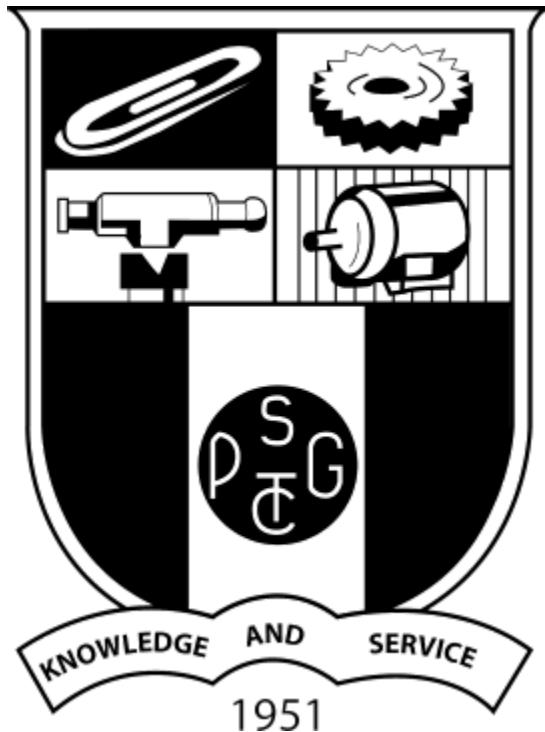


DATABASE MANAGEMENT SYSTEM LABORATORY

19Z411



MINI PROJECT REPORT

DONE BY:

22Z310 ARULKUMARA B R

22Z323 GIRI PRASSATH S

22Z330 JAYANTH S

22Z339 MATHAN KUMAR S

22Z341 MITHUNKARTHIK A S

Art Gallery Management System

Table of Contents

1. Introduction	3
2. Problem Statement	3
3. System Requirements	5
3.1. Functional Requirements	5
3.2. Non Functional Requirements	6
4. Tech Stack	7
5. Enhanced Entity Relationship Model	9
6. Conversion Rules.....	10
7. Relational Model	10
8. Normalization	13
9. Database Schema	15
10. Implementation	16
10.1. Customer Flow	16
10.2. Admin Flow	19
11. Contribution	21
12. Conclusion	22

1. Introduction

The Art Gallery Management System project aims to provide a comprehensive solution for managing the operations of an art gallery. This system will streamline processes related to managing art collections, artist information, exhibitions, customer interactions, employee management and sales transactions. The system is designed to improve operational efficiency, data integrity, and overall management effectiveness.

2. Problem Statement

Galleries need to manage comprehensive information about individuals, including personal details, contact information, and login credentials. This includes artists, customers, and employees. Secure and consistent storage of this data is critical to ensure privacy and operational efficiency. Proper management of personal information helps in building a strong foundation for all gallery operations, allowing for accurate tracking of individuals and their roles within the gallery.

Efficiently tracking artists and their unique styles is essential for cataloging art pieces and organizing exhibitions. Accurate linking of artist details to their artworks and exhibitions ensures proper attribution and facilitates smooth event management. This enables the gallery to showcase the correct information about the artists, their styles, and the artworks they have contributed to exhibitions, enhancing the visitor experience and supporting marketing efforts.

Handling customer information is vital for managing purchases, maintaining communication, and conducting marketing activities. The system must securely manage customer data and track their purchase history, enhancing customer service and marketing strategies. By maintaining detailed records of customer interactions and preferences, the gallery can tailor its communications and promotions to better meet customer needs, ultimately driving sales and improving customer satisfaction.

Effective management of employee records, including their roles, employment details, and hierarchical relationships, supports better internal operations. Proper employee management structures the organization efficiently and aids in

reporting and decision-making. Maintaining accurate employee records ensures that the gallery can effectively assign tasks, manage workloads, and track performance, contributing to a more organized and productive workplace.

Recording and linking phone numbers to individuals is crucial for maintaining effective communication channels. Ensuring accurate association of phone numbers with artists, customers, and employees is essential for reliable contact management. This allows the gallery to reach out to individuals quickly and effectively, whether for marketing purposes, customer service, or internal communications.

Galleries often operate across multiple locations, each requiring detailed information management, such as their identities and addresses. Proper management of gallery details supports the organization of exhibitions and the assignment of employees to different locations. By keeping accurate records of each gallery's location and attributes, the system facilitates the planning and coordination of events, staff assignments, and logistics.

Organizing exhibitions involves scheduling events, managing participating artworks, and associating exhibitions with specific galleries. The system must handle these tasks to ensure smooth planning and execution of exhibitions, enhancing visitor experiences and operational efficiency. This includes tracking the start and end dates of exhibitions, the artworks displayed, and the galleries hosting them, ensuring that all aspects of the exhibition are well-coordinated.

Managing art pieces involves tracking their details, pricing, availability, and associations with artists and exhibitions. Accurate record-keeping of inventory and sales is crucial to maintain the integrity and value of the art collection. By maintaining comprehensive records of each artwork, including its price, genre, received year, and availability status, the gallery can ensure that its inventory is up-to-date and that artworks are properly managed and promoted.

