MERN Projects

# 1. Convocation Seating Arrangement System

**Problem Statement:**

Create a web application for organizing and managing seating arrangements for a university's convocation ceremony. Students should be able to enter their registration number to view their assigned seat and section. Admins should be able to dynamically create and modify seating plans, upload seating data, and manage event logistics.

**Key Features:**

* Student login to view seat details.
* Admin dashboard to upload and manage seating data.
* Search functionality to locate students’ seating by registration number.
* Visual map of the seating arrangement for easy navigation.

**Technology Suggestions:**

* Frontend: HTML, CSS, JavaScript
* Backend: Node.js, Express
* Database: MongoDB or JSON files

**Expected Deliverables:**

* A web app where students can search their seat by entering their registration number.
* Admin panel for uploading and managing seating data (with CSV import support).
* A visual map showing the seating arrangement.

**Bonus Features:**

* Real-time seat status updates (e.g., if a student is re-assigned).
* 3D or augmented reality view of the seating plan.
* Notifications via email/SMS to students with seating details and event updates.

**Deployment:**

* Deploy on Heroku or Netlify with a URL for public access.
* Provide GitHub repository with documentation on setup, how to run the project locally, and example data for testing.

# 2. Automatic Attendance via QR Code/Facial Recognition

**Problem Statement:**

Develop a mobile or web application to automatically record classroom attendance. Professors can scan the room using a mobile camera for facial recognition, or generate QR codes for students to scan using their mobile devices upon entering the class.

**Key Features:**

* Facial recognition or QR code-based attendance.
* Real-time attendance logs accessible by students and professors.
* Admin dashboard for faculty to view and download attendance reports.
* Integration with university portals for data synchronization.

**Technology Suggestions:**

* Frontend: React Native (for mobile) / HTML, CSS, JS (for web)
* Backend: Node.js, Express, OpenCV (for facial recognition)
* Database: MongoDB or MySQL

**Expected Deliverables:**

* A working attendance system using either facial recognition or QR codes.
* Admin dashboard for professors to view, edit, and download attendance reports.
* Real-time attendance logs for students and faculty.

**Bonus Features:**

* Integrate both QR code and facial recognition for redundancy.
* Analytics on attendance trends and student performance correlation.
* Integration with third-party platforms for data export (e.g., Google Sheets).

**Deployment:**

* Deploy the web version on Heroku/Netlify, or publish the mobile app on Expo.
* Share GitHub repository with clear instructions for setting up facial recognition and QR functionality.

# 3. Campus Energy Monitoring Dashboard

**Problem Statement:**

Create a web-based dashboard that allows university management to monitor and optimize energy consumption across campus buildings. The system will track and display real-time statistics for electricity, water, and gas usage, helping the administration make informed decisions about energy-saving initiatives.

**Key Features:**

* Real-time data tracking for energy consumption.
* Visualization tools like graphs, charts, and heatmaps.
* Historical data analysis to detect usage patterns.
* Alerts for abnormal spikes in usage to prevent wastage.

**Technology Suggestions:**

* Frontend: HTML, CSS, JavaScript (D3.js for data visualization)
* Backend: Node.js, Express
* Database: MongoDB (for historical data storage)
* IoT: Integration with energy meters (via APIs or simulated data)

**Expected Deliverables:**

* A real-time energy tracking dashboard with data visualization (graphs, heatmaps).
* Admin interface to view historical data and track consumption trends.
* Alerts for abnormal spikes in energy usage.

**Bonus Features:**

* Implement predictive analytics to forecast future energy usage.
* Integrate IoT devices or APIs to pull real data from energy meters (if available).
* User-friendly mobile app for monitoring energy stats on the go.

**Deployment:**

* Deploy on Heroku or similar, making the dashboard accessible to judges via a public URL.
* Include GitHub repository with mock data and instructions to set up the system.

