----------------------**Selenium Test in Azure Pipelines**-----------------------

* This template will help you to have a data driven framework and Selenium test in azure pipelines.Here we are using apache POI to read data from a excel file.
* **Why we need Azure pipelines for integration:-**
  + we can have an automated way of running our regression test. Whenever we have any changes to our VCS repo and we can have our test run in parallel along with the project build.So that we can identify any bugs or defects even when there is a small change of functionality or code.
* I was using -
  + **Java** as the programming language
  + **TestNG** as the assertion framework
  + **Maven** as the build tool
  + **WebDriver** as the browser automation tool
  + **IntelliJ** as the IDE
* Create new maven project and give the project name - Headless chrome framework
* Add the dependency in pom file.
* Add the plugins - TestNG and Maven Helper.
* **Type of framework** - Data driven framework.
* **Packages** - We have a separate package for Com and TestCase and Listeners. All web page related class under in Com package and all test cases in under TestCase and ExtentReport content class under in Listeners.
* **Com package---**
* **Content class** - This class is responsible for loading the test cases from excel sheet.
  + Download Apache POI libraries dependency in pom files.
  + **XSSF(XML Spreadsheet Format)** this is the interface using for read and write the data from excel sheet using **XSSFWorkbook workbook.**
* **DriverUtility class** - This class stores and handles the functions (The code which is repetitive in nature such as waits, capturing screenshots, accessing excels, sending email, etc.,) which can be commonly used across the entire framework.In this class load the properties file and put the urlkeys. And also we handle calendar dates through excel sheet.
* **OpenBrowser Interface** - This file stores the information that remains static throughout the framework such as browser path, application URL, Key path etc.
* **FillTheForm class** - This class is responsible for reading all sheet header values from excel sheet using **List<SheetColumnHeader> sheetColumnHeaders = fileUtil.readFile();**and adding the assertion for each test cases with using **Assert.assertequal()** method.
* **Object.properties -** This file stores the all keys path(css selector) for button and text field.
* **Listeners package ---**
* **ExtentReport class -** This class generate the report.
  + First load the properties file. And set the report path.
  + Set the log with status with test case -
    - **(context.getPassedTests(), LogStatus.PASS);**
    - **(context.getFailedTests(), LogStatus.FAIL);**
    - **(context.getSkippedTests(), LogStatus.SKIP);**
  + Set the description of each test case with using **.getParameters()[0]toString()**
  + Add the Time calendar.
* **Testcase package ---**
* In this package we store excel sheet( store all test cases).
* Create TestNG.xml file and add the Listener in TestNG file
  + **<listeners>**
  + **<listener class-name="Listeners.ExtentReport"/>**
  + **</listeners>**
* **Run the Framework.**
* Check the **user.Extent.html** fileand run this file and get the report.
* **Some dependencies we use in this framework:-**
* To read data from an excel file we need Apache POI maven dependency

**<dependency>**

**<groupId>org.apache.poi</groupId>**

**<artifactId>poi</artifactId>**

**<version>3.9</version>**

**</dependency>**

**<dependency>**

**<groupId>org.apache.poi</groupId>**

**<artifactId>poi-ooxml</artifactId>**

**<version>3.9</version>**

**</dependency>**

* TestNg dependency is mentioned below

**<dependency>**

**<groupId>org.testng</groupId>**

**<artifactId>testng</artifactId>**

**<version>7.1.0</version>**

**<scope>compile</scope>**

**</dependency>**

* Extent Report dependency is mentioned below

**<dependency>**

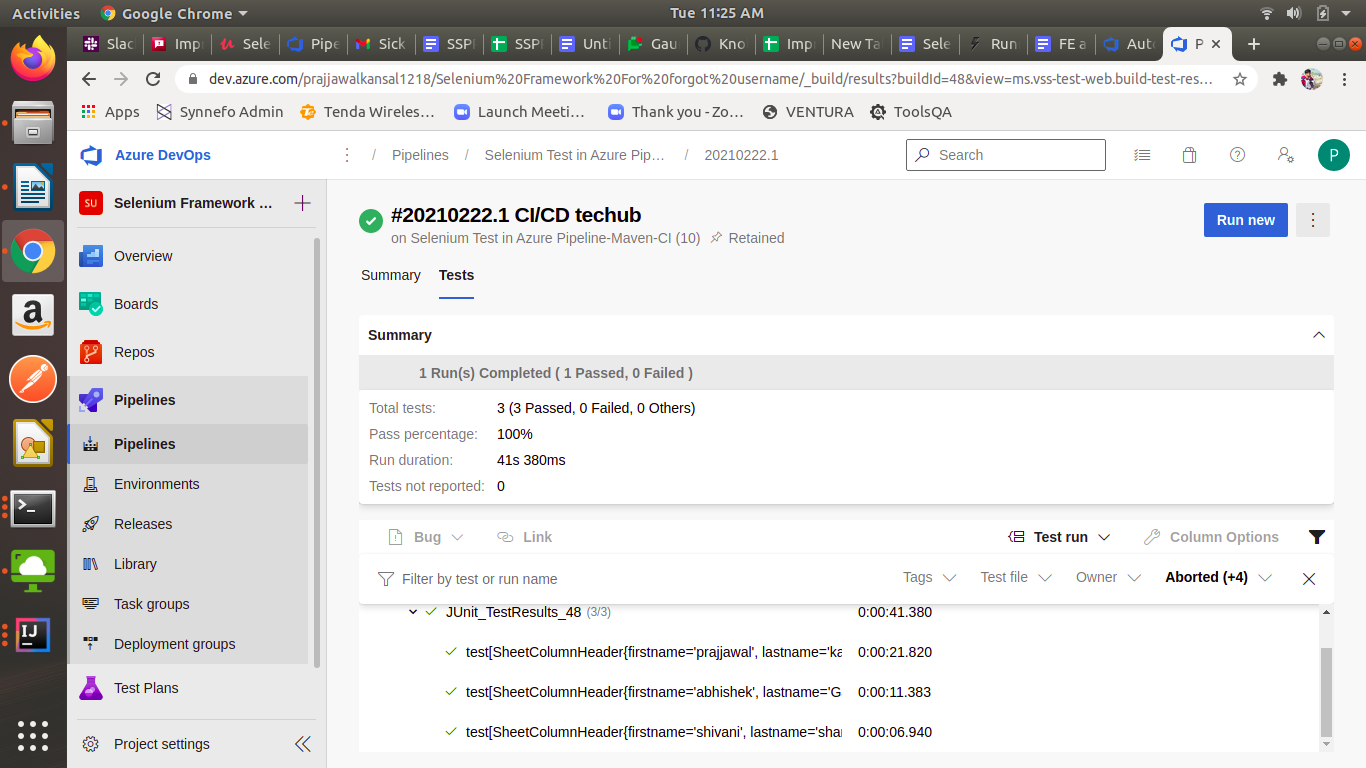
**<groupId>com.relevantcodes</groupId>**

