

```

from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from dotenv import load_dotenv
from supabase import create_client
import os
import httpx

load_dotenv()

app = FastAPI(title="AI Receptionist Backend")

# Environment variables
OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")
SUPABASE_URL = os.getenv("SUPABASE_URL")
SUPABASE_SERVICE_KEY = os.getenv("SUPABASE_SERVICE_KEY")

supabase = create_client(SUPABASE_URL, SUPABASE_SERVICE_KEY)

class ChatRequest(BaseModel):
    message: str
    tenant_id: str

@app.get("/health")
async def health():
    return {"ok": True, "message": "Backend is live"}

@app.post("/chat")
async def chat(req: ChatRequest):
    # Get company details from Supabase
    result = (
        supabase.table("companies")
        .select("*")
        .eq("id", req.tenant_id)
        .single()
        .execute()
    )

    company = result.data
    if not company:
        raise HTTPException(status_code=404, detail="Company not found")

    tone = company.get("tone", "friendly and professional")
    greeting = company.get("greeting", "Welcome in! How can I help you today?")

```

```

system_prompt = (
    f"You are the AI receptionist for {company['name']}. "
    f"Use a {tone} tone. Greet users with '{greeting}'. "
    f"Answer questions, handle check-ins, and gather names or contact info when relevant."
)

```

Send to OpenAI

async with httpx.AsyncClient(timeout=30.0) as client:

```

    r = await client.post(
        "https://api.openai.com/v1/chat/completions",
        headers={
            "Authorization": f"Bearer {OPENAI_API_KEY}",
            "Content-Type": "application/json",
        },
        json={
            "model": "gpt-5",
            "messages": [
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": req.message},
            ],
            "temperature": 0.5,
        },
    )
    r.raise_for_status()
    data = r.json()

```

```

reply = data["choices"][0]["message"]["content"]

```

```

return {"reply": reply, "company": company}

```