# College Automation Portal - Project Guidelines

## Updated Key Features

Teacher Module Teacher Module (Faculty)

- Upload lecture notes (PDFs, PPTs, etc.) to specific courses.

- Add internal/external grades for students in their assigned courses.

- Create and manage online assessment exams (objective/subjective).

- View submissions and grade assessments.

- Share important announcements and resources with students.

Student Module Student Module

- Secure login with student ID.

- View list of enrolled courses.

- Access all uploaded lecture notes, assignments, and resources per course.

- View grades for each course (internal, external).

- Register for and attend online assessments posted by teachers.

- View assessment results and feedback.

## Project Development Guidelines

1. Requirement Analysis

- Define user roles: Admin, Teacher, Student.

- Choose technology stack: Frontend (React.js), Backend (Node.js/Django), DB (MySQL/MongoDB).

2. Database Design

- Tables: Users, Courses, Notes, Grades, Assessments, StudentAssessment.

3. Backend API Development

- Authentication, Course listing, Notes upload, Grade fetch, Assessment creation, Submission endpoints.

4. Frontend Implementation

# College Automation Portal - Project Guidelines

- Role-based Dashboards, Course/Note viewers, Exam UI, Grade view.

- Use react-router for navigation.

5. Authentication & Authorization

- Use JWT for secure login and role-based access.

- bcrypt for password hashing.

6. File Upload & Storage

- Use Multer or equivalent for handling uploads.

- Store on local or cloud storage.

7. Online Exam System

- Timed exams, MCQ & Subjective handling, Auto-score, Result summary.

8. Testing & Debugging

- Postman for API testing.

- Input validation and error handling.

9. Deployment

- Frontend: Vercel/Netlify

- Backend: Render/Heroku/Railway

10. Documentation

- API Docs (Swagger)

- User guides for teachers/students

**Optional Enhancements**

- Admin dashboard to manage users and courses.

- Email/SMS notification system.

# College Automation Portal - Project Guidelines

- Real-time chat using Socket.IO.

- Analytics dashboard for performance tracking.