import java.awt.Color;

import java.awt.Font;

import java.awt.GradientPaint;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.RenderingHints;

import java.awt.event.\*;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import javax.swing.\*;

import java.io.\*;

import java.text.DecimalFormat;

import java.util.Arrays;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.border.BevelBorder;

import javax.swing.border.Border;

import javax.swing.filechooser.FileSystemView;

public class CVA implements ActionListener {

String alllocation = "";

int nfiles = 0;

String shortlist = "";

//log in

JPanel login;

JTextArea a;

JPanel p;

JFrame f;

JLayeredPane lp;

//head

JPanel head;

JButton add, viewone, analyzeone, analyzeall;

//head

//navigation

JPanel navigation;

String cvlist = "";

JTextArea navtext;

//navigation

//body

JPanel view;

JPanel filenameone, dataone;

JLabel enterfilename;

JButton viewdateone;

JTextField getfile;

JTextArea onefiledata;

//analyze

JLabel l1, filenameone1, personname, grade;

JTextField getfile1;

JButton gradeview;

JPanel graderesultone1, graderesultone2;

JTextArea grade1dtata;

JProgressBar jb;

int i = 0;

//analyze

//analyze all

JPanel all, alt;

JLabel alname, algrade;

JTextArea altext;

//analyze all

JScrollPane pb;

//body

JTextField userid;

JPasswordField password;

JLabel loginl;

JButton loginb;

// shorting

String st = "", ad = "", sa = "";

int sectorno = 0;

JButton filtering;

JLabel filterl;

JPanel filter1, filter2;

JMenu filtert;

JMenuItem st1, ad1, sa1;

JMenuBar category;

JTextArea filterd;

public CVA() throws FileNotFoundException, IOException {

userid = new JTextField("USER ID :");

userid.setForeground(Color.WHITE);

userid.setBackground(Color.BLACK);

userid.setBounds(400, 400, 200, 30);

userid.addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

if (userid.getText().equals("USER ID :")) {

userid.setText("");

userid.setForeground(Color.WHITE);

}

}

@Override

public void focusLost(FocusEvent e) {

if (userid.getText().isEmpty()) {

userid.setForeground(Color.WHITE);

userid.setText("USER ID :");

}

}

});

password = new JPasswordField("PASSWORD");

password.setForeground(Color.WHITE);

password.setBackground(Color.BLACK);

password.setBounds(400, 450, 200, 30);

password.addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

if (password.getText().equals("PASSWORD")) {

password.setText("");

password.setForeground(Color.WHITE);

}

}

@Override

public void focusLost(FocusEvent e) {

if (password.getText().isEmpty()) {

password.setForeground(Color.WHITE);

password.setText("PASSWORD");

}

}

});

loginb = new JButton("LOG IN");

loginb.setBounds(450, 500, 100, 30);

loginb.setForeground(Color.WHITE);

loginb.setBackground(Color.BLACK);

ImageIcon logini = new ImageIcon("C:\\Major Project/wallpaper.jpg");

loginl = new JLabel();

loginl.setIcon(logini);

loginl .setBackground(Color.decode("#8ae6a2"));

loginl.setBounds(0, 0, 1200, 700);

loginl.add(password);

loginl.add(userid);

loginl.add(userid);

loginl.add(password);

loginl.add(loginb);

login = new JPanel();

login.setBounds(0, 0, 1200, 700);

login.setLayout(null);

login.add(loginl);

//============ head=================

Border mb1b = BorderFactory.createBevelBorder(BevelBorder.RAISED, Color.LIGHT\_GRAY, Color.GREEN);

add = new JButton("ADD");

add.setFont(new Font("sarif", Font.BOLD, 20));

add.setBackground(Color.decode("#8ae6a2"));

add.setForeground(Color.yellow);

add.setBounds(0, 0, 100, 40);

add.setBorder(null);

add.addMouseListener(new MouseAdapter() {

@Override

public void mouseEntered(MouseEvent me) {

add.setBorder(mb1b);

add.setBackground(Color.CYAN);

add.setForeground(Color.BLUE);

}

@Override

public void mouseExited(MouseEvent me) {

add.setBorder(BorderFactory.createEmptyBorder());

add.setBackground(Color.decode("#8ae6a2"));

add.setForeground(Color.yellow);

}

});

viewone = new JButton("VIEW");

viewone.setFont(new Font("sarif", Font.BOLD, 20));

viewone.setBackground(Color.decode("#8ae6a2"));

viewone.setForeground(Color.yellow);

viewone.setBounds(110, 0, 200, 40);

viewone.setBorder(null);

viewone.addMouseListener(new MouseAdapter() {

@Override

public void mouseEntered(MouseEvent me) {

viewone.setBorder(mb1b);

viewone.setBackground(Color.CYAN);

viewone.setForeground(Color.BLUE);

}

@Override

public void mouseExited(MouseEvent me) {

viewone.setBorder(BorderFactory.createEmptyBorder());

viewone.setBackground(Color.decode("#8ae6a2"));

viewone.setForeground(Color.yellow);

}

});

analyzeone = new JButton("ANALYZE\_ONE");

analyzeone.setFont(new Font("sarif", Font.BOLD, 20));

analyzeone.setBackground(Color.decode("#8ae6a2"));

analyzeone.setForeground(Color.yellow);

analyzeone.setBounds(320, 0, 200, 40);

analyzeone.setBorder(null);

analyzeone.addMouseListener(new MouseAdapter() {

@Override

public void mouseEntered(MouseEvent me) {

analyzeone.setBorder(mb1b);

analyzeone.setBackground(Color.CYAN);

analyzeone.setForeground(Color.BLUE);

}

@Override

public void mouseExited(MouseEvent me) {

analyzeone.setBorder(BorderFactory.createEmptyBorder());

analyzeone.setBackground(Color.decode("#8ae6a2"));

analyzeone.setForeground(Color.yellow);

}

});

analyzeall = new JButton("ANALYZE\_ALL");

analyzeall.setFont(new Font("sarif", Font.BOLD, 20));

analyzeall.setBackground(Color.decode("#8ae6a2"));

analyzeall.setForeground(Color.yellow);

analyzeall.setBounds(530, 0, 200, 40);

analyzeall.setBorder(null);

analyzeall.addMouseListener(new MouseAdapter() {

@Override

public void mouseEntered(MouseEvent me) {

analyzeall.setBorder(mb1b);

analyzeall.setBackground(Color.CYAN);

analyzeall.setForeground(Color.BLUE);

}

@Override

public void mouseExited(MouseEvent me) {

analyzeall.setBorder(BorderFactory.createEmptyBorder());

analyzeall.setBackground(Color.decode("#8ae6a2"));

analyzeall.setForeground(Color.yellow);

}

});

filtering = new JButton("CATEGORIZE");

filtering.setFont(new Font("sarif", Font.BOLD, 20));

filtering.setBackground(Color.decode("#8ae6a2"));

filtering.setForeground(Color.yellow);

filtering.setBounds(740, 0, 200, 40);

filtering.setBorder(null);

filtering.addMouseListener(new MouseAdapter() {

@Override

public void mouseEntered(MouseEvent me) {

filtering.setBorder(mb1b);

filtering.setBackground(Color.CYAN);

filtering.setForeground(Color.BLUE);

}

@Override

public void mouseExited(MouseEvent me) {

filtering.setBorder(BorderFactory.createEmptyBorder());

filtering.setBackground(Color.decode("#8ae6a2"));

filtering.setForeground(Color.yellow);

}

});

head = new JPanel();

head.setBounds(0, 0, 1200, 40);

head.setBackground(Color.decode("#8ae6a2"));

head.setLayout(null);

head.add(add);

head.add(viewone);

head.add(analyzeone);

head.add(analyzeall);

head.add(filtering);

//============ head=================

//============ navigation=================

navtext = new JTextArea();

navtext.setBounds(0, 0, 200, 660);

navtext.setBackground(Color.decode("#663300"));

navtext.setForeground(Color.white);

navtext.setFont(new Font("sarif", Font.BOLD, 20));

navtext.setBorder(null);

navtext.setText("\nCV LIST\n");

navtext.setVisible(true);

navtext.setEditable(false);

navigation = new JPanel();

navigation.setBounds(0, 40, 200, 660);

navigation.setBackground(Color.BLACK);

navigation.setLayout(null);

navigation.add(navtext);

//============ navigation=================

//============ body=================

/\*view = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

g2d.setRenderingHint(RenderingHints.KEY\_RENDERING, RenderingHints.VALUE\_RENDER\_QUALITY);

int w = getWidth();

int h = getHeight();

Color color1 = Color.decode("#cc2b5e");

Color color2 = Color.decode("#753a88");

GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);

g2d.setPaint(gp);

g2d.fillRect(0, 0, w, h);

}

};\*/

view = new JPanel();

view.setBounds(200, 40, 1000, 660);

view.setBackground(Color.CYAN);

view.setLayout(null);

//view

enterfilename = new JLabel("ENETR FILE NAME : ");

enterfilename.setBounds(10, 10, 200, 40);

enterfilename.setFont(new Font("sarif", Font.BOLD, 20));

enterfilename.setForeground(Color.PINK);

getfile = new JTextField(30);

getfile.setBounds(220, 10, 300, 40);

getfile.setFont(new Font("sarif", Font.BOLD, 20));

viewdateone = new JButton("VIEW DATA");

viewdateone.setBounds(530, 10, 200, 40);

viewdateone.setFont(new Font("sarif", Font.BOLD, 20));

viewdateone.setForeground(Color.yellow);

viewdateone.setBackground(Color.decode("#76871e"));

viewdateone.setBorder(null);

filenameone = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

g2d.setRenderingHint(RenderingHints.KEY\_RENDERING, RenderingHints.VALUE\_RENDER\_QUALITY);

int w = getWidth();

int h = getHeight();

Color color1 = Color.decode("#cc205e");

Color color2 = Color.decode("#754a88");

GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);

g2d.setPaint(gp);

g2d.fillRect(0, 0, w, h);

}

};

filenameone.setBounds(200, 40, 1000, 60);

filenameone.setBackground(Color.BLUE);

filenameone.setLayout(null);

filenameone.add(getfile);

filenameone.add(viewdateone);

filenameone.add(enterfilename);

onefiledata = new JTextArea();

onefiledata.setLineWrap(true);

onefiledata.setWrapStyleWord(true);

onefiledata.setBackground(Color.decode("#0c9686"));

onefiledata.setForeground(Color.BLACK);

onefiledata.setFont(new Font("sarif", Font.BOLD, 25));

onefiledata.setBorder(null);

pb = new JScrollPane(onefiledata);

pb.setBounds(00, 00, 1000, 600);

//view

//analyze one

personname = new JLabel("");

personname.setBounds(10, 10, 500, 40);

personname.setFont(new Font("sarif", Font.BOLD, 40));

personname.setForeground(Color.BLACK);

jb = new JProgressBar(0, 2000);

jb.setForeground(Color.red);

jb.setBounds(510, 10, 200, 40);

jb.setValue(0);

jb.setStringPainted(true);

jb.setVisible(false);

grade = new JLabel("");

grade.setBounds(810, 10, 200, 60);

grade.setFont(new Font("sarif", Font.BOLD, 40));

grade.setForeground(Color.PINK);

l1 = new JLabel("DATA ANALYSIS LIKE BELOW");

l1.setBounds(20, 70, 600, 30);

l1.setFont(new Font("sarif", Font.BOLD, 30));

l1.setForeground(Color.magenta);

filenameone1 = new JLabel("ENETR FILE NAME : ");

filenameone1.setBounds(10, 10, 200, 40);

filenameone1.setFont(new Font("sarif", Font.BOLD, 20));

filenameone1.setForeground(Color.PINK);

getfile1 = new JTextField(30);

getfile1.setBounds(220, 10, 300, 40);

getfile1.setFont(new Font("sarif", Font.BOLD, 20));

gradeview = new JButton("ANALYZE");

gradeview.setBounds(540, 10, 200, 40);

gradeview.setFont(new Font("sarif", Font.BOLD, 20));

gradeview.setForeground(Color.yellow);

gradeview.setBackground(Color.decode("#76871e"));

gradeview.setBorder(null);

graderesultone1 = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

g2d.setRenderingHint(RenderingHints.KEY\_RENDERING, RenderingHints.VALUE\_RENDER\_QUALITY);

int w = getWidth();

int h = getHeight();

Color color1 = Color.decode("#cc4b5e");

Color color2 = Color.decode("#752a88");

GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);

g2d.setPaint(gp);

g2d.fillRect(0, 0, w, h);

}

};

graderesultone1.setBounds(200, 40, 1000, 60);

graderesultone1.setBackground(Color.BLUE);

graderesultone1.setLayout(null);

graderesultone1.add(getfile1);

graderesultone1.add(gradeview);

graderesultone1.add(filenameone1);

grade1dtata = new JTextArea();

grade1dtata.setBounds(0, 100, 1000, 500);

grade1dtata.setBackground(Color.decode("#d7de64"));

grade1dtata.setForeground(Color.decode("#d4af37"));

grade1dtata.setFont(new Font("sarif", Font.BOLD, 25));

grade1dtata.setBorder(null);

graderesultone2 = new JPanel();

graderesultone2.setBounds(200, 100, 1000, 600);

graderesultone2.setBackground(Color.decode("#d7de64"));

graderesultone2.setForeground(Color.decode("#d4af37"));

graderesultone2.setLayout(null);

graderesultone2.add(personname);

graderesultone2.add(grade);

graderesultone2.add(grade1dtata);

graderesultone2.add(jb);

//analyze one

//analyze all

alname = new JLabel("NAME OF THE APPLICANT");

alname.setBounds(10, 10, 600, 40);

alname.setFont(new Font("sarif", Font.BOLD, 40));

alname.setForeground(Color.PINK);

algrade = new JLabel("GRADE");

algrade.setBounds(710, 10, 200, 40);

algrade.setFont(new Font("sarif", Font.BOLD, 40));

algrade.setForeground(Color.PINK);

all = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

g2d.setRenderingHint(RenderingHints.KEY\_RENDERING, RenderingHints.VALUE\_RENDER\_QUALITY);

int w = getWidth();

int h = getHeight();

Color color1 = Color.decode("#cc4b5e");

Color color2 = Color.decode("#752a88");

GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);

g2d.setPaint(gp);

g2d.fillRect(0, 0, w, h);

}

};

all.setBounds(200, 40, 1000, 60);

all.setBackground(Color.BLUE);

all.setLayout(null);

all.add(alname);

all.add(algrade);

altext = new JTextArea();

altext.setBounds(0, 100, 1000, 500);

altext.setBackground(Color.decode("#1b1e23"));

altext.setForeground(Color.decode("#d4af37"));

altext.setFont(new Font("sarif", Font.BOLD, 25));

altext.setBorder(null);

alt = new JPanel();

alt.setBounds(200, 00, 1000, 700);

alt.setBackground(Color.decode("#1b1e23"));

alt.setForeground(Color.decode("#d4af37"));

alt.setLayout(null);

alt.add(altext);

//

// filtering

filterl = new JLabel("SELECT THE CATEGORY ");

filterl.setBounds(10, 10, 600, 40);

filterl.setFont(new Font("sarif", Font.BOLD, 40));

filterl.setForeground(Color.PINK);

st1 = new JMenuItem("Software Tester");

st1.setBackground(Color.yellow);

ad1 = new JMenuItem("Application Developer");

ad1.setBackground(Color.yellow);

sa1 = new JMenuItem("Software Analyst ");

sa1.setBackground(Color.yellow);

filtert = new JMenu(" select ");

filtert.setForeground(Color.WHITE);

filtert.setBackground(Color.BLACK);

filtert.setBorder(BorderFactory.createLineBorder(Color.CYAN, 5));

filtert.add(st1);

filtert.add(ad1);

filtert.add(sa1);

category = new JMenuBar();

category.setForeground(Color.WHITE);

category.setBackground(Color.BLACK);

category.add(filtert);

category.setBounds(650, 5, 150, 40);

filter1 = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

g2d.setRenderingHint(RenderingHints.KEY\_RENDERING, RenderingHints.VALUE\_RENDER\_QUALITY);

int w = getWidth();

int h = getHeight();

Color color1 = Color.decode("#cc4b5e");

Color color2 = Color.decode("#752a88");

GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);

g2d.setPaint(gp);

g2d.fillRect(0, 0, w, h);

}

};

filter1.setBounds(200, 40, 1000, 60);

filter1.setBackground(Color.BLUE);

filter1.setLayout(null);

filter1.add(filterl);

filter1.add(category);

filterd = new JTextArea();

filterd.setBounds(0, 100, 1000, 500);

filterd.setBackground(Color.decode("#1b1e23"));

filterd.setForeground(Color.decode("#d4af37"));

filterd.setFont(new Font("sarif", Font.BOLD, 25));

filterd.setBorder(null);

filter2 = new JPanel();

filter2.setBounds(200, 00, 1000, 700);

filter2.setBackground(Color.decode("#1b1e23"));

filter2.setForeground(Color.decode("#d4af37"));

filter2.setLayout(null);

filter2.add(filterd);

//

dataone = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

g2d.setRenderingHint(RenderingHints.KEY\_RENDERING, RenderingHints.VALUE\_RENDER\_QUALITY);

int w = getWidth();

int h = getHeight();

Color color1 = Color.decode("#cb2b5e");

Color color2 = Color.decode("#783a88");

GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);

g2d.setPaint(gp);

g2d.fillRect(0, 0, w, h);

}

};;

dataone.setBounds(200, 100, 1000, 600);

dataone.setLayout(null);

dataone.add(pb);

//============ body=================

//============ all layers=================

lp = new JLayeredPane();

lp.setBounds(0, 0, 1200, 700);

lp.setLayout(null);

lp.setBackground(Color.GRAY);

lp.add(head);

lp.add(navigation);

lp.add(view);

//============ all layers=================

f = new JFrame("CV SORT LISTING APPLICATION");

f.setSize(1200, 700);

f.setLayout(null);

f.setVisible(true);

f.add(login);

//head

add.addActionListener(this);

viewone.addActionListener(this);

analyzeone.addActionListener(this);

analyzeall.addActionListener(this);

//head

//body

viewdateone.addActionListener(this);

//body

//analyze one

gradeview.addActionListener(this);

//

//login

loginb.addActionListener(this);

filtering.addActionListener(this);

st1.addActionListener(this);

ad1.addActionListener(this);

sa1.addActionListener(this);

}

public static void main(String[] args) throws IOException {

CVA c = new CVA();

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == loginb) {

String userid1;

char[] correctpassword = {'M', 'a', 'j', 'o', 'r'};

userid1 = userid.getText();

char[] psw = password.getPassword();

if (userid1.equals("Batch14") && Arrays.equals(correctpassword, psw)) {

f.remove(login);

f.add(lp);

f.repaint();

} else {

userid.setText(" ");

password.setText(" ");

}

}

if (e.getSource() == add) {

lp.removeAll();

shortlist = "";

filterd.setText("");

lp.add(head);

lp.add(navigation);

lp.add(view);

altext.setText("");

personname.setText("");

grade.setText("");

jb.setVisible(false);

jb.setValue(0);

i = 0;

getfile.setText("");

onefiledata.setText("");

getfile1.setText("");

grade1dtata.setText("");

JFileChooser jfc = new JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory());

int returnValue = jfc.showOpenDialog(null);

// int returnValue = jfc.showSaveDialog(null);

if (returnValue == JFileChooser.APPROVE\_OPTION) {

File selectedFile = jfc.getSelectedFile();

cvlist += selectedFile;

String s1 = navtext.getText();

navtext.setText(s1 + "\n" + selectedFile.getName());

alllocation += "==>" + selectedFile.getName() + "==>" + selectedFile.getAbsolutePath();

nfiles++;

}

}

if (e.getSource() == viewone) {

shortlist = "";

filterd.setText("");

getfile.setText("");

altext.setText("");

shortlist = "";

onefiledata.setText("");

personname.setText("");

grade.setText("");

jb.setVisible(false);

jb.setValue(0);

i = 0;

getfile1.setText("");

grade1dtata.setText("");

lp.removeAll();

lp.add(head);

lp.add(navigation);

lp.add(filenameone);

lp.add(dataone);

}

if (e.getSource() == viewdateone) {

shortlist = "";

filterd.setText("");

String separatepath[] = alllocation.split("==>");

altext.setText("");

shortlist = "";

String name = "";

String line = "";

for (int i = 1; i < separatepath.length; i = i + 2) {

String ss = getfile.getText();

if (ss.equals(separatepath[i])) {

try {

File file = new File(separatepath[i + 1]);

BufferedReader bf = null;

try {

bf = new BufferedReader(new FileReader(file));

} catch (FileNotFoundException ex) {

Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex);

}

line = bf.readLine();

while (line != null) {

name += line + "\n";

line = bf.readLine();

}

} catch (IOException ex) {

Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

onefiledata.setText(name);

}

if (e.getSource() == analyzeone) {

shortlist = "";

filterd.setText("");

altext.setText("");

getfile.setText("");

onefiledata.setText("");

personname.setText("");

grade.setText("");

jb.setVisible(false);

jb.setValue(0);

i = 0;

getfile1.setText("");

grade1dtata.setText("");

lp.removeAll();

lp.add(head);

lp.add(navigation);

lp.add(graderesultone1);

lp.add(graderesultone2);

}

if (e.getSource() == analyzeall) {

shortlist = "";

filterd.setText("");

ad = "";

sa = "";

st = "";

altext.setText(".");

getfile.setText("");

onefiledata.setText("");

personname.setText("");

grade.setText("");

f.repaint();

String separatepath[] = alllocation.split("==>");

// for(int j=1;j<=nfiles;j++){

f.repaint();

String separatepath1[] = alllocation.split("==>");

String resultarray[] = new String[separatepath1.length];

int c = 0;

for (int i = 1; i < separatepath1.length; i = i + 2) {

String name1 = "";

String line = "";

double totalpercentage = 0;

try {

File file = new File(separatepath1[i + 1]);

BufferedReader bf = null;

try {

bf = new BufferedReader(new FileReader(file));

} catch (FileNotFoundException ex) {

Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex);

}

line = bf.readLine();

while (line != null) {

name1 += line + "\n";

line = bf.readLine();

}

} catch (IOException ex) {

Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex);

}

String split[] = name1.split("==>");

String name = split[1];

String skills = split[3];

String sk[] = skills.split("=>");

double skillsno = sk.length - 1;

double k = skillsno / 8;

double skpercent = k \* 100;

String ski = "skills percentage =" + skpercent;

String experiance = split[4];

String exp[] = experiance.split("\n");

String experi = exp[2];

String qualifin = "";

if (experi.contains("Application Developer")) {

qualifin = "Application Developer";

sectorno = 1;

} else if (experi.contains("Software Analyst")) {

qualifin = "Software Analyst";

sectorno = 2;

} else if (experi.contains("Software Tester")) {

qualifin = "Software Tester";

sectorno = 3;

}

String exnumber = "";

for (char ch : experi.toCharArray()) {

if (Character.isDigit(ch)) {

exnumber = Character.toString(ch);

}

}

double ex = Double.parseDouble(exnumber);

double expercent = ex / 4 \* 100;

String exi = "experience percentage =" + expercent;

String degree = split[5];

String ed[] = degree.split("\n");

String de = ed[3];

String degp = "";

int coun = 1;

for (char ch : degree.toCharArray()) {

if (Character.isDigit(ch)) {

if (coun <= 2) {

degp += Character.toString(ch);

coun++;

}

}

}

double deg = Double.parseDouble(degp);

double dep1 = deg;

String dep = "degree percentage =" + dep1;

String inter = ed[4];

String interpercentage = "";

coun = 1;

for (char ch : inter.toCharArray()) {

if (Character.isDigit(ch)) {

if (coun <= 2) {

interpercentage += Character.toString(ch);

coun++;

}

}

}

double intep = Double.parseDouble(interpercentage);

double inte1 = intep;

String inte = "inter percentage =" + inte1;

coun = 1;

String school = ed[5];

String schoolpercentage = "";

for (char ch : school.toCharArray()) {

if (Character.isDigit(ch)) {

if (coun <= 2) {

schoolpercentage += Character.toString(ch);

coun++;

}

}

}

double schp = Double.parseDouble(schoolpercentage);

String sch = "school percentage =" + schp;

String awards = split[6];

String aw[] = awards.split("=>");

double awardsc = aw.length - 1;

double perawards = awardsc / 6 \* 100;

DecimalFormat dff = new DecimalFormat("00.00");

String awa = "awards percentage =" + dff.format(perawards);

String languages = split[7];

String la[] = languages.split("=>");

double languagesc = la.length - 1;

double languagesper = languagesc / 4 \* 100;

String lan = "languages percentage =" + languagesper;

String projects = split[8];

String pr[] = projects.split("=>");

double projectsc = pr.length - 1;

double perperojects = projectsc / 3 \* 100;

DecimalFormat df2 = new DecimalFormat("##.##");

String pro = "projects percentage =" + perperojects;

totalpercentage = (skpercent + expercent + dep1 + intep + schp + perawards + languagesper + perperojects) / 800 \* 100;

String result = split[1].substring(0, split[1].indexOf("\n")) + "\t(" + qualifin + ")\t\t " + df2.format(totalpercentage) + "\n";

if (sectorno == 1) {

ad += result + "\n";

} else if (sectorno == 2) {

sa += result + "\n";

} else if (sectorno == 3) {

st += result + "\n";

}

shortlist += result;

resultarray[c] = result;

c++;

}

//}

altext.setText(shortlist);

lp.removeAll();

lp.add(head);

lp.add(navigation);

lp.add(all);

lp.add(alt);

f.repaint();

}

if (e.getSource() == filtering) {

filterd.setText("");

shortlist = "";

altext.setText(" ");

getfile.setText("");

onefiledata.setText("");

personname.setText("");

grade.setText("");

f.repaint();

lp.removeAll();

lp.add(head);

lp.add(navigation);

lp.add(filter1);

lp.add(filter2);

f.repaint();

}

//filtering

if (e.getSource() == st1) {

filterd.setText("");

filterd.setText(st);

}

if (e.getSource() == ad1) {

filterd.setText("");

filterd.setText(ad);

}

if (e.getSource() == sa1) {

filterd.setText("");

filterd.setText(sa);

}

// grageview getfile1 grade1dtata

if (e.getSource() == gradeview) {

filterd.setText("");

shortlist = "";

altext.setText(" ");

getfile.setText("");

onefiledata.setText("");

personname.setText("");

grade.setText("");

f.repaint();

jb.setVisible(false);

jb.setValue(0);

i = 0;

f.repaint();

String separatepath[] = alllocation.split("==>");

String name1 = "";

String line = "";

for (int i = 1; i < separatepath.length; i = i + 2) {

String ss = getfile1.getText();

if (ss.equals(separatepath[i])) {

try {

File file = new File(separatepath[i + 1]);

BufferedReader bf = null;

try {

bf = new BufferedReader(new FileReader(file));

} catch (FileNotFoundException ex) {

Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex);

}

line = bf.readLine();

while (line != null) {

name1 += line + "\n";

line = bf.readLine();

}

} catch (IOException ex) {

Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

String split[] = name1.split("==>");

String name = split[1];

String skills = split[3];

String sk[] = skills.split("=>");

double skillsno = sk.length - 1;

double k = skillsno / 8;

double skpercent = k \* 100;

String ski = "skills percentage =" + skpercent;

String experience = split[4];

String exp[] = experience.split("\n");

String experi = exp[2];

String exnumber = "";

for (char ch : experi.toCharArray()) {

if (Character.isDigit(ch)) {

exnumber = Character.toString(ch);

}

}

double ex = Double.parseDouble(exnumber);

double expercent = ex / 4 \* 100;

String exi = "experience percentage =" + expercent;

String degree = split[5];

String ed[] = degree.split("\n");

String de = ed[3];

String degp = "";

int coun = 1;

for (char ch : degree.toCharArray()) {

if (Character.isDigit(ch)) {

if (coun <= 2) {

degp += Character.toString(ch);

coun++;

}

}

}

double deg = Double.parseDouble(degp);

double dep1 = deg;

String dep = "degree percentage =" + dep1;

String inter = ed[4];

String interpercentage = "";

coun = 1;

for (char ch : inter.toCharArray()) {

if (Character.isDigit(ch)) {

if (coun <= 2) {

interpercentage += Character.toString(ch);

coun++;

}

}

}

double intep = Double.parseDouble(interpercentage);

String inte = "inter percentage =" + intep;

String school = ed[5];

String schoolpercentage = "";

coun = 1;

for (char ch : school.toCharArray()) {

if (Character.isDigit(ch)) {

if (coun <= 2) {

schoolpercentage += Character.toString(ch);

coun++;

}

}

}

double schp = Double.parseDouble(schoolpercentage);

String sch = "school percentage =" + schp;

String awards = split[6];

String aw[] = awards.split("=>");

double awardsc = aw.length - 1;

double perawards = awardsc / 6 \* 100;

DecimalFormat dff = new DecimalFormat("00.00");

String awa = "awards percentage =" + dff.format(perawards);

String languages = split[7];

String la[] = languages.split("=>");

double languagesc = la.length - 1;

double languagesper = languagesc / 4 \* 100;

String lan = "languages percentage =" + languagesper;

String projects = split[8];

String pr[] = projects.split("=>");

double projectsc = pr.length - 1;

double perperojects = projectsc / 3 \* 100;

DecimalFormat df2 = new DecimalFormat("##.##");

String pro = "projects percentage =" + perperojects;

double totalpercentage = (skpercent + expercent + dep1 + intep + schp + perawards + languagesper + perperojects) / 800 \* 100;

String total1 = ski + "\n" + exi + "\n" + dep + "\n" + inte + "\n" + sch + "\n" + awa + "\n" + lan + "\n" + pro + "\ntotal= " + totalpercentage;

personname.setText(name);

grade.setText(df2.format(totalpercentage) + "%");

grade1dtata.setText(total1);

f.repaint();

//score

jb.setVisible(true);

f.repaint();

int prog = (int) (totalpercentage \* 20);

//double prog=totalpercentage;

while (i <= prog) {

if (i > 500) {

jb.setForeground(Color.ORANGE);

}

if (i > 1000) {

jb.setForeground(Color.blue);

}

if (i > 1500) {

jb.setForeground(Color.green);

}

jb.setValue(i);

f.repaint();

i = i + 20;

try {

Thread.sleep(00);

} catch (InterruptedException ek) {

}

}

}

}

}