

FocusTasks - Mini React Project

1. PROJECT OVERVIEW

Build a simple productivity app that allows users to:

- Add tasks
 - Mark tasks as complete
 - Delete tasks
 - Persist tasks using localStorage
 - Switch between Light/Dark theme
 - Fetch a motivational quote from an API
 - Navigate across three pages
-

2. APPLICATION ROUTES (React Router)

Your app will include 3 pages:

1. / → Dashboard
2. /tasks → Task List + Add Task Form
3. /settings → Theme Toggle

Create a Navbar with links to each page.

3. THEME SYSTEM (useContext)

Create ThemeContext that stores:

- theme ("light" or "dark")
- toggleTheme() to switch the theme

Wrap the entire App with ThemeProvider so theme is available across pages.

Use theme to style:

- Navbar background
- Page backgrounds
- Buttons

4. TASK MANAGEMENT (useReducer)

Use useReducer to store and update all tasks.

Task Model:

- id
- title
- isCompleted

Actions:

- ADD_TASK
- TOGGLE_TASK
- DELETE_TASK

Examples:

```
dispatch({ type: "ADD_TASK", payload: newTask });
```

```
dispatch({ type: "TOGGLE_TASK", payload: id });
```

5. ADD TASK FORM (useState + Forms)

Form Fields:

- Task Title (text input)

Requirements:

- Use useState to store the input value
 - Validate: show an error if title is empty
 - On submit:
 - dispatch ADD_TASK to reducer
 - clear input
 - Autofocus input using useRef
-

6. LOCAL STORAGE PERSISTENCE (useEffect)

Use two useEffects:

1) On first render:

- Read tasks from localStorage
- Initialize state if tasks exist

2) Whenever tasks change:

- Save updated tasks to localStorage

7. UI ENHANCEMENTS (useRef)

Use useRef to:

- Autofocus task input on the /tasks page
 - (Optional) Scroll to the latest task when added
-

8. FETCH MOTIVATIONAL QUOTE (API + useEffect)

On the Dashboard page:

- Fetch a motivational quote using:

`https://api.quotable.io/random`

Display:

- Quote text
- Author

States to manage (useState):

- quote
- loading
- error

Provide a "Refresh Quote" button.

9. RECOMMENDED FOLDER STRUCTURE

src/

components/

Navbar.jsx

TaskList.jsx

TaskItem.jsx

context/

ThemeContext.jsx

pages/

Dashboard.jsx

Tasks.jsx

Settings.jsx

reducers/

taskReducer.js

App.jsx

index.jsx