Web Application for displaying a list of applicants for a job vacancy

Document made by : UCHITHA L HEWTHANTHRIGE
Email : uchithalakmali@yahoo.com
GitHub link to the project : https://github.com/uchitha123/GOVJobsPortal.git

Table of Content

Table of Figures

Introduction

This is the document contain about a project Candidate Web Application which has given as an interview assessment task for applied junior software developer role in Home office.

How to Build and run the application

Software requirements

1. Java - Java - jdk1.8.0_65 – (http://www.oracle.com/technetwork/systems/index-jsp-138363.html)

You can download Java SE development kit using above link. I have used for application development windows x64 Operating System compatible Java development kit.

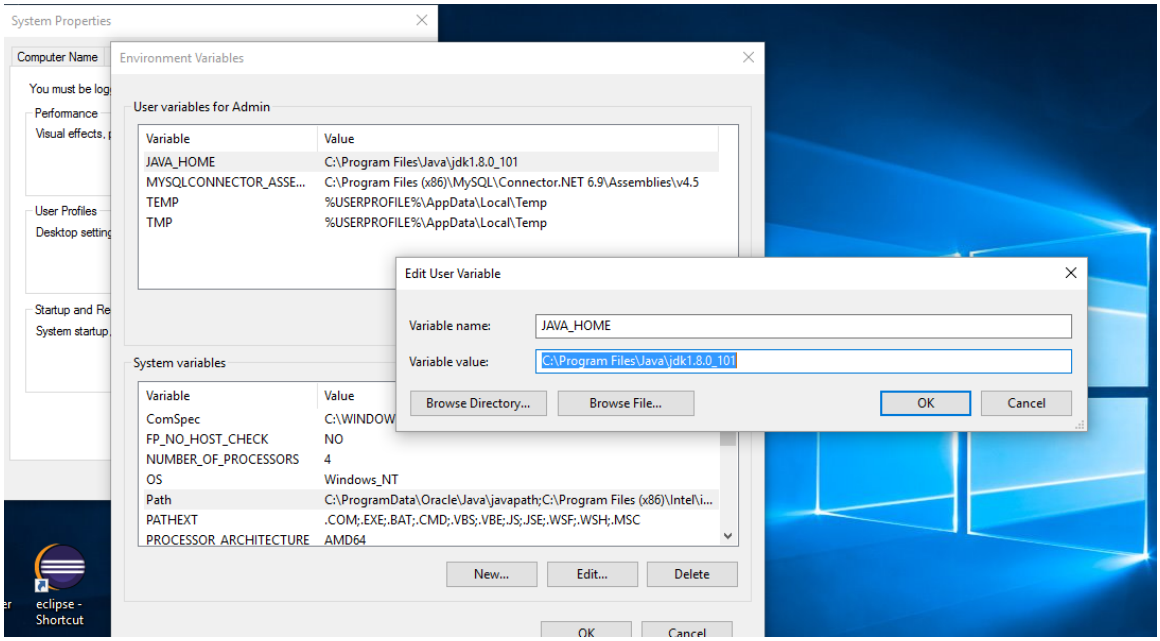

Figure 1: Set JAVA_HOME environment variable to the java installation

After typing java –version command in your command prompt you can see installed version in your machine and confirm the Java is working fine in your machine.

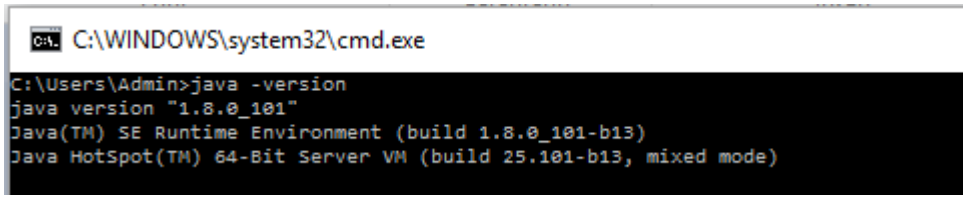

Figure 2: Checking installed java version

2. Maven - apache-maven-3.3.9 - (https://maven.apache.org/download.cgi)

You can download Maven using above link. Unzip the distributed archive Ex: apache-maven-3.3.3-bin.bin.zip and copy into your C: drive.

I have installed in my machine C:\Program Files\maven\apache-maven-3.3.9. Make sure you have added C:\Program Files\maven\apache-maven-3.3.9\bin as your environment variable to your machine.

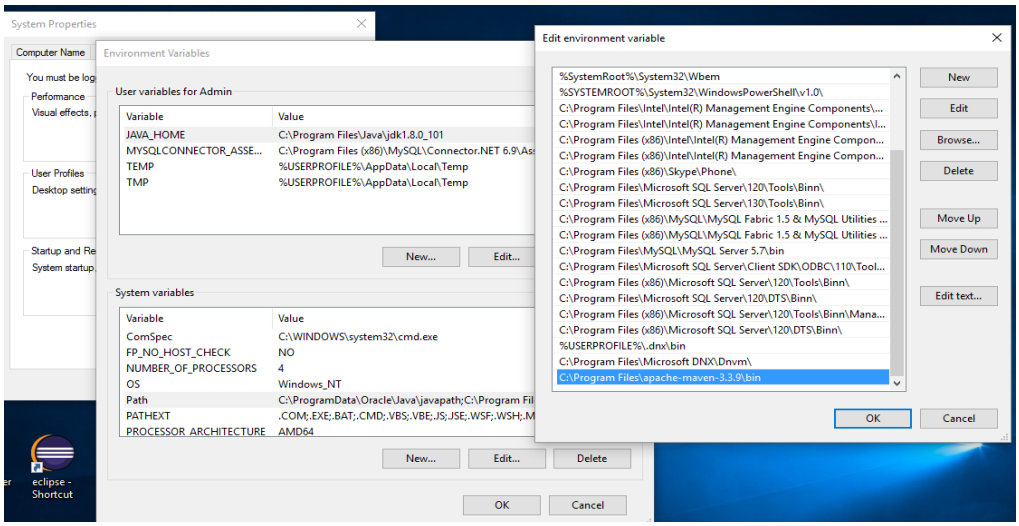

Figure 3: Add C:\Program Files\maven\apache-maven-3.3. to path system variable

Type mvn --version to see your installed version and confirm the maven is running on your machine fine.

