

PALO IT 18 JULY 2024



Advance Prompt Engineering with GitHub Copilot Workshop

By PALO IT

Write Prompt with GitHub Copilot

PALO^{IT}

A **Global** Leading Digital Innovation Consultancy



What We Do ?

PALO^{IT}

PRODUCT CONSULTING

1 New Digital Product Innovation

- Validate Product-Market Fit
- Formulate Implementation Plan
- Minimize Development Cost

2 Existing Product Optimization

- Increase Product Adoption
- Identify Product Improvements
- Improve Viability and Profitability

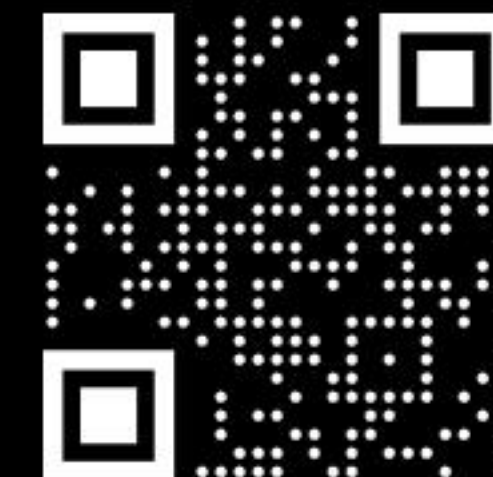
ORGANIZATION CONSULTING

3 Delivery Acceleration

- Reduce Time to Market
- Enable Value-Based Delivery
- Modernize Tech Infrastructure
- Align Business and Product Delivery
- Staff Empowerment

4 Sustainability For Positive Impact

- Sustainability Fundamentals
- ESG Data Management
- Enabling An Impact Mindset
- Tech For Good



Contact Us

Scan to receive complimentary course
with PALO IT



Who are we?

- ปาโก้ Tech Lead
- แจ็ค Fullstack Developer
- ชั่น Fullstack Developer
- เต๋ย Fullstack Developer
- ไม้ Fullstack Developer

Introduction

Agenda

- Prompt Engineering Techniques
 - X-Shots prompting(zero, one, few, many) or In-context learning - เทคนิคพื้นฐานสำหรับ prompt engineering
 - ReAct - เทคนิคพื้นฐานในการเขียน prompt สำหรับ LLM Agent
- Hands-on Exercise
 - Text Extractor
 - ReAct Phase 1
 - ReAct Phase 2
 - ReAct Final Boss

X-Shots Prompting(In Context Learning)

Zero-shot

PALO IT One Shot ReAct

Hello, human!

Switch vocal and vowel letters Input: Gale

If we switch the vocal and vowel letters in "Gale", we get "Lage".

Switch vocal and vowel letters Input: Gale Output: JSON format

Here is the output in JSON format with switched vocal and vowel letters:

```
{
  "input": "Gale",
  "output": "Gilea"
}
```

Explanation:
In the input "Gale", the vowel is "a" and the consonant is "Gle".
To switch the vocal and vowel letters, I replaced the vowel "a" with the consonant "i" and vice versa, resulting in "Gilea".

One-shot

Switch vocal and vowel letters Input: Gale Output: ``json { "explanation": "In the input "Gale", the vowel is "a" and the consonant is "Gle". To switch the vocal and vowel letters, I replaced the vowel "a" with the consonant "i" and vice versa, resulting in "Gilea"." "input": "Gale", "output": "Gilea" }
Input: Palo
Output:

```
{
  "explanation": "In the input \"Palo\", the vowel is \"a\" and the consonant is \"P\". To switch the vocal and vowel letters, I replaced the vowel \"a\" with the consonant \"i\" and vice versa, resulting in \"Poola\".",
  "input": "Palo",
  "output": "Poola"
}
```


ReAct (Reasoning and Acting)

ReAct (Reasoning and Acting) คือ งานวิจัยแนว LLM Agent ที่แนะนำว่า เราควรจะให้โมเดลให้เหตุผลก่อนสั่งให้ทำงาน

LLM Agent คือ LLM model system ที่ติดต่อ หรือลงมือทำงานในโลก ภายนอกได้ เช่น เรียก api หรือ บังคับ แขนกลหุ่นยนต์

ReAct from Paper

Credit: <https://github.com/ysymyth/ReAct/blob/master/hotpotqa.ipynb>

Solve a question answering task with interleaving Thought, Action, Observation steps. Thought can reason about the current situation, and Action can be three types:

- (1) Search[entity], which searches the exact entity on Wikipedia and returns the first paragraph if it exists. If not, it will return some similar entities to search.
- (2) Lookup[keyword], which returns the next sentence containing keyword in the current passage.
- (3) Finish[answer], which returns the answer and finishes the task.

Here are some examples.

Workshop Technical Setup

Workshop Technical Setup

- Download LLM-Bash-Agent?
- Run LLM-Bash-Agent? npm install?
- Groq key?
- Install Github Copilot?

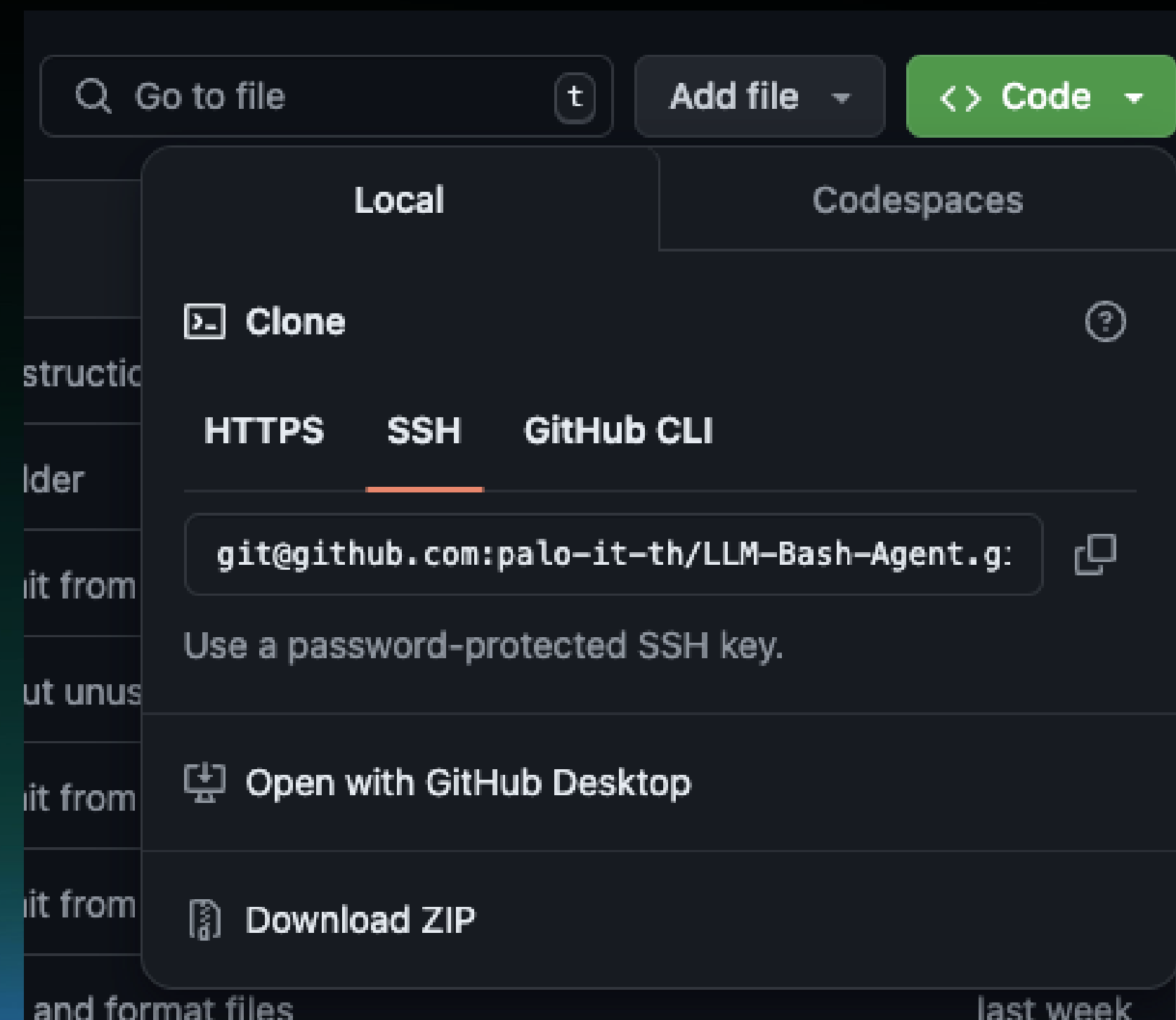
GitHub Repo

1 Open LLM-Bash-Agent repo

github.com/palo-it-th/LLM-Bash-Agent



2 Clone or download ZIP file



GitHub Repo

3 Install Node and NPM with these versions

```
~/Work/LLM-Bash-Agent git:(main) (0.092s)  
node -v  
v21.7.2
```

```
~/Work/LLM-Bash-Agent git:(main) (0.289s)  
npm -v  
10.5.0
```

4 Go to the folder and install dependencies

```
~/Work/LLM-Bash-Agent git:(main) (1.362s)
```

```
npm i
```

```
up to date, audited 733 packages in 1s
```

```
261 packages are looking for funding  
run `npm fund` for details
```

```
found 0 vulnerabilities
```


Groq API Key

https://console.groq.com/

API Keys

Manage your API keys. Remember to keep your API keys safe to prevent unauthorized access.

