

# CAMS (Campus Activity Management System)

---

Group  
Members:

1. Gabriel M S NIE22CS023
2. Joseph Mohan NIE22CS029
3. Akshaykumar V S LNIE22CS054

# 1. Abstract



---

CAMS is an online platform designed to efficiently manage and streamline campus activities, including sports, arts, and events. It involves three user roles:

- **Admin** – Manages the system, faculty, students, and activities.
- **Faculty** – Creates, modifies, and monitors activities; issues participation certificates.
- **Student** – Views and participates in activities, tracks progress, and downloads certificates.

This system enhances efficiency, simplifies participation, and provides an organized approach to campus activity management.

# 2. Proposed Tools & Technologies



---

- **Backend:** Django (Python) – Fast, secure, and scalable web framework.
- **Database:** MySQL – Reliable and efficient relational database.
- **Frontend:** HTML5, CSS3, JavaScript – For responsive and dynamic UI.
- **Templating Engine:** Django Templates – Dynamic content rendering.

- 
- **Authentication:** Django Authentication System – Secure user management.
  - **File Storage:** Django File Storage – Manages certificates & reports.
  - **Data Visualization:** Chart.js – Generates interactive graphs & reports.
  - **Version Control:** Git & GitHub – Tracks and manages code changes.
  - **Deployment:** Initially on a local machine.
  - **Development & Testing:** VS Code & Django Test Framework.

So these are the proposed tools & technologies going to be used to build the project.

# 3.Introduction about the proposed project

---

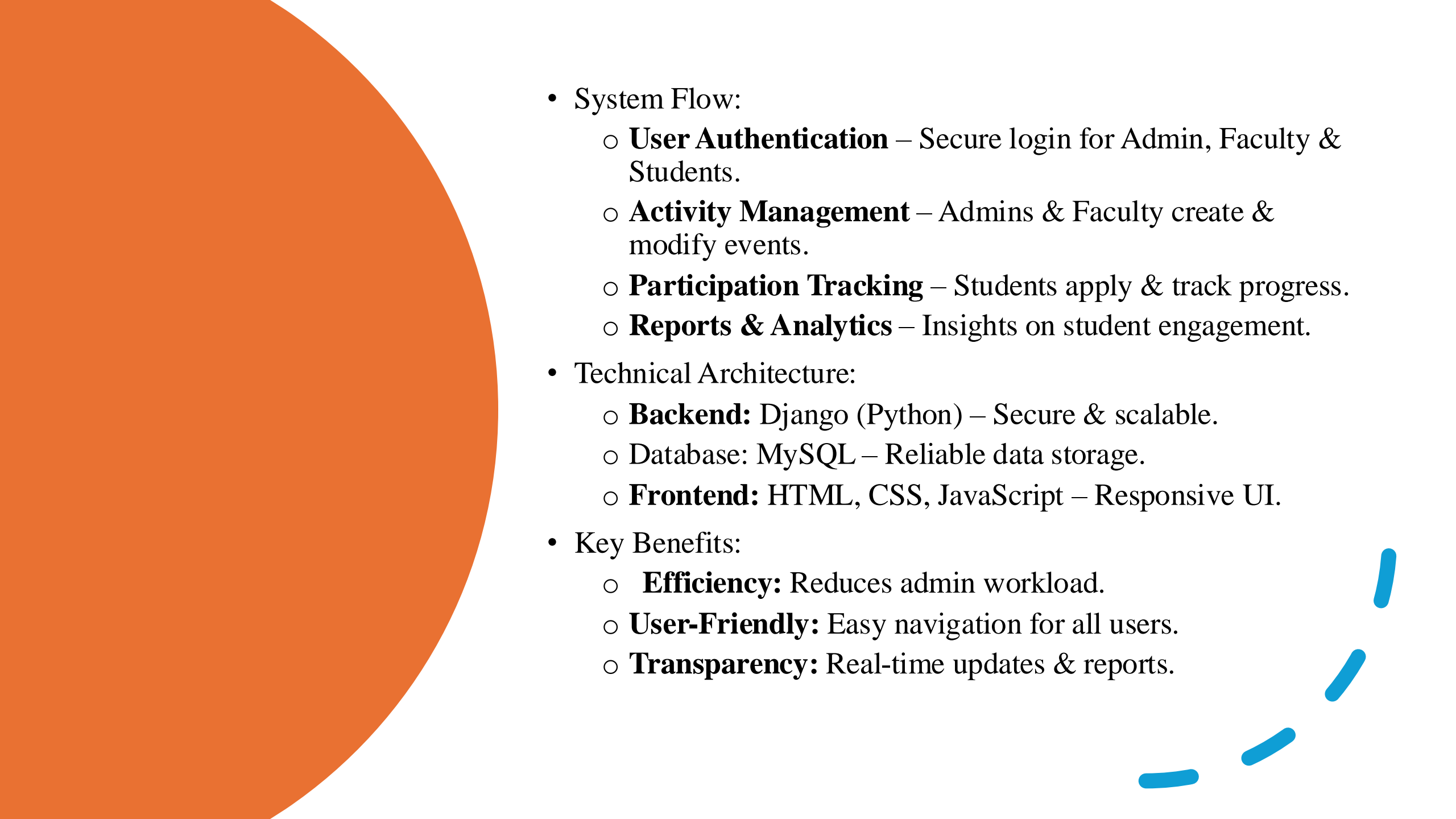
- **Objective:** A web-based platform to streamline campus activity management and participation.
- **Need:** Traditional methods are manual, time-consuming, and inefficient.
- **Solution:** CAMS provides a **centralized, digital platform** for seamless event organization and engagement.
- User Roles:
  - **Admin** – Manages users & activities, oversees participation.
  - **Faculty** – Creates & monitors activities, verifies participation.
  - **Students** – Registers for activities, tracks progress, downloads certificates.

