

ACEGRADE

A Mini Project IT23502- Web Programming

Submitted By:

SRIDHAR NAVEEN .P.A	-2023506067
HARISH .P	-2023506116
HARSHAD .S	-2023506123
NARENDRAN .S	-2023506125

UNDER THE GUIDANCE OF : Dr.S.K.Lavanya

DEPARTMENT OF INFORMATION TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY

CHENNAI -600 044

July / November 2025

Table of Contents

1. Abstract	3
2. Objectives	4
3. Technologies used	5
4. Purpose	6
5. Application	7
6. Tech Stack	8
7. Repository Structure	9
8. API Overview	10
9. Project Output	10
10. Project Overview	15

ABSTRACT

The *AceGrade* web application is a comprehensive academic resource management platform designed to streamline access to university learning materials and enhance student engagement. The system serves as a centralized digital hub that provides students with essential academic resources such as lecture notes, study materials, previous year question papers, and a CGPA calculator to track academic performance. It also includes distinct access modules for both students and administrators, ensuring efficient content management and secure data operations.

Developed using modern Java web technologies, *AceGrade* integrates a robust backend built with Java, Maven, and PostgreSQL for database management, while the frontend is designed using HTML, CSS, and JavaScript for an intuitive user interface. The admin module allows authorized users to upload and manage academic materials, monitor student activity, and update academic content dynamically. The student module provides easy navigation to academic resources, lecture notes, and question papers, along with an interactive CGPA calculator to assist in academic performance evaluation.

The system focuses on promoting a digital academic environment that minimizes dependency on physical documents and facilitates knowledge sharing. It ensures scalability, data consistency, and secure authentication mechanisms for both user types. By integrating essential academic functionalities in one platform, *AceGrade* aims to enhance learning efficiency, accessibility, and institutional resource management — ultimately fostering a more organized and digitally empowered educational ecosystem.

OBJECTIVES

1. **To develop a user-friendly academic resource portal:**

The main objective is to design an intuitive and responsive web interface using HTML and CSS that allows students and administrators to easily navigate through different modules such as notes, lectures, question papers, and CGPA calculator without any technical difficulty.

2. **To implement a robust backend system using Spring Boot:**

The backend is built using Spring Boot to provide a stable, secure, and efficient framework for handling data flow between the frontend and the database. This ensures smooth interaction, modular structure, and simplified management of server-side logic.

3. **To provide secure user authentication and role-based access:**

The system includes login and signup modules with Spring Security integration, enabling distinct access levels for students and administrators. This ensures that only authorized users can upload, view, or modify academic resources.

4. **To enable centralized management of academic resources:**

The platform allows administrators to upload and manage university notes, lecture materials, and previous year question papers in one place. Students can easily access and download these resources whenever required, promoting digital learning.

5. **To include a CGPA calculator for performance tracking:**

The system integrates a simple yet effective CGPA calculator feature where students can input their grades and credits to calculate their cumulative GPA, helping them evaluate and track academic performance efficiently.

TECHNOLOGIES USED

1. HTML (HyperText Markup Language)

HTML is the core technology used for building the structure of the web pages in *AceGrade*. It defines the layout of the website, including pages such as the login page, dashboard, study resources, question papers, and CGPA calculator. Each page uses semantic HTML elements to organize content clearly, making it easy to navigate and accessible to users. HTML ensures that all text, links, images, and forms are properly displayed across browsers.

2. CSS (Cascading Style Sheets)

CSS is used to control the design, style, and responsiveness of the *AceGrade* web interface. It enhances the visual appearance of the website through color schemes, typography, spacing, and animations. CSS files like *styles.css* and *study-resources.css* help maintain consistent styling across multiple pages. It also ensures that the application layout adapts smoothly to different screen sizes, improving the user experience for both desktop and mobile users.

3. Java (Core Programming Language)

Java is the primary backend programming language used to build the logic and functionality of the *AceGrade* system. It handles user authentication, file uploads, CGPA calculations, and server-side communication. Java ensures platform independence, scalability, and security, making it ideal for academic systems where reliability and performance are critical.

