CODTECH Internship - Task 2

RESTful API Development

Instructions:

1. Design a RESTful API for a library or inventory system.

2. Implement CRUD operations (Create, Read, Update, Delete).

3. Deliverable: API documentation and code with functional endpoints.

Below is a sample implementation using Python (Flask).

1. app.py (Flask REST API)

```
from flask import Flask, jsonify, request

app = Flask(__name__)

# Sample data for a library system
books = [
    {"id": 1, "title": "Book One", "author": "Author A"},
    {"id": 2, "title": "Book Two", "author": "Author B"}
]

# Get all books
@app.route('/books', methods=['GET'])
def get_books():
    return jsonify(books)

# Get a book by ID
@app.route('/books/<int:id>', methods=['GET'])
def get_book(id):
    book = next((b for b in books if b['id'] == id), None)
    return jsonify(book) if book else ('', 404)

# Add a new book
@app.route('/books', methods=['POST'])
def add_book():
    new_book = request.json
    new_book['id'] = len(books) + 1
    books.append(new_book)
    return jsonify(new_book), 201

# Update a book
@app.route('/books/<int:id>', methods=['PUT'])
def update_book(id):
    book = next((b for b in books if b['id'] == id), None)
    if not book:
        return ('', 404)
    data = request.json
    book.update(data)
    return jsonify(book)

# Delete a book
@app.route('/books/<int:id>', methods=['DELETE'])
def delete_book(id):
    global books
    books = [b for b in books if b['id'] != id]
    return ('', 204)

if __name__ == '__main__':
    app.run(debug=True)
```

2. API Endpoints Documentation

Base URL: http://localhost:5000

1. GET /books

   - Fetch all books.

2. GET /books/<id>

   - Fetch a book by ID.

3. POST /books

   - Add a new book.

   - Body: {"title": "Book Name", "author": "Author Name"}

4. PUT /books/<id>

   - Update a book's details.

   - Body: {"title": "New Title", "author": "New Author"}

5. DELETE /books/<id>

   - Remove a book by ID.