

---

# **Design Document**

**for**

## **Re\_Store**

**Version 1.0**

**Prepared by**

**Group 4**

**Group Name : Divide and Conquer**

Bharatula Anirudh Srivatsa	230290	<a href="mailto:bharatula23@iitk.ac.in">bharatula23@iitk.ac.in</a>
Voora Rakesh	231174	<a href="mailto:vrakesh23@iitk.ac.in">vrakesh23@iitk.ac.in</a>
Soma Koushik	231018	<a href="mailto:koushiks23@iitk.ac.in">koushiks23@iitk.ac.in</a>
Sanjay Raghav Vangala	230916	<a href="mailto:sanjayra23@iitk.ac.in">sanjayra23@iitk.ac.in</a>
Yashwanth Reddy Junutula	231194	<a href="mailto:yashwanth23@iitk.ac.in">yashwanth23@iitk.ac.in</a>
Saatvik Gundapaneni	230428	<a href="mailto:gundapanen23@iitk.ac.in">gundapanen23@iitk.ac.in</a>
Vempati Prem Santhosh	231137	<a href="mailto:psanthosh23@iitk.ac.in">psanthosh23@iitk.ac.in</a>
Ayush Yadav	230272	<a href="mailto:ayushydv23@iitk.ac.in">ayushydv23@iitk.ac.in</a>
Meghana Kadari	230512	<a href="mailto:kmeghana23@iitk.ac.in">kmeghana23@iitk.ac.in</a>
Chapati Venkata Pranaya	230324	<a href="mailto:chapative23@iitk.ac.in">chapative23@iitk.ac.in</a>

**Course: CS253**

**Mentor TA: Jeswaanth Gogula**

**Date of Submission: 7th February 2025**

## CONTENTS

<b>CONTENTS .....</b>	<b>2</b>
<b>REVISIONS .....</b>	<b>3</b>
<b>1 CONTEXT DESIGN .....</b>	<b>4</b>
1.1 <a href="#"><u>CONTEXT MODEL</u></a> .....	4
1.2 <a href="#"><u>HUMAN INTERFACE DESIGN</u></a> .....	6
<b>2 ARCHITECTURE DESIGN .....</b>	<b>14</b>
<b>3 OBJECT-ORIENTED DESIGN .....</b>	<b>17</b>
3.1 <a href="#"><u>USE CASE DIAGRAM</u></a> .....	17
3.2 <a href="#"><u>CLASS DIAGRAM</u></a> .....	21
3.3 <a href="#"><u>SEQUENCE DIAGRAM</u></a> .....	22
3.4 <a href="#"><u>STATE DIAGRAM</u></a> .....	29
<b>4 PROJECT PLAN .....</b>	<b>30</b>
<b>5 OTHER DETAILS .....</b>	<b>33</b>
<b>APPENDIX A - GROUP LOG .....</b>	<b>34</b>

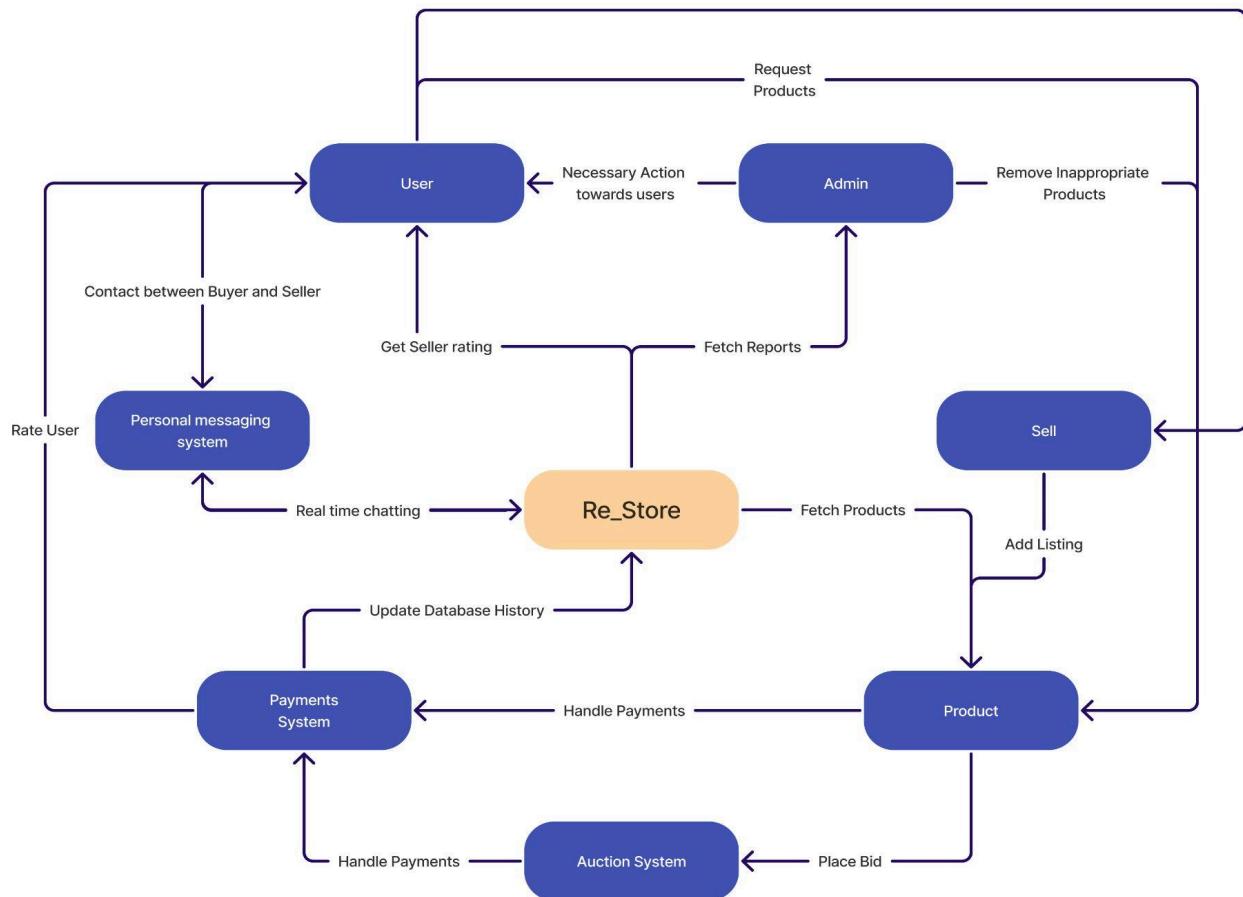
## **Revisions**

<b>Version</b>	<b>Primary Author(s)</b>	<b>Description of Version</b>	<b>Date Completed</b>
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

# 1 CONTEXT MODEL

## 1.1 Context Model

**Re\_Store** is a digital marketplace designed for the campus community, making buying, selling, and connecting seamless. It integrates multiple features that ensure secure transactions, effective communication, and a smooth user experience. Each component works together to create a trusted and efficient platform.



The key subsystems are:

- **User Profiles:** Every user has a profile displaying their listed products, past transactions, and ratings. This builds credibility, helping buyers and sellers interact with confidence.
- **Product Listings & Search:** Users can list products for sale, whether at a fixed price or through an auction. A search system with filters like category, price, and condition makes it easy to find items. Buyers can also post product requests for items that are not currently available.
- **Buy Now and Auctions:** Sellers can choose to list their products with a fixed price or as an auction. If the seller sets a fixed price, buyers can purchase the product using the Buy Now option. Buyers can place bids if the product is listed as an auction, and the highest bid wins.
- **Messaging System:** Buyers and sellers can communicate through a secure, real-time messaging system. This ensures smooth negotiations and clarifies any details before a purchase.
- **Favorites & Notifications:** Users can **favorite** products they're interested in and receive notifications about new listings, auction updates, messages, and important announcements.
- **Secure Payments:** A built-in payment system guarantees safe transactions, encrypting sensitive information and ensuring reliability.
- **Reviews & Feedback:** Users can leave ratings and reviews after completing a transaction. This promotes openness and builds trust in interactions.

## 1.2 Human Interface Design

### 1.2.1 Signup Page

## Sign Up

Username \*

Full Name\*

Email \*

Password\* (eye icon)

Confirm Password \* (eye icon)

Submit



### 1.2.2 Login Page

## Welcome to our Page

### Log In

Username/Email address\*

Password\* (eye icon)

[Forgot Password?](#)

Keep me logged in

Sign In

Don't have an account? [Sign up](#)

[Admin login](#)



### 1.2.3 Forgot Password Page

### Forgot Password

Enter your email and we'll send a verification code to reset your password

  
  
[← Back to login](#)

