Write me a set of python functions for a jiu-jitsu analysis and coaching app. I want to use the current version of this file called jiu-jitsu functions, but adjust the file contents based on this new prompt. I want the following top level functions, using the helper functions I have included in the genai.py, chatgpt.py, and movieai.py files. The top level functions I want are:

1. A function called “generate\_grappling\_plan”: I want to send the function: a still frame image of a grappling match; provide a variable for whether the image is to recognize the top jiu-jitsu player, the bottom jiu-jitsu player, or both players called “player\_variable”; an optional variable for whether or not this is an MMA or Jiu-jitsu match called “isMMA”; and an optional variable for some keywords to prime the ai on what to look for.
   1. Then I want call genai.generate\_image\_description and pass it the image, and the following prompt: “the image is a still frame from a grappling match. The match has <<isMMA>> rules. I want you to analyze: <<player\_variable>>, and provide three options for the next immediate steps towards grappling moves that would work best for them given their current position. I want these steps to be listed in quick bullet format like ex: “1) <<step>> : towards <<move>>, 2) …” Base the recommendations on historical MMA and Jiu-jitsu match performance. I want you to also take special note of the following keywords for your image and recommendation analysis: <<keywords>>”
   2. I then want to return the text of the response as the overall function return
2. A function called “analyse\_grappling\_match”: I want to send the function: a video of a grappling match; provide a variable for whether the image is to recognize the top jiu-jitsu player, the bottom jiu-jitsu player, or both players called “player\_variable”; an optional variable for whether or not this is an MMA or Jiu-jitsu match called “isMMA”; an optional variable for some keywords to prime the ai on what to look for, a variable called “start\_time”, and a variable called “end\_time”.
   1. If provided a start\_time and an end\_time, I want to generate a subset of the video provided which contains only the part of the video between the start and end times.
   2. Then I want to call the Genai.generate\_video\_description function subset of the video just created if present, or the original video if there was no subset generated. I will call the function with the following prompt: “these images are frames from from a grappling match. The match has <<isMMA>> rules. I want you to analyze: <<player\_variable>>, and provide an analysis of what went right or wrong during the match, along with some recommendations for the <player\_variable> in their next match” I want these recommendations to be listed in a quick summary format. Base the recommendations on the athlete’s body type, jiu-jitsu guard situation, and historical MMA and Jiu-jitsu match performance. I want you to also take special note of the following keywords for your analysis: <<keywords>>”
3. A function called “generate\_flow\_chart”: I want to send the function: a text description of the physical attributes of the athlete called “measurables”; a variable for whether the image is to be from a top jiu-jitsu position, bottom jiu-jitsu position, or both positions “position\_variable” (where the default is both positions); an optional variable for whether or not this is an MMA or Jiu-jitsu match called “isMMA” (default is yes); and an optional variable for some keywords to prime the ai on what to come up with for called “favorite\_ideas”
   1. Then I want to call Genai.generate\_text with the following prompt: “I want you to generate a Mermaid based radial visual flow chart of jiu-jitsu moves that will have the greatest likelihood of success for an athlete with the following measurables: <measurables>. I want this chart to focus on <position\_variable> under the <isMMA> ruleset, and take into account that the athlete has the following ideas: <favorite\_ideas> of the athlete. Could you label the flow arrows in the diagram using a description of the primary movement required to get to that bubble. Only return the mermaid object, and not any shoulder text whatsoever”
   2. I want the function to return a string called mermaid\_object which contains the response from Genai.generate\_text
4. A function called “gracie\_talk”: I want to pass the function a text variable for the name of a Gracie to speak with called “gracie\_name”, a text variable called “chat\_history” which contains the history of the chat, and a text variable called “user\_messaage” which contains the next prompt in the conversation. Using these inputs, I want to call Genai.generate\_chat\_response with the <chat\_history>, the <user\_message>, and the following prompt: “you are the famous martial arts practitioner <gracie\_name>. I want you to have a conversation with the user based on that information, and have the conversation focused around improving the user’s grappling and self-defense. I want the conversation to always have an upbeat and motivational conversational style.”
5. A function called: “generate\_mermaid”, which takes in string called mermaid\_object, and coverts it into a displayable javascript based Mermaid object. Then the function will return the Mermaid object.
6. A function called: “display\_graph”, which takes in a Mermaid objected called flow\_chart, and then displays the chart as an HTML object which can be expanded or shrunk to fit the user’s desired size.
7. A function called next\_move, which takes in the mermaid chart called flow\_chart, and the text of the next move selected called move\_text, a text description of the physical attributes of the athlete called “measurables”; an optional variable for whether or not this is an MMA or Jiu-jitsu match called “isMMA” (default is yes); and an optional variable for some keywords to prime the ai on what to come up with for called “favorite\_ideas”It will then search the flow chart for the text of that move represented as one of the lines, and select the next node in sequence as the center of a new diagram. The text of this node will be called new\_position. Then it will call generate\_flow\_chart with all of those inputs and pass it the new\_position variable as the “position\_variable” input. The function will return
8. I also want a function called “get\_attributes”: I want to send the function: a still frame image of a grappling match; provide a variable for whether the image is to recognize the top jiu-jitsu player, the bottom jiu-jitsu player, or both players called “player\_variable”; it then calls genai.generate\_image\_description to ask the ai what the height and weight of the athlete in a particular position is.

