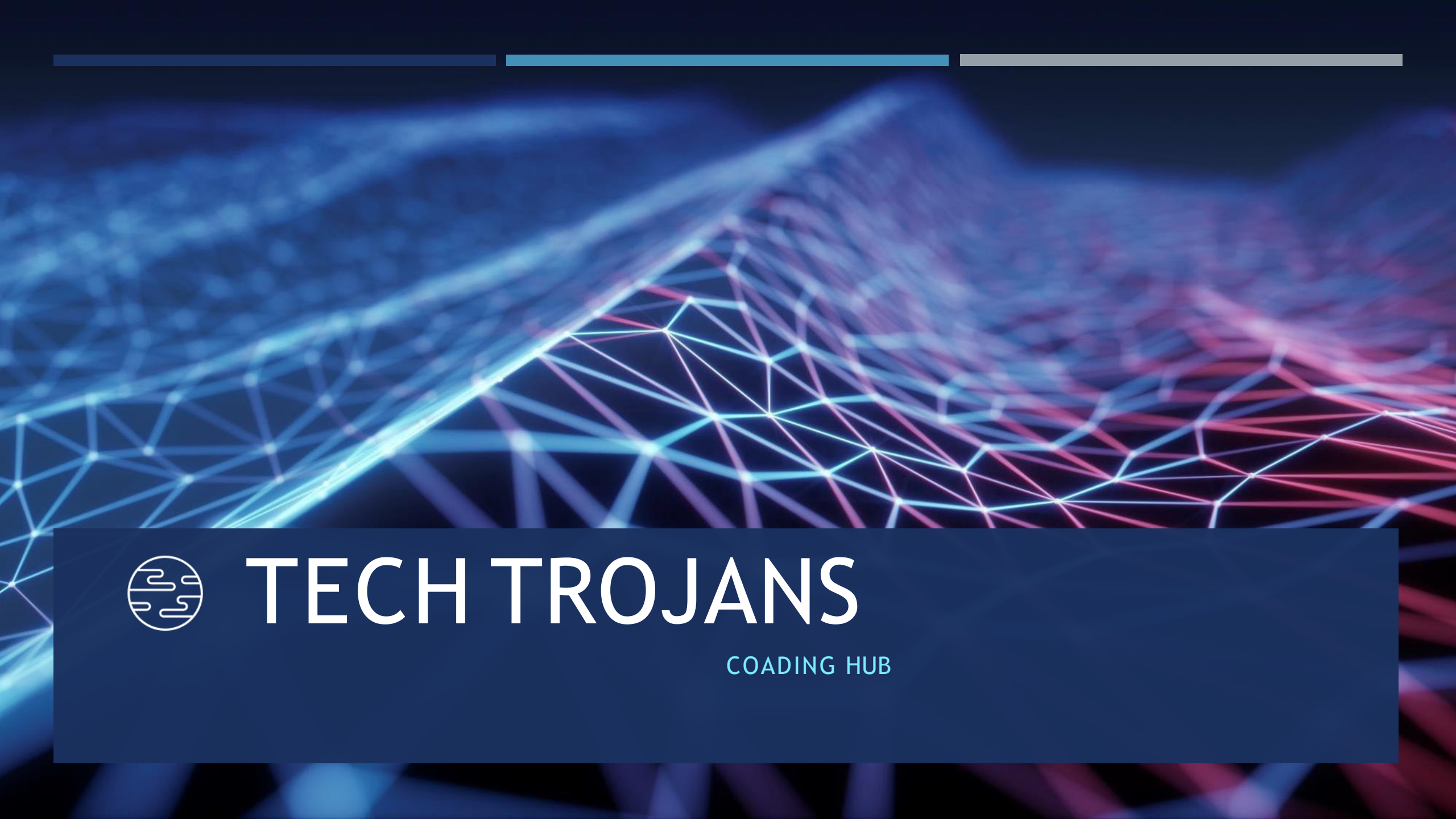


TECH TROJANS

COADING HUB

Ballari Institute Of Technology & Management
Autonomous Institute under VTU,Belagavi | Approved by AICTE,New Delhi | Recognized by Govt. of Karnataka

A Project Presentation on

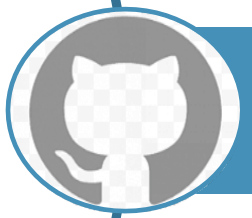
BookTheShow

Presented by

TRoopa Sai - 3BR23CS169
Sirisha T - 3BR23CS156
G Bhavya - 3BR23AI044
Venkat Raavi - 3BR23CS184
Soumya Sree T - 3BR23CS159

Under the guidance of
Ajay Sir

Programming Language [Python]

