RESUME BUILDER

by

Saurav Karki (T.U. Exam Roll No: 362/077)

Kushal Shrestha (T.U. Exam Roll No: 204/077)

Laxman Singh Dhami (T.U. Exam Roll No: 205/077)

A project submitted in partial fulfillment of the requirements for the Degree of Bachelor in Information Technology awarded by IOST, Tribhuvan University

Amrit Campus
Lekhnath Marg, Thamel
Nepal

Falgun, 2081

Student’s Declaration

We hereby declare that project report entitled “**Resume Builder**” submitted in the partial fulfilment of the requirement for Bachelor’s Degree in Information Technology of Tribhuvan University, is our original work and not submitted for the award of any other degree diploma, fellowship, or any other similar title or prize.

…………………………
Saurav Karki

…………………………
Kushal Shrestha

………………………..
Laxman Singh Dhami

Recommendation

This is to certify that this project entitled, “Resume Builder” Prepared and submitted by Saurav Karki , Kushal Shrestha and Laxman Singh Dhami in partial fulfillment of the requirements of the degree of Bachelors in Information Technology awarded by Tribhuvan University, has been completed under my supervision. I recommend the same for acceptance by Tribhuvan University.

Signature:
Name of the supervisor: Mr. Bhim Bahadur Rawat
Designation: Lecturer
Date signed:

Certificate

This project entitled "**Resume Builder**", prepared and submitted by Saurav Karki, Kushal Shrestha and Laxman Singh Dhami has been examined by us and is accepted for the award of the degree of Bachelor in Information Technology awarded by Tribhuvan University.

External Examiner ……………..… ……….……

Asst. Prof. Dhirendra Kumar Yadav …..…..……... ……….…..
Project Coordinator
Department of CSIT

Mr. Bhim Bahadur Rawat ……………… ……………
Supervisor
Lecturer
Department of CSIT

Acknowledgement

We would like to express our deepest appreciation and gratitude to all those people who helped and guide us throughout the planning, development and final outcome of this project. Completion of this project is only made possible by those guidance and help. We express our gratitude and deep regards to our supervisor, **Mr. Bhim Bahadur Rawat** for his guidance, monitoring and providing constant encouragement and correcting us throughout the completion of this project. We would also like to thank our college "**Amrit Campus**" for providing supervision and for supporting hands academically as well as with the work space.

We also thank respected teachers for supporting us in learning and understanding different algorithms and technologies required to complete this project. Our classmate has also helped a lot in sharing their view and thoughts about the project and its use case.

We are also thankful for various teaching and not teaching staffs and seniors for their help in resource and encouragement which helped in the completion of this project.

We also thank various people for providing essentials data and resources for our project that has helped to make our project optimal.

Finally, other mention goes to our friends and family as well as other people who has helped directly or indirectly during this project. Without those guidance and encouragement we won't have able to complete this project. During this project we get to learn and understand many concepts and technology which will be very helpful in actual professional market.

Saurav Karki (T.U. Exam Roll No: 362/077)

Kushal Shrestha (T.U. Exam Roll No: 204/077)

Laxman Singh Dhami (T.U. Exam Roll No: 205/077)

Abstract

In today’s competitive job market, building a professional and appealing resume is very important. People always find difficulty in creating and formatting their resume in a professional way. To ease this difficulty and helping people to create simple but professional resumes we have built this resume builder project. This project helps people to solve their difficulty in creating professional resume by providing user-friendly web application interface. This platform allow users to input their personal and professional details through simple form fields, which are then dynamically transformed into a well-structured and professional resume format. Furthermore this website offers font customization, resume cloning, sharing options, Print and Download as pdf functionality. Here the project also includes objective suggestions algorithm based on the user’s job role i.e. Resume Title which further improves and ease the process of resume building. The main idea behind this project to help normal users to create a professional resume with a simple UI and less effort.

Keywords: Objective Suggestions Algorithm, Resume Builder, User-Friendly Interface, Resume Cloning, Font Customization

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AJAX: Asynchronous JavaScript and XML (Extensible Markup Language)

CSS: Cascading Style Sheet

HTML: Hypertext Markup Language

JS: JavaScript

PHP: Hypertext Preprocessor

ER: Entity Relationship

DFD: Dataflow Diagram

UI: User Interface

CRUD: Create Read Update and Delete

NLP: Natural Language Processing

AI: Artificial Intelligence

ML: Machine Learning

API: Application Programming Interface

JSON: JavaScript Object Notation

SQL: Structured Query Language

Chapter 1

Introduction

1.1. Introduction

Building a resume is the beginning of presenting yourself to the professional job world. It is the first step in a job application process and has a great significance. A resume is like a personal marketing tool containing all information about an employee such as skills, experience, education, qualification and achievements in a well-structured format. It is the key document that employers use to study about the candidate's qualifications for a role. A well-formatted resume can help you stand out among other competitors in a competitive job market. This will increase your chance of getting job opportunities.

Despite having great significance many individuals find it challenging to create a resume that is well formatted and showing their worth for the role. The challenges is due to lack of design, technical skills as well as presentation skills. In the world of these challenges the goal of this web based project named "**Resume Builder**" is to help users in creating a professional looking resume with less time and effort. This web application provides simple UI to users for entering their personal details, education, skills and work experience, and then converts those details in a well-structured professional-looking resume that is ready to be download and shared with potential employers.

Features of Resume Builder:

1. Allow users to perform CRUD (Create, Read, Update, and Delete) operations on the resume.
2. Allow users to download the created resume as PDF
3. User can Share their resume to their contacts on WhatsApp
4. Options to Clone Resumes which can further minimize the time of creating similar type of resumes.
5. User can also print the Resume
6. Role based Objective Suggestions algorithm.
7. Automatically sorts the resume alphabetically through bubble sorting algorithm.
8. User can Search for a specific resume based on the alphabets.

9. Integration of Captcha Security
10. Integration of OTP based password reset.

1.2 Problem Statement

In today’s competitive job market, creating a well-structured resume that showcases an individual’s skills, education background, experience and worth for the role is very challenging. Due to the competitiveness a well-structured and formatted resume is must to have to make yourself standout among other candidates for the role. A resume is the first impression of any candidate an employers can have so it is crucial for the job seekers to make best impression by showcasing themselves in the best possible way. But People often find challenges in creating those well-structured resume that can impress an employer due to lack of formatting capabilities, lack of knowledge about what to include and what not to include, lack of technical and designing knowledge or lack of information presentation skills. As a result of this difficulty people end up making resumes that are not well structured and don't highlight their strengths and worth which won’t impress the employer. This kind of resume does nothing but degrades the chances of landing a job. The goal of this project is to address and solve these problems by providing an easy-to-use, user-friendly platform that helps people create professional, well-formatted resumes that showcases the worth of an individuals without the need of any designing or formatting skills.

1.3 Objectives

The main objectives of this project are:

1.3.1 To Build a Simple and User-Friendly Resume Builder Platform

User always finds difficult and complex in using an interface of such websites. To resolve that difficulty this system provides an easy to use interface which helps both regular users and professionals to create well-structured and professional resumes with just a few clicks and simple form fill ups.

1.3.2 To Eliminate Manual Formatting and Design Efforts

The resume builder system provides pre-designed resume format so that users don't have to manually design or format their resumes. Rather they just have to provide their personal and professional details.

1.3.3 To Save Time and Improve Competitiveness in the Job Market

The resume builder system will improve and ease resume creation process by allowing users to create high quality resumes quickly and easily which will help them stand out in the competitive job market.

1.4 Scope and Limitation

Scope:

The project will focus on creating a web-based application where users can input their details through a form field and generate well formatted resumes. The application will support different functionality such as font customization options, cloning options, download, objective suggestions and share. Users will be able to include basic personal information, education, work experience, and skills in their resumes. The total design and layout is SEO friendly. The application will be fully responsive and cross browser support allowing users to create resumes from any devices and any browsers.

Limitations:

The application will not provide real-time job suggestions based on the resume. It will not offer advanced features like resume optimization based on score or personalized career advice. The application does not includes more templates to choose from. And anther limitation would be these project is not tested on large load so it may not be optimal for larger traffics.

1.5 Development Methodology

The Software Development Methodology used would be Iterative Development approach in order to make adjustment to changing needs of the project. This means that the application is built in small steps, adding one feature at a time. After each step, each features of the system is tested and the improvements is made based on the feedbacks and test results. This method allows for flexibility and helps to make quick adjustment to the project as needed.

Initial Planning: In this iterative methodology, first of all the key features of the project is planned and designed in a simple way and developed its working flow.

Requirements Analysis and Design: After initial planning and workflow design in this phase all the requirements required for this system is collected, analyzed and roadmap design is made.

Development Phases: Development of each features begins, each phase of iteration implements functionality such as CRUD, text customization options, and resume cloning, sharing downloading options.

Feedback and Testing: After each iteration, Testing is carried out to make sure everything worked well and gathered test results for correction and improvements.

Final Integration: Once all features is added and tested final integration of the project is done.

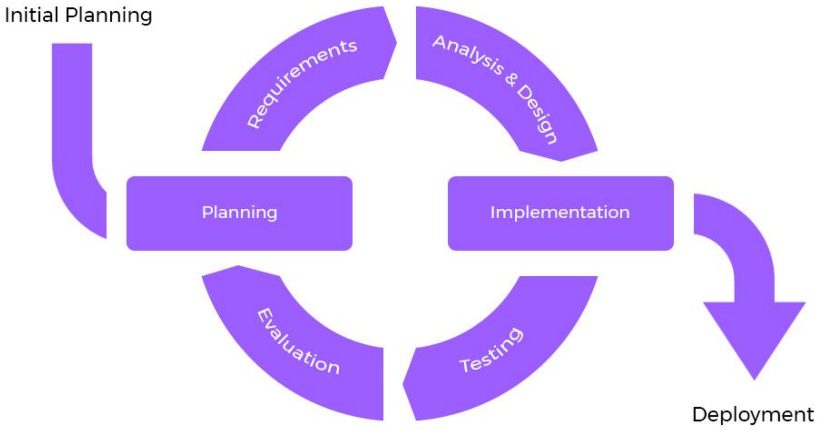


Figure 1.1: Iterative Development Process

1.6 Report Organization

This project report is organized as follows:

Chapter 1: Includes an overview of the project, including the introduction, problem statements, objectives, scope, goals and limitations, development methodology of the system.

Chapter 2: contains background study of the project and Literature review section which covers all the previous research work done in the field of the system.

Chapter 3: This chapter includes system analysis studying about the feasibility of the project along with process and data diagrams.

Chapter 4: gives information about overall system architecture through system design, database design along with the implemented algorithm details.

Chapter 5: Covers the Process of implementation and testing along with all the tools used during the system development.

Chapter 6: Summarizes the outcomes of the project and along with future improvements and work to be done.

Chapter 2

Background Study and Literature Review

2.1. Background Study

Many studies have been performed in the field of resume creation tools showing their evolution and impact on the job application process. Resume builders have become an essential tool for job seekers, helping them create professional resumes quickly and easily. Your resume is just a reflection of yourself in a form of document and it is the way of marketing yourself. Rinki Tyagi et al. [1] solved the problem of manual creation of resumes. Long back resumes were created manually by using MS Word and the format at that time was quite different.

Bharti Kungwani et al. [2] presented a web application - “Analytical Resume Builder” which provides precise knowledge and analyzed stats. In addition, it provides resumes of graduates i.e., placed students. So, the users will get more clarity on their resume. Furthermore, it also informs about upcoming company drives and campus placements. The application uses statistical analysis by collecting quantitative data and then represents it in a visual format. After the successful building of the resume user can export the file in PDF format.

Most recent studies such as Marapaka et al. [3] presented a comprehensive study on resume builder applications in their work titled "Resume Builder Application Vol 8 Issue 3". The study emphasizes the role of these applications in simplifying the resume creation process through customizable templates and efficient data processing capabilities. The findings highlight the significance of resume builder applications in improving user experience and improving the overall quality of resumes. Due to increasing competitiveness of job market there has been a need of creating a professional resume showing job seekers strengths and worth for a job. This need has led to the development of these systems which provides user friendly user interface and experience with various features like customization, templates, suggestions, sharing, downloading as well as real time editing. Due to increase in the use of internet and technologies, job application process has become completely online which has also helped in the rise of these resume builder systems. In this type of systems, predefined suggestions plays a great role by suggesting industry standard suggestions based on their role and strength. Predefined templates and suggestions can significantly reduce the time spent on resume creation, making it easier for users to structure their resumes according to industry norms [4]. Keyword suggestions has also become a critical features for modern resume builder systems as use of keyword relevant to that job specifications increases the chance of getting the job.

In addition to user customization and suggestions, resume builder systems offers grammar checking tools which helps to make resume free from grammatical error as resume is the first way to impress the employers. As mentioned by Charu Aggarwal [5], modern systems have NLP-based grammar analysis and correction techniques. Industry benchmarks are set by tools like Grammarly and Microsoft Word through real-time grammar corrections provided by them. Additionally sorting and searching algorithm such as bubble sort, binary search helps to navigate to the resume easily. According to Cormen et al. [6], Bubble Sort gives an intuitive way of sorting; it is, however, less efficient for large sets of data compared to Quick Sort or Merge Sort. Chew and Ong [7] explored the integration of augmented reality features in resume building applications in their work on "A Resume Generator with Augmented Reality Features". The study introduces innovative approaches to resume creation, leveraging augmented reality technology to enhance user engagement and presentation effectiveness. Nowadays with the rise of AI, modern resume builder system also uses AI and ML to create dynamic resumes based on the user preferences and needs automatically.

2.2. Literature Review

The resume builder systems on the internet were explored and their useful features have been added to the application. The biggest problem in most of these systems is that they charge a high cost every time a user builds a resume. The services do come with expert advice to improve the resume but charges hundreds of dollars. There are a few systems that are free of cost like the one I. Wu [8] describes the Pro Resume that focuses only on infographic resumes. In the world of resume builder systems, several systems shares a same parallel concepts and technologies to the proposed “**Resume Builder**” system. These platform offers similarities in their objective, scope and problem solved. Much like “Resume Builder” these systems aim to improve and enhance the process of building professional resumes in minimal time and efforts.

Zety, Novoresume, Canva, Resume.io and Resume Genius are some of the most popular tools in the field of resume building. Each platform offers unique features, designs, and functionalities, focusing to different user needs.

Zety is an online resume builder known for its simplicity and effectiveness. It has prebuilt customizable templates which offers real time content suggestions based on provided job descriptions. Zety has been designed to help users in creating an ATS-friendly resume easily. Many users has liked this system for resume creation due to its user-friendly interface and time-saving features.

Novoresume is another online resume builder platform. Novoresume is free to use. It is a popular resume builder known for its user friendly interface and professional templates. Users are able to see the changes in their resume, which really makes it easy to experiment with layouts and designs. Users have reviewed Novoresume as easy to work with and producing professional output.

Canva is a graphic design tool with a resume building feature. It allows users to create visually appealing resumes using drag-and-drop functionality. User can choose from a wider templates and customize it according to their needs. Canva focuses more on the aesthetic look and less on the optimization for ATS. Resume created from Canva is more useful for creative industries.

