**Front End**

**🎨 Suggested UI Design (React.js Dashboard)**

**🔹 Left Panel (User Info & Controls)**

✅ **User Profile** (Divyangjan’s name, caregiver details)  
✅ **Brain Signal Settings** (Adjust EEG sensitivity, panic mode thresholds)  
✅ care giver profile  
Session start end profile

**🔹 Main Panel (Live Data & Alerts)**

📌 **Real-time Speech-to-Text** (Google STT, multi-language, with highlighted custom keywords)  
📌 **Live EEG Visualization** (Brainwave graph using Recharts/D3.js)  
📌 **Mind-Controlled UI (EEG-based interactions)** (Speed up/slow text scrolling)  
📌 **Panic Mode Alerts** (Triggered by stress EEG signals, auto-photo capture, send last 15 lines to WhatsApp)  
📌 **Live Camera View** (Show surroundings when panic mode is triggered)  
✅ **Speech-to-Text Settings** (Enable/disable transcription, language selection)  
✅ **Custom Keyword Setting area** (Train numbers, names, emergency words)  
🎭 **Sentiment Pie Chart (Live speech emotion analysis)**  
☁ **Word Cloud (Most frequently spoken words)**  
🔥 **EEG Anomaly Heatmap (Detect seizures/stress)**

**🔹 Bottom Panel (Device & Network Status)**

📡 **Raspberry Pi Connection Status**  
⚡ **Battery/Power Monitoring**  
📶 **Internet Connectivity Check**

**🖥️ UI Design Mockup (Concept)**

🟦 **Header:** "Chetana – Empowering Divyangjans"  
📜 **Real-time text scrolling box (speech-to-text output)**  
📊 **EEG Signal Graph (Live brainwave activity)**  
🚨 **Alert Box (When keyword or panic signal is detected)**  
📷 **Live Camera View (Only in panic mode)**  
⚙️ **Settings Panel (Adjust speech & brainwave preferences)  
Graph:** 📊 **Live EEG Line Graph** (Real-time brainwave monitoring)

**🚀 Tech Stack for the Dashboard**

✔ **Frontend:** React.js + Tailwind CSS  
✔ **Backend:** Node.js (WebSockets for real-time updates)  
✔ **Data Handling:** Firebase/MongoDB (Store settings & logs)  
✔ **Visualization:** Recharts/D3.js (EEG graphs)  
✔ **Communication:** twilio API (Alerts)

**Schema For Database**

**💡 Final Required Database Models**

| **No** | **Model Name** | **Purpose** | **Required?** |
| --- | --- | --- | --- |
| 1 | **User / Divyangjan Profile Model** | Stores user identity, caregiver contact, display preferences, languages | ✔️ Mandatory |
| 2 | **Custom Keyword Alert Model** | Stores custom keywords that trigger alerts, panic, or WhatsApp notification | ✔️ Mandatory |
| 3 | **EEG Session / Brain State Log Model** | Stores brain signal values (optional if you want to show graphs of focus, panic, etc.) | Optional (Good for visualization) |
| 4 | **Speech Text Log Model** | Stores live speech-to-text output lines to be used for last-15-lines WhatsApp alert | ✔️ Mandatory |
| 5 | **Emergency Event Log Model** | Logs whenever an emergency event is triggered (panic, alert, picture clicked) | Optional (Recommended for system auditability) |

**Required models For Chetna**

**✅ Total AI / ML Models You Need for Project Chetana**

**Based on your concept, real-time features, and hackathon time limitation, here is the correct and full list of AI/ML models needed:**

| **Model No.** | **ML Model** | **Purpose** | **Required?** |
| --- | --- | --- | --- |
| **1** | **Speech-to-Text Model** | **Converts audio to text (You are using Google Cloud Speech-to-Text API, so this is pre-trained, you don't need to make it)** | **✔️ Mandatory (External)** |
| **2** | **EEG Brain State Classifier** | **Classifies EEG signals into:** |  |

* **Normal**
* **Panic**
* **Focus**
* **Drowsy  
  (You will make a simple ML model like Decision Tree, SVM, or KNN trained on alpha, beta, gamma values) | ✔️ Mandatory (You will train this custom) |**

