CampusSink - Project Charter Document.

Purpose:

The objective of this project is to create a centralized web-based system that displays all the student assignments, seminars, guest lectures, and project reviews organized by branch and year. Teachers will have full control over adding, editing, and managing assignments and deadlines. Students will only be able to view tasks, seminars, and projects relevant to their year and branch.

Objectives:

- **Role-based Access**: Provide different levels of access for teachers and students.
    - **Teachers**: Can view and manage all assignments, seminars, and projects. Can edit deadlines and track submissions
    - **Students**: Can view assignments, seminars, and projects for their branch and year.
- **Assignment Management**: Teachers can upload, edit, and delete assignments, project reviews, and seminars.
- **Deadlines**: Display upcoming deadlines for assignments, seminars, and projects in a user-friendly format and the user will be notified of the deadlines.
- **Categorization**: Organize the assignments based on the student's branch and academic year.

Pestle Analysis:

Factor	Impact on the Exam Evaluation System

Political	No impact

Economic	The cost of developing, maintaining, and hosting the portal can impact project sustainability.
Social	Some teachers and students may resist the move to digital platforms, particularly in more traditional or under-resourced institutions. Training and support will be essential..
Technological	Using technologies like React.js, Node.js, PostgreSQL, and cloud hosting (AWS) will allow for scalability and flexibility. With sensitive data involved, strong encryption and secure access control measures are necessary to protect against data breaches and cyberattacks.
Environmental	A digital platform for assignments will significantly reduce the use of paper, which contributes positively to the environment by minimizing the carbon footprint associated with traditional education systems.
Legal	Teachers may want to protect their content (e.g., assignments, materials) from being reused or plagiarized. Clear policies on intellectual property will be necessary.

Opportunity based / Demand based:

The **portal** is **opportunity-based**, utilizing advancements in technology to streamline assignment management, improve accessibility, and replace inefficient paper-based systems, offering modern solutions to educational institutions.

Business / High-Level Features:

- **Secure Login:** Role-based access for teachers and students via secure authentication.
- **Task Management:** Teachers can create, edit, and manage assignments, seminars, and project reviews.
- **Student Dashboard:** Students view only relevant tasks filtered by their branch and year.
- **Deadline Alerts:** Automatic notifications for upcoming submission deadlines.
- **Department-wise Support:** Handles multiple departments and academic years simultaneously.
- **Task History:** Option to archive or download past records for audits or review.
- **Data Security:** Ensures encryption and restricted access to protect academic data.
- **Paperless Workflow:** Reduces manual work and promotes eco-friendly practices.
- **Cloud Hosted:** Scalable deployment on platforms like AWS or Vercel.

Technical Requirements:

- **User Authentication**
  - UI for secure login and role-based access (teacher, student, admin)
  - API routes for user registration, login, password encryption, and session handling
  - OAuth integration (optional) for institutional email-based login

---

- **Assignment & Seminar Management**
  - UI forms for teachers to create, edit, and delete assignments, seminars, and project reviews
  - Backend APIs for CRUD operations on academic tasks with validation
  - Database schema to link tasks with department, year, and deadlines

---

- **Student Dashboard & Task Filtering**
  - UI for displaying personalized task list based on user's branch and academic year
  - API endpoints to fetch relevant tasks and deadlines dynamically
  - Pagination, sorting, and filtering features for smooth task navigation

---

- **Deadline Notifications**
  - Scheduled backend job or notification service to send reminders for upcoming deadlines

○ Option to display countdown timers or alert banners in student dashboard

---

● **File Upload & Storage**

○ Teachers can upload PDFs or DOCs as task resources

○ File storage handled through cloud storage (e.g., AWS S3 or Vercel storage)

○ Secure download links accessible only to authorized students

---

● **Data Security**

○ Implementation of authentication tokens (e.g., JWT) to protect user sessions

○ Role-based access controls at both frontend and backend

○ HTTPS encryption for secure data transmission

● **Database Design**

○ PostgreSQL database schema with relationships between users, departments, courses, and tasks

○ Indexing and optimization for fast queries on deadlines and submissions

---

● **Hosting & Deployment**

○ Deployment on AWS, Vercel, or Render with CI/CD pipeline (optional)

○ Environment-based configuration for testing, staging, and production

Technological Requirements:

● **Frontend**

○ React.js

○ CSS3 / Bootstrap (or Tailwind CSS, if used)

● **Backend**

○ Node.js

○ Express.js

○ Firebase Authentication (for email notifications/reminders)

● **Database**

○ PostgreSQL (for structured academic data storage)

● **Authentication & Security**

○ JWT (JSON Web Tokens) for secure session management

○ OAuth (for optional institutional login integration)

● **Hosting & Deployment**

○ Vercel / Render / AWS (for frontend and backend deployment)

