Documentation

Overall Purpose of the Project

To build a full-stack task execution system that:

- Allows users to **create, manage, and run tasks** (defined by shell commands).
- **Tracks** each task's execution history (start time, end time, output).
- Provides a **user-friendly frontend** to interact with tasks.
- Uses a **MongoDB backend** to store task definitions and execution logs.
- Automates command-line operations via a web interface, enabling technical users (like DevOps or engineers) to **execute scripts without needing terminal access**.

Functional Goals by Layer

Frontend (React + Ant Design)

- UI for creating, listing, and deleting tasks.
- Search feature to find tasks by name.
- "Run" button to execute a task.
- Modal popup to view execution logs.
- Makes HTTP requests to the backend using `axios`.

Real-World Use Cases

- **DevOps Automation:** Run deployment, backup, or cleanup scripts from a web interface.
- **Admin Tooling:** Give operations teams the ability to execute predefined tasks without needing direct terminal access.
- **Remote Execution Dashboard:** Useful in cases where tasks must be managed or run remotely (e.g., servers, CI/CD scripts).
- **Education or Demonstration:** Useful for teaching how command-line tasks can be integrated into modern web systems.

Task-3

Frontend

Task Management API Documentation

Base URL

http://localhost:8080/tasks

Endpoints

1. Get All Tasks

GET /tasks

Description: Retrieve a list of all tasks.

Response:


```
[
  {
    "id": "task1",
    "name": "Build Project",
    "owner": "Alice",
    "command": "mvn clean install",
    "taskExecutions": []
  }
]
```


2. Get Task by ID

GET /tasks/{id}

Description: Fetch a specific task by its ID.

Example Request:
GET /tasks/task1

Response:


```
{
  "id": "task1",
  "name": "Build Project",
  "owner": "Alice",
  "command": "mvn clean install",
  "taskExecutions": []
}
```

3. Search Tasks by Name

GET /tasks/search?name={query}

Description: Search for tasks containing the given query in their name.

Example Request:
GET /tasks/search?name=Build

Response:

```
[
  {
    "id": "task1",
    "name": "Build Project",
    "owner": "Alice",
    "command": "mvn clean install"
  }
]
```

4. Create or Update Task

PUT /tasks

Description: Create a new task or update an existing task.

Request Body:

```
{
  "id": "task2",
  "name": "Run Tests",
  "owner": "Bob",
  "command": "npm test"
}
```

Response:

```
{
  "id": "task2",
  "name": "Run Tests",
  "owner": "Bob",
  "command": "npm test"
}
```

! Ensure that the controller method is using @PutMapping and accepts @RequestBody.

5. Delete Task

DELETE /tasks/{id}

Description: Delete a task by its ID.

Example Request:
DELETE /tasks/task2

Response:


```
{
  "message": "Task deleted successfully"
}
```


6. Execute Task

PUT /tasks/{id}/execute

Description: Triggers the execution of a task. Stores execution history with timestamp and output.

