SCHOOL OF TECHNOLOGY

BACHELOR OF BUSINESS INFORMATION TECHNOLOGY

UNIT: BBIT 03107

ART SELLING WEBSITE MANAGEMENT SYSTEM

FINAL DOCUMENTATION

BY: NYATORO TREVOR MAINA

REG NO: 22/05844

EMAIL: 2205844@students.kcau.ac.ke

SUPERVISOR: LISPER KENDI

DECLARATION

I, **Nyatoro Trevor Maina**, hereby declare that the information presented in this Final Project Documentation is, to the best of my knowledge, original and accurate. This work has been completed independently, except where duly acknowledged.

Signature..... Date.....

Student: NYATORO TREVOR MAINA

REG NUMBER: 22/05844

As the student's supervisor, I approve the submission of this project report.

Supervisor: LISPER KENDI

ACKNOWLEDGEMENT

I would like to thank God for giving me the strength to complete this project. I am deeply grateful to my supervisor, **Lisper Kendi**, for her unwavering commitment, invaluable patience, and constant encouragement. Her insightful feedback and meticulous guidance were instrumental in shaping this work and ensuring its successful completion.

Contents

I. INTRODUCTION

1.1 Purpose

The purpose of this document is to provide a comprehensive overview of the **Art Invasion Gallery Management System**, a web application designed to bridge the gap between artists, buyers, and art enthusiasts. This system aims to offer a seamless platform for managing and showcasing art while ensuring an easy-to-use interface for users with varying levels of technical knowledge. The documentation will serve as a guide for developers, administrators, and users of the system.

1.2 Scope

The **Art Invasion Gallery Management System** is an online platform built using **PHP**, **MySQL**, and a **responsive web interface** utilizing **HTML**, **JavaScript**, **AJAX**, and **jQuery**. The system focuses on managing art products and inquiries within an art gallery. It provides distinct modules for **admins** and **users**.

The **admin module** allows administrative tasks such as managing art types, art products, artist profiles, and user inquiries, while the **user module** provides access to art types and products with the ability to make inquiries. The system is built to run on **XAMPP** or other similar software, compatible with web browsers such as Mozilla, Google Chrome, Internet Explorer, and Opera.

This platform is designed to remove the complexities of traditional manual systems in art galleries, focusing on enhancing efficiency, user-friendliness, and transparency.

1.3 Objectives

The primary objectives of the **Art Invasion Gallery Management System** are:

- **Seamless User Experience:** A user-friendly interface that does not require any formal knowledge to operate, ensuring users of all technical levels can navigate the system with ease.
- **Efficient Management:** Empower gallery administrators with tools to manage art products, categories, and artist profiles, as well as handle user inquiries.
- **Error Minimization:** The system ensures that invalid data is caught, and appropriate messages are displayed to guide the user through proper data entry.
- **Data Transparency:** Provide a transparent environment for art management that helps both users and administrators view and manage data accurately and efficiently.

1.4 Significance

The **Art Invasion Gallery Management System** aims to modernize the traditional art gallery system. By leveraging modern web technologies and databases, the system ensures streamlined management of art products and customer inquiries. Additionally, it provides a more interactive experience for users, enabling them to explore various art types and interact with art galleries effectively.

The project not only addresses the challenges faced by traditional gallery systems but also offers a scalable solution for future enhancements, including the addition of payment integrations and mobile app support.

1.5 Overview of the Document

This document contains the following sections:

- **System Requirements Specifications (SRS):** Details the functional and non-functional requirements, including user characteristics, system constraints, and assumptions.
- **System Design Specifications (SDS):** Describes the system architecture, database design, and technical design aspects.
- **Test Plan:** Provides test cases, testing steps, expected outcomes, and the testing environment details to ensure system reliability.
- **User Implementation Guide:** Guides users through using the system, detailing both the admin and user modules with step-by-step instructions and visuals.

II. System Requirement Specifications (SRS) Document for Art Invasion

---

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive description of the functional and non-functional requirements for the Art Invasion platform. It will guide the development team in implementing a reliable, secure, and user-friendly system for online art commerce.

1.2 Scope

Art Invasion is an online platform designed to connect artists, buyers, and art enthusiasts. The system will provide features such as artist profiles, artwork listings, secure transactions, and artwork authentication. The platform will operate initially in Kenya and expand globally. The core goal is to enhance transparency and accessibility in the art market while fostering trust between buyers and sellers.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** System Requirements Specification
- **UI:** User Interface
- **DBMS:** Database Management System
- **API:** Application Programming Interface

1.4 References

- Johnson, D. (2019). "The Evolution of the Online Art Market: A Case Study of Saatchi Art."
- Kim, S. (2020). "Etsy and the Democratization of Art Commerce."
- Thomas, L. (2021). "Artfinder's Journey in the Competitive Art Market."

1.5 Overview

This document details the functional and non-functional requirements, interface designs, system constraints, and development assumptions for the Art Invasion platform.

---

2. Overall Description

2.1 Product Perspective

Art Invasion will be a standalone system, integrated with APIs for features like artwork authentication and personalized recommendations. It will have:

- A responsive web interface for desktops and mobile devices.
- Future provisions for mobile app support using Flutter.

2.2 Product Features

- **Artist Profiles:** Create and manage detailed profiles showcasing portfolios.
- **Artwork Listings:** Upload, categorize, and sell artworks.
- **Artwork Authentication:** Verify artwork originality.
- **Personalized Recommendations:** AI-driven suggestions based on user preferences.

2.3 User Characteristics

The platform will serve:

- **Artists:** Tech-savvy and non-tech-savvy users needing simple tools.
- **Buyers:** Individuals who prioritize transparency and trust.
- **Art Enthusiasts:** Users casually browsing for art.
- **Interior Designers and Galleries:** Professional users requiring curated collections.

2.4 Constraints

- **Initial Hosting Limitations:** Hosted on TrueHost/Safaricom.
- **Budget Constraints:** Limited funds for advanced features like AI integrations in the first phase.
- **High Dependency on Secure and Scalable Payment APIs:** Payment integration and secure transaction processing are critical.

2.5 Assumptions and Dependencies

- **Internet Availability:** Required for users to access the platform.
- **Third-Party APIs:** Timely integration of APIs for payment and artwork authentication.
- **Local Art Community Support:** Collaboration with local artists for onboarding during the initial phase.

---

3. Functional Requirements

- **FR1:** Users shall be able to create accounts as artists, buyers, or administrators.
- **FR2:** Artists shall upload and manage artworks, including adding descriptions, categories, and pricing.
- **FR3:** Buyers shall search and filter artworks by category, price range, and artist location.
- **FR4:** The system shall authenticate artwork originality using image recognition APIs.
- **FR5:** Users shall receive notifications about purchases, updates, and recommendations.

---

4. Non-Functional Requirements

- **NFR1:** The system shall ensure 99.9% uptime.
- **NFR2:** The platform shall support at least 10,000 simultaneous users.
- **NFR3:** All user data shall be encrypted during transmission and storage.
- **NFR4:** The UI shall be responsive and load within 3 seconds on standard connections.
- **NFR5:** The system shall adhere to GDPR and Kenyan data protection regulations.

---

5. Interface Requirements

5.1 User Interfaces

- **Artist Dashboard:** A clean and intuitive UI for artist profiles and artwork management.
- **Buyer Dashboard:** For browsing artworks, purchasing, and viewing recommendations.
- **Admin Dashboard:** For monitoring platform activity, managing user accounts, and reviewing reports.

5.2 Hardware Interfaces

- The platform shall operate on standard servers (in Hosting companies).
- **Future Scalability:** Cloud hosting will be implemented if the user base exceeds initial expectations.

5.3 Software Interfaces

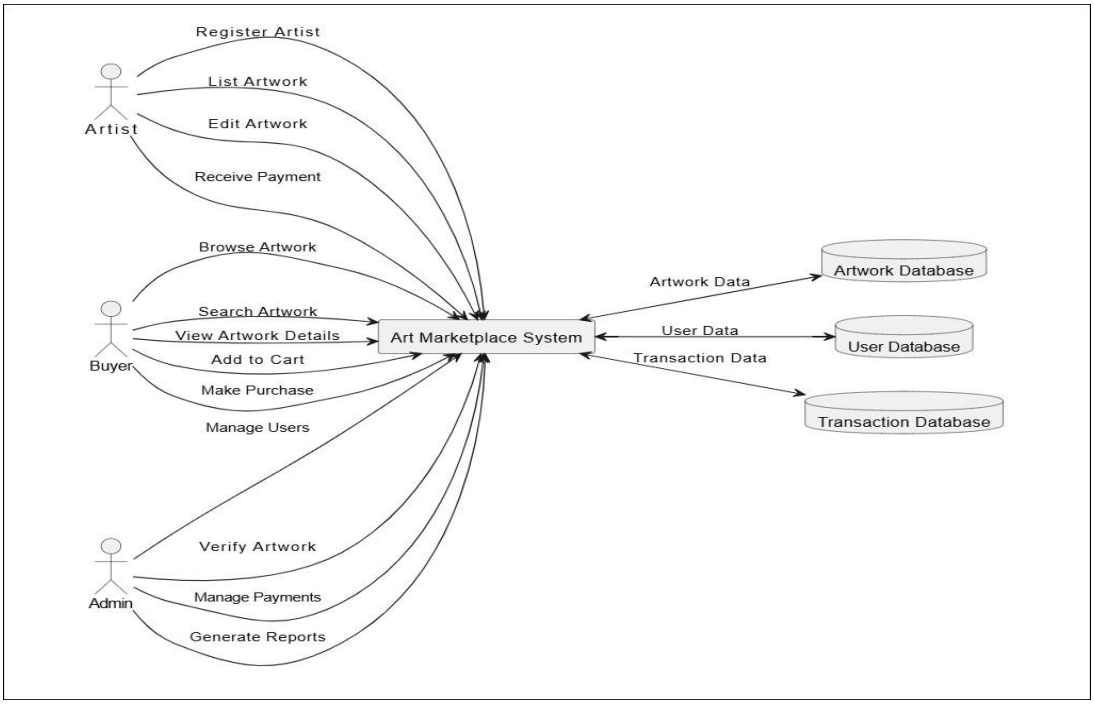
- **MySQL Database:** For storing user profiles, artwork details, and stock inventory.
- **AI Tools:** For personalized recommendations and image recognition of artwork.

Next Page – System Models...

