**TABLE OF CONTENTS**

CHAPTER’S NAME PG NO

[LIST OF FIGURES 9](#_Toc161322100)

[CHAPTER-1 1](#_Toc161322102)

[INTRODUCTION TO PROJECT 1](#_Toc161322103)

[1.1 PURPOSE: 1](#_Toc161322104)

[1.2 OBJECTIVES: 1](#_Toc161322105)

[1.3 ADVANTAGES: 2](#_Toc161322106)

[1.4 LIMITATIONS: 2](#_Toc161322107)

[1.5 PROJECT SCOPE: 2](#_Toc161322108)

[CHAPTER-2 3](#_Toc161322109)

[REQUIREMENT SPECIFICATION 3](#_Toc161322110)

[2.1 SOFTWARE REQUIREMENTS: 3](#_Toc161322111)

[2.2 HARDWARE REQUIREMENTS: 3](#_Toc161322112)

[2.3 DETAILS OF THE SOFTWARE & LANGUAGES: 3](#_Toc161322113)

[2.4 FUNCTIONAL OR SPECIFIC REQUIREMENTS 4](#_Toc161322114)

[2.5 NON-FUNCTIONAL REQUIREMENTS 4](#_Toc161322115)

[2.6 SECURITY REQUIREMENTS: 5](#_Toc161322116)

[CHAPTER-3 6](#_Toc161322117)

[INTRODUCTION TO DBMS 6](#_Toc161322118)

[3.1 KEY FEATURES OF DBMS: 6](#_Toc161322119)

[3.2 DATABASE LANGUAGES: 6](#_Toc161322120)

[3.3 APPLICATIONS OF DBMS: 8](#_Toc161322121)

[3.4 INTRODUCTION TO MYSQL: 9](#_Toc161322122)

[3.5 INTRODUCTION TO XAMPP: 10](#_Toc161322123)

[3.6 INTRODUCTION TO PHP: 11](#_Toc161322124)

[What does PHP stand for? 11](#_Toc161322125)

[What is PHP used for? 12](#_Toc161322126)

[CHAPTER-4 13](#_Toc161322127)

[DATABASE DESIGN 13](#_Toc161322128)

[4.1 ENTITY RELATIONSHIP DIAGRAM: 13](#_Toc161322129)

[4.2 SCHEMA DIAGRAM: 13](#_Toc161322130)

[CHAPTER-5 15](#_Toc161322131)

[IMPLEMENTATION 15](#_Toc161322132)

[5.1 BACK END DATABASE: 15](#_Toc161322133)

[5.2 SQL: 17](#_Toc161322134)

[5.3 BACKEND PHP WITH MYSQL CODE: 17](#_Toc161322135)

[5.4 FRONT END CODE: 27](#_Toc161322136)

[CHAPTER-6 34](#_Toc161322137)

[SCREERNSHOTS 34](#_Toc161322138)

[6.1 HOME PAGE: 34](#_Toc161322139)

[6.2 LOGIN & REGISTRATION PAGE: 35](#_Toc161322142)

[6.3 ADD TO CART SECTION: 36](#_Toc161322145)

[6.4 CHECKOUT SECTION: 37](#_Toc161322148)

[6.5 ADMIN DASHBOARD SECTION: 37](#_Toc161322149)

[6.6 ADD USERS: 38](#_Toc161322150)

[6.7 ADD PRODUCTS: 38](#_Toc161322152)

[6.8 DATABASE TABLE VIEW STRUCTURE: 39](#_Toc161322154)

[6.9 BACKEND EMAIL\_INFO DETAILS: 39](#_Toc161322155)

[6.10 BACKEND PRODUCTS DETAILS: 40](#_Toc161322156)

[6.11 BACKEND CATEGORIES/BRAND DETAILS: 40](#_Toc161322158)

[6.12 BACKEND USERS\_INFO: 41](#_Toc161322159)

[6.13 BACKEND LOGIN/SIGNUP CREDENTIALS: 41](#_Toc161322161)

[6.14 BACKEND ORDERS DETAILS: 42](#_Toc161322163)

[6.15 BACKEND ADMIN\_INFO: 42](#_Toc161322165)

[CHAPTER-7 43](#_Toc161322166)

[7.1 CONCLUSION: 43](#_Toc161322167)

[7.2 REFERENCES: 44](#_Toc161322168)

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig No** | **Fig Name** | **Pg No** |
| 3.3 | DBMS Application | 8 |
| 3.4 | MySQL Architecture | 10 |
| 3.5 | XAMPP Control Panel | 11 |
| 3.6 | PHP Dashboard | 12 |
| 4.1 | E-R Diagram | 11 |
| 4.2 | Schema Diagram | 12 |
| 6.1 | Home Page | 34 |
| 6.2 | Login & Registration Page | 35 |
| 6.3 | Add To Cart Page | 36 |
| 6.4 | Checkout page | 37 |
| 6.5 | Admin Dashboard page | 37 |
| 6.6 | Add Users page | 38 |
| 6.7 | Add products page | 38 |
| 6.8 | Database Table View Structure | 39 |
| 6.9 | Backend Email\_info details | 39 |
| 6.10 | Backend Product Details | 40 |
| 6.11 | Backend categories/brand details | 40 |
| 6.12 | Backend Users\_info | 41 |
| 6.13 | Backend Login/Signup Credentials | 41 |
| 6.14 | Backend Orders Details | 42 |
| 6.15 | Backend Admin\_info Details | 42 |

# CHAPTER-1

## INTRODUCTION TO PROJECT

### 1.1 PURPOSE:

* The purpose of an Art Gallery Management System is to efficiently organize and streamline various aspects of art gallery operations, including:
* Cataloging and organizing artworks, including details such as artist, medium, dimensions, and pricing.
* Managing and tracking the gallery's inventory, ensuring accurate records of available artworks.
* Facilitating the sales process, from generating invoices to tracking transactions, and managing financial aspects.
* Maintaining comprehensive records of artists, their portfolios, and relevant details.
* Enhancing customer interactions by keeping track of purchase histories, preferences, and contact information for personalized services.
* Providing insights through reporting and analytics for informed decision-making and strategic planning.
* The system aims to improve overall gallery efficiency, enhance customer experience, and contribute to better decision-making for gallery management.

### 1.2 OBJECTIVES:

* The main objective of the project is to **design and develop a user friendly-system**, Easy to use and an efficient computerized system.
* The Virtual Art Gallery's primary objective is to simplify and enhance the management of online art galleries.
* It streamlines various administrative tasks, including artwork uploads, artist collaborations, and customer interactions.
* This approach **reduces paper waste and promotes sustainable practices** in art gallery management.
* In today's digital age, the art world is expanding its horizons through innovative online platforms.
* Artists can now reach a global audience and art enthusiasts can explore and purchase artwork from the comfort of their homes.
* To develop an accurate and flexible systems so that it will eliminate data redundancy.
* To make a software fast in processing, with good user interface.
* To provide better Graphical User Interface (GUI).

### 1.3 ADVANTAGES:

* The system streamlines the organization of artworks, improving accessibility and categorization for easy retrieval.
* It facilitates smooth sales processes, tracking transactions and managing financial aspects seamlessly, optimizing the overall sales experience.
* The system provides a centralized platform to efficiently manage artist details and track inventory levels, ensuring accurate and up-to-date information.
* It enables better customer interactions by maintaining a comprehensive database, including purchase history and preferences, enhancing personalized services.
* The system utilizes analytics to derive insights, supporting data-driven decision-making for strategic planning and improving gallery operations.

### 1.4 LIMITATIONS:

* May get Vulnerable to disruptions from technical issues.
* Initial and ongoing costs for implementation and maintenance.
* Challenges in integrating with other tools or databases.

### 1.5 PROJECT SCOPE:

* Streamline gallery operations and enhance overall efficiency.
* Manage artwork, artist information, inventory, and sales.
* Customer relationship management features for improved interactions.
* Focus on usability, accessibility, and intuitive user interfaces.
* Regular updates to address evolving gallery management needs.
* Potential limitations considered for effective system implementation.

