

A
CAPSTONE (Major) Project Report
on
CareerCrafter – Resume Analyzer



Submitted by
Devyansh Rajput
Registration Number – 22MCA10061

Guided by

Dr Ramraj Dangi
Assistant Professor
(Guide)

Dr. Raghavendra Mishra
Assistant Professor
(Reviewer)

Dr. N Jothiaruna
Assistant Professor
(Reviewer)

**Submitted in partial fulfillment of the requirement for
the degree of
“Master of Computer Applications”**

Submitted to
School of Computing Science and Engineering
VIT Bhopal University
Bhopal (MP) – 466114

April 2024



VIT BHOPAL UNIVERSITY, MP - 466114

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

CANDIDATE'S DECLARATION

I hereby declare that the dissertation entitled "CareerCrafter – Resume Analyzer" is my own work conducted under the supervision of Dr Ramraj Dangi, Assistant Professor, School of Computing Science and Engineering at VIT University, Bhopal.

I further declare that to the best of my knowledge this report does not contain any part of work that has been submitted for the award of any degree either in this university or in other university / deemed university without proper citation.

Devyansh Rajput

22MCA10061

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 15 April 2024

Dr Ramraj Dangi
Assistant Professor



VIT UNIVERSITY BHOPAL, MP – 466114

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the work embodied in this Capstone Project Report entitled “CareerCrafter – Resume Analyzer” has been satisfactorily completed by Devyansh Rajput. Registration No. 22MCA10061 in the School of Computing Science and Engineering at VIT University, Bhopal. This work is a bonafide piece of work, carried out under my/our guidance in the School of Computing Science and Engineering for the partial fulfilment of the degree of Master of Computer Applications.

Dr Ramraj Dangi

Assistant Professor

Forwarded by

Approved by

Dr Anjali Mathur
Program Chair

Dr Poonkuntran S
Professor & Dean

ACKNOWLEDGEMENT

I extend my heartfelt gratitude to all those who contributed to the successful completion of the CareerCrafter – Resume Analyzer. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this Capstone Project.

I am using this opportunity to express my deepest gratitude and special thanks to the Dr Anjali Mathur Assistant professor and program chair of MCA at VIT Bhopal who guided me throughout the process, Dr Ramraj Dangi Assistant Professor and Dr.Raghavendra Mishra Senior Assistant Professor who guided and motivated me on the correct path and allowing me to carry out my project development.

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible. If there is a driving force that kept us going on in doing this project, it is the constant support of my guide Dr. Ramraj Dangi and I present my sincere and heartiest thanks to him, for giving me a patient hearing and clearing my doubts.

I would like to express my profound thanks to Dr Poonkuntran S (Dean of SCSE) for their support and encouragement towards me at every stage in successful completion of my project and my degree.

Devyansh Rajput

22MCA10061

ABSTRACT

CareerCrafter is an innovative tool designed to revolutionize the job application process by providing personalized insights and recommendations to job seekers. Using cutting-edge technology, CareerCrafter analyzes resumes and offers tailored advice to help job seekers align their skills with their ideal career paths. This project follows an iterative development process, which incorporates user feedback to refine the analyzer's accuracy and effectiveness.

The design philosophy of CareerCrafter is rooted in simplicity and effectiveness, offering an intuitive user interface and a potent analytical engine. In the rapidly evolving job market, CareerCrafter emerges as a socially relevant solution that democratizes professional growth and bridges the gap between education and employment. This paper presents the development, implementation, and social impact of CareerCrafter as a strategic partner for job seekers in a competitive market.

CareerCrafter's impact extends beyond individual job seekers to the broader community. By democratizing professional growth and increasing access to job search resources, CareerCrafter helps to address the systemic inequalities that exist in the job market. This makes CareerCrafter a socially impactful tool that empowers individuals and promotes economic mobility.

CareerCrafter is committed to data security and privacy. All user data is encrypted and stored securely, and CareerCrafter never shares user information with third parties without explicit consent. This commitment to user privacy and security ensures that job seekers can trust CareerCrafter to provide personalized insights and recommendations without compromising their sensitive information.

Overall, CareerCrafter represents a significant step forward in the job search and career development industries. Its innovative technology, commitment to user privacy and security, and social impact make it a valuable resource for job seekers and the broader community alike.

TABLE OF CONTENTS

Inner first page	(i)
Declaration	(ii)
Certificate	(iii)
Acknowledgement.....	(iv)
Abstract.....	(v)
Table of Contents.....	(vi)
CHAPTER 1.....	
INTRODUCTION	
1.1. Objective	
1.2. Proposed System	
1.3. Profile of the Problem. Rational/Scope of the Study	
CHAPTER – 2.....	
PROBLEM ANALYSIS.....	
2.1 Product Definition	
2.2 Feasibility Study.....	
2.2.1 Technical feasibility	
2.2.2 Financial feasibility	
CHAPTER - 3.....	
SOFTWARE REQUIREMENT ANALYSIS	
3.1 Technologies.....	
3.1.1 HTML.....	
3.1.2 CSS	
3.1.3 Python.....	
3.1.4 MySQL	
3.1.5 Streamlit.....	
CHAPTER - 4.....	
DESIGN.....	
4.1 Data Flow Diagram (DFD).....	
CHAPTER – 5	
IMPLEMENTATION	

5.1	Front-End Implementation
5.2	Back-End Implementation
5.3	Integration of Front-End and Back-End.....
5.4	Testing.....
5.5	Post Implementation and Maintenance

CHAPTER – 6

SYSTEM TESTING

6.1	Testing Strategies
-----	--------------------------

CHAPTER – 7

CONCLUSION.....

7.1	Current state of the project.....
7.2	Remaining areas of concern
7.3	Technical and Managerial lessons learnt
7.4	Future Scope.....

CHAPTER – 8

DEMONSTRATION/SCREENSHOTS/CODES

BIBLIOGRAPHY

CHAPTER 1

INTRODUCTION

Introduction: CareerCrafter - A Personalized Resume Analyzer built with Python and MySQL

The landscape of job applications is constantly evolving, demanding tools that prioritize precision, personalization, and a seamless user experience. This project delves into the design, development, and evaluation of "CareerCrafter," a novel resume analyzer built using Python and MySQL.

Addressing Job Application Needs:

Crafting a perfect resume has become an essential aspect of professional growth, yet many job seekers struggle to align their resumes with industry standards and expectations. This project tackles this challenge by leveraging the strengths of Python and MySQL.

The Power of Python and MySQL:

This work explores how Python's powerful libraries and MySQL's robust database capabilities come together to create a seamless and efficient user experience. This ensures smooth interaction even for users with limited technical knowledge.

CareerCrafter: A Focus on Personalization

The title "CareerCrafter" emphasizes the application's unwavering commitment to providing personalized guidance. The project delves into the robust analytical protocols implemented to provide actionable feedback and guidance.

Beyond Personalization: A Feature-Rich Experience

While personalization is paramount, CareerCrafter goes beyond this core function. The application boasts a comprehensive feature set including:

Resume Analysis: Experience detailed and personalized analysis of your resume.

Skill Enhancement Recommendations: Foster dynamic growth with recommendations for skill enhancement.

Industry Alignment: Align your skills and experiences with industry standards and expectations.

Evaluation and Contribution:

The project incorporates a thorough analysis of the challenges encountered during development, such as optimizing resume analysis, handling diverse data efficiently, and integrating personalization protocols. An evaluation section presents performance benchmarks across various test cases alongside usability studies conducted with target users.

This work aims to demonstrate the efficacy of Python and MySQL as a foundation for building feature-rich and personalized resume analyzers. It contributes valuable insights into optimizing resume analysis within the Python and MySQL framework and integrating robust personalization practices. The project concludes with recommendations for future enhancements and paves the way for further research in the domain of personalized resume analyzers.

1.1 Objective

Overall Objective:

To design, develop, and evaluate a personalized and efficient resume analyzer named "CareerCrafter" using Python and MySQL.

Technical Objectives:

Investigate the effectiveness of Python and MySQL in building a real-time resume analyzer with a focus on precision and responsiveness across various user inputs.

Design and implement a personalized recommendation system utilizing robust analytical protocols to provide actionable feedback within the CareerCrafter.

Explore efficient methods for handling diverse data within the application, ensuring smooth interaction and low latency.

Usability and User Experience Objectives:

Develop a user-friendly and intuitive interface leveraging Python's libraries and MySQL's capabilities to promote ease of use and foster a seamless user experience within the CareerCrafter.

Evaluate the usability and user experience of the CareerCrafter through user studies to identify areas for improvement.

Personalization Objective:

Analyze and implement best practices for personalized guidance within the CareerCrafter, ensuring relevant and beneficial recommendations.

Additional Considerations:

You can tailor these objectives to better align with your specific research interests and the functionalities implemented in your CareerCrafter.

Consider including an objective that focuses on comparing the performance and personalization of your application to existing resume analyzers.

User Experience Objectives

Intuitive UI Design: Craft a user-friendly and visually appealing interface for CareerCrafter, leveraging Python's libraries and MySQL's capabilities to ensure a seamless interaction experience.

Usability Evaluation: Conduct usability studies with target users to assess CareerCrafter's ease of use, navigation efficiency, and overall user satisfaction.

