Question Answering System - Project Documentation

**Project Structure**  
qa\_app/  
│  
├ app.py # FastAPI app entry point  
├ model\_utils.py # Utility functions to load model and get answers  
├requirements.txt # Dependencies  
├qa\_app/  
│ └── model/ # Trained QA model (config, tokenizer, pytorch\_model)  
├data/  
│ └── squad\_sample.json # (Optional) Sample dataset used for evaluation  
├ README.md # Project overview and instructions

# Project Objective

The goal of this project is to:  
- Fine-tune a pre-trained QA model on a subset of the SQuAD dataset  
- Evaluate the model's performance  
- Deploy it using FastAPI for real-time question answering

# Steps Overview

# 1. Environment Setup

python -m venv qa  
source qa/bin/activate # On Windows: qa\Scripts\activate  
pip install -r requirements.txt

**2.Model Training**

- Model: distilbert-base-uncased  
- Dataset: SQuAD (subset)  
- Trainer API from Hugging Face was used for training with reduced dataset to speed up experimentation.  
  
Training Output:  
 TrainOutput(global\_step=514, training\_loss=2.54, ...)

**3.Evaluation**

- Metric: squad\_v2 (F1 and Exact Match)  
- Key Results:  
 - F1 Score: 30.26  
 - Exact Match: 26.0  
 - HasAns\_f1: 67.25  
 - NoAns\_f1: 0.0

**4.FastAPI Deployment**

Download Model: Place your fine-tuned model inside:  
.\qa\_app\model

Sample model\_utils.py:

from transformers import pipeline  
  
qa\_pipeline = pipeline("question-answering", model="qa\_app/model", tokenizer="qa\_app/model")  
  
def get\_answer(question: str, context: str) -> str:  
 result = qa\_pipeline(question=question, context=context)  
 return result['answer']

Sample app.py:

from fastapi import FastAPI  
from pydantic import BaseModel  
from model\_utils import get\_answer  
  
app = FastAPI()  
  
class QARequest(BaseModel):  
 question: str  
 context: str  
  
@app.post("/qa/")  
def qa\_endpoint(payload: QARequest):  
 answer = get\_answer(payload.question, payload.context)  
 return {"answer": answer}

Run Server:  
uvicorn app:app –reload

**Example API Request**

POST /qa/  
  
Request:  
{  
 "question": "Who wrote the theory of relativity?",  
 "context": "Albert Einstein developed the theory of relativity in the early 20th century."  
}  
  
Response:  
{  
 "answer": "Albert Einstein"  
}

**Future Improvements**

- Handle unanswerable questions better (improve NoAns metrics)  
- Add streaming capability for real-time datasets  
- Include Swagger/OpenAPI docs (FastAPI auto-generates this)