TABLE OF CONTENTS

CHAPTER NO	TOPIC	PAGE NO
	ABSTRACT	**1**
1	**INTRODUCTION**	**2**
2	**SYSTEM REQUIREMENTS** 2.1 Hardware Requirements 2.2 Software Requirements	**4**
3	**SYSTEM ANALYSIS** 3.1 Existing System 3.2 Proposed System 3.3 Software Description 3.4 Module Description	**5**
4	**SYSTEM DESIGN** 4.1 Architectural Design 4.2 Data Flow Diagram 4.3 Database Design	**10**
5	**SYSTEM IMPLEMENTATION**	**20**
6	**TESTING** 6.1 Unit Testing 6.2 System Testing 6.3 Validation Testing	**47**
7	**CONCLUSION**	**51**
8	**FUTURE SCOPE**	**52**
9	**REFERENCE**	**53**
10	**APPENDIX** 10.1 Sample Source code	**54**

ABSTRACT

In the modern recruitment process, Applicant Tracking Systems (ATS) play a crucial role in filtering resumes before they reach human recruiters. This project presents a Resume Builder integrated with ATS optimization. The system features form-based data entry, template selection, and PDF export functionality, providing an interactive and guided resume. It incorporates data validation, session handling, and optionally includes user authentication for saving and editing resumes later. By automating the resume creation process, this project not only facilitates the creation of professional and structured resumes but also educates users on best practices for passing ATS filters, ultimately increasing their chances of securing interviews in competitive job markets. Restrictions are also imposed to ensure data accuracy and relevance, including limits on word count for each section, controlled vocabulary for skills, and mandatory fields to avoid incomplete submissions. This project is developed using technologies such as HTML, CSS, JavaScript, and backend support with PHP/MySQL.

CHAPTER 1

INTRODUCTION

In today's competitive job market, the recruitment process has evolved significantly with the integration of advanced technologies. Most organizations now rely on **Applicant Tracking Systems (ATS)** to efficiently manage, analyze, and filter large volumes of job applications. These systems automatically **scan, parse, and rank resumes** based on predefined criteria such as keyword relevance, structure, and formatting accuracy. While ATS software streamlines the hiring process for recruiters, it also presents a major challenge for applicants—many qualified candidates fail to pass automated screenings due to **non-ATS-compliant resume designs, missing keywords, or inconsistent formatting**. Consequently, a significant number of resumes are filtered out before reaching human recruiters, reducing candidates' chances of selection.

To address this challenge, this project presents an **ATS-Based Resume Builder**, a web application designed to simplify and optimize resume creation while ensuring compatibility with modern recruitment systems. The platform allows users to input their personal, educational, and professional information through an intuitive form interface and automatically generates a **machine-readable, ATS-optimized resume**. Unlike traditional resume builders that focus solely on design, this system integrates **AI-powered resume analysis using the Google AI Studio API**, which evaluates keyword density, structure, and relevance against a provided job description. This ensures that users receive an accurate **ATS score and feedback**, helping them improve their resumes for higher recruiter visibility.

The system is developed using **PHP** and **MySQL** for backend functionality, ensuring secure data handling, efficient storage, and reliable performance. The frontend is built using **HTML5, CSS3, Bootstrap, and JavaScript**, offering a responsive, interactive, and user-friendly interface.

The project operates through two primary modules:

- **User Module** – Allows job seekers to register, login, and manage their profiles. Users can enter details, choose templates, preview resumes, check ATS compatibility, and download the final version in PDF format. The module provides real-time validation and AI-assisted keyword optimization for better results.

- **Admin Module** – Provides administrators with full control over the system. Admins can manage users, upload or edit resume templates, and ensure the secure operation of the application. The admin panel also supports adding premium templates and maintaining system updates.

By integrating ATS optimization with AI-driven analysis and a user-friendly interface, this project bridges the gap between candidates and modern recruitment technologies. Overall, the **RESUME BUILDER BASED ON APPLICANT TRACKING SYSTEM** offers a reliable, scalable, and intelligent solution that streamlines the resume creation process and improves candidate visibility.

