Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY
University with Graded Autonomy Status
(An ISO 21001 : 2018 Certified Institution)
Periyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.

PROJECT REPORT

JAVA PROGRAMMING FUNDAMENTALS (EBDS22ET2)

2024-2025(EVEN SEMESTER)

DEPARTMENT OF B.Tech DS & AI (E&T)

COURSE : B-TECH CSE-DS (AI)

YEAR/SEM/SEC : **I**st year / II^nd sem / AB1

PROJECT TITLE : **Simple Chat BOT (Text based)**

Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY
University with Graded Autonomy Status
(An ISO 21001 : 2018 Certified Institution)
Periyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.

BONAFIDE CERTIFICATE

JAVA PROGRAMMING FUNDAMENTALS

DEPARTMENT OF B.Tech DS & AI (E&T)

Certified that this project report **" SIMPLE CHAT BOT (Text Based) "** is confirmed work of **YUVARAJ SINGH AB1-03 , P.BALAJI AB1-08 , S.NAVEEN AB1-09** I-year B-Tech CSE- DS(AI) in **JAVA PROGRAMMING FUNDAMENTALS (EBDS22ET2)** who carried out the project work under the supervision

Signature of Lab-in-Charge

Signature of Head of Dept

Submitted for the Practical Examination held on_____

Internal Examiner

External Examiner

ABSTRACT

This project presents the implementation of a simple Java-based **console chatbot** that performs multiple predefined tasks based on user input. The chatbot is designed to engage users in basic conversation and respond intelligently to specific commands. It can perform **ten tasks**, including greeting the user, displaying the current time and date, reversing a sentence, calculating the sum of two numbers, determining whether a number is even or odd, telling a joke, and providing motivational quotes. The chatbot uses standard Java libraries such as Scanner for input handling, LocalTime and LocalDate for time/date operations, and Random for generating jokes and quotes. The program operates in a continuous loop, allowing real-time interaction until the user chooses to exit by typing "bye". This chatbot demonstrates core programming concepts such as **control structures**, **string manipulation**, **array handling**, and **user interaction** in Java, making it an excellent beginner-level project for learning interactive Java applications.

INTRODUCTION TO JAVA:

Java is a **high-level, object-oriented programming language** developed by Sun Microsystems (now owned by Oracle Corporation). It was designed with the principle of **"Write Once, Run Anywhere" (WORA)**, which means Java programs can run on any device equipped with the Java Virtual Machine (JVM), regardless of the underlying hardware or operating system.

Java is widely used for building a range of applications — from web and desktop software to mobile apps and embedded systems. Its **strong memory management, multithreading support, platform independence,** and a vast collection of libraries make it a preferred choice for developers.

In this project, Java is used to create a **console-based chatbot**, demonstrating key features of the language such as:

- **Input/output handling** using the Scanner class
- **Conditional and control statements** for decision-making
- **Loops** for continuous user interaction
- **String manipulation** and **array operations**
- **Use of standard libraries** like java.util, java.time, and java.lang

This project illustrates how Java can be employed to build **interactive and user-friendly command-line applications**, reinforcing fundamental programming concepts in a practical and engaging manner.

PROJECT TITLE

AIM:

To design and implement a Java-based chatbot that interacts with users through the console and performs 10 predefined tasks including greeting, showing time/date, reversing text, arithmetic operations, and more**.**

ALGORITHMS:

Step 1: Start the program

- Display a welcome message to the user.

Step 2: Initialize necessary components

- Create a Scanner object to take user input.
- Create arrays for jokes and motivational quotes.
- Set up a Random object to select random jokes/quotes.

Step 3: Show instructions

- Prompt the user to type help to view available commands.

Step 4: Enter a loop to process user input

- Repeat until the user types bye.

Inside the loop:

1. **Read user input**
    - Convert input to lowercase and trim extra spaces
2. **Check the input using a switch-case or if-else**
    - If input is:
        - "hi" or "hello" → Greet the user.
        - "time" → Display the current system time.
        - "date" → Display today's date.
        - "reverse" → Ask for a string and print its reverse.
        - "sum" → Ask for two numbers and print their sum.
        - "even or odd" → Ask for a number and check if it's even or odd.
        - "joke" → Display a random joke from the array.
        - "quote" → Display a random motivational quote.
        - "help" → Show the list of available commands.
        - "bye" → Display exit message and break the loop.
    - Else → Print an error message for unrecognized input.