- 
- **Technology Stack:** Django (Python), MySQL, HTML/CSS, JavaScript.
  - **Benefits:**
    - **Automates event management** – Reduces paperwork & admin workload.
    - **Enhances student participation** – Easy registration & tracking.
    - **Provides transparency** – Real-time updates & reports.

# 4. System Overview

---

- **Purpose:** Automates and simplifies campus activity management.
- **User Roles & Features:**
  - **Admin:** Manages users & activities, generates reports.
  - **Faculty:** Creates & verifies activities, issues certificates.
  - **Students:** Registers, tracks progress, downloads certificates.

- 
- System Flow:
    - **User Authentication** – Secure login for Admin, Faculty & Students.
    - **Activity Management** – Admins & Faculty create & modify events.
    - **Participation Tracking** – Students apply & track progress.
    - **Reports & Analytics** – Insights on student engagement.
  - Technical Architecture:
    - **Backend:** Django (Python) – Secure & scalable.
    - Database: MySQL – Reliable data storage.
    - **Frontend:** HTML, CSS, JavaScript – Responsive UI.
  - Key Benefits:
    - **Efficiency:** Reduces admin workload.
    - **User-Friendly:** Easy navigation for all users.
    - **Transparency:** Real-time updates & reports.



# 5. Design Considerations

---

- Assumptions & Dependencies:
  - **Software:** Django (Python), MySQL, Web Browsers.
  - **Hardware:** Sufficient storage for file uploads (certificates, reports).
  - **Network:** Requires stable internet for real-time access.
  - **Email Service:** For notifications & confirmations.
- General Constraints:
  - **Hardware Limitations:** Server resources must support scalability.
  - **Database Growth:** Needs optimized storage for large datasets.
  - **Internet & Browser Compatibility:** Cross-browser responsive design.
  - **User Access Control:** Role-based access (Admin, Faculty, Student).

- 
- System Maintenance & Security:
    - Regular updates, backups, & security patches.
    - User authentication & data encryption for privacy.
    - Prevents unauthorized access with strict access control.
  - Time Constraints:
    - Prioritize core functionalities for initial release.
    - Optimize & expand features in future updates.

## 6. Developmental Method

### Agile Development (Scrum Framework)

- **Why Chosen:** Flexibility, stakeholder involvement, and rapid prototyping. Agile supports evolving requirements, regular feedback, and quick delivery of working prototypes.
- **Key Features:**
  - Incremental development in manageable sprints
  - Collaboration and feedback from faculty, students, and admins
  - Continuous improvement with sprint retrospectives
  - User stories and prioritization of critical features

- 
- MVC design patterns:
    - **Why Chosen:** Clear separation of concerns, scalability, and ease of maintenance. The MVC pattern supports large applications, facilitating organized development.
    - Structure:
      - **Model:** Data and business logic (MySQL)
      - **View:** User interface (HTML templates)
      - **Controller:** Handles user requests and interaction between model and view
    - **Framework Support:** Django follows the MVC (MTV) pattern, aligning with the project's needs.

