**Phase 2: Design**

The design phase focused on planning the architecture and user interface of the Notes App. This phase translated research insights into a clear blueprint for development.

**🎯 Objectives of Design**

* Define the component structure for the React frontend.
* Design API routes and backend architecture.
* Plan the data flow between frontend and backend.
* Choose UI components and layout for optimal user experience.

**🖥️ Frontend Design**

**Component Structure:**

* **App Component:** Manages the main state and layout.
* **NoteList Component:** Displays the list of all notes.
* **NoteItem Component:** Represents each individual note with delete functionality.
* **AddNoteForm Component:** Allows users to input and submit new notes.

**State Management:**

* Use React’s useState hook to maintain notes in the frontend.
* Use useEffect to fetch notes from the backend when the app loads.

**API Integration:**

* Use Axios to interact with backend API endpoints.
* Centralize API calls in a service module for cleaner code.

**UI Design:**

* Use Material UI components for a clean, consistent look.
* Ensure responsiveness and accessibility.

**⚙️ Backend Design**

**API Endpoints:**

* GET /api/notes – Retrieve all notes.
* POST /api/notes – Add a new note.
* DELETE /api/notes/:id – Delete a note by ID.

**Middleware:**

* Use CORS to allow requests from the frontend.
* Use JSON parsing middleware for request bodies.

**Data Storage:**

* Start with in-memory storage of notes.
* Design data model to easily migrate to databases like MongoDB later.

**🔄 Data Flow**

* On app load, frontend sends a GET request to fetch notes.
* Adding or deleting notes sends POST or DELETE requests.
* Backend updates data and responds with updated information.
* Frontend updates UI state accordingly to keep synchronized.

**✅ Outcome of Design**

* Created a modular and maintainable app structure.
* Planned efficient API communication.
* Prepared for smooth development and integration.
* Designed an intuitive, user-friendly interface.