SCRUM BOOK

SUBMITTED BY

MUHAMMED ANAS A

(NCE22MCA-2031)

SUBMITTED TO

ASHISH L

(ASSISTANT PROFESSOR)

PROJECT TOPICS

SUBMITTED ON

29 JAN 2024

TOPIC 1:

USER BEHAVIOUR AUTHENTICATION SYSTEM

Creating a secure user behaviour-based authentication system using unique typing patterns instead of Passwords. Capture key press details to form distinct user profiles, analysed by an AI Model. Retraining mechanism adapts to changing habits. Anomalies trigger discreet user notifications and enhancing security and user functions.

TOPIC 2:

CTF (CAPTURE THE FLAG)

A website that hosts a CTF Competition. Users or Ethical Hackers can create a profile and participate in the competition. They can submit the Flags for each task or levels to system. The website displays all their ranks, levels, overall performance. This helps the beginners in the field of cybersecurity to make them more familiar with Bug Bounty programmes and other activities.

TOPIC 3:

STEGANOGRAPHY WITH CRYPTOGRAPGHY

This project involves the development of a website that use to conceal the information with the help of steganography along with cryptography. Cryptography is used to encrypt and decrypt the message and later this encrypted message is made hidden using steganography, a technique that allows the hidden embedding of information within image or other file system.

ABSTRACT

ON

CIPHERVEIL

STEGANOGRAPHY WITH CRYPTOGRAPHY

SUBMITTED ON

9 FEB 2024

RESEARCH AREA

This project involves the development of a website that use to conceal the information with the help of steganography along with custom cipher.

PROBLEM STATEMENT

Military and defence organizations rely on various methods to communicate covertly with each other or their operatives. However, many of these methods are vulnerable to interception, decryption, or exposure by adversaries. Therefore, there is a need for more secure and robust techniques to ensure the confidentiality and integrity of sensitive information especially while sharing the information with RAW agents or spy

EXISTING SYSTEM

In the current landscape, confidential information is often exchanged within private networks, which are presumed to be secure. However, despite these precautions, there remains a potential vulnerability. Unauthorized access, whether due to human error, system flaws, or malicious intent, can jeopardize the integrity of the entire system. If such a breach occurs, the consequences are severe: sensitive information may fall into the wrong hands, compromising not only security but also the trust of stakeholders. To mitigate this risk, it is crucial to explore advanced methods of secure communication

PROPOSED SYSTEM

In this system, custom ciphers are employed to encode or decode sensitive information. These ciphers can be tailored to meet specific security requirements, ensuring robust protection for confidential data.

Additionally, the system incorporates steganography, a technique that allows the hidden embedding of information within seemingly innocuous files, such as images. By concealing the encoded data within an image, the system achieves a dual layer of security: the encrypted content remains confidential, while the steganographic cover ensures that the presence of hidden data is inconspicuous.

Overall, this proposed system combines cryptographic strength with covert communication, making it a powerful tool for secure information exchange and protection against unauthorized access.

OBJECTIVES

The main objective is to create a website which can perform the following features

- Steganography of data
- Custom cipher creation and usage

REQUIREMENTS

HARDWARE REQUIREMENTS:

PC or Laptop

- Processor i5 or more
- 4GB RAM

SOFTWARE REQUIREMENTS:

OS: Windows 10 or 11

IDE: VS code

DATABASE: PostgreSQL

FRAMEWORK: Django Framework

LANGUAGES:

- Python for Backend
- HTML, CSS, JS & Jinja for frontend

MODULES

ADMIN

This module encompasses the administration functions and systems within the overall system, including data handling and user management.

USER

This module encompasses user registration, login, and all other user-related activities.

STEGANOGRAPHY

This module mainly encompasses of codes and algorithms related to steganography.

CUSTOM CIPHER

This module is related to cryptography, where the messages or information is encrypted or decrypted using special algorithms and techniques which makes unauthorized users to decrypt and gain the message or information.