# CHAPTER-2

## REQUIREMENT SPECIFICATION

### SOFTWARE REQUIREMENTS:

Frontend- HTML, CSS, Java Script, Ajax (Asynchronous JavaScript and XML), Bootstrap

Backend- PHP ( PHP5.6, PHP7.x), MySQL 5x, jQuery

Operating System: Windows 10 Pro or Higher

* + - Google Chrome/Internet Explorer
    - XAMPP (Version-3.7)
    - workspace editor: VS Code

### HARDWARE REQUIREMENTS:

* + 1. Computer with a 1.1 GHz or faster processor
    2. Minimum 6GB of RAM or more
    3. 2.5 GB of available hard-disk space
    4. 5400 RPM hard drive
    5. 1366 × 768 or higher-resolution display

### 2.3 DETAILS OF THE SOFTWARE & LANGUAGES:

* + Operating System: Compatible Operating system like Windows or MacOS.
  + Programming Languages and Technologies: Implemented for user interface
    - * HTML5: Markup language for structuring web pages.
      * CSS: Style sheet language for designing the presentation of web pages.
      * JavaScript (JS): Programming language for adding interactivity and dynamic behavior to web pages.
      * Ajax (Asynchronous JavaScript and XML): Technique for making asynchronous HTTP requests from the browser to the server, enabling dynamic content updates without reloading the entire page.
      * jQuery: JavaScript library that simplifies client-side scripting and DOM manipulation.
      * PHP: Server-side scripting language for building dynamic web applications and interacting with databases. [Required: PHP5.6, PHP7.x]
  + Integrated Development Environment (IDE):
    - Visual Studio Code (VS Code): A lightweight yet powerful source code editor developed by Microsoft, with support for various programming languages, debugging, and version control integration. It provides features such as syntax highlighting, code completion, and extensions that enhance productivity for developers working on web applications.
  + Database Management System: Software for storing and managing data related to artwork, artists, transactions, and user accounts securely. Relational database management system for storing and retrieving structured data. [Required: MySQL 5x]
  + Web Hosting Tools: The system includes a free web hosting services for maintaining a web presence. [Required: 000webhosting.com or hostinger]
  + Web Browser: Mozilla, Google Chrome, IE8, OPERA
  + Software Bundles: Software bundles provide a complete development environment for web applications. They include Apache, MySQL, and PHP allowing developers to set up a local server environment for building and testing websites or web applications. [Required: XAMPP / Wamp / Mamp/ Lamp (anyone) ]

### 2.4 FUNCTIONAL OR SPECIFIC REQUIREMENTS

The system should satisfy the following requirements:

* Logging into the system
* Signup option
* View category Details
* Checkout & add to cart Option
* View payments
* Logout option

### NON-FUNCTIONAL REQUIREMENTS

All of the application data is stored in an Oracle database, and therefore an Oracle Database must also be installed on the host computer. As with Apache2, this software is freely available and can be installed and run under most operating systems. The server hardware can be any computer capable of running both the web and database servers and handling the expected traffic. For a small-scale art gallery management that is not expecting to see much web traffic, an average personal computer may be appropriate. Once the site starts generating more hits, though, it will likely be necessary to upgrade to a dedicated host to ensure proper performance. The exact cut-offs will need to be determined through a more thorough stress testing of the system.

### 2.6 SECURITY REQUIREMENTS:

Some of the factors that are identified to project the software from accidental or malicious access, use, modification, destruction, or disclosure are described below.

* + - * + Ascertain functions to different modules
        + Restrict communication between areas of the program
        + Check data integrity for critical variables
        + A later version of the software will incorporate encryption techniques in the user/license authentication process
        + Communication needs to be restricted when the application is validating the user or license.

# CHAPTER-3

## INTRODUCTION TO DBMS

A Database Management System (DBMS) is a software system that is designed to manage and organize data in a structured manner. It allows users to create, modify, and query a database, as well as manage the security and access controls for that database.

DBMS provides an environment to store and retrieve the data in coinvent and efficient manner

### 3.1 KEY FEATURES OF DBMS:

* + - * **Data modeling:** A DBMS provides tools for creating and modifying data models, which define the structure and relationships of the data in a database.
      * **Data storage and retrieval:** A DBMS is responsible for storing and retrieving data from the database, and can provide various methods for searching and querying the data.
      * **Concurrency control:** A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.
      * **Data integrity and security:** A DBMS provides tools for enforcing data integrity and security constraints, such as constraints on the values of data and access controls that restrict who can access the data.
      * **Backup and recovery:** A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.
      * **DBMS can be classified into two types:** Relational Database Management System (RDBMS) and Non-Relational Database Management System (NoSQL or Non-SQL).

### 3.2 DATABASE LANGUAGES:

* + - Data Definition Language
    - Data Manipulation Language
    - Data Control Language
    - Transactional Control Language

#### **3.2.1 DATA DEFINITION LANGUAGE:**

**DDL** is the short name for Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

* + - **CREATE:** to create a database and its objects like (table, index, views, store procedure, function, and triggers)
    - **ALTER:** alters the structure of the existing database
    - **DROP:** delete objects from the database

#### **3.2.2 DATA MANIPULATION LANGUAGE:**

**DML** is the short name for Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

* + - **SELECT:** retrieve data from a database
    - **INSERT:** insert data into a table
    - **UPDATE:** updates existing data within a table
    - **DELETE:** Delete all records from a database table
    - **MERGE:** UPSERT operation (insert or update)
    - **CALL:** call a PL/SQL or Java subprogram
    - **EXPLAIN PLAN:** interpretation of the data access path
    - **LOCK TABLE:** concurrency Control
    - **TRUNCATE:** remove all records from a table, including all spaces allocated for the records are removed
    - **COMMENT:** add comments to the data dictionary
    - **RENAME:** rename an object

#### **3.2.3 DATA CONTROL LANGUAGE:**

**DCL** is short for Data Control Language which acts as an access specifier to the database (basically to grant and revoke permissions to users in the database

* + - **GRANT:** grant permissions to the user for running DML(SELECT, INSERT, DELETE,…) commands on the table
    - **REVOKE:** revoke permissions to the user for running DML(SELECT, INSERT, DELETE,…) command on the specified table

#### **3.2.4 TRANSACTIONAL CONTROL LANGUAGE:**

**TCL** is short for Transactional Control Language which acts as an manager for all types of transactional data and all transactions. Some of the command of TCL are

* + - **Roll Back:** Used to cancel or Undo changes made in the database
    - **Commit:** It is used to apply or save changes in the database
    - **Save Point:** It is used to save the data on the temporary basis in the database.

### 3.3 APPLICATIONS OF DBMS:

* + - * **Enterprise Information:** Sales, accounting, human resources, Manufacturing, online retailers.
      * **Banking and Finance Sector:** Banks maintaining the customer details, accounts, loans, banking transactions, credit card transactions. Finance: Storing the information about sales and holdings, purchasing of financial stocks and bonds.
      * **University:** Maintaining the information about student course enrolled information, student grades, staff roles.
      * **Airlines:** Reservations and schedules.
      * **Telecommunications:** Prepaid, postpaid bills maintained.

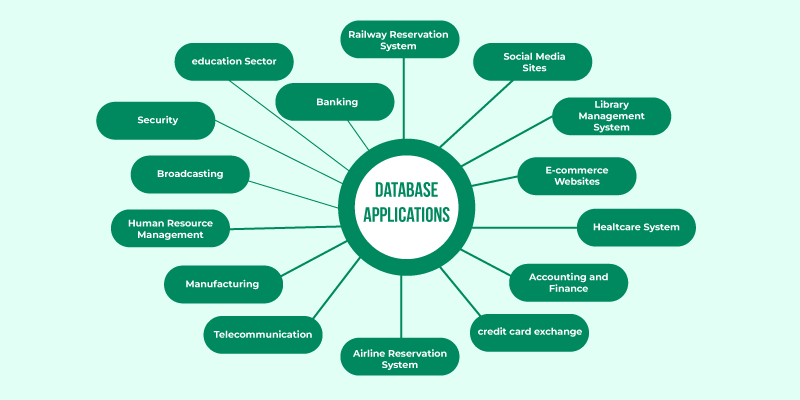


Fig 3.3 DBMS Application

### 3.4 INTRODUCTION TO MYSQL:

What is MySQL?

MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold the vast amounts of information in a corporate network. In particular, a relational database is a digital store collecting data and organizing it according to the relational model. In this model, tables consist of rows and columns, and relationships between data elements all follow a strict logical structure.

An DBMS is simply the set of software tools used to actually implement, manage, and query such a database.

MySQL is integral to many of the most popular software stacks for building and maintaining everything from customer-facing web applications to powerful, [data-driven B2B services](https://www.talend.com/resources/business-intelligence-data-analytics/). Its open-source nature, stability, and rich feature set, paired with ongoing development and support from Oracle, have meant that internet-critical organizations such as Facebook, Flickr, Twitter, Wikipedia, and YouTube [all employ MySQL backends](https://www.mysql.com/customers/industry/).

**MySQL is widely compatible**

Though often associated with internet applications or web services, MySQL was designed to be extensively compatible with other technologies and architectures. The RDBMS runs on all major computing platforms, including Unix-based operating systems, such as the myriad Linux distributions or Mac OS, and Windows.

MySQL’s client-server architecture means it can support a variety of backends, as well as different programming interfaces. Data can be directly [migrated from MySQL](https://www.talend.com/resources/understanding-data-migration-strategies-best-practices/) to its forks (e.g. MariaDB), as well as most other RDBMSs thanks to architectural and language similarities.

Established Oracle and third-party migration tools further allow MySQL to move data to and from a vast set of general storage systems, whether these are designed to be on-premises or cloud-based. MySQL can be deployed in virtualized environments, distributed or centralized, and even exists as portable standalone libraries for learning purposes, testing, or small applications.

MySQL’s wide compatibility with all these other systems and software makes it a particularly practical choice of RDBMS in most situations.

