***A logo with blue text

AI-generated content may be incorrect.***

**Egyptian E-Learning University**

Faculty of Computers & Information Technology

**DIABETES PREDICTION AND MANAGEMNT**

**By:**

|  |  |
| --- | --- |
| Name: Antony Ehab Shehata | ID: 21-00511 |
| Name: Aya Ayman Esmat | ID: 21-00539 |
| Name: Beshoy Waseem Moneir | ID: 21-00577 |
| Name: Nouran SafaaEldein Anwar | ID: 21-01362 |
| Name: Youstena Emad Gaber | ID: 21-01760 |
| Name: Youssef Mamdouh Makram | ID: 21-01736 |
| Name: Ziad Moustafa Thabet | ID: 21-01788 |

***Supervised by***

**Dr. Ahmed Hamza**

**Assistant**

**Eng. Safinaz AalaaElDin**

[Assuit]-2025

**Acknowledgement**

First and foremost, we express our deepest gratitude to **God Almighty** for granting us the perseverance and strength to reach this milestone in our education.

This project is the culmination of our hard work and dedication. However, it would not have been possible without the invaluable support and encouragement of many individuals and institutions.

We extend our heartfelt thanks to the **Egyptian E-Learning University**, and especially to the **Asyut Center**, for providing us with the resources and guidance needed to enhance our awareness and knowledge throughout this project.

Our sincere appreciation goes to our Project Supervisor, **Dr. Ahmed Hamza**, for his invaluable guidance and mentorship. It has been a great privilege and honor to work under his supervision.

And many thanks to **Eng. Safinaz AlaaEldin ,** We are truly grateful for her guidance, constant support, and patience.

Finally, we wish to express our gratitude to everyone who offered their help and support during this graduation project.

**Abstract**

This project presents an interactive, web-based system for diabetes risk assessment and health evaluation. Leveraging **machine learning (ML)** models and the **Django web framework**, the system aims to provide users with accurate, real-time predictions about their diabetes risk and type classification, along with helpful health tools.

Key objectives included:  
1.Developing robust ML models to predict diabetes presence and classify diabetes type.

2.Creating an accessible and intuitive web interface.

3.Integrating additional features like BMI calculation, glucose level evaluation, and dietary guidance for comprehensive health support.

The system’s **modular architecture** ensures scalability and ease of maintenance. While not a substitute for medical diagnosis, it provides valuable insights and promotes proactive health management

**Table of Contents**

*Chapter 1*

*Introduction*

* 1. Introduction
  2. History
  3. Background and Motivation
  4. Importance of the problem being addressed
  5. Problem Statement
  6. Problem Solution
  7. Objectives
  8. Brief overview of the proposed solution
  9. Project phases

*Chapter 2*

*Literature Reviews/ Related Work*

2.1 Summary of Existing research and technologies

2.2 Gaps in current solutions

*Chapter 3*

*Proposed Systems*

3.1 Approaches used to solve the problem

3.2 System architecture

3.3 website view

*Chapter 4*

*Implementation*

4.1 Frontend

4.2 Backend

4.3 Machine learning

*Chapter 5*

*Testing and Evaluation*

5.1 Testing Stratiges

5.2 Performance metrics

*Chapter 6*

*Results and Discussion*

6.1 Introduction

6.2 Summary of findings

6.3 Interpretation of results(Meeting projects goals)

6.4 Limitations of proposed solutions

*Chapter 7*

*Conclusion and Future Work*

7.1 Summary of contributions

7.2 Possible of improvements or extensions for future work

*Chapter 8*

*References*

**Chapter 1**

**Introduction**

**1.1. Introduction**

The increasing of diabetes globally has highlighted the need for accessible, reliable, and efficient tools for early detection and monitoring.

By leveraging modern web technologies and machine learning techniques, we aim to build an integrated system that empowers users to proactively assess their diabetes status and take preventive measures. This project also serves as a practical application of our knowledge in web development, machine learning.

Diabetes mellitus is a chronic metabolic disorder that affects millions of people worldwide. Early detection and proper classification of diabetes types are crucial for effective treatment and lifestyle management. To address this challenge, our project integrates machine learning and web technologies to provide a user-friendly platform for diabetes prediction, diabetes type classification, BMI calculation, and glucose level evaluation.

This web-based system utilizes machine learning algorithms trained on real-world datasets to analyze user-provided data and deliver accurate predictions. The user interface is designed with simplicity in mind, enabling users to seamlessly interact with the system and receive immediate feedback on their health metrics.

**1.2. History**

Diabetes is a chronic metabolic disorder that has affected humanity for centuries. Its history can be traced back to ancient times, with the first descriptions appearing in ancient Egyptian, Indian, and Greek medical texts. The term “diabetes” was coined by the Greek physician Aretaeus of Cappadocia in the first century AD, describing the excessive urination associated with the condition.

Despite this early recognition, diabetes remained poorly understood for many centuries. It wasn’t until the late 19th and early 20th centuries that significant progress was made in understanding its causes and developing treatments. The discovery of insulin in 1921 by Frederick Banting and Charles Best revolutionized diabetes treatment, transforming it from a fatal disease to a manageable chronic condition.

With the advancement of technology and medical knowledge, diabetes care has evolved dramatically. Modern blood glucose monitoring devices and laboratory tests have become essential tools in diabetes management. However, despite these advancements, many patients still struggle with interpreting blood glucose data and managing their weight, which are crucial factors in preventing complications.

In recent years, the integration of artificial intelligence (AI) and machine learning (ML) into healthcare has opened new avenues for improving diabetes care. Machine learning models can analyze large amounts of data to predict diabetes risk and help patients and healthcare providers make more informed decisions.

**1.3. Background and Motivation**

Diabetes is a major global health issue, affecting millions of people worldwide. The condition is characterized by high blood glucose levels resulting from either insufficient insulin production (Type 1 diabetes) or insulin resistance (Type 2 diabetes). Poorly managed diabetes can lead to serious health complications, including heart disease, nerve damage, kidney failure, and vision problems.

Despite the availability of glucose monitoring devices and modern treatments, many patients face difficulties in managing their condition. They often struggle to interpret blood glucose readings, which are typically displayed as numerical values without clear indications of whether they are high, normal, or low. Additionally, many patients are unaware of the early symptoms of diabetes, which can delay diagnosis and treatment. The lack of accessible tools to assist with weight management and dietary planning further complicates effective diabetes management.

Our project was motivated by the need to address these challenges using modern technologies such as web development and machine learning. By creating an interactive, user-friendly web platform, we aim to make diabetes care more accessible and understandable for patients. Our goals include:

1. **Improving Early Detection:** By developing a prediction tool that allows users to input their symptoms and receive an assessment of their diabetes risk, we hope to encourage earlier diagnosis and treatment.
2. **Simplifying Blood Glucose Interpretation:** With tools that clearly explain whether blood glucose readings are high, normal, or low, users can better manage their condition and reduce their risk of complications.
3. **Supporting Weight and Lifestyle Management:** Through a BMI calculator , we seek to help patients maintain a healthy weight, which is critical for diabetes prevention and control.
4. **Leveraging Technology for Better Care:** By integrating machine learning models into our web platform, we aim to harness the power of data analysis to deliver accurate, personalized, and actionable insights.

Ultimately, our project is driven by the desire to empower diabetic patients and those at risk of diabetes, using technology to make health management simpler and more effective.

**1.4. Importance of the Problem Being Addressed**

Diabetes is a growing global health crisis, with the number of people affected increasing year after year. It is a leading cause of serious health complications, such as heart disease, nerve damage, kidney failure, and vision problems. Beyond its physical toll, diabetes also has significant social and economic impacts, placing a heavy burden on healthcare systems and families.

Despite the availability of blood glucose monitoring devices and other medical technologies, many patients still face significant challenges in managing their condition. The difficulty in understanding blood glucose readings, lack of awareness of early symptoms, and limited access to tailored dietary advice and weight management support all contribute to poor diabetes management and increased risk of complications.

Early detection and effective management of diabetes are crucial in reducing these health risks and improving patients’ quality of life. However, many individuals remain unaware of their risk of developing diabetes or how to properly interpret the information from their monitoring devices. In addition, lifestyle factors, such as obesity and poor diet, play a significant role in the development and progression of diabetes.

By addressing these challenges through the development of a comprehensive web platform that integrates prediction tools, blood glucose evaluation, BMI calculation, and dietary guidance, this project aims to bridge the gap between technology and patient care. It empowers patients to take control of their health, promotes early diagnosis and treatment, and encourages healthier lifestyle choices.

The importance of this problem lies not only in the health outcomes of individuals but also in the broader impact on society. By improving diabetes awareness and management, we can reduce the overall burden of the disease and help people live healthier, longer lives.

**1.5. Problem statement**

1

One of the major challenges in managing diabetes is the delayed diagnosis of the disease.

diabetes is often diagnosed at a late stage, which can delay effective treatment and management. Delayed diagnosis increases the risk of serious complications such as cardiovascular diseases, nerve damage, kidney failure, and vision problems.

Early detection and management are crucial to preventing these complications and improving patients’ quality of life.

Addressing these issues is essential to improving diabetes care and reducing the burden of the disease.

2

Many diabetic patients are often unaware of their actual blood glucose levels and whether these levels are normal, high, or low. This lack of understanding can lead to mismanagement of their condition and poor glycemic control.

Furthermore, most blood glucose monitoring devices currently available do not clearly indicate whether the measurement result is considered low, normal, or high. Instead, they only provide numerical values, which can be confusing and hard to interpret for some patients.

3

Another critical issue is the lack of attention to weight management.

Obesity and overweight are major risk factors for developing diabetes and its complications.

Unfortunately, many people neglect the importance of maintaining a healthy weight, which leads to increased health risks and the development of various diseases, including diabetes.

**1.6. Problem solution**

To address these challenges, we are developing a comprehensive web application that supports diabetic patients and those at risk of diabetes. Our platform provides the following features:

1. **Diabetes Prediction Test**

The website includes an interactive test that allows users to select the symptoms they are experiencing. Based on these inputs, the test predicts whether the symptoms might indicate diabetes or not using machine learning , helping users become more aware of their health status and encouraging early diagnosis.

1. **Blood Glucose Evaluation**

The platform helps users interpret their blood glucose readings by telling them whether their measurement is high, normal, or low. This evaluation is done for various situations, including two hours after eating, fasting blood sugar, and HbA1c levels. This feature addresses the confusion many patients face when interpreting numerical glucose values, thus promoting better glycemic control.

1. **BMI Calculator**

The web application also includes a Body Mass Index (BMI) calculator, which allows users to assess whether they are underweight, normal weight, overweight, or obese.

Since maintaining a healthy weight is crucial for preventing and managing diabetes, this tool helps users track their weight status and make informed decisions about their lifestyle.

1. **Dietary Support and Food Recommendations**

The website provides dietary assistance tailored to the type of diabetes (Type 1 or Type 2). It offers information about foods to avoid and suggests healthy recipes that suit each diabetes type. This helps users make better food choices and manage their condition more effectively.

By integrating these features, the website aims to enhance diabetes awareness, encourage early detection, support better blood glucose management, and promote a healthy lifestyle for diabetic patients.

**1.7. Objectives**

**Main Objective:**  
The main objective of this project is to develop an accessible and intelligent web-based tool that assists diabetic patients and those at risk of diabetes in better understanding and managing their health. By leveraging modern technologies such as machine learning and user-friendly interfaces, the project aims to create a comprehensive platform that empowers users to make informed decisions about their health and well-being.

Specific Objectives:

1. **Design a seamless web interface:** Develop an intuitive and easy-to-navigate website that ensures a positive user experience for people of different ages and technical backgrounds.

2-**Integrate machine learning models:** That analyze symptoms and provide assessments to help users recognize potential diabetes risks.

3-**Implement interactive tools:** Create user-friendly calculators and evaluation modules for blood glucose interpretation and BMI assessment.

4-**Develop personalized dietary features:** Provide tailored food recommendations and dietary tips that address different types of diabetes and individual health needs.

6-**Facilitate continuous updates and improvements:** Build a system architecture that allows for future expansion and refinement based on user feedback and evolving medical standards.

These objectives collectively support the project’s main goal: to bridge the gap between health information and patient understanding, ultimately contributing to improved diabetes management and prevention.

**1.8. Brief Overview of the Proposed Solution**

The proposed solution is a web-based platform designed to support diabetic patients and individuals at risk of diabetes. This platform combines modern web development and machine learning techniques to deliver a comprehensive set of tools for diabetes awareness, prevention, and management.

The platform will include an interactive diabetes prediction test, enabling users to assess their risk based on symptoms they experience. It will also feature a blood glucose evaluation tool that interprets blood glucose readings—clarifying whether measurements are low, normal, or high in different contexts, such as fasting, post-meal, or HbA1c levels.

Additionally, the platform offers a BMI calculator to help users assess their weight status, encouraging healthy weight management—a crucial factor in diabetes prevention and control. To further support lifestyle management, the platform will provide tailored dietary advice and recipes specific to each diabetes type.