4. Spring Boot (Backend Framework)

Spring Boot is the main backend framework used to develop and run the *AceGrade* application. It simplifies Java web development by providing built-in configurations, dependency management, and an embedded server. Spring Boot handles HTTP requests, connects the frontend with backend services, and executes application logic. It is also used for implementing RESTful APIs to handle user operations like login,

registration, and resource management. The framework's simplicity and modular design make deployment and testing faster.

5. PostgreSQL (Database Management System)

PostgreSQL is used as the database to store all the academic data and user information. It maintains details such as login credentials, uploaded notes, lectures, question papers, and other resources. PostgreSQL provides high data integrity, strong security, and efficient query processing, ensuring reliable storage and retrieval of university-related content.

PURPOSE

The primary purpose of the *AceGrade* web application is to create a **centralized digital academic platform** that simplifies access to essential educational materials and enhances communication between students and administrators. In most universities, academic resources like lecture notes, previous year question papers, and study materials are often scattered across multiple platforms or shared manually, causing inefficiency and limited accessibility. *AceGrade* aims to eliminate these challenges by offering a single, well-structured system where all academic content can be securely uploaded, managed, and accessed anytime.

The project is designed to **bridge the gap between students and academic resources** through a user-friendly web interface built using HTML and CSS, and a robust backend powered by Spring Boot. It enables **students** to easily browse and download notes, lectures, and question papers, while **administrators** can upload, update, or remove study materials as needed. Additionally, the built-in **CGPA calculator** allows students to track their academic performance in real time, encouraging self-assessment and academic improvement.

Overall, the purpose of *AceGrade* is to **digitalize academic management** by combining simplicity, accessibility, and automation. It promotes paperless resource sharing, efficient data handling, and organized content management within educational institutions. By integrating both student and admin functionalities into one platform,

AceGrade serves as a scalable and maintainable solution for modern digital learning environments.

APPLICATIONS

1. **University and College Resource Portals:**

AceGrade can be implemented in universities and colleges to serve as a central platform for managing academic resources. Students can access notes, previous question papers, and lecture materials without relying on physical copies or manual distribution.

2. **E-Learning Platforms:**

The project can act as a lightweight e-learning system where educational content is easily shared and managed. It provides an organized structure for storing lectures and study resources, supporting self-paced learning among students.

3. **Departmental Material Management:**

Academic departments can use *AceGrade* to maintain a department-specific repository for courses, lab manuals, and tutorials. This ensures every student has equal access to updated and verified content uploaded by the faculty or admin.

4. **Student Progress Monitoring:**

With its CGPA calculator feature, *AceGrade* helps students evaluate their academic performance continuously. Institutions can further extend this functionality for maintaining student progress records and generating performance insights.

5. **Institutional Knowledge Sharing System:**

The system encourages collaborative knowledge sharing by allowing faculties to upload valuable materials and students to access them easily. This enhances academic engagement and promotes continuous learning within the institution.

TECH STACK

Backend: Spring Boot 3 (Java 17), Spring Web, Spring Data JPA, Bean Validation

Database: PostgreSQL

Build: Maven

Frontend: HTML, CSS, Vanilla JavaScript

File Storage: Local filesystem (/uploads)

Core Features and Use Cases

Study Resources

- Filter by semester, department, and regulation
- View subjects and download uploaded PDF notes
- Upload new notes as PDF with contributor name and description

Question Papers

- Upload and download previous year question papers (PDF)
- Filter by semester/department/regulation

CGPA Tracker

- Persist per-semester courses, grades, and credits per user
- Save and retrieve data tied to a user

Authentication (Basic)

