### ****A. Explanation of How the Final Database Achieved 3NF****

The process of normalization to 3rd Normal Form (3NF) involves systematically organizing data to reduce redundancy and ensure data integrity. Here's how the final database design achieved 3NF, starting from raw data:

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

The unnormalized data might look like this:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BookingID | UserID | UserName | Email | EventID | EventName | Tickets | TotalPrice | AdultPhoto | Location | EventPrice | ... |
| 10 | 6 | User | user@gmail.com | 6 | A Fairy Christmas | 20 | 20000 | NULL | Texas | 1000 | ... |

Problems in UNF:

1. Repeating groups (e.g., user and event information repeated in multiple bookings).
2. Lack of atomic values (e.g., combining event and location details in one table).

#### ****Step 2: 1st Normal Form (1NF)****

To achieve 1NF:

1. Eliminate repeating groups by ensuring each field contains atomic values.
2. Separate data into different tables for users, events, and bookings.

The resulting tables:

* **Users**: Contains only user-specific data.
* **Events**: Contains only event-specific data.
* **Bookings**: Contains transactional booking data.

#### ****Step 3: 2nd Normal Form (2NF)****

To achieve 2NF:

1. Eliminate partial dependencies. All non-primary key attributes must depend on the **entire primary key**.
2. Move attributes like Location, EventPrice, and SeatingType to the **Events** table because they describe events and not bookings.

The intermediate structure:

* **Users**:
  + Primary Key: UserID
  + Attributes: UserName, Email, Phone, Password, IsAdmin
* **Events**:
  + Primary Key: EventID
  + Attributes: Name, Description, DateTime, Location, Price, Image, IsSupervised, SeatingType
* **Bookings**:
  + Primary Key: BookingID
  + Foreign Keys: UserID, EventID
  + Attributes: Tickets, AdultSeat, TotalPrice, AdultPhoto, BookingDate

#### ****Step 4: 3rd Normal Form (3NF)****

To achieve 3NF:

1. Eliminate transitive dependencies, where non-key attributes depend on other non-key attributes.
2. Verify that all non-primary key attributes are dependent only on the primary key and not on other attributes.

Changes made:

* The Bookings table ensures no dependency on UserName or EventName (now stored in their respective tables).
* Attributes like Location and SeatingType are part of the **Events** table to prevent redundancy.

### ****Final Database Design****

1. **Users Table**:
   * Stores unique user information.
   * Avoids redundancy (e.g., user details do not repeat in bookings).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id (PK) | name | email | phone | password | is\_admin |
| 6 | Zeus | user@gmail.com | 0123456789 | ... | 0 |

1. **Events Table**:
   * Stores event-specific details to avoid repeating the same data in bookings.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id (PK) | name | description | date\_time | location | price | image | is\_supervised | seating\_type |
| 6 | A Fairy Merry Christmas | Come explore the fairy | 024-12-25 12:00:00 | Texas | 1000.0 | fairychristmas.jpeg | 0 | Without Tables |

1. **Bookings Table**:
   * Stores transactional data.
   * Links users and events via foreign keys, ensuring integrity.

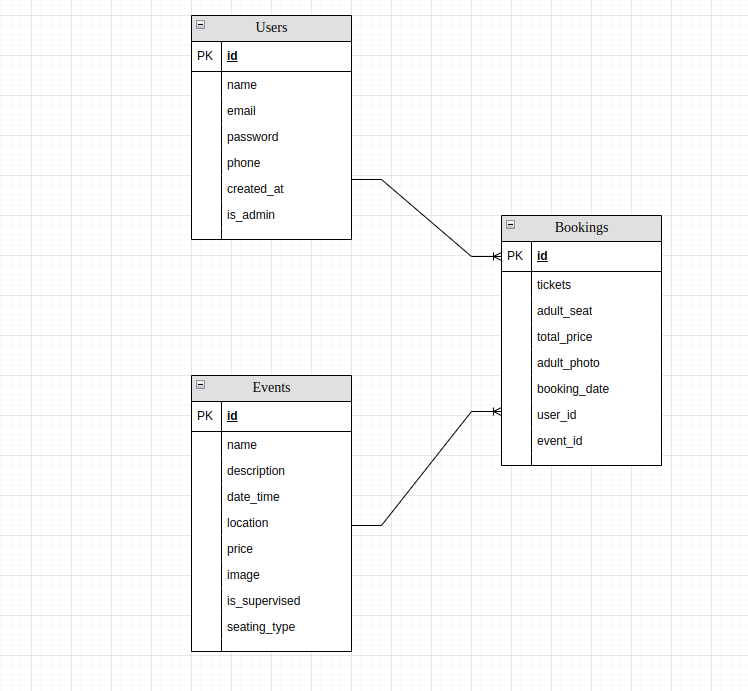
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id (PK) | user\_id (FK) | event\_id (FK) | tickets | adult\_seat | total\_price | adult\_photo | booking\_date |
| 10 | 6 | 6 | 20 | 3 | 20000 | adult\_father.png | 2024-11-20 13:51:00 |

The final database achieves 3NF because:

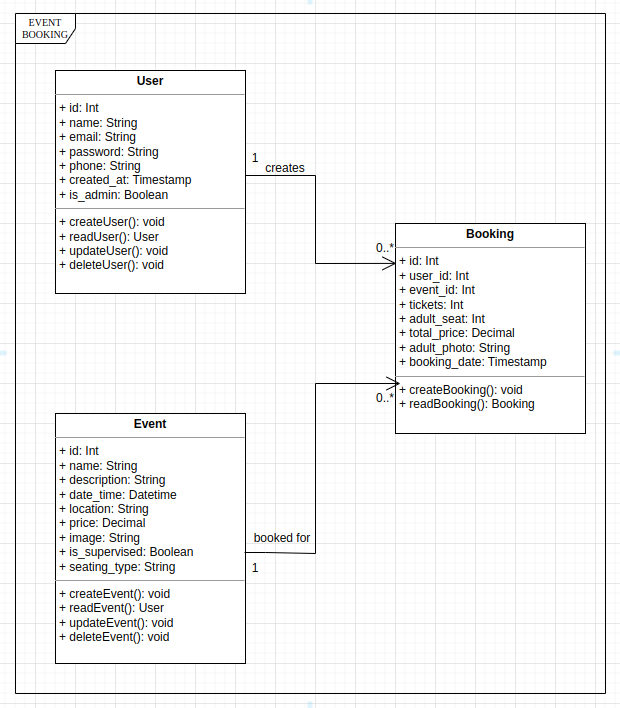
1. **Atomic Values:** Each table has atomic attributes.
2. **No Partial Dependencies:** Non-key attributes depend only on the whole primary key.
3. **No Transitive Dependencies:** Data is separated into logical entities (Users, Events, and Bookings).

This structure ensures minimal redundancy, data integrity, and scalability.

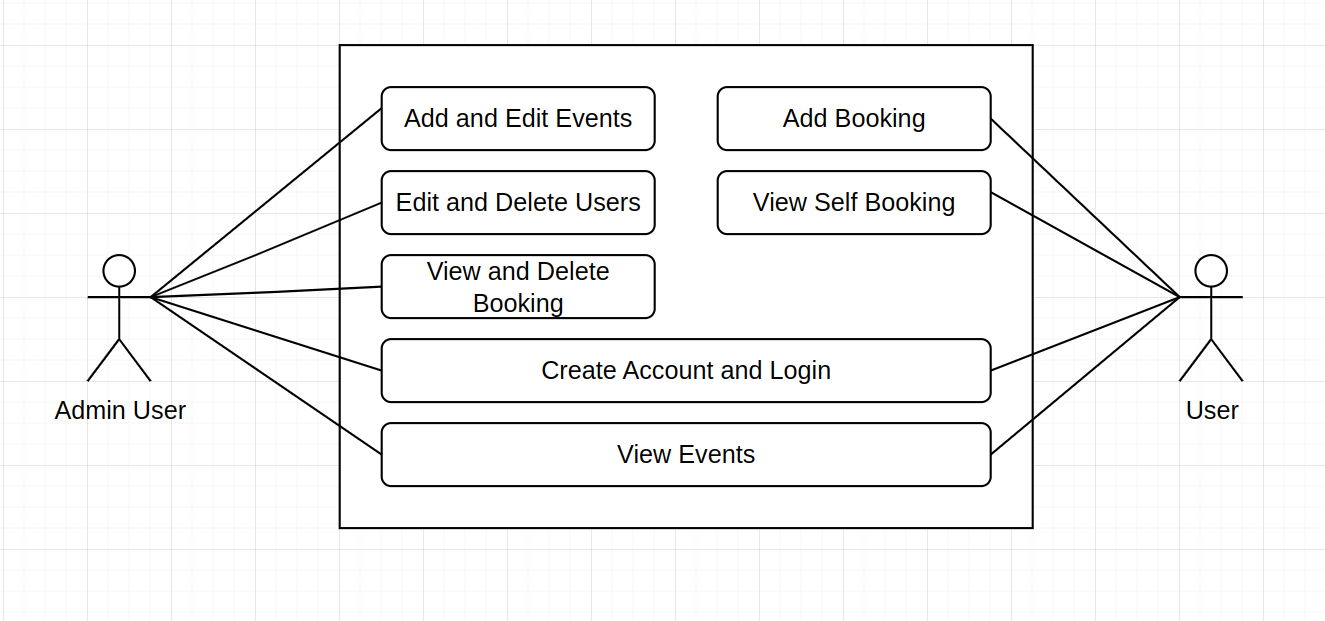
### ****B. Entity Relationship Diagram****



