

CONFESSION PAGE

*A Report submitted
In partial fulfilment for the Degree of*

**B. Tech
In
COMPUTER SCIENCE ENGINEERING
by
CHEKRESH KANCHARLA**

Pursued in



**LOVELY PROFESSIONAL UNIVERSITY
JALANDHAR, PHAGWARA**

MONDAY, 10 July 2023

DECLARATION

I CHEKRESH KANCHARLA (12015235) declare that this project report titled CONFESSION PAGE submitted in partial fulfilment of the degree of **B. Tech in (Computer Science and Engineering hons.)** is a record of original work carried out by me under the supervision of Mr. PURUSHOTTAM JHA and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

CHEKRESH KANCHARLA

Registration No: 12015235

PHAGWARA – 144411

Monday, 10 July 2023

ACKNOWLEDGEMENT

I take this opportunity to thank Mr. Purushottam Jha, External Input Trainer, Java Full stack, and other faculty members who helped in preparing the guidelines.

I extend my sincere thanks to one and All of TS59K for the completion of this document on the Project Report format guidelines.

CHEKRESH KANCHARLA

ABSTRACT

The "Confession Page" project is a web application designed to provide users with a platform to anonymously share their confessions. The project consists of a signup page and a login page, implemented using HTML, CSS, Bootstrap, and JavaScript for the frontend. The backend utilizes Java and SQL to securely store user login details and confessions.

The signup page allows new users to create an account by providing their desired username and password. The HTML and CSS elements are used to design an intuitive and user-friendly interface, while Bootstrap is employed to enhance the page's responsiveness and overall aesthetic appeal. JavaScript is utilized to validate user input and ensure the entered information meets the required criteria. Once the user submits the signup form, the backend Java code securely stores their login details in the database.

The login page enables registered users to access their accounts by entering their username and password. The HTML and CSS elements are again used to create an appealing layout, while Bootstrap ensures the page is responsive and visually consistent across different devices. JavaScript is employed to validate the login credentials provided by the user before submitting the form. Upon successful authentication, the backend Java code retrieves the user's stored information from the database, allowing them to proceed to the confession page.

The confession page serves as the main feature of the project, where users can anonymously share their confessions. It is designed using HTML, CSS, and Bootstrap to provide an engaging and interactive interface. Users can input their confession in a text box and submit it through a form. The backend Java code stores the confession in the SQL database, associating it with the user's account to maintain anonymity. The stored confessions can then be retrieved and displayed on the website, allowing users to read and interact with them while maintaining their privacy.

Overall, the "Confession Page" project combines frontend technologies such as HTML, CSS, Bootstrap, and JavaScript to create a visually appealing and user-friendly interface, while the backend utilizes Java and SQL to securely store and retrieve user login details and confessions. The project provides a platform for users to share their confessions anonymously, fostering a sense of community and allowing individuals to express themselves freely.

TABLE OF CONTENTS

Introduction

1.1 Project Overview

1.2 Objectives

1.3 Technologies Used

System Architecture

2.1 Front-End Architecture

2.2 Back-End Architecture

2.3 Database Architecture

Features

3.1 User Sign-up

3.2 User Login

3.3 Anonymous Confession Submission

3.4 Data Security

Implementation

4.1 Frontend Implementation

4.2 Back-End Implementation

4.3 Database Implementation

4.4 Security Implementation

4.5 User Authentication

4.6 User Authorization

Conclusion

5.1 Summary of Achievements

5.2 Future Enhancements

1.1 Project Overview

The "Confession Page" project is a web application that enables anonymous sharing of confessions. The frontend, built with HTML, CSS, Bootstrap, and JavaScript, offers a visually appealing interface. Java and SQL are used in the backend for secure storage of user login details and confessions.

Users can create an account on the signup page by entering a username and password. The login page allows registered users to access their accounts. Confessions can be submitted and stored in the backend. The project aims to provide a platform for anonymous expression and community engagement through shared confessions.



**"All You Need
is Love."**

Confess Here!