Processing sales transactions effectively involves linking customers with purchased artworks and maintaining accurate transaction records. The system must support secure and efficient purchase handling to ensure customer satisfaction and financial accuracy. By tracking transactions accurately, the gallery can maintain a clear record of sales, manage its finances effectively, and provide customers with reliable service and support.

3. System Requirements

The System Requirements section of the report outlines the foundational elements necessary for the successful deployment and operation of the Art Gallery Management System. This includes specifications for hardware, software, and other essential components required to support the system's functionalities.

3.1. Functional Requirements

Within the Functional Requirements section, the report details the core functionalities and features expected from the Art Gallery Management System. These requirements specify the specific tasks, behaviors, and services the system must provide to meet the needs of users, including managing artist information, handling customer interactions, and organizing exhibitions.

1. User Management

- Ability to manage information about artists, customers, and employees.
- Secure login and authentication for all users.
- Role-based access control to ensure appropriate access levels for different user types.

2. Artist Management

- Track artist information, including personal details and artistic styles.
- Link artists to their artworks and exhibitions.

3. Customer Management

- Manage customer data securely, including contact information and purchase history.
- Support for customer communications and marketing activities.

4. Employee Management

- Maintain employee records, including roles, employment details, and reporting structures.
- Track employee assignments to different galleries and exhibitions.

5. Gallery Management

- Manage gallery locations, including their addresses and operational details.
- Link galleries to exhibitions and employees.

6. Exhibition Management

- Organize and schedule exhibitions, including participating artworks and associated galleries.
- Manage exhibition details such as start and end dates.

7. Artwork Management

- Track details of art pieces, including name, price, genre, received year, and availability.
- Link artworks to artists and exhibitions.

8. Sales and Transaction Management

- Handle sales transactions securely, linking customers with purchased artworks.
- Maintain accurate transaction records for financial reporting.

3.2. Non Functional Requirements

Non-Functional Requirements are addressed in the report to describe the operational attributes and quality characteristics of the Art Gallery Management System. This section outlines criteria such as performance, security, usability, scalability, reliability, and maintainability, ensuring the system's efficiency, user satisfaction, and robustness in various operating conditions.

1. Performance
 - The system should handle multiple concurrent users efficiently.
 - Quick response times for database queries and transactions.
2. Security
 - Ensure secure storage and transmission of sensitive data.
 - Implement strong authentication and access control mechanisms.
3. Usability
 - User-friendly interfaces for different user roles.
 - Easy navigation and intuitive design for efficient operation.
4. Scalability
 - Ability to scale the system to handle growing amounts of data and users.
 - Support for additional features and integrations in the future.
5. Reliability
 - Ensure high availability and reliability of the system.
 - Implement robust backup and recovery procedures.
6. Maintainability
 - Modular design for easy maintenance and updates.
 - Clear documentation for developers and users.

4. Tech Stack

The Art Gallery Management System leverages a comprehensive tech stack to provide efficient functionality and a seamless user experience. Here's an expanded overview of the technologies involved:

Backend Development:

Node.js: Node.js is utilized as the runtime environment for server-side scripting. Its event-driven architecture and non-blocking I/O operations make it suitable for building scalable and high-performance web applications.

Express.js: Express.js, a minimalist web application framework for Node.js, is employed to streamline server-side development. It simplifies routing, middleware integration, and handling of HTTP requests, thereby accelerating the development process.

mysql2: mysql2 serves as the MySQL database driver for Node.js. It enables asynchronous interaction with MySQL databases, facilitating efficient data retrieval, manipulation, and storage within the application.

Database Management:

MySQL: MySQL is chosen as the relational database management system (RDBMS) for storing and managing data related to artists, customers, employees, artworks, galleries, exhibitions, and transactions. Its robust features, scalability, and reliability make it a preferred choice for handling structured data.

Frontend Development:

HTML/CSS: HTML (Hypertext Markup Language) is employed for structuring the web pages, while CSS (Cascading Style Sheets) is used for styling and visually enhancing the user interface. Together, they enable the creation of interactive and visually appealing frontend components.

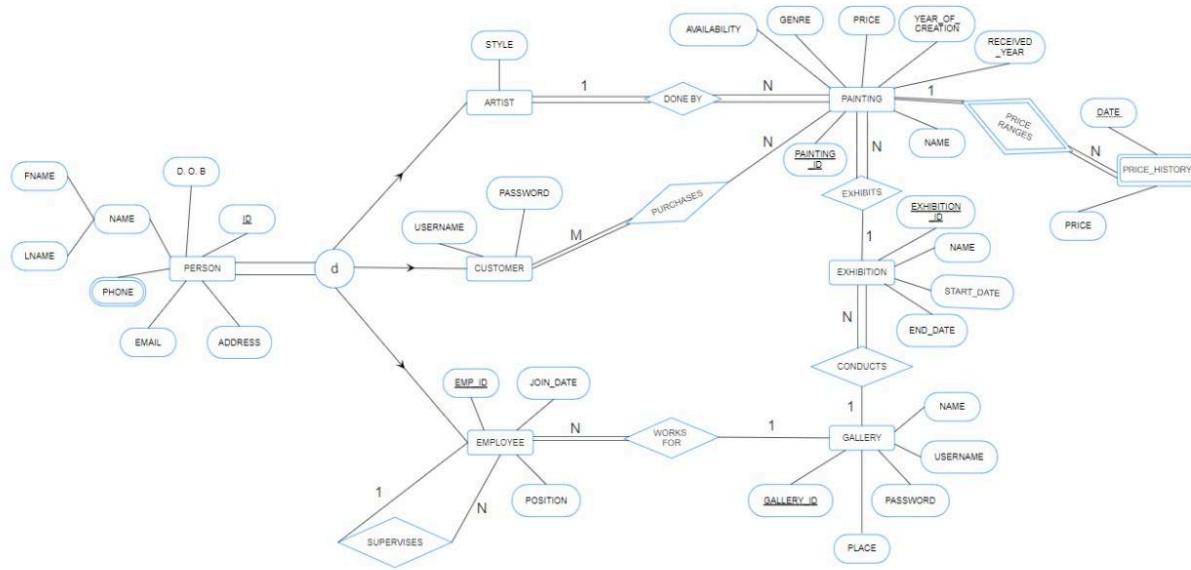
Development Tools:

Visual Studio Code: Visual Studio Code (VS Code) serves as the primary integrated development environment (IDE) for writing, editing, debugging, and managing the project's codebase. Its extensive ecosystem of extensions and built-in features enhances productivity and facilitates collaboration among team members.

MySQL Workbench 8.0 CE: MySQL Workbench 8.0 CE serves as a crucial development tool in the Art Gallery Management System project. It provides a user-friendly graphical interface for designing, modeling, and managing MySQL databases. With features like visual database design, SQL code editing, schema modeling, and data visualization, MySQL Workbench streamlines database development tasks, enhances productivity, and simplifies database management.

By leveraging this tech stack, the Art Gallery Management System achieves a balance between performance, functionality, and usability, ensuring a seamless experience for both administrators and end-users.

5. Enhanced Entity Relationship Model



Entities:

- Customer:** Represents individuals who visit the gallery and potentially purchase artwork. Attributes include customer name, username, phone number, email address, and address.
- Employee:** Represents individuals employed by the gallery. Attributes include username, password, position (e.g., curator, salesperson), and join date.
- Gallery:** Represents the physical location where exhibitions are held and staff are employed. Attributes include gallery name and location.
- Exhibition:** Represents events showcasing a collection of paintings for a specific period. Attributes include exhibition name, start date, and end date.
- Painting:** Represents individual artworks within the gallery's collection. Attributes include artist, genre, availability (for purchase), style, and current price.

6. **Price History (Weak Entity):** Represents historical price data for each painting. Because a single price point doesn't uniquely identify a record, it's a weak entity dependent on the Painting entity.
- Identifying Relationship: **Date (or Timestamp)** - Uniquely identifies a specific price record along with the Painting ID.
 - Price - The price of the painting at a specific point in time.

Relationships:

- **Purchases:** Links customers to paintings they have purchased.
- **Supervision:** Represents the hierarchical structure within the gallery staff, where one employee supervises others.
- **Employment:** Establishes the relationship between employees and the gallery where they work.
- **Showcases:** Links exhibitions to the paintings displayed within them.
- **Created By:** Connects paintings to the artists who created them.
- **Has Price History:** Connects a painting to its historical price data stored in the Price History entity.

Key Points:

- The model captures the current price of a painting in the Painting entity and historical prices in the Price History entity.
- The Price History entity is weak because it relies on the Painting entity for existence and a single date wouldn't uniquely identify a price record. The Date attribute along with the relationship to Painting acts as the unique identifier.
- This model facilitates tracking price changes for paintings over time, enabling analysis of pricing trends and informed decision-making.

6. Conversion Rules

Identify Relational Tables:

Each entity in the EER diagram directly translates to a relational table in the database. The table will have columns corresponding to the attributes of the entity.

- Person

- Customer (inherits from Person)
- Employee (inherits from Person)
- Artist (inherits from Person)
- Gallery
- Exhibition
- Painting
- Price History

Address Foreign Key Relationships:

Inheritance:

Since Customer, Artist and Employee inherit attributes from Person, we don't need separate foreign keys in those tables. They'll reuse the primary key of Person (usually Person ID).

One-to-Many Relationships:

For mandatory one-to-many relationships, create a foreign key in the child table that references the primary key of the parent table.

