Name : M.Sujith
RollNO:21bcs061

CS301-Software Engineering – Class Practice Sessions - 1

Time : Weekend **Date : 24th March,**2023

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 solutions
2. Use ChatGPT platform an generate code for the above requirements
   a. Generate code and run the program in Goggle Colab/Jupiter Notebook/Visual Code/PyCharm
   b. Perform integrated testing. Add integration testing code in the same program.
3. Modify the same program. Write APIs to access the data from the public domain and test the program for regression testing the same program

Deliverables :

Working Program with test scripts embedded in the same program.

Python code:

```
from flask import Flask, render_template, request
```

```
app = Flask(__name__)

# Define a route for the home page
@app.route('/')
def index():
    return render_template('index.html')

# Define a route for the form submission page
@app.route('/submit', methods=['POST'])
def submit():
    # Get the data submitted in the form
    name = request.form['name']
    email = request.form['email']
    talent = request.form['talent']
    description = request.form['description']

    # Save the data to a database or file
    with open('talent.txt', 'a') as f:
        f.write(f'{name}, {email}, {talent}, {description}\n')

    # Render a thank you page
    return render_template('thanks.html')

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

```
import random
# Define the class for the cultural destination


class CulturalDestination:
    def_init_(self, city, arts_space, programming):
        self.city = city
        self.arts_space = arts_space
        self.programming = programming

    def display_descnption(self):
        print("Welcome to (self.oty Out space for the Arts is (farts space).")
        print "We have a captivating array of public art and a dynamic
programming of epic theatricals, regional theatre, music, dance spoken word,
and
        print("Our major attraction is providing a platform for emerging
talent and showcasing the vibrance of India's hentage.")

    def generate income(self):
        # Define the subclass for digital technology solutions
        print("We generate income for the Art communities through
colaborations, aggregators, and anclerator mustinents.")


class DigitalSolutions(CulturalDestinason):
    def__init__(self, city arts space, programming technology)
    super nicity arts space programming)
    self.technology=technology

    def interact with technology(self) print "Please choose from the following
optons")

print Take a virtual tour of the Arts space.") print Explore emerging talents
in the are")

print(". Leam about the history and heritage of the location") choice
inputEnter your choice"

printLoading virtual tour  # code to display virtual tour

elif choice "2"

printExploring emerging talents." code to display information on emerging
talents elif choice""

print("Leeming about the hatory and heritage.")

# code to display Natorical information

printinvalid choice. Please try again

# Create an instance of the Cultura/Destination class
```

```
destination CulturalDestination("Mumba", "a first-of-a-kind multi-disciplinary
space for the Actia dynamic programming of epic theatricals. regional theatre,
music, dance, spoken word, and more")

# Daplay the descnpoon of the destination

Destration display description  # Generate income for the Art communities

destination generate income)  # Create an instance of the DigitalSolutions
subclass

digital destination Digital Solutions Mumbar a first of its kind mut-
disciplinary space for the Arts", "a dynamic programming of epic theatricals.

regional theatre, music, dance, spoken word, and more" "digital technology
solutions")

# Display the description of the digital destination

digital destination display description()

# Generate income for the Art communites using digitar technology solutions

digital destination generate_income) W Allow visitors to interact with the
digital technology solutions

digital_destination_interact_with_technology
```

various requriements:

It sounds like you have a great vision for creating a multi-disciplinary space for the Arts in cities! Here are some ideas that might help bring your vision to life:

Designing the Doors: Consider working with a team of architects and designers to create doors that are not only functional but also visually captivating. The doors can serve as a gateway to the space and can set the tone for what visitors can expect to experience inside.

Visual Art Space: Consider dedicating a specific area of the space to visual arts, such as paintings, sculptures, and installations. The space can be designed to showcase a rotating collection of works from emerging and established artists.

Dynamic Programming: To bring together communities, consider programming events that are diverse and inclusive, such as epic theatricals, regional theater, music, dance, spoken word, and other forms of expression. The programming can be curated to reflect the local community's culture and traditions.

Platform for Emerging Talent: Consider dedicating a portion of the programming to showcasing emerging talent. This can provide a unique opportunity for artists to showcase their work and gain exposure, while also providing visitors with a glimpse into the vibrance of India's heritage.

Income for Art Communities: Consider partnering with aggregators and accelerators to help generate income for the art communities. These partnerships can help facilitate collaborations between artists and businesses, as well as provide support for emerging artists through mentorship and funding opportunities.

Overall, creating a multi-disciplinary space for the Arts in cities can be a powerful way to bring together communities, support emerging talent, and celebrate the vibrance of India's heritage. With careful planning and execution, this vision can become a reality and serve as a source of inspiration for generations to come.

