Full Stack TODO app

* FastAPI + PostgreSQL + Python for the backend
* Angular for the frontend
* open backend and frontend separately in two different terminals

## 🗂️ Folder Structure Overview

Let’s assume this project is structured like this:

/todo\_app

│

├── to\_do\_backend/

│ └── main.py ← FastAPI backend

│

└── to\_frontend/

src ----> app

├── app.component.ts ← Angular frontend logic

├── app.component.html ← Angular template

├── app.component.css ← (Optional, your style file)

└── (Other Angular files which come when we install angular)

## 🔸 Backend: to\_do\_backend/main.py

### What it does:

* Connects to a PostgreSQL database named angular
* Provides three endpoints:
  + GET /get\_task: Fetches all tasks
  + POST /add\_task: Adds a new task (via form data)
  + POST /delete\_task: Deletes a task using the id

### Requires:

You need a table in the todo database:

CREATE TABLE todo (

id SERIAL PRIMARY KEY,

task TEXT

);

### How to run the backend:

1. Install dependencies:

pip install fastapi uvicorn psycopg2-binary

1. Run the server:

uvicorn main:app --reload

## 🔹 Frontend: to\_do\_frontend/ (Angular)

### Key files:

* app.component.ts: Your logic for sending/receiving tasks via HTTP
* app.component.html: UI with input, task list, and delete button

### How it works:

* Uses ngModel for two-way binding of the input
* Calls FastAPI endpoints using Angular’s HttpClient
* Displays tasks and lets users add/delete with button clicks

### How to run the frontend:

1. Make sure you have Angular CLI:

npm install -g @angular/cli

1. Install dependencies:

cd to\_do\_frontend

npm install

1. Run the Angular app:

**ng serve**

1. Open browser at: http://localhost:4200/

## ⚙️ CORS Setup

Your backend allows any frontend to access it via:

app.add\_middleware(

CORSMiddleware,

allow\_origins=["\*"], # Allow all

...

)

Which is good for development. In production, you'd want to restrict this.