Software Requirements Specification

Prediction of Diabetes Web App

CPSC 462

Professor: Dr. Lidia Morrison

Department of Computer Science

California State University, Fullerton

Spring, 2022

Group 4:

Fulya Kocaman

Tilak Ghorashainee

Trinh Hoang

Spandhana Gandlkurma

Vyshnavi Reddy Kamballapally

Table of Content

Table of Figures

1.0 Introduction

1.1 Purpose

This document aims to describe the software requirements for our 462-group project in detail. This document focuses on how our Diabetes Web App predicts whether a person has diabetes using our Machine Learning algorithm, including the software requirements and overall goals.

1.2 Scope of the problem

Diabetes is a common chronic disease. Diabetes is among the most widespread chronic diseases in the US. Early diagnosis can lead to lifestyle changes and more effective treatment. This app predicts diabetes as an essential first stage that can lead to helping patients improve treatment. This app is an important tool for public and public health officials. With the rapid development of machine learning, this diabetes prediction application can predict at higher accuracy and efficiently be used by anyone.

1.3 Intended Audience

The Intended Audience of this document includes Dr. Lidia Morrison and the members of our group to verify the functionality of the software. Other users have all students enrolled in CPSC 462 for (Spring 2022) at California State University, Fullerton. The application is also intended for adults 18 or older who want to know whether or not they have diabetes.

2.0 Overall Description

2.1 User Objectives

The user will input their age, high blood pressure, high cholesterol, smoking, obesity, diet, exercise, and Body Mass Index. Our application uses the most accurate prediction by collecting the data from users who want to check whether they have diabetes.

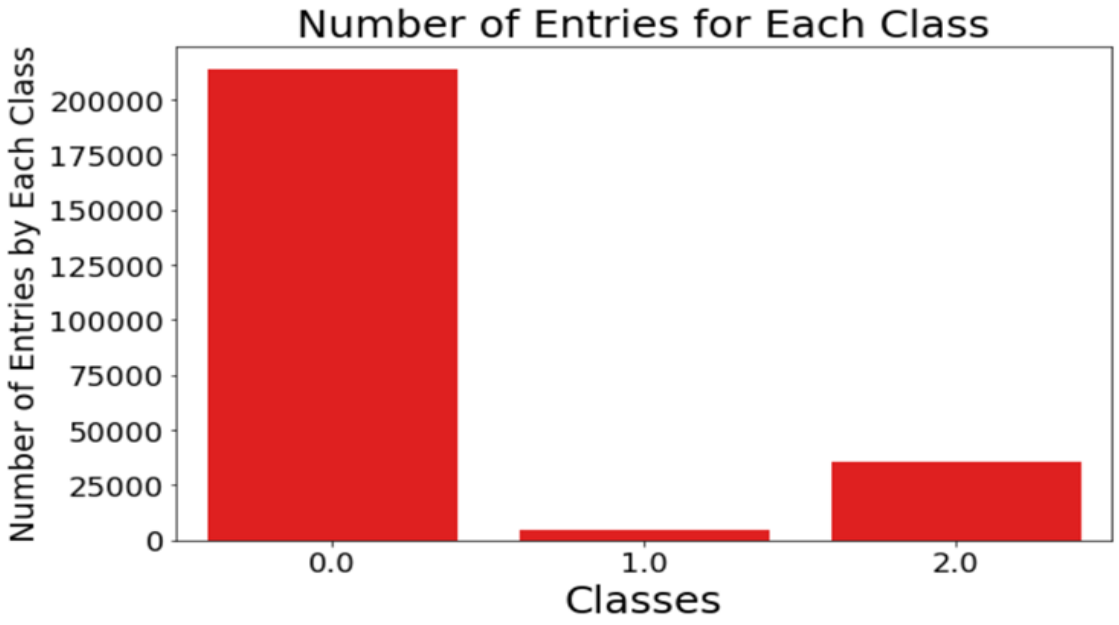

Figure 2.1.1 Distribution of Classes over the Entire Dataset

2.2 Diabetes Web App Outline

- The dataset from Kaggle for the year 2015 contains 253,680 survey responses from the Behavioral Risk Factor Surveillance System (BRFSS), a health-related telephone survey that is collected annually by the CDC.
- The target variable has 3 classes which are 0 is for no diabetes or only during pregnancy, 1 is for prediabetes, and 2 is for diabetes. Figure 2.1.1 shows how unbalanced the dataset over the classes.
- 21 feature variables: High Blood Pressure, High Cholesterol, Smoking, Obesity, Age, Sex, Race, Diet, Exercise, Alcohol Consumption, BMI, Household Income, Marital Status, Sleep, Time Since Last Checkup, Education, Health Care Coverage, and Mental Health.
- Support Vector Machine as a Machine Learning Classification Algorithm

2.3 Product Functions

Our system will collect the data from the user and analyze their data using our trained model. Our proposed model is trained by the Support Vector Machine (SVM) Algorithm, a supervised machine learning algorithm used for classification. Moreover, our web application is built using Flask, one of Python's high-performance web frameworks.

2.4 Operating Environment

Diabetes Web App is a web application that utilizes Python as a software language, HTM and CSS as web development, Flask as one of python's web frameworks, and PyhtonAnywhere as a hosting service for our Python website. Figure 2.4.1 shows how the files are organized in VS code of Windows 10.

