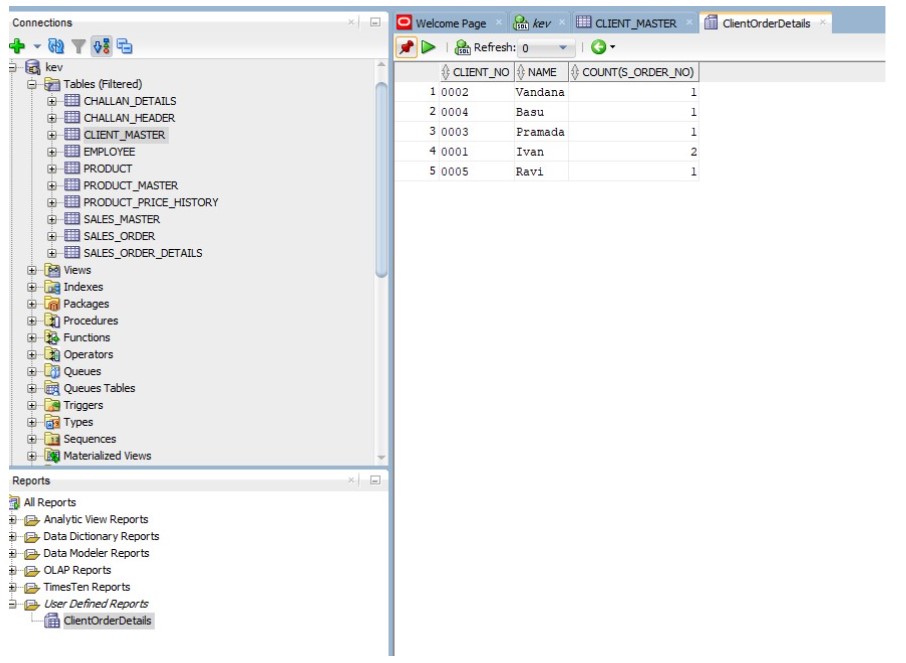
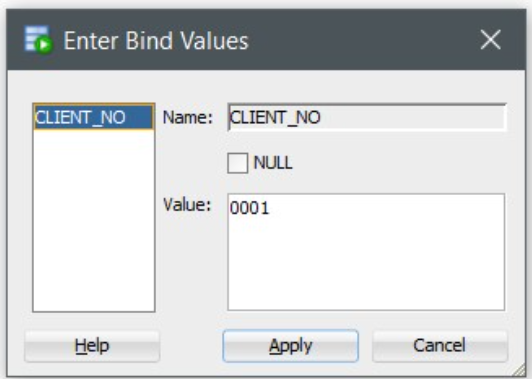
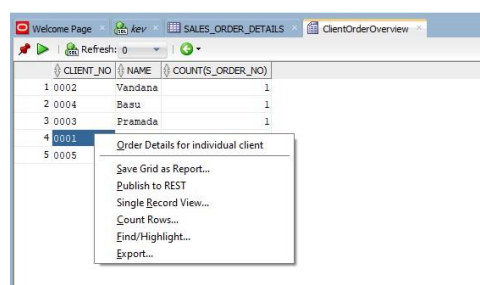
LAB - 10

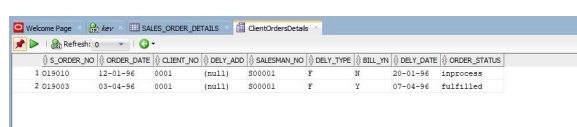
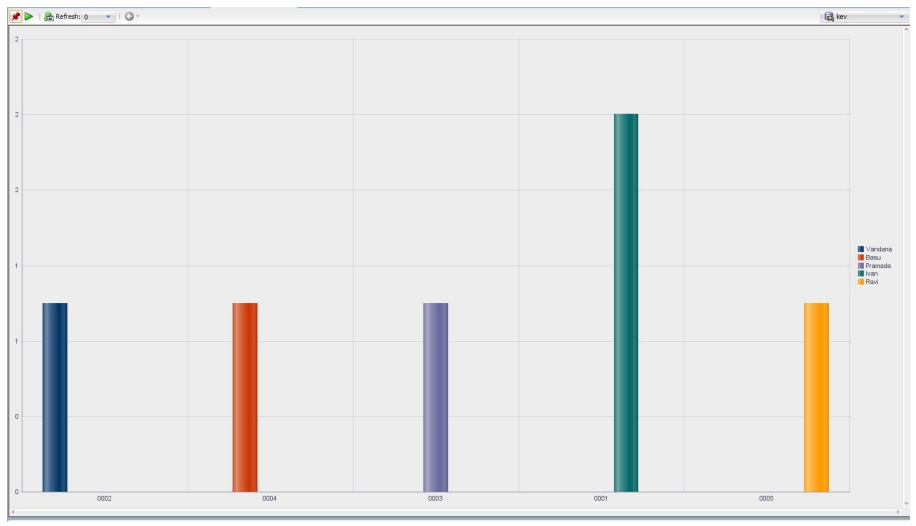
|  |  |
| --- | --- |
| Name | Keval D Gandevia |
| Roll Number | CE046 |
| ID | 19CEUEG017 |
| Subject | Big Data Analytics |