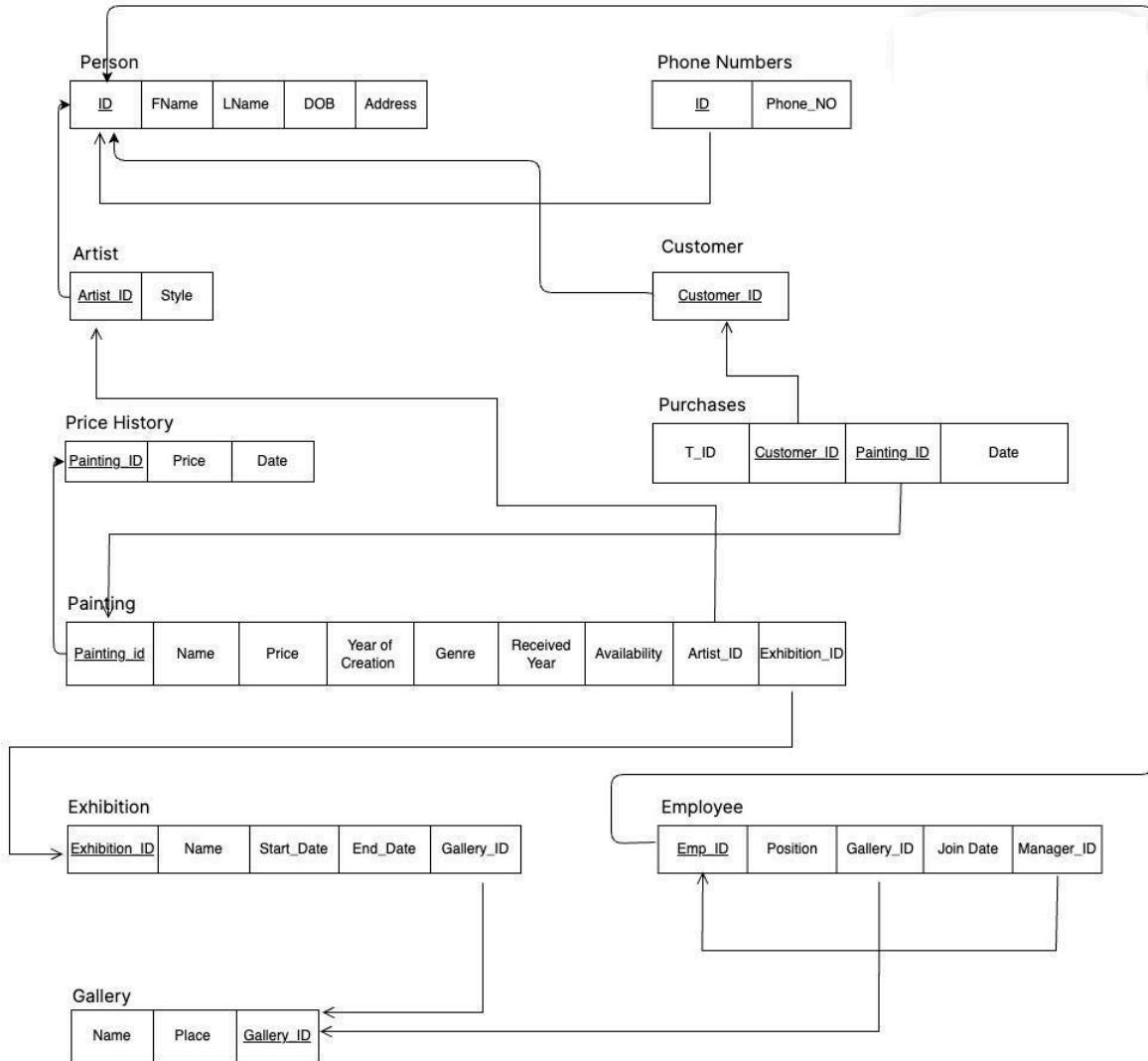
- Employee table: Create a foreign key 'Gallery ID' referencing the primary key 'Gallery ID' in the Gallery table (one gallery can have many employees).
- Exhibition table: Create a foreign key 'Gallery ID' referencing the primary key 'Gallery ID' in the Gallery table (one gallery can have many exhibitions).
- Painting table: Create a foreign key 'Artist ID' referencing the primary key in the Artist table (one artist creates many paintings).

Handle Many-to-Many Relationships:

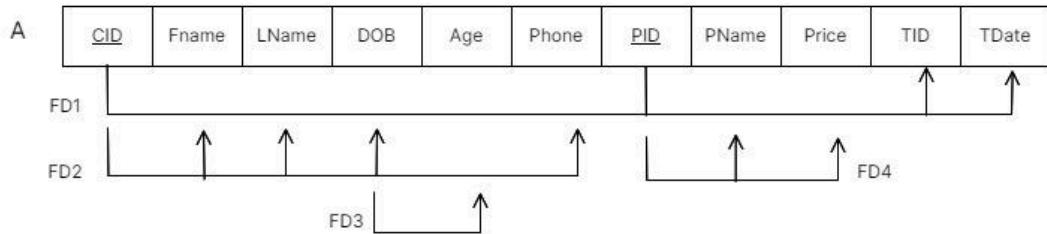
For many-to-many relationships, create a new table to associate the two related tables. This new table will have foreign keys referencing the primary keys of both the original tables.

- **Purchases:** A customer can purchase many paintings, and a painting can be purchased by many customers. Create a new table 'Purchases' with foreign keys 'Customer ID' referencing Customer and 'Painting ID' referencing Painting.

