Q. 1. Create a ASP.NET MVC project using C# and .NET Framework in Visual Studio 2019 Community Edition. Complete the build structure in the following MSBuild Process. Copy the following targets at the end of the .csproj file before </Project>. Run the targets Build and Clean separately and attach the screen shots of the output in PDF format.

  <Target Name="CustomAfterBuild" AfterTargets="Build">

<PropertyGroup>

     <\_OutputCopyLocation>$(OutputPath)..\..\CustomOutput\</\_OutputCopyLocation>

  </PropertyGroup>

    <ItemGroup>

      <\_FilesToCopy Include="$(OutputPath)\*\*\\*"/>

    </ItemGroup>

    <Message Text="\_FilesToCopy: @(\_FilesToCopy)" Importance="high"/>

    <Message Text="DestFiles:

        @(\_FilesToCopy->'$(\_OutputCopyLocation)%(RecursiveDir)%(Filename)%(Extension)')"/>

a.      <!—Write a build statement which will copy all the output files to a new folder **CustomOutput**-->

  </Target>

  <Target Name="CustomClean" BeforeTargets="Clean">

<PropertyGroup>

     <\_OutputCopyLocation>$(OutputPath)..\..\CustomOutput\</\_OutputCopyLocation>

  </PropertyGroup>

    <Message Text="Inside Custom Clean" Importance="high"/>

    <ItemGroup>

      <\_CustomFilesToDelete Include="$(\_OutputCopyLocation)\*\*\\*"/>

    </ItemGroup>

b.      <!—Write a build statement to clean up the files created by the custom build operation with a CustomClean target by using a BeforeTargets attribute and specifying that the custom clean operation runs before the CoreClean target-->

  </Target>

</Project>

Q. 2. Create a Maven enabled Java project in Eclipse. Create a sample java file in the /src/main/java folder. Complete the POM structure in the following Maven Build Process: Copy the code into pom.xml file. Build the project and attach the screen shots of the output in PDF format.

<project>

  <build>

    <!—Specify the TARGET Dirctory -->

    <!—Specify the OUTPUT Dirctory -->

    <finalName>${project.artifactId}-${project.version}</finalName>

    <testOutputDirectory>${project.build.directory}/test-classes</testOutputDirectory>

    <!—Specify the SOURCE Dirctory -->

    <scriptSourceDirectory>${project.basedir}/src/main/scripts</scriptSourceDirectory>

    <testSourceDirectory>${project.basedir}/src/test/java</testSourceDirectory>

    <resources>

      <resource>

        <directory>${project.basedir}/src/main/resources</directory>

      </resource>

    </resources>

    <testResources>

      <testResource>

        <directory>${project.basedir}/src/test/resources</directory>

      </testResource>

    </testResources>

  </build>

</project>

Q. 3. Write the steps to create a repository named MySourceHub in GitHub and clone the repository using [Git](https://mazenet-tms.com/mod/lesson/view.php?id=1218) Commands. Push a sample application using [Git](https://mazenet-tms.com/mod/lesson/view.php?id=1218) Commands to GitHub Repository named MySourceHub and attach the screen shots in PDF format.