Chatbox Backend (FastAPI)

Final Polished Code (without SQLAlchemy)

```
from fastapi import FastAPI, Depends, HTTPException, status, UploadFile, File
from fastapi.middleware.cors import CORSMiddleware
from fastapi.security import OAuth2PasswordBearer, OAuth2PasswordRequestForm
from pydantic import BaseModel
from typing import Optional
from datetime import datetime, timedelta
from jose import JWTError, jwt
from dotenv import load_dotenv
import os, shutil, PyPDF2
from openai import OpenAI

# ==============================
# Load environment variables
# ==============================
load_dotenv()

SECRET_KEY = os.getenv("SECRET_KEY", "supersecretkey")  # fallback
ALGORITHM = "HS256"
ACCESS_TOKEN_EXPIRE_MINUTES = 30
OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")

# Debugging
print("Using OPENAI_API_KEY:", OPENAI_API_KEY[:10], "..." if OPENAI_API_KEY else "■ NOT FOUND")

if not OPENAI_API_KEY:
    raise RuntimeError("■ OPENAI_API_KEY is missing. Please set it in your .env file.")

client = OpenAI(api_key=OPENAI_API_KEY)

# ==============================
# FastAPI App Setup
# ==============================
app = FastAPI()

app.add_middleware(
    CORSMiddleware,
    allow_origins=["http://localhost:5173"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

oauth2_scheme = OAuth2PasswordBearer(tokenUrl="token")

# ==============================
# In-Memory Storage
# ==============================
users_db = {}
projects_db = {}
project_counter = 1

# Schemas
class RegisterRequest(BaseModel):
    email: str
    password: str

class ProjectRequest(BaseModel):
    name: str
    description: Optional[str] = ""

class ProjectUpdateRequest(BaseModel):
    name: Optional[str] = None
    description: Optional[str] = None

class PromptRequest(BaseModel):
    title: str
    content: str

class ChatRequest(BaseModel):
    project_id: int
    message: str
    reset: Optional[bool] = False

# ==============================
# Auth Helpers
```