CHAPTER 2
SYSTEM REQUIREMENTS

2.1 Hardware Requirements

- Processor : Intel Core i3
- RAM : Minimum 4 GB

2.2 Software Requirements

- Operating System : Windows 11 (64-bit)
- Frontend Technologies : HTML5, CSS3, JavaScript
- Script Language : PHP
- Database : MySQL

CHAPTER 3
SYSTEM ANALYSIS

3.1 Existing System

1. **Visual-centric design**: Most existing resume builders prioritize aesthetics and visual layouts over the structured formatting required for Applicant Tracking Systems (ATS).

2. **Poor ATS compatibility**: They often create resumes that are difficult for ATS to parse, leading to data extraction errors and a higher chance of the resume being rejected before a human even sees it.

3. **Lack of optimization**: These builders generally don't help users optimize their resumes for **keyword matching**, which is critical for ranking high in ATS searches by recruiters.

3.2 Proposed System

1. **Structured, form-based input**: The system uses a form to guide users through the resume creation process, ensuring a consistent and logical data structure.

2. **ATS-friendly output**: The resume builder generates a clean, **ATS-friendly PDF** output that is easily readable by parsing systems, which increases the likelihood of a candidate being screened.

Key features:

- **Template selection**: Users can choose from a variety of professional templates designed for both visual appeal and ATS compatibility.
- **Live preview**: A real-time preview allows users to see how their data will look on the final resume.
- **Reusable data**: The system stores user data so it can be easily updated and reused for creating different versions of a resume or for future job applications.

3.3 Software Description

HTML5: The foundation of the web application, HTML5 is used to create the structure of all the web pages, including the resume builder form, the user dashboard, and the templates page. It provides the semantic tags needed to organize content like forms, headers, and navigation menus, which improves accessibility and is crucial for the ATS optimization of the final resume.

CSS3: This is the styling language that gives the resume builder its modern and professional appearance. It's responsible for the visual layout, colors, fonts, and responsive design, ensuring the user interface works well on different devices like desktops and phones. CSS3 features like animations and grid layouts are used to make the platform interactive and visually appealing.

JavaScript: This client-side scripting language adds interactivity and dynamic functionality. JavaScript handles actions like live preview, form validation to ensure correct data entry, and other real-time updates without requiring a full page reload. It enhances the user experience by making the platform faster and more responsive.

PHP: As the core server-side scripting language, PHP manages the project's business logic. It handles user authentication (login/logout), manages resume data, and connects the front-end with the MySQL database to perform CRUD (Create, Read, Update, Delete) operations. It's essential for creating and managing user sessions and for the secure storage of data.

MySQL: This is the relational database management system used to store and manage all the application's data. MySQL tables are designed to store information on users, templates, and the created resumes themselves, ensuring data consistency and security. Its seamless integration with PHP allows for efficient data retrieval and storage.

Python: Python is used specifically for the ATS scoring system. It can be integrated into the system's backend to analyze resumes, compare them against job descriptions, and provide a compatibility score. This is a key feature that helps users optimize their resumes for keyword matching and structured readability, which is often a significant weakness in other resume builders. The use of Python with a library like Google AI Studio enables the ATS checker to provide an automated evaluation of the resume's effectiveness.

3.4 MODULE DESCRIPTION

This process contains the following main modules:

- ADMIN MODULE
- USER MODULE

1. Admin Module

- **Admin Dashboard:**
  The dashboard provides a comprehensive overview of the platform, including the total number of users, available templates, and created resumes.

- **Template Management:**
  Admins can manage resume templates by adding, editing, or deleting them. They can also designate templates as ATS-friendly or premium.

- **User Management:**
  The admin can view all registered users and their details, including the number of resumes they have created. They can also view and delete user accounts.

- **System Monitoring:**
  The admin monitors resume activity, including the history of downloads.

2. User Module

- **Secure Login & Profile Management:**
  Users can securely log in to their accounts and manage their personal information.

- **Resume Management:**
  Users can create, edit, and delete their resumes. They can also view their past resumes.

- **Template Selection & Customization:**
  Users can select from a range of professional templates to create their resumes. They can also purchase premium templates.

- **ATS Checker:**
  A key feature that allows users to upload their resumes and paste a job description to receive an instant ATS (Applicant Tracking System) score. This helps them optimize their resume for keyword matching and readability.

- **PDF Generation:**
  The system allows users to generate an ATS-friendly resume in PDF format instantly.

3. Connected Functionalities

- **Authentication & Authorization:**
  The system provides secure login functionality for both users and admins. It also includes features for user registration and password management.