NAME	SECRET KEY	CREATED	LAST USED
		7/1/2024	7/8/2024

Create API Key

1

Create API Key

Enter a display name for the key

2

Submit

Create API Key

Your new API key has been created. **Copy it now, as we will not display it again.**

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

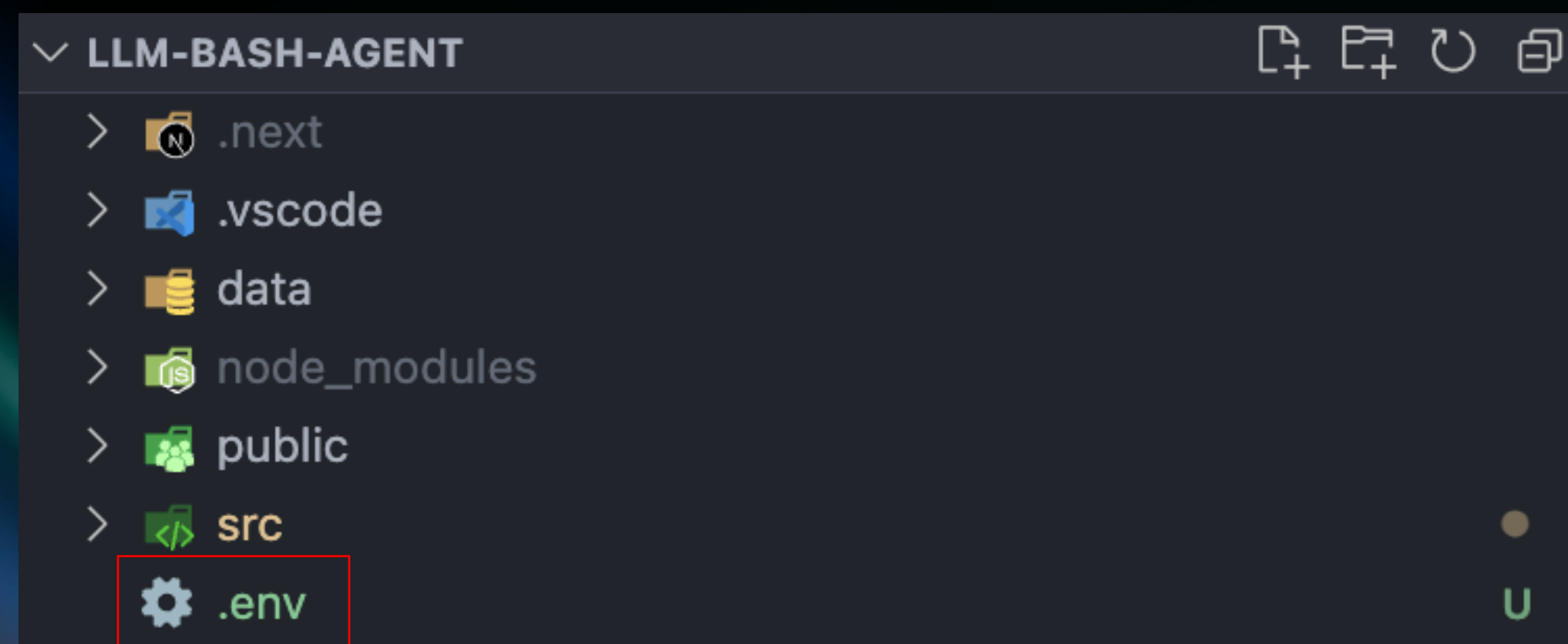
Copy

3

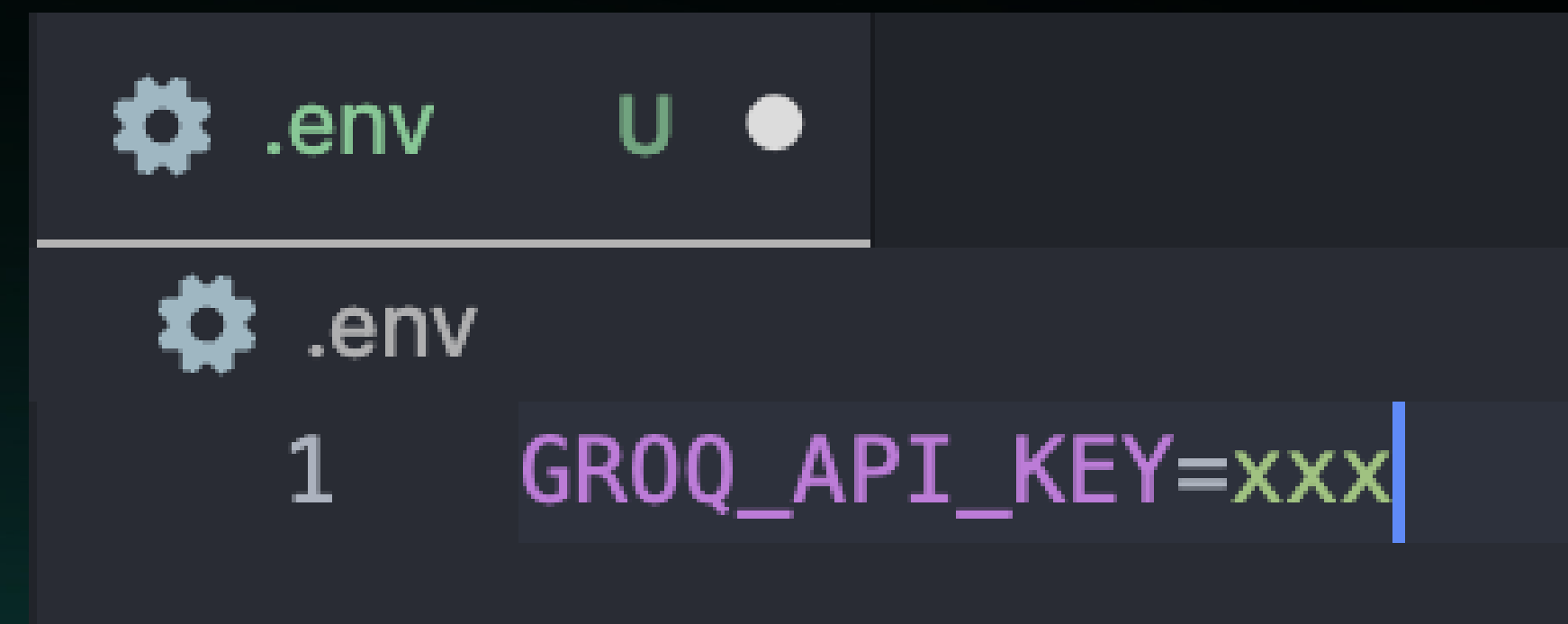
Done

Environment setup

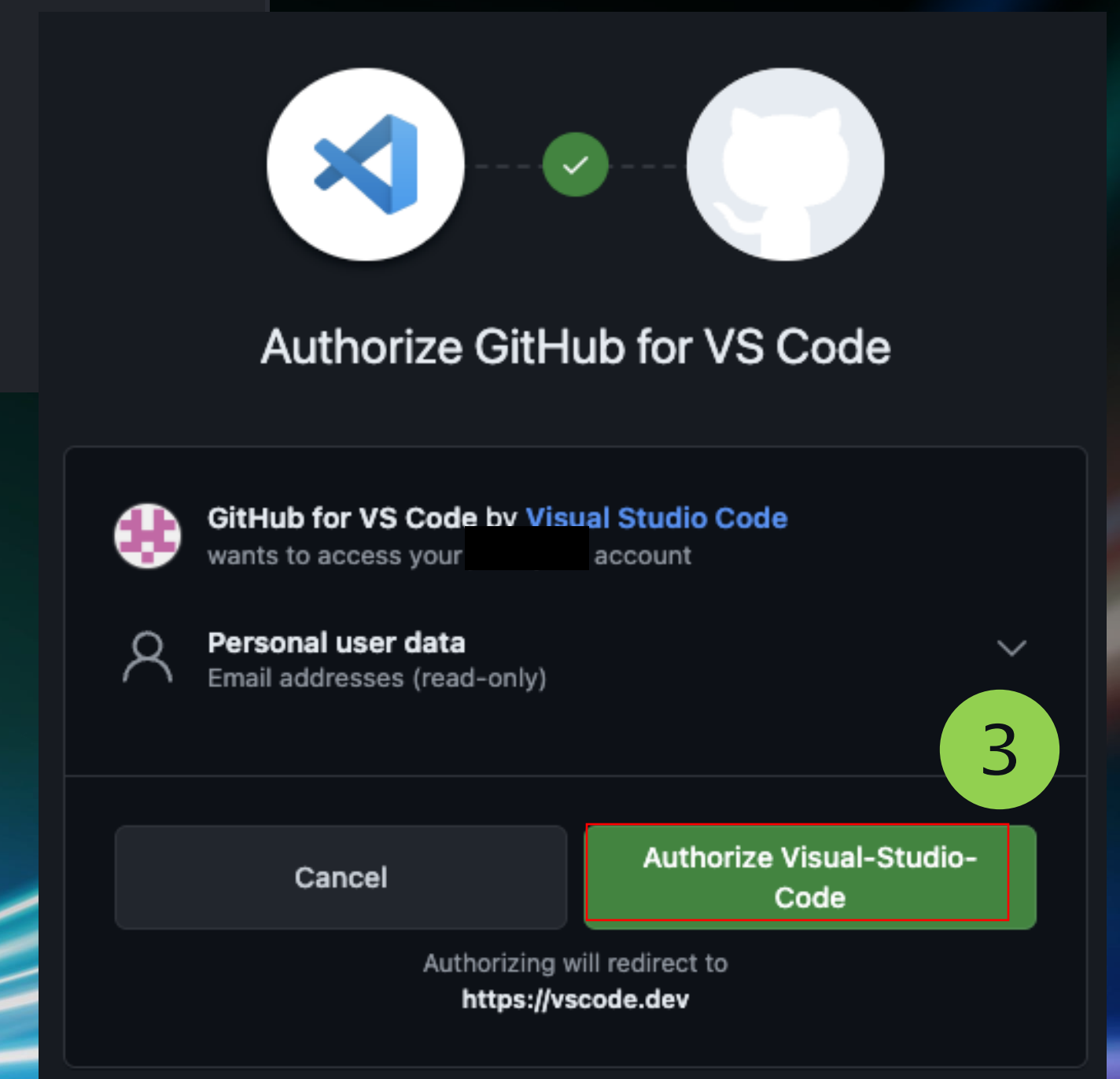
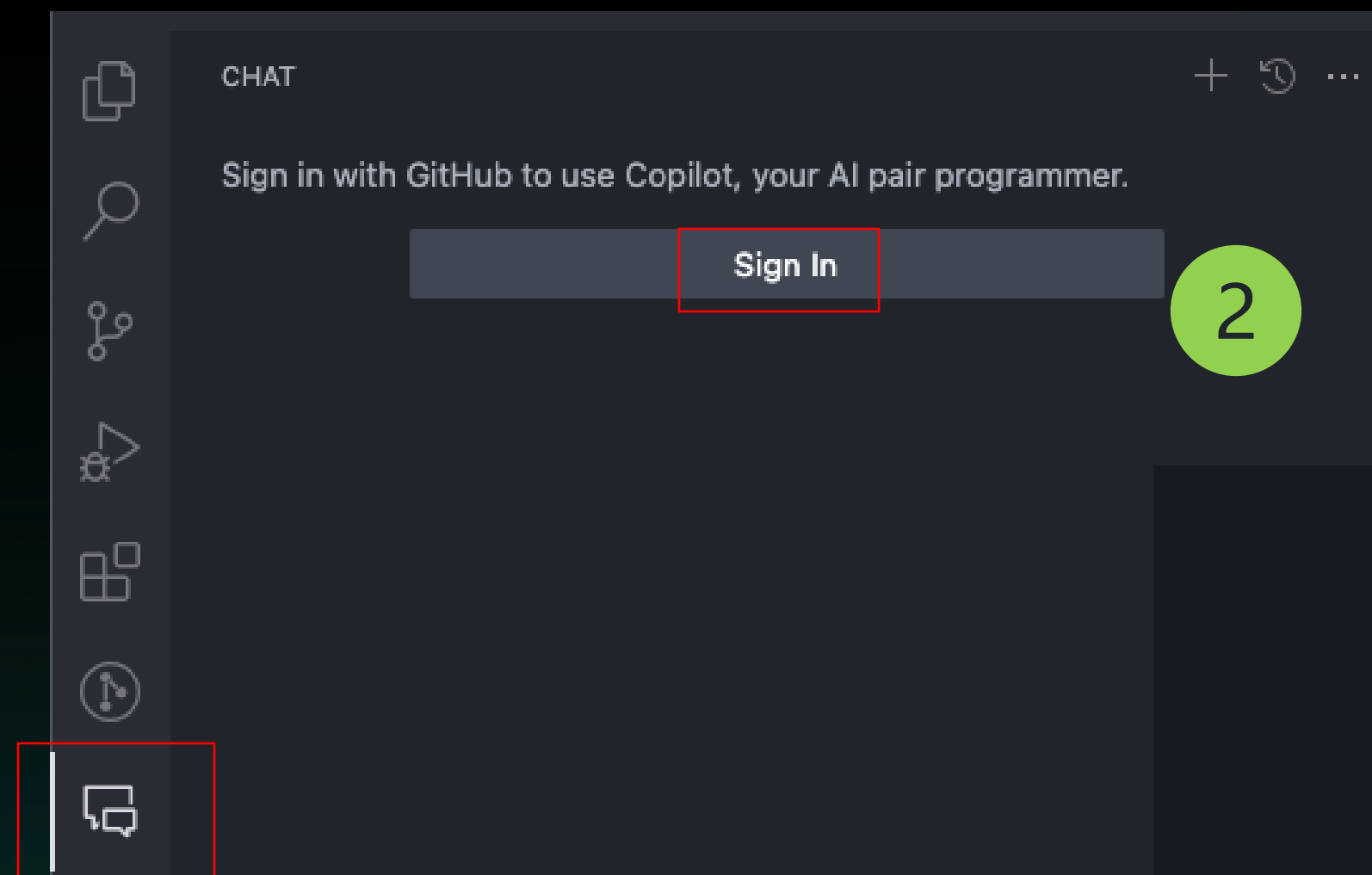
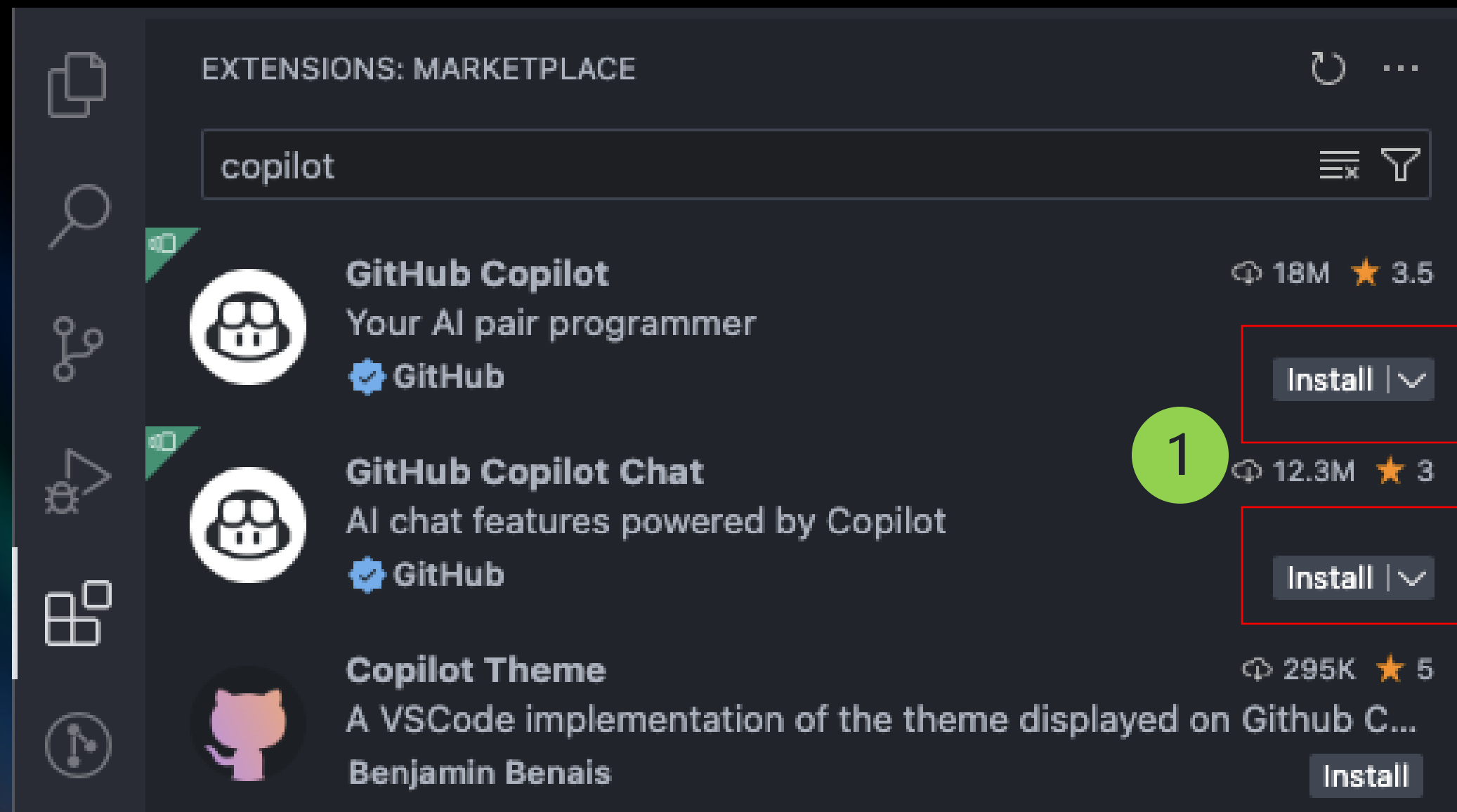
5 Create .env file at project root directory



