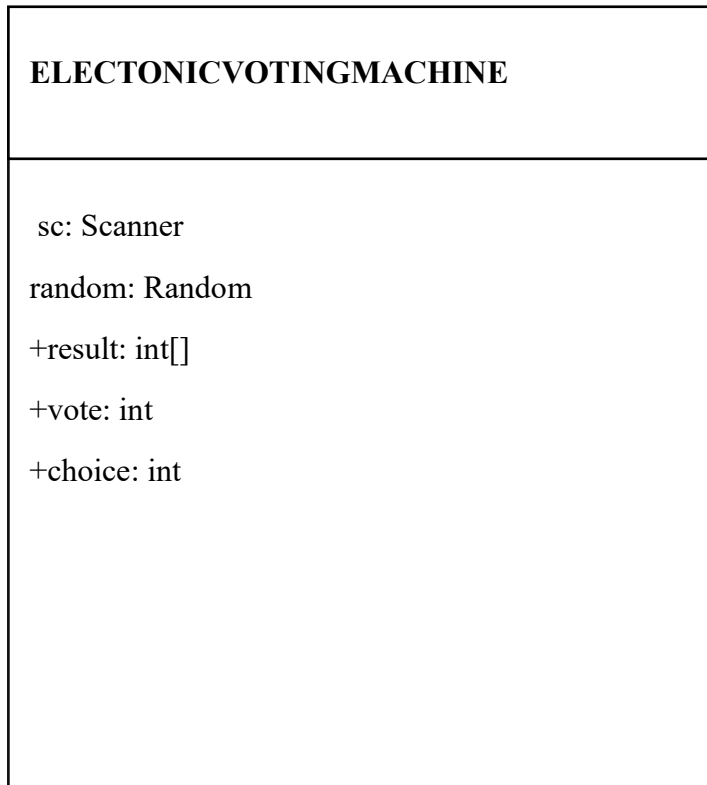
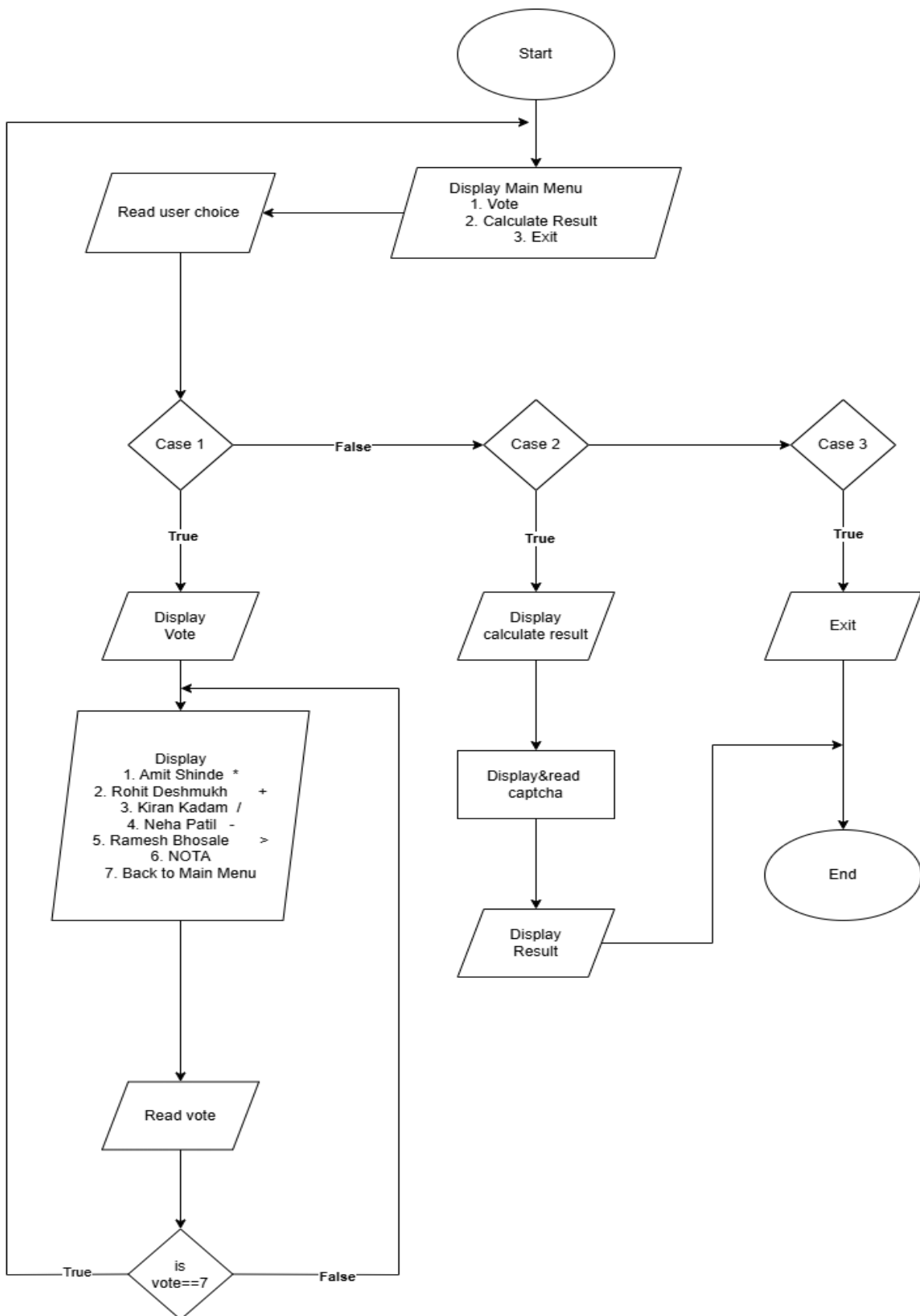


