**BDA LAB11**

**CE142**

**Aim : Delivering Reports including visualizations and customized properties.**

**Lab Setup :**

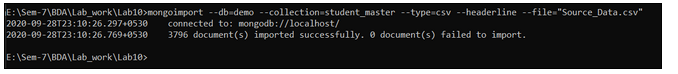
Jaspersoft Studio download link :

<https://community.jaspersoft.com/project/jaspersoft-studio>

**Exercise :**

**Task 1 : University Name on X-axis, Avg Percentage by various students on Y-Axis. Use a bar chart.**

Import **Source\_Data.csv** file in mongodb which is in the Lab10 directory using **mongoimport** command



Open **Jaspersoft Studio**.

In **Repository explorer** Right Click **Data Adapters** Select **Create Data Adapter**>>**MongoDB Connection**.

Provide Name :**MongoDB Connection**

Mongo URI :**mongodb://localhost:27017/{database\_name}**

Click **Finish.**

Goto **File**>> **New** >>**JasperReport**.

From **Report Templates** Select **Blank Letter** Layout click **Next**.

Provide **File name** : Task1.jrxml

Click **Finish**.

You can see the **Task1.jrxml** in the **Workspace** Part.

Click the **Dataset and Query Dialog** icon.

Select **MongoDB Connection** from **Data Adapter** icon.

Select **MongoDbQuery** from **Language** dropdown menu and write following query,

{

collectionName : "student\_master",

findQuery : {TrainingExam1Percentage : {$ne : ''},

TrainingExam2Percentage:{$ne : ''}},

limit : 25

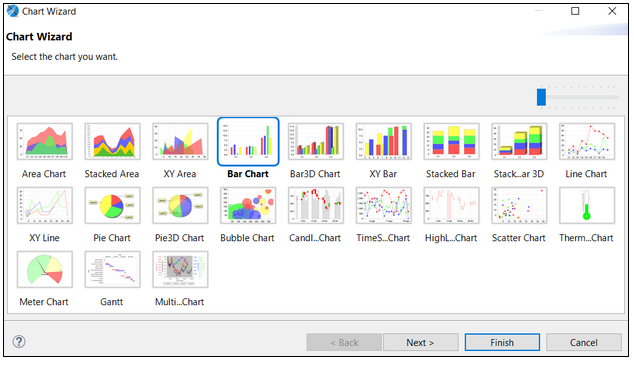
}

Click **Read Fields.** It shows all fields of collection.

Click **OK**.

Now, At the Right side From **Basic Elements** drag **Chart** icon on **Design** Part.

In the **Chart Wizard** section select **Bar Chart**. Click **Next**



Then In **Chart Data Configuration** Set the parameters as below:

Series :$F{First Name}

Value :($F{TrainingExam1Percentage}+$F{TrainingExam2Percentage})/2

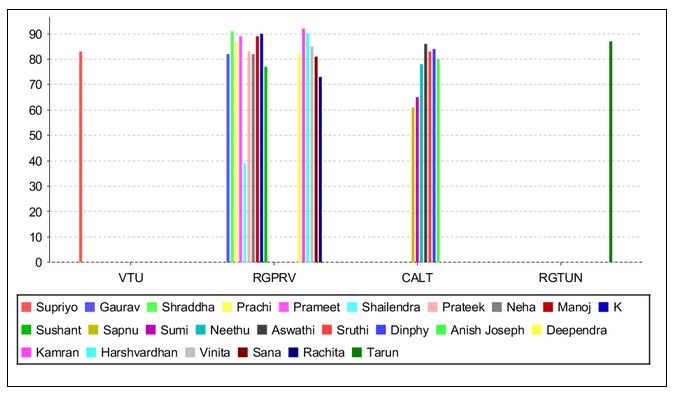
Label :$F{First Name}

Category :$F{University Name}

Then Click **Finish**.

Goto**Preview**.

It will Generate Graph like below.