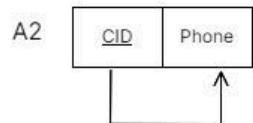
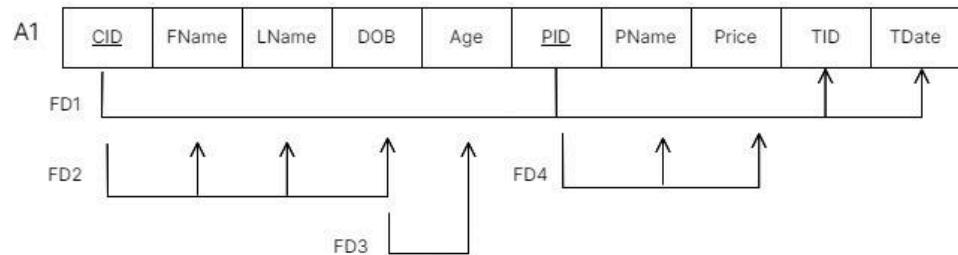
7. Relational Model



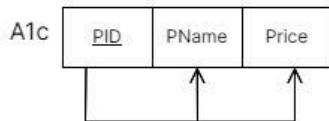
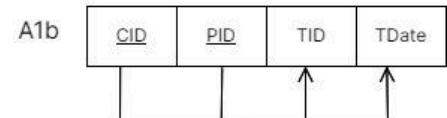
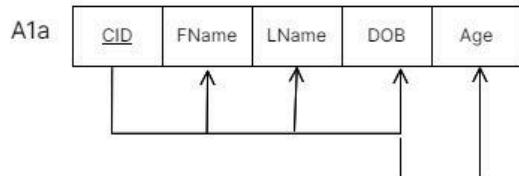
8. Normalization



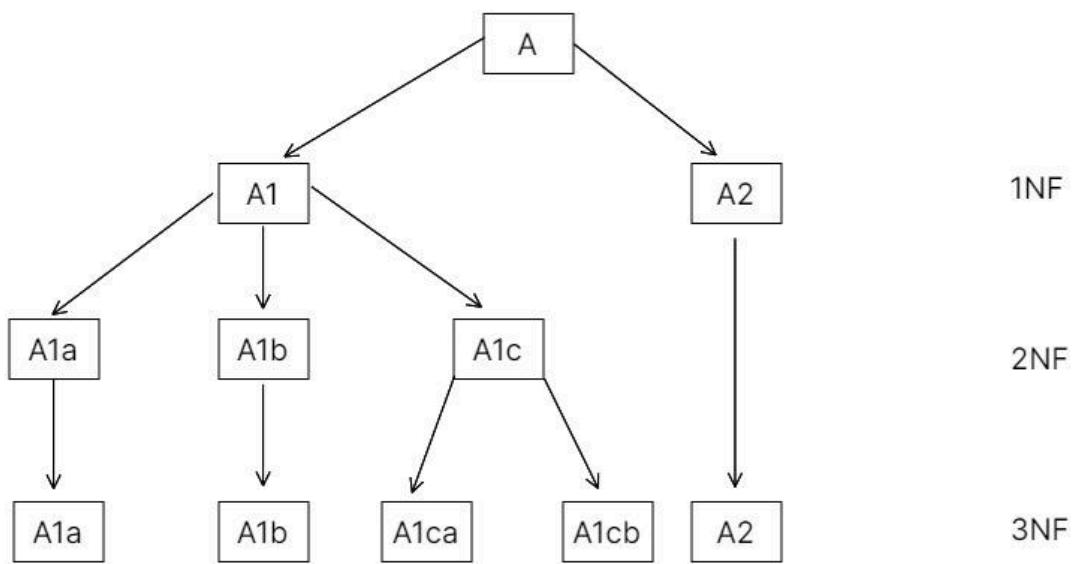
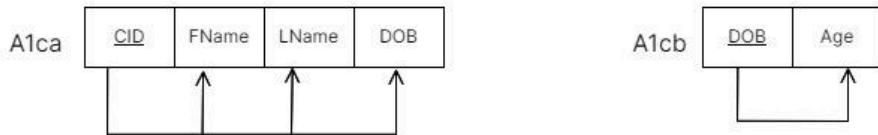
1NF