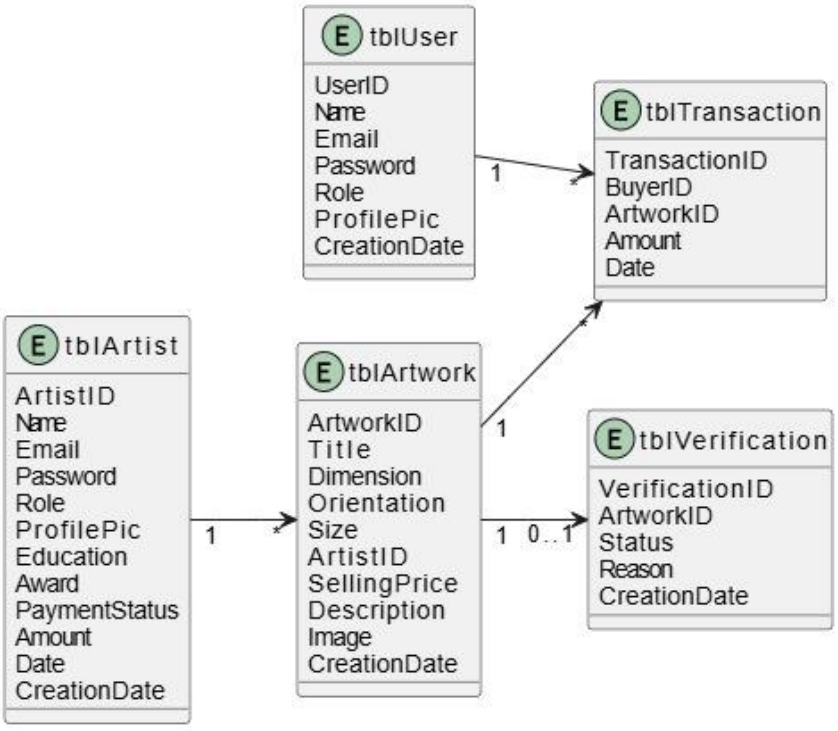
6. System Models

High-level diagrams such as Data Flow Diagrams (DFDs), Entity-Relationship Diagrams (ERDs), and Use Case diagrams are included here.

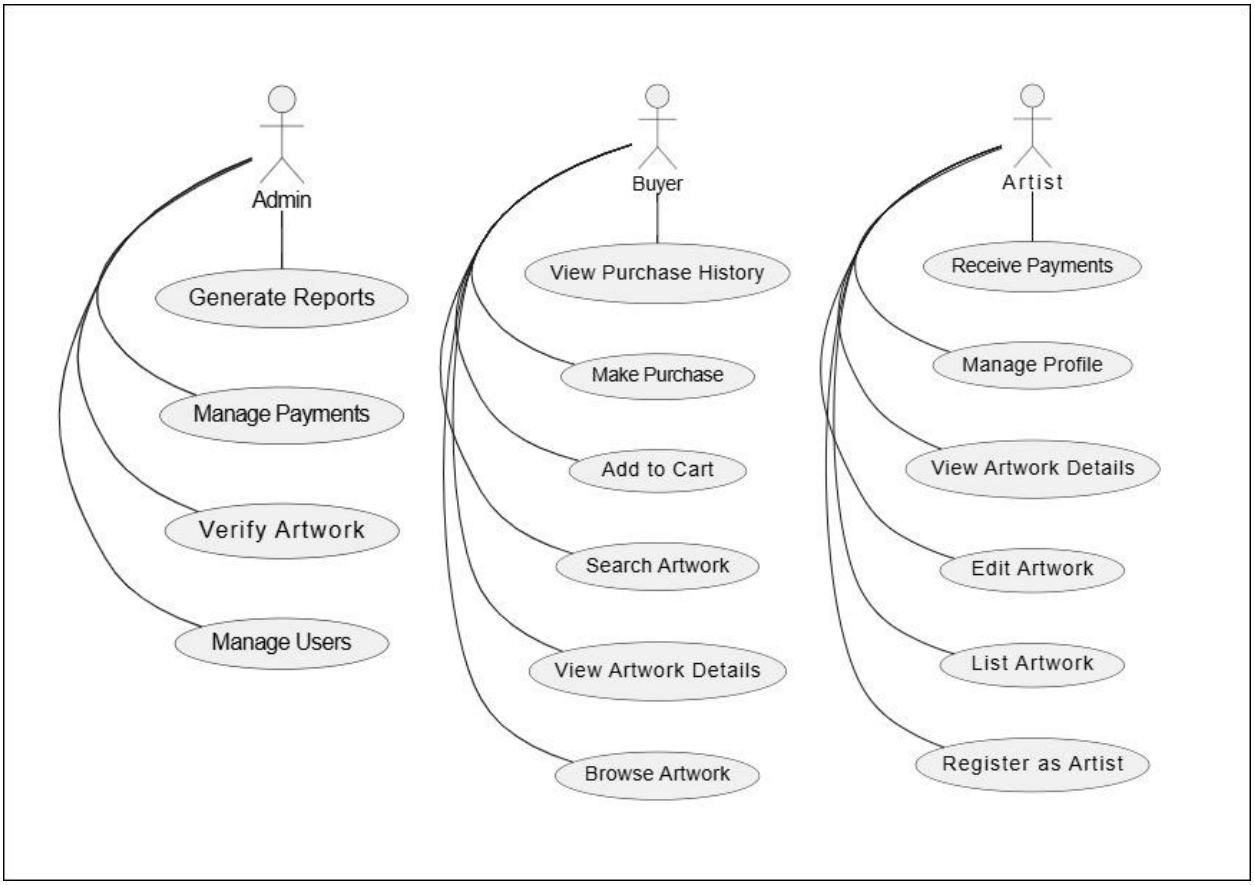
Data Flow Diagram:



Entity-Relationship Diagram:



Use Case Diagram:



---

7. Other Requirements

- **Regulatory Compliance:** The system shall comply with local and international e-commerce and art trade regulations.
- **Documentation:** Comprehensive user manuals and final documentation are to be attached.

III. System Design Specification (SDS) for Art Invasion

---

1. Introduction

1.1 Purpose

The purpose of this document is to outline the design of the Art Invasion platform, describing its architecture, components, and how it will fulfill the requirements specified in the System Requirements Specification (SRS). It serves as a blueprint for the development and implementation of the platform.

1.2 Scope

This system will allow artists to create profiles, list artworks, and facilitate secure transactions with buyers. It will address challenges such as limited market access for local artists, lack of global exposure, and issues of trust between buyers and sellers. The platform will operate initially in Kenya and expand globally.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** System Requirements Specification
- **UI:** User Interface
- **DBMS:** Database Management System
- **API:** Application Programming Interface

---

2. System Architecture

The system architecture for Art Invasion is designed to be a scalable and secure platform with the following components:

- **Frontend (Client-side):**
  - Developed using **ReactJS** for dynamic, responsive web pages.
  - For future purposes, Mobile support will be added using **Java** and **Swift** for future mobile app development.
- **Backend (Server-side):**
  - Built using **PHP**, enabling efficient handling of multiple requests and real-time features.
  - **MySQL** will be used as the database management system for storing user and artwork data.

---

3. Module Design

The system is divided into the following key modules:

3.1 User Module

- **Artists:**
  - **Account Creation:** Artists will create profiles and upload artworks, including details like category, price, description, and images.
  - **Artwork Management:** Artists can edit, delete, or update their listings at any time.
  - **View Transactions:** Artists can view their transaction history and monitor sales.
- **Buyers:**
  - **Account Creation:** Buyers will create profiles to manage their purchases and preferences.
  - **Artwork Discovery:** Buyers will search and filter artworks based on various parameters such as price, category, artist, and location.

3.2 Authentication Module

- **Artwork Verification:** Use image recognition APIs for verifying the authenticity of artworks before they are listed.
- **Profile Verification:** Authentication of user identities to ensure credibility.

3.3 Recommendation Module

- **Personalized Recommendations:** Ad-driven system that suggests artworks to buyers based on their past behavior, interests, and preferences.

3.4 Admin Module

- **Dashboard:** Admins can monitor user activity, review transactions, approve artwork listings, and manage users.
- **Reporting:** Admins will have access to platform performance and transaction reports.

---

4. Database Design

The database design for Art Invasion includes the following main entities:

1. tblArtist

i. ArtistID (int): Primary Key
ii. Name (varchar(250))
iii. Email (varchar(250))
iv. Password (varchar(250))
v. Role (enum('Artist', 'Admin'))
vi. ProfilePic (varchar(250))
vii. Education (mediumtext)
viii. Award (mediumtext)
ix. PaymentStatus (enum('Pending', 'Processed'))
x. Amount (decimal(10,2))
xi. Date (timestamp)
xii. CreationDate (timestamp)

2. tblArtwork

- ArtworkID (int): Primary Key
- Title (varchar(250))
- Dimension (varchar(250))
- Orientation (enum('Landscape', 'Portrait'))
- Size (enum('Medium', 'Large'))
- ArtistID (int): Foreign Key referencing tblArtist.ArtistID
- SellingPrice (decimal(10,2))
- Description (mediumtext)
- Image (varchar(250))
- CreationDate (timestamp)

3. tblUser

- UserID (int): Primary Key
- Name (varchar(250))
- Email (varchar(250))
- Password (varchar(250))
- Role (enum('Artist', 'Buyer', 'Admin'))
- ProfilePic (varchar(250))
- CreationDate (timestamp)

4. tblTransaction

- TransactionID (int): Primary Key
- BuyerID (int): Foreign Key referencing tblUser.UserID
- ArtworkID (int): Foreign Key referencing tblArtwork.ArtworkID
- Amount (decimal(10,2))
- Date (timestamp)

5. tblVerification

- VerificationID (int): Primary Key
- ArtworkID (int): Foreign Key referencing tblArtwork.ArtworkID
- Status (enum('Pending', 'Approved', 'Rejected'))
- Reason (text)
- CreationDate (timestamp)

Relationships:

- **One-to-many:** One artist can have many artworks. (tblArtist to tblArtwork)
- **One-to-many:** One buyer can have many transactions. (tblUser to tblTransaction)
- **One-to-many:** One artwork can be involved in many transactions (e.g., if resold, reproduced, or relisted). (tblArtwork to tblTransaction)

---

5. User Interface Design

The system will feature an intuitive, clean, and responsive user interface:

5.1 Admin Dashboard

- **Profile Management:** Allows artists to update their profiles and manage their artworks.
- **Artwork Listings:** A list of artworks they have uploaded with the option to update or delete.

5.2 Buyer Homepage

- **Artwork Discovery:** A clean interface for browsing and filtering artworks by category, price, and artist.
- **Purchase Flow:** Simple, clear steps for purchasing artwork with payment options.
- **Recommendation Section:** AI-powered recommendations based on user preferences.

---

6. Security Design

6.1 Authentication and Authorization

- **Login/Signup:** Secure user login using email and password.
- **Role-based Access:** Different access levels for artists, buyers, and administrators.