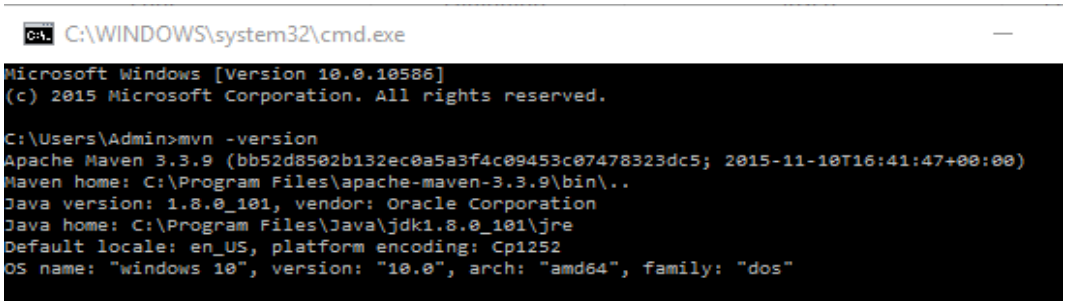

Figure 4: Checking installed Maven version

3. Eclipse IDE for Java EE Developers - Eclipse Jee Neon – (http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/neonr)

I have used "Eclipse IDE for Java EE Developer" to my machine. Please follow this link for Eclipse Troubleshooting: https://wiki.eclipse.org/Eclipse/Installation

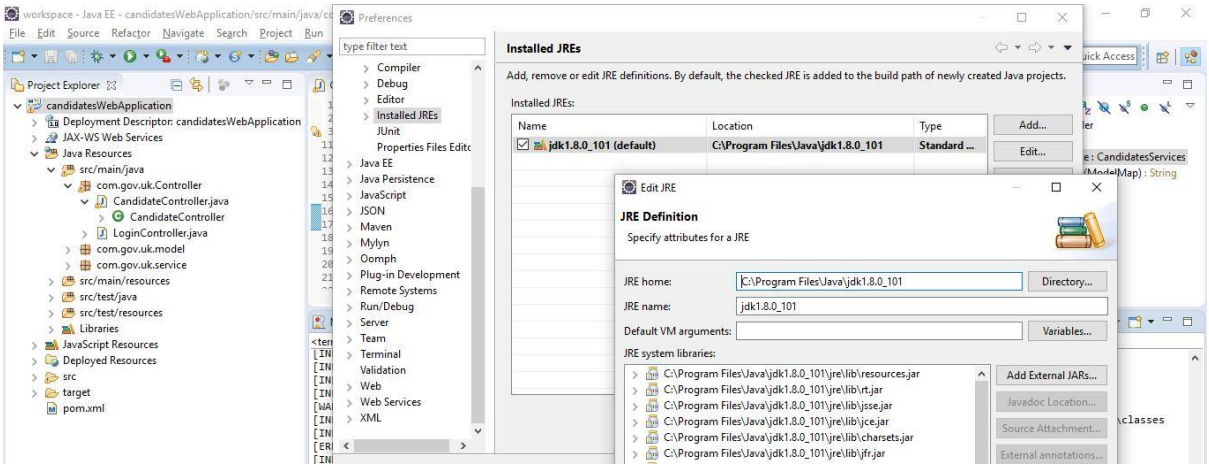

Figure 5: Setup JREs path to Eclipse IDE

Before install the **Candidate Web Application** make sure above software requirements satisfied in your machine.

Steps to open and run the application using Eclipse

Step 1: open your Eclipse IDE and import the project into working environment

File → Import → select maven project type from the **select an import wizard ->** from maven project type select Existing maven Projects → select next → browse the root Directory (CandidatesWebApplication) and select the existing pom.xml file form project wizard →select finish button

Step 2: How to setup and run the CandidateWebApplication

After importing project to Eclipse working environment you can see project structure in Project Explorer wizard. You can find sample image below.

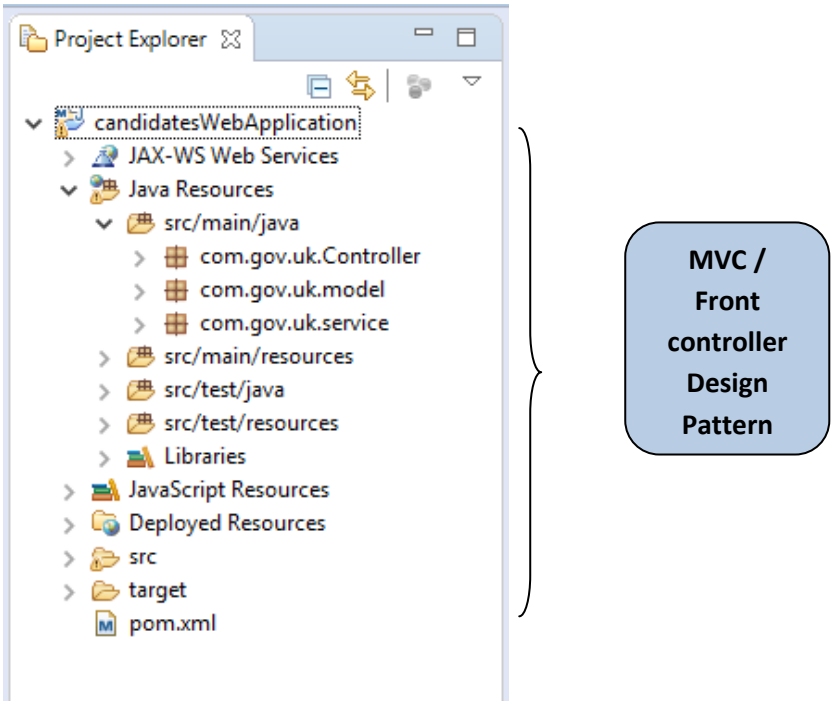

Figure 6: Candidates Web Application Project Wizard

Before you run the project make sure you have set up JREs path into JDK directory path.

