Data Breach Alert and Management System Using Spring Boot and Web Technologies, A Secure Web-Based Approach for Monitoring and Alerting Data Breaches

Prof D.V.Biradar[1],Prof K.A.Choure[2], Rutavik Yuvraj Dhumal[3], Pratik Parmeshwar Biradar[4], Omkar Chikhale Santosh[5]

[1,2,3,4,5]Dept of Information Technology, Bidve Engneering College , Latur, Maharashtra, India

***Abstract*—Data breaches have become a critical security concern due to the rapid growth of web-based applications and digital data storage. Organizations face significant risks when sensitive information such as user credentials and personal data are exposed through cyber-attacks or unauthorized access. This paper presents a Data Breach Alert and Management System designed to monitor user data, detect potential breaches, and provide timely alerts to affected users. The proposed system is developed using Java Spring Boot for backend services, HTML, CSS, and JavaScript for the frontend interface, and MySQL for secure data storage. The system checks user credentials against compromised datasets, generates alerts upon detecting suspicious activity, and maintains breach records for analysis and reporting. By integrating secure authentication, database validation, and real-time alert mechanisms, the system enhances data protection and user awareness. The proposed solution aims to reduce the impact of data breaches by enabling early detection and efficient management within web applications.**

***Index Terms*—Data Breach Detection; Alert System; Spring Boot; Web Security; Mysql Database**

I. INTRODUCTION

The increasing use of web-based applications for storing and processing sensitive information has significantly raised concerns about data security. User credentials, personal details, and confidential records are frequently targeted by attackers through techniques such as SQL injection, weak authentication, and unauthorized database access. Data breaches not only compromise user privacy but also lead to financial loss and loss of trust for organizations.

Most existing security solutions focus on preventing attacks but do not provide sufficient mechanisms to detect breaches after they occur. In many cases, users remain unaware that their data has been exposed until serious damage is done. Therefore, there is a strong need for systems that continuously monitor data and notify users when suspicious activities or breaches are detected.

This paper proposes a Data Breach Alert and Management System that detects potential data breaches and alerts users in a timely manner. The system is developed using Java Spring Boot for backend services, HTML, CSS, and JavaScript for the frontend interface, and MySQL for database management. The proposed system aims to enhance data security, improve user awareness, and provide an efficient solution for managing breach-related information in web applications.

II. PROPOSED SYSTEM

The proposed Data Breach Alert and Management System provides an integrated solution for monitoring user data and detecting potential data breach incidents in web applications. The system works by validating user credentials during login and continuously checking stored data for unusual or unauthorized access patterns [8]. When a possible breach is detected, the system triggers alert notifications to inform affected users and system administrators.

The backend of the system is developed using Java Spring Boot, which manages user authentication, business logic, and breach detection processes. The frontend is implemented using HTML, CSS, and JavaScript, providing a responsive and easy-to-use interface for users to register, log in, and view alert notifications. The MySQL database securely stores user details, encrypted credentials, and breach records for future reference and analysis.

By maintaining detailed breach logs and providing real-time alerts, the system helps organizations respond quickly to security incidents. The modular architecture of the proposed system ensures scalability, maintainability, and improved overall security for modern web applications.

III. SYSTEM ARCHITECTURE

The Data Breach Alert and Management System is designed with a modular architecture that ensures scalability, maintainability, and security. The system consists of three main components: Frontend, Backend, and Database, along with the Breach Detection & Alert Module.

It is designed using a layered architecture that separates the user interface, business logic, and data storage layers [6]. This architectural approach improves scalability, security, and maintainability of the system. The system primarily consists of a Frontend Layer, Backend Layer, and Database Layer, integrated with breach detection and alert mechanisms.

A. Frontend Architecture

The frontend layer is responsible for user interaction with the system. It is developed using HTML, CSS, and JavaScript and provides interfaces for user registration, login, and alert visualization.

