Full Stack Developer Assignment:

Develop a notes taking app based on the given requirement and the Example UI (Creating Own UI to earn extra points).

RESTRICTIONS

1) **Framework & Version Lock**
   a) Must use *only* the specified front-end and back-end frameworks (and their mandated versions).
   b) No mixing in other SPA libraries or switching to alternative server frameworks.

2) **No Pre-Made UI Component Libraries**
   a) Do *not* use libraries like Material-UI, Bootstrap, Ant Design, Tailwind UI, etc.
   b) All UI components (forms, buttons, modals, grids, etc.) must be hand-crafted (tailwind css allowed).

3) **Original Code Only**
   a) You may **not** fork, clone or otherwise copy from any external (public or private) repository.
   b) Copy-pasting code snippets from StackOverflow or blogs is prohibited unless you explicitly attribute and justify it in your README.

4) **No AI-Generated Code**
   a) Solutions generated by ChatGPT, GitHub Copilot, or similar must not be submitted as-is.
   b) Any AI-assisted snippets must be clearly noted with rationale.

5) **Dependency Constraints**
   a) You may only install (via npm/yarn/pip/etc.) packages explicitly whitelisted in the assignment spec.
   b) No additional third-party plugins, middleware, or CLI tools.

6) **Version Control & Commit Hygiene**
   a) All work must live in a **new**, private Git repository created by you.
   b) Commit at least every 2–3 hours of work with clear, descriptive messages (e.g. “feat: add user-profile API endpoint”).

7) **Documentation & README**
   a) Include a top-level README that describes:
      i) How to install and run your app locally
      ii) Any design decisions or trade-offs
      iii) Justification for any external snippets or resources used (with links).

8) **Environment & Delivery**
   a) The app must run in a standard local Docker container (no cloud services).
   b) Do not expose any secrets or API keys; use environment variables only.

c) Package your deliverable as a private Git link or ZIP—no public forks/clones.

FRONTEND

1. Requirements

- **a.** Next.js
- **b.** State Management Redux / Zustand / Recoil / Others
- **c.** Fetching Axios / React-query with Axios
- **d.** Animations Framer Motion (Optional) (Extra Points)

2. Pages

- **a.** **HOME PAGE –** to show the list of notes.
- **b.** **SIGN IN –** for user sign in.
- **c.** **SIGN UP –** for user registration.

3. Functions & Features

- **a.** Add create / view / edit / delete functions for the notes
- **b.** Add Rich text Editor for user message (Optional) (Extra Points)
- **c.** Add SEO related data such as title, description, OG, keywords in metatags (Optional)
- **d.** Optimize Code – Code Splitting / Reusable Components / Render Optimization / Others.

BACKEND

1. Requirements

- **a.** Python Django / Flask / Fastapi frameworks are only allowed.
- **b.** MySQL (SQL) / Mongodb (NOSQL) database is your preference choose wisely for better performance.
- **c.** Require proper authentication system with JWT or session based.
- **d.** File & Folder Structure should be properly structured.
- **e.** Optimize your code for better performance. (Performance Testing Reports have to be submitted)