6.2 Data Encryption

- All sensitive data (e.g., passwords, payment information) will be encrypted using industry-standard algorithms like **AES** and **SHA-256**.

6.3 Payment Security

- Payments will be handled securely, through the artist directly, which ensures robust security measures, including impersonation and fraud detection.

---

7. Performance Considerations

- **Scalability:** The platform will support at least 10,000 simultaneous users initially, with future scalability to handle increased traffic.
- **Load Testing:** Performance testing to ensure the platform responds in under 3 seconds under normal load.
- **Reliability:** The platform will ensure 99.9% uptime, leveraging a robust hosting provider (examples like TrueHost/Safaricom/Hostinger/Dreamhost hosting services).

---

8. Error Handling and Logging

- **Error Management:** Graceful handling of errors with appropriate error messages shown to the user.
- **Logging:** All transactions and errors will be logged for debugging and auditing purposes.

---

9. Testing Strategy

- **Unit Testing:** Ensure individual modules work as expected.
- **Integration Testing:** Test the interactions between modules and third-party integrations.
- **System Testing:** Ensure the entire platform functions smoothly and meets all requirements.

---

10. Deployment and Maintenance

- **Hosting:** The platform will be hosted on **TrueHost/Safaricom**, with provisions for scaling if needed.
- **Maintenance:** Regular updates and patches will be applied to ensure system stability and security.

IV. Test Plan Document for Art Invasion Gallery Management System

---

1. Introduction

1.1 Purpose:

The purpose of this Test Plan document is to outline the strategy, objectives, scope, resources, schedule, and criteria for testing the **Art Invasion Gallery Management System**. This document ensures the successful testing of the application's modules, including the admin and user functionalities, with a focus on ensuring usability, security, and error-free operation.

1.2 Scope:

This test plan will cover the following features:

- **Admin Module:** Dashboard, Art Type management, Art Medium management, Art Product management, Enquiry management, and page management (About Us, Contact Us).
- **User Module:** Home page, Art Type view, and Art Product enquiry submission.
- **Data Validation:** Ensuring that invalid data entry is handled and appropriate messages are displayed.
- **UI and Compatibility:** Testing compatibility with Mozilla, Chrome, IE8, Opera, and ensuring the application is user-friendly.

1.3 Objectives:

The main objectives of the testing phase include:

- Validating the functionalities in both the Admin and User modules.
- Ensuring the application handles data input errors gracefully and displays relevant messages.
- Verifying compatibility with major web browsers.
- Ensuring that the system does not require formal knowledge for users to operate and that it is intuitive.

1.4 Assumptions and Dependencies:

- Testing will be conducted on a local server using **XAMPP** or equivalent environments with **PHP7.x+** and **MySQL5.x+**.
- The application will be tested on the supported browsers (Mozilla, Google Chrome, IE8, Opera).
- The system must perform well even with minimal user training, as no formal knowledge is required to operate.

---

2. Test Strategy

2.1 Testing Levels:

The following levels of testing will be performed:

- **Unit Testing:** Testing individual components (admin and user modules) such as the Art Type management, Art Product management, and the Enquiry handling system.
- **Integration Testing:** Ensuring all components (Art Type, Art Medium, Art Product, Enquiry management) work together.
- **System Testing:** Validating the end-to-end functionality of the entire system, including user interaction and data flow.
- **Acceptance Testing:** Ensuring that the system meets business requirements and is ready for deployment.

2.2 Types of Testing:

- **Functional Testing:** To test if all features are working according to the requirements.
- **Security Testing:** Ensuring data validation is robust and no unauthorized access is possible.
- **Usability Testing:** Ensuring that the interface is user-friendly and easy to operate without prior formal knowledge.
- **Compatibility Testing:** Verifying that the application runs smoothly on all supported browsers.
- **Performance Testing:** Ensuring the system performs optimally under typical load scenarios.

2.3 Testing Approach:

- Testing will primarily be **manual** for UI and functional features.
- Automated backend testing will be used where applicable, such as data input validation.
- **AJAX, JQuery**, and JavaScript features will be tested to ensure smooth client-side interactions.

---

3. Test Deliverables

- **Test Cases:** Detailed test cases for each module and feature, including input validation.
- **Test Logs:** Logs documenting each test case's execution.
- **Test Reports:** Summarized reports indicating which features passed and failed the tests.
- **Defect Reports:** Documentation of any identified defects, including steps to reproduce, severity, and resolution status.

---

4. Test Scope

4.1 In-Scope:

- **Admin Module:**
  - Dashboard summary functionality.
  - Art Type, Art Medium, and Art Product management (add/update/delete).
  - Enquiry management (view/search).
  - Profile and password management.

- **User Module:**
  - Home page functionality.
  - Viewing art products based on art type.
  - Submitting enquiries for art products.
- **Data Validation:** Ensure correct and informative error messages are displayed when invalid data is entered.
- **UI Testing:** Confirm that all pages are user-friendly and functional without requiring formal knowledge.

4.2 Out-of-Scope:

- Payment API integration, as this will not be part of this phase.

---

5. Test Environment

5.1 Hardware:

Testing will be performed on:

- Local or staging servers with **XAMPP** or any equivalent PHP and MySQL environment.
- Testing will be conducted on standard desktop computers and laptops.

5.2 Software:

- **Web Server:** Apache or Nginx for hosting.
- **PHP Version:** PHP 7.x or higher.
- **Database:** MySQL 5.x or higher.
- **Web Browsers:** Mozilla, Google Chrome, IE8, and Opera.
- **Testing Tools:** Selenium for backend testing (where applicable), and browser developer tools for UI testing.

5.3 Network:

- A stable internet connection will be required for testing on staging servers.
- Local network testing will be conducted for initial stages.

---

6. Test Schedule

6.1 Testing Milestones:

- **Test Case Design Completion:** 11-11-2024
- **Initial Testing Phase:** 12-11-2024 to 14-11-2024
- **Final Testing Phase:** 22-11-2024 to 23-11-2024
- **Test Report Submission:** 02-12-2024

6.2 Test Execution Schedule:

Testing will occur at the end of each development sprint, with more intensive testing in the final phase once the system is fully integrated.

---

7. Test Resources

7.1 Test Team:

- **Test Lead:** Oversees the testing process and ensures the test plan is followed.
- **Testers:** Execute the test cases, log results, and report bugs.
- **Developers:** Fix defects identified during testing and provide support.

7.2 Tools and Equipment:

- **Browser Developer Tools** for front-end testing.
- **PHPUnit** for unit testing backend functionalities.

7.3 Budget and Resource Requirements:

Testing will use existing infrastructure and tools without additional costs.

---

8. Risk Management

Potential risks and mitigation strategies:

- **Risk:** Incomplete or buggy functionalities due to rushed development.
  - **Mitigation:** Ensure thorough testing after each sprint, with proper regression testing.
- **Risk:** Browser compatibility issues (older versions of browsers may cause problems).
  - **Mitigation:** Test across all supported browsers and ensure fallback for older versions (e.g., IE8).
- **Risk:** Incorrect data validation leading to incorrect data entry.
  - **Mitigation:** Focus on thorough testing of input fields and ensure proper error messages.

---

9. Test Case Design

9.1 Test Case Format:

Test cases will include:

- **Test Case ID**
- **Test Case Description**
- **Preconditions**
- **Test Steps**
- **Expected Results**
- **Actual Results**
- **Pass/Fail Status**

Example test cases will be created for each major functionality, such as:

- **Admin adding/updating/deleting Art Types and Mediums.**
- **User submitting an enquiry for an artwork.**
- **Validating input fields to ensure only correct data is accepted.**

---

10. Entry and Exit Criteria

10.1 Entry Criteria:

- All features have been developed and integrated.
- Test environment is set up and functional.
- Test cases are prepared.

10.2 Exit Criteria:

- All test cases have been executed, with no critical defects remaining.
- All major functionalities (Admin and User modules) are working as expected.
- Test results indicate that the system is ready for deployment.

---

11. Metrics and Reporting

Metrics:

- Number of test cases executed.
- Number of defects reported and resolved.
- Pass/Fail rate of test cases.
- Time spent on testing each feature.

Test Reporting:
Test reports will be provided after each major testing phase, summarizing the results and detailing any critical issues or blockers.

---

12. Approval and Sign-Off

Once the test plan is reviewed and all criteria are met, the test plan will be signed off by the project stakeholders and the development team.

V. User Implementation Document for Art Invasion Gallery Management System

---

1. Introduction

1.1 Purpose:
This document provides instructions for users on how to implement and operate the **Art Invasion Gallery Management System**. It is intended for both **Admin** and **User** roles to understand their respective features, functionalities, and how to efficiently use the system.

1.2 Audience:
The document is intended for the following:

- **Admin Users** (responsible for managing art types, mediums, products, and enquiries).
- **End Users** (interested in browsing art products and submitting enquiries).
- **Technical Team** (for maintaining the system and troubleshooting issues).

---

2. System Overview

2.1 Description:
The **Art Invasion Gallery Management System** is an online platform built using **PHP** and **MySQL**. The system provides functionalities for managing art products, art types, and handling user enquiries. The platform has two main roles:

- **Admin:** Manages the backend of the platform (art types, art mediums, products, and enquiries).
- **User:** Browses art products and submits enquiries.

2.2 Key Features:

- **Admin Module:** Dashboard, Art Type management, Art Medium management, Art Product management, Enquiry handling.
- **User Module:** Art browsing and enquiry submission.

3. System Requirements

- **PHP:** Version 7.x or higher.
- **MySQL Database:** Version 5.x.
- **Web Browser:** Mozilla Firefox, Google Chrome, Internet Explorer 8 (or higher), Opera.
- **Server:** XAMPP or equivalent.

