Step1: package registration;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import org.openqa.selenium.Alert;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.Select;

public class Candidateprofilestep1 {

public WebDriver driver;

public Candidateprofilestep1(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver, this);

}

//Step1 candidate profile

@FindBy(id="Username")

private WebElement uname;

@FindBy(id="Password")

private WebElement pwd;

public void username(String usernme)

{

uname.sendKeys(usernme);

}

public void password(String pswd)

{

pwd.sendKeys(pswd);

}

@FindBy(xpath="//button[text()='Login']")

private WebElement loginbtn;

public void login()

{

loginbtn.click();

}

@FindBy(xpath="//a[text()='Application Form']")

private WebElement applform;

@FindBy(xpath="//a[text()='Admission ']")

private WebElement admision;

public void appformlink()

{

Actions actions = new Actions(driver);

actions.moveToElement(admision).click().perform();

applform.click();

}

@FindBy(id="candRegRegisterNumber")

private WebElement candregnumber;

@FindBy(id="candRegFirstName")

private WebElement candfrstname;

@FindBy(id="candRegLastName")

private WebElement candlastname;

@FindBy(id="candRegDob")

private WebElement canddob;

public void candinvalidfrstname()

{

candfrstname.clear();

}

public void candinvalidlastname()

{

candlastname.clear();

}

public void candinvaliddobname(Date birthdate)

{

SimpleDateFormat date1 = new SimpleDateFormat("dd-MM-yyyy");

String dob1=date1.format(birthdate);

canddob.sendKeys(dob1);

}

public String candno()

{

String regno=candregnumber.getText();

return regno;

}

public String candfrstname()

{

String candfrst=candfrstname.getText();

return candfrst;

}

public String candlstname()

{

String candlst=candlastname.getText();

return candlst;

}

public String canddob()

{

String birthday=canddob.getText();

return birthday;

}

@FindBy(id="candRegGender")

private WebElement candgender;

public void gender(String gender)

{

Select sel=new Select(candgender);

if(gender.equals("Female"))

{

sel.selectByVisibleText("Female");

}

else if(gender.equals("Male"))

{

sel.selectByVisibleText("Male");

}

else if (gender.equals("Select"))

{

sel.selectByVisibleText("Select");

}

}

@FindBy(id="candRegNationality")

private WebElement candnationality;

public void nationality(String nation) throws InterruptedException

{

Actions actions = new Actions(driver);

actions.moveToElement(candnationality).click().perform();

Select s1=new Select(candnationality);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement nat : options) {

if (nat.getText().equalsIgnoreCase(nation)) {

nat.click();

}

}

}

@FindBy(id="candRegMotherTounge")

private WebElement candmothertongue;

public void mothertongue(String lang) throws InterruptedException

{

Actions actions = new Actions(driver);

actions.moveToElement(candmothertongue).click().perform();

Select s1=new Select(candmothertongue);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement mothertongue : options) {

if (mothertongue.getText().equalsIgnoreCase(lang)) {

mothertongue.click();

}

}

}

@FindBy(id="candRegReligon")

private WebElement candreligion;

public void religion(String rel) throws InterruptedException

{

Actions actions = new Actions(driver);

actions.moveToElement(candreligion).click().perform();

Select s1=new Select(candreligion);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement relg : options) {

if (relg.getText().equalsIgnoreCase(rel)) {

relg.click();

}

}

}

@FindBy(id="candRegCasteCategory")

private WebElement candcastecat;

public void castecat(String castcat) throws InterruptedException

{

Actions actions = new Actions(driver);

actions.moveToElement(candcastecat).click().perform();

Select s1=new Select(candcastecat);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement cstcat : options) {

if (cstcat.getText().equalsIgnoreCase(castcat)) {

cstcat.click();

}

}

}

@FindBy(id="candRegCaste")

private WebElement candcaste;

public void caste(String cast) throws InterruptedException

{

Actions actions = new Actions(driver);

actions.moveToElement(candcaste).click().perform();

Select s1=new Select(candcaste);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement cst : options) {

if (cst.getText().equalsIgnoreCase(cast)) {

cst.click();

}

}

}

@FindBy(id="candRegOpenCasteName")

private WebElement candopencat;

public void opencat(String castecat, String caste) throws InterruptedException

{

if(castecat.equalsIgnoreCase("Open"))

{

candopencat.sendKeys(caste);

}

else

{

caste(caste);

}

}

@FindBy(id="nc\_yes")

private WebElement creamylayer;

@FindBy(id="nc\_no")

private WebElement noncreamy;

public void creamy(String castecat)

{

Actions actions = new Actions(driver);

if(castecat.equalsIgnoreCase("Yes"))

{

actions.moveToElement(creamylayer).click().perform();

}

else if(castecat.equalsIgnoreCase("No"))

{

actions.moveToElement(noncreamy).click().perform();

}

else

{

System.out.println("Open caste category");

}

}

@FindBy(id="candRegEmailId")

private WebElement candemail;

//@FindBy(id="AadhaarNumber")

//private WebElement aadharnum;

@FindBy(id="candRegPrimaryMobileNumber")

private WebElement candprimaryno;

@FindBy(id="candRegSecondaryMobileNumber")

private WebElement candsecndmobileno;

public String candemail()

{

String candmail=candemail.getText();

return candmail;

}

/\*public void aadharentry(String anum)

