TEAM B1-01
Paleti Krishnasai - CED18I039 [Team Lead]
Uppalapati Pranita - CED18I062
Hrishikesh Rajesh Menon - COE18B024
Preety Banjare - COE17B038

EPIONE

RECOVER TO A HEALTHY LIFE

Virtual Medical Assistant

Project Idea

PROBLEM STATEMENT

To analyze and maintain a healthy lifestyle in the younger-middle age generation by identifying the diabetes risk factor.

PRODUCT DESCRIPTION:-

We at EPIONE aim to tackle the lack of healthy lifestyle in children through our products, the EPIONE BAND and the EPIONE HEALTHCARE APP.

The band measures a child's physical state in real time and provides this information to the child in the form of an LED Heart. The band has a bunch of styles to choose from.Through the use of Physical games and an online leaderboard system we create an interest in the child to be more physically active.

The EPIONE HEALTHCARE is an application that is given to the parents. It stores and provides nearly accurate information on the child's health measured by the band ,keeps track of the data and provides a specialized report to the user.

Validate Requirements :

- Cloud Service -Dataset computation
- Dataset for Model Training
- Microweb framework
- Heart rate monitor,PPG sensor)
- Rigid plastics for chassis of the band
- Enclosures/ clasps for inside work
- Strap

Product Specifications :-

- Product takes in Real Time Data and calculates a rating which is then represented in a visual form for easier interpretation.
- A WebApp that
  - generates a report when asked by the user for a fixed time basis.
  - Provides nutritional and physical plans depending on calculated rating.

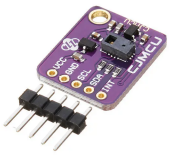
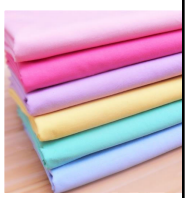
MVP:-

As creating the band is currently not possible we are creating a WebApp that takes in manually inputted data and

- generates a summarized report when asked by the user for a fixed time basis.
- The Webapp is an **Medical Virtual Assistant** outputs the risk factor of diabetes based on certain minimum viable parameters as inputs and gives suitable advice on moving forward.

BILL OF PRODUCT:-

Physical

Name of Material	Image	Price/Product	Status
Heart rate monitor		Rs.240 per Product	On Hold
ECG Sensor Module		Rs.130 per Product	On Hold
SpO2 Sensor		Rs.617 per Product	On Hold
Gesture sensors		Rs.637.43 per Product	On Hold
Rigid plastics(PVC) (0.008kg)		Rs 109.64 per Kg Rs. 8.77 per Product	On Hold
Carbon steel(0.025kg)		Rs 87.84 per Kg Rs.21.96 per Product	On Hold
biodegradable fibre(cloth based or silicone elastomers) (0.15m)		Rs. 50 per meter Rs 7.5 per product	On Hold

Digital

Name of Material	Image	Price
Kaggle Datasets		Opensource
Cloud Services - AWS		Frequent Access Tier, Over 500 TB / Month $0.021 per GB
Cloud Services , Cloud hosting - Heroku		$25 per month
Micro Web Framework - Flask		OpenSource

Jupyter Notebook , Google Collab .		OpenSource
HTML5, CSS3, JS		OpenSource
Python Libraries		OpenSource
Machine Learning Algorithms		OpenSource

Project Plan with Appropriate Work Breakdown

1. Data Collection
    a. Collecting Proper Datasets for our MVP,through open source websites.

2. Data Visualization
    a. Descriptive Statistics
    b. Statistical analysis
    c. Visualizing data distribution.

3. Data Preprocessing
    a. Data Cleaning
    b. Data handling(missing/corrupted).

4. Data Model Creation
    a. Feature Selection
    b. Using Machine Learning Algorithms we can
        i. Risk index generation, predictive modelling.
        ii. As well as for health care ratings.
        iii. Performance testing.

5. Web App Development
    a. Using HTML,CSS,JavaScript to create a visually appealing web application to house our backend
    b. Using the Flask Web Framework we will build the MVP using our data models.

Task Assignment

Week	Task Alloted	Working Member
Week 1	Project Description, Requirements and Specifications and MVP	Krishnasai,Hrishikesh, Pranita,Preety
Week 2	Project Plan,Work Breakdown, Task Assignment.	Krishnasai,Hrishikesh, Pranita,Preety
Week 3	Data Collection from kaggle and segregation.	Krishnasai,Hrishikesh, Pranita, Preety
Week 4	Descriptive Statistics Statistical analysis	Krishnasai
	Data distribution visualization	Hrishikesh
	Research on various algorithms that can be potentially used.	Preety
	Research on optimal algorithms and feature selection .	Pranita

Week 5	Conceptual Frontend Design	Hrishikesh
	Data Cleaning	Preety
	Data Handling	Pranita
	Feature selection	Krishnasai
Week 6	Setting up the required environment features for the webapp build.	Hrishikesh
	Predictive modelling	Krishnasai
	Training the chosen algorithm with the selected features .	Preety
	Testing the accuracy and precision of the trained model using the test data.	Pranita
Week 7	Risk index Generation	Krishnasai

	Performance Testing.	Pranita
	Coding the Frontend	Hrishikesh
	Defining the routing logic of the web pages using Flask	Preety
Week 8	Coding the backend	Krishnasai
	Coding the Frontend	Hrishikesh
	Responsive web design and choosing the appropriate colors for better user Experience UX	Preety
	Integrating the back end and front end of the web app and synchronisation using Flask and Python	Pranita
Week 9	Finishing up the Frontend	Hrishikesh Pranita

	Enabling two way communication between frontend and backend.	Preety
	Model Optimization if possible.	Krishnasai
Week 10	Integrating Frontend,Backend and the model trained.	Hrishikesh, Preety, Krishnasai,Pranita
Week 11	Testing and running the web app on the localhost machine.	Hrishikesh , Preety
	Webapp testing and verification.	Pranita,Krishnasai

Weekly Update:-

Week 1:- Tasks Done

- Product Description has been provided
- Requirements and Specifications has been Provided
- MVP has been Defined

Status - After week 1:-

- **Completed** the allotted tasks.

Week 2:- Tasks Done

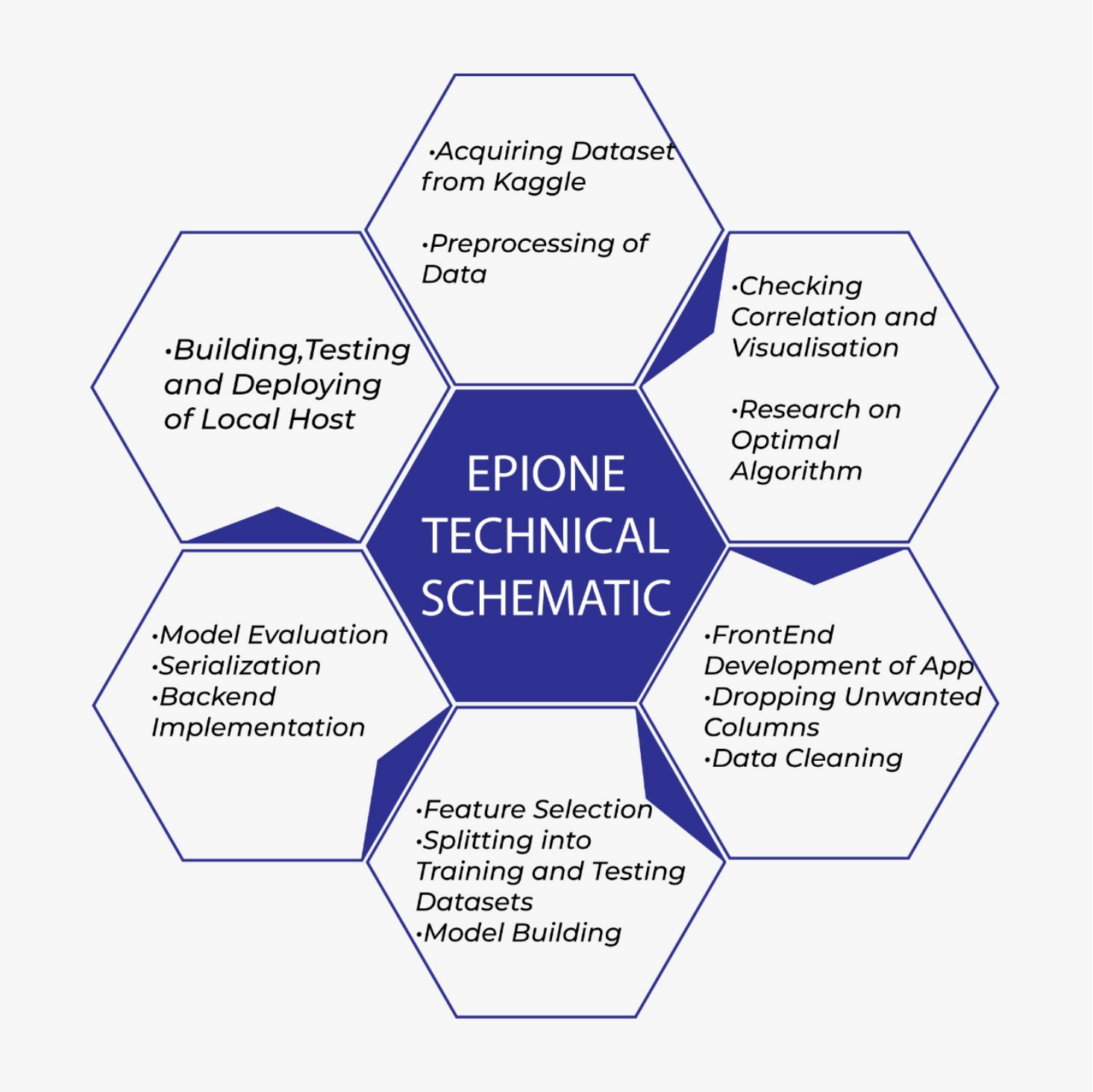
- Project Plan and Workflow has been created
- Tasks have been Allocated between members

Status - After week 2:-

- **Completed** the allotted tasks.

