perform activities such as managing orders, products, and users, or viewing sales forecasts. Users, on the other hand, navigate through activities including browsing products, viewing product details, and receiving personalized recommendations along with similar product suggestions. Users can then add items to their cart, proceed to checkout, complete payment, and receive order confirmation. Throughout the process, decision points handle scenarios such as invalid login credentials, and users may optionally continue browsing at various stages. The process concludes when the session ends or after order completion, capturing the sequential and conditional flow of activities within the system.

### 3.3.3 Database Schema Diagram



**Figure 3.6: Database Schema Diagram of DigiMart**

This database schema represents a digital marketplace (Digimart) system with six core entities connected through defined relationships. The User entity has one-to-many relationships with Product, Order, and Review. Orders contain multiple OrderItems with each OrderItem

referencing a specific Product, while Orders are also associated with ShippingAddress and PaymentResult entities enabling complete transaction and fulfillment tracking across the marketplace platform.

### 3.3.4 Algorithm Details

**Content-Based Recommendation:** The content based recommendation system suggests products similar to those a user has interacted with by analyzing product features extracted from descriptions and comparing them with the user's preference profile:

1. **Check User Activity:** Determine the source of recommendation request.
  - **No Activity:** Return the latest products sorted by creation date in Explore Mode and terminate the process.
  - **Has Activity:** Use cart items, viewed products, and purchase history to build comprehensive user profile.
2. **Build User Profile:**
  - Aggregate the last 10 products from each relevant source based on activity type.
  - Extract key value features from product descriptions using pattern matching.
  - Aggregate features across all reference items, collecting multiple values for each attribute.
  - Calculate average price and average rating from valid items.
  - Compile unique categories and brands from reference products.
3. **Get All Products:** Retrieve all products from the database for scoring.
4. **Score Each Product:** For every candidate product, calculate two similarity components:
  - (a) **Traditional Similarity with 40% weight:**
    - Category Match: 40% if product category matches profile categories
    - Brand Match: 30% if product brand matches profile brands
    - Price Similarity: 20% based on normalized difference from profile average price

- Rating Similarity: 10% based on normalized difference from profile average rating

(b) **Description Similarity with 60% weight:**

- Extract key value features from the product description
- Compare each product feature against profile features
- Assign match scores: 1.0 for exact match, 0.5 for partial match, 0 for no match
- Calculate average match score across all comparable features
- Track matching attributes with their product and profile values

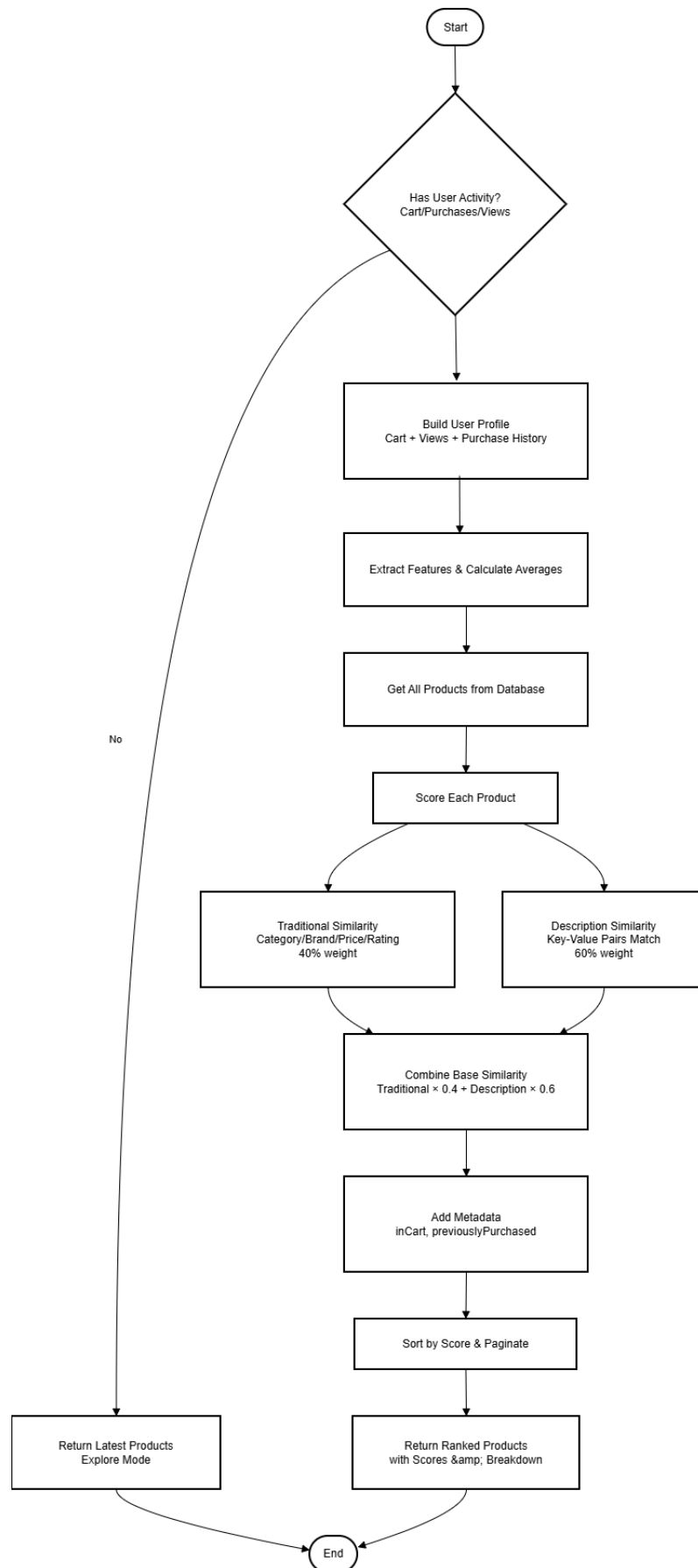
(c) **Compute Final Score:** Calculate weighted combination of traditional and description similarities.

