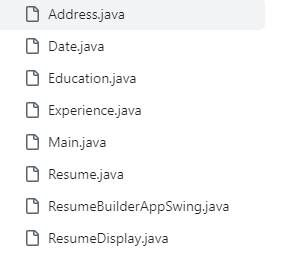
**InstaResume**

**Readme:**  
**Today’s Agenda:**

* Complete the input form for the project
* Create an output display element to display the data collected
* Data validation of input data streams

**Remaining Development Processes:**

* Develop user logins and separate profiles
* Store generated resume for each profile
* Generate PDF output for the resume



**Address:**

public class Address {  
 private String street;  
 private String city;  
 private String state;  
 private String zip;  
  
 public Address(String street, String city, String state, String zip) {  
 this.street = street;  
 this.city = city;  
 this.state = state;  
 this.zip = zip;  
 }  
  
 @Override  
 public String toString() {  
 return street + ", " + city + ", " + state + " " + zip;  
 }  
}

**Date:**

public class Date {  
 private String month;  
 private String year;  
  
 public Date(String month, String year) {  
 this.month = month;  
 this.year = year;  
 }  
  
 @Override  
 public String toString() {  
 return month + " " + year;  
 }  
}

**Education**public class Education {  
 private String degree;  
 private String institution;  
 private Date graduationDate;  
  
 public Education(String degree, String institution, Date graduationDate) {  
 this.degree = degree;  
 this.institution = institution;  
 this.graduationDate = graduationDate;  
 }  
  
 @Override  
 public String toString() {  
 return degree + " from " + institution + " (Graduated: " + graduationDate + ")";  
 }  
}

**Experience**public class Experience {  
 private String jobTitle;  
 private String company;  
 private Date duration;  
  
 public Experience(String jobTitle, String company, Date duration) {  
 this.jobTitle = jobTitle;  
 this.company = company;  
 this.duration = duration;  
 }  
  
 @Override  
 public String toString() {  
 return jobTitle + " at " + company + " (Duration: " + duration + ")";  
 }  
}

**Resume:**

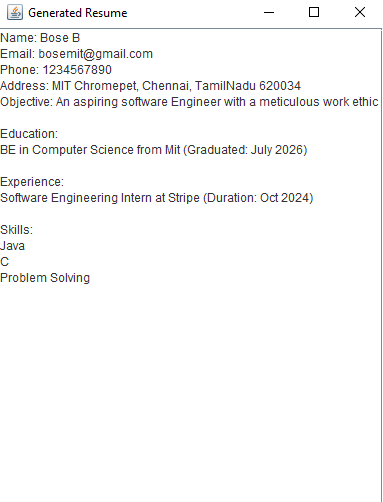
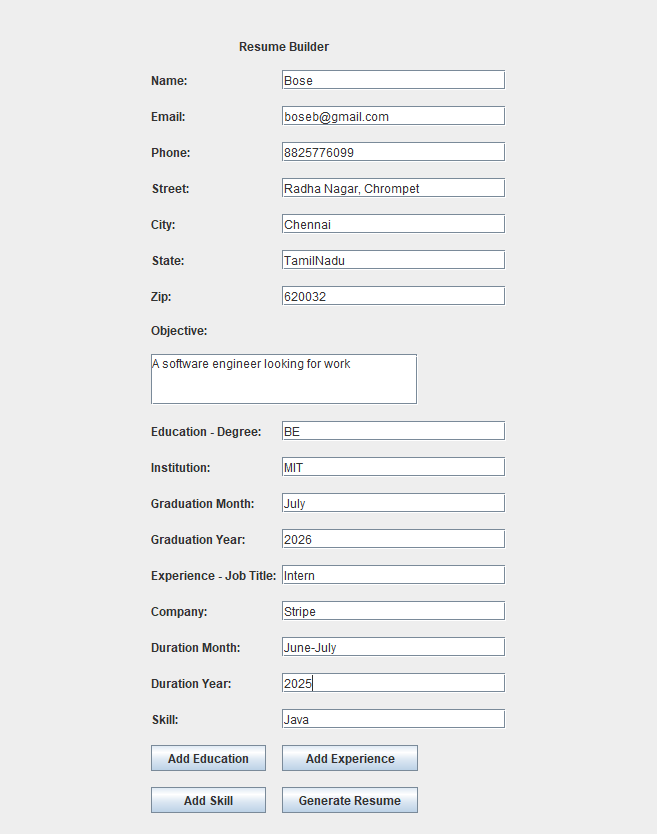
import java.util.ArrayList;  
import java.util.List;  
  
public class Resume {  
 private String name;  
 private String email;  
 private String phone;  
 private Address address;  
 private String objective;  
 private List<Education> educationList;  
 private List<Experience> experienceList;  
 private List<String> skills;  
  
 public Resume() {  
 educationList = new ArrayList<>();  
 experienceList = new ArrayList<>();  
 skills = new ArrayList<>();  
 }  
  
