AI Implementation: With API vs Without API

This document explains the difference between writing pure AI logic and exposing that same AI logic through an API so it can be used by Node.js backends, frontends, or other services.

1. Core AI Logic (Same in Both Cases)

The AI logic is written as a normal Python function. This function does the actual prediction and remains unchanged whether or not an API is used.

```
# ai_model.py

def predict_sentiment(text: str) -> dict:
    text = text.lower()

    if "good" in text or "awesome" in text:
        return {"sentiment": "positive", "score": 0.9}
    elif "bad" in text or "worst" in text:
        return {"sentiment": "negative", "score": 0.1}
    else:
        return {"sentiment": "neutral", "score": 0.5}
```

2. Using AI Without an API

In this case, the AI logic is used directly inside Python. There is no HTTP layer, so external systems like Node.js or frontend apps cannot access this logic.

```
# run_local.py
from ai_model import predict_sentiment

result = predict_sentiment("This product is awesome")
print(result)
```

Limitation: This approach is suitable only for local testing or research. It cannot be consumed by other services.

3. Using AI With an API (Production Approach)

To make AI usable by a backend or frontend, the same function is wrapped inside an API using frameworks like FastAPI or Flask.

```
# main.py (FastAPI)
from fastapi import FastAPI
from pydantic import BaseModel
from ai_model import predict_sentiment

app = FastAPI()

class PredictRequest(BaseModel):
    text: str

@app.post("/predict")
def predict_api(payload: PredictRequest):
    return predict_sentiment(payload.text)
```

4. How Node.js Uses This AI API

Once the API exists, Node.js can communicate with the AI service using HTTP requests.

```
// Node.js example
import axios from "axios";

const response = await axios.post("http://localhost:8000/predict", {
  text: "awesome service"
});

console.log(response.data);
```

5. Summary

- AI logic and API are separate concerns. - The AI function remains the same in both approaches. - An API is mandatory for integration with Node.js or frontend applications. - Without an API, AI code is not accessible outside Python.