This combination ensures an adaptable, scalable, and user-focused system.

---

# 7. System Architecture

## 1. **Modular Architecture:**

The system is divided into smaller, manageable subsystems, ensuring scalability, maintainability, and adaptability. Three main layers:

- **Presentation Layer:** Frontend interface (user interaction).
- **Application Layer:** Backend logic (handling user requests and data processing).
- **Data Layer:** Database (stores all system data).

## 2. **Main Responsibilities:**

- **User Authentication & Authorization:** Secure access for Admins, Faculty, and Students.



- 
- **Activity Management:** Admins and Faculty create, update, and manage activities; Students browse and apply.
  - **Participation Tracking:** Monitors student participation and verifies status.
  - **Certificate Generation:** Generates downloadable certificates for successful activity completion.
  - **Reporting & Analytics:** Provides detailed reports on student activity and performance

### 3. Subsystems & Components:

- **User Management:** Handles user authentication, roles, and profile management.
- **Activity Management:** Allows creation, modification, and participation in activities.

- 
- **Participation Management:** Tracks student participation and verification by faculty.
  - **Certificate Generation:** Generates and allows downloading of participation certificates.
  - **Reporting & Analytics:** Provides activity and performance reports.
  - **Notification & Communication:** Sends email updates and alerts for activity statuses.
  - **Data Management:** Manages the storage and backup of system data.

#### 4. **Component Interaction:**

- Clear communication between subsystems via interfaces and API endpoints.
- Example Flow:

- 
- Faculty/Admins manage activities, students apply, participation is tracked, certificates are generated, and reports are created based on the data.

## **5. Collaboration:**

- All subsystems work in tandem to provide a seamless user experience and efficient system operation. Data integrity and security are maintained through MySQL database and backup processes.

## **6. Low-Level Design:**

- UML Diagrams (use cases, class diagrams, and sequence diagrams) define the system interactions and structure.

This architecture ensures the system is flexible, efficient, and scalable while offering a seamless experience for all users.



# 8. Detailed System Design

---

## 1. System Architecture

The system follows the **MVC (Model-View-Controller)** architecture using the **Django framework** to separate concerns and ensure scalability, maintainability, and flexibility.

- **Frontend:** Django templates utilizing HTML, CSS, JavaScript, and Bootstrap for a responsive user interface.
- **Backend:** The system is built on the **Django framework** using Python for efficient server-side logic.
- **Database:** **MySQL** is used to store and manage the system's data, ensuring fast retrieval and scalability.
- **Authentication & Authorization:** **Django Authentication System** for managing user login, roles, and access control.

- **Hosting:** Deployed either on a **cloud server** (e.g., AWS, Azure) or an **on-premise setup**.

## 2. Key System Components:

- **User Management System:**

- Role-based authentication for **Admin**, **Faculty**, and **Student** roles
- **Registration** and **Login** system with **email-based verification**
- Profile management functionality allowing users to view/edit personal details

- **Activity Management:**

- Admins and Faculty can perform **CRUD (Create, Read, Update, Delete)** operations for activities.
- Activities are categorized into various types like **Sports**, **Arts**, **Cultural**, etc
- Participation management allowing students to browse and apply for activities.

---

- **Participation Tracking:**

- Students can **apply for activities**
- Faculty can **approve or reject** applications.
- Upon completion of activities, faculty can **verify participation** and issue **certificates** for successful participation.

- **Reporting & Dashboard:**

- Track student activity and generate **semester-wise reports** for participation, points, and performance.
- Admin and Faculty have their own **dashboards** for quick insights into student engagement.
- Ability to **export reports** in **PDF/Excel** formats for further analysis or records.

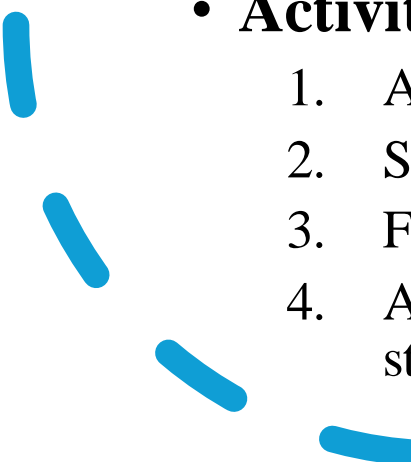


### 3. System Workflow:

- **User Authentication:**

1. Users **register** with an email, which is verified via email.
2. Role-based access control ensures that users only access appropriate sections of the system (Admin, Faculty, or Student).
3. Logged-in users can access their **personal dashboards** based on their role

