Password Toolkit - Documentation

A Modern, Educational Web Toolkit for Password Security, Hashing, Encryption, and More

Author and Developer:
Mohammad Bilal

Date:
April 24, 2025

Abstract

This documentation presents the design, motivation, features, and technical details of Password Toolkit, a modern, educational web application for exploring password security. The toolkit provides a suite of tools including password generation, strength testing, hashing, encryption, steganography, password leak checking, and rainbow table demo. The project targets students, developers, and anyone interested in understanding real-world password and security practices via a user-friendly, interactive platform.

Table of Contents

List of Figures

List of Tables

1 Project Overview

1.1 Description

Password Toolkit is a web-based application developed with frontend React.js and Tailwind CSS. It provides a collection of educational password and security tools, including a password generator, strength tester, encryption and hashing playgrounds, steganography, password leak checker, hash generator, and a rainbow table demonstration. The application features a modern, responsive UI with full dark/light theme support and requires authentication to access all functionality.

1.2 Motivation

The motivation for Password Toolkit stems from the need to:

- Educate users and students about password security and modern cryptographic practices.
- Provide a practical and interactive environment to learn about hashing, encryption, and attacks like rainbow tables.
- Offer a visually appealing, easy-to-use toolkit for both experimentation and real use.
- Encourage adoption of strong password habits and secure credentials management.

1.3 Objectives

- Provide a comprehensive set of password and security utilities in a single, accessible web application.
- Demonstrate cryptographic concepts (hashing, encryption, steganography) in a practical and visual way.
- Improve user understanding of password best practices (strength, uniqueness, randomness, and compromise checking).
- Encourage experimentation with real cryptographic primitives and educate about potential attacks.

1.4 Target Audience

- Students and educators in cybersecurity, computer science, and IT.
- Developers seeking tools for password and security testing.
- Anyone interested in learning about modern password security practices and cryptography.

1.5 Features

Table 1: Password Toolkit Features

Feature	Description
Password Generator	Generates strong, random passwords based on user criteria (length, symbols, etc.).
Password Strength Tester	Analyzes and scores password strength with actionable feedback.
Encryption Playground	Interactive tools for AES, RSA, and Base64 encryption/decryption.
Hashing Playground	Hash text with MD5, SHA1, and SHA256, and visualize hash outputs.
Cracker Demo	Simulate cracking passwords using brute-force and dictionary attacks.
Steganography	Hide and reveal messages (simple text) within images using encoding/decoding.
Password Leak Checker	Checks if a password appears in public data breaches (demo, client-side).
Hash Generator	Quickly generate hashes for files or text inputs.
Rainbow Table Demo	Visual demonstration of rainbow table attacks using a precomputed hash table.
Responsive UI & Theming	Modern, mobile-friendly interface with dark/light theme toggle.

1.6 Operational Details

Table 2: Technology Stack

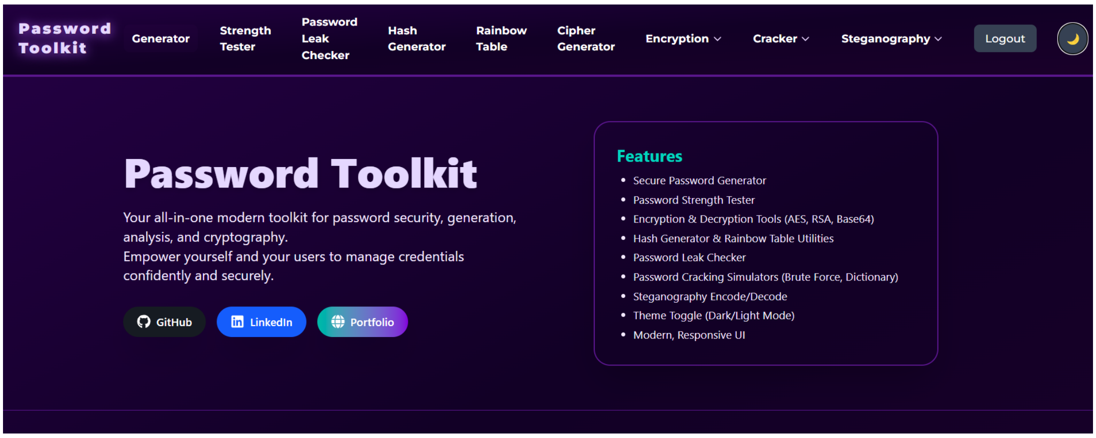
Framework	React.js with functional components and hooks
UI Styling	Tailwind CSS, responsive design, custom SVG icons
Language	JavaScript (ES6+)
Client-side Crypto	CryptoJS, WebCrypto API, custom implementations
State Management	useState, useEffect, useRef hooks; Context API for theming
Routing	React Router
Hosting	GitHub Pages
Coding Style	Modern functional React, idiomatic JS, modular separation