By integrating these features into a single, user-friendly website, the solution aims to simplify diabetes management, improve patient understanding of their health data, and promote early detection and lifestyle adjustments. The platform’s design prioritizes ease of use, security, and future adaptability to ensure it remains a valuable tool for patients and caregivers alike.

**1.9. Project Phases**

The project was implemented in several phases:

1-**Research and Planning:** Gathering requirements, understanding existing solutions, and outlining the system architecture.

2-**Data Collection and Preprocessing:** Preparing datasets for machine learning model training and validation.

3-**Model Development:** Implementing machine learning algorithms to predict diabetes and classify diabetes types.

4-**Web Application Development:** Building the Django-based web application with pages for each feature.

5-**Integration:** Connecting the machine learning models to the web interface to provide real-time predictions.

6-**Testing and Validation:** Conducting extensive testing to ensure accuracy and usability.

**Chapter 2**

**Literature Review / Related Work**

1. **Summary of Existing Research and Technologies**

**1.Diabetes Risk Calculators**

Common risk calculators use **fixed rules** based on clinical thresholds (age, BMI, family history).

While easy to use, they lack the flexibility of ML models to **adapt to complex patterns** in health data.

2.Many **blood glucose meters** only display the raw measurement number (e.g., 120 mg/dL).

**2.2. Gaps in Current Solutions**

**1.Limited Flexibility ,**  **No Personalization**

While diabetes risk calculators based on clinical thresholds are easy to use, they rely on **fixed rules**.

They cannot **adapt to more complex interactions** or consider less obvious patterns in a user’s health data.

These calculators **do not account for individual variations** beyond a few risk factors (e.g., family history, BMI).

As a result, they may not capture more subtle signs of early diabetes risk.

**2.Lack of Interpretation in Glucose Measurement Devices**

They **do not provide context** or classify the result as “normal,” “low,” or “high” for the user.

**Chapter 3**

**Proposed system**

#### ****3.1 Approach Used to Solve the Problem****

The project was approached with a **multi-layered methodology** that combines:  
 **Machine Learning (ML)** for diabetes risk and type prediction,  
 **Web Development** for a user-friendly interface,  
 **Health education tools** like the BMI calculator, glucose evaluator, and nutrition page.

The goal was to create a system that is:

**Accurate** in its predictions,

**Interactive** for users,

**Modular and scalable** for future enhancements.

The development followed these steps:  
 Data collection and preparation.  
 Model training and evaluation (testing multiple algorithms).  
 Building the Django web application.  
 Integrating ML models with the web backend.  
 Designing user-friendly frontend interfaces.  
 Testing and validating the system.

#### ****3.2 System Architecture****

The system follows a **three-tier architecture** typical of modern web applications:

**Presentation Layer (Frontend)**

HTML, CSS, JavaScript for user interfaces.

Forms for data input (age, weight, symptoms, etc.).

Result pages and visualization.

**Backend**

Django framework for request handling and rendering templates.

ML model integration using joblib to load pre-trained models and make predictions.

Data preprocessing and validation.

**Data Layer**

Pre-trained ML models saved as .pkl files.

This modular flow ensures **separation of concerns** and **easy maintenance**.

#### ****Algorithms or Frameworks Used****

**Frameworks and Libraries**

**Django**: Python-based web framework for backend development.

**scikit-learn**: Machine learning library for training, evaluating, and using ML models.

**joblib**: For saving and loading ML models efficiently.

**pandas** and **numpy**: For data manipulation and numerical operations.

**Machine Learning Algorithms**

Multiple algorithms were tested:

**Logistic Regression**

**Decision Tree Classifier**

**Support Vector Machine (SVM)**

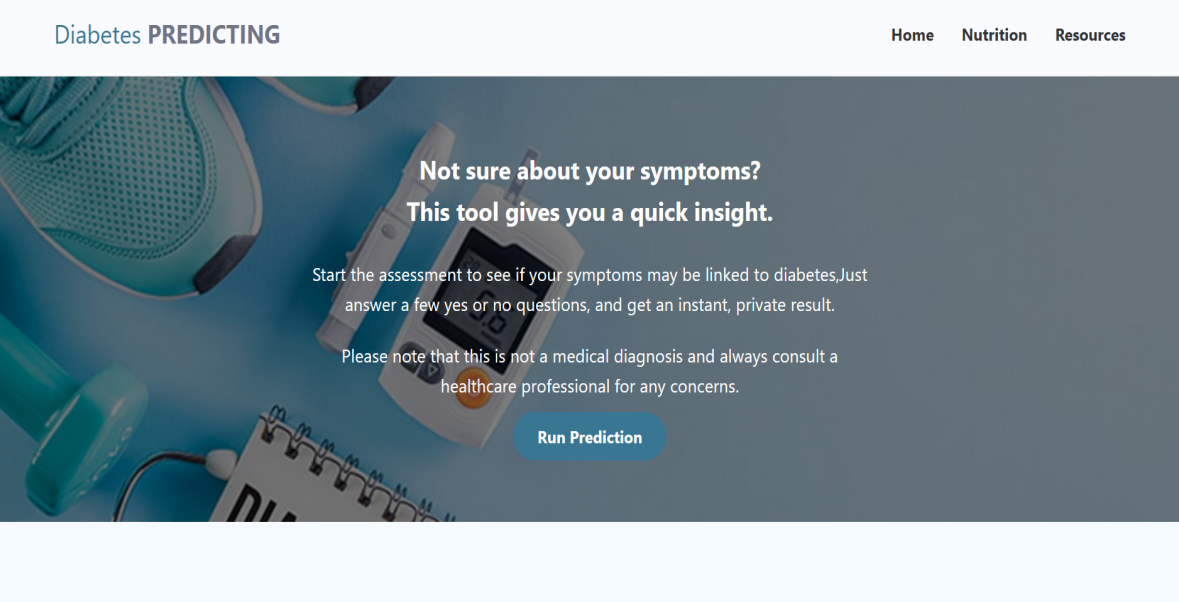
**Naive Bayes (BernoulliNB)**

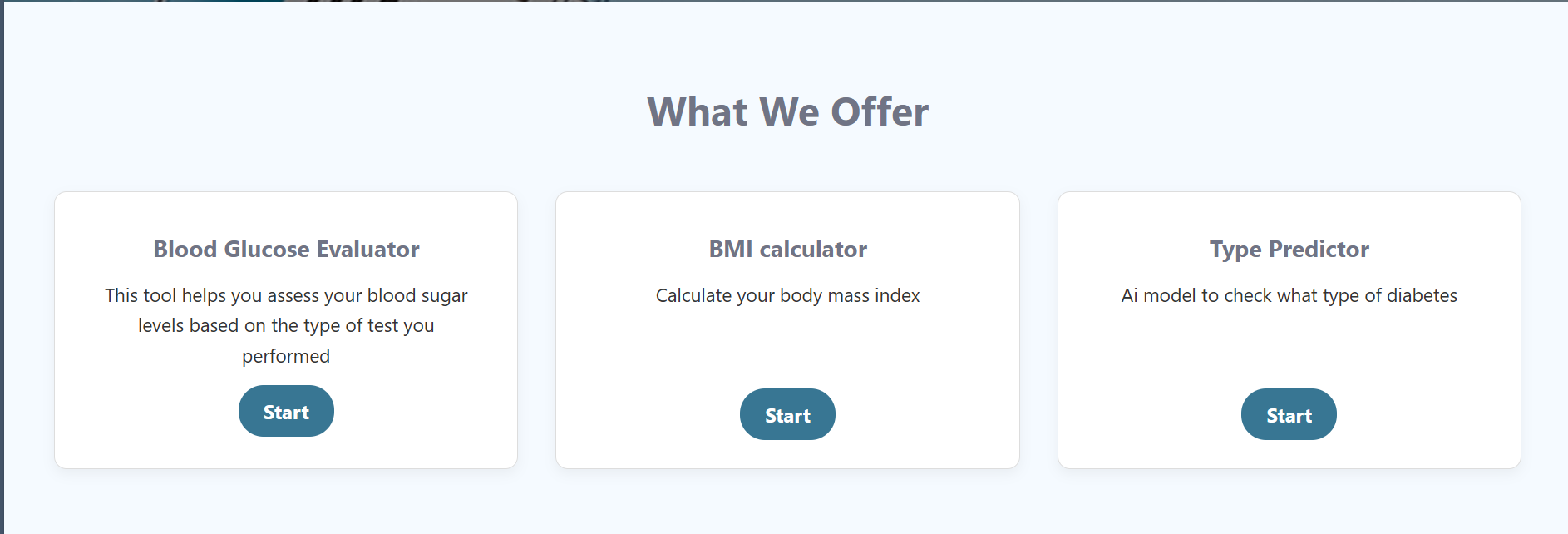
**Random Forest Classifier**

**Final Model**:  
The **Random Forest Classifier** was selected due to its **highest accuracy** and robustness. It uses an ensemble of decision trees to make predictions, reducing overfitting and improving stability.

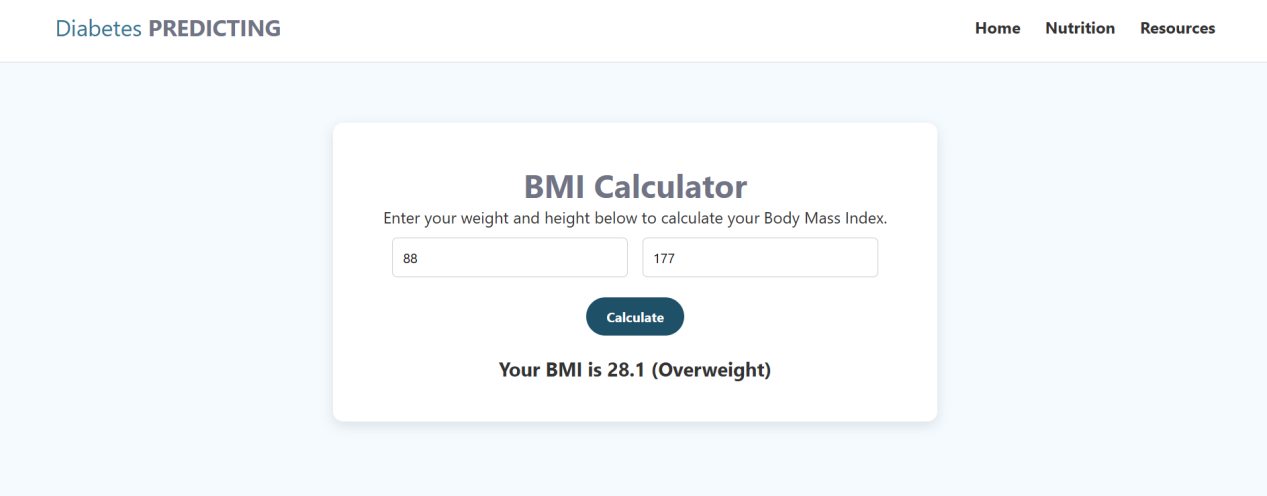
**3.3 Website view**:

