

**Tribhuvan University**

Faculty of Computer Science & Information Technology

Kirtipur, Kathmandu

**A FINAL YEAR PROJECT REPORT**

**ON**

**“Dine Delight- A restaurant recommendation portal”**

For the partial fulfillment of the requirements for the Bachelor’s Degree in

Computer Science and Information Technology under Tribhuvan University

**Submitted by**

Ishan Bhusal (24925/076)

**Submitted to**

**Chitwan College of Technology**

Department of IT

Affiliated to Tribhuvan University

Bharatpur, Chitwan

**December, 2023**

# **COPYRIGHT**

The author has agreed that the Library and Department of Computer Science and Information Technology of Chitwan College of Technology may make this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisor who supervised the project work recorded here in, or in their absence, by the Head of Department where in the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Computer Science and Information Technology, Chitwan College of Technology in any use of the material of this project report. Copying or publication or the other use of this report for the financial gain without approval of to the Department of Computer Science and Information Technology, Chitwan College of Technology and author's written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of Department

Department of Computer Science and Information Technology

Chitwan College of Technology, Bharatpur-12, Asthachok, Chitwan

# **DECLARATION**

We, the undersigned solemnly declare that the report of the project work entitled " **Dine Delight - A restaurant recommendation portal** " submitted to office of the Dean, Faculty of Science and Information Technology, Tribhuvan University, is based on our work carried out during the course of study under the supervision and guidance of Mr. Ravi Tiwari. We assure that the statements made and conclusions drawn are an outcome of the project work. We further declare that, to the best of our knowledge and belief that the project report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University or any other University.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ishan Bhusal

(Symbol no: 24925/076)

# **SUPERVISOR’S RECOMMENDATION**

I hereby recommend that this project prepared under my supervision by **Mr. Anurag Ghimire**, **Mr. Sagar Thapa**, and **Mr. Bidhan Chaudhary** entitled “Homestore” subjected to online shopping in partial fulfillment of the requirements for the degree of four years Bachelor Degree in Bachelors of Computer Science in Computer Science and Information Technology be processed for the evaluation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Ravi Tiwari

Project Supervisor

Admin Head

Chitwan College of Technology

Bharatpur, Chitwan, Nepal

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **ACKNOWLEDGEMENT**

The successful completion of this project would not have been possible without the support and guidance of our teachers, colleagues, and project team, and we are grateful for their contributions throughout this project. We would like to extend our deepest appreciation to each of them.

We are highly indebted to **Mr. Ravi Tiwari**, our project supervisor, for his constant guidance, supervision, and unwavering support. Without his invaluable input and encouragement, this project would not have been possible. Additionally, we appreciate the efforts of **Mr. Ram Binay Gupta**, without his assistance, the project would not have been close to being finished.

We would also like to express our gratitude to **Chitwan College of Technology** for providing us with all the necessary infrastructure, resources, and a supportive environment to carry out our project. Our heartfelt thanks go out to all the staff and faculty members of the BSc.CSIT department for their constant support, encouragement, and timely assistance throughout the project.

Lastly, we would like to extend our appreciation to all our colleagues and friends who provided us with their invaluable insights, feedback, and encouragement in developing this project. Their unwavering support and enthusiasm helped us successfully complete this project.

# **ABSTRACT**

The project's main goal is to develop an online shopping store where users can search for and make purchases of products based on title, author, and subject. The user's chosen products are displayed in a tabular format, and they can order them online. Instead of going to a physical store and wasting time, the user can purchase products on this website.

An online shopping store is a web application that allows customers to buy products online. Customers can search for a products by title or category using a web browser, add it to their shopping cart, and then purchase it. The user can log in using his or her account information, or new customers can swiftly create an account. They should include their full name, phone number, and shipping address. A user can also provide feedback to a products by rating it on a scale of one to five. The products are organized into several categories based on the subject matter, such as Price and Categories.

Customers can shop online for products at the Online Shopping Store using a web browser. A customer can create an account, sign in, add items to a shopping cart, and make purchases. When compared to a regular user, the Administrator will have more features. The Administrator can add, delete, and edit product’s data, and member information, as well as confirm an order that has been placed.

# **TABLE OF CONTENT**

[**COPYRIGHT** i](#_Toc154407969)

[**DECLARATION** ii](#_Toc154407970)

[**SUPERVISOR’S RECOMMENDATION** iii](#_Toc154407971)

[**ACKNOWLEDGEMENT** iv](#_Toc154407972)

[**ABSTRACT** v](#_Toc154407973)

[**TABLE OF CONTENT** vi](#_Toc154407974)

[**LIST OF FIGURES** viii](#_Toc154407975)

[**LIST OF TABLES** ix](#_Toc154407976)

[**LIST OF ABBREVIATIONS** x](#_Toc154407977)

[**CHAPTER 1: INTRODUCTION** 1](#_Toc154407978)

[**1.1.** **Introduction** 1](#_Toc154407979)

[**1.2.** **Problem Statement** 1](#_Toc154407980)

[**1.3.** **Objectives** 1](#_Toc154407981)

[**1.4.** **Scope and Limitation** 2](#_Toc154407982)

[**1.5.** **Development Methodology** 3](#_Toc154407983)

[**Figure 1: 1.1 Agile Development Methodology** 3](#_Toc154407984)

[**1.6.** **Report organization** 4](#_Toc154407985)

[**CHAPTER 2: BACKGROUND STUDY & LITERATURE REVIEW** 6](#_Toc154407986)

[**2.1. Background Study** 6](#_Toc154407987)

[**2.2. Literature Review** 7](#_Toc154407988)

[**CHAPTER 3: SYSTEM ANALYSIS & DESIGN** 8](#_Toc154407989)

[**3.1. System Analysis** 8](#_Toc154407990)

[**3.1.1. Requirement Analysis** 8](#_Toc154407991)

[**Figure 2: 3.1.1.1. High Level Diagram** 9](#_Toc154407992)

[**Figure 3: 3.1.1.2. Use Case Diagram**  10](#_Toc154407993)

[**3.1.2. Feasibility Study** 11](#_Toc154407996)

[**Table 1: 3.1. Gantt Chart** 13](#_Toc154407997)

[**3.1.3 Analysis** 14](#_Toc154407998)

[**Fig 4: 3.2.2. System Architecture** 14](#_Toc154407999)

[**Figure 5: 3.2.5. ER Diagram** 15](#_Toc154408000)

[**Figure 6: 3.2.3. Flowchart** 16](#_Toc154408001)

[**Figure 7: Context Diagram** 17](#_Toc154408002)

[**Figure 8: Level 1 Data Flow Diagram** 18](#_Toc154408003)

[**CHAPTER 4: SYSTEM DESIGN** 21](#_Toc154408005)

[**4.1. Design** 21](#_Toc154408006)

[**4.2. Algorithm Details** 21](#_Toc154408007)

[**CHAPTER 5: IMPLEMENTATION AND TESTING** 23](#_Toc154408008)

[**5.1** **Implementation** 23](#_Toc154408009)

[**5.1.1. Tools Used** 23](#_Toc154408010)

[**5.1.2. Implementation Details of Modules** 25](#_Toc154408011)

[**5.2. Testing** 26](#_Toc154408012)

[**5.2.1. Unit Testing** 26](#_Toc154408013)

[**Table 2: 5.2.1. Unit Testing** 26](#_Toc154408014)

[**5.2.2. Integration Testing** 28](#_Toc154408015)

[**Table 3:5.2.2. Integration Testing** 29](#_Toc154408016)

[**5.2.3. System Testing** 31](#_Toc154408017)

[**Figure 12: 5.2.3. System Testing** 31](#_Toc154408018)

[**5.3 Result Analysis** 31](#_Toc154408019)

[**CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATIONS** 33](#_Toc154408020)

[**6.1. Lesson learned** 33](#_Toc154408021)

[**6.2. Conclusion** 33](#_Toc154408022)

[**6.3. Future Recommendations** 33](#_Toc154408023)

