ONLINE RESUME BUILDER USING JAVA

Submitted to the

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES by

K.JAYA SAI AKASH (Reg No:192111528)

Under the guidance of

**Dr.M. BUVANESVARI**

Professor



Institute of Computer Science and Engineering

SAVEETHA SCHOOL OF ENGINEERING CHENNAI – 602 105 TAMILNADU, INDIA MARCH 2024

**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled “ONLINE RESUME BUILDER” submitted by “K.JAYA SAI AKASH (192111528)” to Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, is a record of bonafide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this institution and in my appraisal meets the required standards for submission.

Dr.M. Buvanesvari

Professor

Department of

Biosciences,

Saveetha School of Engineering

SIMATS, Chennai – 602 105

Internal examiner External Examiner

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENTS** | **PAGE NO** |
| 1 | ABSTRACT | 4 |
| 2 | INTRODUCTION | 5 |
| 3 | DESCRIPTION | 6 |
| 4 | ADVANTAGES | 7 |
| 5 | SYSTEM REQUIREMENTS | 8 |
| 6 | EXISTING WORK | 9 |
| 7 | PROPOSED WORK | 10 |
| 8 | TECHNOLOGY USED | 11 |
| 9 | USE CASE DIAGRAM | 13 |
| 10 | SOURCE CODE | 14 |
| 11 | SCREENSHOTS(OUTPUTS) | 19 |
| 12 | FUTURE ENHANCING | 20 |
| 12 | CONCLUSION | 21 |
| 13 | REFERENCES | 22 |

**ABSTRACT**

The Online Resume Builder (ORB) is an innovative Java-based solution engineered to revolutionize the resume creation process. It aims to simplify and streamline the task of crafting professional resumes, leveraging Java's robust capabilities to provide users with a seamless experience. By integrating various resume-building functionalities into a cohesive digital platform, ORB is committed to enhancing efficiency and user satisfaction in resume creation.

ORB's primary focus is on improving user experience and productivity. Through intuitive user interfaces and streamlined workflows, the system enables users to effortlessly input their information, select from a range of templates, and customize their resumes to suit their needs. By automating repetitive tasks and offering real-time editing capabilities, ORB reduces manual effort, allowing users to focus on presenting their qualifications effectively.

Moreover, ORB prioritizes data accuracy and security. By leveraging Java's strong data management features, the system ensures that user information is securely stored and readily accessible. With a centralized repository for resume data, ORB facilitates easy retrieval and updating, empowering users to maintain accurate and up-to-date resumes. Overall, the Online Resume Builder strives to provide a user-centric environment that promotes productivity, creativity, and professionalism in resume creation.

**INTRODUCTION**

The Online Resume Builder (ORB) represents a paradigm shift in the realm of professional development, offering a comprehensive solution to the complexities of resume creation and management. Powered by Java's advanced technology, this innovative system streamlines the process of crafting resumes, embodying the essence of modern job-seeking practices. From simplifying data entry to providing dynamic template options, ORB caters to the diverse needs of job seekers and employers, ensuring a seamless and efficient experience for all involved parties.

ORB's user-friendly interface serves as a gateway to effortless resume creation, making it accessible to users of all skill levels. By leveraging intuitive design principles, the system empowers job seekers to craft polished resumes that effectively showcase their qualifications and experience. Moreover, ORB's integration with cutting-edge communication tools fosters collaboration between job seekers and hiring professionals, facilitating seamless interaction throughout the application process.

In today's competitive job market, the integration of automation into resume-building tasks is paramount. ORB reduces the manual burden of resume creation by automating repetitive processes, allowing job seekers to focus on refining their qualifications and tailoring their applications to specific roles. Additionally, the system's analytical capabilities offer valuable insights into resume trends, enabling users to make informed decisions and optimize their job search strategies for greater success.

One of ORB's key strengths lies in its ability to enhance communication between job seekers and employers. By providing a centralized platform for resume submission and review, the system streamlines the hiring process, reducing delays and improving overall efficiency. This seamless communication fosters a collaborative environment where both parties can engage effectively, leading to more meaningful interactions and better outcomes for all involved.

Ultimately, the Online Resume Builder represents a transformative tool that empowers job seekers to navigate the complexities of the modern job market with confidence and ease. Its user-centric design, advanced technology, and robust feature set combine to create a streamlined and efficient experience that aligns seamlessly with the demands of today's dynamic employment landscape. As job seekers embrace the future of professional development, ORB stands as a beacon of innovation, offering a glimpse into a world where resume creation is as effortless as it is effective.

