
**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

# ‘MY AUCTION-ONLINE AUCTION SYSTEM’

## A PROJECT II FINAL REPORT

**Submitted to**

**Department of Computer Application**

**Lumbini City College**

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

Submitted by

Gaurav Karki [6-2-1134-11-2021],

Anjita Pathak [6-2-1134-5-2021]

July, 2025

Under the Supervision of

**Suraj Kumar Khattri**


**Tribhuvan University**

**Bachelor of Computer Application (BCA)**

**Faculty of Humanities and Social Science**

**Lumbini City College**

Tilottama, Rupandehi

# SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project(Project-II) prepared under my supervision by Mr. GAURAV KARKI and Ms. ANJITA PATHAK entitled "**MY AUCTION-ONLINE AUCTION SYSTEM**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

……..

Suraj Kumar Khattri

**SUPERVISOR**

Lumbini City College (LCC)

Tilottama, Rupandehi

**Tribhuvan University**

**Bachelor of Computer Application (BCA)**

**Faculty of Humanities and Social Science**

**Lumbini City College**

Tilottama -04, Rupandehi

# LETTER OF APPROVAL

This is to clarify that this project prepared by Mr. GAURAV KARKI and Ms. ANJITA PATHAK entitled "**MY AUCTION-ONLINE AUCTION SYSTEM**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

| | |
|---|---|
| ……<br>Suraj Kumar Khattri<br>Lecturer<br>Lumbini City College<br>Tilottama, Rupandehi | ……<br>Bishal Kandel<br>Coordinator<br>Lumbini City College<br>Tilottama, Rupandehi |
| ……<br>**Internal Examiner** | ……..<br>**External Examiner** |

# ABSTRACT

This project (My Auction-Online Auction System) introduces an Online Auction System developed using the MERN stack comprising React.js for the front end, Node.js and Express.js for the back end integrated with MySQL as the relational database management system. The platform is designed to facilitate real-time bidding, providing a seamless and interactive experience for both buyers and sellers. Users can register, log in securely, list items for auction, and place bids on available products. The system ensures that only the highest valid bid is recorded before the auction deadline, with real-time updates and notifications to keep participants informed. Key features include user authentication, role-based access, bid tracking, countdown timers, and auction history. While the traditional MERN stack typically uses MongoDB, this system uses MySQL to leverage the advantages of structured relational data storage, offering robust support for transactions, data integrity, and complex queries. The system provides an efficient, scalable, and user-friendly solution that modernizes conventional auctions, making them accessible anytime and anywhere via the web. It demonstrates the flexibility of combining modern JavaScript technologies with relational databases to build responsive and reliable applications

**Keywords:** *Online Auction System, MERN stack, real-time bidding, web application, user authentication, bid tracking, auction management, relational database , reliable & responsive*

# ACKNOWLEDGEMENT

The development of the '**My Auction-Online Auction System'** has been a rewarding experience, made possible through the support and guidance of several individuals and groups.

We would like to sincerely thank our project mentor, Mr. **Suraj Kumar Khattri** (Project Supervisor), for his valuable advice, timely feedback, and continuous encouragement throughout this project. His technical expertise and guidance were crucial in shaping the core functionalities of the system.

We are also thankful to Mr. **Bishal Kandel** (Coordinator), **Mr. Sujan Dhakal** (Coordinator), **Mr. Ashok Gurung**, Mr. **Bhuban Panthee**, Mr. **Bishnu Prasad Chaudhary**, Mr. **Kamal Thapa** and Mr. **Dinesh Neupane** of Lumbini City College for creating an environment that fosters learning and innovation. Their support laid the groundwork for this project's success.

Gratitude is also due to my peers and collaborators who offered their time for discussions, reviews, and testing, helping ensure the system's usability and performance.

Lastly, we acknowledge the indirect support from our friends and family, whose patience and motivation were essential throughout this process.

This project allowed us to explore real world software development and user interaction challenges, and we are grateful for the experience.

Gaurav Karki
Anjita Pathak
Lumbini City College

# TABLE OF CONTENTS

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| API | : | Application Programming Interface |
| CDN | : | Content Delivery Network |
| DBMS | : | Database Management System |
| DFD | : | Data Flow Diagram |
| ER | : | Entity Relation |
| HTTP | : | Hyper Text Transfer Protocol |
| ID | : | Identification |
| I/O | : | Input/Output |
| JS | : | JavaScript |
| MacOS | : | Macintosh Operating System |
| MERN | : | Mango DB Express JS React JS Node JS |
| MySQL | : | My Structured Query Language |
| OTP | : | One Time Password |
| ROI | : | Return Of Interest |
| SDLC | : | System Development Life Cycle |
| UI | : | User Interface |
| VS | : | Visual Studio |
| XAMPP | : | X-operating system, Apache, MySQL, PHP, Perl |

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1 Introduction

## 1.1 Introduction to My Auction-Online Auction System

An auction system refers to the buying and selling of products with high values and uniquely acceptable in the selling markets. One can provide products details and wait for the response form the auction committee to accepts such item in order to place it in the website. Any product is uniquely identified by the committee and placed in the list of items to be sold. If it is not taken within the given period of time, the items are replaced or removed from the list for the temporary period of time or until there is desire of such archived items.

## 1.2 Problem Statement

It is legally accepted to bid the amount of money in auction unless there is violation of rules. One must be there to bid during the auction in the real time which makes it difficult for any normal circumstances. It has to be done in a particular place where only native users can participate. No options for multiple variety of items to choose making it hard to buy and sells product from the different parts of country. Needs manpower and over cost estimation for the venue and staffing can be problems in mostly auction house. So, in order to get rid of such problems, online bidding system where one can use their free time to check the post related to listed items and bid as per the available amounts. It can be done anywhere, everywhere and anytime within the bidding dates fixed by the committee.

## 1.3 Objectives

Let us discuss some of the main objectives and goals of this project:

- To provide a platform that allows buyers and sellers from diverse locations to participate in auctions.
- To reduce overhead costs associated with physical auctions, including venue rental and staffing.
- To enable sellers to reach large number of buyers to access a wider variety of products.

## 1.4 Scope

- **Wide Range of Products:** Online auction platforms can cater to a broad spectrum of products, from electronics, communication, transportation and antiques to real estate and services.

- **Wider Audience:** Buyers and sellers from all regions of the country can participate, increasing the potential market size.
- **24/7 Availability:** Auctions can be conducted at any time, providing flexibility and convenience for participants in different time zones.
- **Real-time Bidding:** Participants can place bids in real-time, ensuring a dynamic and engaging auction experience.

## 1.5 Limitations

- **No Card and International Payment System:** System do not contain any international or card payment method rather than local service such as eSewa, Khalti etc.
- **No Mobile App Service:** User may not be able to use the mobile application for this service which might affect in user's experience.

## 1.6 Development Methodology

The project is built with the help of Agile method of SDLC. The project has been divided into multiple sprints and all those works has been performed parallelly concerning with group members and supervisors.


**Figure 1.6: Agile Software Development Model** [1]

### 1.6.1 Project Timeline

This project using Agile development method is sub-divided into multiple phase and sprints as mentioned below: Process and Date of Execution are the two factors used to describe the timeline.

Date of Execution →

| Process ↓ | 13 Feb | 23 Feb | 12 March | 28 March | 14 June |
|---|---|---|---|---|---|
| Requirement Analysis | | | | | |
| Planning of Sprints | | | | | |
| System Design | | | | | |
| Implementation | | | | | |
| Review and monitor | | | | | |
| Documentation | | | | | |

**Figure 1.6.1: Gantt Chart**

## 1.7 Report Organization

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

# Chapter 2: Background Study and Literature Review

## 2.1 Background Study

An online auction is a type of auction conducted over the internet, allowing bidders the flexibility to participate from any location with internet connectivity. The primary advantage of online auctions is this freedom and convenience, compared to traditional in-person auctions [1].

The initial phase in developing an online auction platform involves creating a reliable system capable of managing core auction activities such as handling multiple bids, adding or removing product details, displaying auction items, and synchronizing data in real time [1]. Bidders must register on the platform to participate, which involves technical considerations like securely storing bidder information, managing access policies, and implementing authorization and verification mechanisms to ensure each bid is uniquely identified and recorded.

The bidding process typically starts with the auctioneer announcing the starting price for each item and inviting bids. Many online platforms support live streaming of auctions, enabling real-time bidding interactions. Some systems incorporate automation features, such as virtual clerks, which place bids on behalf of bidders who are physically present or unable to bid manually. Each bid placed online is recorded in the system, allowing updates to the user interface in real time for all participants to see.