- **Activity Management:**

1. Admin and Faculty create and manage activities.
  2. Students can browse available activities and **apply** for participation.
  3. Faculty review and either **approve** or **reject** student applications
  4. After activity completion, faculty **verify participation** and **issue certificates** to eligible students
- 

---

- **Participation & Reporting:**

1. Students can view their **participation history** and track **earned points**.
2. Faculty monitor student engagement and verify participation status.
3. Admins generate **semester-wise reports** based on data from participation and activity records.
4. Students can download **participation certificates** upon activity completion.





- 4. Security Measures:

- **Role-based Access Control (RBAC)** to restrict unauthorized access or modifications based on user roles.
- **Data Encryption** for sensitive information like passwords and personal data.
- **Input Validation & SQL Injection Prevention** to ensure system integrity and protect against malicious attacks.



## 5. Deployment Strategy:

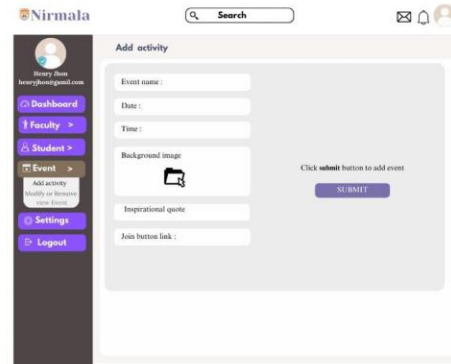
- **Version Control:** Utilizes **Git/GitHub** for managing code versions and collaboration among developers.
- **Database Hosting:** **MySQL** database hosted on either cloud servers or on-premise, ensuring reliable data storage.
- **CI/CD:** Automated deployment pipelines via **GitHub Actions**, ensuring smooth and error-free continuous integration and delivery.



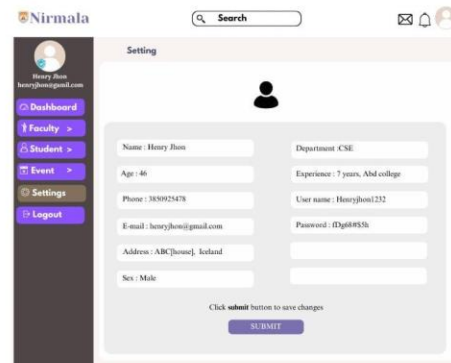
This design ensures the system is secure, user-friendly, and scalable while offering the flexibility needed to manage campus activities and student participation efficiently

# 9.UI/DB

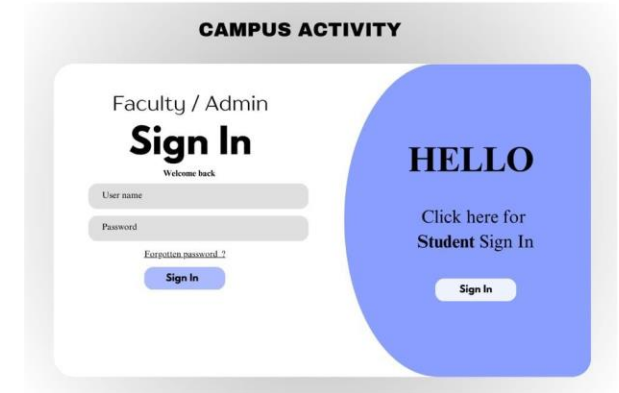
## 9.1 UI



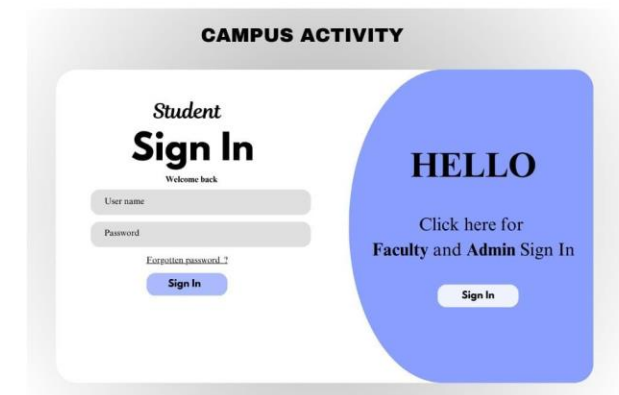
The 'Add activity' form in the Nirmala application features a sidebar with navigation links: Dashboard, Faculty, Student, Event, Add activity (active), Settings, and Logout. The main form area includes input fields for Event name, Date, Time, and Inspirational quote, a text area for Background image, and a text field for Join button link. A SUBMIT button is located at the bottom right. A message above the button states: 'Click submit button to add event'.



