

ChatGPT cheatsheet





TABLE OF CONTENTS

1. Getting Started with ChatGPT

- Difference between GPT-3.5, GPT-4, and GPT-4o for devs
- When to use free vs. pro version
- ChatGPT web UI vs API usage
- Token limits and pricing overview (OpenAI API)

2. Prompt Engineering for Developers

- Prompt formatting tips (bullets, numbered lists, delimiters)
- Using ### for sections in prompts
- How to tell ChatGPT your programming context
- Using JSON/YAML in prompts
- Anchoring responses with examples (e.g. prompt technique)

3. Code Generation by Language

- Python (scripts, Flask/Django APIs, pandas)
- JavaScript (DOM, async/await, fetch API)
- TypeScript (typing suggestions, DTOs)
- React/Vue/Angular (components, props, hooks)
- Java/Kotlin (Spring Boot, Android snippets)
- PHP/Laravel, Ruby on Rails, Go, Rust, C++

4. Debugging and Error Resolution

- Asking "Why is this not working?" with code context
- Common bugs ChatGPT can catch (infinite loops, wrong syntax)
- Asking for log-level debugging
- Copy-pasting error output for explanation

5. Code Optimization and Refactoring

- Improving performance (e.g., Big-O reduction)
- Rewriting nested code into cleaner logic
- Transforming imperative → functional programming
- Secure coding practices (e.g., input validation)

6. Documentation Generation

- Automatic docstring and comment generation
- Swagger/OpenAPI spec creation
- Markdown README generator
- Code examples with explanation for blogs/tutorials



TABLE OF CONTENTS

7. DevOps / Terminal Commands

- Dockerfile + docker-compose.yml generation
- Bash/shell scripts (automate git, zipping, cronjobs)
- Kubernetes manifest templates
- CI/CD pipelines (GitHub Actions, GitLab CI)

8. Database Help

- SQL queries with JOINS, GROUP BY, CTEs
- MongoDB queries and aggregation pipelines
- Sequelize/TypeORM queries
- Migrations and schema design
- Example database dump + restore commands

9. APIs and Backend Support

- REST API creation + explanation
- Express.js, FastAPI, Flask, Spring Boot
- JWT and OAuth implementation
- Rate limiting, CORS, and logging setup

10. Frontend/UI Help

- HTML/CSS UI snippets
- Tailwind CSS components
- Responsive layout generation
- GSAP/Framer Motion animations
- Accessibility (a11y) improvements

11. Testing

- Unit test generation (Jest, PyTest, JUnit)
- End-to-end test flows (Cypress, Selenium)
- Test data mocks and stubs
- TDD workflow with ChatGPT prompts

12. AI & Data Projects

- NumPy, pandas, sklearn helpers
- NLP and image processing examples
- Dataset generation and augmentation
- Code explanation for ML pipelines

13. Project Assistance

- Project ideas based on skill level
- How to ask ChatGPT to build MVP step-by-step
- Feature planning and task breakdown
- Bug tracking and feature refinement with prompts



TABLE OF CONTENTS

14. Integrations & Plugins

- ChatGPT for VS Code / Cursor IDE
- Using ChatGPT in Postman
- GitHub Copilot vs ChatGPT
- Creating your own custom GPT for dev workflows

15. Interview Prep & Career Help

- Solving DSA questions
- Simulating system design interviews
- Improving GitHub profile & resume
- Creating a personal dev portfolio

16. Best Practices & Pitfalls

- Avoiding over-reliance
- Validating generated code
- When to avoid using ChatGPT
- How to give clear context in prompts

17. Smart Prompts for Developers

- Code Writing & Generation Prompts
- Code Refactoring & Optimization Prompts
- Bug Fixing & Debugging Prompts
- Code Explanation Prompts
- Documentation & Commenting Prompts
- Testing (Unit, Integration, E2E) Prompts
- DevOps & CLI Prompts
- Database Query & Design Prompts
- API Design & Implementation Prompts
- UI/UX & Frontend Prompts
- Learning New Tech Stack Prompts
- Data Structures & Algorithms Prompts
- Project & MVP Building Prompts
- Version Control (Git) Prompts
- Productivity & Automation Prompts
- Resume, GitHub & Portfolio Optimization Prompts
- Interview Preparation Prompts
- Prompt Engineering for Better Results



1. GETTING STARTED WITH CHATGPT

- This section helps you understand what ChatGPT is, which version to use, how to access it, and how tokens (words) are counted.

1.1 Difference between GPT-3.5, GPT-4, and GPT-4o (For Developers)

GPT-3.5

- GPT-3.5 is a smart model, but not the smartest.
- It is fast and free.
- Good for simple coding help, basic answers, and short tasks.

GPT-4

- GPT-4 is smarter than GPT-3.5.
- It understands longer questions and gives better answers.
- It is only available in the ChatGPT Plus (paid) plan.

GPT-4o ("o" stands for Omni)

- GPT-4o is the newest and smartest version.
- It is faster than GPT-4 and can understand text, images, and voice.
- It gives the best answers and is great for developers.
- Available in ChatGPT Plus, and also in the API.

Feature	GPT-3.5 (Free)	GPT-4 (Pro)	GPT-4o (Pro)
Speed	Fast	Slower	Fast
Accuracy	Medium	High	Very High
Multimodal	No	No	Yes (text, image)
Price	Free	Paid	Paid

1.2 When to Use Free vs. Pro Version

Free Version (GPT-3.5)

- Good for beginners.
- You can try basic coding help, homework, small answers.
- No cost, but not always correct.

Pro Version (GPT-4 or GPT-4o)

- Use when you need better answers, coding help, or you want to build apps.
- Use if you are working on professional tasks, content writing, or ChatGPT plugins.
- Pro version costs \$20/month.

Tip:

- Start with free. If you feel it gives wrong answers or can't understand your question, try the pro version.



1. GETTING STARTED WITH CHATGPT

1.3 ChatGPT Web UI vs. API Usage

Web UI (chat.openai.com)

- You type your question and get answers.
- Easy to use — no coding needed.
- Good for learning, testing, chatting, writing, and getting code examples.

API (for developers)

- Used to connect ChatGPT with your own app, software, or website.
- You send requests using code (like Python or JavaScript).
- Good for making chatbots, AI tools, or automating tasks.

Feature	Web UI	API
Use Case	Ask questions manually	Automate and build tools
Code Needed	No	Yes
Speed	Medium	Fast (customizable)
Example	Ask “What is HTML?”	Make a bot that answers questions

1.4 Token Limits and Pricing Overview (OpenAI API)

What is a token?

- A token is like a small piece of text.
- For example:
 - "ChatGPT is great!" = 4 tokens
 - Each word or part of a word counts as a token.

Token Limits (Max Text Length):

Model	Max Tokens
GPT-3.5	16,385
GPT-4	128,000
GPT-4o	128,000

- This means GPT-4 can remember more in a single conversation.



1. GETTING STARTED WITH CHATGPT

Pricing (as of 2025, via OpenAI API):

Model	Input (per 1K tokens)	Output (per 1K tokens)
GPT-3.5	\$0.00	\$0.00
GPT-4	\$0.03	\$0.06
GPT-4o	\$0.01	\$0.02

Note:

- Input tokens = what you send (your prompt).
- Output tokens = what ChatGPT replies.

So if you send 1000 tokens and get 1000 tokens back, you pay for 2000 tokens.



2. PROMPT ENGINEERING FOR DEVELOPERS

- This section helps you write better questions (called “prompts”) so ChatGPT understands you clearly, especially for coding.