Payment processes in online auctions employ secure payment gateways combined with encryption and fraud detection technologies to ensure safe transactions between buyers and sellers. Once an auction concludes successfully, the platform issues a certificate of sale to the winning bidder, serving as legal proof of purchase, including details like auction date, item description, price, and buyer information [1].

## 2.2 Literature Review

Online auction systems, also known as e-auctions, enable users to bid on products or services via the internet. The English auction format, which features progressively increasing bids until the highest bidder wins, is among the most common types. Pioneering platforms such as eBay and OnSale support millions of concurrent users and have streamlined auction management processes significantly [5], [9].

Despite their popularity, online auctions face challenges related to security, trust, and transparency. Issues include bid repudiation, delivery fraud, and identity misuse by dishonest parties such as buyers, sellers, or auctioneers [5]. To mitigate these risks, platforms deploy authentication methods (passwords, two-factor authentication), fraud monitoring, and anonymous proxy bidding, though vulnerabilities like shill bidding, bid shielding, and seller identity obfuscation still persist [5], [6], [10], [11].

Trust mechanisms such as bidirectional feedback and reputation systems allow buyers and sellers to rate each other, fostering transparency and influencing participant behavior. These mechanisms have been shown to increase final bid values; however, they remain susceptible to manipulation through collusion, fake reviews, and feedback inflation [2], [8], [11].

To address security concerns, researchers have proposed cryptographic auction protocols. For example, anonymous English auction schemes leveraging group or ring signatures ensure properties like unforgeability, non-repudiation, public verifiability, and bidder anonymity [9], [10], [11]. Trevathan et al. introduced a secure English auction design distributing trust across servers to prevent issues such as early closing and bid-blocking [4]. Blockchain-based approaches use smart contracts to penalize fraudulent activities like shill bidding in real time, employing metrics such as a Bid-Shill Score [5], [7].

Hybrid trust-management frameworks combining policy-based authentication with reputation systems employed in marketplaces like eBay have demonstrated improved user confidence experimentally [6], [7].

System reliability is also enhanced by implementing data integrity measures and disaster recovery strategies, including redundant backups and geographically replicated servers, as adopted by large platforms such as eBay and Christie's [6].

Lastly, empirical studies analyzing auction efficiency with real-world datasets have provided insights into bidder behaviors, pricing strategies, and market dynamics, aiding in optimizing revenue and platform design [3].

# Chapter 3: System Analysis and Design

## 3.1 System Analysis

### 3.1.1 Requirement Analysis

Analysis for requirements is done by categorizing into two parts i.e. functional and non-functional requirement. Let us discuss some requirements for the development of this project:

- **Functional Requirement**

  **Search Functionality:** The application should have a robust search functionality that can retrieve products quickly based on data provided through the input.

  **Create Functionality:** The application should provide create section for admin to upload details of movies, shows and series that are recently available for the online platforms.

  **User Authentication:** The application should require users to log in before they can create post, and should have a secure authentication mechanism.

  **Payment Gateway:** The application should integrate with a payment gateway to enable users to pay the amount of their subscriptions online using different payment methods.

  **Customer Support:** The application should provide users with a way to contact customer support, whether through email or phone.

  **Administration:** The application should have admin panel to verify and allow access to the user profile, edit and update the details, view the details of actions performed by user in the site as well as the post done by the admin itself.

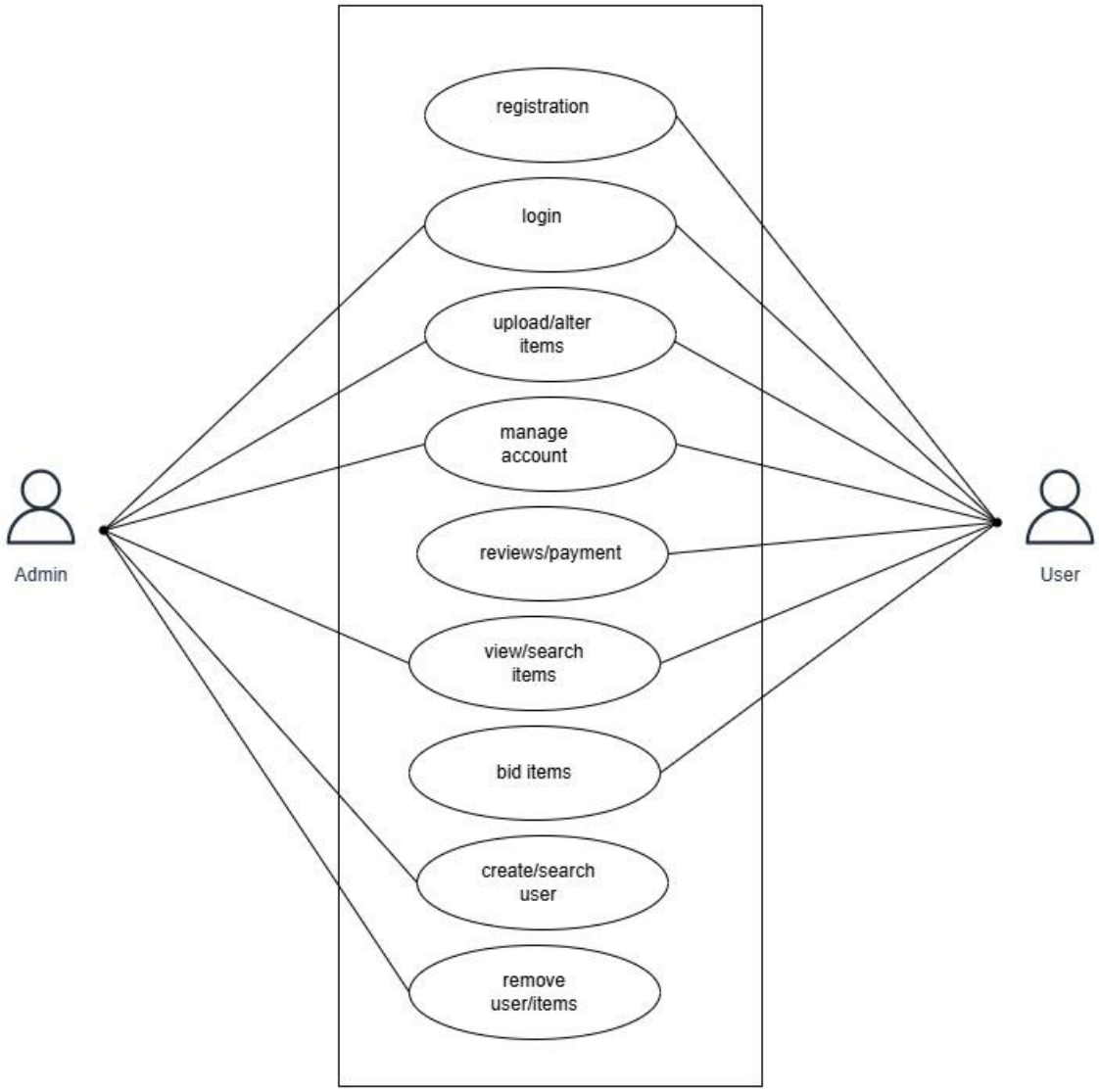



**Figure 3.1.1: Use-case Diagram**

- **Non-functional Requirements**

  **Availability:** The application should be available for 24/7 for all the users. They should be able to perform their activities anytime they feel good. Application must be able to serve whenever and wherever a user wants.

  **Security:** The security is the biggest concern for the user so the application should be made more secure with major authentication and executable codes in server-side scripting.

  **Performance:** The system should have good user interface with smooth performance. All the basic functions and operations should be able to run smoothly and there must not be any chance of network lagging or access error.

**Reliability:** The application should be developed to perform its intended functions without failures or disruptions consistently as user can easily relay on this system.

**Error Handling:** The system error must be hidden from the interface and is handle accordingly if appears any at the server-side so user will not have to deal with such unexpected situations.

### 3.1.2 Feasibility Study

The following feasibility factors are studied for the development of this project. Let us discuss some of them below:

- Technical feasibility:

  MERN is used for the development of the project that supports the required features like user authentication, online payment and real time data updates. There is the implementation of strong security measures to protect user data and prevent unauthorized access with system design to handle growth in users and content without performance degradation. So, the project is technically feasible for the development.

