**ONLINE NATIONAL JOB SEEKING PORTAL**

**TANMAY VINOD KOHAD –PRN:200950181106**

**nAGIREDDY aKHIL REDDY -PRN:200950181007**

**SRI VIDYA BURRA –PRN:200950181026**

**1. INTRODUCTION**

In the National Career Service (NCS) portal a Government department is akin to an individual who is responsible for managing activities of a particular government organization on the portal. As a Government department, the NCS portal assists you to search for jobseekers who match the job requirements of your particular department and hire those jobseekers. To provide employment to jobseekers, you, as a Government department have to first get yourself registered on the portal. Step-by-step instructions to perform different tasks on the NCS portal along with the objective of those tasks is described in various sections of this document.

To access the NCS portal, type the following website address or Uniform Resource Locator (URL) in the address bar of your Internet browser: http://www.ncs.gov.in/. The Home page of the NCS portal displays.

1. Government Department Registration
2. View/Edit Government Department Profile
3. Post New Job
4. Search Candidate
5. Jobs and Responses
6. User Management
7. Feedback on Local Services
8. Events
9. Change Password
10. Reports
11. Documents.

**JOB PORTAL MANAGEMENT SYSTEM**

Technology is constantly changing. Society as we know it depends on this fact. That which we take for granted today would have been the stuff of science fiction as little as fifty years ago. In fifty years time, we will doubtless be excited, perturbed and baffled by yet more new developments. In the early years of the twenty first century, it is computers and the Internet that have captured the public imagination, and found their way into not just the working environments, but increasingly into the domestic spaces. In this modern society, if we are not capable to cope up with these changes than we are not going to stand or survive anywhere in this technical world. Today there is no place for errors, so as to make a system more effective and efficient we need such technology where error prone chances must be least. In the scenario of the assignment, we are required to develop a web-based application on Job Portal Management System. In this time of recession where everyone, is either experienced or fresher, is in search for a job. This job portal can prove to be very helpful since it allows users of different profile to upload their CVs, search job on the basis of their qualification. Every user can access through user id and apply for multiple jobs at a time. Currently, we are working on a manual system where data is stored in the form of registers. Viewing available jobs, or applying for the job at the agency can be done for which jobseekers has to go to the agency and check the available jobs at the agency. Job seekers check the list of jobs available and apply the job. Then the agency will show available jobs for the job seeker for his qualifications and then updates the jobs database. The developed Job Portal management system is web-based which Requires Employee Registration & Profiles, Job Search, Employer Registration & Profiles, and Subscription Option for both Employee and Employer etc. Employer can add Own Profile and post jobs and Job Seeker can Search Jobs based on Geographical Area (Country, State, and City), Qualification, and Company wise or on the Basis of Experience and Expertise wise.

**3. SYSTEM ANALYSIS**

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering), [information systems](http://en.wikipedia.org/wiki/Information_systems) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), is the process of creating or altering systems, and the models and [methodologies](http://en.wikipedia.org/wiki/Methodologies) that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of [software development methodologies](http://en.wikipedia.org/wiki/Software_development_methodologies).

## 3.1 EXISTING SYSTEM

In the existing system all the data or records are maintained manually which takes lot of time. Data can’t be updated or edited. Need special focus on maintaining the records. Misuse or loss of data may cause sometime. Lot of time spent in calculating and checking the grades.

## 3.2 DISADVANTAGES OF EXISTING SYSTEM

* Time consuming
* Error prone
* Large volume of data
* Security
* Retrieval of Information
* Preparation of information between any two dates at any time

## 3.3 PROPOSED SYSTEM

## The proposed system maintains all student information in the database. Information can be entered online. Time can be saved and don’t need any special maintenance. Information once stored can be updated, edited and deleted. Calculation of the grades will be easy.

## 3.4 ADVANTAGES OF PROPOSED SYSTEM

* Menu driven
* Eliminates manual intervention as far as possible
* Error free modification facilities
* On line error modification facilities
* Secured environment
* Immediate report generation for may give two dates
* To maintain security.

**3.5 MODULES**

**User Module:**

#### 4. SOFTWARE REQUIREMENT SPECIFICATION

#### 4.1 Requirements Specification:

#### Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

#### 4.1.1 HARDWARE REQUIREMENTS:

#### The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

#### System : Pentium IV 2.4 GHz.

#### Hard Disk : 100 GB.

#### Monitor : 15 VGA Color.

#### Mouse : Logitech.

#### RAM : 1 GB.

#### 4.1.2 SOFTWARE REQUIREMENTS:

#### The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

#### Operating system : Windows XP/7/10

* Coding Language : Html, JavaScript, Java/J2EE (Jsp Servlet)
* Development Kit : JDK 1.7
* Database : MySQL
* IDE : Netbeans
* Server : Tomcat 7.0

#### 4.2 FUNCTIONAL REQUIREMENTS:

The functional requirement refers to the system needs in an exceedingly computer code engineering method.

The key goal of determinant “functional requirements” in an exceedingly product style and implementation is to capture the desired behavior of a software package in terms of practicality and also the technology implementation of the business processes.

#### 4.3 NON FUNCTIONAL REQUIREMENTS

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system is capable of change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic. Comfortable network information measure may additionally be a non-functional requirement of a system.

**4.4 PERFORMANCE** **REQUIREMENTS**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

**4.5 Feasibility Study:**

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operation Feasibility
* Economical Feasibility

**4.5.1 Technical Feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

**4.5.2 Operational Feasibility**

**User-friendly**

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

**Reliability**

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

**Security**

The web server and database server should be protected from hacking, virus etc

**Portability**

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

**Availability**

This software will be available always.

**Maintainability**

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

**4.5.3 Economic Feasibility**

The computerized system takes care of the present existing system’s data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

Open source software like TOMCAT, JAVA, Mysql and Linux is used to minimize the cost for the Customer.

## 5. System Design

## 5.1 SYSTEM ARCHITECTURE

## 

## The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It’s a serious impact on the later part, notably testing and maintenance.

## The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

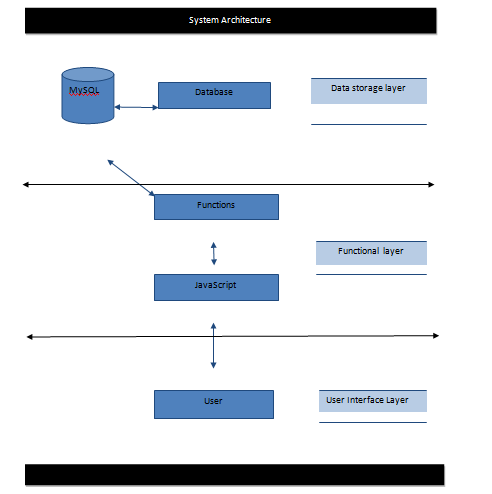
## System Design conjointly referred to as top-ranking style aims to spot the modules that ought to be within the system, the specifications of those modules, and the way them move with one another to supply the specified results.

## At the top of the system style all the main knowledge structures, file formats, output formats, and also the major modules within the system and their specifications square measure set. System design is that the method or art of process the design, components, modules, interfaces, and knowledge for a system to satisfy such as needs. Users will read it because the application of systems theory to development.

## Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced.

## In system design the main target is on distinguishing the modules, whereas throughout careful style the main target is on planning the logic for every of the modules.

## 



## Figure 5.1: Architecture diagram

**5.3 UML DIAGRAMS**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

**User Model View**

This view represents the system from the user’s perspective. The analysis representation describes a usage scenario from the end-users perspective.

**Structural Model view**

In this model the data and functionality are arrived from inside the system. This model view models the static structures.

**Behavioral Model View**

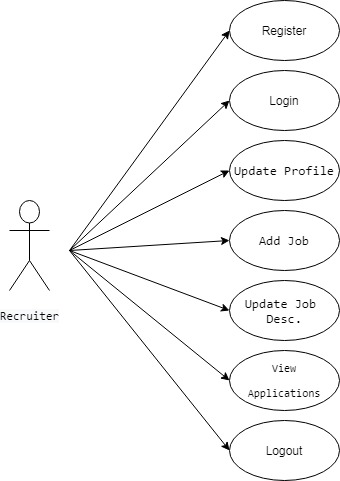
It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

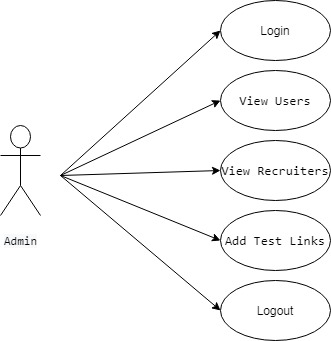
**Implementation Model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

**5.3.1 USE CASE DIAGRAM**

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

****

****

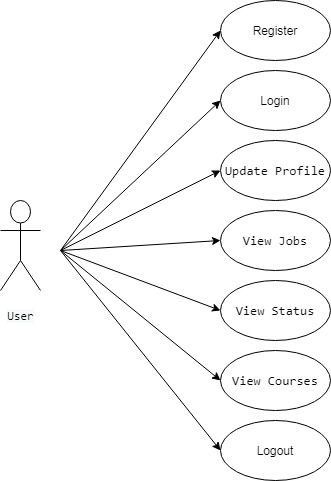
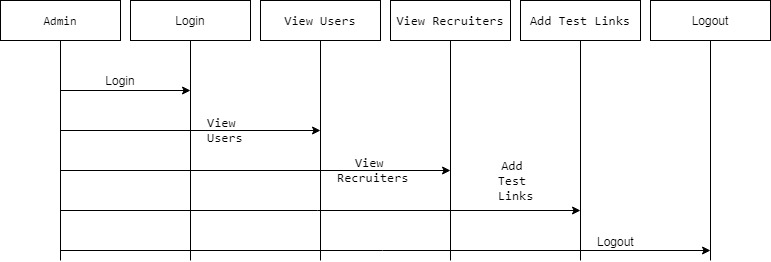
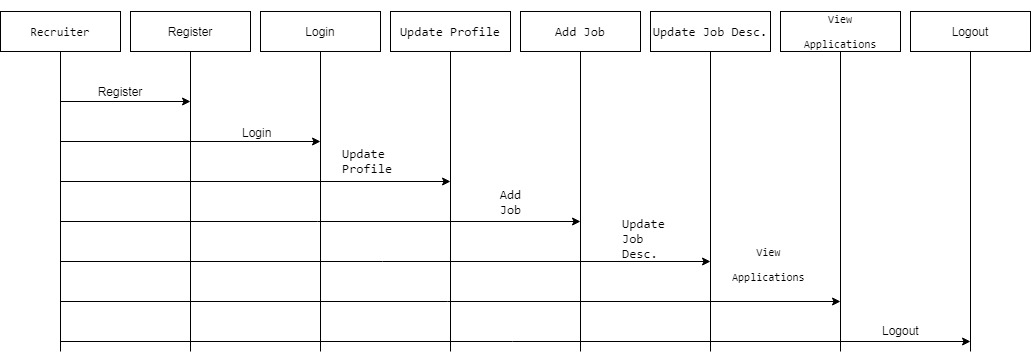
****

Figure 5.3.1 Use Case Diagram

## 5.3.2 SEQUENCEDIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

****

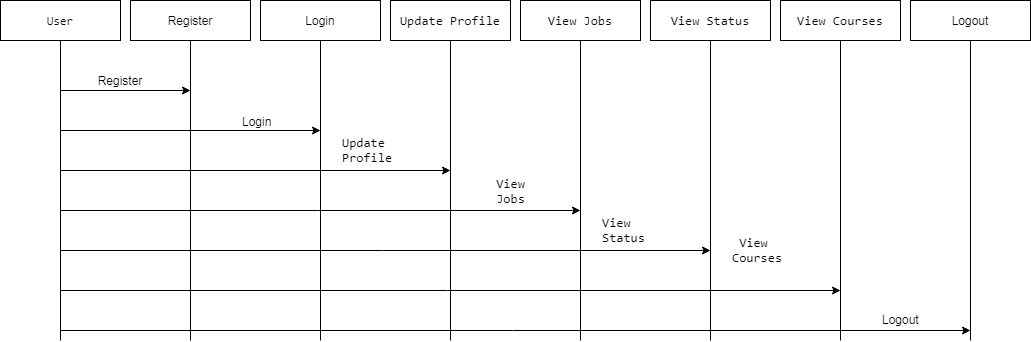
****

Figure 5.3.2: Sequence diagram

## 5.3.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

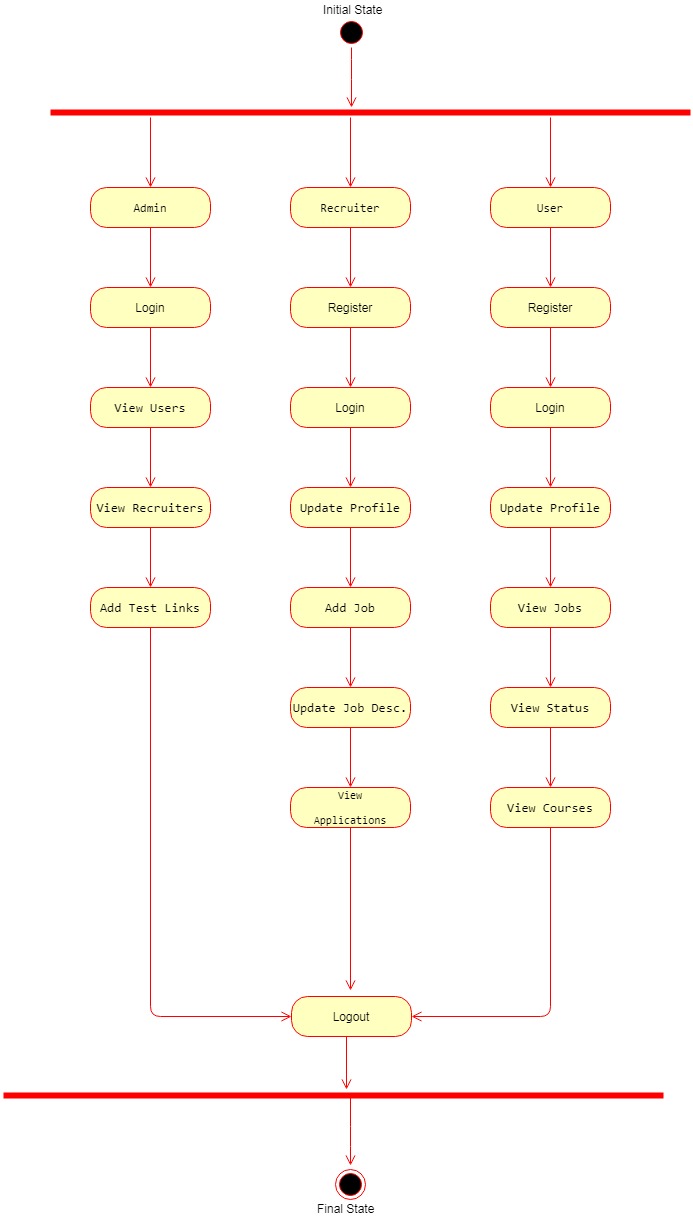
****

Figure 5.3.3: Activity Diagram

## 5.3.3 CLASS DIAGRAM

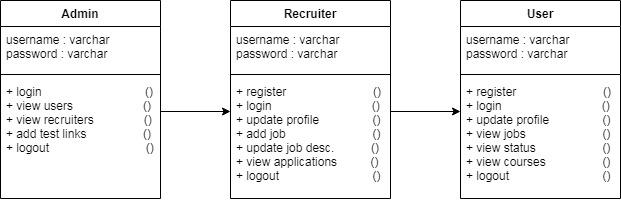
****

Figure 5.3.4: Class Diagram

**6. SOURCE CODE**

**Database Connection Code:**

package databaseconnection;

import java.sql.\*;

public class databasecon

{

static Connection co;

public static Connection getconnection()

