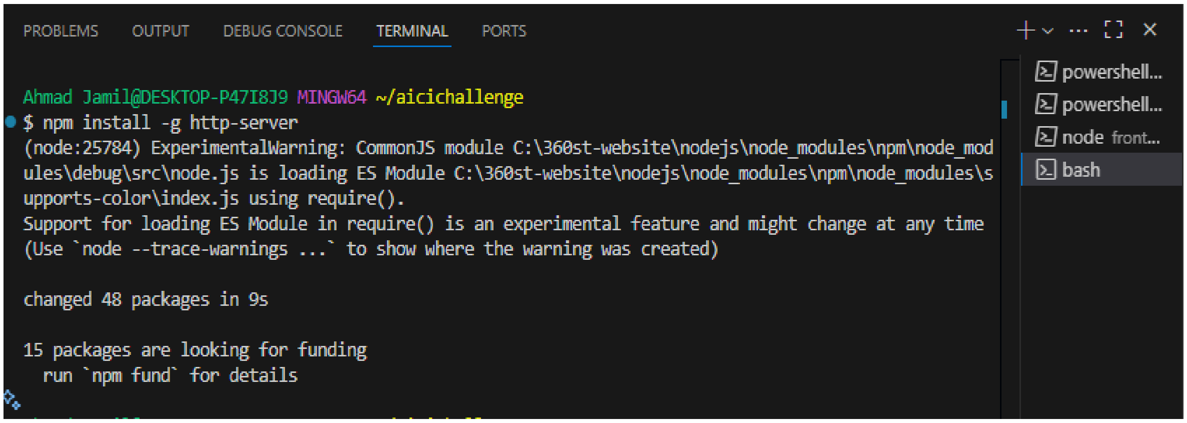
Here are the detailed, **step-by-step instructions to build and run the entire Todo Application system**. This guide covers all three components: the User Service, the Todo Service, and the Frontend.

Prerequisites

You must have the following installed before you begin:

- **Node.js & npm**: Download and install Node.js from the official website. `npm` is included with Node.js.
- **MongoDB**: Have a running MongoDB instance, either locally or a connection string to a cloud-hosted one.
- **A Code Editor**: A program like VS Code or Sublime Text to edit your project files.
- **A Web Server**: You'll need a simple local web server to serve the frontend files. If you don't have one, you can install `http-server` via npm:

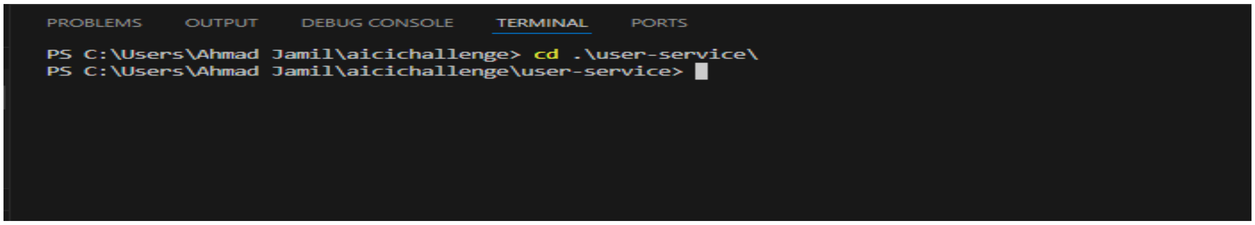


```
Ahmad Jamil@DESKTOP-P47I8J9 MINGW64 ~/aicichallenge
$ npm install -g http-server
(node:25784) ExperimentalWarning: CommonJS module C:\360st-website\nodejs\node_modules\npm\node_mod
ules\debug\src\node.js is loading ES Module C:\360st-website\nodejs\node_modules\npm\node_modules\s
upports-color\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)

changed 48 packages in 9s

15 packages are looking for funding
  run `npm fund` for details
```