- Economic feasibility:

  Economic feasibility evaluates the financial viability of the project. It is cost free development where there is no any cost for development, infrastructure, marketing, and upcoming maintenance. It is thoroughly studied about the target audience, market demand, and competition to ensure there is a viable market for the service. After calculating the expected ROI(rate of interest),so it is determined that the project is economically feasible.

- Operational feasibility:

  Operational feasibility examines how well the project will work in practice. It is ensured that this webapp a user-friendly and provides a seamless experience for browsing and streaming content. It includes multiple processes for acquiring, managing, and updating content to keep the library fresh and engaging. It is provided with customer support, troubleshooting, and regular maintenance to address any issues that arise. The project ensures to complies with legal and regulatory requirements, including licensing agreements and copyright laws that makes it operationally feasible.

## 3.2 System Design

The system is designed with multiple features and functionality allowing user to have better user experience while accessing the website. Some graphical representation of this system using Context level DFD and System Flowchart are mentioned below:

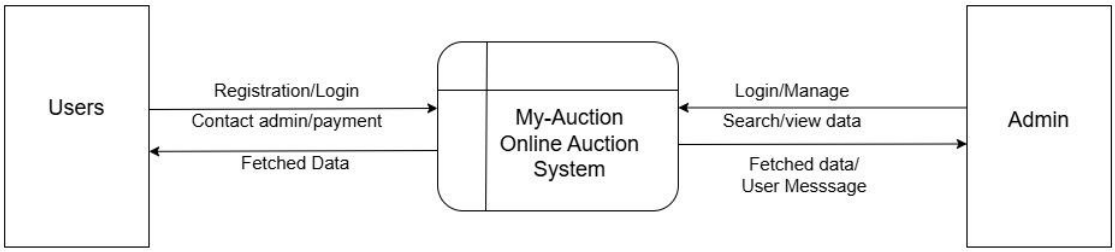



**Figure 3.2.1: Context Level DFD**

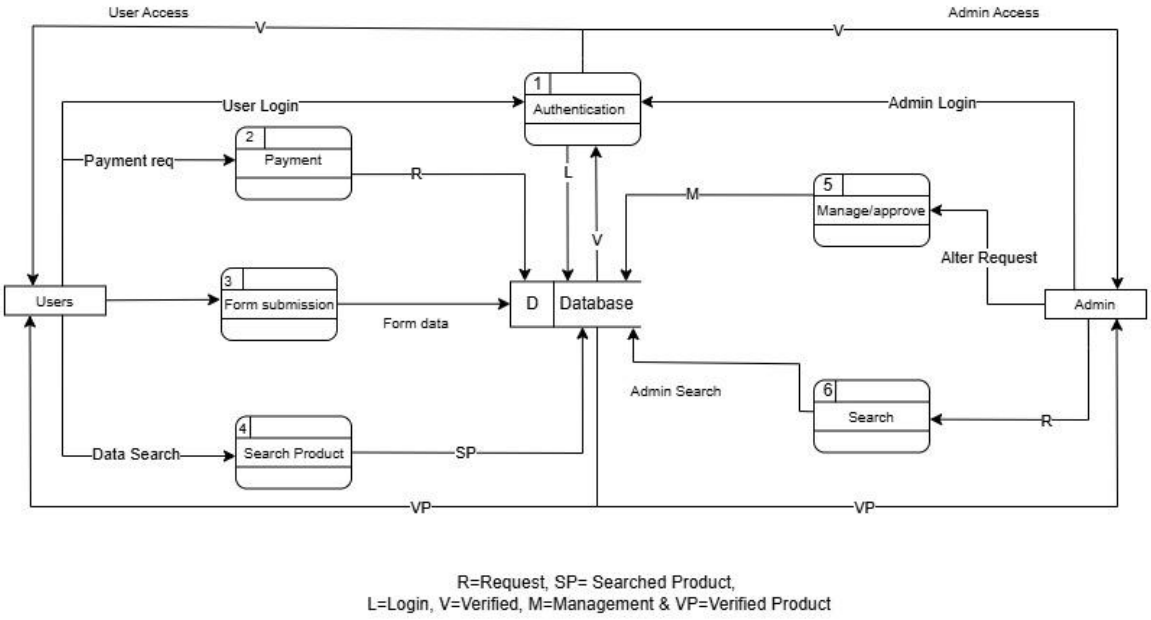
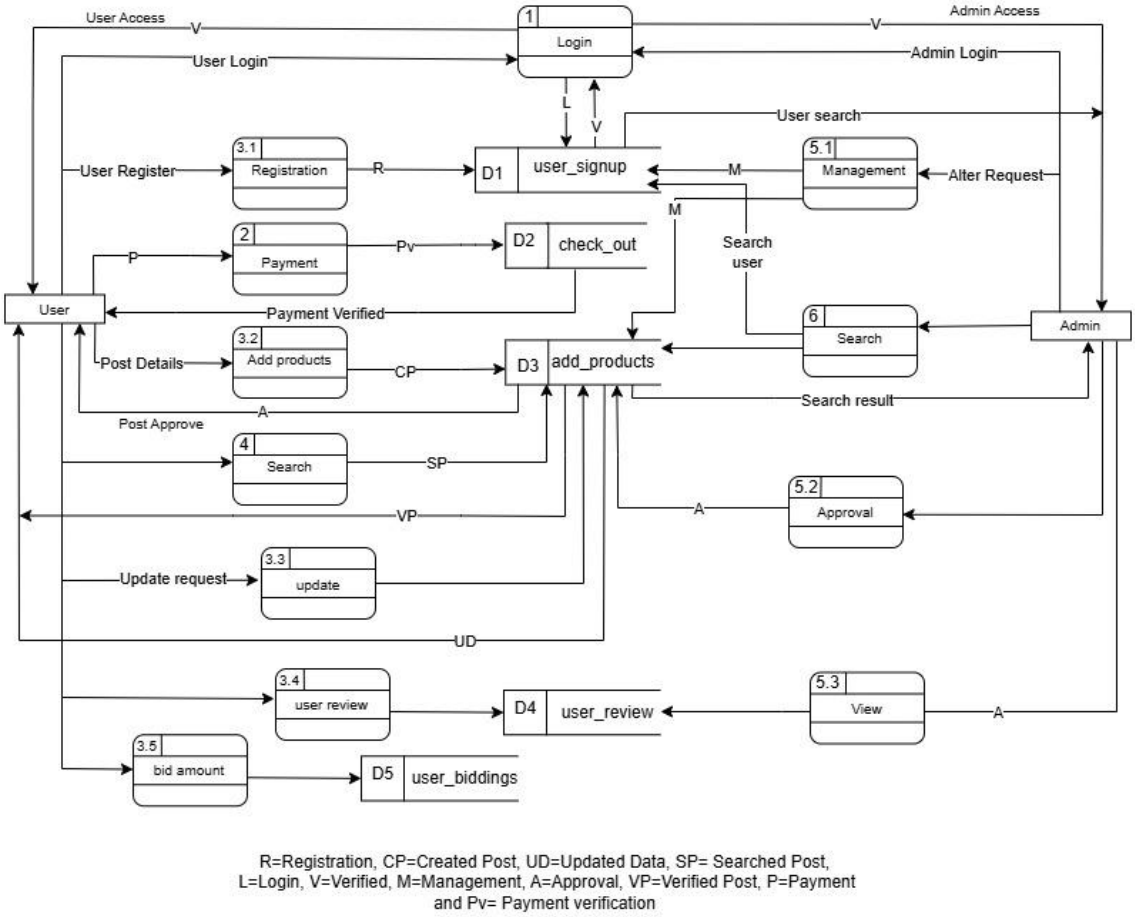



**Figure 3.2.2: Level One DFD**




R=Registration, CP=Created Post, UD=Updated Data, SP= Searched Post,
L=Login, V=Verified, M=Management, A=Approval, VP=Verified Post, P=Payment
and Pv= Payment verification