4. Installation and Setup

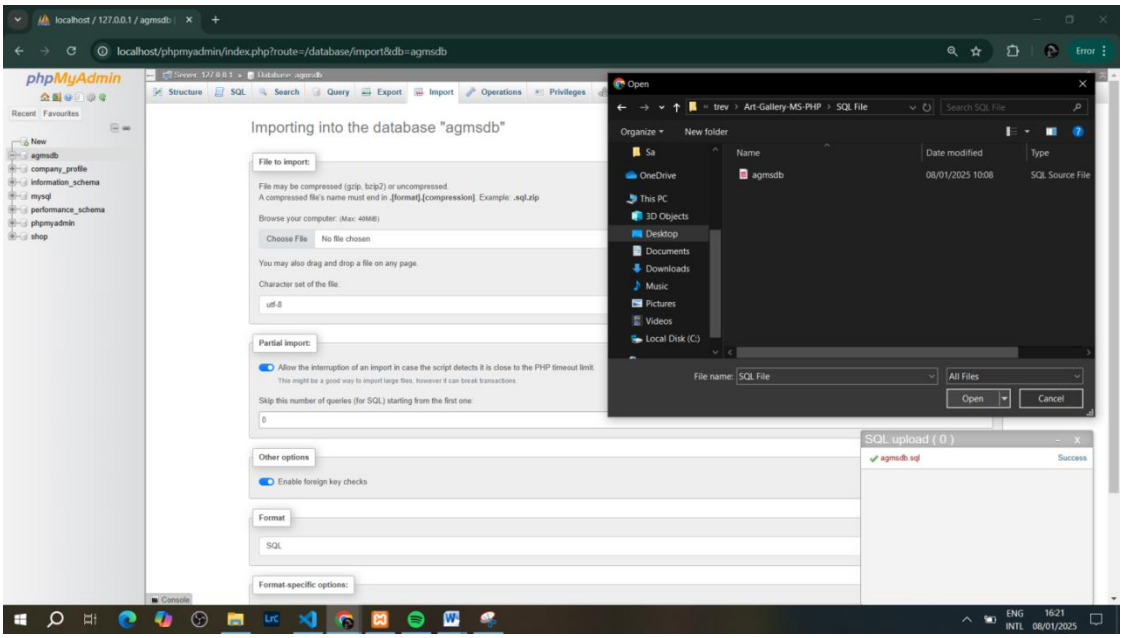
4.1 Prerequisites:

Before setting up the system, ensure you have the following installed:

- **XAMPP** or another PHP server environment.
- **MySQL** for database management.
- A modern **web browser**.

4.2 Steps to Install:

1. **Download XAMPP** from the official website (https://www.apachefriends.org/index.html).
2. **Install XAMPP** and start the Apache and MySQL services.
3. **Download the Art Invasion Gallery Management System** code.
4. **Extract** the downloaded folder to the htdocs directory of your XAMPP installation.
5. **Set up the database**:
    - Open phpMyAdmin (usually accessible at http://localhost/phpmyadmin).
    - Create a new database named agmsdb.
    - Import the provided SQL file to set up the required tables and data.

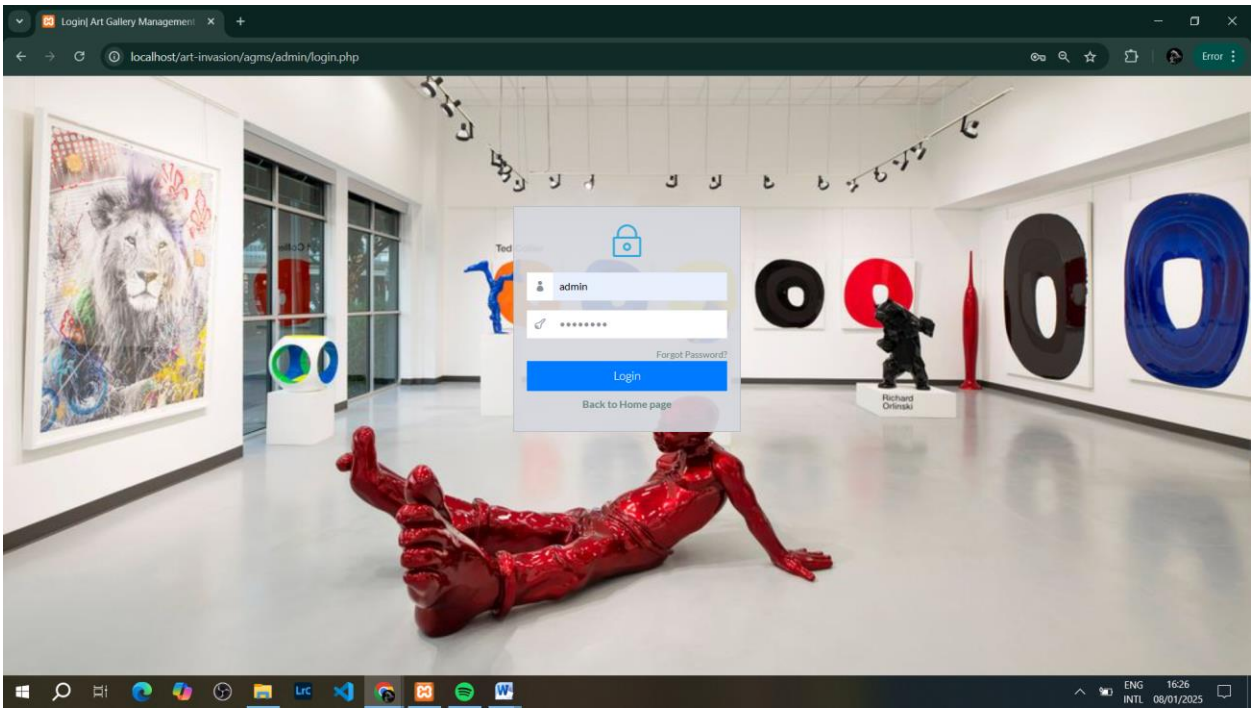


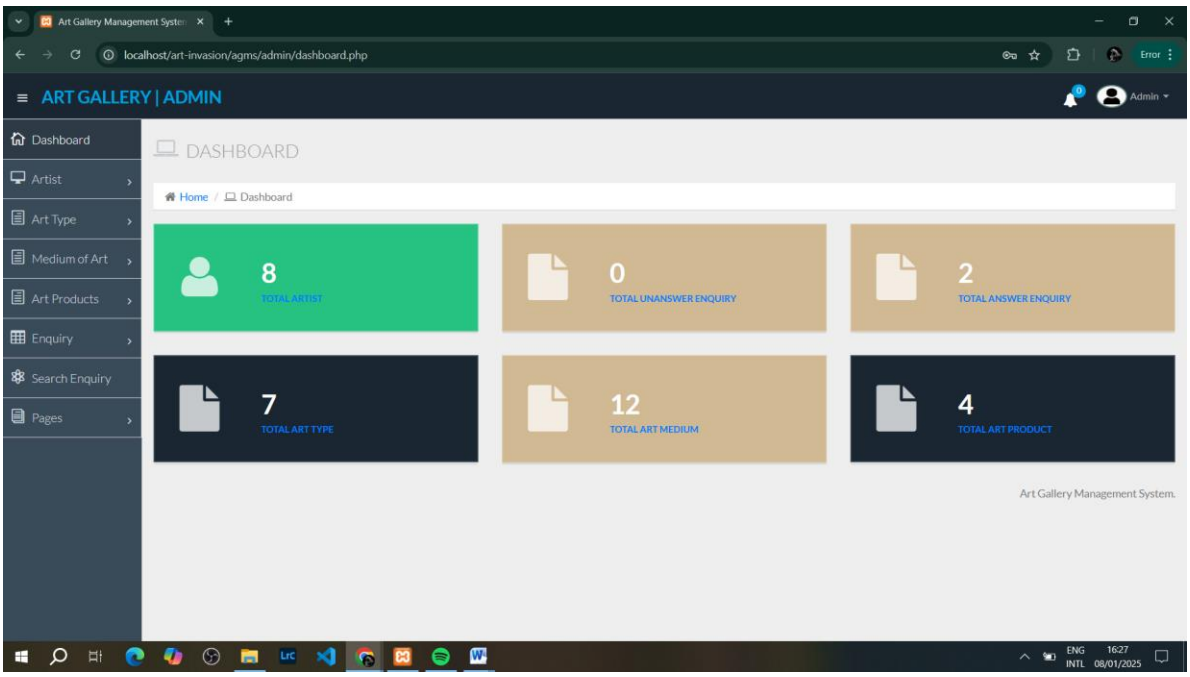
5. Admin Module Instructions

5.1 Admin Dashboard Overview:

The admin dashboard provides an overview of important metrics such as the number of artists, art types, mediums, products, and user enquiries.
To access the admin dashboard:

1. Open the web browser and go to http://localhost/art-invasion/agms/admin/login.php
2. Log in with the **admin credentials**