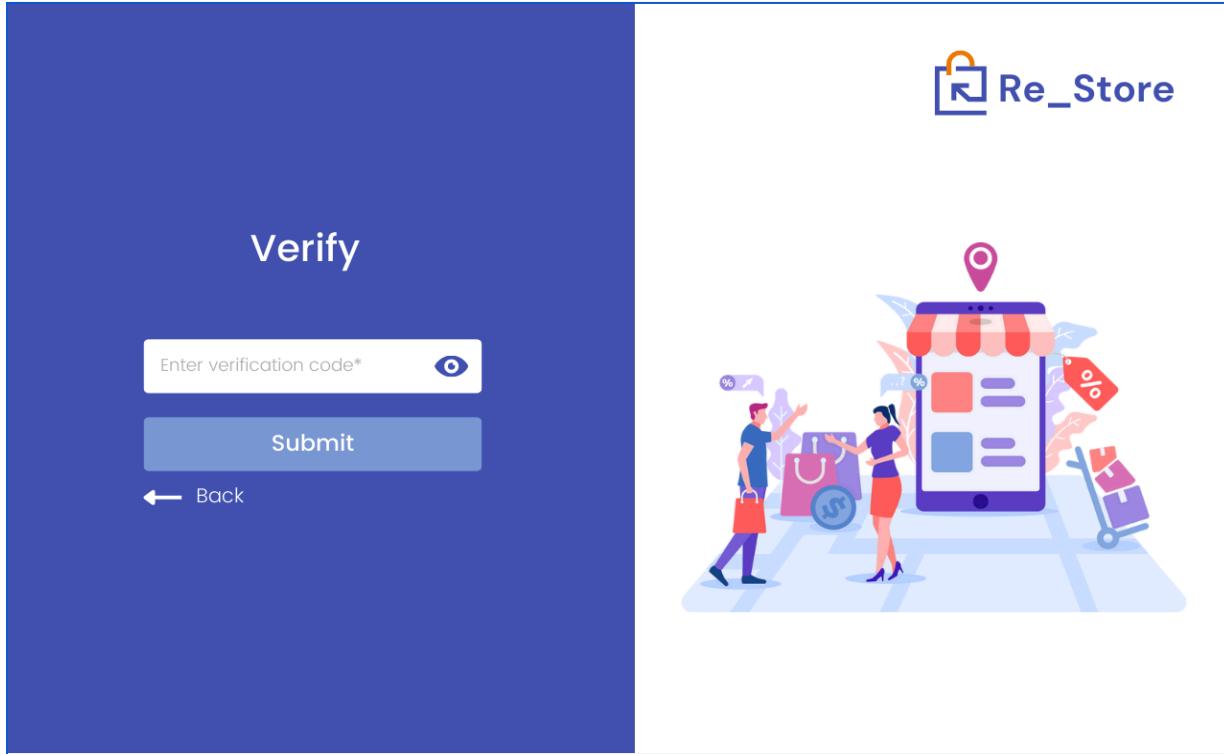


### 1.2.4 Reset page

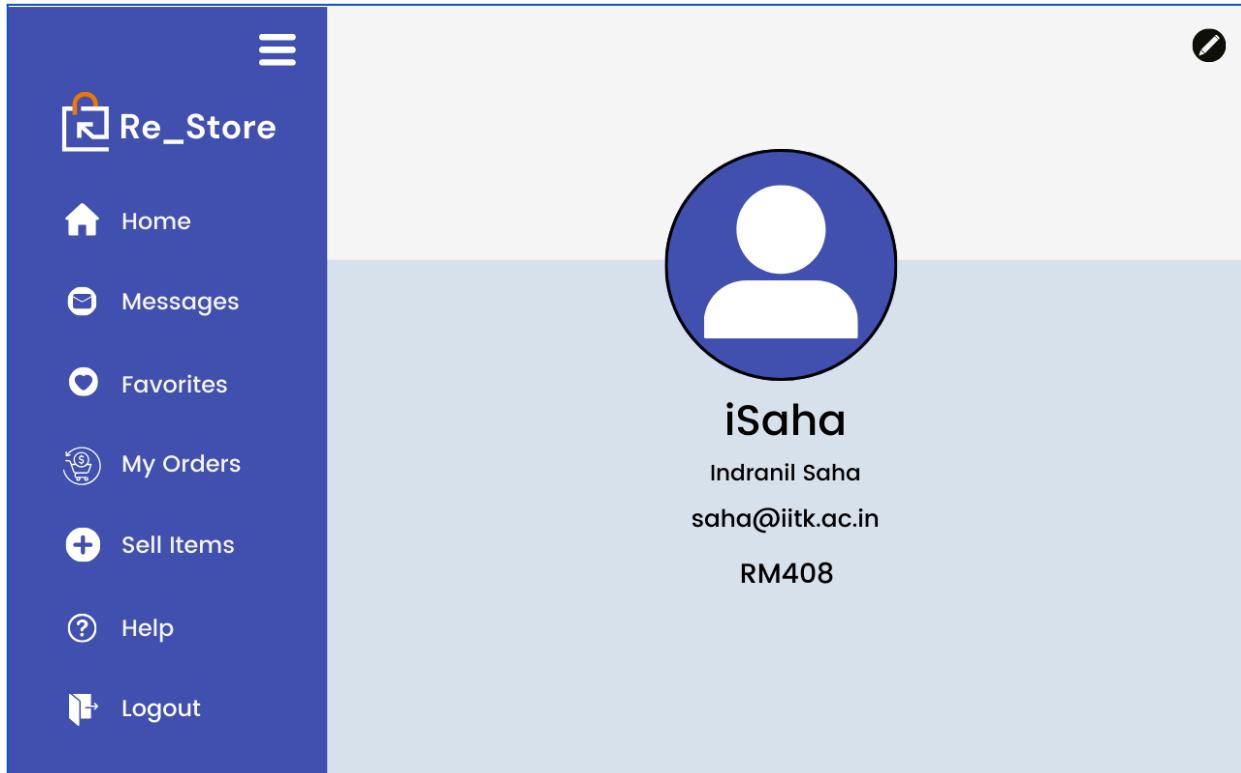
### Reset Password



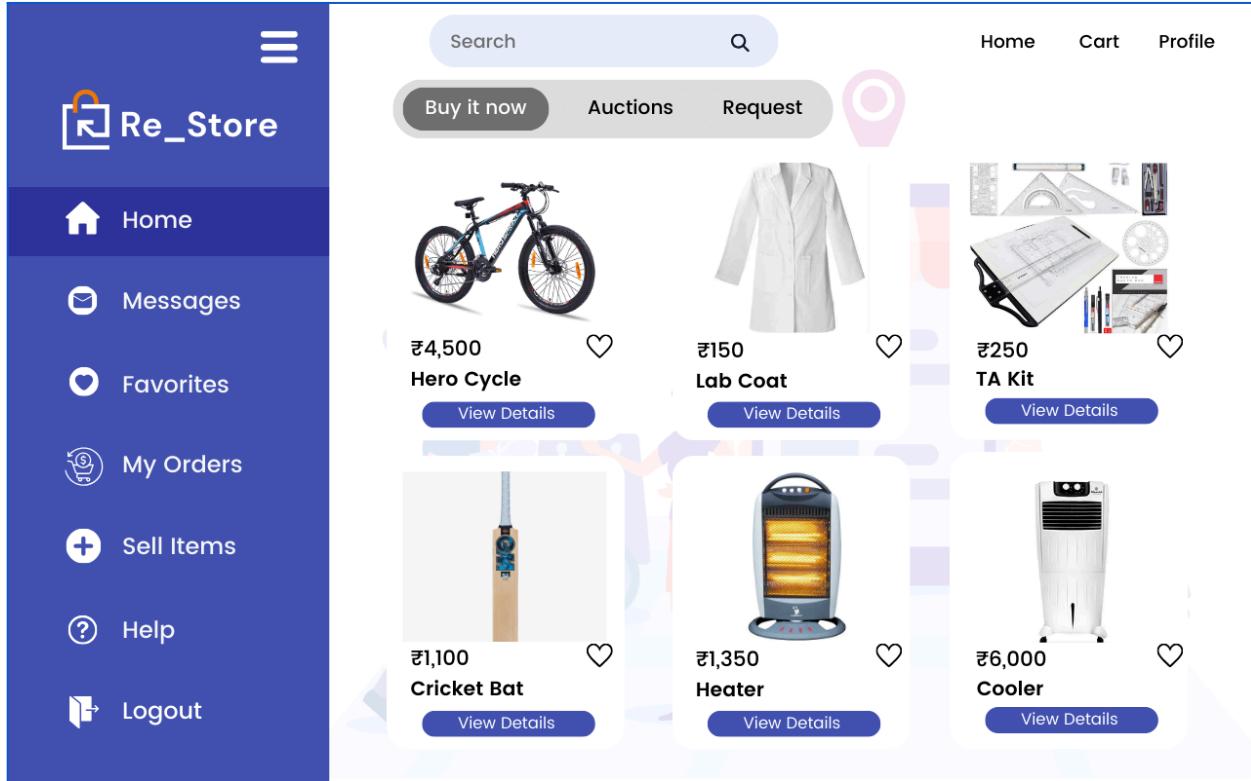
### 1.2.5 Verification Page



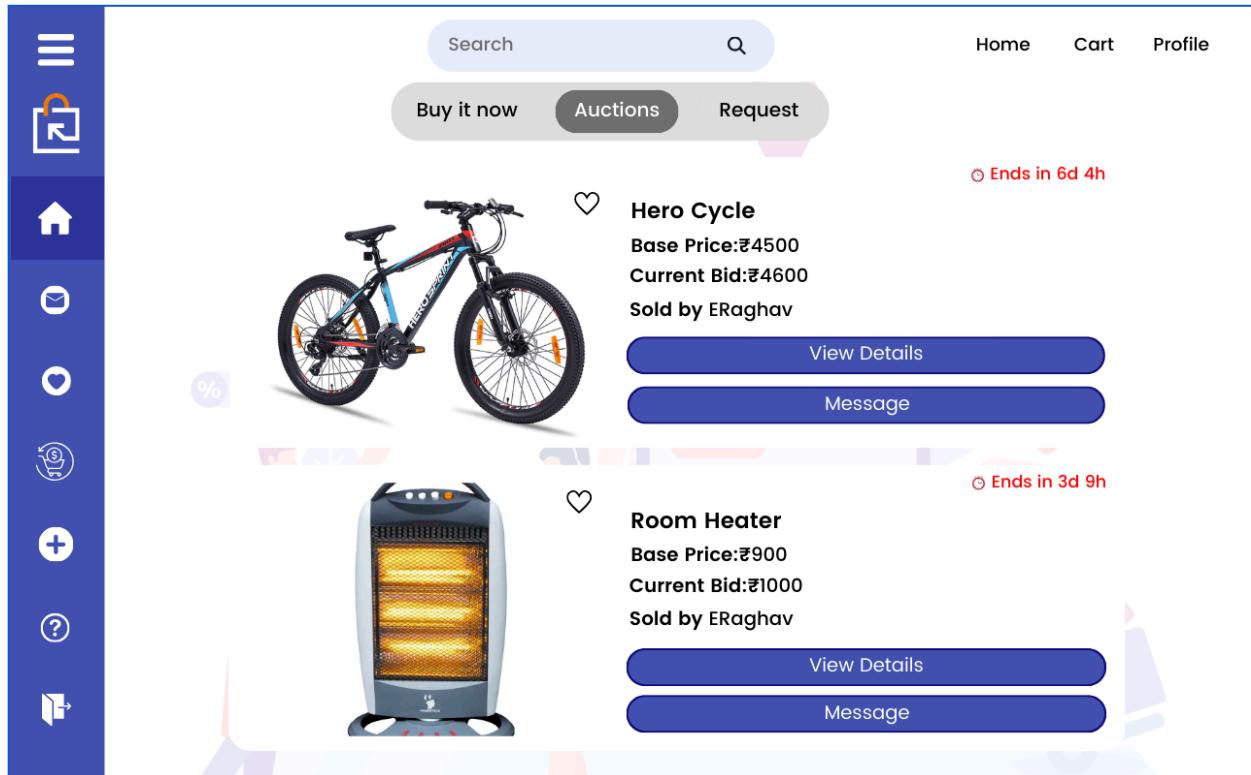
### 1.2.6 Profile Page



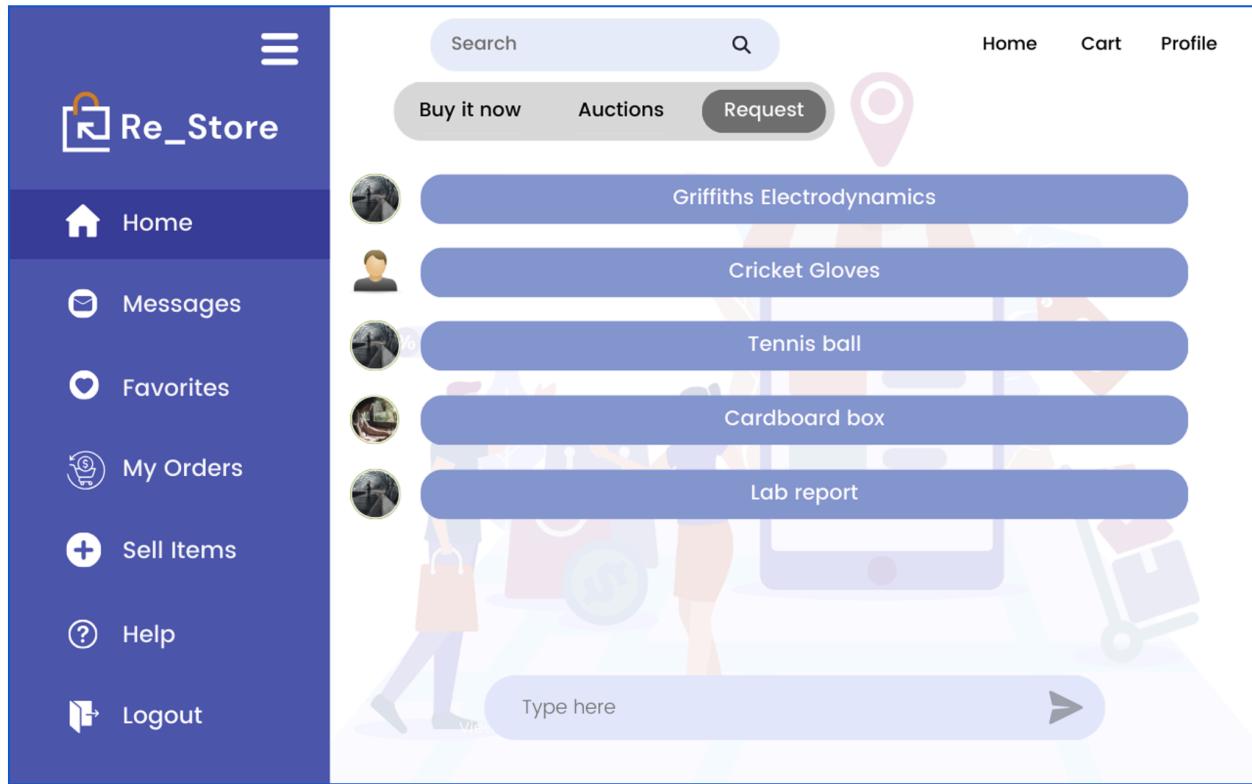
## 1.2.7 Home Page (Direct Sale)



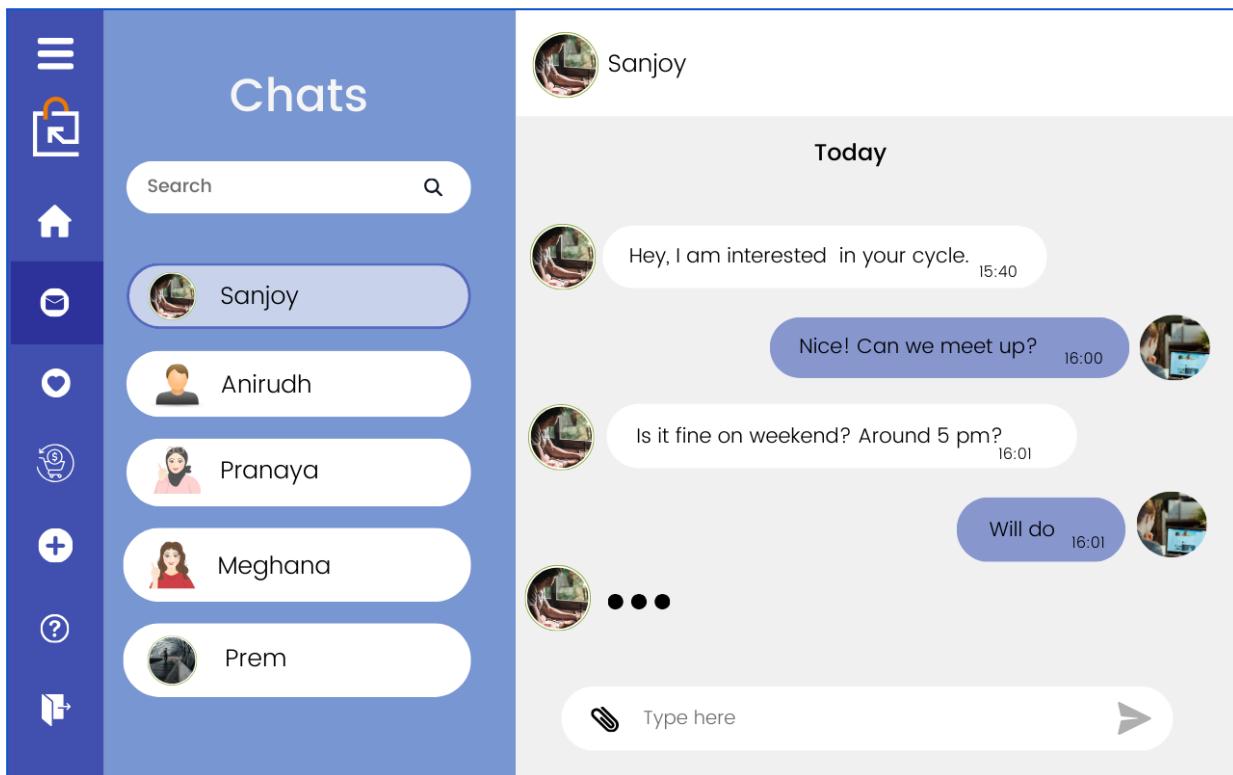
## 1.2.8 Home Page (Auctions)



