

INNOVATION PROJECT REPORT

On

WEB PRIVACY INSPECTOR – A PRIVACY LEAK DETECTION WEBSITE

Submitted by

Name: Namita Rana

Submitted To

BrooskieHub

Date of Submission

30 Oct 2025

Table of Contents

1. Introduction

- 1.1 Background
- 1.2 Problem Statement
- 1.3 Objectives

2. System Overview

- 2.1 Project Scope
- 2.2 Proposed Solution
- 2.3 Features

3. System Design and Architecture

- 3.1 System Architecture
- 3.2 Modules Description
- 3.3 Data Flow

4. Implementation Details

- 4.1 Technologies Used
- 4.2 Frontend Design
- 4.3 Backend Logic

5. Working Principle

- 5.1 URL Input and Fetching
- 5.2 Privacy Data Analysis
- 5.3 Information Type Summary

6. Privacy Scoring and Visualization

- 6.1 Privacy Score Algorithm
- 6.2 Colour Coding and Risk Levels
- 6.3 Output Display

7. Results and Discussion

- 7.1 Sample Test Cases
- 7.2 Analysis of Results
- 7.3 Limitations

8. Deployment and Usage

- 8.1 Deployment Options
- 8.2 Running Locally
- 8.3 User Instructions

9. Innovation and Future Enhancements

- 9.1 Innovative Aspects
- 9.2 Future Enhancements
- 9.3 Potential Applications

10. Conclusion

1. Introduction

1.1 Background

With the rise of digital connectivity, websites have become central to communication, business, and entertainment. However, many sites silently collect personal user data without explicit consent. These hidden mechanisms can range from cookies to complex trackers that profile user behaviour. As online privacy becomes an increasingly important concern, the need for automated privacy analysis tools has grown rapidly. The Web Privacy Inspector project emerges from this very need to empower users with awareness of how their data is being used.

Data protection laws like GDPR and India's DPDP Act have emphasized transparency in digital platforms. Despite these regulations, users often lack the technical knowledge to identify when their privacy is at risk. This project bridges that gap by providing a simple, web-based interface for privacy inspection and risk evaluation.

1.2 Problem Statement

Many internet users unknowingly interact with websites that harvest personal data such as emails, passwords, or location. These data leaks can lead to spam, phishing, identity theft, or targeted tracking. Existing privacy tools are often browser-based extensions or paid software, making them inaccessible to common users. Therefore, there is a strong need for a free, web-based tool that detects, categorizes, and visualizes the data-collection behaviour of any given webpage.

The Web Privacy Inspector seeks to make this complex process simple and accessible. Users should be able to input a website URL and instantly understand what data it collects, how it does so, and whether it poses a privacy threat.

1.3 Objectives

The key objectives of this project include:

- To design a website that can scan and analyze other web pages for privacy leaks.
- To identify different types of data collected such as emails, passwords, cookies, and location.
- To display results in a user-friendly format with a privacy score.
- To use colour codes for representing risk levels for easy visual understanding.
- To educate users about online privacy risks and data transparency.

2. System Overview

2.1 Project Scope

The Web Privacy Inspector focuses on privacy detection and risk visualization. It does not collect user data or interfere with the scanned website; instead, it analyzes the HTML structure and network requests to find data-collection patterns. The system is built to be lightweight, cost-free to host, and accessible on any device with a browser.

It is suitable for students, researchers, and privacy-conscious individuals who want quick insights into how websites handle user information. The scope can later expand into browser plug-ins or cybersecurity assessment tools.

2.2 Proposed Solution

The proposed solution is a web-based privacy analysis platform that accepts a URL and returns a detailed privacy report. The backend fetches and analyzes the webpage using web scraping and pattern detection. It identifies HTML input fields, cookies, and third-party requests. Then it calculates a privacy score and visualizes findings with charts and colour codes.

This approach is simple yet effective. It uses open-source tools and can be hosted freely, ensuring students or developers can deploy it without cost barriers.

2.3 Features

- **Real-time webpage scanning** using the entered URL.
- **Detection of sensitive inputs** like email, password, phone number, etc.
- **Identification of cookies and external trackers.**
- **Privacy scoring system** from 0 to 100 based on risk factors.
- **Information Type Summary** with colour-coded visualization.
- **User-friendly interface** and free online deployment.

3. System Design and Architecture

3.1 System Architecture

The project follows a client-server architecture. The user interacts with the frontend (HTML, CSS, JavaScript), which communicates with a Flask backend via HTTP requests. The backend fetches and parses the target webpage, performs privacy analysis, and returns structured data to the frontend for visualization.

This modular design separates logic from presentation, making the system flexible and scalable.

