**Static file connection:**

**CSS file:**

    <link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename="styles.css") }}">

**JAVASCRIPT:**

    <script type="text/javascript" src="{{ url\_for('static', filename= "script.js") }}" ></script>

**fhjj ffdr ksnu mzbq**

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

**Flask Flash Messages**

 {% with messages = get\_flashed\_messages() %}

    {% if messages %}

        <ul>

        {% for message in messages %}

            <li>{{ message }}</li>

        {% endfor %}

        </ul>

    {% endif %}

    {% endwith %}

    {% for error in form.name.errors %}

        <div class="error">{{ error }}</div>

    {% endfor %}

**For categories without form**

{% with messages = get\_flashed\_messages(with\_categories=true) %}

      {% if messages %}

        <ul>

          {% for category, message in messages %}

            {% if category == 'success' %}

              <li class="success">{{ message }}</li>

            {% elif category == 'error' %}

              <li class="error">{{ message }}</li>

            {% endif %}

          {% endfor %}

        </ul>

      {% endif %}

    {% endwith %}

**Flask Flash Messages with cagtegory**

{% with messages = get\_flashed\_messages(with\_categories=true) %}

    {% if messages %}

        <ul>

        {% for message, category in messages %}

            {% if category == 'success' %}

                <li class="success">{{ message }}</li>

            {% elif category == 'error' %}

                <li class="error">{{ message }}</li>

            {% elif category == 'warning' %}

                <li class="warning">{{ message }}</li>

            {% else %}

                <li>{{ message }}</li>

            {% endif %}

        {% endfor %}

        </ul>

    {% endif %}

{% endwith %}

Ebooks **:** <https://github.com/tecladocode/rest-apis-flask-python>

***DOCKER***

To upload image to docker (***docker build -t rest-apis-with-flask .***) . means the floder where to you want to copy.

If you want to run docker file from cmd (***docker run -p 5005:5000 rest-apis-with-flask***) 5005 means the port on which you want to run the app.py

To know ID(***docker run -dp 5005:5000 rest-apis-with-flask***)

***docker run -dp 5000:5000 -w /app -v "/c/Documents/yourproject:/app" flask-smorest-api***

GENERATE SECRET KEY :

  app.config["JWT\_SECRET\_KEY"] =  secrets.SystemRandom().getrandbits()

python

import secrets

secrets.SystemRandom().getrandbits(128)

Store ID:

from flask import Flask, request, jsonify

import uuid

app = Flask(\_\_name\_\_)

stores = {} # Initialize an empty dictionary to store stores

@app.route('/store', methods=['POST'])

def create\_store():

store\_data = request.get\_json() # Get store data from request JSON

store\_id = uuid.uuid4().hex # Generate a unique ID for the store

store = {\*\*store\_data, "id": store\_id} # Merge store data with ID

stores[store\_id] = store # Add the store to the stores dictionary

return jsonify(store), 201 # Return the created store as JSON with status code 201

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True )

**GIT COMMANDS :**

<https://git-workshop.tecladocode.com/docs/what_is_a_commit>

<https://git-workshop.tecladocode.com/>

<https://git-workshop.tecladocode.com/docs/what_is_git_repository>

These all three links are indicate to Git Crash Commands which is extremely used in Git.

To commit or restore.