Resume.io is an online resume builder designed for ease of use. It is fast, effective and creates strong and professional resumes. Resume.io provides 30+ resume and cover letter templates to guide and give resume examples. Resume.io provides online builder tool and recruiter-tested templates which are used by more than 25 million people around the globe.

Resume Genius uses a questionnaire based approach to gather user information and generate resumes. This system simplifies the process for users from generating bullet points to automatic formatting of the resume. You can have an option to select different resume categories such as minimalist, elegant, modern, advanced, corporate, and majestic and so on.

Research on “How to make ATS Friendly Resumes”

The Ladders eye-tracking study suggested that a recruiter spends about 7.4 seconds on a resume hence it is very important to focus on having a clean-looking resume, having short declarative statements, and most importantly having the standard headings and subheading without any extraordinary design experiments [9]. Thus, the details presented by Risavy in the Resume Research Literature [10] serve a very important role in the development of every template for the project. The valuable tips shared by Clift in the “How to beat ATS” [11] also list a checklist that should be followed while making resumes. The design tips shared on the RIT’s Career Services and Co-op and Purdue’s Resume Design articles [12] [13] also provide important insights.

Each of these platforms and studies performed provides valuable insights into the field of resume building. By analyzing these systems, we can better understand the features and design principles need to make a better and user friendly resume builder system. By analyzing existing platforms we can find current systems valuable features, and also how they failed to address key user needs such as ease of use, dynamic customization, affordability and so on.

Chapter 3

System Analysis

3.1 System Analysis

System Analysis is the process of collecting and analyzing data identifying issues, and breaking down a system into multiple components. The purpose of system analysis is to achieve the desired functionality. It specifies what the system do. A system analysis of a Resume Builder System involves analyzing different features and components that makes the system. They are:

- User registration and authentication
- Resume Management
- Objectives Suggestion and searching
- Resume sharing
- PDF Generation

3.1.1 Requirement Analysis

Requirement analysis determines the needs and expectations of stakeholders, users in new software products. Gathering, confirming, and recording needs and criteria are all part of this process. This process helps to reduce the misunderstanding and confusion during actual project implementation. It is a documentation that contains the functional and non-functional requirements for the creation of Resume Builder system.

3.1.1.1 Functional Requirements

This is the main requirement of the system that covers all the fundamental functions of the Resume Builder system. These requirements outline the tasks that the system must complete in order to satisfy the users and give value to them Functional requirements include:

User Authentication:

Authentication is the process of identifying the user. . First user need to login in order to create or see the resumes. Authentication creates a safe environment and helps to protect our system resources from unauthorized access by limiting access of data to authorized individuals only. Implemented Password verification and classification algorithm, captcha, email and OTP based user verification to achieve Authentication. By entering basic information like their name, email address, and a strong password, new users can create accounts. Email verification is perform to verify the account creation

Resume Creation:

Users can click the "Add New" button to create a new resume.
Input fields include details such as:

- Resume title
- Full name
- Email address
- Objective
- DOB
- Address

To add more personal touch, other field includes Gender, Nationality, Martial Status, Hobbies, and Language Spoken and so on.

Resume Management:

Users can manage all their resumes from myresumes dashboard.

Available operations:

- Open
- Edit
- Delete
- Clone

To improve resume details, users can add work experience, education, certifications, and skills.

Resume Viewing:

Users can view resumes on preformatted, print-ready pages. To view the resume user must click on open button.

Key features include:

- Share via WhatsApp: Instantly share resumes with your contacts.
- Print: Directly print resumes in a professional format.
- Customization: Change font styles and background colors based on individual preferences.
- Download as PDF: Save resumes locally for offline use.

Resume Filtration:

- Users can sort and filter the resume alphabetically as well as search it through specific key from the search bar.

If a password is forgotten, a reset email is sent using PHPMailer integrated with Gmail’s SMTP server. This ensures secure recovery of user accounts.

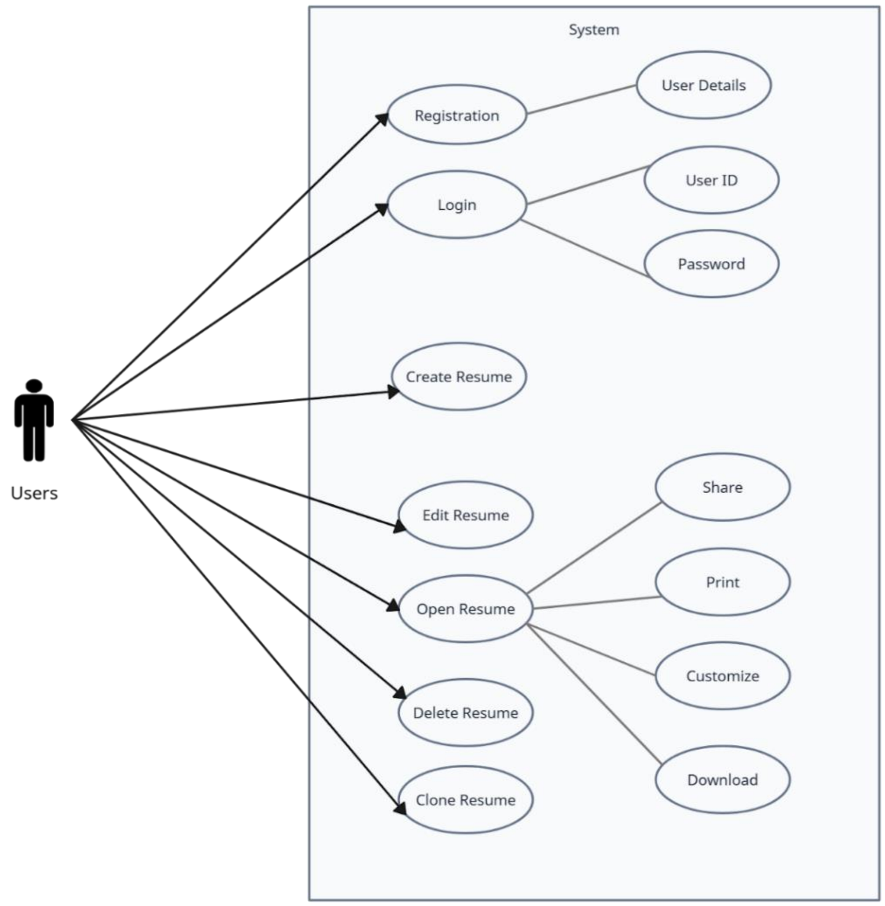


Figure 3.1: Use Case Diagram

3.1.1.2 Non-Functional Requirements

Nonfunctional requirements are secondary requirements of software system that defines the quality attributes of the system such as performance, security, Load, usability, Maintainability and so on. These requirements ensure that the system performs properly under the given conditions.

Here are the non-functional requirements for Resume Builder System:

Usability:

Resume builders user interface is made simple and straightforward to use so that any individuals with technical skills or not, they can use it. Clear and visually pleasing feedback is provided for different activities like saving, account creation or deleting resumes using features like SweetAlert notifications.

Performance:

To provide real time and fast interactions, the system makes use of AJAX and jQuery. This reduces page reloads and enhances the user experience by loading the page fast.

Scalability:

The database schema is designed for future extensions, such as adding more resume sections or accommodating more number of users.

Security:

Security is maintained by securely hashing the Passwords by using password_hash functions and kept to prevent unauthorized access. Only authorized users can reset their credentials which is implemented through OTP based email verification.

Compatibility:

The system is compatible with all major web browsers, including Chrome, Firefox, and Safari. System is also compatible with all the devices including mobile phones.

3.1.2 Feasibility Analysis

During feasibility analysis, the system's technical, operational, and financial feasibility is analyzed. Feasibility study helps to determine whether the project is worth doing or not. This stage guarantees that the project is feasible and can be achievable with our available resources. The scope of this project is to create simple to use resume builder system. In feasibility study phase of Resume Builder we had undergone the following steps:

3.1.2.1 Technical Feasibility

Technical feasibility determines whether the current technical resources is sufficient for the new system or not. The systems will be designed with the stable and manageable tool and technologies like:

- Backend: PHP is used for backend development because of its ease of use, adaptability, and MySQL compatibility.
- Frontend: Used Html, CSS, jQuery and Bootstrap to provide dynamic content and responsive design to the users.

- Database: XAMPP-hosted MySQL provides a dependable and safe database management environment.
- Email Integration: For account verification and password recovery, PHPMailer is combined with Gmail server for safe and effective email delivery.
- Hardware specification: laptop , Desktop
- Software specification: Window Operating system

All these tools are completely free for students. This proves that this new system is technically feasible.

3.1.2.2 Operational Feasibility

Operational feasibility evaluates the system's capacity to meet user needs successfully. As we are divided into group of three people. Every team members is familiar with Software Development Lifecycle, system analysis, Web Development. So there won’t be any problem in operating and implementing the system and its functionality. System contains easy to use, simple User Interface so that user can easily operate and use this systems.

3.1.2.3 Economic Feasibility

Economic feasibility evaluates the cost-effectiveness of the project. It determines whether the time and budget is available for the system development or not. In this project we will used Open-source tools like PHP, MySQL, and Bootstrap so there is no any cost overhead. This proves that is system is economically feasible.

3.1.2.4 Schedule Feasibility

Schedule feasibility guarantees the project tasks to be completed within the allotted time frames. Schedule feasibility is one of the most important factor in the development of the systems like resume builder. Schedule feasibility determines the completion of the project with in the time with the team work and cooperation of all the team members. With proper project schedule we have maintained the schedule feasibility of our project. Here is time schedule of the project.

S.N	Task	Start Date	End Date	Duration (days)
1.	Initial Planning	2024-09-29	2024-10-04	6 days
2.	Requirements Analysis and Design	2024-10-05	2024-10-16	12 days
3.	Development of Key modules	2024-10-17	2024-11-18	33 days
4.	Testing and Feedback	2024-11-19	2024-11-25	7 days
5.	Development of secondary features	2024-11-26	2024-12-15	20 days
6.	Final Testing and Optimization	2024-12-16	2024-12-27	12 days
7.	Report Writing	2025-01-09	2025-01-20	12 days

Table 3.1: Project Scheduling

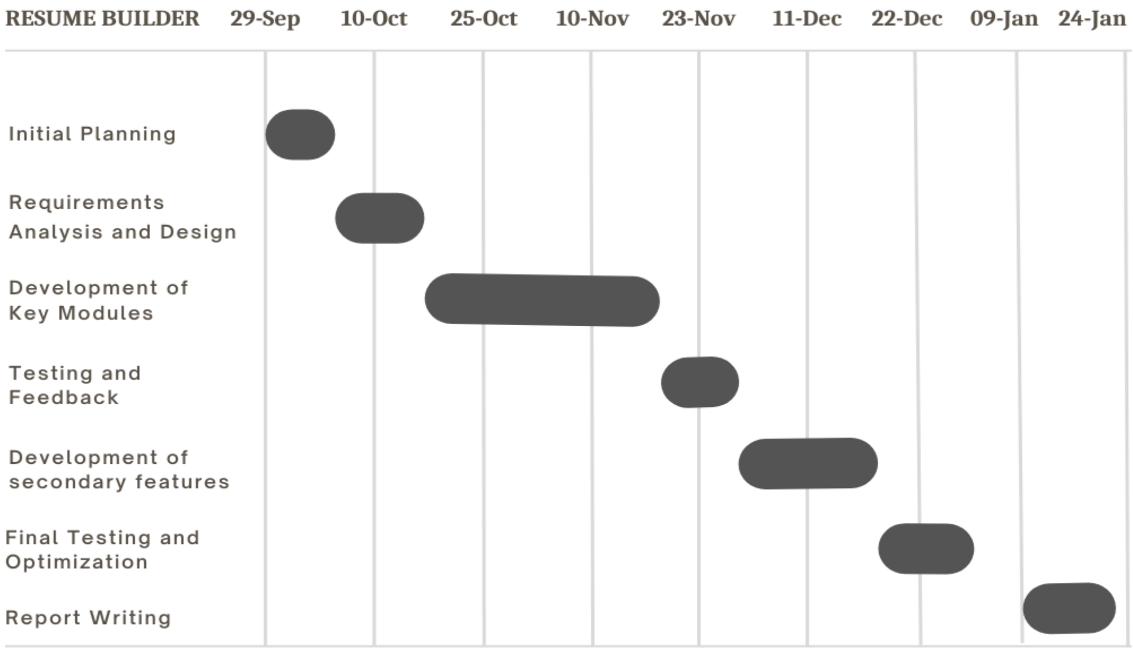


Figure 3.2: Gantt chart

3.1.3 Analysis

In this project, a structured method is used for both analysis and design phases. . The approach focuses on organizing and documenting the system in a structured and systematic manner which provides clarity in data and process flow.

3.1.3.1 ER Diagram

The Entity Relationship (ER) diagram is a type of flowchart that shows how entities such as people, objects or concepts relate to each other within a system. It emphasizes the one-to-many interactions between users and resumes, as well as resumes and their corresponding components (for example, education, experience, and skills).