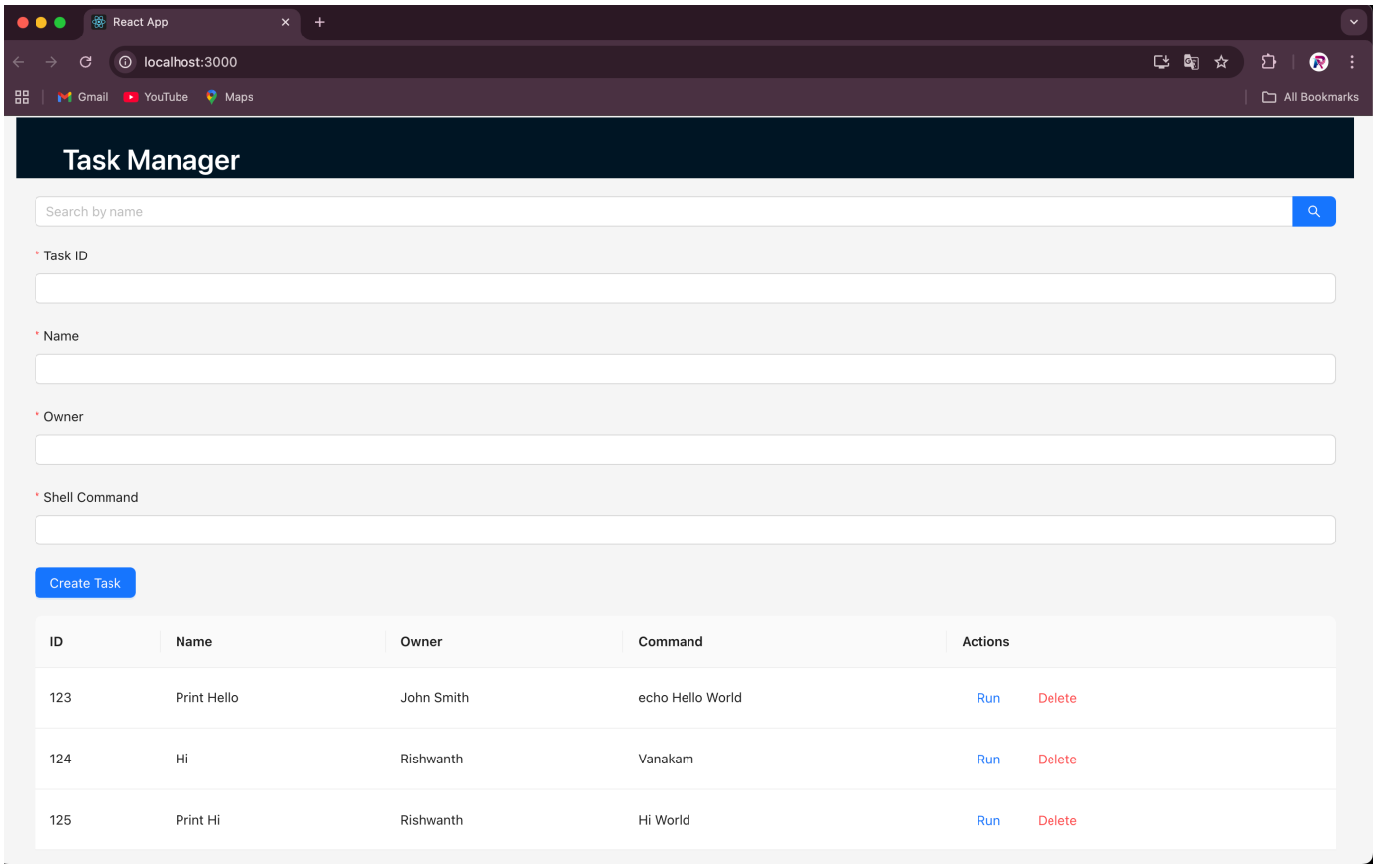
Example Request:
PUT /tasks/task1/execute

Response:

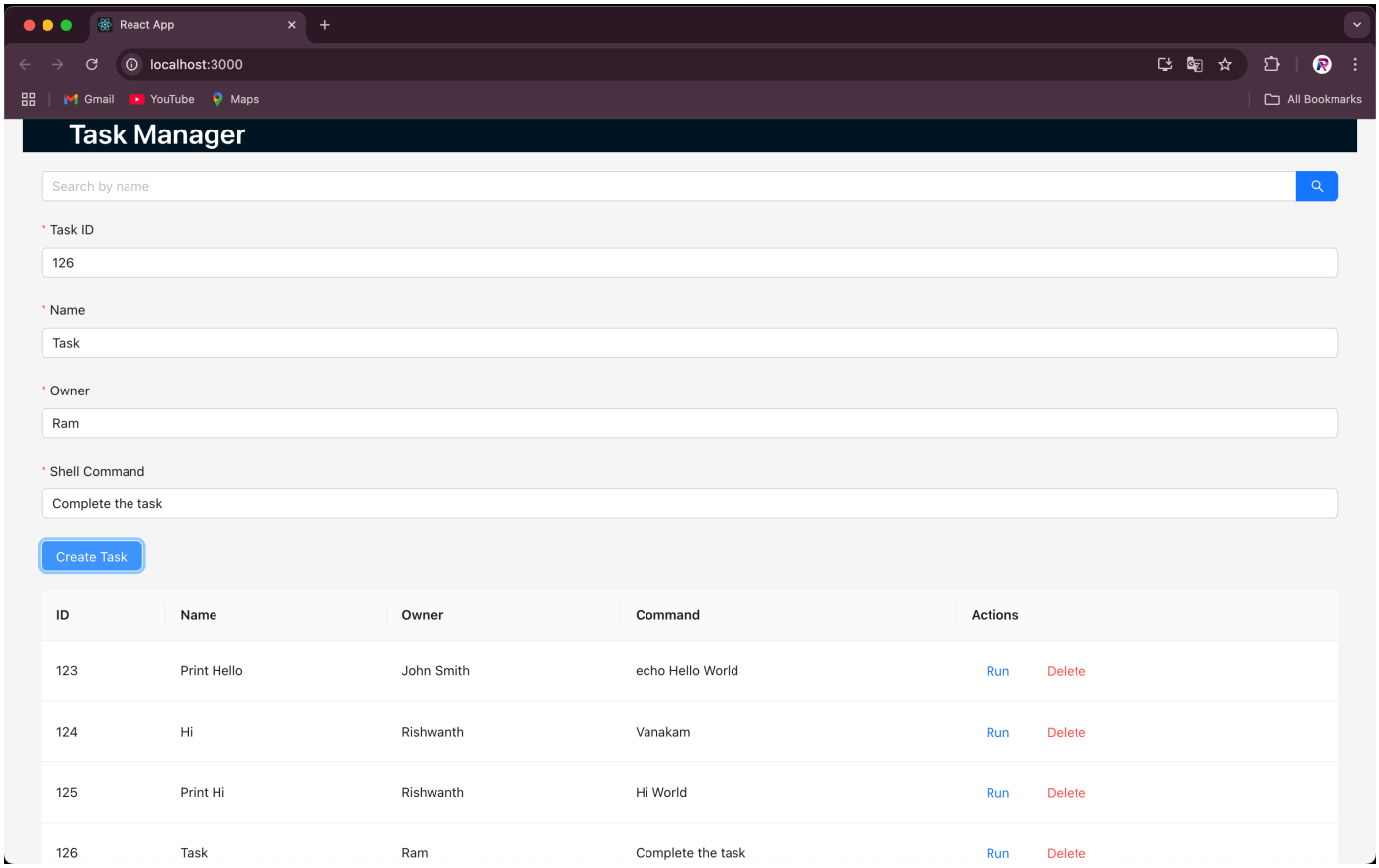

```
{
  "startTime": "2025-06-11T13:34:00",
  "endTime": "2025-06-11T13:34:02",
  "output": "BUILD SUCCESS"
}
```

Images:

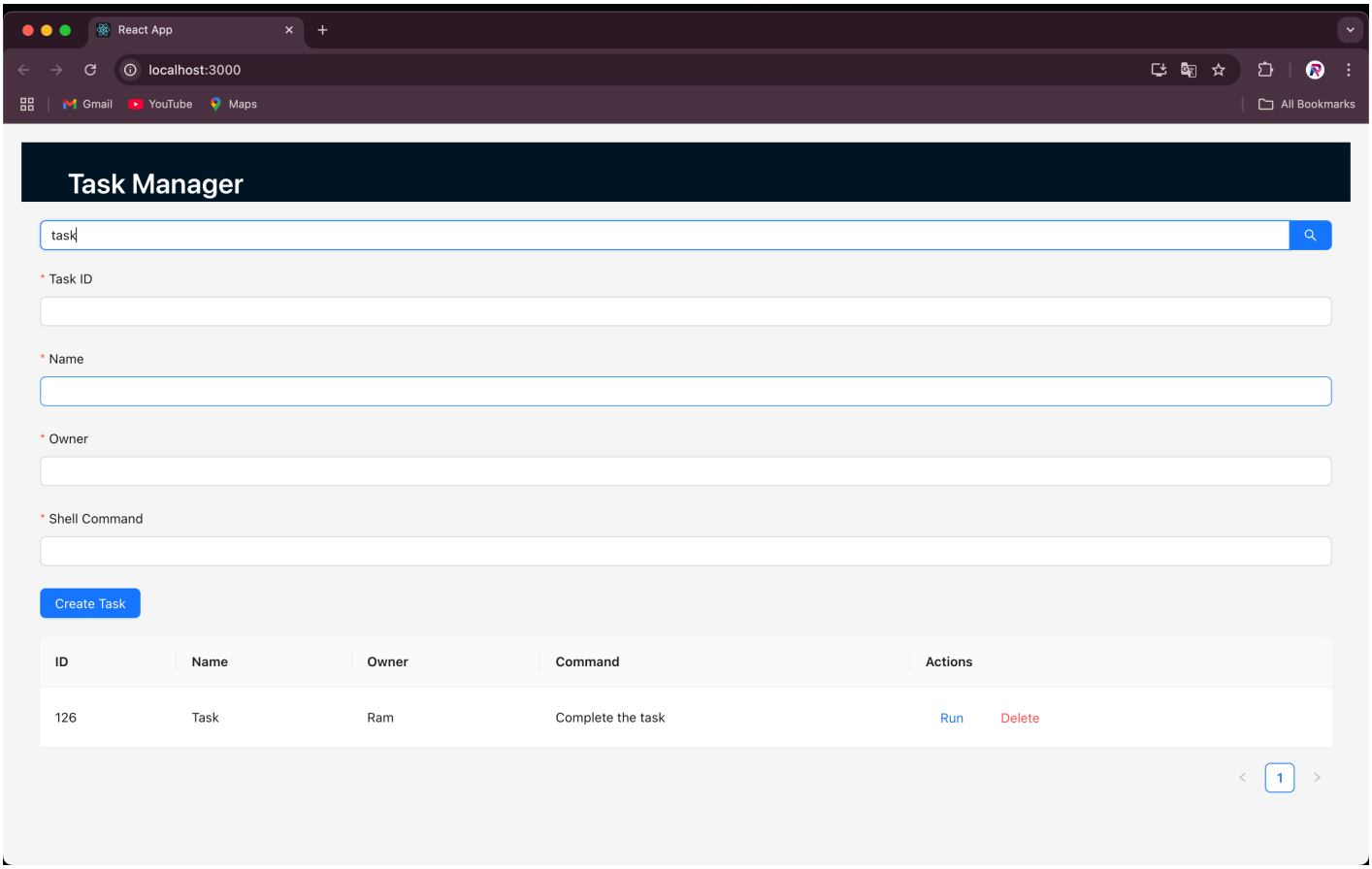
1. Web page



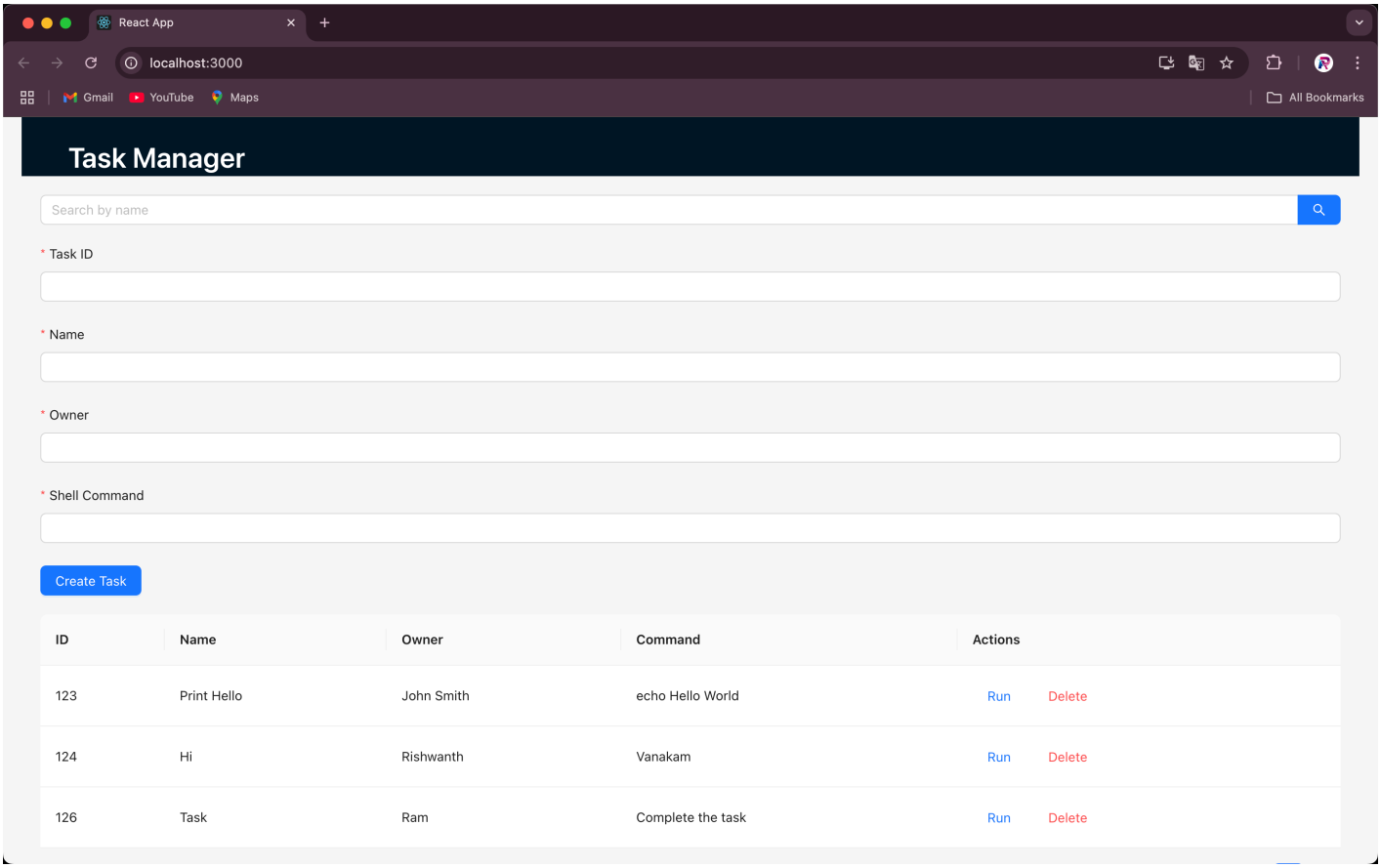
2. Creating a new task



3. Searching for a Task by name



4. Deleting a Task Id- 125



NPM/Yarn Dependencies

1. **React and React DOM.**
   These are essential for any React project.
   ```
   npm install react react-dom
   ```

2. **`Ant Design`**
   `is used for UI components (Form, Input, Layout, Table, etc.).`
   ```
   npm install antd
   ```

3. **dayjs**
   Used for formatting date and time in the execution history.
   ```
   npm install dayjs
   ```

4. **Axios**
   Needed for API communication (`getTasks`, `createTask`, etc. seem to use Axios).
   ```
   npm install axios
   ```

Optional (if not already configured)

5. **React Scripts (for Create React App)**
   If you're using Create React App:
   ```
   npm install react-scripts
   ```

6. **TypeScript**
   If you're using TypeScript instead of JS (not required for the file you uploaded):
   ```
   npm install typescript @types/react @types/react-dom
   ```