Windows → Preferences → Installed JREs

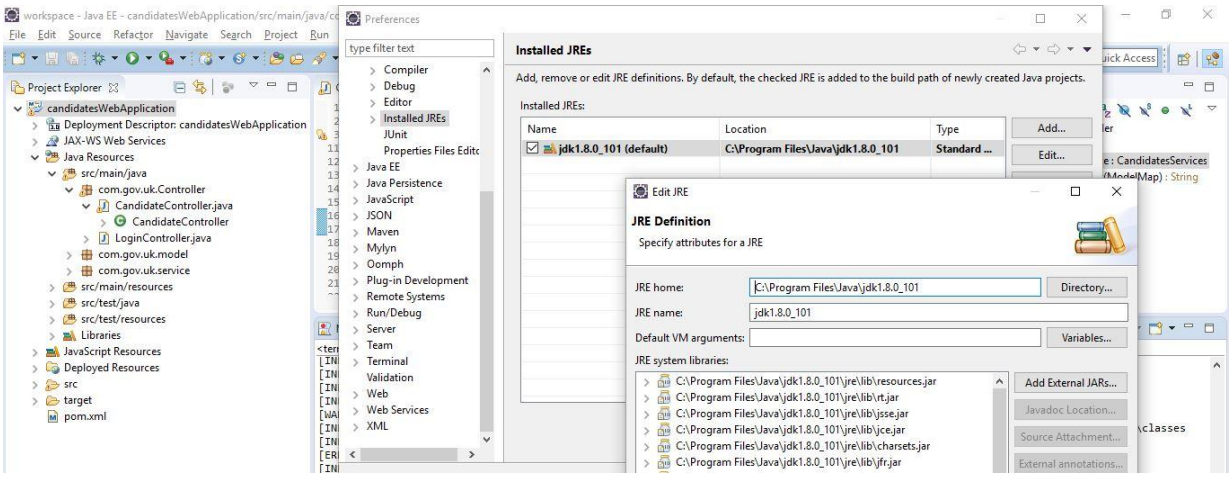

Figure 7: setup JREs path

Select the CandidatesWebApplication project → right click → in pop up wizard select Maven → update project →OK

Again Select the CandidateWebApplication project→ right click→ in pop up wizard select Run As → Maven build.. →Edit configuration and launch wizard in Goal text box type **tomcat7:run**

In console wizard can see project is build and run successfully.

In browser type - **localhost:8080/login**

Step 3: Login to CandidateWebApplication and navigate to pages

In login page, type below user name and password.

Username: admin
Password: admin123
After validating username and password, you will navigate to job list page.

Once you click the one vacancy, it will navigate to display List of candidates applied for that specific role.

Once you click the name of the candidate you can see more details about the candidates including summary of the work experience.

Site Map

The below diagram is visualized Sitemap for the CandidatesWebApplication website.

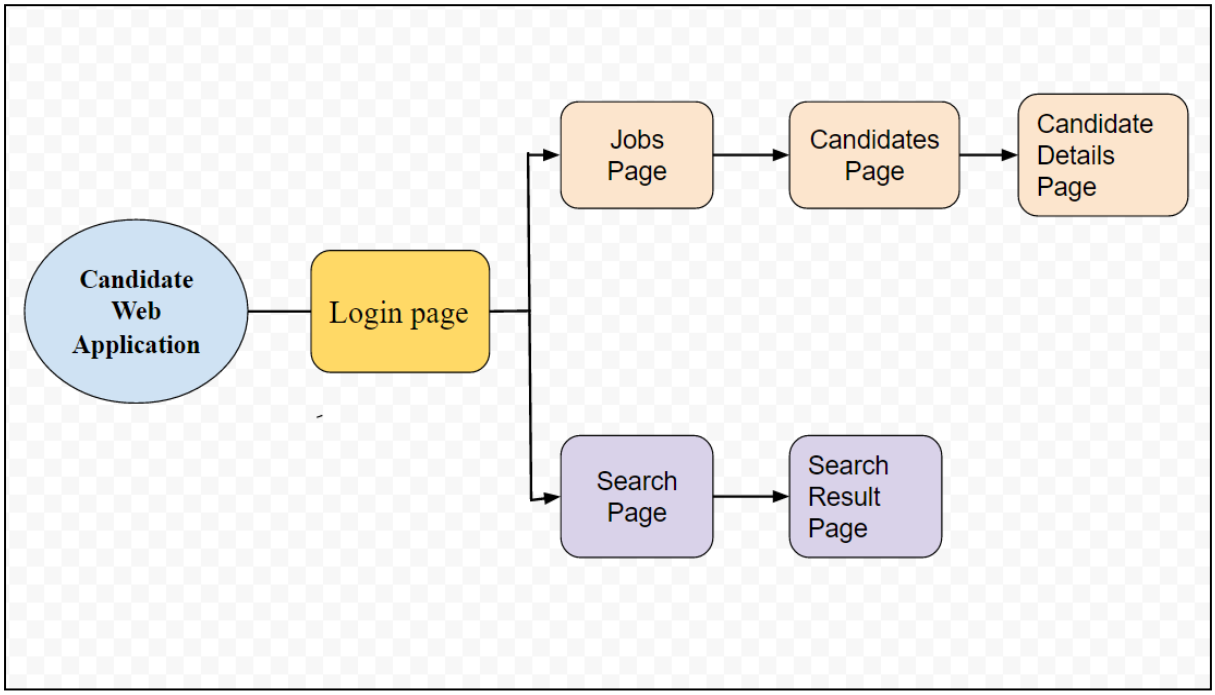

Figure 8: Sitemap for CandidatesWebApplication

Screenshot of CandidiateWebApplication

Login Page

The below screenshot is displayed the login page of the candidate web application. User should enter username and password to access the web application.

