**Simple Web-Based Chat Application**

### 

### **A PROJECT REPORT**

***Submitted by***

**Dhairya Pandey(2115000351) Anuj Chaudhary(2115000172) Vishal Singh(2115001133)**

***in partial fulfillment for the award of the degree of***

**Bachelor of Technology**

**IN**

Computer Engineering and Application

**GLA University, Mathura**

November 2023

#### BONAFIDE CERTIFICATE

Certified that this project report “Simple Web-Based Chat Application” is the bonafide work of “Dhairya Pandey, Anuj Chaudhary, Vishal Singh” who carried out the project work under my/our supervision.

**SIGNATURE SIGNATURE**

Dr. Rohit Agarwal Mr. Amrendra Singh

**SUPERVISOR**

**HEAD OF THE DEPARTMENT**  Assistant Professor

**Computer Science & Engineering Computer Science & Engineering**

|  |  |  |
| --- | --- | --- |
|  |  |  |

Submitted for the project viva-voce examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**TABLE OF CONTENTS**

**Chapter 1: Introduction**

* 1. . Client Identification/Need Identification
  2. Identification of Problem
  3. Objectives
  4. Scope of the Project

**Chapter 2: Literature Review**

**2.1.** Trends in Real-Time Communication Solutions

2.2. Comparative Analysis of Existing Platforms

2.3. Frameworks and Technologies

2.4. Gaps in Existing Literature

2.5. Problem Definition

**Chapter 3: Design Flow/Process**

* 1. . User Authentication and Login
  2. . Login Handling
  3. . WebSocket Connection
  4. . Chatroom Joining
  5. . Real-Time Chat
  6. . User Interface Updates
  7. . Workflow Summary

**Chapter 4: Results Analysis and Validation**

* 1. Unit Testing
  2. Integration Testing
  3. Functional Testing
  4. User Experience Testing

**Chapter 5: Conclusion and Future Work**

* 1. User Authentication and Login
  2. WebSocket Communication
  3. Chatroom Functionality
  4. Real-Time Messaging
  5. User Authentication Enhancements
  6. Extended Chatroom Features
  7. Security Measures

**Chapter 6: References**

* 1. ……
  2. ……

**Chapter 7: Appendix**

* 1. User Landing Page Code
  2. JavaScript Code

**Chapter 8: User Manual**

**INTRODUCTION**

In an era characterized by dynamic connectivity and global collaboration, effective real-time communication stands as a cornerstone for successful interactions. This project endeavors to address the inherent challenges present in existing communication tools by introducing a web-based chatting application. Grounded in the identified need for a more responsive and feature-rich platform, this project aims to redefine the landscape of online communication.

* 1. **Client Identification/Need Identification**
* **Client:** XYZ Corporation, a medium-sized tech company with teams distributed across various locations.
* **Need:** XYZ Corporation has identified a growing need for seamless and secure real-time communication among its remote and on-site teams. The current communication tools are causing delays and inefficiencies in collaborative projects.
  1. **Identification of Problem**
* **Delays in Communication:** Team members experience delays in receiving and responding to messages, leading to project bottlenecks.
* **Impact on User Experience:** Reports indicate that the limitations in real-time interaction are impacting the overall user experience and hindering the platform's ability to cater to diverse communication needs.
  1. **Objectives**

**This project seeks to address the identified problem by:**

* Developing a web-based chatting application with real-time capabilities.
* Implementing features that enhance user engagement and responsiveness.
* Enabling users to participate in multiple dynamic conversations concurrently.
  1. **Scope of the Project**

The scope encompasses the entire lifecycle of the project, from the identification of the problem and conceptualization of the solution to the development and testing phases. The resulting web-based chatting application aims to not only meet but exceed the expectations of users in terms of real-time communication and collaboration.

#### **LITERATURE REVIEW**

In exploring the landscape of web-based communication tools, it is essential to delve into existing literature to gain insights, understand precedents, and identify established practices. This literature review provides a comprehensive overview of studies, research, and solutions pertinent to the development of a web-based chatting application with a focus on real-time communication and collaboration.

**2.1. Trends in Real-Time Communication Solutions**

Contemporary literature highlights emerging trends in real-time communication solutions, emphasizing the importance of responsiveness, user-friendly interfaces, and support for multimedia interactions. Studies showcase successful implementations of web-based chatting applications that enhance user experiences in various domains.

**2.2. Comparative Analysis of Existing Platforms**

A comparative analysis of existing web-based communication platforms provides valuable insights into the strengths and weaknesses of various solutions. By examining case studies and user feedback, this literature review aims to distill lessons learned and best practices from successful implementations.

**2.3. Frameworks and Technologies**

* Explore frameworks and technologies in real-time communication.
* Technologies like WebSocket are effective for responsive and scalable applications.

**2.4. Gaps in Existing Literature**

* Identify gaps in literature regarding tailored solutions for web-based chatting applications.
* Project aims to address these gaps and overcome identified shortcomings.