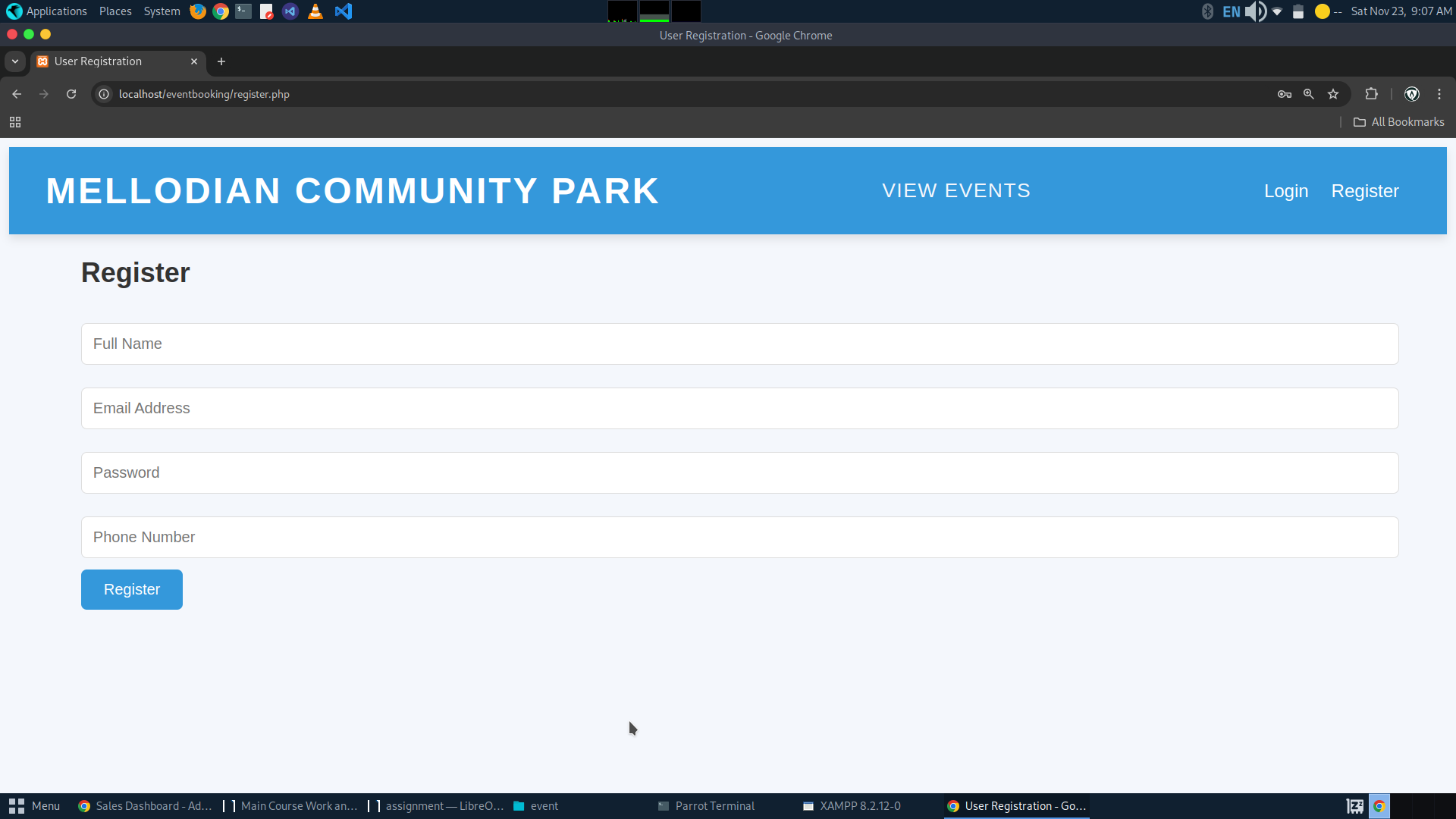
### ****C. Unified Modeling Language Class Diagram****