[**REFERENCES** 35](#_Toc154408024)

[**Appendix** 36](#_Toc154408025)

[**Appendix A: Snapshots** 36](#_Toc154408026)

[**Appendix B: Source Code** 40](#_Toc154408027)

# **LIST OF FIGURES**

[Figure 1: 1.1 Agile Development Methodology 3](#_Toc135324802)

[Figure 2: 3.1.1.1. High Level Diagram 9](#_Toc135324810)

[Figure 3: 3.1.1.2. Use Case Diagram of Admin 10](#_Toc135324811)

[Figure 4: 3.1.1.2. Use Case Diagram 11](#_Toc135324812)

[Figure 5: 3.2.5. ER Diagram 17](#_Toc135324818)

[Figure 6: 3.2.3. Flowchart 18](#_Toc135324819)

[Figure 7: Context Diagram 19](#_Toc135324820)

[Figure 8: Level 1 Data Flow Diagram 20](#_Toc135324821)

[Figure 9: Level 2 Data Flow Diagram 21](#_Toc135324822)

[Figure 10: 5.2.3. System Testing 31](#_Toc135324836)

# **LIST OF TABLES**

[Table 1: 3.1. Gantt Chart 15](#_Toc135324815)

[Table 2: 5.2.1. Unit Testing 27](#_Toc135324832)

[Table 3:5.2.2. Integration Testing 30](#_Toc135324834)

# **LIST OF ABBREVIATIONS**

E-Commerce Electronic Commerce

E-Catalogue Electronic Catalogue

DFD Data Flow Diagram

ERD Entity Relationship Diagram

HTML Hyper Text Markup Language

CSS Cascading Style Sheet

JS JavaScript

DB Database

VS Code Visual Studio Code

UI User Interface

# **CHAPTER 1: INTRODUCTION**

## **Introduction**

In today's digital era, the rapid advancement of technology has transformed various industries, including the publishing and products selling sector. With the increasing popularity of ecommerce and the convenience of online shopping, traditional physical stores are facing new challenges. As a result, the emergence of online product selling websites has revolutionized the way items are bought and sold, offering a convenient and efficient platform for buyers. This project report aims to present a comprehensive analysis of the development and implementation of an online product selling website. The objective of this project is to create a user friendly, secure, and feature rich platform that connects product enthusiasts with a vast collection of products, allowing them to browse, purchase, and engage with their favorite categories. By undertaking this project and developing an online product selling website, we aim to contribute to the growth and evolution of the digital market, enabling buyers to access a vast collection of products with ease and convenience. The project report serves as a comprehensive documentation of the development process, guiding future endeavors and inspiring innovation in the field of online selling. The Home Store project proposal is a significant step towards achieving Nepal development goals especially in the field of education. This project will help to improve the literacy and increase access to quality material.

## **Problem Statement**

E-commerce provides fast and easy platform to perform online transaction. However, there are a lot of competition among multiple e-commerce sites when users land on an e-commerce site. It is very difficult for users to find what they are looking for because of bad UI/UX, difficulty in user navigation, lack of authorization, lack of verification and validation and so on. It is not only difficult for users or individuals but also for business organization because of competitive environment and agility challenge. Some common problems for users are like unprofessional and outdated design, fake product reviews, poor on site engine, no categories suggestions, payment failures, unclear product information, lack of security and so on.

Following listed points highlights the problems in e-commerce sites;

* 1. Lack of professional and updated design
  2. Lack of easier checkout process
  3. Unavailability of category suggestions
  4. Lack of support and AI chat

**1.3 Objectives**

The objectives in e-commerce might differ based on business but generally they comprise customer satisfaction, easy checkout process, personalization of contents, availability of category suggestions and support and live chat. The objective of e-commerce is to define goals, guide elements of decision making, identify conflicting activities and ensure accountability.

Few of the top objectives of the project are as listed below;

a. Professional and updated design

1. Easier checkout process

c. Category suggestions and personalization

d. Availability of support and live chat

.

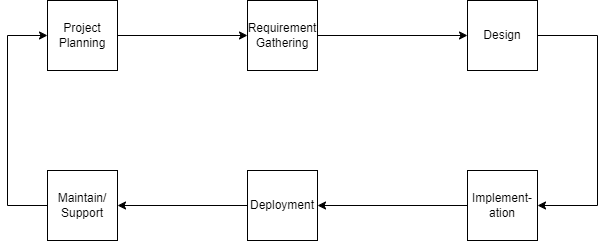
## **Scope and Limitation**

The scope of the Homestore project encompasses the development of a web-based platform that connects clients with proper buyers. The platform will allow clients to search for products based on their names, specialties, and categories. Client reviews and a verification process for admins will help improve transparency in online shopping store. The platform will also have an admin panel, where administrators can verify or unverified users and send notifications to admin.

The limitations of the Homestore project include the fact that it only provides access to buyers who are registered on the platform. The platform may not have a comprehensive list of all available products, and the quality of services provided by admin on the platform may vary. Furthermore, while the platform aims to improve transparency in the online shopping platform.

**1.5 Development Methodology**

The development methodology for Homestore will follow an iterative and incremental approach, with an emphasis on Agile development principles. The following steps will be followed:



## **Figure 1: 1.1 Agile Development Methodology**

**Project Planning:**

The project planning phase will involve defining project scope, goals, and objectives. The development team will create a product roadmap that outlines the key milestones and deliverables.

**Requirements Gathering:**

The requirements gathering phase will involve engaging with stakeholders to understand their needs and requirements. The development team will create user stories, use cases, and functional specifications to document the requirements.

**Design:**

The design phase will involve creating high-level and low-level designs of the system. The development team will create architecture diagrams, database schemas, and UI mockups.

**Implementation:**

The implementation phase will involve coding, testing, and integrating different components of the system. The development team will follow Agile development practices such as continuous integration, automated testing, and code reviews.

**Deployment:**

The deployment phase will involve deploying the system to a production environment. The development team will work closely with the operations team to ensure that the system is deployed and configured correctly.

**Maintenance and Support:**

The maintenance and support phase will involve monitoring and maintaining the system after deployment. The development team will provide ongoing support to users, fix bugs, and make enhancements to the system based on feedback from users.

Throughout the development process, the development team will hold regular meetings to review progress, identify issues, and adjust, as necessary. The development team will also prioritize features based on user feedback and business needs, ensuring that the most important features are developed first.

## **Report organization**

Chapter plan regarding our project is structured as specified in the format below:

**Chapter 1: Introduction**

This chapter consists of a brief introduction to our Project. This chapter also discusses the problem definition, the objectives of the project, its scope and limitation.

**Chapter 2: Background Study & Literature Overview**

This chapter includes a review of the area being researched, current information surrounding the issue, previous studies on the issue, and relevant history on the issue. This chapter focuses on the study of existing system and Projects from various sources such as internet blogs, website, project work reports, products, and journals.

**Chapter 3: System Analysis**

This chapter includes studying a procedure or business to identify its goals. and purposes and create systems and procedures that will achieve them in an efficient way. It also explains the requirements specification and feasibility study conducted during project initiation. This chapter consists of Data Flow Diagram (DFD), Database Schema, Flow chart, Sequence Diagram.

**Chapter 4: System Design**

This chapter consists of High-Level Design, Data Flow Diagram (DFD), Use Case Diagram., that describe the whole system.

**Chapter 5: Implementation and Testing**

This chapter focuses on the Software, Tools, Protocol used for the initiation and completion of our Project. Also, it defines various testing of the System.

**Chapter 6: Conclusion and Future recommendations**

This section depicts the overall conclusion of the project and future upgrades provided to the clients.

**References**

This section lists references of journals, conferences, products, websites, and other sources cited throughout the project.

**Appendix**

This section comprises a collection of various screenshots, source codes, and configuration files that provide a detailed insight into the working mechanism of our project. The screenshots help illustrate the various stages and functionalities of the project.

