**Set 1**

**Question 1:**

public class Solution1 extends TestBase{

@BeforeMethod

@Parameters({ "browser" })

public void beforeMethod(String browser) throws Throwable {

init(browser, "https://exammay2020.agiletestingalliance.org/");

checkPageReady();

}

@Test

public void f() throws InterruptedException {

d.findElement(By.xpath("//\*[@class='eicon-close']")).click();

checkPageReady();

List <WebElement> more=d.findElements(By.xpath("//\*[contains(text(),'More')]//ancestor::a[1]"));

for(WebElement e : more) {

System.out.println(e.getText() +" : "+e.getAttribute("href"));

}

scrollIntoView(d.findElement(By.className("social\_icon")));

takeScreenShot(d, "Social\_icon");

List<WebElement> socialIconLinks=d.findElements(By.xpath("//\*[@class='social\_icon']//a"));

String mainwindow=d.getWindowHandle();

for(WebElement icon : socialIconLinks) {

System.out.println(icon.getAttribute("href"));

icon.click();

Set <String> childwindow=d.getWindowHandles();

Iterator <String> winItr=childwindow.iterator();

while(winItr.hasNext()) {

String child=winItr.next();

if(!child.equals(mainwindow)) {

d.switchTo().window(child);

checkPageReady();

System.out.println(d.getTitle());

takeScreenShot(d, "Social\_Pages");

d.close();

}

d.switchTo().window(mainwindow);

}

}

List<WebElement> doppatext=d.findElements(By.xpath("//\*[contains(text(),'doppa2020')]"));

System.out.println("Doppa2020 is displayed : "+ doppatext.size());

}

@AfterMethod

public void afterMethod() {

d.quit();

}

}

**Question 2:**

public class Solution2 extends TestBase {

int counter=1;

@BeforeMethod

@Parameters({ "browser" })

public void beforeMethod(String browser) throws Throwable {

init(browser, "https://exammay2020.agiletestingalliance.org/");

waitForElementToVisible(d.findElement(By.xpath("//\*[@class='eicon-close']")));

d.findElement(By.xpath("//\*[@class='eicon-close']")).click();

mouseover(d, d.findElement(By.xpath("//ul[@id='menu-main-1']//a[text()='Data']")));

d.findElement(By.xpath("//ul[@id='menu-main-1']/li[4]/ul/li[1]/a[1]")).click();

checkPageReady();

}

@Test(dataProvider = "datapro")

public void f(String id, String name, String address, String city, String country) throws Exception {

HashMap<String,String> act= new HashMap<String, String>();

act.put("ID", id);

act.put("NAME", name);

act.put("ADDRESS", address);

act.put("CITY", city);

act.put("COUNTRY", country);

boolean resultFlag=false;

int count=1;

int pageSize=d.findElements(By.xpath("//\*[@class='ea-advanced-data-table-pagination ea-advanced-data-table-pagination-button clearfix']//a")).size();

int pagecount=1;

boolean flag=false;

for(int j=1;j<5;j++) {

count =j;

String getID=d.findElement(By.xpath("//tbody//tr[@style='display: table-row;']["+j+"]//td[1]")).getText();

if(getID.trim().equalsIgnoreCase(id)){

resultFlag=true;

break;

}

if (count==5) {

d.findElement(By.xpath("//\*[@class='ea-advanced-data-table-pagination ea-advanced-data-table-pagination-button clearfix']//a["+pageSize+"]")).click();

checkPageReady();

j=1;

pagecount++;

if(pageSize==pagecount +2) {

break;

}

}

}

String getID=null;

String getName=null;

String getAdd=null;

String getCity=null;

String getCountry=null;

if(resultFlag) {

System.out.println("PASS");

getID=d.findElement(By.xpath("//tbody//tr[@style='display: table-row;']["+count+"]//td[1]")).getText();

getName=d.findElement(By.xpath("//tbody//tr[@style='display: table-row;']["+count+"]//td[2]")).getText();

getAdd=d.findElement(By.xpath("//tbody//tr[@style='display: table-row;']["+count+"]//td[3]")).getText();

getCity=d.findElement(By.xpath("//tbody//tr[@style='display: table-row;']["+count+"]//td[4]")).getText();

getCountry=d.findElement(By.xpath("//tbody//tr[@style='display: table-row;']["+count+"]//td[5]")).getText();

