University of Messina

Bachelor of Data Analysis

A web portal for the University class timetable considering room availability and maximum capacity, different subjects and courses.

University timetable system – Project Report

Academic Year 2024/2025

Student ():

J.A.D. Serena (531401)

Table of Contents

Introduction

The **University Timetable System** is a web-based application designed to streamline course scheduling, room allocation, and timetable management for educational institutions. Built with PHP, MySQL, HTML5, and CSS3, it features:

- Role-based access (Admin/Student)
- Conflict-free room scheduling
- Real-time timetable visualization
- Responsive interface

designed to manage university courses, subjects, rooms, and timetables. It provides distinct functionalities for students and administrators, including user registration, login, session management, and specific dashboards for timetable viewing and administrative tasks. our system utilizes a database to store and retrieve all relevant information, ensuring data persistence and integrity.

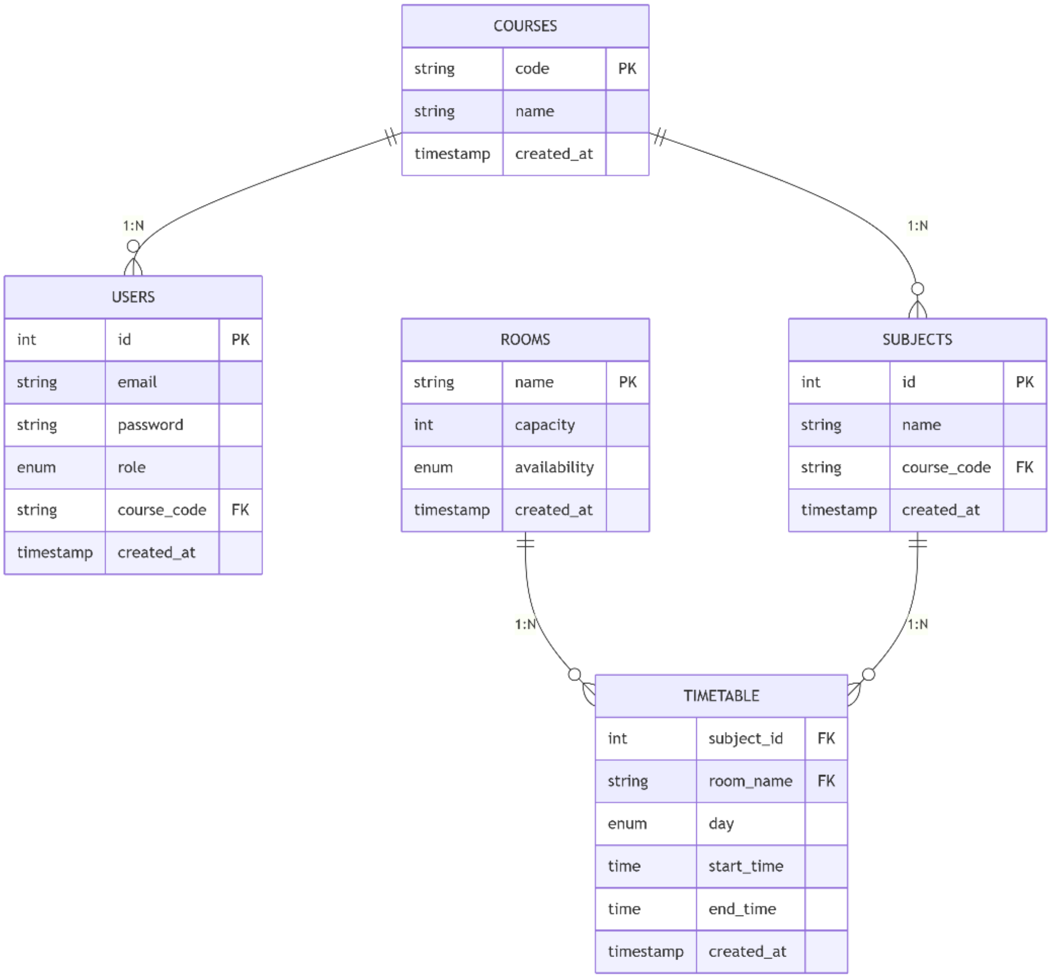
Key Technologies ive used are,

- Frontend: Modern CSS layout techniques
- Backend: Secure parameterized queries
- Database: Enforced data integrity through constraints