**Home page**

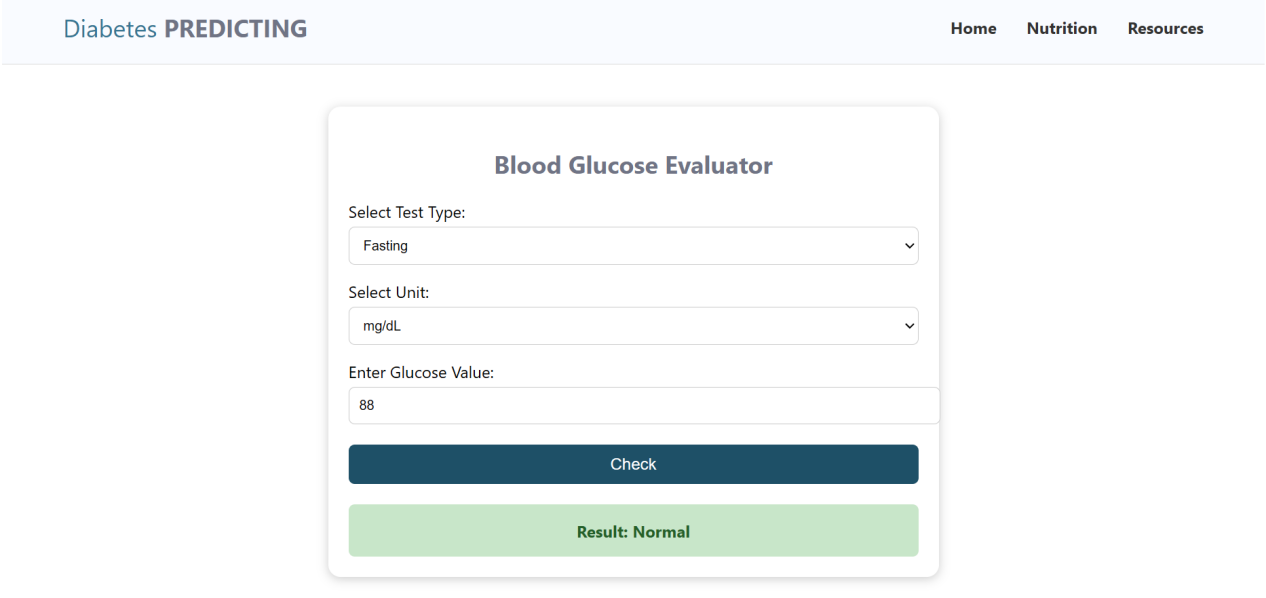


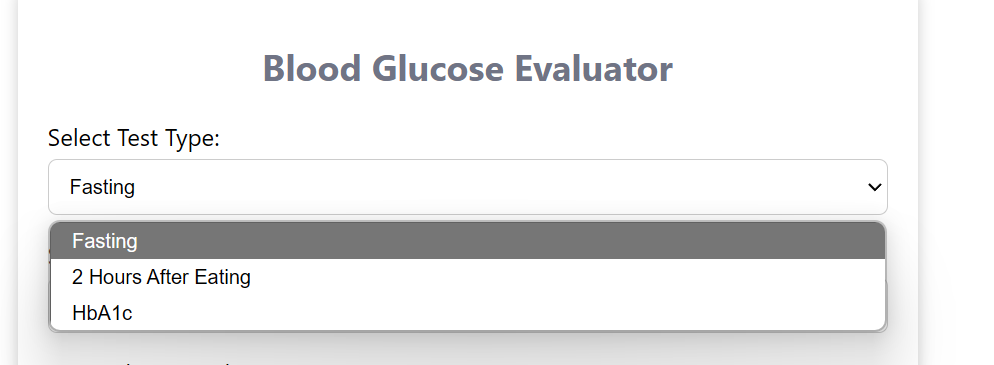


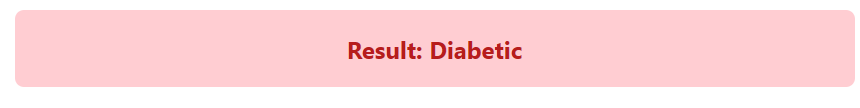
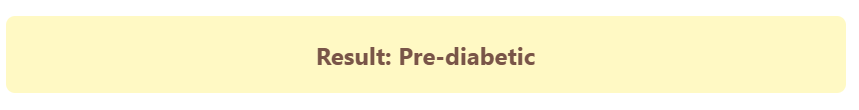
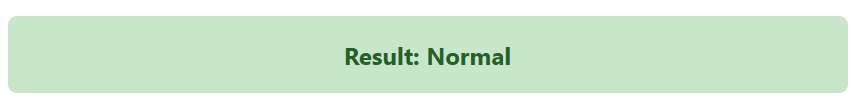
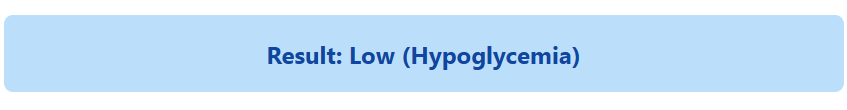
**BMI calculator page**

****

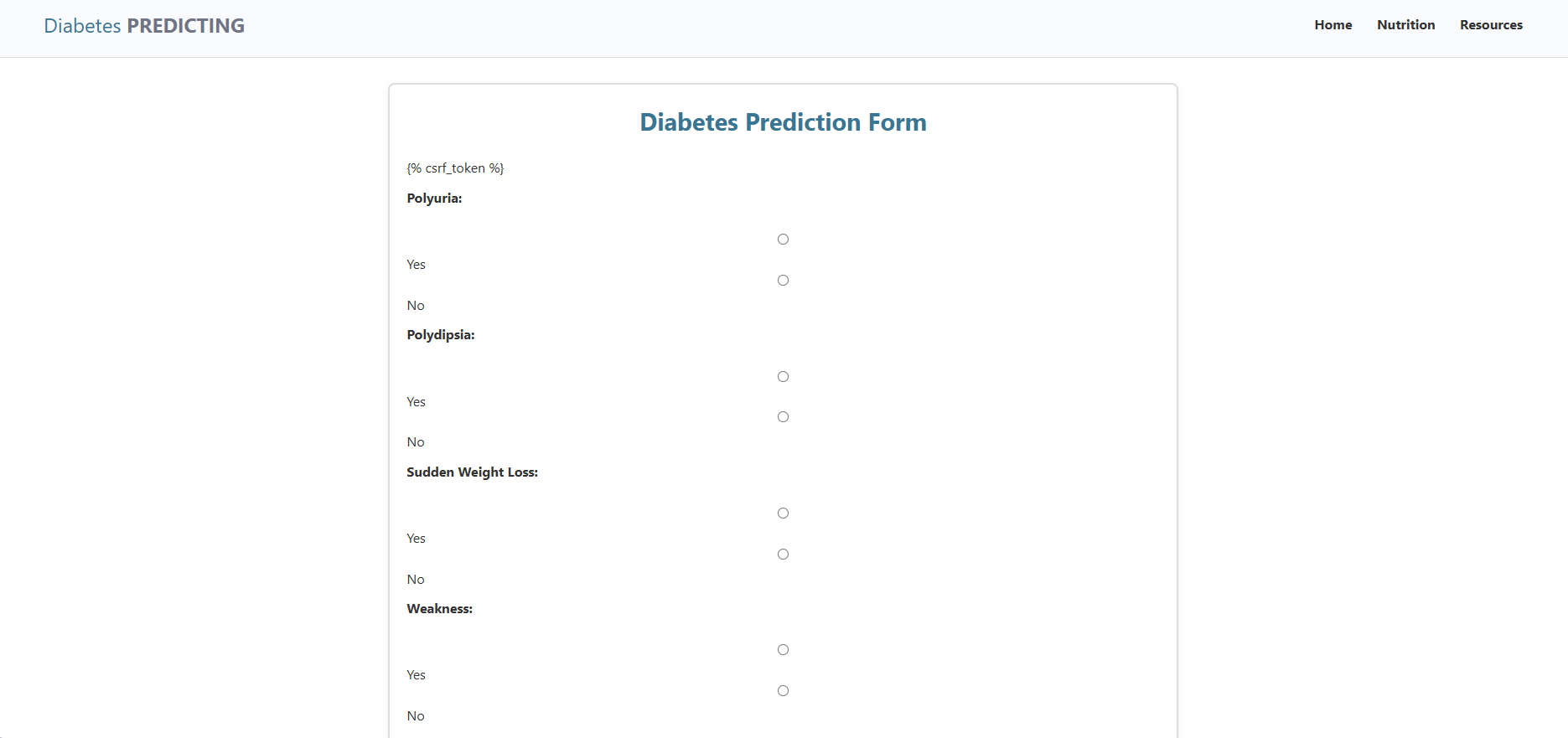
## Blood Glucose Evaluator page

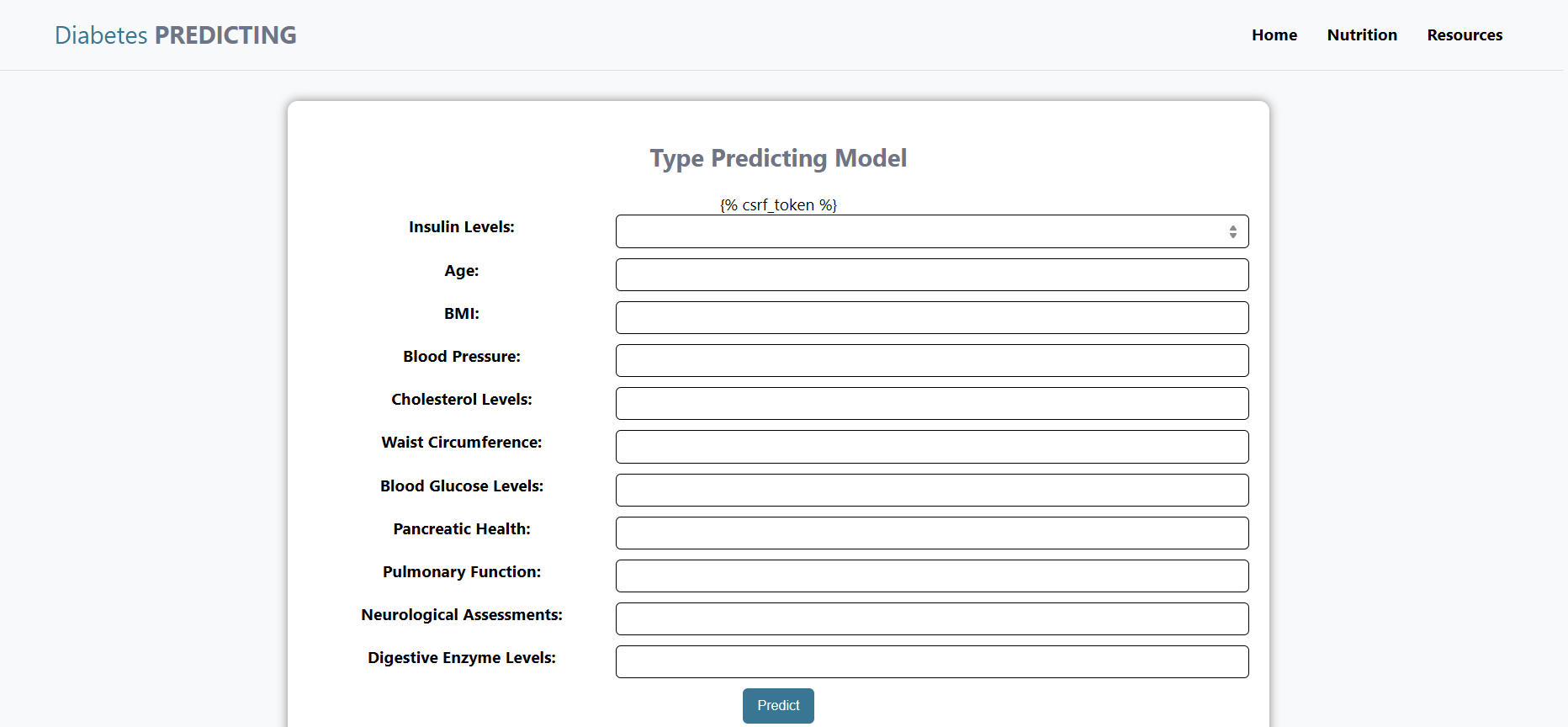
****

****

****

****

****

****

**Chapter 4**

**Implementation**

**4.1 Frontend**

we use Visual Studio Code:

Visual Studio Code is a script editor from Microsoft. The editor is open source and works on Windows, Mac and Linux operating systems and was launched by Microsoft in April 2015. Through it, we can write (Html , Css and js ).

1. Html :

HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. Here are some key reasons why HTML is essential:

1. Structure and Organization

Content Structuring: HTML provides a way to structure web content using elements such as headings, paragraphs, lists, links, and other forms of media. This makes the content readable and accessible.

Semantic Markup: HTML5 introduced semantic elements like <header>, <footer>, <article>, <section>, and <nav>, which provide meaning to the web content. This helps search engines and other technologies understand the context and structure of the content.

1. Web Page Creation

Foundation of Web Pages: HTML is the backbone of all web pages. It defines the basic layout and elements that make up the page.

Compatibility: HTML is supported by all web browsers, ensuring that content can be displayed consistently across different platforms and devices.

1. Integration with Other Technologies

CSS Integration: HTML works seamlessly with CSS (Cascading Style Sheets) to style and layout web pages. While HTML provides the structure, CSS enhances the visual presentation.

JavaScript Integration: HTML provides the foundation for incorporating JavaScript, which adds interactivity and dynamic behavior to web pages.

4. Accessibility

Accessibility Features: HTML includes features that enhance accessibility for users with disabilities. For example, the <alt> attribute in images provides alternative text for screen readers.

Responsive Design: HTML, in conjunction with CSS, supports responsive design principles, allowing web pages to adapt to different screen sizes and devices.

5.Standardization and Interoperability

Web Standards: HTML is maintained by the World Wide Web Consortium (W3C) and WHATWG, ensuring that it adheres to web standards that promote consistency and interoperability across different browsers and devices.

Future-Proof: HTML continues to evolve, with new versions and features being developed to meet the growing needs of the web.

1. Ease of Learning and Use

Simplicity: HTML is relatively easy to learn and use, making it accessible for beginners. Its straightforward syntax allows new developers to start building web pages quickly.

Extensive Documentation: There is extensive documentation and a wealth of resources available for learning HTML, making it easier for developers to find help and improve their skills.

2- Css language:

We use CSS (Cascading Style Sheets) in our project( interior design) that can significantly enhance the presentation and interactivity of your design, especially if you are presenting your project through a web-based platform. Here are several reasons why incorporating CSS is beneficial:

1. Visual Consistency

Design Consistency: CSS ensures that your design elements maintain consistent styling across different web pages and devices. This includes fonts, colors, margins, and spacing, which helps in creating a cohesive look and feel for your project presentation.

1. Responsive Design

Device Compatibility: CSS allows you to create responsive designs that adapt to various screen sizes and resolutions. This ensures that your interior design project looks great on desktops, tablets, and smartphones.