Step 1: User Service Setup

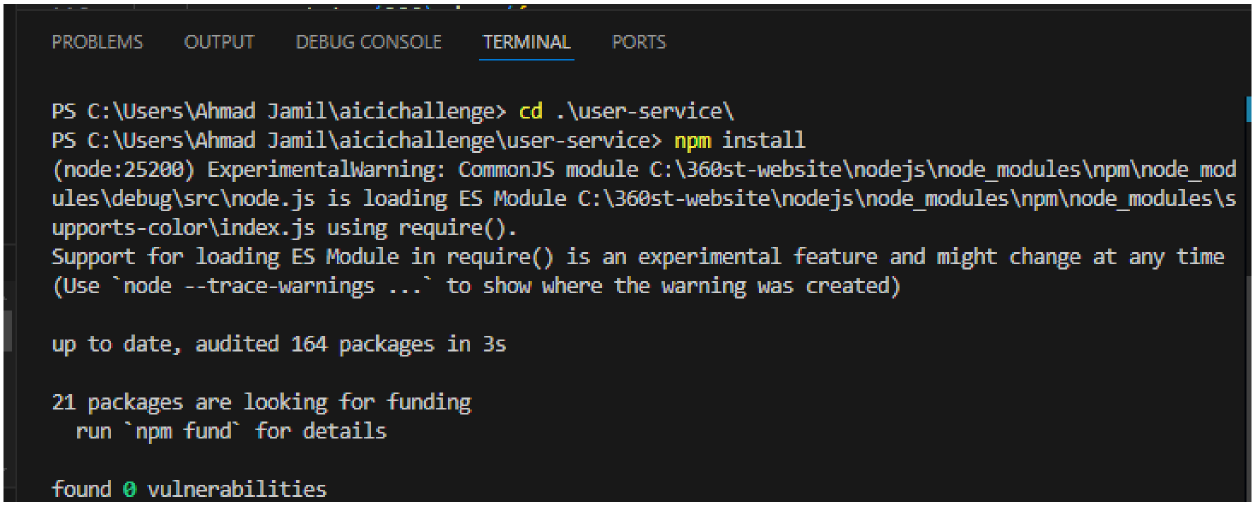
This service is responsible for user registration and authentication.

1. **Navigate to the directory**: Open your terminal or command prompt and go to the root folder of the User Service project.



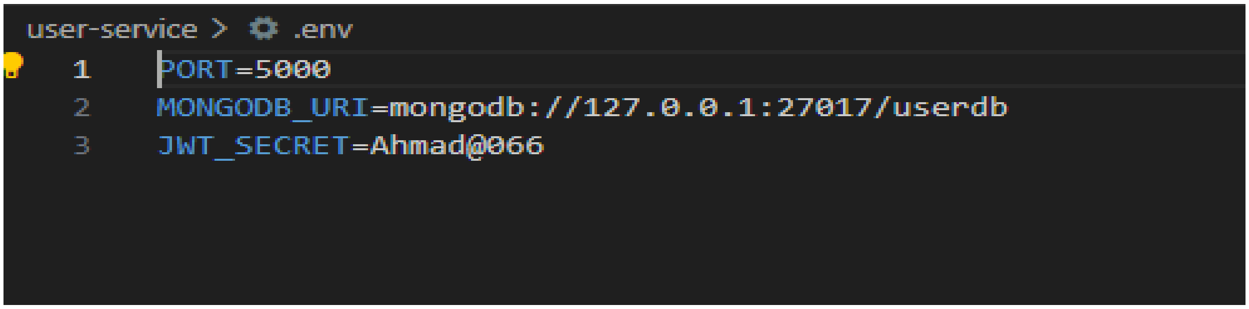
```
PS C:\Users\Ahmad Jamil\aicichallenge> cd .\user-service\
PS C:\Users\Ahmad Jamil\aicichallenge\user-service>
```

