

Project Report

Table of Contents

Introduction.....	3
Description.....	3
Security, Privacy, and Legal Considerations in the Messaging Platform.....	3
Security Issues:	4
Privacy Issues:.....	4
Legal Issues:	5
Database dump	12

Introduction

In today's interconnected world, communication platforms play a vital role in facilitating interactions and fostering connections among individuals. The advent of messaging platforms has revolutionized how people communicate, enabling instant messaging, sharing of updates, and engagement in discussions with ease. With the proliferation of digital communication, the need for secure, user-friendly messaging platforms has become increasingly apparent.

The aim of this report is to present the final project, a messaging platform designed to address the communication needs of users while prioritizing security, privacy, and legal compliance. The platform offers a comprehensive suite of features, including user registration, messaging, profile management, and feed updates, to create a seamless communication experience. Through the implementation of robust security measures, privacy considerations, and adherence to legal regulations, the platform endeavors to provide users with a safe and trustworthy environment for online communication.

Description

The final project is a messaging platform designed to facilitate communication and interaction between users. It includes features such as user registration, authentication, creating and sharing messages, browsing a feed of posts, accessing personal profiles, and sending private messages. Upon accessing the platform, users are presented with a login screen where they can enter their credentials to access their accounts. New users can register for an account by providing their email, username, and password.

Once logged in, users can navigate through different sections of the platform using the navbar, including the home page, registration page, feed, messages, and profile. The feed section displays a chronological list of posts shared by users, while the messages section allows users to view and send private messages to each other. Users can also create new posts and messages, edit their profiles with personal information such as name, age, job, and city, and customize their profiles with profile pictures.

The platform incorporates security measures such as secure authentication, input validation, data encryption, and session management to protect user accounts and data. Privacy considerations include clear data collection and usage policies, user anonymity options, and end-to-end encryption for private messages.

To ensure legal compliance, the platform includes comprehensive terms of service covering user responsibilities, prohibited activities, intellectual property rights, and limitations of liability. It also complies with relevant data protection regulations such as GDPR and CCPA and implements measures to prevent copyright infringement.

Overall, the messaging platform provides users with a secure, privacy-respecting, and legally compliant environment for communication and interaction online.

Security, Privacy, and Legal Considerations in the Messaging Platform

In today's digital landscape, where data breaches and privacy violations are prevalent, ensuring the security, privacy, and legal compliance of a messaging platform is paramount. This section delves into the

various issues and strategies involved in addressing these concerns within the context of our messaging platform.

Security Issues:

Authentication and Authorization:

Implementing robust authentication mechanisms is crucial to prevent unauthorized access to user accounts. By utilizing secure authentication methods like bcrypt for password hashing, we ensure that user credentials are adequately protected against brute-force attacks and password cracking attempts. Additionally, implementing granular authorization mechanisms helps control access to different parts of the application, ensuring that users only have access to the resources they are authorized to view or modify.

Input Validation:

Proper input validation is essential to mitigate common vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). By validating user inputs on both the client and server sides, we prevent malicious users from exploiting vulnerabilities in the system to execute unauthorized actions or access sensitive information.

Data Encryption:

Encrypting sensitive data, such as user credentials and messages, during transmission over the network using the HTTPS protocol, ensures that data remains confidential and secure against eavesdropping and interception attacks. Additionally, considering encrypting data at rest, especially if storing sensitive information in databases, provides an extra layer of protection against unauthorized access to stored data.

Session Management:

Implementing secure session management practices helps prevent session hijacking and fixation attacks. By using techniques like session tokens with short expiration times and regenerating tokens after successful login, we reduce the risk of unauthorized access to user sessions, thereby enhancing the overall security of the platform.

Privacy Issues:

Data Collection and Usage:

Clearly stating in the privacy policy what user data is collected, how it's used, and whether it's shared with third parties is essential to maintaining transparency and trust with users. Obtaining explicit consent from users before collecting any personal information and providing them with options to opt-out of data sharing helps respect user privacy preferences and rights.

User Anonymity:

Offering users the option to use the platform anonymously or with pseudonyms enhances their privacy and anonymity, protecting them from potential privacy breaches and identity theft. Minimizing the collection and disclosure of personally identifiable information unless necessary for the functionality of the platform further reinforces user anonymity and privacy.

Message Encryption:

Implementing end-to-end encryption for private messages ensures that only the sender and intended recipients can read the messages, safeguarding user privacy from unauthorized access by third parties, including the platform owner. By encrypting messages before transmission and decrypting them only upon receipt by the intended recipients, we ensure the confidentiality and integrity of user communications.

