**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **15/07/2020** | **Name:** | **Dhavala** |
| **Course:** | **Coursera**  **Mathematics for Machine learning: Linear algebra** | **USN:** | **4AL17EC027** |
| **Topic:** | **Matrices, Vectors and solving simultaneous equation problem** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **SESSION DETAILS** |

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **15/07/2020** | **Name:** | **Dhavala** |
| **Course:** | **Salseforce** | **USN:** | **4AL17EC027** |
| **Topic:** | * **Lighting Web components** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **SESSION DETAILS**      **Introduction**  Lightning Web Components is a new programming model for building Lightning components. It uses web standards breakthroughs, can coexist and interoperate with the Aura programming model, and delivers unparalleled performance. To create and develop Lightning Web Components and use their powerful features and performance benefits, you need to set up Salesforce DX. For this Quick Start, you also use Visual Studio Code, which is the recommended code editor for developing on the Salesforce platform. Once your developer environment is ready, you learn how to write a simple Lightning web component and add it to a page in Lightning Experience.  The Salesforce Developer Experience (DX) is a set of tools that streamlines the entire development life cycle. It improves team development and collaboration, facilitates automated testing and continuous integration, and makes the release cycle more efficient and agile. **Set Up Visual Studio Code**Install Salesforce Extensions for Visual Studio Code Visual Studio Code is the go-to code editor for Salesforce developers. It's free, open-source, and available for Windows, Linux, and macOS. This editor has easy-to-install extensions for syntax highlighting, code completion, and more.  In this project, we install Visual Studio Code and the recommended Salesforce Extension Pack.   1. Download and install the latest version of [Visual Studio Code](https://code.visualstudio.com/) for your operating system. If you already have Visual Studio Code installed, there’s no need to reinstall it. 2. Launch Visual Studio Code. 3. On the left toolbar, click the Extensions icon Visual Studio Code's Extension icon. . 4. Search for Salesforce Extension Pack and click **Install**. If you already have it installed, then you just need to click on the **Reload** button.   **Ensure Your Development Environment Is Ready**  Now that you’ve installed Visual Studio Code and enabled the necessary extensions, you need to test them out.   1. In Visual Studio Code, open the Command Palette by pressing Ctrl+Shift+P (Windows) or Cmd+Shift+P (macOS). 2. Enter sfdx to filter for commands provided by the Salesforce Extensions.   As you use more SFDX commands, those commands will show up in the recently used panel.  In the final step, you create your first Lightning web component and add it to your org’s home page.  We won’t check any of your setup. Click **Verify Step** to go to the next step in the project.  **Create a Salesforce DX Project**   1. In Visual Studio Code, open the Command Palette by pressing Ctrl+Shift+P (Windows) or Cmd+Shift+P (macOS). 2. Type SFDX. 3. Select **SFDX: Create Project**. 4. Press Enter to accept the standard option. 5. Enter HelloWorldLightningWebComponent as the project name. 6. Press Enter. 7. Select a folder to store the project. 8. Click **Create Project**. You should see something like this as your base setup. Visual Studio Code with the newly created HelloWorldLightningWebComponent folder   **Authorize Your Trailhead Playground**   1. In Visual Studio Code, open the Command Palette by pressing Ctrl+Shift+P (Windows) or Cmd+Shift+P (macOS). 2. Type SFDX. 3. Select **SFDX: Authorize an Org**. 4. Press Enter to accept the Project Default login URL option. 5. Press Enter to accept the default alias. 6. Log in using your Trailhead Playground credentials. 7. If prompted to allow access, click **Allow**. Allow Access dialog 8. After you authenticate in the browser, the CLI remembers your credentials. The success message should look like this: Success message of authorizing an org |