

The Developer's Guide to ChatGPT

Enhancing Your Skills with AI



C#Corner

Author
Chris Love

The Developer's Guide to ChatGPT: Enhancing Your Skills with AI

All rights reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

Although the author and publisher have made every effort to ensure that the information in this book was correct at press time, the author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

The resources in this book are provided for informational purposes only and should not be used to replace the specialized training and professional judgment of a health care or mental health care professional.

Neither the author nor the publisher can be held responsible for the use of the information provided within this book. Please always consult a trained professional before making any decision regarding the treatment of yourself or others.

Author – Chris Love

Publisher – C# Corner

Editorial Team – Archie Vankar, Deepak Tewatia

Publishing Team – Praveen Kumar, Amit Kumar

Promotional & Media – Rohit Tomar, Rohit Sharma

Author Bio



Chris Love is a seasoned front-end developer with over 25 years of experience building a wide variety of websites and applications. He's passionate about helping people and companies navigate the complex world of the modern web and user experience standards. In recent years, he's focused his attention on progressive web apps and web performance optimization, applying his expertise to run a successful web consulting company called Love2Dev.

As the author of four web development books, including the acclaimed High-Performance Single Page Web Applications (<http://amzn.to/1b0twcm>), Chris is recognized as a leading authority in the industry. He's also a 14-time Microsoft MVP in ASP.NET and Web Platform and has spoken at hundreds of user groups and conferences around the world. In addition to writing and speaking, Chris has produced over 1000 blog articles, magazine articles, and videos, sharing his insights and opinions on a range of web development topics.

Over the course of his career, Chris has helped over 1000 businesses leverage the power of the web to engage with customers and streamline worker productivity. His work has been featured in various magazines, popular podcasts, and web sites. He takes pride in shaking up his clients' approach to web development, while making the process fun and engaging.

When he's not working on front-end development, Chris enjoys spending time with his step-kids, learning karate, and training for Spartan races. You can follow Chris' blog at love2dev.com and on Twitter @ChrisLove for more insights and tips on modern web technology.

Acknowledgement

I want to express my heartfelt gratitude to the wonderful team at C# Corner for their unwavering support and encouragement of developers worldwide. Their platform has been a tremendous resource for online education and networking, empowering countless individuals to pursue their passion for technology.

I also believe that any book on the cutting-edge topic of AI would be incomplete without acknowledging the brilliant minds behind the technology. I am humbled by the incredible work of engineers at companies like OpenAI, Microsoft, and many others, who are using their skills to improve the lives of people around the world. Their dedication and innovation are an inspiration to us all.

- Chris Love

Table of Contents:

| | |
|--|----|
| Introduction..... | 5 |
| Chapter 1 | 6 |
| What is ChatGPT..... | 6 |
| Can ChatGPT and other AI Tools Replace Software Developers?..... | 7 |
| Getting Started with ChatGPT..... | 8 |
| ChatGPT UI | 8 |
| Capabilities of ChatGPT | 9 |
| ChatGPT have possesses capabilities:..... | 10 |
| Chapter 2 | 11 |
| Prompt in ChatGPT..... | 11 |
| How to Design Effective Prompts..... | 11 |
| Chapter 3 | 15 |
| Generating Code with ChatGPT..... | 15 |
| Generating SQL Queries..... | 15 |
| Generating Individual Functions..... | 17 |
| Creating REST APIs | 18 |
| Creating Unit and Integration Tests..... | 18 |
| Generating Test Cases | 19 |
| Test Automations..... | 20 |
| Generating Code Documentation..... | 20 |
| Creating User Interfaces | 23 |
| Prompt Examples:..... | 23 |
| Updating Legacy Code | 23 |
| Decoding Cryptic Error Messages..... | 25 |
| Chapter 4 | 28 |
| Jobs and Careers..... | 28 |
| Conclusion | 34 |

Introduction



Fig 1

Artificial intelligence (AI) is rapidly transforming the world we live in, and nowhere is this more evident than in the field of software development. The emergence of AI tools has opened up new possibilities for increased productivity and quality in software development workflows. One of the most exciting tools to emerge in recent years is ChatGPT, a powerful language model developed by OpenAI.

ChatGPT has experienced rapid growth since its launch, shattering records for other companies and technologies in terms of user adoption. In fact, it reached 100 million users in less than 60 days, a stunning achievement that demonstrates the power and potential of this AI tool. This explosive growth is a testament to the value that ChatGPT provides to developers, who are using it to generate code, write documentation, and answer common questions.

In this eBook, we'll explore the capabilities of ChatGPT for software development. We'll discuss how these tools can help developers increase their productivity, improve code quality, and streamline their workflows. We'll also provide guidance on how to use these tools effectively and efficiently and discuss best practices for integrating them into your software development process.

As the world of software development continues to evolve, staying up-to-date with the latest technologies is essential for remaining competitive. We hope this eBook will provide a valuable resource for developers looking to explore the latest AI tools and take their software development workflows to the next level.

Chapter 1

What is ChatGPT

ChatGPT is a powerful language model developed by OpenAI that can understand and generate human-like language. It is designed to assist in natural language processing tasks such as language translation, text summarization, and conversational agents. Essentially, it allows us to communicate with machines using normal conversational language in a more effective and efficient manner.



Fig 1.1

ChatGPT is a super-smart computer system that can talk to humans using normal language, just like we talk to each other. This system is called an AI language model and it can understand the words and sentences we say to it, and then give us really helpful answers in a very short amount of time. Think of it like having a really smart friend who knows a lot about a lot of different things, and can help us find the answers to our questions quickly and easily. Without ChatGPT, we might have to do a lot of research and spend a lot of time looking for the answers ourselves, but with it, we can get the information we need right away.

ChatGPT is great because it can improve the workflow, quality and much more for almost any profession. Chefs and home cooks can create or find new recipes. Authors can get ideas, feedback and drafts for books and articles. It can be a research assistant. We can have it write emails, tweets and even draft legal documents. It can also translate legalese so everyone can understand laws and statutes.

For programmers it can even write, test, comment and much more with our code.

Before I dive into how software developers can leverage ChatGPT, I want to address a big question, can ChatGPT and other AI tools replace software developers?

Can ChatGPT and other AI Tools Replace Software Developers?

As the capabilities of AI tools like ChatGPT and GitHub Copilot continue to expand, some are wondering whether these tools could eventually replace human software developers. While it's true that AI tools can automate many tasks that were previously performed by humans, there are several reasons why AI is unlikely to replace developers entirely.

Firstly, AI tools are not capable of understanding the complex nuances of software development in the same way that human developers are. While AI can perform tasks based on specific prompts or instructions, it doesn't have the same level of intuition or creativity that humans possess. This means that while AI can automate certain tasks, it can't replace the judgment and decision-making skills of human developers.

Secondly, software development is a highly collaborative field that involves a great deal of communication and teamwork. While AI tools can assist with specific tasks, they can't replicate the human-to-human interactions that are essential for successful software development projects. The ability to communicate effectively, collaborate with others, and understand the needs of end-users is a critical part of software development that AI tools can't replace.

Finally, at this point in time at least, AI is not capable of generating a complete application from a prompt. At least an application beyond a very basic example. Of course you could string together a series of prompts to create portions of the application, a person would still be needed to assemble the application and test it before shipping.

As you will see as this book progresses, using an AI tool like ChatGPT to generate good code requires a skilled developer to know how to ask the model to do the desired task correctly.

It's important to note that ChatGPT is an AI tool and, like any tool, it can sometimes generate incorrect or suboptimal results. While ChatGPT can be a useful resource for troubleshooting error messages or updating legacy code, it's important to exercise caution and not blindly accept its suggestions without careful consideration.

I've personally encountered situations where ChatGPT has generated poor code that doesn't work as intended. In these cases, it can be helpful to go back and forth with the model until you arrive at a correct solution. This process isn't much different from a normal development process, and I've found that it can often save time and effort compared to making similar mistakes on my own.

In summary, while ChatGPT can be a helpful tool for developers, it's important to use it judiciously and exercise caution when applying its suggestions. By working collaboratively with ChatGPT and verifying its suggestions through testing and careful evaluation, you can arrive at solutions that are both effective and efficient.

Despite these limitations, there's no doubt that AI tools will continue to change the way we develop software in the future. AI can help developers automate repetitive or time-consuming tasks, and can

provide new insights and perspectives on software development projects. As AI tools become more sophisticated, we can expect them to play an increasingly important role in software development workflows.

However, it's important to remember that AI tools are not a replacement for human developers. Rather, they are a valuable tool that can assist developers in performing certain tasks more efficiently and effectively. As the field of software development continues to evolve, it's likely that AI tools will become even more integral to the development process, but they will always work in concert with human developers rather than replacing them entirely.