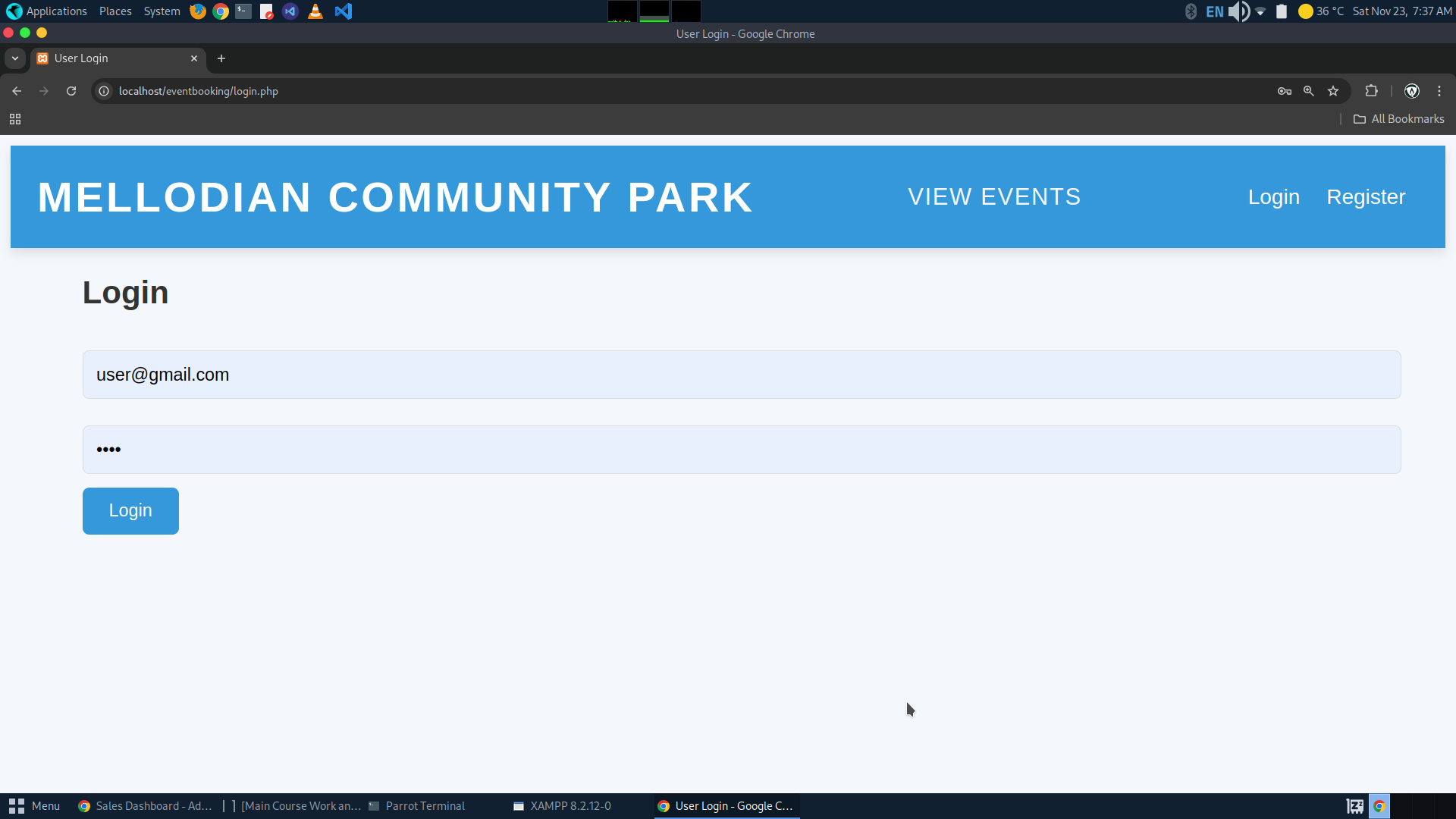


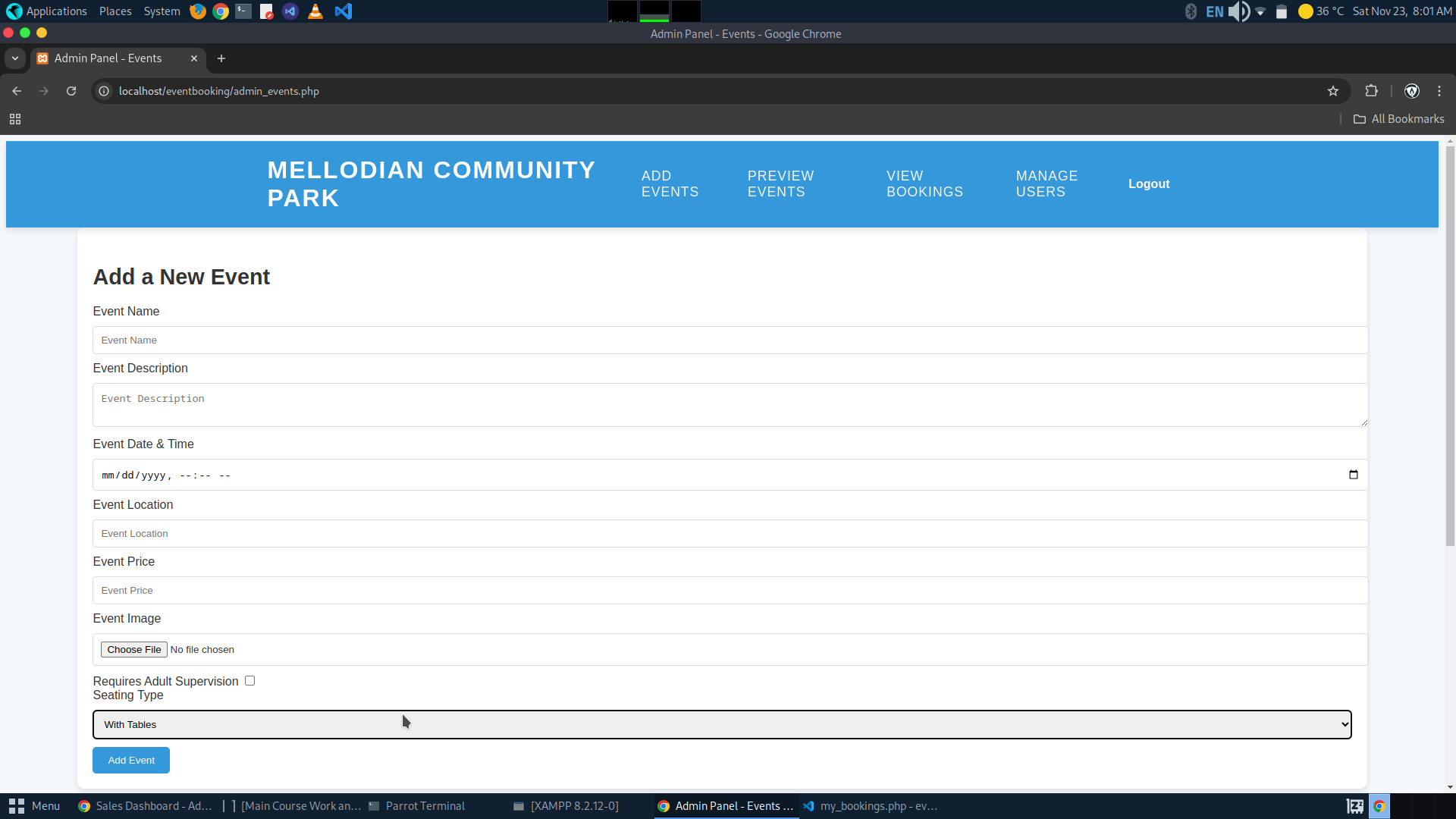
### ****D. Use Case Diagram****

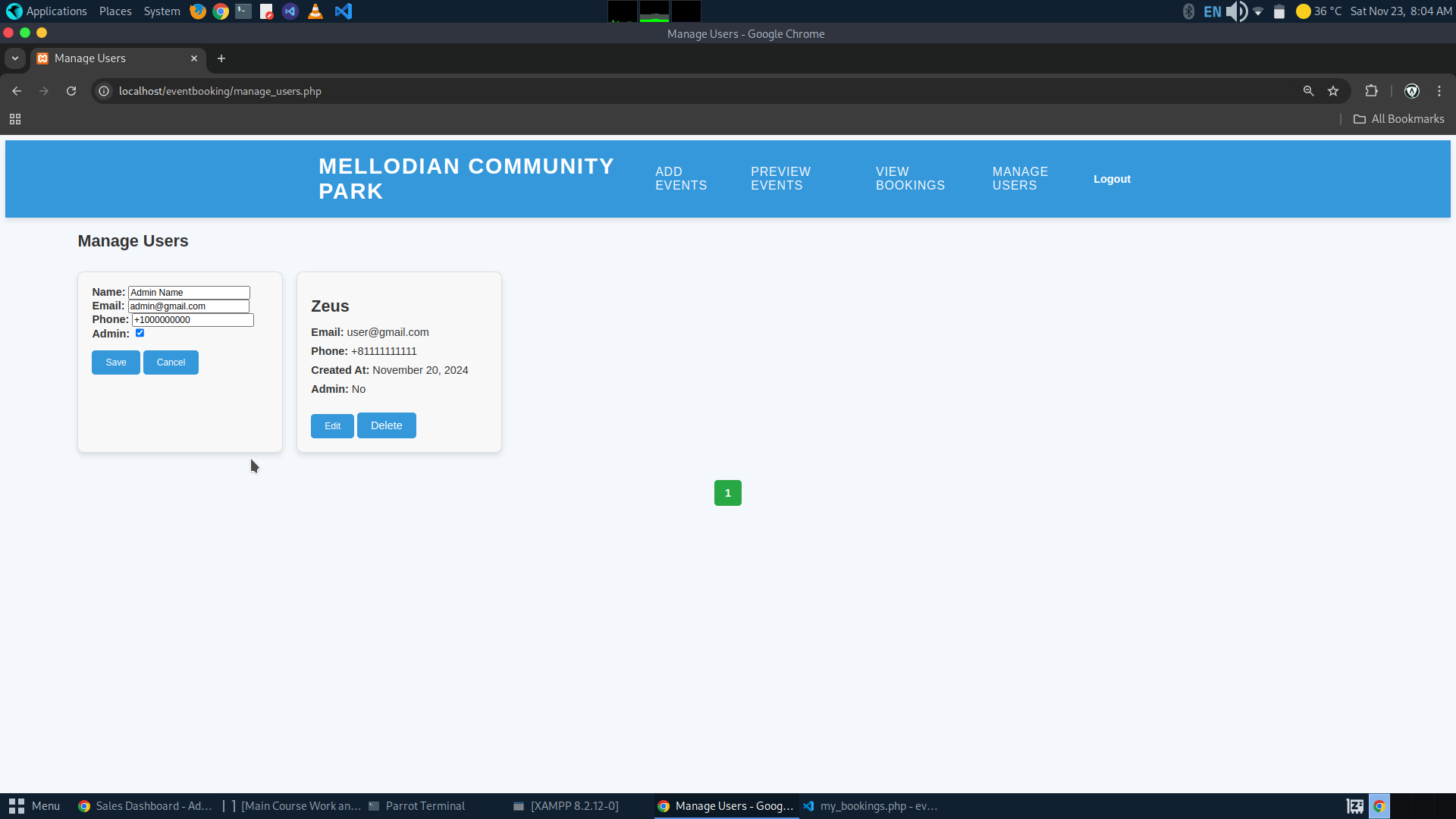


### ****E. User Interface Front End Forms****

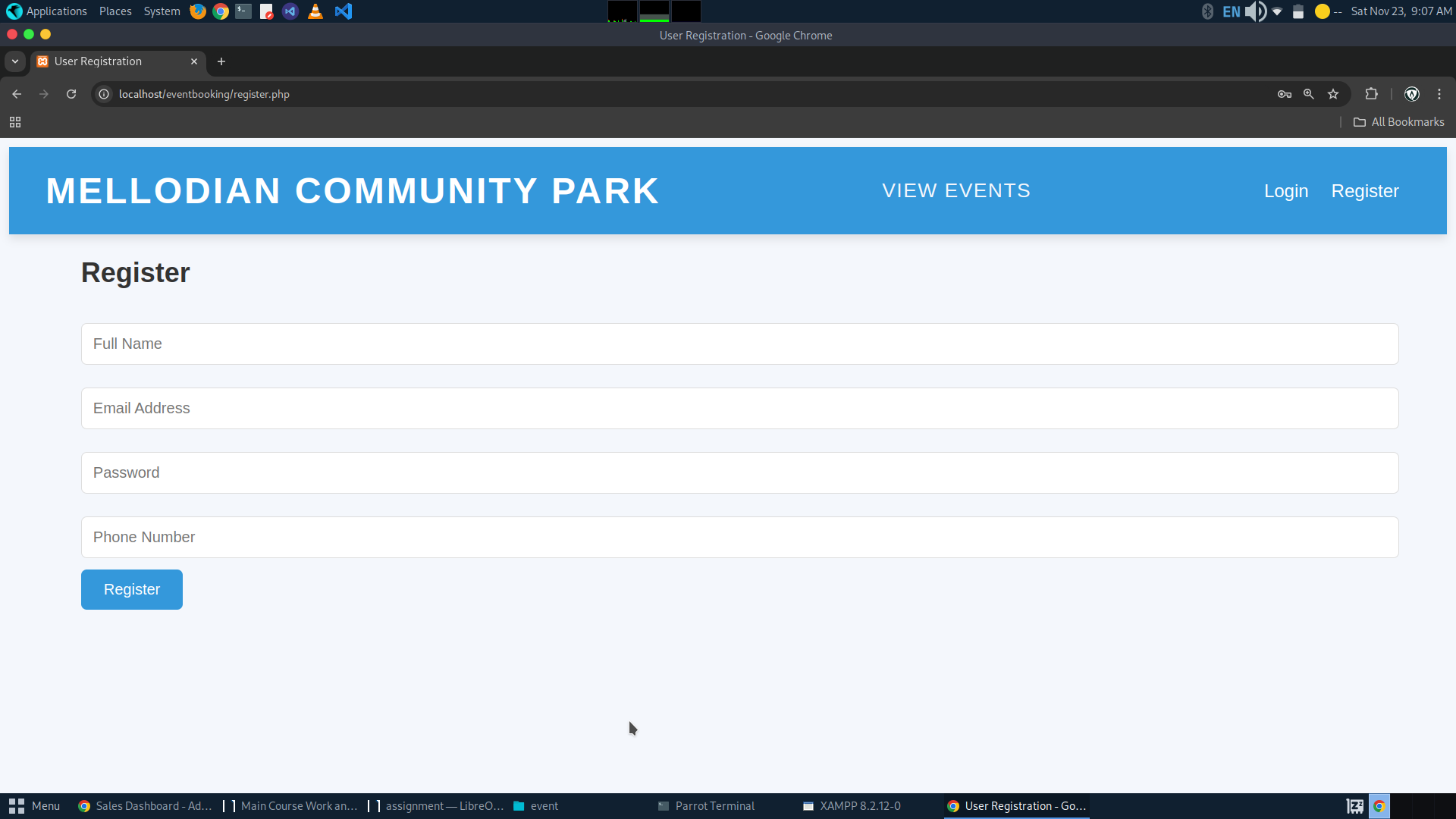