**2.5. Problem Definition**

1. **What is to be done:**
2. **Real-Time Interaction Enhancement:**

* **Objective:** Develop a web-based chatting application to enhance real-time interactions among users.
* **Action:** Implement features that reduce message latency and support simultaneous, dynamic conversations.

1. **User Engagement Improvements:**

* **Objective:** Improve user engagement within the platform.
* **Action:** Incorporate features that promote active participation, multimedia interactions, and an intuitive user interface.

1. **Multiple Concurrent Conversations:**

* **Objective:** Enable users to engage in multiple conversations simultaneously within the same chatroom.
* **Action:** Implement a scalable solution that supports concurrent conversations without compromising responsiveness.

1. **How it is to be done:**
2. **Identification Phase:**

* **Task:** Conduct thorough analysis of existing communication tools and user feedback.
* **Objective:** Identify shortcomings and user pain points to inform the development process.

1. **Building Phase:**

* **Task:** Utilize HTML, CSS, and JavaScript for frontend development.
* **Objective:** Create an intuitive user interface that enhances the overall user experience.
* **Task:** Implement a Node.js server and WebSocket for real-time communication.
* **Objective:** Develop a robust backend that ensures seamless and scalable real-time interactions.

1. **Testing Phase:**

* **Task:** Conduct unit testing to validate individual components.
* **Objective:** Ensure each component functions as intended.
* **Task:** Perform system testing to evaluate overall application functionality.
* **Objective:** Assess the platform's performance in real-world scenarios and user scenarios.

1. **What not to be done:**
2. **Avoid Overlooking User Experience:**

* **Caution:** Do not compromise on the user experience by neglecting intuitive design and responsiveness.

1. **Avoid Message Latency:**

* Caution: Do not implement features that introduce delays in message delivery.

1. **Avoid Complexity in Concurrent Conversations:**

* **Caution:** Do not create a system that becomes cumbersome or complex when users engage in multiple conversations simultaneously.

#### **DESIGN FLOW/PROCESS**

* 1. **User Authentication and Login:**

1. **Frontend:** 
   * **Task:** Create a user interface with an input field for the username.
   * **Objective:** Allow users to input their names.
2. **Frontend (JavaScript):** 
   * **Task:** Implement a function (joinChat) triggered by the "Click To Enter" button.
   * **Objective:** Capture the entered username and initiate the login process.
   1. **Login Handling:**
3. **Frontend (JavaScript):**

* **Task:** Validate the entered username (ensure it is not empty, etc.).

1. **Backend (Node.js):**

* **Task:** Handle the login request from the frontend.
* **Objective:** Establish a connection to the WebSocket server.
  1. **WebSocket Connection:**

1. **Backend (Node.js - Using a WebSocket Library):**

* **Task:** Set up a WebSocket server.
* **Objective:** Enable real-time communication between clients and the server.
  1. **Chatroom Joining:**

1. **Backend (Node.js):**

* **Task:** Upon successful login, assign the user to a chatroom.
* **Objective:** Prepare the user for participation in the chatroom.
  1. **Real-Time Chat:**

1. **Frontend (JavaScript):**

* **Task:** Establish a WebSocket connection from the client-side.
* **Objective:** Enable real-time communication between the client and server.

1. **Backend (Node.js):**

* **Task:** Handle incoming messages from clients and broadcast them to the appropriate chatroom.
* **Objective:** Facilitate real-time messaging within the chatroom.
  1. **User Interface Updates:**

1. **Frontend (JavaScript):**

* **Task:** Dynamically update the user interface with incoming messages.
* **Objective:** Provide a responsive and real-time chat experience.
  1. **Workflow Summary:**

1. User enters a name on the login page.
2. The frontend validates and sends a login request to the WebSocket server.
3. The WebSocket server assigns the user to a chatroom upon successful login.
4. A WebSocket connection is established between the client and server.
5. Real-time messaging is facilitated through WebSocket communication.
6. The user interface is dynamically updated with incoming messages.

**RESULTS ANALYSIS AND VALIDATION**

* 1. **Unit Testing:**

1. **Frontend:** Confirm Join chat function correctly handles and validates usernames.
2. **Backend:** Verify server processes login requests and assigns users to chatrooms.
   1. **Integration Testing:**
3. Test seamless integration between frontend and backend during login.
4. Confirm successful establishment of WebSocket connections.
   1. **Functional Testing:**
5. Confirm users join the correct chatroom after login.
6. Verify real-time messaging works within the chatroom.
   1. **User Experience Testing:**
7. Validate user interface updates in real-time with incoming messages.

#### **CONCLUSION AND FUTURE WORK**

1. **Conclusion:**
   1. **User Authentication and Login:**

* Implemented a user-friendly login interface where users can enter their names to join the chatroom.
  1. **WebSocket Communication:**
* Established WebSocket connections to enable real-time communication between clients and the server.
  1. **Chatroom Functionality:**
* Assigned users to specific chatrooms upon successful login, facilitating multiple concurrent conversations.
  1. **Real-Time Messaging:**
* Implemented real-time messaging functionality, allowing users to engage in dynamic conversations.

