Tabby: Your Keyboard Pal

TEAM TABBYCAT

Avinash Anish
Mrida Pradhan

Project Overview

Problem

Keyboards Are Passive!

- They capture keystrokes, not intent
- No awareness of context, app, or goal
- Constant context-switching to prompts & tools

Solution

- An AI-powered desktop keyboard layer
- Works across apps, not inside one app
- Text, voice, and hybrid input
- Real-time, context-aware assistance

Why It Matters

- Cute cat mascot
- AI meets users at the point of input
- Work stays continuous and focused
- Intelligence becomes ambient, not intrusive

Project Objective

Main Goal

To embed AI directly into the keyboard workflow, enabling continuous, low-latency, context-aware assistance without breaking user focus or flow.

Supporting Objectives

- Design an intuitive desktop app suitable for desktop use
- Include persistent AI memory using RAG + Knowledge Graphs
- Keep the system lightweight, offline-friendly, and distraction-free

Who We Designed For

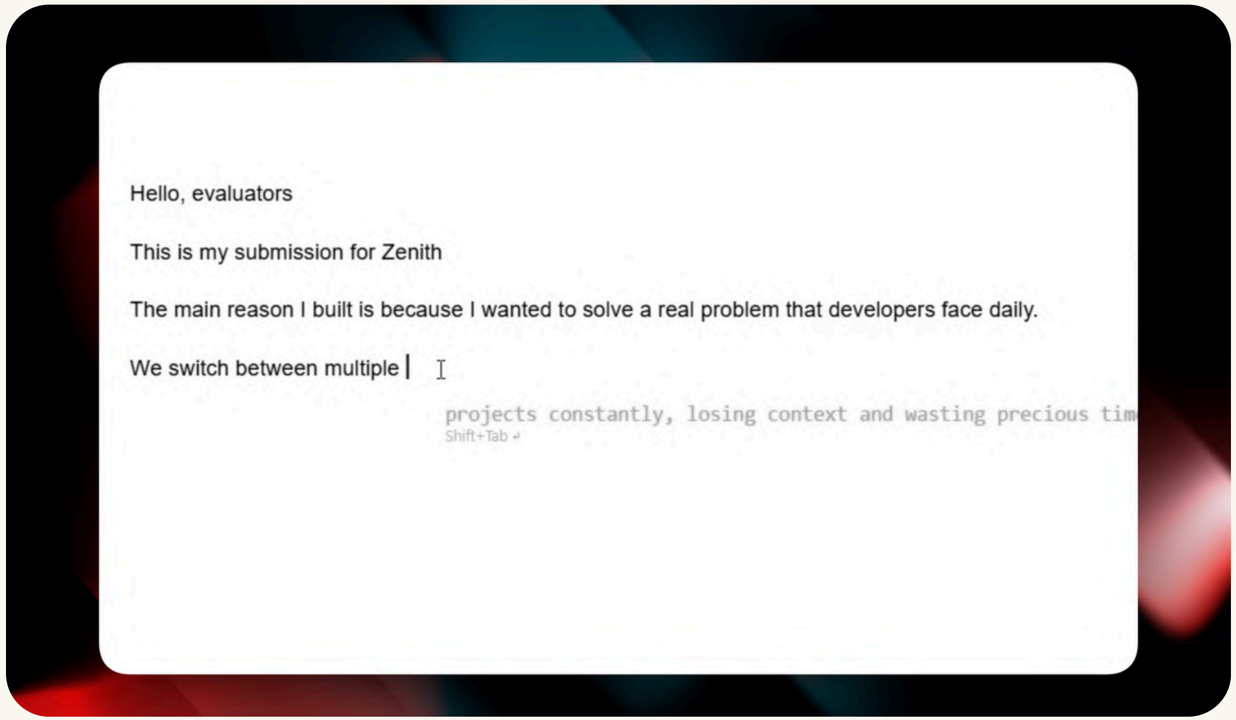
Target User

People who live-laugh-keyboard.