{

try

{

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

co = DriverManager.getConnection("jdbc:mysql://localhost:3306/evoting","root","root");

}

catch(Exception e)

{

System.out.println("Database Error"+e);

}

return co;

}

}

**Database Code:**

/\*

SQLyog - Free MySQL GUI v5.13

Host - 5.1.30-community : Database - gradeprocessing

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Server version : 5.1.30-community

\*/

SET NAMES utf8;

SET SQL\_MODE='';

create database if not exists `gradeprocessing`;

USE `gradeprocessing`;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO';

/\*Table structure for table `itc521` \*/

DROP TABLE IF EXISTS `itc521`;

CREATE TABLE `itc521` (

`StudentID` varchar(20) NOT NULL DEFAULT '',

`StudentName` varchar(20) DEFAULT NULL,

`Quiz` varchar(20) DEFAULT NULL,

`Assignment1` varchar(20) DEFAULT NULL,

`Assignment2` varchar(20) DEFAULT NULL,

`Assignment3` varchar(20) DEFAULT NULL,

`Exam` varchar(20) DEFAULT NULL,

`Results` varchar(20) DEFAULT NULL,

`Grade` varchar(20) DEFAULT NULL,

PRIMARY KEY (`StudentID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*Data for the table `itc521` \*/

insert into `itc521` (`StudentID`,`StudentName`,`Quiz`,`Assignment1`,`Assignment2`,`Assignment3`,`Exam`,`Results`,`Grade`) values ('1','Vikas','40','30','30','30','30','30.5','FL');

insert into `itc521` (`StudentID`,`StudentName`,`Quiz`,`Assignment1`,`Assignment2`,`Assignment3`,`Exam`,`Results`,`Grade`) values ('2','34','77','77','77','777','77','77.0','DI');

insert into `itc521` (`StudentID`,`StudentName`,`Quiz`,`Assignment1`,`Assignment2`,`Assignment3`,`Exam`,`Results`,`Grade`) values ('3','88','88','888','88','88','888','648.0','HD');

insert into `itc521` (`StudentID`,`StudentName`,`Quiz`,`Assignment1`,`Assignment2`,`Assignment3`,`Exam`,`Results`,`Grade`) values ('4','4','4','4','4','4','4','4.0','FL');

SET SQL\_MODE=@OLD\_SQL\_MODE;

**Student Java Code:**

package student;

import java.sql.\*;

class Student{

public static void main(String args[]){

try{

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

Connection con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/sonoo","root","password");

//here sonoo is database name, root is username and password

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from emp");

while(rs.next())

System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));

con.close();

}catch(Exception e){ System.out.println(e);}

}

}

**StudentDB Code:**

package student;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

public class StudentDB {

private static String dbName="GradeProcessing";

private static String tbName="ITC521";

private StudentMarks stdMarks;

public StudentMarks getStdMarks() {

return stdMarks;

}

public void setStdMarks(StudentMarks stdMarks) {

this.stdMarks = stdMarks;

}

public Connection getConection() {

Connection con = null;

try {

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

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/"+dbName, "root", "root");

// here sonoo is database name, root is username and password

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return con;

}

//select StudentID,StudentName,Quiz,Assignment1,Assignment2,Assignment3,Exam,Results,Grade

public boolean addStudentToDB(StudentMarks student) {

this.stdMarks=student;

Connection con = getConection();

try {

PreparedStatement ps=con.prepareStatement("insert into "+tbName+" values(?,?,?,?,?,?,?,?,?)");

prepareStamentQueryForStudentMarks(ps, this.stdMarks);

int i=ps.executeUpdate();

con.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return true;

}

public boolean updateStudentToDB(StudentMarks student) {

this.stdMarks=student;

Connection con = getConection();

try {

String updateQuery ="update "+tbName +" set StudentID=?,StudentName=?,Quiz=?,"

+ "Assignment1=?,Assignment2=?,Assignment3=?,Exam=?,Results=?,Grade=? where StudentID=?";

PreparedStatement ps=con.prepareStatement(updateQuery);

prepareStamentQueryForStudentMarks(ps, student);

ps.setString(10, student.getId());

int i=ps.executeUpdate();

con.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return true;

}

public ArrayList<StudentMarks> fetchStdMarksFromDB(String Key,String searchValue){

Connection con = getConection();

ArrayList<StudentMarks> stdMarksList=new ArrayList<StudentMarks>();

String dbKey=matchKeyTODB(Key);

try {

Statement stmt=con.createStatement();

String fetchQuery=null;

if(dbKey == "None"){

fetchQuery="select \* from "+ tbName;

}else{

fetchQuery="select \* from "+ tbName+" where "+dbKey+" = "+searchValue;

}

ResultSet rs=stmt.executeQuery(fetchQuery);

while(rs.next()){

StudentMarks stdMarks=mapStudentWithResultSet(rs);

stdMarksList.add(stdMarks);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return stdMarksList;

}

public StudentMarks mapStudentWithResultSet(ResultSet rs){

StudentMarks stdMarks=null;

try {

stdMarks = new StudentMarks(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getString(6), rs.getString(7), rs.getString(8), rs.getString(9));

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return stdMarks;

}

public void prepareStamentQueryForStudentMarks(PreparedStatement ps,StudentMarks stdMarks) throws SQLException{

ps.setString(1, stdMarks.getId());

ps.setString(2, stdMarks.getName());

ps.setString(3, stdMarks.getQuizmark());

ps.setString(4, stdMarks.getA1());

ps.setString(5, stdMarks.getA2());

ps.setString(6, stdMarks.getA3());

ps.setString(7, stdMarks.getExam());

ps.setString(8, stdMarks.getResults());

ps.setString(9, stdMarks.getGrade());

}

public StudentMarks fetchStudentForID(int id){

Connection con = getConection();

StudentMarks stdMarks=null;

try {

Statement stmt=con.createStatement();

String fetchQuery="select \* from "+ tbName +" where StudentID="+id;

ResultSet rs=stmt.executeQuery(fetchQuery);

if(rs.next()){

stdMarks=mapStudentWithResultSet(rs);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return stdMarks;

}

public String matchKeyTODB(String Key){

if(Key =="StudentID"){

return "StudentID" ;

}else if(Key =="StudentName"){

return "StudentName" ;

}else if(Key =="Quiz Marks"){

return "Quiz" ;

}else if(Key =="Assignment1"){

return "Assignment1" ;

}else if(Key =="Assignment2"){

return "Assignment2" ;

}else if(Key =="Assignment3"){

return "Assignment3" ;

}else if(Key =="Exam"){

return "" ;

}else if(Key =="Results"){

return "Results" ;

}else if(Key =="Grade"){

return "Grade" ;

}

return Key;

}

}

/\*

\* try{ Class.forName("com.mysql.jdbc.Driver"); Connection

\* con=DriverManager.getConnection(

\* "jdbc:mysql://localhost:3306/sonoo","root","password"); //here sonoo is

\* database name, root is username and password Statement

\* stmt=con.createStatement(); ResultSet

\* rs=stmt.executeQuery("select \* from emp"); while(rs.next())

\* System.out.println(rs.getString(1)+" "+rs.getString(2)+" "+rs.getString(3)

\* ); con.close(); }catch(Exception e){ System.out.println(e);} }

\*/

**StudentMarks page:**

package student;

public class StudentMarks {

private String name;

private String id;

private String quizmark;

private String a1;

private String a2;

private String a3;

private String exam;

private String results;

private String grade;

public StudentMarks(String id,String name, String quizmark, String a1, String a2, String a3,String exam, String results, String grade)

{

this.name=name;

this.id=id;

this.quizmark=quizmark;

this.a1=a1;

this.a2=a2;

this.a3=a3;

this.exam=exam;

this.results=results;

this.grade=grade;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getQuizmark() {

return quizmark;

}

public void setQuizmark(String quizmark) {

this.quizmark = quizmark;

}

public String getA1() {

return a1;

}

public void setA1(String a1) {

this.a1 = a1;

}

public String getA2() {

return a2;

}

public void setA2(String a2) {

this.a2 = a2;

}

public String getA3() {

return a3;

}

public void setA3(String a3) {

this.a3 = a3;

}

public String getExam() {

return exam;

}

public void setExam(String exam) {

this.exam = exam;

}

public String getResults() {

return results;

}

public void setResults(String results) {

this.results = results;

}

public String getGrade() {

return grade;

}

public void setGrade(String grade) {

this.grade = grade;

}

}

**Grade Processing Java Code:**

package grade;

import java.util.ArrayList;

import java.util.Iterator;

import javafx.application.Application;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import student.StudentDB;

import student.StudentMarks;

public class GradeProcessing extends Application {

private TableView<StudentMarks> table;

private TableView<StudentMarks> searchTable;

private ObservableList<StudentMarks> data = FXCollections.observableArrayList();

private ObservableList<StudentMarks> searchData = FXCollections.observableArrayList();

//double res1 = 0;

public static void main(String[] args) {

// TODO Auto-generated method stub

launch(args);

}

@SuppressWarnings("unchecked")

public void start(Stage primaryStage) {

Label stidLB = new Label("Student ID");

TextField stidTF = new TextField();

Button btn = new Button();

btn.setText("Fetch");

HBox stidHB = new HBox(stidLB, stidTF,btn);

stidHB.setSpacing(28);

stidHB.setAlignment(Pos.CENTER);

Label stdnameLB = new Label("Student Name");

TextField stdnameTF = new TextField();

HBox stdnameHB = new HBox(stdnameLB, stdnameTF);

stdnameHB.setSpacing(28);

stdnameHB.setAlignment(Pos.CENTER);

Label quizLB = new Label("Quiz marks");

TextField quizTF = new TextField();

HBox quizHB = new HBox(quizLB, quizTF);

quizHB.setSpacing(50);

quizHB.setAlignment(Pos.CENTER);

Label assign1LB = new Label("Assignment 1");

TextField assign1TF = new TextField();

HBox assign1HB = new HBox(assign1LB, assign1TF);

assign1HB.setSpacing(3);

assign1HB.setAlignment(Pos.CENTER);

Label assign2LB = new Label("Assignment 2");

TextField assign2TF = new TextField();

HBox assign2HB = new HBox(assign2LB, assign2TF);

assign2HB.setSpacing(3);

assign2HB.setAlignment(Pos.CENTER);

Label assign3LB = new Label("Assignment 3");

TextField assign3TF = new TextField();

HBox assign3HB = new HBox(assign3LB, assign3TF);

assign3HB.setSpacing(3);

assign3HB.setAlignment(Pos.CENTER);

Label examLB = new Label(" Final Exam Marks");

TextField examTF = new TextField();

Button BtnInsert = new Button();

BtnInsert.setText("Insert");

Button BtnUpdate = new Button();

BtnUpdate.setText("Update");

BtnUpdate.setDisable(true);

Button BtnReset = new Button();

BtnReset.setText("ReSet");

HBox examHB = new HBox(examLB, examTF,BtnInsert,BtnUpdate,BtnReset);

examHB.setSpacing(10);

examHB.setAlignment(Pos.CENTER);

btn.setOnAction(new EventHandler<ActionEvent>(){

@Override

public void handle(ActionEvent event) {

// TODO Auto-generated method stub

StudentMarks stdMarks1=new StudentDB().fetchStudentForID(Integer.parseInt(stidTF.getText()));

if(stdMarks1 !=null){

stdnameTF.setText(stdMarks1.getName());

quizTF.setText(String.valueOf(stdMarks1.getQuizmark()));

assign1TF.setText(String.valueOf(stdMarks1.getA1()));

assign2TF.setText(String.valueOf(stdMarks1.getA2()));

assign3TF.setText(String.valueOf(stdMarks1.getA3()));

examTF.setText(String.valueOf(stdMarks1.getExam()));

stidTF.setDisable(true);

BtnUpdate.setDisable(false);

BtnInsert.setDisable(true);

}

}

});

BtnReset.setOnAction(new EventHandler<ActionEvent>(){

@Override

public void handle(ActionEvent event) {

// TODO Auto-generated method stub

BtnUpdate.setDisable(true);

BtnInsert.setDisable(false);

stidTF.setDisable(false);

stdnameTF.clear();

stidTF.clear();

quizTF.clear();

assign1TF.clear();

assign2TF.clear();

assign3TF.clear();

examTF.clear();

}

});

VBox stdVB = new VBox(stidHB, stdnameHB, quizHB, assign1HB, assign2HB,assign3HB, examHB);

stdVB.setSpacing(3);

table = new TableView<>();

searchTable = new TableView<>();

