

Web Application Firewall

CS-342 Information Security



Project Supervisor

Prof. Dr. Faiza Iqbal

Project Members

Kabir Ahmad	2021-CS-4
Mahnoor Fatima	2021-CS-6

Computer Science Department
University of Engineering and Technology, Lahore

Table of Contents

1. Introduction.....	4
Problem Statement	4
Abstract.....	4
Description.....	4
2. Feature & Requirements	5
For Tailor	5
For Customer	5
Application Management	6
Security Features.....	7
Application Services	8
3. Problem Identification and Solutions.....	9
Cyber Security Problems and vulnerabilities that relates with our application:.....	9
Solutions	10
Problems during working.....	11
Tools and Techniques	12
Tools	12
Techniques:	13
Application Structure	14
Database Structure	14
Interview Insights.....	15
Senior Software Engineer @Educative.....	15
Approach.....	15
Mail.....	15
Main questions with brief Answers	16
Review	16
What we Learn from Interview	17
Google Form Circulation	18
Visualization & Case Study.....	19
Wire-Frames:	19
Project poster:	23
Security Best Practice & Future Directions	24
Security Best Practices Implemented:.....	24

Future Directions for Security Enhancement.....	24
1. Continuous Monitoring: Implement continuous security monitoring mechanisms to detect and respond to security incidents or suspicious activities in real-time.....	24
2. Google OAuth Integration: Implement Google OAuth for user authentication, allowing users to sign in using their Google accounts. This enhances security by leveraging Google's robust authentication infrastructure and reducing the risk of password-related security issues.....	24
3. Enhanced Authentication: Explore the implementation of advanced authentication methods such as biometric authentication or multi-factor authentication (MFA) to further strengthen user authentication and access controls.....	24
4. Incident Response Plan: Develop and regularly update an incident response plan outlining the steps to be taken in the event of a security breach or incident, including communication protocols and recovery procedures.....	24
5. External Security Testing: Engage third-party security experts to conduct external security testing, including penetration testing and vulnerability assessments, to identify potential security gaps and validate the effectiveness of existing security measures.....	24
Conclusion	25

1. Introduction

In contemporary web development, the prevalence of cyber threats poses significant challenges to the security and integrity of web applications. Among the most notorious vulnerabilities are Cross-Site Scripting (XSS) and SQL Injection attacks, which exploit weaknesses in web application defenses to manipulate data, compromise user sessions, and potentially gain unauthorized access to sensitive information. Addressing these vulnerabilities is paramount to ensure the trustworthiness and reliability of web-based systems. This project endeavors to develop a comprehensive web firewall solution tailored to mitigate XSS and SQL Injection risks in a web application environment.

Problem Statement

Web applications are inherently vulnerable to XSS and SQL Injection attacks, which can lead to severe consequences such as data breaches, session manipulation, and system compromise if proper security measures are not implemented. To mitigate these risks, it's imperative to develop an application that is resilient to such insecurities. This entails employing robust security mechanisms like input validation, parameterized queries, and secure authentication to ensure the application remains free from these vulnerabilities, safeguarding user data and system integrity.

Abstract

The target web application caters to two distinct user roles: tailors and customers. Tailors assume administrative privileges, encompassing the responsibilities of overseeing client orders, managing product types and feature services, as well as approving and managing user accounts. Conversely, customers are end-users who register within the system, subject to approval by tailors before gaining access to the application's features. Upon approval, customers can engage in various activities such as placing orders, providing feedback, reviewing order history, and managing their profiles.

Description

The core objective of this project is to implement a sophisticated web firewall system to mitigate the inherent risks associated with XSS and SQL Injection attacks. The proposed solution comprises several key components meticulously designed to bolster the security posture of the web application:

- Password Encryption: Leveraging advanced encryption techniques to safeguard user passwords stored in the database, thereby thwarting unauthorized access attempts.
- Input Sanitization using Entity Framework MVC: Employing stringent input validation and sanitization mechanisms within the Model-View-Controller (MVC) architecture to meticulously sanitize user inputs, mitigating the risk of XSS and SQL Injection attacks.
- User Approval Workflow: Instituting a rigorous user approval workflow, wherein tailors possess the authority to review and approve customer registrations, ensuring that only legitimate users gain access to the system.
- Continuous Security Monitoring and Updates: Instituting a proactive approach to security maintenance by continually monitoring emerging threats and vulnerabilities, and promptly updating the web firewall system to incorporate requisite patches and enhancements.

By integrating these fundamental components, the web firewall system aims to establish a robust defense mechanism, fortifying the web application against XSS and SQL Injection vulnerabilities. This project underscores the paramount importance of adopting a proactive stance towards web application security, emphasizing the imperative of safeguarding sensitive data, preserving user privacy, and upholding the integrity of the application's functionalities. Through the meticulous implementation of the web firewall system, the project endeavors to mitigate XSS and SQL Injection risks, thereby bolstering the overall security posture of the web application and enhancing its resilience against evolving cyber threats.

2. Feature & Requirements

For Tailor

- **Administrative Privileges:**
 - View client orders, feedbacks, manage product types, feature services, and user registrations.
 - Approve and manage customer accounts.
 - Administer tailor profile settings.
- **Registration Approval Workflow:**
 - Review and approve customer registrations before granting login access.
 - Maintain control over user authentication and access privileges.
- **Service Management:**
 - Manage product types and feature services to cater to customer needs.
 - Customize service offerings based on market trends and customer preferences.
- **Order Management:**
 - Track and manage client orders efficiently.
 - Ensure timely fulfillment and delivery of orders.
- **Profile Management:**
 - Update personal information, preferences, and profile settings.
 - Maintain tailor-specific details and preferences securely.

For Customer

- **Registration and Approval Process:**
 - Register within the system and await tailor approval for login access.
 - Provide necessary information for registration and account approval.
- **Order Placement and Tracking:**
 - Place orders for products or services offered by the tailor.
 - Track order status and receive updates on order progress.
- **Feedback Provision:**
 - Provide feedback on products, services, and overall customer experience.
 - Contribute to service improvement and refinement based on user input.
- **Profile Management:**
 - Manage personal information, preferences, and profile settings securely.

Application Management

Application's data flow diagram explains its Management that what is the flow of the application and what features it provides to its end users are all well explained in the diagram below.

