**Phase-1 Submission Template**

**Student Name:** Anusha S

**Register Number:** 71772317103

**Institution:** Government College of Technology, Coimbatore

**Department:** Computer Science and Engineering

**Date of Submission:** 13-04-2025

**1.Problem Statement**

In many healthcare systems, disease diagnosis is primarily reactive, occurring after noticeable symptoms emerge. This often results in delayed treatments, increased healthcare costs, and poorer patient outcomes. The lack of predictive tools for early disease detection is especially problematic for chronic and life-threatening conditions such as heart disease, diabetes, and cancer. The aim of this project is to leverage AI technologies to predict potential diseases using patient data after some early symptoms . This proactive approach can drastically improve early intervention, personalize treatments, and ultimately save lives.

**2.Objectives of the Project**

Develop a machine learning model capable of predicting the likelihood of specific diseases based on patient data.

Identify and analyze key health indicators contributing to the early onset of diseases.

Provide actionable insights to healthcare providers and patients to support early intervention and preventive care

Build a simple, user-friendly interface (optional) for users to input data and receive disease risk predictions.

# 3.Scope of the Project

**Features to Analyze /Build:**

* Patient demographics (age, gender, etc.)
* Vital signs (blood pressure, glucose levels, etc.)
* Medical history and lifestyle factors (smoking, physical activity)
* Predictive models for common diseases (e.g., diabetes, heart disease)

**Constraints:**

* Use of publicly available datasets
* Limited to offline prediction models (no real-time data streaming)

# 4.Data Sources

# Dataset Source: Kaggle (e.g., *Heart Disease UCI*, *Diabetes Dataset*)

# Availability: Public

# Type: Static (downloaded and used as-is)

# Additional Notes: May use synthetic data to supplement specific attributes if necessary

# 5.High-Level Methodology

**Data Collection**

Data will be sourced primarily from public repositories such as **Kaggle** (e.g., Heart Disease UCI dataset, Pima Indians Diabetes dataset). These datasets are well-structured and widely used in research. Additional data may be synthetically generated using Python libraries like scikit-learn's make\_classification() to simulate missing variables or balance class distributions.

**Data Cleaning**

Common data quality issues such as **missing values**, **duplicate records**, and **inconsistent data formats** will be addressed. Techniques include:

* Imputation (mean/median for numerical features, mode for categorical)
* Dropping or merging duplicates
* Normalizing inconsistent units or formats
* Encoding categorical variables where needed (e.g., one-hot or label encoding)

**Exploratory Data Analysis (EDA)**

EDA will be conducted using **Pandas**, **Matplotlib**, and **Seaborn** to visualize:

* Class distributions
* Correlation heatmaps
* Histograms and boxplots for feature distributions
* Scatter plots and pair plots for relationship analysis This will help in understanding feature relevance and data patterns.

**Feature Engineering**

* New features may be derived from existing ones (e.g., BMI from height and weight)
* Normalization or standardization of features to improve model performance
* Dimensionality reduction techniques (like PCA) may be used if necessary

**Model Building**

We will experiment with several classification algorithms such as:

* **Logistic Regression** – For baseline performance and interpretability
* **Random Forest** – For handling non-linear relationships and feature importance
* **Support Vector Machines (SVM)** – For high-dimensional classification
* **XGBoost** – For boosting accuracy through ensemble learning

These models are chosen based on their proven effectiveness in healthcare prediction tasks.

**Model Evaluation**

Models will be evaluated using:

* **Accuracy**, **Precision**, **Recall**, **F1-score**
* **ROC-AUC curve** to evaluate performance across thresholds
* **K-fold cross-validation** to ensure robustness and avoid overfitting

**Visualization & Interpretation**

Key insights and predictions will be visualized through:

* Confusion matrices
* ROC curves
* Feature importance plots
* Dashboards or Jupyter notebooks with interactive visualizations using **Plotly** or **Streamlit** (optional)

**Deployment**

If feasible, the final model will be deployed as a simple **Streamlit web app** or a **Jupyter Notebook-based interactive report**. This will allow users to input test data and view predictions directly, demonstrating practical applicability.

# 6.Tools and Technologies

* **Programming Language:**  
  Python,,HTML,CSS
* **Notebook/IDE:**  
  Google Colab and Jupyter Notebook (for development, testing, and visualization)
* **Libraries:**
  + **Data Processing:** pandas, numpy
  + **Visualization:** matplotlib, seaborn, plotly
  + **Modeling:** scikit-learn, xgboost, statsmodels
  + **Others (optional):** imbalanced-learn (for handling class imbalance), joblib or pickle (for model saving)
* **Tools for Deployment:**
  + **Streamlit** – For building an interactive web app to demonstrate model predictions
  + **Flask** – If a lightweight API is needed for backend model serving
  + **Gradio** – For rapid prototyping of model interfaces

**Additional Frontend Technologies**

**HTML** – For custom templates (e.g., Flask + Jinja)

**CSS** – For UI styling

# 7.Team Members and Roles

1. Data and research - Dharani A
2. Model Development - Sibitha S
3. Frontend and UI Developer - Geetharani C
4. Integration and Backend Developer - Anusha S
5. Deployment and documentation - Devisri V