**Figure 3.2.3: Level Two DFD**

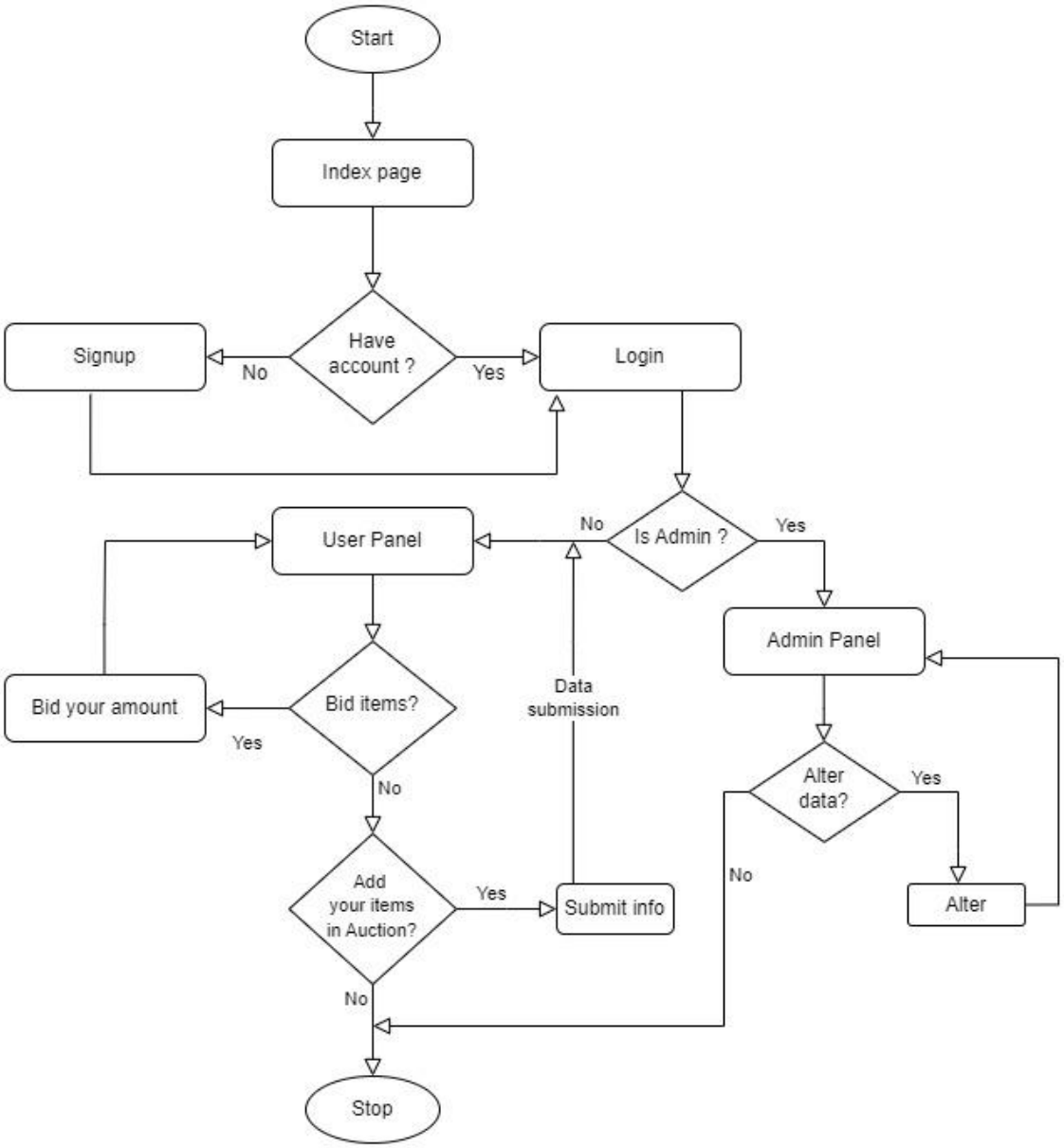



**Figure 3.2.4: System Flowchart for Auction System**

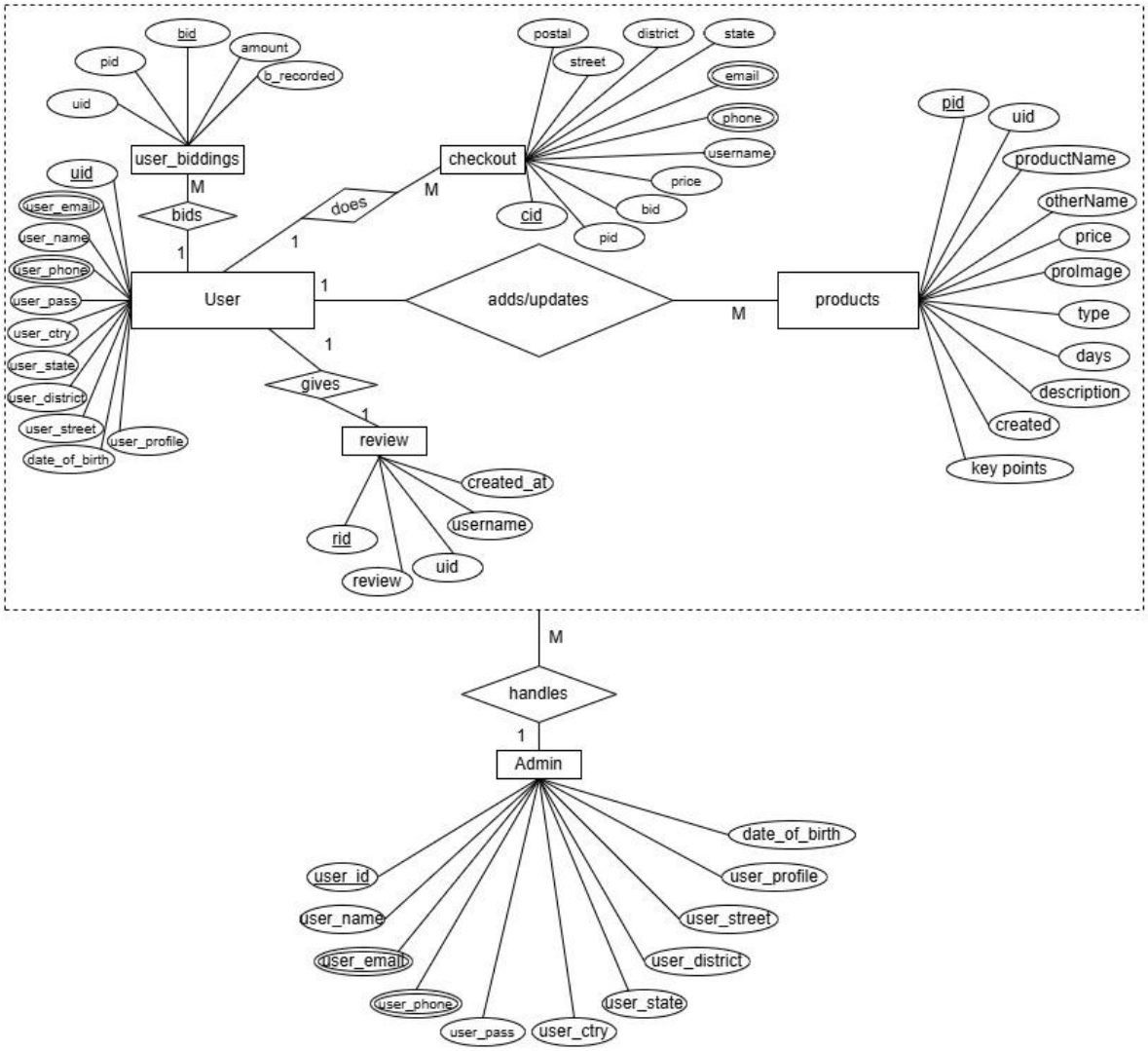



**Figure 3.2.5: ER-Diagram**

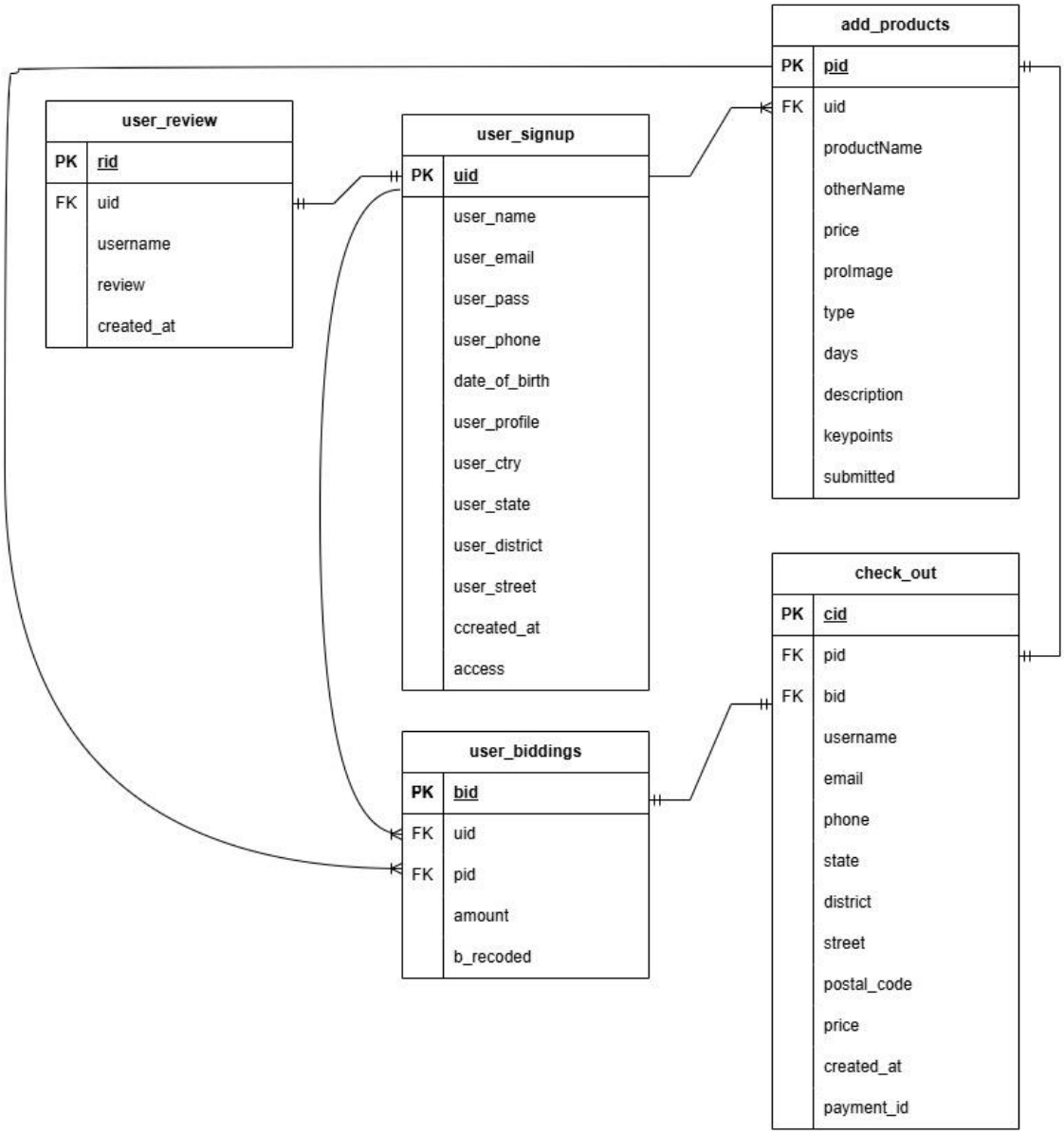



**Figure 3.2.6: Schema Diagram of Auction System**

## 3.3 Algorithms

### 3.3.1 Search Optimization

This algorithm helps user to get the most relevant items, either by popularity, pricing, number of visits etc. related to the product they search for, filtering them by their ‘type’. It helps in performance and gives the best outcome to the user.

**Frontend:**

ON COMPONENT MOUNT:

1. Initialize state:

   - query = ''

   - results = []

   - loading = false

2. DEFINE fetchResults(text):

   IF text is empty THEN

      SET results = []

      RETURN

   SET loading = true

   TRY:

      SEND GET request to /api/search?query=text

      ON success:

         SET results = response.data

   CATCH error:

      LOG "Search error"

   SET loading = false

3. DEFINE debouncedSearch = debounce(fetchResults, 400ms)

4. ON query state change:

CALL debouncedSearch(query)

5. RETURN UI:

- Input field to update query

- Results list displayed from `results` state

**Backend:**

ON HTTP GET REQUEST to /api/search with query parameter:

1. SET query = req.query.query or empty string

2. PERFORM database query:

SELECT * FROM add_products WHERE title LIKE '%query%'

3. FOR EACH matching product:

UPDATE add_products SET search_count = search_count + 1 WHERE id = product.id

4. PERFORM another query to return products:

SELECT * FROM add_products WHERE title LIKE '%query%' ORDER BY search_count DESC

5. RETURN sorted product list as JSON response

### 3.3.2 Highlighted Products

This algorithm displays the top products in a particular section of page that are being bided live in the website, allowing users to see and bid if interested. It can be the most visited, most popular, most bided, most likable products etc.

**Frontend:**

ON COMPONENT MOUNT:

1. Initialize state:

- topProducts = empty list

2. MAKE GET request to /api/top_bids endpoint

3. IF response successful:

   SET topProducts state to response data

4. IF error occurs:

   LOG error message

5. DISPLAY topProducts list in UI

**Backend:**

ON HTTP GET REQUEST to /api/top_bids:

1. DEFINE SQL QUERY:

   SELECT product ID, product name, MAX(bid amount) as highest_bid

   FROM products table (add_products) JOIN bids table (user_biddings) ON product IDs match

   GROUP results by product ID and product name

   ORDER results by highest_bid descending

   LIMIT results to 10 rows

2. EXECUTE the SQL query

3. IF error occurs:

   LOG the error

   RESPOND with HTTP 500 status and error message

4. ELSE:

   RESPOND with JSON containing the top 10 products with highest bids

### 3.3.3 User Biddings and User Restriction

This algorithm allows user to bid some certain amount of money increased by certain percentile of the initial price within a single click and there are some validations set for the

user to bid in given time a single user can never bid 2 times simultaneously until the break time for the user to bid again ends.

*BEGIN*

1. CALCULATE Auction End Time:

IF product.submitted AND product.days exist THEN