I think developers that master how to use AI tools to create better applications will thrive. Those that don't will wither away over time. This is a process that has repeated itself throughout time in every profession. It is not developers that will be replaced, it is how we have written code that will be upgraded.

Getting Started with ChatGPT

ChatGPT is a powerful language model developed by OpenAI that can be used for a wide range of tasks related to natural language processing.

In software development workflows, ChatGPT can be particularly useful for generating code, writing documentation, and answering common questions. In this section, we'll provide an introduction to ChatGPT and its capabilities, as well as guidance on how to get started using the tool effectively.

If you have not yet started using ChatGPT, you can visit <https://chat.openai.com/>. From there you can create an account. ChatGPT is free to use, but with some limitations. You won't have access to the latest models, for example GPT-4 is the most recent model. You will also find the service is slow or unresponsive during certain times of the day.

A pro version, or account is available for \$20 per month (at the time of writing). This will give you access to the latest models early as well as faster response times and priority. Since upgrading to the paid account I do not find the service slow or unresponsive.

ChatGPT UI

Navigating the ChatGPT user interface is straightforward, but it's helpful to know what to expect. On the left-hand side, you'll see a list of your chat sessions. If you haven't started a chat, the list will be empty. To initiate a new chat, simply click the "New Chat" button in the top left.

One of the benefits of ChatGPT is its ability to retain context within a chat session. You can have multiple "active" chat sessions on different topics.

To begin a conversation with ChatGPT, enter your prompt at the bottom of the page. The text box here is labeled "Send a message," but it's actually where you'll enter your prompt. To submit your prompt, click the right-facing arrow or press ENTER.

If you need to include a line break in your prompt, you can press Shift + ENTER. Once you've submitted your prompt, it will appear in the chat conversation above ChatGPT's response. The prompt text box will stay at the bottom of the page, while the chat content will scroll up. If the chat is longer than the available screen and you've scrolled up, you can quickly return to the last response by pressing the down arrow key to the right of the chat conversation.

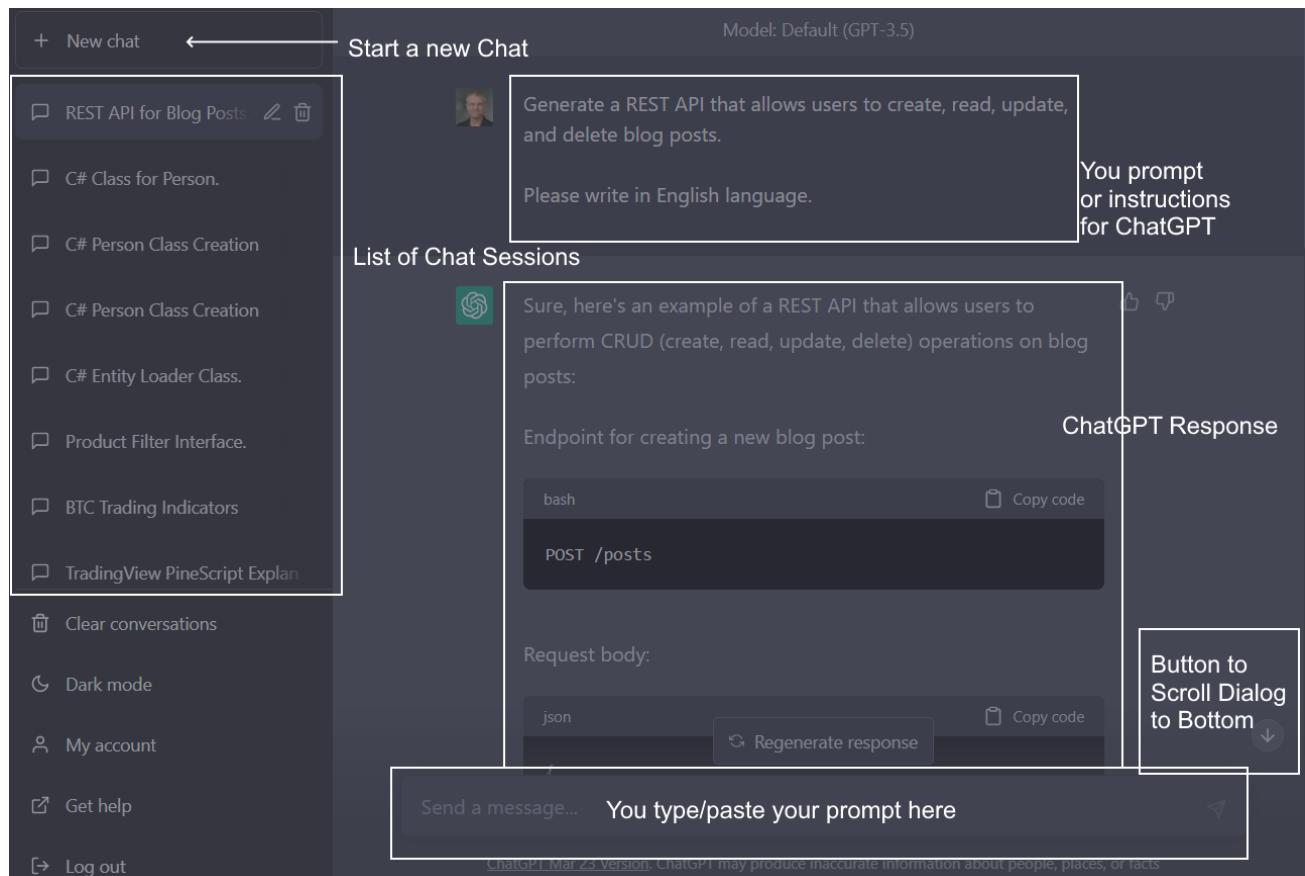


Fig 1.2

Capabilities of ChatGPT

ChatGPT is a language model that is built on the same technology as the highly successful GPT-3 model. This means that ChatGPT has access to a vast corpus of text data, which allows it to generate highly accurate and contextually relevant responses to specific prompts.

Although GPT-3 is the current model, OpenAI has released GPT-4 to pro subscribers. GPT-4 has an even better model than its predecessor, but it's currently a bit slower as the OpenAI team scales it to serve a larger subscriber base. As with any new technology, it's important to expect similar experiences with newer models as they are released.

ChatGPT is a powerful tool that can provide highly accurate responses to a wide range of prompts. With access to a vast corpus of text data and ongoing improvements in AI technology, ChatGPT and other language models are likely to continue to evolve and improve over time.

GPT stands for “Generative Pre-trained Transformer”. It’s a fancy way of saying that it’s a computer program that has been trained to understand and generate human-like language. Think of it like a really smart robot that can understand what you’re saying and respond in a way that makes sense, just like how you talk to your friends.

ChatGPT have possesses capabilities:

- Language Generation
- Conversational Engagement
- Information Retrieval
- Creative Writing
- Language Translation
- Answering Questions
- Task-based Assistance
- Text Completion and Suggestion
- Understanding Context and Intent

Chapter 2

Prompt in ChatGPT

When using ChatGPT, a prompt is a piece of text that is provided to the language model in order to generate a response. A prompt can be thought of as a question or statement that provides context for the response that the model generates.

Before a prompt is provided to the model, it is tokenized. Tokenization is the process of breaking a piece of text into smaller units called tokens. Each token represents a word or a piece of punctuation, and the tokens are used to represent the meaning of the text.

ChatGPT has a limit of 4000 tokens for both the prompt and response. This means that the length of the prompt cannot exceed 4000 tokens, and the length of the response that the model generates cannot exceed 4000 tokens either.

While 4000 tokens may seem like a lot, it is important to note that this limit can impact the complexity of the questions or statements that can be provided to the model. For example, it would not be possible to provide an entire software application in a single prompt due to the token limit.

To create a good prompt, it is important to provide clear and concise context for the model. This includes specifying the type of response that is desired, as well as any relevant background information or constraints. A good prompt should also be specific and avoid ambiguity to ensure that the model can provide an accurate response.

How to Design Effective Prompts

When interacting with ChatGPT, it's important to provide a clear and concise prompt that will help the model understand what you're looking for. In this chapter, we'll break down the anatomy of a good prompt and provide some tips for crafting effective prompts.

To start, let's take a look at a simple prompt that someone might use when working on a C# project:

"Create a C# class to represent a person with a name and an age."

This prompt is straightforward and provides a clear goal for the output. When we submit this prompt to ChatGPT, the model can generate a C# class that meets the requirements specified in the prompt.

Many would be satisfied with the output. But you would be leaving so much on the table. If you provide ChatGPT with a more colorful set of instructions you will have a much more detailed response.