2. **Install dependencies**: Run the following command to install all the required Node.js packages.



```
PS C:\Users\Ahmad Jamil\aicichallenge> cd .\user-service\
PS C:\Users\Ahmad Jamil\aicichallenge\user-service> npm install
(node:25200) ExperimentalWarning: CommonJS module C:\360st-website\nodejs\node_modules\npm\node_mod
ules\debug\src\node.js is loading ES Module C:\360st-website\nodejs\node_modules\npm\node_modules\s
upports-color\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)

up to date, audited 164 packages in 3s

21 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

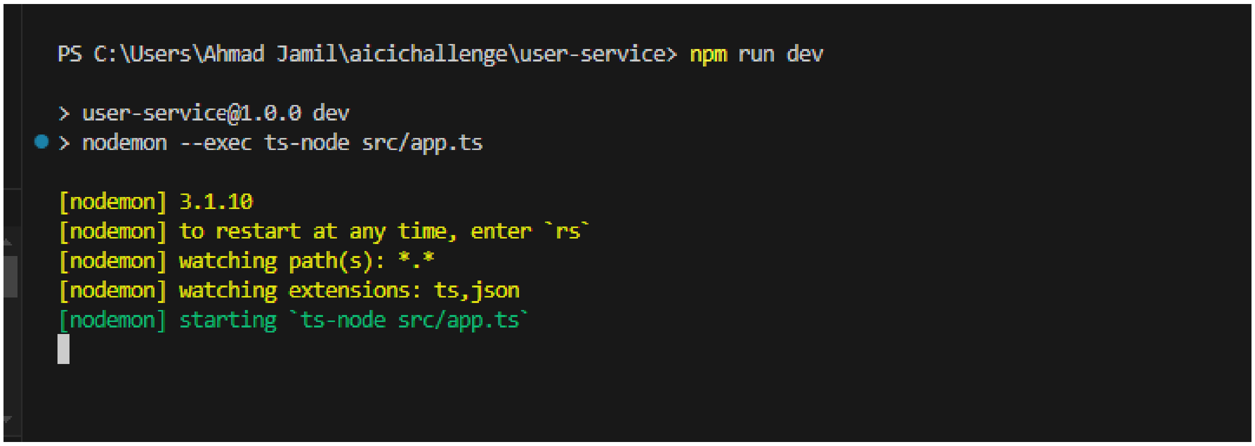
3. **Configure environment variables**: Create a file named `.env` in this directory. Paste the following content into it, replacing the values as needed.



```
PORT=5000
MONGODB_URI=mongodb://127.0.0.1:27017/userdb
JWT_SECRET=Ahmad@066
```

   a. **PORT**: The port the service will run on. `5000` is the default.
   b. **DATABASE_URL**: The connection string for your MongoDB database.
   c. **JWT_SECRET**: A strong, random key used to sign and verify JWTs. Remember this key for the Todo Service.

4. **Start the server**: Run the start command.

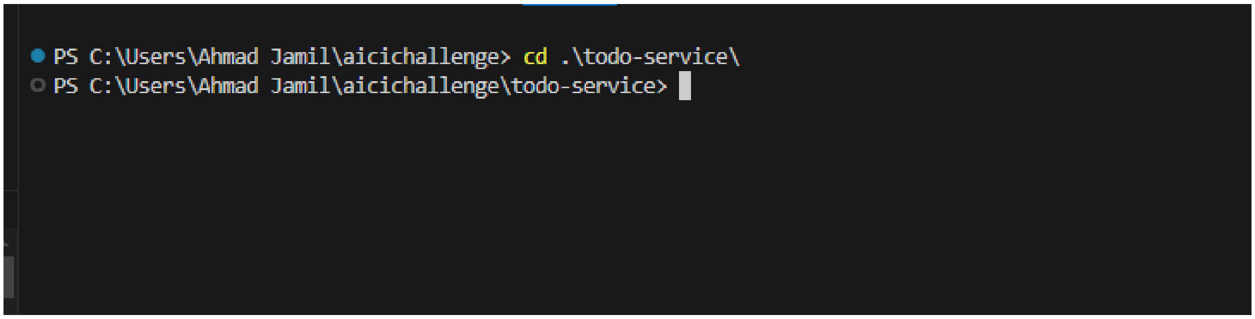


