# **POC Documentation**

# **Introduction**

# The Testcases Generating project is a web application built to streamline the creation of test cases across different scenarios. It utilizes a Web API core in the backend to manage the business logic, while employing React in the frontend to deliver an intuitive user interface.

P**rerequisites/Technologies**

* .net 6 with Web API core.
* Node.js and npm
* Open AI Key

**Explanation**

• I employed React for the frontend and .NET Core Web API for the backend.

• Leveraging .NET Core Web API, I developed a controller responsible for automatically generating test cases using the Open AI key.

• With React, I crafted a user interface seamlessly integrated with the backend functionalities.

**Usage Guide/Run**

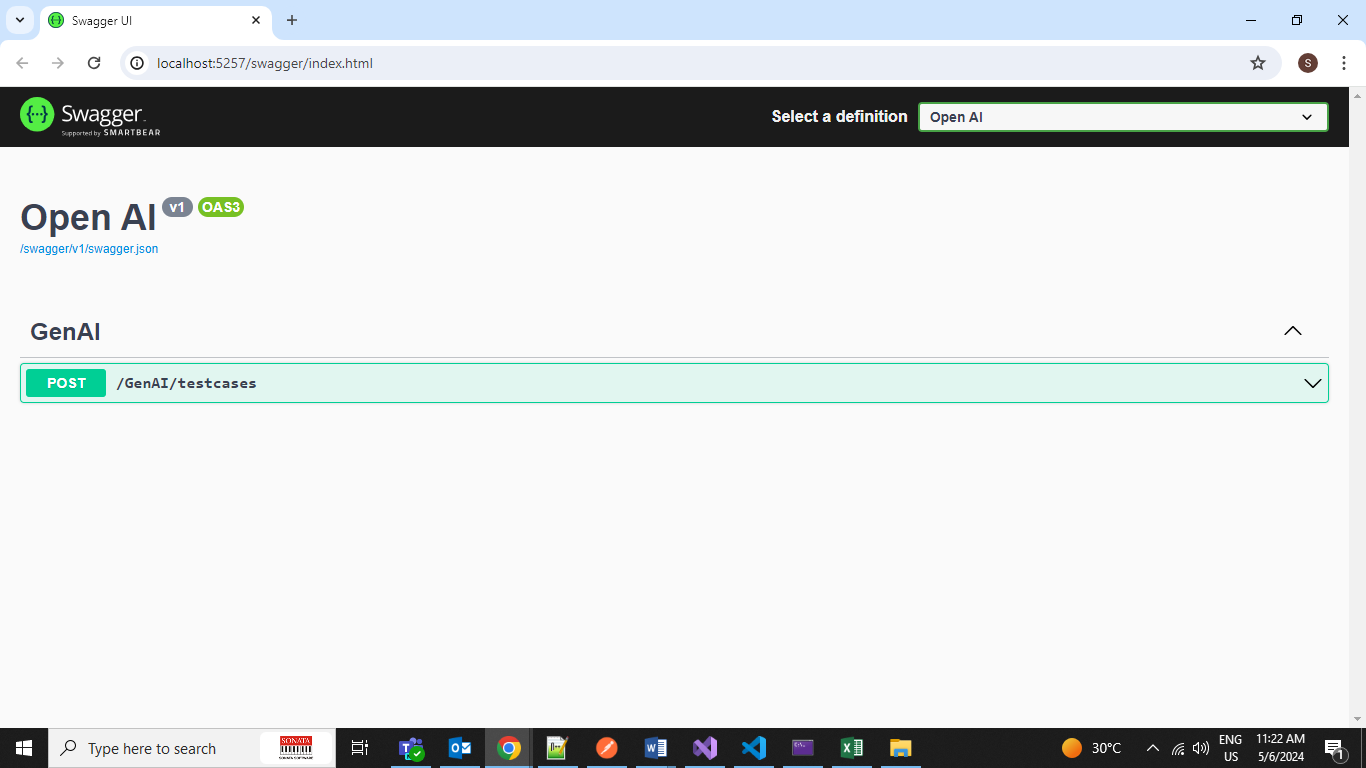
**Run the Web API application:**

To start the Web API application, open the solution file in Visual Studio, then build the code and run it using the F5 key. This will automatically launch Swagger in the Chrome browser.

