The Great Hack
Techfluence Hackathon 2023

TEAM DELTA

- Atharva Parkhe (LinkedIn, GitHub, Instagram, 8007609672)
- Pratham Shankhwalker (LinkedIn,GitHub, Instagram, 9307051204)
- Medam Greeshma (LinkedIn,GitHub, Instagram, 7972321283)

PROBLEM STATEMENT

How can we utilise wearable devices and sensors to monitor health indicators like heart rate, blood pressure, and glucose levels, and notify users or caregivers of any abnormalities or emergencies?

PROJECT NAME

VitalSense

VISION

- Creating a Web Application with a user friendly interface to connect all the wearable devices.
- Leveraging the data from the sensors of the wearable to detect abnormalities
- Notifying caregivers during an abnormalities/emergency situation.

APPROACH

- Creating a Login-Signup system for the user, which allows the user to create an account and setup all his wearables.
- Making a dashboard for data visualisation and activity tracking using the sensors data.
- Leveraging the sensor/s data to detect abnormalities and emergency situations, using various algorithms and machine learning models.
- Notifying the caregivers/loved ones whenever any abnormality/emergency situation is detected.

FEATURES

- **Wearables Data Visualisation**
  - Data Visualisation of wearable devices data.

- **Activity Tracking**
  - This feature gives us the status about the current activity of the user using Machine Learning.
  - The ML model classifies the current activity of the user ie : sleeping, sitting, walking, running by using features like height, weight, resting heart rate, steps , calories burnt, etc.

- **Abnormality Detection**

1. **Arrhythmia (Atrial Fibrillation) Classification**
   - Arrhythmia is a condition in which heart beats with irregular/abnormal rhythm.
   - Our Machine Learning model predicts different arrhythmia types of on ECG data collected by the wearable.
     I. **Non Ectopic Beats:** These are normal type of beats which do not imply any abnormality.
     II. **Supra-ventricular Ectopic Beats:** These beats are considered to be abnormal and do not indicate any emergency.
     III. **Ventricular Ectopic Beats:** These beats are considered to be abnormal as well as indicates an emergency. In this case, a notification will be sent to the caregivers.
     IV. **Fusion Beats:** These are the beats when supra-ventricular and ventricular beats coincide and produce a hybrid mixture. These beats are considered as abnormal but are not a sign of an emergency.

2. **Hypertension Prediction**
   - Hypertension is a situation of abnormal blood pressure.
   - Our Machine Learning Model predicts if the user is possibly undergoing hypertension or not.
   - The model used data like age, gender, blood pressure, fasting blood sugar, arrhythmia class, etc to predict if a person is going through hypertension or not.

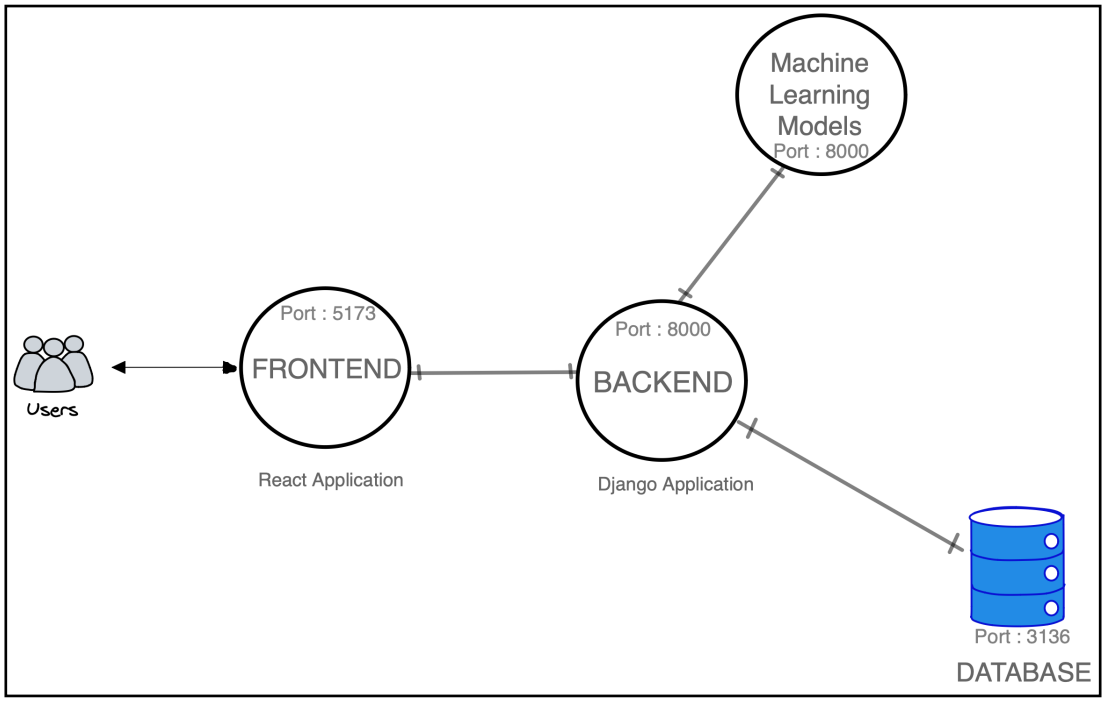
3. **Abnormal Glucose Level (normal 70-140 mg/dL)**
   - **Hypoglycaemia:** It is a condition in which the blood sugar level falls below 70 mg/dL. The situation is alarming but not an emergency situation.
   - **Hyperglycaemia:** It is a condition in which the blood sugar lever goes above 140 mg/dL. The situation is alarming but not an emergency situation.