- Simple login via college email and password
- Health check endpoint

Repository Structure

```
AceGrade/
  backend/
    pom.xml                # Spring Boot + Maven configuration (Java 17)
    database-setup.sql     # PostgreSQL schema/seed helper
    src/main/java/com/acegrade/
      AceGradeApplication.java
      controller/
        CgpaController.java    # /api/cgpa
        LoginController.java   # /api/auth
        QpaperController.java  # /api/qpaper
        StudyResourceController.java # /api/study-resources
      dto/                     # Request/response DTOs
      entity/                  # JPA entities (User, Subject, StudyResource, CGPA*)
      repository/              # Spring Data repositories
    src/main/resources/
      application.properties   # DB, CORS, and multipart config
      uploads/                 # Runtime: stored files (notes/qpapers)

  frontend/
    index.html               # Landing page
    login.html               # Login UI
    dashboard.html           # User dashboard
    cgpa-calculator.html     # CGPA tracker UI
    study-resources.html      # Notes UI (upload/list/download)
    study-resources.js        # Notes page scripting
    qpapers.html             # Question papers UI
    qpapers.js               # QP page scripting
    styles.css / study-resources.css # Styling
    FONTS/, IMAGES/          # Assets
```

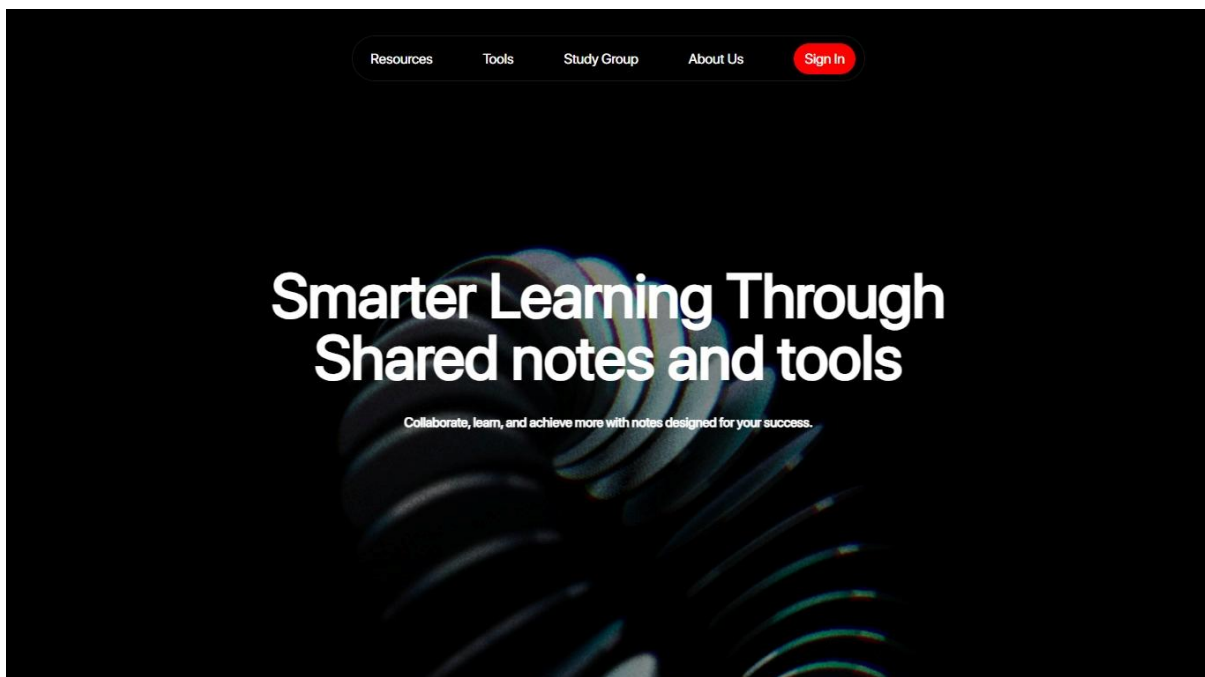
Github link: <https://github.com/SridharNaveenPA/AceGrade>

API Overview (high level)