Now, let's take a closer look at the different parts of a good prompt.

A prompt is made up of several different elements, each of which plays an important role in shaping the output produced by the language model. Here are the key parts of a prompt:

1. **The persona:** The persona element provides a specific context or identity for the language model to assume when generating output. By defining a persona, you can guide the language model towards generating output that is more relevant or tailored to a particular scenario or audience. The persona might include information about the user's background, expertise, preferences, or any other relevant characteristics.
2. **The task description:** This is the core of the prompt, and it should clearly describe the task you want the language model to perform. The task description should be specific and unambiguous, with all relevant details included. It should also be concise and easy to understand.
3. **The context:** The context provides additional information that the language model can use to generate more accurate and relevant output. The context might include things like sample inputs or outputs, related tasks or concepts, or any other information that will help the language model better understand what you're asking for.
4. **The constraints:** Constraints are any limitations or requirements that should be considered when generating output. For example, you might specify a maximum output length, a specific format or structure for the output, or any other conditions that need to be met.
5. **The examples:** Examples are sample inputs and outputs that help the language model learn how to perform the task correctly. You might provide one or more examples to demonstrate the desired behavior of the output.
6. **The prompts:** Prompts are additional phrases or sentences that can be included to guide the language model towards a specific type of output. Prompts might provide hints, suggestions, or constraints that can help the language model generate better output.

Using these guidelines, let's revise our original prompt to make it even clearer and more effective:

Persona: You are a skilled C# programmer with experience creating classes for various projects.

Task Description: Create a C# class to represent a person with a name and an age. Please include data validation and simple error/exception handling.

Context: You are working on a new project that needs to store information about people. You need to create a C# class that represents a person with a name and an age.

Constraints: The class should be named "Person". The name should be a string, and the age should be an integer.

Examples:

Input: Person p = new Person("Alice", 30); Output: p.Name == "Alice", p.Age == 30

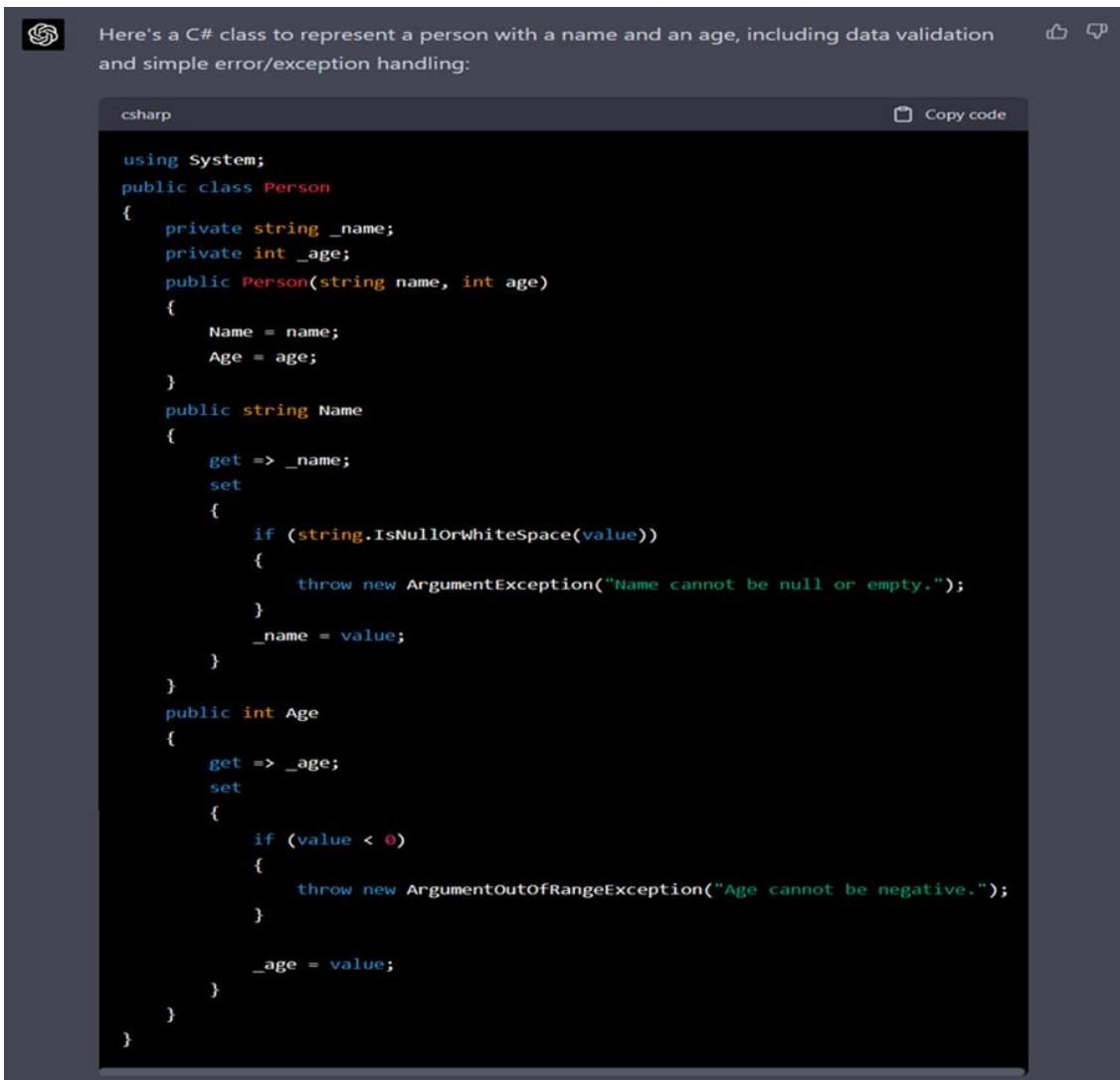
Input: Person p = new Person("Bob", 42); Output: p.Name == "Bob", p.Age == 42

Prompts: Write a constructor for the Person class that takes a name and an age as arguments.

This revised prompt is more specific about the requirements for the C# class and provides a clear name for the class. With this prompt, ChatGPT can generate a class that meets the exact requirements specified.

This is where you, as a skilled software engineer, should know how to prompt the AI engine to produce good quality code. Something a non-programmer would not know. I say this to emphasize how AI tools should not replace developers, but enhance good developers.

After executing the full prompt to generate a C# person class this is what I was able to generate:



Here's a C# class to represent a person with a name and an age, including data validation and simple error/exception handling:

```

csharp
using System;
public class Person
{
    private string _name;
    private int _age;
    public Person(string name, int age)
    {
        Name = name;
        Age = age;
    }
    public string Name
    {
        get => _name;
        set
        {
            if (string.IsNullOrWhiteSpace(value))
            {
                throw new ArgumentException("Name cannot be null or empty.");
            }
            _name = value;
        }
    }
    public int Age
    {
        get => _age;
        set
        {
            if (value < 0)
            {
                throw new ArgumentOutOfRangeException("Age cannot be negative.");
            }
            _age = value;
        }
    }
}

```

With this implementation, you can create `Person` objects as shown in the examples:

```
csharp
Person p1 = new Person("Alice", 30);
Console.WriteLine(p1.Name); // Output: Alice
Console.WriteLine(p1.Age); // Output: 30

Person p2 = new Person("Bob", 42);
Console.WriteLine(p2.Name); // Output: Bob
Console.WriteLine(p2.Age); // Output: 42
```

The class ensures that the name is not null or empty, and the age is not negative by throwing appropriate exceptions if invalid data is provided.

Fig 2.1

In conclusion, a good prompt is an essential part of working with ChatGPT. By following the guidelines outlined in this chapter, you can create effective prompts that will help the model generate accurate and relevant responses.

Another example: - Suppose I want a report with detailing the Progress, Challenges, and Future Plans in Developing Time Manipulation Technology

Prompt: You are a scientist working on a top-secret research project for a government agency. Your task is to develop a revolutionary technology that can manipulate time. Write a report detailing the progress you have made so far, the challenges you have encountered, and your plans for the future.

Chapter 3

Generating Code with ChatGPT

ChatGPT can be used to generate code for a wide range of tasks, including common programming paradigms and libraries. Here are some examples of how ChatGPT can help generate code to save time and increase efficiency in the development process:

Generating SQL Queries

ChatGPT can generate SQL queries based on specific prompts, such as “Generate a SQL query that selects all customers from the ‘customers’ table where the customer’s age is greater than 18.” ChatGPT can generate the necessary code to meet this criteria and save developers time in writing complex SQL queries.

