Got it ✅ You already have the **README** and now the **requirements.txt**.  
Next, I can create **developer-friendly documentation** in the same style — clear sections, practical, and directly tied to your code.

Here’s a good structure we can use for your project documentation (DOCUMENTATION.md):

**📖 Project Documentation – AI-Powered Natural Language Reporting and Charting Tool**

**1. Introduction**

This tool allows business users to interact with relational databases using **natural language queries**. Instead of writing SQL, users can ask questions in plain English (or via voice). The system translates queries into SQL, fetches results, and generates **interactive visualizations and summaries**.

**2. System Overview**

**🔹 Frontend (app\_gradio.py)**

* Built with **Gradio**
* Responsibilities:
  + Capture **user input** (text or voice)
  + Call backend API (/query)
  + Render **tables, charts, and summaries**
  + Allow query refinement & export results

**🔹 Backend (backend.py)**

* Built with **FastAPI + LangChain + OpenAI**
* Responsibilities:
  + Convert natural language → **SQL** using LangChain & GPT
  + Execute SQL against MySQL database
  + Generate **natural language summaries** of query results
  + Suggest/create **visualizations** using AI (matplotlib)

**3. Data Flow**

flowchart TD

A[User Query in NL] -->|Gradio| B[Frontend app\_gradio.py]

B -->|POST /query| C[Backend backend.py]

C --> D[LangChain + OpenAI - NL → SQL]

D --> E[Execute SQL on MySQL DB]

E --> F[Return Results]

F --> G[AI Summary + Visualization]

G --> B

B --> H[Interactive Charts, Tables, Exports]

**4. Features**

* ✅ **Natural Language to SQL** – Users type questions like *"Show me top 5 products by revenue"*
* ✅ **Voice Queries** – Speak instead of typing
* ✅ **Interactive Visualization** – Auto-generate bar, line, pie, scatter charts
* ✅ **AI Summaries** – Results explained in natural language
* ✅ **Export Options** – Download as CSV, Excel, or PDF
* ✅ **Query Refinement** – Add filters to previous queries
* ✅ **Query History** – Track past interactions

**5. API Documentation**

**POST /query**

**Description:** Converts natural language to SQL, executes query, and returns results.

**Request Body:**

{

"natural\_language\_text": "Show me the sales trend for 2025"

}

**Response:**

{

"query": "SELECT year, SUM(sales) FROM Sales WHERE year=2025 GROUP BY year;",

"results": [

{"year": 2025, "SUM(sales)": 1200000}

],

"natural\_language\_response": "In 2025, the total sales amounted to 1.2 million.",

"visualization\_data": "<base64 encoded chart image>"

}

**6. Setup Guide**

1️⃣ **Backend Setup**

python backend.py

Runs on → http://localhost:5000

2️⃣ **Frontend Setup**

python app\_gradio.py

Runs on → http://localhost:7860

3️⃣ **Database Setup**

* Ensure MySQL is running
* Update backend.py → db\_config with your credentials
* Load sample data into task1 database

**7. Example Usage**

* **Text Query:**  
  "Compare revenue by region for the last 3 years"
* **Voice Query:**  
  Speak: *"Show me clients who traded options"*
* **Refine Query:**  
  "only for New York region"

**8. Future Enhancements**

* Multi-database support (SQL Server, PostgreSQL, Oracle)
* Role-based access & authentication
* Dashboard saving & sharing
* Real-time data streaming

⚡ This documentation serves as a **developer & user guide** for your project.

👉 Do you want me to **write this as a DOCUMENTATION.md file** (so it sits next to your README), or should I **merge it into your README** as an extended section?