Frontend responsibilities include:

- User registration and login forms
- Input validation at client side
- Displaying breach alerts and warning messages
- Providing a simple and responsive user interface

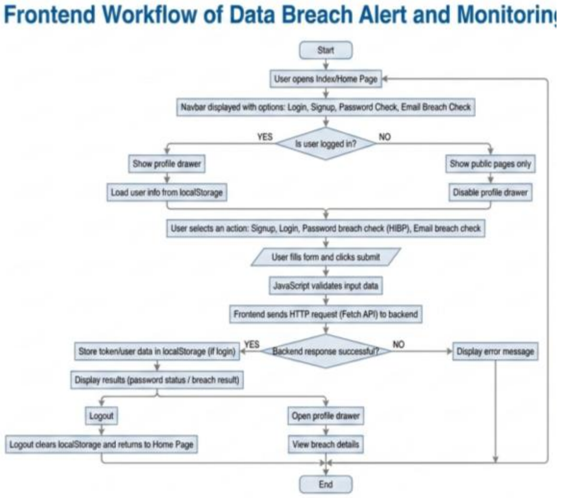

Frontend Workflow of Data Breach Alert and Monitorin
Start
User opens Index/Home Page
Navbar displayed with options: Login, Signup, Password Check, Email Breach Check
Is user logged in?
YES
NO
Show profile drawer
Load user info from localStorage
Show public pages only
Disable profile drawer
User selects an action: Signup, Login, Password breach check (HIBP), Email breach check
User fills form and clicks submit
JavaScript validates input data
Frontend sends HTTP request (Fetch API) to backend
Backend response successful?
YES
NO
Store token/user data in localStorage (if login)
Display error message
Display results (password status / breach result)
Logout
Logout clears localStorage and returns to Home Page
Open profile drawer
View breach details
End

Fig. 1. Frontend Architecture of the Data Breach Alert System

B. Backend Architecture

The backend layer is implemented using Java Spring Boot and acts as the core processing unit of the system. It manages authentication, password verification, breach detection logic, and communication with the database.

Backend responsibilities include:

- User authentication and authorization
- Password strength and breach checking [7]
- Processing user requests through REST APIs
- Alert generation and breach logging
- Secure interaction with the database

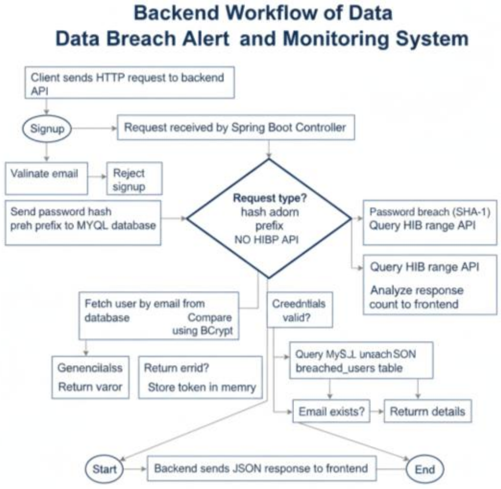

Backend Workflow of Data
Data Breach Alert and Monitoring System
Client sends HTTP request to backend API
Signup
Request received by Spring Boot Controller
Valinate email
Reject signup
Send password hash preh prefix to MYQL database
Request type? hash adorn prefix NO HIBP API
Password breach (SHA-1) Query HIB range API
Query HIB range API
Analyze response count to frontend
Fetch user by email from database
Compare using BCrypt
Creedntials valid?
Genencilalss Return varor
Return errid? Store token in memry
Query MyS⊥ unzachSON breached_users table
Email exists?
Returrn details
Start
Backend sends JSON response to frontend
End

Fig. 2. Backend Architecture Using Spring Boot

C. Database Layer

The database layer uses MySQL to store application data securely. It maintains structured tables for users, passwords, and breach records.

Stored data includes:

- User profile information
- Encrypted passwords
- Breach and alert logs with timestamps

This layer ensures data consistency and supports efficient retrieval for breach detection and reporting.

D. Overall System Flowchart

The overall working of the system follows a sequential flow starting from user interaction to alert generation. The flow begins when a user attempts to register or log in. The system verifies credentials and checks the password against security rules and compromised patterns. If a breach is detected, an alert is generated; otherwise, normal access is granted. All breach-related events are recorded in the database.