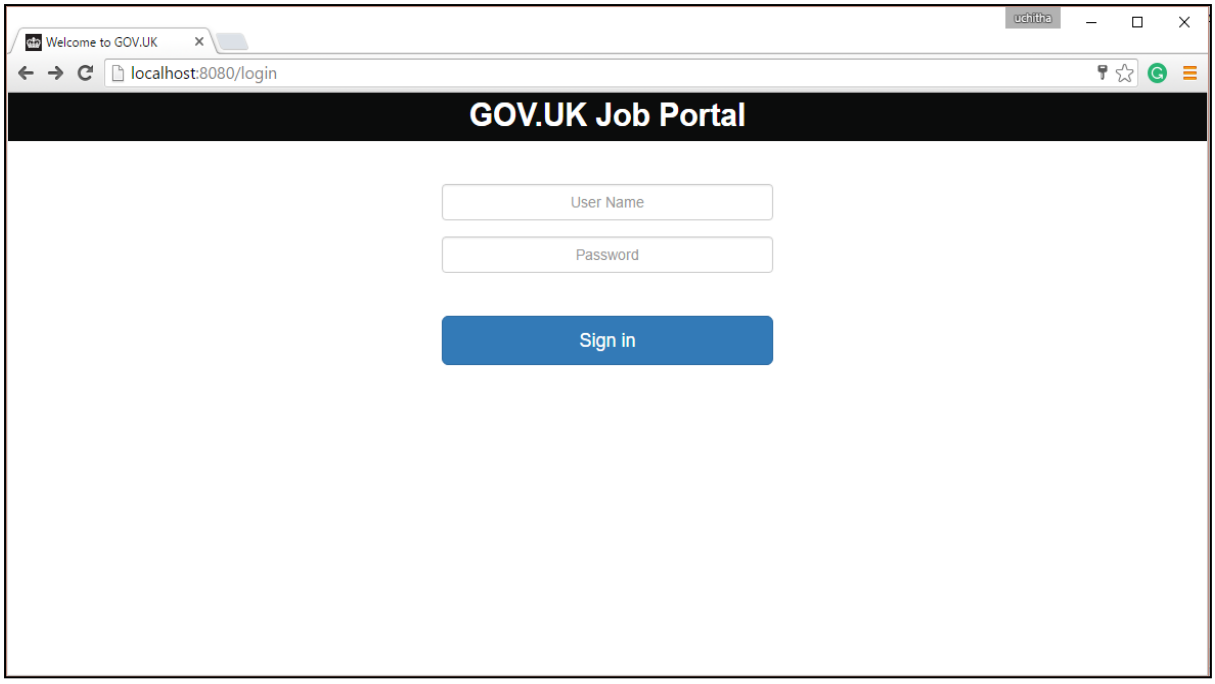

Figure 9: Login Page

Jobs page

The below screenshot is displayed the jobs page of the candidate web application. It is contained list of jobs published on GOV.UK. Once you click the specific job user can navigate to see list of applicant applied for that job.

Figure 10: Jobs Page

Candidates Page

The below screenshot is displayed the Candidates page of the candidate web application. This page list the all candidate applied for that Job. User can navigate to more details view by clicking the candidate name.

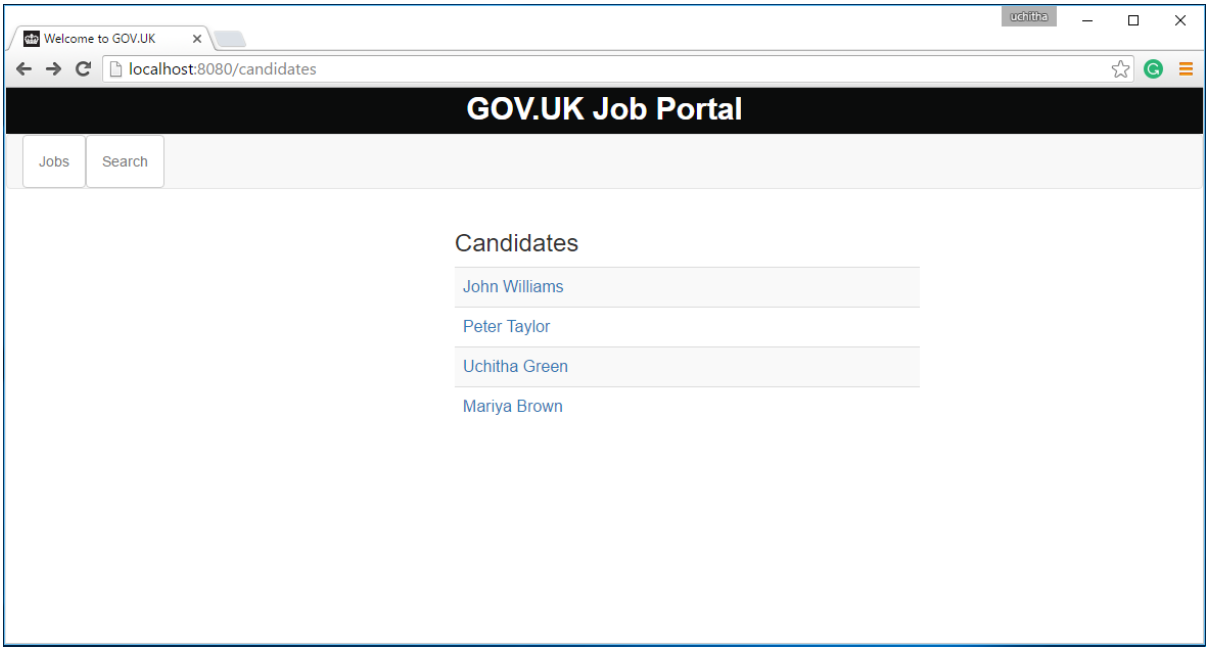

Figure 11 : Candidates Page

Candidate Details Page

The below screenshot is displayed the candidateDetails page of the candidate web application. This page consists more details about the one candidate Ex: email, phone, work summary..