Week 3 - Tasks Done

Technical Schematic



BOM Status

Name of Material	Status
Kaggle Datasets	**Downloaded- Tested**
Micro Web Framework - Flask	**Downloaded and Ready**
Jupyter Notebook , Google Collab .	**Downloaded and running**
HTML5, CSS3, JS	**Downloaded and running**
Python Libraries	**Downloaded and running**
Machine Learning Algorithms	**Work in progress**

Since the proposed MVP can be built using open source frameworks and libraries, the **finalized BOM** stands at zero as all the above mentioned resources are accessible for free through the internet.

Allotted Tasks : -

- Collecting Proper Datasets which are having better suitability for our MVP,through open source websites. Dataset link

Status - After week 3:-

- **Completed** the allotted tasks.

Week 4 - Tasks Done

Allotted Tasks : -

- **Data Analysis**

In [3]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [4]:
```
dataset.isnull().sum()
```

Out[4]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

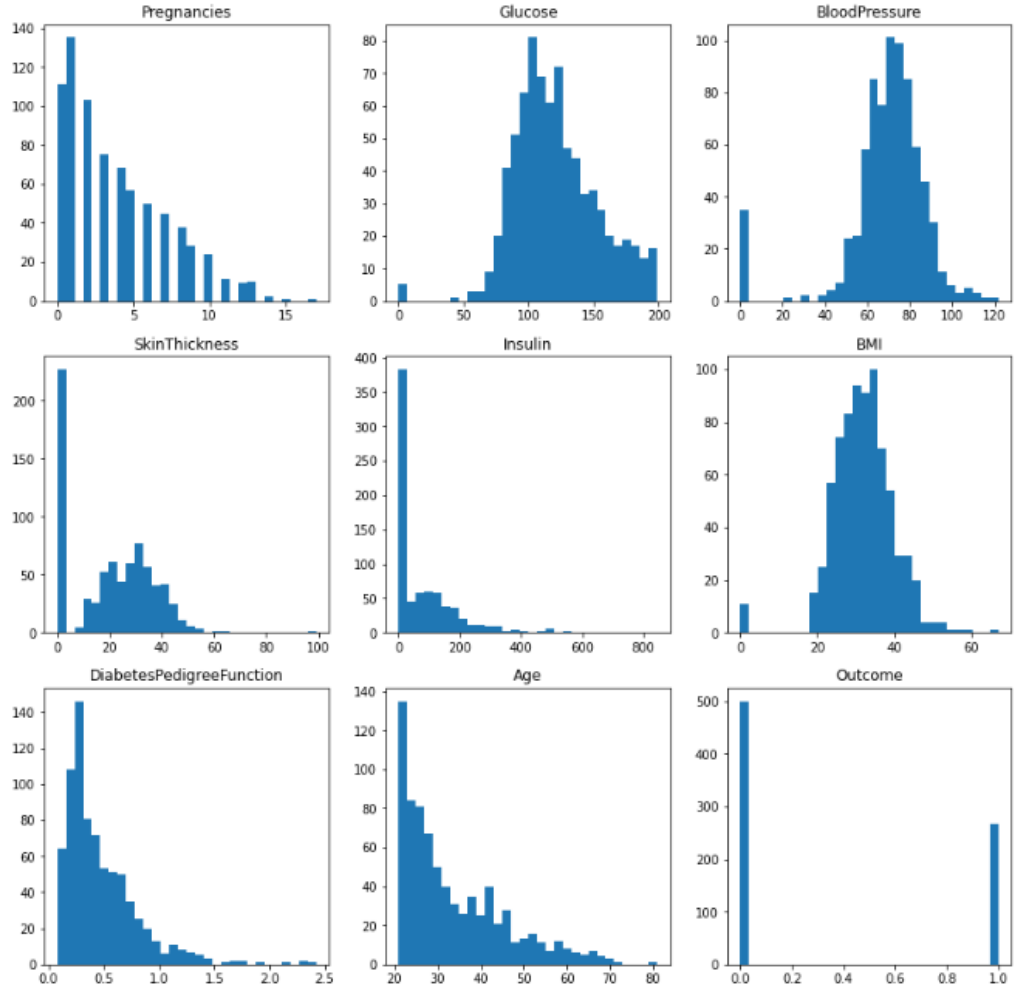
In [5]:
```
dataset.describe()
```

Out[5]:

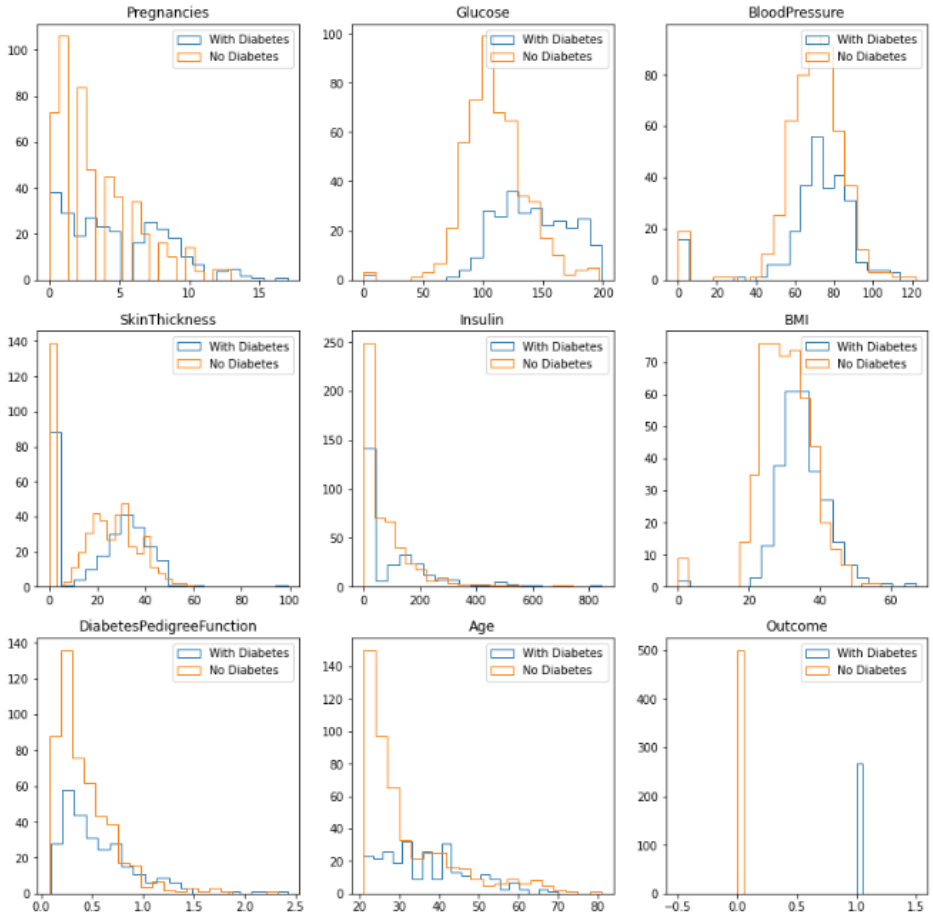
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

- **Data Visualization**

In [6]:
```
cols = dataset.columns.tolist()
fig = plt.figure(figsize=(14,14))

for position, col in enumerate(cols):
    ax = fig.add_subplot(3,3,position+1)
    ax.hist(dataset[col], bins=30)
    plt.title(col)

plt.show()
```

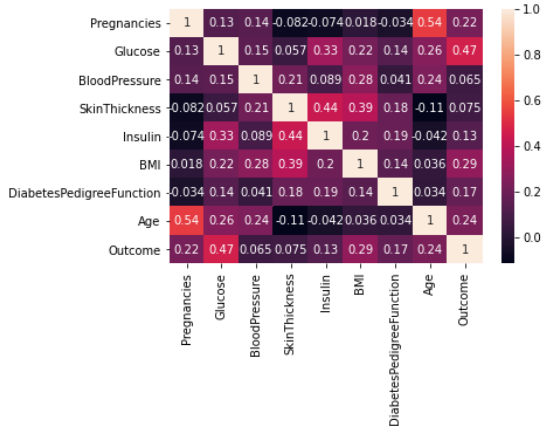


In [8]:
```
fig = plt.figure(figsize=(14,14))
for position, col in enumerate(cols):
    ax = fig.add_subplot(3,3,position+1)
    ax.hist(with_diabetes[col], histtype ='step', label='With Diabetes', bins = 20)
    ax.hist(without_diabetes[col], histtype ='step', label = 'No Diabetes', bins=20)
    ax.legend(loc=0)
    plt.title(col)

plt.show()
```



In [9]:
```
import seaborn as sns
```

In [10]:
```
sns.heatmap(dataset.corr(), annot = True)
plt.show()
```



- # Research on optimal algorithms and feature selection

After taking into consideration algorithms such as Naive Bayes and Logistic Regression .

