Data Upload Module:

- Description: In this module, the endpoint will allow the user to upload images for their entire dataset. This is assuming that the user will have the same image type and image name corresponding to the images and label.
- Endpoint: '/upload'
- Method: POST
- Request parameters:
    - Images: A folder containing image files, with each filename is just a number. Ex: the first image will be named "1.png", then the second image will be "2.png" and so on. This would be their image ID.
    - Labels: A folder containing labels for the corresponding image files. The way the folder should work is to have the exact images in the images folder, but their name would contain the image ID number, followed by its label name and separated by a specific character.
- Response:
    - '201 OK': Upload successful.
    - '401 Bad Request': If there are any issues with the requests.

Example request:

```
upload_data = {
   "images": ["1.png", "2.png"],
   "labels": ["1 : Lionel Messi", "2 : LeBron James"]
}

response = requests.post("http://api-url/upload")
```

Training Data Module:

- Description: In this module, the endpoint will start the process for the image classification model training using the uploaded data. It should split the uploaded dataset into training and testing data. After running the training configurations, it will return a data container storing the model's parameters (weights and biases) that have been tuned, then it will apply those parameters onto the test dataset and calculate the statistics for it (error/loss, accuracy, etc.)
- Endpoint: '/train'
- Method: POST
- Request parameters:

    - Train-Test Split: Specify the percentages for training and testing data split. Optional: Validation split.
    - Regularization term (optional)
    - Loss function

- Example of usage:

  ```
  response = requests.post("http://api-url/train")
  ```