**DESCRIPTION**

The Online Resume Builder (ORB) application is a Java-based solution meticulously crafted to emulate the functionalities of a digital system tailored for simplifying the process of resume creation and management. This program adheres to the principles of object-oriented programming, modeling user data and resume-building functionalities with precision and efficiency.

Key Features of the ORB Program:

1. User Profile Creation: Users can effortlessly create personalized profiles, inputting essential information such as contact details, skills, and work experience.

2. Template Selection: The system offers a diverse range of professionally designed resume templates, allowing users to choose layouts that best represent their professional identity.

3. Dynamic Editing: ORB provides intuitive editing tools that enable users to customize their resumes dynamically, adjusting formatting, content, and styling to suit individual preferences.

4. Real-time Preview: Users can instantly preview their resumes as they edit, ensuring a seamless editing experience and facilitating real-time adjustments for optimal presentation.

Resume Content Management:

The program effectively manages resume content, allowing users to input and organize information such as education history, work experience, skills, and achievements. With seamless data management capabilities, ORB ensures accuracy and coherence in resume content, empowering users to create compelling and professional resumes.

Overall, the Online Resume Builder emerges as a versatile and user-friendly tool, empowering individuals to craft polished resumes with ease and efficiency. By combining the power of Java programming with intuitive design principles, ORB sets a new standard for resume-building applications, offering a seamless and streamlined experience that aligns with the demands of today's competitive job market.

**ADVANTAGES :**

1. Convenience and Accessibility: Job seekers can conveniently create and manage resumes from anywhere, whether at home or on the go, utilizing online platforms or mobile applications, ensuring 24/7 accessibility to resume-building tools.

2. Time Savings: Automated resume-building systems significantly reduce the time required for crafting resumes, eliminating the need for physical visits to career centers or spending hours formatting documents manually.

3. Real-time Editing: Users can access real-time editing features, enabling instant modifications to resume content, formatting, and design, facilitating informed decision-making and efficient resume refinement.

4. Flexible Template Options: ORB offers a wide range of professionally designed resume templates, providing flexibility in choosing layouts and styles that best showcase individual qualifications and experiences.

5. Secure Data Management: The program ensures the security and confidentiality of user data through encrypted storage and secure access protocols, safeguarding sensitive personal and professional information during the resume-building process.

6. Efficient Presentation: Job seekers can efficiently organize and present their qualifications, skills, and experiences, optimizing the layout and content of their resumes to attract potential employers effectively.

7. Enhanced User Experience: ORB features a user-friendly interface and intuitive navigation, offering a seamless and enjoyable resume-building experience for users of all skill levels, contributing to a positive overall experience.

8. Cost Savings: Job seekers can take advantage of free or low-cost resume-building services, eliminating the need for expensive resume writing services and potentially saving money in their job search endeavors.

**SYSTEM REQUIREMENTS**

System Requirements for the Online Resume Builder (ORB) Program:

1. Operating System Compatibility:The ORB application is compatible with desktop computers running Java Runtime Environment (JRE) version 8 or higher.

2. Minimum Hardware Specifications:Users should have a computer with at least 2 GB of RAM and a processor with speeds above 1.8 GHz (any make).

3. Storage Requirements: The application requires a minimum of 100 MB of storage space for installation and additional space for storing user data. As the number of resumes and data entries increases, the storage requirements may dynamically adjust.

4. Java Development Kit (JDK): Users need to have JDK version 8 or higher installed on their systems to run the ORB application and develop custom features or plugins.

5. Internet Connection:A stable internet connection is required for accessing online features, such as template updates, job search functionalities, and cloud-based storage options.

6. Permissions:Users may need administrative privileges to install software on their computers and grant necessary permissions for the ORB application to function correctly.

7. Compatibility: The ORB application is compatible with various operating systems, including Windows, macOS, and Linux, providing flexibility for users across different platforms.

8. Security Measures:The ORB application employs industry-standard encryption protocols to ensure the security and confidentiality of user data during transmission and storage.

9. JavaFX Support:The ORB application utilizes JavaFX for its graphical user interface, requiring users to have JavaFX libraries installed or included in their Java environment to run the application smoothly.

**EXISTING WORK:**

Open Source Development Platforms: Platforms like GitHub or SourceForge host numerous open-source projects related to resume building and career development. By exploring repositories focusing on Java-based resume builders, such as ResumeBuilder or OpenResume, developers can gain insights into existing work and contribute to collaborative projects.