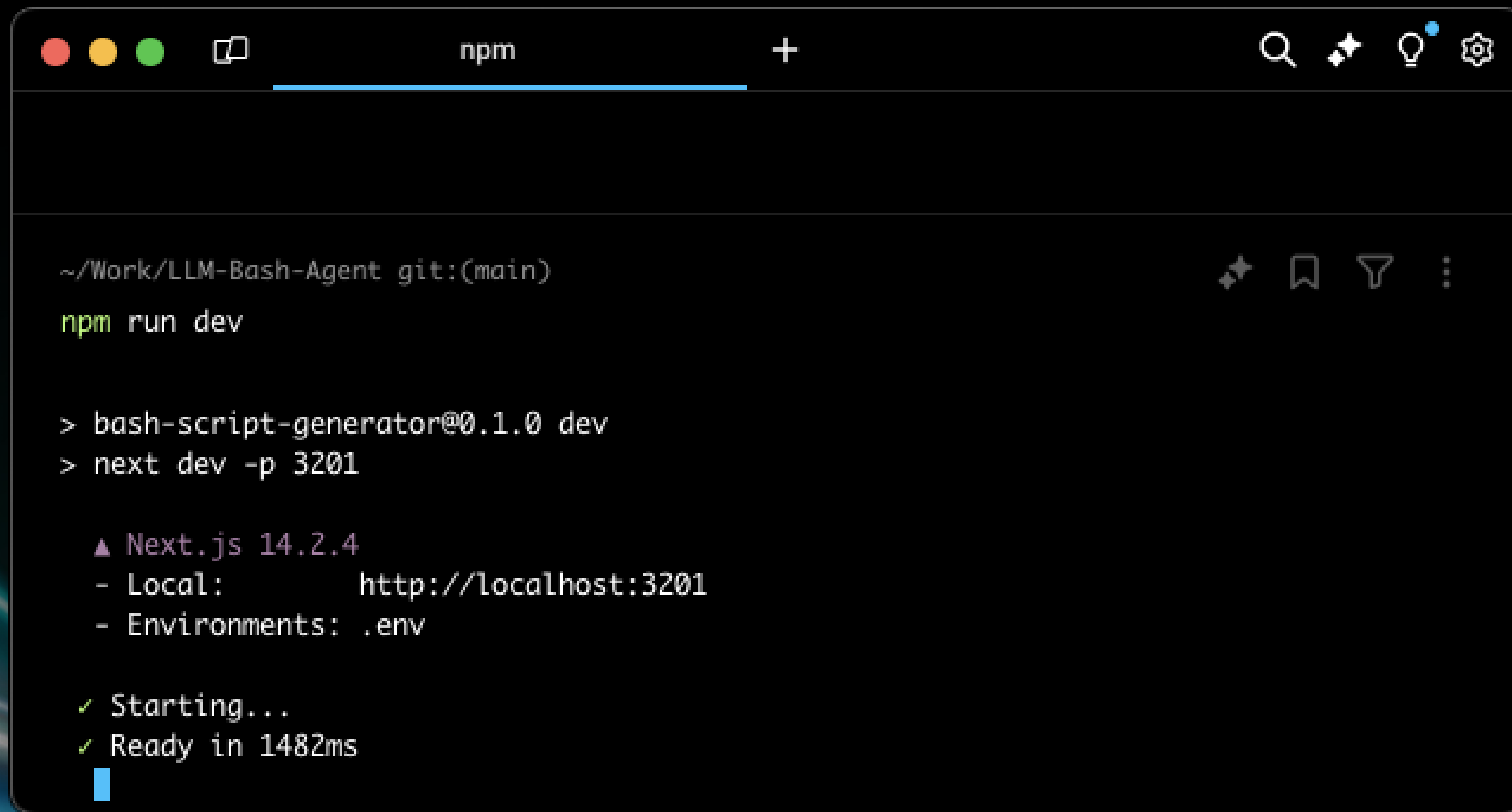
6 Paste Groq API key



Github Copilot on VS Code



Running the project



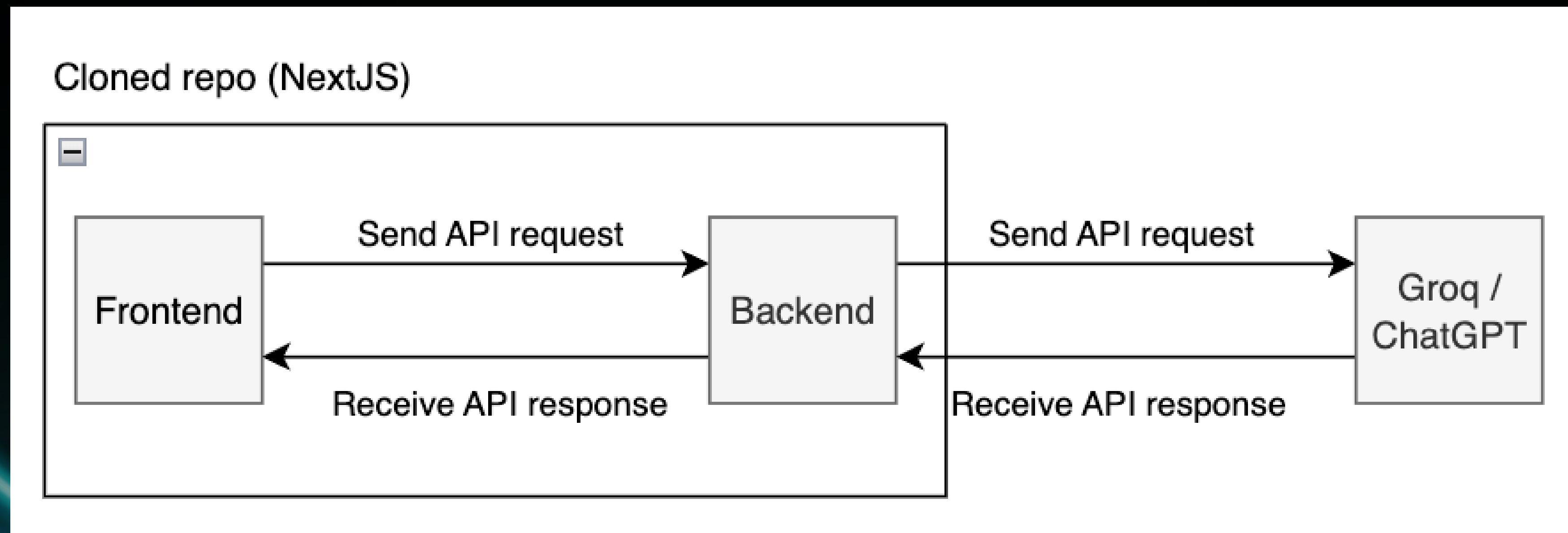
```
~/Work/LLM-Bash-Agent git:(main)
npm run dev

> bash-script-generator@0.1.0 dev
> next dev -p 3201

  ▲ Next.js 14.2.4
  - Local:      http://localhost:3201
  - Environments: .env

✓ Starting...
✓ Ready in 1482ms
█
```


LLM-Bash-Agent Basic Architecture 1



Q & A

Hands-On Lab

Github Copilot for Prompt Engineering

- Generate example for few shot prompt
- Test prompt effectiveness

One-shot Prompt v1: Text Extractor

One-shot Prompt V1 Step 1

เปิด vscode เพื่อทำโจทย์นี้กัน

One-shot Prompt V1 Step 2

โจทย์: ให้ extract ข้อมูลจากข้อความและ output ออกมาเป็น format(div+json)

ข้อมูลที่ต้องการ extract คือ Job Title, Company, Location in JSON format inside <div> tag

