UE22CS352B - Object Oriented Analysis & Design

Mini Project Report

College Fest Management System

Submitted by:

Praneetha Praveen Kalbhavi : PES1UG22CS438
Prapti Arali : PES1UG22CS439
Prayashi Verma : PES1UG22CS446
Priyanka Kumari : PES1UG22CS454

Semester: 6 Section: H

Meghana G

January - May 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

Problem Statement:

College fests require coordination among various roles, including organizers, volunteers, participants, and administrators. Traditional management methods often lead to communication gaps, scheduling conflicts, inefficient task handling, and a lack of centralized information. These issues result in poorly organized events and a diminished experience for everyone involved.

To overcome these challenges, a centralized, role-based digital system is needed. This system should streamline event creation, registration, attendance tracking, notification, and reporting while improving coordination, transparency, and overall fest management.

Key Features:

1. **Event Creation & Approval**
    - **Actors**: Organizer, Admin
    - **Functionality**: Organizers can create and manage events. Admins have authority to approve or reject these events. Rejected events can be modified and resubmitted.
    - **Implementation**:
        i. Event model and OrganizerController, AdminController handle business logic.
        ii. Status tracking for events (Pending/Approved/Rejected).

2. **Event Registration & Participation**
    - **Actors:** Participant
    - **Functionality:** Participants can view all approved events, register for events, and track their registrations via a dashboard.
    - **Implementation:**
        i. Participant, Event, and Participant Attendance models.
        ii. ParticipantAuthController and MainController manage registration workflows.
        iii. Each user has access to a dashboard for interaction and event tracking.
3. **Volunteer Task Assignment & Management**
    - **Actors**: Organizer, Volunteer
    - **Functionality**: Organizers create tasks for volunteers. Volunteers can view tasks through their dashboard.
    - **Implementation**:
        i. VolunteerController and OrganizerController for task operations.
        ii. Decorator pattern (VolunteerDecorator, LoggingVolunteerDecorator) is used to extend task-related functionalities.

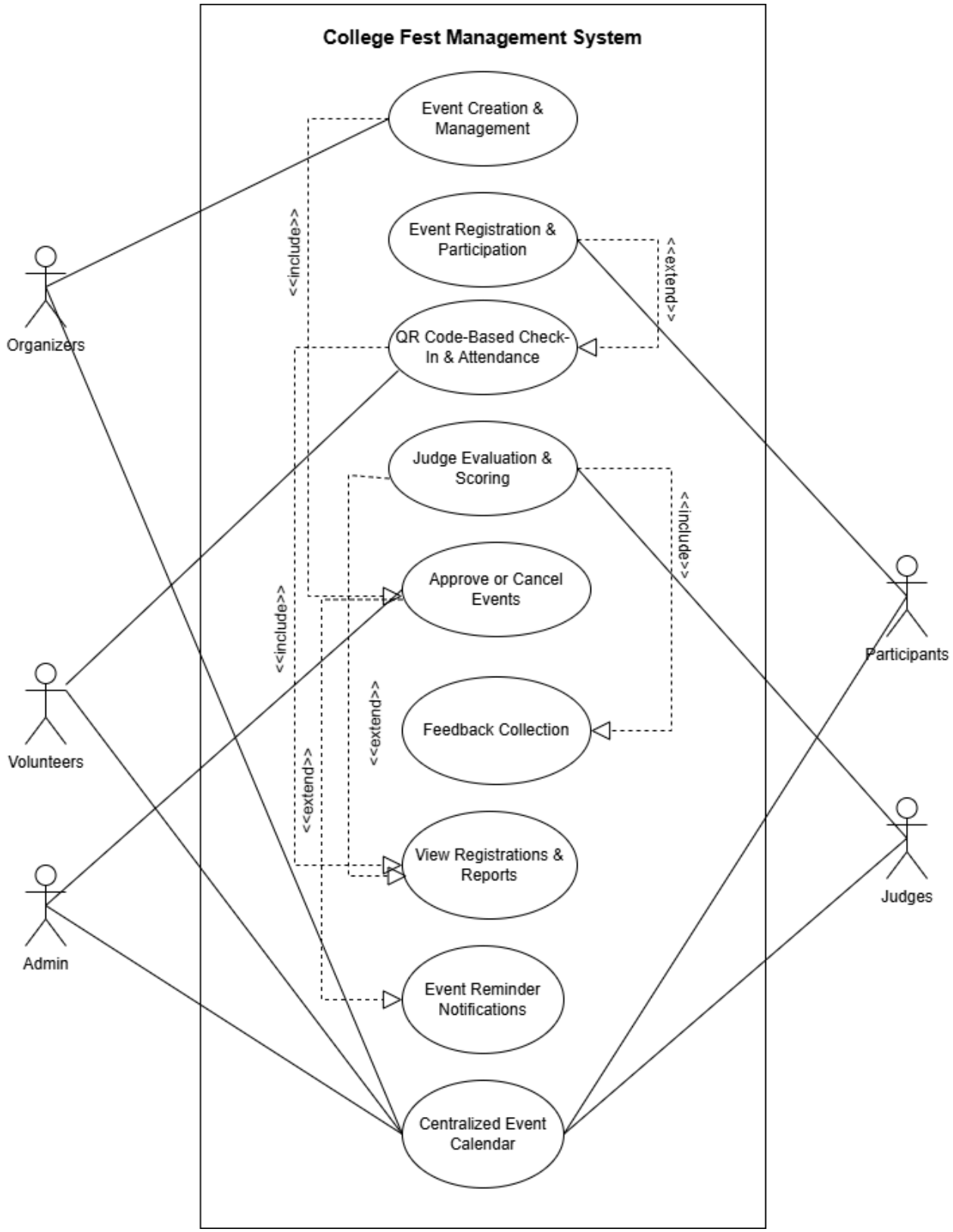
4. **Attendance Tracking System**
    - **Actors:** Volunteer, Participant, System
    - **Functionality:** Volunteers mark participant attendance during events.
    - **Implementation:**
        i. Attendance and ParticipantAttendance models manage records.
        ii. AttendanceController handles attendance updates and fetches.

5. **Notification System**
    - **Actors:** Admin, Organizer
    - **Functionality: The** Organizer receives notifications for event approvals.
    - **Implementation:**
        i. Observer Pattern: Ensures real-time update propagation to users.
        ii. Factory Pattern: Used to create notification types like email and system pop-ups.

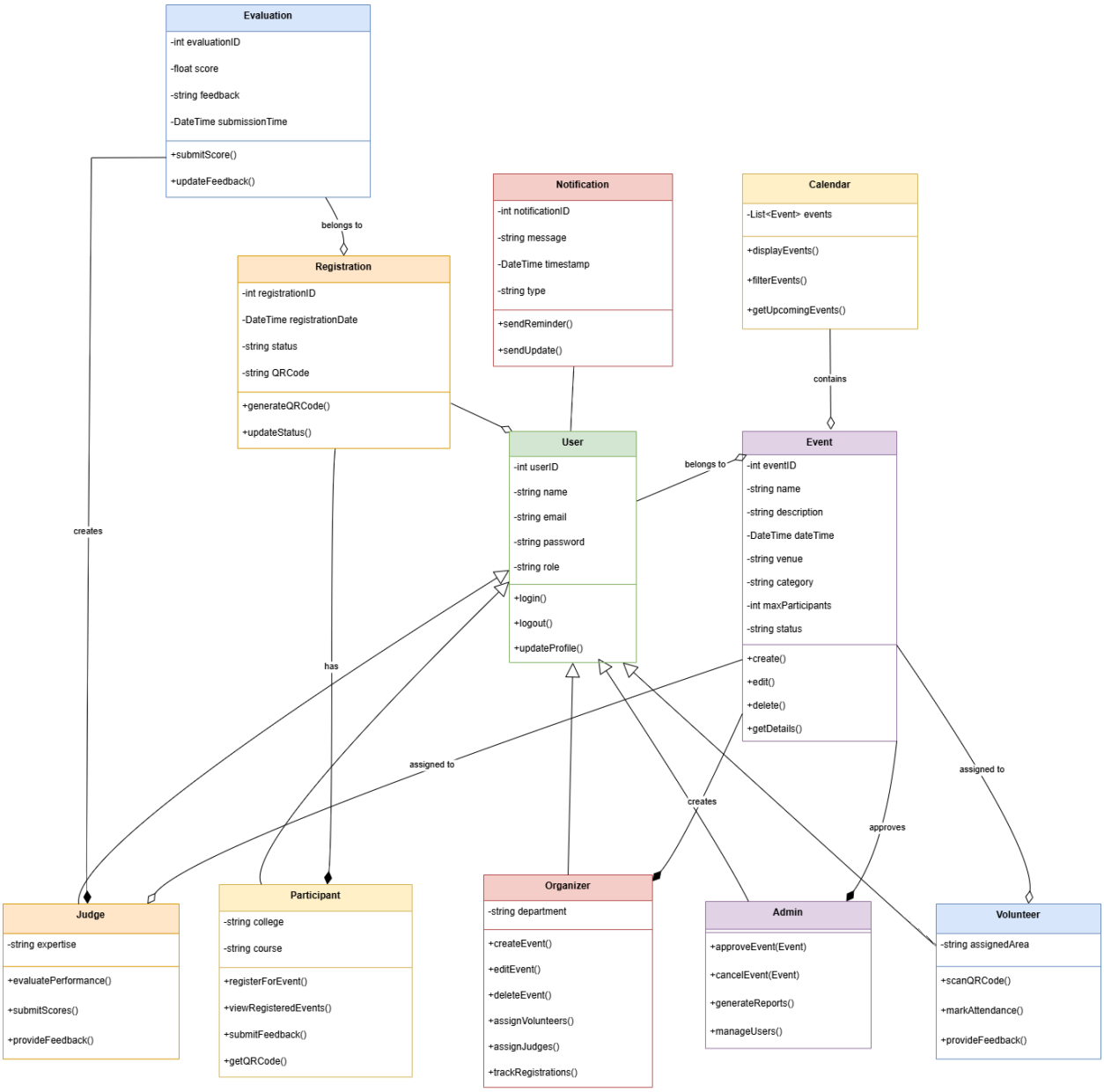
6. **Authentication & Role-Based Authorization**
    - **Actors:** Admin, Organizer, Volunteer, Participant
    - **Functionality:** All users authenticate through a role-based login system. The system dynamically selects an authentication strategy depending on the user role or configuration.
    - **Implementation:**
        i. Implements the Strategy Pattern with classes like AuthenticationStrategy, BasicAuthenticationStrategy, and PasswordAuthenticationStrategy.
        ii. Used in VolunteerController and other role-specific controllers to manage login logic.
7. **Analytics**
    - **Actors:** Admin, Organizer
    - **Functionality:** Reports are generated for event statuses and events per organizer.
    - **Implementation:**
        i. Admin can view statistics such as event approvals, user activity, and attendance.
        ii. Uses data from models like Event, ParticipantAttendance, and Task.
8. **Volunteer Attendance Tracking**
    - **Actors:** Volunteer, System
    - **Functionality:** Volunteers check in and out during their event shifts. Their attendance is recorded along with timestamps to track hours worked.
    - **Implementation:**
        i. AttendanceRepository logs check-in/check-out data.
        ii. AttendanceController manages attendance logic and updates status (PRESENT/ABSENT)**.**
9. **Event Calendar View**
    - **Actors:** Admin
    - **Functionality:** Admin can view a centralized calendar of all approved events to oversee scheduling and avoid conflicts.
    - **Implementation:**
        i. AdminController fetches events by date.
        ii. Calendar displayed on the admin dashboard includes event status and details.