The 'Setting' form in the Nirmala application displays user profile information and allows for updates. The sidebar is identical to the 'Add activity' form. The main form area shows fields for Name (Henry Jhon), Department (CSE), Age (46), Experience (7 years, Abid college), Phone (989825478), User name (Henryjhon232), E-mail (henryjhon@gmail.com), Password (Dy@0#55b), Address (ABC@house1, Indland), and Sex (Male). A SUBMIT button at the bottom is accompanied by the instruction: 'Click submit button to save changes'.



The 'Faculty / Admin Sign In' screen, titled 'CAMPUS ACTIVITY', features a login form with fields for User name and Password, and a 'Sign In' button. A 'Welcome back' message is displayed above the form. To the right, a large blue circle contains the text 'HELLO' and a link 'Click here for Student Sign In' with a 'Sign In' button.



The 'Student Sign In' screen, also titled 'CAMPUS ACTIVITY', features a login form with fields for User name and Password, and a 'Sign In' button. A 'Welcome back' message is displayed above the form. To the right, a large blue circle contains the text 'HELLO' and a link 'Click here for Faculty and Admin Sign In' with a 'Sign In' button.

Nirmala

Search

Henry Jhon

henryjhon@gmail.com

Dashboard

Faculty >

Student >

Event >

Settings

Logout

Analytics Dashboard

Event

Students

Course

Event

Category

SPORTS

40 %

MARCH 2022

Faculty

Name : John

Depart: Mech

Name : Phalita

Depart: CSE

Nirmala

Search

Henry Jhon

henryjhon@gmail.com

Dashboard

Faculty >

Student >

Event >

Settings

Logout

Modify / Remove Activity

Event name : SPORTS

Date : 01/05/2025

Time : 9:00 - 5:00

Background image

Inspirational quote: Never Ever Give Up

Click here to SAVE changes

Click here to DELETE Event

Nirmala

Search

Henry Jhon

henryjhon@gmail.com

Dashboard

Faculty >

Student

Event

Settings

Logout

Add or Remove Faculty

Name :

Department :

Age :

Experience :

Phone :

User name :

E-mail :

Password :

Address :

Sex :

Click submit button to Add faculty

Click Delete button to Remove faculty

Nirmala

Search

Henry Jhon

henryjhon@gmail.com

Dashboard

Faculty >

Student >

Event >

Settings

Logout

View students details

Name

Department

Course

Id

Restrict

Jain

computer

AI

888444

Restrict





- Student pages

## 9.2 DB Design

- User table:

Field	Type	Description
User_id	int	Unique User ID(PK)
Name	Varchar(25)	Full Name
Email	Varchar(20)	User email(unique)(CK)
Password	Varchar(25)	Hashed password
Role	Char(20)	User role
Created at	timestamp	Account creation time

### Activity table:

Field	Type	Description
Activity id	int	Unique activity id(PK)
Title	Varchar(25)	Activity name
Description	text	Activity details
Category	Varchar(10)	Sports, arts, culture, others...
Created by	int	Faculty/Admin
Start date	datetime	start time
End date	datetime	end time

- Participation table:

Field	Type	Description
Participation_id	int	Unique participation id(PK)
Student_id	int	Unique student id(CK)
Activity id	int	Unique activity id(FK)
status	Boolean	Applied/approved/ completed
Certificate link	varchar	Path to certificate
points	int	Activity points awarded

# 10. User Roles

---


## 1. Admin:

- Manages the entire system.
- Adds/removes faculty members.
- Restricts or removes students if necessary.
- Creates, modifies, or deletes activities.
- Accesses and manages all student information.
- Monitors overall participation and generates reports.






## 2. Faculty

- Creates, modifies, or deletes activities.
- Monitors student participation in activities.
- Approves or rejects student participation requests.
- Issues participation certificates after verification.
- Tracks student performance based on activity points.



## 3. Student

- 
- 
- 
- Views and applies for campus activities.
  - Cancels participation if needed.
  - Downloads participation certificates.
  - Tracks personal activity points and semester-wise reports.

# Thank you

Any queries?

