DBMS Mini Project Review – 1

Exhibition Management System (EMS)

Name	SRN
Deepthi M	PES1UG23AM092
Samrudhi Rao	PES1UG23AM915

Purpose of the Project

The purpose of the Exhibition Management System is to provide a unified digital platform for managing exhibitions held at a single venue. The system streamlines the work of administrators, artists, and visitors by replacing manual processes with structured, database-driven workflows.

For visitors, the system offers a simple process to register, log in, and purchase tickets for exhibitions.
For artists, it provides tools to participate in events by registering artworks and paying a registration fee.
For administrators, the system supports managing employee details, managing exhibitions, scheduling exhibition days, monitoring ticket sales, handling registration fees, tracking artwork sales, and maintaining accurate records of all stakeholders.

By integrating modules for ticketing, artist participation, artwork management, and payments into one platform, the project ensures efficiency, accuracy, and a better experience for everyone involved.

Scope of the Project

The scope of this system is limited to managing exhibitions hosted at a single venue, where multiple events may occur over time. It includes the following modules:

1. **Customer Module**
   Users with a *customer* role can sign up, log in, browse upcoming exhibitions, and purchase tickets.Tickets are recorded per exhibition date, with details such as price, visitor ID, and payment method.
2. **Artist Module**
   Users with an *artist* role can register, log in, and participate in exhibitions by submitting artworks.
   A registration fee is recorded for each exhibition they participate in. Each artwork is linked to a specific exhibition, with details like price, status (sold/not_sold), and buyer (if sold).
3. **Administrator Module**
   Administrators (with *employee* role) can log in and create exhibitions by setting event details such as name, start date, and end date. They can manage exhibition days by configuring ticket availability and monitoring tickets sold. They can track registered artists, artworks, ticket revenue, and artwork sales.
4. **Ticket Sales Management**
   Each ticket purchase is linked to an exhibition and date. Ticket sales are stored with visitor ID, price, purchase date, and payment method.
5. **Registration Fee Management**
   Stores payments made by artists to participate in an exhibition. Includes exhibition ID, artist ID, amount, payment date, and method.
6. **Artwork Management**
   Each artwork record stores exhibition ID, artist ID, price, sold status, customer (if sold), payment method, and payment date. When an artwork is sold, both artwork details and payment information are updated.

Exhibition Management System – A Case Study

To ensure efficient coordination of events, participants, and financial transactions, a venue is implementing a robust Exhibition Management System. The system centralizes exhibition hosting, ticket sales, artwork registration, and administrative control, benefiting visitors, artists, and administrators alike.

At the core of the system is the **Exhibition** entity, uniquely identified by an `exhibition_id`. Each exhibition is associated with attributes such as exhibition name, start date, and end date. Exhibitions serve as the central link for both customer participation and artist involvement. Daily ticket information, including tickets available and tickets sold, is maintained in the **exhibition_days** table to support real-time tracking.

From the **customer perspective**, registered users (`users` table with role = 'customer') can log in to the system, browse upcoming exhibitions, and purchase tickets. Each ticket is tied to a `visitor_id` (customer), the corresponding `exhibition_id`, and `exhibition_date`, ensuring proper entry allocation. Ticket purchase details such as ticket ID, price, purchase date, and payment method are recorded in the **ticket_sales** table. This enables accurate financial tracking and event revenue calculation.

From the **artist perspective**, artists sign up in the system (`users` with role = 'artist') and register under a chosen exhibition. Artist details such as date of birth, nationality, and biography are maintained in the **artist** table. Registration requires payment of a registration fee, stored in the **registration_fee** table, which books their slot in the exhibition. Each artwork is assigned a unique `artwork_id` and is linked to the artist and exhibition in the **artwork_details** table. Attributes such as price, sold status (sold/not_sold), purchase date, payment method, and customer ID (once sold) are maintained.

On the **administrator side**, the admin is represented by employees (`users` with role = 'employee` linked to the **employee** table) who have privileged access to exhibition and financial management. Administrators can view complete exhibition details, including number of tickets sold, number of artists registered, artworks sold, and revenue generated from tickets, registrations, and artwork sales. They are also responsible for configuring exhibitions by setting exhibition names, start/end dates, and ticket availability.

The system provides a structured workflow for handling **artwork sales**. A customer presents their Customer ID (linked to their ticket), which the administrator verifies in the system. The administrator then retrieves Artwork IDs of the purchased items, automatically fetching details from the **artwork_details** table. Once confirmed, the sale is finalized, updating the artwork record (marking it as sold, storing purchase date and buyer ID, and recording payment method).

Unlike a centralized Payment entity, the system distributes financial information across relevant tables: **ticket_sales** for ticket purchases, **registration_fee** for artist registrations, and **artwork_details** for artwork sales. This ensures that payment details are directly tied to the entities they belong to, supporting transparent financial tracking.

The system enforces **controlled access**: Customers and artists can sign up and log in to perform their respective actions, while employees (administrators) have pre-configured credentials in the database, ensuring secure, role-based usage.

Together, these interconnected entities—**Exhibitions, Users (customers, artists, employees), Exhibition Days, Tickets, Registrations, and Artworks**—form a modular framework for managing exhibitions. The system simplifies ticketing, artwork registration, and sales while providing administrators with real-time insights into exhibition performance and revenue, ensuring smooth and transparent exhibition management.

The total profit generated by the system for each exhibition is calculated as:
Profit = (Total Ticket Revenue) + (Total Registration Fees) + (5% Commission on Each Artwork's Price)
This model ensures steady income from tickets and artist registrations, while also allowing the organization to benefit from artwork sales without taking full ownership of the sale.

Limitations of the Project

- The system is restricted to managing exhibitions at a single venue only.
  Multi-venue support is not included.
- Administrators cannot sign up; they are pre-registered as employees in the system.
- Purchases are restricted to online transactions only (tickets, artist registrations, and artwork sales).
- The system does not handle refunds or cancellations of tickets, fees, or artwork sales.
- Offline purchases are not supported.
- Reporting is limited to ticket sales, registration fees, and artwork sales. Advanced analytics or predictive insights are not included.

List of Softwares/Tools/Programming Languages Used

- Frontend: HTML, CSS, JavaScript
- Backend: PHP / Flask (Python) / Node.js (any of these depending on implementation choice)
- Database: MySQL
- Tools:
  - VS Code (IDE for development)
  - Browser (Chrome for testing)