**LABORATORY EXERCISE 3**

**SYSTEM DEVELOPMENT BASED ON PROTOTYPE – LOGIN AND DASHBOARD IMPLEMENTATION**

# Learning Objectives

By the end of this laboratory exercise, students should be able to:

* Implement a login system based on the approved prototype and system design.
* Develop a dashboard interface that meets functional and user experience requirements.
* Integrate user authentication and session management into the system.
* Apply basic security measures such as input validation and password hashing.
* Conduct testing and debugging to refine system functionality.
* Present and document the developed system with supporting screenshots and explanations.

# Prerequisite student experiences and knowledge

Students should have a foundational knowledge of System prototyping and workflow design. Database design and entity-relationship diagrams (ERD). Programming fundamentals in PHP, Laravel, Python, or other programming languages, as well as session management and user authentication principles.

# Background

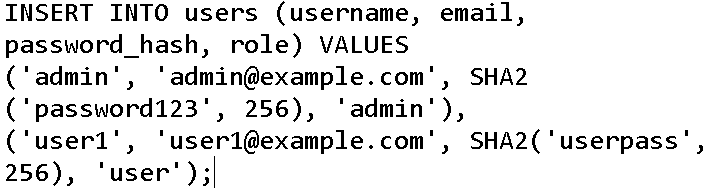
A functional system prototype is a crucial step in system development, allowing developers to test and refine key functionalities. The login system and dashboard serve as users' entry points and control centers, requiring proper authentication and navigation design. Adequate security measures must be implemented to protect user credentials and data integrity.

# Materials/Resources

* PC/Internet
* PC with an Internet connection
* Development environment (XAMPP, WAMP, Laravel Homestead, or equivalent)
* Code editor (VS Code, Sublime Text, PHPStorm, or others)
* Database system (MySQL, PostgreSQL)
* Wireframing tool (Figma, Balsamiq, Lucidchart, or others)

