Phase 3- Submission Template

# Student Name : B.PREM KUMAR

# Register Number : 421323205038

# Institution : Krishnasamy College Of Engineering And Technology

# Department : B.Tech IT

# Date Of Submission : 13-05-2025

# Github Repository link :[ https://github.com/Srikanth121705/project-22.git]

# 1. Problem Statement

In modern healthcare, timely and accurate disease prediction plays a critical role in improving patient outcomes and reducing medical costs. However, traditional diagnostic methods often fall short when dealing with high-dimensional data, nonlinear patterns, and heterogeneous patient populations. This project addresses the challenge by building an AI-based predictive model that leverages electronic health records (EHRs) to identify potential diseases in patients. By applying supervised machine learning to structured patient data—including symptoms, history, and demographic details—we aim to automate the diagnostic process and improve clinical decision-making.

# 2. Project Objectives

- Develop an AI-driven disease diagnosis model using structured patient datasets.  
- Minimize diagnostic errors by increasing prediction accuracy and reducing false positives/negatives.  
- Identify key symptoms and patterns associated with various diseases.  
- Provide an interactive and scalable framework deployable in real-time clinical environments.  
- Ensure model transparency for interpretation by healthcare professionals using feature importance and explainable AI techniques.

# 3. Project Workflow

1. Data Collection – Import structured health records from disease\_diagnosis.csv.  
2. Data Cleaning – Handle missing data, outliers, and inconsistencies.  
3. Feature Engineering – Create derived features and normalize data.  
4. Exploratory Data Analysis (EDA) – Understand distributions and relationships.  
5. Model Selection – Train multiple classifiers like Decision Trees, Random Forests, and Neural Networks.  
6. Model Evaluation – Use accuracy, precision, recall, F1-score.  
7. Visualization & Interpretation – Highlight key features influencing predictions.  
8. Deployment (optional) – Develop a simple interface using Gradio or Flask.

# 4. Data Description

Dataset: disease\_diagnosis.csv  
- Records: 4920 entries  
- Features: Symptoms encoded as binary features (1 = present, 0 = absent)  
- Target Variable: Disease (e.g., 'Fungal infection', 'Allergy', etc.)  
- Nature: Binary classification with multiple disease labels (multi-class classification)  
- Cleaned dataset with no null values

# 5. Data Preprocessing

- Missing Value Treatment: Dataset had no missing values.  
- Encoding: Labels in prognosis (target) were label-encoded.  
- Feature Scaling: Not required due to binary input features, but models like SVM benefit from scaling.  
- Data Split: 80% for training, 20% for testing using train\_test\_split(random\_state=42)

# 6. Exploratory Data Analysis (EDA)

- Symptom Frequency: Plotted bar graphs showing the most common symptoms across diseases.  
- Disease Distribution: Imbalanced dataset; some diseases have significantly more samples.  
- Correlation Analysis: Checked co-occurrence of symptoms using heatmaps.  
- PCA (optional): Visualized separation of diseases in 2D for model interpretability.

# 7. Feature Engineering

- Symptom Count Feature: Total number of symptoms per patient.  
- Interaction Terms: Combined key symptoms that often co-occur (e.g., itching + skin\_rash).  
- Dimensionality Reduction (optional): Applied PCA for visualization, but full feature set was retained for training.

# 8. Model Building

Algorithms Used:  
- Decision Tree Classifier: Interpretable and effective on categorical data.  
- Random Forest: Best performer; handles overfitting and provides feature importance.  
- Multinomial Naive Bayes: Good baseline due to binary symptom features.  
- K-Nearest Neighbors: Tested but slower for high-dimensional space.  
- MLPClassifier (Neural Net): Achieved competitive results with proper tuning.  
  
Best Performing Model: Random Forest Classifier  
- Accuracy: ~98%  
- Precision, Recall, F1-score: Above 95% for most classes

# 9. Visualization & Model Insights

- Confusion Matrix: Displayed using Seaborn to visualize classification performance.  
- Feature Importance Plot: Identified top 10 symptoms contributing to predictions.  
- t-SNE: Used for visualizing class separability in 2D space.

# 10. Tools & Technologies Used

- Languages: Python  
- Libraries: pandas, numpy, scikit-learn, seaborn, matplotlib, xgboost, tensorflow (optional)  
- IDE: Google Colab  
- Deployment (optional): gradio for web-based interface

# 11. Team Members and Contributions

- S.SAMBATH: Project leader, coordinated task distribution, model validation.  
- P.BAVIN RAJ: Provided clinical insights, verified symptom-disease mapping.  
- S.MOHAN RAM: Managed preprocessing pipeline and implemented model training logic.  
- B.PREM KUMAR: Developed deployment prototype and ensured end-to-end testing.  
- A.THAVASELVAN: Led EDA and feature visualization efforts for model interpretability.