|  |
| --- |
| **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  **BELAGAVI, KARNATAKA**    **Mini Project Report**  **On**  **“CONCERT MANAGEMENT”**  **Submitted to**  **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  In partial fulfillment of the requirements for the award of degree of  **BACHELOR OF ENGINEERING**  **IN**  **COMPUTER SCIENCE ENGINEERING**  **PROJECT ASSOCIATES**  **SAKSHI K-3VC21CS154**  **B H HARSHITHA-3VC21CS018**  **Under the Guidance of**  **Mrs.APARNA K S**  **Assistant Professor , Dept of CSE**    **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  **RAO BAHADUR Y.MAHABALESHWARAPPA COLLEGE OF ENGINEERING**  **(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI & APPROVED**  **BY AICTE & ACCREDITED BY NBA, NEW DELHI)**  **BALLARI - 583104, KARNATAKA**  **2022-2023** |

|  |
| --- |
| VEERASHAIVA VIDYAVARDHAKA SANGHA’S  **RAO BAHADUR Y MAHABALESHWARAPPA ENGINEERING COLLEGE**  (AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI  APPROVED BY AICTE & ACCREDITED BY NBA, NEW DELHI) BALLARI - 583104, KARNATAKA  **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**    **C E R T I F I C A T E**  Certified that the Mini Project work entitled **“CONCERT MANAGEMENT”** carried out by **SAKSHI K (3VC21CS154), B H HARSHITHA (3VC21CS018)** are bonafide students of **Rao Bahadur Y.Mahabaleswarappa College of Engineering** in partial fulfillment for the award of **Bachelor of Engineering in Computer Science Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2023-2024**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said Degree.  Signature of the Project Guide Signature of the HOD  **Mrs. APARNA K S Dr. H. GIRISHA**  **Assistant Professor ,Dept of CSE HOD, Dept of CSE**  **External viva**  **Name of the Examiners Signature with date** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | | |
|  | **CO No** | | **Description** | | | | | | | | | | | |  | |
| **C308.1** | | Use DDL and DML commands for database creation and manipulation. | | | | | | | | | | | | | |
| **C308.2** | | Demonstrate the working of different concepts of DBMS | | | | | | | | | | | | | |
| **C308.3** | | Use SQL commands for processing the queries. | | | | | | | | | | | | | |
| **C308.4** | | Implement, analyse and evaluate the project developed for an application | | | | | | | | | | | | | |
| **CO** | **Program Outcome** | | | | | | | | | | | | | **Specific**  **Outcome** | |
| **1** | | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **S1** | **S2** |
| **1** | **2** | | **3** | **3** |  | **3** | **2** |  |  |  |  | **2** |  |  | **2** |
| **2** | **2** | | **2** |  |  | **3** | **2** |  |  |  |  | **2** |  |  | **2** |
| **3** | **2** | | **2** | **3** | **2** | **3** | **2** |  |  | **3** | **2** | **2** |  |  | **2** |
| **AVG** | **2** | | **2.3** | **3** | **2** | **3** | **2** |  |  | **3** | **2** | **2** |  |  | **2** |
|  | | | | | | | | | | | | | | | |

**ACKNOWLEDGEMENT**

After the completion of our “**INVENTORY MANAGEMENT SYSYTEM**”. We learnt many things such as the basics of web development and the database management system along with the lines of team work, coordination and commitment. The successful completion of our Mini Project work, **would** be incomplete if we did not mention the people, whose guidance, encouragement, noble gesture, affection and support crowned our efforts with success. It is our privilege to express our gratitude and respect to all those who inspired and helped me in the completion of our project. All the expertise in this project belongs to those listed below.

We are very much indebted to **Assistant Prof. Aparna K S** our respected guides, for their noble gesture, support, coordination, valuable suggestions and guidance given to us in completing the project.

We also thank **Dr. Girish H,** H.O.D., C.S.E. Department, for extending all his valuable support and encouragement.

We express our sincere thanks to our beloved Principal **Dr. T. Hanumanth Reddy**,

R.Y.M.E.C**,** Ballari, for giving us an opportunity to carry out our academic project.

We also thank all the teaching and non-teaching staff of all the departments of R.Y.M.E.C., for giving their kind cooperation during development of our mini project.

**Project associates:**

**BHARGAVI U(3VC20CS036)**

**DIVYA SHREE(3VC20CS044)**

**Table of Contents:**

|  |  |  |
| --- | --- | --- |
| **Serial no.** | **Chapter** | **Page no** |
| 1 | Introduction | 3 |
| 2 | Software Requirement | 4 |
| 3 | Design | 6 |
| 4 | Implementation | 8 |
| 5 | Testing | 9 |
| 6 | Tables | 11 |
| 7 | Screen Shots | 15 |
| 8 | Conclusion & Future Scope | 20 |
| 9 | Bibliography | 21 |

# Chapter 1: INTRODUCTION

CONCERTS represent a vibrant intersection of artistry, entertainment, and logistical coordination. Whether it's a small local gig or a massive festival with thousands in attendance, the successful orchestration of a concert requires meticulous planning, efficient organization, and seamless execution. In the digital age, the integration of technology has revolutionized the way concerts are managed, with database-driven solutions leveraging **MySQL** and dynamic web development facilitated by **PHP** playing pivotal roles in streamlining operations.

Traditionally, concert management relied heavily on manual processes,spreadsheets, and disparate systems, leading to inefficiencies, errors, and delays. However, the advent of relational database management systems (RDBMS) like MySQL ushered in a new era of centralized data storage, enabling organizers to consolidate information related to artists, venues, tickets, and attendees in a structured and scalable manner. Coupled with the flexibility and power of server-side scripting languages like PHP, concert management systems have evolved into sophisticated platforms capable of handling the complexities inherent in organizing and promoting live events.

# Chapter 2 : SOFTWARE REQUIREMENTS

Requirement are major source to develop a project.

The software required for creating a web application for concert management using MySQL, PHP, HTML, CSS, and JavaScript include:

1. **Text Editor or Integrated Development Environment (IDE):**

Software like Visual Studio Code, Sublime Text, or PhpStorm for writing and editing code efficiently.

1. **Web Server:**

Apache, Nginx, or another web server to host the PHP files and serve web pages to users.

1. **Database Management System (DBMS):**

MySQL or MariaDB for storing and managing data related to concerts, artists, venues, tickets, and users.

1. **PHP Interpreter:**

PHP interpreter to parse and execute PHP scripts on the server-side.

1. **Web Browser:**

Chrome, Firefox, Safari, or another modern web browser for testing and debugging the web application.

For the front end, you'll need:

1. **HTML (Hypertext Markup Language):**

For creating the structure and content of web pages.

**2.** **CSS (Cascading Style Sheets):**

For styling the appearance and layout of web pages, including colors, fonts, spacing, and more.

**3. JavaScript:**

For adding interactivity and dynamic behavior to web pages, such as form validation, animations, and AJAX requests.

**Functional Requirements:**

**User Authentication and Authorization:**

Allow users to register, log in, and manage their accounts. Differentiate between administrators, artists, and attendees, with appropriate permissions for each role.

**Artist Management:**

Enable artists to create profiles, upload promotional materials (e.g., images, videos, bios), and submit requests for concert bookings.

**Venue Management:**

Provide a database of venues with details such as capacity, location, availability, and amenities. Allow administrators to manage venue information and availability.

**Concert Scheduling:**

Allow administrators to schedule concerts, assign artists and venues, set ticket prices, and manage concert details. Implement a calendar view for easy visualization of scheduled events.

**Ticketing System:**

Facilitate the sale and management of tickets for concerts. Provide options for different ticket types (e.g., general admission, VIP) and integrate with payment gateways for secure transactions.

**Audience Engagement:**

Enable attendees to browse upcoming concerts, view artist profiles, purchase tickets, and provide feedback or reviews after attending events. Implement email notifications and social media integration for promoting concerts and engaging with the audience.

**Reporting and Analytics:**

Generate reports on ticket sales, attendance, revenue, and other relevant metrics. Provide data visualization tools for analyzing trends and making informed decisions about future concerts.

