PropVivo Full Stack Coding Assessment Document

Project Overview

Build a customer support portal that automatically shows customer information when calls come in. The system should also change Indian accents to American accents during calls.

What You Need to Build

The Problem

- Support agents waste time switching between multiple systems to find customer info
- Customers wait while agents search for their details
- Some customers prefer American accents

The Solution

- One screen that shows customer info automatically when calls arrive
- Built-in voice calling with accent conversion
- Fast and efficient customer lookup

Core Features Required

1. **Incoming Call Detection**
    - Integrate with a call-receiving API
    - Show incoming calls in real-time on the frontend
2. **Automatic Customer Lookup**
    - Search database using caller's phone number
    - Display customer details instantly
3. **Two-Way Voice Communication**

  - Allow voice conversations through the interface
  - Clear audio quality for both parties

4. **Real-Time Voice Modulation**
   - Convert Indian accent to American accent during calls
   - Transform support agent's voice in real-time
   - Works automatically while speaking to customers
5. **Error Handling**
   - Handle API failures gracefully
   - Show helpful error messages
   - System should remain stable

Technical Requirements

Frontend

- **Use any framework you prefer:** React, Vue, Angular, or plain JavaScript but React is preferred.
- Must display real-time call notifications
- Must show customer information clearly
- Must have voice call controls

Backend

- **Must use C#** (any version/framework)
- Handle API integration for calls
- Manage database operations
- Process voice modulation
- Implement proper error handling

Database

- **Use any database:** SQL Server, MySQL, PostgreSQL, MongoDB, etc.
- Store customer information
- Track call logs

- Must support phone number searches

System Flow

Incoming Call → API Integration → Customer Lookup → Display Info → Voice Chat

---

Minimum Database Requirements

Customer Information:

- Name, phone number, email
- Any additional details you think are useful

Call Logs:

- Basic call tracking (optional but recommended)

Deliverables

1. **Frontend Application**
   - Working user interface
   - Real-time call display
   - Customer information view
   - Voice controls
2. **Backend API**
   - C# application
   - Call API integration
   - Database connectivity
   - Real-time voice modulation processing
3. **Database**
   - Customer data storage
   - Any database technology

Success Criteria

- Incoming calls appear on screen automatically
- Customer info shows up when calls arrive
- Voice communication works both ways
- Real-time accent modulation (Indian to American) functions during calls
- Support agents' voices are transformed while customers hear American accent
- System handles errors without crashing

Implementation Notes

- **Choose your tools:** Pick the frontend framework and database you're most comfortable with
- **Focus on functionality:** Make sure core features work well
- **Keep it simple:** Don't overcomplicate the design
- **Test thoroughly:** Make sure voice features work properly

---

Getting Started

1. Set up your chosen frontend framework
2. Create C# backend project
3. Set up your database
4. Integrate with a call API service
5. Build and test each feature step by step

Note: The backend code structure folder will be provided to guide implementation

Remember: Both frontend and backend are required. Use C# for backend, but you have flexibility with everything else!