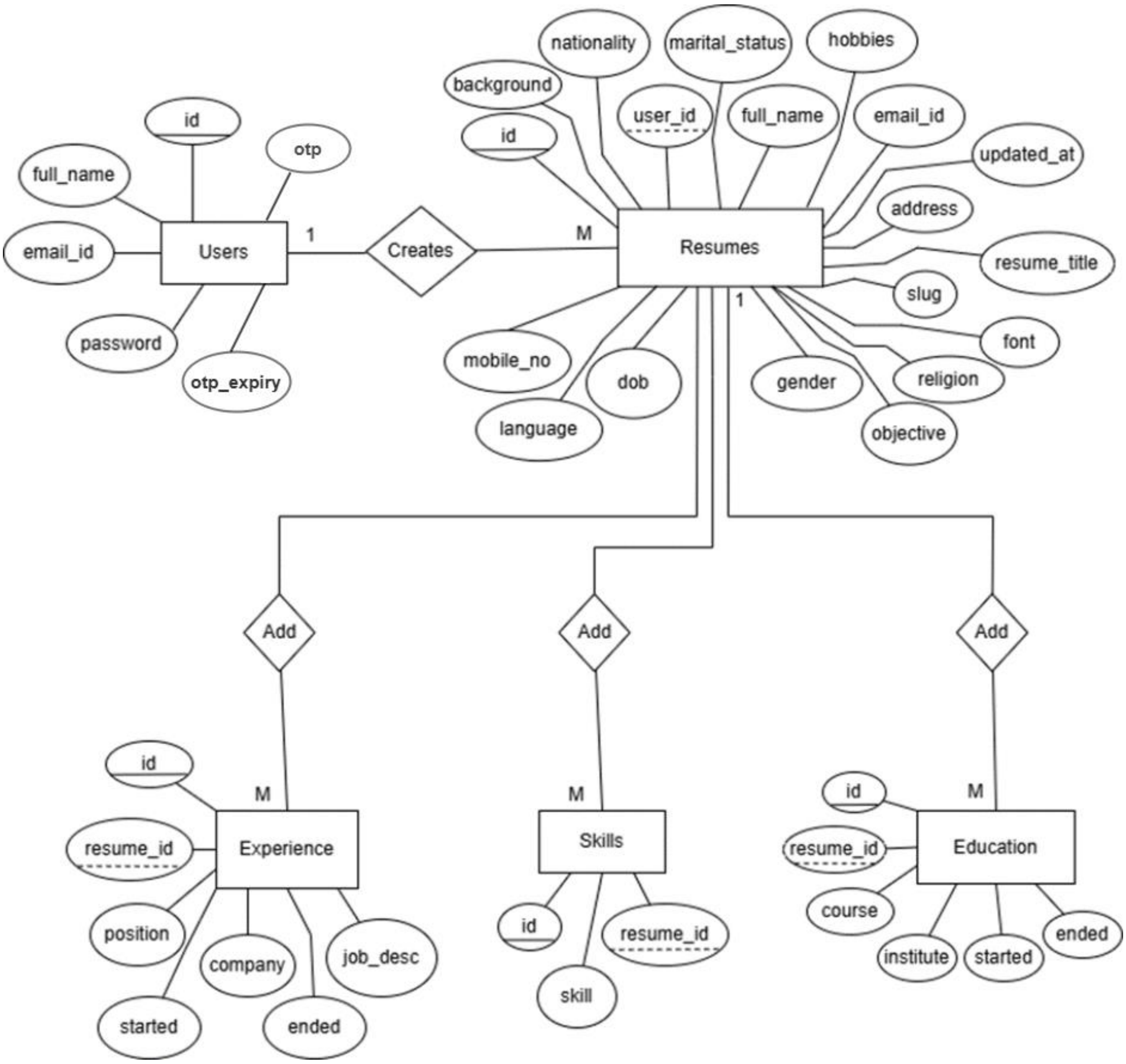


Figure 3.3: ER Diagram

3.1.3.2 DFD Diagram

Dataflow Diagram (DFD) shows the flow of data with the systems. There are different level of DFD such as DFD Level 0, DFD Level 1, and DFD Level 2. Each level represents the data flow.

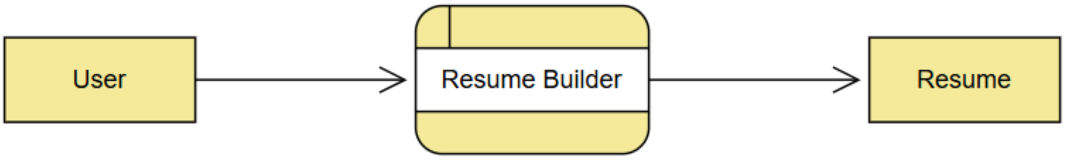


Figure 3.4: DFD Level 0

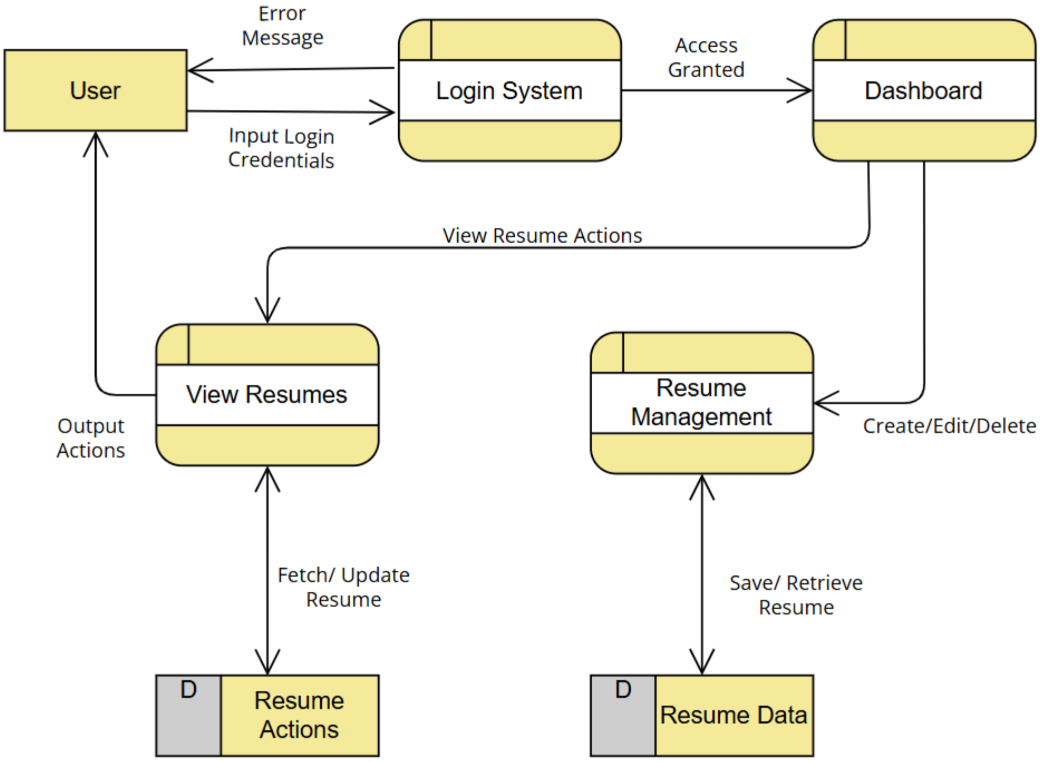


Figure 3.5: DFD Level 1

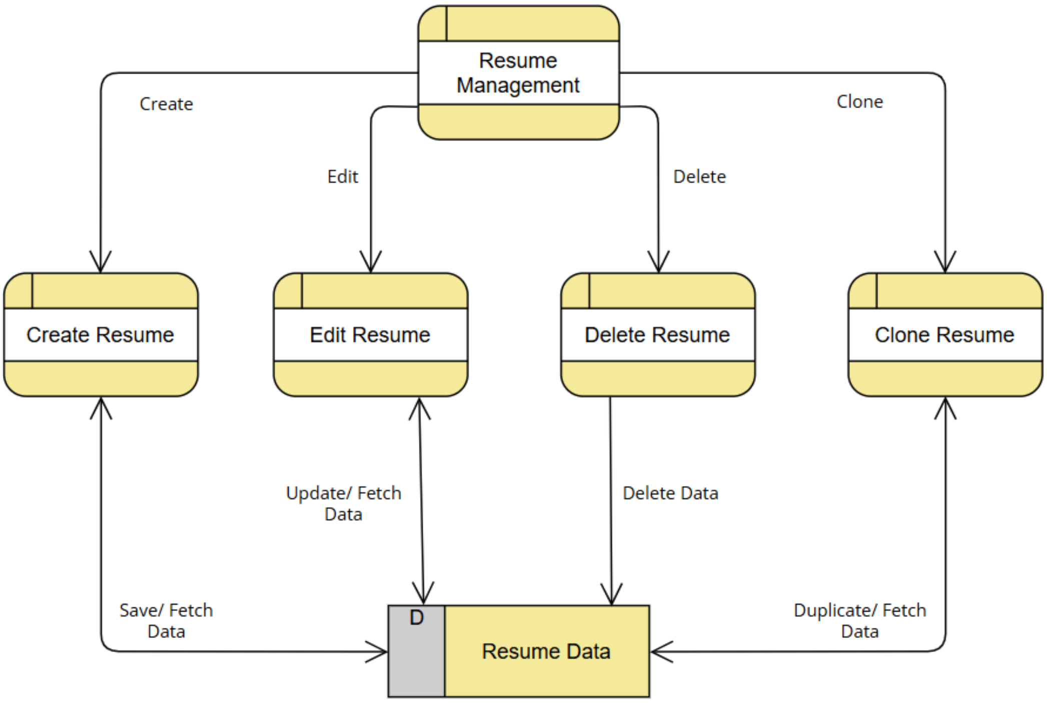


Figure 3.6: DFD Level 2

3.1.3.3 Flow Chart Diagram

The flow chart is a graphical diagram that shows a picture of the each steps of a process in a sequential order. It shows the logical sequence of operations in the system. Key workflows, such as user registration, login and resume management, are visualized to help users understand system activities.

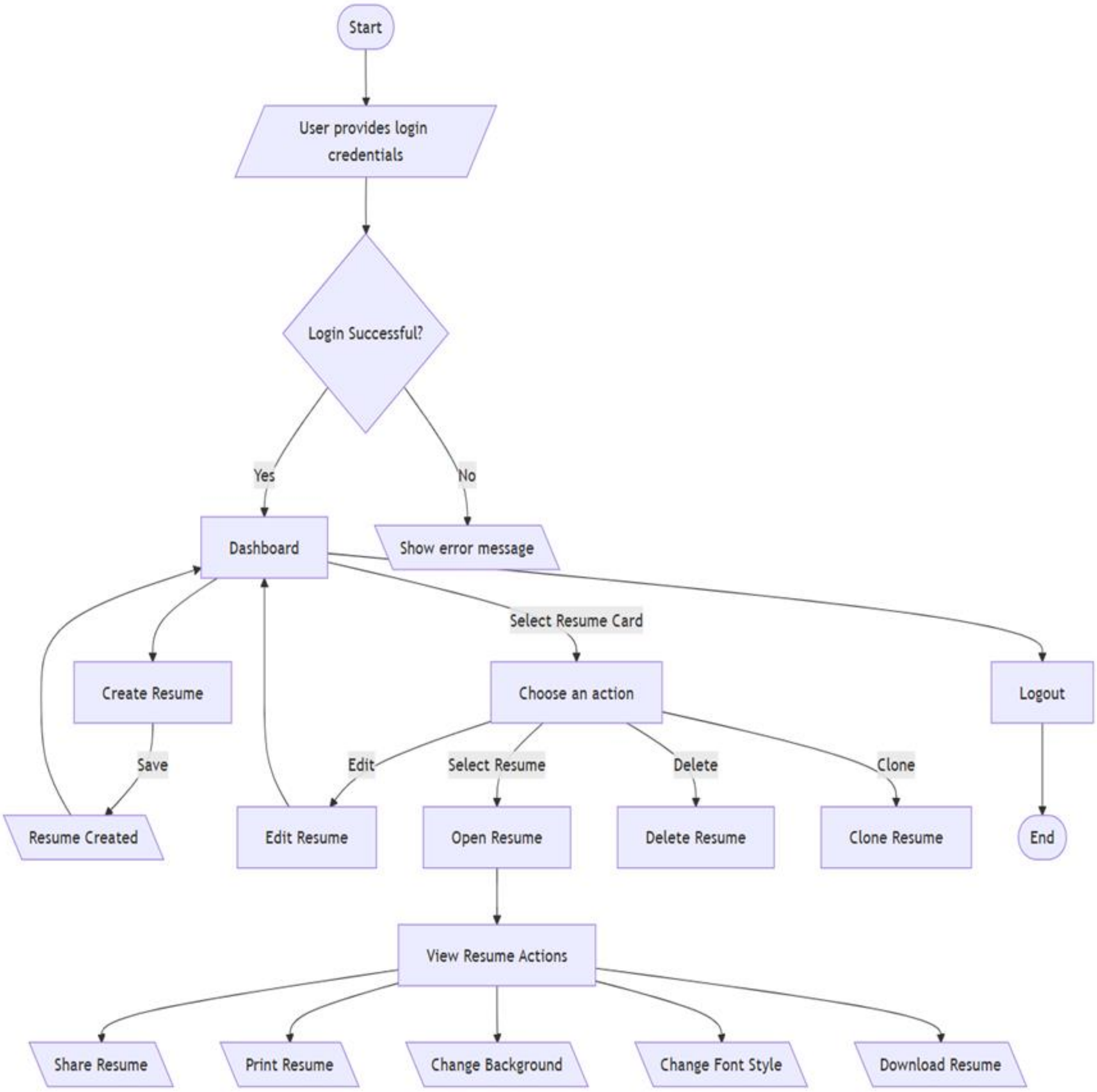


Figure 3.7: Flow Chart Diagram

Chapter 4

System Design

4.1 System Design

System Design is the process of creating a design or blueprint of components, modules, architecture to meet the requirements. It involves converting the systems requirements into a detailed design the acts as the guidelines for the system implementation phase. This project uses structured method in both design and analysis.

4.1.1 Architecture Design

The System architecture design for Resume Builder is client server architecture. Due to this modular client-server architecture this resume builder systems is scalable and maintainable. It also maintains smooth interaction between components, efficiency in data processing, and a responsive user experience.

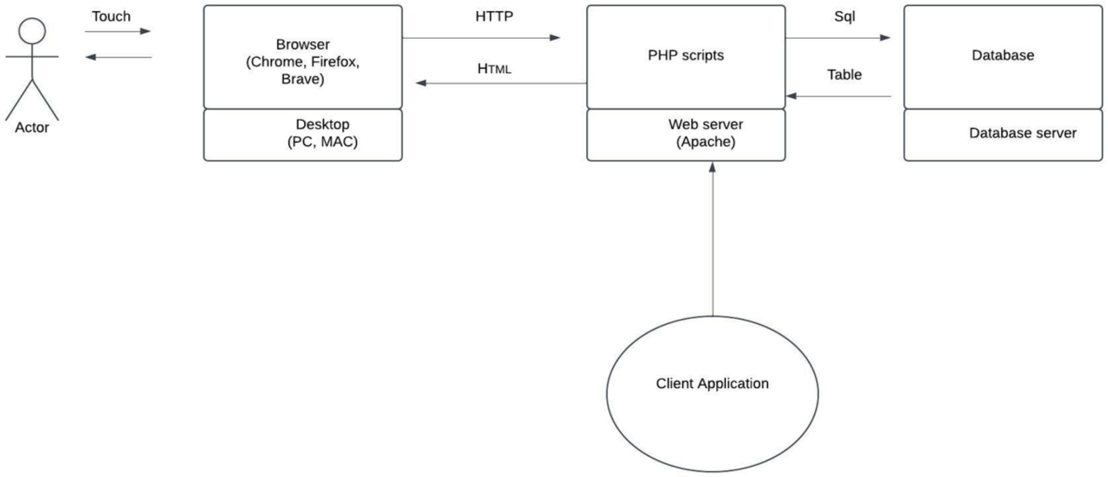


Figure 4.1: 3 tier client- server Architecture

4.1.2 Database Design

Database design is the set of tasks involving various steps to implement a database. The main goal of database design is to provide a schema for effective management and organization of systems data.

Database Tables:

Here are the database tables specified in the project:

users table: Stores user credentials and basic profile information.

resumes table: Contains single-value fields for each resume.

