

# Tribhuvan University

**Institute of Science and Technology**

**Mernify: “The Ultimate Shopping Experience” Project Report**

## Submitted to

**Department of Computer Science and Information Technology**

**National College of Computer Studies**

*In partial fulfillment of the requirements for Bachelor’s in Computer Application*

## Submitted By:

**……………………**

**Under the Supervision of**

**Teksan Gharti Magar**



**Tribhuvan University**

**Institute of Science and Technology**

**National College of Computer Studies**

# SUPERVISOR’S RECOMMENDATION

We hereby recommend that this project prepared under our supervision by

**Teksan Gharti Magar** entitled “**Mernify: The Ultimate Shopping Experience**” in partial fulfillment of the requirements for a degree of Bachelor’s in Computer Science and Information technology is recommended for the final evaluation.

**SIGNATURE**

**Teksan Gharti Magar SUPERVISOR**

**National College of Computer**

# Abstract

Mernify is a modern e-commerce platform designed to provide a seamless online shopping experience for both buyers and sellers. Built using the MERN (MongoDB, Express.js, React, and Node.js) stack, it ensures high performance, scalability, and user-friendly interactions. The platform offers a wide range of features, including product listings, secure payment integration, order management, and user authentication. Additionally, Mernify supports advanced search and filtering options, personalized recommendations, and a responsive design for an optimal shopping experience across all devices. With its intuitive interface and robust backend, Mernify aims to redefine digital commerce, making it more accessible and efficient for businesses and customers alike.

# Acknowledgement

Bringing this study to life has been both rewarding and challenging. Throughout this journey, many people have stepped in at the right moments, offering their help and support, and they deserve special thanks.

We are deeply grateful to the department for their guidance and continuous support. Their valuable information and supervision played a crucial role in helping us complete this project. We also want to express our heartfelt thanks to everyone who contributed, directly or indirectly, to make this study possible.

Lastly, we sincerely appreciate those who take the time to read this project and hope it will benefit them now and in the future.

|  |  |
| --- | --- |
| **Table of Contents** |  |
| SUPERVISOR’S RECOMMENDATION | i |
| Abstract | ii |
| Acknowledgement | iii |
| Table of Contents | iv |
| List of Figures | vi |
| List of Tables | vii |
| Chapter 1: Introduction | 1 |
| 1.1. Introduction | 1 |
| 1.2. Problem Statement | 1 |
| 1.3. Objectives | 1 |
| 1.4. Scope and Limitation | 2 |
| 1.5. Report Organization | 3 |
| 1.5.1. Introduction | 3 |
| 1.5.2. Background Study | 3 |
| 1.5.3. System Analysis and Design | 3 |
| 1.5.4. Implementation and Testing | 3 |
| 1.5.5. Conclusion and Future Recommendations | 3 |
| Chapter 2: Background Study | 4 |
| 2.1. Background Study | 4 |
| 2.1.1. Study of existing system | 4 |
| 2.1.2. Literature Review | 5 |
| Chapter 3: System Analysis and Design | 7 |
| 3.1. System Analysis | 7 |
| 3.1.1. Requirement Identification | 8 |
| 3.1.1.1. Functional Requirements | 8 |
| 3.1.1.2. Non-Functional Requirements | 10 |

* + 1. [Feasibility Study 11](#_TOC_250023)
       1. [Technical Feasibility 11](#_TOC_250022)
       2. [Operational Feasibility 13](#_TOC_250021)
       3. [Economic Feasibility 14](#_TOC_250020)
       4. [Schedule Feasibility 15](#_TOC_250019)
    2. [Data Modeling 15](#_TOC_250018)
    3. [Process Modeling 17](#_TOC_250017)
    4. [Sequence Diagram 19](#_TOC_250015)
    5. [Activity Diagram 20](#_TOC_250014)
  1. [System Design 21](#_TOC_250013)
     1. [Architectural Design 21](#_TOC_250012)
     2. [Database Schema Design 22](#_TOC_250011)

[Chapter 4: Implementation and Testing 23](#_TOC_250010)

* 1. [Implementation 23](#_TOC_250009)
     1. [Tools used 23](#_TOC_250008)
  2. [Testing 24](#_TOC_250007)
     1. [Test Case 24](#_TOC_250006)

[Chapter 5: Conclusion and Future Recommendation 29](#_TOC_250005)

* 1. [Lesson Learnt / Outcome 29](#_TOC_250004)
  2. [Conclusion 29](#_TOC_250003)
  3. [Future Recommendations 29](#_TOC_250002)

[References 30](#_TOC_250001)

[Appendices 31](#_TOC_250000)

# List of Figures

Figure 1: Use Case Diagram of WriteOn 7

Figure 2: Use Case Diagram of Kapi 9

Figure 3: Timeline of Kapi 15

Figure 4: E-R Diagram of Kapi 15

Figure 5: Data Flow Diagram of Kapi 17

Figure 6: Class Diagram of Kapi 18

Figure 7: Sequence Diagram of Kapi 19

Figure 8: Activity Diagram of Kapi 20

Figure 9: Basic Database Schema of Kapi 22

Figure 10: Homepage of Kapi 31

Figure 11: Blog page of Kapi 32

Figure 12:Shop Page of Kapi 32

Figure 13: Registration and Login Page of Kapi 33

Figure 14: User Dashboard of Kapi 33

Figure 15: Quick View of Products of Kapi 34

Figure 16: Products Page of Kapi 34

Figure 17: Cart Page of Kapi 35

Figure 18: Checkout Page of Kapi 36

Figure 19: Payment Received through Cash On Delivery of Kapi 37

Figure 20: Khalti Test Payment Prompt of Kapi 38

Figure 21: Payment Received through Khalti of Kapi 38

Figure 22: Order History and Tracking Page of Kapi 39

Figure 23: Contact Us Page of Kapi 39

|  |  |  |
| --- | --- | --- |
|  | **List of Tables** |  |
| Table 1: Test Case 1 |  | 24 |
| Table 2: Test Case 2 |  | 25 |
| Table 3: Test Case 3 |  | 26 |
| Table 4: Test case 4 |  | 27 |
| Table 5: Test Case 5 |  | 28 |

# Chapter 1: Introduction

## Introduction

E-commerce, or electronic commerce, refers to the buying and selling of goods and services over the internet. It has transformed the way businesses operate, allowing consumers to shop conveniently from anywhere at any time. With the rise of digital technologies, e-commerce has become an essential part of the global economy, offering businesses new opportunities to expand their reach beyond geographical limitations.

E-commerce platforms provide a seamless shopping experience through features like product listings, secure payment gateways, order tracking, and customer reviews. They cater to various business models, including Business-to-Consumer (B2C), Business-to-Business (B2B), and Consumer-to-Consumer (C2C) transactions. Advanced technologies such as artificial intelligence, personalized recommendations, and mobile commerce have further enhanced the efficiency and convenience of online shopping.

## Problem Statement

Many businesses and consumers face challenges with existing e-commerce platforms, including complex navigation, security risks, high costs, and limited customization. Small businesses struggle to establish an online presence due to technical barriers, while consumers encounter inefficient search, slow performance, and data privacy concerns.

Mernify addresses these issues by providing a secure, user-friendly, and scalable e-commerce platform that simplifies online shopping and selling, ensuring a seamless and efficient experience for all users.

## Objectives

The main objective of Mernify is:

* To make online selling easy by providing a simple platform for businesses to list and manage products.
* To improve the shopping experience with a fast, user-friendly, and responsive design.
* To help customers find products easily with smart search and filtering options..

## Scope and Limitation

The scope of the Mernify site include:

* Mernify makes buying clothes online simple and easy.
* Provides a user-friendly way to browse and buy clothes.
* Mernify uses online strategies to reach more customers.
* Cover important topics like customer security and privacy.
* Focus on enhancing customer management and improving the overall user experience.
* Focus on making shopping personal and enjoyable.

Some limitations of the Mernify site might include:

* The website will only sell products, not services.
* It might not have smart features like showing products based on what you like
* There may not be a live chat to talk to customer support
* The site may not work well on very old web browsers
* Product information will be simple and may not include all technical details.

## Development Methodology

## Report Organization

### Introduction

This chapter introduces the concept of the project. It describes the problem the existing problems faced and our objectives to tackle it. It also briefs about the project objective, scope, and limitations.

### Background Study

This chapter focuses on study of existing systems and review of different articles studied. It also explores the base ideology for building the system.

### System Analysis and Design

This chapter covers the project's requirements collecting, feasibility assessment, and design. Diagrams, functionality analysis, a technique for obtaining requirements, and a process model are all included.

### Implementation and Testing

This chapter is intended to provide details on how the project was implemented, the software and tools that were utilized, and the types of testing that the project underwent.

### Conclusion and Future Recommendations

This chapter discusses the project's potential outcomes, as well as the conclusion and future recommendations.

# Chapter 2: Background Study

## Background Study

Mernify is an e-commerce platform made to give users a simple and smooth online shopping experience. It allows people to browse products, add them to a cart, and buy them easily from their phones or computers. The platform has a clean and user-friendly design that makes it easy for anyone to use, whether they are shopping online for the first time or regularly.

With Mernify, users can create accounts, view product details, and place orders without any confusion. The website helps users find what they need quickly and safely. It also includes features like order history and a simple checkout process. Mernify is built using the MERN stack-MongoDB, Express, React, and Node.js which helps make the platform fast, secure, and reliable. The goal of Mernify is to make online shopping easy, enjoyable, and accessible for everyone.

**Study of existing system**

In our research of current online shopping platforms, we looked at **Amazon** and **Daraz** as references while planning the development of **Mernify**. These platforms are popular and widely used for their large product selections, user-friendly design, and smooth shopping experience.

Amazon and Daraz are successful because they make it easy for users to find products, add them to a cart, and place orders quickly. They also offer features like product reviews, order tracking, and secure payment methods. Their clean design, smart search tools, and fast performance help users shop with ease. Both platforms also support mobile apps, making shopping more convenient on smartphones.

The simplicity and effectiveness of Amazon and Daraz helped guide how we designed and built Mernify. Our aim is to provide a similarly easy and smooth shopping experience, with simple navigation, secure checkout, and helpful features—so that anyone can shop online without confusion or technical skills.

