

ASSIGNMENT 7

Microservices Framework.

- Priyanka Gowda
BE IT B
47023

CODE:

Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <title>INDEX</title>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
  </head>
  <body>
    <a href="http://127.0.0.1:5000/users">Click here to view user data </a><br />
    <a href="http://127.0.0.1:5000/movies">Click here to view movies data </a><br />
    <a href="http://127.0.0.1:5000/showtimes">Click here to view showtimes data </a><br />
    <a href="http://127.0.0.1:5000/booking">Click here to view booking data </a><br />
  </body>
</html>
```

script.py

```
from flask import Flask
from flask import jsonify, json
import pandas as pd

app = Flask(__name__)

@app.route('/')
def index():
    return "HELLO WORLD"

@app.route('/users')
def disp_users():
    data = pd.read_json('users.json')
    data = data.values.tolist()
    return jsonify(data)

@app.route('/movies')
def disp_movies():
```

```

data_movies = pd.read_json('movies.json')
data_movies = data_movies.values.tolist()
return jsonify(data_movies)

@app.route('/booking')
def disp_booking():
    data_booking = pd.read_json('booking.json')
    data_booking = data_booking.values.tolist()
    return jsonify(data_booking)

@app.route('/showtimes')
def disp_showtimes():
    data_showtimes = pd.read_json('showtimes.json')
    data_showtimes = data_showtimes.values.tolist()
    return jsonify(data_showtimes)

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

```

booking.json

```

{
  "booking": [
    {
      "movie-name": "Elephant X",
      "user-name": "A",
      "screen": 1,
      "Start-time": "10:00"
    },
    {
      "movie-name": "Elephant X",
      "user-name": "B",
      "screen": 1,
      "Start-time": "10:00"
    },
    {
      "movie-name": "Sherlock",
      "user-name": "C",
      "screen": 1,
      "Start-time": "13:30"
    },
    {
      "movie-name": "Antman",
      "user-name": "D",
      "screen": 2,
      "Start-time": "17:50"
    },
  ],
}

```

```
{
  "movie-name": "MNOP",
  "user-name": "E",
  "screen": 2,
  "Start-time": "9:30"
}
]
```

movies.json

```
[
  {
    "title": "The Shawshank Redemption",
    "rank": "1",
    "id": "tt0111161"
  },
  {
    "title": "The Godfather",
    "rank": "2",
    "id": "tt0068646"
  },
  {
    "title": "Elephant X",
    "rank": "3",
    "id": "tt0071562"
  },
  {
    "title": "Pulp Fiction",
    "rank": "4",
    "id": "tt0110912"
  },
  {
    "title": "In The Woods of Romania",
    "rank": "5",
    "id": "tt0060196"
  },
  {
    "title": "The Dark Knight",
    "rank": "6",
    "id": "tt0468569"
  },
  {
    "title": "12 Angry Men",
    "rank": "7",
    "id": "tt0050083"
  },
  {
```

```

    "title": "Sherlock",
    "rank": "8",
    "id": "tt0108052"
  },
  {
    "title": "Antman",
    "rank": "9",
    "id": "tt0167260"
  },
  {
    "title": "Fight Club",
    "rank": "10",
    "id": "tt0137523"
  }
]

```

showtimes.json

```

{
  "showtimes": [
    {
      "movie-name": "Elephant X",
      "screen-no": "1",
      "Start-time": "10:00"
    },
    {
      "movie-name": "Sherlock",
      "screen-no": "1",
      "Start-time": "13:30"
    },
    {
      "movie-name": "In The Woods of Romania",
      "screen-no": "2",
      "Start-time": "12:15"
    },
    {
      "movie-name": "Antman",
      "screen-no": "2",
      "Start-time": "17:50"
    }
  ]
}

```

users.json

```

[
  {
    "id": 1,
    "name": "Leanne Graham",

```

```
"username": "Bret",
"email": "Sincere@april.biz",
"address": {
  "street": "Kulas Light",
  "suite": "Apt. 556",
  "city": "Gwenborough",
  "zipcode": "92998-3874",
  "geo": {
    "lat": "-37.3159",
    "lng": "81.1496"
  }
},
"phone": "1-770-736-8031 x56442",
"website": "hildegard.org",
"company": {
  "name": "Romaguera-Crona",
  "catchPhrase": "Multi-layered client-server neural-net",
  "bs": "harness real-time e-markets"
}
},
{
  "id": 2,
  "name": "Ervin Howell",
  "username": "Antonette",
  "email": "Shanna@melissa.tv",
  "address": {
    "street": "Victor Plains",
    "suite": "Suite 879",
    "city": "Wisokyburgh",
    "zipcode": "90566-7771",
    "geo": {
      "lat": "-43.9509",
      "lng": "-34.4618"
    }
  },
  "phone": "010-692-6593 x09125",
  "website": "anastasia.net",
  "company": {
    "name": "Deckow-Crist",
    "catchPhrase": "Proactive didactic contingency",
    "bs": "synergize scalable supply-chains"
  }
},
{
  "id": 3,
  "name": "Clementine Bauch",
  "username": "Samantha",
  "email": "Nathan@yesenia.net",
```