Industry-Specific Communities and Forums: Active participation in forums and platforms dedicated to software development or career advancement, such as Stack Overflow or specialized resume-building forums, offers opportunities to discuss existing work, share challenges, and exchange innovative ideas in the realm of Java-based online resume builders.

Research Publications and Conferences: Research journals, conferences, and academic publications in the field of software engineering and career development often feature studies, advancements, and case studies related to online resume builders developed in Java. By reviewing these resources, developers can stay informed about the latest trends and methodologies in building robust and user-friendly online resume platforms

**PROPOSED WORK**

Innovative User Interface Design: Design and develop an intuitive and visually appealing user interface for online resume builders, ensuring a seamless and engaging experience for users as they create and customize their resumes.

Advanced Customization Features:Implement advanced customization features that allow users to tailor their resumes to specific industries, job roles, and personal preferences, such as choosing different resume formats, styles, and content layouts.

Integration with AI and Natural Language Processing (NLP):Explore the integration of AI and NLP technologies to enhance resume analysis capabilities, provide personalized suggestions for content improvement, and optimize keyword usage for improved search engine visibility.

Dynamic Content Generation: Develop algorithms for dynamic content generation, enabling the automatic extraction of relevant information from user input, such as educational background, work experience, and skills, to populate resume sections accurately and efficiently.

Collaborative Editing and Feedback Mechanisms: Incorporate collaborative editing features and feedback mechanisms that allow users to share their resumes with mentors, peers, or professional reviewers for feedback and suggestions, fostering continuous improvement in resume quality and effectiveness

**TECHNOLOGIES USED**

Web Development Technologies:HTML, CSS, and JavaScript serve as foundational elements for creating dynamic and interactive user interfaces in online resume builders, enhancing user experience and usability.

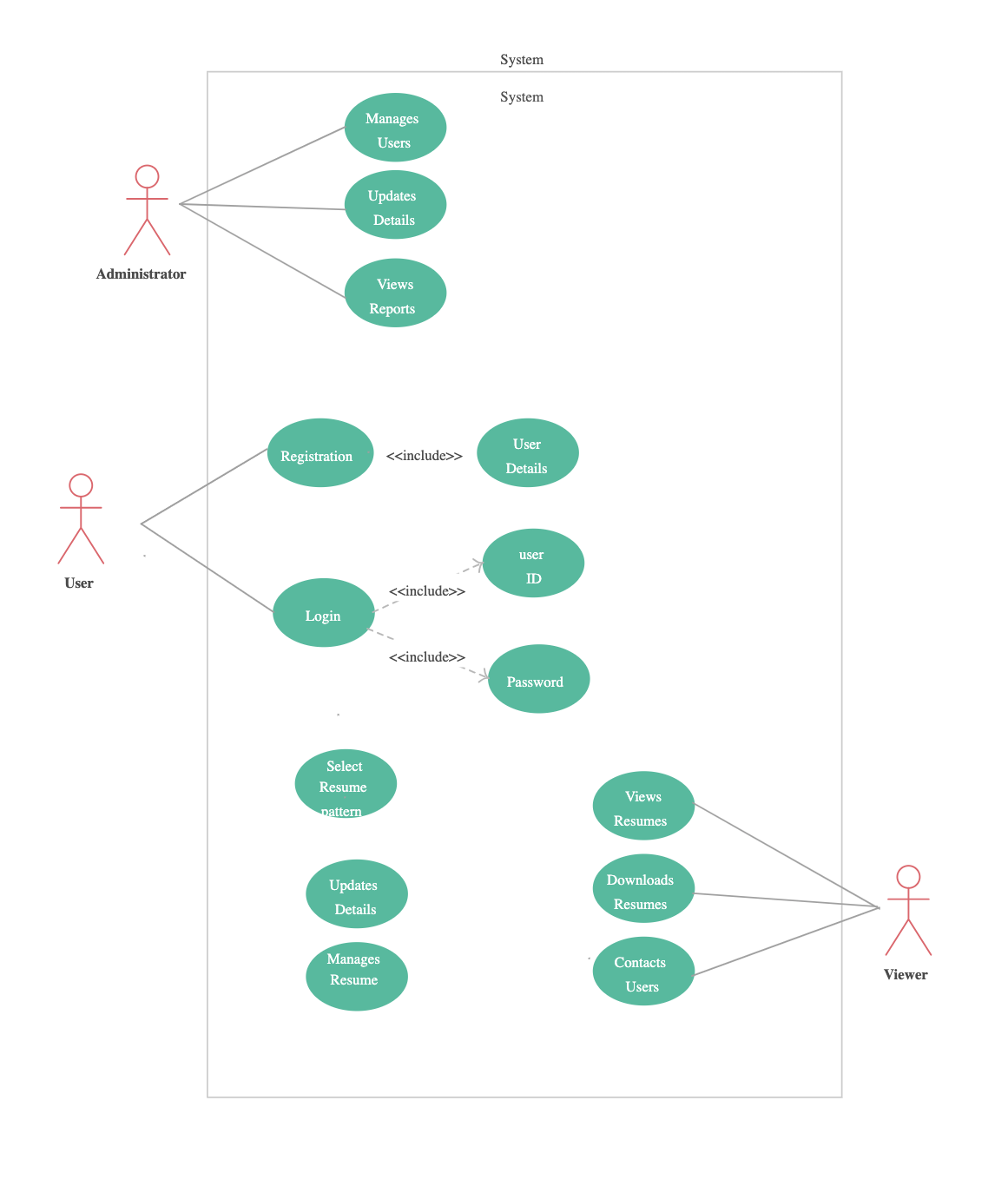
Backend Development Technologies:Server-side frameworks such as Spring Boot or Apache Struts for Java facilitate the development of robust backend systems, handling data processing, authentication, and API integration efficiently.

