No problem! Here’s the **complete full-stack ToDo app** using:

* **Backend**: Python + Flask
* **Frontend**: React + Vite

**🧠 Goal**

Build a To-Do app where users can:

* See all tasks
* Add a task
* Delete a task

**🔹 Backend (Flask)**

**✅ Step 1: Create project folder**

bash

CopyEdit

mkdir todo-backend

cd todo-backend

**✅ Step 2: Install Flask and CORS**

bash

CopyEdit

pip install flask flask-cors

**✅ Step 3: Create app.py**

python

CopyEdit

from flask import Flask, request, jsonify

from flask\_cors import CORS

app = Flask(\_\_name\_\_)

CORS(app)

todos = [

{"id": 1, "task": "Learn Flask", "done": False},

{"id": 2, "task": "Build a React app", "done": False}

]

@app.route('/')

def home():

return "ToDo API is running!"

@app.route('/todos', methods=['GET'])

def get\_todos():

return jsonify(todos)

@app.route('/todos', methods=['POST'])

def create\_todo():

new\_todo = request.json

new\_todo["id"] = len(todos) + 1

todos.append(new\_todo)

return jsonify(new\_todo), 201

@app.route('/todos/<int:todo\_id>', methods=['DELETE'])

def delete\_todo(todo\_id):

global todos

todos = [t for t in todos if t["id"] != todo\_id]

return '', 204

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**✅ Step 4: Run Flask app**

bash

CopyEdit

python app.py

Make sure it runs at http://localhost:5000.

**🔷 Frontend (React + Vite)**

**✅ Step 1: Create frontend with Vite**

bash

CopyEdit

npm create vite@latest todo-frontend -- --template react

cd todo-frontend

npm install

**✅ Step 2: Project Structure**

css

CopyEdit

todo-frontend/

├── src/

│ ├── App.jsx

│ ├── components/

│ │ └── TodoList.jsx

│ └── index.css

**✅ Step 3: Code for src/App.jsx**

jsx

CopyEdit

import { useEffect, useState } from 'react';

import TodoList from './components/TodoList';

function App() {

const [todos, setTodos] = useState([]);

const [newTask, setNewTask] = useState('');

useEffect(() => {

fetch('http://localhost:5000/todos')

.then(res => res.json())

.then(data => setTodos(data));

}, []);

const addTodo = () => {

if (!newTask.trim()) return;

fetch('http://localhost:5000/todos', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ task: newTask, done: false })

})

.then(res => res.json())

.then(todo => {

setTodos([...todos, todo]);

setNewTask('');

});

};

const deleteTodo = (id) => {

fetch(`http://localhost:5000/todos/${id}`, { method: 'DELETE' })

.then(() => {

setTodos(todos.filter(todo => todo.id !== id));

});

};

return (

<div style={{ padding: 20, maxWidth: 500, margin: 'auto' }}>

<h1>To-Do List</h1>

<div style={{ display: 'flex', marginBottom: 20 }}>

<input

value={newTask}

onChange={e => setNewTask(e.target.value)}

placeholder="Enter new task"

style={{ flex: 1, marginRight: 10 }}

/>

<button onClick={addTodo}>Add</button>

</div>

<TodoList todos={todos} onDelete={deleteTodo} />

</div>

);

}

export default App;

**✅ Step 4: Code for src/components/TodoList.jsx**

jsx

CopyEdit

import React from 'react';

function TodoList({ todos, onDelete }) {

return (

<ul>

{todos.map(todo => (

<li key={todo.id} style={{ marginBottom: 10 }}>

{todo.task}

<button onClick={() => onDelete(todo.id)} style={{ marginLeft: 10 }}>

❌

</button>

</li>

))}

</ul>

);

}

export default TodoList;

**✅ Step 5: Run the frontend**

bash

CopyEdit

npm run dev

Vite will serve the app at something like:  
http://localhost:5173

**✅ Result**

* Open http://localhost:5173
* Your Flask backend runs at http://localhost:5000
* React app connects via RESTful API