**✅ Optional / Nice-to-have Models (if time permits)**

| **Model No.** | **Optional Model** | **Purpose** | **Required?** |
| --- | --- | --- | --- |
| **3** | **Brain Signal Intensity Model** | **To estimate signal intensity for adjusting text speed (slower / faster display based on user focus or drowsiness)** | **Optional (Can skip if no time)** |
| **4** | **Emergency Event Prediction Model** | **Predicts possible emergency situations by analyzing combined EEG + speech activity (panic, accident detection)** | **Optional (too big for 24hr hackathon)** |
| **5** | **Full Session Report Generator (Graphical Summary)** | **Automatically generates an overall session graph (EEG + speech trend) at the end** | **Optional (Leave for future)** |

**✅ In Hackathon you should limit to**

**✔️ Speech-to-Text (Google Pre-trained)  
✔️ EEG Classifier (Small custom ML model you will train)  
✔️ Optional simple logic-based (non-ML) adjustment of text speed using EEG signal trends.**

**✅ Final Recommended ML/AI Stack**

| **Purpose** | **Model Type** | **Comment** |
| --- | --- | --- |
| **STT** | **Pre-trained (Google STT API)** | **Production Ready** |
| **EEG Classifier** | **Custom Light ML Model** | **Build this one yourself (can even be Decision Tree / KNN)** |

**⚡ Note:**

**Even just these 2 models are enough to win the hackathon if you focus on reliability + real-time behavior.**

**Do you want me also to prepare for you:  
1️⃣ EEG Classifier Architecture Suggestion  
2️⃣ Preprocessing Steps  
3️⃣ Sample dataset idea  
4️⃣ Model Training Flow for your project**

**Reply *Yes give me*, and I will show you the fastest way to build it in hackathon time 🟣**

**Back end Features Functionality list and technology used :  
For your project Chetana (Real-time AR headset for Divyangjans with EEG + Audio + Emergency + Dashboard)**

**✅ Full Backend Features**

| **Feature** | **Description** | **Purpose** |
| --- | --- | --- |
| **1** | **User Registration & Management** | **Register Divyangjan and Caregiver, Update Profile and multilanguage support** |
| **2** | **Custom Keyword System** | **Add/Delete/Update custom words like Train 1234, Help, Divyangjan** |
| **3** | **Live Audio Stream Handling** | **Accept audio stream from Pi** |
| **4** | **Google Speech-to-Text Integration** | **Converts audio to text using Google STT** |
| **5** | **Custom Keyword Detection** | **Detect custom keyword in live text** |
| **6** | **Live Text Logging** | **Store all transcribed text line-by-line** |
| **7** | **Recent Text Fetcher** | **Fetch last 15 lines for WhatsApp alert** |
| **8** | **EEG Data Logging** | **Store incoming EEG signals from Pi** |
| **9** | **EEG Classifier** | **Detect Panic / Normal / Focus from EEG using ML model** |
| **10** | **Panic Event Handler** | **On Panic: Click Picture + Save + Send WhatsApp** |
| **11** | **WhatsApp Alert** | **Send emergency alert with last 15 text lines** |
| **12** | **LED / OLED Trigger** | **Display alerts on AR Headset using I2C** |
| **13** | **Event Logging** | **Store events like panic, picture clicked, alert sent** |
| **14** | **Dashboard Data Feed** | **Serve EEG & Text data to React Dashboard via WebSocket** |
| **15** | **Session Start / End** | **Control to Start and Stop Session** |

**✅ Full Suggested API Endpoints**

| **API** | **Method** | **Description** |
| --- | --- | --- |
| **/api/user/register** | **POST** | **Register Divyangjan + Caregiver** |
| **/api/user/:id** | **GET** | **Get User Profile** |
| **/api/user/:id** | **PUT** | **Update User Profile** |
| **/api/keyword/add** | **POST** | **Add a Custom Keyword** |
| **/api/keyword/delete/:id** | **DELETE** | **Delete a Custom Keyword** |
| **/api/keyword/list** | **GET** | **Get All Custom Keywords** |
| **/api/audio/stream** | **POST** | **Send audio chunks (Pi to backend)** |
| **/api/text-log/new** | **POST** | **Store Live Transcribed Text** |
| **/api/text-log/recent** | **GET** | **Get Last 15 Text Lines** |
| **/api/eeg-data/new** | **POST** | **Store Incoming EEG Signal** |
| **/api/eeg-data/recent** | **GET** | **Get Recent EEG Data (for graph)** |
| **/api/eeg/classify** | **POST** | **Classify EEG into Normal / Panic / Focus** |
| **/api/event-log/new** | **POST** | **Store New Event (Panic / Photo / Alert)** |
| **/api/event-log/history** | **GET** | **Get All Past Events** |
| **/api/camera/capture** | **POST** | **Click Picture + Store** |
| **/api/whatsapp/send** | **POST** | **Send WhatsApp Emergency Alert** |
| **/api/session/start** | **POST** | **Start a Session**  **Language selection** |
| **/api/session/end** | **POST** | **End a Session** |

