



# K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

Recognised under the section 2 (f) of the UGC Act 1956



*Empowering the Youth; Empowering the Nation*





**K.R. MANGALAM UNIVERSITY**

THE COMPLETE WORLD OF EDUCATION

# MINI PROJECT

**B.TECH CSE [CORE]**

**First Year Project Synopsis**

**Submitted by:**

NAME	ROLL NO.
ANSHIKA KASHYAP	2401010021
KHUSHI	2401010089

## Project Overview

# College Event Management System

- The College Event Management Website is a centralized platform designed to automate and simplify the planning, registration, and certification process of college events and workshops.
- The system provides a 3D user interface experience and supports dark mode, theme customization, and role-based dashboards.
- It ensures smooth coordination among students, faculty, and event coordinators while keeping track of upcoming activities, registrations, and certificate generation.



# Problem Statement

---

In many educational institutions, managing college events, workshops, and certificate distribution is still done manually or through disconnected tools. This leads to several challenges such as inefficient registration processes, poor communication among stakeholders, absence of a centralized event calendar, and delays in certificate issuance.

Additionally, students lack a personalized dashboard to track their participation, and faculty face difficulties in coordinating events and verifying attendees. These issues highlight the urgent need for a unified, interactive, and role-based platform that streamlines event management and enhances user experience for students, faculty, and coordinators alike.

# Problem Identified:

## 1. **Manual Event Coordination**

organizing campus events required offline coordination, leading to delays and miscommunication.

## 2. **Scattered Information**

Notices, certificates, and event details being managed from different platforms,(emails, paper, verbal), causing confusion. Tracking student participation and event registration was difficult

## 3. **No Central Access for Admins**

Admins had no unified platform to manage academic or extracurricular data.

## 4. **Lack of Real-Time Updates**

Students and faculty weren't getting timely updates due to delayed communication.

## 5. **Limited Role-Based Access**

Without a system, anyone with basic access could modify or view sensitive information.

## 6. **No Upload Mechanism for Certificates**

Issuing and managing student certificates lacked a secure upload/download process.



# Objectives

To develop a web-based platform for end-to-end event management

To implement secure login/signup functionality with role-based access

To allow students to register for events and view participation history

To enable faculty to create, manage, and monitor events/workshops

To allow coordinators to oversee event operations and status

To integrate an event calendar for visual timeline tracking

To auto-generate and provide downloadable certificates to students

To include a settings panel for UI customization (dark mode, themes, volume)

To provide notifications and updates for upcoming or modified



# Solution

1. Offers a modern, responsive 3D UI with theme and dark mode customization
2. Increases student engagement through a user-friendly dashboard and event calendar
3. Enable quick and secure certificate generation and download
4. Improve communication and transparency in event management
5. Enhance user experience with modern UI and customization options
6. Streamlines the entire event registration and management process
7. Centralizes event data for easy access by students, faculty, and coordinators
8. Speeds up certificate generation and provides instant downloads for students



# Methodology and Implementation

1.

**Programming Language:** Python  
used for readability, rapid development, and rich ecosystem.

2.

**Web Framework:** Django (High-level Python Web Framework),  
includes built-in admin features.

3.

**Database:** SQLite (default during development)  
Lightweight, file-based database managed by Django.

4.

**Authentication & Authorization:**  
Uses *Django's built-in User model* and ***admin permissions***,  
(*is\_staff, is\_superuser*) for secure access control.

5.

**Admin Interface:** Django Admin Panel, Auto-generated CRUD interface  
based on models. Allows admin/staff users to manage data (events, notices,  
certificates)

6.

**File Handling:**  
*FileField* for certificate uploads.  
Django handles storage and URL routing via MEDIA\_URL and MEDIA\_ROOT.

7.

**Development Server:**  
Runs via `python manage.py runserver`





# Methodology Flowchart

## Programming Language

Python is known for its readability and rich ecosystem. It allows rapid development.



## Database

SQLite is a lightweight, file-based relational database. It is managed by Django ORM during development.



## Admin Interface

Django Admin Panel is an auto-generated CRUD interface based on models. It allows data management without coding.



## Development Server

Runs via python manage.py runserver. Django's lightweight WSGI server used in local development.



## Web Framework

Django follows the MVT architecture and includes built-in admin features. It is a high-level Python Web Framework.



## Authentication & Authorization

Uses Django's built-in User model and admin permissions for secure access control. This includes is\_staff and is\_superuser.