2.1 Prompt Formatting Tips (Bullets, Numbered Lists, Delimiters)

- ChatGPT gives better answers when you write clearly.

Tips:

Use bullet points (-) to separate ideas

- Example:

Explain the difference between:

- HTML
- CSS
- JavaScript

Use numbered lists to show steps

- Example:

Give me the steps to create a login page:

- Create HTML form
- Add CSS styles
- Write JavaScript for validation

Use triple backticks (```) to separate code

- Example:

Write Python code to add two numbers:

```
● ● ● python  
def add(a, b):  
    return a + b
```

- Be specific in your question
- Not good: "Write code"
- Better: "Write Python code to reverse a string without using built-in functions"

2.2 Using ### for Sections in Prompts

- Use ### to divide parts of your prompt. ChatGPT understands sections better this way.



2. PROMPT ENGINEERING FOR DEVELOPERS

2.2 Using ### for Sections in Prompts

- Example Prompt:

```
markdown

### Goal:
Build a React login form

### Features:
- Email input
- Password input
- Show/hide password
- Submit button

### Output Format:
Give me only the final code with no explanation
```

- ChatGPT will now break the answer based on your sections.

2.3 How to Tell ChatGPT Your Programming Context

- If you tell ChatGPT about your coding level or tools you are using, it will give better answers.

Examples:

- “I’m a beginner in JavaScript. Explain simply.”
- “Use Tailwind CSS in the answer.”
- “Use React 18 syntax only.”
- “Show example using fetch() not axios.”

Tip:

- Write your needs clearly at the top of your prompt. Like:

```
css

I want beginner-friendly explanation with small code snippets in Python only.
```

2.4 Using JSON or YAML in Prompts

- You can use JSON or YAML formats to give structured instructions.

Why?

- ChatGPT understands structure better when data is clean and labeled.

JSON Prompt Example:

```
json

{
  "task": "Create a REST API",
  "language": "Node.js",
  "framework": "Express",
  "features": [
    "GET endpoint",
    "POST endpoint",
    "MongoDB connection"
  ]
}
```



2. PROMPT ENGINEERING FOR DEVELOPERS

2.4 Using JSON or YAML in Prompts

- YAML Prompt Example:

```
task: Build To-Do App
language: Python
framework: Flask
features:
  - Add new task
  - Mark task complete
  - Delete task
```

Tip:

- Use this when asking for code generation, documentation, or templates.

2.5 Anchoring Responses with Examples (Prompt Technique)

- If you give an example, ChatGPT will follow that style.
- This is called "few-shot prompting."

Example Prompt:

```
Explain the following JavaScript functions like this:
Function: map()
Description: Used to transform every item in an array.
Example: [1,2,3].map(x => x * 2) => [2,4,6]

Now explain:
- filter()
- reduce()
```

Why it works:

- You show the format you want. ChatGPT follows the style exactly.



3. CODE GENERATION BY LANGUAGE

- Learn how to ask ChatGPT for code in different programming languages and what it can do for you.

3.1 Python

What to ask ChatGPT:

- Simple scripts (like calculator, to-do list, file reader)
- APIs using Flask or Django
- Data analysis with pandas

Example Prompts:

- Write a Python script to rename all files in a folder
- Build a REST API with Flask that has GET and POST routes
- Using pandas, how can I remove duplicate rows from a DataFrame?

Good for:

- Beginners
- Automation
- Backend APIs
- Data analysis

3.2 JavaScript

What to ask ChatGPT:

- DOM manipulation (changing content on a web page)
- Async/await functions (handling delay in code)
- Working with APIs using fetch()

Example Prompts:

- Write JavaScript to change the text of a button when clicked
- Create an async function that fetches JSON data from a URL
- Show how to use fetch to get user data from an API

Good for:

- Web pages
- Frontend tasks
- Client-side scripting

3.3 TypeScript

What to ask ChatGPT:

- Type-safe code examples
- DTOs (Data Transfer Objects)
- Function/type definitions

Example Prompts:

- Convert this JavaScript function to TypeScript
- Create a DTO for a user with id, name, and email
- Show how to use union and intersection types



3. CODE GENERATION BY LANGUAGE

3.3 TypeScript

Good for:

- Large projects
- Safer JavaScript
- Teams and structure

3.4 React / Vue / Angular

React

What to ask:

- Components
- Props and state
- Hooks (useState, useEffect, etc.)

Example Prompt:

- Create a React component with a counter using useState

Vue

What to ask:

- Options API or Composition API
- Reactive data
- Vue directives

Example Prompt:

- Build a Vue component that toggles dark mode

Angular

What to ask:

- Components, services, routing
- Angular CLI usage

Example Prompt:

- Create an Angular component that shows a list of users

Good for:

- Frontend development
- Building user interfaces
- Reusable components

3.5 Java / Kotlin

Java (Spring Boot):

- APIs
- Controllers
- Entity classes

Kotlin (Android):

- UI XML and Kotlin files
- Button click listeners
- Retrofit API usage



3. CODE GENERATION BY LANGUAGE

3.5 Java / Kotlin

Example Prompts:

- Create a Spring Boot REST API with CRUD for products
- Write Kotlin code to open a new Android activity on button click

Good for:

- Enterprise apps
- Android development
- Backend services

3.6 PHP / Laravel

What to ask:

- PHP scripts
- Laravel routes, controllers, migrations

Example Prompts:

- Create a Laravel controller for handling blog posts
- Write a PHP script to upload a file to the server

3.7 Ruby on Rails

What to ask:

- Models, views, and controllers (MVC)
- Rails routes
- ActiveRecord queries

Example Prompts:

- Create a Rails scaffold for a Task model with title and status
- Show how to write a migration to add a column to users

3.8 Go (Golang)

What to ask:

- Simple HTTP servers
- REST APIs
- Go routines and channels

Example Prompts:

- Create a Go server that responds with "Hello World"
- Build a REST API in Go to manage products

3.9 Rust

What to ask:

- Basic CLI tools
- Structs and enums
- Simple APIs (with Actix or Rocket)

Example Prompts:

- Write Rust code to read a file line by line
- Create a simple Rust struct for a Book with title and author



3. CODE GENERATION BY LANGUAGE

3.10 C++

What to ask:

- Programs using loops, arrays, functions
- OOP (classes, inheritance)
- File handling

Example Prompts:

- Write a C++ program to sort an array
- Show how to create a class with a constructor and destructor



4. DEBUGGING AND ERROR RESOLUTION

- Learn how to ask ChatGPT to help you fix broken code, step by step.

4.1 Asking "Why is this not working?" with Code Context

- If your code isn't working, don't just say:
- "My code doesn't work."
- Instead, follow this structure:



vbnet

```
Here is my code (below). It's supposed to [what the code should do].  
But the issue is [what goes wrong].  
Can you help fix it?  
  
[Your code goes here]
```

Example:



css

```
This JavaScript code should show an alert when I click a button.  
But nothing happens when I click.  
  
HTML:  
<button id="btn">Click</button>  
  
JS:  
document.querySelector("btn").addEventListener("click", () => {  
    alert("Clicked!");  
});
```

- ChatGPT can now understand what's wrong and help.

4.2 Common Bugs ChatGPT Can Catch

ChatGPT is very good at spotting common coding mistakes like:

- Wrong syntax:
- Example: Missing ;, wrong brackets (), {}, or quotes ".
- Typos in variable names:
- Using userName vs username.
- Infinite loops:
- Loops that never end like:



python

```
while True:  
    print("oops") # runs forever
```

- Missing return or break statements
- Incorrect use of functions or APIs

4.3 Asking for Log-Level Debugging

- You can tell ChatGPT to explain line-by-line what your code is doing.