 *// Setters* public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public void setPhone(String phone) {  
 this.phone = phone;  
 }  
  
 public void setAddress(Address address) {  
 this.address = address;  
 }  
  
 public void setObjective(String objective) {  
 this.objective = objective;  
 }  
  
 public void addEducation(Education education) {  
 educationList.add(education);  
 }  
  
 public void addExperience(Experience experience) {  
 experienceList.add(experience);  
 }  
  
 public void addSkill(String skill) {  
 skills.add(skill);  
 }  
  
 *// Getters* public String getName() {  
 return name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public String getPhone() {  
 return phone;  
 }  
  
 public Address getAddress() {  
 return address;  
 }  
  
 public String getObjective() {  
 return objective;  
 }  
  
 public List<Education> getEducationList() {  
 return educationList;  
 }  
  
 public List<Experience> getExperienceList() {  
 return experienceList;  
 }  
  
 public List<String> getSkills() {  
 return skills;  
 }  
  
 @Override  
 public String toString() {  
 StringBuilder sb = new StringBuilder();  
 sb.append("Name: ").append(name).append("\n")  
 .append("Email: ").append(email).append("\n")  
 .append("Phone: ").append(phone).append("\n")  
 .append("Address: ").append(address).append("\n")  
 .append("Objective: ").append(objective).append("\n\n")  
 .append("Education:\n");  
  
 if (educationList.isEmpty()) {  
 sb.append("No education entries added.\n");  
 } else {  
 for (Education edu : educationList) {  
 sb.append(edu).append("\n");  
 }  
 }  
  
 sb.append("\nExperience:\n");  
 if (experienceList.isEmpty()) {  
 sb.append("No experience entries added.\n");  
 } else {  
 for (Experience exp : experienceList) {  
 sb.append(exp).append("\n");  
 }  
 }  
  
 sb.append("\nSkills:\n");  
 if (skills.isEmpty()) {  
 sb.append("No skills added.\n");  
 } else {  
 for (String skill : skills) {  
 sb.append(skill).append("\n");  
 }  
 }  
 return sb.toString();  
 }  
}

**ResumeBuilderAppSwing**

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class ResumeBuilderAppSwing extends JFrame {  
 private Resume resume;  
  
 public ResumeBuilderAppSwing() {  
 resume = new Resume();  
 setTitle("Resume Builder");  
 setSize(500, 700);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(new GridLayout(0, 1));  
  
 *// Personal Info* add(new JLabel("Name:"));  
 JTextField nameField = new JTextField();  
 add(nameField);  
  
 add(new JLabel("Email:"));  
 JTextField emailField = new JTextField();  
 add(emailField);  
  
 add(new JLabel("Phone:"));  
 JTextField phoneField = new JTextField();  
 add(phoneField);  
  
 add(new JLabel("Address (Street):"));  
 JTextField streetField = new JTextField();  
 add(streetField);  
  
 add(new JLabel("City:"));  
 JTextField cityField = new JTextField();  
 add(cityField);  
  
 add(new JLabel("State:"));  
 JTextField stateField = new JTextField();  
 add(stateField);  
  
 add(new JLabel("Zip:"));  
 JTextField zipField = new JTextField();  
 add(zipField);  
  
 add(new JLabel("Objective:"));  
 JTextArea objectiveArea = new JTextArea();  
 add(new JScrollPane(objectiveArea));  
  
 *// Education fields* add(new JLabel("Education - Degree:"));  
 JTextField degreeField = new JTextField();  
 add(degreeField);  
  
 add(new JLabel("Institution:"));  
 JTextField institutionField = new JTextField();  
 add(institutionField);  
  
 add(new JLabel("Graduation Month:"));  
 JTextField gradMonthField = new JTextField();  
 add(gradMonthField);  
  
 add(new JLabel("Graduation Year:"));  
 JTextField gradYearField = new JTextField();  
 add(gradYearField);  
  