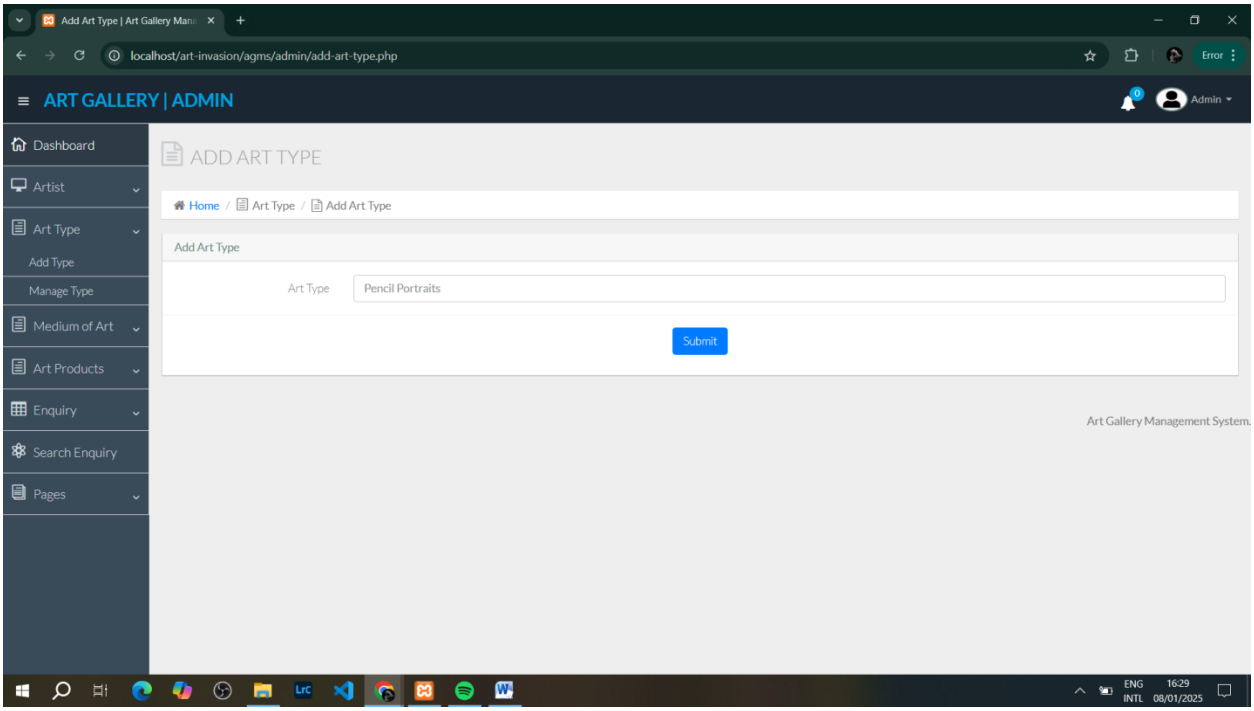


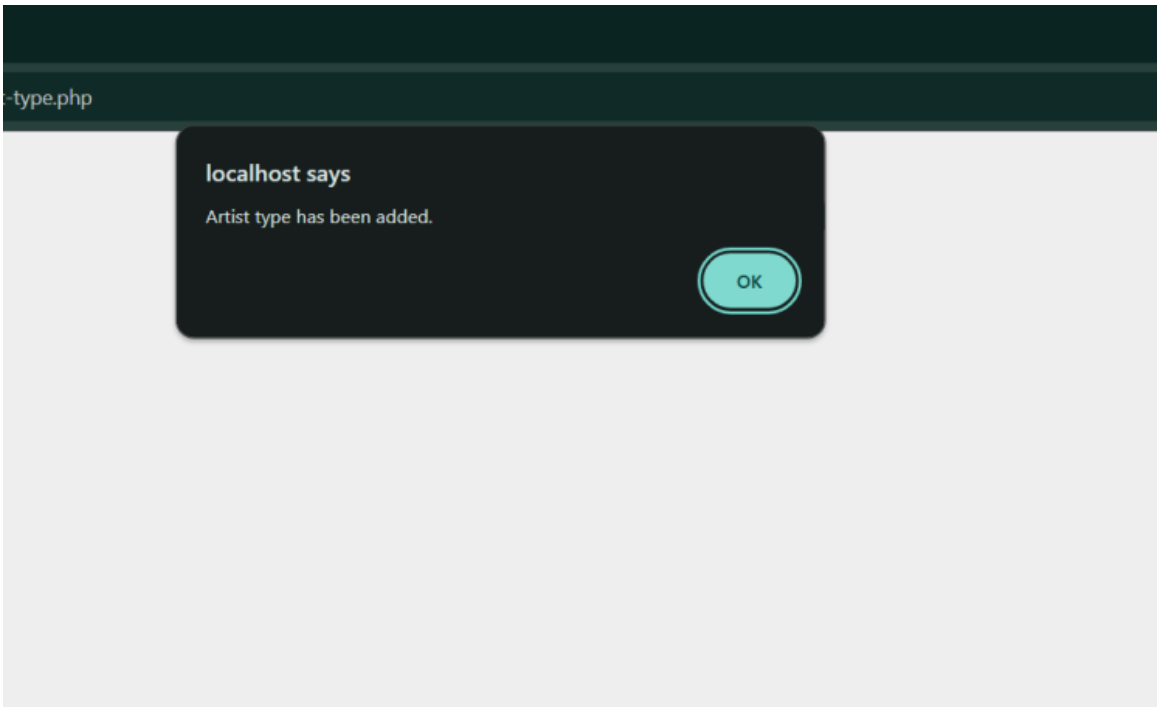


5.2 Managing Art Types:

Admin can add, edit, or delete art types (e.g., Paintings, Sculptures, Digital Art).

- To add a new art type:
    1. From the Admin Dashboard, go to the **Art Type** section.
    2. Click **Add Art Type**.
    3. Enter the name of the art type and click **Submit**.

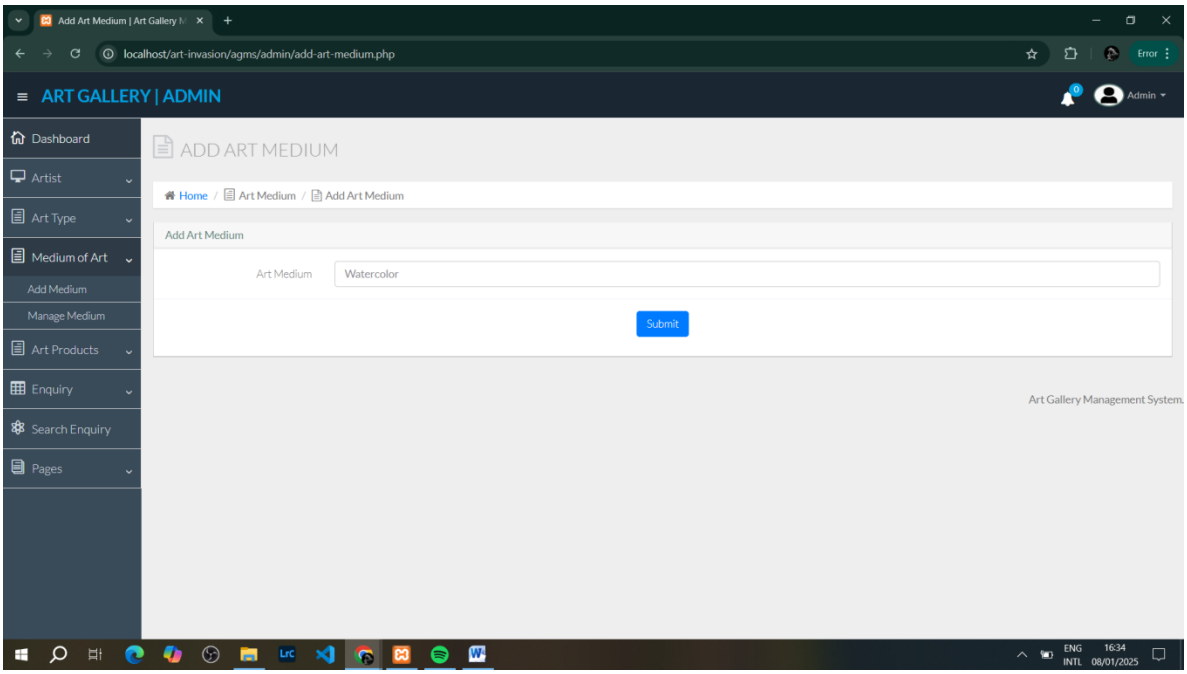


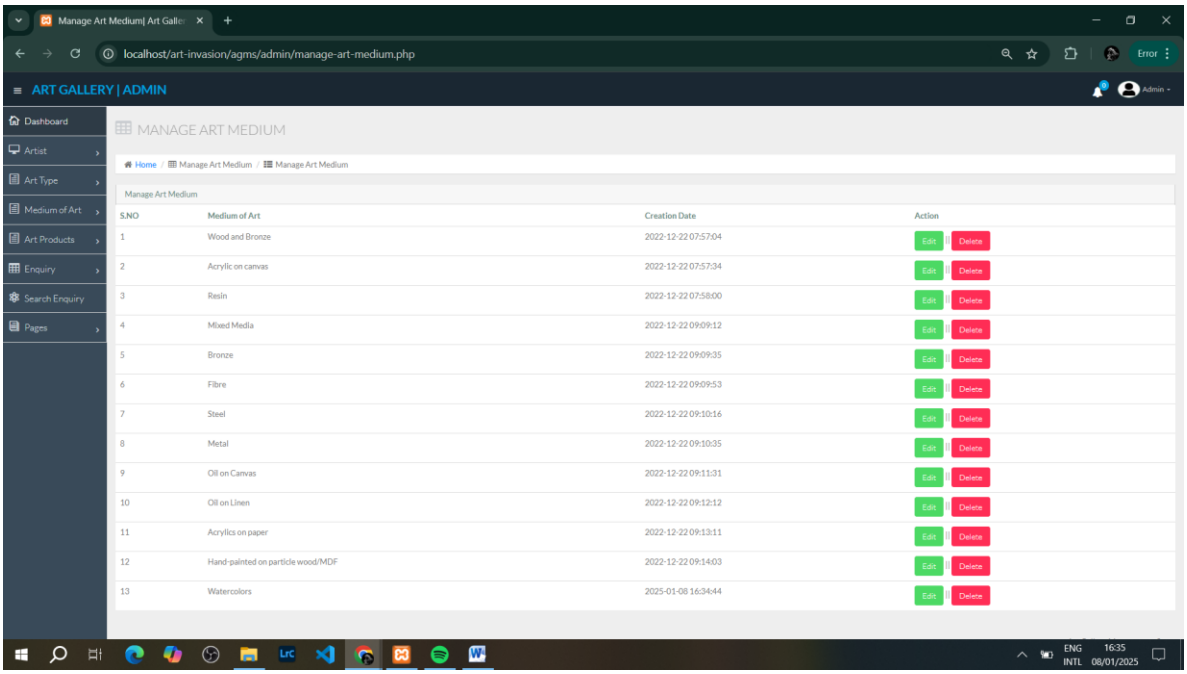


5.3 Managing Art Mediums:

Admin can manage different mediums (e.g., Oil, Acrylic, Watercolor).

- To add a new medium:
    1. Go to the **Art Medium** section.
    2. Click **Add Medium**.
    3. Enter the medium name and save.

