globsyn

Taking People To The Next Level

Automated Student Registration System

Group Members:

Deep Prakash Dey, Netaji Subhash Engineering College, 111090110037
Somnath Ghose Roy, Netaji Subhash Engineering College, 111090110097
Sabyasachi Pahari, Netaji Subhash Engineering College, 111090110077
Moumita Paul, Netaji Subhash Engineering College, 121090120014
Manish Nag, Netaji Subhash Engineering College, 121090120011

Table of Contents

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty Mr. Bhavesh Sangwan for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him/her time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Deep Prakash Dey
Somnath Ghose Roy
Sabyasachi Pahari
Moumita Paul
Manish Nag

Project Objective

Developing and implementing a Web based Automated Student Registration System for Photon Finishing School to help students register for a course.

The primary project goals consists of:

- To create different accounts for counsellors and registrar.
- Counsellor can register students in any of the available courses.
- Registrar can add a new course.
- Registrar can modify the capacity of a course.

Project Scope

A possible implementation of our software is as follows:

Counsellors are spread over the internet and can access the site through their respective system browsers. So are the registrars. The students can go to the counsellors who so ever is close to them. There they can apply for a course by filling a printed form and then submit the same to the counsellor. Then the counsellor registers the student by entering the necessary details and hence assign the student a unique student id.

The registrar in the institute can update the courses or add a new course.

The database will be centralized and hence the registration made from different places could be fetched in the institute from the server.

Requirement Specification

(This section contains the business challenge that the application is solving from the point of view of a user of the system)

- Domain Descrip

➢ JEE
➢ JSP
➢ XML
➢ CSS
➢ Servlet
➢ JDBC
➢ MySQL

XML we are not using separately but it used with JEE framework ... like in web.xml etc. So no need to mention here. Otherwise reviewer would find that you are using webservices or something like this

- Problem Definition

An educational institute (Photon Finishing School) wants to automate its student registration system. Institute offers courses on IT subjects. Example of some course are C, C++, and Java, JEE etc. Each course has certain duration, start date, end date, total number of students for admission.

Develop software which must have following facilities:-A data entry screen through which councilor enters data of students who is taking admission.

Screen has fields to accept students name, email id, and phone no .Choices of courses is available through dropdown list. For every student who has taken admission, a unique number will be generated by the system and that will be displayed on screen.

A screen where all course names are given. When councilor selects one of the course and in another screen total capacity and enrolled student no will be displayed.

Registrar can create a new course. For that purpose a screen needs to be developed which will accept all necessary details for a creating a course and save them in disk. For every course system must generate a unique course id.

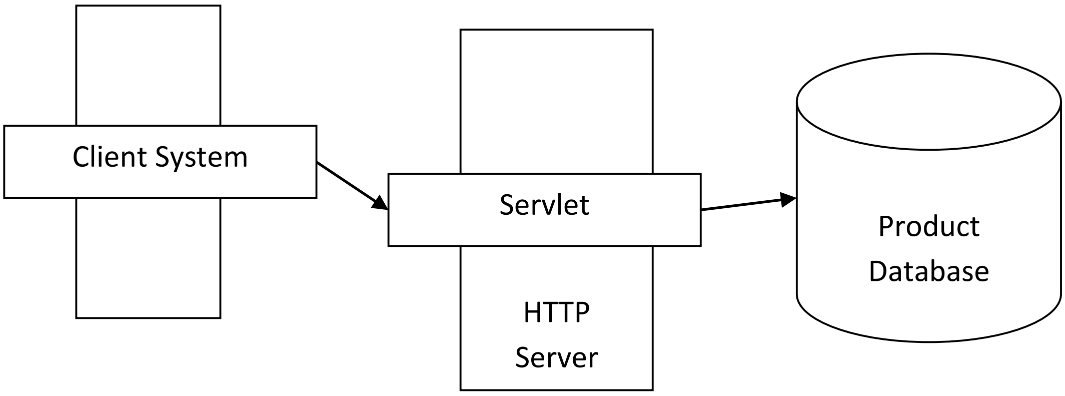
Registrar can increase capacity of students. A screen to be developed for that purpose.

- ## Functional Requirements

1. Login page has got three fields: emailid, password, and designation.

   If the user is a valid user, then he logins as a counsellor or registrar.
2. If the user is Counsellor, then he can register a student, view available seats for a course or change password.
3. On clicking register a student link, then a new window asking to select a course gets displayed. Once he chooses a course, and clicks the submit button, another page appears where the student details are to be entered. After submitting the details a student id is generated for the student. From there on, the counsellor can go back to his profile.
4. If the user is Registrar, then he can add a new course, update capacity of existing course or change password.
5. Once he clicks on add new courses, a page opens where the registrar has to enter the course details. After submitting, an unique course id is generated.
6. If he clicks on Update capacity, he gets a view of the details of the specified course. From there he can update the capacity of the existing course.
7. When they (registrar or counsellor) click on logout button the login page appears.
8. In our software the registrar has the sole right to access and update the database.
9. Also the counsellor cannot delete or validate a student.
10. The system will limit access to authorized user only.

Workflow



- Hardware /Software Requirements

- Hardware requirement

➢ Client side

- Processor : Intel(R) Core(TM) i5-2430M CPU @2.40GHz 2.40 GHz
- System type : 64-bit Operating System
- Installed Memory: 4.00 GB.

➢ Server side

- Processor : Intel(R) Core(TM) i5-2430M CPU @2.40GHz 2.40 GHz
- System type : 64-bit Operating System
- Installed Memory: 4.00 GB.

- Software Requirement

- ➢ Client side
- Google Chrome/Mozilla Firefox

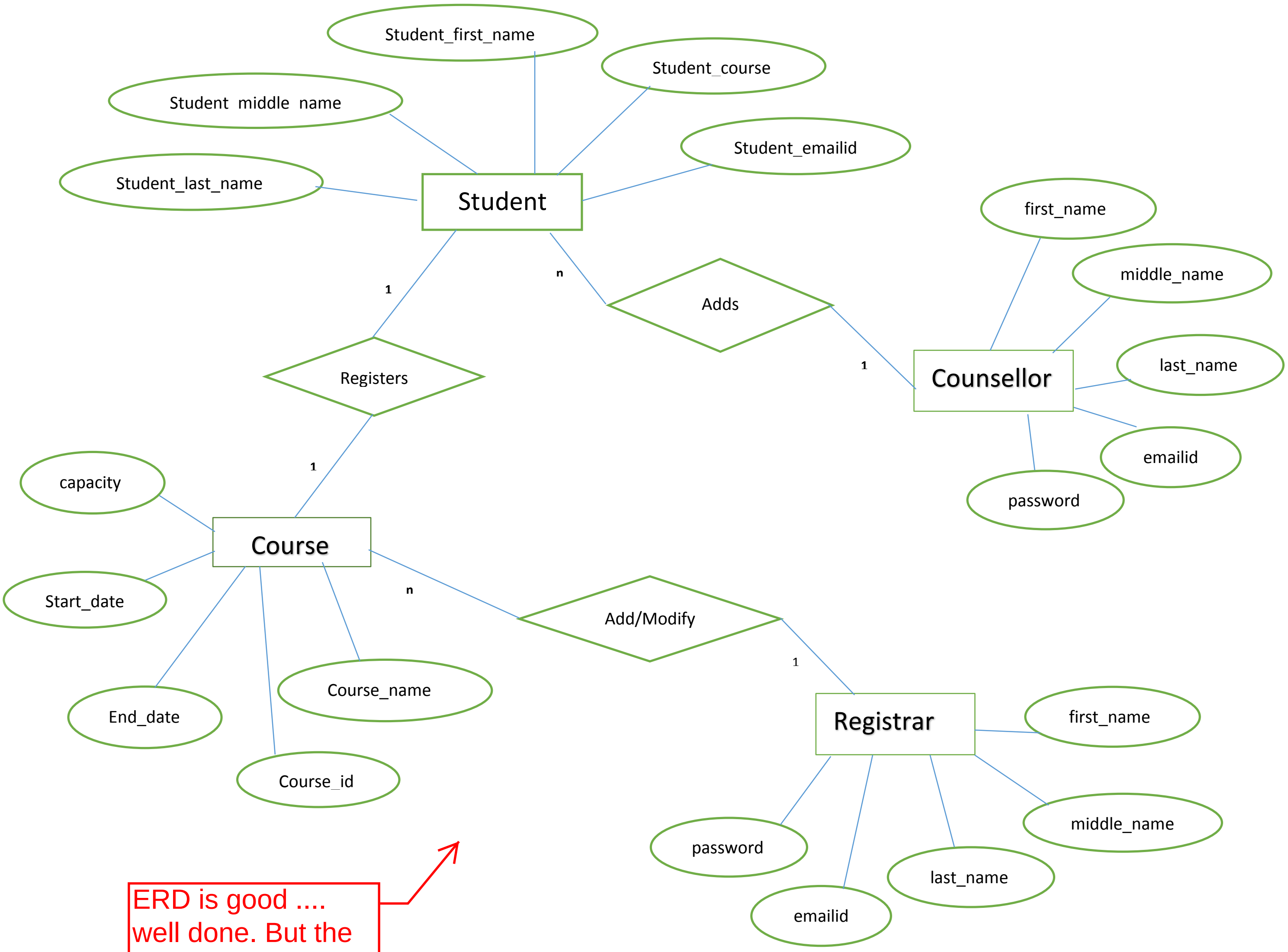
➢ Server side

- Mysql-connector-java-5.1.27-bin
- JBoss 7.1 Runtime[jboss-as-7.1.0 Final Runtime]
- Jdk 1.7.0_13
- Jre7

Database Design

(This section describes the Entity Relationship (ER) Diagram and description of the table structure)

ER DIAGRAM:



ERD is good .... well done. But the relation between student and course could be M-M

This is a more normalized entity relationship diagram than the one for the database we have used.

Description of table structure:

Table name: student

Field	Type	Null	Key	Default	Extra
student_id	Int(11)	NO	PRI	NULL	auto_increment
student_first_name	Varchar(20)	YES		NULL	
student_middle_name	Varchar(20)	YES		NULL	
student_last_name	Varchar(20)	YES		NULL	
student_emailid	Varchar(30)	YES		NULL	
student_course	Varchar(25)	YES		NULL	

Table name: member

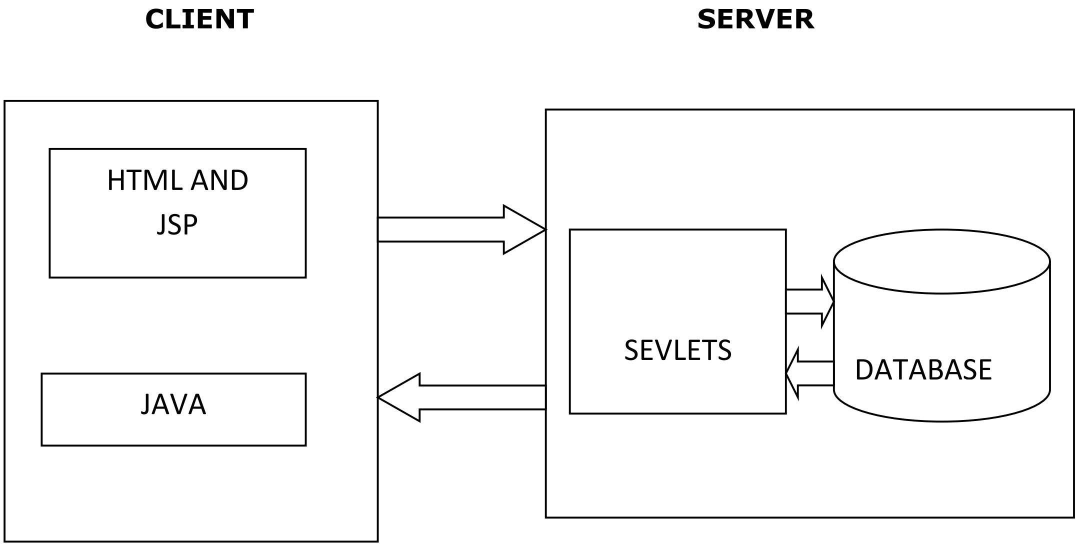
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(20)	YES		NULL	
middle_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
emailid	varchar(30)	YES		NULL	
password	varchar(10)	YES		NULL	
designation	varchar(20)	YES		NULL	

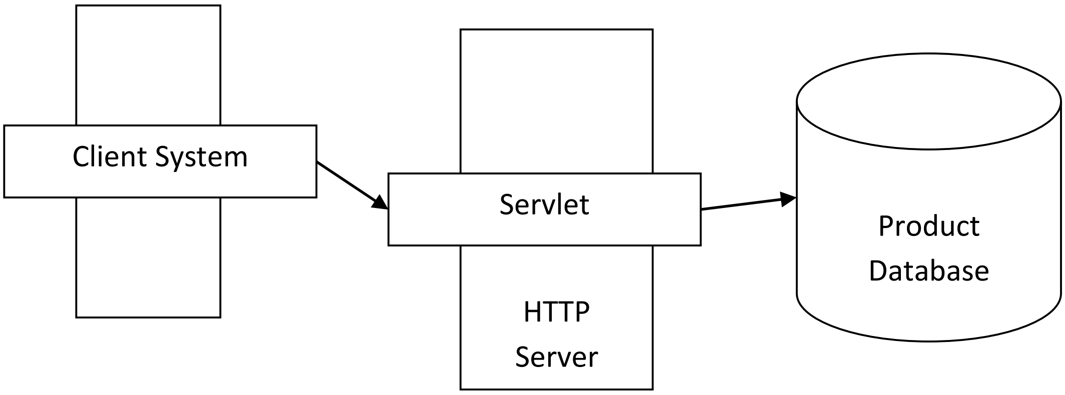
Table name: course

Field	Type	Null	Key	Default	Extra
course_id	int(11)	NO	PRI	NULL	auto_increment
course_name	varchar(25)	YES		NULL	
start_date	date	YES		NULL	
end_date	date	YES		NULL	
capacity	int(11)	YES		NULL	

Application Work Flow

(This section displays the flow of information in the application)

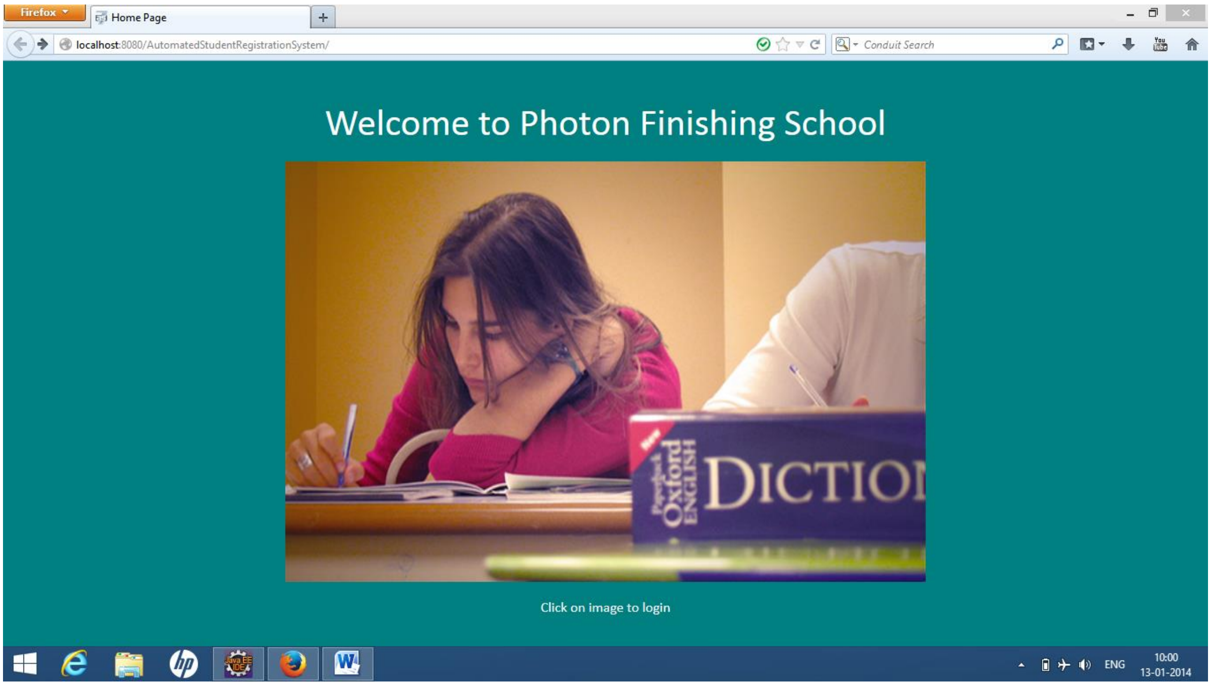




Screenshots

(This section contains the important screenshots of the application)

