Bennett Movies — Code Introspection & Explanation

This document provides a detailed explanation of each important section and concept used in the **main.py** file of the Bennett Movies — Ticket Booking System. It is designed to help web developers (especially from a React + Node.js background) understand how Streamlit and Python are used together in this project.

1. Imports & Setup

Code:

```
import streamlit as st
import random, string, os, json
```

Explanation:
- streamlit: Handles all UI and user interactions.
- random & string: Used to generate random ticket IDs.
- os: Used for file checking and paths.
- json: Used to read and write movie data stored in movies_data.json.
- st.set_page_config(...) sets the web page title and layout.

2. Database File Reference

Code:

```
DB_FILE = "movies_data.json"
```

Explanation:
This defines the path of the JSON file used as the “database.” Whenever we load or save movie and booking data, Streamlit reads/writes this file.

3. Loading Data Function

Code:

```
def load_data_from_db():
if not os.path.exists(DB_FILE):
st.error("Database file not found")
return {'movies': []}
with open(DB_FILE, 'r') as f:
data = json.load(f)
return data
```

Explanation:
This function loads all movie data from the JSON file. If the file doesn’t exist, it returns an empty structure to prevent crashes.

4. Saving Data Function

Code:
```
def save_data_to_db():
movies_to_save = st.session_state.movies
for movie in movies_to_save:
movie_id = movie['id']
movie['booked_seats'] = sorted(list(st.session_state.booked[movie_id]))
with open(DB_FILE, 'w') as f:
json.dump({'movies': movies_to_save}, f, indent=2)
```

Explanation:
This function saves any new seat bookings made by users into the JSON file, keeping the database updated for future sessions.

5. Initializing the App State

Code:
```
def init_app_state():
if 'movies' not in st.session_state:
db_data = load_data_from_db()
st.session_state.movies = db_data.get('movies', [])
st.session_state.booked = {m['id']: set(m['booked_seats']) for m in st.session_state.movies}
```

Explanation:
Streamlit re-runs your script on every action. So, session_state keeps track of what’s currently happening (like which movie is open and what seats are selected) without losing data on page refresh. This works like React’s useState() or Context API.

6. Helper Functions

Functions:
- seat_label(): Converts row and column numbers to seat codes (e.g., A1, B5).
- generate_seat_ids(): Creates all seat IDs for each movie dynamically.
- format_currency(): Displays ticket price in Rupees format.
- generate_ticket_id(): Creates unique booking IDs like XH2G7YQ9.

Explanation:
These functions help generate display data for the UI.

7. Displaying the Movie List

Code:
```
st.markdown("## Now Showing")
for movie in movies:
st.image(movie['poster'])
st.write(movie['genre'])
st.button("View & Book")
```

Explanation:
This creates cards for each movie using Streamlit widgets. Each button opens a detailed booking page for that movie.

8. Viewing a Movie (Seat Selection)

Code:
```
if st.session_state.view_movie:
movie = next((m for m in movies if m['id'] == st.session_state.view_movie), None)
seats = generate_seat_ids(movie)
for seat in seats:
st.checkbox(seat)
```

Explanation:
When a movie is selected, its seat map is generated. Booked seats are disabled (shown in red), available ones are green, and selected seats are blue. Each checkbox updates session_state automatically.

9. Booking Confirmation Logic

Code:
```
if st.button("Book Tickets"):
for s in selected_seats:
st.session_state.booked[movie['id']].add(s)
save_data_to_db()
st.success("Booking confirmed!")
```

Explanation:
When the “Book Tickets” button is clicked, selected seats are added to the movie’s booked_seats list and saved to the JSON file. Streamlit then re-renders to show updated seat availability.

10. Overall Flow

Steps:
1. App loads → Data read from JSON.
2. Movie list shown → User selects a movie.
3. Seat grid generated → User selects available seats.
4. Booking confirmed → Seats saved in JSON.
5. UI refreshes with updated booked seats.

Explanation:
Streamlit handles UI, backend logic, and data updates — all in one Python file. The JSON file acts as the persistent storage.