```
import { NextResponse } from "next/server";
import simpleGit from "simple-git";
import fs from "fs";
import path from "path";
import OpenAI from "openai";
import { readProject } from "@/lib/readFiles";

export async function POST(req) {
  const { repo, prompt, systemPrompt } = await req.json();

  const tempDir = "/tmp/ai-project";
  const git = simpleGit();
  const openai = new OpenAI({
    apiKey: process.env.OPENAI_KEY
  });

  try {
    if (fs.existsSync(tempDir)) {
      fs.rmSync(tempDir, { recursive: true });
    }

    await git.clone(repo, tempDir);

    const files = readProject(tempDir);

    const completion = await openai.chat.completions.create({
      model: "gpt-4o-mini",
      messages: [
        { role: "system", content: systemPrompt },
        {
          role: "user",
          content: `
Arquivos do projeto:
${JSON.stringify(files)}

Pedido:
${prompt}

Responde APENAS com JSON no formato:
[
  { "file": "caminho/arquivo.js", "content": "novo código" }
```

```
]
`
      }
    ]
  });

  const changes = JSON.parse(
    completion.choices[0].message.content
  );

  for (const change of changes) {
    fs.writeFileSync(
      path.join(tempDir, change.file),
      change.content
    );
  }

  await git.cwd(tempDir);
  await git.add(".");
  await git.commit("Atualização automática via IA");
  await git.push(
    "https://x-access-token:" +
      process.env.GITHUB_TOKEN +
      "@github.com/" +
      repo.split("github.com/")[1]
  );

  return NextResponse.json({
    message: "✅ Projeto atualizado com sucesso!"
  });

 } catch (error) {
  console.error(error);
  return NextResponse.json(
    { message: "❌ Erro ao atualizar o projeto" },
    { status: 500 }
  );
 }
}
```