**Project Proposal**

**Project Name**

**"Grade A"  
An e-learning platform that delivers engaging, interactive, and accessible educational content.**

**1. Project Idea**

**The e-learning management system is a web-based platform designed to facilitate online education. It enables:**

* **Instructors to create and manage courses.**
* **Students to enroll and access materials.**
* **Administrators to oversee the system's operations.**

**The platform provides a seamless and engaging learning experience while addressing challenges like accessibility, scalability, and interactivity.**

**2. Problem Statement**

**Education systems face numerous challenges during digital transformation:**

1. **Limited accessibility: Remote learners often lack access to quality education.**
2. **Progress tracking difficulties: It’s challenging to monitor students’ engagement in a virtual setting.**
3. **Inefficient tools: Instructors struggle with inadequate tools for creating, managing, and assessing courses.**
4. **Lack of integration: Centralized systems that combine content delivery, assignments, and performance tracking are missing.**

**Solution:  
The proposed e-learning system addresses these issues with a user-friendly, collaborative platform.**

**3. User Personas**

**Student Persona**

**Description: College students seeking online courses to enhance their education.  
Goals:**

* **Enroll in courses with approved eligibility.**
* **Access materials easily and track grades and progress.**
* **Receive timely feedback on assignments and tests.  
  Pain Points:**
* **Limited interactivity with course options.**
* **Lack of clear progress tracking.**

**Instructor Persona**

**Description: Educators or professionals creating and managing online courses.  
Goals:**

* **Efficiently upload and organize course materials.**
* **Engage with students via assignments and forums.**
* **Track and assess student performance.  
  Pain Points:**
* **Managing large virtual classes.**
* **Difficulty assessing engagement and progress.**

**Admin Persona**

**Description: System administrators overseeing users, courses, and platform configurations.  
Goals:**

* **Ensure the platform operates smoothly.**
* **Monitor user activity and resolve issues.**
* **Maintain data security and integrity.  
  Pain Points:**
* **Managing large datasets.**
* **Ensuring reliability under heavy traffic.**

**4. High-Level Functional Requirements**

**1. User Authentication and Role Management**

* **Secure login system with session management.**
* **Role-based access control for students, instructors, and admins.**

**2. Course Management**

* **Instructors: Create, edit, delete courses, and upload materials.**
* **Students: Browse, enroll, and access course materials post-enrollment.**
* **Admins: Approve, manage, or deactivate courses.**

**3. Content Delivery**

* **Instructors: Upload various materials (PDFs, videos, quizzes, etc.).**
* **Students: Download materials and submit assignments.**

**4. Progress Tracking**

* **Students: View reports showing completed modules and grades.**
* **Instructors: Monitor performance through visual analytics like grade trends.**

**5. Communication Tools (Optional)**

* **Message boards for discussions.**
* **Notifications for updates, deadlines, and grades.**

**5. Using Supabase for the Prototype**

**Reason for Using Supabase as Backend**

**For the initial development, Supabase will serve as the backend instead of Django. Supabase simplifies backend development by offering built-in authentication, database access, and APIs, enabling faster prototyping and iteration.**

**Advantages of Using Supabase as Backend Now**

1. **Rapid Prototyping: With minimal setup, Supabase allows quick development and deployment of backend functionality, such as authentication and role management.**
2. **Built-in Features: Supabase provides integrated real-time capabilities, storage, and API generation, eliminating the need for custom backend implementation during the early stages.**
3. **Focus on Frontend: Using Supabase as the backend lets the team concentrate on building and refining the frontend interface.**
4. **Scalable Infrastructure: Supabase scales automatically to handle growing demands as the project expands.**

**Final Architecture Plan: Supabase as Database with Django Backend**

**At the end of the project, Django will take over as the backend, enabling advanced workflows, custom logic, and enhanced security. Supabase will continue to serve as the database due to its high scalability and robust PostgreSQL foundation.**

**Advantages of Supabase as a Final Database**

1. **PostgreSQL-Based: Supabase uses PostgreSQL, a reliable and well-established relational database system, ensuring data integrity and scalability.**
2. **Real-Time Updates: Supabase's real-time capabilities allow instant synchronization of grades, assignments, and progress tracking across the platform.**
3. **Secure Data Storage: Built-in encryption and access control ensure secure storage and handling of sensitive user data.**
4. **Global Accessibility: As a cloud-based database, Supabase ensures consistent performance regardless of geographic location.**

**6. High-Level Non-Functional Requirements**

**Performance**

* **The platform must handle up to 100 concurrent users with minimal latency.**
* **Average response time should remain under 2 seconds during normal load.**

**Security**

* **Authentication and data storage will follow industry-standard encryption.**
* **HTTPS will be enforced for all communications to protect data in transit.**

**Usability**

* **The platform will feature a responsive, intuitive interface accessible on desktops, tablets, and smartphones.**
* **Users will navigate the system easily with clear menus, dashboards, and tooltips.**

**Maintainability**

* **Modular code design will simplify updates and debugging.**
* **Comprehensive documentation will ensure smooth onboarding of new developers.**

**7. Architecture Diagram**

**Architecture Overview:  
The architecture diagram demonstrates the flow of requests from the front end to the back end and database. Here's how the system functions:**

1. **User Requests:**
   * **Requests such as login, viewing courses, or submitting assignments are sent via HTTPS from the frontend to the Supabase backend.**
2. **Supabase Backend:**
   * **Supabase processes the requests, enforces business logic, and interacts with the PostgreSQL database to fetch or update data.**
3. **Database Interaction:**
   * **Supabase’s database layer securely stores and retrieves data, ensuring real-time synchronization for dynamic updates (e.g., grades or course progress).**
4. **Response Flow:**
   * **Supabase sends the processed response back to the frontend, ensuring seamless interaction for the user.**

**.**

**A diagram of a computer

Description automatically generated**

**3. Wireframes**

Wireframes are simple, visual blueprints of how the user interface will look. They help in understanding user navigation and interactions.

**Wireframe Components:**

1. **Home Page**:
   * Navbar with login/logout, dashboard link.
   * Course categories or featured courses.
2. **Login Page**:
   * Input fields for email and password.
   * Login button and "Forgot password?" link.
3. **Instructor Dashboard**:
   * Create a course.
   * View and manage existing courses.
   * Upload content.
4. **Student Dashboard**:
   * View enrolled courses.
   * Access course materials.
   * Track progress.
5. **Admin Dashboard**:
   * User management (approve instructors, manage students).
   * View platform reports.