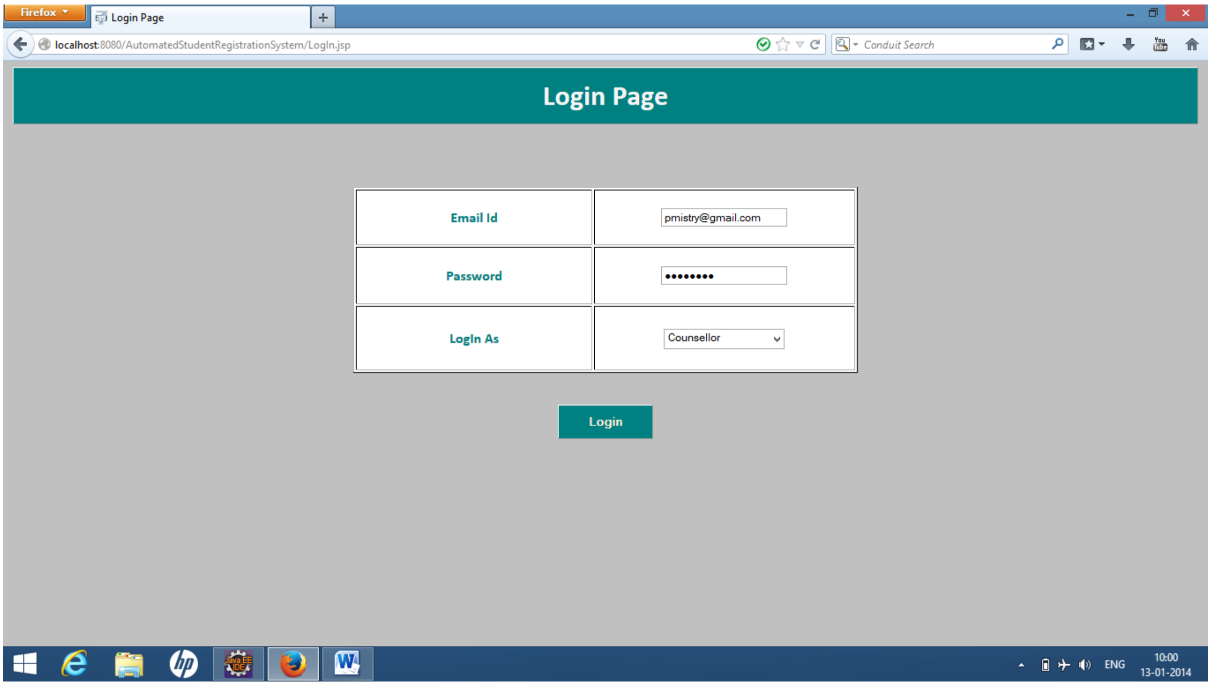
Welcome page



Welcome to Photon Finishing School

Click on image to login

Login Page

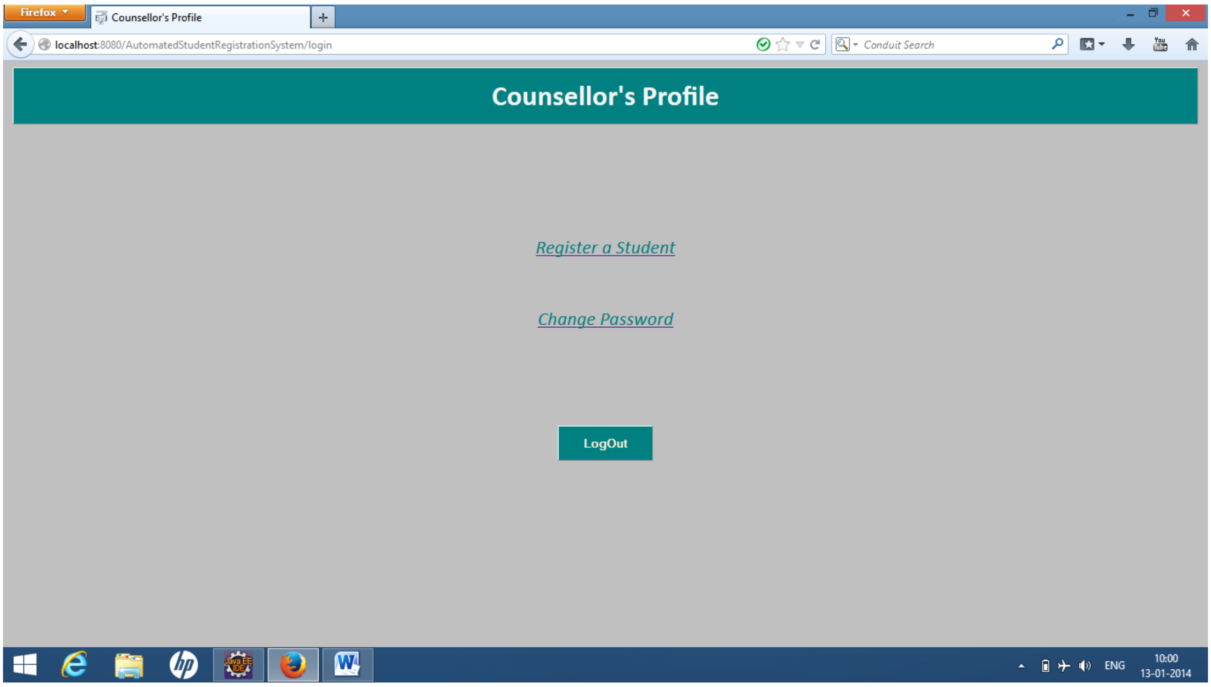


Login Page

Email Id	pmistry@gmail.com
Password	••••••••
LogIn As	Counsellor

Login

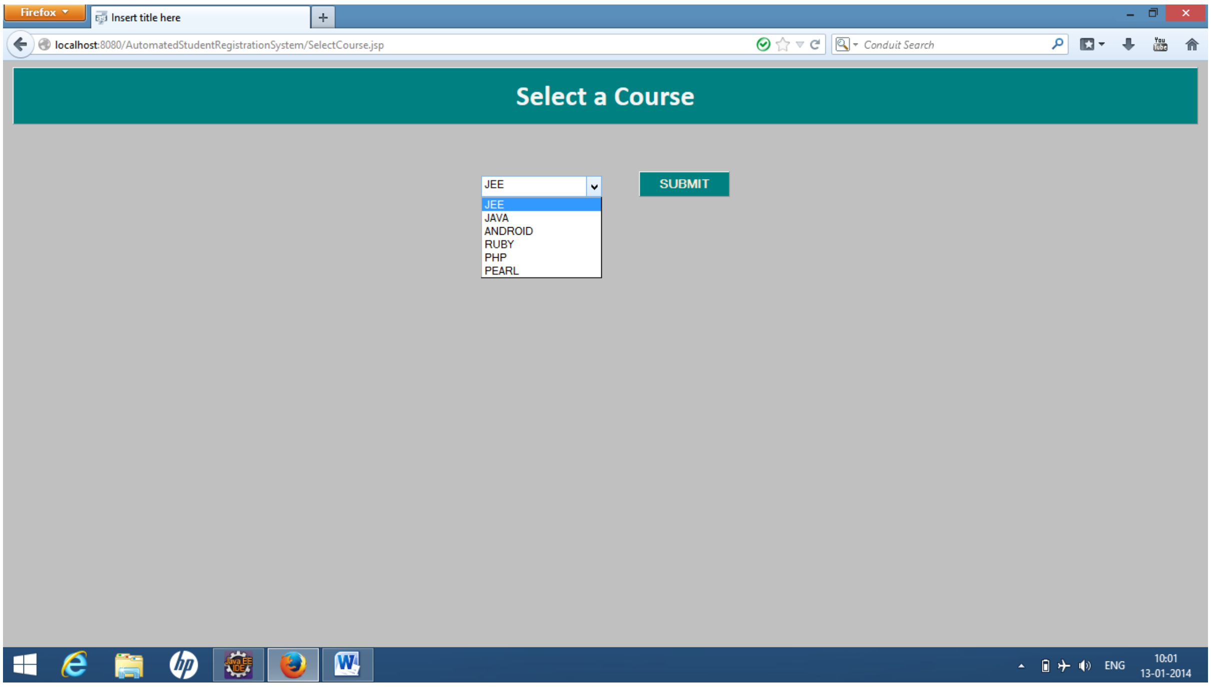
Counsellor's profile



localhost:8080/AutomatedStudentRegistrationSystem/login

Counsellor's Profile

Register a student

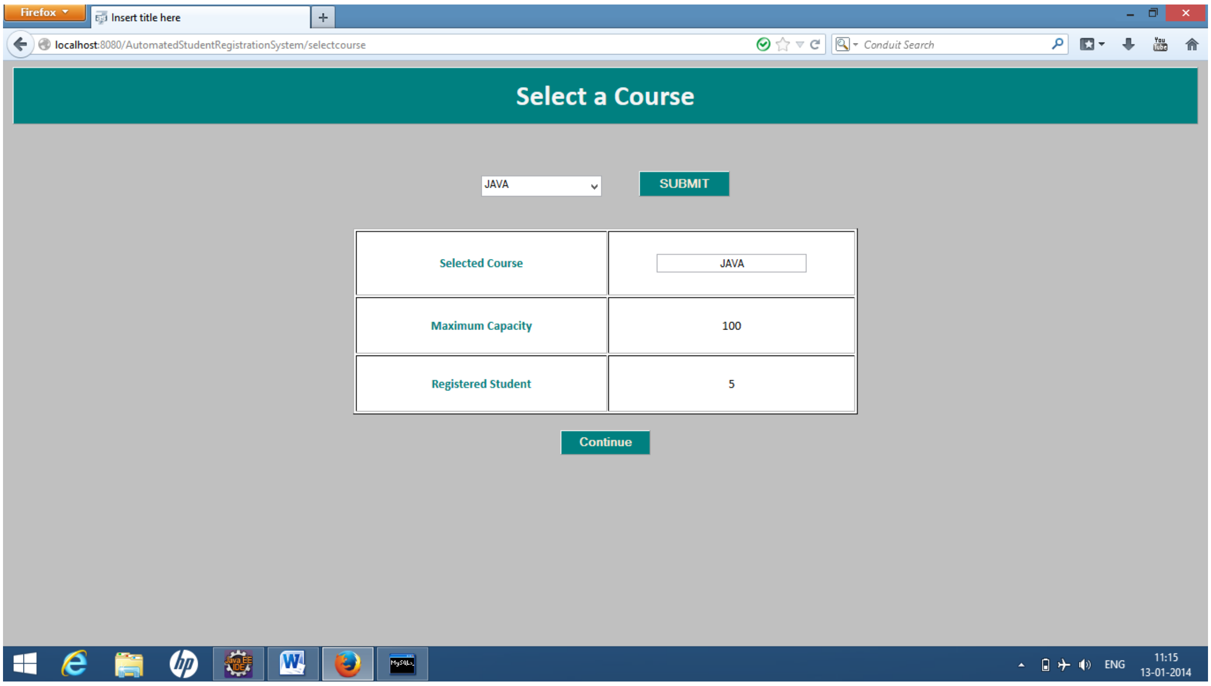


localhost:8080/AutomatedStudentRegistrationSystem/SelectCourse.jsp

Select a Course

Explanation for screen shot is required ....

Seats are available, so we can continue.

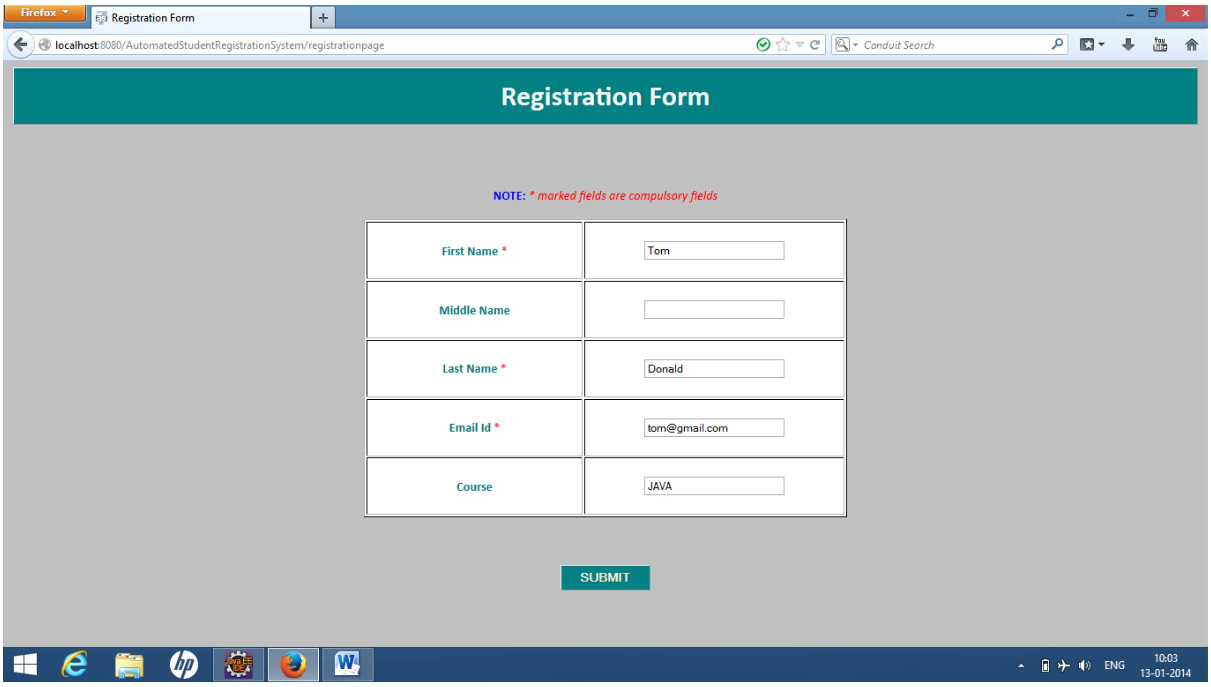


Select a Course

JAVA

Selected Course	JAVA
Maximum Capacity	100
Registered Student	5

Student registration form

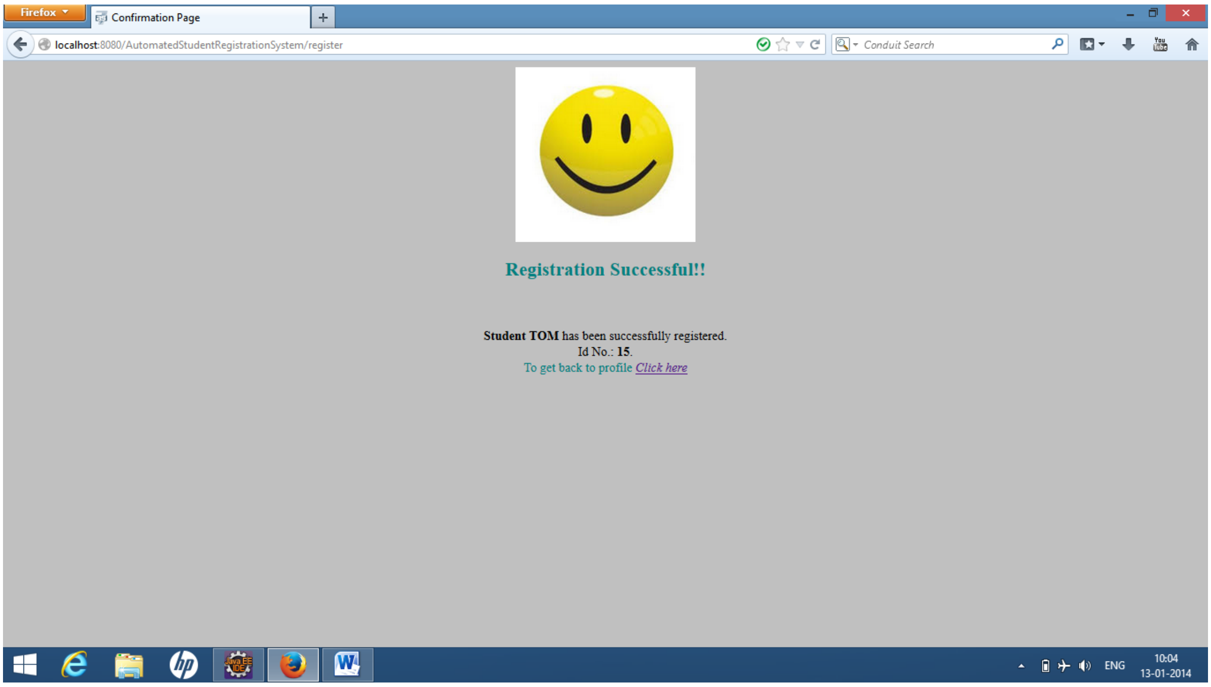

Firefox
Registration Form
localhost:8080/AutomatedStudentRegistrationSystem/registrationpage
Conduit Search
SUBMIT
ENG
10:03
13-01-2014

Registration Form

NOTE: * marked fields are compulsory fields

First Name *	Tom
Middle Name	
Last Name *	Donald
Email Id *	tom@gmail.com
Course	JAVA

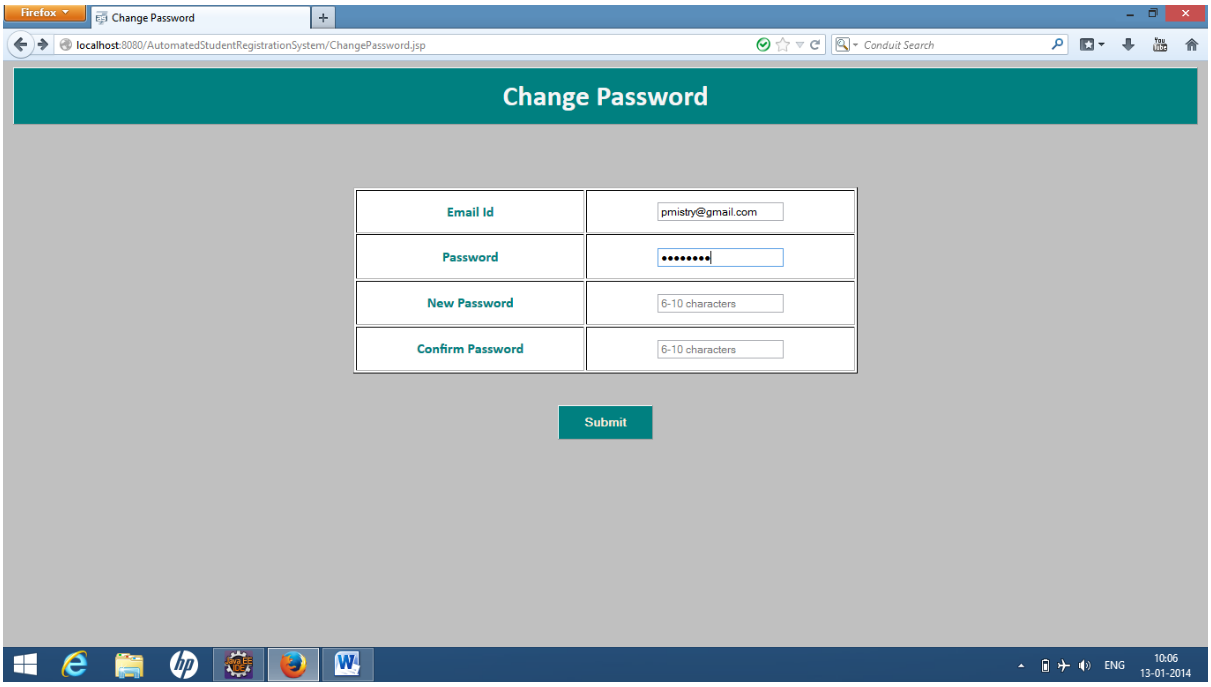
Registration confirmation page



Registration Successful!!

Student TOM has been successfully registered.
Id No.: **15**.
To get back to profile *Click here*

Back to Counsellor's profile

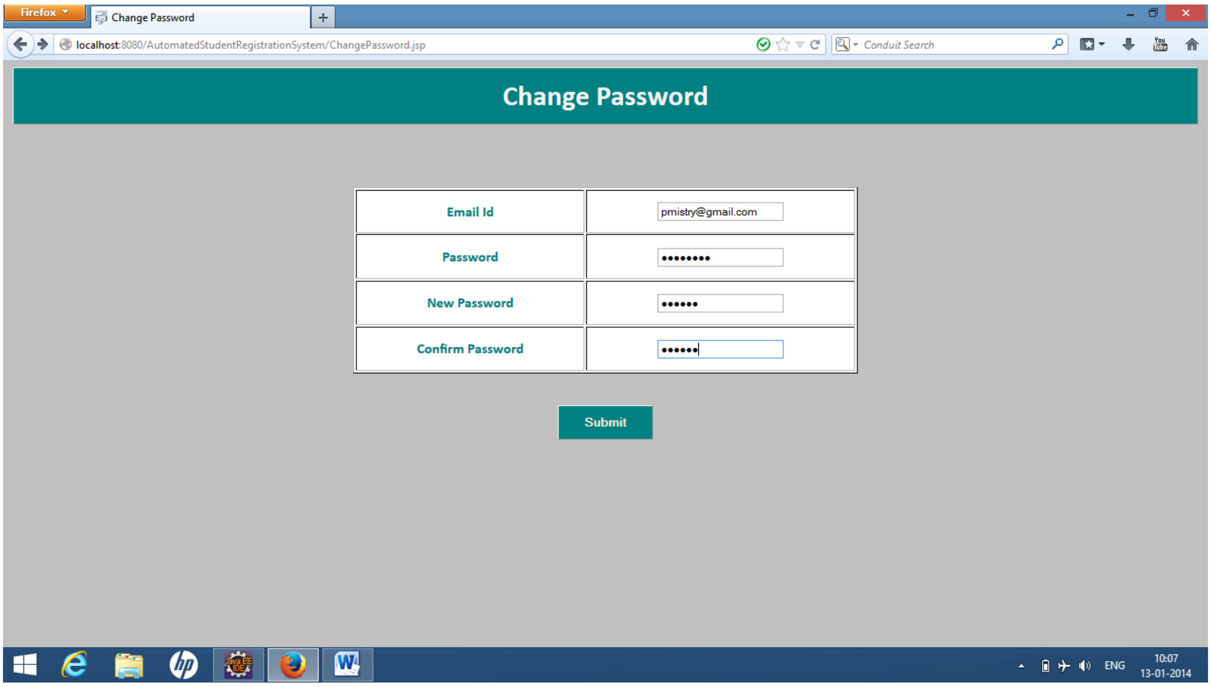

Firefox
Change Password
localhost:8080/AutomatedStudentRegistrationSystem/ChangePassword.jsp
Conduit Search
ENG
10:06
13-01-2014

Change Password

Email Id	pmistry@gmail.com
Password	••••••••
New Password	6-10 characters
Confirm Password	6-10 characters

Submit

Change password page

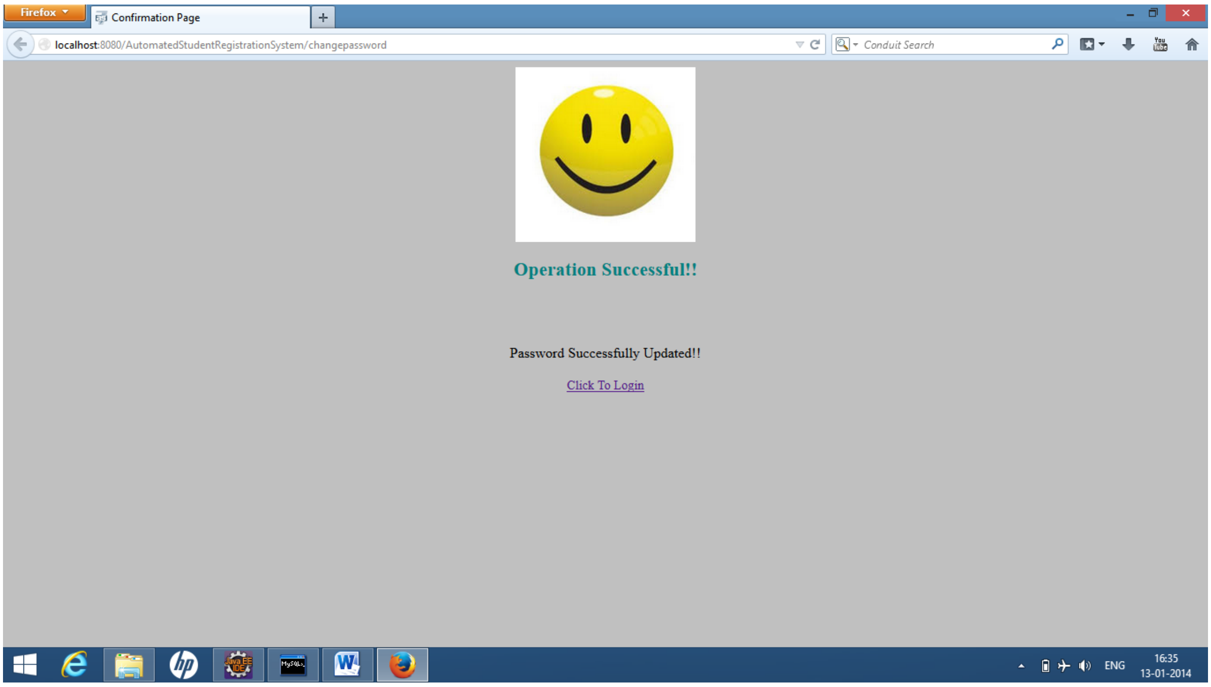


localhost:8080/AutomatedStudentRegistrationSystem/ChangePassword.jsp

Change Password

Email Id	pmistry@gmail.com
Password	••••••••
New Password	••••••
Confirm Password	••••••

