Project 1: Create a automatic resume builder by fetching details from the database.

Database must contains:

Employee details (Summary, Personal, Technical)

Project Details

Employee to Project Mapping

Generate the resume as per the below format shared. The header and footer can be present in the default layout but everything else must be generated.

The application must contain a page where user can select an employee from the search box by entering the employee id or the name with a button to fetch resume. On click of button, an PDF copy of the resume must be downloaded.

**CODE:**

python

from flask import Flask, render\_template, request, send\_file

from io import BytesIO

import pdfkit

import sqlite3

app = Flask(\_\_name\_\_)

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/resume', methods=['POST'])

def resume():

employee\_id = request.form['employee\_id']

conn = sqlite3.connect('database.db')

cursor = conn.cursor()

cursor.execute("SELECT \* FROM employees WHERE id=?", (employee\_id,))

employee = cursor.fetchone()

cursor.execute("SELECT \* FROM projects WHERE id IN (SELECT project\_id FROM employee\_projects WHERE employee\_id=?)", (employee\_id,))

projects = cursor.fetchall()

cursor.close()

conn.close()

resume\_html = render\_template('resume.html', employee=employee, projects=projects)

pdf = pdfkit.from\_string(resume\_html, False)

pdf\_buffer = BytesIO(pdf)

return send\_file(pdf\_buffer, attachment\_filename='resume.pdf', as\_attachment=True)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)