DOCUMENTATION

A. Running the Script

To run the FastAPI application, follow these steps:

1. Extract the files to a folder
   main.py
   address_router.py
   models.py
   database.py
   config.py
   db_utils.py
   create_sample_db.py (optional)
   view_db.py (optional)
2. Open Command prompt in project directory.
3. Install the required dependencies by running the following command in your terminal or command prompt:
   pip install -r requirements.txt
4. Navigate to the project directory containing the `main.py` file.
5. Run the application using the following command:
   uvicorn main:app --reload

   This command will start the FastAPI server and automatically reload the application.

   When you make changes to the code, the `--reload` option watches for changes in the source code and restarts the server automatically.

6. Once the server is running, you should see output similar to the following:

   ```
   INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
   INFO:     Started reloader process [15124] using WatchFiles
   INFO:     Started server process [27832]
   INFO:     Waiting for application startup.
   INFO:     Application startup complete.
   ```

   This indicates that the application is running and accessible at `http://127.0.0.1:8000`.

B. Working with the FastAPI Swagger UI

1. With the FastAPI server running, open your web browser and navigate to ` http://localhost:8000/docs.`.
2. The Swagger UI should be displayed, showing the available API endpoints and their descriptions.
3. You can interact with the API endpoints directly from the Swagger UI. Each endpoint is documented with its parameters, request body, and response models.
4. To test an API endpoint, follow these steps:

   - Expand the endpoint by clicking on it.
   - If the endpoint requires request parameters, enter the values in the provided fields.
   - If the endpoint requires a request body, click on the "Try it out" button and enter the request body in the provided editor.
   - Click on the "Execute" button to send the request to the API.
   - The response from the API will be displayed in the "Responses" section below.
5. The application provides the following API endpoints:
   a) POST /addresses: Create a new address.
      - Click the Try it out button.
      - Enter the address details in the Request body section, such as:

```
{
  "name": "New Address",
  "latitude": 40.7128,
  "longitude": -74.0060
}
```

      - Click Execute to send the request.

   b) GET /addresses: Get a list of all addresses.
      - Click the Try it out button.
      - Click Execute to send the request.

Curl

```
curl -X 'GET' \
  'http://localhost:8000/addresses' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8000/addresses
```

Server response

Code	Details
200	Response body

```
[
  {
    "name": "Address 1",
    "latitude": 0,
    "longitude": 0
  },
  {
    "name": "Address 1",
    "latitude": -60,
    "longitude": 60
  }
]
```

Download

Response headers

```
content-length: 108
content-type: application/json
date: Thu,16 May 2024 11:00:55 GMT
server: uvicorn
```

c) GET /addresses/nearby: Get a list of addresses within a certain distance from a given location.

- Click the Try it out button.
- Enter the latitude, longitude, and distance parameters in the respective fields.
- Click Execute to send the request.

Curl

```
curl -X 'GET' \
  'http://localhost:8000/addresses/nearby?latitude=-60&longitude=60&distance=1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8000/addresses/nearby?latitude=-60&longitude=60&distance=1
```

Server response

Code	Details
200	Response body

```
[
  {
    "name": "Address 1",
    "latitude": -60,
    "longitude": 60
  }
]
```

Download

Response headers

```
content-length: 56
content-type: application/json
date: Thu,16 May 2024 11:02:44 GMT
server: uvicorn
```

d) PUT /addresses/{address_id}: Update an existing address.

- Click the Try it out button.
- Enter the address_id (Number) in the path parameter field.
- Enter the updated address details in the Request body section.
- Click Execute to send the request.

Curl

```
curl -X 'PUT' \
  'http://localhost:8000/addresses/8' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "name": "Address 2",
  "latitude": 0,
  "longitude": 0
}'
```

Request URL

```
http://localhost:8000/addresses/8
```

Server response

Code	Details
200	Response body

```
{
  "name": "Address 2",
  "latitude": 0,
  "longitude": 0
}
```