Flowchart steps:

- User registration or login
- Credential validation
- Password strength and breach check
- Breach detection decision
- Alert generation or access approval
- Logging of breach details

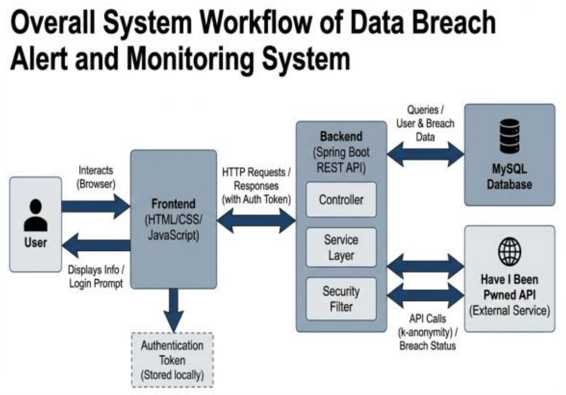

Overall System Workflow of Data Breach Alert and Monitoring System
User
Interacts (Browser)
Displays Info / Login Prompt
Frontend (HTML/CSS/ JavaScript)
Authentication Token (Stored locally)
HTTP Requests / Responses (with Auth Token)
Backend (Spring Boot REST API)
Controller
Service Layer
Security Filter
Queries / User & Breach Data
MySQL Database
Have I Been Pwned API (External Service)
API Calls (k-anonymity) / Breach Status

Fig. 3. Overall System Flowchart of the Proposed System

E. Module Description

The system is divided into the following functional modules:

- User Management Module: Handles user registration and authentication
- Password Check Module: Verifies password strength and compromised status
- Breach Detection Module: Detects potential data exposure
- Alert Management Module: Notifies users of detected breaches
- Database Management Module: Stores and manages all system data

Each module operates independently but communicates through the backend to ensure secure and efficient system operation.

IV. IMPLEMENTATIONS DETAILS

The Data Breach Alert and Management System is implemented as a secure web-based application using a combination of backend, frontend, and database technologies. The implementation focuses on secure authentication, password validation, breach detection, and alert notification.

A. Backend Implementation

The backend of the system is developed using Java Spring Boot, which provides a robust framework for building scalable and secure web applications [5]. RESTful APIs are created to handle user registration, login, password verification, and breach detection processes. Spring Boot manages request handling, session control, and communication between different modules of the system.

Security mechanisms such as password encryption and validation are implemented to protect sensitive user data. The backend also records breach events and alert details in the database for monitoring and analysis.

B. Frontend Implementation

The frontend is developed using HTML, CSS, and JavaScript to provide a responsive and user-friendly interface. Forms are designed for user registration and login, with client-side validation to improve usability and reduce invalid inputs. Alert messages and breach notifications are displayed dynamically to inform users about potential security risks.

C. Database implementation

The system uses MySQL as the backend database for storing user information and breach records. Structured tables are designed to store user details, encrypted passwords, and alert logs. Database queries are optimized to ensure efficient retrieval of information during authentication and breach detection processes.

D. Integration And Data Flow

The frontend communicates with the backend through HTTP requests. The backend processes these requests, performs password and breach checks, and interacts with the MySQL database. Based on the results, appropriate responses and alerts are sent back to the frontend, ensuring smooth data flow across the system.

V. METHDOLOGY

The methodology of the proposed Data Breach Alert and Management System follows a systematic approach to ensure secure handling of user data, effective detection of compromised credentials, and timely alert generation. The system operates through the following sequential steps:

User Registration

- The user provides basic details such as username, email, and password.
- The password is validated for minimum security requirements.
- Valid data is securely stored in the database after encryption.

A. User Authentication

- The user logs into the system using registered credentials.
- The backend verifies the credentials using secure authentication logic.
- Unauthorized access attempts are rejected.

B. Password Strength and Compromise Check

- The entered password is checked for strength and common breach patterns.
- Previously compromised or weak passwords are identified.
- Users are warned and prevented from continuing with unsafe passwords.

C. Breach Detection Process

- User credentials and access behavior are analyzed.
- The system checks for matches with known compromised datasets.
- Suspicious activity is identified as a potential data breach.

D. Alert Generation

- If a breach or compromised password is detected, alerts are generated.
- Notifications are displayed to users through the frontend interface.
- Breach events are logged for monitoring purposes.