Useful helper functions

Genai.generate\_text (needs prompt + possibly instance instructions)

Genai.generate\_chat\_response (chat\_history and user\_message)

Genai.generate\_image(prompt)

Genai.encode\_image(file)

Genai.generate\_image\_description(image, instructions)

Genai.extract\_frames (video, max samples)

Genai.generate\_video\_description(vide, max samples, instructions)

Give me a list of 15 top jiujistu masters. I want a focus on both historical and modern jiujitsu. The list must include helio, royce, renner, renzo, John Danaher, and Eddie bravo + whoever else males the most senseEditTop 15 Jiu-Jitsu Masters

Here’s a list of 15 top jiu-jitsu masters spanning both historical and modern eras:

Helio Gracie – Modified traditional Japanese jiu-jitsu to create Brazilian Jiu-Jitsu, adapting techniques to work for smaller practitioners.

Royce Gracie – Revolutionized martial arts by winning early UFC tournaments, demonstrating BJJ’s effectiveness against larger opponents.

Rickson Gracie – Considered by many to be the greatest Gracie fighter, with legendary technical precision and an undefeated record.

Mitsuyo Maeda – Japanese judoka who taught Carlos Gracie and was instrumental in bringing jiu-jitsu to Brazil.

Renzo Gracie – Expanded the Gracie legacy globally and developed a more well-rounded approach to combat sports.

John Danaher – Revolutionary coach and strategist who systematized leg locks and transformed modern BJJ through his “Death Squad.”

Roger Gracie – Ten-time BJJ world champion known for mastering fundamental techniques rather than flashy moves.

Eddie Bravo – Creator of 10th Planet Jiu-Jitsu, revolutionizing no-gi grappling with his rubber guard system.

Marcelo Garcia – Five-time world champion and ADCC gold medalist who pioneered modern butterfly guard techniques.

Kyra Gracie Guimarães – Five-time world champion who helped popularize women’s jiu-jitsu globally.

Andre Galvao – Multiple-time ADCC champion and founder of Atos Jiu-Jitsu, one of the most successful competition teams.

Jigoro Kano – Founder of Judo, which preserved many traditional jiu-jitsu techniques when traditional schools were declining.

Renner Gracie – Continuing the Gracie legacy as a respected instructor and competitor.

Gordon Ryan – Dominant modern competitor known for his technical innovations and submission prowess in no-gi competition.

Keenan Cornelius – Innovator who created the worm guard and other modern competition techniques.

Write me a front end for an app called Jiu-Jitsu Genie. This app will have python based functions, but does not overall need to be in python. I recommend streamlit because it is easy to deploy, but do whatever is easiest to display mermaid based flow charts. This app will use the helper functions I have included in the jiu-jitsu functions.py, genai.py, chatgpt.py, and movieai.py files. It will also use the master list located in masters.txt

I want this app to have a nice black and white GUI. I want the left hand side to have a selection bar which allows the user to choose between a number of functions that will be described in detail later. Across all of the functions, I want to have state variables which persist. Those variables are called “current\_image” which is a string that points to an imported image into the app , “current\_chat” which is a string containing the back and forth from the gracie\_talk function , current\_flowchart which is a mermaid flowchart object , current\_video which points to a video that the user can upload, and current\_attributes which is a string that is updated with the results of get\_atrributes every time it is called.