SET postDate = new Date(product.submitted)

SET endDate = postDate + product.days (in days)

SUBTRACT 30 minutes from endDate

START timer to update every second:

SET now = current time

CALCULATE timeRemaining = endDate - now

IF timeRemaining <= 0 THEN

DISPLAY "Auction Ended"

ELSE

DISPLAY remaining time as "Xd:Xh:Xm:Xs"

IF timeRemaining <= 30 minutes THEN

SET isLast30Min = true

2. FETCH HIGHEST BID:

CALL /highestBid/:productId

IF success THEN

SET highestBid

3. FETCH LAST BID BY USER:

CALL /lastUserBid/:productId/:userId

SET lastUserBidTime if available

4. FETCH BID HISTORY FOR PRODUCT:

CALL /productBidUsers/:productId

SET bidUserIds = list of userIds who placed bid

5. ON BID INPUT OR ONE-CLICK BID:

CHECK isLast30Min

IF TRUE AND user_id NOT IN bidUserIds THEN

DISPLAY "New users cannot bid in the last 30 minutes"

EXIT

CALCULATE current time

IF current time - lastUserBidTime < 10 minutes THEN

DISPLAY "You must wait 10 minutes before bidding again"

EXIT

CALCULATE BID AMOUNT:

IF one-click bid THEN

- Set bid = product.price * 1.2 (rounded)

SUBMIT BID:

SEND POST to /submitBid with:

- productId

- userId

- bid amount

IF success THEN

UPDATE lastUserBidTime = current time

ADD user_id to bidUserIds

DISPLAY "Bid submitted successfully"

ELSE

DISPLAY error toast

6. ON FORM SUBMIT:

CALL above bid logic

*END*

# Chapter 4: Implementation and Testing

## 4.1 Implementation

Some of the tools and technologies that were used in this project are listed below:

### 4.1.1 Tools Used

- Visual Studio Code – VS Code: It is a lightweight, opensource code editor developed by Microsoft. It is widely used by developers to create multiple websites, application and programs, module etc. It is a cross-platform software supports on Windows, MacOS and Linux. It supports extension for additional functionality and provides code completion and debugging capabilities.
- DBMS: DBMS is a database management software that allows user to store, retrieve, update and manipulate data effectively. As to store the user records of this application, we have used XAMPP as controller, Apache as server and MySQL as query language.

### 4.1.2 Languages Used

The tools listed below will be used for the complete development of this project. Let us discuss the use of each of them in our project:

Tools like *VS Code, Node JS extension* and *Xampp Control* will be used for the code implementation.

*MERN* stands for *Mango DB, Express JS, React JS and Node JS* but MySQL database in case of Mango DB with other java script features will be used in this website.

**MySQL DB:** It is one of a popular database system used in multiple cross platform software to store the data of users, products and content of the websites. We will be using this database to store and retrieve users and movies information.

**Express JS**: A web application framework for Node.js, designed for building robust web applications and APIs. It simplifies the process of writing server-side code, handling HTTP requests, and routing.

**React JS**: It is a framework that provides us with multiple features such as libraries and CDNs allowing developer to use it without writing all the syntax and codes. It is basically used to create UI i.e. outer face of a website for users.

