Activity 1: Todo List App

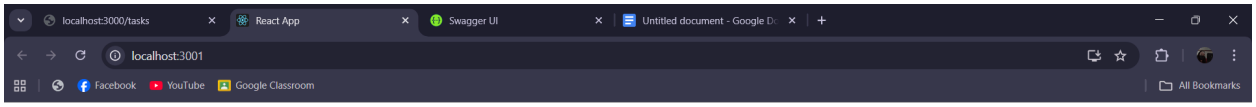
GROUP MEMBERS:
Abelita, Czyrell Gwen R.
Aranzasu, Darryl Jedidiah A.
Aranzasu Darrylene Kaela A.
Gangan, Demrose Carla C.
Moral, Andrie B.
Relox, Nathaniel

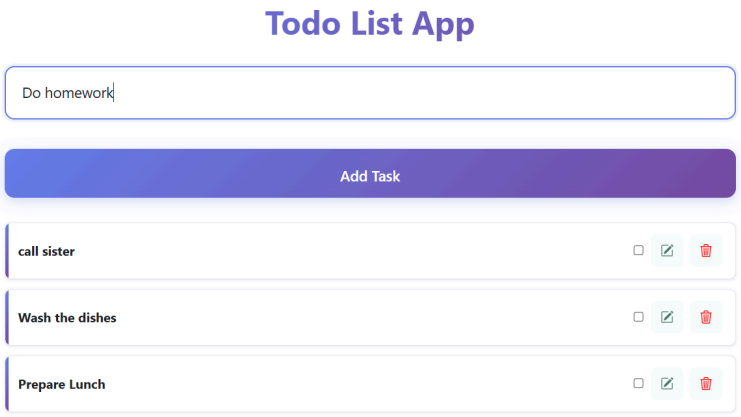
Activity 1: Todo List App

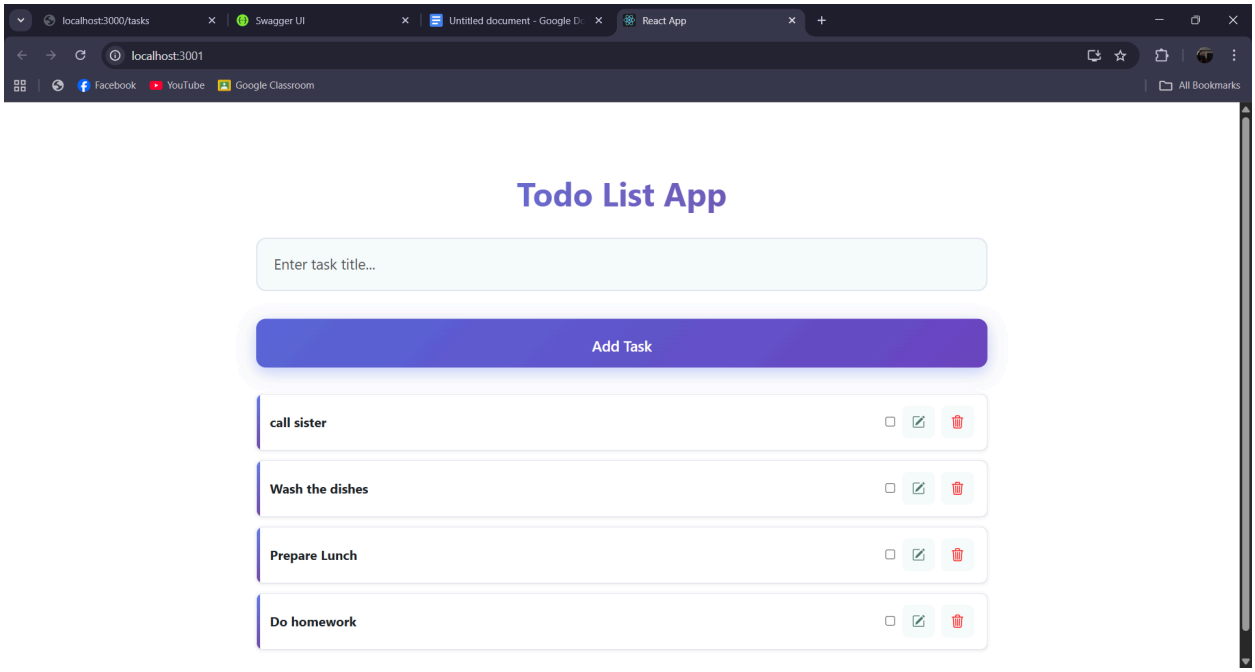
The todo list app allows users to manage their daily tasks. Users can create tasks, update the existing tasks and delete tasks once they're done doing it. It helps to organize work and keep track of things that they need to complete during the whole day. The interface is user-friendly so anyone can use it without difficulty.