- Students & job-seekers
- Engineers & builders
- Writers, consultants, analysts

Their Challenges

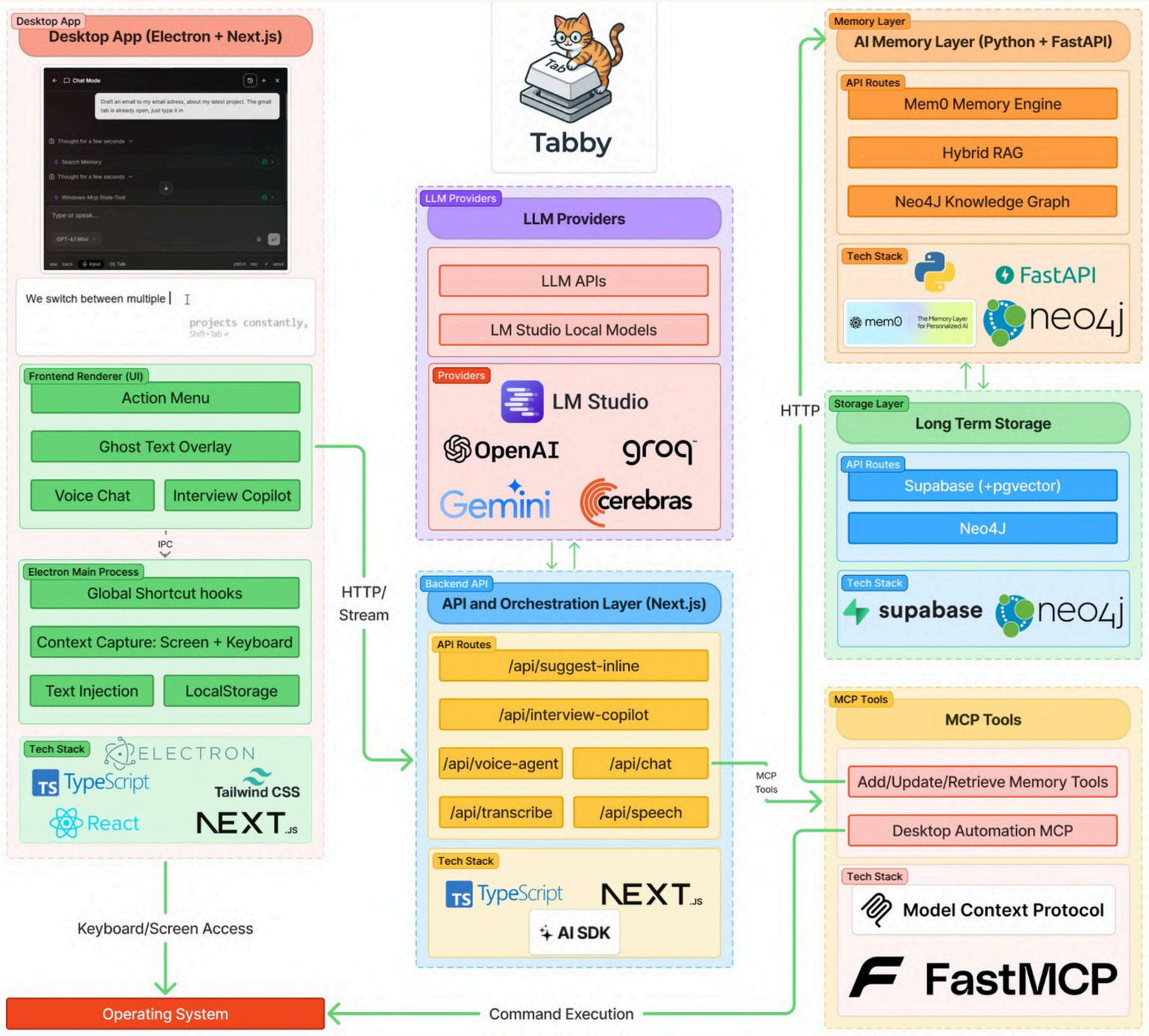
- Jumping between apps to “ask AI”
- Rewriting the same things repeatedly
- Losing context mid-task
- Voice tools that don’t integrate with work



Architecture Diagram

TEAM TABBYCAT

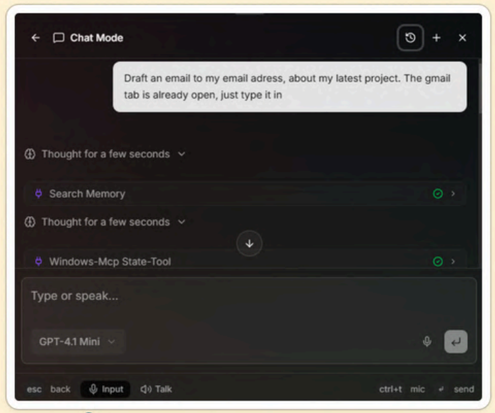
- **Desktop App:** Electron + Next.js with global shortcuts, ghost text overlay & context capture
- **Multi-LLM Support**: OpenAI, Gemini, Groq, Cerebras + local models via LM Studio
- **Memory System:** Mem0 (local) + Hybrid RAG with Neo4j knowledge graph & Supabase vector storage for vector embeddings
- **MCP Tools:** Model Context Protocol for desktop automation & memory tool access
- **Modular API Layer:** Next.js backend connecting all components via HTTP/SSE



