Institute Management System (IMS)

Project Overview

This project is called **Institute Management System (IMS)**. It is a web application built using **React JS**, **React Router**, **Redux**, and **Firebase**.

This project is designed for **beginner React students**, but the overall project level is **intermediate**. This means:

- You will practice real-world concepts
- You will understand how a complete app works
- You will gain confidence to build bigger projects

This will be your **first proper full-stack style project** using frontend + backend services (Firebase).

Technologies Used

Frontend

- React JS
- React Router DOM
- Redux Toolkit

Backend / Services

- Firebase Authentication
- Firebase Firestore (Database)
- Firebase Hosting (Optional)

User Roles

There are **two roles** in this project:

1. **Admin**
2. **Student**

For beginner level, we will keep roles simple and manageable.

Core Features (What this app can do)

Admin Can:

- Login
- Add courses
- View all courses
- Add students
- View all students
- Assign courses to students
- View dashboard summary

Student Can:

- Login
- View assigned courses
- View profile

Application Pages & Routes

Below are **all routes** in this project. Each route is explained in **simple English**.

---

1. Authentication Routes

1.1 Login Page

Route: `/login`

Who can access: Admin & Student

Purpose:

- User enters email and password
- Firebase checks credentials
- If correct, user is logged in

Functionality:

- Email input
- Password input
- Login button
- Show error if login fails
- Redirect based on role

1.2 Protected Routes

Some pages should **not open without login**.

Examples:

- `/dashboard`
- `/courses`
- `/students`

How it works (simple):

- Check if user exists in Redux state
- If not logged in → redirect to `/login`

---

2. Dashboard

2.1 Admin Dashboard

Route: `/dashboard`

Purpose:

- Show overview of institute

What to show:

- Total students count
- Total courses count
- Logged-in admin name

Functionality:

- Fetch data from Firebase
- Display numbers in cards

2.2 Student Dashboard

Route: `/student/dashboard`

Purpose:

- Show student basic info

What to show:

- Student name
- Student email
- Number of enrolled courses

---

3. Courses Module

3.1 Add Course (Admin)

Route: `/courses/add`

Purpose:

- Admin can create a new course

Fields:

- Course name
- Course description
- Course duration

Functionality:

- Form submission
- Save data to Firestore
- Show success message

3.2 View Courses

Route: `/courses`

Purpose:

- Show list of all courses

Functionality:

- Fetch courses from Firestore
- Display in table or cards
- Show course name and duration

---

4. Students Module

4.1 Add Student (Admin)

Route: `/students/add`

Purpose:

- Admin can register a student

Fields:

- Student name
- Email
- Password

Functionality:

- Create user in Firebase Auth
- Save student info in Firestore
- Assign role = "student"

4.2 View Students

Route: `/students`

Purpose:

- Admin can see all students

Functionality:

- Fetch students from Firestore
- Display name and email

---

5. Assign Course to Student

5.1 Assign Course Page

Route: `/assign-course`

Purpose:

- Admin assigns courses to students

Fields:

- Select student (dropdown)
- Select course (dropdown)

Functionality:

- Save assigned course in Firestore
- One student can have multiple courses

---

6. Student Features

6.1 My Courses

Route: `/my-courses`

Purpose:

- Student can see their enrolled courses

Functionality:

- Fetch courses using student ID
- Display course name and duration

6.2 Student Profile

Route: `/profile`

Purpose:

- Show logged-in student info

What to show:

- Name
- Email
- Role

Redux State Structure (Simple)

Auth Slice

- user
- role
- loading

Course Slice

- courses list

Student Slice

- students list

Firebase Collections Structure

users

- id
- name
- email
- role

courses

- id
- name
- description
- duration

enrollments

- studentId
- courseId

Learning Outcomes

By completing this project, students will learn:

- How routing works in React
- How authentication works
- How to use Redux in real apps
- How to connect frontend with Firebase
- How to structure a real project

Final Notes for Students

- Don’t rush this project
- Build one module at a time
- Focus on understanding, not copying
- This project will make you confident

This project is **hard at first**, but once completed, your React skills will level up 🚀