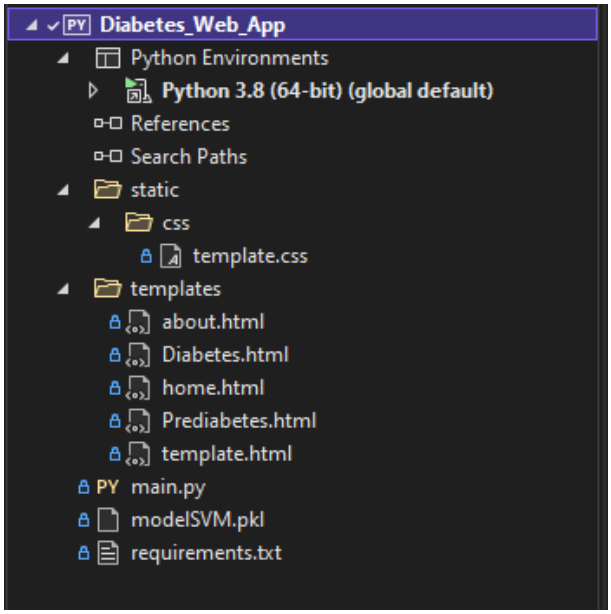

Figure 2.4.1 Project Files in VS Code

2.5 User characteristics

This application is intended for adults 18 or older who want to know whether or not they have diabetes. There are no gender restrictions or special skills required to operate this app. A user-friendly interface and informative data entry questions make the app very easy to use by everyone.

2.6 Design and Implementation Constraints

This application is intended to use as a desktop application in Windows and macOS operating systems. It is not intended to be a mobile application, yet it can be used in smart mobile devices with a bit of page size adjustment.

This app does not require the user to create an account, register to our website, or collect data from the user's information. The user solemnly uses it to check whether or not they have diabetes in order to raise public awareness of diabetes.

2.7 Assumptions

We assume that the user has access to reliable internet and a device to get connected to our Diabetes Web App website.

3.0 Functional Requirements

3.1 The website shall display Homepage

Description

On entering the diabetes URL, the website will load in 3000 milliseconds

Pre-condition – User has the correct website address

Post-condition – The homepage is loaded

3.2 The website shall display a various hyperlinks

Description

A homepage is displayed and the user can access any hyperlinks.

Pre-condition – User is using valid diabetes prediction website.

Post-condition – The hyperlinks in home page of the app is displayed.

3.3 The user shall use the check your diabetes link for diagnosis

Description

The application shall let the user access the form page for prediction.

Precondition – User selects the predict diabetes link in the page.

Post-condition – The user is required to enter the input fields like health factors, physical exercise, diet and mental wellness and clicks on the predict button.

3.4 The application displays the prediction value on the screen

Description

After entering the input values for different fields and clicks on the predict button

Pre-condition – User entered the valid values and clicks on the predict button

Post-condition – User is prompted with prediction result value on the screen

3.5 The app shall display accurate prediction

Description

The website should display accurate prediction depending on the values provided by the user.

Pre-condition – User was able to provide valid values in the index form page.

Post-condition – User will have the most accurate prediction whether he/she is diabetic or not depending on his lifestyle values

3.6 The website shall allow users for multiple predictions

Description

The user can make multiple predictions in the diabetes prediction application.

Pre-condition – User selected diabetes prediction page

Post-condition – User can enter another set of values and click on predict button, can have multiple tries.

3.7 The application shall display about diabetes link on the homepage

Description

The diabetes prediction website shall display aboutdiabetes URL in the homepage

Pre-condition – The user is in homepage and selected about diabetes link

Post-condition – If user selected aboutdiabetes link in the homepage, the information about diabetes, prediction and diet recommendation regarding diabetes.

3.8 The application should display Diabetes Info

Description

The user can select Diabetes link and retrieve information

Pre-condition – User selected the “Diabetes” button from the home page menu.

Post-condition – User lands on the Diabetes page which contains information about diabetes, recommended diet, lifestyle changes

3.**9 The application should display Pre-Diabetes Info**

Description

The user can select Pre-Diabetes link and retrieve information

Pre-condition – User selected the “Pre-Diabetes” button from the home page menu.

Post-condition – User lands on the Pre-Diabetes page which contains information about diabetes, recommended diet, lifestyle changes

3.10 The application selects articles link on the homepage

Description

The user clicks on the articles link in the prediction homepage.

Pre-condition – The user is on the valid website URL.

Post-condition – The user lands on the articles link which has many articles related to diabetes.

3.11 The application URL must load within 3 seconds

Description

The user enters the website URL, and it must be loaded in three seconds.

Pre-condition – The user is using the valid website URL.

Post-condition – The user lands on the diabetes prediction

The website and the site should be loaded in 3 seconds.

3.12 The application should display the result within 3 seconds

Description

The user clicks on the predict button and the result is displayed.

Pre-condition – The user filled the prediction form with valid input values and clicks on the predict button.

Post-condition – The prediction value is displayed within 10 seconds.

3.13 When a user enters a value not in the range of a field, the system shall display an error message

Description

The user enters out of range value and error message is displayed.

Pre-condition – The user is on prediction form, and enters input values.

Post-condition – The application displays error message when out of range input values are entered.

3.14 When the home button is clicked, the application routes to homepage.

Description

The user clicks on the home button.

Pre-condition – The user is on the diabetes prediction page and clicks on the home button.

Post-condition – The user routes to home page screen when home button clicks.

3.15 The application displays error message when the input field is empty

Description

The user selects prediction empty input values.

Pre-condition – The user clicks on the predict button with empty input field values.

Post-condition – The application prompts user with message to fill out empty input fields.

4.0 Quality Attributes of Diabetes Web App

- The application shall respond to predict diabetes in less than 3 seconds after clicking the “Predict” button.

- The application shall give errors for the user to correctly input each data entry.
- After clicking the home page button, the application shall reset all the previous inputs.

