OBJECT ORIENTED PROGRAMMING LAB

3122 21 5001 010 - C B ANANYA
3122 21 5001 017 - B AYSHWARYA

CV Builder

- A real-world application built using JAVA

Problem Statement

To Create a CV Builder application that

- Provides a template regarding what information to include in a CV (Ex: Picture, Bio-data, Skills, Qualification, Work Experience)
- Collects the information in a form
- Presents it in a structured template in a document by placing fields in appropriate positions in the page
- Generates the document as a pdf and lends it to the user for their utility
- Stores previously entered information (in a database) and retrieves it when searched for making changes to existing CV.

Motivation to take up the problem

A candidate with apt credentials, required skill set and experience for a job often finds difficulty in presenting the same to employers. Crisp and clear format for a resume is essential in order to project the skills one has acquired.

The team, being fervent in building our CV from our first year of college, saw this as an opportunity to find a solution to this common problem.

We also felt that java has a programmer-friendly methodology to provide Graphical-User-Interface to create a form and provides convenient modes of connection to databases. Hence, the team decided to implement the idea in the form of a simple application with a user-friendly interface and sql connectivity for storage and retrieval of data.

Scope and Limitations

Currently, we have provided only one template to the user to present their data. However, the scope of the project could be extended to provide multiple templates with various sets of fields to be included. Choice of color palettes could also be introduced.

The project, as yet, does not have a login/signup process where a user can access only their CV. Searching for one's CV is simply based on a search by first name. Hence, to improve the privacy of the user, we need to implement a registration/signup process, provide each user with an ID and allow them to set a password which could be stored in a database in an encrypted format.

Limitations of the application also include the fact that it just takes the data as it is and helps only with the alignment, orientation and designing part of the CV and converting the input of the form into a pdf. It does not provide suggestions regarding the quality of content.

Future improvements could involve usage of AI to provide suggestions to the user regarding phrasing of sentences and prioritizing the skills acquired based on the position applied for.

Design of the Solution

1. The project is created using netbeans IDE and the following packages are imported:
    - javax.swing
    - java.awt
    - java.sql
    - java.io
    - java.util
    - java.text
    - java.imageio

    The following jar files are also added to the library

    - Javafx-swt.jar
    - Javafx.base.jar
    - Javafx.controls.jar
    - Javafx.fxml.jar
    - Javafx.graphics.jar
    - Javafx.media.jar
    - Javafx.swing.jar
    - javafx.web.jar
    - sqlite -jdbc-3.40.10.jar (for java-sqlite connectivity)

2. The user interface is designed using Swing components such as labels, text fields, buttons and panes. The layout includes sections for personal information, work experience, education and skills.
3. Next, event handlers are implemented for the buttons, such as
    - “SAVE CV” - allows user to save the data in SQLite database
    - “Generate CV” - allows the user to create a pdf from the data
    - “CLEAR” - empties all the fields of the form and allows the user to enter data freshly.
4. The input fields are validated to ensure that all the required fields are filled in before the user can save the CV.
5. iText is used to create the PDF document, which can be readily downloaded.
6. The user’s information is retrieved from the SQLite database and populates the text fields with the saved information when the user wants to edit their CV (with the help of user’s name).

Explore Design Alternatives

- Eclipse can be used to create a Java project in which a template engine, such as Apache Freemarker can be added to generate the CV in PDF, Word, or HTML format from a predefined template.
- Eclipse can be used to create a JavaFX project to create a Desktop application that allows users to create and edit their CVs and export them in different formats like word, PDFs.
- WonderShare PDFElement could be used instead of itextPDF
- Many data fields could be declared as members of the CV class instead of using them inside the methods alone
- File Handling could be used instead of sql tables so that, every time a new field is added to the form, it could be added to the files
- The design could be focused on the sequence of actions performed rather than on the button clicked by the user.

Identification of the process/modules/algorithms

- ☐ saveActionPerformed(ActionEvent evt)
- ☐ generateActionPerformed(ActionEvent evt)
- ☐ clearActionPerformed(ActionEvent evt)
- ☐ attachimageActionPerformed(ActionEvent evt)
- ☐ txt_searchKeyReleased(KeyEvent evt)

OOPs features used

Packages:

The entire java code is **one package- cv.generator**

Java packages primarily used/imported: **javax.swing**, **java.awt**, **java.sql**, **java.io**, **java.util**

Other addition: **itextpdf, jdbc connector**

Swing is used for the GUI of the project - the form and the search menus.

Sql connectivity is used for storing the input data into the sql table and retrieving the stored data when searched for

Itextpdf is used to write available content into the pdf file at the correct location with right amount of spacing and format.

Exception Handling:

Exception handling using try-catch-finally blocks are used at every step of the way for checking

- Null entries
- File Open/close - pdf
- Data type checks (retrieving string data from sql into string variable and int to int, etc)

- Image Formats
- Folder accessible/not when it is selected for saving CV

Inheritance:

The CV class inherits from javax.swing.JLabel class alone. However, many classes of swing package are used throughout the program in the form of packages.

Implementation of modules

The primary methods are:

- saveActionPerformed()
- generateActionPerformed():
- clearActionPerformed():
- attachImageActionPerformed():

When the **SAVE** button is clicked - saveActionPerformed method is invoked. The data entered in the form is filled into the sql table.

