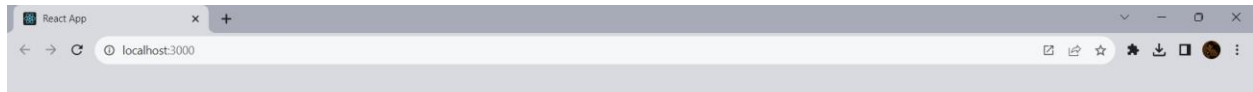


## TODO-LIST BY YASH VERMA



# TODO-LIST

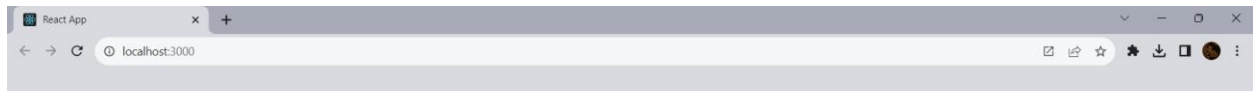
Enter new Todo...

ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS

- new
- Yash Make Todo-App



# TODO-LIST

Enter new Todo...

ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS

- new
- Yash Make Todo-App





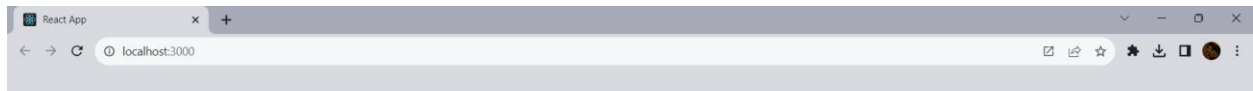
# TODO-LIST

Enter new Todo...

ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS



# TODO-LIST

Enter new Todo...

ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS

Remove Done Todos

- new
- Yash Make Todo-App





# TODO-LIST

Enter new Todo...

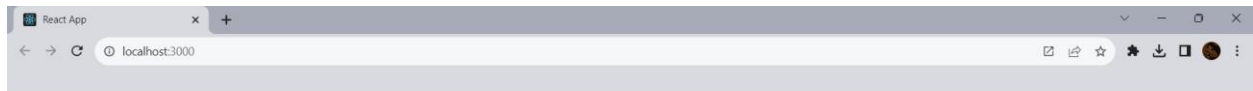
ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS

Remove Done Todos

- New-Task



# TODO-LIST

Enter new Todo...

ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS

Remove Done Todos

- New-Task



- new



- Yash Make Todo-App





# TODO-LIST

Enter new Todo...

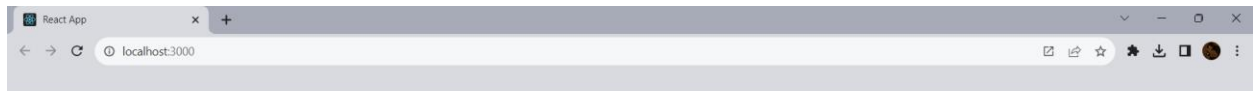
ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS

Remove Done Todos

- New-Task
- new
- Yash Make Todo-App



# TODO-LIST

Enter new Todo...

ALL\_TODOS

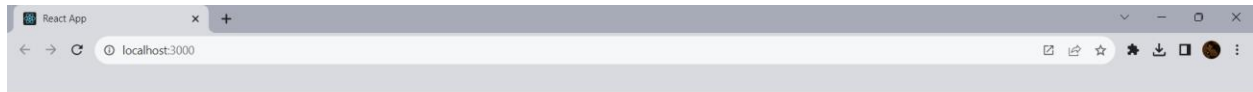
ACTIVE\_TODOS

DONE\_TODOS

Remove Done Todos

- Yash Verma Todo List task
- new
- Yash Make Todo-App





# TODO-LIST

Enter new Todo...