5.0 Non-Functional Requirement

Security:

- The system will require the user to input a valid number for each data entry to predict diabetes correctly.
- The system will require the user to answer all 21 questions to predict diabetes.

Usability:

- After starting the web application, the software will load within 3 seconds.
- When the "Predict" button is clicked, the system will predict within 3 seconds.

6.0 Interface Requirement

6.1 GUI

Figures 6.1.1, 6.1.2, and 6.1.3 show the User Interface for Diabetes Web App Home Page, Diabetes Info Page, and Diabetes Result Page, respectively.

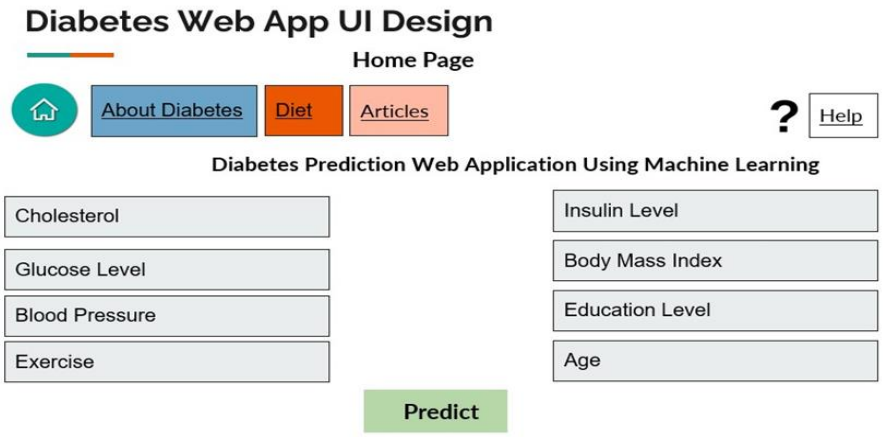

Figure 6.1.1 User Interface of Diabetes Web App Home Page

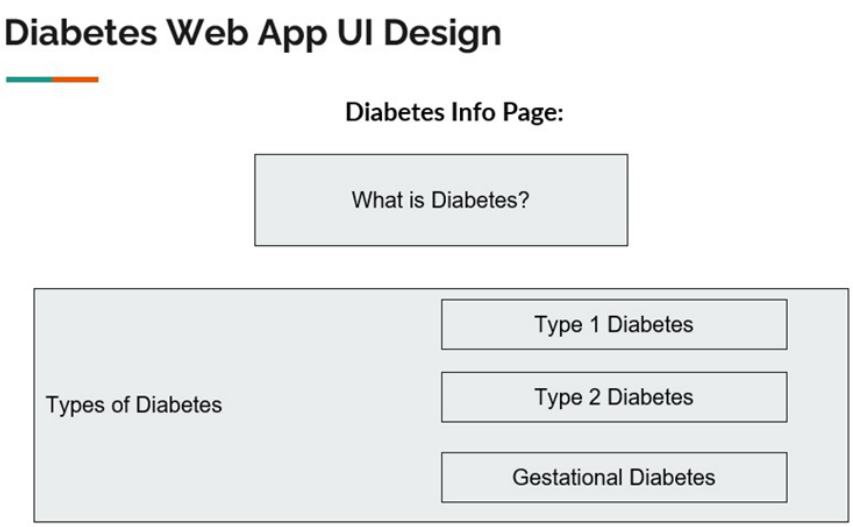

Figure 6.1.2 User Interface of Diabetes Info Page

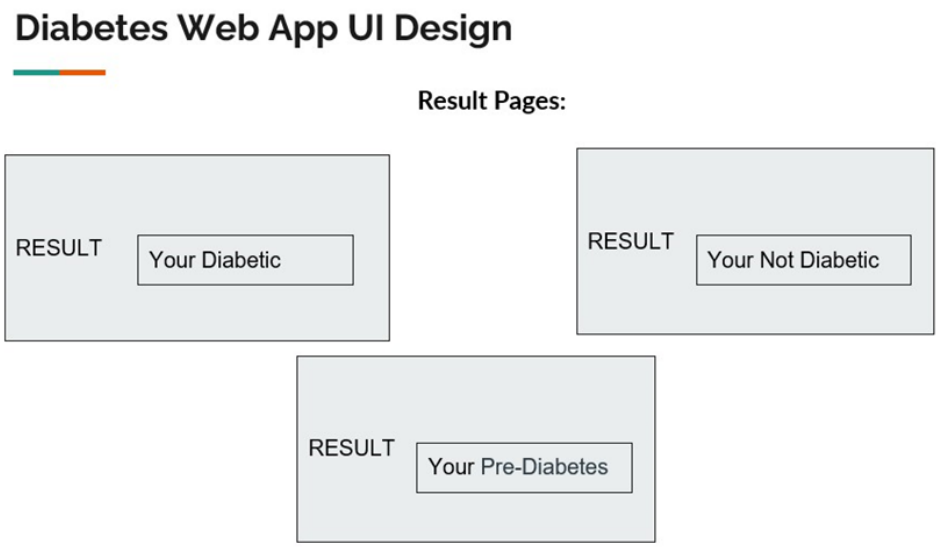

Figure 6.1.3 User Interface of Diabetes Result Page

6.2 Hardware Interfaces

The application does not require special hardware to function correctly. However, the user needs to have access to reliable internet and a laptop or a smart device to get connected to our Diabetes Web App website.

6.3 Software Interfaces

The application requires the user to have an Operating Environment, either Windows or macOS.