Media Queries: You can use media queries to apply different styles for different devices, enhancing the user experience across platforms.

1. Enhanced Visual Appeal

Styling Elements: CSS provides the ability to style HTML elements with great detail, such as applying gradients, shadows, and transitions, which can make your presentation visually appealing

Animations: CSS animations can bring your project to life by adding dynamic effects to elements, such as hover effects, slideshows, and fade-ins, making your presentation more engaging.

4. Customization and Flexibility

Custom Styles: CSS allows you to customize the appearance of your project to match your branding or the aesthetic of your interior design.

Flexibility: It provides flexibility to change the design quickly by modifying the CSS file without altering the HTML structure, which is efficient for updates and revisions.

5. Separation of Content and Style

Maintainability: By separating the content (HTML) from the styling (CSS), you

make your project easier to maintain and update. This separation allows for cleaner

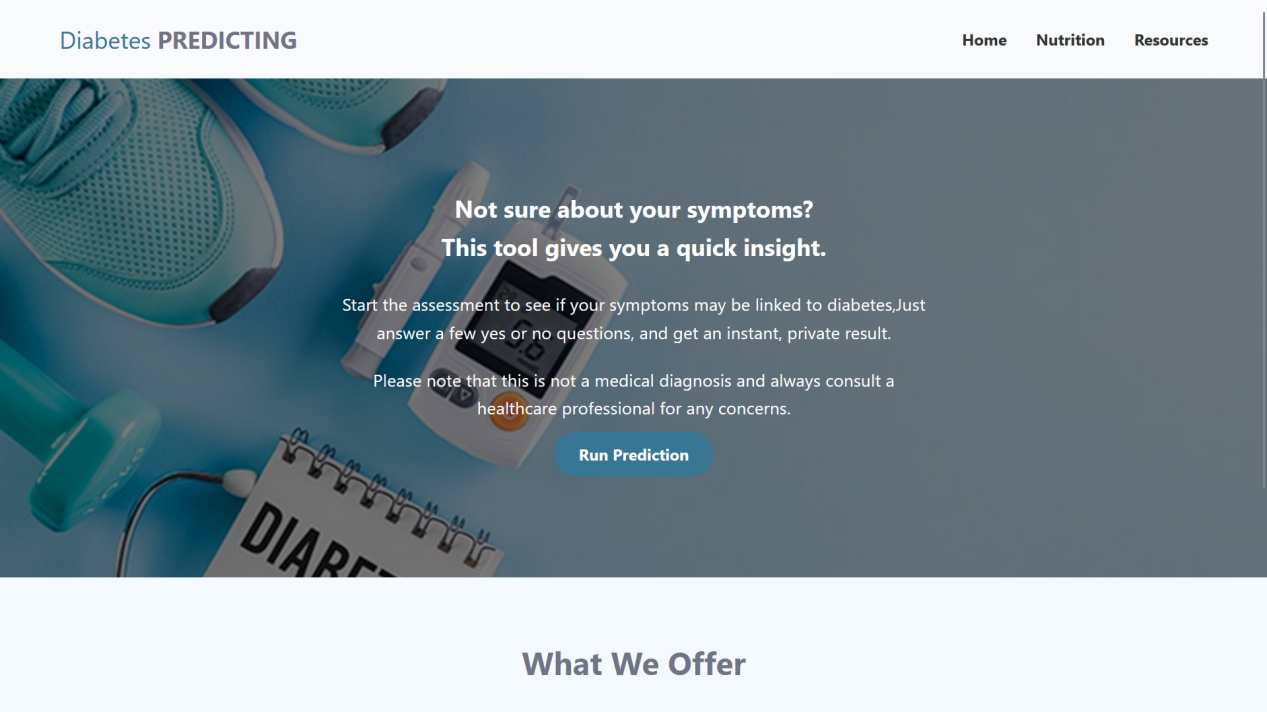
and more organized code.

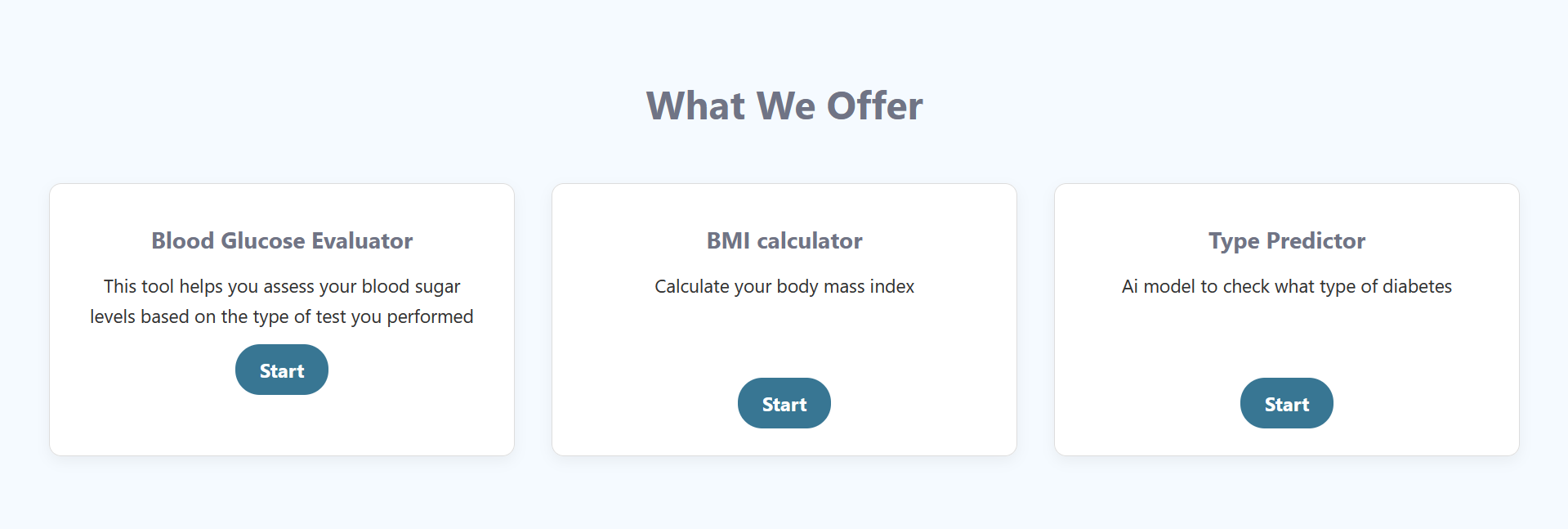
Reusability: You can reuse the same CSS styles across multiple pages or projects, saving time and ensuring consistency.

3- JavaScript (JS) language :

JavaScript (JS) can play a crucial role in enhancing of our project , particularly when it involves web-based presentations, interactive features, and real time visualizations.

**Home Page**



****

The **Home Page** serves as the entry point to the entire system. It features a visually appealing design with a **section** that introduces the purpose of the tool and encourages users to start their assessment. This section includes:

A **title** highlighting the purpose: helping users evaluate their symptoms.

A **call-to-action** button labeled “Run Prediction” for quick access to the diabetes risk assessment tool.

**Informational text** reassuring users that this tool is not a substitute for professional medical diagnosis.

Below the first section, there is a **“What We Offer”** section. This section introduces the three main tools available in the system:

**Blood Glucose Evaluator**: A tool to evaluate blood sugar levels.

**BMI Calculator**: A quick and easy calculator for assessing body mass index.

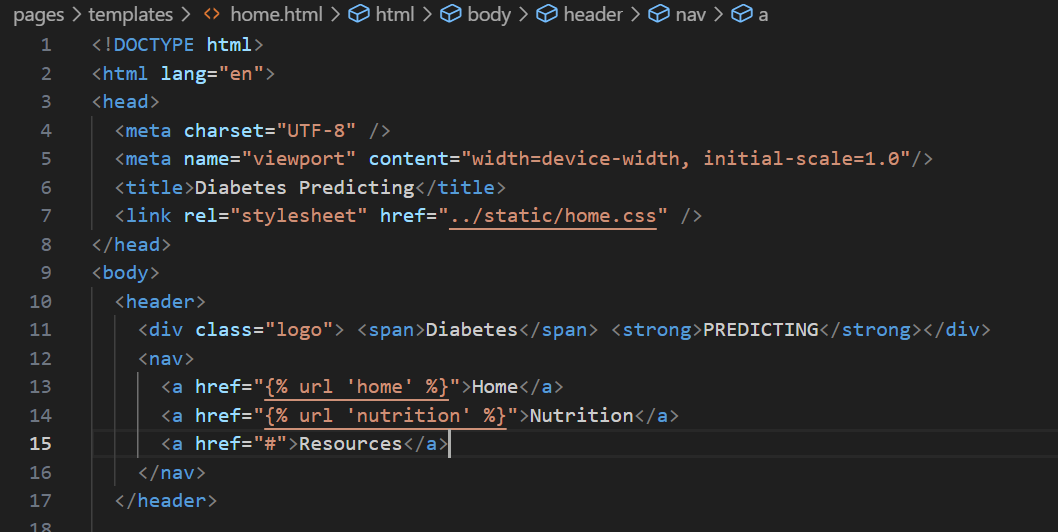
**Type Predictor**: A feature that predicts the type of diabetes using machine learning.

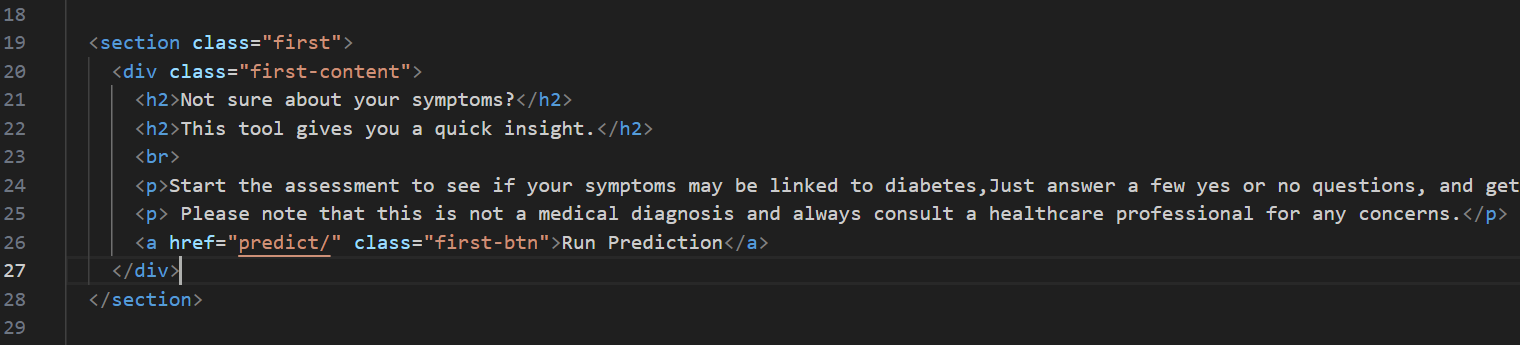
Each tool is presented as a **card** with a brief description and a “Start” button to guide users to the respective page. This layout ensures users can **quickly find** and **access** the tools they need.

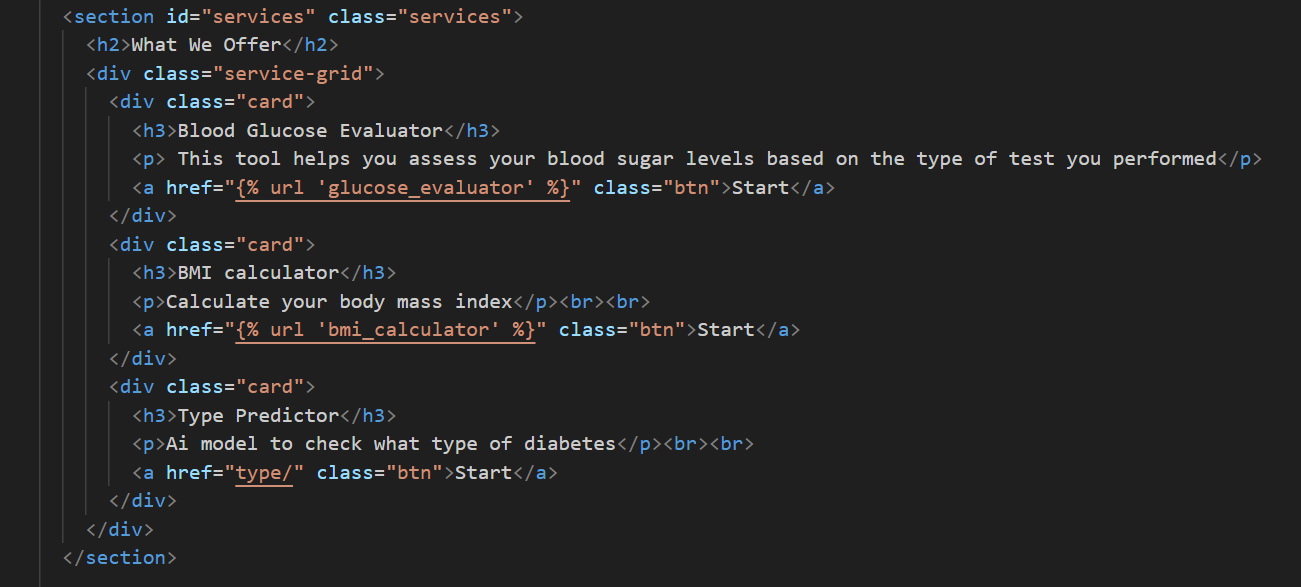
The home page’s **design and content** establish the system’s purpose and provide an **easy-to-navigate experience** for users, regardless of their technical background

**Code**:

The **home page** of the diabetes prediction system is created using a combination of **HTML** and **CSS**, ensuring a clear and engaging layout.