<div>

...JSON

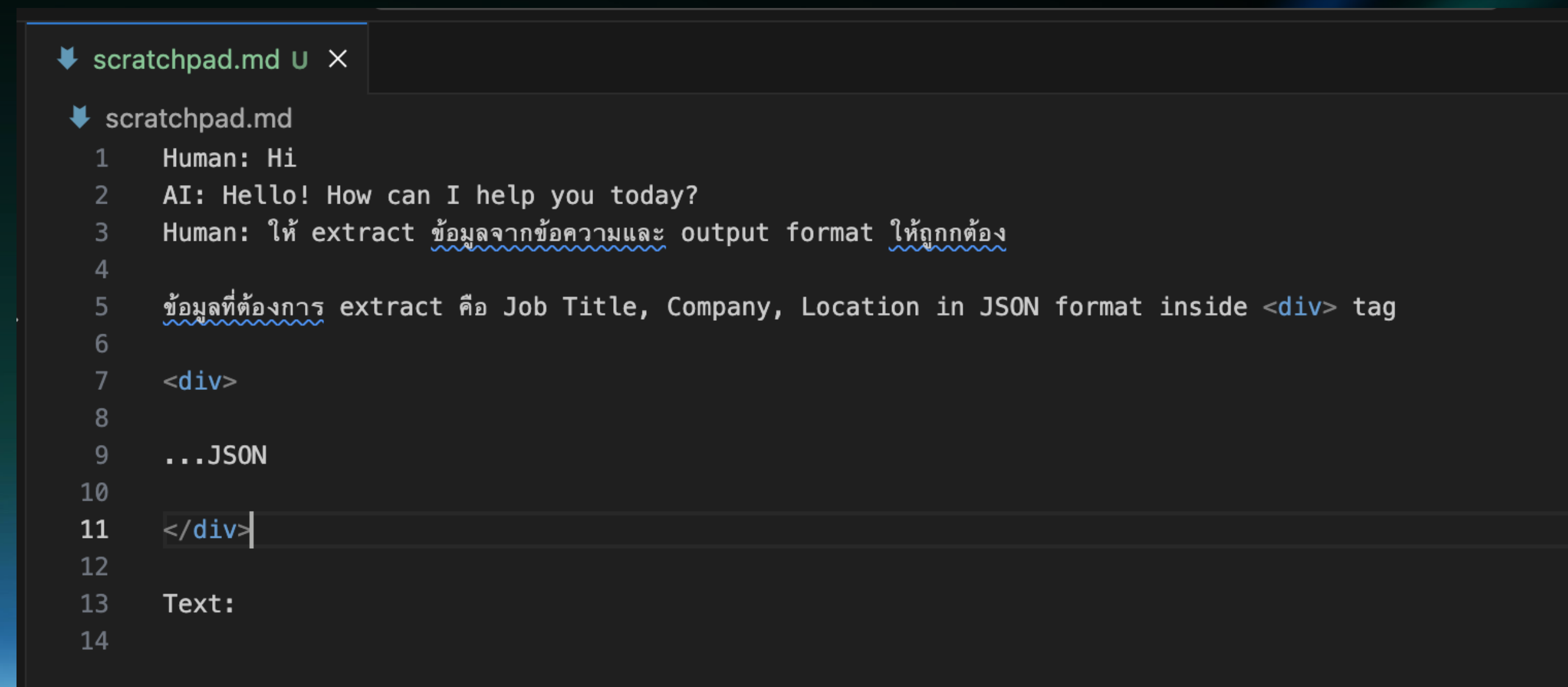
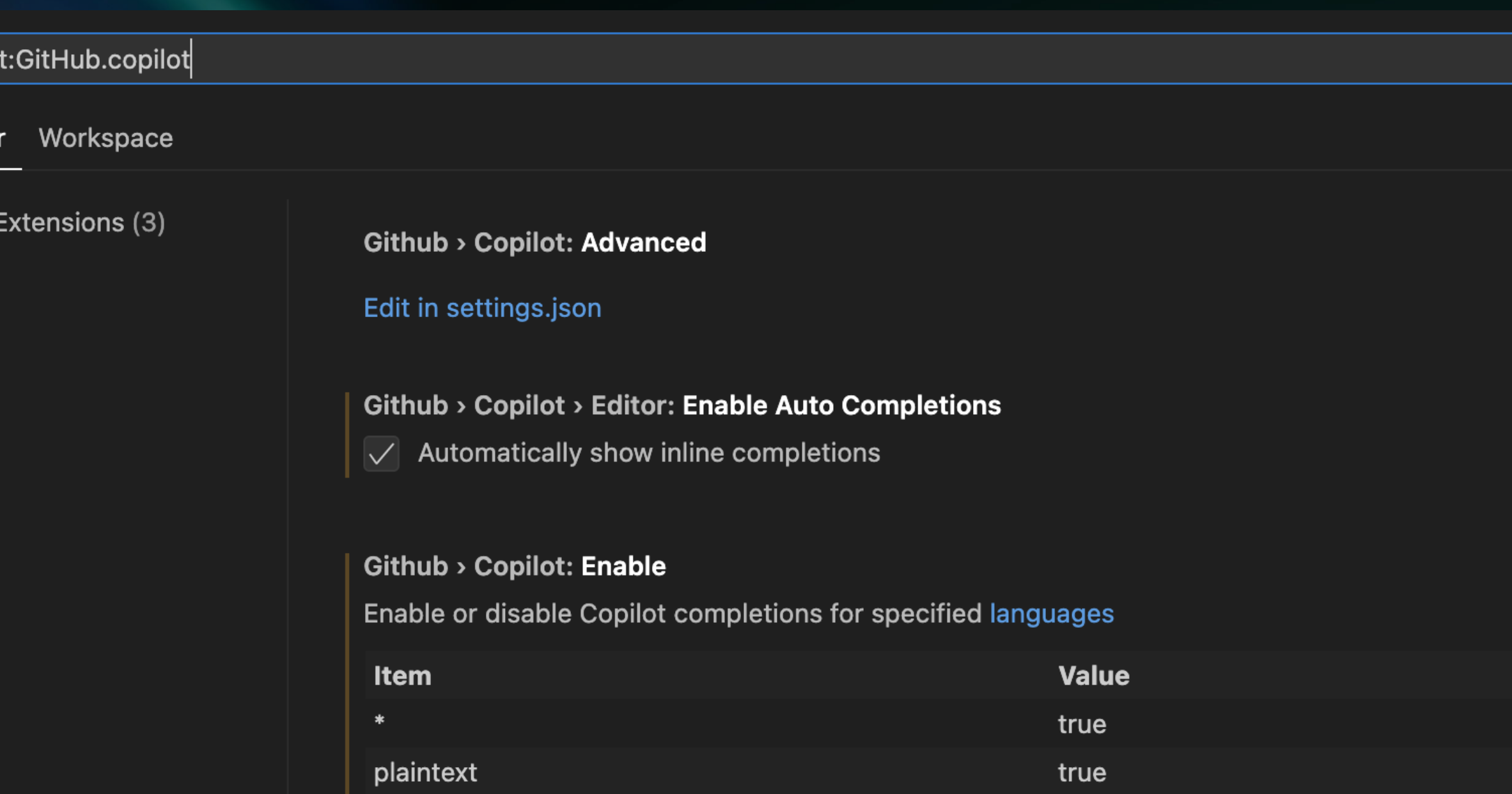
</div>

One-shot Prompt V1 Step 3

มาใช้ Github Copilot ช่วยเขียน Prompt กันเถอะ

1. เปิดไฟล์ scratchpad.md

2. Config github copilot ให้ทำงานกับ md ได้ละ ไป
ที่ Extensions > Github Copilot > Setting



Hands-On Lab

One-shot Prompt V1 Step 3

ลงมือแก้ scratchpad กัน!

One-shot Prompt V1 Step 4

เริ่มเรียนรู้ API กับคำศัพท์กันไว้ๆ ก่อนจะประยุกต์ใช้ของที่อยู่ใน scratchpad.md

System Prompt คือ Prompt สำหรับนำทางบทสนทนาทั้งหมดที่จะตามมา

User Prompt คือ Prompt ที่ user input เข้ามา

AI Prompt คือ Prompt ที่ ai ตอบมา

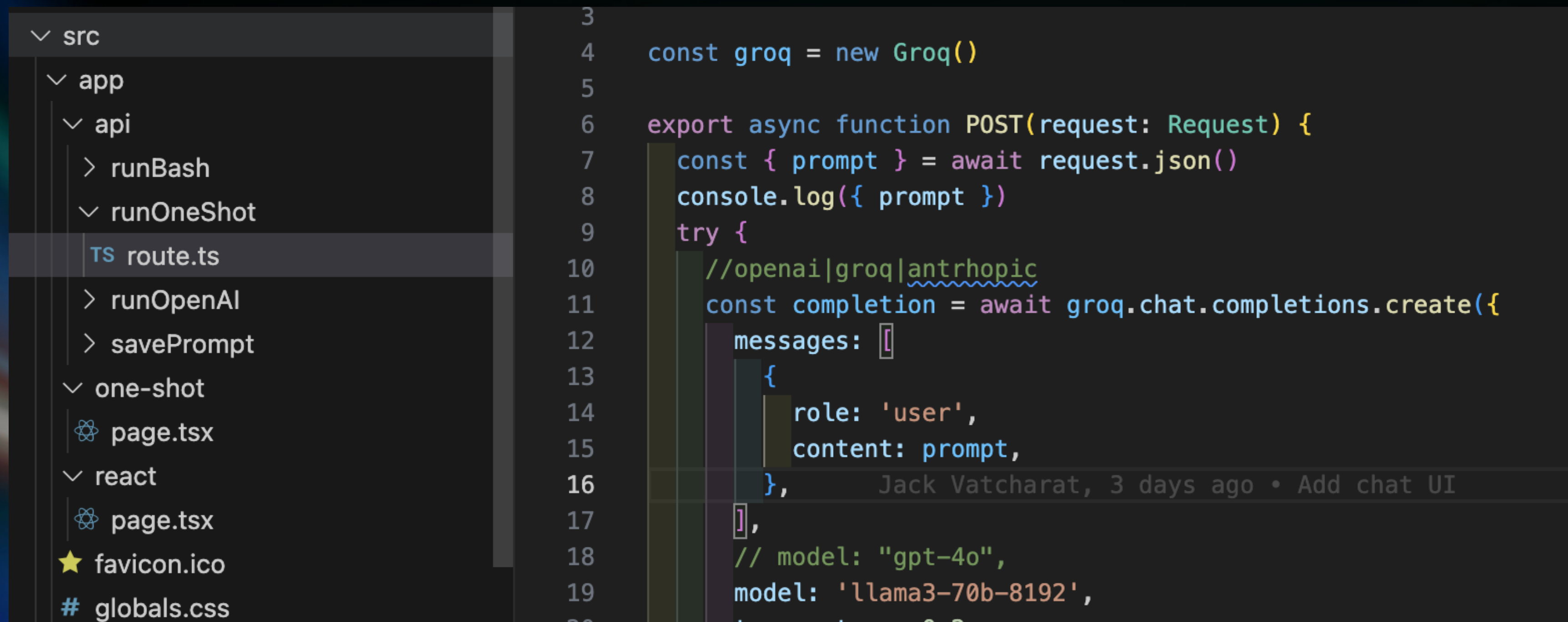
One-shot Prompt V1 Step 5

เขียน scratchpad เสร็จแล้วมาเขียนโค้ดด้วย Github Copilot กัน

1. ทดสอบ LLM-Bash-Agent OneShot tab
2. เปิดไฟล์ `src/app/api/runOneshot/route.ts`