HashMap<String,String> exp=new HashMap<String,String>();

exp.put("ID", getID);

exp.put("NAME", getName);

exp.put("ADDRESS", getAdd);

exp.put("CITY", city);

exp.put("COUNTRY", getCountry);

flag=exp.equals(act);

}

String result="";

if(flag) {

result="PASS";

}

else

{

result="FAIL";

}

setData(Constants.pathForWritingData, "Sheet1", counter, 0, id);

setData(Constants.pathForWritingData, "Sheet1", counter, 1, name);

setData(Constants.pathForWritingData, "Sheet1", counter, 2, address);

setData(Constants.pathForWritingData, "Sheet1", counter, 3, city);

setData(Constants.pathForWritingData, "Sheet1", counter, 4, country);

setData(Constants.pathForWritingData, "Sheet1", counter, 4, result);

counter++;

}

@DataProvider

public Object[][] datapro() throws Exception {

Object[][] data = readData(Constants.TestDataExcel\_PATH, "Sheet1");

return data;

}

@AfterMethod

public void afterMethod() {

d.quit();

}

}

**Question 3:**

public class Solution3 extends TestBase{

@Before

public void beforeMethod() throws Throwable {

init("Chrome", "https://exammay2020.agiletestingalliance.org/data1");

checkPageReady();

}

@Test

public void f() throws InterruptedException {

Solution3pageObject obj=new Solution3pageObject(d);

obj.getDataSetName();

obj.searchData("india");

}

@After

public void afterMethod() {

d.quit();

}

**POM:**

public class Solution3pageObject extends TestBase {

WebDriver d;

public Solution3pageObject(WebDriver d) {

this.d = d;

}

By header=By.xpath("//h2[contains(text(),'DATA SET 1')]");

By searchBox=By.xpath("//input[@class='ea-advanced-data-table-search']");

By searchResult=By.xpath("//tbody//tr[@style='display: table-row;']//td");

public void getDataSetName() {

String heading=d.findElement(header).getText();

System.out.println("Header is: "+heading);

}

public void searchData(String keyword) {

d.findElement(searchBox).clear();

d.findElement(searchBox).sendKeys(keyword);

List <WebElement> results=d.findElements(searchResult);

System.out.println("Records found= " +results.size());

for(WebElement e: results) {

System.out.println(e.getText());

}

}

}

**Question 4:**

public class Solution4 extends TestBase{

@BeforeMethod

@Parameters({ "browser" })

public void beforeMethod(String browser) throws Throwable {

init(browser, "https://exammay2020.agiletestingalliance.org/");

checkPageReady();

}

@Test

public void f() throws Exception {

wait(2);

d.findElement(By.xpath("//\*[@class='eicon-close']")).click();

checkPageReady();

WebElement ele = d.findElement(By.xpath("//div[@id='primary-menu']//a[contains(text(),'Pages')][1]"));

Actions action=new Actions(d);

action.moveToElement(ele).perform();

d.findElement(By.xpath("//div[@id='primary-menu']//a[contains(text(),'Pages')][1]//following::a[contains(text(),'Certifications')][1]")).click();;

scrollDown();

scrollDown();

wait(2);

d.findElement(By.xpath("//i[@class='eicon-close']")).click();;

takeScreenShot(d, "Upcomingtickers");

List <WebElement> tickers=d.findElements(By.xpath("//div[@class='ticker-content']//a"));

String tickerMsg="";

int counter=1;

for(WebElement ticker: tickers) {

if(counter==1) {

tickerMsg=ticker.getText();

}

else

{

if(tickerMsg.equalsIgnoreCase(ticker.getText())) {

}

else {

System.out.println("The ticker messages are not same");

break;

}

}

counter++;

}

for(WebElement ticker: tickers) {

if(ticker.getAttribute("href")==null || ticker.getAttribute("href")=="") {

}

else

{

System.out.println(ticker.getAttribute("href"));

}

}

}

@AfterMethod

public void afterMethod() {

d.quit();