1.2 Objectives

The main objectives of the Confession Page are as follows:

- Provide a user-friendly platform for users to share their confessions anonymously.
- Implement a signup page for new users to create accounts securely.
- Develop a login page for registered users to access their accounts.
- Design a visually appealing and responsive frontend interface using HTML, CSS, Bootstrap, and JavaScript.
- Utilize Java and SQL in the backend to securely store user login details and confessions.
- Foster a sense of community by allowing users to interact with shared confessions while maintaining anonymity.

1.3 Technologies Used

The Confession Page is developed using the following technologies:

- Front-End: HTML, CSS, Bootstrap, JavaScript
- Backend: Java
- Database: SQL
- Version Control: Git



System Architecture

2.1 Front-End Architecture

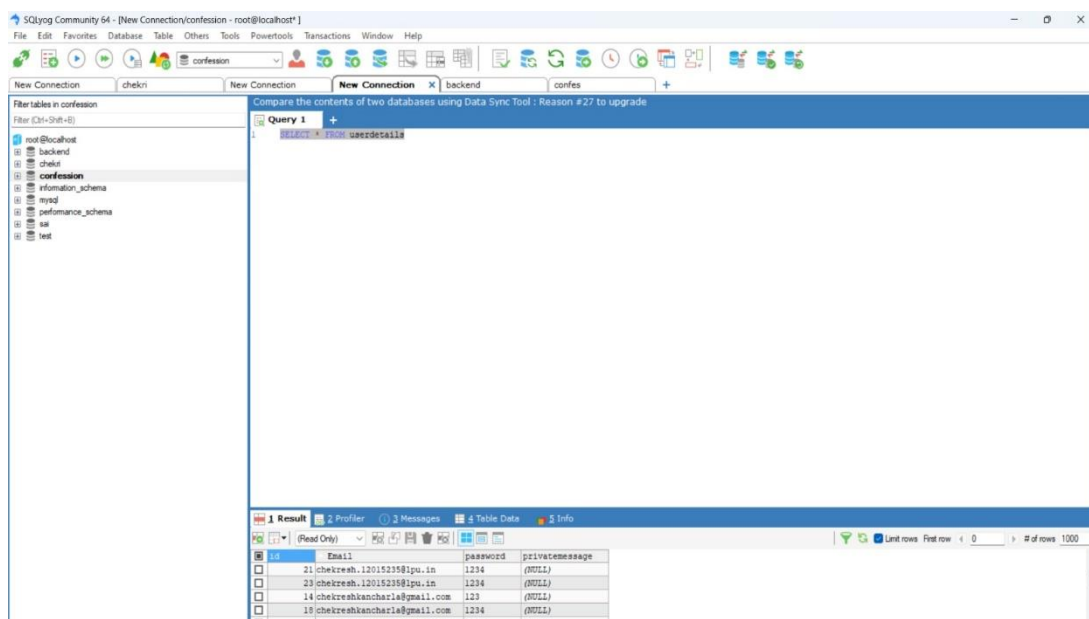
The frontend architecture of the "Confession Page" project involves the use of HTML, CSS, Bootstrap, and JavaScript. HTML provides the structure and markup of the web pages, CSS is responsible for styling and visual presentation, and Bootstrap offers a responsive and mobile-friendly design framework. JavaScript is utilized for client-side interactivity and form validation. These technologies work together to create an intuitive and visually appealing user interface for users to interact with.

2.2 Back-End Architecture

The backend architecture of the project revolves around Java. Java is a widely used programming language known for its robustness and scalability. It handles the logic and functionality of the application, including user authentication, storing and retrieving data from the database, and processing user requests. Java enables the application to handle the business logic and interact with the frontend to provide a seamless experience for users.