**Node JS**: A runtime environment that allows JavaScript to be executed on the server side. It uses an event-driven, non-blocking I/O model, making it lightweight and efficient for building scalable network applications.

### 4.1.3 Implementation Details of Modules

- **Index Page**

Index page is the entry page for the website where data, records and information about website or application is provided. It consists of data related to web-app, navigation menu, user registration and login page to enter inside the portal. Our index page includes a small navigation menu consisting login registration and about section of website. User can visit us and see the information provided in the webpage, contact us through multiple social platforms given in the about page and lastly user can register and login to the server.

- **Login Page**

Login is the gate of every software or website. It includes username or email and password to enter inside the pages. We have included email and password system in our login page so that user can easily enter without any other verification. If the provided email and password is not matched in the database, a user is not valid or is not allowed to pass through the gate. So, a user must be registered into the server to login.

- **Registration Page**

Registration is the collection of data to be stored in the database by which a user can enter into the portal after the login successful. If a user is not registered and tries to login, they can't go further. We have included user name, user email, phone number and password in the registration page where the data submitted by the user is directly sent to the database and stored as record file for user which is used to verify the user during the login.

- **Forget Password**

In case if user is unable to remember password or is not able to login because of some issue, they can easily retrieve their password from the forget page. After user clicks in the link 'Forget Password', they are redirected to the recovery page where user needs to go through 2 steps verification. Let's discuss these 3 steps recovery process by which a user can recover their lost password:

Step 1: First page of recovery simply ask user to enter their name and email that was used during the registration. If the username and email are matched in the records of database, they are taken to the next page.

Step 2: Second page of recovery ask user for the password verification code i.e. OTP one time password which is sent to the user through the same email which is verified from the database. After the successful verification, user is taken to the next page.

Step 3: Third page of verification ask user to enter their new and confirm password. After user enter the password, it is then sent to the database and updated in the field with new password. Now user is redirected to the login page where they can login to the account.

- **User Panel**

User panel is the interface provided to the user after the successful login. It includes multiple function and buttons creating easy access to the user. There are many functionalities available for users which are listed below:

**User-dash Board**

User-dash shows all the listed post about empty room so that user can see the recent post and create a deal if interested. Posts are visible as per the time they were posted. The last post is visible at top. A user post is not visible to himself unless he searches for it but is visible to all other users at the same time.

**Search Option**

Search is available at the top of webpage in between the navigation bar. If user wants to choose the room as per the need, they can easily search the area they are interested. It filters the area from the database, and display all the post related to the searched item. It can be activated from all the page of user panel.

**User Profile**

Profile of user keeps the record of each user when they submitted the data during the registration. The upload is not given in the registration page but if they are interested to upload profile picture, they can easily upload simply by clicking in profile button of 'Profile page.

**Edit Profile**

Edit option is available to let user update the records they want. There are multiple fields where they can submit the details about them so that it will be easy for the both renter and landlord to find each other. They can easily change the details and update any data they want.

**User Products**

All the products of user are displayed in this section. These posts can be updated any time from this section. The product list is also displayed in same section but different compartment.

**Payment Integration**

User is required to pay some specific amount of money in order to post the details online so there are some options available for them to pay the price online. eSewa has been integrated for the local users as the online payment method.

**User review**

User is able to give their reviews about the overall use of this application which will help people by providing quick visual clue to potential customer about the quality of features and service.

- **Admin Panel**

Admin has all the control over the web application so admin panel consist most of the functionality than that of user. Admin needs to deal with the problems and find out the solution of each issue so they might need to work a lot more than user. As it includes interface with multiple options to create, read, update & delete users and user's post. Some are listed below:

**Admin-dash Board**

Admin dash board shows all the post that needs to be approved before making it live in the application. It has many other options such as edit, create, manage, search, and the total number of available data as per the given four fractions.

**List of Approved Post**

After admin approves the post of users, they are notified through the mail notification and the list of approved post are stored in particular table that can be accessed in this section.

**Create, Update & Delete Section**

Admin can create new user account simply by clicking on button 'Create' in the top right corner of admin panel. As like that they can update their account and manage the login data from the listed buttons in the top right corner.

Admin can edit the user post and update the records of users as well by clicking on 'Edit' in the user list or post list or by searching them in search bar. They can delete each user or post as well.

**Admin Account**

Admin account consists of the details that are provided by the admin themselves. They can edit or update the data as per their need and are able to change their login email and password.

## 4.2 Testing

The goal of testing is to improve the quality and reliability of the product or system and to minimize the risk of errors or failures that could cause harm to users or property damage. So, our system also went for multiple unit testing, integration testing and system testing.

### 4.2.1 Test case for unit testing

Unit testing let us move forward to create a complete program without showing issue when it is attached after the successful result. Small components are easier to test during the coding so they were tested right at the moment when program was created.

**Table 4.2.1.1: Unit Testing for all user (admin/user) functionalities**

| Test No | Test Cases | Precondition | Input data | Expected result | Actual result | Pass/Fail |
|---|---|---|---|---|---|---|
| 01 | If registration is valid | Valid registration should bypass the form and email verification is asked | User data | Form data should be submitted successfully | Registration details were submitted successfully | Pass |
| 02 | If user can create multiple account with same email | User should not have their account in this website with the provided email | User data | Alert message with user already existed | Alert message with user already exist was shown | Pass |
| 03 | Email verification | Google verified Email should be provided | User email | Email verification and creation of user account | Code was successfully verified and account was created | Pass |
| 04 | If user is valid for login | User should be already registered | Verified email and password | Successful login to the user dashboard | User was successfully logged in to the user dashboard | Pass |
| 05 | If forget password is working | User should be already registered | Registered username and email | Redirect user to next page for | User was sent OTP and password | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | OTP verification | was reset successfully | |
| 06 | Update user account | User should be logged in to the website | As per needs (image, username, gender, phone, email, password etc.) | Alert message as Update successful | Alert message was shown and update was successful | Pass |

**Table 4.2.1.2: Unit Testing for only user functionalities**

| | | | | | | |
|---|---|---|---|---|---|---|
| 09 | User product validation | User must fill up all the details in the form | Product details | Error message with respected field can't be empty | Message was shown as respected field can't be empty | Pass |
| 10 | Updating user product | User should at least have a post to update | New product details | Alert message saying update successful | Alert message was shown as update successful | Pass |
| 11 | Search Bar | User must provide valid data | Pokhara, Butwal and Kathmandu | Search result related to Pokhara, Butwal and Kathmandu if available | Result was shown as per the user input | Pass |
| 12 | Filter Search | User must fill all the form to search the data | Antique, Clothing, Communication etc | Search related to Antique, Clothing, Communication etc should be shown if available | Result was shown as per the submitted data | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| 13 | Delete user post and user account | User must have at least one post | Click functionality | User alert message for confirm and operation | User was alerted and deletion was successful | Pass |
| 14 | User review | User must be logged in to the system | User reviews | Reviews should be submitted successfully | Review was submitted successfully | Pass |
| 15 | Message to admin regarding their issue | User must have accounts | Username, user email and message | Mail should be submitted successfully | Mail was submitted successfully | Pass |

**Table 4.2.1.3: Unit Testing for admin functionalities**

| | | | | | | |
|---|---|---|---|---|---|---|
| 16 | If user registration is submitted | User should not be already existed | User data | Records should be displayed in the system table | Records were displayed in the system table | Pass |
| 17 | If user post detail is submitted | Form validation must be successful | Product data | Records should be displayed in the system table | Records were displayed in the system table | Pass |
| 18 | Mail to users | Users email should be google verified | Input is given by the admin themselves | Mail to selected user's email id should be sent | Mail was sent and received successfully | Pass |
| 19 | If admin is able to alter records | Records data (post data or user data) | Click functionality or search for the user/post | Record should be updated in database and updated value must be displayed | Records were altered successfully | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| 20 | If admin is able to search user and post | Valid data should be provided | Data to search i.e. (username, post details like area, city etc.) | Result related to data should be fetched | Result was shown as per the user input | Pass |

### 4.2.2 Test case for system testing

System testing evaluates the entire system or software application as a whole, rather than testing individual modules or components. The objective of system testing was to verify that the software meets its specified requirements and performs as expected in its intended environment.

