

// PROFILE

Detail-oriented and analytical Full-Stack Web Developer with practical experience building and deploying responsive web applications. Seeking to leverage strong foundational knowledge in software engineering, technical troubleshooting, and modern web frameworks to contribute effectively to a forward-thinking development team.

// TECHNICAL SKILLS

CORE LANGUAGES

Python

HTML5

CSS3

JavaScript (ES6+)

SQL

TECHNOLOGIES & TOOLS

WebRTC

CodeMirror

DOM Manipulation

REST APIs

Git

Whisper

sounddevice

webrtcvad

pyttsx3

CONCEPTS

Data Structures & Algorithms

System Architecture

Responsive UI/UX

// PROJECTS

Qube

VOICE-BASED AI ASSISTANT

01

Real-time voice assistant that detects speech, transcribes it locally with Whisper, and replies out loud — see full breakdown on page 2.

My Buddy

PEER-TO-PEER COMMUNICATION APP

02

Real-time video streaming, screen sharing, and chat over direct WebRTC connections — see full breakdown on page 3.

Quick Compile

BROWSER-BASED DEV ENVIRONMENT

03

Offline-capable playground for testing HTML, CSS and JS in real time, no backend required — see full breakdown on page 4.

// EDUCATION

Bachelor of Computer Applications (BCA)

2024 — 2026 (Exp.)

University of Rajasthan, Jaipur

Core focus on software engineering methodologies, web technologies, and database management systems.

Senior Secondary (Class XII)

Completed

Board of Secondary Education

// STRENGTHS

ANALYTICAL PROBLEM SOLVING

Adept at breaking down complex development bugs into structured, logical technical solutions.