Click here for detailed documentation

Why we chose Logistic Regression

Naïve Bayes is a classification method based on Bayes' theorem that derives the probability of the given feature vector being associated with a label. Naïve Bayes has a naive assumption of conditional independence for every feature, which means that the algorithm expects the features to be independent which is not always the case.

Logistic regression is a linear classification method that learns the probability of a sample belonging to a certain class. Logistic regression tries to find the optimal decision boundary that best separates the classes.

The learning mechanism is a bit different between the two models, where Naive Bayes is a generative model and Logistic regression is a discriminative model. What does this mean?

Generative model: Naive Bayes models the joint distribution of the feature X and target Y, and then predicts the posterior probability given as P(y|x)

Discriminative model: Logistic regression directly models the posterior probability of P(y|x) by learning the input to output mapping by minimising the error.

Naïve Bayes assumes all the features to be conditionally independent. So, if some of the features are in fact dependent on each other (in case of a large feature space), the prediction might be poor.

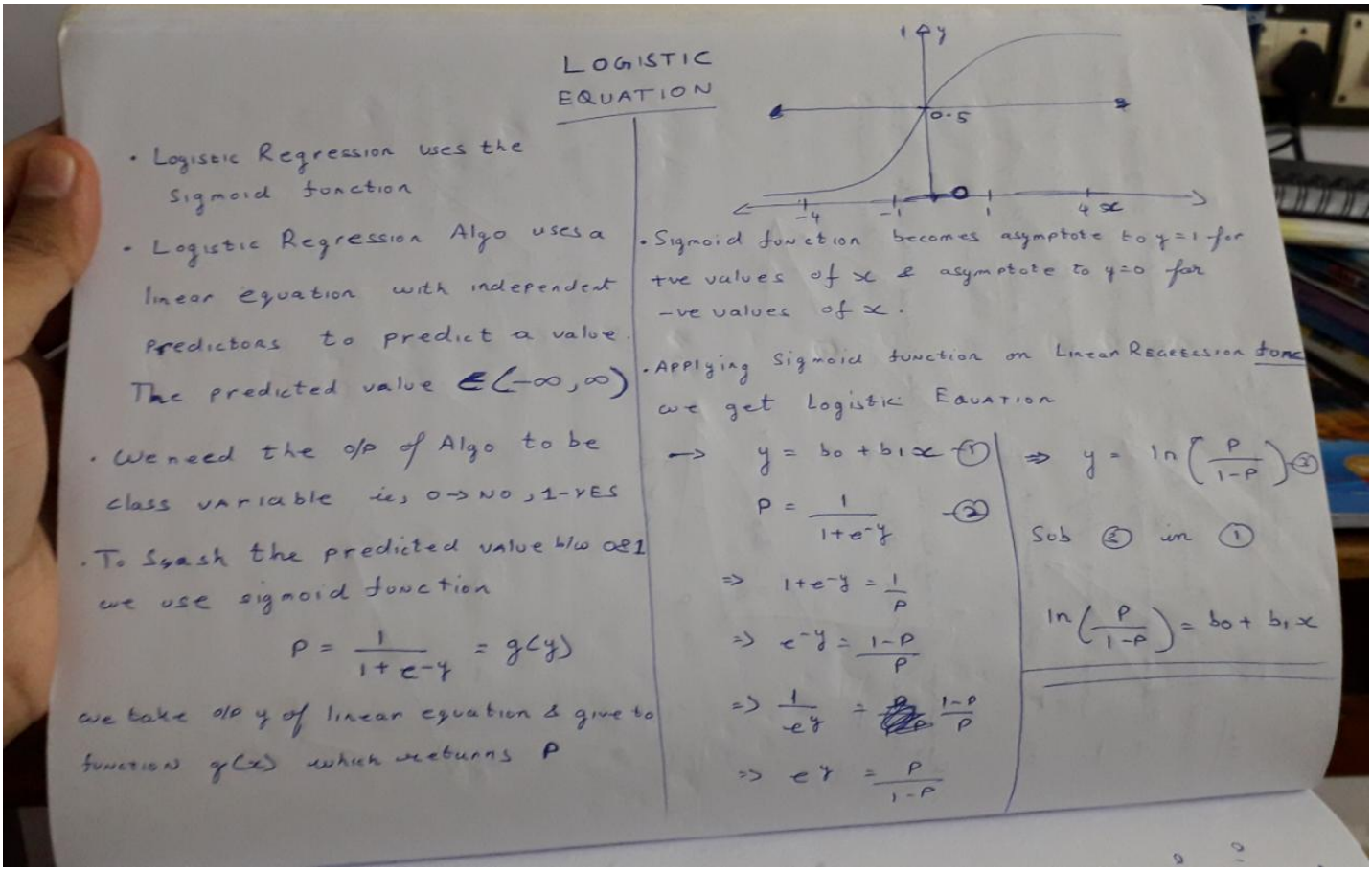
Logistic regression splits feature space linearly, and typically works reasonably well even when some of the variables are correlated.

Given the correlated feature space which is following a normal distribution after cleaning up of noisy samples , and given that logistic regression is a generalized linear model which works well with correlated features, LOGISTIC REGRESSION was chosen as a more compatible algorithm for the MVP dataset.

Feature Selection

The Pregnancies column contains information on the number of pregnancies, but while there is a correlation between diabetes likelihood and number of pregnancies, our data-set does not actually distinguish between males and females. If, for instance, a male is more likely to have diabetes based on other indicators, we don't want the prediction for him having diabetes being marked down just because his number of Pregnancies is 0. To avoid the problem of the predictive model being confounded by this, we will get rid of the Pregnancies column.

There is also the fact that we have very limited understanding on how the DiabetesPedigreeFunction column's values are determined. This column measures an individual's hereditary predisposition to diabetes. Realistically, it is not going to be possible for an individual to input this measurement into our predictive model. We'll thus drop this column as well



Status - After week 4:-

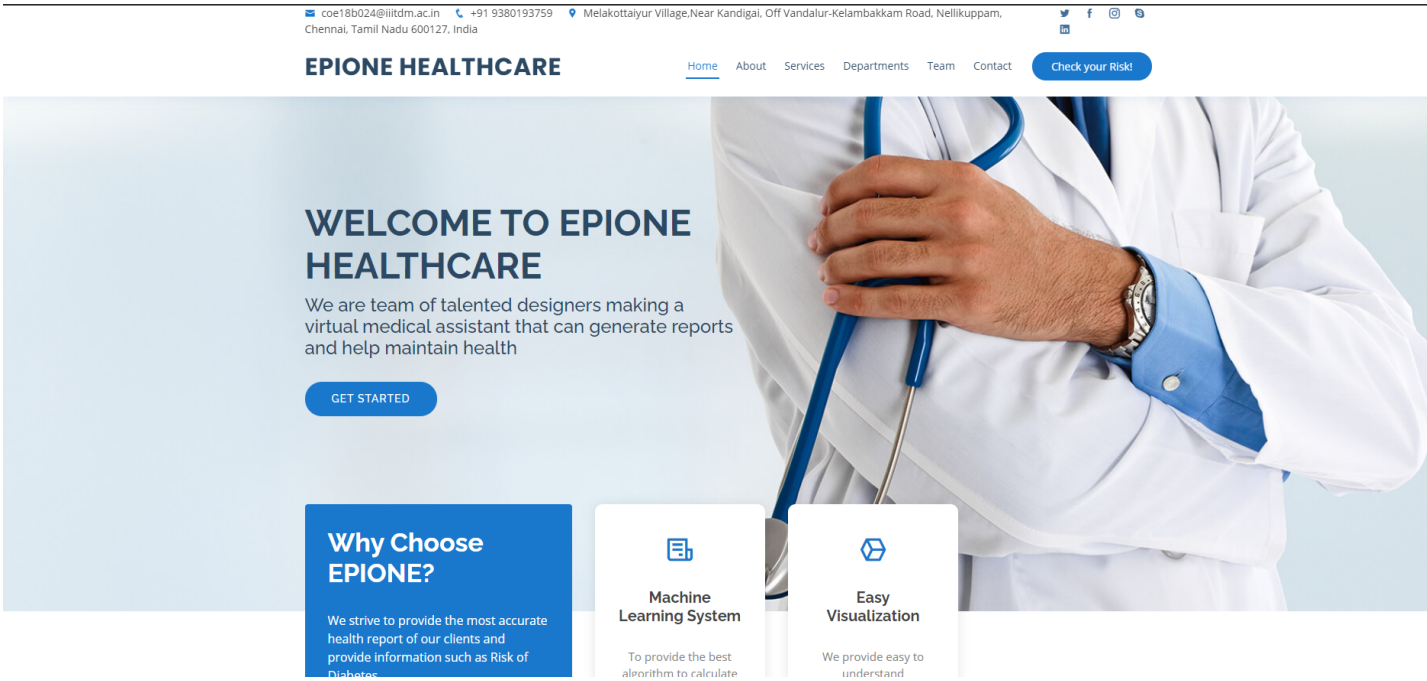
Completed the allotted tasks.