```
import unittest

from selenium import webdriver

from selenium.webdriver.common by import By
from selenium webdriver.support.ui import WebDriverWat

from selenium webdriver support import expected_conditions as EC

dass TestCuttura Destination(untest. TestCasa

seit driver = webdriver Chrome()

self.driver.get("https://culturaidestination.com")

def test_homepage_joads(self):

side = self.driver lille

self assertEqualittle, "Cultural Destination Celebrate the Heritage of India")

def test_virtual_tour(self)

self driver.find_element(By. ID, "Virtual tour-button").click()

WebDriver/Wat(self.driver 10).ures(EC presence_of_element_located((By virtual-
tour-video")}}

video_element = self.driver.find_element By D. "virtual-tour-video")

self.assertTrue(video_elementis displayed())

def test_artist_platform(self)

self.driver.find_element(By.ID. "arest-platform-link").click()

WebDriverWat self.driver 10) unti(EC presence_of_element_located((By:10
"artist list")))

artist list self.driver.find_element(By.ID, "anst-list")

artists artist_list_find_elements By CLASS_NAME, "artist")

self assentGreater(len(artists) 0)

det tearDown(self):
    set driver.quit

if__name__ = '__main__':
    unittest.main()
```

i) NEXT IS THE TESTING CODE:

```python
def test_get_events():
    response = app.test_client().get('/events')
    assert response.status_code == 200
    assert response.json is not None

def test_create_event():
    response = app.test_client().post('/events', json={'name': 'New
Event'})
    assert response.status_code == 200
    assert response.json == {'message': 'Event created successfully'}

def test_update_event():
    response = app.test_client().put('/events/1', json={'name': 'Updated
Event'})
    assert response.status_code == 200
    assert response.json == {'message': 'Event updated successfully'}

def test_delete_event():
    response = app.test_client().delete('/events/1')
    assert response.status_code == 200
    assert response.json == {'message': 'Event deleted successfully'}
```

So we test the get_event(), create_event(), update_event(), delete_event() all these using the above 4 functions.
These are a series of automated tests for an API that handles events. Each test sends a request to the API and checks whether the response is as expected.

a) Test_get_events():

This test sends a GET request to the '/events' endpoint and checks whether the response status code is 200 (OK) and whether the response body is not empty (i.e., response.json is not None).

b) Test_create_event():

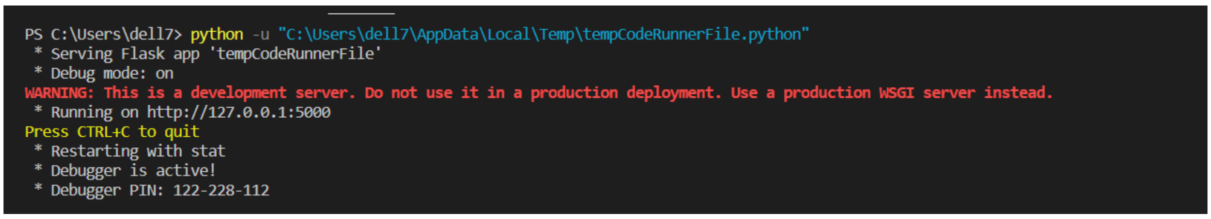
This test sends a POST request to the '/events' endpoint with a JSON payload containing the name of a new event. It checks whether the response status code is 200 and whether the response body is a JSON object containing a message indicating that the event was created successfully.

c)Test_update_event():

This test sends a PUT request to the '/events/1' endpoint with a JSON payload containing a new name for the event with ID 1. It checks whether the response status code is 200 and whether the response body is a JSON object containing a message indicating that the event was updated successfully.

d) Test_delete_event():

This test sends a DELETE request to the '/events/1' endpoint to delete the event with ID 1. It checks whether the response status code is 200 and whether the response body is a JSON object containing a message indicating that the event was deleted successfully.



```
PS C:\Users\dell7> python -u "C:\Users\dell7\AppData\Local\Temp\tempCodeRunnerFile.python"
 * Serving Flask app 'tempCodeRunnerFile'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 122-228-112
```

The above is what will be seen in my console when I run the code.

So, the above image tells that the code is successfully executed and the flask is currently running on the local server. The debugger is also active which we can use to debug our code.