TechStack

TEAM TABBYCAT

Tech Stack- Desktop App



ELECTRON

TypeScript

Tailwind CSS

React

NEXT.JS

Data Storage

supabase

Database + VectorDB

neo4j

Knowledge Graph

Tech Stack- Backend

TypeScript

NEXT.JS

AI SDK

Model Context Protocol

FastMCP

Memory + LLMs

FastAPI

mem0 | The Memory Layer for Personalized AI

OpenAI

groq

Gemini

cerebras

Technology Stack

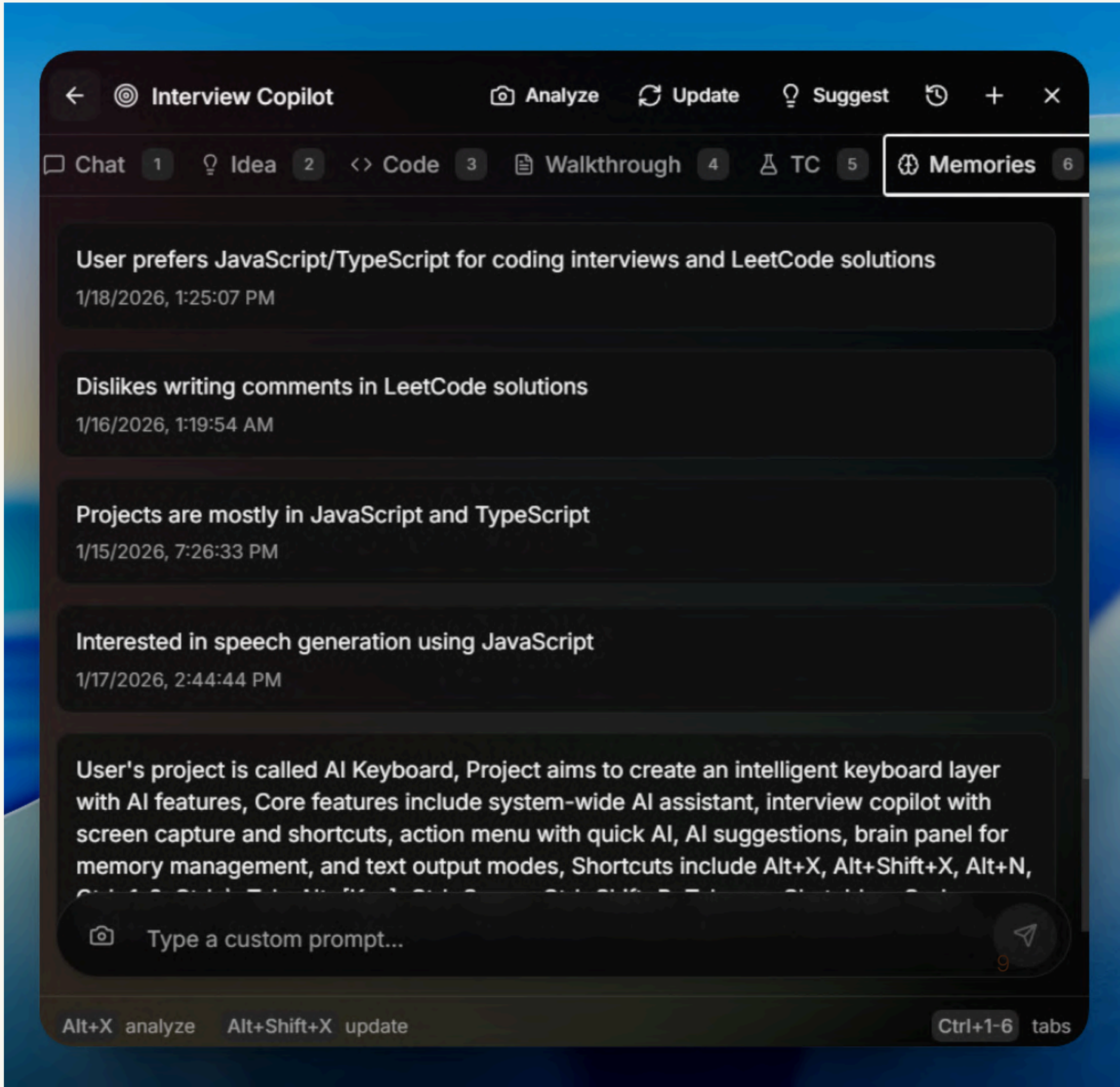
KEY FEATURES

Tabby as a FRAI Plugin

Tabby Improves FRAI's Interview Copilot. Here's how:

- Extends FRAI with a persistent memory system
- Inculcates Memory-Powered Assistance
- Remembers user profile, role, and interview context
- Tracks past answers and feedback
- Avoids repetition and generic responses
- Adapts guidance across the interview session

The copilot becomes contextual with time, as opposed to being stateless.