Design Pipeline - 27/01/21 (20% Progress)

•Data Collection and Preprocessing
(Status: Completed)
• Data Analysis and Visualization
(Status: Completed)
• Data Trimming and Cleaning
(Status: Ongoing)
• Algorithm Selection
(Status: Ongoing)
•FrontEnd Work
(Status: Ongoing)
•Data Model Training and Testing
(Status: Upcoming)
•Performance Testing and Bug Identifying
(Status: Upcoming)
•Backend Implementation and UI/UX Design
(Status: Upcoming)
•Front to Back Integration
(Status: Upcoming)
•Testing, Optimizing and Deployment
(Status: Upcoming)

Week 5 - Tasks Done

- **Conceptual Frontend Design**



Using HTML,CSS,and JavaScript we have created an interactive webpage for our MVP

- **Data Cleaning and Data Handling**

Dropping Duplicate Rows

In [11]:
```
dataset = dataset.drop_duplicates(keep='first')

dataset.shape
```

Out[11]: (768, 9)

Handling incomplete Data

In [12]:
```
dataset = dataset.drop(dataset[dataset['BMI']==0].index)
dataset = dataset.drop(dataset[dataset['BloodPressure']==0].index)
dataset = dataset.drop(dataset[dataset['Insulin']==0].index)
dataset = dataset.drop(dataset[dataset['Glucose']==0].index)
dataset = dataset.drop(dataset[dataset['SkinThickness']==0].index)

dataset.shape
```

Out[12]: (392, 9)

- **Feature Selection**

Feature Selection

The Pregnancies column contains information on number of pregnancies, but while there is a correlation between diabetes likelihood and number of pregnancies, our data-set does not actually distinguish between males and females. If, for instance, a male is more likely to have diabetes based on other indicators, we don't want the prediction for him having diabetes being marked down just because his number of Pregnancies is 0. To avoid the problem of the predictive model being confounded by this, we will get rid of the Pregnancies column.

There is also the fact that we have very limited understanding on how the DiabetesPedigreeFunction column's values are determined. This column measures an individual's hereditary predisposition to diabetes. Realistically, it is not going to be possible for an individual to input this measurement into our predictive model. We'll thus drop this column as well

In [13]:
```
features = cols.copy()
features.remove('Outcome')
features.remove('DiabetesPedigreeFunction')
features.remove('Pregnancies')


print(features)
```

```
['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'Age']
```

Status - After week 5:-

Completed the allotted tasks.

Design Pipeline - 03/02/21 (40% Progress)

•Data Collection and Preprocessing
(Status: Completed)
• Data Analysis and Visualization
(Status: Completed)
• Data Trimming and Cleaning
(Status: Completed)
• Algorithm Selection
(Status: Completed)
•FrontEnd Work
(Status: Ongoing)
•Data Model Training and Testing
(Status: Upcoming)
•Performance Testing and Bug Identifying
(Status: Upcoming)
•Backend Implementation and UI/UX Design
(Status: Upcoming)
•Front to Back Integration
(Status: Upcoming)
•Testing, Optimizing and Deployment
(Status: Upcoming)

Week 6 - Tasks Done

- **Setting up the required environment features for the webapp build.**

```
Dell@DESKTOP-O3TH7JO MINGW64 ~/new-project
$ pipenv install requests
Installing requests…
Adding requests to Pipfile's [packages]…
Installation Succeeded
Pipfile.lock not found, creating…
Locking [dev-packages] dependencies…
Locking [packages] dependencies…
[=== ] Locking..Success!
Updated Pipfile.lock (444a6d)!
Installing dependencies from Pipfile.lock (444a6d)…

Dell@DESKTOP-O3TH7JO MINGW64 ~/new-project
$ pipenv install flask
Installing flask…
Adding flask to Pipfile's [packages]…
Installation Succeeded
Pipfile.lock (9234af) out of date, updating to (444a6d)…
Locking [dev-packages] dependencies…
Locking [packages] dependencies…
[    ] Locking..Success!
Updated Pipfile.lock (9234af)!
Installing dependencies from Pipfile.lock (9234af)…
```

- **Predictive modelling**

Predictive Modelling - Logistic Regression

lets use Logistic Regression.

The error metric we'll be focusing on for now is accuracy (total correct predictions/total predictions).

We will assess its accuracy with KFold Cross Validation, to use our data with maximum efficiency.

In [15]:
```
warnings.filterwarnings("ignore")

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
```

In [16]:
```
# Instantiate the model
weight = {
    1:1.05,
    0:1
}
log = LogisticRegression(class_weight = weight)

kf = KFold(n_splits=6)
score = cross_val_score(log, dataset[features], dataset['Outcome'], cv=kf, scoring='accuracy')
print(score)
print("The mean accuracy is:",score.mean())
```

```
[0.81818182 0.71212121 0.72307692 0.83076923 0.87692308 0.78461538]
The mean accuracy is: 0.790947940947941
```

Using KFold cross validation with slightly more weight placed on a positive check for Diabetes, with 6 folds, we get an accuracy of around 79%.

We'll train this model on the entire data-set, and then enter in a custom values to see the probabilities of Diabetes assigned to us.

- **Training the chosen algorithm with the selected features**

```
In [17]: weight = {
             1:1.05,
             0:1
         }

         log_model = LogisticRegression(class_weight = weight)
         log_model.fit(dataset[features], dataset['Outcome'])
         log_model.predict_proba(np.array(dataset[features].iloc[0]).reshape(1, -1))

Out[17]: array([[0.95759497, 0.04240503]])

In [18]: dataset[features].iloc[0]

Out[18]: Glucose          89.0
         BloodPressure    66.0
         SkinThickness    23.0
         Insulin          94.0
         BMI              28.1
         Age              21.0
         Name: 3, dtype: float64

In [19]: testing1 = [120,10,100,190,50,60]
         features

         log_model.predict_proba(np.array(testing1).reshape(1, -1))[0,1]

Out[19]: 0.9478842255658008

In [20]: features

Out[20]: ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'Age']
```

- **Testing the accuracy and precision of the trained model using the test data.**

```
kf = KFold(n_splits=6)
score = cross_val_score(log, dataset[features], dataset['Outcome'], cv=kf, scoring='accuracy')
print(score)
print("The mean accuracy is:",score.mean())
```

```
[0.81818182 0.71212121 0.72307692 0.83076923 0.87692308 0.78461538]
The mean accuracy is: 0.790947940947941
```

Using KFold cross validation with slightly more weight placed on a positive check for Diabetes, with 6 folds, we get an accuracy of around 79%.

Status - After week 6:-

Completed the allotted tasks.

Design Pipeline - 17/02/21 (50% Progress)

•Data Collection and Preprocessing
(Status: Completed)
• Data Analysis and Visualization
(Status: Completed)
• Data Trimming and Cleaning
(Status: Completed)
• Algorithm Selection
(Status: Completed)
•FrontEnd Work
(Status: Ongoing)
•Data Model Training and Testing
(Status: Completed)
•Performance Testing and Bug Identifying
(Status: Ongoing)
•Backend Implementation and UI/UX Design
(Status: Upcoming)
•Front to Back Integration
(Status: Upcoming)
•Testing, Optimizing and Deployment
(Status: Upcoming)

Week 7 - Tasks Done

- ## Risk index Generation

define a function which receives inputs for the 6 feature columns, and then returns a Diabetes risk index.

In [21]:
```
def diabetes_risk_prediction(glucose, bp, skinthickness, insulin, bmi, age):
    indicator_list = [glucose, bp, skinthickness, insulin, bmi, age]
    predictions = log_model.predict_proba(np.array(indicator_list).reshape(1, -1))
    risk = predictions[0,1]
    print("-"*len("Health Indicator Analysis"))
    print("Health Indicator Analysis")
    print("-"*len("Health Indicator Analysis"))
    if risk < 0.3:
        print("You are probably in good health, keep it up.")
    elif risk > 0.7:
        print("See a doctor as soon as you can . You might be on the way to developing diabetes if you don't change

    elif risk > 0.9:
        print("Go to a hospital. Odds are high you have diabetes.")

    return print("Your Diabetes Risk Index is {:.2f}/50.".format(risk*0.5*100))

diabetes_risk_prediction(90,80,30,120,60,50)
```

```
-------------------------
Health Indicator Analysis
-------------------------
Your Diabetes Risk Index is 32.81/50.
```

- ## Performance Testing.

Re-instantiate the logistics regression and model and re-fit the data.