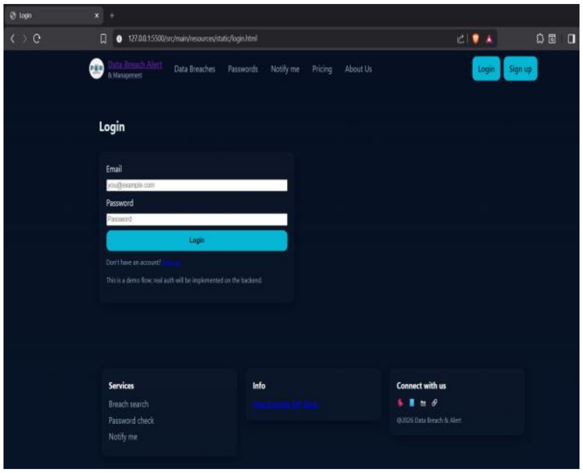
E. Data Logging and Monitoring

- All breach-related activities are recorded with timestamps.
- Stored logs are used for analysis and reporting.
- This ensures traceability and accountability.
-

VI. RESULT AND DISCUSSION

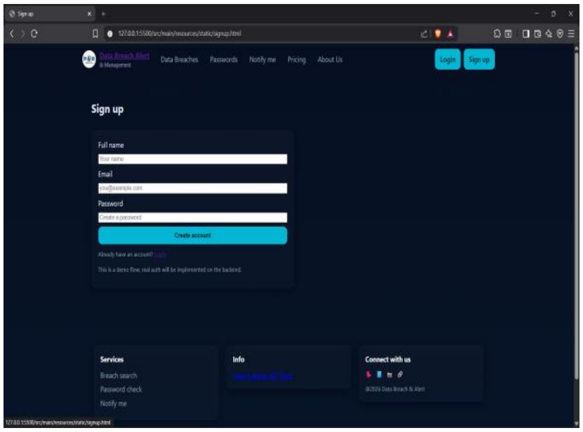
The Data Breach Alert and Management System was implemented and tested using sample user data to evaluate its functionality, accuracy, and usability. The system was tested under different scenarios such as new user registration, valid login, weak password usage, and compromised password detection.

The following are some of the images of the system containg the login , signup , email breach check , password check modules. They are implemented and tested with several sample dataset using different methods of testing. The each and every module of the system is working properly and efficiently.



This is a demo flow; real auth will be implemented on the backend.

Fig 4-: Login Page



Sign up

Full name

Your name

Email

you@example.com

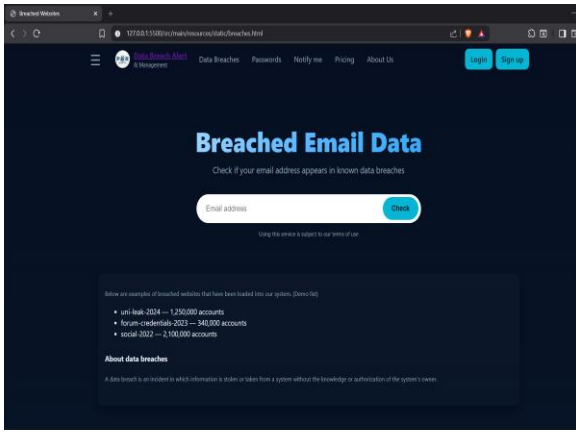
Password

Create a password

Already have an account? Login

This is a demo flow; real auth will be implemented on the backend.

Fig 5-:Signup Page



Breached Email Data

Check if your email address appears in known data breaches

Email address

Using this service is subject to our terms of use

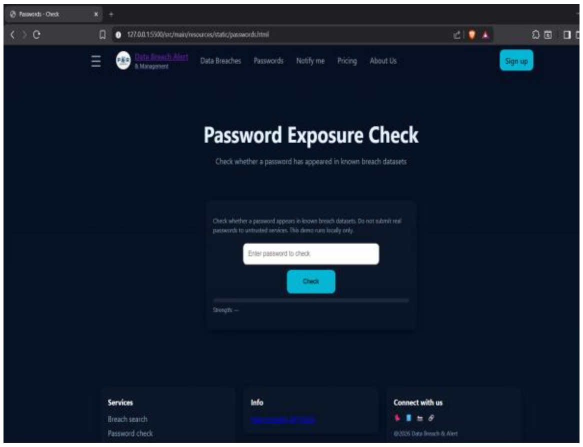
Below are examples of breached websites that have been loaded into our system. (Demo list)

- uni-leak-2024 — 1,250,000 accounts
- forum-credentials-2023 — 340,000 accounts
- social-2022 — 2,100,000 accounts

About data breaches

A data breach is an incident in which information is stolen or taken from a system without the knowledge or authorization of the system's owner.

Fig 6-: Breach Check Page



Password Exposure Check

Check whether a password has appeared in known breach datasets

Check whether a password appears in known breach datasets. Do not submit real passwords to untrusted services. This demo runs locally only.