Other Key Features

- **Context-Aware Inline AI Autocomplete**
  Understands what you're doing from screen + keyboard context in any app

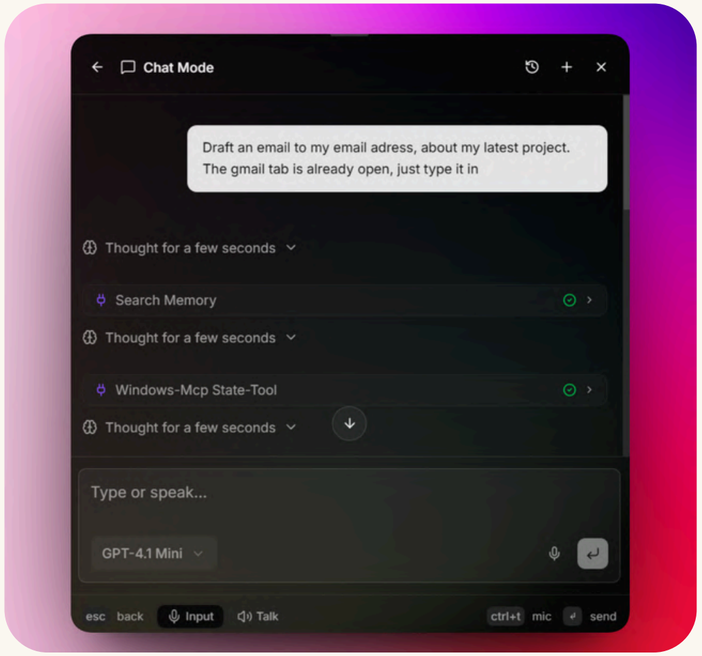
- **Always-On, Low-Latency Assistance**
  AI that works in real time without breaking flow

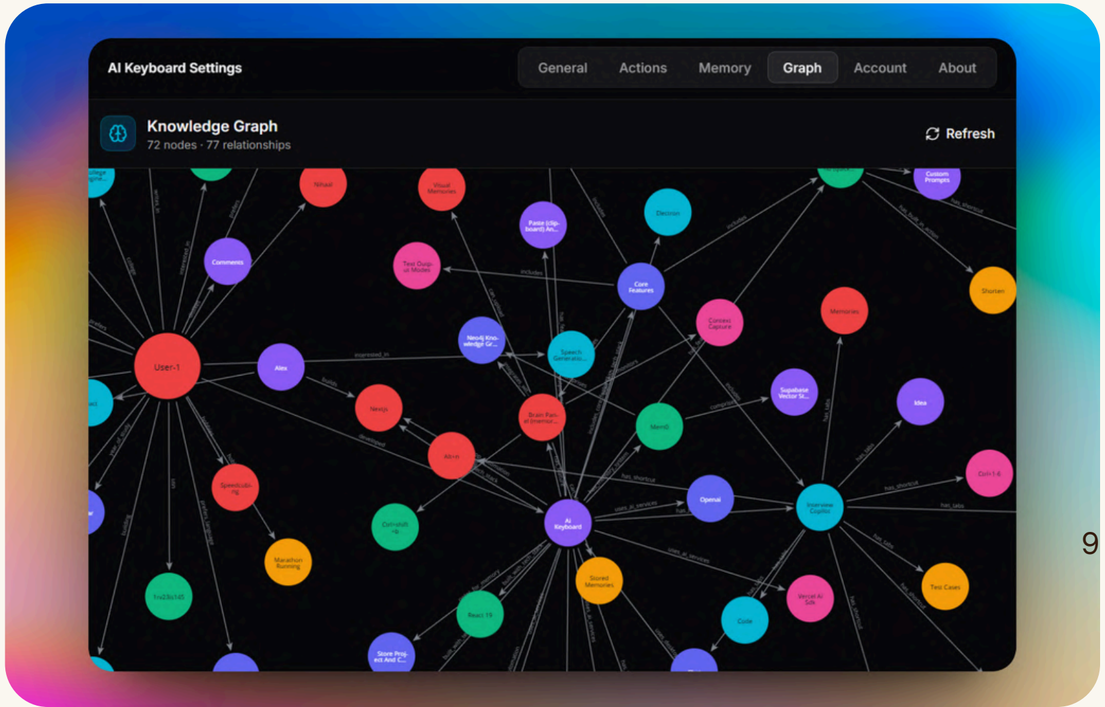
- **Text, Voice & Hybrid Input**
  Seamlessly switch between typing and speaking

