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

**By**: Gautam Poojari (Intern ID: 251)  
**Organization**: DigiSuraksha - Parhari Foundation  
**Tech Stack**: Python, Flask, HTML/CSS

**🔍 Objective:**

To develop a lightweight URL shortener that allows users to create custom shortened URLs using a **custom alias** (e.g., linkedin-gautam), making them easy to share and remember.

**⚙️ Features Implemented:**

1. Shorten Long URLs — Converts lengthy URLs into compact, shareable links.
2. Custom Alias Support — Users can personalize aliases (e.g., github-gautam).
3. SQLite Database — Stores URL mappings locally for persistence.
4. Clickable Short Links — Generated links are directly accessible from the interface.

**📁 Folder Structure:**

url-shortener-app/

│

├── app.py # Flask backend logic

├── templates/

│ ├── index.html # User input form

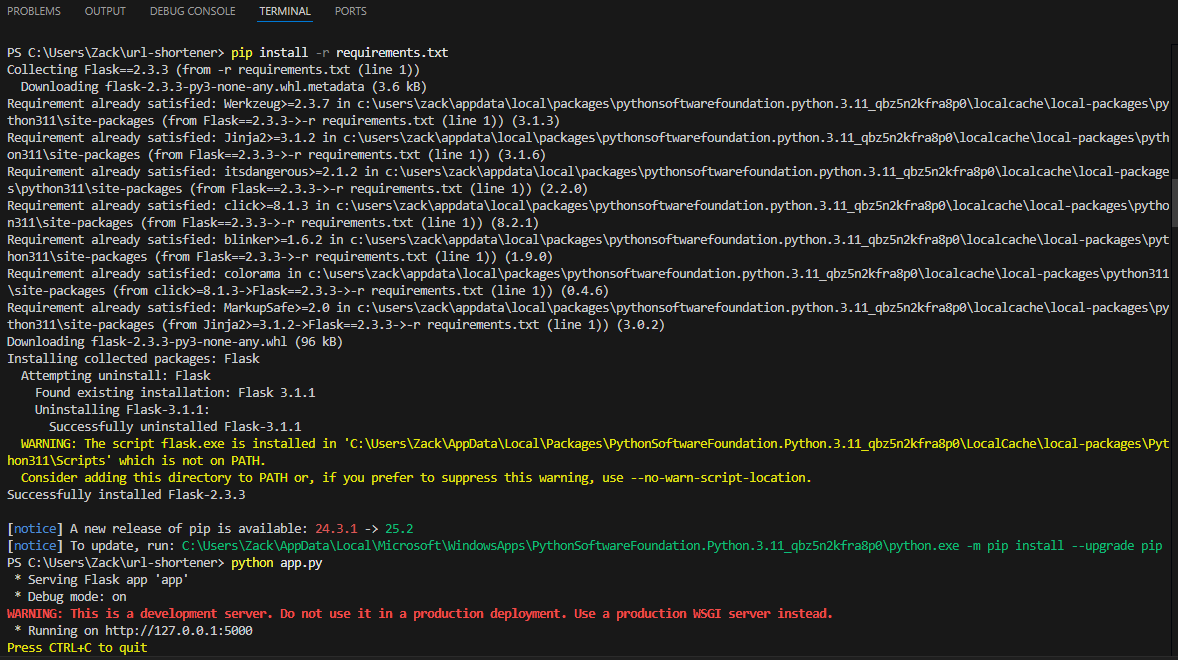
│ └── result.html # Success/error messages

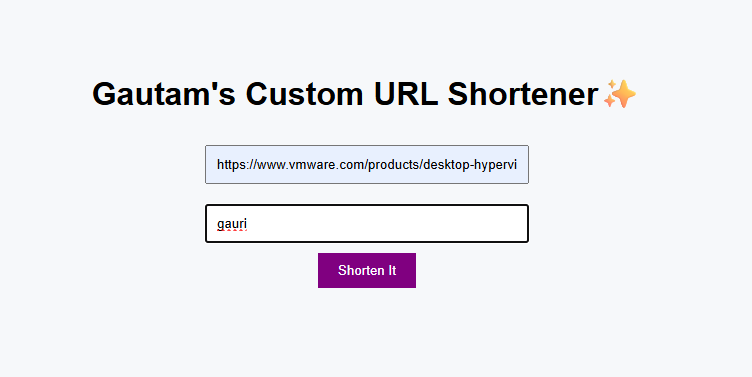
├── static/

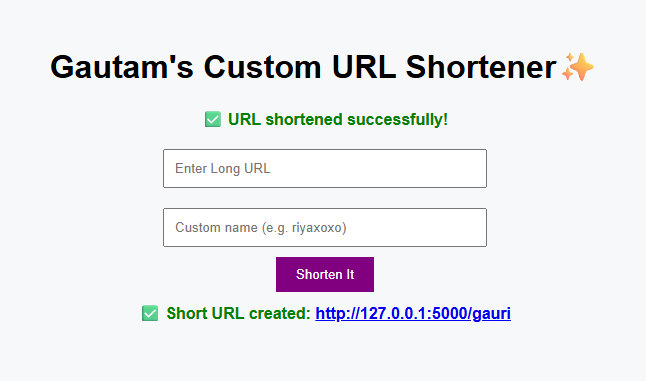
│ └── style.css # Basic styling

└── urls.db # SQLite database

**🧠 How It Works:**

1. **User Input** — Submits a long URL and desired alias.
2. **Alias Validation** — Checks for duplicate aliases in the database.
3. **Storage** — Saves the (alias, long URL) pair if the alias is available.
4. **Redirection** — Visiting /<alias> (e.g., /linkedin-gautam) redirects to the original URL.





**📜 Example Workflow:**

1. **User Inputs:**
   * **Long URL**: https://www.linkedin.com/in/gautam-poojari
   * **Alias**: linkedin-gautam
2. **App Generates:**
   * http://localhost:5000/linkedin-gautam
3. **Result:** Clicking the short link redirects to the LinkedIn profile.

**✅ Success Message (Frontend Snippet):**

<div class="success">

Shortened URL: <a href="{{ short\_url }}" target="\_blank">{{ short\_url }}</a>

</div>

**🔐 Limitations:**

* Localhost Only — No custom domain support.
* No User Accounts — Basic anonymous usage.
* No Analytics — Click tracking not implemented.

**🚀 Future Enhancements:**

1. Custom Domains — Allow users to replace localhost with their domain.
2. Analytics Dashboard — Track clicks and geographic data.
3. Cloud Hosting — Deploy on Render, Heroku, or AWS.
4. QR Codes — Generate QR codes for short URLs.

**🧪 Testing:**

* Validated alias uniqueness and error handling.
* Tested URL redirection with special characters.
* Verified database persistence.

**Conclusion:**

This project demonstrates core web development skills using **Flask**, **SQLite**, and **routing**, serving as a foundation for scalable URL-shortening tools.

**Intern ID**: 251  
**Organization**: DigiSuraksha --- Parhari Foundation