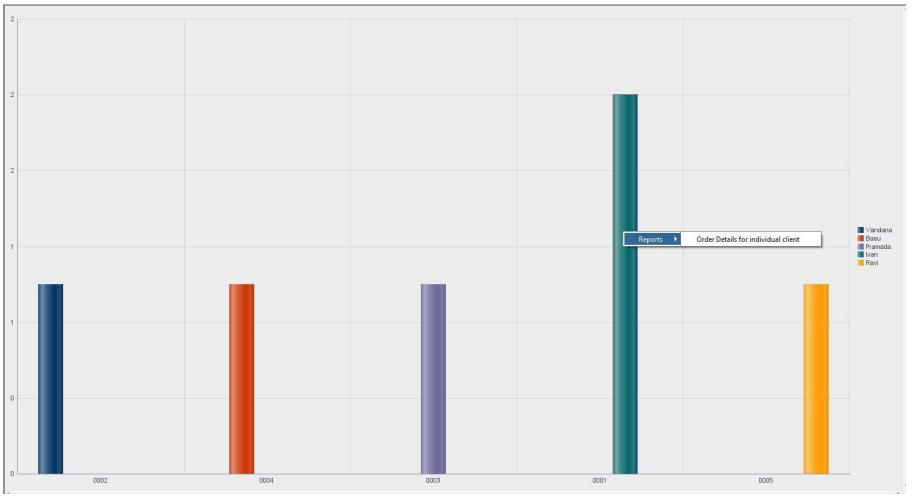
**Aim:** Delivering reports including visualizations and customized properties.