Database Design

The following are the tables created,

1. Courses, which has variables → code (PK), name, created_at
2. Users→id (PK), email, password, role, course_code, created_at
3. Subjects→id (PK), name, course_code (FK), created_at
4. Rooms→name (PK), capacity, availability, created_at
5. Timetable→subject_id (FK), room_name (FK), day, start_time, end_time



COURSES		
string	code	PK
string	name	
timestamp	created_at	

USERS		
int	id	PK
string	email	
string	password	
enum	role	
string	course_code	FK
timestamp	created_at	

ROOMS		
string	name	PK
int	capacity	
enum	availability	
timestamp	created_at	

SUBJECTS		
int	id	PK
string	name	
string	course_code	FK
timestamp	created_at	

TIMETABLE		
int	subject_id	FK
string	room_name	FK
enum	day	
time	start_time	
time	end_time	
timestamp	created_at	

Core Functionality

-Authentication-

Secure Login-login.php

Compares user-entered password with bcrypt-hashed version from database, On match this stores user role in session (admin/student) ,Sets session variables (user_id, email, course_code) and Redirects based on role like Admins → Admin Dashboard and Students → Personal Timetable

helps to Prevents unauthorized access **,**Enables role-specific interfaces and Secures sensitive operations with session validation

Session Management-auth.php

auth.php Acts as gateway script included in all protected pages. It Checks for valid session on every page load , Redirects unauthenticated users to login , Maintains session security by Automatic session start ,Session regeneration on login and Strict role validation

so makes a Impact on portal as it Creates secure access boundaries, Prevents URL manipulation to bypass login, Maintains user context across navigation.

_-Admin Features-

Conflict Detection-timetable.php

Here Use SQL interval overlap logic Which Checks if new time slot falls between existing bookings that Verifies `room_name` + `day` combination

1. Prevents happening overlaps and shows that if the `room_name` + `day` combinations are valid or not
   ```
   Existing: 10:00-12:00
   Invalid:  11:00-13:00 (overlaps)
   Valid:    13:00-15:00 (no overlap)
   ```
2. Returns error before database insertion

So this makes a big Impact on portal by Eliminating double-booking of rooms, Maintains timetable integrity and Prevents scheduling conflicts automatically

Room Management -(rooms.php)

Functionalities are Add Rooms: Admins can add new rooms with name, capacity, and availability status ,**Edit Rooms**: Modal-based interface for updating room details, **Delete Rooms**: Removes rooms and associated timetable entries, **Room Listing**: Displays all rooms in a sortable table. These functionalities will Central repository for physical resources, Ensures room availability for scheduling, Prevents double-booking conflicts and will do Capacity tracking for class size matching.

Student View-timetable.php

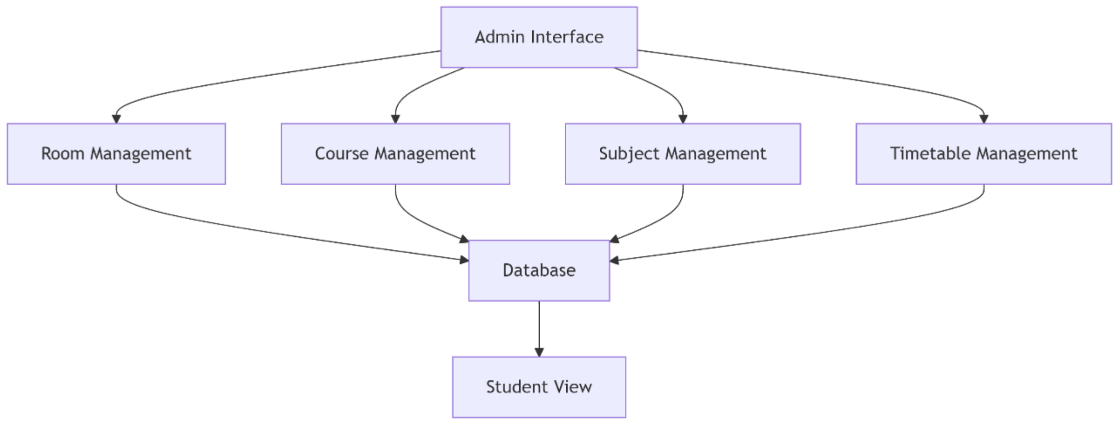
Here we Add Entries to Schedule classes with subject, room, day, and time, do Conflict Detection that Prevents overlapping room bookings, Delete Entries which is Remove scheduled classes and Displays all timetable entries. By doing these our web portal Creates conflict-free class schedules, Visual timetable representation, Ensures optimal resource utilization and Provides basis for student timetable views