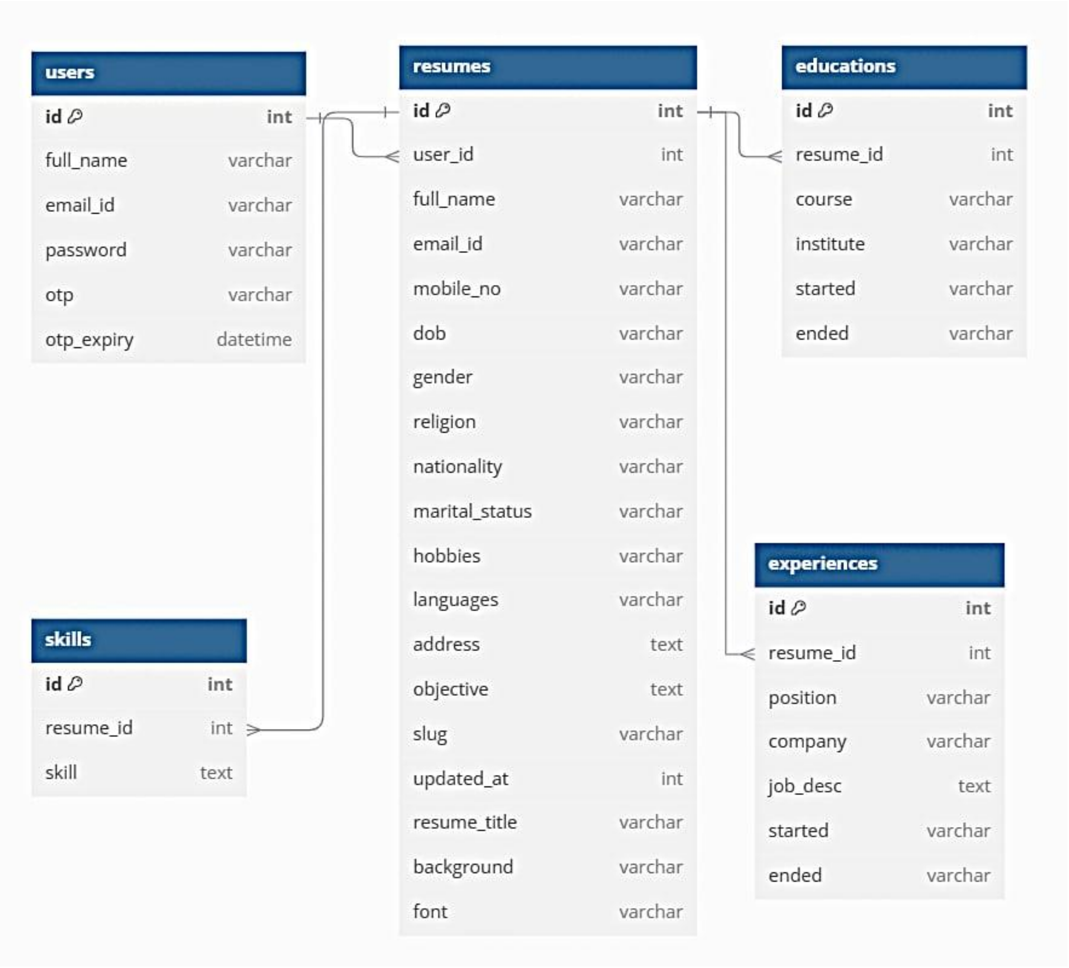
education table: Tracks educational qualifications of the resumes.

experience table: Tracks job experiences linked to resumes.

skills table: Records skills associated with resumes.

4.1.2.1. Schema Diagram

The schema diagram shows the relationship between database tables, showing all the foreign key constraints and data integrity. For example, the user_id in the Resumes Table acts as a foreign key, connecting resumes to their corresponding users.



users	
id	int
full_name	varchar
email_id	varchar
password	varchar
otp	varchar
otp_expiry	datetime

resumes	
id	int
user_id	int
full_name	varchar
email_id	varchar
mobile_no	varchar
dob	varchar
gender	varchar
religion	varchar
nationality	varchar
marital_status	varchar
hobbies	varchar
languages	varchar
address	text
objective	text
slug	varchar
updated_at	int
resume_title	varchar
background	varchar
font	varchar

educations	
id	int
resume_id	int
course	varchar
institute	varchar
started	varchar
ended	varchar

experiences	
id	int
resume_id	int
position	varchar
company	varchar
job_desc	text
started	varchar
ended	varchar

skills	
id	int
resume_id	int
skill	text

Figure 4.2: Schema Diagram

4.1.3 Forms and Report Design

4.1.3.1 Forms Design

4.1.3.1.1 User Registration Form Design

Shows how user can specify their input details and register themselves into the Resume Builder system.

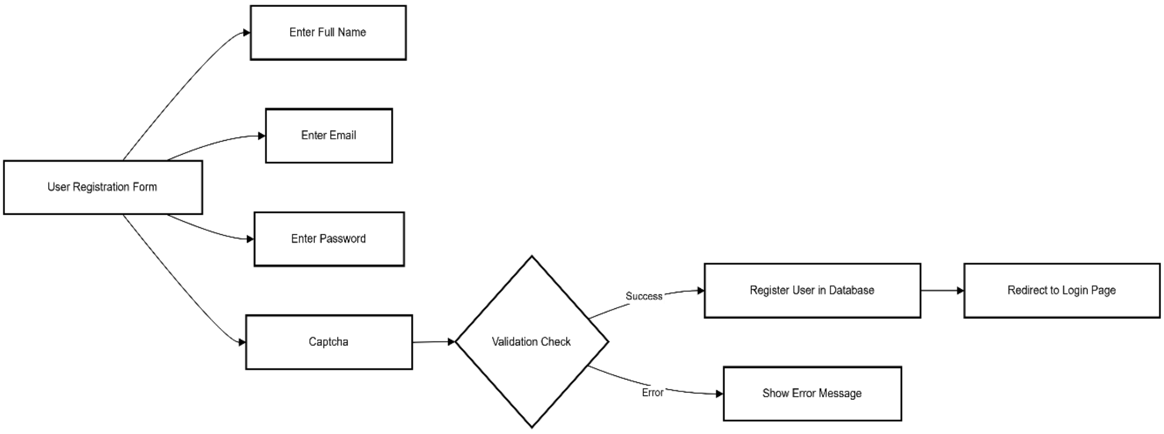


Figure 4.3: Registration Form Design

4.1.3.1.2 Login Form Design

Shows how registered user can specify their valid credentials to gain access to resume builder system dashboard.

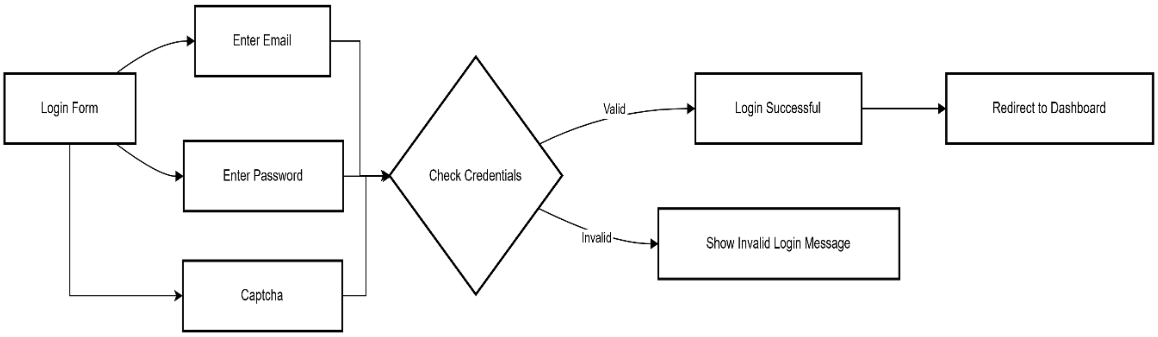


Figure 4.4: Login Form Design

4.1.3.1.3 Create Resume Form Design:

After successful validation and login into the system, Create Resume form design shows how user can create their resumes:

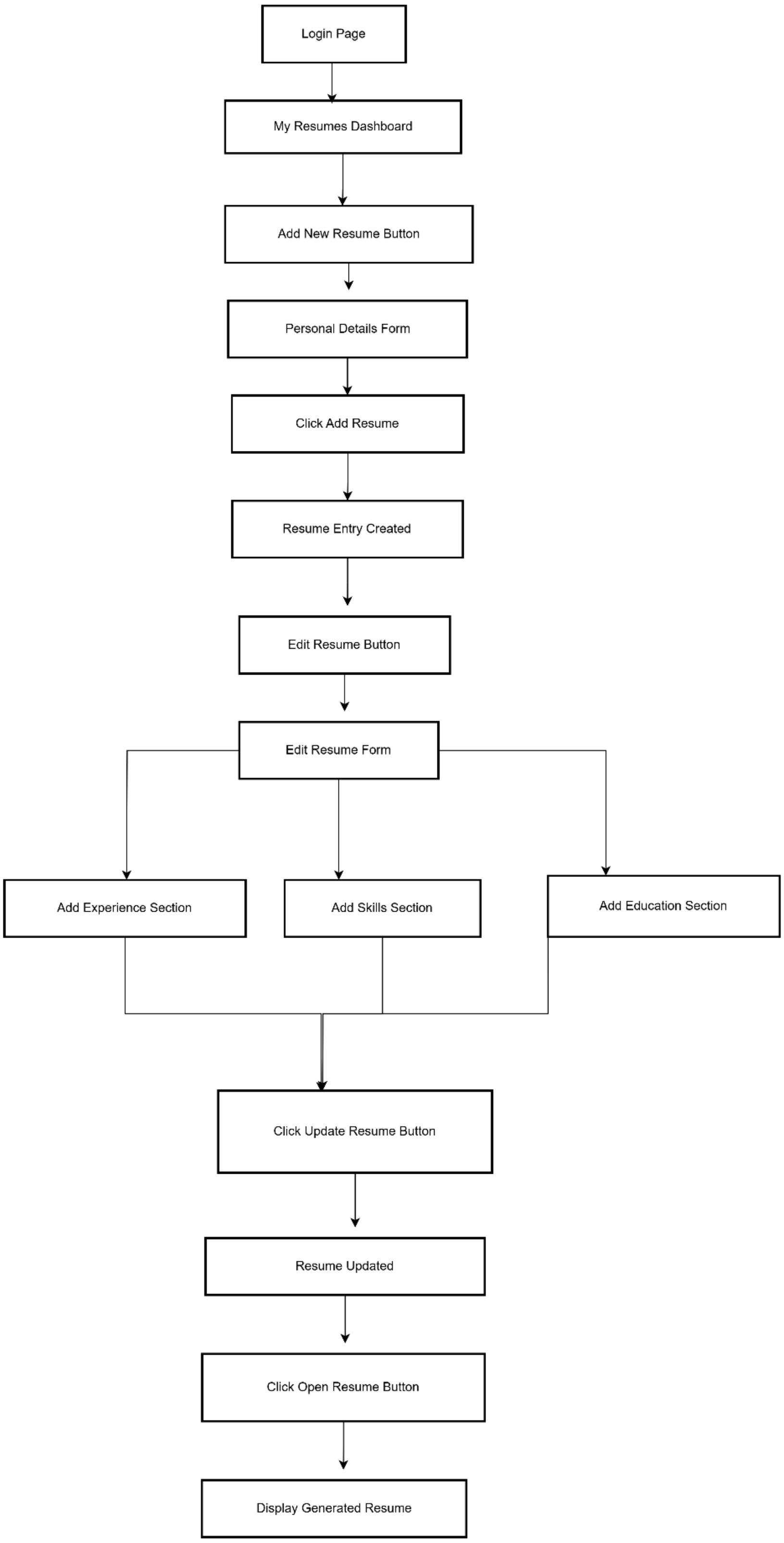


Figure 4.5: Create Resume Form Design

4.1.3.2 Reports

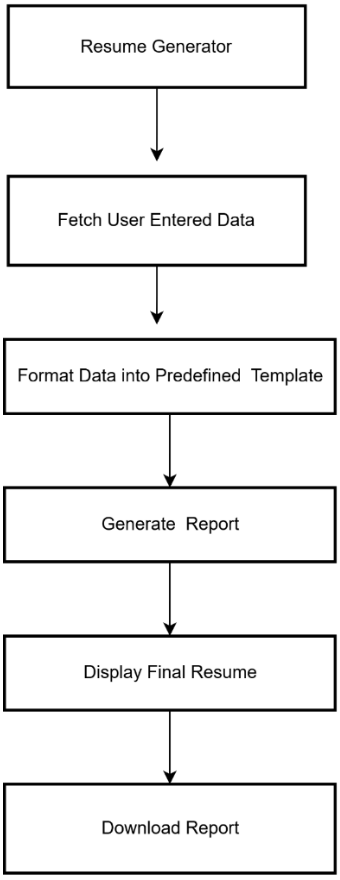


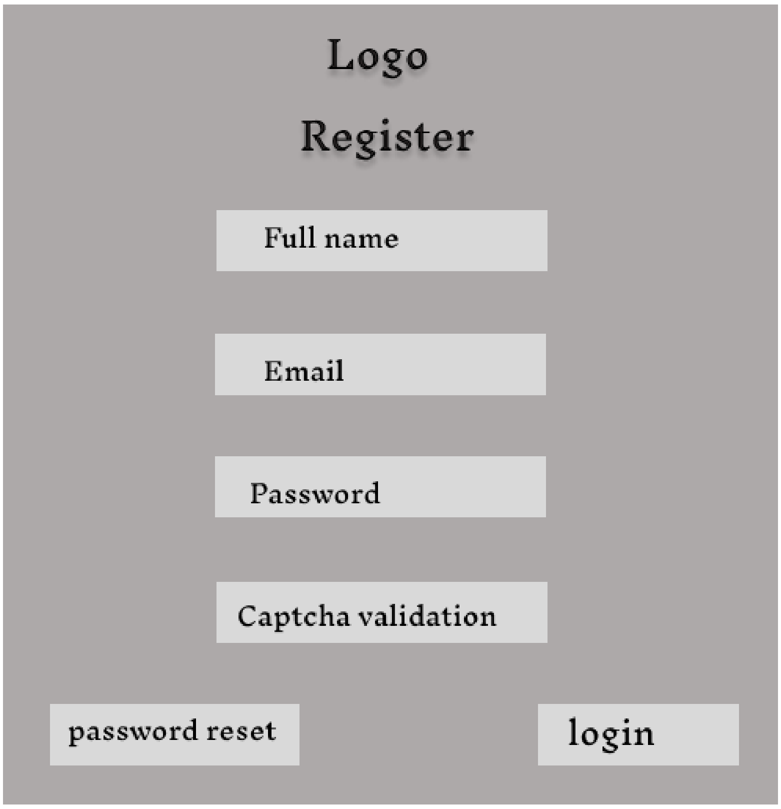
Figure 4.6: Resume Generation Report design.