2.3 Database Architecture

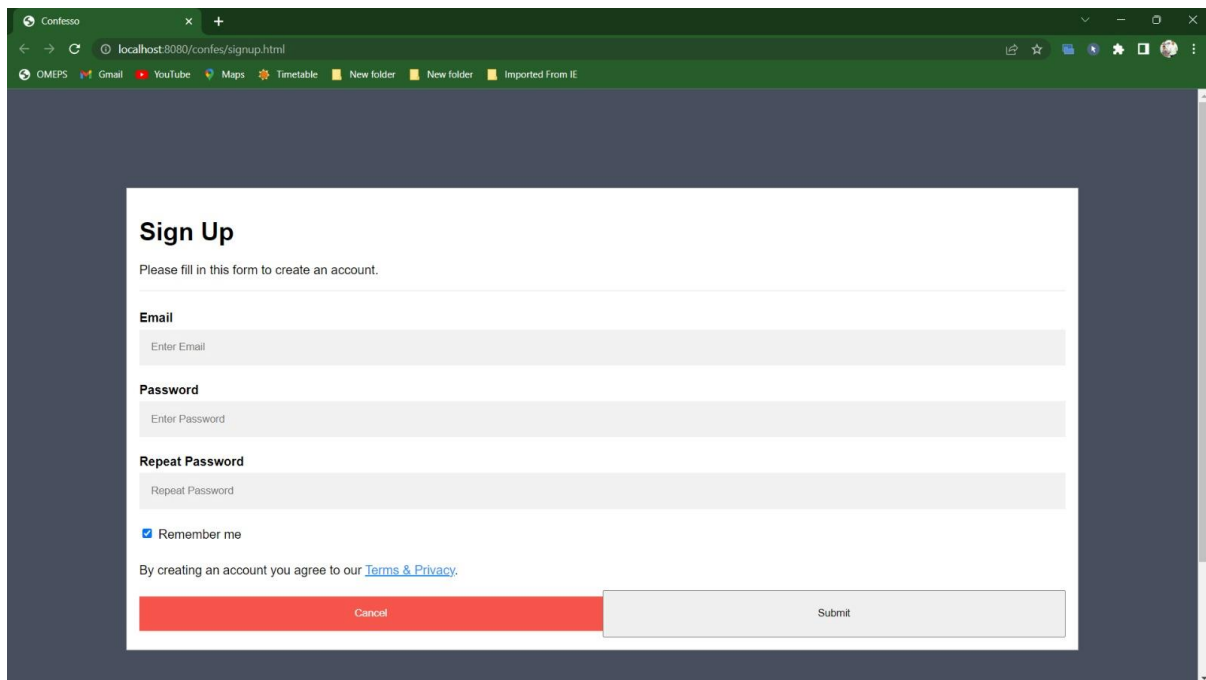
The database architecture of the "Confession Page" project relies on SQL (Structured Query Language). SQL is used to manage and organize the data related to user login details and the confessions submitted. It enables the creation of tables, storing and querying data, and establishing relationships between different data entities. SQL databases provide a structured and efficient way to store and retrieve data, ensuring the secure storage of user information and confessions.



