Full Stack Developer

The Scenario: Administrators at our partner venues require a streamlined dashboard to monitor and confirm incoming bookings in real-time. Your objective is to develop a comprehensive full-stack, real-time "Live Bookings" viewer.

Your Task: Construct a "Live Bookings Viewer."

Create a straightforward web application where new bookings are displayed instantaneously.

Requirements & Tech Stack:

- **Backend:** Node.js, utilizing Express.js and Socket.IO.
- **Frontend:** Standard HTML, CSS, and JavaScript (React is not required).
- **Database:** Employ an in-memory array on the server to store booking data.
- **Functionality:**
    1. **Backend:**
        - Establish a fundamental Express server.
        - Configure Socket.IO for real-time communication.
        - Implement a "mock" function to simulate new bookings every 5 seconds (e.g., using `setInterval`). This function should generate a random booking object (`{ venueName: "...", partySize: "...", time: "..." }`) and broadcast it to all connected clients via Socket.IO.
    2. **Frontend:**
        - Develop a single HTML page.
        - Integrate the Socket.IO client library to connect with the backend.
        - Listen for the "new-booking" event from the server.
        - Upon receiving a new booking, dynamically create and append a new `<div>` or list item to the page to display the booking details. New bookings should appear at the top of the list without requiring a page refresh.

Deliverables:

- A publicly accessible GitHub repository link containing your complete full-stack project.
- A clear `README.md` file with instructions for installing dependencies and executing both the server and the frontend.

Evaluation Criteria:

- **Full-Stack Integration:** Successful establishment of connection and data flow between the backend and frontend.
- **Real-Time Logic:** Accurate implementation of Socket.IO for pushing updates to the client.
- **Problem Solving:** Demonstrated ability to configure and manage both client-side and server-side environments.