Assignment Code: DA-AG-004

Restful API & Flask | **Assignment**

Instructions: Carefully read each question. Use Google Docs, Microsoft Word, or a similar tool to create a document where you type out each question along with its answer. Save the document as a PDF, and then upload it to the LMS. Please do not zip or archive the files before uploading them. Each question carries 20 marks.

Total Marks: 200

Question 1 : What is a RESTful API?

Answer:

A RESTful API is an application programming interface that follows the principles of REST (Representational State Transfer). It uses HTTP to allow clients and servers to communicate, typ exchanging data in JSON or XML format. REST emphasizes stateless communication, resource-based URLs, and standard HTTP methods like GET, POST, PUT, and DELETE.

Question 2: What is Flask, and why is it popular for building APIs?

Answer:

Flask is a lightweight and flexible Python web framework used to build web amework applications a APIs. It is popular because it is easy to learn, highly customizal has a simple structure, and supports extensions for features like authentication, database interaction, and serialization.

Question 3:What are HTTP methods used in RESTful APIs?

Answer:

Common HTTP methods in RESTful APIs include:

GET - Retrieve data

PUT - Update existing data

PATCH - Modify part of existing data PATCH - Modify part of existing data methods help perform (
(Create, Read, Update, Delete) operati

Question 4: What is the purpose of the @app.route() decorator in Flask?

Answer:

The @app.route() decorator maps a URL to a Python function. When a user visits that URL, Flas
runs the associated ed function. It is used to define API endpoints and decide what HTTP meth
are allowed for them

Question 5: What is the role of Flask-SQLAlchemy?

Answer:

Flask-SQLAlchemy is an extension that simplifies database management in Flask applications. It provides ORM (Object Relational Mapping) capabilities, allowing developers to interact with databases using Python classes instead of SQL queries.

Question 6: How do you create a basic Flask application?

Answer:

```
from flask import Flask app = Flask( nar

@app.route("/")

def home():

return "Hello World"

if name ==" main ":

app.run(debug=True)
```

Question 7:How do you return JSON responses in Flask?

Answer:

```
from flask import jsonify

@app.route("/data")
def data():
    return jsonify({"message": "Success", "status": 20
```

Question 8:How do you handle POST requests in Flask?

Answer:

```
from flask import request

@app.route("/submit", methods=["POS
def submit():
   data = request.json
   return jsonify({"received": data})
```

Question 9: How do you handle errors in Flask (e.g., 404)?

Answer:

```
@app.errorhandler(404)
def not_found(e):
   return jsonify({"error": "Page Not Found"}),
```

Question 10: How do you structure a Flask app using Blueprints?

Answer:

```
from flask import Blueprint

user_bp = Blueprint("user", __name_

@user_bp.route("/user")
def get_user():
   return "User Page"
```