# 4. Intelligent Event Management System

**Problem Statement:**

Design a web application that uses AI to help universities schedule events efficiently. The system should recommend optimal venues, time slots, and manage resources like seating arrangements, projectors, and catering based on event requirements and availability.

**Key Features:**

* AI-powered recommendations for venue selection and resource allocation.
* Event registration for students and faculty.
* Event logistics management (seating, equipment, catering).
* Integration with campus calendars and facility schedules.

**Technology Suggestions:**

* Frontend: HTML, CSS, JavaScript
* Backend: Node.js, Express
* AI Tools: TensorFlow.js or a basic AI recommendation engine
* Database: MongoDB or MySQL

**Expected Deliverables:**

* AI-powered event scheduling system with venue and resource recommendations.
* Event registration system for students and faculty.
* Admin dashboard for event logistics management (seating, equipment, catering).

**Bonus Features:**

* Automated reminders and notifications for event participants.
* Integrate with calendar APIs (Google Calendar, Outlook) for syncing events.
* Use machine learning to suggest event timings based on past attendance patterns.

**Deployment:**

* Host the app on a public platform (Heroku, AWS, etc.).
* Share GitHub repo with code and data used for AI model training, and provide a user guide for testing the event management system.

# 5. Research Project Collaboration Hub

**Problem Statement:**

Build a portal where students and faculty can collaborate on research projects. The platform will provide project management tools such as task assignment, progress tracking, and document sharing. The goal is to foster interdisciplinary collaboration across departments.

**Key Features:**

* Research project creation and team collaboration.
* Task assignment, progress tracking, and real-time chat.
* Document sharing and version control.
* Faculty and student profiles showcasing research areas of interest.

**Technology Suggestions:**

* Frontend: React.js or Angular
* Backend: Node.js, Express
* Database: MongoDB or PostgreSQL

**Expected Deliverables:**

* A working collaboration portal where users can create projects, assign tasks, and track progress.
* Real-time chat for project discussions.
* Document sharing and version control.

**Bonus Features:**

* Integration with GitHub or Google Drive for document management.
* A recommendation engine to suggest collaborators based on research interests.
* Notifications for task deadlines and project updates.

**Deployment:**

* Deploy on a public platform like Heroku, with the app URL shared with judges.
* Provide a GitHub repository with setup instructions and sample user accounts for testing.

# 6. Smart Canteen Management System

**Problem Statement:**

Develop a smart canteen management system that allows students and faculty to pre-order meals from the university cafeteria. The system should use predictive analytics to optimize food preparation and reduce waste, while users can view nutritional information and customize orders.

**Key Features:**

* Online pre-ordering and payment for meals.
* Predictive analytics for inventory management.
* Customization of meal preferences and nutrition tracking.
* Integration with a university ID card payment system.

**Technology Suggestions:**

* Frontend: HTML, CSS, React.js
* Backend: Node.js, Express
* Database: MongoDB
* Analytics: Simple machine learning model for predicting orders

**Expected Deliverables:**

* Online ordering platform for students and staff.
* Predictive analytics to manage inventory and optimize food preparation.
* Nutritional info display and customizable meal preferences.

**Bonus Features:**

* Mobile-friendly design or mobile app version.
* Integration with the university ID card for payment processing.
* Real-time updates on order status (ready, in progress, etc.).

**Deployment:**

* Deploy on Heroku or Firebase for real-time updates.
* Provide GitHub repo with setup guide and instructions on testing order flows.

# 7. Skill Development and Internship Matching Platform

**Problem Statement:**

Design a platform where students can upload their resumes and personal profiles, and the system will match them with relevant internship or skill development opportunities. The platform should use algorithms to match students based on their academic background, interests, and career goals.

**Key Features:**

* Student profile and resume uploads.
* Matching algorithms that recommend internships or skill programs.
* Integration with company portals and university partnerships.
* Tracking of application status and reminders for deadlines.