```
PS C:\Users\Ahmad Jamil\aicichallenge\user-service> npm run dev

> user-service@1.0.0 dev
> nodemon --exec ts-node src/app.ts

[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: ts,json
[nodemon] starting `ts-node src/app.ts`
```

Step 2: Todo Service Setup

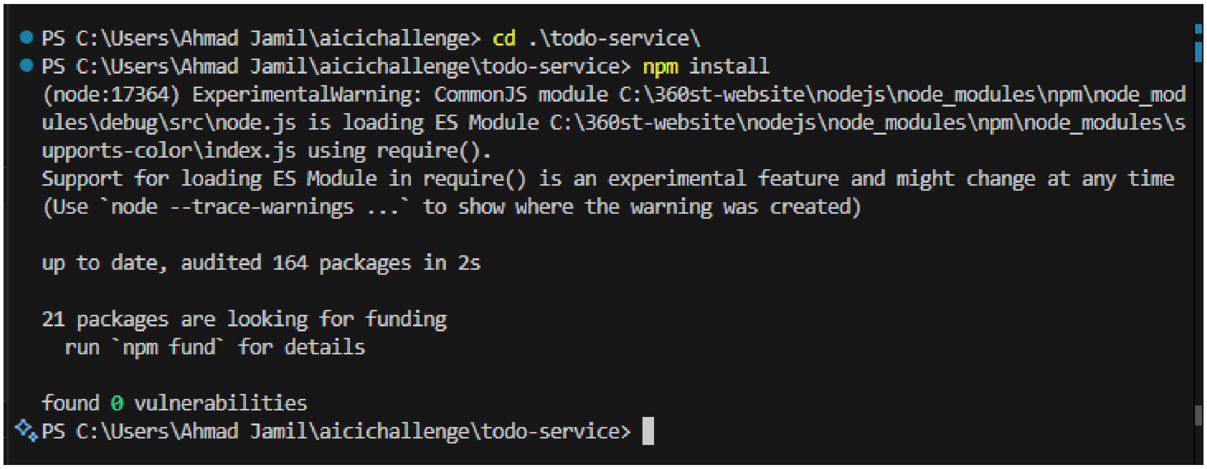
This service manages the creation, retrieval, updating, and deletion of todo items.

1. **Open a new terminal**: Open a **new** terminal window and navigate to the Todo Service project directory.



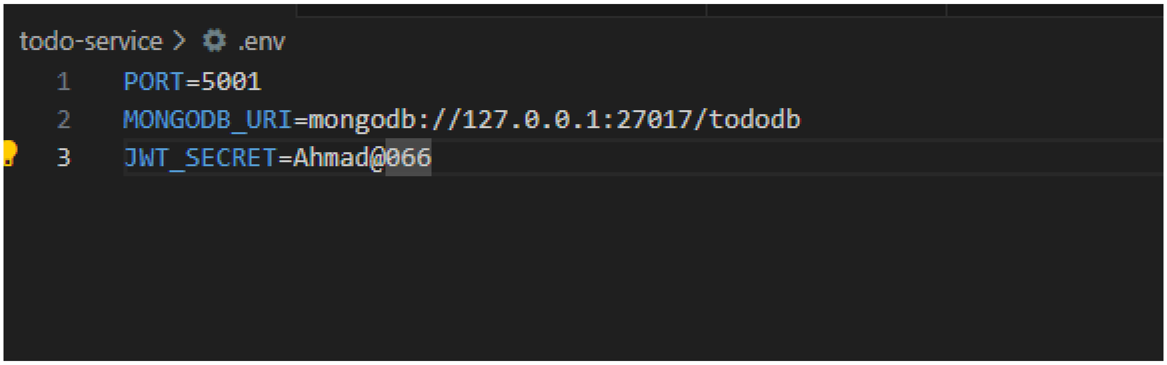
```
PS C:\Users\Ahmad Jamil\aicichallenge> cd .\todo-service\
PS C:\Users\Ahmad Jamil\aicichallenge\todo-service>
```

2. **Install dependencies**: Install the required Node.js packages for this service.



```
PS C:\Users\Ahmad Jamil\aicichallenge> cd .\todo-service\
PS C:\Users\Ahmad Jamil\aicichallenge\todo-service> npm install
(node:17364) ExperimentalWarning: CommonJS module C:\360st-website\nodejs\node_modules\npm\node_mod
ules\debug\src\node.js is loading ES Module C:\360st-website\nodejs\node_modules\npm\node_modules\s
upports-color\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)