```
"address": {
  "street": "Douglas Extension",
  "suite": "Suite 847",
  "city": "McKenziehaven",
  "zipcode": "59590-4157",
  "geo": {
    "lat": "-68.6102",
    "lng": "-47.0653"
  }
},
"phone": "1-463-123-4447",
"website": "ramiro.info",
"company": {
  "name": "Romaguera-Jacobson",
  "catchPhrase": "Face to face bifurcated interface",
  "bs": "e-enable strategic applications"
},
{
  "id": 4,
  "name": "Patricia Lebsack",
  "username": "Karianne",
  "email": "Julianne.OConner@kory.org",
  "address": {
    "street": "Hoeger Mall",
    "suite": "Apt. 692",
    "city": "South Elvis",
    "zipcode": "53919-4257",
    "geo": {
      "lat": "29.4572",
      "lng": "-164.2990"
    }
  },
  "phone": "493-170-9623 x156",
  "website": "kale.biz",
  "company": {
    "name": "Robel-Corkery",
    "catchPhrase": "Multi-tiered zero tolerance productivity",
    "bs": "transition cutting-edge web services"
  }
},
]
```

EXECUTION STEPS:

Step 1: Install Flask

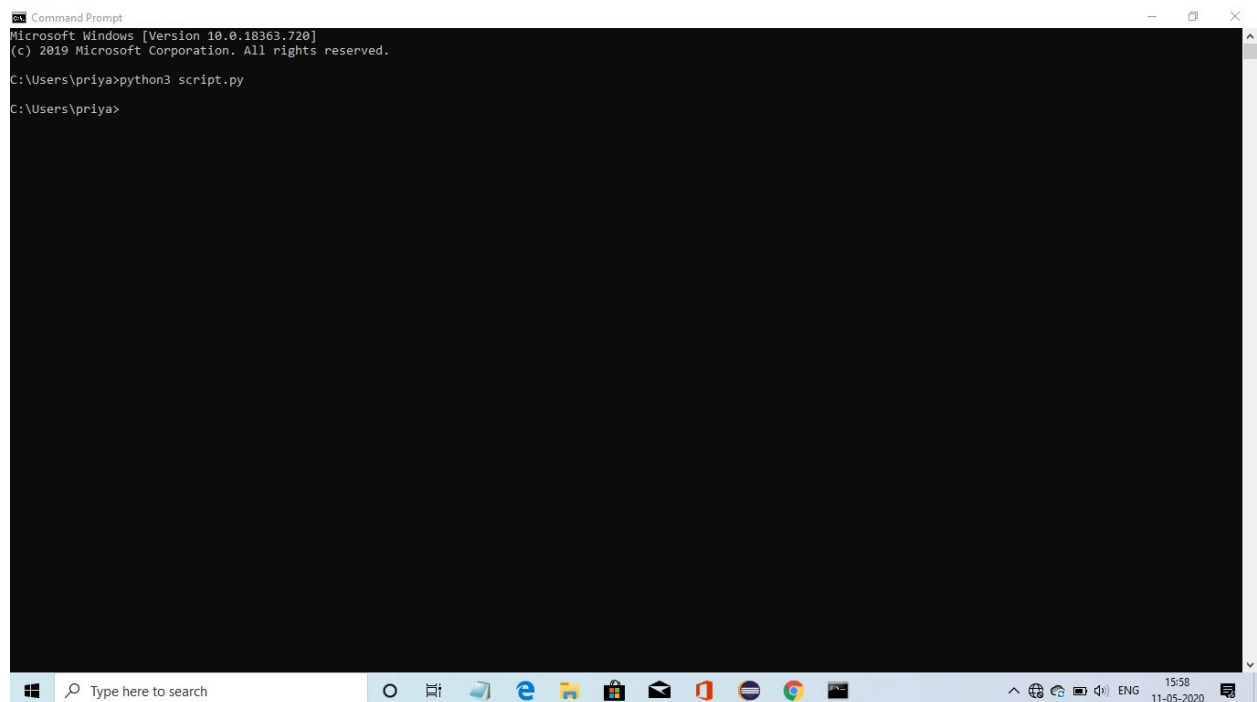
Step 2: Install pandas

Step 3: Run the script in command prompt:

``python3 script.py``

Step 4: Open ``index.html`` in browser

EXECUTION SNAPSHOTS:



```
Command Prompt
Microsoft Windows [Version 10.0.18363.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\priya>python3 script.py
C:\Users\priya>
```

