React JS Practice Challenges

Mastery of React Basics.

Overview.

Do these mini-projects individually.

You can write the code, run it locally on your VS Code editor, and then push it to GitHub. Make sure you add comments to your code so that it's easy to read and understand.

Please Take Note:

- **DO NOT copy solutions online or from anywhere**.
  Try to do this yourself

  Prerequisites.
- You should **create a GitHub repo** for these mini-projects.
  Then create separate folders for each solution.

Submission deadline:
25th September 2025 at 10:00 pm (EAT)

Mini Projects (Set 1)

—----------

1. **Random Quote Generator**
   Build a `QuoteGenerator` component that displays a random quote from a Quotes API like https://dummyjson.com/docs/quotes. Add a **next** and **previous** button so that a user a go to the previous or next quote from the list of quotes fetched from the API.

2. **Weather Card**
   Build a WeatherCard component that takes city, temperature, and condition (e.g., "Sunny" or "Rainy") as props and displays them. Create multiple weather cards to display different cities. *Get the weather data from a weather data API like https://www.weatherapi.com/*

Mini Projects (Set 2)

—----------

1. **Simple User Authentication**

   Build a simple authentication flow using Mantine UI, React Router, and Auth0. Your application should have three separate pages:

1. **Signup Page** (`/signup`)
2. **Login Page** (`/login`)
3. **Dashboard Page** (`/dashboard`)

Use React Router to navigate between these pages.

- **On the Signup Page**, create a form that captures the user's full name, email, and password. Validate all fields and store the user's information securely in `localStorage`.
- **On the Login Page**, create a form with email and password fields. When the user submits the form, validate the inputs and check if the credentials match the data saved in `localStorage`. If they do, display a "Welcome back!" message and redirect the user to the Dashboard. If not, show an error.
- Additionally, implement a **"Sign in with Google"** feature using Auth0. If the user logs in successfully via Google, redirect them to the Dashboard page as well.
- **The Dashboard Page** should only be accessible after a successful login and should display a simple welcome message or user details.

2. **News Feed Page**

Build a **News Feed Application** with a custom backend using **Strapi** and a frontend using **React**. You are required to set up both the backend and frontend as follows:

Backend (Strapi)

Use **Strapi** to build your backend API for managing news posts.
Each post should include the following fields: **title**, **author**, **content**,

excerpt, and **category**.

Ensure the API supports fetching:

- All posts (with pagination)
- A single post by ID or slug
- Posts filtered by category or title

Frontend (React)

Use **React Router** to create the following pages:

1. **News Feed Page** (`/news`)
2. **Single Post Page** (`/news/:id`)

Use **Mantine** (or any React component library of your choice) to implement the UI.

On the News Feed Page:

Display a **paginated** list of posts retrieved from your Strapi API.

Each post should show:

- Title
- Cover Image
- Author
- A short excerpt
- A "Read More" button

Clicking "Read More" should navigate to the **Single Post Page** using **dynamic routing** with `React Router`.

On the Single Post Page:

Display the **full content** of the selected post. Use the dynamic route to fetch the post data based on its ID or slug from the URL.

Additional Features for news feed:

- Implement a **search bar** to filter posts by **title** or **category** on the News Feed Page.

- Add **loading indicators** and **error handling** to improve the user experience during API calls.

📌📌

Submit your Project

Use this form: https://forms.gle/CFoqWekEJrh5Tc4K6