**1.1. Registration Page**

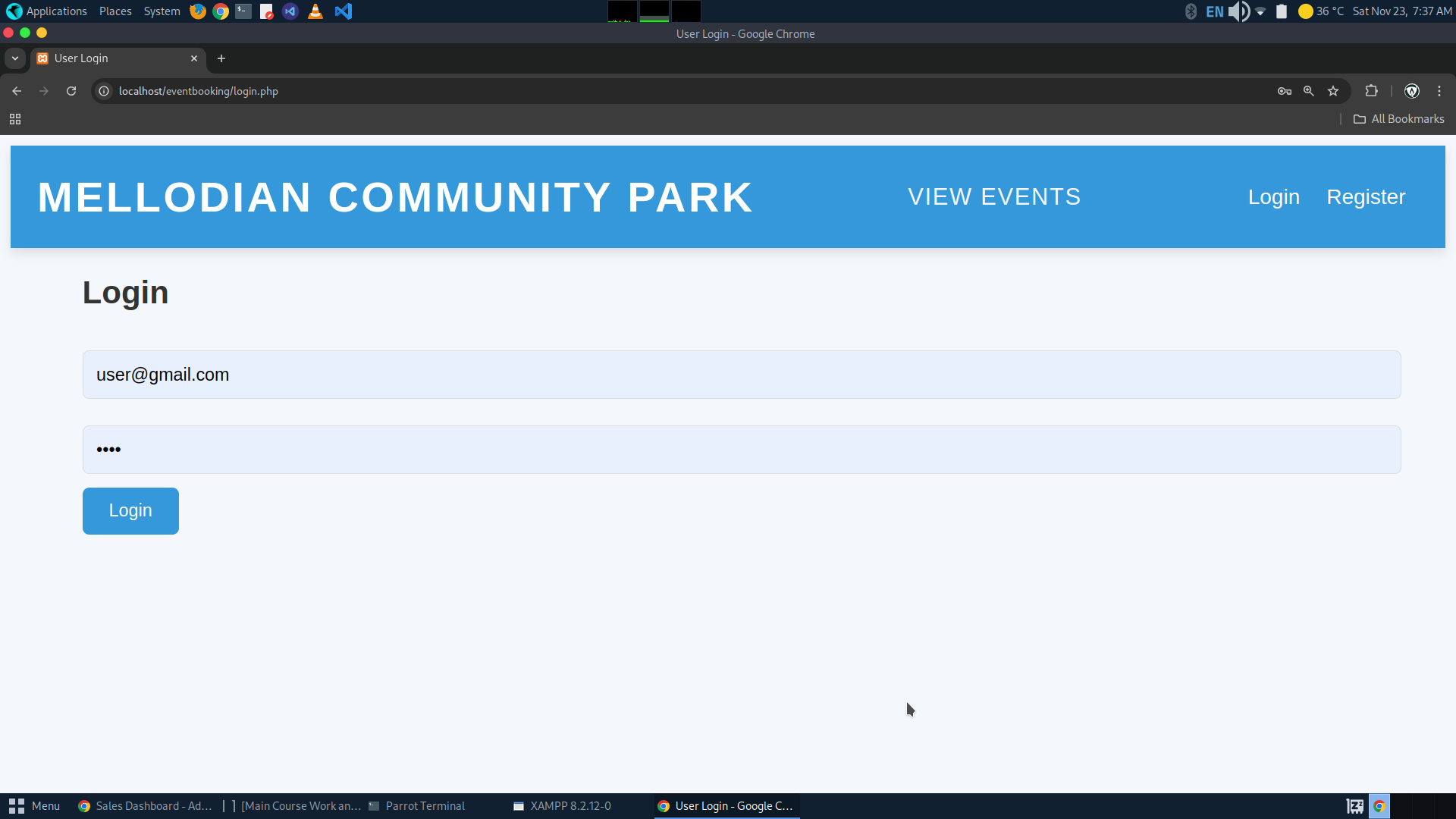
**1.2. Login** **Form**

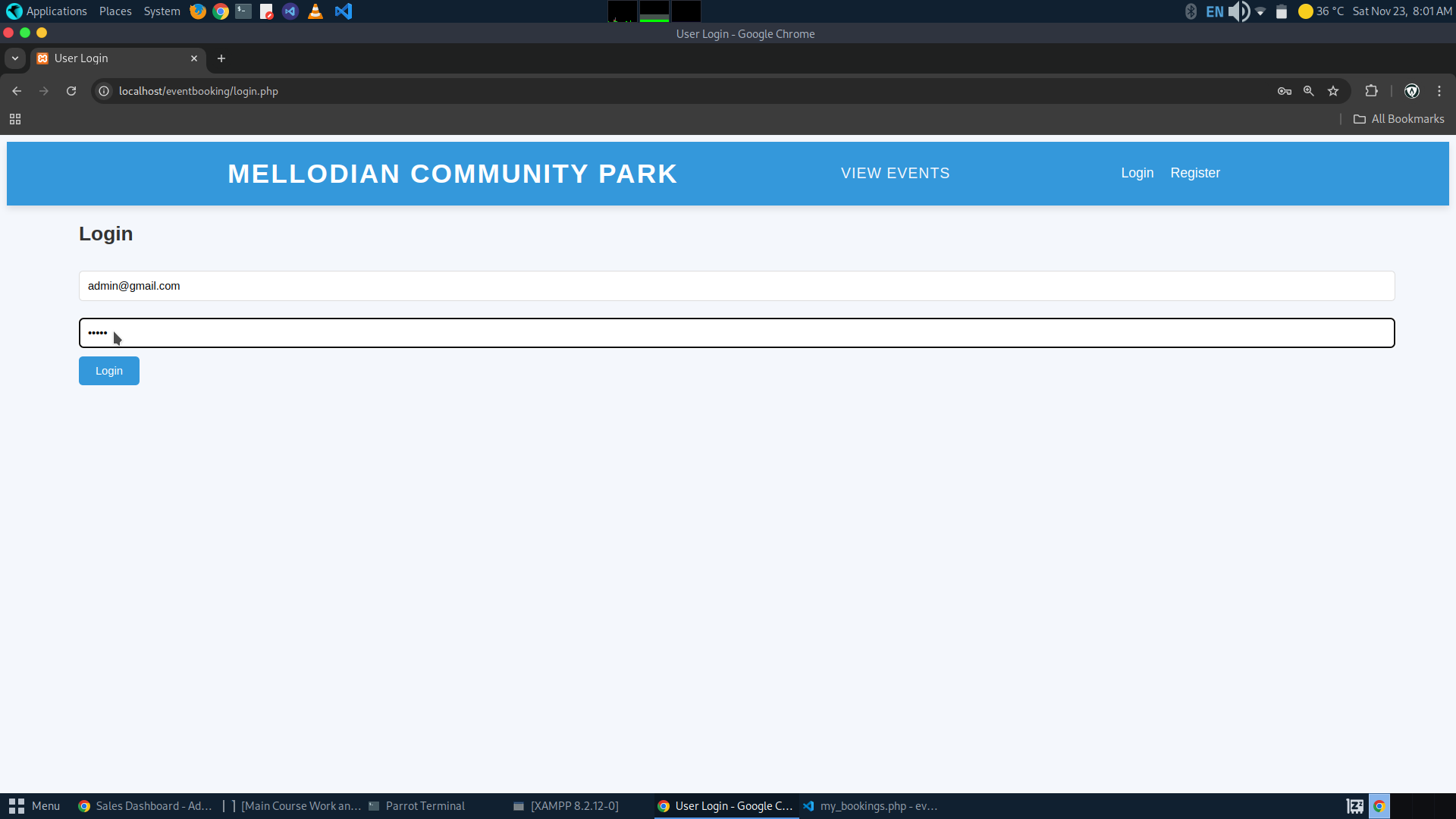
**1.3. Admin Add Event Form**

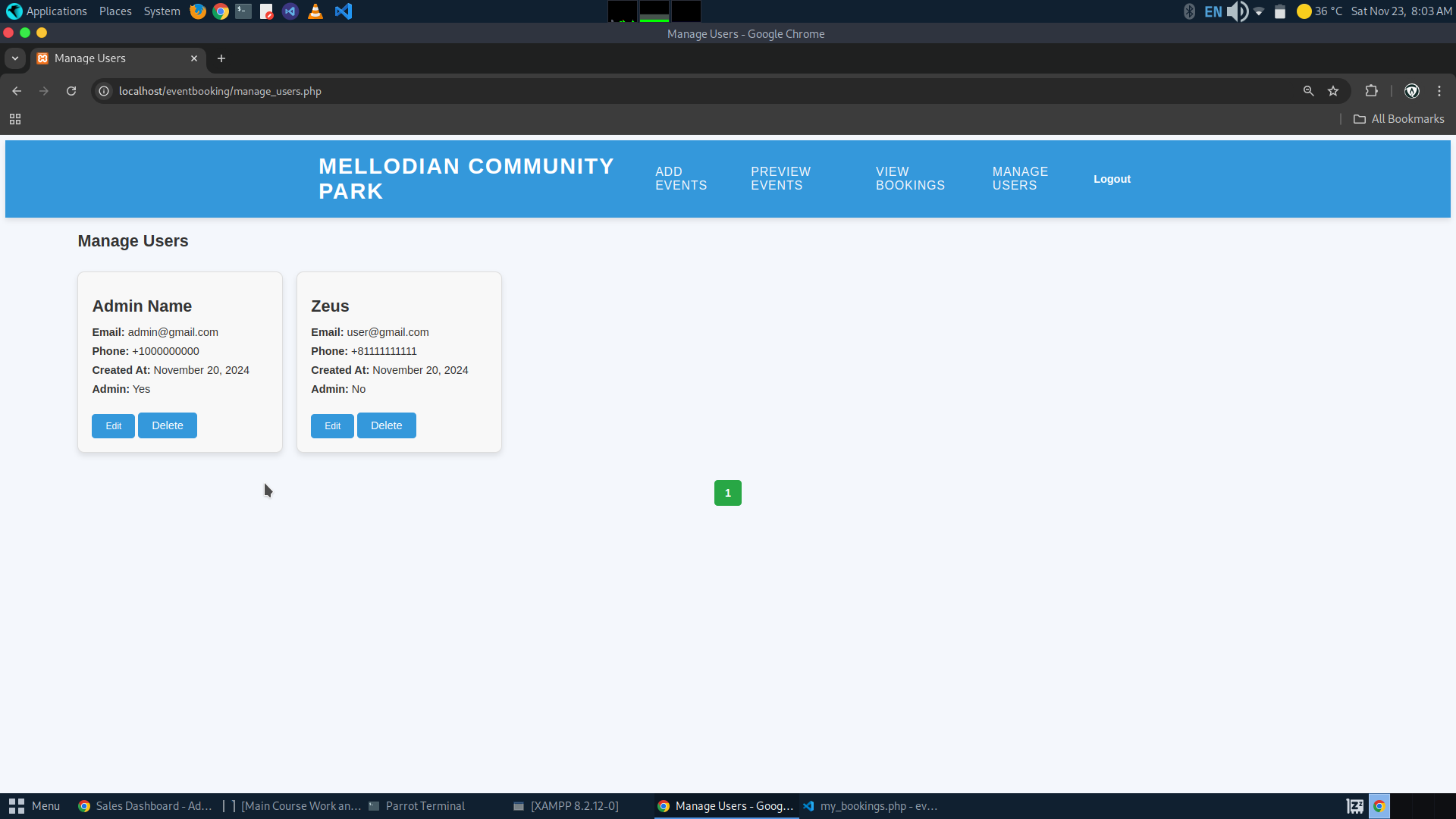
