Proof of Concept (PoC) Report

Name : Prateek

Intern ID: 166

Tool Name: URL Shortener

Executive Summary

The Modern URL Shortener is a fully client-side web application built with HTML, CSS (Tailwind), and JavaScript. It enables users to generate shortened, memorable links, optionally with custom slugs, and provides enhanced features such as click tracking, link history, and QR code generation. Unlike traditional server-side solutions, this version leverages the browser's localStorage for persistence, making it lightweight, fast, and ideal for personal use without external dependencies.

Objective

The objectives of this PoC include:

• Accepting long URLs and generating short versions.
• Supporting custom short codes (slugs) provided by the user.
• Persisting mappings using localStorage.
• Redirecting short codes to original URLs via hash-based routing.
• Providing QR codes, click counts, and link history for user reference.

Scope

In Scope:

• URL shortening with random or custom codes.
• Persistent storage in the browser via localStorage.
• History tracking with ability to delete individual links or clear all.
• Basic analytics like click count per short link.
• QR code generation and download option.

Out of Scope:

• Server-side database and centralized storage.
• Authentication or multi-user systems.
• Advanced analytics (geo-tracking, device stats).
• Security validation against malicious inputs beyond basic checks.

Tool Overview

The application operates entirely in the browser, making use of Tailwind CSS for styling and JavaScript for core logic. Users can input a long URL, generate a short one (randomly or custom), and manage their history directly. The results section displays the short link, copy functionality, QR code, and click statistics. Persistence is handled using localStorage, ensuring data remains available across sessions on the same browser.

Key Features

• Client-side implementation with no backend dependency.
• Random and custom short code generation.
• Click count tracking for each shortened link.
• QR code integration with download support.
• Local history management (view, delete, clear all).
• Responsive, modern UI with TailwindCSS.

Requirements

Software Requirements:

• A modern web browser (Chrome, Firefox, Edge, Safari).
• TailwindCSS (via CDN).
• Internet connection for optional QR code image resource.

Data Requirements:

• Browser localStorage for mappings and history.
• HTML file (URL_Shortener.html) containing all functionality.

