Design Document - Campus Event Management System

# 1. Introduction

The Campus Event Management System is designed to streamline the process of organizing, managing, and participating in campus events. The system allows administrators to create and monitor events while providing students with the ability to register, attend, and give feedback. The objective of this project is to provide an efficient, user-friendly event management platform.

# 2. System Overview / Architecture

The system follows a client-server architecture with a web-based frontend, Flask-based backend, and a relational database for data persistence.

Major Components:

* • Frontend: HTML, CSS for user interfaces
* • Backend: Flask (Python) for business logic
* • Database: SQLite (or MySQL) for data storage

# 3. Modules / Components

The system consists of the following modules:

* • Admin Module – Allows event creation, management, and report generation.
* • Student Module – Allows students to register for events and provide feedback.
* • Authentication Module – Handles login and role-based access.
* • Reports Module – Generates insights such as event popularity, active students, and feedback averages.

# 4. Database Design

The database schema consists of the following main tables:

* • Users – Stores admin and student account details.
* • Events – Stores event information such as title, type, and capacity.
* • Registrations – Tracks student registrations for events.
* • Feedback – Stores ratings and comments from students.

An ER diagram can be added here to show relationships between tables.

# 5. Data Flow / Sequence

Typical data flow includes:

* • A student logs in → Views available events → Registers → Attends → Submits feedback.
* • An admin logs in → Creates events → Monitors registrations and attendance → Generates reports.

Data flow diagrams (DFD) can be included here for illustration.

# 6. UI Design

The system includes the following main screens:

* • Login Page – Provides authentication for admin and students.
* • Admin Portal – Event creation, management, and reports dashboard.
* • Student App – Event registration, attendance, and feedback submission.

# 7. Assumptions & Constraints

* • The system currently supports only seeded accounts for login.
* • The application runs on localhost and is not yet deployed on a server.
* • The system is limited to two roles: Admin and Student.

# 8. Technology Stack

* • Backend: Flask (Python)
* • Frontend: HTML, CSS
* • Database: SQLite / MySQL
* • Version Control: Git, GitHub