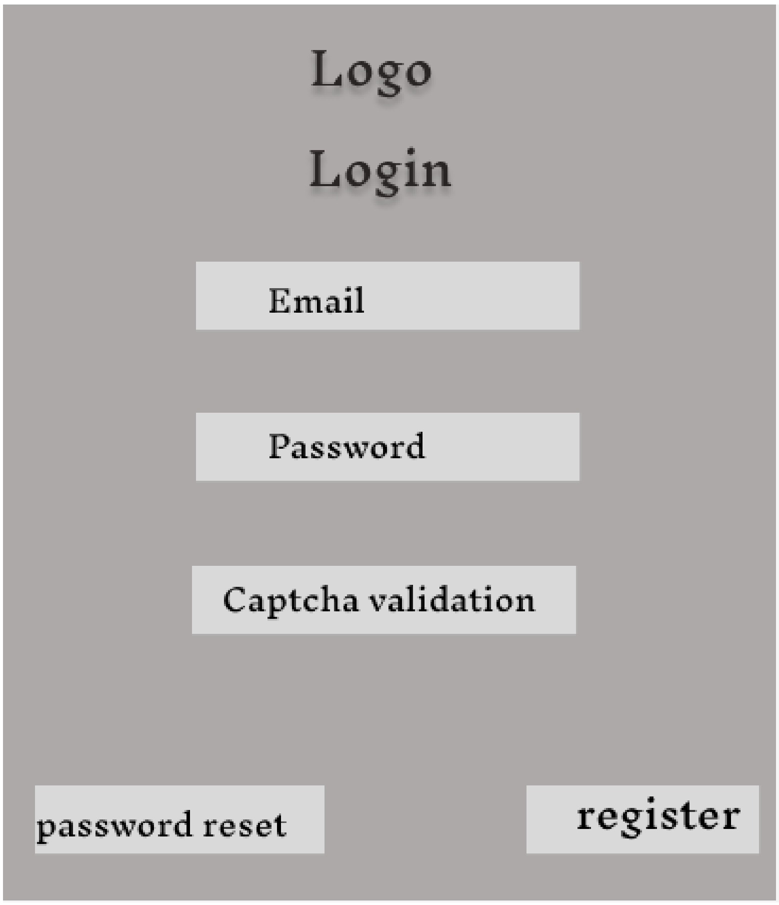
This is the final resume created when user enters their details through form fields. After creating the new resume, click on open button to generate the resume as shown in the figure above.

4.1.4 Interface and Dialogue Design

4.1.4.1 Interface Design

The user interface (UI) is a crucial part of any system because it affects how easily new users can understand and interact with the system. A well-designed UI helps users quickly find what they need and complete their tasks without confusion. In our system, the interface will be designed to be clear and intuitive, making it straightforward for users to navigate and achieve their goals. Here is how our system will look like:


Logo
Register
Full name
Email
Password
Captcha validation
password reset
login


Logo
Login
Email
Password
Captcha validation
password reset
register

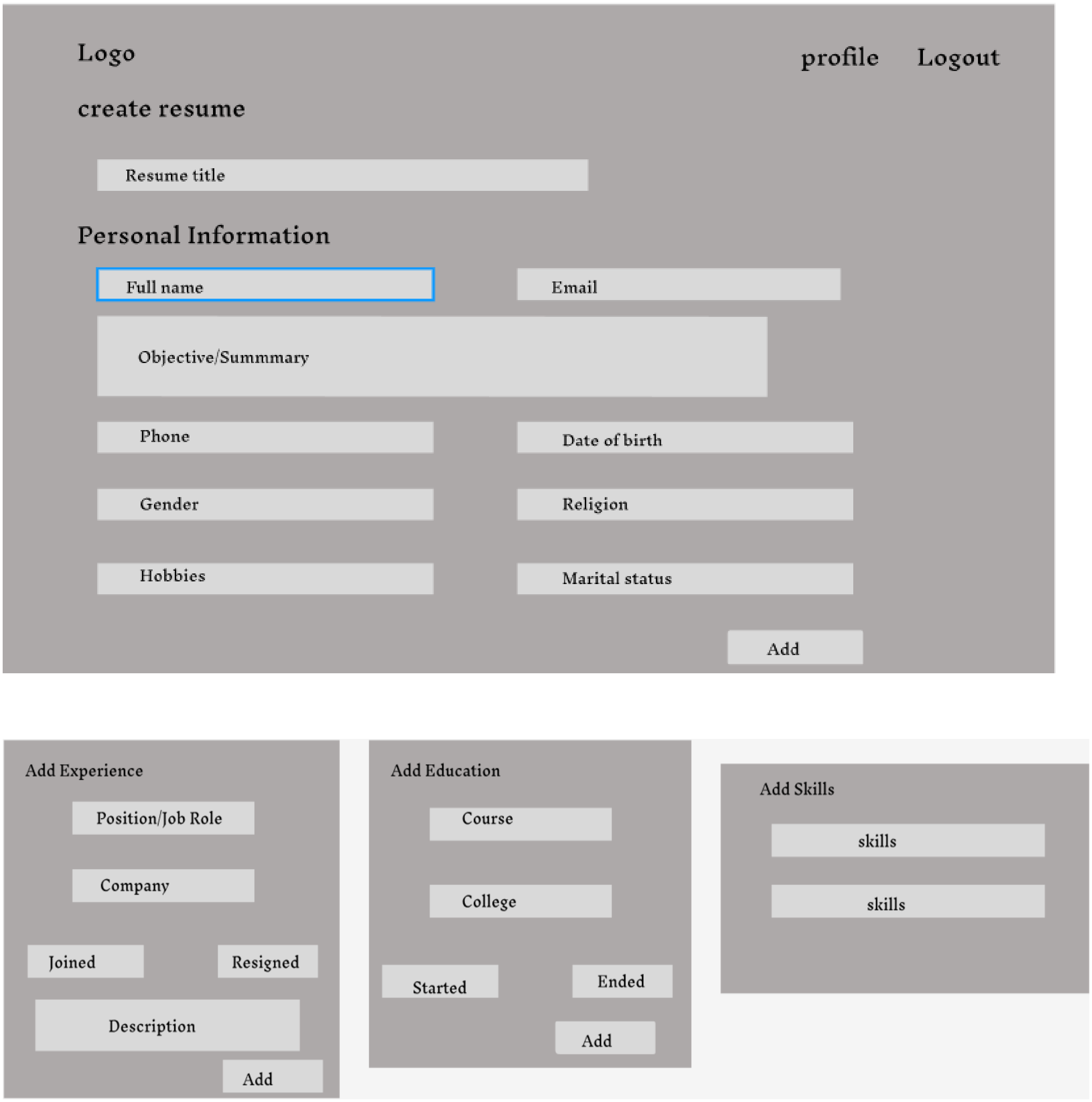


Fig 4.7: Interface Design

4.1.4.2 Dialogue Design

Dialogue design shows how the system communicates with the user to guide them through each process within the system.

4.1.4.2.1 Registration Dialogue Design:

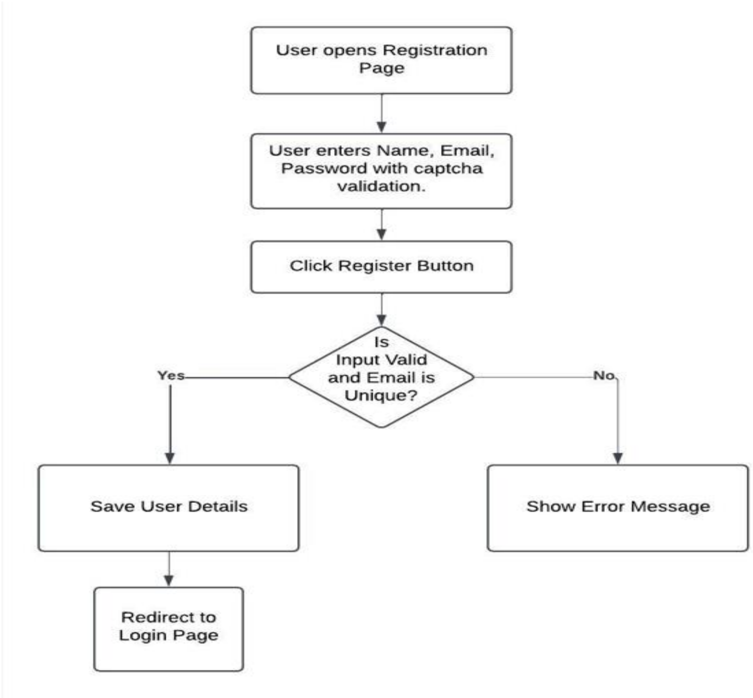

User opens Registration Page
User enters Name, Email, Password with captcha validation.
Click Register Button
Is Input Valid and Email is Unique?
Yes
No
Save User Details
Show Error Message
Redirect to Login Page

Figure 4.8: Registration Dialogue Design

4.1.4.2.2 Login Dialogue Design

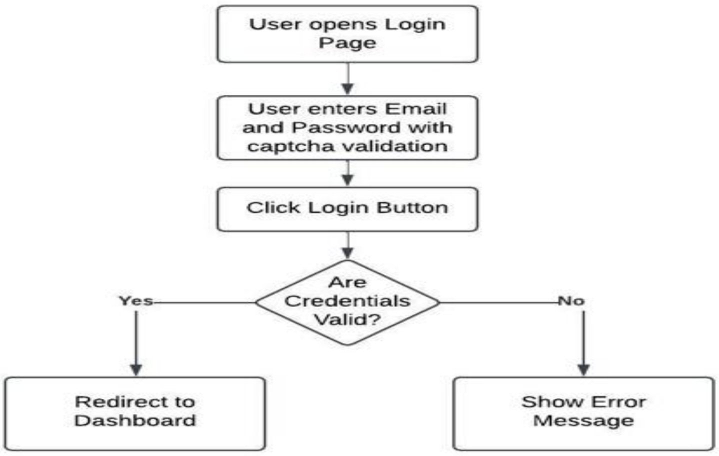


Figure 4.9: Login Dialogue Design

4.2 Algorithm Details

In this project we have implemented several algorithm that has significance impact in our project. These algorithm may be simpler but provides great value to the project. Here are the algorithms implemented in this project:

Password Strength Validation and Categorization Algorithm:

In this algorithm, system checks whether a password meets certain criteria to determine if it has strength. On the basis of different criteria such as length, combinations of uppercase, lowercase letters, special characters this algorithm determines the strength of the password and classifies it. Depending on the score and previous calculations password is categorized into one of five strength levels which are: Very Weak, Weak, Medium, Strong, or Very Strong. If password is very weak and weak then user cannot use that password. Purpose of implementing this algorithm to understand and learn about the best security practices to protect our system from different unauthorized attacks.

Logic Overview:

Length Check: The password must be at least 8 characters to pass.

Uppercase Letters: The password must contain at least one uppercase letter (A-Z).

Lowercase Letters: The password must contain at least one lowercase letter (a-z).

Numbers: The password must contain at least one digit (0-9).

Special Characters: The password must contain at least one special character.

Categorization:

Very Weak: If password meets less than 3 criteria.

Weak: If 3 criteria is met.

Medium: If 4 criteria is met.

Strong: If 5 criteria is met.

Very Strong: All 6 criteria met.

Pseudo code:

Algorithm: CheckPasswordStrength

Input: password (string)

Output: strength_category (string)

```
1. Initialize strength ← 0
2. Calculate password_length ← length of password
3. // Evaluate password length
  If password_length ≥ 8:
     Increment strength by 1
  If password_length ≥ 14:
     Increment strength by 1
4. // Check for uppercase letters
  If password contains at least one uppercase letter (regex: [A-Z]):
     Increment strength by 1
5. // Check for lowercase letters
  If password contains at least one lowercase letter (regex: [a-z]):
     Increment strength by 1
6. // Check for digits
  If password contains at least one numeric digit (regex: [0-9]):
     Increment strength by 1
7. // Check for special characters
  If password contains at least one special character (regex: [^a-zA-Z0-9]):
     Increment strength by 1
8. // Categorize password strength
  If strength < 3:
     strength_category ← "Very Weak"
  Else if strength = 3:
     strength_category ← "Weak"
  Else if strength = 4:
     strength_category ← "Medium"
  Else if strength = 5:
     strength_category ← "Strong"
  Else:
     strength_category ← "Very Strong"
9. Return strength_category
```

This algorithm plays a vital role in:

- Protecting user data and accounts.
- Educating users on password strength.
- Reducing vulnerabilities in the system.
- Ensuring the project follow modern security standards.

Objective Suggestion Algorithm

Another useful algorithm implemented is objective suggestions. It provides predefined list of suggestions for the Objective of a resume based on the selected resume title (such as Frontend Developer, QA etc.). The algorithm is designed to improve the user experience by offering resume content based on the user's entered role. These suggestions are dynamically fetched using a query string parameter (title), and its suggestions are returned in a JSON format for easy use on the frontend.

Algorithm Overview:

- The algorithm takes a resume title (e.g., "Frontend Developer", "Quality Assurance", "React Developer") as input from the query string parameter title in the create resume form.
- The algorithm looks up the resume title in a predefined list of suggestions.
- If the title is found, the algorithm returns an array of suggested objective statements for that resume_title.
- If no title is provided or if the title doesn't exist in the predefined list, an empty array is returned.

Use Case:

A user types "Frontend Developer" as their resume title. The algorithm fetches a list of suggested objectives based on the frontend development roles and displays them as options to the user.

Pseudo Code:

Algorithm: ObjectiveSuggestions

Input: user_input (string), resume_title (string)

Output: Displayed list of suggestions

Step 1: Backend Logic (PHP)

```
1. Defined predefinedSuggestions ← a dictionary containing titles as keys and their respective suggestions as values.
2. Get title from the query string:
  a. If "title" exists in the query string:
     title ← value of title
  b. Else:
     title ← empty string
3. Retrieve suggestions based on the title:
  a. If predefinedSuggestions contains the given title:
     suggestions ← predefinedSuggestions[title]
  b. Else:
     suggestions ← empty array
4. Convert suggestions to JSON format:
  response ← JSON object with "suggestions" as the key and suggestions as the value.
5. Return response.
```

Step 2: Frontend Logic (JavaScript)

```
1. Added an event listener to the input field ('objective-field') for 'input' events.
2. On user input:
  a. Trim user_input to remove leading and trailing spaces.
  b. Get resume_title from the input field ('resume_title').
3. If length of user_input > 2:
  a. Construct the API URL:
     - api_url ← "actions/objective_suggestions.php?objective=" + user_input + "&title=" + resume_title
  b. Send an AJAX GET request to api_url.
  c. On receiving a successful response:
     i. Parse the response into a JSON object.
     ii. Clear suggestions_list container.
     iii. If the response contains suggestions:
        - For each suggestion in the response:
           - Create a new list item.
           - Set the text of li to the suggestion.
           - Add a click event listener to list:
              - On click:
                 - Set the value of 'objective-field' to the suggestion.
                 - Hide suggestions_list.
           - Append li to suggestions_list.
        - Display suggestions_list in the UI.
     iv. Else:
        - Hide suggestions_list if no suggestions are available.
  d. On error:
     - Log the error message to the console.
4. If length of user_input ≤ 2:
  a. Hide suggestions_list.
---
```

Step 3: Hide Suggestions When Clicking Outside

```
1. Added an event listener to the document for 'click' events.
2. On a click event:

   a. If the clicked element is not within suggestions_list and is not the 'objective-field':
      - Hide suggestions_list.
```

This Objective Suggestion will efficiently provide the users with helpful and relevant resume objective suggestions based on their selected job title. It simplifies the resume creation process by saving time as well as by giving professional and creative objectives that align with the job specifications.

Chapter 5

Implementation and Testing

5.1. Implementation

During implementation of this project, Iterative Development was followed which involved dividing larger project into smaller tasks or iterations and developing each iterations individually until the actual process is completed. We implemented this process as it is particularly effective in meeting both functional and non-functional requirements through continuous improvement, testing and adaptability.