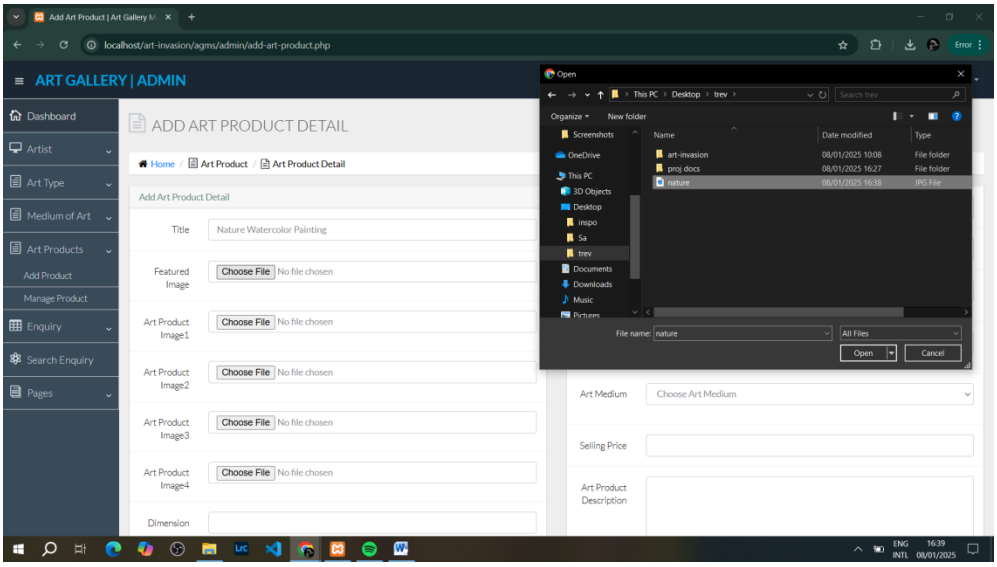


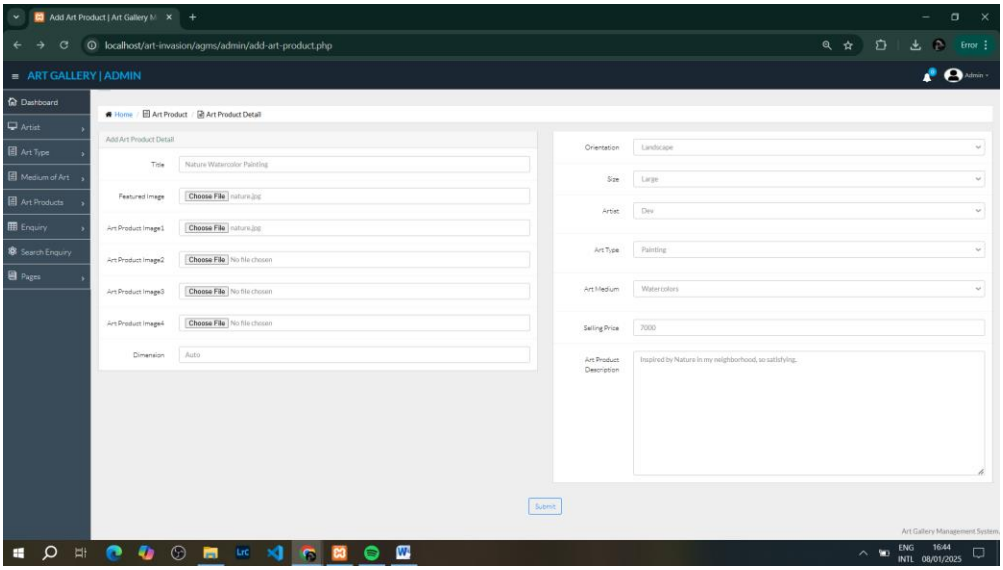


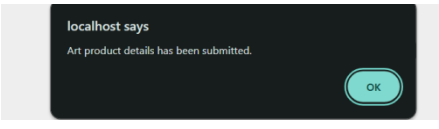
5.4 Managing Art Products:

Admin can add, update, or delete art products listed on the platform.

- To add a new art product:
    1. Go to **Art Product** section.
    2. Click **Add Product**.
    3. Fill in the details such as title, description, price, and select the art type and medium.
    4. Upload an image for the product and click **Save**.



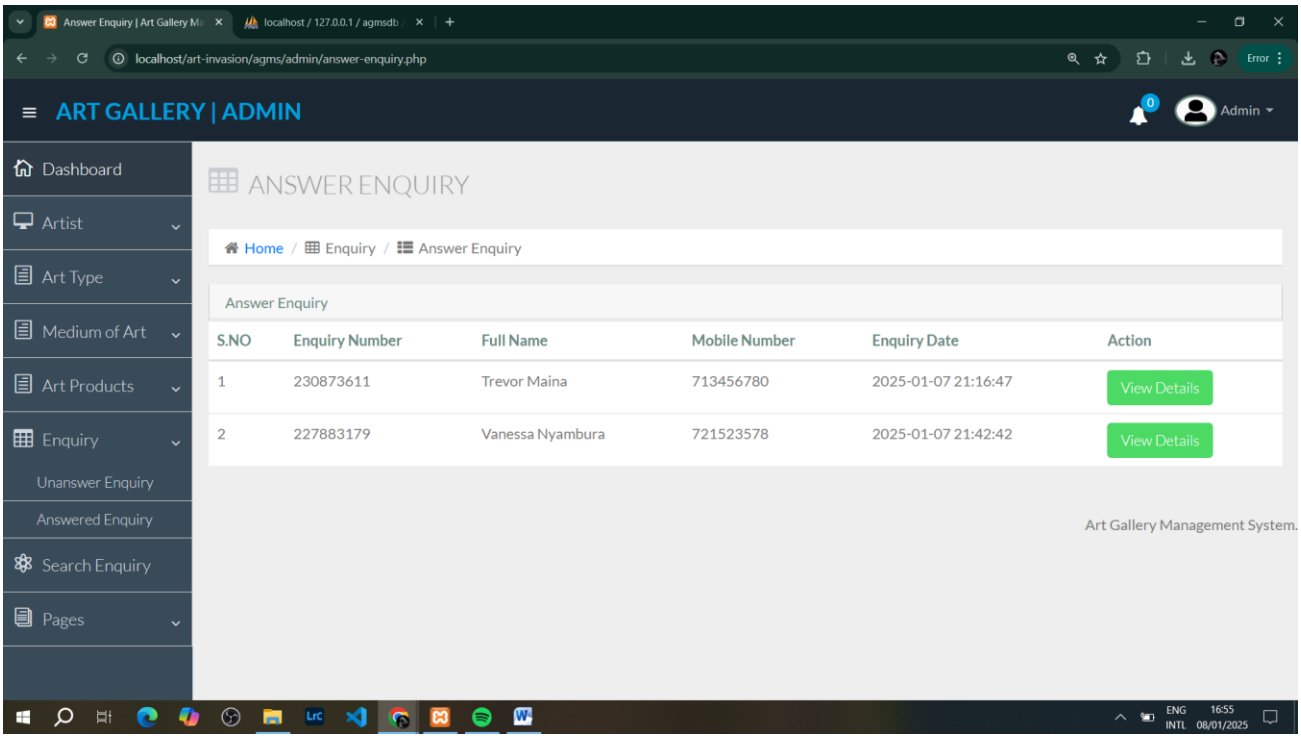




5.5 Handling User Enquiries:

Admin can view and manage user enquiries.

- To view enquiries:
    1. Navigate to the **Enquiry** section.
    2. Click on the enquiry number to view the details.
    3. You can mark enquiries as answered or delete them as needed.

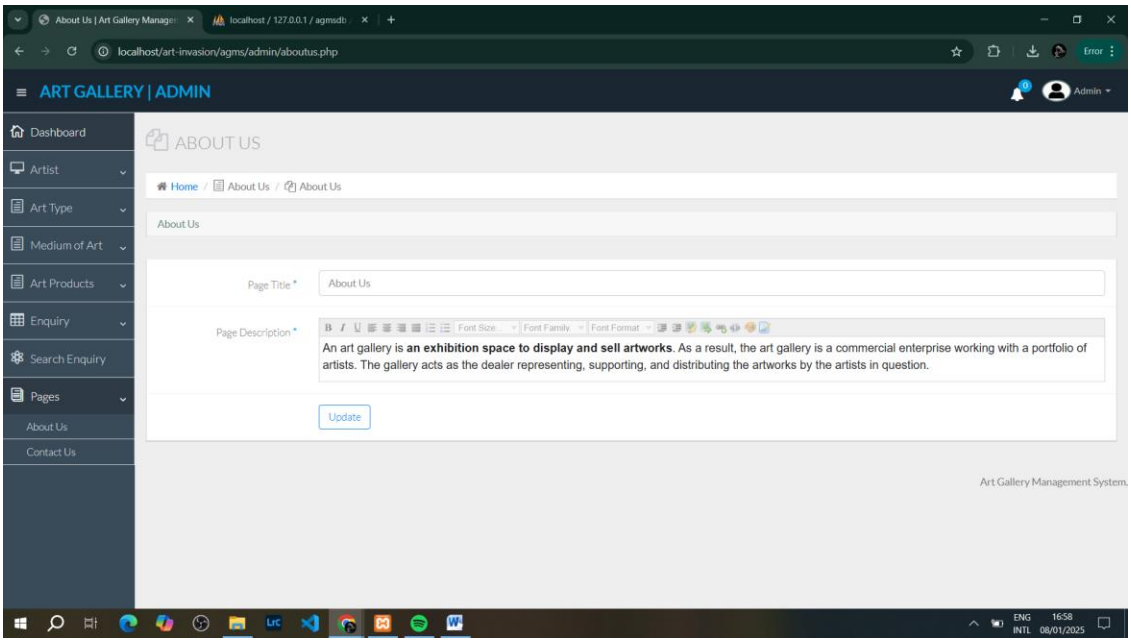


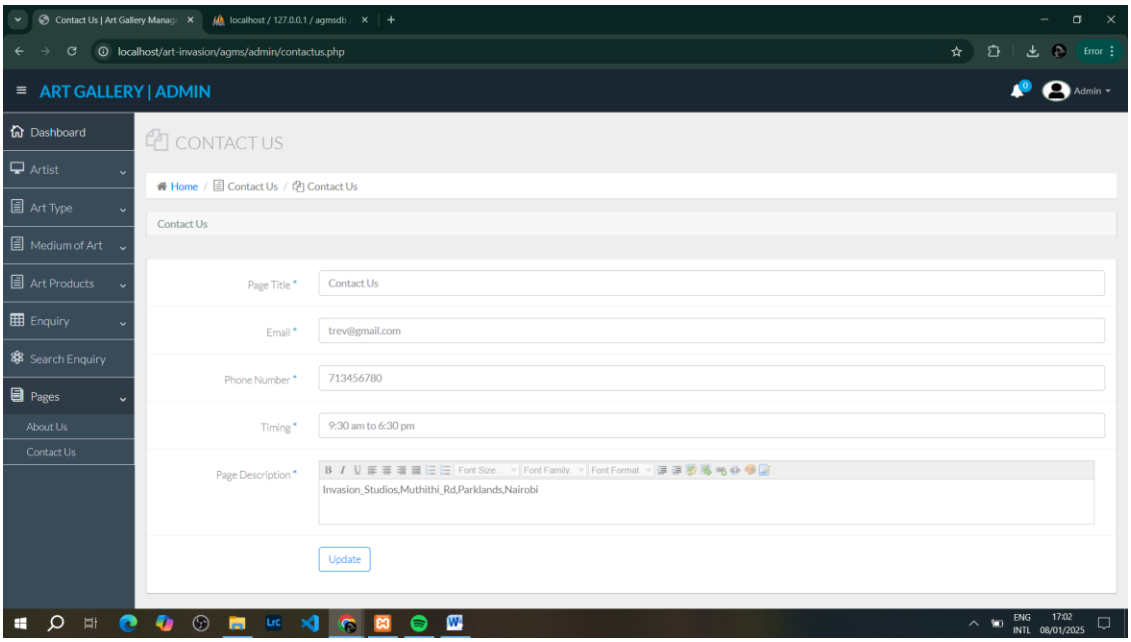
Next Page – Managing Pages...

5.6 Managing Pages:

Admin can update the About Us and Contact Us pages.

- To edit a page:
    1. Go to the **Page** section.
    2. Select **Edit About Us** or **Edit Contact Us**.
    3. Update the content and click **Save**.

