# **ChatGPT Cheatsheet and Prompts**

# **1. Golden Rules**

1. **Always test your code! DON’T RELY COMPLETELY ON CHATGPT**
2. Use ChatGPT ! **ESPECIALLY** if you don’t know or can’t remember something
   1. **DO NOT OVER use ChatGPT (do not pass page 2 of results)**
   2. **DO NOT USE CODE YOU DO NOT UNDERSTAND**
   3. **DO NOT COPY PASTE FROM ChatGPT! (read the code, understand it, and recreate it your own way)**
3. Make mistakes! **Seriously, don’t be scared of making mistakes**. The worst thing that can happen is you go back and fix it. Mistakes are a great way to learn (it’s called **trial and error** for a reason).
4. Good code is lazy and efficient! **Efficiency and laziness go hand in hand.**
5. Ask your classmates! There is no shame in not knowing something or asking for help. **Leverage the community!**
6. Hop in the queue! If you get stuck on something for 15 minutes or more, ask us.
7. Never overestimate how much time you have!
8. Don’t Panic!
9. **Don’t use terminal commands you don’t know!**

# **2. ChatGPT Golden Shortcuts**

Using shortcuts or keyboard commands with ChatGPT can enhance your learning experience. Here are some handy shortcuts for learning web development with ChatGPT:

#### **2.1 Context Setup:**

**/context` or `/set-context:** Set the context by specifying the topic you want to discuss with ChatGPT. For example: **`/context HTML5` .**

#### **2.2 Code Formatting:**

Surround your code with **triple backticks (```)** to format it as a code block.

#### **2.3 Requesting Code Examples:**

Ask for specific code examples related to your learning needs. For example, **"Code example for a responsive CSS grid."**

#### **2.4 Learning Resources:**

Request information about specific web development topics or concepts. For instance**, "Learn more about CSS Flexbox."**

#### **2.5 Debugging Assistance:**

**Debug my code…:** Seek help in identifying and resolving coding issues. Share your code and describe the problem you're facing.

#### **2.6 Clarify Concepts:**

Ask for explanations of web development concepts or terminology. For example, **"Explain the concept of 'Express'."**

#### **2.7 Best Practices:**

**`Best practices for …`** Inquire about best practices for web development tasks like SEO, responsive design, or accessibility.

#### **2.8 Quick Tips:**

**Quick tip for ...:** Request quick tips or tricks related to specific coding tasks or challenges.

#### **2.9 Project Ideas:**

**Project idea for ...:** Get suggestions for web development projects that match your skill level and interests.

#### **2.10 Comparison Queries:**

**Compare ... and ...:** Compare different technologies, libraries, or frameworks. For example, **"Compare React and Vue.js."**

#### **2.11 Resource Recommendations:**

**Recommend a book/tutorial for ...:** Ask for book or tutorial recommendations tailored to your learning goals.

#### **2.12 Advanced Queries**:

For advanced learners, you can ask complex questions and discuss advanced topics by providing context and using clear language.

#### **2.13 Extra info**

* Share the code that you want to review with ChatGPT.
* Ask ChatGPT specific questions about the code you want to review. For example:

**"Can you check if all the HTML tags are properly closed?"**

**"Are there any syntax errors in this CSS code?"**

* You can also ask for best practices or recommendations related to the topic. For instance:

**"Are there any accessibility improvements I can make to this \_\_\_\_\_\_ code?"**

**"What are some common design principles to consider for this \_\_\_\_\_\_?"**

* If ChatGPT provides feedback or suggestions, you can ask follow-up questions for clarification or additional guidance.
* Carefully review the responses provided by ChatGPT. Keep in mind that ChatGPT can offer suggestions and identify potential issues.
* If there are any error messages or unexpected behavior you've observed, include them
* Ask for guidance on debugging this specific code snippet. You can request a step-by-step explanation or advice on how to identify the issue

# **3. List of Prompts**

Given below is a list of prompts and topics that can be helpful for a beginner who has recently started earning web development

#### **3.1 HTML (HyperText Markup Language):**

* What is HTML, and why is it important in web development?
* How do you create an HTML document structure (HTML5)?
* Explain the basic HTML tags like <html>, <head>, <title>, <body>, and <p>.
* What are HTML attributes and how do you use them?
* How to create hyperlinks and anchor tags (<a>) in HTML?
* How to create lists (<ul>, <ol>, <li>) in HTML?
* What are HTML forms, and how do you create them?
* Describe the role of semantic HTML elements like <header>, <nav>, <main>, <section>, <article>, and <footer>.
* Explain the purpose of HTML tables (<table>, <tr>, <td>) and how to create them.
* How to include images (<img>) in HTML documents?