4. DEBUGGING AND ERROR RESOLUTION

4.3 Asking for Log-Level Debugging

- Example Prompt:



pgsql

Here is my code. Please give a log-level explanation (step-by-step), and find out why it fails.

[Your code]

- Result: ChatGPT will go line by line, like:



pgsql

1. Reads input from user
2. Calls function calculateTotal()
3. Error at line 6 because variable 'price' is undefined

- This helps you understand not just the fix, but why the problem happened.

4.4 Copy-Pasting Error Output for Explanation

- If you get an error message, just paste it in.

Example:



javascript

TypeError: Cannot read properties of undefined (reading 'length')

Then ask:



csharp

Can you explain this error and how to fix it? My code is:

[Your code]

ChatGPT will:

- Tell you what the error means
- Point out which line is causing it
- Suggest a fix



5. CODE OPTIMIZATION AND REFACTORING

- Learn how to improve your code — make it faster, cleaner, and safer with ChatGPT's help.

5.1 Improving Performance (e.g., Big-O Reduction)

What is performance?

- It means how fast and efficient your code runs.

What is Big-O?

- Big-O tells how fast your code grows with bigger inputs.
- Example: Sorting 10 items is easy, but what about 1,000 or 1,000,000?

Common Big-O examples:

- $O(1)$ → Constant time (fast)
- $O(n)$ → Slower as data grows
- $O(n^2)$ → Much slower with big data

How ChatGPT helps:

You can ask:

```
● ● ●          css  
Here's my code. Can you tell if it's slow? How can I make it faster?
```

Example Prompt:

```
● ● ●          javascript  
This function checks duplicates in an array.  
Is there a faster way using better Big-O?  
  
function hasDuplicates(arr) {  
    for (let i = 0; i < arr.length; i++) {  
        for (let j = i + 1; j < arr.length; j++) {  
            if (arr[i] === arr[j]) return true;  
        }  
    }  
    return false;  
}
```

- ChatGPT might suggest using a Set to improve it to $O(n)$.

5.2 Rewriting Nested Code into Cleaner Logic

- Nested code = Code with too many layers, like this:

```
● ● ●          python  
if user:  
    if user['active']:  
        if user['role'] == 'admin':  
            do_admin_task()
```

- This is hard to read.



5. CODE OPTIMIZATION AND REFACTORING

5.2 Rewriting Nested Code into Cleaner Logic

- ChatGPT can help rewrite like:

```
● ● ● python  
if not user or not user['active'] or user['role'] != 'admin':  
    return  
  
do_admin_task()
```

Prompt Example:

```
● ● ● csharp  
Can you clean this code and reduce the nesting?  
[Paste your code]
```

5.3 Transforming Imperative → Functional Programming

- Imperative = Step-by-step instructions
- Functional = Use of map, filter, reduce, pure functions
- Example Prompt:

```
● ● ● arduino  
Change this for loop to functional style using map or reduce
```

Before:

```
● ● ● js  
let numbers = [1, 2, 3];  
let double = [];  
for (let i = 0; i < numbers.length; i++) {  
    double.push(numbers[i] * 2);  
}
```

After:

```
● ● ● js  
let double = numbers.map(n => n * 2);
```

- Functional code is shorter, cleaner, and often faster.

5.4 Secure Coding Practices (e.g., Input Validation)

- Even if your code works, it must be safe.
- What to ask:

```
● ● ● css  
Can you help make this input code secure?
```



5. CODE OPTIMIZATION AND REFACTORING

5.4 Secure Coding Practices (e.g., Input Validation)

What ChatGPT can add:

- Input validation
- SQL injection protection
- XSS prevention
- Try/catch blocks

Example:

```
● ● ● js  
// Unsafe:  
let username = req.query.name;  
db.query(`SELECT * FROM users WHERE name = '${username}'`);
```

ChatGPT Suggestion:

- Use parameterized queries to prevent attacks.



6. DOCUMENTATION GENERATION

- Learn how to ask ChatGPT to help you write clear, useful documentation for your code, APIs, and projects.

6.1 Automatic Docstring and Comment Generation

- What is a docstring?
- A docstring is a short explanation written inside a function or class.

It tells:

- What the function does
- What input it takes
- What output it gives

Example Prompt:

```
● ● ●          kotlin  
Add a Python docstring to this function:  
def add(a, b):  
    return a + b
```

ChatGPT Output:

```
● ● ●          python  
def add(a, b):  
    """  
    Adds two numbers.  
  
    Parameters:  
    a (int or float): First number  
    b (int or float): Second number  
  
    Returns:  
    int or float: Sum of the two numbers  
    """  
    return a + b
```

You can also ask:

- "Add inline comments to explain this code"
- "Explain what each line in this function does"

6.2 Swagger / OpenAPI Spec Creation

- What is Swagger?
- Swagger (now called OpenAPI) is a way to describe your API so others can use it easily.

Example Prompt:

```
● ● ●          pgsql  
Create Swagger JSON for a GET and POST API for /users
```



6. DOCUMENTATION GENERATION

ChatGPT Output:

```
● ● ● json
{
  "paths": {
    "/users": {
      "get": {
        "summary": "Get all users",
        "responses": {
          "200": {
            "description": "List of users"
          }
        }
      },
      "post": {
        "summary": "Add a new user",
        "requestBody": {
          "required": true,
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/User"
              }
            }
          }
        },
        "responses": {
          "201": {
            "description": "User created"
          }
        }
      }
    }
  }
}
```

You can also say:

```
● ● ● css
Generate OpenAPI YAML for a login API
```

6.3 Markdown README Generator

- What is a README?
- It tells others what your project is, how to install it, and how to use it.
- Example Prompt:

```
● ● ● css
Generate a README.md file for my Node.js weather app
```

ChatGPT Output (in Markdown):



6. DOCUMENTATION GENERATION

ChatGPT Output (in Markdown):

```
● ● ● markdown
# Weather App

This is a simple Node.js app that fetches weather data from an API.

## Features
- Get current weather
- Works with city name
- Simple UI

## Installation
```bash
git clone https://github.com/yourname/weather-app
cd weather-app
npm install
```

```

Usage:

```
● ● ● bash
node app.js
```

License

MIT

```
● ● ● yaml
---
## 6.4 Code Examples with Explanation (for Blogs/Tutorials)

You can ask ChatGPT to create **step-by-step tutorials** for your blog or post.

**Example Prompt:**
```

- Write a blog-style explanation for a JavaScript function that filters even numbers

```
● ● ● javascript
**ChatGPT Output:** 
```markdown
How to Filter Even Numbers in JavaScript

Sometimes you want to remove odd numbers from a list. Here's how:

Code:
```js
const numbers = [1, 2, 3, 4, 5, 6];
const even = numbers.filter(n => n % 2 === 0);
console.log(even); // [2, 4, 6]
```

```

Explanation:

- `filter()` is a function that keeps only the values that return true
- `n % 2 === 0` checks if the number is even



## 6. DOCUMENTATION GENERATION

- $n \% 2 === 0$  checks if the number is even

```
yaml

Summary Table:

Task	What to Ask ChatGPT
Add function docstrings	"Write docstring for this function"
Comment code	"Add comments explaining each line"
Create API docs	"Make Swagger/OpenAPI spec for these routes"
Make README	"Write README.md for my app"
Blog tutorial	"Explain this code like a tutorial"

Let me know if you want:
- All 6 sections combined into a **PDF guide**
- Each section turned into **Instagram carousel content**

