

A  
PROJECT PAHSE-1  
R E P O R T  
On  
**“Real time Prediction of Sugar level using Sweat  
through BSL Band”**

Submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Technology in Information Technology**

By

- 1) Atharva Nandkishor Pingale (T2054491246008)
- 2) Aniket Prakash Wakle (T2054491246006)
- 3) Dhiraj Mahendra Patil (T2054491246015)
- 4) Rohan Manoj Tawar (T2054491246047)

Under the guidance of

**Mr. Sachin Kamble**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

SHRI VILE PARLE KELAVANI MANDAL'S

**INSTITUTE OF TECHNOLOGY, DHULE**

Survey No. 499, Plot No. 02, Behind Gurudwara, Mumbai-Agra National Highway,  
Dhule424001, Maharashtra, India.

Office Phone: 02562-297801 / 297601, Fax : 02562-297801, Mail: IOTDhule@svkm.ac.in

**Academic Year 2023 – 24**

SHRI VILE PARLE KELAVANI MANDAL'S

# INSTITUTE OF TECHNOLOGY, DHULE

Survey No. 499, Plot No. 02, Behind Gurudwara, Mumbai-Agra National Highway,  
Dhule424001, Maharashtra, India.

Office Phone: 02562-297801 / 297601, Fax : 02562-297801, Mail: IOTDhule@svkm.ac.in

**Academic Year 2023 – 24**



## **CERTIFICATE**

This is to certify that Mr. Atharva Nandkishor Pingale

Mr. Aniket Prakash Wakle

Mr. Dhiraj Mahendra Patil

Mr. Rohan Manoj Tawar

students of Information Technology, bearing has successfully completed project phase-1 report on  
“Real time Prediction of Sugar level using Sweat through BSL Band” to my satisfaction and submitted the same during the academic year 2023-2024 towards the partial fulfillment of Bachelor of Technology under Dr. Babasaheb Ambedkar Technological University, Lonere, under the guidance of Mr. Sachin Kamble.

Date: / /2023

Place: SVKM's IOT, Dhule

Mr. Sachin Kamble  
**Project Guide**

Ms. Rubi Mandal  
**Project Co-ordinator**

Dr. Bhushan Chaudhari  
**HOD**

Dr. Nilesh Salunkhe  
**Principal**

Name and Sign with date  
Examiner-1

Name and Sign with date  
Examiner-2

## **DECLARATION**

We declare that this written submission represents ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

### **Signatures**

Mr. Atharva Nandkishor Pingale

---

Mr. Aniket Prakash Wakle

---

Mr. Dhiraj Mahendra Patil

---

Mr. Rohan Manoj Tawar

---

## **Acknowledgments**

We have immense pleasure in expressing our interest and deepest sense of gratitude towards our project guide Prof. Sachin Kamble for the assistance, valuable guidance and co-operation in carrying out this project successfully. It is a privilege for us to have been associated with Prof. Sachin Kamble, our Project Guide, during our Project Phase 1 work. We have greatly benefited from his valuable suggestion and ideas. It is with great pleasure that we express our deep sense of gratitude to him for his valuable guidance, constant encouragement and patience throughout this work. We express our gratitude and are thankful to all people who have contributed in their way in making this final year project phase 1 success. Particularly we want to thank Prof. Rubi Mandal, Project Coordinator for our department for making this process seamless for us and arranging everything so perfectly. I take this opportunity to thank all the classmates for their company during the course work and for the useful discussion, I had with them. We take this opportunity to express our heartfelt gratitude towards the Department of Information Technology of Shri Vile Parle Kelvani Mandal's Institute of Technology, Dhule and Dr. Nilesh Salunkhe, Principal of Shri Vile Parle Kelvani Mandal's Institute of Technology, Dhule, that gave us an opportunity for the presentation of our project phase 1 in the esteemed organization and for providing the required facilities in completing this project. We are greatly thankful to our parents, friends and other faculty members for their motivation, guidance and help whenever needed.

## **Abstract**

Human health stands as an enduringly significant subject. Ongoing enhancements in health monitoring hardware and systems have led to the development of a health services terminal facilitating interaction between acquisition modules and smartphones. This innovation is founded on an analysis of Bluetooth technology and the Android operating system. The causes of various types of diabetes differ significantly. Type 1 diabetes can be prompted by bacterial infections, toxins in food, unidentified elements triggering immune responses, while type 2 diabetes may result from negative lifestyle habits, advancing age, family history, obesity, pancreatitis, polycystic ovary syndrome (PCOS), Cushing's syndrome, glucagonoma, steroid-induced diabetes, among others. Monitoring blood glucose levels enables the determination of glucose concentrations. The system can provide user feedback through a smartphone application by receiving data via Bluetooth from the device. The measured physiological parameters align with the accuracy standards set by clinical norms. For over three decades, diabetic patients have relied on glucose monitoring technology to track their blood glucose levels. This paper reviews primary methods of blood glucose detection and efficient insulin regulation. The conventional and widely used method involves invasive finger-pricking for blood drawing. However, recent advancements have led to the development of non-invasive techniques for monitoring blood glucose, marking a rapidly evolving field. In this context, a mobile physical health monitoring system is proposed, centered around the Android smartphone framework. Progress in modern technology has enabled the creation of compact wireless mobile health monitoring systems, capable of continuous monitoring while remaining energy efficient. This evolution signifies a paradigm shift in health monitoring, wherein wearable body sensors are gaining popularity. A medical survey reveals that one to two individuals in every household manage sugar diabetes, necessitating frequent painful blood sugar level checks. To address this challenge, the concept of the BSL Band was conceived.

**Keywords** - Diabetic Patient, Cloud, Glucose measurement, health sensors, Blood Glucose level, IOT Technology.

## Table of Contents

Certificate	II
Declaration	III
Acknowledgement	IV
Abstract	V
List of Figures	VIII
List of Tables	IX

S.N.	Content	Page No
1	Introduction	1
	1.1 Project Title	1
	1.2 Domain of Project	1
	1.3 Problem Statement	1
	1.3 Motivation behind project topic	1
	1.4 Objectives and Expected Outcomes	1
	1.5 Overview of Project Domain	1
	1.6 Project Purpose and Applicability	2
	1.4 Scope of Project	2
2	Literature Survey	3
	2.1 Related Work Done	3
	2.2 Limitation of Existing System	3
3	Project Requirement Specification	4
	3.1 Hardware and Software Requirement	4
	3.2 Functional Requirement	4
	3.3 Non Functional Requirement	5
	3.4 Data set Description / Database Schema	5

4	Proposed System	9
4.1	System Design and Architecture	9
5	High Level Design of the Project	10
5.1	Use-case Diagram	10
5.2	Data Flow Diagram	11
6	System Implementation	13
6.1	Splash Screen	13
6.2	Output Screen	21
7	Feasibility Study	25
7.1	Introduction to Feasibility Study	25
7.2	Economic Feasibility	25
7.3	Time Feasibility	25
8	Conclusion	26
9	Reference	

## LIST OF FIGURES

<b>Figure No.</b>	<b>Title</b>	<b>Page No.</b>
3.1	Pregnancies and Glucose Data	6
3.2	Blood Pressure and SkinThickness Data	6
3.3	Insuline and BMI Data	7
3.4	DiabetesPedigreeFunction and Age Data	7
3.5	Outcome Data	8
4.1	System Architecture	9
5.1	Use-case Diagram	10
5.2	Data Flow Diagram Level 0	11
5.3	Data Flow Diagram Level 1	11
5.4	Data Flow Diagram Level 2	11
6.1	Urls Implementation	13
6.2	Views 1 Implementation	13
6.3	Views 2 Implementation	14
6.4	Views 3 Implementation	14
6.5	Index 1 Implementation	15
6.6	Index 2 Implementation	15
6.7	Index 3 Implementation	16
6.8	Login 1 Implementation	16
6.9	Login 2 Implementation	17
6.10	Login 3 Implementation	17
6.11	Register 1 Implementation	18
6.12	Register 2 Implementation	18
6.13	Register 3 Implementation	19
6.14	Register 4 Implementation	19
6.15	Dashboard 1 Implementation	20
6.16	Dashboard 2 Implementation	20
6.17	Home Page	21
6.18	SignUp Page	21
6.19	SignIn Page	22
6.20	Dashboard Page	22
6.21	Dashboard 2 Page	23
6.22	About Us Page	23
6.23	Contact Us Page	24

## **LIST OF TABLES**

<b>Table No.</b>	<b>Title</b>	<b>Page No.</b>
3.1	Hardware and Software Requirements	4

# **Chapter 1: Introduction**

## **1.1 Project Title**

- Real time Prediction of Sugar level using Sweat through BSL Band

## **1.2 Domain of Project**

- Machine Learning

## **1.3 Problem Statement**

- In Public health and surveillance there is need for better surveillance methods or tools in public health care to measure and track blood sugar level of patient as they can be unconscious if the blood sugar level is down or up significantly. Sweat contains glucose that can accurately reflect blood glucose. However, skin surface glucose can confound these measurements.

## **1.4 Motivation behind project topic**

According to medical survey there one and two patient in every home who is dealing with sugar diabetes and every time they need to check blood for sugar level which is painful to over come this we came up with the idea of BSL Band

## **1.5 Objectives and Expected Outcomes**

- To overcome daily painful blood sugar level measurement.
- To keep track of the patient sugar level and his daily diet.
- Provide accurate information as compared to traditional methods of blood sugar level calculation.

## **1.6 Overview of Project Domain**

- **Machine Learning:**

The AI segment was developed using the Python programming language. The installation of Python was successfully completed, followed by the setup of the core program. Prediction tasks were carried out using both virtual and authentic datasets. In the proposed framework, an authentic dataset was specifically employed for analysis.

- **Virtual database:**

The dataset used in this study was obtained from the Pima Indians Diabetes Database. It includes various parameters such as age, number of pregnancies, glucose levels, blood pressure, skin thickness, insulin levels, BMI, family history of diabetes, and the outcome variable. The dataset comprises 768 entries with 9 different attributes. To facilitate analysis, the dataset was partitioned into distinct sets for training and testing purposes.

- **Real Database:**

The authentic dataset was collected through a comprehensive survey that gathered details from numerous individuals. This dataset includes various parameters such as age, glucose levels, blood pressure, sodium levels, weight, height, BMI, family history, and the outcome variable. Comprising 50 rows and 10 columns, the dataset was meticulously divided into separate sets for training and testing purposes, ensuring a comprehensive analysis of the entire dataset.

## 1.7 Project Purpose and Applicability

In the realm of public health and surveillance, there exists a necessity for improved surveillance techniques or instruments within healthcare systems to effectively monitor and trace changes in patients' blood sugar levels, particularly when they might be unconscious due to significant fluctuations in these levels. Sweat has the potential to contain glucose, offering a potential avenue to accurately gauge blood glucose levels. Nevertheless, the presence of glucose on the skin surface can introduce potential challenges in these measurements.