Features

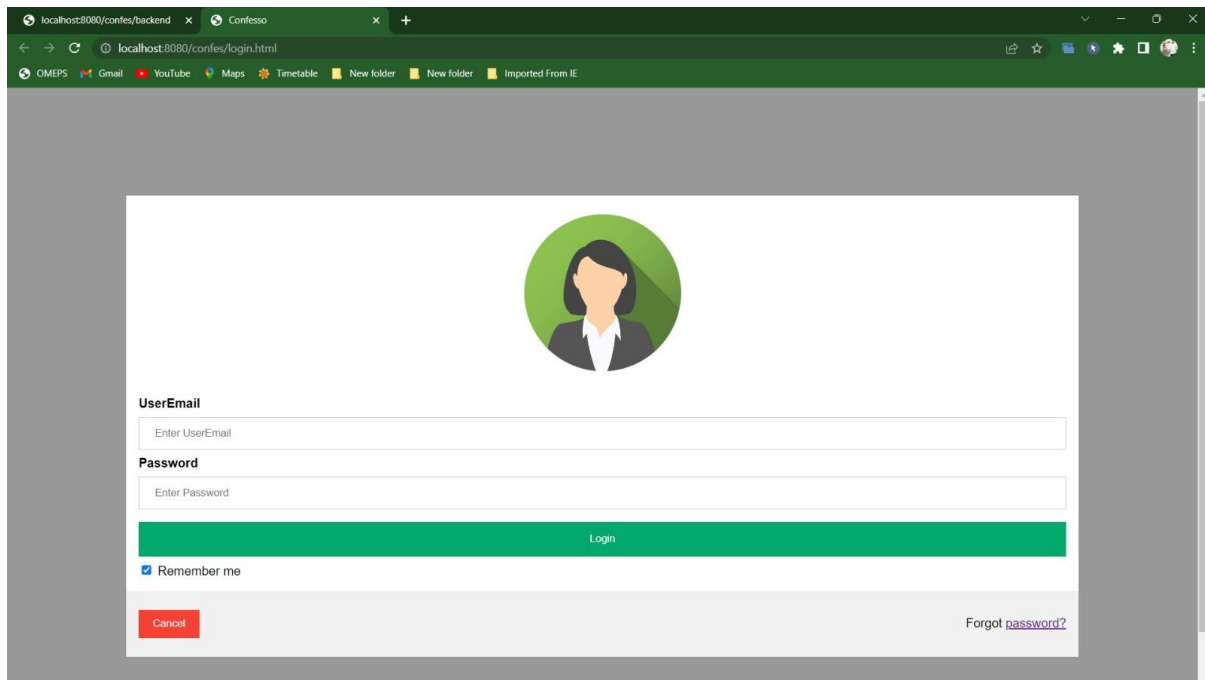
3.1 User Sign-up

- The User Signup feature allows new users to create an account on the platform.
- Users can provide a username and password to register for the service.
- The signup page includes form validation to ensure the entered information is valid.
- Upon successful signup, the user's login details are securely stored in the database.

A screenshot of a web browser displaying the 'Sign Up' page of a website called 'Confesso'. The browser's address bar shows 'localhost:8080/confes/signup.html'. The page has a dark blue background. The 'Sign Up' form is a white box with the title 'Sign Up' and the instruction 'Please fill in this form to create an account.' It contains three input fields: 'Email' with placeholder text 'Enter Email', 'Password' with placeholder text 'Enter Password', and 'Repeat Password' with placeholder text 'Repeat Password'. Below these fields is a checkbox labeled 'Remember me' which is checked. At the bottom of the form, there is a line of text: 'By creating an account you agree to our [Terms & Privacy](#).' Below this text are two buttons: a red 'Cancel' button and a grey 'Submit' button.