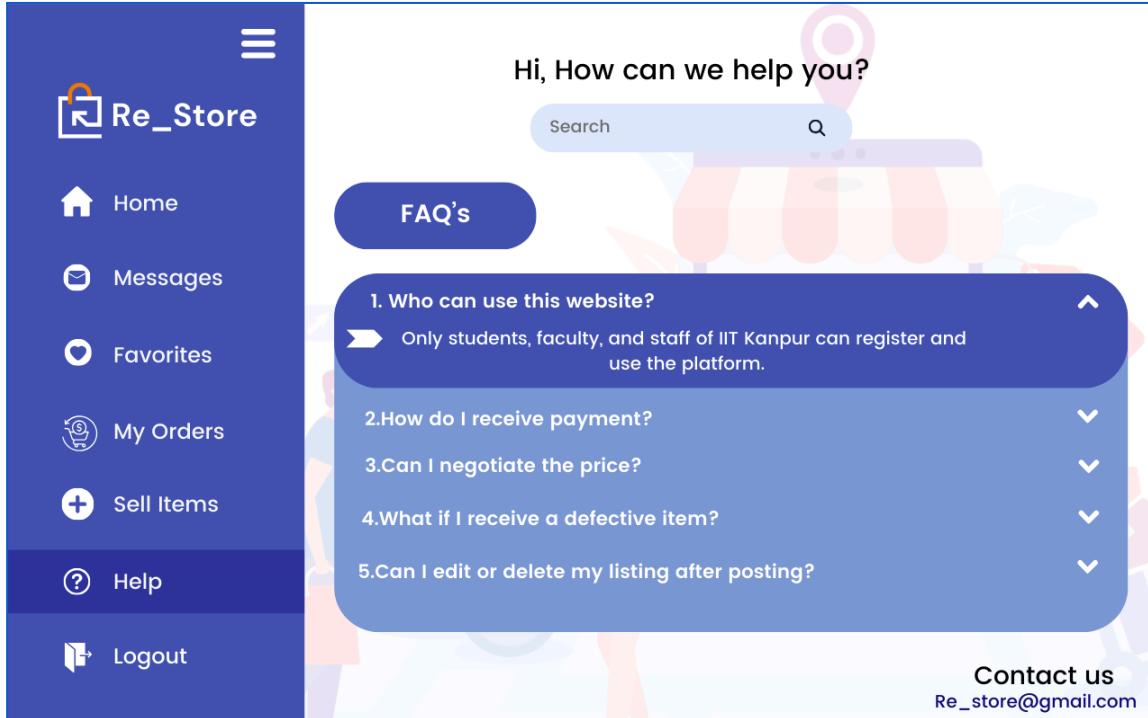
### 1.2.9 Home Page (Product Request)



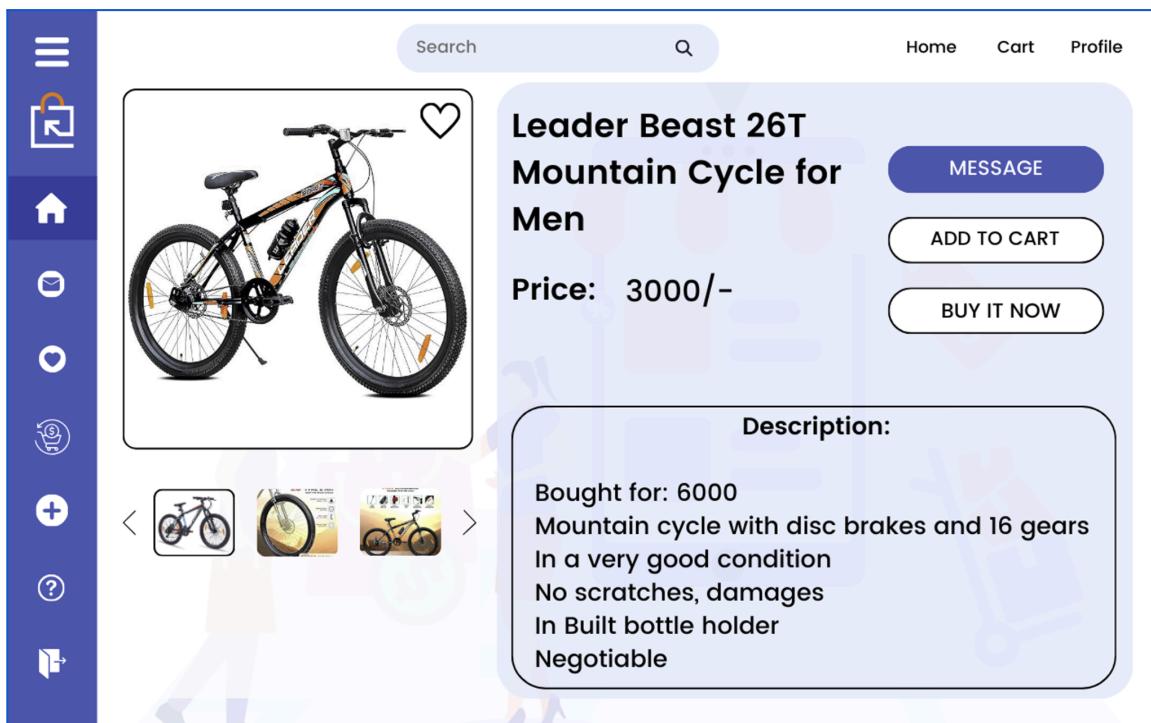
### 1.2.10 Chat Page



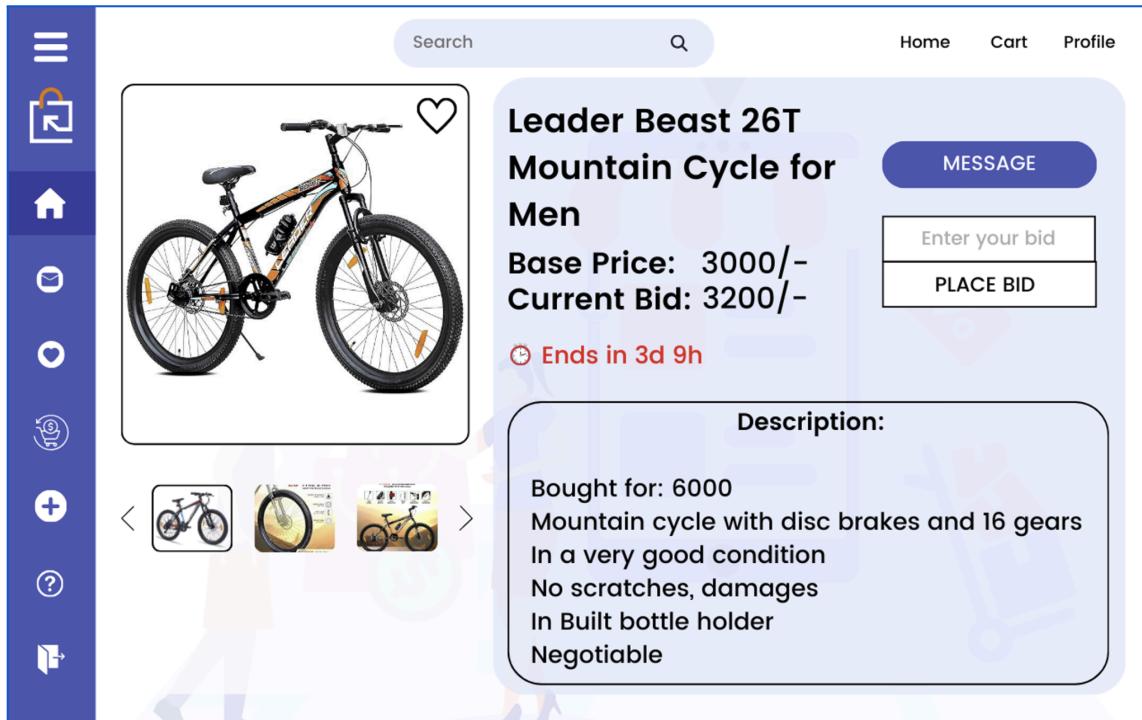
## 1.2.11 Help Page



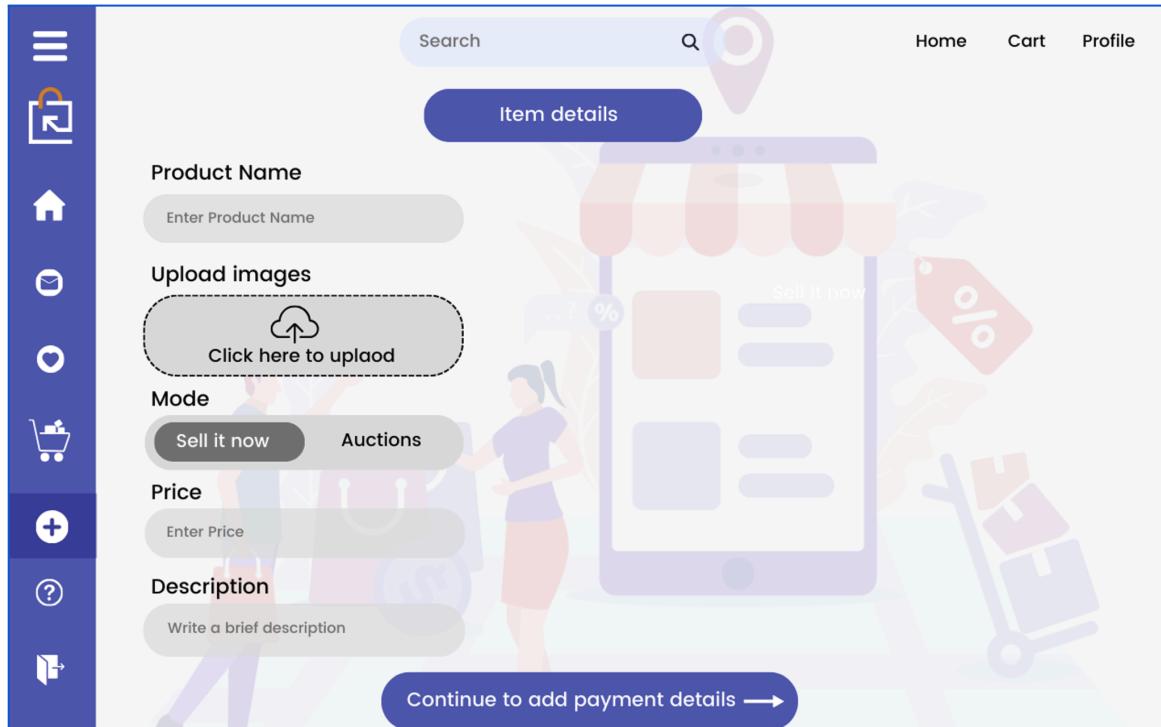
## 1.2.12 Product View Page (Direct Sale)