## 1.8 Scope of Project

- The design enhancement requires a suitable and efficient power source to be reconsidered.
- Additionally, future improvements in device accuracy could involve adopting specific precautions and preparations during ECG and body temperature measurements for optimized vital sign readings

## Chapter 2: Literature Survey

### 3.1 Related Work Done

Wearable sensors capture body data, like blood sugar levels, transmitting it wirelessly to smartphones. This information is then stored securely in an Electronic Health Record (EHR) within a cloud server. A trusted data manager oversees connections from various apps, ensuring they comply with set standards. Health development tools are recommended for developers to certify the reliability of Mobile Medical Applications (MMAs). Security is verified using a model involving spatial-temporal hybrid automata (STHA). Body sensor networks are pivotal in advancing IoT in healthcare, enabling patient monitoring through small, lightweight sensor nodes. These modern healthcare systems satisfy critical needs such as safety, sustainability, and security. Control inputs from MMAs undergo security checks through a hybrid automata-based system. Sensors equipped with Time Division Multiplexing (TDM) provide sustained long-term support with an eco-friendly design. Data collected by different applications are securely stored and accessible only by authorized apps. Physiology-based end-to-end security (PEES) establishes a direct, secure communication channel between the sensor and the medical cloud. Wireless Sensor Network (WSN) nodes offer diagnostic features and large networks for practical analysis, though they may have high deployment costs. Frameworks like UPHIAC (Ubiquitous Personal Health Information Access) and PRISM (Platform for Remote Sensing using Smartphones) ensure data security using APIs connecting smartphone sensors to cloud storage. Pervasive Health Management System (PHSM) adapts settings based on user mobility. The Hierarchical Power Management (HPM) architecture maintains consistency by integrating ultra-low-power processors. The structure streamlines interactions between external sensors and client Android devices. Smartphones establish connections with external sensors through various channels, presenting challenges in integrating diverse sensors with different communication methods and data formats. The framework aims to simplify these interactions, emphasizing portability, simplicity, and ease of deployment.

### 3.2 Limitation of Existing System

- Wearing it for whole day might cause inconvenience during working hours.
- The device uses a 9V power adapter because some parts like biosensors, the liquid crystal display LCD, and Bluetooth consume a lot of power.
- Most trackers have short battery life, so you can't track your workouts while charging.
- Trackers do not provide proper accuracy.

## Chapter 3: Project Requirement Specification

### 3.1 Hardware and Software Requirement

- Hardware Requirement:

Table 3.1: Hardware Requirements:

Sr. No.	Parameter	Minimum Requirement	Justification
1	Node MCU	ESP 12e	For Working Of Sensor
2	Raindrop Sensor	-	For Glucose Measurements
3	CPU Speed	1 GHz	For Processing
4	RAM	3 GB	-

- Software Requirement:
  1. Operating System: Microsoft Windows
  2. Programming Language: Python, Django, HTML, JavaScript.

### 3.2 Functional Requirement

- Sweat sensor which gives input to glucomonitor and then show it on user login page.
- Data undergoes cross-validation to enhance the accuracy and efficiency of the machine learning algorithm during training.
- An Arduino Uno serves as the system's controller, receiving power supply.
- Skin conductance, inversely linked to salt content, is converted to voltage readings. GSR sensor inversely correlates with glucose levels.
- Support Vector Regression (SVR) tackles non-linearity in data through kernel functions like Linear, Polynomial, Sigmoid, and Radial Basis Function (RBF).
- SVR's strength lies in recognizing and predicting non-linearities in data without altering descriptive values, thus enhancing model accuracy.

- SVR relies on kernel functions rather than solely on the spread of dependent and independent variables. This allows for the construction of a non-linear model while preserving descriptive values, thereby enhancing the resulting model's quality.

### **3.3 Non Functional Requirement**

- Portability: This proposed System is easily portable because it is wearable band.
- Security Measures: The proposed system ensures scalable security by authenticating users each time they log in.  
In the event of a cyber attack, the system initiates an automatic shutdown.  
Verification Email: A verification email is dispatched to users upon their initial registration within the software system.
- Flexibility: Proposed System is flexible to work with and flexible to understand
- Patient Recognition: The system requires patients to identify themselves via their phone.
- User Authentication: All system users must possess a unique Logon ID and password.
- Database Changes: Any alterations such as insertion, deletion, or updates within the database are swiftly synchronized and exclusively executed by the ward administrator.

### **3.4 Data set Description / Database Schema**

#### **Context:**

The dataset originates from the National Institute of Diabetes and Digestive and Kidney Diseases. Its primary aim is to predict the likelihood of diabetes in patients using specific diagnostic measurements. To create this dataset, specific criteria were applied to select instances from a larger database. Notably, all the included patients are females aged at least 21 years old and belong to the Pima Indian heritage.