### Literature Review

Guillaume Thevenot (June 7, 2007) published a paper highlighting that blogging is one of the most popular tools in social media. On these platforms, people share their thoughts, experiences, and perspectives through various formats such as text, images, and videos. Blogging often starts with one person posting an article, which readers then comment on, creating a conversation. Thevenot mentions that approximately 120,000 new blogs are created daily worldwide, which equates to about 1.4 blogs per second. Many of these blogs are related to travel. The paper also discusses different types of blogs, including individual, collaborative, corporate, and traditional media blogs. [1]

Dou Shen, Jian-tao Sun, Qiang Yang, and Zheng Chen (2006) wrote an article on extracting hidden friendships from blog data. Blogs are a valuable source of data that provide rich resources for social community mining. This process involves analyzing large amounts of user-generated content from social media and mobile apps to identify patterns, often for advertising or research purposes. The authors define latent friends as individuals who share similar topics in their blogs. The paper outlines three methods for detecting these hidden friendships: the cosine similarity-based method, the topic model-based method, and the two-level similarity-based method. [2]

Child, J. T., Haridakis, P. M., and Petronio, S. (2012) published a paper discussing privacy rules and management in blogging. The authors studied different stages of blogging, focusing on how it functions before and after blogs are posted. They categorized various privacy rules and explored why bloggers might delete previously posted content. The paper collected data on bloggers' activities to understand their reasons for deleting content before it is shared. The study highlights the role of privacy rules, privacy management, and the motivations for deleting posts after blogging. [3]

# Chapter 3: System Analysis and Design

## System Analysis

System development is usually done in two main parts: system analysis and system design. In this chapter, we look at how the system was planned based on research and how we used diagrams to better understand and visualize how everything should work. These diagrams help us design the system clearly, using object-oriented concepts.

Since this is a small-scale project with clear goals, we needed a development approach that is flexible and can keep up with changing trends in online shopping. After looking at different options, we chose the Rational Unified Process (RUP), which is a type of Agile method.

We picked RUP because it allows us to build the system step-by-step through iterations. This means we can improve the platform over time based on feedback, fix problems early, and make sure everything is on track. RUP also helps us manage risks better and keeps the development process organized.

Using this method, we aim to build a strong, flexible, and easy-to-use e-commerce platform that meets the needs of users. It also ensures that our system can grow or change in the future as user needs or market trends change. In short, RUP gives us the right balance between structure and flexibility for creating a successful online shopping experience with Mernify.

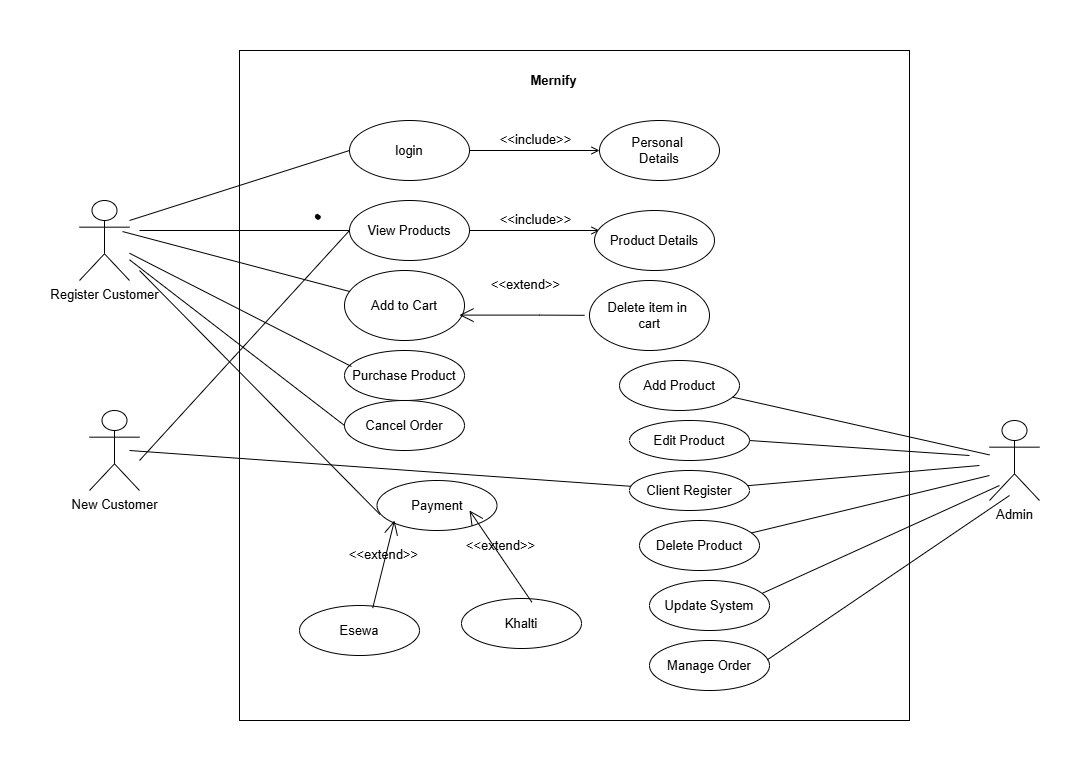
### Requirement Identification

Requirements of the system are identified through personal research of visiting various ecommerce sites.

### Functional Requirements

Functional requirements for e-commerce platform:

* + - * 1. User registration and login - Users should be able to create an account and log in using their email and password to access their profile, order history, and make purchases.
        2. Product Browsing and Search - Users should be able to browse all available products and search for specific items using keywords or categories.
        3. Product Details Page - Users should be able to click on a product to view detailed information, including name, price, description, images, and available stock.
        4. Shopping Cart System- Users should be able to add products to their cart, update quantities, and remove items before proceeding to checkout.
        5. Checkout and Payment Process- Users should be able to place an order by providing delivery information and choosing a payment method.
        6. Order Management: Users should be able to view their past orders, including product details, order status, and delivery tracking (if available).
        7. Admin Panel: Admins should be able to manage the website by adding, editing, or deleting products, and viewing customer orders.
        8. User Profile Management: Users should be able to view and update their personal details such as name, address, and password.
        9. Wishlist Feature: Users should be able to save products to a wishlist for future reference or purchase.
        10. Responsive Design: The platform should work smoothly on all devices desktop, tablet, and mobile for a good user experience.



*Figure 1: Use Case Diagram of Mernify*

The use case diagram for Mernify blog platform outlines the key user interactions. It includes actors such as Customer and Admin, with use cases view products, add to cart, purchase product, payment etc. The diagram shows how users interact with the platform and how admin manage the system. It helps visualize the system’s functionality and user interactions, providing a clear understanding of its behaviour.

### Non-Functional Requirements

Non-Functional requirements for Mernify e-commerce platform:

* + - * 1. Performance - The platform should run smoothly, even when many users are online at the same time, ensuring no noticeable slowdown during high traffic periods
        2. Scalability - As the business grows, the platform should easily handle more users and transactions without needing major changes or causing performance issues.
        3. Availability - The platform should be up and running around the clock, with only minimal interruptions for necessary maintenance or updates to ensure a reliable experience for users
        4. Ease of Use - The system should be straightforward and user-friendly, making it easy for both administrators and regular users to navigate and use all features efficiently.
        5. Compliance- The platform must comply with all relevant laws, regulations, and industry standards to protect both the business and its users.
        6. User Experience- The platform should offer a positive and engaging user experience, with intuitive navigation, a clean and responsive design, and an overall satisfying interface that encourages users to return.
        7. Mobile Compatibility- The platform should work seamlessly on mobile devices, offering a responsive design that ensures a smooth and enjoyable experience whether users are on a phone, tablet, or computer.

### Feasibility Study

A feasibility study evaluates the operational, technical, and financial aspects of the proposed project. Its purpose is to conduct an initial review to determine if the project is viable enough to move forward to the detailed analysis phase. For system analysts, the feasibility study is a crucial tool for deciding whether to recommend continuing with the project or halting it.

It ensures that the project is practical, achievable, and worth pursuing based on the available resources and expected outcomes

### Technical Feasibility

The technical feasibility of the Mernify ecommerce platform focused on determining whether the project could be successfully built and deployed using available technology and resources.

**Technology and Tools:** Mernify is built using reliable and widely available web technologies, ensuring a smooth development process. The platform leverages modern frameworks and tools to provide essential blogging features, including user registration, content creation, and management. Customization options allow users to personalize their blogs to suit their preferences.

**Payment Integration:** The platform supports multiple payment options for premium features, including the integration of secure online payment gateways esewa and Khalti. This ensures smooth and safe transactions, which is a critical component of the platform.

In summary, Mernify has a high level of technical feasibility. Even though payment integration is not yet finalized, all necessary technology and resources are in place to build and deploy the platform successfully

### Operational Feasibility

Operational feasibility assesses whether the Mernify e-commerce platform can be effectively implemented and integrated into current operations and whether it will be well-received by its intended users.

**Target Users:** Mernify is mainly designed for general online shoppers, small business owners, and sellers who want to buy or sell products through a digital platform. The system aims to provide a simple and efficient way to explore products, make purchases, and manage sales. With the growing demand for user-friendly and accessible online shopping platforms, Mernify is expected to attract a wide range of users looking for a smooth and reliable e-commerce experience.

**Integration with Existing Operations:** Mernify is built to work smoothly with current business operations, such as inventory management, order tracking, and customer service tools. This allows sellers and store owners to manage their products and orders without changing their existing workflow. The platform is also flexible and can be updated in the future to support other useful services like payment gateways, shipping providers, or analytics tools to improve both the seller and customer experience.