2 Use Cases

2.1 Landing Page

Table 3: Landing Page

Name	Landing Page
Actor	Authenticated User
Description	Entry point with project description, features, developer info, and quick navigation to all toolkit features.



Password Toolkit

Your all-in-one modern toolkit for password security, generation, analysis, and cryptography.
Empower yourself and your users to manage credentials confidently and securely.

Features

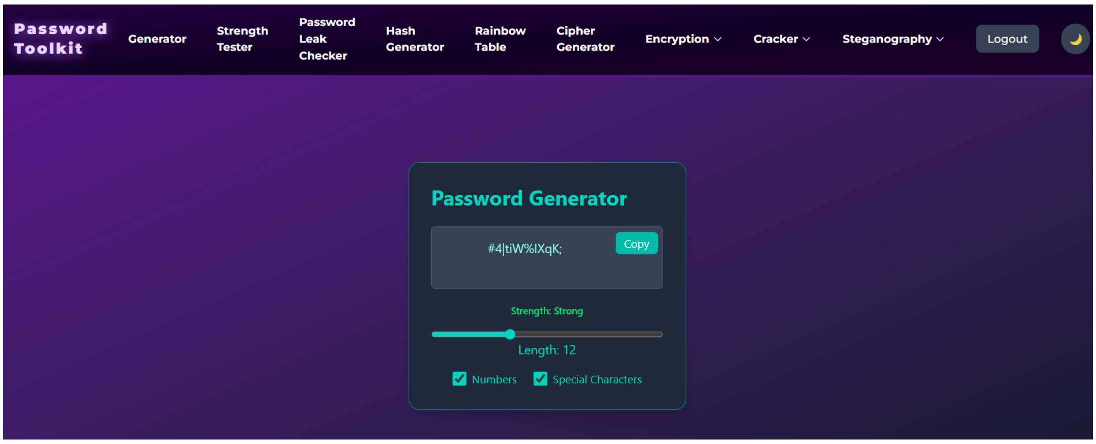
- Secure Password Generator
- Password Strength Tester
- Encryption & Decryption Tools (AES, RSA, Base64)
- Hash Generator & Rainbow Table Utilities
- Password Leak Checker
- Password Cracking Simulators (Brute Force, Dictionary)
- Steganography Encode/Decode
- Theme Toggle (Dark/Light Mode)
- Modern, Responsive UI

Figure 1: Landing Page

2.2 Password Generation and Strength Testing

Table 4: Password Generation and Strength Testing

Name	Password Generator / Strength Tester
Actor	Authenticated User
Description	Users generate passwords with custom settings and check strength with immediate feedback.



Password Generator

#4|tiW%lXqK;

