RESUME MAKER

A PROJECT REPORT

Submitted by

Hansika Gupta
(590023988)

in partial fulfilment for the award of the degree
of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN, UTTARAKHAND: 248007

NOVEMBER 2025

ABSTRACT

This project is a Resume Maker developed using the C programming language. It allows the user to input personal details, educational qualifications, skills, projects, experience, and achievements. The program then automatically generates a professionally formatted resume and saves it inside a text file (resume_output.txt).

The project demonstrates real-world applications of C such as structures, functions, file handling, string handling, loops, modular programming, and header files. It provides an easy and user-friendly way to generate structured resumes directly from the terminal.

PROBLEM DEFINATION

Many students and job seekers struggle to format a resume properly or do not have access to professional resume-building tools. Existing tools are either paid, complicated, or require internet access.

To solve this,I have designed a terminal-based C application that collects the necessary information and creates a clean, readable resume.

The project focuses on:

- Collecting personal and academic details
- Saving data using structures
- Reusing functions for modularity
- Writing formatted output into a file
- Providing quick and repeatable resume generation

This ensures a simple, offline, platform-independent method for anyone to generate a structured resume.

SYSTEM DESIGN

Algorithm for resume.h

1. Start
2. Define constants(MAX_TEST,DATAFILE)
3. Define all resume structure with personal and resume fields.
4. Declare functions:
    - add_resume():-add and save new resume
    - list_resume():-show all resumes
    - view_resume():-display resume by ID
    - edit_resume():-edit email of a resume
    - delete_resume():-remove resume by ID
    - export_resume():-export resume to text file
    - wait_for_key():-pause program
5. Stop

Algorithm for main.c

1. Start
2. Declare a variable choice for menu section.
3. Enter an infinite loop to display the main menu.
4. Show menu options:
    - Add Resume
    - List Resumes
    - View Resume
    - Edit Resume

    - Delete Resume
    - Export Resume
    - Exit
5. Read user input (choice).
6. Use switch(choice) to perform actions:
    - If choice = 1 → call add_resume()
    - If choice = 2 → call list_resumes()
    - If choice = 3 → call view_resume()
    - If choice = 4 → call edit_resume()
    - If choice = 5 → call delete_resume()
    - If choice = 6 → call export_resume()
    - If choice = 7 → exit program
    - Otherwise → print "Invalid choice"
7. Call wait_for_key() to pause and allow the user to read the output.
8. Loop back to show the main menu again.
9. Stop

