Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna
College of Computer Studies

Document Metadata	
Project Name	Diabetes Risk Predictor (Flask Edition)
Version	1.0 (Finals Submission)
Status	In Development
Date	December 30, 2025
Author	Mamaw Coders
Target Audience	General Public / Academic Review Board

Product Requirements Document (CSST 101)

Project: Diabetes Risk Assessment AI (Sprint Edition)

1.0 Executive Summary

1.1 Problem Statement

Type 2 Diabetes is a silent epidemic in the Philippines and globally. A significant portion of the population exists in a "Pre-diabetic" or high-risk state but remains undiagnosed due to the cost of medical screening, lack of awareness, and fear of invasive procedures. Existing digital solutions often focus on management for diagnosed patients (e.g., glucometer logs) rather than accessible, early-stage detection for the general public.

1.2 Proposed Solution

The **Diabetes Risk Predictor** is a web-based artificial intelligence tool designed to democratize access to health screening. By leveraging machine learning (Random Forest Classification) trained on over 70,000 clinical records, the application provides users with an instant, data-driven assessment of their diabetes risk profile.

1.3 Success Criteria

- **Accuracy:** The underlying model achieves a recall rate sufficient to minimize false negatives (missing a high-risk patient).
- **Accessibility:** The web application loads in under 2 seconds on standard mobile networks and requires no user registration.
- **Usability:** A "Guest User" can complete the assessment and understand their risk status in less than 30 seconds.

2.0 Goals & Non-Goals

2.1 Primary Goals (Sprint Scope)

- **Professional User Interface:** Develop a responsive, "Medical Grade" interface using **Bootstrap 5** that instills trust and clarity.
- **Machine Learning Integration:** Successfully deploy a Python Flask backend that serves real-time inference from a pre-trained **Random Forest Classifier**.
- **Localization:** Adapt the US-centric dataset (BRFSS 2015) to Filipino contexts by implementing automatic unit conversion (e.g., Kilograms/Centimeters to BMI).
- **Privacy-First Design:** Ensure no personal health data is persisted in a database, adhering to strict ethical standards for student projects.

2.2 Non-Goals (Out of Scope)

- **User Authentication:** No login, signup, or password management features will be implemented to reduce user friction.
- **Long-Term Tracking:** The system will not store historical data or track user progress over time.
- **Medical Diagnosis:** The tool is explicitly for *screening* and *educational purposes*, not for medical diagnosis or prescription.
- **Native Mobile Application:** The project is limited to a responsive web application; no iOS/Android binaries will be released.

3.0 Stakeholders

Role	Description	Key Interests
The Guest User	General public (Age 18+).	Fast results, privacy, mobile compatibility, and clear, non-technical language.
The Evaluator	Course Instructor / Academic Board.	Model validity, code quality (PEP-8 standards), and UI polish.
The Developer	Project Team.	Feasibility within the 1-week timeline and successful deployment to a public URL.

Functional Requirements

4.1 AI & Backend Logic (Flask)

- **FR-01 (Inference Engine):** The application shall expose a RESTful endpoint (/predict)

capable of accepting POST requests containing form data.

- **FR-02 (Model Loading):** The backend must load the serialized model artifact (model.pkl) into memory upon application startup to ensure low-latency inference.
- **FR-03 (Input Validation):** The system must sanitize all inputs to prevent errors (e.g., ensuring Age > 0, Weight > 0) before passing data to the model.

4.2 User Interface (Bootstrap/HTML)

- **FR-04 (Responsive Layout):** The input form shall utilize the Bootstrap Grid System to stack interface elements vertically on mobile devices and horizontally on desktops.
- **FR-05 (Grouped Inputs):** Inputs shall be logically categorized into three distinct sections: "Demographics," "Vitals," and "Medical History," using Bootstrap Cards.
- **FR-06 (Result Visualization):** Inference results must be displayed prominently via a Bootstrap Alert component (Green for Low Risk, Red for High Risk), appearing immediately after form submission.

4.3 Business Logic & Localization

- FR-07 (BMI Calculation): The application must accept Weight in Kilograms (kg) and Height in Centimeters (cm). The backend will execute the following transformation before inference:

$$BMI = \frac{Weight\ (kg)}{(Height\ (cm) / 100)^2}$$

- **FR-08 (Age Mapping):** The system must map user-selected age ranges (e.g., "30-34") to the specific categorical integers required by the BRFSS dataset schema.

