

ASSIGNMENT- 4

AI Resume and Portfolio Builder

Code:

```
import streamlit as st
from jinja2 import Template
import pdfkit
import tempfile
import base64
import openai
import os

# Optional: Set path to wkhtmltopdf
path_to_wkhtmltopdf = r"C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe"
config = pdfkit.configuration(wkhtmltopdf=path_to_wkhtmltopdf)

st.set_page_config(page_title="AI Resume Builder", layout="centered")
st.title("📄 AI Resume & Portfolio Builder")

# Sidebar for API Key
st.sidebar.title("🔒 OpenAI Settings")
api_key = st.sidebar.text_input("Enter your OpenAI API Key", type="password")

if api_key:
    openai.api_key = api_key

# Input Form
with st.form("resume_form"):
    st.subheader("👤 Profile Info")
    name = st.text_input("Full Name")
```

```
title = st.text_input("Job Title")
```

```
image = st.file_uploader("Upload Profile Image", type=['png', 'jpg', 'jpeg'])
```

```
st.subheader("📞 Contact Info")
```

```
phone = st.text_input("Phone")
```

```
email = st.text_input("Email")
```

```
linkedin = st.text_input("LinkedIn or Portfolio Link")
```

```
address = st.text_input("Address")
```

```
st.subheader("💭 About Me")
```

```
about = st.text_area("Write a brief about yourself")
```

```
if st.form_submit_button("🚀 Auto Generate About Me") and api_key and title:
```

```
    with st.spinner("Generating about me..."):
```

```
        response = openai.ChatCompletion.create(
```

```
            model="gpt-3.5-turbo",
```

```
            messages=[
```

```
                {"role": "system", "content": "You are a professional resume writer."},
```

```
                {"role": "user", "content": f"Write a professional About Me section for a resume. The job  
title is: {title}."}]
```

```
            ]
```

```
        )
```

```
        about = response.choices[0].message.content
```

```
        st.session_state['about'] = about
```

```
        st.success("About Me generated!")
```

```
st.subheader("🎓 Education")
```

```
edu1 = st.text_input("Degree 1")
```

```
edu1_year = st.text_input("Years (e.g. 20XX - 20XX)", key="edu1_year")
```

```
edu2 = st.text_input("Degree 2")
```

```
edu2_year = st.text_input("Years (e.g. 20XX - 20XX)", key="edu2_year")
```

```
st.subheader("👛 Experience")
```

```
exp1 = st.text_area("Experience 1")
```

```
st.subheader("📁 Projects")
```

```
proj1 = st.text_area("Project 1")
```

```
proj2 = st.text_area("Project 2")
```

```
st.subheader("🔧 Skills")
```

```
skills = st.text_area("List skills separated by commas")
```

```
submitted = st.form_submit_button("✅ Generate Resume")
```

```
if submitted:
```

```
    html_template = f"""
```

```
    <html>
```

```
    <head>
```

```
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <style>
```

```
        @page {{
```

```
            size: A4;
```

```
            margin: 0;
```

```
        }}
```

```
    body {{
```

```
        font-family: Arial, sans-serif;
```

```
        margin: 0;
```

```
        padding: 0;
```

```
    }}
```

```
    .container {{
```

```
        display: flex;
```

```
        width: 100%;
```

```
        border: 1px solid #ccc;
```

```
}}  
.left {{  
    background: #f2f2f2;  
    width: 35%;  
    padding: 20px;  
    text-align: center;  
}}  
.right {{  
    width: 65%;  
    padding: 20px;  
}}  
img.profile {{  
    width: 120px;  
    height: 120px;  
    border-radius: 50%;  
    object-fit: cover;  
}}  
h2 {{  
    margin-bottom: 5px;  
}}  
.section-title {{  
    background: #555;  
    color: white;  
    padding: 5px;  
    font-weight: bold;  
    margin-top: 20px;  
}}  
ul {{  
    padding-left: 20px;  
    margin: 0;  
}}
```

```

    li {{
        margin-bottom: 5px;
    }}
</style>
</head>
<body>
    <div class="container">
        <div class="left">
            {f'' if image else ''}
            <h2>{name}</h2>
            <p><i>{title}</i></p>
            <hr>
            <h4>Contacts:</h4>
            <p>{phone}<br>{email}<br>{linkedin}<br>{address}</p>
            <div class="section-title">Education:</div>
            <p>{edu1_year}<br>{edu1}</p>
            <p>{edu2_year}<br>{edu2}</p>
            <div class="section-title">Skills:</div>
            <ul>
                {''.join([f'<li>{skill.strip()}</li>' for skill in skills.split(',') if skill.strip()])}
            </ul>
        </div>
        <div class="right">
            <div class="section-title">About Me:</div>
            <p>{about}</p>
            <div class="section-title">Experience:</div>
            <ul>
                <li>{exp1}</li>
            </ul>
            <div class="section-title">Projects:</div>