When the **FILE** button is clicked, attachimageActionPerformed method is invoked and it takes the user to a navigation panel where they can select an image from the system they are working in.

When the **GENERATE** button is clicked - generateActionPerformed method is invoked. The data in the input labels are written into the pdf in the required format.

When the **CLEAR** button is clicked - clearActionPerformed method is invoked, the data input fields are set to value empty string.

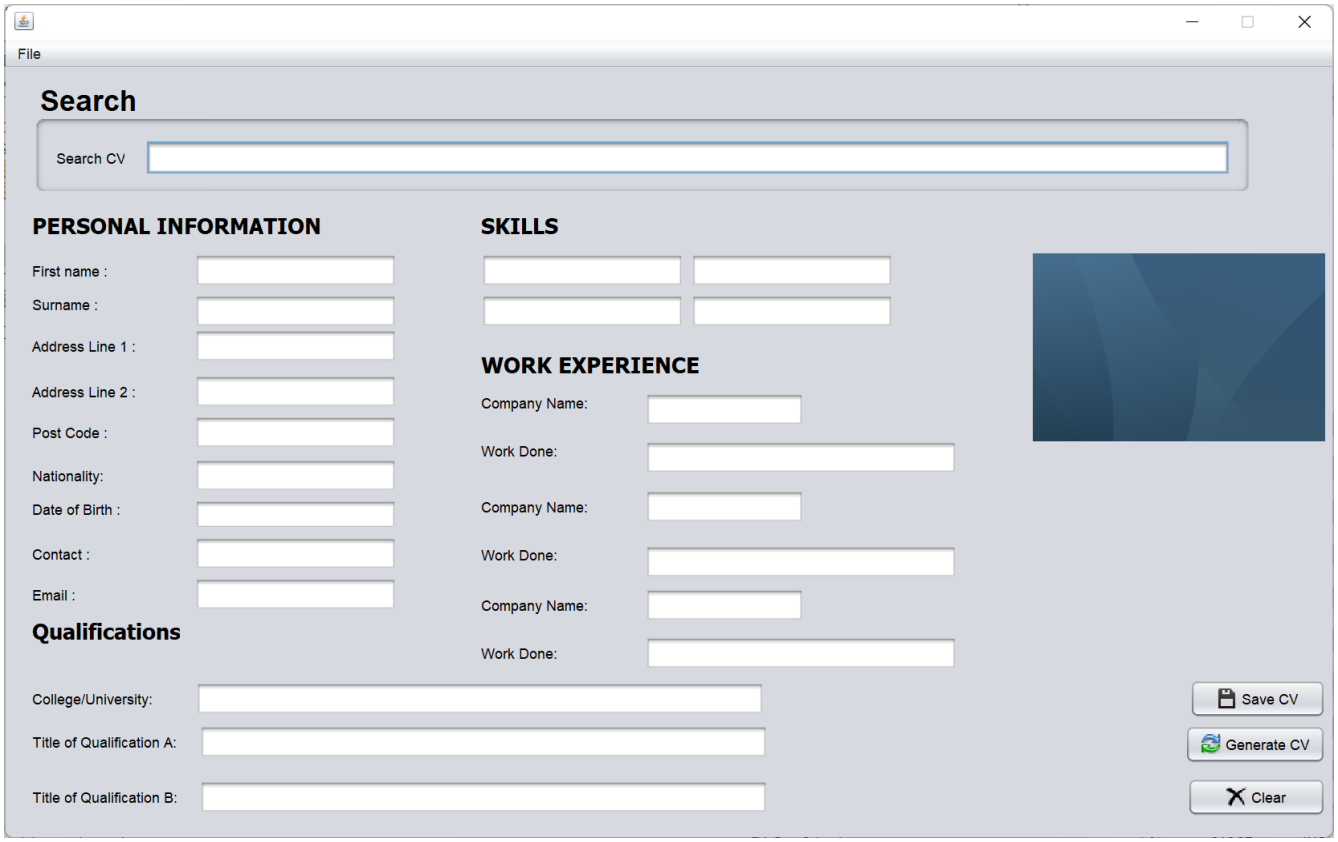
In the search bar, when a name is searched, If the entered name matches with an existing CV's name, the data fields of the form automatically get populated with the data of that CV from the sql table.