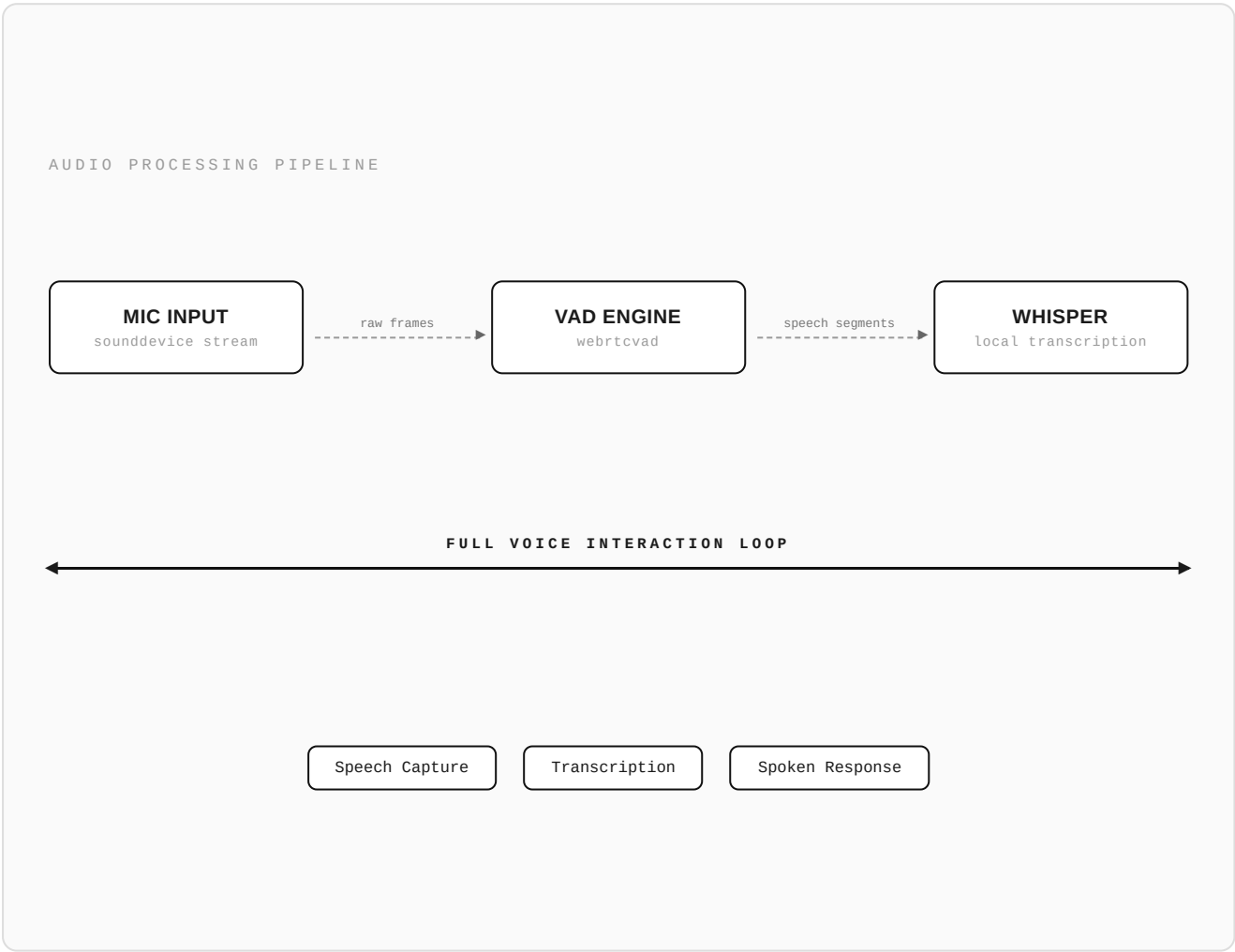
EFFECTIVE COMMUNICATION

Skilled at translating complex technical requirements into clear milestones for peers and stakeholders.

RAPID ADAPTABILITY

Proven track record of mastering new programming frameworks and tools quickly under tight timelines.

Qube is a Python-based voice assistant that listens continuously through the microphone, detects when someone is actually speaking, transcribes that speech locally, and replies with natural spoken audio — enabling hands-free, conversational interaction from start to finish.



// CORE TECHNICAL IMPLEMENTATIONS

AUDIO CAPTURE

Streams live microphone input using sounddevice at 16kHz mono, continuously buffering raw audio frames for downstream processing.

VOICE ACTIVITY DETECTION

Uses webrtcvad to distinguish speech from silence in real time, isolating only meaningful audio segments before they reach transcription.

SPEECH-TO-TEXT

Runs OpenAI's Whisper model locally to convert captured speech segments into accurate text, without relying on a cloud API.

