**Lumbini City College**

(Affiliated to Tribhuvan University)

****

**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**A Project Report On**

**“Machine Learning Integrated E-Commerce Application”**

**Submitted to:**

**Lumbini City College**

**Department of Computer Application**

**Tilottama - 4, Rupandehi**

***In the partial fulfillment of the requirement for the Bachelor of Computer Application***

**Submitted by:**

Bishwa Bhusal

[6-2-1134-15-2018]

**Under the Supervision of**

Mr. Kamal Bahadur Thapa

ACKNOWLEDGEMENT

The successful completion of this project report would not have been possible without the support and assistance of many individuals and organizations. I feel immensely blessed to have got this support during my internship program. I would like to take this opportunity to offer my earnest admiration for each one of them.

I would like to express our profound thanks to my project supervisor “Mr. Kamal Thapa” who took a keen interest in our project and guided me throughout the entire project by providing the necessary ideas, information, and knowledge for developing “Machine Learning Integrated E-Commerce Application”.

I am especially grateful to Mr. Bishal Kandel (Coordinator), Mr. Ashok Gurung, Mr. Bhuban Panthee, and Mr. Suraj Kumar Khatri of Lumbini City College, who have also been a positive and encouraging influence on my research efforts.

I owe my deepest gratitude to the authors whose work has been used in this report for the completion of this report.

Finally, I thank all my teachers and colleagues, who were the people who prepared me for this endeavor. I owe you all my success.

ABSTRACT

This project has been submitted in fulfillment of the requirements for the Bachelor of Computer Application. This report describes the design and implementation of an e-commerce application that utilizes machine learning techniques to improve the user experience and increase sales. The application employs various machine learning models, including recommendation systems, sentiment analysis, and image recognition, to personalize product recommendations, analyze customer feedback, and streamline the product search process. The report outlines the key features and functionality of the application, including its user interface, back-end architecture, and machine learning algorithms. It also discusses the challenges encountered during development, such as data quality and model performance, and presents solutions to overcome them. The results of user testing and performance evaluation demonstrate the effectiveness of the application in enhancing customer engagement and improving sales. Overall, this report provides insights into the potential benefits of integrating machine learning into e-commerce applications and highlights the importance of data-driven decision making in the development process. **Keywords*:*** *E-Commerce Application, Machine Learning, Product Recommendation*

TABLE OF CONTENTS

[Chapter 1: Introduction 1](#_Toc134795530)

[1.1 Introduction 1](#_Toc134795531)

[1.2 Problem Statement 1](#_Toc134795532)

[1.3 Objective 2](#_Toc134795533)

[1.4 Scope and Limitations 2](#_Toc134795534)

[1.5 Report Organization 3](#_Toc134795535)

[Chapter 2: Background Study and Literature Review 4](#_Toc134795536)

[2.1 Background Study 4](#_Toc134795537)

[2.2 Literature Review 4](#_Toc134795538)

[3.1 Research Methodology 6](#_Toc134795539)

[3.2 System Analysis 6](#_Toc134795540)

[3.2.1 Requirement Analysis 6](#_Toc134795541)

[3.2.1.1 Functional Requirements: 6](#_Toc134795542)

[3.2.1.2 Non-functional Requirements: 7](#_Toc134795543)

[3.2.2 Feasibility Study 8](#_Toc134795544)

[3.2.3 Analysis 8](#_Toc134795545)

[3.3 System Modelling 9](#_Toc134795546)

[3.3.1 Data Modelling: ER-Diagram 9](#_Toc134795547)

[3.3.2 Use–case Diagram 10](#_Toc134795548)

[3.3.3 Process Modelling: Data Flow Diagram (DFD) 11](#_Toc134795549)

[3.3.4 Gantt Chart 14](#_Toc134795550)

[Chapter 4: System Design 15](#_Toc134795551)

[4.1 System Design 15](#_Toc134795552)

[5.1 Implementation 16](#_Toc134795553)

[5.1.1 Tools Used (CASE tools, Programming languages, Database platforms) 16](#_Toc134795554)

[5.1.1.1 CASE 16](#_Toc134795555)

[5.1.2 Languages Used 17](#_Toc134795556)

[5.2 Testing 18](#_Toc134795557)

[5.2.2 Integration Testing 20](#_Toc134795558)

[5.2.3 Test Cases for System Testing 21](#_Toc134795559)

[Chapter 6: Conclusion and Future Recommendation 22](#_Toc134795560)

[6.1 Conclusion 22](#_Toc134795561)

[6.2 Future Recommendation 22](#_Toc134795562)

LIST OF TABLES

Table-1: Unit and System Testing 21

LIST OF FIGURES

Fig-i: Use Case Diagram 7

Fig-1: Level-0 DFD 10

Fig-2: Level-1 DFD 10

Fig-0: ER-Diagram 11

LIST OF ABBREVIATIONS

# Chapter 1: Introduction

## 1.1 Introduction

A machine learning integrated e-commerce application is an online platform that uses machine learning algorithms to help online retailers better understand their customer's behavior, preferences, and purchasing patterns, and then provide personalized and targeted recommendations and marketing strategies.