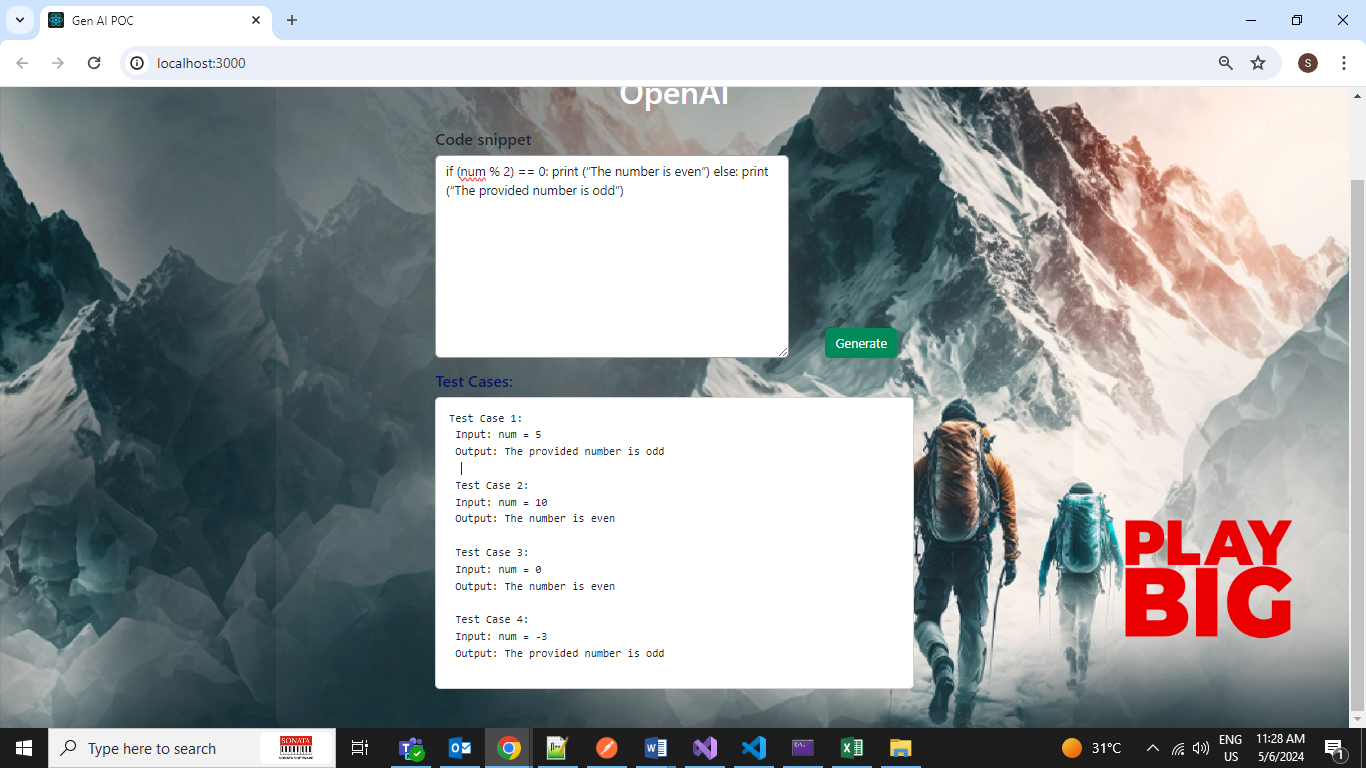
**Run the React application**

To Run the react application (TestCasesPocUI) follow the below commands.  
 npm install  
 npm start  
By Default it will open on the port number 3000(http://localhost:3000)

You can see the Swagger URL once you successfully ran the web API application.



You can see the output of the React application in the below snap



**Input and output**

If we gave the code snippet in the UI as input, it would give the testcases as Output.

**Sample Input:**

"if (num % 2) == 0: print (“The number is even”) else: print (“The provided number is odd”)"

**Sample output:**

Test cases are:

1. Test case: num = 4 Expected output: The number is even

2. Test case: num = 7 Expected output: The provided number is odd

3. Test case: num = 0 Expected output: The number is even

4. Test case: num = -6 Expected output: The number is even

5. Test case: num = 15 Expected output: The provided number is odd

**References**

[Asp.net](https://spring.io/projects/spring-boot) Core

[React](https://legacy.reactjs.org/docs/getting-started.html)

**Conclusion**

In implementing the Testcase Generation Proof of Concept (POC) through the integration of React and Web API core application, coupled with the utilization of an API key, significant time and effort savings for developers have been achieved. This innovative approach not only streamlines the testing process but also enhances productivity by automating the generation of test cases.