SCRUM REVIEW

ON

CIPHERVEIL

SUBMITTED ON

27 FEB 2024

MODULES DESCRIPTION

ADMIN

This module encapsulates administrative functionalities and superuser capabilities, including administrative site configurations and database tables pertinent to administration for ensuring the smooth operation of a website. It is automatically generated by Django during project creation. Developers can initiate the creation of a superuser by executing the following command within the project's primary directory:

```
python manage.py createsuperuser
```

Subsequently, developers are prompted to input the desired username, password, and email address. These details can be modified later by accessing the admin site.

Additionally, this module incorporates numerous built-in functions provided by Django, facilitating rapid, streamlined, and effective website development and utilization.

Primarily, the core database tables associated with this module include:

- auth_group
- auth_group_permissions
- auth_permissions
- auth_user
- auth_user_groups
- auth_user_user_permissions

CIPHERVEIL

This is the core module which contains basic and overall functions. Below mentioned files are present in this module

1. settings.py:

This file contains settings for your Django project. It includes database configuration, static files settings, middleware configuration, and other project-specific settings. You can customize various aspects of your Django application by modifying this file.

2. urls.py:

This file is responsible for defining the URL patterns for your Django project. It maps URL patterns to views, allowing Django to determine which view should handle a particular HTTP request. You define the routing logic in this file, specifying how URLs are mapped to specific views and controllers.

3. wsgi.py:

WSGI stands for Web Server Gateway Interface. This file is used to expose your Django application to a WSGI server, which is a standard interface between web servers and Python web applications. It provides a way for external web servers (like Apache or Nginx) to communicate with your Django application.

4. asgi.py:

ASGI stands for Asynchronous Server Gateway Interface. Like WSGI, this file is used for exposing your Django application to an ASGI server, which supports asynchronous communication. ASGI is particularly useful for handling long-lived connections and handling multiple concurrent requests efficiently.

These files are crucial components of a Django project and are typically found at the top level of the project directory. They help in configuring settings, defining URL patterns, and connecting your Django application to web servers using WSGI or ASGI.

FRONTLINEAPP

This module incorporates features related to the Homepage, Index page, Sign-in functionality, and a foundational HTML template utilized as an extension for other HTML files throughout the website. It encompasses the structuring of static and templates directories, accommodating CSS

and JavaScript files, along with Sign-in, Home, Index, and base HTML files.

Within the module, the views.py and urls.py files contain the code responsible for rendering the Homepage, Index Page, Sign-in page, and Sign-out functionality. Currently, the module utilizes the auth_user table for the purpose of user authentication during the sign-in process.

CONTACTAPP

This module manages the CRUD (Create, Read, Update, Delete) operations for other users to whom the user sends messages, encompassing their identification and names. Presently, no table has been instantiated for this module; however, there are plans to implement one in the future, enabling users to store contact details of other users.

STEGANOAPP

This module is dedicated to the implementation of steganography on the website, a technique that facilitates the covert embedding of information within seemingly benign files, such as images. By concealing encoded data within an image, the system achieves a dual layer of security: the encrypted content remains confidential, and the steganographic cover ensures that the presence of hidden data is inconspicuous.

As of now, this module does not incorporate any database tables or templates, but there are plans to integrate them in the future.

DATABASE TABLES

- auth_group
- auth_group_permissions
- auth_permissions
- auth_user
- auth_user_groups
- auth_user_user_permissions
- Django_admin_log
- Django_migrations
- Django_content_type
- Django_sessions

LANGUAGES

FRONTEND

- HTML
- CSS
- JS
- JINJA

BACKEND

- PYTHON

FRAMEWORK

DJANGO

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Developed in 2003 by Adrian Holovaty and Simon Willison, Django follows the "don't repeat yourself" (DRY) principle and emphasizes reusability and pluggability of components, rapid development, and the principle of "explicit is better than implicit."