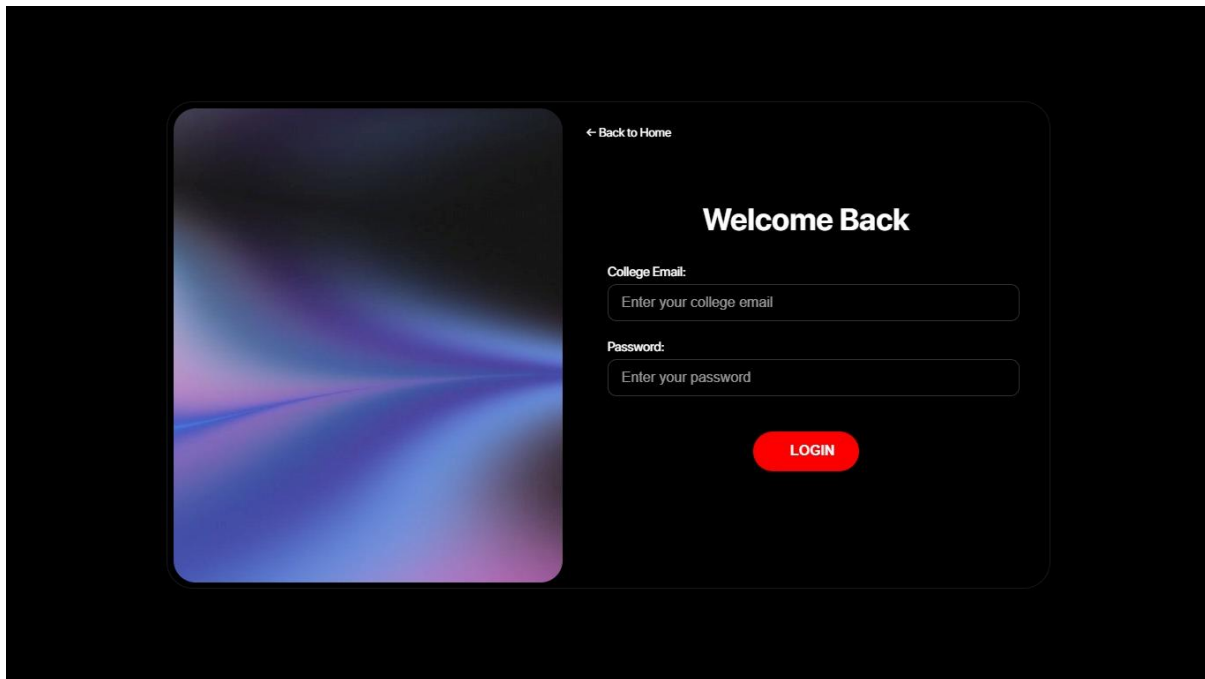
- Auth: POST /api/auth/login, GET /api/auth/health
- Study Resources:
- GET /api/study-resources/subjects (filters: semester, department, regulation)
- GET /api/study-resources (filters + optional searchTerm)
- POST /api/study-resources (multipart PDF upload)
- GET /api/study-resources/download/{id}
- Question Papers: similar endpoints under /api/qpaper
- CGPA: GET /api/cgpa/{userId}, POST /api/cgpa/{userId}