Example SQL Prompts

- “Generate a SQL query that selects all products from the ‘products’ table where the product price is less than \$50. The query should return the product name, price, and quantity in stock.”
- “Write a SQL query that joins the ‘customers’ table with the ‘orders’ table on the customer ID, and selects the customer name and order date for each order. The query should only include orders that were placed in the year 2022.”
- “Generate a SQL query that calculates the total revenue from the ‘sales’ table for the month of January. The query should group the sales by product and include the product name and total revenue for each product.”
- “Write a SQL query that selects the top 10 customers from the ‘customers’ table with the highest total purchase amount. The query should join the ‘customers’ table with the ‘orders’ and ‘order_items’ tables, and calculate the total purchase amount for each customer.”
- “Generate a SQL query that creates a new table called ‘employee_details’ with columns for employee ID, name, salary, and department. The table should have a primary key on the employee ID column and a foreign key on the department column that references a ‘departments’ table.”

Let's use the first example, but apply the rules of a great prompt to make it even better.

Persona: An experienced SQL developer with knowledge of writing efficient and optimized SQL queries.

Target Audience: A junior developer who needs help in writing SQL queries to retrieve data from a relational database.

Define the task: Generate an SQL query to retrieve data from the "products" table where the product price is less than \$50. The query should return the product name, price, and quantity in stock.

Provide context: The database is used for an online store that sells various products. The "products" table contains all the products in the store with their corresponding price and stock quantity.

Clarify requirements: The query should only return products with a price less than \$50 and should include the product name, price, and quantity in stock.

Provide examples: The desired output of the query should be a table with columns for the product name, price, and quantity in stock. An example of the output could be:

| Product Name | Price | Quantity in Stock |
|--------------|-------|-------------------|
| Product 1 | \$10 | 50 |
| Product 2 | \$25 | 20 |
| Product 3 | \$45 | 30 |

Provide constraints: The query should be optimized for performance and should not include any unnecessary joins or subqueries. It should also follow best practices for SQL syntax and formatting.

Provide feedback: The query should be tested to ensure that it returns the correct results and does not have any errors or syntax issues.

I provided the previous example to demonstrate how a basic prompt can be improved by clearly specifying the expectations for the response. This can lead to a more effective and accurate result from the AI model.

Generating Individual Functions

ChatGPT can also be used to generate functions based on specific prompts or requirements. By providing a clear and concise prompt, ChatGPT can generate a function that meets your needs, saving developers time and effort in writing boilerplate code. Here are some examples of how ChatGPT can be used to generate functions:

- “Generate a function that accepts an array of numbers and returns the sum of all odd numbers in the array.”
- “Write a function that takes a string as input and returns a new string with all vowels removed.”
- “Generate a function that accepts an object and returns an array of all keys whose values are arrays.”
- “Write a function that takes an array of objects as input and returns a new array sorted by a specified property.”
- “Generate a function that takes two arrays as input and returns a new array with all elements that appear in both arrays.”

This is an example output for the last prompt merging 2 arrays in JavaScript:

Sure, here's a JavaScript function that takes two arrays as input and returns a new array with all elements that appear in both arrays:

```
function getCommonElements(arr1, arr2) {  
    let commonElements = [];  
  
    for(let i = 0; i < arr1.length; i++) {  
        for(let j = 0; j < arr2.length; j++) {  
            if(arr1[i] === arr2[j]) {  
                commonElements.push(arr1[i]);  
            }  
        }  
    }  
  
    return commonElements;  
}
```

The `getCommonElements` function takes two arrays `arr1` and `arr2` as parameters. It initializes an empty array called `commonElements` to hold the elements that are common to both arrays.

Then, it loops through each element of the first array (`arr1`) using a `for` loop, and for each element of `arr1`, it loops through each element of the second array (`arr2`) using another `for` loop. If the element of `arr1` matches the element of `arr2`, the function pushes that element to the `commonElements` array.

Finally, the function returns the commonElements array which contains all the elements that appear in both arrays.

You can call this function with two arrays like this:

```
const array1 = [1, 2, 3, 4, 5];
const array2 = [4, 5, 6, 7, 8];

const commonElements = getCommonElements(array1, array2);
console.log(commonElements); // Output: [4, 5]
```

In this example, the commonElements array will contain the elements that are common to both array1 and array2, which are [4, 5].

Creating REST APIs

ChatGPT can generate code to create REST APIs based on specific prompts, such as “Generate a REST API that allows users to create, read, update, and delete blog posts.” ChatGPT can generate the necessary code to set up a REST API that meets this criteria, saving developers time in setting up API endpoints and handling HTTP requests.

- “Generate code to create a REST API that allows users to create, read, update, and delete blog posts. The API should use JSON as the data format and should authenticate users using JWT tokens.”
- “Write code to create a REST API that allows users to upload and download files. The API should support multiple file types and should limit file sizes to 10MB.”
- “Generate code to create a REST API that integrates with a third-party payment gateway. The API should allow users to create and manage payments, and should handle both one-time and recurring payments.”
- “Write code to create a REST API that allows users to search and book flights. The API should integrate with a flight booking service and should provide real-time availability and pricing information.”
- “Generate code to create a REST API that allows users to schedule appointments with doctors. The API should integrate with a calendar service and should provide availability and scheduling options based on the doctor’s schedule.”

Creating Unit and Integration Tests

Unit and integration tests are critical for ensuring that software functions as intended and meets the requirements of its users. ChatGPT can help developers create unit and integration tests by generating test cases based on specific prompts. In this section, we’ll provide some examples of how ChatGPT can help developers create unit and integration tests.

- “Write a unit test that verifies a login function returns a token for a valid user and throws an error for an invalid user.”

- “Create an integration test that validates file upload functionality. The test should simulate a user uploading a file and ensure that the file is saved to the correct location with the correct metadata.”
- “Generate test cases to ensure that an email sending function sends emails to the correct recipients, with the correct subject line, and includes the correct message body.”
- “Create a unit test that checks the behavior of a function that sorts an array of integers in ascending order. The test should verify that the function returns the correct output for arrays of varying sizes and with varying integer values.”
- “Write an integration test to validate a payment processing function. The test should simulate a user initiating a payment, and ensure that the payment is processed correctly and the user’s balance is updated accordingly.”

Generating Test Cases

ChatGPT can generate test cases for a wide range of tasks, including testing individual functions or modules, as well as testing interactions between components. Developers can provide prompts that specify the input parameters and expected output for a particular test case, and ChatGPT can generate the necessary code to run the test case. This can save developers time and increase efficiency in the testing process.

- “Generate test cases for a function that calculates the factorial of a given number. The test cases should include edge cases for negative and zero values, as well as normal test cases for positive integers.”
- “Write test cases to ensure that a login function for a web application is secure and properly validates user credentials. The test cases should include both correct and incorrect user inputs, and should cover edge cases such as SQL injection and cross-site scripting attacks.”
- “Generate test cases for a REST API that allows users to search for products based on various criteria, such as product name, category, and price range. The test cases should cover all possible combinations of search parameters, and should ensure that the API returns the expected results.”
- “Write test cases to ensure that a billing system for a subscription-based service is properly calculating subscription fees and handling cancellations. The test cases should cover different subscription types and durations, as well as scenarios where users cancel their subscriptions before the end of their billing period.”
- “Generate test cases for a function that sorts an array of integers in ascending order. The test cases should include arrays of varying lengths, arrays with duplicate values, and arrays with negative numbers.”

Test Automations

ChatGPT can be used to automate the execution of test cases, by generating code for test runners and test frameworks. Developers can provide prompts that specify the testing environment and expected outcomes, and ChatGPT can generate the necessary code to automate the testing process. This can save developers time and increase efficiency in the testing process.

- “Write code to automate the execution of unit tests for a web application. The tests should run in a headless browser and should validate the functionality of all user interactions.”
- “Generate code to automate the testing of a REST API. The tests should ensure that all endpoints return the expected status codes and data formats, and should include both positive and negative test cases.”
- “Create code to automate the testing of a machine learning model. The tests should evaluate the model’s accuracy and performance on a variety of input data, and should include both unit tests and integration tests.”
- “Write code to automate the testing of a mobile application. The tests should run on both iOS and Android devices and should validate the functionality of all user interactions and device sensors.”
- “Generate code to automate the testing of a database schema. The tests should ensure that all tables and columns are present and have the correct data types and constraints, and should validate the functionality of all stored procedures and triggers.”

Generating Code Documentation

Code documentation is essential for ensuring that software can be easily understood and maintained over time. ChatGPT can help developers generate code documentation by generating descriptions and explanations of code functions, methods, and classes. In this section, we’ll provide some examples of how ChatGPT can help developers generate code documentation.

This is something I have found especially helpful. Personally I find writing documentation to be tedious. I also want to write code JSDoc can consume to produce documentation to help other developers. This is where an AI tool can really help.