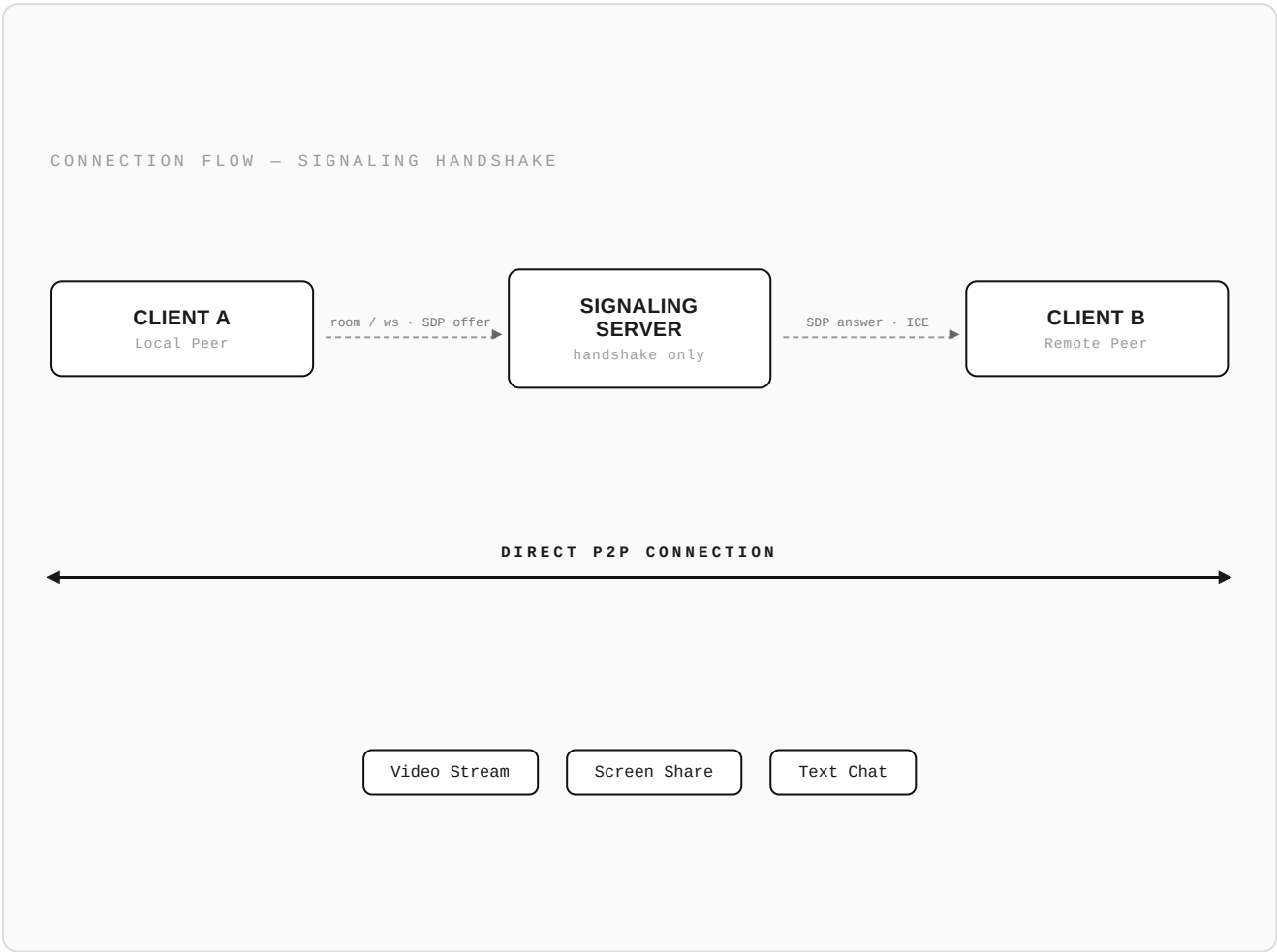
VOICE RESPONSE

Uses pyttsx3 to synthesize and play back spoken replies offline, completing a fully voice-driven, hands-free interaction loop.

// BUILT WITH

- Python
- sounddevice
- OpenAI Whisper
- webrtcvad
- pyttsx3

My Buddy is a decentralized web application engineered for real-time video streaming, screen sharing, and interactive chatting. By utilizing WebRTC, the application establishes direct client-to-client connections — ensuring privacy, minimizing server latency, and significantly reducing bandwidth costs.



// CORE TECHNICAL IMPLEMENTATIONS

MEDIA HANDLING

Captures local media streams (camera and screen share) and handles dynamic video rendering within a responsive grid interface. Automatically mirrors local video for an intuitive user experience.

STATE MANAGEMENT

Manages complex UI states via vanilla JavaScript, including full-screen video toggling, dynamic control docks, and visual badges differentiating local versus remote connections.

DIRECT DATA CHANNELS

Facilitates remote text chatting directly between peers via WebRTC Data Channels, bypassing traditional database storage for maximum privacy and transmission speed.

ERROR HANDLING & CONNECTION STATE

Monitors ICE connection states to gracefully handle dropouts, notifying users immediately upon peer disconnection.

// BUILT WITH

JavaScript (ES6+)