PROJECT OUTPUT

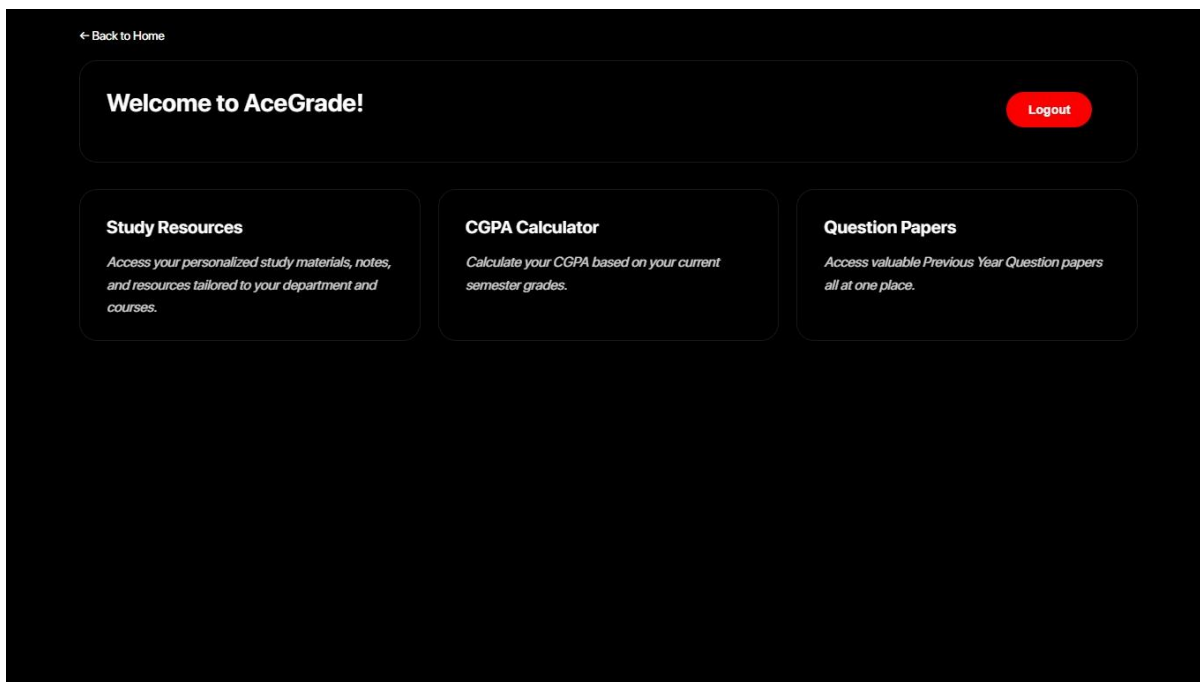
Homepage : The homepage of *AceGrade* provides a clean and modern interface that introduces the platform's purpose — *"Smarter Learning Through Shared Notes and Tools."* It features a simple navigation bar with links to **Resources**, **Tools**, **Study Group**, and **About Us**, along with a **Sign In** button. Designed using HTML and CSS, it highlights collaboration and easy access to academic materials, giving users a clear overview of the system's goal and smooth navigation experience.



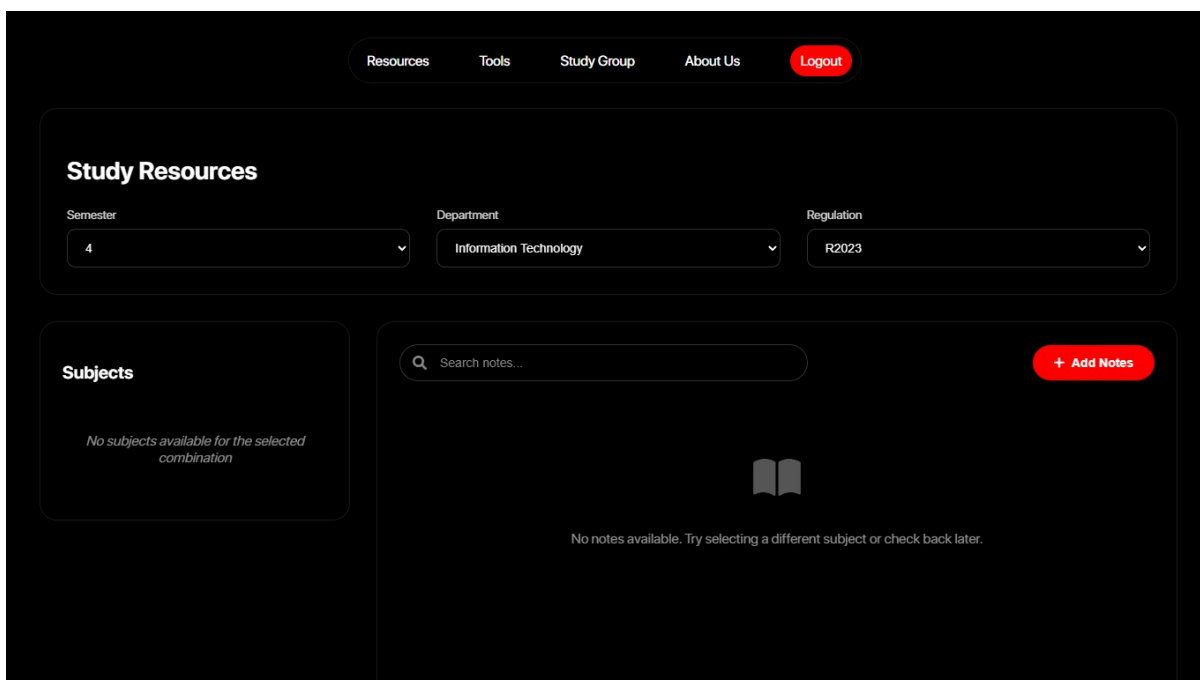
Login Page: The login page of *AceGrade* provides a secure entry point for users to access the system. It features a simple and clean interface with fields for entering a **college email** and **password**, ensuring authorized access for both students and administrators. The **“Login”** button initiates authentication through the Spring Boot backend. Designed with HTML and CSS, the page maintains a modern aesthetic with a dark background and a soft gradient image, offering a professional and user-friendly experience.



