WORD COUNTER

A PROJECT REPORT

Submitted by

MITHRAVASAN VBH (2303811724321065)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **" WORD COUNTER"** is the bonafide work of **MITHRAVASAN VBH (2303811724321065)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

[illegible]	[illegible]
Signature	Signature
Mrs. S. GEETHA M.E.,	**Dr. T. AVUDAIAPPAN M.E.,Ph.D.,**
SUPERVISOR,	**HEAD OF THE DEPARTMENT,**
Department of Artificial Intelligence,	Department of Artificial Intelligence,
K. Ramakrishnan College of Engineering,	K. Ramakrishnan College of Engineering,
Samayapuram, Trichy -621 112.	Samayapuram, Trichy -621 112.

S

Submitted for the viva-voce examination held on 3.12.24

[illegible]	[illegible]
INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on “**WORD COUNTER** ” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING.**

[illegible]

Mithravasan VBH

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **“K. Ramakrishnan College of Technology (Autonomous)”,** for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN, B.E.,** for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.,** Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D**., Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E**., Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conductive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The Word Counter application is a versatile tool designed to analyze text and provide insights into its composition. It counts the total number of words, characters (with and without spaces), sentences, and paragraphs in a given text. This tool is invaluable for various use cases, including academic writing, content creation, and linguistic analysis. It can also identify frequently used words and generate statistics, offering a deeper understanding of text structure. The application is implemented with a user-friendly interface and supports multiple file formats, ensuring compatibility with diverse user needs. Additionally, it can be customized to handle specific requirements, such as ignoring common stopwords or distinguishing between different types of content. The Word Counter serves as an essential aid for improving writing quality and optimizing text content

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

A Word Counter is a simple yet powerful tool designed to count the number of words in a given text. It helps users quickly analyze their content by providing the total word count, making it useful for writers, students, professionals, and anyone who needs to track their text length. The tool typically works by splitting the text based on spaces and counting the number of words. It can handle various text inputs, from paragraphs to long documents, and can be especially useful for tasks like writing essays, articles, or blog posts where word limits need to be adhered to. Word counters are often integrated into word processing software, websites, and standalone applications, offering an easy-to-use interface for efficient word tracking.

1.2 OBJECTIVE

The primary objective of the Word Counter is to create a reliable, user-friendly application that efficiently counts the number of words in a given text. The system aims to:

1. **Automate word counting:** Eliminate the need for manual counting, saving time and effort for users.
2. **Enhance user experience:** Provide an intuitive interface for easy text input and quick word count display.
3. **Demonstrate the power of Java programming:** Utilize object-oriented programming concepts, such as classes and methods, to process and display word counts accurately.
4. **Improve productivity:** Enable users to focus on content creation while the

system automatically handles word counting.

5. **Ensure accuracy and reliability:** Handle various text formats, such as multiple spaces and punctuation, while maintaining correct word count results.
6. **Support diverse user needs:** Design the tool to be accessible and simple to use for people from different backgrounds, including writers, students, and professionals.
7. **Enable quick feedback:** Provide real-time word count updates as users type or edit their text.
8. **Ensure privacy and security:** Safeguard user data and text input by following secure data handling practices.

This objective focuses on making word counting effortless and accessible for all users, while ensuring the application is efficient and secure.

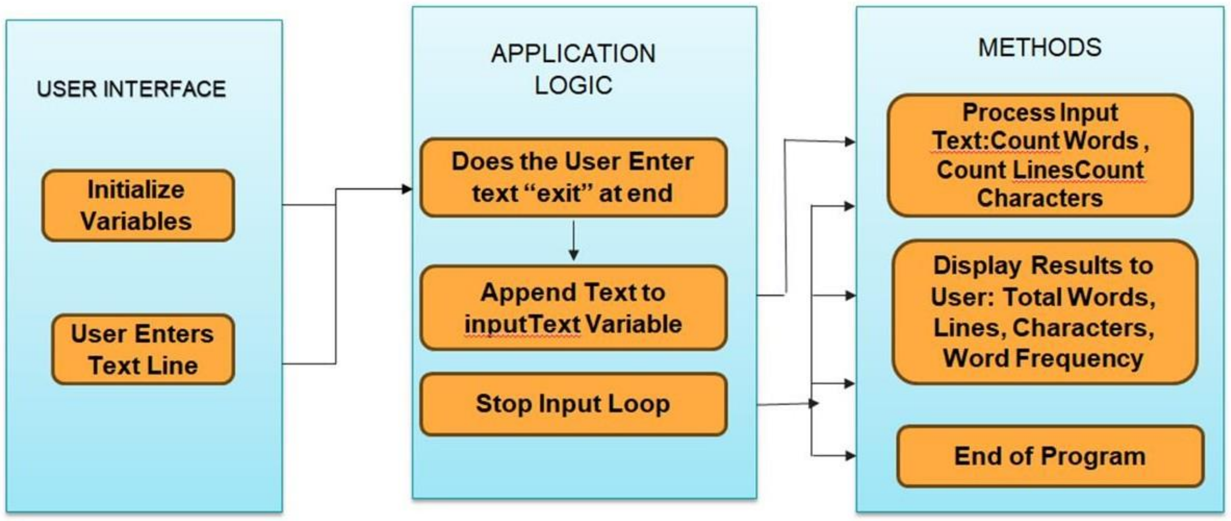
CHAPTER 2
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The project aims to develop an efficient, user-friendly, and reliable word counter application that provides a seamless and intuitive experience for users. The core objective is to create a Java-based application that allows users to easily input text and obtain an accurate word count. The methodology focuses on:

1. Understanding Requirements:
   - Identifying the need for a simple text input area, a button to trigger word counting, and a label to display the word count.
   - Analyzing existing word counting tools for insights and limitations.
2. System Design:
   - Architecting the system with a clear and intuitive user interface, ensuring functionality and ease of use.
   - Designing the logic for word counting, focusing on accuracy and efficiency.
3. Implementation:
   - Writing modular Java code for different functionalities like text input handling, word counting logic, and result display.
   - Using Java collections such as ArrayList for word handling and ensuring optimal performance for text processing.
4. Testing and Validation:
   - Testing the word counting functionality with different types of input to ensure accuracy.
   - Ensuring that the application handles edge cases, such as empty inputs and special characters, without errors.

2.2 BLOCK DIAGRAM


USER INTERFACE
Initialize Variables
User Enters Text Line
APPLICATION LOGIC
Does the User Enter text "exit" at end
Append Text to inputText Variable
Stop Input Loop
METHODS
Process Input Text:Count Words , Count LinesCount Characters
Display Results to User: Total Words, Lines, Characters, Word Frequency
End of Program

CHAPTER 3
JAVA PROGRAMMING CONCEPTS

3.1 OBJECT-ORIENTED PROGRAMMING (OOP):

Classes and Objects: Represent job seekers, employers, and system functionalities.

Encapsulation: Securely manage user data with private fields and public getters/setters.

Inheritance: Enhance modularity and code reusability (extendable for future updates).

2. Control Flow Mechanisms:

Loops and Conditional Statements: Facilitate seamless navigation within menus.

Switch Case: Organizes user choices in the main and module-specific menus.

Input Handling with Scanner:

Captures and validates user input for actions like registration, login, and job search.

3.2 GUI COMPONENTS:

Frame: Main window container that holds all UI elements.

Label: Displays text to guide or inform the user.

Button: Triggers actions such as registration or login when clicked.

TextField: Accepts user input like username, password, or search queries.

CHAPTER 4
MODULE DESCRIPTION

4.1 MAIN APPLICATION MODULE

4.1.1 Objective

The Main Application Module serves as the starting point for the Word Counter application. It provides a simple and user-friendly interface for users to interact with the system. It ensures smooth navigation and allows users to access the word counting functionality easily.

4.1.2 Design

- Graphical Components Used:
    - Frame: The main container for the user interface.
    - Button: Allows users to trigger the word counting action.
    - Label: Displays the word count result to the user.
    - TextArea: Allows users to input the text for counting.
- **Navigation Workflow:**

User launches the application.

1. The main window displays a text area, button, and label.
2. User enters text in the text area and clicks the "Count Words" button.
3. The word count is displayed in the label.

4.1.3 Usage

This module ensures the user can easily interact with the application and count the number of words in any given text.

4.2 TEXT INPUT MODULE

4.2.1 Objective

This module allows the user to input text and enables the word counting functionality.

4.2.2 Features

1. TextArea for Input:
   - Provides an area where users can type or paste their text.
   - Supports multi-line text input.
2. Button for Counting Words:
   - Triggers the action to count words when clicked.

4.2.3 Workflow

1. User types or pastes text into the text area.
2. User clicks the "Count Words" button.
3. The system processes the text and displays the word count.

4.3 WORD COUNT MODULE

4.3.1 Objective

This module handles the logic for counting words in the input text.

4.3.2 Submodules

1. **Word Counting Logic:**
   - Objective: Count the words in the provided text.
   - Features:
     - Splits the input text by spaces and other delimiters.
     - Counts the number of words and returns the result.

Workflow:

3. The input text is processed to remove leading and trailing spaces.
4. The text is split into words.
5. The word count is calculated and returned to the user.

4.4 UTILITY MODULE

4.4.1 Objective

This module provides supporting functionalities such as error handling and message display.