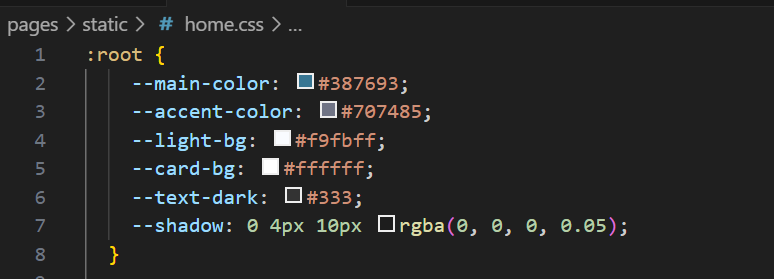
The structure of the page is defined in a standard <!DOCTYPE html> document.

The **header** contains the site logo and navigation links.

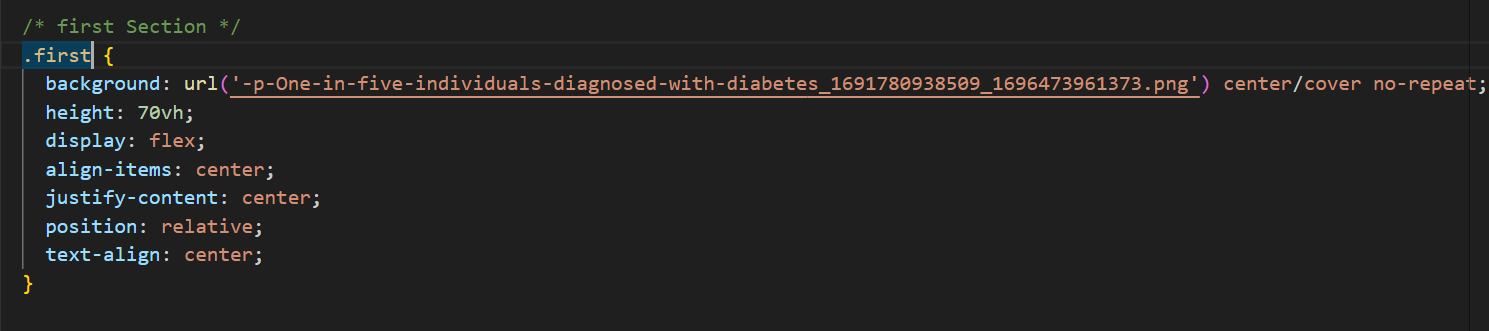
The first **section** includes headlines, descriptive text, and a prominent button to start the prediction.

The **“What We Offer”** section uses a **grid layout** to display three cards, each linking to one of the system’s main tools.

**CSS code:**

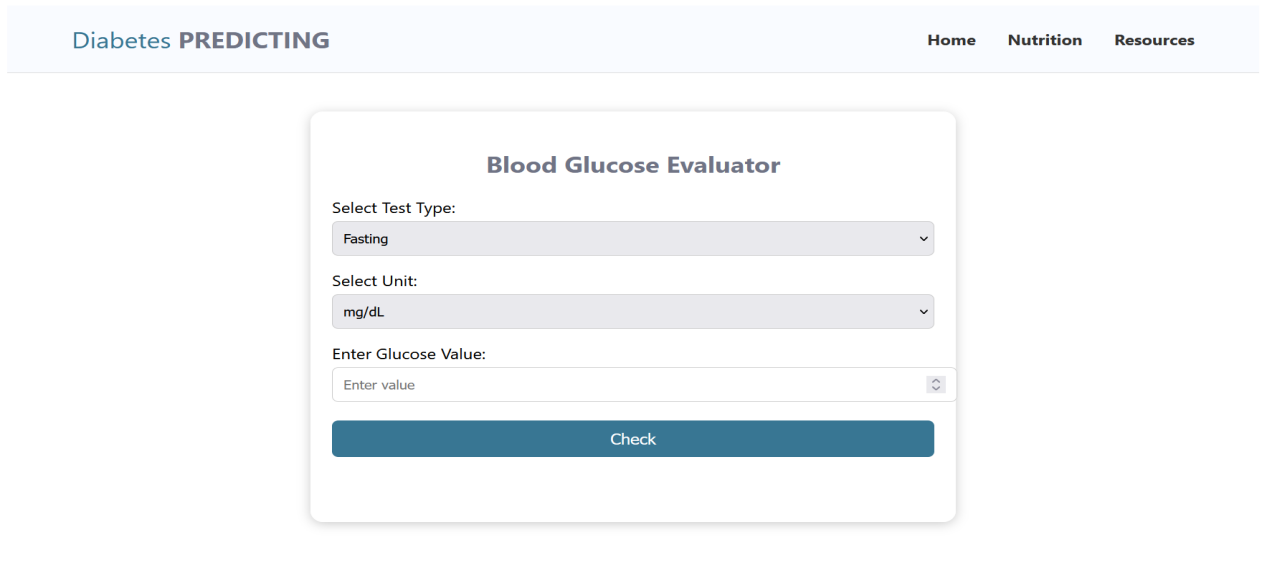
****

Styles are defined using **CSS variables** (like --main-color, --accent-color) for easy customization.



The first section uses a **background image** with an overlay to improve text readability.

**Blood Glucose Evaluator**



The **Glucose Evaluator page** allows users to assess their blood glucose levels quickly and conveniently. This page has a clean and structured layout with:

**Dropdown menus** for selecting:

The type of glucose test (e.g., fasting, post-meal, HbA1c).

The unit of measurement (mg/dL or mmol/L).

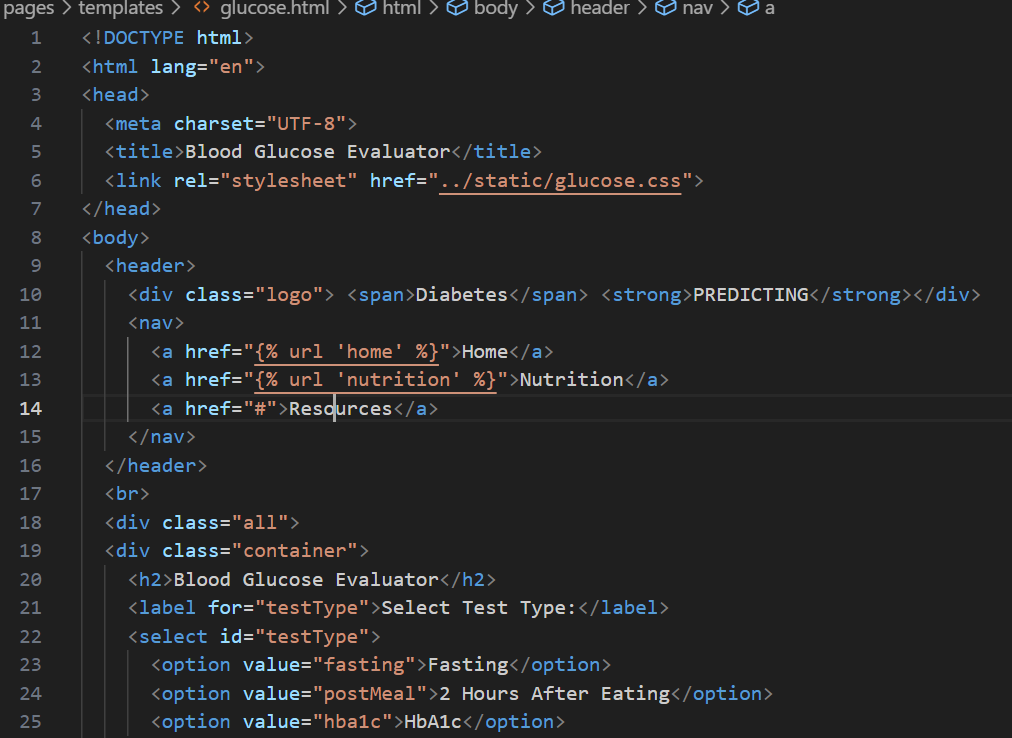
**Input field** to enter the glucose value.

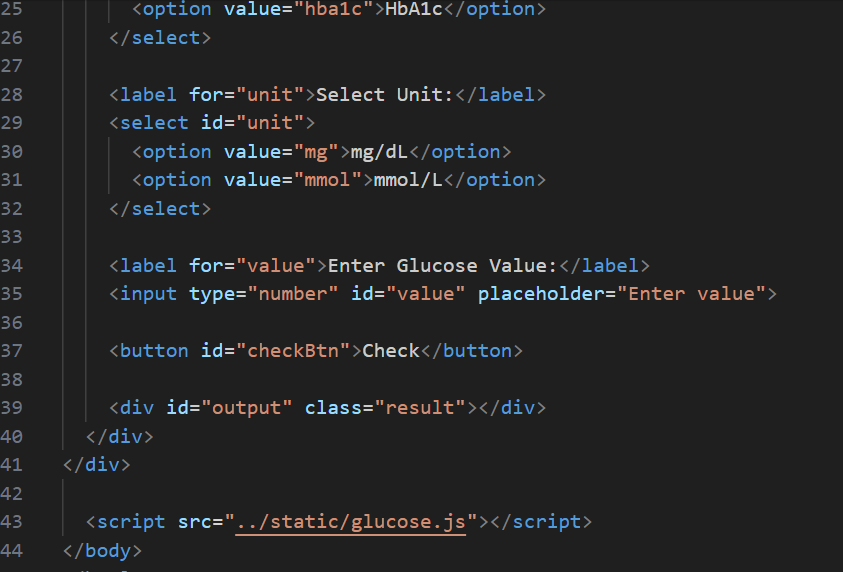
A **Check button** to run the evaluation.

A **result area** that displays the evaluation status (low, normal, pre-diabetic, diabetic).

This tool provides **immediate feedback** to users, helping them understand their blood glucose status and promoting awareness about diabetes risk.

**Html Code**





The **Glucose Evaluator page** is implemented using **HTML, CSS, and JavaScript** to provide a dynamic and interactive tool for users.

**HTML Structure**:

Contains form elements for selecting the **test type** and **unit** (dropdown menus).

Includes an **input field** where users enter their glucose measurement.

A **“Check” button** triggers the evaluation when clicked.

An **output area** displays the result in a clear, color-coded format.

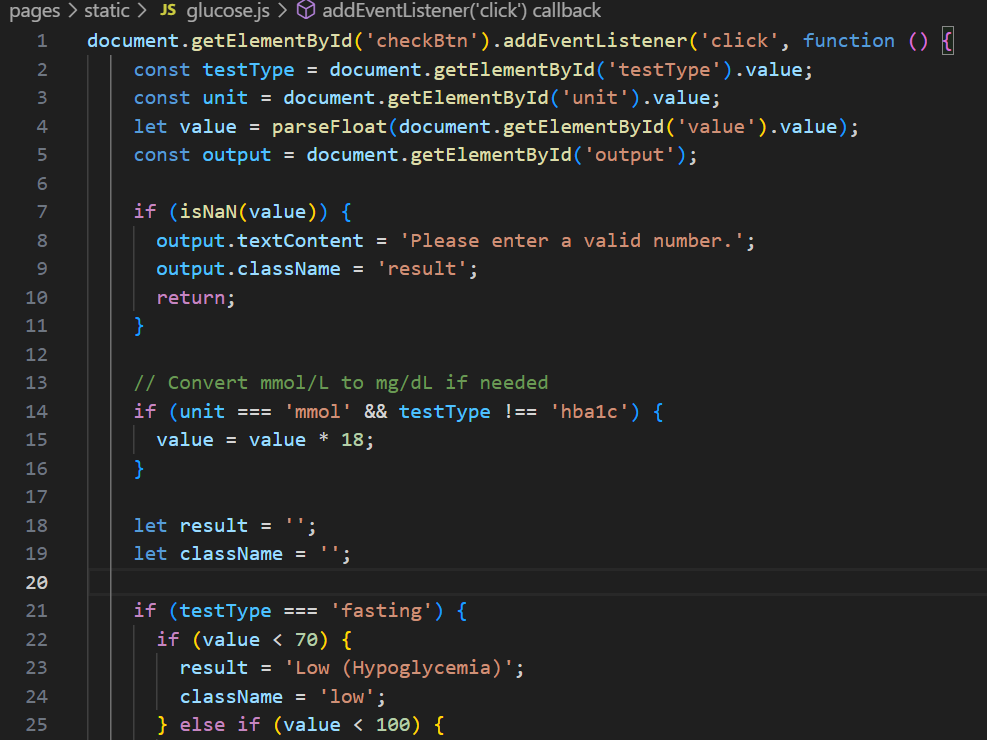
**CSS Styling**:

The page uses a **card-style container** to highlight the form.