 JButton addEducationButton = new JButton("Add Education");  
 add(addEducationButton);  
  
 *// Experience fields* add(new JLabel("Experience - Job Title:"));  
 JTextField jobTitleField = new JTextField();  
 add(jobTitleField);  
  
 add(new JLabel("Company:"));  
 JTextField companyField = new JTextField();  
 add(companyField);  
  
 add(new JLabel("Duration Month:"));  
 JTextField durationMonthField = new JTextField();  
 add(durationMonthField);  
  
 add(new JLabel("Duration Year:"));  
 JTextField durationYearField = new JTextField();  
 add(durationYearField);  
  
 JButton addExperienceButton = new JButton("Add Experience");  
 add(addExperienceButton);  
  
 *// Skills field* add(new JLabel("Skill:"));  
 JTextField skillField = new JTextField();  
 add(skillField);  
  
 JButton addSkillButton = new JButton("Add Skill");  
 add(addSkillButton);  
  
 *// Generate Resume Button* JButton generateButton = new JButton("Generate Resume");  
 add(generateButton);  
  
 *// Text Area for displaying the resume* JTextArea resumeDisplayArea = new JTextArea();  
 resumeDisplayArea.setEditable(false);  
 add(new JScrollPane(resumeDisplayArea));  
  
 *// Event Handlers* addEducationButton.addActionListener(e -> {  
 String degree = degreeField.getText();  
 String institution = institutionField.getText();  
 String month = gradMonthField.getText();  
 String year = gradYearField.getText();  
 if (!degree.isEmpty() && !institution.isEmpty() && !month.isEmpty() && !year.isEmpty()) {  
 Date graduationDate = new Date(month, year);  
 resume.addEducation(new Education(degree, institution, graduationDate));  
 degreeField.setText("");  
 institutionField.setText("");  
 gradMonthField.setText("");  
 gradYearField.setText("");  
 }  
 });  
  
 addExperienceButton.addActionListener(e -> {  
 String jobTitle = jobTitleField.getText();  
 String company = companyField.getText();  
 String month = durationMonthField.getText();  
 String year = durationYearField.getText();  
 if (!jobTitle.isEmpty() && !company.isEmpty() && !month.isEmpty() && !year.isEmpty()) {  
 Date duration = new Date(month, year);  
 resume.addExperience(new Experience(jobTitle, company, duration));  
 jobTitleField.setText("");  
 companyField.setText("");  
 durationMonthField.setText("");  
 durationYearField.setText("");  
 }  
 });  
  
 addSkillButton.addActionListener(e -> {  
 String skill = skillField.getText();  
 if (!skill.isEmpty()) {  
 resume.addSkill(skill);  
 skillField.setText("");  
 }  
 });  
  
 generateButton.addActionListener(e -> {  
 resume.setName(nameField.getText());  
 resume.setEmail(emailField.getText());  
 resume.setPhone(phoneField.getText());  
 Address address = new Address(streetField.getText(), cityField.getText(), stateField.getText(), zipField.getText());  
 resume.setAddress(address);  
 resume.setObjective(objectiveArea.getText());  
  
 *// Display the resume* JFrame displayFrame = new JFrame("Generated Resume");  
 displayFrame.setSize(400, 600);  
 JTextArea displayArea = new JTextArea(resume.toString());  
 displayArea.setEditable(false);  
 displayFrame.add(new JScrollPane(displayArea));  
 displayFrame.setVisible(true);  
 displayFrame.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 this.dispose(); *// Close the input frame* });  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(() -> {  
 ResumeBuilderAppSwing app = new ResumeBuilderAppSwing();  
 app.setVisible(true);  
 });  
 }  
}

**Resume Display**

import javax.swing.\*;  
import java.awt.\*;  
  
public class ResumeDisplay extends JFrame {  
 public ResumeDisplay(Resume resume) {  
 setTitle("Generated Resume");  
 setSize(500, 700);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(new BorderLayout());  
  
 JTextArea resumeTextArea = new JTextArea();  
 resumeTextArea.setEditable(false);  
 resumeTextArea.setText(resume.toString());  
  
 JScrollPane scrollPane = new JScrollPane(resumeTextArea);  
 add(scrollPane, BorderLayout.*CENTER*);  
  
 *// Optional: Add a close button* JButton closeButton = new JButton("Close");  
 closeButton.addActionListener(e -> System.*exit*(0));  
 add(closeButton, BorderLayout.*SOUTH*);  
 }  
}

****