**Generate report using SQLDeveloper.**





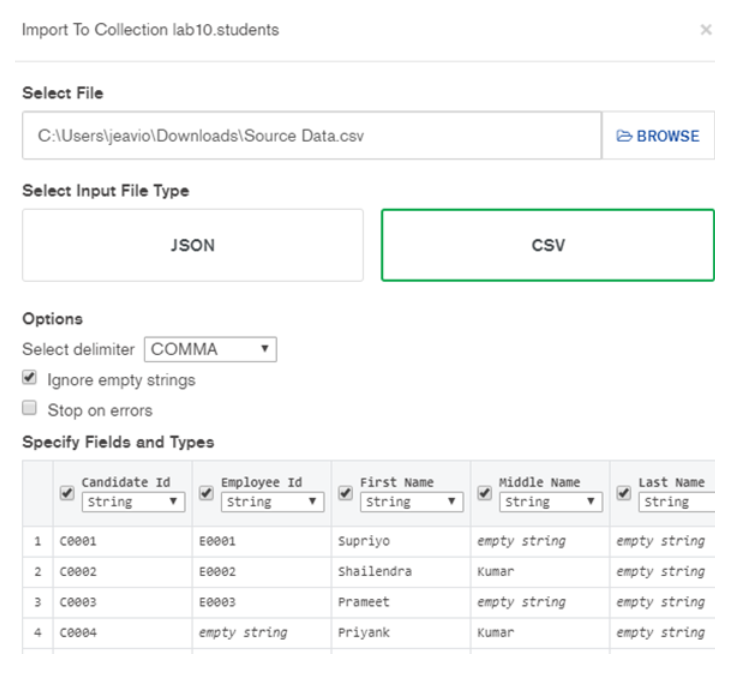




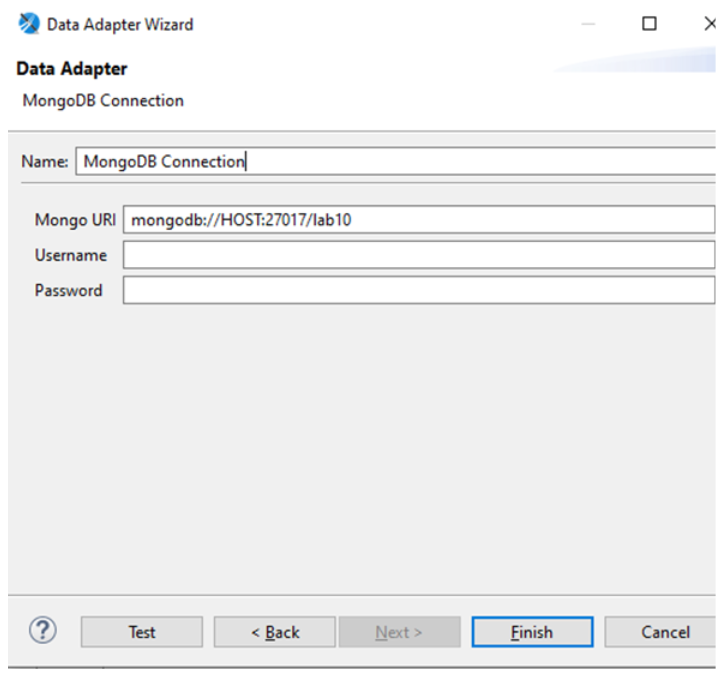
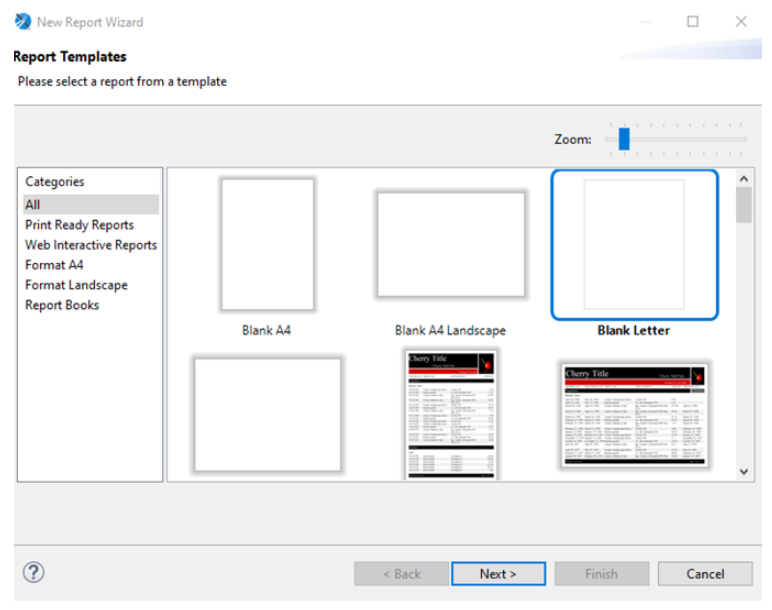
**Using Jasper studio.**

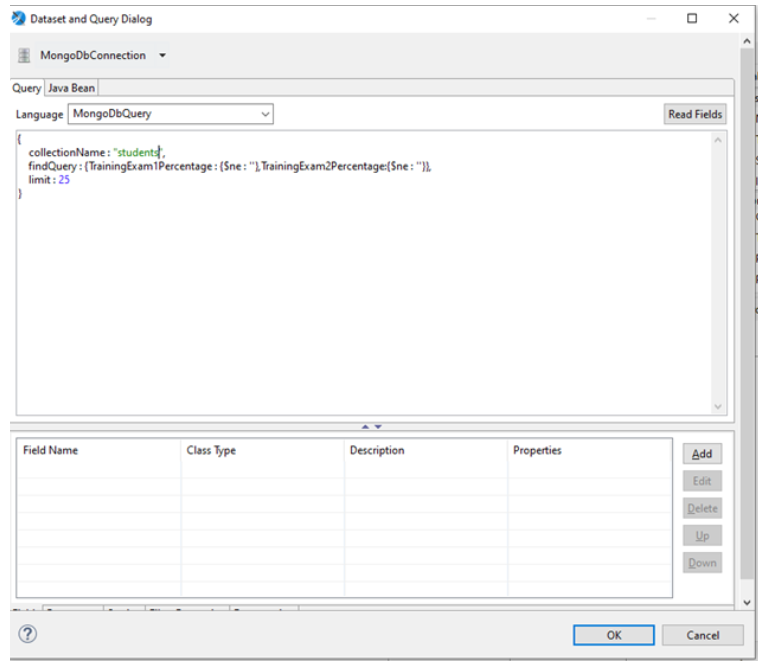
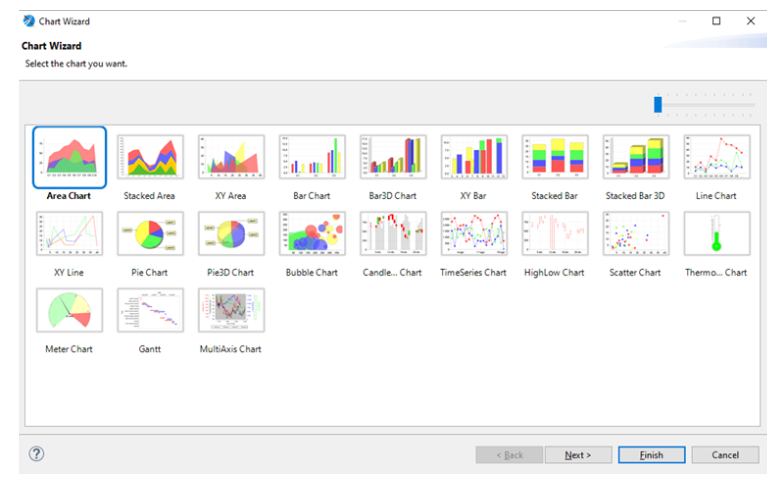
**Q. 1: University Name on X-axis, Avg Percentage by various students on Y-Axis. Use bar chart.**