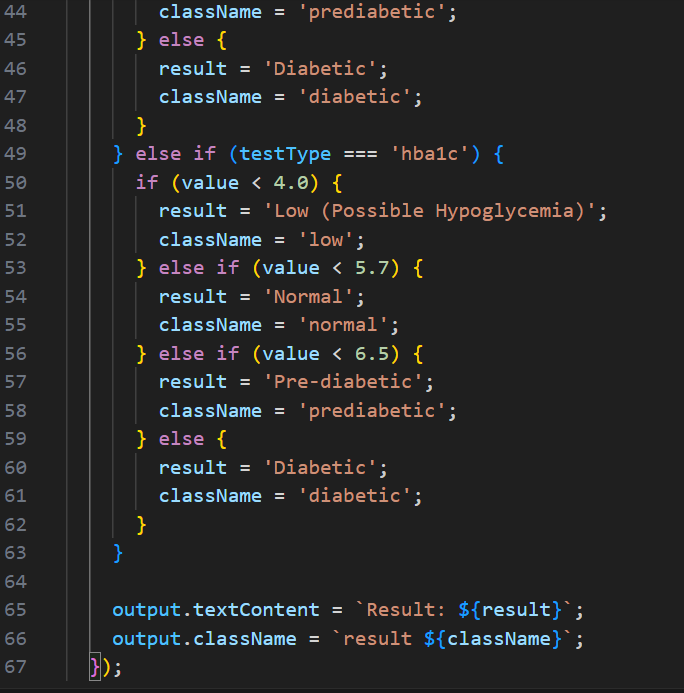
Custom **color themes** and shadows create a modern and accessible design.

Classes like .normal, .prediabetic, .diabetic, and .low apply different background colors to the result area, visually indicating the outcome.

**Javascript code**







The **JavaScript code** in the Glucose Evaluator page adds dynamic functionality, enabling real-time evaluation of the user’s blood glucose input.

The script listens for the user’s **button click**, processes the input data, and determines whether the glucose level is **low, normal, pre-diabetic, or diabetic** based on standard medical thresholds.

**How It Works**

**Input Retrieval**:

The code uses document.getElementById() to get the values for:

**Test type** (fasting, post-meal, HbA1c)

**Unit** (mg/dL or mmol/L)

**Glucose value** (user’s numeric input)

**Validation**:

It checks if the glucose value is a valid number. If not, it prompts the user to enter a valid number.

**Unit Conversion**:

If the user selects mmol/L, the code converts the glucose value to mg/dL (using the formula: 1 mmol/L = 18 mg/dL).

**Classification Logic**:

Uses if...else conditions to compare the input glucose value against medical thresholds for each test type.

Determines if the result is:

**Low (hypoglycemia)**

**Normal**

**Pre-diabetic**

**Diabetic**

**Output Display**:

Updates the **text** inside the result <div> to show the outcome.

Dynamically applies **CSS classes** (like .normal, .prediabetic, .diabetic, or .low) to color-code the result for easier interpretation.

This **JavaScript functionality** makes the page interactive, providing **instant feedback** without needing a page reload or external API. It enhances the user experience by:

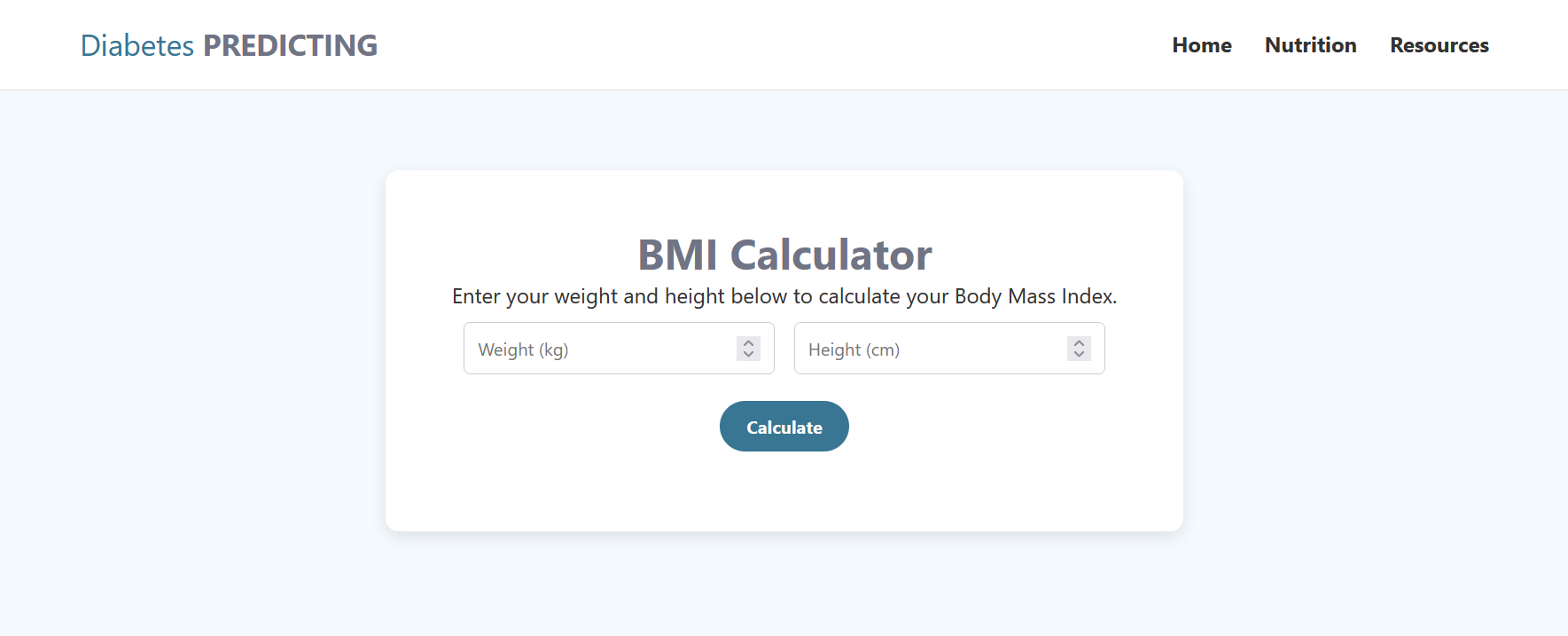
Giving immediate, clear, and visually distinct feedback.

Empowering users to understand their health data quickly.

Demonstrating the power of combining **web technologies** (HTML/CSS) with **JavaScript logic** for real-time evaluations.

This **interactive feature** allows users to **instantly assess** their blood glucose status based on medical guidelines, making it an essential part of the system’s mission to support health awareness and early diabetes detection.

**BMI Calculator**

****

The **BMI Calculator page** provides users with an easy-to-use tool for calculating their Body Mass Index (BMI). It helps users quickly assess whether they are underweight, normal, overweight, or obese.

The page features:

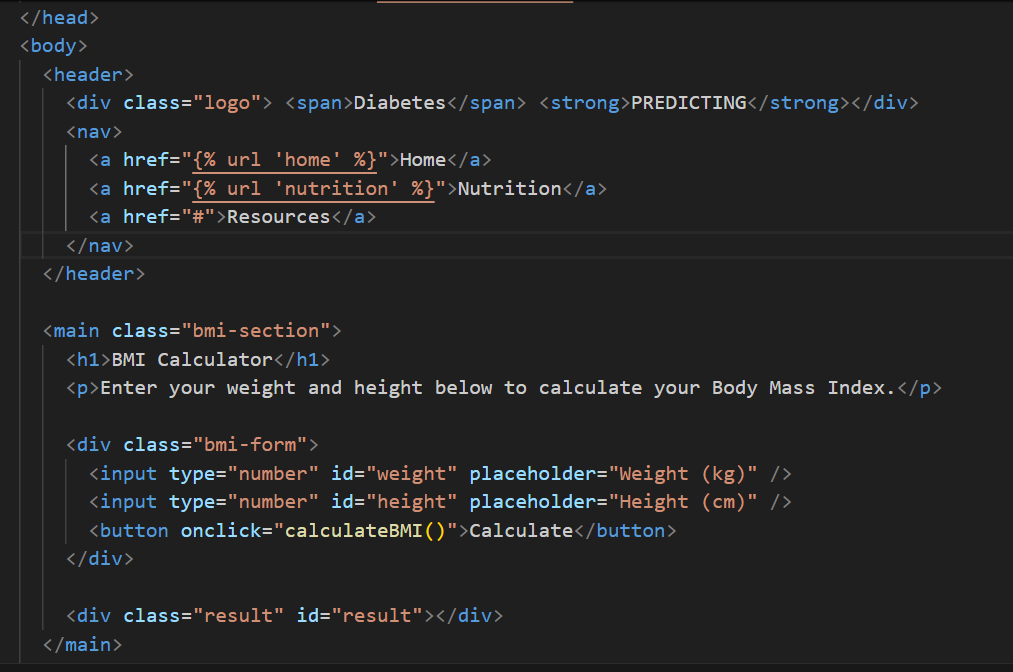
**Input fields** for **weight (kg)** and **height (cm)**.

A **Calculate button** to run the BMI calculation.

A **result area** that displays the BMI value and a short health status message.

This simple tool gives users immediate feedback about their weight relative to height, promoting healthy living and raising awareness about potential health risks.

**HTML code**

****

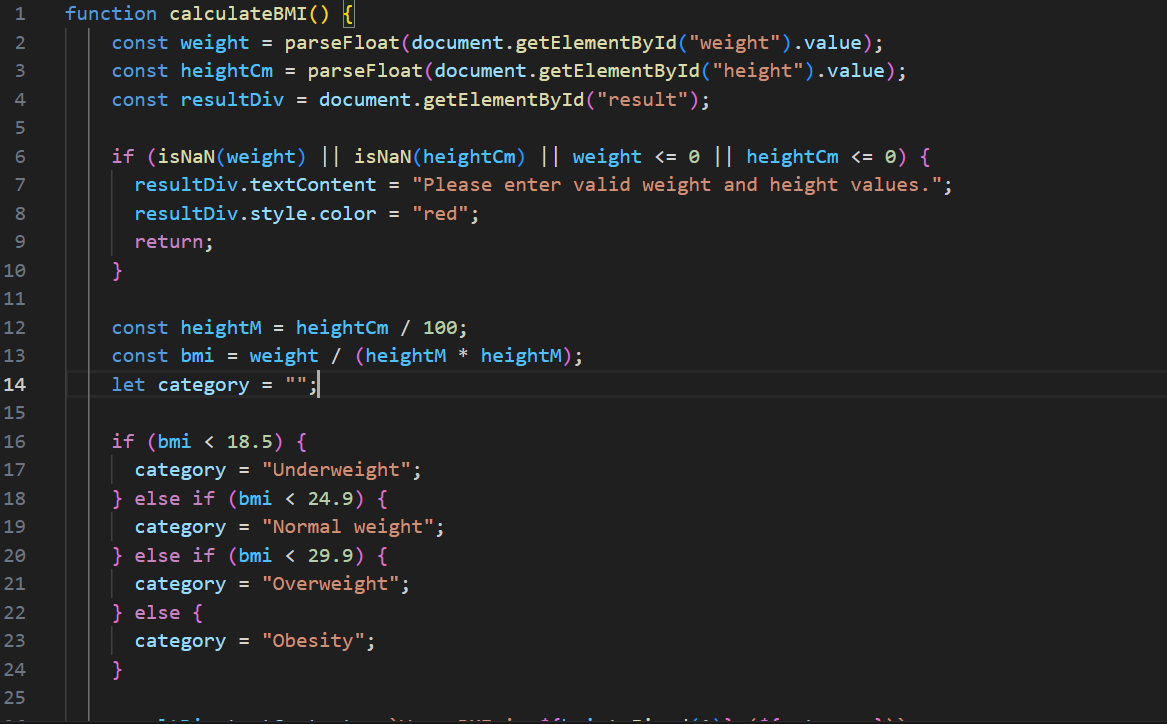
**HTML Structure**:

Contains input fields for the user’s **weight** and **height**.

A **Calculate button** that triggers the BMI calculation.

A **result section** to display the BMI value and weight status.

**Javascript code**

****

**JavaScript Logic**:

Listens for a **click event** on the Calculate button.

Retrieves the weight and height values entered by the user.

Performs the BMI calculation using the formula:

BMI = weight(kg)/(height(m))power2

(Note: The height is converted from **cm to m** for the calculation).

Determines the BMI category:

Underweight

Normal weight

Overweight

Obese

Displays the result in a user-friendly message with appropriate styling (green for normal, red for obese).