Algorithm for resume.c

1. Start
2. Repeat menu
    - Display options:Add, List, View, Edit, Delete, Export, Exit
    - Read user choice

3. If choice=Add
    - Read user details
    - Assign next ID
    - Save resume to file

4.If choice=List

    - Read all resumes
    - Display ID, Name, Email

5.If choice = View

    - Ask for ID
    - Search and display full resume details

6. If choice = Edit

    - Ask for ID
    - Update email in file

7.If choice = Delete

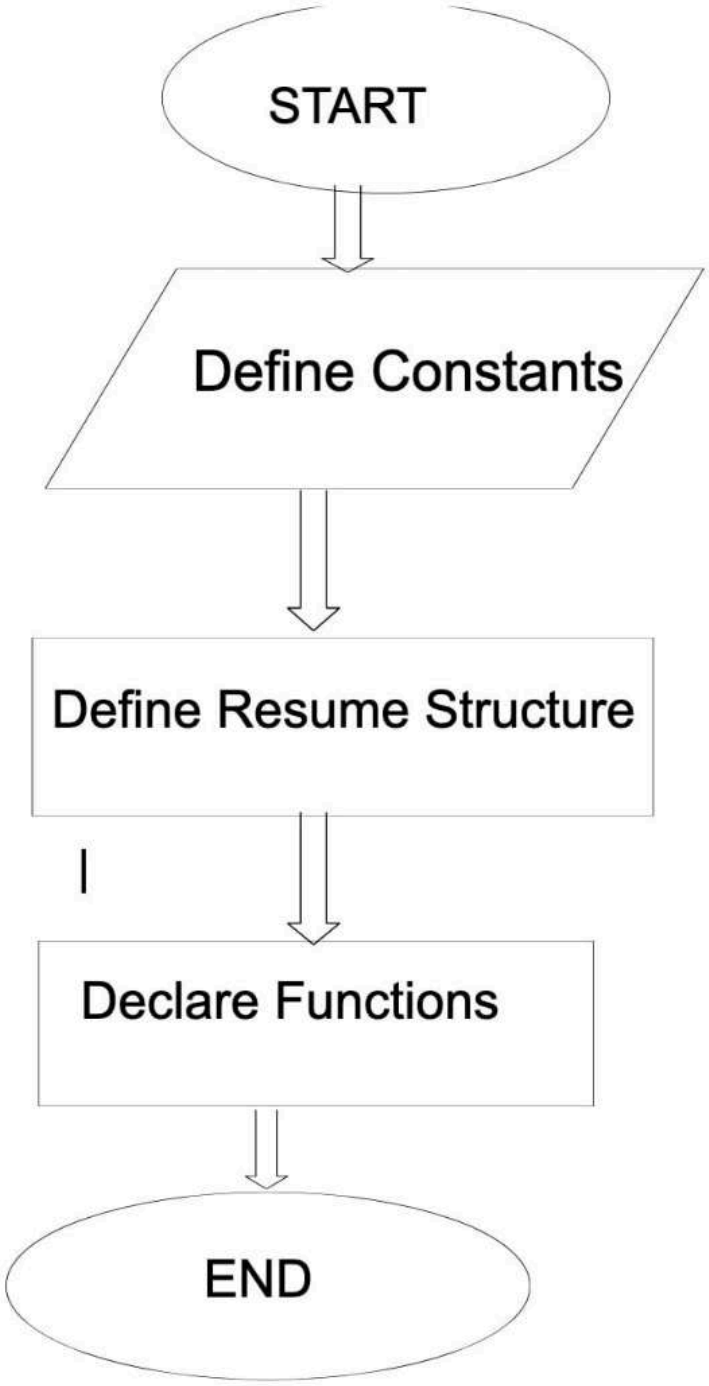
    - Ask for ID
    - Copy all except that ID into a new file
    - Replace original file