**Technology Suggestions:**

* Frontend: HTML, CSS, React.js
* Backend: Node.js, Express
* Matching Algorithm: Basic AI algorithm for skill matching
* Database: MongoDB

**Expected Deliverables:**

* A working system for students to create profiles and upload resumes.
* Matching algorithm for internship/skill opportunities.
* Admin interface for companies to post internships.

**Bonus Features:**

* Integrate LinkedIn for automatic resume/profile updates.
* Real-time notifications for matched opportunities.
* Analytics on internship application status and performance.

**Deployment:**

* Host the app on Heroku or AWS with a public URL.
* Include GitHub repository with setup instructions and sample data for testing.

# 

# 8. Mental Health and Counseling App

**Problem Statement:**

Build a confidential web application for students to access mental health resources, schedule counseling sessions, and receive well-being tips. The app could provide personalized content based on AI that monitors student interactions to detect stress or emotional states.

**Key Features:**

* Student login with secure, confidential access to mental health resources.
* AI-based content recommendations for mental well-being.
* Scheduling system for booking counseling sessions.
* Mood tracker and chatbot for mental health support.

**Technology Suggestions:**

* Frontend: HTML, CSS, React.js
* Backend: Node.js, Express
* AI Tools: TensorFlow.js for mood analysis
* Database: MongoDB

**Expected Deliverables:**

* Secure login for students with access to mental health resources.
* Counseling appointment booking system.
* AI-based system for personalized well-being recommendations.

**Bonus Features:**

* Integration with a chatbot for instant mental health support.
* Notifications/reminders for scheduled counseling sessions.
* Mood tracker feature to assess emotional well-being over time.

**Deployment:**

* Deploy on a platform like Firebase for real-time data syncing and security.
* Provide GitHub repository with AI model setup and instructions for testing.

# 9. Interactive Learning Pathway System

**Problem Statement:**

Create a system that helps students design and navigate their academic journey based on their interests, completed courses, and career goals. The system will offer personalized course recommendations, elective suggestions, and show career trajectories for different academic pathways.

**Key Features:**

* Student profile and academic history tracking.
* Personalized course recommendations based on academic performance.
* Visual representation of academic paths and career outcomes.
* Alerts for critical registration deadlines and course requirements.

**Technology Suggestions:**

* Frontend: HTML, CSS, JavaScript
* Backend: Node.js, Express
* Database: MongoDB or MySQL

**Expected Deliverables:**

* Student profile and academic history tracking.
* Personalized course recommendations.
* Visual pathway representation showing academic progression and career outcomes.

**Bonus Features:**

* Integration with job market data to suggest courses relevant to in-demand skills.
* Analytics on the student's progress toward their career goals.
* Notifications for critical deadlines (course registrations, exams).

**Deployment:**

* Deploy on Heroku with public access for testing.
* Share GitHub repo with mock data for academic history and career goal matching.

# 10. Campus Accessibility Map for Disabled Students

**Problem Statement:**

Design a web application that provides real-time navigation for disabled students across the university campus. The map will highlight accessible routes, entrances, and facilities like ramps, elevators, and accessible restrooms.

**Key Features:**

* Interactive map with real-time navigation for accessibility routes.
* Information on accessible facilities (ramps, restrooms, elevators).
* Real-time alerts for temporary closures or obstacles (construction).
* Option to report accessibility issues.

**Technology Suggestions:**

* Frontend: HTML, CSS, JavaScript (Leaflet.js or Google Maps API)
* Backend: Node.js, Express
* Database: MongoDB

**Expected Deliverables:**

* Interactive map showing accessible routes, ramps, elevators, and restrooms.
* Real-time navigation for students with mobility issues.
* Option for students to report accessibility problems (e.g., broken elevators).

**Bonus Features:**

* Use of AR (Augmented Reality) for better accessibility guidance.
* Integration with GPS for real-time navigation on mobile devices.
* Alerts for temporary closures or obstacles on campus.

**Deployment:**

* Host the app on a public platform (e.g., Heroku, Netlify).
* Provide GitHub repo with instructions and sample data for testing navigation features.