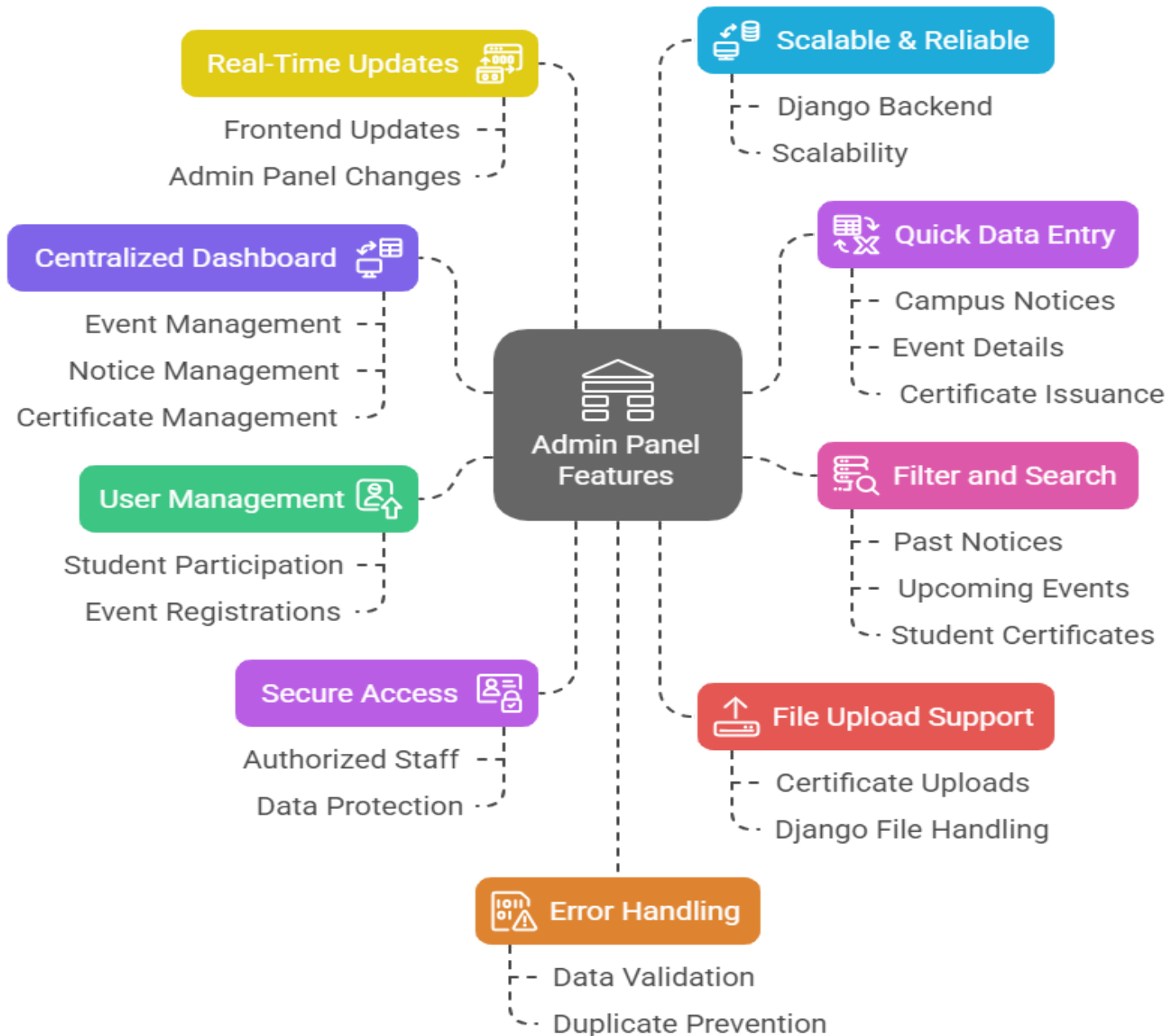


## Media/File Handling

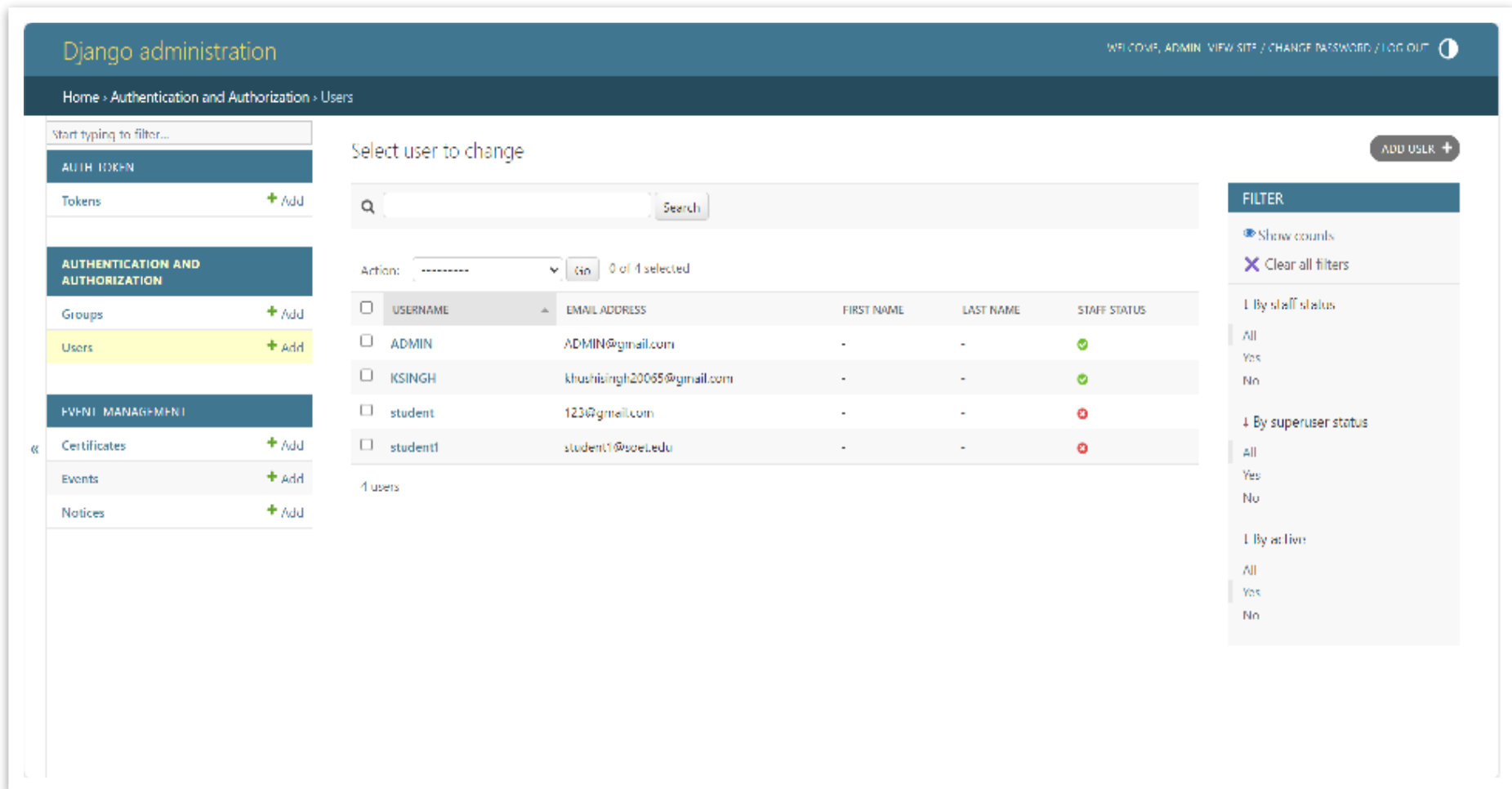
FileField is used for certificate uploads. Django handles storage and URL routing via MEDIA\_URL and MEDIA\_ROOT.



# Results and Implementation



# ADMIN INTERFACE



## ADMIN-USER PANEL

Start typing to filter...

## AUTH TOKEN

Tokens [+ Add](#)

## AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)Users [+ Add](#)

## EVENT\_MANAGEMENT

Certificates [+ Add](#)Events [+ Add](#)Notices [+ Add](#)

## Select Token to change

ADD TOKEN [+](#)