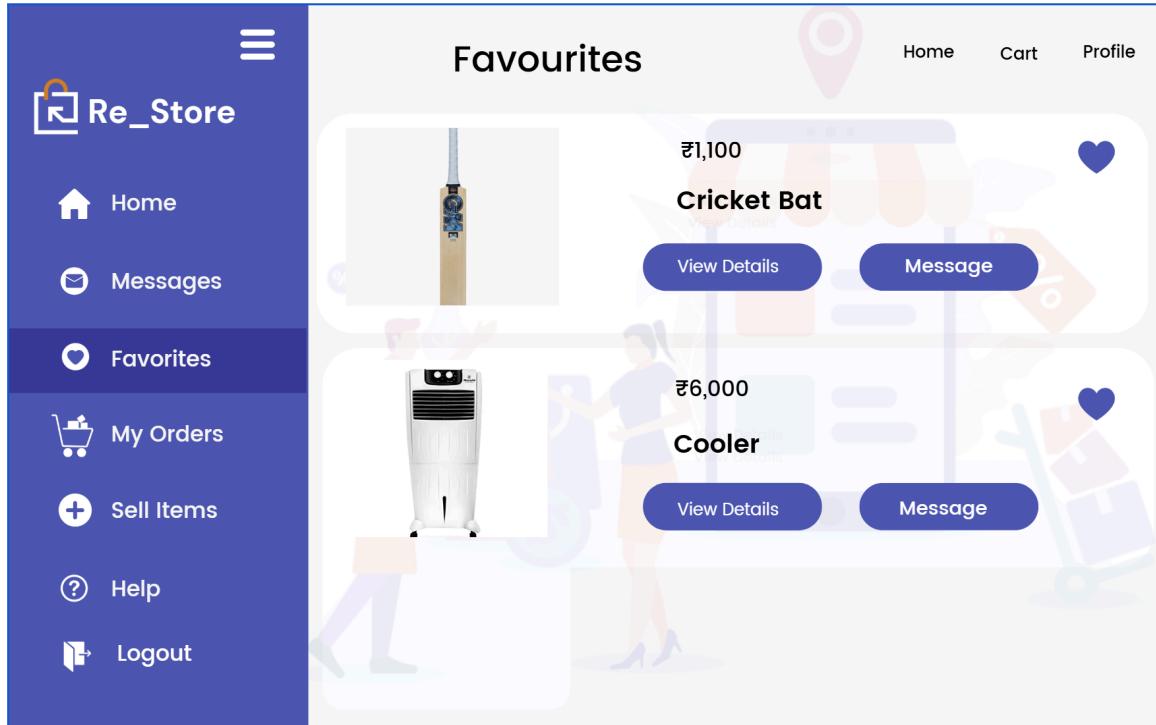
### 1.2.13 Product View Page (Auction)



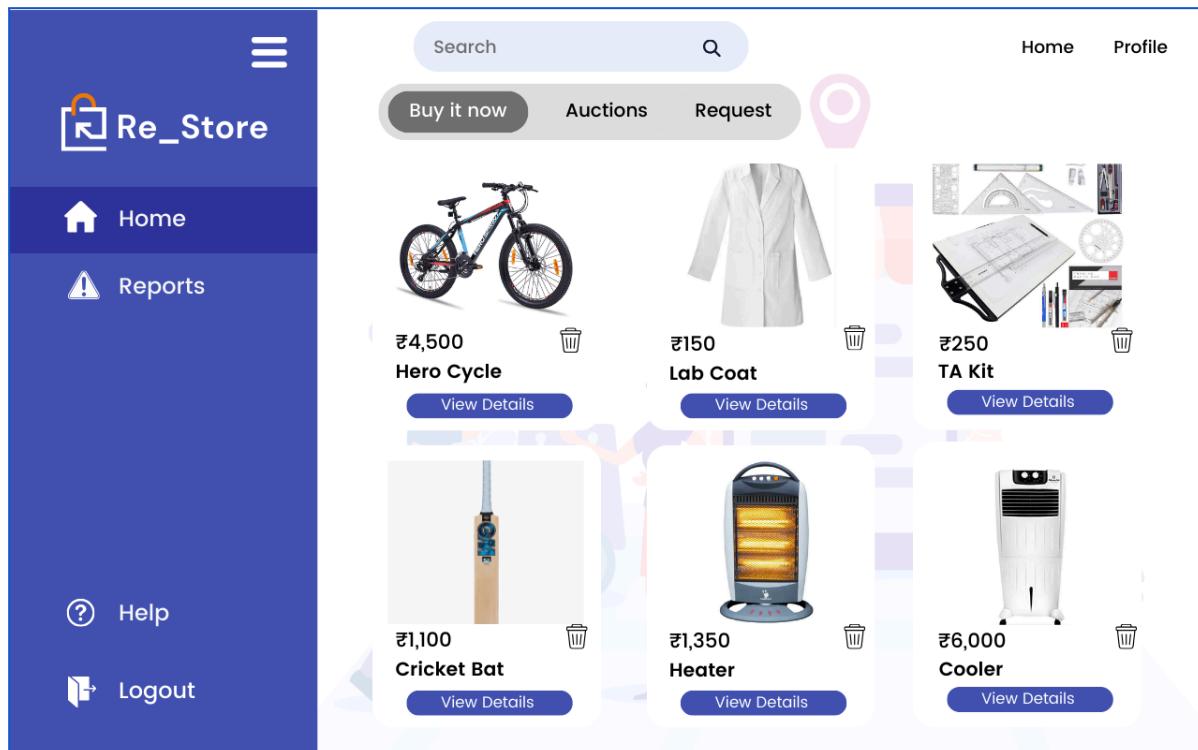
### 1.2.14 List Item Page



## 1.2.15 Favourites Page



## 1.2.16 Admin interface:



## 2 ARCHITECTURE DESIGN

The architectural design of **Re\_Store** can be broken into several key components:

### 1. User Interface (UI)

This component provides the interface for buyers, sellers, and administrators to interact with the system. Built with **React**, the UI includes features like browsing product listings, managing user profiles, participating in auctions, and sending/receiving messages through real-time chat. Administrators have a dashboard to monitor system activity and address disputes or errors.

### 2. Authentication Module

User Authentication: Verifies the identity of users through IITK email-based registration.

Authorization: Restricts access to certain features (e.g., administrative tools) based on user roles.

Two-Step Verification: A feature that adds a layer of security during logins.

### 3. Auction Management Module

This component handles the dynamic, time-bound bidding system.

For Sellers: Allows setting up auctions with a starting price and deadline.

For Buyers: Provides real-time updates on bid status and allows placing competitive bids.

Auction Closing: Automatically determines the winning bidder and notifies both parties once the auction ends.

### 4. Chat Module

Supports real-time messaging between buyers and sellers using WebSocket-based communication. Ensures secure and seamless interaction during transactions.

### 5. Data Storage Module

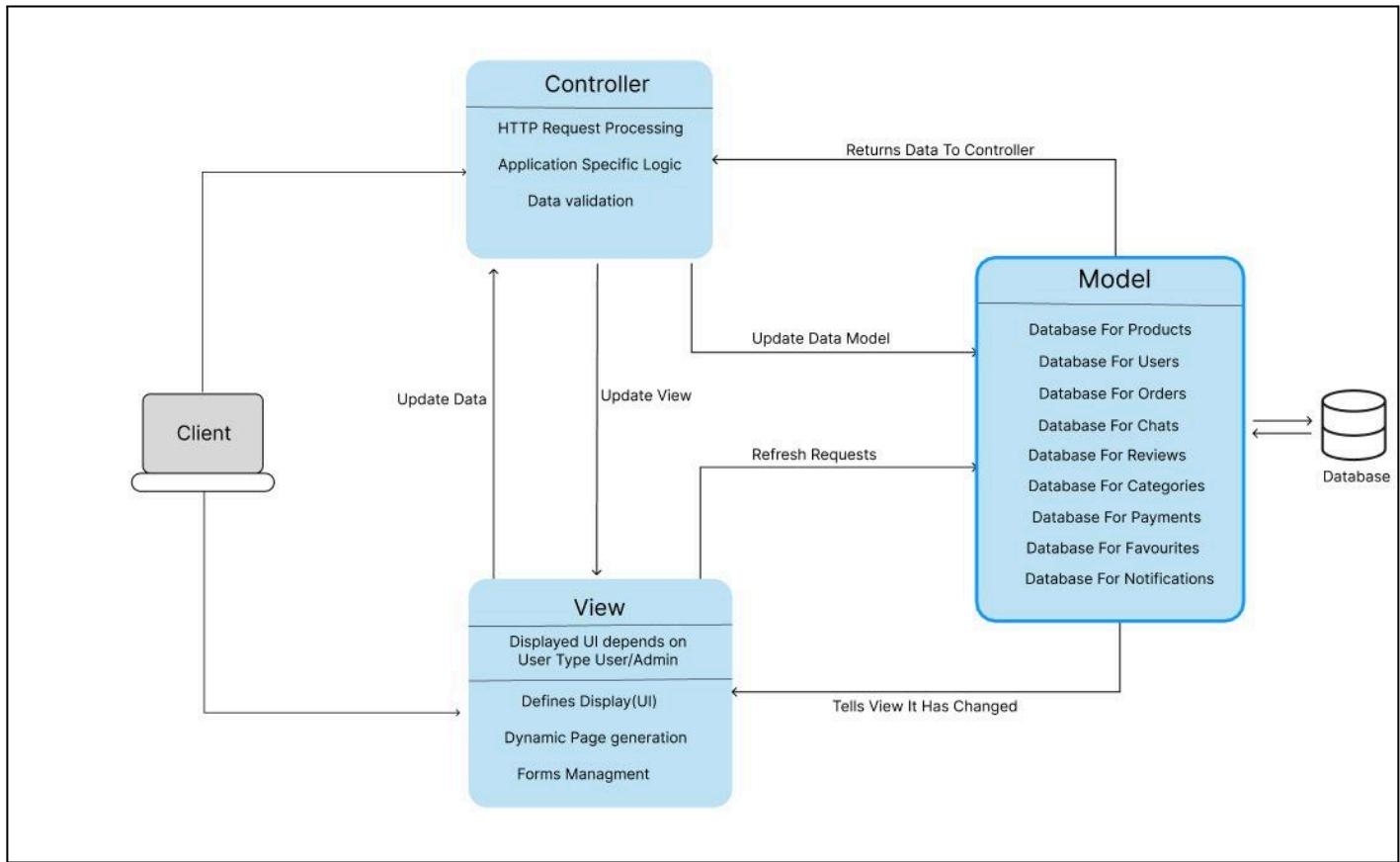
This module stores all user data, product details, transaction records, auction histories, and chat conversations. MongoDB is used for data storage, ensuring efficient management and scalability.

### 6. Reporting and Analytics Module

Provides administrators with data insights, such as user activity, auction performance, and transaction history, for decision-making and auditing.

### 7. Security Module

Ensures encrypted communication using **HTTPS** and safeguards against unauthorized access, data breaches, and auction fraud.



## MODEL-VIEW-CONTROLLER

The architecture follows the **MVC pattern**, which organizes the application into three logical components: **Model**, **View**, and **Controller**. Each component plays a specific role, making the application easier to develop, manage, and maintain.

### Model

Manages and organizes the database layer using **MongoDB**. It stores user profiles, product details, auction records, chat histories, and transactions.

### View

The user interface was developed with **React**. It handles all user-facing functionality, including login/signup, product browsing, bidding, and messaging. The View also interacts dynamically with the Controller for updates in real time.

### Controller

The Controller is built with **Node.js** and **Express.js**. It processes incoming requests from the View and interacts with the Model. It handles core logic such as authentication, CRUD operations for products, managing auction bids, and chat messages between users.

## Data Flow in MVC

**User Action:** A user interacts with the UI to request a product bid on an auction or send a message.

**Controller Logic:** The Controller processes the request, applies business logic, and interacts with the Model.

**Database Updates:** The Model updates or fetches relevant data from the database.

**View Response:** The View dynamically displays the updated state, such as the status of the product request, the new highest bid, or the latest chat message.

## Advantages

- **Modularity:** Simplifies development and debugging.
- **Scalability:** Supports increased users and features.
- **Real-Time Updates:** Enables live bidding and notifications.
- **Secure:** Encrypted communication ensures data safety.

