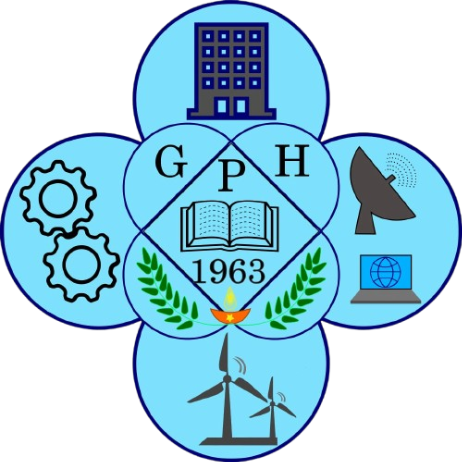
**AI RESUME MAKER**

**SYNOPSIS OF MINOR PROJECT**

SUBMITTED FOR THE PARTIAL FULFILMENT

OF

**DIPLOMA IN COMPUTER ENGINEERING**

**FEBRUARY-2025**

**SUBMITTED BY:**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **NAME OF STUDENT** | **SBRN** |
| **1** | SHRISHTI CHAUHAN | 230810204043 |
| **2** | SANIA KUMARI | 230810204050 |

**PROJECT GUIDE**

**SH. AMIT NAYYER & SH. RITESH AWASTHI**

**DEPARTMENT OF COMPUTER ENGINEERING,**

**GOVERENMENT POLYTECHNIC HAMIRPUR (H.P)-177001**

**ABOUT THE PROJECT**

The AI-Based Resume Builder is a desktop application designed to help users create professional resumes effortlessly. Built using Python and Tkinter for GUI, this tool allows users to input their personal information, experience, education, skills, projects, and awards, then generates a formatted resume in MS Word (.docx) using the python-docx library.

**PROJECT FEATURES**

* **User-Friendly GUI**: Simple and intuitive interface built with Tkinter.
* **Resume Sections**: Users can enter personal details, experience, education, skills, projects, and awards.
* **Word Document Generation**: Uses python-docx to format and export the resume as a .docx file.
* **Error Handling**: Ensures required fields (Name, Email, Phone) are filled.
* **Multi-Line Support**: Supports structured input for experience, education, and projects.
* **File Saving Options**: Users can choose where to save their resume using filedialog.

**System Capabilities:**

* **Automated Formatting:** Ensures consistent and professional resume layouts.
* **Data Validation:** Prevents incomplete or incorrect entries.
* **File Saving & Export:** Allows users to save and modify resumes in Word format.

**SCOPE OF THE PROJECT**

1. **User Input Handling**: Collects user details via Tkinter-based forms.
2. **Resume Generation**: Converts user input into a structured Word document.
3. **File Management**: Saves resumes to a user-specified location.
4. **Error Prevention**: Prevents blank or incomplete resumes.
5. **GUI Design**: Provides an easy-to-use graphical interface.

### ****TECHNICAL SCOPE****

* **Technology Used:** Python (with Tkinter for GUI).
* **Data Handling:** Local storage (JSON files or simple text files).
* **Resume Generation:** python-docx for Word file creation.
* **User Interface:** Tkinter-based graphical interface.

**LIMITATIONS**

1. **Limited Templates:** Initial version supports a few predefined resume formats.
2. **No Cloud Storage:** Resumes are saved locally on the user’s device.
3. **Manual Editing Required:** Users need to open Word for advanced formatting (if required).

**METHODOLOGY**

* **Hardware Requirements:**

**✔ Processor:** Core i3 or higher. **✔ Hard Disk:** Minimum 160 GB. **✔ Memory:** At least 2 GB RAM**.  
✔ Display:** Standard monitor (for Tkinter GUI).

* **Software Requirements:**

**✔ Operating System:** Windows 10/11 (or any OS supporting Python). **✔ Programming Language:** Python (Tkinter & python-docx).  
**✔ Dependencies: Python libraries** – tkinter, python-docx, os, filedialog.

**DESCRIPTION OF CODE FUNCTIONALITY**

**1. Imports:**

* **tkinter:** For creating the GUI.
* **ttk:** For themed widgets (better look and feel).
* **messagebox:** For displaying message boxes (errors, success, etc.).
* **filedialog:** For opening file save dialogs.
* **docx:** For creating and manipulating Word documents.
* **docx.shared.Inches:** For setting sizes in the Word document.

**2. create\_resume() Function:**

* Called when the "Create Resume" button is clicked.
* Retrieves user input from entry and text widgets.
* Checks if required fields (name, email, phone) are filled.
* Creates a new docx.Document object.
* Adds user details, summary, experience, education, skills, projects, and awards, with proper formatting.
* Uses filedialog.asksaveasfilename() to allow users to choose a location for saving the resume.
* Saves the document in .docx format.
* Displays success or error messages.
* Includes a try/except block for error handling.

**3.GUI Setup:**

* A tkinter.Tk window is created.
* Labels, entry widgets, and text widgets are designed for user input.
* Uses the grid() layout for structured alignment.
* window.columnconfigure(1, weight=1) ensures text boxes resize dynamically.
* A "Create Resume" button triggers the create\_resume() function.
* window.mainloop() starts the GUI event loop.

**ADVANTAGES**

**1. Saves Time**: Automates resume formatting and generation.  
2. **Real-Time Updates**: Instant feedback and resume creation.  
3. **Accessibility**: Works on any system with Python installed.  
4. **User-Friendly**: No coding skills required for users.

This AI-based Resume Builder is an efficient and accessible tool, making resume creation quick and effortless.

**ROLE OF THE EACH PARTICIPANT:-**

|  |  |  |
| --- | --- | --- |
| SBRN | NAME OF THE STUDENT | PARTICIPANT ROLE |
| 230810204043 | SHRISHTI CHAUHAN | Project Designer, Programmer |
| 230810204050 | SANIA KUMARI | GUI Developer, Documentation & Testing |

**TEAM MEMBERS:->**

|  |  |  |
| --- | --- | --- |
| SBRN | NAME OF THE STUDENT | SIGNATURE |
| 230810204043 | SHRISHTI CHAUHAN |  |
| 230810204050 | SANIA KUMARI |  |

**CONCLUSION:**

The AI-Based Resume Maker is a simple yet efficient tool for creating professional resumes with ease. Using Python, Tkinter, and python-docx, it provides a user-friendly interface for structured input and automated formatting. This project reduces manual effort, ensures consistency, and simplifies resume creation. While it currently has limitations like limited templates and no cloud storage, it lays a solid foundation for future improvements. Overall, this tool helps users generate high-quality resumes quickly and provides valuable learning in GUI development and document processing.