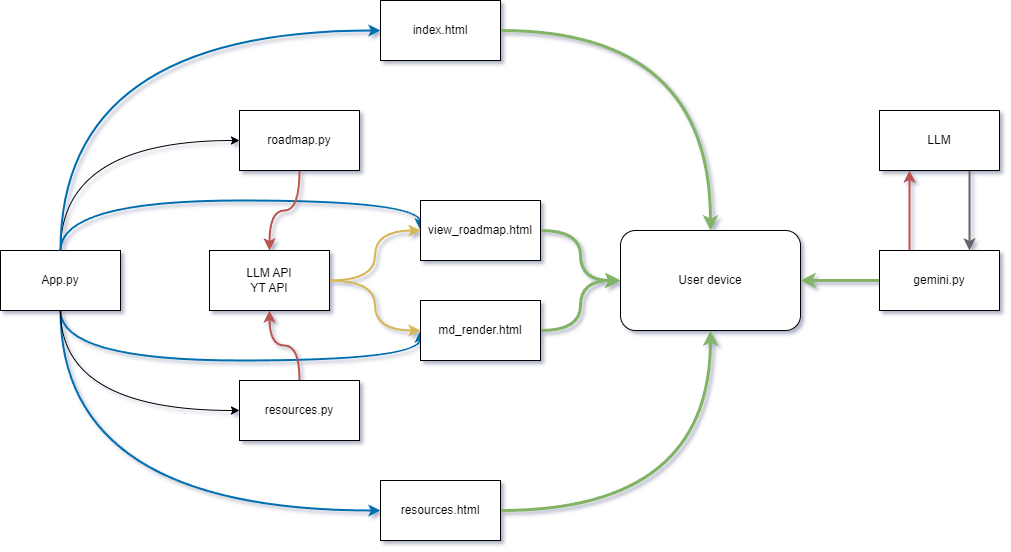
Code Documentation for PathFinder

The documentation format is as follows:

* File name
  + Description
  + Function name
    - Documentation
    - Future scope (If possible)
* Folder name
  + Sub-folders
  + Info
* \generated\_resources
  + Sub-folders: No sub-folders
  + Info: All the generated resource files (.md, .js) generated using the Gemini API and YouTube API is stored in this folder.
* \generated\_roadmaps
  + Sub-folders: No sub-folders
  + Info: All the generated roadmap files (.md, .js) generated using the Gemini API is stored in this folder.
* \static
  + Sub-folders: \css, \js
  + Info: Consists of static file (css and javascript) for the website in their respective directories.
* \templates
  + Sub-folders: \html
  + Info: It contains the html template files for the website.
* app.py
  + **Description**: One of the most important files for the project. It is responsible for routing events of the webpage to the specified file for API calls and read/write operations.
  + **Function**: parse\_md\_to\_json
    - **Documentation**: Parses the markdown format text to json format for easier usage in javascript.
    - **Future** **scope**: As of now it can parse markdown of very specific structure given the prompt to the LLM is also structured to match the requirements. Thus it is preferred to update the function to parse a wider variety of markdown text structure.
  + **Function:** index
    - **Documentation**: Manages the index html file when the website is started. It is responsible for taking user input and generating the roadmap by sending the API call to the LLM and it can take file input of previously generated roadmap and calls the necessary files to render it.
    - **Future scope:** As of now the front page is very basic, so in future it will be a good idea to make it more interactive and engaging for the user.
  + **Function:** resource
    - **Documentation:** makes the call to the python file `resources.py` to generate the resources and is also responsible for rendering the resource html page.
    - **Future scope**: Nothing that I can think of as of now.
  + **Function**: download
    - **Documentation**: It is responsible for downloading any file generated by the LLM to be downloaded in the specified directory based on the use like resource files goes to `generated\_resources` and roadmap files goes to `generated\_roadmap`.
    - **Future scope:** As of now it works just fine, so no changes needed unless major changes are made to the program structure.
  + **Function:** get\_roadmap\_json
    - **Documentation:** It retrieves the specified roadmap file from the `generated\_roadmaps` directory or returns a 404 error in case the file is not found.
    - **Future scope:** As of now a different function makes the call to generate the roadmap file. It will be better if both the function can be merged to reduce code complexity.
  + **Function:** get\_resource\_md
    - **Documentation:** It retrieves the specified resource files from the `generated\_resources` directory or returns a 404 error in case the files are not found.
    - **Future scope:** As of now a different function makes the call to generate the roadmap file. It will be better if both the function can be merged to reduce code complexity.
  + **Function:** get\_resource\_News\_json
    - **Documentation**: It retrieves the specified resource files related to the topic news from the `generated\_resources` directory or returns a 404 error in case the files are not found.
    - **Future scope:** As of now there are two different function doing the same job, just each of them are linked to two different buttons. In future if possible it will be good to merge the two functions together.
  + **Function:** get\_resource\_Tutorial\_json
    - **Documentation:** It retrieves the specified resource files related to the topic tutorial from the `generated\_resources` directory or returns a 404 error in case the files are not found.
    - **Future scope:** As of now there are two different function doing the same job, just each of them are linked to two different buttons. In future if possible it will be good to merge the two functions together.
  + **Function:** view\_roadmap
    - **Documentation:** It is responsible for rendering the roadmap page. It is linked to the submit button on the index page.
    - **Future scope:** There is no need of any change as of now.
  + **Function:** view\_md
    - **Documentation:** It is responsible for rendering the resource aggregation page. It is linked to the submit button on the resources page.
    - **Future scope:** There is no need of any change as of now.
  + **Function:** resources
    - **Documentation:** It is responsible for rendering the resources page. It is linked to the submit button on the index and roadmap viewer page.
    - **Future scope:** There is no need of any change as of now.
* **gemini.py**
  + **Description:** It loads the streamlit app which can be used to chat with Gemini using the Google Gemini API key.
  + **Function:** load\_custom\_css
    - **Documentation:** It is to make the UI more user friendly by adding a profile image for the user and the chat bot.
    - **Future scope:** There is a huge scope of improvement in the design of the streamlit app.
  + **Function:** call\_gemini\_api
    - **Documentation:** Everytime the `send` button is pressed, this function is called which use the input as the prompt and return the generated output from the LLM.
    - **Future scope:** Trying to use the image, video and other file uploading feature to increase it scope of usage.
* **resources.py**
  + **Description:** This file is responsible for generating resources markdown file using the Gemini API.
  + **Function:** generate\_resources
    - **Documentation:** It uses structured prompt format to send a structured pre-loaded prompt along with the new prompt to get a markdown file of the same format as of the pre-loaded prompts.
    - **Future scope:** The structure of the example prompts can be improved to get better results.
* **roadmap.py**
  + **Description:** This file is responsible for generating roadmaps markdown and json file using the Gemini API.
  + **Function:** parse\_response\_to\_json
    - **Documentation:** Parses the markdown format text to json format for easier usage in javascript.
    - **Future scope:** As of now it can parse markdown of very specific structure given the prompt to the LLM is also structured to match the requirements. Thus it is preferred to update the function to parse a wider variety of markdown text structure.
  + **Function**: generate\_roadmap
    - **Documentation:** It uses structured prompt format to send a structured pre-loaded prompt along with the new prompt to get a markdown file of the same format as of the pre-loaded prompts.
    - **Future scope:** The structure of the example prompts can be improved to get better results.
* YT.py
  + **Description**: This file is responsible for aggregating YouTube video links to be displayed on the resources html page.
  + **Function:** tutorial\_vid
    - **Documentation:** It makes a call to the YouTube API to gather video tutorials about the given topic.
    - **Future scope:** There doesn’t seem to be any need of change as of now.
  + **Function:** news\_vid
    - **Documentation:** It makes a call to the YouTube API to gather video news about the given topic.
    - **Future scope:** There doesn’t seem to be any need of change as of now.

**PathFinder Flowchart**

****

Blue arrows: Template rendering

Black arrows: File calls

Red arrows: API calls

Yellow arrows: Shared files for read/write

Green arrows: website/webapp display