up to date, audited 164 packages in 2s

21 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Ahmad Jamil\aicichallenge\todo-service>
```

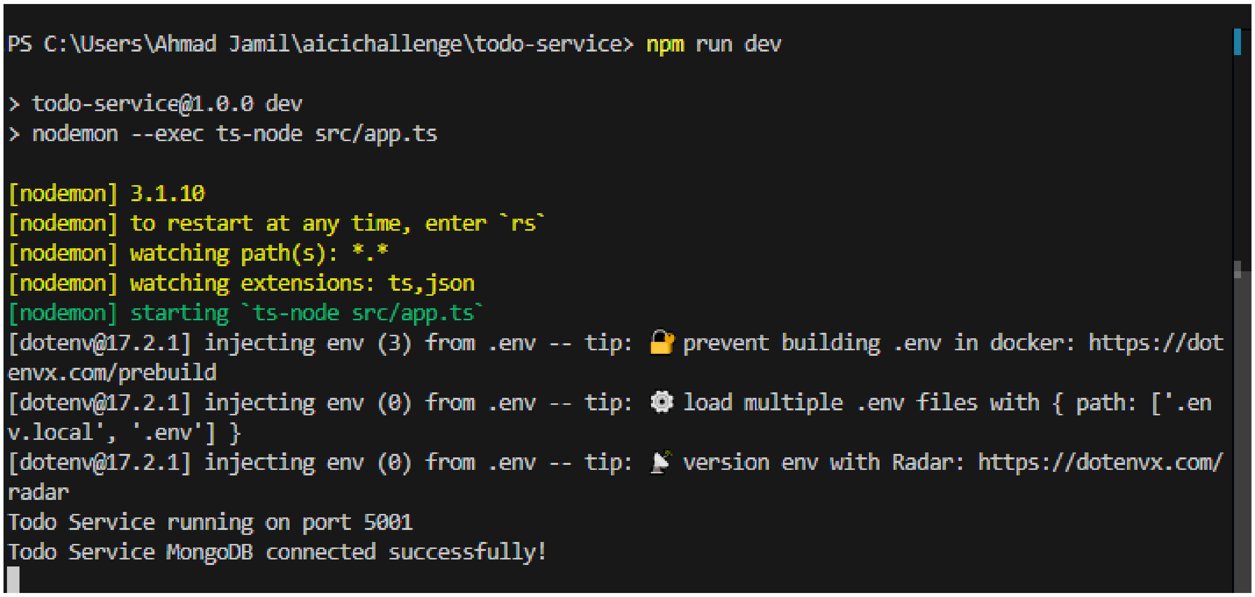
3. **Configure environment variables**: Create a `.env` file in this directory and add the following content.



```
PORT=5001
MONGODB_URI=mongodb://127.0.0.1:27017/tododb
JWT_SECRET=Ahmad@066
```

   a. **IMPORTANT**: The `JWT_SECRET` **must be identical** to the one you set in the User Service's `.env` file. This allows the Todo Service to validate tokens issued by the User Service.

4. **Start the server**: Start the server with the following command.

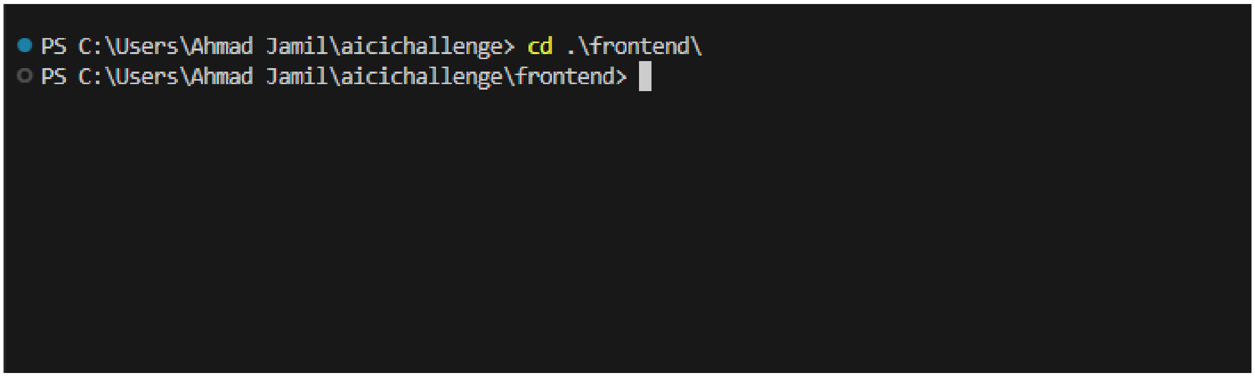