(d) **Add Metadata:** Flag products that are currently in cart or previously purchased.

5. **Sort and Paginate:**

- Sort all products by final similarity score in descending order.
- Assign ranks to products based on similarity.
- Apply pagination: full page for general recommendations, top 4 for product page.

6. **Return Results:** Output includes ranked products with similarity scores, detailed scoring breakdown, feature match details, user profile summary, and product status flags.



**Figure 3.7: Content Based Recommendation Flowchart**

**Time Series Forecasting with ARIMA:** The ARIMA model predicts future revenue and product demand based on historical sales data with train test validation:

**1. Load and Prepare Data:**

- Load sales data from CSV containing date, revenue, quantity, and product category information.
- Group by date to create daily aggregates.
- Apply a 3-day rolling average smoothing to reduce high-frequency noise.

**2. Preprocess Time Series:**

- Apply a  $\log(1+p)$  transformation to stabilize variance while handling zero values.
- Extract the trend component using a 7-day moving average.
- Calculate residuals by removing the trend effect, then apply standardization.

**3. Determine Optimal Parameters:**

- Test combinations of  $p$  (0–2),  $d$  (0–1), and  $q$  (0–2).
- Compute an approximate AIC score for each combination based on residual variance.
- Select the parameter set with the minimum AIC value for the best model fit.

**4. Train-Test Split:**

- Split the historical data into 75% training and 25% testing subsets.
- Train the model using only the training dataset.
- Generate predictions for the test period to validate model accuracy.
- Evaluate model performance using MAE, RMSE, and MAPE metrics.

**5. Fit Final Model and Forecast:**

- Retrain the model on the complete historical dataset using the selected parameters.
- Estimate AR coefficients using autocorrelation analysis.
- Estimate MA coefficients based on residual patterns.

- Generate forecasts for the next 7 or 15 days beyond the available data.

#### **6. Post-Process Forecasts:**

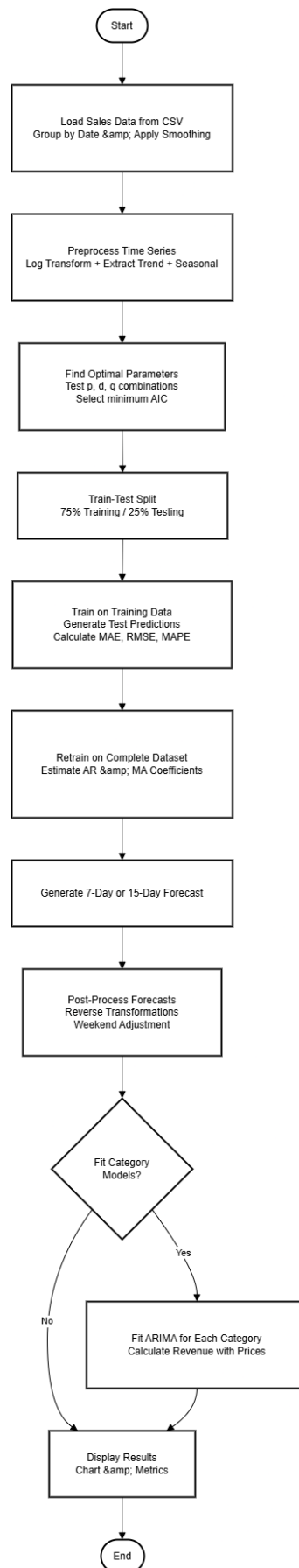
- Apply inverse standardization and cumulative integration to reconstruct the original series.
- Add back trend component.
- Apply the inverse log transformation to obtain actual revenue values.

#### **7. Generate Category Forecasts:**

- Fit separate ARIMA models for each product category using quantity-based data.
- Exclude unreliable categories with excessively high prediction errors.
- Estimate predicted revenue using historical average prices.
- Provide a daily breakdown of forecasted quantities and revenue by category.

#### **8. Display Results:**

- Present predicted revenue, daily averages, and growth rates.
- Plot a line chart showing historical data, test predictions, and future forecasts.
- Display MAE, RMSE, and MAPE metrics as normalized percentage values.
- Show per-category breakdowns with predicted quantities and revenues.



**Figure 3.8: ARIMA Based Forecasting Flowchart**

## CHAPTER 4: IMPLEMENTATION AND TESTING

### 4.1 Implementation

#### 4.1.1 Tools Used

Tools used for the development of this project are discussed below:

- **MongoDB Compass**

A GUI tool for managing MongoDB databases. It allows developers to visually explore collections, run queries, analyze document structures, and monitor database performance in an intuitive interface.

- **VS Code**

A powerful and lightweight Integrated Development Environment (IDE) for writing, debugging, and testing code. It supports extensions, intelligent code suggestions, and version control integration, making development faster and more efficient.

- **Postman**

A comprehensive tool for testing and debugging API endpoints. It enables developers to send requests, inspect responses, automate tests, and verify backend functionality quickly.

- **NPM (Node Package Manager)**

A package manager for Node.js that allows developers to install, update, and manage project dependencies. It also provides scripts for running development servers, building projects, and automating repetitive tasks.

- **Git**

A distributed version control system used to track changes in code, collaborate with team members, and maintain project history. It ensures organized development and easy rollback of code when needed.

#### 4.1.2 Implementation Details of Modules

The DigiMart application is divided into several core modules, each responsible for a specific functionality within the system:

#### **i. Recommendation Module**

This module implements a personalized product recommendation system for the DigiMart platform. It analyzes user interactions, including browsing history, purchase history, and cart items, through the `getRecommendedProducts` method to build a user preference profile based on categories, brands, product descriptions, average prices, and product ratings, which is then used to generate relevant personalized product suggestions.

#### **ii. Forecasting Module**

The sales forecasting module utilizes the ARIMA statistical model to predict future sales trends on the marketplace. It processes historical daily sales data stored as objects and implements the `generateforecast` method to produce time series predictions. This module enables data driven inventory management, helps sellers optimize stock levels, and supports strategic business planning through accurate demand forecasting.