Subject Management-subjects.php

Here we create course-specific subjects, Remove subjects and related timetable entries, Display all subjects with associated courses

By that we easily Defines academic content, Enables class scheduling, Links courses to specific classes ,Provides teaching material context

So basically, Admin Workflow will be, **Resource Setup** (Rooms, Courses, Subjects), **Timetable Creation**, **Conflict Validation**, and finally **Ongoing Management** (Edits/Deletions)

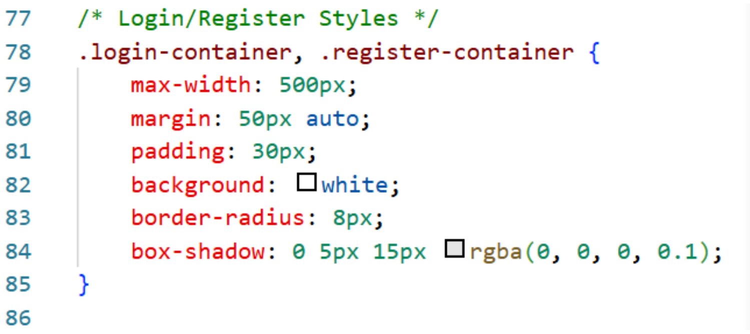


Key Interfaces and css features

Main user experience impacts made by CSS are Intuitive Navigation: Consistent header and card patterns, Clear Hierarchy as it gives Visual distinction between content types, Responsive Feedback which Interactive elements show state changes, Sufficient contrast and readable typography, Modal dialogs prevent page reloads during edits so makes efficiency and gives Visual Consistency by using unified color scheme and spacing

login/Register Interface

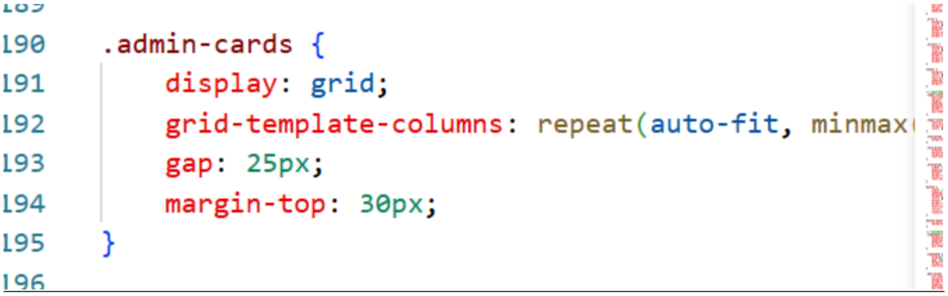
- Here we do, User authentication and account creation
- Key Components are,
  - Email/password fields
  - Course selection (registration)
  - Form validation



```
/* Login/Register Styles */
.login-container, .register-container {
    max-width: 500px;
    margin: 50px auto;
    padding: 30px;
    background: □white;
    border-radius: 8px;
    box-shadow: 0 5px 15px □rgba(0, 0, 0, 0.1);
}
```

Admin Dashboard

- Use as Central hub for administrative functions
- Key Components are
    - Management cards (Courses, Rooms, Subjects, Timetable)
    - Quick access to all admin features