:
```
def diabetes_risk_prediction2(glucose, bp, skinthickness, insulin, bmi, age):
    weight = {
    1:1.05,
    0:1}

    log_model2 = LogisticRegression(class_weight = weight)
    log_model2.fit(dataset[features], dataset['Outcome'])

    indicator_list = [glucose, bp, skinthickness, insulin, bmi, age]
    predictions = log_model2.predict_proba(np.array(indicator_list).reshape(1, -1))
    risk = predictions[0,1]
    return risk

diabetes_risk_prediction2(100,80,30,120,30,45) #Test array.
```

:
```
0.22443666622161737
```

- **Defining the routing logic of the web pages using Flask**

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
from joblib import load
app = Flask(__name__)
# model = pickle.load(open('d.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
```

```
@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls trought request
    '''
    data = request.get_json(force=True)
    prediction = model.y_predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

if __name__ == "__main__":
    app.run(debug=True)
```

Status - After week 7:-

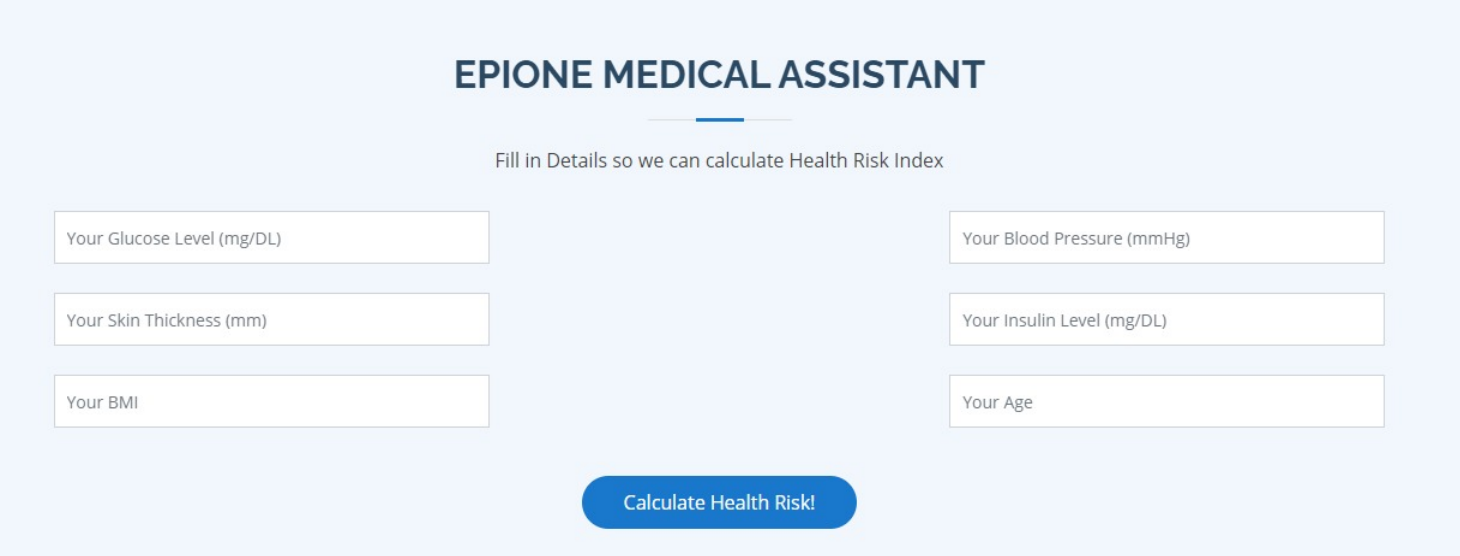
Completed the allotted tasks.

Design Pipeline - 24/02/21 (60% Progress)

•Data Collection and Preprocessing
(Status: Completed)
• Data Analysis and Visualization
(Status: Completed)
• Data Trimming and Cleaning
(Status: Completed)
• Algorithm Selection
(Status: Completed)
•FrontEnd Work
(Status: Ongoing)
•Data Model Training and Testing
(Status: Completed)
•Performance Testing and Bug Identifying
(Status: Completed)
•Backend Implementation and UI/UX Design
(Status: Upcoming)
•Front to Back Integration
(Status: Upcoming)
•Testing, Optimizing and Deployment
(Status: Upcoming)

Week 8 - Tasks Done

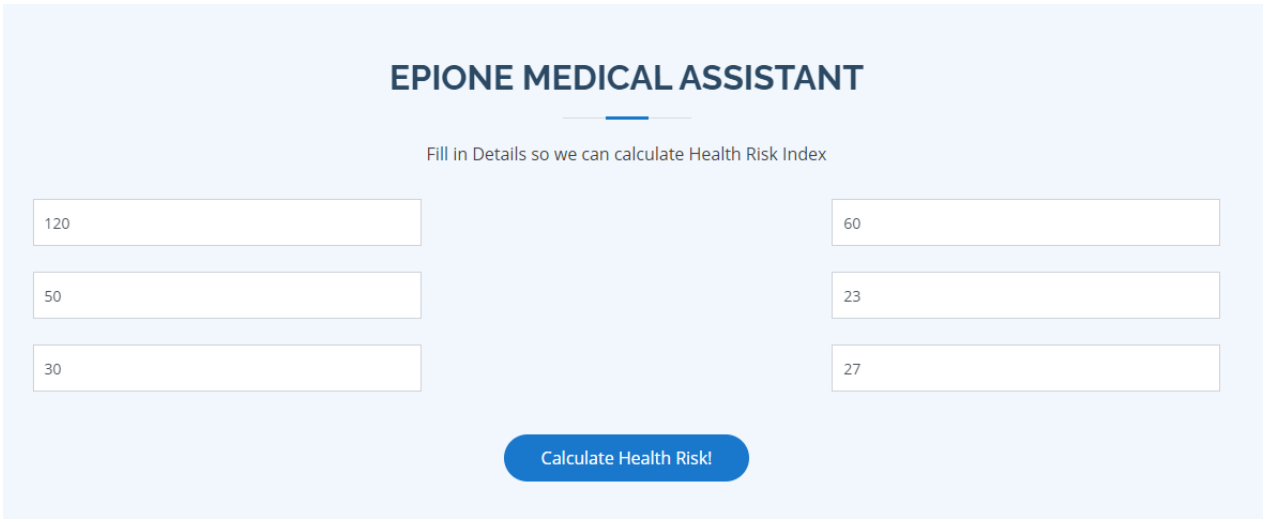
- **Front End Coding**
  - Form created of user input to the Risk Index Algorithm



- **Front End Reasoning**
  - **Color Psychology:-**
    - As our webapp is a medical assistant we use colors to appease certains emotions in the user
      - Blue:- We use the color blue to give the user the feeling of **calmness and loyalty** in trusting our product to give them accurate results
      - White:- We use this color to give the user the feeling of **cleanliness and simplicity** so as to not overcomplicate the procedure of the Web App
  - **Millers Law:-**
    - Miller's Law states that “The average person can only keep around 7 items in their working memory”
    - So following that rule we have simplified the procedure of our webapp by only requiring **6** pieces of information making it easier to follow.

- **Responsive Web Design:-**

- Backend Integration Using Flask:-



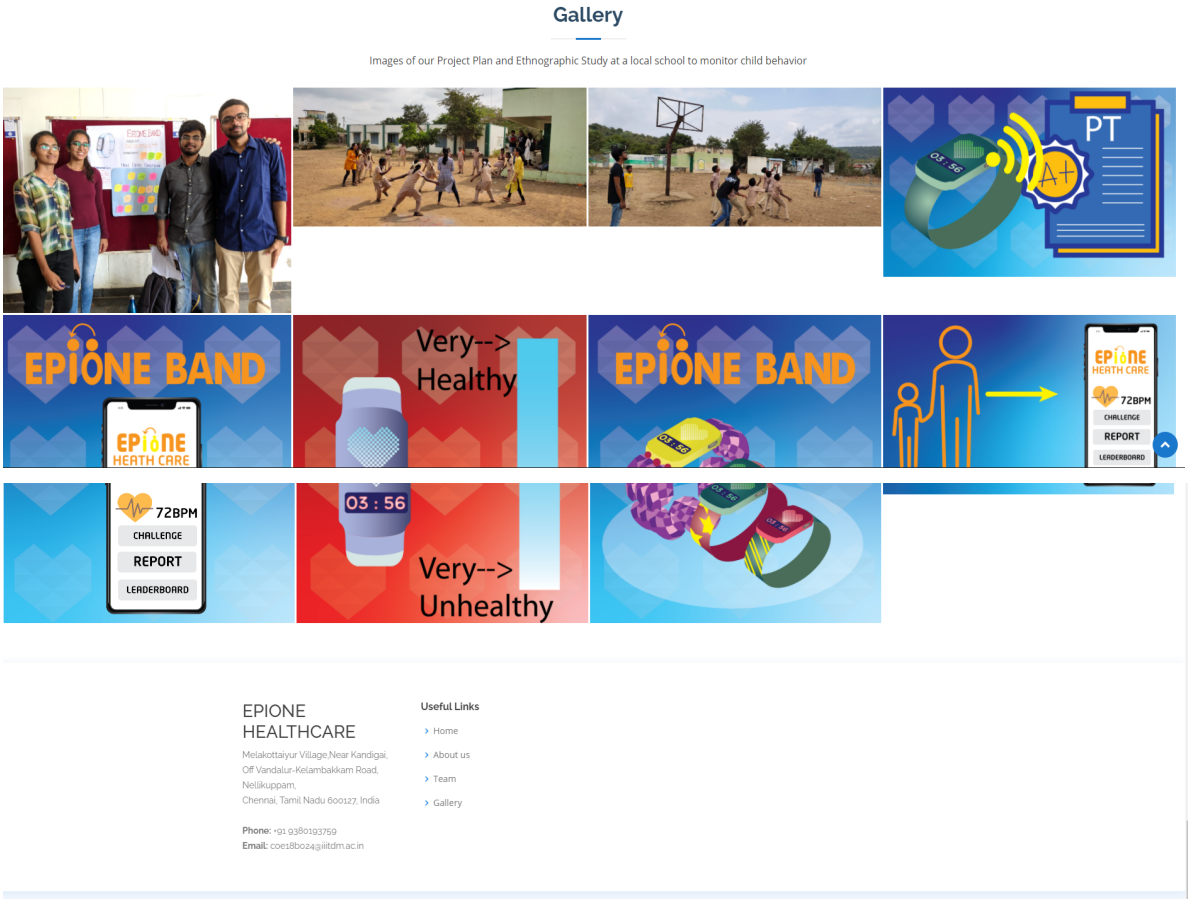

