**Nora Framework Chatbot Documentation**

**Table of Contents**

1. **Introduction**
   * 1.1 Purpose
   * 1.2 Scope
   * 1.3 Audience
   * 1.4 Document Conventions
   * 1.5 Terminology
2. **Requirements**
   * 2.1 Nora Framework Overview
   * 2.2 Chatbot Features
   * 2.3 System Requirements
   * 2.4 Language Support
3. **Development Tools**
   * 3.1 Programming Language
   * 3.2 Integrated Development Environment (IDE)
   * 3.3 Version Control
   * 3.4 Natural Language Processing (NLP) Libraries (if used)
4. **Design**
   * 4.1 Conversation Flow
   * 4.2 File Verification Logic
   * 4.3 Class Structure
   * 4.4 Regular Expressions
   * 4.5 Conversation Logic
5. **Implementation**
   * 5.1 Chatbot Class
   * 5.2 Regular Expressions
   * 5.3 Conversation Logic
6. **Testing**
   * 6.1 Unit Testing
   * 6.2 Conversation Testing
   * 6.3 File Verification Testing
7. **Documentation**
   * 7.1 Code Documentation
   * 7.2 User Guide
8. **Deployment**
   * 8.1 Deployment Platform
   * 8.2 Integration
9. **Security**
   * 9.1 Secure File Handling
   * 9.2 Authentication and Authorization
10. **Continuous Improvement**
    * 10.1 User Feedback
    * 10.2 Updates
11. **Monitoring and Maintenance**
    * 11.1 Monitoring
    * 11.2 Maintenance
12. **Conclusion**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to provide comprehensive documentation for the Nora Framework Chatbot. It covers details about the design, implementation, usage, and maintenance of the chatbot.

**1.2 Scope**

This documentation is intended for developers, maintainers, and users involved in the Nora Framework Chatbot project. It outlines the project's goals, features, and technical details.

**1.3 Audience**

The primary audience for this documentation includes developers, project managers, and anyone involved in the deployment or usage of the Nora Framework Chatbot.

**1.4 Document Conventions**

* **Bold Text:** Indicates section headings.
* *Italic Text:* Indicates placeholders or user-input information.

**1.5 Terminology**

* **Nora Framework:** The set of guidelines and standards for file management.
* **Chatbot:** The Nora Framework Chatbot developed as part of this project.

**2. Requirements**

**2.1 Nora Framework Overview**

Provide an overview of the Nora framework, outlining its purpose and guidelines.

**2.2 Chatbot Features**

List and briefly describe the features of the Nora Framework Chatbot, including information provision, file submission guidance, and file verification.

**2.3 System Requirements**

Specify the system requirements for running the chatbot, including supported platforms and dependencies.

**2.4 Language Support**

Outline the supported languages for user interactions, emphasizing English and Arabic.

**3. Development Tools**

**3.1 Programming Language**

The chatbot is developed using the Python programming language.

**3.2 Integrated Development Environment (IDE)**

The preferred IDE for development is [Your Preferred IDE].

**3.3 Version Control**

The project uses Git for version control. The repository is hosted on [Git Hosting Service].

**3.4 Natural Language Processing (NLP) Libraries (if used)**

If applicable, specify the NLP libraries used for language understanding.

**4. Design**

**4.1 Conversation Flow**

Provide a detailed explanation of the conversation flow, illustrating how the chatbot responds to different user queries.

**4.2 File Verification Logic**

Describe the logic implemented for verifying files against the Nora framework guidelines.

**4.3 Class Structure**

Explain the structure of the **NoraChatbot** class, highlighting key methods and attributes.

**4.4 Regular Expressions**

Detail the regular expressions used for matching and verifying file names.

**4.5 Conversation Logic**

Elaborate on the logic implemented to handle different user inputs and generate appropriate responses.

**5. Implementation**

**5.1 Chatbot Class**

Provide code snippets or excerpts illustrating the implementation of the **NoraChatbot** class.

**5.2 Regular Expressions**

Include code examples demonstrating the implementation of regular expressions for file name matching.

**5.3 Conversation Logic**

Show examples of conversation logic, explaining how the chatbot generates responses.

**6. Testing**

**6.1 Unit Testing**

Describe the unit testing approach, including key tests for individual methods.

**6.2 Conversation Testing**

Explain the process of conversation testing, providing sample dialogues and expected outcomes.

**6.3 File Verification Testing**

Detail the testing procedures for file verification, including scenarios with compliant and non-compliant files.

**7. Documentation**

**7.1 Code Documentation**

Explain the code documentation standards followed, including comments and docstrings.

**7.2 User Guide**

Provide a user guide outlining how users can interact with the chatbot, submit files, and navigate through the system.

**8. Deployment**

**8.1 Deployment Platform**

Specify the deployment platform chosen, whether it's a website, chatbot platform, or server.

**8.2 Integration**

Explain the integration process, detailing how the chatbot is integrated into the chosen platform or channel.

**9. Security**

**9.1 Secure File Handling**

Discuss the secure file handling practices implemented to protect user-submitted files.

**9.2 Authentication and Authorization**

If applicable, explain the authentication and authorization mechanisms in place.

**10. Continuous Improvement**

**10.1 User Feedback**

Outline strategies for gathering user feedback and incorporating it into future updates.

**10.2 Updates**

Describe the process of updating the chatbot with new Nora framework guidelines or relevant information.

**11. Monitoring and Maintenance**

**11.1 Monitoring**

Explain how monitoring is implemented to track user interactions and identify issues.

**11.2 Maintenance**

Detail the maintenance procedures, including addressing bugs, updates, and improvements.

**12. Conclusion**

Summarize the key points covered in the documentation, expressing gratitude and encouraging collaboration and feedback.