Online Examination Portal

Project Report

1. Abstract

The Examination Portal Management System is a web-based platform developed using React (Frontend), Spring Boot (Backend), and MySQL (Database) to facilitate online examinations. The system allows students to register, take exams, and view their results, while users (such as faculty) can create, manage, and evaluate exams efficiently.

2. Objectives

- **To provide an easy-to-use online platform** for students to register, take exams, and view results.
- **To enable administrators and faculty** to create, manage, and evaluate exams efficiently.
- **To implement a secure and automated examination system** with features like time-bound tests and instant result generation.

3. System Users

3.1 Students (User)

- **Registration & Login:**

Students can sign up and log in to the system.

- **Exam Selection:**

Browse available exams based on subjects.

View exam details such as duration, question format, and instructions.

- **Exam Attempt:**

Start the selected exam.

Time-bound tests to ensure fairness and prevent delays.

Answer multiple-choice.

- **Auto-Grading & Result Viewing:**

Objective (MCQ) questions are graded instantly.

3.2 Faculty

- **Registration & Login:**

Faculty members can sign up and log in.

- **Exam Creation & Management:**

Create exams by specifying subjects, and questions.

Set exam duration, total marks, and pass criteria.

- **Question Management:**

Upload and manage a questions for different subjects.

Add, edit, or delete multiple-choice questions.

3.3 Admin

- **Registration & Login:**

Admin users can sign up and log in.

- **Course & Subject Management:**

Create and manage courses.

Add, edit, or remove subjects under each course.

Assign subjects to specific faculty members.

- **Faculty Management:**

Add new faculty members and assign them roles.

Approve or deactivate faculty accounts.

4. Functional Requirements

- **User Authentication**: Registration and login for student and faculty.
- **Faculty Functionality**: Faculty can create, update, and delete exam.
- **Student Functionality**: Student can give exam, view result instantly.
- **Course and Subject Management**:
    - Admin can add courses and under courses can add subjects.

5. Non-Functional Requirements

- **Scalability**: The system is designed to handle multiple student and faculty.
- **Performance**: Efficient database queries for fast response times.
- **Usability**: Simple and user-friendly UI using **React**.
- **Maintainability**: Modular codebase for easy updates.

6. System Workflow

1. **User Registration/Login** → Users (Admin, Faculty, or Students) enter details and log in.

2. **Faculty Request** → Faculty members create exams by setting subjects, and adding questions.

3. **Student Request** → Students browse available exams, select an exam, and start the test.

4. **Exam Attempt** → Students take the exam with time-bound restrictions, answering MCQs.

5. **Result Evaluation** → MCQs are auto-graded instantly.

6. **Score Calculation** → The system calculates total marks based on correct answers and predefined grading criteria.

7. **Admin Management** → Admin users manage courses, subjects, faculty in the system.

7. Database Design

Tables:

1. **Faculty Table** → Stores faculty details.
2. **Student Table** → Stores student details.
3. **Courses Table** → Stores course details.
4. **Subject Table** → Stores subject details under course.
5. **Quiz Table** → Stores quiz details under subject.
6. **Question Table** → Stores question details under quiz.

8. Technologies Used

- **Frontend**: React, HTML, CSS, JavaScript
- **Backend**: Spring Boot
- **Database**: MySQL