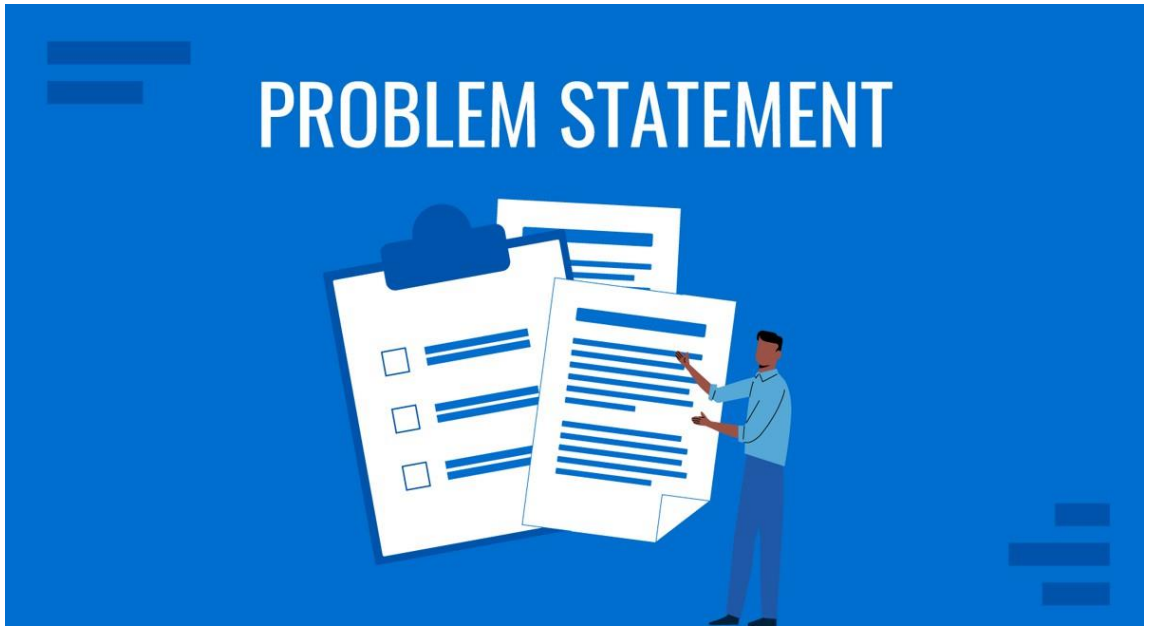


GITHUB [Hosting n Collab]

Link

TECH REQUIREMENTS

PROBLEM STATEMENT :



BookTheShow :

- To book movie ticket.
- Writing code to –

1) Select city
2) Theater
3) Movie
4) Screen number
5) Time
6) No of tickets
7) Selecting Seats

INTRODUCTION :

A Movie Ticket Booking System Project in Python is a digital platform that allows customers to access the services of a business, reserve seats and buy tickets. This platform provides details such as what time a movie will be played, what seats are available, movie previews and so much more. It includes several features that allow users to reserve movie tickets and provide information about movies. The principle of this system, as well as the Python application, is quite obvious.

CODE EXPLANATION:

1)

```
def initialize_seat_matrix():
    return [['O' for _ in range(5)] for _ in range(5)]
 # 'O' indicates the seat is available
```

Explanation:

- This function creates a 5x5 seat matrix, where each seat is represented by 'O', indicating that the seat is available.

2)

```
def display_seat_matrix(seats):
    print("\nCurrent seat availability (O = Available, X = Booked):")
    for row in seats:
        print(' '.join(row))
    print() # Empty line for better readability
```

Explanation:

- This function prints the current seat availability.
- It iterates through each row of the seats matrix and joins the elements (seats) with a space for better visualization.

```
3) def select_seats(seats, num_tickets):
        selected_seats = []
    for _ in range(num_tickets):
        try:
            row = int(input("Enter seat row (1-5): ")) - 1
            col = int(input("Enter seat column (1-5): ")) - 1
            if seats[row][col] == 'O':
                seats[row][col] = 'X'  # Mark seat as booked
                selected_seats.append((row + 1, col + 1))  # Save seat number
for confirmation
                print(f"Seat ({row + 1}, {col + 1}) successfully booked.")
                display_seat_matrix(seats)  # Show updated seat matrix after
booking
            else:
                print("Seat is already booked, please select another seat.")
                continue
        except (ValueError, IndexError):
            print("Invalid input. Please select a valid seat.")
            continue
    return selected_seats
```

Explanation :

- allows users to select seats for booking.
- user gives input of row and column. If the selected seat is available ('O'), it marks it as booked ('X') and adds it to the selected_seats list.
- Then updated seating arrangement.
- If the seat is already booked or input is invalid, it suggests the user for a new selection.

4) def t_movie():
    print("Which movie do you want to watch?")
    print("1. Movie 1")
    print("2. Movie 2")
    print("3. Movie 3")
    print("4. Back")

    try:
        movie = int(input("Choose your movie (1-4): "))
          if movie == 4:
            center() # Go back to theater selection
    else:
        theater()# Proceed to theater selection
     except ValueError:
        print("Invalid input, please choose a valid option.")
    t_movie()

Explanation :

- The user can choose a movie.
- It navigates to the theater selection or goes back to the previous menu.
- It handles invalid input by re-prompting the user.