Figure 3.1: Pregnancies and Glucose Data

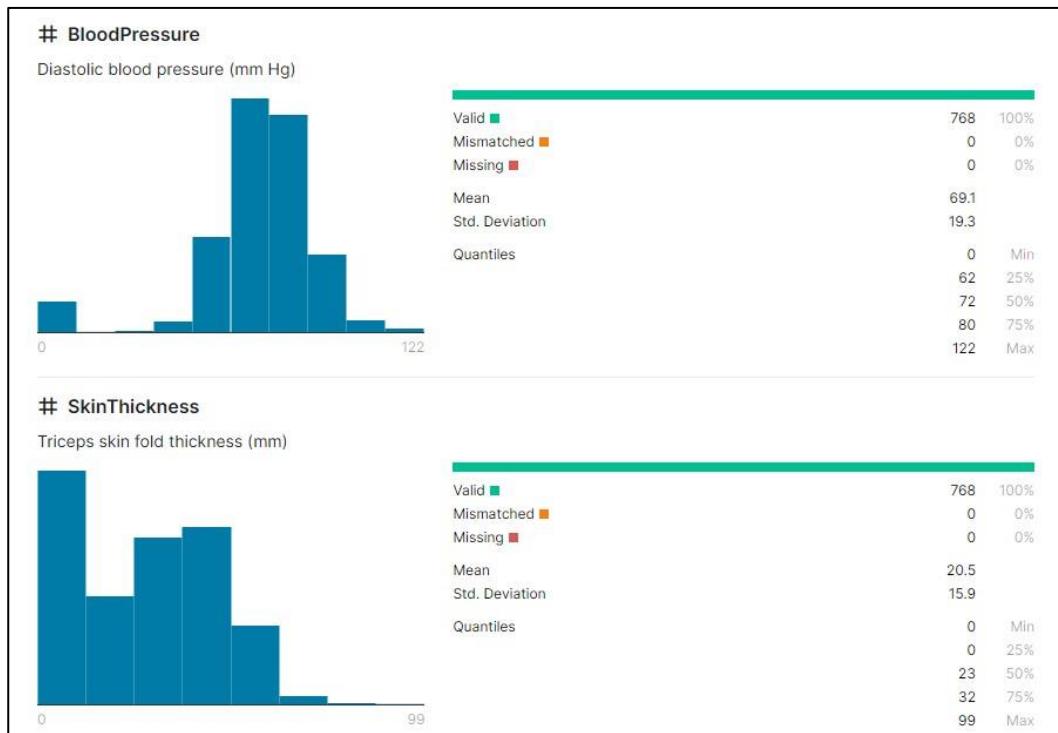


Figure 3.2: Blood Pressure and SkinThikness Data

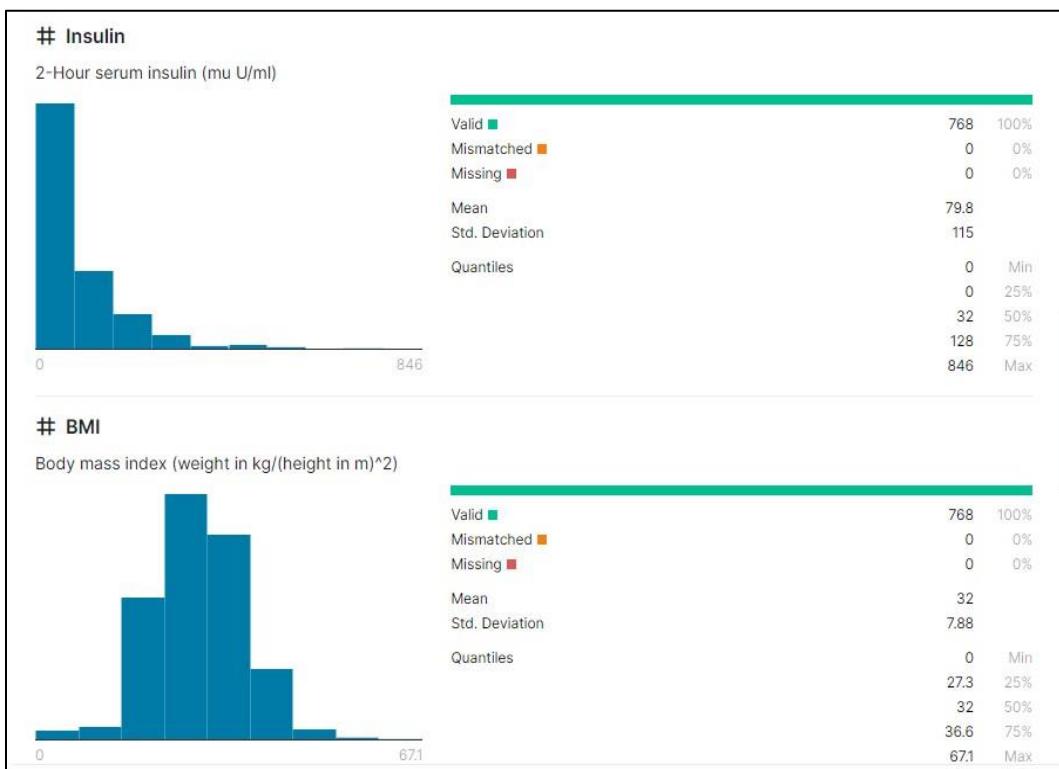


Figure 3.3: Insuline and BMI Data

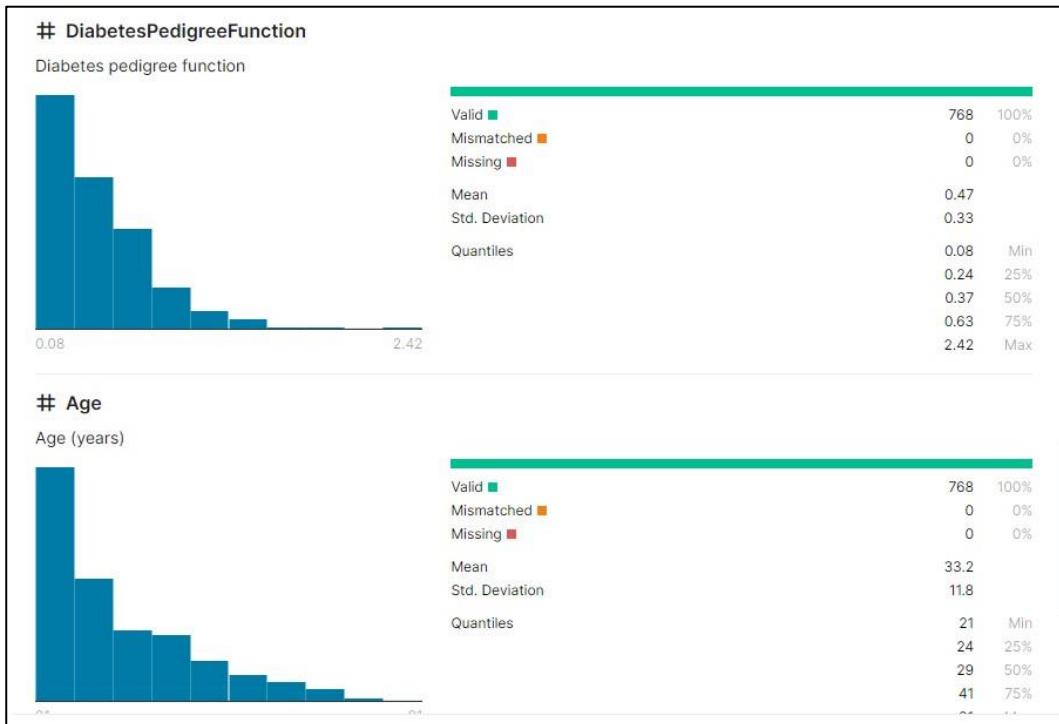


Figure 3.4: DiabetesPedigreeFunction and Age Data

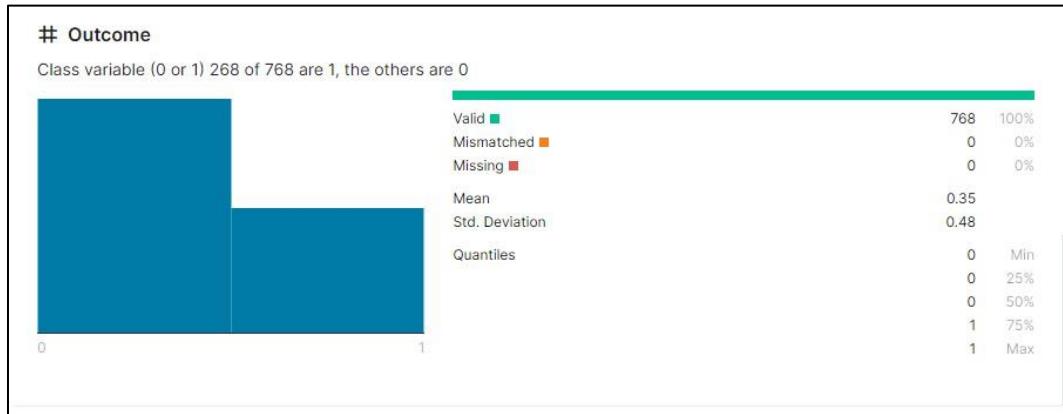
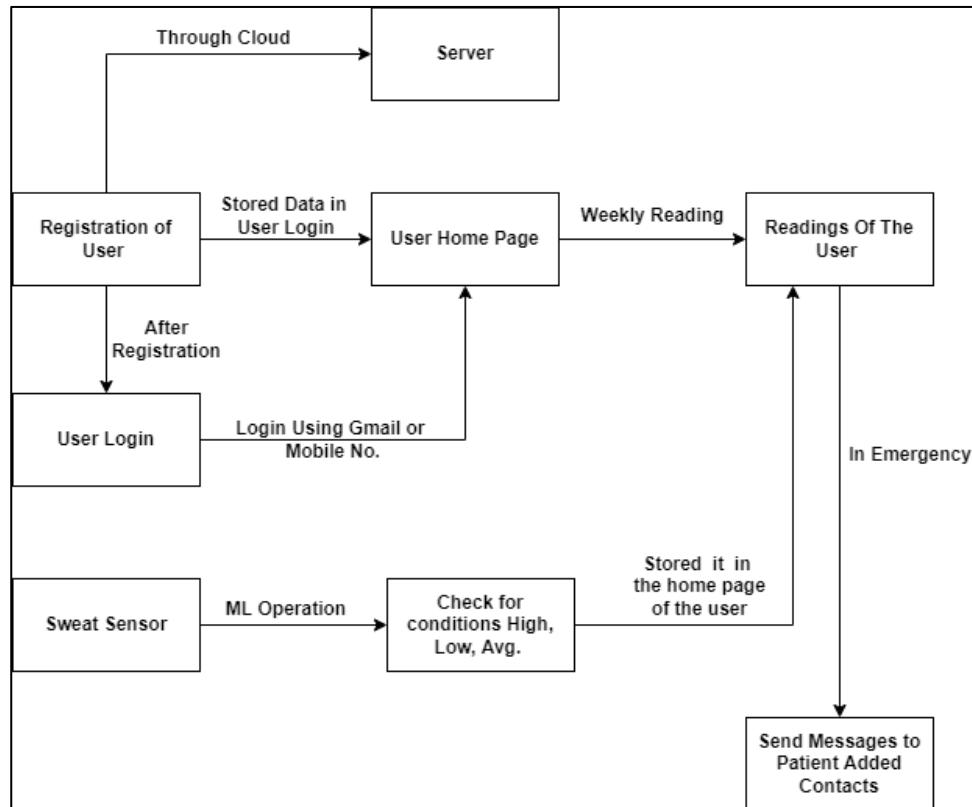


Figure 3.5: Outcome Data

## Chapter 4: Proposed System

### 4.1 System Design and Architecture:



4.1 System Architecture

**Registration:** Firstly user will register in our system, for registration user add details like name, username, contact, height, weight, age, email, address and the preferred contact number.

**Login:** After registration user have to login into the system using Gmail or Mobile number and user will redirect to user home page.

**Sweat Sensor:** Sweat sensor which gives input to glucomonitor and then show it on user login page.

**User Readings:** On the user's Home page, user reading are available. These readings are in the form of graph of Blood Sugar Level like Weekly Blood Sugar Level.

**Conditions:** User readings are stored in the home page of the user these readings are used to check for conditions like High, Low and Average.

# Chapter 5: High Level Design of the Project

## 5.1 Use-case Diagram:

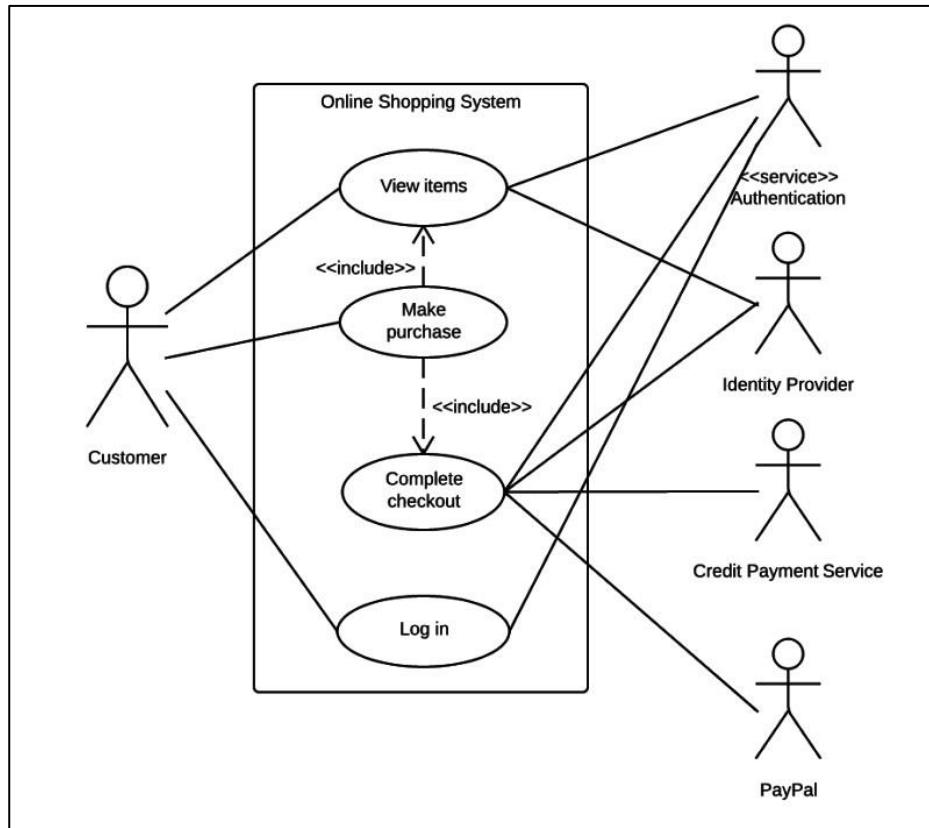


Figure 5.1: Use-case Diagram

The UML Diagram follows a sequence where users first register before gaining login access. Within this system, a sweat sensor doubles as an indicator for blood-related metrics. Specifically, when this sensor detects a count below a predefined threshold, it activates a notification mechanism. This triggers the system to send alert messages directly to the contact number registered by the user. These messages serve as timely warnings, notifying the user about the detected low count indicated by the sweat sensor. This entire process is designed to ensure that users receive immediate alerts regarding critical blood levels, emphasizing the importance of prior registration for this alert system. Upon successful registration, users gain access to the system's functionalities via a login mechanism. The sweat sensor continuously monitors blood levels in real time, providing proactive alerts to address low counts promptly and effectively.

## 5.2 Data Flow Diagram:

### 5.2.1 DFD Level 0:



Figure 5.2: Data Flow Diagram Level 0

It Shows the Simple architecture of our idea In DFL 0 We have sweat sensor which gives input to glucomonitor and then show it on user login page

### 5.2.2 DFD Level 1:

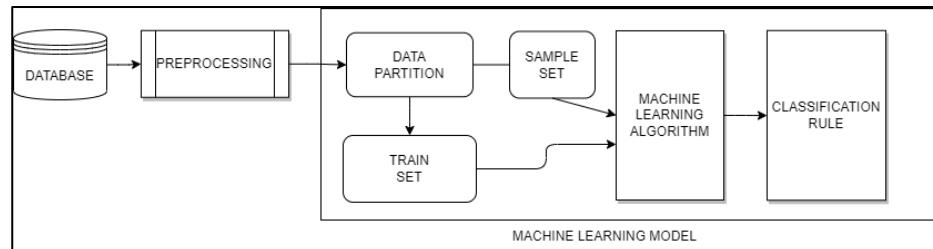


Figure 5.3: Data Flow Diagram Level 1

In the DFD Level 1 Diagram, the database serves as the initial input. It undergoes a preprocessing phase where the string data is converted into integers. The database is subsequently split into two segments: a testing part and a training part. The training part is utilized to enable the machine to identify and understand patterns within the data. To enhance accuracy and efficiency, cross-validation of the data is performed, optimizing the algorithm used in training the machine.