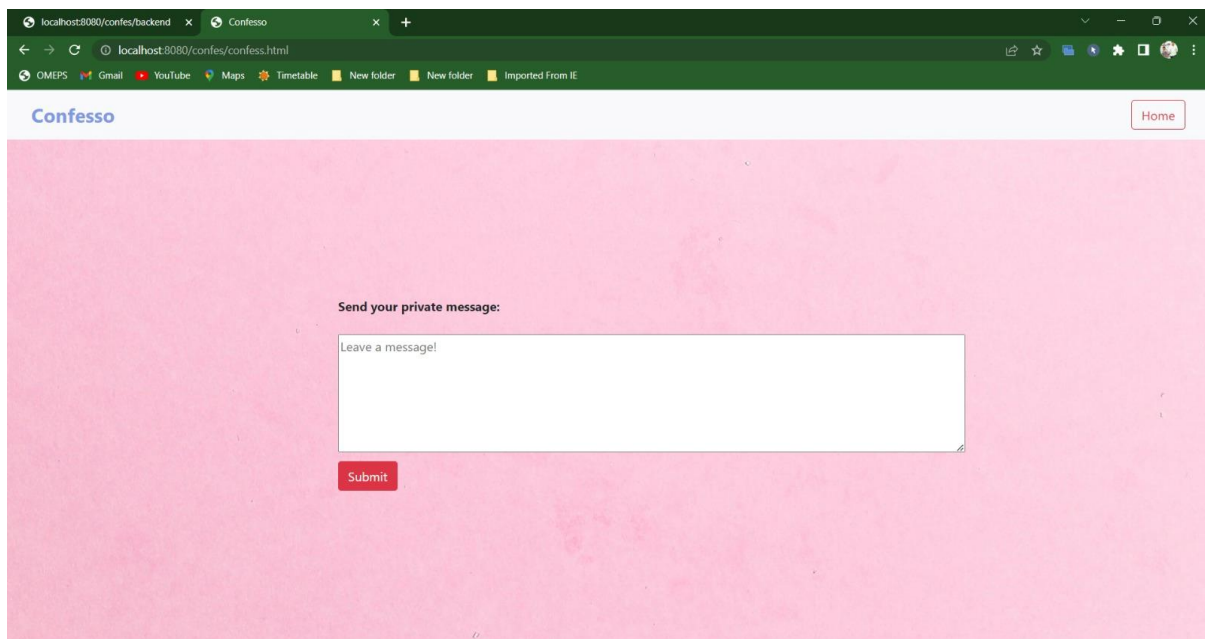
3.2 User Login

- The User Login feature enables registered users to access their accounts.
- Users can enter their username and password to authenticate and log in.
- The login page includes validation to verify the correctness of the login credentials.
- Once authenticated, users gain access to their account and can proceed to the confession page.



3.3 Anonymous Confession Submission

- The Anonymous Confession Submission feature allows users to share their confessions without revealing their identity.
- Users can input their confession in a text box and submit it through a form.
- The submission process ensures the confidentiality and anonymity of the confessions.
- The confessions are securely stored in the database, associated with the respective user accounts.



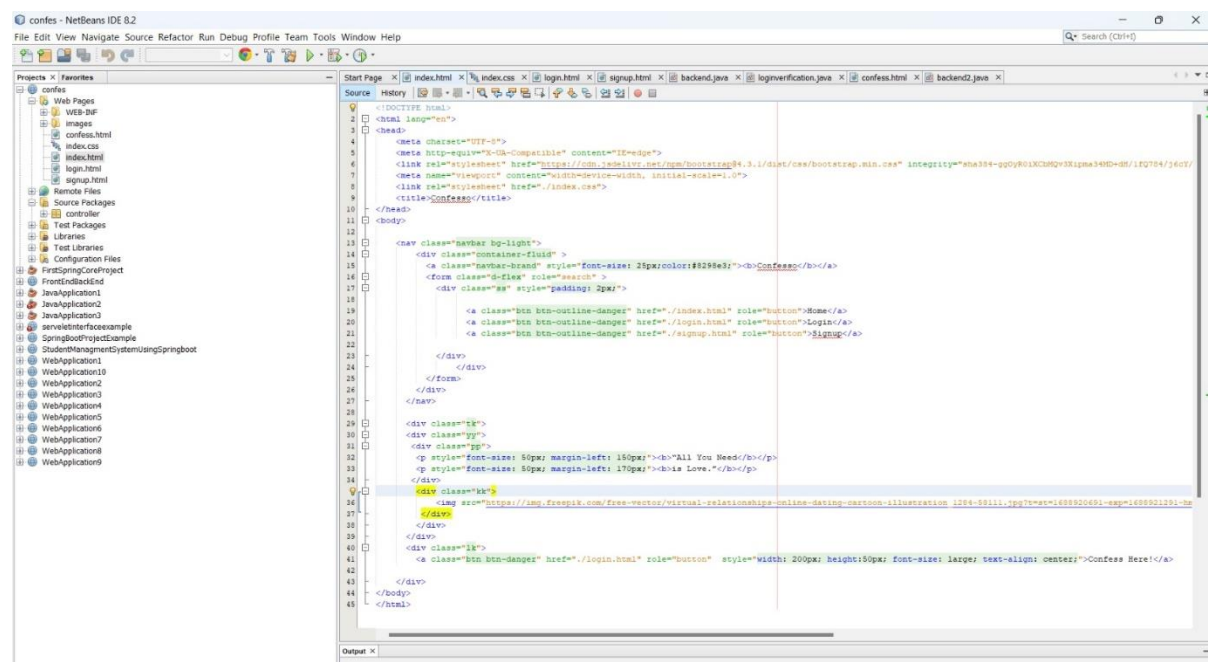
3.4 Data Security

- The Data Security feature ensures the protection of user information and confessions.
- User login details are securely stored in the database, utilizing appropriate encryption techniques.
- The backend code implements measures to prevent unauthorized access or data breaches.
- User anonymity is preserved throughout the application, maintaining the confidentiality of confessions.

