📘 Technical Documentation Summary

## page\_1

📄 Source: tmp\_repo\StepUpAI\app.py

The `page\_1` function is responsible for displaying a welcome message and prompting the user to enter their full name and email address. If both fields are filled out and the user clicks the "Proceed to Resume Analysis" button, the user's name and email are stored in the session state and the user is directed to page 2.  
  
This code snippet is essential for capturing user information and progressing to the next step in the resume analysis process. It provides a user-friendly interface for inputting personal details and seamlessly transitioning to the next stage.

——————————————————————————————

## load\_examples

📄 Source: tmp\_repo\StepUpAI\app.py

This code snippet defines a function called load\_examples that reads data from a JSON file containing examples for skill gap analysis. It then extracts relevant information from the examples, such as resume summaries and target roles, and generates embeddings for each example text using a function called get\_embedding. The function returns a tuple containing the examples and their corresponding embeddings. This code is essential for loading and processing data for skill gap analysis in the StepUpAI platform.

——————————————————————————————

## load\_role\_skills

📄 Source: tmp\_repo\StepUpAI\app.py

This code snippet loads role-specific technical and soft skills from a JSON file and returns a dictionary mapping each role to its corresponding skills. The JSON file path is "/mount/src/stepupyourcareer.ai/StepUpAI/role\_skills.json". The returned dictionary structure includes keys for "technical\_skills" and "soft\_skills" for each role. If a role entry in the JSON file does not contain the required fields, it will be excluded from the final dictionary.

——————————————————————————————

## get\_embedding

📄 Source: tmp\_repo\StepUpAI\app.py

Function: get\_embedding  
  
Description:  
This function takes a text input and generates an embedding using the specified model. It then returns the embedding for the input text.  
  
Parameters:  
- text: The text input for which the embedding needs to be generated.  
- model (optional): The name of the model to be used for generating the embedding. Default value is "text-embedding-ada-002".  
  
Return Value:  
- The embedding generated for the input text.  
  
Usage:  
embedding = get\_embedding("This is a sample text.")  
print(embedding)  
  
Note: Make sure to have the necessary client setup and permissions to use the embeddings service.

——————————————————————————————

## extract\_text\_from\_pdf

📄 Source: tmp\_repo\StepUpAI\app.py

Function: extract\_text\_from\_pdf  
  
Description:  
This function takes an uploaded PDF file as input and extracts text from each page of the PDF using the PdfReader class. It then concatenates the extracted text from each page into a single string with newline characters separating the text from each page.  
  
Parameters:  
- uploaded\_file: The PDF file to extract text from.  
  
Returns:  
A string containing the extracted text from each page of the PDF file, with newline characters separating the text from each page. If no text is extracted from a page, an empty string is included in the final output.  
  
Example Usage:  
extracted\_text = extract\_text\_from\_pdf("sample.pdf")  
print(extracted\_text)

——————————————————————————————

## anonymize

📄 Source: tmp\_repo\StepUpAI\app.py

Function: anonymize  
  
Description:  
This function takes a text input, sends it to a chat completion model "gpt-4o-mini" to anonymize personal identifiers from resumes, and returns the anonymized text.  
  
Parameters:  
- text: The input text containing personal information to be anonymized.  
  
Return Value:  
- The anonymized text with personal identifiers removed.  
  
Usage:  
anonymize("John Doe\n123 Main St\njohndoe@email.com\n555-555-5555")  
  
Example:  
Input: "John Doe\n123 Main St\njohndoe@email.com\n555-555-5555"  
Output: "123 Main St"

——————————————————————————————

## retrieve\_examples

📄 Source: tmp\_repo\StepUpAI\app.py

Function: retrieve\_examples  
  
Description:  
This function retrieves the top k examples that are most similar to a given query based on their embeddings.  
  
Parameters:  
- query: the query for which similar examples are to be retrieved  
- embeddings: a list of embeddings for all examples  
- examples: a list of examples corresponding to the embeddings  
- k: the number of top examples to retrieve (default is 3)  
  
Returns:  
A list of the top k examples that are most similar to the query based on their embeddings.  
  
Example Usage:  
query = "apple"  
embeddings = [embedding1, embedding2, embedding3]  
examples = ["apple pie", "apple cider", "apple orchard"]  
top\_examples = retrieve\_examples(query, embeddings, examples, k=2)

——————————————————————————————

## generate\_skill\_gap

📄 Source: tmp\_repo\StepUpAI\app.py

This code snippet defines a function called generate\_skill\_gap that takes in parameters such as resume\_text, target\_role, retrieved\_examples, and fallback\_skills. The function generates a prompt based on the examples provided, required skills, and a new resume for skill gap analysis. The prompt includes information on technical skill gaps, soft skill gaps, and transferable skills.  
  
