Allure & Feature Replication Guide This guide explains how to add Allure reporting and replicate other important features from the original Java-based project into the C# POM implementation. ## 1. Adding Allure Reporting - **Project file updates**: Edit `POMCSharpAutomation.csproj` and add the following package references: ```xml   <PackageReference Include="Allure.NUnit" Version="2.12.1" />   <PackageReference Include="Allure.Net.Commons" Version="2.12.1" />   ``` - **Allure configuration**: Create an `allureConfig.json` file in the root of your project with a minimal configuration: ```json   {   "allure": {   "directory": "allure-results"   }   }   ``` - **Copy configuration to output**: Update your `.csproj` file so the Allure configuration is copied to the output directory: ```xml   <ItemGroup>   <None Update="allureConfig.json">   <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>   </None>   </ItemGroup>   ``` - **Annotate tests**: Decorate your NUnit test classes with `\[AllureNUnit]` and import the Allure namespace from `Allure.NUnit`. This activates Allure reporting for those tests. ## 2. Other Feature Replications - **Listeners**: Implement custom NUnit listeners if you need to mirror the Java project's listeners such as `ExtentReportListener` or other event hooks. NUnit provides interfaces for test event listening. - **Utilities**: Port utility classes like Excel reading (`ExcelUtil`), JavaScript execution helpers, and timing utilities (`TimeUtil`) by translating their logic into C# using relevant .NET libraries. - **Additional pages**: Continue adding page classes (e.g., `AmazonPage`, `CartPage`, etc.) following the Page Object pattern demonstrated in `LoginPage` and `AccountsPage`. - **Additional tests**: Create extra NUnit tests for new pages and features, similar to `LoginPageTest` and `AccountsPageTest`. ## 3. Conclusion Following these steps will allow you to add Allure reporting and achieve feature parity between the original Java project and your new C# implementation. Continue expanding the project as needed by implementing remaining pages, utilities, and listeners.