Password change confirmation page



localhost:8080/AutomatedStudentRegistrationSystem/changepassword

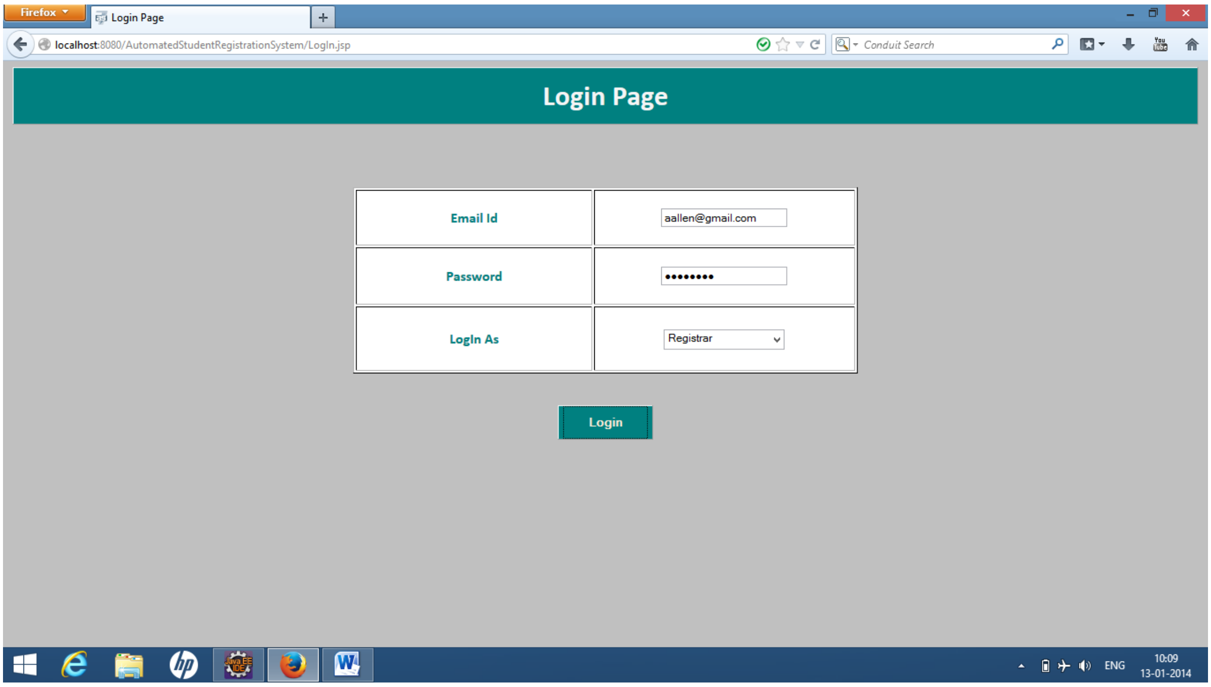
Operation Successful!!

Password Successfully Updated!!

Click To Login

<End of Counsellor's functionalities>

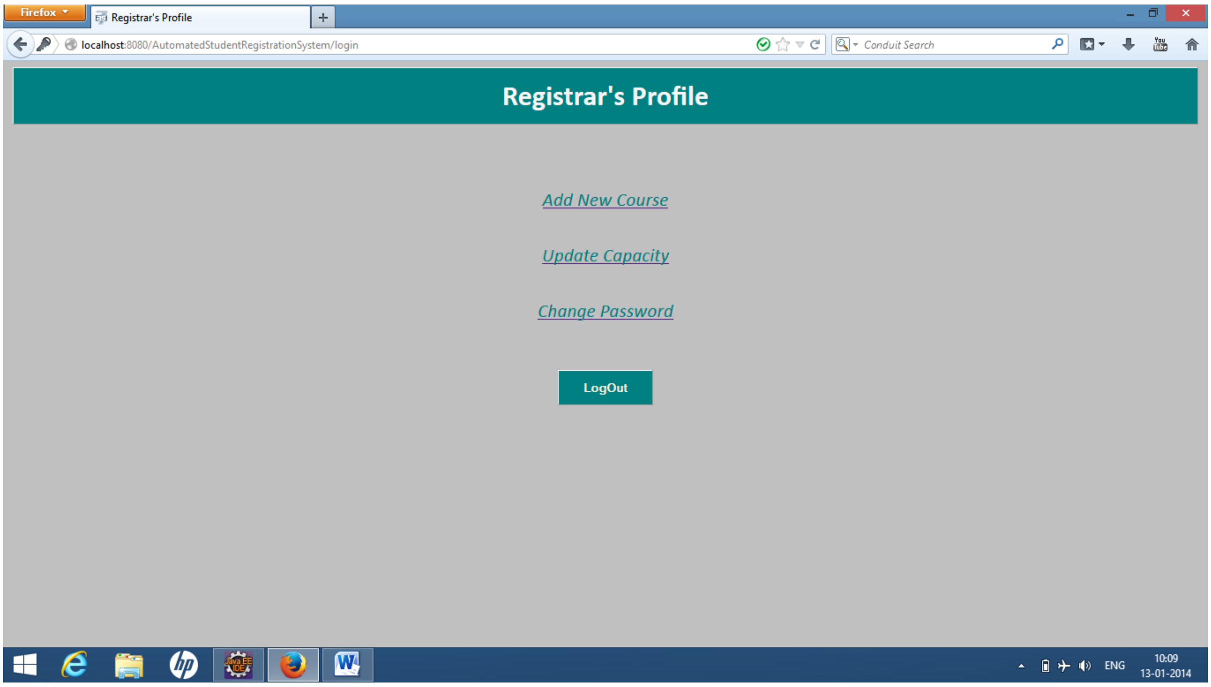
Registrar’s login


localhost:8080/AutomatedStudentRegistrationSystem/LogIn.jsp

Login Page

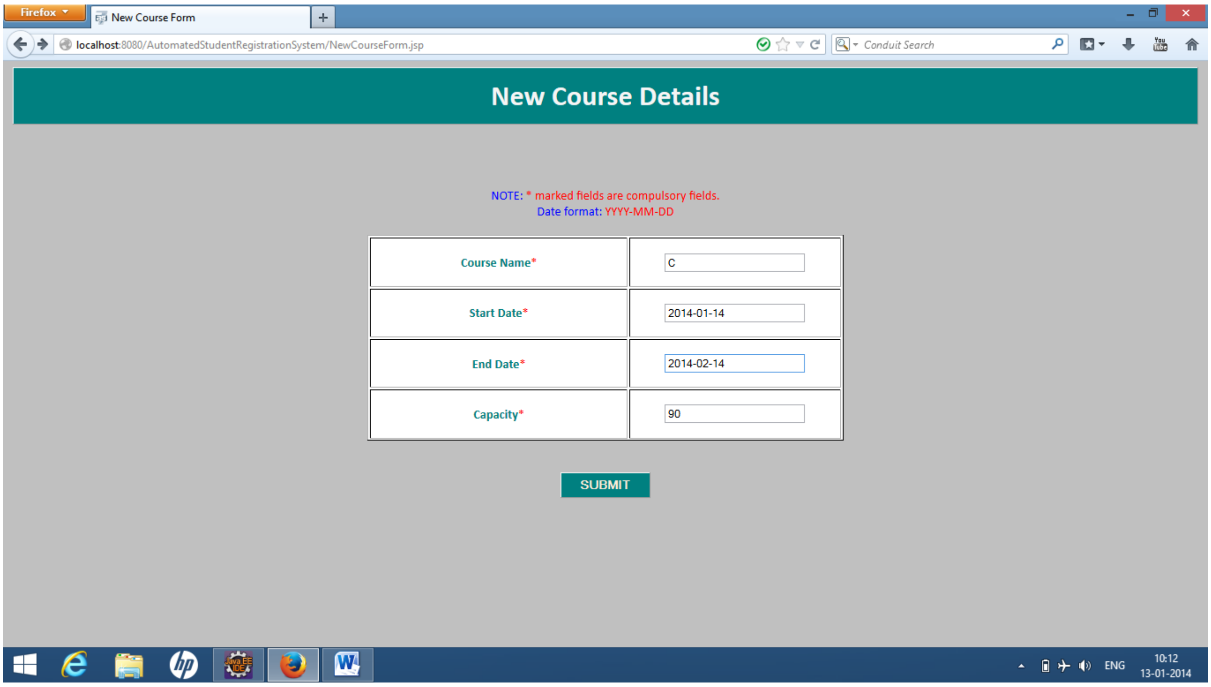
Email Id	aallen@gmail.com
Password	••••••••
LogIn As	Registrar

Registrar’s profile


localhost:8080/AutomatedStudentRegistrationSystem/login

Registrar's Profile

Add new course



New Course Form
localhost:8080/AutomatedStudentRegistrationSystem/NewCourseForm.jsp
Conduit Search
10:12
13-01-2014

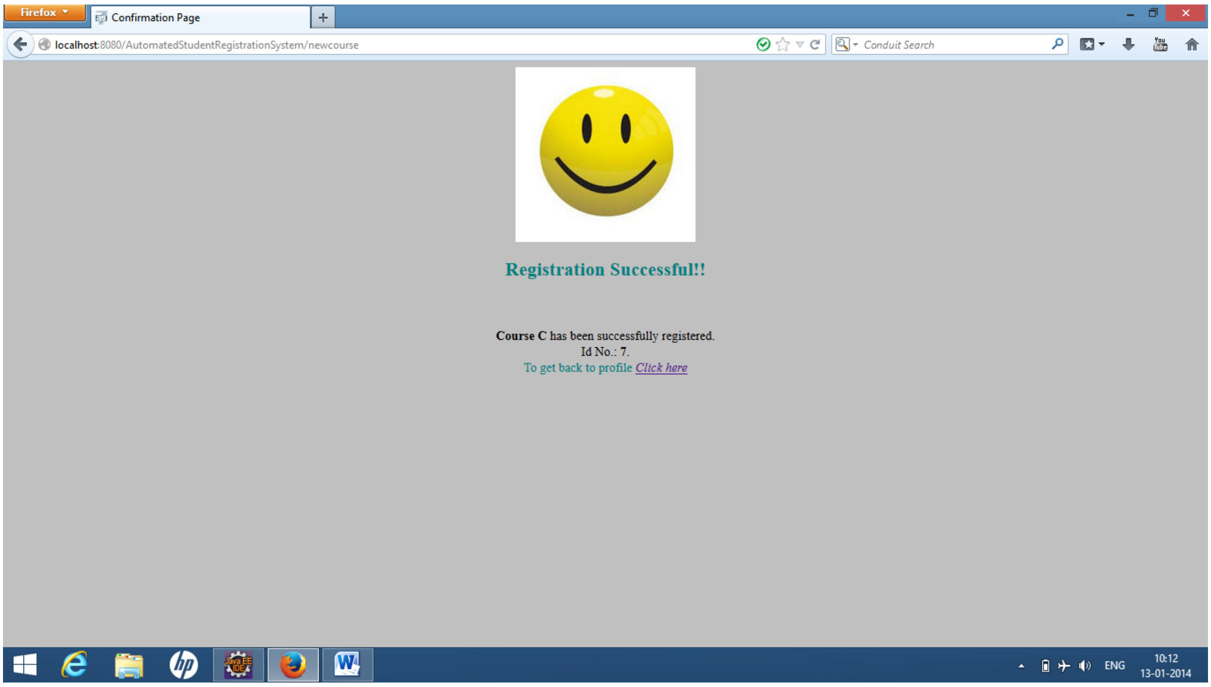
New Course Details

NOTE: * marked fields are compulsory fields.
Date format: YYYY-MM-DD

Course Name*	C
Start Date*	2014-01-14
End Date*	2014-02-14
Capacity*	90

SUBMIT

Course addition confirmation page

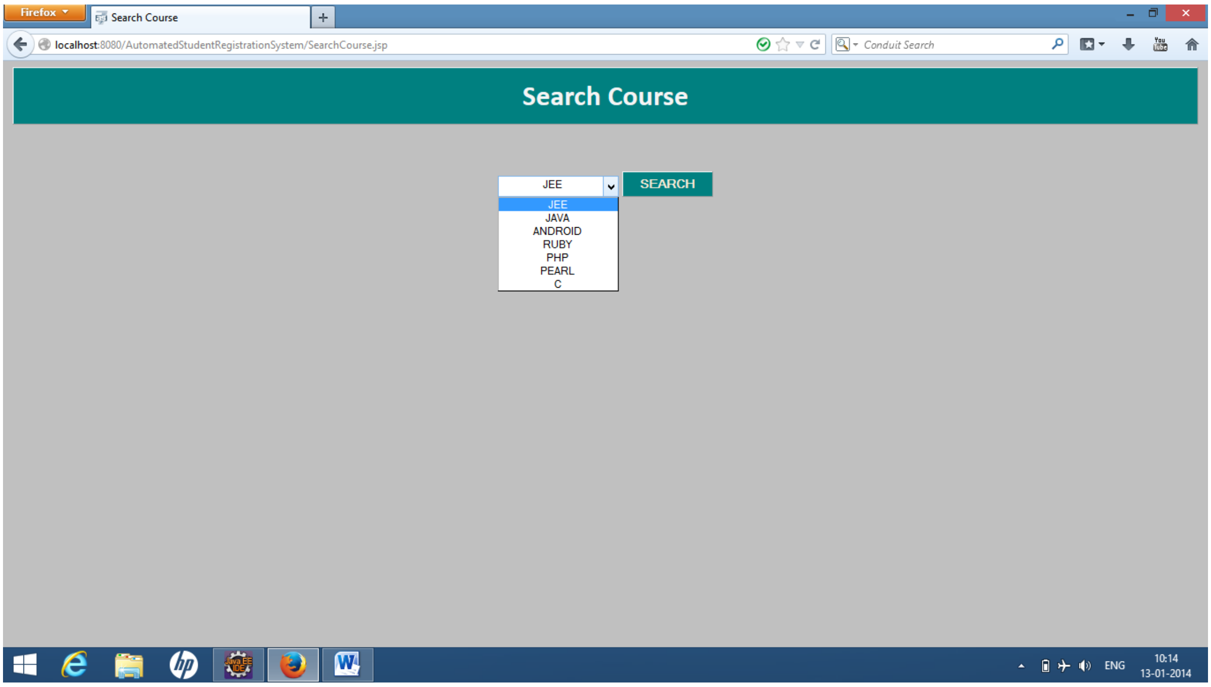


Registration Successful!!

Course C has been successfully registered.
Id No.: 7.
To get back to profile *Click here*

<Back to registrar's profile>

Update Course



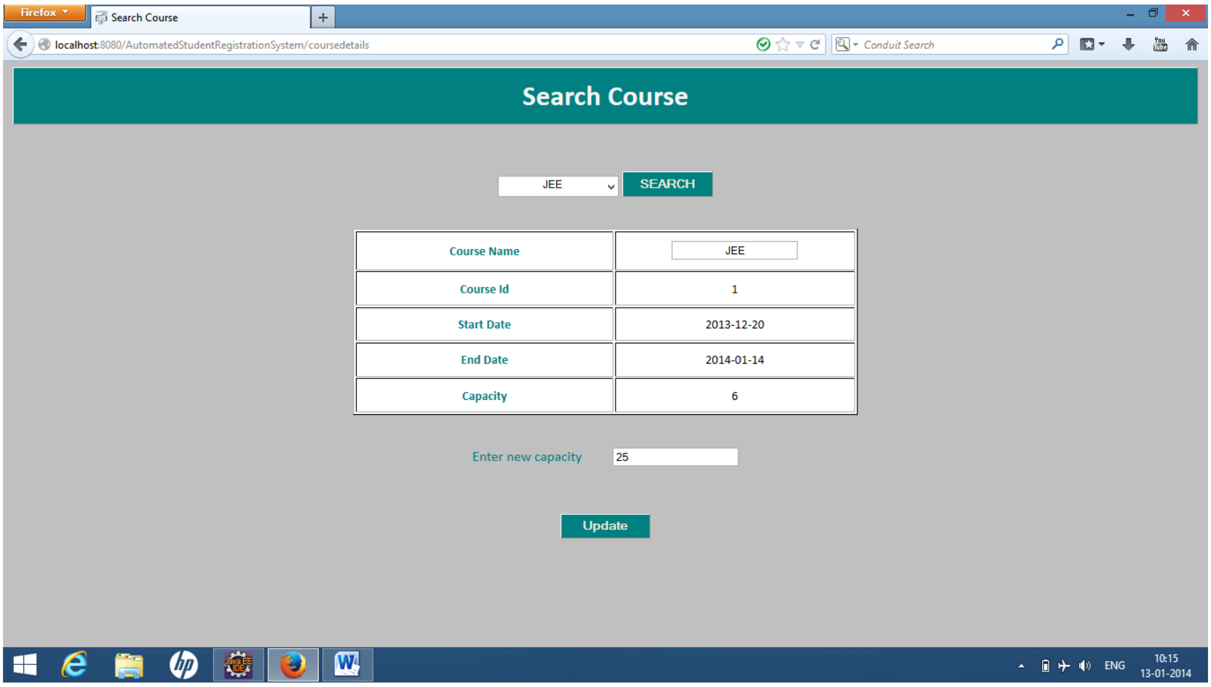
Firefox
Search Course
localhost:8080/AutomatedStudentRegistrationSystem/SearchCourse.jsp
Conduit Search

Search Course

JEE SEARCH

JEE
JAVA
ANDROID
RUBY
PHP
PEARL
C

Search course page



Firefox
Search Course
localhost:8080/AutomatedStudentRegistrationSystem/coursedetails
Conduit Search

Search Course

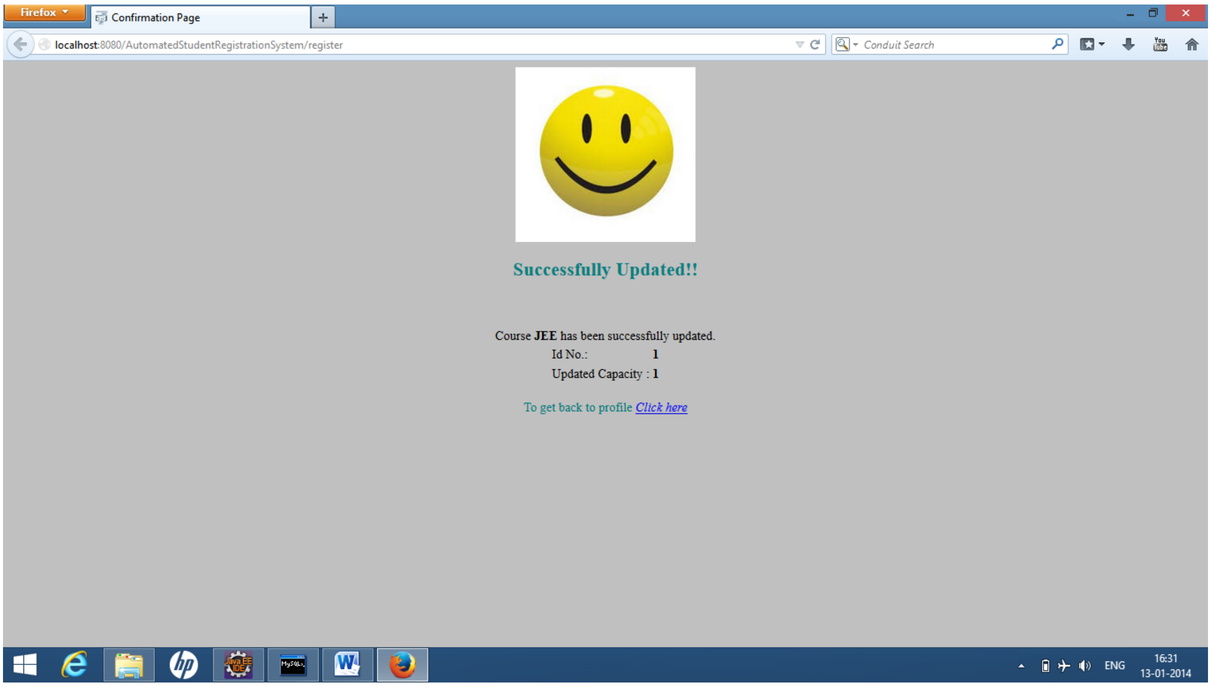
JEE SEARCH

Course Name	JEE
Course Id	1
Start Date	2013-12-20
End Date	2014-01-14
Capacity	6

Enter new capacity 25

Update

Course updated confirmation page



Successfully Updated!!