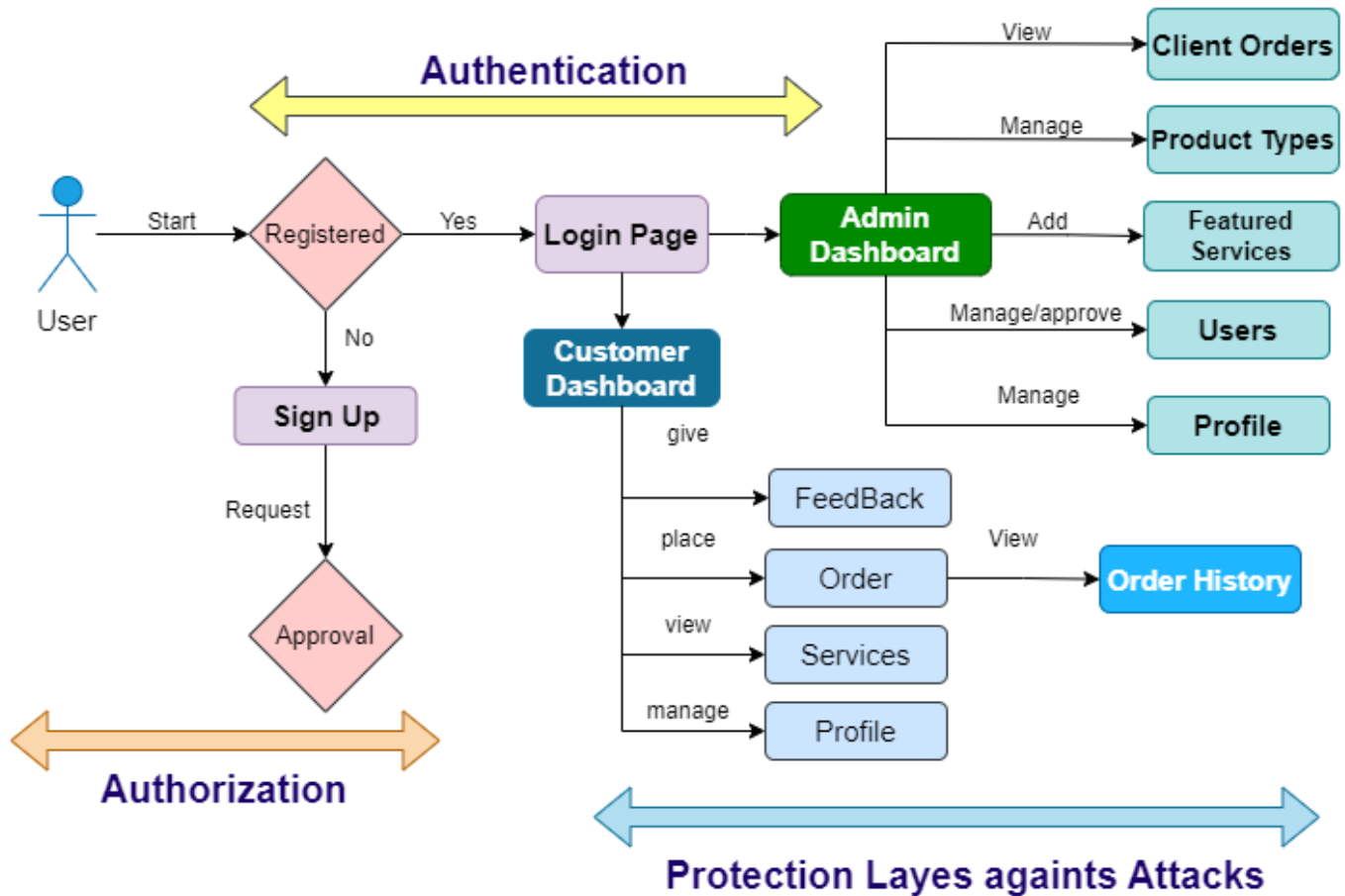


Figure 1: Data Flow Diagram (DFD) of Application

Security Features

1. **Web Firewall Implementation:**

- Deploy a robust web firewall to protect against cross site scripting (xss) and SQL Injection attacks.
- Ensure the integrity and security of user data and application functionalities.

2. **Password Encryption and Secure Storage:**

- Encrypt passwords using advanced encryption algorithms before storing them in the database.
- Prevent unauthorized access to user credentials and sensitive information.

3. **Input Sanitization and Validation:**

- Utilize Entity Framework MVC for thorough input sanitization and validation.
- Convert user inputs into strings and validate against predefined rules to mitigate risks of malicious code injections.

4. **Account Lockout Policy:**

- Implement an account lockout policy to temporarily suspend user access after repeated failed login attempts.
- Prevent brute force attacks and unauthorized access to user accounts.

5. **Continuous Security Monitoring and Updates:**

- Monitor security threats and vulnerabilities continuously.
- Apply regular security updates and patches to maintain a proactive security posture and mitigate emerging risks.

Application Services

This Application Majorly provide almost all mandatory services as a web application firewall approach. Services are described in figure shown below:

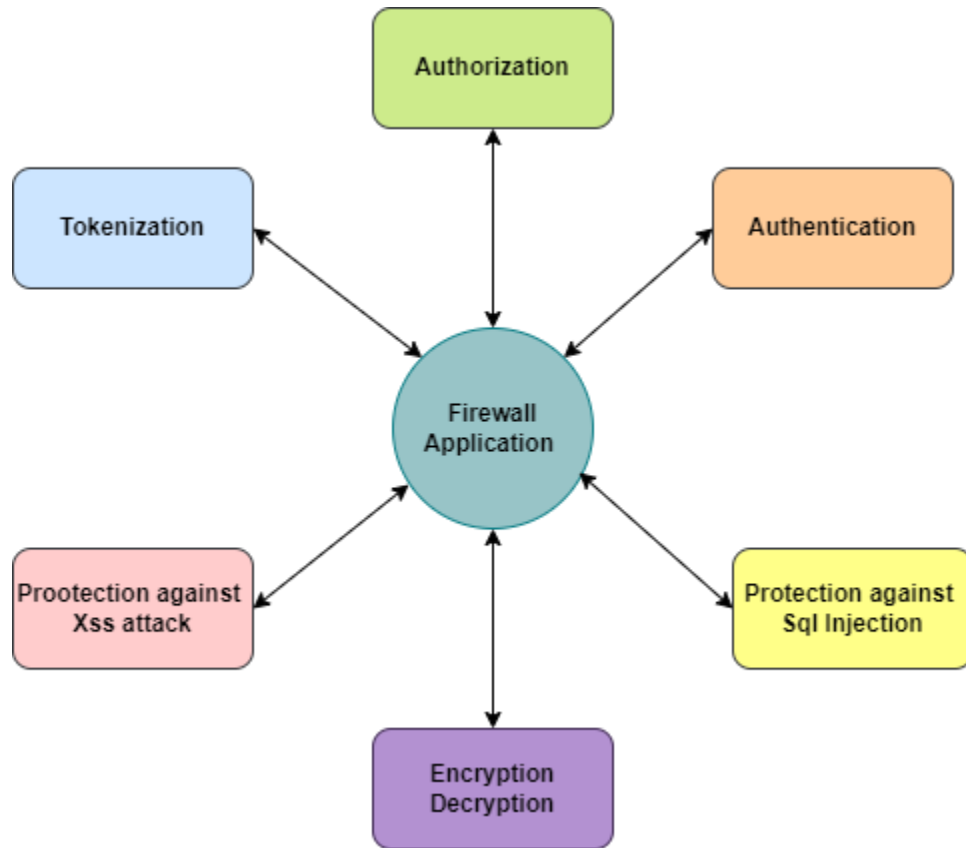


Figure 2: shows services that the Application provides

3. Problem Identification and Solutions

Some identified Problems and their solution are:

Cyber Security Problems and vulnerabilities that relates with our application:

1. Cross-Site Scripting (XSS) Vulnerabilities:

- Since our web application involves user inputs and interactions, there's a risk of XSS vulnerabilities if proper input validation and output encoding measures are not implemented. Attackers could inject malicious scripts into web pages viewed by other users, leading to session hijacking, data theft, or unauthorized actions.

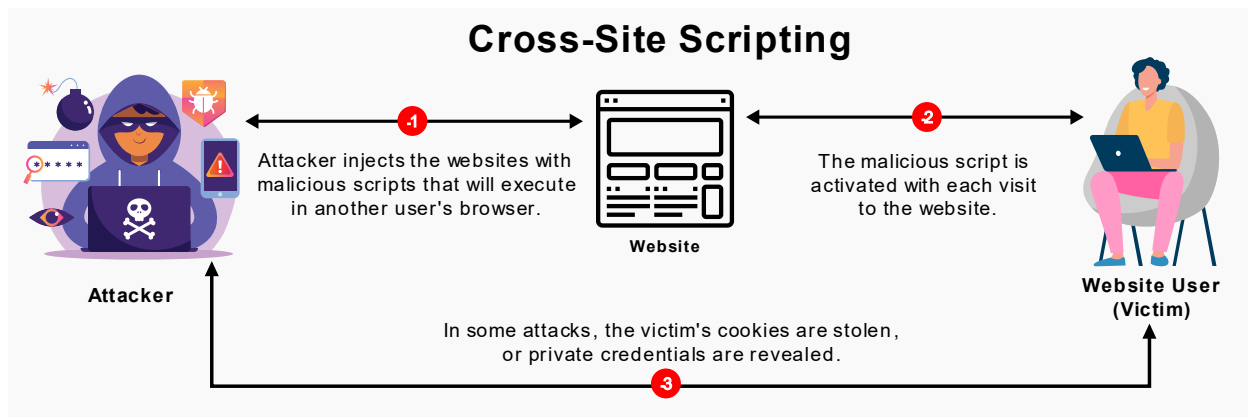


Figure 3: Cross Site Scripting Explanation

2. SQL Injection Attacks:

- Without adequate input validation and parameterized queries, our application is susceptible to SQL injection attacks. Attackers could manipulate SQL queries to access, modify, or delete sensitive data in the database, potentially leading to data breaches or loss of confidentiality.