Screenshot(s) of working System (UI + API example)

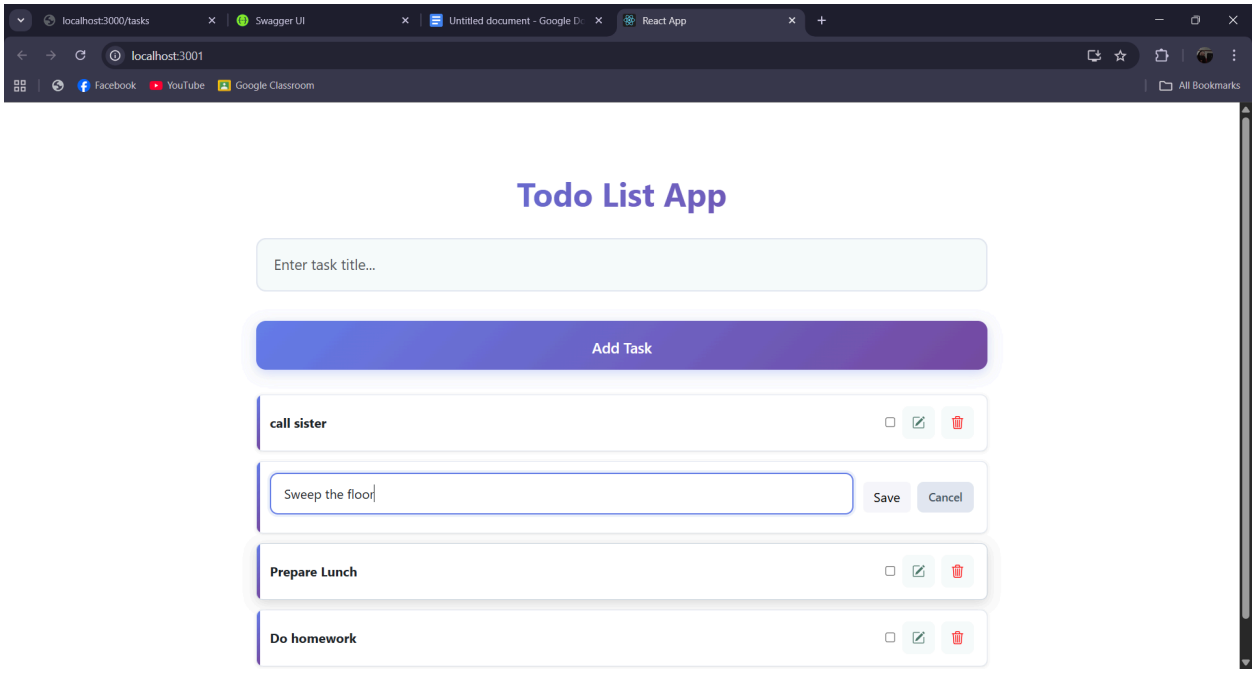
UI:
Adding a Task:

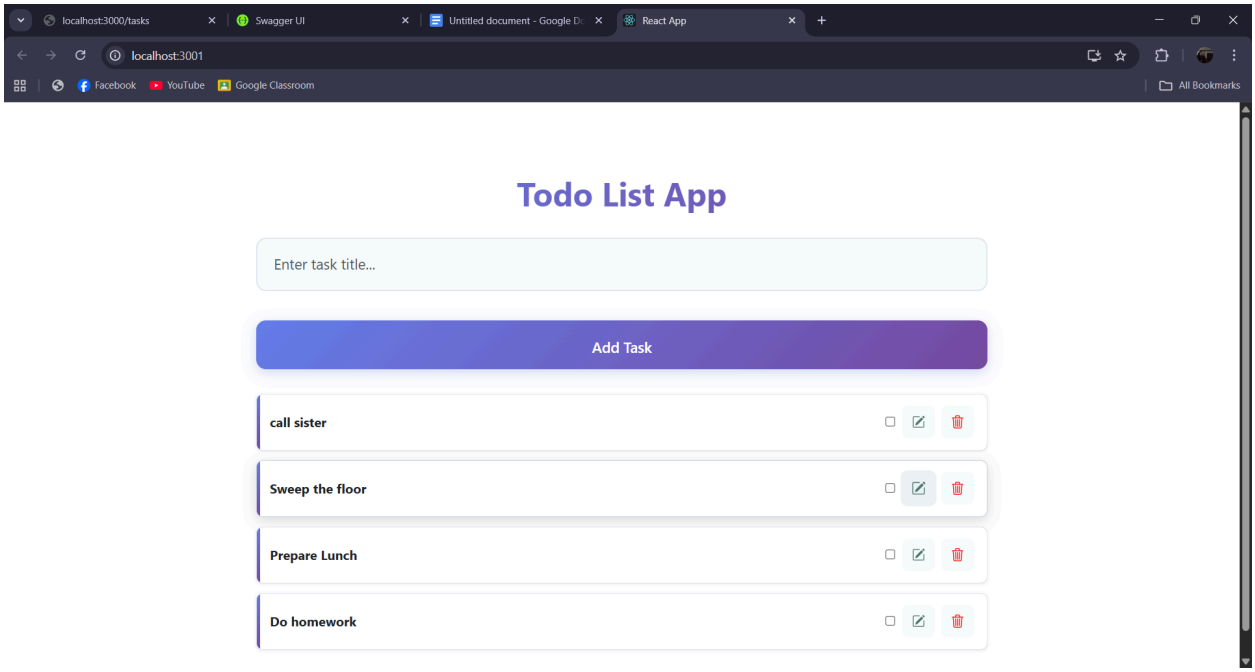




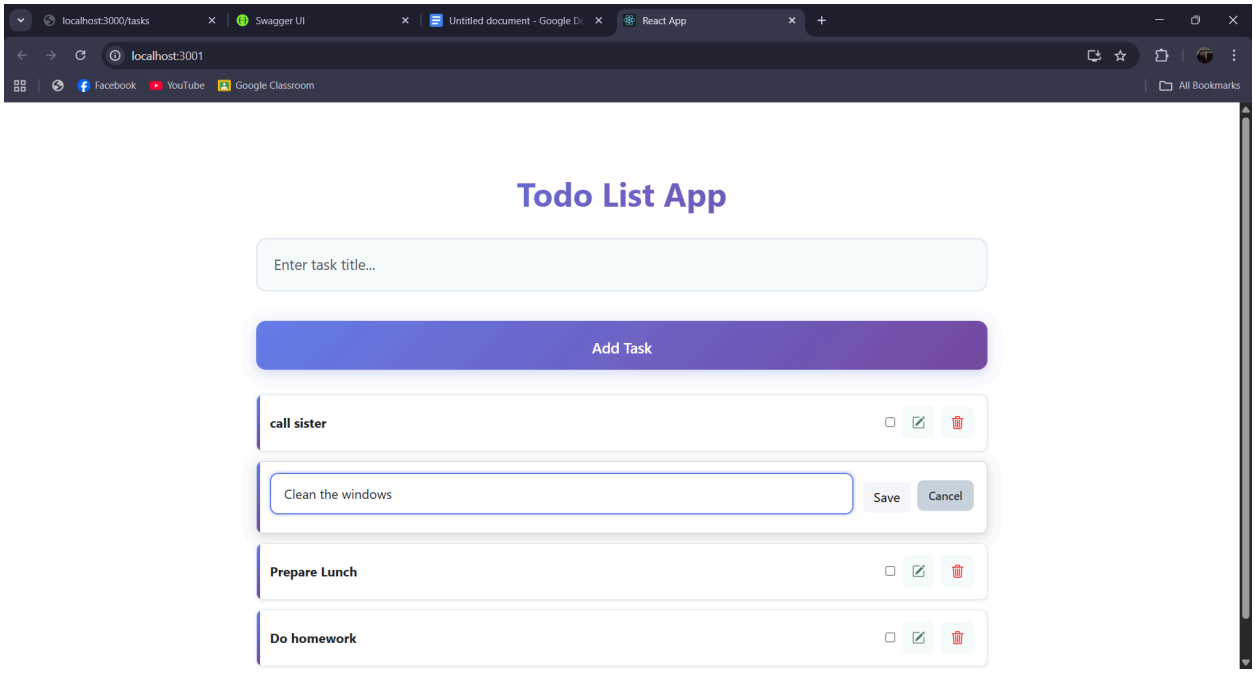


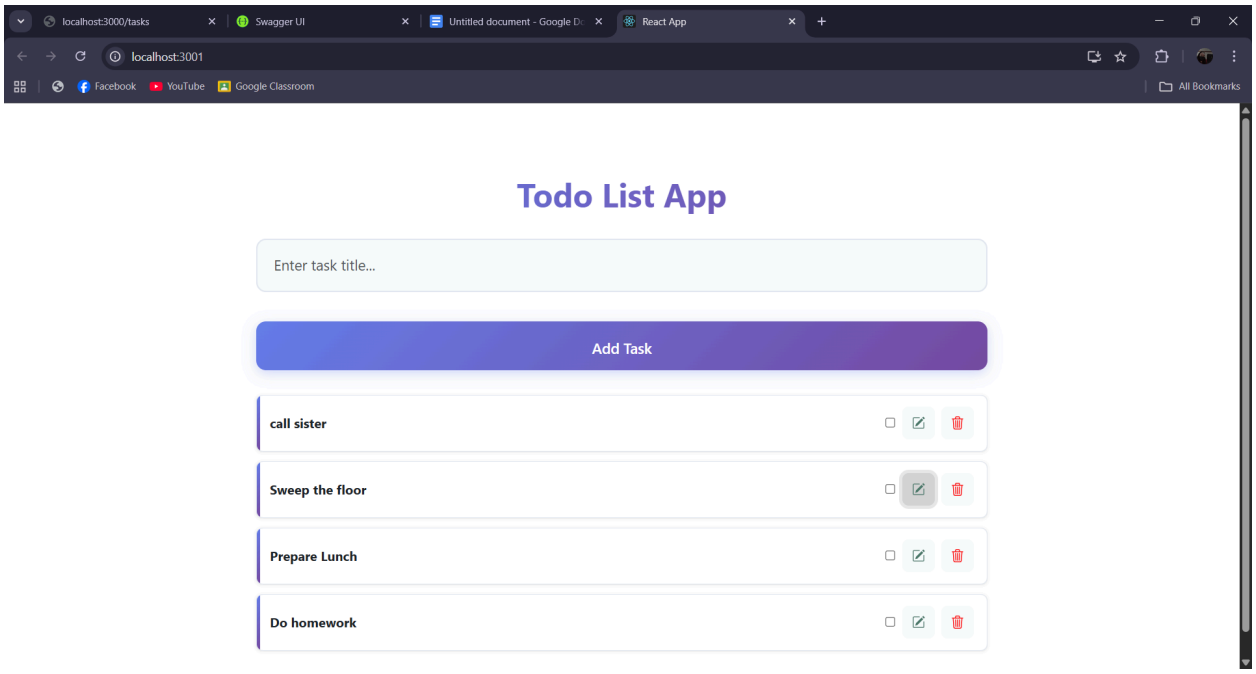
Updating the Task:



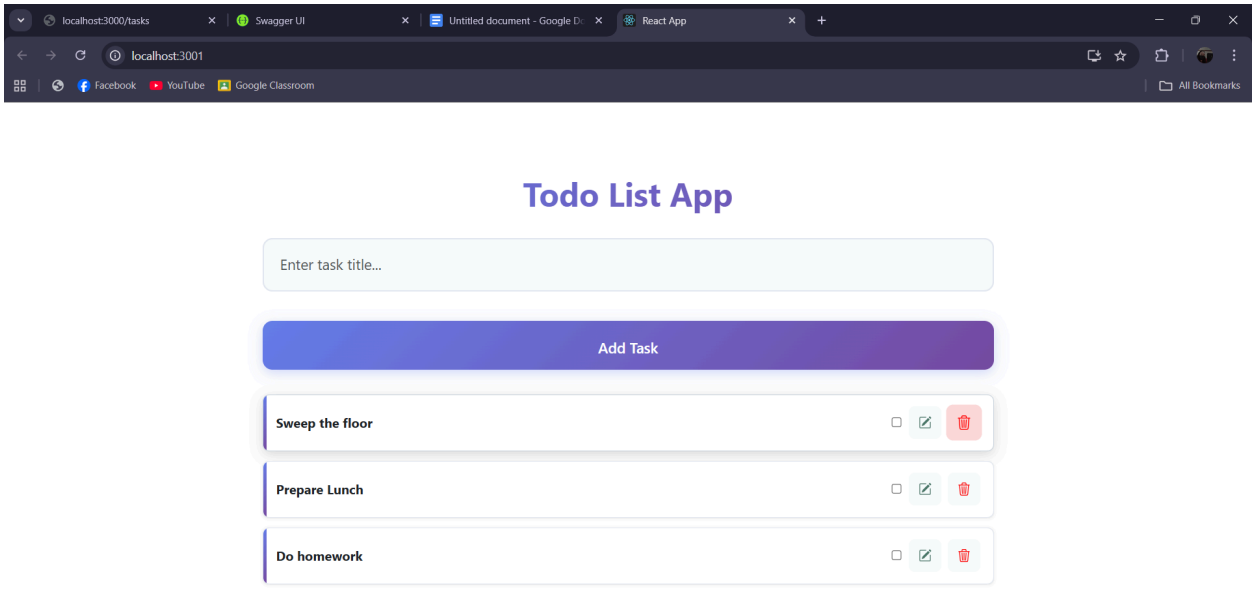


Cancelling a updating Task:



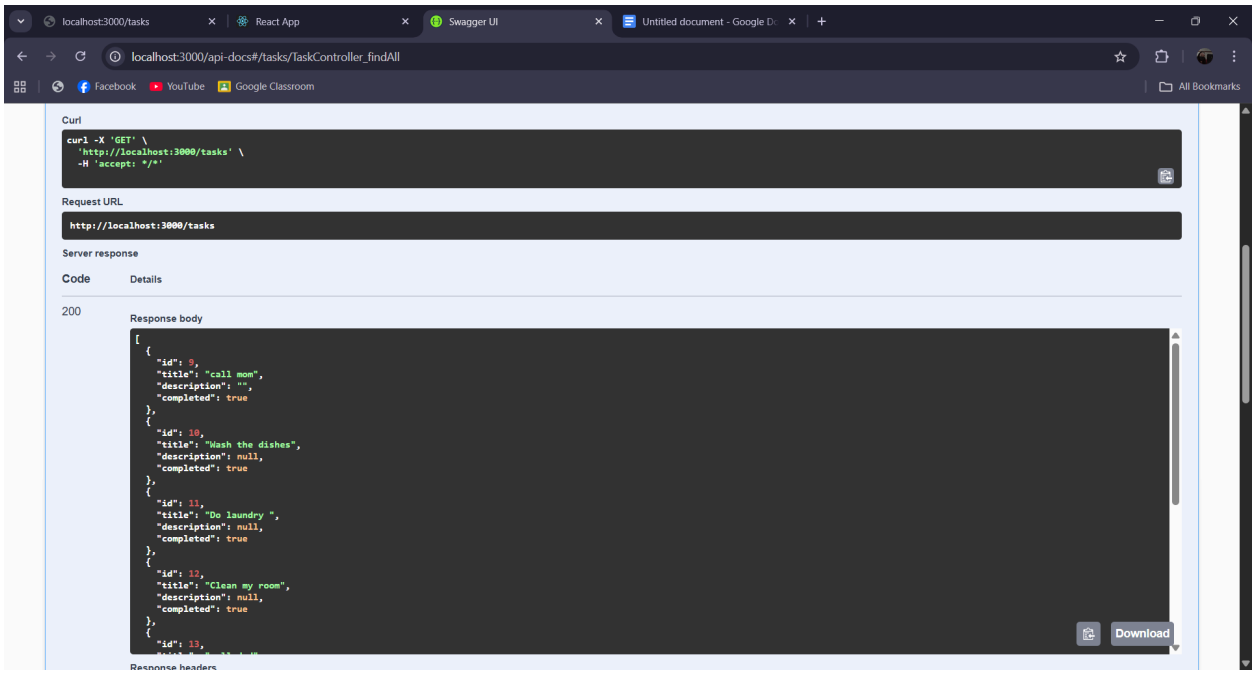


Deleting a task:

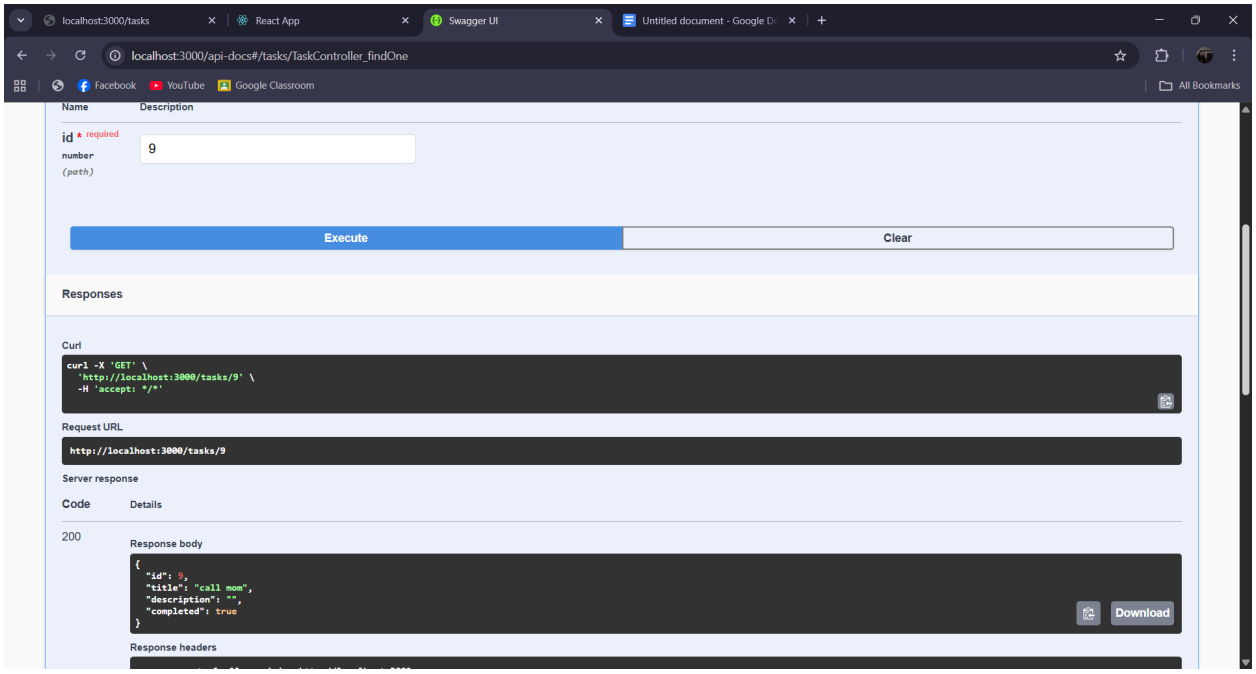


Swagger (API Docs)

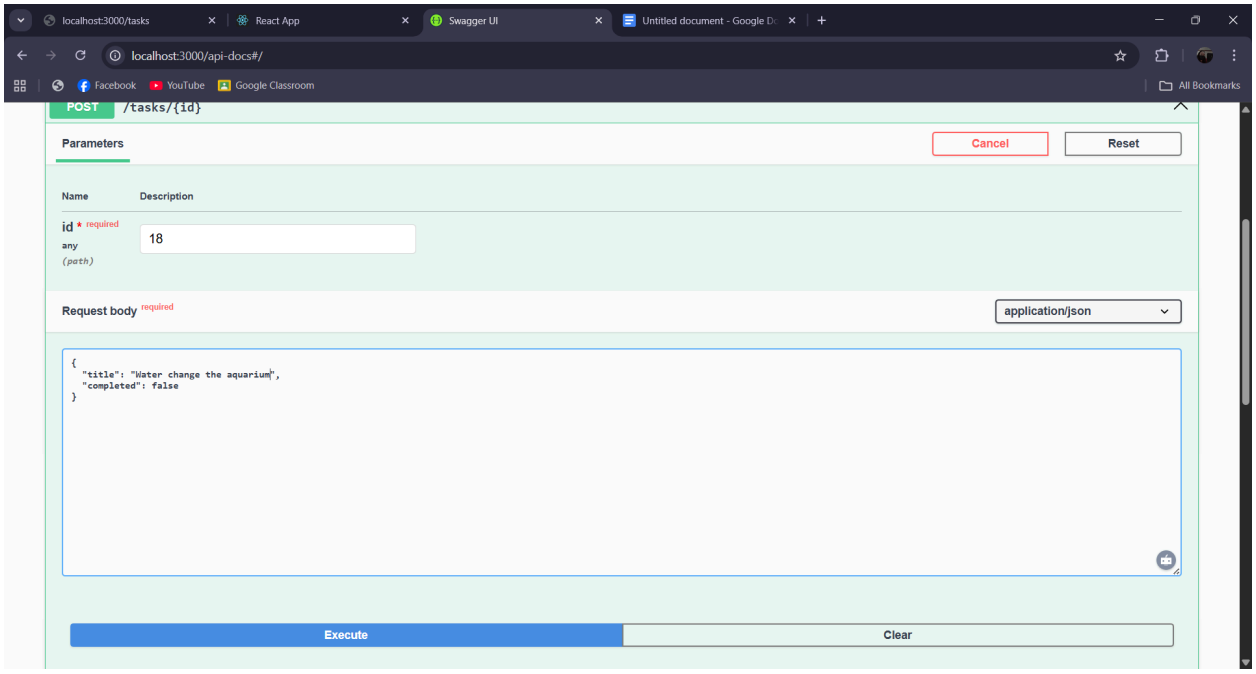
Get All:

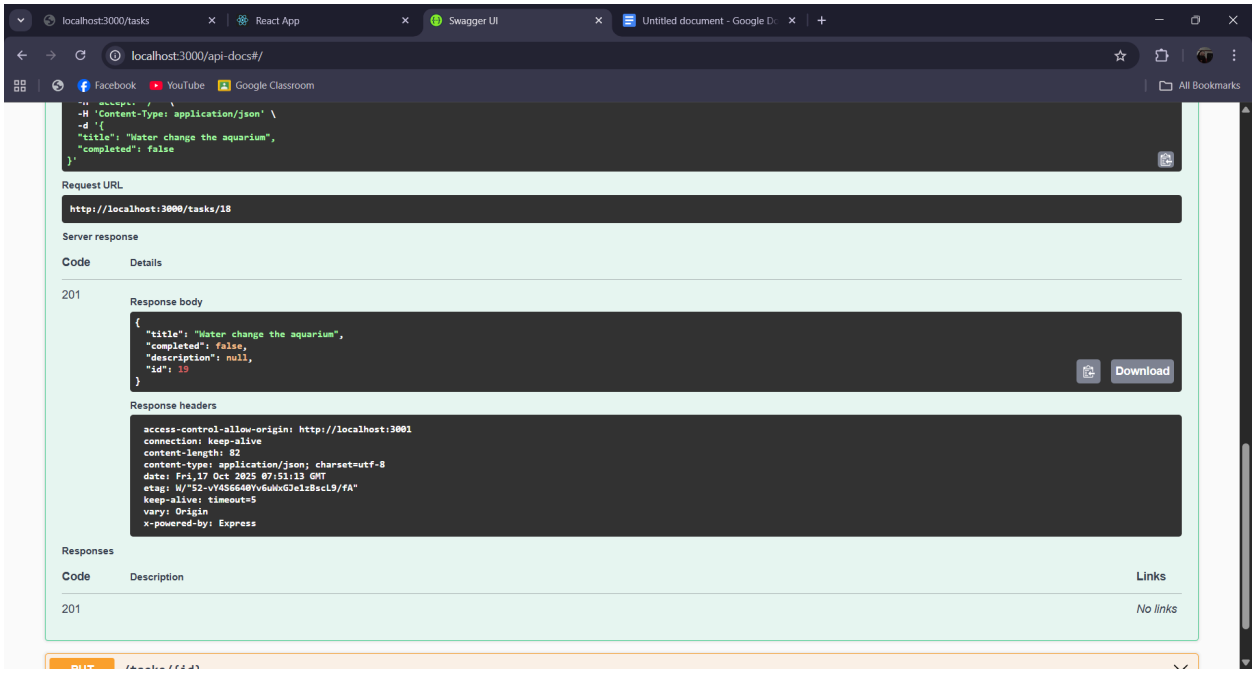


Get ID:

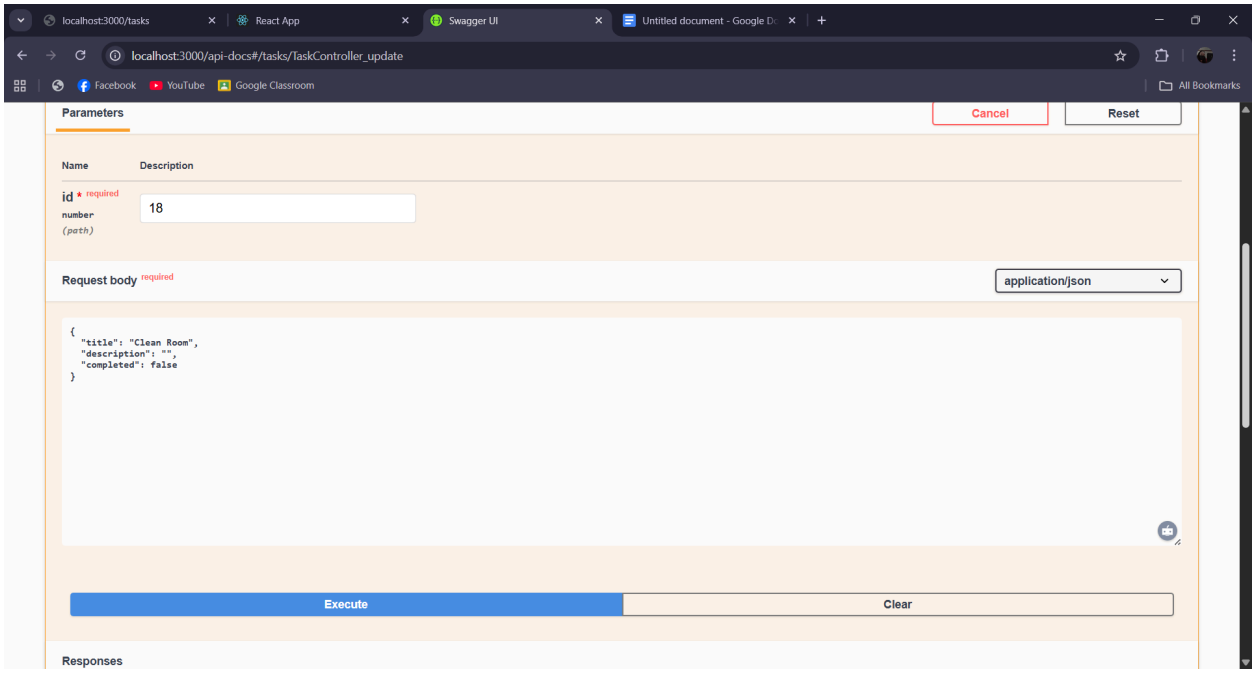


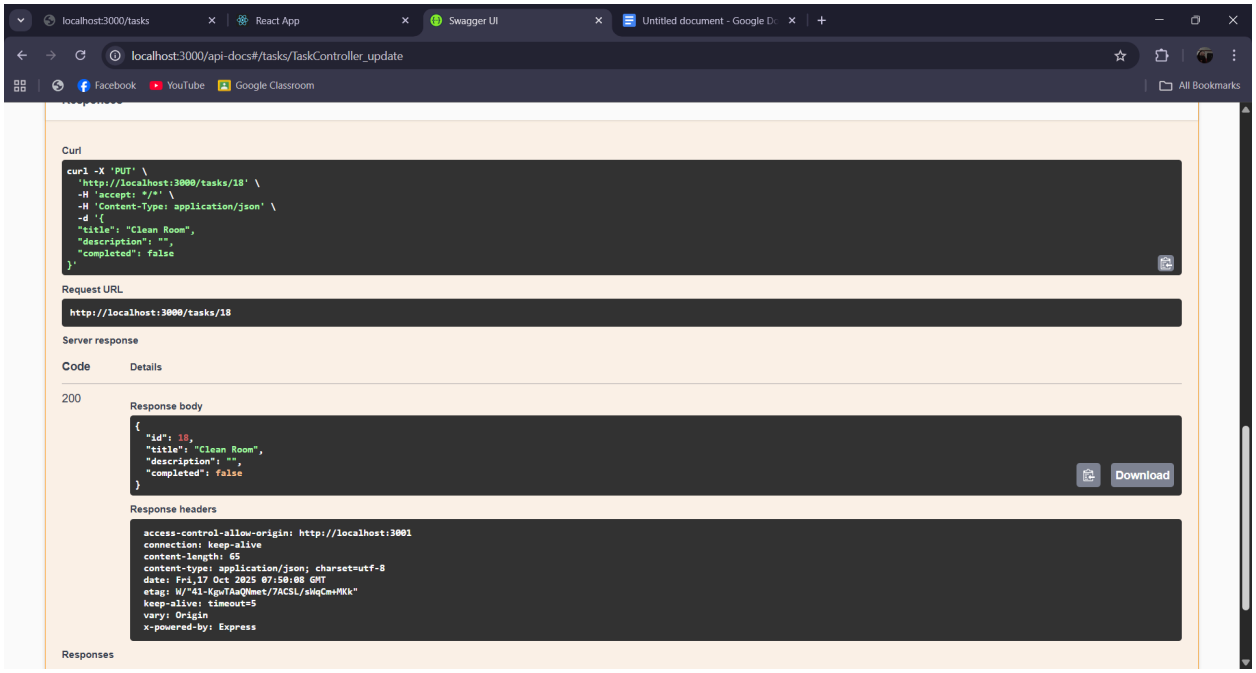
Post ID:


localhost:3000/tasks
React App
Swagger UI
Untitled document - Google Do
localhost:3000/api-docs#/
Facebook
YouTube
Google Classroom
All Bookmarks
POST
/tasks/{id}
Parameters
Cancel
Reset
Name
Description
id * required
any
(path)
18
Request body required
application/json
{
"title": "Water change the aquarium",
"completed": false
}
Execute
Clear