APIs (Application Programming Interfaces):APIs like LinkedIn API or GitHub API can be leveraged to integrate additional functionalities into the online resume builder, such as importing work history or projects from external platforms, enhancing the completeness and accuracy of user resumes.

Data Storage and Management:Database technologies like MySQL or MongoDB are utilized for storing and managing user data securely, ensuring efficient retrieval and manipulation of resume content.

Version Control Systems:Tools like Git are essential for managing code repositories and collaborating with team members in the development process, enabling version control and tracking of changes over time.

Testing Frameworks:Testing frameworks such as JUnit for Java are employed to ensure the reliability and robustness of the online resume builder application, validating the functionality of different components and features through automated testing procedures.

**USE CASE DIAGRAM:**

**SOURCE CODE**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.LayoutStyle.ComponentPlacement;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class SignUp extends JFrame {

private JPanel contentPane;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_2;

private JTextField textField\_3;

private JTextField textField\_4;

private JTextField textField\_5;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

SignUp frame = new SignUp();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public SignUp() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblFullName = new JLabel("Full Name:");

textField = new JTextField();

textField.setColumns(10);

JLabel lblUsername = new JLabel("Username:");

textField\_1 = new JTextField();

textField\_1.setColumns(10);

JLabel lblEmailId = new JLabel("Email Id: ");

textField\_2 = new JTextField();

textField\_2.setColumns(10);

JLabel lblPhoneNo = new JLabel("Phone No.:");

textField\_3 = new JTextField();

textField\_3.setColumns(10);

JLabel lblPassword = new JLabel("Password:");

textField\_4 = new JTextField();

textField\_4.setColumns(10);

JLabel lblRetypePassword = new JLabel("Retype password:");

textField\_5 = new JTextField();

textField\_5.setColumns(10);

JButton btnRegister = new JButton("Register");

btnRegister.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

SignUpDb s = new SignUpDb();

s.database(textField.getText().toString(),textField\_1.getText().toString(),textField\_2.getText().toString(),textField\_3.getText().toString(),textField\_4.getText().toString());

ResumeUI rui = new ResumeUI();

rui.setVisible(true);

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblFullName)

.addComponent(lblUsername)

.addComponent(lblEmailId)

.addComponent(lblPhoneNo)

.addComponent(lblPassword)

.addComponent(lblRetypePassword))

.addGap(5)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(12)

.addComponent(btnRegister))

.addComponent(textField\_5, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textField\_4, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textField\_3, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textField\_2, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textField\_1, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addContainerGap(170, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblFullName)

.addComponent(textField, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblUsername)

.addComponent(textField\_1, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblEmailId)

.addComponent(textField\_2, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblPhoneNo)

.addComponent(textField\_3, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(ComponentPlacement.UNRELATED)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblPassword)

.addComponent(textField\_4, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(22)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblRetypePassword)

.addComponent(textField\_5, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(btnRegister)

.addContainerGap(21, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

public JTextField getTextField() {

return textField;

}

public void setTextField(JTextField textField) {

this.textField = textField;

}

public JTextField getTextField\_1() {

return textField\_1;

}

public void setTextField\_1(JTextField textField\_1) {

this.textField\_1 = textField\_1;