- “Generate JSDoc for a JavaScript function that retrieves data from an external API. The JSDoc should include information about the function’s input parameters, expected return values, and any error handling.”
- “Create documentation for a C# class library that handles file I/O operations. The documentation should include descriptions of each class, method, and property, as well as examples of usage.”
- “Generate API documentation for a REST API that allows users to manage their profiles. The documentation should include information about each endpoint, including input parameters, expected return values, and error handling.”

- “Create documentation for a Python module that implements machine learning algorithms. The documentation should include explanations of each algorithm, as well as instructions for training and evaluating models.”
- “Generate code comments for a Java program that implements a database management system. The comments should describe the purpose and functionality of each class and method, as well as any relevant design decisions.”

These prompts provide specific instructions for generating documentation for different programming languages and contexts, while still allowing ChatGPT to generate the necessary descriptions and explanations.

However, these prompts are not complete. Assuming I have given ChatGPT its persona and other aspects of a good prompt I also give it a copy of the function or object I need it to document. It looks something like this:

“Please include any comments to explain the following function. I would like JSDoc comments included.

```
function mergeScripts(parentScripts = [], childScripts = []) {  
    const uniqueChildScripts = [...new Set(childScripts)];  
  
    return [...parentScripts, ...uniqueChildScripts];  
}
```

This is the ChatGPT output:

 Sure! Here's an explanation of the `mergeScripts()` function with JSDoc comments included:

```
/**  
 * Merges an array of parent script sources with an array of child script sources, removing  
 * any duplicates.  
 * @param {string[]} parentScripts - An array of parent script sources.  
 * @param {string[]} childScripts - An array of child script sources.  
 * @returns {string[]} - A merged array of script sources with duplicates removed.  
 */  
function mergeScripts(parentScripts = [], childScripts = []) {  
    // Remove any duplicate script sources in childScripts  
    const uniqueChildScripts = [...new Set(childScripts)];  
    // Return merged array of script sources  
    return [...parentScripts, ...uniqueChildScripts];  
}
```

This function takes in two arrays of script sources, `parentScripts` and `childScripts`, and returns a merged array of script sources with duplicates removed.

The `parentScripts` array contains the script sources that are already present on the page, while the `childScripts` array contains additional script sources that need to be added. The `parentScripts` array is passed as the first argument to the function, while `childScripts` is passed as the second argument.

The function first removes any duplicate script sources from the `childScripts` array using the `Set` object and the spread operator. This creates a new array `uniqueChildScripts` with only unique script sources.

Then, the function uses the spread operator again to combine the `parentScripts` and `uniqueChildScripts` arrays into a single merged array. This merged array is then returned as the output of the function.

The JSDoc comments for this function provide useful information about the parameters and return value, helping developers to understand how to use the function correctly.

Fig 3.1

I would also point out that when I ran this through ChatGPT there were several more paragraphs where the AI model explained in detail how this function works beyond the comments. This too can be very helpful for you as a developer.

Creating User Interfaces

ChatGPT can generate code to create user interfaces based on specific prompts, such as “Generate an HTML and CSS template for a login page with email and password fields.” ChatGPT can generate the necessary code to set up a user interface that meets this criteria, saving developers time in writing complex HTML and CSS code.

Prompt Examples:

This example should generate a basic list of products in a shopping cart.

“You are an expert at HTML, CSS, JavaScript and designing user interfaces.

Generate code to create a web-based user interface that allows users to search and filter a database of products. The interface should display product images, descriptions, and prices, and should allow users to add products to a shopping cart.

The code should be fully responded to with a mobile first approach. Please be mindful of small screens and allow for adequate space to touch key interface elements.”

- “Write code to create a mobile app user interface that allows users to view and edit their personal information. The interface should include forms for entering and updating data, as well as buttons for saving and canceling changes.”
- “Generate code to create a desktop application user interface that allows users to view and manipulate a 3D model. The interface should include tools for rotating, zooming, and panning the model, as well as options for changing its appearance and texture.”
- “Write code to create a voice user interface that allows users to control a smart home device. The interface should respond to voice commands, and should allow users to turn on and off lights, adjust temperature, and perform other actions.”
- “Generate code to create a chatbot user interface that allows users to order food from a restaurant. The interface should respond to natural language inputs, and should be able to process orders, display menus, and handle payment processing.”

Updating Legacy Code

ChatGPT can be a useful tool for suggesting updates to older codebases that use outdated or suboptimal patterns.

If you’re working with legacy code, which is a common scenario for many developers, you may find yourself facing the challenge of updating old code to newer standards and conventions. This

task can be time-consuming and resource-intensive, which is why ChatGPT can be a valuable asset in reducing the time required for updating code by up to 90% or more.

By following some simple steps, you can leverage ChatGPT's capabilities to modernize legacy code and improve its maintainability and performance.

Identify the code to be updated: Take a look at the codebase you want to update and identify the parts that need to be improved. This could be outdated syntax, poor performance, or other issues.

Write a prompt for ChatGPT: Craft a prompt that describes the specific problem you want to solve, and provide some context around the code. This will help ChatGPT understand the problem you're trying to solve and provide more relevant suggestions.

Provide the legacy code as input to ChatGPT: Once you have a prompt ready, provide the legacy code as input to ChatGPT. You can do this by pasting the code into a text editor, or by providing a link to a GitHub repository.

Review the suggestions: ChatGPT will provide suggestions based on the prompt and the input code. Review these suggestions and decide which ones are relevant to your use case.

Apply the updates: Once you've reviewed the suggestions, apply the updates to your codebase. Make sure to test the code thoroughly to ensure that it still works as expected.

Here's an example prompt for updating a legacy function that uses `then.catch.finally` chains to use modern `async/await` syntax:

"Update this function to use modern `async/await` syntax:

```
function legacyFunction() {
  return fetch('https://example.com/data')
    .then(response => {
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      return response.json();
    })
    .then(data => {
      console.log(data);
    })
    .catch(error => {
      console.error('Error:', error);
    });
}
```

In response to this prompt, ChatGPT might suggest an updated version of the function that uses `async/await` syntax instead:

```
async function updatedFunction() {  
    try {  
        const response = await fetch('https://example.com/data');  
        if (!response.ok) {  
            throw new Error('Network response was not ok');  
        }  
        const data = await response.json();  
    } catch (error) {  
        console.error('Error:', error);  
    }  
}
```

By following this process, you can use ChatGPT to update legacy code and bring it up to modern standards and conventions.

Decoding Cryptic Error Messages

As developers, we often encounter cryptic error messages that are difficult to understand and can slow down the debugging process. ChatGPT can be a helpful tool for troubleshooting error messages and finding solutions to common problems. By providing ChatGPT with an error message or a description of the problem, you can get suggestions for possible solutions or next steps to take.

Here's an example of how you might use ChatGPT to troubleshoot a cryptic error message:

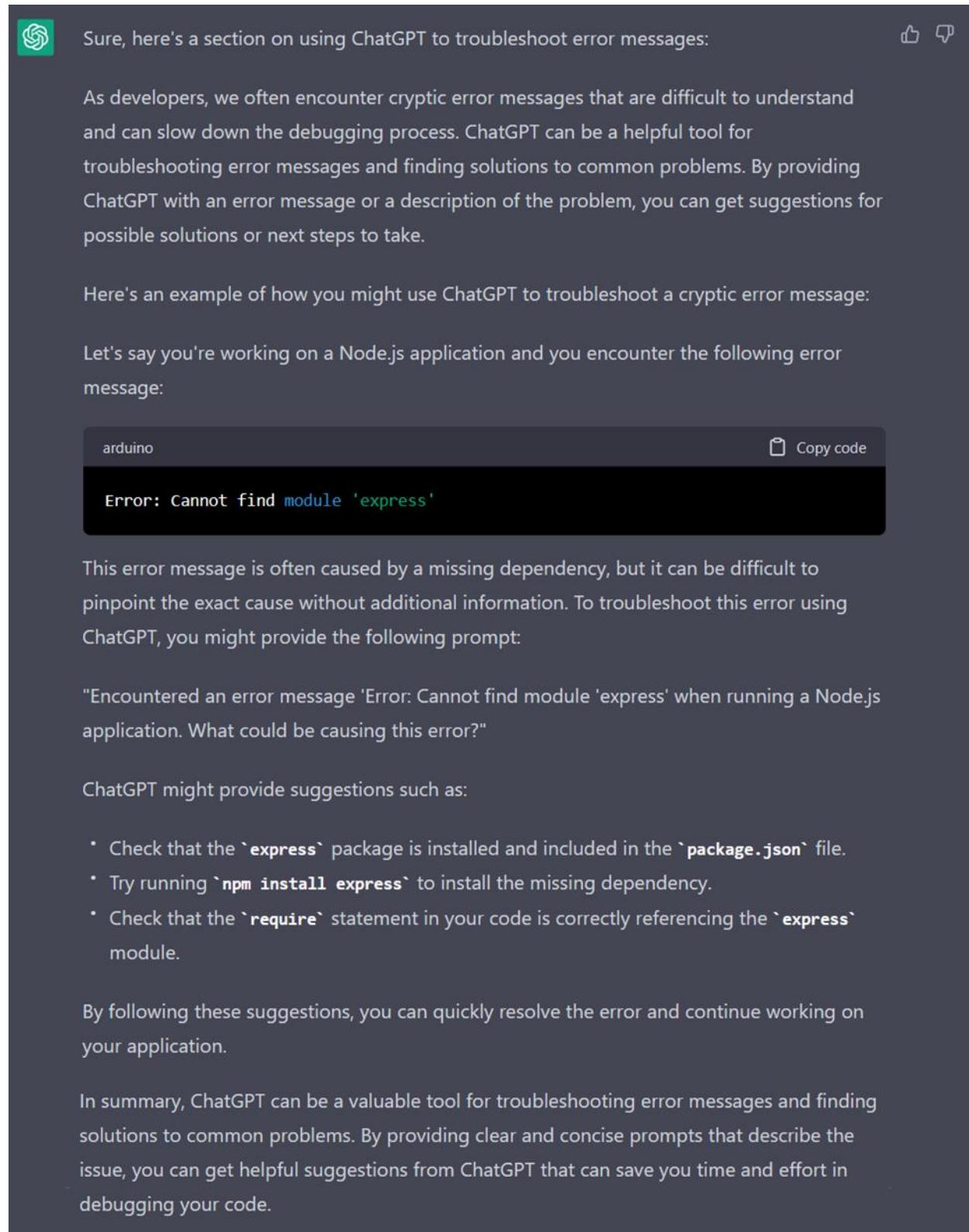
Let's say you're working on a Node.js application and you encounter the following error message:

Error: Cannot find module 'express'

This error message is often caused by a missing dependency, but it can be difficult to pinpoint the exact cause without additional information. To troubleshoot this error using ChatGPT, you might provide the following prompt:

"Encountered an error message 'Error: Cannot find module 'express' when running a Node.js application. What could be causing this error?"

ChatGPT might provide suggestions such as:



Sure, here's a section on using ChatGPT to troubleshoot error messages:

As developers, we often encounter cryptic error messages that are difficult to understand and can slow down the debugging process. ChatGPT can be a helpful tool for troubleshooting error messages and finding solutions to common problems. By providing ChatGPT with an error message or a description of the problem, you can get suggestions for possible solutions or next steps to take.

Here's an example of how you might use ChatGPT to troubleshoot a cryptic error message:

Let's say you're working on a Node.js application and you encounter the following error message:

```
arduino
Error: Cannot find module 'express'
```

This error message is often caused by a missing dependency, but it can be difficult to pinpoint the exact cause without additional information. To troubleshoot this error using ChatGPT, you might provide the following prompt:

"Encountered an error message 'Error: Cannot find module 'express' when running a Node.js application. What could be causing this error?"

ChatGPT might provide suggestions such as:

- Check that the `express` package is installed and included in the `package.json` file.
- Try running `npm install express` to install the missing dependency.
- Check that the `require` statement in your code is correctly referencing the `express` module.

By following these suggestions, you can quickly resolve the error and continue working on your application.

In summary, ChatGPT can be a valuable tool for troubleshooting error messages and finding solutions to common problems. By providing clear and concise prompts that describe the issue, you can get helpful suggestions from ChatGPT that can save you time and effort in debugging your code.

Fig 3.2

By following these suggestions, you can quickly resolve the error and continue working on your application.

In summary, ChatGPT can be a valuable tool for troubleshooting error messages and finding solutions to common problems. By providing clear and concise prompts that describe the issue, you can get helpful suggestions from ChatGPT that can save you time and effort in debugging your code.

Chapter 4

Jobs and Careers

I would be leaving without something ChatGPT can do to help you enhance your career, and that is find a job or new projects!

I am going to start with creating a job description. This prompt assumes you have already established what ChatGPT is and the target audience. In this case you are an IT department project manager and are trying to find candidates for a common .NET project.

"Can you write a job description for a C# developer with 5 years of experience in .NET for a .NET Core application? The application must know Entity Framework, error, and transaction logging. The position will be responsible for building REST API with full business and data layers using a SQL Server database."

 Job Title: C# Developer  

We are seeking a highly skilled C# developer with at least 5 years of experience in .NET development for a .NET Core application. The ideal candidate will have extensive knowledge of Entity Framework, error, and transaction logging. The developer will be responsible for designing and implementing REST API with full business and data layers using a SQL Server database.

Responsibilities:

- Design and develop REST API with full business and data layers
- Implement error and transaction logging for the application
- Collaborate with the team to design and implement new features and improvements
- Write clean, well-documented, and maintainable code
- Ensure the application meets performance and security requirements
- Troubleshoot and debug application issues
- Participate in code review and continuous improvement processes

Requirements:

- Bachelor's degree in Computer Science, Engineering or a related field
- At least 5 years of experience in C# development and .NET Core
- Proficiency in Entity Framework and SQL Server database
- Experience in error and transaction logging
- Strong understanding of RESTful API principles and implementation
- Experience in building scalable, maintainable and high-performance applications

Fig 4.1

You could also use ChatGPT to help you understand job descriptions better. Here I am asking ChatGPT to summary and give a little advice about a possible job:

"Can you review this job description and give me a summary of what skills I need to know. Also, can you advise me what the project might actually entail?

Job Description

What you can expect from the job:

Good knowledge in testing web & mobile applications

Analyze user stories, use cases, requirements and prepare test scenarios and test cases.

Execute all levels of testing (System, Integration, Compatibility, Security, Performance, API and Regression).

Analyze test results, detect issues, and track root causes.

Ability to convert the manual test cases to automated script and build regression test suite and execute.

Integrate and execute automated tests in CI/CD pipeline

Assist Test Analyst for preparing test strategies and implementing the test strategies.

Good knowledge in using git or any version control system.

What will help you do the job well?

Job Description: 4-8 years of experience in Testing Web, Mobile and PWA Applications using manual and automation testing. Minimum 2-3 years in test automation.

A track record of performing end-to-end testing activities through the test life cycle in web and mobile applications with good experience with GUI Testing, cross browser testing, regression testing and integration testing.

Good communication, presentation, analytical, and problem-solving skills.

Hands on experience in at least one defect management and test management tool.

Experience in non-functional testing using JMeter, OWASP ZAP.

Good knowledge and experience in test automation using data driven test automation frameworks using POM with Selenium WebDriver, Appium, TestNG, Extent report, Maven, and scripting language Java.

Experience in testing banking domain applications.

Experience working in an Agile, Scrum development process.

Good Knowledge in CI Tool Jenkins with Selenium WebDriver and Appium.

We would love it if you have:

Certification in ISTQB or CSTE.

Good knowledge for using Clouds for Test Automation, Manual Testing.

Knowledge in testing Banking domain applications (Finacle, Temenos)

Senior Test Engineer - Automation

You can succeed if:

You enjoy hands on work with end-to-end exposure

You are intrinsically motivated

You have personal drivers other than money

You are patient

You are real and prefer less of sophistication

You want to grow with the team"



This job description is seeking a Senior Test Engineer with 4-8 years of experience in testing web, mobile, and PWA applications using both manual and automation testing, with a minimum of 2-3 years in test automation. The candidate should have experience in end-to-end testing activities throughout the test life cycle, including GUI testing, cross-browser testing, regression testing, and integration testing. They should have good communication, presentation, analytical, and problem-solving skills, as well as hands-on experience in at least one defect management and test management tool. Additionally, the candidate should have experience in non-functional testing using JMeter, OWASP ZAP, and test automation using data-driven test automation frameworks using POM with Selenium WebDriver, Appium, TestNG, Extent report, Maven, and Java scripting language. It is also desired that the candidate has experience in testing banking domain applications and working in an Agile, Scrum development process.

The job entails testing web, mobile, and PWA applications using manual and automation testing, with an emphasis on end-to-end testing activities. The Senior Test Engineer will be responsible for analyzing user stories, use cases, requirements, and preparing test scenarios and test cases, executing all levels of testing, analyzing test results, detecting issues, and tracking root causes. They will also be responsible for converting manual test cases to automated scripts, building regression test suites, and integrating and executing automated tests in CI/CD pipelines. The Senior Test Engineer will assist the Test Analyst for preparing test strategies and implementing the test strategies, using git or any version control system, and utilizing non-functional testing using JMeter, OWASP ZAP.