# Laboratory Activity

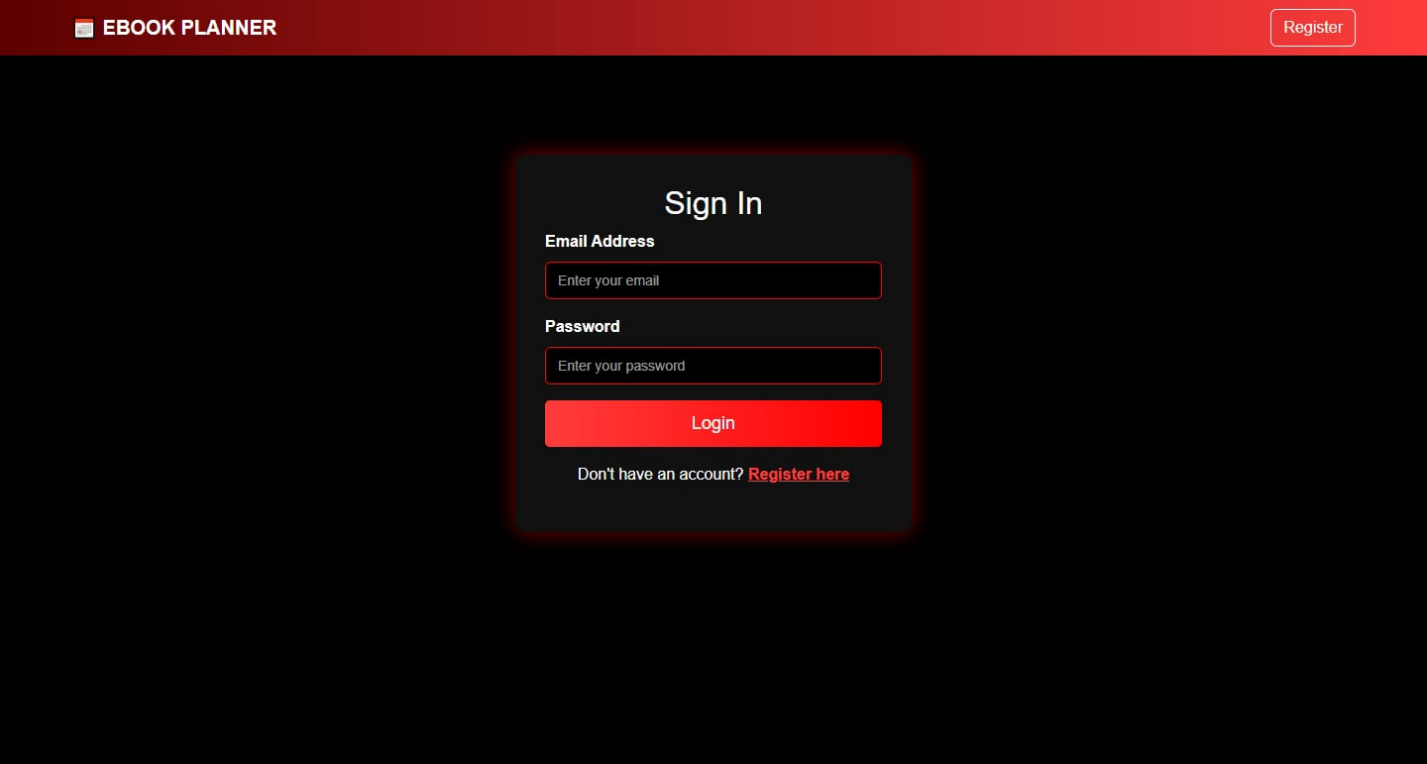
1. **Review of Prototype and System Design**
   * Open and review your approved system prototype from previous exercises.
   * Identify the login system workflow and dashboard structure.
   * Ensure that the database schema supports user authentication and role-based access.
2. **Prototyping Phase**
   * **Wireframe and UI Design**
     + Create a low-fidelity wireframe of the login page and dashboard using:
     + Figma, Balsamiq, Lucidchart, or hand-drawn sketches or other tools.
     + Define the user flow and navigation between components.
   * **Database Schema Setup**
     + Create a user table with the following fields: **user\_id, username, email, password, role, created\_at**
     + Insert sample users:



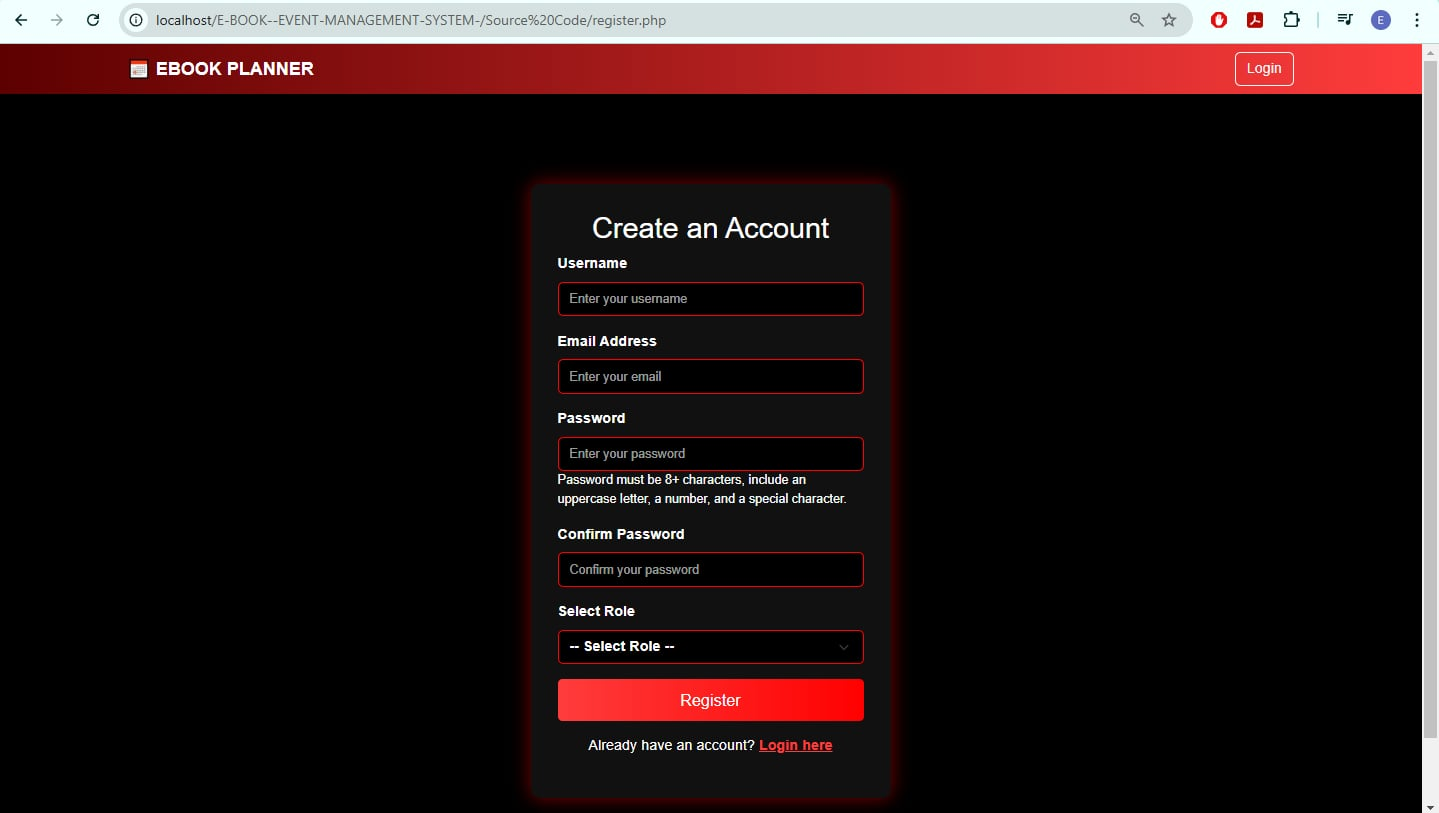
1. **System Implementation**
2. Login System Implementation
   * Create a login form
   * Handle login logic
3. Dashboard Implementation
   * Create a **dashboard page** with dynamic content based on user roles
4. **Testing and Debugging**
   * Conduct unit testing for login validation, session management, and redirections
   * Test invalid inputs (e.g., Invalid Username/Password, empty fields).
   * Verify database interactions and ensure the password is hashed and stored correctly.
5. **Documentation and Submission**
   * Prepare a **report** including the following:
     + Screenshots of the login and dashboard of your group works
     + Database schema diagram
     + Code snippets for authentication and session management.
     + Challenges encountered and solutions applied.
6. **Submission Guidelines**
   * Submit
     + The source code must be in GitHub with a movement of version controls and include the following:
       1. Database dump (.sql file)
       2. Prototype screenshots
       3. Documentation report (PDF format).