Strength: Strong

Length: 12

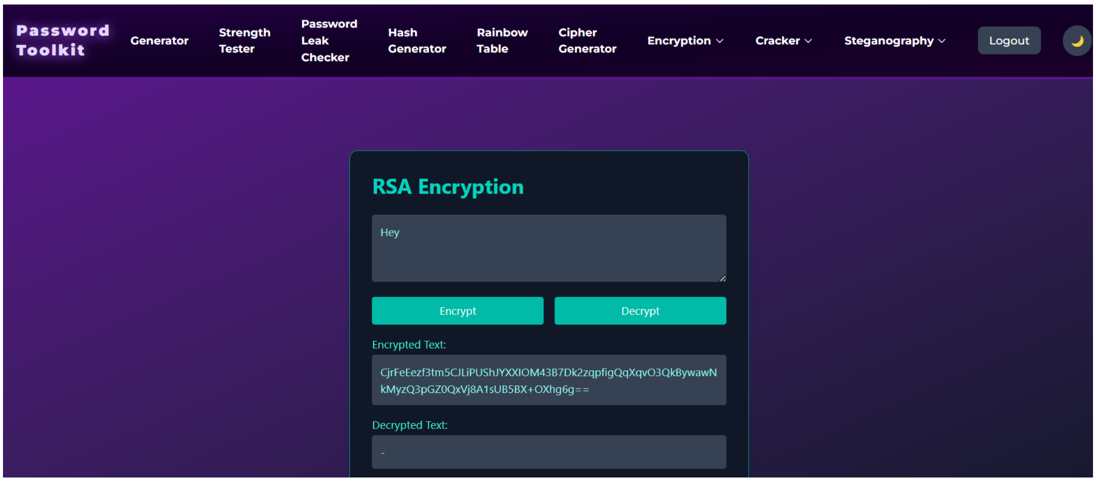
☑ Numbers ☑ Special Characters

Figure 2: Password Generator and Strength Tester

2.3 Encryption / Decryption Playground

Table 5: Encryption Playground

Name	Encryption Page
Actor	Authenticated User
Description	Users can encrypt/decrypt messages using AES, RSA, and Base64. Useful for educational demos.



Hey

Encrypted Text:

CjrFeEezf3tm5CJLiPUShJYXXIOM43B7Dk2zqpfigQqXqvO3QkBywawNkMyzQ3pGZ0QxVj8A1sUB5BX+OXhg6g==

Decrypted Text:

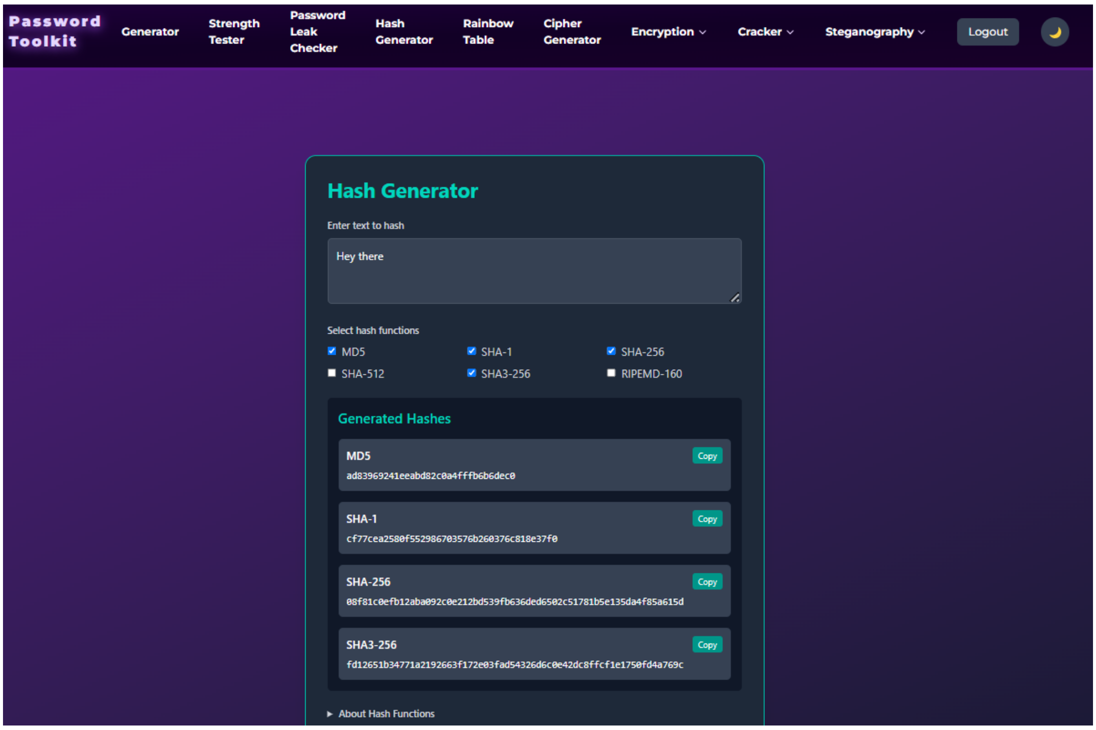
-

Figure 3: Encryption Playground

2.4 Hashing Playground

Table 6: Hashing Playground

Name	Hashing Page
Actor	Authenticated User
Description	Test and visualize MD5, SHA1, and SHA256 hashes for user input.