By leveraging the vast amounts of data generated by online shoppers, machine learning algorithms can identify trends and patterns that are not immediately apparent to human analysts, and then use this information to improve the shopping experience for customers and increase revenue for retailers.

Some common applications of machine learning in e-commerce include product recommendations based on a customer's browsing and purchasing history, personalized search results based on a customer's preferences and behavior, fraud detection and prevention, and dynamic pricing strategies based on real-time supply and demand data.

Overall, a machine learning-integrated e-commerce application can help online retailers gain a deeper understanding of their customers and market trends, improve the customer experience, and increase revenue through personalized and targeted marketing strategies.

## 1.2 Problem Statement

The e-commerce industry is rapidly growing and becoming more competitive, and there is a need for e-commerce applications that can provide personalized experiences to customers to improve their shopping experiences. Traditional e-commerce applications are limited in their ability to understand customer behavior, preferences, and purchase patterns, which results in a lack of personalized recommendations and promotions. To address this issue, we propose the development of a machine learning integrated e-commerce application that can analyze customer data to provide personalized recommendations, promotions, and a seamless shopping experience. The application should be able to analyze customer behavior, preferences, and purchase patterns to generate relevant and personalized product recommendations, optimize pricing and promotions to maximize sales, and provide an easy-to-use interface that enhances the shopping experience. By developing such an application, we aim to provide a competitive edge to e-commerce businesses and improve customer satisfaction, retention, and loyalty.

## 1.3 Objective

1. To develop an e-commerce application that provides a personalized and seamless shopping experience to customers.
2. To integrate machine learning techniques into the e-commerce application to enhance the customer experience.
3. To develop a recommendation system that suggests products to customers based on their browsing and purchase history.

## 1.4 Scope and Limitations

Machine learning (ML) integrated e-commerce applications have become increasingly popular due to their ability to personalize user experience, improve customer service, and increase revenue. However, like any technology, they also have their scope and limitations.

**Scope:**

1. Personalization: One of the main advantages of ML-integrated e-commerce applications is their ability to personalize the user experience. Machine learning algorithms can analyze customer data and behavior to make personalized recommendations, predict purchasing patterns, and offer relevant products or services.
2. Fraud detection: ML algorithms can be trained to detect fraudulent activities and protect e-commerce businesses from financial losses.
3. Inventory management: ML algorithms can analyze data on customer preferences, seasonal demand, and product popularity to optimize inventory management and avoid stockouts.
4. Customer service: ML can automate customer service tasks such as responding to frequently asked questions and providing personalized support.
5. Marketing: ML can help optimize marketing campaigns by predicting customer behavior and targeting specific audiences with personalized advertisements.

**Limitations:**

1. Data quality: ML algorithms heavily rely on the quality of the data they are fed. Inaccurate or biased data can lead to incorrect predictions or recommendations.
2. Lack of human oversight: While automation can help streamline processes, it's important to have human oversight to ensure that the algorithms are making the right decisions.
3. Cost: Implementing ML-integrated e-commerce applications can be costly, especially for small businesses.
4. Privacy concerns: The use of customer data can raise privacy concerns, and e-commerce businesses must ensure that they are following ethical and legal guidelines for data collection and usage.
5. Complexity: ML algorithms can be complex and difficult to understand, which can make it challenging for businesses to implement them effectively.

## 1.5 Report Organization

* Chapter One **Introduction** includes the problem statement, objectives, scope and limitation, and development methodology of this project.
* Chapter Two **Organization Details and Literature Review** includes a background study and Literature Review of this project.
* Chapter Three **System Analysis** consists of Requirement Analysis, Feasibility Analysis, and Structured or Object-Oriented Analysis used in the system which will be developed.
* Chapter Four **Implementation and Testing** includes topics like Implementation, Testing, and Result Analysis of this project.
* Chapter Five **Conclusion and Future Recommendation** includes the conclusion of the system and future recommendations from the project developed.

# Chapter 2: Background Study and Literature Review

## 2.1 Background Study

Machine learning-integrated e-commerce applications have become increasingly popular in recent years, as businesses seek to gain a competitive advantage in the rapidly evolving online marketplace. To develop such applications, it is important to have a strong background in both machine learning and e-commerce.

Firstly, a thorough understanding of machine learning is essential. This involves knowledge of various machine learning algorithms, such as neural networks, decision trees, and clustering techniques. Additionally, proficiency in programming languages such as Python and R is crucial, as well as familiarity with popular machine learning libraries such as TensorFlow and Scikit-Learn.