7.0 SWOT Analysis for Diabetes Web App

Strengths:

S1: All the functional requirements are met in developing this app before the deadline

S2: We have many available software tools and techniques required for developing this web app.

S3: Developing this web is completely free.

Weakness:

W1: Developing a web application is new to everyone in the team. There are challenges to developing a python framework that will work with the Machine Learning algorithm and deploy the app online. Our weakness was to spend more time learning how to use these software tools, not to improve the functionality of our app.

W2: We could have added more functionalities, such as creating a database for the user to create an account to store their results and have a place to see all previous submission results.

Opportunities:

O1: We all know how expensive healthcare is in the US. This free app gives user to be aware of their conditions in the comfort of their home instead of visiting the doctor's office and

getting blood work done. This app provides an opportunity for everyone to see whether or not they have diabetes as often as they would like.

O2: This project encourages us to learn new software tools that make us well-rounded as an engineer and create something that will be beneficial for humanity.

Threats:

T1: Since we are using Machine Learning Algorithms, our app is only efficient as our algorithm. Our app will function poorly if our model is less accurate.

8.0 UML Diagram

8.1. Use Case Diagram

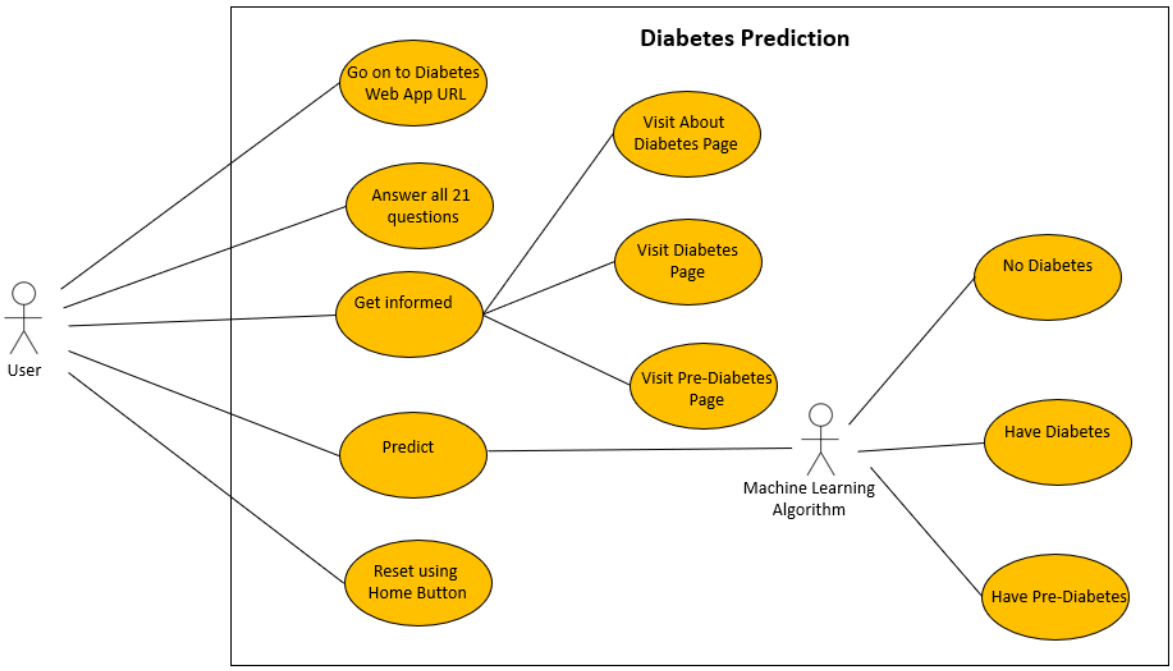

Figure 8.1.1 Use Diagram for Diabetes Prediction

8.2 Sequence Diagram

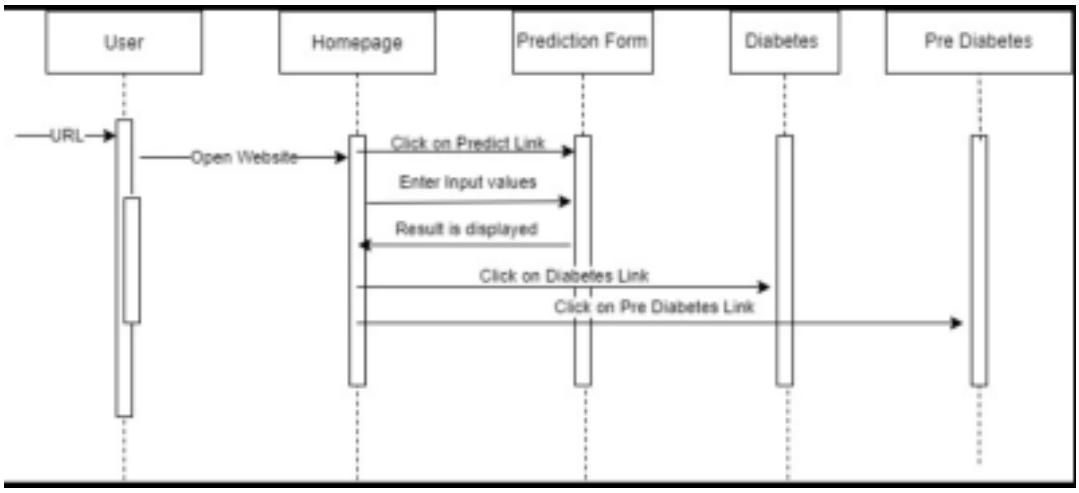

Figure 8.2.1 Sequence Diagram for Prediction Use Case

8.3 Activity Diagram for Prediction Use Case

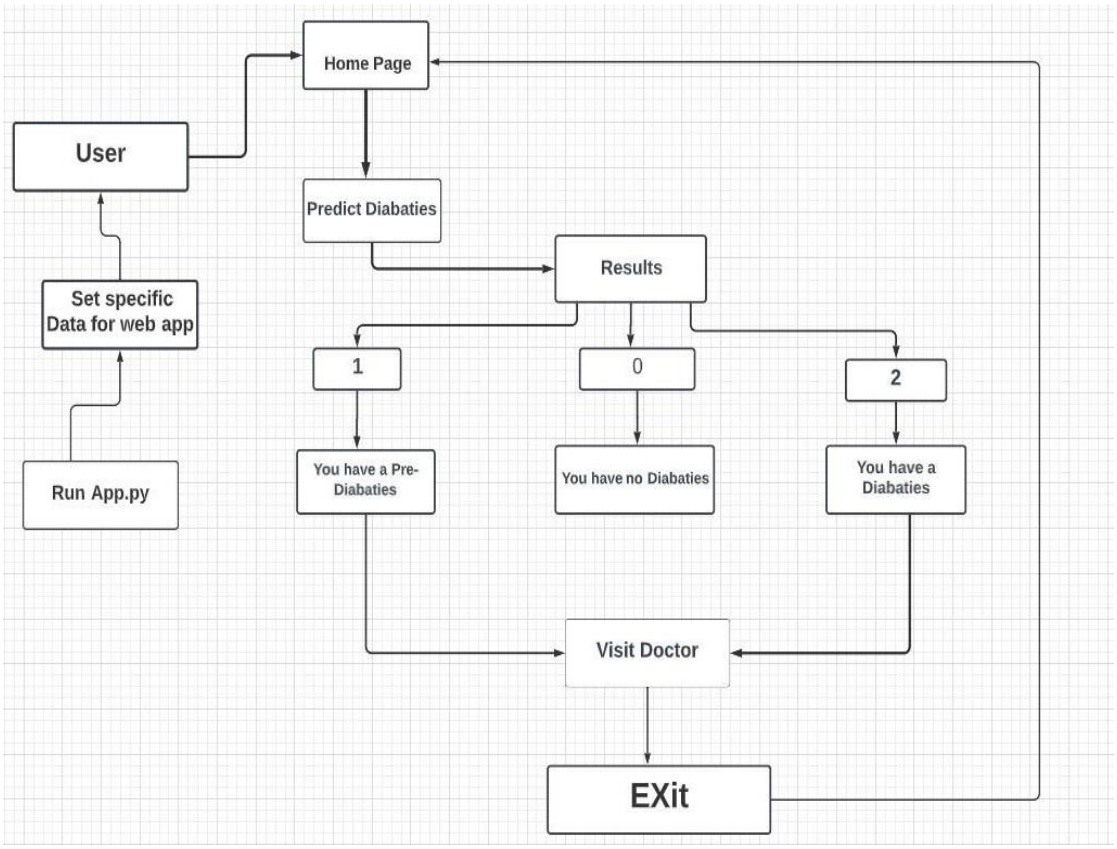

Figure 8.3.1 Activity Diagram for Prediction Use Case

APPENDIX A: User Operations Manual

1. Our Diabetes Web App can be reached by going to https://diabeteswebapp01.pythonanywhere.com/ from any web browser. Figure 4.2.1 displays the home screen of our application.

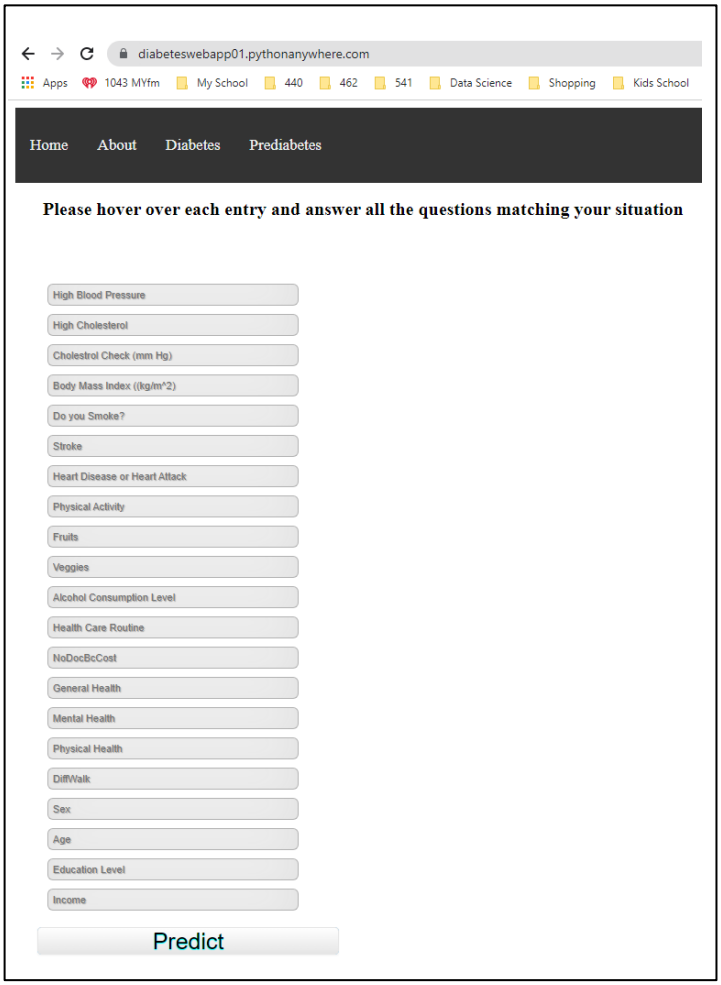

Figure Appendix A.1 Home Screen of Diabetes Web App

2. Figure Appendix A.2 shows a sample of how a user can see the information of each data entry to input their values correctly.