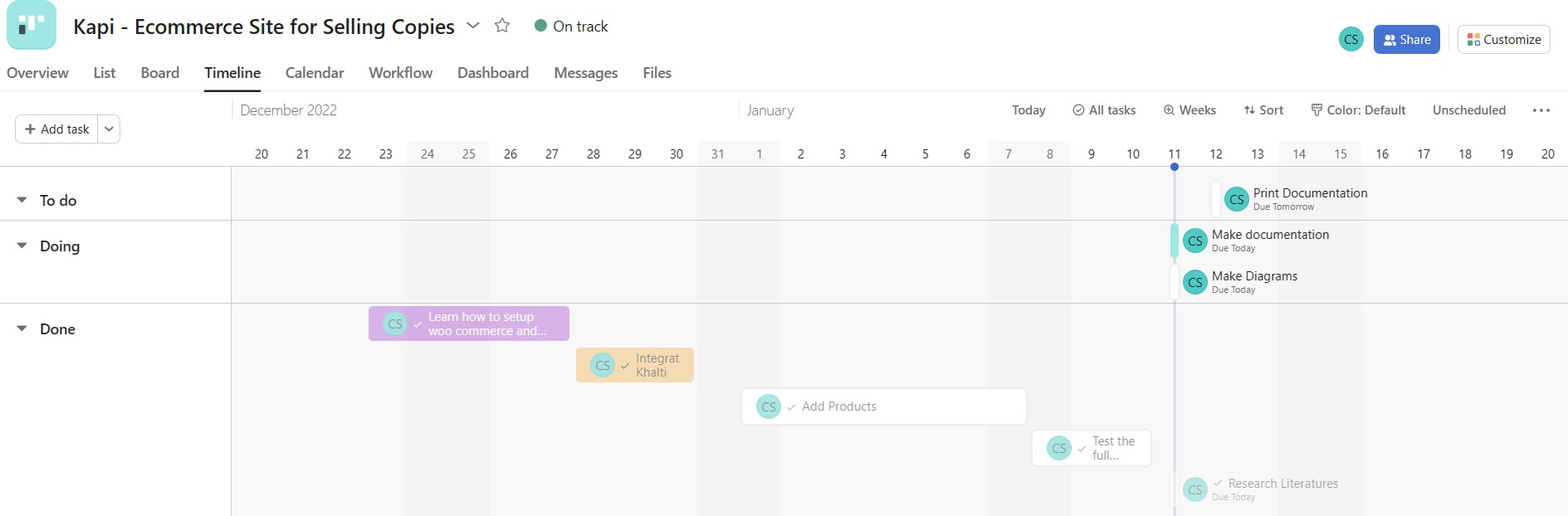
### Economic Feasibility

Economic feasibility evaluates whether the Mernify e-commerce platform is financially viable, ensuring that the benefits outweigh the costs of its development and operation.

**Development Costs:** Mernify is built using the MERN stack (MongoDB, Express.js, React, and Node.js), which is open-source and free to use. This choice minimizes software licensing costs. However, there will be expenses related to developing and customizing the platform to meet user needs, such as designing unique features and enhancing the user experience. Additionally, hosting services and domain registration will incur recurring costs.

**Operational Costs:** The platform requires ongoing maintenance, including server management, software updates, and security enhancements. Customer support and content moderation will also add to operational expenses. Despite these costs, the platform is expected to generate revenue through premium subscriptions, ads, or additional services, helping to balance operational costs.

### Schedule Feasibility

****

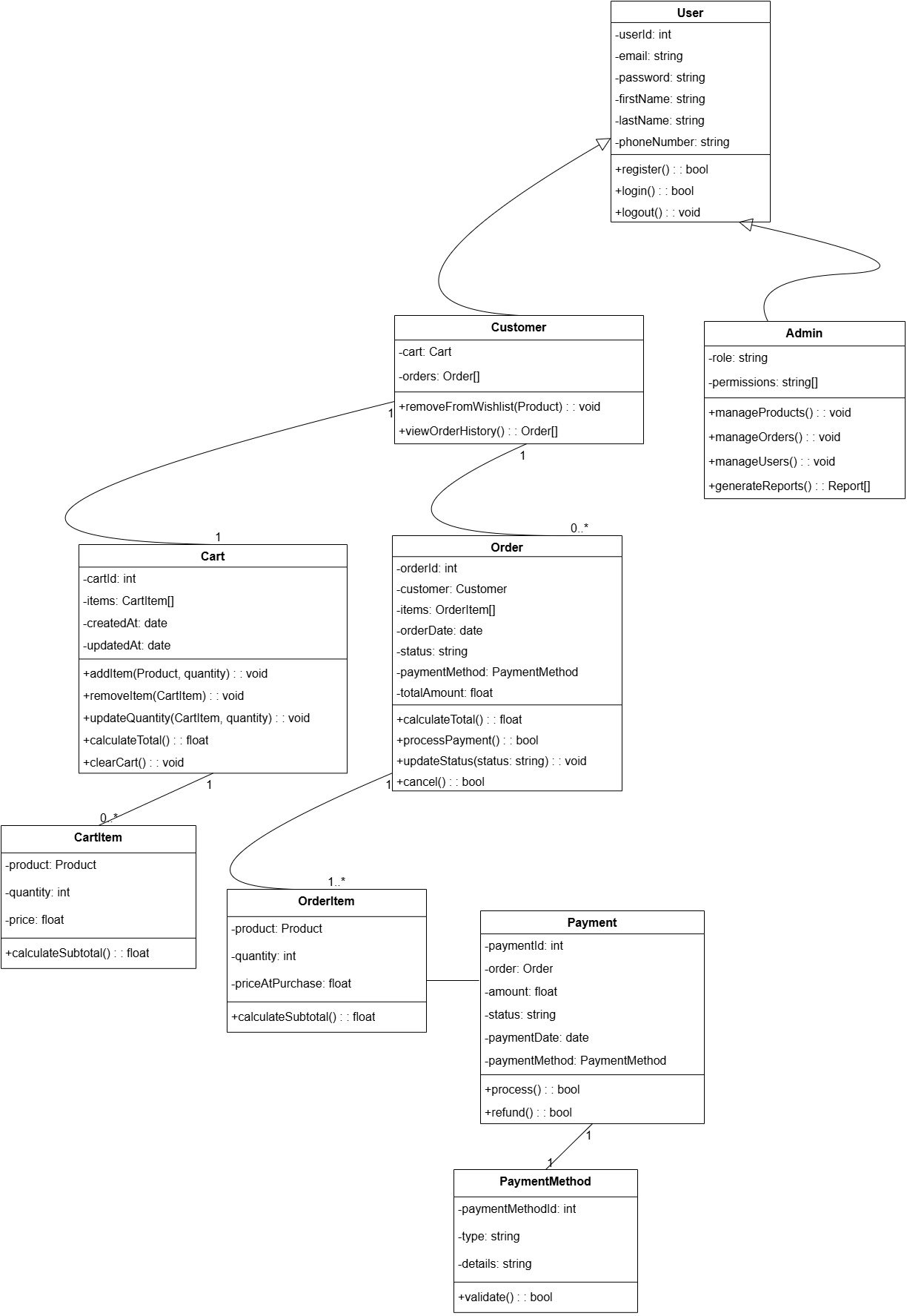
*Figure 3: Timeline of WriteOn*

Schedule feasibility assesses whether the WriteOn platform can be developed and launched within the desired time frame.

Given the project's scope and the use of the MERN stack, the development is expected to take approximately 4–6 months, including phases for design, development, testing, and deployment. The timeline accounts for potential challenges and allows for iterative feedback.

With a well-organized development process and clear milestones, the platform can be delivered on time without compromising quality

### Object Modeling

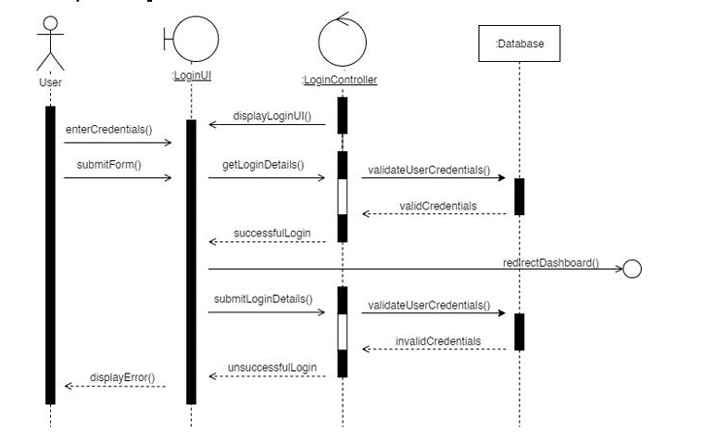
1. 

*Figure 4: Class Diagram of Mernify*

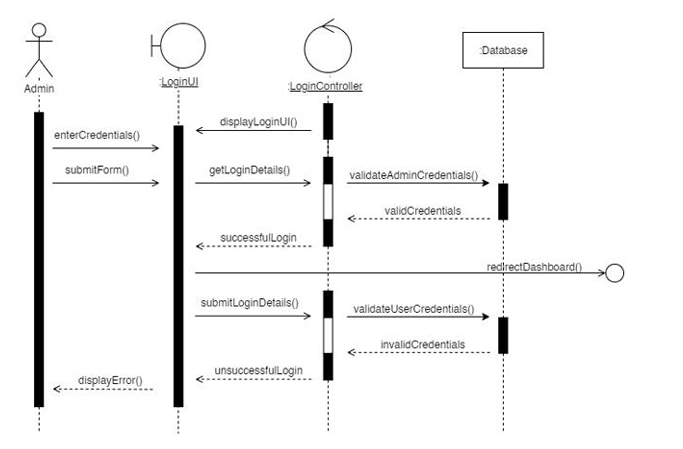
The class diagram of Mernify provides a clear overview of the core objects in the e-commerce platform and how they interact with each other. It defines the system’s structure by showing classes, their attributes, methods (functions), and the relationships among them. The main classes in the Mernify system are: User, Product, Cart, Order, Payment, and Admin. Each class is represented by a rectangle, listing its name, properties (attributes), and functions (methods)