Course **JEE** has been successfully updated.
Id No.: **1**
Updated Capacity : **1**

To get back to profile *Click here*

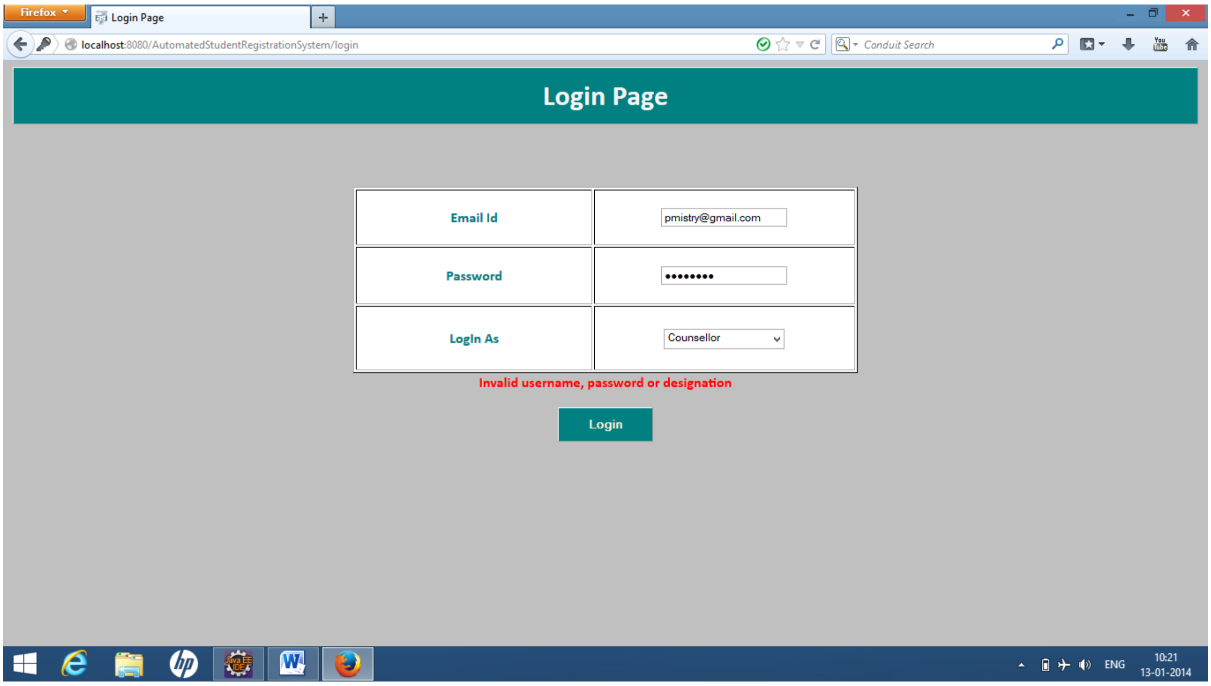
<Back to profile>

Change password works same as that for counsellor.
On clicking logout from profile, user reaches the login page.

<End of screenshots>

Smart Features:

1. Invalid Login.



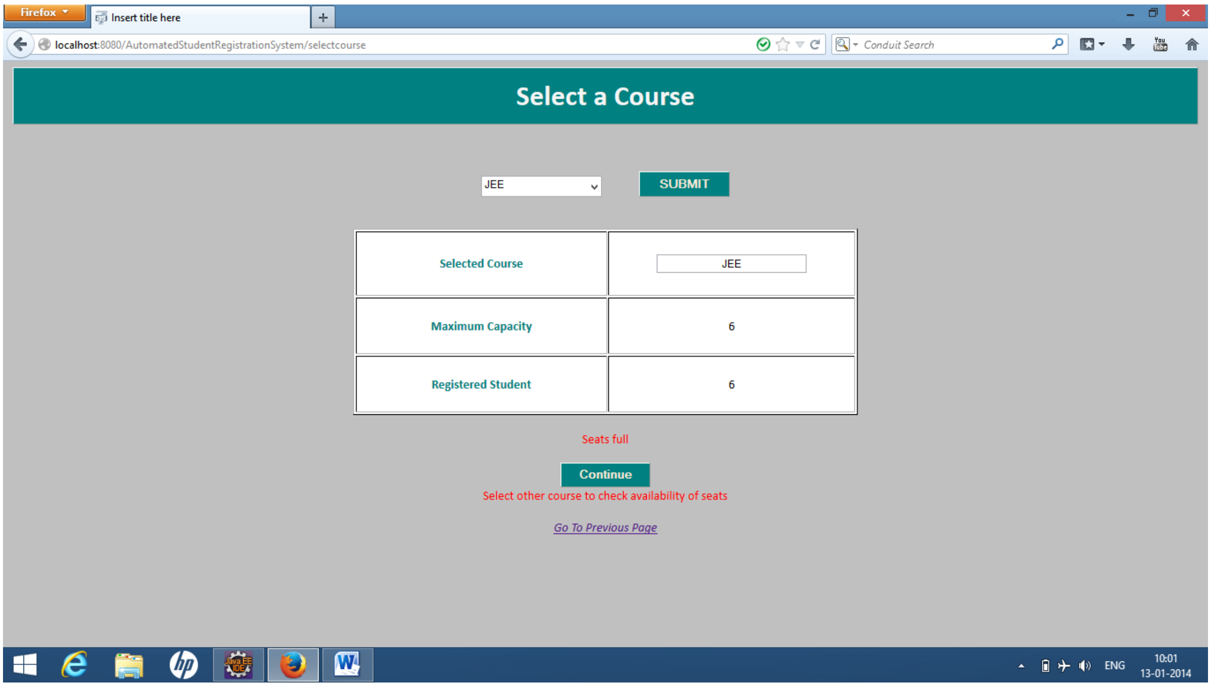
Login Page

Email Id	pmistry@gmail.com
Password	••••••••
Login As	Counsellor

Invalid username, password or designation

Login

2. When seats for a course have filled up and there is a request for registration.



Firefox
Insert title here
localhost:8080/AutomatedStudentRegistrationSystem/selectcourse
Conduit Search
ENG
10:01
13-01-2014

Select a Course

JEE SUBMIT

Selected Course	JEE
Maximum Capacity	6
Registered Student	6

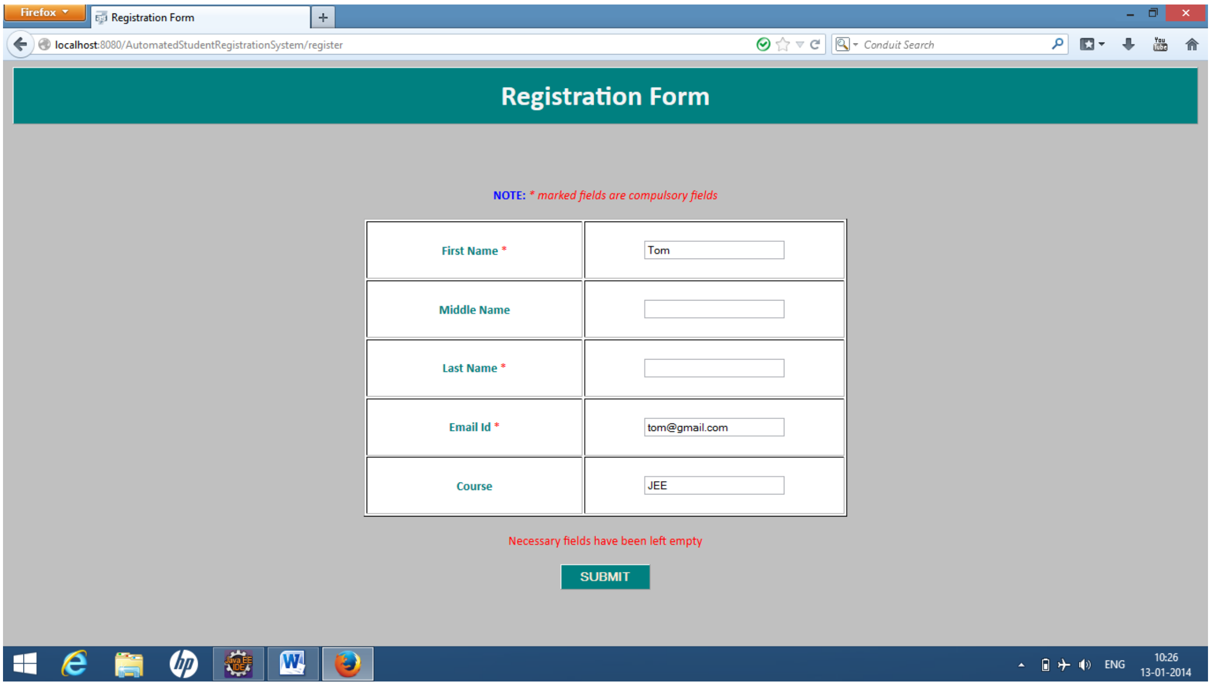
Seats full

Continue

Select other course to check availability of seats

Go To Previous Page

3. When necessary fields are left empty.



Registration Form

NOTE: * marked fields are compulsory fields

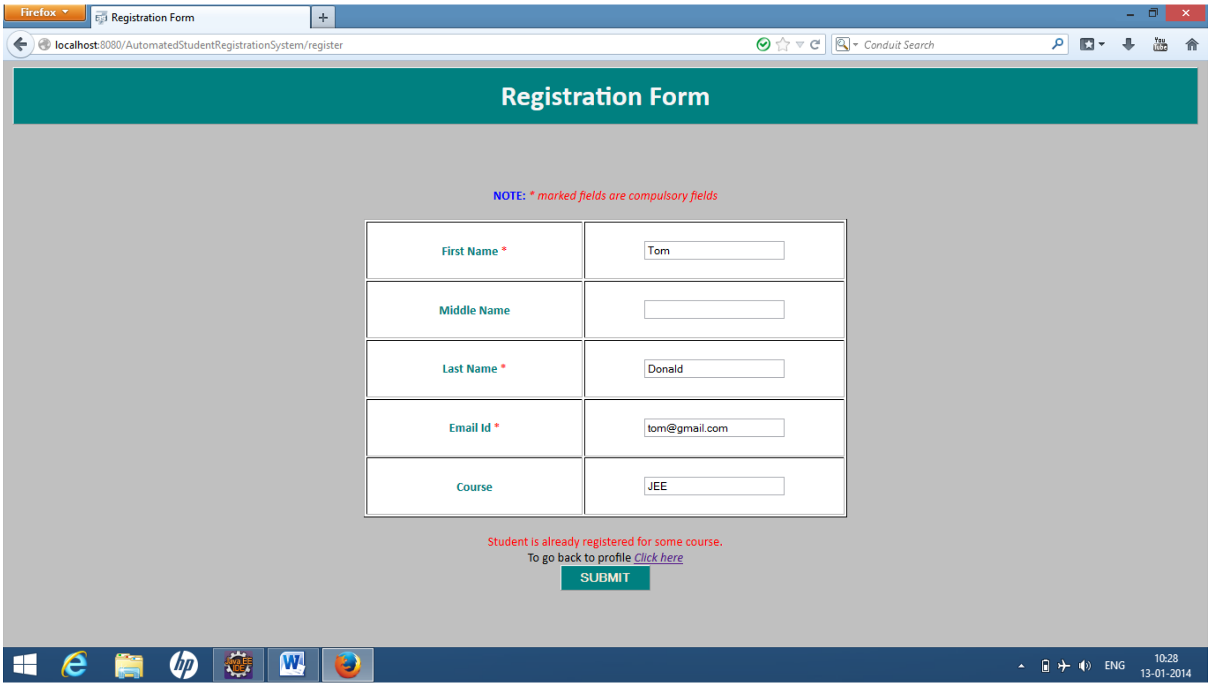
First Name *	Tom
Middle Name	
Last Name *	
Email Id *	tom@gmail.com
Course	JEE

Necessary fields have been left empty

SUBMIT

NOTE: The course field cannot be modified on this page.

4. If a student already registered for a course tries to register again.



Registration Form

NOTE: * marked fields are compulsory fields

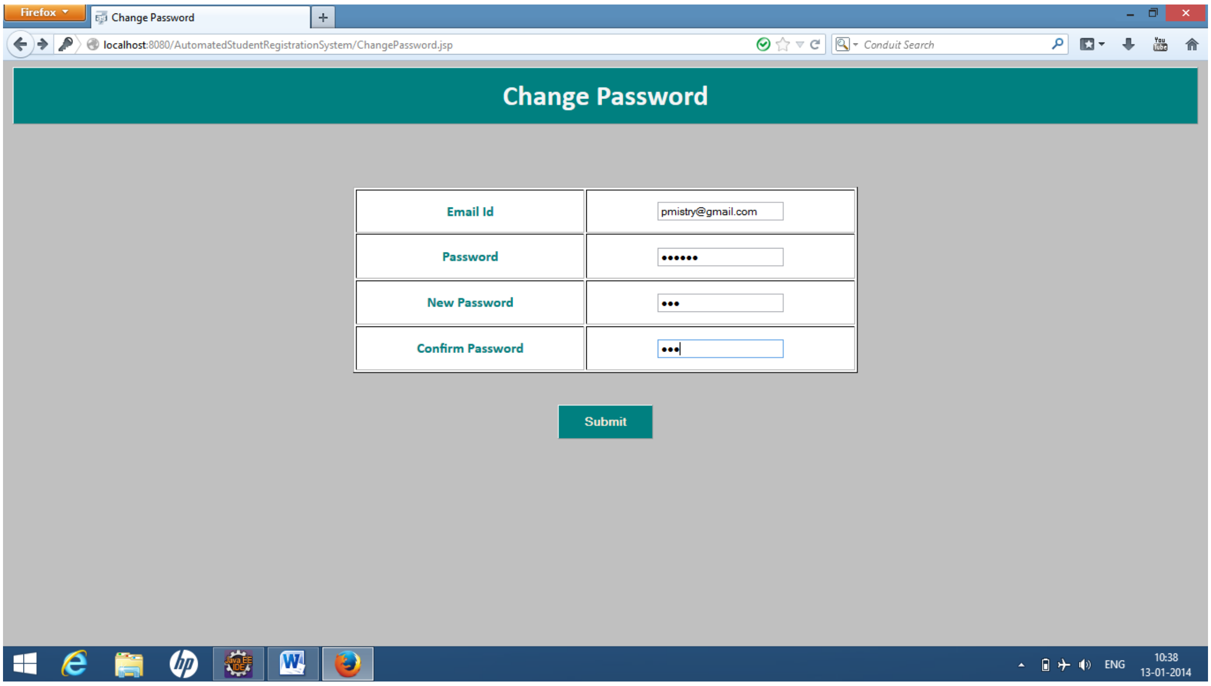
First Name *	Tom
Middle Name	
Last Name *	Donald
Email Id *	tom@gmail.com
Course	JEE

Student is already registered for some course.

To go back to profile Click here

SUBMIT

5. While changing the password.
    - If password length is not 6-10 characters long.



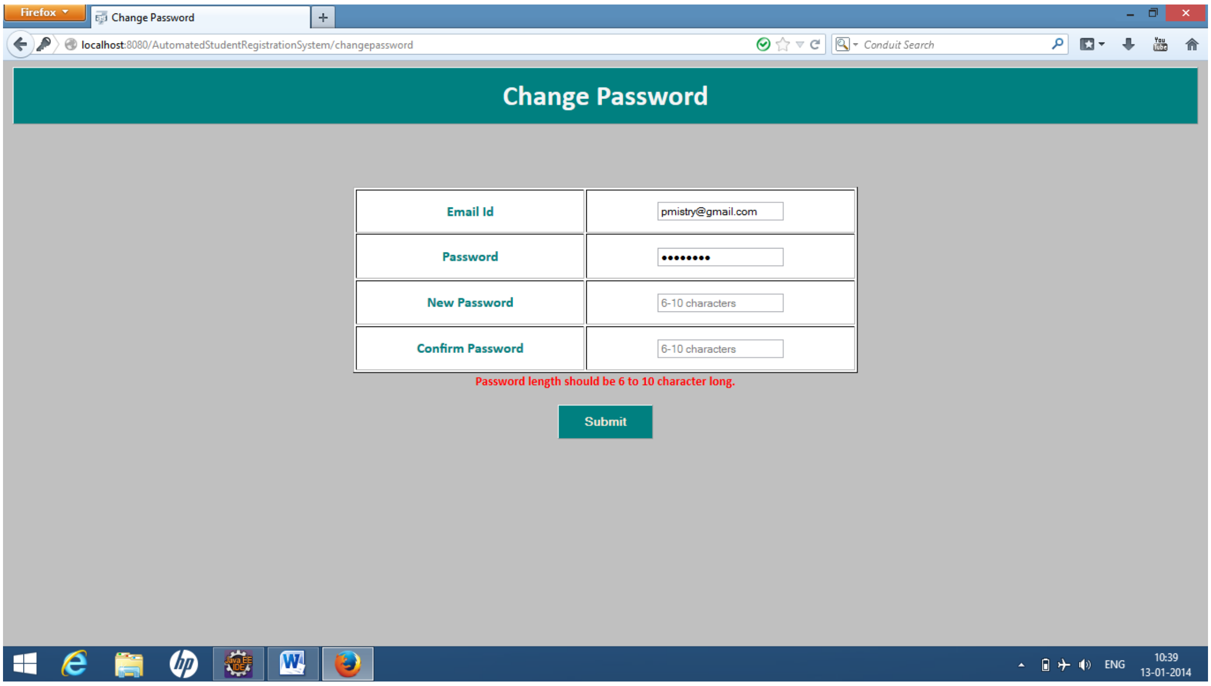
Firefox
Change Password
localhost:8080/AutomatedStudentRegistrationSystem/ChangePassword.jsp
Conduit Search
10:38
13-01-2014

Change Password

Email Id	pmistry@gmail.com
Password	••••••
New Password	•••
Confirm Password	•••

Submit

Then,



Firefox
Change Password
localhost:8080/AutomatedStudentRegistrationSystem/changepassword
Conduit Search
10:39
13-01-2014

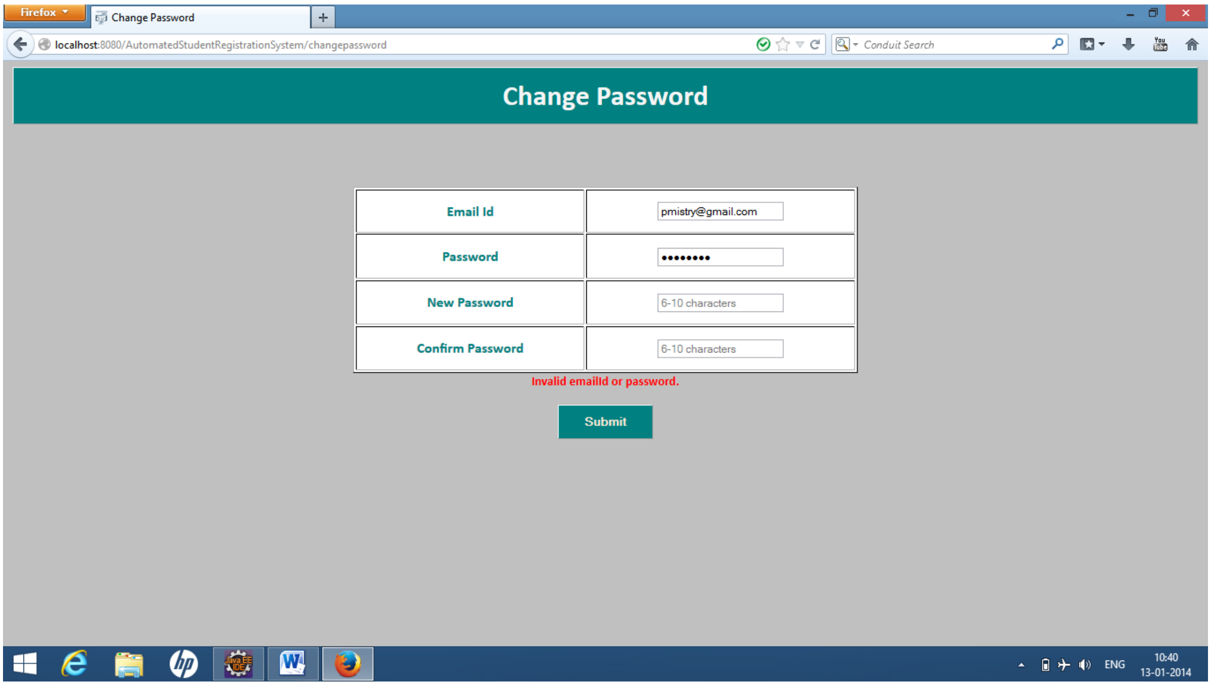
Change Password

Email Id	pmistry@gmail.com
Password	••••••••
New Password	6-10 characters
Confirm Password	6-10 characters

Password length should be 6 to 10 character long.