Legal Issues:

Terms of Service:

Drafting comprehensive terms of service (ToS) that outline user responsibilities, prohibited activities, intellectual property rights, and limitations of liability is essential for establishing legal agreements between the platform and its users. Making the ToS easily accessible to users ensures that they are aware of their rights and obligations when using the platform.

Compliance with Regulations:

Ensuring compliance with relevant data protection regulations such as GDPR and CCPA is imperative, especially if the platform serves users in those jurisdictions. Providing mechanisms for users to access, update, and delete their personal data helps fulfill regulatory requirements and protects user privacy rights.

Copyright Infringement:

Implementing measures to prevent users from sharing copyrighted material without proper authorization helps mitigate the risk of copyright infringement. Providing a mechanism for reporting and taking down infringing content upon receiving valid DMCA notices demonstrates a commitment to respecting intellectual property rights and complying with applicable copyright laws.

In conclusion, addressing security, privacy, and legal concerns in our messaging platform requires a comprehensive approach that encompasses robust security measures, transparent privacy practices, and adherence to relevant legal frameworks. By prioritizing the protection of user data and privacy rights, we aim to build trust and confidence among our users while ensuring compliance with legal requirements.

Future Considerations:

- 1. Enhanced Encryption:** Implementing advanced encryption protocols and algorithms to further strengthen message encryption and ensure the confidentiality of user communications.
- 2. Two-Factor Authentication:** Introducing two-factor authentication (2FA) as an additional layer of security to protect user accounts from unauthorized access and enhance overall authentication security.
- 3. Regular Security Audits:** Conducting regular security audits and penetration testing to identify and address potential vulnerabilities and security weaknesses in the messaging platform.
- 4. Privacy-Enhancing Features:** Introducing additional privacy-enhancing features such as ephemeral messaging options and enhanced user control over data sharing and visibility settings.
- 5. Compliance Monitoring:** Continuously monitoring and updating the platform's privacy and legal compliance measures to adapt to evolving regulations and standards, ensuring ongoing adherence to data protection laws.
- 6. User Education:** Implementing user education and awareness programs to educate users about best practices for protecting their privacy and security while using the messaging platform.

7. Integration with Secure Channels: Exploring integration with secure communication channels and platforms to provide users with alternative options for encrypted and confidential communication.

8. Incident Response Plan: Developing a comprehensive incident response plan to effectively handle and mitigate security breaches or privacy incidents that may occur in the future.

By incorporating these future considerations, the messaging platform can continue to evolve and adapt to emerging security, privacy, and legal challenges, ensuring the ongoing protection of user data and privacy rights.

HTML and CSS validation

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by **text input** ☐ css

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Chatroom</title>
  <link rel="stylesheet" href="styles.css">
  <link href='https://fonts.googleapis.com/css?family=Poppins' rel='stylesheet' />
</head>
<body>
  <div class="sidebar left">
    <div class="left">
      <h1>Chatroom</h1>
      <nav>
        <div class="top-br"></div>
```

[Check](#)

Use the Message Filtering button below to display options for hiding/showing particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

1. **Info** Trailing slash on void elements **has no effect and interacts badly with unquoted attribute values.**

[From line 8 column 5 to line 8 column 84](#)

```
ss"> <link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet" /></hea
```

Document checking completed. No errors or warnings to show.

Source