}

public JTextField getTextField\_2() {

return textField\_2;

}

public void setTextField\_2(JTextField textField\_2) {

this.textField\_2 = textField\_2;

}

public JTextField getTextField\_3() {

return textField\_3;

}

public void setTextField\_3(JTextField textField\_3) {

this.textField\_3 = textField\_3;

}

public JTextField getTextField\_4() {

return textField\_4;

}

public void setTextField\_4(JTextField textField\_4) {

this.textField\_4 = textField\_4;

}

public JTextField getTextField\_5() {

return textField\_5;

}

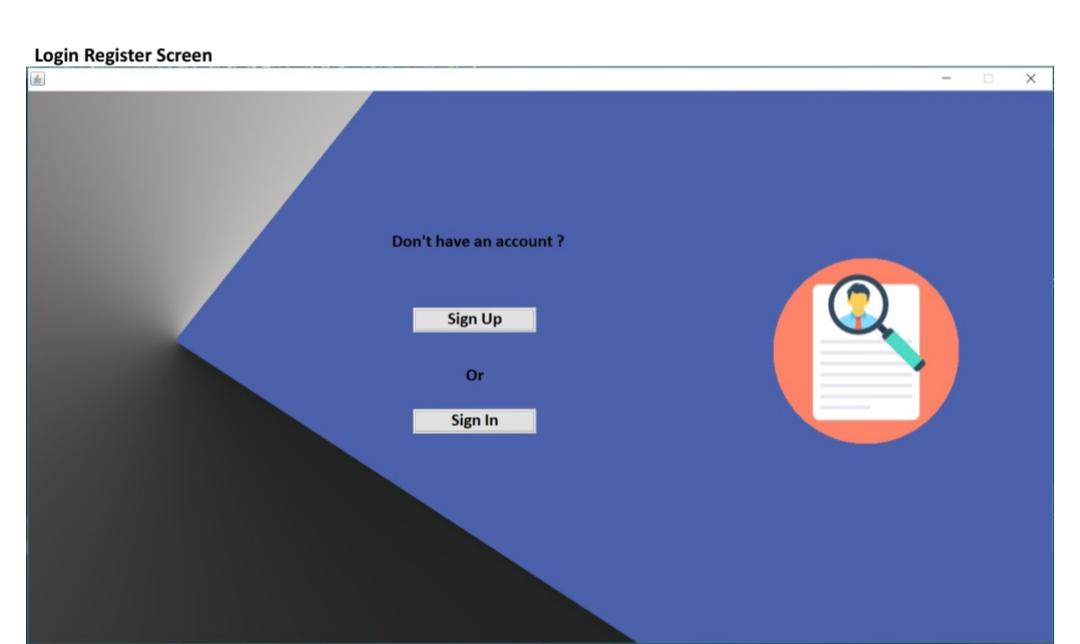
public void setTextField\_5(JTextField textField\_5) {

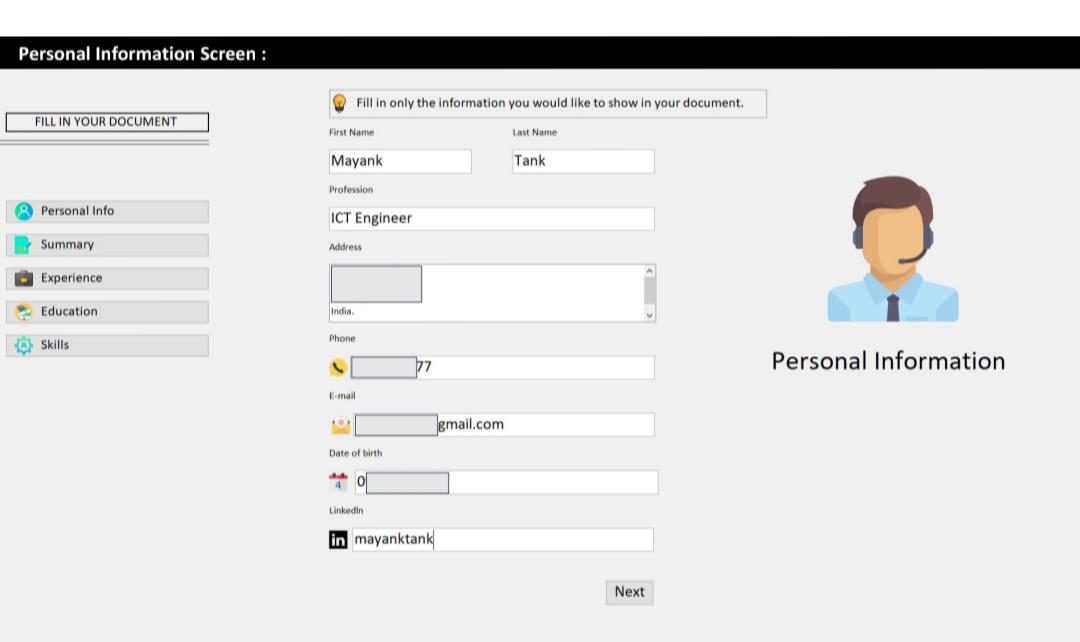
this.textField\_5 = textField\_5;