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

## **2.1. Background Study**

The Internet has been favored by more and more people for its high efficiency and richness, and ecommerce has emerged. The online store is a form of ecommerce and product sales industry in one form, it has many advantages, such as: store size is relatively small, cost savings; transaction activities can be anytime, anywhere, improve service efficiency, The information is complete, more convenient retrieval, the new product information on the new, consumers can see in a timely manner, trading activities can be launched immediately, so online product store in today's era of development is extremely rapid. Online store system is the main function of the trading platform for the site, consumers can connect to the Internet through the computer into the online store and then check the product information, if you need to purchase should be registered landing, select their own products, submit orders and pay Operation to complete the entire product ordering process, to achieve online transactions.

## **2.2. Literature Review**

When we use Google to search for e-commerce websites and applications, we will find a lot of options. However, there is some ambiguity in selecting suitable content at the proper time. Some websites have been developed that contain products and other types of content. Electronic Commerce (ecommerce) applications allow multiple parties involved in a commerce transaction to connect in order to shop for new, secondhand, rare, and out of print products. Review of literature for an online product store. The prototype serves as a roadmap for establishing a solid Online Products Ordering System based on user feedback, notably from the perspective of business, which will be handled by the products store. The arranging, however, is still done manually. According to a software development firm, an E catalogue delivers vital information about product specifications to potential customers. It makes it easier for potential customers to find the items they want in the format they want. It is the ideal and ideal product catalogue because it does not require printing, has no number or color restrictions, and does not require distribution.

# **CHAPTER 3: SYSTEM ANALYSIS & DESIGN**

## **3.1. System Analysis**

It includes collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted to study a system or its parts to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. It consists of requirement analysis and feasibility study for the system.

### **3.1.1. Requirement Analysis**

Requirement Analysis, also known as Requirement Engineering, includes defining user expectations for a new software being built or modified. In software engineering, it is sometimes referred to loosely by names such as requirements gathering or requirements capturing. It includes functional and non-functional requirements for the system.

#### **i. Functional Requirements**

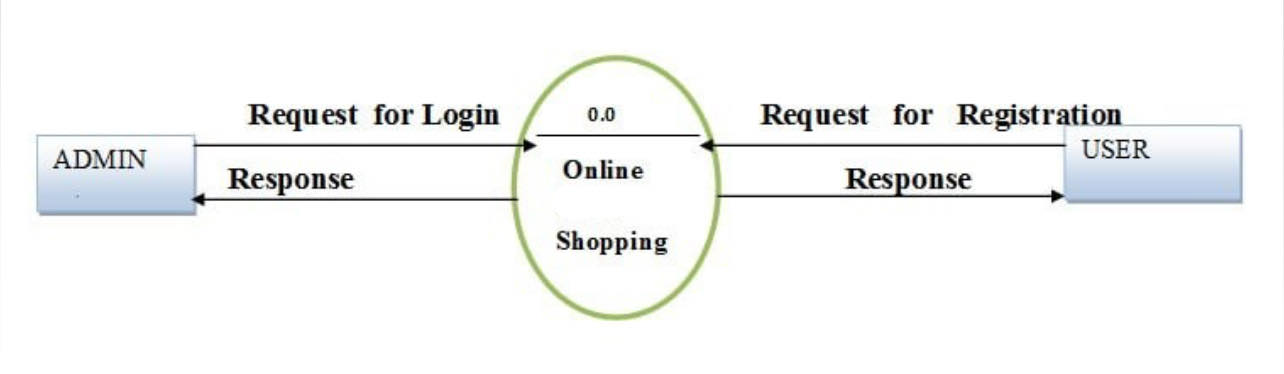
To ensure that the Homestore platform meets the needs and expectations of its users, a comprehensive set of functional requirements was identified and developed. These requirements were informed by the system analysis results and feedback from potential users and stakeholders.

The functional requirements for the Homestore platform can be grouped into several key areas, including user account management, appointment scheduling, video call and chat functionality, payment processing, and administrative capabilities.

1. **User Account Management:** The platform must allow for the creation and management of both client and lawyer accounts, with client accounts not requiring verification while lawyer accounts must be verified by platform administrators. The platform must also include the ability to sign up using a Google account.
2. **Searching:** Clients must be able to search for and connect with products based on their name, categories, or prices, and admins must be able to accept or reject transaction requests from clients.
3. **AIChat Functionality:** The platform must provide AIchat capabilities to facilitate communication between buyers and admin during transactions.
4. **Administrative Capabilities:** The platform must include a robust administrative system that allows for the verification of buyers and management of the platform. This includes the ability to send notifications to admin and buyers and provide robust reporting and analytics capabilities to help better understand usage patterns and user behavior.

To ensure the Homestore platform meets the needs of its users and provides a comprehensive solution for connecting admins with buyers, these functional requirements were carefully developed and will serve as the basis for the design and development of the platform.

#### **3.1.1.1. High Level Diagram**

****

## **Figure 2: 3.1.1.1. High Level Diagram**

**3.1.1.2. Use Case Diagram**

## **Figure 3: 3.1.1.2. Use Case Diagram of Admin**

#### **ii. Non-Functional Requirements**

There are a number of non-functional requirements that must be met in order for the Homestore platform to be successful. These non-functional requirements relate to the overall performance, reliability, and usability of the platform, and are essential for ensuring that users have a positive experience when using the platform.

1. **Performance:** The platform must be fast and responsive, with minimal lag or delay when accessing or using any of its features.
2. **Reliability:** The platform must be highly reliable, with a high degree of uptime and minimal downtime. The platform should also be able to handle a high volume of traffic without slowing down or crashing.
3. **Scalability:** The platform must be designed with scalability in mind and should be able to accommodate growth and increased usage over time.
4. **Security:** The platform must include robust security measures to protect user data and ensure the confidentiality and privacy of user information.
5. **Usability:** The platform must be easy to use and navigate, with a user-friendly interface that makes it simple for users to find and access the information and features they need.
6. **Accessibility:** The platform must be accessible to users with a range of abilities and disabilities, including those with visual, auditory, and motor impairments.

To ensure the success of the Homestore platform, these non-functional requirements were identified and will be incorporated into the design and development of the platform, to ensure that users have a positive and seamless experience when using the platform.

### **3.1.2. Feasibility Study**

A feasibility study is a crucial part of the project planning process for Homestore, as it helps to determine if the platform is viable and practical to develop.

#### **i. Technical Feasibility**

The technical feasibility of Homestore was determined through the evaluation of its proposed technology stack, which includes React JS as the frontend framework, Material UI for styling, Node JS for the backend, and Firebase for database integration. These cutting-edge technologies ensure the system will be highly performant, scalable, and secure, allowing it to meet the project's functional and non-functional requirements. Also, we have extensive experience in developing web applications, further increasing the project's feasibility from a technical perspective.

#### **ii. Operational Feasibility**

The operational feasibility of Homestore was evaluated by considering factors such as user acceptance, ease of use, compatibility with current systems and processes, and resource availability. We conducted surveys and focus groups with lawyers and clients to gather feedback on the proposed system and understand their needs and requirements. Based on the results of these assessments, it was determined that Homestore will be well-received by its target users as it provides a convenient and efficient way for lawyers and clients to communicate and collaborate. Furthermore, Homestore simple and user-friendly interface, coupled with its seamless integration with existing systems and processes, makes it an ideal solution for lawyers and clients. The operational feasibility of the project is therefore considered high.

