Review Code with Generative AI

Estimated time needed: 30 minutes

As a software developer, you may be very comfortable coding. But when it comes to maintaining some of the coding standards, taking care of security vulnerabilities, styling and other overlooked errors, it is often a challenge, as the focus is on the logic and the code. Generative AI uses Retrieval Augmented Generation and aids you in reviewing the code for you.

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 ones and the response received. Change the wording if required to get 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 Review code
2. Iteratively fix issues in code and check

Prerequisites

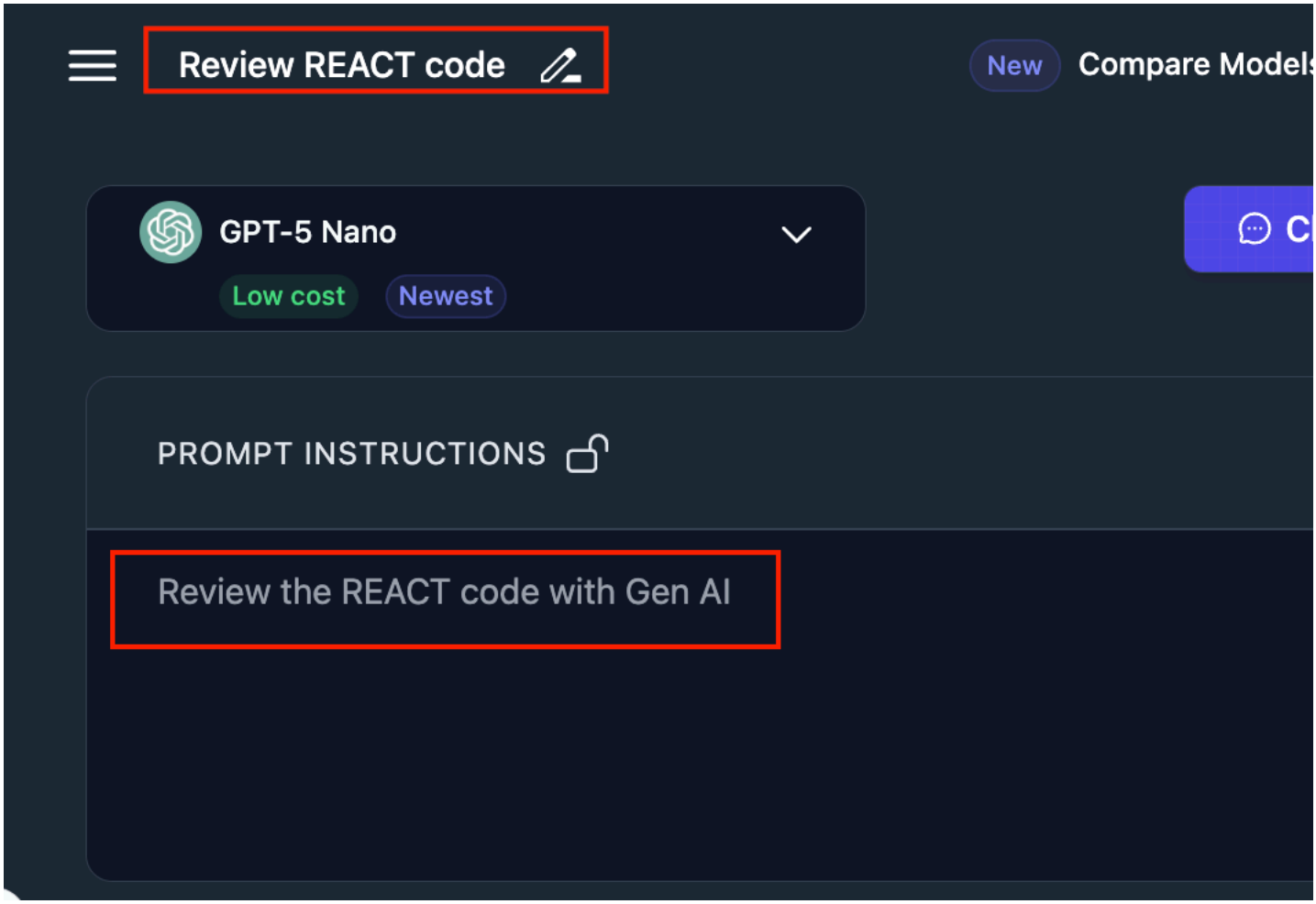
- You should be comfortable with writing REACT code.
- You should have access to an IDE.

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 it will help in continuing the conversation at a late point.

Also, provide a prompt instruction that is specific to the conversation in this particular lab. Let's get started with the task of reviewing the REACT 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 review

You are comfortable with REACT coding. You have created the code in JavaScript. You will now use Gen AI model to create the same in Python.