Hash Generator

Enter text to hash

Hey there

Select hash functions

☑ MD5 ☑ SHA-1 ☑ SHA-256
□ SHA-512 ☑ SHA3-256 □ RIPEMD-160

Generated Hashes

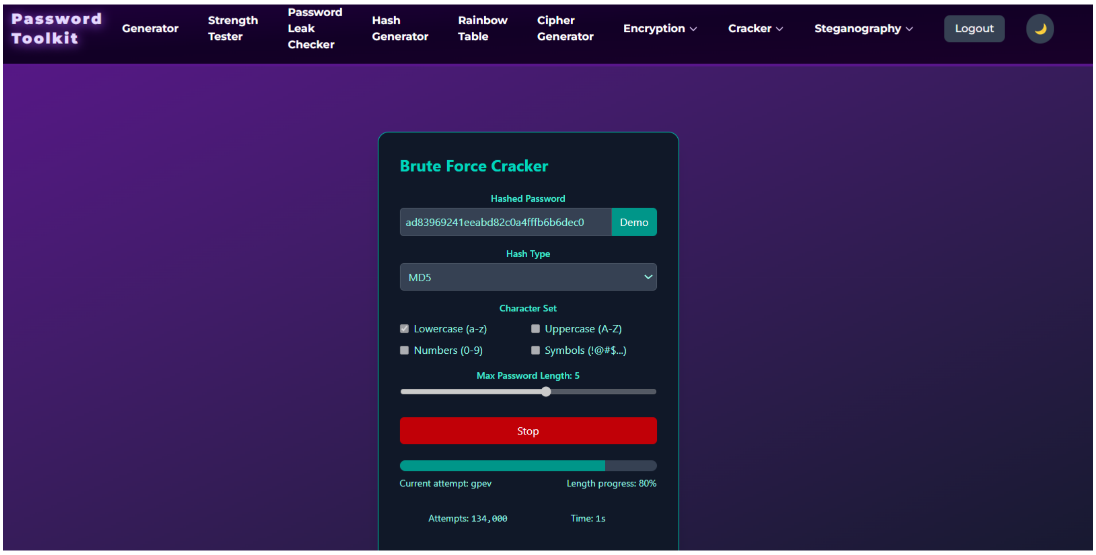
```
MD5
ad83969241eeabd82c0a4fffb6b6dec0

SHA-1
cf77cea2580f552986703576b260376c818e37f0

SHA-256
08f81c0efb12aba092c0e212bd539fb636ded6502c51781b5e135da4f85a615d

SHA3-256
fd12651b34771a2192663f172e03fad54326d6c0e42dc8ffcf1e1750fd4a769c
```

Figure 4: Hashing Playground

2.5 Cracker Demo: Brute Force and Dictionary

Table 7: Cracker Demo

Name	Cracker Demo
Actor	Authenticated User
Description	Simulate password cracking by brute-force or dictionary attack. Visualizes progress and results.



Brute Force Cracker

Hashed Password: ad83969241eeabd82c0a4fffb6b6dec0

Hash Type: MD5

Character Set

☑ Lowercase (a-z) □ Uppercase (A-Z)
□ Numbers (0-9) □ Symbols (!@#$...)

Max Password Length: 5

Current attempt: gpev Length progress: 80%

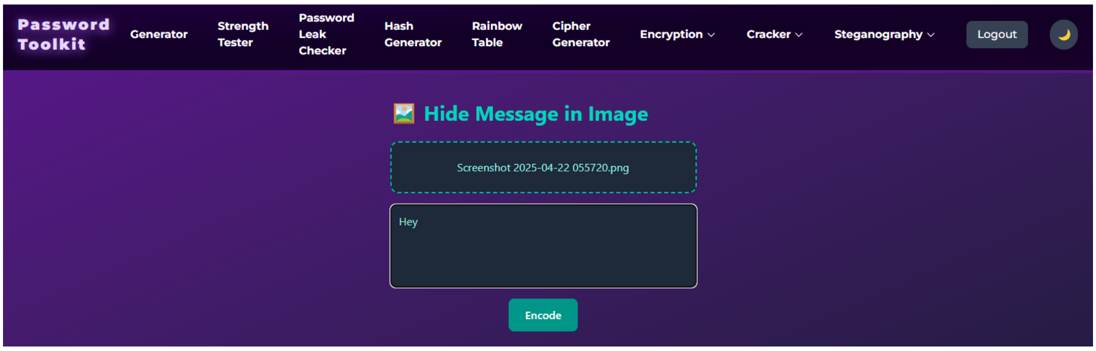
Attempts: 134,000 Time: 1s

Figure 5: Brute Force / Dictionary Cracker Demo

2.6 Steganography Encode/Decode

Table 8: Steganography Encode/Decode

Name	Steganography
Actor	Authenticated User
Description	Hide a message inside an image or extract it (for educational purposes).