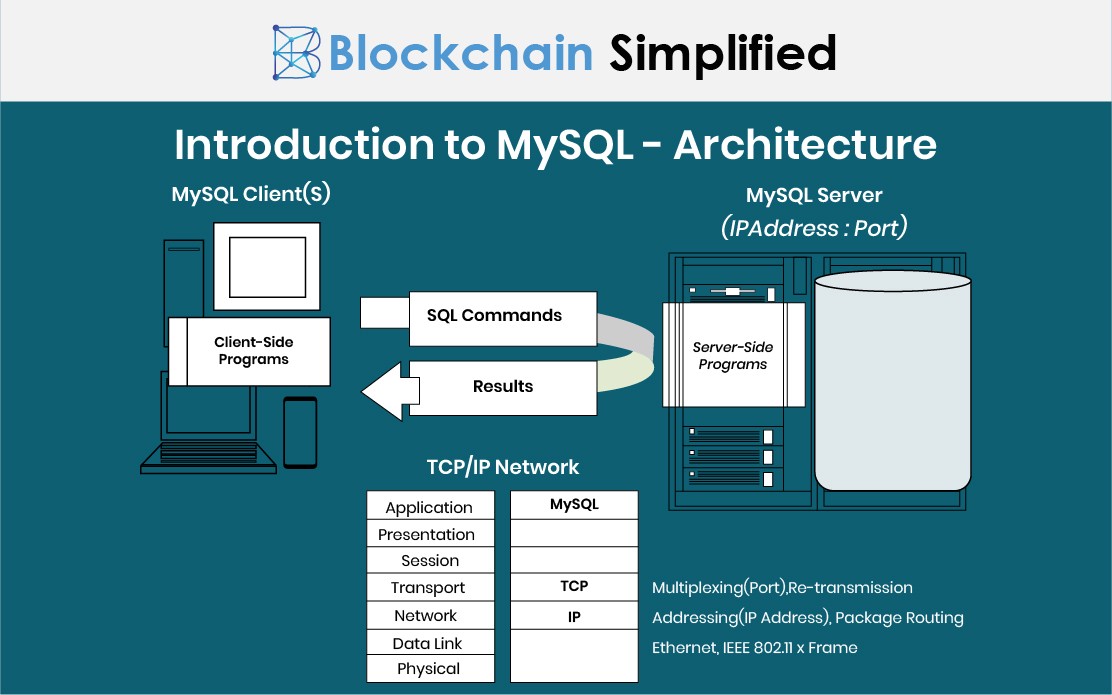


Fig 2.2 MySQL Architecture

### 3.5 INTRODUCTION TO XAMPP:

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

Many other components are also part of this collection of software and are explained below.

1. Cross-Platform Versatility:

- Apache distributions support Windows, Linux, and macOS, ensuring compatibility across diverse operating systems.

2. Apache Web Server:

- A widely-used, free HTTP server by the Apache Software Foundation, delivering web content globally upon user requests.

3. MariaDB Database Management:

- Replacing MySQL in XAMPP, MariaDB offers robust relational database management services.

4. PHP for Dynamic Web Development:

- A versatile backend scripting language supporting various databases, installed on any platform, and developed in C language.

5. Perl for Dynamic Solutions:

- Combining Perl 5 and Perl 6, Perl is flexible, robust, and ideal for solving system administration, web development, and networking challenges.

6. phpMyAdmin Database Tool:

- Version 4.0.4 facilitates MariaDB administration, serving as a crucial tool in the XAMPP package.

7. OpenSSL Security Implementation:

- XAMPP includes version 0.9.8 of OpenSSL, ensuring secure data transmission through SSL and TLS protocols.

8. XAMPP Control Panel:

- Version 3.2.1 offers efficient operation and regulation of XAMPP components, enhancing user control and management.

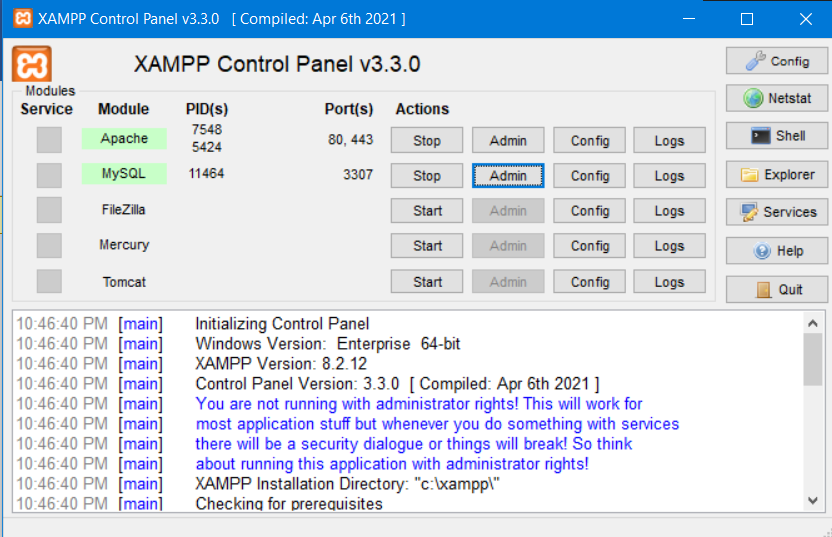


Fig 2.3 XAMPP Control Panel

### 3.6 INTRODUCTION TO PHP:

PHP is an open-source, server-side programming language that can be used to create websites, applications, customer relationship management systems and more. It is a widely-used general-purpose language that can be embedded into HTML. This functionality with HTML means that the PHP language has remained popular with developers as it helps to simplify HTML code.

## What does PHP stand for?

PHP stands for ‘PHP: Hypertext Preprocessor’, with the original PHP within this standing for ‘Personal Home Page’. The acronym has changed as the language developed since its launch in 1994 to more accurately reflect its nature. Since its release, there have been 8 versions of PHP, as of 2022, with version 8.1 currently a popular choice among those using the language on their websites.

## What is PHP used for?

PHP programming can be used to create most things that a software developer needs. However, there are three main areas in which it thrives.

1. **Server-side scripting**  
   Server-side Script is PHP’s main strength. If you are just learning to code and want to explore server-side scripting, PHP is a great language to learn. To get cracking with PHP server-side scripting you’ll need to have a PHP parser, web server and web browser.
2. **Command-line scripting**  
   Command-line scripting is ideal for scripts made using cron (Linux) or Task Scheduler (Windows). It is also great for simple text processing.
3. **Writing desktop applications**  
   PHP is probably not the best language to use to create desktop applications but for the advanced web developer, it provides you with many more options than its competitors.

PHP can do many other things. For example, it is excellent at collecting form data, encrypting user data and sending and receiving cookies. One of the major features of PHP that makes it so usable is that it is compatible with all major operating systems so you can code no matter what tech you are using.

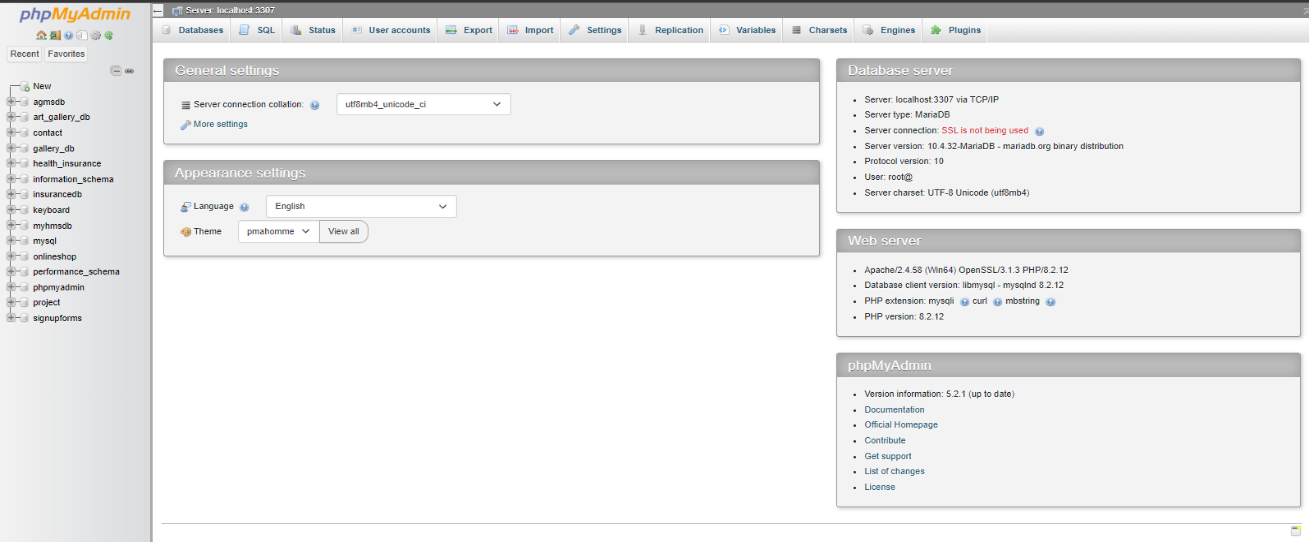
****

Fig 2.4 PHP Dashboard

# CHAPTER-4

## DATABASE DESIGN

### 4.1 ENTITY RELATIONSHIP DIAGRAM:

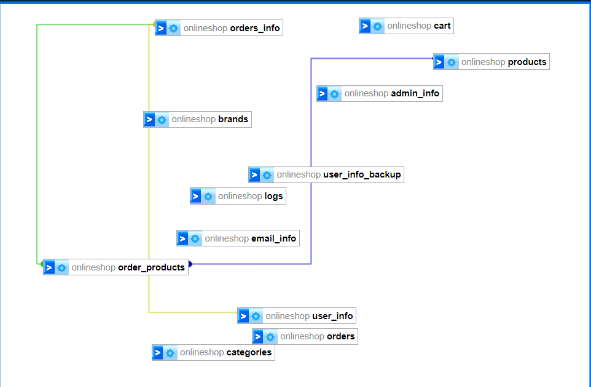
ER Relationship model allows us to describe the data involved in a real-world enterprise in terms of objects and their relationship widely used to develop an initial database design. It is primarily important in its role in database design.