Enter password to check

Strength: —

Fig 7-:Password Check Page

VII. LIMITATIONS

Although the proposed Data Breach Alert and Management System provides an effective approach for detecting compromised passwords and generating alerts, certain limitations exist. The system currently relies on predefined breach patterns and datasets, which may not cover newly emerging or unknown attack methods. As a result, the effectiveness of the proposed Data Breach Alert and Management System, the system has certain limitations that need to be considered:

A. Limited Breach Dataset:

The system relies on predefined breach patterns and known compromised password datasets. Newly emerging or zero-day breaches may not be detected immediately.

B. Restricted Alert Mechanism:

Currently, alerts are displayed only through the web interface. The absence of email, SMS, or push notifications may delay user response in critical situations.

C. Controlled Testing Environment:

The system has been tested using sample and simulated data. Performance and accuracy may vary when deployed in large-scale or real-world production environments.

D. No Behavioral Analysis:

The system does not analyze advanced user behavior patterns such as abnormal login frequency or geographic anomalies, which could further enhance breach detection.

E. Scalability Constraints:

The current implementation is suitable for small to medium-scale applications

ACKNOWLEDGMENT

The authors would also like to acknowledge the support provided by the institutional infrastructure and laboratory facilities, which contributed to the successful development and testing of the proposed system. The availability of necessary software tools and computing resources played an important role in completing this research work.

VIII. FUTURE SCOPE

The Data Breach Alert and Management System can be extended in multiple directions to improve its security capabilities and practical applicability. One major enhancement includes the integration of real-time breach intelligence services that continuously update compromised credential databases, enabling faster detection of newly exposed data. This would significantly improve the system's responsiveness to emerging threats.

In the future, the system can incorporate multi-factor authentication (MFA) and advanced encryption standards to further secure user accounts and reduce the risk of unauthorized access. Machine learning techniques may be applied to analyze user behavior patterns and detect anomalies, allowing the system to predict potential breaches before they occur.

The alert mechanism can also be expanded to support email, SMS, and mobile notifications, ensuring timely communication with users during critical breach events. Additionally, the system can be adapted for cloud-based and enterprise-level deployment, supporting large-scale user bases and high data volumes. These enhancements would make the proposed system more robust, scalable, and suitable for real-world cybersecurity applications.

IX. CONCLUSION

In this paper, a Data Breach Alert and Management System has been proposed and implemented to address the growing security concerns related to data breaches in web applications. The system focuses on detecting compromised credentials, validating password strength, and generating timely alerts to inform users about potential security risks. By integrating password checking and breach detection mechanisms, the system provides an additional layer of security beyond traditional authentication methods.

The use of Java Spring Boot, HTML, CSS, JavaScript, and MySQL ensures a secure, scalable, and efficient implementation of the proposed solution. Overall, the proposed system enhances user awareness and helps reduce the impact of data breaches by enabling early detection and efficient management of security incidents.

REFERENCES

[1] OWASP Foundation, "OWASP Top Ten Web Application Security Risks," OWASP, 2021. [Online]. Available: https://owasp.org/www-project-top-ten/ OWASP Foundation

[2] OWASP Foundation, "OWASP Data Security Top 10," OWASP,2025.[Online]. Available: https://owasp.org/www-project-data-security-top-10/ OWASP Foundation

[3] Wikipedia, "Credential stuffing," Wikipedia, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Credential_stuffing Wikipedia

[4] Wikipedia, "2023 MOVEit data breach," Wikipedia, 2025. [Online].Available: https://en.wikipedia.org/wiki/2023_MOVEit_data_breach Wikipedia

[5] Wikipedia, "List of data breaches," Wikipedia, 2025. [Online].Available: https://en.wikipedia.org/wiki/List_of_data_breaches Wikipedia

[6] R. Holthouse, S. Owens & S. Bhunia, "The 23andMe Data Breach: Analyzing Credential Stuffing Attacks, Security Vulnerabilities, and Mitigation Strategies," arXiv, 2025. [Online]. Available: https://arxiv.org/abs/2502.04303 arXiv

[7] Time, "Billions of Passwords Have Been Leaked in Massive Breach, Researchers Say," Time, 2025. [Online]. Available: https://time.com/7296254/passwords-leaked-data-breach/ TIME

[8] Check Point Research, "Credential theft has surged 160% in 2025," ITPro.com, 2025. [Online]. Available: https://www.itpro.com/security/cyber-attacks/credential-theft-has-surged-160-percent-in-2025