**BIT 319: Management Information System**

**Practical Assignment**

**Assignment: Developing a Simple Information System with Python**

**Movie Ticket Booking System Documentation**

**Author:** Victor kiplangat

Reg\_No: P100/2182g/22

**Date:** 30/10/2024

# Overview of the system

The **Movie Ticket Booking System** is a Python web application designed to facilitate movie ticket reservations, manage showtime schedules, and visually track ticket sales. Built using the Flask framework, this system provides a user-friendly interface for selecting movies and booking tickets. Additionally, it includes a visual representation of ticket sales through a dynamically generated chart.

Table of Contents

[**BIT 319: Management Information System** 1](#_Toc181135513)

[**Practical Assignment** 1](#_Toc181135514)

[**Assignment: Developing a Simple Information System with Python** 1](#_Toc181135515)

[Overview of the system 2](#_Toc181135516)

[System Design 2](#_Toc181135517)

[Project Structure 2](#_Toc181135518)

[Flask Routes 2](#_Toc181135519)

[Data Management 3](#_Toc181135520)

[Functionality 4](#_Toc181135521)

[Ticket Booking 4](#_Toc181135522)

[Viewing Sales Data 4](#_Toc181135523)

[Error Handling 4](#_Toc181135524)

[Output and Visualization 4](#_Toc181135525)

[Sales Data Table 4](#_Toc181135526)

[Sales Chart 5](#_Toc181135527)

[Setup and Usage Instructions 5](#_Toc181135528)

[Prerequisites 5](#_Toc181135529)

[Running the Application 5](#_Toc181135530)

[Booking Process 5](#_Toc181135531)

[Conclusion 6](#_Toc181135532)

# System Design

## Project Structure

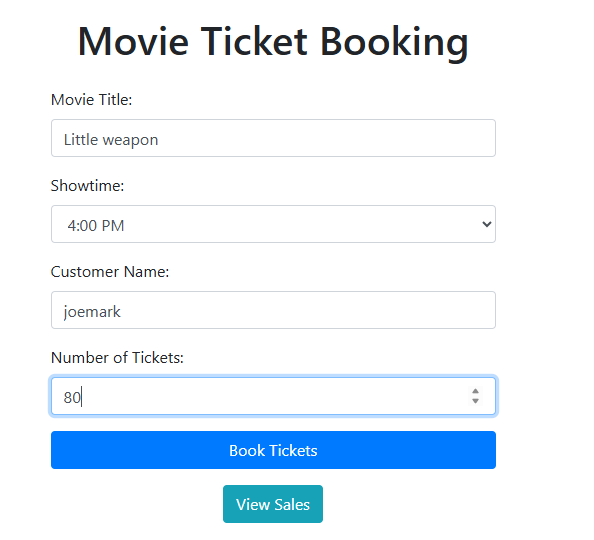
The project follows a structured organization to keep all components accessible:

* **app.py**: Main Flask application script, handling routes and logic.
* **data/sales\_data.json**: JSON file storing booking and sales data, updated with each transaction.
* **templates/index.html**: HTML form where users can book tickets.
* **templates/sales.html**: HTML page displaying the sales data and visual chart.
* **static/sales\_chart.png**: Auto-generated chart image showing ticket sales.

## Flask Routes

The Flask application uses specific routes to handle different operations:

* **Homepage (/)**: Displays the booking form where users can input movie information, select showtime, and specify the number of tickets.
* **Booking Route (/book\_ticket)**: Processes form submissions, validates input, and updates the ticket sales data.
* **Sales Data Page (/view\_sales)**: Shows the sales data in table form and provides a chart visualizing tickets sold.
* **Sales Chart Generation (/sales\_chart)**: Generates a bar chart displaying ticket sales for each movie and showtime.
* **The figure below shows the activities above**



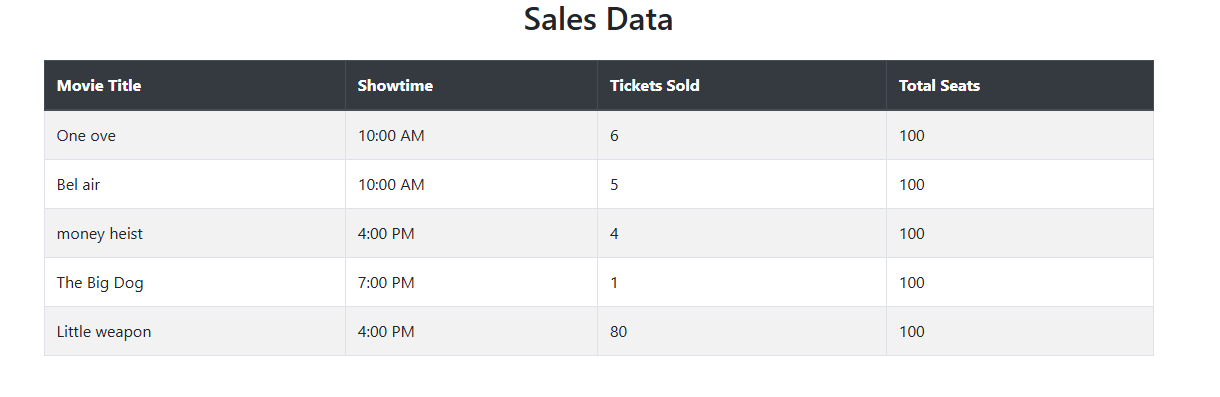
## Data Management

Sales data is persistently stored in a JSON file. This includes:

* **Movie Title**: Name of the movie.
* **Showtime**: Specific time slot for the movie.
* **Tickets Sold**: Count of tickets sold for the particular showtime.
* **Total Seats**: Total available seats for each movie showtime.

