

GitHub:

<https://github.com/MasoumehSedighi/Educational-platform-MERN-Stack>

Component 1

Code Overview

App.js

This file helps users move between the home page and course details easily. The Navbar is always there to guide them.

Home.jsx

The home page shows a grid of courses with their titles, images, and how long each one takes.

CourseDetail.jsx

When users want to learn more about a course, this file shows all the important information, including its modules. There's also a "Enroll Now" button for easy access.

Navbar.jsx

The Navbar helps users navigate the app and provides a link back to the home page. It makes moving around the app simple.

index.css

This file makes the app look good by adding styles to the layout. It ensures everything is nice and clear on any device.

Technical Details

App.js

Functionality: This file manages how users move around the app.

Input Data: None.

Expected Output: It lets users go to the home page and course detail pages easily through the Navbar.

Home.jsx

Functionality: Shows a grid of courses .

Input Data: Gets an array of courses from the courses data file.

Expected Output: A grid displays each course's title, image, description, and duration, making it easy for users to browse available courses.

CourseDetail.jsx

Functionality: Shows detailed information about a specific course based on the selected course ID.

Input Data: Uses useParams to get the id from the URL.

Expected Output: Displays the course title, image, description, modules, and an "Enroll Now" button. If the course is not found, it shows a message that the course was not found.

Navbar.jsx

Functionality: Provides links for navigation in the app.

Input Data: None.

Expected Output: A navigation bar with a link back to the home page, making it easy for users to navigate.

index.css

Functionality: Styles the layout and appearance of the app.

Input Data: None.

Expected Output: A clean and attractive layout that works well on different devices, improving the user experience.

Problem-Solving

1. Responsive Design

Some parts of the app didn't fit well on smaller screens. I fixed this by using a mobile-first design and adding media queries in the CSS files.

2. Routing Issues

When switching between pages, there were errors with routes and reloading components. I solved this by setting up React Router correctly and making sure all routes were defined properly.

3. Image Scaling

Images on the course detail page weren't resizing correctly. I fixed this by changing the width and height of the images in CSS so they look good on all devices.

4. Handling Empty Course Data

Sometimes, a course ID didn't exist, and the app needed to show a message instead of crashing. I added a check to display a "Course not found" message when the app couldn't get the course data.

Achievements

1. Responsive Design

The app works well on all devices, like desktops, tablets, and mobiles. I achieved this by starting with a mobile design and using CSS Grid and Flexbox for the layout.

2. Error Handling

I added checks and helpful error messages to handle cases where data is missing. This improves user experience and stops the app from crashing.