TableColumn<StudentMarks, String> name = new TableColumn<>("Name");

name.setCellValueFactory(new PropertyValueFactory<StudentMarks, String>("name"));

TableColumn<StudentMarks, Integer> id = new TableColumn<>("ID");

id.setCellValueFactory(new PropertyValueFactory<StudentMarks, Integer>("id"));

TableColumn<StudentMarks, Integer> quizmark = new TableColumn<>("Quiz Mark");

quizmark.setCellValueFactory(new PropertyValueFactory<StudentMarks, Integer>("quizmark"));

TableColumn<StudentMarks, Integer> a1 = new TableColumn<>("A1");

a1.setCellValueFactory(new PropertyValueFactory<StudentMarks, Integer>("a1"));

TableColumn<StudentMarks, Integer> a2 = new TableColumn<>("A2");

a2.setCellValueFactory(new PropertyValueFactory<StudentMarks, Integer>("a2"));

TableColumn<StudentMarks, Integer> a3 = new TableColumn<>("A3");

a3.setCellValueFactory(new PropertyValueFactory<StudentMarks, Integer>("a3"));

TableColumn<StudentMarks, Integer> exam = new TableColumn<>("Exam");

exam.setCellValueFactory(new PropertyValueFactory<StudentMarks, Integer>("exam"));

TableColumn<StudentMarks, Double> res = new TableColumn<>("Results");

res.setCellValueFactory(new PropertyValueFactory<StudentMarks, Double>("results"));

TableColumn<StudentMarks, String> gde = new TableColumn<>("Grade");

gde.setCellValueFactory(new PropertyValueFactory<StudentMarks, String>("grade"));

table.getColumns().addAll(name, id, quizmark, a1, a2,a3, exam, res, gde);

searchTable.getColumns().addAll(name, id, quizmark, a1, a2,a3, exam, res, gde);

// data=FXCollections.observableArrayList();

table.getItems().clear();

table.setItems(data);

table.setMaxHeight(200);

searchTable.setMaxHeight(200);

searchTable.setItems(searchData);

Label avgLB = new Label("Average Marks");

TextField avgTF = new TextField();

Button avgBT = new Button("Average Marks");

Button stdBT = new Button("Student Marks");

HBox btHB = new HBox(avgLB, avgTF, avgBT, stdBT);

btHB.setSpacing(3);

btHB.setAlignment(Pos.CENTER);

BtnInsert.setOnAction((ActionEvent e) -> {

String sname = stdnameTF.getText();

int sid = Integer.parseInt(stidTF.getText());

int squizmark = Integer.parseInt(quizTF.getText());

int sa1 = Integer.parseInt(assign1TF.getText());

int sa2 = Integer.parseInt(assign2TF.getText());

int sa3 = Integer.parseInt(assign3TF.getText());

int sexam = Integer.parseInt(examTF.getText());

double results = (squizmark \* 0.05) + (sa1 \* 0.20) + (sa2 \* 0.25) + (sexam \* 0.5);

String grade = null;

if (results >= 85) {

grade = "HD";

} else if (results >= 75 && results < 85) {

grade = "DI";

} else if (results >= 65 && results < 75) {

grade = "CR";

} else if (results >= 50 && results < 65) {

grade = "PS";

} else {

grade = "FL";

}

StudentMarks stdMarks = new StudentMarks(String.valueOf(sid),String.valueOf(sname), String.valueOf(squizmark),

String.valueOf(sa1), String.valueOf(sa2),String.valueOf(sa3), String.valueOf(sexam),

String.valueOf(results), String.valueOf(grade));

data.add(stdMarks);

new StudentDB().addStudentToDB(stdMarks);

stdnameTF.clear();

stidTF.clear();

quizTF.clear();

assign1TF.clear();

assign2TF.clear();

assign3TF.clear();

examTF.clear();

});

BtnUpdate.setOnAction((ActionEvent e) -> {

String sname = stdnameTF.getText();

int sid = Integer.parseInt(stidTF.getText());

int squizmark = Integer.parseInt(quizTF.getText());

int sa1 = Integer.parseInt(assign1TF.getText());

int sa2 = Integer.parseInt(assign2TF.getText());

int sa3 = Integer.parseInt(assign3TF.getText());

int sexam = Integer.parseInt(examTF.getText());

double results = (squizmark \* 0.05) + (sa1 \* 0.20) + (sa2 \* 0.25) + (sexam \* 0.5);

String grade = null;

if (results >= 85) {

grade = "HD";

} else if (results >= 75 && results < 85) {

grade = "DI";

} else if (results >= 65 && results < 75) {

grade = "CR";

} else if (results >= 50 && results < 65) {

grade = "PS";

} else {

grade = "FL";

}

StudentMarks stdMarks = new StudentMarks(String.valueOf(sid),String.valueOf(sname), String.valueOf(squizmark),

String.valueOf(sa1), String.valueOf(sa2),String.valueOf(sa3), String.valueOf(sexam),

String.valueOf(results), String.valueOf(grade));

data.add(stdMarks);

new StudentDB().updateStudentToDB(stdMarks);

stdnameTF.clear();

stidTF.clear();

quizTF.clear();

assign1TF.clear();

assign2TF.clear();

assign3TF.clear();

examTF.clear();

});

ObservableList<String> searchOptions =

FXCollections.observableArrayList(

"None","Name","StudentID","Quiz Marks",

"Assignment 1","Assignment 2","Assignment 3",

"Average","Grade"

);

final ComboBox comboBox = new ComboBox(searchOptions);

TextField searchTxtBox = new TextField();

HBox searchBox = new HBox(searchTxtBox,comboBox);

comboBox.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent e) {

ArrayList<StudentMarks> stdMarksList =new StudentDB().fetchStdMarksFromDB((String)comboBox.getValue(),searchTxtBox.getText()) ;

Iterator<StudentMarks> it=stdMarksList.iterator();

searchTable.getItems().clear();

searchData.clear();

while( it.hasNext()) {

StudentMarks std = (StudentMarks) it.next();

searchData.add(std);

}

}

});

/\* avgBT.setOnAction((ActionEvent e) -> {

double res1 = 0;

for (StudentMarks std : data) {

res1 = res1 + std.getResults();

}

double avgMarks = res1 / (data.size());

avgTF.setText((Double.toString(avgMarks)));

});

\*/

final VBox vbox = new VBox();

vbox.setSpacing(5);

vbox.getChildren().addAll(stdVB, table, searchBox,searchTable);

Scene scene = new Scene(vbox, 633, 458);

primaryStage.setTitle("Grade Proccessing");

primaryStage.setScene(scene);

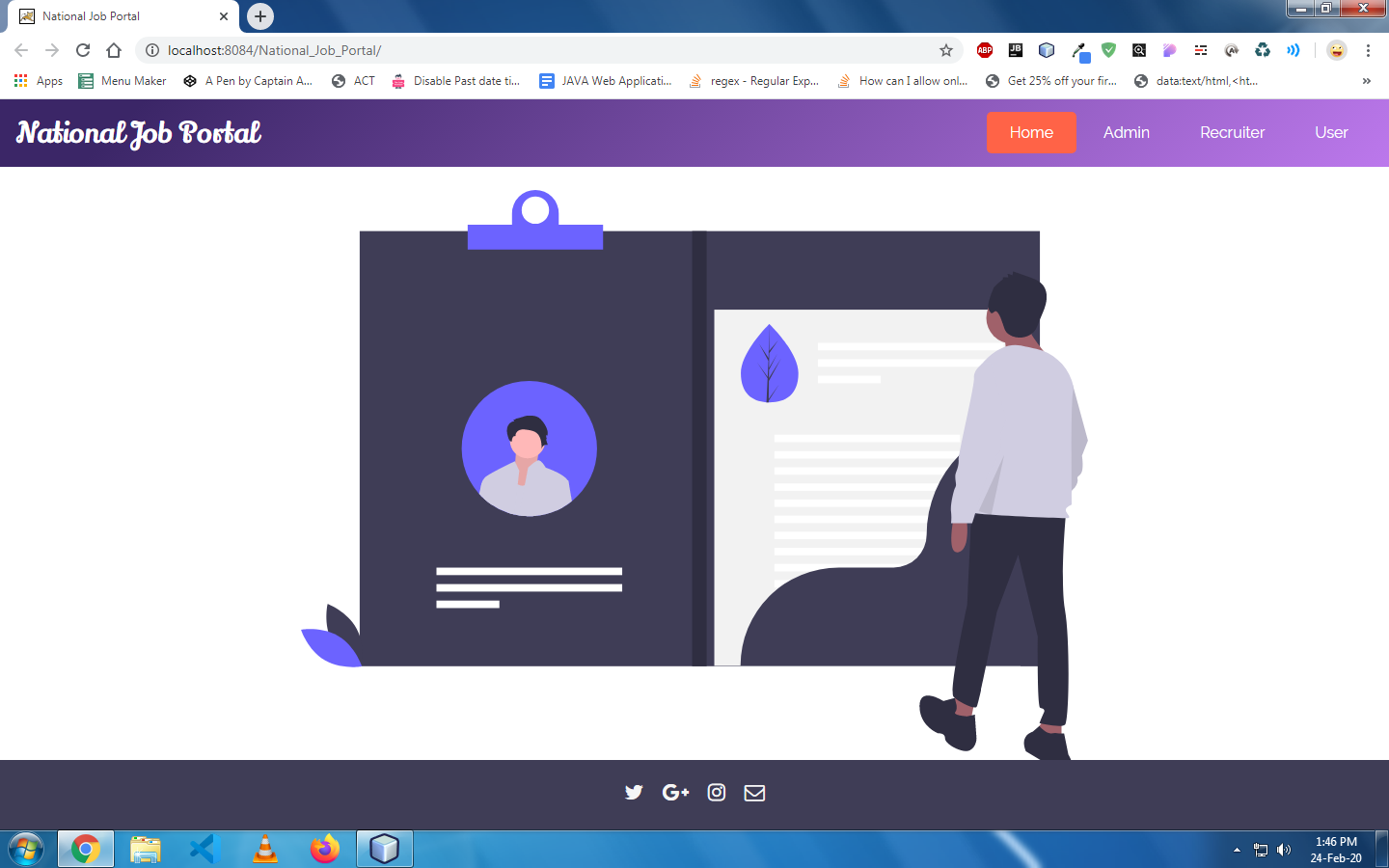
primaryStage.setResizable(false);

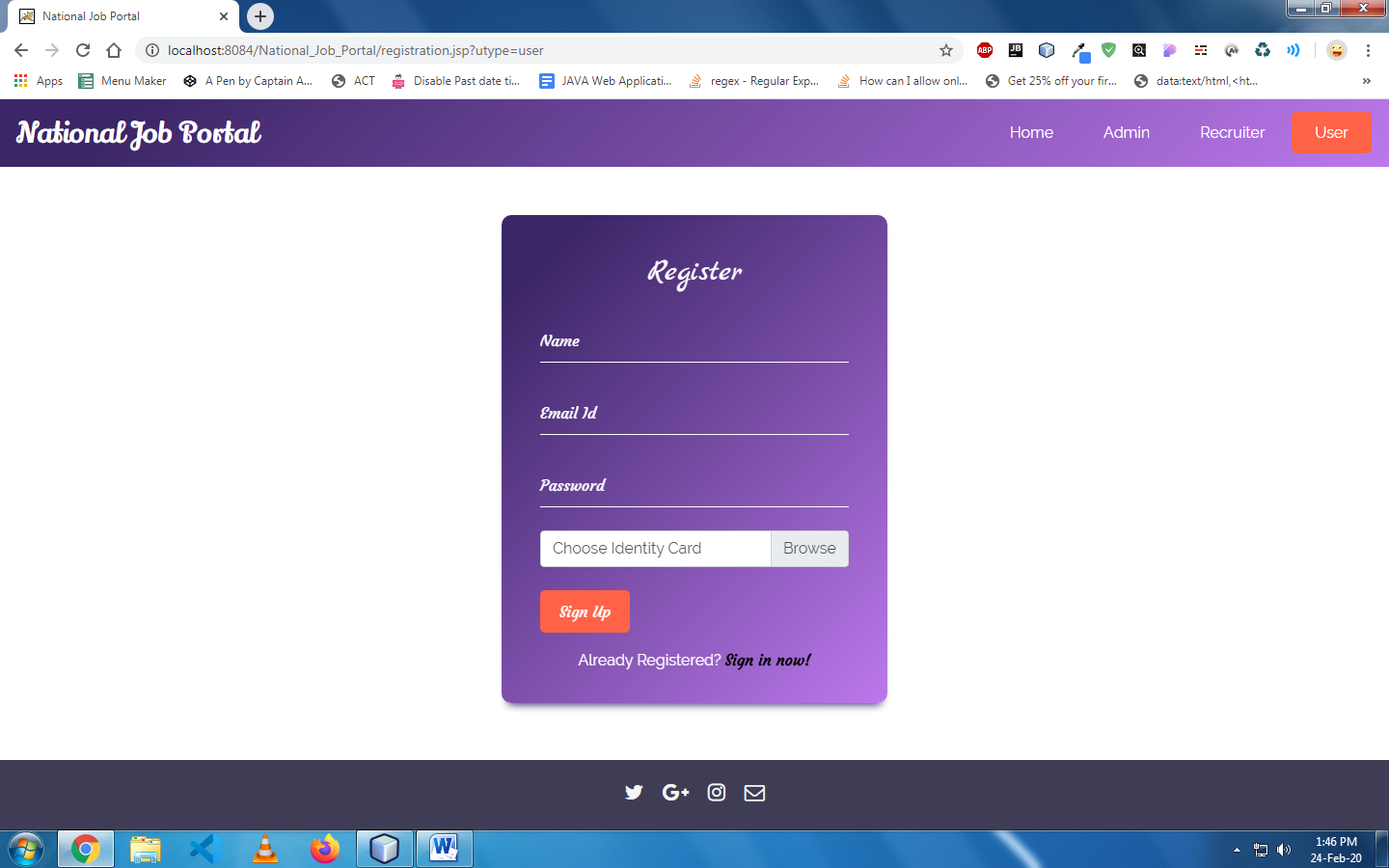
primaryStage.show();

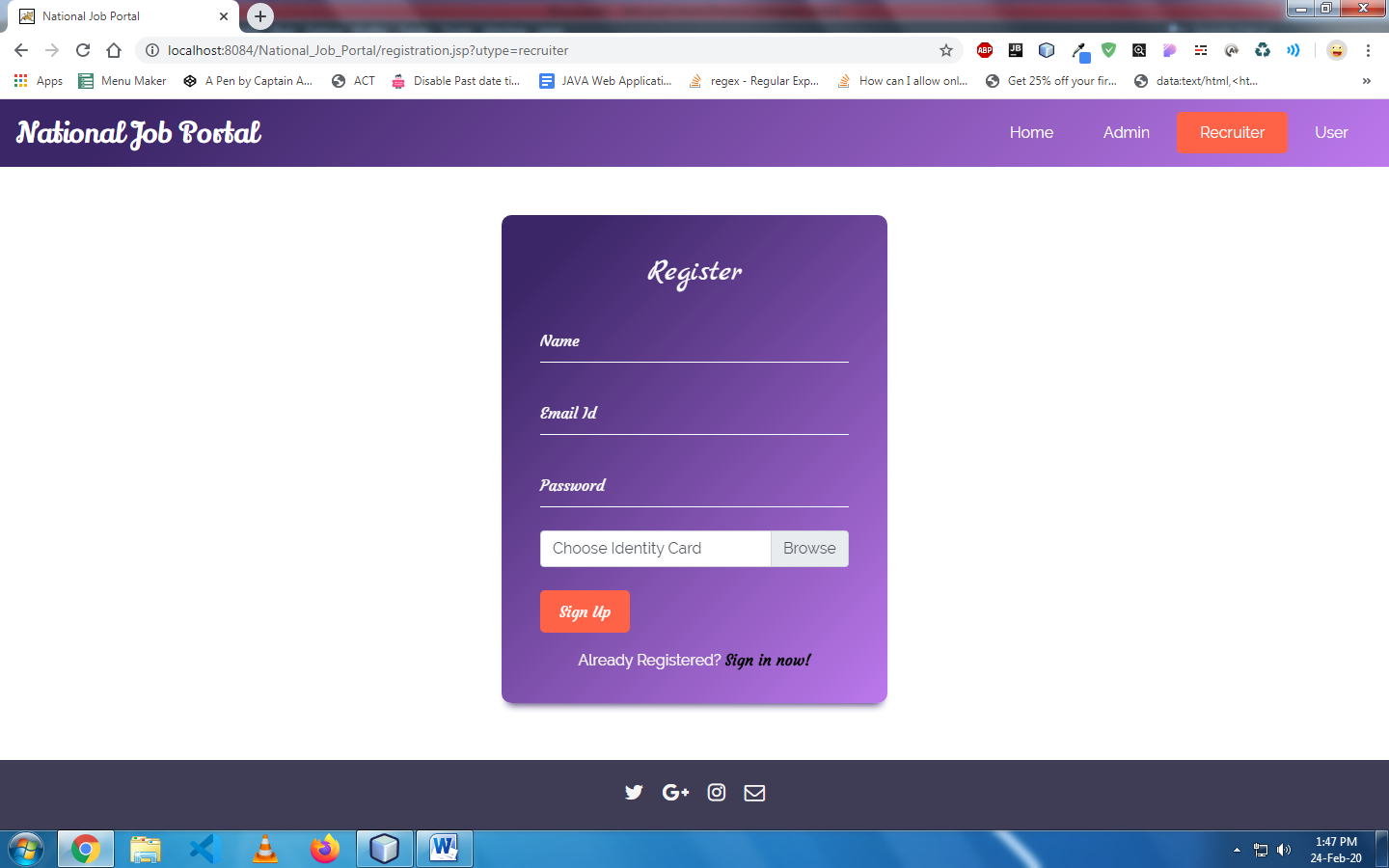
}

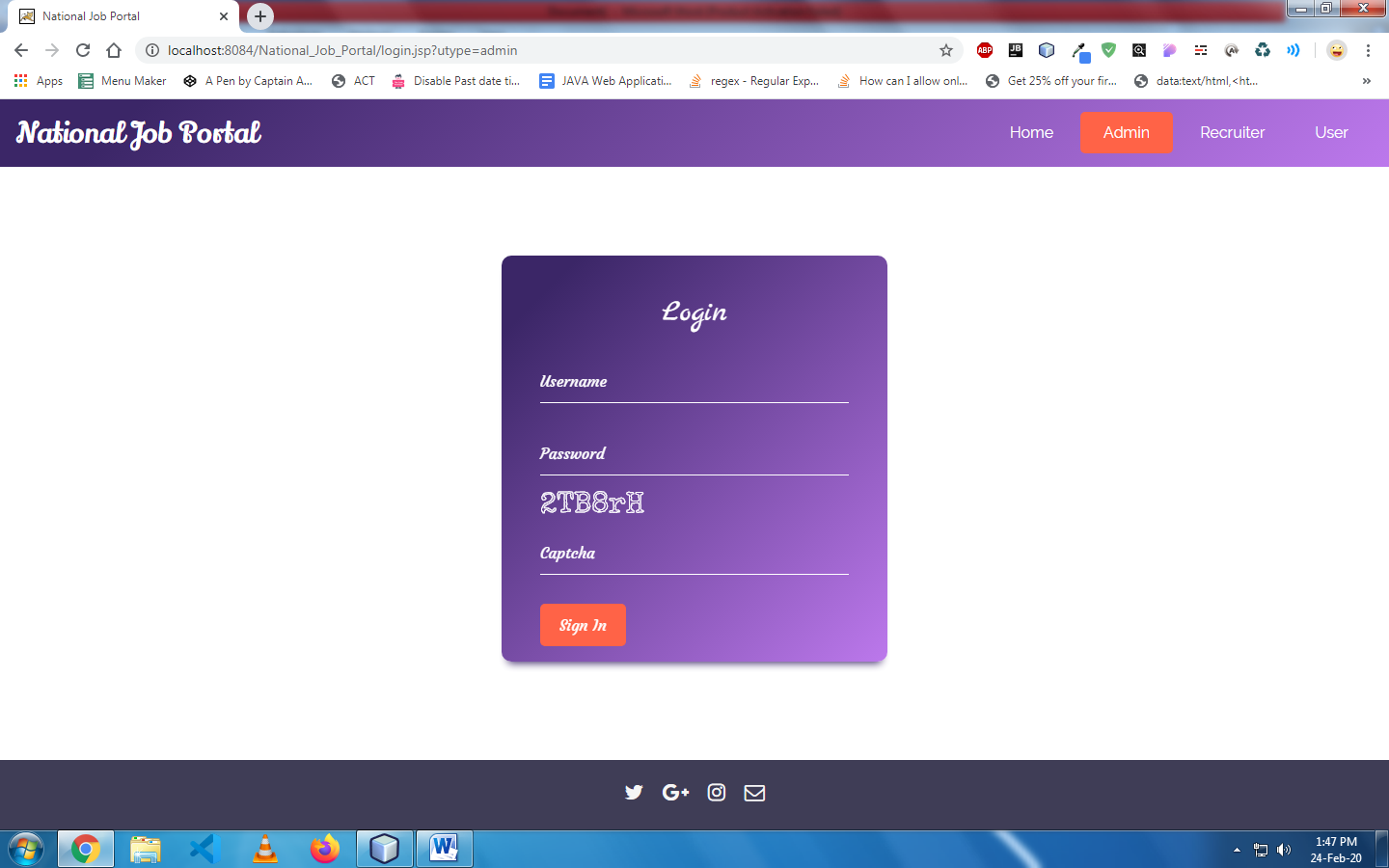
}

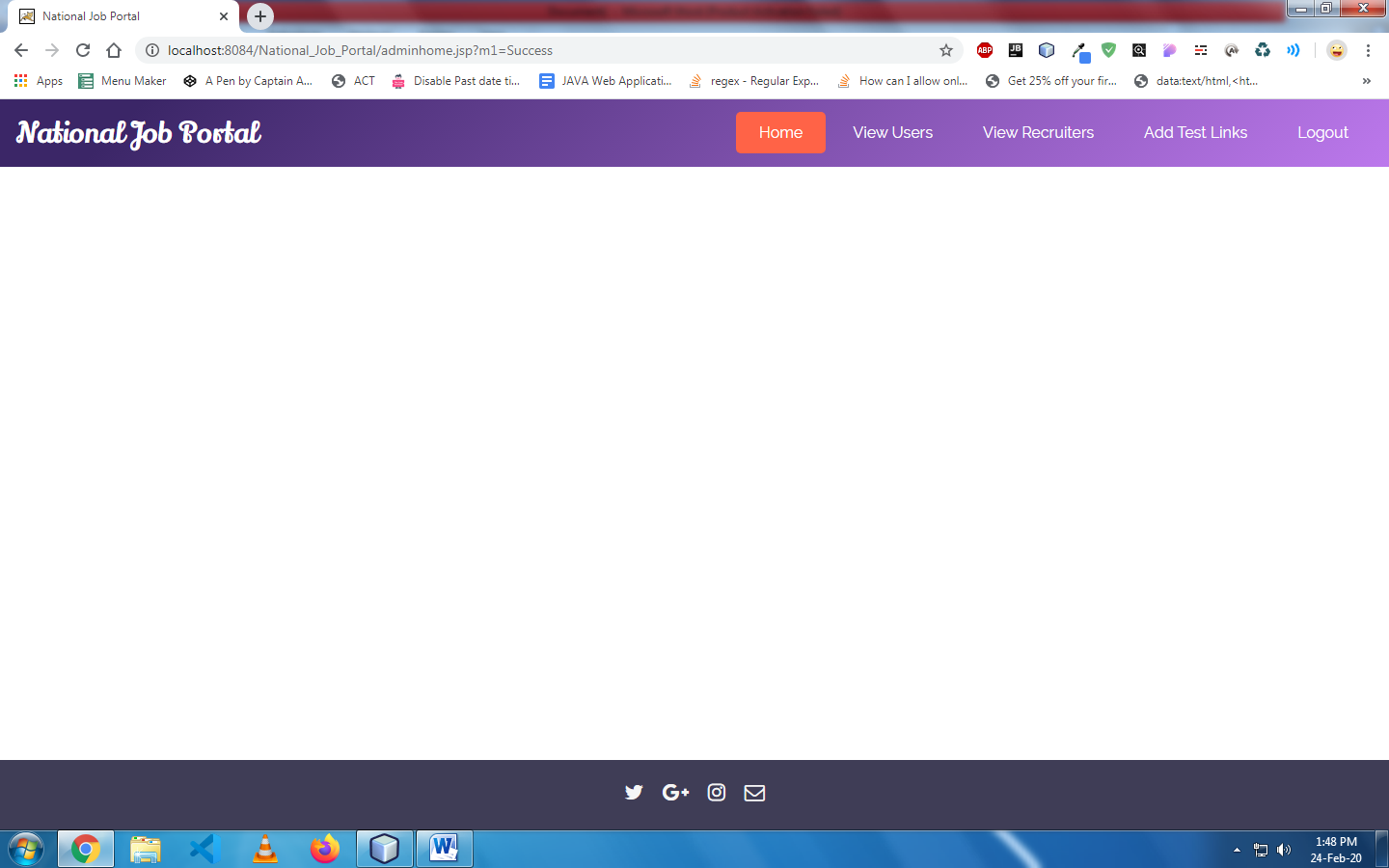
**7. SCREEN SHOTS**

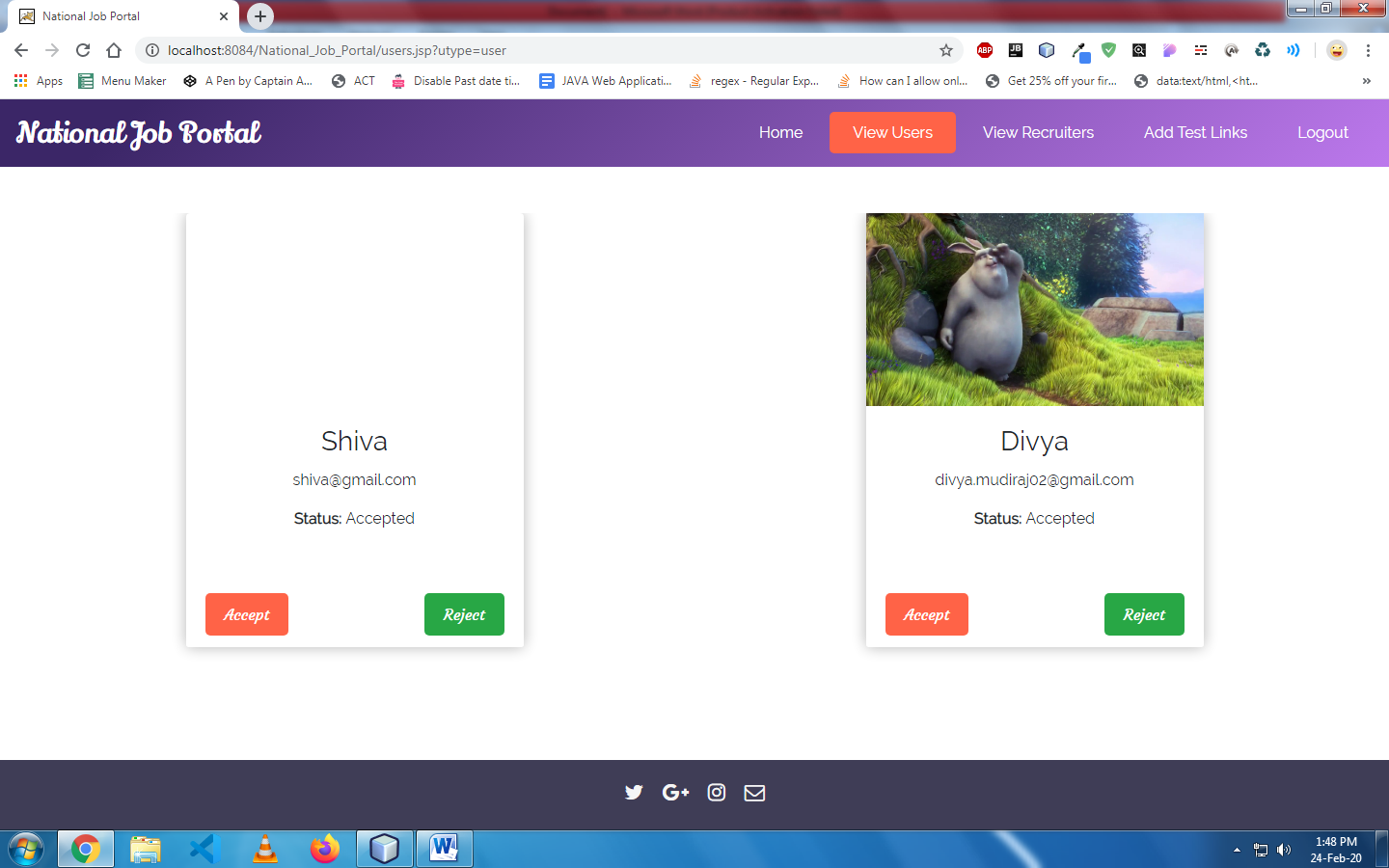


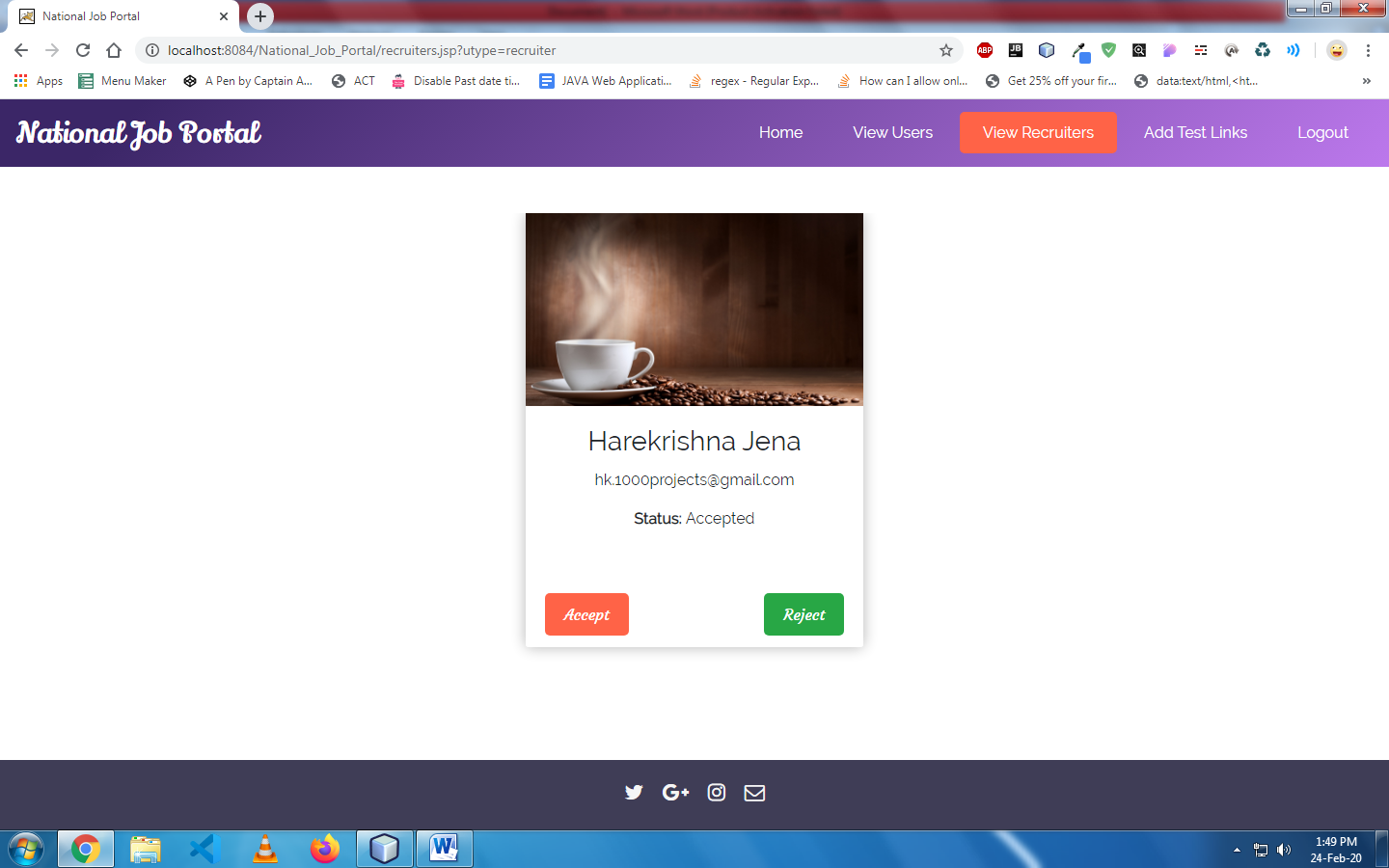


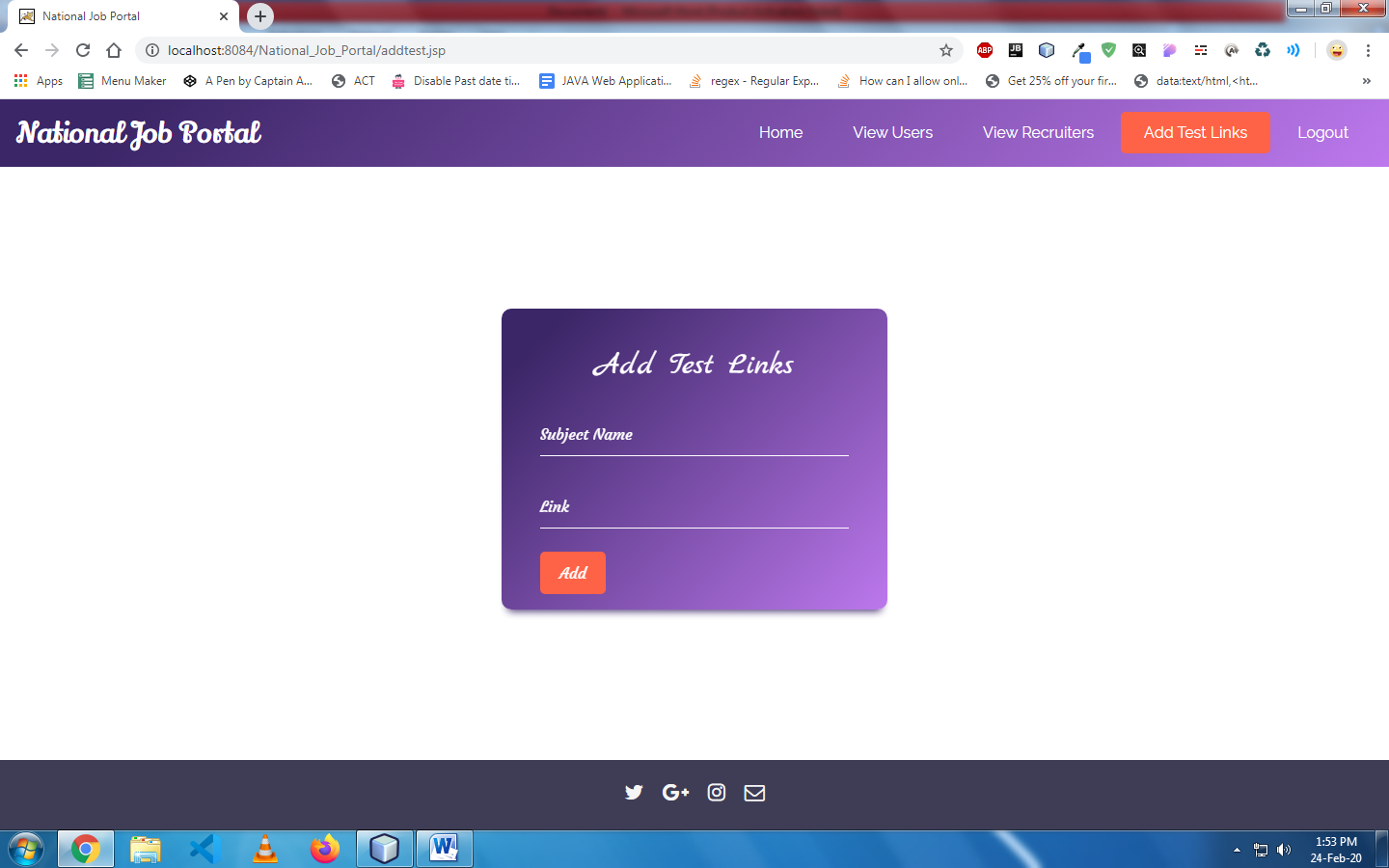


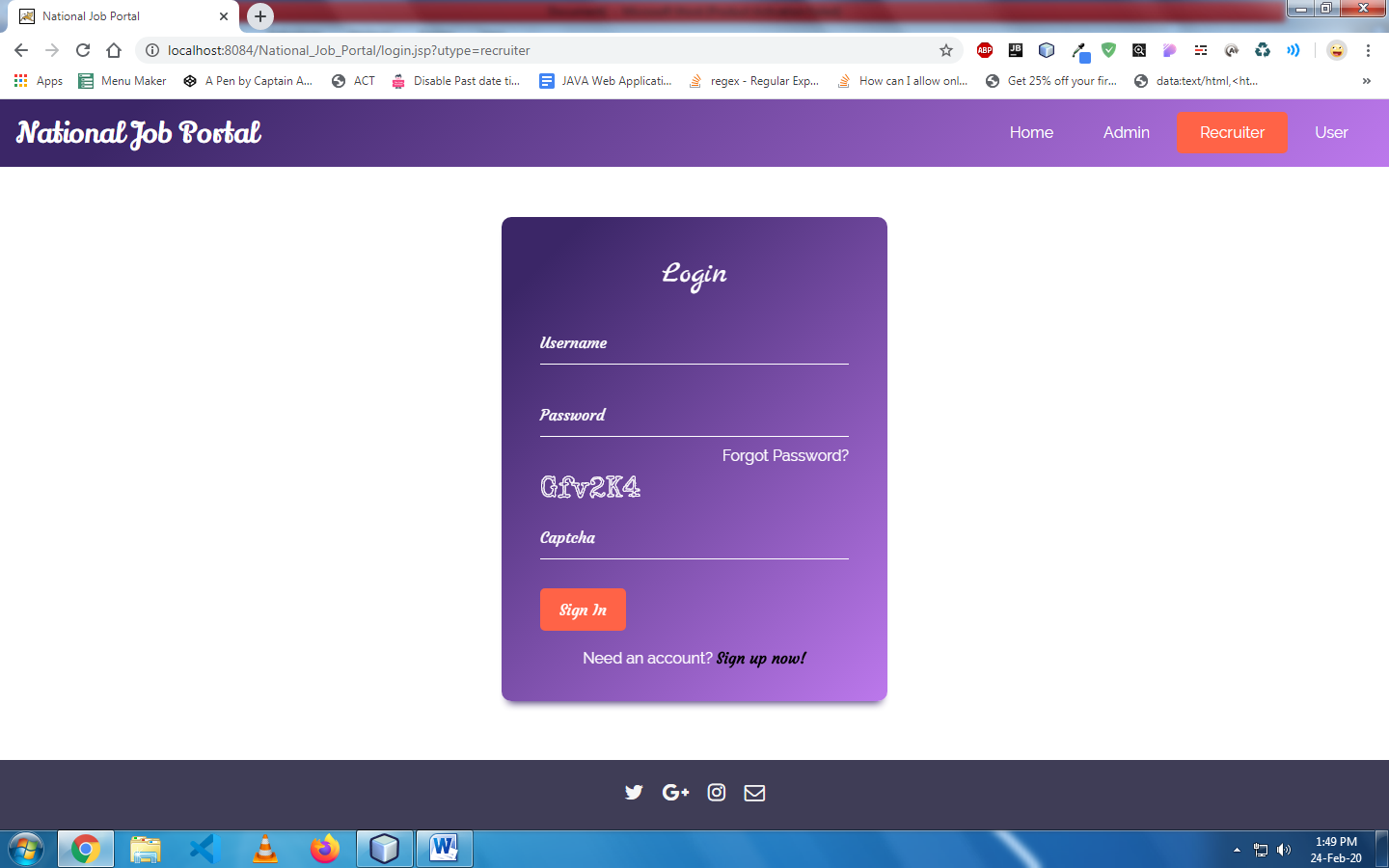


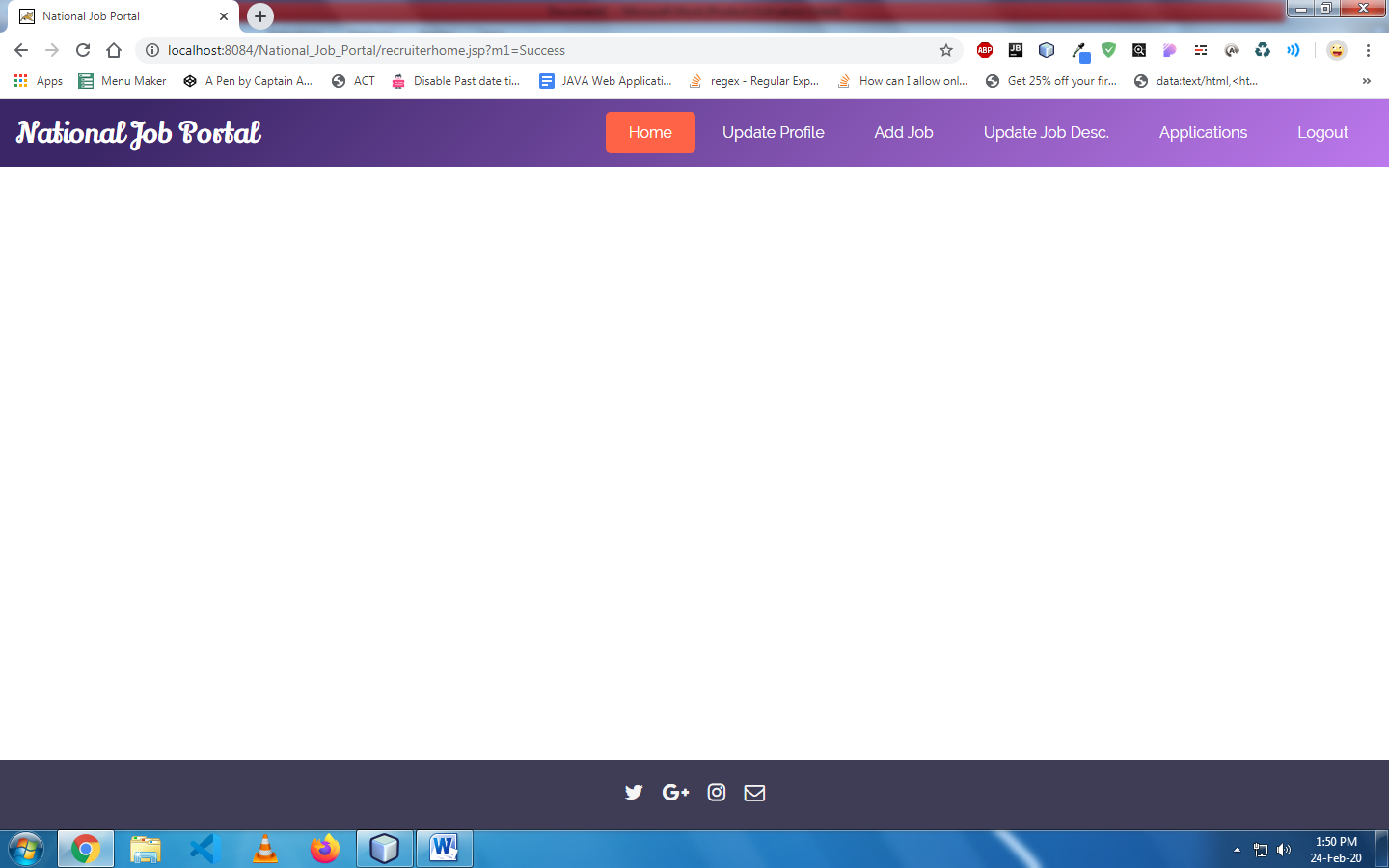


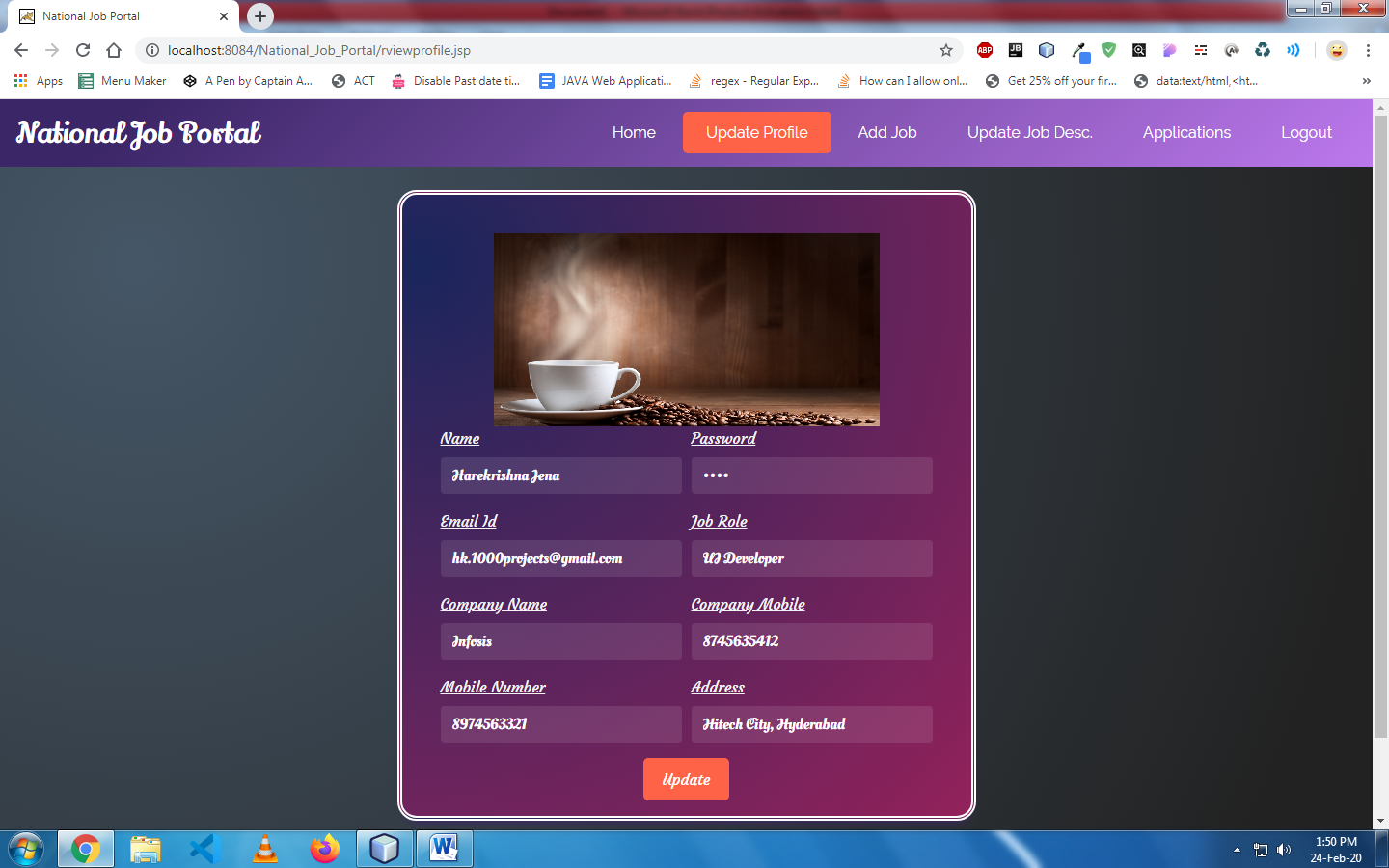


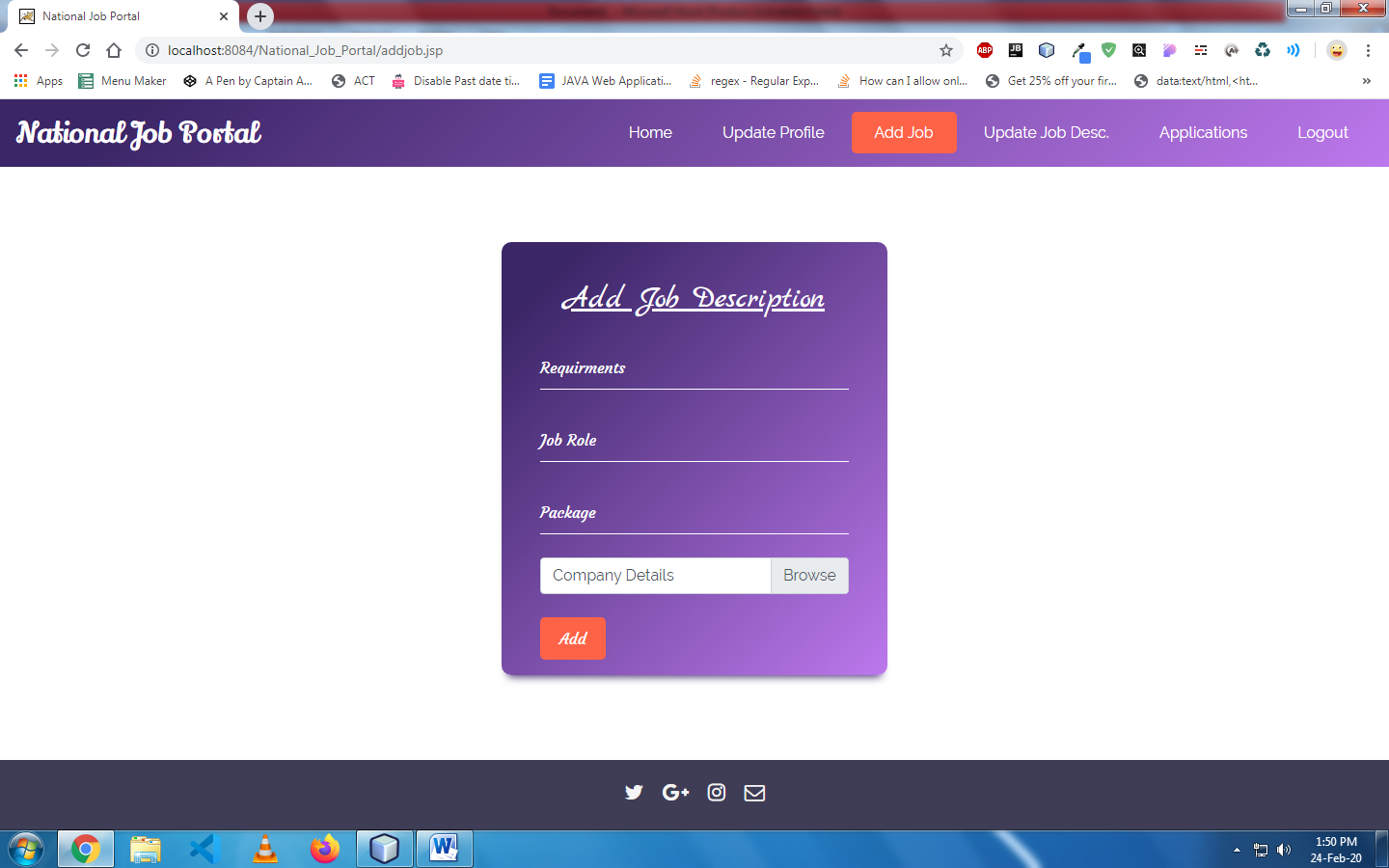






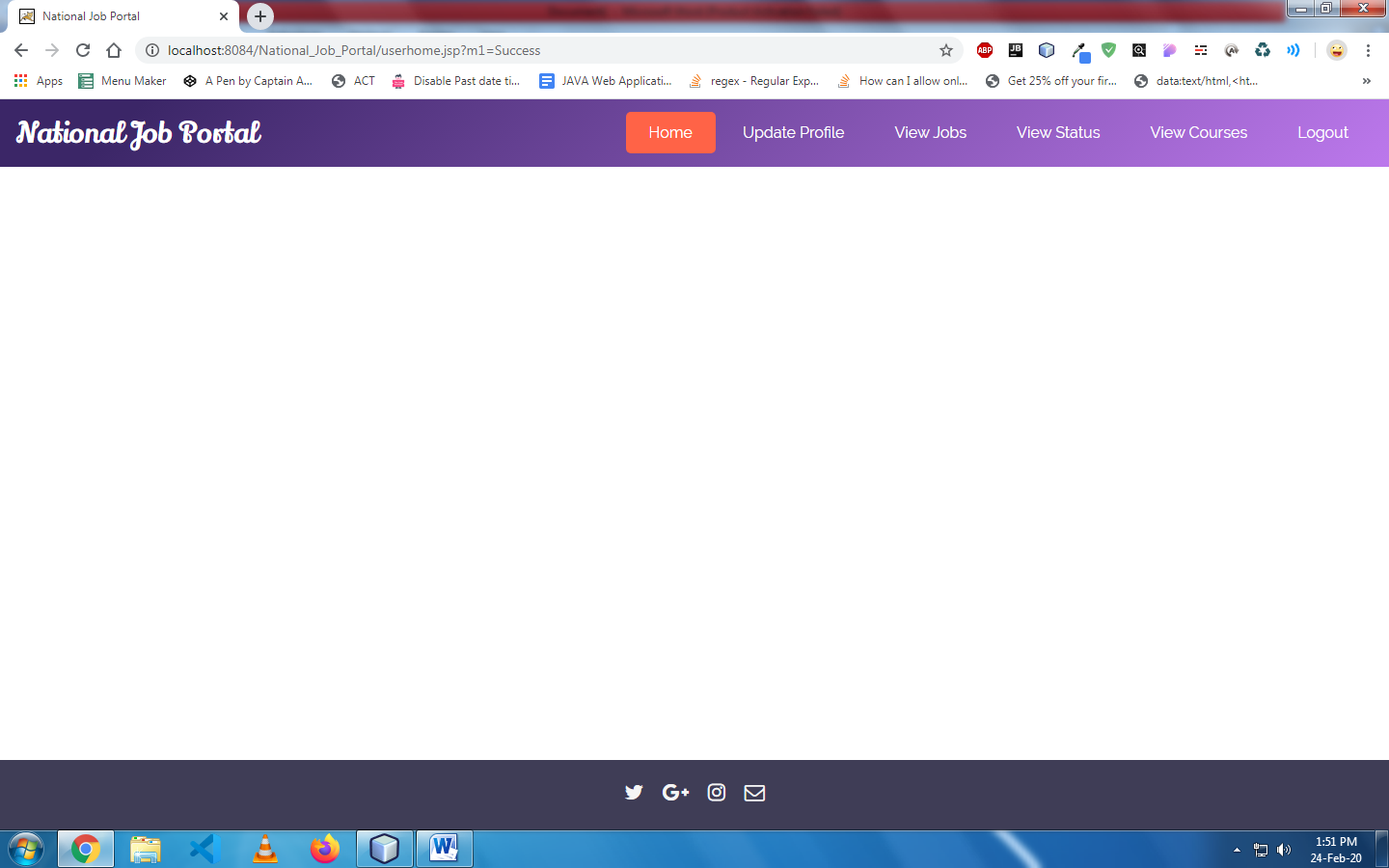


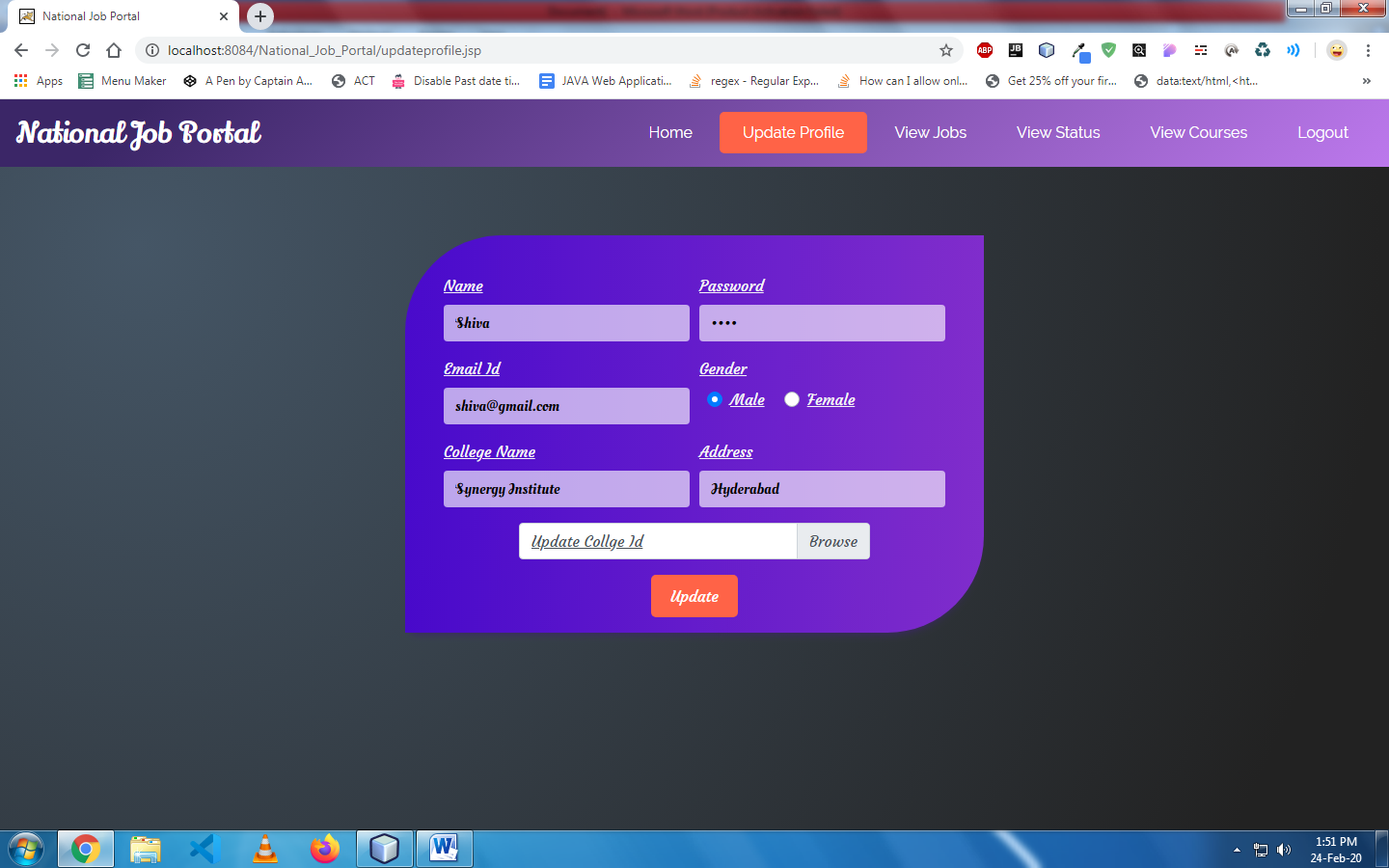


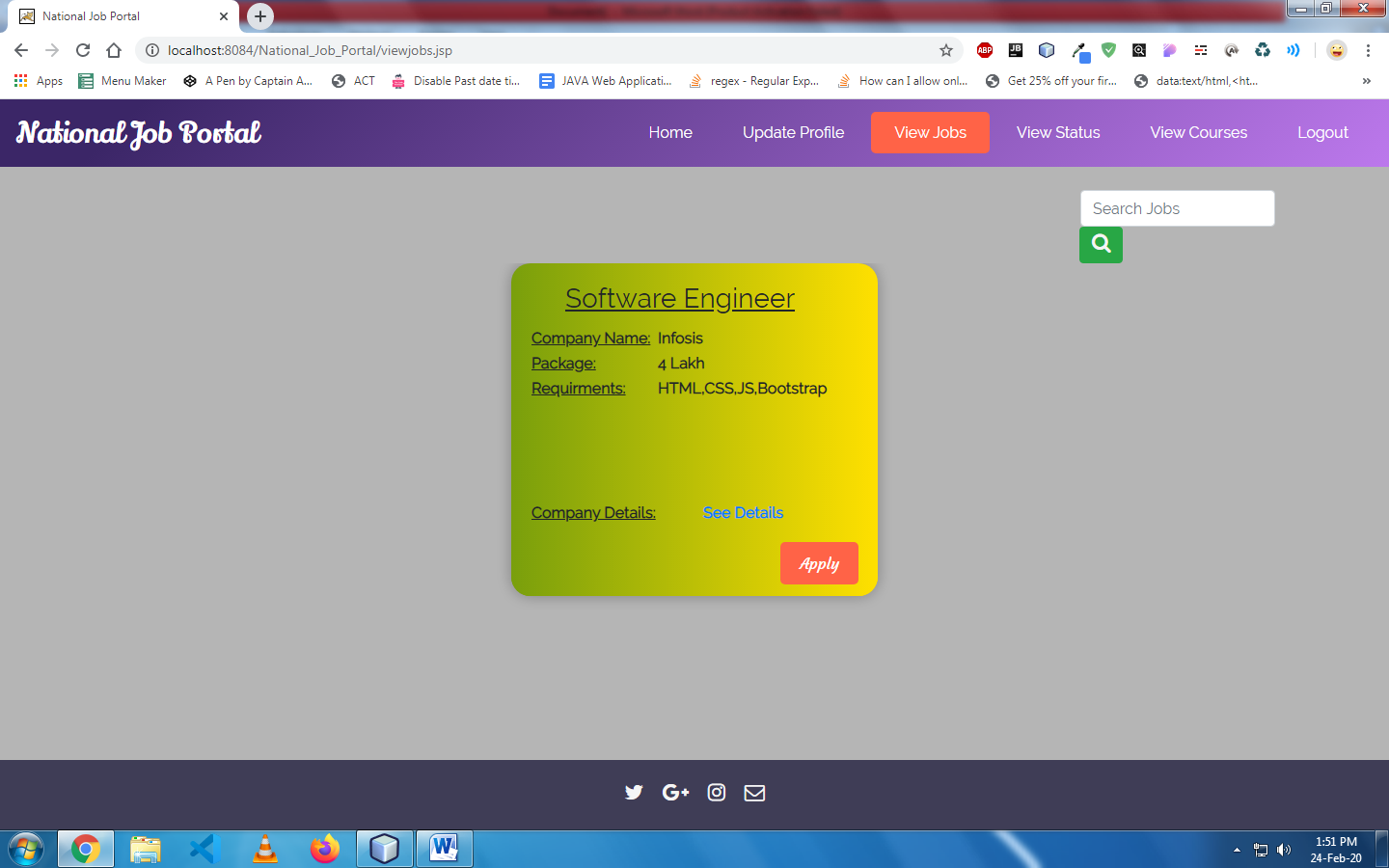


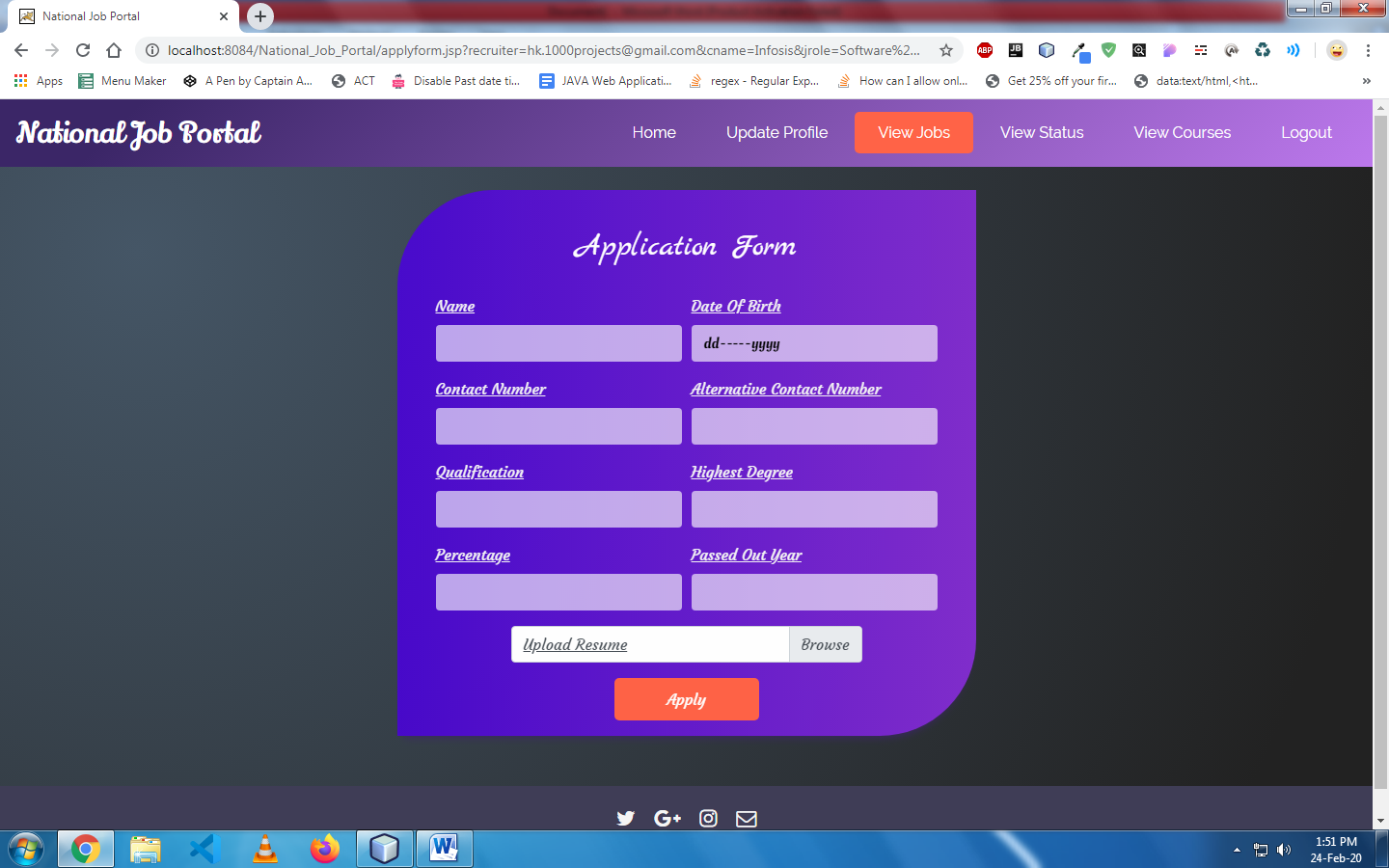


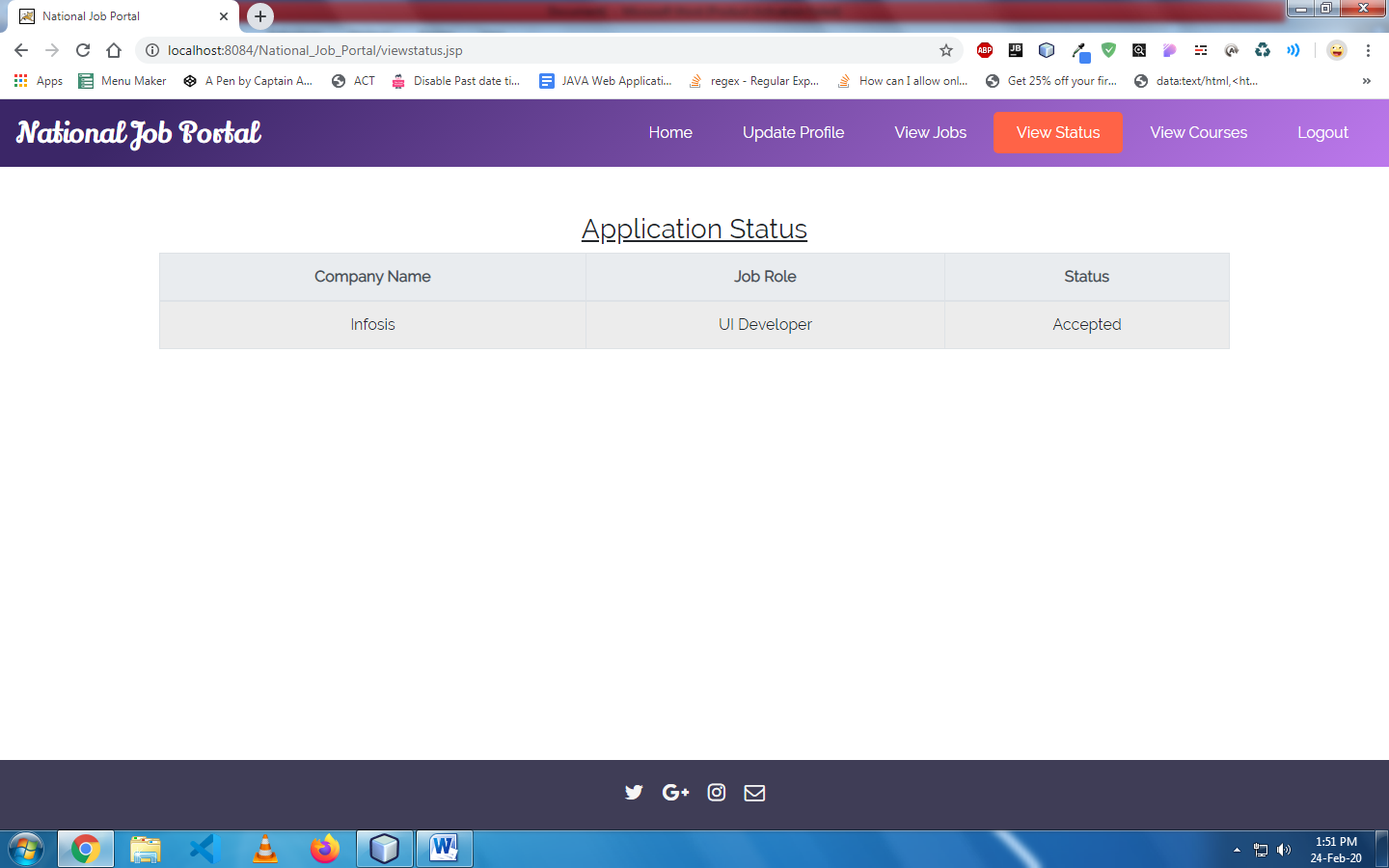


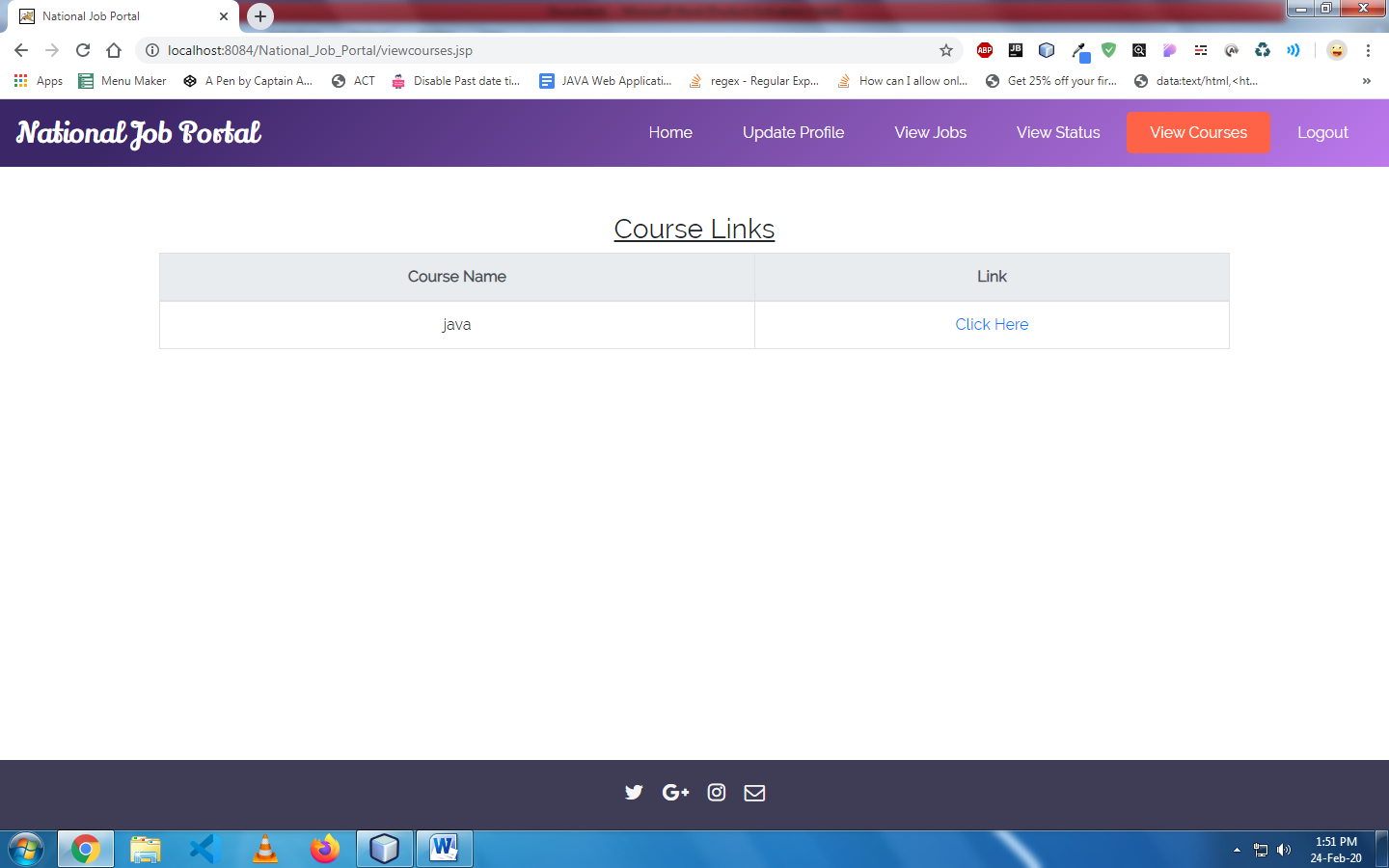


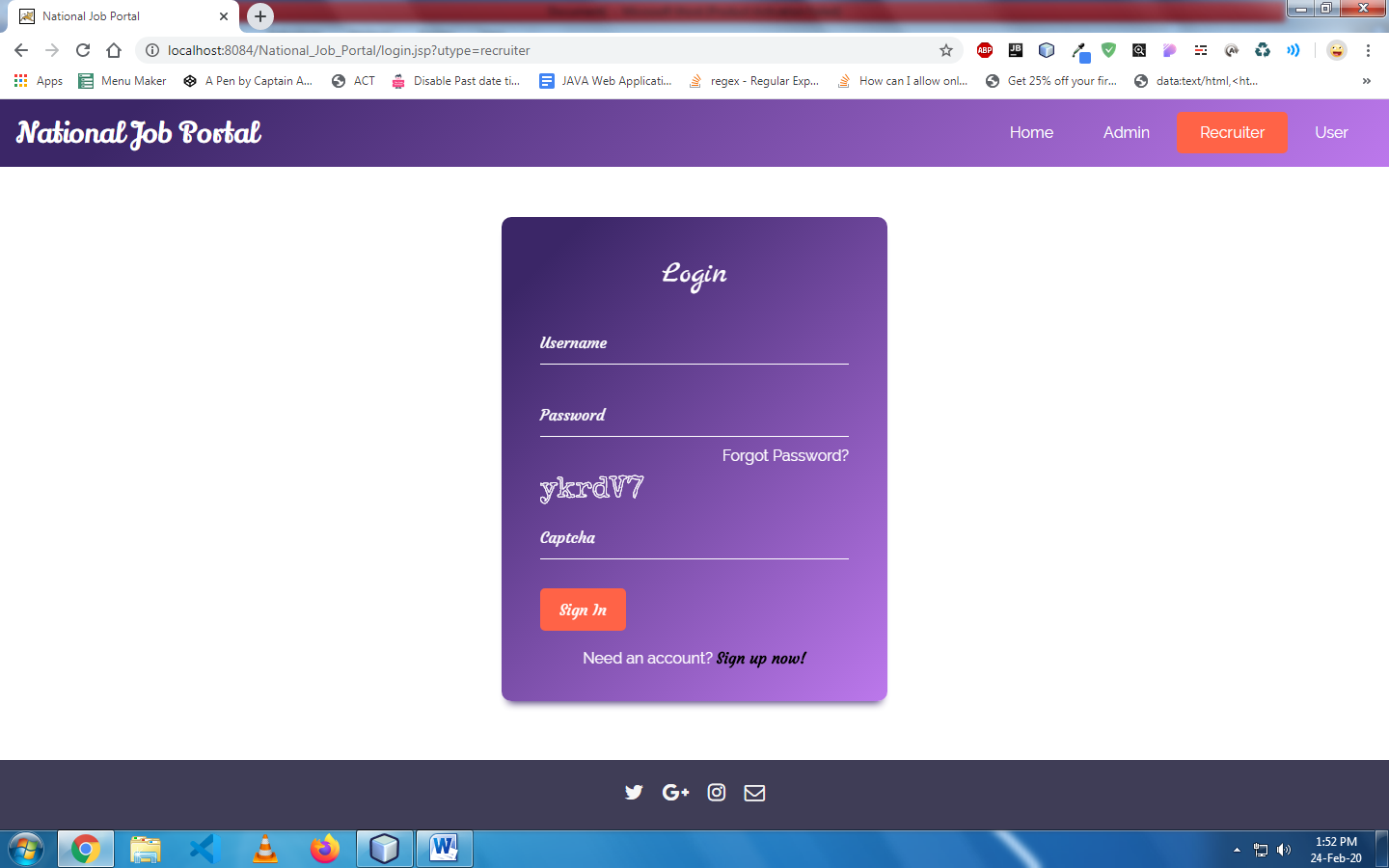


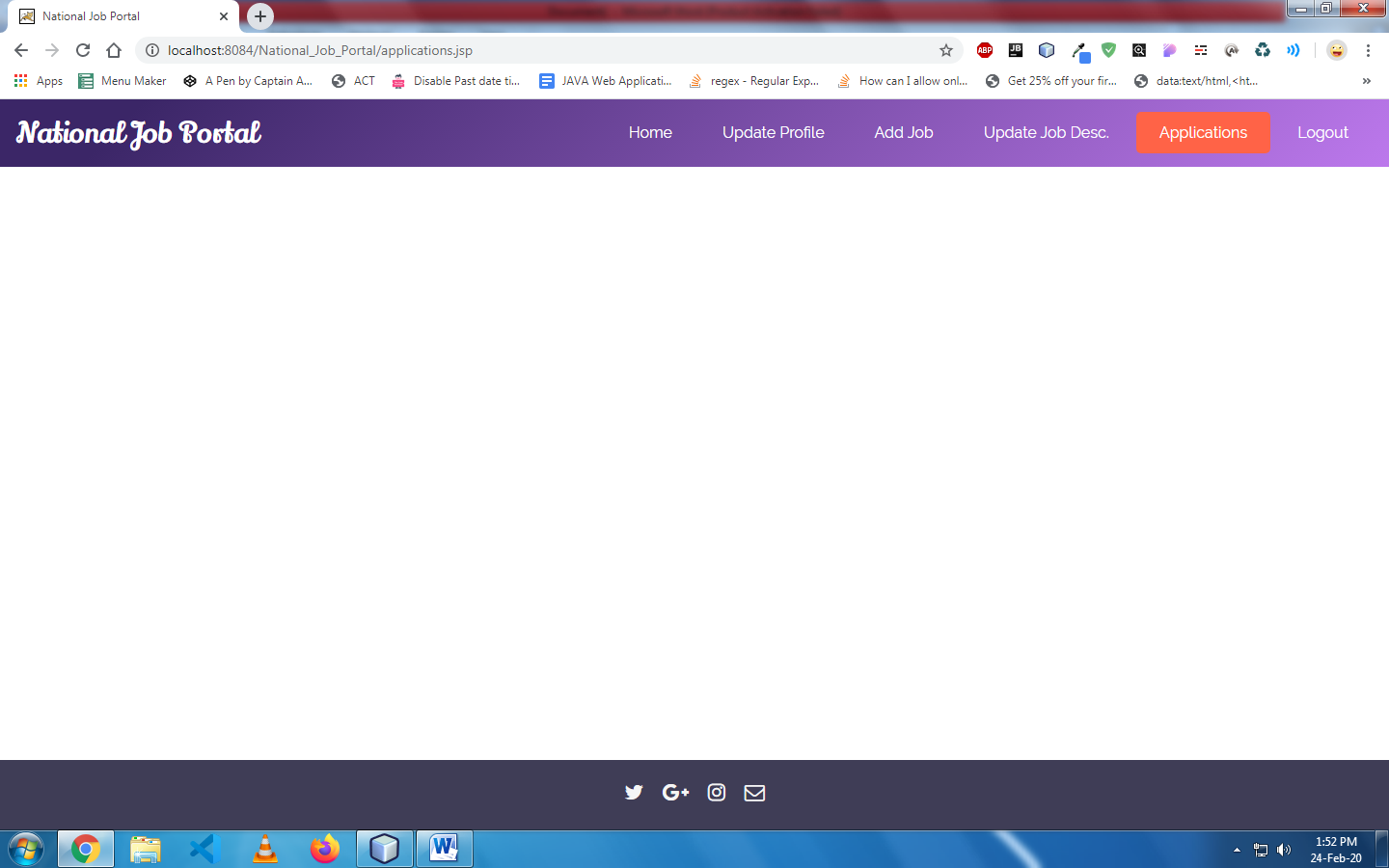












**8. TESTING**

## Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

## 8.1 SYSTEM TESTING

## Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

## 8.2 MODULE TESTING

## To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

## 8.3 INTEGRATION TESTING

## After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

## 8.4 ACCEPTANCE TESTING

## When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**8.5 Test Cases**

1. **User**

* Registration Test Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Input | Test case  Description | Expected  Output | Actual  Output | Status |
| 1 | Invalid user id and  password | User registration | User selects already existing user name | Displays message to choose different username | pass |
|  | Valid  User id and  password | User registration | User Enters valid details | user registered successfully | pass |

Table 8.5.1: **Test case for user registration**

Table 8.5.1 shows that, user has to register in order to login to the account if the details entered already exists, then it displays message to choose different username. If the user enters the valid details then registration will be successful.

* Login Test case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Input | Test case  Description | Expected  Output | Actual  Output | Status |
| 2 | Invalid user id and  password | Account  login | User enters wrong user id and password | Displays the message user id or password is incorrect | pass |
|  | Valid User id and password | Account  login | User enters correct user id and password | User logs in successfully | pass |

Table 8.5.2: **Test case for user login**

**9. CONCLUSION**

This system has successfully maintained the information of students. This system made uploading and calculating students marks easy. Students and teachers can save lots of time by using this system. All the drawbacks in the previous system has been overtaken by this system. This system creates easy and user friendly environment the users. It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in Java (servlets, jsp) & MySQL, but also about all handling procedure related with “National Job Portal”. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently. Tools used to develop this project are NetBeans 8.1, JDK 1.7, MySQL, SQLYog, HTML, JavaScript and CSS.

**Future Enhancements:**

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

* As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
* Based on the future security issues, security can be improved using emerging technologies like single sign-on.

**10. REFERENCES**

<https://www.ncs.gov.in/>

<https://www.slideshare.net/masterpiyush/documentation-18684935>

<http://www.economicsdiscussion.net/recruitment/sources-of-recruitment-in-hrm/31758>

<https://www.shrm.org/hr-today/trends-and-forecasting/special-reports-and-expert-views/documents/talent-acquisition-recruitment.pdf>

<https://careers.un.org/lbw/attachments/ManualfortheHiringManager.pdf>

<https://docs.oracle.com/en/cloud/saas/talent-management/18c/faimh/implementing-hiring.pdf>

**BIBLIOGRAPHY**

JAVA Technologies

JAVA Complete Reference

Java Script Programming by Yehuda Shiran

Mastering JAVA Security

J2EE Professional by Shadab siddiqui

JAVA server pages by Larne Pekowsley

JAVA Server pages by Nick Todd