Hotel Kiosk - Product Overview

07 February 2026 13:46

1. Product Vision

Nexus is a next-generation self-service hotel kiosk designed to streamline the guest check-in and booking process. It combines a modern, touch-friendly interface with an advanced AI Voice Agent ("Siya AI") to provide a seamless, premium guest experience.

The system is built to be a **"Dumb Frontend, Smart Backend"** architecture, where the frontend acts purely as a renderer for the state and events driven by the backend services.

2. Core Features

2.1 Dual Interaction Modes

- **Voice Mode (Default)**: Users interact naturally with "Siya AI". An animated "Orb" provides visual feedback for listening, thinking, and speaking states.
- **Manual Mode (Touch)**: Users can switch to a traditional touch interface with clear buttons for key actions.

2.2 Key User Journeys

The kiosk supports two primary workflows:

1. **Check-In (Reservation Holders)**:
    - Guest identifies themselves (Voice or Touch).
    - System requests ID scan.
    - System verifies reservation.
    - Payment verification (if needed).
    - Room Key dispensed.

2. **Book a Room (Walk-ins)**:
    - Guest requests a room.
    - System displays available rooms with pricing and amenities.
    - Guest selects a room.
    - Payment processing.
    - ID Scan & Key dispensing.

2.3 Hardware Simulation

The application currently simulates hardware interactions for development and demo purposes:

- **ID Scanner**: Simulates scanning a physical ID document.
- **Key Dispenser**: Simulates the physical encoding and dispensing of a distinct room key card.

3. User Flow & State Machine

The User Interface is a strict function of the UIState. The frontend does not decide "what happens next"; it only emits **Intents** (e.g., CHECK_IN_SELECTED) to the backend, which constructs the next state.

State Transitions (Happy Path)

Session StartUser chooses Check-inUser chooses Book RoomScan CompleteRoom ChosenPayment ConfirmedHardware ActionTimeout/ResetIDLEWELCOMESCAN_IDROOM_SELECTPAYMENTKEY_DISPENSINGCOMPLETE

4. Feature Breakdown

4.1 Welcome Screen (

WelcomePage.tsx)

- **Visuals**: Animated gradient background, Voice Orb.
- **Actions**:
  - "Check In" ->
    Emits CHECK_IN_SELECTED
  - "Book Room" ->
    Emits BOOK_ROOM_SELECTED
  - "Help" ->
    Emits HELP_SELECTED (Notifies staff)
  - "Switch to Manual/Voice" -> Toggles local UI mode.

4.2 Room Selection (

RoomSelectPage.tsx)

- **Purpose**: Allows walk-in guests to view and select available rooms.
- **Data Driven**: Displays rooms provided by the backend (mocked in rooms.mock.ts).
- **Content**: Room images, price per night, amenities list.

4.3 Payment & checkout (

PaymentPage.tsx)

- **Purpose**: Review bill and capture payment.
- **Details**: Shows breakdown of Room Cost + Taxes.
- **Interaction**: Simulates card dip/tap.

4.4 ID Scanning (

ScanIdPage.tsx)

- **Purpose**: Security and verification.
- **Interaction**: Prompts user to place ID on scanner window.

- **Feedback**: Visual progress bar or "Scanning..." animation.

5. Technical Architecture

5.1 Technology Stack

- **Framework**: React (Vite)
- **Language**: TypeScript
- **Styling**: Tailwind CSS, Framer Motion (for animations)
- **Icons**: Lucide React

5.2 Frontend-Backend Contract

- **Rule**: The frontend never assumes logic. It listens to AppState via

  mockBackend.ts.
- **Mock Backend**: A local service (

  services/mockBackend.ts) simulates the server, handling delays, state transitions, and data persistence during the session.

6. Future Roadmap (Out of Scope for current MVP)

- Integration with real Property Management System (PMS).
- Integration with physical ID Scanner SDK.
- Integration with Payment Gateway (Stripe/Adyen).
- Real-time Voice STT/TTS (Speech-to-Text) integration (currently mocked).