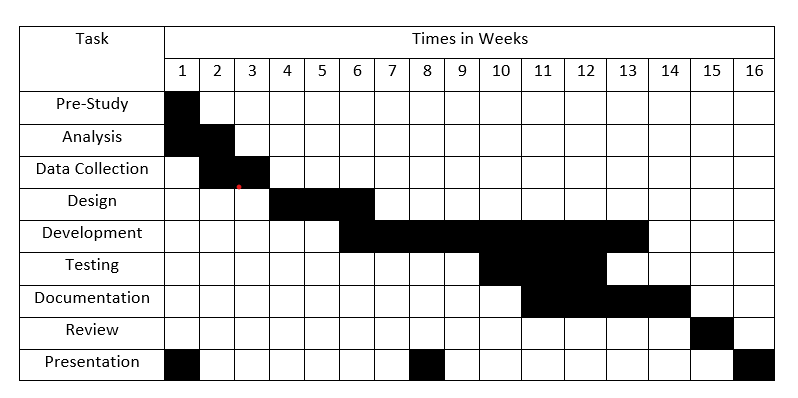
#### **iii. Economic Feasibility**

We conducted a thorough cost-benefit analysis to determine the potential financial impact of the project. This analysis included factors such as development costs, ongoing maintenance costs, potential revenue streams, and any other costs associated with the implementation of Homestore. The results of this analysis indicated that the economic feasibility of Homestore is high, with the potential to provide significant cost savings and revenue opportunities for clients. With its innovative approach to connecting buyers and sellers, Homestore is poised to play an important role in the bussiness industry, providing a valuable and cost-effective solution to the challenges faced by both parties.

#### **iv. Schedule Feasibility**

The schedule feasibility analysis accumulated to be 16 weeks (about 3 and a half months) for the project's completion. Multiple milestones and minutes were created to keep track of time. The following Gantt Chart shows the time allocated for the design completion:

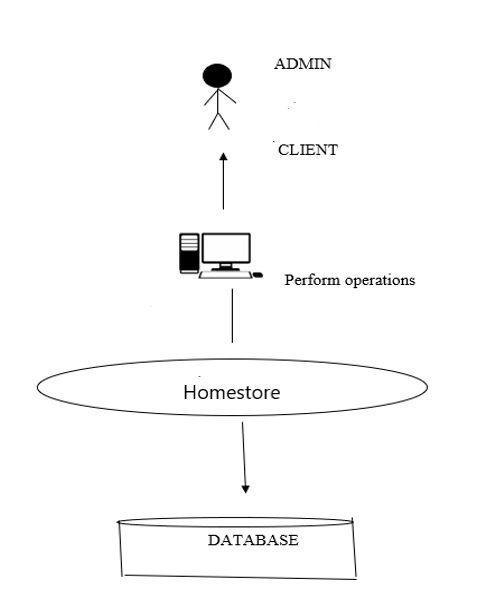
## **Table 1: 3.1. Gantt Chart**

****

### **3.1.3 Analysis**

**3.1.3.1 Structured approach Data modeling**

We have developed a system that makes use of the computer peripherals. This system is supported by many different devices that have browsers. The system architecture is clearly defined by the following figure:



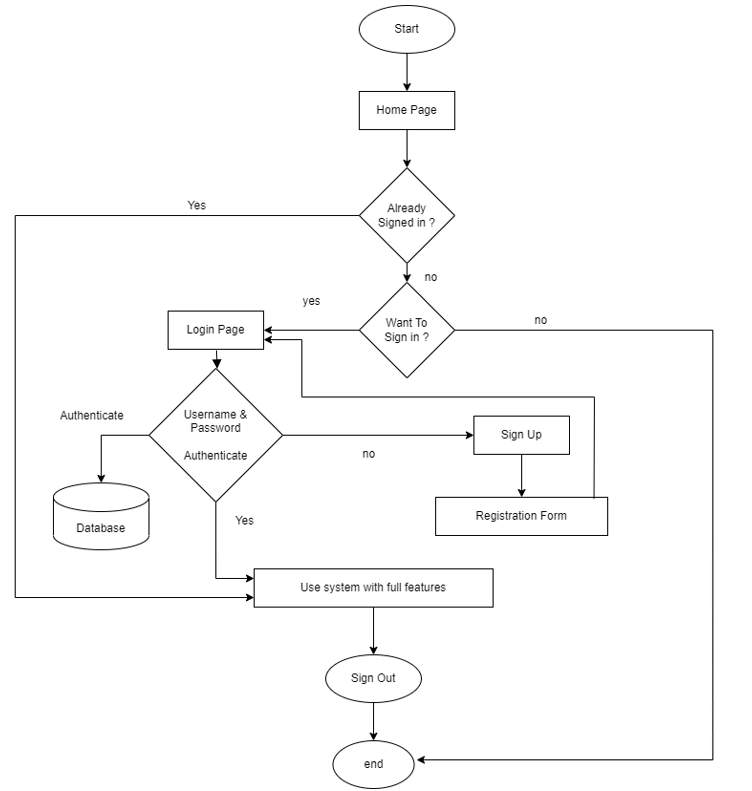
## **Fig 6: 3.2.2. System Architecture**

* **ER Diagram**



## 

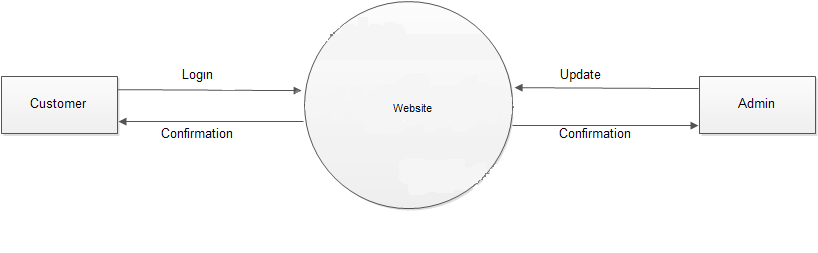
## **Figure 7: 3.2.5. ER Diagram**

* **Flow Chart**

## **Figure 8: 3.2.3. Flowchart**

* **DFD /Context Diagram**

DFD level 0 is also known as context diagram. ￼



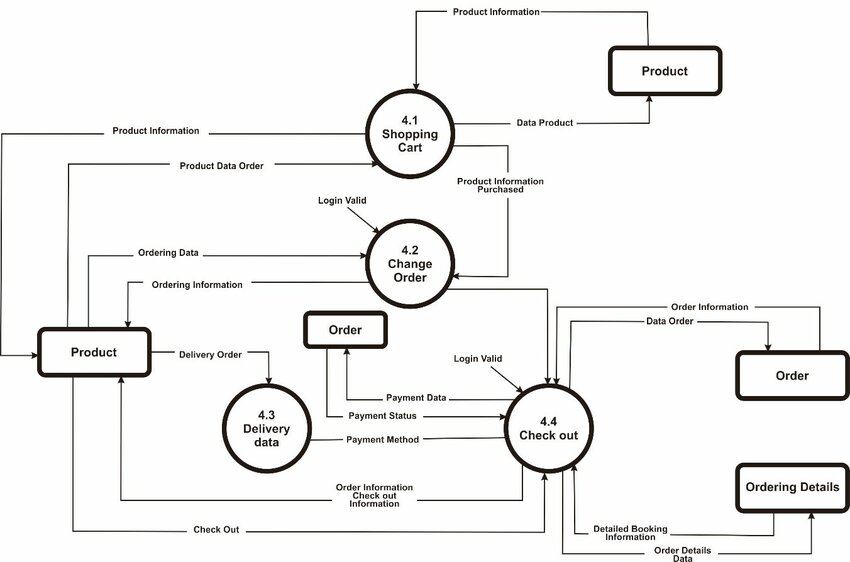
## **Figure 9: Context Diagram**

* **Level 1 Data Flow Diagram**

****

## **Figure 10: Level 1 Data Flow Diagram**

* **Level 2 Data Flow Diagram**

****

## **Figure 11: Level 2 Data Flow Diagram**

# **CHAPTER 4: SYSTEM DESIGN**

## **4.1. Design**

The Homestore system is built using a client-server architecture. The client will be a web application that can be accessed through a web browser, and the server will be responsible for processing user requests and providing responses. The system will consist of the following components:

**1. User-side:** The User-side will consist of the following components:

* Web Browser: The web browser will be used by the users to access the Homestore web application. Our system will work on browsers like Firefox, Google Chrome, Microsoft Edge, Opera browser, Brave etc.
* User Interface: The user interface will be responsible for displaying the information and allowing users to interact with the system.