#### **iii. Authentication**

This module handles user authentication, registration, and profile management for the Digimart platform. It implements secure login and registration methods, manages user attributes including email and admin privileges, and maintains user sessions. The module coordinates with other components by providing user data to the recommendation engine, tracking user interactions across orders, carts, and view history.

#### **iv. Database**

This module implements the data persistence layer for the DigiMart platform using MongoDB. It manages core data such as users, products, orders, reviews, order items, shipping addresses, and payment information, while maintaining relationships through document references. The database supports efficient storage and retrieval of user accounts, product catalogs, purchase transactions, order details, and other marketplace data required by the system.

#### **v. Frontend**

This module provides the user facing interface for the Digimart platform enabling seamless interaction with all marketplace features. It implements responsive views

for product browsing, shopping cart management, order placement, and user authentication including login and registration flows. The frontend integrates with the recommendation engine to display personalized product suggestions, presents sales analytics and forecasts through visual charts, and allows users to submit ratings and reviews while managing their profiles, view history, and order tracking across the digital marketplace.

## 4.2 Testing

Software testing is a critical phase in the software development lifecycle, ensuring that applications function as intended, are free from defects, and meet user requirements. By systematically identifying and addressing issues before deployment, testing guarantees a seamless user experience and enhances overall software reliability. This essential process not only improves product quality but also increases user satisfaction.

### 4.2.1 Test Cases for Unit Testing

**Table 4.1: Login and Registration Testing**

| Sn | Test Cases   | Input                                        | Expected Results        | Actual Results          | Status  |
|----|--------------|----------------------------------------------|-------------------------|-------------------------|---------|
| 1  | Login        | Entered Login Info i.e Username and Password | Login Successful        | Login Successful        | Success |
| 2  | Login        | Entered Wrong Login Info                     | Login Failed            | Login Successful        | Fail    |
| 3  | Login        | Entered Wrong Login Info                     | Login Failed            | Login Failed            | Success |
| 4  | Registration | Entered valid details                        | Registration Successful | Registration Successful | Success |

**Table 4.2: Admin Modules Testing**

| Sn | Test Cases                            | Input                     | Expected Results    | Actual Results                           | Status  |
|----|---------------------------------------|---------------------------|---------------------|------------------------------------------|---------|
| 1  | Adding Product                        | Pressed on Create Product | New Product Created | New Product Created                      | Success |
| 2  | Editing Product                       | Edited some attributes    | Edit Successful     | Edit Successful                          | Success |
| 3  | Delete Product                        | Deleting a Product        | Deletion Success    | Deletion Success                         | Success |
| 4  | Viewing Order Detail                  | Open Orders Tab           | Display order list  | Display order list                       | Success |
| 5  | Give Administrative Privilege to user | Updating User Profile     | Update Successful   | Success message received but not updated | Fail    |
| 6  | Give Administrative Privilege to user | Updating User Profile     | Update Successful   | Failed Message                           | Fail    |
| 7  | Give Administrative Privilege to user | Updating User Profile     | Update Successful   | Update Successful                        | Success |

**Table 4.3: User Testing**

| Sn | Test Cases         | Input                            | Expected Results      | Actual Results        | Status  |
|----|--------------------|----------------------------------|-----------------------|-----------------------|---------|
| 1  | Open Product Page  | Click on Product Page            | Product Page Opens    | Product Page Opens    | Success |
| 2  | Cart               | Add Items To Cart                | Items are added       | Items are added       | Success |
| 3  | Checkout           | Go through the check-out Process | Order Confirmed       | Error Message         | Fail    |
| 4  | Checkout           | Go through the check-out Process | Order Confirmed       | Order Confirmed       | Success |
| 5  | View Order History | Open Profile                     | List of ordered items | Error while loading   | Fail    |
| 6  | View Order History | Open Profile                     | List of ordered items | List of ordered items | Success |

**Table 4.4: Forecasting Module Testing**

| Sn | Test Cases                         | Input             | Expected Results               | Actual Results                  | Status  |
|----|------------------------------------|-------------------|--------------------------------|---------------------------------|---------|
| 1  | Load CSV Data                      | Valid CSV file    | Data loaded successfully       | Data loaded successfully        | Success |
| 2  | Load Missing CSV                   | Non existent file | Empty dataframe handled        | Empty dataframe handled         | Success |
| 3  | Category Sales Aggregation         | Standard data     | Category wise sales calculated | Some categories missing         | Fail    |
| 4  | Category Sales Aggregation Retry 1 | Standard data     | Category wise sales calculated | Some categories missing         | Fail    |
| 5  | Category Sales Aggregation Retry 2 | Standard data     | Category wise sales calculated | Some categories missing         | Fail    |
| 6  | Category Sales Aggregation Retry 3 | Standard data     | Category wise sales calculated | All categories calculated       | Success |
| 7  | Fit Category ARIMA                 | Category series   | Category ARIMA models trained  | Some models not trained         | Fail    |
| 8  | Fit Category ARIMA Retry           | Category series   | Category ARIMA models trained  | All models trained              | Success |
| 9  | Forecast 15 Days                   | Valid period      | 15 day forecast generated      | Forecast failed for some days   | Fail    |
| 10 | Forecast 15 Days Retry             | Valid period      | 15 day forecast generated      | Forecast generated successfully | Success |
| 11 | Metrics Calculation                | 7 day forecast    | MAE, RMSE, MAPE calculated     | Metrics calculation failed      | Fail    |
| 12 | Metrics Calculation Retry          | 7 day forecast    | MAE, RMSE, MAPE calculated     | Metrics calculated successfully | Success |

