**Project Requirement Document (PRD)**

**Project Title:** Chetana - AI-Powered Smart AR Glasses for Divyangjans

**Version:** V2.0 | March 2025  
Prepared by Mayank @ Dash 3.0

**Prepared For:** Hackathon (24 Hours Challenge)

**1. Objective**

Chetana aims to empower Divyangjans (persons with disabilities) by providing a wearable AR system capable of real-time speech-to-text conversion, brain-state detection (using EEG), and emergency alerting via WhatsApp. The system helps hearing-impaired and paralyzed users to perceive spoken information, trigger panic alerts through brain signals, and notify caregivers instantly.

**2. Key Features**

**Frontend (React.js Dashboard)**

**Left Panel**

* User Profile (Name, Caregiver Details)
* EEG Settings (Adjust sensitivity, panic thresholds)
* Caregiver Profile
* Session Controls (Start / End Session)

**Main Panel**

* Real-time Speech-to-Text Display (Google STT, multilingual, keyword highlighting)
* EEG Brainwave Graph (Realtime visualization using Recharts)
* Mind-Controlled UI (EEG based interaction, adjust scrolling speed)
* Panic Mode (EEG Triggered):
  + Auto Photo Capture
  + Send last 15 lines + photo to Caregiver via WhatsApp
* Live Camera Feed (During Panic Mode)
* Speech Settings (Language selection, transcription on/off)
* Custom Keyword Configuration (Add/Delete Emergency Keywords)
* Sentiment Pie Chart (Speech Emotion Analysis)
* Word Cloud (Frequent Words Visualization)
* EEG Anomaly Heatmap (Stress / Seizure Detection)

**Bottom Panel**

* Device Status (Pi Connection, Battery, Internet)

**3. Backend Features (Node.js + Express + WebSockets)**

* User & Caregiver Management
* Custom Keyword System
* Real-Time Audio Stream Handler
* Google STT API Integration
* Text Logging (Live + Recent Fetcher)
* EEG Data Logging & Classifier (Normal / Panic / Focus)
* Panic Event Handling (Photo Capture, Alert, WhatsApp Notification)
* OLED & LED Trigger
* Event Logging (Panic, Alerts, Session)
* Dashboard Live Feed (EEG + STT + Events)
* Session Control (Start / End)

**4. Database Models**

1. User Model (Divyangjan & Caregiver Profiles)
2. Custom Keyword Model
3. EEG Session Log (Optional)
4. Speech Text Log
5. Emergency Event Log (Optional)

**5. Machine Learning Models**

|  |  |  |
| --- | --- | --- |
| **Model** | **Purpose** | **Status** |
| Google Speech-to-Text | Convert audio to text | Pre-Trained |
| EEG Classifier | Detect Normal / Panic / Focus | Custom (Train with EEG Dataset) |
| Text Speed Adjuster | Adjust scrolling speed | Optional |
| Emergency Predictor | Predict critical events | Optional |

**6. Hardware Stack**

|  |  |
| --- | --- |
| Component | Purpose |
| EEG Electrodes | Capture Brain Signals |
| ADS620 | Amplify & Filter EEG |
| ADS1115 | ADC for EEG Signals |
| Arduino UNO | Read EEG, Preprocess, Forward Data |
| Raspberry Pi Zero | Main Controller (Audio, Camera, Display, Networking) |
| INMP441 | Capture Audio |
| OLED Display | Show Transcribed Text + Alerts |
| LED | Panic or Keyword Indicator |
| Camera | Capture image during panic |
| Caregiver Mobile | Receives WhatsApp Alert |

**7. Flow of Execution**

**EEG Flow**

1. EEG -> ADS620 -> ADS1115 -> Arduino
2. Arduino detects brain state -> Sends to Raspberry Pi
3. Pi forwards data to Backend + Dashboard + OLED
4. If Panic:
   * Capture Image
   * WhatsApp Alert (Photo + Last 15 Transcripts)
   * OLED / LED Alert

**Audio Flow**

1. INMP441 -> Pi Zero
2. Pi -> Google STT API
3. Backend:
   * Store Text
   * Keyword Detection
   * Live Display on Dashboard
   * Alert if Keyword detected

**8. APIs Overview**

|  |  |  |
| --- | --- | --- |
| API | Method | Function |
| /api/user | POST | Register User & Caregiver |
| /api/user/:id | GET | Get User Profile |
| /api/keyword | POST | Manage Custom Keywords |
| /api/audio/stream | POST | Audio Streaming |
| /api/text-log | POST/GET | Store / Fetch Speech Logs |
| /api/eeg-data | POST/GET | EEG Data Store / Recent |
| /api/eeg/classify | POST | Brain State Classification |
| /api/event-log | POST/GET | Event Logging |
| /api/camera/capture | POST | Trigger Camera |
| /api/whatsapp/send | POST | Send Alert |
| /api/session/start | POST | Start Session |
| /api/session/end | POST | End Session |

**9. Tech Stack**

* Frontend: React.js + Tailwind CSS
* Backend: Node.js + Express + WebSocket
* Database: MongoDB / Firebase
* Visualization: Recharts / D3.js
* Speech API: Google Cloud STT
* Messaging API: Twilio
* Hardware: Raspberry Pi Zero, Arduino Uno, ADS1115, INMP441, OLED, LED

**10. Notes**

* Keep EEG Classifier simple (Decision Tree / KNN)
* Prioritize Panic Mode and Real-Time Features
* Sentiment Analysis and Heatmap are Optional
* Prepare pre-trained dataset for EEG Classifier before hackathon