

Project TODOs

Project #8675309 Assigned: Bob (NY)

Department IT/Engineering/Development

Due Date: ASAP (delayed from: *yesterday*)

Requirements Meeting: TBD (pending: requirements planning meeting)

Any questions should be directed to the project manager: donotreply@bigcompany.com

Functional Requirements

Add Todo:

- Users can enter a todo item in an input field.
- A button labeled "Add Todo" will trigger the addition of the todo. **(Pending marketing on verbiage)**
- Each todo should have a unique ID.

View Todos:

- Todos will be displayed in a list format.
- Each todo item should display the text and have a checkbox to mark it as complete.

Mark as Complete:

- Users can mark a todo as complete by checking the checkbox.
- Completed todos will have a visual indicator (e.g., strikethrough text). **(pending UI/UX designer on line color)**

Delete Todo:

- Users can delete a todo item using a delete button next to each todo. ~~items should be archived on Amazon S3~~
- Upon deletion, the todo should be removed from the list.

Persist Data:

- Todos will be stored in local storage to retain the list across page reloads. **(database/backend under development for Q4)**
- Data will be serialized to JSON format for storage and deserialized upon retrieval. **(question: YAML?)**

Component Structure:

- Use of functional components with hooks for state management. **(see dev style guide, somewhere on the intranet)**
- A custom hook to handle local storage interactions.

- A Higher Order Component (HOC) to wrap the main todo component for added functionality, such as logging.

Styling:

- Styled Components will be used to create a responsive and appealing UI. **(until in-house UI library is completed)**
- Each component should have its own styles encapsulated.

Testing:

- Each component should have corresponding unit tests to ensure functionality. **(question: cut testing for timeline?)**
- Integration tests *should* verify the interaction between components.

Non-Functional Requirements:

- The application should have a user-friendly interface. (big text with lots of emojis 😊)
- It should load quickly and be responsive across different ALL screen sizes on ALL devices on ALL platforms ALL the time **(should be easy just different sizes of screens, no?)**.
- Code should follow best practices for readability and maintainability. *(NOTE: this will all go out the door once the project deadline gets cut short)*