One-shot Prompt Step 6

เปิดไฟล์ src/app/api/runOneShot/route.ts บรรทัดที่ 15



```
3
4  const groq = new Groq()
5
6  export async function POST(request: Request) {
7    const { prompt } = await request.json()
8    console.log({ prompt })
9    try {
10     //openai|groq|anthropic
11     const completion = await groq.chat.completions.create({
12       messages: [
13         {
14           role: 'user',
15           content: prompt,
16         },
17         {
18           // model: "gpt-4o",
19           model: 'llama3-70b-8192',
20           temperature: 0.2
```


One-shot Q&A

ReAct Hands-On

เหตุการณ์สมมติ

- ณ บริษัทที่ปรึกษา IT แห่งหนึ่ง Tech Lead ต้องการนำ LLM มาใช้แทน Junior Software Engineer
- งานแบ่งออกเป็น 2 เฟส

Phase 1

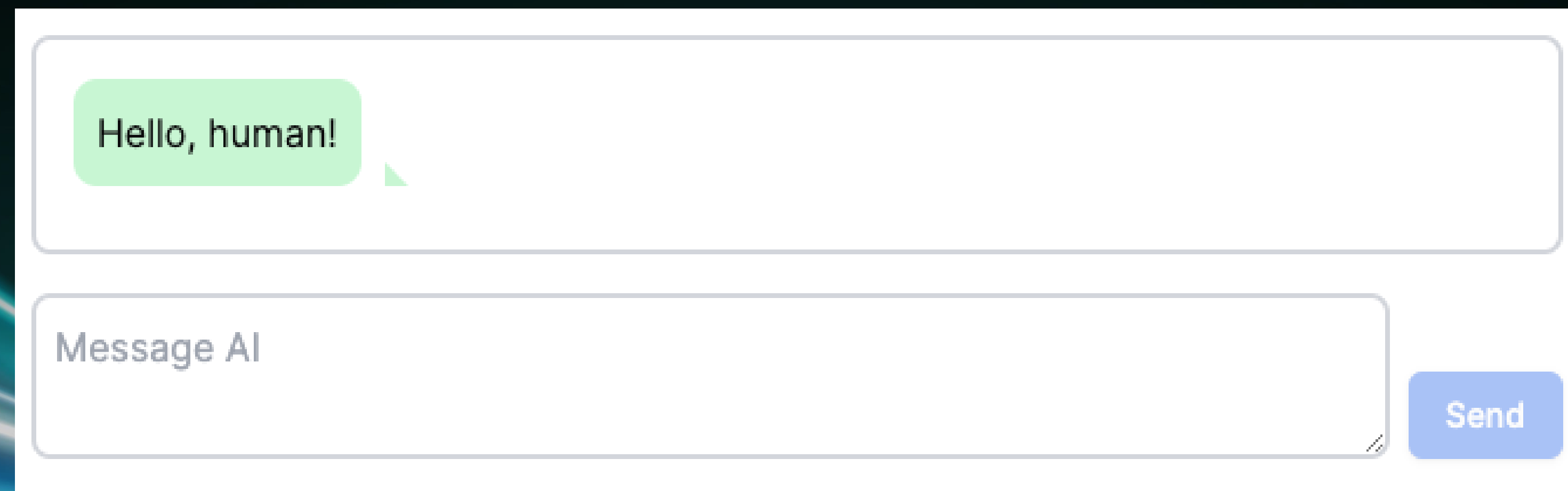
- เพื่อขอจบจากบริษัท Tech lead คนนั้นจำเป็นต้องโชว์ความสามารถพื้นฐานของ LLM ว่าสามารถเขียนโค้ดได้จริง
- ให้ผู้บริหารของบริษัทดู เพื่อขอจบประมาณในการทำระบบ
- ทาง Tech Lead ตัดสินใจใช้ One shot prompt ที่ได้เรียนมาจาก Youtube เพื่อแสดงความสามารถ ของ LLM ให้ทางผู้บริหารดู

One-shot Prompt V2 Step 1

เขียน prompt ที่แปลง input ที่ได้รับเป็น bash script

ตัวอย่าง Input

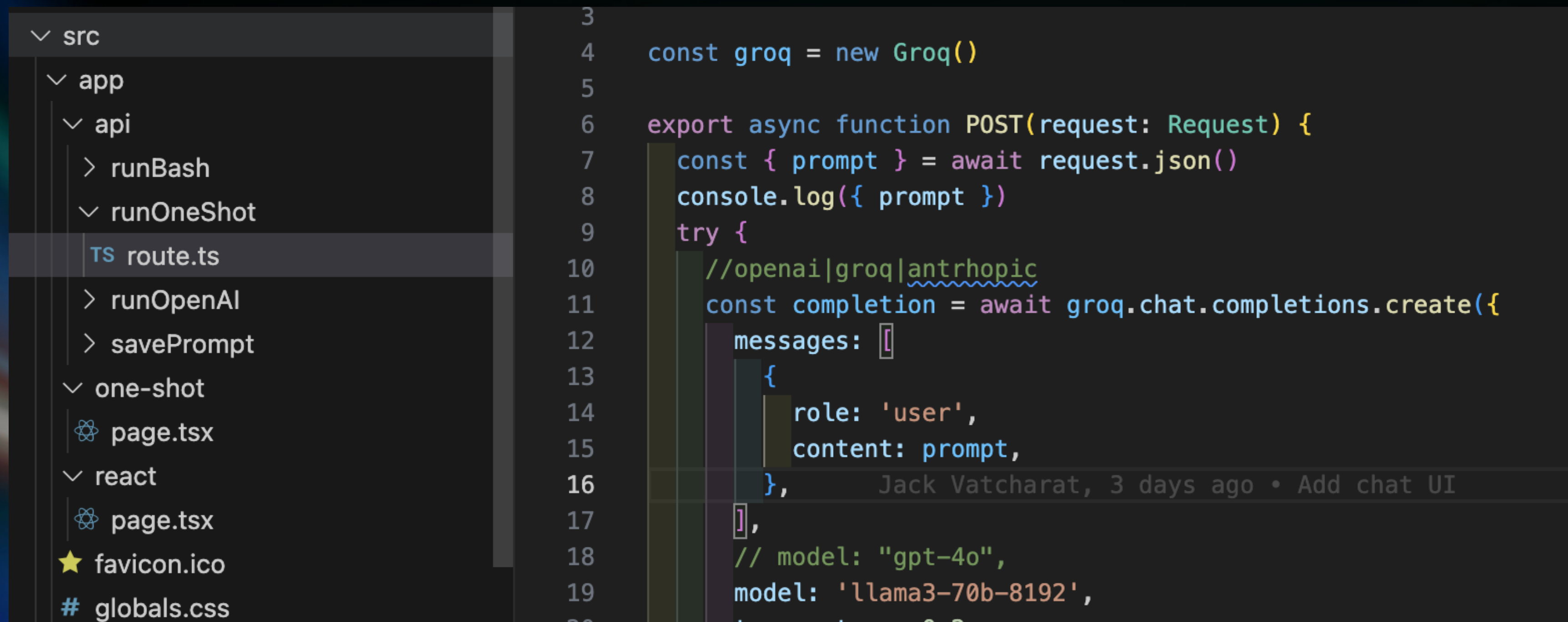
"Set up a local backend REST API for /products and return a list of 5 products."



The image shows a chat interface. At the top, there is a green bubble containing the text "Hello, human!". Below this, there is a white input field with the placeholder text "Message AI". To the right of the input field is a blue button labeled "Send".

One-shot Prompt V2 Step 2

เปิดไฟล์ src/app/api/runOneShot/route.ts บรรทัดที่ 15



```
3
4  const groq = new Groq()
5
6  export async function POST(request: Request) {
7    const { prompt } = await request.json()
8    console.log({ prompt })
9    try {
10     //openai|groq|anthropic
11     const completion = await groq.chat.completions.create({
12       messages: [
13         {
14           role: 'user',
15           content: prompt,
16         },
17         {
18           // model: "gpt-4o",
19           model: 'llama3-70b-8192',
20           temperature: 0.2
```


Let's code it together!