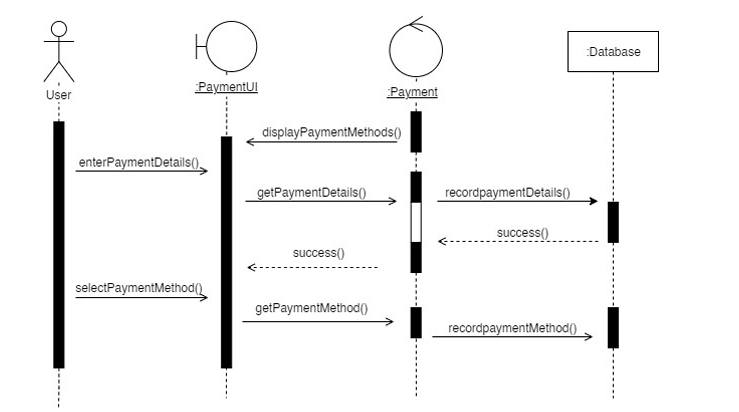
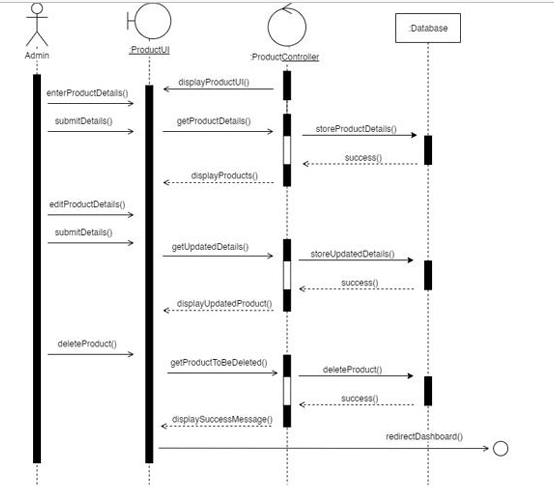
* **User Class**:  
  Represents customers who use the platform. Attributes may include userID, name, email, password, and address. Methods may include register(), login(), logout().
* **Product Class**:  
  Represents items listed for sale. Attributes include productID, name, description, price, stockQuantity, and category. Methods include addProduct(), updateProduct(), deleteProduct().
* **Cart Class**:  
  Represents the shopping cart. Attributes include cartID, userID, and productList. Methods include addItem(), removeItem(), clearCart().
* **Order Class**:  
  Represents the customer’s purchase. Attributes include orderID, userID, cartID, totalAmount, orderDate, and status. Methods include placeOrder(), cancelOrder(), trackOrder().
* **Payment Class**:  
  Handles payment processing. Attributes include paymentID, orderID, amount, paymentDate, and paymentMethod. Methods include processPayment() and refundPayment().
* **Admin Class**:  
  Represents the administrator. Attributes include adminID, name, email, and password. Methods include manageUsers(), manageProducts(), and viewReports().

### Sequence Diagram



*Figure 7: User login sequence diagram*

*Figure 8: Admin login sequence diagram*

*Figure 9: Checkout sequence diagram* **

*Figure 10: Create Product sequence diagram*

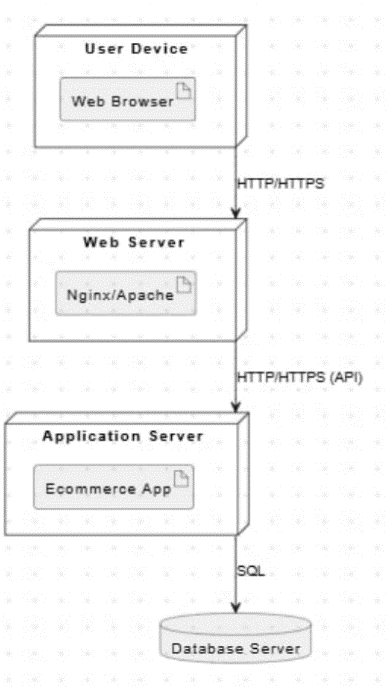
**

Figure 1: Deployment diagram

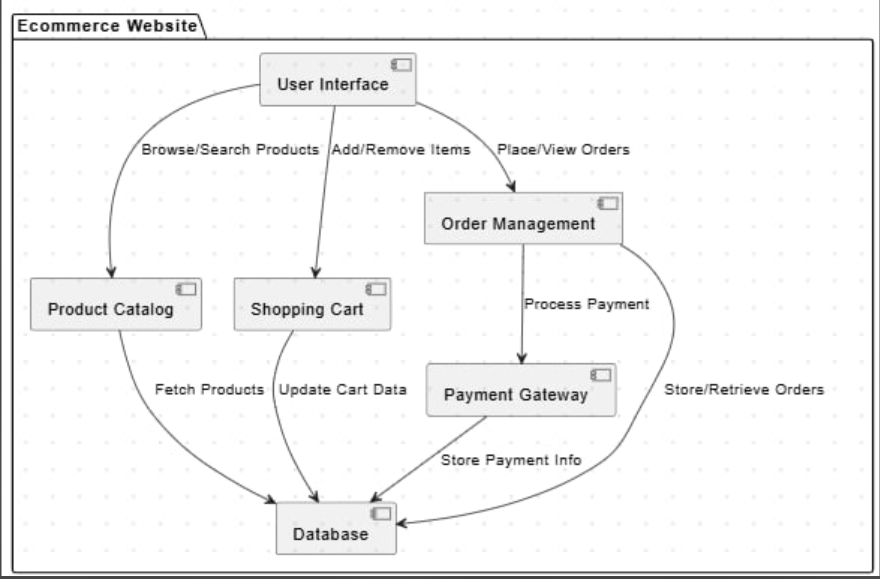
**

Figure 2: Deployment diagram

### Process Modeling

*Figure 5: Activity Diagram of Mernify*

The Activity Diagram of Mernify illustrates the step-by-step flow of activities a user experiences when interacting with the platform, from browsing products to placing an order.

The process begins when a user visits the website, leading to the initial state:

* Browse Products – Users can explore various products listed on the platform.
* If a user is interested in a product, they move to the View Product Details state, where they can see more information about the item.

Once a product is added to the cart:

* Users can choose to either go to the cart (View Cart) or continue shopping.
* In the View Cart state, users can perform actions like updating the quantity of items, removing items, or proceeding to checkout.

During the checkout process:

* Users are prompted to either log in, register, or checkout as a guest.
  + Existing users enter their credentials (Login).
  + New users complete a quick sign-up (Register).

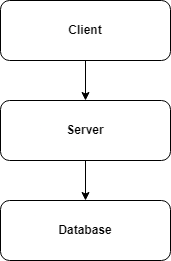
After identification, all users must:

* Enter their shipping information, followed by payment details.
* Upon submission, they reach the Order Confirmation state, where they receive confirmation and options to:
  + Track their order,
  + Continue shopping, or
  + Exit the website.

This activity diagram helps visualize the end-to-end user journey within Mernify, ensuring that the system design supports a smooth, logical, and user-friendly e-commerce experience. It also highlights possible branching paths (e.g., guest vs registered user), making it easier to handle different user flows effectively.

# Chapter 4: System Design

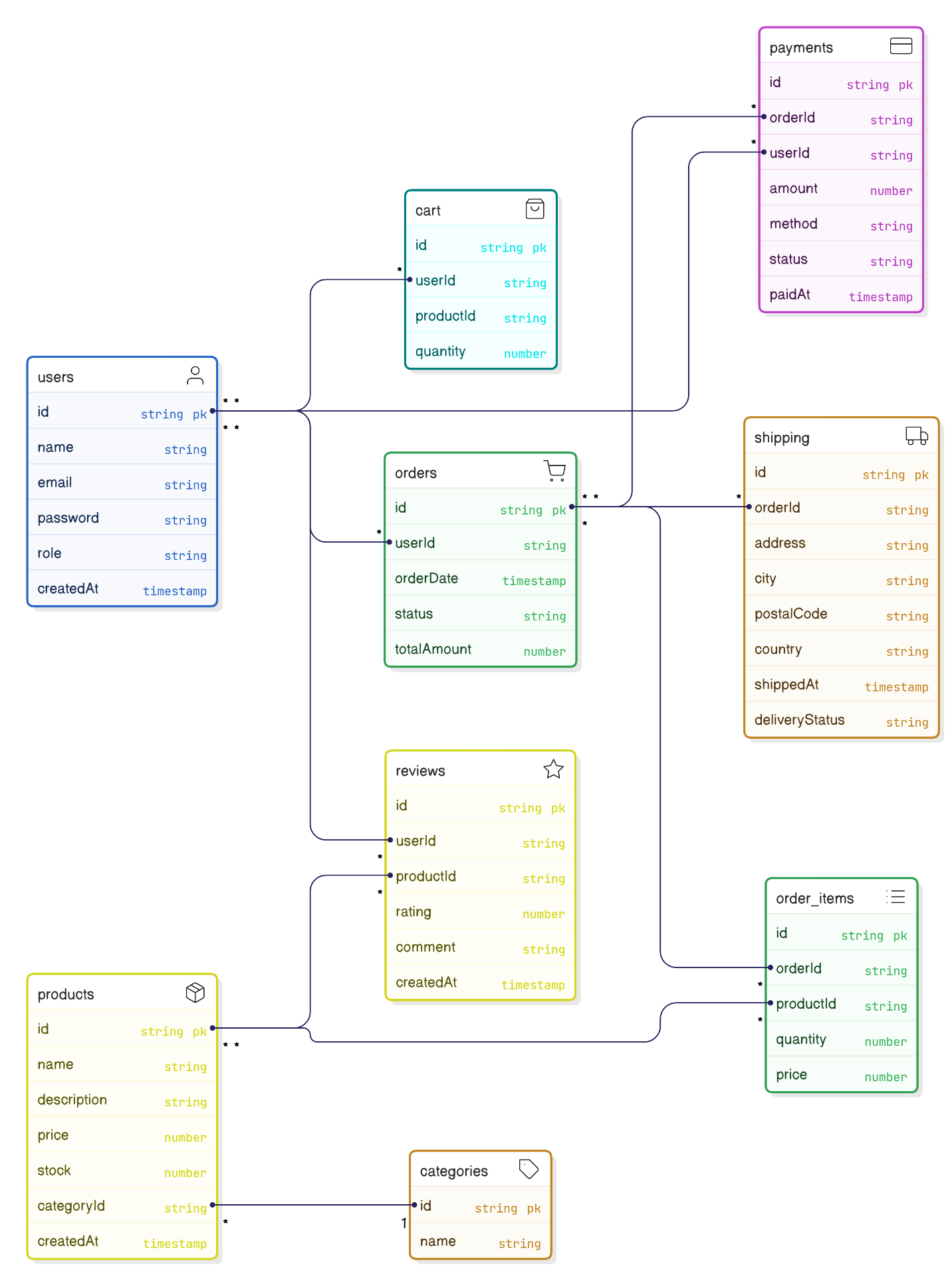
## 4.1 System Architecture and Overview

The System design involves defining the structure, components, and data flow of a system to meet specific goals. For our e-commerce platform, we’ve built a **web-based system** that requires an internet connection, a database, and devices like mobiles or laptops with browser support.

The system follows a **client-server architecture**, where users (clients) interact with the platform through a web interface, while the server handles data processing, product management, and user authentication. This setup ensures smooth performance, scalability, and easy maintenance.

### Database Schema Design

Database schema design is a strategy for constructing a framework for data management. Just like in architecture, a solid database needs to have a blueprint to keep the project on track.



*Figure 9: Basic Database Schema of Mernify*

The Mernify project uses MongoDB, part of the MERN stack, for its database. Unlike traditional relational databases, MongoDB stores data in collections, offering flexibility.