**2. Server-side:** The server-side will consist of the following components:

* Web Server: The web server will host the web application and handle the communication between the client and the server.
* Application Server: The application server will be responsible for processing user requests and providing responses.
* Database: The database will store all the user data and details.

## **4.2. Algorithm Details**

* **Hashing Algorithm** : Using secure and well-established password hashing techniques in Homestore is essential for protecting sensitive user data, ensuring legal compliance, maintaining trust and reputation, preventing unauthorized access, mitigating credential stuffing attacks, providing long-term security, and enabling secure password recovery and reset mechanisms. It is a fundamental security measure to safeguard user information and maintain the integrity of an e-commerce platform.

**Step wise Hashing Algorithm :**

* 1. Start
  2. Set the Length: Decide on the length of the salt (e.g., 16 characters).
  3. Choose Characters: Define a set of characters to pick from (letters A-Z, a-z, andnumbers 0-).
  4. Generate Salt:

Loop 16 times (or the chosen length):

1. Pick a random character from the character set.
2. Add the chosen character to the storage.
   1. Return Salt:Provide the generated string of characters as the salt.
   2. End

* **Binary Search Algorithm** : A binary search algorithm in a web application for online buying and selling products can be used to efficiently search for products based on specific criteria like price, seller, or categories. Binary Search retrives the data from product database of Homestore by with finding mid value to less the time complexity.

**Step wise Binary search Algorithm**

* 1. Start
  2. Check Range: If left is greater than right:
     + End:
     + Return -1, indicating that the key is not found in the array arr.
  3. Calculate Middle Index: Calculate the mid index as the floor of the average of left and right.
  4. Retrieve Middle products: Access the product at the mid index in the array arr.
  5. Compare Keys: If the title of the book at index mid matches the key:
     + End:
     + Return mid, indicating the index of the found item.
  6. Recursive Calls: If the title of the book at index mid is less than the key:
     + Recursively call binSearch with the range from mid + 1 to right.
     + Else:

Recursively call binSearch with the range from left to mid

**CHAPTER 5: IMPLEMENTATION AND TESTING**

## **Implementation**

### **5.1.1. Tools Used**

**Coding Tools**

Coding tools are the tools for software developers to create, debug, maintain, or otherwise support other programs and applications. The term usually refers to simple programs that can be combined to accomplish a task, much as one might use multiple hands to fix a physical object. Different front end and back-end tools were used in this project.

**Front End**

Front end tools are the tools to build attractive website layouts and apps with ease. Front end is the part the users interact with. React, Tailwind CSS, Visual Studio code etc. were used in the front end. Front end tools are used to convert data to a graphical interface, through HTML, CSS, and java script, so users can view and interact with that data.

**React**

React JS is a JavaScript library for building user interfaces that was developed and maintained by Faceproducts. It was released in 2013 and since then, it has become one of the most popular front-end development frameworks in the market. React JS uses a virtual DOM, which makes it highly efficient for handling updates and rendering components. Its modular design and reusable components make it easy for developers to create scalable and maintainable applications. Additionally, React JS has a strong developer community that provides a wealth of resources, tutorials, and support, making it a great choice for front-end development projects.

**5.1.1.2 Tailwind CSS**

Tailwind CSS is a popular React UI library that provides a set of pre-designed components that follow Google's Material Design guidelines. It is designed to help developers create user-friendly and visually appealing applications with ease. The components provided by Tailwind CSS are customizable and can be easily styled to match the look and feel of an application. Additionally, Tailwind CSS provides built-in accessibility features, making it easy for developers to create applications that are accessible to all users.

**5.1.1.3 Visual Studio Code**

#### Visual Studio Code (VS Code) is a free and open-source code editor developed by Microsoft. It is widely used by developers for its intuitive interface, fast performance, and powerful features. VS Code supports multiple programming languages and provides a range of tools for code development, such as debugging, code highlighting, embedded Git, and IntelliSense (code completion). In addition to its features, VS Code has a vast library of extensions that can be installed to add more functionalities.

**5.1.1.4 GitHub**

GitHub is a web-based platform that provides hosting for software development and version control. It enables developers to store and manage their code, collaborate with others, and track changes to their projects. GitHub makes it easy for developers to work together on the same codebase, even from distinct locations. By using GitHub, it is possible to keep track of changes to the project and collaborate effectively. Additionally, GitHub provided a centralized platform for storing and managing the project's code, which made it easy to access and update the code as needed.

**5.1.2 Back End**

Back End means the server, application and database that work behind the scenes to deliver information to the user and the tools that are used in this process are called back-end tools. Simple-peer library (node.js style API for WebRTC, express js) were used in the back end with Firebase to handle the database.

.

**5.1.2.2 Node JS**

#### Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client's browser. Ryan Dahl developed it in 2009. Node.js is used to create server-side web applications, and it is perfect for data-intensive applications since it uses an asynchronous, event-driven model.

#### **5.1.2.3 MongoDB**

#### MongoDB is a Backend-as-a-Service (BaaS) platform that provides a flexible and scalable database solution for web and mobile applications. It provides real-time data synchronization and automatic data backup, which ensures that the data stored in the database is always up-to-date and secure. The platform also offers easy integration with other Google services, Authentication, Realtime database, database, Storage.

### **5.1.2. Implementation Details of Modules**

The implementation details of the modules in the Homestore platform are as follows:

**User Authentication Module:** This module handles user authentication and authorization on the platform. It uses a combination of email and password authentication and social media logins such as Google to authenticate users.

**Buyers and sellers Verification Module:** This module verifies user’s profiles and credentials before they are approved to provide legal services on the platform. It involves a rigorous vetting process that includes background checks, document verification, and interviews.

**AIChat Module:** This module allows clients and admins to communicate through a real-time chat interface during video calls. It includes features such as file sharing, message history, and message notifications.

**Admin Panel Module:** This module provides administrators with access to the platform's management tools, including user management, lawyer verification, and notification management. It includes features such as user roles and permissions, data analytics, and reporting.

These modules are developed using various technologies such as HTML, CSS, JavaScript, Node.js, Express, Firebases, and WebRTC.

## **5.2. Testing**

### **5.2.1. Unit Testing**

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated, but it can be done manually. Unit testing must be done with an awareness that it may not be possible to test a unit for every input scenario that will occur when the program is run in a real-world environment. In inventory management system, Unit testing were done in the following forms:

## 

## **Table 2: 5.2.1. Unit Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Login Module | | | | |
| No | Test Case | Expected Result | Actual  Result | Remark |
| 1 | Login with correct username and wrong  password | Display details. alert message to user “Username and password not matched” | As expected | PASS |
| 2 | Login with wrong username and wrong password | Display alertmessage to user “Username and password not matched” | The system correctly displayed a message. | PASS |
| 3 | Login with blank username and blank password | Display alert message to user “Please fill out  this field” | As expected | PASS |
| 4 | Login with correct username and correct password | Display alert message to user “Logged In Successfully.” | As expected | PASS |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | Login with Changed Password | Display alert message to user “Invalid Password” | Display alert message to user “Invalid Password” | FAIL |

### **5.2.2. Integration Testing**

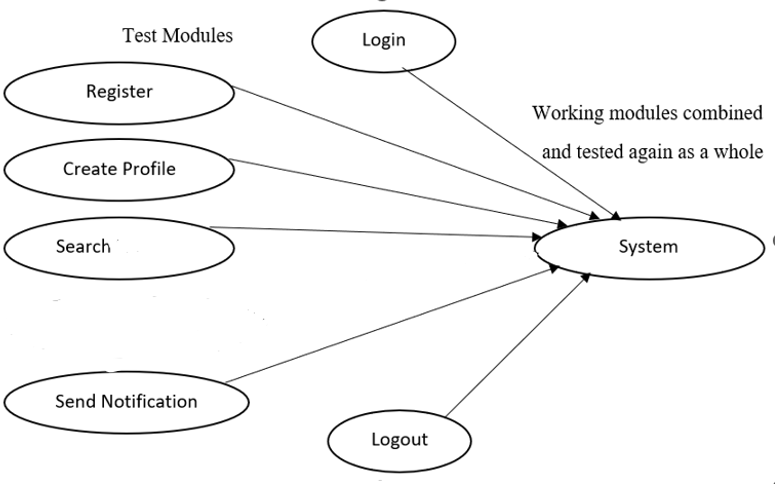
Integration testing is done to test the modules/components when integrated to verify that they work as expected i.e., to test the modules which are working fine individually do not have issues when integrated. Suppose if an application has three modules A, B, and C, then testing done by combining the modules A & B or module B & C or module A& C is known as Integration testing.