Models:

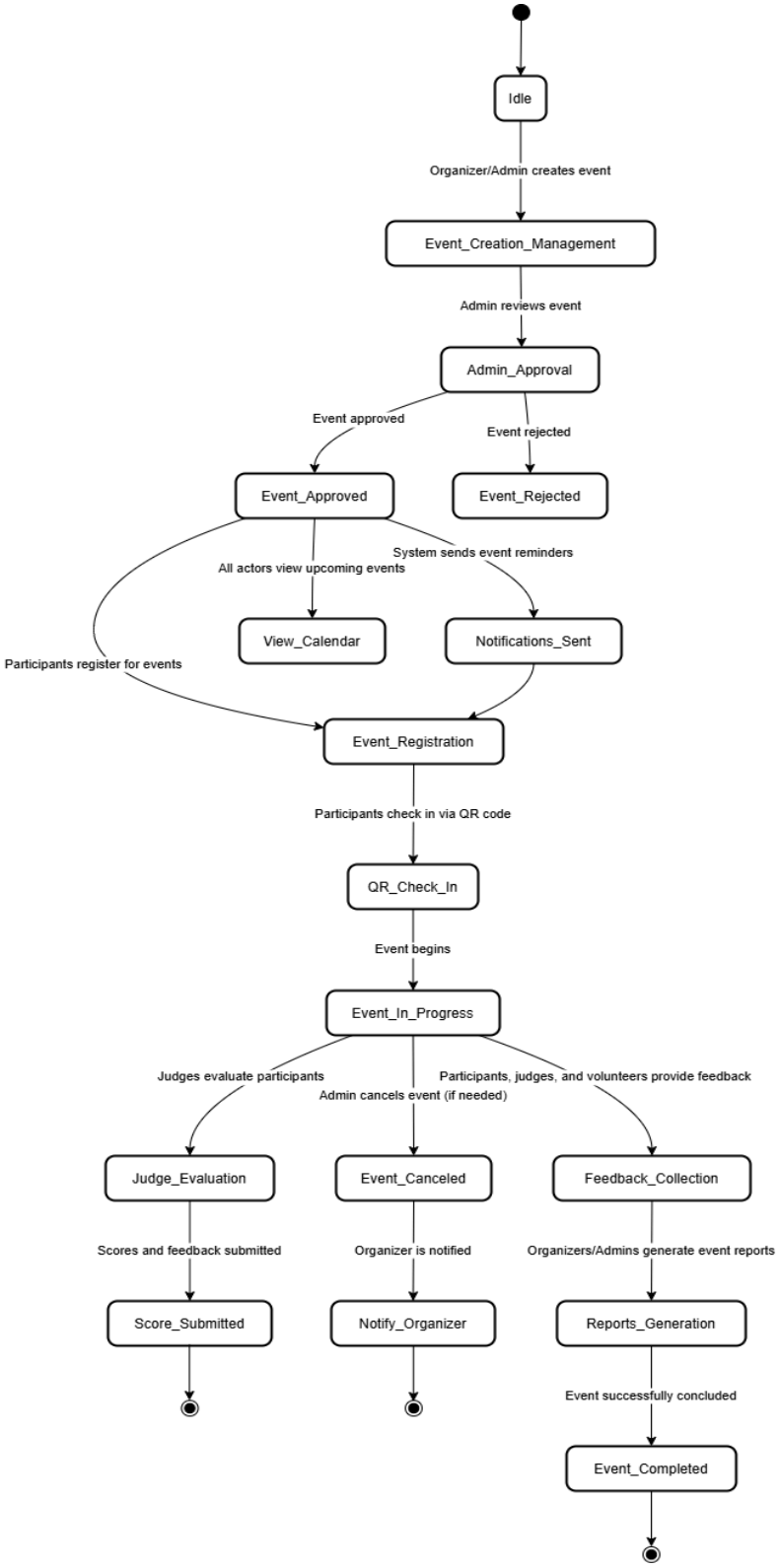
1. Use Case Diagram:



2. Class Diagram:

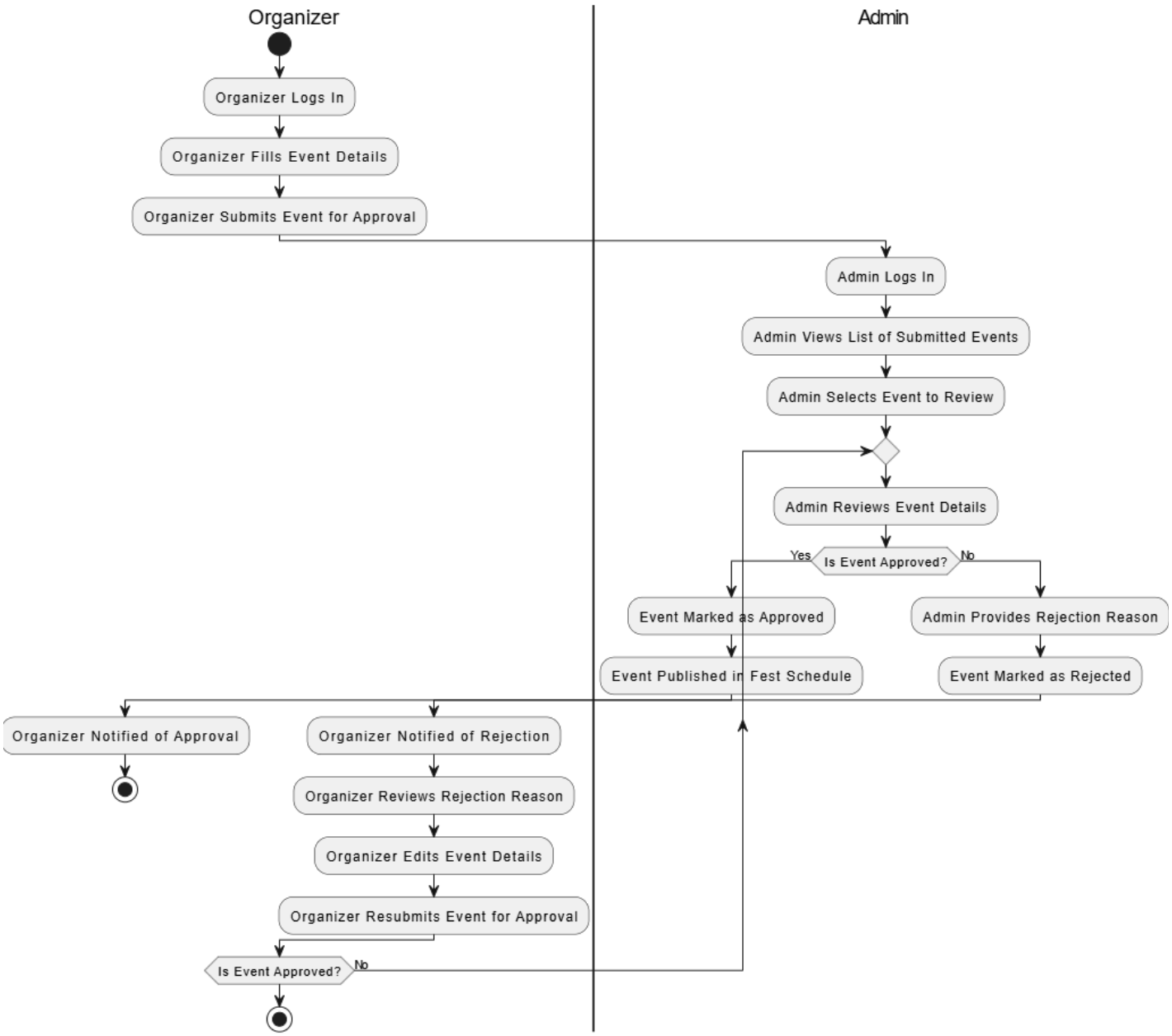


3. State Diagram:

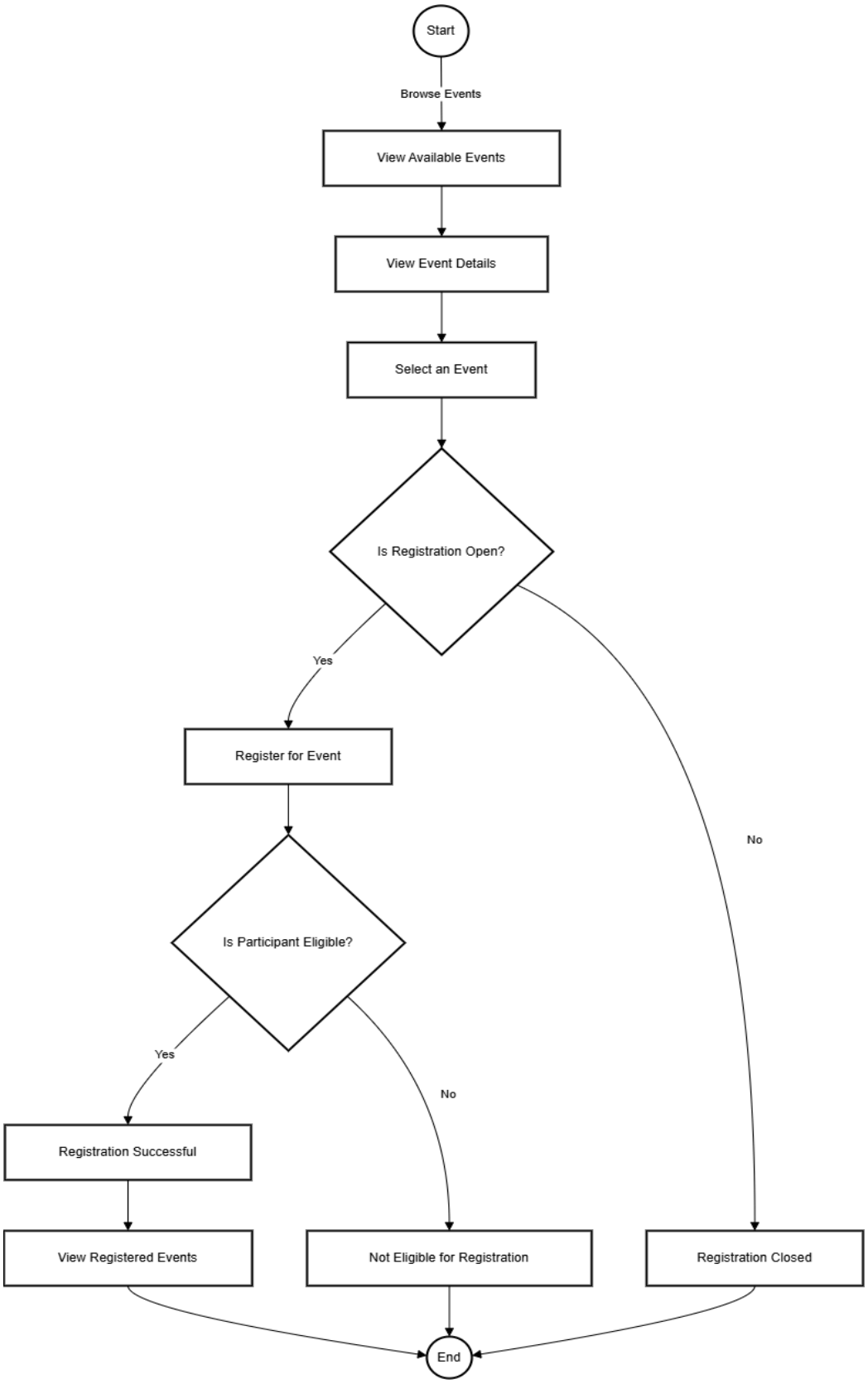


Activity Diagrams:

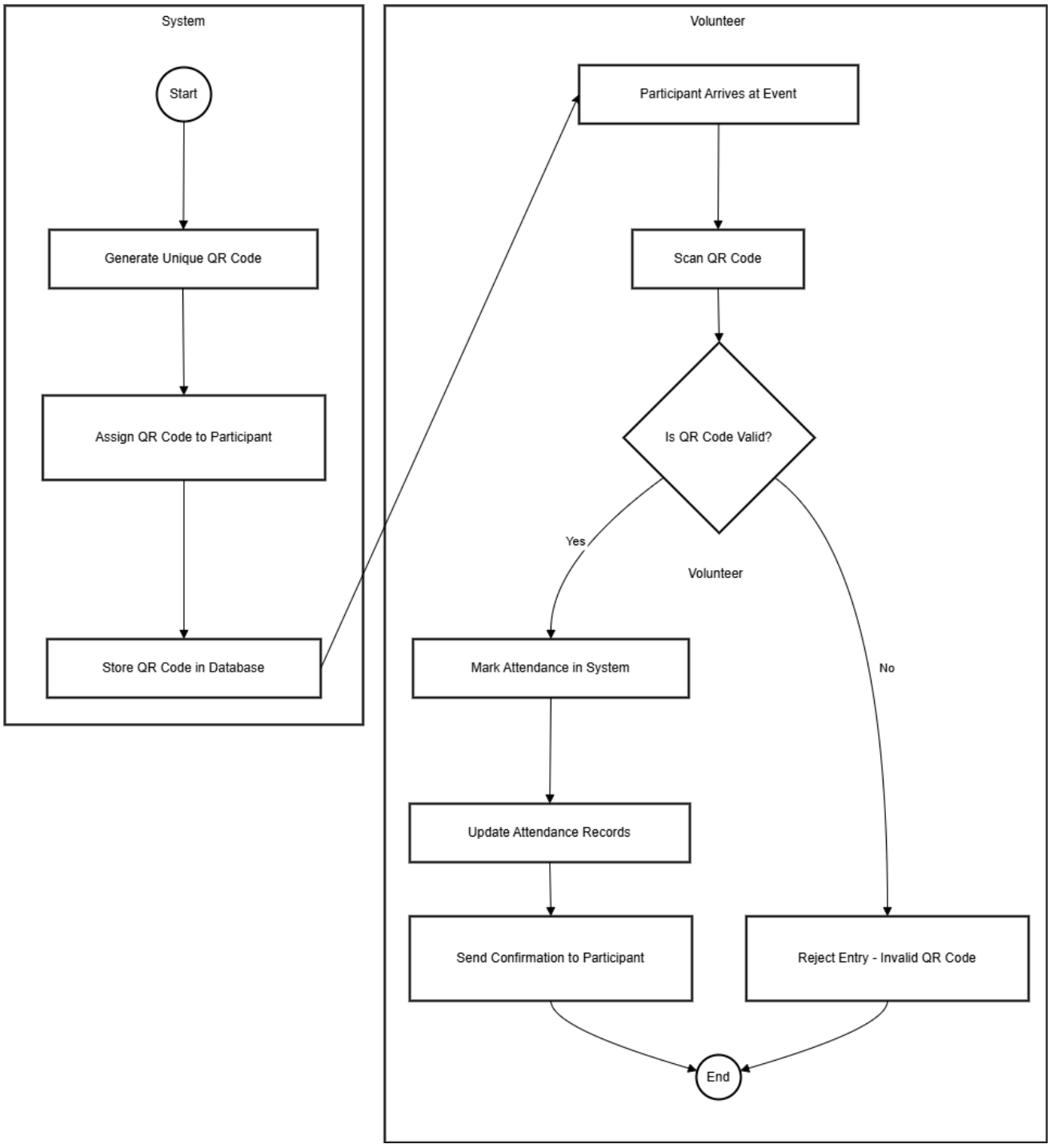
1. Event Approval Workflow (Admin, Organizer)



2. **Event Registration & Participation (Participants)**



3. Check-In & Attendance (Volunteers, System)



Architecture Patterns

The system is designed using the MVC (Model–View–Controller) architecture to separate concerns, improve maintainability, and enable scalable development.

Implementation in the Project:

- **Model**: Handles core business logic and data structures. Examples: Event, Participant, Volunteer, Task, Attendance.
- **View**: User-facing components that display information and allow interaction. Example: JSP pages / HTML interfaces for login, dashboards, and registration.
- **Controller**: Bridges the model and view by handling user input and updating the model or view accordingly. Examples: OrganizerController, AdminController, VolunteerController, ParticipantAuthController.

- **Design Patterns**
  - **Creational Patterns:**
    - **Factory Method:** Decouples object creation from usage.
      - NotificationFactory.createNotification() instantiates EmailNotification or DashboardNotification.
  - **Structural Patterns:**
    - **Decorator:** Extends functionality dynamically (Open/Closed compliance).
      - LoggingVolunteerDecorator adds logging to Volunteer without modifying the base class.
  - **Behavioral Patterns:**
    - **Observer:** Decouples event processing from notifications.
      - NotificationObserver notifies users when event status changes (e.g., APPROVED → REJECTED).
    - **Strategy:** Encapsulates interchangeable algorithms.
      - VolunteerController switches auth methods (e.g., Basic vs. Password) via AuthenticationStrategy.

Design Principals

1. Single Responsibility Principle (SRP)

- Each controller and class is assigned a single, well-defined responsibility (e.g., OrganizerController for event management, AttendanceController for attendance tracking).
- Domain models such as Event, Task, and Volunteer represent distinct entities, encapsulating only relevant data and behavior.

2. Open/Closed Principle (OCP)

- The authentication system is built using the Strategy Pattern, allowing new authentication methods to be added without altering existing logic.
- The notification system employs the Factory Pattern, enabling new notification types to be integrated without modifying the dispatch logic.

3. Liskov Substitution Principle (LSP)

- Strategy interfaces (e.g., AuthenticationStrategy) allow for interchangeable use of different

authentication methods without breaking system behavior.
- Substitutability is respected, ensuring that derived classes uphold the expectations set by their base interfaces.

4. Interface Segregation Principle (ISP)

- Classes are only exposed to relevant methods, avoiding unnecessary dependencies and ensuring clean, focused interfaces.
- Modularization in authentication and notification systems reflects this principle, preventing bloated interfaces.

5. Don’t Repeat Yourself (DRY)

- Common logic for authentication, notification handling, and task logging is abstracted and reused across the application.
- The system avoids redundant code in controllers and services, ensuring consistency and reducing maintenance overhead.

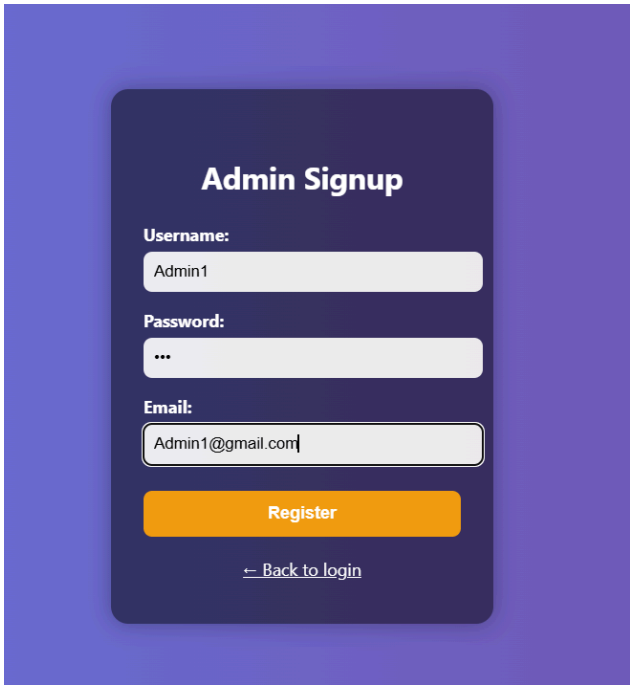
Github link to the Codebase:

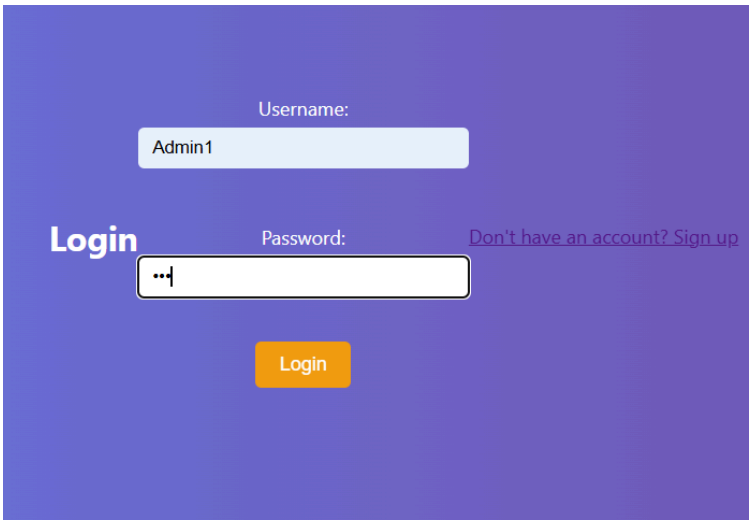
https://github.com/priyanka-kumari26/College_Fest_Management_System