By breaking the project into smaller phases, we make sure that each stage was carefully planned, implemented, tested, and refined. The process included defining the project concept, developing core features, additional functionalities, and iteratively testing and optimizing the system. Here are the steps you can follow to implement this process:

- Define the systems concept and create a prototype of the system
- Identify the features and create the features importance
- Develop the core functionalities
- Test and Refine
- Develop other secondary features
- Test and Iterate
- Optimize the system
- Final Testing and Release

This methodology has helped in collaboration of each phase of the project like tech stack selection, database setup, and feature development to achieve an efficient and scalable Resume Builder website.

5.1.1. Tools Used

To develop the Resume Builder project, the following tools, technologies, and libraries were used:

Front End:

- **HTML, CSS, and JavaScript:** These frontend technologies are used to build the user interface i.e. client side of the project and used JavaScript to handle interactivity of the project.

Libraries:

- **Bootstrap:** Used frontend frameworks like bootstrap to create responsive design, icons, buttons and pre-styled components.
- **Google Fonts:** Integrated for using customize fonts functionality.
- **Custom Styling**: Applied for components like forms and navigation.

Back End:

- **PHP:** PHP is an open source server side language. In this project PHP is used for handling all server-side logic and for dynamic implementation of all functionalities.
- **Apache server:** Apache Web Server is open source Web server creation, deployment and management software designed to create Web servers that have the ability to host one or more http based websites.

Some PHP scripts (e.g., verification.php) are used for validation and processing.

Database:
MySQL: MySQL is used for storing user data and resume content as it is free, open-source and ideal for handling small as well as larger applications. All the database connections and queries is managed via PHP.

Database Name: resumebuilder

Tables:

- **educations:** Stores user educational background details
- **experiences:** Stores users experience details
- **skills:** Holds user skill data.
- **resumes:** Stores complete detail of the resume created
- **users:** stores user credentials

Code Editor:

- **VS Code:** VS Code is a code editor for building and debugging modern web application and cloud application .Used VS Code for editing the project's frontend and backend code.

Testing Tools:

- **Test Case:** Used Excel, Zephyr Scale for managing and organizing test cases.
- **Issue Reporting:** Jira is used for tracking issues and bugs in the project.
- **Testing:** Used automation tools i.e. Selenium for automating the testing of the project.

Collaboration and versioning Tools:

- Used Git and GitHub to collaborate among the team members.

System Analysis and Designing Tools:

- Used Draw.io, visual paradigm and dbdiagram.io for system designing.

Project Management Tools:

- Used Jira and Microsoft project for project management activities such as project initiation, planning, execution, scheduling and all.

JavaScript Libraries and Plugins:

- **html2canvas:** It is used in this project to capture the screenshot of HTML elements to generate PDFs.
- **JsPDF:** this is a node JS package used for generating downloadable PDF resumes
- **AJAX and jQuery**: JQuery is a JavaScript library where Ajax is a framework which is used in this project for creating a dynamic and responsive user interface through asynchronous data transfer between the client and server without refreshing the page. These were used for handling font changes, background customization, implementing search and providing objective suggestions via a predefined list as a JSON response.

Mailing Libraries:

- **PHPMailer:** It is a PHP package used to send resumes to users via email. This library ensured secure and efficient email delivery through Gmail's mail server.

5.1.2. Implementation Details of Modules

Frontend Module:

This module is responsible for showing the Frontend UI to the users. Technologies like HTML, CSS, Bootstrap, JQuery and JS is used in the implementation of this module.

The landing UI of the system is index.html. This serves as the entry point of the Resume Builder System. It provides the UI elements to navigate to key functionalities like login and registration. The footer, header and navbar of the system is defined inside the includes folder and linked with all the other files. footer.php, header.php and navbar.php are the reusable UI components thought the project files.

Backend Module:

1. **User Registration/Login and Password Reset:**

   In this module implementation of user registration, Password Reset and login along with form and captcha validation is performed. Also implemented hashing of password using password_hash() function before storing it in MySQL database. Used PHPMailer library along with Gmail mail server to implement email based password reset functionality. User requests for password reset which will be validated by a six digit opt code stored in the database.

2. **CRUD Operations on Resume:**

   In this module, Features that allow Users to create, read, update and delete the resume is implemented. Custom PHP logic handles all those CRUD operations and logic along with interaction with database to perform the operations. For alert, session and redirection function.class.php is invoked.

3. **Clone, Print, Share and Download Resume**

   This module handles all other features of the project including cloning the existing resume. Other function of this module including the feature that allow users to print the created resume and share it via WhatsApp through direct link generated by the system.

PDF download features is implemented using html2canvas to capture the resume's HTML structure and converting it to PDF file using jsPDF and allowing users to download it to their local machine.

4. **Objectives Suggestions , Sort and Search Modules**
   We implemented algorithms in this module. For objective suggestion, a predefined set of objectives suggestion is implemented to assist user in creating resume easily. The suggestion is dynamically retrieved through API endpoint based on resume title i.e. job roles like: Web Developer, App Developer and so on. Through implemented search function user can easily locate and find the resume based on the title and contents. SQL queries with LIKE is used to fetch user requested results.

Database Module:

Implemented MySQL database which stores user data, resumes, skills, experiences, and education details. PHP is used to handle SQL queries for all CRUD operations, password resets, download and cloning. Defined database.class.php class to perform database connection using mysqli connect.

5.2. Testing

Testing is one of the phase of software development lifecycle in which we evaluate the software application to identify defects and bugs and to ensure the software application functions as intended.

5.2.1. Test Cases for Unit Testing

Unit testing involves checking each individual components of the system to determine whether the component is running correctly or not.

Table 1: Test case for login and register

S.N	1
Test case id	TC-01
Test Description	Verify registration with valid credentials
Test steps	1. Land on the landing page of the website 2. Click on create account

	3. Provide all the credentials and click on Register button
Test data	Full name: Ram rai Email: ram@gmail.com Password: @RamisGreat123 Captcha: dwztMK
Expected Output	User should be registered and get a welcome email.
Actual Output	As expected
Status	Passed

Test case id	TC-02
Test Description	Verify registration with empty fields
Test steps	1. Land on the landing page of the website 2. Click on create account 3. Without filling the form with all the credentials hit on Register button.
Test data	Full name: Ram rai Email: empty Password: empty Captcha: empty
Expected Output	User should not be able to register and get fill out this field message.
Actual Output	As expected
Status	Passed

S.N	3
Test case id	TC-03
Test Description	Verify login with valid credentials
Test steps	1. Land on the landing page of the website 2. Click on Access Your Resume 3. Provide all the valid credentials.
Test data	Email: ram@gmail.com Password: @RamisGreat123 Captcha: D6f3dy

Expected Output	User should be able to login into their account.
Actual Output	As expected
Status	Passed

S.N	4
Test case id	TC-04
Test Description	Verify login with invalid credentials
Test steps	1. Land on the landing page of the website 2. Click on Access Your Resume 3. Provide invalid password.
Test data	Email: ram@gmail.com Password: @Ram123 Captcha: D6f3dy
Expected Output	User shouldn't be able to login but get Invalid credentials error message
Actual Output	As expected
Status	Passed

S.N	5
Test case id	TC-05
Test Description	Verify login without captcha validation
Test steps	1. Land on the landing page of the website 2. Click on Access Your Resume 3. Provide all valid credentials with invalid captcha and hit on Register button.
Test data	Email: ram@gmail.com Password: @Ram123 Captcha: randomchar
Expected Output	User should not be able to register and get invalid captcha error message
Actual Output	As expected

Status	Passed

Table 5.1: Test case for login and register

Table 2: Test case for CRUD

S.N	6
Test case id	TC-06
Test Description	Verify creating new resume with valid data.
Test steps	1. Land on the landing page of the website. 2. Click on access your resume and enter valid login credentials 3. Click on add new button to create resume.
Test data	Valid credentials
Expected Output	New resume must be created and displayed in the dashboard.
Actual Output	As expected
Status	Passed

S.N	7
Test case id	TC-07
Test Description	Verify updating a resume with valid data.
Test steps	1. Land on the landing page of the website. 2. Click on access your resume and enter valid login credentials 3. Click on edit icon of the resume you want to update.
Test data	Valid credentials.
Expected Output	Resume must be updated and displayed in the dashboard.
Actual Output	As expected
Status	Passed

S.N	8
Test case id	TC-08
Test Description	Verify deleting a resume.
Test steps	1. Land on the landing page of the website. 2. Click on access your resume and enter valid login credentials

	3. Out of the available resume click on delete button of one resume.
Test data	
Expected Output	Resume must be deleted and removed from the dashboard.
Actual Output	As expected
Status	Passed

Table 5.2: Test case for CRUD

5.2.2. Test Cases for System Testing

System Testing is the higher level of testing done to verify the complete system functionality.

Table 3: System Testing Test Cases

S.N	09
Test case id	TC-09
Test Description	Verify end-to-end user registration
Test steps	1. Land on the landing page. 2. Click on create account and provide valid credentials 3. Click on register and verify account creation email
Test data	Full name: Ram Rai Email: ram@gmail.com Password: @RamisGreat123 Captcha: dwztMK
Expected Output	User should be registered successfully, and a welcome email should be received.
Actual Output	As expected
Status	Passed

S.N	10
Test case id	TC-10

Test Description	Verify resume creation workflow
Test steps	1. Login to the system. 2. Click on add new 3. Fill the valid resume details and add. 4. Open the resume and verify.
Test data	Valid credentials
Expected Output	Resume should be created and visible on the dashboard.
Actual Output	As expected
Status	Passed

S.N	11
Test case id	TC-11
Test Description	Verify PDF download functionality
Test steps	1. Login to the system. 2. Open an existing resume. 3. Click "Download” button
Test data	
Expected Output	Resume should download as a properly formatted PDF file.
Actual Output	As expected
Status	Passed

S.N	12
Test case id	TC-12
Test Description	Verify cross-browser compatibility.
Test steps	1. Open the website on Chrome, Firefox, and Edge. 2. Perform user workflows: Register, Login, Create resume.
Test data	Browsers: Chrome (v100+), Firefox (v95+), Edge (v96+)
Expected Output	Application should work consistently across all browsers.
Actual Output	As expected
Status	Passed

S.N	13
Test case id	TC-15
Test Description	Verify invalid login attempts
Test steps	1. Land on the login page. 2. Enter invalid credentials and click "Login".
Test data	Email: ram@gmail.com Password: WrongPassword123 Captcha: D6f3dy
Expected Output	User should see an "Invalid credentials" error.
Actual Output	As expected
Status	Passed

S.N	14
Test case id	TC-18
Test Description	Verify search functionality
Test steps	1. Enter a search term in the search bar. 2. Check the search results.
Test data	Search Term: SEO
Expected Output	All resumes containing the term SEO should be displayed
Actual Output	As expected
Status	Passed

Table 5.3: System Testing Test Cases

5.3 Result Analysis

- The Resume Builder project is successfully implemented and working as expected meeting all the functional and non-functional requirements. Each modules and functionality is implemented and working correctly. Here are the key takeaways of what was achieved:
- All modules including user registration, login, CRUD operations, objective suggestions, resume cloning, share and PDF download, are implemented and working as expected.

- Developed resume builder system has following key features:
  - The system includes features like resume cloning, suggestions, otp based verification and customizable.
  - PDF generation, printing, and WhatsApp sharing.
- The search is fetching the resumes perfectly by using SQL Like queries.
- The objective suggestions is correctly suggesting the predefined objectives based on the resume title.
- The user-friendly UI simplifies navigation and interaction along with multiple device responsiveness.
- Included Google Fonts and background customization which allows users creating personalized resume.
- Unit and system testing helps in validation and verification of the project.
- Selenium, Postman, Jmeter, Jira and Zephyr Scale tools helped in the process of testing and bug tracking.
- Passwords are securely hashed and validated during login, register and reset processes.
- Captcha integration minimizes automated attacks during registration and login.

Chapter 6

Conclusion and Future Recommendation

6.1. Conclusion

To make yourself standout among others in this competitive job market, creating a professional resume plays a vital role. This "resume builder" is an effective way to make yourself standout among others through professional and appealing resumes. Here, user can use simple UI without any manual efforts to create a professional resume that help users to stand out among others. In this resume builder system, we have implemented key features like Resume creation, resume cloning, share via WhatsApp, font customization, background customization, password strength validation and classification, otp based email verification for password reset, Captcha implementation and Role based objectives suggestions.

The main motto of this project is to help users with no designing and technical skills to create a professional resumes in just a few clicks and data input.

6.2. Future Recommendations

Currently this project only contains one standard resume format. There are many scope available for including more templates and format. Some other future scope will be AI integration for resume suggestions based on job roles, profiles or industry. Integrating with job portals to show jobs based on the resume. Implementation of analytics dashboard to show how the resume performs or its impressions.

References

[1] R. Tyagi, A. Sharma, and A. Kapoor, “Automated Resume Generation: Solving the Manual Process Problem,” International Journal of Job Seekers, vol. 5, no. 4, pp. 12-17, 2018.

[2] B. Kungwani et al., “Analytical Resume Builder: Providing Precise Knowledge and Analyzed Stats,” Journal of Web Applications, vol. 7, pp. 210-220, 2020.

[3] M. Marapaka, P. S. Singh, and S. K. Yadav, “Resume Builder Application Vol 8 Issue 3,” International Journal of Software Engineering and Applications, vol. 8, no. 3, pp. 111-120, 2022.

[4] C. Aggarwal, Recommender Systems. New York: Springer, 2018.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms. 3rd ed. Cambridge, MA: MIT Press, 2009.

[6] J. Chew and K. Ong, “A Resume Generator with Augmented Reality Features,” International Journal of Innovation and Technology, vol. 9, pp. 150-160, 2021.

[7] Purdue University, “Resume Design Guide,” Purdue Career Services, [Online]. Available: https://www.purdue.edu/career_services/resumes. [Accessed: Jan. 04, 2025].

[8] I. Wu, “Pro Resume: Focused on Infographic Resumes,” Journal of Online Resume Tools, vol. 4, no. 2, pp. 58-61, 2019.

[9] The Ladders, “Eye-Tracking Study on Resume Screening,” The Ladders, [Online]. Available: https://www.theladders.com/eye-tracking-study. [Accessed: Jan. 04, 2025].

[10] R. Risavy, Resume Research Literature. Journal of Resume Design, vol. 3, pp. 35-42, 2017.