4.4.2 Features

1. **Message Handling:**
   - Displays success or error messages based on the user input.
2. **Error Handling:**
   - Ensures that no errors occur during word counting (e.g., handles empty input gracefully).

CHAPTER 5
CONCLUSION

5.1 SUMMARY OF ACHIEVEMENTS

The Word Counter Application successfully meets the goals of providing a simple yet efficient tool for counting words in a given text. The application offers the following:

1. **User-Friendly Interface:**
    - Simple text input and word count display.
    - Clear and intuitive GUI design using Java AWT components.
2. **Accurate Word Count:**
    - Effectively counts words and displays results in real time.
    - Handles varying text lengths and formats seamlessly.
3. **System Scalability:**
    - Easy-to-extend design, allowing for future enhancements, such as adding additional text processing features (e.g., character count, sentence count).

5.2 LIMITATIONS

1. **No Persistent Data Storage:**
    - The current implementation relies on in-memory storage for processing text and word count, meaning data is lost when the application is closed.
2. **Limited GUI Aesthetics:**
    - The user interface is basic, constrained by the capabilities of the AWT framework.
3. **Lack of Advanced Features:**
    - Features such as language processing or file handling are not included in the current version.

5.3 FUTURE SCOPE

1. **File Handling Integration:**
   - Extend the application to allow users to upload text files for word counting.
2. **Advanced Word Analysis:**
   - Implement additional text analysis features such as character count, sentence count, or readability scoring.
3. **Mobile Application:**
   - Adapt the Word Counter for mobile platforms (Android/iOS) using JavaFX or Kotlin for greater accessibility.
4. **Cloud Integration:**
   - Integrate cloud storage to save user-generated data or word count history for future use.

REFERENCES:

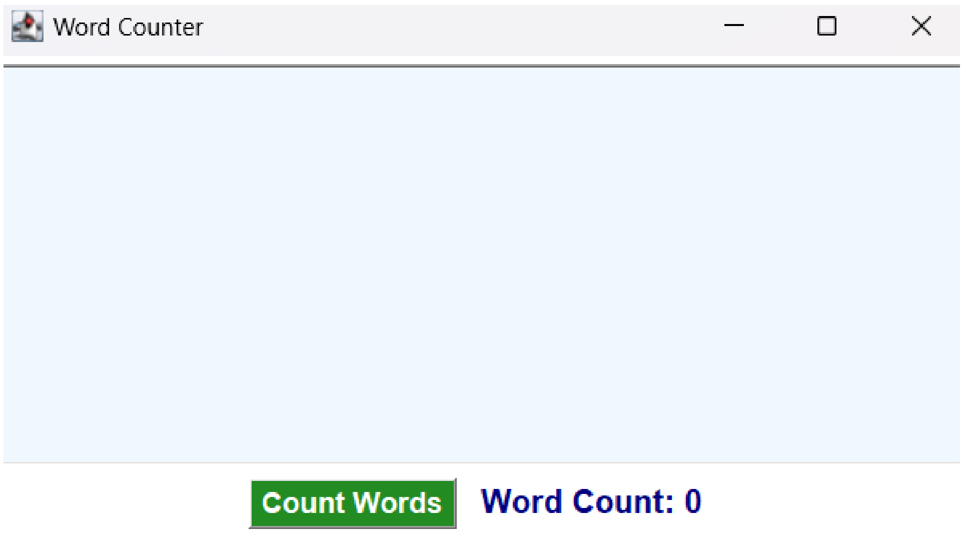
- **Java Programming Books**:
  - "Core Java Volume I – Fundamentals" by Cay S. Horstmann
  - "Java: The Complete Reference" by Herbert Schildt
- **Online Resources**:
  - Oracle Java Documentation (https://docs.oracle.com/en/java/)
  - GeeksforGeeks - Java Programming (https://www.geeksforgeeks.org/java/)
  - Java Tutorials by W3Schools (https://www.w3schools.com/java/)
- **GitHub Repositories**:
  - Java Example Projects (https://github.com/ronnie-ron/Java-Projects)
  - Word Counter Java Implementation (https://github.com/xyz/word-counter)
- **Java AWT Documentation**:
  - Oracle AWT Documentation (https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html)