Integration of Modules

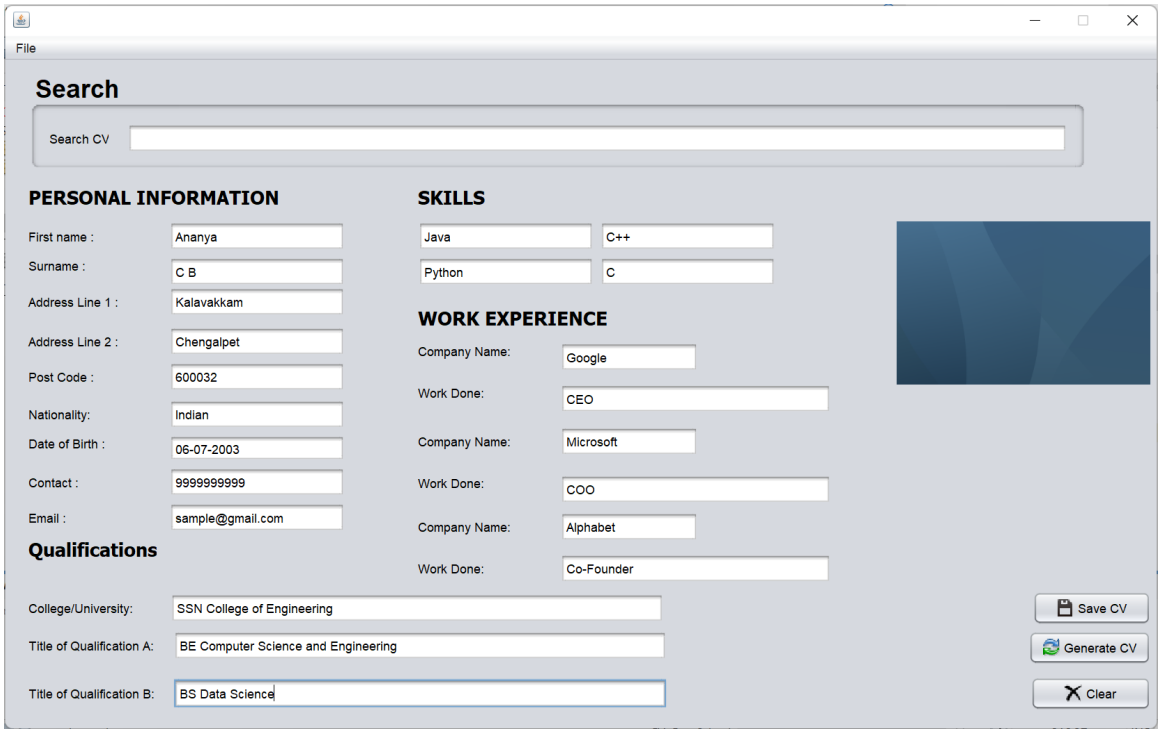
- Data taken as input in one method is carried towards the “Generate CV” function.
- Data from search is used in the program to make editions and subsequent “save” and “generate” actions.
- Every step of the application has try-catch blocks and additional finally blocks to avoid any error that could arise in subsequent methods. This allows smooth integration of modules allowing
    - ➔ No null pointer exceptions
    - ➔ Data of correct data type to flow through the program as parameters
    - ➔ Data from and to correct location (i.e. data comes from the text fields of the form for save function; data comes from the sql tables for the search function)