**Table 4.5: Recommendation Testing**

| Sn | Test Cases             | Input                                       | Expected Results                         | Actual Results                          | Status  |
|----|------------------------|---------------------------------------------|------------------------------------------|-----------------------------------------|---------|
| 1  | Purchase History Fetch | User with valid ID                          | All past purchased products returned     | All past purchased products returned    | Success |
| 2  | View Based Suggestion  | User views product page                     | Similar products recommended             | Irrelevant products shown               | Fail    |
| 3  | View Based Suggestion  | User views product page                     | Similar products recommended             | Error in similarity mapping             | Fail    |
| 4  | View Based Suggestion  | User views product page                     | Similar products recommended             | Similar products recommended            | Success |
| 5  | Cart Suggestion        | User adds item to cart                      | Accessories and similar items suggested  | No suggestions triggered                | Fail    |
| 6  | Cart Suggestion        | User adds item to cart                      | Accessories and similar items suggested  | Accessories and similar items suggested | Success |
| 7  | Feature Extraction     | Product with detailed description           | Key value pairs extracted correctly      | Partial extraction; missing keys        | Fail    |
| 8  | Feature Extraction     | Product with detailed description           | Key value pairs extracted correctly      | Key value pairs extracted correctly     | Success |
| 9  | Feature Aggregation    | Multiple products with overlapping features | Aggregated unique feature set produced   | Aggregated feature set produced         | Success |
| 10 | Traditional Scoring    | Product matches category/brand              | Score increases for category and brand   | Score increases for category and brand  | Success |
| 11 | Description Similarity | Product with matching features              | High description similarity score        | Low similarity score detected           | Fail    |
| 12 | Description Similarity | Product with matching features              | High description similarity score        | Inconsistent similarity output          | Fail    |
| 13 | Description Similarity | Product with matching features              | High description similarity score        | High description similarity score       | Success |
| 14 | Final Ranking          | Multiple candidate products                 | Products ranked by total score           | Ranked incorrectly (price biased)       | Fail    |
| 15 | Final Ranking          | Multiple candidate products                 | Products ranked by total score           | Products ranked by total score          | Success |
| 16 | Error Handling         | Invalid userId provided                     | Empty result set returned safely         | Empty result set returned safely        | Success |
| 17 | Empty Cart Handling    | User has no items in cart                   | Recommendations based on views/purchases | Recommendations generated correctly     | Success |
| 18 | No Activity User       | User has no purchases or views              | Latest products displayed                | Latest products displayed               | Success |

### 4.2.2 Test Cases for System Testing

**Table 4.6: System Integration Testing**

| Sn | Test Cases                               | Input                               | Expected Results                               | Actual Results                                | Status  |
|----|------------------------------------------|-------------------------------------|------------------------------------------------|-----------------------------------------------|---------|
| 1  | Login and Product Page Access            | User logs in and opens product page | Product page loads successfully                | Product page loads successfully               | Success |
| 2  | Product Detail and Recommendation Detail | Open product page                   | Product details displayed with recommendations | Recommendations missing                       | Fail    |
| 3  | Product Detail and Recommendation Detail | Retry opening product page          | Product details and recommendations displayed  | Product details and recommendations displayed | Success |
| 4  | Admin Product Addition and Visibility    | Admin adds a new product            | Product visible to users                       | Product not visible                           | Fail    |
| 5  | Admin Product Addition and Visibility    | Retry adding the product            | Product visible to users                       | Product not visible                           | Fail    |
| 6  | Admin Product Addition and Visibility    | Successful addition                 | Product visible to users                       | Product visible to users                      | Success |
| 7  | Admin Product Deletion and User View     | Admin deletes product               | Product no longer visible to users             | Product still visible                         | Fail    |
| 8  | Admin Product Deletion and User View     | Retry deletion                      | Product no longer visible                      | Product no longer visible                     | Success |
| 9  | Combined Forecast Retrieval              | Request forecast for all categories | Forecast data returned for each category       | Some category forecasts missing               | Fail    |
| 10 | Combined Forecast Retrieval              | Retry request                       | Forecast data returned correctly               | Forecast data returned correctly              | Success |
| 11 | Retrain Forecast Engine                  | Trigger retrain API                 | All models retrained successfully              | Retrain failed                                | Fail    |
| 12 | Retrain Forecast Engine                  | Retry retrain                       | Models retrained successfully                  | Models retrained successfully                 | Success |
| 13 | Forecast Top Products Calculation        | Request top product forecast        | Top products returned with predicted sales     | Top products missing                          | Fail    |
| 14 | Forecast Top Products Calculation        | Retry top product forecast          | Top products returned correctly                | Top products returned correctly               | Success |
| 15 | Forecast Metrics Calculation             | Request metrics for main model      | Metrics calculated correctly                   | Error calculating metrics                     | Fail    |
| 16 | Forecast Metrics Calculation             | Retry metrics calculation           | Metrics calculated correctly                   | Metrics calculated correctly                  | Success |

### 4.3 Result Analysis

**ARIMA-based Time Series Forecasting model** was tested with multiple train-test split configurations to identify the most effective setup for accurate demand prediction. Among all evaluated splits, the 75-25 configuration produced the best results, achieving the lowest error rates and highest overall accuracy, indicating an optimal balance between training and

testing data.

**Table 4.7: ARIMA Model Evaluation Summary**

| Train Test Split | MAE           | RMSE          | MAPE        | Accuracy     | Rank       |
|------------------|---------------|---------------|-------------|--------------|------------|
| 70-30            | 14,503        | 18,094        | 9.6%        | 90.4%        | 3rd        |
| <b>75-25</b>     | <b>12,079</b> | <b>14,765</b> | <b>8.5%</b> | <b>91.5%</b> | <b>1st</b> |
| 80-20            | 18,323        | 21,933        | 12.2%       | 87.8%        | 4th        |
| 90-10            | 13,902        | 17,713        | 9.4%        | 90.6%        | 2nd        |

**Content Based recommendation** system generates personalized recommendations by analyzing product features and user interaction history, building profiles that capture category preferences, brand affinity, price ranges, and technical specifications. It computes Traditional Similarity (high-level attributes: category, brand, price, ratings) and Description Similarity (deep feature matching of technical specs), combining them as:

$$\text{Final Score} = 0.40 \times \text{Traditional Similarity} + 0.60 \times \text{Description Similarity}$$

Home page recommendations suggest diverse items using complete interaction history, while Product page recommendations generate similar items based on a single product context. Evaluation used a test profile with Tablets, Smartwatches, and Laptops from Samsung and Apple.