****

Fig 4.1 E-R Diagram

### SCHEMA DIAGRAM:

A database schema can be represented in a visual diagram, which shows the database object and their relationship which represents the logical view of the database and how the relationships among them are represented.



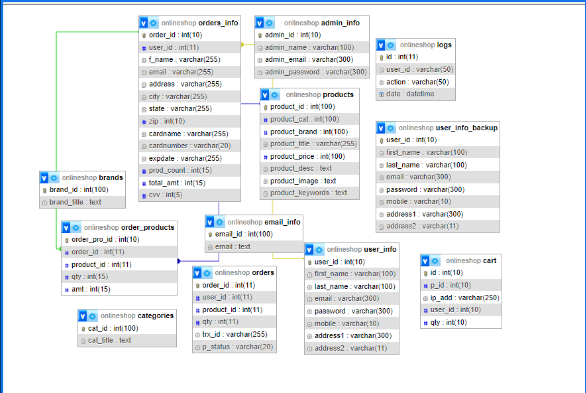


Fig 4.2 Schema Diagram

# CHAPTER-5

## IMPLEMENTATION

The implementation of PHP and MySQL represents a dynamic and powerful synergy in web development. PHP, a server-side scripting language, seamlessly integrates with MySQL, a robust relational database management system. PHP's flexibility allows for the creation of dynamic and interactive web pages, facilitating the integration of databases for efficient data handling. MySQL, renowned for its reliability and speed, provides a scalable and secure platform for storing, retrieving, and managing data. Together, PHP and MySQL form a popular stack for web development, enabling developers to build feature-rich, responsive, and scalable applications. Their collaborative implementation empowers developers to create dynamic websites and web applications that seamlessly interact with databases, offering a seamless user experience. This combination has become a cornerstone for web developers, ensuring the delivery of high-performance and database-driven solutions across the digital landscape.

Top of Form

### BACK END DATABASE:

A Database Management System (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

* + - A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
    - The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model,
    - since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
    - Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory). A database query language and report

writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.

* + - Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called sub schemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and student data.
    - If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multiuser organization. These controls are only available when a set of application programs are customized for each data entry and updating function.
    - A transaction mechanism, that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).
      * It also maintains the integrity of the data in the database.
      * The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance).

When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. to the Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

### 5.2 SQL:

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

* + - In the relational model, data is stored in structures called relations or tables*.*

SQL statements are issued for the purpose of:

* + - Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes)

### 5.3 BACKEND PHP WITH MYSQL CODE:

*-- phpMyAdmin SQL Dump*

*-- version 5.2.1*

*-- https://www.phpmyadmin.net/*

*--*

*-- Host: localhost:3307*

*-- Generation Time: Mar 03, 2024 at 01:07 PM*

*-- Server version: 10.4.32-MariaDB*

*-- PHP Version: 8.2.12*

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

*/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/*;

*/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/*;

*/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/*;

*/\*!40101 SET NAMES utf8mb4 \*/*;

*--*

*-- Database: `onlineshop`*

*--*

DELIMITER $$

*--*

*-- Procedures*

*--*

CREATE DEFINER=`root`@`localhost` PROCEDURE `getcat` (IN `cid` INT)   SELECT \* FROM categories WHERE cat\_id=cid$$

DELIMITER ;

*-- --------------------------------------------------------*

*--*

*-- Table structure for table `admin\_info`*

*--*

CREATE TABLE `admin\_info` (

  `admin\_id` int(10) NOT NULL,

  `admin\_name` varchar(100) NOT NULL,

  `admin\_email` varchar(300) NOT NULL,

  `admin\_password` varchar(300) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `admin\_info`*

*--*

INSERT INTO `admin\_info` (`admin\_id`, `admin\_name`, `admin\_email`, `admin\_password`) VALUES

(1, 'admin', 'admin@gmail.com', '25f9e794323b453885f5181f1b624d0b');

*-- --------------------------------------------------------*

*--*

*-- Table structure for table `brands`*

*--*

CREATE TABLE `brands` (

  `brand\_id` int(100) NOT NULL,

  `brand\_title` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `brands`*

*--*

INSERT INTO `brands` (`brand\_id`, `brand\_title`) VALUES

(1, 'Winsor & Newton'),

(2, 'Golden Artist Colors'),

(3, 'Faber-Castell'),

(4, 'Rembrandt'),

(5, 'Michael Harding'),

(6, 'Derwent');

*-- --------------------------------------------------------*

*--*

*-- Table structure for table `cart`*

*--*

CREATE TABLE `cart` (

  `id` int(10) NOT NULL,

  `p\_id` int(10) NOT NULL,

  `ip\_add` varchar(250) NOT NULL,

  `user\_id` int(10) DEFAULT NULL,

  `qty` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `cart`*

*--*

INSERT INTO `cart` (`id`, `p\_id`, `ip\_add`, `user\_id`, `qty`) VALUES

(6, 26, '::1', 4, 1),

(9, 10, '::1', 7, 1),

(10, 11, '::1', 7, 1),

(11, 45, '::1', 7, 1),

(55, 5, '::1', 14, 1),

(56, 1, '::1', 9, 1),

(57, 2, '::1', 9, 1),

(71, 61, '127.0.0.1', -1, 1),

(160, 1, '::1', 1, 1),

(166, 61, '::1', -1, 1),

(167, 62, '::1', -1, 1);

*-- --------------------------------------------------------*

*--*

*-- Table structure for table `categories`*

*--*

CREATE TABLE `categories` (

  `cat\_id` int(100) NOT NULL,

  `cat\_title` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `categories`*

*--*

INSERT INTO `categories` (`cat\_id`, `cat\_title`) VALUES

(1, 'Sketches'),

(2, 'Conceptual Arts'),

(3, 'Sculptures'),

(4, 'Painting Art'),

(5, 'Visual Arts'),

(6, 'Oil Painting'),

(7, 'Street arts');

*-- Table structure for table `email\_info`*

*--*

CREATE TABLE `email\_info` (

  `email\_id` int(100) NOT NULL,

  `email` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `email\_info`*

*--*

INSERT INTO `email\_info` (`email\_id`, `email`) VALUES

(3, 'admin@gmail.com'),

(4, 'puneethreddy951@gmail.com'),

(5, 'puneethreddy@gmail.com'),

(6, 'altayyabtravels646@mail.com'),

(7, 'hajeera@gmail.com');

*-- Table structure for table `logs`*

*--*

CREATE TABLE `logs` (

  `id` int(11) NOT NULL,

  `user\_id` varchar(50) NOT NULL,

  `action` varchar(50) NOT NULL,

  `date` datetime NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*-- --------------------------------------------------------*

*--*

*-- Table structure for table `orders`*

*--*

CREATE TABLE `orders` (

  `order\_id` int(11) NOT NULL,

  `user\_id` int(11) NOT NULL,

  `product\_id` int(11) NOT NULL,

  `qty` int(11) NOT NULL,

  `trx\_id` varchar(255) NOT NULL,

  `p\_status` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `orders`*

*--*

INSERT INTO `orders` (`order\_id`, `user\_id`, `product\_id`, `qty`, `trx\_id`, `p\_status`) VALUES

(1, 12, 7, 1, '07M47684BS5725041', 'Completed'),

(2, 14, 2, 1, '07M47684BS5725041', 'Completed');

*--*

*-- Table structure for table `orders\_info`*

*--*

CREATE TABLE `orders\_info` (

  `order\_id` int(10) NOT NULL,

  `user\_id` int(11) NOT NULL,

  `f\_name` varchar(255) NOT NULL,

  `email` varchar(255) NOT NULL,

  `address` varchar(255) NOT NULL,

  `city` varchar(255) NOT NULL,

  `state` varchar(255) NOT NULL,

  `zip` int(10) NOT NULL,

  `cardname` varchar(255) NOT NULL,

  `cardnumber` varchar(20) NOT NULL,

  `expdate` varchar(255) NOT NULL,

  `prod\_count` int(15) DEFAULT NULL,

  `total\_amt` int(15) DEFAULT NULL,

  `cvv` int(5) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `orders\_info`*

*--*

INSERT INTO `orders\_info` (`order\_id`, `user\_id`, `f\_name`, `email`, `address`, `city`, `state`, `zip`, `cardname`, `cardnumber`, `expdate`, `prod\_count`, `total\_amt`, `cvv`) VALUES

(1, 12, 'Puneeth', 'puneethreddy951@gmail.com', 'Bangalore, Kumbalagodu, Karnataka', 'Bangalore', 'Karnataka', 560074, 'pokjhgfcxc', '4321 2345 6788 7654', '12/90', 3, 77000, 1234),

(4, 27, 'Mohammed Thayeeb  Shariff', 'altayyabtravels646@mail.com', 'banglorefgmnhfmxghmgmghm', 'karnataka', 'ghmghmghm', 560018, 'mohammed thayeeb', '5566 5526 3262 1516', '12/22', 2, 1010, 5),

(5, 28, 'Saba  Shariff', 'akbarmd416@gmail.com', 'banglore', 'karnataka', 'karnataka', 560020, 'saba', '5522 5523 6492 4514', '12/21', 1, 1000, 2),

(6, 29, 'Nidith SS', 'hajeerab93@gmail.com', 'banglore', 'karnataka', 'karnataka', 560020, 'nidith', '5505 6630 8890', '12/22', 4, 8600, 2);

