Object Oriented Programming

(Lab)

**Project Proposal Final**

**[AI Mimic Chatbot R.A.Y.N]**

**Student Name: Abdul Basit**

Contents

[Project Declaration 3](#_Toc191674931)

[Title: AI-Powered Chatbot: R.A.Y.N 4](#_Toc191674932)

[Objective: 4](#_Toc191674933)

[Project Overview: 4](#_Toc191674934)

[Program Login 4](#_Toc191674935)

[Project Details: 5](#_Toc191674936)

[Program Flow: 5](#_Toc191674937)

[Key Functionalities: 5](#_Toc191674938)

[Error Handling: 6](#_Toc191674939)

[Program Structure: 6](#_Toc191674940)

[Output Screen: 6](#_Toc191674941)

[Conclusion: 7](#_Toc191674942)

[References: 7](#_Toc191674943)

# ****Project Declaration****

I, Abdul Basit, hereby declare that the project titled "AI-Powered Chatbot: R.A.Y.N," developed as part of the requirements for the **Fundamentals of Object-Oriented Programming (OOP)** course in the **Bachelor of Science in Artificial Intelligence (BSAI)** program, is the result of my individual effort and contribution.

This chatbot system, implemented using the **C++ programming language**, utilizes **object-oriented programming principles** such as **encapsulation, inheritance, and polymorphism** to create an interactive AI assistant.

The chatbot, named **R.A.Y.N (Responsive AI with Neural Processing),** is designed to:

* Simulate human-like conversations.
* Provide basic math assistance via a sub-module called **MathBot**.
* Learn and adapt over multiple interactions.

I affirm that the work presented in this project is entirely original and the result of my own effort. Any external resources utilized during development have been properly acknowledged and referenced.

**Student Name:** Abdul Basit  
**Date:**

**Signature**  
  
  
  
Abdul Basit \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date:** ,

# ****Title: AI-Powered Chatbot: R.A.Y.N****

## **Objective:**

The goal of this project is to develop a chatbot using **C++ and object-oriented programming concepts** to enhance user interaction and automation. The chatbot will:

* Process and respond to user queries.
* Assist with mathematical calculations.
* Maintain a conversational history for improved responses.
* Provide a **real-time AI assistant experience.**

## **Project Overview:**

The chatbot system consists of two primary modules:

1. **R.A.Y.N Chatbot:**
   * Engages in conversation with users.
   * Provides predefined responses based on input.
   * Logs chat history for reference.
2. **MathBot:**
   * Performs basic mathematical operations such as addition, subtraction, multiplication, and division.
   * Includes error handling for invalid inputs and division by zero.

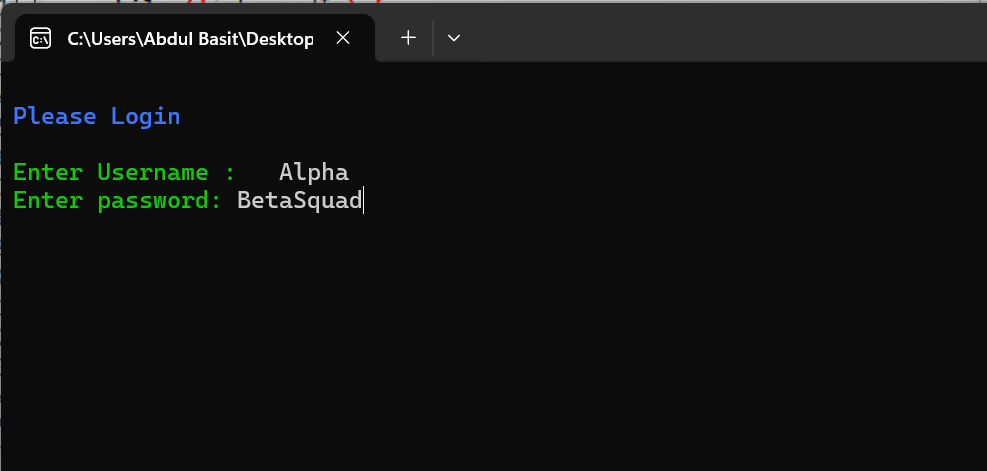
The chatbot is **menu-driven** and allows users to interact with R.A.Y.N or MathBot seamlessly.