1. **Future Work**
   1. **User Authentication Enhancements:**

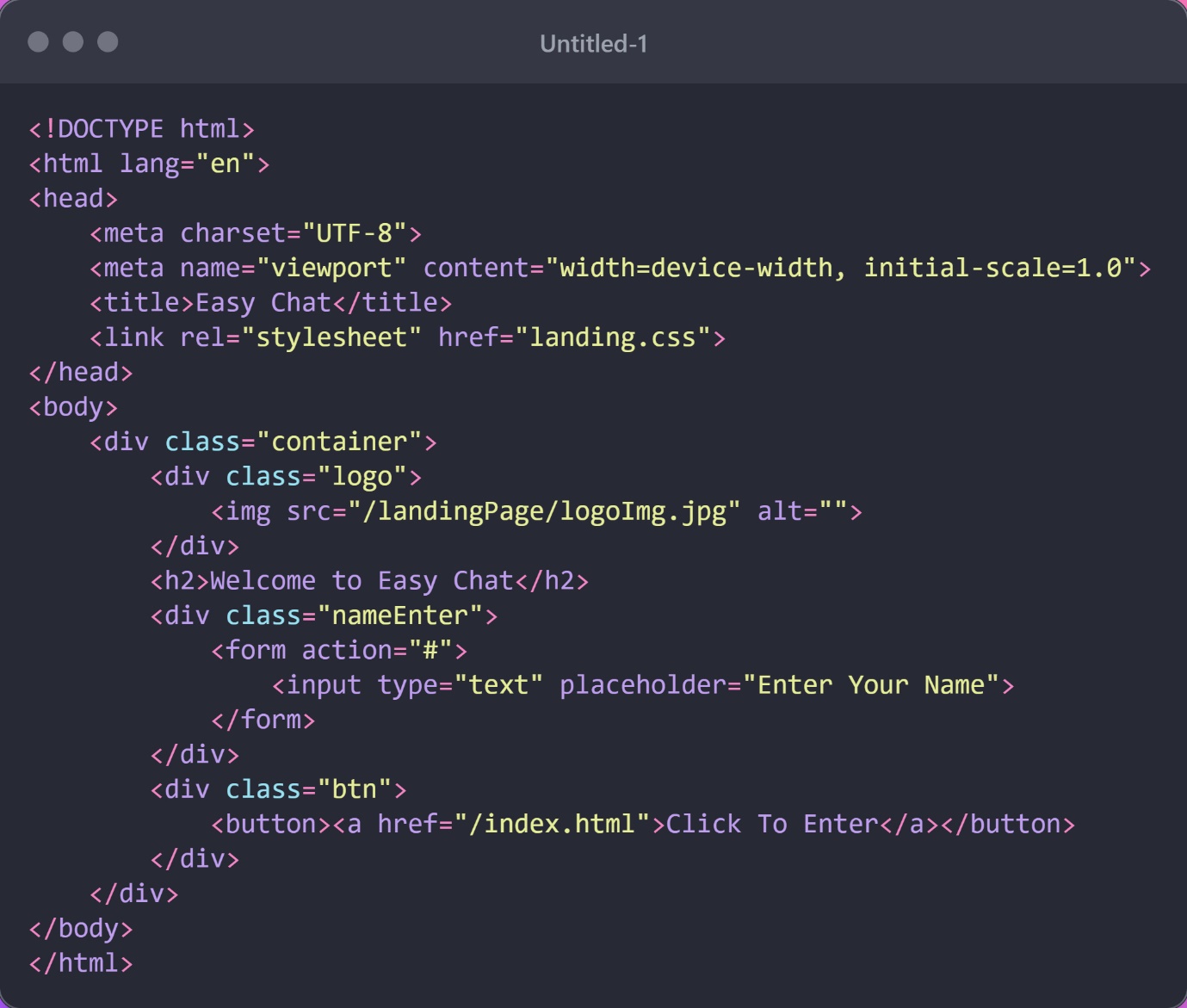
* Implement more robust user authentication mechanisms, including user accounts and secure login processes.
  1. **Extended Chatroom Features:**
* Explore additional chatroom features, such as private messaging, multimedia sharing, and user roles.
  1. **Security Measures:**
* Implement security measures to protect against potential vulnerabilities, ensuring a secure communication environment.

**REFERENCES**

* 1. WebSocket Chat App Using Node.js & Socket.io. YouTube Video.
  2. Socket.IO. (2022). socket.io. GitHub Repository.

**APPENDIX**

* 1. **User Landing Page Code**

****

* 1. **JavaScript Code**

**A screen shot of a computer code

Description automatically generated**

**USER MANUAL**

**Step.1.** User Enter His/Her Name Here to Join the Chat Room

A screenshot of a computer

Description automatically generated

**Step.2.** Then After Clicking On Button User must again that same to verify his identity again

A screenshot of a computer screen

Description automatically generated

**Step.3.** Last after Enter Name again User Can chat in the Chat Room by Sending Messages Like Below

A screenshot of a video game

Description automatically generated