**Milestone 02**



**Spring 2025**

Submitted by:

FAIZAN 2134

ANEES 2168

ZEESHAN 2217

Class Section: **B**

“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:

Eng. Sumayyea Salahuddin

Month Day, Year (26 05, 2025)

Department of Computer Systems Engineering

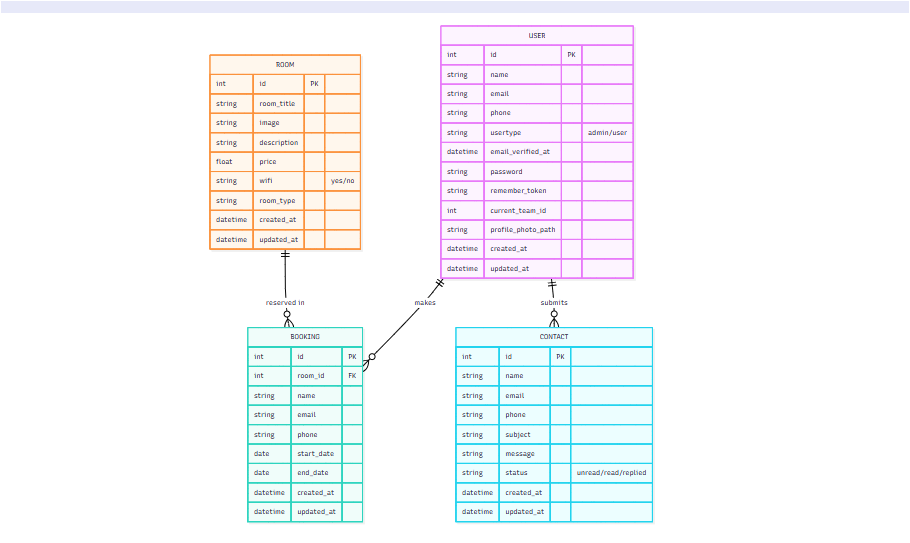
University of Engineering and Technology, Peshawar

### **Relationships Between Tables**:

1. **User (1) -> Booking (Many)**: One user can have many bookings.
2. **Room (1) -> Booking (Many)**: One room can have many bookings.
3. **User (1) -> Contact (Many)**: One user can submit many contact requests.

**Room** and **Contact** have no direct relationship

Relational-schema :



### **Conceptual Schema Details**

#### **1. ROOM Table**

* **Attributes:**
  + id: Primary Key (PK), unique identifier for each room.
  + room\_title: The title or name of the room (e.g., "Luxury Suite").
  + image: Path to an image representing the room.
  + description: A detailed description of the room (features, size, etc.).
  + price: Price per night or booking period for the room.
  + wifi: Indicates if the room has Wi-Fi (yes or no).
  + room\_type: Type of room (e.g., "Single", "Double", "Suite").
  + created\_at: Timestamp when the room was created.
  + updated\_at: Timestamp when the room details were last updated.
* **Relationships:**
  + **One-to-Many** with the **Booking** table: Each room can have many bookings. The room\_id in the **Booking** table references the id of the **Room** table.

#### **2. USER Table**

* **Attributes:**
  + id: Primary Key (PK), unique identifier for each user.
  + name: Full name of the user.
  + email: Email address of the user (unique).
  + phone: Phone number of the user.
  + usertype: Type of user (admin or user).
  + email\_verified\_at: Timestamp when the user's email was verified.
  + password: Hashed password for authentication.
  + remember\_token: Token for remembering the user's session.
  + current\_team\_id: ID of the team associated with the user (optional).
  + profile\_photo\_path: Path to the user's profile photo.
  + created\_at: Timestamp when the user was created.
  + updated\_at: Timestamp when the user details were last updated.
* **Relationships:**
  + **One-to-Many** with the **Contact** table: Each user can submit multiple contact messages. The id of the **User** table is referenced in the **Contact** table.
  + **Many-to-One** with the **Booking** table: A user can make multiple bookings. If desired, a user\_id foreign key could be added to track which user made a particular booking.

#### **3. BOOKING Table**

* **Attributes:**
  + id: Primary Key (PK), unique identifier for each booking.
  + room\_id: Foreign Key (FK) referencing the **Room** table. Each booking is associated with a room.
  + name: Name of the person making the booking.
  + email: Email address of the person making the booking.
  + phone: Phone number of the person making the booking.
  + start\_date: The start date of the booking.
  + end\_date: The end date of the booking.
  + created\_at: Timestamp when the booking was created.
  + updated\_at: Timestamp when the booking details were last updated.
* **Relationships:**
  + **Many-to-One** with the **Room** table: Each booking is associated with a single room. The room\_id in the **Booking** table references the id of the **Room** table.
  + **Many-to-One** (implied) with the **User** table: If desired, a user\_id foreign key can be added to track the user who made the booking.

#### **4. CONTACT Table**

* **Attributes:**
  + id: Primary Key (PK), unique identifier for each contact message.
  + name: Name of the person submitting the contact request.
  + email: Email address of the person submitting the contact request.
  + phone: Phone number of the person submitting the contact request.
  + subject: Subject or title of the contact message.
  + message: Content or body of the contact message.
  + status: Current status of the message (e.g., unread, read, replied).
  + created\_at: Timestamp when the contact entry was created.
  + updated\_at: Timestamp when the contact entry was last updated.
* **Relationships:**
  + **Many-to-One** with the **User** table: Each contact message is submitted by a single user. The id of the **User** table is referenced in the **Contact** table.

### **Diagram Explanation:**

* **Room ↔ Booking**: One room can be booked multiple times, but a booking is made for a specific room. The relationship between the **Room** and **Booking** tables is a **One-to-Many**.
* **User ↔ Contact**: A user can submit multiple contact requests. This forms a **One-to-Many** relationship between the **User** and **Contact** tables.
* **User ↔ Booking** (Implied): Although not shown in the diagram, a **User** can make multiple bookings, and each booking would be linked to a user (through user\_id if added). This would be a **One-to-Many** relationship.
* **Contact ↔ User**: A contact message is submitted by a user, establishing a **Many-to-One** relationship from **Contact** to **User**.

Normalization:

1. Address is multi-value attributes so each entity of this is written in unique row.(1NF)
2. No partial dependency in this table (2NF)
3. No transitive dependency