{

aadharnum.sendKeys(anum);

}\*/

public String candprmymno()

{

String candpmno=candprimaryno.getText();

return candpmno;

}

public String candsecndmno()

{

String candscndmno=candsecndmobileno.getText();

return candscndmno;

}

@FindBy(xpath="//button[text()=' Next Page']")

private WebElement nxtpagebtn1;

public void nextpage()

{

nxtpagebtn1.click();

}

@FindBy(partialLinkText="Logout")

private WebElement logout;

public void signout()

{

logout.click();

Alert a1=driver.switchTo().alert();

a1.accept();

}

}

Step2: package registration;

import java.util.List;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.Select;

public class Candidateprofilestep2 {

public WebDriver driver;

public Candidateprofilestep2(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver, this);

}

//Step2 candidate address details

@FindBy(id="candRegAddCountry")

private WebElement candcntry;

public void country(String nation) throws InterruptedException

{

Actions actions = new Actions(driver);

actions.moveToElement(candcntry).click().perform();

Select s1=new Select(candcntry);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement cntry : options) {

if (cntry.getText().equalsIgnoreCase(nation)) {

cntry.click();

}

}

}

@FindBy(id="candRegAddLineOne")

private WebElement candadd1;

public void add1(String add)

{

candadd1.sendKeys(add);

}

@FindBy(id="candRegAddLineTwo")

private WebElement candadd2;

public void add2(String add2)

{

candadd2.sendKeys(add2);

candadd2.sendKeys(Keys.TAB);

}

@FindBy(id="candRegAddState")

private WebElement candstate;

@FindBy(id="candRegAddDistrict")

private WebElement canddt;

@FindBy(xpath="//\*[@id='candRegAddTaluka']")

private WebElement candtaluka;

@FindBy(id="candRegAddCityVillage")

private WebElement candcity;

@FindBy(id="candRegPincode")

private WebElement candpincode;

public void statesel(String st,String dt,String tal,String city,String pin) throws InterruptedException

{

if(st.equalsIgnoreCase("Maharashtra"))

{

Select sel=new Select(candstate);

sel.selectByVisibleText("Maharashtra");

//district in MH

JavascriptExecutor je=(JavascriptExecutor)driver;

je.executeScript("window.scrollBy(0,250)", "");

Actions actions = new Actions(driver);

actions.moveToElement(canddt).click().perform();

Select s1=new Select(canddt);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement dst : options) {

if (dst.getText().equalsIgnoreCase(dt)) {

dst.click();

}

}

//Taluka in MH

Thread.sleep(5000);

Actions actions1 = new Actions(driver);

actions1.moveToElement(candtaluka).click().perform();

Select s2=new Select(candtaluka);

Thread.sleep(3000);

List<WebElement> options1 = s2.getOptions();

for (WebElement taluk : options1) {

if (taluk.getText().equalsIgnoreCase(tal)) {

taluk.click();

}

}

//City in MH

Thread.sleep(5000);

Actions actions2 = new Actions(driver);

actions2.moveToElement(candcity).click().perform();

Select s3=new Select(candcity);

Thread.sleep(3000);

List<WebElement> options2 = s3.getOptions();

for (WebElement cty : options2) {

if (cty.getText().equalsIgnoreCase(city)) {

cty.click();

}

}

Thread.sleep(3000);

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

}

else

{

otherstates(st);

Thread.sleep(5000);

othcitydetails(dt,tal,city,pin);

}

}

public void otherstates(String st) throws InterruptedException

{

Actions actions = new Actions(driver);

actions.moveToElement(candstate).click().perform();

Select s1=new Select(candstate);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement state : options) {

if (state.getText().equalsIgnoreCase(st)) {

state.click();

}

}

}

@FindBy(id="candRegDistrictText")

private WebElement canddtoth;

@FindBy(xpath="//\*[@id='candRegTalukaText']")

private WebElement candtalukaoth;

@FindBy(id="candRegCityVillageText")

private WebElement candcityoth;

@FindBy(id="candRegPincode")

private WebElement candpincodeoth;

public void othcitydetails(String dist,String taluk,String city,String pin)

{

JavascriptExecutor je=(JavascriptExecutor)driver;

je.executeScript("window.scrollBy(0,250)", "");

canddtoth.sendKeys(dist);

candtalukaoth.sendKeys(taluk);

candcityoth.sendKeys(city);

candpincodeoth.sendKeys(pin);

}

@FindBy(xpath="//button[text()=' Next Page']")

private WebElement nxtpagebtn2;

public void nextpage2()

{

nxtpagebtn2.click();

}

}

Step3: package registration;

import java.util.List;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.Select;

public class Candidateprofilestep3 {

public WebDriver driver;

public Candidateprofilestep3(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver, this);

}

//Step3 candidate parents details

@FindBy(id="orp\_no")

private WebElement orpno;

@FindBy(id="orp\_yes")

private WebElement orpyes;

@FindBy(id="guardNameofOrpanage")

private WebElement orpname;

@FindBy(id="orphanState")

private WebElement orpstate;

@FindBy(id="orphanDistrict")

private WebElement orpdt;

@FindBy(id="orphanTaluka")

private WebElement orptaluka;

public void orphanno(String orpstatus) throws InterruptedException

{

if(orpstatus.equalsIgnoreCase("No"))