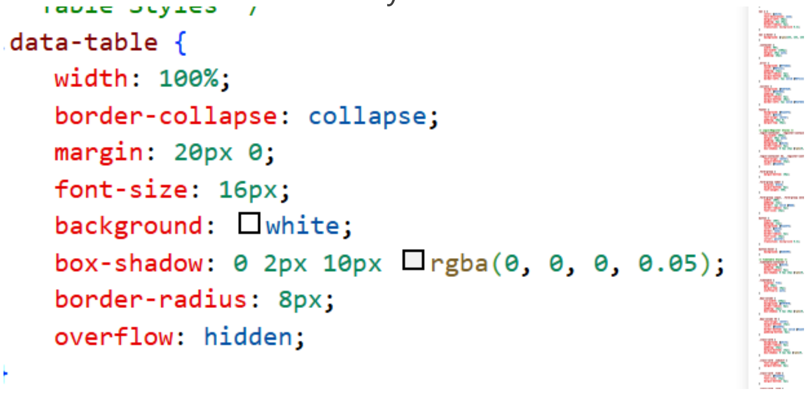


```
.admin-cards {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(
    gap: 25px;
    margin-top: 30px;
}
```

Data Management Tables

- Here we do CRUD operations for courses, rooms, subjects
- Components are
    - Sortable data tables
    - Action buttons (Edit/Delete)

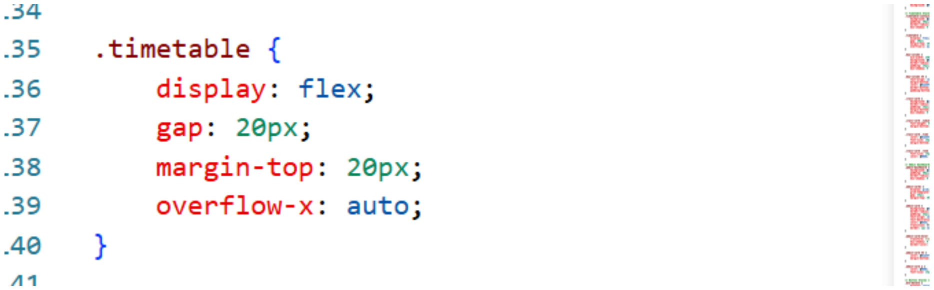
   - Add new entry forms



```
  Table Styles */
.data-table {
    width: 100%;
    border-collapse: collapse;
    margin: 20px 0;
    font-size: 16px;
    background: □white;
    box-shadow: 0 2px 10px □rgba(0, 0, 0, 0.05);
    border-radius: 8px;
    overflow: hidden;
}
```

Timetable View

- We Display scheduled classes
- Components
   - Day-based columns
   - Class cards with subject/time/room details
   - Responsive layout



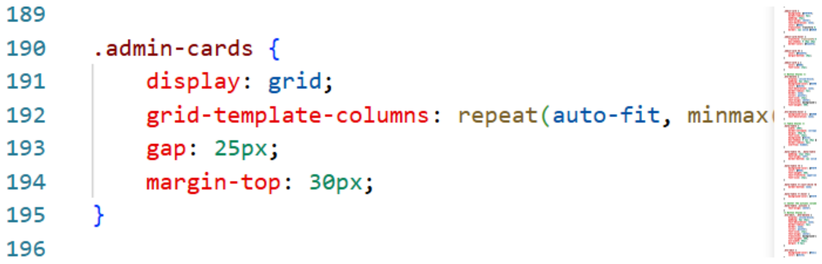
```

.timetable {
    display: flex;
    gap: 20px;
    margin-top: 20px;
    overflow-x: auto;
}
```

Modal Dialogs

- We use these to edit forms without page reload. Main components are,
   - Edit forms
   - Close functionality
   - Overlay background

Key CSS Features and Their Impact

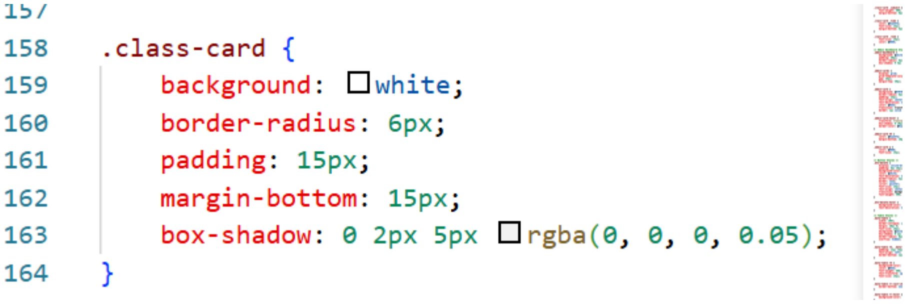
1. **Flexbox & Grid Layouts**



```css

.admin-cards {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(
    gap: 25px;
    margin-top: 30px;
}