```
1. <!DOCTYPE html>↵
2. <html lang="en">↵
3. <head>↵
4.   <meta charset="UTF-8">↵
5.   <meta name="viewport" content="width=device-width, initial-scale=1.0">↵
6.   <title>Chatroom</title>↵
7.   <link rel="stylesheet" href="styles.css"> ↵
8.   <link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet" />↵
9. </head>↵
10. <body>↵
11.   <div class="sidebar left">↵
12.     <div class="left">↵
13.       <h1>Chatroom</h1>↵
14.       <nav>↵
15.         <div class="top-br"></div>↵
16.         <ul>↵
17.           <li><a href="#" id="home-btn" class="active">Home</a></li>↵
18.           <li><a href="#" id="notifications-btn" class="overlay-trigger">Notifications</a></li>↵
19.           <li><a href="#" class="open-messages overlay-trigger">Messages</a></li>↵
20.           <li><a href="#" id="bookmarks-btn" class="overlay-trigger">Bookmarks</a></li>↵
21.           <li><a href="#" id="friend-requests-btn" class="overlay-trigger">Friend Requests</a></li>↵
22.           <li><a href="#" id="settings-btn" class="overlay-trigger">Settings</a></li>↵
23.         </ul>↵
24.       </nav>↵
25.     </div>↵
26.   </div>↵
27.   <div class="header">↵
28.     <div class="container">↵
29.       <nav>↵
30.         <ul>↵
31.           <li><a href="#signin" id="signin-link">Login</a></li>↵
32.           <li><a href="#register" id="register-link">Registration</a></li>↵
33.         </ul>↵
34.       </nav>↵
35.       <div class="search-bar">↵
36.         <input type="text" placeholder="Search">↵
37.         <button type="submit">Search</button>↵
38.       </div>↵
39.     </div>↵
40.   </div>↵
41.   <div class="user-details-container" id="user-details-container">↵
42.     <span class="close-button" id="close-button">&times;</span>↵
43.   </div>↵
44.   <div class="user-details" id="signin-form">↵
45.     <h2>Sign In</h2>↵
46.     <form>↵
47.       <label for="signin-username">Username:</label>↵
48.       <input type="text" id="signin-username" name="username" required>↵
49.       <label for="signin-password">Password:</label>↵
50.       <input type="password" id="signin-password" name="password" required>↵
51.       <button type="submit">Sign In</button>↵
52.     </form>↵
53.   </div>↵
54.   <!-- Register form -->↵
55.   <div class="user-details" id="register-form">↵
56.     <h2>Register</h2>↵
57.     <form>↵
58.       <label for="reg-username">Username:</label>↵
59.       <input type="text" id="reg-username" name="username" required>↵
60.       <label for="reg-password">Password:</label>↵
61.       <input type="password" id="reg-password" name="password" required>↵
62.       <label for="confirm-password">Confirm Password:</label>↵
63.       <input type="password" id="confirm-password" name="confirm-password" required>↵
64.       <button type="submit">Register</button>↵
65.     </form>↵
66.   </div>↵
67. </div>↵
68. </html>
```



The W3C CSS Validation Service

W3C CSS Validator results for TextArea (CSS level 3 + SVG)

Jump to: [Warnings \(8\)](#) [Validated CSS](#)

W3C CSS Validator results for TextArea (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as [CSS level 3 + SVG](#) !

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:



```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```



```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```

front end of the website



[Home](#) [Registration](#) [Feed](#) [Messages](#) [Profile](#)


Log In

Username

Password

[Submit](#)

New User ? [Sign up here](#)

 [Home](#) [Registration](#) [Feed](#) [Messages](#) [Profile](#)

Registration

Email Address


Name

Password


Confirm Password

Submit


127.0.0.1:5502/root/index.html#

 [Home](#) [Registration](#) [Feed](#) [Messages](#) [Profile](#)


Add new message




Amy 7:43PM
Why ? What Happened ?



Sally 4:27PM
Of course, let me know



Jane 2:11PM
Yay



Jack 12:20PM
Have a good Day.

127.0.0.1:5502/root/index.html#



Add new message

Submit a message

Name

Add new message



share what's new



Jane Smith 2:15PM

Anniversary Weekend Antics !!!



Sally Hansen 12:45PM

Today is my Birthday



[Home](#) [Registration](#) [Feed](#) [Messages](#) [Profile](#)

Create new Profile

Name

Age

Job

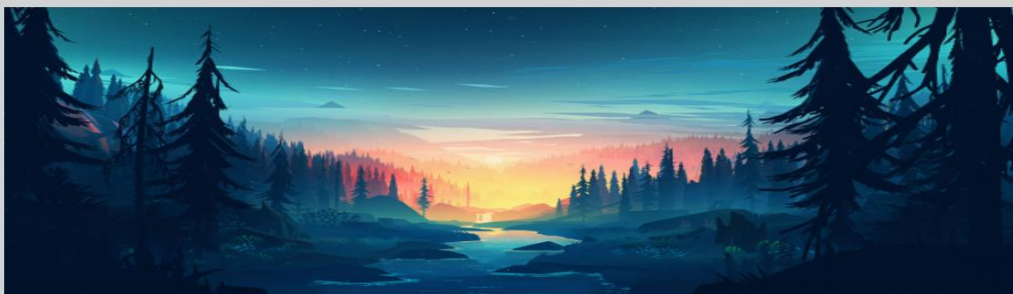
City

Profile Created



[Home](#) [Registration](#) [Feed](#) [Messages](#) [Profile](#)

[Create new Profile](#)



Lauren Freeman

27

Stylist

Paris , France

You 6:43PM

Some Picture From Paris Fashion Week :/



Database dump

MongoDB Compass - localhost:27017/test.users

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases Search

ConnectClub admin config test createmessages createprofiles users

test.users

0 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA

1 - 1 of 1

```
{
  "_id": ObjectId("6610d241bc1741739c4eb91b"),
  "username": "Harwa Awil",
  "password": "12345678",
  "confirmPassword": "12345678",
  "email": "MA3655@live.mdx.ac.uk",
  "__v": 0
}
```

MongoDB Compass - localhost:27017/test.createmessages

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases Search

ConnectClub admin config test createmessages createprofiles users

test.createmessages

1 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA

1 - 2 of 2

