# Key Features

**Student Features:**

* **Registration and Authentication**: Sign up, login, password recovery.
* **Profile Management**: Profile view, edit, document upload.
* **Course Enrolment**: Course listing, course details, enrolment.
* **Report Submission**: Report upload, submission tracking.
* **Notifications**: Notification alerts, announcements view.

**Admin Features:**

* **Student Management**: Profile view, approval/rejection of documents.
* **Course Management**: Course creation, editing, assignment of instructors.
* **Report Management**: Report view, grading, feedback provision.
* **Notifications**: Create and send announcements, notification management.
* **Analytics and Reporting**: Report generation, data visualization, statistical analysis.

**Additional Features:**

* **Security**: JWT-based authentication, role-based access control.
* **Responsive Design**: Mobile-friendly interface.
* **Audit Trail**: Action logging, audit reports.

# Functionalities:

1. **User Management**:
   * Student registration and authentication
   * Admin authentication and role-based access control
2. **Profile Management**:
   * Students can view and edit their profiles
   * Upload and manage necessary documents (e.g., ID proofs, certificates)
3. **Course Management**:
   * Display available courses
   * Course enrollment by students
   * Manage course creation and assignment by admin
4. **Report Management**:
   * Submission of assignments and reports by students
   * Review, grade, and provide feedback on submissions by admin
5. **Notification System**:
   * Notify students of important updates, deadlines, and announcements
   * Admin can send announcements to students
6. **Analytics and Reporting**:
   * Generate reports on student performance and course enrollments
   * Admin can view enrollment statistics and submission statuses

# Stakeholders and Their Roles

1. **Students**:
   * Register and create profiles
   * Enroll in courses
   * Submit assignments and reports
   * Receive notifications and updates
2. **Institute Administrators**:
   * Manage student profiles
   * Approve or reject documents
   * Create and manage courses
   * Review and grade reports
   * Send announcements and notifications
3. **Instructors** (if applicable):
   * Create and manage course content
   * Review and grade student submissions
   * Provide feedback on reports

# Project Plan with Milestones and Timelines

1. **Planning (1 week)**:
   * Define scope and purpose
   * Identify stakeholders
   * Develop project plan
2. **Requirements Analysis (2 weeks)**:
   * Gather requirements from stakeholders
   * Document and validate requirements
3. **Design (3 weeks)**:
   * System architecture design
   * Database schema design
   * UI/UX design
   * Detailed design documentation
4. **Implementation (8 weeks)**:
   * Backend development (4 weeks)
     + Set up Spring Boot project
     + Implement entities, repositories, services, and controllers
     + Configure security and authentication
   * Frontend development (4 weeks)
     + Develop UI components
     + Integrate with backend APIs
5. **Testing (2 weeks)**:
   * Unit testing
   * Integration testing
   * System testing
   * User acceptance testing
6. **Deployment (1 week)**:
   * Deploy to production environment
   * Set up CI/CD pipeline
   * Monitor deployment
7. **Maintenance (ongoing)**:
   * Bug fixes
   * Updates and new features
   * Performance monitoring

## Success Criteria:

* Successful user registration and authentication.
* Efficient course enrollment and management.
* Timely submission and grading of reports.
* Effective communication through notifications.
* Accurate generation of analytics and reports.

## Risk Management:

* **Identify Risks**: Potential technical issues, scope creep, delays in timelines.
* **Mitigation Strategies**: Regular progress reviews, clear communication channels, contingency plans.

## Resource Allocation:

* **Team Members**: Define roles for developers, testers, designers, project managers.
* **Tools and Technologies**: Java (Spring Boot) for backend, a frontend framework (React, Angular, or Vue), database (MySQL, PostgreSQL), CI/CD tools, testing tools.

## Budget and Cost Estimates:

* **Budget Planning**: Estimate costs for development, testing, deployment, and maintenance.
* **Cost Control**: Monitor expenses to stay within the budget.

## Communication Plan:

* **Internal Communication**: Regular team meetings, progress updates, issue tracking.
* **External Communication**: Regular updates to stakeholders, user feedback mechanisms.

## Additional Considerations

* **Legal and Compliance**: Ensure the system complies with relevant data protection regulations (e.g., GDPR).
* **Training and Documentation**: Plan for user training sessions and create comprehensive user manuals and technical documentation.