## **Table 3:5.2.2. Integration Testing**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T.C ID** | **Test Scenario** | **Test Case** | **Test Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | Check Login Functionality | Check response on entering valid username and password | 1. Launch App  2. Click Get Started button  3. Click Already have on account button  4. Enter username and password  5. Click Signin button | Login must be successful | Login Successful | Pass |
| 2 | Check Registration Functionality for Users and Admins | Check response on creating new valid username and password | 1. Launch App  2. Click Get Started button  3. Enter new username and password  4. Enter Signup button | Registration must be successful | Registered Successful | Pass |
| 3 | Check Login  Functionality with invalid data | Check response on Login with invalid username and password | 1. Launch App  2. Click Get Started button  3. Click Already have on account button  4. Enter username and password  5. Click Signin button | Login must not be successful | Login  Failed | Pass |
| 4 | Check create Profile Functionality for Admins and Users | Check response on creating profile | 1. Login as Admin/User  2. Fill the data  3. Click Update button | Creating profile must be successful | Created Successful | Pass |
| 5 | Check Dashboard Functionality | Check response on all options of Dashboard | 1. Login  2. Click on all options available on navbar of Dashboard | Using all Dashboard features must be successful | Working Successful | Pass |

### **5.2.3. System Testing**

System Testing means testing the system. All the modules/components are integrated to verify if the system works as expected or not. System testing is done after integration testing. The system testing part of a testing methodology involves testing the entire system for errors and bugs. No errors and bugs were found during system testing.



## **Figure 12: 5.2.3. System Testing**

## **5.3 Result Analysis**

To conduct the result analysis, several key activities are performed:

**Functional Testing:** The functionality of different features and modules of the Homestore platform is thoroughly tested to ensure they perform as expected. This includes testing the searching system, chat feature, and user-admin interaction.

**Performance Testing:** The performance of the system is evaluated under different scenarios to measure its response time, scalability, and resource utilization. This testing ensures that the platform can handle a significant number of users and concurrent video calls without compromising performance.

**Usability Testing:** The usability of the platform is assessed through user testing and feedback. This involves conducting user surveys or interviews to gather insights on the ease of navigation, intuitiveness of the user interface, and overall user experience. The feedback helps identify any usability issues and make necessary improvements.

**Error Handling and Exception Testing:** The system's ability to handle errors and exceptions is thoroughly tested to ensure that it can gracefully handle unexpected situations, such as network failures, invalid inputs, or server errors. This testing helps identify and rectify any vulnerabilities or error-handling gaps in the system.

**Security Testing:** The security measures implemented in the Homestore platform are assessed to ensure the protection of user data and privacy. This includes testing for vulnerabilities such as data breaches, unauthorized access, and secure transmission of sensitive information during chat sessions.

# **CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATIONS**

## **6.1. Lesson learned**

Doing something for long periods always gives good lessons. Some of the things that our team learnt are listed as below:

i. Basically, we learnt to work in a team.

ii. Learnt about the Web-app platform.

iii. Learnt about front end, back-end technology & report documentation.

iv. Learnt to manage the database.

## **6.2. Conclusion**

The development of Homestore project has been a successful effort to provide a platform for buyers and admins to connect and engage in business services through chat. The project was designed with the goal of making e-commerce services more accessible and convenient for clients, while providing users with a platform to showcase their services and connect with potential admins. The project was implemented using the latest technologies, including React JS, Tailwind CSS, Node JS, MongoDB, and GitHub. The technical feasibility, operational feasibility, and economic feasibility of the project were analyzed, and all aspects of the project were tested and refined to ensure a seamless user experience. Lessons were learned throughout the development process and will be used to improve future projects. In the end, the development of Homestore project demonstrates the potential of technology to transform traditional industries and make services more accessible to people everywhere.

## **6.3. Future Recommendations**

Some of the future enhancements that can be enhanced are as follows:

1. Adding new features to the platform such as secure payment options and appointment scheduling tools.
2. Improving the user interface and overall user experience
3. Integrating advanced technologies such as artificial intelligence.
4. Providing better support and maintenance services to users.

# **‘REFERENCES**

[[1]](#Lawyerist) Vaidya, Rashesh. “Online Shopping in Nepal: Preferences and Problems.” Journal of Nepalese Business Studies 12, no . 1 (December 31, 2019): 71-86.

http://dx.doi.org/10.3126/jnbs.v12i1.28184

[2] Sabar, Wardihan, “Prospects of E-Commerce Innovation in Increasing Students Consumer Trust.” Jurnal Pendidikan Ekonomi (JUPE) 10, no. 2 (May 2, 2022): 121-29.

https://dx.doi.org/10.26740/jupe.v10n2p.p121-129

[3] Prashar, Sanjeev, T.Sai Vijay, and Chandan Parsad. “Selecting a Web Portal for Online

Shopping: A Conceptual Approach Using Interpretive Structural Modeling.” East Asian Journal of Business Management 5, no. 4 (October 30, 2015): 37-46.

http://dx.doi.org/10.13106/eajbm.2015.vol5.no4.37.

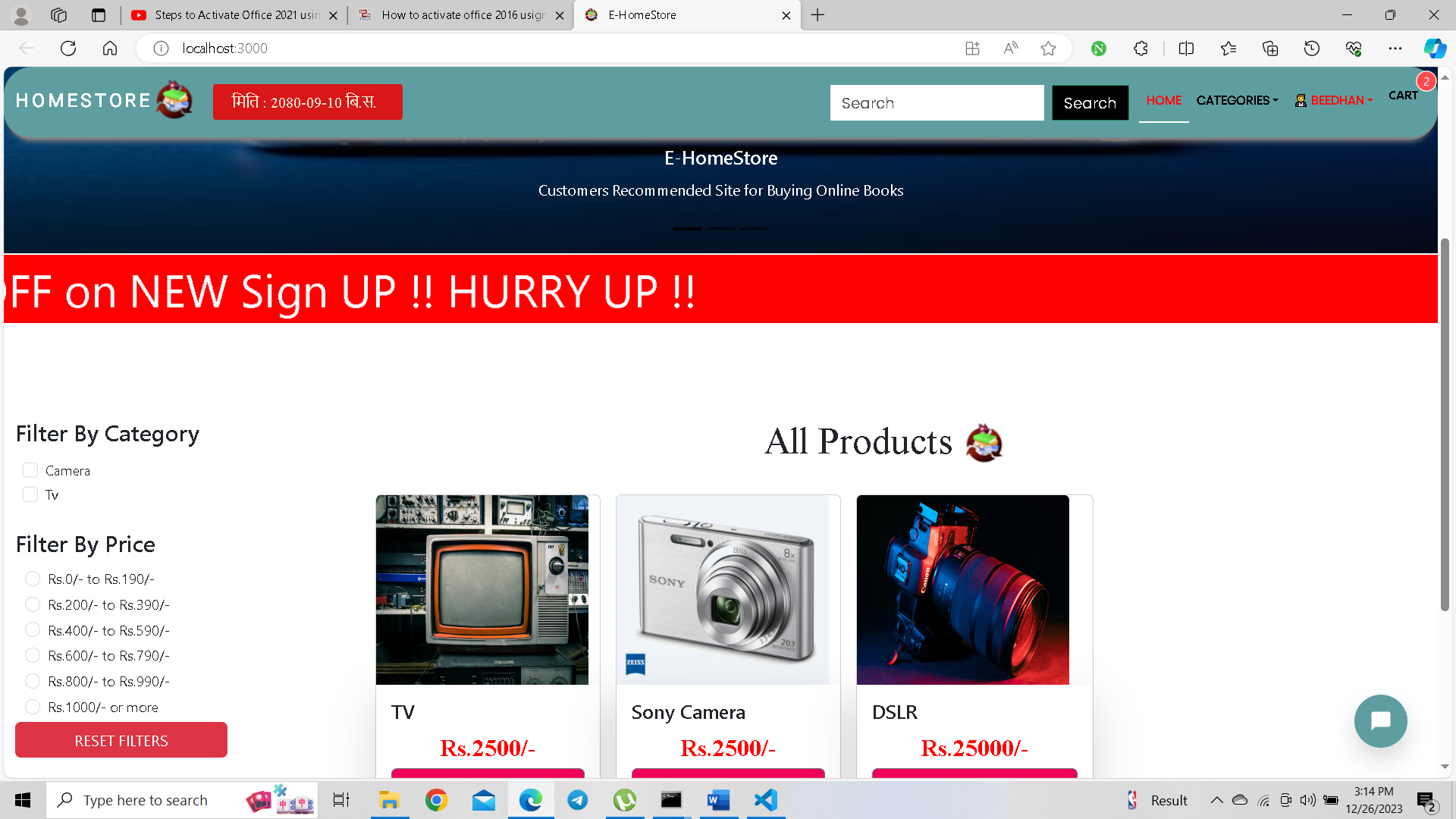
[4] Lang. C. (2018). Perceived risks and enjoyment of access based consumption:

Identifying barriers and motivations to fashion renting. Fashion and Textiles, 5(1), 118.

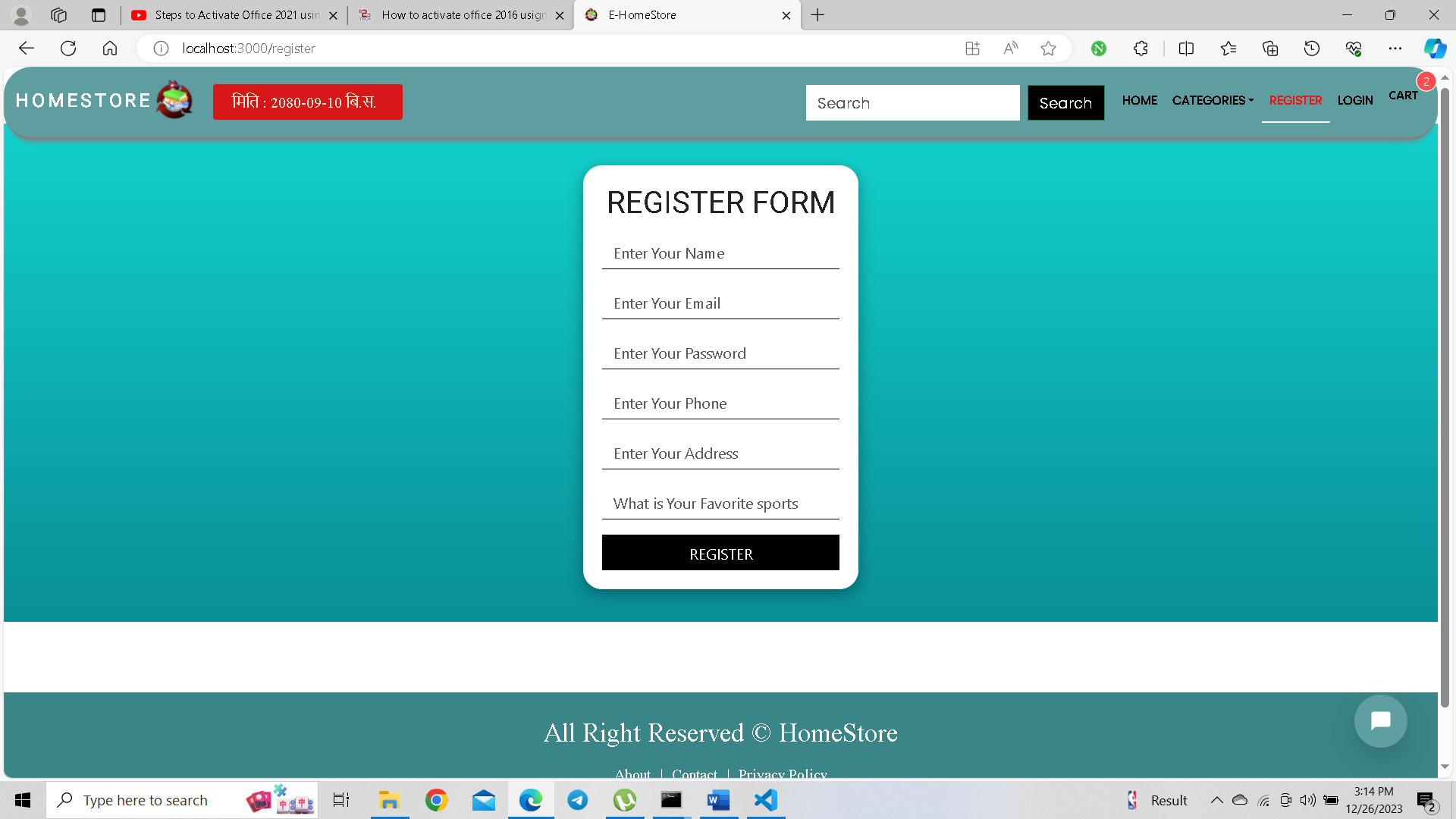
# **Appendix**

## **Appendix A: Snapshots**

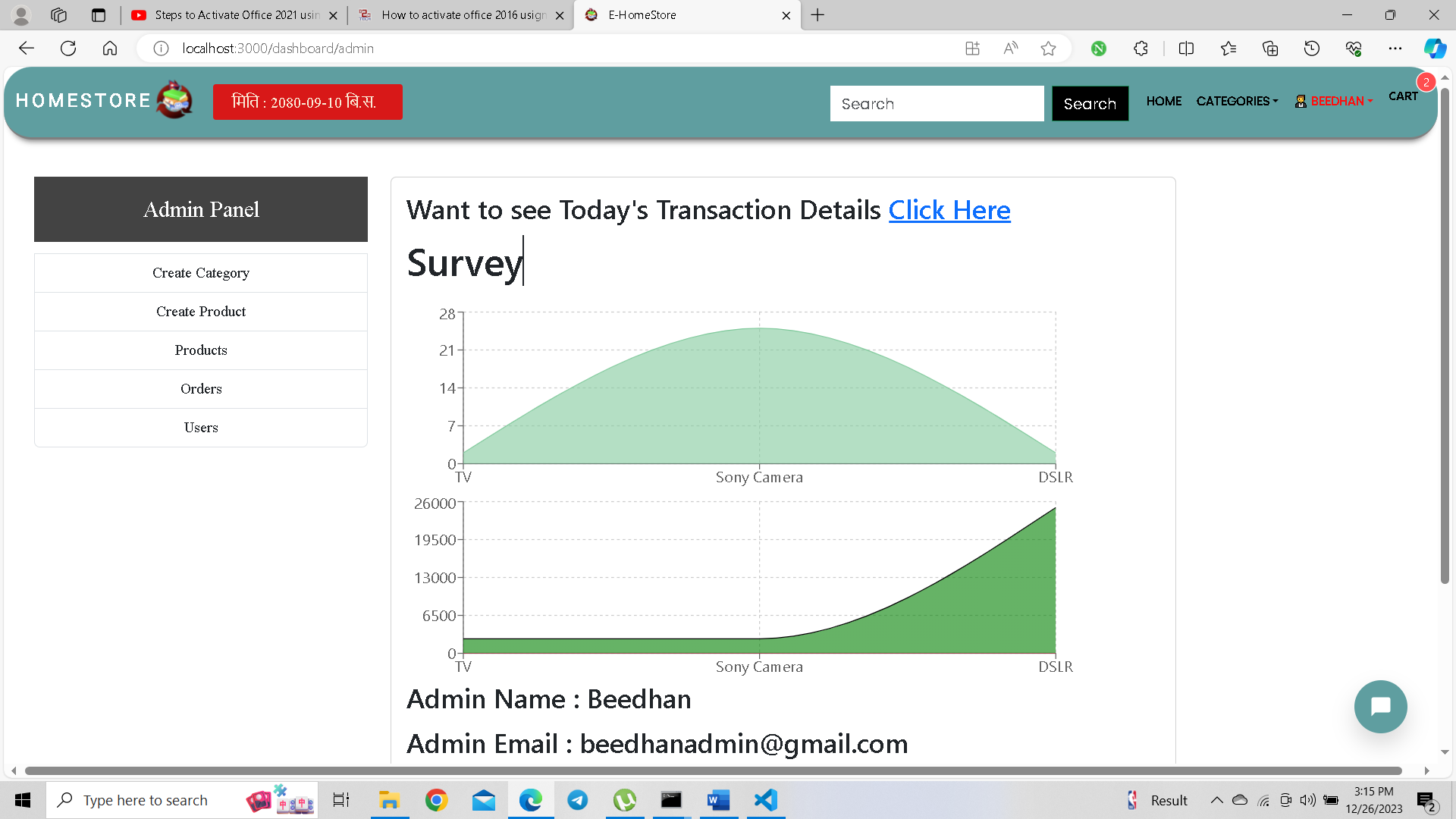
**­Homepage:**

****

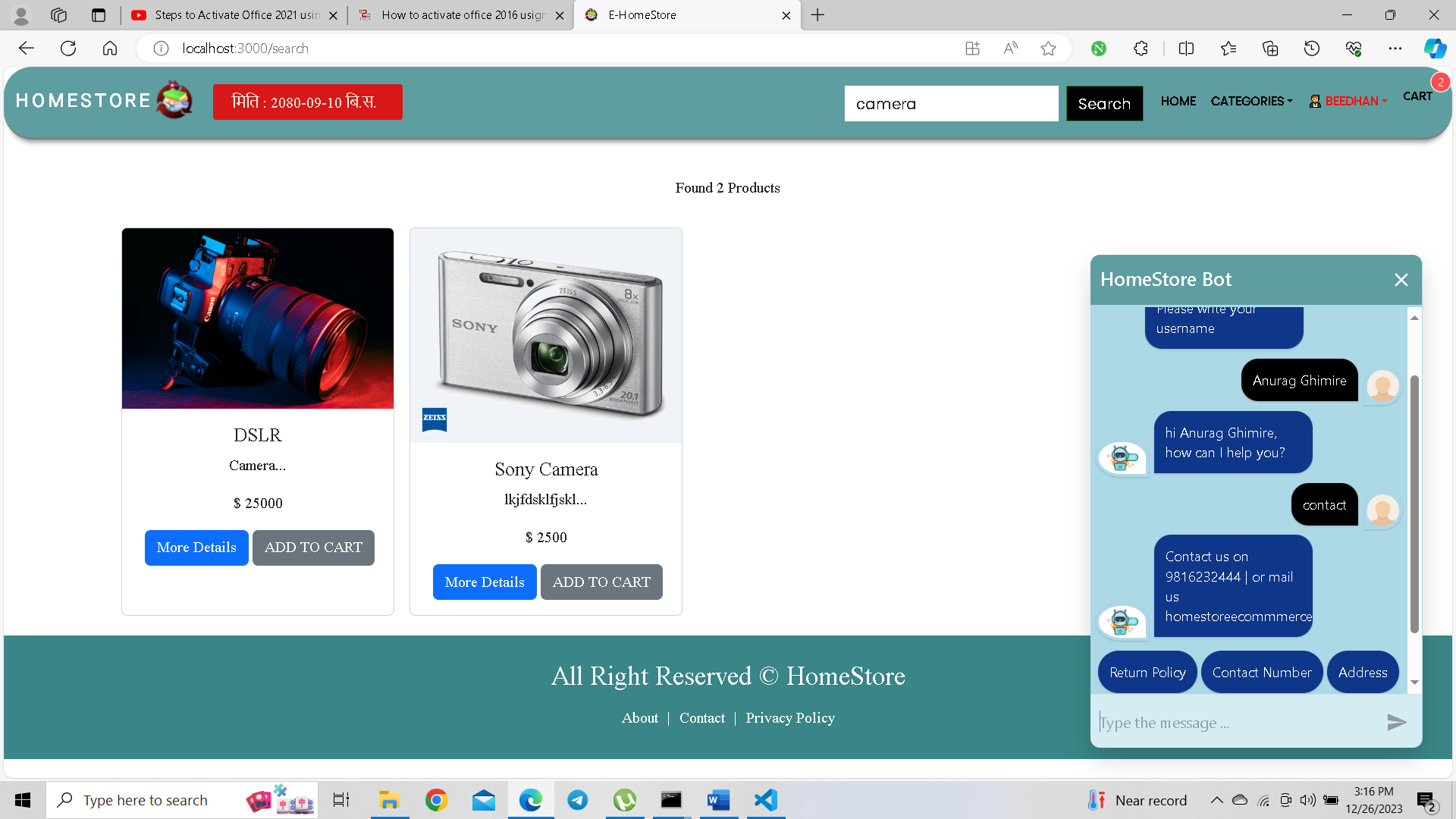
**Registration Form:**

****

**Dashboard**

****

**Chat**

****

## **Appendix B: Source Code**

**Server**

**Admin Dashboard**

const AdminDashboard = () => {

const [chartData,setChartData] = useState([]); const [chartOrderData,setChartOrderData] = useState([]); const [auth] = useAuth();

const getAllProducts =

async () => { try {

const { data } = await axios.get("/api/v1/product/getproduct"); setChartData(data.products);

} catch (error) { console.l og(error)

;

toast.error("Something Went Wrong");

}};

//lifecycle

useEffect(() => { getAllProducts();

}, []);

const getAllOrders = async () => { try {

const { orderdata } = await axios.get("/api/v1/auth/allorders"); setChartOrderData(orderdata);

}

catch (error) { console.l og(error)

;

toast.error("Something Went Wrong");

}

};

//lifecycle method useEffect((

) => {

getAllOrde rs();

}

,

[

]

)

const gradientOffset = () => {

const dataMax = Math.max(...chartData.map((i) => i.quantity)); const dataMin = Math.min(...chartData.map((i) => i.quantity));