Output Screenshots

Initial Window

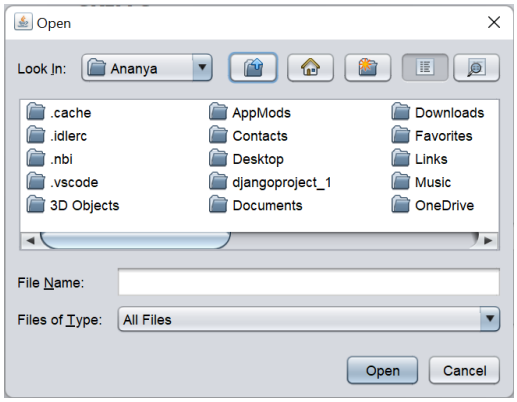


After entering information in text fields

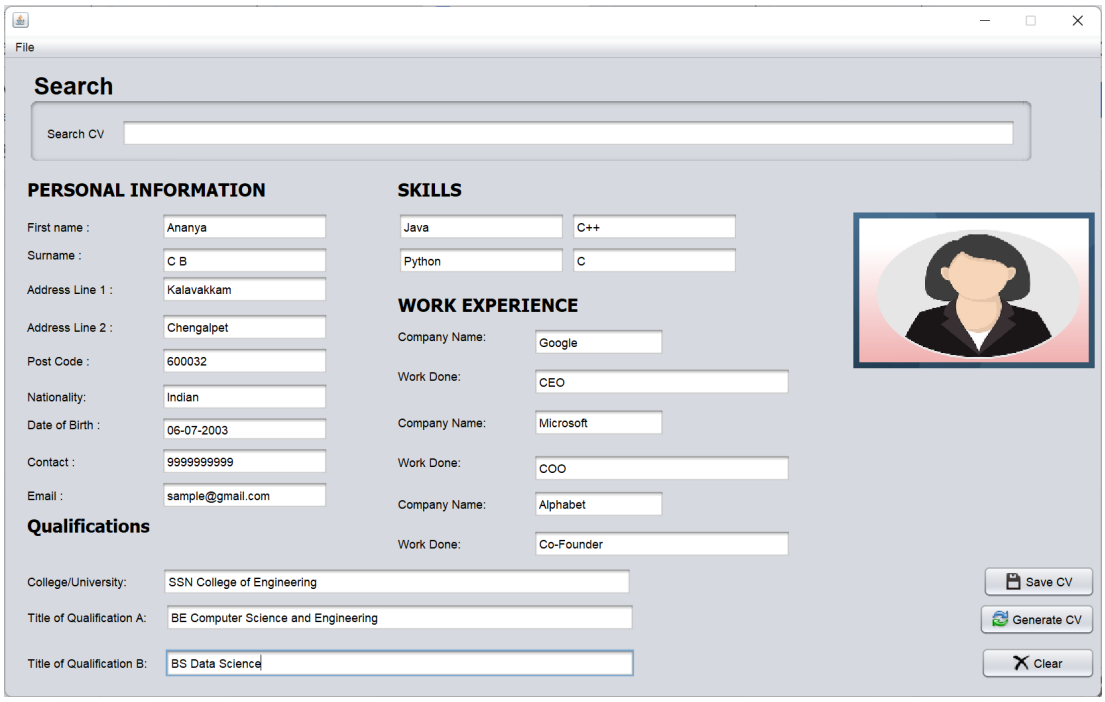


Attach Image Icon

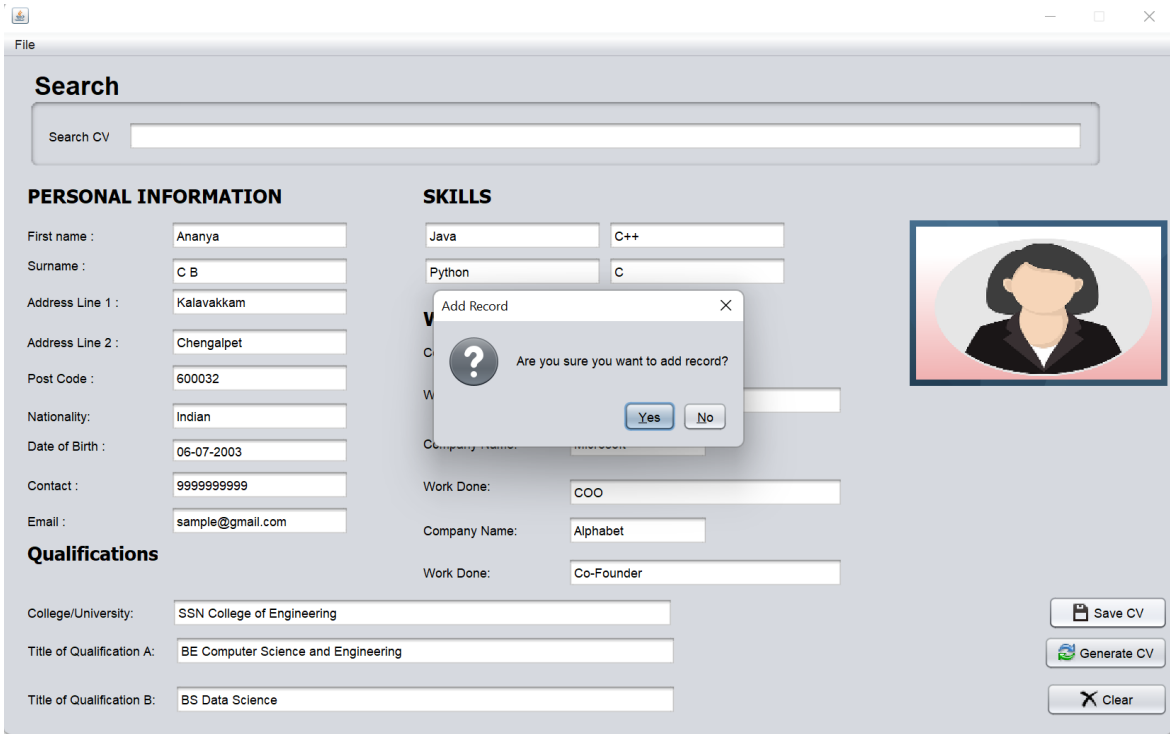
After clicking “Attach Image” - File Navigation window shown



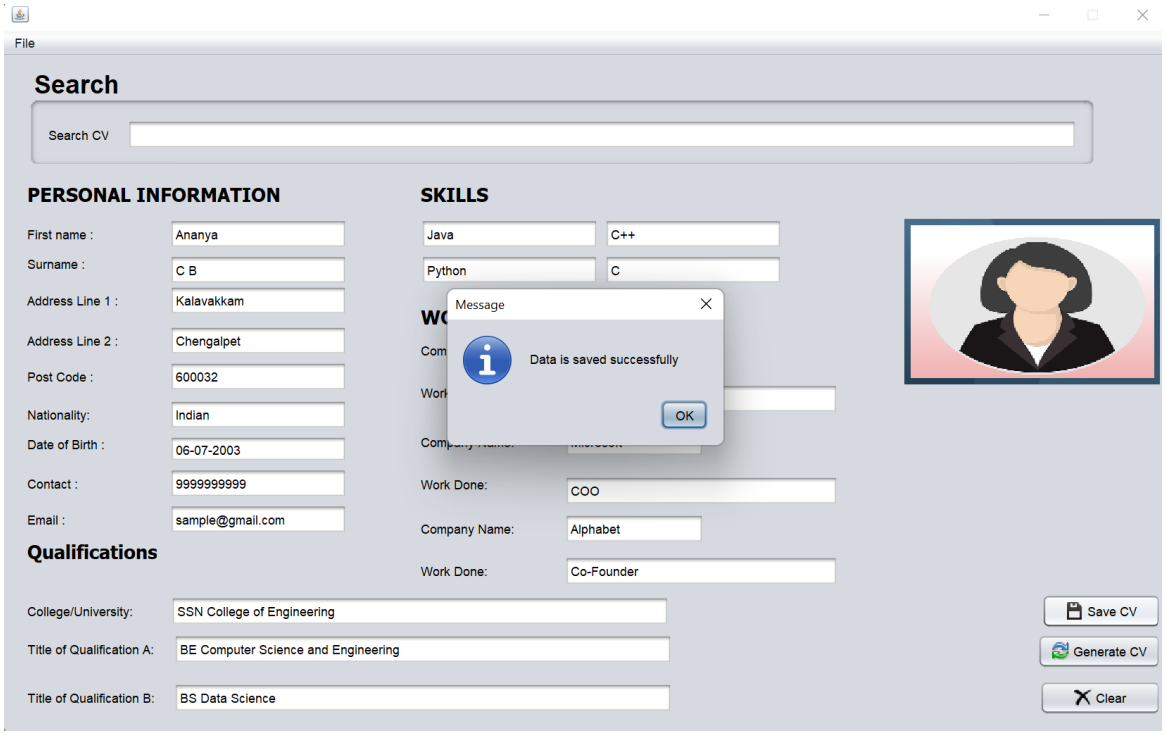
After going to the required location, selecting the required image and clicking open.