## Disadvantages

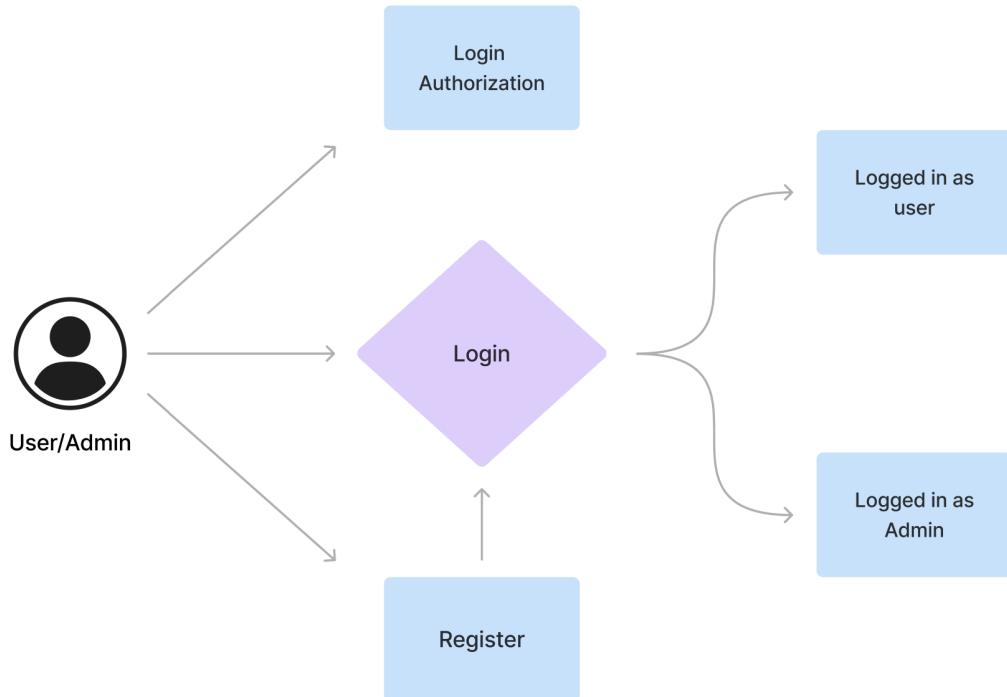
- **Complexity:** Real-time features need more of initial development efforts.
- **Resource Intensive:** Advanced functionality requires higher computation.
- **Increased code complexity:** MVC's separation of concerns may add complexity to straightforward tasks.

## 3 OBJECT-ORIENTED DESIGN

### 3.1 Use Case Diagrams

#### 3.1.1 Use Case #1: Sign Up and Login

**Purpose:** This use case is for the login and signup of the user/admin.



#### 3.1.2 Use Case #2: Forgot Password

**Purpose:** This use case is for creating a new password.



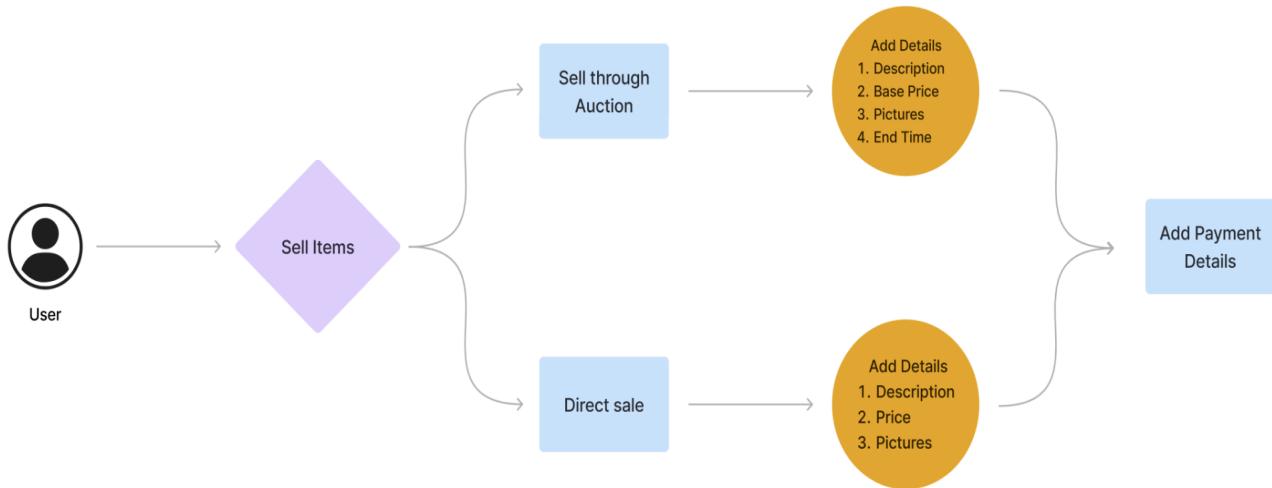
### 3.1.3 Use Case #3:Update Profile

**Purpose:** This use case is for updating the user profile(Name, Profile picture and some other details).



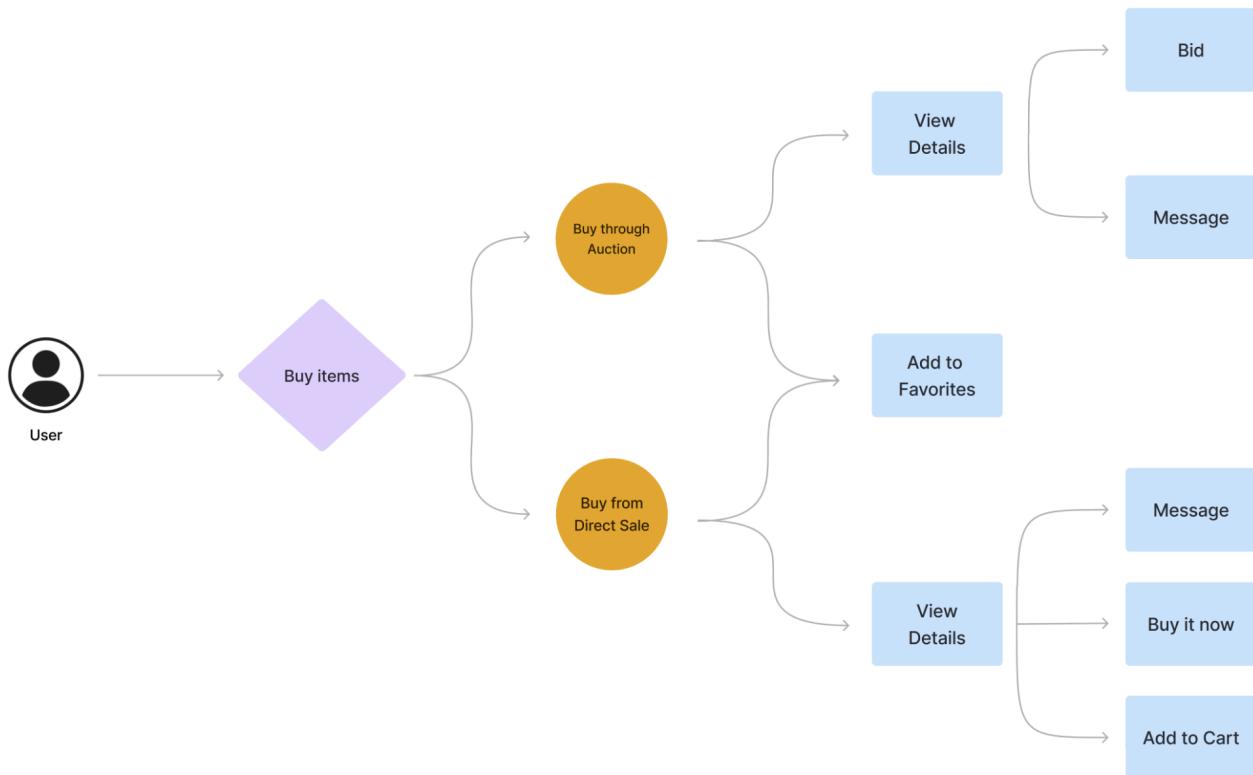
### 3.1.4 Use Case #4:Sell Items

**Purpose:** This use case is initiated when a user wants to sell an item on Re\_store. The user is presented with two options for listing the item: either via an auction format or as a direct sale at a fixed price. In order to proceed, the user must supply all required details, including the asking price, auction duration (if applicable), the condition of the item, and payment information. Once these details are provided, the item is successfully listed for sale on the platform.



### 3.1.5 Use Case #5:Buy items

**Purpose:** This use case is initiated when a user wishes to purchase an item on Re\_store. The platform offers two distinct methods of purchase: if the item is available through an auction, the user can participate by placing bids and, if they have the highest bid at the end of the auction period, complete the purchase; alternatively, if the item is listed for direct sale, the user can immediately buy the product at the listed price. In both scenarios, the user has the ability to message the seller and add the item to their favorites for future reference.



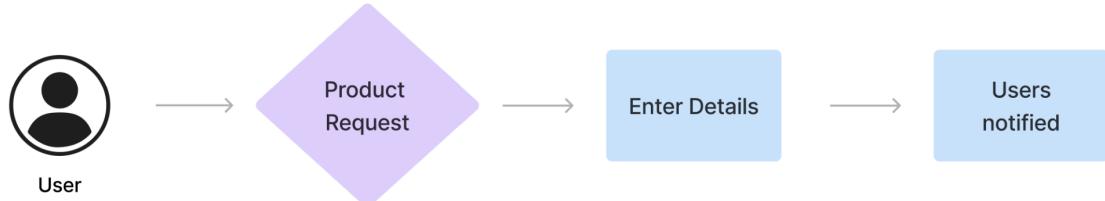
### 3.1.6 Use Case: Logout

**Purpose:** This use case is initiated when the user wants to sign out from the account.

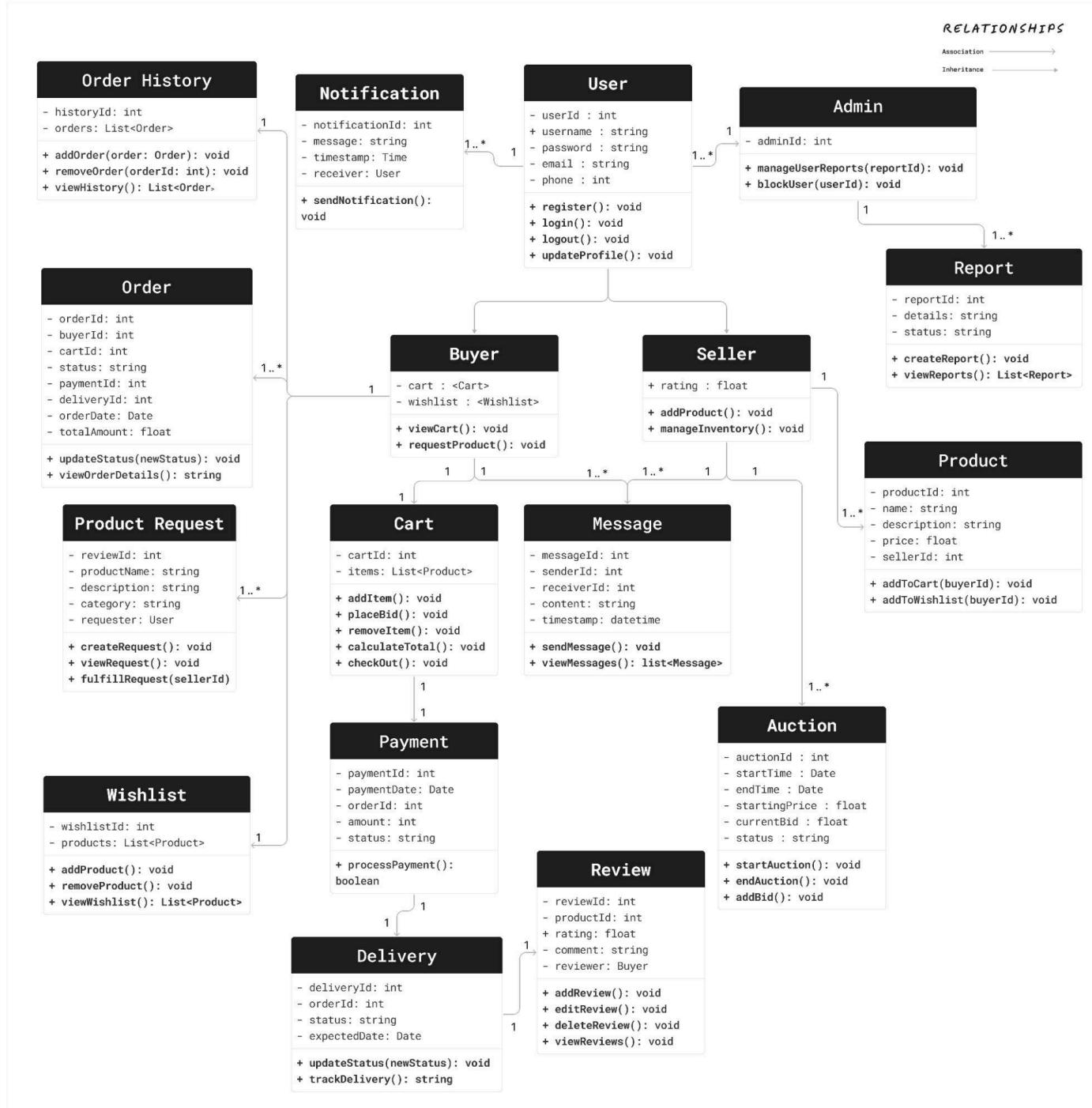


### 3.1.7 Use Case: Product Request

**Purpose:** This use case is initiated when the user wants to request for a product which is not present in the website.