Overall, the integration of machine learning and e-commerce has the potential to transform the way we shop online. By leveraging the power of machine learning algorithms, e-commerce businesses can provide a more personalized and efficient shopping experience for their customers, leading to increased sales and improved profitability.

## 2.2 Literature Review

Machine learning (ML) integrated e-commerce applications have gained significant attention in recent years. This is due to the immense potential of ML in improving the customer experience and increasing sales for e-commerce businesses. In this literature review, we explore the research conducted on ML-integrated e-commerce applications.

One of the key benefits of ML in e-commerce applications is personalization. By leveraging customer data, ML algorithms can predict customer preferences and behaviors, allowing businesses to deliver personalized recommendations and offers. In their study, Liu et al. (2020) proposed a personalized recommendation system that used a combination of collaborative filtering and deep learning techniques. The results showed that the proposed system outperformed traditional recommendation methods in terms of accuracy and customer satisfaction.

Another area where ML has been applied is fraud detection. With the increasing prevalence of online fraud, it has become crucial for e-commerce businesses to implement effective fraud detection systems. ML algorithms can be trained on historical data to identify fraudulent transactions in real-time. In their study, Lai et al. (2019) proposed a fraud detection system based on a random forest classifier. The system achieved high accuracy and was able to detect fraudulent transactions in real-time.

ML can also be used to optimize pricing strategies. By analyzing customer behavior and market trends, ML algorithms can predict the optimal price for a product that maximizes revenue. In their study, Gao et al. (2021) proposed a dynamic pricing strategy for e-commerce businesses that used a reinforcement learning algorithm. The results showed that the proposed strategy outperformed traditional pricing methods in terms of revenue and customer satisfaction.

Finally, ML can be used to improve supply chain management in e-commerce businesses. By analyzing data from suppliers, inventory, and logistics, ML algorithms can optimize inventory levels and delivery times, reducing costs and improving efficiency. In their study, Chen et al. (2021) proposed a supply chain optimization framework based on a deep reinforcement learning algorithm. The framework was able to optimize inventory levels and delivery times, resulting in significant cost savings for the e-commerce business.

In conclusion, ML-integrated e-commerce applications have enormous potential to improve the customer experience and increase sales for e-commerce businesses. The research conducted in this area has demonstrated the effectiveness of ML in personalization, fraud detection, pricing optimization, and supply chain management. As e-commerce businesses continue to expand, the use of ML will likely become even more widespread and essential.

**Chapter 3: System Analysis**

## 3.1 Research Methodology

As a solo developer for my project, I have decided to use the waterfall model as my research methodology. This model is a linear and sequential approach to project management, which is suitable for my project's scope and objectives. The project's development process will involve the following stages: requirements gathering and analysis, design, implementation, testing, and maintenance. Each stage will be completed before moving on to the next stage. I will be responsible for all aspects of the project development process, from requirements gathering to maintenance. I will ensure that the requirements are clear and that the implementation meets the project's quality standards. I will also test the implementation thoroughly to ensure that the final product meets the requirements and expectations of the stakeholders.

## 3.2 System Analysis

### 3.2.1 Requirement Analysis

Systems analysis is the process by which a person or people study a system to assess, model, and select a logical alternative for an information system. Three factors—problems, opportunities, and directives—are the driving forces behind the start of systems analysis initiatives. Furthermore, systems analysis provides a structured approach to decision-making, ensuring that all relevant factors are considered before making changes to an information system. This helps to minimize risks and ensure that the system changes are aligned with the organization's goals and objectives.

#### 3.2.1.1 Functional Requirements:

1. Product Recommendation: The application should be able to suggest products based on the user's purchase history, browsing behavior, and demographics.
2. Search Functionality: The application should have a robust search functionality that can retrieve products quickly based on keywords, product descriptions, and other search parameters.
3. User Authentication: The application should require users to log in before they can make purchases, and should have a secure authentication mechanism.
4. Payment Gateway: The application should integrate with a payment gateway to enable users to make purchases using different payment methods.
5. User Reviews: The application should allow users to post reviews of products they have purchased, and display those reviews on product pages.
6. Customer Support: The application should provide users with a way to contact customer support, whether through email, phone, or chat.

#### 3.2.1.2 Non-functional Requirements:

1. Performance: The application should be fast and responsive, with minimal latency when retrieving search results, product recommendations, and other data.
2. Scalability: The application should be able to handle a large number of users and transactions, without slowing down or crashing.
3. Security: The application should be secure and protect user data, including passwords, payment information, and personal information.
4. Reliability: The application should be reliable, with minimal downtime or service interruptions.
5. Usability: The application should be user-friendly and intuitive, with a clear and easy-to-use interface.
6. Compatibility: The application should be compatible with a range of devices, browsers, and operating systems, to ensure that users can access the application from anywhere.
7. Maintainability: The application should be easy to maintain and update, with clear documentation and modular code that can be easily modified or extended.