{

Actions actions = new Actions(driver);

actions.moveToElement(orpno).click().build().perform();

}

}

public void orphanyes(String orpstatus,String orporgname,String orphstate,String orpdst,String orptaluk) throws InterruptedException

{

if(orpstatus.equalsIgnoreCase("Yes"))

{

Thread.sleep(3000);

Actions actions = new Actions(driver);

actions.moveToElement(orpyes).click().perform();

Thread.sleep(3000);

orpname.click();

orpname.sendKeys(orporgname);

//Orphan state

Actions actions1 = new Actions(driver);

actions1.moveToElement(orpstate).click().perform();

Select s1=new Select(orpstate);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement orpst : options) {

if (orpst.getText().equalsIgnoreCase(orphstate)) {

orpst.click();

}

}

//Orphan District

Thread.sleep(2000);

Actions actions2 = new Actions(driver);

actions2.moveToElement(orpdt).click().perform();

Select s2=new Select(orpdt);

Thread.sleep(3000);

List<WebElement> options1 = s2.getOptions();

for (WebElement orpdt : options1) {

if (orpdt.getText().equalsIgnoreCase(orpdst)) {

orpdt.click();

}

}

//Orphan taluka

Thread.sleep(2000);

Actions actions3 = new Actions(driver);

actions3.moveToElement(orptaluka).click().perform();

Select s3=new Select(orptaluka);

Thread.sleep(3000);

List<WebElement> options2 = s3.getOptions();

for (WebElement orptal : options2) {

if (orptal.getText().equalsIgnoreCase(orptaluk)) {

orptal.click();

}

}

}

}

@FindBy(id="guardfatherFirstName")

private WebElement fatherfrstname;

@FindBy(id="guardmotherFirstName")

private WebElement motherfrstname;

public void fatmatfrstname(String ffname,String mfname)

{

fatherfrstname.click();

fatherfrstname.sendKeys(ffname);

motherfrstname.click();

motherfrstname.sendKeys(mfname);

}

@FindBy(id="guardfatherLastName")

private WebElement fatherlstname;

@FindBy(id="guardmotherLastName")

private WebElement motherlstname;

public void fatmatlastname(String flname,String mlname)

{

fatherlstname.click();

fatherlstname.sendKeys(flname);

motherlstname.click();

motherlstname.sendKeys(mlname);

}

@FindBy(id="FatherGuardianOccupation")

private WebElement fatheroccp;

public void fatheroccp(String occp) throws InterruptedException

{

//fatheroccp.click();

//father occupation

Actions actions = new Actions(driver);

actions.moveToElement(fatheroccp).click().perform();

Select s1=new Select(fatheroccp);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement fatherocp : options) {

if (fatherocp.getText().equalsIgnoreCase(occp)) {

fatherocp.click();

}

}

}

@FindBy(id="MotherGuardianOccupation")

private WebElement motheroccp;

public void motheroccp(String occp) throws InterruptedException

{

//motheroccp.click();

//father occupation

Actions actions = new Actions(driver);

actions.moveToElement(motheroccp).click().perform();

Select s1=new Select(motheroccp);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement motherocp : options) {

if (motherocp.getText().equalsIgnoreCase(occp)) {

motherocp.click();

}

}

}

@FindBy(id="FatherGuardianWorkingState")

private WebElement fatheroccpstate;

@FindBy(id="FatherGuardianWorkingDistrict")

private WebElement fatheroccpdt;

@FindBy(id="FatherGuardianWorkingTaluka")

private WebElement fatheroccptaluka;

@FindBy(id="FatherGuardianIncome")

private WebElement fatherincome;

@FindBy(id="FatherGuardianDomicileState")

private WebElement fatherdomstate;

@FindBy(id="FatherGuardianDomicileDistrict")

private WebElement fatherdomdt;

@FindBy(id="FatherGuardianDomicileTaluka")

private WebElement fatherdomtaluka;

public void fatherocpdetails(String st,String dt,String tal,String income,String domstate,String domdt,String domtal) throws InterruptedException

{

//Father occupation state

//fatheroccpstate.click();

Thread.sleep(3000);

Actions actions = new Actions(driver);

actions.moveToElement(fatheroccpstate).click().perform();

Select s1=new Select(fatheroccpstate);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement state : options) {

if (state.getText().equalsIgnoreCase(st)) {

state.click();

}

}

//Father occupation DISTRICT

//fatheroccpdt.click();

Thread.sleep(2000);

JavascriptExecutor je=(JavascriptExecutor)driver;

je.executeScript("window.scrollBy(0,250)", "");

Actions actions1 = new Actions(driver);

actions1.moveToElement(fatheroccpdt).click().perform();

Select s2=new Select(fatheroccpdt);

Thread.sleep(3000);

List<WebElement> options1 = s2.getOptions();

for (WebElement dst : options1) {

if (dst.getText().equalsIgnoreCase(dt)) {

dst.click();

}

}

//Father occupation Taluka

Thread.sleep(3000);

//fatheroccptaluka.click();

Actions actions2 = new Actions(driver);

actions2.moveToElement(fatheroccptaluka).click().perform();

Select s3=new Select(fatheroccptaluka);

Thread.sleep(3000);

List<WebElement> options2 = s3.getOptions();

for (WebElement taluk : options2) {

if (taluk.getText().equalsIgnoreCase(tal)) {

taluk.click();

