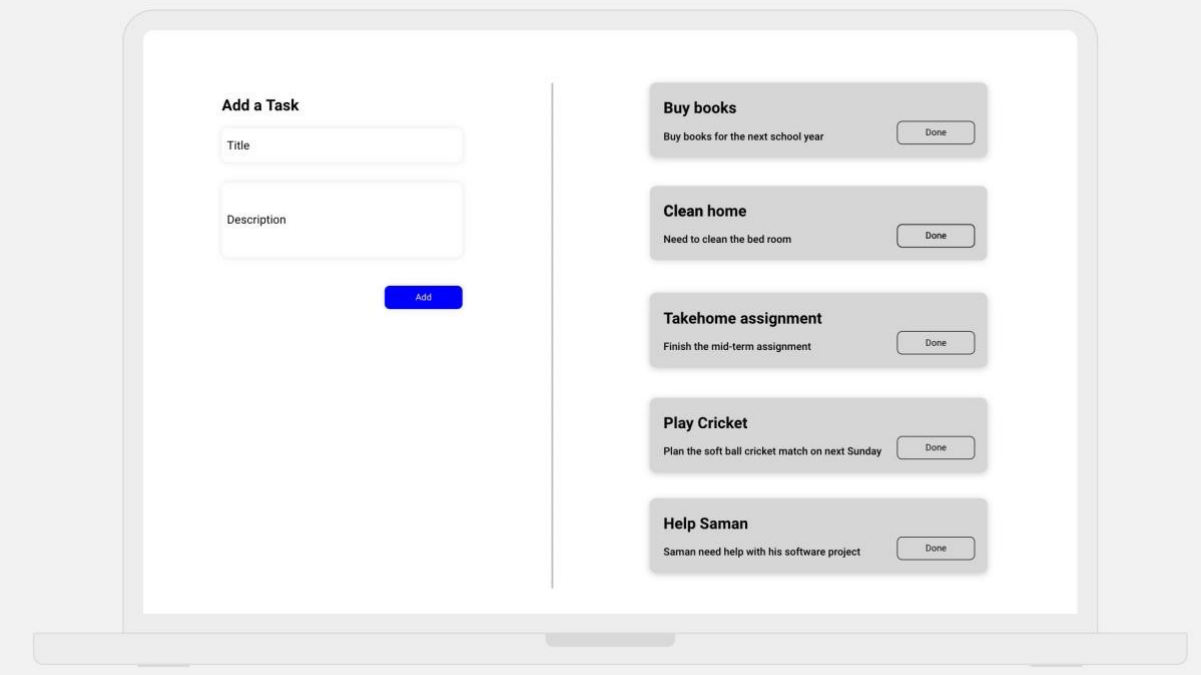
Full Stack Engineer/ Intern - Take Home Assessment

In this assessment you are going to build a small to-do task web application.

User Requirements

- End users should be able create to do tasks through the web UI, by providing a title and a description. Please refer to the UI mockup below.

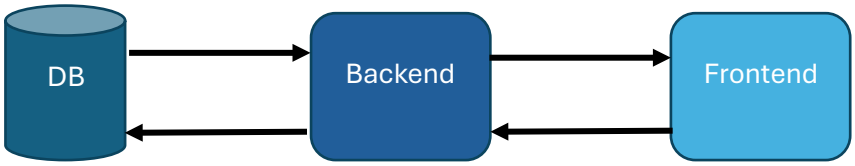


- Only the most recent 5 to-do tasks are listed in the UI. No need to list them all.
- Users can mark a task as completed by clicking on the “Done” button in each task card. A completed task should not be visible in the UI.

Architecture Requirements

- This system has 3 main components, the database, the backend API, and the frontend

UI.



- Todo tasks are stored in the database, under a table named “task”. You can decide on what columns should be in the “task” table.
- Backend should comprise with a REST API that facilitates the user operations. You can decide on how the REST API should be designed.
- Frontend should be a simple SPA.
- All components should be able to run using Docker. You can use `docker-compose` to start the 3 containers accordingly.
- Choosing the tech stack:
    - For the database you can choose any relational DBMS (like MySQL, MariaDB, PostgreSQL, Oracle DB, etc.). Don’t use NoSQL databases. ○ For the backend API implementation, you can choose a programming language like Java, JS/TS, PHP, C#, or Python. Feel free to use appropriate libraries and frameworks of the language you chose.
    - For the frontend UI implementation, you can use HTML, CSS, vanilla JS or a frontend framework like React, Vue, or Angular.
- If any of the components needs to be built before executing (e.g. you chose Java for the backend API, or chose TS for the frontend UI), then use Docker containers to run the build process. Only assume that the assessment evaluator has Docker and a Linux dev environment (with Bash and other GNU tools) in their local setup.

Evaluation Criteria

- Approach to the solution.
- Overall system architecture.
- Completion of the functionality requirements.
- Database design.
- Unit tests and integration tests for the backend. Test coverage will be evaluated.
- Unit tests for the frontend components if you are using a frontend framework to implement the UI.
- Following clean code principles and SOLID principles.

Extra Marks

- End-to-end tests.
- Pretty UI.

Submission

- Commit your code to a Git repository in GitHub and share the link to that public repository.
- Make sure to add a README file that clearly states the instructions to build and run the project.
- Assessment should be submitted within 3 days.

If you have any questions about this assessment, feel free to reach out to the recruiter that sent you this assessment.