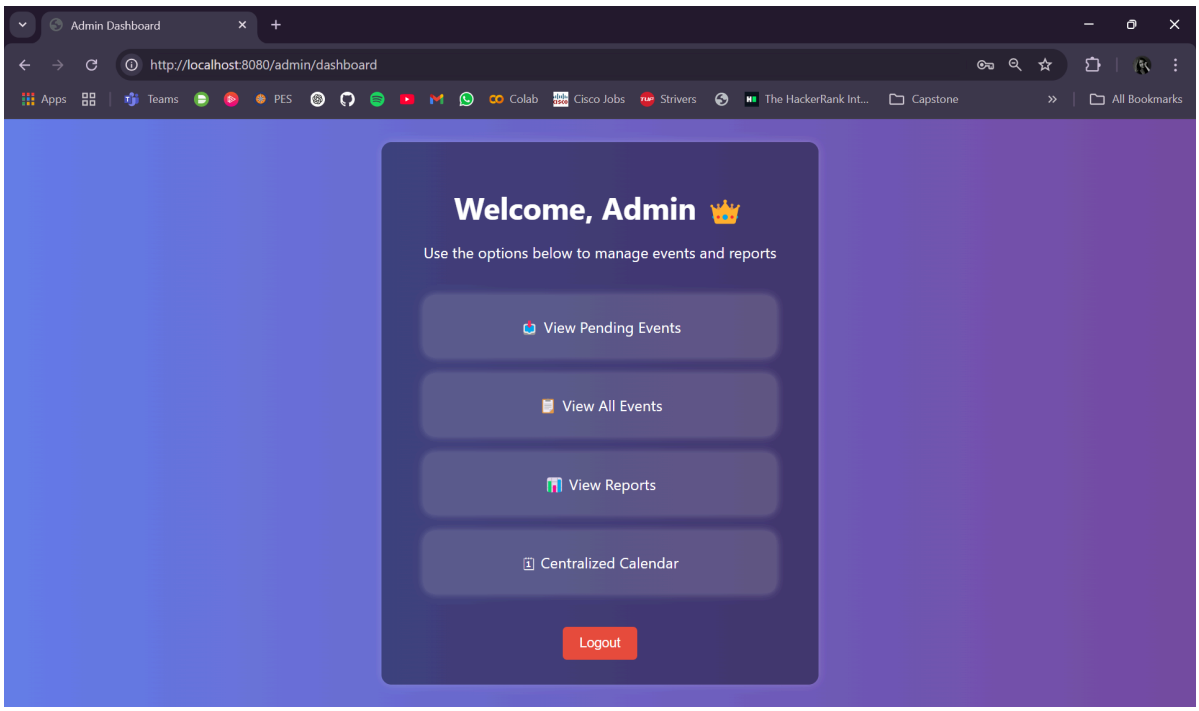
Screenshots

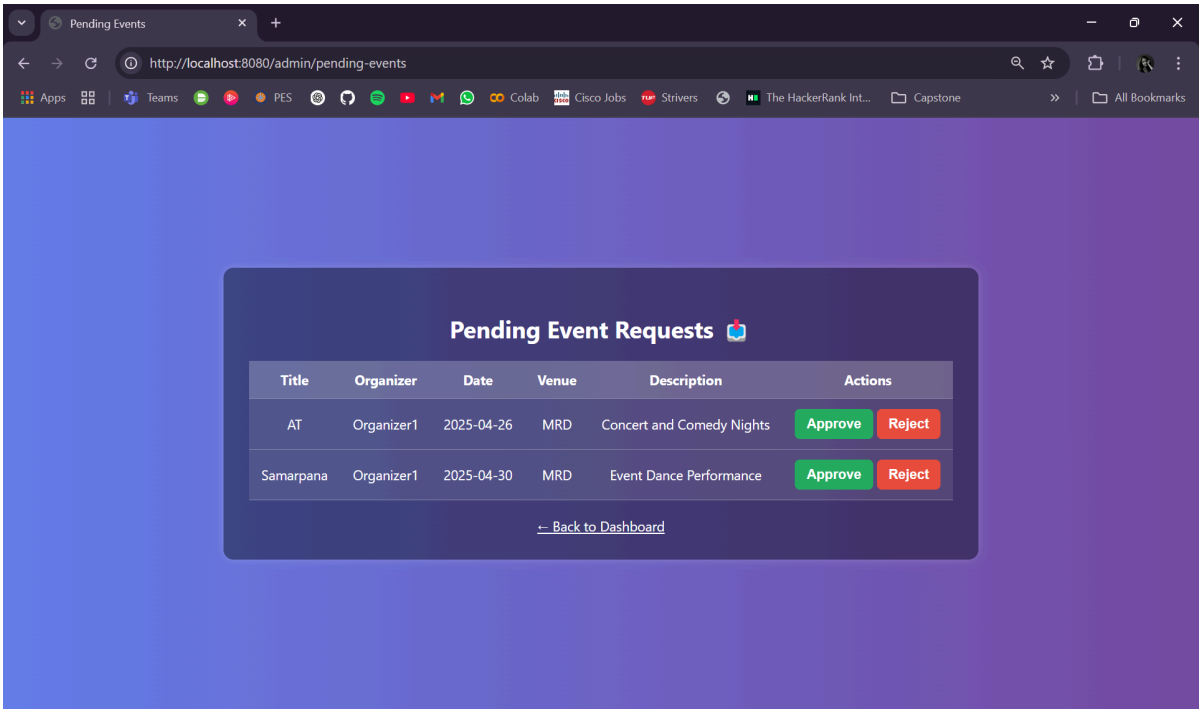
UI:

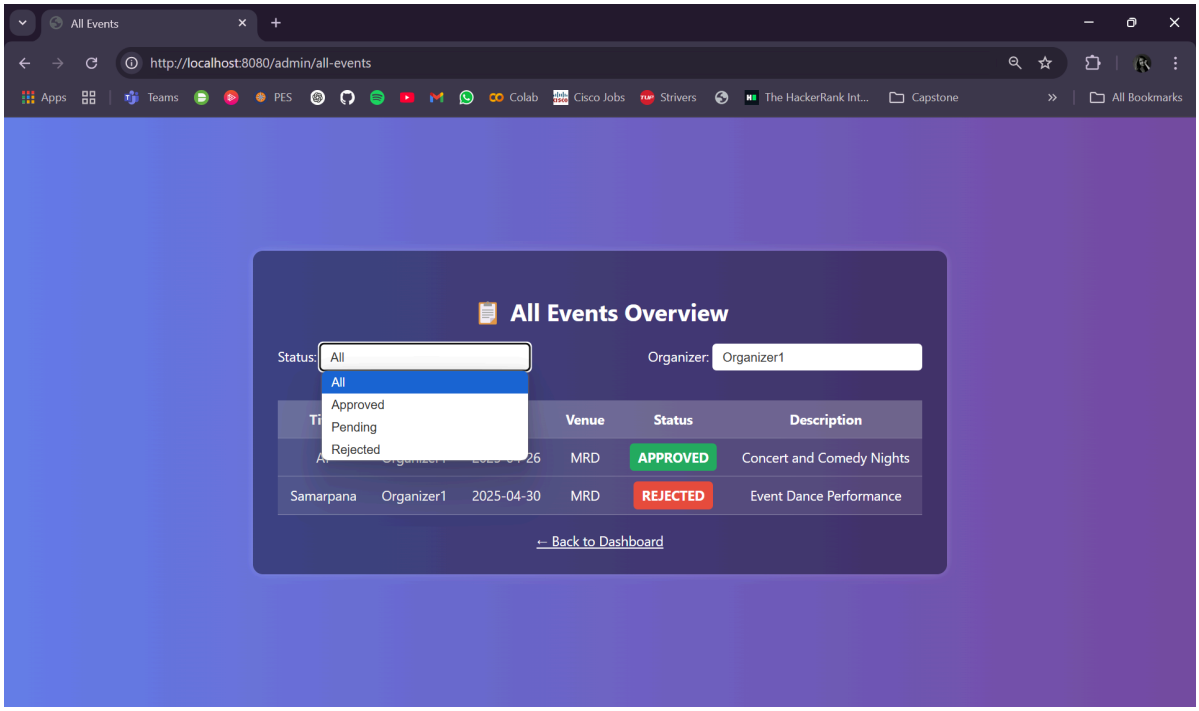
1. **Admin**

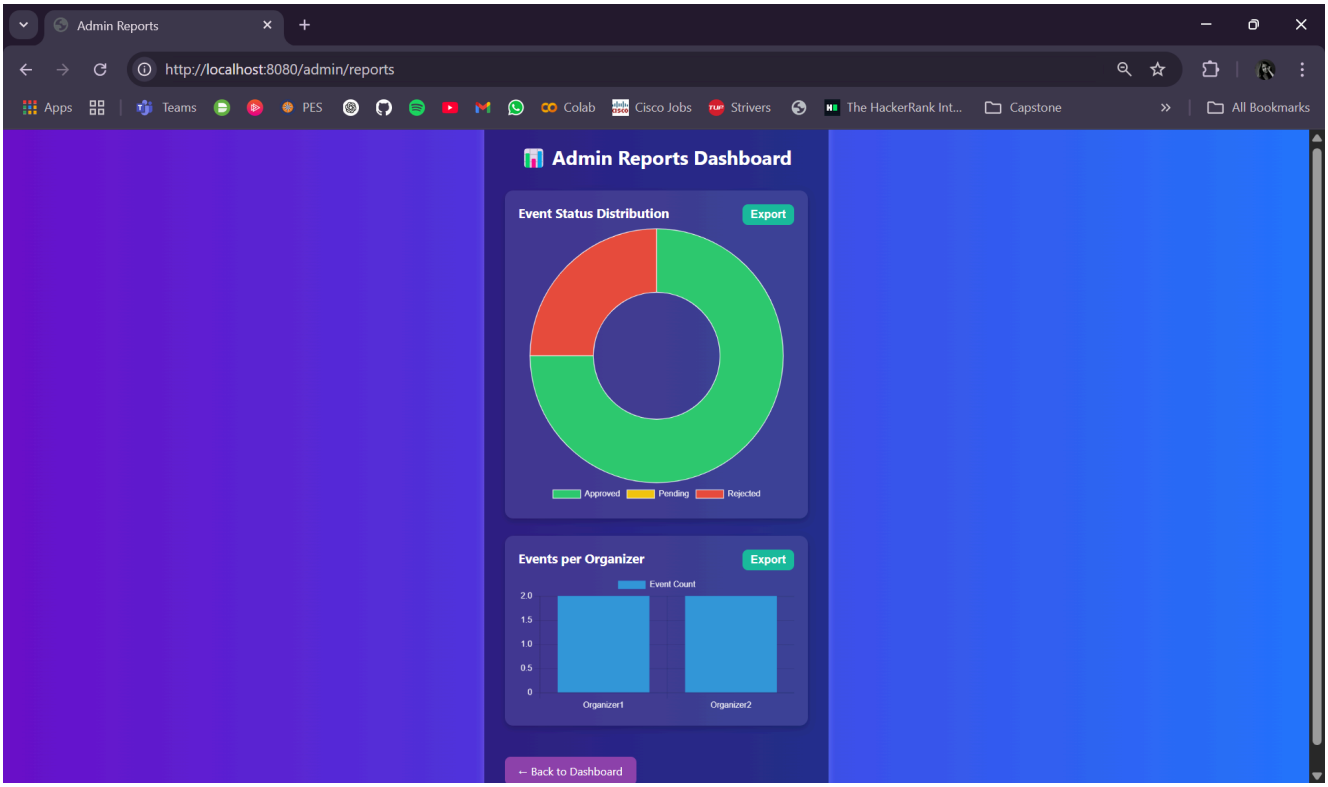


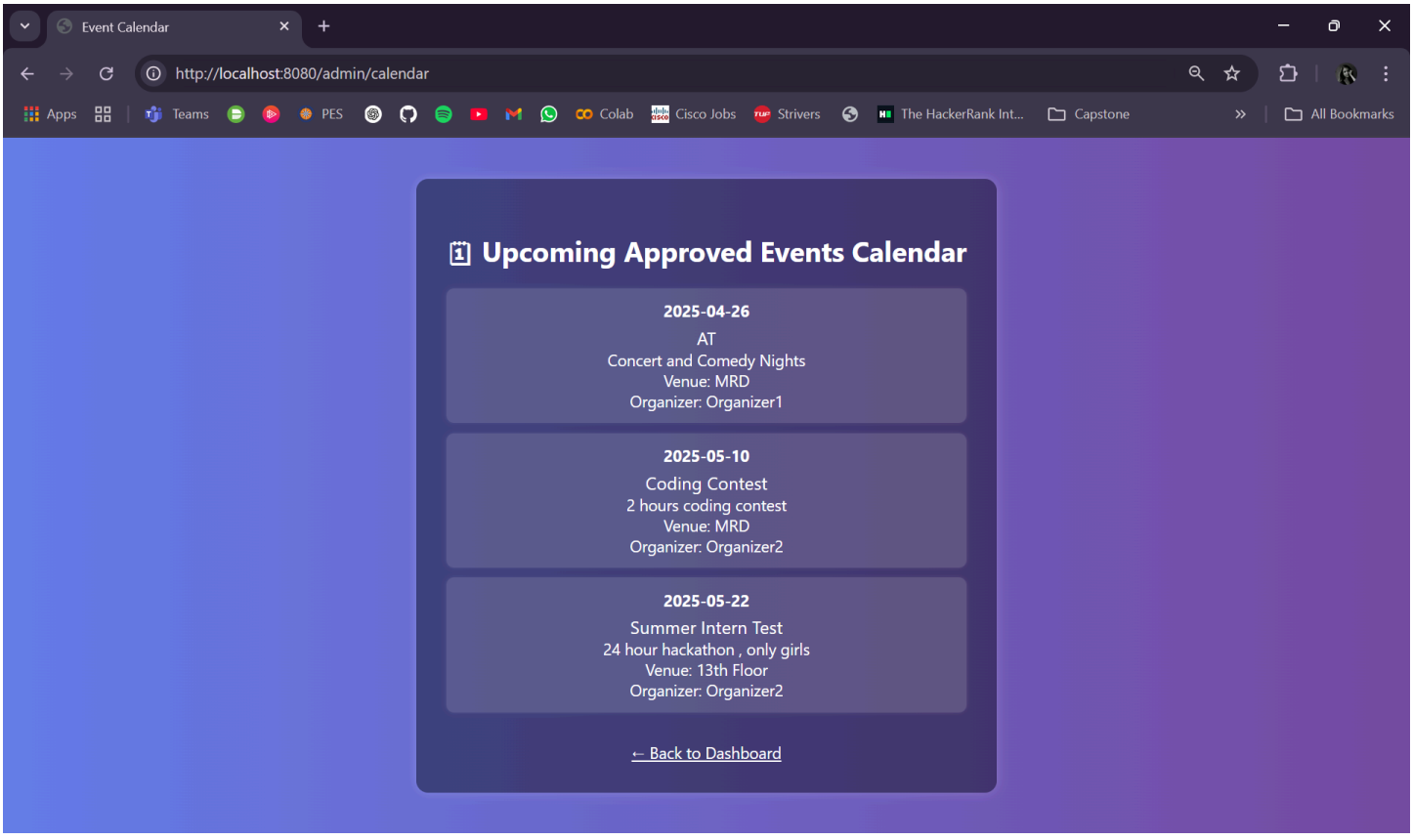




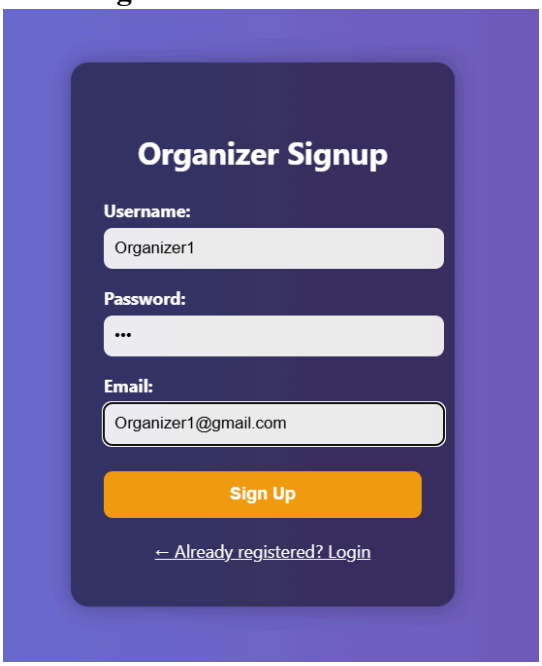

Pending Events
http://localhost:8080/admin/pending-events
Apps
Teams
PES
Colab
Cisco Jobs
Strivers
The HackerRank Int...
Capstone
All Bookmarks
Pending Event Requests
Title
Organizer
Date
Venue
Description
Actions
AT
Organizer1
2025-04-26
MRD
Concert and Comedy Nights
Approve
Reject
Samarpana
Organizer1
2025-04-30
MRD
Event Dance Performance
Approve
Reject
← Back to Dashboard

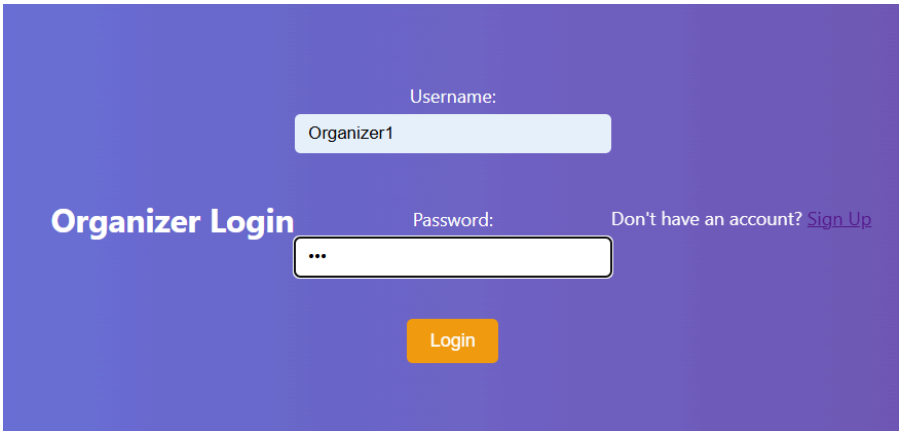

All Events
http://localhost:8080/admin/all-events
Apps
Teams
PES
Colab
Cisco Jobs
Strivers
The HackerRank Int...
Capstone
All Bookmarks
All Events Overview
Status:
All
Approved
Pending
Rejected
Organizer:
Organizer1
Venue
Status
Description
MRD
APPROVED
Concert and Comedy Nights
Samarpana
Organizer1
2025-04-30
MRD
REJECTED
Event Dance Performance
← Back to Dashboard

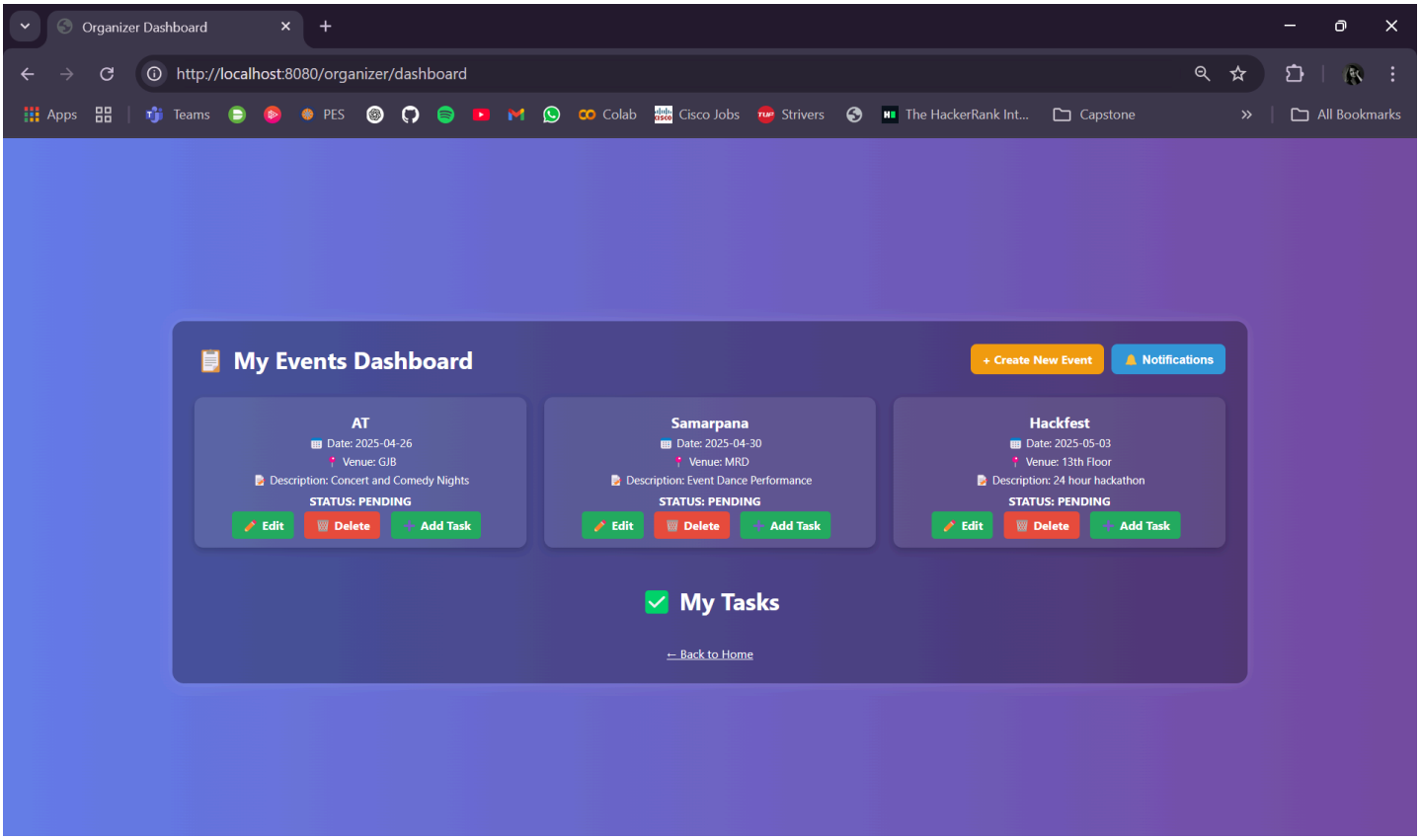

Admin Reports
http://localhost:8080/admin/reports
Apps
Teams
PES
Colab
Cisco Jobs
Strivers
The HackerRank Int...
Capstone
All Bookmarks
Admin Reports Dashboard
Event Status Distribution
Export
Approved
Pending
Rejected
Events per Organizer
Export
Event Count
2.0
1.5
1.0
0.5
0
Organizer1
Organizer2
← Back to Dashboard

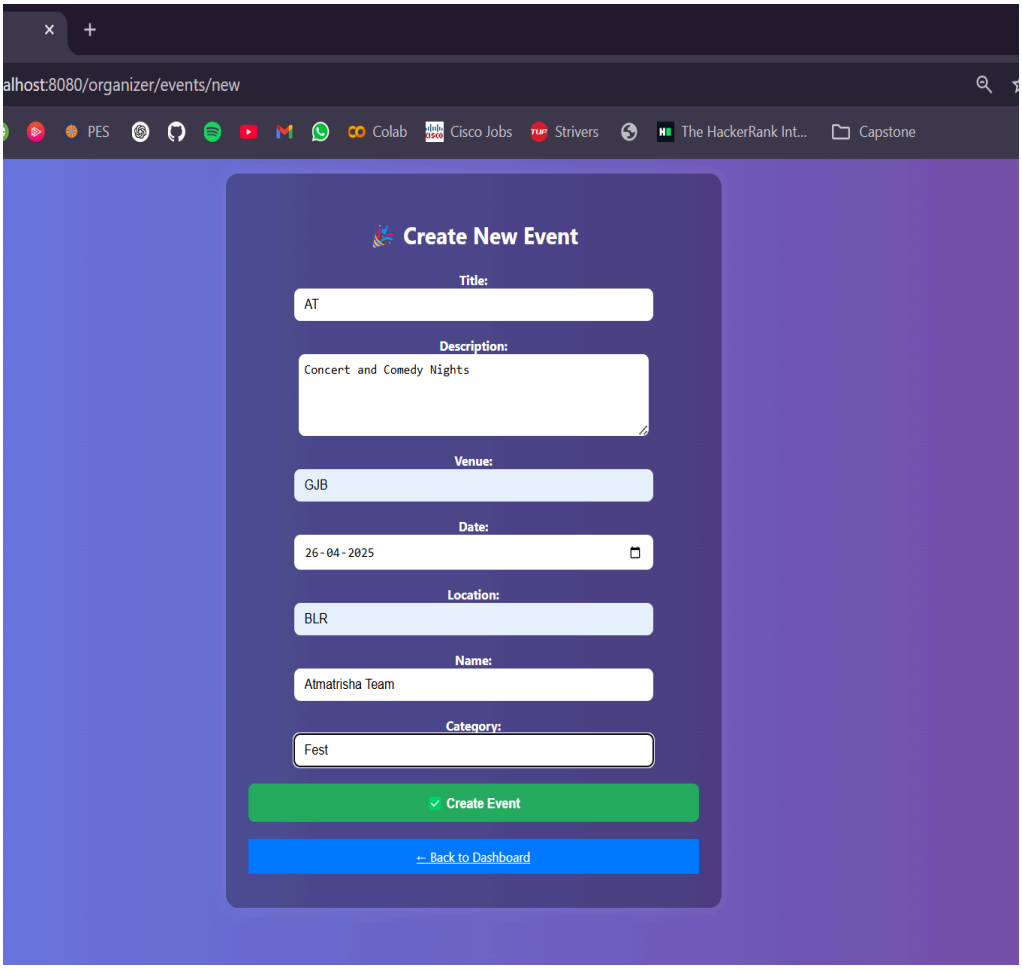

Event Calendar
http://localhost:8080/admin/calendar
Apps
Teams
PES
Colab
Cisco Jobs
Strivers
The HackerRank Int...
Capstone
All Bookmarks
Upcoming Approved Events Calendar
2025-04-26
AT
Concert and Comedy Nights
Venue: MRD
Organizer: Organizer1
2025-05-10
Coding Contest
2 hours coding contest
Venue: MRD
Organizer: Organizer2
2025-05-22
Summer Intern Test
24 hour hackathon , only girls
Venue: 13th Floor
Organizer: Organizer2
← Back to Dashboard