Phase 2

- ทาง Tech Lead ได้งบประมาณ 3 ล้านบาท และพร้อมที่จะพัฒนา Web App + LLM ที่จะมาแทนที่ Junior Software Engineer 50% ของบริษัท
- เขาจึงเริ่มวาด Architecture ขึ้น พร้อมกับมอบหมายงานให้ Junior Software Engineer 50% ที่เหลืออยู่ไปทำ Frontend และ Backend ขึ้นมา
- เหลือเพียงแค่ Tech Lead เองต้องเขียน Prompt โดยใช้ ReAct Prompting

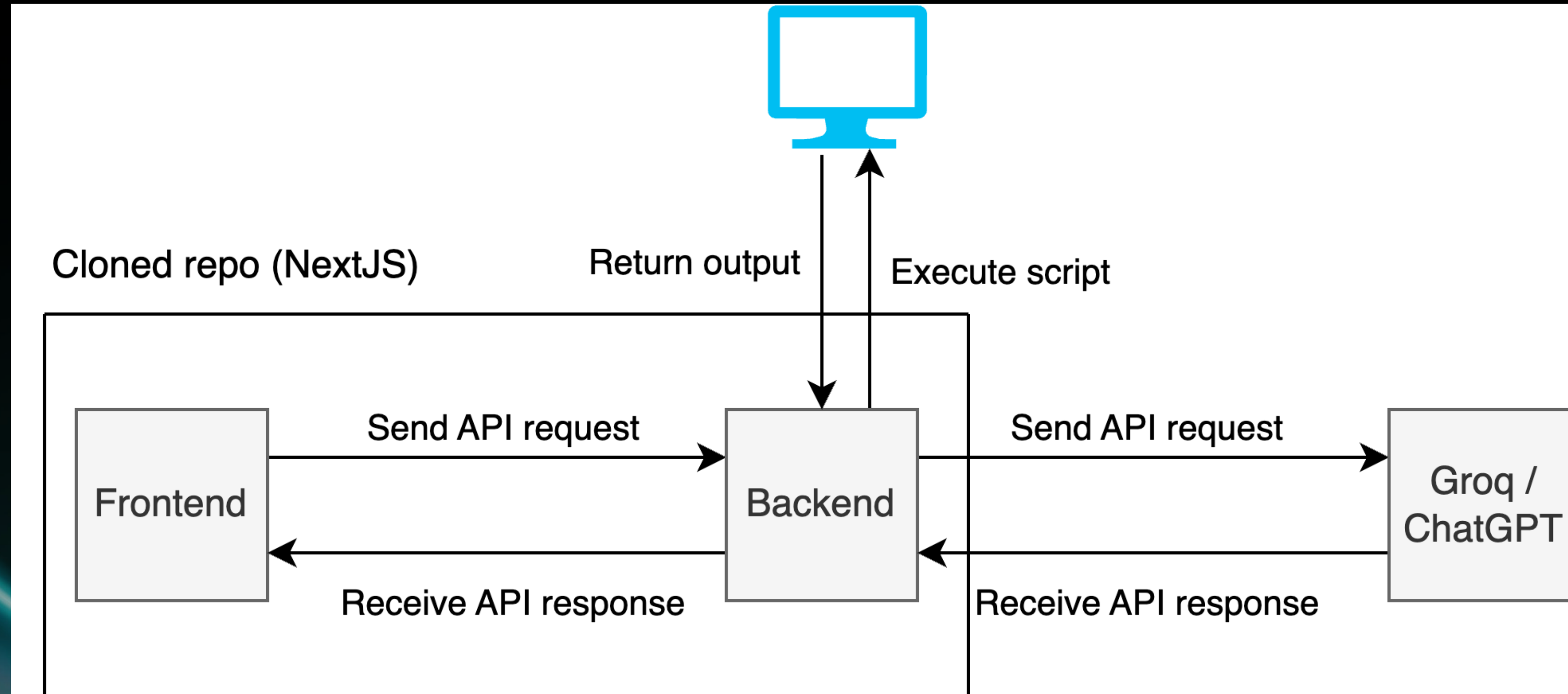
Pause here and go to ReAct.md

ReAct Basics!

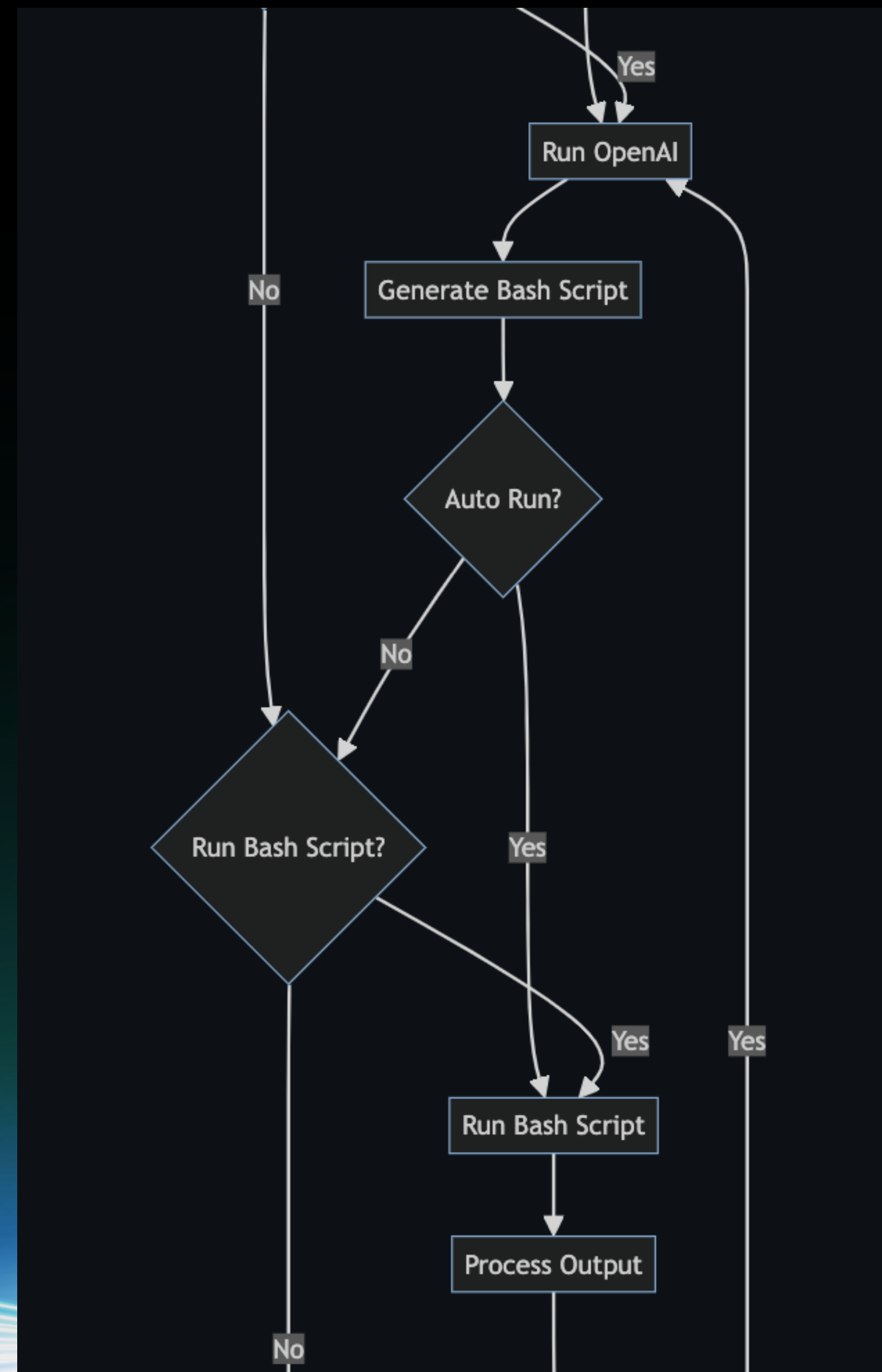
เราจะมาเริ่มทำความคุ้นเคยกับ ReAct จาก ReAct.md กันก่อน มาดูความสามารถของ Github Copilot กันว่าจะช่วยให้เราเข้าใจ ReAct ได้อย่างไร

