**Customer Support Logger document**

**Git Link:**

[**https://github.com/pravallikaadineni/CustomerSupportLogger.git**](https://github.com/pravallikaadineni/CustomerSupportLogger.git)

**Azure SQL**

* Create a SQL server and SQL database in azure portal and connect it to ssms
* Create tables user info and custloginfo tables in ssms
* Insert values in tables

**Visual studio**

* Create a class library project (DAL)
* Create a class library project (DAL Test)
* Create ASP.Net Web MVC application project with Docker Support
* Install entity framework to the project
* Add azure SQL connection to it
* Write the functionalities to validate the user information
* In MVC project add references and dependencies of dal
* Add controllers and models and views in MVC application
* Build and run the project
* In dal test class write the test cases to validate the users info
* Build and run in test explorer

**JENKINS:**

* Push project to the Git.
* Now goto Jenkins Dashboard.
* Choose Freestyle project.
* Goto Github and copy the repository link.
* Once you save the project is saved in Jenkins.
* Choose Build now on the left to build it.
* Click on build history to check the build status.
* Click on it. Goto Console output.
* Configure the MSBuild in Jenkins to build it. Goto Manage Jenkins.
* Goto Manage Plugins.
* Goto Mange Jenkins -> Global Tool Configuration.
* Choose Add MSBuild
* Save.
* Once the MSBuild plugin is installed successfully goto the project.
* Goto Configure
* Choose visual studio project
* Choose the MSBuild version from the dropdownlist and select msbuild.
* Now goto the project and build the solution.
* It should build the project and should be successful if there are no build errors in the code.

**DOCKER**

* + Build and run the project
  + Open file explorer and paste docker file
  + Open command prompt and executed the commands
    - **Docker build –t pravallikaadineni/dal .**
  + After this docker image is created
    - **Docker run -p 9080:80 pravallikaadineni/dal .**
  + A container is created
  + Success now our application is running as a docker image in this URL