Hide Message in Image

Screenshot 2025-04-22 055720.png

Hey

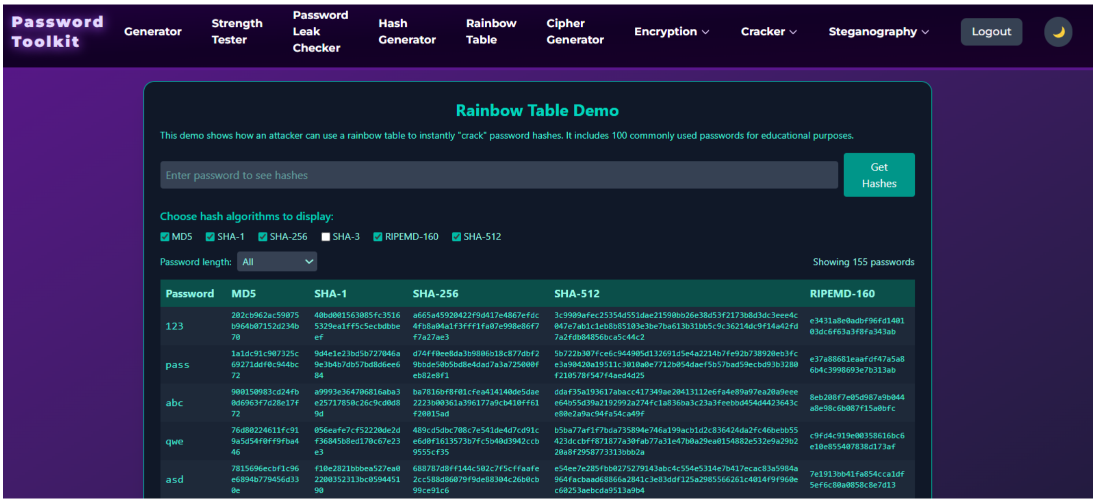
Figure 6: Steganography Encode/Decode

2.7 Rainbow Table Demo

Table 9: Rainbow Table Demo

Name	Rainbow Table Demo
Actor	Authenticated User

Description	Visualizes how precomputed hashes are used to ”crack” passwords, teaching the danger of weak hashing.



Rainbow Table Demo

This demo shows how an attacker can use a rainbow table to instantly "crack" password hashes. It includes 100 commonly used passwords for educational purposes.

Enter password to see hashes

Choose hash algorithms to display:

☑ MD5 ☑ SHA-1 ☑ SHA-256 □ SHA-3 ☑ RIPEMD-160 ☑ SHA-512

Password length: All

Showing 155 passwords

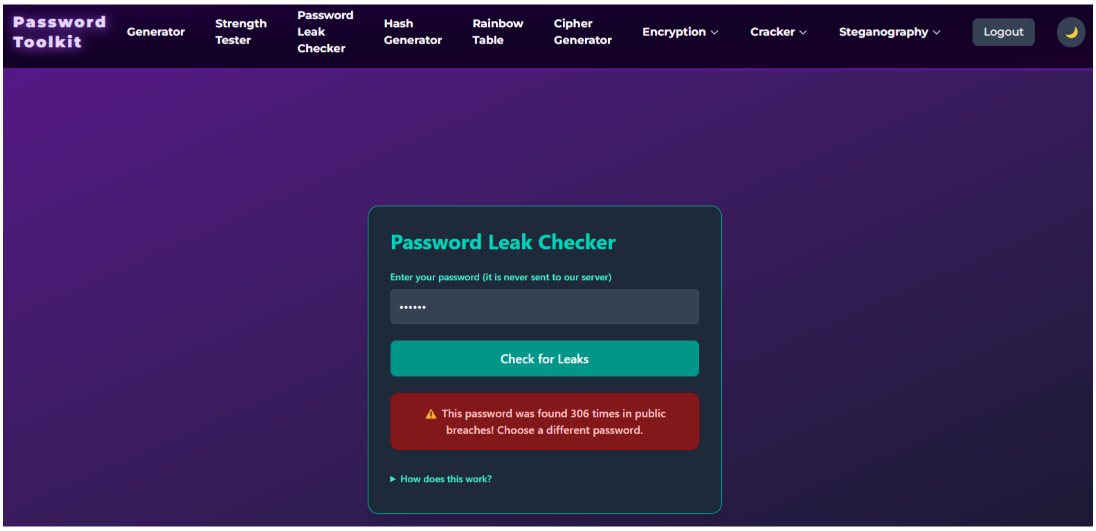
Password	MD5	SHA-1	SHA-256	SHA-512	RIPEMD-160
123	202cb962ac59075b964b07152d234b70	40bd001563085fc35165329ea1ff5c5ecbdbbeef	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	3c9909afec25354d551dae21590bb26e38d53f2173b8d3dc3eee4c047e7ab1c1eb8b85103e3be7ba613b31bb5c9c36214dc9f14a42fd7a2fdb84856bca5c44c2	e3431a8e0adbf96fd140103dc6f63a3f8fa343ab
pass	1a1dc91c907325c69271ddf0c944bc72	9d4e1e23bd5b727046a9e3b4b7db57bd8d6ee684	d74ff0ee8da3b9806b18c877dbf29bbde50b5bd8e4dad7a3a725000feb82e8f1	5b722b307fce6c944905d132691d5e4a2214b7fe92b738920eb3fce3a90420a19511c3010a0e7712b054daef5b57bad59ecbd93b3280f210578f547f4aed4d25	e37a88681eaafdf47a5a86b4c3998693e7b313ab
abc	900150983cd24fb0d6963f7d28e17f72	a9993e364706816aba3e25717850c26c9cd0d89d	ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad	ddaf35a193617abacc417349ae20413112e6fa4e89a97ea20a9eeee64b55d39a2192992a274fc1a836ba3c23a3feebbd454d4423643ce80e2a9ac94fa54ca49f	8eb208f7e05d987a9b044a8e98c6b087f15a0bfc
qwe	76d80224611fc919a5d54f0ff9fba446	056eafe7cf52220de2df36845b8ed170c67e23e3	489cd5dbc708c7e541de4d7cd91ce6d0f1613573b7fc5b40d3942ccb9555cf35	b5ba77af1f7bda735894e746a199acb1d2c836424da2fc46bebb55423dccbff871877a30fab77a31e47b0a29ea0154882e532e9a29b220a8f2958773313bbb2a	c9fd4c919e00358616bc6e10e855407838d173af
asd	7815696ecbf1c96e6894b779456d330e	f10e2821bbbea527ea02200352313bc059445190	688787d8ff144c502c7f5cffaafe2cc588d86079f9de88304c26b0cb99ce91c6	e54ee7e285fbb0275279143abc4c554e5314e7b417ecac83a5984a964facbaad68866a2841c3e83ddf125a2985566261c4014f9f960ec60253aebcda9513a9b4	7e1913bb41fa854cca1df5ef6c80a0858c8e7d13

Figure 7: Rainbow Table Demo

2.8 Password Leak Checker

Table 10: Password Leak Checker

Name	Leak Checker
Actor	Authenticated User
Description	Checks (demo or via API) if a password has appeared in public data breaches.



Password Leak Checker

Enter your password (it is never sent to our server)

••••••

⚠ This password was found 306 times in public breaches! Choose a different password.

Figure 8: Password Leak Checker

3 Implementation Overview

3.1 Directory Structure

- **src/pages**: All major page components (Generator, StrengthTester, Encryption, etc.)
- **src/pages/encryption**: Subpages for AES, RSA, Base64.
- **src/pages/cracking**: BruteForce and Dictionary cracker demos.
- **src/pages/steganography**: Encoder and decoder.
- **src/pages/contexts**: Custom contexts.