I can generate either (or both) for you right away.
```



## 7. DEVOPS / TERMINAL COMMANDS

- `n % 2 === 0` checks if the number is even

```
yaml

Summary Table:

Task	What to Ask ChatGPT
Add function docstrings	"Write docstring for this function"
Comment code	"Add comments explaining each line"
Create API docs	"Make Swagger/OpenAPI spec for these routes"
Make README	"Write README.md for my app"
Blog tutorial	"Explain this code like a tutorial"

Let me know if you want:
- All 6 sections combined into a **PDF guide**
- Each section turned into **Instagram carousel content**

I can generate either (or both) for you right away.
```

- Learn how to use ChatGPT to write commands and files that automate deployment, scripting, and cloud tasks.

### 7.1 Dockerfile + docker-compose.yml Generation

What is Docker?

- Docker lets you package your app so it runs the same everywhere — on your PC, your server, or in the cloud.

What is a Dockerfile?

- A Dockerfile is a script that tells Docker how to build your app.

Example Prompt

```
css
Create a Dockerfile for a Node.js app that runs index.js
```

ChatGPT Output:

```
dockerfile
Dockerfile
FROM node:18
WORKDIR /app
COPY . .
RUN npm install
CMD ["node", "index.js"]
```

What is docker-compose?

- It lets you run many containers at once — like your app + database.
- Example Prompt:

```
css
Create a docker-compose.yml file for a Node app with MongoDB
```



## 7. DEVOPS / TERMINAL COMMANDS

- ChatGPT Output:

```
yaml

version: '3'
services:
 app:
 build: .
 ports:
 - "3000:3000"
 depends_on:
 - mongo
 mongo:
 image: mongo
 ports:
 - "27017:27017"
```

### 7.2 Bash/Shell Scripts (Automate Git, Zipping, Cron Jobs)

- You can ask ChatGPT to write terminal scripts that do tasks automatically.

Common tasks:

- Auto git push
- Zip folders
- Schedule tasks with cron

Example 1: Auto Git Push Script

Prompt:

```
pgsql

Write a bash script to git add, commit, and push with a message
```

Output:

```
bash

#!/bin/bash
git add .
git commit -m "$1"
git push
```

Example 2: Zip Folder Script

```
bash

#!/bin/bash
zip -r backup.zip /path/to/folder
```

Example 3: Cron Job (Run every day at 7 PM)

Prompt:

```
arduino

Give a cronjob line to run backup.sh at 7 PM daily
```

Output:

```
pgsql

0 19 * * * /path/to/backup.sh
```



## 7. DEVOPS / TERMINAL COMMANDS

### 7.3 Kubernetes Manifest Templates

- What is Kubernetes?
- It runs your app in the cloud using containers. You control how it runs using YAML manifest files.

Example Prompt:

```
● ● ● css
Create a Kubernetes deployment file for a Node.js app
```

ChatGPT Output:

```
● ● ● yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 name: node-app
spec:
 replicas: 2
 selector:
 matchLabels:
 app: node
 template:
 metadata:
 labels:
 app: node
 spec:
 containers:
 - name: node
 image: node:18
 ports:
 - containerPort: 3000
```

### 7.4 CI/CD Pipelines (GitHub Actions, GitLab CI)

- What is CI/CD?
- It means automatically testing and deploying your code when you push it to Git.
- Example: GitHub Actions

Prompt:

```
● ● ● psql
Create GitHub Actions workflow to build a Node.js app on push
```

Output:



## 7. DEVOPS / TERMINAL COMMANDS

### 7.4 CI/CD Pipelines (GitHub Actions, GitLab CI)

Output:

```
● ● ● yaml

.github/workflows/node.yml
name: Node CI

on:
 push:
 branches: [main]

jobs:
 build:
 runs-on: ubuntu-latest

 steps:
 - uses: actions/checkout@v2
 - name: Install dependencies
 run: npm install
 - name: Run tests
 run: npm test
```

Example: GitLab CI

Prompt:

```
● ● ● css

Create a GitLab CI config to build a Python app
```

Output:

```
● ● ● yaml

.gitlab-ci.yml
stages:
 - test

test:
 image: python:3.10
 script:
 - pip install -r requirements.txt
 - python -m unittest
```



## 8. DATABASE HELP

- Learn how to use ChatGPT to help with SQL, MongoDB, ORMs, and database design — even if you're just starting out.

### 8.1 SQL Queries with JOINs, GROUP BY, and CTEs

- JOINs (Combining tables)

Prompt Example:

```
● ● ● pgsql
Write a SQL query to join "users" and "orders" where user_id matches
```

Output:

```
● ● ● sql
SELECT users.name, orders.amount
FROM users
JOIN orders ON users.id = orders.user_id;
```

- GROUP BY (Summarizing data)

Prompt Example:

```
● ● ● pqsql
Group orders by user and count how many orders each user made
```

Output:

```
● ● ● sql
SELECT user_id, COUNT(*) AS total_orders
FROM orders
GROUP BY user_id;
```

- CTEs (Common Table Expressions – like temp queries)

Prompt Example:

```
● ● ● bash
Write a CTE that gets users who spent more than $500
```

Output:

```
● ● ● sal
WITH high_spenders AS (
 SELECT user_id, SUM(amount) AS total
 FROM orders
 GROUP BY user_id
)
SELECT * FROM high_spenders WHERE total > 500;
```



## 8. DATABASE HELP

### 8.2 MongoDB Queries and Aggregation Pipelines

- Basic Query
- Prompt Example:



bash

```
Find all users older than 30
```

- Output:



js

```
db.users.find({ age: { $gt: 30 } })
```

- Aggregation Example
- Prompt Example:



sql

```
Get total order amount per user
```

- Output:



js

```
db.orders.aggregate([
 { $group: { _id: "$user_id", total: { $sum: "$amount" } } }
])
```

ChatGPT can help write:

- \$match, \$group, \$lookup (MongoDB JOIN)
- Paginated queries
- Complex filters

### 8.3 Sequelize / TypeORM Queries

- Sequelize (Node.js ORM)
- Prompt Example:



pgsql

```
Find all users with include orders
```

- Output:



js

```
User.findAll({
 include: [
 model: Order
]
});
```



## 8. DATABASE HELP

### 8.3 Sequelize / TypeORM Queries

- TypeORM (for NestJS / TS devs)
- Prompt Example:



pgsql

```
Write a TypeORM query to get users with their posts
```

- Output:



ts

```
const users = await dataSource
 .getRepository(User)
 .find({ relations: ["posts"] });
```

### 8.4 Migrations and Schema Design

ChatGPT can:

- Design your database tables
- Help you generate migration scripts
- Recommend field types (VARCHAR, INT, etc.)

Example: Schema Design Prompt



pgsql

```
Design tables for a blog with users, posts, and comments
```

ChatGPT Output:



sql

```
-- users
id INT PRIMARY KEY
name VARCHAR(100)
email VARCHAR(100)

-- posts
id INT PRIMARY KEY
user_id INT (FK to users)
title TEXT
content TEXT

-- comments
id INT
post_id INT (FK to posts)
user_id INT (FK to users)
message TEXT
```

### 8.5 Example: Database Dump + Restore Commands

#### MySQL

Dump (Backup):



bash

```
mysqldump -u root -p mydb > backup.sql
```



## 8. DATABASE HELP

### 8.5 Example: Database Dump + Restore Commands

#### Restore:

```
● ● ● bash
mysql -u root -p mydb < backup.sql
```

#### MongoDB

#### Dump:

```
● ● ● bash
mongodump --db=mydb
```

#### Restore:

```
● ● ● bash
mongorestore --db=mydb dump/mydb
```



## 9. APIs AND BACKEND SUPPORT

- Learn how to create APIs and make them secure, fast, and ready for real-world apps.

### 9.1 REST API Creation + Explanation

- What is a REST API?
- A REST API lets apps talk to each other using HTTP (like GET, POST).

Example Prompt:

```
● ● ● sql
Create a REST API with GET and POST routes using Express.js
```

Output:

```
● ● ● js
const express = require('express');
const app = express();
app.use(express.json());

let users = [];

app.get('/users', (req, res) => {
 res.json(users);
});

app.post('/users', (req, res) => {
 users.push(req.body);
 res.status(201).send('User added');
});

app.listen(3000, () => console.log("API running"));
```

Explanation:

- GET /users → returns all users
- POST /users → adds a new user

You can do the same with other languages like Python, Java, etc.

### 9.2 Express.js, FastAPI, Flask, Spring Boot

Express.js (Node.js)

- Good for fast, simple APIs in JavaScript

Prompt:

```
● ● ● sql
Create a basic Express.js API with three routes
```

FastAPI (Python)

- Very fast and supports automatic docs

Prompt:

```
● ● ● bash
Build a FastAPI with /hello and /goodbye routes
```



## 9. APIs AND BACKEND SUPPORT

### 9.2 Express.js, FastAPI, Flask, Spring Boot

- Output:

```
python

from fastapi import FastAPI

app = FastAPI()

@app.get("/hello")
def hello():
 return {"msg": "Hello!"}

@app.get("/goodbye")
def goodbye():
 return {"msg": "Goodbye!"}
```

#### Flask (Python)

- Lightweight and beginner-friendly

Prompt:

```
sql

Create a Flask app with GET and POST endpoints
```

#### Spring Boot (Java)

- Used in enterprise-level apps, supports full backend features

Prompt:

```
sql

Create a Spring Boot API with /users endpoint
```

### 9.3 JWT and OAuth Implementation

#### JWT (JSON Web Token)

- Used to secure APIs — like a digital pass

Prompt:

```
pgsql

Add JWT authentication to an Express.js API
```

#### Output (short example):

```
js

const jwt = require("jsonwebtoken");

const token = jwt.sign({ userId: 123 }, "secret", { expiresIn: "1h" });

jwt.verify(token, "secret", (err, decoded) => {
 if (err) return res.status(401).send("Invalid token");
});
```



## 9. APIs AND BACKEND SUPPORT

### OAuth

- Used for logging in with Google, Facebook, etc.

Prompt:



pgsql

How to add Google OAuth login to a Node.js app?

- ChatGPT will guide you step-by-step with packages like passport or firebase.

### 9.4 Rate Limiting, CORS, and Logging Setup

#### Rate Limiting

- Stops users from calling your API too many times.

Prompt:



pgsql

Add rate limiting to Express.js using express-rate-limit

Output:



js

```
const rateLimit = require('express-rate-limit');
const limiter = rateLimit({ windowMs: 1 * 60 * 1000, max: 10 });
app.use(limiter);
```

#### CORS (Cross-Origin Resource Sharing)

- Allows or blocks frontend apps from accessing your API.

Prompt:



nginx

Enable CORS in Express.js



js

```
const cors = require('cors');
app.use(cors());
```

#### Logging

- Keeps records of requests and errors.

Prompt:



pgsql

Add basic logging to Express.js using morgan



js

```
const morgan = require('morgan');
app.use(morgan('dev'));
```



## 10. FRONTEND / UI HELP

- Learn how to use ChatGPT to build good-looking, mobile-friendly, and accessible user interfaces.

### 10.1 HTML / CSS UI Snippets

- You can ask ChatGPT to build small UI parts like buttons, forms, cards, etc.

Example Prompt:

● ● ●                            css  
Create a login form using HTML and CSS

ChatGPT Output:

● ● ●                            html  

```
<form class="login-form">
 <label>Email:</label>
 <input type="email" placeholder="Enter email" />
 <label>Password:</label>
 <input type="password" placeholder="Enter password" />
 <button>Login</button>
</form>
```

You can also say:

- “Make a pricing card with CSS only”
- “Create a navbar with hover dropdown”

### 10.2 Tailwind CSS Components

- Tailwind is a CSS framework where you use small utility classes.

Example Prompt:

● ● ●                            cpp  
Build a card UI using Tailwind CSS

Output:

● ● ●                            html  

```
<div class="max-w-sm p-4 rounded-lg shadow-md bg-white">
 <h2 class="text-xl font-bold">Product Name</h2>
 <p class="text-gray-500">Short description</p>
 <button class="mt-4 px-4 py-2 bg-blue-500 text-white rounded">Buy Now</button>
</div>
```

You can also ask:

- “Make a responsive navbar using Tailwind”
- “Create a login form in Tailwind CSS”

### 10.3 Responsive Layout Generation

What is responsive design?

- It means your page works well on both mobile and desktop.



## 10. FRONTEND / UI HELP

### 10.3 Responsive Layout Generation

- Example Prompt:

```
● ● ● sql
Create a 2-column responsive layout with HTML and CSS
```

ChatGPT Output:

```
● ● ● css
 @media (min-width: 768px) {
 .container {
 display: flex;
 }
 }
```

Ask for:

- Grid layouts
- Flexbox layouts
- Mobile-first design
- Tailwind responsive classes like md:flex, lg:grid

### 10.4 GSAP / Framer Motion Animations

- GSAP (GreenSock) → For vanilla JS animation
- Framer Motion → For React animations

Example: GSAP

```
● ● ● css
Create a GSAP animation for a box to move from left to right

● ● ● js
gsap.to(".box", { x: 300, duration: 1 });
```

Example: Framer Motion

```
● ● ● css
Add a fade-in animation to a React component using Framer Motion

● ● ● jsx
<motion.div initial={{ opacity: 0 }} animate={{ opacity: 1 }} transition={{
duration: 1 }}>
 Hello
</motion.div>
```

Ask for:

- Slide in / fade in effects
- Scroll-based animation
- Button hover animations



## 10. FRONTEND / UI HELP

### 10.5 Accessibility (a11y) Improvements

What is a11y?

- It means making your website usable for everyone, including people with disabilities.

Ask ChatGPT:



css

How to improve accessibility in this form?

ChatGPT might suggest:

- Use <label for="id">
- Add alt to images
- Use correct heading order (<h1>, <h2>)
- Add ARIA labels for screen readers

Example Fix:



html

```
<label for="email">Email:</label>
<input id="email" type="email" />
```

Other ideas:

- “Make this button keyboard accessible”
- “Add ARIA tags for screen readers”



## 11. TESTING

- Learn how to ask ChatGPT to help you write and understand tests for your code.

### 11.1 Unit Test Generation (Jest, PyTest, JUnit)

- Unit tests check if small parts of your code (like one function) work correctly.
- Jest (JavaScript)

Example Prompt:

```
● ● ● css
Write a Jest test for this function:
function add(a, b) {
 return a + b;
}
```

Output:

```
● ● ● js
test('adds 2 + 3 to equal 5', () => {
 expect(add(2, 3)).toBe(5);
});
```

You can also ask:

- “Test edge cases”
- “Test for negative numbers”

PyTest (Python)

Example Prompt:

```
● ● ● kotlin
Write a PyTest for this function:
def multiply(a, b):
 return a * b
```

Output:

```
● ● ● python
def test_multiply():
 assert multiply(2, 3) == 6
```

JUnit (Java)

Example Prompt:

```
● ● ● pgsql
Write a JUnit test for a sum() method
```



## 11. TESTING

### Unit (Java)

Example Prompt:

```
● ● ● pgsql
Write a JUnit test for a sum() method
```

Output:

```
● ● ● java

@Test
public void testSum() {
 assertEquals(5, sum(2, 3));
}
```

### 11.2 End-to-End (E2E) Test Flows (Cypress, Selenium)

- E2E tests check the entire user experience — from clicking buttons to submitting forms.
- Cypress (Frontend JS testing)

Example Prompt:

```
● ● ● pgsql
Write a Cypress test for login form with email and password
```

Output:

```
● ● ● JS

describe('Login Test', () => {
 it('logs in successfully', () => {
 cy.visit('/login');
 cy.get('#email').type('test@example.com');
 cy.get('#password').type('123456');
 cy.get('button').click();
 cy.url().should('include', '/dashboard');
 });
});
```

### Selenium (cross-browser testing)

Example Prompt:

```
● ● ● pgsql
Use Selenium in Python to test Google search
```

Output:

```
● ● ● python

from selenium import webdriver

driver = webdriver.Chrome()
driver.get("https://google.com")
driver.find_element("name", "q").send_keys("ChatGPT")
driver.find_element("name", "btnK").click()
```



## 11. TESTING

### 11.3 Test Data: Mocks and Stubs

- Mocks and stubs simulate real data or functions in tests.
- Example: Mocking a function in Jest

Prompt:



bash

```
Mock a function that fetches data in Jest
```



js

```
const fetchData = jest.fn(() => Promise.resolve("mocked data"));
```

Example: Using Pytest fixture to fake data

Prompt:



css

```
Mock a database response in PyTest
```



python

```
@pytest.fixture
def fake_user():
 return {"name": "Test", "email": "test@example.com"}
```

Use mocks when:

- You don't want to call the real API
- You want to test faster
- You want to avoid side effects (like deleting real files)

### 11.4 TDD Workflow with ChatGPT Prompts

- What is TDD?
- Test-Driven Development = Write the test first, then write the code to pass the test.

Ask ChatGPT:



vbnet

```
Help me write tests first using TDD. I want to build a function to reverse a string.
```

ChatGPT will respond like:

Step 1: Write the test



js

```
test('reverses the string hello to olleh', () => {
 expect(reverseString("hello")).toBe("olleh");
});
```



## 11. TESTING

Step 2: Then write the function

```
● ● ● js
function reverseString(str) {
 return str.split("").reverse().join("");
}
```

You can do this in:

- Python with PyTest
- Java with JUnit
- JavaScript with Jest



## 12. AI & DATA PROJECTS

- Learn how to use ChatGPT to get help with data analysis, machine learning, NLP, and image processing.

### 12.1 NumPy, pandas, scikit-learn Helpers

These are popular Python libraries for data science:

- NumPy → For arrays and math
- pandas → For data tables
- scikit-learn (sklearn) → For machine learning

#### NumPy Examples

Prompt:



pgsql

```
Show how to create a NumPy array and do basic math
```

Output:



python

```
import numpy as np

a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

print(a + b) # [5 7 9]
```

#### pandas Examples

Prompt:



sql

```
Create a DataFrame and filter rows where age > 25
```



python

```
import pandas as pd

df = pd.DataFrame({
 'name': ['Alice', 'Bob', 'Charlie'],
 'age': [24, 30, 22]
})

print(df[df['age'] > 25])
```

#### scikit-learn Example

Prompt:



css

```
Train a decision tree on a small dataset
```



## 12. AI & DATA PROJECTS

### scikit-learn Example

Prompt:

```
● ● ● python
from sklearn.tree import DecisionTreeClassifier
X = [[0, 0], [1, 1]]
y = [0, 1]
model = DecisionTreeClassifier()
model.fit(X, y)
```

### 12.2 NLP and Image Processing Examples

- NLP (Natural Language Processing)

Prompt:

```
● ● ● css
Use spaCy to extract nouns from a sentence
```

Output:

```
● ● ● python
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("ChatGPT is helpful for developers.")

nouns = [token.text for token in doc if token.pos_ == "NOUN"]
print(nouns) # ['developers']
```

Other NLP tasks ChatGPT can help with:

- Text classification
- Sentiment analysis
- Summarization

### Image Processing

Prompt:

```
● ● ● arduino
Use OpenCV to read and resize an image
```

```
● ● ● python
import cv2

img = cv2.imread('image.jpg')
resized = cv2.resize(img, (100, 100))
cv2.imwrite('resized.jpg', resized)
```

You can ask for:

- Grayscale conversion
- Face detection
- Drawing shapes on images



## 12. AI & DATA PROJECTS

### 12.3 Dataset Generation and Augmentation

- Dataset Generation (Synthetic)

Prompt:



css

```
Use scikit-learn to generate a classification dataset
```



python

```
from sklearn.datasets import make_classification

X, y = make_classification(n_samples=100, n_features=2, n_classes=2)
```

- Data Augmentation (Images)

Prompt:



arduino

```
Augment image data using TensorFlow or PyTorch
```

Output (TensorFlow):



python

```
import tensorflow as tf

image = tf.io.read_file('img.jpg')
image = tf.image.decode_jpeg(image)
image = tf.image.random_flip_left_right(image)
image = tf.image.random_brightness(image, max_delta=0.2)
```

### 12.4 Code Explanation for ML Pipelines

- If you're confused by a pipeline or ML model, just ask:

Prompt:



arduino

```
Explain this ML pipeline step-by-step:
[Paste code]
```

ChatGPT can:

- Break down fit(), predict(), transform()
- Explain preprocessing steps
- Describe model evaluation

Example Prompt:



css

```
Explain this sklearn pipeline:
Pipeline([
 ('scaler', StandardScaler()),
 ('model', LogisticRegression())
])
```



## 12. AI & DATA PROJECTS

ChatGPT Output:

- First scales data using StandardScaler
- Then trains logistic regression on scaled data



## 13. PROJECT ASSISTANCE

- Learn how to ask ChatGPT to help with project ideas, building your MVP (Minimum Viable Product), planning features, and tracking bugs.

### 13.1 Project Ideas Based on Skill Level

- You can ask ChatGPT:



pqsql

Give me 5 project ideas for my skill level in [language/tool]

Examples:

Beginner in HTML/CSS:

- Personal Portfolio Website
- Restaurant Menu Page
- Login Form UI
- Basic Blog Page
- Responsive Landing Page

Intermediate in JavaScript:

- To-Do List with LocalStorage
- Calculator App
- Weather App using API
- Quiz Game
- Expense Tracker

Advanced in React/Node:

- Real-time Chat App (with Socket.io)
- Job Portal
- E-commerce Dashboard
- Blogging Platform with Login
- AI-powered note taker using OpenAI API

You can also say:



vbnet

Give me frontend + backend project ideas for 2 people

### 13.2 How to Ask ChatGPT to Build MVP Step-by-Step

- What is MVP?
- MVP means Minimum Viable Product — a very simple working version of your project.

Prompt:



vbnet

Help me build a simple notes app with login – step by step



## 13. PROJECT ASSISTANCE

ChatGPT might reply with steps like:

1. Set up project folder with HTML, CSS, JS
2. Create UI: input, textarea, save button
3. Store notes in local storage
4. Add basic login with dummy credentials
5. Save notes per user

Then you can ask:

```
● ● ● vbnet
Give me code for step 1
```

...and continue one step at a time.

### 13.3 Feature Planning and Task Breakdown

Prompt:

```
● ● ● css
I want to build a task manager app. Help me break it into features and tasks.
```

ChatGPT might give you:

Features:

- User registration/login
- Add/edit/delete tasks
- Set due date and priority
- Mark tasks as complete
- Filter by status

Tasks:

- Create login page
- Connect frontend to backend API
- Store tasks in database
- Add filters on UI
- Add authentication with JWT

You can also ask:

```
● ● ● csharp
Make this list into a Trello-style task board
```

### 13.4 Bug Tracking and Feature Refinement with Prompts

- If something is broken or you want to improve a feature, just say:

Example:

```
● ● ● less
In my to-do app, the delete button doesn't work. Here's the code:
[Paste code]
Why doesn't it work? How can I fix it?
```



## 13. PROJECT ASSISTANCE

ChatGPT will:

- Debug your code
- Suggest improvements
- Help track fixed vs. pending issues

You can also ask:



cpp

```
Suggest 5 new features to improve this app
```

ChatGPT might reply:

- Add search bar
- Enable dark mode
- Sort tasks by date
- Export task list as PDF
- Add voice input for new tasks



## 14. INTEGRATIONS & PLUGINS

- Learn how to use ChatGPT with your favorite tools like VS Code, Postman, GitHub, and even make your own custom GPT.

### 14.1 ChatGPT for VS Code / Cursor IDE

What it does:

- Writes code while you type
- Explains code blocks
- Helps with debugging
- Adds comments and docstrings

#### A. ChatGPT in VS Code

How to install:

1. Open VS Code
2. Go to Extensions tab (left sidebar)
3. Search: ChatGPT - Code Companion
4. Click Install
5. Sign in with your OpenAI account

What you can do after:

- Right-click code → "Explain code"
- Select code → "Fix this code"
- Ask: "Refactor this" or "Optimize it"

#### B. Cursor IDE (Built-in ChatGPT)

- Cursor is a full IDE like VS Code, but it has ChatGPT built-in and better integrated.
- Website: <https://www.cursor.so>

Best features:

- Ask questions inside your code
- Autocomplete using GPT
- Highlight bugs with natural-language explanations
- Fast responses without copying code manually

### 14.2 Using ChatGPT in Postman

- Postman is used for testing APIs. You can now add ChatGPT directly to help:

Option 1: Use Postbot (AI in Postman)

- Built-in tool called Postbot helps generate API test cases
- Writes test scripts in JavaScript
- Helps debug errors from API calls



## 14. INTEGRATIONS & PLUGINS

### Option 2: Use ChatGPT externally

Prompt Example:

```
● ● ● pgsql
Write a Postman test that checks if the response status is 200 and has a "user" field
```

ChatGPT Output:

```
● ● ● js

pm.test("Status is 200", function () {
 pm.response.to.have.status(200);
});

pm.test("Response has user", function () {
 var jsonData = pm.response.json();
 pm.expect(jsonData).to.have.property("user");
});
```

### 14.3 GitHub Copilot vs ChatGPT

Feature	GitHub Copilot	ChatGPT
Works in IDE?	Yes (VS Code, JetBrains)	Yes (with plugin)
Writes code while typing	✓	✗ (unless in Cursor or via plugin)
Explains code	✗	✓
Refactors/debugs on request	✗	✓
Chat interface	✗	✓
Good for conversations	✗	✓ (step-by-step help)

- Use both together for best results!

### 14.4 Creating Your Own Custom GPT for Dev Workflows

- You can make a Custom GPT that acts like your personal coding assistant.

Use Case Examples:

- A GPT that explains your company's codebase
- A GPT that helps only with HTML + Tailwind
- A GPT that responds in Gujarati for devs



## 14. INTEGRATIONS & PLUGINS

### How to create one:

1. Go to <https://chat.openai.com/gpts>
2. Click Create a GPT
3. Use the GPT Builder to guide setup
4. Set instructions like:
  - “You are a code helper for JavaScript projects”
  - “Respond with simple explanations like for 4th graders”
5. Upload files, docs, or code samples if needed
6. Share with your dev team or community

Example Prompt to Create Custom GPT:

A screenshot of a dark-themed application window. In the top left corner, there are three colored dots (red, yellow, green). In the top right corner, the word "css" is written. The main area contains a text input field with the following content:

```
Help me build a GPT that explains React code and gives clean examples using Tailwind CSS. Keep language easy for beginners.
```



## 15. INTERVIEW PREP & CAREER HELP

- Get help from ChatGPT for coding interviews, resumes, GitHub, and creating your own portfolio site.

### 15.1 Solving DSA (Data Structures & Algorithms) Questions

What you can ask:

- "Give me easy DSA questions in Python/JavaScript"
- "Explain binary search with dry run and code"
- "Solve this LeetCode problem step-by-step"

Example Prompt:

```
● ● ● sql
Explain how binary search works with code and dry run
```

ChatGPT Output (Short Version):

```
● ● ● python

def binary_search(arr, target):
 left, right = 0, len(arr) - 1
 while left <= right:
 mid = (left + right) // 2
 if arr[mid] == target:
 return mid
 elif arr[mid] < target:
 left = mid + 1
 else:
 right = mid - 1
 return -1
```

- Then it explains step-by-step how the value is searched in the list.

### 15.2 Simulating System Design Interviews

- System design = how big apps work, like WhatsApp, Netflix, etc.

What you can ask:

- "Simulate a system design round for me"
- "How to design a URL shortener"
- "What are the key components of designing an e-commerce app?"

Example Prompt:

```
● ● ● pgsql
Design a URL shortener like bit.ly. Break it into parts.
```

ChatGPT Output:

1. Frontend (input field, short URL display)
2. Backend (API, logic to shorten URLs)
3. Database (store original and short URL)
4. Optional: Analytics, User Login, Expiry



## 15. INTERVIEW PREP & CAREER HELP

### 15.3 Improving GitHub Profile & Resume

GitHub Optimization Prompts:

- "Review my GitHub profile and suggest improvements"
- "Give ideas for good beginner-friendly repositories"
- "Generate a clean README for my JavaScript project"

Example Prompt:



css

Generate a README template for a weather app in JavaScript

Resume Review Prompts:

- "Improve this resume for a front-end role"
- "Rewrite this resume section to make it stronger"
- "Create a resume using only my GitHub projects"

Extra Tip:

You can also ask:



kotlin

Make this resume ATS-friendly

(ATS = applicant tracking system used by HRs)

### 15.4 Creating a Personal Dev Portfolio

ChatGPT can help you:

- Plan the structure
- Design in HTML/CSS or React
- Write content for About, Skills, Projects, Contact
- Add responsiveness and animations

Prompt Example:



csharp

Build a simple portfolio site using HTML, CSS, and JavaScript with dark mode and responsive design

Prompt Example:



pgsql

Write the 'About Me' section for a front-end developer portfolio

#### ChatGPT Output:

- I'm a passionate front-end developer who enjoys turning ideas into interactive web apps...



## 16. BEST PRACTICES & PITFALLS

Learn how to use ChatGPT the right way when writing code, and what to avoid.

### 16.1 Avoiding Over-Reliance

What this means:

- Don't become fully dependent on ChatGPT for every line of code.

Why?

- You may stop learning how things really work
- You might not understand errors when they happen
- Not everything generated is always correct

What to do instead:

- Use ChatGPT as a helper, not a replacement
- Try first → then ask ChatGPT to review or improve
- Learn the "why," not just the "what"

### 16.2 Validating Generated Code

- What does "validate" mean?
- It means you check that the code works and makes sense before using it.

Why it matters:

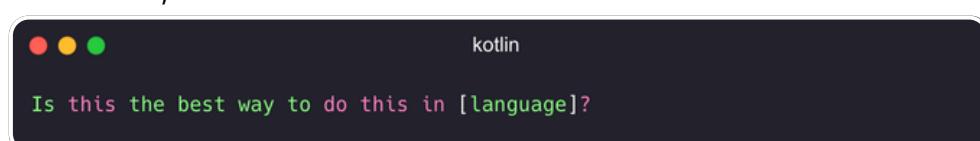
- Sometimes, ChatGPT can make mistakes
- Code might look right but give wrong results

What you should do:

- Run the code on your computer or IDE
- Use console.logs / print statements to test output
- Ask: "Can you explain what this code does line-by-line?"
- Use comments to understand

Extra Tip:

If unsure, ask:



### 16.3 When to Avoid Using ChatGPT

- ChatGPT is not always the right tool.
- Here's when not to use it:

Situation	Why Not to Use ChatGPT
In an exam or test	It's not allowed – learn it yourself
For writing <b>private company code</b>	May not be secure to share
When working with <b>very new tools</b>	ChatGPT may be outdated (not latest versions)
When speed is more important than help	Sometimes searching Google or Stack Overflow is faster



## 16. BEST PRACTICES & PITFALLS

### 16.4 How to Give Clear Context in Prompts

What is "context"?

- It means extra details that help ChatGPT understand your situation better.

Why it's important:

- Better context = better answers

Example 1 — Bad prompt:



kotlin

Fix this code

Example 2 — Good prompt:



bash

This Python function should return the sum of two numbers, but it's giving an error. Can you fix it?