Comparative Objective:

Benchmark Performance: Compare CareerCrafter's performance (speed, responsiveness, memory usage) against popular market alternatives, demonstrating the potential advantages of Python and MySQL in resume analyzer development.

1.2 Proposed System

The proposed system, CareerCrafter, is a personalized and efficient resume analyzer built using Python and MySQL. Here's a breakdown of its key components:

1. Client-Side Architecture (Streamlit App):

Streamlit UI: Leverages Python's powerful libraries to create an intuitive and responsive user interface. This ensures a smooth user experience across diverse user inputs.

Real-time Analysis: Integrates a real-time analysis library (e.g., Natural Language Toolkit) to facilitate instant resume analysis and personalized feedback.

Personalized Recommendations: Implements functionalities for providing personalized recommendations for skill enhancement and industry alignment.

2. Backend Architecture:

The choice of backend architecture depends on your specific requirements and scalability goals. Here are two potential options:

MySQL: MySQL's capabilities to promote ease of use and foster a seamless user experience within the CareerCrafter.

Custom Server Implementation: Offers greater control and flexibility over data management and functionalities. This approach requires building and maintaining server-side components, potentially increasing development complexity.

3. Communication Protocol:

Define a communication protocol for exchanging user data and recommendations between the client-side app and the backend server.

4. Security Measures:

Implement robust security measures throughout the system:

Secure user authentication and authorization to prevent unauthorized access.

Secure communication channels with encryption (HTTPS) to protect data in transit.

Secure data storage practices to safeguard user information on the server-side.

5. Development and Deployment:

Utilize best practices for Python app development, including code organization, unit testing, and integration testing.

Deploy the application to relevant platforms following their guidelines:

This proposed system acts as a blueprint for developing your CareerCrafter. You can refine it further based on your chosen backend solution, specific security protocols, and any additional features you plan to implement.

1.3 PROFILE OF THE PROBLEM. RATIONALE/SCOPE OF THE STUDY

Profile of the Problem

The ubiquity of job applications has made resume analyzers an integral part of professional growth. However, many existing resume analyzers suffer from limitations that hinder a truly personalized and efficient user experience:

Performance Issues: Resume analyzers built with traditional technologies often face performance bottlenecks, especially with complex resumes. This can translate into slower response times, lagging interfaces, and an overall diminished user experience.

Personalization Concerns: Many resume analyzers lack robust personalization protocols or employ generic feedback practices.

Inflexible UI Frameworks: Traditional development frameworks sometimes restrict the ability to create highly customized, visually appealing, and responsive user interfaces, ultimately impacting user satisfaction.

Rationale for the Study

This project introduces the CareerCrafter, aiming to address the identified limitations of existing resume analyzers. Key motivations behind this project include:

Exploring Python and MySQL's Potential: Investigating Python and MySQL as viable solutions for building resume analyzers that prioritize both precision and an exceptional user experience.

Prioritizing Personalization: Developing personalized recommendation protocols as the foundation of the application, making personalization a core feature of CareerCrafter.

User-Centric Design: Leveraging Python's libraries and MySQL's capabilities to create an intuitive and engaging user interface that facilitates effortless interaction.

Scope of the Study

This project focuses on the design, development, and evaluation of the CareerCrafter. The scope encompasses:

Technical Implementation: Selection of architectural components, implementation of real-time analysis, diverse data handling, and personalization mechanisms using Python and MySQL.

Usability Assessment: Conducting user studies to evaluate the user experience, ease of use, and perceived quality of the CareerCrafter.

Performance Analysis: Benchmarking the application's performance metrics on various test cases against established resume analyzers.

Personalization Evaluation: Assessing the efficacy of implemented personalization protocols and identifying potential improvements.

CHAPTER - 2

PROBLEM ANALYSIS

To develop the CareerCrafter, a thorough analysis of the challenges associated with resume analyzer development is necessary. The following key areas of analysis inform the design and implementation process in this project:

Performance Optimization:

Real-time Analysis: Managing the efficient and reliable propagation of analysis in real-time between users, particularly for complex resumes. This requires exploring suitable real-time analysis libraries or protocols.

UI Responsiveness: Maintaining a smooth and responsive user interface, even as a high volume of data is rapidly processed. Python's processing and state management capabilities must be carefully assessed for this purpose.

Diverse Data Handling: Optimizing diverse data processing without compromising quality or introducing performance lags. Considerations include compression techniques, efficient data caching, and smart loading strategies.

Personalization Implementation:

Personalized Recommendations: Researching and selecting robust analytical algorithms to implement personalized recommendations that provide actionable feedback. Key management aspects should also be analyzed carefully.

Secure Authentication: Implementing secure user authentication methods to prevent unauthorized account access and protect sensitive information. This includes investigating best practices for password management or integration with third-party authentication providers.

Data Storage and Handling: Designing secure mechanisms for storing user data and recommendations, both on the device and on the backend server (if applicable). Best practices for data encryption and access controls are crucial.

User Experience (UX) Design:

Interface Design: Utilizing Python's libraries effectively to design an intuitive interface that supports easy navigation, clear feedback threading, and a visually appealing experience.

Usability Testing: Planning and conducting usability studies to gather feedback on the application's navigation, feature set, and overall ease of use, ultimately informing necessary refinements.

Additional Considerations

Backend Architecture: Choosing a suitable backend architecture based on the scalability requirements, security constraints, and data management needs of the CareerCrafter.

Platform Compatibility: Thoroughly testing the application across target platforms to ensure a consistent user experience and address platform-specific issues.

This problem analysis highlights the fundamental technical and user-centered challenges addressed within this project to deliver a personalized, efficient, and user-friendly resume analyzer using Python and MySQL.

2.1 Product Definition

1. Introduction

The CareerCrafter is a personalized and efficient resume analyzer built using Python and MySQL. It aims to provide users with a seamless, intuitive, and personalized experience for real-time resume analysis.

2. Project Objectives

Technical Objectives:

Develop a real-time resume analyzer leveraging Python for a responsive and visually appealing user interface across various user inputs.

Implement robust personalization protocols, including end-to-end encryption, for personalized feedback and user data protection.

Design an efficient architecture for handling diverse data with minimal impact on performance.

Usability Objectives:

Create a user-friendly and intuitive interface that fosters ease of use and a smooth interaction experience.

Conduct user studies to evaluate usability and identify areas for improvement.

3. Functionality

Real-time Analysis:

Users can receive detailed and personalized analysis of their resumes instantaneously.

Explore integration with real-time analysis libraries like Natural Language Toolkit for efficient resume analysis.

Personalized Recommendations:

Users can receive personalized recommendations for skill enhancement and industry alignment.

User Management:

Secure user registration and login processes.

Security Features:

Enforce strong password policies for user accounts. Utilize end-to-end encryption for messages to ensure message confidentiality. Secure data storage practices to safeguard user information on the server-side.

4. Target Platform

The CareerCrafter will be developed as a cross-platform application targeting:

Web (primary focus)

5. Technology Stack

Front-End (Web App):

Python Framework for building the user interface and application logic.

Additional Python packages as needed for specific functionalities (e.g., real-time analysis library, UI components).

Back-End (Server):

Two potential solutions are considered:

Firebase: A cloud-based platform offering real-time database, user authentication, and storage services. This provides a quick and scalable solution for the project.

Custom Server Implementation: Offers greater control over data management and functionalities but requires building and maintaining server-side components, increasing development complexity. The specific choice will depend on project requirements and scalability goals.

6. Development Methodology

Agile development methodology will be used to facilitate iterative development, testing, and feedback loops. A version control system (e.g., Git) will be used to manage code changes and track development progress. Unit testing and integration testing will be conducted to ensure code quality and functionality.

7. Evaluation Plan

Performance Testing:

Benchmark the application's performance (speed, responsiveness, memory usage) on various test cases. This helps identify potential bottlenecks and areas for optimization.

Usability Testing:

Conduct user studies with target users to evaluate the application's usability, user interface design, and overall user experience.

Personalization Evaluation:

Assess the effectiveness of implemented personalization protocols and identify potential improvements.

8. Project Deliverables

A fully functional CareerCrafter resume analyzer for Web.

Project documentation including:

System design document outlining the application architecture.

Source code with clear documentation and comments.

User manual explaining how to use the application.

2.1 Feasibility Study

2.1.1 Technical Feasibility

Python and MySQL's Suitability:

Python and MySQL are powerful tools, allowing you to develop a robust and efficient resume analyzer. This can significantly reduce development time and resources compared to building separate native apps.

Python is known for its capabilities in creating high-performance and visually appealing user interfaces (UI). This is crucial for a resume analyzer where responsiveness and aesthetics are key factors in user experience.

Python and MySQL are actively maintained, ensuring a large and supportive developer community. This provides access to resources, tutorials, and assistance if needed during development.

Backend Requirements:

Several backend solutions can support a resume analyzer like CareerCrafter. Cloud-based platforms like Firebase offer real-time database functionality, user authentication, and storage services, all readily available through APIs. This simplifies backend development for your project.

A custom server implementation might be considered if you require more control over data storage, personalization protocols, or unique features beyond what a cloud platform offers. However, this would require more development expertise and ongoing server maintenance.

Skills Availability:

Assess your current skills in Python development. A strong understanding of Python programming and MySQL's system is essential.

If you lack some skills, consider taking online courses, attending workshops, or collaborating with someone who possesses the necessary expertise. Remember to factor skill development time into your project timeline.