# Chapter 4: Implementation and Testing

## Implementation

### Tools used.

* + - * **Platform:** Windows
      * **Backend Framework**: Node.js with Express.js
      * **Frontend**: ReactJS
      * **Database**: MongoDB
      * **Design and prototyping Tool**: Draw.io, eraser.io

### Implementation Detail of Module

The WriteOn project was implemented using the MERN stack. Node.js and Express.js were used for the backend, while MongoDB served as the database. React.js was used for building the frontend. The application’s design and functionality were custom-built to meet the project's requirements.

## Testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system.

### Test Case

**Project Name: WriteOn**

**Test Case ID:** 1

**Test Title:** Verify login with valid username and password.

**Test Designed Date:**  January 5, 2025

**Pre-conditions:** User has valid username and password.

**Dependencies:**

*Table 1: Test Case 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1. Navigate to login page    * Provide valid username    * Provide valid password    * Click login button | Email: test1@gmail.com  Password: test | User should be able to login | User is navigated to the dashboard with successful login | Pass |
| Provide invalid username and password. | Email: test1@gmail.com  Password: 00 | User login invalid | User login invalid | Pass |

**Post-condition:**

User is validated and the page is redirected to present the user dashboard.

**Test Case ID:** 2

**Test Title:** Blog Post Creation

**Test Designed Date:** January 5, 2025

**Pre-conditions:** User has opened the create post page.

**Dependencies:**

*Table 2: Test Case 2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| User fills all the input fields. | Title: ReactJS | The post is created successfully. | The post is created successfully | Pass |
|  | Tags: Programming  Category: Tech, Programming |  |  |  |
| Click on the create post |  |  |  |  |
| button. |  |  |  |  |
|  |  |  |  |  |

**Post-condition:**

The user’s post has been created.

**Test Case ID:** 3

**Test Title:** Delete user’s post

**Test Designed Date:** January 5, 2025

**Pre-conditions:** UserPostmust be deleted.

**Dependencies:**

*Table 3: Test Case 3*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| Go to post page. | Do you want to delete the blog? | The post is removed. | The post is removed. | Pass |
| Click on the delete |  |  |  |  |
| button. |  |  |  |  |
| . |  |  |  |  |
|  |  |  |  |  |

**Post-condition:**

The post is removed.

**Test Case ID:** 4

**Test Title:** Like/Comment/Share Process

**Test Designed Date:** January 5, 2025

**Pre-conditions:** User should be logged in.

**Dependencies:**

*Table 4: Test case 4*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| Provide all the user login credentials. | Blog Post: Increment Like, Share, Comment | The like count should increment when the "Like" button is clicked and decrement when "Unlike" is clicked.  . | The like count should increment when the "Like" button is clicked and decrement when "Unlike" is clicked. | (Pass) |
| User tries to like/unlike, share and comment on a post |  | The comment should be successfully added to the post and displayed | The comment should be successfully added to the post and displayed |  |
| Click on Like, Comment and share button |  |  |  |  |

**Post-condition:**

The admin should review the post and the user details page of the user .

**Test Case ID:** 5

**Test Title:** Graph Data Accuracy

**Test Designed Date:** January 5, 2025

**Pre-conditions:** User has several blog posts with recorded likes, shares, and comments.

*Table 5: Test Case 5*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| Manually calculate the number of likes, shares, and comments from the database | Post1: Like:10, Shares:5, Comment:2  Post2:Like:2,Shares:0, Comment:1 | Data in the graph matches the manual count from the database | Data in the graph matches the manual count from the database | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Post-condition:**

The graph data is up-to-date and visually represents the interaction trends.

# Chapter 5: Conclusion and Future Recommendation

## Lesson Learnt / Outcome

In the eCommerce website project, we gained hands-on experience in building a complete full-stack application, from designing user-friendly interfaces to managing databases and server-side operations. Implementing payment gateway integrations like eSewa and Khalti taught us how to securely handle real-time transactions, adding practical knowledge of API integration and financial workflows. We also learned how to manage product listings, user authentication, and order tracking effectively. Overall, this project strengthened our understanding of full-stack development and equipped us with the skills to confidently develop and manage real-world commercial applications.

## Conclusion

MERNIFY is a feature-rich eCommerce platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js). It enables users to browse products, manage their carts, and place orders with ease. The platform integrates popular Nepali payment gateways like eSewa and Khalti, offering a smooth and secure checkout experience. With functionalities like user authentication, product management, and order tracking, MERNIFY showcases the capabilities of the MERN stack in building scalable and responsive online shopping systems. The project is further supported by comprehensive system diagrams, including use case, ER, activity, and sequence diagrams, which effectively illustrate the platform’s architecture and flow.

## Future Recommendations

# ● Enhance the design, implementation, and documentation to improve usability and overall performance.

# ● Optimize the website to load efficiently, even on slower internet connections.

# ● Develop an Android application to make the eCommerce platform accessible on mobile devices.

# ● Integrate email and SMS notifications to keep users updated on order status and promotional offers.

# ● Continuously update the system based on user feedback and changing market needs.

# ● Introduce a payment system for premium features, such as featured product listings or seller tools.

# ● Extend the platform to support multiple vendors, turning MERNIFY into a full marketplace.

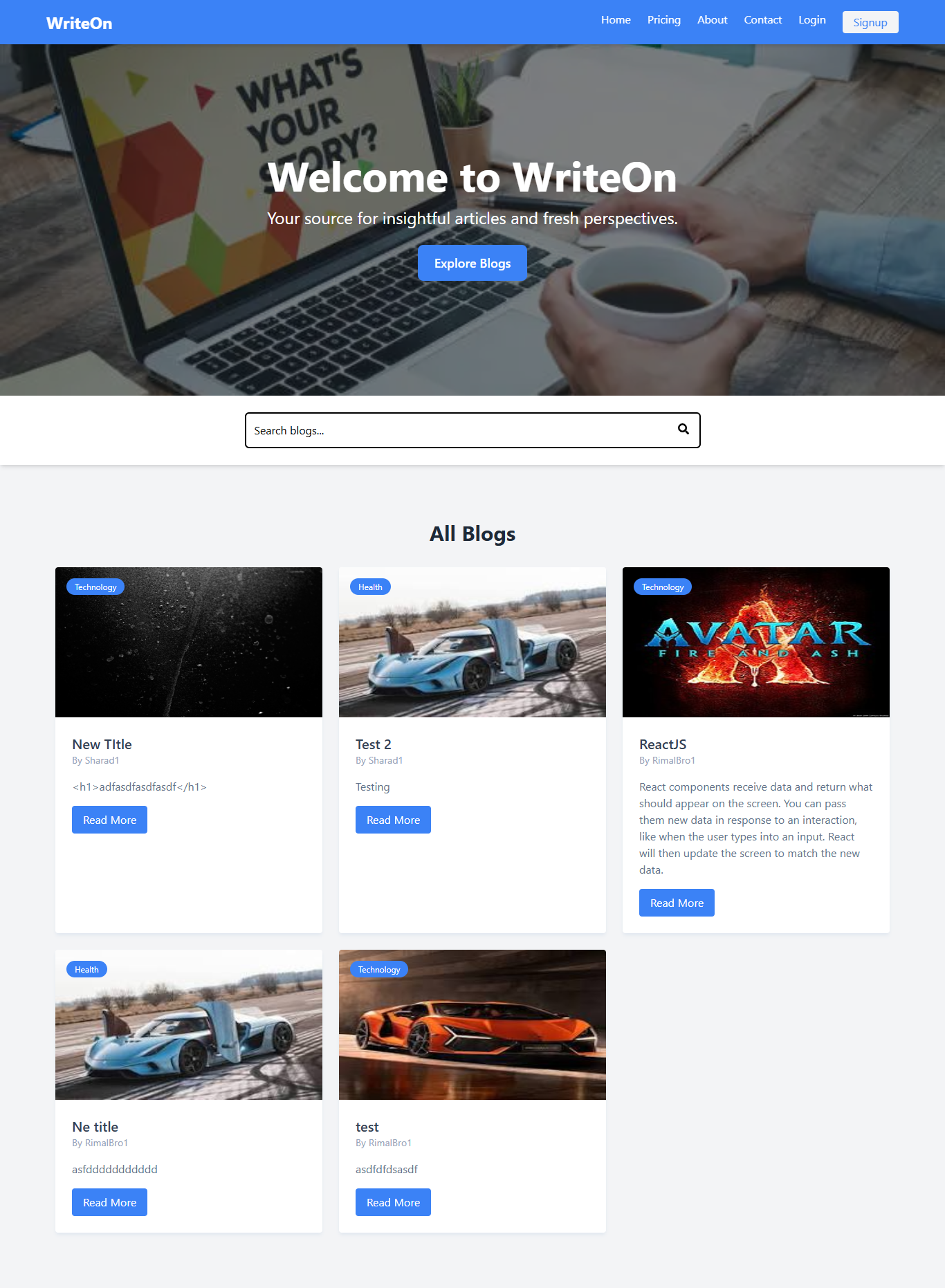
# References

|  |  |
| --- | --- |
| [1] | G. Thevenot, "Blogging as a Social Media," 2007. |
| [2] | J. S. Q. Y. Z. C. D Shen, "Latent friend mining from blog data," 2006. |
| [3] | P. M. H. a. S. P. J. T. Child, Blogging privacy rule orientations, privacy management, and content deletion practices, 2012. |