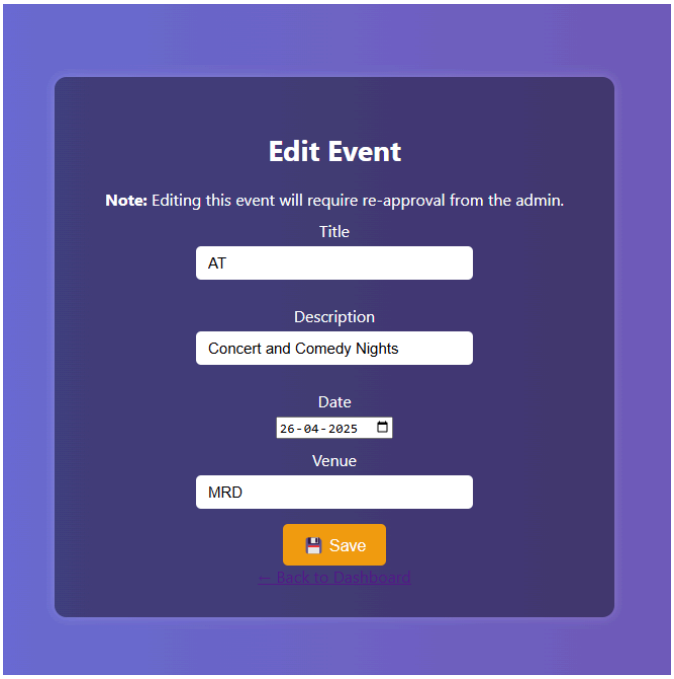
2. Organizer

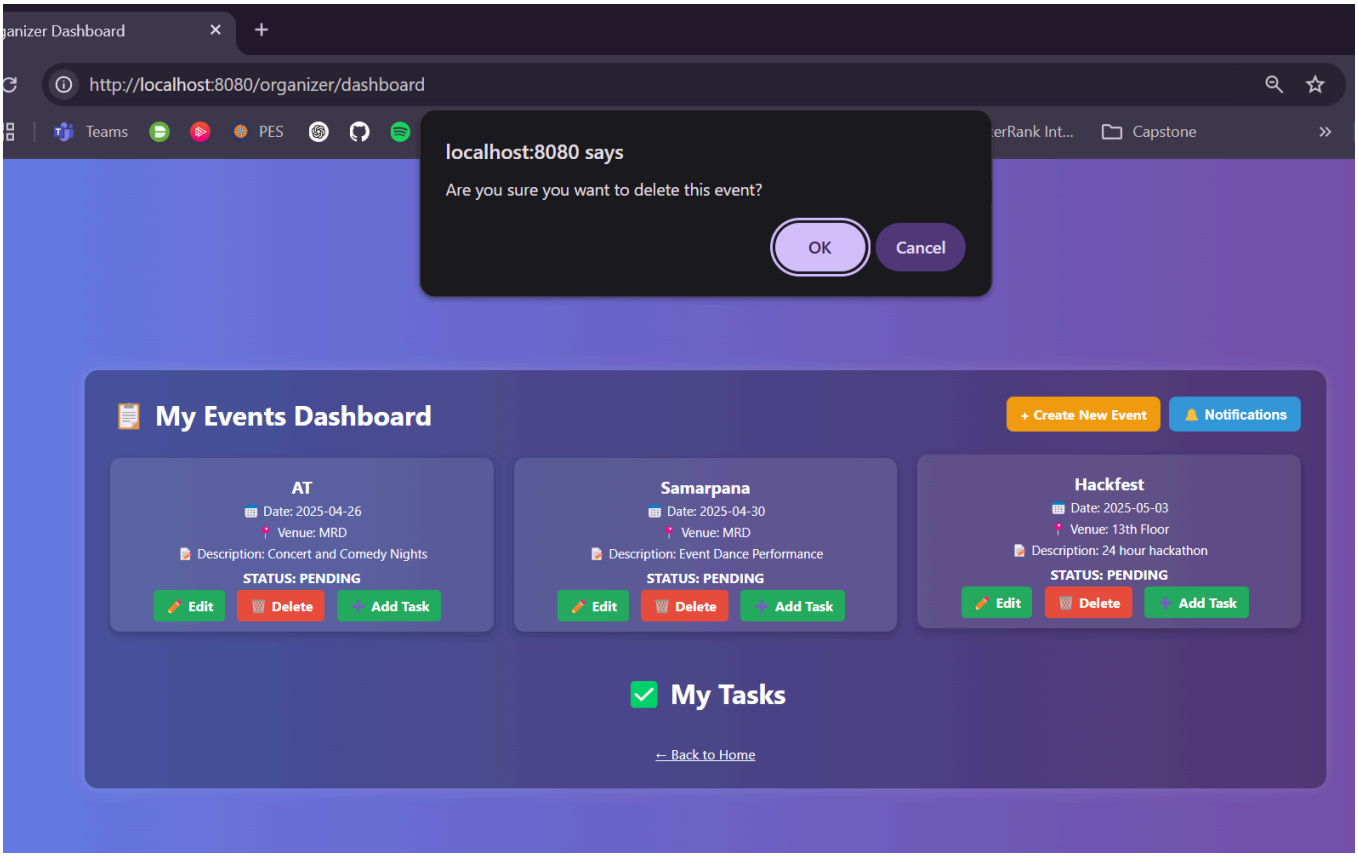

Organizer Signup
Username:
Organizer1
Password:
Email:
Organizer1@gmail.com
Sign Up
← Already registered? Login

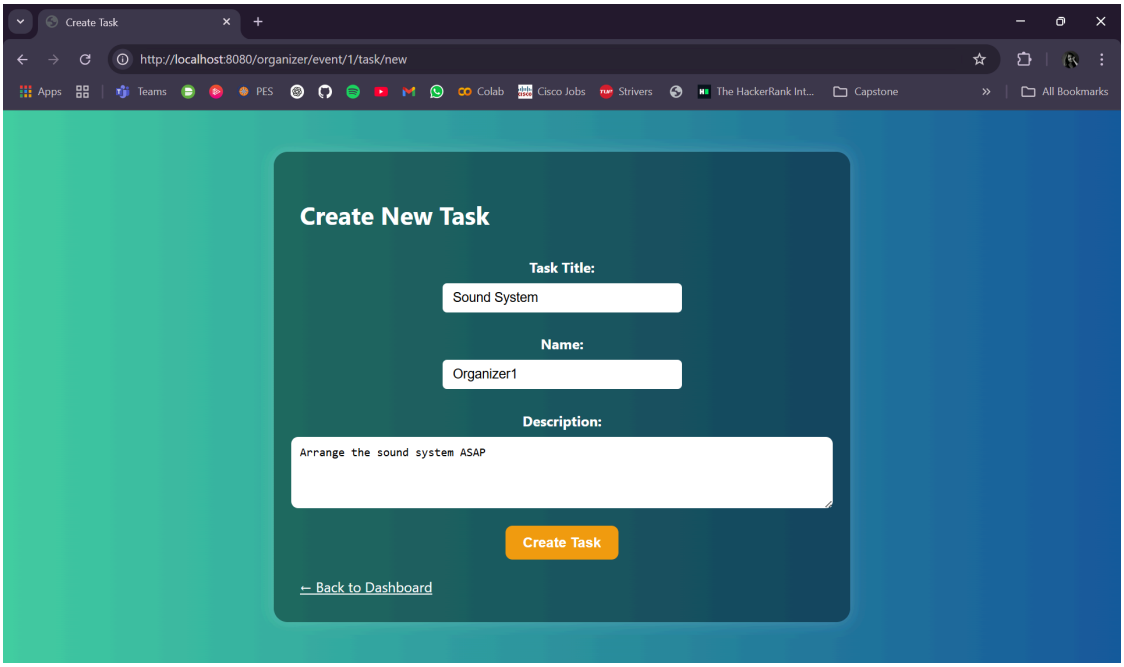

Username:
Organizer1
Organizer Login
Password:
Don't have an account? Sign Up
Login

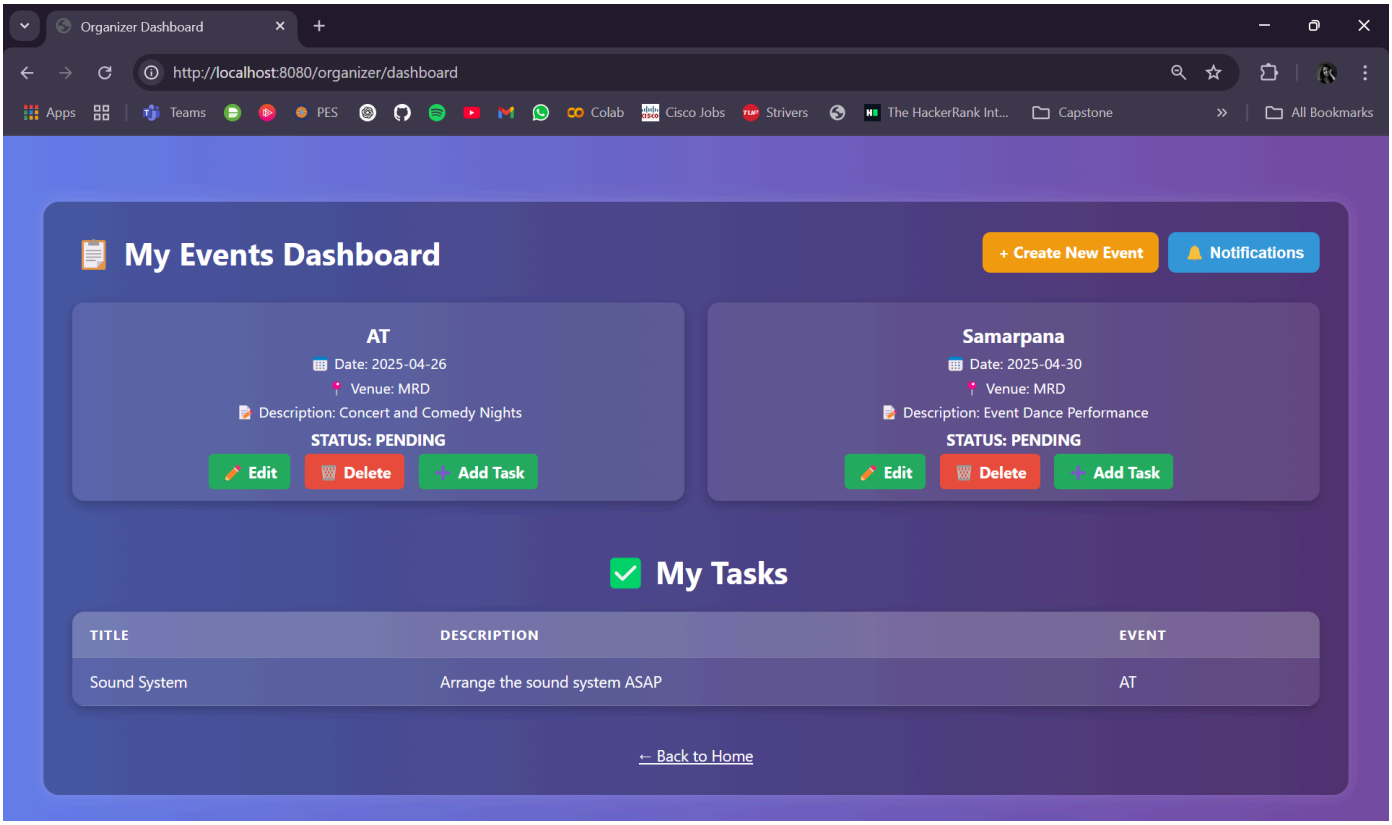

Organizer Dashboard
http://localhost:8080/organizer/dashboard
My Events Dashboard
+ Create New Event
Notifications
AT
Date: 2025-04-26
Venue: GJB
Description: Concert and Comedy Nights
STATUS: PENDING
Edit
Delete
Add Task
Samarpana
Date: 2025-04-30
Venue: MRD
Description: Event Dance Performance
STATUS: PENDING
Edit
Delete
Add Task
Hackfest
Date: 2025-05-03
Venue: 13th Floor
Description: 24 hour hackathon
STATUS: PENDING
Edit
Delete
Add Task
My Tasks
← Back to Home