2.1.2 Financial Feasibility

Development Costs:

Developer Costs: Estimate the time required to develop the CareerCrafter application based on your feature set and chosen backend solution. If you're a student developer, factor in the time investment as a cost associated with your project work.

Software and Tools: Identify any specialized software or development tools needed (e.g., design tools, IDEs). Factor in any associated licensing costs.

Backend Costs: If using Firebase, estimate usage costs based on projected features and user base. Building a custom server incurs costs for server infrastructure, database services, and ongoing maintenance.

Potential Revenue Models (Post-Thesis):

While your project may not focus on commercialization, exploring potential revenue models demonstrates an understanding of its long-term viability.

Subscription Model: Offer premium features like increased storage, custom themes, or advanced functionalities through tiered subscription plans.

In-App Advertisements: Implement non-intrusive advertising that doesn't disrupt user experience. This can be a source of revenue without compromising user satisfaction.

Partnerships: Explore collaborations with other companies or services to integrate features or content within the CareerCrafter. This could provide additional value to users and generate revenue streams.

CHAPTER - 3

SOFTWARE REQUIREMENT ANALYSIS

The Software Requirement Analysis (SRA) phase is crucial for defining what the CareerCrafter needs to achieve. It involves gathering, documenting, and understanding the needs of stakeholders (users, developers) to ensure the final application meets all expectations. Here's a breakdown of key aspects involved in the SRA for CareerCrafter:

1. Functional Requirements:

User Requirements:

Users should be able to register and log in to the application securely. (User Management)

Users should be able to upload their resumes for analysis. (Resume Upload)

Users should be able to receive detailed and personalized analysis of their resumes in real-time. (Real-time Analysis)

Users should be able to receive personalized recommendations for skill enhancement and industry alignment. (Personalized Recommendations)

System Requirements:

The application should function seamlessly on various web browsers with varying specifications.

The application should maintain responsiveness and performance even with a high volume of data.

The application should utilize secure communication protocols (e.g., end-to-end encryption) to protect user data and messages.

The application should be scalable to accommodate a growing user base (if anticipated).

2. Non-Functional Requirements:

Usability:

The application interface should be intuitive, user-friendly, and easy to navigate. This includes clear labeling of features, logical layout, and efficient interaction patterns.

Performance:

The application should load data quickly, respond promptly to user actions, and maintain smooth operation even on resource-constrained devices.

Reliability:

The application should function reliably and minimize crashes or unexpected behavior.

Security:

The application should prioritize user data and message security. This includes secure user authentication, data encryption in transit and at rest, and protection against vulnerabilities.

Maintainability:

The application's code should be well-documented, organized, and modular to facilitate future maintenance and updates.

3. Techniques for Gathering Requirements:

User Interviews:

Conduct interviews with potential users to understand their job application needs, preferences, and pain points with existing resume analyzers.

Use Case Diagrams:

Create use case diagrams to illustrate the interaction between users and the application, outlining different user scenarios (e.g., registering, uploading a resume, receiving analysis).

Stakeholder Workshops:

Facilitate workshops with stakeholders (developers, potential investors) to gather their input and ensure the application aligns with their vision.

Competitor Analysis:

Analyze existing resume analyzers to understand their strengths, weaknesses, and identify opportunities for differentiation with CareerCrafter.

4. Documenting Requirements:

Create a Software Requirements Specification (SRS) document. This document will formally outline all the functional and non-functional requirements for CareerCrafter, ensuring a clear understanding for developers and future reference. The SRS should include:

Introduction:

Briefly describe the CareerCrafter application and its purpose.

Overall Description:

Explain the application's functionalities and target audience.

Specific Requirements:

Detail the functional and non-functional requirements using clear and concise language.

System Interfaces:

Define how CareerCrafter interacts with external systems (e.g., backend server, push notification service).

Quality Attributes:

Specify desired performance, security, and usability standards.

5. Verification and Validation:

Requirement Reviews:

Conduct reviews with stakeholders to ensure the documented requirements accurately reflect their needs and expectations for the CareerCrafter.

Prototyping:

Develop low-fidelity or high-fidelity prototypes to allow users to interact with a simulated version of the CareerCrafter and provide feedback on usability and functionality. This step is crucial in ensuring that the final product aligns with user expectations and provides a seamless and efficient user experience. It also allows for early detection and rectification of any potential issues or challenges.

3.1 → Technologies:

3.1.1 HTML

1. Service Setup and Configuration:

HTML (HyperText Markup Language) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. There's no need for a specific setup or configuration as HTML runs on any standard web browser.

2. Key Features:

HTML is easy to learn and use.

It is supported by all browsers.

It is the default language for creating web pages.

3. Architecture:

HTML follows a tag-based structure where each element on a webpage is created using tags. These tags are enclosed in angle brackets <> and usually come in pairs.

4. Advantages of using HTML:

HTML is free and requires no additional software to run.

It is supported by all modern browsers.

HTML is the foundation of web pages, and understanding HTML is beneficial for SEO and web development.

5. Potential Considerations:

HTML on its own cannot be used for creating dynamic and interactive websites. It needs to be used in conjunction with CSS and JavaScript.

HTML can become complex as the structure of the web pages increases.

6. Core Services:

HTML provides a wide range of elements for developing web pages, including tags for headings, paragraphs, links, images, lists, tables, forms, and many more.

7. Additional Services:

HTML5, the latest version of HTML, provides additional functionalities like video/audio playback, drawing graphics on screen (canvas), and support for local storage.

8. Development and Debugging:

HTML can be written in any text editor and run in any web browser. Browsers also provide developer tools for debugging and testing HTML code.

8. Monitoring and Analytics:

HTML doesn't directly provide monitoring or analytics features. However, it can be used with other tools like Google Analytics for tracking website activity.

3.1.2 CSS

1. Service Setup and Configuration:

CSS (Cascading Style Sheets) is a style sheet language used for describing the look and formatting of a document written in HTML. Like HTML, CSS code can be written in any text editor and run in any web browser, and there's no need for a specific setup or configuration.

2. Key Features:

CSS is used to control the style and layout of multiple web pages all at once.

CSS enables you to adjust the appearance of elements across the entire website instead of on a per-element basis.

With CSS, you have full control over the styling of a webpage, including layout, color, fonts, and more.

3. Architecture:

CSS follows a declarative paradigm, meaning you specify what you want to achieve (e.g., "I want the main heading to be red") and the browser figures out how to achieve it.

4. Advantages of using CSS:

CSS saves time: You can write CSS once and then reuse the same sheet in multiple HTML pages.

CSS offers greater consistency across different web pages of the same website.

With CSS, you have more control over the presentation of your webpage, including layout, color scheme, typography, and more.

5. Potential Considerations:

CSS has a steep learning curve, especially for complex layouts and responsive designs.

Cross-browser compatibility can be an issue, as different browsers interpret CSS in slightly different ways.

6. Core Services:

CSS provides a wide range of styling options, including color, background, font family, size, and weight, text alignment and decoration, margin, padding, border, and many more.

7. Additional Services:

CSS3, the latest version of CSS, provides additional functionalities like animations, gradients, transitions, and transformations.

8. Development and Debugging:

CSS can be written in any text editor and run in any web browser. Browsers also provide developer tools for debugging and testing CSS code.

8. Monitoring and Analytics:

CSS doesn't directly provide monitoring or analytics features. However, it can be used with other tools like Google Analytics for tracking website activity.

3.1.3 Python

1. Service Setup and Configuration:

Python is an interpreted, high-level, general-purpose programming language. It is easy to set up and configure. You can download and install Python from the official website. Python comes with pip (Python's package installer) which makes it easy to install additional libraries.

2. Key Features:

Python is known for its simplicity and readability which reduces the cost of program maintenance. It supports modules and packages, which encourages program modularity and code reuse. Python's standard library supports many Internet protocols, file formats, and data structures.

3. Architecture:

Python follows an object-oriented programming paradigm with dynamic semantics. It's simple, easy-to-learn syntax emphasizes readability and reduces the cost of program maintenance.

4. Advantages of using Python:

Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Python is often described as a "batteries included" language due to its comprehensive standard library.

5. Potential Considerations:

Python's execution speed may not be as fast as compiled languages like C++ or Java.

Python is not ideal for applications that require high memory consumption.

6. Core Services:

Python provides a wide range of libraries for various needs. For example, NumPy for numerical computations, Pandas for data manipulation, and Matplotlib for data visualization.

7. Additional Services:

Python offers various libraries for specific needs. For example, Scikit-learn for machine learning, Django and Flask for web development, and Pygame for game development.

8. Development and Debugging:

PyPython provides a robust set of development and debugging tools. For example, PyCharm and Jupyter Notebook for development, and pdb for debugging.

8. Monitoring and Analytics:

Python offers libraries such as logging and cProfile for monitoring and analytics.

3.1.4 MySQL

1. Service Setup and Configuration:

MySQL is a relational database management system. It is easy to set up and configure. You can download and install MySQL from the official website.

2. Key Features:

MySQL is known for its speed and reliability.

It supports a large subset of the SQL standard, allowing you to manage and manipulate your data using SQL queries.

3. Architecture:

MySQL follows a client-server architecture. The server runs on the machine where the databases are stored, and clients connect to the server to access the databases.

4. Advantages of using MySQL:

MySQL is open-source and free to use.