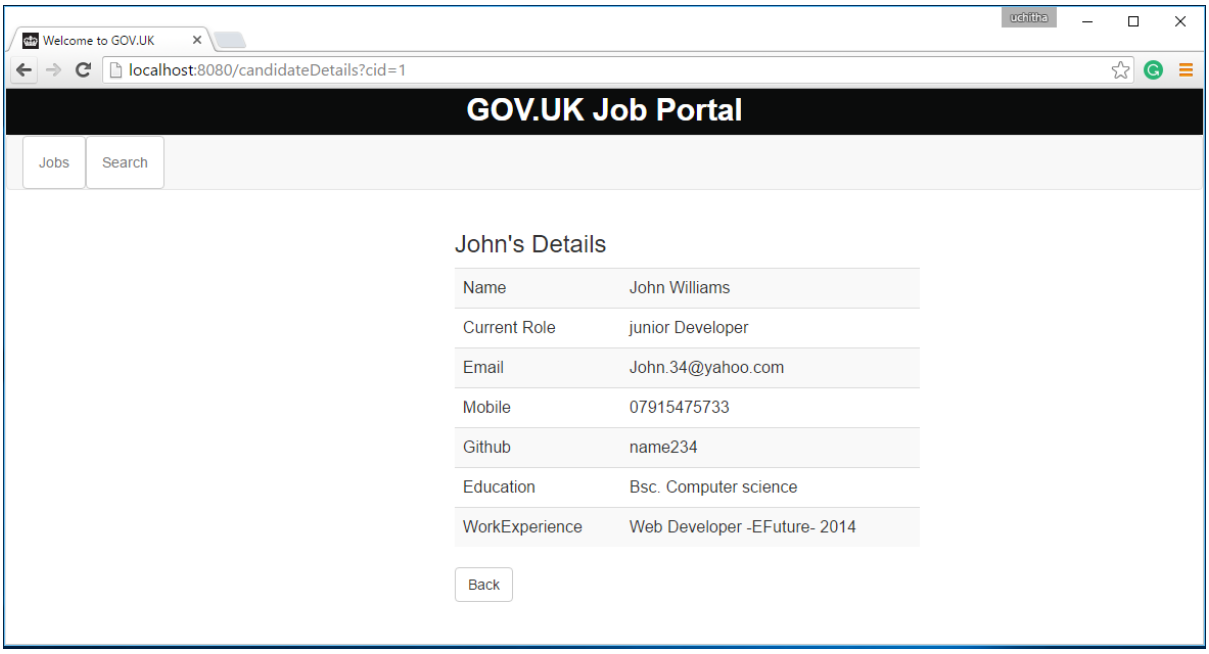

Figure 12: Candidate Detail Page

Search Page

The below screenshot is displayed the Search page of the candidate web application. User can search candidate details by candidate name or Id.

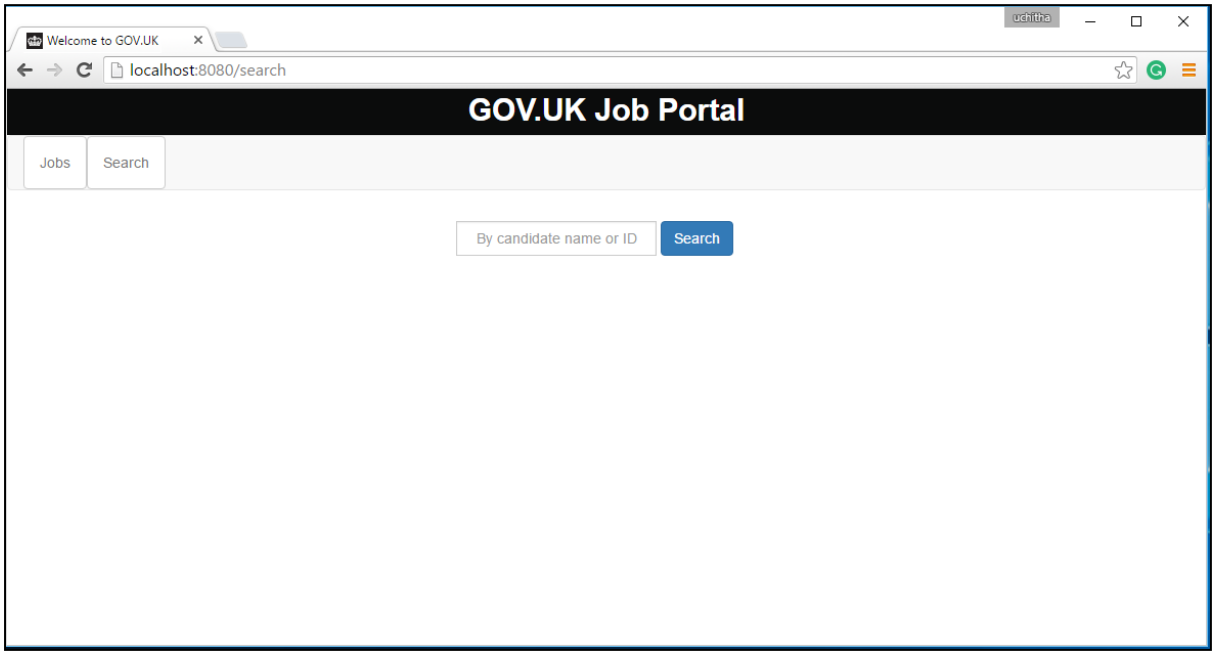

Figure 13: Search page

Search Result Page

The below screenshot is displayed the Search Result page of the candidate web application. It's displayed search candidate details.

