Student Portal Requirements Document

1. Functional Requirements

1.1 User Registration and Authentication

Students can create accounts by providing necessary personal details (e.g., name, email, password).

Secure authentication mechanisms (e.g., email verification, multi-factor authentication)

Students can reset their passwords if forgotten.

1.2 Course Management

Students can browse available courses, view details, and register for courses.

Ability to track course progress (e.g., grades, assignments, exams).

Admins can add, remove, or modify courses and set prerequisites.

1.3 Scheduling

Students can view course schedules, including lecture timings and locations.

Automatic conflict checking for course schedules to prevent overlap.

Option to add courses to a personal calendar.

1.4 Grade and Assignment Management

Students can view grades, feedback, and download assignments.

Instructors can upload grades and assignments.

Ability to track deadlines for assignments and exams.

1.5 Notifications and Alerts

Email/SMS notifications for course updates, deadlines, and announcements.

Alert students about changes in schedules, course availability, or upcoming events.

1.6 Profile Management

Students can update personal details (e.g., name, email, phone number).

View a history of registered courses and academic progress.

1.7 Role-Based Access Control

Different user roles (e.g., Admin, Instructor, Student) with defined permissions for course creation, grade viewing, etc.

1.8 Search and Filtering

Students can search for courses by category, schedule, instructor, or topic.

Ability to filter courses based on availability, prerequisites, and level.

1.9 Payment Integration (Optional)

Integration of a payment gateway or course registration fees.

Payment history tracking for students.

1.10 Data Export and Reporting

Admins and instructors can generate reports on course registrations, attendance, grades, etc.

Students can export academic history in formats like CSV or PDF.

2. Non-Functional Requirements

2.1 Performance

The system should handle at least 10,000 concurrent users with response times < 3 seconds.

Page load times should be under 2 seconds.

2.2 Scalability

The system must scale vertically (e.g., better hardware) and horizontally (e.g., load balancing).

It should handle traffic spikes, especially during peak periods such as registration.

2.3 Availability

The system should have 99.9% uptime (excluding scheduled maintenance).

Redundancy for critical components to ensure high availability.

2.4 Security

All sensitive data (e.g., passwords, grades) should be encrypted using SSL/TLS and AES.

User credentials should be securely stored with hashing (e.g., bcrypt).

Protection against common web vulnerabilities (SQL injection, XSS).

Regular security updates and vulnerability assessments.

2.5 Usability

Intuitive and user-friendly interface.

Mobile-responsive design for access on smartphones and tablets.

Accessibility features for users with disabilities (e.g., WCAG 2.1 compliance).

2.6 Backup and Recovery

Automatic backups of critical data every 24 hours.

Quick data recovery with minimal data loss in case of failure.

2.7 Compliance

The system must comply with relevant data protection regulations such as GDPR, FERPA.

Regular audits to ensure ongoing compliance.

2.8 Localization

Support for multiple languages (e.g., English, Spanish, French) based on the user demographic.

Time zone-aware scheduling and notifications.

2.9 Integration

Integration with third-party systems (e.g., payment gateways, academic tools, APIs).

Data export/import capabilities for integration with external educational systems.

3. Constraints

3.1 Budget

The system must be developed within a specified budget, limiting the scope of certain features (e.g., third-party integrations or advanced functionalities).

3.2 Timeline

The project should be completed within a predefined timeline (e.g., 6-12 months for an MVP).

Some features may be prioritized for initial releases, with others added in later phases.

3.3 Technical Stack

The system will be built using specific technologies such as React/Angular for frontend, Node.js/Django for backend, and MySQL/PostgreSQL for the database.

The hosting platform may be limited to certain providers (e.g., AWS, Azure, GCP) depending on project needs.

3.4 Compliance with Legal Requirements

The system must adhere to applicable regulations such as GDPR in Europe and FERPA in the US.

Data storage and handling must comply with all relevant privacy and security regulations.

3.5 Technology Limitations

The system must be compatible with popular browsers (Chrome, Firefox, Safari, Edge) and operating systems (Windows, macOS, Linux).

Legacy systems (e.g., older database versions) may limit integration with newer tools.

3.6 User Capacity

The system must support a defined number of concurrent users during peak traffic periods.

Hardware limitations may impact initial scalability.

3.7 User Adoption

The platform must be easy to use for students with varying levels of technical proficiency.

The interface should include onboarding tutorials and guides to help users get started.