*-- Table structure for table `order\_products`*

*--*

CREATE TABLE `order\_products` (

  `order\_pro\_id` int(10) NOT NULL,

  `order\_id` int(11) NOT NULL,

  `product\_id` int(11) NOT NULL,

  `qty` int(15) DEFAULT NULL,

  `amt` int(15) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `order\_products`*

*--*

INSERT INTO `order\_products` (`order\_pro\_id`, `order\_id`, `product\_id`, `qty`, `amt`) VALUES

(73, 1, 1, 1, 5000),

(74, 1, 4, 2, 64000),

(75, 1, 8, 1, 40000),

(91, 2, 73, 1, 3500),

(92, 3, 1, 1, 5000),

(93, 3, 3, 1, 30000),

(94, 3, 2, 1, 25000),

(95, 4, 63, 1, 550),

(96, 4, 63, 1, 460),

(97, 5, 5, 1, 1000),

(98, 6, 5, 1, 1000),

(99, 6, 2, 1, 2500),

(100, 6, 4, 1, 3200),

(101, 6, 23, 1, 1900);

*-- Table structure for table `products`*

*--*

CREATE TABLE `products` (

  `product\_id` int(100) NOT NULL,

  `product\_cat` int(100) NOT NULL,

  `product\_brand` int(100) NOT NULL,

  `product\_title` varchar(255) NOT NULL,

  `product\_price` int(100) NOT NULL,

  `product\_desc` text NOT NULL,

  `product\_image` text NOT NULL,

  `product\_keywords` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

*--*

*-- Dumping data for table `products`*

*--*

INSERT INTO `products` (`product\_id`, `product\_cat`, `product\_brand`, `product\_title`, `product\_price`, `product\_desc`, `product\_image`, `product\_keywords`) VALUES

(1, 1, 2, 'Sketch', 5000, 'Its a sketch', 'product07\_sketch.jpg', 'Sketch Elighted'),

(2, 1, 3, 'beautiful sketch', 2500, 'iphone 5s', 'product0\_sketch.jpg', 'Its a sketch'),

(3, 1, 3, 'jawaan sketch', 3000, 'its a sketch', 'product02\_sketch.jpg', 'sketch'),

(4, 1, 3, ' Astronaut sketch', 3200, 'its a astronaut sketch', 'product03\_sketch.jpg', 'Extraordinary sketches '),

(80, 1, 1, 'ytrfdkjsd', 12343, 'sdfhgh', '1542455446\_thythtf .jpeg', 'dfgh'),

(81, 4, 6, 'Painting Art - 6', 300, 'Eyes', 'product06\_painting.jpg', 'Painting Art'),

(82, 2, 2, 'bib', 100, 'ophioih', '1709295362\_a1.jpg', 'thj'),

(83, 5, 0, 'Visual arts', 5500, 'Visaul Art', '1709466144\_product0\_visual.jpg', 'Visual');

-- --------------------------------------------------------

--

-- Table structure for table `user\_info`

--

CREATE TABLE `user\_info` (

  `user\_id` int(10) NOT NULL,

  `first\_name` varchar(100) NOT NULL,

  `last\_name` varchar(100) NOT NULL,

  `email` varchar(300) NOT NULL,

  `password` varchar(300) NOT NULL,

  `mobile` varchar(10) NOT NULL,

  `address1` varchar(300) NOT NULL,

  `address2` varchar(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

--

-- Dumping data for table `user\_info`

--

INSERT INTO `user\_info` (`user\_id`, `first\_name`, `last\_name`, `email`, `password`, `mobile`, `address1`, `address2`) VALUES

(12, 'puneeth', 'Reddy', 'puneethreddy951@gmail.com', 'puneeth', '9448121558', '123456789', 'sdcjns,djc'),

(15, 'hemu', 'ajhgdg', 'puneethreddy951@gmail.com', '346778', '536487276', ',mdnbca', 'asdmhmhvbv'),

(16, 'venky', 'vs', 'venkey@gmail.com', '1234534', '9877654334', 'snhdgvajfehyfygv', 'asdjbhfkeur'),

(28, 'Saba ', 'Shariff', 'akbarmd416@gmail.com', '12345678912', '9341319646', 'banglore', 'karnataka'),

(29, 'Nidith', 'SS', 'hajeerab93@gmail.com', '0987654321', '8085896961', 'banglore', 'karnataka'),

(30, 'Thayeeb', 'Shariff', 'devgenius9211@gmail.com', '2345678912', '9341319646', 'bengaluru', 'madya prade');

--

-- Triggers `user\_info`

--

DELIMITER $$

CREATE TRIGGER `after\_user\_info\_insert` AFTER INSERT ON `user\_info` FOR EACH ROW BEGIN

INSERT INTO user\_info\_backup VALUES(new.user\_id,new.first\_name,new.last\_name,new.email,new.password,new.mobile,new.address1,new.address2);

END

$$

DELIMITER ;

-- --------------------------------------------------------

--

-- Table structure for table `user\_info\_backup`

--

CREATE TABLE `user\_info\_backup` (

  `user\_id` int(10) NOT NULL,

  `first\_name` varchar(100) NOT NULL,

  `last\_name` varchar(100) NOT NULL,

  `email` varchar(300) NOT NULL,

  `password` varchar(300) NOT NULL,

  `mobile` varchar(10) NOT NULL,

  `address1` varchar(300) NOT NULL,

  `address2` varchar(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

--

-- Dumping data for table `user\_info\_backup`

--

INSERT INTO `user\_info\_backup` (`user\_id`, `first\_name`, `last\_name`, `email`, `password`, `mobile`, `address1`, `address2`) VALUES

(12, 'puneeth', 'Reddy', 'puneethreddy951@gmail.com', '123456789', '9448121558', '123456789', 'sdcjns,djc'),

(14, 'hemanthu', 'reddy', 'hemanthreddy951@gmail.com', '123456788', '6526436723', 's,dc wfjvnvn', 'b efhfhvvbr'),

(25, 'otheruser', 'user', 'otheruser@gmail.com', 'puneeth@123', '9535688928', 'Bangalore', 'Kumbalagodu'),

(26, 'test', 'test', 'tes1t@gmail.com', '123456789', '9994949949', 'test', 'testa'),

(27, 'Mohammed Thayeeb ', 'Shariff', 'altayyabtravels646@mail.com', 'thay.2ggg', '7338219832', 'banglore', 'karnataka'),

(28, 'Saba ', 'Shariff', 'akbarmd416@gmail.com', '12345678912', '9341319646', 'banglore', 'karnataka'),

(29, 'Nidith', 'SS', 'hajeerab93@gmail.com', '0987654321', '8085896961', 'banglore', 'karnataka'),

(30, 'Thayeeb', 'Shariff', 'devgenius9211@gmail.com', '2345678912', '9341319646', 'bengaluru', 'madya prade');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `admin\_info`

--

ALTER TABLE `admin\_info`

  ADD PRIMARY KEY (`admin\_id`);

--

-- Indexes for table `brands`

--

ALTER TABLE `brands`

  ADD PRIMARY KEY (`brand\_id`);

--

-- Indexes for table `cart`

--

ALTER TABLE `cart`

  ADD PRIMARY KEY (`id`);

--

-- Indexes for table `categories`

--

ALTER TABLE `categories`

  ADD PRIMARY KEY (`cat\_id`);

--

-- Indexes for table `email\_info`

--

ALTER TABLE `email\_info`

  ADD PRIMARY KEY (`email\_id`);

--

-- Indexes for table `logs`

--

ALTER TABLE `logs`

  ADD PRIMARY KEY (`id`);

--

-- Indexes for table `orders`

--

ALTER TABLE `orders`

  ADD PRIMARY KEY (`order\_id`);

--

--

ALTER TABLE `user\_info`

  MODIFY `user\_id` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=31;

--

-- AUTO\_INCREMENT for table `user\_info\_backup`

--

ALTER TABLE `user\_info\_backup`

  MODIFY `user\_id` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=31;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `orders\_info`

--

ALTER TABLE `orders\_info`

  ADD CONSTRAINT `user\_id` FOREIGN KEY (`user\_id`) REFERENCES `user\_info` (`user\_id`);

--

-- Constraints for table `order\_products`

--

ALTER TABLE `order\_products`

  ADD CONSTRAINT `order\_products` FOREIGN KEY (`order\_id`) REFERENCES `orders\_info` (`order\_id`) ON DELETE NO ACTION ON UPDATE CASCADE,

  ADD CONSTRAINT `product\_id` FOREIGN KEY (`product\_id`) REFERENCES `products` (`product\_id`);

COMMIT;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

### 5.4 FRONT END CODE:

*<?php*

session\_start();

*?>*

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

        <meta *charset*="utf-8">

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

        <meta *name*="viewport" *content*="width=device-width, initial-scale=1">

         <!-- *The above 3 meta tags \*must\* come first in the head; any other head content must come \*after\* these tags* -->

        <title>Art Gallery</title>

        <!-- *Google font* -->

        <link *href*="https://fonts.googleapis.com/css?family=Montserrat:400,500,700" *rel*="stylesheet">