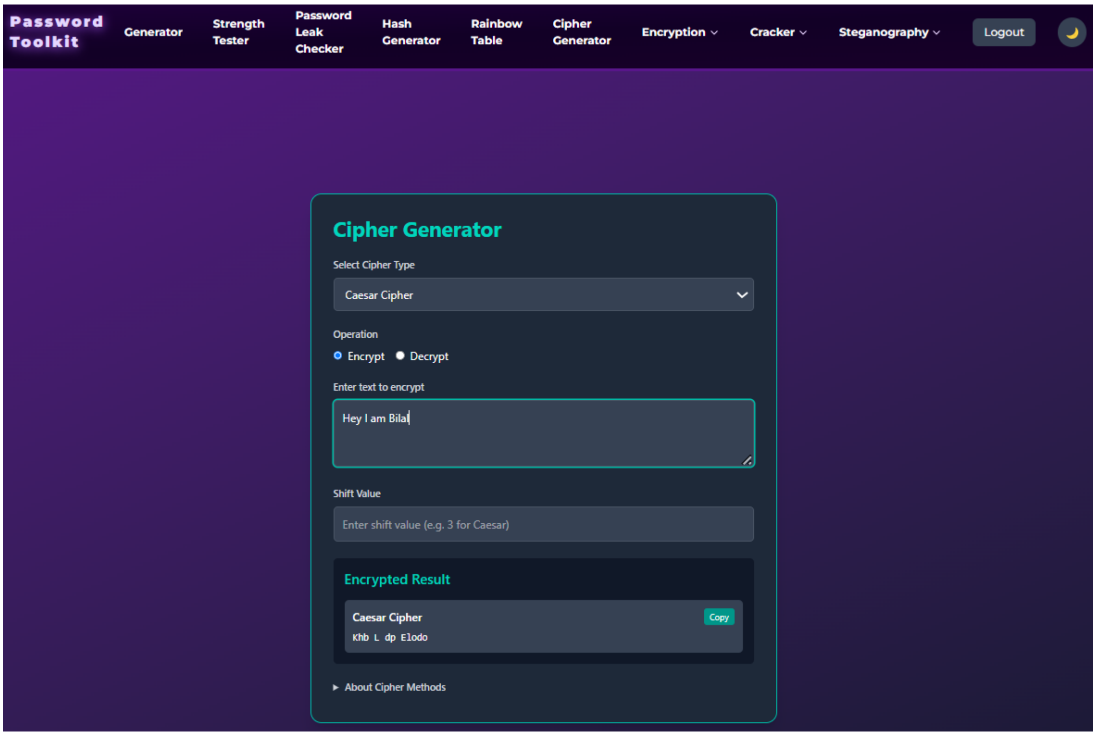
3.2 Core Components/Utilities

- **App.jsx**: Main app file, handles routing and navbar (responsive, theme-aware, authentication-aware).
- **useTheme**: Custom hook for theme toggling and persistence.
- **CryptoJS/WebCrypto**: Used for hashing/encryption implementations.
- **Rainbow Table**: Demo uses a small precomputed hash table, shown in a table.
- **Password Strength**: Custom entropy-based strength metric.

3.3 UI/UX Design

- Responsive navigation bar, dropdowns, and mobile menu.
- Dark/light mode with theme toggle and persistent setting.
- Consistent color palette: teal and purple gradients, matching all themes.
- SVG logo designed for security/toolkit identity.
- All major features accessible from navbar or hamburger on mobile.
- Professional, clean landing page with project and developer details.

