

# Python Project Report

---

## DIABETES AI TEST WEBAPP

Ayoub AKANOUN • Abderrahmane AARAB

Zakaria AKERDAD • Walid FAJRI

Amina AIT MHA

---

A report submitted in partial fulfillment of the  
OOP Python Master's class project

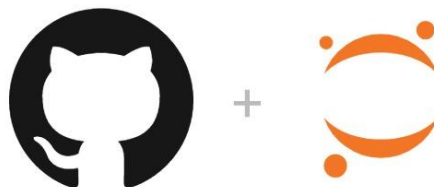


Department of Computer Science  
Moulay Ismail University, Meknes



## Abstract

This report presents a Django web application that utilizes machine learning to predict the likelihood of an individual having diabetes. The app uses a machine learning model that is trained on a dataset of patient health records and symptoms, and returns a prediction based on inputted data. The user interface is implemented using Django, providing a convenient and user-friendly way to access the prediction capabilities of the model. The app was evaluated using accuracy metrics and showed promising results, demonstrating its potential to aid medical professionals in the diagnostic process.



AbdoAarab/**Diabetes-Ai-Test**

Predict whether a person has diabetes or not using Machine Learning Algorithms

# Table of Contents

Abstract .....	2
List of Figures.....	4
Introduction .....	5
Chapter 1: Data Collection & Analysis .....	6
1.1 Dataset .....	6
1.2 Understanding the Data.....	7
Chapter 2: Machine Learning Model .....	11
2.1 Tested Learning Algorithms .....	11
2.2 Used Learning Algorithms.....	13
Chapter 3: Web Application.....	15
3.1 Tools & Technologies.....	15
3.3 Implementation.....	19
3.4 Web App Preview.....	20
Conclusion .....	23

# List of Figures

Figure 1: Git Repository ..... 2

Figure 2: Features information ..... 6

Figure 3: Dataset preview ..... 7

Figure 4: Data heatmap ..... 8

Figure 5: Age graph ..... 9

Figure 6: Age segregation ..... 10

Figure 7: Age pychart ..... 10

Figure 8: Model scores ..... 13

Figure 9: Home page ..... 20

Figure 10: Form page ..... 21

Figure 11: Results page ..... 22

# Introduction

Diabetes is a chronic condition that affects millions of people worldwide and can lead to serious health complications if not properly managed. Early diagnosis is crucial in the effective treatment and management of diabetes, but traditional diagnostic methods can be time-consuming and unreliable. This is where technology and machine learning can play a vital role in helping medical professionals to make informed decisions.

The aim of this report is to present the development and evaluation of the Django application and to demonstrate its potential as a tool for supporting diabetes diagnosis. The report will discuss the data and techniques used to build the machine learning model, as well as the implementation of the Django app and its **evaluation** results.

Machine learning techniques are a developing field of study with numerous potential applications. Machine learning technology will become increasingly important to healthcare professionals and health systems for extracting meaning from medical data as patient data becomes more readily available.

In recent years, machine learning algorithms demonstrated incredible precision in predicting a variety of medical phenomena. From creating better diagnostic tools to analyzing medical imagery. Investing in these artificial intelligence solutions is becoming the norm in the health care industry worldwide.

For the healthcare industry, machine learning is particularly valuable because it can help us make sense of the massive amounts of healthcare data that is generated every day within electronic health records. Using machine learning in healthcare like machine learning algorithms can help us find patterns and insights that would be impossible to find manually.

# Chapter 1: Data Collection & Analysis

## 1.1 Dataset

The National Institute of Diabetes, Digestive, and Kidney Diseases is the original source of this dataset. Based on specific diagnostic metrics present in the dataset, the app's goal is to diagnostically forecast whether a patient has diabetes or not. These samples were chosen from a bigger database under several restrictions. Particularly, all patients here are Pima Indian women who are at least 21 years old.

We focused on pregnant females in this scenario.