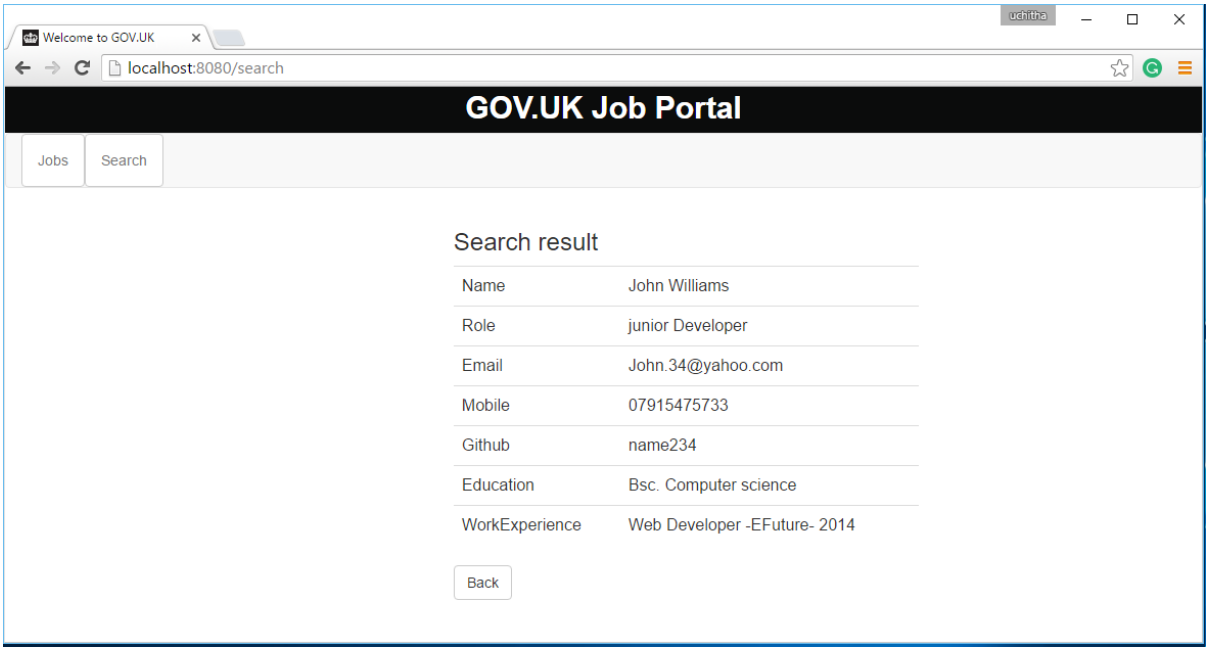

Figure 14 : Search Result Page

Project Structure and technologies used

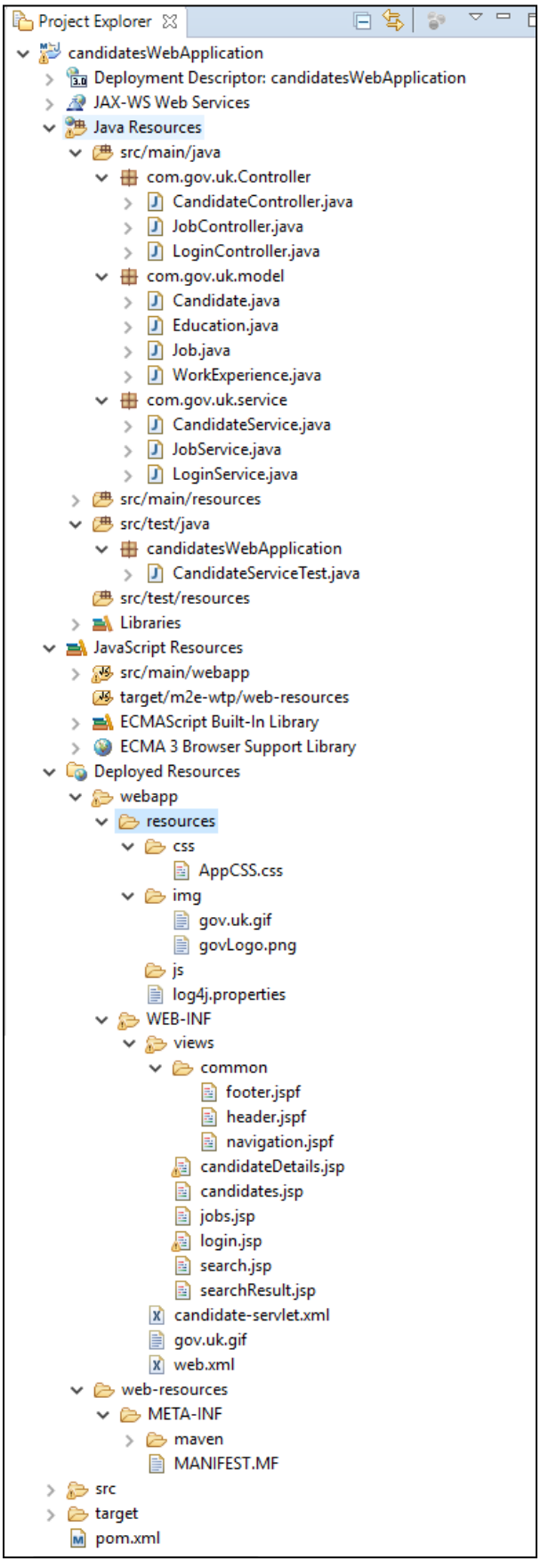

Figure 15: Full project structure

List of technologies, design patterns and structure are used for the candidates web application development describes below.

Java, Maven, Spring MVC, JSP, JSTL, CSS, HTML, Junit Front controller design pattern, bootstrap are the main technologies and design patterns used.