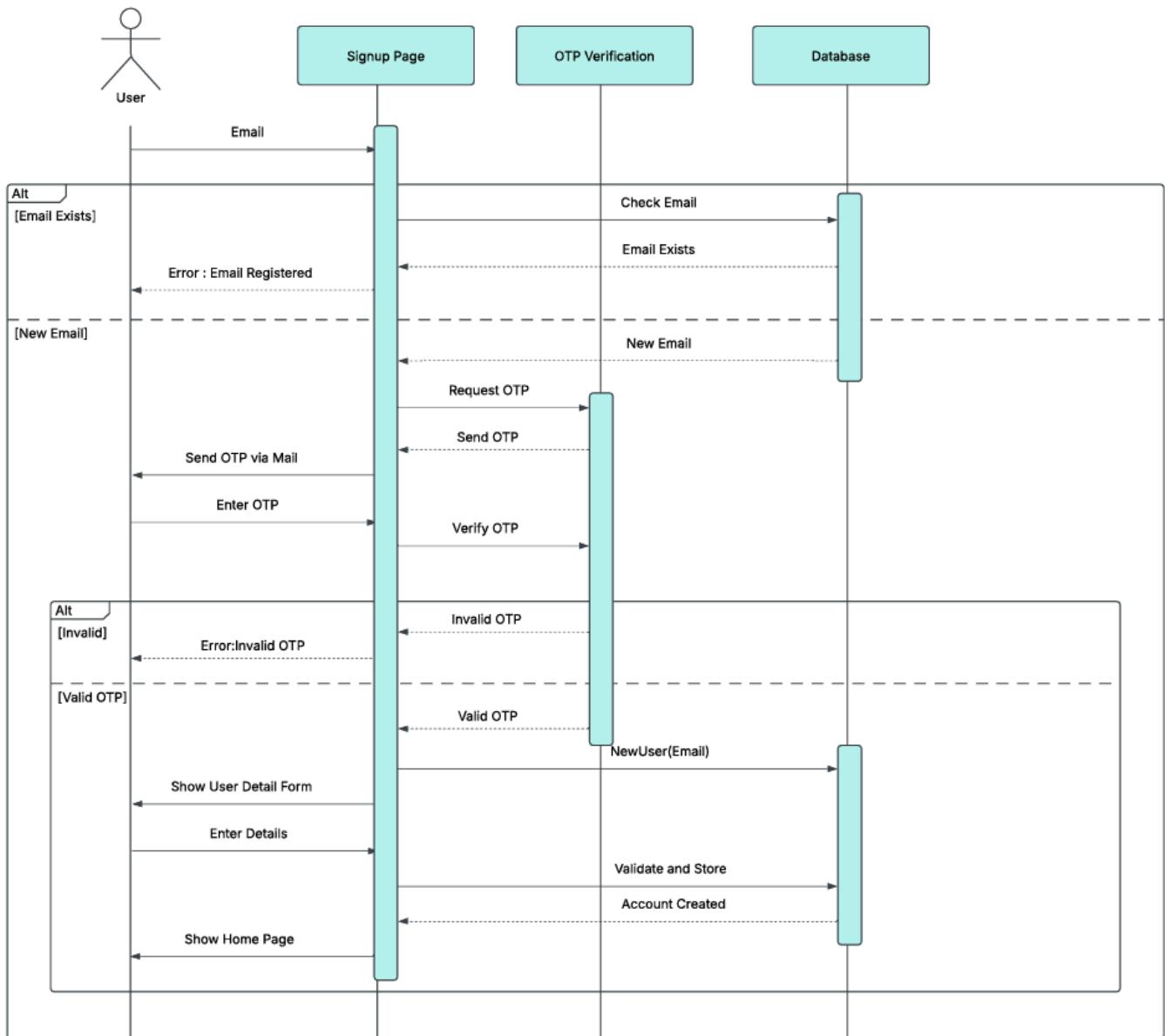


## 3.2 Class Diagram

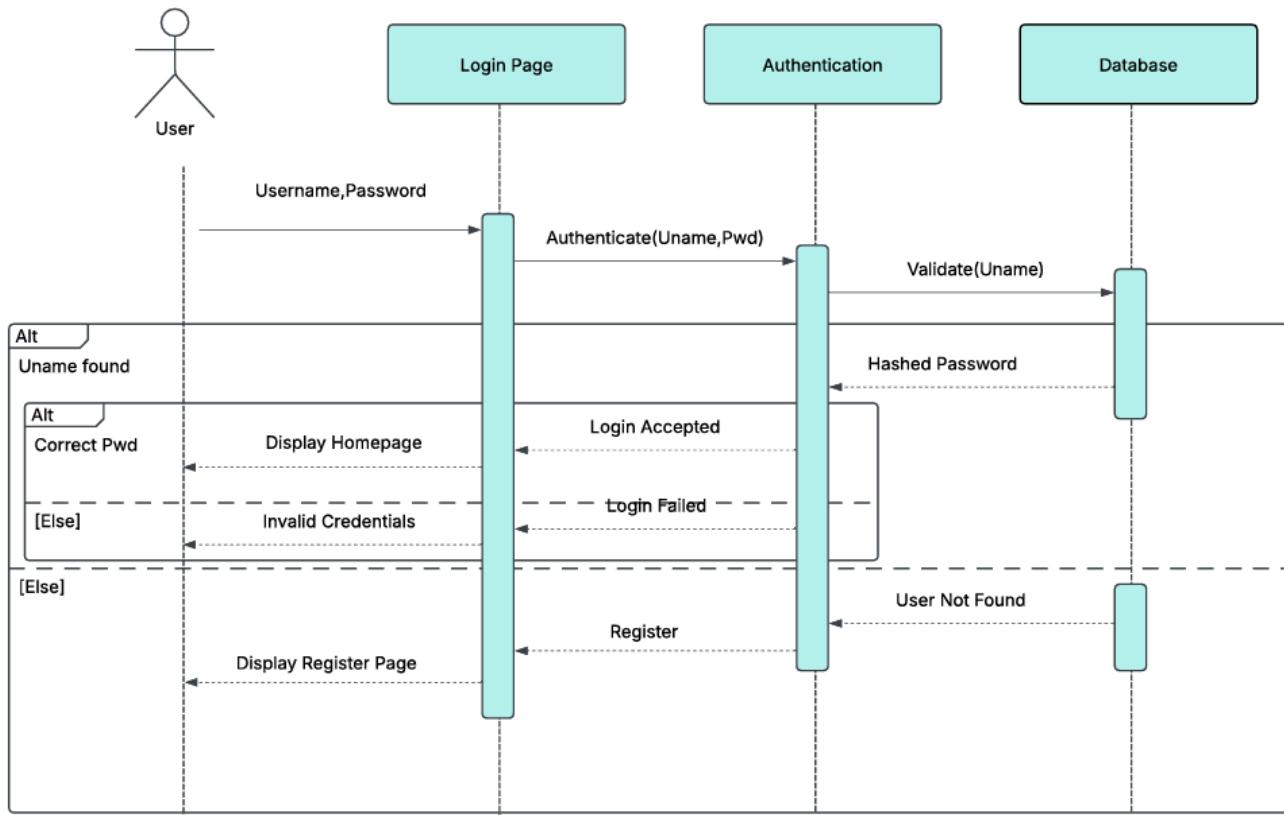


## 3.3 Sequence Diagrams

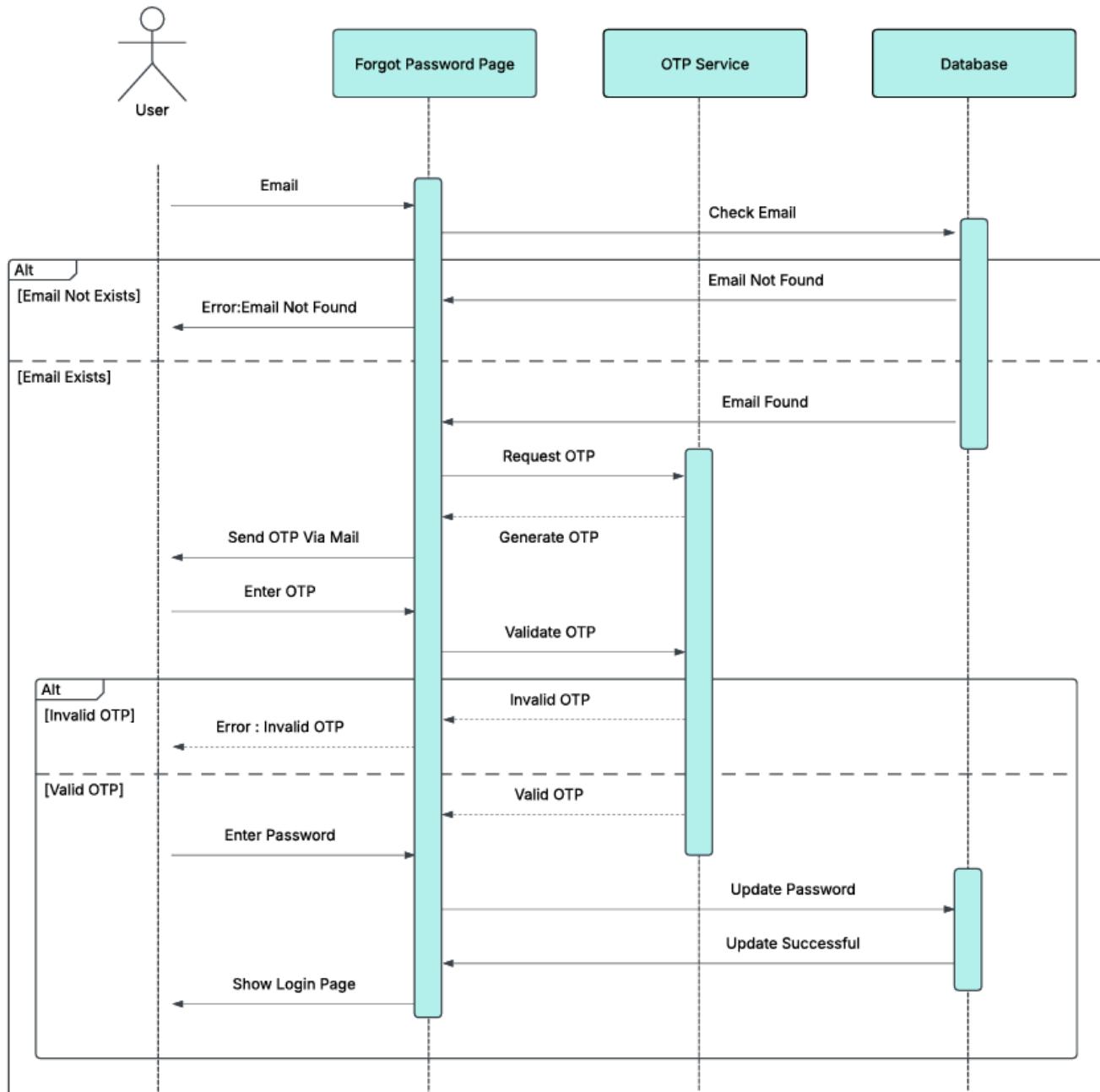
### 3.3.1 Signup



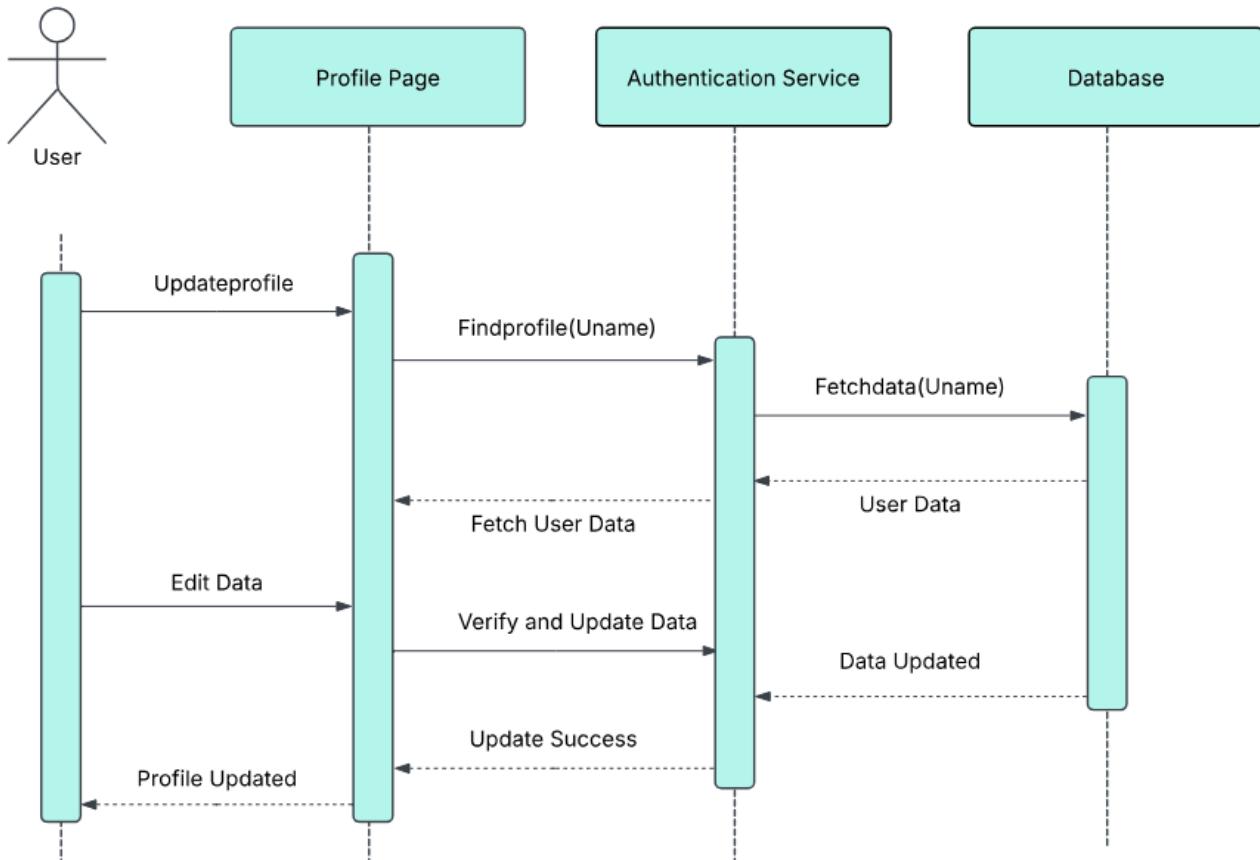
### 3.3.2. Login



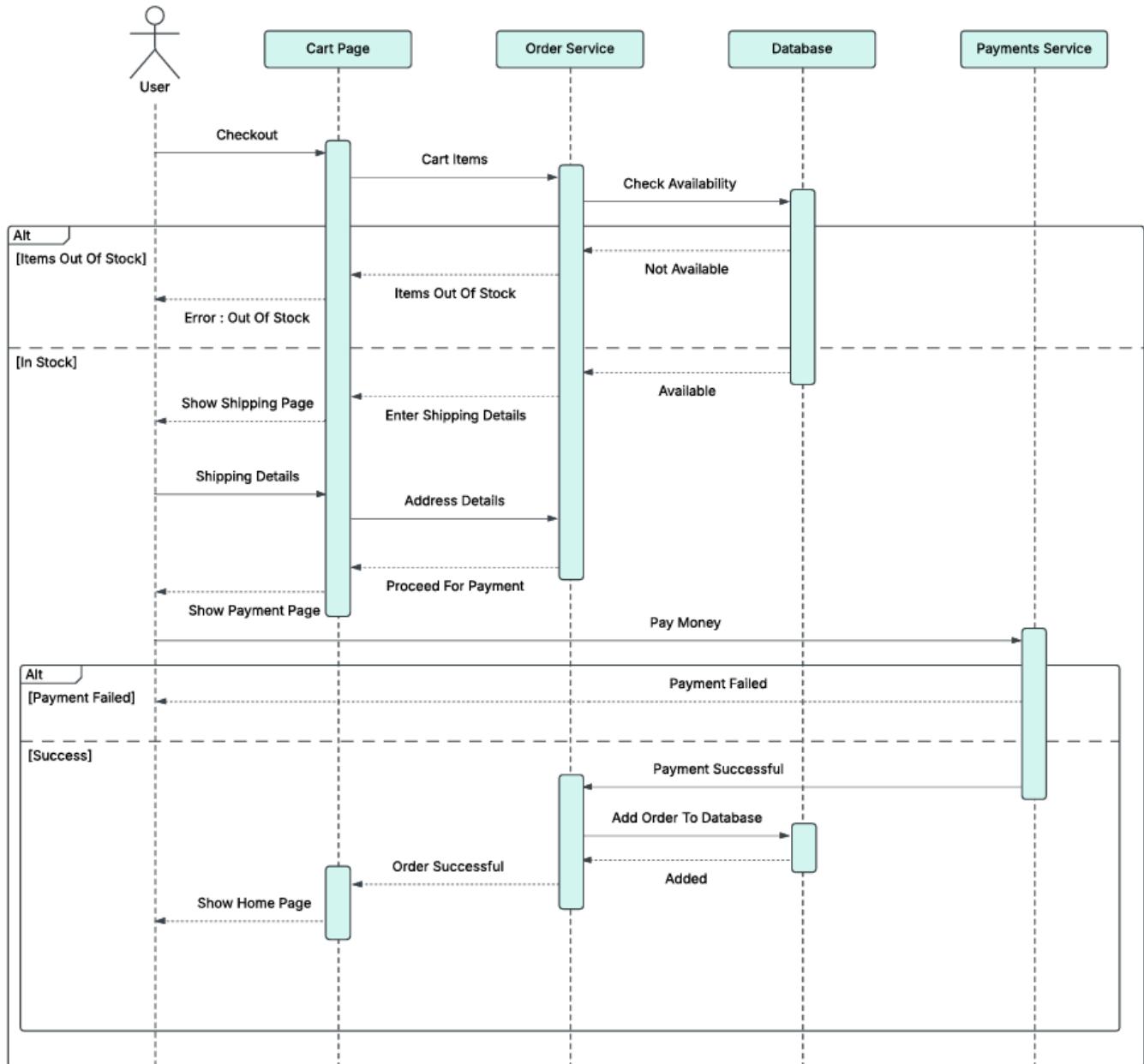
### 3.3.3. Forgot Password



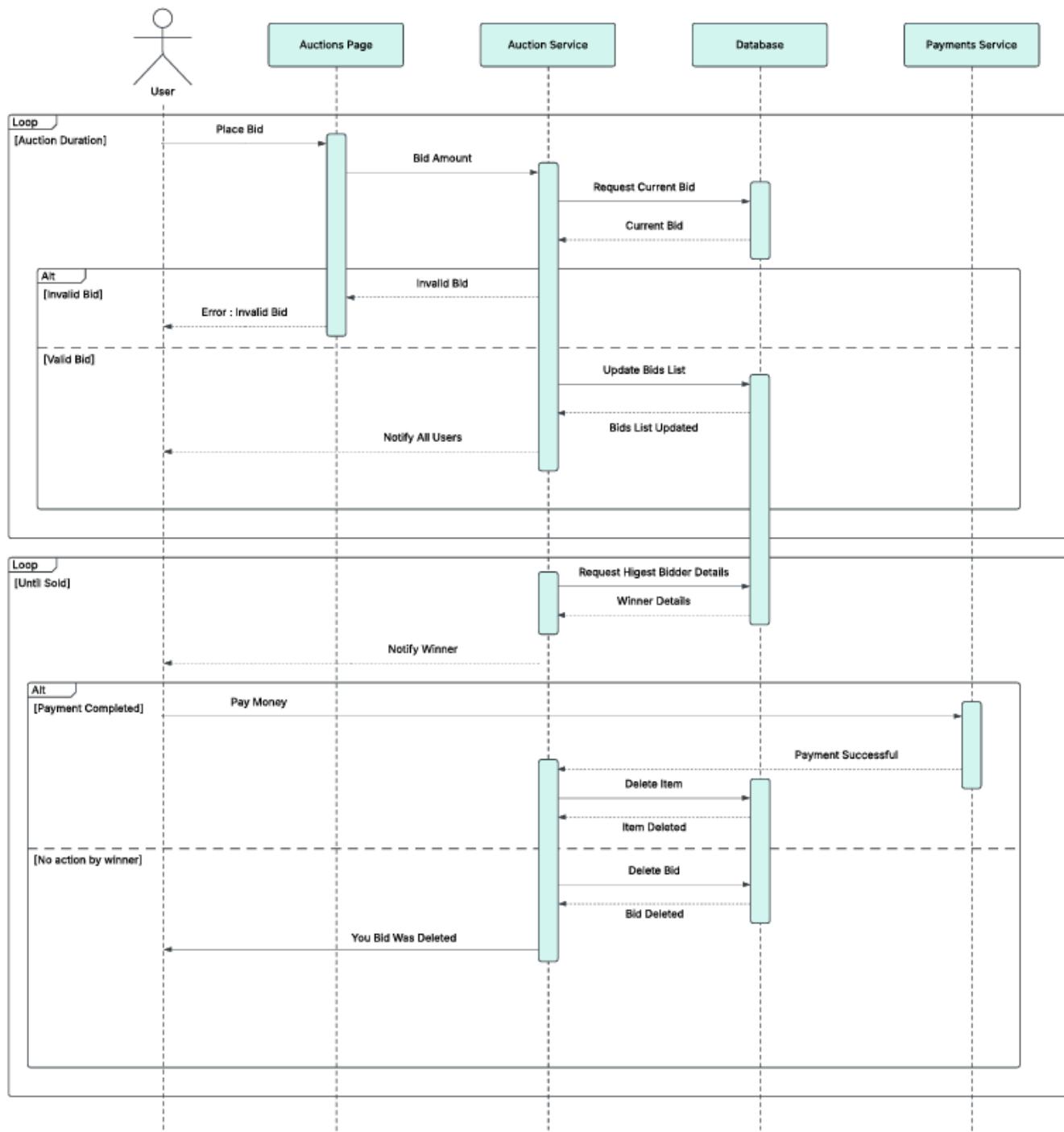
### 3.3.4. Profile



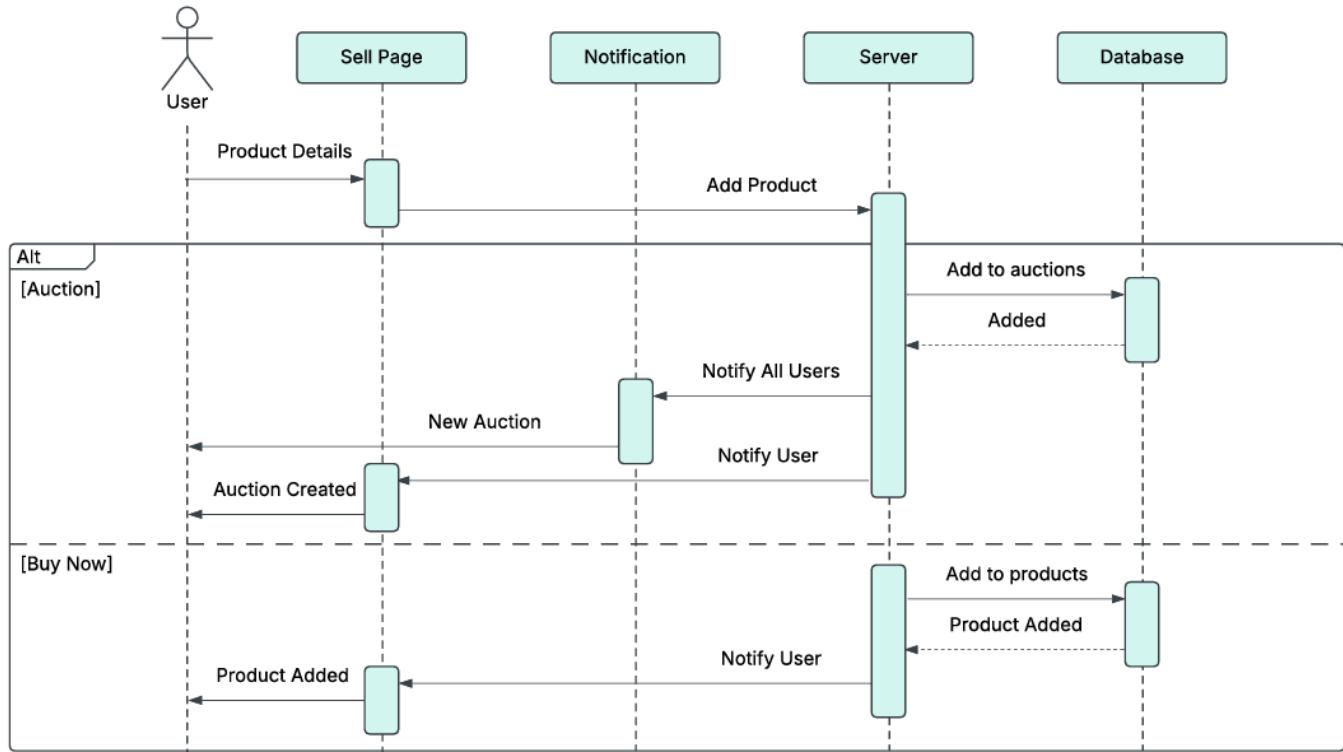
### 3.3.5. Order



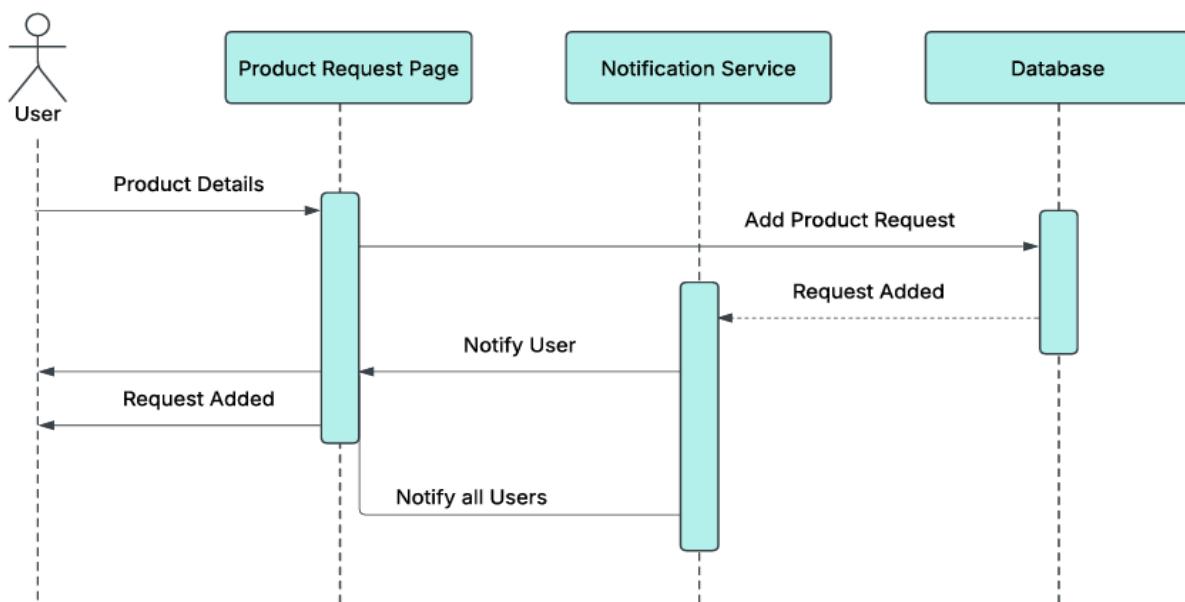
### 3.3.6. Auction



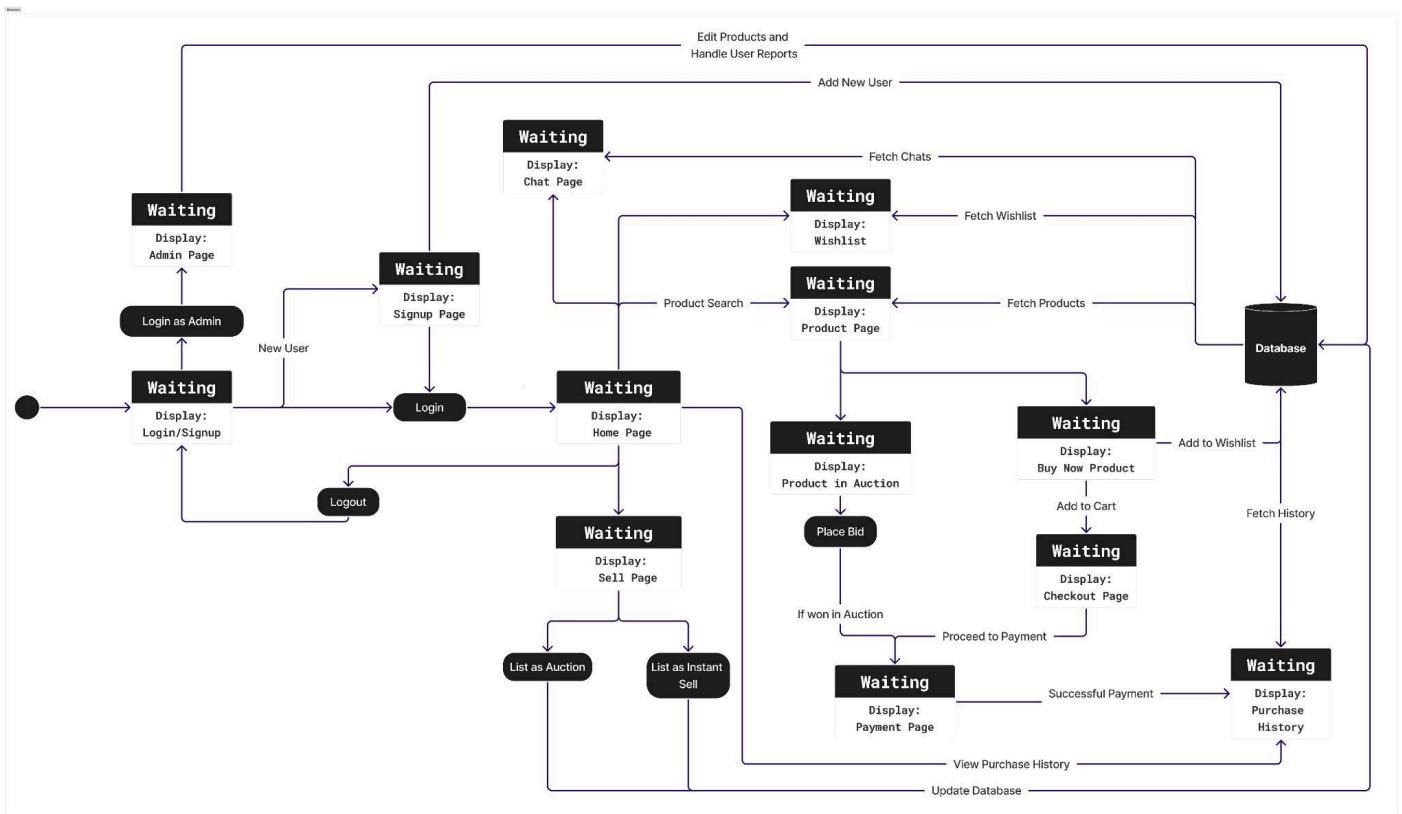
### 3.3.7. Item Listing



### 3.3.8. Product Request



### 3.4 State Diagram



## 4 PROJECT PLAN

### 4.1 Landing Page:

This page serves as the entry point for the website, offering two primary options:

#### A. Register

- ❖ New users (students) input their institute email.
- ❖ Email verification is initiated through an OTP sent to the provided mail ID.
- ❖ After successful verification, users set a password for their account.
- ❖ Direct redirection to the login page upon completion.

#### B. Login

- ❖ Existing users enter their email and password.
- ❖ Pressing the login button verifies credentials; successful login redirects to the homepage.
- ❖ Error message pops up for unsuccessful login attempts, allowing users to retry.
- ❖ A "Forgot Password" option enables users to set a new password via the email verification page used during registration.

### 4.2 Homepage

#### A. Navbar

1. **Chat Page**
  - Displays previous and unseen chats.
  - Allows users to send messages, documents, and images.
2. **Personal Profile**
  - Directs users to their profile.
  - Enables viewing and editing of personal details and profile photos.
  - Users can manage their listed items and auctions.
3. **List Product**