### 5.2.3 DFD Level 2:

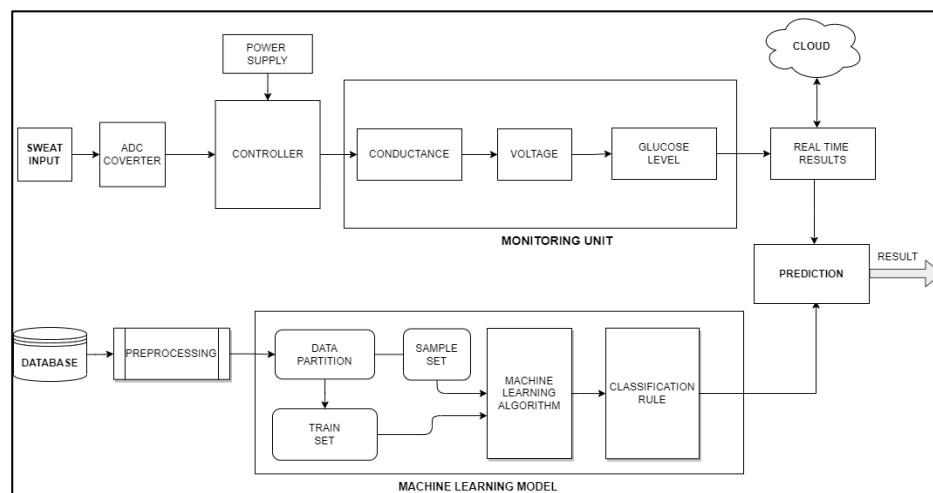
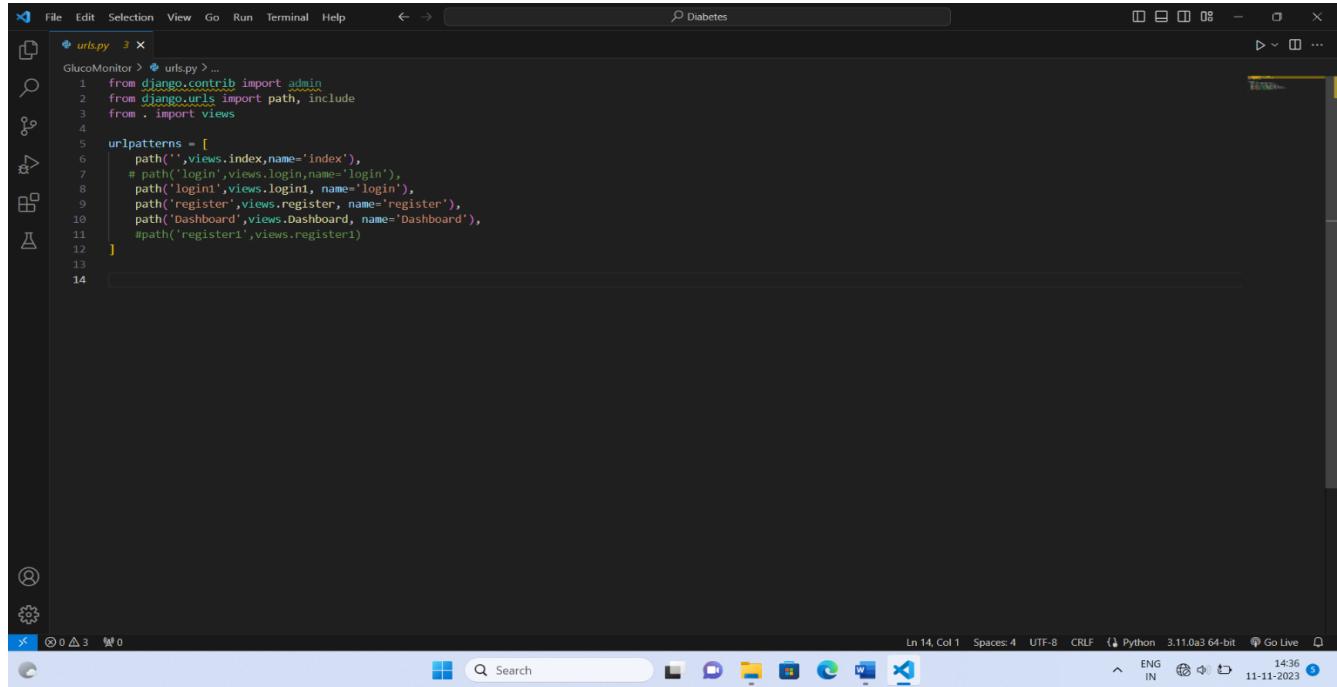


Figure 5.4: Data Flow Diagram Level 2

The DFD Level 2 Diagram comprises two main sections: the machine learning component and the real-time monitoring system for glucose, stress, and hydration. In the machine learning segment, a database is preprocessed from string to integer format, split into testing and training parts. This division ensures pattern recognition by the machine, employing cross-validation for accuracy and efficiency. Different machine learning algorithms are tested, and the most accurate one is selected for predictions. The real-time monitoring section uses a GSR sensor to read sweat data. An Arduino Uno acts as the controller, powered to manage the system. The GSR sensor's analog output is converted to digital via an analog-to-digital converter (ADC). This data helps gauge skin conductance, inversely linked to salt content and glucose levels, as well as stress and hydration levels. These readings are converted, stored in the cloud using IoT (Thing-Speak), facilitating remote access.

# Chapter 6: System Implementation

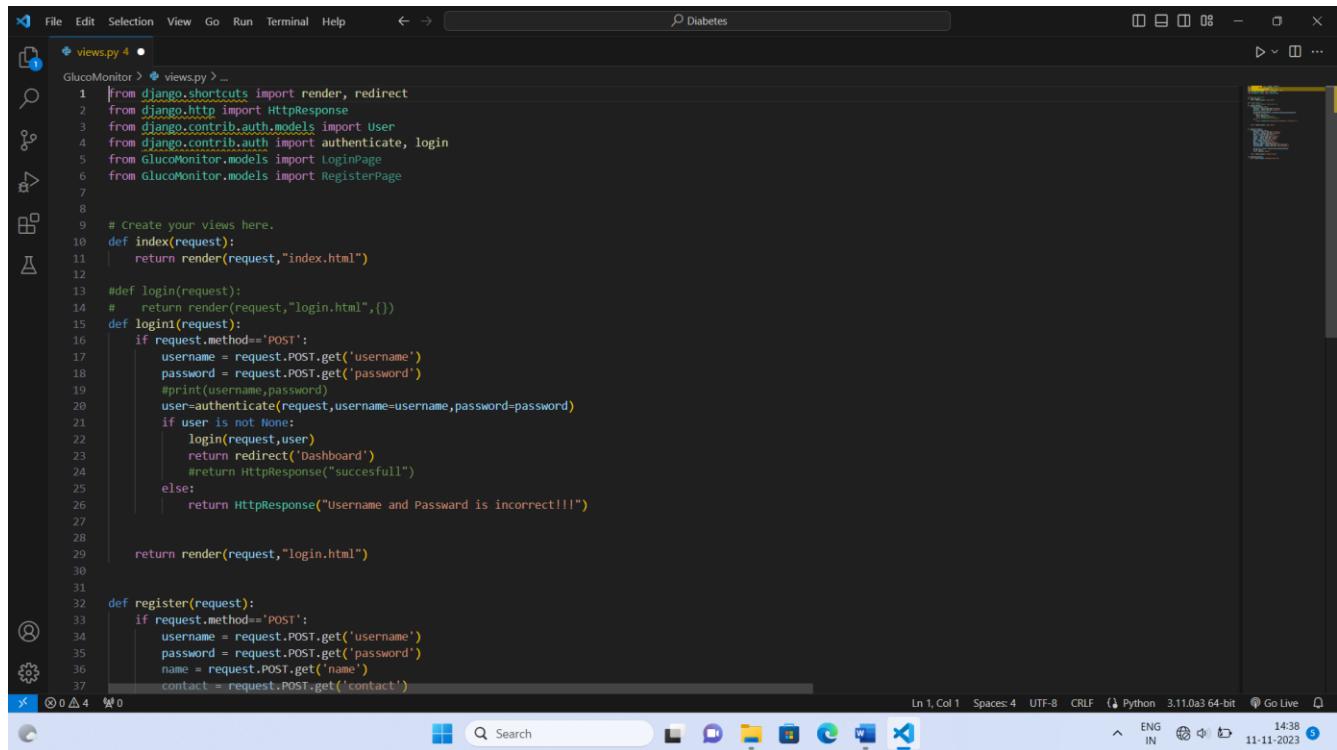
## 6.1 Splash Screen:



The screenshot shows a code editor window with the file 'urls.py' open. The code defines URL patterns for a Django application, including paths for index, login, register, and dashboard pages.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏪ Diabetes
urls.py 3 ×
GlucoMonitor > urls.py > ...
1 from django.contrib import admin
2 from django.urls import path, include
3 from . import views
4
5 urlpatterns = [
6     path('', views.index, name='index'),
7     # path('login', views.login, name='login'),
8     path('login', views.login1, name='login'),
9     path('register', views.register, name='register'),
10    path('Dashboard', views.dashboard, name='dashboard'),
11    #path('register1',views.register1)
12 ]
13
14
```

Figure 6.1: Urls Implementation



The screenshot shows a code editor window with the file 'views.py' open. The code implements various views for user authentication and registration, including index, login, and register functions.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏪ Diabetes
views.py 4 ●
GlucoMonitor > views.py > ...
1 from django.shortcuts import render, redirect
2 from django.http import HttpResponse
3 from django.contrib.auth.models import User
4 from django.contrib.auth import authenticate, login
5 from GlucoMonitor.models import LoginPage
6 from GlucoMonitor.models import RegisterPage
7
8
9 # Create your views here.
10 def index(request):
11     return render(request, "index.html")
12
13 #def login(request):
14 #    return render(request, "login.html", {})
15 def login(request):
16     if request.method == 'POST':
17         username = request.POST.get('username')
18         password = request.POST.get('password')
19         #print(username, password)
20         user = authenticate(request, username=username, password=password)
21         if user is not None:
22             login(request, user)
23             return redirect("Dashboard")
24             #return HttpResponse("successful")
25         else:
26             return HttpResponse("Username and Password is incorrect!!!")
27
28
29     return render(request, "Login.html")
30
31
32 def register(request):
33     if request.method == 'POST':
34         username = request.POST.get('username')
35         password = request.POST.get('password')
36         name = request.POST.get('name')
37         contact = request.POST.get('contact')
```

Figure 6.2: Views 1 Implementation

The screenshot shows a Python development environment with the following code:

```
File Edit Selection View Go Run Terminal Help ← → ⌘ Diabetes
```

```
GlucoMonitor > views.py 4 ●
```

```
18 password = request.POST.get('password')
19 #print(username,password)
20 user=authenticate(request,username=username,password=password)
21 if user is not None:
22     login(request,user)
23     return redirect('dashbboard')
24     #return HttpResponse("succesfull")
25 else:
26     return HttpResponse("Username and Password is incorrect!!!")
27
28
29 return render(request,"login.html")
30
31
32 def register(request):
33     if request.method=="POST":
34         username = request.POST.get('username')
35         password = request.POST.get('password')
36         name = request.POST.get('name')
37         contact = request.POST.get('contact')
38         height = request.POST.get('height')
39         weight = request.POST.get('weight')
40         age = request.POST.get('age')
41         email = request.POST.get('email')
42         address = request.POST.get('address')
43         familyPhoneNo1 = request.POST.get('familyPhoneNo1')
44         familyPhoneNo2 = request.POST.get('familyPhoneNo2')
45
46         my_user=User.objects.create_user(username,password)
47         my_user.save()
48         return redirect('login')
49
50     return render(request,"register.html")
51
52 def Dashboard(request):
53     return render(request,"Dashboard.html",{})

Ln 53, Col 46 Spaces: 4 UTRF-8 CRLF ↴ Python 3.11.0a3 64-bit ⚡ Go Live ⌂ ENG IN 14:38 11-11-2023
```