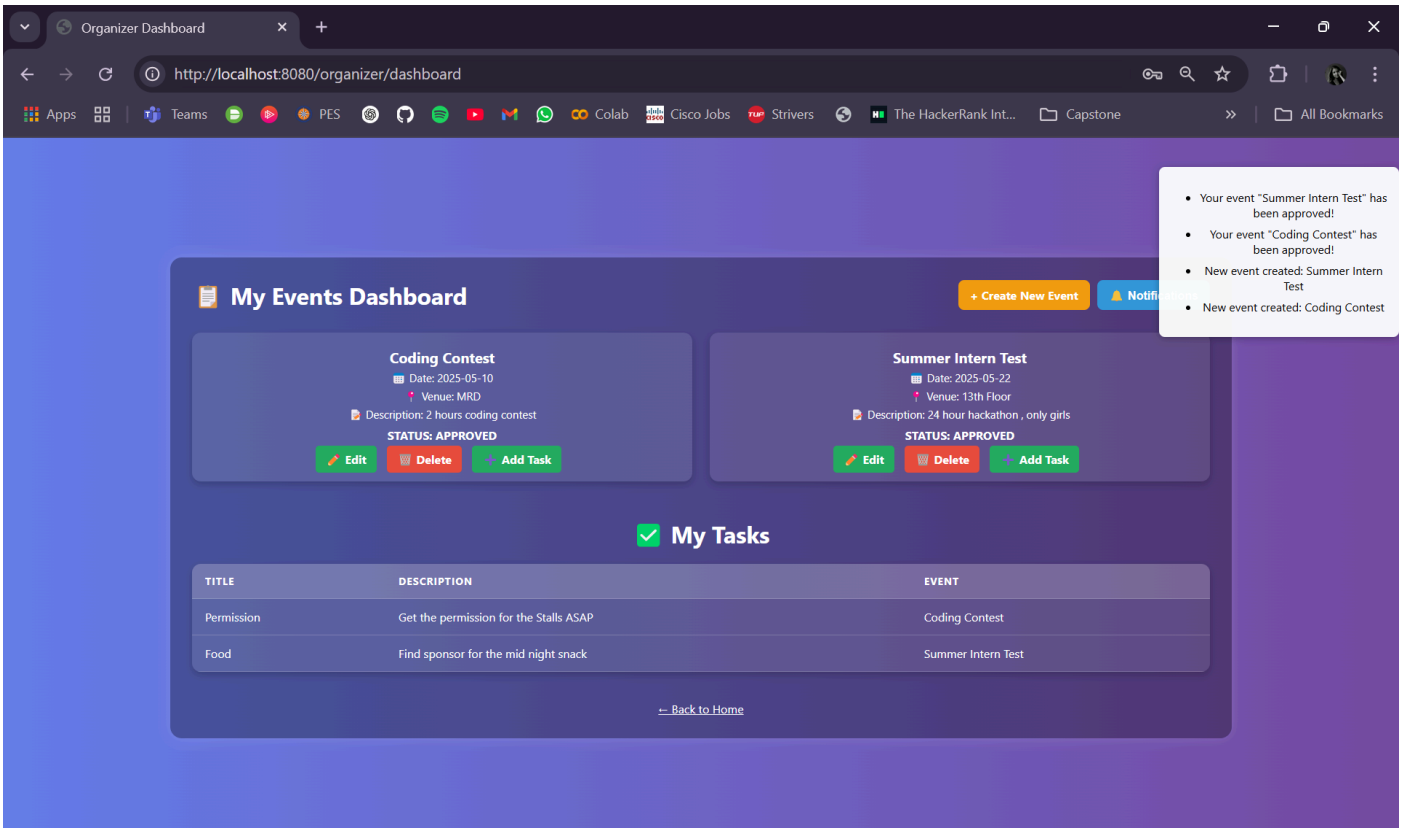

alhost:8080/organizer/events/new
PES
Colab
Cisco Jobs
Strivers
The HackerRank Int...
Capstone
Create New Event
Title:
AT
Description:
Concert and Comedy Nights
Venue:
GJB
Date:
26-04-2025
Location:
BLR
Name:
Atmatrisha Team
Category:
Fest
Create Event
← Back to Dashboard


Edit Event
Note: Editing this event will require re-approval from the admin.
Title
AT
Description
Concert and Comedy Nights
Date
26-04-2025
Venue
MRD
Save
← Back to Dashboard

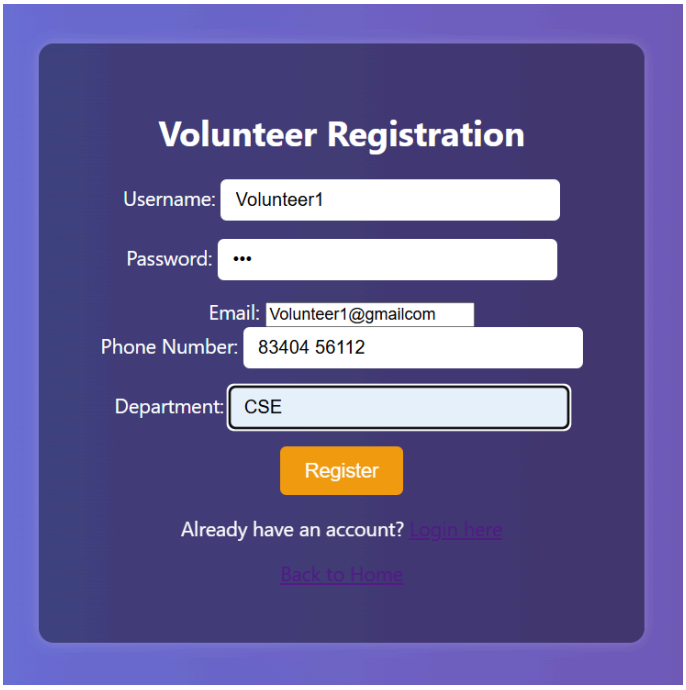

ganizer Dashboard
http://localhost:8080/organizer/dashboard
Teams
PES
erRank Int...
Capstone
localhost:8080 says
Are you sure you want to delete this event?
OK
Cancel
My Events Dashboard
+ Create New Event
Notifications
AT
Date: 2025-04-26
Venue: MRD
Description: Concert and Comedy Nights
STATUS: PENDING
Edit
Delete
Add Task
Samarpana
Date: 2025-04-30
Venue: MRD
Description: Event Dance Performance
STATUS: PENDING
Edit
Delete
Add Task
Hackfest
Date: 2025-05-03
Venue: 13th Floor
Description: 24 hour hackathon
STATUS: PENDING
Edit
Delete
Add Task
My Tasks
← Back to Home

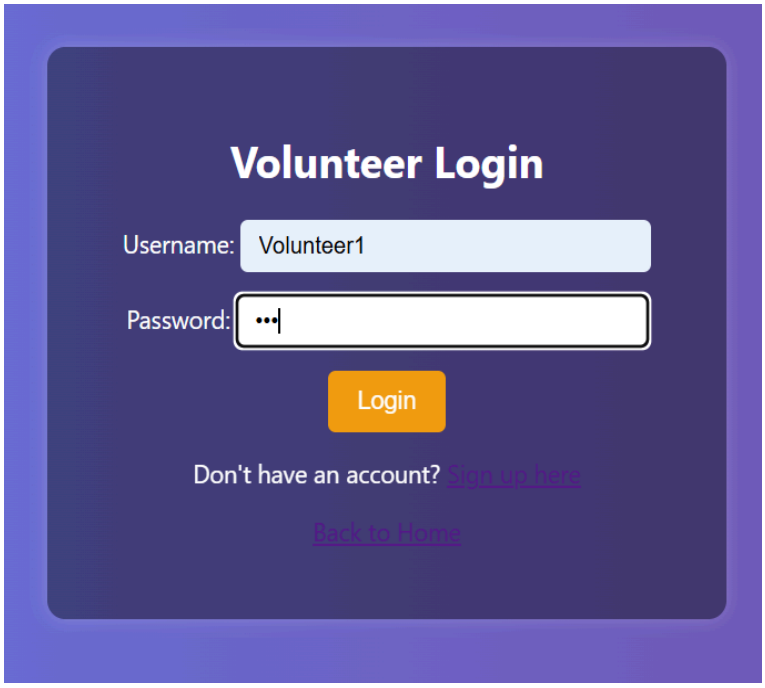

Create Task
http://localhost:8080/organizer/event/1/task/new
Apps
Teams
PES
Colab
Cisco Jobs
Strivers
The HackerRank Int...
Capstone
All Bookmarks
Create New Task
Task Title:
Sound System
Name:
Organizer1
Description:
Arrange the sound system ASAP
Create Task
← Back to Dashboard


Organizer Dashboard
http://localhost:8080/organizer/dashboard
Apps
Teams
PES
Colab
Cisco Jobs
Strivers
The HackerRank Int...
Capstone
All Bookmarks
My Events Dashboard
+ Create New Event
Notifications
AT
Date: 2025-04-26
Venue: MRD
Description: Concert and Comedy Nights
STATUS: PENDING
Edit
Delete
Add Task
Samarpana
Date: 2025-04-30
Venue: MRD
Description: Event Dance Performance
STATUS: PENDING
Edit
Delete
Add Task
My Tasks
TITLE
DESCRIPTION
EVENT
Sound System
Arrange the sound system ASAP
AT
← Back to Home


Organizer Dashboard
http://localhost:8080/organizer/dashboard
Apps
Teams
PES
Colab
Cisco Jobs
Strivers
The HackerRank Int...
Capstone
All Bookmarks
Your event "Summer Intern Test" has been approved!
Your event "Coding Contest" has been approved!
New event created: Summer Intern Test
New event created: Coding Contest
My Events Dashboard
+ Create New Event
Coding Contest
Date: 2025-05-10
Venue: MRD
Description: 2 hours coding contest
STATUS: APPROVED
Edit
Delete
Add Task
Summer Intern Test
Date: 2025-05-22
Venue: 13th Floor
Description: 24 hour hackathon , only girls
STATUS: APPROVED
Edit
Delete
Add Task
My Tasks
TITLE
DESCRIPTION
EVENT
Permission
Get the permission for the Stalls ASAP
Coding Contest
Food
Find sponsor for the mid night snack
Summer Intern Test
← Back to Home

3. Volunteer


Volunteer Registration
Username: Volunteer1
Password: •••
Email: Volunteer1@gmailcom
Phone Number: 83404 56112
Department: CSE
Register
Already have an account? Login here
Back to Home


Volunteer Login
Username: Volunteer1
Password: •••
Login
Don't have an account? Sign up here
Back to Home

Volunteer Dashboard

Logout

Welcome, Volunteer1

USERNAME	EMAIL	PHONE	DEPARTMENT
Volunteer1	Volunteer1@gmailcom	83404 56112	CSE

Upcoming Events

2025-04-26
AT
Concert and Comedy Nights
MRD Organizer1

2025-05-10
Coding Contest
2 hours coding contest
MRD Organizer2

2025-05-22
Summer Intern Test
24 hour hackathon , only girls
13th Floor Organizer2