}

}

//Father income

Thread.sleep(3000);

//fatherincome.click();

Actions actions3 = new Actions(driver);

actions3.moveToElement(fatherincome).click().perform();

Select s4=new Select(fatherincome);

Thread.sleep(3000);

List<WebElement> options3 = s4.getOptions();

for (WebElement fatincome : options3) {

if (fatincome.getText().equalsIgnoreCase(income)) {

fatincome.click();

}

}

//Father domicile state

Thread.sleep(3000);

// fatherdomstate.click();

Actions actions4 = new Actions(driver);

actions4.moveToElement(fatherdomstate).click().perform();

Select s5=new Select(fatherdomstate);

Thread.sleep(3000);

List<WebElement> options4 = s5.getOptions();

for (WebElement domst : options4) {

if (domst.getText().equalsIgnoreCase(domstate)) {

domst.click();

}

}

//Father district district

Thread.sleep(3000);

//fatherdomdt.click();

JavascriptExecutor je1=(JavascriptExecutor)driver;

je1.executeScript("window.scrollBy(0,250)", "");

Actions actions5 = new Actions(driver);

actions5.moveToElement(fatherdomdt).click().perform();

Select s6=new Select(fatherdomdt);

Thread.sleep(3000);

List<WebElement> options5 = s6.getOptions();

for (WebElement domdst : options5) {

if (domdst.getText().equalsIgnoreCase(domdt)) {

domdst.click();

}

}

//Father domicile taluka

Thread.sleep(3000);

//fatherdomtaluka.click();

Actions actions6 = new Actions(driver);

actions6.moveToElement(fatherdomtaluka).click().perform();

Select s7=new Select(fatherdomtaluka);

Thread.sleep(3000);

List<WebElement> options6 = s7.getOptions();

for (WebElement domtaluk: options6) {

if (domtaluk.getText().equalsIgnoreCase(domtal)) {

domtaluk.click();

}

}

}

@FindBy(id="MotherGuardianWorkingState")

private WebElement motheroccpstate;

@FindBy(id="MotherGuardianWorkingDistrict")

private WebElement motheroccpdt;

@FindBy(id="MotherGuardianWorkingTaluka")

private WebElement motheroccptaluka;

@FindBy(id="MotherGuardianIncome")

private WebElement motherincome;

@FindBy(id="MotherGuardianDomicileState")

private WebElement motherdomstate;

@FindBy(id="MotherGuardianDomicileDistrict")

private WebElement motherdomdt;

@FindBy(id="MotherGuardianDomicileTaluka")

private WebElement motherdomtaluka;

public void motherocpdetails(String st,String dt,String tal,String income,String domstate,String domdt,String domtal) throws InterruptedException

{

//Mother occupation state

//motheroccpstate.click();

Thread.sleep(3000);

Actions actions = new Actions(driver);

actions.moveToElement(motheroccpstate).click().perform();

Select s1=new Select(motheroccpstate);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement state : options) {

if (state.getText().equalsIgnoreCase(st)) {

state.click();

}

}

//Mother occupation DISTRICT

//motheroccpdt.click();

Thread.sleep(3000);

Actions actions1 = new Actions(driver);

actions1.moveToElement(motheroccpdt).click().perform();

Select s2=new Select(motheroccpdt);

Thread.sleep(3000);

List<WebElement> options1 = s2.getOptions();

for (WebElement dst : options1) {

if (dst.getText().equalsIgnoreCase(dt)) {

dst.click();

}

}

//Mother occupation Taluka

Thread.sleep(3000);

//motheroccptaluka.click();

Actions actions2 = new Actions(driver);

actions2.moveToElement(motheroccptaluka).click().perform();

Select s3=new Select(motheroccptaluka);

Thread.sleep(3000);

List<WebElement> options2 = s3.getOptions();

for (WebElement taluk : options2) {

if (taluk.getText().equalsIgnoreCase(tal)) {

taluk.click();

}

}

//Mother income

Thread.sleep(3000);

//motherincome.click();

Actions actions3 = new Actions(driver);

actions3.moveToElement(motherincome).click().perform();

Select s4=new Select(motherincome);

Thread.sleep(3000);

List<WebElement> options3 = s4.getOptions();

for (WebElement matincome : options3) {

if (matincome.getText().equalsIgnoreCase(income)) {

matincome.click();

}

}

//Mother domicile state

Thread.sleep(3000);

// motherdomstate.click();

Actions actions4 = new Actions(driver);

actions4.moveToElement(motherdomstate).click().perform();

Select s5=new Select(motherdomstate);

Thread.sleep(3000);

List<WebElement> options4 = s5.getOptions();

for (WebElement domst : options4) {

if (domst.getText().equalsIgnoreCase(domstate)) {

domst.click();

}

}

//Mother domicile district

Thread.sleep(3000);

//motherdomdt.click();

Actions actions5 = new Actions(driver);

actions5.moveToElement(motherdomdt).click().perform();

Select s6=new Select(motherdomdt);

Thread.sleep(3000);

List<WebElement> options5 = s6.getOptions();

for (WebElement domddst : options5) {

if (domddst.getText().equalsIgnoreCase(domdt)) {

domddst.click();

}

}

//Mother domicile taluka

Thread.sleep(3000);

// motherdomtaluka.click();

Actions actions6 = new Actions(driver);

actions6.moveToElement(motherdomtaluka).click().perform();

