GreenStitch

Frontend Technical Assessment

Seat Booking System

Thank you for taking the time to complete this assessment.

We look forward to seeing your implementation!

1 Introduction

Thank you for taking the time to interview with us at **GreenStitch**! As part of the interview process, we would like you to complete a frontend technical assessment.

You can find all files necessary for the assignment in the `/src` folder. Feel free to make any changes to the provided files, including adding new files, installing new packages, and modifying any provided code. Please use **JavaScript/React** for this assessment.

2 Project Overview

A seat booking system with UI implemented. Your job: implement the booking logic.

2.1 What's Provided

- Complete UI with 8×10 seat grid
- Three seat states: Available (grey), Selected (green), Booked (red)
- Pricing tiers: Premium (1000), Standard (750), Economy (500)
- State management structure

3 Requirements

3.1 Functional Requirements

3.1.1 Seat Selection

Users must be able to click on available seats to select them for booking. Selected seats should be visually distinct from available and booked seats.

3.1.2 Seat Pricing

The system implements tiered pricing based on seat location:

- Premium seats (Rows A-C): 1000 per seat
- Standard seats (Rows D-F): 750 per seat
- Economy seats (Rows G-H): 500 per seat

3.1.3 Price Calculation

The system must display the total price of all currently selected seats in real-time as users select or deselect seats.

3.1.4 Seat Counters

The interface must show live counts of:

- Available seats (not selected, not booked)
- Selected seats (currently selected by user)
- Booked seats (confirmed bookings)

3.1.5 Booking Validation

- Users cannot book more than 8 seats in a single transaction
- Attempting to book more than 8 seats must show an error message
- Booked seats cannot be selected or modified

3.1.6 Booking Confirmation

Before confirming a booking, the system must:

- Show the number of seats being booked
- Display the total price
- Request user confirmation
- Only proceed with booking if confirmed

3.1.7 Booking Persistence

All confirmed bookings must persist across page refreshes. When users reload the page, their booked seats must remain booked.

3.1.8 Clear Selection

Users must be able to clear their current seat selection without affecting already booked seats.

3.1.9 System Reset

The system must provide functionality to reset all seats to available state and clear all persisted booking data.

4 Technical Implementation

4.1 Getting Started

Navigate to the project directory and run:

1. `npm install` – Install dependencies
2. `npm start` – Start development server
3. Open `http://localhost:3000` in your browser

4.2 Code Structure

The main implementation file is `src/SeatBooking.js`. You will find:

- Seat state constants and pricing tiers already defined
- UI components and layout already implemented
- Functions marked with TODO that need implementation
- The seat grid structure (8 rows × 10 seats) initialized

4.3 Implementation Guidelines

- Maintain React best practices (immutable state updates)
- Use React hooks appropriately (useState, useEffect)
- Implement proper error handling
- Ensure the UI updates correctly after each action
- Test all edge cases before submission

4.4 Creativity & Innovation

Going above and beyond is encouraged. While meeting the core requirements is essential, we highly value creativity and original thinking. Feel free to add features, improve the user experience, or implement innovative solutions beyond the base requirements. There are no limitations to creativity.

5 Submission Requirements

Your submission must include the following components:

5.1 1. Code Implementation

- Complete implementation of all three required functions
- All functionality working as specified
- No console errors or warnings
- Clean, well-organized code

5.2 2. Video Walkthrough

- Record a **5-minute video** walking through your code
- The video should cover:
  - Brief overview of the application structure
  - Explanation of your implementation
  - Demonstration of the working application
  - Any design decisions or trade-offs you made
- You can use any screen recording tool (Loom, QuickTime, OBS, etc.)
- Ensure audio is clear and video quality is good enough to read code

5.3 3. Git History

- Initialize a git repository if not already done: `git init`
- Make **meaningful commits** as you work on different parts
- Write clear, descriptive commit messages
- We want to see your development process through your commits

5.4 4. Submission Format

- Create a ZIP file containing:
  1. The complete project folder (including `.git` directory for git history)
  2. A link to your video walkthrough (in a `VIDEO_LINK.txt` file)
  3. **OR** include the video file directly if the file size is reasonable
- **DO NOT** upload your code to GitHub or any public repository
- Name your ZIP file: `GreenStitch-Assessment-[YourName].zip`
- Ensure the ZIP includes the `.git` folder so we can review your commit history

---

Good luck, and we look forward to reviewing your submission!

– *GreenStitch Team*