**Online Exam Portal Documentation**

**1. Project Overview**

The **Online Exam Portal** is a web-based application designed to conduct assessments for educational institutions and corporate training programs. It supports multiple question types, automated scoring, real-time monitoring, and performance analytics.

**Key Objective:** Provide a scalable, secure, and interactive platform for online testing with role-based access.

**Roles:**

* Super Admin
* Admin
* Student / Client
* Employee

**2. Features**

**2.1 User Management**

* Multi-role registration (Student, Admin, Super Admin, Employee)
* JWT-based authentication and secure login
* Profile management, password reset, and email verification
* Role-based access control (RBAC)

**2.2 Test & Question Management**

* Categories for exams (e.g., Python, java, Aws, etc..)
* Question bank with tagging by difficulty
* Multiple question types: MCQ, True/False
* Randomized questions per exam
* Exam scheduling with duration and start/ end time.

**2.3 Exam Features**

* Timer-based exams with live countdown
* Auto-save user answers
* Question navigation (Next, Previous, Review)
* Anti-cheating mechanism (optional browser lock, webcam monitoring)
* Exam cancellation and resumption support

**2.4 Evaluation & Results**

* Automated scoring for objective questions
* Dynamic pass/fail calculation per student
* Detailed result display per exam and question
* Leaderboard and category-wise analytics

**2.5 Reporting & Analytics**

* Exam-wise charts for pass/fail and average scores
* Attempts per category and student performance tracking
* Export results as CSV/Excel (optional)

**3. System Architecture**

**3.1 Frontend**

* **React.js** with **Material-UI** for responsive UI
* React-Router for role-based navigation
* Axios for API calls with JWT authorization
* Components include:
  + **LoginPage.jsx** – Handles authentication
  + **RegisterPage.jsx** – New user registration
  + **Navbar.jsx** – Role-based menu
  + **StudentDashboard.jsx** – Exam list, timer, questions, submit button
  + **ChartModal.jsx** – Exam analytics charts
  + **AdminDashboard.jsx** – Leaderboard, pass/fail, exam analytics
  + **ExamCard.jsx** – Card view for each exam
  + **Leaderboard.jsx** – Student ranking

**3.2 Backend**

* **Django + DRF** (Django REST Framework)
* JWT authentication with SimpleJWT
* Apps/modules:
  1. **Accounts** – Handles user registration, login, password reset, role-based access
  2. **Exams** – Manages categories, exams, questions, and schedules
  3. **Attempts** – Tracks student attempts and submitted answers
  4. **Leaderboard** – Computes top students and performance metrics

**3.3 Database**

* **MySQL** (primary) / SQLite (development)
* Tables:
  + **Users**: id, username, email, password, role
  + **Categories**: id, name
  + **Exams**: id, title, description, category\_id, duration\_minutes
  + **Questions**: id, exam\_id, text, options (JSON), correct\_option
  + **Attempts**: id, user\_id, exam\_id, score, submitted\_answers (JSON), pass/fail
  + **Leaderboard**: id, exam\_id, user\_id, score, rank

**4. API Endpoints**

**4.1 Accounts**

| **Endpoint** | **Method** | **Description** |
| --- | --- | --- |
| /api/accounts/register/ | POST | Register new user |
| /api/accounts/login/ | POST | Login and return JWT |
| /api/accounts/logout/ | POST | Logout user |

**4.2 Categories & Exams**

| **Endpoint** | **Method** | **Description** |
| --- | --- | --- |
| /api/categories/ | GET | List all categories |
| /api/categories/<id>/exams/ | GET | List exams by category |
| /api/exams/<id>/questions/ | GET | Get all questions for exam |
| /api/exams/<id>/submit/ | POST | Submit answers and compute score |

**4.3 Attempts & Leaderboard**

| **Endpoint** | **Method** | **Description** |
| --- | --- | --- |
| /api/attempts-per-category/ | GET | Get user attempts per category |
| /api/leaderboard/ | GET | Fetch leaderboard and scores per exam |

**5. Control Flow**

flowchart TD

A[Login/Register] --> B[Dashboard]

B -->|Student| C[Select Category]

C --> D[Select Exam]

D --> E[Start Exam]

E --> F[Answer Questions]

F --> G[Submit Exam]

G --> H[Compute Score & Pass/Fail]

H --> I[Show Result & Leaderboard]

B -->|Admin| J[View Analytics]

J --> K[Charts: Pass/Fail, Avg Score, Attempts]

**6. Frontend Components**

| **Component** | **Description** |
| --- | --- |
| LoginPage.jsx | Login page with JWT |
| RegisterPage.jsx | Registration form for new users |
| Navbar.jsx | Navigation with role-based links |
| StudentDashboard.jsx | View exams, take test, timer, submit |
| ChartModal.jsx | Exam analytics charts for students |
| AdminDashboard.jsx | Exam performance analytics for admins |
| ExamCard.jsx | Individual exam card |
| Leaderboard.jsx | Student ranking table |
| AttemptsList.jsx | Student attempts per category |

**7. Security**

* JWT authentication for all API requests
* Role-based access control (RBAC):
  + Super Admin – Full control
  + Admin – Manage categories, exams, analytics
  + Student – Take exams, view results
* Passwords stored hashed with PBKDF2 / bcrypt

**8. Tech Stack**

* **Frontend:** React.js, Material-UI, Axios, React-Router
* **Backend:** Django, Django REST Framework, SimpleJWT
* **Database:** MySQL (production), SQLite (development)
* **Charts:** Highcharts for analytics
* **Deployment:** AWS EC2

**9. Future Enhancements**

* Webcam proctoring and browser lockdown
* Export reports to PDF/Excel
* Real-time notifications for exam start/end
* Advanced analytics: question difficulty, progress tracking
* Multi-language support for exams