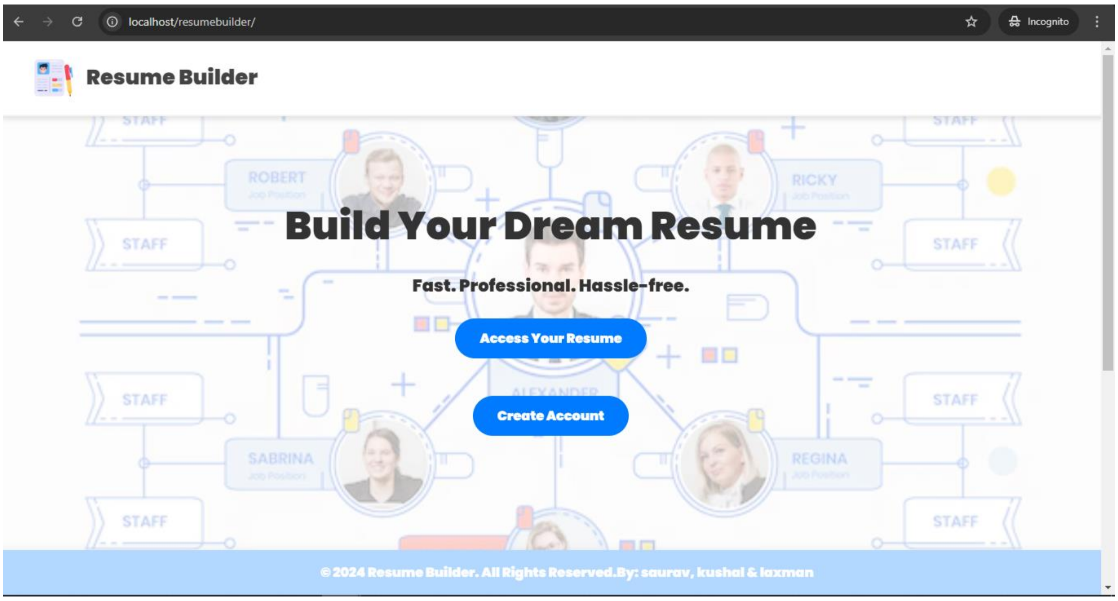
[11] A. Clift, How to Beat ATS Systems: An In-depth Guide. New York: TechPress, 2021.

[12] Purdue University, “Resume Design Guide,” Purdue Career Services, [Online]. Available: https://www.purdue.edu/career_services/resumes. [Accessed: Jan. 04, 2025].

[13] RIT Career Services, “Resume Design Tips,” Rochester Institute of Technology, [Online]. Available: https://www.rit.edu/career-services/resume-design. [Accessed: Jan. 04, 2025].

Appendices

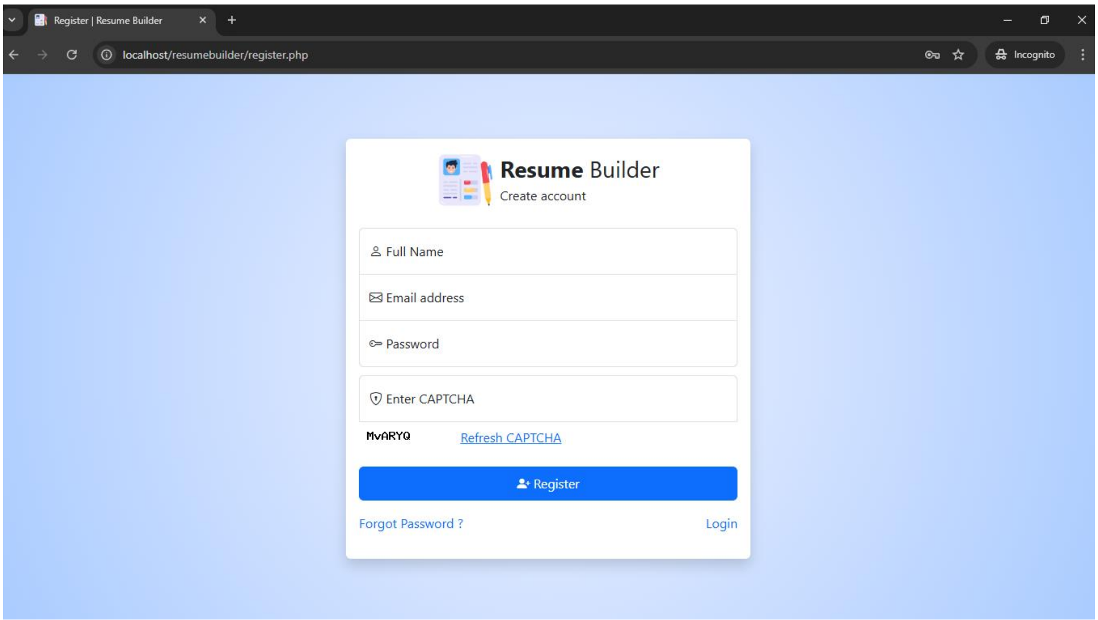
Appendix 1: Screenshots

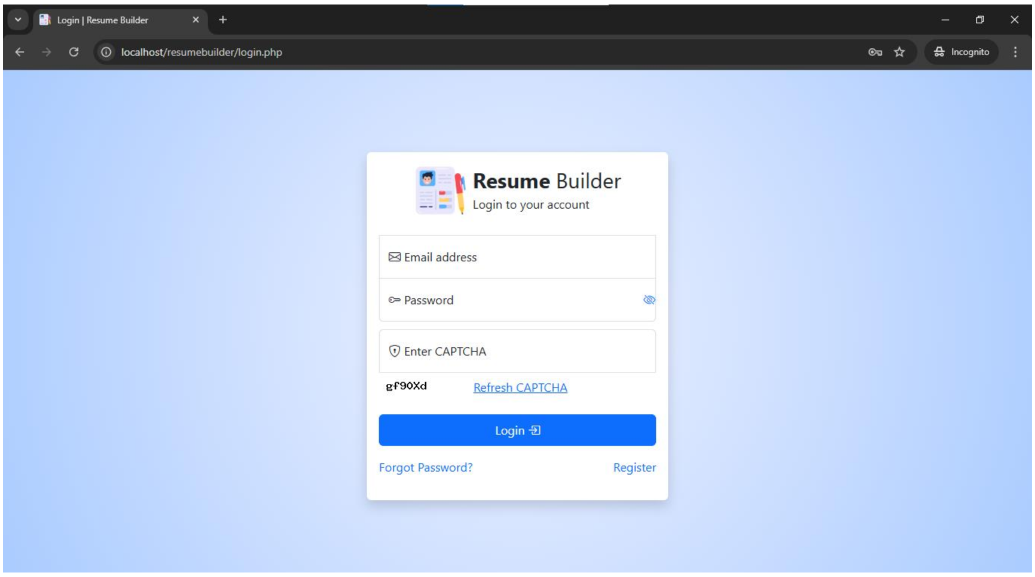


Build Your Dream Resume

Fast. Professional. Hassle-free.

© 2024 Resume Builder. All Rights Reserved.By: saurav, kushal & laxman





Resume Builder
Login to your account

Email address

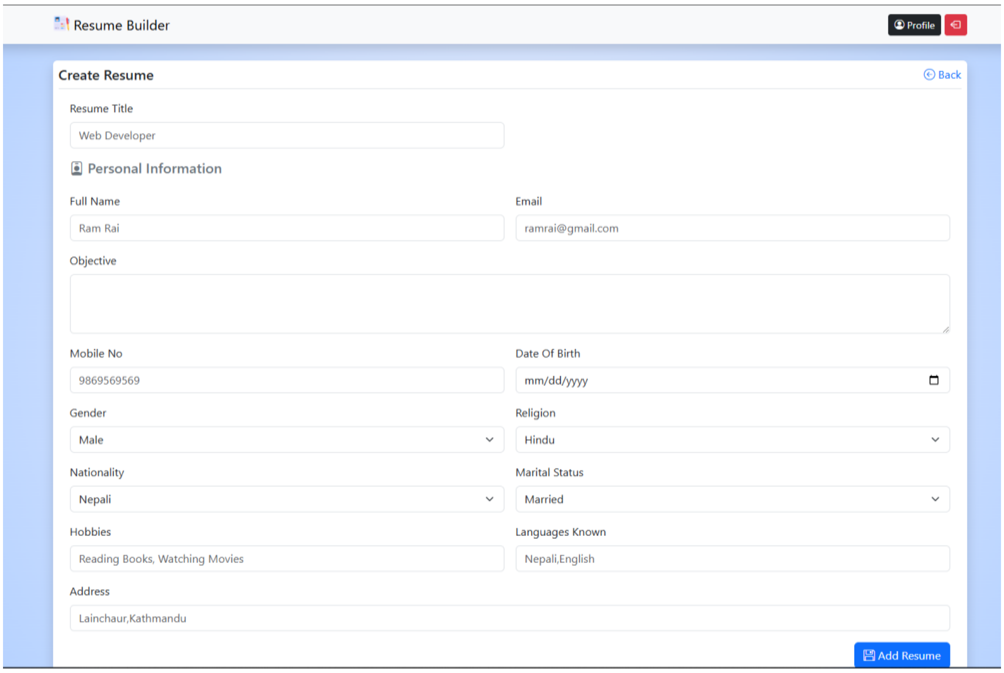
Password

Enter CAPTCHA

gf90Xd Refresh CAPTCHA

Login

Forgot Password? Register



Create Resume

Resume Title: Web Developer

Personal Information

Full Name: Ram Rai

Email: ramrai@gmail.com

Objective:

Mobile No: 9869569569

Date Of Birth: mm/dd/yyyy

Gender: Male

Religion: Hindu

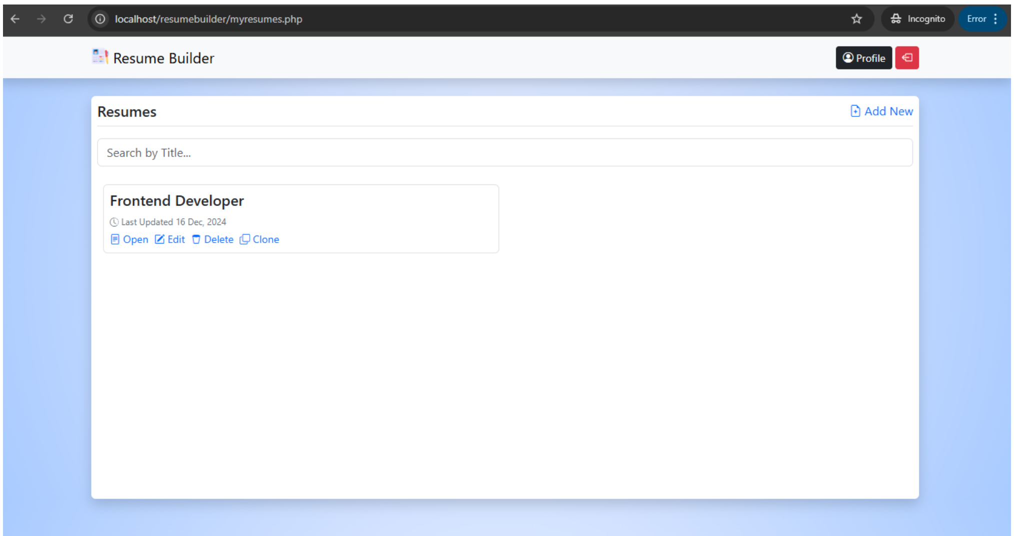
Nationality: Nepali

Marital Status: Married

Hobbies: Reading Books, Watching Movies

Languages Known: Nepali,English

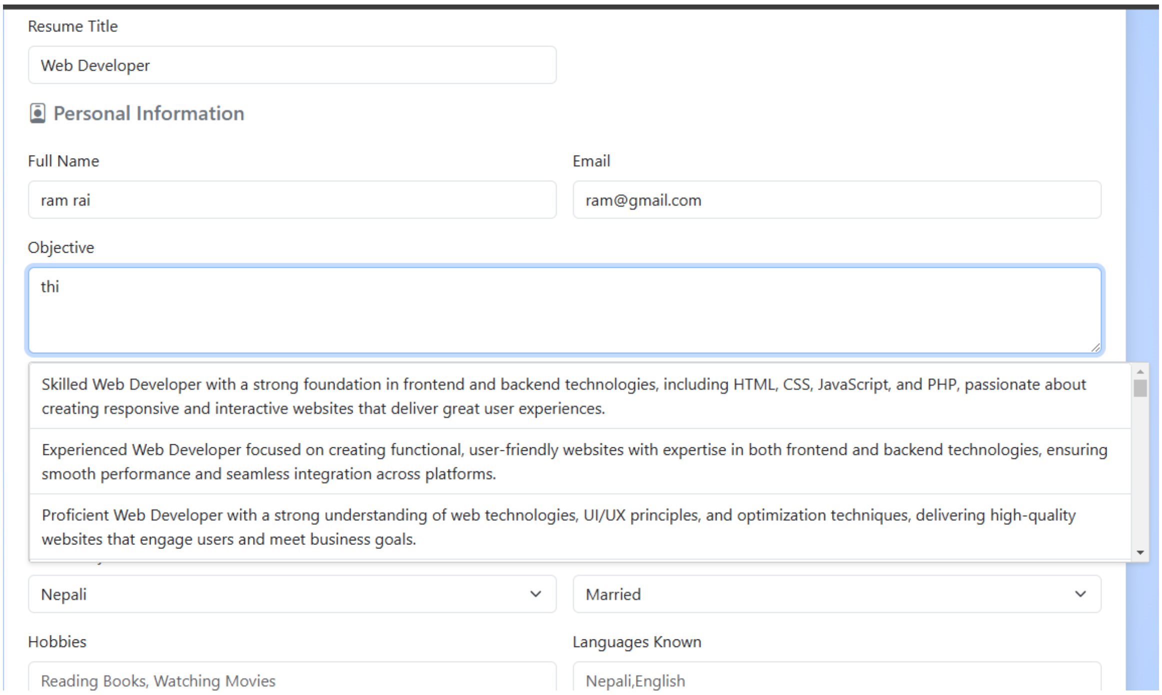
Address: Lainchaur,Kathmandu



Resumes

Search by Title...

Frontend Developer
Last Updated 16 Dec, 2024



Resume Title

Web Developer

Personal Information

Full Name

ram rai

Email

ram@gmail.com

Objective

thi

Skilled Web Developer with a strong foundation in frontend and backend technologies, including HTML, CSS, JavaScript, and PHP, passionate about creating responsive and interactive websites that deliver great user experiences.

Experienced Web Developer focused on creating functional, user-friendly websites with expertise in both frontend and backend technologies, ensuring smooth performance and seamless integration across platforms.

Proficient Web Developer with a strong understanding of web technologies, UI/UX principles, and optimization techniques, delivering high-quality websites that engage users and meet business goals.