2. Database Models

a. USER

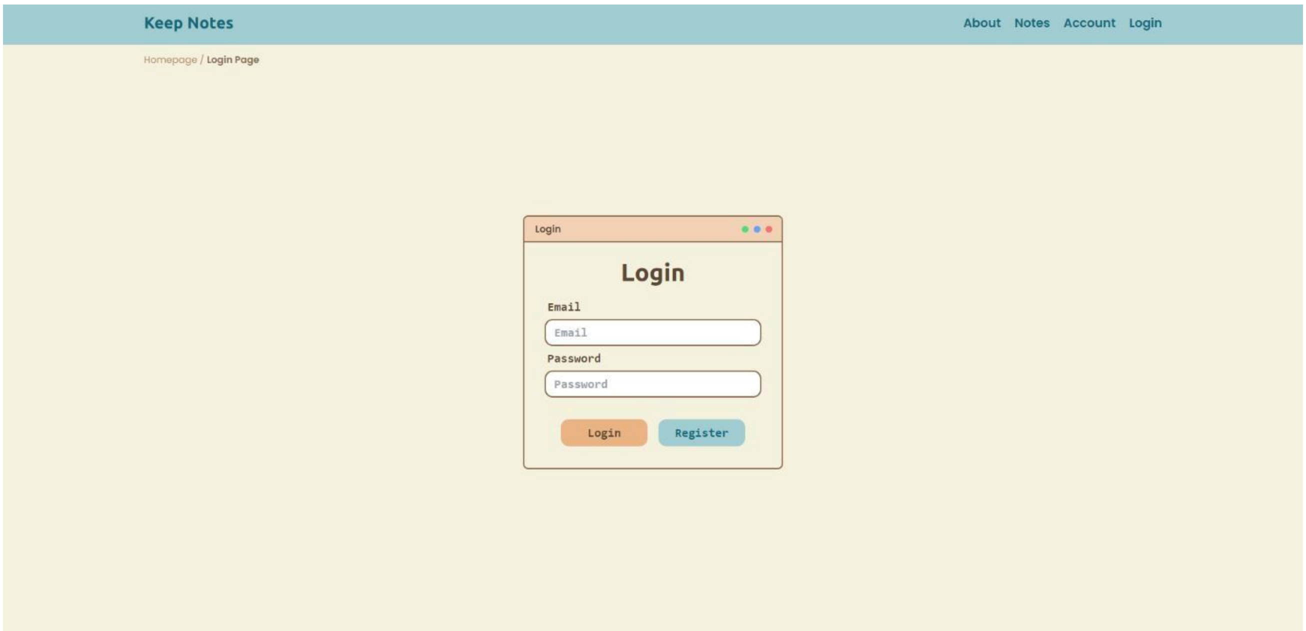
- **i.** user_id – uuid
- **ii.** user_name – varchar
- **iii.** user_email – varchar **iv.** password – varchar
- **v.** last_update – date
- **vi.** create_on – date