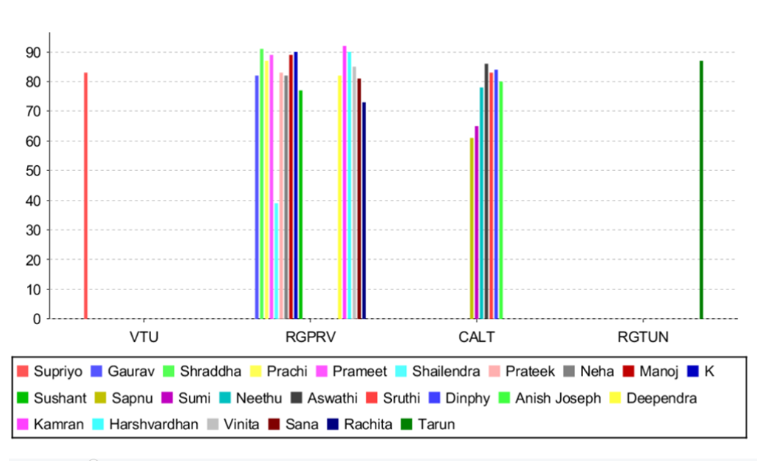
* **Import csv file:**



* **In Jasper studio create a data adapter for mongo db connection.**
* Provide Name: MongoDB Connection
* Mongo URI: mongodb://localhost:27017/lab10







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

* Create a new project in netbeans.
* Adding dependencies required for the project.
* **POM.xml file:**

|  |
| --- |
| <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.
* **Create following files:**

1. Student.java file:

|  |
| --- |
| public class Student  {  public Double percentage;  public String university;  public String background;  public Student(Double percentage, String university, String background) {  this.percentage = percentage;  this.university = university;  this.background = background;  }  public Student() {  }  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 "Student{" + "percentage=" + percentage + ", university=" + university + ", background=" +  background + '}';  }  } |

1. MongoDB connection file:

|  |
| --- |
| import com.mongodb.MongoClient;  import com.mongodb.client.MongoCollection;  import org.bson.Document;  public class MongoConnection  {  MongoClient mongoc;  public MongoCollection<Document> getMongoCollection()  {  mongoc = new MongoClient( "localhost" , 27017 );  return mongoc.getDatabase("lab10").getCollection("students");  }    } |

1. Retriever.java:

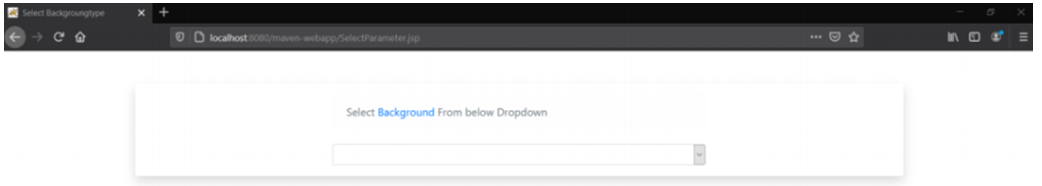
|  |
| --- |
| package com.webapp.maven.webapp;  import com.mongodb.BasicDBObject;  import com.mongodb.DBObject;  import com.webapp.maven.webapp.\*;  import com.mongodb.client.MongoCollection;  import com.mongodb.client.MongoCursor;  import java.util.ArrayList;import java.util.HashMap;  import java.util.HashSet;  import java.util.Iterator;  import java.util.List;  import java.util.Map.Entry;  import java.util.Set;  import org.bson.Document;  public class Retriever {  MongoConnection mcon;  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;  }  } |

1. SelectParameter.jsp:

|  |
| --- |
| <%@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> |

1. 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);  JRBeanCollectionDataSource beanColDataSource = 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);  ServletOutputStream outStream = response.getOutputStream();  outStream.write(bytes, 0, bytes.length);  outStream.flush();  outStream.close();  %>   </body>  </html> |





****