Perfect — with **Streamlit confirmed as the frontend**, and hosting included — here is your **finalized Developer-Focused Blueprint Document**. This version is structured and ready to be exported as a PDF after your review.

---

# 🧠 Blueprint Document: AI-Powered Diabetes Risk Prediction System

---

## 1. 🎯 Project Overview

This is an end-to-end ML application that predicts diabetes risk using the Pima Indians Diabetes Dataset. The system includes data preprocessing, model training, explainability, a REST API, and a web-based **Streamlit frontend**, all deployable on the cloud.

---

## 2. 🧭 Goals & Objectives

**Goals:**

- Accurately predict diabetes risk.
- Build an explainable, web-accessible AI solution.
- Deliver a complete full-stack AI application using APIs and a UI.

**Objectives:**

- Train and evaluate ML models.
- Implement SHAP-based model explainability.
- Build a REST API for predictions.
- Develop and deploy a **Streamlit frontend** to interact with the API.

---

## 3. 📌 Project Scope

**In Scope:**

- Data cleaning, EDA, feature engineering
- ML model training and evaluation
- SHAP explainability integration
- REST API using FastAPI or Flask
- Streamlit frontend UI (mandatory)
- Web deployment of both API and UI

**Out of Scope:**

- User login/auth systems

- Real-time data streams
- Multi-language or mobile app support

---

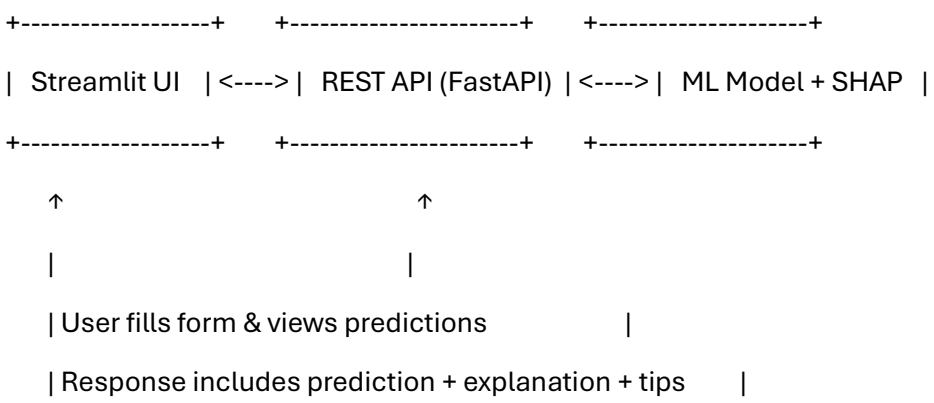
### 4. 🗓️ Timeline & Milestones

| Phase | Description | Duration | Target Dates |
|---|---|---|---|
| Phase 1: Planning | Scope, design, setup | 1 day | Day 1 |
| Phase 2: Data Processing | EDA, cleaning, transformation | 1–2 days | Day 2–3 |
| Phase 3: Modeling | Train/test models, metrics, tuning | 2 days | Day 4–5 |
| Phase 4: Explainability | SHAP/LIME integration | 1 day | Day 6 |
| Phase 5: API Development | REST API for predictions & health tips | 2 days | Day 7–8 |
| Phase 6: Frontend UI | Build Streamlit interface | 1 day | Day 9 |
| Phase 7: Deployment | Host API + Streamlit on Render/Streamlit Cloud | 1 day | Day 10 |
| Phase 8: Final Testing | Postman tests, cleanup, README, polish | 1 day | Day 11 |

**Total Estimated Duration: 11 Days**

---

### 5. 🏗️ System Architecture




---

### 6. 🧰 Technology Stack

| Component | Tool/Library |
|---|---|
| Language | Python 3.10+ |

| Component | Tool/Library |
|---|---|
| ML Libraries | scikit-learn, XGBoost, SHAP |
| API Framework | FastAPI or Flask |
| UI | **Streamlit** |
| Visualization | Matplotlib, Seaborn |
| Deployment | Render, Streamlit Cloud |
| Testing | pytest, Postman |
| Version Control | Git + GitHub |

---

**7. 🔬 ML Workflow**

1. Load and clean dataset
2. Handle zero or missing values
3. Normalize/scale features
4. Train multiple models (LogReg, RF, XGB)
5. Evaluate via Accuracy, F1, ROC-AUC
6. Save best model (joblib or pickle)
7. Apply SHAP to explain predictions

---

**8. 🌐 API Design**

**Base URL: /api/v1**

**POST /predict**

- **Input**: JSON with 8 patient health metrics
- **Output**: {"risk": "high", "probability": 0.84, "top_features": [...]}

**GET /health-tips**

- **Input**: Optional parameters (age, BMI, glucose)
- **Output**: JSON advice (e.g., "Increase fiber intake")

**POST /explain**

- **Input**: Same as /predict
- **Output**: SHAP value summary

---

### 9. ✅ Feature List (Frozen Scope)

| Feature | Status |
|---|---|
| Data preprocessing & EDA | ✔ |
| Model training & evaluation | ✔ |
| SHAP integration | ✔ |
| REST API with 3 endpoints | ✔ |
| **Streamlit web interface** | ✔ |
| API + UI Hosting | ✔ |
| Testing & Documentation | ✔ |

🚫 **No new features will be added** after this list is locked.

---

### 10. ⚙ Assumptions & Constraints

- Dataset is stable and not changing
- Model doesn't require real-time training
- Hosting is done on free-tier platforms (limited resources)
- UI is for demo purposes, not production-grade UX

---

### 11. 📦 Deliverables

- notebooks/: EDA, modeling, evaluation
- api/: Python scripts for REST API
- frontend/: Streamlit app
- models/: Trained model files
- reports/: Final PDF blueprint + optional prediction reports
- requirements.txt: Full dependency list
- README.md: Project overview and setup instructions
- 📎 Deployed App URL(s)

---

### 12. ☁ Deployment Plan

| Component | Platform | Notes |
|---|---|---|
| API | Render.com | Deploy via Gunicorn + FastAPI |
| Frontend | Streamlit Cloud | Deploy directly from GitHub repo |
| Optional Bundle | Docker | Unified API + frontend (optional) |

---

### 13. 🚧 Risks & Mitigation

| Risk | Mitigation |
|---|---|
| SHAP performance on large data | Sample data or use summary SHAP |
| Streamlit load times | Optimize model load & SHAP caching |
| Deployment issues | Use standard deployment guides |
| Over-scoping | Scope locked to avoid creep |

---

### 14. 🧪 Success Criteria

- Model achieves ≥80% accuracy or ROC-AUC
- REST API responds to all planned routes
- Streamlit app works end-to-end with live API
- Project is fully documented and hosted live

---

### 15. 🔮 Future Enhancements (Post-project Only)

- Voice input via speech-to-text
- User authentication + session tracking
- Expand to time-series modeling
- Email PDF reports to users
- Mobile responsive UI (Streamlit + CSS hacks)

---