It provides high performance, scalability, and robust data protection.

5. Potential Considerations:

MySQL may not be as feature-rich as some other database management systems like PostgreSQL.

It may not be ideal for handling very large databases.

6. Core Services:

MySQL provides a wide range of services such as data security, scalability, and manageability.

7. Additional Services:

MySQL offers various additional services such as replication for improving performance and supporting backups, and partitioning for improving performance and management of large databases.

8. Development and Debugging:

MySQL provides a robust set of development and debugging tools. For example, MySQL Workbench for development, and Performance Schema for debugging.

8. Monitoring and Analytics:

MySQL offers services such as Performance Schema and Information Schema for monitoring and analytics.

9. Integration Steps:

MySQL can be easily integrated with various programming languages such as Python, PHP, and Java using their respective MySQL connectors.

3.1.5 Streamlit

1. Service Setup and Configuration:

Streamlit is an open-source Python library that allows you to create custom web apps for machine learning and data science. To set up and configure Streamlit, you need to install it via pip (`pip install streamlit`) and run it using the command `streamlit run your_script.py`.

2. Key Features:

Streamlit allows you to turn data scripts into shareable web apps in just a few minutes. It includes several built-in commands for visualizing data, including charts, maps, and more. Streamlit supports hot-reloading, so your app updates live as you edit and save your file.

3. Architecture:

Streamlit follows a script execution model, meaning it reruns your entire Python script from top to bottom with every user interaction.

4. Advantages of using Streamlit:

Streamlit is easy to use and requires minimal web development skills. It allows for rapid prototyping and iteration of data apps. Streamlit has a strong and growing community, providing a wealth of resources and community solutions.

5. Potential Considerations:

Since Streamlit reruns the entire script from top to bottom with each interaction, it might not be suitable for applications with heavy computations or large datasets. Streamlit's customization options for layout and styling are somewhat limited compared to full-fledged web development frameworks.

6. Core Services:

Streamlit provides a wide range of features for developing data apps, including data caching, widgets for user input (like sliders and checkboxes), and utilities for displaying output (like tables and charts).

7. Additional Services:

Streamlit offers additional functionalities like theme customization, layout options (like columns and expanders), and session state for maintaining information across reruns.

8. Development and Debugging:

Streamlit can be written in any text editor and run in any web browser. It also provides debugging information in the browser when exceptions occur.

8. Monitoring and Analytics:

Streamlit doesn't directly provide monitoring or analytics features. However, you can use Python's logging library for logging events and errors.

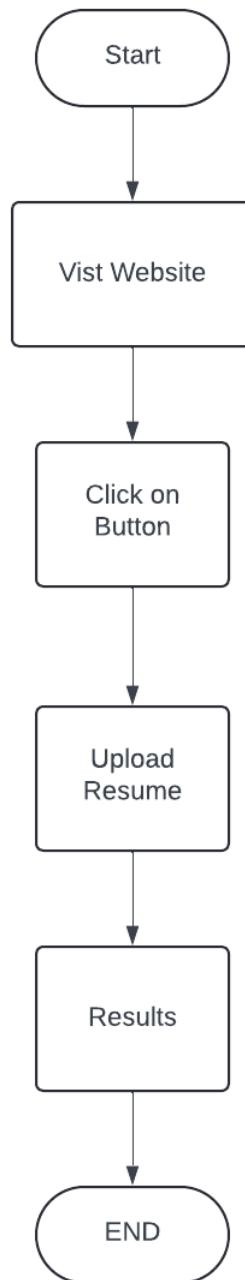
9. Integration Steps:

Streamlit can be easily integrated with other Python libraries for data analysis (like pandas and NumPy), machine learning (like scikit-learn and TensorFlow), and data visualization (like Matplotlib and Plotly).

CHAPTER - 4

DESIGN

4.1 Data Flow Diagram (DFD)



CHAPTER – 5

IMPLEMENTATION

5.1 Front-End Implementation:

The front-end of the application is built using Streamlit. Key steps in the front-end implementation include:

Setting up the Streamlit development environment: This involves installing Streamlit via pip and running it using the command `streamlit run your_script.py`.

Creating the Streamlit project: Initialize a new Streamlit project and structure your codebase following best practices (e.g., separating UI code from business logic).

Building the user interface: Use Streamlit's built-in commands to create interactive widgets and display data.

5.2 Back-End Implementation:

The back-end of the application is built using Python and MySQL. Key steps in the back-end implementation include:

Setting up the Python and MySQL development environment: This involves installing Python and MySQL, and setting up the database.

Creating the Python project: Initialize a new Python project and structure your codebase following best practices (e.g., separating data processing code from application logic).

Building the data processing functionality: Use Python's built-in functions and additional libraries like NLTK and spaCy for resume parsing and analysis.

5.3 Integration of Front-End and Back-End:

The front-end and back-end of the application are integrated using Python. Key steps in the integration include:

Setting up communication between the front-end and back-end: This involves creating Python functions that the Streamlit app can call to perform data processing tasks.

Testing the integrated system: Ensure that the front-end and back-end work together as expected, and that the user interface correctly displays the results of the data processing tasks.

5.4 Testing:

The application is thoroughly tested to ensure it works as expected. This includes:

Unit testing: Test individual components of the application to ensure they work correctly in isolation.

Integration testing: Test the application as a whole to ensure the components work correctly together.

User testing: Have users test the application and provide feedback on its functionality and usability

5.5 Post Implementation and Maintenance:

Deployment:

The application is deployed to a suitable environment where it can be accessed by users. This could be a web server for a web application, or an app store for a mobile application. The deployment process involves packaging the application, setting up the necessary server infrastructure, and configuring any necessary settings.

Monitoring:

Once the application is deployed, it's important to monitor its performance and usage. This can involve tracking metrics like user engagement, error rates, and resource usage. Tools like Google Analytics can be used for monitoring web applications, while various other tools are available for monitoring mobile applications.

User Feedback:

Collecting and analyzing user feedback is a crucial part of the post-implementation phase. This feedback can provide valuable insights into how the application is being used, what features are popular, and where users are encountering problems. Feedback can be collected through user surveys, feedback forms, or by analyzing user behavior within the application.

Updates and Improvements:

Based on the insights gained from monitoring and user feedback, updates and improvements can be made to the application. This could involve fixing bugs, improving performance, or adding new features. It's important to prioritize these updates based on their impact and feasibility.

Regular Maintenance:

Regular maintenance is necessary to ensure the ongoing performance and security of the application. This can involve tasks like updating libraries and frameworks, optimizing databases, and performing security audits.

CHAPTER – 6

SYSTEM TESTING

6.1 Testing Strategies

System testing is a level of software testing where a complete and integrated software system is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. Here's an overview of the key aspects involved in system testing:

1. Test Plan:

A test plan is created outlining the testing strategy, scope, schedule, and resources required for the system testing. The test plan also includes the test cases to be executed and the expected results.

2. Test Case Execution:

Test cases are executed on the integrated system to validate that it works as expected. This involves running the system with various inputs and comparing the actual results with the expected results.

4. Functional Testing:

Functional testing involves testing the system's functionality to ensure it behaves as expected. This includes testing all the functional requirements outlined in the system specification.

5. Non-Functional Testing:

Non-functional testing involves testing aspects of the system that don't relate to specific user actions or system functionality. This includes performance testing, security testing, usability testing, and more.

6. Regression Testing:

Regression testing is performed after modifications (like bug fixes or enhancements) are made to the system, to ensure that these changes haven't negatively impacted existing functionality.

6. User Acceptance Testing:

User acceptance testing (UAT) is the final phase of system testing, where the system is tested by the intended users. The goal of UAT is to validate that the system is ready for release and will meet the users' needs.

6. Issue Tracking:

Any issues or defects discovered during system testing are tracked and managed using an issue tracking system. These issues are prioritized and fixed, and the affected areas of the system are re-tested to ensure the fixes are successful.

6. Test Report:

After the system testing phase is complete, a test report is created summarizing the testing activities, the number and severity of defects found, the test coverage, and the status of the test cases.

CHAPTER – 7

CONCLUSION

7.1. Current State of the Project - A Technical Perspective

As of now, the project has made significant progress from both a development and implementation standpoint. Here's a brief overview:

Front-End Development: The user interface of the application has been successfully built using Streamlit. It is interactive, user-friendly, and provides real-time updates.

Back-End Development: The back-end of the application, built using Python and MySQL, is functioning as expected. It effectively handles data processing tasks and communicates seamlessly with the front-end.

Integration: The front-end and back-end of the application have been successfully integrated. The system is able to take user input, process it, and display the results in an efficient manner.

Testing: Preliminary testing has been conducted to ensure that the application works as expected. The application has passed all the defined test cases.

Deployment: The application has been deployed and is accessible to users. Initial user feedback has been positive.

Maintenance and Updates: Regular maintenance is being carried out to ensure the smooth functioning of the application. Necessary updates and improvements are being made based on user feedback and monitoring data.

Moving forward, the focus will be on enhancing the application based on user feedback, improving performance, and adding new features. Regular system testing will be conducted to ensure the reliability and efficiency of the application. The ultimate goal is to provide a robust and efficient application that meets the needs of the users and provides a seamless user experience.