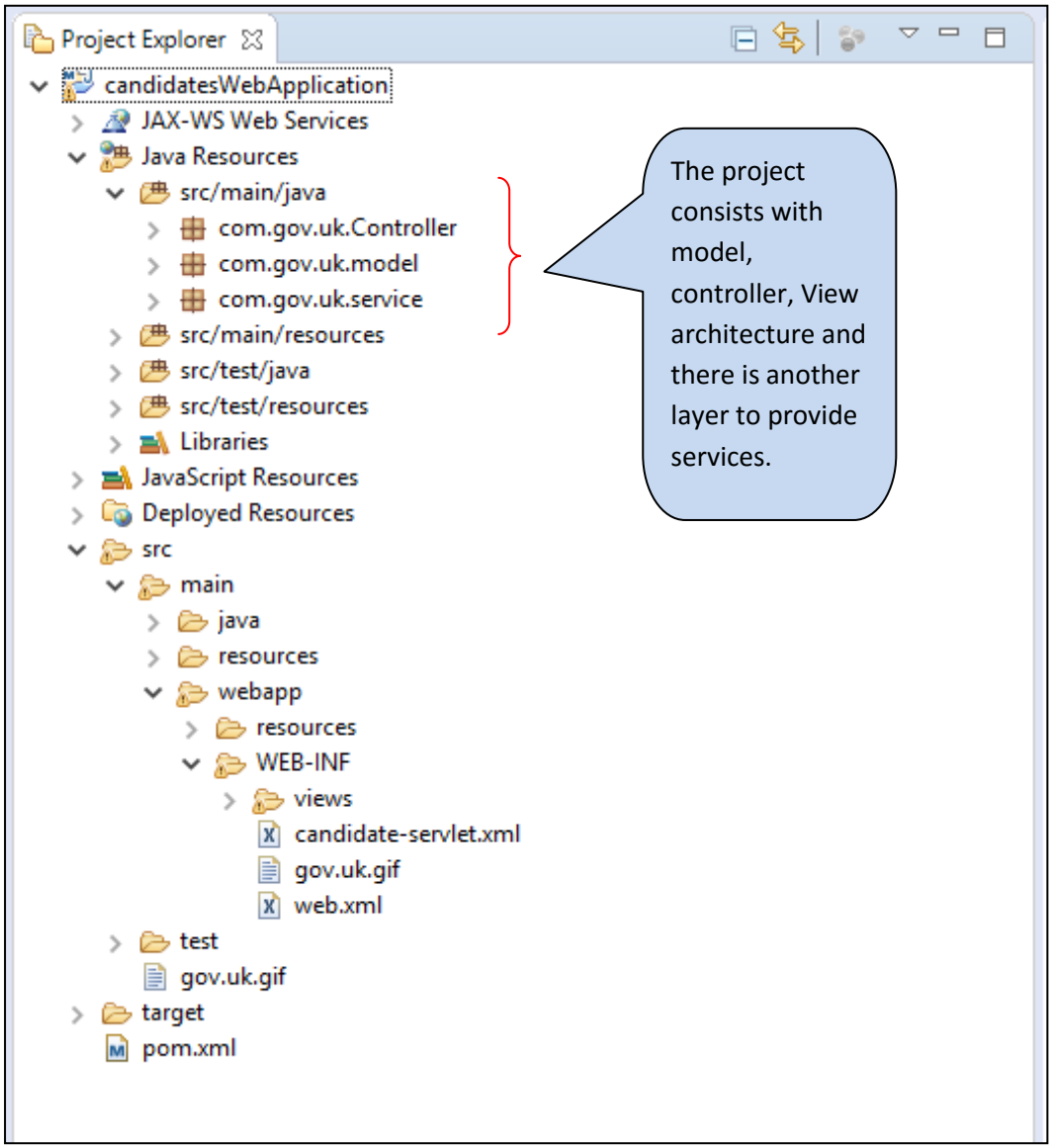

Figure 16: Model View Controller architecture in candidate web application

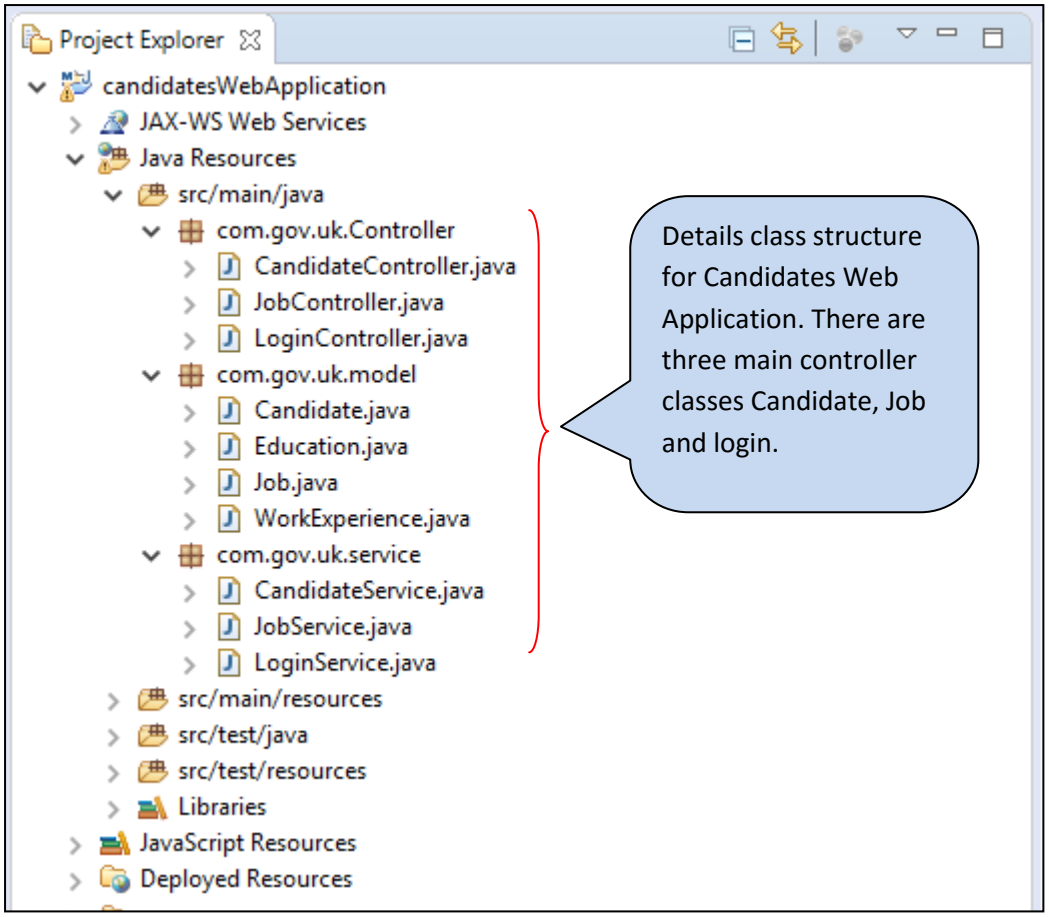

Figure 17: Detailed class structure of the Project

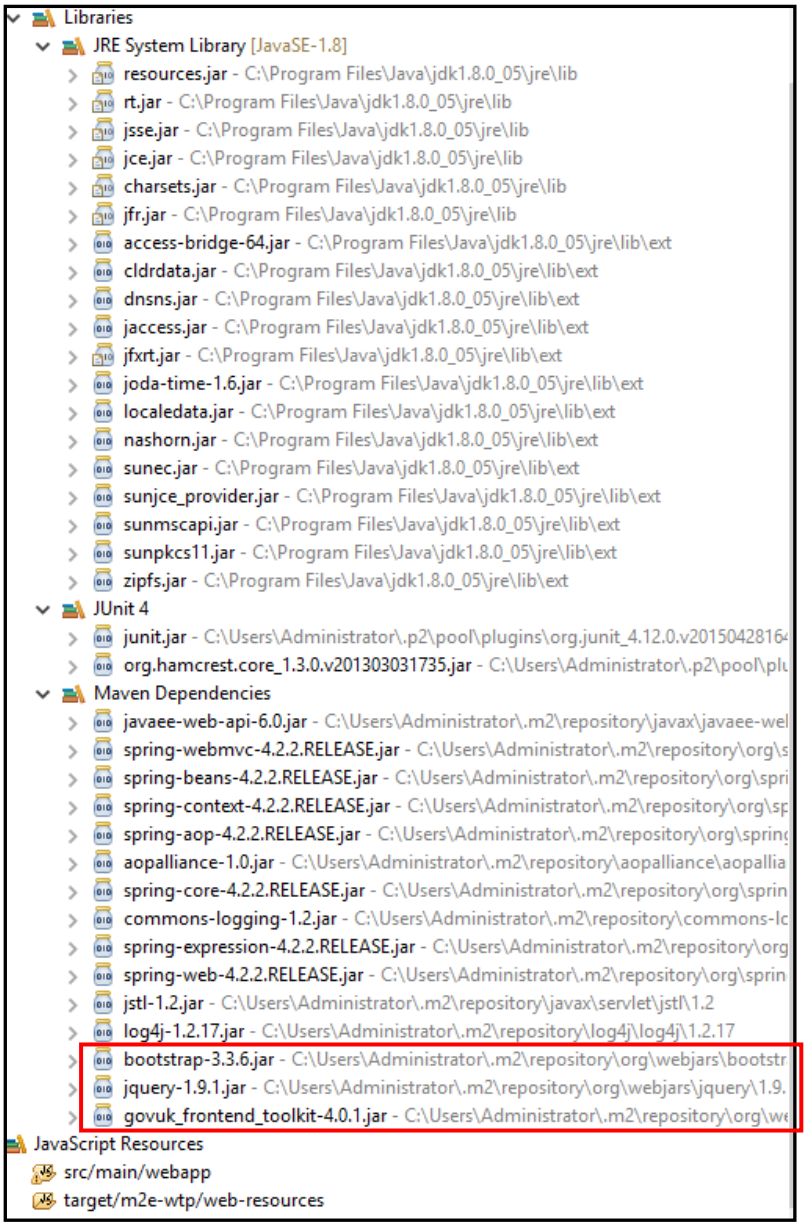

Figure 18: All jar file used for CandidateWebApplication development

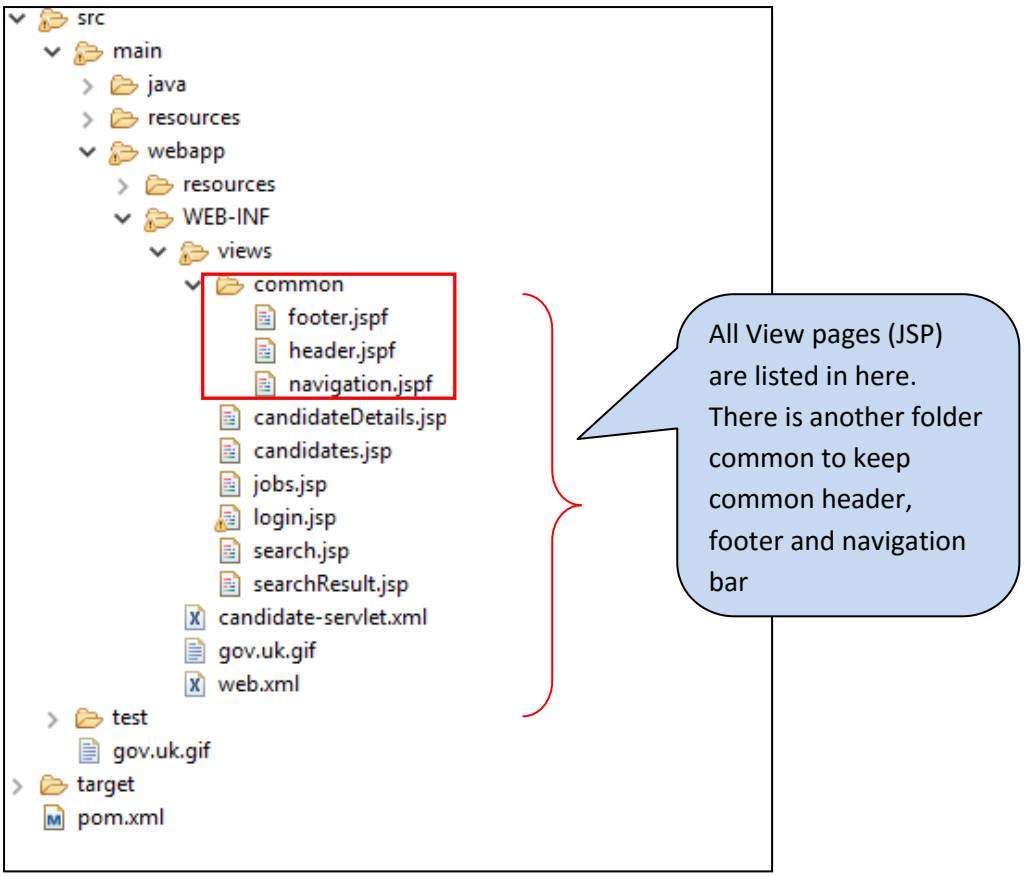

Figure 19: Front End Views (All JSP pages)

Test Cases

Candidate Service Test class has implemented to mapping with the functions implemented in CandidateService class

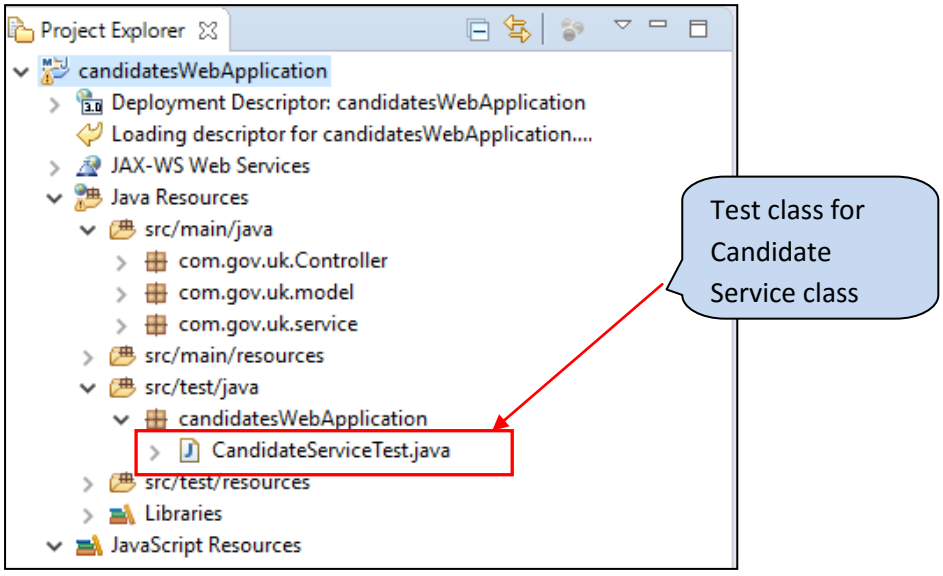

Figure 20: Test Class for Candidate Service class

```
package candidatesWebApplication;

import static org.junit.Assert.*;

import java.util.ArrayList;

import java.util.List;

import org.junit.Test;

import com.gov.uk.model.Candidate;

import com.gov.uk.service.CandidateService;


public class CandidateServiceTest {

	private Candidate candidate;

	private CandidateService candidateService = new CandidateService();

	List<Candidate> candidateList = new ArrayList<Candidate>();

	@Test

	public void retrieveCandidate() {

		candidateList = candidateService.retrieveCandidate();

		assertTrue(candidateList.size()>0);

		System.out.println("candidateList is not empty");

	}

	@Test

	public void getCandidateDetails() {

		candidate = new Candidate(1, "peter","Brown", "web Devloper", "07915475744",
"peter@yahoo.com", "peter34", "Bsc Information Technology", "None");

		candidate = candidateService.getCandidateDetails(1);

		assertEquals(candidate, candidate);



	}

}
```