Select s7=new Select(motherdomtaluka);

Thread.sleep(3000);

List<WebElement> options6 = s7.getOptions();

for (WebElement domtaluk: options6) {

if (domtaluk.getText().equalsIgnoreCase(domtal)) {

domtaluk.click();

}

}

}

@FindBy(xpath="//button[text()=' Next Page']")

private WebElement nxtpagebtn3;

public void nextpage3()

{

nxtpagebtn3.click();

}

}

Step4: package registration;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.Select;

public class Candidateprofilestep4

{

public WebDriver driver;

public Candidateprofilestep4(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver, this);

}

@FindBy(id="deno")

private WebElement defno;

@FindBy(id="deyes")

private WebElement defyes;

public void defence(String defstatus,String defcat1) throws InterruptedException

{

Actions actions = new Actions(driver);

if(defstatus.equalsIgnoreCase("Yes"))

{

//JavascriptExecutor jse = (JavascriptExecutor) driver;

//jse.executeScript("document.getElementById('deyes').value = 'yes';");

actions.moveToElement(defyes).click().perform();

Thread.sleep(2000);

defencecat(defcat1);

}

else if(defstatus.equalsIgnoreCase("No"))

{

//JavascriptExecutor jse = (JavascriptExecutor) driver;

//jse.executeScript("document.getElementById('deno').value = 'No';");

actions.moveToElement(defyes).click().perform();

actions.moveToElement(defno).click().perform();

}

}

@FindBy(className="active-result")

private WebElement defcat;

public void defencecat(String defcat1) throws InterruptedException

{

Actions actions = new Actions(driver);

WebElement ele= driver.findElement(By.className("chosen-single"));

actions.moveToElement(ele).click().perform();

Thread.sleep(3000);

List<WebElement> options = driver.findElements(By.className("active-result"));

for (WebElement defcat : options) {

if (defcat.getText().equalsIgnoreCase(defcat1)) {

defcat.click();

}

}

}

@FindBy(id="disno")

private WebElement nodisability;

@FindBy(id="disyes")

private WebElement yesdisability;

public void disablestat(String disstatus,double disprcnt,String discat1) throws InterruptedException

{

Actions actions = new Actions(driver);

if(disstatus.equalsIgnoreCase("Yes"))

{

//JavascriptExecutor jse = (JavascriptExecutor) driver;

//jse.executeScript("document.getElementById('disyes').value = 'yes';");

actions.moveToElement(yesdisability).click().perform();

Thread.sleep(5000);

discat(disprcnt,discat1);

}

else if(disstatus.equalsIgnoreCase("No"))

{

//JavascriptExecutor jse = (JavascriptExecutor) driver;

//jse.executeScript("document.getElementById('disno').value = 'No';");

actions.moveToElement(nodisability).click().perform();

}

}

@FindBy(id="profileDisabiltyPercenetage")

private WebElement disablepercentage;

@FindBy(xpath="active-result")

private WebElement disablecatgry;

public void discat(double disprcnt,String discat1) throws InterruptedException

{

Double d=new Double(disprcnt);

int disprcntval=d.intValue();

String disper=Integer.toString(disprcntval);

disablepercentage.sendKeys(disper);

Actions actions = new Actions(driver);

WebElement ele= driver.findElement(By.xpath("(//a[@class='chosen-single'])[2]"));

actions.moveToElement(ele).click().perform();

Thread.sleep(3000);

List<WebElement> options = driver.findElements(By.className("active-result"));

for (WebElement discat : options) {

if (discat.getText().equalsIgnoreCase(discat1)) {

discat.click();

}

}

}

@FindBy(id="interno")

private WebElement drawingfail;

@FindBy(id="interyes")

private WebElement drawingpass;

public void drawstat(String drawstatus)

{

JavascriptExecutor js=(JavascriptExecutor)driver;

js.executeScript("window.scrollBy(0,250)", "");

Actions actions = new Actions(driver);

if(drawstatus.equalsIgnoreCase("Yes"))

{

actions.moveToElement(drawingpass).click().perform();

}

else if(drawstatus.equalsIgnoreCase("No"))

{

actions.moveToElement(drawingfail).click().perform();

}

}

@FindBy(id="otherCurricular")

private WebElement othcircular;

public void xtracircular(String xtracir) throws InterruptedException

{

//othcircular.click();

JavascriptExecutor je=(JavascriptExecutor)driver;

je.executeScript("window.scrollBy(0,250)", "");

Actions actions = new Actions(driver);

actions.moveToElement(othcircular).click().perform();

Select s1=new Select(othcircular);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement othcir : options) {

if (othcir.getText().equalsIgnoreCase(xtracir)) {

othcir.click();

}

}

}

@FindBy(id="sports\_no")

private WebElement nosports;

@FindBy(id="sports\_yes")

private WebElement insports;

public void sportsstat(String sportstatus,String slevel) throws InterruptedException

{

Actions actions = new Actions(driver);

if(sportstatus.equalsIgnoreCase("Yes"))

{

actions.moveToElement(insports).click().perform();

Thread.sleep(3000);

sportlevel(slevel);

}

else if(sportstatus.equalsIgnoreCase("No"))

{

actions.moveToElement(nosports).click().perform();

}

}

@FindBy(id="otherSportsLevel")

private WebElement sportlevel;

public void sportlevel(String slevel) throws InterruptedException

{

//sportlevel.click();

