**Task 3: Create Quality and Testing Requirements for FYP - AI Chatbot for Mental Health Support**

Your AI Chatbot for Mental Health Support could stand out with the following **innovative features**:

1. **Emotionally Adaptive Response System**: The chatbot adjusts its tone and content based on the user’s emotional state, learning from past interactions to provide more personalized, empathetic support over time.
2. **AI-Powered Conversation Memory**: It remembers user preferences, goals, and coping mechanisms securely, offering personalized advice and continuous support while respecting privacy.
3. **Real-Time Stress Level Detection**: The chatbot detects stress through voice tone and text analysis, offering immediate coping strategies or connecting the user with a therapist.
4. **Virtual Support Groups with AI Moderation**: Users can join peer support groups moderated by AI to ensure safe, healthy conversations while providing insights to improve group dynamics.
5. **Self-Reflection Journal with AI Insight**: The chatbot helps users reflect on their feelings, offering feedback based on sentiment analysis, encouraging self-awareness, and tracking personal growth.
6. **Mindfulness Coach with Adaptive Meditation**: The chatbot adapts mindfulness practices (e.g., meditation, breathing exercises) in real-time based on the user’s stress levels and preferences.
7. **CBT-based Real-Time Support**: It offers Cognitive Behavioral Therapy sessions, helping users identify and challenge negative thought patterns, with gamified progress tracking.

These features combine **AI**, **NLP**, and **behavioral science** to create a unique, personalized mental health experience that adapts to user needs, offering continuous, empathetic support.

**1. Non-Functional Requirements**

**Performance Requirements**

The performance of the AI Chatbot will directly impact its effectiveness in providing real-time mental health support. These are the key performance benchmarks:

**1.1 Response Time**

* **Definition**: The time taken by the chatbot to respond to user input.
* **Goal**: The chatbot should respond within **2 seconds** of receiving a user’s input, ensuring a smooth conversational flow.
* **Test Metric**: Measure response time for different types of user inputs (text, voice).
* **Tools for Testing**: k6, Apache JMeter.

**1.2 Throughput**

* **Definition**: The number of users the system can support simultaneously.
* **Goal**: The system should be able to handle **1000 concurrent users** without degradation in performance.
* **Test Metric**: Requests per second (RPS).
* **Tools for Testing**: Load testing with k6.

**1.3 Scalability**

* **Definition**: The system should be able to scale based on load, ensuring that the performance does not degrade under increasing usage.
* **Goal**: The application should be able to scale up to handle up to **10,000 concurrent users** without performance degradation.
* **Tools for Testing**: Horizontal scaling and testing with cloud services like AWS, Azure.

**1.4 Availability**

* **Definition**: The system should be highly available with minimal downtime.
* **Goal**: **99.9% uptime** to ensure availability of support to users at all times.
* **Test Metric**: Uptime monitoring.
* **Tools for Testing**: Cloud monitoring tools like Datadog, Prometheus.

**Security Requirements for Restful API**

The AI Chatbot will be accessing and processing sensitive mental health data. Hence, security is paramount to ensure that users' privacy and data integrity are maintained.

**2.1 Authentication and Authorization**

* **Definition**: Ensure that only authenticated users can access the chatbot and associated data.
* **Goal**: Implement **OAuth 2.0** or **JWT-based** authentication for secure access.
* **Test Metric**: Verify unauthorized access attempts.
* **Tools for Testing**: Postman, OWASP ZAP.

**2.2 Data Encryption**

* **Definition**: All communication between the client and server must be encrypted.
* **Goal**: Use **HTTPS/TLS** for secure data transmission.
* **Test Metric**: Ensure that all endpoints support HTTPS with valid SSL certificates.
* **Tools for Testing**: SSL Labs, Burp Suite.

**2.3 Data Privacy**

* **Definition**: Personal data should be protected and only accessible to authorized users.
* **Goal**: The chatbot should only store the user data locally on the device and should not retain personal information longer than needed.
* **Test Metric**: Verify data retention policies, ensure data is anonymized or deleted after a session.
* **Tools for Testing**: Privacy testing tools, GDPR compliance testing.

**2.4 Input Validation**

* **Definition**: Prevent injection attacks and ensure proper input sanitization.
* **Goal**: Validate all inputs to prevent XSS, SQL injection, and other malicious attacks.
* **Test Metric**: Attempt attacks with SQL injection, XSS, and other malicious inputs.
* **Tools for Testing**: OWASP ZAP, Burp Suite.

**2.5 API Rate Limiting**

* **Definition**: Protect the system from abusive usage and ensure fair use.
* **Goal**: Implement rate-limiting for each API to handle a maximum number of requests per minute (e.g., **1000 requests per minute per user**).
* **Test Metric**: Simulate high request rates to test API throttling behavior.
* **Tools for Testing**: Postman, k6.

