

# 🧠 Task 1 - Voice AI Booking Assistant

## 📅 Date

Jul 6, 2025

## 👥 Participants

- @TASNIM MAHDIYA / UPM

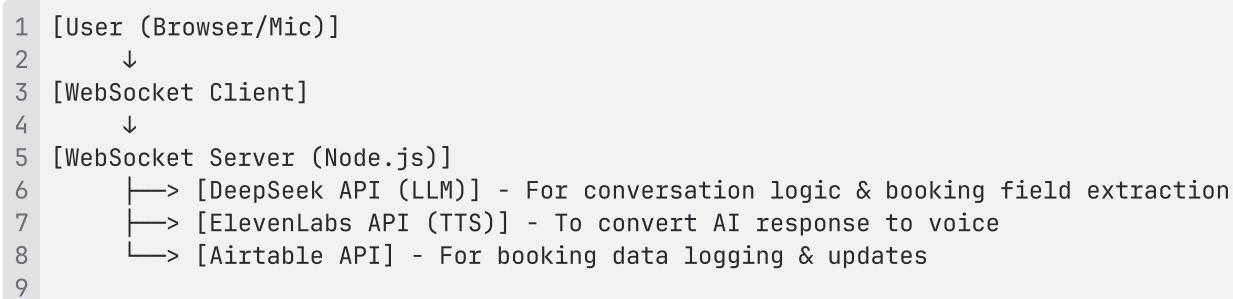
## 📌 Objective

This module implements a real-time AI-powered **Voice Booking Assistant** for a dialysis center. It enables users to book appointments via natural conversation through a microphone-enabled web app, with all data processed in real-time using LLMs and stored in Airtable as CRM.

## ✅ Scope

- Real-time communication via WebSocket
- Natural voice-based interaction for appointment booking
- Intelligent AI response with memory of user session
- Booking data extraction and Airtable integration
- TTS playback using ElevenLabs
- Support for multilingual input

## 🔧 System Architecture



## 🔧 Tech Stack

Layer	Technology	Purpose
Backend	Node.js + Express	Core server and WebSocket communication
AI/NLP	DeepSeek API	Understanding intent and generating replies
TTS	ElevenLabs API	Generating natural voice from text

CRM Database	Airtable	Logging booking info and conversation
Real-time Comm	WebSocket ( ws )	Persistent full-duplex communication
File System	fs module	Storing voice responses as audio files

## 📌 Environment Variables (.env)

Variable	Description
PORT	Server port
AIRTABLE_API_KEY	Airtable personal access token
AIRTABLE_BASE_ID	Airtable base ID
VOICE_ID	ElevenLabs voice ID
DEEPSEEK_API_KEY	DeepSeek model API key

## 📖 API Integration Summary

### DeepSeek API (LLM)

- **Purpose:** Understanding booking intent and generating friendly replies
- **Endpoint:** `https://api.deepseek.com/chat/completions`
- **Input:** User text + optional system prompt
- **Output:** AI response (natural language)

### ElevenLabs API (TTS)

- **Purpose:** Converting AI reply text into realistic voice audio
- **Endpoint:** `https://api.elevenlabs.io/v1/text-to-speech/{VOICE_ID}/stream`
- **Output:** Audio buffer ( `reply.mp3` )

### Airtable API (CRM)

- **Purpose:** Log and update booking-related fields
- **Base/Table:** Bookings > Conversations
- **Fields:** Name, Date, Time, Location, Preferences, AI Reply, User Transcript

## Session Design

- Each WebSocket connection is uniquely identified using:

```
1 socketId = remoteAddress + ":" + remotePort
```

- Server maintains:







```

1 sessions[socketId] = {
2   recordId,           // Airtable record ID
3   bookingData: {      // Tracks user-provided fields across conversation
4     Name, Date, Time, Location, Preferences
5   }
6 }
7

```

- Booking progresses naturally without repeating fields. AI asks only for missing info.

## Feature Highlights

Feature	Description
 Session Memory	Remembers fields already provided in the session
 Missing Field Detection	Automatically checks for missing fields and prompts user accordingly
 Airtable Record Sync	Creates one record per session and updates incrementally
 Natural LLM Responses	Conversational AI with booking-specific guidance
 Humanlike TTS Responses	Converts text replies into voice using ElevenLabs
 Multilingual Understanding	AI adapts to user's language when responding

## Airtable Table Schema

**Table:** `Conversations` (under `Bookings` base)

Field Name	Type	Description
<code>Name</code>	Single line text	Patient's full name
<code>Date</code>	Date	Appointment date
<code>Time</code>	Single line text	Time slot of booking
<code>Location</code>	Single line text	Dialysis center location
<code>Preferences</code>	Single line text	Room or doctor preference
<code>User Transcript</code>	Long text	Full user input

AI Reply	Long text	AI-generated response
----------	-----------	-----------------------

## ⚙️ How Backend Works (Simplified Flow)

- 1. User speaks into the mic.**
  - Browser transcribes speech and sends text to backend via WebSocket.
- 2. Backend receives message.**
  - Extracts booking data using regex and stores it per session.
- 3. Backend checks session.**
  - Fills missing fields progressively and updates Airtable record.
- 4. DeepSeek generates AI response.**
  - Based on current state of fields and system prompt.
- 5. ElevenLabs generates reply voice.**
  - AI reply is turned into voice audio and sent back to the user.
- 6. Frontend plays reply audio.**
  - A smooth, conversational experience continues until booking is complete.

## 📖 Deployment Instructions

```
1 # Step 1: Install dependencies
2 npm install
3
4 # Step 2: Create .env file with API keys and IDs
5 PORT=3000
6 AIRTABLE_API_KEY=...
7 AIRTABLE_BASE_ID=...
8 VOICE_ID=...
9 DEEPSEEK_API_KEY=...
10
11 # Step 3: Start the server
12 node server.js
13
```

Server runs at:

📍 <http://localhost:3000> (with WebSocket support)