**Task 2 : Embed the JasperReport to your java application/web.**

I have created a Java web application with apache maven using NetBeans IDE 12.1.

Open the **Apache NetBeans IDE 12.1**.

From Menu Bar Select **File**>>**New Project**.

Categories :   **Java with Maven**

Projects :**Web Application**.

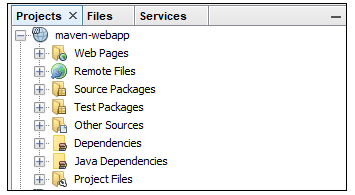
Provide **Project Name** and **Project Location** then click **Next**.

Select Server :**Apache Tomcat or TomEE**.

Java EE Version :**Java EE 7 Web**

Click **Finish**.

It creates a web application containing the following structure.



First we need to download necessary dependencies for the project.

Goto **Project Files**>>**pom.xml**.

Add the dependencies as below.

<dependency>

            <groupId>org.mongodb</groupId>

            <artifactId>mongo-java-driver</artifactId>

            <version>3.12.7</version>

    </dependency>

<dependency>

            <groupId>net.sf.jasperreports</groupId>

            <artifactId>jasperreports</artifactId>

            <version>6.14.0</version>

</dependency>

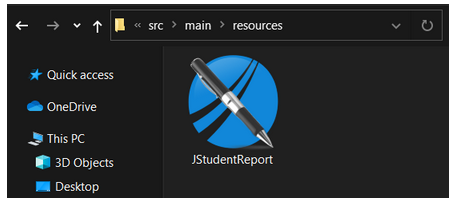
Right Click **Dependencies** select **Download Declared Dependencies**.

Then choose the jasper report file that you want to embed with a web application. Remember that this file should not contain any query that only contains report configuration.

In the **Workspace** section Select **MyReports**>>**JasperReports**>>**Compile Report**.

Then copy the .jasper file and the following location.

**{project-name}/src/main/resources/**



After this You need to create the following files.

Click on  https://lh5.googleusercontent.com/1IQgxRdD0mfbj8mw26xyq0rD1O0_58hsMbDMbVTfhJNjlnGUBsQ9WawDkN27-npCtLuSE9vx_C657CQsl8db9XF1hZxhEQPdFdJQxHglfYaxcUxRK92x-QTUYJTyQX6ML6KvSunG to create a new file.

**StudentRc.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package com.webapp.maven.webapp;

/\*\*

 \*

 \* @author HP

 \*/

public class StudentRc {

    public Double percentage;

    public String university;

    public String background;

    public StudentRc(Double percentage, String university, String background) {

        this.percentage = percentage;

        this.university = university;

        this.background = background;

    }

    public StudentRc() {

    }

    public Double getPercentage() {

        return percentage;

    }

    public String getUniversity() {

        return university;

    }

    public String getBackground() {

        return background;

    }

    public void setPercentage(Double percentage) {

        this.percentage = percentage;

    }

    public void setUniversity(String university) {

        this.university = university;

}

    public void setBackground(String background) {

        this.background = background;

    }

    @Override

    public String toString() {

        return "StudentRc{" + "percentage=" + percentage + ", university=" + university + ", background=" + background + '}';

    }

}

**MongoConnection.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package com.webapp.maven.webapp;

/\*\*

 \*

 \* @author HP

 \*/

import com.mongodb.MongoClient;

import com.mongodb.client.MongoCollection;

import org.bson.Document;

public class MongoConnection {

    MongoClientmongoc;

    public  MongoCollection<Document>getMongoCollection()

    {

        mongoc = new MongoClient( "localhost" , 27017 );

        return mongoc.getDatabase("demo").getCollection("student\_master");

    }

}

**Retriever.java**

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package com.webapp.maven.webapp;

import com.mongodb.BasicDBObject;

import com.mongodb.DBObject;

import com.webapp.maven.webapp.\*;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoCursor;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.HashSet;

import java.util.Iterator;

import java.util.List;

import java.util.Map.Entry;

import java.util.Set;

import org.bson.Document;