```
127.0.0.1 - - [31/Mar/2021 16:33:30] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [31/Mar/2021 16:33:30]
"GET/static/vendor/static/owl.carousel.min.css HTTP/1.1" 404 -
120 60 50 23 30 23
127.0.0.1 - - [31/Mar/2021 16:33:47] "POST /y_predict HTTP/1.1"
200 -
```

Design Pipeline - 3/03/21 (70% Progress)

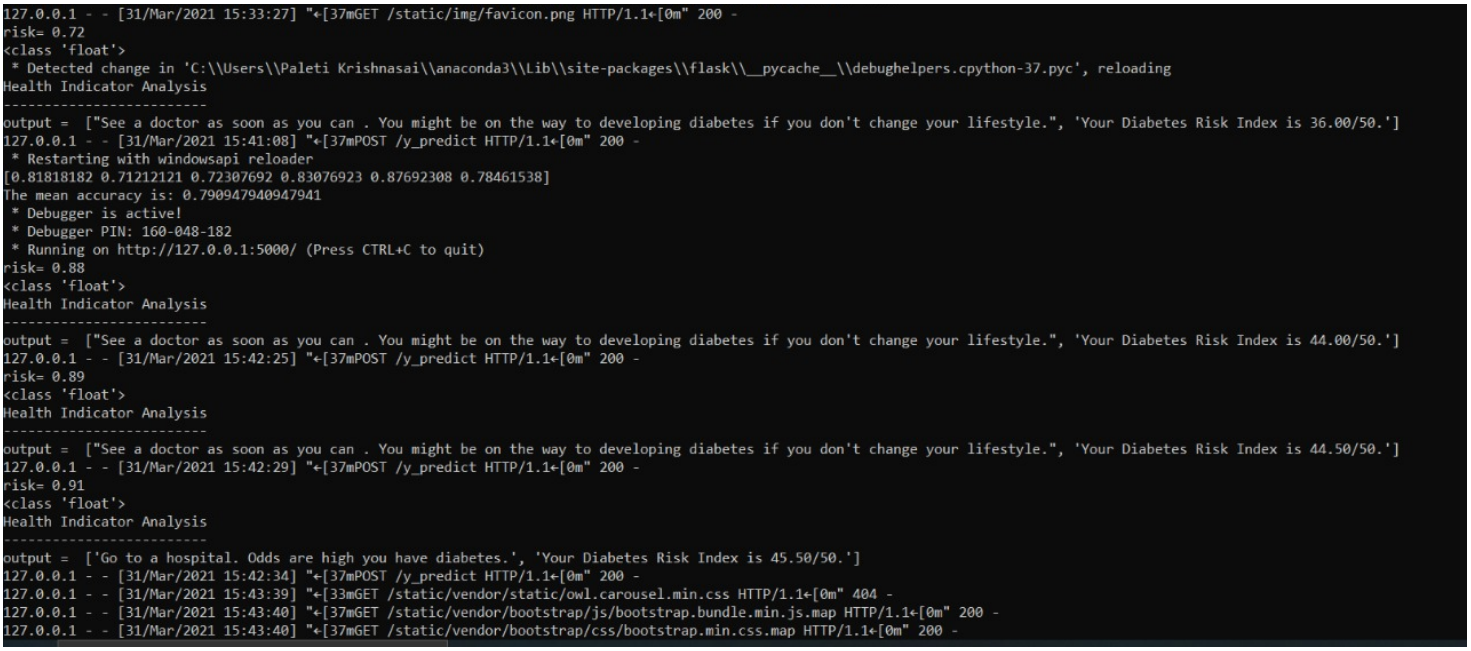
•Data Collection and Preprocessing
(Status: Completed)
• Data Analysis and Visualization
(Status: Completed)
• Data Trimming and Cleaning
(Status: Completed)
• Algorithm Selection
(Status: Completed)
•FrontEnd Work
(Status: Completed)
•Data Model Training and Testing
(Status: Completed)
•Performance Testing and Bug Identifying
(Status: Completed)
•Backend Implementation and UI/UX Design
(Status: Ongoing)
•Front to Back Integration
(Status: Ongoing)
•Testing, Optimizing and Deployment
(Status: Upcoming)

Week 9 - Tasks Done

- **Finishing Up Front End**



- **Enabling two way communication between frontend and backend.**



- **Model Optimization:-**

Final Accuracy of Model:-80%

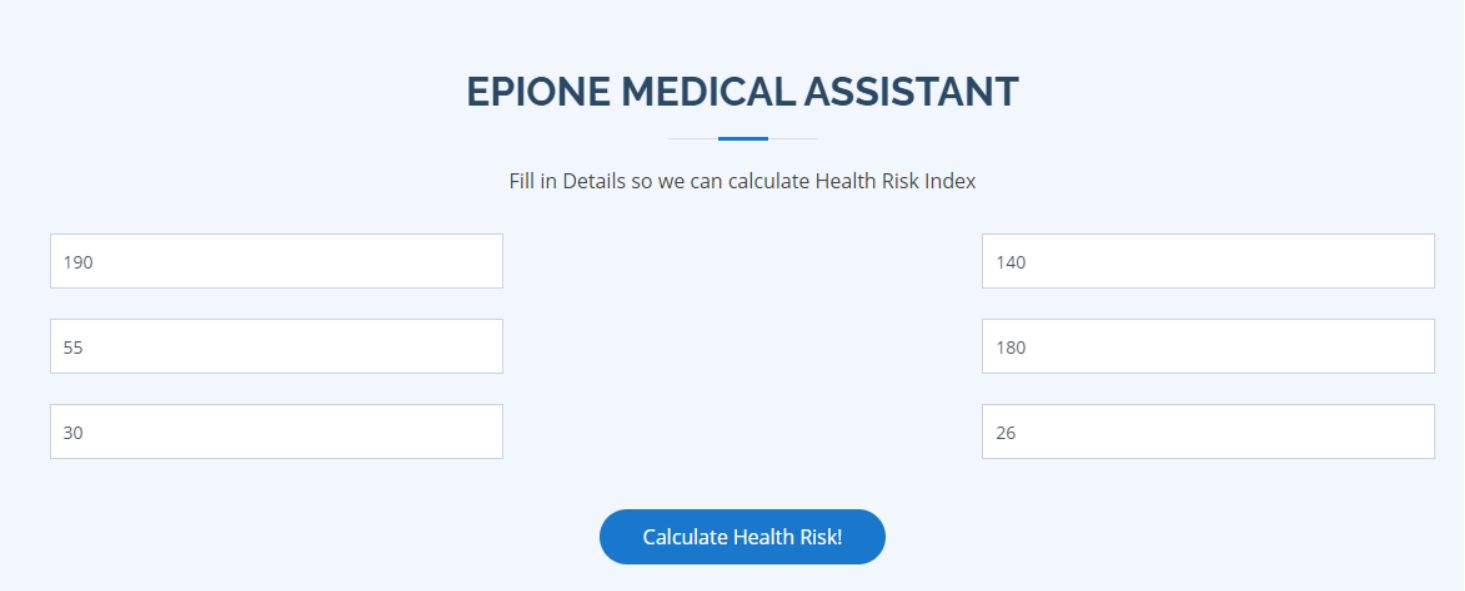
Design Pipeline - 10/03/21 (80% Progress)

•Data Collection and Preprocessing
(Status: Completed)
• Data Analysis and Visualization
(Status: Completed)
• Data Trimming and Cleaning
(Status: Completed)
• Algorithm Selection
(Status: Completed)
•FrontEnd Work
(Status: Completed)
•Data Model Training and Testing
(Status: Completed)
•Performance Testing and Bug Identifying
(Status: Completed)
•Backend Implementation and UI/UX Design
(Status: Completed)
•Front to Back Integration
(Status: Ongoing)
•Testing, Optimizing and Deployment
(Status: Ongoing)

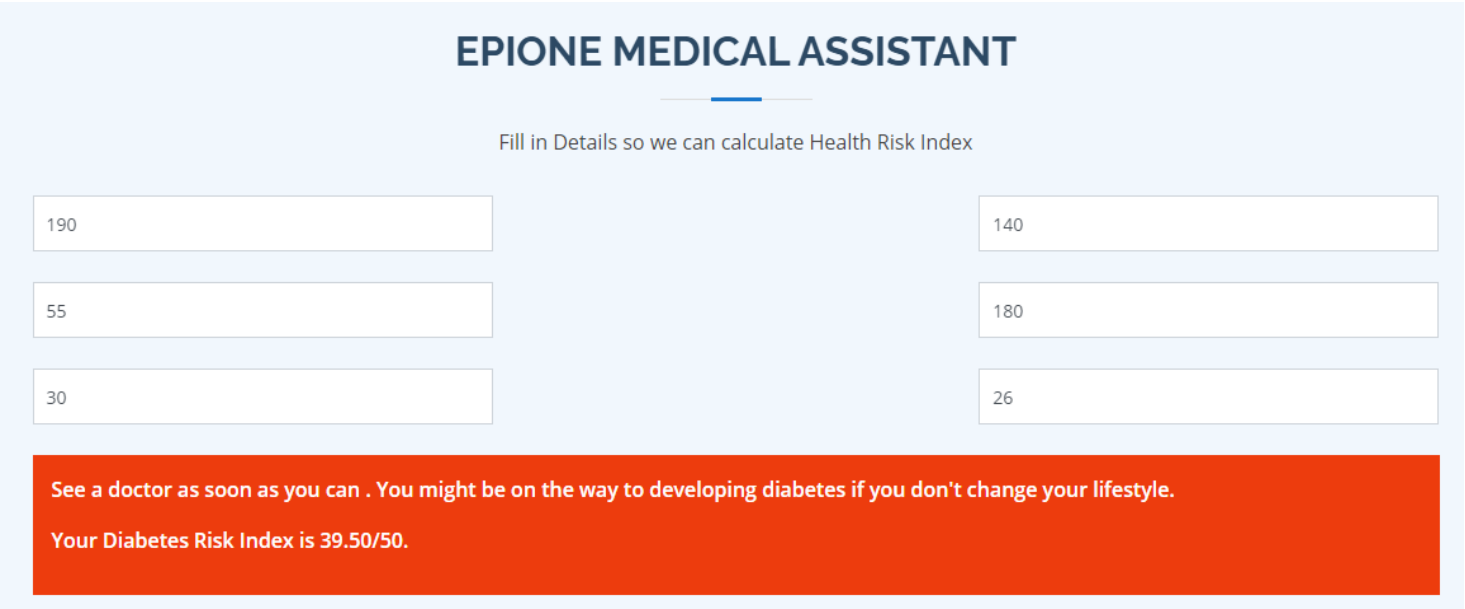
Week 10 - Tasks Done

- **Integrating Frontend,Backend and the model trained**

Input:-



Result:-



Design Pipeline - 17/03/21 (90% Progress)

•Data Collection and Preprocessing
(Status: Completed)
• Data Analysis and Visualization
(Status: Completed)
• Data Trimming and Cleaning
(Status: Completed)
• Algorithm Selection
(Status: Completed)
•FrontEnd Work
(Status: Completed)
•Data Model Training and Testing
(Status: Completed)
•Performance Testing and Bug Identifying
(Status: Completed)
•Backend Implementation and UI/UX Design
(Status: Completed)
•Front to Back Integration
(Status: Completed)
•Testing, Optimizing and Deployment
(Status: Ongoing)

Week 11 - Tasks Done

- **Testing and running the web app on the localhost machine:-**

