CS301-Software Engineering – Class Practice Sessions – 1

21bcs043 – HETH THARUN KOORMA

Theme: Create new cultural destination to celebrate the heritage of India and provide a platform for emerging Talents using Digital Technology solutions

Aim:

- Creating doors for a first-of-its-kind, multi-disciplinary space for the Arts in cities
- Encourage Visual art space and captivating array of public art
- Bring together communities through a dynamic programming of epic theatricals, regional theatre, music, dance, spoken word etc.
- Major attraction is to provide a platform for emerging talent and showcases the vibrance of India's heritage
- Generate source of income for the Art communities through collaborations, aggregators, and accelerators investments

Target audiences:

- Home to Art, Artists, the audience from India and around the world.

Assignment scope:

1. Identify various requirements for the above program initiative that can be developed as a digital solution
2. Use ChatGPT platform an generate code for the above requirements
    a. Generate code and run the program in Google Colab/Jupiter Notebook/Visual Code/PyCharm
    b. Perform integrated testing. Add integration testing code in the same program.

Modify the same program. Write APIs to access the data from the public domain and test the program for regression testing the same program

Deliverable: (DIGITAL KIOSKS' CODE WITH TESTING CODE)

Here I have proposed of creating digital kiosk in public places which can generate tickets. Such kiosks can be publicised using various methods of advertising and marketing which generates more people buying the tickets rather than having a website or an app for it. This makes it easier to reach the people that would not themselves get on the app or website to buy the ticket and would make a bigger impression on them like that of a vending machine which would make them buy the tickets to the shows and exhibitions.

CODE:

(User Code)

```
import requests
import json

API_URL = "http://localhost:5000"

def get_events():
    response = requests.get(API_URL + "/events")
    if response.status_code == 200:
        return response.json()
    else:
        raise Exception("Failed to retrieve events")

def buy_ticket(event_id):
    payload = {'event_id': event_id}
    response = requests.post(API_URL + "/buy-ticket", json=payload)
    if response.status_code == 200:
        return response.json()
    else:
        raise Exception("Failed to buy ticket")

def test_get_events():
    events = get_events()
    assert type(events) == dict, "get_events() did not return a dictionary"
    assert len(events) > 0, "get_events() returned an empty dictionary"
    print("get_events() test passed")

def test_buy_ticket():
    events = get_events()
    assert len(events) > 0, "get_events() returned an empty dictionary"
    event_id = list(events.keys())[0]
    result = buy_ticket(event_id)
    assert result['success'] == True, "buy_ticket() was not successful"
    print("buy_ticket() test passed")

if __name__ == '__main__':
    test_get_events()
    test_buy_ticket()
    print("all tests passed.")
```

(API Code)

```
from flask import Flask, jsonify, request

app = Flask(__name__)

# Define some example events
events = {
    "event1": {"name": "Concert", "date": "2022-05-01", "tickets_available": 100},
    "event2": {"name": "Theater", "date": "2022-06-15", "tickets_available": 50},
    "event3": {"name": "Comedy Show", "date": "2022-07-20", "tickets_available": 75}
}

@app.route('/events', methods=['GET'])
```

```python
def get_events():
    return jsonify(events)

@app.route('/buy-ticket', methods=['POST'])
def buy_ticket():
    data = request.get_json()
    event_id = data['event_id']
    if event_id in events:
        if events[event_id]['tickets_available'] > 0:
            events[event_id]['tickets_available'] -= 1
            return jsonify({'success': True})
        else:
            return jsonify({'success': False, 'error': 'Sold out'})
    else:
        return jsonify({'success': False, 'error': 'Event not found'})

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

OUTPUT:

```
"C:\Users\LENOVO\projects\SE - testing\venv\Scripts\python.exe" "C:\Users\LENOVO\projects\SE - testing\SE_21BCS043_User.py"
get_events() test passed
buy_ticket() test passed
all tests passed.

Process finished with exit code 0
```