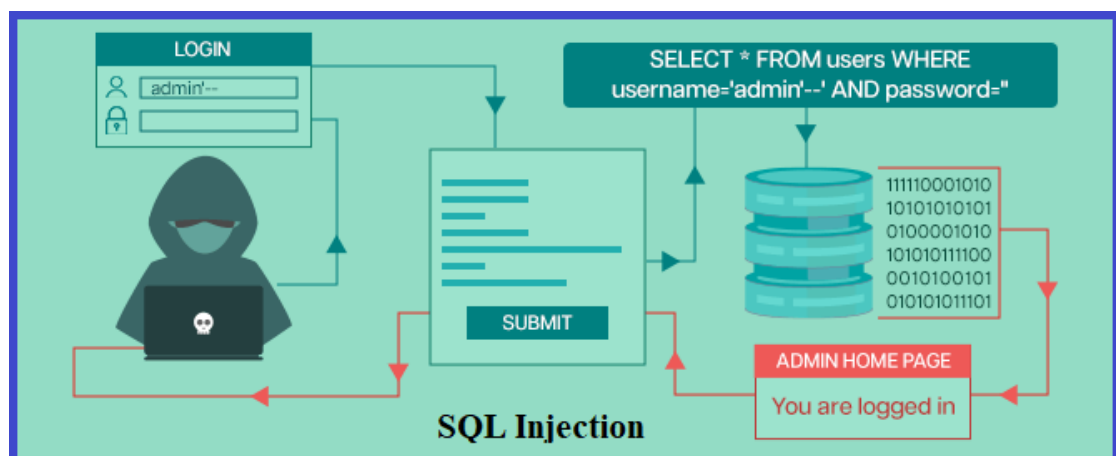


Figure 4: SQL Injection Problem Explanation

3. **Insecure Authentication and Authorization:**

- If authentication mechanisms are not properly implemented or if authorization controls are weak, attackers may exploit these vulnerabilities to gain unauthorized access to sensitive functionalities or user accounts. This could result in data manipulation, unauthorized transactions, or exposure of confidential information.

4. **Inadequate Access Controls:**

- If access controls are not granular or if privilege escalation vulnerabilities exist, attackers may exploit these weaknesses to gain unauthorized access to sensitive features or administrative functionalities. This could result in unauthorized data access, manipulation, or disruption of service.

5. **Encryption and Data Protection:**

- While encrypting passwords is a good practice, ensure that all sensitive data is adequately protected. Failure to do so could lead to data breaches, regulatory non-compliance, and loss of customer trust.

6. **Secure Configuration and Hardening:**

- Improperly configured servers, databases, or web application frameworks may introduce additional security risks. Follow security best practices for server and application hardening, disable unnecessary services, and apply security configurations based on industry standards.

Solutions

1. **Cross-Site Scripting (XSS) Vulnerabilities:**

- Implemented rigorous input validation and output encoding techniques to sanitize user inputs and prevent XSS attacks.
- Utilized security libraries or frameworks to automatically sanitize user inputs and escape special characters, reducing the risk of XSS vulnerabilities.

2. **SQL Injection Attacks:**

- Utilized parameterized queries and stored procedures to prevent SQL injection attacks, ensuring that all database interactions are securely handled.
- Implemented input validation to validate user inputs against predefined rules and sanitize them before processing.

3. **Insecure Authentication and Authorization:**

- Implemented strong authentication mechanisms, including password hashing and salting, to protect user credentials stored in the database.
- Utilized industry-standard authentication protocols and encryption algorithms to securely authenticate and authorize users.

4. **Inadequate Access Controls:**

- Implemented granular access controls to restrict access to sensitive features and administrative functionalities based on user roles and permissions.
- Conducted regular access control reviews to ensure that access permissions are properly configured and enforced.
- Utilized logging and monitoring mechanisms to detect and mitigate any unauthorized access attempts or privilege escalation vulnerabilities.

5. **Encryption and Data Protection:**

- Implemented encryption mechanisms to protect sensitive data, including passwords
- Utilized strong encryption algorithms and secure protocols (e.g., TLS) to encrypt data and ensure its confidentiality and integrity.
- Conducted regular security scans and vulnerability assessments to identify and remediate any vulnerabilities introduced by third-party dependencies.

6. **Secure Configuration and Hardening:**

- Followed security best practices for server and application hardening, including disabling unnecessary services, applying security configurations, and implementing firewall rules.
- Conducted regular security assessments and penetration testing to identify and remediate any configuration weaknesses or security gaps.
- Implemented proactive security measures, such as intrusion detection and prevention systems, to detect and mitigate any security threats or attacks.

Problems during working

➤ **Errors and Vulnerabilities:**

Occurrence of errors and vulnerabilities within the system, leading to potential security breaches, data leaks, or system instability.

- Data Connection Lost Error.
- Merge Code Error During Push and Pull request (Git Connection).
- Services Call error.
- Packages re-install error.

We face these problems during our working.

➤ **Email and Google Services Integration:**

Challenges in integrating email services and Google APIs, resulting in difficulties in implementing user authentication, notifications, or data synchronization functionalities. Email Services are paid. Only first 10 tries are free then google don't provide the password for account email services.

➤ **Authentication and Two-Factor Authentication (2FA):**

Insufficient authentication mechanisms leading to security risks, prompting the need for a robust solution such as paid Google Authenticator or other 2FA options to bolster account security. As it was paid so after some try, we applied the User Approval that was like the same approach as two factor authentication.

➤ **Account Blocking Mechanism:**

Lack of an effective account blocking mechanism causing difficulties in managing and mitigating unauthorized access, potentially exposing the system to security threats or malicious activities.

➤ **Stack Decision Making:**

Challenges in making informed decisions regarding the selection of technology stacks, frameworks, or tools, impacting the project's performance, scalability, and security posture.

Tools and Techniques

Tools

1. **Visual Studio (2022):**

- Utilized Visual Studio 2022 as the integrated development environment (IDE) for coding, debugging, and testing the web application.

2. **HTML (HTML5):**

- Employed HTML5 (Hypertext Markup Language) to structure the content of web pages.



Figure 5: Technology Stack

3. **CSS (Tailwind):**

- Utilized Tailwind CSS framework for styling and layout of the web application, offering utility-first CSS classes for rapid development and customization.

4. **Database (SQL Server):**

- Utilized SQL Server as the database management system to store application data.

5. **Database Controller (SSMS) (Version 19):**

- Used SQL Server Management Studio (SSMS) version 19 as the database management tool for managing and querying the database.

6. **Draw.io Flow Diagram:**

- Utilized Draw.io for creating flow diagrams, illustrating the architecture, data flow, or process flow of the web application.

7. **Creatlily Services Diagram:**

- Utilized Creatlily or similar tools for creating service diagrams, depicting the interaction between different services or components in the application architecture.

Techniques:

1. Entity Framework (Use):

- Leveraged Entity Framework, an Object-Relational Mapping (ORM) framework, for data access and manipulation. Entity Framework simplifies database interactions by allowing developers to work with database entities using object-oriented programming techniques.

2. MVC (Model View Controller):

- Adopted the MVC architectural pattern for structuring the web application into three interconnected components: Model, View, and Controller. MVC promotes separation of concerns, making the application more modular, maintainable, and testable.

These tools and techniques contributed to the efficient development, management, and architecture of your web application, enabling a streamlined development process and ensuring scalability, maintainability, and performance.