The function then sends this prompt to a chat client using the GPT-4o-mini model for completion. The response is processed to extract the JSON-formatted skill gap analysis and returned as the output of the function.  
  
This code is designed to automate the process of identifying skill gaps in resumes for a given target role, based on provided examples and required skills. The output is a structured JSON format listing missing technical skills, missing soft skills, and transferable skills from the resume.  
  
This function can be integrated into a larger system for career advising or recruitment purposes to streamline the analysis of candidate resumes and identify areas for skill development or improvement.

——————————————————————————————

## get\_skill\_priorities\_from\_gpt

📄 Source: tmp\_repo\StepUpAI\app.py

This function, `get\_skill\_priorities\_from\_gpt`, takes in a list of skills and a target role as input. It uses the GPT-4o-mini model to generate a JSON mapping of each skill to an importance score from 1 to 100 based on the 80/20 (Pareto) principle. The output JSON format should list each skill along with its corresponding importance score. The function makes a call to the GPT-4o-mini model through a client to generate the response. If the response is in valid JSON format, it is returned; otherwise, an empty dictionary is returned.

——————————————————————————————

## extract\_json\_from\_response

📄 Source: tmp\_repo\StepUpAI\app.py

Function: extract\_json\_from\_response  
  
Description:  
This function takes in a raw string and extracts a JSON object from it. It uses regular expressions to find the JSON object within the raw string and then attempts to parse it using the json.loads() method. If successful, it returns the parsed JSON object. If there is an error decoding the JSON object, it returns an empty dictionary.  
  
Parameters:  
- raw: a string containing the raw data from which to extract the JSON object  
  
Returns:  
- A dictionary representing the extracted JSON object, or an empty dictionary if the JSON object cannot be decoded.

——————————————————————————————

## load\_skill\_resources

📄 Source: tmp\_repo\StepUpAI\app.py

Function: load\_skill\_resources()  
  
Description:  
This function is responsible for loading skill resources from a JSON file located at "/mount/src/stepupyourcareer.ai/StepUpAI/skill\_resource\_mapping.json".  
  
Parameters:  
None  
  
Returns:  
A dictionary containing the skill resources loaded from the JSON file.  
  
Usage:  
Call this function to retrieve the skill resources needed for the StepUpAI application.

——————————————————————————————

## split\_skills\_by\_rag\_presence

📄 Source: tmp\_repo\StepUpAI\app.py

Function: split\_skills\_by\_rag\_presence  
  
Description:  
This function takes a list of skills and a list of keys representing skills that are present (rag\_skill\_keys). It splits the input list of skills into two separate lists - one containing skills that are present in the rag\_skill\_keys list and one containing skills that are missing.  
  
Parameters:  
- skills: a list of skills to be split  
- rag\_skill\_keys: a list of keys representing skills that are present  
  
Returns:  
- present: a list of skills that are present in the rag\_skill\_keys list  
- missing: a list of skills that are missing from the rag\_skill\_keys list  
  
Example Usage:  
skills = ["Python", "Java", "SQL"]  
rag\_skill\_keys = ["Python", "SQL"]  
present, missing = split\_skills\_by\_rag\_presence(skills, rag\_skill\_keys)  
print("Present skills:", present)  
print("Missing skills:", missing)  
  
Output:  
Present skills: ["Python", "SQL"]  
Missing skills: ["Java"]

——————————————————————————————

## generate\_hybrid\_action\_plan

📄 Source: tmp\_repo\StepUpAI\app.py

This code snippet defines a function called generate\_hybrid\_action\_plan that takes in technical, soft, and transferable skills along with a dictionary of skill resources. The function splits the input skills into two categories - those present in the skill resources dictionary and those not present. It then constructs a plan with relevant resources for the skills found in the dictionary.  
  
If any skills are not found in the dictionary, the function generates a prompt for a GPT (Generative Pre-trained Transformer) model to provide resources for those uncovered skills. The prompt specifies the format for the response in JSON and includes instructions for providing top-rated courses, books, and articles/videos for each skill category. The function then calls the GPT model to generate resources for the uncovered skills and merges the results into the existing plan.  
  
In case of any errors during the GPT generation process, an error message is displayed. The function ultimately returns the complete action plan with resources for all skills, including any additional resources generated by the GPT model.

——————————————————————————————