PropVivo Coding Assessment (JUET)

Full Stack Developer

Problem Statement

A company's customer support team handles a high number of calls every day but finding a caller's details means jumping between multiple websites and systems. It's slow and inefficient. To fix this, the company wants to build a simple portal that automatically pulls up a customer's information the moment a call comes in.

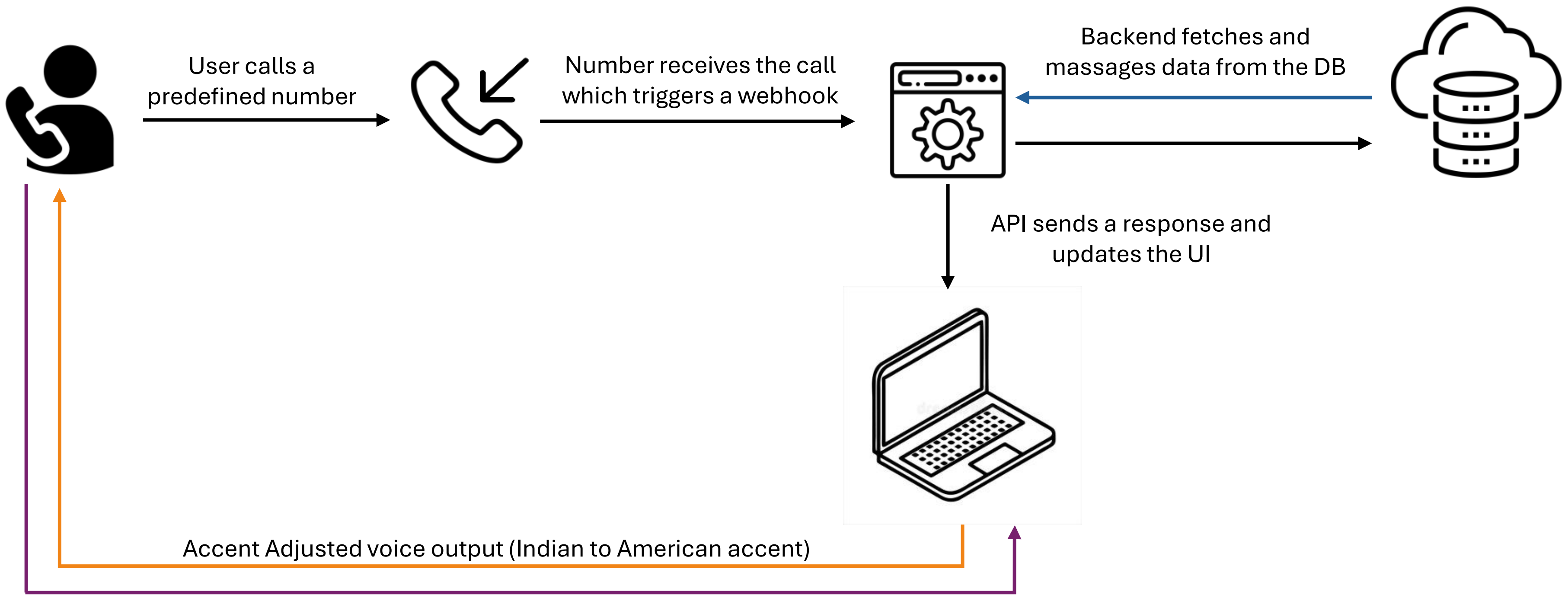
Also, since many American customers prefer speaking with someone who sounds American, the company wants to add real-time voice modulation to convert the support team's Indian accent into an American one during calls.

Core Task

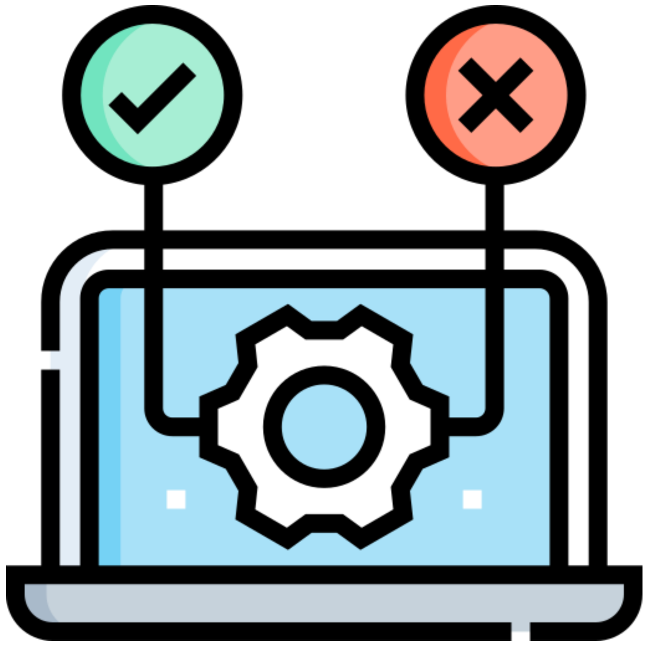
Integrate an existing call-receiving API into the provided codebase to enable the following functionality:

- When a call is received on a specified phone number, a frontend interface should display the incoming call in real time.
- The backend should automatically search through the user database to identify the user whose phone number matches the incoming call.
- Once identified, the user's details should be fetched and displayed on the frontend interface.
- The system must allow the user to engage in a two-way voice conversation through the interface.
- Additionally, implement real-time voice modulation that transforms the local user's Indian accent into an American accent during the call.
- The system must have proper error handling
- Each person is required to build both the front end and the backend
- The backend Should be written in C# and you can use any DBs or frameworks

Visual Representation


User calls a predefined number
Number receives the call which triggers a webhook
Backend fetches and massages data from the DB
API sends a response and updates the UI
Accent Adjusted voice output (Indian to American accent)

Software Tester



Problem Statement

The company has launched a customer support portal, but three core features need to be tested: the login flow, the service request flow, and the board task flow.
The login flow ensures that users can securely sign into the portal with valid credentials.

A service request is raised when a customer needs help or reports an issue—this flow handles the submission, tracking, and assignment of such requests.

A board task refers to internal tasks assigned to board members. These tasks help manage matters related to the association, escalations, or other action items stemming from board-related decisions or requests.

The credentials and website link will be shared with the assigned testers

Core Task

Test the Login Flow

•Verify sign-in with valid/invalid credentials
•Check error handling and logout functionality

Test the Service Request Flow

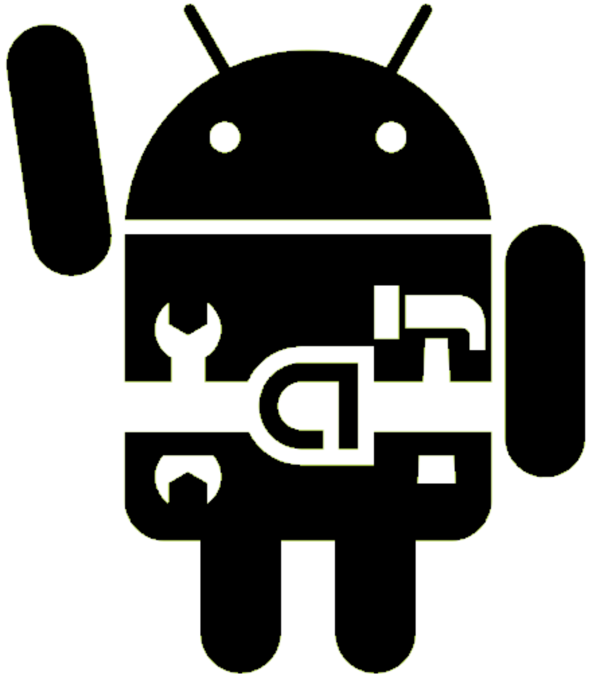
•Create and submit a service request
•Track status and verify proper assignment

Test the Board Task Flow

•Assign and update tasks for board members
•Validate tracking and completion related to association actions
•Report bugs, edge cases, and usability issues

Candidates must write test cases and may also suggest improvements

Android Developer



Problem Statement

Build an android-based Task and Attendance Management System that enables employees to track their work hours, manage tasks, and communicate with superiors effectively.

Core Task

Develop a system with:

Authentication: Login with Employee/Superior roles

Time Tracking: Real-time timer with pause/resume, enforcing 8-hour daily requirement