```

- Used for Responsive card layouts. It will automatically adapts to screen size, maintains consistent spacing

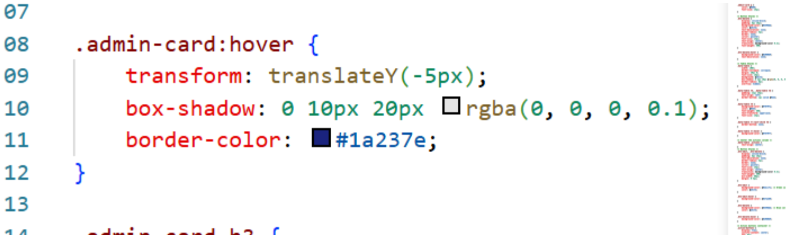
2.Box Model Enhancements



```css

.class-card {
    background: □white;
    border-radius: 6px;
    padding: 15px;
    margin-bottom: 15px;
    box-shadow: 0 2px 5px □rgba(0, 0, 0, 0.05);
}
```

- Used for Visual depth and containment so, it Creates card-based UI with clear boundaries and hierarchy

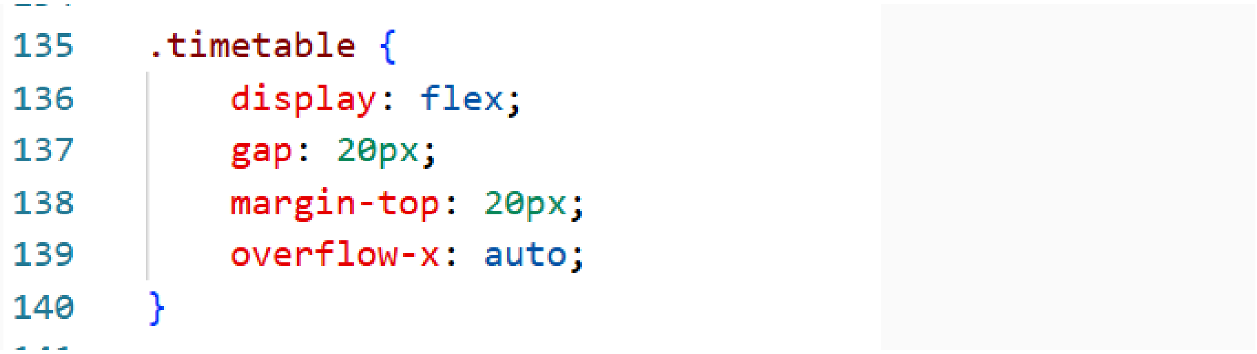
3. **Interactive States**



```css

.admin-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
    border-color: #1a237e;
}

.admin-card h3 {
```

Used to show User feedback on interaction as it will Provides visual cues for clickable elements.

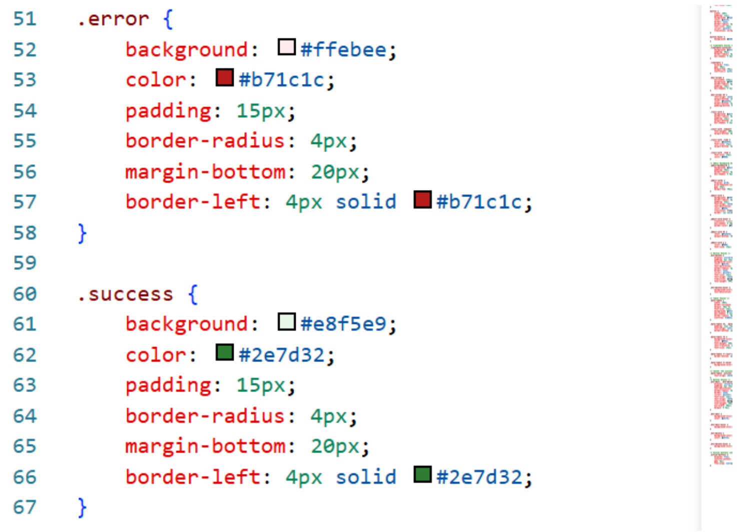
4. **Responsive Design**



```css
.timetable {
    display: flex;
    gap: 20px;
    margin-top: 20px;
    overflow-x: auto;
}
```

- for Device-agnostic experience as it Works on mobile/tablet/desktop without separate layouts.