```
  Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
[0.81818182 0.71212121 0.72307692 0.83076923 0.87692308 0.78461538]
The mean accuracy is: 0.790947940947941
 * Debugger is active!
 * Debugger PIN: 884-027-251
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Deployed:-

- **Webapp testing and verification:-**

Verification:-

```
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[36mGET /static/vendor/waypoints/jquery.waypoints.min.js HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[36mGET /static/img/about.jpg HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[36mGET /static/img/gallery/gallery-8.jpg HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[36mGET /static/vendor/jquery.easing/jquery.easing.min.js HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[36mGET /static/vendor/counterup/counterup.min.js HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[36mGET /static/js/main.js HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[36mGET /static/vendor/bootstrap-datepicker/js/bootstrap-datepicker.min.js HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[37mGET /static/vendor/boxicons/fonts/boxicons.ttf HTTP/1.1←[0m" 200 -
127.0.0.1 - - [31/Mar/2021 15:32:24] "←[36mGET /static/vendor/owl.carousel/owl.carousel.min.js HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:33:26] "←[37mGET / HTTP/1.1←[0m" 200 -
127.0.0.1 - - [31/Mar/2021 15:33:26] "←[33mGET /static/vendor/static/owl.carousel.min.css HTTP/1.1←[0m" 404 -
127.0.0.1 - - [31/Mar/2021 15:33:26] "←[37mGET /static/vendor/owl.carousel/owl.carousel.min.js HTTP/1.1←[0m" 200 -
127.0.0.1 - - [31/Mar/2021 15:33:26] "←[36mGET /static/vendor/icofont/fonts/icofont.woff2 HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:33:27] "←[36mGET /static/img/hero-bg.jpg HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:33:27] "←[36mGET /static/vendor/boxicons/fonts/boxicons.woff2 HTTP/1.1←[0m" 304 -
127.0.0.1 - - [31/Mar/2021 15:33:27] "←[37mGET /static/img/favicon.png HTTP/1.1←[0m" 200 -
risk= 0.72
<class 'float'>
 * Detected change in 'C:\\Users\\Paleti Krishnasai\\anaconda3\\Lib\\site-packages\\flask\\__pycache__\\debughelpers.cpython-37.pyc', reloading
Health Indicator Analysis
-------------------------
output =  ["See a doctor as soon as you can . You might be on the way to developing diabetes if you don't change your lifestyle.", 'Your Diabetes Risk Index is 36.00/50.']
127.0.0.1 - - [31/Mar/2021 15:41:08] "←[37mPOST /y_predict HTTP/1.1←[0m" 200 -
 * Restarting with windowsapi reloader
[0.81818182 0.71212121 0.72307692 0.83076923 0.87692308 0.78461538]
The mean accuracy is: 0.790947940947941
 * Debugger is active!
 * Debugger PIN: 160-048-182
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
risk= 0.88
<class 'float'>
Health Indicator Analysis
-------------------------
output =  ["See a doctor as soon as you can . You might be on the way to developing diabetes if you don't change your lifestyle.", 'Your Diabetes Risk Index is 44.00/50.']
127.0.0.1 - - [31/Mar/2021 15:42:25] "←[37mPOST /y_predict HTTP/1.1←[0m" 200 -
risk= 0.89
<class 'float'>
Health Indicator Analysis
-------------------------
output =  ["See a doctor as soon as you can . You might be on the way to developing diabetes if you don't change your lifestyle.", 'Your Diabetes Risk Index is 44.50/50.']
127.0.0.1 - - [31/Mar/2021 15:42:29] "←[37mPOST /y_predict HTTP/1.1←[0m" 200 -
risk= 0.91
<class 'float'>
Health Indicator Analysis
-------------------------
output =  ['Go to a hospital. Odds are high you have diabetes.', 'Your Diabetes Risk Index is 45.50/50.']
127.0.0.1 - - [31/Mar/2021 15:42:34] "←[37mPOST /y_predict HTTP/1.1←[0m" 200 -
127.0.0.1 - - [31/Mar/2021 15:43:39] "←[33mGET /static/vendor/static/owl.carousel.min.css HTTP/1.1←[0m" 404 -
127.0.0.1 - - [31/Mar/2021 15:43:40] "←[37mGET /static/vendor/bootstrap/js/bootstrap.bundle.min.js.map HTTP/1.1←[0m" 200 -
127.0.0.1 - - [31/Mar/2021 15:43:40] "←[37mGET /static/vendor/bootstrap/css/bootstrap.min.css.map HTTP/1.1←[0m" 200 -
```

Design Pipeline - 24/03/21 (100% Progress)

•Data Collection and Preprocessing
(Status: Completed)
• Data Analysis and Visualization
(Status: Completed)
• Data Trimming and Cleaning
(Status: Completed)
• Algorithm Selection
(Status: Completed)
•FrontEnd Work
(Status: Completed)
•Data Model Training and Testing
(Status: Completed)
•Performance Testing and Bug Identifying
(Status: Completed)
•Backend Implementation and UI/UX Design
(Status: Completed)
•Front to Back Integration
(Status: Completed)
•Testing, Optimizing and Deployment
(Status: Completed)

Things to improve/ work on in the current version of the prototype.

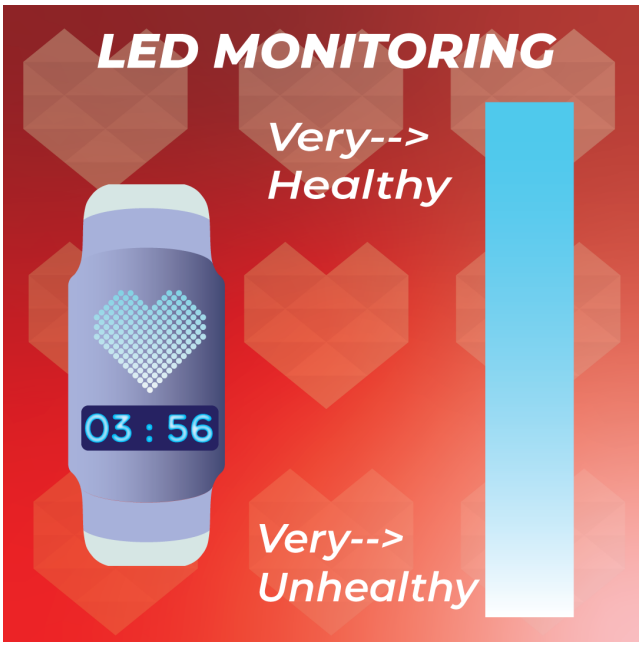
- Improving the accuracy of the model by working with a larger dataset(if available in the future).