Submit

- User inputs invalid username or password.


Firefox
Change Password
localhost:8080/AutomatedStudentRegistrationSystem/changepassword
Conduit Search
10:40
13-01-2014

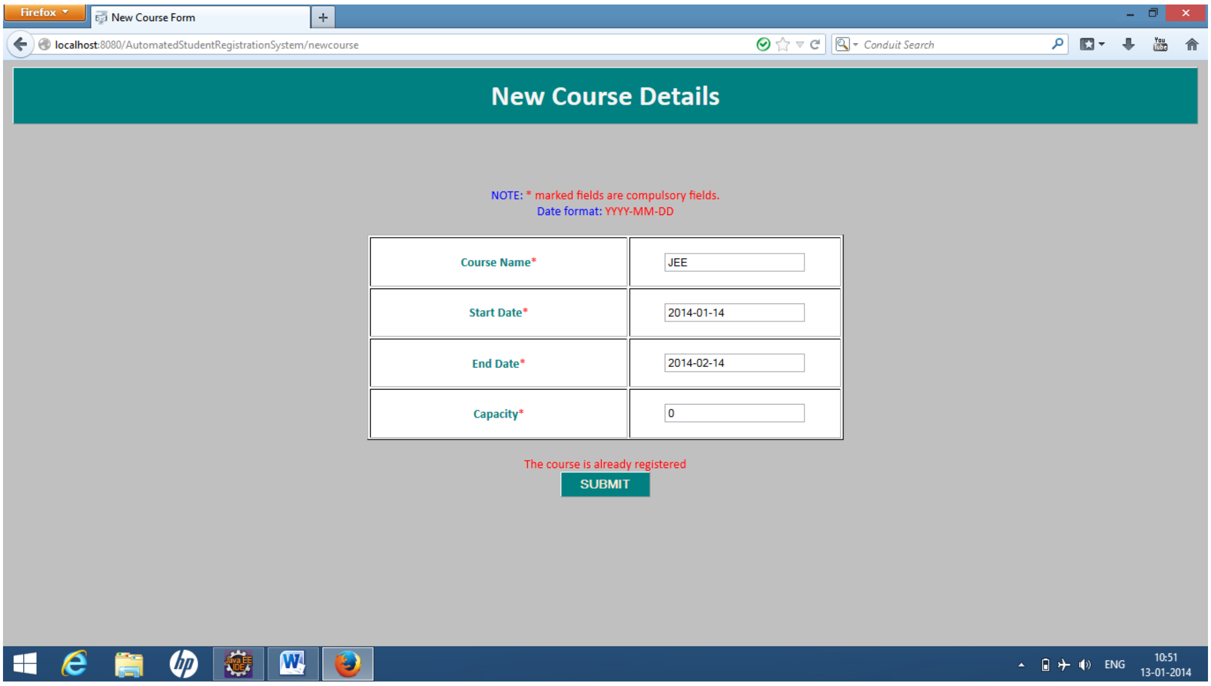
Change Password

Email Id	pmistry@gmail.com
Password	••••••••
New Password	6-10 characters
Confirm Password	6-10 characters

Invalid emailId or password.

Submit

6. When registrar adds new course, if
   - Course with the same name is already registered, then



New Course Details

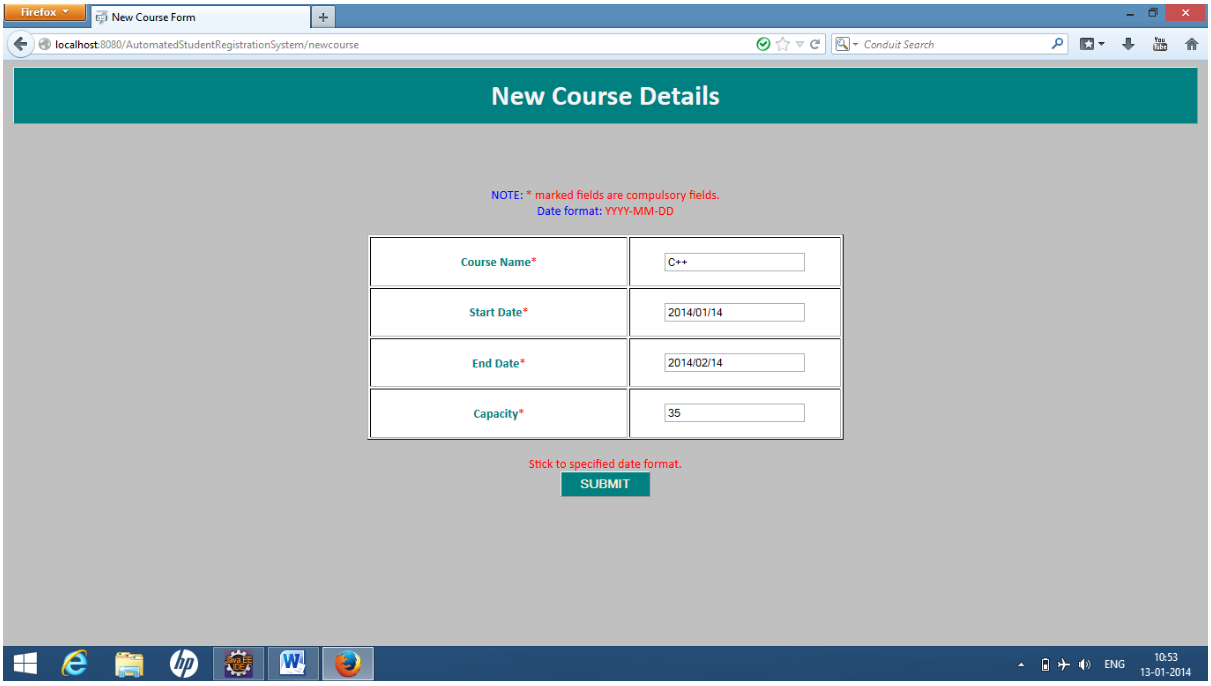
NOTE: * marked fields are compulsory fields.
Date format: YYYY-MM-DD

Course Name*	JEE
Start Date*	2014-01-14
End Date*	2014-02-14
Capacity*	0

The course is already registered

SUBMIT

- Date is entered in any arbitrary format



New Course Details

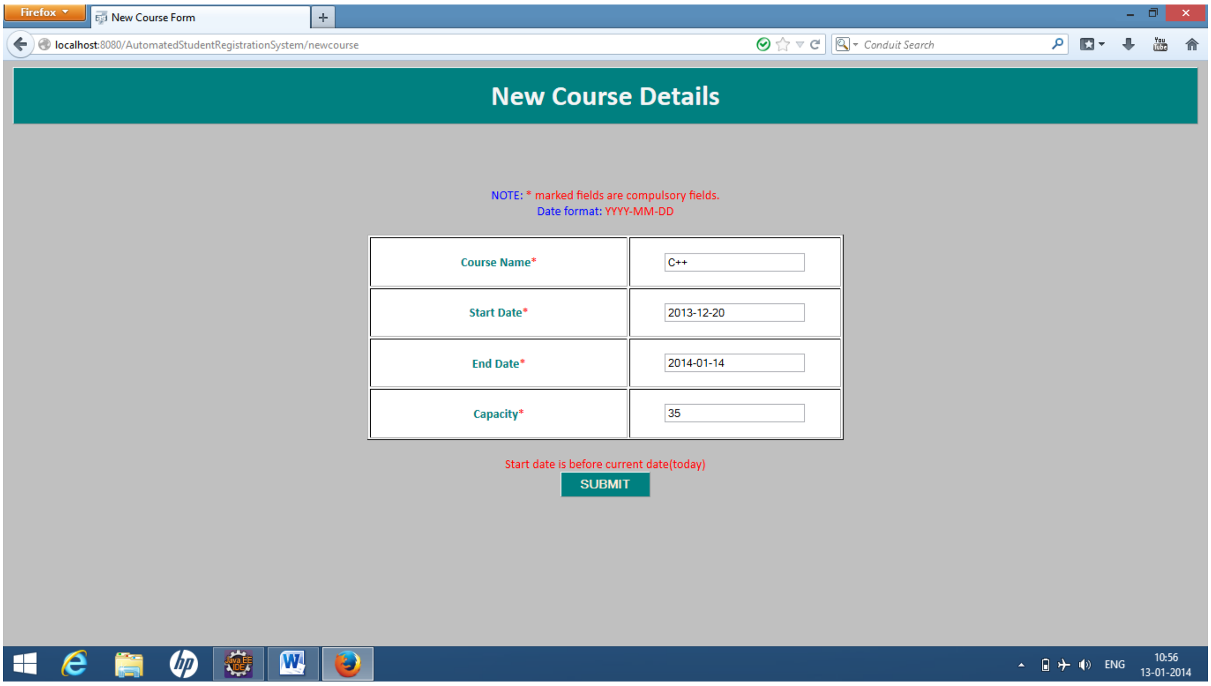
NOTE: * marked fields are compulsory fields.
Date format: YYYY-MM-DD

Course Name*	C++
Start Date*	2014/01/14
End Date*	2014/02/14
Capacity*	35

Stick to specified date format.

SUBMIT

- Start date is earlier than current date.



New Course Details

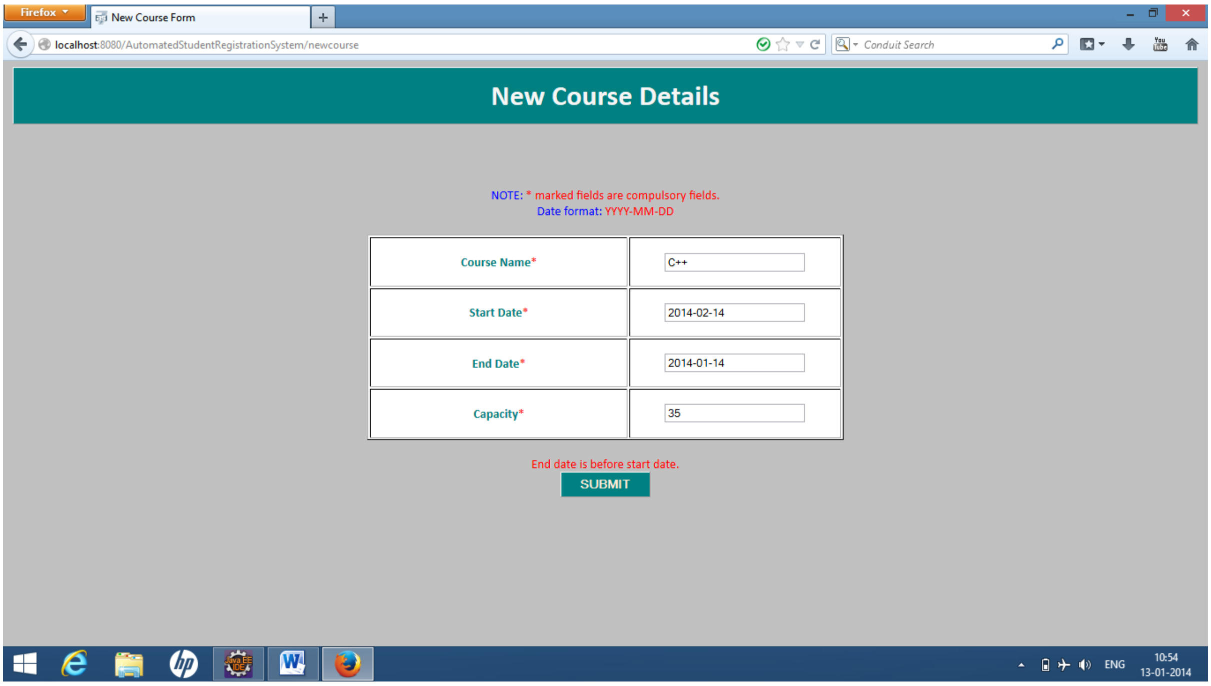
NOTE: * marked fields are compulsory fields.
Date format: YYYY-MM-DD

Course Name*	C++
Start Date*	2013-12-20
End Date*	2014-01-14
Capacity*	35

Start date is before current date(today)

SUBMIT

- End date is before start date.



New Course Details

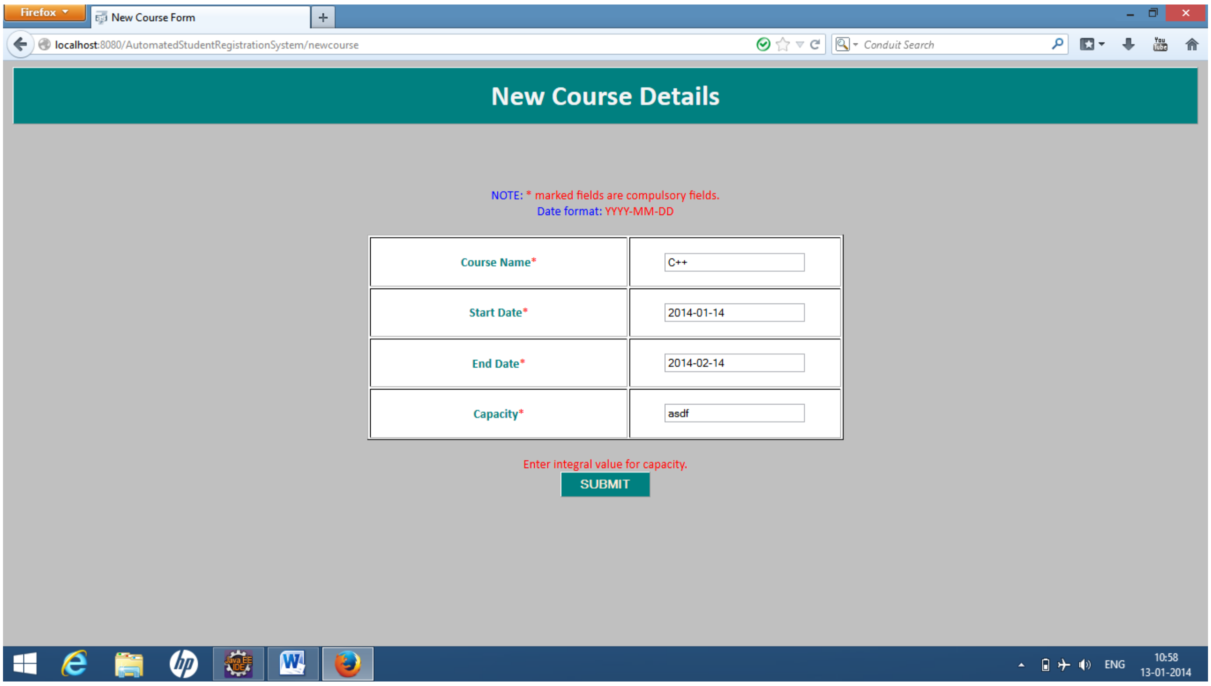
NOTE: * marked fields are compulsory fields.
Date format: YYYY-MM-DD

Course Name*	C++
Start Date*	2014-02-14
End Date*	2014-01-14
Capacity*	35

End date is before start date.

SUBMIT

- Capacity is assigned any value that is not a positive integer.



New Course Details

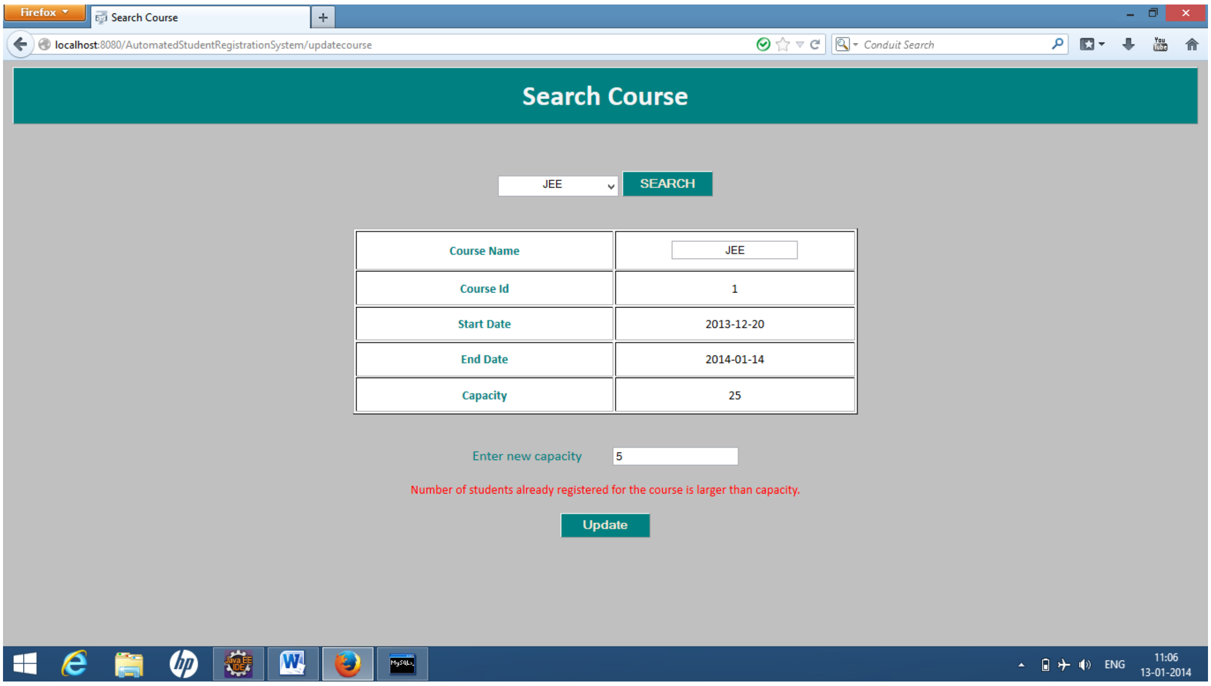
NOTE: * marked fields are compulsory fields.
Date format: YYYY-MM-DD

Course Name*	C++
Start Date*	2014-01-14
End Date*	2014-02-14
Capacity*	asdf

Enter integral value for capacity.

SUBMIT

7. While updating course capacity if the registrar enters capacity which is lesser than the number of students already registered for the course, then



Firefox
Search Course
localhost:8080/AutomatedStudentRegistrationSystem/updatecourse
Conduit Search

Search Course

JEE SEARCH

Course Name	JEE
Course Id	1
Start Date	2013-12-20
End Date	2014-01-14
Capacity	25

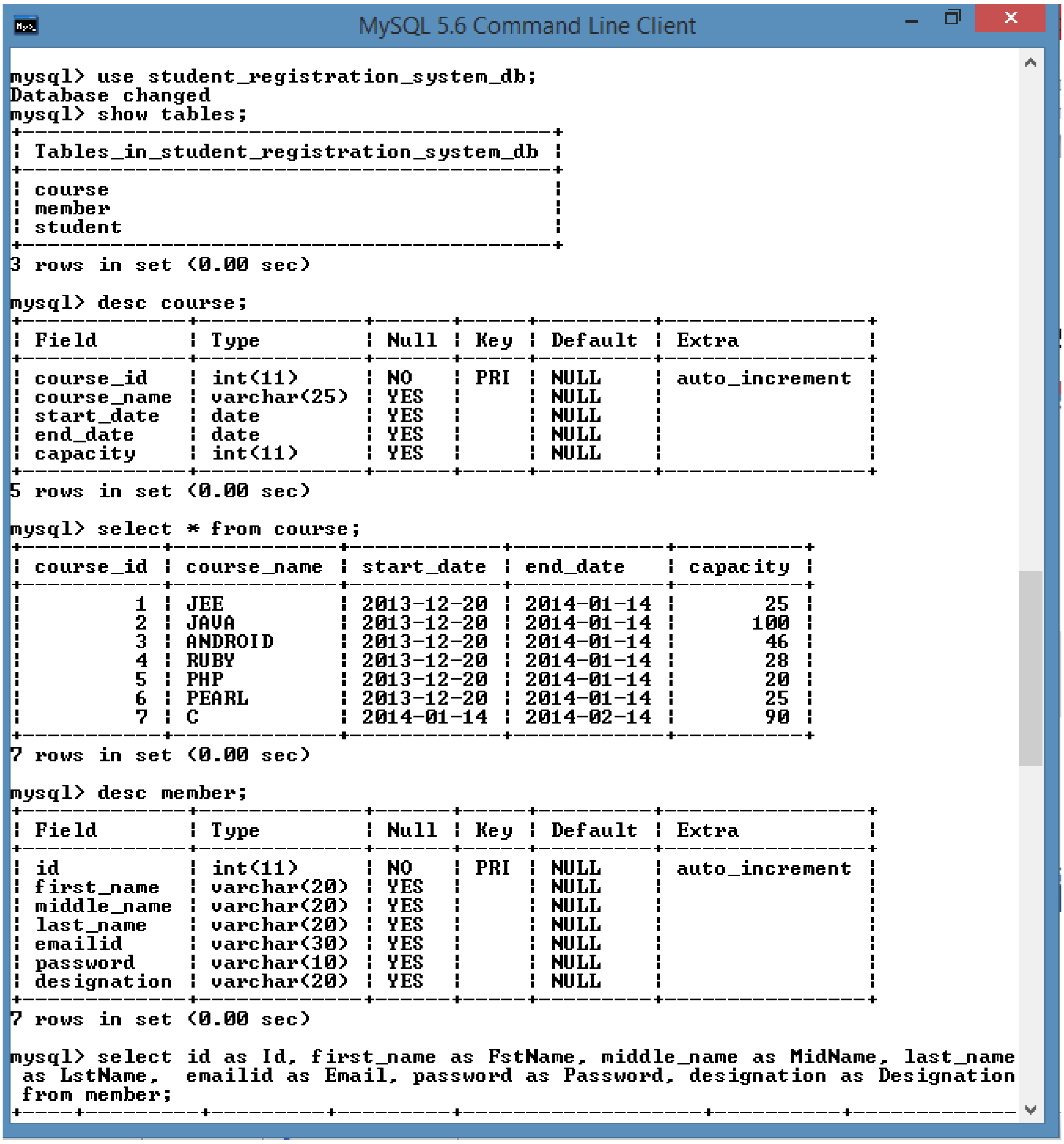
Enter new capacity 5

Number of students already registered for the course is larger than capacity.