#### **3.2 CSS (Cascading Style Sheets):**

* What is CSS, and why is it used in web development?
* How to apply inline, internal, and external CSS styles?
* Describe CSS selectors and how they target HTML elements.
* Explain the box model in CSS and how it affects layout.
* What are CSS properties like margin, padding, border, width, and height used for?
* How to style text and fonts using CSS?
* What is CSS Flexbox, and how can it be used for layout?
* Describe CSS Grid and its advantages in creating responsive layouts.
* What are CSS transitions and animations, and how do you implement them?
* How to make a website responsive using media queries and CSS?

#### **3.3 JavaScript:**

* What is JavaScript, and how does it differ from HTML and CSS
* How to include JavaScript in an HTML document?
* Explain variables, data types, and basic operators in JavaScript.
* What are control structures like if, else, and switch used for in JavaScript?
* How to create loops (e.g., for and while) in JavaScript?
* Describe functions and how to define and call them in JavaScript.
* What is the Document Object Model (DOM) and how does JavaScript interact with it?
* How to select and manipulate DOM elements using JavaScript?
* What are JavaScript events and event listeners?
* How to handle asynchronous operations using callbacks and Promises?

#### 

#### **3.4 React:**

* What is React, and why is it popular for building user interfaces?
* What is the Document Object Model (DOM) and how does JavaScript interact with it?
* How to select and manipulate DOM elements using JavaScript?
* What are JavaScript events and event listeners?
* How to handle asynchronous operations using callbacks and Promises?
* Explain the concept of components in React.
* How to create functional components in React.
* What are React props, and how do you pass data to child components?
* Describe the useState and useEffect hooks in React.
* What is React Router, and how can it be used for client-side routing?
* How to manage form state and user input in React.
* Explain the useContext hook and its use in state management.
* How to make API requests in React using the Fetch API or third-party libraries

#### **3.5 Node.js and Express.js:**

* What is Node.js, and how does it differ from traditional JavaScript?
* How to create a basic server using Node.js and Express.js.
* What are HTTP methods (GET, POST, PUT, DELETE) and how are they used in web development?
* Describe RESTful API design principles.
* How to handle route parameters and query parameters in Express.js.
* What is middleware in Express.js, and how can it be used for request processing?
* Explain the concept of authentication and authorization in Express.js.
* How to handle file uploads in a Node.js/Express.js application.
* What are web sockets, and how can you implement real-time communication in Node.js?
* How to deploy a Node.js/Express.js application to a hosting platform like Vercel or Render.

#### **3.6 MongoDB Atlas and Database Operations:**

* What is MongoDB, and why is it used as a NoSQL database?
* How to set up a MongoDB Atlas cluster for cloud-based database hosting.
* Describe CRUD (Create, Read, Update, Delete) operations in the context of MongoDB.
* How to connect a Node.js application to MongoDB Atlas.
* How to perform database operations like inserting documents, querying data, updating documents, and deleting documents.
* How to implement pagination and sorting in MongoDB queries.
* What is indexing in MongoDB, and how can it improve query performance?
* How to handle database errors and implement error handling in a Node.js application.
* Security best practices when working with databases and user data.

# **3. Bonus Links**

* [Chat GPT Repos- Github](https://github.com/taishi-i/awesome-ChatGPT-repositories)
* [ChatGPT Prompts - github](https://github.com/prathamkumar14/chatgpt-prompts)
* [Sider: ChatGPT Sidebar, GPT-4, Draw & Web](https://chrome.google.com/webstore/detail/sider-chatgpt-sidebar-gpt/difoiogjjojoaoomphldepapgpbgkhkb)

</nav>

<main>

<!-- Main content goes here -->

</main