RESUME BUILDER (CONSOLE-BASED JAVA MINI PROJECT)

A PROJECT REPORT

Submitted by

Bharat Bhatti (23BCS12220)

Anshraj Rai (23BCS11664)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

Chandigarh University

Nov-2025

ABSTRACT

The "Resume Builder" is a Java-based console application designed to help users create a structured and formatted resume automatically. The system collects essential user details such as name, contact information, education, skills, and experience. Using Java's File Handling mechanism, it generates two files: one in plain text (.txt) and another in XML (.xml) format. This project focuses on core Java concepts such as user input handling, data organization, file writing, and exception management. The application demonstrates simplicity, functionality, and reliability, making it a great learning tool for beginners exploring Java file handling.

CHAPTER 1: INTRODUCTION

1.1 Identification of Need:

A resume is an essential document for every student and professional. Many users struggle with formatting or creating resumes manually. To simplify this process, a console-based Resume Builder has been developed using Java. It automates resume creation and demonstrates file handling techniques.

1.2 Problem Identification:

Users often need a quick way to generate resumes without manually editing templates. Existing systems are web-based or GUI-heavy. Hence, a lightweight console-based program using Java was chosen for simplicity and educational clarity.

1.3 Objectives:

- To take user input and generate a formatted resume.
- To store data in both text and XML formats.
- To demonstrate Java file handling and exception control.

1.4 Task Identification:

- Collect user input (name, email, phone, education, skills, experience).
- Write data to resume.txt (formatted resume).
- Write structured data to resume.xml (hierarchical storage).
- Handle errors gracefully using try-catch blocks.

1.5 Project Timeline:

Phase 1: Requirement & Design (Week 1-2)

Phase 2: Implementation (Week 3-4)

Phase 3: Testing & Validation (Week 5)

Phase 4: Documentation (Week 6)

1.6 Organization of Report:

Chapter 1 - Introduction

Chapter 2 - System Design and Methodology

Chapter 3 - Results and Validation

Chapter 4 - Conclusion and Future Work

CHAPTER 2: SYSTEM DESIGN AND METHODOLOGY

2.1 System Architecture:

The application follows a simple linear flow:

User Input -> Data Processing -> File Generation -> Confirmation Message

2.2 Design Flow:

- Step 1: Prompt the user for input.
- Step 2: Store inputs in variables.
- Step 3: Use FileWriter to create 'resume.txt' and 'resume.xml'.
- Step 4: Write formatted content.
- Step 5: Display success message.

2.3 Algorithm Description:

1. Start the program.
2. Collect user inputs using Scanner.
3. Create and open two files (resume.txt, resume.xml).
4. Write the details in both files with formatting.
5. Close files and display confirmation.

2.4 Technology Stack:

- Programming Language: Java 17
- IDE: Visual Studio Code
- Core Packages: java.io, java.util
- Operating System: macOS

2.5 Advantages:

- Lightweight console application.
- Works on any OS with Java installed.
- Demonstrates core Java concepts clearly.
- Easy to extend for GUI or web-based version.

CHAPTER 3: RESULTS, ANALYSIS AND VALIDATION

3.1 Implementation Overview:

The program successfully collects data and writes two files to disk: resume.txt and resume.xml.

3.2 Output Example (resume.txt):

========== RESUME ==========

Name: Anshraj Rai

Email: anshraj@example.com

Phone: 9876543210

----- Education -----

B.Tech in Computer Science

----- Skills -----

Java, HTML, CSS, SQL

----- Experience / Projects -----

Mini Project on Resume Builder

==============================

3.3 Output Example (resume.xml):

```
<resume>
  <name>Anshraj Rai</name>
  <email>anshraj@example.com</email>
  <phone>9876543210</phone>
  <education>B.Tech in Computer Science</education>
  <skills>Java, HTML, CSS, SQL</skills>
  <experience>Mini Project on Resume Builder</experience>
</resume>
```

3.4 Validation:

- File creation and writing were tested successfully.
- Inputs and outputs were verified for accuracy.
- Error handling tested using invalid inputs.

3.5 Result:

The project fulfills its objectives and demonstrates Java file handling effectively.

CHAPTER 4: CONCLUSION AND FUTURE WORK

4.1 Conclusion:

The Resume Builder Console Application successfully automates resume creation. It enhances understanding of user input handling and file operations in Java. The project provides an efficient and simple solution for beginners to learn file I/O and data formatting.

4.2 Future Work:

- Add GUI using Swing or JavaFX.
- Enable PDF generation using iText.
- Integrate database to save user resumes.
- Convert it into a web application.

4.3 Learning Outcomes:

- Practical experience with Java FileWriter and Scanner.
- Understanding of XML and plain text data storage.
- Improved knowledge of exception handling and modular programming.

4.4 Summary:

This project demonstrates the integration of basic Java concepts to solve a real-world task resume creation. It is simple, efficient, and showcases strong programming fundamentals.

REFERENCES

1. Oracle Java Documentation - https://docs.oracle.com/javase/8/docs/api/
2. W3Schools Java File Handling - https://www.w3schools.com/java/java_files_create.asp
3. GeeksforGeeks Java I/O Streams - https://www.geeksforgeeks.org/file-handling-in-java/
4. TutorialsPoint Java Scanner Class - https://www.tutorialspoint.com/java/util/java_util_scanner.htm