/\*\*

 \*

 \* @author HP

 \*/

public class Retriever {

    MongoConnectionmcon;

    public List<StudentRc>getStudentList(String bck)

    {

        mcon = new MongoConnection();

        MongoCollection<Document>mcol = mcon.getMongoCollection();

        DBObject query = BasicDBObject.parse("{$group:{\_id:\"$University Name\",avgp:{$avg:\"$DegreePercentage\"}}}");

        MongoCursor<Document>mcr = mcol.find().iterator();

        Set<String>unlist = getUniversityList();

        List<StudentRc>lsr = new ArrayList<StudentRc>();

        HashMap<String,List<Double>>avpercentage = new HashMap<String,List<Double>>();

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

        while(mcr.hasNext())

        {

            Document row  =mcr.next();

if(!row.getString("Background(CS\\NCS)").equals(bck) && !bck.equals("\_\_ALL\_\_"))

                continue;

            String uname = row.getString("University Name");

            if(!avpercentage.containsKey(uname))

            {

                    avpercentage.put(uname, new ArrayList<Double>());

                    backuni.put(uname, row.getString("Background(CS\\NCS)"));

            }

            avpercentage.get(uname).add(Double.parseDouble(row.get("DegreePercentage").toString()));

        }

        Iterator<Entry<String,String>>bguin = backuni.entrySet().iterator();

        while(bguin.hasNext())

        {

            Entry<String,String> temp = bguin.next();

            Double avpr = 0.0;

            avpr = avpercentage.get(temp.getKey()).stream().map(pr ->pr).reduce(avpr, (accumulator, \_item) -> accumulator + \_item);

            avpr = avpr / avpercentage.get(temp.getKey()).size();

            lsr.add(new StudentRc(avpr, temp.getKey(), temp.getValue()));

        }

        mcon.mongoc.close();

        return lsr;

    }

    public Set<String>getUniversityList()

    {

        mcon = new MongoConnection();

        MongoCursor<Document>mcr = mcon.getMongoCollection().find().iterator();

        Set<String>stu = new HashSet<String>();

        while(mcr.hasNext())

        {

            Document row  =mcr.next();

            stu.add(row.getString("University Name"));

        }

        mcon.mongoc.close();

        return stu;

    }

}

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="java.util.Set"%>

<%@page import="com.webapp.maven.webapp.Retriever"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>Select Backgroungtype</title>

        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>

    </head>

    <body>

        <br><br>

        <div class="container w-75 m-auto shadow p-3 mb-5 bg-white rounded">

            <div class="alert alert-light text-secondary w-50 m-auto " role="alert">

                Select <text class="text-primary">Background</text> From below Dropdown

            </div>

            <br>

            <select id="background" class="form-control form-control-sm w-50 m-auto" onchange="navigateToUni()" >

                    <option></option>

                    <option>CS</option>

                    <option>NCS</option>

                    <option>\_\_ALL\_\_</option>

            </select><!-- comment --></div>

    </body>

    <script>

        function navigateToUni()

        {

            uname = document.getElementById("background");

            window.location.replace("/maven-webapp/DisplayReport.jsp?bckg="+uname.value);

        }

    </script>

</html>

**DisplayReport.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="net.sf.jasperreports.engine.JasperPrintManager"%>

<%@page import="com.webapp.maven.webapp.Retriever"%>

<%@page import="net.sf.jasperreports.engine.JasperRunManager"%>

<%@page import="java.util.List"%>

<%@page import="java.util.ArrayList"%>

<%@page import="com.webapp.maven.webapp.StudentRc"%>

<%@page import="net.sf.jasperreports.engine.JasperPrint"%>

<%@page import="net.sf.jasperreports.engine.JasperFillManager"%>

<%@page import="java.util.Map"%>

<%@page import="java.util.HashMap"%>

<%@page import="net.sf.jasperreports.engine.data.JRBeanCollectionDataSource"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