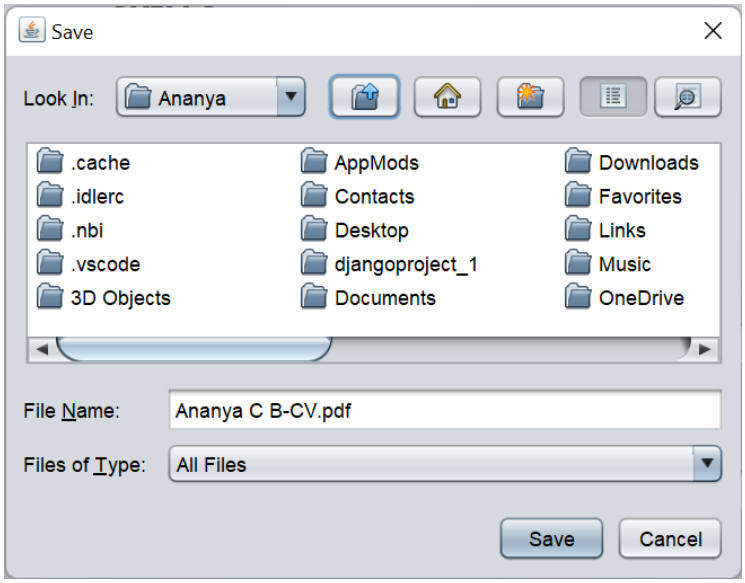
On clicking save CV



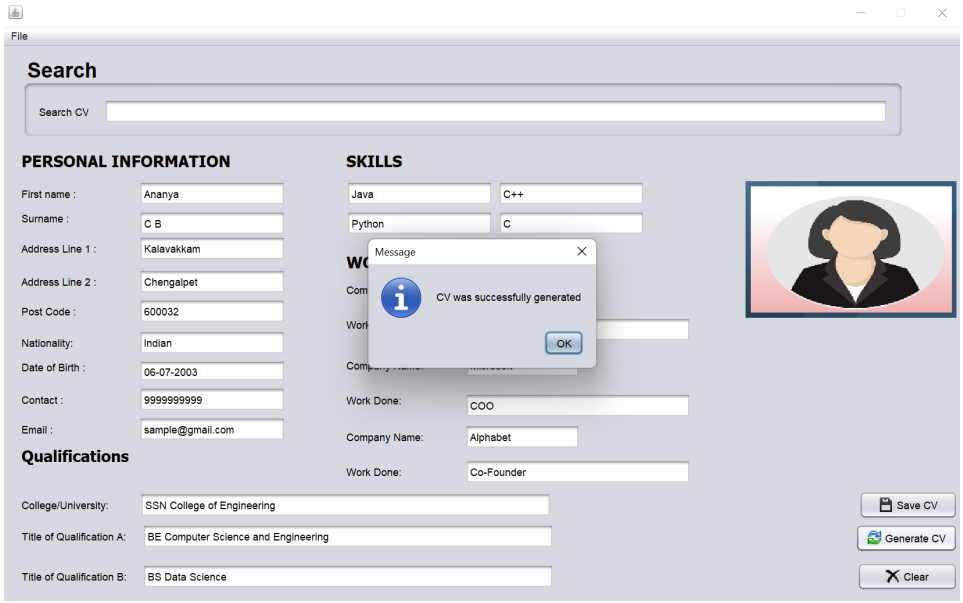
Clicking Yes:



Clicking OK, and then selecting Generate CV. The application will direct you to the Navigation Menu



Go to the required folder and click SAVE.