3.2 Modules Description

The system comprises six main modules:

1. **User Interface Module** – Handles URL input and displays reports.
2. **Web Scraper Module** – Fetches HTML using requests library.
3. **Parser Module** – Uses BeautifulSoup to extract forms, scripts, and cookies.
4. **Detection Module** – Identifies information types and categorizes risk.
5. **Scoring Module** – Calculates privacy score using a weighted algorithm.
6. **Visualization Module** – Displays results in tables, charts, and colour codes.

3.3 Data Flow

The data flow starts with user input. Once a URL is entered, the backend retrieves the HTML, analyzes it, and returns JSON-formatted results. These results are visualized dynamically using JavaScript. The flow ensures minimal delay and efficient processing even on limited hardware.

4. Implementation Details

4.1 Technologies Used

The project is built using:

- **Frontend:** HTML5, CSS3, JavaScript, Chart.js
- **Backend:** Python Flask
- **Libraries:** BeautifulSoup, Requests, Regex
- **Deployment:** Render or GitHub Pages (Free)
- **IDE:** VS Code

These technologies are open-source, lightweight, and ideal for educational projects.

4.2 Frontend Design

The frontend features a clean, responsive interface with an input box for URLs and a “Scan Now” button. The results section includes a privacy summary, risk table, and score meter. Chart.js is used to render the privacy score visually, while CSS ensures clarity and mobile compatibility.

4.3 Backend Logic

The backend uses Flask to handle API requests. It scrapes target pages with requests and BeautifulSoup. Regular expressions detect keywords and patterns related to data collection. A scoring algorithm evaluates risk factors and assigns ratings. The backend then sends this structured data to the frontend as JSON.

5. Working Principle

5.1 URL Input and Fetching




When a user enters a URL, the backend fetches the web page's content securely. The system ensures it doesn't store or modify any data. This read-only operation ensures user privacy and ethical analysis.

5.2 Privacy Data Analysis

The fetched HTML is parsed to identify forms, cookies, and scripts. The system looks for sensitive fields like `<input type="email">`, `<input type="password">`, or third-party domains. Each element contributes to the overall risk score.

5.3 Information Type Summary

The system generates an "Information Type Summary" showing what data the webpage may collect and assigns a colour-coded risk level:

-  **High:** Passwords, IP, or biometric data.
-  **Medium:** Emails, location, or phone numbers.
-  **Low:** Cookies or minimal information.

6. Privacy Scoring and Visualization

6.1 Privacy Score Algorithm

A numeric score (0–100) is calculated based on detected risks. Secure pages with HTTPS, minimal tracking, and no sensitive input fields receive higher scores. Risky pages lose points accordingly.

6.2 Colour Coding and Risk Levels

Colour codes enhance readability:

□ Green (Safe), □ Yellow (Moderate), ● Red (High Risk).

This visual system allows users to instantly understand privacy levels.

6.3 Output Display

The frontend shows a summary table, privacy score chart, and textual remarks. The design ensures clarity even for non-technical users.

7. Results and Discussion

7.1 Sample Test Cases

Testing included various websites — social platforms, e-commerce sites, and blogs. High-risk pages with multiple trackers scored below 50, while secure portals scored above 80.

7.2 Analysis of Results

Results showed that even popular sites collect multiple user data types. The tool effectively distinguishes safe vs risky domains, highlighting its educational and practical value.

7.3 Limitations

- Cannot access pages requiring login.
- Limited to static content (no JavaScript execution).
- May misclassify complex tracker scripts.

8. Deployment and Usage

8.1 Deployment Options

The system can be deployed for free on **Render**, **Vercel**, or **GitHub Pages**, making it accessible worldwide without server costs.

8.2 Running Locally

Users can run it on their system using simple commands:

```
pip install flask requests beautifulsoup4
```

```
python app.py
```

Then open `http://localhost:5000` in the browser.

8.3 User Instructions

Users simply enter a website URL and press “Scan Now.” The results are generated automatically within seconds, offering an easy learning and evaluation experience.

9. Innovation and Future Enhancements

9.1 Innovative Aspects

This project uniquely combines privacy awareness, data visualization, and open access. Unlike traditional scanners, it focuses on user education and free availability.

9.2 Future Enhancements

Planned upgrades include real-time browser extensions, integration of machine learning for risk prediction, and APIs for automated privacy auditing.

9.3 Potential Applications

The project can be used in educational institutions, cybersecurity training, and digital privacy awareness campaigns. It can also serve as a foundational prototype for privacy compliance auditing tools.

10. Conclusion

The **Web Privacy Inspector** successfully demonstrates how technology can promote transparency and data protection. It empowers users to understand the privacy risks of websites they visit daily. The project stands out for its innovation, simplicity, and zero-cost deployment model — making it a valuable contribution to both cybersecurity education and user privacy awareness.