**Security and Performance:**

Implement secure coding practices to prevent common vulnerabilities like SQL injection and cross-site scripting (XSS). Optimize database queries and server-side scripts for efficient performance, even under high traffic conditions.

# Chapter 3 : DESIGN

After the requirements are collected and analyzed, we create a schema for the database.

Schema Diagram: Brand table

I

d

C

ategory

id

b

name

status

Category table category id

N

am

e

S

tatus

Customer table

I

d

name

address

mobile

Balance

Order table

Order\_id Product\_id Total\_shipped Customer\_id Order\_date

Product table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pid | categoryid | grandid | pname | model | description | quatity | unit | Base\_price |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tax | | Minimum\_order | | supplier | | status | | date | |
| Purchase table | |  | |  | |  | |  | |
| Purchase\_id | | Supplier\_id | | Product\_id | | quatity | | Purchse\_id | |
| Supplier table | |  | |  | |  | |  | |
| Supplier \_id | | Supplier\_name | | mobile | | address | | status | |
| User table | |  | |  | |  | |  | |
| Userid | email | | Password | | name |  | type |  | Status |

Fig : SCHEMA DIAGRAM

Entity-Relation Diagram:

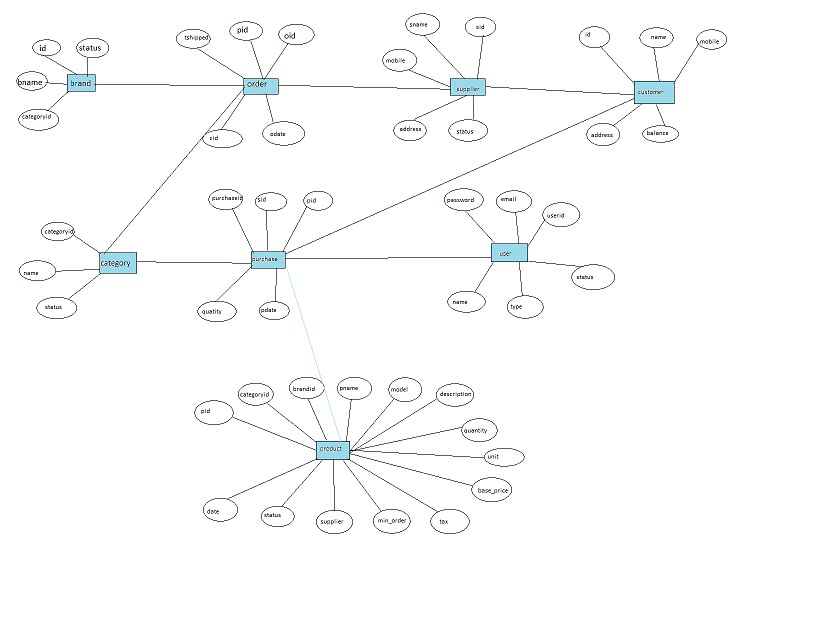


Fig : ER-DIAGRAM

# Chapter 4: IMPLEMENTATION

It is an important process of development. Consisting of the following modules:

4.1 Login:

* Administrator logs into the system with validated credentials. o The credentials of Admin includes valid username and password.
* After logging in Administrator can access the system, he/she can perform various operation.

4.2 Home Page:

o After the admin enters into the system. Home Page is displayed.

* It consists of welcome page and other information about page. The Module is used to jump to different tables.
* A disclaimer is displayed on home page. The navigation bar contains links to other modules, so that the admin can perform required operation.

4.3 Customer Details:

o This module is connected with the customer table of the database.

* It contains all the details regarding the customer or viewer. The details include name, address, gender, phone number, email

# Chapter 5: TESTING

Testing can be vital for as to incur any if all discrepancy or problems with the software and take necessary precautions and make changes as possible. Testing is vital before rolling out or implementing the software

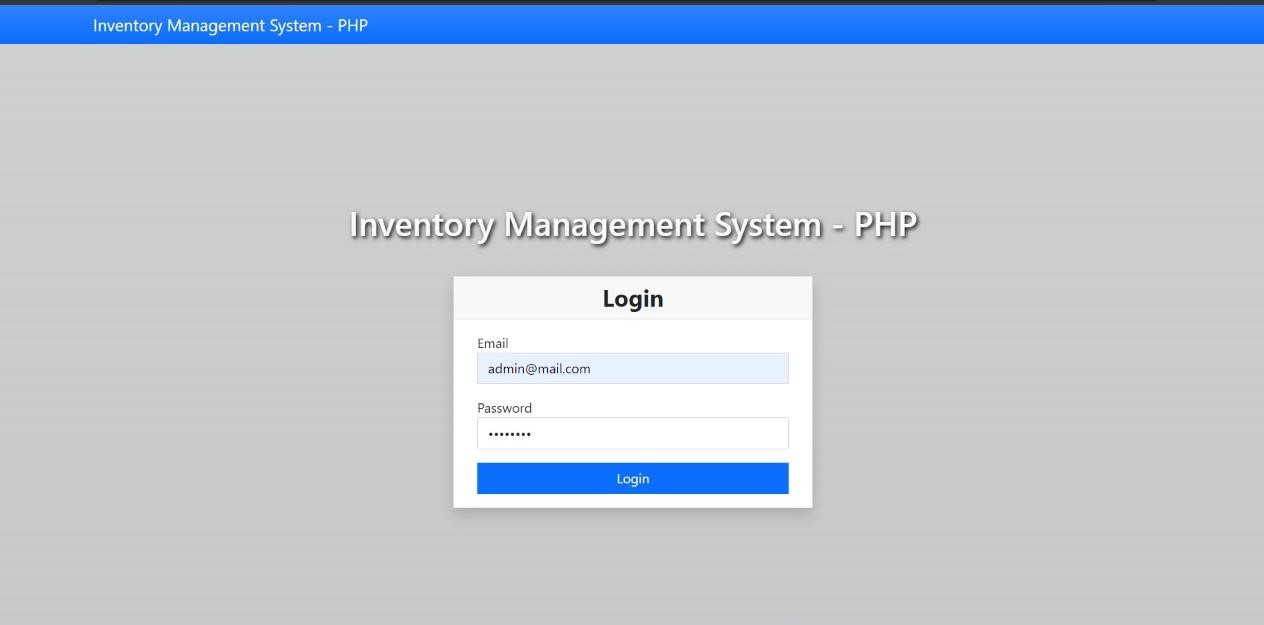
5.1 Test 1:

Case Description: Check response when valid user id and password is entered. Test data :username-admin@mail.com, password-admin123

Expected result: Login should be successful.

Actual result: Login was successful.

Test Pass/Fail: Pass.



5.2 Test 2:

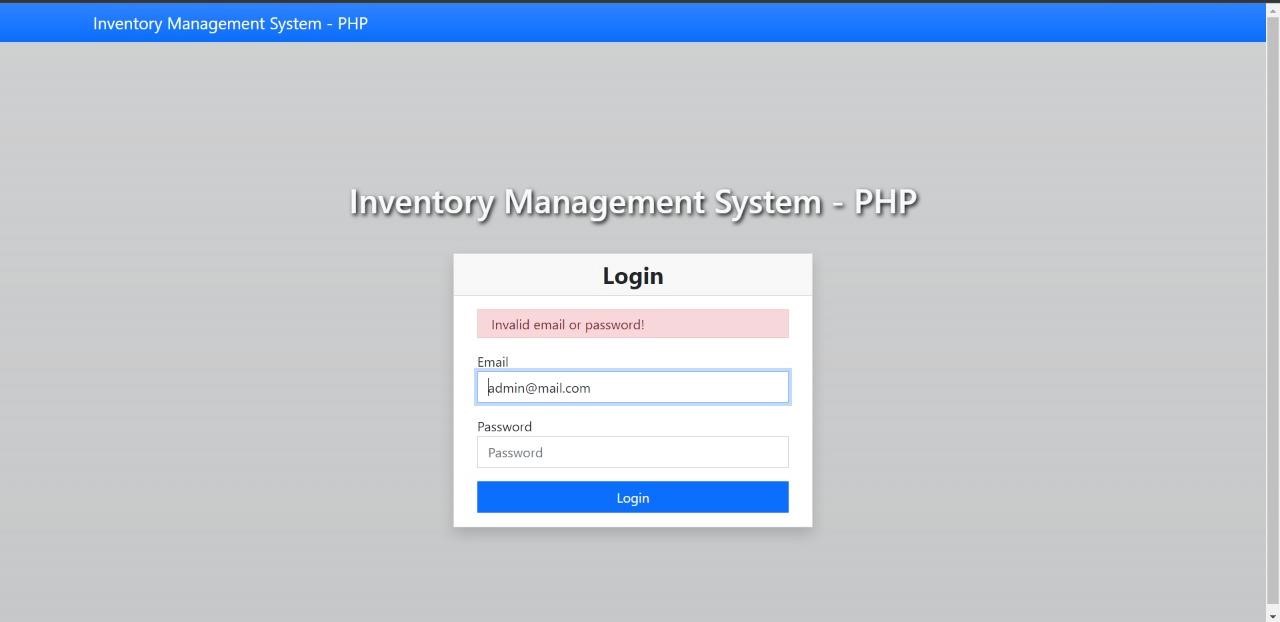
Case Description: Check response when invalid or unauthenticated user id and password is entered.