# Results/Output *(At least two samples)*

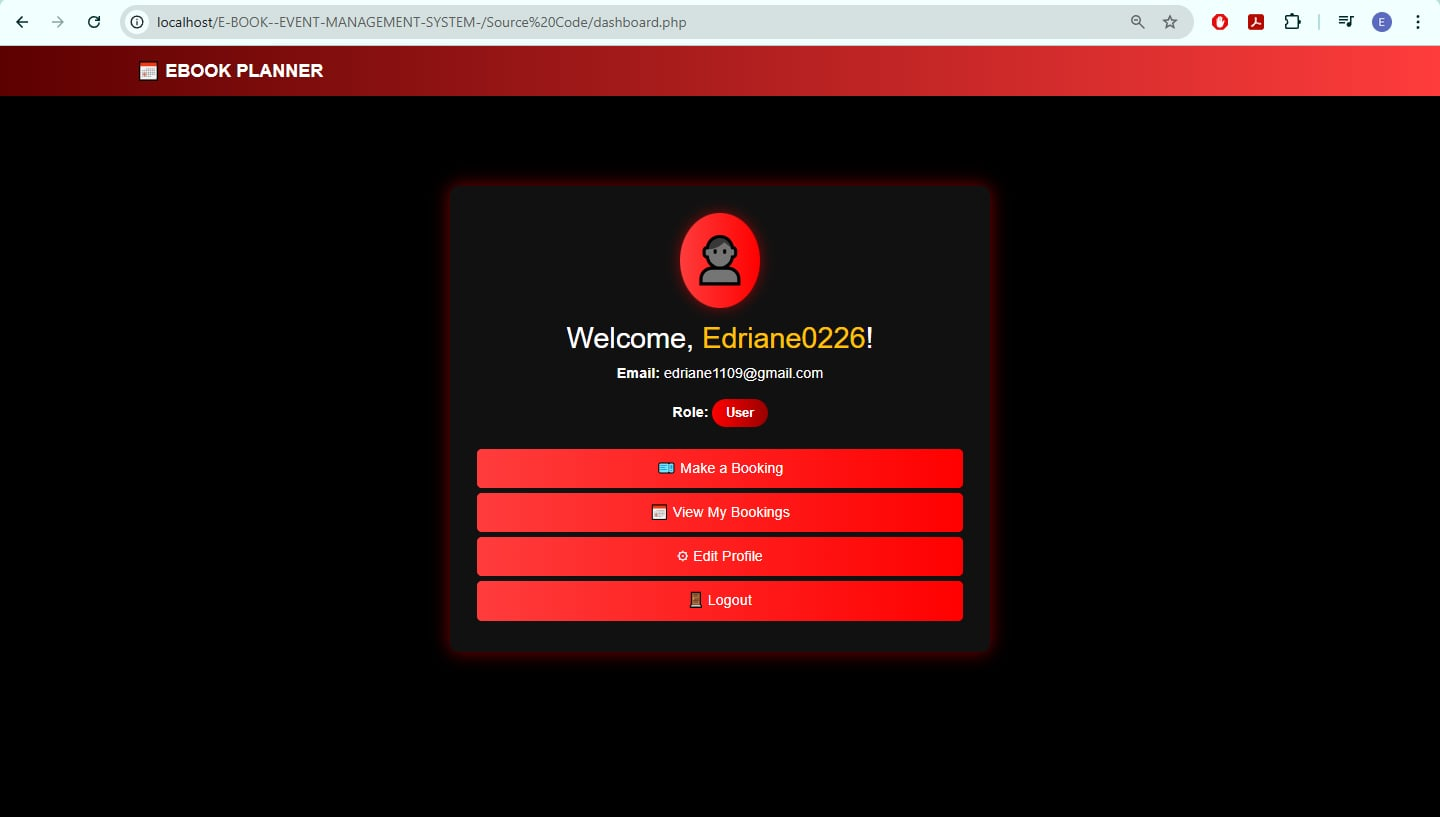
**Log-in**

****

**Register**

****

**Dashboard**



**GitHub Link**

https://github.com/Edriane0226/E-BOOK--EVENT-MANAGEMENT-SYSTEM-

# Conclusion

Basic login functionality with a role-based dashboard was developed and tested by us. As a start, we reviewed our prototype for the ability to maintain the database structure for user authentication purposes. Subsequently, we added wireframes for the login page and provided the layout for the dashboard. The login system was developed and implemented, ensuring that users log in and access different content on the dashboard depending on their role. The system underwent several tests for incorrect inputs, session management, and password security. Finally, everything was documented with screenshots, database schema, and important code snippets. The project was submitted to GitHub with version control, a database dump, and a report. Overall, this activity opened more avenues for learning about system design, authentication, and debugging.