Tasks

All Tasks

Title	Description	Associated Event
Sound System	Arrange the sound system ASAP	AT
Food	Find sponsor for the mid night snack	Summer Intern Test
Permission	Get the permission for the Stalls ASAP	Coding Contest

Participant Attendance Management

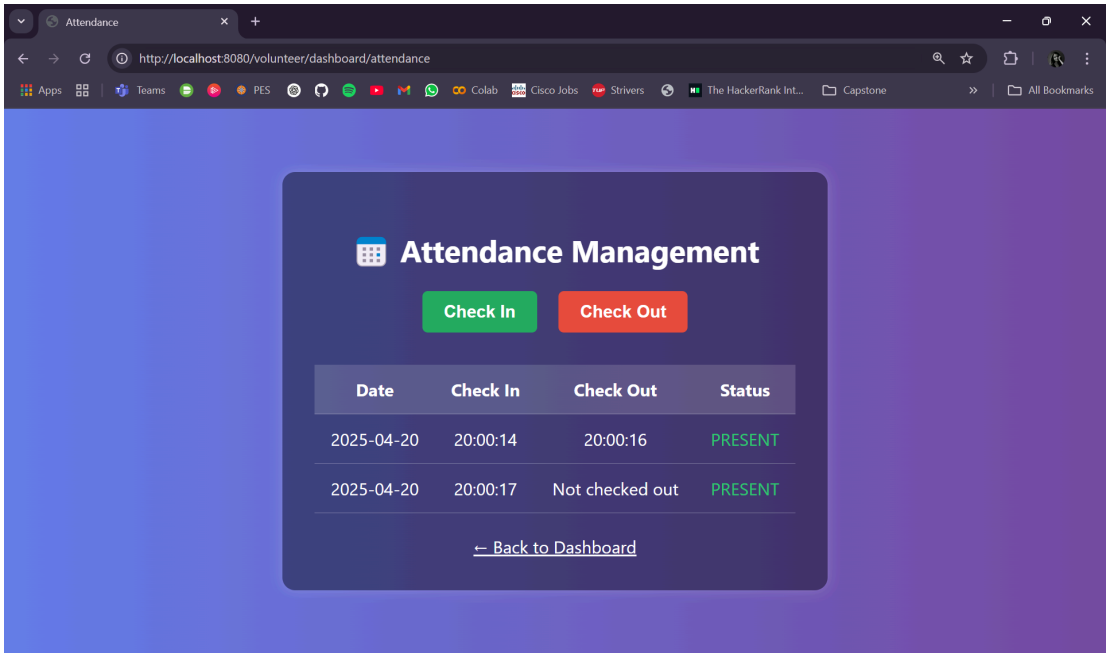
Mark participants as present or absent for events

Participant	Event	Date	Status	Actions
Participant1	Coding Contest	2025-05-10	Pending	Present Absent
Participant1	Summer Intern Test	2025-05-22	Present	Present Absent
Participant2	Summer Intern Test	2025-05-22	Absent	Present Absent
Participant2	AT	2025-04-26	Present	Present Absent
Participant3	Coding Contest	2025-05-10	Pending	Present Absent

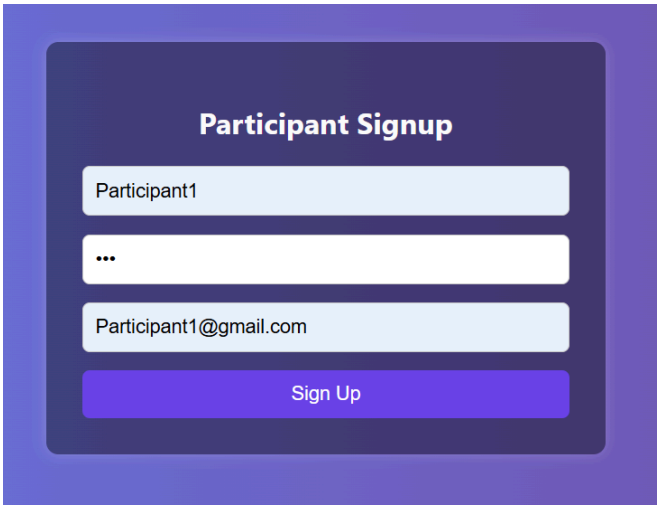
Your Attendance

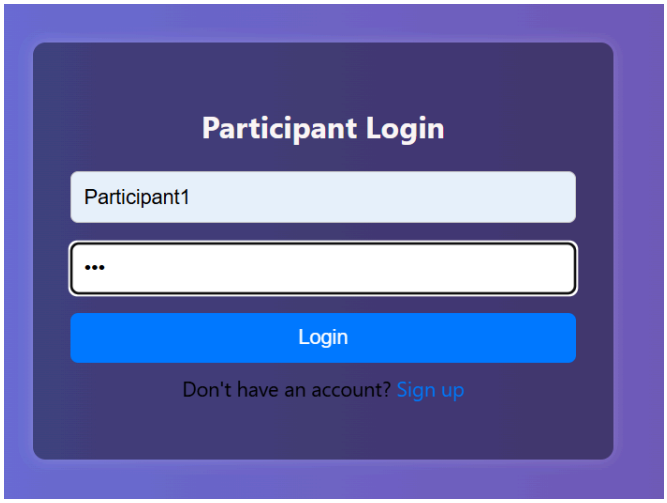
Manage your own attendance records

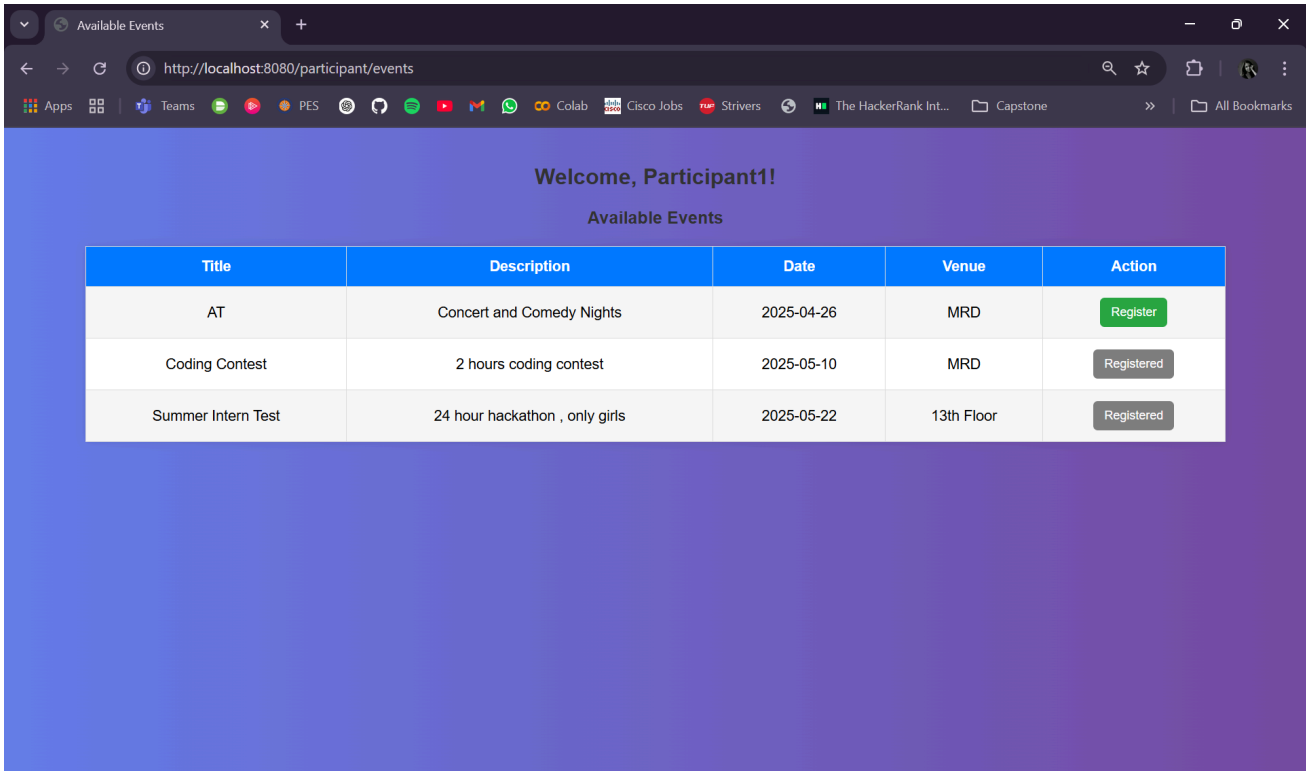
Check In



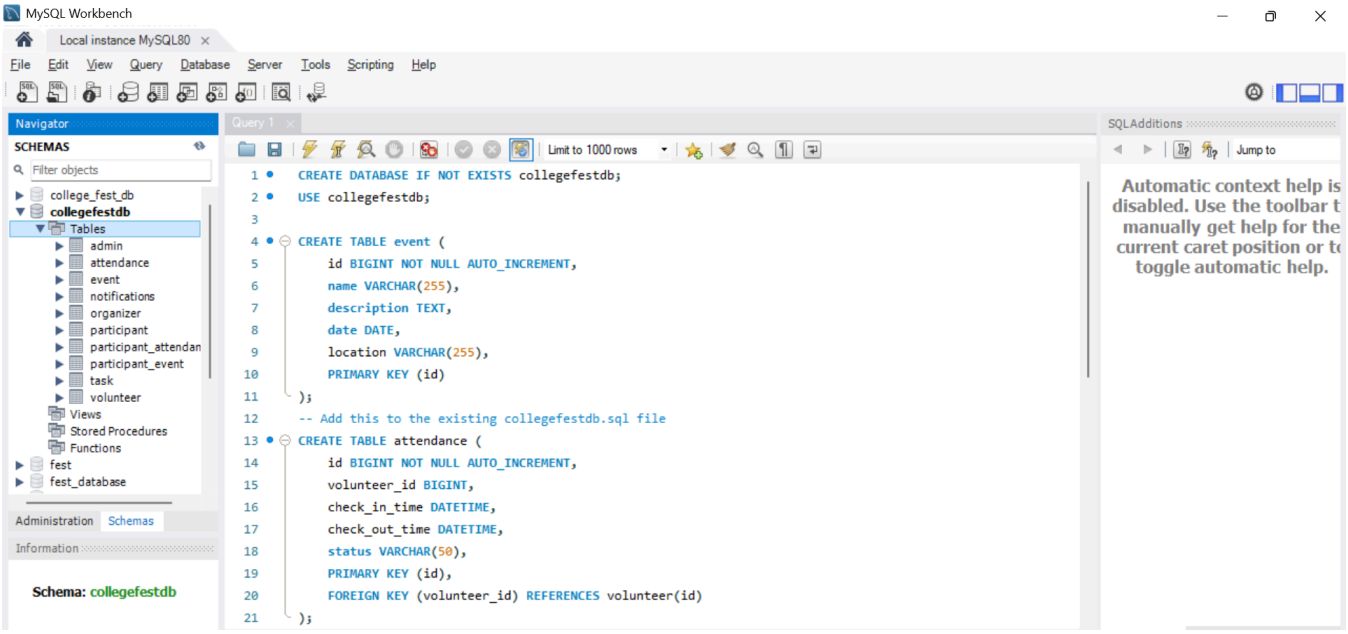
4. Participant







Database Structure:



Individual contributions:

Name	Module worked on
Praneetha Praveen Kalbhavi	Organizer, Volunteer
Prapti Arali	Organizer, Volunteer
Prayashi Verma	Organizer, Volunteer, Participants
Priyanka Kumari	Admin, Organizer