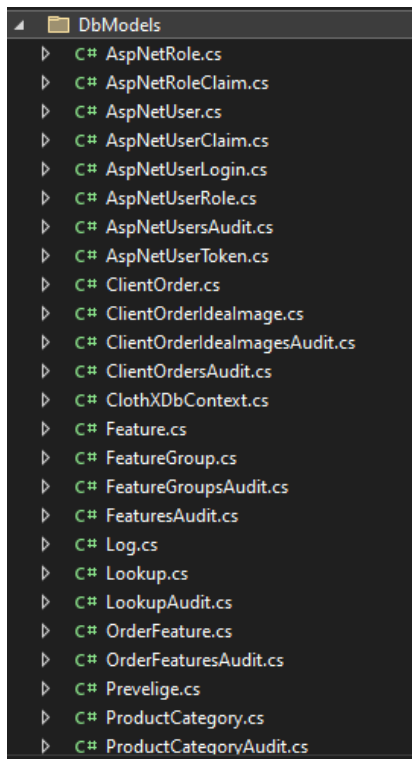


Figure 6: Model

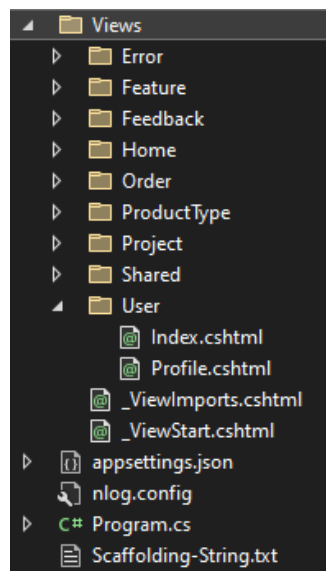


Figure 7: Views

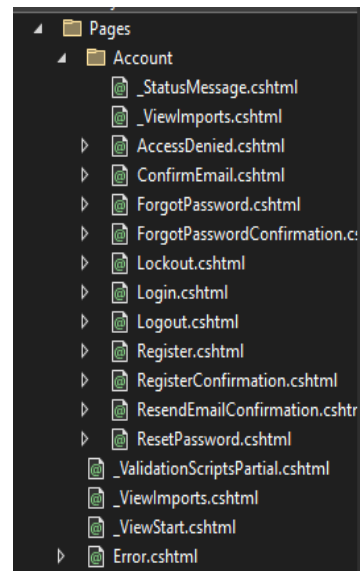


Figure 8: Pages

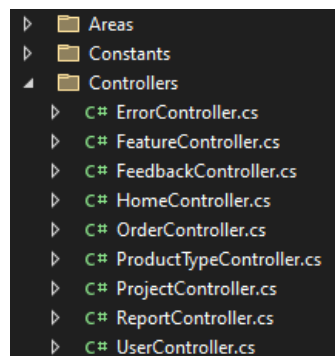
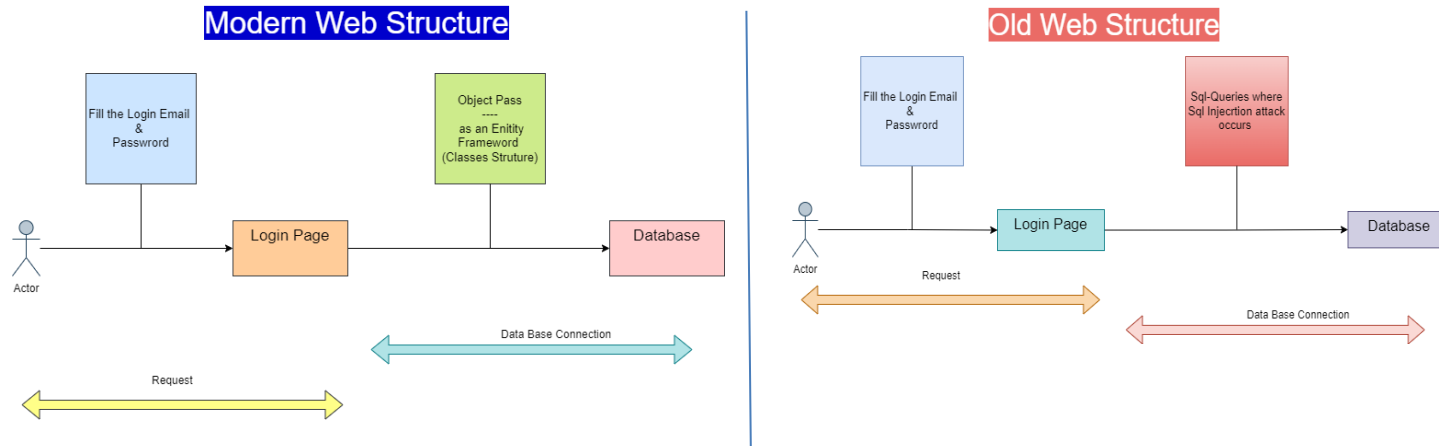


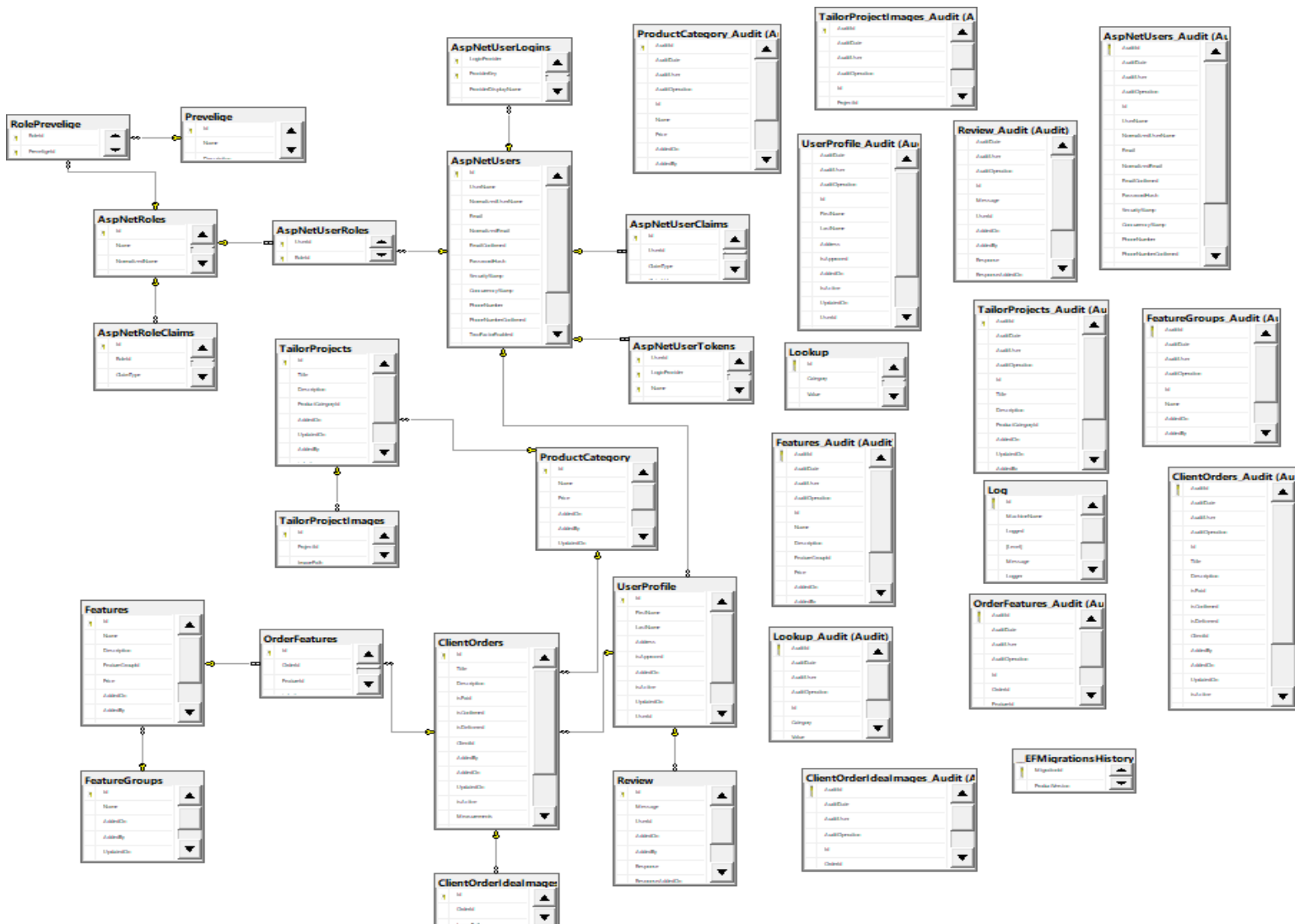
Figure 8: Controllers

Application Structure

We Followed the Modern Web Structure that relates with Object Oriented Programming, Entity framework (classes structure in which web pages are connected not directly linked with database and server [that was the old web structure]).



Database Structure



Interview Insights

Senior Software Engineer @Educative

Muhammad Zubair


For our Professional Interview Insights we engage the meeting with Muhamad Zubair Senior Software Developer at Educative.

Approach

We professionally approach Muhammad Zubair through Google Mails.