The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

### 1.1.1 Features

#### Feature Information

Pregnancies: Number of times pregnant
Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
BloodPressure: Diastolic blood pressure (mm Hg)
SkinThickness: Triceps skin fold thickness (mm)
Insulin: 2-Hour serum insulin (mu U/ml)
BMI: Body mass index (weight in kg/(height in m)^2)
DiabetesPedigreeFunction: Diabetes pedigree function
Age: Age (years)
Outcome: Class variable (0 or 1)

**Figure 2: Features information**

### 1.1.2 Preprocessing

One can see from a close look at the dataset that there isn't much of a requirement for data pretreatment. Due to the absence of any kind of missing values.

```
# Basic info of columnsabs
dataset.info()# the data has been cleaned , no lissing

<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
```

**Figure 3: Dataset preview**

## 1.2 Understanding the Data

Examining the correlation between different input features could be a useful technique to analyze our data. It can be applied to evaluate the nature, strength, and direction of relationships between different input features.



1.2.1 Heat Map

We used a heatmap in the figure below since it is an easy method to visualize all the information without any misunderstanding. Although we could have performed the same procedure with an array.

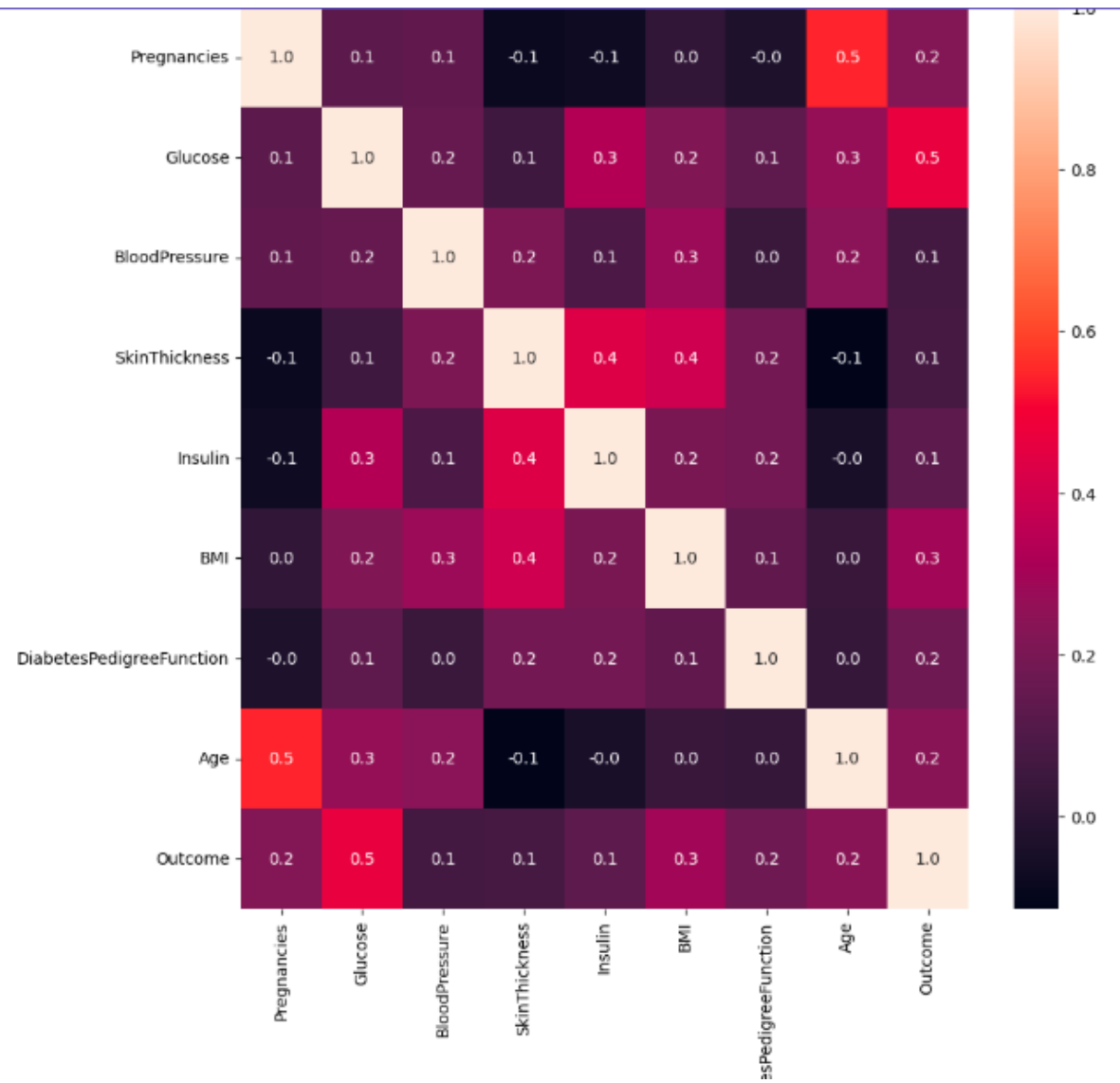
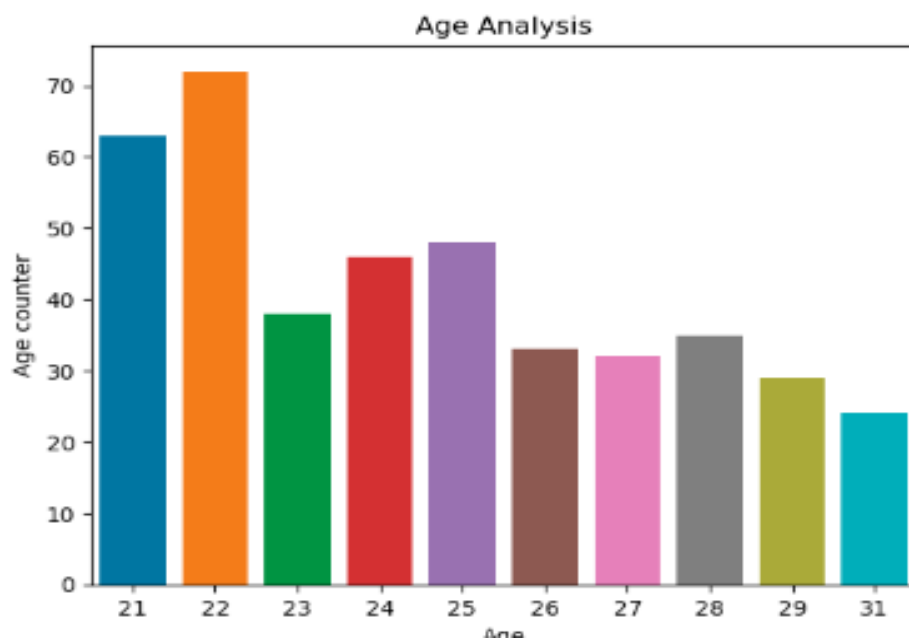


