**Booking Page Documentation**

**1. Overview**

The Booking Page is the central feature of a Navatar application. It enables users to book, view, and manage their consultation slots with ease. The system is designed for both user-friendly interaction and robust backend reliability.

**2. What Was Implemented**

**Backend**

* **API Development:**
  + Built using FastAPI, a modern Python web framework.
  + Provides endpoints for creating, viewing, and canceling bookings.
* **Database Integration:**
  + Initially used SQLite for local development and testing.
  + Migrated to a cloud-based PostgreSQL database (Neon) for production, ensuring scalability and reliability.
* **Cloud Deployment:**
  + Deployed the backend to Vercel, leveraging serverless functions for efficient scaling.
  + Managed environment variables for secure database access and configuration.
* **CORS Configuration:**
  + Configured CORS middleware to allow secure communication between the frontend and backend, resolving cross-origin issues.

**Frontend**

* **User Interface:**
  + Developed with React and Vite, offering a responsive and intuitive booking interface.
  + Users can select dates, choose flexible time slots, and manage their bookings.
* **Booking Logic:**
  + Implemented logic for checking overlapping bookings and preventing double bookings.
  + Added validation for past and overlapping time slots.
* **Notification System:**
  + Enhanced with reminder notifications for upcoming consultations.
  + Added loading and processing indicators during booking and cancellation, improving user experience and preventing multiple submissions.
* **Error Handling:**
  + Clear error messages and feedback for user actions, such as invalid time slots or failed booking attempts.

**3. How It Works**

**Booking Flow**

1. **User Authentication:**
   * Users log in using their credentials.
2. **Booking Creation:**
   * Users select a date and start-time and end-time.
   * The system checks for overlaps and validity before confirming the booking.
3. **Booking Management:**
   * Users can view their upcoming bookings.
   * Bookings can be canceled at any time.
4. **Notifications:**
   * Users receive reminders for upcoming consultations.
   * Loading indicators are shown during booking and cancellation processes.

**Technical Highlights**

* **Backend:**
  + FastAPI handles all booking logic and data persistence.
  + PostgreSQL on Neon ensures reliable and scalable data storage.
  + CORS middleware allows secure cross-origin requests between the frontend and backend.
* **Frontend:**
  + React provides a dynamic and responsive user interface.
  + Vite ensures fast development and build times.
  + Notification and loading logic enhance user experience and prevent errors.

**4. User Experience**

* **Intuitive Interface:**
  + Simple navigation and clear feedback for user actions.
* **Reliable Operations:**
  + Robust error handling and validation.
  + Instant feedback during booking and cancellation.
* **Timely Reminders:**
  + Users are notified of upcoming consultations, reducing missed appointments.

**Repository Links**

* **Backend (FastAPI):**  
  <https://github.com/Suja2004/Navatar>
* **Frontend (React/Vite):**  
  <https://github.com/Suja2004/Navtar>