## **Program Login**

Since this is an individual project, the system features a **basic login authentication** for administrative access:

|  |  |  |
| --- | --- | --- |
| **Role** | **Username** | **Password** |
| Admin (User) | Alpha | BetaSquad |

The login credentials allow access to additional **debugging and system control functionalities**.



## **Project Details:**

**Programming Language:** C++ 🡪 (OOP)

**Libraries Used:**

• <iostream> – Input and output handling  
• <fstream> – File handling for chat history storage  
• <map> – Associative container for storing chatbot responses  
• <string> – String manipulation and processing  
• <cstdlib> – Randomized response selection and utility functions  
• <ctime> – Time-based operations for response variation  
• <cmath> – Mathematical operations for advanced processing  
• <chrono> – Time-based delays for a realistic chatbot experience  
• <thread> – Implementing pauses and delays for smooth interaction

**Class Structure:**

1. **Chatbot (Base Class)**
   * Virtual function for handling responses
   * Saves chat history
2. **RYANbot (Derived Class)**
   * Implements AI-based responses
   * Loads predefined responses from a file
3. **MathBot (Separate Class)**
   * Performs mathematical calculations
   * Handles errors for invalid operation

## **Program Flow:**

1. **Welcome Screen:** Displays project details.
2. **Main Menu:** Offers interaction with:
   * Chatbot R.A.Y.N
   * MathBot
   * Exit
3. **User Input Handling:** Accepts and processes user queries.
4. **Response Generation:** Uses a predefined response list or generates AI-based responses.
5. **Error Handling:** Handles invalid inputs, unknown queries, and math errors.
6. **Exit Program:** Displays a thank-you message and saves chat history.

## **Key Functionalities:**

**Chatbot Features:**

* Responds to general queries.
* Uses file handling for chat history.
* Implements basic AI-based responses.

**MathBot Features:**

* Performs **addition, subtraction, multiplication, and division**.
* Handles **negative and zero-division errors**.

**Admin Features:**

* Allows debugging and chat history access with an option to reset the system.

## **Error Handling:**

1. **Invalid Input Handling:** If an unknown command is entered, the chatbot provides guidance.
2. **Mathematical Error Handling:**
   * Division by zero is prevented.
   * Negative square roots are flagged.
3. **File Handling Errors:** Checks if the response file exists before accessing.