Figure 4: Data heatmap

### 1.2.2 Age Analysis

Age seems to be a significant factor in deciding whether someone is susceptible to diabetes, so we thought it could be useful to run some sort of analysis on this attribute. Simply by plotting the datasets age values, we get the following graph:



**Figure 5: Age graph**

Most participants were in their 20s.

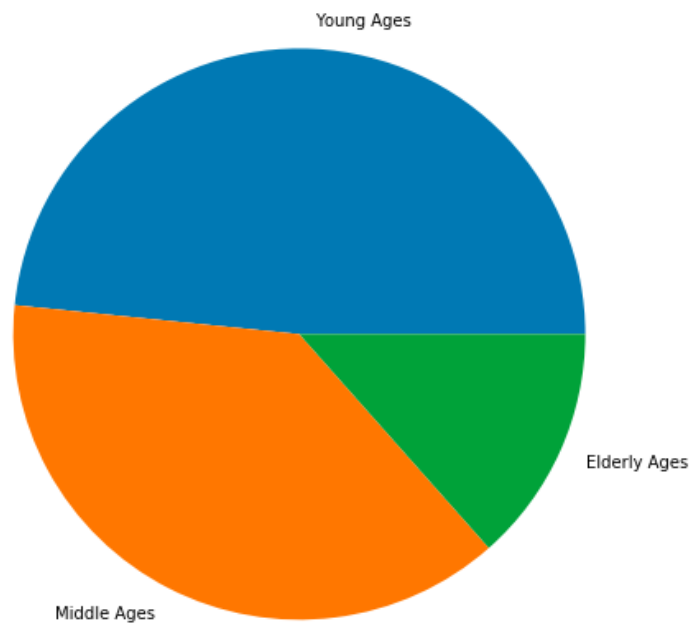
We consider a person to be young if their age is between 29 and 40, middle-aged if it is between 41 and 55, and old if it is over 56.