8.If choice = Export

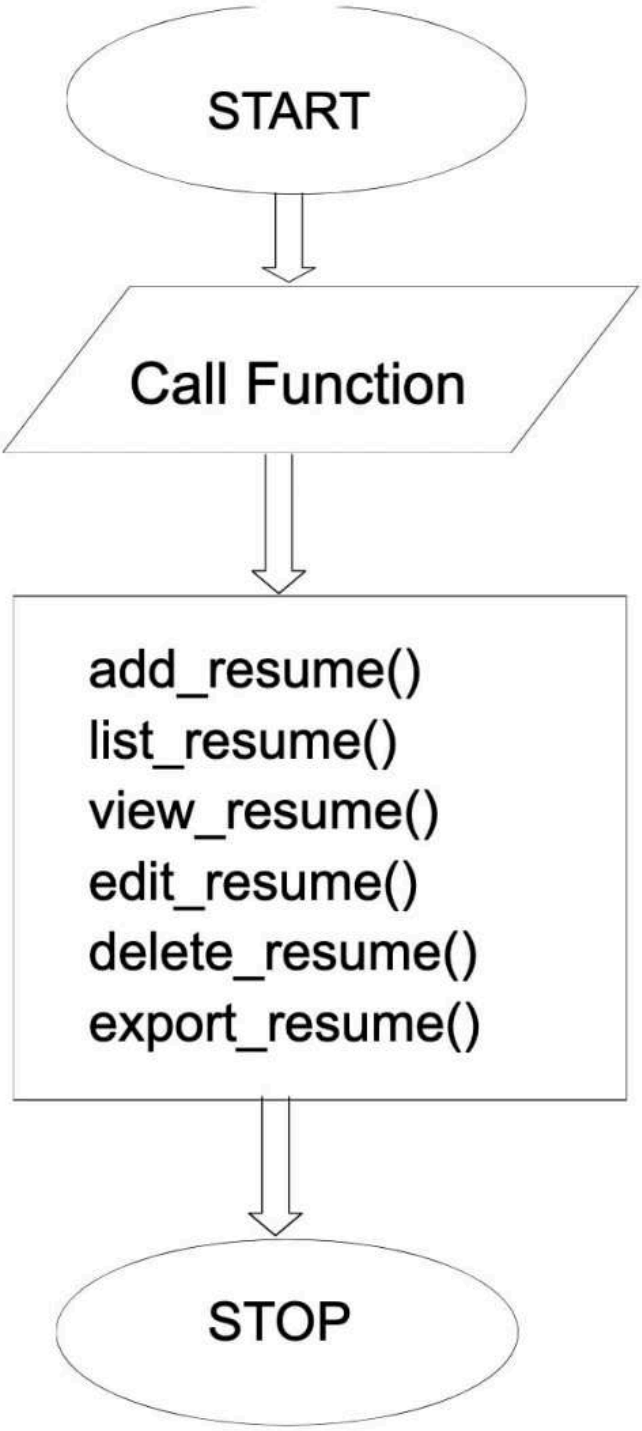
    - Ask for ID
    - Write resume content to a text file

9. If choice = Exit
    - Stop program
10. Otherwise
    - Show “Invalid choice”
11. Stop

Flowchart of resume.h



Flowchart of main.c



IMPLEMENTATION DETAILS

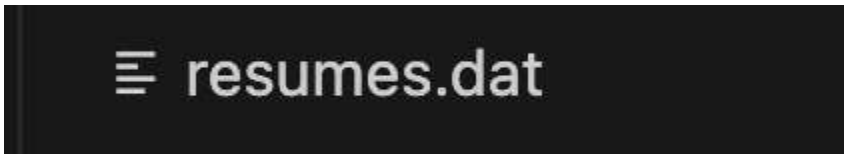
This section explains how the resume maker is built with the use of some snippets.

1.Project Structure

```
ResumeMaker/
│
├── include/
│   └── resume.h
│
├── src/
│   ├── main.c
│   └── resume.c
│
├── assets/
├── sample_input.txt
└── README.md
```

2.Data Storage Method

All resumes are stored in a binary file:



3.Use of Structures

The program uses structures in a simple way to store the record.

```c
//creating a structure to store all details entered by user
typedef struct {
    int id;   //Unique ID assigned

    //Personal Information
    char name[100];
    char email[100];
    char phone[50];
    char address[200];

    //Resume Content
    char objective[MAX_TEST];   //User's carrer objective
    char skills[MAX_TEST];      //User's skills
    char education[MAX_TEST];   // Education history of user
    char experience[MAX_TEST];  //Work or internship experience of user
    char certification[MAX_TEST];  //Extra certifications
    char hobbies[MAX_TEST];        //Hobbies or interests
    char languages[MAX_TEST];      //Languages known
} Resume;
```

4.Resume Selection Menu

The main menu provides options:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

vishnugupta@hansika RESUME MAKER % make
gcc src/main.c src/resume.c -Iinclude -o resume
vishnugupta@hansika RESUME MAKER % ./resume

=== Resume Maker ===
1. Add Resume
2. List Resumes
3. View Resume
4. Edit Resume
5. Delete Resume
6. Export Resume
7. Exit
Choose:
```

5.Input Handling Logic

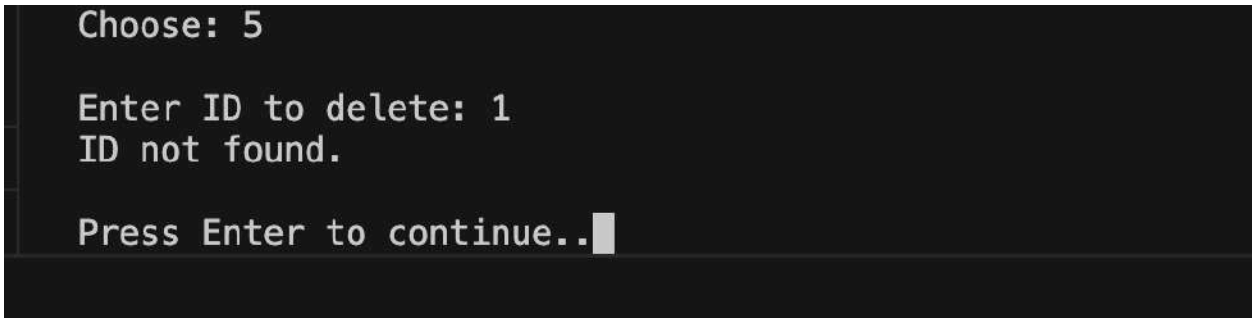
- read_line():-Removes newline characters
- getchar():-Clears leftover buffer
- fgets():-String input
-

6.Program Flow Logic

START → Display Menu → Get Choice → Perform Operation → Pause → Return to Menu → EXIT
Program will run continuously until user selects Exit.

7.Error Handling

The program checks File not found

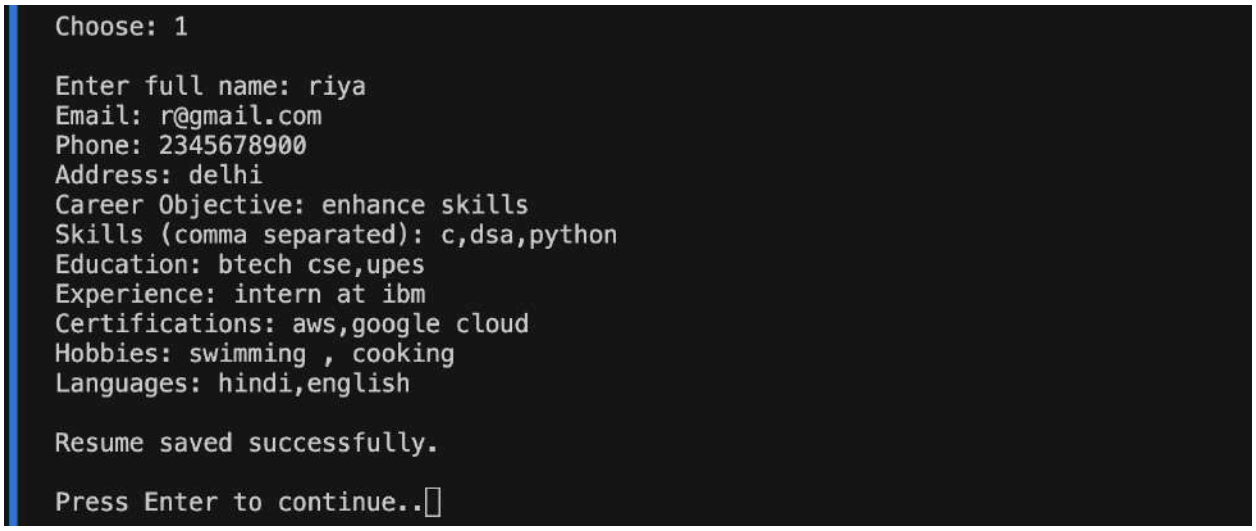


RESULTS

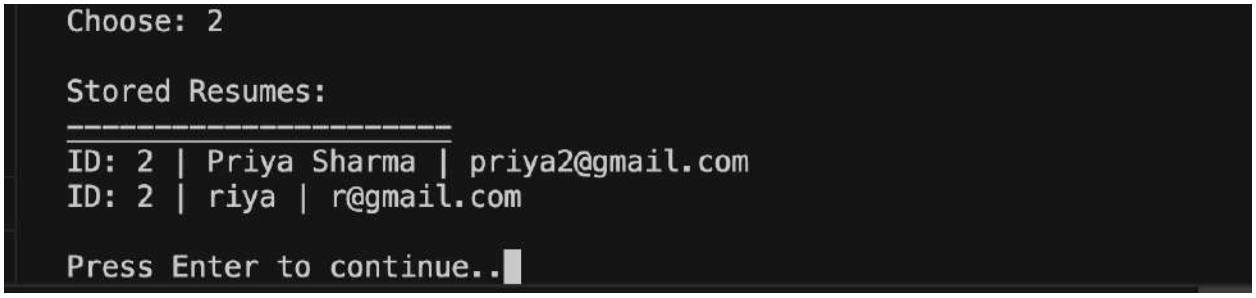
How it works

The menu runs in a while(1) infinite loop.Switch-case block is used to call the correct function.

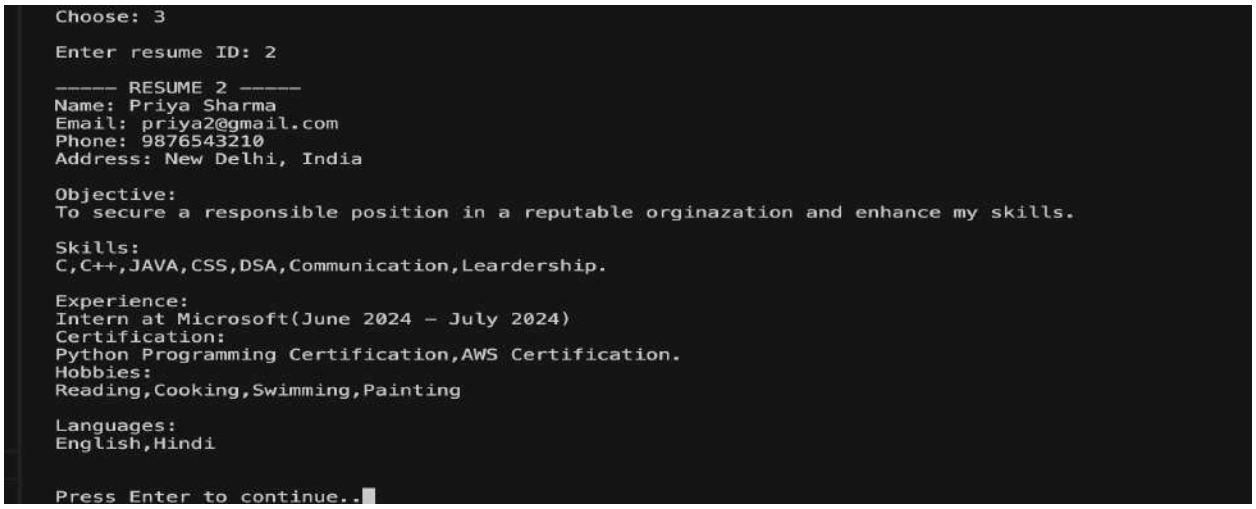
- add_resume()



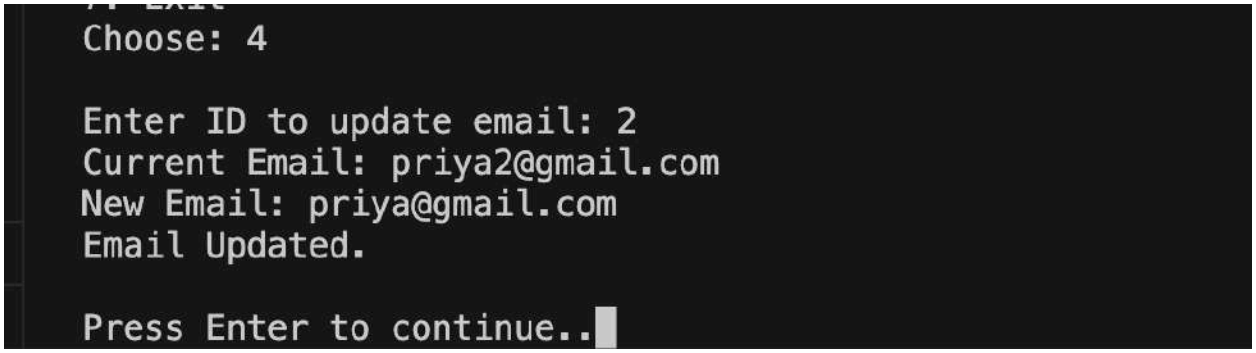
- list_resume()



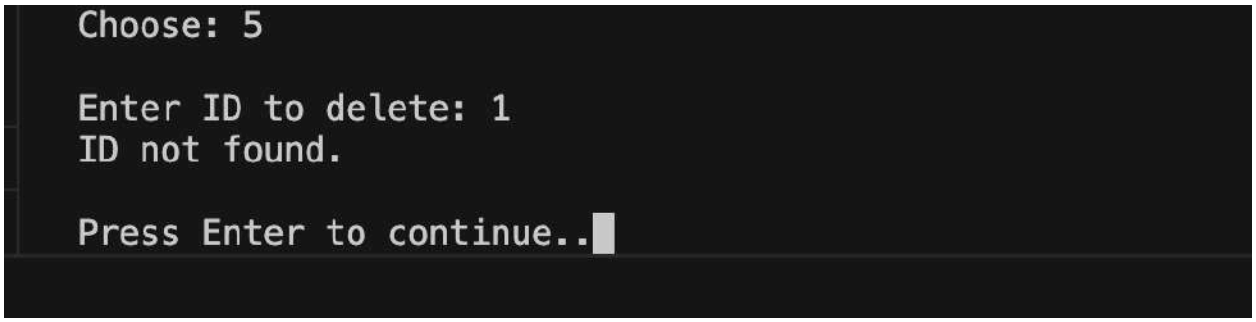
- view_resume()



- edit_resume()



- delete_resume()



- export_resume()

```
Choose: 6

Export which ID? 2
Exported to resume_export.txt

Press Enter to continue..
```

- exit_resume()

```
=== Resume Maker ===
1. Add Resume
2. List Resumes
3. View Resume
4. Edit Resume
5. Delete Resume
6. Export Resume
7. Exit
Choose: 7
vishnugupta@hansika RESUME MAKER %
```

CONCLUSION

The Resume Maker project demonstrates how C programming can be used to build a functional, menu-driven application for managing multiple resumes efficiently. Throughout the development of this system, important programming concepts such as file handling, structures, dynamic input handling, and menu control flow were implemented in a practical way.

The project provides users with a simple interface to create, view, edit, delete, and export resumes. By organizing the program into separate files(main.c,resume.c and resume.h), the system becomes easier to maintain, extend, and understand. The use of structures ensures that all resume components stay organized, and the overall program flow allows smooth interaction with different operations.

Overall, this project not only fulfills the objective of creating a resume management application but also strengthens knowledge of C programming concepts, logical structuring, and real-world problem solving.

Future Work

1. Adding More Resume Templates
2. Support for PDF or DOCX Export
3. Graphical User Interface (GUI)
4. Integration With Resume-Building Suggestions
5. Enhanced Error Handling

REFERENCES

1. https://www.geeksforgeeks.org/
2. Let Us C
3. Class PPTs
4. Github