Mail

Want to cover your interview about Web Application Firewalls / Security Inbox x

 **Muhammad Kabir Ahmad** <mkabirahmad01@gmail.com> May 4, 2024, 5:40 PM ☆ 😊 ↶ ⋮
to muhammadzubair3321 ▾

Asslamu Alikum Zubair bhai , Hope you are fine.
Zubair bhai , I'm Muhammad Kabir Ahmad from Uet Lahore. I want to cover an interview of yours about Web Application securities.


As, our semester project of Information Security (under the supervision of Prof.Dr. Faiza Iqbal)
is linked with web securities so me and my project mate want to get your insights about my project and on this Topic.

Kindly share your available time. Hope for positive response.

Thank you .


You can visit profile of Prof.Dr.Faiza Iqbal: https://www.linkedin.com/in/faiza-iqbal-58799221?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=android_app

From the linked provided above. Thank you

 **Muhammad Zubair** May 5, 2024, 1:13 PM ☆ 😊 ↶ ⋮
to me ▾

Walikum Assalam Kabir, I have seen your mail and saw your portfolio and Ma'am profile. I would love to share my insights about your project . You can ping me at 11.30pm tonight just share meet link with me, then we will do the meeting.
JazakAllah Kabir.

...

 **Muhammad Kabir Ahmad** <mkabirahmad01@gmail.com> May 5, 2024, 11:25 PM ☆ 😊 ↶ ⋮
to Zubair ▾

Asslamu Alikum Zubair Bhail
Meet Link: <https://meet.google.com/xok-eucx-xrc>
You can join meeting from here

...

↶ Reply ↷ Forward 😊 AI Reply

Main questions with brief Answers

Question:

As a Software Engineer, what are the security measures must be necessary for a web application?

Answer:

Security encompasses multiple aspects, with password hashing as a fundamental practice. Additional security measures depend on the product's sensitivity; however, baseline features like authentication and authorization are essential, whether for sensitive products or social media platforms.

Question:

As a Software Engineer, from what kind of Web Attacks an application should protected in web application firewall?

Answer:

In terms of security priorities, SQL injection should be addressed first, especially when using SQL databases, as unprotected input fields pose a significant threat. Additionally, mitigating cross-site scripting risks by thoroughly validating input fields, such as restricting certain data formats like phone numbers, is crucial for preventing further vulnerabilities.

Question:

As a Software Engineer, have you ever faced any security challenge in your career in educative in authentication and authorization of any application and how you copped up that challenge?

Answer:

Yes, at the beginning of my career. Team lead gave me a problem related to MERN stack web in which I have to create a static page for users and had to apply authentication with out rendering the page. Then I copped up this challenge by using Google API for authentication and authorization (but it was paid and suggested by my senior mate at educative).

Question:

After showing and explaining our project, we asked him for his Insights about our project, as well as asked for review and suggestion for our project.

Answer:

The Answer of this question is mentioned in Review below.

Review

We had shown over project and get the positive reviews from the industry Interviewer as: He appreciates us for covering the key concepts and applying the similar approaches as admin approval rather than losing spirit (as firstly we lose our access from google email services and then can't afford it and Two factor authentication due to Finance Issue as their keys were sold by 19\$ by Google) but he appreciates us to cover these problems with similar approaches rather than skipping them.

Then, he suggests us to use Google Authenticator for Authentication in replace with user approval that we performed in our project and suggests to use Google APIs but as these are paid, he recommends us to use these tools in FYP applications (with the help of funding).

His Words were:

'The project should incorporate two-factor authentication, enhancing security with options like Google Authenticator. Implementing Single Sign-On (SSO) streamlines user experience and bolsters security by leveraging Google's authentication infrastructure. Additionally, considering rate limiting for APIs and leveraging services like Cloudflare for DDoS protection can further fortify the project's security posture. Overall, it's a commendable effort for a semester project, with potential for further enhancements with additional funding.'

You can watch review of our project by Educative' s Senior Software Developer.

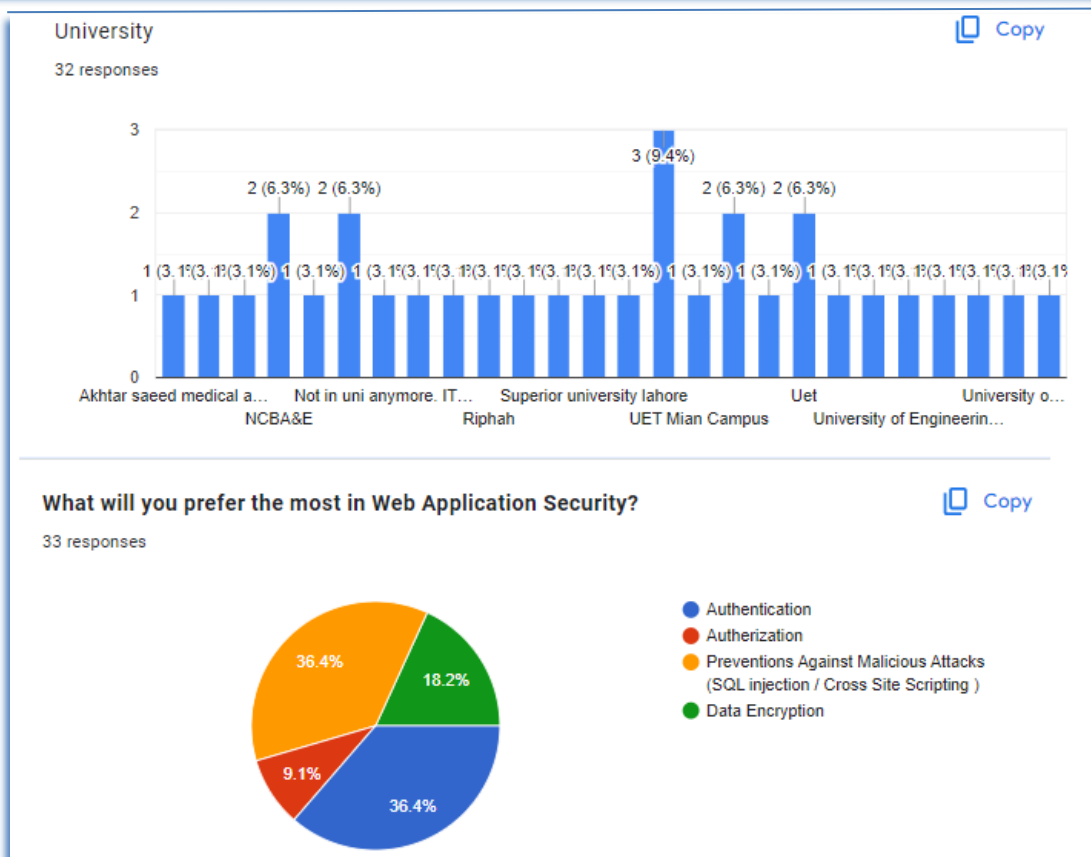
From this [link](#).

What we Learn from Interview

We learn some new techniques and get the information of some new tools from this interview that was described below:

- Use of API's can make you work easy and more headed towards your goal in a concise way rather than making the whole function from zero (whose APIs exist already).
- Use of Google services, as we know it was paid but worthful to use.
- Email Services also can be done through API's.
- Web applications must use cloud services rather than databases if application run on big level.

For People Insights we circulate the Google form in different universities that was shown below with the question and the responses we get from it as:



From this Activity, We get the insights of people from different universities about web application securities.