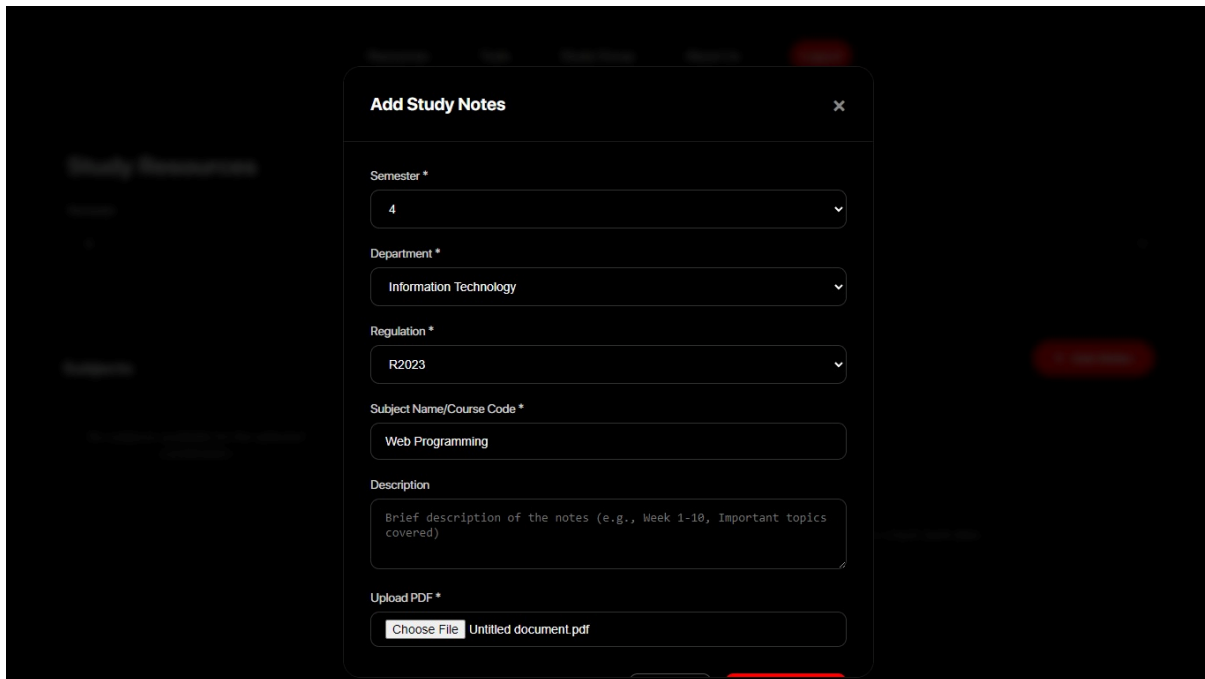
Dashboard: The dashboard page of *AceGrade* serves as the **main control panel** for users after login. It provides quick access to key modules such as **Study Resources**, **CGPA Calculator**, and **Question Papers**, allowing students to navigate efficiently through the system. The interface welcomes the user and includes a **Logout** button for secure session termination. Designed with HTML and CSS, the page maintains a minimal, dark-themed layout that ensures clarity, simplicity, and an organized user experience.