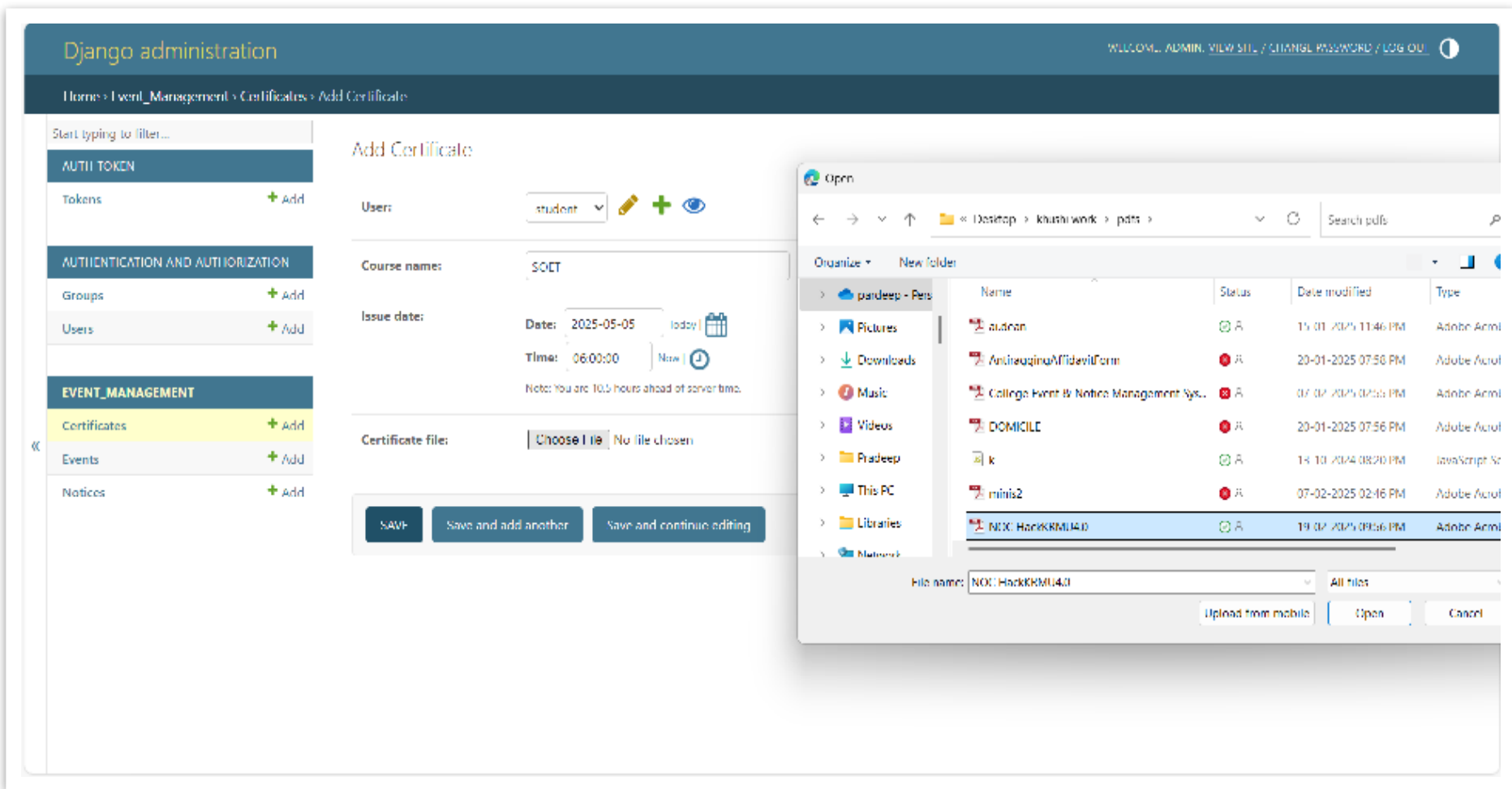
Search

Username

KEY	USER	CREATED
f49c9b256ebad5bde526bb1bcd03aede13c525af	student	April 28, 2025, 12:59 p.m.
0da1b347ce19636e3c9a5452cee7044ed03c9d1f	student1	April 27, 2025, 12:56 p.m.
d9d7764cc50302cf737243Ba66bf8e20b50266ee	KSINGH	April 29, 2025, 7:44 p.m.

3 Tokens

# USER TOKEN GENERATION



## CERTIFICATE UPLOAD FOR STUDENTS VIA ADMIN

Start typing to filter...


## AUTH TOKEN

Tokens [+ Add](#)

## AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)Users [+ Add](#)

## EVENT MANAGEMENT

Certificates [+ Add](#)Events [+ Add](#)Notices [+ Add](#) The Event "hackathon workshop 2.0" was changed successfully.


