VIDYABHARTI TRUST COLLEGE OF BUSINESS, COMPUTER SCIENCE & RESEARCH. UMRAKH

PROJECT REPORT

AS A PARTIAL REQUIREMENT

FOR THE DEGREE OF

BACHELOR OF COMPUTER APPLICATION

(B.C.A)

YEAR: 2022 - 2023

"ONLINE AUCTION SYSTEM"

GUIDED BY:

Ms. Rinkal Patel

SUBMITTED BY:

Mr. Mishra Saurabh (2096)

Mr. Mistry Madan (2100)

Vidyabharti Trust College of Business, Computer Science & Research. Umrakh

At & Po: Umrakh, Ta: Bardoli, Dist: Surat – 394345, Gujarat, INDIA.

CERTIFICATE

This is to certify that Mr. ***Mishra Saurabh , Mistry Madan*** *Exam Seat Number:* ***2096 , 2100*** *has satisfactorily completed their project work entitled* ***Online Auction System*** *as a partial fulfillment of the requirements for* ***6th Semester –B.C.A. (Bachelor of Computer Application)****, during the academic Year 2022 - 2023.*

Date:

Principal

Dr. Snehal H. Mistry

Place:

Vidyabharti Trust College of Business, Computer-Science & Research. Umrakh, Bardoli, Surat

PROJECT OF B.C.A.
Academic Year_____
Approved by: _____
_____
_____
(Examiners)

ACKNOWLEDGEMENT

It gives us great pleasure in presenting this project report titled **"Online Auction System"** and we wish to express our immense gratitude to the peoplewho provided invaluable knowledge and support in the completion of this project. Their guidance and motivation has helped in making this project a great success.

We express our gratitude to our project guide **Ms. Rinkal Patel** who provided us all the guidance and encouragement throughout the project development. We wouldalso like to express our sincere gratitude to the respective Project coordinators.

We are eager and glad to express our gratitude to the Head of the BCA Dept. **Prof. Amit Patel** for his approval of this project. We are also thankful to him for providing us the needed assistance, detailed suggestions and also encouragement to do the project.

We would like to express our sincere gratitude to our respected principal **Dr. Snehal Mistry,** vice principal **Dr. Payal Mahida** and the management of our College for providing such an ideal atmosphere to build up this project with well-equipped library with all the most necessary reference materials and up to date IT Laboratories. We are extremely thankful to all staff and the management of the college for providing us all the facilities and resources required.

Thanking All,

Mistry Madan
Mishra Saurabh

ABSTRACT

Online auctions have revolutionized the way people buy and sell goods and services. The Online Auction System (OAS) is a platform that enables buyers and sellers to interact and conduct transactions in a virtual environment. The OAS is designed to be easy to use, secure, and transparent It provides an efficient marketplace for buyers to bid on items and for sellers to list and sell their items. The system incorporates a bidding mechanism that allows buyers to compete with each other in real-time, ensuring that sellers receive the best possible price for their items. The OAS also provides tools for sellers to manage their listings, monitor bidding activity, and communicate with buyers The platform offers a variety of payment options, ensuring that transactions are completed quickly and securely Additionally, allowing buyers and sellers to rate each other and build trust within the community with the rise of e-commerce and the growing demand for online marketplaces, the OAS is poised to continue its growth and success in the year to come.

Index

1. Introduction

A world that loves to participate in real-time activities would certainly appreciate the idea of an online auction system that opens 24/7. Unlike the traditional auctioning method, this modern system brings together buyers and sellers from across the globe to a single location. Since the items are listed for few days, bidders can think and study the deal before bidding. Since, there is no geographical or time restrictions, the number of bids received will be more.

Online Auction System Allow Users to sell used or new product as conductiong Auction of Products. An Online Auction System is a digital platform that enables users to buy and sell goods and services through an online bidding process. It is a virtual marketplace that allows people from all over the world to participate in the auction without physically being present at the auction location.

The system is designed to allow users to setup their products for auctions and bidders to register and bid for various products available for bidding. This project developed using PHP and all the record stores in MySQL database.

1.1 College Profile

Project Title	Online Auction System
College Name	Vidyabharti Trust College of Business, Computer Science & Research, Umrakh
Address	Vidyabharti Campus, At & P.O. Umrakh-Ta- Bardoli Surat, Gujarat 394345, India
Email Id	bbabca@vtcbb.edu.in
Contact No	(026622) 220581
Internal Guide Name	Miss. Rinkal Patel

1.2 Project Profile

PROJECT TITLE	Online Auction System
PROJECT DEFINATION	An online auction system is a digital platform that allows users to buy and sell goods or services through an auction-style bidding process over the internet. It enables Users to old products for sale and allows other peoples to place bids on those products. The highest bidder at the end of the auction period wins the product and completes the transaction with the owner of product.
DURATION	3 Months
FRONTEND	PHP
DATABASE	MYSQL
PLATFORM	Visual Studio Code
OPRATING SYSTEM	MICROSOFT WINDOWS 11
PROJECT INTERNAL GUIDE	Ms. RINKAL PATEL

2. Proposed System

2.1 Features & Scope

The Online Auction System is designed to allow users to set up their products for an auction and bid for various products available for bidding.

- Registered users can participate in auction from anywhere at any time through online auction.
- Those who wish to take part in bidding or sell products at the site have to register at the site as customer. Only authenticated users can take part in product selling or in bidding.
- This will store bidders record and bidding record.
- Auction winners and losers get an Email notification whether they won or lost the bid.

2.2 OBJECTIVE

Online Auction System is the website where registered users can bid for the products and can sell used or new products for auction. In online auction project that holds online auctions of various products on a website and serves other users accordingly.

The objective of an Online Auction System is to provide a platform where users can buy and sell products in an efficient and transparent manner. The system helps to users to sell and buy products. This is used sell and buy anything on the website from house.

The specific objectives of an online auction system project could include:

1. Providing a user-friendly interface for users to list products, place bids, and manage auctions.
2. Implementing a fair and transparent bidding process to prevent fraudulent activity.
3. Providing administrative tools to view & verify users, auction products and complaints.

2.3 Constraints

2.3.1 H/w Constraints

- Minimum Microsoft windows 7 required.
- Google chrome, Mozilla Firefox any other browser required.
- MySQL database

2.3.2 S/W Constraints

- Band width 10 mbps minimum required.
- Processor 1 GB RAM and 100 GB HDD minimum required.

2.4 Advantages

- **Faster viewing and retrieval of data**

With this new system data will be viewed faster and data will be retrieved as per the requirements. No information about anybody will be left out.

- **No moving:**

Items are pictured, sold and picked up from the same location. Since it is an online auction, people can participate in it from the comfort of their homes. Since it is an online auction, people can participate in it from the comfort of their homes and sell their products.

- **Convenience:**

Buyers can bid when and where they want and bid on multiple auctions in the same day.

- **Exposure:**

Buyers can search and find products of interest because every product is categories.

- **Time Saving:**

User sell or purchase products that are brand new or used. Online Auction System provides an accessible platform for general consumers to buy and sell items and save time for user to sell new or used Products.

2.5 Limitation

- User Registration is compulsory to sell products and bid for available Auction
- It is compulsory to verify your account through email.
- Need Internet to run all functions
- For Payment only qr-code and cod are available

3. Environment Specification

Implementation is used here to mean the process of converting a new or a revised system design into an operational one. Implementation of a new system to replace an existing one is usually difficult. If not properly planned, there can be many problems. This phase is less creative than system design and is primarily concerned with user training; the type of implementation that is incorporated in the development of the project is implementation of a new application to replace an existing one.

3.1 Hardware & software requirements

HARDWARE	
Development Environment	Processor – Intel Core i3 RAM-1GB HDD-100GB Operating System-Windows 11
Clint Configuration	Internet-512 kbps (Minimum)
Web Hosting	Windows Server 2003 and above
Web Requirement	Band width-10mbps (Minimum)

SOFTWARE	
Front End	PHP
Back End	MYSQL
Scripting	JavaScript
Other Technology	HTML XAMP server
Browser	Google Chrome, Mozilla Firefox

3.2 Development Description

- **PHP: Server Site Scripting Language**

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1994, the reference implementation of PHP (powered by the Zend Engine) is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Pre-processor, which is a recursive backronym.

PHP code can be simply mixed with HTML code, or it can be used in combination with various templating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable.

After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications.

The canonical PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

PHP development began in 1994 when Rasmus Lerdorf wrote a series of Common Gateway Interface (CGI) binaries in C, which he used to maintain his personal homepage.

- **MYSQL: The World Most Popular Open-Source Database**

Many of the applications that a Web developer wants to use can be made easier by the use of a standardized database to store, organize, and access information. MySQL is an Open Source (GPL) Standard Query Language (SQL) database that is fast, reliable, easy to use, and suitable for applications of any size. SQL is the ANSI-standard database query language used by most databases (though all have their nonstandard extensions).

MySQL can easily be integrated into Perl programs by using the Perl DBI (Database Independent interface) module. DBI is an Application Program Interface (API) that allows Perl to connect to and query a number of SQL databases (among them MySQL, mSQL, PostgreSQL, Oracle, Sybase, and Informix).

MySQL is a powerful, sophisticated, and easy-to-use SQL database program. Using Perl and DBI, one can easily create programs to automate database management tasks. With this knowledge, the prospective web designer should be able to construct a database (for lack of a better term) web site that is portable, sophisticated, easy to manage, and professional appearing. We have examined only a small subset of all that MySQL provides.

❖ **PHPMAILER: A full-featured email creation and transfer class for PHP**

Many PHP developers need to send email from their code. The only PHP function that supports this directly is mail (). However, it does not provide any assistance for making use of popular features such as encryption, authentication, HTML messages, and attachments.

Formatting email correctly is surprisingly difficult. There are myriad overlapping (and conflicting) standards, requiring tight adherence to horribly complicated formatting and encoding rules – the vast majority of code that you'll find online that uses the mail () function directly is just plain wrong, if not unsafe!

The PHP mail () function usually sends via a local mail server, typically fronted by a send mail binary on Linux, BSD, and macOS platforms, however, Windows usually doesn't include a local mail server; PHPMailer integrated SMTP client allows email sending on all platforms without needing a local mail server. Be aware though, that the mail () function should be avoided when possible; it's both faster and safer to use SMTP to localhost.

➢ **Features**

- Probably the world's most popular code for sending email from PHP!
- by many open-source projects: WordPress, Drupal, 1CRM, SugarCRM, Yii, Joomla! and many more
- Integrated SMTP support – send without a local mail server
- Send emails with multiple To, CC, BCC, and Reply-to addresses
- Multipart/alternative emails for mail clients that do not read HTML email
- Add attachments, including inline

- Support for UTF-8 content and 8bit, base64, binary, and quoted-printable encodings
- SMTP authentication with LOGIN, PLAIN, CRAM-MD5, and XOAUTH2 mechanisms over SMTPS and SMTP+STARTTLS transports
- Validates email addresses automatically
- Protects against header injection attacks
- Error messages in over 50 languages!
- DKIM and S/MIME signing support
- Compatible with PHP 5.5 and later, including PHP 8.2
- Name spaced to prevent name clashes

❖ **ReCaptcha: reCAPTCHA v2 is a CAPTCHA service implementation from Google**

reCAPTCHA v2 is a CAPTCHA service implementation from Google. It provides website protection against automated bots and spam by adding an additional widget to web forms which verifies that the user accessing the page is a real human being. In contrast to the first version, users are now no longer required to solve hard to identify text CAPTCHA challenges each time they submit a form. They now only need to solve an image challenge once and afterwards only have to tick a box to successfully verify that they're indeed human.

- **A typical reCAPTCHA v2 challenge workflow looks as follows:**

  - If a user is considered trustworthy by Google's backend (e.g., because he has already solved a CAPTCHA challenge successfully before), the service will recognize this and mark the CAPTCHA as successfully solved. No further input is required.
  - If the user has not solved a CAPTCHA before or the Google backend cannot confidently predict if the user is human, a visual CAPTCHA challenge will be presented and the user is required to solve it:
  - Once the user has successfully solved the CAPTCHA (or was already automatically recognized as trustworthy), the protected form can be submitted.
  - Most of the time a user should not need to solve a CAPTCHA with reCAPTCHA v2. This makes it much more user friendly than the old implementation which required the user to identify sometimes hard-to-read text fragments. In addition to that, the new solution is much better suited for responsive storefronts.

4. System Planning

4.1 Feasibility Study

The system feasibility can be divided into the following sections:

- **Technology and system feasibility**

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not.

- **Economic feasibility**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost / benefit analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. If a benefit outweighs cost, then the decision is made to design and implement the system.

- **Legal feasibility**

Determines whether the proposed system conflicts with legal requirements, e.g., a Data Processing system must comply with the local Data Protection Acts

- **Operational feasibility**

  Is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

- **Schedule feasibility**

  Is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development

- **Resource feasibility**

  This involves questions such as how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and number of resources required, dependencies.

4.2 Software Engineering Model

- **Spiral Model: -**

The -spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. Spiral model is a combination of iterative development process model and sequential linear development model i.e., waterfall model with very high emphasis on risk analysis. It allows for incremental releases of the product, or incremental refinement through each iteration around the spiral.

- **Spiral model design**

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

- **Identification:**

  - This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase
  - This also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral the product is deployed in the identified market.

- **Design:**

  - Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and final design in the subsequent spirals.

