**Proof of Concept (PoC) – Custom URL Shortener**

**By**: Riya Bhutal (Intern ID: 248)  
**Organization**: DigiSuraksha — Parhari Foundation  
**Tech Stack**: Python, Flask, HTML/CSS

**🔍 Objective:**

To develop a simple, lightweight URL shortener similar to Bitly that allows users to create custom shortened URLs using a **custom alias** (e.g., insta-riya), which can be easily shared and remembered.

**⚙️ Features Implemented:**

1. **Shorten Long URLs** – Converts a long URL into a short one.
2. **Custom Alias Support** – Users can define their own alias instead of random strings.
3. **SQLite-based Storage** – URL mappings are stored in a local SQLite database.
4. **Clickable Result** – Generated short links are clickable from the interface.

**~~📁~~**

**Folder Structure:**

url-shortener-app/

│

├──

app.py

├──

templates/

│ ├──

index.html

│ └──

result.html

├──

static/

│ └──

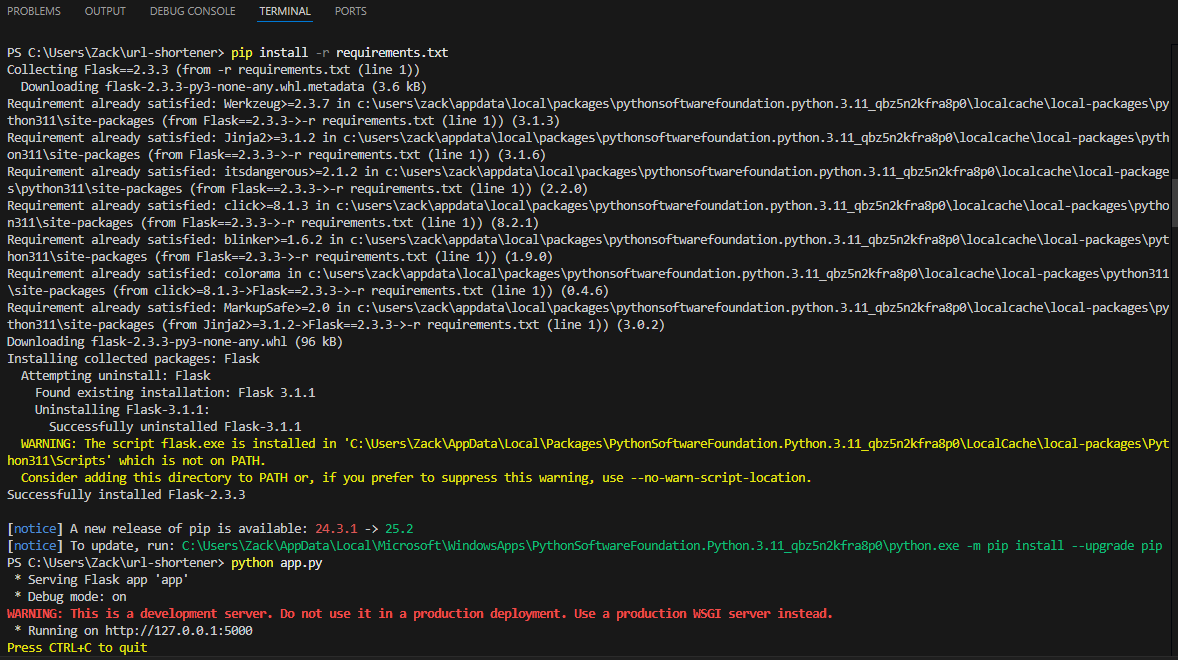
style.css

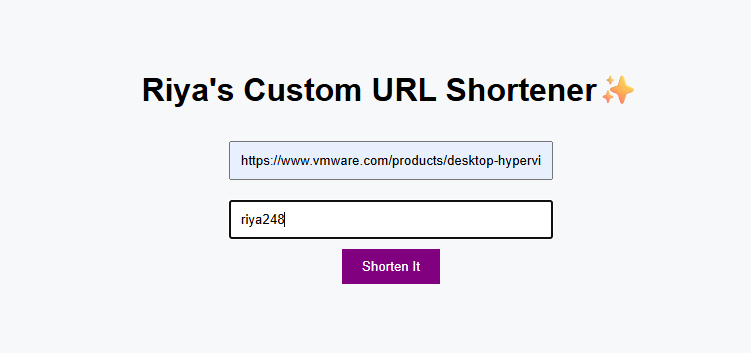
└──

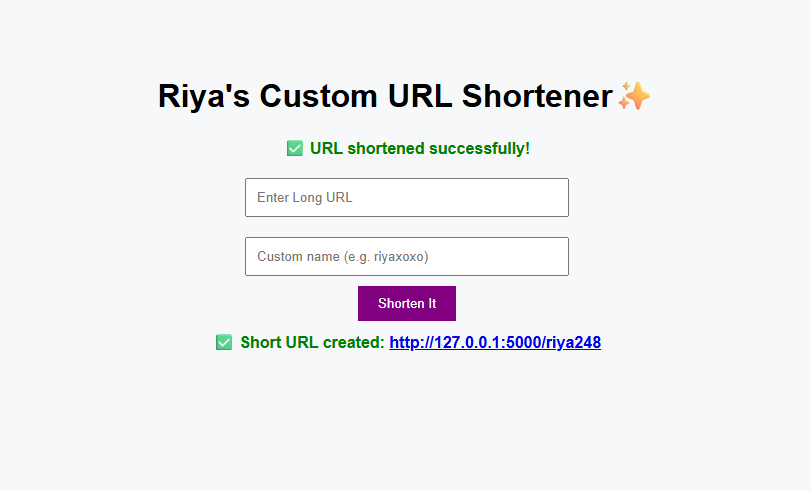
urls.db

**🧠**

**How It Works:**

1. **User Input** – User provides a long URL and a desired alias.
2. **Alias Check** – If the alias is already taken, the app notifies the user.
3. **Store Mapping** – If available, the (alias, long URL) pair is saved in the database.
4. **Redirection Logic** – Visiting the alias (e.g., /insta-riya) redirects to the original URL.





**📜 Example Workflow:**

1. User inputs:
   * **Long URL**: https://www.instagram.com/riyabhutal
   * **Alias**: insta-riya
2. App generates:
   * http://localhost:5000/insta-riya
3. Clicking on the above link redirects to the actual Instagram profile.

**✅**

**Success Message (Frontend Snippet):**

div

<

class=

"success"

>

Shortened URL: <a

href=

"{{ short\_url }}"

target=

"\_blank"

>

short\_url }}</a

>{{

<

/div

>

**🔐**

**Limitations:**

* **No custom domain functionality** – Uses localhost or server IP only.
* **No user accounts or login system**.
* **Analytics not available.**

**🚀 Future Enhancements:**

1. Integrate **custom domain functionality**.
2. Add **click count analytics**.
3. Host the app using **Render**, **Heroku**, or **AWS**.
4. Allow **QR code generation** for each short URL.

**🧪 Testing**

* Validated for alias duplication and collisions.
* Tested for proper redirection and error messages.
* Checked URL encoding for special characters.

**Conclusion:**

This project is a simple but effective clone of Bitly-like tools with a core feature of custom aliasing. It serves as a strong PoC to demonstrate **Flask**, **routing**, **form handling**, and **database operations**.