5.0 System Architecture

The application follows a **decoupled frontend/backend architecture** for independent deployment and scalability.

5.1 Technology Stack

- **Frontend (View):** HTML5, CSS3, **Bootstrap 5.3** (UI Framework), Vanilla **Javascript** (ES6+).
- **Backend (Controller): Python 3.9+**, **Flask** (Micro-framework), **Marshmallow** (Validation).
- **Machine Learning (Model): Scikit-Learn** (Random Forest Classifier), Joblib (Serialization).
- **Infrastructure: Render** (Cloud PaaS) or similar Python-compatible hosting.

5.2 Data Flow

1. **User Input:** Client submits HTML Form -> HTTP POST Request.
2. **Processing:** Flask Controller accepts request -> Validation -> BMI Calculation.
3. **Inference:** Controller calls model.predict(input_vector).
4. **Response:** Controller renders index.html with the result string injected via Jinja2.

6.0 Data Dictionary & Schema

The following table defines how user inputs are mapped to the machine learning model's expected features.

UI Label	HTML name Attribute	Python Variable	Dataset Feature	Data Type	Notes
High BP	high_bp	high_bp	HighBP	Binary (0/1)	History of Hypertension
High Cholesterol	high_chol	high_chol	HighChol	Binary (0/1)	History of High Cholesterol
Weight (kg)	weight	weight	BMI	Integer	Used to calculate BMI
Height (cm)	height	height	BMI	Integer	Used to calculate BMI
Smoker	smoker	smoker	Smoker	Binary (0/1)	Smoked >100 cigs in lifetime
History of Stroke	stroke	stroke	Stroke	Binary (0/1)	Ever diagnosed with stroke
Heart Disease	heart_disease	hda	HeartDiseaseorAttack	Binary (0/1)	Coronary Heart Disease / MI
Physically Active	phys_activity	phys_activity	PhysActivity	Binary (0/1)	Exercise in past 30 days
General Health	gen_health	gen_hlth	GenHlth	Int (1-5)	1=Excellent, 5=Poor
Age Group	age	age	Age	Int (1-13)	1=18-24 ... 13=80+

7.0 UI/UX Design Specification

7.1 Design Philosophy

The interface mimics a standard medical form—clean, sterile, and trustworthy. We utilize a "Card-based" layout to reduce cognitive load.

7.2 Interface Structure

- **Header (Jumbotron):**
  - Title: "Diabetes Risk Assessment AI"
- **Form Section:**
  - *Card A (Demographics):* Age Dropdown, Sex Radio Buttons.
  - *Card B (Vitals):* Weight (kg) Input, Height (cm) Input.
  - *Card C (History):* Toggle Switches for BP, Cholesterol, Smoking, Activity.
- **Footer:**
  - Disclaimer: "For educational purposes only. Not a substitute for professional medical advice."

8.0 Development Sprint Plan (7 Days)

Day	Phase	Key Activities
1	**Data Engineering**	Load diabetes_binary_5050split.csv. Train Random Forest model. Evaluate metrics. Export model.pkl.
2	**Backend Setup**	Initialize Flask app.py. Create routes. Test local server.
3	**Frontend Skeleton**	Create templates/index.html. Implement Bootstrap 5 starter template. Build HTML form.
4	**Logic Integration**	Connect Form POST to Flask. Write BMI calculation logic. Debug input types.
5	**Model Wiring**	Load model.pkl. Pass inputs to model. Return prediction results to the UI.
6	**Deployment**	Create requirements.txt. Push code to GitHub. Deploy to Render/Cloud Platform.

7	**Quality Assurance**	Mobile responsiveness testing. Add visual polish (colors, icons). Final submission.

9.0 Risks & Mitigations

Risk Area	Risk Description	Mitigation Strategy
Deployment	Hosting Flask on free-tier services (Render) can be complex compared to Streamlit.	Allocate Day 6 entirely to deployment. Follow a verified "Flask to Render" tutorial.
Model Bias	The model is trained on US data (BRFSS 2015), which may not accurately reflect Filipino physiological norms.	Explicitly state this limitation in the application footer and presentation slides.
User Input Error	Users may input unrealistic values (e.g., Weight: 5kg) which could skew predictions.	Implement basic HTML5 form validation (min="30", required).