Lab: Software Documentation Using Generative AI

Estimated time needed: 30 minutes

As a software developer, you may be very comfortable with coding. However, coding skills alone do not suffice in the real world. The code you create is more often than not used and maintained by many people. To ensure that your code is readable and understandable, you need to include software documentation. While this activity is of pivotal importance, it is time-consuming. You can avail the help of Generative AI to generate the documentation.

Please remember the prompts that you feed to Generative AI are like a conversation with a subject matter expert and the consecutive prompts depend on the previous prompts used and the response received. Change the wording if required to get a specific desired result. The example showcases one possible chat conversation to attain the objective.

Learning objectives

At the end of the lab, you will be able to:

1. Use Generative AI to generate documentation for the code you have written.
2. Learn how documentation is written for JavaScript.

Prerequisites

It would be good if you know how to code in JavaScript, as this lab is based on creating documentation for JavaScript code.

Set up chat title and prompt instruction

When the chat session opens, a new chat conversation begins. Give the chat conversation an appropriate title. This will help you revisit the chat conversation. It is a good practice to segregate the conversations topically as this will help you continue the conversation at a later point.

Also provide a prompt instruction which is specific to the conversation in this particular lab. Let's get started with the task of creating software documentation for the JavaScript code.

> Please note Generative AI is an evolving field. As you attempt the labs, your experience and output might be different than what is seen here.

Tool capability to translate

You are already comfortable coding in JavaScript. You have created the code in JavaScript. You now need to include documentation in the code. You decide to optimally utilize your time by using Gen AI to write the documentation.

Firstly, you need to determine if the Gen AI tool has the capability to do the required software documentation for JavaScript. In the prompt type:

```
I have code written in JavaScript. Can you help me add documentation to the code?
```

If the GenAI model has the ability to add software documentation to the code, you will get an affirmative response in line with what is shown in the image below. JavaScript is one the most commonly used languages and this task of creating software documentation for JavaScript code will be doable. But if there are other newer or scarcely used languages, you will need to use your judgment to perceive the confidence in the response. Based on these responses you can decide on continuing to use the tool either entirely or as a support tool. Iterative prompts can be used to ensure the correctness of the responses.

▶ Click here to view the sample response generated

Provide the code to Gen AI

Paste the following request along with the code in the prompt and send it. This process will take a few seconds as it needs to process the code and return it with the documentation.

```
Add documentation to the following code.
const express = require('express');
const axios = require('axios');
const app = express();
const port = 3000;
app.get('/weather/:city', async (req, res) => {
  try {
    const city = req.params.city;
    const weatherData = await getWeatherData(city);
    res.json(weatherData);
  } catch (error) {
    res.status(500).json({ error: 'Failed to fetch weather data' });
  }
});
async function getWeatherData(city) {
  const apiKey = 'YOUR_API_KEY';
  const apiUrl = `https://api.weatherapi.com/v1/current.json?key=${apiKey}&q=${city}`;
  const response = await axios.get(apiUrl);
  const weatherData = response.data;
  return {
    city: weatherData.location.name,
    temperature: weatherData.current.temp_c,
    condition: weatherData.current.condition.text,
  };
}
app.listen(port, () => {
  console.log(`Server is running on [http://localhost:$%7bport%7d%60]http://localhost:${port}`);
});
```

It will produce an output in JavaScript code with documentation similar to what is shown below.

▶ Click here to view the sample output generated

This is assuming that the JavaScript code has been typed properly. If there are minor coding mistakes, Gen AI has the abilitiy to correct them and give the right output.

The documentation may not cover all the aspects you intended to cover. You may need to follow up with specific prompts.

Variable names casing format

Particular kinds of casing formats are recommended as good coding practice. Camel case has been used in the code here. In camel case, the variable name starts with lowercase and every consecutive word starts with uppercase. For instance,

`studentNumber`. Type the following prompt to find which case is better?

```
Let me know which case is better for variable names: camel case or snake case?
```

The output would be similar to what is shown below. You can iteratively prompt for more specific questions you may have.

▶ Click here to view the sample response generated

Change the case

Assume your organization wants to standardize snake case for readability. In the code we have here, there are not too many lines of code. In real-life scenarios, it is rarely the case. Changing the case manually can be very challenging and possibly error-prone. You can use Gen AI to do the same for you. Type the following prompt:

```
Change the case of the variable names to snake case.
```

The output would be similar to what is given below. Notice that all the variable names are retained with just the case changed.

▶ Click here to view the sample response generated

Including comments

Comments are extremely important for code readability. It is always good coding practice to include comments explaining what the code does. Type the next prompt as:

```
Include comments in the code which will help in readability of the code
```

▶ Click here to view the sample response generated

Conclusion

Congratulations! You have created a complete documentation for a JavaScript code in such little time. If you are happy with the output produced, you may copy the code and use it. If you want to improvise, you may ask additional relevant questions. Generative AI depends on external sources to supplement its responses with more facts and realism. However, it is your prerogative as a software developer to ascertain the accuracy of the output.

You can always visit the conversation during the live session using the history option by clicking on the main menu on top-left.

Then, choose the chat by title among all the labs listed.

Author(s)

Ramanujam
Lavanya TS