**1.4. Admin Edit Users Form**

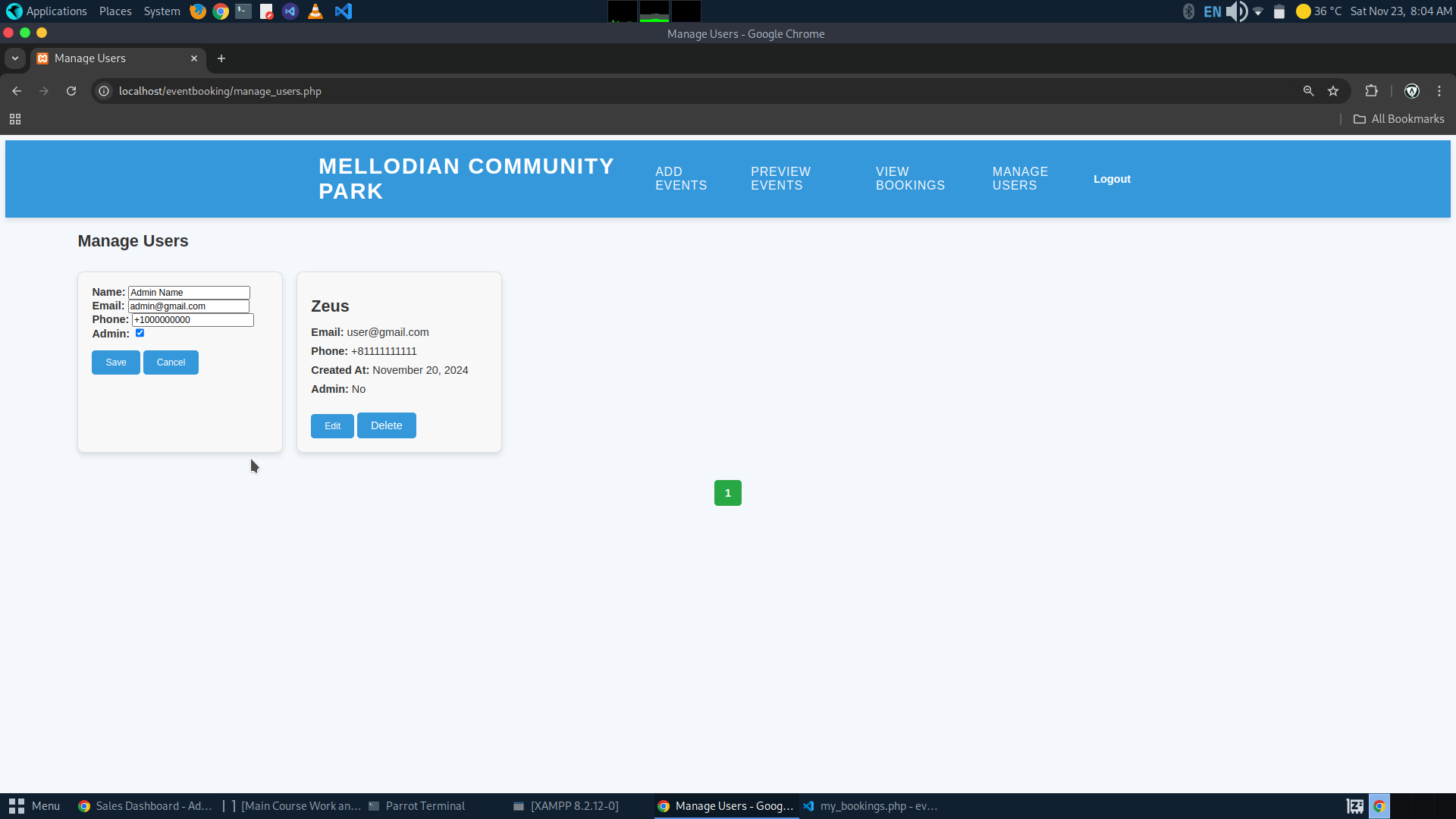
**PART 2 (CODE IMPLEMENTATION):**

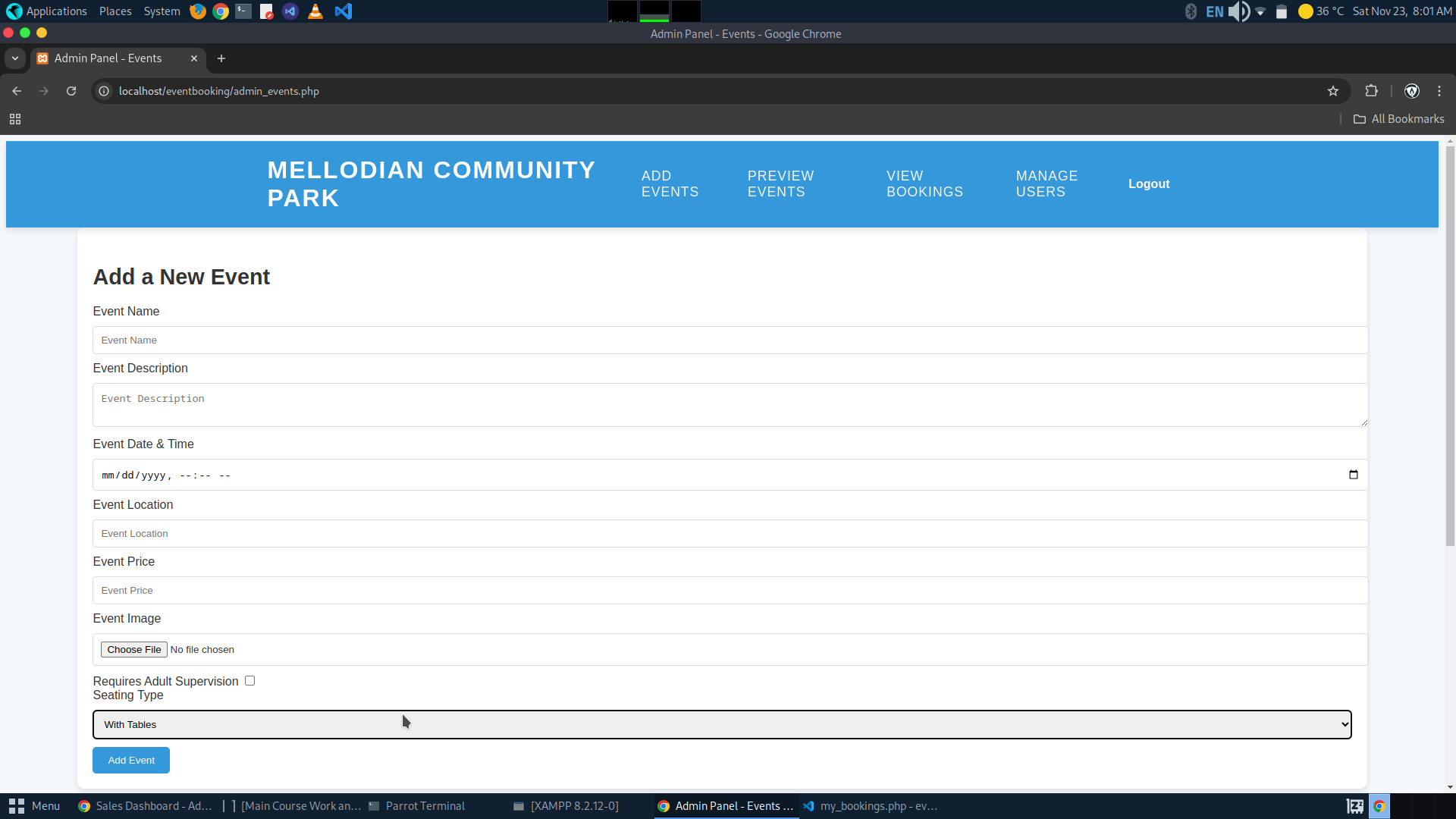
**2.1. Registration Form for Collection of User Details and Credentials**