2NF

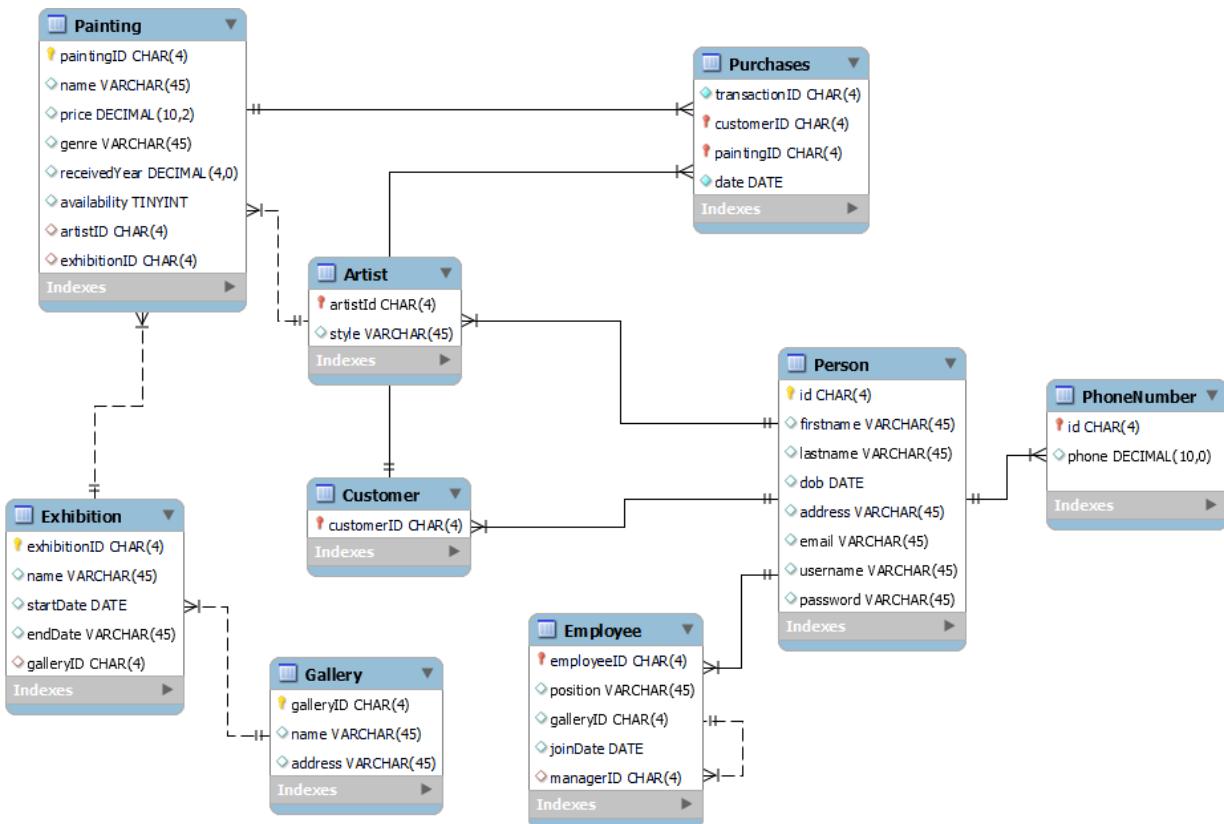


3NF



9. Database Schema

A database schema is like a blueprint for organizing information in a database. In this case, it's for an art gallery. Imagine separate filing cabinets for artists, customers, employees, paintings, and other things. The schema defines what goes in each cabinet and how the cabinets are linked. Each cabinet (table) has categories (fields) to store details, and unique IDs help keep track of everything. Special links connect related things between cabinets, such as artists and their paintings or customers and their purchases.

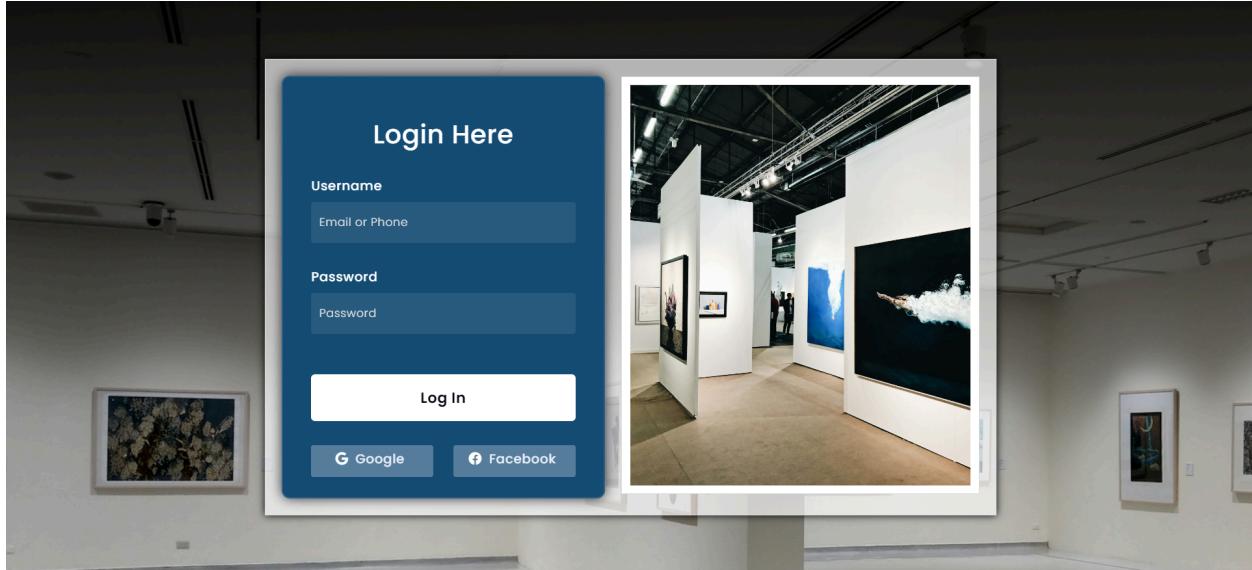
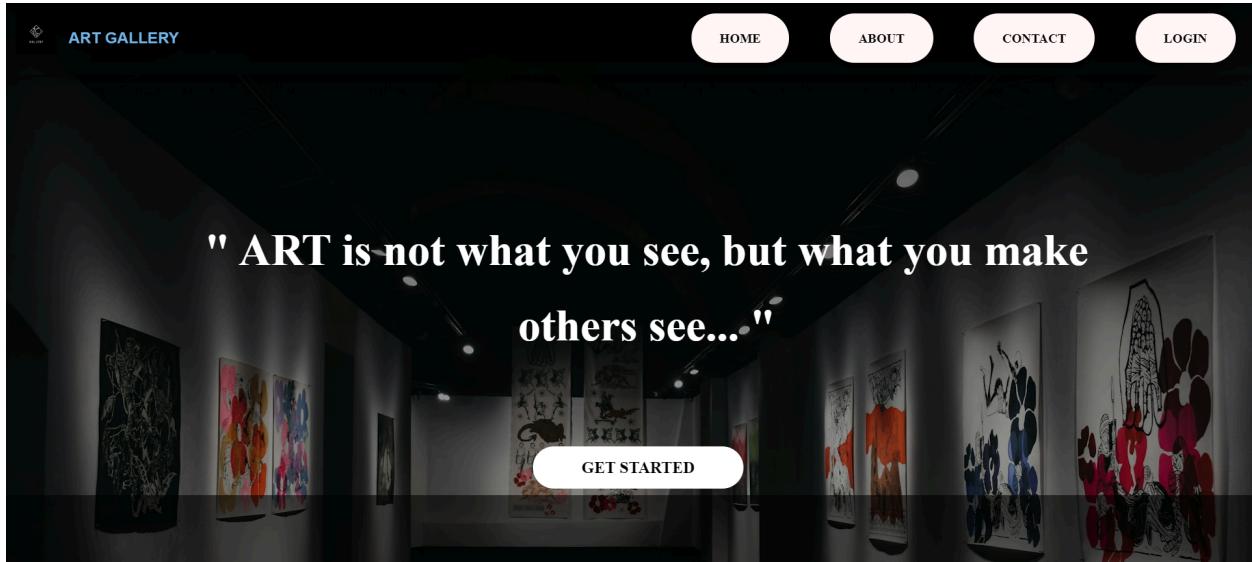