The results of this improvised segregation are as follows:

```
young_ages = dataset[(dataset.Age>=29)&(dataset.Age<40)]  
middle_ages = dataset[(dataset.Age>=40)&(dataset.Age<55)]  
elderly_ages = dataset[(dataset.Age>=55)]  
  
print("Young Ages", len(young_ages))  
print("Middle Ages", len(middle_ages))  
print("Elderly Ages", len(elderly_ages))
```

```
Young Ages 194  
Middle Ages 153  
Elderly Ages 54
```

**Figure 6: Age segregation**



**Figure 7: Age pychart**

## Chapter 2: Machine Learning Model

In this section, we will be discussing the various learning algorithms that were utilized in our project. These algorithms include Logistic Regression, Logistic RegressionCV, Random Forest Classifier, and Gradient Boosting Classifier. Each algorithm was trained and evaluated using cross-validation techniques to assess their accuracy. The results showed that Logistic Regression had the highest accuracy of 76.95%, followed by Logistic RegressionCV with 76.56%, Random Forest Classifier with 75.52%, and Gradient Boosting Classifier with 75.91%.

### 2.1 Tested Learning Algorithms

Let's take a closer look at each of these algorithms and understand why they were chosen for this project.

#### 2.1.1 Logistic Regression

Logistic Regression is a popular linear classifier that is commonly used for binary classification problems. It makes predictions based on the input features by fitting a line to the data. In our project, the Logistic Regression algorithm was implemented using the scikit-learn library and its accuracy was evaluated using cross-validation techniques.

#### 2.1.2 Logistic RegressionCV

Logistic RegressionCV is an extension of the Logistic Regression algorithm that implements cross-validation for hyperparameter tuning. This allows for the algorithm to automatically select the best hyperparameters for the given data, leading to improved accuracy. The Logistic RegressionCV algorithm had an accuracy of 76.56% in our project.

### 2.1.3 Random Forest Classifier

Random Forest Classifier is an ensemble learning algorithm that builds multiple decision trees and combines their predictions to make a final prediction. This algorithm is known for its ability to handle non-linear relationships and its robustness to overfitting. The Random Forest Classifier had an accuracy of 75.52% in our project.

### 2.1.4 Gradient Boosting Classifier

Gradient Boosting Classifier is another ensemble learning algorithm that builds multiple weak models and combines them to make a stronger prediction. It is commonly used for complex classification problems and has been known to achieve state-of-the-art results on many datasets. The Gradient Boosting Classifier had an accuracy of 75.91% in our project.

**In conclusion**, the results from cross-validation showed that the Logistic Regression algorithm had the highest accuracy, followed by Logistic RegressionCV, Random Forest Classifier, and Gradient Boosting Classifier. Each of these algorithms was chosen for this project due to their effectiveness and ability to handle the complexities of the given data.

## 2.2 Used Learning Algorithms

After evaluating the accuracy of various learning algorithms in our project, it was determined that the Logistic Regression algorithm had the highest accuracy of 76.95%. As a result, the Logistic Regression algorithm was chosen for the final implementation of the project.

The reason for this choice lies in the simplicity and interpretability of the Logistic Regression algorithm. Logistic Regression provides a clear understanding of the relationships between the input features and the output predictions, making it easy to interpret the results and make decisions based on them. Additionally, Logistic Regression is fast and efficient, making it well suited for large datasets.

It is important to note that the accuracy of 76.95% does not guarantee that the algorithm will perform optimally for all future data. However, the results from cross-validation give us confidence that the Logistic Regression algorithm will perform well on the given data and will provide us with accurate predictions.

After making predictions. To measure the efficiency of our model, we can generate a report.

	precision	recall	f1-score	support
0	0.84	0.92	0.88	107
1	0.76	0.62	0.68	47
accuracy			0.82	154
macro avg	0.80	0.77	0.78	154
weighted avg	0.82	0.82	0.82	154

**Figure 8: Model scores**

Certainly! Precision, recall, F1-score, and support are commonly used metrics to evaluate the performance of a machine learning model.

**Precision:** refers to the number of true positive predictions out of all positive predictions made by the model. It measures the accuracy of the positive predictions made by the model.

**Recall:** refers to the number of true positive predictions out of all actual positive observations. It measures the ability of the model to correctly identify positive observations.

**F1 score:** is the harmonic mean of precision and recall. It provides a single value that balances both precision and recall, giving equal weight to both metrics.