Figure 6.3: Views 2 Implementation

```
File Edit Selection View Go Run Terminal Help ← → ⌘ Diabetes
```

views.py 4 ● index.html x

```
templates > index.html > ...
```

```
1  [% load static %]
2  <!DOCTYPE html>
3  <html lang="en">
4
5  <head>
6      <meta charset="utf-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1">
8      <title>Glucose Monitor</title>
9      <meta name="description" content="Free Bootstrap Theme by BootstrapMade.com">
10     <meta name="keywords" content="free website templates, free bootstrap themes, free template, free bootstrap, free website template">
11
12     <link rel="stylesheet" type="text/css" href="{% static 'https://fonts.googleapis.com/css?family=Open+Sans|Raleway|Candal' %}">
13     <link rel="stylesheet" type="text/css" href="{% static 'css/font-awesome.min.css' %}">
14     <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css' %}">
15     <link rel="stylesheet" type="text/css" href="{% static 'css/style.css' %}">
16
17     <!-- =====
18     Theme Name: Medilab
19     Theme URL: https://bootstrapmade.com/medilab-free-medical-bootstrap-theme/
20     Author: BootstrapMade.com
21     Author URL: https://bootstrapmade.com
22     ===== -->
23 </head>
24
25 <body id="myPage" data-spy="scroll" data-target=".navbar" data-offset="60">
26     <!--banner-->
27     <section id="banner" class="banner">
28         <div class="bg-color">
29             <nav class="navbar navbar-default navbar-fixed-top">
30                 <div class="container">
31                     <div class="col-md-12">
32                         <div class="navbar-header">
33                             <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">
34                                 <span class="icon-bar"></span>
35                                 <span class="icon-bar"></span>
36                                 <span class="icon-bar"></span>
37                             </button>
```

Figure 6.4: Views 3 Implementation

```

38     <a class="navbar-brand" href="#"></a>
39   </div>
40   <div class="collapse navbar-collapse navbar-right" id="myNavbar">
41     <ul class="nav navbar-nav">
42       <li class="active"><a href="#banner">Home</a></li>
43       <li class=""><a href="#">Login</a></li>
44       <li class=""><a href="#">Register</a></li>
45       <li class=""><a href="#">About</a></li>
46       <li class=""><a href="#">Contact</a></li>
47     </ul>
48   </div>
49 </div>
50 </nav>
51 <div class="container">
52   <div class="row">
53     <div class="banner-info">
54       <div class="banner-logo text-center">
55         
56       </div>
57       <div class="banner-text text-center">
58         <h1 class="white">Check Your Glucose with Us!!</h1>
59         <p>Diabetes is a great example whereby, giving the patient the tools, you can manage yourself very well.</p>
60         <a href="#">Get Started</a> <!--Register Form-->
61       </div>
62       <div class="overlay-detail text-center">
63         <a href="#"><i class="fa fa-angle-down"></i></a>
64       </div>
65     </div>
66   </div>
67 </div>
68 </div>
69 </div>
70 </section>
71 <!-- banner-->
72
73 <!-- service-->
74 <section id="service" class="section-padding">

```

Ln 73, Col 1 Spaces: 2 UTF-8 CRLF HTML ⚡ Go Live

ENG IN 14:39 11-11-2023

Figure 6.5: Index 1 Implementation

```

109 <section id="about" class="section-padding">
110   <div class="container">
111     <div class="row">
112       <div class="col-md-3 col-sm-4 col-xs-12">
113         <div class="section-title">
114           <h2 class="head-title lg-line">The Medilap <br>Ultimate Dream</h2>
115           <hr class="bottom-line">
116           <p class="sec-para">GlucoMonitor is known for providing advanced emergency services with rapid response.</p>
117           <a href="#" style="color: #00b8b6; padding-top:10px;">Know more..</a>
118         </div>
119       </div>
120       <div class="col-md-9 col-sm-8 col-xs-12">
121         <div style="visibility: visible;" class="col-sm-9 more-features-box">
122           <div class="more-features-box-text">
123             <div class="more-features-box-text-icon" style="text-align: center;"><i class="fa fa-angle-right" aria-hidden="true"></i></div>
124             <div class="more-features-box-text-description">
125               <h3>It's something important you want to know.</h3>
126               <p>GlucoMonitor is known for providing advanced emergency services with rapid response. We have well-equipped emergency and trauma center with facilities ...</p>
127             </div>
128           </div>
129           <div class="more-features-box-text">
130             <div class="more-features-box-text-icon" style="text-align: center;"><i class="fa fa-angle-right" aria-hidden="true"></i></div>
131             <div class="more-features-box-text-description">
132               <h3>It's something important you want to know.</h3>
133               <p>GlucoMonitor is known for providing advanced emergency services with rapid response. We have well-equipped emergency and trauma center with facilities ...</p>
134             </div>
135           </div>
136         </div>
137       </div>
138     </div>
139   </div>
140 </section>
141 <!-- about-->
142 <!-- cta 2-->
143 <section id="cta-2" class="section-padding">
144   <div class="container">

```

Ln 144, Col 1 Spaces: 2 UTF-8 CRLF HTML ⚡ Go Live

ENG IN 14:39 11-11-2023

Figure 6.6: Index 2 Implementation

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Search term: Diabetes.
- Code Area:** The code is a template for an index.html file. It includes a header section with contact information (name, address, phone), a message box for sending messages, and a contact form with fields for name, email, subject, and message, along with a submit button.
- Code Content:**

```
templates> index.html > html > body#myPage > section#contact.section-padding > div.container > div.row > div.col-md-8.col-sm-8.marb20 > div.contact-info > form.contactForm
170   <h3>Contact Info</h3>
171   <div class="space"></div>
172   <p><i class="fa fa-map-marker fa-fw pull-left fa-2x"></i>321 Awesome Street<br>Dhule, Maharashtra</p>
173   <div class="space"></div>
174   <p><i class="fa fa-envelope-o fa-fw pull-left fa-2x"></i>info@glucomonitor.com</p>
175   <div class="space"></div>
176   <p><i class="fa fa-phone fa-fw pull-left fa-2x"></i>+91 800 123 1234</p>
177 </div>
178 <div class="col-md-8 col-sm-8 marb20">
179   <div class="contact-info">
180     <h3>Having Any Query! or Book an appointment</h3>
181     <div class="space"></div>
182     <div id="sendmessage">Your message has been sent. Thank you!</div>
183     <div id="errormessage"></div>
184     <form action="" method="post" role="form" class="contactForm">
185       <div class="form-group">
186         <input type="text" name="name" class="form-control br-radius-zero" id="name" placeholder="Your Name" data-rule="minlen:4" data-msg="Please enter at least 4 characters" data-validation="required">
187         <div class="validation"></div>
188       </div>
189       <div class="form-group">
190         <input type="email" class="form-control br-radius-zero" name="email" id="email" placeholder="Your Email" data-rule="email" data-msg="Please enter a valid email" data-validation="required">
191         <div class="validation"></div>
192       </div>
193       <div class="form-group">
194         <input type="text" class="form-control br-radius-zero" name="subject" id="subject" placeholder="Subject" data-rule="minlen:4" data-msg="Please enter at least 4 characters" data-validation="required">
195         <div class="validation"></div>
196       </div>
197       <div class="form-group">
198         <textarea class="form-control br-radius-zero" name="message" rows="5" data-rule="required" data-msg="Please write something for us" placeholder="Message" data-validation="required"></textarea>
199         <div class="validation"></div>
200       </div>
201
202       <div class="form-action">
203         <button type="submit" class="btn btn-form">Send Message</button>
204       </div>
205     </form>
206   </div>
```
- Status Bar:** Ln 205, Col 8 | Spaces: 2 | UTF-8 | CRLF | HTML | Go Live | ENG | IN | 14:39 | 11-11-2023

Figure 6.7 : Index 3 Implementation

```
File Edit Selection View Go Run Terminal Help ← → ⌘ Diabetes
```

```
login.html
```

```
templates > login.html > ...  
1  [% load static %]  
2  
3  <!DOCTYPE html>  
4  <html lang="en">  
5  <head>  
6      <meta charset="UTF-8">  
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
8      <meta http-equiv="X-UA-Compatible" content="ie=edge">  
9      <link rel="stylesheet" type="text/css" href="{% static 'regcss.css' %}">  
10     <link rel="stylesheet" type="text/css" href="{% static 'studdashcss.css' %}">  
11     <link rel="stylesheet" type="text/css" href="{% static 'css/font-awesome.min.css' %}">  
12     <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css' %}">  
13     <link rel="stylesheet" type="text/css" href="{% static 'css/style.css' %}">  
14  </title></title>  
15  
16  <style>  
17  
18  | | | | * {  
19  | | | | box-sizing: border-box;  
20  | | }  
21  body{  
22  | | background: url();  
23  | | background-size: cover;  
24  | }  
25  
26  hr{  
27  | | border: none;  
28  | | height: 70px;  
29  | | background: #e0e0e0;  
30  | | margin-top: -70px;  
31  | }  
32  .wrapper{  
33  | | box-shadow: -30px 30px 30px -10px rgba( 100, 30, 22 ,0.5);  
34  | | margin-left: 430px;  
35  | | background-color: #f1f1f1;  
36  | | background: #rgb(0, 0, 0); /* Fallback color */  
37  | | background: rgba(0,0,0, 0.5); /* Black background with 0.5 opacity 241, 148, 138 */
```

```
LN 1, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live
```

```
ENG IN ⌘ 11-11-2023 14:40 S
```

Figure 6.8: Login 1 Implementation

The screenshot shows a code editor interface with a dark theme. The file being edited is 'login.html'. The code is a template for a login page, featuring a header with a logo and navigation links, and a main content area with a form for user input. The code includes CSS-like styles for button hover states and font imports from Google Fonts.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏴ Diabetes
templates > login.html > html > body > nav > div > .navbar.navbar-default.navbar-fixed-top > div.container
  .button:hover{
    background: #AEB6BF;
    color: #1D9661;
    font-weight: bold;
  }
</style>
<link href="https://fonts.googleapis.com/css?family=Raleway:300,400,700&display=swap" rel="stylesheet">
</head>
<body>
  <nav style="background: #1b1b1b; height: 70px;">
    
    <div>
      </div>
    </div>
    <div>
      <nav class="navbar navbar-default navbar-fixed-top">
        <div class="container">
          <div class="col-md-12">
            <div class="navbar-header">
              <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
              </button>
            </div>
            <div class="collapse navbar-collapse navbar-right" id="myNavbar">
              <ul class="nav navbar-nav">
                <li class="active" style="margin-top: -5px;"><a href="#">Home</a></li>
                <li class="" style="margin-top: -5px;"><a href="#">register</a></li>
              </ul>
            </div>
          </div>
        </nav>
      </div>
    </div>
  </nav>
</body>
```

Figure 6.9: Login 2 Implementation

The screenshot shows a code editor interface with a dark theme. The file being edited is 'login.html'. The code defines a form for user login, including fields for username and password and a submit button. It uses CSS classes for styling the input fields and labels.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏴ Diabetes
templates > login.html > ...
</div>
<div class="wrapper">
<div>
  <h2 style="text-align: center; font-family: Gabriola; font-weight: bold; font-size: 50px; margin-top: 20px; color: white; line-height: 50px;">SignIn</h2>
</div>
<form action="" style="margin-top: 0px" method="POST">
  {% csrf_token %}
  <div class="input-data" style="margin-top: 30px; margin-left: 50px">
    <input type="text" name="username" required>
    <div class="underline"></div>
    <label>Username</label>
  </div>
  <div class="input-data" style="margin-top: 30px; margin-left: 50px">
    <input type="password" name="password" required>
    <div class="underline"></div>
    <label>Password</label>
  </div>
  <input type="submit" class="button" name="sbm" value="SignIn" style="margin-left: 60px; width: 200px; height: 40px; font-family: Gabriola; font-style: bold" />
</form>
<!-- {{username}} -->
<!-- {{password}} -->
</div>
</body>
</html>
```

