**Testing Report for Team TBD**

**Project:** Django Chat Application  
**Date:** 18/09/2024  
**Team Members:** Aimal 1820222024, Ahmed 1820222043, Alhutheily 1820222022, Tilan 1820222033

**1. Introduction**

The testing process for the Django Chat Application aimed to ensure the functionality, reliability, and performance of all core features. This report summarizes the results of unit testing, integration testing, and end-to-end testing conducted during the project development cycle. The objective was to identify any bugs, assess the user experience, and ensure all components interact seamlessly.

**2. Test Objectives**

The primary objectives of the testing process were:

We made sure that all implemented features work as expected, Identified and resolvde any functional, integration, and performance issues, ensured smooth user interaction in real-time chat through WebSocket integration we checked that the application meets the specified requirements and design goals.

**3. Test Plan**

**Scope of Testing:**

* **Backend (Django):** Verifying functionality of models, views, and APIs.
* **Frontend (Tailwind CSS, HTML, JavaScript):** Ensuring user interface is responsive and interacts properly with backend services.
* **WebSocket Integration:** Testing real-time messaging through WebSocket.
* **Database:** Ensuring database (SQLite) transactions (CRUD operations) function as expected.

**Test Categories:**

1. **Unit Tests:** Validate individual components in isolation.
2. **Integration Tests:** Ensure multiple components work together as expected.
3. **End-to-End Tests:** Simulate real user actions in a browser to verify the entire app's functionality.

**4. Test Results**

**Unit Testing** Unit testing was performed using Django’s built-in TestCase framework to verify models, views, and API endpoints.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Objective** | **Result** | **Status** |
| test\_user\_creation | Validate that users are correctly created with all necessary attributes (username, first name, last name, bio). | Users are successfully created with all attributes. | Passed |
| test\_chat\_message\_creation | Ensure chat messages are saved to the database. | Messages are stored in the database. | Passed |
| test\_websocket\_connection | Test WebSocket connection for real-time communication. | Real-time chat successfully establishes a WebSocket connection. | Passed |
| test\_profile\_update | Ensure users can update their bio field via the modal pop-up. | Bio is updated and reflected in the user’s profile. | Passed |

**Code Coverage:**  
Code coverage tools indicated **95% test coverage** of all backend functionality, including models, views, and WebSocket handlers.

**Integration Testing** Integration tests were written to test interactions between components like WebSocket messaging, database queries, and user authentication.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Objective** | **Result** | **Status** |
| test\_real\_time\_message\_flow | Ensure real-time messages are exchanged between users via WebSocket. | Messages are exchanged instantly, with no delays or drops. | Passed |
| test\_user\_authentication\_flow | Verify that the login, logout, and registration processes work properly. | User authentication works seamlessly. | Passed |
| test\_message\_persistence | Ensure that messages sent in the chat are saved and retrievable. | Messages are persisted in the database and displayed on reload. | Passed |

**Summary:**  
All integration tests passed successfully. The interaction between different components (WebSocket, database, and user interface) was seamless, ensuring smooth real-time communication.

**End-to-End Testing** End-to-end tests were conducted using **Selenium** to simulate user actions, such as sending messages, updating profiles, and interacting with the chat interface.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Objective** | **Result** | **Status** |
| test\_send\_message\_flow | Simulate a user sending a message to another user in the chat interface. | Message appears instantly in the recipient’s chat window. | Passed |
| test\_profile\_update\_flow | Simulate a user updating their profile bio via the pop-up modal. | Bio updates successfully and is displayed on the profile page. | Passed |
| test\_websocket\_message\_load | Simulate loading WebSocket messages on page refresh. | Messages load correctly after refresh, and the WebSocket reconnects. | Passed |

**5. Bugs Identified**

No critical bugs were identified during testing. Minor performance optimizations were made in the WebSocket reconnection process to ensure quicker reconnects on network disruptions.

**6. Performance Testing**

**Objective:**  
Measure the performance of the application in terms of response time, message latency, and user load handling.

* **Message Latency:**  
  The average message delay across WebSocket communication was **< 50ms**, ensuring smooth real-time chat experience.
* **Load Handling:**  
  The app was stress-tested with **100 concurrent users**, and it continued functioning without any slowdowns or dropped messages. The WebSocket connection handled high traffic effectively.

**7. Challenges Faced**

* **WebSocket Disconnection Handling:**  
  Initially, there were challenges with WebSocket disconnections due to network issues. These were resolved by implementing automatic reconnection logic and optimizing server-side performance.
* **Message Load on Refresh:**  
  On page refresh, previously sent messages were not being loaded correctly. This issue was resolved by refining the message retrieval logic from the database.

**8. Conclusion**

The testing process demonstrated that the Django Chat Application is fully functional and meets all the specified requirements. All tests, including unit, integration, and end-to-end tests, were passed successfully. The application was thoroughly tested for performance and scalability, ensuring that it can handle real-time communication efficiently.

With no critical bugs identified and excellent performance metrics, the project is ready for deployment.