5)

```
def theater():
    print("Which screen do you want to watch the movie
on?")
    print("1. SCREEN 1")
    print("2. SCREEN 2")
    print("3. SCREEN 3")

    try:
        screen = int(input("Choose your screen (1-3): "))
        tickets = int(input("Number of tickets you want?: "))
        timing(screen, tickets)
    except ValueError:
        print("Invalid input, please enter valid choices.")
        theater()
```

Explanation:

- Allows user to choose a theater screen and the number of tickets.
- After getting valid inputs, it calls the timing function to proceed to the next step.

```
6 )
def timing(screen, num_tickets):
    time1 = { ...}
    time2 = { ...}
    time3 = { ...}

    time_options = {1: time1, 2: time2, 3: time3}

    if screen in time_options:
        print("Choose your time:")
        for k, v in time_options[screen].items():
            print(f"{k}. {v}")
        try:
```

Explantion:

- It gives timing option to selete.
- If the user selects a valid time, it initializes the seating matrix, displays it, and calls select_seats() to book the desired number of tickets.

```
            time_choice = input("Select your time (1-4): ")
            if time_choice in time_options[screen]:
                print(f"Selected time:
{time_options[screen][time_choice]}")
                seats = initialize_seat_matrix()  # Initialize seat
availability
                display_seat_matrix(seats)  # Show available seats
                selected_seats = select_seats(seats, num_tickets)  #
Book seats
                print(f"Successfully booked seats: {selected_seats}")
                print(f"Enjoy your movie at
{time_options[screen][time_choice]}!")
            else:
                print("Invalid time selection. Please try again.")
                timing(screen, num_tickets)
        except KeyError:
            print("Invalid input. Please select a valid time.")
            timing(screen, num_tickets)
    else:
        print("Invalid screen choice.")
        theater()
```

7)

```
def movie(theater_choice):
    if theater_choice in [1, 2, 3]:
        t_movie()
    elif theater_choice == 4:
        city() # Go back to city selection
    else:
        print("Invalid choice.")
        center()
```

Explanation:

- Based on the user's theater choice , this function decides which movie function to call.
- It either proceeds to select a movie, goes back to the city selection, or handles invalid input.

8)

```
def center():
    print("Which theater do you wish to see a movie at?")
    print("1. Inox")
    print("2. Icon")
    print("3. PVP")
    print("4. Back")

    try:
        choice = int(input("Choose your option (1-4): "))
        movie(choice)
    except ValueError:
        print("Invalid input, please choose a valid option.")
        center()
```

Explanation :

- This function allows users to select a movie theater.
- Based on the user's choice, it calls the movie() function or handles invalid input by re-prompting.

9)

```
def city():
    print("Hi, welcome to movie ticket booking!")
    print("Where do you want to watch a movie?")
    print("1. City 1")
    print("2. City 2")
    print("3. City 3")

    try:
        place = int(input("Choose your option (1-3): "))
        if place in [1, 2, 3]:
            center()
        else:
            print("Invalid choice, please select again.")
            city()
    except ValueError:
        print("Invalid input, please enter a number.")
        city()
```

Explanation :

- This function starts the movie booking process by asking the user to select a city.
- Based on the user's choice, it navigates to the theater selection or handles invalid input.

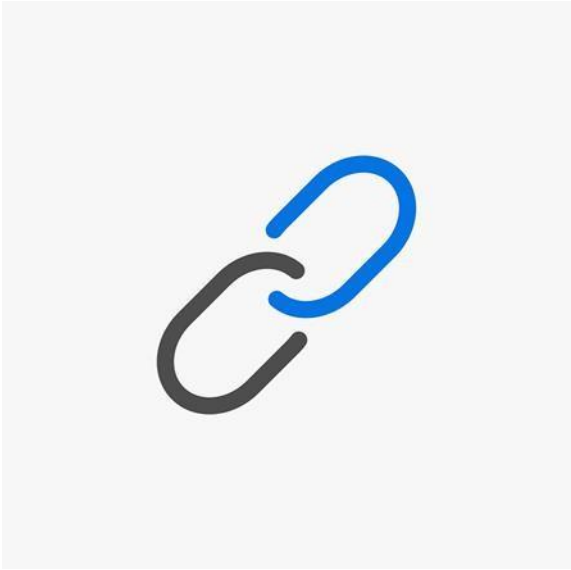
10)

city()

- initiating the entire movie ticket booking process.

CODE :

CLICK ON LINK



Conclusion :

The code defines a simple movie ticket booking system with functions for selecting a city, theater, movie, screen, timing, and booking seats. It uses basic input/output to interact with the user and manages state through a seat matrix and lists of selected seats. Error handling is included to guide users through valid inputs.

THANK YOU