1. Page to upload a resume from a candidate
2. A second page to upload title and job description
3. Resume and JD will be fed to AI
4. AI will find keywords from JD
5. Then it will try to find these keywords in the resume
6. It will give back result of total number of keywords in JD and total number of keywords matched to resume
7. 75% or more is favorable; less than 75% is unfavorable give as an output.

Help me create a web application. I am using VSC on windows computer. I plan on deploying this in azure platform and using Python Django language. I want step by step instruction on how to start in VSC as well as run the code locally.

I want to create a front end that will do the following:

1. A modern looking page that asks user to upload a candidate’s resume. This resume is going to be in a pdf format. There should be a submit button which starts the process for a) below.
2. Once submit button is clicked in 1), a different page should ask for a job title and job description. User will disclose job title in a textbox. User will upload job description from local drive in a word document.
3. Once d) below is complete, in a new page, the web app is going to display that the resume is favorable if the division is 75% or higher. If less than 75% then it will display that the resume is unfavorable.

For the backend:

1. Once 1) above is complete, I want to extract all texts from the pdf resume. This should be saved in a file in my C: drive in a folder called “Resume Code” as “Resume extracted”.
2. Once 2) above is complete, I want to extract all texts from the word job description. This should be saved in a file in my C: drive in a folder called “Resume Code” as “JD extracted”.
3. I want to provide text from “Resume extracted” document and text from “JD extracted” document to gpt 4 model using Microsoft azure open ai api call (I only want MS azure code and not openai api call). The prompt will ask gpt 4 to find all the keywords from both texts. It will then match these keywords with the two texts. Then gpt 4 will provide the number of keywords in the job description and the number of keywords matched between job description and resume in a json format. I only want the numbers and no text.
4. Once we have the number of keywords in the job description (lets call it JDKey) and the number of keywords matched between job description and resume(lets call it MatchKey). We will divide MatchKey by JDKey.

Think harder and make sure that the logic flows properly.