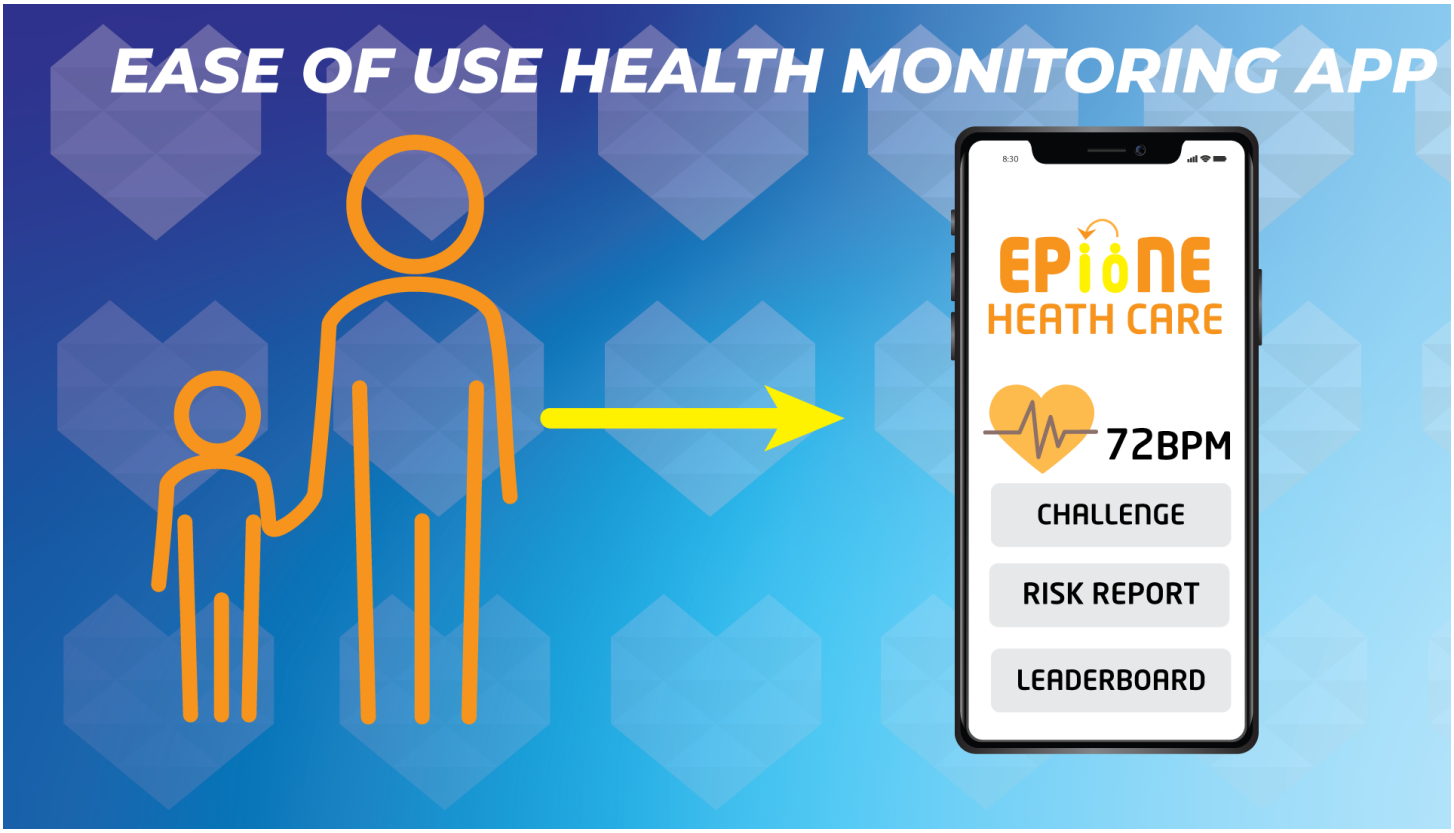
- Having a color-coded display based on the risk index .
    - The proposed scheme :-
        - Risk Index <= 15 [**Green**]R
        - Risk Index > 15 and <35 [**BLUE**]
        - Risk Index >=35 and <45 [**YELLOW**]
        - Risk Index >=45 [**ORANGE**]

The current version only supports **ORANGE** background as shown below, this could be scaled up to the above proposed scheme for a better User Experience and View. Implementation of the proposed scheme proved troublesome in the allotted time frame with the current semester load and deployment bugs encountered in the process .

We hope to implement the above mentioned points in upcoming versions depending on the timeframe and schedule.

Concept Images:-





MVP Images:-


DYNAMIC FRONTEND
Desktop
Mobile
EPIONE HEALTHCARE
Home
About
Team
Gallery
Check your Risk!
WELCOME TO EPIONE HEALTHCARE
We are team of talented designers making a virtual medical assistant that can generate reports and help maintain health
GET STARTED
CHECK YOUR RISK!
Why Choose EPIONE?
We strive to provide the most accurate health report of our clients and provide information such as Risk of Diabetes
Machine Learning System
To provide the best algorithm to calculate the most accurate risk for our clients
Easy Visualization
We provide easy to understand Visualization so that it is easier to interpret our report


EASY TO FOLLOW PROCEDURE
(using Millers Law)
EPIONE MEDICAL ASSISTANT
Fill in Details so we can calculate Health Risk Index
Your Glucose Level (mg/DL)
Your Blood Pressure (mmHg)
Your Skin Thickness (um)
Your Insulin Level (mg/DL)
Your BMI
Your Age
Calculate Health Risk!

POC

Product Description:-

EPIONE is a product first thought of to tackle the growing number of diabetic cases in young adults which was reasoned due to the poor lifestyle in their young age. The product broadly consists of 2 aspects, Hardware and Software . The Hardware aspect includes the EPIONE BAND which collects the user's data such as heart-rate,calories burned etc.

This collected data is stored in a database/cloud which is in turn used to train/teach the model so as to get personalized reports. There are several features in the EPIONE HEALTHCARE application such as a risk index generator which acts as a virtual medical assistant and predicts the risk of getting diabetes in the future based on the current set of data points- this can be used by all ages though the product is focused on teens and young adults, a scoreboard to motivate healthy competition among peers, a report generator which analysis the data collected from the band and generates a personalized report which has information such as:- on which part of the day are we the most inactive or on which days in a week we have the most sedentary schedule etc.

The overall goal of the product is to promote and help maintain a healthy lifestyle in individuals, as the popular saying goes " A Healthy Body , Leads to a Healthy Mind" . We have conducted few ethnographic studies to understand the social aspects of the product .

The MVP involves the EPIONE HEALTHCARE ONLY. The core functionality of the product is reflected in the MVP through the risk index generator (as the report generator is not feasible without a substantial amount of data , which cannot be fabricated as the EPIONE BAND is hard to prototype at this stage with our current knowledge, skills and resources) .

We have explored and researched as much as possible and have documented our findings and reasons for the choices made during the MVP prototyping(can be presented upon request from the panel members/faculty/IIC coordinators)

Client Requirements:-

- Simple and Clean UI
- Easy to Use Procedure
- Accurate Results
- Promotes Healthy Lifestyle

Clients Input and Outputs:-(MVP)

- **Inputs:-**
    - Glucose Level
    - Blood Pressure
    - Skin Thickness
    - Insulin Level
    - BMI
    - Age
- **Outputs:-**
    - Health Risk Level
    - Advisory Action

Problem Statement:-

“The growing number of diabetic cases in teens and young adults which was due to the poor lifestyle in their young age.”

Action Planning:-

Since we are uniquely positioned and have a wide range of marketing potential, there is a huge boost/differentiation in terms of other products. This combined with almost zero delay/limbo time while manufacturing the product , there is a good scope for concurrency to play a role when the firm has gained some experience and maturity.

The product has a certain level of novelty and abstraction which along with the sustainability aspect of it would have the potential to place it apart from the direct competitors.

Product Packaging and Communication Design:-

We package the product in 2 different ways:-

- One packaging with the entire product
- And another set of packaging that consists of Different Straps Thanks to the modular design of the product

Communication:-

We can advertise our Product through Youtube Ads and Amazon Selling Promotions/Offers.

Who are the current and potential competitors?

- Our current competitors are the Fitness bands that target a general audience with no set age range and our potential competitors are Fitness Products targeted at the same target users as well.
- The strategic advantage of being the **first** fitness product to target teens and young adults.
- Another advantage is intrinsic motivation to maintain healthiness cannot be easily replicated by competitors.(As it takes time to adjust to a new product with that form of motivation,users will tend to stay within their comfortable product).
- Trying to make the product selling price reasonable for a teen as well as durable enough to show the parents that the product is worth the money is our best strategy to main success in the industry.
- Once the customer base increases server costs will increase,but we can use sponsors to create contests to maintain server costs as well as through subscriptions)(Sponsors will stay as long as they see results. Sponsors like India supplement and other health and fitness companies.(Fitness companies will see potential customers in physically active teens).

---

*The Product is explained in brief in the attached concept pitch video [**B1-01_Pitch.mp4**]*