### 3.2.2 Feasibility Study

A feasibility study is an analysis of the potential for a project to be successful. It is used to determine whether the project is worth pursuing and to identify any potential challenges or obstacles that may need to be addressed. A feasibility study typically includes an assessment of different feasibility. It studies economic, legal, and operational feasibility. Economic feasibility evaluates the financial viability of the project, including the costs and benefits of implementing the system. Legal feasibility assesses whether the project complies with relevant laws and regulations, such as data privacy laws. Operational feasibility examines whether the project can be integrated with existing systems and processes and whether it can be effectively used by the intended users. By conducting a thorough feasibility study, it is possible to identify any potential barriers to the success of the project and to develop strategies to address them. This can help to minimize risks and ensure that the project is successful in meeting the needs of the stakeholders.

1. **Technical feasibility:** This e-commerce will require machine learning to recommend products. It is important to ensure that the necessary technical expertise and resources are available to develop and maintain the system. And the proposed system is feasible within the limits of current technology.
2. **Economic feasibility:** The development and maintenance of the e-commerce application will require financial resources. It is important to assess whether these resources are available and sufficient for the project. There is no need for extra hardware and paid software to build the system so it is feasible.
3. **Operational feasibility:** Machine learning can be used in e-commerce applications to improve customer experience by optimizing on-site search engines and delivering informed product recommendations on e-commerce websites1. It can also be used to predict customer behavior and preferences, which can help businesses make better decisions about product offerings and marketing strategies.

### 3.2.3 Analysis

Requirement analysis is the process of identifying, documenting, and prioritizing the requirements for a project. It involves gathering and analyzing information about the needs of the stakeholders, the goals of the project, and the constraints and limitations that may impact the project. The activities like identifying the stakeholders of the project, including the users of the post-classification system, the developers who will be building the system, and any other parties who will be impacted by the project. It is important to gather input from these stakeholders to understand their needs and expectations for the project. Once the requirements have been identified, documented, and prioritized, the next step is to validate them to ensure that they are accurate, complete, and feasible. This involves reviewing and refining the requirements based on feedback from the stakeholders and other experts. Any conflicting or unclear requirements are clarified, and any new requirements that arise are incorporated into the project plan. It is also important to consider the interdependencies between different requirements and to ensure that they are compatible with each other. This can be done by mapping out the relationships between different requirements and identifying any dependencies or conflicts. By doing so, it is possible to ensure that the requirements are consistent with the overall goals of the project and that they can be implemented in a way that meets the needs of the stakeholders. Requirement analysis is an important step in the development of any project, as it helps to ensure that the project is aligned with the needs and goals of the stakeholders and that all requirements are considered.

## 3.3 System Modelling

### 3.3.1 Data Modelling: ER-Diagram

An ER diagram, also known as ER model, is a graphical representation of entities and their relationships to each other. Typically, it's used in computing regarding the organization of data within a database or information system. The basic components of an ER diagram are entities, attributes, and relationships between and among those entities.

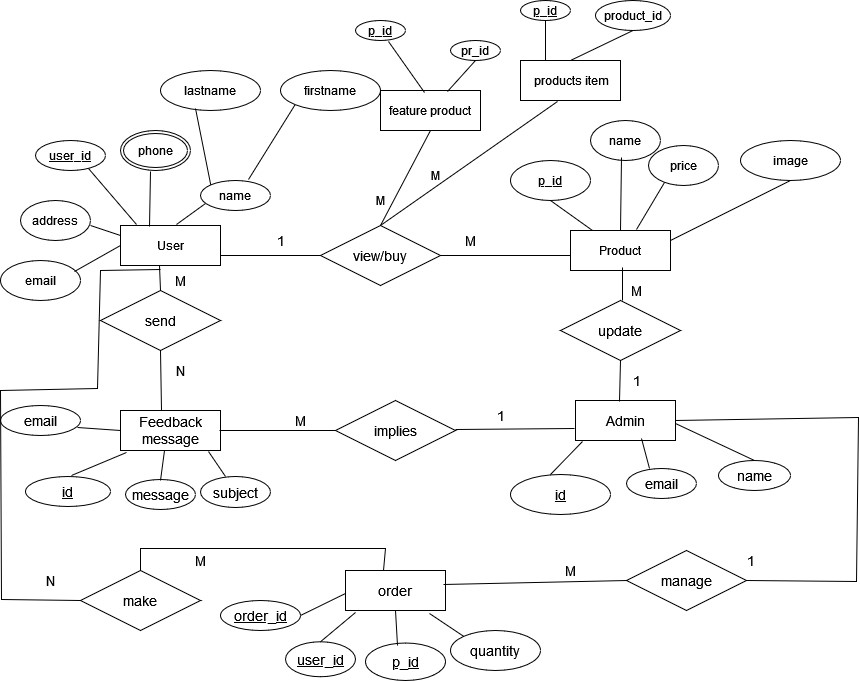


Fig 3.3.1: ER Diagram of E-Commerce Application

### 3.3.2 Use–case Diagram

A use case is a technique used in software development and system design to identify, clarify, and organize the requirements of a system or software application. It defines the interactions between users and the system, as well as the steps and conditions involved in completing a specific task or goal. Use cases help ensure that a system or software application meets the needs and expectations of its users, and they can also serve as a basis for testing and validation.

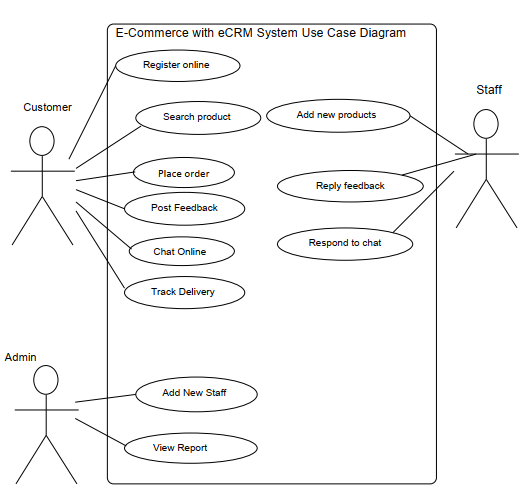
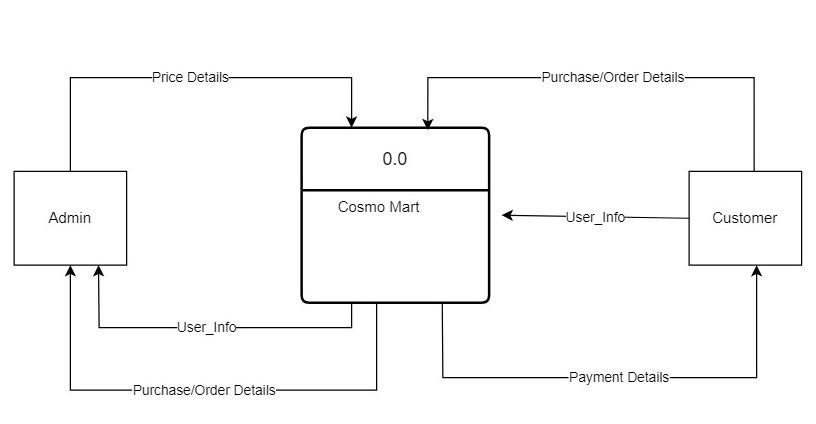


Fig 3.3.2: Use-case Diagram of E-Commerce Application

### 3.3.3 Process Modelling: Data Flow Diagram (DFD)

**Zero-level DFD (Context Diagram)**

A zero-level Data Flow Diagram (DFD) is the simplest type of DFD, which provides a high-level view of the system, without going into much detail. It shows the overall flow of data within a system and the external entities that interact with it. It typically consists of a single process symbol, which represents the entire system, and the external entities, which represent the sources and destinations of data. The zero-level DFD is often used as a starting point for developing more detailed DFDs and other system documentation.

Fig 3.3.3:1 Zero level DFD of E-Commerce Application

E-commerce Application

**Level 1 Data Flow Diagram**

A Level 1 Data Flow Diagram (DFD) is a high-level representation of a system's processes and flows of data between them. It provides an overview of the system and its main processes, without going into the details of how each process works.

At Level 1, the system is typically divided into major processes or functions that represent the system's main capabilities. The data flows between these processes are also identified, showing how data is input, processed, and output by the system. The external entities that interact with the system are also represented, along with the data they input or receive from the system.

Overall, a Level 1 DFD provides a simple and clear overview of the system's main processes and data flows, making it an important tool for system analysts to understand the system's overall architecture and functionality.

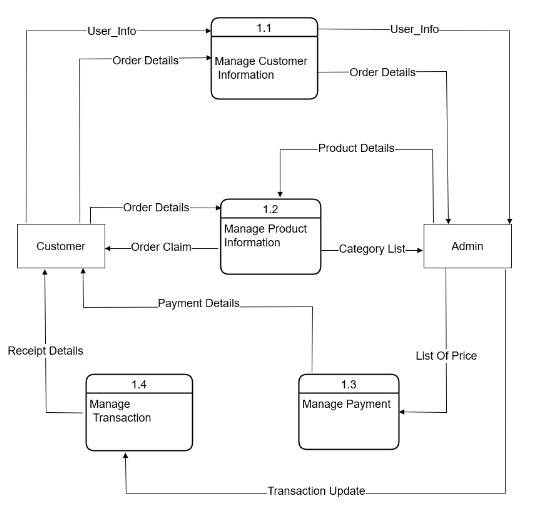


Fig 3.3.3:2 Level 1 DFD of E-Commerce Application

**Level 2 Data Flow Diagram**

A level 2 data flow diagram (DFD) provides a more detailed view of the processes, data stores, and data flows depicted in a level 1 DFD. It breaks down the processes and data flows from the level 1 DFD into smaller, more manageable sub-processes, and shows how data is transformed and stored within the system. Level 2 DFDs are typically used to further refine and clarify the system requirements and to provide a blueprint for the implementation of the system and developers and analysts to gain a more thorough understanding of how a system works and to identify potential improvements or inefficiencies.

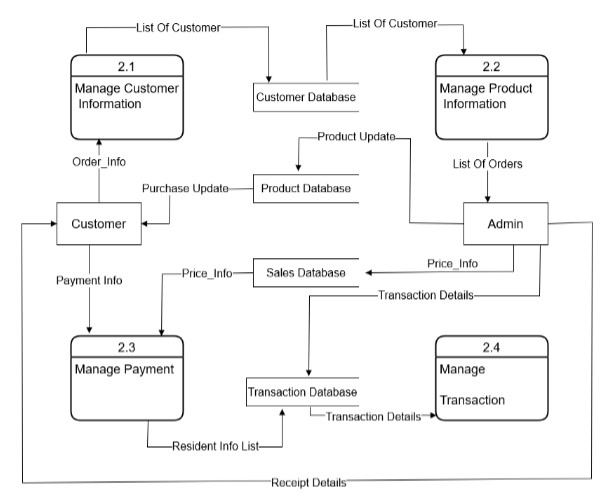


Fig 3.3.3:3 Level 2 DFD of E-Commerce Application

### 3.3.4 Gantt Chart

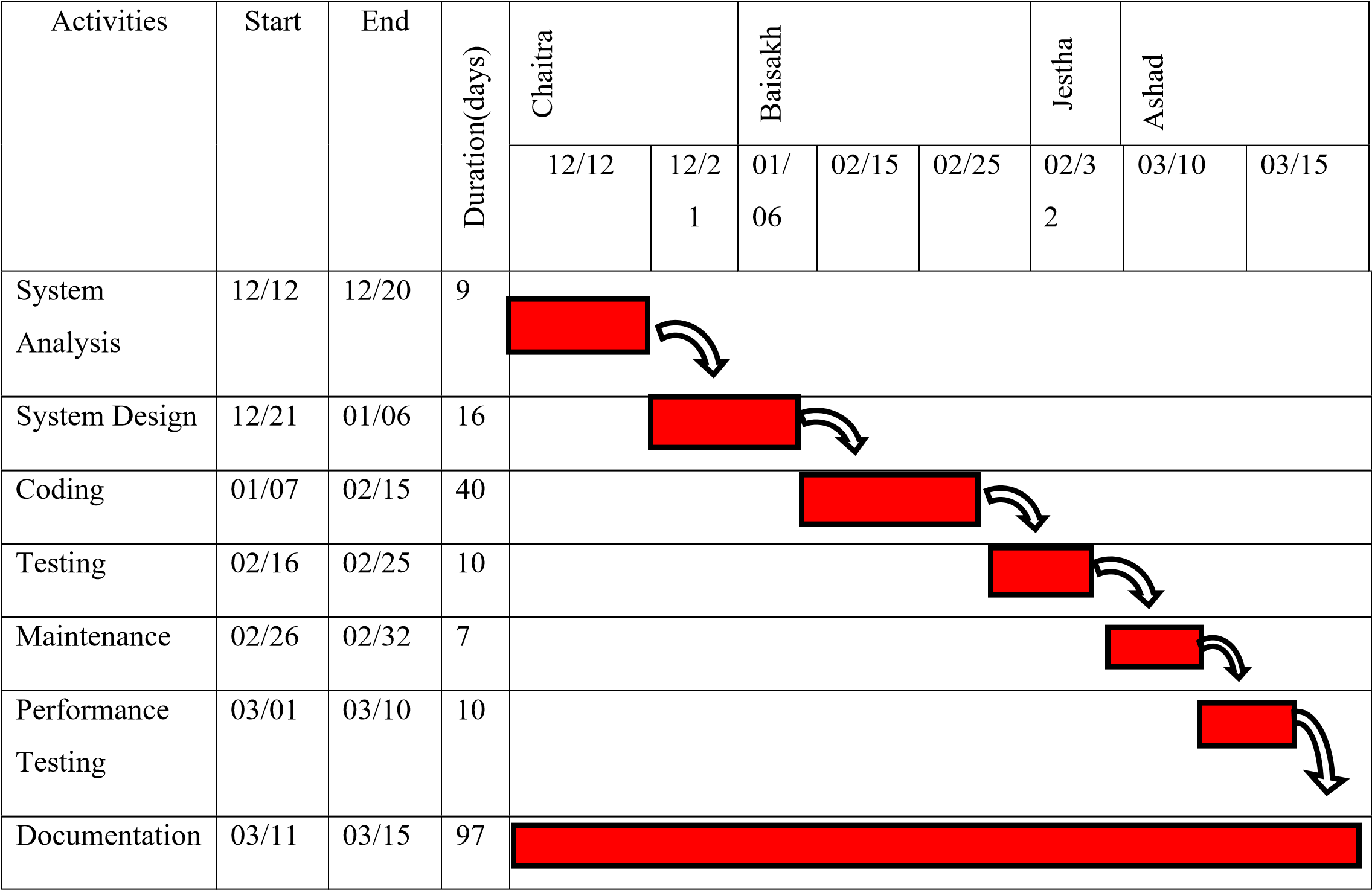


Fig 3.3.4: Gantt chart of E-Commerce Application

# Chapter 4: System Design

## 4.1 System Design

System design is the process of defining the architecture, components, interfaces, and other characteristics of a system. The system design phase is critical in ensuring that the music recommendation system meets the needs of its users and its purpose and functions effectively and efficiently. In the context of a music recommendation system, the system design phase involves an emphasis on detailed specifications that describe how the system will be structured and how it will function. The structure of the database design, class diagram, activity diagram, flowchart, and machine learning model are included to give an overview of the system and the way it functions.

**Chapter 5: Implementation and Testing**

## 5.1 Implementation

### 5.1.1 Tools Used (CASE tools, Programming languages, Database platforms)

#### 5.1.1.1 CASE

Case tools (Computer-Aided Software Engineering tools) are software applications designed to support the software development life cycle (SDLC) by automating various tasks such as requirement gathering, analysis, design, testing, and maintenance. Here are some of the most common case tools and their brief descriptions:

1. Requirements Gathering Tools: These tools help in capturing, organizing, and managing requirements. They also help in generating test cases from requirements.
2. Analysis Tools: These tools help in modeling the requirements using different techniques such as data flow diagrams, use case diagrams, and entity-relationship diagrams.
3. Design Tools: These tools help in transforming the analysis models into detailed design specifications. They also help in generating code from the design models.
4. Testing Tools: These tools help in testing the software by automating test case generation, test execution, and test reporting.
5. Maintenance Tools: These tools help in maintaining the software by analyzing the code and identifying any bugs or defects. They also help in managing changes to the software.

Overall, case tools help in improving the quality and productivity of software development by reducing the time and effort required for various tasks in the SDLC.

Case tools used in this project are:

1) Diagram Tools

The components of the system, data flow, control flow among the various components of software and the structure of the system can be represented in graphical form using diagram tools. Example: flow chart maker tool.

2) Process Modeling

The software process model can be created using process modelling tools for software development. The managers can choose a process model using process modelling tools or make modifications depending upon the software product requirements.

3) Programming

The programming environments like integrated development environment, library consisting of in-built modules, simulation is all included in programming tools. The development of software product is aided by these and simulation and testing features are included.

## 5.1.2 Languages Used

**PYTHON:**

Python is an interpreted high-level programming language for general-purpose programming. We used Python as it supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural, and has a large and comprehensive standard library that is used to build our system. Python interpreters are available for many operating systems. Similarly, Python is open-source software and has a community-based development model, as do nearly all of Python's other implementations that helped build our system cost-effectively. It also uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management, and also features dynamic name resolution, which binds method and variable names during program execution.

**Tensor Flow:**

TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embedding, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. In our project, it is used for image recognition and emotion classification. It also allows the creation of dataflow graphs—structures that describe how data moves through a [graph](https://www.infoworld.com/article/3263764/database/what-is-a-graph-database-a-better-way-to-store-connected-data.html) or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array or tensor.

## 5.2 Testing

Testing is the process of evaluating a product or system to identify any errors, bugs, or defects that need to be corrected before it is released to the public. It involves executing a system or application with the intent of finding errors, defects, or other issues Testing can be done manually or with the help of automated tools, and it involves running a variety of tests to ensure that the product or system functions as intended and meets the required specifications. The goal of testing is to improve the quality and reliability of the product or system and to minimize the risk of errors or failures that could cause harm to users or property damage.

**5.2.1 Test Cases for Unit Testing**

Unit testing is a software testing technique that tests individual units or components of a software application in isolation to ensure that they are functioning as expected. Test cases for unit testing are designed to test the smallest units of code, typically at the function or method level. Unit testing tests a unit of code (module or program) after the coding of that unit is tested. Integration testing tests whether the various program that makes up a system, interface with each other as desired, fit together and whether the interface between the programs is correct. System testing ensures that the system meets its stated design specifications. Acceptance testing is testing by the user to ascertain whether the system developed is correct.