**2. Software Test Plan for AI Chatbot for Mental Health Support**

**3.1 Test Strategies**

**3.1.1 Unit Testing**

* **Definition**: Ensure that individual components of the chatbot (e.g., text analysis, NLP engine, user memory) function as expected.
* **Goal**: Achieve **90% code coverage**.
* **Tools for Testing**: JUnit, Mocha.

**3.1.2 Functional Testing**

* **Definition**: Test individual features of the chatbot, such as conversation flow, mood detection, and personalized responses.
* **Goal**: All features should work as expected and provide meaningful responses to users.
* **Tools for Testing**: Selenium, Appium.

**3.1.3 Regression Testing**

* **Definition**: Ensure that new changes or features do not break existing functionality.
* **Goal**: Run automated tests after every new code release.
* **Tools for Testing**: Selenium, JUnit.

**3.1.4 Load Testing**

* **Definition**: Simulate high traffic to check how the system behaves under heavy load.
* **Goal**: Ensure the system can handle **1000 concurrent users** without performance degradation.
* **Tools for Testing**: k6, JMeter.

**3.1.5 Integration Testing**

* **Definition**: Ensure the integration between the chatbot and its APIs (such as NLP models, database, or external services) is smooth.
* **Goal**: Verify that data flow between components is accurate.
* **Tools for Testing**: Postman, Rest Assured.

**3.1.6 Security Testing**

* **Definition**: Test the security aspects of the chatbot, including data protection and authorization.
* **Goal**: Ensure secure communication, proper authentication, and data encryption.
* **Tools for Testing**: OWASP ZAP, Burp Suite.

**3.1.7 User Acceptance Testing (UAT)**

* **Definition**: Verify that the chatbot meets the needs of the target users and offers value.
* **Goal**: Conduct real-user testing to verify if the chatbot is effective in providing mental health support.
* **Tools for Testing**: Manual testing, SurveyMonkey for feedback.

**3.2 Tools and Frameworks**

Here’s a list of tools you can use for testing different aspects of the AI Chatbot:

| **Test Type** | **Tools** |
| --- | --- |
| Unit Testing | JUnit, Mocha, Chai |
| Functional Testing | Selenium, Appium |
| Regression Testing | Selenium, JUnit |
| Load Testing | k6, JMeter, Artillery |
| Integration Testing | Postman, Rest Assured |
| Security Testing | OWASP ZAP, Burp Suite |
| UAT Testing | Manual Testing, SurveyMonkey |

**3.3 Test Plan Example**

**Test Case 1: User Authentication**

* **Objective**: Ensure that users can only access the chatbot after authentication.
* **Pre-condition**: User must log in using valid credentials.
* **Test Steps**:
  1. Open the chatbot.
  2. Enter valid username and password.
  3. Click "Log In".
  4. Verify successful login and access to the chatbot.
* **Expected Result**: User is successfully logged in and redirected to the chatbot interface.

**Test Case 2: Sentiment Analysis**

* **Objective**: Ensure that the chatbot can accurately detect the user's mood from text input.
* **Pre-condition**: User is logged in and interacting with the chatbot.
* **Test Steps**:
  1. Type a negative sentence like “I’m feeling really down today”.
  2. Submit the input.
  3. Observe the chatbot’s response.
* **Expected Result**: The chatbot should respond empathetically and suggest helpful strategies for managing low mood.

**Test Case 3: Rate Limiting**

* **Objective**: Ensure that the API enforces rate-limiting.
* **Pre-condition**: A registered user has access to the API.
* **Test Steps**:
  1. Send **1001 API requests** within a minute.
  2. Observe the response for the 1001st request.
* **Expected Result**: The system should respond with a **rate limit exceeded** error for the 1001st request.

**3.4 Performance Testing Scenarios**

**Scenario 1: Stress Testing**

* **Objective**: Test the system under heavy load to identify breaking points.
* **Test Steps**:
  1. Simulate **2000 concurrent users**.
  2. Measure the chatbot's response times and stability.
* **Expected Result**: The system should remain responsive for the first 1000 users and gradually degrade performance after the limit.

**Scenario 2: Load Testing**

* **Objective**: Test the system's capacity to handle multiple simultaneous requests.
* **Test Steps**:
  1. Gradually increase traffic from **50 to 1000 users**.
  2. Monitor system performance and response time.
* **Expected Result**: The system should maintain response times within 2 seconds for up to 1000 users.

**Conclusion**

This document outlines the **quality** and **testing requirements** for the **AI Chatbot for Mental Health Support**. By adhering to the performance, security, and testing requirements, the chatbot will be robust, secure, and capable of handling user needs

effectively.