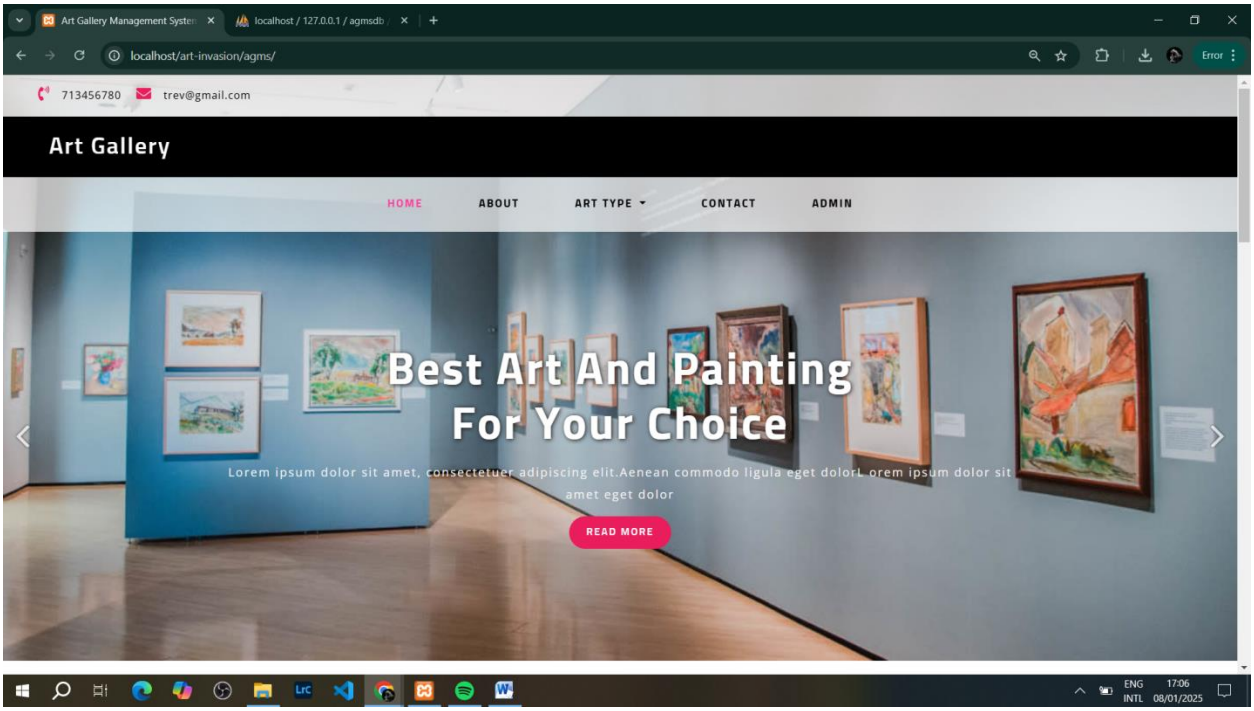




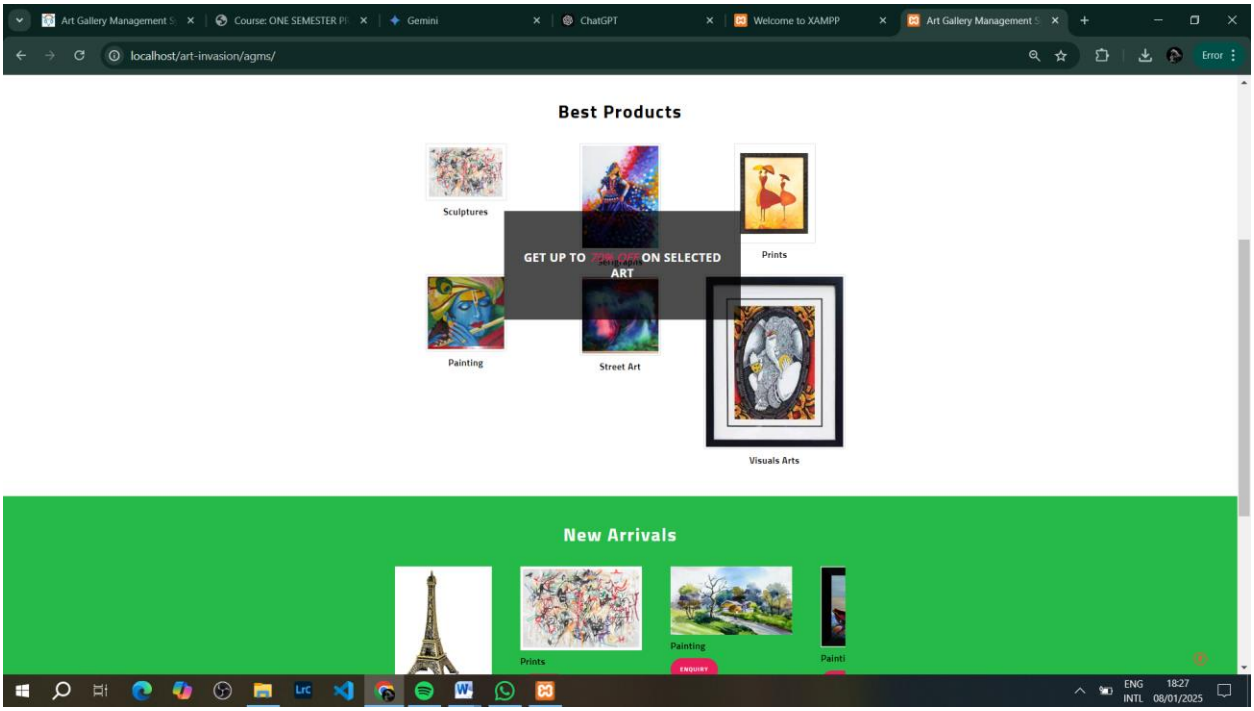
6. User Module Instructions

6.1 Home Page Overview:

- The Home page serves as the welcome page for users. Here, users can browse various art products based on art type and medium.



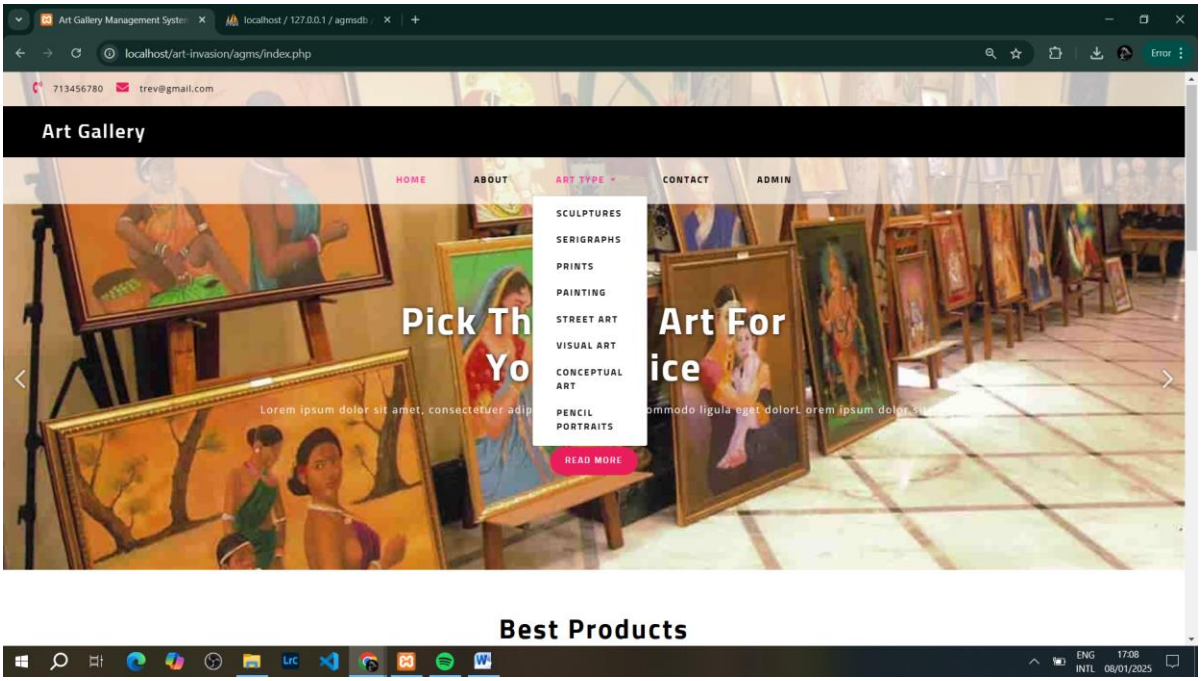
Scroll Down...

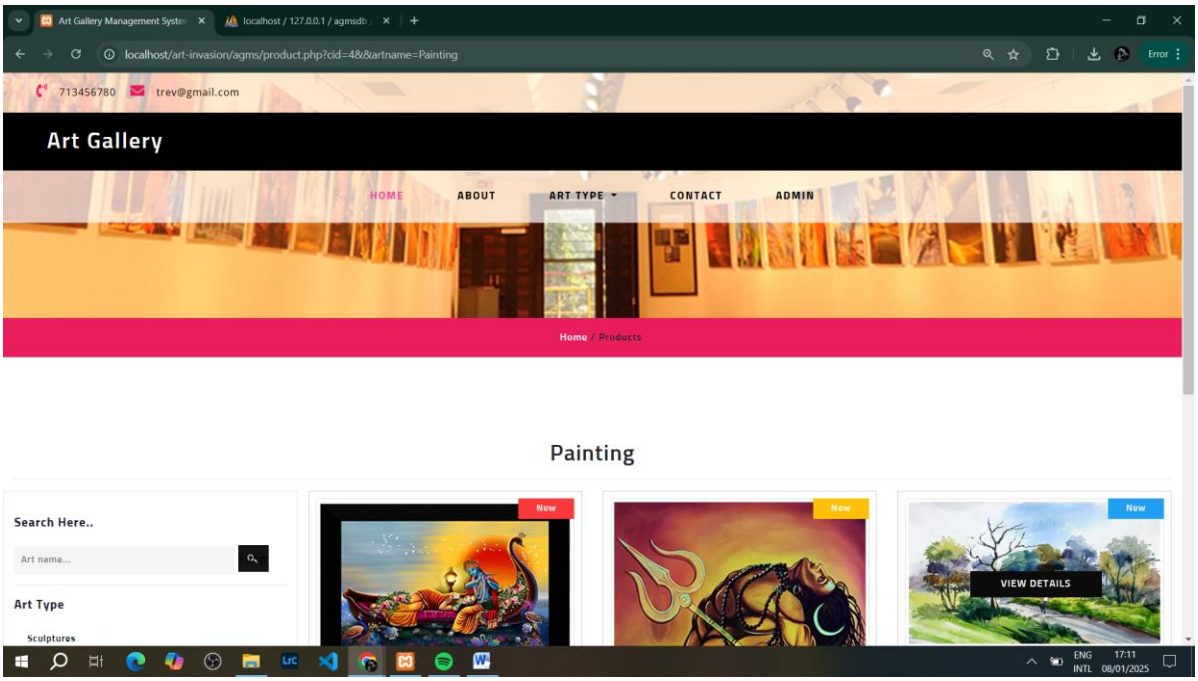


6.2 Browsing Art Types:

Users can view art products filtered by different types.