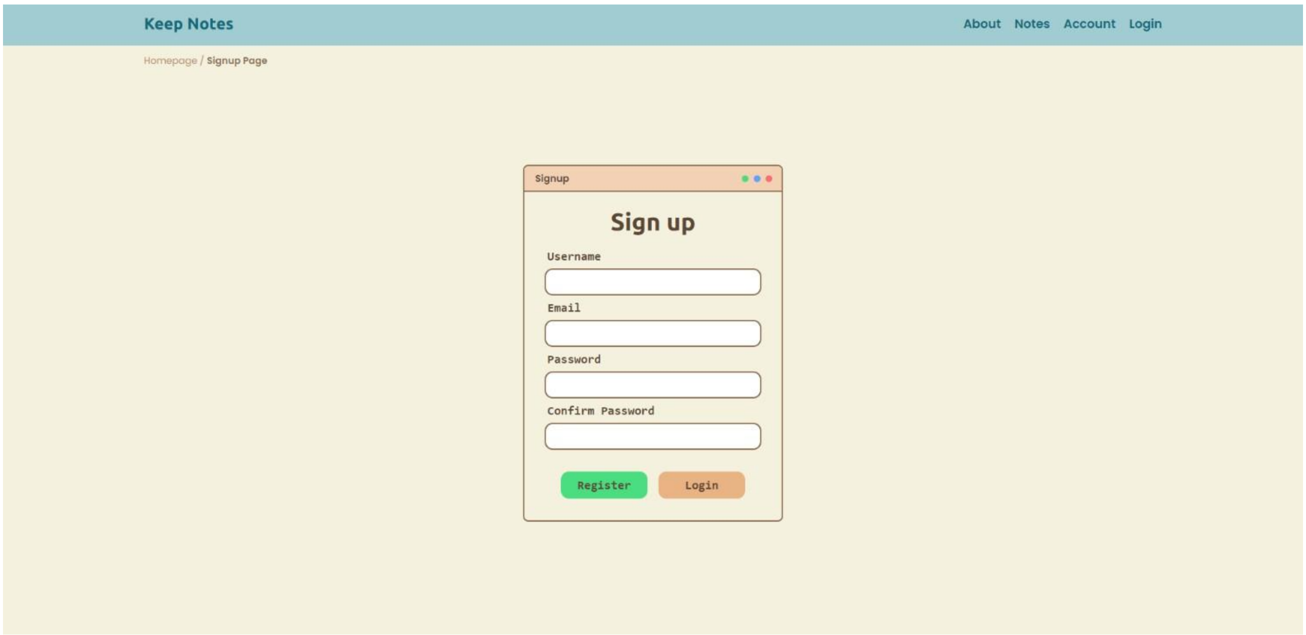
b. NOTES

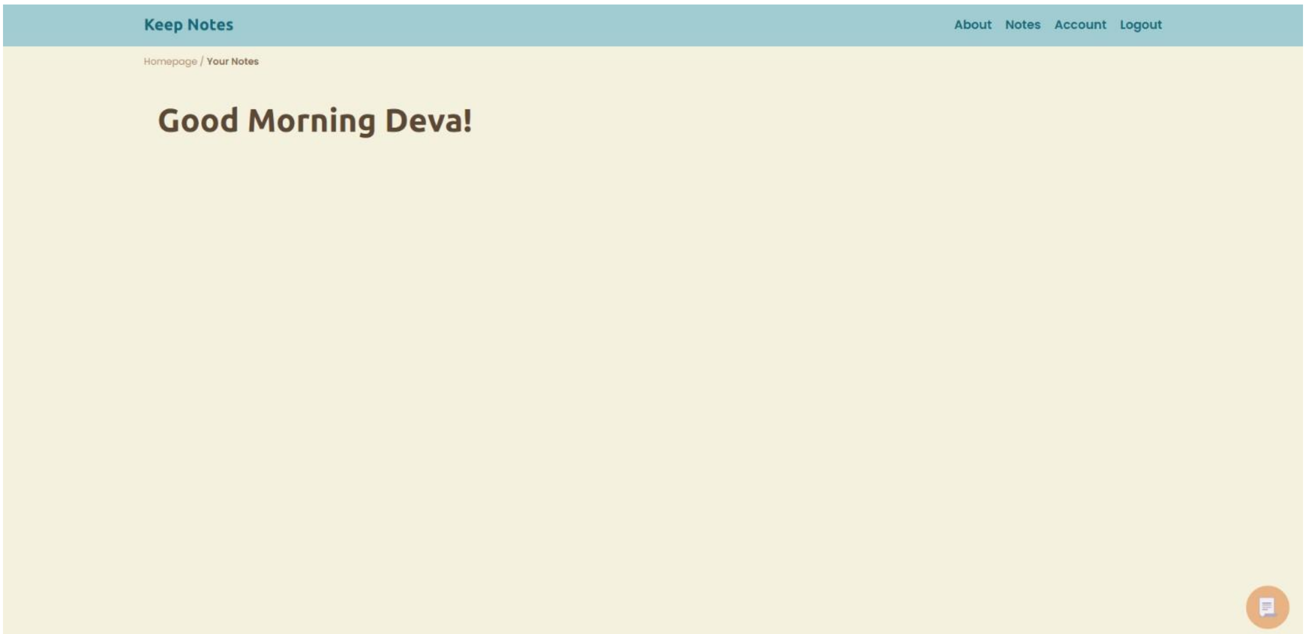
- **i.** note_id – uuid
- **ii.** note_title – varchar

