Documentation

List of technique and their explanations.

1. **Get Method:**  
   The **get ()** method is used to avoid situations where the value is not defined. This method returns the value for the given key, if present in the dictionary. If not, then it will return None (if get() is used with only one argument).
   1. Login/logout Service  
      When user visits the page for first time it is welcome’s the user through welcome page and give option to login. For achieving this condition, a conditional code is written; if user login for first time it shows welcome page and navigate through email entering page than to application page. If user is previously logged in than it directly navigates application page. On welcome page there is a login button shaded in blue colour. So, user could able to define where to click.
   2. List driver  
      Note: Model: Here model is where it stores information of GPU “name, manufacture & date issued”. Firstly, model file name device.py has created and Boolean value has been given to it.  
      When user try search a GPU for specific features it fetches the data from database if the given value matches to data it represents that GPU.
2. **Boolean** values are the two constant objects False and True. They are used to represent truth values (although other values can also be considered false or true). ... But also **Python** has the **boolean**-able concept for every object, which is used when function **bool**([x]) is called.
   1. When user add device in datastore, and return to main page and user try to search a device through features search option it try to match the input data of user to database data and both the values match and it give true value than it is display on main page under device list.
3. **Redirect**: When called, it returns a response object and **redirects** the user to another target location with specified status code
   1. Whenever user add a device or update a device or visit add/update page and click on cancel or add or update button it will redirect it to the main page. In case of update once changes has been made and clicked on update button it will redirect it to main page or if user decides not to change anything and clicks on cancel button it redirect to main page. Else in new add device case if user add details and click on add button it will redirect to main or if user decides to add device later on and cancel button it will redirect user to main page.
4. **Post Method define:**   
   To submit data to be processed to the server.
   1. When user is on add device page and adds device name, drive name, API, vendor and the date. As well as set, it features, this method will send data to server and it will store in database as well.
5. **Len function:**  
   **len**() **function** is an inbuilt **function in Python** programming language that returns the **length** of the string. Syntax: **len**(string) Parameters: It takes string as the parameter.
   1. If user tries to add a device of same name which is already present in database it won’t allow it to add. This will help, not to create same device with different values.
6. **Description of all models:**
   1. myUser: This model is about user email.
   2. Device: This model describes about GPU data model in which it contains device name, drive name, API, vendor and the date**.** Features name eometryShader, tesselationShader, shaderInt16, sparseBinding, textureCompressionETC2, vertexPipelineStoresAndAtomics
   3. Adddriver: This model defines that adding a new device and user can add details and select it features.
   4. Update: This model is about updating an existing device features only, it doesn’t allow user to change device name of another details expect its features.
   5. Showdetails: This model is about to show details of existing device and allowing user to update it feature wise only.
7. **UI design decision:**  
   First of all, I have kept login page simple because it will be helpful to user to understand how to proceed. After login page there is an email addition page. For this I have used a blue box and email address space, it has been designed this way because of simplicity of the entire application and user need add only one detail. After that it navigates to application page where UI start presenting welcome message and user email address displays just beside to the welcome message. Below that there is a Add devices option with **click here** button in bold and bigger font than rest of the text on the page, this help user to define to how to add new devices. On the add device page it gives user to add device name, drive name, API, vendor and the date which display when it has been add in database. This help to check on which date has been added help user to remember it. As well as it allows user to select features of the device by toggling the checkbox. In this way if user in future wants to search a device that it has been added, it will help them to search the exact device and if user want it can update it features also. This give user the convenience of updating a device, when device has a minor or major update in it. On main page there is device search option through specific checkbox toggled features, this help user to find an exact device with the enable feature it was looking for. A added device list is display in a manner that it shows in a table which has a tile of added device in blue colour font. Which contain device name, driver name, API and vendor name in black colour font and which help user to where find added devices. This table has blue colour scheme. Below this add devices list table there is a logout here button for user in blue shade.