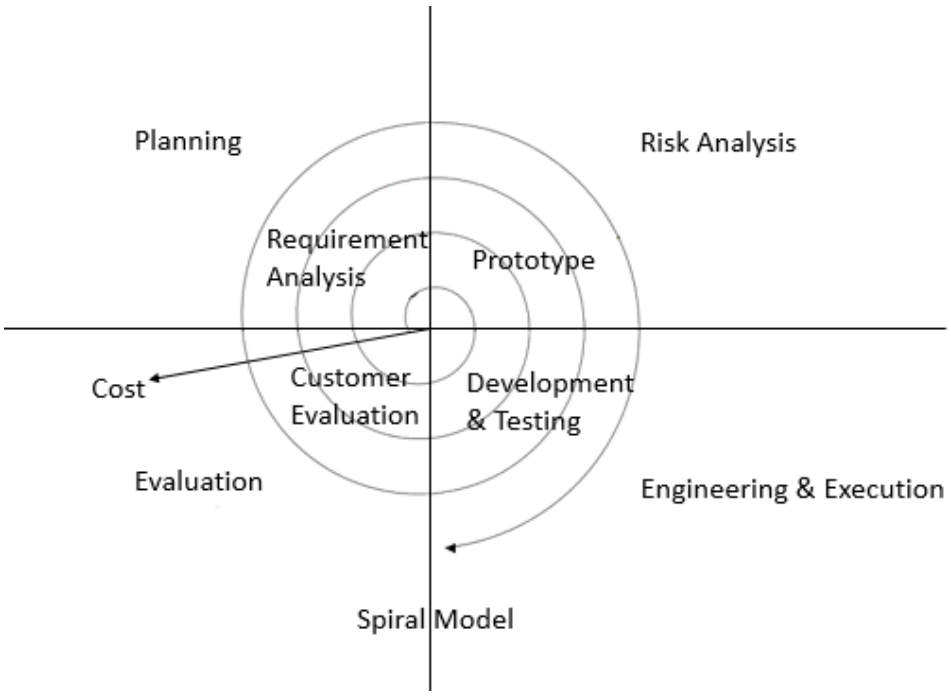
- **Construct or Build:**

  - Construct phase refers to production of the actual software product at every spiral. In the baseline spiral when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.
  - Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to customer for feedback

- **Evaluation and Risk Analysis:**

  - Risk Analysis includes identifying, estimating, and monitoring technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

- Following is a diagrammatic representation of spiral model listing the activities in each phase:



- Based on the customer evaluation, software development process enters into the next iteration and subsequently follows the linear approach to implement the feedback suggested by the customer. The process of iterations along the spiral continues throughout the life of the software.

- Spiral model application
  - Spiral Model is very widely used in the software industry as it is in synch with the natural development process of any product i.e., learning with maturity and also involves minimum risk for the customer as well as the development firms
  - Following are the typical uses of Spiral model:
  - When costs there is a budget constraint and risk evaluation is important.
  - For medium to high-risk projects.

- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which is usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

- **Model pros and cons**

  - The advantage of spiral lifecycle model is that it allows for elements of the product to be added in when they become available or known. This assures that there is no conflict with previous requirements and design.
  - This method is consistent with approaches that have multiple software builds and releases and allows for making an orderly transition to a maintenance activity. Another positive aspect is that the spiral model forces early user involvement in the system development effort.
  - On the other side, it takes very strict management to complete such products and there is a risk of running the spiral in indefinite loop. So, the discipline of change and the extent of taking change requests is very important to develop and deploy the product successfully.
  - The following table lists out the pros and cons of Spiral SDLC Model:

Props	Cons
Changing requirements can be accommodated.	Management is more complex.
Allows for extensive use of prototypes	End of project may not be known early.
Requirements can be captured more accurately.	Not suitable for small or low risk projects and could be expensive for small projects.
Users see the system early.	Process is complex
Development can be divided into smaller parts and more risky parts can be developed earlier which helps better risk management.	Spiral may go indefinitely. Large number of intermediate stages requires excessive documentation.

4.3 Risk Analysis

- **Schedule risk**

  - Project schedule get slip when project tasks and schedule release risks are not addressed properly.
  - Schedule risks mainly affected on project and finally on company economy and may lead to project failure.
  - Schedule often slip due to following reasons
    - Wrong time estimation resources are not tracked properly. All resources like staff, systems, skills of individuals etc.
    - Failure to identify complex functionalities and time required to develop those functionalities.

❖ **Operational risk**

- Risks of loss due to improper process implementation, failed system or some external events risks.
- Causes of Operational risks:
  - Failure to address priority conflicts Failure to resolve the responsibilities
  - Insufficient resources
  - No proper subject training
  - No resource planning
  - No communication in team

❖ **Technical risk**

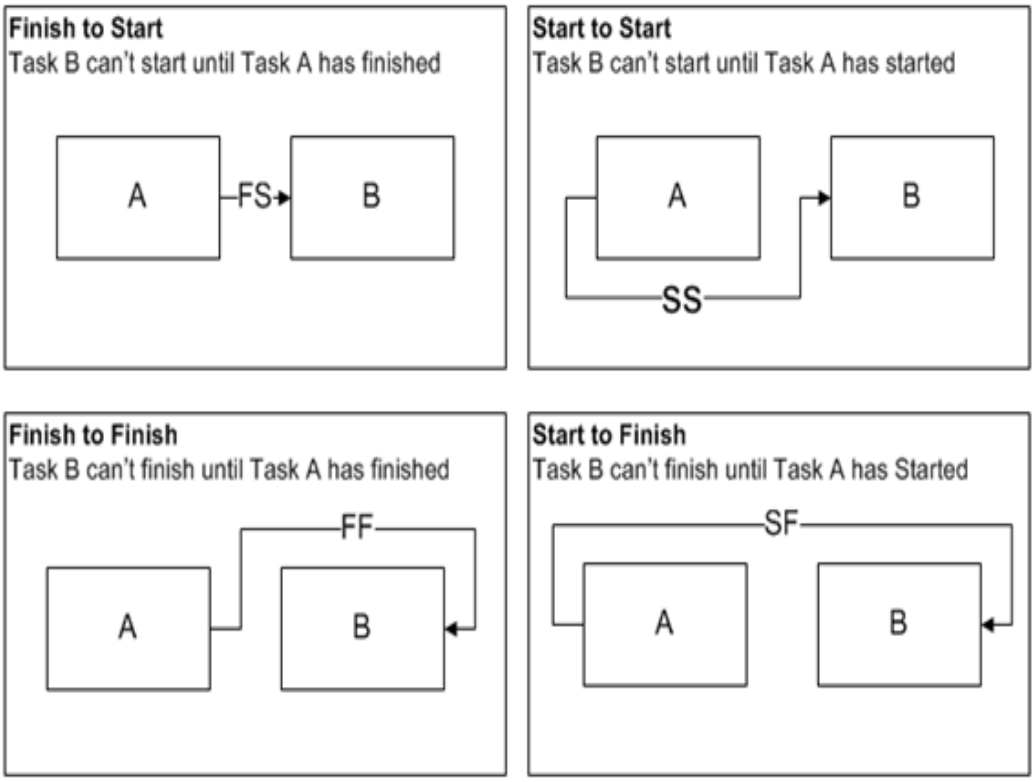
- Technical risks generally lead to failure of functionality and performance.
- Causes of technical risks are:
  - Continuous changing requirements
  - No advanced technology available or the existing technology is in initial stages.
  - Product is complex to implement.
  - Difficult project modules integration.

❖ **Programmatic Risk**

- These are the external risks beyond the operational limits. These are all uncertain risks are outside the control of the program.
- These external events can be:
  - Running out of fund Market development.
  - Changing customer product strategy and priority.
  - Government rule changes.

4.4 Project Schedule

4.4.1 Task Dependency: -



- Finish to start (FS): Task B cannot start until Task B is completed. For example, you need to assemble your pizza before you slide it into the oven.

- Finish to finish (FF): Task A can only finish when Task B ends. In software development, the QA team finds and reports bugs (Task A) and the engineering team fixes them (Task B). In this case, testing is complete only when bug fixes are completed.

- Start to start (SS): Task B cannot start before Task A starts. When Asphalt is poured on the road, it must be immediately levelled as it dries up quickly. So, the tasks of pouring Asphalt and levelling road must start together.

- Start to finish (SF): Task B must start for Task A to be completed. Consider the post of a security guard. A night guard will not be relieved until the morning guard takes charge.

4.4.2 Timeline Chart: -

Tasks	**1-JAN-2023 TO 30-MARCH-2023**										
Weeks	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	12
1.Introduction											
Requirement Gathering											
Analyze Gathered Information											
Determine Basic Functionality											
Determine Scope of system											
Milestone Completed											
2.Planning and Risk analysis											
Analyze data for possible risk											
Identify potential risk											
Determines different modules											
Milestone Completed											
3.Designing											
Design basic interface											
Design database table											
Design UI model											
Milestone Completed											
4.Coding and integrated modules											
Define code logic											
Implementation of logics											
Admin Modules											
User Modules											
Milestone Completed											
5.Testing											
Validate input data on control											
Check accuracy of transaction											
Test system with multiple user											
Milestone completed											

5. System Analysis

5.1 Detailed SRS

❖ **Modules :-**

1. **Admin: -**
   - Add / Remove Category & Subcategory
   - View & Remove Unverified Users

2. **User**
   - Register
   - Verify Account Through OTP
   - Change Password
   - Add Product for an auction
   - Bid for Active Products Auction
   - Update Profile
   - Search and select category wise Products
   - QRCode Scanning Payment

3. **Notification**
   - Send Notifications to registered Users of an important activity

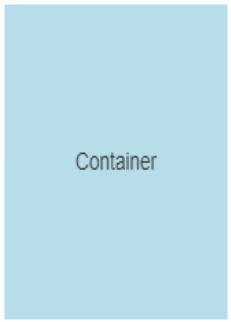
4. **Complaint**
   - User Register Complaint

5. **Payment**
   - Provide Payment options to users
     - Cash
     - QR Code Scanning