iii. note_content – varchar **iv.** last_update – date
v. created_on – date

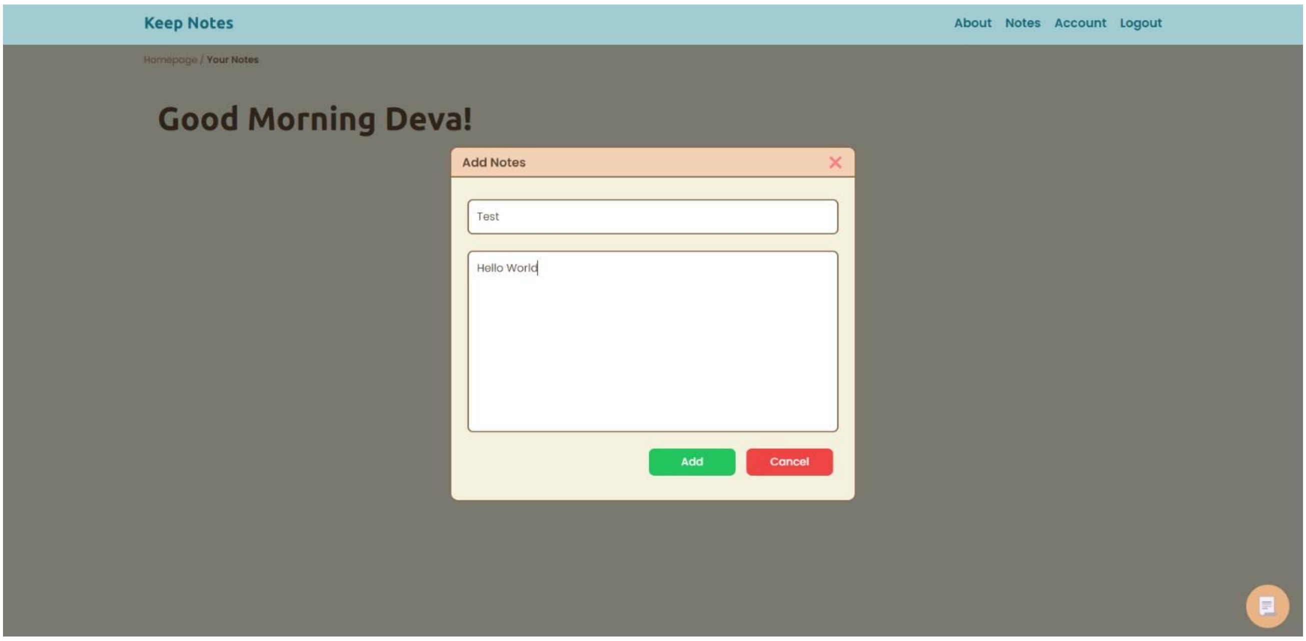
FRONTEND UI







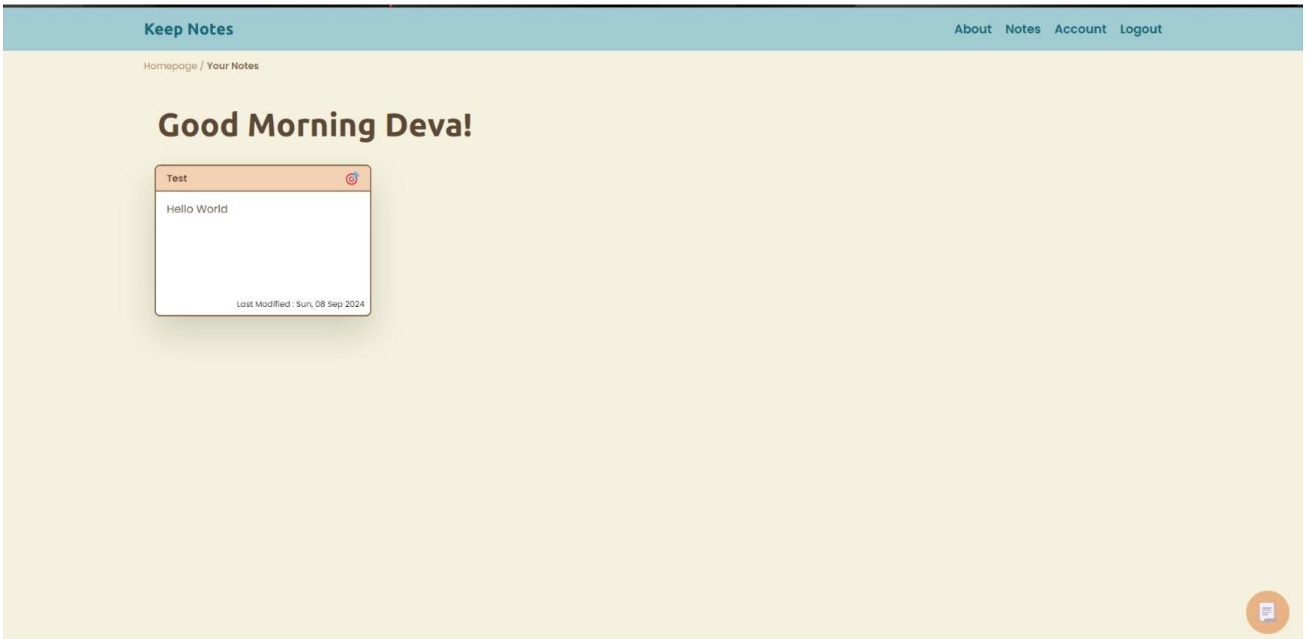
Good Morning Deva!