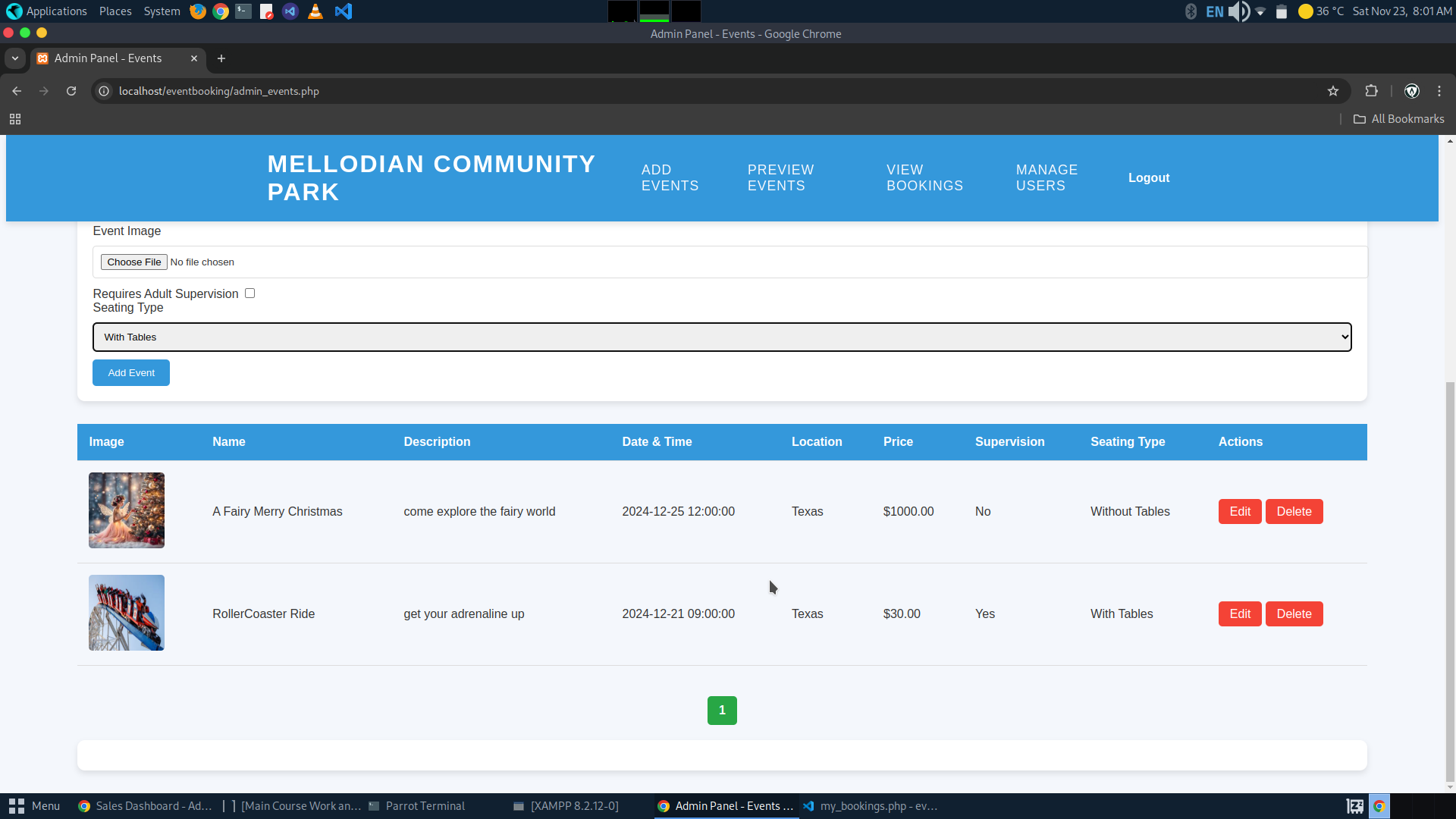
**2.2. User Login Form**

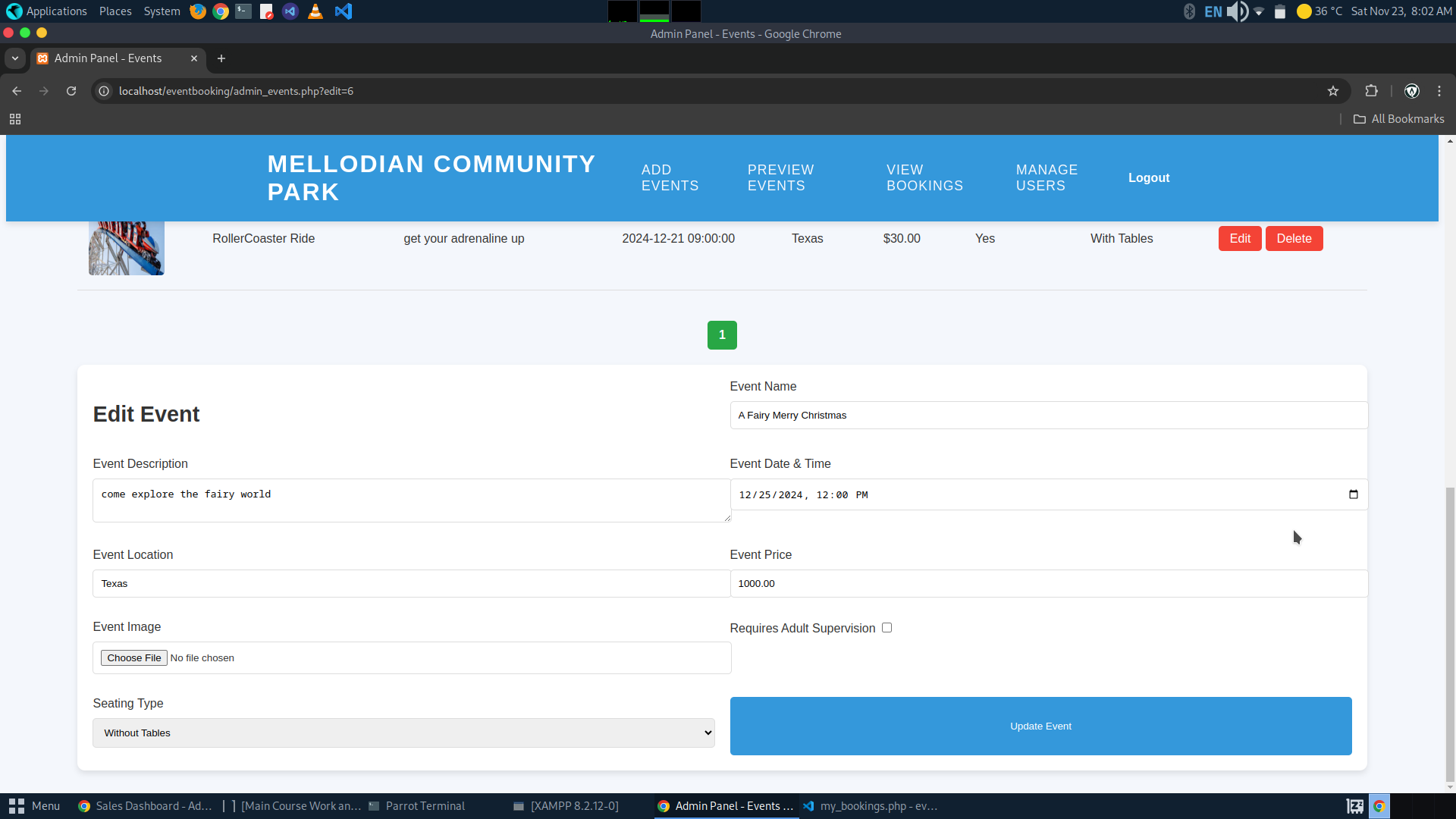
**2.3. Admin Login Form**

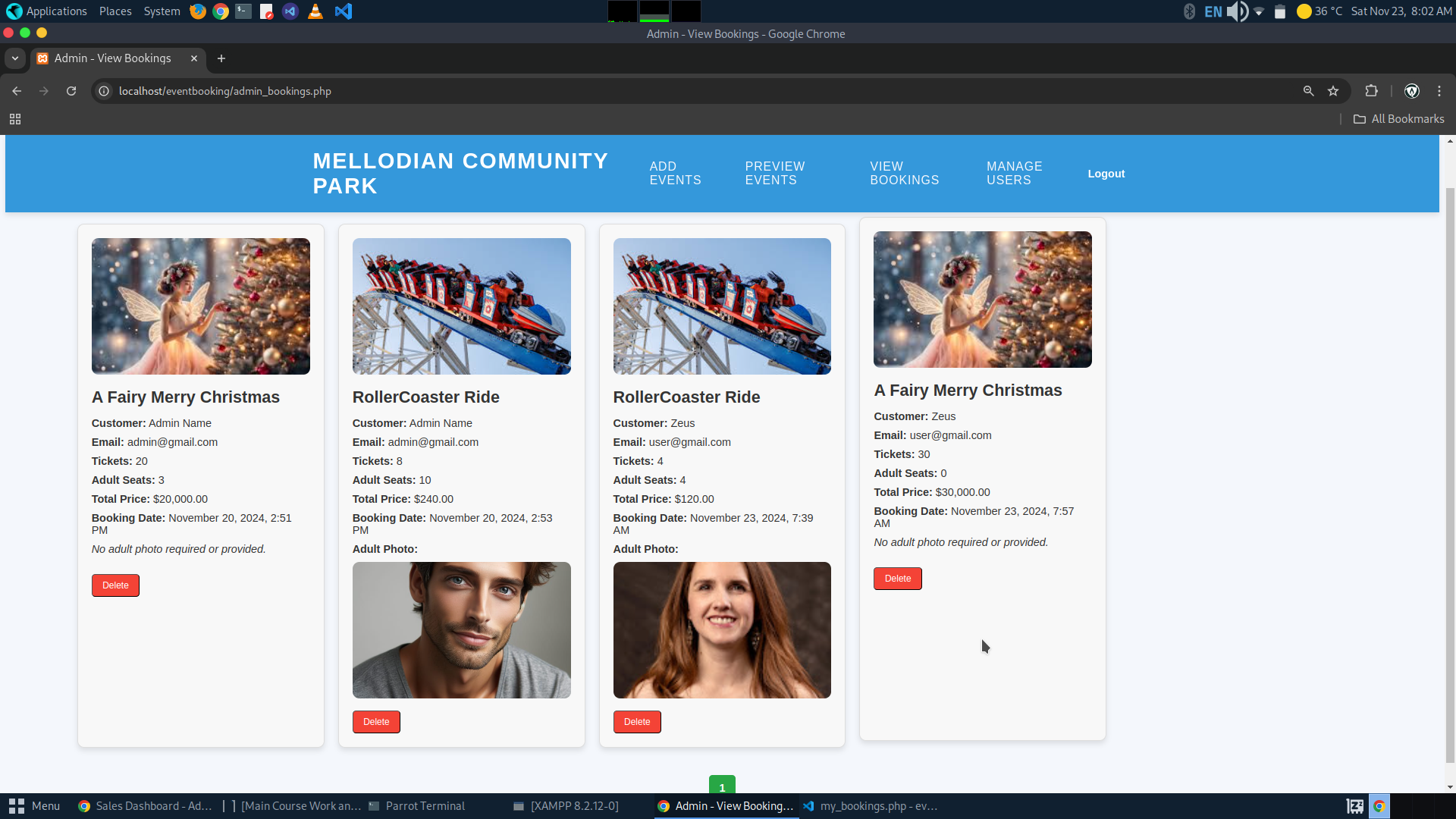
**2.4. Admin Page Listing all Users**

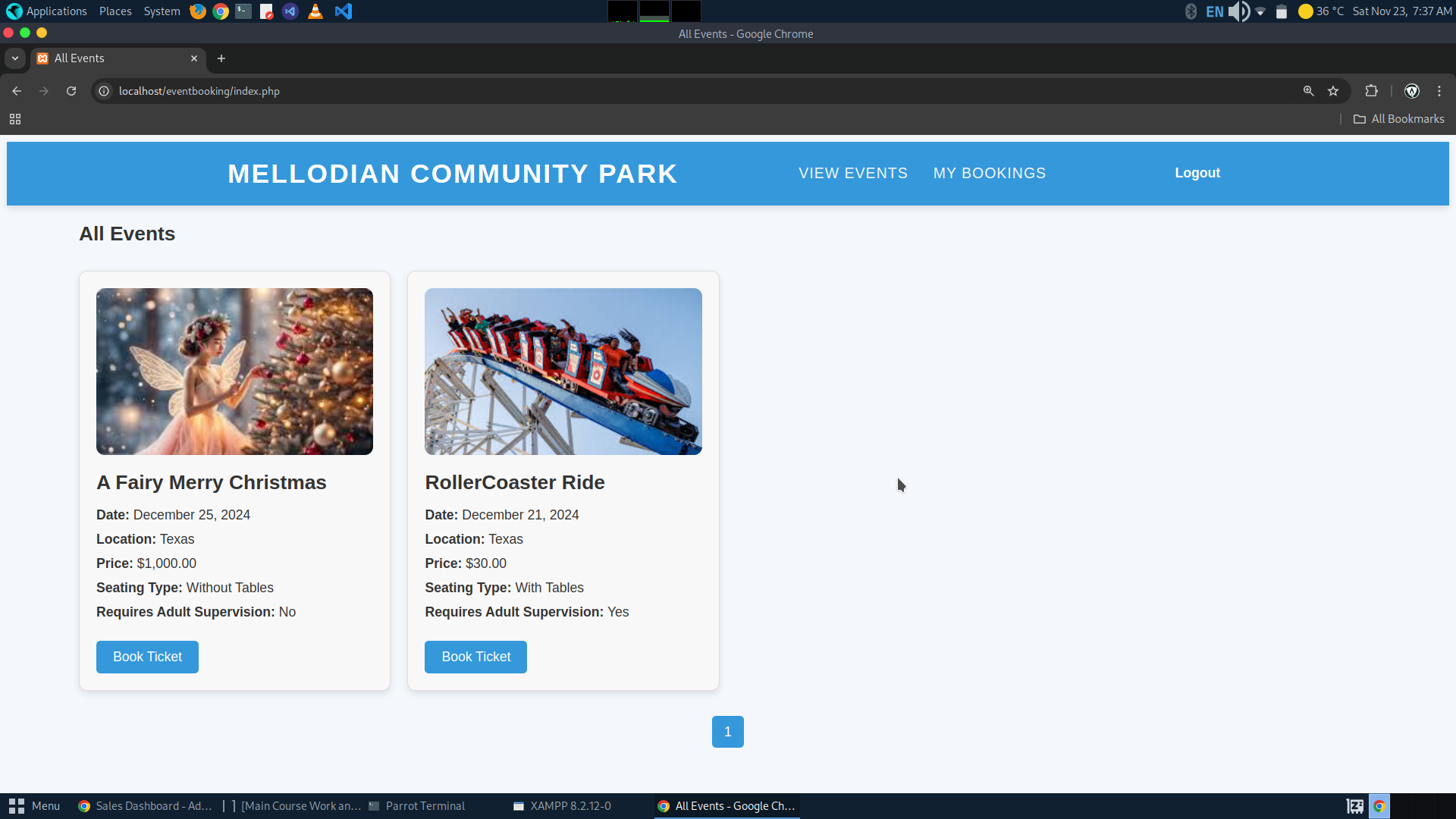
