# 2. Database Design & Data Modeling

## 1. Entity-Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) illustrates the structure of the database, showing the main entities (tables), their attributes (fields), and the relationships between them. This model ensures data is organized logically, and it clarifies how different parts of the system are connected.

### Entities:

**- User  
- Event  
- Reminder**

### Relationships:

- A User can create multiple Events (One-to-Many relationship).  
- An Event can have multiple Reminders (One-to-Many relationship).

### ER Diagram (Textual Representation)

+-----------------+ +-------------------+  
| User | | Event |  
+-----------------+ +-------------------+  
| - UserID (PK) |<------>| - EventID (PK) |  
| - UserName | | - EventTitle |  
| - Email | | - EventDate |  
| - Password | | - CountdownTime |  
+-----------------+ | - UserID (FK) |  
 +-------------------+  
  
+---------------------+  
| Reminder |  
+---------------------+  
| - ReminderID (PK) |  
| - EventID (FK) |  
| - ReminderTime |  
+---------------------+

### Explanation:

- The User entity holds information about users registered in the app.  
- The Event entity stores event details linked to the users.  
- The Reminder entity contains reminder information tied to events.

## 2. Logical Schema

### User Table

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Description | Constraints |
| UserID | INT | Unique identifier for user | Primary Key (PK) |
| UserName | VARCHAR | Name of the user | Not Null |
| Email | VARCHAR | Email address | Unique, Not Null |
| Password | VARCHAR | User password (hashed) | Not Null |

### Event Table

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Description | Constraints |
| EventID | INT | Unique identifier for event | Primary Key (PK) |
| EventTitle | VARCHAR | Title or name of the event | Not Null |
| EventDate | DATE | Date of the event | Not Null |
| CountdownTime | TIME | Countdown time until the event starts | Optional |
| UserID | INT | Reference to the user who owns event | Foreign Key (FK) |

### Reminder Table

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Description | Constraints |
| ReminderID | INT | Unique identifier for reminder | Primary Key (PK) |
| EventID | INT | Reference to the related event | Foreign Key (FK) |
| ReminderTime | TIME | Time to send the reminder | Not Null |

## 3. Physical Schema & Normalization

- Normalization: The database is normalized up to Third Normal Form (3NF). This ensures data redundancy is minimized, and data integrity is maintained.  
 - 1NF (First Normal Form): All columns contain atomic values.  
 - 2NF (Second Normal Form): No partial dependency on primary keys.  
 - 3NF (Third Normal Form): No transitive dependency exists.  
  
- Indexes:  
 - Indexes are created on the Primary Keys (UserID, EventID, ReminderID).  
 - Foreign keys (UserID in Event table and EventID in Reminder table) are indexed for faster JOIN operations.

## Diagram Summary

This database design ensures:  
- Data integrity between users, events, and reminders.  
- Flexibility for users to manage multiple events and set multiple reminders.  
- Easy scalability for future features such as notifications, event sharing, or additional user preferences.