

Isko GLAM Documentation

(<http://jcprng.pythonanywhere.com/>)

Description

Isko GLAM is an online wiki that features the GLAMs of the University of the Philippines. “GLAMs” is short for Galleries, Libraries, Archives, and Museums, these are the four main types of information centers. In this wiki, users can navigate to find out more about the GLAMs within the campus, know where they are located, and even add more to the wiki.

Features

To properly inform users about the GLAM, each wiki entry of the registered place in this website must contain the following information:

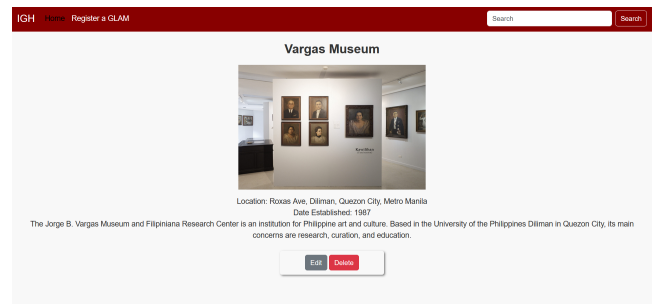
- Type of GLAM
- Name of GLAM
- Date Established
- Location
- Image of place
- Description

Sample Pictures

Homepage



GLAM entry



Register

The screenshot shows the registration form for a GLAM. The header is dark red with 'IGH' and 'Register a GLAM' links. A search bar is on the right. The main content area has a title 'Register a Gallery, Library, Archive, or Museum to the Hub'. Below the title is a dropdown menu for 'Type of GLAM'. The form fields are: 'Name of GLAM', 'Date Established', 'Location', 'Image URL', and a text area for 'Tell something about the GLAM...'. A 'Register Now!' button is at the bottom.

app.py

```
1 from flask import Flask, request, jsonify, render_template
2 app = Flask(__name__)
3
4 # Routes
5 @app.route("/")
6 def index():
7     return render_template("index.html")
8
9 # Add route to list of GLAMs
10 @app.route("/list-glam")
11 def list_glam():
12     return render_template("list_glam.html")
13
14 # Add route to add new GLAM
15 @app.route("/add-glam")
16 def add_glam():
17     return render_template("add_glam.html")
18
19 # Add route to edit GLAM
20 @app.route("/edit-glam")
21 def edit_glam():
22     return render_template("edit_glam.html")
23
24 # Register new GLAM
25 @app.route("/register")
26 def register():
27     return render_template("register.html")
28
29 # Process form data
30 @app.route("/process", methods=["POST"])
31 def process():
32     data = request.get_json()
33     name = data.get("name")
34     type = data.get("type")
35     location = data.get("location")
36     description = data.get("description")
37     url = data.get("url")
38     # Insert into database
39     insert_glam(data)
40     return jsonify({"message": "GLAM added successfully"})
41
42 # Modify existing GLAM
43 @app.route("/modify", methods=["POST"])
44 def modify():
45     data = request.get_json()
46     name = data.get("name")
47     type = data.get("type")
48     location = data.get("location")
49     description = data.get("description")
50     url = data.get("url")
51     # Update in database
52     update_glam(data)
53     return jsonify({"message": "GLAM updated successfully"})
```

This code uses the Flask application that provides routes for the listed GLAM by type, registering new GLAMs, and modifying the existing GLAMs.

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>GLAMs</title>
6 </head>
7 <body>
8 <div class="container">
9 <h1>GLAMs</h1>
10 <div class="row">
11 <div class="col">
12 <h2>List GLAMs</h2>
13 <div class="list">
14 <div class="item">
15 <div class="name">Name</div>
16 <div class="type">Type</div>
17 <div class="location">Location</div>
18 <div class="description">Description</div>
19 <div class="url">URL</div>
20 </div>
21 </div>
22 </div>
23 </div>
24 </div>
25 </div>
26 </div>
27 </div>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
```

This code is under the template file that extends from the “base.html”. It has the CSS codes for the styling of the webpage.

data.py

```
1 import sqlite3
2
3 # Database connection
4 db = sqlite3.connect("glam.db")
5
6 # Create table
7 db.execute("CREATE TABLE IF NOT EXISTS glam (name TEXT, type TEXT, location TEXT, description TEXT, url TEXT)")
8
9 # Insert data
10 def insert_glam(data):
11     name = data.get("name")
12     type = data.get("type")
13     location = data.get("location")
14     description = data.get("description")
15     url = data.get("url")
16     db.execute("INSERT INTO glam VALUES (?, ?, ?, ?, ?)", (name, type, location, description, url))
17     db.commit()
18
19 # Update data
20 def update_glam(data):
21     name = data.get("name")
22     type = data.get("type")
23     location = data.get("location")
24     description = data.get("description")
25     url = data.get("url")
26     db.execute("UPDATE glam SET name=?, type=?, location=?, description=?, url=? WHERE name=?", (name, type, location, description, url, name))
27     db.commit()
28
29 # Delete data
30 def delete_glam(name):
31     db.execute("DELETE FROM glam WHERE name=?", (name,))
32     db.commit()
33
34 # Get data
35 def get_glam():
36     db.execute("SELECT * FROM glam")
37     data = db.fetchall()
38     return data
39
40 # Close connection
41 db.close()
```

This code provides the interaction with the SQLite database where the information about GLAMs is stored.