JavascriptExecutor je=(JavascriptExecutor)driver;

je.executeScript("window.scrollBy(0,250)", "");

Actions actions = new Actions(driver);

actions.moveToElement(sportlevel).click().perform();

Select s1=new Select(sportlevel);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement sportlevel : options) {

if (sportlevel.getText().equalsIgnoreCase(slevel)) {

sportlevel.click();

}

}

}

//@FindBy(id="/html/body/div[1]/div/div/div/div/div/div[2]/div/div/form/div/div[2]/div/div/div[2]/div/div[3]/div[2]/div/div[4]/div/div/div[1]/label")

//private WebElement blosterno;

@FindBy(xpath="/html/body/div[1]/div/div/div/div/div/div[2]/div/div/form/div/div[2]/div/div/div[2]/div/div[3]/div[2]/div/div[4]/div/div/div[1]/label")

private WebElement blosterno;

@FindBy(xpath="/html/body/div[1]/div/div/div/div/div/div[2]/div/div/form/div/div[2]/div/div/div[2]/div/div[3]/div[2]/div/div[4]/div/div/div[2]/label")

private WebElement blosteryes;

public void blosterstat(String blosterstatus,String bdt,String btal) throws InterruptedException

{

Thread.sleep(2000);

Actions actions = new Actions(driver);

if(blosterstatus.equalsIgnoreCase("Yes"))

{

JavascriptExecutor je=(JavascriptExecutor)driver;

je.executeScript("window.scrollBy(0,250)", "");

//actions.moveToElement(blosteryes).click().perform();

Thread.sleep(5000);

blosteryes.click();

blosterdttal(bdt,btal);

}

else if(blosterstatus.equalsIgnoreCase("No"))

{

//actions.moveToElement(blosterno).click().perform();

blosterno.click();

}

}

@FindBy(id="otherBolsterDistrict")

private WebElement blosterdt;

@FindBy(id="otherBolsterTaluka")

private WebElement blostertaluka;

public void blosterdttal(String bdt,String btal) throws InterruptedException

{

//Bloster district

//blosterdt.click();

Actions actions = new Actions(driver);

actions.moveToElement(blosterdt).click().perform();

Select s1=new Select(blosterdt);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement blosdt : options) {

if (blosdt.getText().equalsIgnoreCase(bdt)) {

blosdt.click();

}

}

//Bloster district

Thread.sleep(2000);

//blostertaluka.click();

Actions actions1 = new Actions(driver);

actions1.moveToElement(blostertaluka).click().perform();

Select s2=new Select(blostertaluka);

Thread.sleep(3000);

List<WebElement> options1 = s2.getOptions();

for (WebElement blostal : options1) {

if (blostal.getText().equalsIgnoreCase(btal)) {

blostal.click();

}

}

}

@FindBy(xpath="/html/body/div[1]/div/div/div/div/div/div[2]/div/div/form/div/div[2]/div/div/div[2]/div/div[4]/div/div/div/button[2]")

private WebElement nxtpagebtn4;

public void nextpage4()

{

nxtpagebtn4.click();

}

}

Step5: package registration;

import java.util.List;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.Select;

public class Candidateprofilestep5 {

public WebDriver driver;

public Candidateprofilestep5(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver, this);

}

@FindBy(id="schloc\_maha")

private WebElement MHschool;

@FindBy(id="tenthSchoolDistrict")

private WebElement MHschooldt;

@FindBy(id="tenthSchoolTaluka")

private WebElement MHschooltaluka;

@FindBy(id="tenthSchoolVillage")

private WebElement MHschoolcity;

@FindBy(id="schloc\_oth")

private WebElement otherthanMH;

@FindBy(id="schloc\_outs")

private WebElement nri;

public void schoolsel(String sklsel,String dt,String tal,String city) throws InterruptedException

{

if(sklsel.equalsIgnoreCase("Maharashtra"))

{

MHschool.click();

//district in MH

//MHschooldt.click();

Actions actions = new Actions(driver);

actions.moveToElement(MHschooldt).click().perform();

Select s1=new Select(MHschooldt);

Thread.sleep(3000);

List<WebElement> options = s1.getOptions();

for (WebElement dst : options) {

if (dst.getText().equalsIgnoreCase(dt)) {

dst.click();

}

}

//Taluka in MH

Thread.sleep(2000);

//MHschooltaluka.click();

Actions actions1 = new Actions(driver);

actions1.moveToElement(MHschooltaluka).click().perform();

Select s2=new Select(MHschooltaluka);

Thread.sleep(3000);

List<WebElement> options1 = s2.getOptions();

for (WebElement taluka : options1) {

if (taluka.getText().equalsIgnoreCase(tal)) {

taluka.click();

}

}

//city in MH

Thread.sleep(2000);

//MHschoolcity.click();

Actions actions2 = new Actions(driver);

actions2.moveToElement(MHschoolcity).click().perform();

Select s3=new Select(MHschoolcity);

Thread.sleep(3000);

List<WebElement> options2 = s3.getOptions();

for (WebElement cty : options2) {

if (cty.getText().equalsIgnoreCase(city)) {

cty.click();

}

}

}

else if (sklsel.equalsIgnoreCase("Other than Maharashtra "))

{

otherthanMH.click();

}

else if (sklsel.equalsIgnoreCase("Outside India (NRI Candidate)"))