WebRTC

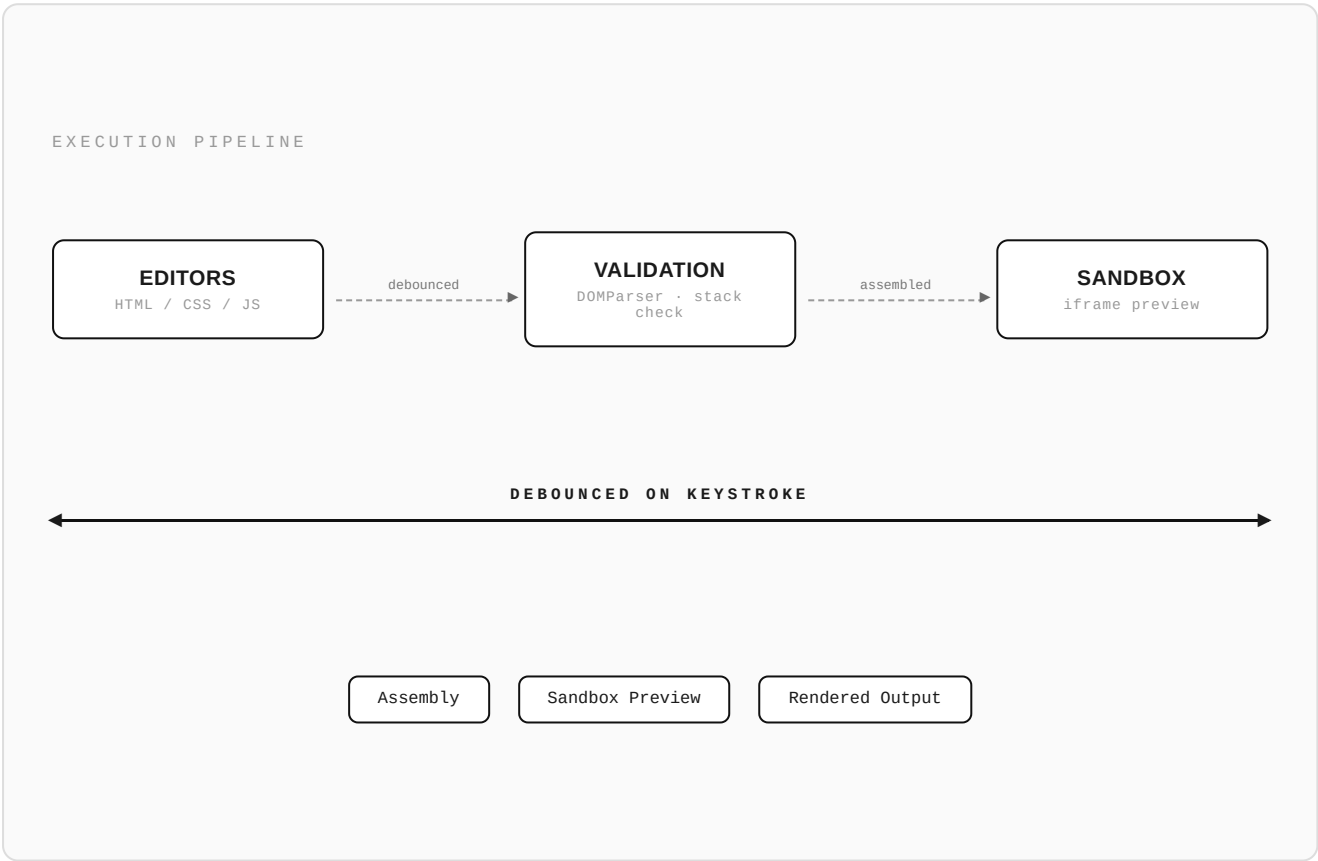
Data Channels

HTML5 / CSS3

WebSocket Signaling

Quick Compile

Quick Compile is a fast, free, and offline-capable web playground designed to test HTML, CSS, and JavaScript in real time. It processes all user code entirely on the client side, ensuring total data privacy while removing the need for continuous backend compilation servers.



// CORE TECHNICAL IMPLEMENTATIONS

EDITOR INTEGRATION

Leverages the CodeMirror library to provide a robust developer editing experience, featuring syntax highlighting, line wrapping, and auto-closing tags.

PERFORMANCE OPTIMIZATION

Incorporates a debounce algorithm on editor input listeners, delaying code execution until the user pauses typing — preventing the browser from freezing during continuous input.

ALGORITHMIC VALIDATION

Employs a custom stack-based validation system utilizing regular expressions to accurately detect mismatched or unclosed HTML elements before execution.

CLIENT-SIDE EXPORT

Packages the active editor contents into a self-contained, downloadable HTML file utilizing the JavaScript Blob and URL.createObjectURL APIs.

// BUILT WITH

JavaScript (ES6+)

CodeMirror

DOM Manipulation

HTML5 / CSS3

Blob API

Thank you for reviewing
Portfolio: samarupreti.netlify.app