7.2. Remaining Areas of Concern - Addressing Potential Challenges

Technical Challenges: This could involve dealing with complex algorithms, integrating different technologies, or optimizing performance. It's important to identify these challenges early and plan for them, possibly by learning new skills, using different tools or libraries, or getting help from others with more experience in those areas.

Resource Constraints: This could be in terms of time, budget, or human resources. If the project timeline is tight, it might be challenging to complete all tasks as planned. If the budget is limited, you might need to find cost-effective solutions. If there's a shortage of team members or skills, it might be necessary to recruit more people or provide training for existing members.

Scope Creep: This refers to the project scope gradually expanding beyond its original objectives. It's a common challenge in many projects and can lead to delays and increased costs. To address this, it's crucial to clearly define the project scope from the beginning and stick to it. Any changes to the scope should be carefully considered and managed.

Communication Issues: Effective communication is key to the success of any project. Challenges can arise if team members are not clear about their tasks, if there's a lack of communication among team members, or if stakeholders are not kept informed about the project's progress. To address these challenges, establish clear

communication channels and regular updates.

Risk Management: Every project comes with risks, such as unexpected technical issues, changes in market trends, or team members leaving. It's important to identify potential risks early on and have a plan to mitigate them.

7.3. Technical and Managerial Lessons Learned - Key Takeaways

Technical Lessons:

Importance of Robust Design: A well-thought-out design can significantly ease the development process and result in a more robust and maintainable application.

Choosing the Right Tools: The selection of programming languages, libraries, and frameworks can greatly impact the efficiency and performance of the application.

Testing is Crucial: Regular and thorough testing is essential to catch bugs early and ensure the application works as expected.

Code Quality Matters: Writing clean, well-documented code makes it easier to maintain and update the application.

Stay Updated: Technology evolves rapidly. Keeping up-to-date with the latest technologies and trends can provide new ways to solve problems and improve the application.

Managerial Lessons:

Effective Communication: Clear and regular communication among team members and stakeholders is key to ensuring everyone is aligned and working towards the same goals.

Project Planning: Detailed project planning helps in setting realistic expectations, managing resources effectively, and staying on track.

Risk Management: Identifying potential risks early and having a plan to mitigate them can prevent many issues down the line.

User-Centric Approach: Keeping the end-user in mind during the development process helps in building an application that meets user needs and provides a good user experience.

Continuous Improvement: Post-deployment, the work doesn't stop. Collecting user feedback and making continuous improvements is crucial for the success of the application.

7.4. Future Scope - Exploring the Potential

Expansion of Analysis Metrics: In the future, CareerCrafter could expand its analysis metrics to include more factors such as industry-specific keywords, role-specific skills, and more. This would provide a more comprehensive evaluation of the user's resume.

Integration with Job Portals: CareerCrafter could be integrated with various job portals. This would allow users to directly apply the feedback from CareerCrafter to their online job applications.

Personalized Learning Recommendations: Based on the skill gaps identified in the resume analysis, CareerCrafter could provide personalized learning recommendations. This could include suggesting online courses, books, or other resources to help the user improve their skills.

Career Path Prediction: Using machine learning algorithms, CareerCrafter could predict potential career paths for users based on their skills and interests. This would provide users with a clearer direction for their professional development.

Multilingual Support: To cater to a global audience, CareerCrafter could add support for multiple languages. This would make the tool accessible to non-English speaking users.

Interview Preparation: Beyond resume analysis, CareerCrafter could also provide assistance with interview preparation. This could include providing common interview questions, tips for answering them, and even mock interview sessions.

Networking Opportunities: CareerCrafter could also facilitate networking opportunities by connecting users with professionals in their desired industry. This could provide users with valuable insights and potentially lead to job opportunities.

CHAPTER 8

DEMONSTRATION/SCREENSHOTS/CODES

SOURCE CODE : --

App.py :-

```
import streamlit as st
import pandas as pd
import base64,random
import time,datetime
from pyresparser import ResumeParser
from pdfminer3.layout import LAParams, LTTextBox
from pdfminer3.pdfpage import PDFPage
from pdfminer3.pdfinterp import PDFResourceManager
from pdfminer3.pdfinterp import PDFPageInterpreter
from pdfminer3.converter import TextConverter
import io,random
from streamlit_tags import st_tags
from PIL import Image
import pymysql
from Courses import ds_course,web_course,android_course,ios_course,uiux_course
import nltk
nltk.download('stopwords')

def get_table_download_link(df,filename,text):
    csv = df.to_csv(index=False)
    b64 = base64.b64encode(csv.encode()).decode()
    href = f'{text}</a>'
    return href

def pdf\_reader\(file\):
    resource\_manager = PDFResourceManager\(\)
    fake\_file\_handle = io.StringIO\(\)
    converter = TextConverter\(resource\_manager, fake\_file\_handle, laparams=LAParams\(\)\)
    page\_interpreter = PDFPageInterpreter\(resource\_manager, converter\)
    with open\(file, 'rb'\) as fh:
        for page in PDFPage.get\_pages\(fh,
                                      caching=True,
                                      check\_extractable=True\):
            page\_interpreter.process\_page\(page\)
            print\(page\)
    text = fake\_file\_handle.getvalue\(\)

    converter.close\(\)
    fake\_file\_handle.close\(\)
    return text
```

```

def show_pdf(file_path):
    with open(file_path, "rb") as f:
        base64_pdf = base64.b64encode(f.read()).decode('utf-8')
    pdf_display = F'<iframe src="data:application/pdf;base64,{base64_pdf}" width="700" height="1000" type="application/pdf"></iframe>'
    st.markdown(pdf_display, unsafe_allow_html=True)

def course_recommender(course_list):
    st.subheader("**Courses & Certificates Recommendations 🎓**")
    c = 0
    rec_course = []
    no_of_reco = st.slider('Choose Number of Course Recommendations:', 1, 10, 5)
    random.shuffle(course_list)
    for c_name, c_link in course_list:
        c += 1
        st.markdown(f"({c}) [{c_name}]({c_link})")
        rec_course.append(c_name)
        if c == no_of_reco:
            break
    return rec_course

connection =
pymysql.connect(host='localhost', user='root', password='8755', db='resume_analyzer')
cursor = connection.cursor()

def
insert_data(name,email,res_score,timestamp,no_of_pages,reco_field,cand_level,skills,recommended_skills,courses):
    DB_table_name = 'user_data'
    insert_sql = "insert into " + DB_table_name + """
    values (0,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"""
    rec_values = (name, email, str(res_score), timestamp,str(no_of_pages), reco_field,
    cand_level, skills,recommended_skills,courses)
    cursor.execute(insert_sql, rec_values)
    connection.commit()

st.set_page_config(
    page_title="AI Resume Analyzer",
    page_icon="resume_analyzer\Logo\logo2.png",
)
def run():
    img = Image.open("resume_analyzer\Logo\resume.png")
    # img = img.resize((370,300))
    st.image(img)
    st.title("AI Resume Analyser")
    st.sidebar.markdown("# Choose User")
    activities = ["User", "Admin"]
    choice = st.sidebar.selectbox("Choose among the given options:", activities)
    link = '[@Developed by Devyansh Rajput](https://www.devyanshrajput.live)'

```

```

st.sidebar.markdown(link, unsafe_allow_html=True)

db_sql = """CREATE DATABASE IF NOT EXISTS resume_analyzer;"""
cursor.execute(db_sql)

DB_table_name = 'user_data'
table_sql = "CREATE TABLE IF NOT EXISTS " + DB_table_name + """
    (ID INT NOT NULL AUTO_INCREMENT,
     Name varchar(500) NOT NULL,
     Email_ID VARCHAR(500) NOT NULL,
     resume_score VARCHAR(8) NOT NULL,
     Timestamp VARCHAR(50) NOT NULL,
     Page_no VARCHAR(5) NOT NULL,
     Predicted_Field BLOB NOT NULL,
     User_level BLOB NOT NULL,
     Actual_skills BLOB NOT NULL,
     Recommended_skills BLOB NOT NULL,
     Recommended_courses BLOB NOT NULL,
     PRIMARY KEY (ID));
"""

cursor.execute(table_sql)
if choice == 'User':
    st.markdown('''<h5 style='text-align: left; color: #FFD700;'> Upload your resume, and
get smart recommendations</h5>''',
                unsafe_allow_html=True)
pdf_file = st.file_uploader("Choose your Resume", type=["pdf"])
if pdf_file is not None:
    with st.spinner('Uploading your Resume...'):
        time.sleep(4)
    save_resume_path = 'resume_analyzer/Uploaded_Resumes/' + pdf_file.name
    with open(save_resume_path, "wb") as f:
        f.write(pdf_file.getbuffer())
    show_pdf(save_resume_path)
    resume_data = ResumeParser(save_resume_path).get_extracted_data()
    if resume_data:
        resume_text = pdf_reader(save_resume_path)

    st.header("**Resume Analysis**")
    st.success("Hello " + resume_data['name'])
    st.subheader("**Your Basic info**")
    try:
        st.text('Name: ' + resume_data['name'])
        st.text('Email: ' + resume_data['email'])
        st.text('Contact: ' + resume_data['mobile_number'])
        st.text('Resume pages: ' + str(resume_data['no_of_pages']))
    except:
        pass
    cand_level = ''
    if resume_data['no_of_pages'] == 1:
        cand_level = "Fresher"