# Tools and Technology

## Front-End Development

These tools and technologies will help you build the user interface and ensure it is responsive and user-friendly.

* **HTML5:**
  + **Purpose:** Structure the content of your web pages.
  + **Description:** Basic markup language for creating web pages and web applications.
* **CSS3:**
  + **Purpose:** Style and layout your web pages.
  + **Description:** Used to define the look and feel of your web pages, including colors, fonts, and layout.
* **JavaScript:**
  + **Purpose:** Add interactivity to your web pages.
  + **Description:** A scripting language that allows you to create dynamically updating content, control multimedia, animate images, and much more.
* **Bootstrap:**
  + **Purpose:** Simplify the design and layout process.
  + **Description:** A front-end framework that includes pre-designed HTML and CSS templates for forms, buttons, navigation, and other interface components.

## Back-End Development

These tools and technologies will help you manage the server, database, and business logic using Java.

* **Java:**
  + **Purpose:** General-purpose programming language.
  + **Description:** A widely-used language for building server-side applications. Known for its portability, performance, and robustness.
* **Spring Boot:**
  + **Purpose:** Framework for building Java applications.
  + **Description:** Simplifies the development of Java applications by providing a comprehensive infrastructure. Great for creating RESTful services.
* **Hibernate:**
  + **Purpose:** ORM (Object-Relational Mapping) tool.
  + **Description:** Facilitates the mapping of Java objects to database tables, allowing you to interact with the database using Java objects.
* **MySQL:**
  + **Purpose:** Database management.
  + **Description:** A widely-used relational database management system. You can use it to store and manage student information and other data.

## Development Tools

These tools will assist in coding, version control, and project management.

* **IntelliJ IDEA:**
  + **Purpose:** Integrated development environment (IDE).
  + **Description:** A powerful IDE for Java development, offering extensive features for coding, debugging, and testing.
* **Git:**
  + **Purpose:** Version control.
  + **Description:** A distributed version control system to track changes in your source code during software development.
* **GitHub:**
  + **Purpose:** Code hosting platform.
  + **Description:** A web-based platform used for version control and collaborative development, allowing you to store and manage your code.

## Deployment and Hosting

These tools and platforms will help you deploy and host your application.

* **Heroku:**
  + **Purpose:** Cloud platform for deployment.
  + **Description:** A cloud platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud.
* **AWS (Amazon Web Services):**
  + **Purpose:** Cloud hosting.
  + **Description:** Provides reliable, scalable, and inexpensive cloud computing services. Services like EC2 and RDS can be used to host your Java application and MySQL database.

## Authentication and Security

These tools and services will help you manage user authentication and ensure security.

* **Spring Security:**
  + **Purpose:** Security framework for Java applications.
  + **Description:** Provides comprehensive security services for Java applications, including authentication, authorization, and protection against common attacks.
* **bcrypt:**
  + **Purpose:** Password hashing.
  + **Description:** A password-hashing function to securely store passwords.

## APIs and Integration

These tools and technologies will help you integrate various services and functionalities.

* **RESTful APIs:**
  + **Purpose:** Enable communication between client and server.
  + **Description:** Use RESTful principles to design and implement APIs that allow different parts of your application to communicate with each other.

## Key Features Description

1. **User Registration and Authentication:**
   * **Description:** Allows new students to create an account and securely log in. Use HTML forms for input, styled with CSS, and JavaScript for validation. Implement authentication using Spring Security for secure login.
2. **Profile Management:**
   * **Description:** Allows students to view and update their personal information. Create a user-friendly form with HTML/CSS and use JavaScript to handle form submissions and updates.
3. **Document Upload:**
   * **Description:** Enables students to upload necessary documents for verification. Use an HTML file input element, styled with CSS, and handle file uploads with Java and Spring Boot.
4. **Dashboard:**
   * **Description:** Provides an overview of the student's profile, uploaded documents, and notifications. Build a dynamic dashboard using HTML, CSS, and JavaScript to fetch and display data.
5. **Notifications:**
   * **Description:** Allows the administration to send notifications to students. Implement a notification system using Java and Spring Boot to push updates and JavaScript to display them in the user's dashboard.
6. **Data Reporting:**
   * **Description:** Generate reports based on student data. Use Hibernate with MySQL to query and aggregate data, and Spring Boot to create endpoints that serve the reports to the front-end.