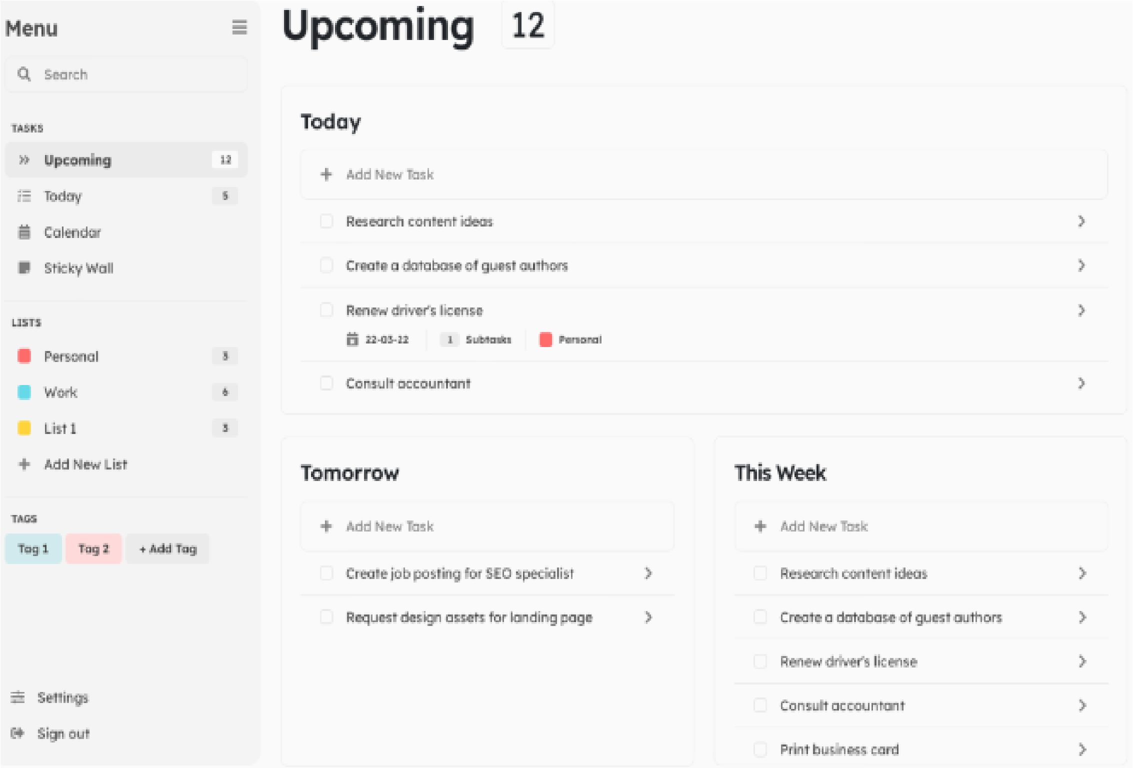
Group Project Final Proposal

Chris McCorkle, Elian Zamora

1. Detailed Plan Documentation
    - List of React Components
        - **React** - Front end dev and handling the logic behind the tasks and allowing users to add and delete new tasks as well as edit tasks as needed.
        - **BootStrap** - Used for creating a seamless front end project development to have more control over the appearance and components in the app.
    - Functions and Logic
        - **Adding** - Function will add a new task and display to the user
        - **Deleting** - Function will delete a task when completed as marked by the user
        - **Editing** - Function will allow a task to be edited by the user if a certain task is wanted to be replaced by a different one
        - **Single state-lifted master list** - This will allow the site to keep track of the user's specific tasks and act as their list of tasks. By using a single state-lifted 'database' we can easily add/remove and update tasks as deemed by the end user
    - API Details
        - **GET** - Used for updating / retrieving the current data the user will need in regards to their task lists
        - **PUT / POST** - Used for adding a new resource to the task list, or updating a task that is already inside of the list
        - **DELETE** - Used for when a user would like to remove an item from the list
        - Along with the API requests, our system will respond to these using the status codes, (200, 400, etc…) to provide feedback to the user when they make changes or modify the list in any way, or if something goes wrong an error message will be displayed.

- Endpoints
    - **‘/deleteTask’** - This endpoint will be used when a user wants to delete a specific task. Data passed to this endpoint will be the unique identifier (key) uuid4 value that is associated with the task
    - **‘/createTask’** - This endpoint will be used when a user wants to add a new task to their list. Data passed will be the user's task name, a description, any data associated with the task; and the function will automatically assign a unique ID to the task and update the list.
    - **‘/updateTask’** - This endpoint will be utilized when a user wants to edit an existing task. Data passed here will be the changes they want to make to the task (such as changing the name). As the task is already created, the key ID and the unchanged data will not need to be passed
- Design
    - Inspiration



    - A design like this would allow users to be able to clearly see their tasks in a neat and concise way. This much detail is out of our scope, however, the general format and display is our goal. We would want to create a user friendly design that allows the user to easily add, delete, and edit the tasks in a way that is quick and efficient.