Figure 6.10: Login 3 Implementation

```
File Edit Selection View Go Run Terminal Help ← → ⌘ Diabetes
```

register.html 1 x

templates > register.html ...

```
1  [% load static %]
2
3  <!DOCTYPE html>
4  <html lang="en">
5  <head>
6      <meta charset="UTF-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <meta http-equiv="X-UA-Compatible" content="ie=edge">
9      <link rel="stylesheet" type="text/css" href="{% static 'regcss1.css' %}">
10     <link rel="stylesheet" type="text/css" href="{% static 'studdashcss1.css' %}">
11     <link rel="stylesheet" type="text/css" href="{% static 'css/font-awesome.min.css' %}">
12     <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css' %}">
13     <link rel="stylesheet" type="text/css" href="{% static 'css/style.css' %}">
14     <title></title>
15     <style>
16
17     * {
18         box-sizing: border-box;
19     }
20     body{
21         background: url(i39.jpg);
22         background-size: cover;
23     }
24
25     hr{
26         border: none;
27         height: 70px;
28         background: #404040;
29         margin-top: -70px;
30     }
31     .wrapper{
32         box-shadow: -30px 30px 30px -10px rgba( 100, 30, 22 ,0.5);
33         margin-left: 260px;
34         background-color: #fififi;
35         background: #rgb(0, 0, 0); /* Fallback color */
36         background: #rgb(0,0,0 , 0.5); /* Black background with 0.5 opacity 241, 148, 138 */
37         margin-top: 30px;
```

Figure 6.11: Register 1 Implementation

Figure 6.12: Register 2 Implementation

```
File Edit Selection View Go Run Terminal Help ← → ⌘ Diabetes

register.html 1 ×
templates > register.html > html > body > nav > div.wrapper > form > div.input-data
114  <div>
115  <h2 style="text-align: center;font-family: Gabriola;font-weight: bold ;font-size: 50px;margin-top: 0px;color: white;line-height: 50px">SignUp</h2>
116  </div>
117
118  <form action="" style="margin-top: 25px;text-transform: none;" method="POST">
119
120  {% csrf_token %}
121
122      <div class="input-data" style="margin-top: -10px;">
123          <input type="text" name="name" required>
124          <div class="underline"></div>
125          <label style="text-transform: none;">Full Name</label>
126      </div>
127      <div class="input-data" style="margin-top: 30px;">
128          <input type="text" name="username" required>
129          <div class="underline"></div>
130          <label>username</label>
131
132      </div>
133      <div class="input-data" style="margin-top: 30px;">
134          <input type="text" name="password" required>
135          <div class="underline"></div>
136          <label>Password</label>
137
138      </div>
139      <div class="input-data" style="margin-top: 30px;">
140
141          <input type="text" name="contact" required>
142          <div class="underline"></div>
143          <label>Contact </label>
144
145      </div>
146      <div class="input-data" style="margin-top: 30px;">
147
148          <input type="text" name="height" required>
149          <div class="underline"></div>
150          <label>Height </label>
151
152      </div>
```

Figure 6.13: Register 3 Implementation

The screenshot shows a web browser window with a registration form titled "registerhtml 1". The form contains fields for weight, age, email, address, family phone numbers, and a submit button. The code is displayed in a code editor on the left, and the browser interface is visible on the right.

```
File Edit Selection View Go Run Terminal Help ← → Diabetes registerhtml 1 x templates > register.html > html > body > nav > div.wrapper > form > div 153 | <div class="input-data" style="margin-top: 25px;"> 154 |   <input type="text" name="weight" required> 155 |   <div class="underline"></div> 156 |   <label>Weight</label> 157 | </div> 158 | <div class="input-data" style="margin-top: 30px;"> 159 |   <input type="text" name="age" required> 160 |   <div class="underline"></div> 161 |   <label>Age </label> 162 | </div> 163 | 164 | 165 | <div class="input-data" style="margin-top: 30px;"> 166 |   <input type="text" name="email" required> 167 |   <div class="underline"></div> 168 |   <label>Email</label> 169 | </div> 170 | 171 | <div class="input-data" style="margin-top: 20px;"> 172 |   <input type="textarea" name="address" required> 173 |   <div class="underline"></div> 174 |   <label>Address</label> 175 | </div> 176 | <div class="input-data" style="margin-top: 20px;"> 177 |   <input type="text" name="familyPhoneNo1" required> 178 |   <input type="text" name="familyPhoneNo2" required> 179 |   <div class="underline"></div> 180 |   <label>Preferred Contact No .</label> 181 | </div> 182 | 183 | 184 | 185 | 186 | <div> 187 | <input type="submit" class="button" name="sbm" value="signUp" style="margin-left: 160px; width: 200px; height: 49px; font-family: Gabriola; font-style: bold; font-weight: normal; border-radius: 10px; background-color: #007bff; color: white; border: none; padding: 10px; text-decoration: none; cursor: pointer; transition: all 0.3s ease;"/> 188 | </div> 189 </form>
```

Figure 6.14: Register 4 Implementation

The screenshot shows a Microsoft Edge browser window with a dashboard.html file open in a code editor. The dashboard displays a chart, a table, and a pie chart. The code editor shows the following structure:

```
File Edit Selection View Go Run Terminal Help ← → ⌘ Diabetes
```

Dashboard.html

```
templates > Dashboard.html > html > head > style > .card
```

```
15  {%- load static %} 16  <!DOCTYPE html> 17  <html lang="en" dir="ltr"> 18      <head> 19          <meta charset="utf-8"> 20          <script src="https://cdnjs.cloudflare.com/ajax/libs/chart.js/2.9.4/chart.js"></script> 21          <meta name="viewport" content="width=device-width, initial-scale=1"> 22          <link rel="stylesheet" type="text/css" href="{% static 'regcss1.css' %}"> 23          <link rel="stylesheet" type="text/css" href="{% static 'studdashcss1.css' %}"> 24          <link rel="stylesheet" type="text/css" href="{% static 'css/fontawesome.min.css' %}"> 25          <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css' %}"> 26          <link href="{% static 'http://netdna.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css' %}" rel="stylesheet"> 27          <link rel="stylesheet" type="text/css" href="{% static 'css/style.css' %}"> 28      </style> 29      <body> 30          margin-top:20px; 31          color: #1a202c; 32          text-align: left; 33          /* background-color: */ 34      </body> 35  <.main-body { 36      padding: 15px; 37  } 38  .card { 39      box-shadow: 0 1px 3px 0 rgba(0,0,0,.1), 0 1px 2px 0 rgba(0,0,0,.06); 40  } 41  .card { 42      position: relative; 43      display: flex; 44      flex-direction: column; 45      min-width: 0; 46      word-wrap: break-word; 47      background-color: #fff; 48      background-clip: border-box; 49      border: 0 solid rgba(0,0,0,.125); 50  }
```

Figure 6.15: Dashboard 1 Implementation

```
File Edit Selection View Go Run Terminal Help ← → ⌘ Diabetes
```

Dashboard.html

```
templates > Dashboard.html > html
```

```
384 |     <input type="text" class="fullname" placeholder="Skills" required>
385 |     </div>
386 |     <div style="margin-left: 250px; margin-top: -255px">
387 |         <input type="text" class="fullname" placeholder="Mobile" required>
388 |         <input type="text" class="fullname" placeholder="CGPA" required>
389 |         <input type="text" class="fullname" placeholder="Projects" required>
390     </div>
391
392
393     <div style="margin-top: 50px; margin-left: 100px">
394         <button type="submit" class="btn" style="width: 240px">Update</button>
395         <button type="button" class="btn cancel" onclick="closeForm()" style="width: 240px">Close</button>
396     </div>
397     </form>
398 </div>
399 </div>
400
401 <script>
402 function openForm() {
403     document.getElementById("myForm").style.display = "block";
404 }
405
406 function closeForm() {
407     document.getElementById("myForm").style.display = "none";
408 }
409 </script>
410
411
412     <script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>
413     <script src="http://netdna.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
414     <script type="text/javascript">
415
416     </script>
417     </body>
418 </html>
```