**Table 4.2.2.1: System Testing of whole application**

| Test No | Test Cases | Expected Result | Actual Result | Pass/Fail |
|---|---|---|---|---|
| 01 | User registration, login, data upload and data search | Successful user registration, login, data upload and data search | All the actions were successful | Pass |
| 02 | User payment for products and user review | Successful payment and user review system | Payment was successful and user was able to give review | Pass |
| 03 | Management system of admin panel | All the mechanism and functions should response and provide error free results | All functions were working fine as per the command instructed for them | Pass |

# Chapter 5: Conclusion and Future Recommendations

## 5.1 Lesson Learnt

After the completion of this project, we have fully understood about team work and its effectiveness in the project management. It helped us to gain more experience in analyzing the problems, finding requirements and handling the problem within the limited amount of time. It helped in creating mutual understanding between team members and share the general knowledge and ideas regarding the project issue and management process. It also helped to share our prospective towards each other and help each other to learn. Many mistakes were framed during the project so that now we can minimize such mistakes in upcoming event.

Being a team member, one must trust others and help them by sharing thoughts and ideas on the subject they are working on. Team leader should appreciate and give their feedback to encourage them to work harder. Picking the challenges and analyzing the possibility helps in making great leadership and complete the project.

## 5.2 Conclusion

Online auction system will give new approach and dimension to auction system. It is a new experience and has greatly impacted the lives of consumers in its short time of existence. This system has made consumers more effective and efficient in their behavior and has driven businesses to a new level, forcing many to make the necessary adjustments and changes to reach the new market of knowledgeable consumer. The huge benefit of using online auction sites to buy any items is that you can place offers at all hours of the day, also it removes geographical boundaries, location etc.

In summary, the designed online bidding system prioritizes scalability, responsiveness, and data consistency. The system's capacity estimation aligns with high scalability targets, ensuring it can handle a large user base and concurrent bids effectively. Despite challenges and trade-offs, the chosen solutions position the system for success in delivering a seamless and reliable experience for bidders and sellers in the dynamic e-commerce landscape.

## 5.3 Future Recommendations

The development of this project has been made with different features and functionalities but if it is necessary then can be updated with some more modification and features to create better UI. It is already at its maximum potential as most of the needed functions have

been provided to make it more robust and flexible which can serve at any basic environment or platform.

Some of the things can be suitable scope to add in the future:

- Different payment methods for payment transaction
- Mobile Based Application or mobile responsiveness
- Chat message to interact between admin and users

# REFERENCES


[1] Rosemet LLC, "Agile Methodology Steps," [Online]. Available: https://www.rosemet.com/wp-content/uploads/2024/07/WHATAR1.jpg. [Accessed: Mar. 1, 2025].

[2] "Role of Feedback in Auction Platforms: Auction Feedback and Reputation Trust and Transparency," FasterCapital, [Online]. Available: https://fastercapital.com. [Accessed: Mar. 1, 2025].

[3] J. Sow and P. Anthony, "Online Auctions Efficiency: A Survey of eBay Auctions," 2012.

[4] J. Trevathan, H. Ghodosi, and W. Read, "Design issues for electronic auctions," in *E-Business Telecommunication Networks*, 2005.

[5] M. A. Bouaicha, A. Bakdi, A. Ait Ouahmed, and A. Kharraz, "Shill bidding prevention in decentralized auctions using smart contracts," *arXiv preprint* arXiv:2505.12345, May 2025.

[6] S. M. Yasir, A. Gir, and J. Lundberg, "An integration of policy and reputation based trust mechanisms," *arXiv preprint* arXiv:2406.12345, Jun. 2024.

[7] "Building Trust through Transparency in Online Auctions," FasterCapital, [Online]. Available: https://fastercapital.com. [Accessed: Mar. 2, 2025].

[8] M. Kumar and S. Feldman, "Secure online English auctions," in *Proc. 3rd Int. Conf. on E-Business and Telecommunication Networks (ICETE)*, 2006, pp. 25 - 30.

[9] "Ensuring Secure Access to Auction Platforms," FasterCapital, [Online]. Available: https://fastercapital.com. [Accessed: Mar. 2, 2025].

[10] "Bidder-Anonymous English Auction Scheme," *ScienceDirect*, 2007, [Online]. Available: https://sciencedirect.com. [Accessed: Mar. 2, 2025].

[11] "Criticisms and Controversies," *Wikipedia*, eBay, Mar. 2025, [Online]. Available: https://en.wikipedia.org/wiki/EBay#Criticism_and_controversies. [Accessed: Mar. 5, 2025].