1.UML DIAGRAM



2.Flowchart



3.Code of EVM

```
import java.util.Random;
import java.util.Scanner;

public class EVM {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Random random = new Random();
        int[] result = new int[6]; // Adjusted to 6 to match the 6 voting options
        int vote, choice;

        do {
            System.out.println("---- Parliament Election 24-25 ----");

            System.out.println("Enter choice : ");
            System.out.println("1. Vote\n2. Calculate Result\n3. Exit");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    do {
                        System.out.println("---- Parliament Election Candidate list 24-25 ----");
                        System.out.println("1. Amit Shinde\t*\n2. Rohit Deshmukh\t+\n3. Kiran Kadam\t/\n4. Neha Patil\t-\n5. Ramesh Bhosale\t>\n6. NOTA\n7. Back to Main Menu");
                        vote = sc.nextInt();

                        if (vote > 0 && vote <= 6) {
                            result[vote - 1]++;
                            System.out.println("Vote recorded for choice " + vote);
                        } else if (vote == 7) {
                            break;
                        } else {
                            System.out.println("Invalid vote. Please try again.");
                        }
                    } while (vote != 7);
                    break;

                case 2:
                    int systemCaptcha = random.nextInt(900000) + 100000; // 6-digit captcha
                    System.out.println("Enter captcha: " + systemCaptcha);
                    int captcha = sc.nextInt();
                    if (captcha == systemCaptcha) {
                        System.out.println("Calculating results...");
                        // Display results sorted
                        for (int i = 0; i < result.length; i++) {
                            System.out.println("Candidate " + (i + 1) + ": " + result[i] + " votes");
                        }
                    }
            }
        }
    }
}
```

```

        choice=3;// Set choice to 3 to exit after displaying results
    } else {
        System.out.println("Incorrect captcha. Returning to main menu.");
    }
    break;

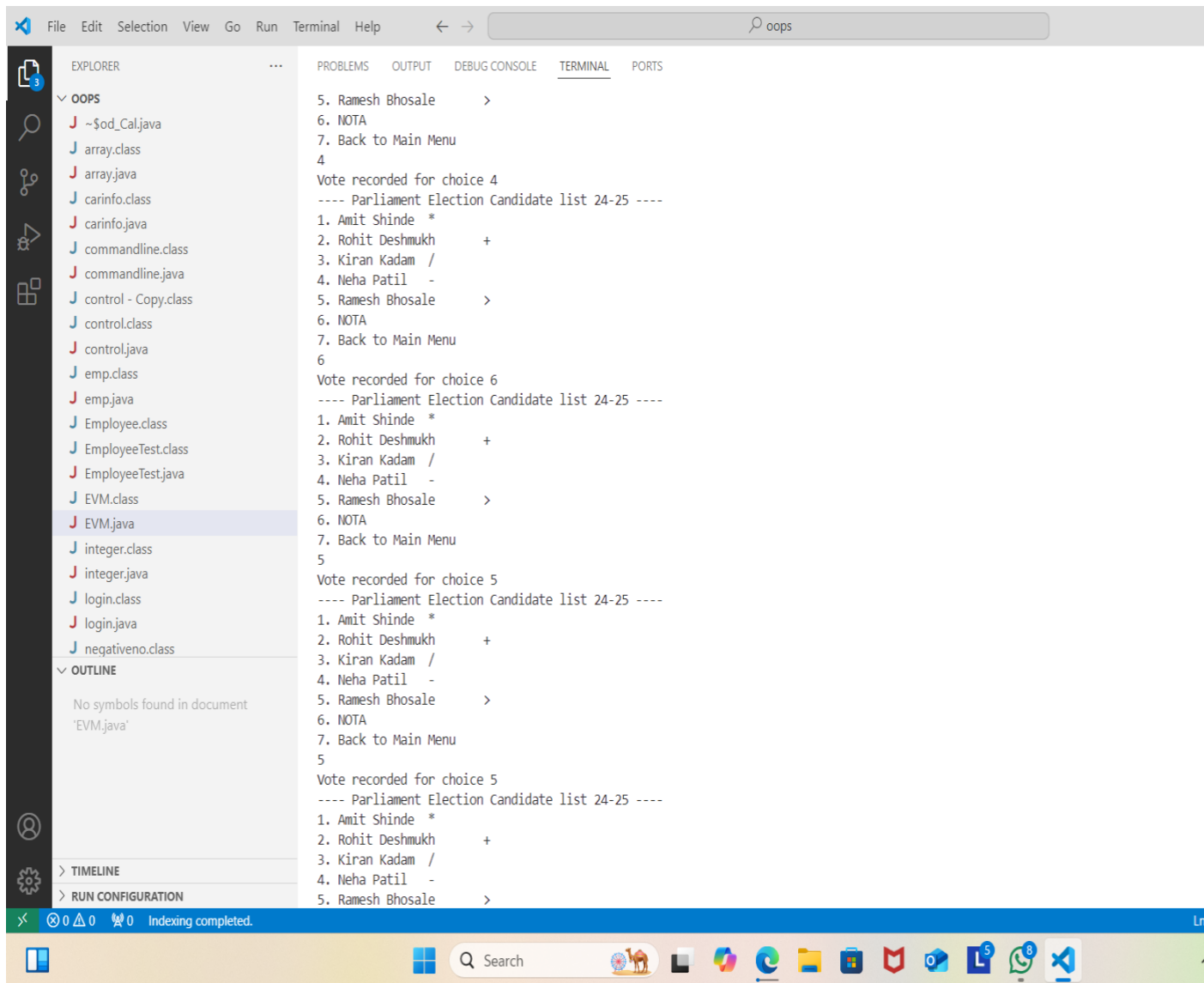
case 3:
    System.out.println("Exiting...");
    break;

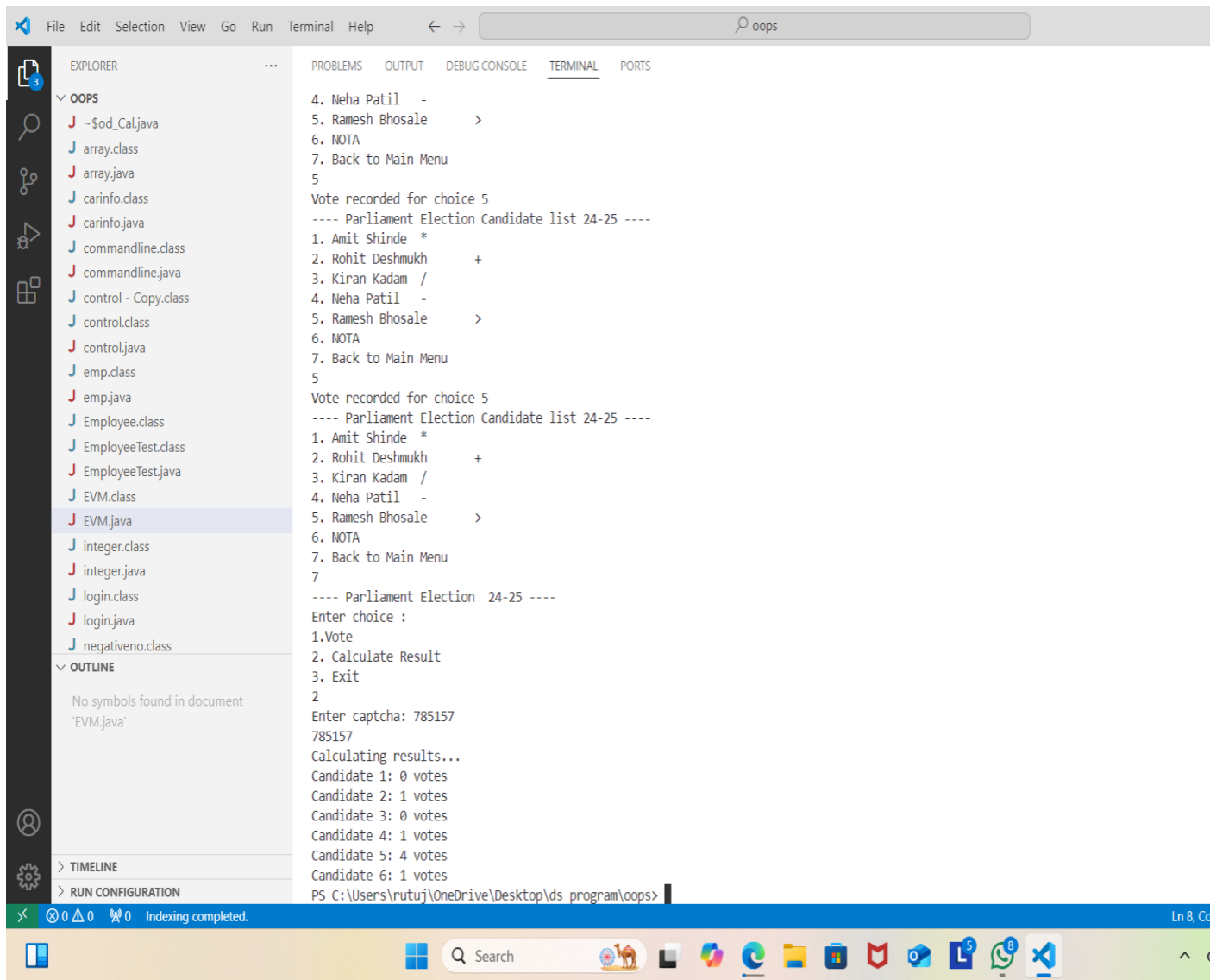
default:
    System.out.println("Invalid choice. Please select again.");
    break;
}
} while (choice != 3);
}
}