10. Implementation

10.1. Customer Flow

1. Server Setup: The application uses Express.js as the web framework and MySQL as the database. It initializes the server and configures a MySQL connection pool.
2. Middleware Configuration: Middleware such as express.static, express.json, and bodyParser.urlencoded is configured to handle static files and parse request bodies.
3. Route Handlers:
 - Home Route: Serves the home page.
 - Base Route: Serves the base page.
 - Customer Details Route: Serves the customer details page.
 - Fetch and Update Customer Details: Handles POST requests to update customer details in the database.
 - Customer Registration: Handles POST requests to register a new customer, generating a unique customer ID and inserting the customer details into the database.
4. Database Operations:
 - Generate Customer ID: A function to generate a unique customer ID based on existing IDs in the database.
 - Generate Transaction ID: A function to generate a unique transaction ID based on the number of existing transactions for a painting.
 - Login Authentication: Handles login authentication by querying the database for matching username and password.
5. Exhibition and Purchase Routes:
 - Exhibition Route: Retrieves information about available paintings in a specific exhibition.
 - Purchase Route: Handles POST requests to purchase a painting, updating its availability and recording the transaction in the database.
 - Purchase History Route: Retrieves the purchase history of a customer from the database.

Screenshots:



Art Gallery Customer Page

[Home](#) > [Art Gallery](#) > Exhibitions

Exhibition ID	Exhibition Name	Start Date	End Date	View Paintings
X001	Spectrum	5/1/2024	5/8/2024	View Paintings
X002	Artistic Echoes	6/11/2024	6/25/2024	View Paintings
X003	PaintFest	1/1/2025	1/31/2025	View Paintings
X004	Expo	2/3/2025	2/10/2025	View Paintings
X005	MineCraft	5/19/2024	5/19/2024	View Paintings

 Exhibitions

 Purchases

Art Gallery Customer Page

[Home](#) > [Art Gallery](#) > Purchases

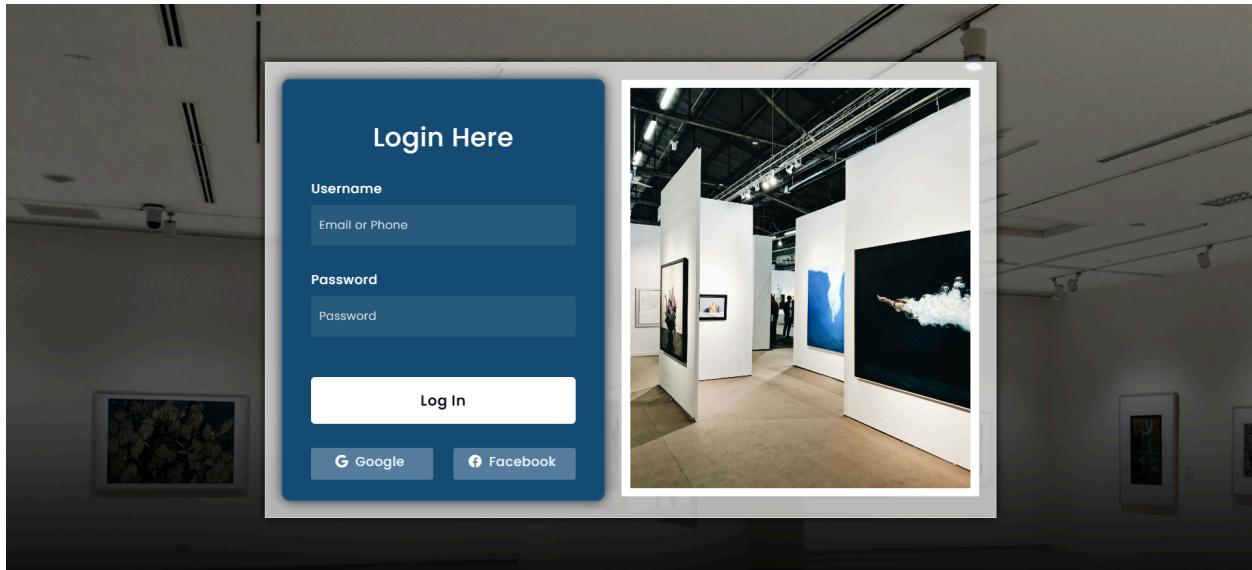
Exhibition	Painting	Artist	Price	Transaction ID	Purchased Date
Artistic Echoes	Painting1	Person2	10000.00	T001	5/19/2024