```
{
  "_id": ObjectId("6610d2b3bc1741739c4eb922"),
  "content": "Why ? What Happened ?",
  "createdBy": "Amy",
  "__v": 0
}
```

```
{
  "_id": ObjectId("6610d2dc1741739c4eb924"),
  "content": "Offcourse let you know",
  "createdBy": "Sally",
  "__v": 0
}
```

Connect Edit View Collection Help

localhost:27017 ...

{ } My Queries Performance Databases Search

ConnectClub admin config test createprofiles users

test.createprofiles

0 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

EXPLAIN RESET FIND Options

ADD DATA EXPORT DATA

1 - 2 of 2

```

{
  "_id": ObjectId("6610d2feb1741739c4eb926"),
  "name": "Sally",
  "age": "20",
  "job": "Web Developer",
  "city": "UK",
  "__v": 0
}

```

```

{
  "_id": ObjectId("6610d44ebc1741739c4eb931"),
  "name": "Sally",
  "age": "20",
  "job": "Web",
  "city": "UK",
  "__v": 0
}

```

Postman

Home Workspaces Explore Search Postman Sign In Create Account

History New Import

Today

- GET http://localhost:8080/MOO755729/
- GET http://localhost:8080/MOO755729/
- POST http://localhost:8080/MOO755729/
- POST http://localhost:8080/MOO755729/

GET http://localhost:8080/MOO755729/

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 8 ms Size: 441 B Save Response

Pretty Raw Preview Visualize JSON

```

1
2  "message": "User created successfully",
3  "status": 200,
4  "users": [
5    {
6      "_id": "65d7ba35d8b9fa1dac2e4bbb",
7      "name": "Mazwa Aw11",
8      "email": "MA3658@live.mdx.ac.uk",
9      "id": "MOO755729",
10     "__v": 0
11   }
12 ]
13

```

Home Workspaces Explore Search Postman Sign In Create Account

History New Import

POST http://localhost:8080/MOO755729/ GET http://localhost:8000/ + ...

http://localhost:8080/MOO755729/ Save </>

POST http://localhost:8080/MOO755729/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "name": "Mazwa Awil",
3   "email": "MA3658@live.mdx.ac.uk",
4   "id": "MOO755729"
5 }
6
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 12 ms Size: 438 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "User created succesfully",
3   "status": 200,
4   "user": {
5     "name": "Mazwa Awil",
6     "email": "MA3658@live.mdx.ac.uk",
7     "id": "MOO755729",
8     "_id": "65d7ba350809fa1dc2e4bbb",
9     "__v": 0
10  }
11 }
```

Home Workspaces Explore Search Postman Sign In Create Account Minimize

History New Import

[CONFLICT] POST http://localhost:8080/MOO755729/ POST http://localhost:8080/MOO755729/ + ...

http://localhost:8080/MOO755729/ Save </>

POST http://localhost:8080/MOO755729/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "name": "Sally",
3   "age": "20",
4   "job": "Web Develelopez",
5   "city": "UK"
6 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 192 ms Size: 429 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "User created succesfully",
3   "status": 200,
4   "user": {
5     "name": "Sally",
6     "age": "20",
7     "job": "Web Develelopez",
8     "city": "UK",
9     "_id": "6610d838b481cb279c18f6a2",
10    "__v": 0
11  }
12 }
```

History

NewImport

[CONFLICT] POST http://local... POST http://localhost:8080/M/...

▼ Today

- POST http://localhost:8080/MOO755729/
- POST http://localhost:8080/MOO755729/
- POST http://localhost:8080/MOO755729/
- POST http://localhost:8080/MOO755729/
- POST http://localhost:8080/MOO755729/
- POST http://localhost:8080/MOO755729/
- POST http://localhost:8080/MOO755729/

▼ April 4

- POST http://localhost:8080/MOO755729
- GET http://localhost:8080/MOO755729
- POST http://localhost:8080/MOO755729
- GET http://localhost:8080/MOO755729

http://localhost:8080/MOO755729/

POST http://localhost:8080/MOO755729/

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryJSON

```
1{"name": "Sally",
2"message": "offcourse let you know"
3}
4
```

BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 43 msSize: 419 BSave Response

PrettyRawPreviewVisualizeJSON

```
1{"message": "User created succesfully",
2"status": 200,
3"user": {
4  "name": "Sally",
5  "message": "offcourse let you know",
6  "_id": "6610d894d34fa8894f3fa06c",
7  "__v": 0
8}
9
10
```