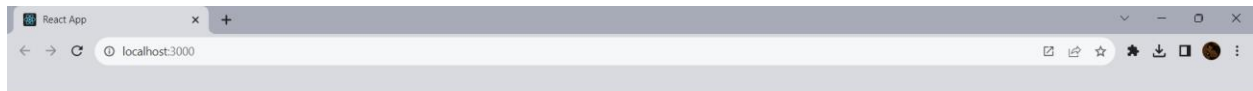
ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS

Remove Done Todos

- Yash Verma Todo List task
- new
- Yash Make Todo-App



# TODO-LIST

Enter new Todo...

ALL\_TODOS

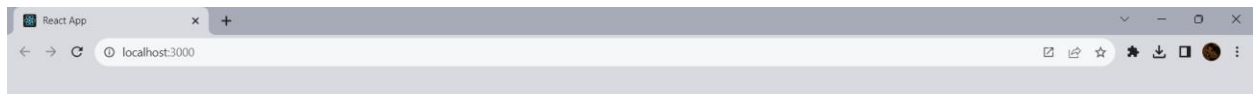
ACTIVE\_TODOS

DONE\_TODOS

Remove Done Todos

- new
- Yash Make Todo-App





# TODO-LIST

Enter new Todo...

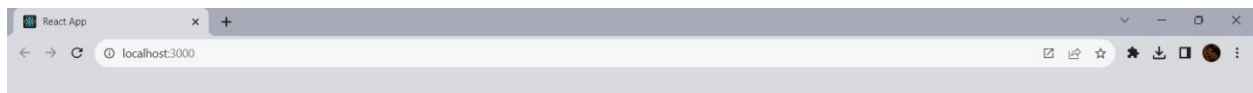
ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS

Remove Done Todos

- new



# TODO-LIST

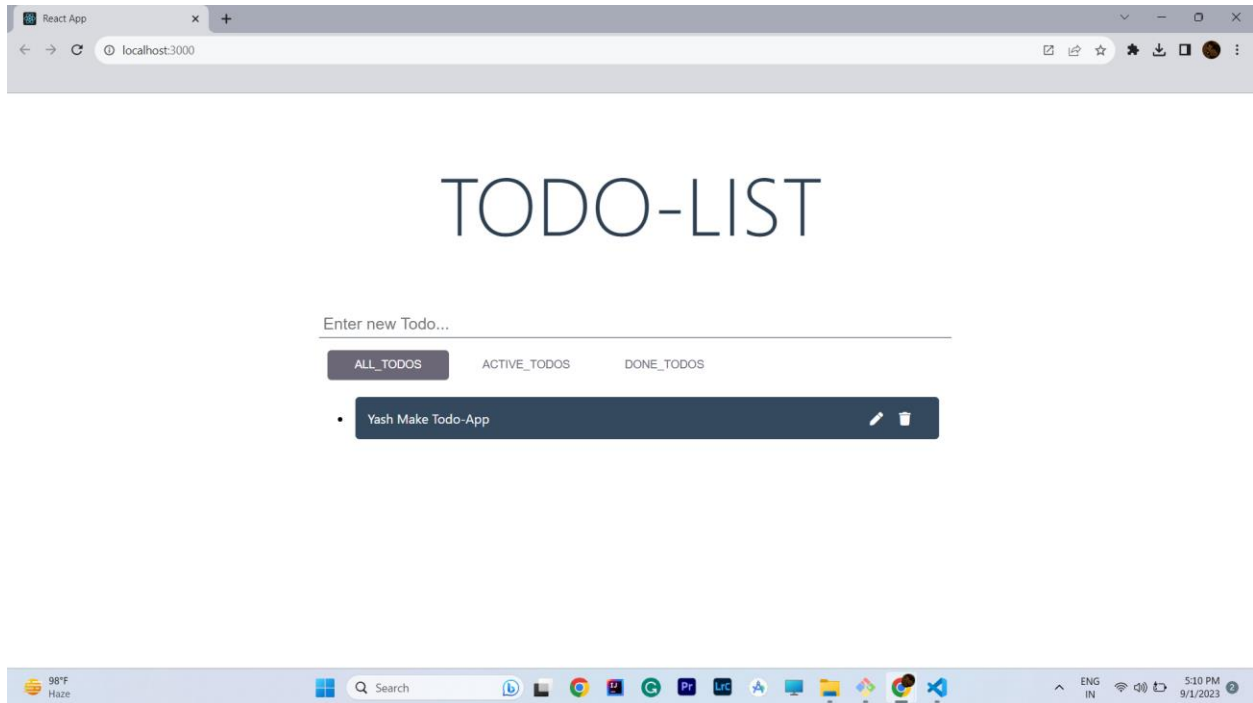
Enter new Todo...

