**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **28 may 2020** | **Name:** | **Shreya poojary** |
| **Course:** | **Python** | **USN:** | **4al16ec074** |
| **Topic:** | **Build a personal website with python** |  | **8-B** |
| **Github Repository:** | **Shreya-test** |  |  |

|  |
| --- |
|  |
| **Image of session** |
| **BROWSER CATCHING**  we will add CSS styling to the webpage. Sometimes, when you make a change to the CSS file and reload the webpage, the changes are not shown because the browser uses the previous cached styling. If this happens, open the browser in private (incognito) mode and load the webpage there.  **HTML TEMPLATE**  The <template> tag holds its content hidden from the client.  Content inside a <template> tag will not be rendered.  The content can be made visible and rendered later by using JavaScript.  Use the <template> tag when you have HTML code you want to use over and over again, but not until you ask for it. To do this without the <template> tag, you have to create the HTML code with JavaScript to prevent the browser from rendering the code.  The **HTML Content Template (**<template>**) element** is a mechanism for holding [HTML](https://developer.mozilla.org/en-US/docs/Glossary/HTML) that is not to be rendered immediately when a page is loaded but may be instantiated subsequently during runtime using JavaScript.  Think of a template as a content fragment that is being stored for subsequent use in the document. While the parser does process the contents of the <template>element while loading the page, it does so only to ensure that those contents are valid; the element's contents are not rendered, however.  <table id="producttable">  <thead>  <tr>  <td>UPC\_Code</td>  <td>Product\_Name</td>  </tr>  </thead>  <tbody>  <!-- existing data could optionally be included here -->  </tbody>  </table>  <template id="productrow">  <tr>  <td class="record"></td>  <td></td>  </tr>  </template>  The Python script handles the communication between the web server and the web client (i.e. browser) while the HTML documents are responsible for the structure of theage content.  Now we need to add some style formatting to the HTML structure using CSS (Cascading Style Sheets). That is done by creating a CSS file and connecting it to our HTML files. CSS is a style language that likewise HTML it is also very easy to learn. Python is much harder to learn than CSS,. So a rule of thumb is if you know Python, learning CSS should be a breeze.  Remember that HTML template files HTML go inside the templates folder. CSS stylesheets are considered static files. There is no interaction with their code, like there is with HTML templates. Therefore, flask has reserved a separate folder where you should put static files such as CSS, Javascript, images or other files. That folder should be created by you and should be named static. It’s also good practice to create anotherfolder inside static and name it css.Now, create an empty file inside the css and name the file something like main.css How to create Virtual Environment?Step – 1 Open your terminal and create a directory to store all your virtual environments, using the command mkdir Environments which is an acronym of “make directory”.Now go inside the directory using the command CD which stands for call Directory, CD Environments Step 2 Now we will use a module named **virtualenv** to create isolated virtual environments.But first, let’s install this module by the following command,  pip install virtualenv  If you get an error like pip command not found then you have to install **pip** package manager first, you can learn this here.To verify a successful installation run this  virtualenv–versionNow we can proceed to create virtual environment. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  | | | |
|  | | | |
|  | | | |