**Importance of this Feature**:  
The BMI calculator helps users quickly understand whether their weight is in a healthy range. It **raises awareness** about the importance of maintaining a balanced weight and can encourage healthier lifestyle choices.

**4.2. Backend**

**Django** is a high-level Python web framework that promotes rapid development and clean, pragmatic design. It follows the **Model-Template-View (MTV)** architectural pattern, making it easier to build robust and maintainable web applications.

#### ****Key Features of Django****

**Scalability** – Django is designed to help developers scale their applications smoothly.  
**Security** – It comes with built-in security features like protection against SQL injection, cross-site scripting, and cross-site request forgery.  
**Built-in Admin Panel** – Django includes an automatically generated admin interface to manage application data.

**Object-Relational Mapping (ORM)** – It simplifies database operations by mapping database tables to Python classes.  
**Reusability and Modularity** – Django projects are built with modular apps that can be reused and extended in other projects.

#### Django’s modular design make it an ideal framework for healthcare applications, such as this diabetes prediction system. It allows you to:

Efficiently **connect user input** (like health data) with the ML models.

**Handle data securely** using Django forms and CSRF protection.

**Serve predictions** in real time through dynamic views and templates.

**Project Structure**

manage.py: Entry point for running the project.

myproject/: Contains settings and configurations for the Django project.

diabetes\_predictor/ and typepredicting/: Separate Django apps handling different machine learning models.

pages/: Templates and views for the web interface.

**Django Apps and Views**

The diabetes\_predictor **app** handles diabetes risk prediction, using **Django views** to receive user input, load ML models, and return predictions.

The typepredicting **app** deals with classifying diabetes type (Type 1 or Type 2).

**Views** in each app use the **Django request-response cycle** to:

Validate form data.

Load ML models (joblib(pickle)).

Make predictions.

Render HTML templates to display results.

**Backend Integration with ML Models**

Models are trained separately and stored as .pkl files.

Django **loads these models dynamically** in the views, ensuring efficient resource usage.

The **results are formatted** and displayed to users using Django’s templating engine.

Integration of Machine Learning

A crucial part of the project is how the **Machine Learning models** are integrated with the Django web application. This integration allows users to input their data directly on the website and receive **real-time predictions** powered by trained ML models.

#### Main Steps in ML Integration

#### ****1. Pre-trained Models****

The ML models are trained beforehand using Python libraries such as:

**scikit-learn**

**pandas**

**numpy**

Once trained, the models are saved as .pkl files using joblib or pickle, making them easy to load later.

**2. Loading the Model in Django Backend**

In the **views** of Django project, the model is loaded when needed (when the user submits a form).

**3. Preparing User Input**

User data is retrieved from the form inputs in the HTML pages.

The data is then formatted to match the input format expected by the ML model

**4. Running the Prediction**

The model uses the input data to generate a prediction

**5. Displaying the Result**

The prediction result is sent back to an HTML template to be displayed to the user in a clear and user-friendly manner.

#### This Integration gives

**User-Friendly Interaction**: Users can get real-time predictions without needing to understand the technical details of the ML models.

**Efficient Automation**: The ML logic runs on the server instantly without reloading or using external software.

Prediction Workflow Summary

The entire flow can be summarized like this:

[User fills out the form]

↓

[Django View: Validates and processes input]

↓

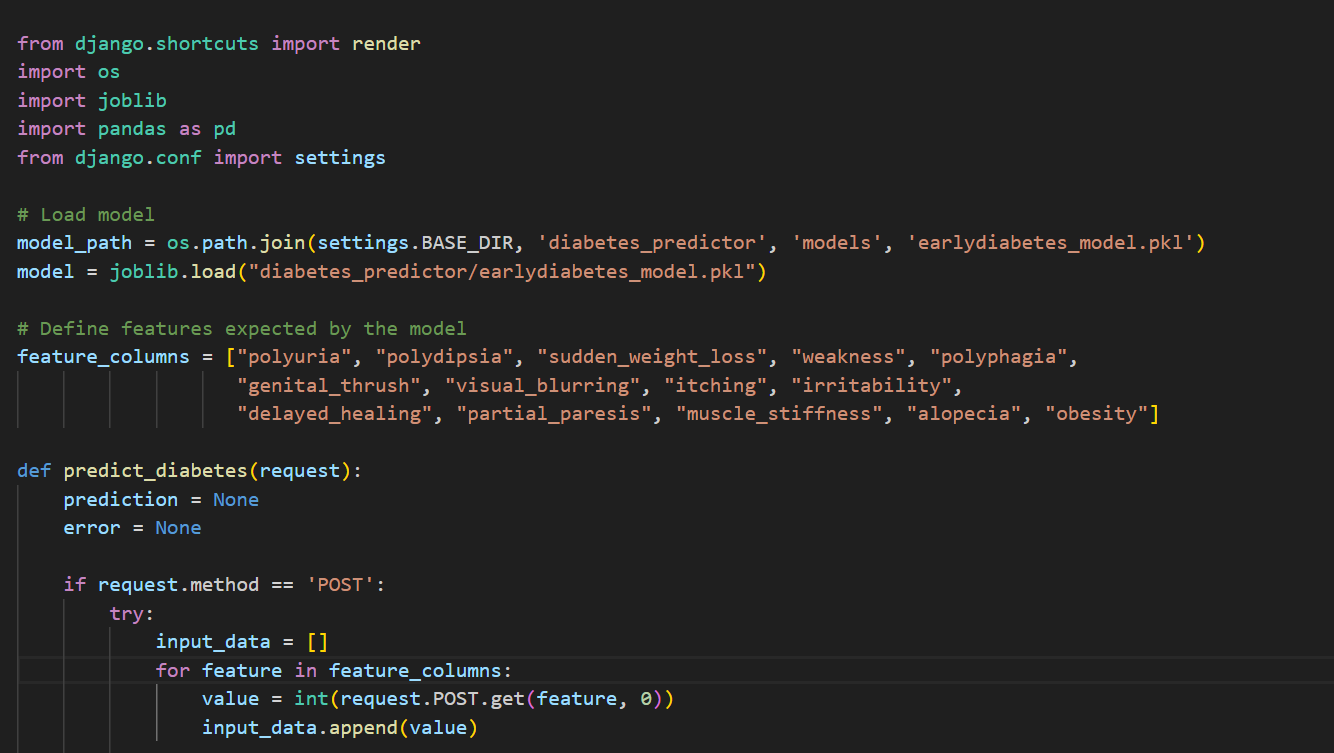
[Model is loaded from a file]

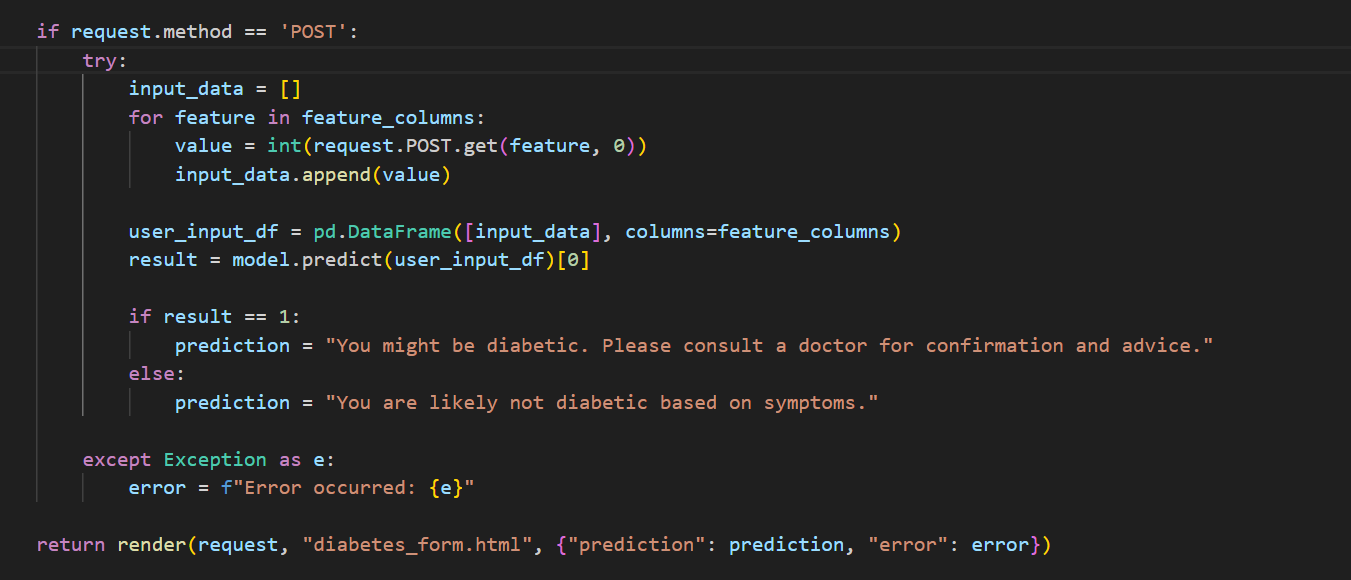
↓

[Prediction is made]

↓

[Result is displayed to the user]





#### Purpose of the View

This **view function** is the core logic that powers the **Diabetes Prediction** feature. It receives user input from an HTML form, prepares the data for the ML model, runs the prediction, and displays the result to the user.

**1. Importing Required Libraries**

joblib is used to load the pre-trained ML model (earlydiabetes\_model.pkl).

pandas helps format the user input as a DataFrame (as the ML model expects).

**2. Loading the ML Model**

The model is loaded once when the server starts:

model\_path = os.path.join(settings.BASE\_DIR, 'diabetes\_predictor', 'models', 'earlydiabetes\_model.pkl')

model = joblib.load(model\_path)

This avoids repeatedly loading the model for every request, improving performance.

**3. Defining Input Features**

feature\_columns lists the **14 symptoms/inputs** that the ML model expects.

These include symptoms like:

Polyuria

Polydipsia

Sudden weight loss

Weakness

…and others.

**4. Handling User Input (POST Request)**

The view collects data from the **submitted HTML form** using request.POST.get().

The collected values are converted to integers and stored in input\_data (a list of 14 features).

**5. Formatting Data for the Model**

User input is **converted to a pandas DataFrame**:

user\_input\_df = pd.DataFrame([input\_data], columns=feature\_columns)

This ensures the data structure matches what the model was trained on.

**6. Making the Prediction**

The ML model’s predict() function is called:

result = model.predict(user\_input\_df)[0]

The output (0 or 1) indicates whether the user is likely diabetic.

**7. Preparing User-Friendly Output**

If result is 1, a warning message encourages the user to see a doctor.

If 0, the user is reassured that no diabetes symptoms are detected.

**8. Error Handling**

A try/except block ensures that any unexpected errors (e.g., missing input) are caught and reported.

**9. Rendering the Result Page**

The final prediction or error message is passed to the HTML template (diabetes\_form.html) using Django’s render() function.

#### Key Benefits

**User-Friendly**: Converts complex ML predictions into simple health advice messages.

**Secure and Validated**: Input is handled safely within Django’s request framework.

**Efficient**: Loads the ML model only once and reuses it for fast predictions.

**4.3. Machine Learning**

This is the machine learning model for predicting whether a user has diabetes based on symptoms, It uses **pandas** for data manipulation, **scikit-learn** for machine learning algorithms, and **joblib** for saving the model.

:

****

#### ****1. Dataset Loading**** The code starts by **loading the dataset** from a CSV file, which contains several health-related symptoms as features, and a target column **“class’’** indicating whether the user has diabetes or not.

**2. Data Preparation**

**Features (**x**)**: All columns except class, gender, and age.

**Target (**y**)**: The class column, representing whether the user has diabetes.

**3. Splitting the Data**  
The dataset is split into a **training set** (80%) and a **testing set** (20%) to evaluate the model’s performance:

#### Model Comparison

we experimented several machine learning models:

**Logistic Regression**

**Decision Tree Classifier**

**Random Forest Classifier**

**Support Vector Machine (SVM)**

**Naive Bayes (BernoulliNB)**

#### Choosing the Best Model: Random Forest Classifier

After testing multiple models, the **Random Forest Classifier** was selected as the best-performing algorithm because it **achieved the highest accuracy**. Here’s why:

**Robustness**: It uses an ensemble of decision trees, which reduces overfitting and improves performance on new data.

**Accuracy**: Random Forest achieved the best results among the models tested.

#### How Random Forest Works

Random Forest is an **ensemble learning algorithm** that combines multiple **decision trees** to make predictions.

**Key Features**:

**Bootstrap Sampling**: Each decision tree is trained on a random sample of the data (with replacement), ensuring diversity in the forest.

**Feature Randomness**: At each split in the decision tree, only a random subset of features is considered, increasing robustness and reducing bias.

**Voting/Averaging**: In classification, each tree votes on the prediction, and the most frequent vote is chosen (majority rule).

**Advantages**:

High accuracy and generalization performance.