Nepali

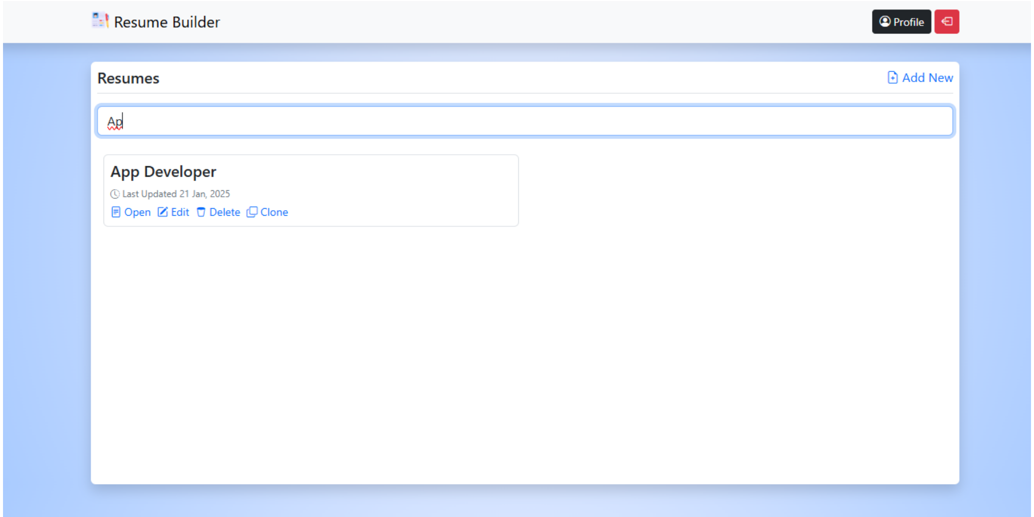
Married

Hobbies

Reading Books, Watching Movies

Languages Known

Nepali,English

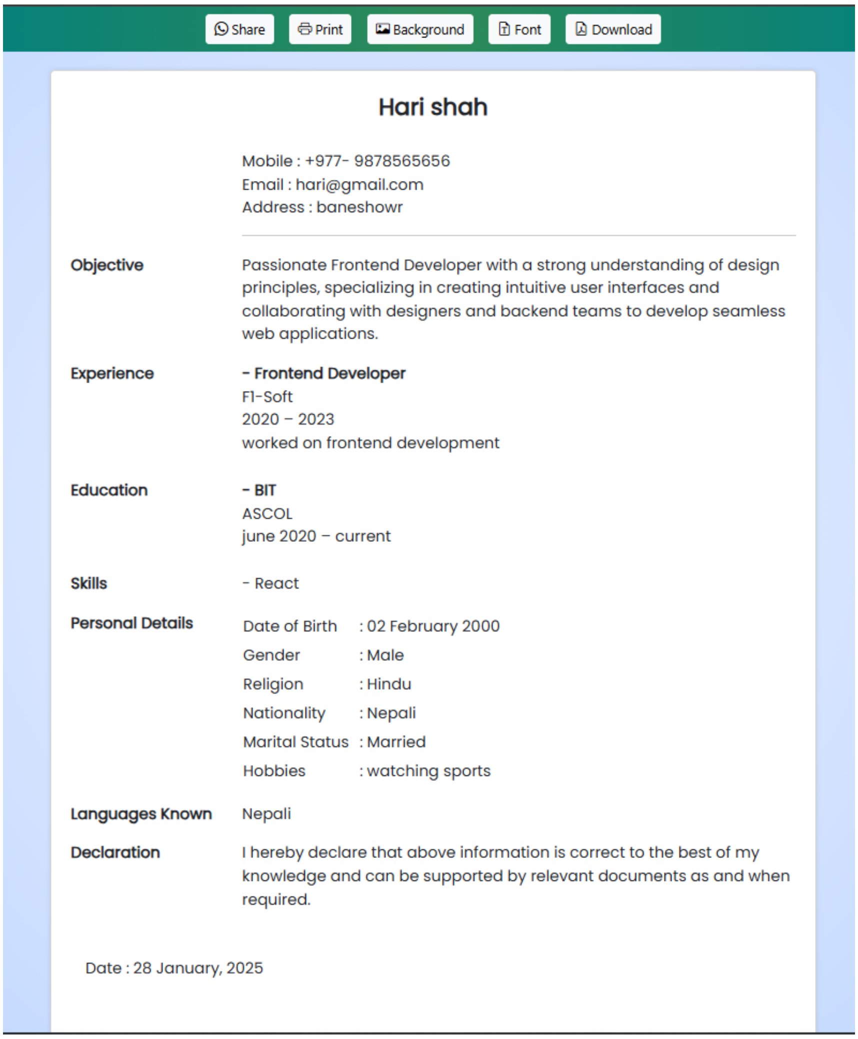


Resumes

Ap

App Developer

Last Updated 21 Jan, 2025



Hari shah

Mobile : +977- 9878565656
Email : hari@gmail.com
Address : baneshowr

Objective

Passionate Frontend Developer with a strong understanding of design principles, specializing in creating intuitive user interfaces and collaborating with designers and backend teams to develop seamless web applications.

Experience

- Frontend Developer
F1-Soft
2020 – 2023
worked on frontend development

Education

- BIT
ASCOL
june 2020 – current

Skills

- React

Personal Details

Date of Birth	: 02 February 2000
Gender	: Male
Religion	: Hindu
Nationality	: Nepali
Marital Status	: Married
Hobbies	: watching sports

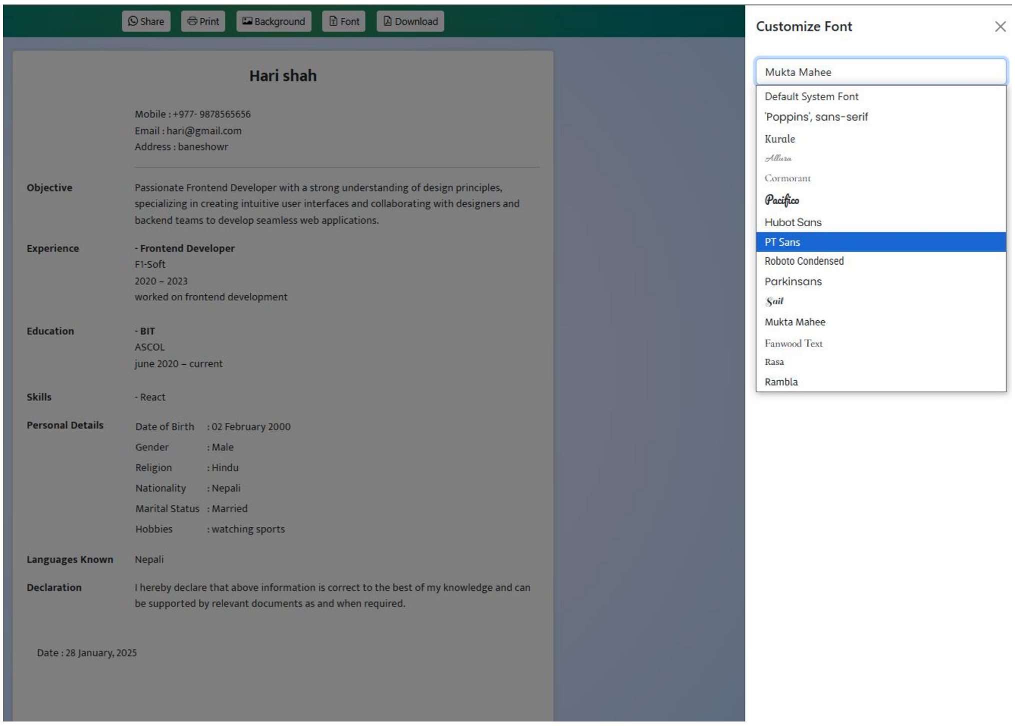
Languages Known

Nepali

Declaration

I hereby declare that above information is correct to the best of my knowledge and can be supported by relevant documents as and when required.

Date : 28 January, 2025



Hari shah

Mobile : +977- 9878565656
Email : hari@gmail.com
Address : baneshowr

Objective

Passionate Frontend Developer with a strong understanding of design principles, specializing in creating intuitive user interfaces and collaborating with designers and backend teams to develop seamless web applications.

Experience

- Frontend Developer
F1-Soft
2020 – 2023
worked on frontend development

Education

- BIT
ASCOL
june 2020 – current

Skills

- React

Personal Details

Date of Birth	: 02 February 2000
Gender	: Male
Religion	: Hindu
Nationality	: Nepali
Marital Status	: Married
Hobbies	: watching sports

Languages Known

Nepali

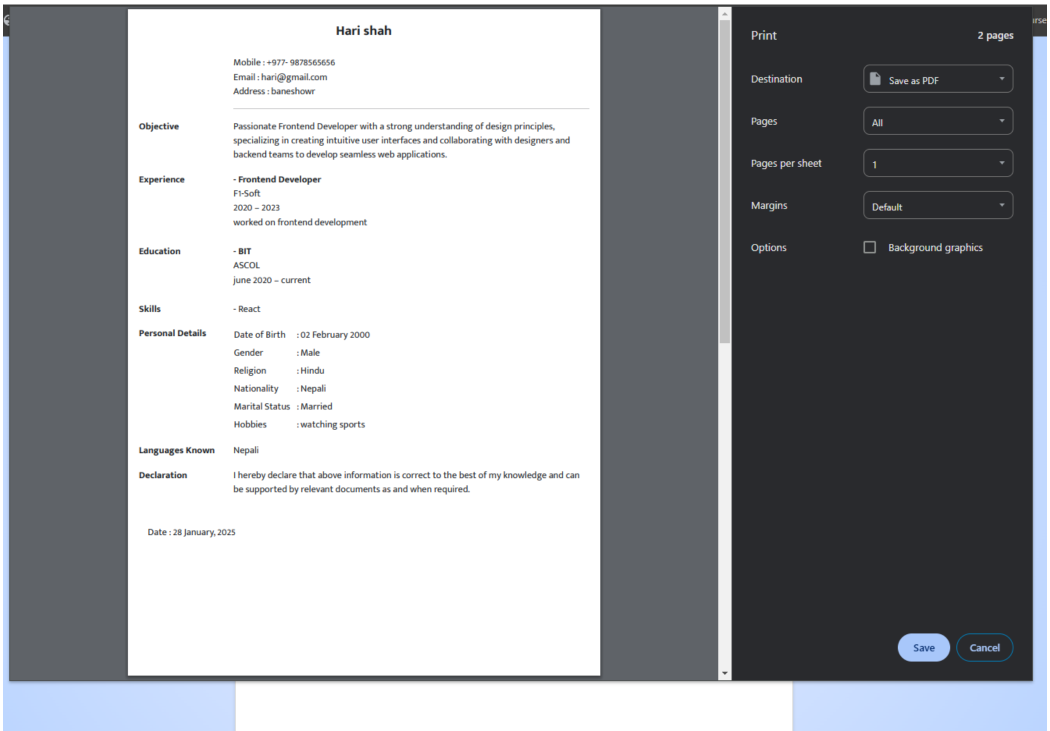
Declaration

I hereby declare that above information is correct to the best of my knowledge and can be supported by relevant documents as and when required.

Date : 28 January, 2025

Customize Font

Mukta Mahee



Hari shah

Mobile : +977- 9878565656
Email : hari@gmail.com
Address : baneshowr

Objective

Passionate Frontend Developer with a strong understanding of design principles, specializing in creating intuitive user interfaces and collaborating with designers and backend teams to develop seamless web applications.

Experience

- Frontend Developer
F1-Soft
2020 – 2023
worked on frontend development

Education

- BIT
ASCOL
june 2020 – current

Skills

- React

Personal Details

Date of Birth	: 02 February 2000
Gender	: Male
Religion	: Hindu
Nationality	: Nepali
Marital Status	: Married
Hobbies	: watching sports

Languages Known

Nepali

Declaration

I hereby declare that above information is correct to the best of my knowledge and can be supported by relevant documents as and when required.

Date : 28 January, 2025

Print — 2 pages

Destination	Save as PDF
Pages	All
Pages per sheet	1
Margins	Default
Options	□ Background graphics

Appendix 2: Source Code

Createresume.action.php

```
<?php
if ($_POST) {
    $post = $_POST;
    if (
        !empty($post['full_name']) &&
        !empty($post['email_id']) &&
        !empty($post['objective']) &&
        !empty($post['mobile_no']) &&
        !empty($post['dob']) &&
        !empty($post['religion']) &&
        !empty($post['nationality']) &&
        !empty($post['marital_status']) &&
        !empty($post['hobbies']) &&
        !empty($post['languages']) &&
        !empty($post['address'])
    ) {
        $columns = '';
        $values = '';
        $slug = $fn->randomstring(); // Generate unique slug
        $authid = $fn->Auth()['id'];
        // Check for duplicate resume title for the user
        $resumeTitle = $db->real_escape_string($post['resume_title']);
        $checkQuery = "SELECT id FROM resumes WHERE resume_title = '$resumeTitle' AND
user_id = $authid";
        $result = $db->query($checkQuery);

        if ($result->num_rows > 0) {
            $fn->setError('A resume with this title already exists. Please use a different title.');
            $fn->redirect('../createresume.php');
            exit;
        }
        // Prepare columns and values for insertion
        foreach ($post as $index => $value) {
            $value = $db->real_escape_string($value);
            $columns .= $index . ',';
            $values .= "'$value',";
        }
        // Add slug, updated_at, and user_id to the columns and values
        $columns .= 'slug, updated_at, user_id';
        $values .= "'$slug', NOW(), $authid";
        try {
            // Insert resume into database
            $query = "INSERT INTO resumes ($columns) VALUES ($values)";
            if ($db->query($query)) {
                $fn->setAlert('Resume Added!');
                $fn->redirect('../myresumes.php');
                exit;
            } else {
                throw new Exception('Database insertion failed: ' . $db->error);
            }
        } catch (Exception $error) {
            // Log and display error
            error_log($error->getMessage()); // Log to server error log
            $fn->setError('Error adding resume: ' . $error->getMessage());
            $fn->redirect('../createresume.php');
            exit;
        }
    } else {
        $fn->setError('Please fill all required fields!');
        $fn->redirect('../createresume.php');
        exit;
    }
} else {
    $fn->redirect('../createresume.php');
    exit;
}
?>
```

```
// JavaScript for objective suggestions
document.getElementById('objective-field').addEventListener('input', function() {
    const objective = this.value.trim();
    const resumeTitle = document.querySelector('input[name="resume_title"]').value.trim(); // Get resume title
    const suggestionsList = document.getElementById('suggestions-list');
    if (objective.length > 2) {
        // Make an AJAX request to fetch suggestions
   fetch(`actions/objective_suggestions.php?objective=${objective}&title=${resumeTitle}`)
            .then(response => response.json())
            .then(data => {
                suggestionsList.innerHTML = ''; // Clear previous suggestions
                if (data.suggestions.length > 0) {
                    data.suggestions.forEach(suggestion => {
                        const listItem = document.createElement('li');
                        listItem.classList.add('list-group-item');
                        listItem.textContent = suggestion;
                        listItem.onclick = () => {
                            document.getElementById('objective-field').value = suggestion;
                            suggestionsList.style.display = 'none';
                        };
                        suggestionsList.appendChild(listItem);});
                    suggestionsList.style.display = 'block'; // Show the suggestion list
                } else {
                    suggestionsList.style.display = 'none';
                }})
            .catch(err => console.error('Error fetching suggestions:', err));
    } else {
        suggestionsList.style.display = 'none'; // Hide suggestions if input is short
    }
});
```

```
// Hide suggestions when clicking outside

document.addEventListener('click', function(event) {

    const suggestionsList = document.getElementById('suggestions-list');

    if (!suggestionsList.contains(event.target) && event.target !== document.getElementById('objective-field')) {

        suggestionsList.style.display = 'none';

    }

});

</script>
```