```

```

        st.markdown( '''<h4 style='text-align: left; color: #d73b5c;'>You are at
Fresher level!</h4>''',unsafe_allow_html=True)
    elif resume_data['no_of_pages'] == 2:
        cand_level = "Intermediate"
        st.markdown('''<h4 style='text-align: left; color: #1ed760;'>You are at
intermediate level!</h4>''',unsafe_allow_html=True)
    elif resume_data['no_of_pages'] >=3:
        cand_level = "Experienced"
        st.markdown('''<h4 style='text-align: left; color: #fba171;'>You are at
experience level!''',unsafe_allow_html=True)

# st.subheader("**Skills Recommendation 🤔 **")
## Skill shows
keywords = st_tags(label='### Your Current Skills',
text='See our skills recommendation below',
value=resume_data['skills'],key = '1 ')
## keywords
ds_keyword = ['tensorflow','keras','pytorch','machine learning','deep
Learning','flask','streamlit']
web_keyword = ['react', 'django', 'node js', 'react js', 'php', 'laravel',
'magento', 'wordpress',
'javascript', 'angular js', 'c#', 'flask']
android_keyword = ['android','android
development','flutter','kotlin','xml','kivy']
ios_keyword = ['ios','ios development','swift','cocoa','cocoa touch','xcode']
uiux_keyword = ['ux','adobe
xd','figma','zeplin','balsamiq','ui','prototyping','wireframes','storyframes','adobe
photoshop','photoshop','editing','adobe illustrator','illustrator','adobe after
effects','after effects','adobe premier pro','premier pro','adobe
indesign','indesign','wireframe','solid','grasp','user research','user experience']

recommended_skills = []
reco_field = ''
rec_course = ''
## Courses recommendation
for i in resume_data['skills']:
    ## Data science recommendation
    if i.lower() in ds_keyword:
        print(i.lower())
        reco_field = 'Data Science'
        st.success("** Our analysis says you are looking for Data Science
Jobs.**")
        recommended_skills = ['Data Visualization','Predictive
Analysis','Statistical Modeling','Data Mining','Clustering & Classification','Data
Analytics','Quantitative Analysis','Web Scraping','ML
Algorithms','Keras','Pytorch','Probability','Scikit-learn','Tensorflow','Flask','Streamlit']
        recommended_keywords = st_tags(label='### Recommended skills for
you.',
```

```
        text='Recommended skills generated from
System',value=recommended_skills,key = '2')
        st.markdown('''<h4 style='text-align: left; color: #1ed760;'>Adding
this skills to resume will boost🔗 the chances of getting a
Job</h4>''',unsafe_allow_html=True)
        rec_course = course_recommender(ds_course)
        break

## Web development recommendation
elif i.lower() in web_keyword:
    print(i.lower())
    reco_field = 'Web Development'
    st.success("** Our analysis says you are looking for Web Development
Jobs **")
    recommended_skills = ['React','Django','Node JS','React
JS','php','laravel','Magento','wordpress','Javascript','Angular JS','c#','Flask','SDK']
    recommended_keywords = st_tags(label='### Recommended skills for
you.',

        text='Recommended skills generated from
System',value=recommended_skills,key = '3')
        st.markdown('''<h4 style='text-align: left; color: #1ed760;'>Adding
this skills to resume will boost🔗 the chances of getting a
Job🔗</h4>''',unsafe_allow_html=True)
        rec_course = course_recommender(web_course)
        break

## Android App Development
elif i.lower() in android_keyword:
    print(i.lower())
    reco_field = 'Android Development'
    st.success("** Our analysis says you are looking for Android App
Development Jobs **")
    recommended_skills = ['Android','Android
development','Flutter','Kotlin','XML','Java','Kivy','GIT','SDK','SQLite']
    recommended_keywords = st_tags(label='### Recommended skills for
you.',

        text='Recommended skills generated from
System',value=recommended_skills,key = '4')
        st.markdown('''<h4 style='text-align: left; color: #1ed760;'>Adding
this skills to resume will boost🔗 the chances of getting a
Job🔗</h4>''',unsafe_allow_html=True)
        rec_course = course_recommender(android_course)
        break

## IOS App Development
elif i.lower() in ios_keyword:
    print(i.lower())
    reco_field = 'IOS Development'
    st.success("** Our analysis says you are looking for IOS App
Development Jobs **")
```

```

        recommended_skills = ['IOS','IOS Development','Swift','Cocoa','Cocoa
Touch','Xcode','Objective-C','SQLite','Plist','StoreKit',"UI-Kit",'AV Foundation','Auto-
Layout']
        recommended_keywords = st_tags(label='### Recommended skills for
you.',

            text='Recommended skills generated from
System',value=recommended_skills,key = '5')
            st.markdown('''<h4 style='text-align: left; color: #1ed760;'>Adding
this skills to resume will boost ↗ the chances of getting a
Job ↗</h4>''',unsafe_allow_html=True)
            rec_course = course_recommender(ios_course)
            break

## Ui-UX Recommendation
elif i.lower() in uiux_keyword:
    print(i.lower())
    reco_field = 'UI-UX Development'
    st.success("** Our analysis says you are looking for UI-UX Development
Jobs **")
    recommended_skills = ['UI','User Experience','Adobe
XD','Figma','Zeplin','Balsamiq','Prototyping','Wireframes','Storyframes','Adobe
Photoshop','Editing','Illustrator','After Effects','Premier
Pro','InDesign','Wireframe','Solid','Grasp','User Research']
    recommended_keywords = st_tags(label='### Recommended skills for
you.',

            text='Recommended skills generated from
System',value=recommended_skills,key = '6')
            st.markdown('''<h4 style='text-align: left; color: #1ed760;'>Adding
this skills to resume will boost ↗ the chances of getting a
Job ↗</h4>''',unsafe_allow_html=True)
            rec_course = course_recommender(uiux_course)
            break

## Insert into table
ts = time.time()
cur_date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
cur_time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
timestamp = str(cur_date+'_'+cur_time)

### Resume writing recommendation
st.subheader("**Resume Tips & Ideas ☰ **")
resume_score = 0
if 'Objective' in resume_text:
    resume_score = resume_score+20
    st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome!
You have added Objective</h5>''',unsafe_allow_html=True)
else:

```

```
        st.markdown(''''<h5 style='text-align: left; color: #ff0000;'>[-] Please  
add your career objective, it will give your career intension to the  
Recruiters.</h4>''',unsafe_allow_html=True)

    if 'Declaration' in resume_text:  
        resume_score = resume_score + 20  
        st.markdown(''''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome!  
You have added Delcaration/h4>''',unsafe_allow_html=True)  
    else:  
        st.markdown(''''<h5 style='text-align: left; color: #ff0000;'>[-] Please  
add Declaration. It will give the assurance that everything written on your resume is true and  
fully acknowledged by you</h4>''',unsafe_allow_html=True)

    if 'Hobbies' or 'Interests'in resume_text:  
        resume_score = resume_score + 20  
        st.markdown(''''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome!  
You have added your Hobbies</h4>''',unsafe_allow_html=True)  
    else:  
        st.markdown(''''<h5 style='text-align: left; color: #ff0000;'>[-] Please  
add Hobbies. It will show your personality to the Recruiters and give the assurance that you  
are fit for this role or not.</h4>''',unsafe_allow_html=True)

    if 'Achievements' in resume_text:  
        resume_score = resume_score + 20  
        st.markdown(''''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome!  
You have added your Achievements </h4>''',unsafe_allow_html=True)  
    else:  
        st.markdown(''''<h5 style='text-align: left; color: #ff0000;'>[-] Please  
add Achievements. It will show that you are capable for the required  
position.</h4>''',unsafe_allow_html=True)

    if 'Projects' in resume_text:  
        resume_score = resume_score + 20  
        st.markdown(''''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome!  
You have added your Projects</h4>''',unsafe_allow_html=True)  
    else:  
        st.markdown(''''<h5 style='text-align: left; color: #ff0000;'>[-] Please  
add Projects. It will show that you have done work related the required position or  
not.</h4>''',unsafe_allow_html=True)

    st.subheader("**Resume Score  **")
    st.markdown(
        """
<style>
    .stProgress > div > div > div > div {
        background-color: #d73b5c;
    }
</style>""",
        unsafe_allow_html=True,
    )
)
```

```

my_bar = st.progress(0)
score = 0
for percent_complete in range(resume_score):
    score +=1
    time.sleep(0.1)
    my_bar.progress(percent_complete + 1)
    st.success('** Your Resume Writing Score: ' + str(score) +'**')
    st.warning("** Note: This score is calculated based on the content that you
have in your Resume. **")
    st.balloons()

    insert_data(resume_data['name'], resume_data['email'], str(resume_score),
timestamp,
                           str(resume_data['no_of_pages']), reco_field, cand_level,
str(resume_data['skills']),
                           str(recommended_skills), str(rec_course))

    connection.commit()
else:
    st.error('Something went wrong..')
else:
## Admin Side
st.success('Welcome to Admin Side')

ad_user = st.text_input("Username")
ad_password = st.text_input("Password", type='password')
if st.button('Login'):
    if ad_user == 'devyansh' and ad_password == 'dell8755':
        st.success("Welcome Devyansh !")
        # Display Data
        cursor.execute('''SELECT*FROM user_data''')
        data = cursor.fetchall()
        st.header("User's Data")
        df = pd.DataFrame(data, columns=['ID', 'Name', 'Email', 'Resume Score',
'Timestamp', 'Total Page',
                           'Predicted Field', 'User Level', 'Actual
Skills', 'Recommended Skills',
                           'Recommended Course'])
        st.dataframe(df)
        st.markdown(get_table_download_link(df, 'User_Data.csv', 'Download Report'),
unsafe_allow_html=True)
    else:
        st.error("Wrong ID & Password Provided")
run()