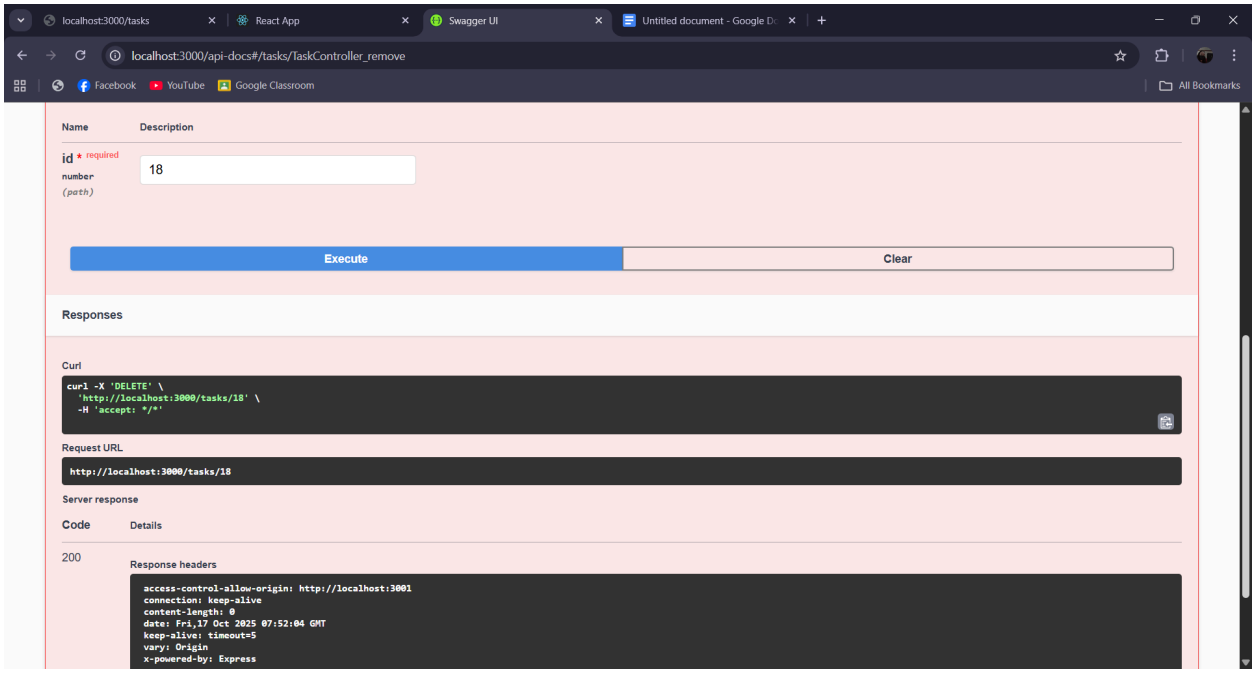

localhost:3000/tasks
React App
Swagger UI
Untitled document - Google Do
localhost:3000/api-docs#/
Facebook
YouTube
Google Classroom
All Bookmarks
-H 'Content-Type: application/json' \
-d '{
"title": "Water change the aquarium",
"completed": false
}'
Request URL
http://localhost:3000/tasks/18
Server response
Code
Details
201
Response body
{
"title": "Water change the aquarium",
"completed": false,
"description": null,
"id": 19
}
Download
Response headers
access-control-allow-origin: http://localhost:3001
connection: keep-alive
content-length: 82
content-type: application/json; charset=utf-8
date: Fri,17 Oct 2025 07:51:13 GMT
etag: W/"52-vY4S6640Yv6uNxGJe1zBscL9/fA"
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
Responses
Code
Description
Links
201
No links

Put ID:


localhost:3000/tasks
React App
Swagger UI
Untitled document - Google Do
localhost:3000/api-docs#/tasks/TaskController_update
Facebook
YouTube
Google Classroom
All Bookmarks
Parameters
Cancel
Reset
Name
Description
id * required
number
(path)
18
Request body required
application/json
{
"title": "Clean Room",
"description": "",
"completed": false
}
Execute
Clear
Responses


localhost:3000/tasks
React App
Swagger UI
Untitled document - Google Do
localhost:3000/api-docs#/tasks/TaskController_update
Facebook
YouTube
Google Classroom
All Bookmarks
Curl
curl -X 'PUT' \
'http://localhost:3000/tasks/18' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
"title": "Clean Room",
"description": "",
"completed": false
}'
Request URL
http://localhost:3000/tasks/18
Server response
Code
Details
200
Response body
{
"id": 18,
"title": "Clean Room",
"description": "",
"completed": false
}
Download
Response headers
access-control-allow-origin: http://localhost:3001
connection: keep-alive
content-length: 65
content-type: application/json; charset=utf-8
date: Fri,17 Oct 2025 07:50:08 GMT
etag: W/"41-KgwTAaQNmet/7ACSL/sWqCm+MKk"
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
Responses

Delete ID:



Instructions:

A. Download project from Google Classroom

Step 1: Click the attached zip or folder. Choose **Download**. If teacher gave a Git URL, copy it.
Step 2: Unzip the file and open the extracted folder in VS Code.

B. Prepare your machine

Step 3: Install Node.js (recommended LTS) from nodejs.org. Verify:

```
node -v
npm -v
```

Step 4: Install Yarn:

```
npm install -g yarn
```

C. Install dependencies

Step 5: Open terminal and go to backend folder:

```
cd path/to/project/backend/todo-backend
```

Step 6: Install packages:

```
npm install
```

or if using yarn

```
yarn
```

Step 7: In a new terminal tab, go to frontend folder:

```
cd path/to/project/frontend/todo-frontend
npm install
```

or `yarn`

D. Run backend and frontend

Step 8: Start backend (in backend folder):

```
npm run start:dev
```

or `npm run start`

Step 9: Wait for output like `App is running at http://localhost:3000`

Step 10: Start frontend (in frontend folder):

```
npm start
```

or `npm run dev` or `yarn start`

Step 11: Open browser: frontend usually `http://localhost:3001` or `http://localhost:3000` — check terminal output. If backend uses 3000, frontend will use another port like 3001 or 5173 (Vite)

Step 12: To view API docs (if using Swagger with NestJS): open `http://localhost:3000/api-docs`

F. Test

15. Use the frontend UI to add/read/update/delete tasks. Or use Postman to call API endpoints.

Backend Link:

`http://localhost:3000`

Frontend Link:

`http://localhost:3001`

API DOCU Link (Swagger):

`http://localhost:3000/api-docs`