Graduation Project Documentation – PIMA Diabetes Prediction System

1. Project Planning

1.1 Project Idea

The project aims to build an intelligent machine learning system for predicting the likelihood of diabetes using the PIMA Indians Diabetes dataset. The system integrates data preprocessing, model training, hyperparameter tuning, model tracking with MLflow, and an interactive Streamlit interface for real-time predictions.

1.2 Objectives

- Develop a reliable ML model for diabetes prediction.
- Improve accuracy through feature engineering and advanced algorithms.
- Implement MLflow for experiment tracking and deployment.
- Provide a user-friendly Streamlit interface for model interaction.
- Ensure explainability through SHAP and performance visualizations.

1.3 Project Scope

Included: - Data preprocessing, cleaning, and handling missing values. - Exploratory Data Analysis (EDA). - Model development, optimization, and evaluation. - MLflow logging & model registry. - Streamlit UI for training & prediction.

Excluded: - Medical diagnosis or clinical decision-making. - Real-time deployment on cloud services.

1.4 Milestones

1. Dataset acquisition & cleaning.
2. EDA & feature engineering.
3. Model benchmarking.
4. Hyperparameter tuning & ensemble models.
5. MLflow integration.
6. Streamlit application creation.
7. Final documentation and presentation.

1.5 Timeline

Phase	Duration	Output
Data Preparation	Week 1	Clean dataset, EDA
Model Development	Week 2	Baseline & tuned models
MLflow Integration	Week 3	Tracked experiments
UI/UX + Streamlit	Week 4	Full web interface
Testing & Deployment	Week 5	Stable release
Final Documentation	Week 6	Full project submission

2. Stakeholder Analysis

Primary Stakeholders

1. Data Scientist (Developer)
- **Role:** Build and train ML models.
- **Benefit:** Ability to track experiments, evaluate performance, and deploy models.

2. End User / Patient
- **Role:** Enters health metrics to receive a prediction.
- **Benefit:** Gets early risk awareness and insights.

3. Healthcare Professional
- **Role:** Uses predictions as supportive insight.
- **Benefit:** Helps in preliminary screening and decision support.

Secondary Stakeholders

4. Instructors / Evaluators
- **Role:** Assess the project.
- **Benefit:** Understand structure, performance, and documentation quality.

5. ML Engineers (Future Work)
- **Role:** Integrate model with real systems.
- **Benefit:** A scalable and explainable ML pipeline.

3. Database Design

3.1 ERD Overview

The system uses a simple structured dataset with the following conceptual schema:

Entities:

- **Patient** (age, pregnancies, glucose, BMI, blood pressure, etc.)
- **Prediction** (model used, probability, result)
- **Model Metadata** (experiment ID, parameters, metrics)

3.2 Database Schema

Table	Columns	Description
patients	id, pregnancies, glucose, insulin, BMI, age	Stores patient input data
predictions	id, patient_id, model_name, prediction, probability	Saves predictions for analysis
model_runs	run_id, params, accuracy, f1, auc	Stores MLflow-tracked runs / metadata

Note: In this project, data is stored as CSV/MLflow artifacts, but the schema reflects future DB implementation.

4. UI/UX Design

4.1 Design Philosophy

- Dark-theme, modern UI.
- Clean layout with component grouping.
- Minimal interaction steps for prediction.
- Clear metric visualization.

4.2 Main Screens (Mockup Description)

1. Home Page

- Project introduction.
- Dataset summary.
- Navigation to training & prediction.

2. Model Training Page

- Dataset upload or default dataset.
- Model selection.
- Hyperparameter tuning options.
- Real-time metrics panel.

3. Model Performance Dashboard

- Confusion matrix.
- ROC curve.
- Precision-Recall curve.
- SHAP explanations.
- Overfitting analysis.

4. Prediction Page

- Input form for patient medical data.
- Option to load .pkl models.
- Output result (Diabetic / Not Diabetic).
- Confidence score.

---

5. Additional Required Sections

5.1 Implementation Summary

- ML pipeline built using Python, Scikit-Learn, LightGBM, and XGBoost.
- Experiment tracking and model versions handled via MLflow.
- Final models exported as .pkl for use inside Streamlit.
- Ensemble model created using VotingClassifier for better accuracy.

5.2 Technology Stack

Component	Technology
Backend / ML	Python, scikit-learn, LightGBM, XGBoost
Tracking	MLflow
UI	Streamlit
Visualization	Matplotlib, Seaborn, Plotly
Storage	CSV + MLflow Artifacts

5.3 Challenges & Solutions

- **Data quality issues** → Cleaning + imputations.
- **Model overfitting** → Cross-validation + regularization.
- **Tracking many experiments** → MLflow integration.
- **UI performance issues** → Caching results in Streamlit.

5.4 Future Work

- Integration with real medical systems.
- Adding deep learning models.
- Mobile app version.
- Deployment on cloud platforms.

---

6. Conclusion

This project delivers a robust ML prediction system with a modern interface, strong experiment tracking, and explainable AI components. It demonstrates the full ML project lifecycle from dataset to deployment.