 Exhibitions

 Purchases

10.2. Gallery Admin Flow

1. Server Setup: Initializes Express application and creates a MySQL connection pool.
2. Middleware Configuration: Configures middleware for serving static files and parsing request bodies.
3. Route Handlers:
 - Home Route: Serves the login page.
 - Login Route: Handles POST requests for user authentication.
 - Dashboard Route: Serves the dashboard page.
 - Artist Registration Route: Handles POST requests to register new artists.
 - Data Routes: Provide endpoints to retrieve data about artists, exhibitions, customers, paintings, and employees.
 - Employee Registration and Deletion Routes: Handles POST requests to register new employees and DELETE requests to delete employees.
 - Exhibition Registration Route: Handles POST requests to register new exhibitions.
 - Painting Registration Route: Handles POST requests to register new paintings.
 - Purchase Route: Retrieves purchase history.
4. Database Operations:
 - Functions to generate unique IDs for artists, employees, exhibitions, and paintings.
 - Query functions to insert new records into the database and check for existing records.
5. Error Handling: Error handling for database operations and route handlers.
6. Server Initialization: Starts the server on port 7777.

A screenshot of the "Art Gallery | Admin" dashboard. The top navigation bar shows "Dash Board" and "Home / Dash Board". A sidebar on the left lists categories: "Exhibition", "Artist", "Painting", "Purchases", "Employee", and "Customer". The main area displays four summary cards: "No. of Employees" (1), "No. of Paintings" (1), "No. of Artists" (2), and "Total Sales" (10000.00).

No. of Employees	No. of Paintings
1	1

No. of Artists	Total Sales
2	10000.00

Art Gallery Admin	
Dash Board > Home / Exhibition	
Exhibition >	
 Add Exhibition	
 View Exhibition	
 Artist >	
 Painting >	
 Purchases >	
 Employee >	
 Customer >	
Exhibition ID	Name
X001	Spectrum
X002	Artistic Echoes
X003	PaintFest
X004	Expo
X005	MineCraft
Start Date	End Date
5/1/2024	5/8/2024
6/11/2024	6/25/2024
1/1/2025	1/31/2025
2/3/2025	2/10/2025
5/19/2024	5/19/2024

Art Gallery Admin	
Dash Board > Home / Employee / Add Employee	
Exhibition >	
 Add Exhibition	
 View Exhibition	
 Artist >	
 Painting >	
 Purchases >	
 Employee >	
 Add Employee	
 Manage Employee	
 Customer >	
Employee Details	
Name:	<input type="text"/>
Email:	<input type="text"/>
Phone:	<input type="text"/>
Date of Birth:	<input type="text" value="dd-mm-yyyy"/>
Address:	<input type="text"/>
Position:	<input type="text" value="Employee"/>
Date of Joining:	<input type="text" value="dd-mm-yyyy"/>
<input type="button" value="Submit"/>	

11. Contribution

Jayanth S: Responsible for **MySQL database creation**, ensuring optimal schema design and data integrity.

Arulkumara B R & Mathan Kumar S: Manages the development of **Node.js** and **Express.js**, overseeing backend logic and API creation. Conducts **Input** and **Validation testing** to verify precise data handling and ensure system reliability.

Mithun Karthik A S: Designs **Admin flow HTML/CSS**, focusing on intuitive interface design for efficient gallery management.

Giri Prassath S: Designs **Customer flow HTML/CSS, Node.js** and **Express.js**, prioritizing user experience and accessibility for seamless interaction with the gallery.

12. Conclusion

In conclusion, the Art Gallery Management System represents a pivotal advancement in streamlining gallery operations. Leveraging modern technologies and collaborative efforts, the system offers a robust platform for efficient management. With its intuitive interface and scalable architecture, it promises enhanced productivity and customer engagement.

The dedication and expertise of our team have been instrumental in delivering a solution that meets functional and aesthetic requirements. Moving forward, continual refinement will be key to adapting to evolving industry demands. By embracing this digital transformation, art galleries can expect improved efficiency and a more dynamic art community.