```
PS C:\Users\Ahmad Jamil\aicichallenge\todo-service> npm run dev

> todo-service@1.0.0 dev
> nodemon --exec ts-node src/app.ts

[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: ts,json
[nodemon] starting `ts-node src/app.ts`
[dotenv@17.2.1] injecting env (3) from .env -- tip: 🔐 prevent building .env in docker: https://dot
envx.com/prebuild
[dotenv@17.2.1] injecting env (0) from .env -- tip: ⚙️ load multiple .env files with { path: ['.en
v.local', '.env'] }
[dotenv@17.2.1] injecting env (0) from .env -- tip: 📡 version env with Radar: https://dotenvx.com/
radar
Todo Service running on port 5001
Todo Service MongoDB connected successfully!
```

Step 3: Frontend Setup (Optional)

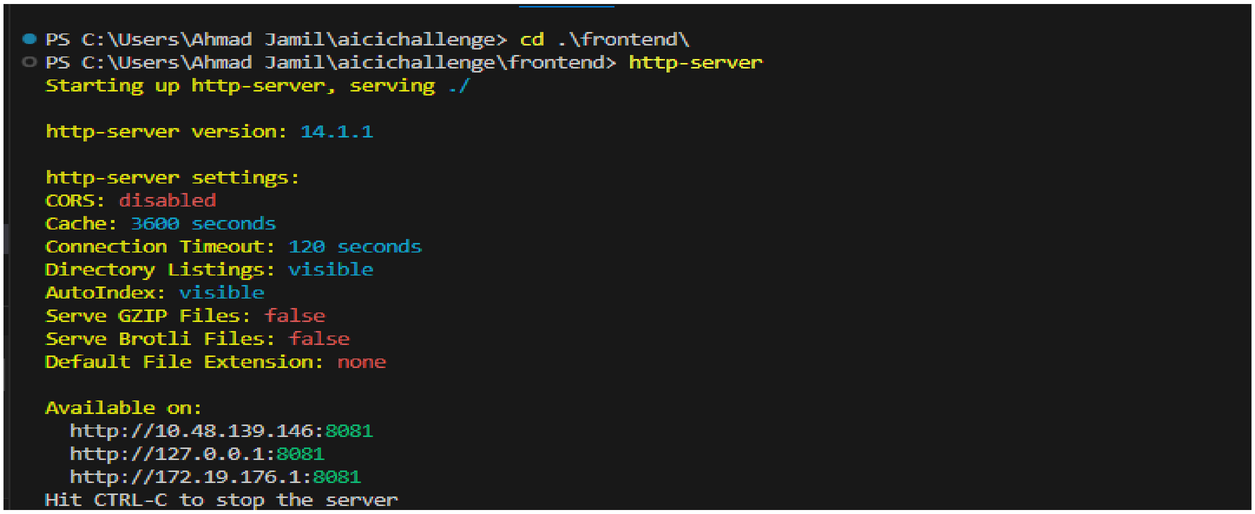
This is the web interface that allows you to interact with the backend services.

1. **Open a third terminal**: Open a final terminal window and navigate to the `frontend` project folder.