ReAct Q&A

Architecture



Core functionality diagram

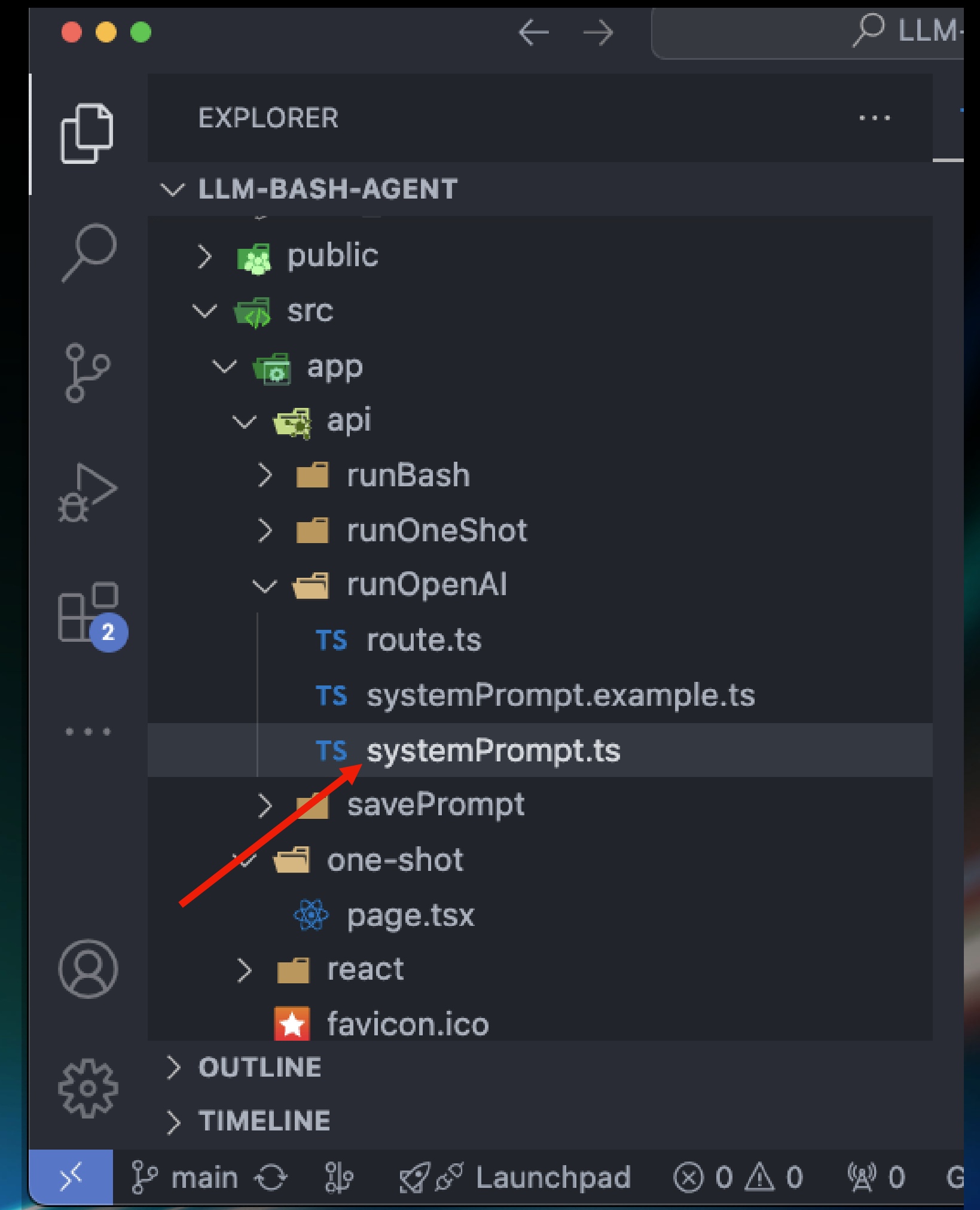
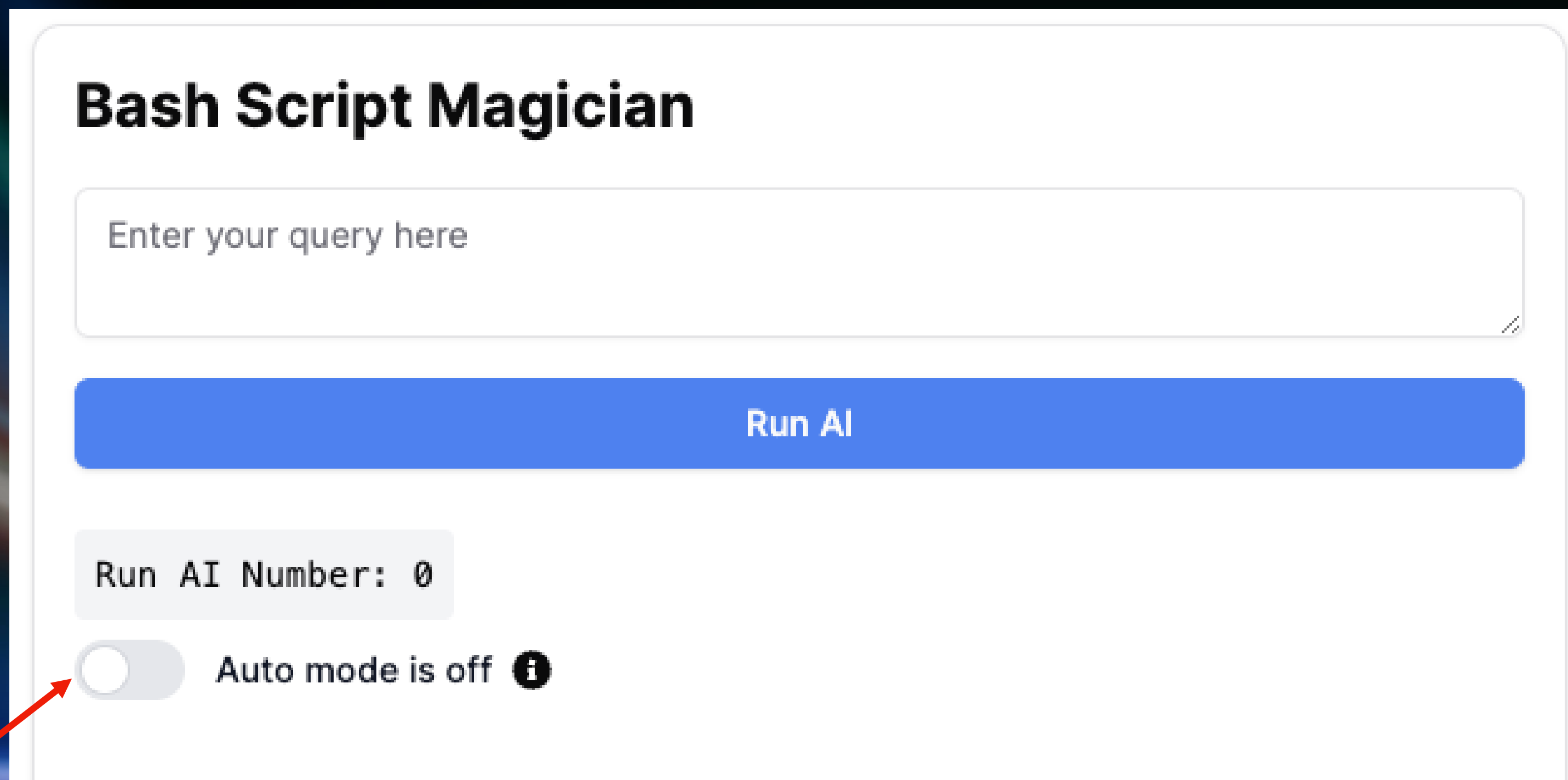


ReAct V1 - Auto Mode is Off

ReAct V1 Step 1

เริ่มจาก ปิด Auto mode แล้วเปิดไฟล์ `src/app/api/runOpenAI/systemPrompt.ts`

และเปิด ReAct.md อีก tab หนึ่ง



ReAct - Hands-On Lab

ReAct V1 Step 2



มาเขียน Code และ Prompt สำหรับ Manual Mode กัน

ReAct V1 - Auto Mode is On - Have Fun!

Bash Script Magician

Bash Script Magician

Enter your query here

Run AI (Auto)

Run AI Number: 0

☒ Auto mode is on ⓘ

Formatted Prompt Log

Log ⓘ

Generated Bash Script

Output

OpenAI Log

Prompt Log

Bash Log

NEXT.js

Bash Script Magician

Bash Script Magician

Say Hello with Current Time

Stop

Status: AutoRun: Run AI

Run AI Number: 2

☒ Auto mode is on

Formatted Prompt Log

Instruction 1:
Say Hello with Current Time

Thought 1:
The task is started. Start at Home directory. And create a new directory with the name of the task.

ActionType 1:
Execute

Action 1:

```
```bash
cd $HOME/SayHello [&&/] mkdir -p
$HOME/SayHello
```
```

Observation 1:
Changed directory to:
/Users/jack/SayHello
Error executing command in
/Users/jack/SayHello: spawn /bin/sh
ENOENT

Bash Script Magician

Bash Script Magician

Say hello with Current Time

Run AI (Auto)

Status: Task is done

Run AI Number: 6

☒ Auto mode is on ⓘ

Thought 5:

Now that the bash script is created, I need to execute it to see the output.

ActionType 5:

Execute

Action 5:

```
```bash
cd $HOME/SayHello
./say_hello.sh
```
```

Observation 5:

Hello, the current time is Fri Jul 5
17:19:09 +07 2024
Hello, the current time is Fri Jul 5
17:25:18 +07 2024

Thought 6:

The task is done.

Log ⓘ

Generated Bash Script

```
cd $HOME/SayHello
./say_hello.sh
```

Output

Task is done

OpenAI Log

Thought: Now that the bash script is created, I need to execute it to see the output.
ActionType: Execute
Action: ```bash
cd \$HOME/SayHello
./say_hello.sh
```

### Prompt Log

Instruction: Say hello with Current Time  
Let's start the task.



Scan to receive complimentary course  
with PALO IT





Q&A



Thank you