Steps to Run the Tool

1. Download the HTML file (URL_Shortener.html).
2. Open the file in any modern browser.

3. Enter a long URL in the input box.

4. Optionally select and enter a custom short code.

5. Click 'Shorten URL' to generate a short link.

6. Use the copy button, QR code, or click count features as needed.

7. Manage your link history within the same browser session.

Go to the correct directory folder and run: .\URL_Shortener.html

Images PoC
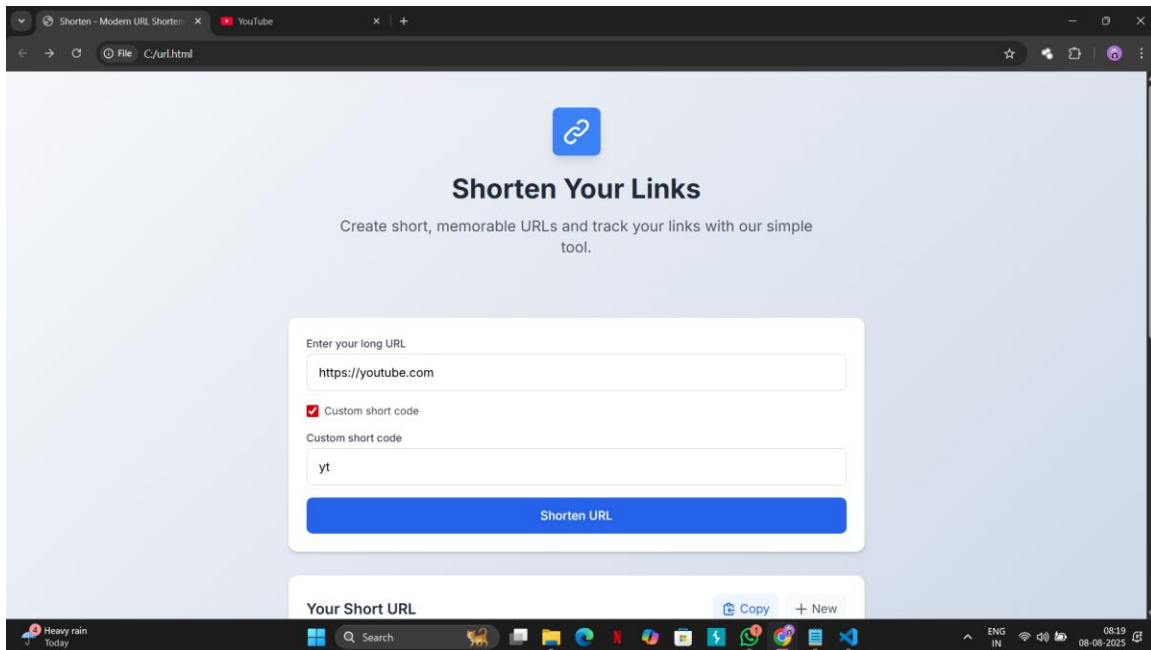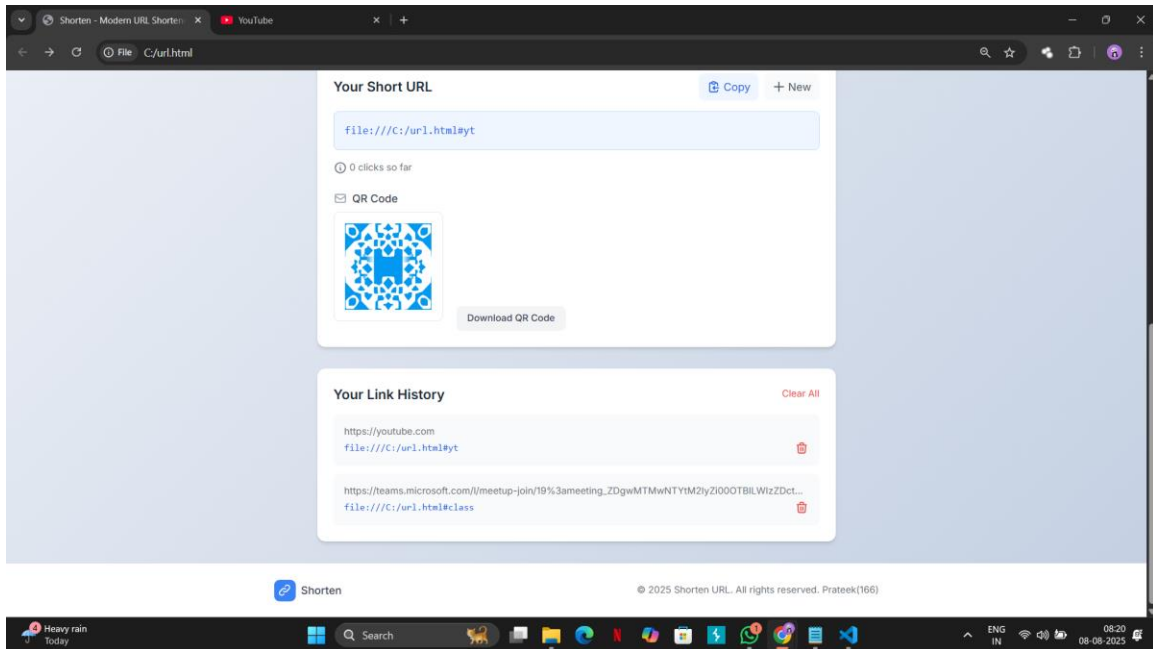
Use Case Scenario

A student preparing a presentation wants to share reference links with the audience. Instead of showing long and complex URLs on slides, the student uses this tool to create short, custom links (e.g., 'notes2025'). These links can be quickly accessed by attendees, and QR codes make mobile access even easier.

Advantages

• No setup required — runs directly in a browser.
• Persistent localStorage ensures links remain across sessions.
• Includes modern usability features like QR code, copy button, and history.
• Custom slug support provides flexibility for memorable short codes.

Threat Impact Analysis

Potential Risks:

• Long URLs may clutter communication without shortening.
• Lack of server validation could allow malicious URLs.

Security Concerns:

• Users could unknowingly shorten and share phishing links.
• Click count manipulation is possible since data is stored client-side.

Mitigation / Future Work:

• Add integration with safe-browsing APIs to validate links.
• Provide an option to export/import history securely.
• Extend to a backend with centralized storage and analytics.

Conclusion

The Modern URL Shortener demonstrates how powerful a purely client-side tool can be. It allows URL shortening, QR code generation, and click tracking without any backend infrastructure. While best suited for personal or educational use, with enhancements such as server-side storage, robust security, and analytics, this tool could evolve into a production-grade service.