5.Visual Feedback



```
.error {
    background: #ffebee;
    color: #b71c1c;
    padding: 15px;
    border-radius: 4px;
    margin-bottom: 20px;
    border-left: 4px solid #b71c1c;
}

.success {
    background: #e8f5e9;
    color: #2e7d32;
    padding: 15px;
    border-radius: 4px;
    margin-bottom: 20px;
    border-left: 4px solid #2e7d32;
}
```

- we use it for System status communication it will give Immediate visual confirmation of user actions.

so, in this way the CSS implementation creates a professional, accessible interface that supports both administrative efficiency (tables, forms) and student usability (clear timetable view), while maintaining responsive behavior across all device types.

Setup Guide

01)XAMPP Installation

1. Place uni_portal in htdocs/
2. Start Apache/MySQL
3. Access http://localhost/uni_portal

02)DB Initialization (config.php):

- Automatically creates tables with foreign keys , by,
    - in browser:
      `http://localhost/uni_portal/config.php`

03)Admin Account Setup

1. Register First Account
    - Access: `http://localhost/uni_portal/register.php`
    - Fill: Email, Password, Course Selection
2. Promote to Admin

   In phpMyAdmin or MySQL console:

   ```
   UPDATE users SET role = 'admin' WHERE email = 'your@email.com';
   ```

04) Populate Initial Data

1. Add Courses (Courses Management),

   ```
   http://localhost/uni_portal/admin/courses.php
   ```

   For Example add,

    - Code: `CS101`
    - Name: `Computer Science`
2. Add Rooms (Room Management),

   ```
   http://localhost/uni_portal/admin/rooms.php
   ```

   for Example:

    - Name: `A101`
    - Capacity: `30`
    - Availability: `Available`
3. Add Subjects (Subject Management),

   ```
   http://localhost/uni_portal/admin/subjects.php
   ```

    - Example:

        - Name: `Web Programming`
        - Course: `Computer Science`

05)Schedule Classes

1. Now Create Timetable Entries,
   `http://localhost/uni_portal/admin/timetable.php`
    - Select: Subject, Room, Day, Start/End Time
    - Conflict Detection: Automatically prevents overlapping bookings

06)Student Registration

1. Then Create Student Accounts,
   `http://localhost/uni_portal/register.php`
    - Assigning existing courses and Students automatically inherit course timetable

07) Access Views

1. Now focus on Admin Dashboard,
   `http://localhost/uni_portal/admin.php`

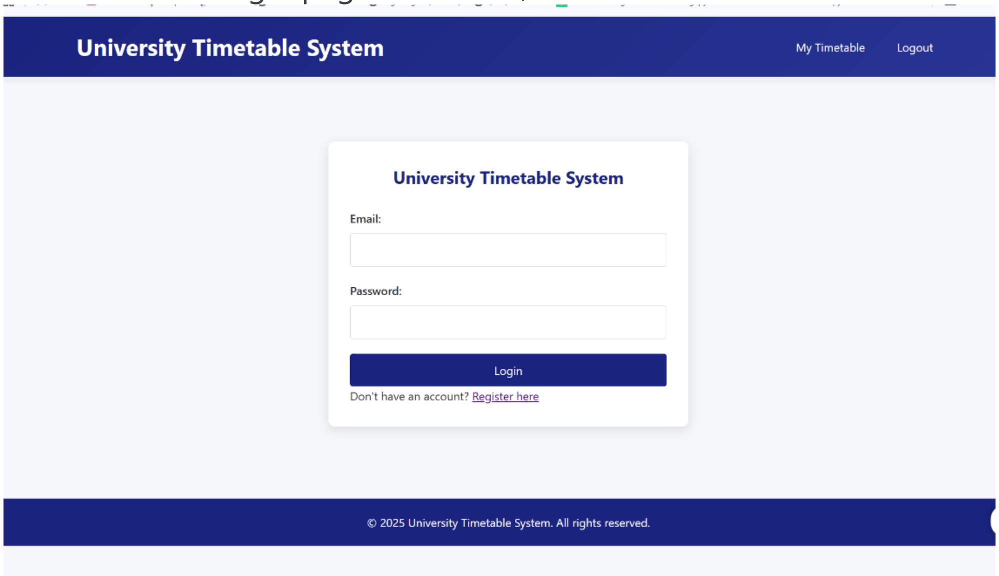
   which is our Central management hub
2. Student Timetable:
   `http://localhost/uni_portal/timetable.php`