# APPENDICES

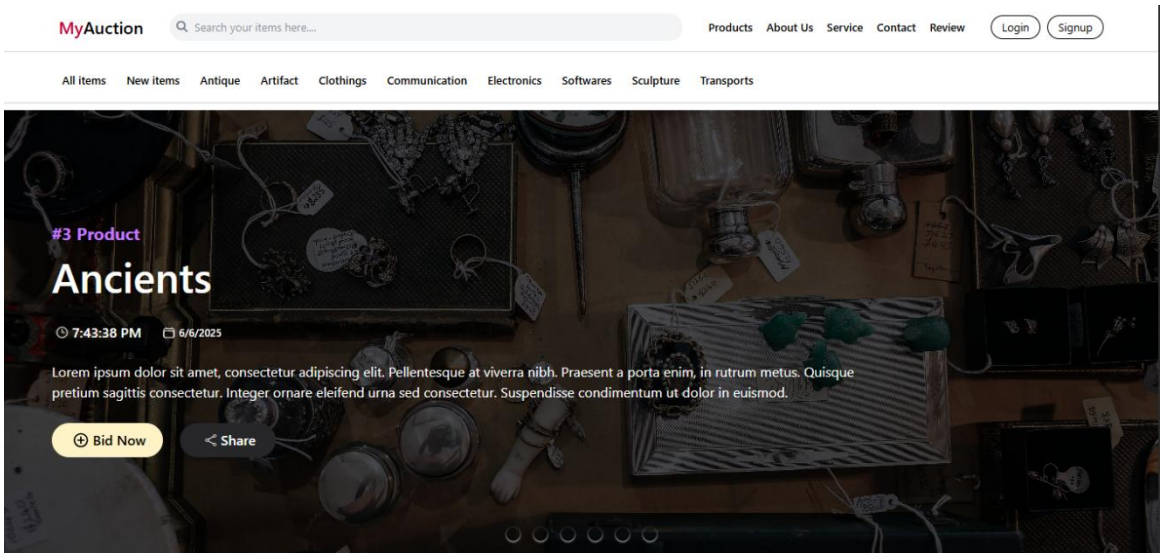



**Fig AP1: Index Page**

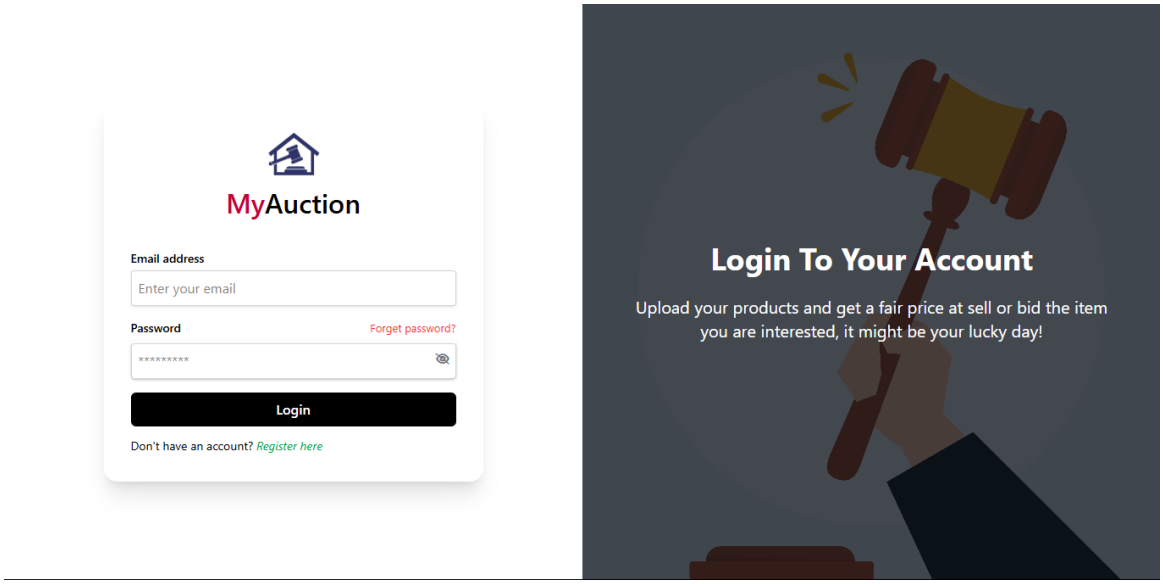



**Fig AP2: Login Page**

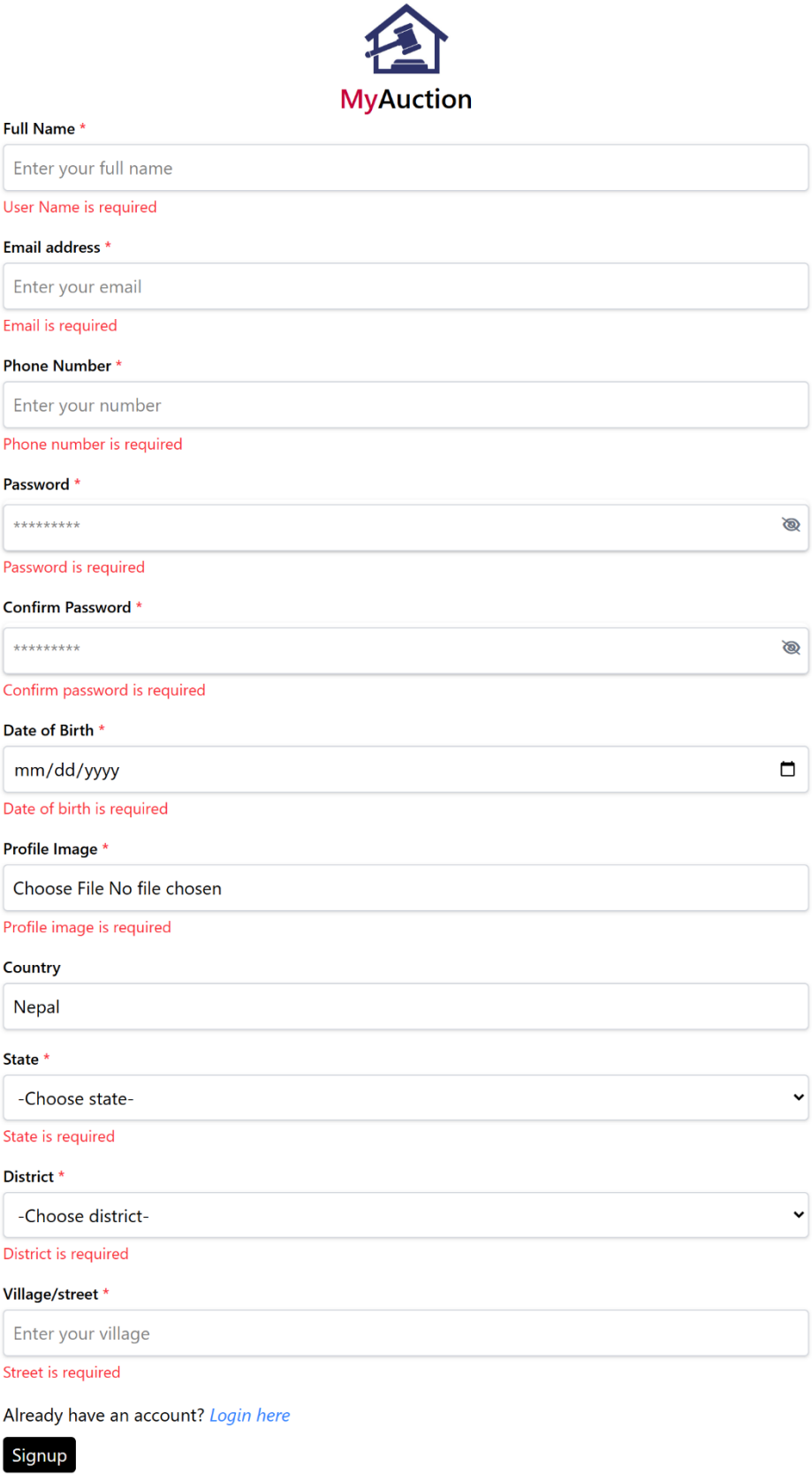



**Fig AP3: Registration Page**

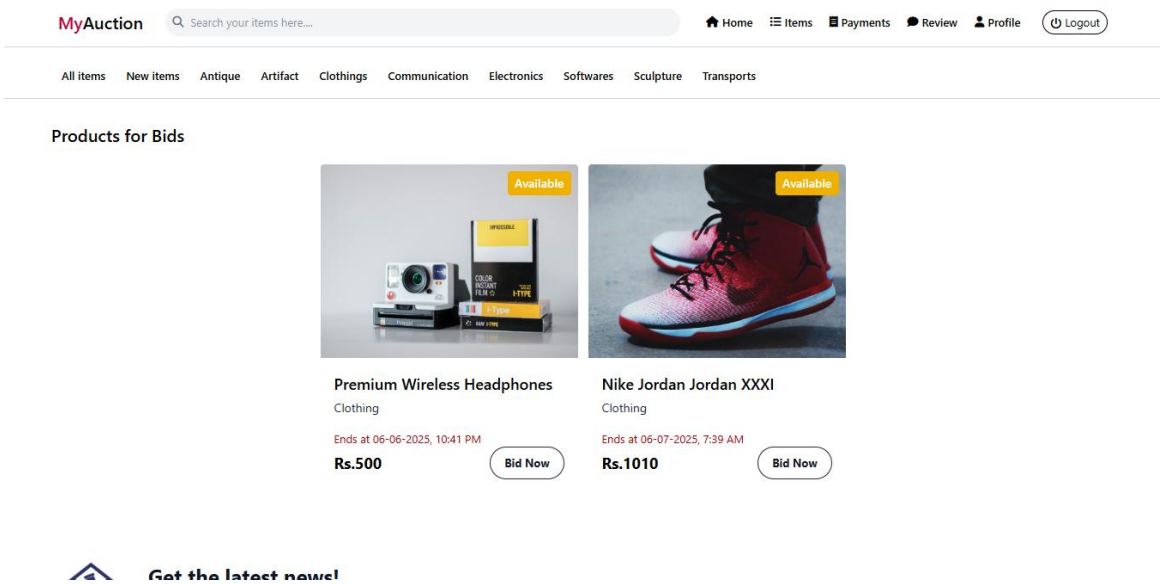



**Fig AP4: User Dashboard**

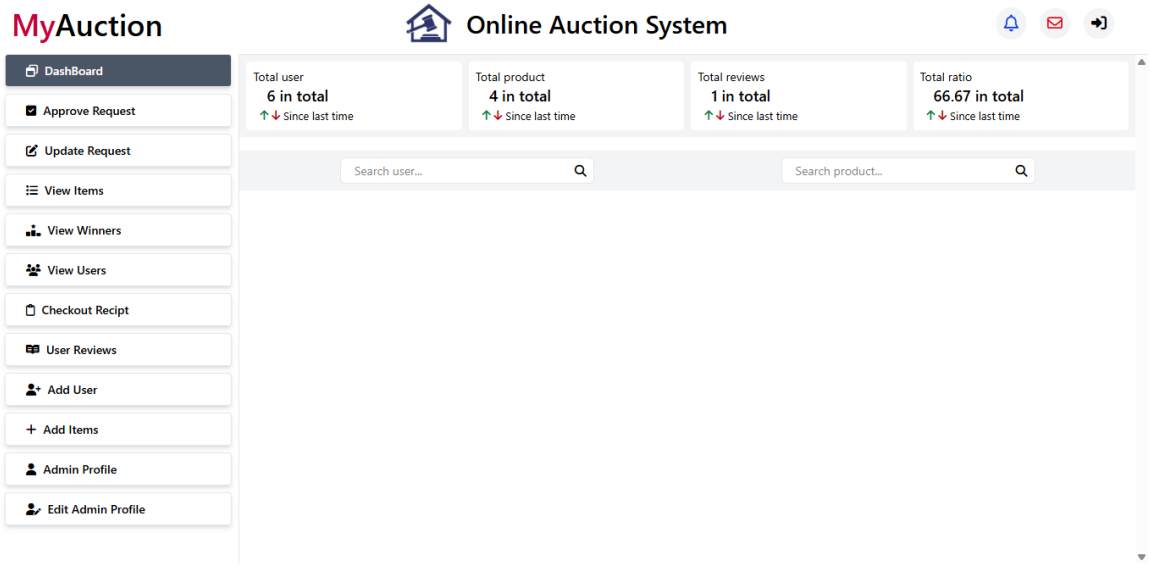



**Fig AP5: Admin Page**