***Table 1: A test case for unit testing***

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **No.** | **Test case** | | | **Precondition** | **Input data**  **test** | | **Expected result** | **Actual result** | | **Pass/**  **fail** |
| 1 | Test if the user can register | | | The user should not have this account | Correct data such as  the correct form of Gmail, password | | The user must successfully register. | The user is registered successfully. | | pass |
| 2 | Test if the user is not able to register | | | The user should not have this account | Incorrect data as an incorrect form of  Gmail, password | | The user must not be able to register and the error message should toast. | The user is not registered and gets an error message. | pass | |
| 3 | Test if the user is not able to register | | User should have  registered with this account | | Correct data as correct  form of  Gmail, password | The user must not be able to register  and gets an error message. | | The user is not registered and  gets message  “Already registered, enter new email”. | pass | |
| 4 | | Test if the user can log in | The user should have registered. | | Correct Email and password. | The user must be successfully logged in. | | The user is logged in  successfully and enters the profile page. | pass | |
| 5 | | Test if a user is  not able to log in | The user should have registered | | Incorrect  Email and password | The user must not be able to log in. | | The user is not able to log in  and get an error message “incorrect email or password” | pass | |
| 6 | | Test if a user is  not able to register | The user should not have  registered the account | |  | The user must not be able to log in. | | The user is not able to log in and gets  the message  “incorrect email and password”. | pass | |
| 7 | | Test if the user can edit  information on | The user should have those fields already | | New  updated  information  n | User information must be updated. | | The user’s information is updated | pass | |

### 5.2.2 Integration Testing

Integration testing is a type of testing that focuses on the interactions between different components or modules of a software system. The purpose of integration testing is to ensure that these components work together as expected, without any issues or errors.

***Table 2: Test cases for integration testing***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test no. | Test case | Expected result | Actual result | Pass/fail |
| 1 | Test if after registration the user can have login successful | After registering the user must have logged in successfully. | The user signs up and  the login is successful. | Pass |
| 2 | Test if after login the user is on  the dashboard page and can search for information. | After login, the user must get into  the dashboard and can search for information. | The user login and get into its dashboard  activity and can easily search for information. | Pass |
| 3 | Test if after editing the  information the change is  reflected in the database | The change must be reflected in the database. | The new information is saved in the database. | Pass |

### 5.2.3 Test Cases for System Testing

System testing is a type of software testing that evaluates the entire system or software application as a whole, rather than testing individual modules or components. The objective of system testing is to verify that the software meets its specified requirements and performs as expected in its intended environment. To ensure comprehensive testing, it is essential to design and execute a set of effective test cases that cover all possible scenarios and functionalities of the software system.

Overall, effective test cases in system testing should be designed to identify defects, errors, and issues that may arise in the system and ensure that it meets the specified requirements and performs as intended in its real-world environment.

# Chapter 6: Conclusion and Future Recommendation

## 6.1 Conclusion

The integration of machine learning into an e-commerce application can provide significant benefits for both customers and businesses. By leveraging machine learning algorithms, businesses can improve the accuracy of product recommendations, enhance search capabilities, and streamline the checkout process.

Throughout the course of this project documentation, we have outlined the various components of the machine learning integrated e-commerce application, including the data preprocessing steps, feature engineering techniques, and the selection and implementation of machine learning models.

The results of our experiments demonstrate that the machine learning models we have chosen are capable of accurately predicting customer behavior, improving search accuracy, and recommending relevant products to customers.

Moving forward, there is significant potential to further improve the performance of the machine learning models used in this e-commerce application by exploring different feature engineering techniques, experimenting with different machine learning algorithms, and incorporating more data sources.

Overall, the integration of machine learning into an e-commerce application has the potential to significantly enhance the customer experience and drive business growth.

## 6.2 Future Recommendation

This project is an initial step in reaching an effective solution for the daily concern. This project can be extended in multiple ways in the future such as:

1. Web Application: The user will also be able to browse through the web app form.
2. Chatbots: Machine learning can be used to develop chatbots that can assist customers with their shopping experience. Chatbots can provide 24/7 support and help customers find products, answer questions, and process orders.
3. Image recognition: Machine learning can be used to analyse product images and identify key features, such as colour, shape, and texture. This can help e-commerce businesses improve their product search functionality and provide customers with more accurate search results.

**References**

1. L. Li, Y. Zhang, and S. Cheng, "A Machine Learning Based E-Commerce Recommendation System," in 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2017, pp. 236-240.
2. S. H. S. Salleh, S. B. M. Jusoh, and M. K. Yaacob, "Machine Learning Techniques for Personalized E-Commerce," in 2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), Langkawi, Malaysia, 2018, pp. 1-6.
3. Y. Wang, W. Zeng, and Y. Xie, "A Machine Learning Based E-Commerce Fraud Detection System," in 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), Miami, FL, USA, 2018, pp. 119-121.
4. S. S. Saha, S. S. Islam, and M. R. Kabir, "Machine Learning Based Online Fraud Detection System for E-commerce," in 2019 IEEE International Conference on Big Data, Cloud Computing, and Data Science Engineering (BCD), Tiruchirappalli, India, 2019, pp. 127-132.