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

```
I have a written REACT code. Can you review the code?
```

If you want more specific review for security lapses or styling you may add that to your prompt.

```
I have a written REACT code. Can you review the code for styling as per standards?
```

or

```
I have a written REACT code. Can you review the code for security vulnerability?
```

If the GenAI model has the ability to review, you will get an affirmative response in line with what is shown in the image below. 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 code in the prompt and send it. This process will take a few seconds as it needs to process the code given and provide the reviews.

```
import React, { useState } from 'react';
const movies = [
{ title: 'The Shawshank Redemption', genre: 'Drama' },
{ title: 'The Godfather', genre: 'Crime' },
{ title: 'Pulp Fiction', genre: 'Crime' },
{ title: 'The Dark Knight', genre: 'Action' },
{ title: 'Fight Club', genre: 'Drama' },
];
const MovieRecommendationApp = () => {
const [genre, setGenre] = useState('');
const [recommendation, setRecommendation] = useState('');
const handleGenreChange = (event) => {
setGenre(event.target.value);
};
const handleRecommendation = () => {
const moviesByGenre = movies.filter((movie) => movie.genre === genre);
const randomIndex = Math.floor(Math.random() * moviesByGenre.length);
const recommendedMovie = moviesByGenre[randomIndex];
if (recommendedMovie) {
setRecommendation(recommendedMovie.title);
} else {
setRecommendation('No movies found for the selected genre');
}
};
return (
<div>
<h1>Movie Recommendation App</h1>
<label htmlFor="genre">Select a genre:</label>
<select id="genre" value={genre} onChange={handleGenreChange}>
<option value="">-- Select --</option>
<option value="Drama">Drama</option>
<option value="Crime">Crime</option>
<option value="Action">Action</option>
</select>
<button onClick={handleRecommendation}>Get Recommendation</button>
{recommendation && (<p>Recommended Movie: {recommendation}</p>)}
</div>
);
};
export default MovieRecommendationApp;
```

If the previous prompt did not specify what kind of review is required, the response might be too generic as given below.

▶ Click here to view the sample response generated

Styling specific code review

If the previous response you obtained was to generic, you need to prompt with a more specific ask. This will help Generative AI give a focused response. If is always good to give the context for Chat GPT to evince specific answers to the question.

```
Review the code for styling as per standards.
```

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

Make changes as per the suggestions in any of your editors like notepad or vscode.

Review to check feedback after correction

Now that you have made the suggested changes in your notepad or vscode (or any other code editor), copy and paste the code in the space provided and get the corrected code reviewed again.

```
I have made the changes as per suggestions. Please review to see if it ok now.
# Paste the corrected code here
```

The output would be similar to what is given below if all the suggestions are implemented. If all suggestions are not implemented, you can do so iteratively.

▶ Click here to view the sample response generated

Confirm latest version

While it is fair to assume that the Gen AI model will call out use of an older version of REACT, for example code with class components, it is always safe to check and confirm.

Type the next prompt as:

```
Is this code inline with the latest version of REACT?
```

If it is as per the latest standards, you will get a response similar to the one below.

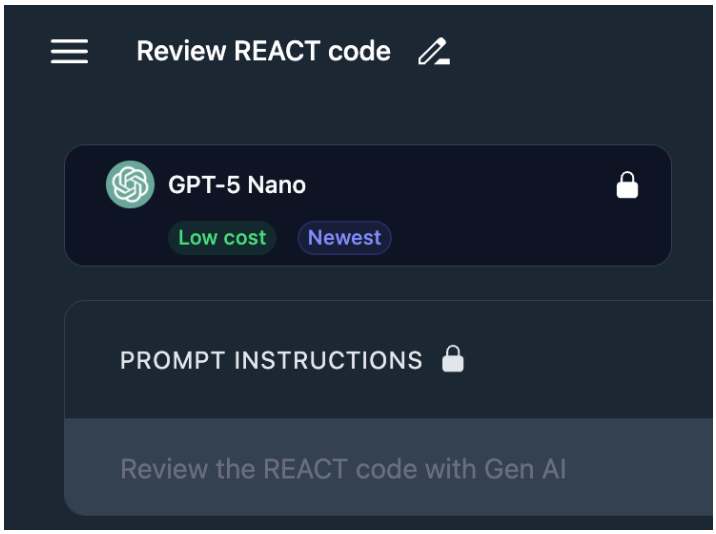
▶ Click here to view the sample response generated

You can try to further prompt for REACT to do a security vulnerability review, programming errors review and so on.

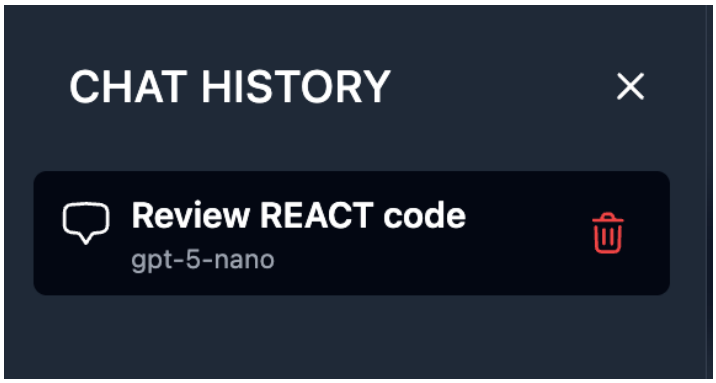
Conclusion

Congratulations! You have reviewed and corrected the REACT code. If you are happy with the output produced, you may use the code as the final code. If you want to improvise, you may ask additional relevant questions. Generative AI depends on external sources for supplementing the responses with more facts and realism. Due diligence should be given to the fact that Generative AI is still nascent and evolving. Individual judgement should also be taken into consideration.

You can always view the conversation, during the live session using the history option, by clicking on the burger menu on top-left.



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



Author(s)

Rama
Lavanya