```
def add(a, b)
 return a + b
```

**What to include in your prompt:**

- What the code is supposed to do
- What's not working
- What you've already tried
- The language or framework you're using
- If you want a beginner explanation



## 17. SMART PROMPTS FOR DEVELOPERS – TABLE OF CONTENTS

- These are smart ways to talk to ChatGPT so it can help you code better, faster, and easier.

### 17.1 Code Writing & Generation Prompts

- Prompts that help you write new code in any language.

Example:

- "Write a login form in HTML and CSS"
- "Create a calculator app using React"

### 17.2 Code Refactoring & Optimization Prompts

- Prompts to help make your code cleaner and faster.

Example:

- "Make this Python code more efficient"
- "Refactor this JS function to use async/await"

### 17.3 Bug Fixing & Debugging Prompts

- Prompts to find and fix errors in your code.

Example:

- "Why is this code crashing? Here's the error..."
- "Fix this bug: [Paste code]"

### 17.4 Code Explanation Prompts

- Prompts that explain what your code does in simple terms.

Example:

- "Explain this JavaScript function line-by-line"
- "What does this regex mean?"

### 17.5 Documentation & Commenting Prompts

- Prompts to generate comments or docstrings for your code.

Example:

- "Add comments to this function"
- "Write a Python docstring for this class"

### 17.6 Testing (Unit, Integration, E2E) Prompts

- Prompts to help you write tests for your app or code.

Example:

- "Write a Jest test for this function"
- "Create a Cypress test for login flow"



## 17. SMART PROMPTS FOR DEVELOPERS – TABLE OF CONTENTS

### 17.7 DevOps & CLI Prompts

- Prompts to help you with terminal commands, Docker, CI/CD, etc.
- Example:
- "Create a Dockerfile for my Node app"
- "Write a bash script to zip files and push to Git"

### 17.8 Database Query & Design Prompts

- Prompts for writing SQL queries or designing a database schema.

Example:

- "Write SQL to find users who signed up last week"
- "Create a schema for a blog app"

### 17.9 API Design & Implementation Prompts

- Prompts to help build and test APIs (REST, GraphQL, etc).

Example:

- "Create a REST API with Express and MongoDB"
- "Write Postman test script for status code check"

### 17.10 UI/UX & Frontend Prompts

- Prompts to design beautiful user interfaces and frontend layouts.

Example:

- "Make a responsive navbar using Tailwind CSS"
- "Animate button on hover using CSS"

### 17.11 Learning New Tech Stack Prompts

- Prompts to help you learn new frameworks or tools quickly.

Example:

- "Explain React Hooks with examples"
- "What is Tailwind CSS and how do I use it?"

### 17.12 Data Structures & Algorithms Prompts

- Prompts to learn and practice DSA questions for interviews or school.

Example:

- "Explain binary search with code and dry run"
- "Write a program to reverse a linked list"

### 17.13 Project & MVP Building Prompts

- Prompts to build real-life projects and break them into steps.

Example:

- "Help me build a notes app step-by-step"
- "Break this feature into small tasks"



## 17. SMART PROMPTS FOR DEVELOPERS – TABLE OF CONTENTS

### 17.14 Version Control (Git) Prompts

- Prompts to help you with Git and GitHub commands.

Example:

- "How to resolve a merge conflict?"
- "Write a bash script to auto push to Git"

### 17.15 Productivity & Automation Prompts

- Prompts to make your coding faster using automation scripts or tools.

Example:

- "Automate file renaming with Python"
- "Make a VS Code shortcut to format code"

### 17.16 Resume, GitHub & Portfolio Optimization Prompts

- Prompts to improve your online presence as a developer.

Example:

- "Review my resume for a front-end developer role"
- "Suggest GitHub projects to add to my portfolio"

### 17.17 Interview Preparation Prompts

- Prompts to practice for coding interviews or technical rounds.

Example:

- "Mock interview questions for JavaScript developer"
- "System design interview questions for beginners"

### 17.18 Prompt Engineering for Better Results

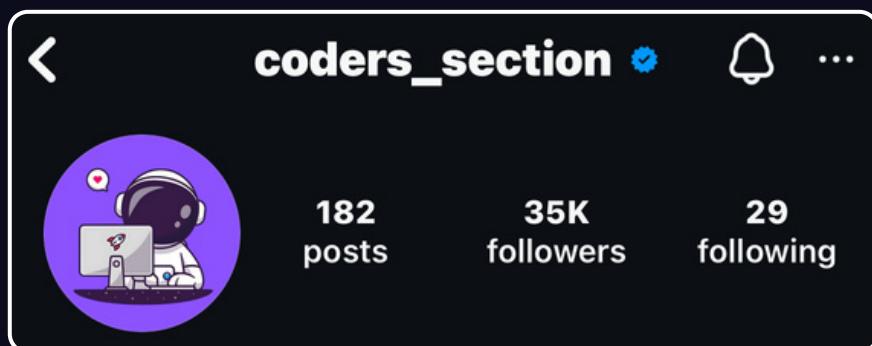
- Prompts to help you ask better prompts — so ChatGPT gives better answers.

Example:

- "How to structure a prompt for generating clean code?"
- "What details should I add to get correct output?"



# Was this post helpful ?



A screenshot of an Instagram profile for the account "coders\_section". The profile picture shows a person wearing a headset and working on a computer. The bio is blank. The statistics are: 182 posts, 35K followers, and 29 following. There is a blue verification checkmark next to the account name.



Follow Our 2<sup>nd</sup> Account



A screenshot of an Instagram profile for the account "webstrom\_tech". The profile picture shows the Webstrom Tech logo. The bio reads "Coding | Programming | HTML | CSS". The statistics are: 8 posts, 457 followers, and 1 following. A blue "NEW" badge is visible at the bottom left of the profile picture.



## Follow For More