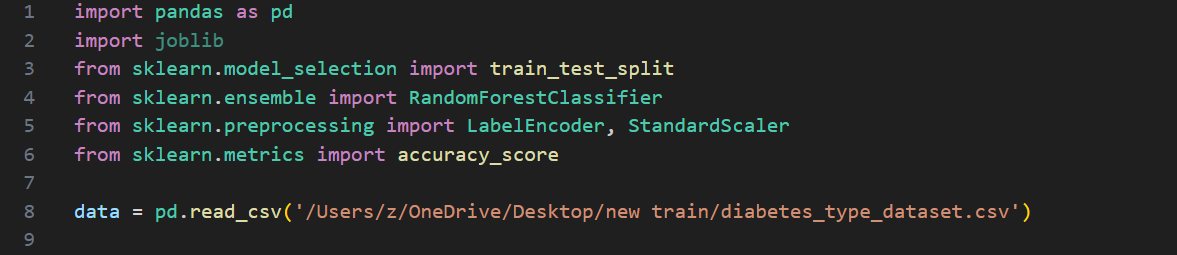
Resistant to overfitting (compared to single decision trees).

Can handle both numerical and categorical features well.

This well-trained Random Forest model is later **integrated into the Django backend**, allowing real-time predictions based on user symptoms. It’s a **core part** of our system’s ability to give users an accurate evaluation of their diabetes risk.

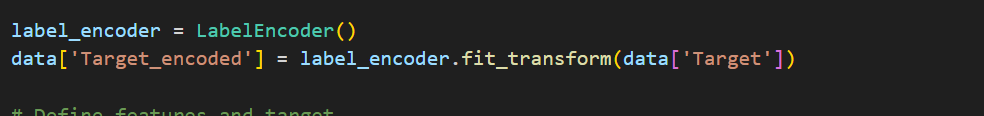
**This the machine learning model** that predicts the **type of diabetes** a user may have (Type 1, Type 2, or Pre-diabetic). It uses **pandas** for data manipulation, **scikit-learn** for machine learning algorithms, and **joblib** for saving the model.

**1. Dataset Loading**  
The code starts by **loading the dataset** from a CSV file:



This dataset contains health-related data and a **Target** column indicating the diabetes type.

**2. Label Encoding**  
Since the target column (Target) contains categorical text labels (e.g., “Type 1”, “Type 2”), it is **encoded** into numerical values using LabelEncoder:



### Label Encoding

Label encoding is a technique used to **convert categorical data (text labels) into numeric form**, making it usable by machine learning algorithms.

In the dataset, the **“Target”** column contains the diabetes type as text (e.g., “Type 1”, “Type 2”, “Pre-diabetic”). Machine learning models require **numerical values**, so we use LabelEncoder from scikit-learn:

This automatically assigns a unique number to each category, like:

Type 1 → 0

Type 2 → 1

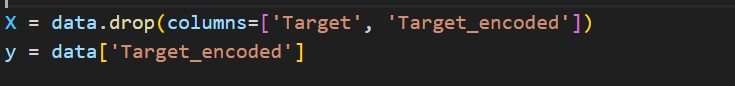
Pre-diabetic → 2

This transformation ensures the ML model can interpret and learn from the **categorical target variable**.

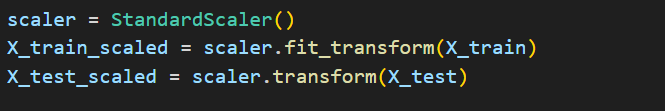
**3. Feature and Target Separation**

**Features** (X): All columns except the target.

**Target** (y): The newly created Target\_encoded column.



**4. Data Splitting**  
The data is split into **training and testing sets** to evaluate the model’s performance:Screenshot 2025-06-08 044018

**5. Feature Scaling**  
Standardization ensures that all features have a similar range:

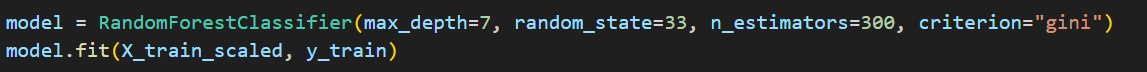
### Feature Scaling

Feature scaling ensures that all input features have **similar numeric ranges**, preventing features with larger values from dominating the training process.

**standard scaling** is used with StandardScaler from scikit-learn. It **centers** the data around 0 and scales it to have a **standard deviation of 1**:

This process Improves model **training stability**

**6.Model Training**  
A **Random Forest Classifier** is used for the prediction task:

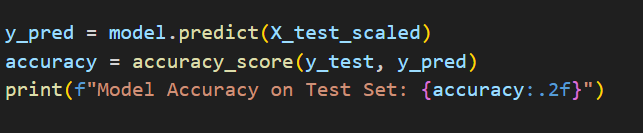


**max\_depth**: Limits the tree depth to prevent overfitting.

**n\_estimators**: Uses 300 decision trees to boost performance.

**criterion**: Uses the **Gini impurity** to evaluate splits.

**7. Model Evaluation**  
The model’s accuracy is calculated on the **test set:91%**



Chapter 5

Testing & Evaluation

### 5.1. Testing Strategies (Unit Testing, Integration Testing, User Testing)

The **testing phase** was crucial to ensure that the system is robust, accurate, and user-friendly.

**Unit Testing**

Each Django view and ML prediction logic was tested individually to verify that:

User inputs are correctly processed.

Models load and run without errors.

Results are rendered properly in the HTML templates.

**Integration Testing**

The **full pipeline** was tested to ensure smooth integration between:

User input forms

Backend logic (views)

Machine learning models

Result displays

Verified that predictions were **accurate** and **pages loaded seamlessly**.

**User Testing**

Conducted **manual testing** by interacting with the web application as an end user:

Checked forms for usability and clarity.

Ensured that error messages and feedback were clear and helpful.

Validated that results matched known test cases and expected behavior.

### 5.2. Performance Metrics (Accuracy, Speed, Scalability, etc.)

The performance of the system was measured using several key metrics:

**Accuracy**

For **Diabetes Prediction** :

Test Accuracy: **95%**

Precision: **0.95 ,** Recall: **0.96 ,** F1-score: **0.96**

For **Diabetes Type Prediction**:

Test Accuracy: **91%**

Precision: **0.93 ,** Recall: **0.91 ,** F1-score: **0.91**

**Speed**

Predictions are generated **instantly** using pre-trained ML models loaded with joblib.

No noticeable delay for users, ensuring a smooth experience.

**Scalability**

The Django framework’s **modular design** allows easy deployment on more powerful servers or cloud infrastructure for handling increased traffic.

**Robustness**

**Random Forest Classifier** was chosen after testing several models, including Logistic Regression, Decision Trees, SVM, and Naive Bayes.

Random Forest consistently delivered the **highest accuracy** and best generalization performance.

Chapter 6

Results & Discussion

#### ****Introduction****

This chapter presents the key findings and outcomes of the project, discussing how well the implemented system achieved its objectives and identifying key insights gained during development and testing. It also highlights the strengths and potential areas for improvement, providing a comprehensive discussion of the project’s overall impact and performance.

#### ****Summary of Findings****

**High Accuracy of Predictions**

The **diabetes prediction model** achieved an impressive accuracy of **95%** on the test dataset.

The **diabetes type prediction model** achieved an accuracy of **91.34%**, demonstrating balanced performance across Type 1, Type 2, and pre-diabetic classes.

**Interactive Health Tools**

Beyond ML predictions, the system features two important health assessment tools:

**BMI Calculator**: Allows users to input their weight and height and instantly receive their BMI and health status (underweight, normal, overweight, or obese).

**Glucose Evaluator**: Lets users enter their glucose measurement, select the type of test performed (fasting, post-meal, or HbA1c), and receive real-time feedback on whether their levels are low, normal, pre-diabetic, or diabetic.

**Nutrition Page for Dietary Advice**

The system also includes a **Nutrition page** that provides dietary recommendations for **Type 1** and **Type 2 diabetes**.

This page educates users about what foods to avoid or limit to support diabetes management, extending the system’s role as a tool for **health education**.

**Seamless Integration of ML and Web Technologies**

The project successfully combines ML models with a Django-based web application, delivering **real-time, secure, and user-friendly** predictions and health advice.

#### ****6.3 Interpretation of Results (Did the project meet its objectives?)****

The project successfully achieved its primary goals:

**Objective 1: Accurate Predictions**

Both ML models demonstrated high accuracy, confirming that ML can provide reliable predictions for diabetes risk and classification.

**Objective 2: Comprehensive Health Evaluation**

The inclusion of the BMI calculator, glucose evaluator, and Nutrition page supports a **holistic approach** to health awareness and diabetes management.

**Objective 3: User-friendly System**

The clean interface and clear feedback make the system accessible and easy to use for a wide range of users, regardless of their technical background.

**Objective 4: Robust Integration**

ML predictions are seamlessly delivered within the Django framework, ensuring **smooth real-time operation** and consistent user experience.

**Objective 5: Education and Support**

The Nutrition page provides practical, easy-to-understand dietary advice, supporting users in **making healthier choices**.

Overall, the project successfully combines **data-driven insights** with an engaging web interface to meet the project’s mission of **early health awareness** and **user empowerment**.

#### ****6.4 Limitations of the Proposed Solution****

**Manual Data Entry**

The BMI calculator and glucose evaluator rely on **self-reported data** (weight, height, glucose measurements), which could be subject to user error. Integration with wearable devices would enhance data accuracy.

**No Real-time Clinical Integration**

The system does not yet integrate with clinical health systems or real-time wearable data, limiting its ability to support continuous health monitoring.

**General Dietary Recommendations**

While the Nutrition page provides helpful advice, it is general and may not fully address **personalized nutritional needs**. Users should consult a dietitian or doctor for tailored guidance.

**Not a Substitute for Professional Diagnosis**

This system is intended as an educational and awareness tool only and is not a replacement for professional medical assessment and diagnosis.

#### Conclusion of Chapter 6

The project successfully delivers a **highly accurate, interactive, and educational** system for diabetes risk prediction, type classification, and basic health awareness. It provides a solid foundation for future enhancement and demonstrates the power of combining **machine learning** with **modern web frameworks** to promote health education and proactive self-care. While there are limitations to consider, the project meets its primary objectives and showcases the potential of data-driven healthcare tools in real-world scenarios.

Chapter 7

Conclusion & Future Work

#### ****7.1 Summary of Contributions****

This project successfully demonstrated how **machine learning** can be combined with **web development** to create an interactive tool that predicts diabetes risk, classifies diabetes types, and evaluates important health indicators.

Key contributions include:

**Accurate ML Predictions**

Developed and evaluated high-performing ML models (Random Forest) for predicting diabetes risk and classifying diabetes type.

**Interactive Health Tools**

Integrated BMI and glucose evaluation tools to provide users with a holistic health risk profile.

**Nutrition Guidance**

Developed a Nutrition page with dietary recommendations for Type 1 and Type 2 diabetes to support better health choices.

**User-friendly Web Interface**

Designed a responsive and intuitive Django-based interface that allows real-time predictions and health feedback.

**Secure, Modular Design**

Created a modular and secure system that can be **scaled or extended** for future enhancements.

#### ****7.2 Possible Improvements or Extensions for Future Work****

While the current system is robust and functional, there are many exciting opportunities for improvement and expansion:

**1. Personalization and User Accounts**

Implement user authentication and profiles to track personal health data over time.

Allow users to compare past predictions and monitor trends.

**2. Advanced Visualization and Feedback**

Add **data visualization** (charts, progress bars) to help users better understand trends in their health data.

**3. Scalability and Security Enhancements**

Optimize the system for cloud deployment and larger user bases.

Implement **stronger security measures** (e.g., HTTPS, data encryption) to protect sensitive health data.

#### Conclusion of Chapter 7

The project has laid a strong foundation for combining **ML-driven insights** with **user-friendly web technology** to support health awareness and risk assessment. With these potential improvements, the system can evolve into a **powerful tool** for proactive health management and education.

**Chapter 8**

**References**

* [**https://www.kaggle.com/code/ahmedaboraida/early-classification-of-diabetes/input**](https://www.kaggle.com/code/ahmedaboraida/early-classification-of-diabetes/input)
* [**https://www.w3schools.com/python/python\_ml\_scale.asp**](https://www.w3schools.com/python/python_ml_scale.asp)
* [**https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/**](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/)
* [**https://www.w3schools.com/django/django\_views.php**](https://www.w3schools.com/django/django_views.php)
* [**https://www.w3schools.com/python/python\_ml\_preprocessing.asp**](https://www.w3schools.com/python/python_ml_preprocessing.asp)
* [**https://www.moh.gov.sa/awarenessplateform/ChronicDisease/Pages/Diabetes.aspx**](https://www.moh.gov.sa/awarenessplateform/ChronicDisease/Pages/Diabetes.aspx)
* [**https://www.dailymail.co.uk/health/article-12136373/Countries-worst-diabetes-rates-world-REVEALED-Americas-position-shock-you.html**](https://www.dailymail.co.uk/health/article-12136373/Countries-worst-diabetes-rates-world-REVEALED-Americas-position-shock-you.html)
* [**https://www.w3schools.com/python/pandas/default.asp**](https://www.w3schools.com/python/pandas/default.asp)