○ Environment configuration via .env files

● **Cloud Storage (Optional)**

○ AWS S3 or similar (for storing assignment-related files)

● **Tools**

○ Visual Studio Code – Code Editor

○ Git & GitHub – Version Control

○ Postman – API Testing

○ Figma – UI/UX Design

Risk Analysis:

Risk Category	Description	Impact	Mitigation Strategy
1. Technical Risk	The scope of the project may change due to changing requirements or technical challenges.	High	Clearly define and freeze requirements in the planning phase.
2. Cost Risk	The project may face unexpected cost overruns due to unforeseen design, development, or operational needs.	High	- Establish detailed cost estimates with contingency buffers.
3. Schedule Risk	There may be delays in activities such as regulatory approval, detailed design, and integration of features, which could push back project timelines.	Medium	- Use a detailed project plan with adequate buffer time for each task. - Regular progress tracking and early identification of potential bottlenecks.
4. Budget Risk	- The project budget may be underestimated or insufficient, leading to financial strain. - Scope creep could add unanticipated expenses.	High	- Establish a realistic budget with allowances for scope changes. - Monitor scope closely to prevent scope creep and use change control processes to manage budgetary impact.
5. Business Risk	Delays in management approvals or inputs from key stakeholders (teachers, institutions) may result in the loss of market timing, affecting project profitability.	Medium	- Ensure timely communication with stakeholders. - Set firm deadlines for decision-making, approvals, and input from business entities.
6. Infrastructure	- Lack of adequate	Medium	- Plan infrastructure needs early and

Risk	infrastructure, such as cloud		work closely with suppliers (cloud

	services or hosting platforms, could delay project activities such as system setup or operations.		providers). - Ensure proper allocation of resources for setup and operation phases.
7. Operational Risk	Improper execution of deployment, or insufficient resources during critical stages such as testing and launch, could result in system failures or poor performance.	High	- Implement clear operational guidelines and assign adequate resources for critical phases. - Conduct thorough testing before launch and assign a dedicated support team.
8. Resource Risk	- Availability of key personnel, such as developers or testers, may become a challenge, resulting in delays.	Medium	- Secure necessary resources early and maintain backup plans in case of staffing shortages.
9. Supplier Risk	Third-party suppliers, such as cloud service providers or database service providers, may fail to deliver expected results or meet deadlines.	Medium	Create clear contracts with performance metrics and penalties for delays or failure to deliver.

Stakeholders:

End User	Students, Teachers (Administrators of assignments).
Sponsors	Educational institutions, academic boards
Suppliers	Cloud service providers (AWS, Azure), database services
Frontend developer	Dhiraj Rathod
Testers	Shravani Sakore

Design	Diksha Tiwari
Backend developer	Shivshankar Ghyar
Consultant	Jayesh Raut
Project Lead	Dr.Anuradha Pawar

Project ROI (Return on Investment):

- **Revenue Models**:
  - Subscription-based access for institutions to manage and monitor assignments.
  - Pay-per-student or pay-per-year model for accessing assignment management.
- **ROI Timeline**:
  - Break-even expected within 1-2 years after launching, with adoption in multiple educational institutions.

Resources & Technologies Needed:

Human Resources:

- **Frontend Developers**: 2 (React.js,CSS).
- **Backend Developers**: 2 (Node.js, Express.js, PostgreSQL).
- **UI/UX Designers**: To design user interfaces that are easy for students and teachers to navigate.
- **Security Experts**: To ensure compliance with data privacy laws.

Technologies & Tools:

- **Cloud Computing**: AWS, Azure for hosting.
- **Database**: PostgreSQL for storing structured data like assignments and user profiles

Timeline & Milestones:

	Tasks	Timeline
Phase 1: Planning	Market research, requirements gathering	1 week
Phase 2: Design	UI/UX design, database schema, architecture	3 week
Phase 3: Development	Frontend & backend coding, API integration	3 week
Phase 4: Testing	Testing UI, security audits, bug fixing	½ week
Phase 5: Deployment	Deploy to AWS/Vercel, final roll-out	½ week

RACI CHART:

Task/Deliverable	Shivshankar	Shravani	Dhiraj	Diksha	Dr. Anuradha Pawar	Jayesh Rout
Backend Development (APIs, Logic)	R	A	C	I	I	I
Frontend Development (UI Components)	C	R	I	A	I	I
Database Design & Integration	I	A	R	I	C	I
UI/UX Design (Wireframes, Layout)	I	A	C	R	I	I
Feature Testing & Bug Fixes	R	C	A	I	I	I
Final Integration & Optimization	A	I	R	C	I	I

- **R (Responsible)**: Person who performs the work.
- **A (Accountable)**: Person accountable for task completion.
- **C (Consulted)**: Person whose input is needed before proceeding.
- **I (Informed)**: A person who must be kept updated.