**2.5. Admin Page for Editing Users**

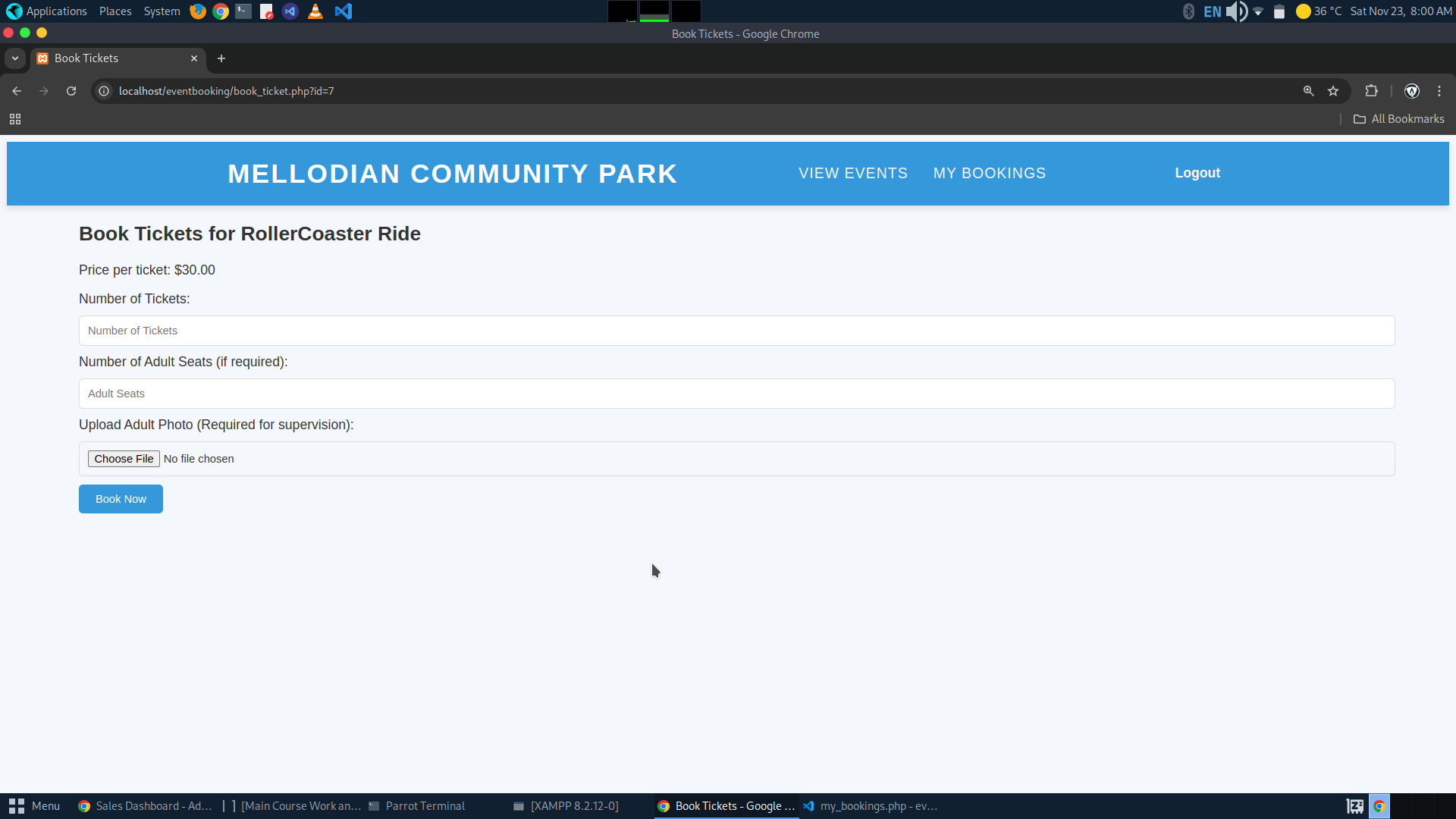
**2.6. Admin Page for Adding Events**

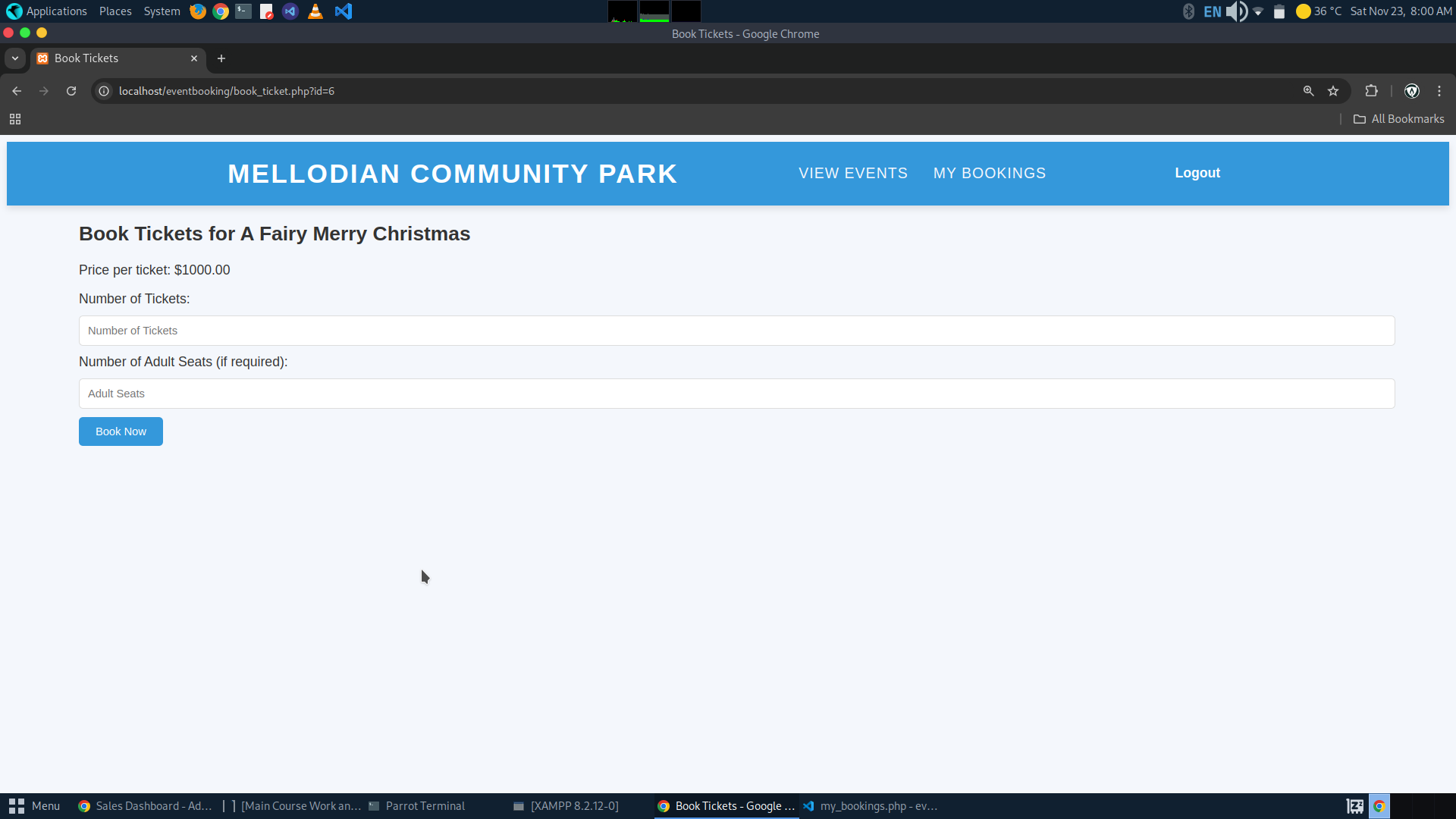
**2.7. Admin Page for Listing Events**

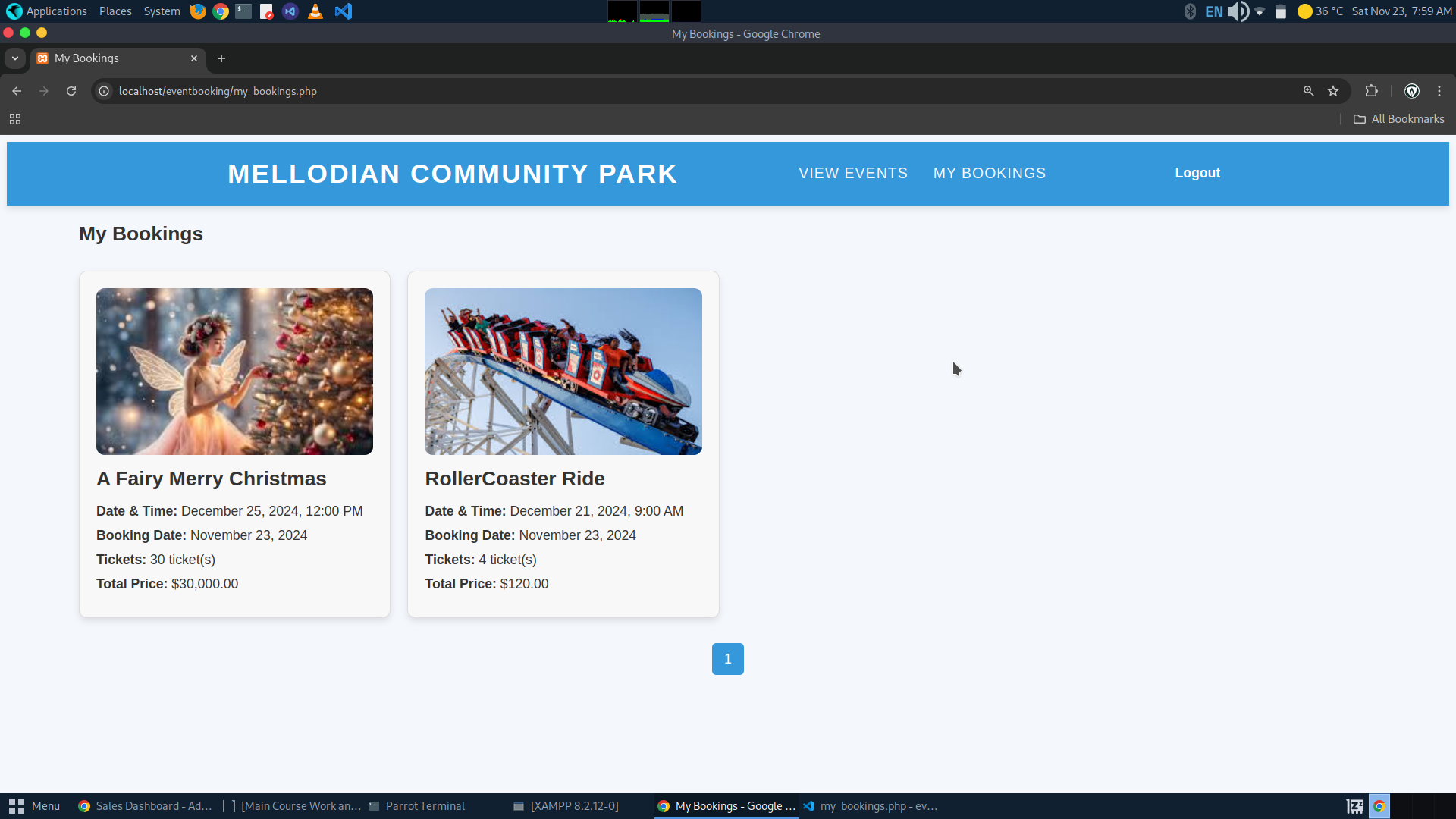
**2.8. Admin Page for Editing Events**

**2.9. Admin Page for Viewing All Bookings (with or without adult supervision)**

**2.10. User Page for Viewing All Events**

**2.11. User Page for Booking Event that requires Adult Supervision**

**2.12. User Page for Booking Event that does not require Adult Supervision**

**2.13. User Page for Viewing All Bookings**