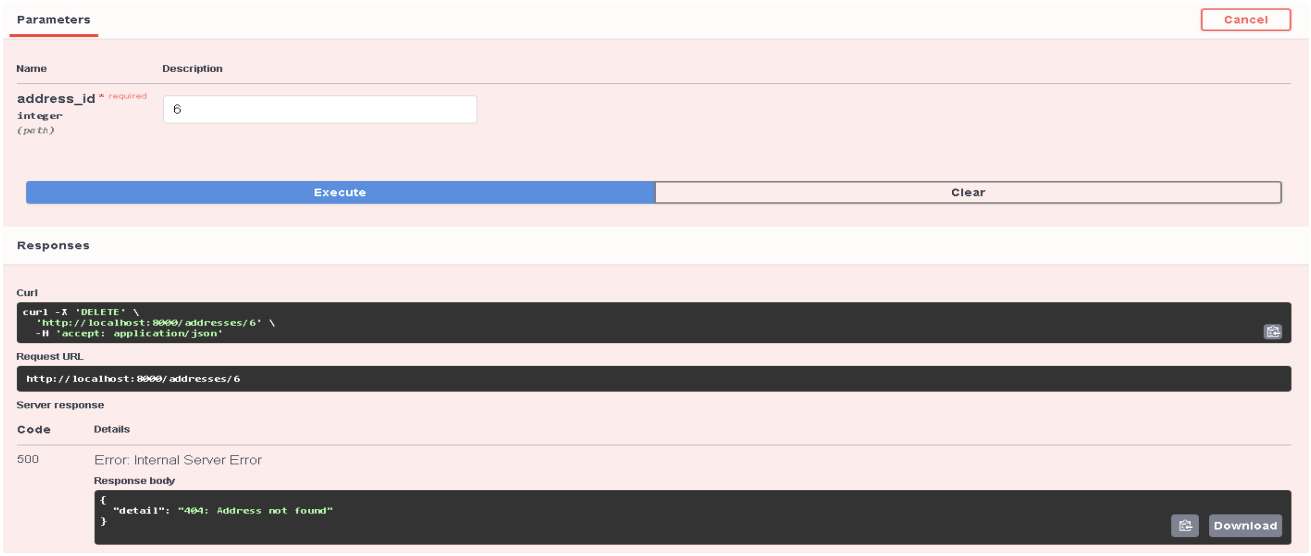
Download

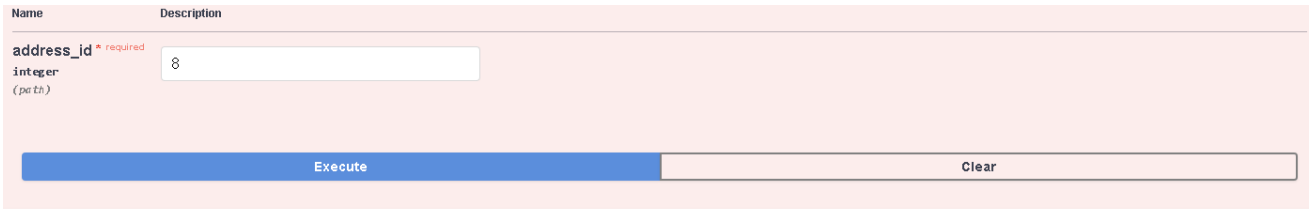
Response headers

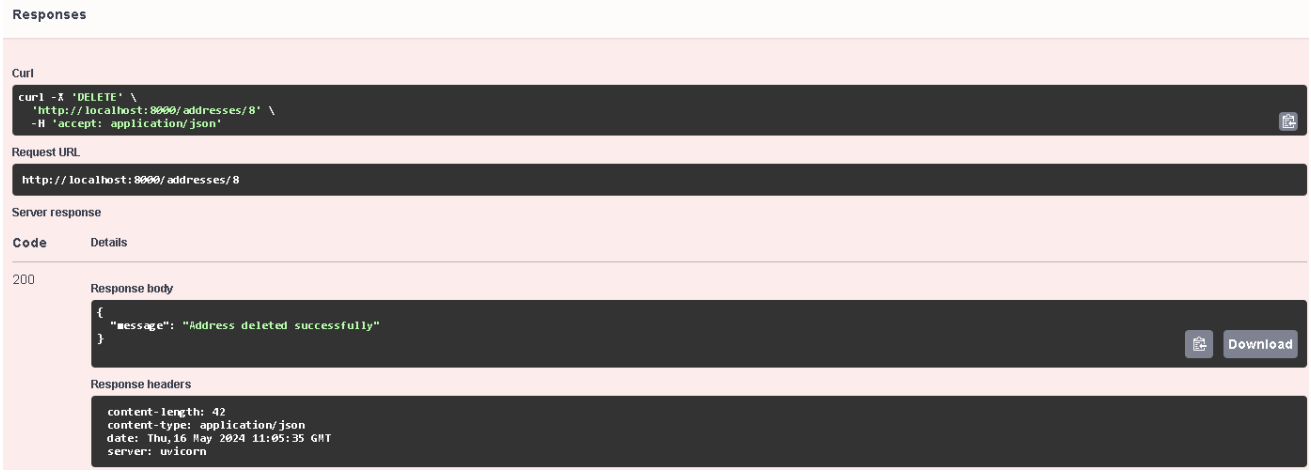
```
content-length: 51
content-type: application/json
date: Thu,16 May 2024 11:04:08 GMT
server: uvicorn
```

e) DELETE /addresses/{address_id}: Delete an address.

- Click the Try it out button.
- Enter the address_id (Number) in the path parameter field.
- Click Execute to send the request.







NOTES

1. The application uses an SQLite database named **addresses.db** to store the address data. The database file will be created automatically if it doesn't exist.
2. The create_sample_db.py and view_db.py scripts are optional and can be used to create a sample database and view the database content, respectively.