   gives us Course-specific view

Final System

1. **Admin Functions** are Add/delete courses, Manage room availability, Schedule conflict-free classes

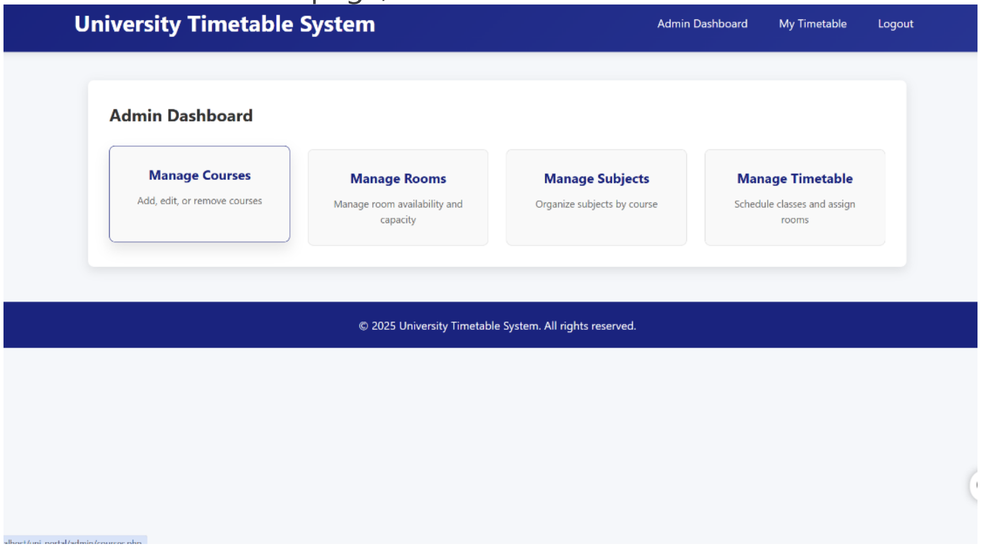
This is how the login page looks like ,



Don't have an account? Register here

© 2025 University Timetable System. All rights reserved.

And this is the admin page,


University Timetable System
Admin Dashboard
My Timetable
Logout
Admin Dashboard
Manage Courses
Add, edit, or remove courses
Manage Rooms
Manage room availability and capacity
Manage Subjects
Organize subjects by course
Manage Timetable
Schedule classes and assign rooms

© 2025 University Timetable System. All rights reserved.

2. **Student Experience** will be Register new account and Viewing personalized timetable
3. **Security Checks** are we Cannot access admin pages without login ,SQL injection attempts blocked and Password hashing verified
4. **Data Integrity** is guaranteed by Deleting course cascades to subjects, Room deletion removes timetable entries and by Foreign key constraints enforced.

Conclusion

Our **University Timetable System** successfully addresses the core challenges of academic scheduling through a robust, secure, and user-friendly web application. By implementing role-based access control, conflict-free scheduling algorithms, and responsive design principles. Conflict Detection by SQL interval overlap algorithm to Zero double-bookings. role System by Session-based admin/student separation to Secure data partitioning. Responsive Timetable as we used CSS Flexbox with dynamic day columns to 100% mobile compatibility. DB Architecture designed using 5 normalized tables with cascading relations.