Test data: username- admin@mial.com, password-12453

Expected result: Login should be failed.

Actual result: login is failed.

Test Pass/Fail: Pass.



# Chapter 6 : TABLES

All the tables used in the project are shown here.

6.1 Brand list table:

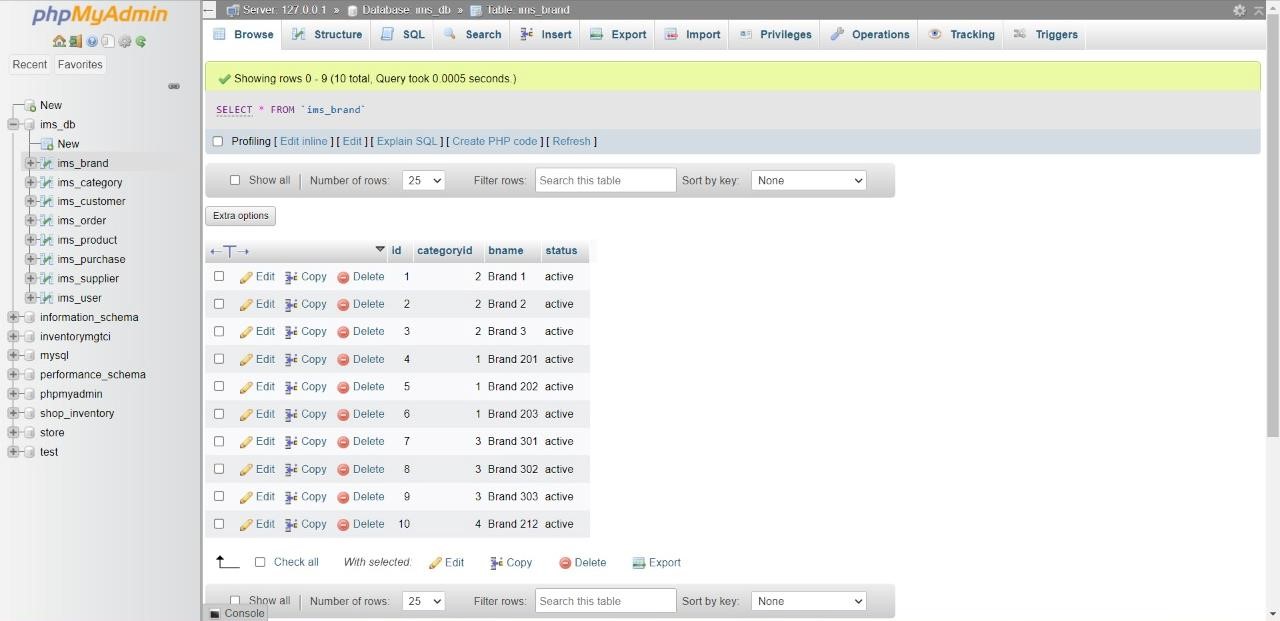
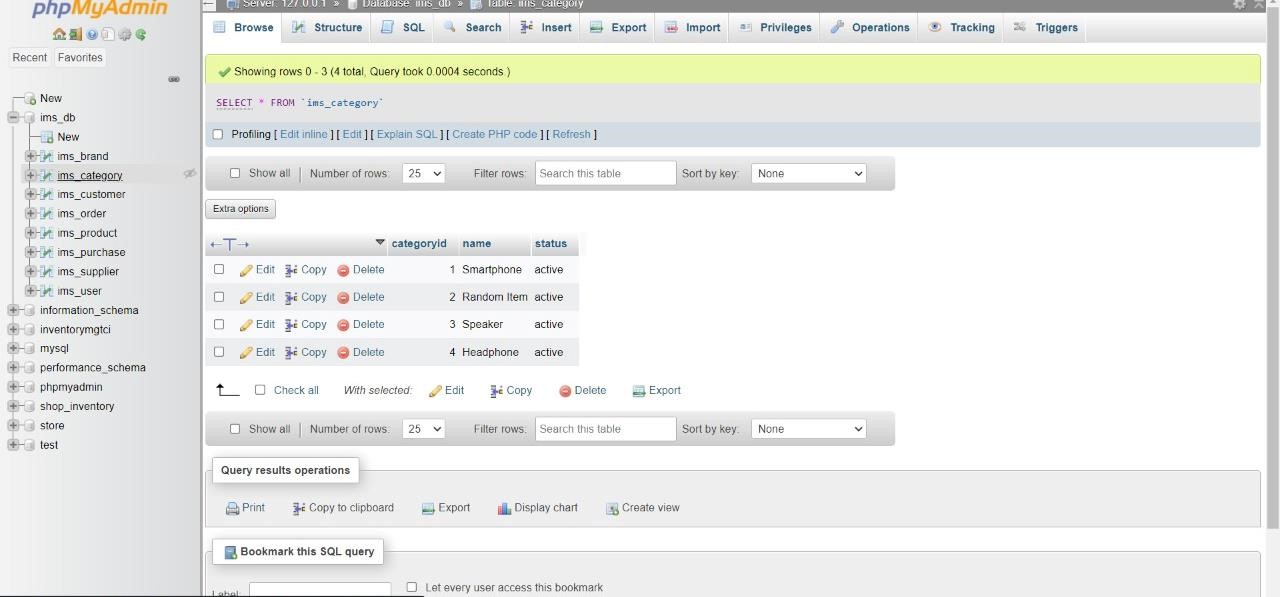