    </head>

    <body>

        <%

        String bckg = request.getParameter("bckg");

         List<StudentRc> li = (new Retriever()).getStudentList(bckg);

        JRBeanCollectionDataSourcebeanColDataSource = new JRBeanCollectionDataSource(li);

        Map<String, Object> parameters = new HashMap<String, Object>();

        String jpath = "C:\\Users\\HP\\Documents\\NetBeansProjects\\maven-webapp\\src\\main\\resources\\JStudentReport.jasper";

        byte bytes[] = JasperRunManager.runReportToPdf(jpath, parameters,beanColDataSource);

        response.setContentType("application/pdf");

            response.setContentLength(bytes.length);

            ServletOutputStreamoutStream = response.getOutputStream();

            outStream.write(bytes, 0, bytes.length);

            outStream.flush();

            outStream.close();

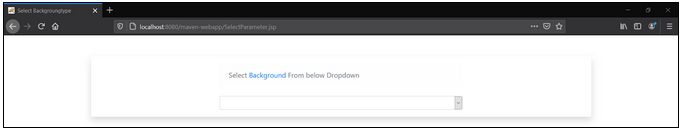
        %>

    </body>

</html>

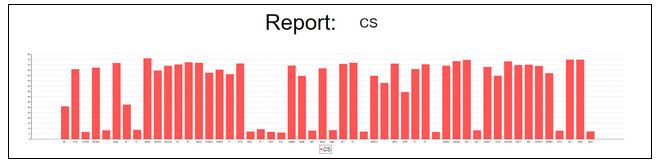
Finally click on **https://lh5.googleusercontent.com/sMtMqmkKkY0BxThi32kSnmu9FcJgahY1TvF806-YUnWkW6Rgtw_vOGNJjH2Rj9vXc-AhyqH4t5e5Nih4aNQPCaLeyiEELGXy4h7uEFttW4y95GbqZsuEatjRAOoFGD1iuj_HyodS**button to run project**.**

You are able to see web page like below.

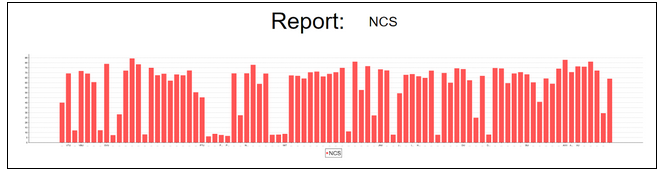


Now you can select the **background type** of student\_master table from the dropdown and it generates the report with a **bar chart** containing **University Name** on **X-axis** and **average percentage** on **Y-axis**.

Bar chart for CS background:



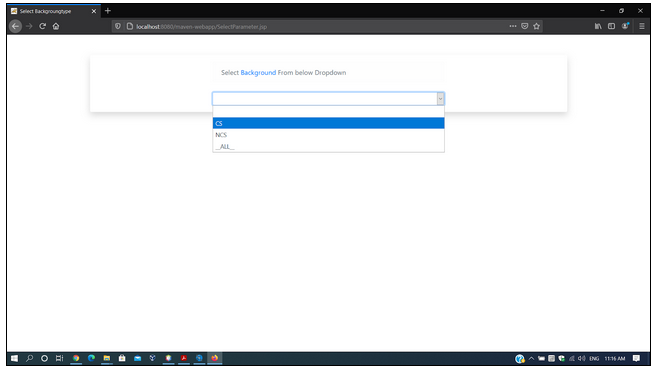
Bar chart for NCS background:



**Task 3 : Populate dropdown for parameter.**

I have already performed this task in the above activity.

Here I have created a dropdown menu for selection of  background type (**CS**/**NCS**).



Bar Chart for all backgrounds (**CS/NCS/None**).

