Model Development Phase Template

Date	15 March 2024
Team ID	SWTID1719942077
Project Title	Career Mapper
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

career_mapper > app.py > ...

```python
from dotenv import load_dotenv
load_dotenv()
import os
import streamlit as st
import google.generativeai as palm

api_key = os.getenv("PALM_API")
if not api_key:
    raise ValueError("PALM_API key not found in environment variables")
palm.configure(api_key=api_key)
model_name = 'models/text-bison-001'

st.set_page_config(page_title="CareerMapper")
# streamlit app
st.title("CareerMapper: AI-Powered Personal Career Mapping")

# Introduction text
st.markdown("""
CareerMapper: AI-Powered Personal Career Mapping

CareerMapper is an innovative platform designed to provide personalized career mapping, guidance, and job recomm

## Scenario 1: Student Career Exploration
Students often face challenges when deciding on their future career paths. CareerMapper helps students explore v

## Scenario 2: Professional Development for Working Professionals
Working professionals seeking career advancement or considering a career change can benefit from CareerMapper's
```

```
## Scenario 3: Career Transition for Job Seekers
Job seekers undergoing career transitions often face uncertainty and challenges in navigating the job market. Ca
""")

def generate_career_pathways(user_data):
    prompt = f"""
    role: system, content: Suggest good career options based on the data provided with proper explanation,
    role: Example,
    content:
    Personal Information: [age:22, gender:male, educational level:UG],
    Interests: [Hobbies:Playing football, coding, Maths],
    Skills: [Skills:C++,PyTorch, ML],
    Career choices:
    1. Software Development
    •   Leverage Skills: Your proficiency in C++ and Python provides a strong foundation for software developmen
    •   Potential Roles: You can explore roles like software engineer, backend developer, game developer (given
    •   Growth Opportunities: The software development field offers ample growth opportunities, with potential t
    2. Machine Learning Engineer
    •   Build on Strengths: Your knowledge of PyTorch and ML is directly applicable to this role.
    •   Industry Demand: Machine learning is a rapidly growing field with high demand for skilled professionals.
    •   Potential Roles: You could work on developing ML models for various applications, such as image recognit
    3. Academic Research
    •   Explore Further: If you have a deep interest in mathematics or machine learning, you could consider purs
    •   Potential Roles: You could work as a research assistant or pursue a career in academia after completing
    role:Query,content: Personal Information: [age:{user_data[0]}, gender:{user_data[1]}, educational level:{use
    """
    response = palm.generate_text(model=model_name, prompt=prompt)
    return response.result
```

```
# Define the form
with st.form(key='career_form'):
    st.subheader("Personal Information")
    age = st.number_input("Age", min_value=0, max_value=100, value=20, step=1)
    gender = st.selectbox("Gender", ["Male", "Female", "Other"], index=0)
    education_level = st.selectbox("Educational Level", ["UG", "PG", "PhD", "Other"], index=0)
    if education_level == "Other":
        other_education = st.text_input("Please specify your education level")
    st.subheader("Interests")
    hobbies = st.text_area("Hobbies (separate by commas)")
    st.subheader("Skills")
    skills = st.text_area("Skills (separate by commas)")
    submit_button = st.form_submit_button(label='Submit')

# Process the form submission
if submit_button:
    if education_level == "Other":
        education_level = other_education
    personal_info = {
        "age": age,
        "gender": gender,
        "education_level": education_level
    }
    interests = [hobby.strip() for hobby in hobbies.split(',')]
    skills_list = [skill.strip() for skill in skills.split(',')]

    # Generate career pathways
    user_data = [age, gender, education_level, hobbies, skills]
    career_pathways = generate_career_pathways(user_data)

    st.subheader("Career Pathways")
    st.write(career_pathways)
```