Ln 419, Col 8 Spaces: 2 UTF-8 CRLF HTML Go Live

Figure 6.16: Dashboard 2 Implementation

## 6.2 Output Screen:



Figure 6.17:Home Page

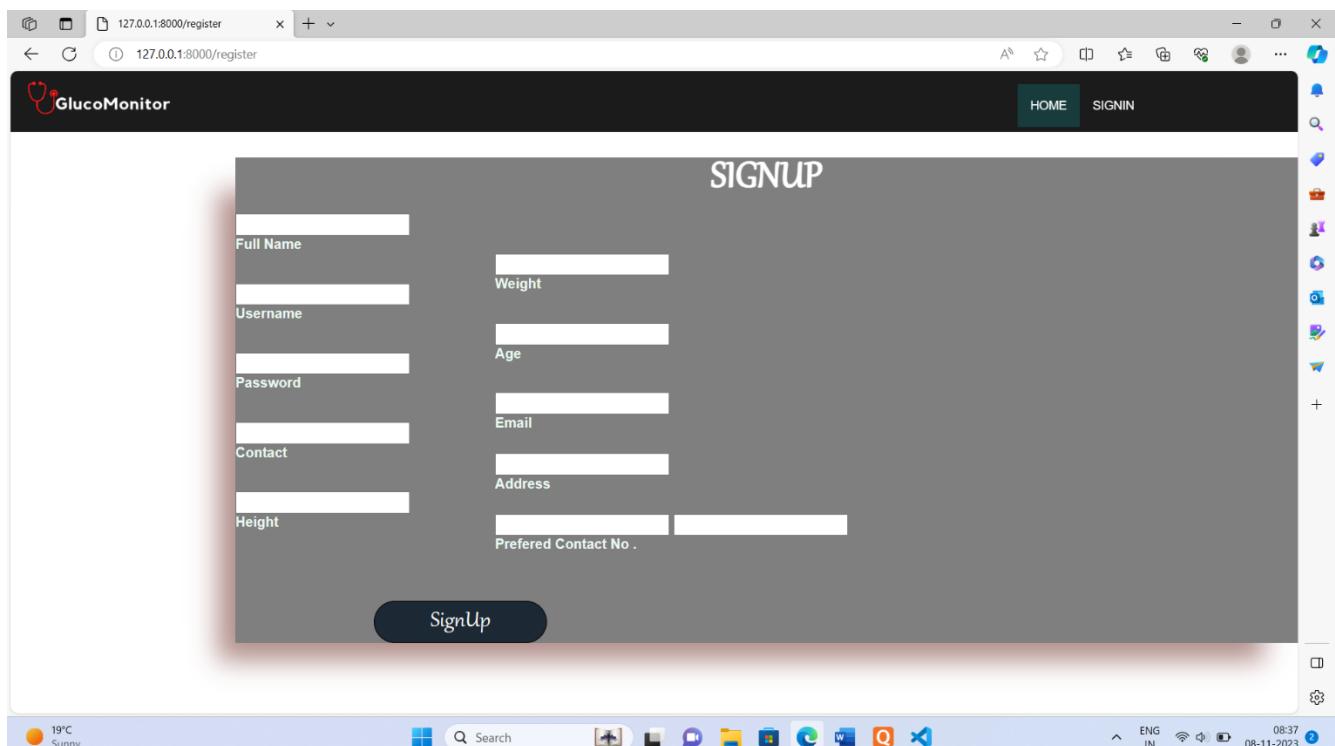


Figure 6.18: SignUp Page

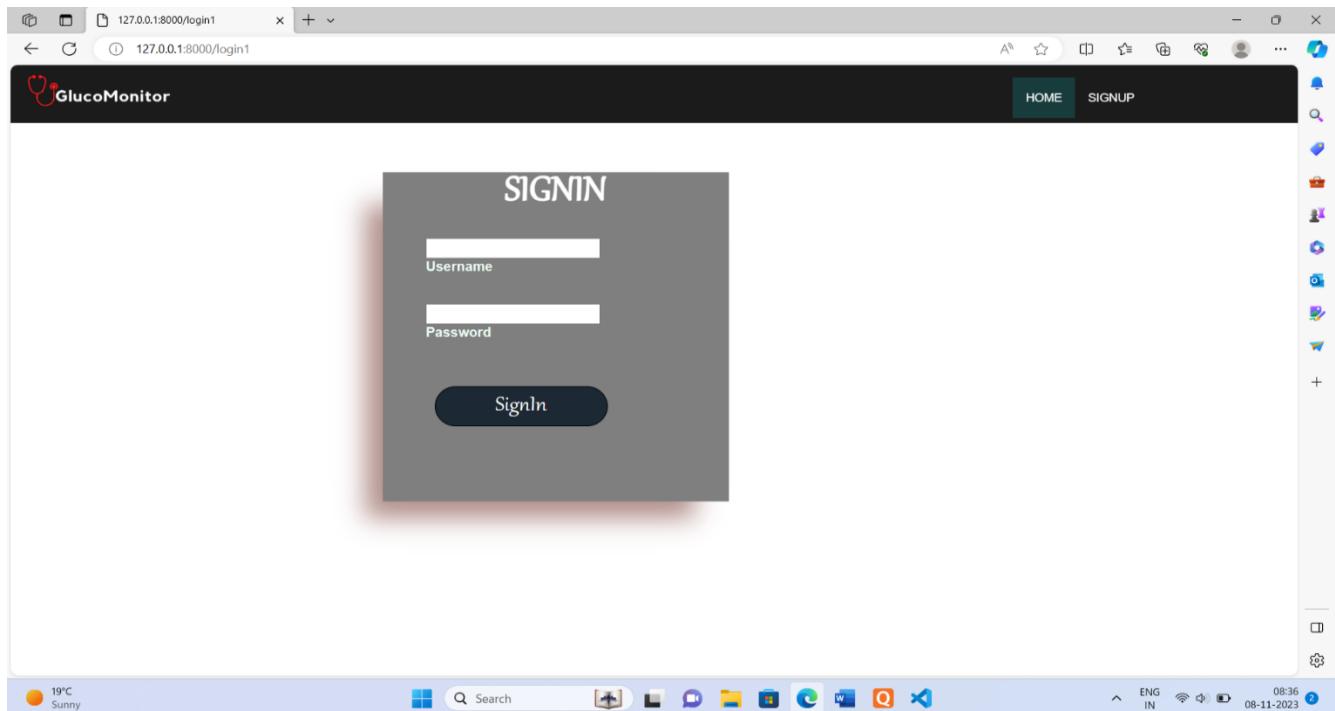


Figure 6.19: SignIn Page

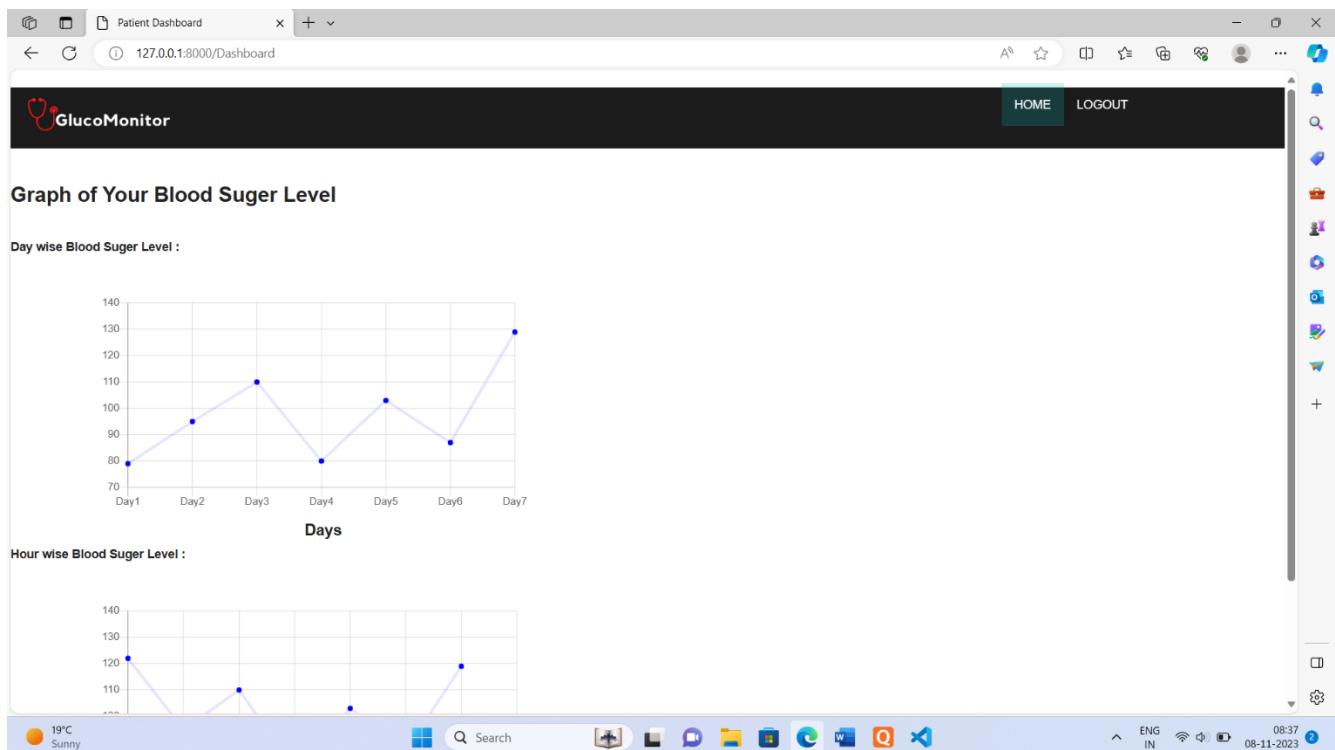


Figure 6.20: Dashboard Page

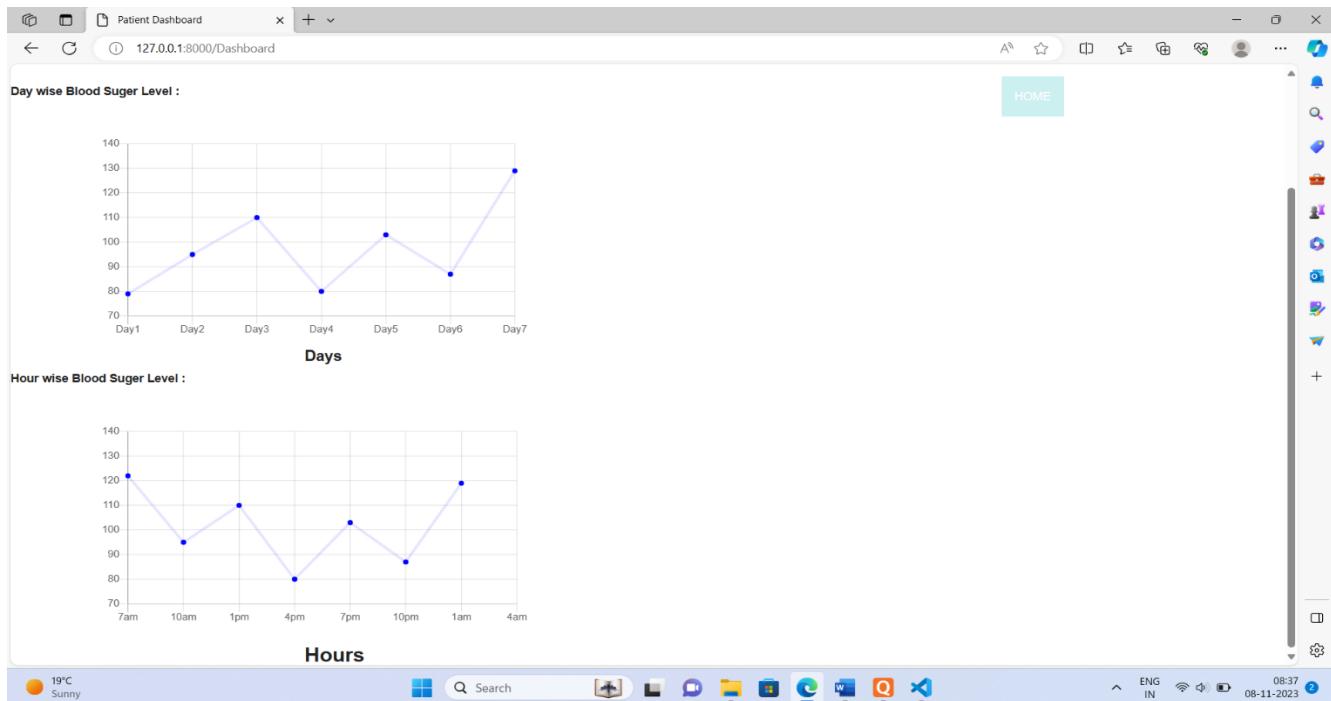


Figure 6.21: Dashboard 2 page

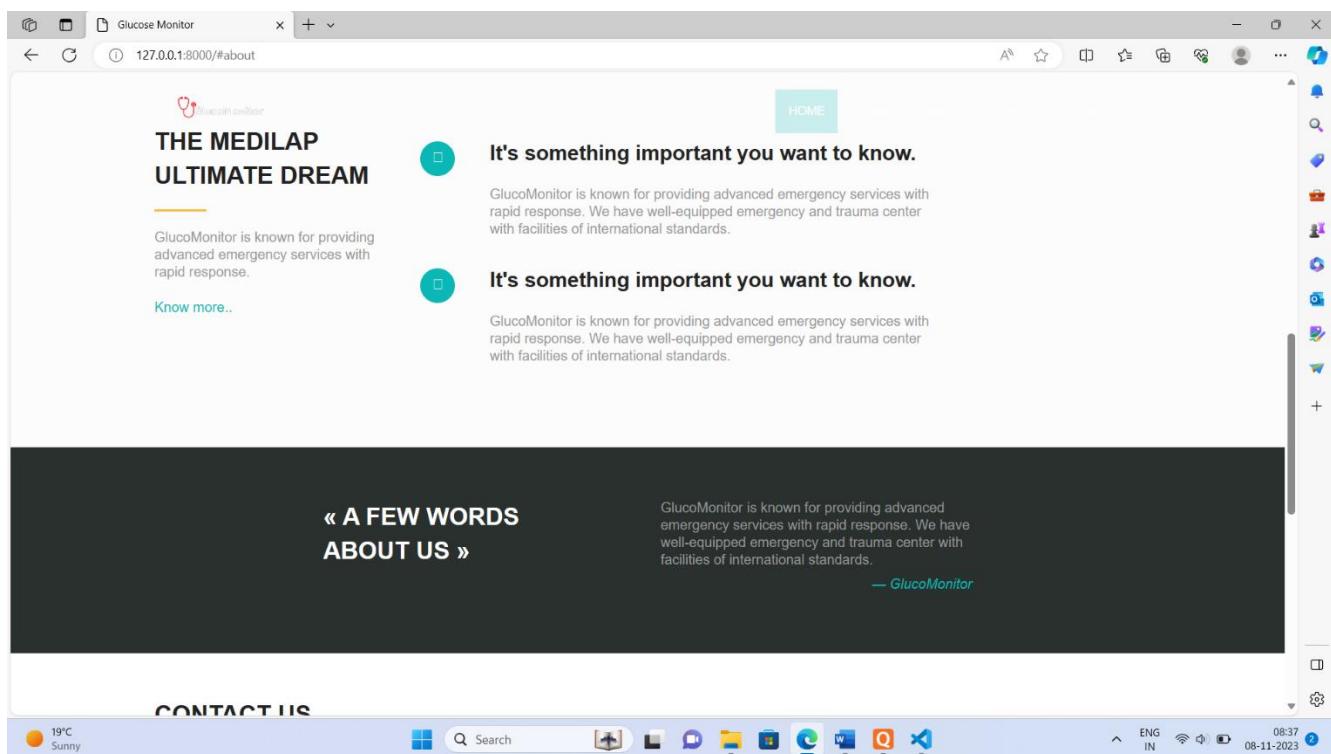


Figure 6.22: About Us Page

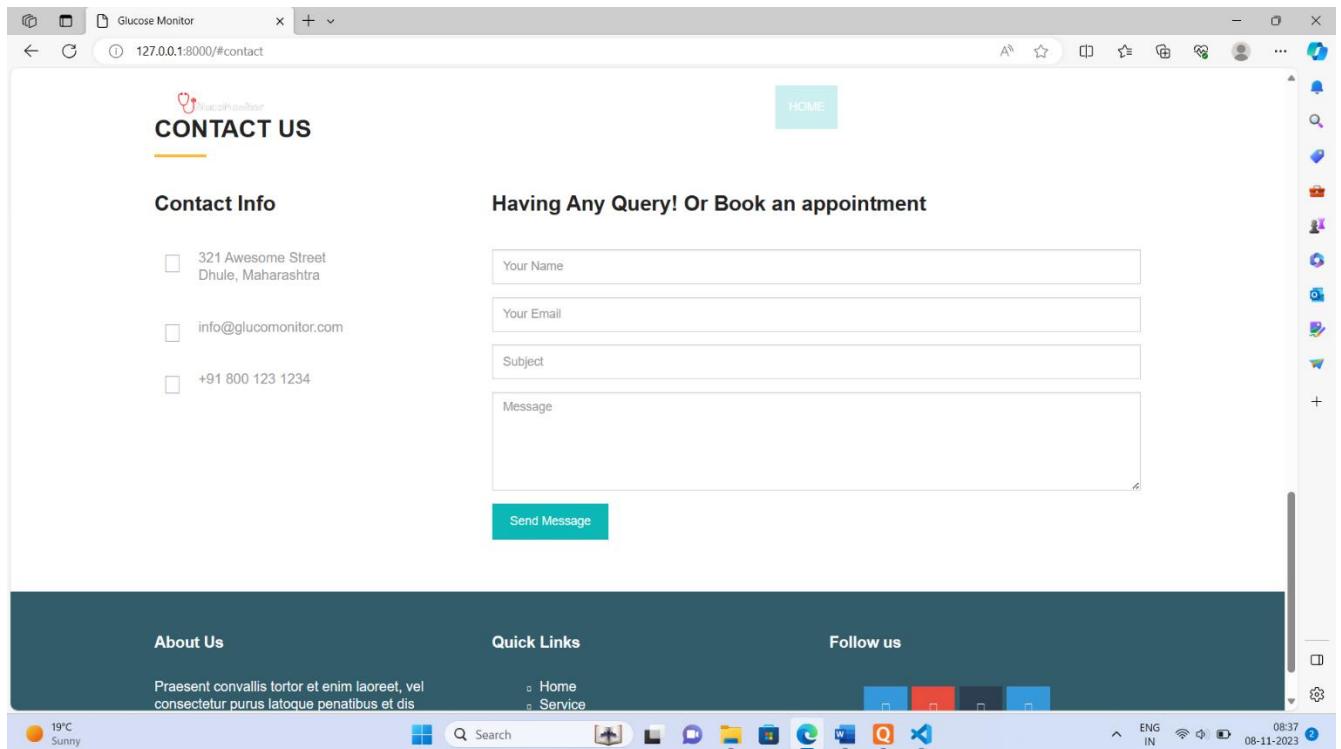


Figure 6.23: Contact Us Page

# **Chapter 7: Feasibility Study**

## **7.1 Introduction to Feasibility Study:**

The feasibility study for the "Real-time Prediction of Sugar Level using Sweat through Blood Sugar Level Band" project evaluates its viability and potential success. This study aims to assess technical, operational, economic, and schedule feasibility, analysing the device's capability to predict blood sugar levels from sweat components. It seeks to determine market demand, technological challenges, financial viability, and user acceptance to gauge the project's potential for development and implementation.

## **7.2 Economical Feasibility:**

The economic feasibility analysis aims to evaluate the financial viability and benefits of implementing the Blood Sugar Level Band project. This assessment explores the project's potential returns on investment, cost-benefit analysis, and overall financial impact.

**Development Costs:**

**Hardware:** Estimation of costs related to the acquisition of sensors, microcontrollers, and manufacturing expenses.

## **7.3 Time Feasibility:**

The project aims to create a system capable of predicting sugar levels using a wearable blood sugar level band. The time feasibility study evaluates the project's practicability concerning the anticipated time required for development, testing, and deployment.

### **Project Phases:**

- Research and Planning: Understanding existing technologies, planning system architecture, and feasibility assessment.
- Design and Prototyping: Creating wireframes, algorithms, and initial prototypes for system functionality.
- Development: Building the software and hardware components, integrating algorithms, and refining the system.
- Testing: Thoroughly testing the system for accuracy, reliability, and usability.
- Deployment: Implementing the solution, user training, and ensuring seamless functionality.

### **Time Estimation:**

- Research and Planning: 4 weeks
- Design and Prototyping: 6 weeks
- Development: 16 weeks
- Testing: 8 weeks
- Deployment: 4 weeks

## **Chapter 8: Conclusion**

An Android-based health monitoring network system utilizes Bluetooth communication to facilitate the monitoring of various health parameters gathered from multiple acquisition modules. Experimental results demonstrate the system's effective operation and its ability to concurrently support multiple acquisition devices. Diabetes is a long-term illness that's hard to prevent or cure completely. Doctors use family history to guess how likely someone might get it. Finding it early helps a lot. Nowadays, smart technology like AI is getting popular because it's really helpful. This new system uses AI to guess if someone might get diabetes by checking things like age, blood sugar, and family history. It tries not to poke or prod too much for checking sugar levels, using sweat instead. All the info is stored online. As these smart technologies get better, they could really help out more in managing diabetes.

### **Future Scope:**

Revaluation of a more suitable and efficient power source for enhancing the design. Further improvements can be made to enhance the accuracy of the device. For example, implementing specific precautions and procedures during ECG and body temperature measurements to optimize vital sign readings.

