# Simple Take-Home Assignment for Interns

> [!IMPORTANT] This assignment is **NOT** a group assignment, everything should be completed by yourself except the PR review work.

> [!CAUTION] As a programmer, you have to understand **EVERY LINE** of **TypeScript** codes in your repo, you may **FAIL** this assignment if you cannot explain the codes when we ask you (**we may ask any line of codes**).
>
> AI tools (e.g Github Copilot, ChatGPT) are **ALLOWED**, but again, it's your responsibility to explain the codes generated from AI tools.
>
> Discussing technical problems with other interns are also **ALLOWED**, but your repo should only contains your own git commits.

## Project Name

Personal Task Manager App

## Objective

Develop a simple React Native mobile application using `Expo` and `TypeScript` that functions as a personal task manager. This assignment aims to assess your skills in React Native development, your understanding of `TypeScript`, and your ability to use `Git` and `GitHub` for proper collaboration.

> [!IMPORTANT] You are free to design your own UI for this app, don't worry if it is ugly or not, you are good to go **as long as it is functional**.

## Project Overview

Create a mobile app with the following features in 2 weeks:

1. **Task list:**
   - Display a list of tasks using hard-coded mock data.
   - Each task should include a title, description, and status (e.g., pending, completed).
2. **Add new tasks:**
   - Provide a form to add a new task to the list.
   - The new task should update the state and display in the task list.
3. **Edit tasks:**
   - Allow users to edit an existing task's title and description.
4. **Delete tasks:**
   - Enable users to remove a task from the list.
5. **Toggle the task status:**
   - Users can mark a task as completed or pending by toggling its status.
6. **Task details screen:**
   - When a task is selected, navigate to a details screen showing the necessary information about the task.
7. **Search Functionality (optional but big plus):**
   - Implement a search bar to filter tasks by title.

## Technical Requirements

### 1. React Native & Expo

- Use the latest stable versions of **React Native** and **Expo**.
- Ensure the app runs smoothly on both **Android** and **iOS** devices.
- Utilize **Expo Router** for navigating between screens.

### 3. State Management

- Use **React hooks** (e.g., `useState`, `useEffect`) for state management.
- No need for external state management libraries like Redux.

### 3. Mock Data

- Use hard-coded mock data for the initial list of tasks.
- Backend API serving, authentication or database integration are not required.

### 4. Code Quality

- **TypeScript Typing:**

  - Define interfaces or types for your task data and component props.
  - Use appropriate typing for components, props, state, and functions.
  - Ensure all components and functions are properly typed.
  - `any` type is prohibited unless you have a strong justification.

- **Clean Code Practices:**

  - Follow industry standards for code formatting and organization.
  - Use meaningful variable and function names.
  - Keep components focused and reusable where possible.

- **Comments:**

  - Add comments to explain complex logic or important sections of code.
  - Use inline comments for clarity where necessary.
  - You do not need to add comments for every function, do it when you feel it is necessary.
  - Well explain your codes by using better function, variable and interface names.

### 5. Documentation

- **README.md:**
  - Provide an overview of the project.
  - Include instructions on how to set up and run the app.

### 6. Git & GitHub Collaboration

- **Repository Setup:**

  - Create a new **GitHub repository** for the project in your own Github account.
  - Clone this repo into your local environment and start to work on it.

- **Branch Management:**

  - Use the `main` branch for stable code.
  - Create feature branches (e.g., `feature/add-task`, `feature/edit-task`) for new features.

- **Commits & Pull Requests:**

  - Invite at least one other dev intern as your peer reviewer to review your PRs (at least review one PR).
  - Write clear and descriptive commit messages.
  - Open pull requests when merging feature branches into `main`.
  - Review at least one other dev intern's PR.

- **Issue Tracking:**

  - Use GitHub Issues to document bugs, tasks, and enhancements.

## Submission

- **Final Checks:**

  - Test the app on both Android and iOS simulators or devices.
  - Ensure all features are working as expected.
  - Merge all feature branches into `main`.
  - Make sure your repo is public that can be viewed by anyone, otherwise mentors cannot evaluate your work.

- **Submit:**

  - Send an email to together-submission@1thing.org, include the below information in the email:
    1. You full name.
    2. Your Github repo URL.

## Evaluation Criteria

1. **Functionality (40%):**

   - The app meets all specified requirements.
   - User experience is smooth and free of major bugs.

2. **Code Quality (25%):**

   - Code is clean, well-organized, and follows best practices.
   - Proper use of `TypeScript` for typing.
   - Appropriate use of React Native components and hooks.

3. **Collaboration (15%):**

   - Effective use of Git and GitHub for collaboration.
   - Evidence of code reviews.

4. **Use of `TypeScript` (10%):**

   - Correct and efficient use of `TypeScript` features.
   - Interfaces, functions and variables are well-defined and used appropriately.

5. **Documentation (10%):**

5. **Documentation (10%):**
    - Clear and comprehensive README.
    - Code is well-commented where necessary.

## Additional Guidelines

- **Time Management:**
    - Allocate time wisely to complete the assignment within the given timeframe.
    - Prioritize core features before optional enhancements.
- **Learning Resources:**
    - React Native Documentation
    - Expo Documentation
    - TypeScript Handbook
    - Git & Github Bascis
    - GitHub Guides

## Example Questions We May Ask You

- What does this line mean?
- Why you want to write the code block like this?
- Can you explain this function? What is it used for? How do you know it can work as expected?

## Support

If you have any questions or need clarification on the assignment, please contact our mentors at intern-support@1thing.org.

---

**We look forward to seeing your skills in action and how you collaborate effectively. Good luck!**