**Table 4.8: Home Page Recommendation Results**

| Rank | Product Name           | Category   | Price      | Rating | Score |
|------|------------------------|------------|------------|--------|-------|
| 1    | Samsung Galaxy Tab S7  | Tablet     | \$499.99   | 4.3    | 96%   |
| 2    | Apple Watch Series 7   | Smartwatch | \$399.99   | 4.7    | 96%   |
| 3    | Samsung Galaxy Watch 4 | Smartwatch | \$249.99   | 4.4    | 94%   |
| 4    | MacBook Pro 14-inch    | Laptop     | \$1,999.99 | 4.8    | 92%   |
| 5    | Dell XPS 13            | Laptop     | \$1,199.99 | 4.5    | 45%   |

**Table 4.9: Product Page Recommendation Results (Context: iPhone 13)**

| Rank | Product Name          | Category   | Price    | Rating | Score |
|------|-----------------------|------------|----------|--------|-------|
| 1    | iPhone 13 (Self)      | Smartphone | \$599.99 | 4.0    | 100%  |
| 2    | Google Pixel 6        | Smartphone | \$599.99 | 4.3    | 43%   |
| 3    | Samsung Galaxy S21    | Smartphone | \$649.99 | 4.2    | 42%   |
| 4    | Samsung Galaxy Tab S7 | Tablet     | \$499.99 | 4.3    | 29%   |
| 5    | Apple iPad Air        | Tablet     | \$599.99 | 4.6    | 27%   |

The results demonstrate that the content-based approach effectively identifies relevant recommendations by matching technical specifications with user preferences. Home page recommendations achieved high scores (92-96%) by aligning multiple product features with the user's historical interaction patterns, while product page recommendations appropriately prioritized similar products within the same category before suggesting alternatives.

## **CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION**

### **5.1 Conclusion**

This project demonstrates the effective integration of machine learning and intelligent recommendation systems within a digital marketplace platform. The DigiMart application analyzes user interactions, purchase history, and product information to generate personalized recommendations and sales forecasts. By applying algorithms for feature extraction, similarity computation, and predictive analysis, the system enhances decision making and provides a dynamic, user oriented experience. The platform highlights the practical implementation of AI techniques for improving efficiency, personalization, and overall system intelligence in modern digital environments.

### **5.2 Future Recommendations**

Future improvements may include implementing collaborative filtering to enhance the recommendation accuracy by leveraging user similarity patterns and purchase behaviors. Additionally, expanding the system into a cross-platform environment by integrating mobile and desktop versions would improve accessibility and user reach.

## REFERENCES

- [1] A. Thami, K. R. Joshi, O. Sapkota, S. M. Lopchan, S. Parajuli, and S. Khanal, “E-commerce platform of hukut store.” <https://www.studocu.com/row/document/tribhuvan-university/bachelor-in-business-administration/e-commerce-report/93805097>, 2022. Accessed: 2025-08-15.
- [2] C. Basu, H. Hirsh, W. Cohen, *et al.*, “Recommendation as classification: Using social and content-based information in recommendation,” in *Aaai/iaai*, pp. 714–720, 1998.
- [3] P. Ramos, N. Santos, and R. Rebelo, “Performance of state space and arima models for consumer retail sales forecasting,” *Robotics and computer-integrated manufacturing*, vol. 34, pp. 151–163, 2015.