  - Users can list items for sale with descriptions and images.
  - Provides options for "Buy Now" or "Auction."
4. **Product Search & Requests**
  - Users can search for items based on category, name, or seller.
  - Buyers can post product requests specifying what they need.

#### B. Feed

- Displays newly listed items, ongoing auctions, and product requests.
- Users can like and inquire about listed products.
- Notifications for auction updates and bid status changes.

## 4.3 Backend Design

### A. Database Structure

- A single database managing collections for all entities.
- User class store user credentials.
- Product, Auction, and ProductRequest classes store item details.
- Auction and Message classes handle auctions and user communication.

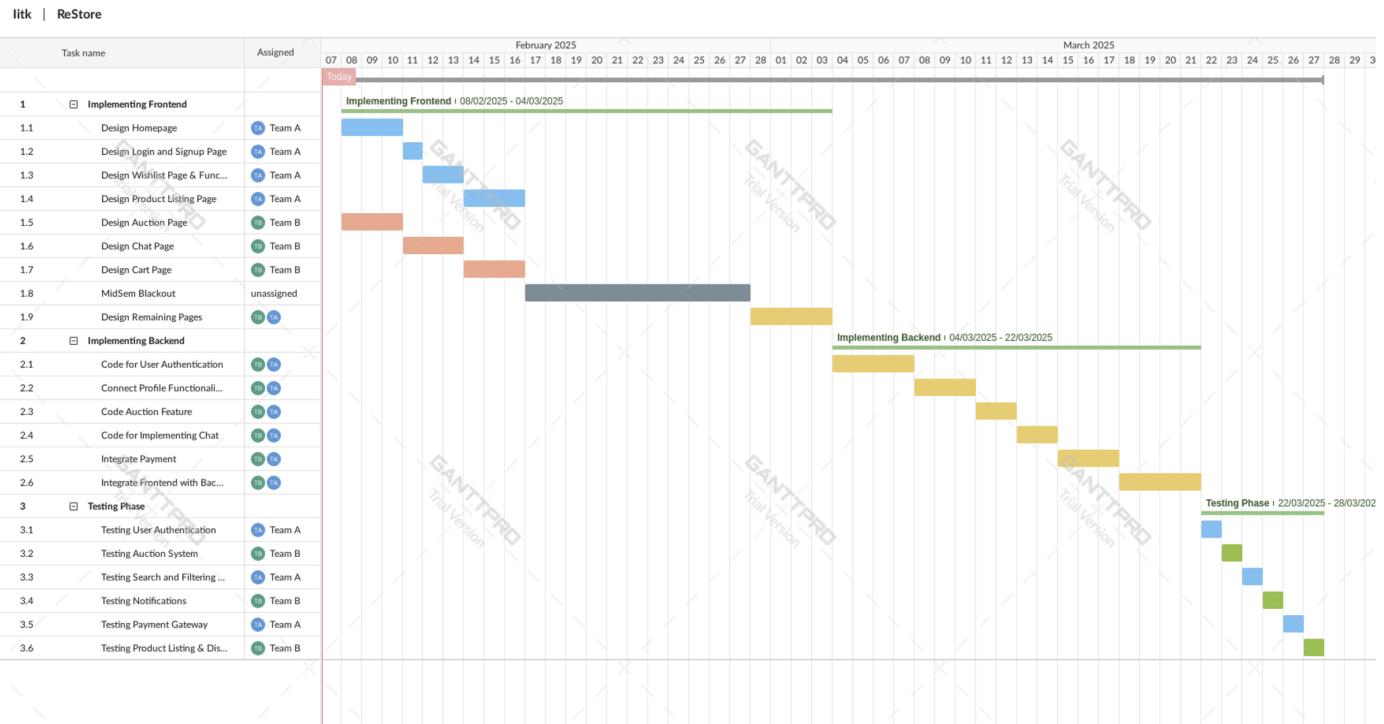
### B. Logic Design

- **registerUser()**: Handles user registration and saves details.
- **verifyLogin()**: Validates login credentials.
- **resetPassword()**: Updates password upon verification.
- **postProduct()**: Enables users to list a product for sale.
- **startAuction()**: Allows sellers to initiate an auction.
- **placeBid()**: Manages bids placed by buyers in real-time.
- **updateBidStatus()**: Updates the current highest bid and notifies participants.
- **postProductRequest()**: Allows users to post item requests.
- **messageUser()**: Enables communication between buyers and sellers.
- **updateProfile()**: Updates user details and listings.
- **renderFeed()**: Fetches and displays relevant products, auctions, and requests.
- **addToWishlist()**: adds a product to wishlist

**Team A:** Saatvik, Prem, Anirudh, Ayush, Meghana.

**Team B:** Koushik, Rakesh, Yashwanth, Sanjay, Pranaya.

## GANTT CHART



---

## **5 OTHER DETAILS**

## Group Log

<b>Meet Date</b>	<b>Members Present</b>	<b>Topics Discussed</b>	<b>Duration</b>
26-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Made an initial draft, and divided the work in a group-wise manner. Prepared timeline to check the work progress.	120 Minutes
27-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Discussed and finalised the architecture design of the application.	90 Minutes
28-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Prepared the context model for our application and also prepared use case diagrams.	90 Minutes
31-01-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Had discussions regarding class and state diagrams and finalised them.	90 Minutes
01-02-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Discussed about various functionalities of the application in detail and prepared sequence diagrams	90 Minutes
02-02-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Discussed various user interface designs and themes and finalised basic user interface.	90 Minutes
03-02-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Designed the first few pages of the application using canva.	90 Minutes

04-02-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Completed the design of remaining pages of the application using canva.	120 Minutes
05-02-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	Had a discussion regarding the future project plan and prepared a Gantt chart.	60 Minutes
06-02-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, PremSanthosh, Sanjay Raghav, Ayush.	Finished the Group Log, made final changes in each section, and completed the document.	150 Minutes
07-02-2025	Koushik, Saatvik, Anirudh, Meghana, Pranaya, Rakesh, Yashwanth, Prem Santhosh, Sanjay Raghav, Ayush.	We had a meeting with the TA to clarify doubts in the document and submitted the document.	30 Minutes