Visualization & Case Study

The visualization of our application is shown as:

Wire-Frames:

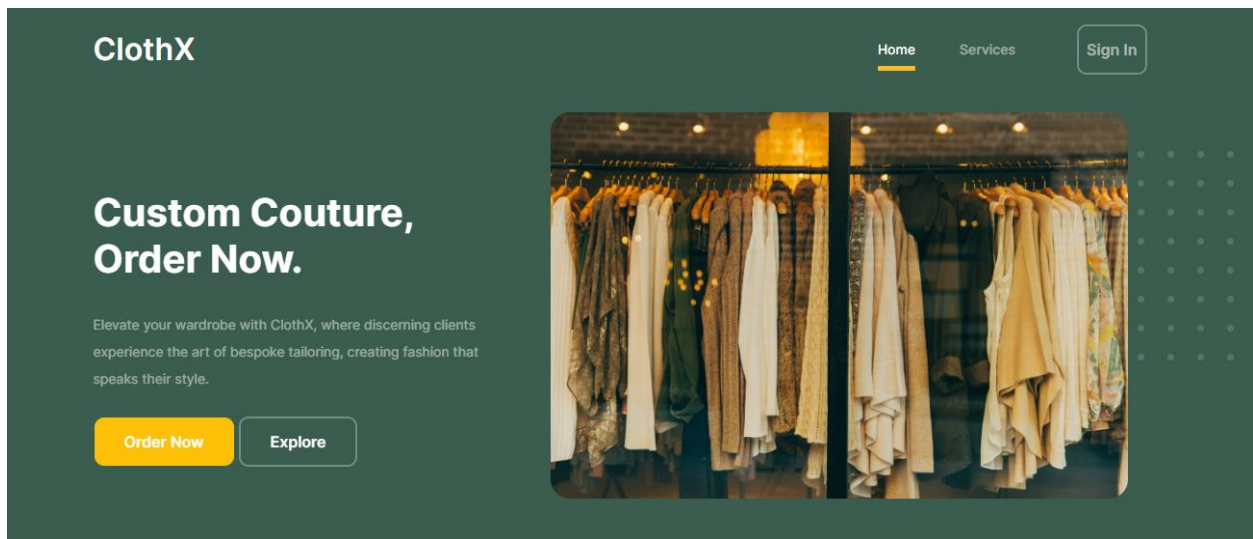


Figure 12: Home Page

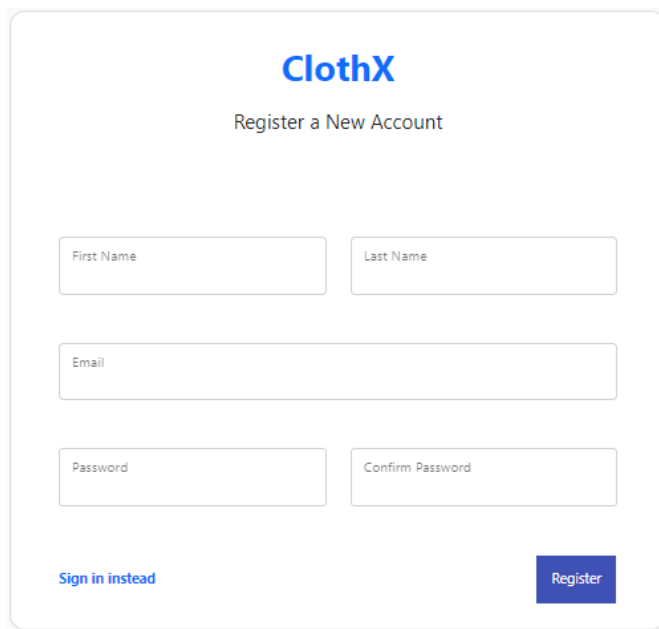
A wireframe of the ClothX Sign Up Page. The header shows the 'ClothX' logo and the text 'Register a New Account'. The form contains four input fields: 'First Name', 'Last Name', 'Email', and 'Password'. The 'Password' field is paired with a 'Confirm Password' field. At the bottom left is a link 'Sign in instead' and at the bottom right is a blue 'Register' button.

Figure 13: Sign Up Page

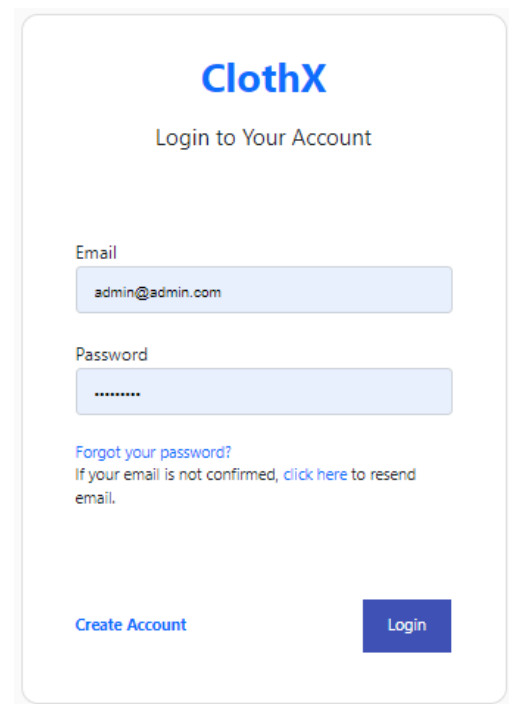
A wireframe of the ClothX Login Page. The header shows the 'ClothX' logo and the text 'Login to Your Account'. The form contains two input fields: 'Email' (with the placeholder 'admin@admin.com') and 'Password' (with the placeholder '*****'). Below the password field is a link 'Forgot your password?' and a text line: 'If your email is not confirmed, [click here](#) to resend email.' At the bottom left is a link 'Create Account' and at the bottom right is a blue 'Login' button.