Update

<End of Smart Features>

Database screenshots:



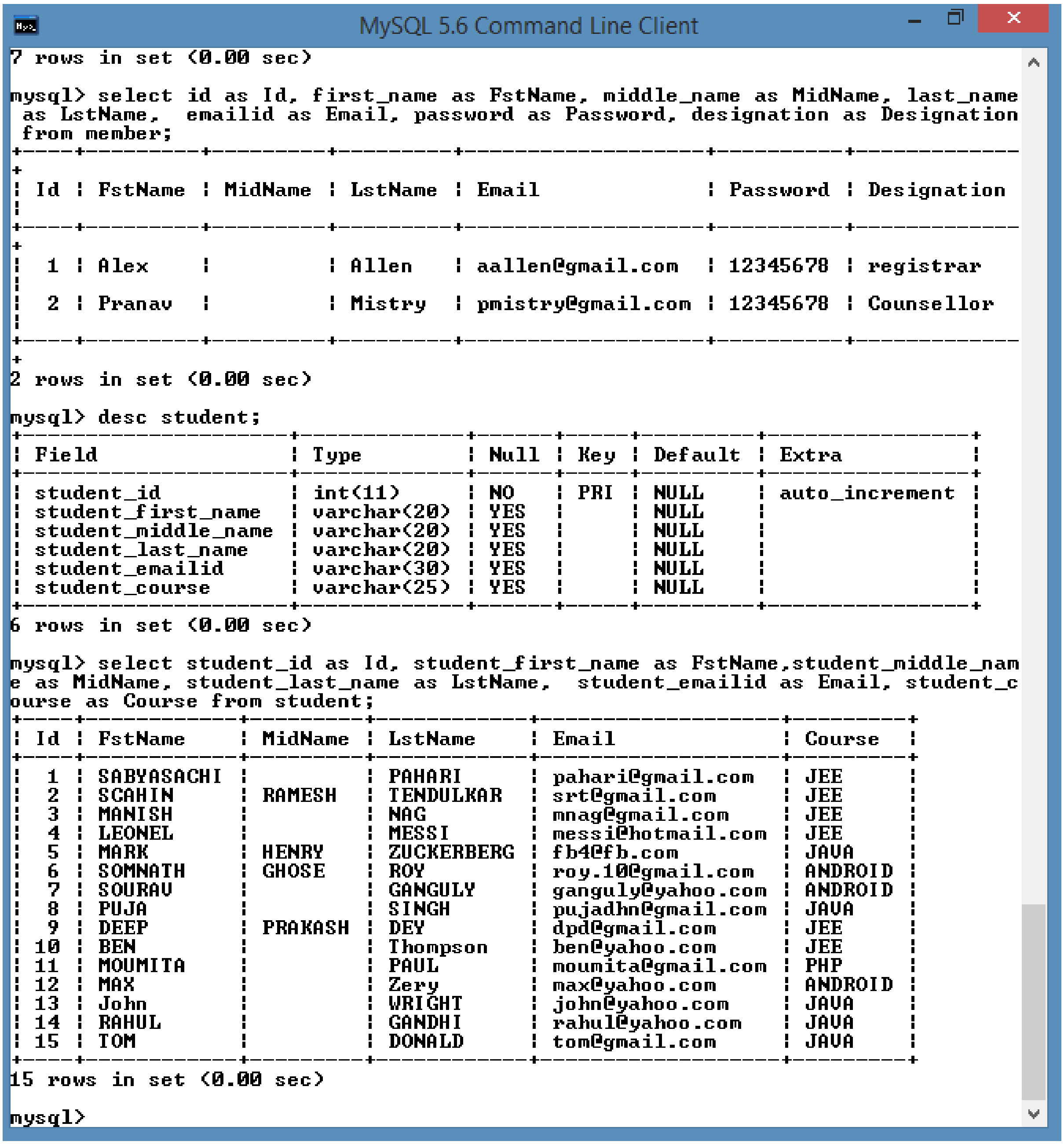
```
mysql> use student_registration_system_db;
Database changed
mysql> show tables;
+-------------------------------------------+
| Tables_in_student_registration_system_db |
+-------------------------------------------+
| course                                    |
| member                                    |
| student                                   |
+-------------------------------------------+
3 rows in set (0.00 sec)

mysql> desc course;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| course_id   | int(11)     | NO   | PRI | NULL    | auto_increment |
| course_name | varchar(25) | YES  |     | NULL    |                |
| start_date  | date        | YES  |     | NULL    |                |
| end_date    | date        | YES  |     | NULL    |                |
| capacity    | int(11)     | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> select * from course;
+-----------+-------------+------------+------------+----------+
| course_id | course_name | start_date | end_date   | capacity |
+-----------+-------------+------------+------------+----------+
|         1 | JEE         | 2013-12-20 | 2014-01-14 |       25 |
|         2 | JAVA        | 2013-12-20 | 2014-01-14 |      100 |
|         3 | ANDROID     | 2013-12-20 | 2014-01-14 |       46 |
|         4 | RUBY        | 2013-12-20 | 2014-01-14 |       28 |
|         5 | PHP         | 2013-12-20 | 2014-01-14 |       20 |
|         6 | PEARL       | 2013-12-20 | 2014-01-14 |       25 |
|         7 | C           | 2014-01-14 | 2014-02-14 |       90 |
+-----------+-------------+------------+------------+----------+
7 rows in set (0.00 sec)

mysql> desc member;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| id          | int(11)     | NO   | PRI | NULL    | auto_increment |
| first_name  | varchar(20) | YES  |     | NULL    |                |
| middle_name | varchar(20) | YES  |     | NULL    |                |
| last_name   | varchar(20) | YES  |     | NULL    |                |
| emailid     | varchar(30) | YES  |     | NULL    |                |
| password    | varchar(10) | YES  |     | NULL    |                |
| designation | varchar(20) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)

mysql> select id as Id, first_name as FstName, middle_name as MidName, last_name
 as LstName,  emailid as Email, password as Password, designation as Designation
 from member;
+----+----------+---------+---------+--------------------+----------+-----------
```



```
7 rows in set (0.00 sec)

mysql> select id as Id, first_name as FstName, middle_name as MidName, last_name
 as LstName,  emailid as Email, password as Password, designation as Designation
 from member;
+----+---------+---------+---------+-------------------+----------+-------------
+
| Id | FstName | MidName | LstName | Email             | Password | Designation
|
+----+---------+---------+---------+-------------------+----------+-------------
+
|  1 | Alex    |         | Allen   | aallen@gmail.com  | 12345678 | registrar
|
|  2 | Pranav  |         | Mistry  | pmistry@gmail.com | 12345678 | Counsellor
|
+----+---------+---------+---------+-------------------+----------+-------------
+
2 rows in set (0.00 sec)

mysql> desc student;
+---------------------+-------------+------+-----+---------+----------------+
| Field               | Type        | Null | Key | Default | Extra          |
+---------------------+-------------+------+-----+---------+----------------+
| student_id          | int(11)     | NO   | PRI | NULL    | auto_increment |
| student_first_name  | varchar(20) | YES  |     | NULL    |                |
| student_middle_name | varchar(20) | YES  |     | NULL    |                |
| student_last_name   | varchar(20) | YES  |     | NULL    |                |
| student_emailid     | varchar(30) | YES  |     | NULL    |                |
| student_course      | varchar(25) | YES  |     | NULL    |                |
+---------------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> select student_id as Id, student_first_name as FstName,student_middle_nam
e as MidName, student_last_name as LstName,  student_emailid as Email, student_c
ourse as Course from student;
+----+------------+---------+------------+-------------------+---------+
| Id | FstName    | MidName | LstName    | Email             | Course  |
+----+------------+---------+------------+-------------------+---------+
|  1 | SABYASACHI |         | PAHARI     | pahari@gmail.com  | JEE     |
|  2 | SCAHIN     | RAMESH  | TENDULKAR  | srt@gmail.com     | JEE     |
|  3 | MANISH     |         | NAG        | mnag@gmail.com    | JEE     |
|  4 | LEONEL     |         | MESSI      | messi@hotmail.com | JEE     |
|  5 | MARK       | HENRY   | ZUCKERBERG | fb4@fb.com        | JAVA    |
|  6 | SOMNATH    | GHOSE   | ROY        | roy.10@gmail.com  | ANDROID |
|  7 | SOURAV     |         | GANGULY    | ganguly@yahoo.com | ANDROID |
|  8 | PUJA       |         | SINGH      | pujadhn@gmail.com | JAVA    |
|  9 | DEEP       | PRAKASH | DEY        | dpd@gmail.com     | JEE     |
| 10 | BEN        |         | Thompson   | ben@yahoo.com     | JEE     |
| 11 | MOUMITA    |         | PAUL       | moumita@gmail.com | PHP     |
| 12 | MAX        |         | Zery       | max@yahoo.com     | ANDROID |
| 13 | John       |         | WRIGHT     | john@yahoo.com    | JAVA    |
| 14 | RAHUL      |         | GANDHI     | rahul@yahoo.com   | JAVA    |
| 15 | TOM        |         | DONALD     | tom@gmail.com     | JAVA    |
+----+------------+---------+------------+-------------------+---------+
15 rows in set (0.00 sec)

mysql>
```

Future Scope of Improvements

As we have abided by the rules and regulations of the project abstract given by Globsyn, we can further improve on various fields like :

- A registered student on successful registration can get a confirmation message through email.
- There will be a provision allotted to the students where they can email the registrar or counsellor their queries which they face during their registration process.
- To make our software more robust, counsellors will also get validated prior to registering a student.

Code

All Beans:

1. **Class name: <u>Course.java</u>**

Comments required

```
package com.project.beans;

public class Course {

        private int course_id;
        private String course_name;
        private String start_date;
        private String end_date;
        private int capacity;

        public int getCourse_id() {
                return course_id;
        }

        public void setCourse_id(int course_id) {
                this.course_id = course_id;
        }

        public String getCourse_name() {
                return course_name;
        }

        public void setCourse_name(String course_name) {
                this.course_name = course_name;
        }

        public String getStart_date() {
                return start_date;
        }
        public void setStart_date(String start_date) {
                this.start_date = start_date;
        }

        public String getEnd_date() {
                return end_date;
        }

        public void setEnd_date(String end_date) {
                this.end_date = end_date;
        }

        public int getCapacity() {
                return capacity;
        }

        public void setCapacity(int capacity) {
                this.capacity = capacity;
        }

}
```

2. Class name: Student.java

```
package com.project.beans;

public class Student {

        private int student_id;
        private String student_first_name;
        private String student_middle_name;
        private String student_last_name;
        private String student_emailid;
        private String student_course;
        public int getStudent_id() {
                return student_id;
        }
        public void setStudent_id(int student_id) {
                this.student_id = student_id;
        }
        public String getStudent_first_name() {
                return student_first_name;
        }
        public void setStudent_first_name(String student_first_name) {
                this.student_first_name = student_first_name;
        }
        public String getStudent_middle_name() {
                return student_middle_name;
        }
        public void setStudent_middle_name(String student_middle_name) {
                this.student_middle_name = student_middle_name;
        }
        public String getStudent_last_name() {
                return student_last_name;
        }
        public void setStudent_last_name(String student_last_name) {
                this.student_last_name = student_last_name;
        }
        public String getStudent_emailid() {
                return student_emailid;
        }
        public void setStudent_emailid(String student_emailid) {
                this.student_emailid = student_emailid;
        }
        public String getStudent_course() {
                return student_course;
        }
        public void setStudent_course(String student_course) {
                this.student_course = student_course;
        }

}
```

Database classes:

1. **Class name: DBUtils.java**

```
package com.project.db;

import java.sql.*;

public class DBUtils {
    static Connection conn = null;
    static Statement st = null;
    static PreparedStatement pst = null;
    static ResultSet rs = null;

    public static Connection connectToDb() {
        try {
            String url = 
"jdbc:mysql://localhost:3306/student_registration_system_db";
            String username = "root";
            String password = "root123";

            Class.forName("com.mysql.jdbc.Driver");

            conn = DriverManager.getConnection(url, username, password);

        } catch (ClassNotFoundException e) {
            System.out.println("Something is wrong with driver");

        } catch (SQLException e) {
            System.out.println("SQL Exception... check ur query");
        }
        return conn;
    }
    public static Statement getStatement(Connection con) {
        try {
            st = con.createStatement();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return st;
    }

    public static PreparedStatement getPreparedStatement(Connection con,
            String query) {
        try {
            pst = con.prepareStatement(query);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return pst;
    }

    public static void closeConnections() {
        try {
            if (rs != null)
                rs.close();
            if (st != null)
                st.close();
            if (pst != null)
                pst.close();
            if (conn != null)
                conn.close();
        } catch (SQLException e) {
            System.out.println("Connection not closed properly...");
        }
    }
}
```

All Servlets:

1. **Class name: LoginServlet.java**

```
package com.project.servlets;

import java.io.IOException;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

import com.project.utils.*;


@WebServlet("/login")
public class LoginServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse 
response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse 
response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		
		String emailId = request.getParameter("email");
		String password = request.getParameter("pswd");
```

```
        String designation =request.getParameter("designation");
        MemberUtils member_util = new MemberUtilsImpl();
        if(member_util.authenticate(emailId, password, designation)){
            String url="CounsellorProfile.jsp";
            if(designation.equals("Registrar"))
                url="RegistrarProfile.jsp";
            RequestDispatcher dispatcher = 
request.getRequestDispatcher(url);
            dispatcher.forward(request, response);
        }
        else{
            request.setAttribute("loginStatus", false);
            RequestDispatcher dispatcher = 
request.getRequestDispatcher("LogIn.jsp");
            dispatcher.forward(request, response);
        }

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse 
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse 
response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request,response);
    }

}
```

2. Class name: StudentRegistrationServlet.java

```
package com.project.servlets;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.project.beans.Student;
import com.project.utils.StudentUtilsImpl;

/**
 * Servlet implementation class StudentRegistrationServlet
 */
@WebServlet("/register")
public class StudentRegistrationServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public StudentRegistrationServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse 
response)
       */
      protected void doGet(HttpServletRequest request, HttpServletResponse 
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             String email = request.getParameter("emailid");
             String first_name = request.getParameter("first_name");
             String middle_name =request.getParameter("middle_name");
             String last_name = request.getParameter("last_name");
             Student student = new Student();
             student.setStudent_first_name(first_name);
             student.setStudent_middle_name(middle_name);
             student.setStudent_last_name(last_name);
             student.setStudent_emailid(email);
```

```
		if(email.equals("")||first_name.equals("")||last_name.equals(""))
		{
			String course = (String)request.getParameter("course");
			request.setAttribute("course", course);
			request.setAttribute("message", true);
			request.setAttribute("studentobj", student);
			RequestDispatcher dispatcher = 
request.getRequestDispatcher("StudentRegistrationForm.jsp");
			dispatcher.forward(request, response);
		}
		else if(new StudentUtilsImpl().isStudentDuplicate(student)){
			String course = (String)request.getParameter("course");
			request.setAttribute("course", course);
			request.setAttribute("message2", true);
			request.setAttribute("studentobj", student);
			RequestDispatcher dispatcher = 
request.getRequestDispatcher("StudentRegistrationForm.jsp");
			dispatcher.forward(request, response);
		}
		else{
			student.setStudent_course(request.getParameter("course"));
			boolean added = new StudentUtilsImpl().addStudent(student);
			if(added)
			{
				Student student2 = new 
StudentUtilsImpl().searchStudent(email);
				request.setAttribute("name", "Student 
"+student2.getStudent_first_name());
				request.setAttribute("id",student2.getStudent_id());
				RequestDispatcher dispatcher = 
request.getRequestDispatcher("ConfirmationMessage.jsp");
				dispatcher.forward(request, response);
			}
		}
	}

	/**
```

```
       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
      }

}
```

3. Class name: <u>RegistrationFormServlet</u>

```
package com.project.servlets;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class RegistrationFormServlet
 */
@WebServlet("/registrationpage")
public class RegistrationFormServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public RegistrationFormServlet() {
        super();
        // TODO Auto-generated constructor stub
    }
```

```
      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse 
response)
       */
      protected void doGet(HttpServletRequest request, HttpServletResponse 
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             String course=request.getParameter("course");
             request.setAttribute("course", course);
             RequestDispatcher dispatcher = 
request.getRequestDispatcher("StudentRegistrationForm.jsp");
             dispatcher.forward(request,response);
      }

      /**
       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse 
response)
       */
      protected void doPost(HttpServletRequest request, HttpServletResponse 
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             doGet(request, response);
      }

}
```

4. Class name: ChangePasswordServlet.java

```
package com.project.servlets;

import java.io.IOException;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import com.project.utils.*;

@WebServlet("/changepassword")
public class ChangePasswordServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ChangePasswordServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws ServletException,
IOException {
        // TODO Auto-generated method stub

        String emailId = request.getParameter("email");
        String password = request.getParameter("pswd");
        String newPassword = request.getParameter("nwpswd");
        String confirmPassword = request.getParameter("cnfrmnwpswd");

        MemberUtils member_util = new MemberUtilsImpl();

        if (newPassword.length() > 5 && newPassword.length() < 11) {
            if (newPassword.equals(confirmPassword)) {
                if (member_util.changePassword(emailId, password,
newPassword)) {
                    RequestDispatcher dispatcher = request

    .getRequestDispatcher("PasswordChangeConfirmation.jsp");
                    dispatcher.forward(request, response);
                } else {
                    request.setAttribute("message",
                            "Invalid emailId or password.");
                    request.setAttribute("Status", false);
                    RequestDispatcher dispatcher = request
        .getRequestDispatcher("ChangePassword.jsp");
                                dispatcher.forward(request, response);
                          }
                    } else {
                          request.setAttribute("message",
                                      "Confirmation password does not match new
password.");
                          request.setAttribute("Status", false);
                          RequestDispatcher dispatcher = request

        .getRequestDispatcher("ChangePassword.jsp");
                          dispatcher.forward(request, response);
                    }
              } else {
                    request.setAttribute("message",
                                "Password length should be 6 to 10 character
long.");
                    request.setAttribute("Status", false);
                    RequestDispatcher dispatcher = request
                                .getRequestDispatcher("ChangePassword.jsp");
                    dispatcher.forward(request, response);
              }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
         *      response)
         */
        protected void doPost(HttpServletRequest request,
                    HttpServletResponse response) throws ServletException,
IOException {
              // TODO Auto-generated method stub
              doGet(request, response);
        }
}
```

5. **Class name: <u>LogoutServlet.java</u>**

```
package com.project.servlets;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class LogOutServlet
 */
@WebServlet("/logout")
public class LogOutServlet extends HttpServlet {
       private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LogOutServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

       /**
        * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
        */
       protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             RequestDispatcher dispatcher =
request.getRequestDispatcher("LogIn.jsp");
             dispatcher.forward(request , response);

       }
	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		doGet(request, response);
	}

}
```