Fig 4.2

ChatGPT can be a valuable tool for developers looking to improve their job prospects. By analyzing job descriptions and industry trends, ChatGPT can help developers identify the skills and knowledge they need to acquire or improve in order to land their dream job.

One way to use ChatGPT for this purpose is to give it access to your resume or CV and ask for feedback on what skills or experiences may be lacking. ChatGPT can provide suggestions for filling in these gaps and improving your qualifications.

In addition to identifying areas for improvement, ChatGPT can also help developers optimize their resumes for the job market. By analyzing job postings and industry buzzwords, ChatGPT can suggest changes to your resume that can make it more attractive to potential employers. It can also help clean up grammar and readability issues, making your resume more polished and professional.

Example:- you may use to prompt ChatGPT to help you see where the market demand might be:

"Hi ChatGPT, I'm a software developer with 5 years of experience in .NET and C#. I'm currently looking for a new job and would like your help in optimizing my resume for the job market. Can you analyze job postings in the software development industry and suggest changes to my resume that can make it more attractive to potential employers? Please also suggest any relevant industry buzzwords that I should include in my resume. Finally, can you help me clean up any grammar and readability issues in my resume to make it more polished and professional? Thank you!"

Sure, I'd be happy to help you optimize your resume for the software development job market. Based on my analysis of job postings in the software development industry, here are some suggestions and relevant buzzwords you should consider including in your resume.

1. Technical Skills:
 - Make sure to highlight your expertise in .NET and C# as these are your primary skills.
 - Add relevant frameworks and libraries you've worked with, such as ASP.NET, .NET Core, Entity Framework, and LINQ.
 - Mention your experience with front-end technologies like HTML, CSS, JavaScript, and popular libraries or frameworks (e.g., Angular, React, or Vue.js) if you have any.
 - Include any experience with cloud platforms like Azure, AWS, or Google Cloud, as well as containerization technologies like Docker and Kubernetes.
 - Don't forget to list any experience with database systems, such as SQL Server, MySQL, PostgreSQL, or NoSQL databases like MongoDB or Cassandra.

Buzzwords to include: .NET Core, ASP.NET, Entity Framework, LINQ, Azure, AWS, Google Cloud, Docker, Kubernetes, SQL Server, MySQL, PostgreSQL, MongoDB, Cassandra, HTML, CSS, JavaScript, Angular, React, Vue.js.

2. Soft Skills:
 - Emphasize your communication and collaboration skills, as teamwork is crucial in software development projects.
 - Showcase your problem-solving abilities and critical thinking skills.
 - Mention any experience working in Agile or Scrum environments, as these methodologies are widely used in the industry.

Buzzwords to include: Communication, Collaboration, Problem-solving, Critical thinking, Agile, Scrum.

3. Projects and Achievements:
 - Highlight any significant projects you've worked on, specifying the technologies used, your role, and the impact of the project.
 - Include any professional certifications, awards, or recognitions you have received in the software development field.

Now, I will need to see your resume to provide specific feedback on grammar and readability issues. Please feel free to share your resume text or link, and I'll be happy to help you make it more polished and professional.

Fig 4.3

It's important to note that while ChatGPT can be a helpful tool in the job search process, it should never be used to misrepresent your skills or qualifications. Instead, use ChatGPT to identify areas for improvement and to polish your resume in a way that accurately reflects your abilities and experience.

Conclusion

ChatGPT is a powerful tool that can assist software developers in a wide range of tasks related to natural language processing. By leveraging the capabilities of ChatGPT for generating code, writing documentation, and answering common questions, developers can save time and increase efficiency in their workflows, and ultimately produce higher quality software.

I have found the integration of AI tools to my development workflow to be invaluable, but it's important to exercise caution and not blindly trust the tool. Always double check any code, comments, or responses before incorporating them into your applications, and perform actual tests to ensure the code functions as intended.

With this in mind, I encourage developers to integrate ChatGPT and other AI tools into their developer toolbox and experiment with different prompts and settings to maximize the benefits they offer.



Where we use ChatGPT?

Generate Text

Prompt-
"Write a blog
post about AI
Latest trend."

Translate Language

Prompt -
"Translate this
content
English to
Spanish."

Write a different kind of creative content

Answer Question in an informative way

Prompt-
"Write a article
on HTML tags."

Prompt-
"How do I
change the
format of
image from
JPG to PNG."

Universal Command

Prompt- "Write a length [type] on [topic] on topic in style of [style]."

List Keywords

Prompt - "Give me a list keywords associated with [subject]."

Write and Debug Code

Prompt - "Please convert this code from Javascript to Python [code here]."

Modification Prompts

Prompt - "Ask it to translate, lengthen, paraphrase, etc translate text [text here]."

Rephrase This Text

Prompt - "Rephrase this text in the style [style] : [insert text]."

Information Extraction

Prompt - "What was the year when C# Corner launched and who is the founder of C# Corner."

Prompts for Developers

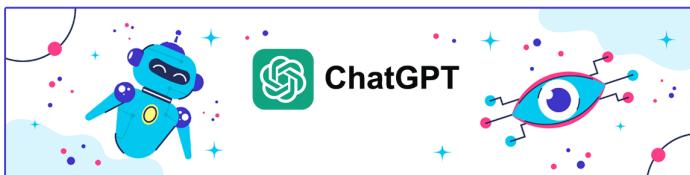
Prompt -
"Develop an architecture and code for [description] website with javascript."
"Help me to find mistakes in the following code [paste you code]."
"I want to implement a sticky header on my Website can you provide me an example using CSS JavaScript?."
"Please continue writing this code for JavaScript post code below."

Prompts for Marketers

Prompt -
"Can you provide me with some ideas for blog post about [topic]."
"Write a product description for my [product or services or company]."
"Suggest inexpensive ways I can promote my company without using social media."
"How can I obtain high-quality backlinks to raise the SEO of [website name]."

Chained Prompting

Prompt -
"Write an article about ChatGPT."
"First give me the outline, which consists of a headline, a teaser, and several subheadings."
[ChatGPT Response]
"Now write 5 different subheadings."
[ChatGPT Response]
"Add 5 keywords for each subheadings."
[ChatGPT Response]





Extensions

ChatGPT + Internet

TweetGPT

Find Prompts

WebChatGPT

ChatGPT as Search

WebChatGPT

ChatGPT + Email

ChatGPT + Writer

ChatGPT + Twitter

ChatOnAI

ChatGPT with Siri

Promptheus

- ChatGPT Prompt to help you analyze the Job Description

Prompt - "Highlight the 3 most important responsibilities in this job description:
[insert job description]"

Plugins

Step 1 - API Key connection

Step 2 - Provide Requirements

Step 3 - Replit Acc creation

Step 4 - Ask for main.py

Step 5 - Manifest file creation

Step 6 - Enter OpenAPI Info

Step 7 - Enter API definition

Step 8 - OpenAPI to Replit

Step 9 - Upload Plugin

Step 10 - Get started with your Plugin!

Sample of Categorizing ChatGPT Queries -

- 1) Ask for explanations: If you need detailed explanations or want ChatGPT to provide step-by-step instructions, ask for them explicitly. For example:

"Can you break down the concept of conversational AI for me?"

"Could you give me an example of a chatbot conversation?"

- 2) Seek opinions or advice: If you're looking for an opinion or advice, explicitly mention it in your prompt. For example:

"What's your opinion on the latest trends in chatbot design?"

"Do you have any tips for improving chatbot engagement?"

Note:

Define the role of ChatGPT

Prompt - "Acts as an English teacher"
"Acts as a Coach"
"Acts as Adviser"
"Acts as a Javascript Console"

Learn from ChatGPT

Prompt - "Clearly explain"
"Uniquely explain"
"Explain in detail"
"Explain with Example"

Preparation for Job Interviews

- ChatGPT Prompt for what companies are looking to interview

Prompt - "What specific skills and experience should I highlight in an interview for [job title] in [specific industry]?"

- ChatGPT Prompt to check what questions can be asked

Prompt - "What are the 10 most common questions companies ask when interviewing [role] [job title]"

- ChatGPT Prompt to help you with your very own cover letter

Prompt - "Please write a personalized cover letter for this [job title] at [company]."

Here is the job description:

[paste job description].

And here is my resume:

[paste resume]."

About C# Corner

Thank you for becoming a member of the C# Corner community. C# Corner, a highly reputed online community founded in year 2000 by Mahesh Chand, a programmer from his apartment. It is a platform where software developers could come together to exchange knowledge and share open-source projects. Today, C# Corner proudly serves nearly 30 million developers annually.



professional growth. To facilitate this, we manage and host in-person chapter events and conferences, which provide opportunities for members to connect and share knowledge. We also offer live streaming of various technology-focused shows, webinars, and virtual conferences to further enhance networking opportunities.