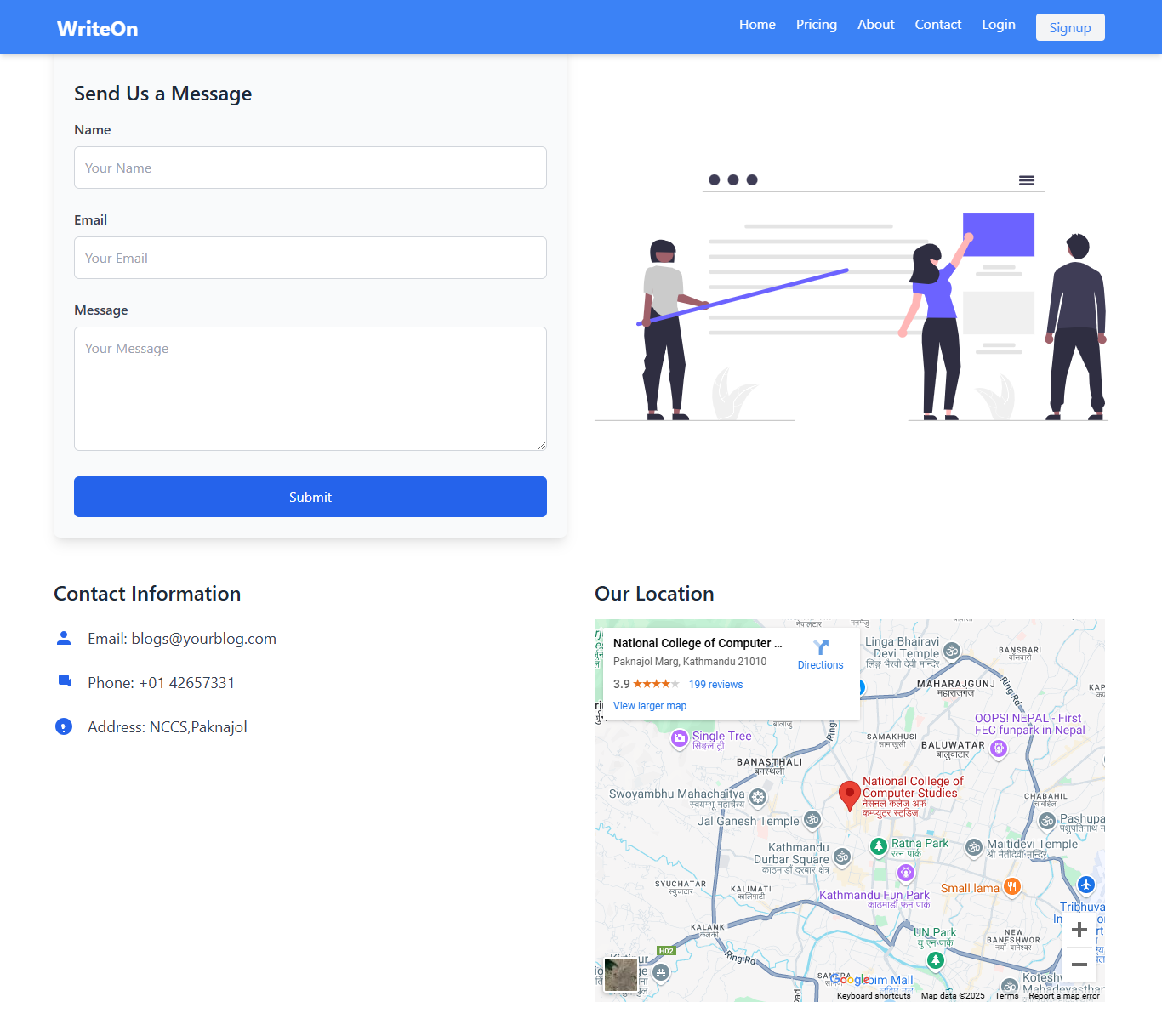
# 

# Appendices

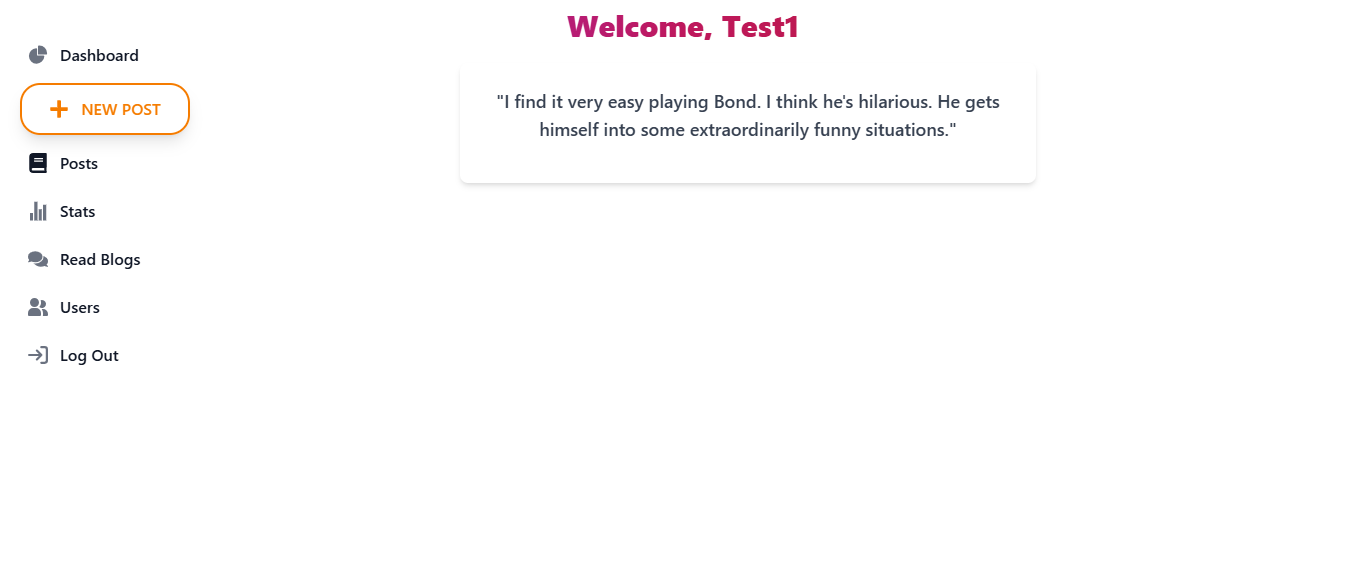
**Snapshots of Output:**

****

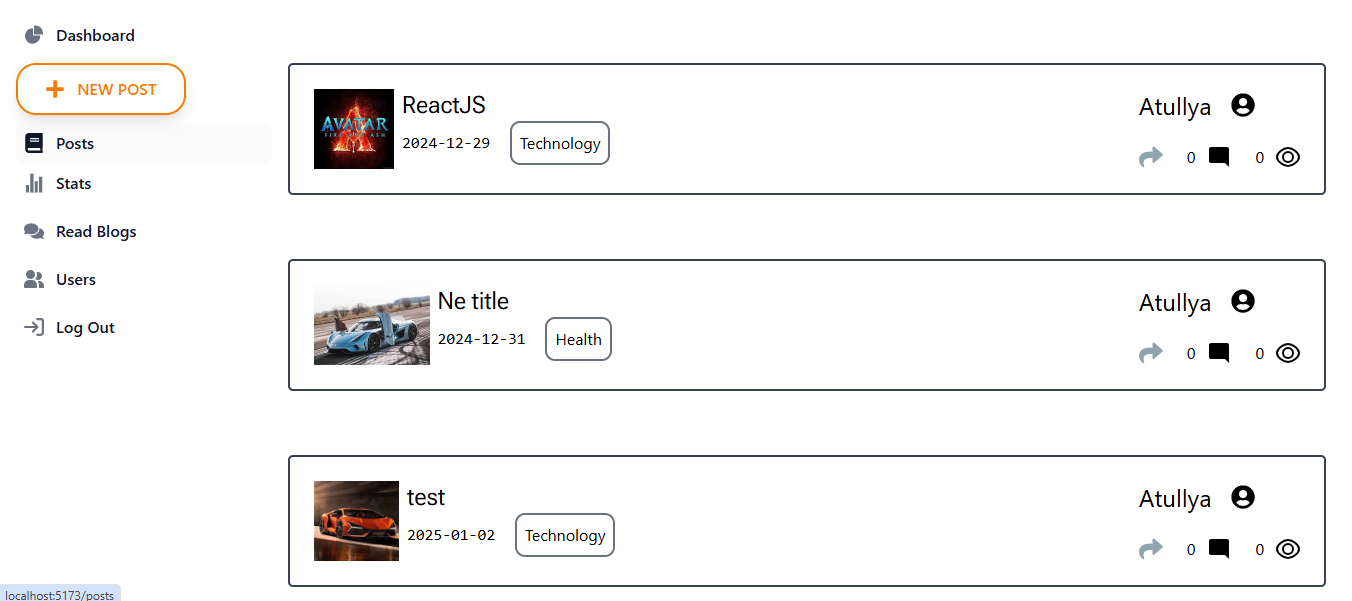
*Figure 10: Homepage of WriteOn*



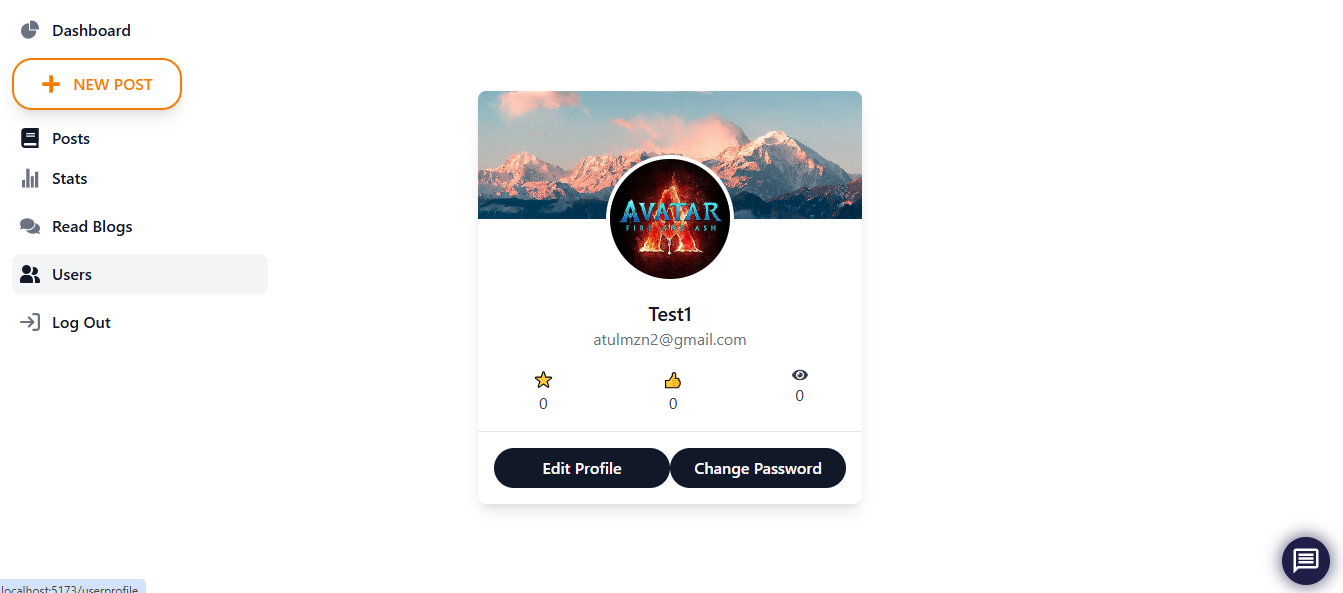
*Figure 11: Contact Page of WriteOn*

