AI-Powered Smartglass Prototype

Group BA5

22CSB08 MDL22CS036 ANANYA P
22CSB36 MDL22CS126 M GAYATHRI
22CSB41 MDL22CS138 NAVNEET RANJISH PILLAI
22CSB44 MDL22CS144 NIKHIL M

2025

1. Project Overview

- A next-gen AI-powered smartglass prototype that merges ML, IoT, and human-computer interaction into one immersive wearable device.
- Driven by a Raspberry Pi Zero 2 W with on-device ML for edge AI functionality and minimal latency.
- Communicates with a Flutter-based mobile app for voice interaction, internet-based APIs (Gemini, Maps, Spotify), and control interface.
- Emphasizes accessibility: helping visually impaired users through on-demand object/person recognition and voice feedback.
- Implements a HUD (Heads-Up Display) using reflected optics for lightweight AR experiences.

2. Key Features

- **AI Assistant ("Hey Glass")** — Natural language response, app control, queries
- **Heads Up Display** — Real-time reflection-based visual overlay using mirrored TFT
- **Object Detection** — On-demand ML-based detection for accessibility
- **Live Translation** — Snap foreign text, translate with Gemini
- **Navigation + Music** — Voice-controlled Maps and Spotify display
- **Facial Recognition** — Local recognition of familiar individuals

3. Sample Workflow

- **Step 1: Power On** Turn on the smartglass and launch the mobile app.
- **Step 2: First-Time Setup** On-screen instructions guide the user to pair the app with the device.
- **Step 3: Auto-Connect (Later Use)** In future sessions, simply turning on both the glass and app enables automatic pairing.
- **Step 4: Voice Activation** Say commands like *"Hey Glass, what's the weather today?"*
- **Step 5: Backend AI Processing**
  - Voice input is converted to text.
  - The app sends the text to the AI agent (Gemini).
  - AI processes and returns a response.
- **Step 6: Output Display** The response is played via speaker or shown on the HUD display.

4. Communication Architecture

- **Rasberry Pi as Local Server:** Hosts FastAPI server on LAN (mobile hotspot)
- **Flutter App:** Sends voice/API requests to Pi, handles Gemini and cloud APIs
- **HUD Display Output:** Responses (like object labels, navigation, weather) are rendered on the smartglass HUD using reflected optics
- **Speaker Output:** Audio responses (e.g., weather reports, object names) are played through onboard speaker on the glass
- **Camera and Microphone Input:** Used for real-time object and facial recognition, and captures user's voice commands
- **Cloud API Communication:** App communicates with cloud services (Gemini, Maps, Spotify) via HTTPS requests

5. Implementation Strategy

- **Phase 1:** Building software architectures and testing them modularly
- **Phase 2:** Developing complex functionalities such as Machine Learning
- **Phase 3:** Integration with hardware components
- **Phase 4:** Prototype testing and validation
- **Phase 5:** Planning and allocating scope for future enhancements

6. Modular Breakdown

- **Voice Assistant** — Wake detection, Gemini integration
- **HUD Engine** — Display mirroring, font scaling, notifications
- **Vision Inference** — TFLite object recognition, local face match
- **Media + Maps** — App integration, HUD rendering
- **Wi-Fi Manager** — Pairing, fallback AP, network switching

7. Hardware Components

- **Raspberry Pi Zero 2 W** — Core computation unit with Wi-Fi, GPIO, and camera support.
- **2.4" 320x240 TFT Display** — Reflective HUD with sharp text and real-time output.
- **Transparent Acrylic Sheet (45)** — Optical combiner for monocle-style HUD.
- **Pi Camera Module** — Vision tasks like object/facial recognition, translation input.
- **MEMS Microphone + Mini Speaker** — Voice commands, feedback, and AI assistant communication.
- **Portable 5V Battery Pack** — Ensures mobility and rechargeability.

8. Software Tech Stack

Smartglass (Raspberry Pi):

- Python + FastAPI for device logic and server-based control
- OpenCV, TensorFlow Lite (TFLite), face_recognition for ML vision tasks
- ALSA for audio, GPIO libraries for peripherals

Mobile App (Flutter):

- Dart + Flutter for UI/UX
- Gemini API for assistant, translation, vision
- Google Maps, Spotify integrations

9. Use Case Scenarios

- A visually impaired user walks into a room — glass identifies objects aloud
- A traveler sees a foreign sign — snaps and sees translation on HUD
- A cyclist checks navigation while listening to music — all hands-free
- Student enters the class — says “Hey Glass, who’s in the room?” — Smartglass uses facial recognition to identify and name known individuals present

10. Future Enhancements

- Gesture controls via IMU sensors (hands-free control)
- Object memory system: Gemini-assisted room scanning to remember items
- Audio call integration (mic+speaker routing)

11. Improvements Over Existing Technology

- Minimal hardware, **max software innovation**
- Affordable prototyping — no proprietary platforms
- On-demand ML to avoid overheating
- Offline-first design using Raspberry Pi as a local server
- Built for real-world problems: vision, translation, mobility

12. Conclusion

- Shows voice-driven control, ML vision, and natural UI
- Modular, testable, and scalable for future add-ons
- Goal: Deliver a reliable, user-centric, AI-assisted demo