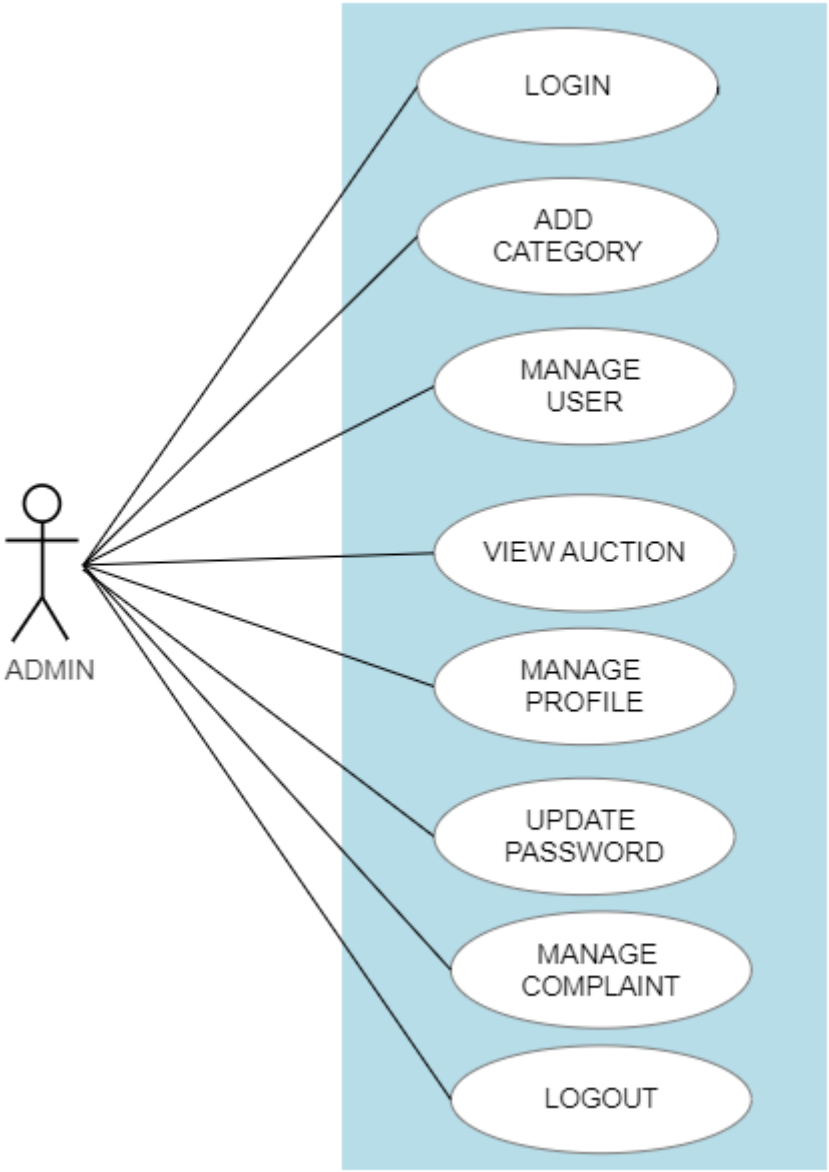
5.2 UML Diagrams

5.2.1 Use Case Diagram

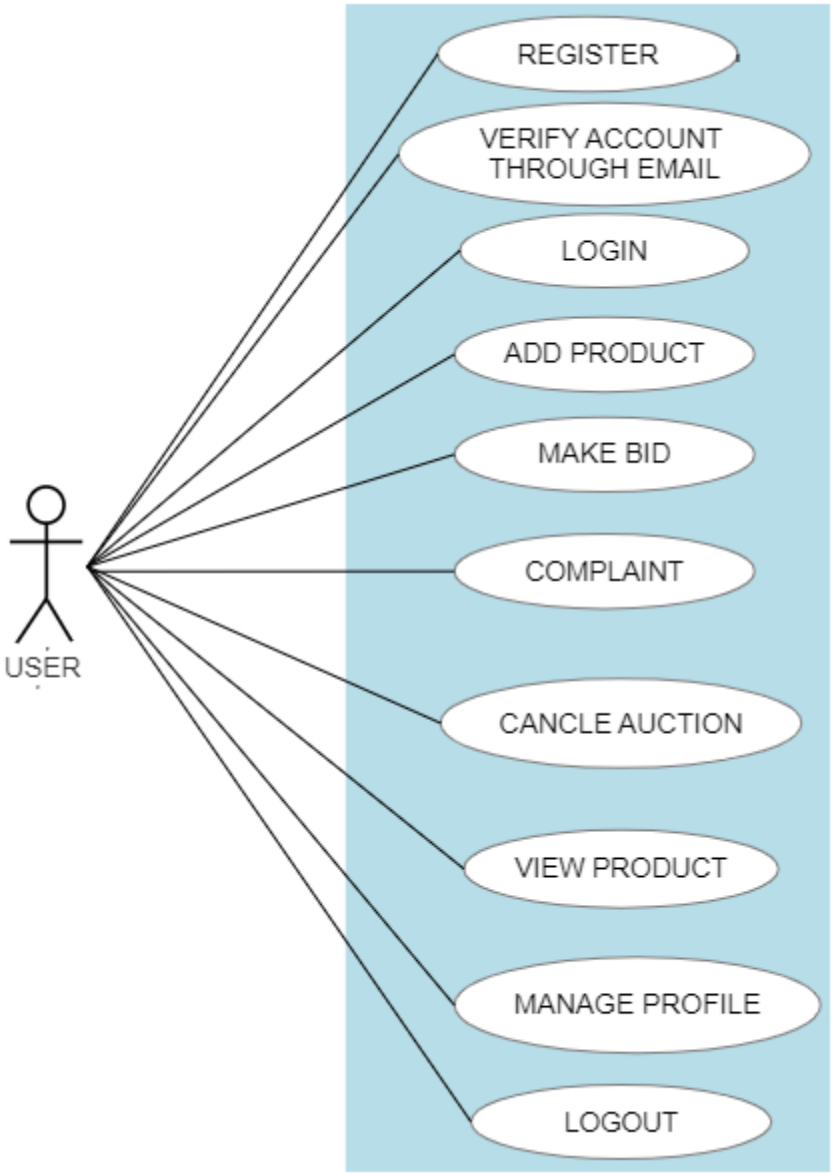
This Use Case Diagram is a graphic depiction of the interactions among the elements of Auction System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Auction System.

Symbol	Symbol Name	Symbol Description
	Actor	An actor is a person, group, or system that interacts with the use case.
	System Boundary	The system boundary separates a system from actors and other systems. To add elements to a system, create them outside the boundary, and then drag them into the boundary.
	Association	A relationship between two elements in the diagram. Associations between actors and use cases and represented by a solid line.
	Use Case	A use case describes a function that a system performs to achieve the user's goal.

1.Admin

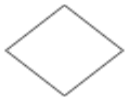
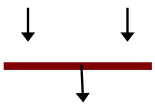

LOGIN
ADD CATEGORY
MANAGE USER
VIEW AUCTION
MANAGE PROFILE
UPDATE PASSWORD
MANAGE COMPLAINT
LOGOUT
ADMIN

1. User

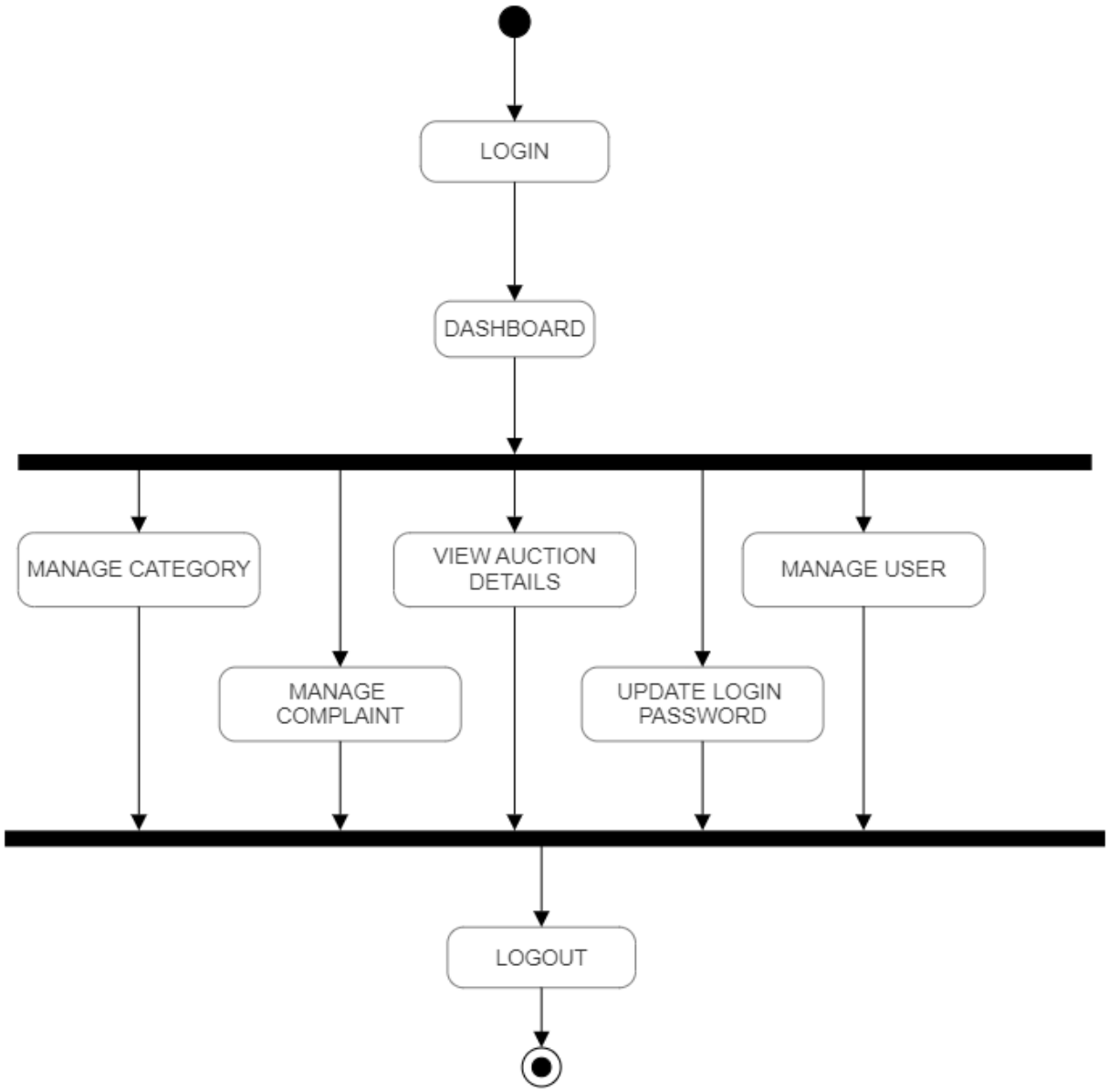

REGISTER
VERIFY ACCOUNT THROUGH EMAIL
LOGIN
ADD PRODUCT
MAKE BID
COMPLAINT
CANCLE AUCTION
VIEW PRODUCT
MANAGE PROFILE
LOGOUT
USER

5.2.2 Activity Diagram

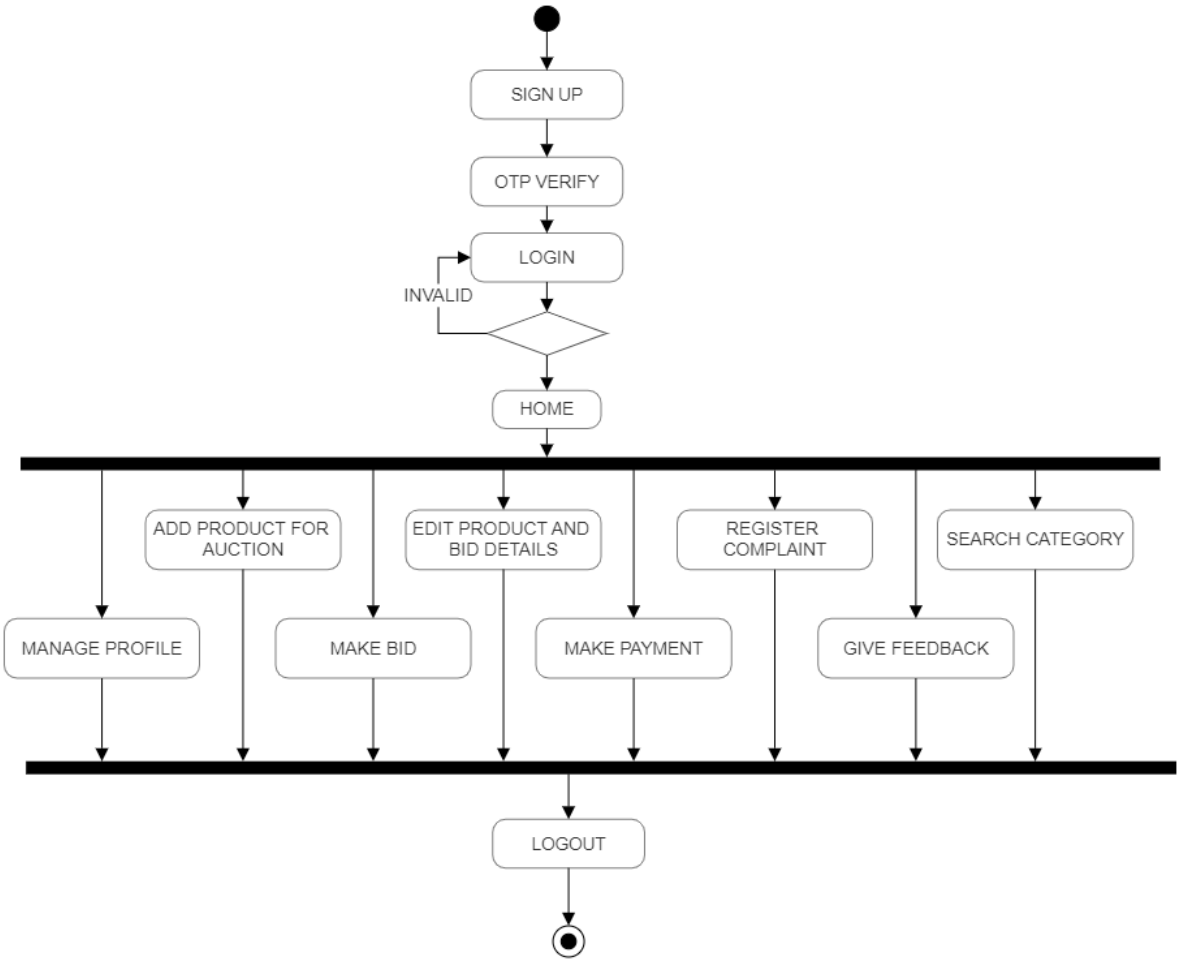
Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system.

Symbol	Symbol Name	Symbol Description
	Initial State	A black circle is the standard notation for an initial state before an activity takes place.
	Activity	The activity symbols are the basic building blocks of an activity diagram and usually have a short description of the activity they represent.
	Diamond	A marker shaped like a diamond is the standard symbol for a decision. There are always at least two paths coming out of a decision and the condition text lets you know which options are mutually exclusive.
	Final State	The black circle that looks like a selected radio button is the UML symbol for the end state of an activity.
	Join	A join combines two concurrent activities back into a flow where only one activities is happening at a time.

- **ADMIN**

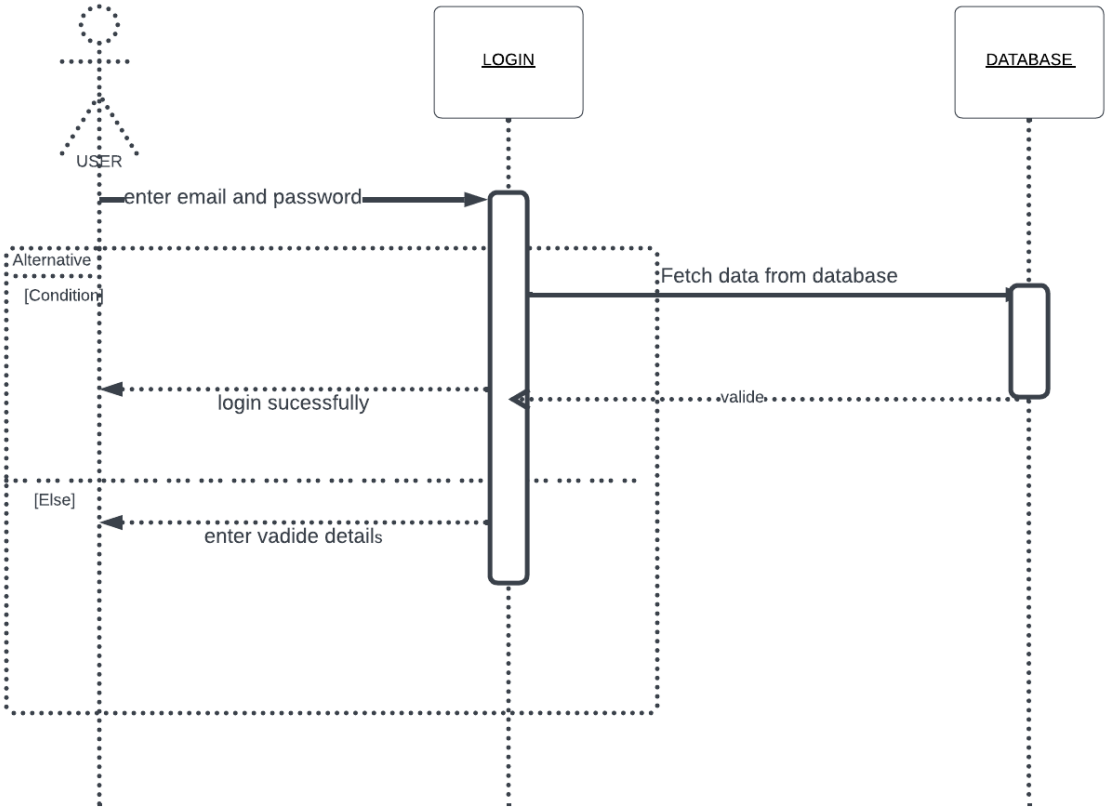

LOGIN
DASHBOARD
MANAGE CATEGORY
MANAGE COMPLAINT
VIEW AUCTION DETAILS
UPDATE LOGIN PASSWORD
MANAGE USER
LOGOUT

- **USER**

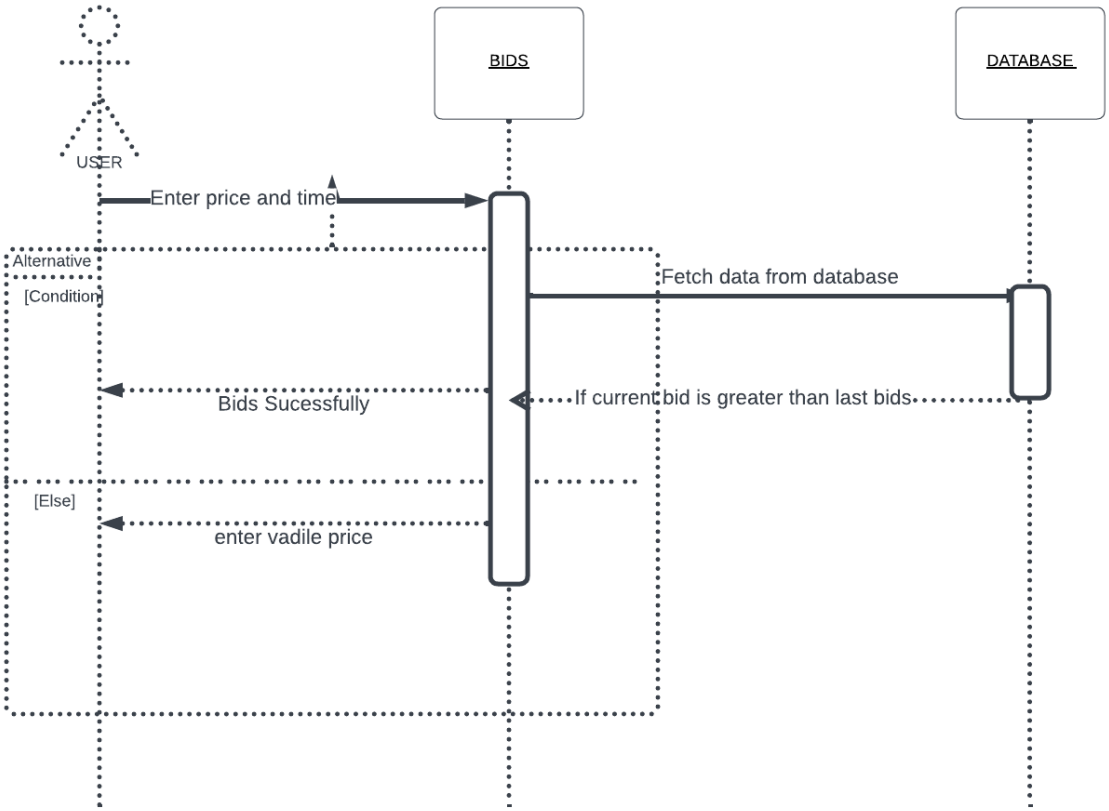

SIGN UP
OTP VERIFY
LOGIN
INVALID
HOME
ADD PRODUCT FOR AUCTION
EDIT PRODUCT AND BID DETAILS
REGISTER COMPLAINT
SEARCH CATEGORY
MANAGE PROFILE
MAKE BID
MAKE PAYMENT
GIVE FEEDBACK
LOGOUT

5.2.3 Sequence Diagram

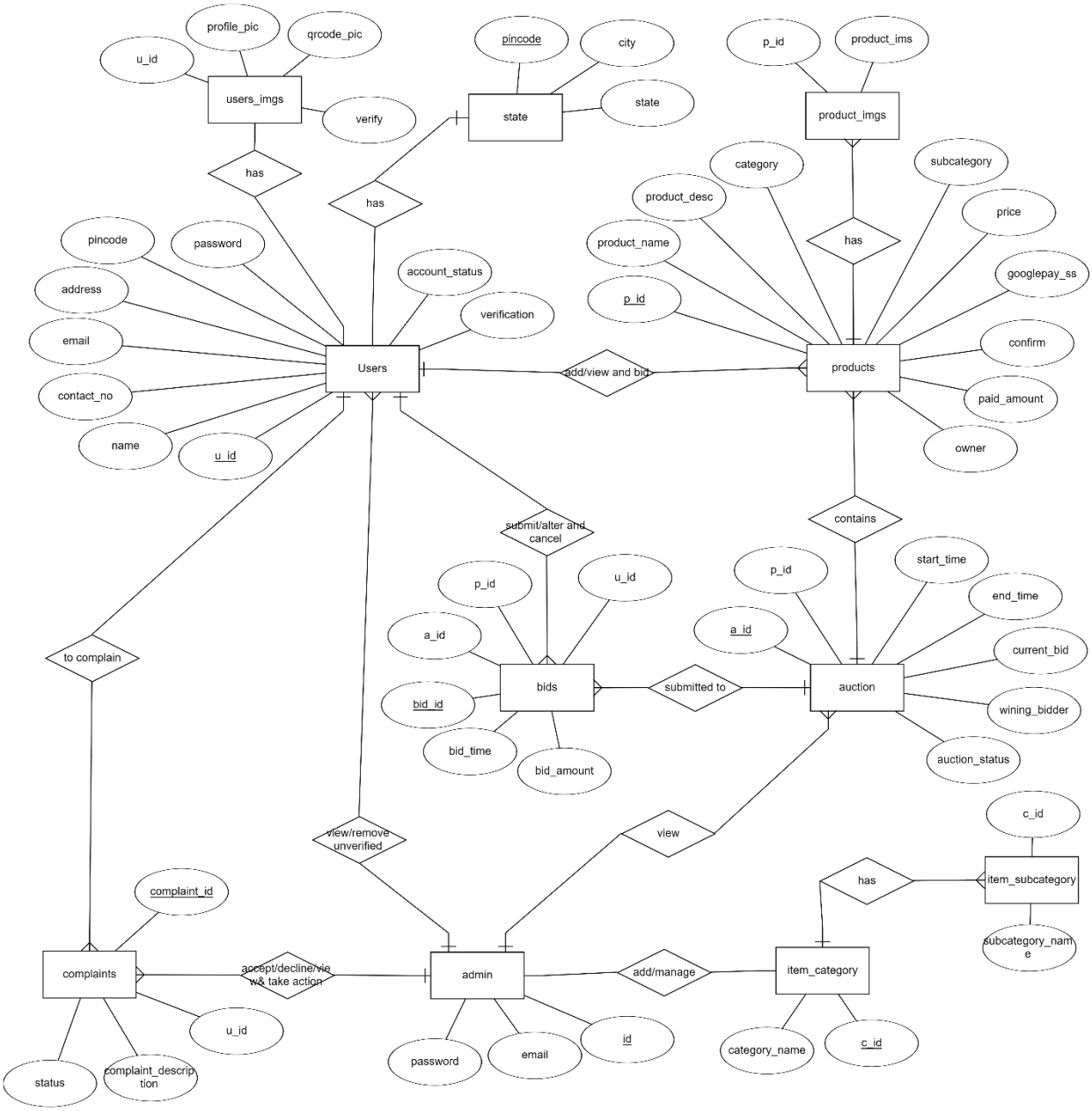
- **Login**



❖ **Auction OR Bid**



5.3 ER Diagram



6. SOFTWARE DESIGN

6.1 Database Design

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model.

Users			
NO.	FIELD NAME	DATA TYPE	CONSTRAINTS
1	U_ID	INT (8)	PRIMARY KEY
2	NAME	VARCHAR (50)	
3	CONTACT_NO	BIGINT (10)	
4	EMAIL	VARCHAR (50)	
5	ADDRESS	TEXT	
6	PINCODE	INT (8)	
7	PASSWORD	VARCHAR (300)	
8	ACCOUNT_STATUS	ENUM	
9	OTP	VARCHAR (100)	
10	VERIFICATION	ENUM	

Users_imgs			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	U_ID	INT (8)	FOREING KEY
2	PROFILE_PIC	LONGBLOB	
3	QRCODE_PIC	LONGBLOB	
4	VERIFY	ENUM	

State			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	PINCODE	INT (8)	PRIMARY KEY
2	CITY	VARCHAR (40)	
3	STATE	VARCHAR (40)	

Admin			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	ID	INT (4)	PRIMARY KEY
2	EMAIL	VARCHAR (40)	
3	PASSWORD	VARCHAR (40)	

Item_Category			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	C_ID	INT (4)	PRIMARY KEY
2	CATEGORY_NAME	VARCHAR (50)	

Item_Subcategory			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	C_ID	INT (4)	FOREING KEY
2	SUBCATEGORY_NAME	VARCHAR (20)	

Products			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	P_ID	VARCHAR (40)	PRIMARY KEY
2	PRODUCT_NAME	VARCHAR (50)	
3	PRODUCT_DESC	TEXT	
4	CATEGORY	INT (8)	
5	SUBCATEGORY	VARCHAR (60)	
6	PRICE	INT (10)	
7	GOOGLEPAY_SS	LONGBLOB	
8	CONFIRM	ENUM	
9	PAID_AMOUNT	INT (10)	
10	OWNER	INT (8)	

Product_imgs			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	P_ID	INT (8)	FOREING KEY
2	P_IMG	LONGBLOB	

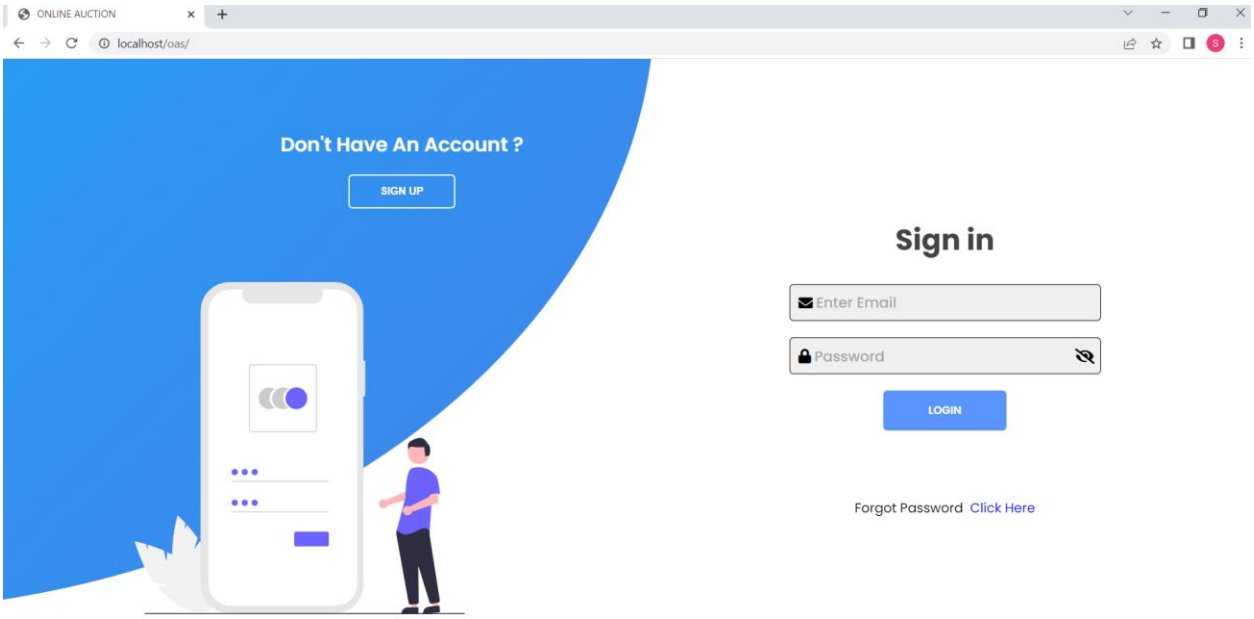
Auction			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	A_ID	INT (8)	PRIMARY KEY
2	P_ID	INT (8)	FOREING KEY
3	START_TIME	DATETIME	
4	END_TIME	DATETIME	
5	CURRENT_BID	INT (10)	
6	WINNING_BIDDER	INT (8)	FOREING KEY
7	AUCTION_STATUS	ENUM	

Bids			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	BID_ID	INT (8)	PRIMARY KEY
2	A_ID	INT (8)	FOREING KEY
3	P_ID	VARCHAR (40)	FOREING KEY
4	U_ID	INT (8)	FOREING KEY
5	BID_AMOUNT	INT (10)	
6	BID_TIME	DATETIME	

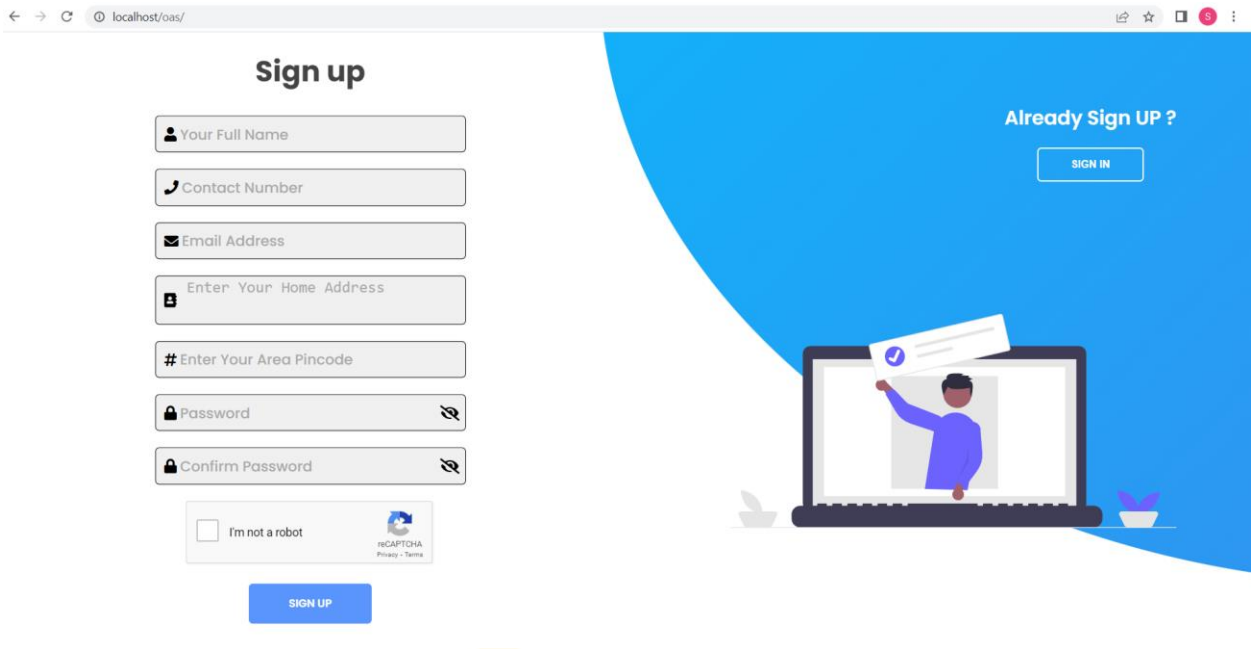
Complaints			
NO.	FIELD NAME	DATATYPE	CONSTRAINTS
1	COMPLAINT_ID	INT (8)	PRIMARY KEY
2	U_ID	INT (8)	FOREING KEY
3	COMPLAINT_DESC	TEXT	
4	STATUS	ENUM	

6.2 Interface Design

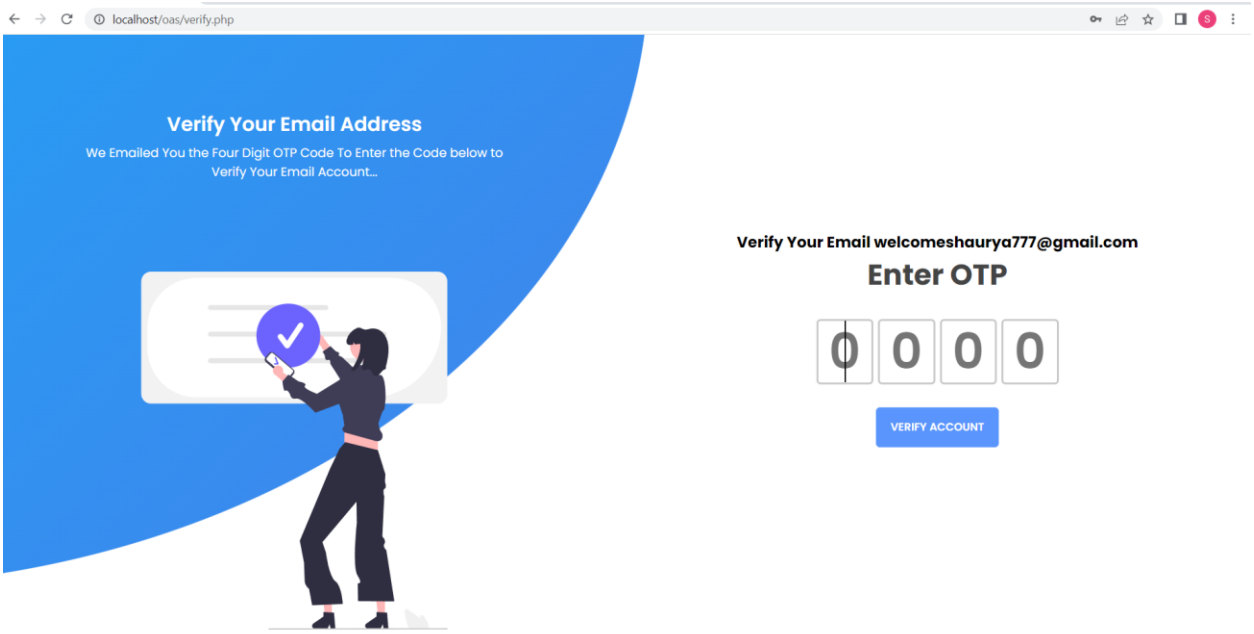
Sign in Page:



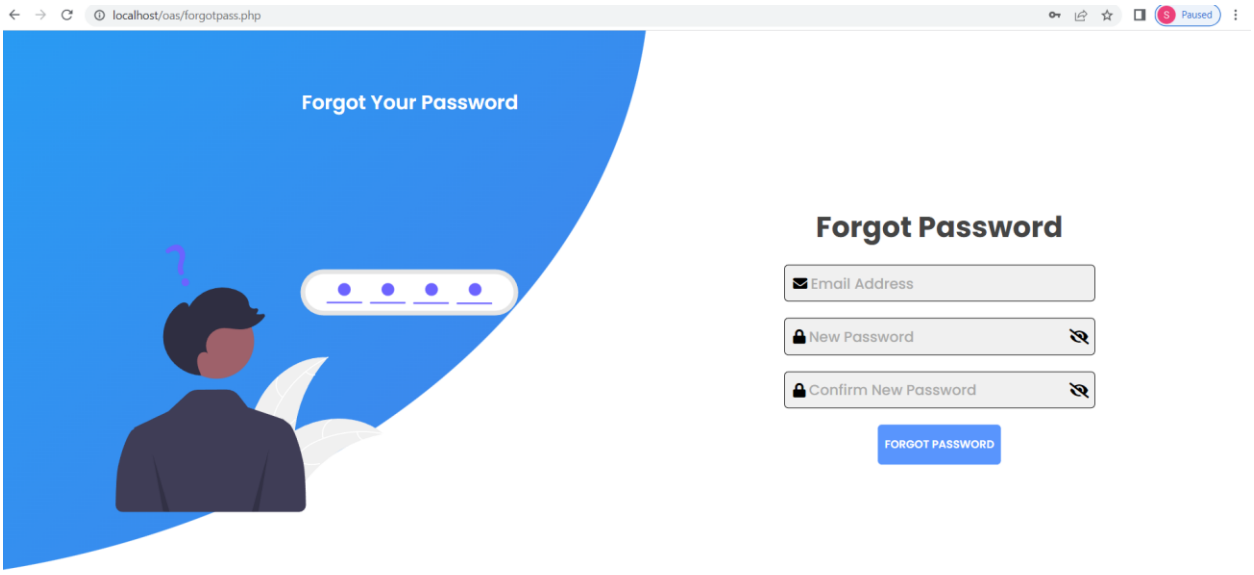
Sign Up page:



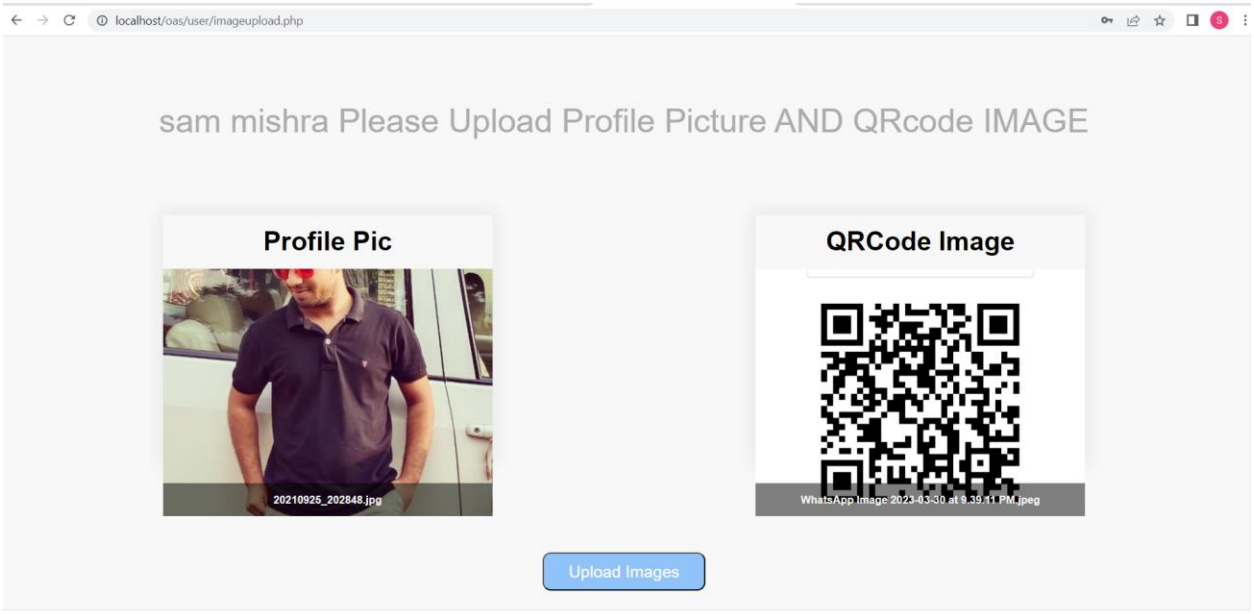
Verify OTP Page:



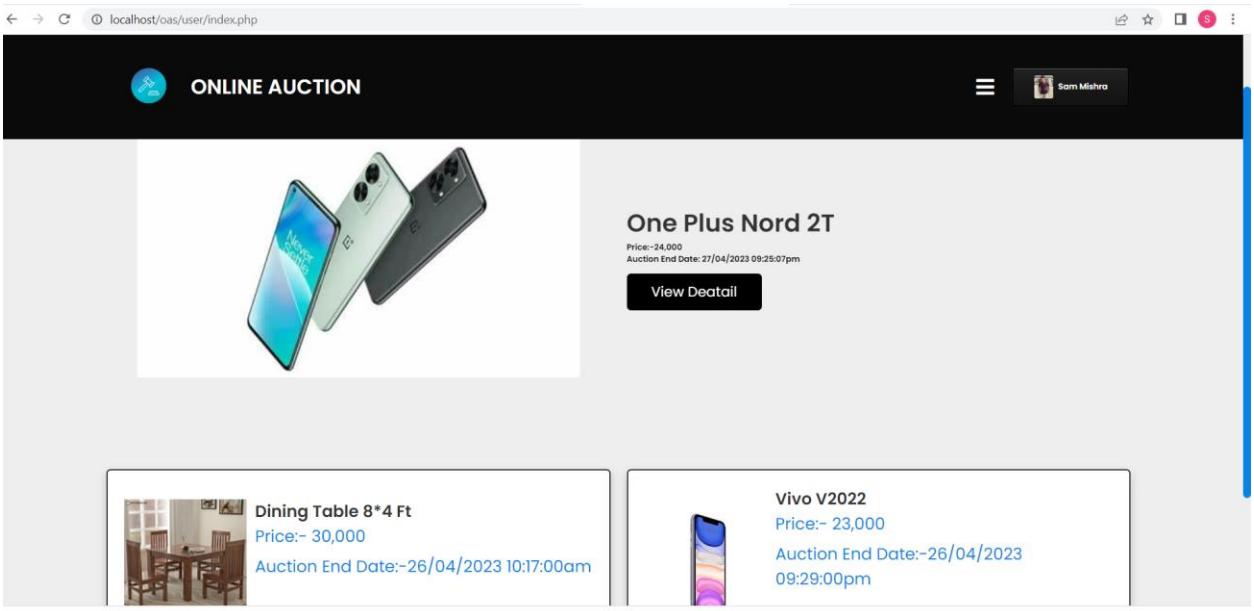
Forgot password page:



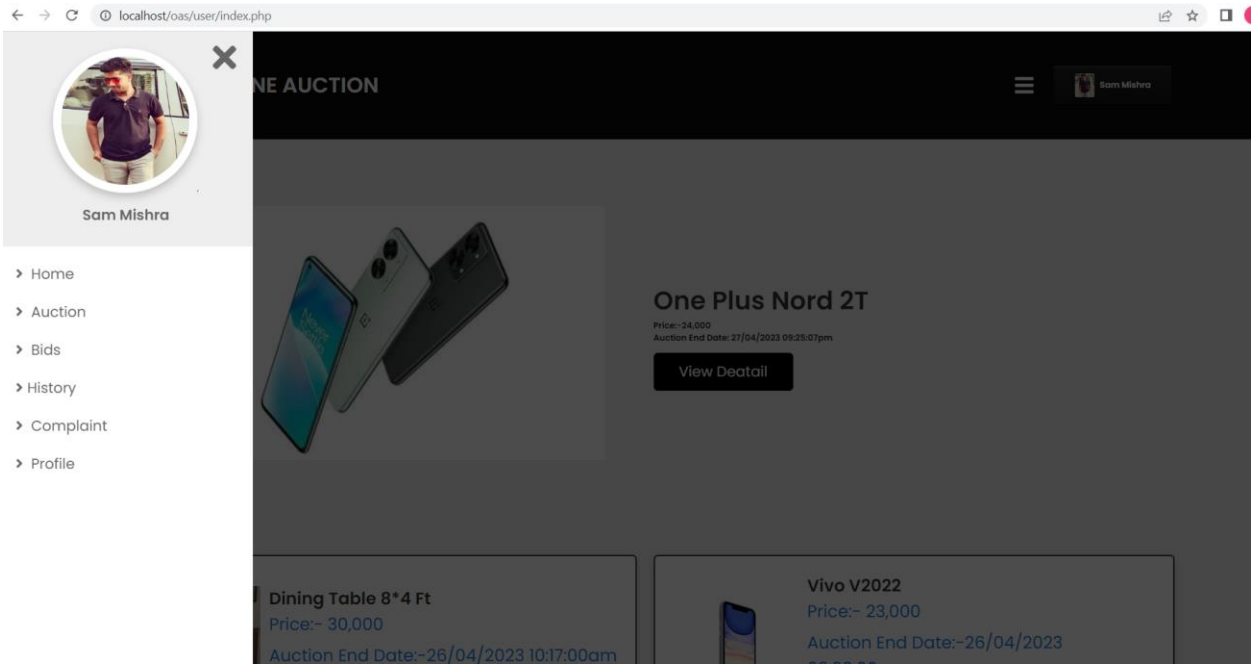
Imgupload page:



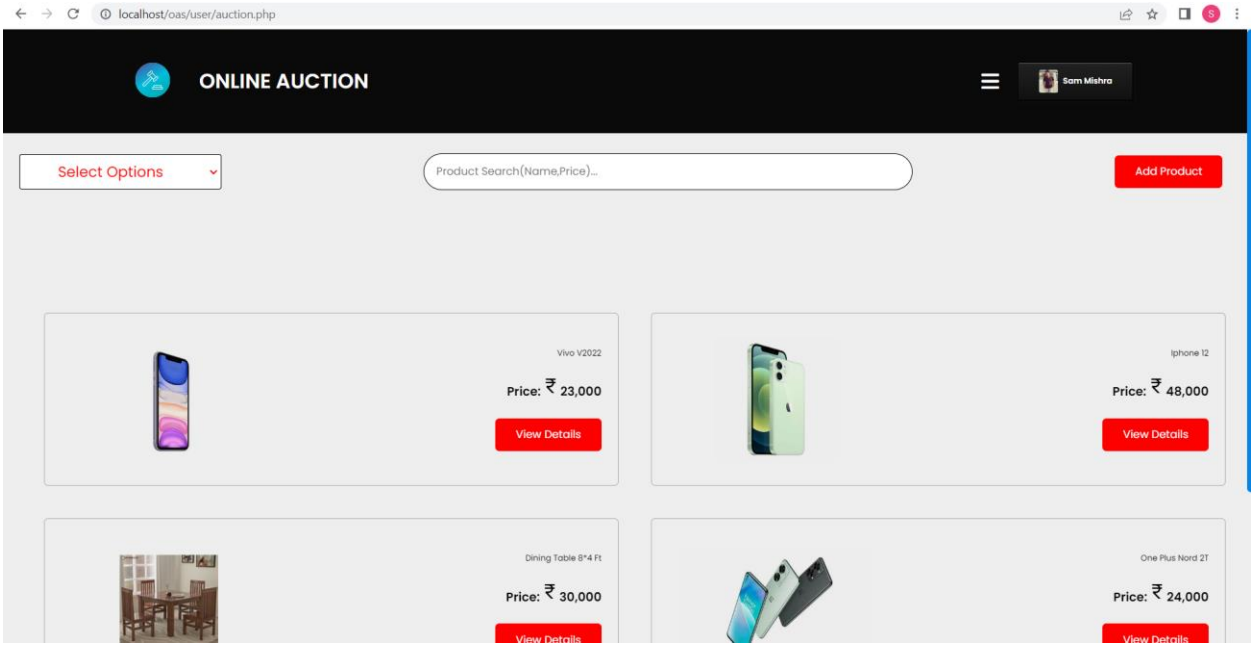
User Interface Page:



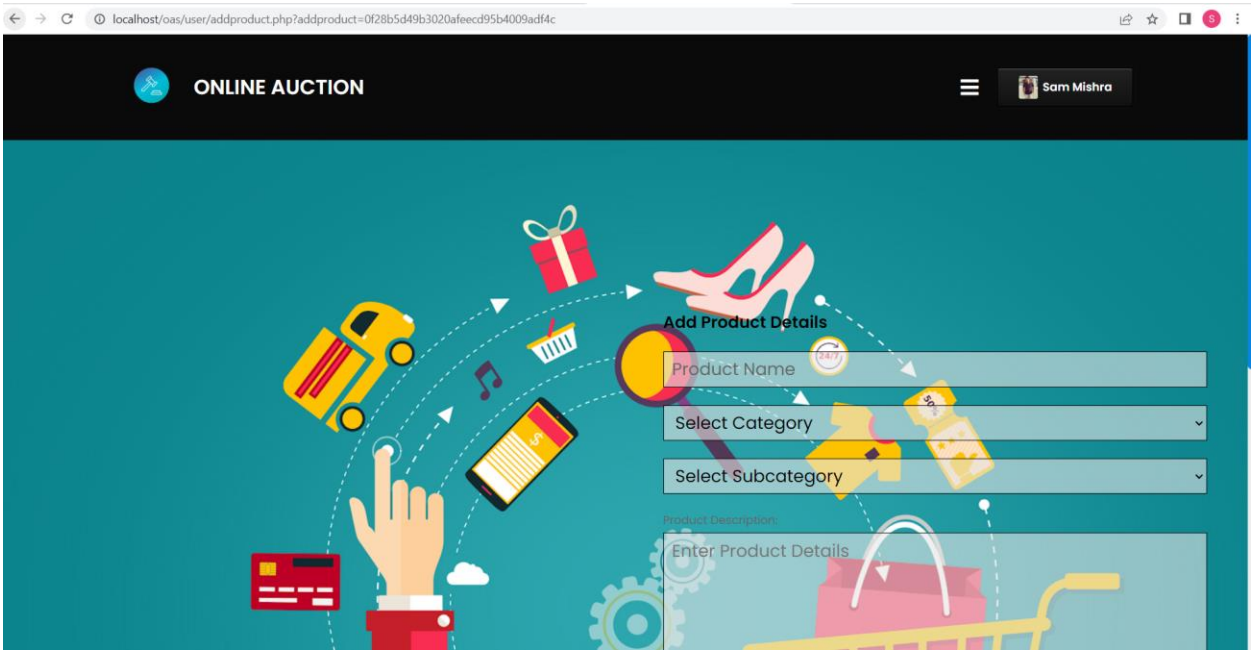
Sidebar:



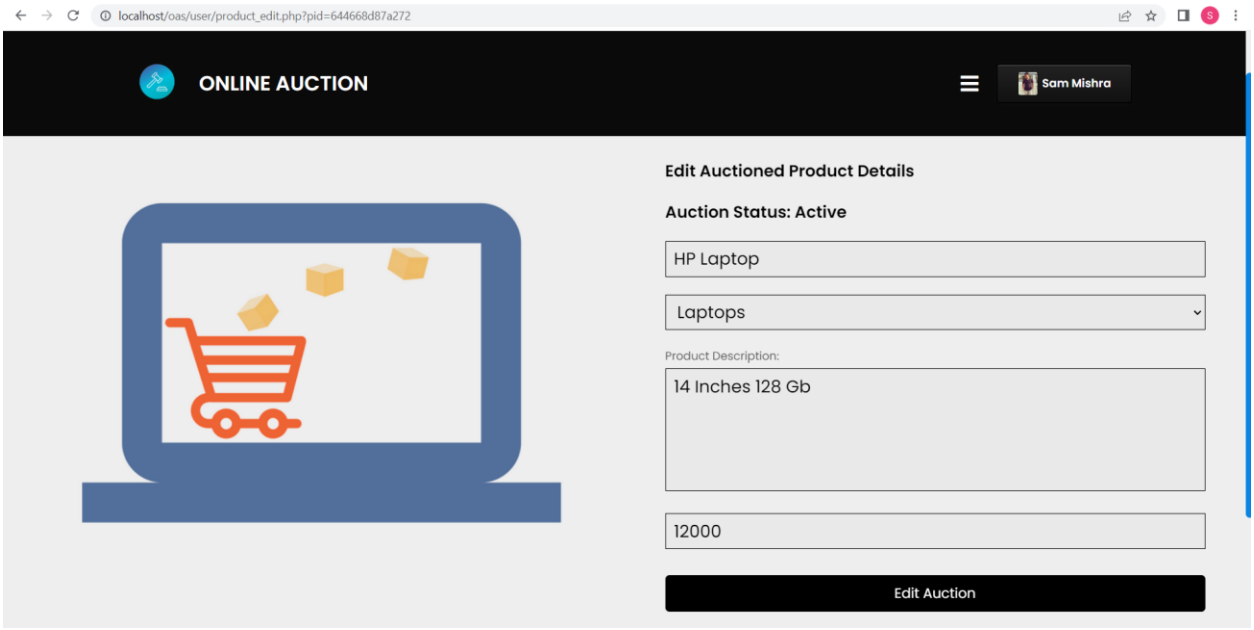
Auction Page:



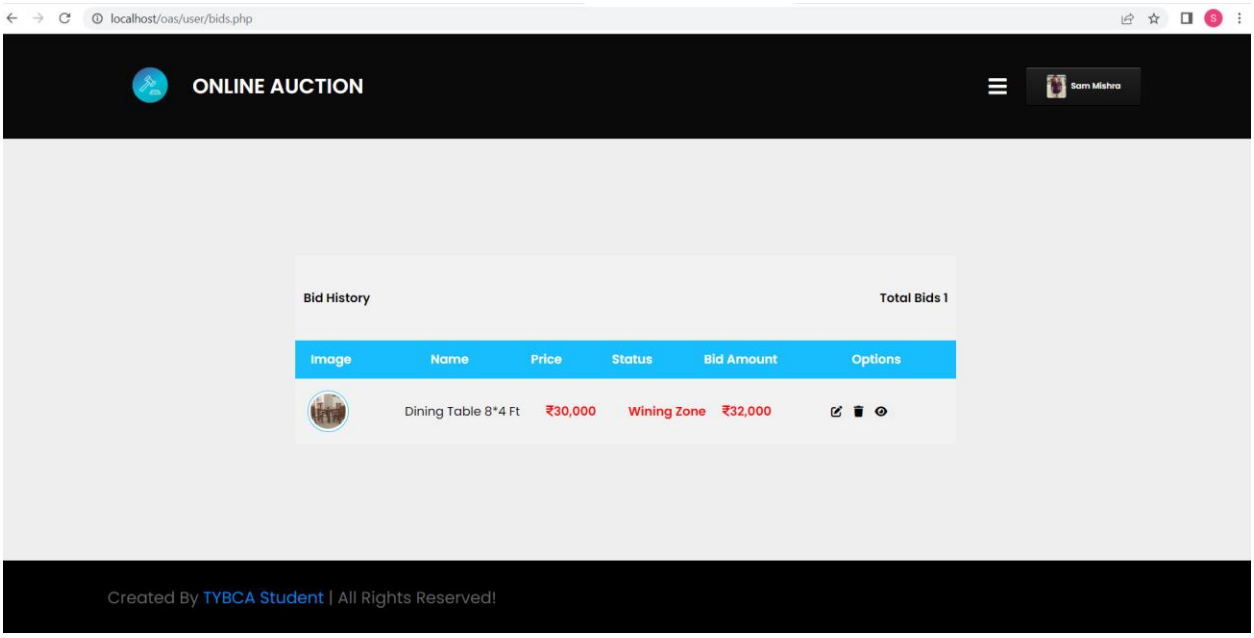
Add Product Page:



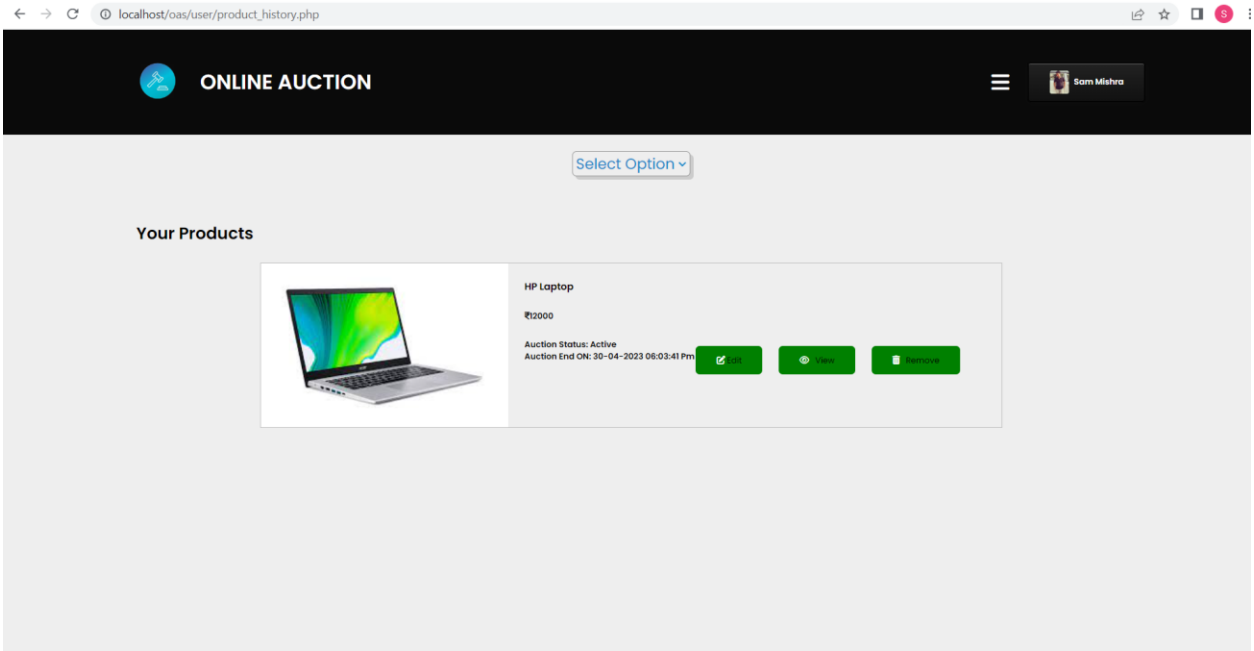
Edit Product:



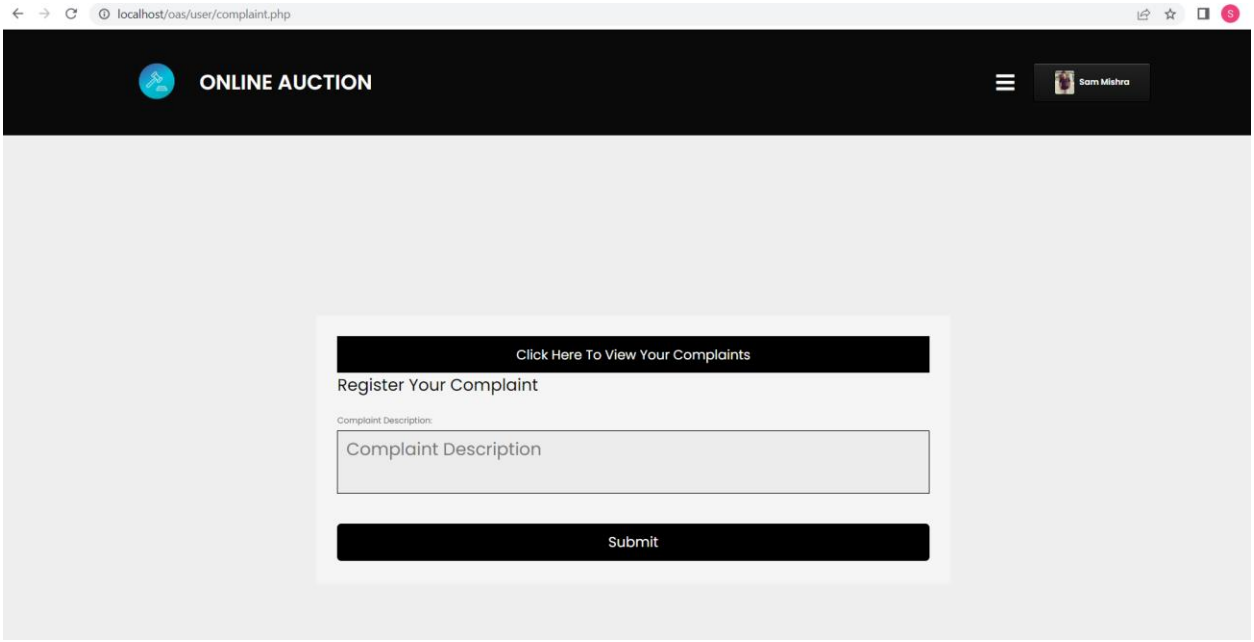
Bid Page:



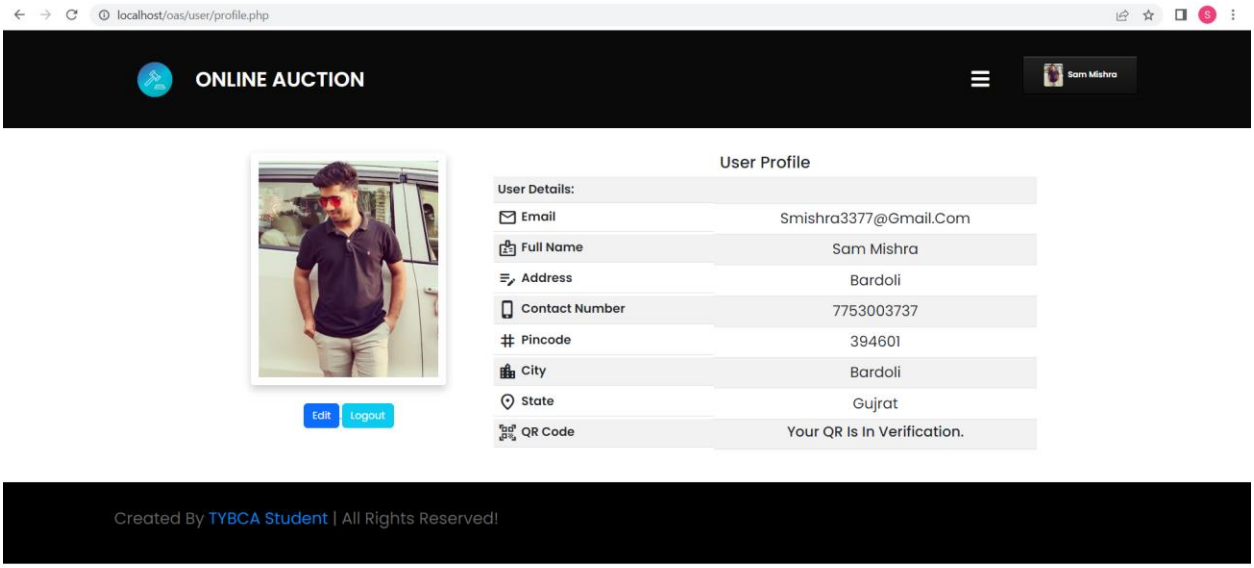
History Page:



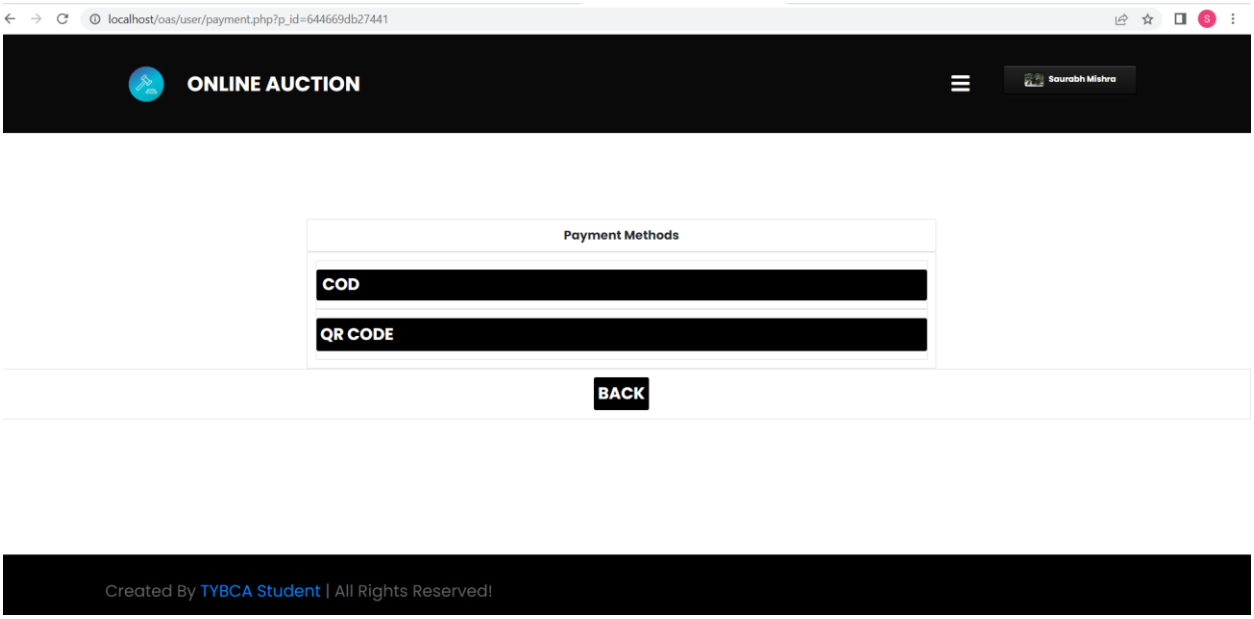
Complaint Page:



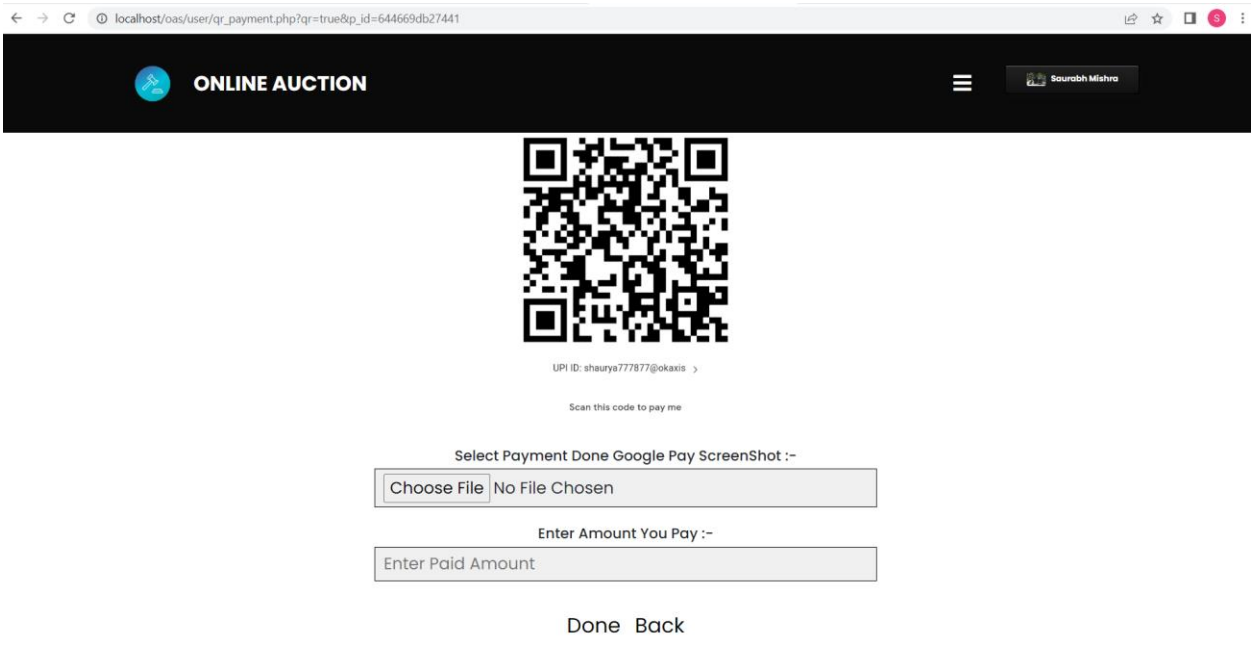
Profile Page:



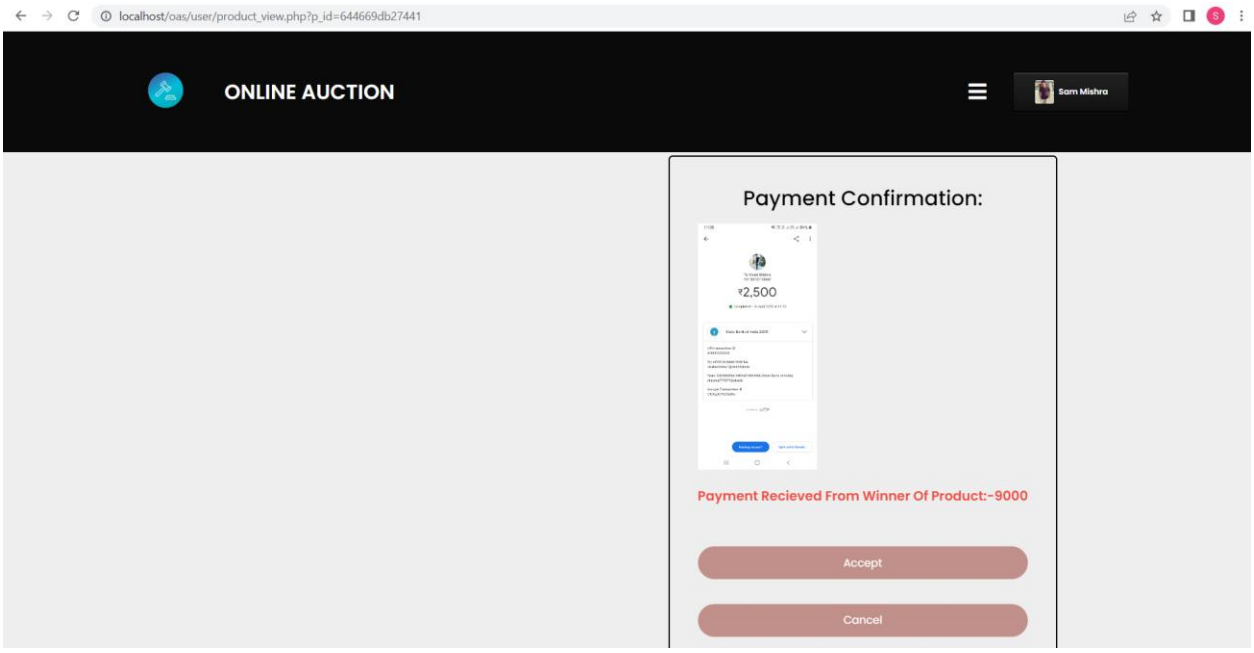
Payment Page:



QR Payment Page:

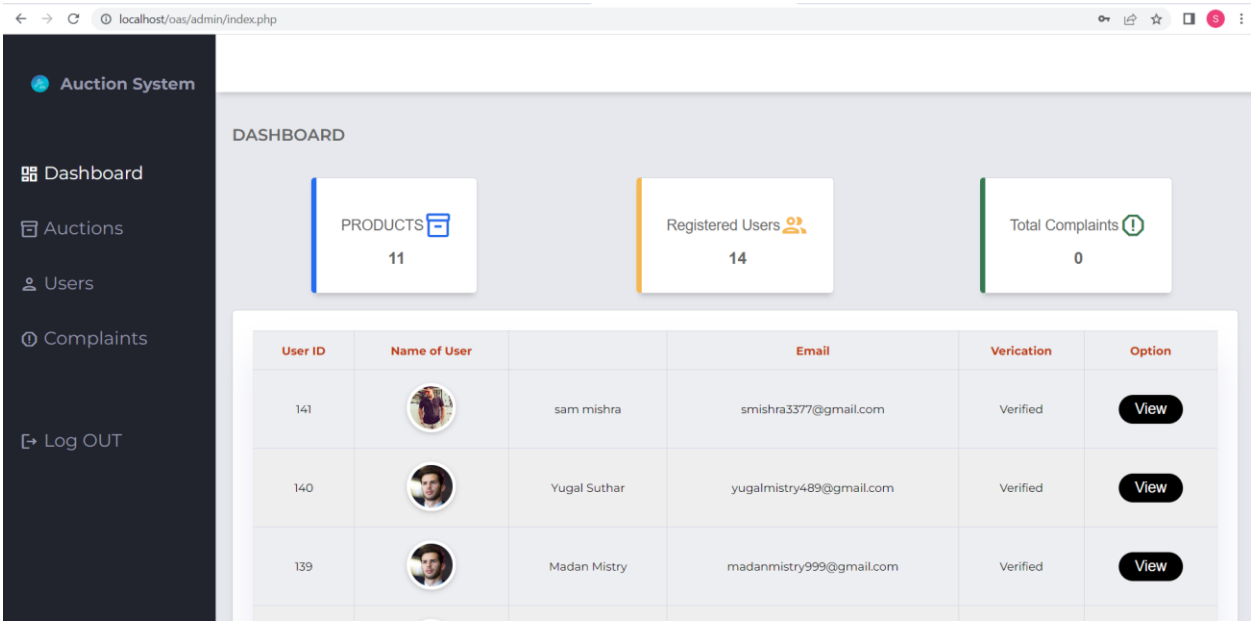


Payment Confirmation:

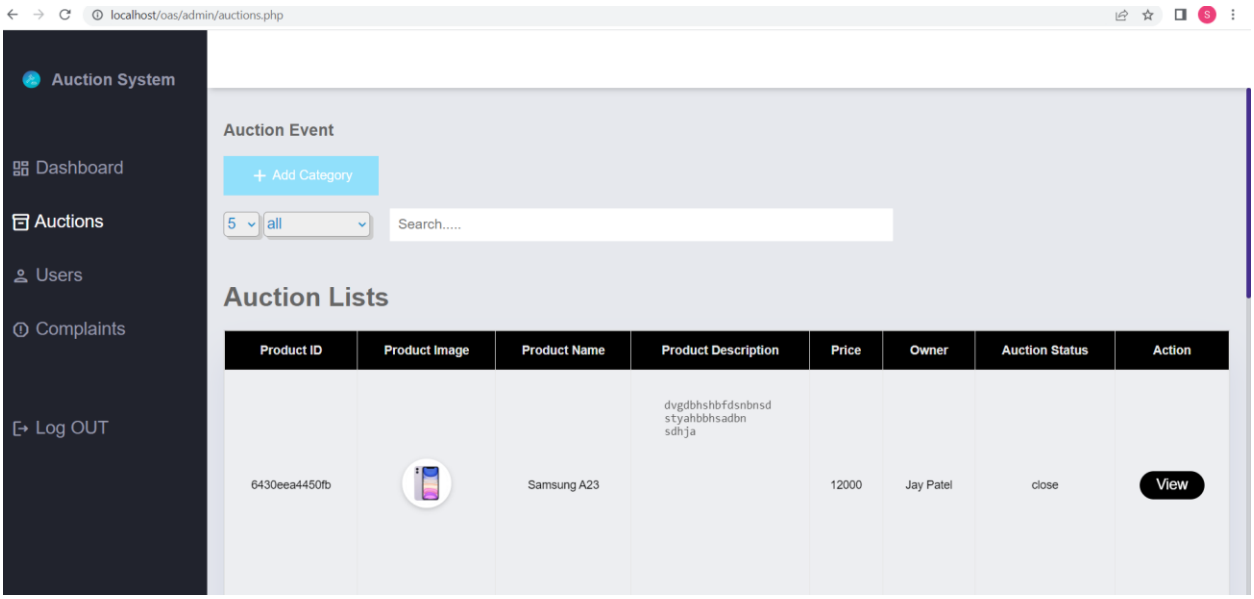


ADMIN SIDE

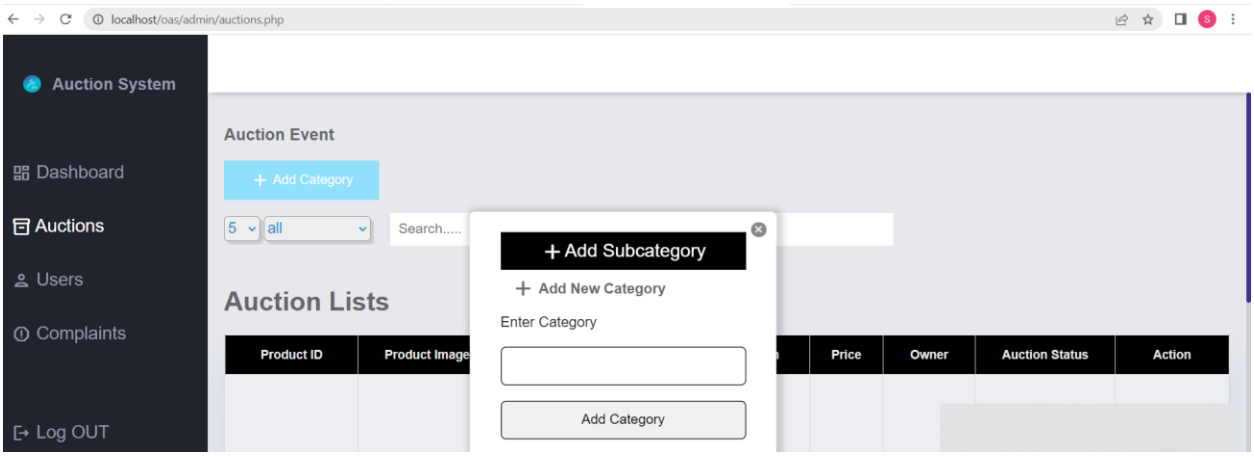
Admin Home Page:



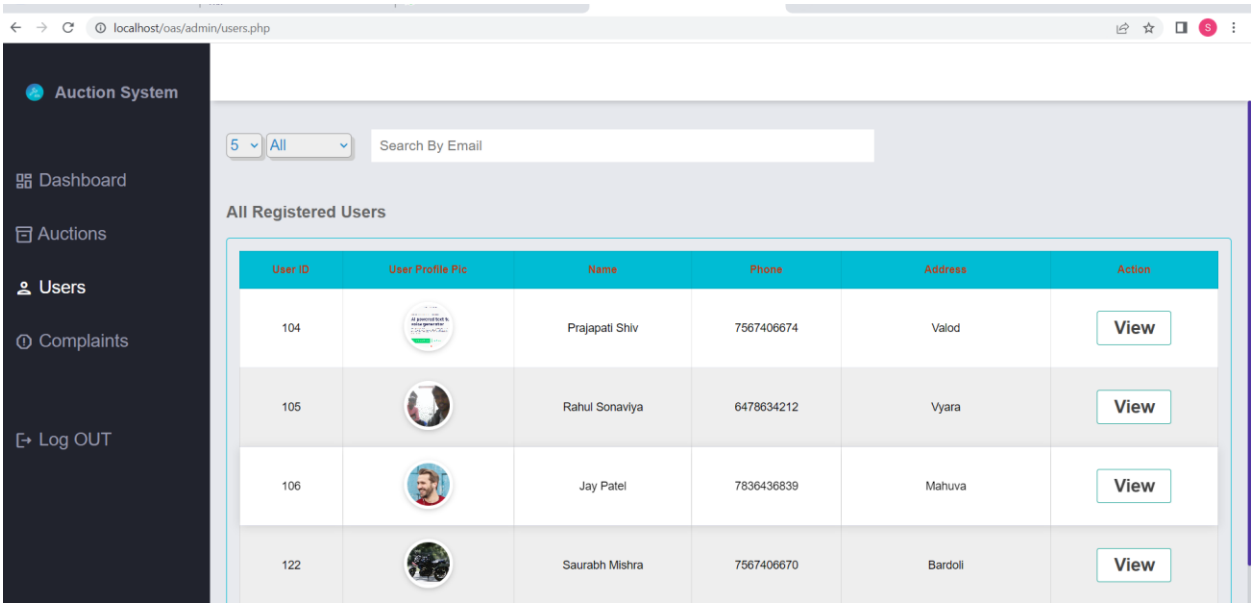
Admin Auction Page:



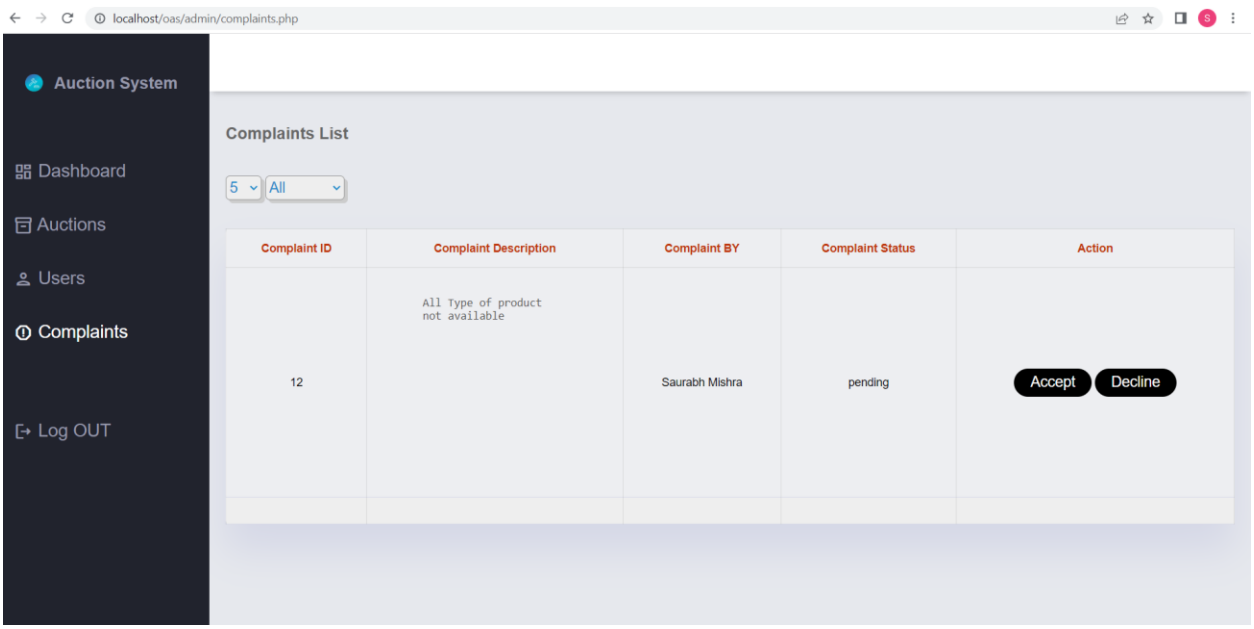
Add Category page:



Admin User Page:



Complaint Page:



7. Testing

7.1 Unit testing

- Unit testing means the verification and validation of software. In unit testing (is the smallest part of a module); programmer tests each unit of source code for fit to use.
- Unit testing focuses on verifying the effort on the smallest unit of software-module.
- The local data structure is examined to ensure that the date stored temporarily maintains its integrity during all steps in the algorithm's execution.
- Unit Tested area to ensure that the module operates properly at boundaries established to limit.

7.2 Integration Testing

- Integration testing takes input as module that means, it is nothing but the set of units tested module. It collects them in bigger aggregates and implements tests explained in an integration test plan. Afterwards, distribute as its outcomes to the integrated system which is prepared for system testing.
- In the Integration testing (Modules) the data can be tested across an interface.
- Conducting tests to uncover errors associated with interring while integration testing is a technique for constructing a program structure

8. FUTURE ENHANCEMENT

- Delivery Module Implementation
- Location wise searching of product
- We will add payment gateway system.

9. REFERENCES

❖ Bibliography: -

- Fundamentals of Software Engineering (Rajiv Mall)
- Software Engineering: A Practitioners Approach (Roger S Pressman)
- Beginning PHP and MySQL, 4th Edition, Apress Publication

❖ Webography: -

- During the Development of Project, we have referred following Websites and Youtube videos.
- Website and Youtube links:
  - https://www.w3schools.com
  - https://www.tutorialspoint.com
  - https://www.geeksforgeeks.org
  - https://stackoverflow.com
  - https://www.phptutorial.net
  - https://www.youtube.com/watch?v=yo0icoDLQQs
  - https://www.youtube.com/watch?v=NJVJRFF-Y6U
  - https://www.youtube.com/watch?v=gortnFd7hgU
  - https://www.youtube.com/watch?v=pPITBtE45bg
  - https://www.youtube.com/watch?v=KbeKkmk5nhY