ALL\_TODOS

ACTIVE\_TODOS

DONE\_TODOS





**P**roject is a comprehensive task management web application that combines a React frontend,

Node.js backend, and MongoDB database to provide users with an efficient and organized way to manage their tasks. With three distinct tabs and various features, it offers a user-friendly experience for task management.

**\*\*Technology Stack\*\***:

- **\*\*Frontend\*\***: The frontend of your application is built using React, a popular JavaScript library for building user interfaces. React allows you to create a dynamic and responsive user interface.

- **\*\*Backend\*\***: Node.js powers the backend of your application. It's an excellent choice for building server-side applications due to its non-blocking, event-driven architecture, making it efficient for handling requests.

- **Database**: MongoDB serves as the database for your application. MongoDB is a NoSQL database that is well-suited for handling data that doesn't fit neatly into a traditional relational database structure, making it ideal for storing and managing tasks.

### **Tab-Based Task Management**:

Your application features three tabs that cater to different aspects of task management:

1. **All Todos**: This tab provides users with an overview of all their tasks, both active and completed. It serves as a comprehensive list of tasks.

2. **Active Todos**: In this tab, users can view only the tasks that are currently active, meaning they haven't been marked as completed.

3. **Done Todos**: The "Done Todos" tab allows users to focus solely on completed tasks, helping them keep track of their accomplishments.

### **Task List and Actions**:

Within each tab, there is a list of tasks, with each task represented as an item. Each task item typically includes details such as the task's title, description, and completion status. Additionally, there are buttons associated with each task for performing actions:

- **Deletion**: Users can delete specific tasks by clicking a "Delete" button next to each task. This action removes the task from the list.

- **Updating**: Users have the option to update task details by clicking an "Update" or "Edit" button. This functionality allows them to modify the title or description of a task.

### **Bulk Deletion of Completed Tasks**:



One notable feature is the presence of a special button that appears dynamically when there are completed tasks in the "Done Todos" tab. This button offers users the convenience of deleting all completed tasks at once with a single click. It streamlines the process of housekeeping and decluttering the completed tasks list.

### **\*\*Database Integration\*\*:**

MongoDB plays a crucial role in your project, acting as the backend data store. Each task that users create corresponds to a document in the MongoDB collection. This NoSQL database's flexibility allows you to store and manage task data efficiently without the rigid structure of traditional SQL databases.

### **\*\*User Interaction and Experience\*\*:**

Your application is designed with user interaction in mind:

- Users can easily add new tasks, providing a seamless experience for creating to-do items.
- The ability to mark tasks as "done" or "active" allows users to organize their tasks according to their current status.
- Editing existing tasks offers flexibility and ensures that users can keep their task descriptions up to date.
- The presence of buttons for deletion and the special bulk deletion button streamlines task management.
- The use of tabs simplifies navigation and allows users to focus on specific subsets of their tasks, enhancing the overall user experience.

### **\*\*Conclusion\*\*:**

In summary, your React project with a Node.js backend and MongoDB database is a powerful and user-friendly task management application. It excels in providing a structured way for users to manage their

tasks, with features such as tab-based categorization, task list actions, and convenient bulk deletion. The combination of these technologies and features makes it a valuable tool for individuals seeking to maintain an organized and efficient to-do list.