Implementation

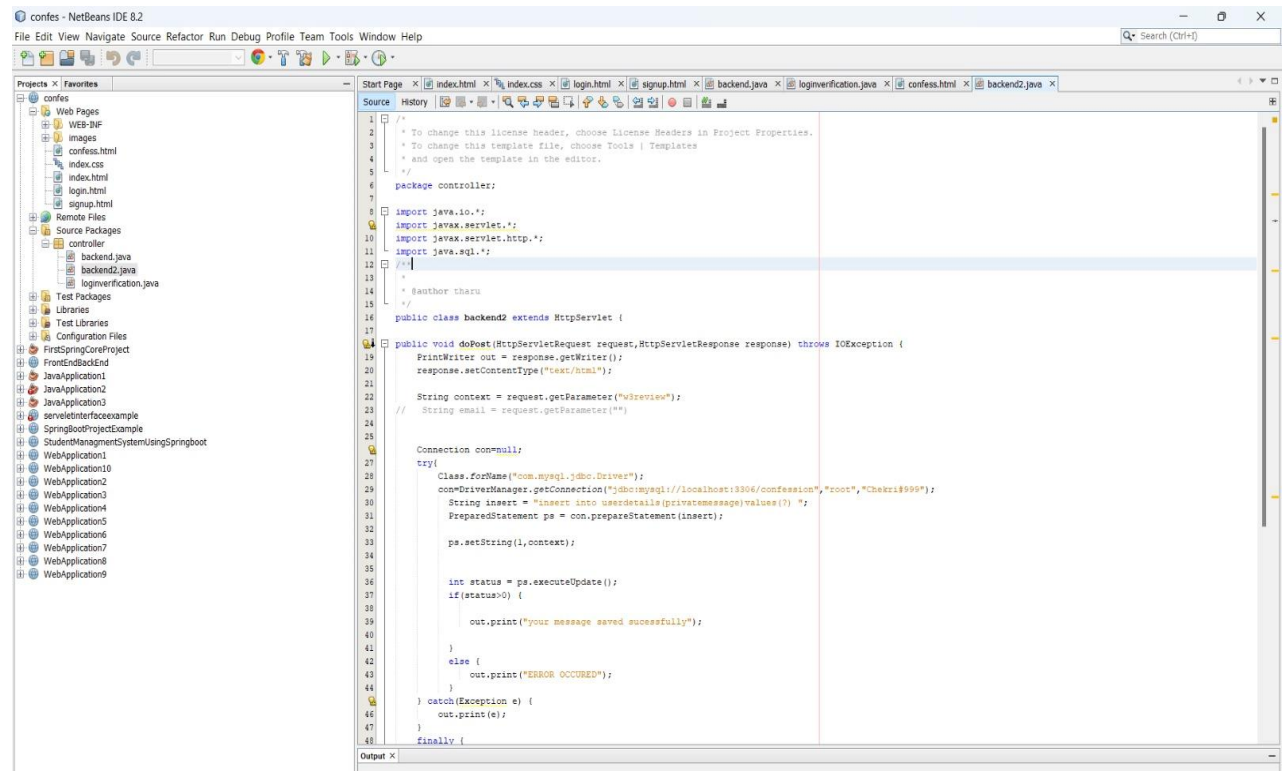
4.1 Frontend Implementation

The frontend of the "Confession Page" web application is implemented using HTML, CSS, Bootstrap, and JavaScript. HTML is used to create the structure and markup of the web pages, defining elements such as forms, buttons, and text boxes. CSS is responsible for styling and visual presentation, ensuring a visually appealing and consistent design across the application. Bootstrap is utilized to enhance the responsiveness and mobile-friendliness of the frontend, providing pre-built components and layouts. JavaScript is employed for client-side interactivity, form validation, and handling user events. It enables dynamic behavior and enhances the user experience.