4 Sample Screenshots



Cipher Generator

Select Cipher Type: Caesar Cipher

Operation: ◉ Encrypt ○ Decrypt

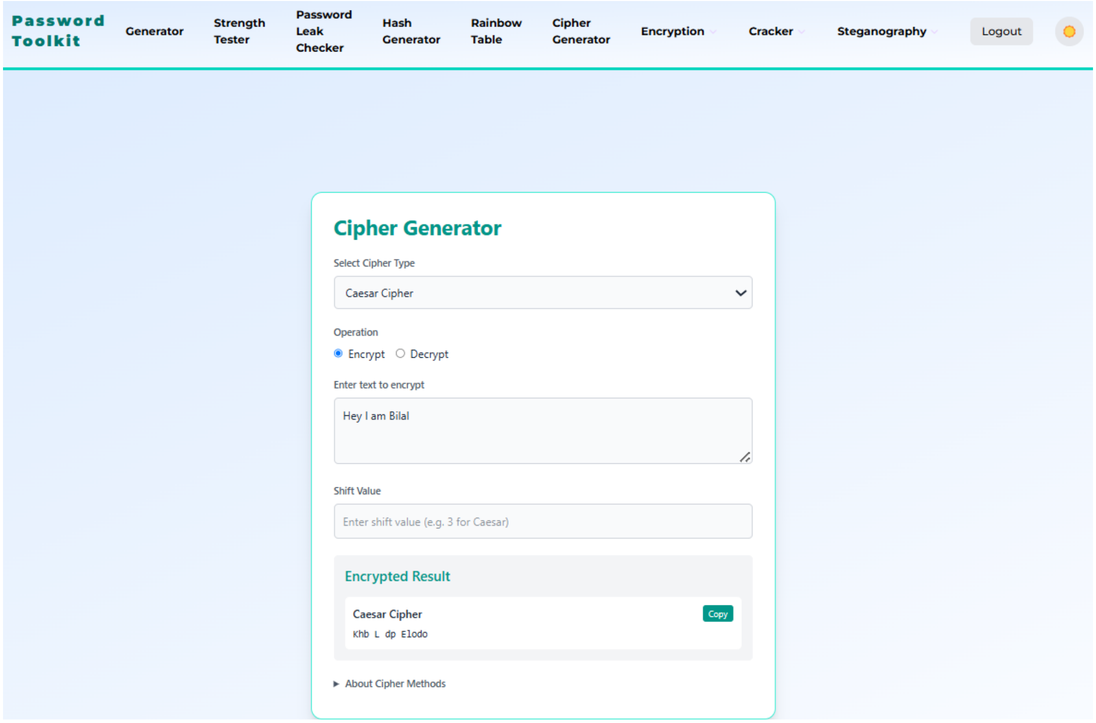
Enter text to encrypt: Hey I am Bilal

Shift Value: Enter shift value (e.g. 3 for Caesar)

Encrypted Result

Caesar Cipher
Khb L dp Elodo

Figure 9: Responsive Navbar (Dark Theme)



Cipher Generator

Select Cipher Type: Caesar Cipher

Operation: ◉ Encrypt ○ Decrypt

Enter text to encrypt: Hey I am Bilal

Shift Value: Enter shift value (e.g. 3 for Caesar)

Encrypted Result

Caesar Cipher
Khb L dp Elodo

Figure 10: Responsive Navbar (Light Theme)

5 Test Cases

5.1 Test Case 1: Password Generation

Description: Generate a password with length 16, including uppercase, lowercase, symbols, and numbers.
Expected Outcome: Password is generated with correct criteria.
Result: Requirement Satisfied.

5.2 Test Case 2: Password Strength Feedback

Description: Enter a weak password (e.g. “password123”).
Expected Outcome: Strength tester identifies it as weak, suggests improvements.
Result: Requirement Satisfied.

5.3 Test Case 3: AES Encryption/Decryption

Description: Encrypt a string using AES and decrypt it.
Expected Outcome: Decrypted output matches original input.
Result: Requirement Satisfied.

5.4 Test Case 4: Rainbow Table Demo Lookup

Description: Enter a password that is in the rainbow table.
Expected Outcome: The tool finds the password instantly via hash lookup.
Result: Requirement Satisfied.

6 Conclusion

6.1 Summary

Password Toolkit demonstrates numerous password security concepts in a hands-on, educational manner. The project includes tools for generating and testing passwords, understanding cryptography, visualizing attacks, and learning about password hygiene. With a modern UI and responsive design, it targets both casual users and technical learners.

6.2 Challenges Faced

- Ensuring secure and correct implementation of cryptographic primitives on the client-side.
- Building a modern, responsive, and accessible user interfaceer interfaceer interfaceer innavigation.r innavigation.withnavigation. navnavigation.and navigation.
- Making complex security concepts approachable for all users.
- Handling browser compatibility for WebCrypto and large hash tables.

6.3 Project's Limitations

- Not suitable for production cryptography (for educational/demo purposes only).
- Limited wordlists/dictionaries in cracker demos for performance reasons.
- Does not store any real user data, so not a password manager.

6.4 Future Enhancements

- Add Diceware passphrase generator.
- More hash/encrypt algorithms (e.g., bcrypt, Argon2, PBKDF2).
- Add visual entropy meters and advanced password pattern analysis.
- Mobile-first progressive web app (PWA) support.

7 References

- **CryptoJS Library**: `https://github.com/brix/crypto-js`
- **WebCrypto API**: `https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API`
- **zxcvbn Password Strength**: `https://github.com/dropbox/zxcvbn`
- **OWASP Passwords**: `https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html`
- **React Documentation**: `https://react.dev`
- **Tailwind CSS**: `https://tailwindcss.com`
- **Have I Been Pwned**: `https://haveibeenpwned.com`
- **Rainbow Tables**: `https://en.wikipedia.org/wiki/Rainbow_table`