if

(dataMa x <= 0)

{ return 0;

if (dataMin >= 0) { return 1;

}

return dataMax / (dataMax dataMin);

};const off = gradientOffset();

**User Dashboard** :

<Layout title={"Your Orders"}>

<div className="containerflui p3 m3 dashboard">

<div className="row">

<div className="colmd3">

<UserMenu />

</div>

<div className="colmd9">

<h1 className="textcenter">All Orders</h1>

{orders?.map(

(o, i) => {

return (

<div className="border shadow">

<table className="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Status</th>

<th scope="col">Buyer</th>

<th scope="col"> date</th>

<th scope="col">Payment</th><th scope="col">Quantity</th>

</tr>

</thead>

<tbody>

<tr>

<td>{i + 1}</td>

<td>{o?.status}</td>

<td>{o?.buyer?.name}</td>

<td>{moment(o?.createAt).fromNow()}</td>

<td>{o?.payment.success ? "Success" : "Failed"}</td>

<td>{o?.products?.length}</td>

</tr>

</tbody>

</table>

<div className="container">

{o?.products?.map((p, i) => (

<div className="row mb2 p3 card flexrow" key={p.\_id}>

<div className="colmd4">

<img src={"/api/v1/product/productphoto/$

{p.\_id}"} className="cardimgtop"

alt={p.n ame} width=" 100px" height={ "100px"

}

/>.

</div>

<div className="colmd8">

<p>{p.name}</p>

<p>{p.description.substring(0, 30)}</p>

<p>Price : {p.price}</p>

</div></div>

))}