```

```
        <ul>
            <li>{proj1}</li>
            <li>{proj2}</li>
        </ul>
    </div>
</div>
</body>
</html>
""""
```

```
st.markdown("---")
st.subheader("📄 Resume Preview")
st.components.v1.html(html_template, height=800, scrolling=True)
```

```
# Save HTML to temp file
```

```
with tempfile.NamedTemporaryFile(delete=False, suffix=".html") as tmp_html:
```

```
    tmp_html.write(html_template.encode("utf-8"))
```

```
    tmp_html_path = tmp_html.name
```

```
pdf_path = tmp_html_path.replace(".html", ".pdf")
```

```
# PDFKit options to match screen and use full width
```

```
options = {
```

```
    'enable-local-file-access': None,
```

```
    'page-size': 'A4',
```

```
    'margin-top': '0in',
```

```
    'margin-bottom': '0in',
```

```
    'margin-left': '0in',
```

```
    'margin-right': '0in',
```

```
    'encoding': "UTF-8",
```

```
    'zoom': '1.3'
```

```
}
```


```
# Convert to PDF
```

```
pdfkit.from_file(tmp_html_path, pdf_path, configuration=config, options=options)
```

```
# Display download link
```

```
with open(pdf_path, "rb") as f:
```

```
    base64_pdf = base64.b64encode(f.read()).decode("utf-8")
```

```
    pdf_download = f'<a href="data:application/pdf;base64,{base64_pdf}"  
download="resume.pdf">  Download Resume as PDF</a>'
```

```
    st.markdown(pdf_download, unsafe_allow_html=True)
```

Description:

- Streamlit web app that lets users build a resume/portfolio interactively.
- Integrates OpenAI's GPT-3.5 API to auto-generate the "About Me" section.
- Uses HTML + CSS styling to format resume content.
- Converts the resume into a PDF file using pdfkit and wkhtmltopdf.
- OpenAI Key Input: Securely input OpenAI API key via sidebar.
- Form-Based Input Sections :
 - Full Name, Job Title, Profile Image Upload
 - Contact Info: Phone, Email, LinkedIn, Address
 - About Me (manual or AI-generated)
 - Education (two entries)
 - Work Experience (one entry)
 - Projects (two entries)
 - Skills (comma-separated)
- If the user provides a job title and clicks "Auto Generate About Me", GPT-3.5 generates a professional paragraph for that section.

Output:

OpenAI Settings

Enter your OpenAI API Key

OpenAI Settings

Enter your OpenAI API Key

OpenAI Settings

Enter your OpenAI API Key

AI Resume & Portfolio Builder

Profile Info

Full Name

gayatri

Job Title

senior developer

Upload Profile Image

Drag and drop file here

Limit 2000KB per file • PNG, JPG, JPEG

Browse files

women.jpeg 18.0KB

Contact Info

Phone

09845269586

Email

gayatribanikar@gmail.com

LinkedIn or Portfolio Link

Address

Abhilash Nilaya 20th ward, near radha krishna school, MMTC colony, MJ nagar, hospete

Education

Degree 1

bachelor of engineering in computer science

Years (e.g. 20XX - 20XX)

2020-2024

Degree 2

Years (e.g. 20XX - 20XX)

Experience

Experience 1

2 Years experience

Projects


Project 1

AQI Monitoring System

Project 2

GenAI Medical Diagnosis Assistant

Resume Preview



gayatri
senior developer

Contacts:

09845269586
gayatribanikar@gmail.com
abhilash nilaya 20th ward,
near radha krishna school,
MMTC colony, MJ nagar,
hospete

Education:

2020-2024
bachelor of engineering in
computer science

Skills:

• java script
• java

About Me:

I am interested in chess,basketball

Experience:

• 2 Years experience

Projects:

• AQI Monitoring System
• GenAI Medical Diagnosis Assistant

Download Resume as PDF