# APPENDIX

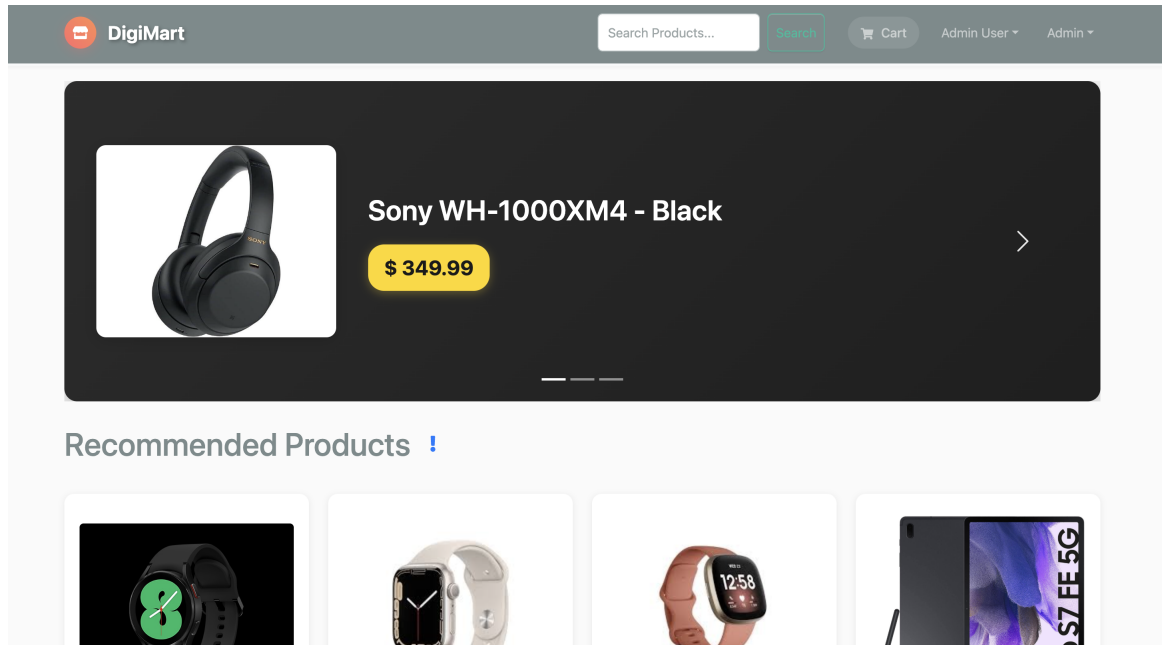


Fig 1: Home Page

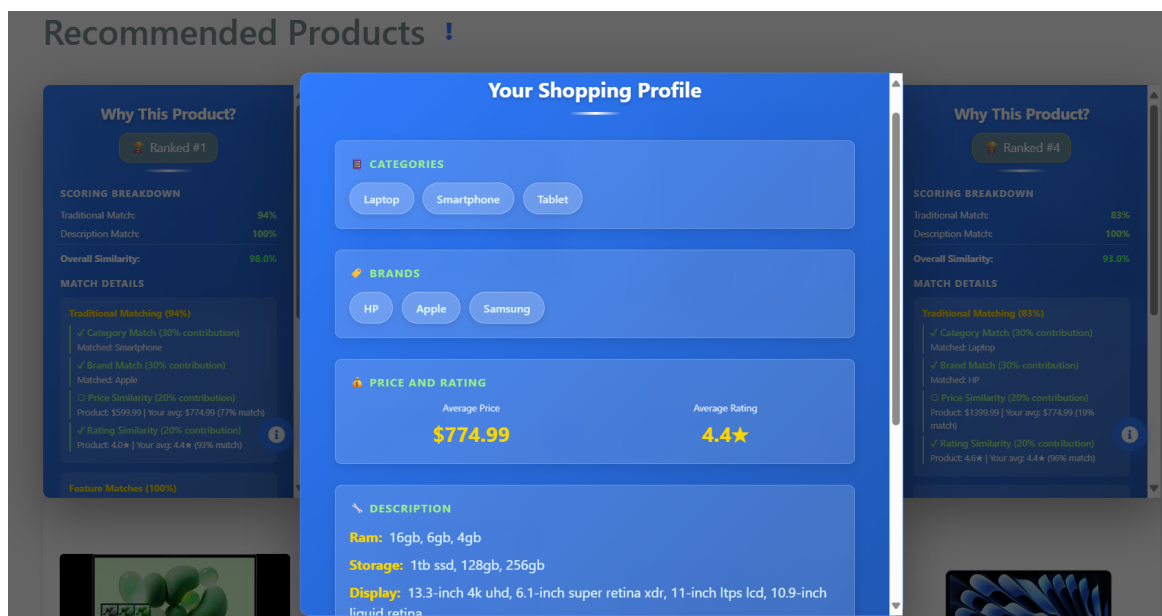


Fig 2: Recommendation Details

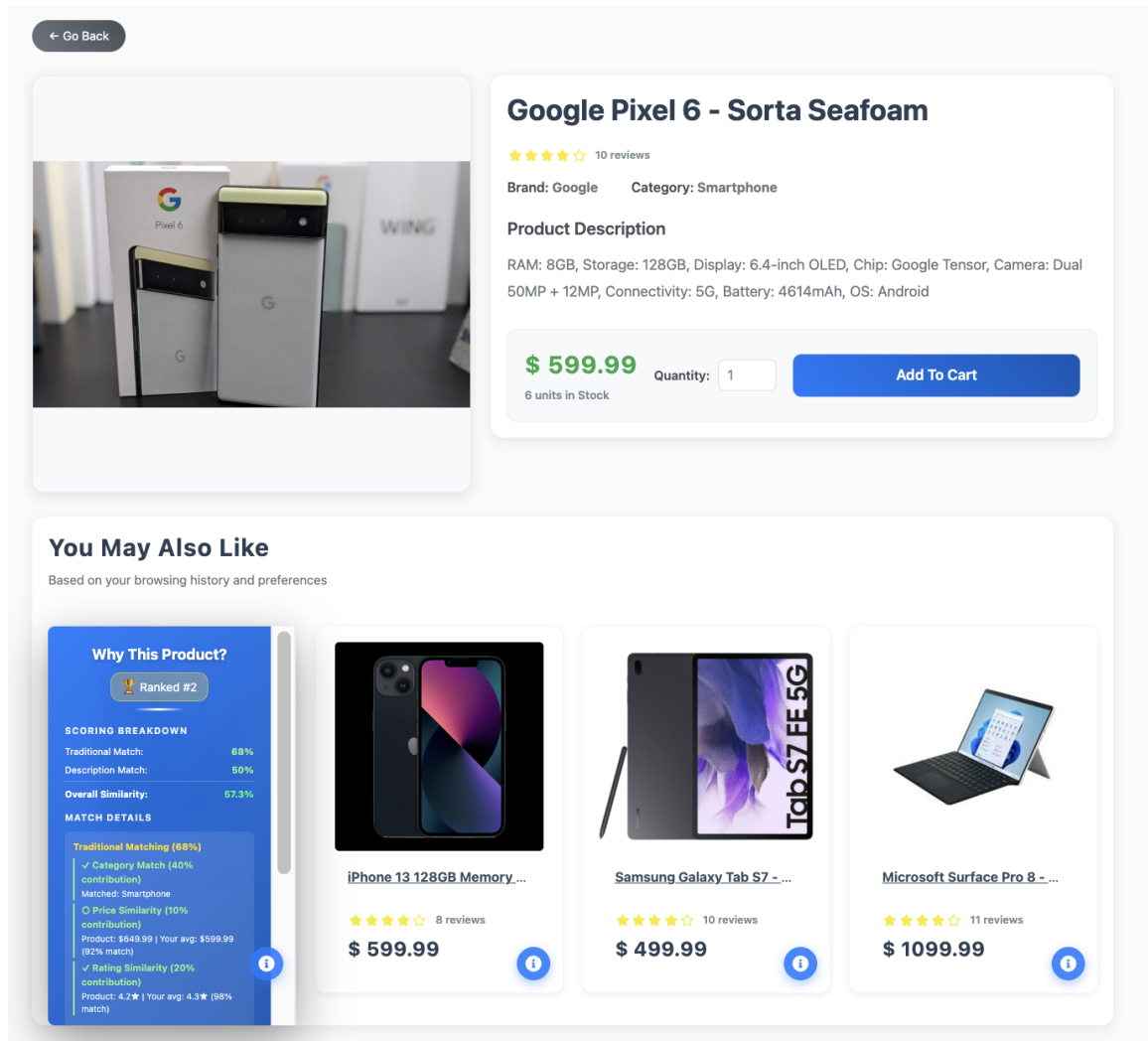


Fig 3: Product Page

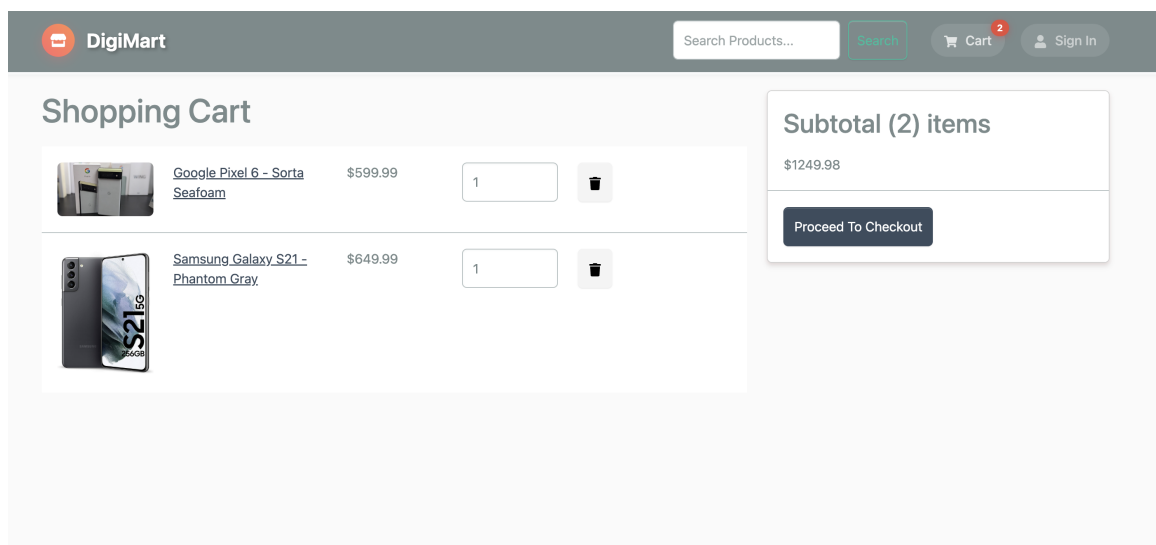


Fig 4: Cart Items

My Profile

Manage your account and view order history

Account Information

Full Name

John Doe

Email Address

ram@gmail.com

New Password

Enter new password (optional)

Leave blank to keep current password

Confirm Password

Confirm new password

Update Profile

My Orders

| ORDER ID                 | DATE         | TOTAL      | ACTIONS            |
|--------------------------|--------------|------------|--------------------|
| 6a815cb05cd4120eb1b2b0e0 | Aug 16, 2026 | \$ 1437.48 | <div>Details</div> |