}

}

**SCREENSHOT(OUTPUT)**





**FUTURE ENHANCEMENTS :**

Web Application Transition:Transition the online resume builder from a desktop application to a web-based platform using technologies such as Spring Boot, React, or Angular. This evolution enables users to access and edit their resumes conveniently from any device with internet connectivity, enhancing accessibility and flexibility.

Mobile Application Development:Develop native mobile applications for iOS and Android platforms using technologies like React Native or Flutter. This advancement empowers users to create and manage their resumes seamlessly on their mobile devices, catering to their on-the-go lifestyle and increasing user engagement.

Cloud Integration:Integrate cloud services such as AWS, Azure, or Google Cloud for enhanced scalability, reliability, and ease of maintenance. Leveraging cloud infrastructure facilitates efficient handling of fluctuating workloads, ensures high system availability, and simplifies software updates and deployments.

Microservices Architecture Implementation:Implement a microservices architecture to enhance scalability, maintainability, and flexibility of the online resume builder. Breaking down the application into smaller, independent services enables more granular control over functionalities such as resume management, template rendering, and user authentication.

Blockchain for Enhanced Security:Explore the integration of blockchain technology to enhance security and transparency in resume management and user authentication processes. Blockchain can provide a decentralized and immutable ledger for storing resume data securely, ensuring data integrity and privacy protection.

AI and Chatbots Integration:Implement artificial intelligence (AI) and chatbots to enhance user experience and support functionalities. AI-powered chatbots can assist users in resume creation, offer personalized recommendations, and provide real-time assistance, enhancing user satisfaction and engagement.

Biometric Authentication Integration:Integrate biometric authentication methods such as fingerprint or facial recognition for secure user authentication and access control. Biometric authentication enhances security and user convenience by replacing traditional password-based authentication methods.

**CONCLUSION**

The Online Resume Builder system provides a robust framework for crafting and managing professional resumes with ease. Its user-friendly interface simplifies the resume creation process, offering features such as template selection, real-time editing, and personalized content suggestions. While the current implementation serves its purpose well, there are numerous avenues for future enhancement to further improve its functionality and user experience.

One proposed enhancement is the integration of a database for persistent data storage, allowing users to save and retrieve their resumes seamlessly. This addition would enhance the system's usability by enabling users to access their resumes from multiple devices and ensuring that their data remains intact even if they navigate away from the platform. Additionally, introducing dynamic pricing strategies and loyalty programs could incentivize users to engage with the platform more frequently, fostering a sense of loyalty and satisfaction.

Furthermore, considering individual user preferences in resume suggestions and formatting could significantly enhance the system's effectiveness. By leveraging machine learning algorithms or natural language processing techniques, the system could analyze user data and provide tailored recommendations based on their career goals, industry preferences, and skill sets. This level of personalization would not only improve user satisfaction but also increase the likelihood of users achieving their desired career outcomes. As the job market continues to evolve, incorporating these enhancements will ensure that the Online Resume Builder remains a valuable tool for job seekers worldwide

**REFERENCE:**

**Horstmann, Cay S., and Gary Cornell. "Core Java Volume I--Fundamentals." Prentice Hall, 2018.**

**Freeman, Eric, and Elisabeth Robson. "Head First Java." O'Reilly Media, 2021.**

**Flanagan, David. "Java in a Nutshell: A Desktop Quick Reference." O'Reilly Media, 2018.**

**Fowler, Martin. "Patterns of Enterprise Application Architecture." Addison-Wesley Professional, 2002.**

**Gamma, Erich, et al. "Design Patterns: Elements of Reusable Object-Oriented Software." Addison-Wesley Professional, 1994.**

**Martin, Robert C. "Clean Code: A Handbook of Agile Software Craftsmanship." Prentice Hall, 2008.**