I want the app to have the following functions accessable from that left sidebar:

1. I want the app to have a function which is called “Position Image Recommendations”. This function have an input where the user can upload an image. If they don’t upload an image format, throw an error and request that they submit an image; the resultant file location will be called image\_location. The page will also have a text box that allows the user to input any jiu-jitsu position to a variable called “position\_variable”; the page will also will allow the user to select whether the ruleset is MMA from a dropdown option menu that provides true or false options, and puts the result into a variable called isMMA; the page will also allow the user to submit into a text field that is called “ideas” and the text submitted goes into a variable called “keywords”. If all variables aren’t chosen or included, do not proceed further and throw an error message. Once variables are all inputted, I want to call the function get\_attributes(image\_location, position\_variable) and the append the results onto the “keywords” variable. Then finally call generate\_grappling\_plan(image\_location, position\_variable, isMMA, keywords) and display the analysis using clean bullet point format inside of a bounded text box. Then update the state variable called current\_image to match image\_location.
2. I want the app to have a function called “master talk”. This function will have a drop down menu, where the user can select one of fifteen top jiujitsu masters from a list and assign that result to a variable called master\_info. The masters to be included are located in a file: masters.txt. I want there to be a conversation flow that scrolls with a text box at the bottom of the page for inputs. Then instantiate a variable called “instructions” with the following text: “you are the jiu-jitsu master <master\_info>. Have a conversation to me as this master and provide me troubleshooting help on my jiu-jitsu based on your fundamental principles of jiujitsu and notable successes.”. Create a variable called “prompt” and instantiate it with: “start a conversation to help me with my jiu-jitsu”. Then call Genai.generate\_text (prompt, master\_info). Place the results of the generate\_text call into the conversation window as the master in master\_info. And update the state variable called Current\_conversation with the conversation history. When the user submits any text after the first time, put that text into a string called next\_comment and call Genai.generate\_chat\_response (current\_conversation, and next\_comment). Whenever the conversation receives a response from generate\_text or generate\_chat\_response, update current\_conversation. At the bottom of the page, put a button called “generate FLOW”. When the button is pressed, I want to change the page selected to “FLOW chart generator”
3. A function called “FLOW chart generator.” This function will have a text box called “starting position” which takes in the string and puts it into a variable called position\_variable. If an image has already been uploaded to the app, when something is submitted into position\_variable, I want to call the function get\_atrributes(current\_image, position\_variable) and store the results in a variable called ”atrributes”. I want another dropdown box for “rules” with options: “Unified MMA” or “IBJJF”. When mma is selected, I want to make a boolean variable isMMA =true. If not, isMMA = false. Finally, I want a text field box titled “Ideas” and have it pull into a string called “ideas.” Then I want to call generate\_flow\_chart(attributes, position\_variable, isMMA, ideas) and put the resultant mermaid flow chart into a global variable called current\_flowchart. Then call display\_graph(current\_flowchart) in a way which would allow uaers to best zoom in and out on the flowchart. Below the graph, put a text box that allows the user to “choose a next move”. The results will be put into a string called “chosen\_next”. There will be a button that says “Flow” and when pressed it will search for the chosen\_next move in the flowchart as a node or a line. If not present, it will throw a jiujitsu inspired error to the user. If it is, it will call next\_move(current\_flowchart, attributes, position\_variable + chosen\_next, isMMA, ideas) and overrite current\_flowchart with the new mermaid option. Then display the new chart using Display\_graph(current flowchart)
4. A function called “Video Match analysis.” Using FFMPEG and movieai.py functions. When the user uploads a video, store it in the current\_video variable and display the video in the window. Then if no image has been uploaded, choose a single frame from the beginning of the video spin that off into the current\_image variable. Then call get\_attributes and store the results in the global variable current\_attributes. Take a selection of images from the video where the number of images is the length of the video in seconds divided by 3. Store them in a dataframe and then call generate\_image\_description for every image based on the jiujitsu master chosen. If none is chosen yet, choose john Danaher. Then display the results in bullet point format. Below the video.