Fig 5: User Profile

Products Management

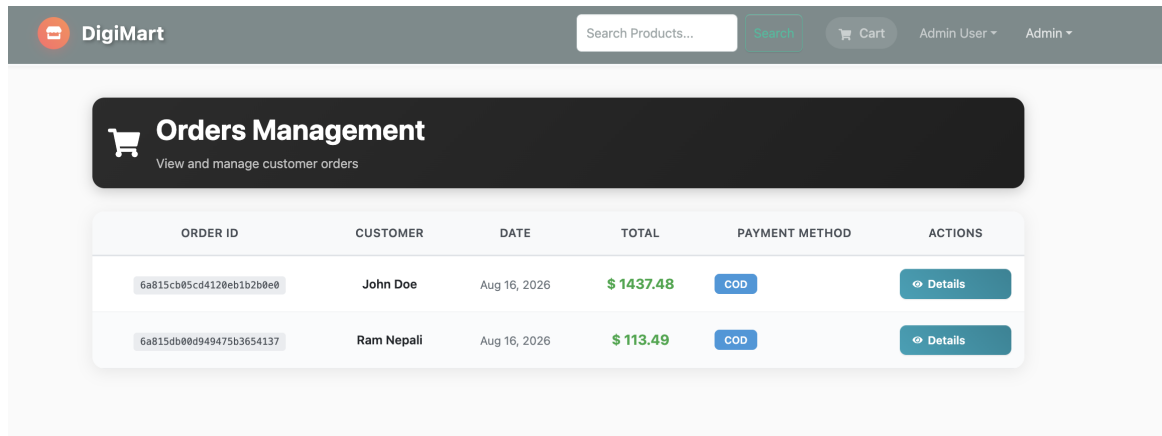
Manage your product catalog

Create Product

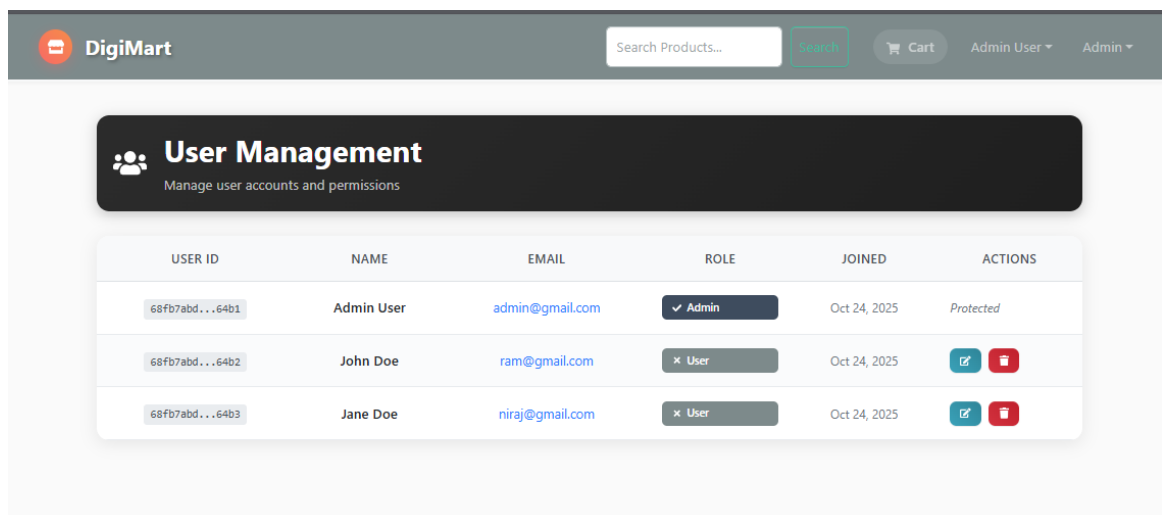
| PRODUCT ID               | NAME                                     | \$ PRICE  | CATEGORY   | BRAND     | ACTIONS                 |
|--------------------------|------------------------------------------|-----------|------------|-----------|-------------------------|
| 6a7d5f4b3da0ec0bebac507e | iPhone 13 128GB Memory - Midnight Black  | \$599.99  | Smartphone | Apple     | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac507f | Samsung Galaxy S21 - Phantom Gray        | \$649.99  | Smartphone | Samsung   | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac5080 | Google Pixel 6 - Sorta Seafoam           | \$599.99  | Smartphone | Google    | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac5081 | Samsung Galaxy Tab S7 - Mystic Black     | \$499.99  | Tablet     | Samsung   | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac5082 | Apple iPad Air - Sky Blue                | \$599.99  | Tablet     | Apple     | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac5083 | Microsoft Surface Pro 8 - Platinum       | \$1099.99 | Tablet     | Microsoft | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac5084 | Dell XPS 13 Laptop - Platinum Silver     | \$1199.99 | Laptop     | Dell      | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac5085 | HP Spectre x360 Laptop - Nightfall Black | \$1399.99 | Laptop     | HP        | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac5086 | MacBook Pro 14-inch - Space Gray         | \$1999.99 | Laptop     | Apple     | <div></div> <div></div> |
| 6a7d5f4b3da0ec0bebac5087 | Apple Watch Series 7 - Starlight         | \$399.99  | Smartwatch | Apple     | <div></div> <div></div> |

Fig 6: Product Management

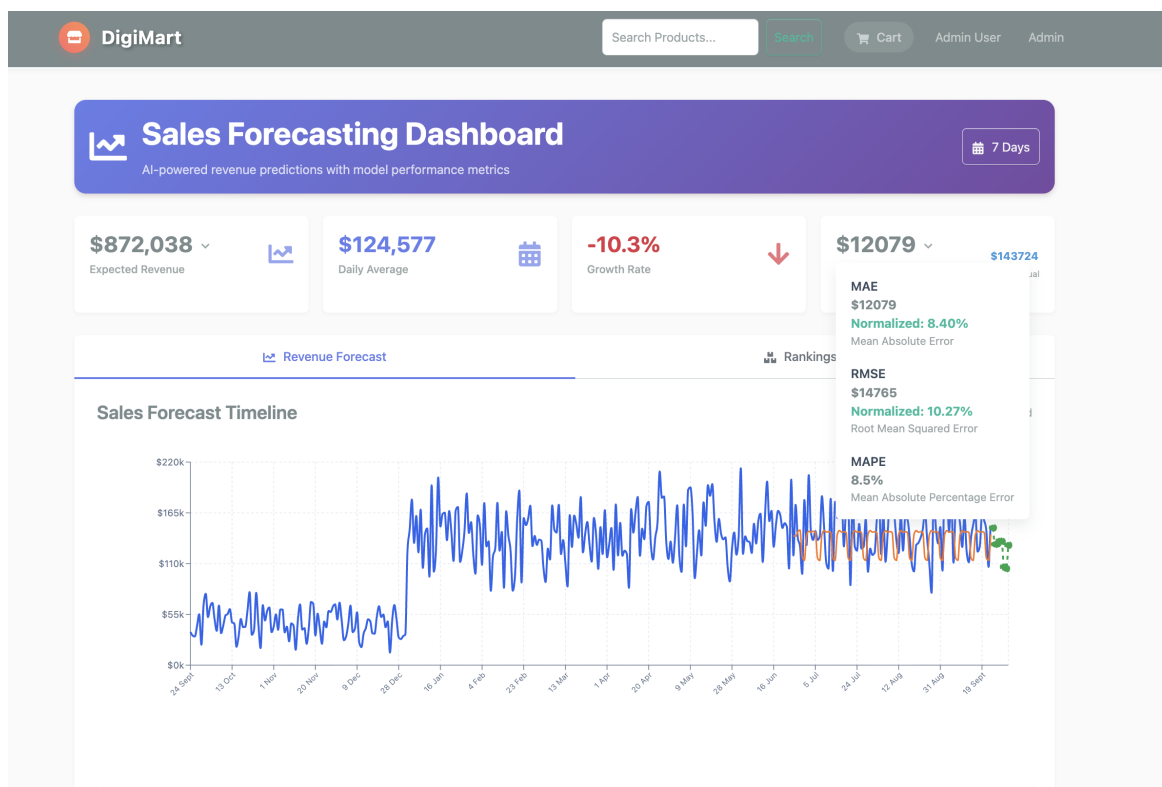
34



**Fig 7: Order Management**



**Fig 8: User Management**



**Fig 9: Sales Forecasting**