Here are some key features of Django:

- Object-Relational Mapping (ORM): Django provides an abstraction layer on top of databases, allowing developers to interact with databases using Python objects. This simplifies database interactions and makes database migrations easier to manage.

- Admin Interface: Django automatically generates a customizable admin interface for managing site content. Developers can use this interface to perform CRUD (Create, Read, Update, Delete) operations on their application data without writing additional code.

- URL Routing: Django uses a clean and elegant URL routing system that allows developers to map URLs to Python functions, called views. This makes it easy to design clean and readable URLs for web applications.

- Template System: Django provides a powerful template system that allows developers to separate the presentation layer from the business logic. Templates are HTML files with embedded Django template language that allows dynamic content rendering.

- Form Handling: Django simplifies form handling by providing form classes that can be used to generate HTML forms, validate user input, and handle form submissions.

- Security Features: Django includes built-in protection against many common security threats, such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and clickjacking.

- Authentication and Authorization: Django provides a robust authentication system that allows users to authenticate via username and password, social authentication, or custom authentication backends. It also includes a flexible authorization system for controlling user access to different parts of the application.

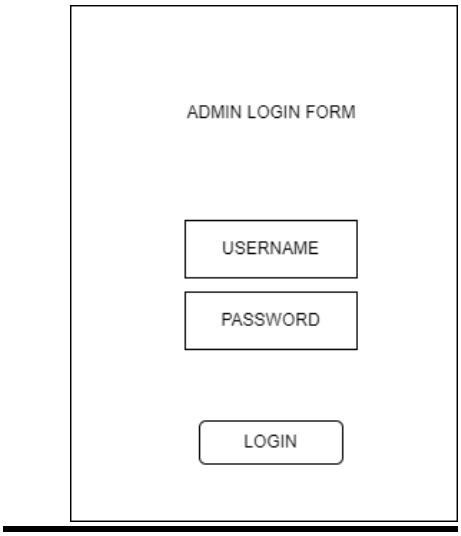
- Internationalization and Localization: Django supports internationalization (i18n) and localization (l10n) out of the box, making it easy to build applications that support multiple languages and locales.

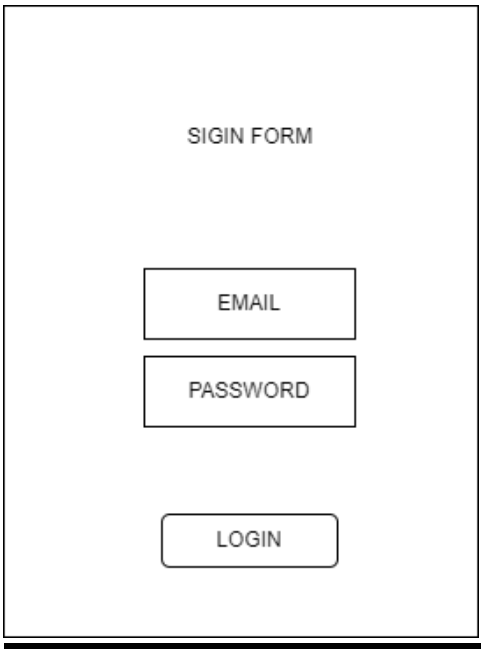
- Middleware: Django middleware is a framework of hooks into Django's request/response processing. It is a lightweight, low-level plugin system for globally altering Django’s input or output.

- Extensibility: Django is highly extensible, with a large ecosystem of third-party packages and libraries available to add additional functionality to Django projects.

Django's design philosophy and built-in features make it a popular choice for building a wide range of web applications, from simple websites to complex web platforms. It powers some of the world's most popular websites and web applications, including Instagram, Pinterest, Disqus, and many others.

USER INTERFACE





GIT

All codes and documents are committed and upload to below mentioned GitHub repository. Scan the QR code below.