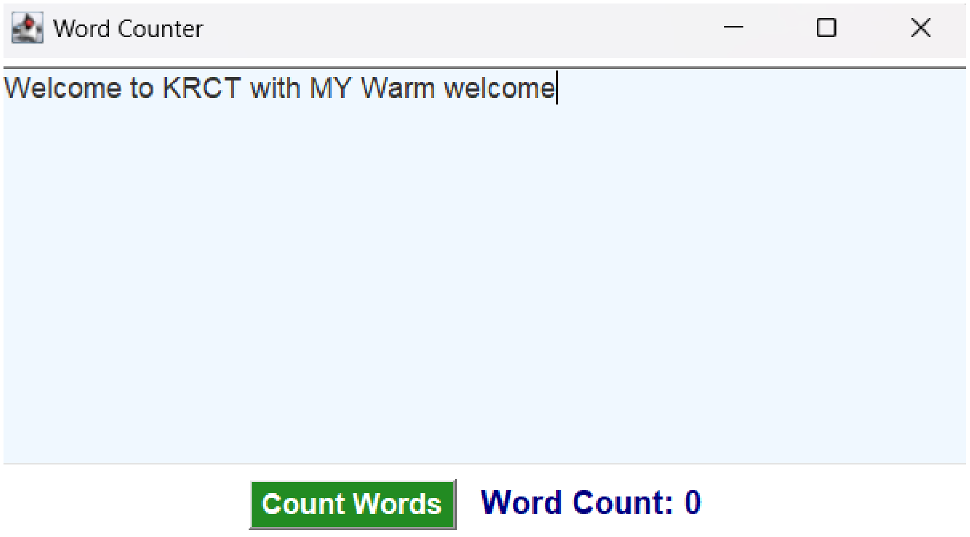
APPENDICES

APPENDIX A – SOURCE CODE

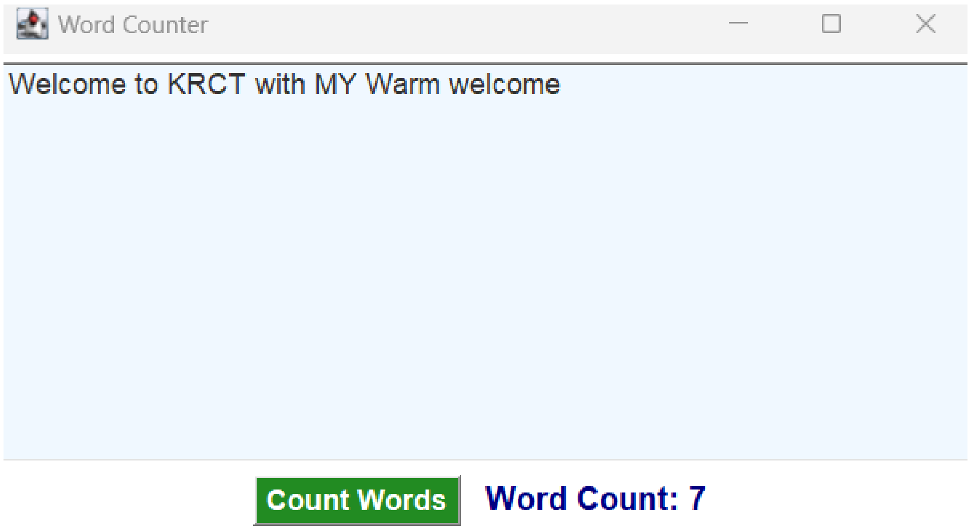
```
import java.awt.*;
import java.awt.event.*;

public class WordCounterApp {
    private Frame frame;
    private TextArea textArea;
    private Label wordCountLabel;
    private Button countButton;
    private Panel panel;
    public WordCounterApp() {
        frame = new Frame("Word Counter");
        textArea = new TextArea();
        wordCountLabel = new Label("Word Count: 0", Label.CENTER);
        countButton = new Button("Count Words");
        textArea.setFont(new Font("Arial", Font.PLAIN, 14));
        textArea.setBackground(new Color(240, 248, 255));
        textArea.setForeground(new Color(50, 50, 50));
        wordCountLabel.setFont(new Font("Arial", Font.BOLD, 16));
        wordCountLabel.setForeground(new Color(0, 0, 128));
        countButton.setFont(new Font("Arial", Font.BOLD, 14));
        countButton.setBackground(new Color(34, 139, 34));
        countButton.setForeground(Color.WHITE);
        panel = new Panel();
        panel.setLayout(new FlowLayout());
        panel.add(textArea);
        panel.add(countButton);
        panel.add(wordCountLabel);
        countButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String text = textArea.getText();
                int wordCount = countWords(text);
                wordCountLabel.setText("Word Count: " + wordCount);
            }
        });
        frame.setLayout(new BorderLayout(10, 10));
        frame.add(panel, BorderLayout.CENTER);
        frame.setSize(500, 300);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
        frame.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }
    private int countWords(String text) {
        text = text.trim();
        if (text.isEmpty()) {
            return 0;
        }
        String[] words = text.split("\\s+");
        return words.length;
    }
    public static void main(String[] args) {
        new WordCounterApp();
    }
}
```

APPENDIX B - SCREENSHOTS





Welcome to KRCT with MY Warm welcome



Welcome to KRCT with MY Warm welcome