Step 5: End the program

- Close the Scanner object.
- Display a final goodbye message.

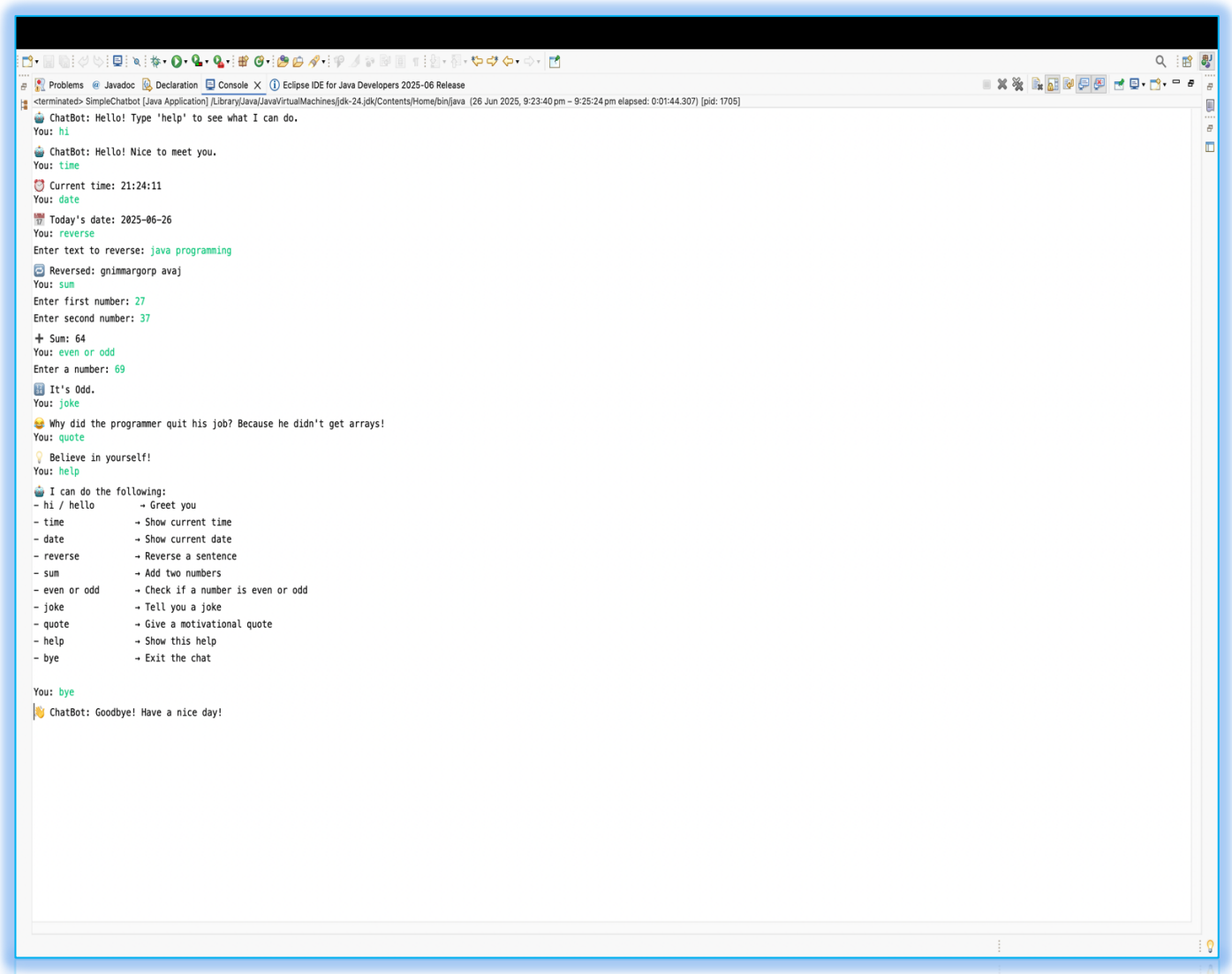
SOURCE CODE:

```
package yuvi.java;


import java.util.*;
import java.time.LocalDate;
import java.time.LocalTime;

public class  SimpleChatbot{

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      Random rand = new Random();
      String userInput;

      String[] jokes = {
         "Why don't scientists trust atoms? Because they make up everything!",
         "Why did the programmer quit his job? Because he didn't get arrays!",
         "I would tell you a construction pun, but I'm still working on it."
      };

      String[] quotes = {
         "Believe in yourself!",
         "Every day is a second chance.",
         "Push yourself, because no one else is going to do it for you."
      };

      System.out.println("🤖 ChatBot: Hello! Type 'help' to see what I can do.");

      while (true) {
         System.out.print("You: ");
         userInput = scanner.nextLine().toLowerCase().trim();

         switch (userInput) {
            case "hi":
            case "hello":
               System.out.println("🤖 ChatBot: Hello! Nice to meet you.");
               break;
```

```
            case "time":
                System.out.println("⏰ Current time: " + LocalTime.now().withNano(0));
                break;

            case "date":
                System.out.println("📅 Today's date: " + LocalDate.now());
                break;

            case "reverse":
                System.out.print("Enter text to reverse: ");
                String toReverse = scanner.nextLine();
                String reversed = new StringBuilder(toReverse).reverse().toString();
                System.out.println("🔁 Reversed: " + reversed);
                break;

            case "sum":
                System.out.print("Enter first number: ");
                int num1 = scanner.nextInt();
                System.out.print("Enter second number: ");
                int num2 = scanner.nextInt();
                scanner.nextLine(); // consume leftover newline
                System.out.println("➕ Sum: " + (num1 + num2));
                break;

            case "even or odd":
                System.out.print("Enter a number: ");
                int number = scanner.nextInt();
                scanner.nextLine(); // consume newline
                System.out.println(number % 2 == 0 ? "🔢 It's Even." : "🔢 It's Odd.");
                break;

            case "joke":
                System.out.println("😂 " + jokes[rand.nextInt(jokes.length)]);
                break;

            case "quote":
                System.out.println(" 💡 " + quotes[rand.nextInt(quotes.length)]);
                break;

            case "help":
              System.out.println("""
🤖 I can do the following:
- hi / hello       → Greet you
- time            → Show current time
- date            → Show current date
- reverse         → Reverse a sentence
- sum             → Add two numbers
- even or odd     → Check if a number is even or odd
- joke            → Tell you a joke
- quote           → Give a motivational quote
```

```
- help          → Show this help
- bye           → Exit the chat
""");
                break;

            case "bye":
                System.out.println("👋ChatBot: Goodbye! Have a nice day!");
                return;

            default:
                System.out.println(" ❓ I don't understand that. Type 'help' to see what I can do.");
        }
    }
  }
}
```