```

4.OUTPUT

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a file tree for a project named "oops". The files include source code (.java) and compiled class files (.class). The file "EVM.java" is selected. Below the Explorer, the Outline sidebar shows "No symbols found in document 'EVM.java'". At the bottom of the sidebar are sections for Timeline and Run Configuration. The main editor area has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the execution of a Java program. The program prompts for a choice between 1 (Vote), 2 (Calculate Result), and 3 (Exit). Choice 1 is entered, leading to a list of candidates for Parliament Election 24-25. Choice 5 is entered, resulting in a recorded vote for candidate 5. This sequence repeats once more.





5.EXPLANATION OF CODE

1. Import Statements:

```
import java.util.Random;
```

```
import java.util.Scanner;
```

- Random: This class is used to generate a random number for the captcha validation.
- Scanner: This is used to take input from the user.
- Scanner: This is used to take input from the user.

2. Main Class and Method:

```
public class EVM {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        Random random = new Random();
```

```
        int[] result = new int[6]; // Array to store votes for 6 candidates
```

```
        int vote, choice;
```

- Scanner sc = new Scanner(System.in);: Creates a Scanner object to read user input.
- Random random = new Random();: Creates a Random object to generate random numbers for captcha.
- int[] result = new int[6];: Declares an array to store the number of votes for each of the 6 candidates. Each index corresponds to a specific candidate.
- int vote, choice;: Declares two integer variables:
 - vote: Stores the user's vote.
 - choice: Stores the user's menu selection.