- To view art types:
    1. On the Home page, click on the **Art Type** menu.
    2. Select an art type (e.g., Paintings, Sculptures).
    3. The system will display all products under that type.

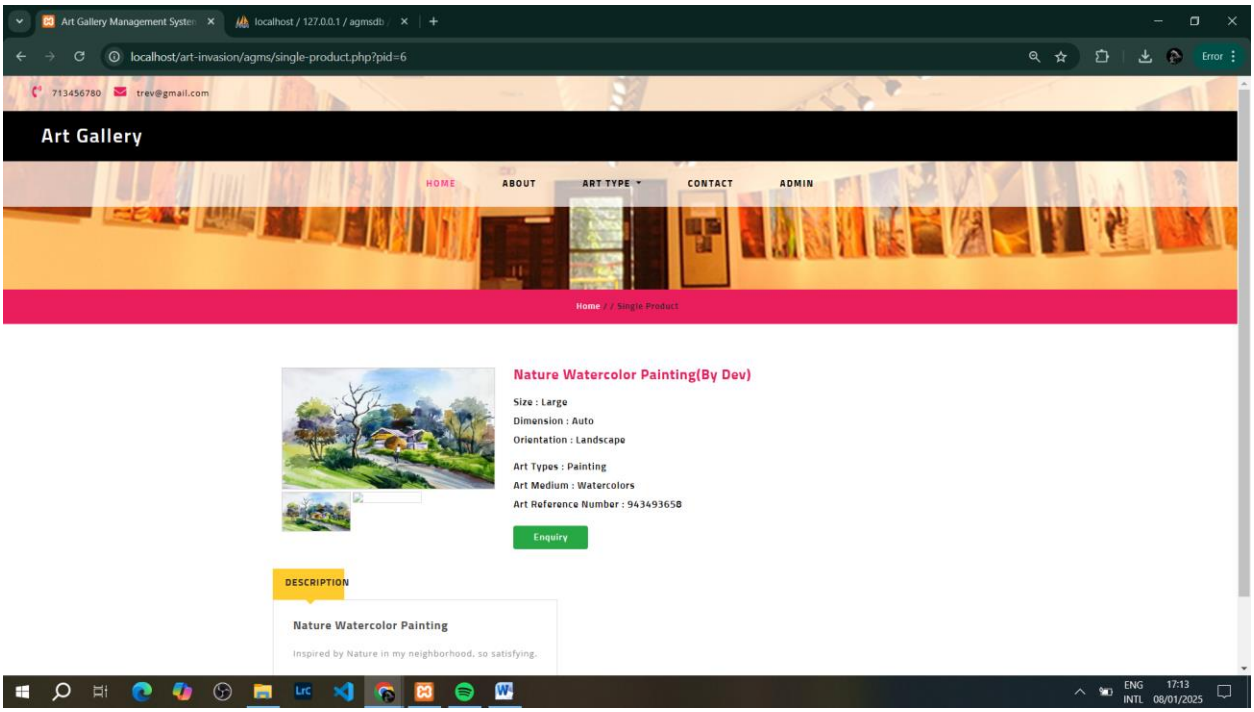




6.3 Viewing Art Products:

Users can view the details of a selected art product.

- To view a product:
    1. Click on any art product listed under the art type.
    2. The product details page will show a larger image, description, and price.

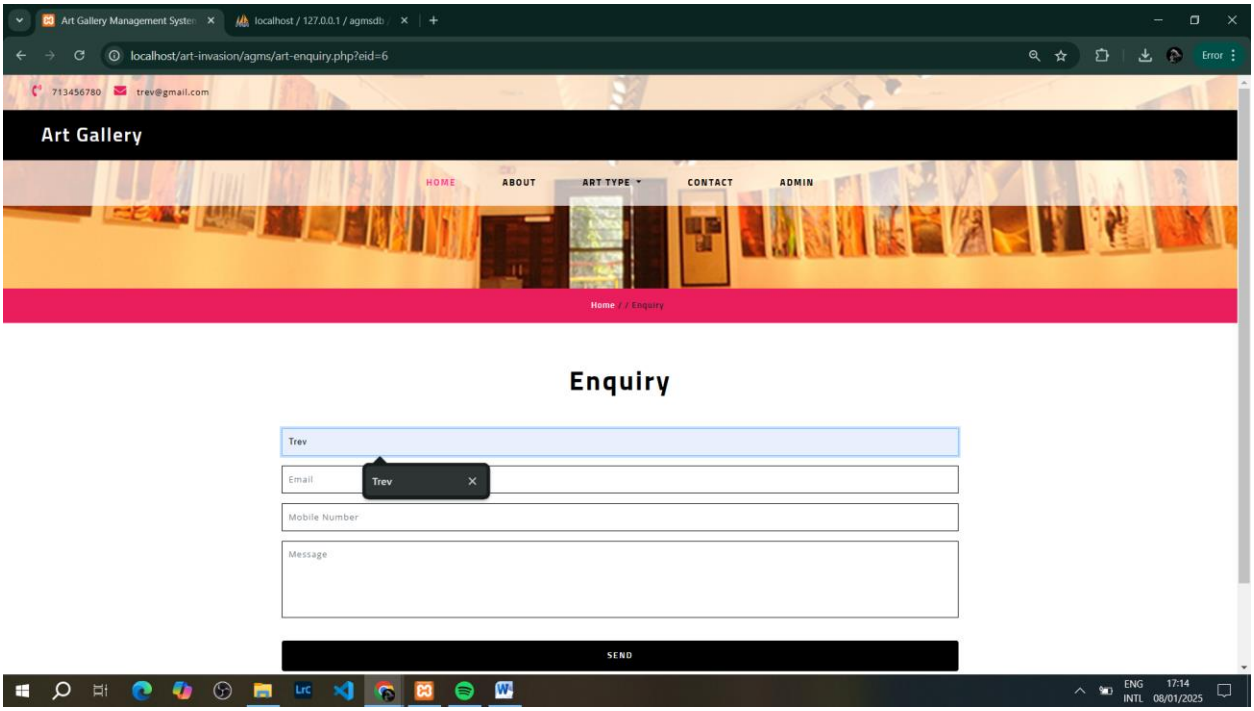


Next Page – Submitting an Enquiry...

6.4 Submitting an Enquiry:

Users can submit an enquiry about a product they are interested in.

- To submit an enquiry:
    1. On the art product details page, click **Submit Enquiry**.
    2. Fill in your name, email, and message.
    3. Click **Send Enquiry**.



---

7. Troubleshooting

7.1 Common Issues and Solutions:

- **Issue:** “Cannot connect to the database.”
    - **Solution:** Ensure MySQL is running in XAMPP and the correct database credentials are configured in the login.php file.
- **Issue:** "Page not loading."
    - **Solution:** Check if Apache and MySQL are running in XAMPP, and verify the file path for the application.

---

8. Appendix

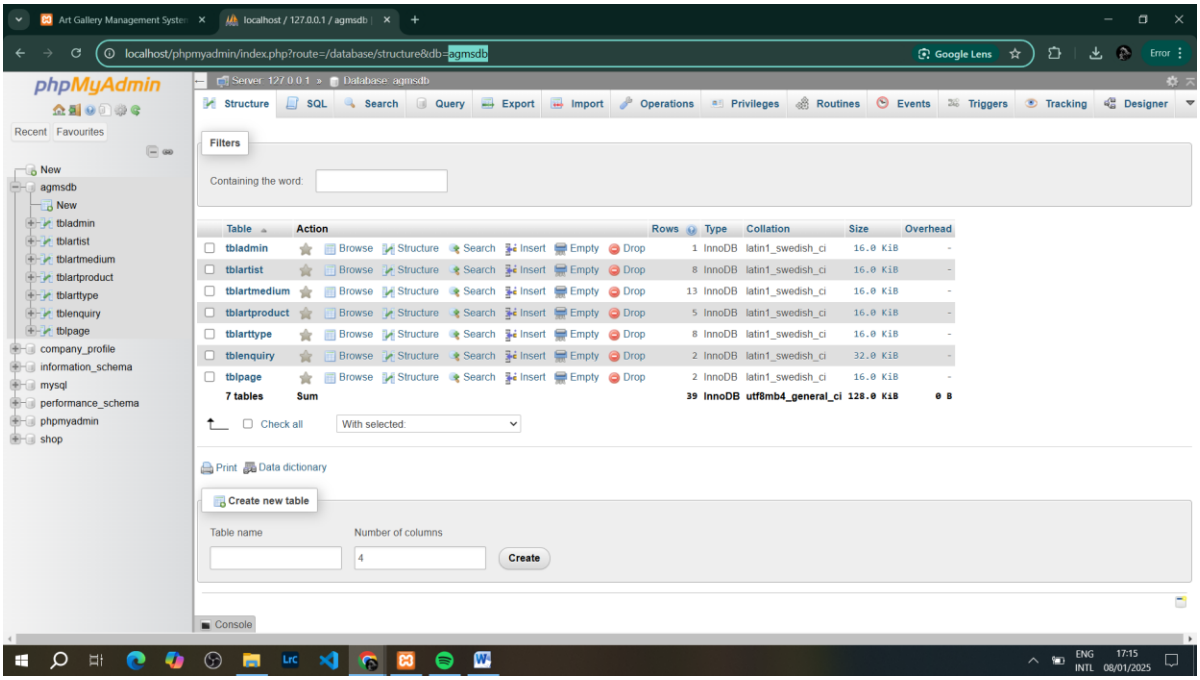
8.1 Code Snippets:

- **Database Connection Example (dbconnection.php):**

```
<?php
$con=mysqli_connect("localhost", "root", "", "agmsdb");
if(mysqli_connect_errno()){
echo "Connection Fail".mysqli_connect_error();
}
?>
```

8.2 Database Schema:

Include SQL schema for setting up the database (e.g., agmsdb.sql).



---

9. Conclusion

This **User Implementation Document** serves as a complete guide for both Admins and Users on how to effectively use the **Art Invasion Gallery Management System**. Follow the steps outlined for each module to ensure smooth operation and management of art products, enquiries, and more.