Going to the selected location and opening the generated pdf

Ananya C B

CONTACT

sample@gmail.com
9999999999
Kalavakkam, Chengalpet, 600032

PERSONAL DETAILS

Nationality: Indian
Date of Birth: 06-07-2003

SKILLS

Java	Python
C++	C

QUALIFICATIONS

SSN College of Engineering
BE Computer Science and Engineering
BS Data Science

WORK EXPERIENCE

Google
- CEO

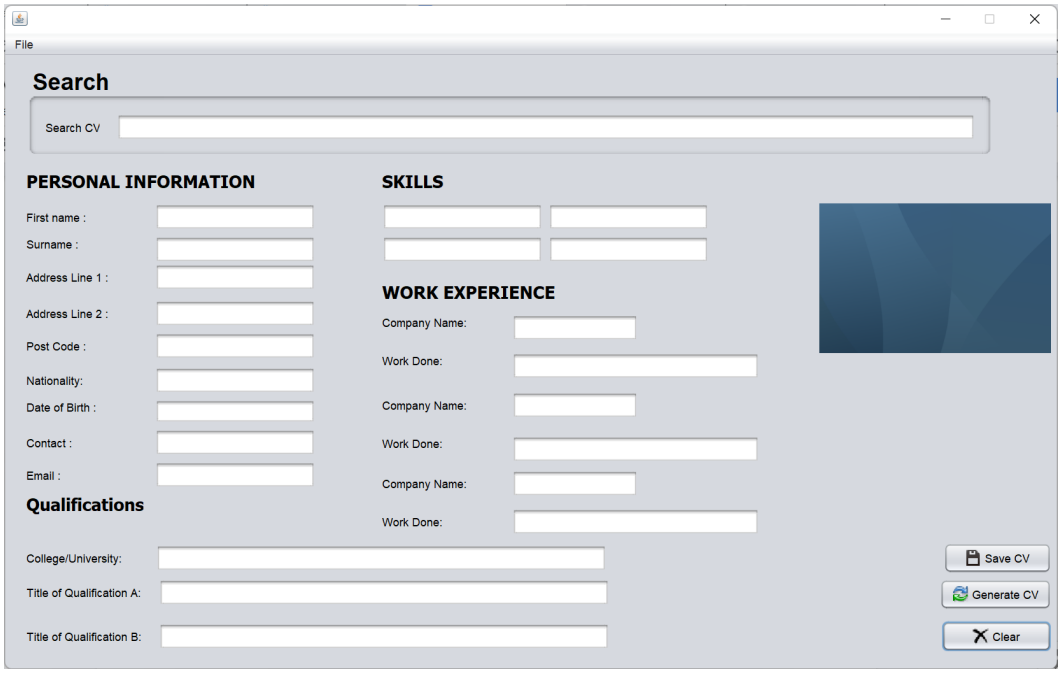
Microsoft
- COO

Alphabet
- Co-Founder

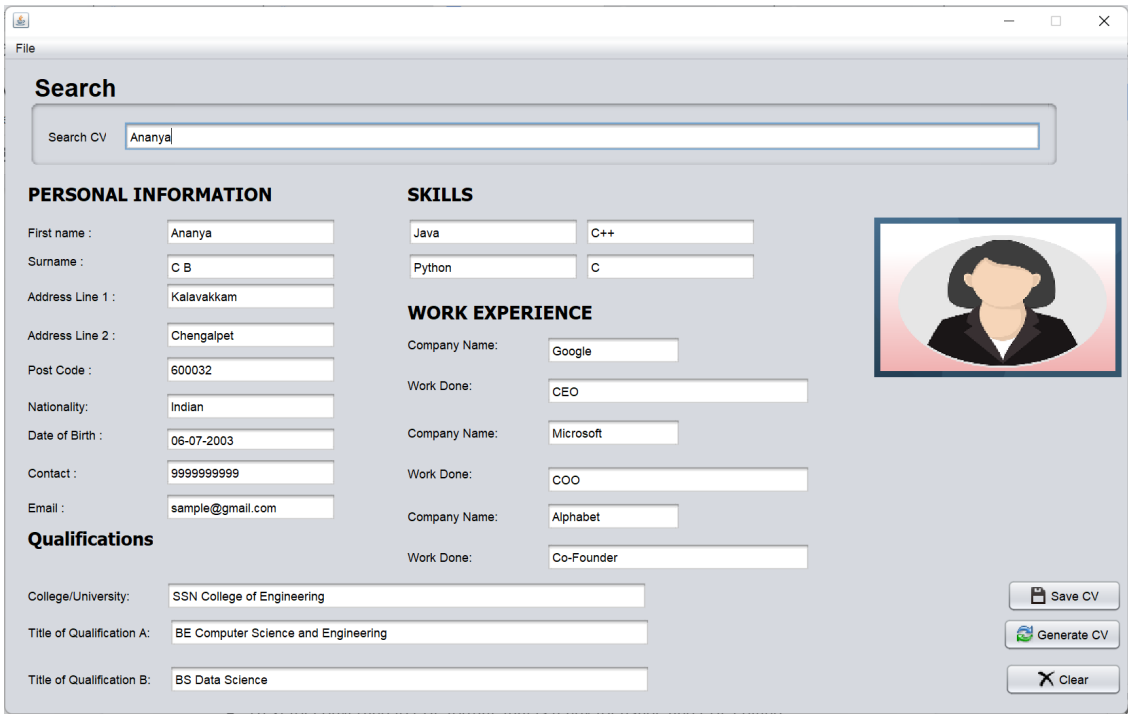
REFERENCES

Available upon request

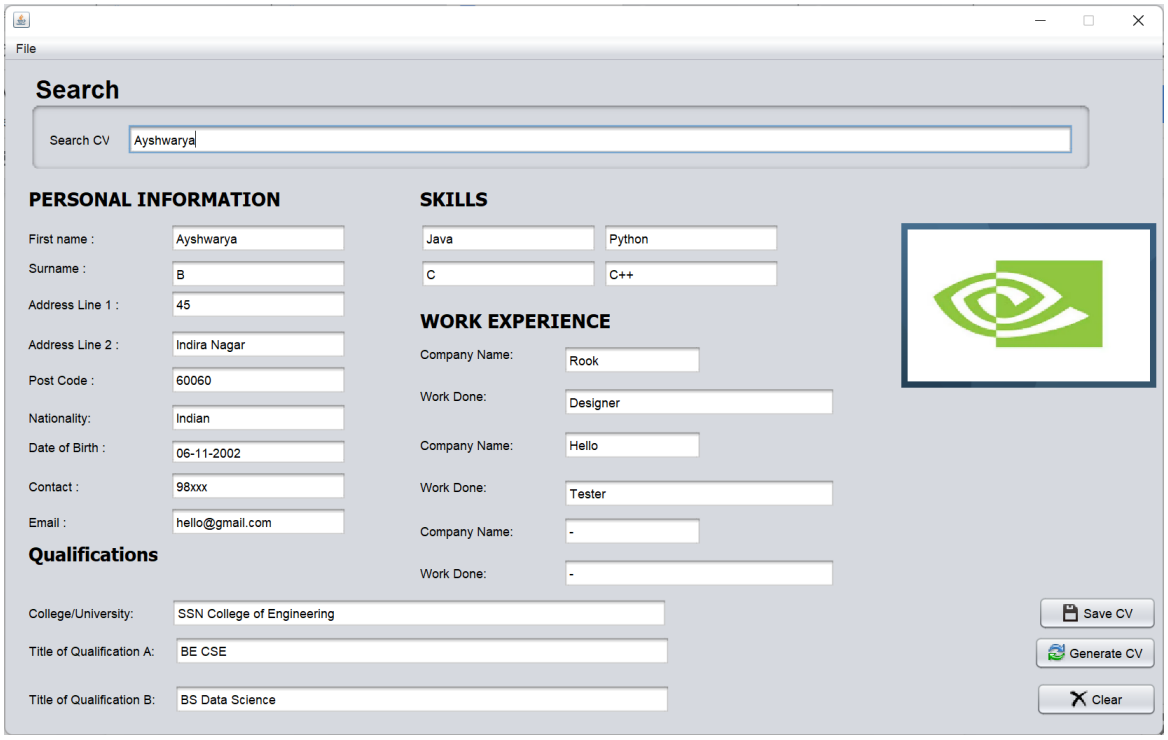
Clicking CLEAR button - clears all entered data



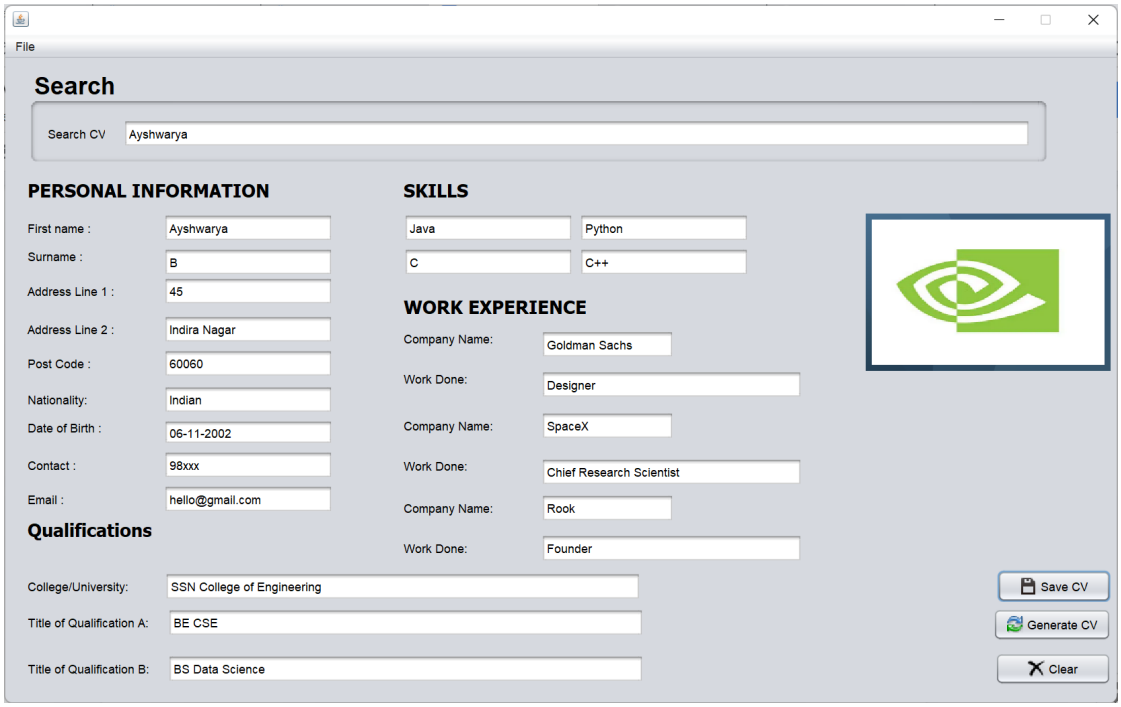
On typing previously saved CV's name in the search bar: (the data populates the text fields)