**Support:** refers to the number of observations in each class. It gives an understanding of the class distribution in the dataset and can be useful for evaluating the performance of the model on class imbalance problems.

## Chapter 3: Web Application

The web application was built using the Django framework, a high-level Python web framework that enables the rapid development of secure and maintainable web applications & pickle for storing our machine-learning model.

### 3.1 Tools & Technologies

#### 3.1.1 Django

Django is a high-level Python web framework that enables the rapid development of secure and maintainable web applications. In the context of our web application, Django was used to build the user interface, handle user inputs, and display the prediction results.



The following are the key components and features of Django in our web application implementation:

1. URL Routing: Django provides an easy-to-use URL routing system that maps URLs to views. In our web application, this was used to route user requests to the appropriate pages, such as the user input form and the prediction results page.
2. Templates: Django provides a template engine that enables the separation of presentation logic from business logic. In our web application, this was used to create a user-friendly and attractive interface for the user input form and prediction results.



3. **Forms:** Django provides a form framework that makes it easy to handle user inputs. In our web application, this was used to create a form for the user to input their data, and to validate the inputs to ensure they are in the correct format.
4. **Views:** Django provides a view framework that handles user requests and returns the appropriate response. In our web application, this was used to handle user inputs, make predictions using the machine learning model, and return the prediction results to the user.
5. **Models:** Django provides a model framework that enables the representation of data as Python objects. In our web application, this was not used as the machine learning model was saved as a pickle file and integrated into the web application using the pickle module.

The use of Django in our web application provided several benefits, including:

1. **Rapid Development:** Django allowed us to rapidly develop the web application, as it provided a comprehensive and well-documented framework for building web applications.
2. **Scalability:** Django is designed to be scalable, making it easy to add new features and functionality as the web application grows.
3. **Security:** Django provides several built-in security features, such as protection against cross-site scripting (XSS) and cross-site request forgery (CSRF) attacks, which ensured that the web application was secure.

In conclusion, the use of Django in our web application was a key component in building a secure, scalable, and user-friendly web application. By leveraging the features and functionality provided by Django, we were able to rapidly develop a high-quality web application that met the needs of our users.

### 3.1.2 Pickle

Pickle is a Python module that is used to serialize and deserialize Python objects. It enables the saving of complex Python objects (such as machine learning models) to disk and reloading them in the same or another Python environment. In the context of our web application, the machine learning model was saved as a pickle file, which was then integrated into the Django application to make predictions directly from the web application.



The following are the steps involved in using pickle with our web application:

1. **Model Serialization:** The machine learning model was trained and then serialized using the pickle module. This was done using the **pickle.dump()** function, which writes a pickled representation of an object to a file.
2. **Model Integration:** The pickle file was integrated into the Django application by adding it to the application's file system and loading it into memory when the application starts. This was done using the **pickle.load()** function, which reads a pickled object representation from a file and converts it back into a Python object.
3. **Model Prediction:** The integrated machine learning model was used to make predictions based on the user input data. This was done by calling the appropriate methods and passing the user input data as arguments. The predictions were then returned to the user via the web application's user interface.

The use of pickle in our web application provided several benefits, including:

1. **Simplified Model Deployment:** Pickle made it easy to deploy the machine learning model within the web application, as it eliminated the need to write complex code to load and use the model.
2. **Improved Performance:** By integrating the model into the web application, predictions could be made directly from the web application, rather than having to call an external model. This improved the overall performance of the web application.
3. **Portable Models:** The use of pickle allowed the machine learning model to be portable, as it could be easily moved from one environment to another.

In conclusion, the use of pickle in our web application was a key component in enabling the efficient deployment and use of the machine learning model. By serializing and integrating the model, we were able to provide users with a fast and reliable way to check if they have diabetes directly from the web application.

### 3.3 Implementation

The following are the key components and features of the web application implementation:

**User Input Form:** A user-friendly form was created to allow users to input their data (such as age, BMI, blood pressure, etc.). The form was designed to ensure that the user inputs the required information and in the correct format.

**Pickle File:** The machine learning model was saved as a pickle file, which was then used to make predictions based on the user input data. The pickle file was integrated into the Django application to allow the model to make predictions directly from the web application.