OUTPUT:


Problems
Javadoc
Declaration
Console
Eclipse IDE for Java Developers 2025-06 Release
<terminated> SimpleChatbot [Java Application] /Library/Java/JavaVirtualMachines/jdk-24.jdk/Contents/Home/bin/java (26 Jun 2025, 9:23:40 pm – 9:25:24 pm elapsed: 0:01:44.307) [pid: 1705]
ChatBot: Hello! Type 'help' to see what I can do.
You: hi
ChatBot: Hello! Nice to meet you.
You: time
Current time: 21:24:11
You: date
Today's date: 2025-06-26
You: reverse
Enter text to reverse: java programming
Reversed: gnimmargorp avaj
You: sum
Enter first number: 27
Enter second number: 37
Sum: 64
You: even or odd
Enter a number: 69
It's Odd.
You: joke
Why did the programmer quit his job? Because he didn't get arrays!
You: quote
Believe in yourself!
You: help
I can do the following:
- hi / hello → Greet you
- time → Show current time
- date → Show current date
- reverse → Reverse a sentence
- sum → Add two numbers
- even or odd → Check if a number is even or odd
- joke → Tell you a joke
- quote → Give a motivational quote
- help → Show this help
- bye → Exit the chat
You: bye
ChatBot: Goodbye! Have a nice day!

RESULT:

✅ Result

The Java-based chatbot was successfully developed and executed as a **console application**. The chatbot is capable of understanding and responding to user inputs based on **10 predefined commands**. It performs tasks such as:

- Greeting the user
- Displaying the current time and date
- Reversing a given string
- Performing addition of two numbers
- Checking if a number is even or odd
- Telling a random joke
- Displaying a motivational quote
- Showing a help menu with command options
- Exiting the program on user request

The chatbot operates in a continuous loop, providing real-time interaction until the user types "bye" to terminate the session. All features were tested and functioned correctly, providing accurate outputs for valid inputs and appropriate error messages for unknown commands.

Thus, the implementation demonstrates how Java can be effectively used to build a functional, interactive, and user-friendly console application using core programming concepts.