        <!-- *Bootstrap* -->

        <link *type*="text/css" *rel*="stylesheet" *href*="css/bootstrap.min.css"/>

        <!-- *Slick* -->

        <link *type*="text/css" *rel*="stylesheet" *href*="css/slick.css"/>

        <link *type*="text/css" *rel*="stylesheet" *href*="css/slick-theme.css"/>

        <!-- *nouislider* -->

        <link *type*="text/css" *rel*="stylesheet" *href*="css/nouislider.min.css"/>

        <!-- *Font Awesome Icon* -->

        <link *rel*="stylesheet" *href*="css/font-awesome.min.css">

        <!-- *Custom stlylesheet* -->

        <link *type*="text/css" *rel*="stylesheet" *href*="css/style.css"/>

        <link *type*="text/css" *rel*="stylesheet" *href*="css/accountbtn.css"/>

        <!-- *HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries* -->

        <!-- *WARNING: Respond.js doesn't work if you view the page via file://* -->

        <!--*[if lt IE 9]>*

*<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>*

*<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>*

*<![endif]*-->

    <style>

        #*navigation* {

          background: #FF4E50;  /\* *fallback for old browsers* \*/

            background: -webkit-linear-gradient(to right, #F9D423, #FF4E50);  /\* *Chrome 10-25, Safari 5.1-6* \*/

            background: linear-gradient(to right, #F9D423, #FF4E50); /\* *W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+* \*/

        }

        #*header* {

            background: #780206;  /\* *fallback for old browsers* \*/

            background: -webkit-linear-gradient(to right, #061161, #780206);  /\* *Chrome 10-25, Safari 5.1-6* \*/

            background: linear-gradient(to right, #061161, #780206); /\* *W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+* \*/

        }

        #*top-header* {

            background: #870000;  /\* *fallback for old browsers* \*/

            background: -webkit-linear-gradient(to right, #190A05, #870000);  /\* *Chrome 10-25, Safari 5.1-6* \*/

            background: linear-gradient(to right, #190A05, #870000); /\* *W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+* \*/

        }

        #*footer* {

            background: #7474BF;  /\* *fallback for old browsers* \*/

            background: -webkit-linear-gradient(to right, #348AC7, #7474BF);  /\* *Chrome 10-25, Safari 5.1-6* \*/

            background: linear-gradient(to right, #348AC7, #7474BF); /\* *W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+* \*/

          color: #1E1F29;

        }

        #*bottom-footer* {

            background: #7474BF;  /\* *fallback for old browsers* \*/

            background: -webkit-linear-gradient(to right, #348AC7, #7474BF);  /\* *Chrome 10-25, Safari 5.1-6* \*/

            background: linear-gradient(to right, #348AC7, #7474BF); /\* *W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+* \*/

        }

        .*footer-links* li a {

          color: #1E1F29;

        }

        .*mainn-raised* {

            margin: -7*px* 0*px* 0*px*;

            border-radius: 6*px*;

            box-shadow: 0 16*px* 24*px* 2*px* rgba(0, 0, 0, 0.14), 0 6*px* 30*px* 5*px* rgba(0, 0, 0, 0.12), 0 8*px* 10*px* -5*px* rgba(0, 0, 0, 0.2);

        }

        .*glyphicon*{

    display: inline-block;

    font: normal normal normal 14*px*/1 FontAwesome;

    font-size: inherit;

    text-rendering: auto;

    -webkit-font-smoothing: antialiased;

    -moz-osx-font-smoothing: grayscale;

    }

    .*glyphicon-chevron-left*:*before*{

        content:"\f053"

    }

    .*glyphicon-chevron-right*:*before*{

        content:"\f054"

    }

        </style>

    </head>

    <body>

        <!-- *HEADER* -->

        <header>

            <!-- *TOP HEADER* -->

            <div *id*="top-header">

                <div *class*="container">

                    <ul *class*="header-links pull-right">

                        <li><a *href*="#"><i *class*="fa fa-inr"></i> INR</a></li>

                        <li>*<?php*

*include* "db.php";

*if*(isset($\_SESSION["uid"])){

                                $sql = "SELECT first\_name FROM user\_info WHERE user\_id='$\_SESSION[uid]'";

                                $query = mysqli\_query($con,$sql);

                                $row=mysqli\_fetch\_array($query);

                                echo '

<div class="dropdownn">

                                  <a href="#" class="dropdownn" data-toggle="modal" data-target="#myModal" ><i class="fa fa-user-o"></i> HI '.$row["first\_name"].'</a>

                                  <div class="dropdownn-content">

                                    <a href="" data-toggle="modal" data-target="#profile"><i class="fa fa-user-circle" aria-hidden="true" ></i>My Profile</a>

                                    <a href="logout.php"  ><i class="fa fa-sign-in" aria-hidden="true"></i>Log out</a>

                                  </div>

                                </div>';

                            }*else*{

                                echo '

                                <div class="dropdownn">

                                  <a href="#" class="dropdownn" data-toggle="modal" data-target="#myModal" ><i class="fa fa-user-o"></i> My Account</a>

                                  <div class="dropdownn-content">

                                    <a href="" data-toggle="modal" data-target="#Modal\_login"><i class="fa fa-sign-in" aria-hidden="true" ></i>Login</a>

                                    <a href="" data-toggle="modal" data-target="#Modal\_register"><i class="fa fa-user-plus" aria-hidden="true"></i>Register</a>

                                  </div>

                                </div>';

                            }

*?>*

                                </li>

                    </ul>

                </div>

            </div>

            <!-- */TOP HEADER* -->

            <!-- *MAIN HEADER* -->

            <div *id*="header">

                <!-- *container* -->

                <div *class*="container">

                    <!-- *row* -->

                    <div *class*="row">

                        <!-- *LOGO* -->

                        <div *class*="col-md-3">

                            <div *class*="header-logo">

<a *href*="index.php" *class*="logo">

                                <font *style*="font-style:normal; font-size: 33px;color: aliceblue;font-family: serif">

                                        Art Gallery

                                    </font>

                                </a>

                            </div>

                        </div>

                        <!-- */LOGO* -->

                        <!-- *SEARCH BAR* -->

                        <div *class*="col-md-6">

                            <div *class*="header-search">

                                <form>

                                    <select *class*="input-select">

                                        <option *value*="0">All Categories</option>

                                        <option *value*="1"> Visual Art</option>

                                        <option *value*="1"> Sketches </option>

                                    </select>

                                    <input *class*="input" *id*="search" *type*="text" *placeholder*="Search here">

                                    <button *type*="submit" *id*="search\_btn" *class*="search-btn">Search</button>

                                </form>

                            </div>

                        </div>

                        <!-- */SEARCH BAR* -->

                        <!-- *ACCOUNT* -->

                        <div *class*="col-md-3 clearfix">

                            <div *class*="header-ctn">

                                <!-- *Cart* -->

                                <div *class*="dropdown">

                                    <a *class*="dropdown-toggle" *data-toggle*="dropdown" *aria-expanded*="true">

                                        <i *class*="fa fa-shopping-cart"></i>

                                        <span>Your Cart</span>

                                        <div *class*="badge qty">0</div>

                                    </a>

                                    <div *class*="cart-dropdown"  >

                                        <div *class*="cart-list" *id*="cart\_product">

                                        </div>

                                        <div *class*="cart-btns">

  <a *href*="cart.php" *style*="width:100%;"><i *class*="fa fa-edit"></i>  edit cart</a>

                                        </div>

                                    </div>

                                    </div>

                                <!-- */Cart* -->

                                <!-- *Menu Toogle* -->

                                <div *class*="menu-toggle">

                                    <a *href*="#">

                                        <i *class*="fa fa-bars"></i>

                                        <span>Menu</span>

                                    </a>

                                </div>

                                <!-- */Menu Toogle* -->

                            </div>

                        </div>

                        <!-- */ACCOUNT* -->

                    </div>

                    <!-- *row* -->

                </div>

                <!-- *container* -->

            </div>

            <!-- */MAIN HEADER* -->

        </header>

        <!-- */HEADER* -->

        <nav *id*='navigation'>

            <!-- *container* -->

            <div *class*="container" *id*="get\_category\_home">

            </div>

                <!-- *responsive-nav* -->

            <!-- */container* -->

        </nav>

        <!-- *NAVIGATION* -->

        <div *class*="modal fade" *id*="Modal\_login" *role*="dialog">

                        <div *class*="modal-dialog">

                          <!-- *Modal content*-->

                          <div *class*="modal-content">

                            <div *class*="modal-header">

                              <button *type*="button" *class*="close" *data-dismiss*="modal">&times;</button>

                            </div>

<div *class*="modal-body">

*<?php*

*include* "login\_form.php";

*?>*

                            </div>

                          </div>

                        </div>

                      </div>

                <div *class*="modal fade" *id*="Modal\_register" *role*="dialog">

                        <div *class*="modal-dialog" *style*="">

                          <!-- *Modal content*-->

                          <div *class*="modal-content">

                            <div *class*="modal-header">

                              <button *type*="button" *class*="close" *data-dismiss*="modal">&times;</button>

                            </div>

                            <div *class*="modal-body">

*<?php*

*include* "register\_form.php";

