**UI and how are we fitting word document**

**Mermaid kaa flow diagram on ui and drop and it will grow**

**To figure out how we will genreate index – hard code it**

**1. Project Overview**

* **Project Name**:
* **Author(s)**:
* **Last Updated**:
* **Version**:
* **Status**: Draft / In Progress / Complete
* **Repository**: [Link if applicable]

**Summary**:

A 2–4 sentence high-level summary of the project, its purpose, and scope.

**2. Objective & Scope – readme, doc string, and if nothing then then we will ask user to give some context**

* **Goals**:
  + What the system is designed to achieve
* **Out of Scope** (if any):
  + Clear exclusions

**3. Architecture Overview – code**

* **System Architecture Diagram**

Insert or link a diagram showing key components and their interaction.

* **Key Components**:

| **Component** | **Description** | **Tech Stack** |
| --- | --- | --- |
| e.g., Backend API | Handles business logic | Python + FastAPI |
| Model Training | ML pipeline | PyTorch + MLflow |
| UI Frontend | User Interface | React + Tailwind |

**4. Tech Stack - code**

* **Languages**: Python 3.10, JavaScript
* **Frameworks**: FastAPI, PyTorch, React
* **Tools**: Docker, GitHub Actions, MLflow, Streamlit
* **Databases**: PostgreSQL, Redis
* **Cloud Services**: GCP / AWS / Azure (specify)

**5. Installation & Setup - code**

**Local Setup:**

bash

CopyEdit

git clone <repo-url>

cd project-name

pip install -r requirements.txt

**Environment Variables:**

| **Variable** | **Description** |
| --- | --- |
| API\_KEY | API Key for external service |
| DB\_URL | PostgreSQL connection string |

**Docker:**

bash

CopyEdit

docker-compose up –build

**8. API Documentation – code or readme**

Use Swagger / Postman / Redoc if applicable.

| **Endpoint** | **Method** | **Description** |
| --- | --- | --- |
| /predict | POST | Predict output |
| /health | GET | Check service health |

**6. Usage Instructions**

* **Run the API**: python main.py