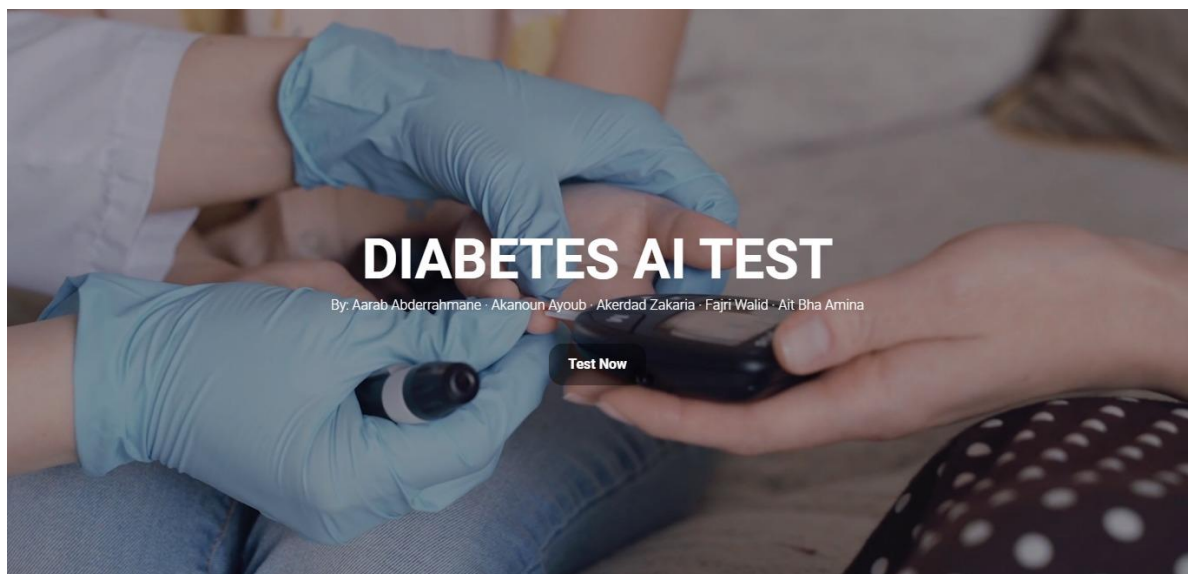
**Prediction Results:** The results of the predictions were displayed to the user in a clear and concise manner. A conditional formatting approach was used to highlight the results, making it easier for users to interpret the output.

Overall, the web application provides a convenient and secure way for users to check if they have diabetes. The use of Django and pickle, combined with a user-friendly interface, makes it easy for users to obtain reliable and accurate results quickly.

## 3.4 Web App Preview

### 3.4.1 Home Page

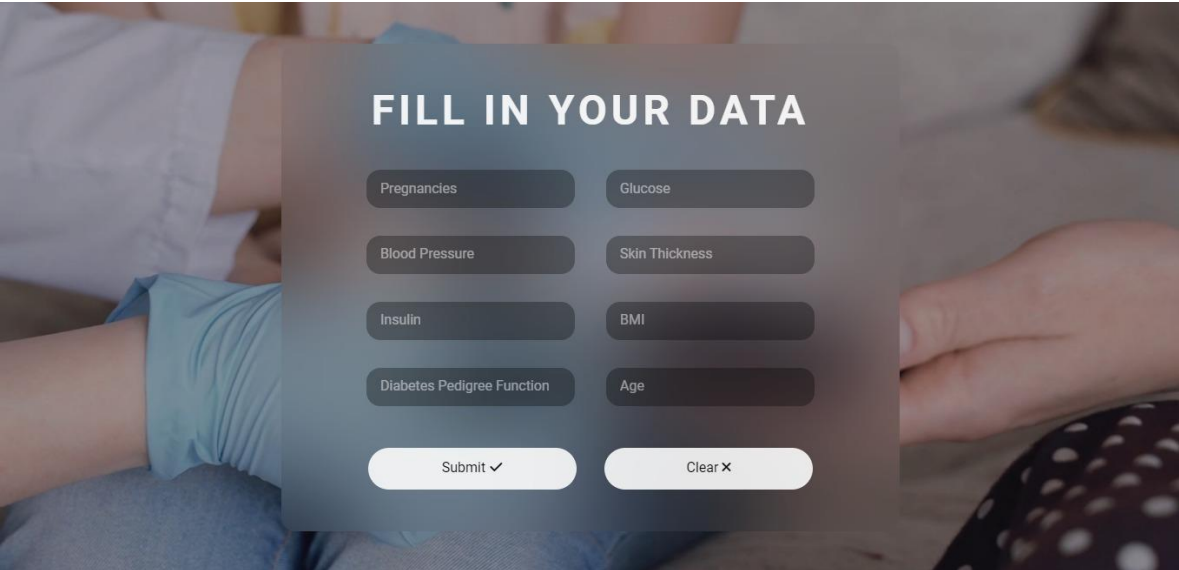
The home page is the starting point for a web application or website. It typically contains a brief introduction to the site or application and links to other pages within the site. In this particular case, there is a "test button" on the home page which the user can click to proceed to the form page.