Study Resources Page: The Study Resources page allows students to easily access academic materials like notes and resources specific to their **semester**, **department**, and **regulation**. Users can filter and view available subjects and search for notes through an intuitive interface. An **“Add Notes”** button enables contribution of new study materials, promoting collaborative learning. Designed with **HTML** and **CSS**, and powered by **Spring Boot**, the page ensures a clean, user-friendly layout that simplifies navigation and resource management for students.



Add Study Notes Page: The Add Study Notes page allows users to contribute academic materials by uploading notes in PDF format. It includes input fields for **Semester**, **Department**, **Regulation**, **Subject Name/Course Code**, and a **Description** of the content. This structured form ensures proper categorization of study materials, making it easier for other students to find relevant resources. Built using **HTML**, **CSS**, and **Spring Boot**, the page provides a clean, interactive, and efficient interface for academic collaboration and resource sharing.

A screenshot of a web application interface for adding study notes. The form is titled "Add Study Notes" and is set against a dark background. It contains several input fields: "Semester *" with a dropdown menu showing "4"; "Department *" with a dropdown menu showing "Information Technology"; "Regulation *" with a dropdown menu showing "R2023"; "Subject Name/Course Code *" with a text input field containing "Web Programming"; "Description" with a text area containing placeholder text "Brief description of the notes (e.g., Week 1-10, Important topics covered)"; and "Upload PDF *" with a file upload button labeled "Choose File" and a text input field showing "Untitled document.pdf". A red "Submit" button is visible on the right side of the form.

CGPA Calculator Page: The CGPA Calculator page enables students to calculate their **Cumulative Grade Point Average (CGPA)** accurately and efficiently. Users can add multiple semesters and courses, entering details like **Course Name**, **Grade**, and **Credits** for each subject. The system automatically computes the CGPA based on the weighted average formula and displays the result instantly. Developed using **HTML**, **CSS**, and **Spring Boot**, this tool provides a user-friendly and responsive interface that helps students easily monitor and track their academic performance across semesters.

Resources
Tools
Study Group
About Us
Logout

CGPA Calculator

Semester 1

+ Add Course
Remove

Course Name	Grade	Credits
Maths 1	A+	4
+ Add Course		

Semester 2

+ Add Course
Remove

Course Name	Grade	Credits
Data Structures	S	4
+ Add Course		

Add Semester
Save

Cumulative GPA

$$GPA = \frac{\sum(\text{grade_points} \times \text{credits})}{\sum(\text{credits})}$$

9.50

Question Papers Page: The Question Papers page provides students with easy access to **previous year question papers** for better exam preparation. Users can filter available papers by **Semester**, **Department**, and **Regulation**, ensuring quick and organized retrieval of relevant materials. A built-in **search feature** helps locate specific subjects, while the **“Add QPaper”** option allows users to upload new question papers, promoting a collaborative academic environment. Designed using **HTML**, **CSS**, and **Spring Boot**, the page offers a simple, responsive, and user-friendly interface for efficient study resource management.

Resources
Tools
Study Group
About Us
Logout

Question Papers

Semester

Department


Regulation

Select Semester
Select Department
Select Regulation

Subjects

Please select semester, department, and regulation to view subjects

Search notes...
+ Add QPaper



No notes available. Select filters to view study materials.

AceGradeApplication.java:

```
package com.acegrade;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class AceGradeApplication {

    public static void main(String[] args) {

        SpringApplication.run(AceGradeApplication.class, args);

    }

}
```

PROJECT OVERVIEW

AceGrade is a web-based Notes Management System designed to simplify access to academic resources for students. The platform integrates essential tools such as **Study Resources**, **CGPA Calculator**, and **Question Paper Repository** into a single user-friendly interface. Built using **HTML**, **CSS**, and **Spring Boot**, it ensures seamless functionality, responsive design, and efficient data management.

The system allows students to browse, upload, and organize study materials and previous year question papers based on **semester**, **department**, and **regulation**. The integrated CGPA Calculator enables students to easily compute their academic performance across semesters. By providing centralized access to essential academic content, AceGrade enhances learning efficiency, supports collaboration, and reduces the dependency on multiple external resources.