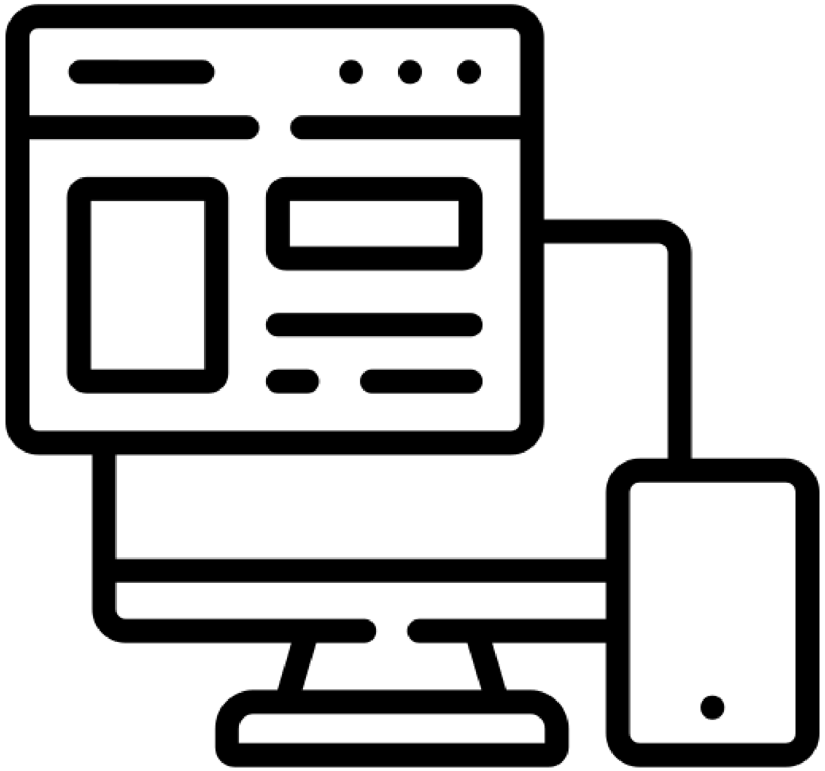
Task Management: View assigned tasks, work on one task at a time, add manual tasks

Dashboards: Employee (tasks, timer, hours) and Superior (all tasks/queries) views

Query System: Employees raise task queries, superiors respond and track status

Reporting: Task-wise time allocation and daily completion status
Key constraints: Only one active task timer, mandatory 8-hour tracking, and role-based access control.

UI / UX



Problem Statement 1

Logistics SaaS Platform: Improving Fleet Operations

Logistics companies face inefficiencies in fleet management due to disjointed communication among drivers, dispatchers, and customers. Drivers encounter unclear routes and payment issues, dispatchers struggle with route optimization, and customers experience delays and lack of transparency.

Core Task 1

Create a SaaS platform for:

Drivers: Clear route guidance, earnings visibility, and real-time updates.

Dispatchers: Route optimization tools, driver tracking, and analytics dashboards.

Customers: Live shipment tracking and delivery feedback mechanisms.

Focus:
Enable cross-platform usage for consistent performance and simplified workflows.

Problem Statement 2

Real Estate SaaS Platform: Empowering Home Buyers and Realtors

Home buyers struggle to navigate the complex home-buying process, with limited access to property details and financing options. Realtors find it challenging to manage leads, schedule showings, and communicate effectively with clients.

Core Task 2

Home Buyers: Property search with personalized filters, mortgage calculators, and document management.

Realtors: Lead tracking, scheduling tools, and analytics for client preferences.

Focus:
Deliver a user-friendly experience that supports decision-making and fosters trust.

Problem Statement 3

Property Management SaaS Platform: Enhancing HOA Management

Homeowners Associations (HOAs) face challenges in managing community rules, maintenance requests, and communication between homeowners, board members, and vendors. Homeowners often feel uninformed, board members are overwhelmed by approvals, and vendors find the process of submitting bids tedious.

Core Task 3

Develop a SaaS platform with:

Homeowners: Simplified access to announcements, payment portals, and maintenance request tracking.

Board Members: Tools for architectural review approvals, financial reporting, and community surveys.

Vendors: A streamlined bidding process and contract management system.

Focus:
Create a unified platform that enables transparent communication and seamless operations, accessible across devices.

Data Science

Problem Statement

- Build a simple AI-powered web application to help multiple users schedule a meeting based on their chat.
- The system should detect scheduling intent, extract availability, compute a meeting time, and send a confirmation email to all participants.

Must-Have Requirements

- Integration with database to display multi-user chats..
- Detect meeting intent and extract availability.
- Propose a common meeting time.
- Ask follow-up questions if needed.
- Send meeting confirmation email to all participants.

Good To Have Features

- A single point of communication or chat where multiple users can interact

System Flow

- Users chat (preloaded or real-time).
- User clicks 'Schedule Meeting’.
- Backend extracts availability & proposes time.
- If info missing, prompt user.
- Schedule meeting and store details.
- Send confirmation email.

Evaluation Criteria

- Judged on backend functionality and logic.
- Must integrate with a simple UI.
- Emphasis on scheduling accuracy and email confirmation.
- Meeting Scheduler Agent architecture

Additional Information

This is not a group project

UI/UX screens should be done on figma