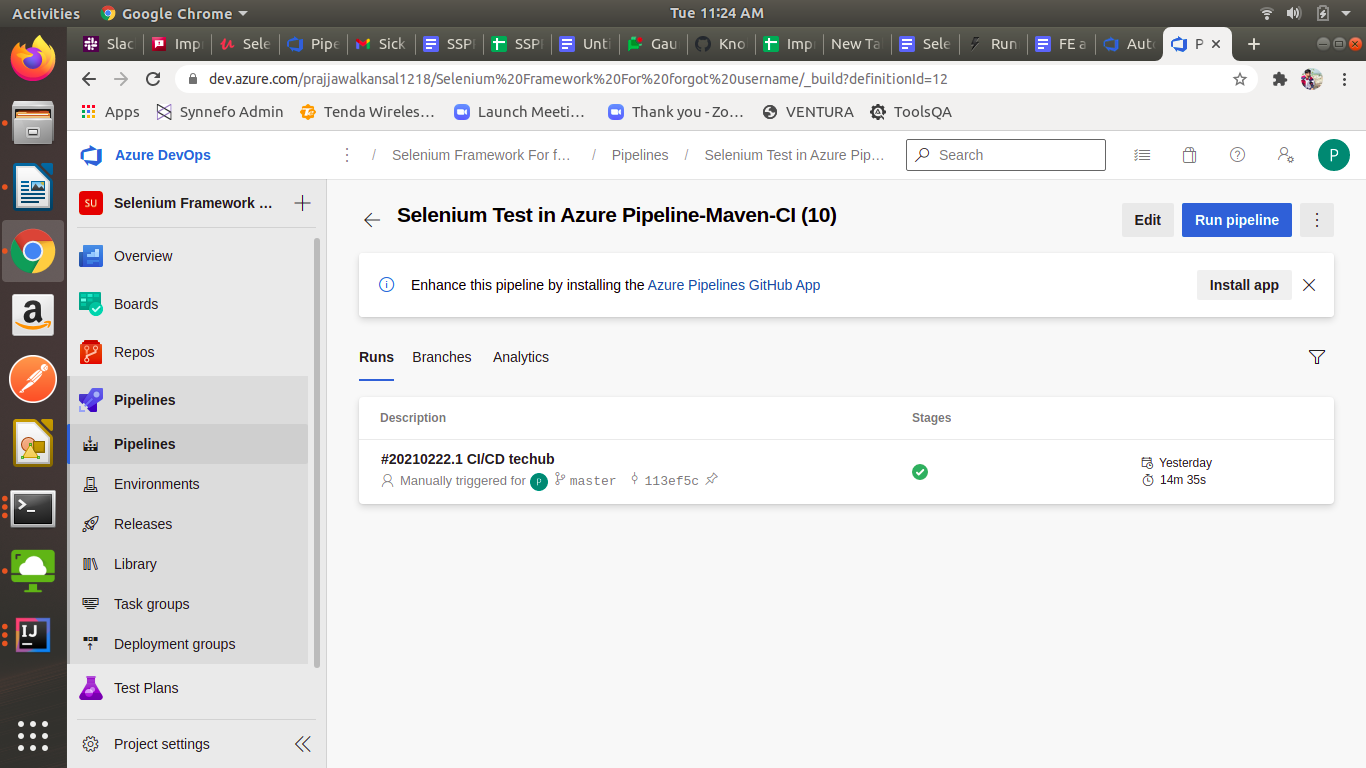
**<artifactId>extentreports</artifactId>**

**<version>2.41.2</version>**

**</dependency>**

* **How to Run in Azure pipeline:-**
  + Go to Microsoft azure url - **https://dev.azure.com** and sign in the microsoft account.
  + Click on start free. After you see yourself in Azure devops page.
  + Create a new project.
  + Create a pipeline .
  + Click on this [**Use the classic editor**](https://dev.azure.com/prajjawalkansal1218/selenium tests in azure/_apps/hub/ms.vss-ciworkflow.build-ci-hub?_a=build-definition-getting-started&id=0)**.** Pickup the selenium code from github.
* **Result in Azure pipeline:**





* **Future Scope of Selenium Framework** --
  + Framework also allow you to read the data from an external data source like excel sheet or database and input the data in your web or desktop application thereby making your Automation script reusable.
  + Framework also allow to load the properties files where we store key path and urls and driver path.
  + Framework also allow to help the adding assertion for all test cases.
  + Framework help to print the description in report through excel sheet.
  + Framework help to handle calender dates with excel sheet and radio button for different different data.