```
PS C:\Users\Ahmad Jamil\aicichallenge> cd .\frontend\
PS C:\Users\Ahmad Jamil\aicichallenge\frontend> 
```

2. **Start the web server**: Use the `http-server` command to serve the `index.html` file.



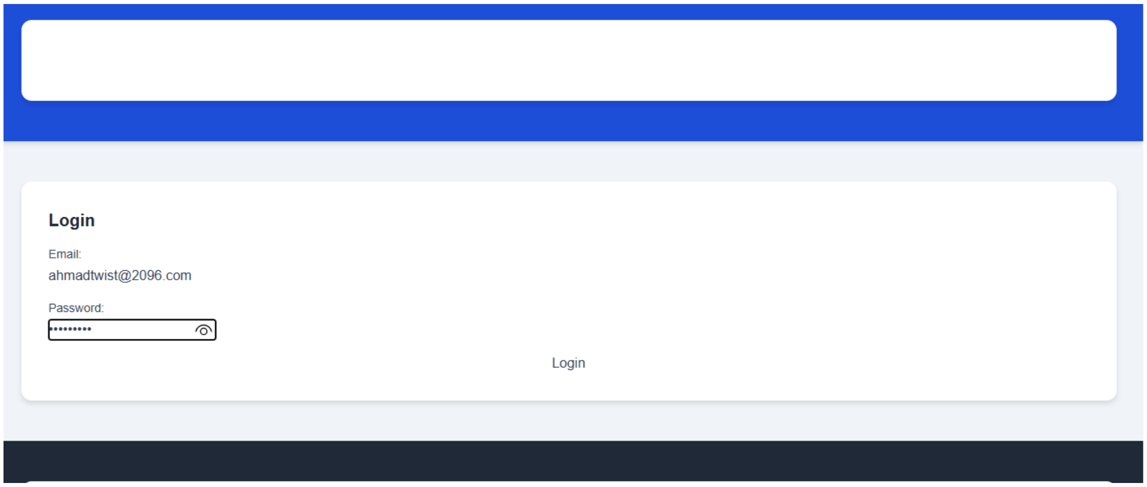
```
PS C:\Users\Ahmad Jamil\aicichallenge> cd .\frontend\
PS C:\Users\Ahmad Jamil\aicichallenge\frontend> http-server
Starting up http-server, serving ./

http-server version: 14.1.1

http-server settings:
CORS: disabled
Cache: 3600 seconds
Connection Timeout: 120 seconds
Directory Listings: visible
AutoIndex: visible
Serve GZIP Files: false
Serve Brotli Files: false
Default File Extension: none

Available on:
  http://10.48.139.146:8081
  http://127.0.0.1:8081
  http://172.19.176.1:8081
Hit CTRL-C to stop the server
```

Final Step: Using the Application

With all three services running in their respective terminal windows, your complete system is now operational.

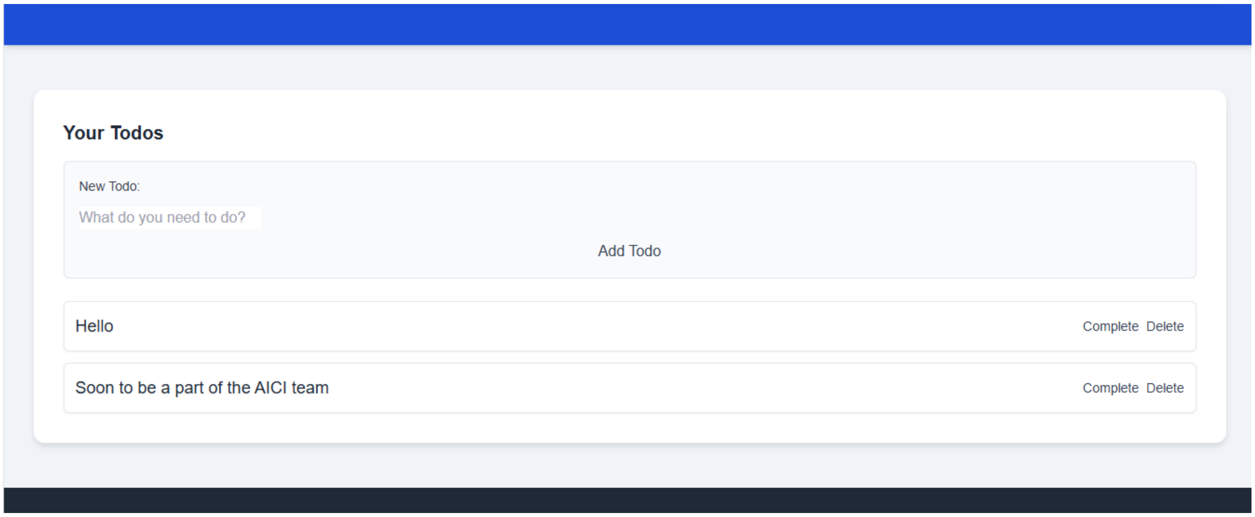
1. Open your web browser and go to **http://localhost:8081**.

2. You can now register a new user, log in, and begin managing your todo list via the web interface.



Email: ahmadtwist@2096.com

Password: •••••••••



Hello

Soon to be a part of the AICI team