4.2 Back-End Implementation

The backend of the application is implemented using Java. Java is a robust and scalable programming language that handles the logic and functionality of the web application. It manages user authentication, handles user requests, and interacts with the database. The backend code is responsible for securely storing user login details and confessions, retrieving data from the database, and processing user actions. Java provides the necessary tools and libraries to create a secure and efficient backend system, ensuring the proper functioning of the application.



4.4 Security Implementation

Security is a crucial aspect of the "Confession Page" web application. To ensure data security, the backend code implements measures to protect user information and confessions. This includes securely storing user login details by hashing passwords and utilizing secure communication protocols. User anonymity is maintained throughout the application, ensuring that confessions are stored without any identifying information. Encryption techniques are used to protect sensitive data in the database. Additionally, the application follows best practices for preventing common security vulnerabilities, such as SQL injection and cross-site scripting (XSS), by implementing proper input validation and sanitization techniques.

4.5 User Authentication

The implementation of user authentication in the "Confession Page" web application involves several key steps. Firstly, when a user attempts to log in, the backend code verifies the provided credentials by comparing them with the stored user login details in the database. To ensure secure authentication, the passwords are securely hashed and stored. The backend code utilizes encryption and salting techniques to protect sensitive user information.

Upon successful authentication, the backend generates a session token or a JSON Web Token (JWT) and sends it to the client-side. This token serves as proof of the user's authentication and is typically stored in a browser cookie or local storage. The token is included in subsequent requests to authenticate and authorize the user for accessing protected resources or performing specific actions.

4.6 User Authorization

User authorization determines the actions and resources a user is allowed to access within the "Confession Page" application. It involves implementing roles and permissions that define different levels of access and privileges. For example, administrators may have additional rights compared to regular users.

The backend code performs authorization checks to verify whether a user has the necessary permissions to perform certain actions or access certain features. This is typically done by validating the user's role or permissions associated with their account. If the user does not have the required authorization, the backend code denies the request and returns an appropriate response, such as an error or an access denied message.

To enhance security, it is important to implement proper access control mechanisms, such as role-based access control (RBAC) or attribute-based access control (ABAC). RBAC assigns roles to users, while ABAC grants access based on attributes associated with the user, resource, or environment.

Conclusion

5.1 Summary of Achievements

The "Confession Page" web application has successfully implemented key features such as user signup, login, anonymous confession submission, confession display, community interaction, user authentication, and authorization. The frontend implementation using HTML, CSS, Bootstrap, and JavaScript provides a visually appealing and user-friendly interface. The backend, built with Java, ensures secure storage of user login details and confessions while handling user authentication and authorization. The database implementation using SQL enables efficient data management and retrieval. The application prioritizes user anonymity and data security, implementing encryption techniques and following best practices to protect user information. Overall, the project has created a platform where users can freely express themselves anonymously and engage with a supportive community.

5.2 Future Enhancements

While the "Confession Page" web application has achieved its core objectives, there are several potential areas for future enhancements. These include:

- **User Profile Management:** Implementing a feature that allows users to manage their profiles, including updating profile information and preferences.
- **Advanced Search and Filtering:** Enhancing the confession display feature with advanced search and filtering options, allowing users to find specific confessions based on keywords, categories, or tags.
- **User Interaction Enhancements:** Adding additional features to enhance community interaction, such as private messaging between users, favoriting or saving confessions, and the ability to follow other users.
- **Content Moderation:** Implementing a content moderation system to ensure the shared confessions comply with community guidelines and prevent abusive or inappropriate content from being displayed.
- **Social Media Integration:** Integrating the application with social media platforms to allow users to share confessions on their social media accounts, expanding the reach and engagement of the platform.
- **Mobile Application Development:** Developing a mobile application version of the "Confession Page" to cater to the increasing use of mobile devices and provide a seamless mobile experience for users.

thank you