Okay, the user is asking for code for an AI tour guide website, specifically mentioning a booking system where users can browse hotels, select rooms, make payments, and get a confirmation email with a QR code. Let me break down what they need.

First, I need to outline the components required for such a website. The main features are hotel browsing, room selection, payment processing, and email confirmation with a QR code. Since it's an AI tour guide, maybe there's also a chatbot or some AI-driven recommendations, but the user's main focus seems to be on the booking part.

I should start by considering the frontend. They'll need a user interface to display hotels and rooms. HTML, CSS, and JavaScript are standard. Maybe use a framework like React or Vue, but keeping it simple with vanilla JS for now. Then, the backend to handle bookings and payments. Node.js with Express is a common choice, and MongoDB for the database since it's NoSQL and flexible.

Payment processing integration is crucial. Stripe is a popular choice and has good documentation. The user will need to handle secure transactions, so using Stripe's API makes sense. Then, generating QR codes. A library like qrcode in Node.js can generate QR codes, which can be sent via email.

Email confirmation requires an email service. Nodemailer is a good option for sending emails from Node.js. The email template should include the QR code image, probably as an attachment or hosted somewhere. But hosting the image might be tricky; maybe generate the QR code as a data URL and embed it directly in the email.

Security aspects: The user needs to ensure that payment details are handled securely. Using Stripe Elements on the frontend to collect card info without it touching their server is best. Also, storing booking details securely in the database, maybe encrypting sensitive data.

Now, structuring the code. The frontend part will have HTML for displaying hotels and a booking form. CSS for styling, and JavaScript to handle form submission, interact with the backend API. The backend will have routes for getting hotels, creating bookings, processing payments. The database models for Hotel, Room, and Booking.

Testing the flow: User selects a hotel and room, enters details, pays, then the server processes the payment, creates a booking, generates QR code, sends email. Need to handle errors, like payment failures, and provide appropriate feedback to the user.

Possible issues: QR code generation might take time, so asynchronous processing is needed. Email delivery could fail, so retry mechanisms or logging is necessary. Also, ensuring the QR code in the email is correctly formatted and scannable.

Dependencies: The backend will need Express, MongoDB with Mongoose, Stripe, qrcode, Nodemailer. Frontend might need Axios for API calls, and Stripe.js for payment processing.

I should provide a basic structure with example code for each part, highlighting the key functionalities. Mention that this is a simplified version and real-world applications need more security, error handling, and scalability considerations. Also, note that the user needs to set up environment variables for Stripe keys, database URI, and email credentials.

Okay, the user wants to create an AI tour guide website with booking functionality using React.js and Material UI. Let me start by recalling the previous answer I provided, which was a basic HTML/CSS and vanilla JavaScript setup with a backend. Now they're moving to React and Material UI, so I need to adjust the approach accordingly.

First, I should outline the structure of the React components. The main components would be the HotelBooking component, which displays hotels, room selection, and the payment form. Material UI components like Card, Grid, Button, and TextField will be useful here.

I need to set up Stripe integration in React. The @stripe/stripe-js package is essential for frontend Stripe elements. Also, using react-router-dom for navigation makes sense, so I should mention that in the setup steps.

For state management, useState and useEffect hooks will handle loading hotels, selected room, and booking status. The backend API calls will be made using fetch or axios. I should include sample code for fetching hotels and handling the payment submission.

Material UI styling is important. I should demonstrate how to use makeStyles or styled components to style the HotelCards and the payment form. Including the Stripe CardElement within a Material UI TextField might improve the UI consistency.

The backend part remains similar to the previous answer, but I should mention using Express.js with CORS enabled for API endpoints. The confirmation email with QR code can be generated using a library like qrcode and sent via Nodemailer or a transactional email service.

Testing is crucial. I should remind the user to replace placeholder keys and set up environment variables. Also, mentioning the necessary npm packages to install, like @material-ui/core, @stripe/stripe-js, and react-router-dom, will help them get started.

I should check if there are any specific Material UI components that fit well with the booking flow. For example, using a Dialog for the payment form or Snackbars for notifications. However, the example might keep it simple with conditional rendering based on state.

Security considerations: The Stripe integration should use public keys on the frontend and handle the actual payment processing on the backend with the secret key. Environment variables in React (via .env) and the backend are important here.

Error handling in the React components, like catching API errors and displaying messages to the user, should be included. Loading states while fetching data will improve user experience.

Finally, the QR code generation and email sending in the backend should be outlined, possibly referencing libraries and providing a code snippet for the confirmation endpoint.