## **Program Structure:**

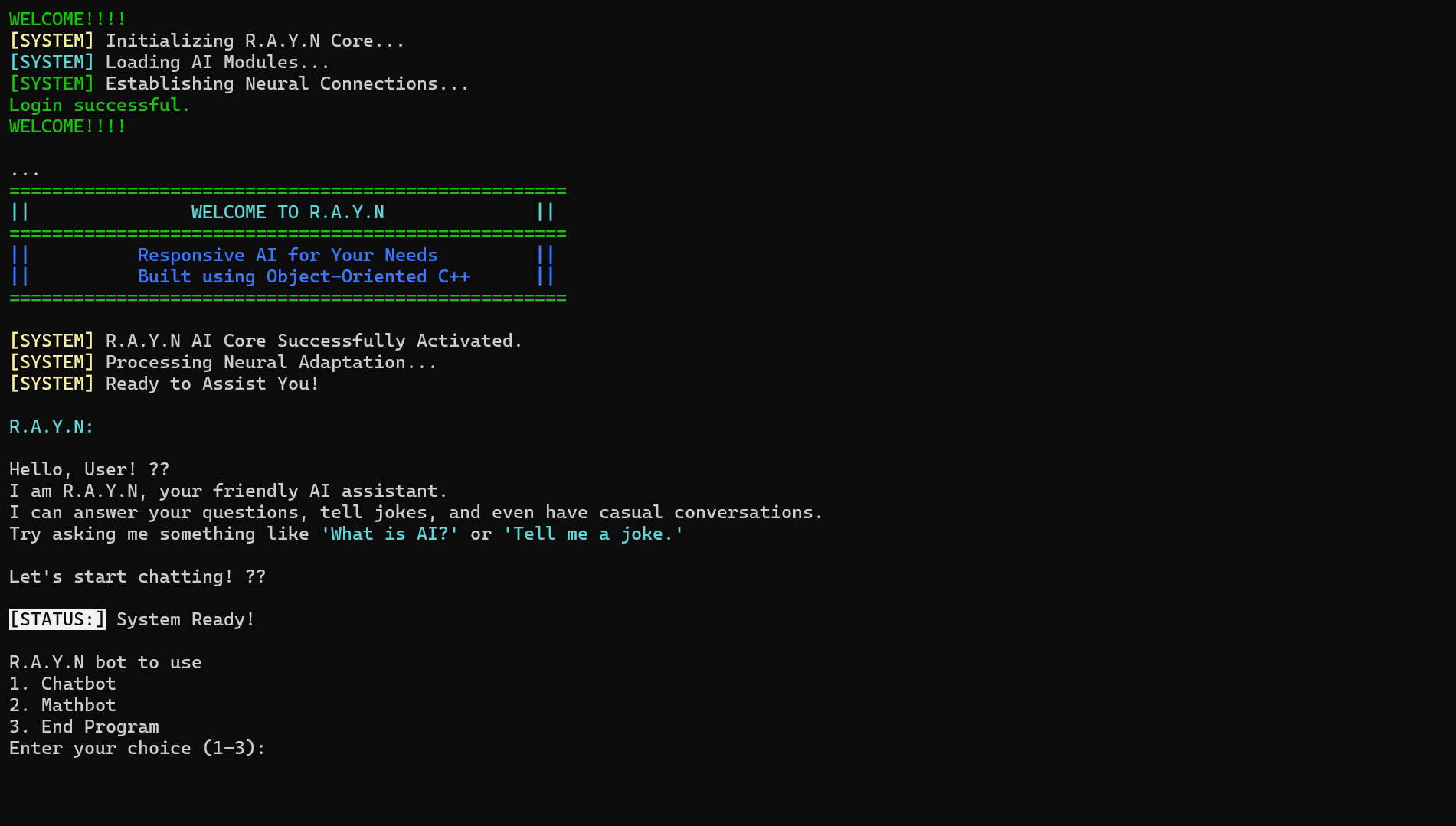
The chatbot uses **OOP principles** to separate logic efficiently:

1. **Encapsulation:** Chatbot, MathBot, and Admin functionalities are contained in separate classes.
2. **Inheritance:** RYANbot inherits from the Chatbot class.
3. **Polymorphism:** Different chatbot responses based on user input.

**Algorithm:**

1. Start the program.
2. Display the welcome screen.
3. Prompt for login (Admin) or direct user to the main menu.
4. Based on user selection:
   * Engage with the chatbot.
   * Perform calculations via MathBot.
5. Loop until the user chooses to exit.
6. Save chat history and terminate.

## **Output Screen:**



## **Conclusion:**

The **AI-Powered Chatbot: R.A.Y.N** effectively demonstrates the implementation of object-oriented programming concepts in C++. It provides interactive user engagement through a conversational chatbot and a mathematical assistant.

The project successfully incorporates:

* **Encapsulation and inheritance** for modular design.
* **File handling** for chat history storage.
* **Dynamic user interaction** with input validation and response generation.

Future improvements may include:

* **NLP integration** for better responses.
* **GUI implementation** for enhanced user experience.
* **Voice interaction** support.

## **References:**

* W3Schools – C++ Tutorial: <https://www.w3schools.com/>
* GeeksforGeeks – C++ Basics: <https://www.geeksforgeeks.org/>
* Stack Overflow – C++ Troubleshooting: <https://stackoverflow.com/>
* C++ Documentation – C++ Reference: <https://cplusplus.com/>

# 