*?>*

                            </div>

                          </div>

                        </div>

                      </div>

# CHAPTER-6

# SCREERNSHOTS

x

### 6.1 HOME PAGE:

### 

### 

Fig 6.1 Home Page

### 6.2 LOGIN & REGISTRATION PAGE:

### 

### 

Fig 6.2 Login & Registration Page

### 6.3 ADD TO CART SECTION:

### 

### 

Fig 6.3 Add to cart Page

### 6.4 CHECKOUT SECTION:

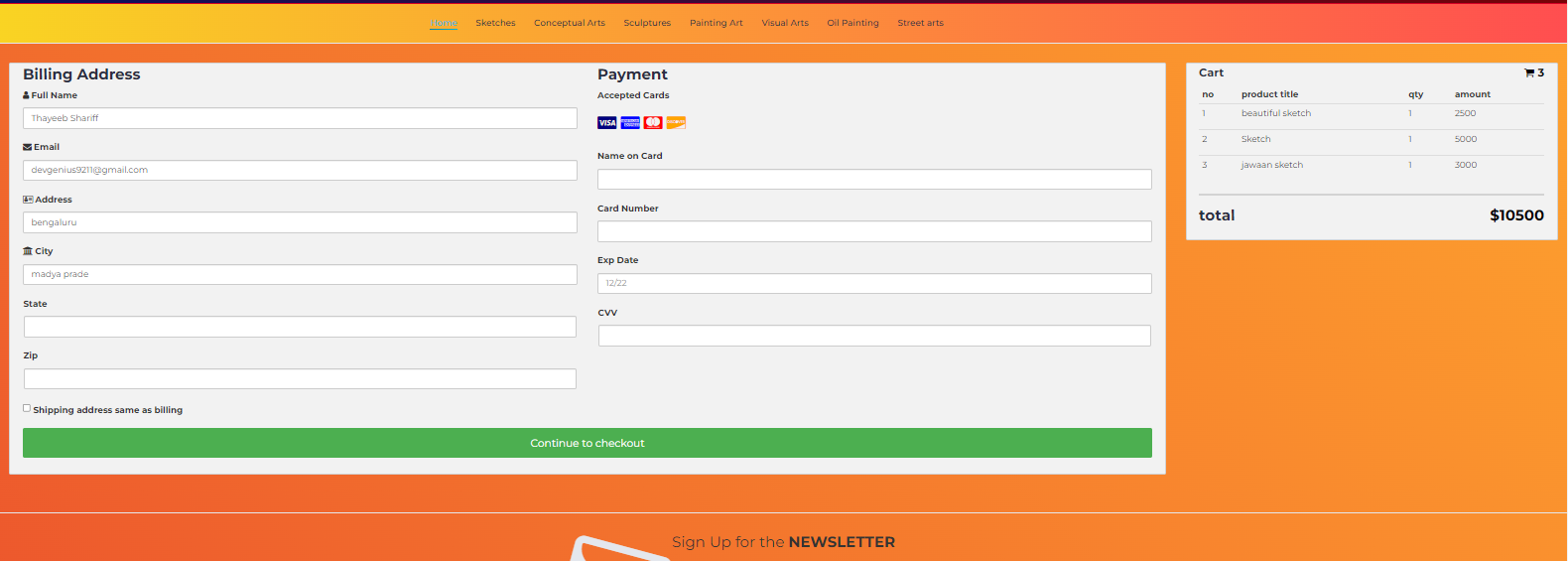


Fig 6.4 Checkout Page

### 6.5 ADMIN DASHBOARD SECTION:



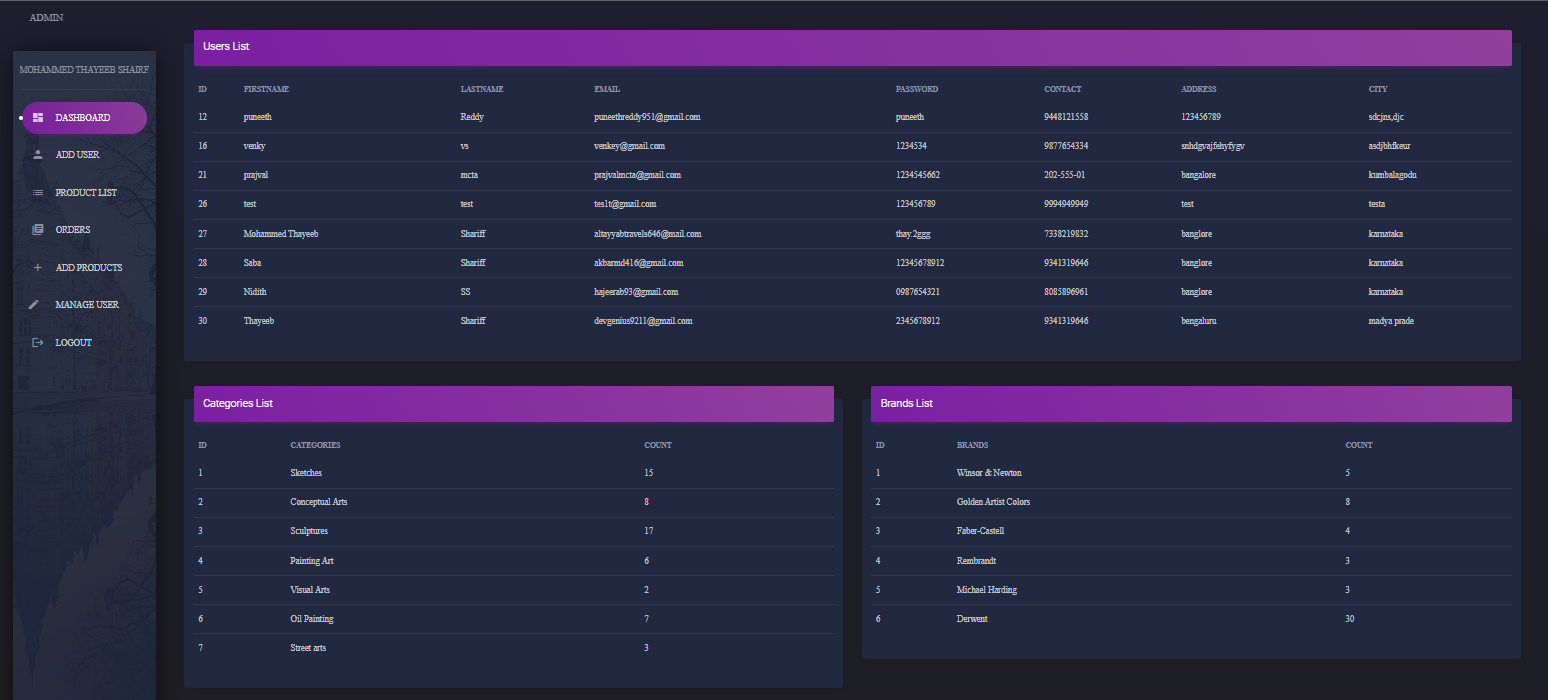


Fig 6.5 Admin Dashboard Page

### 6.6 ADD USERS:

### 

### 6.7 ADD PRODUCTS:

### 

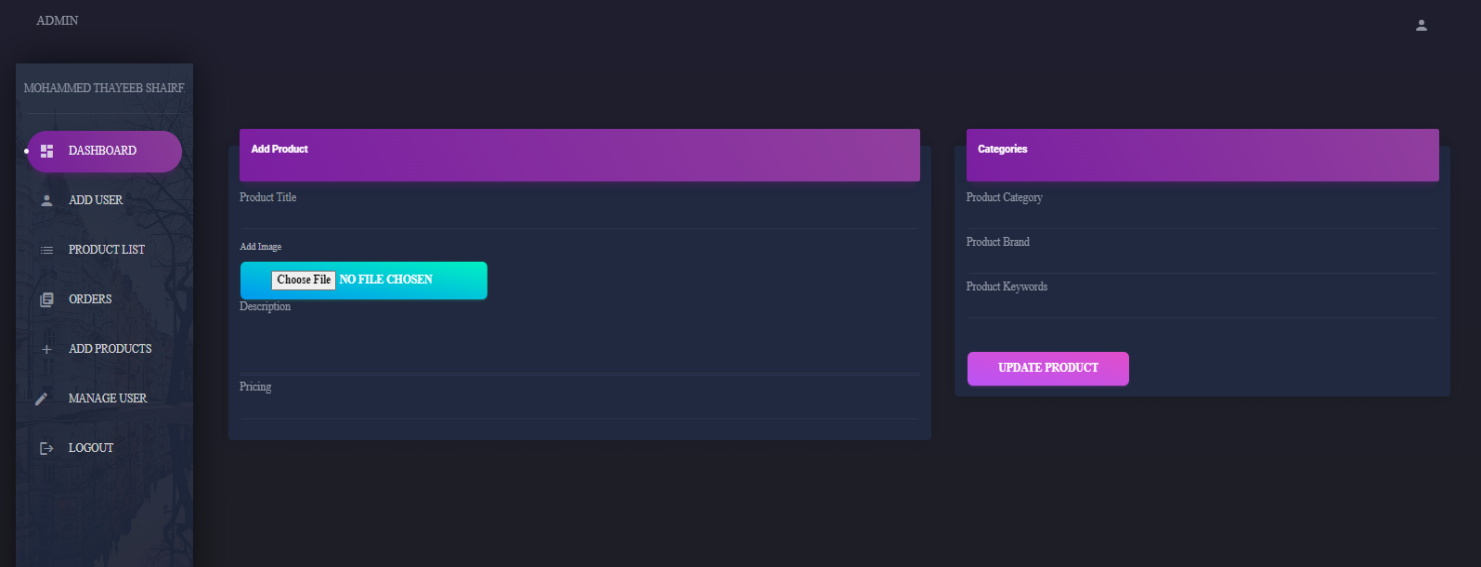


Fig 6.6 & 6.7 Add Users & Products Page

### 

### 6.8 DATABASE TABLE VIEW STRUCTURE:

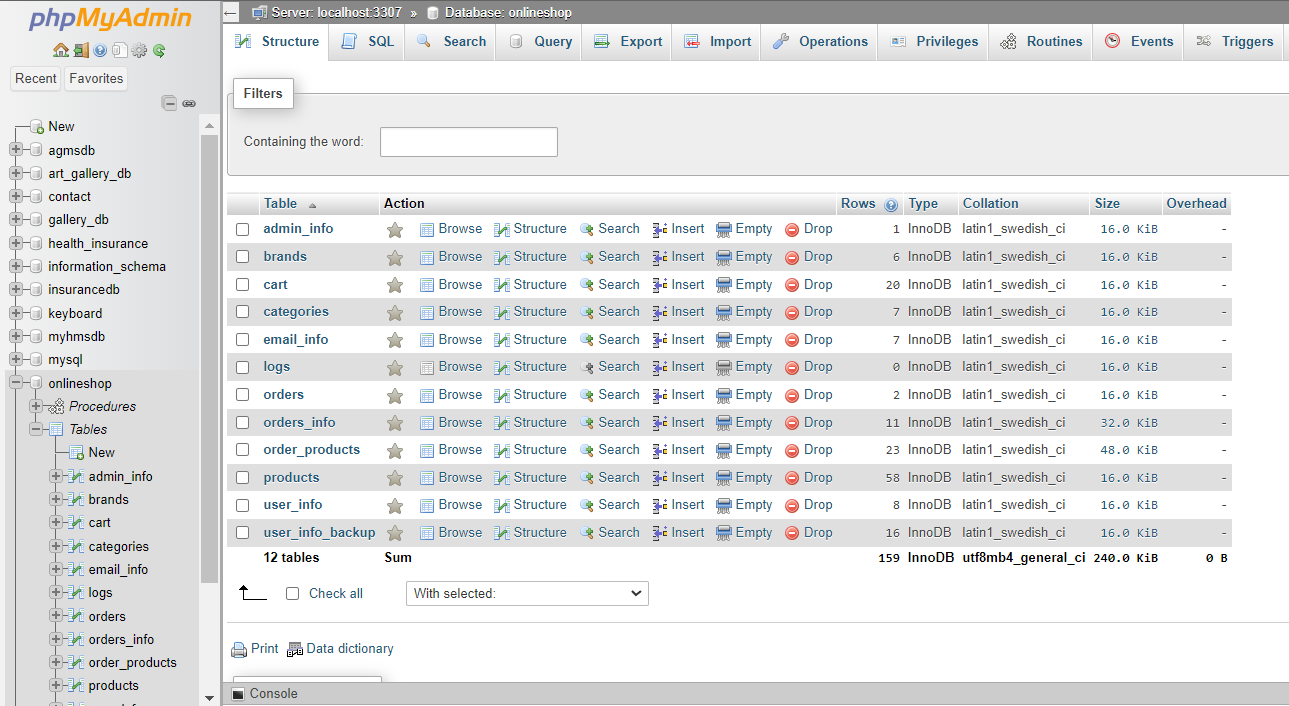
****

Fig 6.8 Database Table view structure

### 6.9 BACKEND EMAIL\_INFO DETAILS:

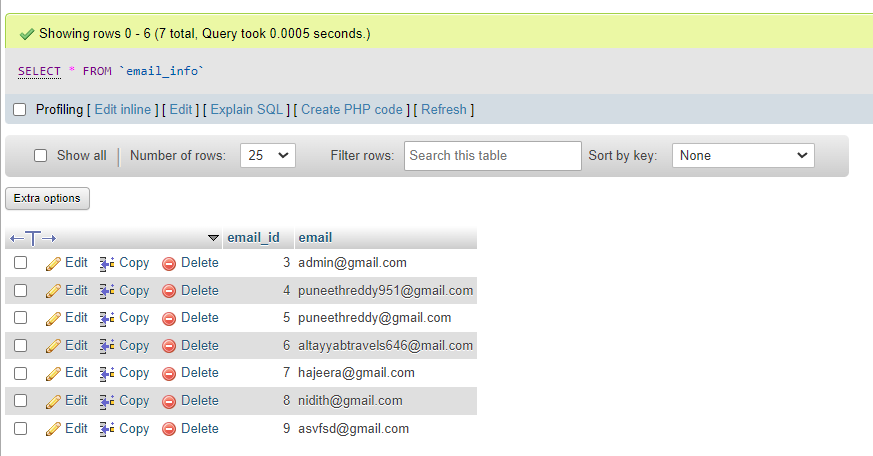


Fig 6.9 Backend Email info Details

### 6.10 BACKEND PRODUCTS DETAILS:

### 

Fig 6.10 Backend products details

### 6.11 BACKEND CATEGORIES/BRAND DETAILS:

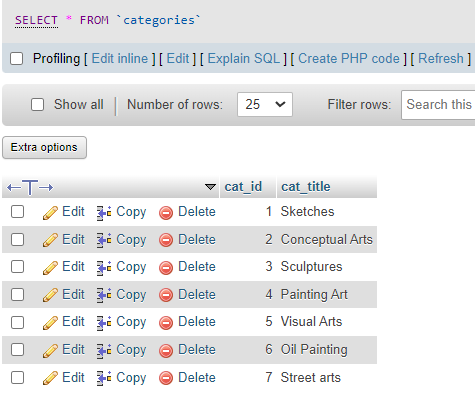
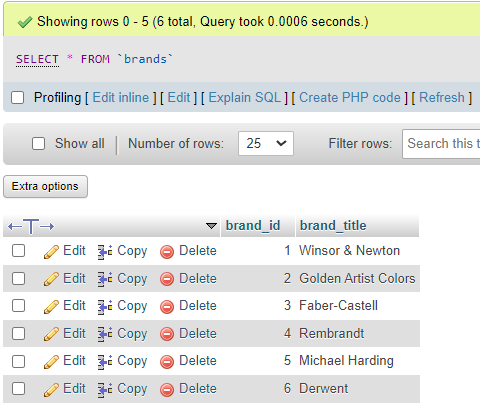
 

Fig 6.11 Backend categories/brand details

### 6.12 BACKEND USERS\_INFO:

### 

Fig 6.12 Backend Users\_info

### 6.13 BACKEND LOGIN/SIGNUP CREDENTIALS:

### 

Fig 6.13 Backend Login/Signup Credentials

### 6.14 BACKEND ORDERS DETAILS:

### 

### 6.15 BACKEND ADMIN\_INFO:

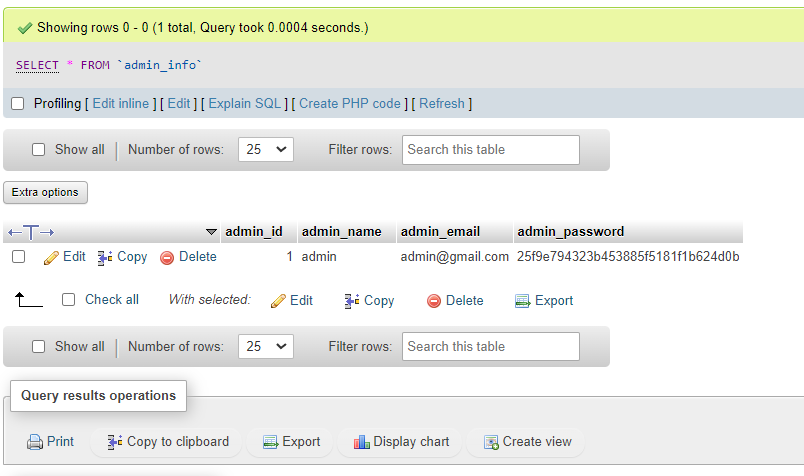


Fig 6.14 Backend Orders details & admin info

