Dear Amy.

I have completed my assignment and here is the link of the source code  
  
<http://glider180.sytes.net:500/glider180/CalculatorAPI.rar>

Following are the few details about approaches/methodologies used in the project.

1. The WebAPI named “CalculatorAPI” has been created in .NET Core v3.1
2. The API has four core functions i.e. Add, Subtract, Multiply & Divide with 2 input parameters each.
3. .NET Core’s Dependency Injection pattern has been used in all controllers for a loosely coupled architecture.
4. All the functions are being asynchronously called for better performance & responsiveness.
5. All the inputs are validated before proceeding to next stage.
6. All the functions return proper HTTP Based Responses i.e. 200 OK, 500 Internal Server error etc to the callers.
7. All the exceptions are being centrally handled in Startup.cs class file & all the exceptions are being properly thrown in Response Http Header of “Application-Error”.
8. The successful result is returned in well-structured form with “result” variable prefixed.
9. Each individual Function’s input parameters & calling Time can be logged by configuration, as in real world sometimes we need to log all or particular api calls and sometimes we don’t need to log any calls.
10. .NET Core Dependency Injection has been used to configure the Logging Destination i.e. on Text File or in DB by changing at only one place i.e. to achieve a loosely coupled architecture approach. Currently implementation to log all info on text file is implemented.
11. The configuration to determine which particular controller we have to log and which one to discard, can be picked from asppsettings.json file or From DB as well by just changing at only one place i..e “LoadConfiguration” key in appsettings.json file to avoid any large code changes. (Write now configuration is being picked from config file).  
      
    ***Additionally & Optionally***
12. Another replica Controller “**CalculateSecure**” with same arithmetic operations is implemented to demonstrate the security features in API Calls.
13. JWT Tokens are being used to Make all HTTP API Calls in this controller, this can be achieved by calling CalculateSecure/Auth/Login first

And then passing username “admin” & password “admin” (hardcoded just for demo) and in successful response it will return a SHA512 based JWT Token and then CalculateSecure/Add/Multiply/Divide/Subtract all functions can be called by passing the received JWT token in the HTTP Authorization header (screenshots attached for demonstration purpose).  
  
 **Unit Test Cases**

1) A separate project “XUnitTestCalculator.API” has been attached which has total 8 test cases out of which 4 test cases are to validate that proper HTTP responses are coming from web api’s controllers.

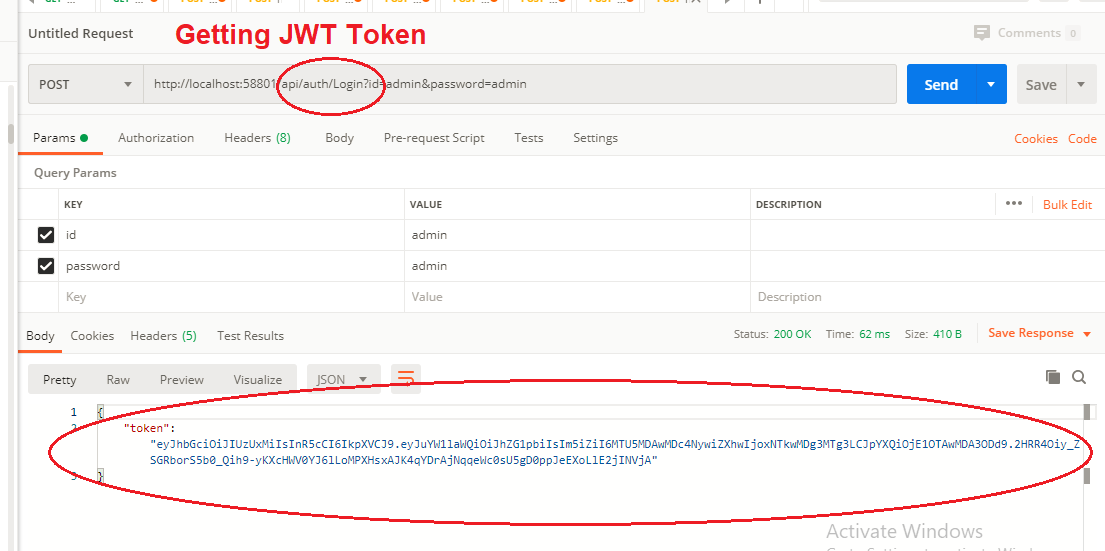
2) Another 4 test cases are written to validate the functional/business response of the arithmetic functions.   
  
All the test cases are being passed (Screenshot attached).  
  
  
Please feel free to write me back in case of any queries/concerns.

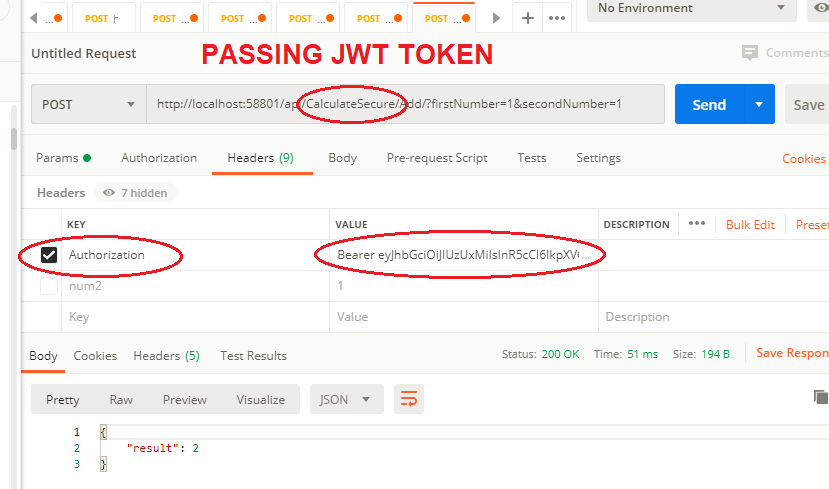
Regards,

Navaid.  
navaid@hotmail.com  
0452557480

**Screenshots**.

1. Getting & Passing JWT Token





1. Unit Test Results  
     
   