6. Class name: NewCourseServlet.java

```
package com.project.servlets;

import java.io.IOException;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import com.project.beans.Course;
import com.project.utils.CourseUtilsImpl;
import com.project.utils.DateUtils;

/**
 * Servlet implementation class CreateNewCourseServlet
 */
@WebServlet("/newcourse")
public class NewCourseServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#HttpServlet()
	 */
	public NewCourseServlet() {
		super();
		// TODO Auto-generated constructor stub
	}
```

```
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request,
                HttpServletResponse response) throws ServletException,
IOException {
        // TODO Auto-generated method stub
        String course_name =
request.getParameter("course_name").toUpperCase();
        String start_date = request.getParameter("start_date");
        String end_date = request.getParameter("end_date");
        int capacity = 0;
        String msg = "";
        Course course = new Course();
        course.setCapacity(capacity);
        course.setCourse_name(course_name);
        course.setStart_date(start_date);
        course.setEnd_date(end_date);
        try {
            capacity = Integer.parseInt(request.getParameter("capacity"));
        } catch (NumberFormatException e) {
            msg = msg + "Enter integral value for capacity. ";
        }
        boolean isDuplicate = new
CourseUtilsImpl().checkDuplicate(course_name);
        if (isDuplicate) {
            request.setAttribute("message", "The course is already
registered");
            request.setAttribute("courseobj", course);
            RequestDispatcher dispatcher = request
                    .getRequestDispatcher("NewCourseForm.jsp");
            dispatcher.forward(request, response);

        } else {
            if (msg.equals(""))
                if (capacity <= 0)
                    msg = msg + "Capacity should be positive. ";
            if (course_name.equals(""))
                msg = msg + "Necessary fields have been left empty. ";
            String message = new DateUtils().checkDates(start_date,
end_date);
            if (message != null) {
                msg = msg + message;
            }
            if (!msg.equals("")) {
                request.setAttribute("courseobj", course);
                request.setAttribute("message", msg);
                RequestDispatcher dispatcher = request
                        .getRequestDispatcher("NewCourseForm.jsp");
                dispatcher.forward(request, response);
            } else {
                course.setCapacity(capacity);
                boolean added = new CourseUtilsImpl().addCourse(course);
                if (added) {
                    Course course2 = new CourseUtilsImpl()
                            .searchCourse(course_name);
                    request.setAttribute("name",
                            "Course " +
course2.getCourse_name());
                    request.setAttribute("id",
course2.getCourse_id());
                    request.setAttribute("profile",
"RegistrarProfile.jsp");
                    RequestDispatcher dispatcher = request

    .getRequestDispatcher("ConfirmationMessage.jsp");
                    dispatcher.forward(request, response);
                }
            }
        }

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws ServletException,
IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

7. Class name: UpdateCourseservlet.java

```
package com.project.servlets;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

import com.project.beans.Course;
import com.project.utils.CourseUtilsImpl;
import com.project.utils.StudentUtilsImpl;

/**
 * Servlet implementation class UpdateCourseServlet
 */
@WebServlet("/updatecourse")
public class UpdateCourseServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public UpdateCourseServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws ServletException,
IOException {
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws ServletException,
IOException {
        // TODO Auto-generated method stub
        int new_capacity = 0;
        String message = null;
        String course_name = request.getParameter("course");
        try {
            new_capacity = Integer.parseInt(request
                    .getParameter("new_capacity"));
            if (new_capacity < new StudentUtilsImpl()
                    .getStudentsInACourseCount(course_name))
                message = "Number of students already registered for the
course is larger than capacity. ";
        } catch (NumberFormatException e) {
            message = "Enter positive integral value for capacity";
        }

        if (message != null) {
            Course courseObj = new
CourseUtilsImpl().searchCourse(course_name);
            request.setAttribute("message", message);
            request.setAttribute("courseobj", courseObj);
            RequestDispatcher dispatcher = request
                    .getRequestDispatcher("SearchCourse.jsp");
            dispatcher.forward(request, response);
        } else {
            boolean updated = new
CourseUtilsImpl().updateCourse(course_name,
                    new_capacity);
            if (updated) {
                Course course = new
CourseUtilsImpl().searchCourse(course_name);
                request.setAttribute("courseobj", course);
                RequestDispatcher dispatcher = request

    .getRequestDispatcher("UpdateConfirmationMessage.jsp");
                dispatcher.forward(request, response);
            }
        }
    }

}
```

8. Class name: CourseSelectionServlet.java

```
package com.project.servlets;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

import com.project.beans.Course;
import com.project.utils.*;

/**
 * Servlet implementation class SelectCourseServlet
 */
@WebServlet("/selectcourse")
public class CourseSelectionServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public CourseSelectionServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             String course = request.getParameter("courses");
             CourseUtils util = new CourseUtilsImpl();
             Course courseObj = util.searchCourse(course);
             request.setAttribute("courseobj", courseObj);
             RequestDispatcher dispacher =
request.getRequestDispatcher("SelectCourse.jsp");
             dispacher.forward(request, response);
      }

      /**
       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             doGet(request,response);
      }

}
```

9. Class name: CourseDetailsservlet.java

```java
package com.project.servlets;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.project.beans.Course;
import com.project.utils.CourseUtilsImpl;

/**
 * Servlet implementation class CourseUpdationServlet
 */
@WebServlet("/coursedetails")
public class CourseDetailsServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public CourseDetailsServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             String course_name = request.getParameter("courses");
             Course courseObj = new CourseUtilsImpl().searchCourse(course_name);
```

```
		request.setAttribute("courseobj",courseObj);
		RequestDispatcher dispatcher = 
request.getRequestDispatcher("SearchCourse.jsp");
		dispatcher.forward(request, response);
		
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse 
response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse 
response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		doGet(request, response);
		
	}

}
```

Utility interfaces and classes:

Interface name: <u>CourseUtils.java</u>

```
package com.project.utils;

import java.util.List;

import com.project.beans.Course;

public interface CourseUtils {

	public boolean addCourse(Course course);
	public Course searchCourse(String course_name);

	public boolean updateCourse(String course_name,int new_capacity);

	public List<String> getNameOfCourses();

	public List<Course> getAllCourses();

	public boolean checkDuplicate(String course_name);

}
```

Implementing class name: CourseUtilImpl.java

```
package com.project.utils;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

import com.project.db.DBUtils;
import com.project.beans.Course;

public class CourseUtilsImpl implements CourseUtils {

	Connection conn = null;
	PreparedStatement pst = null;
	ResultSet rs = null;
	Statement st = null;

	@Override
	public boolean addCourse(Course course) {
		// TODO Auto-generated method stub
		boolean isCourseAdded = false;
		conn = DBUtils.connectToDb();
		String query = "insert into course
(course_name,start_date,end_date,capacity) values (?,?,?,?);";
		pst = DBUtils.getPreparedStatement(conn, query);
		int rows = 0;
```

```
        try {
            pst.setString(1, course.getCourse_name());
            pst.setString(2, course.getStart_date());
            pst.setString(3, course.getEnd_date());
            pst.setInt(4, course.getCapacity());

            rows = pst.executeUpdate();
            DBUtils.closeConnections();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        if (rows > 0)
            isCourseAdded = true;

        return isCourseAdded;
    }

    @Override
    public Course searchCourse(String course_name) {
        // TODO Auto-generated method stub
        conn = DBUtils.connectToDb();
        String query = "select * from course where course_name = ?;";
        pst = DBUtils.getPreparedStatement(conn, query);
        Course course = null;
        try {
            pst.setString(1, course_name);
            rs = pst.executeQuery();
            if (rs != null)
                if (rs.next()) {
                    course = new Course();
                    course.setCourse_id(rs.getInt(1));
                    course.setCourse_name(rs.getString(2));
                    course.setStart_date(rs.getString(3));
                    course.setEnd_date(rs.getString(4));
                    course.setCapacity(rs.getInt(5));
                }
            DBUtils.closeConnections();
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return course;
    }

    @Override
    public boolean updateCourse(String course_name, int new_capacity) {
        // TODO Auto-generated method stub
        boolean isCourseUpdated = false;

        conn = DBUtils.connectToDb();
        String query = "update course set capacity = ? where course_name =
?";
        pst = DBUtils.getPreparedStatement(conn, query);
        int rows = 0;
        try {
            pst.setInt(1, new_capacity);
            pst.setString(2, course_name);
            rows = pst.executeUpdate();
            DBUtils.closeConnections();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        if (rows > 0)
            isCourseUpdated = true;

        return isCourseUpdated;

    }

    @Override
    public List<String> getNameOfCourses() {
        // TODO Auto-generated method stub
        conn = DBUtils.connectToDb();
        String query = "select course_name from course;";
        st = DBUtils.getStatement(conn);
        List<String> list = null;
        try {
            rs = st.executeQuery(query);
            if (rs != null) {
                list = new ArrayList<String>();
                while (rs.next())
                    list.add(rs.getString(1));
            }
            DBUtils.closeConnections();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return list;
    }

    @Override
    public boolean checkDuplicate(String course_name) {
        // TODO Auto-generated method stub
        boolean isDuplicate = false;
        conn = DBUtils.connectToDb();
        String query = "select course_name from course where course_name=
?;";
        pst = DBUtils.getPreparedStatement(conn, query);
        try {
            pst.setString(1, course_name);
            rs = pst.executeQuery();
            if (rs.next())
                isDuplicate = true;
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        DBUtils.closeConnections();
        return isDuplicate;
    }

    @Override
    public List<Course> getAllCourses() {
        // TODO Auto-generated method stub
        List<Course> list = null;
        conn = DBUtils.connectToDb();
        Course course = null;
        String query = "select * from course;";
        try {
            st = conn.createStatement();
            rs = st.executeQuery(query);
            if(rs!=null)
            {
                list = new ArrayList<Course>();
                while(rs.next())
                {
                    course = new Course();
                    
    course.setCourse_id(Integer.parseInt(rs.getString(1)));
                    course.setCourse_name(rs.getString(2));
                    course.setStart_date(rs.getString(3));
                    course.setEnd_date(rs.getString(4));
                    
    course.setCapacity(Integer.parseInt(rs.getString(5)));
                    list.add(course);
                }
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        
        return list;
    }
}
```

Class name: DateUtils.java

```
package com.project.utils;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
```

```java
import java.util.Date;

public class DateUtils {

    public String checkDates(String start_date,String end_date)
    {
        String msg = null;
        SimpleDateFormat ft = new SimpleDateFormat("yyyy-MM-dd");
        ft.setLenient(false);
        try {
            Date dt1 = ft.parse(start_date);
            Date dt2 = ft.parse(end_date);

            Calendar sDate = Calendar.getInstance();
            Calendar eDate = Calendar.getInstance();
            Calendar currentDate = Calendar.getInstance();

            sDate.setTime(dt1);//here inputDate is date given by the user.
            eDate.setTime(dt2);//here inputDate is date given by the user.

            if(sDate.before(currentDate)) {
                msg = "Start date is before current date(today)";
            }else if(eDate.before(sDate)) {
                msg = "End date is before start date.";
            }
        } catch (ParseException e) {
            // TODO Auto-generated catch block
            msg = "Stick to specified date format.";
        }
        return msg;
    }
}
```

Interface name: MemberUtils.java

```java
package com.project.utils;

public interface MemberUtils {
    public boolean authenticate(String email, String password, String
designation);

    public boolean changePassword(String email, String password, String
newPassword);

}
```

Implementing class name: MemberUtilsImpl.java

```
package com.project.utils;

import java.sql.*;

import com.project.db.DBUtils;

public class MemberUtilsImpl implements MemberUtils {

    Connection conn = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    @Override
    public boolean authenticate(String email, String password,
            String designation) {
        // TODO Auto-generated method stub
        boolean authenticationStatus = false;

        conn = DBUtils.connectToDb();
        String query = "select * from member where emailid = ? and password =
? and designation = ?;";
        pst = DBUtils.getPreparedStatement(conn, query);
        try {
            pst.setString(1, email);
            pst.setString(2, password);
            pst.setString(3, designation);
            rs = pst.executeQuery();
```

```
                  if (rs.next()) {
                        authenticationStatus = true;
                  }
            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }
            DBUtils.closeConnections();
            return authenticationStatus;
      }

      @Override
      public boolean changePassword(String email, String password,
                  String newPassword) {
            // TODO Auto-generated method stub

            boolean isPasswordChanged = false;
            conn = DBUtils.connectToDb();
            String query = "update member set password = ? where emailid = ? and
password = ?;";
            pst = DBUtils.getPreparedStatement(conn, query);
            int val = 0;
            try {

                  pst.setString(1, newPassword);
                  pst.setString(2, email);
                  pst.setString(3, password);
                  val = pst.executeUpdate();

            } catch (SQLException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }
            if (val > 0) {
                  isPasswordChanged = true;
            }

            DBUtils.closeConnections();
            return isPasswordChanged;
```

```
        }

}
```

Interface name: StudentUtils.java

```
package com.project.utils;

import com.project.beans.Student;

public interface StudentUtils {

        public boolean addStudent(Student student);

        public int getStudentsInACourseCount(String course);

        public Student searchStudent(String email);

        public boolean isStudentDuplicate(Student student);

}
```

Implementing class name: StudentUtilsImpl.java

```
package com.project.utils;

import java.sql.*;

import com.project.db.DBUtils;
import com.project.beans.Student;

public class StudentUtilsImpl implements StudentUtils {

        Connection conn = null;
        PreparedStatement pst = null;
        ResultSet rs = null;
```

```
    @Override
    public boolean addStudent(Student student) {
        // TODO Auto-generated method stub
        boolean isStudentAdded = false;
        conn = DBUtils.connectToDb();
        String query = "insert into student 
(student_first_name,student_middle_name,student_last_name,student_emailid,student_
course) values (?,?,?,?,?);";
        pst = DBUtils.getPreparedStatement(conn, query);
        int rows = 0;
        try {
            pst.setString(1, 
student.getStudent_first_name().toUpperCase());
            pst.setString(2, 
student.getStudent_middle_name().toUpperCase());
            pst.setString(3, 
student.getStudent_last_name().toUpperCase());
            pst.setString(4, student.getStudent_emailid());
            pst.setString(5, student.getStudent_course());

            rows = pst.executeUpdate();
            DBUtils.closeConnections();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        if (rows > 0)
            isStudentAdded = true;

        return isStudentAdded;
    }

    @Override
    public int getStudentsInACourseCount(String course) {
        // TODO Auto-generated method stub
        conn = DBUtils.connectToDb();
        String query = "select count(*) from student where student_course = 
?;";
        pst = DBUtils.getPreparedStatement(conn, query);
        int rows = 0;
        try {
            pst.setString(1, course);
            rs = pst.executeQuery();
            if (rs.next())
                rows = rs.getInt(1);
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        DBUtils.closeConnections();
        return rows;
    }

    @Override
    public Student searchStudent(String email) {
        // TODO Auto-generated method stub
        Student student = null;
        conn = DBUtils.connectToDb();
        String query = "select student_id, student_first_name from student
where student_emailid = ?;";
        pst = DBUtils.getPreparedStatement(conn, query);
        try {

            pst.setString(1, email);
            rs = pst.executeQuery();
            if (rs != null) {
                if (rs.next()) {
                    student = new Student();
                    student.setStudent_id(rs.getInt(1));
                    student.setStudent_first_name(rs.getString(2));
                }
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        DBUtils.closeConnections();
        return student;
    }
```

```
	@Override
	public boolean isStudentDuplicate(Student student) {
		// TODO Auto-generated method stub
		boolean isStudentDuplicate = false;
		conn = DBUtils.connectToDb();
		String query = "select * from student where student_emailid=?;";
		pst = DBUtils.getPreparedStatement(conn, query);
		try {
			pst.setString(1, student.getStudent_emailid());
			rs = pst.executeQuery();
			if (rs != null)
				if (rs.next())
					isStudentDuplicate = true;
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		return isStudentDuplicate;
	}
}
```

JSPs

Login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Login Page</title>

</head>

<body bgcolor='silver'>
<b><font face="calibri">
<center>
<input type="button" value="Login Page" bgcolor="lightblue"
style="width: 1500px; height: 72px; font-size: 36px; color: WhiteSmoke; font-
style: calibri; font-weight: bolder; font-family: Calibri; background-color: Teal"
disabled="disabled">
</center> <br> <br> <br>
<%
      Boolean loginStatus = true;
      Object status = request.getAttribute("loginStatus");
      if (status != null)
            loginStatus = (Boolean) status;
%>
<center>
<form name="loginform" action="login" method="post">
      <table border=2 align="center" style="width: 638px; height: 235px"
            bgcolor='white'>
      <tr align="center">
            <td><b><font color='white'>...</font><font color='teal'><big>Email
                  Id</big></font><font color='white'>...</font></b>
            <td><big><input type="text" name="email" size="20"
                  style="width: 153px; font-weight: normal;"></big>
      <tr align="center">
            <td><font color='teal'><big><b>Password</font> </big>
            <td><big><input type="password" name="pswd" size="20"
                  style="width: 153px; font-weight: normal;"></big>
      <tr align="center">
            <td style="width: 295px;"><font color='teal'><big><b>LogIn
                  As</b> </big></font>
            <td><select name="designation"style="width: 153px; height: 26px;
            font-weight: normal;">
                  <option value="Counsellor"><big>Counsellor</big></option>
                  <option value="Registrar"><big>Registrar</big></option>
                </select><big> </big>
      </table>
<%
      String msg = "";
      if (!loginStatus)
            msg = "Invalid username, password or designation";
%>

            <font size='4' color='red'><%=msg%></font> <br>
            <br> <input type="submit" name="submit" value="Login"style="width:
      121px;      color: AntiqueWhite; font-weight: bold; height: 44px; font-
      size: 14px; background-color: Teal">
      </form>
</center>
</font> </b>
</body>
</html>
```

ChangePassword.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE    html    PUBLIC    "-//W3C//DTD    HTML    4.01    Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Change Password</title>
</head>
<body bgcolor='silver'>
<center>
<b><font face="calibri" color = 'teal'> <input type="button"
value="Change Password" bgcolor="lightblue"
```

```
style="width: 1500px; height: 72px; font-size: 36px; color: WhiteSmoke; font-style:
calibri; font-weight: bolder; font-family: Calibri; background-color: Teal"
disabled="disabled">
<h2>
<b> </b>
</h2> <br> <br> <br>
<form name="changepasswordform" action="changepassword"
method="post">
<table border=2 align="center" style="width: 638px; height: 235px"bgcolor='white'>
<tr align="center">
<td><b><font color='teal'><big>Email Id</big></font></b>
<td><big><input type="text" name="email" size="20"style="width: 153px; font-weight:
normal;"></big>
<tr align="center">
<td><font color='teal'><big><b>Password</font> </big>
<td><big><input type="password" name="pswd" size="20" style="width: 153px; font-
weight: normal;"></big>
<tr align="center">
<td><font color='teal'><big><b>New Password</font>
</big>
<td><big><input type="password" name="nwpswd"
size="20" style="width: 153px; font-weight: normal;"
placeholder="6-10 characters"></big><big> </big>
<tr align="center">
<td><font color='teal'><big><b>Confirm
Password</b></font> </big>
<td><big><input type="password" name="cnfrmnwpswd"
size="20" style="width: 153px; font-weight: normal;"
placeholder="6-10 characters"></big><big> </big>
</table>
<big> </big>
<%
       String message = (String) request.getAttribute("message");
       if (message != null) {
%>
<!-- <font size='7' color='red'>!</font> -->
<font size='3' color='red'><%=message%></font>
<%
       } else
```

```
%>
<br> <br> <input type="submit" name="submit"
value="Submit"
style="width: 121px; color: AntiqueWhite; font-weight: bold; height: 44px; font-
size: 14px; background-color: Teal">
</form>

</font> </b>
</center>
</body>
</html>
```

ConfirmationMessage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE    html    PUBLIC    "-//W3C//DTD    HTML    4.01    Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Confirmation Page</title>
</head>
<body bgcolor = 'silver'>
<center>
<img src ="confirmationFace.jpg"><br>
<h2><font size = 5 color = 'teal'>Registration Successful!!</font></h2>
<br>
<br><b><%=request.getAttribute("name")%></b> has been successfully registered.<br>
Id No.: <b><%=request.getAttribute("id")%></b>.
<%
      String profile = (String)request.getAttribute("profile");
      if(profile==null)
      {
%>
      <br><font   color   =   'teal'>To   get   back   to   profile</font>   <a
href="CounsellorProfile.jsp"><i>Click
here</i></a>
<%
```

```
       }
       else
       {
%>
<br><font  color  =  'teal'>To  get  back  to  profile</font>  <a  href="<%=profile
%>"><i>Click here</i></a>
<%
       }
%>
</center>
</body>
</html>
```