**

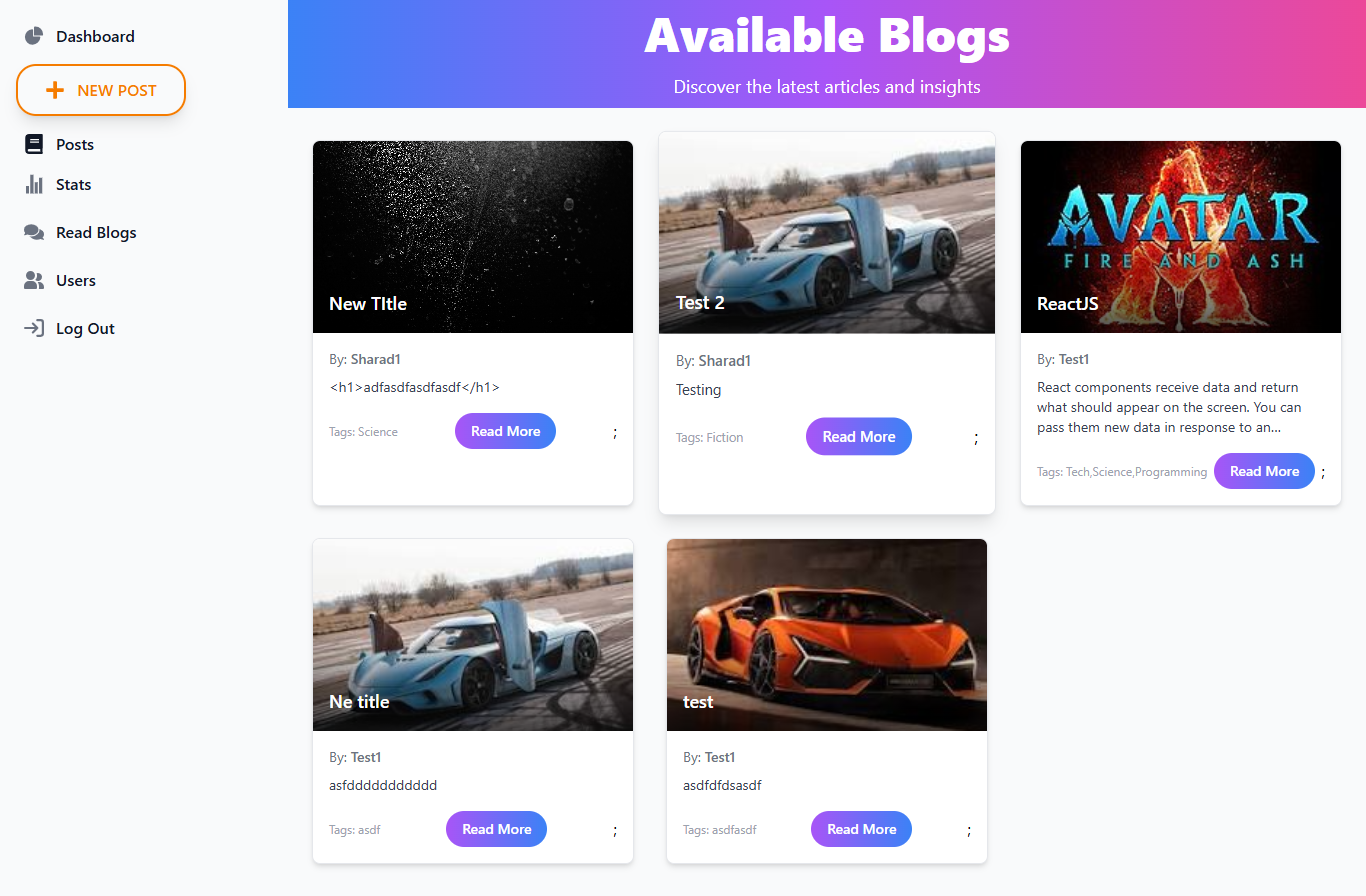
*Figure 12:User Home Page of WriteOn*



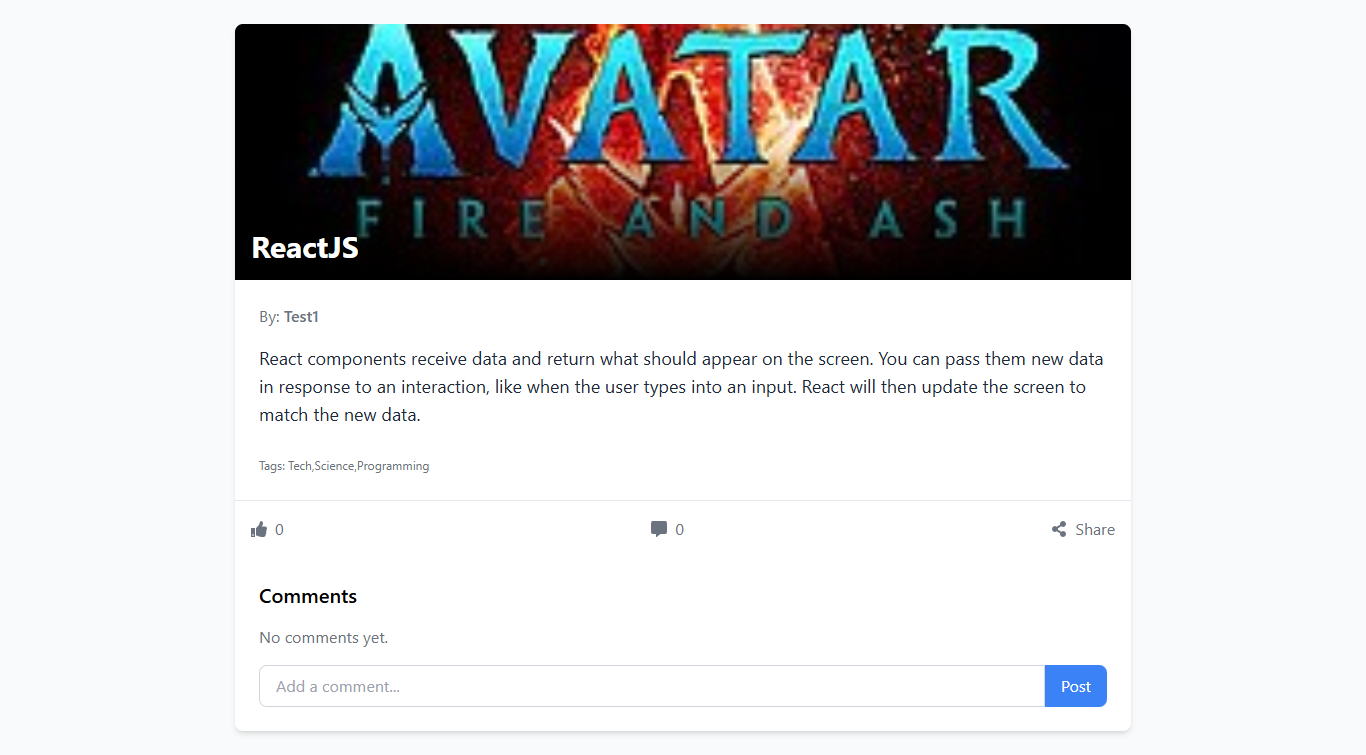
*Figure 13: User Post Page of WriteOn*



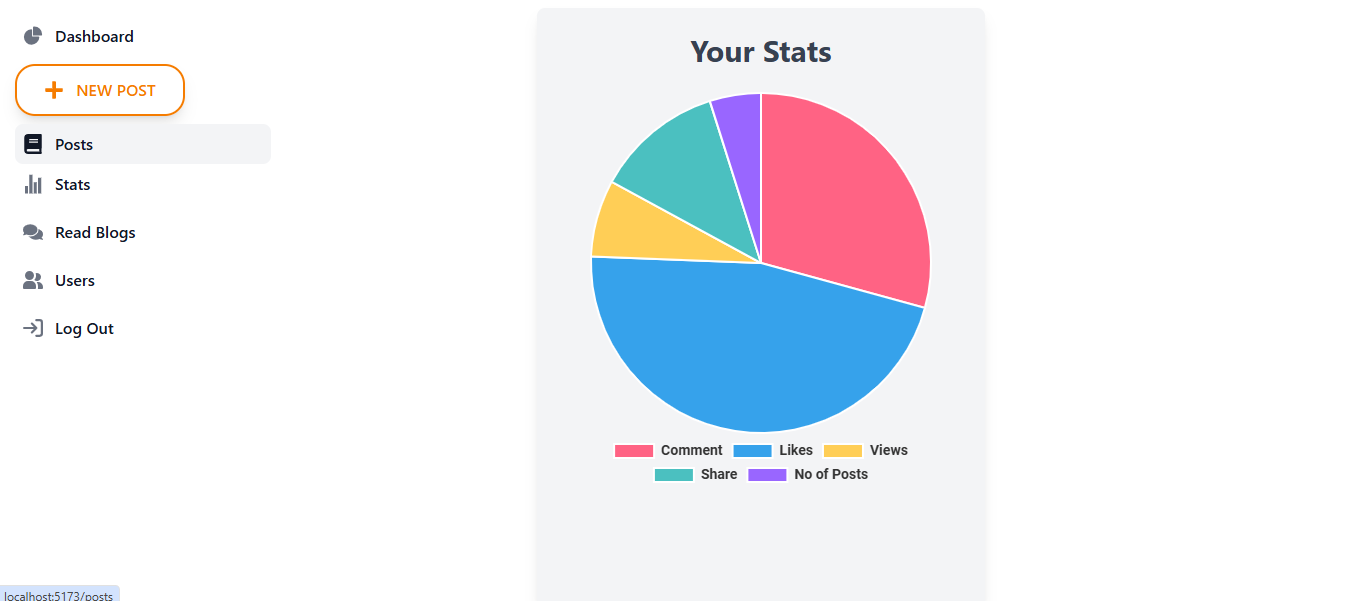
*Figure 14: User Profile of WriteOn*

**

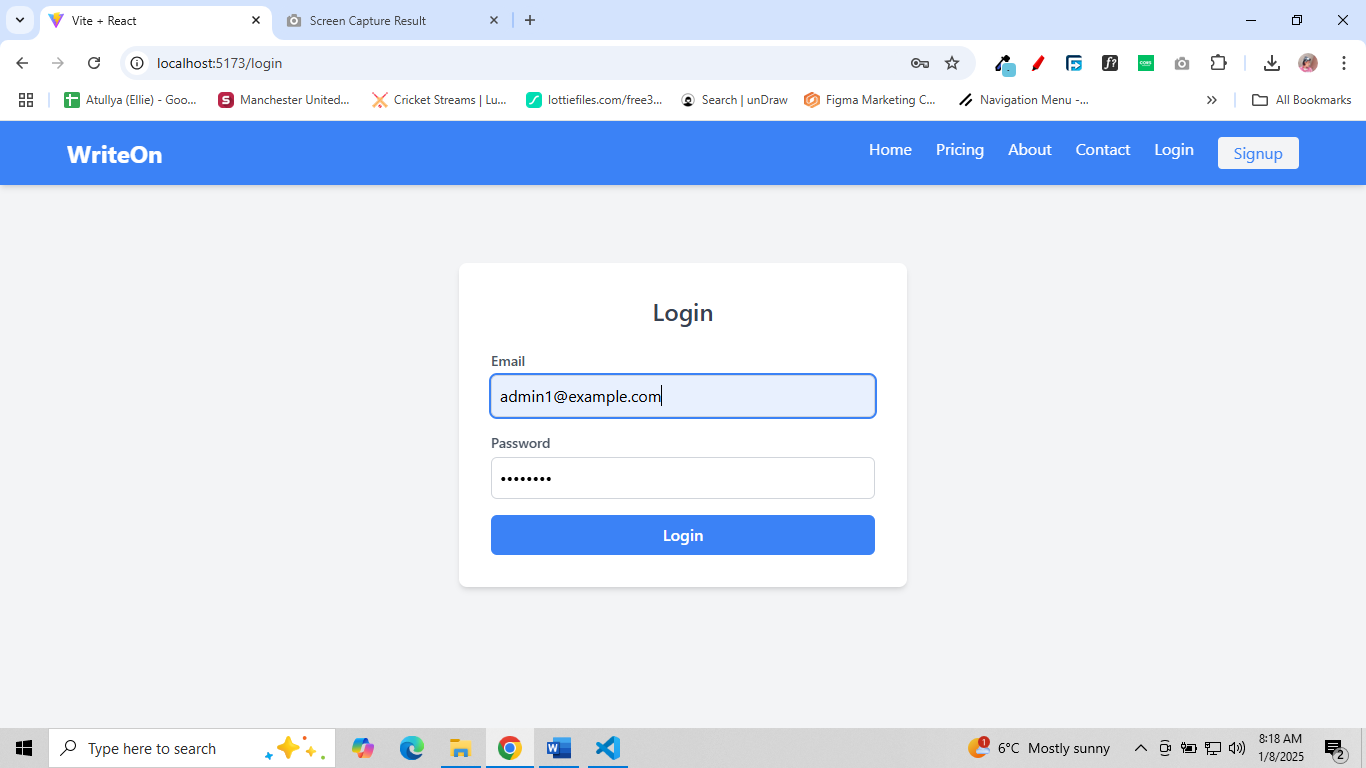
*Figure 15: Quick Blog View of WriteOn*