Select Event to change

ADD EVENT [+](#) Action:   0 of 1 selected

<input type="checkbox"/>	TITLE	DATE	LOCATION
<input type="checkbox"/>	hackathon workshop 2.0	April 3, 2025, noon	A BLOCK, A213

1 Event

## FILTER

 [Show counts](#)[↓ By date](#)[Any date](#)[Today](#)[Past 7 days](#)[This month](#)[This year](#)

# EVENT MANAGEMENT

Start typing to filter...

## AUTH TOKEN

Tokens [+ Add](#)

## AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)Users [+ Add](#)

## EVENT MANAGEMENT

Certificates [+ Add](#)Events [+ Add](#)Notices [+ Add](#)

## Change Notice

## LATE FEE

HISTORY


Title:

LATE FEE

Content:

Students of even semester 2 must clear their dues by the end of month april.  
contact: 98929XXXXX  
gmail: ADMIN@gmail.com

Published date:

Date: 2025-04-17  TodayTime: 18:00:00  Now

Note: You are 10.5 hours ahead of server time.

SAVE

Save and add another

Save and continue editing

Delete

## NOTICES MANAGEMENT



## Site administration

## AUTH TOKEN

Tokens [+ Add](#) [Change](#)

## AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [Change](#)Users [+ Add](#) [Change](#)

## EVENT\_MANAGEMENT

Certificates [+ Add](#) [Change](#)Events [+ Add](#) [Change](#)Notices [+ Add](#) [Change](#)

## Recent actions

## My actions

[Change](#) I AM HERE  
Notice[+ Add](#) I AM HERE  
Notice[Change](#) hackathon workshop 2.0  
Event[+ Add](#) hackathon workshop 2.0  
Event[X](#) webinar soon  
Event[X](#) pradeep  
User[X](#) potato  
User[X](#) khushi  
User[X](#) gfn  
User[X](#) abcd12  
User**TOGGLE BUTTON FOR THEMES**

# Project Timeline

**Week 1-2:** Research & Requirement Gathering.



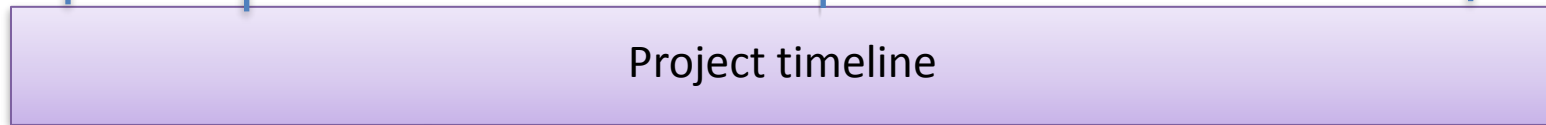
**2. Week 3-4:** Admin platform designing



**4. Week 8-10:** Frontend Development & Integration.



**6. Week 13-18:** Deployment & Final Presentation.



**5. Week 11-12:** Testing & Debugging.



**3. Week 5-7:** Database Setup & Backend Development.



# Expected Results & Impact

## Expected Results:

- Streamlined event management process
- Enhanced user engagement with role-specific dashboards
- Real-time updates and notifications for all users
- Faster and automated certificate generation and distribution
- Increased student participation in events through a central calendar

## Impact:

- Students: Easier registration, tracking, and certificate access
- Faculty: Reduced administrative workload and quicker event management
- Coordinators: Clear overview of events and participant status
- College: Improved organizational efficiency and reduced errors
- Long-Term: Scalable platform for future enhancements like analytics



- Integration of SMS/Email notifications for event reminders and updates

## Future Scope

- Implementation of event feedback systems for improved event planning
- Addition of admin analytics for better decision-making and event insights
- Support for multi-language and accessibility features
- Mobile app development for easier access to events on the go
- Integration with third-party calendar tools (e.g., Google Calendar) for seamless scheduling
- Payment gateway integration for paid events or workshops
- Implementation of user ratings/reviews for events and workshops
- Event streaming capabilities for remote participation in workshops and seminars

# References

1. W3Schools - Web development tutorials and examples for HTML, CSS, JavaScript, and backend frameworks.  
<https://www.w3schools.com/>
2. MDN Web Docs - Comprehensive documentation on web technologies including HTML, CSS, JavaScript, and more.  
<https://developer.mozilla.org/>
3. React Official Documentation - Guide to developing dynamic UIs using React framework.  
<https://reactjs.org/>
4. FastAPI Documentation - For backend development with Python and FastAPI.  
<https://fastapi.tiangolo.com/>
5. SQLite Documentation - For database setup and integration.  
<https://www.sqlite.org/>



# THANK YOU