```

Courses.py :-

```
ds_course = [['Machine Learning Crash Course by Google [Free]',  
'https://developers.google.com/machine-learning/crash-course'],  
['Machine Learning A-Z by  
Udemy', 'https://www.udemy.com/course/machinelearning/'],  
['Machine Learning by Andrew NG', 'https://www.coursera.org/learn/machine-  
learning'],  
['Data Scientist Master Program of Simplilearn  
(IBM)', 'https://www.simplilearn.com/big-data-and-analytics/senior-data-scientist-masters-  
program-training'],  
['Data Science Foundations: Fundamentals by  
LinkedIn', 'https://www.linkedin.com/learning/data-science-foundations-fundamentals-5'],  
['Data Scientist with Python', 'https://www.datacamp.com/tracks/data-scientist-  
with-python'],  
['Programming for Data Science with  
Python', 'https://www.udacity.com/course/programming-for-data-science-nanodegree--nd104'],  
['Programming for Data Science with  
R', 'https://www.udacity.com/course/programming-for-data-science-nanodegree-with-R--nd118'],  
['Introduction to Data Science', 'https://www.udacity.com/course/introduction-to-  
data-science--cd0017'],  
['Intro to Machine Learning with  
TensorFlow', 'https://www.udacity.com/course/intro-to-machine-learning-with-tensorflow-  
nanodegree--nd230']]  
  
web_course = [['Django Crash course [Free]', 'https://youtu.be/e1IyzVyrLSU'],  
['Python and Django Full Stack Web Developer  
Bootcamp', 'https://www.udemy.com/course/python-and-django-full-stack-web-developer-bootcamp'],  
['React Crash Course [Free]', 'https://youtu.be/Dorf8i6lCuk'],  
['ReactJS Project Development  
Training', 'https://www.dotnettricks.com/training/masters-program/reactjs-certification-  
training'],  
['Full Stack Web Developer - MEAN Stack', 'https://www.simplilearn.com/full-  
stack-web-developer-mean-stack-certification-training'],  
['Node.js and Express.js [Free]', 'https://youtu.be/0e421EPjeBE'],  
['Flask: Develop Web Applications in  
Python', 'https://www.educative.io/courses/flask-develop-web-applications-in-python'],  
['Full Stack Web Developer by Udacity', 'https://www.udacity.com/course/full-  
stack-web-developer-nanodegree--nd0044'],  
['Front End Web Developer by Udacity', 'https://www.udacity.com/course/front-end-  
web-developer-nanodegree--nd0011'],  
['Become a React Developer by Udacity', 'https://www.udacity.com/course/react-  
nanodegree--nd019']]  
  
android_course = [['Android Development for Beginners [Free]', 'https://youtu.be/fis26HvvDII'],  
['Android App Development  
Specialization', 'https://www.coursera.org/specializations/android-app-development'],  
['Associate Android Developer  
Certification', 'https://grow.google/androiddev/#?modal_active=none']]
```

```
[ 'Become an Android Kotlin Developer by  
Udacity','https://www.udacity.com/course/android-kotlin-developer-nanodegree--nd940'],  
        ['Android Basics by Google','https://www.udacity.com/course/android-basics-  
nanodegree-by-google--nd803'],  
            ['The Complete Android Developer  
Course','https://www.udemy.com/course/complete-android-n-developer-course/'],  
                ['Building an Android App with Architecture  
Components','https://www.linkedin.com/learning/building-an-android-app-with-architecture-  
components'],  
                    ['Android App Development Masterclass using  
Kotlin','https://www.udemy.com/course/android-oreo-kotlin-app-masterclass/'],  
                        ['Flutter & Dart - The Complete Flutter App Development  
Course','https://www.udemy.com/course/flutter-dart-the-complete-flutter-app-development-  
course/'],  
                            ['Flutter App Development Course [Free]','https://youtu.be/rZLR5o1MR64']]  
  
ios_course = [[['IOS App Development by  
LinkedIn','https://www.linkedin.com/learning/subscription/topics/ios'],  
        ['iOS & Swift - The Complete iOS App Development  
Bootcamp','https://www.udemy.com/course/ios-13-app-development-bootcamp/'],  
            ['Become an iOS Developer','https://www.udacity.com/course/ios-developer-  
nanodegree--nd003'],  
                ['iOS App Development with Swift  
Specialization','https://www.coursera.org/specializations/app-development'],  
                    ['Mobile App Development with Swift','https://www.edx.org/professional-  
certificate/curtinx-mobile-app-development-with-swift'],  
                        ['Swift Course by  
LinkedIn','https://www.linkedin.com/learning/subscription/topics/swift-2'],  
                            ['Objective-C Crash Course for Swift  
Developers','https://www.udemy.com/course/objectivec/'],  
                                ['Learn Swift by Codcademy','https://www.codcademy.com/learn/learn-swift'],  
                                    ['Swift Tutorial - Full Course for Beginners [Free]','https://youtu.be/comQ1-  
x2a1Q'],  
                                        ['Learn Swift Fast - [Free]','https://youtu.be/FcsY1YPBwzQ']]  
uiux_course = [[['Google UX Design Professional  
Certificate','https://www.coursera.org/professional-certificates/google-ux-design'],  
        ['UI / UX Design Specialization','https://www.coursera.org/specializations/ui-  
ux-design'],  
            ['The Complete App Design Course - UX, UI and Design  
Thinking','https://www.udemy.com/course/the-complete-app-design-course-ux-and-ui-design/'],  
                ['UX & Web Design Master Course: Strategy, Design,  
Development','https://www.udemy.com/course/ux-web-design-master-course-strategy-design-  
development/'],  
                    ['The Complete App Design Course - UX, UI and Design  
Thinking','https://www.udemy.com/course/the-complete-app-design-course-ux-and-ui-design/'],  
                        ['DESIGN RULES: Principles + Practices for Great UI  
Design','https://www.udemy.com/course/design-rules/'],  
                            ['Become a UX Designer by Udacity','https://www.udacity.com/course/ux-designer-  
nanodegree--nd578']]
```

```

        ['Adobe XD Tutorial: User Experience Design Course
[Free]', 'https://youtu.be/68w2VwalD5w'],
        ['Adobe XD for Beginners [Free]', 'https://youtu.be/WEljsc2jorI'],
        ['Adobe XD in Simple Way', 'https://learnux.io/course/adobe-xd']]]

resume_videos = ['https://youtu.be/3agP4x8LYFM', 'https://youtu.be/fS_t3yS8v5s',
                 'https://youtu.be/aArb680BFPg', 'https://youtu.be/h-Nuv0eWWh0',
                 'https://youtu.be/BdQniERyw8I', 'https://youtu.be/Tt08KmFFIYQ',
                 'https://youtu.be/CLUsplI4xMU', 'https://youtu.be/bhwEsfXS6y8']

interview_videos = ['https://youtu.be/Tt08KmFFIYQ', 'https://youtu.be/KukmClH1KoA',
                     'https://youtu.be/7_aAicmPB3A', 'https://youtu.be/1mHjMNZZvFo',
                     'https://youtu.be/WfdtKbAJ0mE', 'https://youtu.be/wFbU185CvDU'
                     'https://youtu.be/wFbU185CvDU', 'https://youtu.be/TZ3C_syg90w']

```

HTML code:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
      href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600;700;800;900&display=swap" rel="stylesheet">
    <title>CarrerCrafter</title>
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Additional CSS Files -->
    <link rel="stylesheet" href="assets/css/fontawesome.css">
    <link rel="stylesheet" href="assets/css/templatemo-space-dynamic.css">
    <link rel="stylesheet" href="assets/css/animated.css">
    <link rel="stylesheet" href="assets/css/owl.css">
  </head>

  <body>
    <!-- ***** Preloader Start ***** -->
    <div id="js-preloader" class="js-preloader">
      <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
          <span></span>
          <span></span>
          <span></span>
        </div>
      </div>
    </div>
  </div>

```

```

<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->
<header class="header-area header-sticky wow slideInDown" data-wow-duration="0.75s" data-wow-delay="0s">
    <div class="container">
        <div class="row">
            <div class="col-12">
                <nav class="main-nav">
                    <!-- ***** Logo Start ***** -->
                    <a href="index.html" class="logo">
                        <h4>Carrer<span>Crafter</span></h4>
                    </a>
                    <!-- ***** Logo End ***** -->
                    <!-- ***** Menu Start ***** -->
                    <ul class="nav">
                        <li class="scroll-to-section"><a href="index.html" class="active">Home</a></li>
                        <li class="scroll-to-section"><a href="Our_Analyzer.html">Our Analyzer</a></li>
                    </ul>
                    <a class='menu-trigger'>
                        <span>Menu</span>
                    </a>
                    <!-- ***** Menu End ***** -->
                </nav>
            </div>
        </div>
    </div>
</header>
<!-- ***** Header Area End ***** -->

