

URL Shortener Project

Overview

The URL Shortener project is a web application built using FastAPI, SQLAlchemy, and PostgreSQL. The application provides a simple interface for users to enter a long URL, which is then shortened and stored in a database. Users can access the shortened URL to be redirected to the original long URL. The project also includes a basic frontend, allowing users to interact with the URL shortener service via a web form.

Features

- Shorten Long URLs: Users can input a long URL and receive a shortened version.
- Redirection: Accessing the shortened URL redirects the user to the original long URL.
- Basic Frontend: A simple web interface allows users to easily interact with the service.

Prerequisites

Before running the project, ensure you have the following installed on your machine:

- Python 3.7 or higher
- PostgreSQL
- Git (optional, for version control)

You will also need to install the required Python packages and set up a PostgreSQL database.

Project Setup

1. Clone the Repository

If you have Git installed, you can clone the project repository. Otherwise, download the project files.

```
git clone https://github.com/deveshksh/URL-Shortener-FastAPI.git
cd URL-Shortener-FastAPI
```

2. Set Up a Virtual Environment

It is recommended to use a virtual environment to manage project dependencies.

```
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate
```

3. Install Dependencies

Install the necessary Python packages using pip.

```
pip install -r requirements.txt
```

4. Set Up the PostgreSQL Database

Create a PostgreSQL database:

Log in to your PostgreSQL database and create a new database named url_shortener.

Use the provided script to create the necessary tables in your database.

5. Create the FastAPI Application

Ensure that the main.py file is correctly set up to handle the backend logic, including URL shortening and redirection.

6. Create the Frontend HTML File

The index.html file should be placed in the project directory. This file provides the web interface for users to input long URLs and receive shortened URLs.

7. Run the Application

Start the FastAPI server using Uvicorn:

```
uvicorn main:app --reload
```

8. Access the Application

Open your web browser and navigate to <http://127.0.0.1:8000/>. You should see the URL shortener frontend, where you can input a long URL to shorten.

9. Testing the Application

Enter a long URL in the form on the frontend and click "Shorten".

Copy and visit the shortened URL to ensure it correctly redirects to the original long URL.

Future Enhancements

- Custom Short Codes: Allow users to specify a custom short code.
- Analytics: Track the number of clicks and other metrics for each shortened URL.
- User Accounts: Add user authentication to allow users to manage their shortened URLs.

```

WARNING: StatReload detected changes in 'main.py'. Reloading...
INFO: Started server process [99030]
INFO: Waiting for application startup.
INFO: Application startup complete.
WARNING: StatReload detected changes in 'main.py'. Reloading...
INFO: Shutting down
INFO: Waiting for application shutdown.
INFO: Application shutdown complete.
INFO: Finished server process [99030]
INFO: Started server process [308]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:59489 - "GET / HTTP/1.1" 200 OK
INFO: 127.0.0.1:59682 - "POST /shorten HTTP/1.1" 200 OK
INFO: 127.0.0.1:59682 - "GET /vmI7QK HTTP/1.1" 307 Temporary Redirect
INFO: 127.0.0.1:59694 - "GET /vmI7QK HTTP/1.1" 307 Temporary Redirect
INFO: 127.0.0.1:59720 - "POST /shorten HTTP/1.1" 200 OK
INFO: 127.0.0.1:59720 - "GET /JK5yuS HTTP/1.1" 307 Temporary Redirect
^CINFO: Shutting down
INFO: Waiting for application shutdown.
INFO: Application shutdown complete.
INFO: Finished server process [308]
INFO: Stopping reloader process [30869]
(venv) deveshmac@MacBook-Air fastapi %

```

Overview
Getting started
POST http://127.0.0.1:8000/s
+
No environment

HTTP
http://127.0.0.1:8000/shorten
Save
Share

POST
http://127.0.0.1:8000/shorten
Send

Params
Auth
Headers (8)
Body
Scripts
Settings
Cookies

raw
JSON
Beautify

```

1 {
2   "long_url": "https://[REDACTED].netlify.app"
3 }
4

```

Body
Cookies
Headers (4)
Test Results
200 OK
59 ms
169 B
Save as example

Pretty
Raw
Preview
Visualize
JSON

```

1 {
2   "short_url": "http://127.0.0.1:8000/CEANtn"
3 }

```

Postbot
Runner
Start Proxy
Cookies
Vault
Trash

URL Shortener

Enter a long URL:

`https://www.rabbit.tech/research`

Shorten

Shortened URL: <http://127.0.0.1:8000/JK5yuS>