Each booking updates this file, allowing data persistence across sessions.

The figure below shows how the data management is done



# Functionality

## Ticket Booking

Customers can book tickets through a straightforward form interface:

* **Movie Title**: User enters the movie title.
* **Showtime**: User selects a showtime from a dropdown menu.
* **Customer Name**: Name of the customer booking the tickets.
* **Number of Tickets**: Quantity of tickets desired.

Upon submitting, the system validates the information and checks ticket availability. If the requested tickets are available, the system confirms the booking and updates the sales data.

## Viewing Sales Data

Sales data is accessible through the **View Sales** page, which displays:

* A table showing all movies, showtimes, tickets sold, and total seats.
* A visual bar chart generated in real time to represent ticket sales, providing insights into popular movies and showtimes.

## Error Handling

The system includes several checks to ensure smooth operation:

* **Input Validation**: Ensures required fields are completed and values are appropriate.
* **Availability Check**: Ensures ticket requests do not exceed availability, prompting the user if adjustments are necessary.

# Output and Visualization

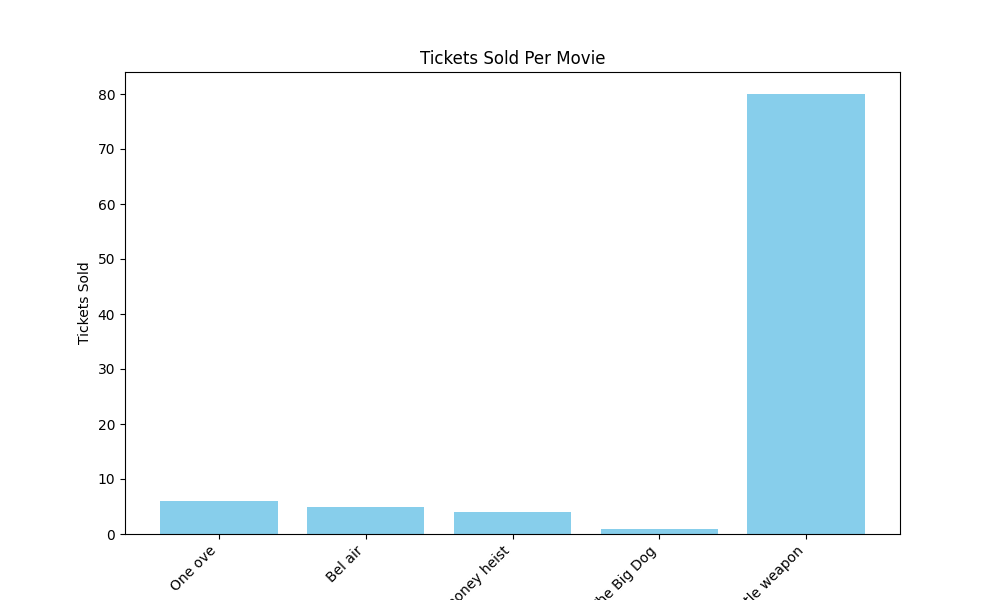
## Sales Data Table

The **Sales Data Table** is presented on the /view\_sales page, displaying:

* **Movie Title**: Title of the movie.
* **Showtime**: Corresponding showtime for the movie.
* **Tickets Sold**: Total number of tickets sold for that showtime.
* **Total Seats**: Total seats available for that showtime.

## Sales Chart

The **Sales Chart** is a bar graph that provides a visual representation of ticket sales across all movies and showtimes. The chart, generated with Matplotlib, allows for quick insights into popular movies and peak showtimes.



# Setup and Usage Instructions

## Prerequisites

1. **Install Flask and Matplotlib**: Use pip to install required libraries.

## Running the Application

1. **Start the Flask Application**: Run app.py in a Python environment to start the server.
2. **Access the Application**: Open a web browser and go to http://localhost:5000 to access the booking form.

## Booking Process

1. **Fill Out the Form**: Enter the movie title, select a showtime, and specify the number of tickets.
2. **Confirm Booking**: Submit the form to complete the booking.
3. **View Sales**: Click the **View Sales** button to see the updated sales data and chart.

# Conclusion

The **Movie Ticket Booking System** provides an efficient, easy-to-use platform for booking movie tickets and tracking sales data. Through its intuitive interface and real-time visualizations, the system simplifies ticket management and provides valuable insights into popular showtimes and movies.