Figure 14: Login Page

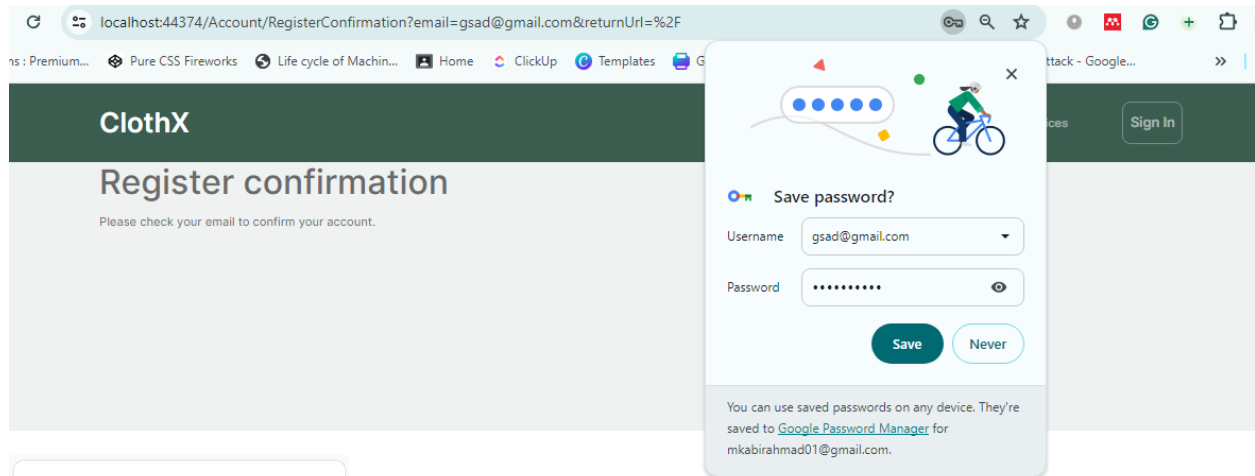


Figure 14: Registration Confirmation page

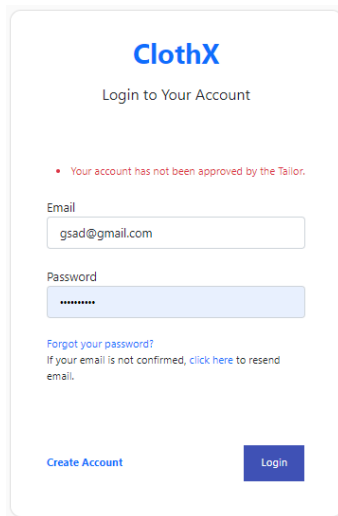


Figure 15: Approval page

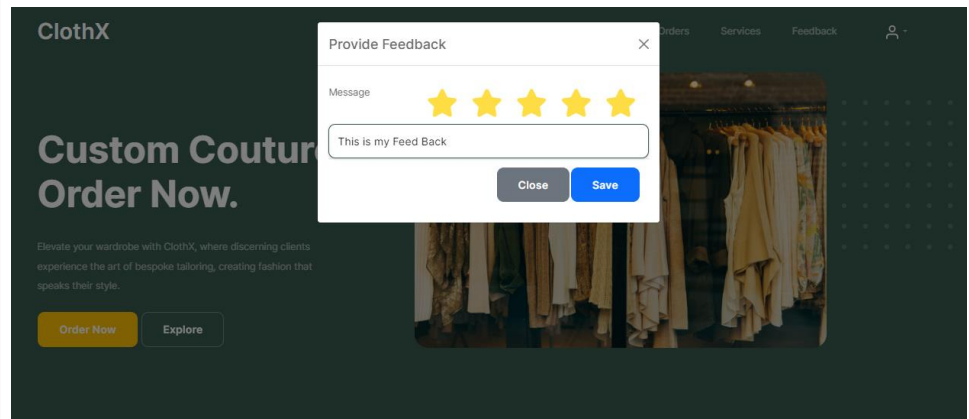


Figure 16: Feed-Back page

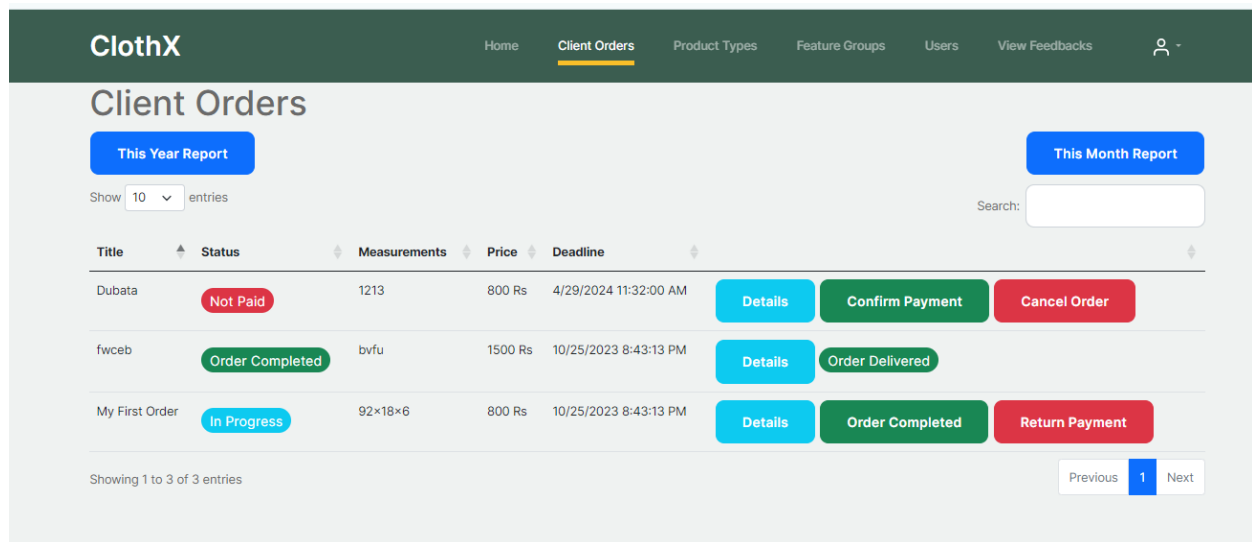


Figure 17: Client Orders

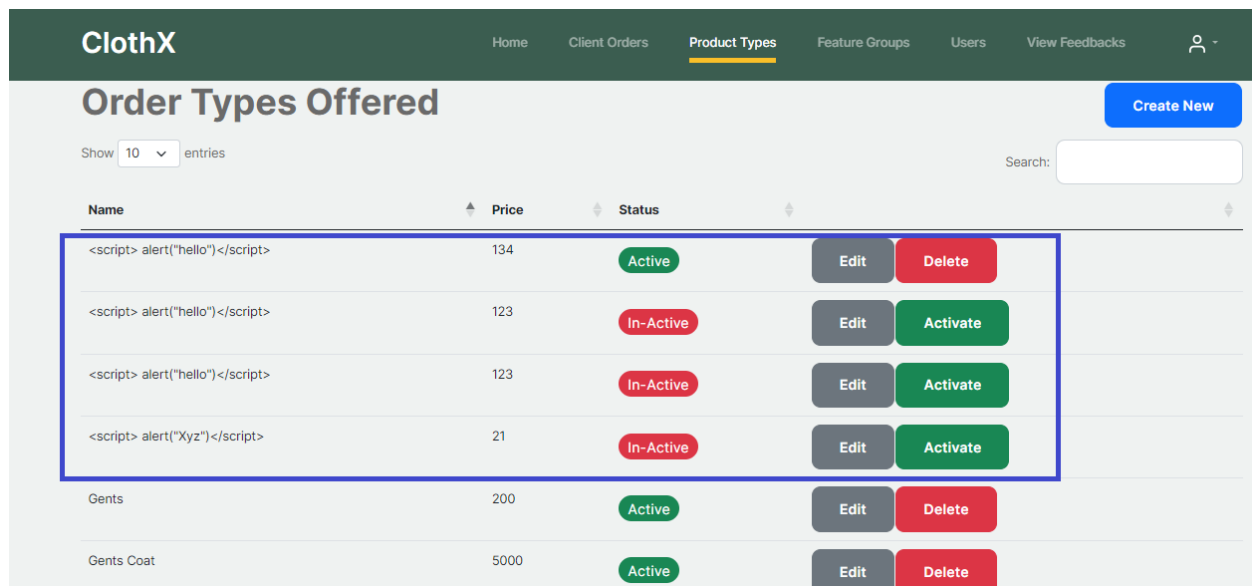


Figure 18: Product Type page where Protection against xss attack is shown

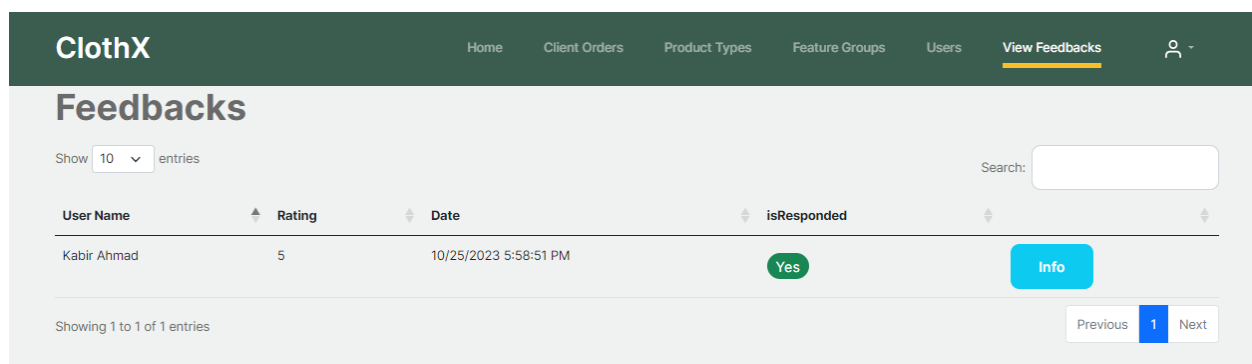


Figure 19: View Feed-Back Page

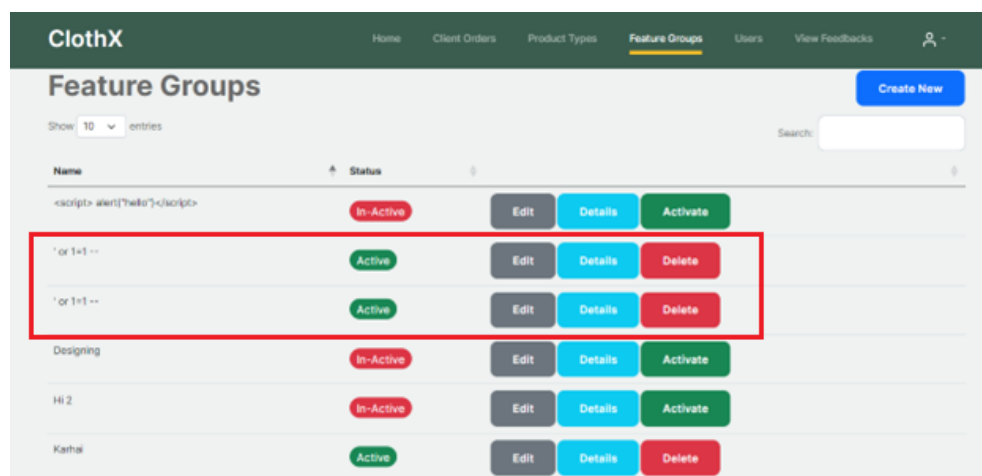


Figure 20: Featured Group page where Protection against SQL Injection attack is shown


