Information Security

(Lab-Project)

Submitted By:

Sufwan Masood	2023-CS-710
Hamna Nasir	2023-CS-714
Muhammad Abdul Rehman Wahla	2023-CS-717

Submitted To:

Ms. Zoha Sohail

Course: IS LAB

Department of Computer Science

University of Engineering and Technology, Lahore

New Campus

Contents

Project Overview 3

Problem Statement 3

Objectives 3

Methodology 5

i. Requirement Analysis 5

ii. Technology Stack Selection 5

iii. Design and Prototyping 5

iv. Development Phases 5

Phase 1: Setup and Authentication 5

Phase 2: Policy Generation 6

Phase 3: Policy Management and Download 6

Phase 4: UI Enhancements 6

v. Testing and Debugging 6

vi. Deployment and Documentation 6

Implementation 7

1. Project Structure 7

2. Database Schema 7

3. UI Enhancements 8

i. Login Page: 8

ii. Dark Theme: 8

iii. Responsive Design: 8

Conclusion 10

Project Overview

The Privacy Policy Generator is a web application built using Flask, a Python web framework, designed to help users create customized privacy policies for their websites. The application allows users to register, log in, and generate privacy policies by providing details about their website and data collection practices. It supports compliance with regulations such as GDPR, CCPA, and LGPD and provides options to download the generated policies as text files.

Problem Statement

Many small businesses, bloggers, and website owners lack the legal expertise or resources to create privacy policies that comply with international data protection regulations like GDPR, CCPA, and LGPD. Manually drafting a privacy policy can be time-consuming, error-prone, and expensive if legal consultation is required. Existing solutions may not offer sufficient customization or may lack user-friendly interfaces for non-technical users.

Objectives

- **User-Friendly Policy Generation:**
  - Provide an intuitive interface for users to generate privacy policies tailored to their website's needs.
  - Allow users to specify compliance requirements (GDPR, CCPA, LGPD) and data collection practices (e.g., cookies, personal information, analytics).
- **Secure User Authentication:**
  - Implement a secure login and signup system with password hashing to protect user accounts.
  - Include a "Remember me" feature for persistent sessions.

- **Customizable and Downloadable Policies:**
  - Generate detailed privacy policies in Markdown format based on user inputs.
  - Allow users to view, manage, and download their policies as plain text files.
- **Modern and Responsive Design:**
  - Create a visually appealing and responsive design using Bootstrap and custom CSS.
  - Enhance the login page with a frosted glass effect for input fields and a centered favicon image for branding.
- **Scalability and Maintainability:**
  - Use MongoDB for scalable data storage of user accounts and policies.
  - Structure the codebase with modular templates and routes for easy maintenance and future enhancements.

The Privacy Policy Generator addresses these issues by:

- Providing a free, web-based tool to generate privacy policies with minimal effort.
- Offering customization options to meet specific compliance and data collection needs.
- Ensuring the generated policies are accessible, downloadable, and easy to manage.
- Implementing a secure and visually appealing interface to enhance user experience.

Methodology

i. Requirement Analysis

- Identified the need for a tool to generate privacy policies for websites.
- Determined key features: user authentication, policy creation with compliance options, policy management, and download functionality.
- Defined design requirements: dark-themed, modern interface with frosted glass effect for input fields and a centered favicon on the login page.

ii. Technology Stack Selection

- **Backend:** Flask (Python)
- **Database:** MongoDB (NoSQL)
- **Frontend:** Bootstrap, Font Awesome, and custom CSS
- **Templating:** Jinja2 (built into Flask)

iii. Design and Prototyping

- Designed a dark-themed interface with a focus on usability and aesthetics.
- Prototyped the login page with a centered favicon image, frosted glass input fields, and a clean layout.
- Created wireframes for all key pages.

iv. Development Phases

Phase 1: Setup and Authentication

- Setup Flask with MongoDB integration.

- Implemented user signup/login with password hashing and session management.

Phase 2: Policy Generation

- Developed policy creation form and a dynamic policy generator.

Phase 3: Policy Management and Download

- Implemented viewing, managing, and downloading policies.

Phase 4: UI Enhancements

- Customized login page design and applied frosted glass effect.

v. Testing and Debugging

- Validated security features and user sessions.
- Tested all policy generation combinations.
- Verified the download feature across browsers.
- Tested UI responsiveness.

vi. Deployment and Documentation

- Deployed the application locally for testing.
- Documented the project objectives, problem statement, implementation details, and methodology.

Implementation

1. Project Structure

- **app.py:** Main Flask application (routes, database logic, policy generation).

- **templates/:**
  - base.html — Base layout.
  - login.html — Login page with frosted glass design.
  - signup.html — Signup page.
  - dashboard.html — User dashboard.
  - create_policy.html — Form for creating policies.
  - my_policies.html — View/manage policies.
  - view_policy.html — Single policy view and download page.

- **static/:**
  - img/favicon.png — Centered favicon image.
  - css/style.css — Custom CSS.