SearchCourse.jsp

```
<%@page import="com.project.utils.CourseUtilsImpl"%>
<%@page import="java.util.List"%>
<%@page import="com.project.utils.StudentUtilsImpl"%>
<%@page import="com.project.beans.Course"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Search Course</title>
</head>
<body bgcolor='silver'>
<font face = "calibri">
<center>
<input type="button" value="Search Course" bgcolor="lightblue" style="width:
1500px; height: 72px; font-size: 36px; color: WhiteSmoke; font-style: calibri;
font-weight: bolder; font-family: Calibri; background-color: Teal"
disabled="disabled"><h2><b></b></h2>
<br>
<br>
<%
        !List<String> list = null;
```

```
%>
<form name="selectcourse" action="coursedetails" method="post">
<select name="courses" style="width: 153px; height: 26px; text-align: center">
<%
        list = new CourseUtilsImpl().getNameOfCourses();
        for (String course : list) {
%>
<option value="<%=course%>"><%=course%></option>
<%
        }
%>
</select> <input type="submit" name="search" value="SEARCH" style="width: 115px;
color: AntiqueWhite; font-weight: bold; height: 32px; font-size: 14px; background-
color: Teal">
</form>
</center>
<br>
<%
        int capacity = 0;
        String course = "";
        String start_date = "";
        String end_date = "";
        int course_id = 0;
        Course obj = (Course) request.getAttribute("courseobj");
        if (obj != null) {
        course = obj.getCourse_name();
        capacity = obj.getCapacity();
        start_date = obj.getStart_date();
        end_date = obj.getEnd_date();
        course_id = obj.getCourse_id();
%>
<br>
<center>
<form name="updatecourse" action="updatecourse" method="post">
        <table border=2 align="center" style="width: 638px; height: 235px;
        background-color: White">
                <tr align="center">
                        <td style="width: 298px; "><font color='teal'><b>Course
                Name</b></font>
```

```
                        <td style="width: 278px; "><input type="text"
                name="course" size="20" style="width: 153px; text-align: center"
                value="<%=course%>" readonly="readonly">
                <tr align="center">
                        <td ><font color='teal'><b>Course Id</b></font>
                        <td><%=course_id%>
                <tr align="center">
                        <td ><font color='teal'><b>Start Date</b></font>
                        <td><%=start_date%>
                <tr align="center">
                        <td><font color='teal'><b>End Date</b></font>
                        <td><%=end_date%>
                <tr align="center">
                        <td><font color='teal'><b>Capacity</b></font>
                        <td><%=capacity%>
        </table>
<br><br>
        <font size = 4 color='teal'>Enter new capacity<font color =
      'silver'>........</font></font> <input type="text" name="new_capacity"
        size="20" style="width: 153px;;"><br><br>
<%
        Object obj2 = request.getAttribute("message");
        if (obj2!=null) {
        String msg = (String) obj2;
%>
        <font size='3' color='red'><%=msg%></font><br>
<%
        } else {
%>
        <br>
<%
        }
%>
        <br/>
        <input type="submit" name="update" value="Update" style="width: 115px;
      color: AntiqueWhite; font-weight: bold; height: 32px; font-size: 14px;
      background-color: Teal">
</form>
</center>
```

```
<%
        }
%>
</font>
</body>
</html>
```

SelectCourse.jsp

```
<%@page import="com.project.utils.CourseUtilsImpl"%>
<%@page import="java.util.List"%>
<%@page import="com.project.utils.StudentUtilsImpl"%>
<%@page import="com.project.utils.StudentUtils"%>
<%@page import="com.project.beans.Course"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE    html    PUBLIC    "-//W3C//DTD    HTML    4.01    Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body bgcolor='silver'>

<font face = "calibri">
<center>
<input  type="button"  value="Select  a  Course"  bgcolor="lightblue"  style="width:
1500px; height: 72px; font-size: 36px; color: WhiteSmoke; font-style: calibri; font-
weight:     bolder;     font-family:     Calibri;     background-color:     Teal"
disabled="disabled"><h2><b></b></h2>
<br>
<br>

<%
        !List<String> list = null;
%>
<form name="selectcourse" action="selectcourse" method="post">
```

```
<select name="courses" style="width: 153px; height: 26px">
<%
        list = new CourseUtilsImpl().getNameOfCourses();
        for (String course : list) {
%>
<option value="<%=course%>"><%=course%></option>
<%
        }
%>
</select>

<font  color  =  'silver'>..........</font>  <input  type="submit"  name="submit"
value="SUBMIT" style="width: 115px; color: AntiqueWhite; font-weight: bold; height:
32px; font-size: 14px; background-color: Teal">
</form>
</center>
<br>
<br>
<center>
<%
        int capacity = 0;
        String course = "";
        int studentCount = 0;
        Course obj = (Course) request.getAttribute("courseobj");
        if (obj != null) {
                course = obj.getCourse_name();
                capacity = obj.getCapacity();
        studentCount = new StudentUtilsImpl()
                        .getStudentsInACourseCount(course);
        Boolean isApplicable = false;
        if (studentCount < capacity)
                isApplicable = true;
%>
<form name="coursedetails" action="registrationpage" method="post">
        <table  border=2  align="center"  style="width:  638px;  height:  235px;
      background-color: White">
        <tr align="center">
                <td  style="width:  289px;  "><font  color  =  "teal"><b>Selected
        Course</b></font>
```

```
		<td  style="width:  284px;  "><input  type="text"  name="course"
	size="20"style="width:   183px;   color:   Black;   text-align:   center"
	readonly="readonly" value="<%=course%>">
	<tr align="center">
		<td ><font color = "teal"><b>Maximum Capacity</b></font>
		<td><%=capacity%>
	<tr align="center">
		<td ><font color = "teal"><b>Registered Student</b></font>
		<td><%=studentCount%>
	</table><br>
<%
	if (isApplicable == false) {
%>
<font size="3" color="red">Seats full</font><br><br>
<input type="submit" name="continue" value="Continue" style="width: 115px; color:
AntiqueWhite; font-weight: bold; height: 32px; font-size: 14px; background-color:
Teal" disabled="disabled">
<br /><font color = "red"> Select other course to check
availability of seats</font><br><br /> <a href="CounsellorProfile.jsp"><i>Go
To Previous Page</i></a>
<%
	} else {
%>
<input type="submit" name="continue" value="Continue" style="width: 115px; color:
AntiqueWhite; font-weight: bold; height: 32px; font-size: 14px; background-color:
Teal">

<%
	}
%>
</form>
<%
	}
%>
</center>

</font>
</body>
</html>
```

StudentRegistrationForm.jsp

```
<%@page import="com.project.beans.Student"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Registration Form</title>
</head>
<body bgcolor="silver">
<font face="calibri">
<center>
<input type="button" value="Registration Form" bgcolor="lightblue"
style="width: 1500px; height: 72px; font-size: 36px; color: WhiteSmoke; font-style:
calibri; font-weight: bolder; font-family: Calibri; background-color:
Teal"disabled="disabled">
<h2>
<b></b>
</h2>
<br> <br> <br> <font size='3' color='blue'><b>NOTE:
</b><i><font size='3' color='red'>* marked fields are compulsory
fields</font></i></font><br> <br>
<form name="studentform" action="register" method="post">
<%
        Student student = (Student) request.getAttribute("studentobj");
        if (student == null) {
%>
<table border=2 align="center" style="width: 612px; height: 377px"
height="360" bgcolor='white'>
        <tr align="center">
                <td style="width: 270px;"><font color="teal"><b>First
                Name</b></font> <font size='3' color='red'>*</font>
                <td><input type="text" name="first_name" size="20"
                style="width: 171px;">
        <tr align="center">
                <td><font color="teal"><b>Middle Name</b></font>
                <td><input type="text" name="middle_name" size="20"
                style="width: 171px;">
        <tr align="center">
                <td><font color="teal"><b>Last Name</b> </font><font size='3'
                color='red'>*</font>
                <td><input  type="text"  name="last_name"  size="20"style="width:
                171px;">
        <tr align="center">
                <td><font color="teal"><b>Email Id</b> </font><font size='3'
                color='red'>*</font>
                <td><input   type="text"   name="emailid"   size="20"style="width:
                171px;">
        <tr align="center">
                <td><font color="teal"><b>Course</b></font>
                <td><input   type="text"   name="course"   size="20"style="width:
                171px;"readonly="readonly"
                value="<%=request.getAttribute("course")%>">
</table>
<br> <br>
<%
        } else {
%>
<table border=2 align="center" style="width: 612px; height: 377px"
height="360" bgcolor='white'>
        <tr align="center">
                <td style="width: 270px;"><font color="teal"><b>First
                Name</b></font> <font size='3' color='red'>*</font>
                <td><input type="text" name="first_name" size="20"
                value="<%=student.getStudent_first_name()%> "
                style="width: 171px;">
        <tr align="center">
                <td><font color="teal"><b>Middle Name</b></font>
                <td><input type="text" name="middle_name" size="20"
                style="width: 171px;"
                value="<%=student.getStudent_middle_name()%>">
        <tr align="center">
                <td><font color="teal"><b>Last Name</b> </font><font size='3'
                color='red'>*</font>
                <td><input type="text" name="last_name" size="20"
                style="width:171px;" value="<%=student.getStudent_last_name()%>">
        <tr align="center">
                <td><font color="teal"><b>Email Id</b> </font><font size='3'
                color='red'>*</font>
                <td><input type="text" name="emailid" size="20"
                style="width:171px;" value="<%=student.getStudent_emailid()%>">
        <tr align="center">
                <td><font color="teal"><b>Course</b></font>
                <td><input type="text" name="course" size="20"
                style="width:171px;" value="<%=request.getAttribute("course")%>"
                readonly="readonly">
</table>

<br>
<%
        }
        Boolean null_flag = false;
        Object obj = request.getAttribute("message");
        if (obj != null)
        null_flag = (Boolean) obj;
        if (null_flag == true) {
%>
<font size='3' color='red'>Necessary fields have been left
empty</font><br> <br>
<%
        } else {
        Boolean duplicate_flag = false;
        Object flagobj = request.getAttribute("message2");
        if (flagobj != null)
                duplicate_flag = (Boolean) flagobj;
        if (duplicate_flag == true) {
%>
<font size='3' color='red'>Student is already registered for
some      course.</font><br>To      go      back      to      profile<i>      <a
href="CounsellorProfile.jsp">Click
here</a></i><br>
<%
```

```
        } else {
%>
<br>
<%
        }
        }
%>
<input type="submit" name="submit" value="SUBMIT"
style="width: 115px; color: AntiqueWhite; font-weight: bold; height: 32px; font-size: 14px; background-color: Teal">

</form>
</center>
</font>
</body>
</html>
```

UPDATE CONFIRMATION MESSAGE

```
<%@page import="com.project.beans.Course"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body bgcolor ='silver'>
<font color ='teal'>

<center>
<img src ="confirmationFace.jpg"><br>
<h2>Successfully Updated!!</h2>
<br>
<br>
<%
        Course course = (Course) request.getAttribute("courseobj");
```

```
%>
Course <b><%=course.getCourse_name()%></b> has been successfully
updated.<br>
<table>
        <tr>
                <td>Id No.:
                <td><b><%=course.getCourse_id()%></b>
        <tr>
                <td>Updated Capacity :
                <td><b><%=course.getCapacity()%></b>
</table>
<br>To get back to profile
<a href = "RegistrarProfile.jsp"><i>Click here</i></a>
</center>
</font>
</body>
</html>
```

Welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE    html    PUBLIC    "-//W3C//DTD    HTML    4.01    Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Home Page</title>
</head>
<body bgcolor = 'teal'>
<center>
<br /> <br />
<font color = 'white' face ='Calibri' size =7 color = 'silver'>Welcome to Photon
Finishing School
</font> <br /> <br /> <a href = "LogIn.jsp"><img src='8.jpg' style="width: 810px;
height: 533px"></a>
<br /> <br /><font color = 'white' face ='Calibri' size =4>Click on image to login
</font>
```

```
</center>

</body>
</html>
```

CounsellorProfile.jsp

```
<%@page import="com.project.beans.Course"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE    html    PUBLIC    "-//W3C//DTD    HTML    4.01    Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>New Course Form</title>
</head>
<body bgcolor="silver">
<font face="calibri">
<center>
<input type="button" value="New Course Details" bgcolor="lightblue"
style="width: 1500px; height: 72px; font-size: 36px; color: WhiteSmoke; font-style:
calibri; font-weight: bolder; font-family: Calibri; background-color: Teal"
disabled="disabled">
<h2>
<b></b>
</h2>
<br> <br> <br> <font size='3' color='blue'>NOTE: </font><font size='3' color='red'>*
marked  fields  are  compulsory  fields.  <br><font  size='3'  color='blue'>Date
format:</font> YYYY-MM-DD<br>

</font><br>
<form name="courseform" action="newcourse" method="post">
<%
Course course = (Course) request.getAttribute("courseobj");
if (course == null) {
%>
<table border=2 align="center" style="width: 602px; height: 260px; background-color:
White">
```

```
<tr align="center">
<td  style="width:  298px;  "><font  color  =  "teal"><b>Course  Name</b></font><font
size='3' color='red'>*</font>
<td style="width: 244px; "><input type="text" name="course_name" size="20"
style="width: 171px;">
<tr align="center">
<td><font     color     =     "teal"><b>Start     Date</b></font><font     size='3'
color='red'>*</font>
<td><input type="text" name="start_date" size="20"
style="width: 171px;" placeholder = "YYYY-MM-DD">
<tr align="center">
<td><font color = "teal"><b>End Date</b></font><font size='3' color='red'>*</font>
<td><input type="text" name="end_date" size="20"
style="width: 171px;" placeholder = "YYYY-MM-DD">
<tr align="center">
<td><font color = "teal"><b>Capacity</b></font><font size='3' color='red'>*</font>
<td><input type="text" name="capacity" size="20"
style="width: 171px;" placeholder = "0">
</table>
<%
} else {
%>
<table border=2 align="center" style="width: 602px; height: 260px; background-color:
White">
<tr align="center">
<td  style="width:  298px;  "><font  color  =  "teal"><b>Course  Name</b></font><font
size='3' color='red'>*</font>
<td style="width: 244px; "><input type="text" name="course_name" size="20"
style="width: 171px;" value="<%=course.getCourse_name()%>">
<tr align="center">
<td><font     color     =     "teal"><b>Start     Date</b></font><font     size='3'
color='red'>*</font>
<td><input type="text" name="start_date" size="20"
style="width: 171px;" value="<%=course.getStart_date()%>">
<tr align="center">
<td><font color = "teal"><b>End Date</b></font><font size='3' color='red'>*</font>
<td><input type="text" name="end_date" size="20"
style="width: 171px;" value="<%=course.getEnd_date()%>">
<tr align="center">
```

```
<td><font color = "teal"><b>Capacity</b></font><font size='3' color='red'>*</font>
<td><input type="text" name="capacity" size="20"
style="width: 171px;" value="<%=course.getCapacity()%>">
</table>

<br><%
}
String msg = "";
Object obj = request.getAttribute("message");
if (obj != null)
msg = (String) obj;
if (!msg.equals("")) {
%>
<font size='3' color='red'><%=msg%></font><br>
<%
} else {
%>
<br><br>
<%
}
%>
<input type="submit" name="submit" value="SUBMIT"
style="width: 115px; color: AntiqueWhite; font-weight: bold; height: 32px; font-
size: 14px; background-color: Teal">
</form>
</center>
</font>
</body>
</html>
```

NewCourseForm.jsp

```
<%@page import="com.project.utils.CourseUtilsImpl"%>
<%@page import="com.project.utils.CourseUtils"%>
<%@page import="com.project.utils.StudentUtilsImpl"%>
<%@page import="com.sun.jndi.url.corbaname.corbanameURLContextFactory"%>
<%@page import="com.project.beans.Course"%>
<%@page import="java.util.List"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>List of courses</title>
</head>
<body bgcolor='silver'>
<b><font face="calibri">
<center>
<input type="button" value="List of courses" bgcolor="lightblue" style="width:
1500px; height: 72px; font-size: 36px; color: WhiteSmoke; font-style: calibri; font-
weight: bolder; font-family: Calibri; background-color: Teal"
disabled="disabled"><h2><b>
</b></h2>
<br>
<br>
<br>
<%
	CourseUtils util = new CourseUtilsImpl();
	List<Course> list = util.getAllCourses();
	if(list!=null){
%>

<table border=2 align="center" style="width: 638px; height: 235px" bgcolor =
'white'>
<tr align="center">
	<td><b><font color ='teal'><big>Course Id</big></font></b>
	<td><b><font color ='teal'><big>Course Name</big></font></b>
	<td><b><font color ='teal'><big>Course Start Date</big></font></b>
	<td><b><font color ='teal'><big>Course End Date</big></font></b>
	<td><b><font color ='teal'><big>Course Capacity</big></font></b>
	<td><b><font color ='teal'><big>Seats filled</big></font></b>
<%
	for(Course course : list)
	{
%>
<tr align="center">
	<td><b><big><%=course.getCourse_name() %></big></b>
```

```
       <td><b><big><%=course.getCourse_name() %></big></b>
       <td><b><big><%= course.getStart_date() %></big></b>
       <td><b><big><%=course.getEnd_date() %></big></b>
       <td><b><big><%=course.getCapacity() %></big></b>
       <td><b><big><%=new
StudentUtilsImpl().getStudentsInACourseCount(course.getCourse_name()) %></big></b>
<%
       }
%>
</table>
<%
       }else
       {
%>
no courses to display...
<%
       }
%>
<br/><br/><br/><br/>To go back to profile <a href = "Registrar's Profile"><i>Click
here</i></a>
</center>
</font>
</b></body>
</html>
```

PasswordChangeConfirmation.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Confirmation Page</title>
</head>
<body bgcolor='silver'>
```

```
<center>
<font color = 'teal'>

<img src ="confirmationFace.jpg"><br>
<h2>
<font size=5 color='teal'>Operation Successful!!</font>
</h2>
<br> <br> <br> <font size=4 color ='black'>
Password Successfully Updated!!</font> <br> <br> <a
href="LogIn.jsp">Click To Login</a>
</font>
</center>

</body>
</html>
```

RegistrarProfile.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Registrar's Profile</title>
</head>
<body bgcolor="silver">
<font face="calibri">
<center>
<input type="button" value="Registrar's Profile" bgcolor="lightblue"
style="width: 1500px; height: 72px; font-size: 36px; color: WhiteSmoke; font-style: calibri; font-weight: bolder; font-family: Calibri; background-color: Teal"
disabled="disabled">
<h2>
```

```
<b></b>
</h2>

<br /> <br /> <br /> <a href="NewCourseForm.jsp"><font size=5
color='teal'><i>Add New Course</i></font></a> <br />  <br />
<br /> <a href="SearchCourse.jsp"><font size=5 color='teal'><i>Update
Capacity</i></font></a>
<br /> <br /> <br /> <a href="ChangePassword.jsp"><font size=5 color='teal'><i>Change
Password</i></font></a>

<br /> <br />
<br/>

<form name="logout" action="logout" method="post">
<br> <input type="submit" name="LogOut" value="LogOut"
style="width: 121px; color: AntiqueWhite; font-weight: bold; height: 44px; font-size: 14px;
background-color: Teal">
</form>
</center>
</font>
</body>
</html>
```

Certificate

This is to certify that Mr. Deep Prakash Dey of Netaji Subhash Engineering College, registration number: 111090110037 has successfully completed a project on Automated Student Registration System using JEE under the guidance of Mr. Bhavesh Sangwan.

Name of faculty is missing... !!!!!!!

[*Name of your faculty*]

Globsyn Finishing School

(*a division of Globsyn Skills*)

Certificate

This is to certify that Mr. Somnath Ghose Roy of Netaji Subhash Engineering College, registration number: 111090110097 has successfully completed a project on Automated Student Registration System using JEE under the guidance of Mr. Bhavesh Sangwan.

[*Name of your faculty*]

Globsyn Finishing School

Certificate

This is to certify that Mr. Sabyasachi Pahari of Netaji Subhash Engineering College, registration number: 111090110077 has successfully completed a project on Automated Student Registration System using JEE under the guidance of Mr. Bhavesh Sangwan.

[*Name of your faculty*]

Globsyn Finishing School

Certificate

This is to certify that Ms. Moumita Paul of Netaji Subhash Engineering College, registration number: 121090120014 has successfully completed a project on Automated Student Registration System using JEE under the guidance of Mr. Bhavesh Sangwan.

[*Name of your faculty*]

Globsyn Finishing School

Certificate

This is to certify that Mr. Manish Nag of Netaji Subhash Engineering College, registration number: 121090120011 has successfully completed a project on Automated Student Registration System using JEE under the guidance of Mr. Bhavesh Sangwan.

[*Name of your faculty*]

Globsyn Finishing School