**DAY -5**

**Create a Resume Builder (Text Based)**

**1. Objective:**  
The aim of this project is to create a text-based Resume Builder using Python, where user inputs personal and professional information, and the program formats and saves it as a well-structured Word document using file handling.

**2.Software Used:**

* Google Colab (Online Python Editor)
* Python Programming Language
* python-docx Library

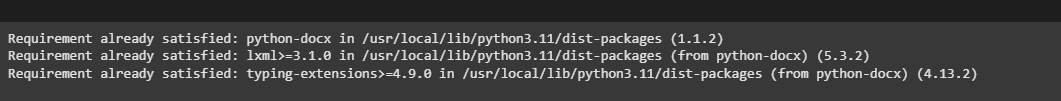
**Step 1: Open Google Colab**

1. Open your browser and search for Google Colab.
2. Click on the first link or visit: https://colab.research.google.com.
3. Click New Notebook from the bottom-right corner.
4. Name your notebook:  
   👉 **Resume\_Builder\_Project.ipynb**

**Step 2: Install python-docx Library**

1. In your first cell, type this command: **!pip install python-docx**
2. Click the Play Button on the left or press Shift + Enter to run.
3. This installs the python-docx library, which helps create and format Word (.docx) files.

**Output:**



**Step 3: Create Full User Data Collection Function**

1. In another new cell, paste this:

def collect\_user\_data():

name = input("Enter your full name: ")

email = input("Enter your email address: ")

phone = input("Enter your phone number: ")

linkedin = input("Enter your LinkedIn profile URL: ")

github = input("Enter your GitHub profile URL (if any): ")

objective = input("Enter your career objective: ")

education = collect\_education\_details()

skills = input("Enter your skills (comma-separated): ")

certifications = input("Enter your certifications (comma-separated): ")

projects = collect\_project\_details()

experience = collect\_experience\_details()

hobbies = input("Enter your hobbies (comma-separated): ")

return name, email, phone, linkedin, github, objective, education, skills, certifications, projects, experience, hobbies

1. Run this cell.

**Step 4: Create Dynamic Education Input Function**

1. In a new cell, copy this code:

def collect\_education\_details():

education\_list = []

count = 1

while True:

print(f"\nEnter Education Details {count}:")

university = input("University Name: ")

year = input("Year: ")

marks = input("Marks/Percentage: ")

education\_list.append(f"{count}. {university} ({year}) - {marks}%")

more = input("Click '+' to add more or press Enter to skip: ")

if more != '+':

break

count += 1

return "\n".join(education\_list)

1. Run the cell by pressing Shift + Enter.

**Step 5: Create Dynamic Experience Input Function**

1. In a new cell, paste this:

def collect\_experience\_details():

experience\_list = []

count = 1

while True:

print(f"\nEnter Experience {count}:")

company = input("Company Name: ")

role = input("Role: ")

duration = input("Duration (e.g., 2020-2023): ")

description = input("Describe your role briefly: ")

experience\_list.append(f"{count}. {company} - {role} ({duration})\n {description}")

more = input("Click '+' to add more or press Enter to skip: ")

if more != '+':

break

count += 1

return "\n".join(experience\_list)

1. Run this cell too.

**Step 6: Create Dynamic Project Input Function**

1. In a new cell, paste this:

def collect\_project\_details():

project\_list = []

count = 1

while True:

print(f"\nEnter Project {count}:")

project\_name = input("Project Name: ")

project\_description = input("Project Description: ")

project\_list.append(f"{count}. {project\_name}\n {project\_description}")

more = input("Click '+' to add more or press Enter to skip: ")

if more != '+':

break

count += 1

return "\n".join(project\_list)

1. Run the cell.

**Step 7: Create Word Document Formatting Function**

1. In a new cell, paste this:

from docx import Document

from docx.shared import Pt

from docx.enum.text import WD\_PARAGRAPH\_ALIGNMENT

from docx.shared import Inches

def create\_word\_resume(name, email, phone, linkedin, github, objective, education, skills, certifications, projects, experience, hobbies):

doc = Document()

for section in doc.sections:

section.page\_height = Inches(11.69)

section.page\_width = Inches(8.27)

title = doc.add\_heading(name.upper(), level=1)

title.alignment = WD\_PARAGRAPH\_ALIGNMENT.CENTER

contact\_info = doc.add\_paragraph(f"Email: {email} | Phone: {phone}\nLinkedIn: {linkedin} | GitHub: {github}")

contact\_info.alignment = WD\_PARAGRAPH\_ALIGNMENT.CENTER

sections = {

"OBJECTIVE": objective,

"EDUCATION": education,

"SKILLS": skills,

"CERTIFICATIONS": certifications,

"PROJECTS": projects,

"EXPERIENCE": experience,

"HOBBIES": hobbies

}

for section\_title, section\_content in sections.items():

doc.add\_heading(section\_title, level=2)

paragraph = doc.add\_paragraph(section\_content)

paragraph.alignment = WD\_PARAGRAPH\_ALIGNMENT.JUSTIFY

for run in paragraph.runs:

run.font.name = 'Times New Roman'

run.font.size = Pt(12)

filename = name.replace(" ", "\_") + "\_Resume.docx"

doc.save(filename)

print(f"\n✅ Word Resume saved as '{filename}'!\n")

1. Run the cell.

**Step 8: Run Your Full Program**

1. In your final cell, write:

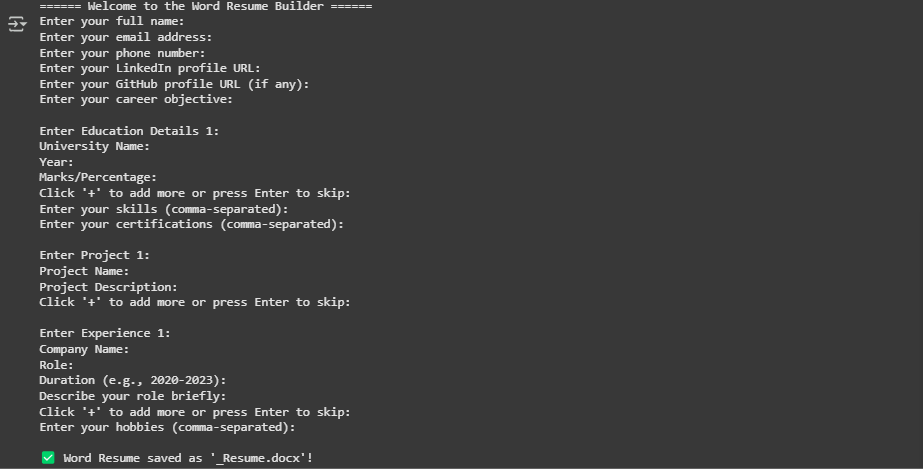
print("====== Welcome to the Word Resume Builder ======")

user\_data = collect\_user\_data()

create\_word\_resume(\*user\_data)

1. Run this cell.
2. Enter your details as the program asks.

Output:



When finished, the program will create a .docx file and save it in the Colab Files tab!