- **Cross-App Intelligence**
  Works across tools, not locked into one app

- **Knowledge Graph of memories**

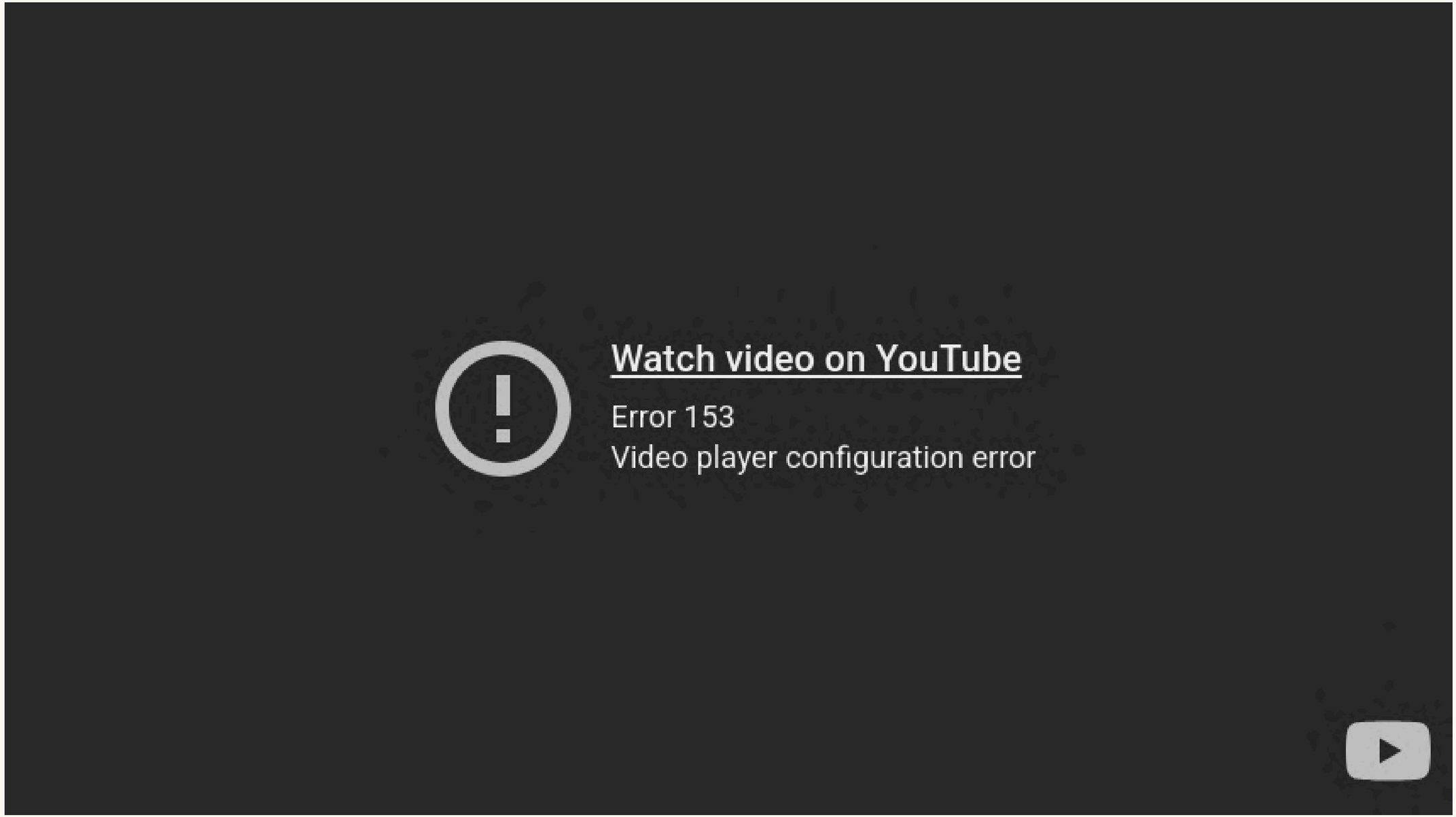
- **Desktop Automation** Open any app and automate mundane tasks





See for yourself!

Our prototype (development in-progress) + face reveal



Link: https://youtu.be/u2kaJNginHw

Tabby says thank you!

“Keycap Tabby isn’t real, it can’t hurt you”

Meanwhile Keycap Tabby: