**# Import Flask class and jsonify & request functions from flask module**

from flask import Flask, jsonify, request

**# Create a Flask application instance**

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

**# Sample task data stored in a list of dictionaries**

tasks = [

{

'id': 1,

'title': 'Task 1',

'description': 'This is task 1',

'done': False

},

{

'id': 2,

'title': 'Task 2',

'description': 'This is task 2',

'done': False

}

]

**# GET request to fetch all tasks**

**'''Route for GET request to /get-all-tasks endpoint.**

**Returns JSONified tasks data.'''**

@app.route('/get-all-tasks', methods=['GET'])

def get\_tasks():

return jsonify({'tasks': tasks})

**# GET request to fetch a specific task by its id**

**'''Route for GET request to /get-task/<id> endpoint**

**Returns task with given id if found, 404 status if not found'''**

@app.route('/get-task/<int:task\_id>', methods=['GET'])

def get\_task(task\_id):

task = next((task for task in tasks if task['id'] == task\_id), None)

if task:

return jsonify({'task': task})

else:

return jsonify({'message': 'Task not found'}), 404

**# POST request to create a new task**

**'''Route for POST request to /create-task endpoint**

**Gets task data from request body, creates new task & adds it to tasks list**

**Returns 201 status code for created '''**

@app.route('/create-task', methods=['POST'])

def create\_task():

data = request.json

new\_task = {

'id': len(tasks) + 1,

'title': data['title'],

'description': data['description'],

'done': False

}

tasks.append(new\_task)

return jsonify({'message': 'Task created successfully', 'task': new\_task}), 201

**# PUT request to update a task**

**'''Updates task if found with data from request body**

**Returns 404 status if task not found'''**

@app.route('/update-task/<int:task\_id>', methods=['PUT'])

def update\_task(task\_id):

task = next((task for task in tasks if task['id'] == task\_id), None)

if task:

data = request.json

task.update(data)

return jsonify({'message': 'Task updated successfully', 'task': task})

else:

return jsonify({'message': 'Task not found'}), 404

**# DELETE request to delete a task**

**'''Deletes task from list if found**

**Returns success message'''**

@app.route('/delete-task/<int:task\_id>', methods=['DELETE'])

def delete\_task(task\_id):

global tasks

tasks = [task for task in tasks if task['id'] != task\_id]

return jsonify({'message': 'Task deleted successfully'})

if \_\_name\_\_ == '\_\_main\_\_':

app.run()