Good Morning Deva!

Test

Hello World

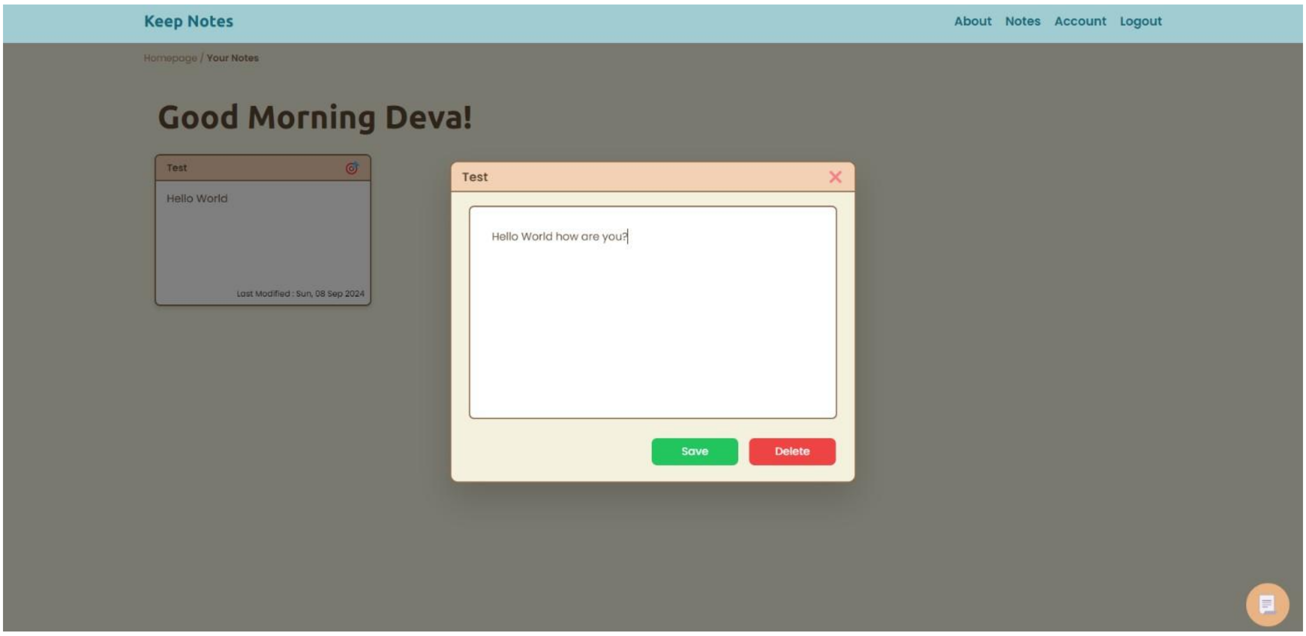


Good Morning Deva!

Test

Hello World

Last Modified : Sun, 08 Sep 2024



Good Morning Deva!

Test

Hello World

Last Modified : Sun, 08 Sep 2024

Test

Hello World how are you?