<div class="main-banner wow fadeIn" id="top" data-wow-duration="1s" data-wow-delay="0.5s">
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <div class="row">
                    <div class="col-lg-6 align-self-center">
                        <div class="left-content header-text wow fadeInLeft" data-wow-duration="1s" data-wow-delay="1s">
                            <h6>Welcome to CarrerCrafter</h6>
                            <h2>We assist in <em>Crafting</em> your <span>Career</span></h2>
                            <p>Surpass the ordinary and land your dream job with our tailored recommendations.</p>
                            <div class="main-blue-button"><a href="#contact">Analyze Resume Now</a></div>
                        </div>
                    </div>
                    <div class="col-lg-6">
                        <div class="right-image wow fadeInRight" data-wow-duration="1s" data-wow-delay="0.5s">
                            
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```
</div>
</div>
</div>
</div>
</div>

<div id="about" class="about-us section">
<div class="container">
<div class="row">
<div class="col-lg-4">
<div class="left-image wow fadeIn" data-wow-duration="1s" data-wow-delay="0.2s">

</div>
</div>
<div class="col-lg-8 align-self-center">
<div class="services">
<div class="row">
<div class="col-lg-6">
<div class="item wow fadeIn" data-wow-duration="1s" data-wow-delay="0.5s">
<div class="icon">

</div>
<div class="right-text">
<h4>Make A Resume That Wins</h4>
<p>Tell a story that engages recruiters and hiring managers</p>
</div>
</div>
</div>
<div class="col-lg-6">
<div class="item wow fadeIn" data-wow-duration="1s" data-wow-delay="0.7s">
<div class="icon">

</div>
<div class="right-text">
<h4>Boost Your Interview Chances</h4>
<p>Optimize your resume to land your best suited dream job</p>
</div>
</div>
</div>
<div class="col-lg-6">
<div class="item wow fadeIn" data-wow-duration="1s" data-wow-delay="0.9s">
<div class="icon">

</div>
<div class="right-text">
<h4>Rate Your Resume</h4>
<p>Is your resume a good match for what a recruiter is looking for</p>
</div>
</div>
```

```

        </div>
        <div class="col-lg-6">
            <div class="item wow fadeIn" data-wow-duration="1s" data-wow-delay="1.1s">
                <div class="icon">
                    
                </div>
                <div class="right-text">
                    <h4>See Your Missing Resume Skills</h4>
                    <p>Tailor your resume so that you highlight the skills recruiters are
searching for</p>
                </div>
            </div>
        </div>
        <div id="services" class="our-services section">
            <div class="container">
                <div class="row">
                    <div class="col-lg-6 align-self-center wow fadeInLeft" data-wow-duration="1s" data-
wow-delay="0.2s">
                        <div class="left-image">
                            
                        </div>
                    </div>
                    <div class="col-lg-6 wow fadeInRight" data-wow-duration="1s" data-wow-delay="0.2s">
                        <div class="section-heading">
                            <h2>Our Platform Instantly Gives You <em>Tailored Feedback</em> on your
<span>Resume</span></h2>
                            <ol>
                                <br>
                                <li>Most companies, including 99% of Fortune 500, use Applicant Tracking Systems
(ATS) to process your resume. These systems cause qualified candidates like you to slip
through the cracks.</li>
                                <br>
                                <li>CareerCrafter is a resume analyzer designed to transform the job application
process by providing personalized insights and recommendations to job seekers</li>
                                <br>
                                <li>Align with the market trend by analyzing skills and use recommendations for
skill enhancement</li>
                            </ol>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    
```

```
<div id="contact" class="contact-us section">
    <div class="container">
        <div class="row">
            <div class="col-lg-6 align-self-center wow fadeInLeft" data-wow-duration="0.5s" data-wow-delay="0.25s">
                <div class="section-heading">
                    <h2>Feel Free To Send Us a Feedback</h2>
                    <div class="phone-info">
                        <h4>For any enquiry, Mail Us @ rajputdevyansh9@gmail.com</h4>
                    </div>
                </div>
            </div>
            <div class="col-lg-6 wow fadeInRight" data-wow-duration="0.5s" data-wow-delay="0.25s">
                <form id="contact" action="" method="post">
                    <div class="row">
                        <div class="col-lg-6">
                            <fieldset>
                                <input type="name" name="name" id="name" placeholder="Name" autocomplete="on" required>
                            </fieldset>
                        </div>
                        <div class="col-lg-6">
                            <fieldset>
                                <input type="surname" name="surname" id="surname" placeholder="Surname" autocomplete="on" required>
                            </fieldset>
                        </div>
                    <div class="col-lg-12">
                        <fieldset>
                            <input type="text" name="email" id="email" pattern="[^@]*@[^@]*" placeholder="Your Email" required="">
                        </fieldset>
                    </div>
                    <div class="col-lg-12">
                        <fieldset>
                            <textarea name="message" type="text" class="form-control" id="message" placeholder="Message" required=""></textarea>
                        </fieldset>
                    </div>
                    <div class="col-lg-12">
                        <fieldset>
                            <button type="submit" id="form-submit" class="main-button ">Send Message</button>
                        </fieldset>
                    </div>
                </div>
                <div class="contact-dec">
                    
                </div>
            </div>
        </div>
    </div>
</div>
```

```
    </form>
  </div>
</div>
</div>
</div>
<!-- Scripts -->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="assets/js/owl-carousel.js"></script>
<script src="assets/js/animation.js"></script>
<script src="assets/js/imagesloaded.js"></script>
<script src="assets/js/templatemo-custom.js"></script>

</body>
</html>
```

SCREENSHOTS OF APPLICATION

The screenshot displays the Smart Resume Analyser application interface. At the top, a 'Choose User' dropdown is set to 'Normal User'. Below it, the main title 'Smart Resume Analyser' is shown with a logo featuring a brain and a magnifying glass.

The central area shows a resume for 'ROBERT SMITH' as a 'Data Scientist'. The resume details his work experience at 'ABC CORPORATION' from May 2010 to June 2013, where he performed Data Mining, Modeling, and Algorithm Development, implemented neural networks for classification, and used deep learning for image recognition. It also mentions his role as a 'Data Scientist' from 2005 to 2010, involving Data Mining, Machine Learning, and Statistical Analysis using Python, Java, and R.

Below the resume, the 'Resume Analysis' section provides a summary of basic info, skill levels, recommended skills, courses, and resume tips. It includes a 'Resume Score' bar and a note about the score being calculated based on resume content.

At the bottom, two bonus video links are displayed:

- Bonus Video for Resume Writing Tips**: A video titled 'HOW TO WRITE THE PERFECT RESUME FOR FRESHERS & EXPERIENCED' by Rohit Vatsa.
- Bonus Video for Interview Tips**: A video titled '11 Interview Questions And Answers For Freshers, With Tips, HR View & Common Mistakes' and another titled 'FRESHERS INTERVIEW सवाल और जवाब' by Rohit Vatsa.

Choose User

Choose among the given options:

Admin

Smart Resume Analyser



Welcome to Admin Side

Username
machine_learning_hub

Password

Login

Welcome Kushal

User's 🧑 Data

ID	Name	Email	Resume Score	Timestamp
32	33 Javascript Developer	info@qwikresume.com	40	2022-03-05_11:4
33	34 ROBERT SMITH	info@qwikresume.com	20	2022-03-05_12:1
34	35 ROBERT SMITH	info@qwikresume.com	20	2022-03-05_12:2
35	36 Android Developer	info@qwikresume.com	40	2022-03-05_10:2i
36	37 art director	hello@allisonbeer.com	20	2022-03-05_10:5:
37	38 art director	hello@allisonbeer.com	20	2022-03-05_10:5:
38	39 art director	hello@allisonbeer.com	20	2022-03-05_11:0
39	40 ROBERT SMITH	info@qwikresume.com	20	2022-03-06_03:1
40	41 Kushal Bhavar	kushalbhavar56@gmail...	80	2022-03-06_03:1
41	42 ROBERT SMITH	info@qwikresume.com	20	2022-03-06_03:1

[Download Report](#)

Pie-Chart for Predicted Field Recommendations

Predicted Field according to the Skills

Field	Percentage
Data Science	38.1%
Web Development	28.6%
UI-UX Development	23.8%
Android Development	4.7%
iOS Development	1.6%

Pie-Chart for User's 🧑 Experienced Level

Pie-Chart for User's 🧑 Experienced Level

Level	Percentage
Intermediate	61.9%
Fresher	38.1%

Bibliography

Official Python Resources:

Python Documentation: Start here for the core concepts and best practices:
<https://docs.python.org/3/reference/index.html>

MySQL Documentation: Firebase integration for real-time features: <https://dev.mysql.com/doc/>

Libraries & Packages:

Nltk: <https://www.nltk.org/>

Spacy: <https://spacy.io/>

En_core_web_sm: <https://spacy.io/models/en>

Streamlit: <https://streamlit.io/>

Tutorials and Blog Posts:

Building a Realtime App with Python and MySQL: Tutorials on Medium or developer blogs covering this topic.
Implementing Push Notifications in Python with MySQL: Guides specifically focused on push notifications.

Code Repositories:

GitHub: Resume Analyzer: Search " Resume Analyzer" on GitHub to find well-structured examples: [invalid URL removed]

Open-Source Apps: Analyze the codebase of existing open-source chat apps built with python for inspiration and learning.