Exam System –Set-Up & Demo

🚀 Quick Start: Things You Need Before Starting

- Install **.NET 8 SDK**
- Install **Node.js 16 or newer**
- Install **Visual Studio Code** or **Visual Studio**

📋 Step-by-Step Demo

1. Backend Setup

1. Go to the **API project folder**
2. Restore required files → `dotnet restore`
3. Build the project → `dotnet build`
4. Run the project → `dotnet run`

What you should see:

- API running at: `http://localhost:5000`
- API Testing Page (Swagger) at: `http://localhost:5000/swagger`
- Database created with sample users (Admin + Student)

2. Frontend Setup

1. Go to the **frontend project folder**
2. Install dependencies → `npm install`
3. Start the app → `npm start`

What you should see:

- React app starts at: `http://localhost:3000`
- The login page shows up with demo login details

🎯 Demo Scenarios

Scenario 1: Admin Workflow

Step 1: Log in as Admin

1. Open: `http://localhost:3000`
2. Enter login details:
   - Username: **admin**
   - Password: **admin123**

3. Click **Login**

What you should see:

- You are redirected to the Admin Dashboard

Step 2: Create Exam

1. Click on **Admin View**
2. Fill in the exam form:
    - Title: *JavaScript Fundamentals*
    - Max Attempts: *3*
    - Cooldown Minutes: *30*

3. Click **Submit**

What you should see:

- Message: *Exam created successfully!*
- Form is cleared for new input

Step 3: View Attempts

1. Scroll to the **Student Attempts History** section
2. Initially shows “No attempts found”

Scenario 2: Student Workflow

Step 1: Log in as Student

1. Log out of the Admin account
2. Enter login details:
    - Username: **student**
    - Password: **student123**
3. Click **Login**

What you should see:

- You are redirected to the Student Dashboard

Step 2: View Exam Details

- Click **Student View**
- You will see:
    - Exam title and settings
    - Remaining attempts
    - Cooldown status

Step 3: Start Attempt

1. Click **Start Attempt**
2. Screen updates

What you should see:

- Button changes to **Submit Attempt**
- Message: *Attempt started successfully!*
- Attempts left decrease by 1
- History shows “In Progress”

Step 4: Submit Attempt

1. Click **Submit Attempt**
2. Screen updates

What you should see:

- Message: *Attempt submitted successfully!*
- Button goes back to **Start Attempt**
- Attempt status changes to “Completed”
- End time is shown

Step 5: Cooldown

- Try to start again immediately

What you should see:

- Button disabled or shows: *Next attempt available at [time]*

Step 6: Multiple Attempts

- Wait for cooldown or set cooldown to 0 (via Admin)
- Start and submit more attempts

What you should see:

- After 3 attempts → message: *No attempts left*
- Button disabled

Scenario 3: Admin Monitoring

Step 1: Switch to Admin View

- Log in as Admin
- Click **Admin View**
- Scroll to Student Attempts

What you should see:

- All student attempts with details

- Live updates

Step 2: Update Exam

1. Change exam settings:
    - Title: *JavaScript Fundamentals – Updated*
    - Max Attempts: *5*
    - Cooldown Minutes: *15*
2. Click **Submit**

What you should see:

- Message: *Exam created successfully!*
- Student attempts reset
- Students can start fresh

🔧 API Testing

- Open **Swagger UI** at: `http://localhost:5000/swagger`
- Test login with Admin details
- Use the returned token for protected endpoints

🧪 Testing Validation Rules

- Try logging in with wrong details → error messages
- Try creating invalid exams (like empty title, too many attempts) → error messages

---

🔍 Database Check

- Admin can see all attempts using `/api/attempt/all`
- Students can see their exam details using `/api/exam/student`

🚨 Error Scenarios

- Wrong login details
- Token expired (after 24 hours)
- Starting attempt after maximum attempts
- Starting attempt during cooldown

📊 Performance Testing

- Try multiple logins at once
- Try concurrent exam attempts
- Check response times:
  - Login < 500 ms
  - Exam operations < 200 ms
  - Attempts < 300 ms

🔧 Troubleshooting

- **CORS Error**: Make sure the API is running on port 5000
- **Database Error**: Restart API
- **Frontend Error**: Check if the API is reachable
- **Token Error**: Re-login to refresh the token