**Project Milestone 2**



**Spring 2025**

**Group Members**

|  |  |
| --- | --- |
| **Name** | **Registration Number** |
| Hassan Zaib Jadoon | 22PWCSE2144 |
| Ahsan Raza | 22PWCSE2099 |
| Mutahhar Fayyaz | 22PWCSE2176 |

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Engr. Summeyea Salahuddin**

Department of Computer Systems Engineering

**University of Engineering and Technology, Peshawar**

Normalized Relational Schema - Event Booking System

**Converted Relational Schema (from Conceptual Schema)**

Based on the Entity Relationship Diagram (ERD), the conceptual schema has been converted into a relational schema and normalized up to **Third Normal Form (3NF)**. Each relation is designed to eliminate redundancy and maintain data integrity.

**Relational Schema**

1. **Admin**  
   (admin\_id, name, email, password, created\_at)
   * Stores information about event administrators
   * Each admin can manage multiple events
2. **Event**  
   (event\_id, admin\_id, title, description, date, time, venue, available\_seats, price, created\_at)
   * Represents individual events created by admins
   * admin\_id is a foreign key referencing the Admin table
   * available\_seats reflects remaining capacity
3. **Customer**  
   (customer\_id, name, email, phone)
   * Contains customer information for bookings
   * Supports booking without requiring full user registration
4. **Booking**  
   (booking\_id, event\_id, customer\_id, status, booking\_date)
   * Links customers to events via bookings
   * event\_id references the Event table
   * customer\_id references the Customer table
   * status indicates current booking state (e.g., pending, confirmed

**Normalization to 3NF**

To ensure the database is efficient, consistent, and free from redundancy, the schema was normalized to the **Third Normal Form (3NF)**. Below is the step-by-step normalization process based on the initial structure of event and booking data.

**Step 1: Unnormalized Form (UNF)**

In the unnormalized form, multiple data points such as seat numbers and customer information were stored together in a single record, resulting in redundancy and multivalued fields.

Example: BookingDetails ( event\_id, event\_title, event\_date, event\_time, venue, admin\_name, admin\_email, customer\_name, customer\_email, seat\_number1, seat\_number2, ..., status)

**Step 2: First Normal Form (1NF)**

In 1NF, all attributes must hold only **atomic** values, and repeating groups must be eliminated.

**Revised Tables in 1NF:**

* **users(user\_id, name, email, password, role)**
* **events(event\_id, title, date, time, venue, price, admin\_id)**
* **seats(seat\_id, event\_id, seat\_number, row, section, status, price)**
* **bookings(booking\_id, event\_id, user\_id, status, total\_amount, selected\_seats)**

*Note: The selected\_seats field in bookings still violates 1NF as it stores multiple values in JSON format.*

**Step 3: Second Normal Form (2NF)**

In 2NF, partial dependencies are removed. All non-key attributes must depend on the entire primary key. A new relation is introduced to handle the many-to-many relationship between bookings and seats.

**Revised Tables in 2NF:**

* **users(id, name, email, password, role)**
* **events(id, title, description, date, time, venue, price, capacity, user\_id)**
* **seats(id, event\_id, seat\_number, row, section, status, price)**
* **bookings(id, event\_id, user\_id, ticket\_number, qr\_code, status, total\_amount, booking\_date)**
* **booking\_seats(booking\_id, seat\_id)**

*The booking\_seats table replaces the JSON-based selected\_seats field to ensure atomic values and proper foreign key relationships.*

**Step 4: Third Normal Form (3NF)**

In 3NF, transitive dependencies are removed. All non-prime attributes must depend only on the primary key.

**Final Relations in 3NF**

1. **users**  
   (id, name, email, password, role, email\_verified\_at, remember\_token, created\_at, updated\_at)
2. **events**  
   (id, title, description, date, time, venue, capacity, price, image, status, user\_id, created\_at, updated\_at)
3. **seats**  
   (id, event\_id, seat\_number, row, section, status, price, created\_at, updated\_at)
4. **bookings**  
   (id, user\_id, event\_id, ticket\_number, qr\_code, status, quantity, total\_amount, booking\_date, created\_at, updated\_at)
5. **booking\_seats** *(new table added for full normalization)*  
   (booking\_id, seat\_id)

The database design has been successfully normalized to **Third Normal Form (3NF)** to ensure:

* Elimination of redundancy
* Improved data consistency and integrity
* Scalability and flexibility in querying booking and seating data

Figure:

A diagram of a computer

AI-generated content may be incorrect.