

Introduction to URL Shortener Project

✦ Project Overview

This **Flask-based URL Shortener** allows users to shorten long URLs into **custom or randomly generated short links**. It includes an **Admin Dashboard** to manage links, track clicks, and generate **QR codes** for each shortened URL. The project is **fully responsive** with a clean UI using **Tailwind CSS** and supports **MySQL** as the database.

🔗 Features of This URL Shortener

- ✓ **Shorten Long URLs** – Users can generate short URLs.
 - ✓ **Custom Short Codes** – Users can create their own short links.
 - ✓ **Click Tracking** – Tracks how many times a short link is clicked.
 - ✓ **QR Code Generation** – Every short URL gets a scannable QR code.
 - ✓ **Admin Dashboard** – Admins can view, delete, and track short URLs.
 - ✓ **Secure Login for Admin** – Only admins can manage short URLs.
 - ✓ **Mobile-Responsive UI** – Uses Tailwind CSS for a clean, modern look.
 - ✓ **Deployment on Railway** – Hosted online for global access.
-

🔍 Where People Can Make Mistakes

- 💡 **Incorrect Database Setup** → Forgetting to create the MySQL database or missing required fields.
 - 💡 **Missing Environment Variables** → Hardcoding MySQL credentials instead of using `.env` or Railway variables.
 - 💡 **Wrong QR Code Path** → QR codes need to be stored in a `static` folder (`static/qr/`).
 - 💡 **Not Handling Custom Short Codes Properly** → Allowing duplicate short codes can cause errors.
 - 💡 **Not Resetting MySQL Auto-Increment IDs** → Deleting an entry doesn't automatically fix ID gaps.
-

📦 Libraries Used in This Project

- ✦ **Flask** – Web framework to build the application.
- ✦ **Flask-MySQLdb** – Connects Flask to MySQL.
- ✦ **qrcode** – Generates QR codes for short URLs.
- ✦ **Tailwind CSS** – Modern styling for a clean UI.

- ✦ **os** – Manages file directories for QR code storage.
 - ✦ **random, string** – Generates random short codes.
-

✂ How to Install and Run Locally

1□ Install Dependencies

```
pip install flask flask-mysqldb qrcode
```

2□ Set Up MySQL Database

```
CREATE DATABASE yukesh;  
USE yukesh;  
  
CREATE TABLE urls (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    long_url TEXT NOT NULL,  
    short_code VARCHAR(10) UNIQUE NOT NULL,  
    clicks INT DEFAULT 0,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

3□ Run the Flask App

```
python app.py
```

4□ Access in Browser

Go to: <http://127.0.0.1:5000/>

🚀 Deployment on Railway

- 1□ Create a Railway account & new project
 - 2□ Upload your Flask project from GitHub or manually
 - 3 Set MySQL credentials in Railway Environment Variables
 - 4□ Modify `config.py` to read database credentials from environment variables
 - 5□ Deploy & get your live URL
-

🎯 Final Thoughts

This project is a great way to learn **Flask, MySQL, and web deployment**. By following the correct **database setup, handling errors properly, and using environment variables**, you can create a **secure and scalable** URL shortener! 🚀