C# Corner MVP Award

The MVP (Most Valuable Professional) award is a highly esteemed annual accolade presented to top influencers and contributors in the community. Typically, MVPs are recognized as thought leaders, speakers, trainers, mentors, C-level executives, and/or leaders of their respective communities.



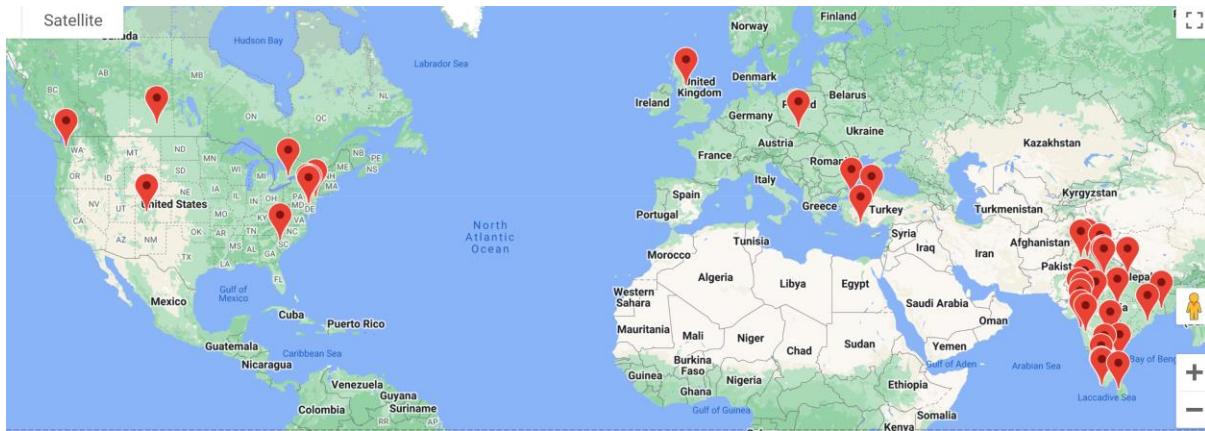
Learn, Share, Network, and Grow!

C# Corner empowers its members to advance in their professional careers by providing a range of online learning resources such as articles, tutorials, videos, and forums. Additionally, C# Corner offers a suite of tools and services for professionals, including resume writing, job challenges, a job board, training, and certifications.

At C# Corner, we recognize that networking is a key factor in

C# Corner Chapters

C# Corner has established local chapters in numerous locations across the globe, with a majority located in India and the US. These chapters serve as hosts for monthly user group meetups, which focus on the latest and most innovative technologies in software development.



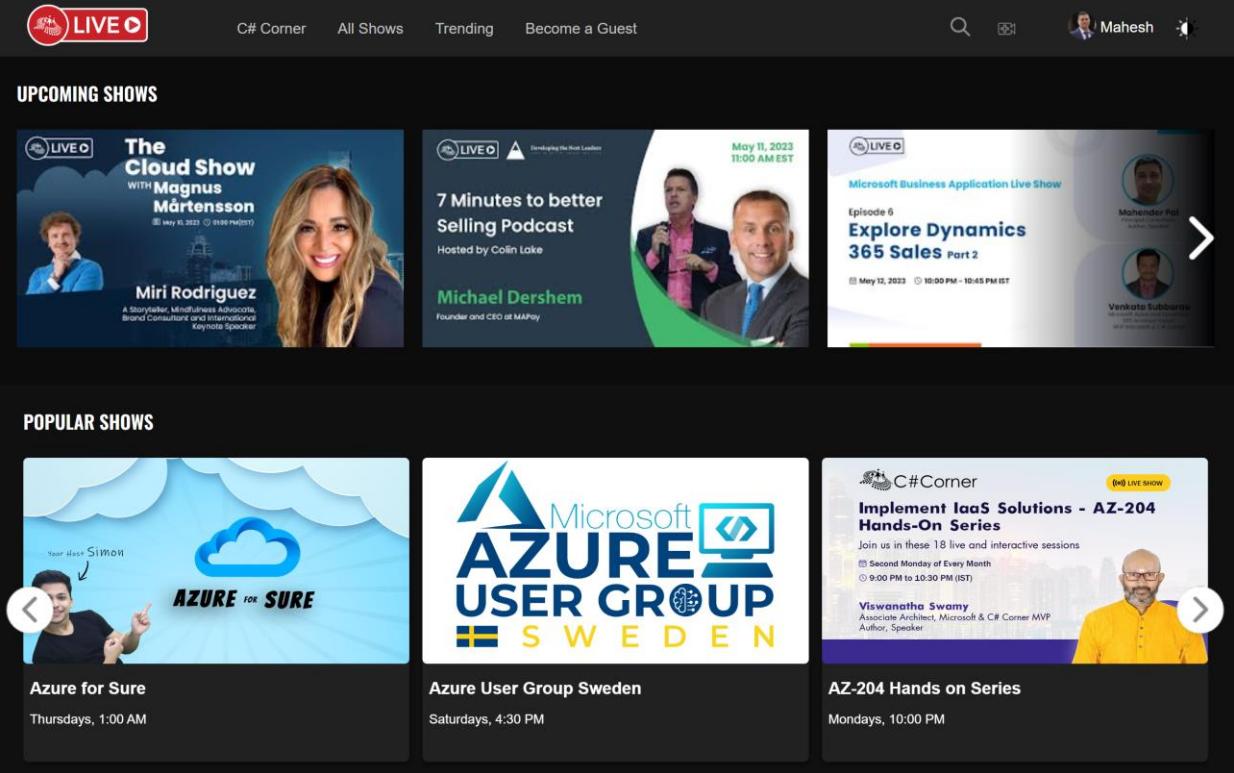
The Annual Conference

C# Corner is proud to organize some of the largest developer conferences worldwide. In 2021 alone, we successfully hosted 20 virtual conferences, which collectively reached an impressive global audience of 550,000 attendees.



CSharp.Live

CSharp Live streams live shows hosted by some of the top experts in the industry.



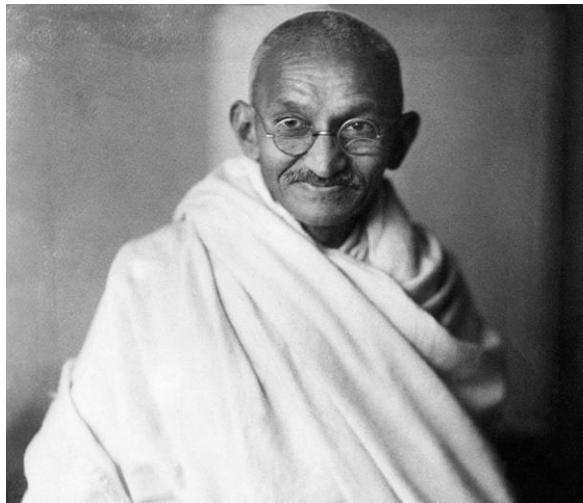
The screenshot displays the CSharp.Live website interface. At the top, there's a navigation bar with links for "C# Corner", "All Shows", "Trending", "Become a Guest", a search icon, and a profile picture for "Mahesh". Below the navigation is a section titled "UPCOMING SHOWS" featuring three live stream cards:

- The Cloud Show with Magnus Mårtensson** (May 11, 2023, 11:00 AM EST) Hosted by Miri Rodriguez and Michael Dershem.
- 7 Minutes to better Selling Podcast** (May 11, 2023, 11:00 AM EST) Hosted by Collin Lake.
- Microsoft Business Application Live Show** (May 12, 2023, 10:00 PM - 10:45 PM IST) Episode 6: Explore Dynamics 365 Sales Part 2. Hosted by Venkatesh Subbarao and Mohender Pal.

Below the upcoming shows is a section titled "POPULAR SHOWS" with three cards:

- Azure for Sure** (Thursdays, 1:00 AM) Hosted by Simon.
- Azure User Group Sweden** (Saturdays, 4:30 PM)
- Implement IaaS Solutions - AZ-204 Hands-On Series** (Mondays, 10:00 PM) Hosted by Viswanatha Swamy.

You can help!



“Live as if you were to die tomorrow. Learn as if you were to live forever.”

Mahatma Gandhi

Spread the word. Become a part of the best community in the world. Please visit us and follow us:

- Website: <https://www.c-sharpcorner.com/>
- LinkedIn: <https://www.linkedin.com/company/csharpcorner/>
- Twitter: <https://twitter.com/CsharpCorner>
- YouTube: <https://www.youtube.com/@CsharpCorner1>

More eBooks from us