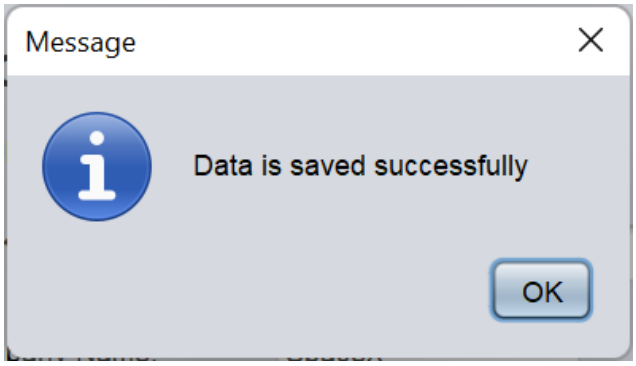
Typing one more existing CV name in the search bar:



Making changes to the existing CV (adding extra work experience)



Clicking SAVE:



Clicking Generate CV, navigating to required location and saving it.

Opening the pdf at the saved location.

Ayshwarya B

CONTACT

hello@gmail.com
98xxx
45, Indira Nagar, 60060

PERSONAL DETAILS

Nationality: Indian
Date of Birth: 06-11-2002

SKILLS

Java	C
Python	C++

QUALIFICATIONS

SSN College of Engineering
BE CSE
BS Data Science

WORK EXPERIENCE

Goldman Sachs
- Designer

SpaceX
- Chief Research Scientist

Rook
- Founder

REFERENCES

Available upon request

Closing the application with the X symbol of the window.

Screenshots of the SQL Table

SQL Table:

select * from cv

id	firstname	surname	add1	add2	postcode	nationality	dob	telephone	email	skill1	skill2	skill3	skill4	company1
3	Gm													bleh
4	hello													bleh
6	Example	Sample	Winter Street	Bangalore	901876	Indian	08-11-2005	99xxxxx	hello@gmail.com	Java	C++	C	Python	Google
8	Ritu	Sharma	T Nagar	Chennai	600040	Indian	06-10-1998	97xxx	example123@gmail.com	Creative Writing	Multi Lingual	Story Telling	Speech Therapist	xSeed
9	Ananya	C B	Kalavakkam	Chengalpet	600032	Indian	06-07-2003	9999999999	sample@gmail.com	Java	Python	C++	C	Google
10	Ayshwarya	B	45	Indira Nagar	60060	Indian	06-11-2002	98xxx	hello@gmail.com	Java	C	Python	C++	Goldman Sachs

wdone1	company2	wdone2	company3	wdone3	university
bleh	bleh	bleh	bleh	bleh	bleh
bleh	bleh	bleh	bleh	bleh	bleh
Testing	Microsoft	Coding	Goldman Sachs	Management	SSN College of Engineering
Kindergarten Teacher	Daisy school for special children	Held English classes for students with hearing and speaking disabilities			MOP Vaishnav College for Women
CEO	Microsoft	COO	Alphabet	Co-Founder	SSN College of Engineering
Designer	SpaceX	Chief Research Scientist	Rook	Founder	SSN College of Engineering

wdone2	company3	wdone3	university	qual1	qual2
bleh	bleh	bleh	bleh	bleh	bleh
bleh	bleh	bleh	bleh	bleh	bleh
Coding	Goldman Sachs	Management	SSN College of Engineering	BE Computer Science and Engineering	BS Data Science
Held English classes for students with hearing and speaking disabilities			MOP Vaishnav College for Women	BSc Psychology	BEd
COO	Alphabet	Co-Founder	SSN College of Engineering	BE Computer Science and Engineering	BS Data Science
Chief Research Scientist	Rook	Founder	SSN College of Engineering	BE CSE	BS Data Science

Inference and future extension

Inference:

A CV/Resume builder (generator) has been implemented using JAVA Programming Language, NetBeans IDE and SQLite.

- NetBeans IDE is used to develop the application's user interface and backend functionality.
- SQLite is used as the database management system to store and retrieve information about users and their resumes.
- iText for converting to PDF format, that is ready for usage and PDF editing.

Future extension:

1. Adding multiple templates: Allowing users to choose from a variety of templates, each with its own design and layout, can address the limitation of limited customization options.
2. Incorporating a built-in spell checker and grammar checker to improve the quality of the CV.
3. Adding a design editor: Allowing users to customize the template's design, colors, font, and layout to make it more visually appealing.
4. Incorporating a built-in resume optimization tool that suggests changes for the user based on their industry and job title.
5. Optimizing for ATS (Applicant Tracking System): Incorporating keywords and phrases commonly used by ATS systems to increase the chances of the CV passing through the initial screening process.
6. Adding login/signup page with increased privacy for users