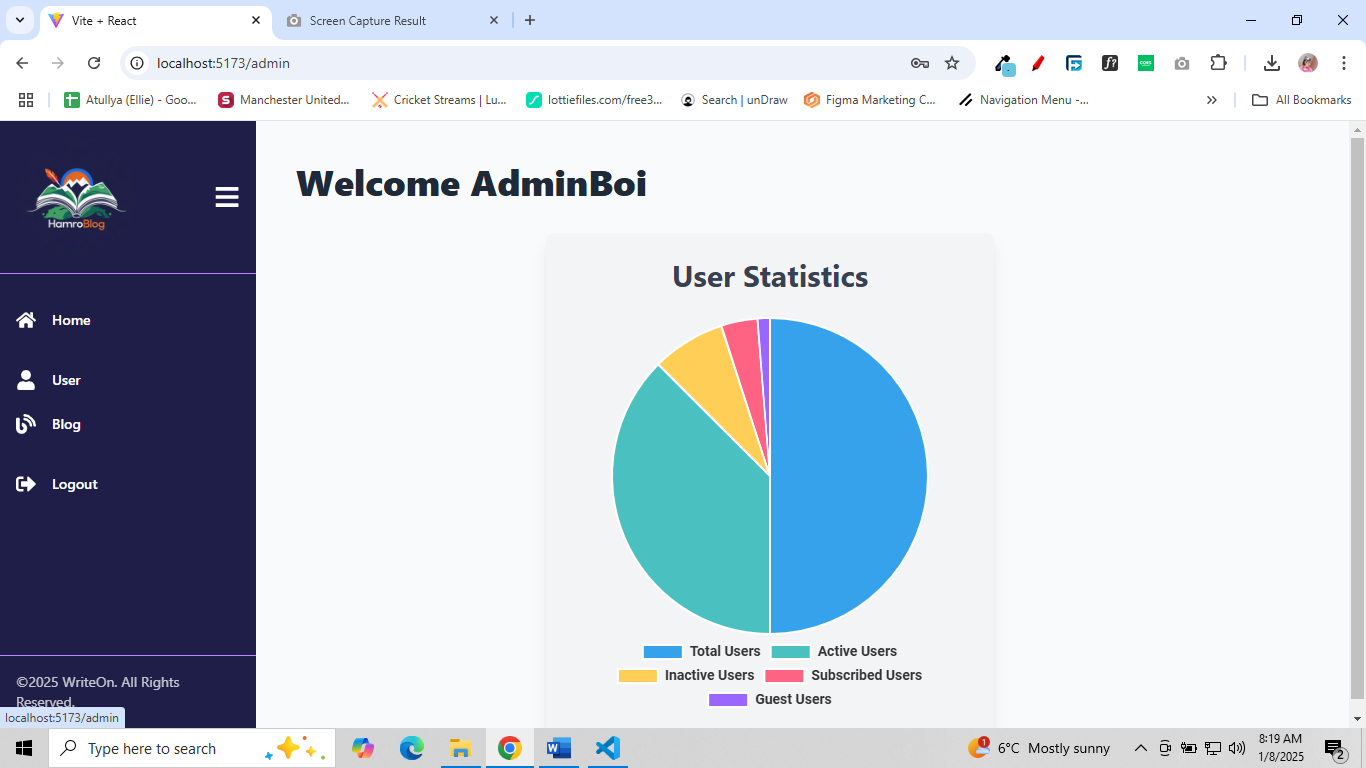
*Figure 16: Blog preview Page of WriteOn*



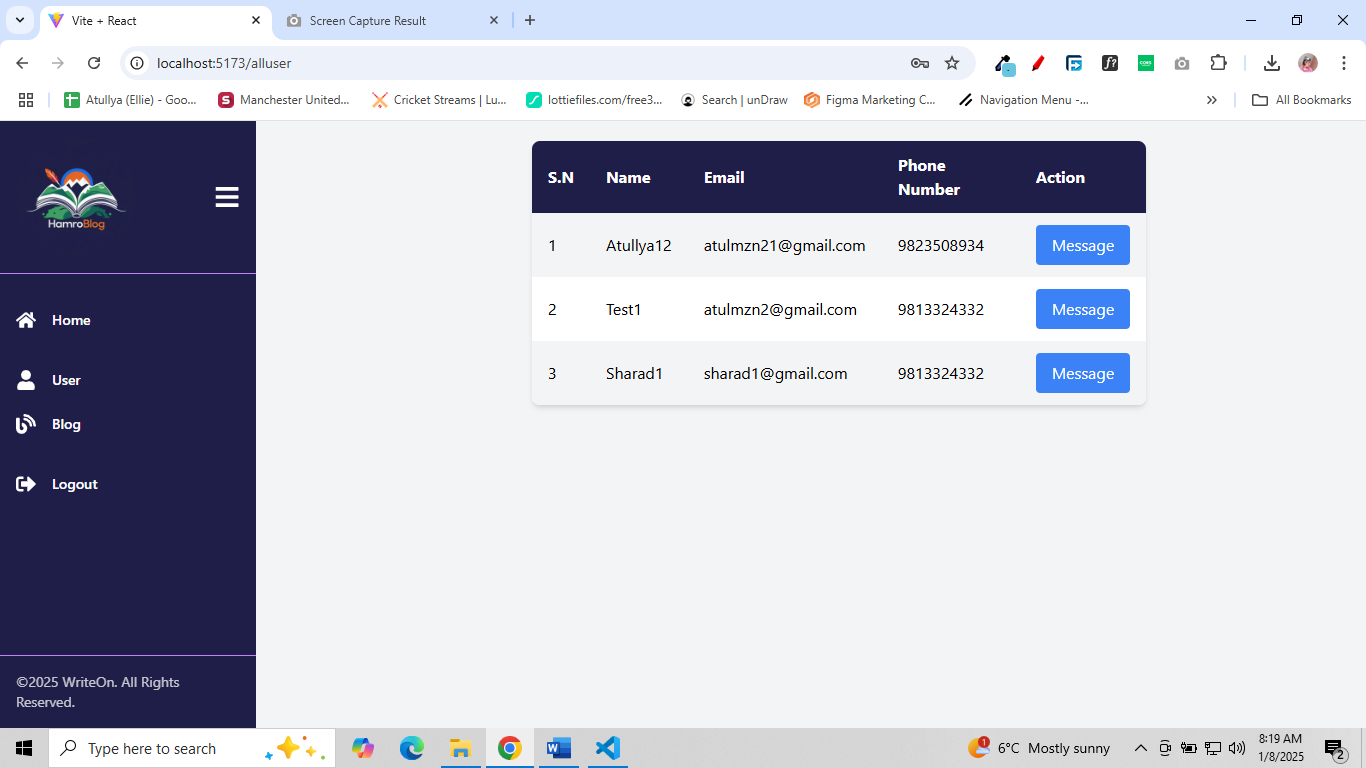
*Figure 17:User Stat Page of WriteOn*



*Figure 18: Login Page of WriteOn*



*Figure 19: Admin Dashboard of WriteOn*



*Figure 20: Available User of WriteOn*