**Figure 9: Home page**

### 3.4.2 Form Page

The form page is a page where the user is prompted to fill out data. In this specific scenario, the form page is used to gather information needed to determine if the user is diabetic or not. The user fills out the form by providing information such as their age, number of pregnancies, BMI, glucose, etc.



**FILL IN YOUR DATA**

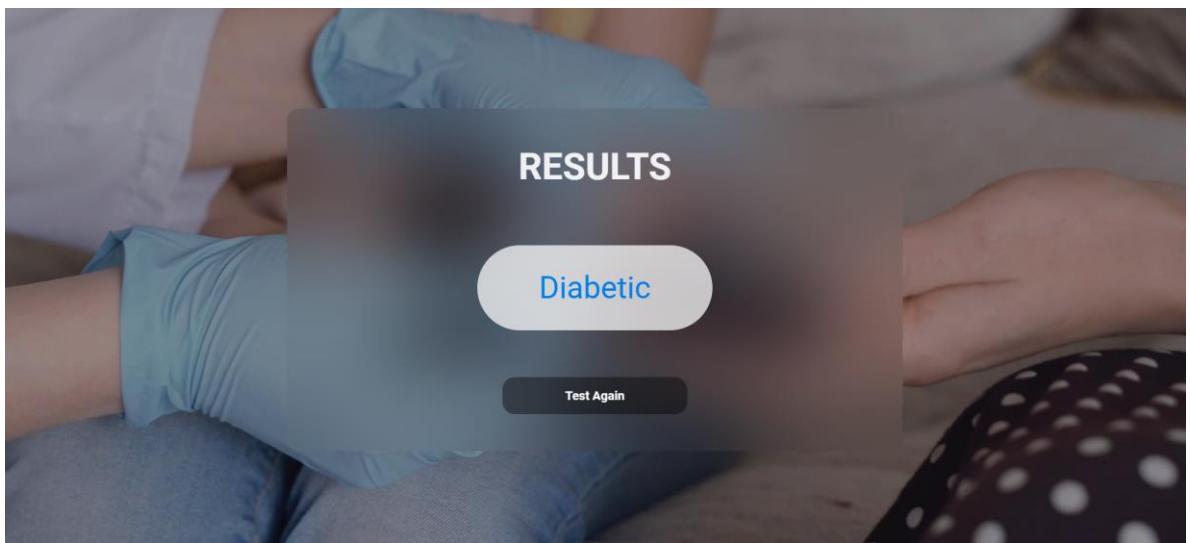
Pregnancies	Glucose
Blood Pressure	Skin Thickness
Insulin	BMI
Diabetes Pedigree Function	Age

Submit ✓ Clear ✕

**Figure 10: Form page**

### 3.4.3 Results Page

The results page is the final step of the process, where the user is shown if they are diabetic or not. The results are determined based on the information provided by the user on the form page, and the results page displays the outcome of the test in a clear and concise manner.



**Figure 11: Results page**

## Conclusion

In conclusion, machine learning has the potential to transform healthcare and improve patient outcomes. Its ability to analyze large amounts of data can help healthcare professionals make better decisions and provide more personalized care. However, it's important to remember that machine learning is still in its early stages and that more research and development are required to fully realize its potential. Furthermore, ethical issues such as data privacy and bias must be addressed to ensure that these technologies are used responsibly in healthcare. Overall, machine learning has the potential to completely transform the healthcare industry, and it will be fascinating to see how it evolves and impacts patient care in the coming years.