## Chapter 9: Reference

- [1] Johan Wannenburg and Reza Malekian, "Body Sensor Network for Mobile Health Monitoring, a Diagnosis and Anticipating System," in IEEE SENSORS JOURNAL, VOL. 15, NO. 12, December 2015.
- [2] J. Achten, and A. E. Jeukendrup, "Heart rate monitoring – Applications and Limitations," Sports Med, vol. 33, no. 7, pp. 517-538, 2003.
- [3] Esther Renner, Nadine Lang, Bernd Langenstein, Matthias Struck and Thomas Bertsch, "Validating sweat ammonia as physiological parameter for wearable devices in sports science," in IEEE 2020.
- [4] Maryamsadat Shokrehodaei, David P. Cistola, Robert C. Roberts, and Stella Quinones, "Non-Invasive Glucose Monitoring Using Optical Sensor and Machine Learning Techniques for Diabetes Applications," in IEEE, vol. 9, 2021.
- [5] Shilpa C Venugopal and Sruthi M, "Pre-Prediction of Diabetes and Sweat based real time Glucose monitoring system," International Journal of Applied Engineering Research ISSN 0973-4562 Volume 1, Number 1, 2020.
- [6] Dr. B. Sargunam and S. Anusha, "IoT Based Mobile Medical Application for Smart Insulin Regulation," in IEEE 2019.
- [7] Andrew P. Bradley, Nahla H. Barakat," Intelligible Support Vector Machines for Diagnosis of Diabetes Mellitus" IEEE Transactions on information technology in biomedicine, vol. 14, no. 4, July 2010.
- [8] Aswathi Anand P, Maya L Pai, "Artificial Neural Network Model for Identifying Early Readmission of Diabetic Patients," International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-6, April 2019.
- [9] Jungil Choi, Tyler Ray, "Soft, skin-interfaced wearable systems for sports science and analytics, "Science Direct, Biomedical Engineering 2019, 9:47-56.
- [10] Chouvarda I., Kavakiotis I, Maglaveras N, Salifoglou A, Tsave O, Vlahava I, "Machine learning and data mining methods in diabetes research". Computer Structure Biotechnology J. 2017, 15:104-16.
- [11] Han Wu, Jian He, Shengqi Yang, Xiaoyi Wang, Zhangqin Huang, "Type 2 diabetes mellitus prediction model based on data mining", Informatics in Medicine unlocked, Volume 8 2018, page 100-107.
- [12] Hyunjae Lee, "Wearable/disposable sweat- based glucose monitoring device with multistage transdermal drug delivery module". Science Advances, 8 March 2017.
- [13] E. J. Lahoda, C. Liu and L. B. Wingard, "Electrochemical evaluation of glucose oxidase immobilized by different methods", Biotechnology eng. 17: 413-22, 1975.
- [14] Maria Viqueira Villarejo, Begona Garcia Zapirain, "A Stress Sensor Based on Galvanic Skin Response (GSR) Controlled by ZigBee," Sensors 2012, 12.
- [15] Dr. Josephine Nirmala Many, R. M. Sobanal "Effect of Hydration Education on College Athlete's Hydration Status and KAP," International Journal of Science and Research (IJSR), 2012.

- [16] Dr. Rajwant Singh Assistant Professor, University of Delhi "Stress Role in Sports Performance of Athlete's," International Journal of Physical Education, Sports and Health 2017; 4(3): 278-280.
- [17] Prof. S. M. Rajbhoj, Soniya Lakudzode, "Review on human stress monitoring system using wearable sensors", International Research journal of Engineering and Technology, Volume 03 Issue: 04, April 2016.
- [18] N. Ramya, K. Nivetha, N. Ramya, R .Thendral," Blood Glucose Measurement By Sweat Using Arduino", Journal of Engineering, Scientific Research and Application, DSEC publication, Volume 4,Issue 2, July-December 2018.
- [19] N. Sneha, Tarun Gangil, "Analysis of diabetes mellitus for early prediction using optimal features selection", Springer International publishing, 6th February 2019.