{

nri.click();

}

}

@FindBy(id="10passfail\_no")

private WebElement tenthfail;

@FindBy(id="10passfail\_yes")

private WebElement tenthpass;

public void tenthstat(String tenthstatus)

{

Actions actions = new Actions(driver);

if(tenthstatus.equalsIgnoreCase("Yes"))

{

actions.moveToElement(tenthpass).click().perform();

}

else if(tenthstatus.equalsIgnoreCase("No"))

{

actions.moveToElement(tenthfail).click().perform();

}

}

@FindBy(id="techsub\_no")

private WebElement techsubno;

@FindBy(id="techsub\_yes")

private WebElement techsubyes;

public void techsub(String techstat)

{

Actions actions = new Actions(driver);

if(techstat.equalsIgnoreCase("Yes"))

{

actions.moveToElement(techsubyes).click().perform();

}

else if(techstat.equalsIgnoreCase("No"))

{

actions.moveToElement(techsubno).click().perform();

}

}

@FindBy(id="natschool\_no")

private WebElement nonatskl;

@FindBy(id="natschool\_yes")

private WebElement yesnatskl;

public void nationalskl(String natskl)

{

Actions actions = new Actions(driver);

if(natskl.equalsIgnoreCase("Yes"))

{

actions.moveToElement(yesnatskl).click().perform();

}

else if(natskl.equalsIgnoreCase("No"))

{

actions.moveToElement(nonatskl).click().perform();

}

}

@FindBy(id="bhschool\_yes")

private WebElement bhsklyes;

@FindBy(id="bhschool\_no")

private WebElement bhsklno;

public void bhschool(String bhskl)

{

Actions actions = new Actions(driver);

if(bhskl.equalsIgnoreCase("Yes"))

{

actions.moveToElement(bhsklyes).click().perform();

}

else if(bhskl.equalsIgnoreCase("No"))

{

actions.moveToElement(bhsklno).click().perform();

}

}

//Marathi/ Regional / First Language

@FindBy(id="tenthTotalFirstLanguage")

private WebElement totfrstlang;

@FindBy(id="tenthObtainedFirstLanguage")

private WebElement securedfrstlang;

public void firstlang(double frst,double frstobt)

{

String frstmark=Double.toString(frst);

String frstobtained=Double.toString(frstobt);

totfrstlang.sendKeys(frstmark);

securedfrstlang.sendKeys(frstobtained);

}

//Hindi/ Sanskrit / Second Language

@FindBy(id="tenthTotalSecondLanguage")

private WebElement totseclang;

@FindBy(id="tenthObtainedSecondLanguage")

private WebElement securedseclang;

public void seclang(double scnd,double scndobt)

{

String secmark=Double.toString(scnd);

String secobtained=Double.toString(scndobt);

totseclang.sendKeys(secmark);

securedseclang.sendKeys(secobtained);

}

//English/ Other / Third Language

@FindBy(id="tenthTotalThirdLanguage")

private WebElement totthirdlang;

@FindBy(id="tenthObtainedThirdLanguage")

private WebElement securedthirdlang;

public void thrdlang(double thrd,double thrdobt)

{

String thrdmark=Double.toString(thrd);

String thrdobtained=Double.toString(thrdobt);

totthirdlang.sendKeys(thrdmark);

securedthirdlang.sendKeys(thrdobtained);

}

//Math details

@FindBy(id="tenthTotalMaths")

private WebElement totmath;

@FindBy(id="tenthObtainedMaths")

private WebElement securedmath;

public void math(double mth,double mathobt)

{

String mathmark=Double.toString(mth);

String mathobtained=Double.toString(mathobt);

totmath.sendKeys(mathmark);

securedmath.sendKeys(mathobtained);

}

//Science

@FindBy(id="tenthTotalScience")

private WebElement totscience;

@FindBy(id="tenthObtainedScience")

private WebElement securedscience;

public void science(double sci,double sciobt)

{

String scimark=Double.toString(sci);

String sciobtained=Double.toString(sciobt);

totscience.sendKeys(scimark);

securedscience.sendKeys(sciobtained);

}

//Social Science (Geography & History)

@FindBy(id="tenthTotalSocial")

private WebElement totsocial;

@FindBy(id="tenthObtainedSocial")

private WebElement securedsocial;

public void social(double soc,double socobt)

{

String socmark=Double.toString(soc);

String socobtained=Double.toString(socobt);

totsocial.sendKeys(socmark);

securedsocial.sendKeys(socobtained);

}

//Technical Subject

@FindBy(id="tenthTotalTechnical")

private WebElement tottech;

@FindBy(id="tenthObtainedTechnical")

private WebElement securedtech;

public void tech(double tech,double techobt) throws InterruptedException

{

String tchmark=Double.toString(tech);

String tchobtained=Double.toString(techobt);

tottech.sendKeys(tchmark);

Thread.sleep(2000);

securedtech.sendKeys(tchobtained);

}

@FindBy(id="tenthTotalMarks")

private WebElement tottenth;

@FindBy(id="tenthObtainedTotalMarks")

private WebElement securedtenth;

@FindBy(id="SystemTotalTenth")

private WebElement systottenth;

@FindBy(id="SystemObtainedTenth")

private WebElement syssecuredtenth;

public void tenthtot()

