Let’s break it down step by step. Here’s how we can structure the project:

Project Structure

```
titanic-chatbot/
□ □ □backend/              # FastAPI backend
□  □ □ □main.py            # API setup
□  □ □ □chatbot.py         # LangChain logic
□  □ □ □data_processing.py  # Load and process Titanic dataset
□  □ □ □requirements.txt    # Dependencies
□
□ □ □frontend/             # Streamlit frontend
□  □ □ □app.py             # Streamlit UI
□  □ □ □requirements.txt    # Dependencies
□
□ □ □dataset/              # Titanic CSV file
□  □ □ □titanic.csv
```

```
README.md              # Project documentation
.gitignore        # Ignore unnecessary files
```

Step 1: Setting Up FastAPI Backend

We’ll create a FastAPI server that loads the Titanic dataset and processes user queries.

backend/main.py

```
from fastapi import FastAPI
from chatbot import process_query

app = FastAPI()

@app.get("/")
```

```
def home():
  return {"message": "Titanic Chatbot API is running"}

@app.get("/query/")
def query(q: str):
  return {"response": process_query(q)}
```

backend/chatbot.py

```
import pandas as pd

df = pd.read_csv("dataset/titanic.csv")

def process_query(q):
  q = q.lower()

  if "percentage of passengers were male" in q:
    male_percentage = (df['Sex'] == 'male').mean() * 100

    return f"{male_percentage:.2f}% of passengers were male."


  elif "average ticket fare" in q:

    avg_fare = df['Fare'].mean()

    return f"The average ticket fare was ${avg_fare:.2f}."


  elif "how many passengers embarked" in q:

    embark_counts = df['Embarked'].value_counts().to_dict()

    return embark_counts


  else:

    return "I don't understand that question."
```

Step 2: Building Streamlit Frontend

frontend/app.py

```
import streamlit as st

import requests


st.title("Titanic Chatbot ▯ ")


query = st.text_input("Ask a question about the Titanic dataset:")


if st.button("Submit"):

  response = requests.get(f"http://127.0.0.1:8000/query/?q={query}").json()

  st.write(response["response"])
```

Step 3: Running the Project

1. Install dependencies:

```
pip install fastapi uvicorn pandas streamlit requests
```

2. Start the FastAPI server:

```
cd backend

uvicorn main:app –reload
```

3. Start Streamlit frontend:

```
cd frontend

streamlit run app.py
```