2. Database Schema

- **Users Collection (users_collection):**
  - _id: ObjectId (unique)
  - username: String (unique)
  - password: String (hashed)
  - name: String
  - email: String (unique)
  - created_at: DateTime
  - last_login: DateTime

- **Policies Collection (policies_collection):**
  - _id: ObjectId (unique)
  - user_id: String (user reference)
  - website_name: String
  - website_url: String
  - company_name: String
  - content: String (Markdown policy)
  - created_at: DateTime
  - last_updated: DateTime
  - gdpr_compliant: Boolean
  - ccpa_compliant: Boolean
  - lgpd_compliant: Boolean
  -

3. UI Enhancements

i. Login Page:

- Replaced lock icon with favicon (64px x 64px).
- Removed text headers.
- Applied frosted glass effect to input fields.
- Set input text color to white.

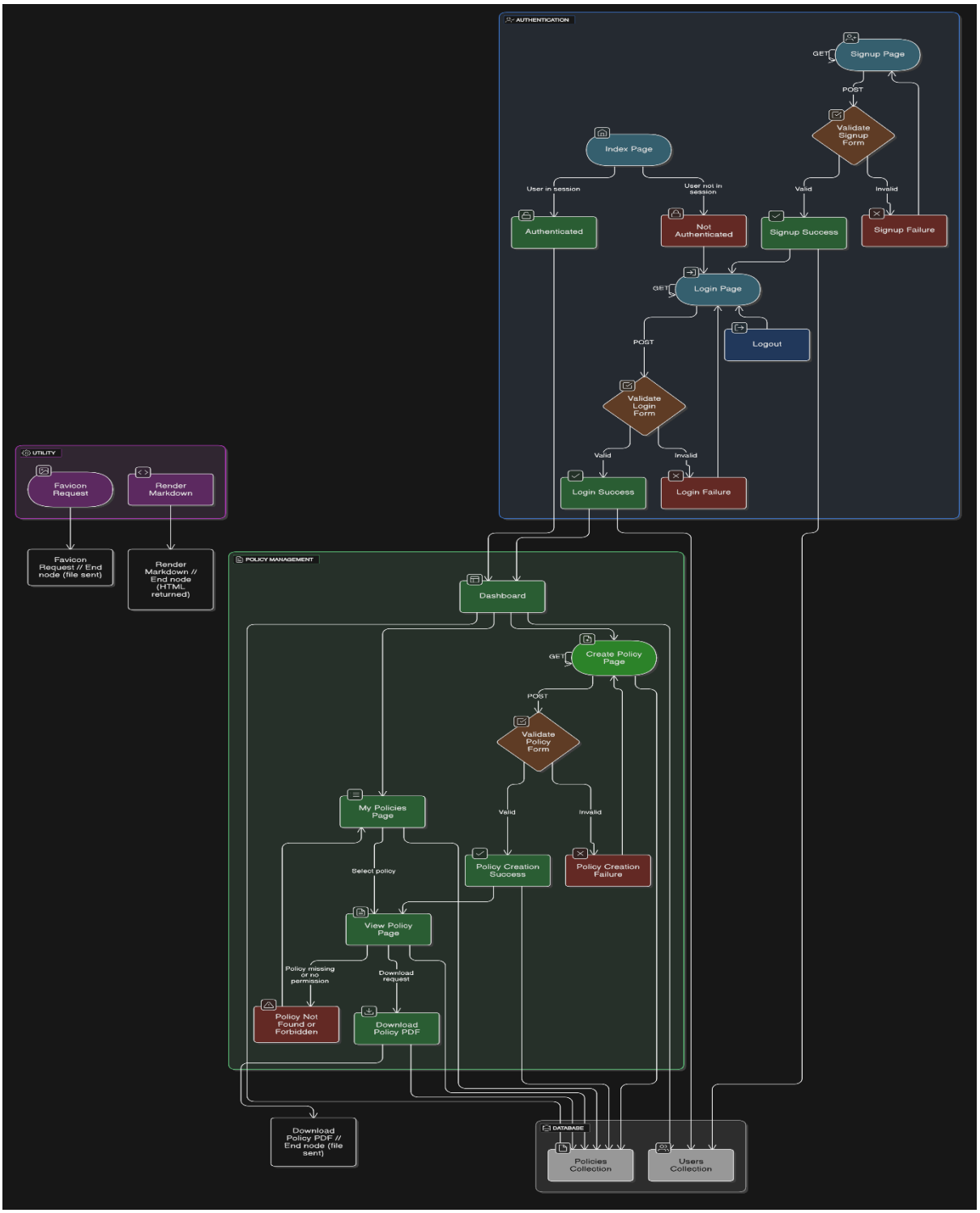
ii. Dark Theme:

- Applied a dark color scheme throughout.
- Used blue accent colors for buttons and links.

iii. Responsive Design:

- Used Bootstrap's grid system for responsiveness.
- Added custom hover effects and transitions.

Flow Diagram:



Conclusion

The Privacy Policy Generator successfully addresses the need for an accessible and user-friendly tool to create privacy policies. It combines a secure backend (Flask, MongoDB) with a modern frontend (Bootstrap, custom CSS) to offer a seamless experience for users to generate, manage, and download their privacy policies. The frosted glass effect and centered favicon on the login page enhance the visual appeal, while the step-by-step policy creation process ensures ease of use.