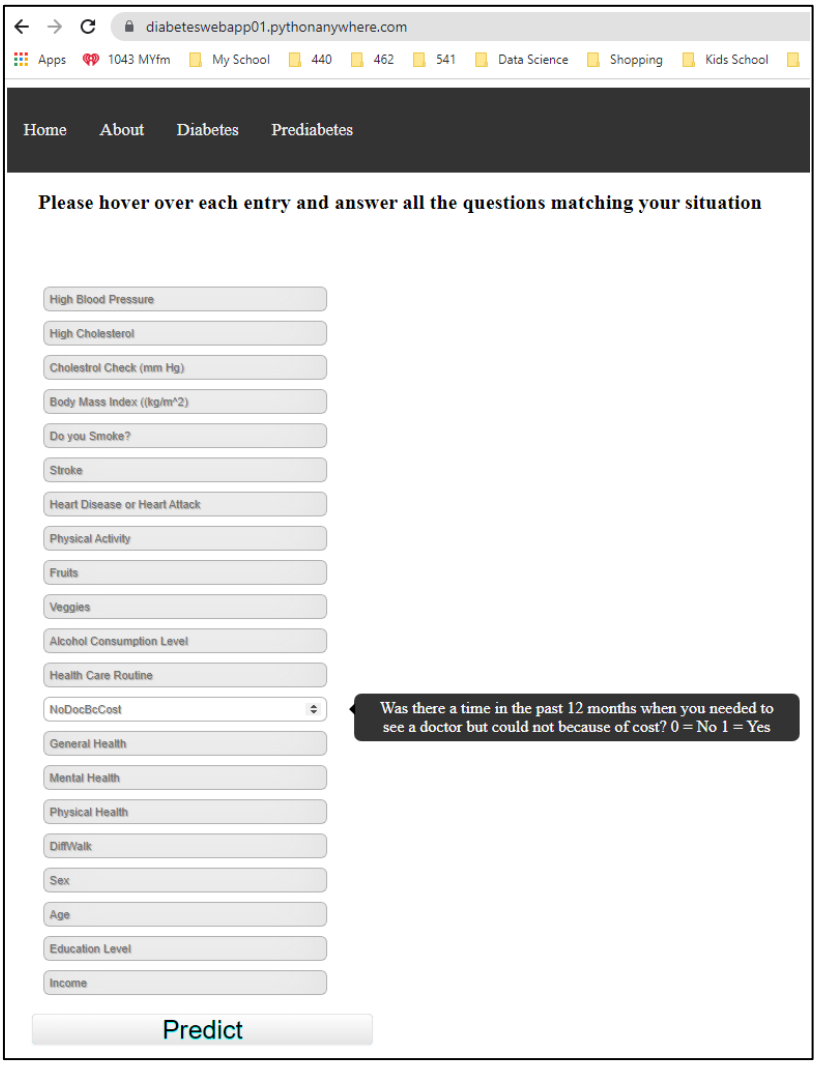

Figure Appendix A.2 Input Information of Diabetes Web App

3. Figures A.3, A.4, and A.5 display the additional pages implemented for this app about diabetes, diabetes, and pre-diabetes pages, respectively.