Fig : Brand list Table

6.2 Category list table

 Fig : Category list Table

6.3 Customer table:

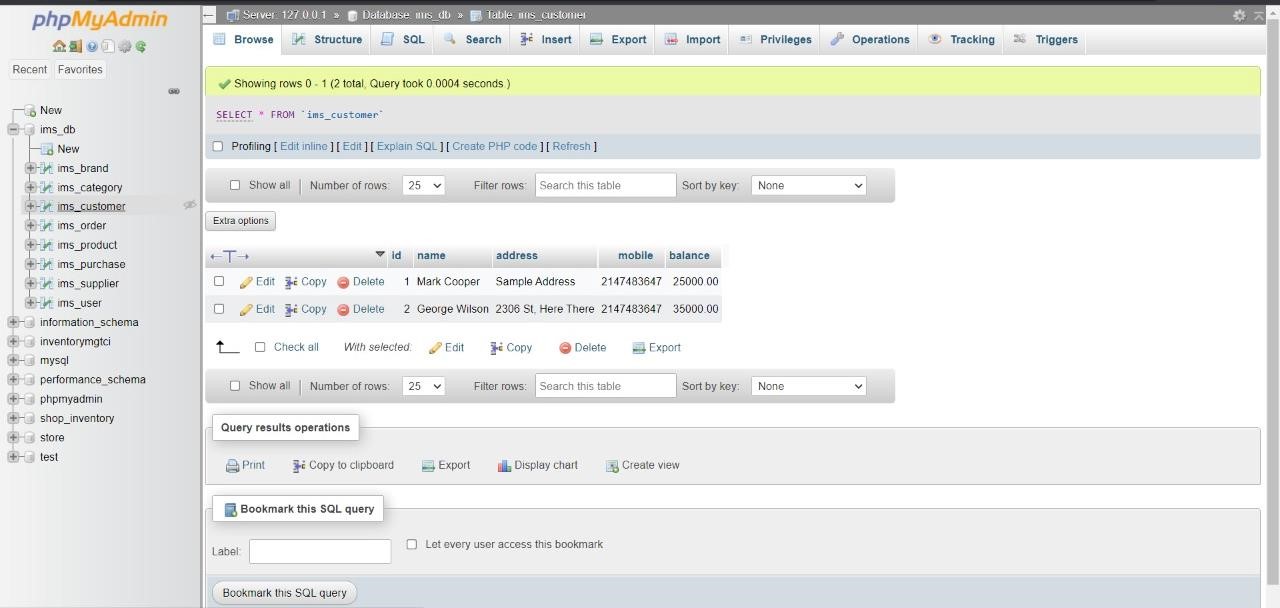


Fig : Customer Table

6.4 Order list table:

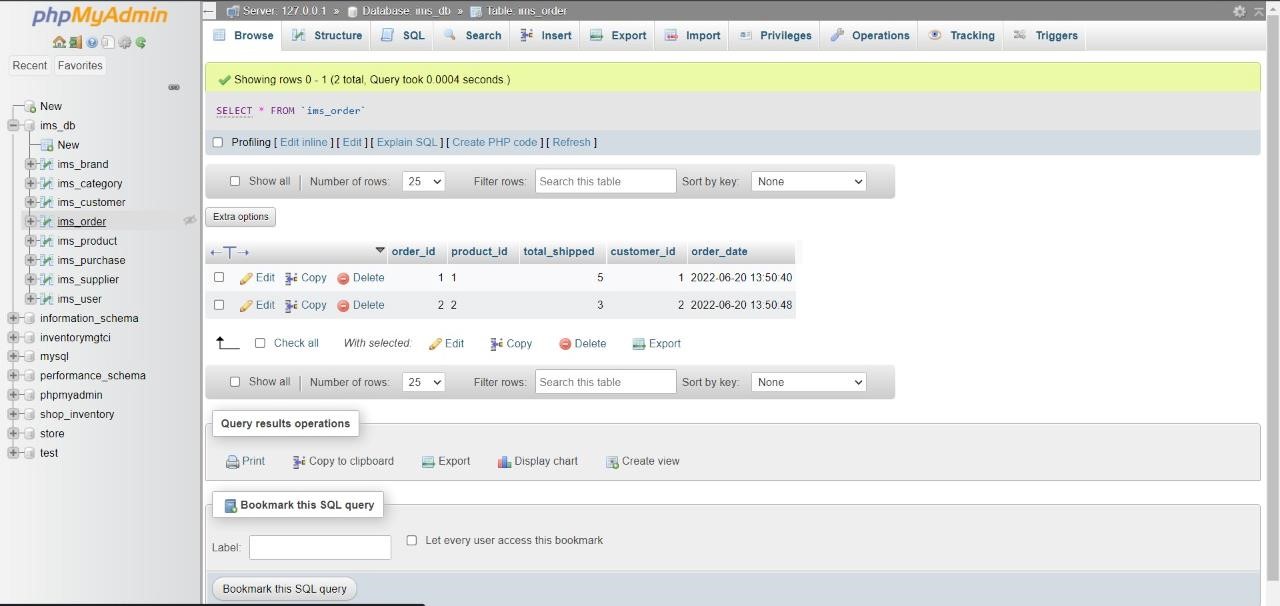


Fig : Order list Table

6.5 Product table:

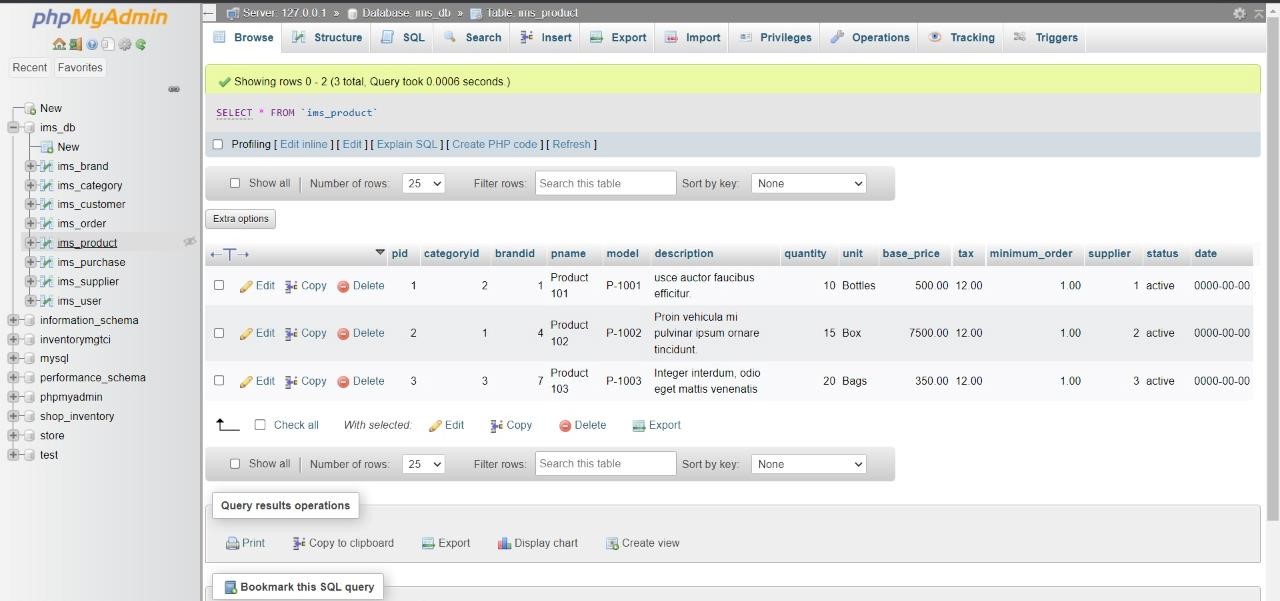


Fig : Product list Table

6.6 Purchase table:

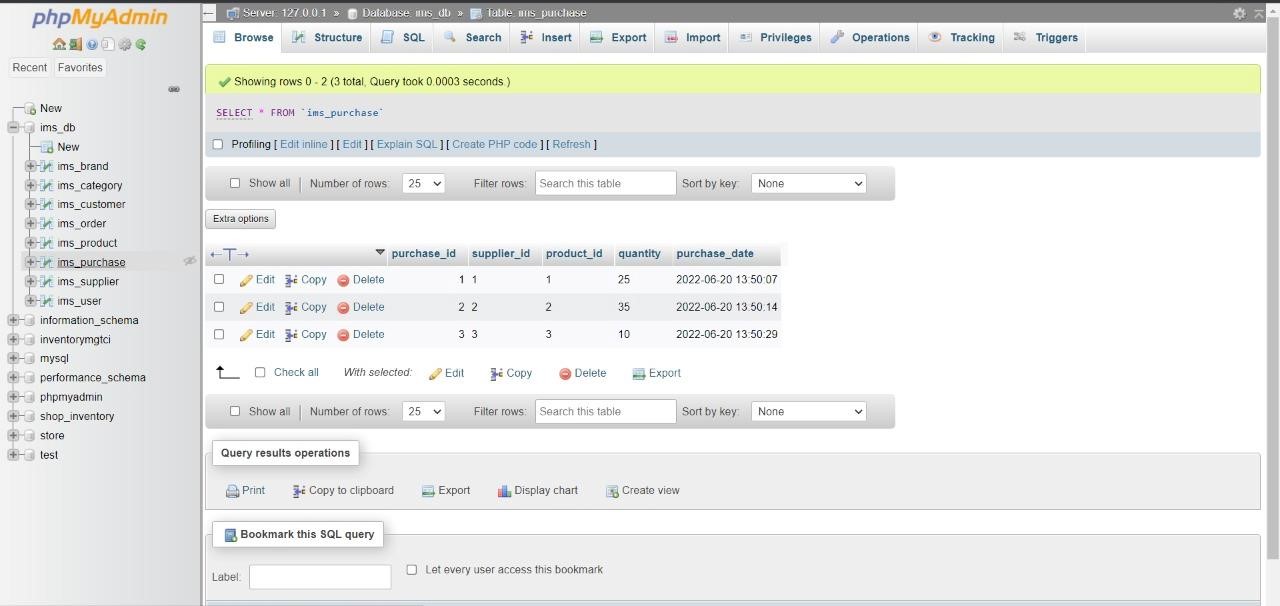


Fig : Purchase list Table

6.7 Supplier list table

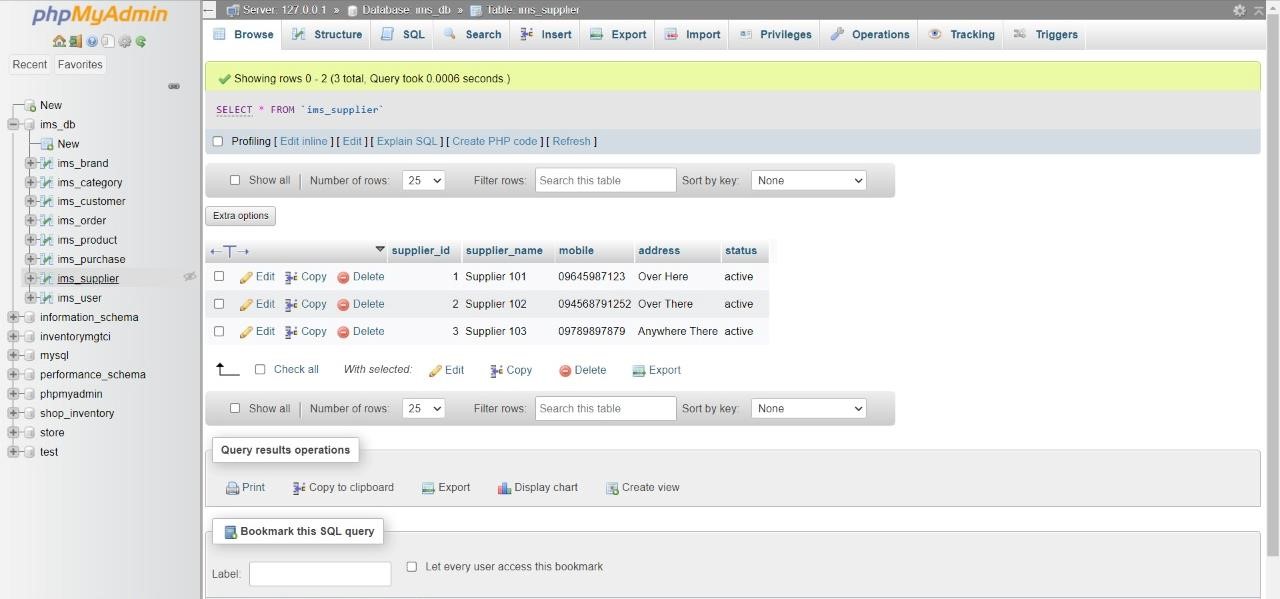


Fig : Supplier list table

6.8 User list table

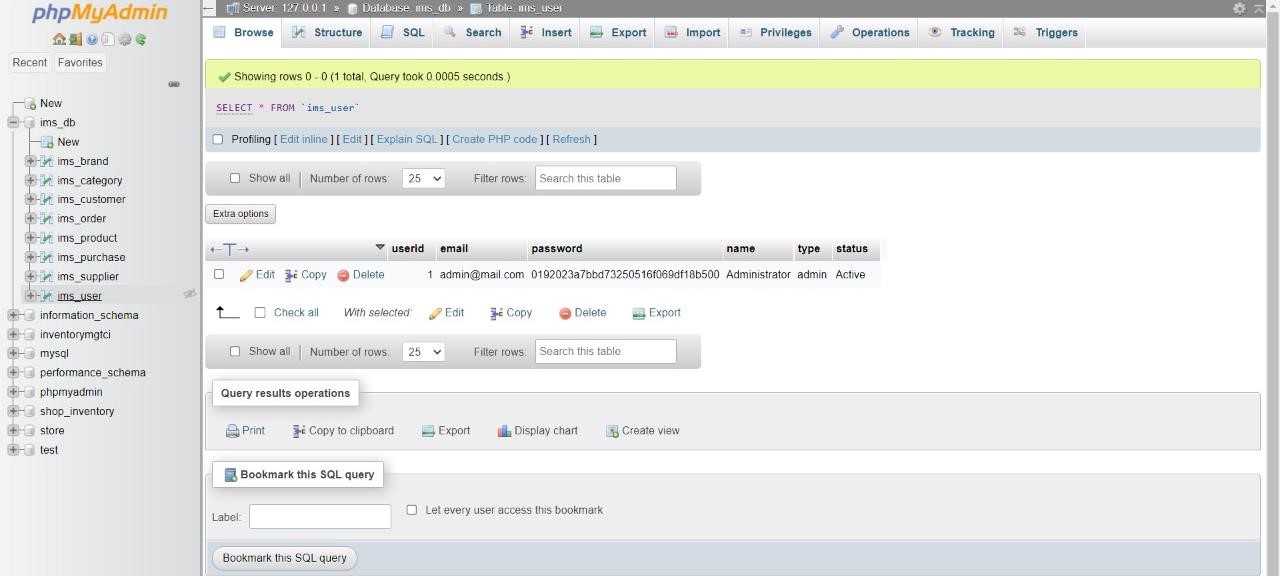


Fig : User list table

# Chapter 7 : SCREENSHOTS

Screenshots show how the user end perceives the software

7.1 Login page:

This is where the user enters the login credentials

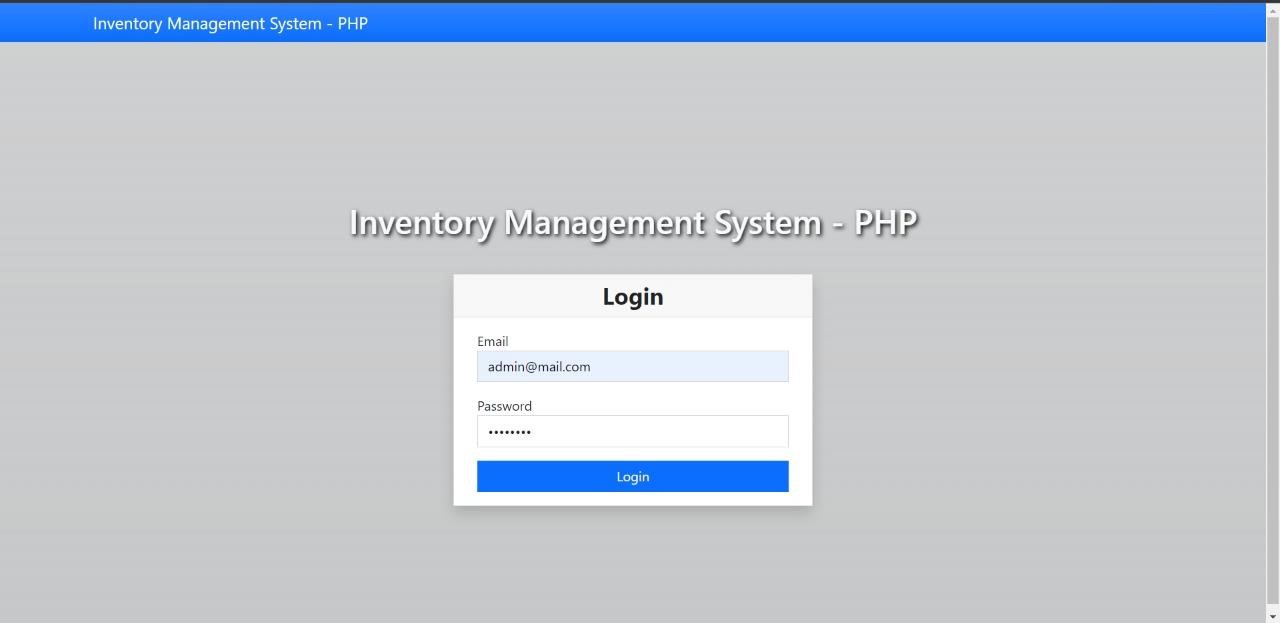
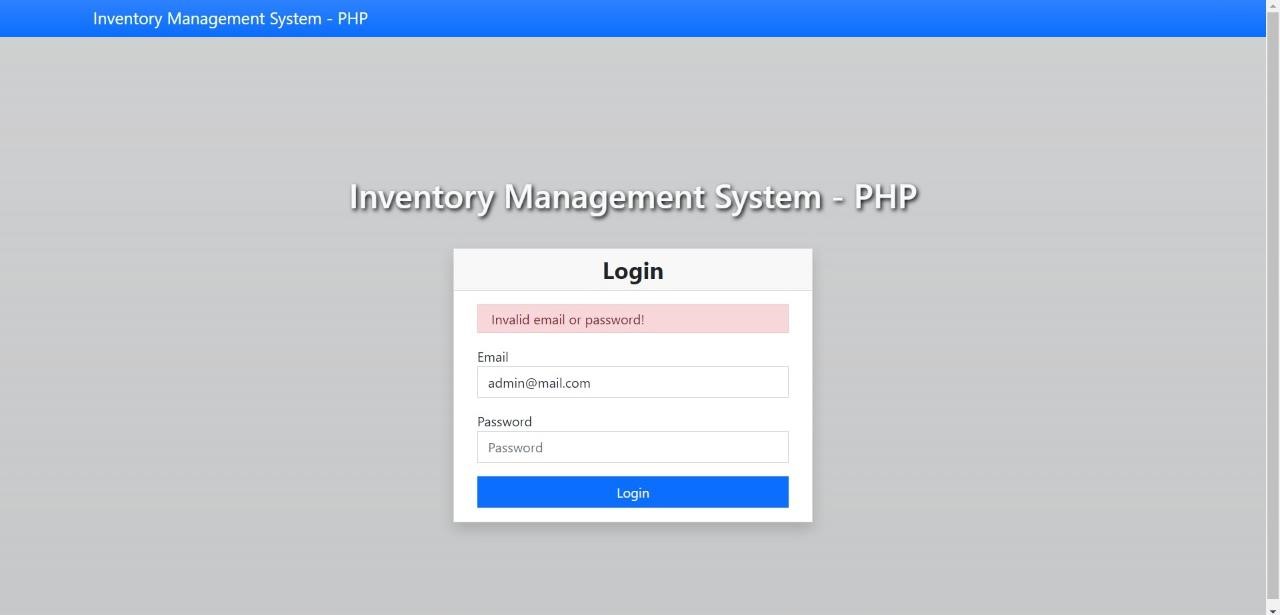


Fig : Login page

7.2Wrong login page:

 Fig : wrong login page

7.3 Home page:

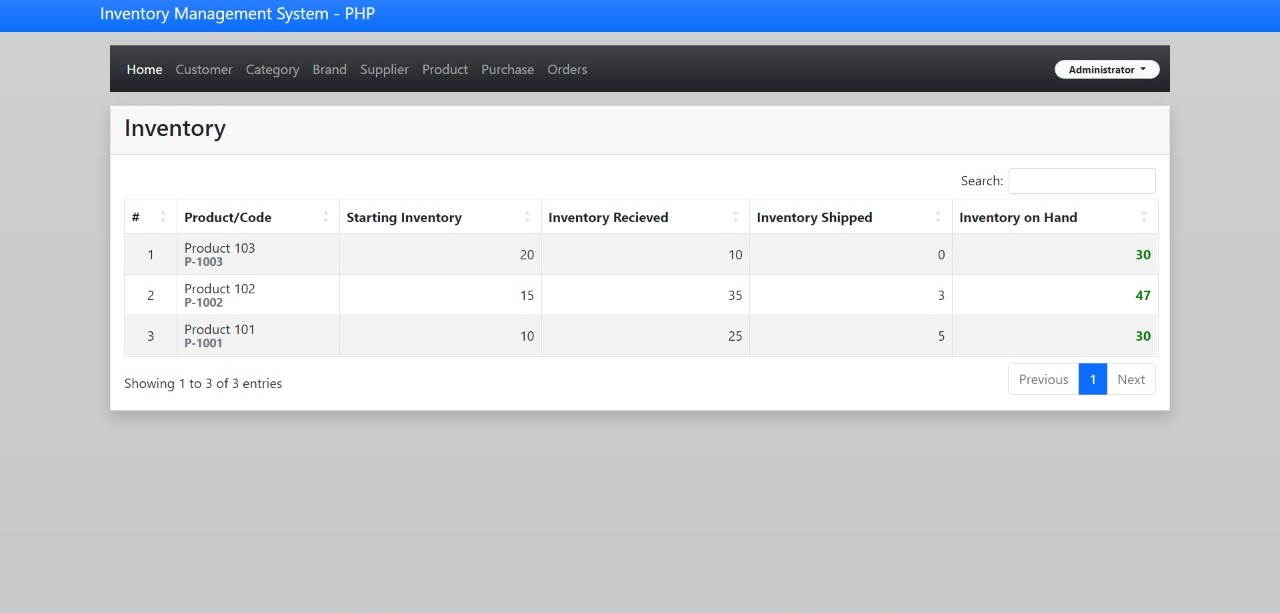


Fig : Home Page

7.4 Customer Page:

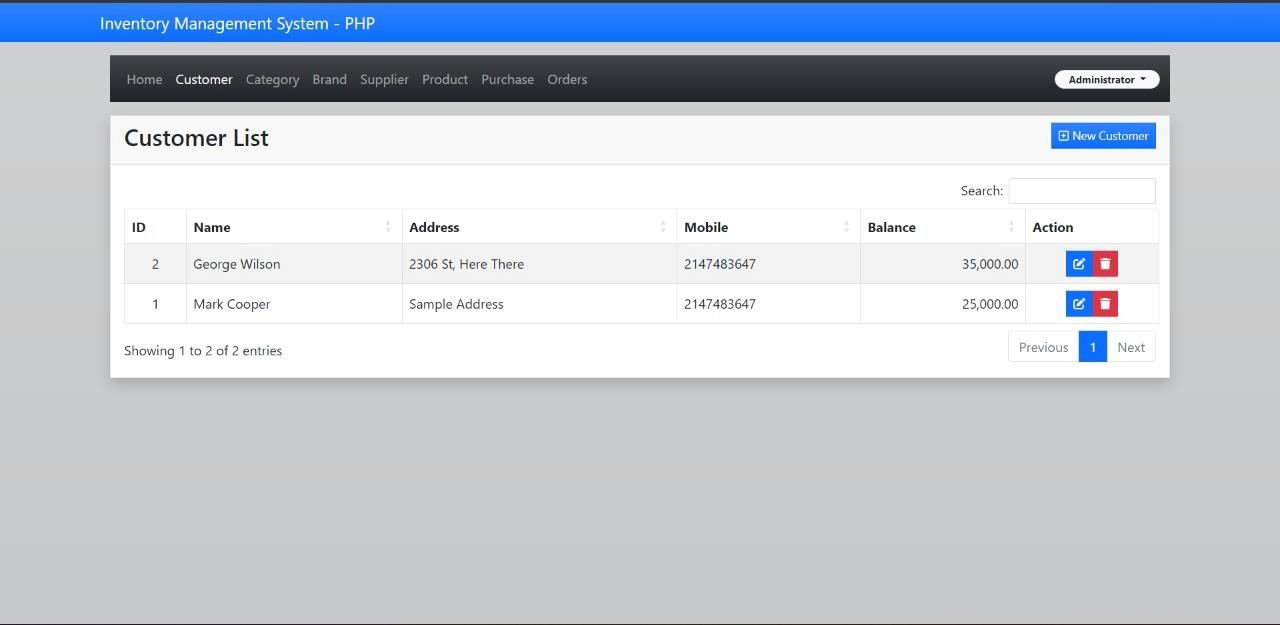


Fig : Customer Page

.5 Category Page:

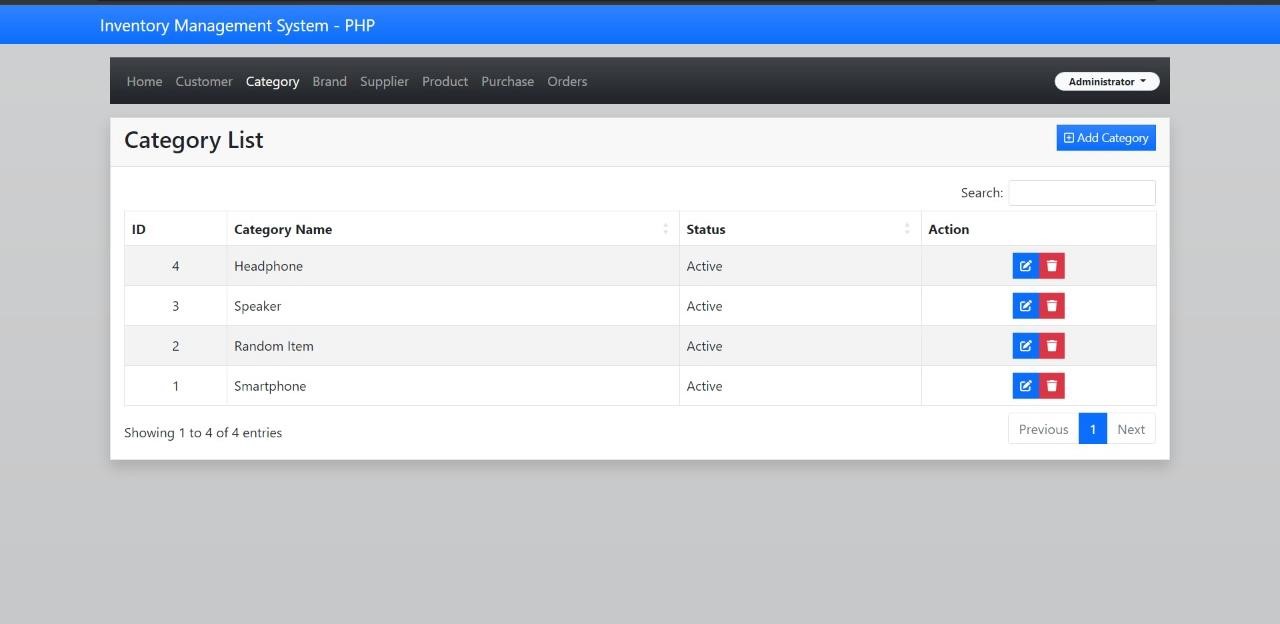


Fig : Category Page

7.6 Brand Page:

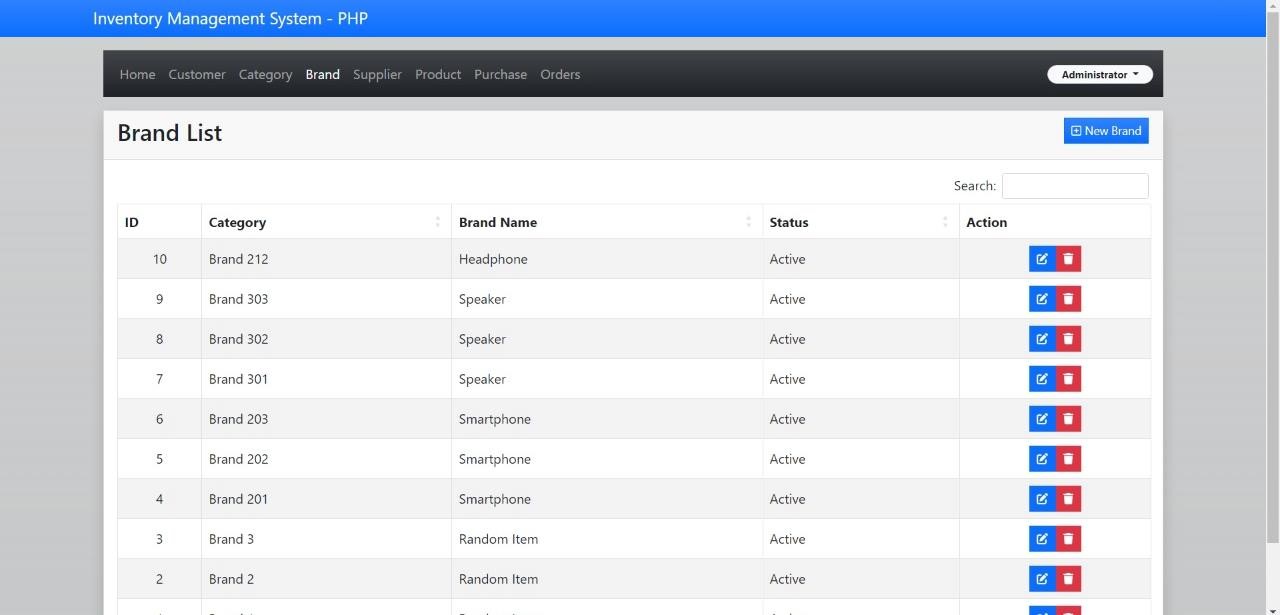


Fig : Brand Page

.7 Supplier Page:

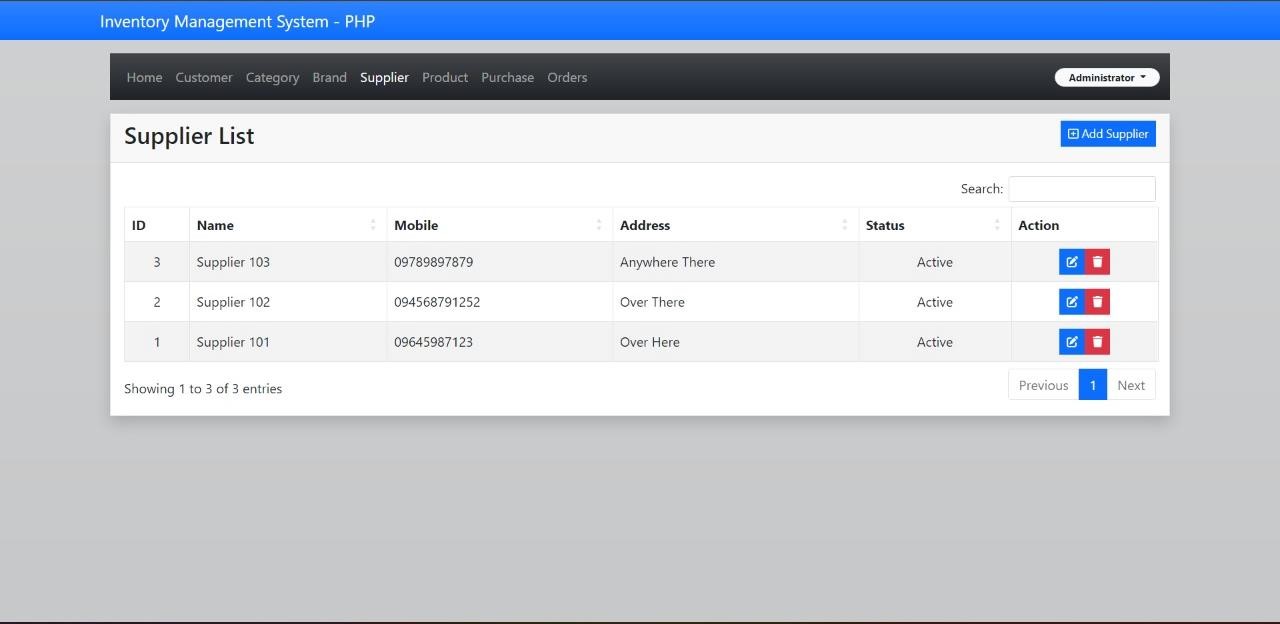


Fig : Supplier Page

7.8 Product Page:

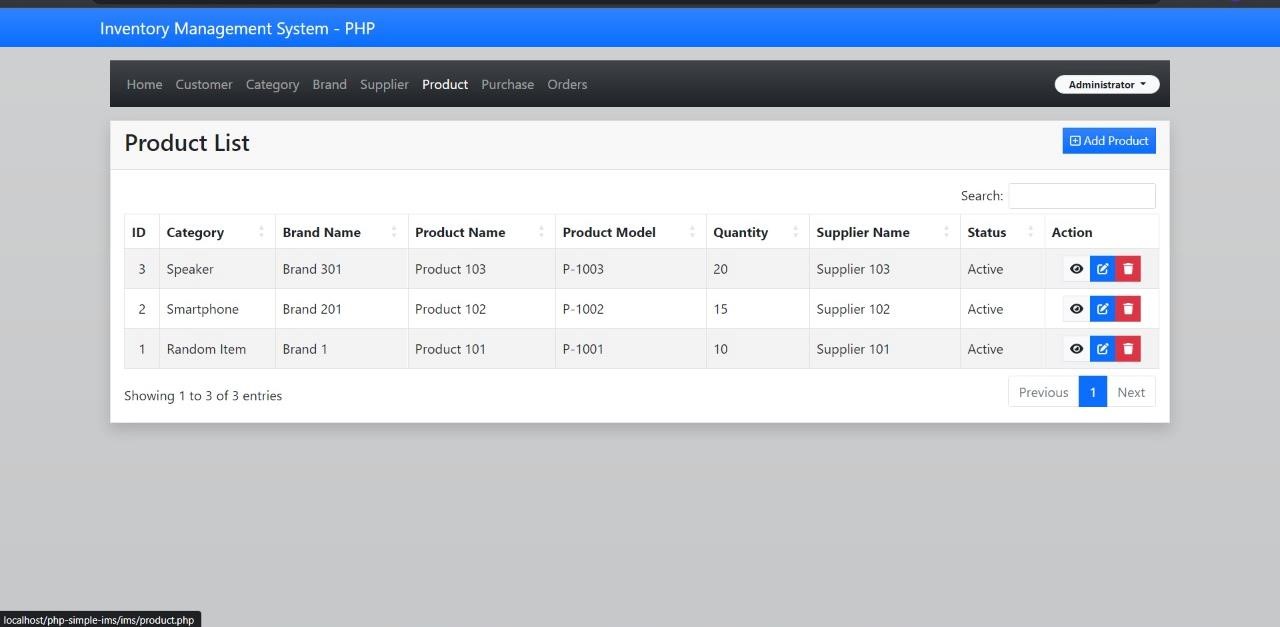


Fig: Product Page

.9 Purchase page:

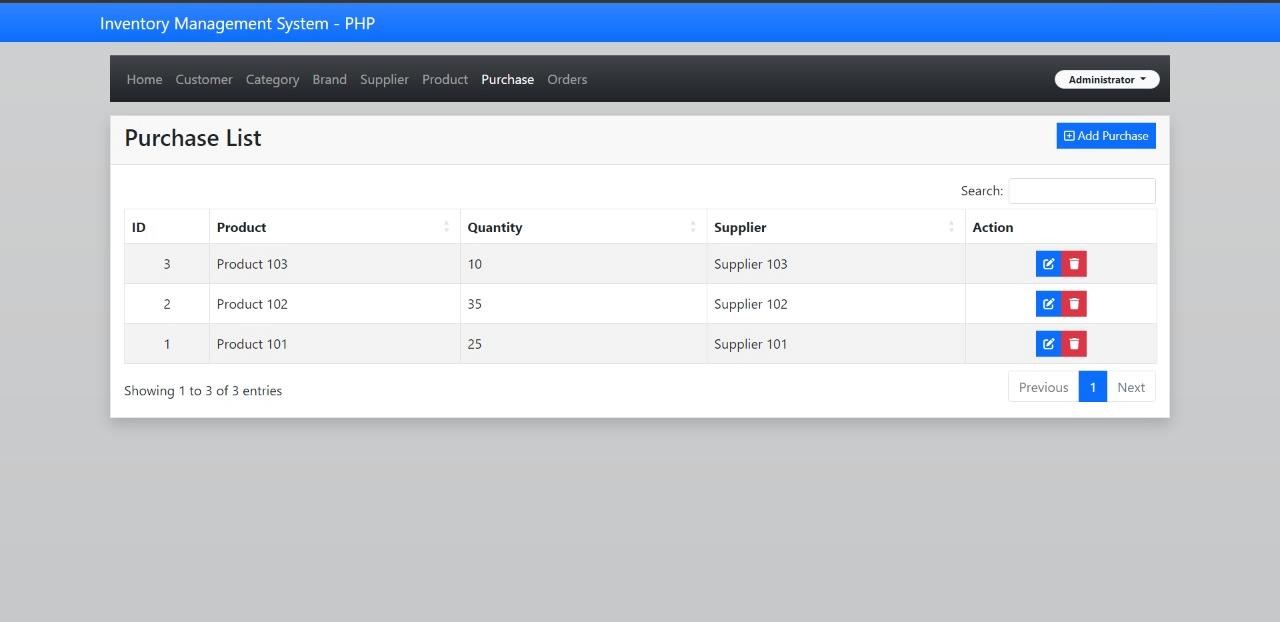


Fig : Purchase Page

7.10 Order page:

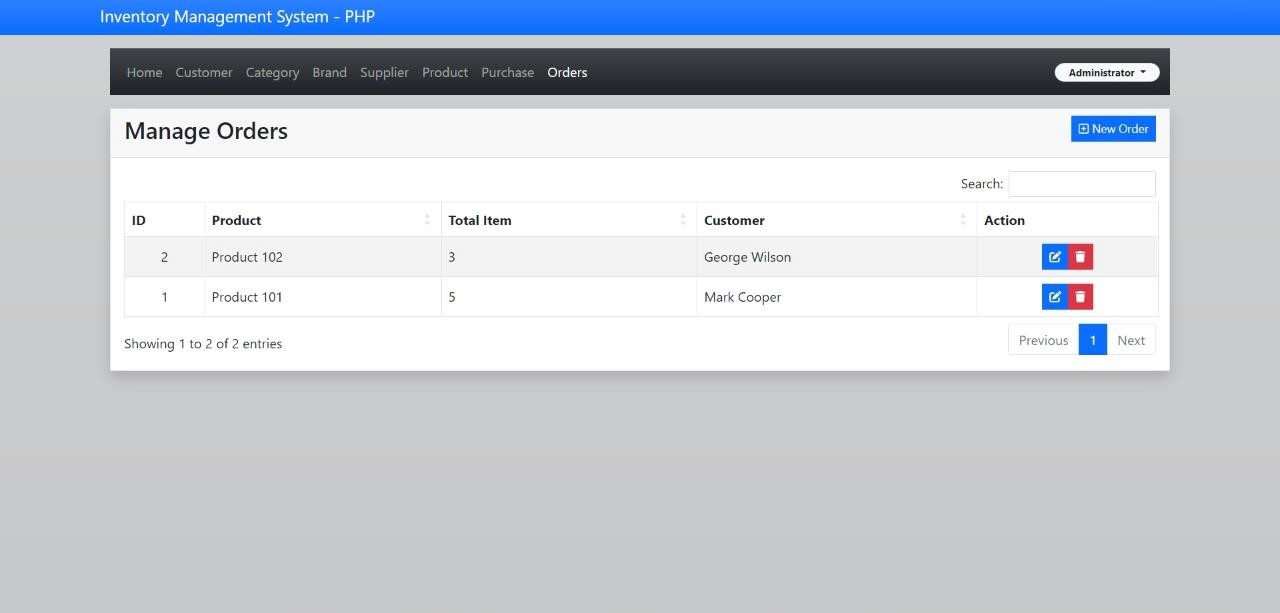


Fig : Order page

# Chapter 8 : CONCLUSION

Our concert management mini project has resulted in a robust web application that efficiently serves organizers, artists, and attendees. By leveraging MySQL, PHP, HTML, CSS, and JavaScript, we've created a user-friendly platform for discovering concerts, purchasing tickets, and engaging with artists. While meeting essential functional requirements, including user authentication, scheduling, ticketing, and reporting, our project showcases the power of technology in revolutionizing live event management. Moving forward, opportunities exist to enhance the platform with features like social media integration and real-time messaging, ensuring its continued relevance and effectiveness in the dynamic world of live entertainment.

# Chapter 9 : BIBLIOGRAPHY

[1]. www.google.com

[2].[www.youtube.com](http://www.youtube.com/)

[3] [www.github.com](http://www.github.com/)