# 

### CONCLUSION:

**The Virtual Art Gallery Management System revolutionizes art appreciation by offering an immersive exploration experience. Users can engage in lifelike representations of artworks, fostering a deeper connection with the creative process. The platform boasts a diverse art collection, spanning various genres and styles, ensuring a rich and multifaceted experience for art enthusiasts.**

**One of its standout features is the encouragement of interactive engagement within a thriving art community. Users can participate in discussions, share their perspectives, and contribute to a dynamic dialogue surrounding the showcased artworks. This collaborative aspect enhances the overall user experience, making it not just a virtual gallery but a communal space for artistic discourse.**

**Moreover, the system prioritizes accessibility, providing users with the flexibility to explore the gallery at any time and from anywhere, across multiple devices. Whether on a desktop, tablet, or smartphone, art lovers can seamlessly connect with the virtual gallery, breaking down traditional barriers to art appreciation.**

**In essence, the Virtual Art Management System combines immersive exploration, a diverse art collection, interactive engagement, and flexible accessibility, creating a comprehensive platform that caters to the evolving landscape of art enjoyment and participation.**

### REFERENCES:

### https://www.geeksforgeeks.org/introduction-of-dbms-database-management- system-set-1/

### https://www.youtube.com/watch?v=2MpZwFoBPjQ

### https://github.com/vatsa287/Art-Gallery-Management-System

### https://chat.openai.com/