

## AI Developer Technical Screening Exam

This exam is designed to test your proficiency in full-stack development using Next.js, Tailwind CSS, and REST APIs. You will be given a series of tasks that will require you to demonstrate your knowledge and understanding of these technologies, with focus on how you use Next.js effectively.

### Instructions

The goal is to develop a full-stack Task Manager application that allows users to efficiently manage their tasks, prioritize their work, and track progress. The application will consist of both front-end and back-end components, offering a seamless and user-friendly experience.

### References

- Next.js: <https://nextjs.org/docs>
- Tailwind CSS: <https://tailwindcss.com/docs/styling-with-utility-classes>

### Task 1: Set up the project

- Ensure you have *Node.js* installed.
- Set up the project by following the automatic installation guide from the official Next.js documentation: <https://nextjs.org/docs/app/getting-started/installation#automatic-installation>
- You can use either the **app router** or the **pages router**. But pages router is encouraged.

### Task 2: Create a login page with authentication

- Ensure the app can only be accessed by authenticated users.
- You are free to use simple JavaScript or authentication libraries.

### Task 3: Display all task on the Home page

- Create a home page that displays all tasks. You can use server actions OR APIs to fetch this data.

- The fetched tasks should be organized and presented according to their status: **To Do**, **In Progress**, and **Completed**. There should be a clear distinction of these categories in the UI.
- Each task card should have functional **edit** and **delete** buttons.
- Each task should be clickable and viewable. Upon clicking, it should redirect to a dedicated page where the specific task can be viewed in detail.
- The user should be able to add a new task from this page. Add a button that redirects to **Create Task** page.
- You are free to layout this page however you like.

#### Task 4: Add a Create Task page

- This page must have a form accepting the following input:
  - Id
  - Title
  - Description
  - Due Date
  - Priority
  - Status
  - Other fields you wish to add.
- After creation, the user should be redirected back to home page.

#### Task 5: Display a specific task on the Task page.

- The task details are displayed here.
- Enable users to **edit** or **delete** a task from this page.

#### Task 6: Use Next.js Layout and Reusable components

- Create a global layout that can be used for all pages in the project. Include a navigation bar that is present on all pages. Additionally, create reusable components as necessary for the project.
- You can add additional pages if desired.

## Rules

- Your project should make use of server actions (or simply, server functions) and API routes. The decision of when and where to use them in your project is up to you, but make sure to use both in this project.
- Creating a database is not required; alternatively, you may utilize the JavaScript Web Storage API or any JavaScript logic.
- You will primarily use *Tailwind* CSS for styling, but minimal custom CSS/SCSS may also be used if needed.
- Your final output should be **mobile responsive**.
- You have the creative freedom to design and present this project in your own style, while ensuring all specified requirements are met.

Once finished, please send the zipped copy of the project. Ensure that you delete the '**node\_modules**' folder before zipping it.