**✅ Real-time via WebSocket**

**For:**

* **Live EEG graph on Dashboard**
* **Live Speech-to-Text display**
* **Live Panic Alerts  
  WebSocket endpoint will broadcast these.**

**✅ Notes**

**✔️ This design is fully suitable for your hardware (Pi0, OLED, INMP441, ADS1115, AD8232)  
✔️ Supports real-time audio, EEG, text, and alerts  
✔️ Perfect balance for hackathon-time + scalable for post-hackathon project  
✔️ No missing feature as per your idea ✅**

**If you want, I will now make:**

1. **API Payload Structures**
2. **Full Backend Folder Structure (Node.js / Express)**
3. **Minimal EEG Classifier Code**
4. **WebSocket Live Flow**

**Reply *Yes make full backend* and I'll continue step-by-step to make sure you are hackathon-ready without wasting time.**

**Hardware processing and components and flow of execution:**

**✅ Complete List of Hardware + One Line Purpose + Flow of Execution**

**✅ Full Hardware List**

| **Hardware** | **Purpose** |
| --- | --- |
| **Custom EEG Electrodes** | **To collect raw EEG signals from the forehead or scalp** |
| **ADS620 (Analog Signal Conditioner)** | **Preprocess (amplify, filter) raw EEG signals before digitization** |
| **ADS1115 (16-bit ADC)** | **Converts analog EEG signals to digital signals for Arduino** |
| **Arduino Uno / Nano** | **Reads digitized EEG data and forwards to ML model for classification (via serial or USB)** |
| **Raspberry Pi Zero W** | **Main controller for Audio, Camera, Display, Networking (WiFi/Bluetooth)** |
| **INMP441 Microphone** | **I2S Digital microphone to capture real-time audio for Speech-to-Text** |
| **Google Speech-to-Text (Cloud API)** | **Converts audio into text with multi-language support** |
| **Camera Module for Pi Zero** | **Captures images on panic event for emergency documentation** |
| **0.96" OLED Screen (I2C)** | **Displays live speech-to-text, alerts, EEG status, and panic messages** |
| **LED Indicator** | **Provides a physical LED alert on keyword detection or panic signal** |
| **WiFi + Bluetooth (Pi0 Built-in)** | **To send real-time data to Dashboard + Backend and enable remote communication** |
| **Caregiver Mobile Phone (WhatsApp)** | **To receive emergency alerts and images during panic events** |

**✅ Flow of Execution**

**🟣 1. EEG Processing Flow**

**pgsql**

**CopyEdit**

**[EEG Electrodes]**

**↓**

**[ADS620 - Amplify & Filter]**

**↓**

**[ADS1115 - Analog to Digital Conversion]**

**↓**

**[Arduino - Reads EEG Data]**

**↓**

**[Arduino - ML Model Classifies EEG State]**

**↓**

**[Arduino → Raspberry Pi via Serial or USB]**

**↓**

**[Pi → Backend API / WebSocket]**

**↓**

**[React Dashboard + OLED Screen] => Show EEG Graphs + State (Normal / Panic)**

**↓**

**If Panic →**

**- Trigger Camera**

**- Save Picture**

**- Send Picture + Last 15 Texts to Caregiver via WhatsApp**

**- Alert on OLED & LED**

**🟣 2. Audio to Text Flow**

**css**

**CopyEdit**

**[INMP441 Microphone]**

**↓**

**[Raspberry Pi Zero]**

**↓**

**[Google Speech-to-Text API]**

**↓**

**[Pi receives Transcribed Text]**

**↓**

**[Backend - Stores Text in DB]**

**↓**

**[Backend - Keyword Matching]**

**↓**

**If Custom Keyword Detected →**

**- Trigger OLED Screen Alert**

**- Light up LED**

**- Send to React Dashboard**

**✅ Both flows run independently, but during PANIC (from EEG) both flows can collaborate to:**

**✅ Send Image + Last 15 Lines of Transcribed Text to Caregiver via WhatsApp  
✅ Update Dashboard & LED + OLED in Real-Time**