Project Synopsis

# Name / Title of the Project

CampusConnect: Smart QR-Based Event & Campus Ecosystem Portal

# Statement about the Problem

In colleges, students face difficulties during fest/event registrations. Currently, registrations are done via Google Forms followed by manual payment verification and ticket collection, leading to long queues, delays, and duplicate entries. Additionally, students lack a single platform for accessing club activities, training materials, placement updates, and hackathon details. Important emails about opportunities are often ignored, causing missed participation.

# Describe the Problem Statement

The absence of a centralized digital system creates inefficiencies in event management and campus coordination. Manual ticketing systems are prone to errors, fraud, and long waiting times. Clubs, departments, and placement cells struggle to effectively communicate their events, training sessions, and recruitment opportunities. Students are unable to stay updated with hackathons and technical events happening in and outside the college. Thus, a smart digital ecosystem is required to centralize all event, academic, and opportunity-related activities.

# Objective and Scope of the Project

Objectives: - To provide an online event registration and ticketing platform with QR-based entry validation. - To enable clubs and departments to manage and promote their events and resources. - To create a study and training hub with PYQs, company info, and placement updates. - To include a hackathon & opportunities section for student growth. - To implement ML modules for recommendations, fraud detection, and crowd prediction. Scope: The project will serve as a

one-stop solution for students, faculty, clubs, and placement cells. It covers event management, QR-ticketing, academic resources, placement notifications, hackathon updates, and student engagement features. It will significantly improve student experience and campus digitalization.

# Methodology

The project will be developed using the MERN stack( EJS). The methodology involves: - Requirement analysis and database design. - Development of authentication, event creation, and student registration modules. - Integration of payment gateway and QR-code generation. - Implementation of QR scanning for gate entry validation. - Building dashboards for students, faculty, and admins. - Integration of ML models for predictions and recommendations. - Final testing, debugging, and deployment. Diagrams like DFDs and Flowcharts will be used to represent the workflow of the system.

# Hardware & Software to be used

Hardware: - Laptop/PC with minimum 8GB RAM, 256GB storage - Smartphone for QR scanning Software: - Frontend: React.js, TailwindCSS/Material UI - Backend: Node.js, Express.js - Database: MongoDB Atlas - Authentication: JWT-based login/signup - Payment Gateway: Razorpay / Stripe / Paytm - QR Code: qrcode (npm), react-qr-reader - ML Models: Python (Flask/FastAPI) or TensorFlow.js - Notifications: NodeMailer, Firebase Messaging

# Future Work of this Project

- Integration of face-recognition along with QR for added security. - Advanced analytics dashboards for faculty and clubs. - Integration with third-party APIs for global hackathons and competitions. - Mobile app version of the system for better accessibility. - Gamification with rewards and leaderboards for student engagement.

# Schedule of the Project (Gantt Chart)

|  |  |
| --- | --- |
| **Week** | **Task** |
| Week 1 | Requirement analysis, finalize scope, tools setup |
| Week 2 | Database schema design, authentication module |
| Week 3 | Event creation module for faculty & clubs |
| Week 4 | Student registration + payment integration |
| Week 5 | QR code generation + ticket validation system |
| Week 6 | Gate entry system + Admin dashboard |
| Week 7 | ML integration (recommendations, fraud detection, crowd prediction) |
| Week 8 | Testing, bug fixing, final deployment, documentation |

1. ***References / Bibliography***

[1] MERN Stack Documentation: [https://www.mongodb.com/mern-stack](http://www.mongodb.com/mern-stack) [2] Razorpay Developer Docs: https://razorpay.com/docs/ [3] TensorFlow.js Documentation: [https://www.tensorflow.org/js](http://www.tensorflow.org/js)

[4] ReportLab Documentation for PDF generation: [https://www.reportlab.com/docs/](http://www.reportlab.com/docs/) [5] Firebase Cloud Messaging Docs: https://firebase.google.com/docs/cloud-messaging