{

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

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

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

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

}

//ADDITIONAL WEIGHTAGE

@FindBy(id="additionalDrawing")

private WebElement adddrawing;

@FindBy(id="additionalExtraCurricullar")

private WebElement adddextracircular;

@FindBy(id="additionalSports")

private WebElement addsports;

@FindBy(id="additionalBolsterCandidate")

private WebElement addbolstercand;

@FindBy(id="additionalOrphanage")

private WebElement addorphanage;

@FindBy(id="additionalRuralArea")

private WebElement addruralarea;

@FindBy(id="additionalTotalAdditionalweightage")

private WebElement addtotweightage;

@FindBy(id="additionalWeightage")

private WebElement addweightage;

@FindBy(id="additionalTotalMarksWeightage")

private WebElement addtotmarksweightage;

public void additionaltot()

{

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

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

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

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

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

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

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

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

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

}

//TOTAL MARKS FOR ITI ADMISSION MERIT LIST

@FindBy(id="additionalTotalMarks")

private WebElement addtotmarks;

@FindBy(id="additionalTotalWeightage")

private WebElement addtotalweightage;

@FindBy(id="additionalTotalPercent")

private WebElement addtotpercentage;

public void totmarksitimeritlist()

{

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

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

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

}

@FindBy(xpath="//button[text()=' Next Page']")

private WebElement nxtpagebtn5;

public void nextpage5()

{

nxtpagebtn5.click();

}

}

Step6: package registration;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.Select;

public class Candidateprofilestep6 {

public WebDriver driver;

public Candidateprofilestep6(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver, this);

}

@FindBy(id="previti\_yes")

private WebElement previtiyes;

@FindBy(id="previti\_no")

private WebElement previtino;

@FindBy(id="itiYearofAdmission")

private WebElement adminyear;

@FindBy(id="itiYearofPassing")

private WebElement yearpassing;

public void prevtrain(String transtat,int adminyear1,int passyear,String itname,String ittrade) throws InterruptedException

{

Actions actions = new Actions(driver);

if(transtat.equalsIgnoreCase("Yes"))

{

actions.moveToElement(previtiyes).click().perform();

Thread.sleep(3000);

adminiti(adminyear1,passyear);

itdetails(itname,ittrade);

}

else if(transtat.equalsIgnoreCase("No"))

{

actions.moveToElement(previtino).click().perform();

}

}

public void adminiti(int adminyear1,int passyear) throws InterruptedException

{

adminyear.click();

Thread.sleep(4000);

Select sel=new Select(adminyear);

switch(adminyear1)

{

case 1950: sel.selectByVisibleText("1950");

break;

case 2014: sel.selectByVisibleText("2014");

break;

}

yearpassing.click();

Thread.sleep(4000);

Select sel1=new Select(yearpassing);

switch(passyear)

{

case 1952: sel1.selectByVisibleText("1952");

break;

case 2016: sel.selectByVisibleText("2016");

break;

}

}

@FindBy(id="itiNameofIti")

private WebElement itiname;

@FindBy(id="itiNameofTrade")

private WebElement ititrade;

public void itdetails(String itname,String ittrade)

{

itiname.sendKeys(itname);

ititrade.sendKeys(ittrade);

}

@FindBy(xpath="//button[text()=' Next Page']")

private WebElement nxtpagebtn6;

public void nextpage6()

{

nxtpagebtn6.click();

}

}

Step7: package registration;

import java.util.List;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.Select;

public class Candidateprofilestep7 {

public WebDriver driver;

public Candidateprofilestep7(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver, this);

}

@FindBy(id="HomeTaluka")

private WebElement hometaluk;

public void hometaluk(String hometal) throws InterruptedException

{

Actions actions1 = new Actions(driver);

actions1.moveToElement(hometaluk).click().perform();

Select s2=new Select(hometaluk);

Thread.sleep(3000);

List<WebElement> options1 = s2.getOptions();

for (WebElement hometaluk : options1) {

if (hometaluk.getText().equalsIgnoreCase(hometal)) {

hometaluk.click();

}

}

}

@FindBy(xpath="//\*[@id='accept']")

private WebElement acept;

public void accpt()

{

JavascriptExecutor je=(JavascriptExecutor)driver;

je.executeScript("window.scrollBy(0,250)", "");

Actions actions = new Actions(driver);

actions.moveToElement(acept).click().perform();

}

@FindBy(xpath="//button[text()=' Next Page ']")

private WebElement nxtpagebtn7;

public void nextpage7()

{

JavascriptExecutor je=(JavascriptExecutor)driver;

je.executeScript("window.scrollBy(0,150)", "");

nxtpagebtn7.click();

}

}

Step8: package registration;

import org.openqa.selenium.Alert;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class Candidateprofilestep8 {

public WebDriver driver;

public Candidateprofilestep8(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver, this);

}

@FindBy(xpath="//a[text()='Generate PDF & Save']")

private WebElement generatepdf;

public void pdf()

{

generatepdf.click();

}

@FindBy(xpath="//button[text()=' Previous Page ']")

private WebElement prevpage;

public void prepage()

{

prevpage.click();

}

@FindBy(partialLinkText="Logout")

private WebElement logout;

public void signout()

{

JavascriptExecutor je2=(JavascriptExecutor)driver;

je2.executeScript("window.scrollBy(250,0)", "");

logout.click();

Alert a1=driver.switchTo().alert();

a1.accept();

}

}