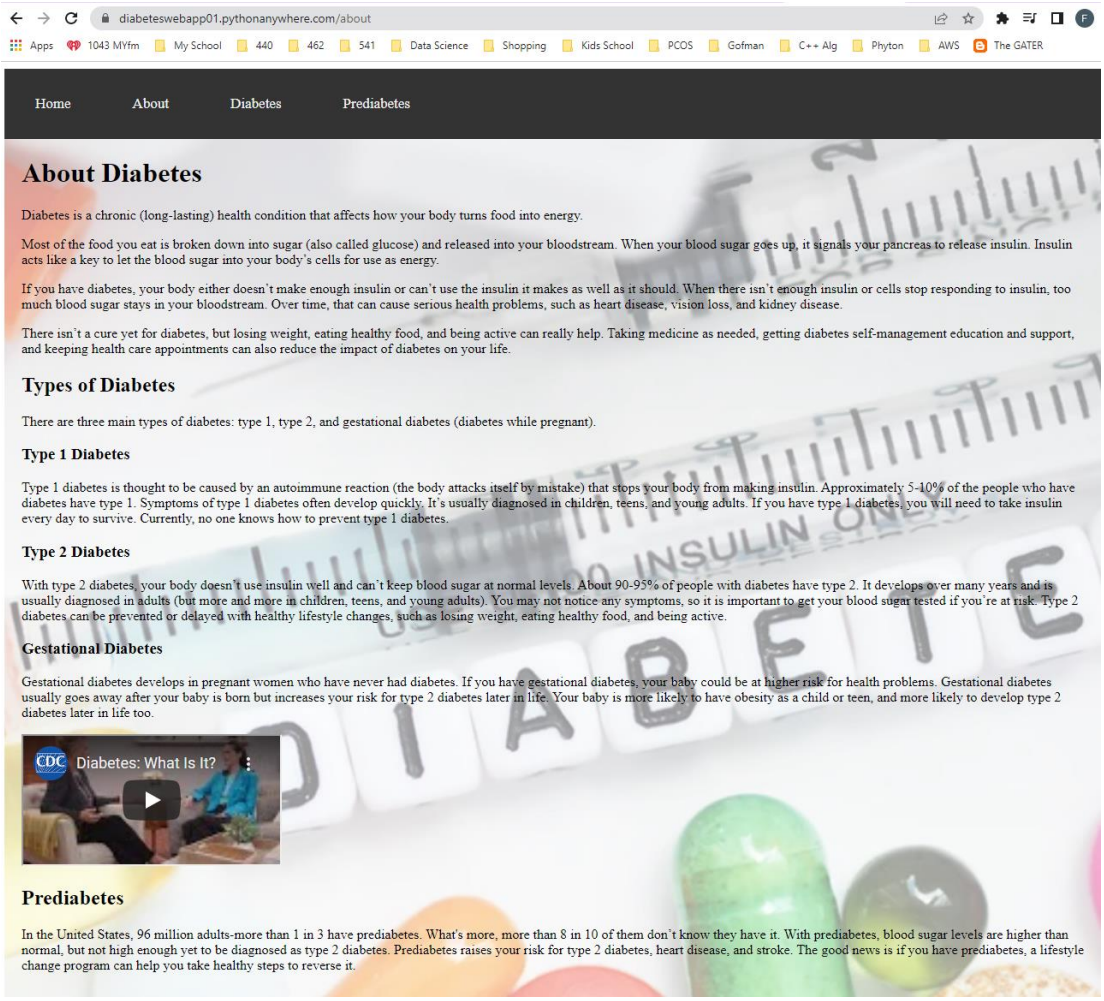

Figure Appendix A.3 About Diabetes Info Page

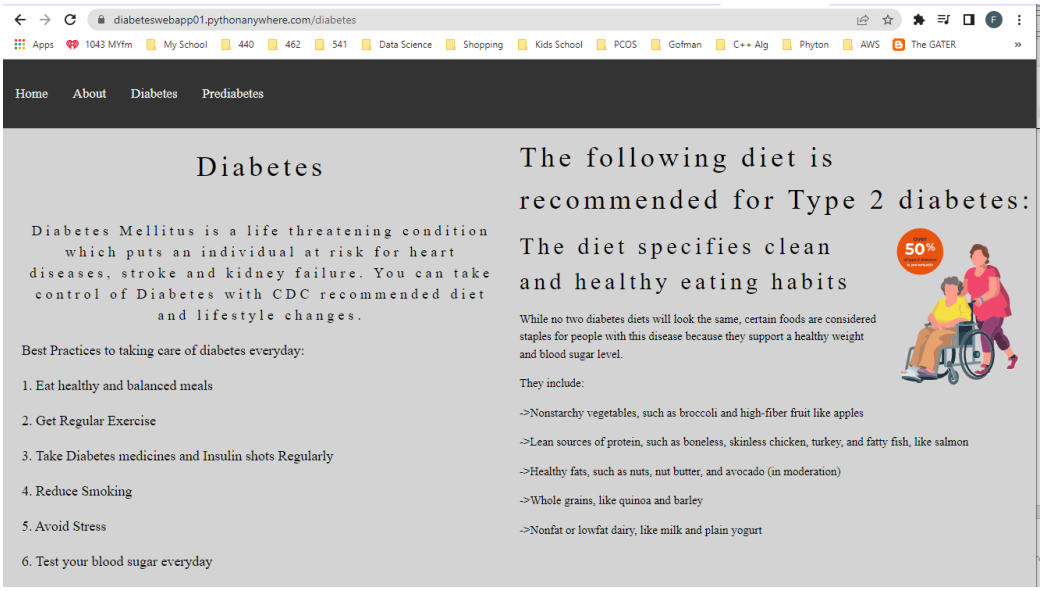

Figure Appendix A.4 Diabetes Info Page

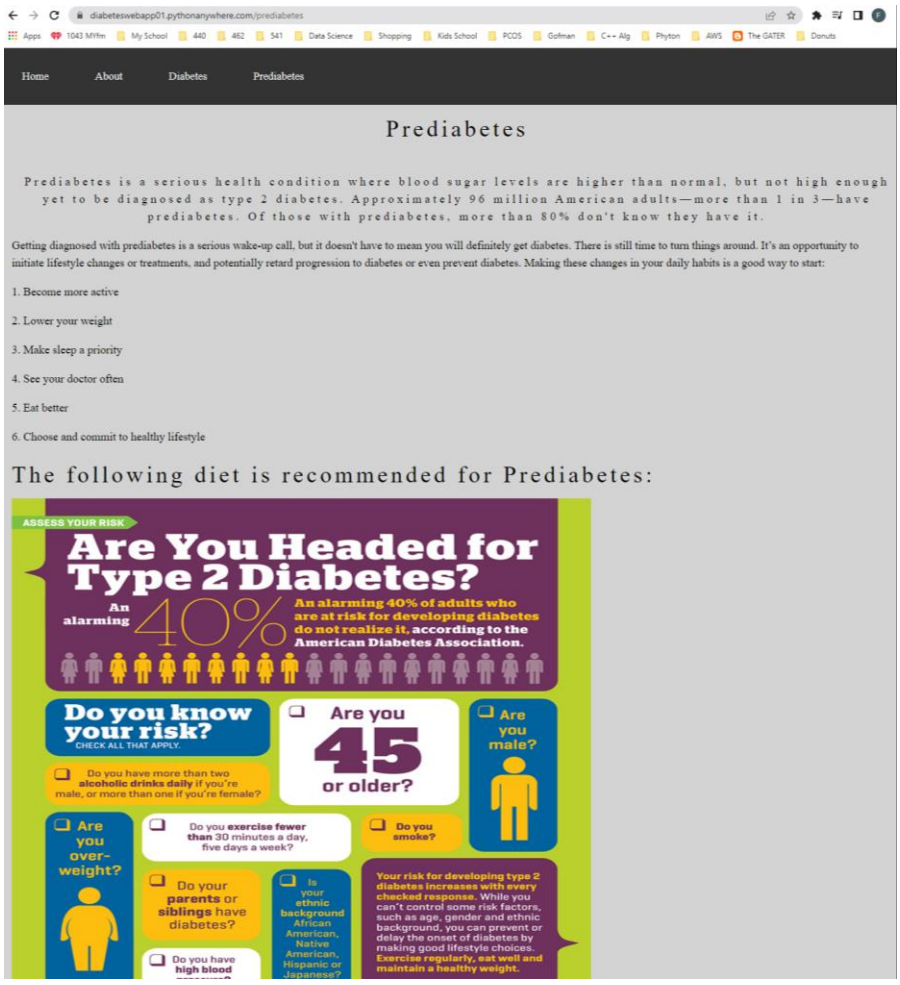

Figure Appendix A.5 Pre-Diabetes Info Page

4. Figure A.6 shows how our application predicts that you do not have diabetes.

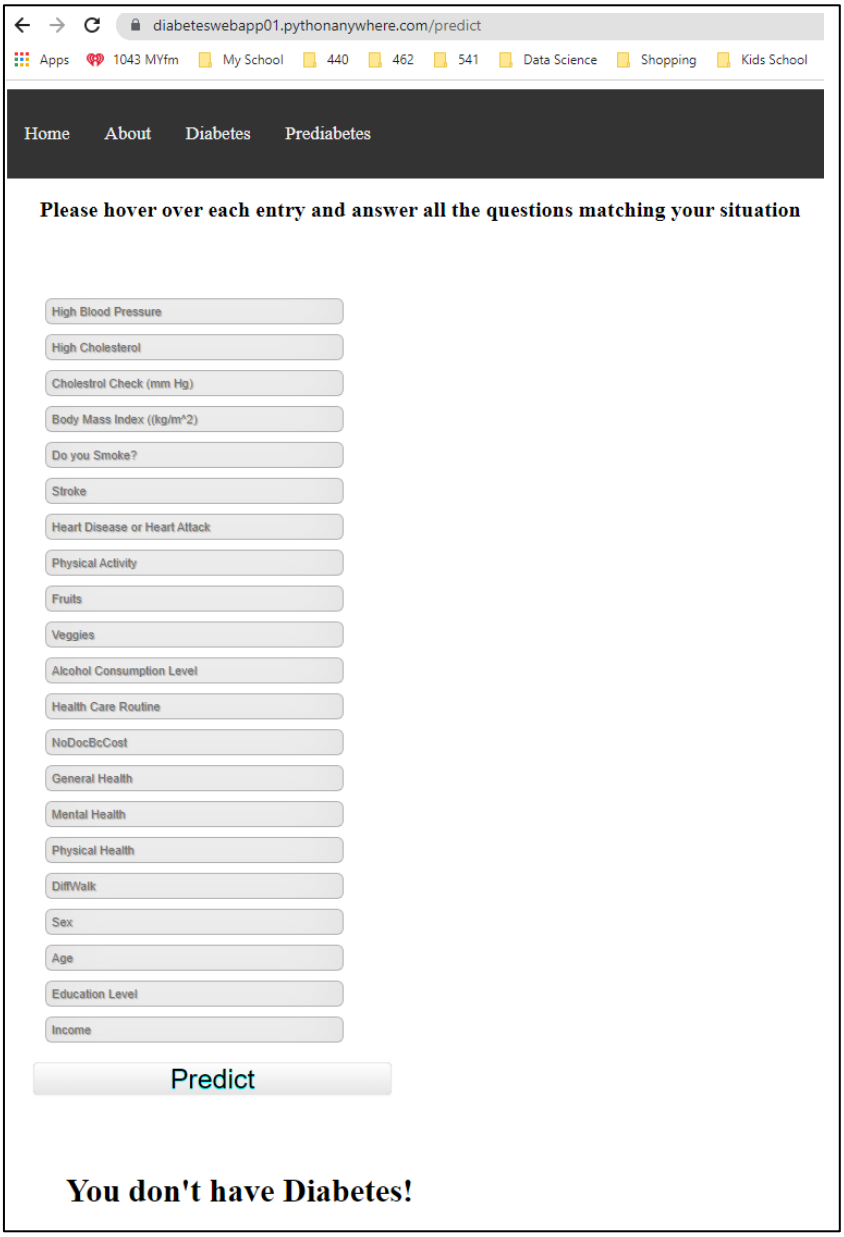

Figure Appendix A.6 No Diabetes Prediction Page

5. Figure A.7 shows how our application predicts that you have diabetes.

Figure Appendix A.7 Diabetes Prediction Page

APPENDIX B: Reference and Tools Used

References

- Software Requirements Specification for Banker Buddy from Fall, 2014.
- Lecture Notes and Slides by Dr. Lidia Morrison
- Learn more about BRFSS dataset: https://www.cdc.gov/brfss/index.html
- Download the BRSS Dataset: https://www.kaggle.com/code/alexteboul/diabetes-health-indicators-dataset-notebook/notebook
- Learn how to implement Flask framework: https://www.youtube.com/watch?v=Koh6Bp33hVQ
- Learn how to deploy a python app online using PythonAnywhere: https://www.youtube.com/watch?v=5jbdkOlf4cY

Tools Used

- VS Code
- Google Colab
- Google Doc
- Draw IO
- Microsoft Office