ClothX				
Index				
Show 10 entries		Search:		
Name	Address	IsApproved	Status	
ABC ABC		Approved	Active	Reject
Client1		Approved	Active	Reject
Dr.Faiza Iqbal		Approved	Active	Reject
Kabir Ahmad	uet lahore	Approved	Active	Reject
Mahnoor Fatima	Gutter	Approved	Active	Reject
xyz xyz		Approved	Active	Reject
xyz xyz		Approved	Active	Reject
Showing 1 to 7 of 7 entries				Previous 1 Next

Figure 21: View Users Page

ClothX	
Home	Client Orders
Product Types	Feature Groups
Users	View Feedbacks

Account settings

General



Upload Profile photo

Allowed JPG, GIF, or PNG. Max size of 800K

First Name

Kabir

Last Name

Ahmad

Phone Number

0308-4859233

Address

Uet lahore

Submit

Figure 22: Profile Page

ClothX	
Home	Services
Sign In	

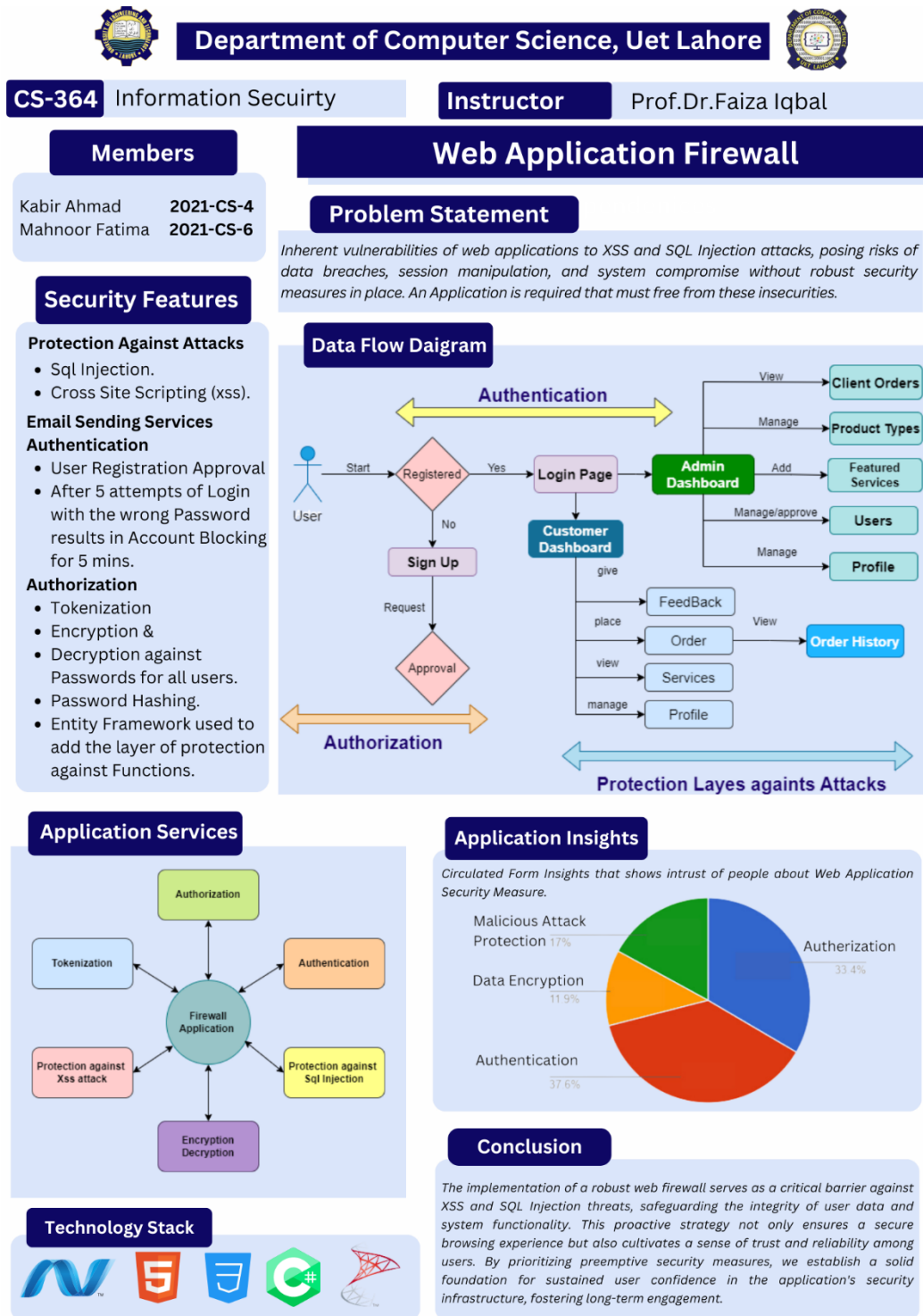
Locked out

This account has been locked out, please try again later.

Figure 23: Account Blockage Page

Project poster:

We made the project poster for our project which is shown as:



Security Best Practice & Future Directions

Security Best Practices Implemented:

1. **User Authentication and Authorization:**

- Implemented a robust user authentication system with password encryption and two-factor authentication (2FA) to ensure secure user access.
- Utilized role-based access control (RBAC) to enforce granular access permissions and restrict unauthorized access to sensitive functionalities.

2. **Input Validation and Sanitization:**

- Implemented rigorous input validation and output encoding to prevent common web attacks such as XSS and SQL injection.
- Utilized parameterized queries and stored procedures to mitigate SQL injection vulnerabilities and ensure secure database interactions.

3. **Session Management and Account Blocking:**

- Implemented secure session management techniques, including session expiration and regeneration, to prevent session-related attacks.
- Implemented account blocking mechanisms to temporarily lock user accounts after multiple failed login attempts, enhancing security against brute force attacks.

4. **Data Encryption and Protection:**

- Utilized strong encryption algorithms to protect sensitive data both in transit and at rest, ensuring confidentiality and integrity.
- Implemented secure password hashing with salt to safeguard user passwords stored in the database against unauthorized access.

Future Directions for Security Enhancement

1. **Continuous Monitoring:** Implement continuous security monitoring mechanisms to detect and respond to security incidents or suspicious activities in real-time.
2. **Google OAuth Integration:** Implement Google OAuth for user authentication, allowing users to sign in using their Google accounts. This enhances security by leveraging Google's robust authentication infrastructure and reducing the risk of password-related security issues.
3. **Enhanced Authentication:** Explore the implementation of advanced authentication methods such as biometric authentication or multi-factor authentication (MFA) to further strengthen user authentication and access controls.
4. **Incident Response Plan:** Develop and regularly update an incident response plan outlining the steps to be taken in the event of a security breach or incident, including communication protocols and recovery procedures.
5. **External Security Testing:** Engage third-party security experts to conduct external security testing, including penetration testing and vulnerability assessments, to identify potential security gaps and validate the effectiveness of existing security measures.

Conclusion

In conclusion, our web application project has achieved significant milestones in enhancing security, usability, and functionality. Key findings and achievements include:

1. **Robust Security Measures:** Implemented advanced security measures such as encryption, input validation, and access controls to protect against common web attacks like XSS and SQL injection. Additionally, the implementation of account blocking and two-factor authentication enhances user account security.
2. **Streamlined User Experience:** Utilized tools like Visual Studio, HTML5, CSS with Tailwind, and Entity Framework MVC to create a user-friendly interface and improve overall user experience.
3. **Efficient Database Management:** Leveraged SQL Server and SQL Server Management Studio (SSMS) for efficient database management, ensuring data integrity and security.
4. **Clear Visualization:** Used Draw.io and Creatlily Services Diagram for creating clear flow diagrams and service diagrams, aiding in visualizing the application architecture and data flow.
5. **Adoption of MVC Architecture:** Adopted the Model-View-Controller (MVC) architectural pattern to enhance code organization, maintainability, and scalability.
6. **Continuous Security Enhancement:** Committed to continuous security monitoring, updates, and training to proactively address emerging threats and maintain a strong security posture.