     · In both these cases an alert message will be sent to the user.

4. **Abnormal Heart Rate (normal 60-100 BPM)**
   - **Tachycardia:** It is a condition where in the heart rate goes above 100 BPM. The situation is alarming and our system will issue an alert to the user.
   - **Bradycardia:** It is a condition where the heart rate falls below 60 BPM.The situation is alarming and our system will issue and alert to the user.

- **Cardiac Arrest:** It is a situation where the heart stops beating. This is a case of emergency and all the emergency contacts (family) will get notified.

SYSTEM ARCHITECTURE



MACHINE LEARNING MODELS

1. Arrhythmia (Atrial Fibrillation) Classification using Convolutional Neural Networks.
2. Hypertension Detection Model using Random Forest Classification
3. User Activity prediction model using xgboost model

- Linear Regression using Sci-Kit Learn, Pandas and Numpy

TEACH STACK

- **Frontend :** Django (Python)
- **Backend :** React (Java Script)
- **Machine Learning :** Pandas, Numpy, Pickle, Sci-Kit Learn

API REFERENCE

https://api.postman.com/collections/25119294-f74c4f03-13d8-4975-a569-81b3286b881a?access_key=PMAT-01GW8ZM4P62QT3Q005GPRRX01E

TEACH STACK

https://github.com/Intensa-Council-GEC/Team-Delta

DEVELOPER MACHINE SETUP

1. Run everything in docker containers
   - Install Docker (Link)
2. To run each service locally
   - Install Python (Installation Link)
   - Install Node (Installation Link)

DEVELOPER MACHINE SETUP

1. Run using docker compose
   - **STEP#1:** git clone https://github.com/Intensa-Council-GEC/Team-Delta && cd Team-Delta
   - **STEP#2:** docker-compose up
   - **STEP#3:** Open browser http://localhost:3000/
2. Run services individually
   1. **STEP#1:** git clone https://github.com/Intensa-Council-GEC/Team-Delta && cd Team-Delta
   2. **STEP#2:** To run Backend Server
      1. cd machine-learning
      2. virtualenv env
      3. On Windows Machine
         - env/Scripts/activate
      4. On Linux Machine
         - source env/bin/activate
      5. pip install -r requirements.txt
      6. python3 manage.py runserver
3. **STEP#3:** To run Frontend
   1. cd frontend
   2. npm install –legacy-peer-deps
   3. npm run dev

REFERRAL LINKS

1. **React JS Documentation:** https://reactjs.org/docs/getting-started.html
2. **Django Documentation:** https://docs.djangoproject.com/en/4.1/
3. **Scikit Learn Documentation:** https://scikit-learn.org/0.21/documentation.html
4. **Numpy Documentation:** https://numpy.org/doc/stable/