AI-Powered Career Guide Web App

**🌐 Live Project Link:**

**🔗** [**https://career-guide-ai-2.onrender.com**](https://career-guide-ai-2.onrender.com)

**🔍 Project Overview:**

**CareerAI** is a machine learning–powered web application that provides **personalized career guidance** to students and job seekers. Users enter their background (skills, interests, education, and experience), and the app recommends suitable career paths with future market insights.

**🧩 Key Features:**

1. ✅ **User Input Form**
   * Collects details like name, education, experience, skills, interests, and learning style.
2. ✅ **AI-Powered Analysis**
   * Processes input using natural language parsing and ML models to match user profiles with suitable career paths.
3. ✅ **Personalized Recommendations**
   * Generates detailed JSON-based suggestions including required skills, industry demand, and future scope.
4. ✅ **Result Download (PDF)**
   * Users can download career guidance results as a professionally formatted PDF.
5. ✅ **Live Deployment**
   * Hosted and publicly accessible via [Render](https://career-guide-ai-2.onrender.com).

| **Layer** | **Technology** |
| --- | --- |
| Frontend | HTML, CSS, JavaScript |
| Backend | Python, Flask |
| AI/ML Logic | Custom ML model in career\_guide\_ai.py |
| Deployment | Render (cloud hosting) |
| PDF Export | ReportLab (Python library) |

**🎯 Objective of the Project:**

To assist individuals in choosing the right career path using technology-driven analysis, by matching personal preferences and qualifications with real-world industry demand.

**📁 File: career\_ai\_web\_app.py**

This file is the **core backend** for your **CareerAI** web application using **Python + Flask**.

**✅ 1. Imports and Initialization:**

from flask import Flask, render\_template, request, jsonify, session, redirect, send\_file

import json, os, io

from datetime import datetime

from career\_guide\_ai import CareerGuideAI

from reportlab.pdfgen import canvas

* **Flask** – Web server framework to handle routing and web pages.
* **CareerGuideAI** – Custom ML logic for generating career suggestions (imported from your model file).
* **ReportLab** – Library to generate PDF files dynamically.
* **session** – Temporarily stores user data between requests.

✅ **2. App Setup:**

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

app.secret\_key = 'career\_guide\_ai\_secret\_key\_2024'

career\_ai = CareerGuideAI()

* Initializes the Flask app and loads the career guidance model.

✅ **3. Routes Explained:**

**🏠 / – Home Page**

@app.route('/')

def index():

return render\_template('index.html')

* Loads the HTML form for the user to input data.

**📊 /analyze – Main Logic (POST)**

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

def analyze():

* **Receives JSON data** from the user form (name, skills, education, etc.)
* **Formats it** into a readable string.
* Passes it to the ML model:

user\_profile = career\_ai.parse\_user\_input(user\_input)

guidance\_text, json\_output = career\_ai.generate\_guidance(user\_profile)

* **Stores results** in session for later use.
* Returns the analysis as a JSON response (used by frontend to show results).

**📃 /results – Results Page**

@app.route('/results')

def results():

if 'guidance\_text' not in session:

return redirect('/')

return render\_template('results.html')

* Displays the **results page** only if the user has submitted data.

**🧾 /download/<format> – PDF Download**

@app.route('/download/<format>')

def download(format):

* Lets users **download their career guidance** as a PDF file.
* Uses ReportLab to generate a PDF **on the fly**.
* Returns the file as an attachment to be saved by the user.

**🧠 /api/career-tracks – Get Career Paths (API)**

@app.route('/api/career-tracks')

def get\_career\_tracks():

* Returns all available **career tracks** from the AI model (skills, growth, market demand).

**🔎 /api/skills – Autocomplete API**

@app.route('/api/skills')

def get\_skills():

* Sends a list of skills supported by the app (used for **autocomplete** in the frontend).

**📊 /api/results – Get Result from Session**

@app.route('/api/results')

def get\_results():

* Sends stored career suggestion results in JSON format (used if the user refreshes the page).

**ℹ️ /about and /contact**

@app.route('/about')

@app.route('/contact')

* Render static **About** and **Contact** pages.

✅ **4. Run App Locally:**

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

app.run(debug=True, host='0.0.0.0', port=5000)

* Starts the web server on **localhost** or network IP.