3. Main Loop:

```
    do {
```

```
        System.out.println("---- Parliament Election 24-25 ----");
```

```
        System.out.println("Enter choice : ");
```

```
        System.out.println("1. Vote\n2. Calculate Result\n3. Exit");
```

```
        choice = sc.nextInt();
```


- The program displays the main menu with three options: Voting, Calculate Result, and Exit.
- The user is prompted to enter a choice, which is stored in choice.

4. Switch-Case for Menu Choices:

The program uses a switch statement to handle different menu choices based on the value of choice.

4.1 Case 1: Voting:

java

Copy code

```

        case 1:
            do {
                System.out.println("---- Parliament Election Candidate list 24-25 ----");
                System.out.println("1. Amit Shinde\t*\n2. Rohit Deshmukh\t+\n3. Kiran Kadam\t/\n4. Neha Patil\t-\n5. Ramesh Bhosale\t>\n6. NOTA\n7. Back to Main Menu");
                vote = sc.nextInt();

```

- The user is shown a list of candidates and given the option to vote for one of the candidates or "NOTA" (None of the Above). There is also an option to go back to the main menu.
- The vote variable stores the user's selection (1 to 7).

```

        if (vote > 0 && vote <= 6) {
            result[vote - 1]++; // Increment the vote count for the selected candidate
            System.out.println("Vote recorded for choice " + vote);
        } else if (vote == 7) {
            break; // Exit the voting loop and go back to the main menu
        } else {
            System.out.println("Invalid vote. Please try again.");
        }
    } while (vote != 7); // Keep prompting the user until they choose option 7 (Back to Main Menu)

    break;

```

- If the user selects a valid candidate (1 to 6), the vote count for that candidate is incremented in the result array.

- If the user selects 7, the program breaks out of the voting loop and returns to the main menu.
- If the input is invalid (e.g., a number outside the valid range), the user is prompted to try again.

4.2 Case 2: Calculate Results:

java

Copy code

case 2:

```
int systemCaptcha = random.nextInt(900000) + 100000; // Generate a 6-digit captcha
System.out.println("Enter captcha: " + systemCaptcha);
int captcha = sc.nextInt();
```

- A random 6-digit captcha is generated using `random.nextInt(900000) + 100000`, which ensures the captcha is always a 6-digit number.
- The user is prompted to enter the captcha.

```
if (captcha == systemCaptcha) {
    System.out.println("Calculating results...");
    // Display results sorted
    for (int i = 0; i < result.length; i++) {
        System.out.println("Candidate " + (i + 1) + ": " + result[i] + " votes");
    }
    choice = 3; // Exit the program after displaying the results
} else {
    System.out.println("Incorrect captcha. Returning to main menu.");
}
break;
```

- If the user enters the correct captcha, the program calculates and displays the vote counts for each candidate by iterating over the result array.
- If the captcha is incorrect, the program informs the user and returns to the main menu.
- After displaying the results, `choice = 3` is set, which will exit the program.

4.3 Case 3: Exit:

case 3:

```
System.out.println("Exiting...");
```

```
break;
```

- If the user selects option 3, the program prints "Exiting..." and exits the loop, ending the program.

5. Invalid Choice Handling:

default:

```
System.out.println("Invalid choice. Please select again.");
```

```
break;
```

- If the user enters an invalid menu choice (not 1, 2, or 3), the program prints an error message and prompts the user to select again.

6. Main Loop Continuation:

```
} while (choice != 3); // Loop continues until the user selects option 3 (Exit)
```

- The do-while loop ensures the program continues to run until the user selects option 3 (Exit).

CONCLUSION

This program simulates a basic Electronic Voting Machine (EVM) system for a parliamentary election. The code effectively implements the core functionalities required for such a system, including voting, result calculation, and user authentication through captcha validation.

While this is a basic prototype, the program demonstrates the core functionality of an electronic voting system and can be easily extended with additional features, such as user authentication, persistent storage of results, or the ability to handle multiple elections. The program serves as a foundational model for understanding the operation of an EVM system, and it provides a useful starting point for building more sophisticated voting applications.

In terms of improvement, future iterations could involve adding more security features (e.g., encrypted captcha or multi-factor authentication), implementing a graphical user interface (GUI), or allowing for the storage and retrieval of election data from a database for long-term record-keeping.