- **Data Storage:**
  User data, templates, and created resumes are securely stored in a MySQL database. This allows users to save and edit their resumes at a later time.

- **Payment:**
  The system includes a payment process for purchasing premium templates, with options like UPI and credit/debit cards.

- **System Analytics:**
  The system provides analytics, such as the total number of resumes created and an ATS success rate, to help users track their progress.

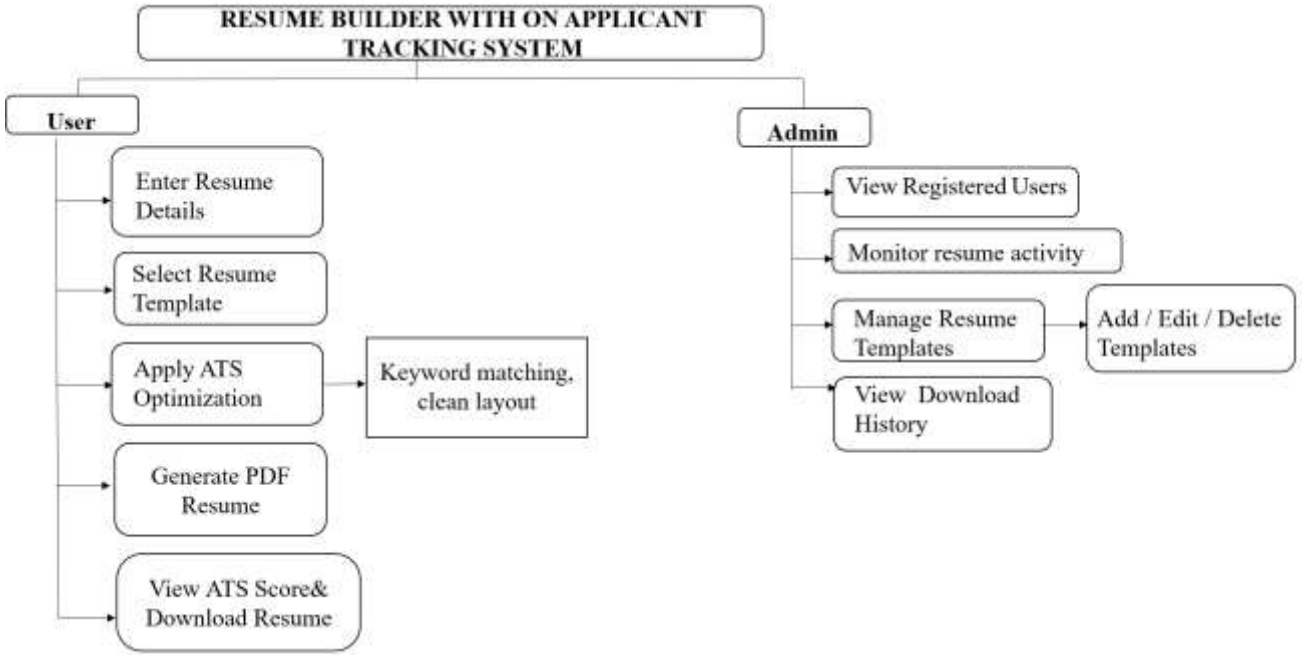
CHAPTER 4
SYSTEM DESIGN

System design is to translate user needs into a structured framework that will drive development. It specifies how various modules interact and assures that the system is functional, scalable, and efficient. Architectural modeling, process flow diagrams, and database schema preparation are all part of the system design process for this project.

4.1 Architectural design

Architectural design creates a high-level blueprint of the software, displaying its essential components and interactions. It makes it easier to see how data goes through the system, how modules communicate, and what boundaries exist between different layers. This acts as the foundation for implementation and future system expansion.

ARCHITECTURAL DESIGN



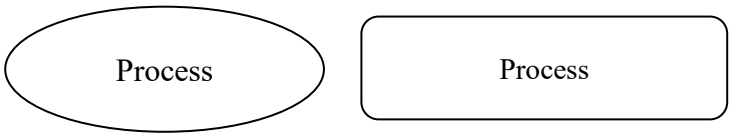
4.2 DATAFLOW DIAGRAM

A data flow diagram (DFD) is a visual representation that illustrates how data moves and transforms within a system. The network's nodes indicate points where data is processed or altered. These nodes mainly perform functions such as combining, splitting, or modifying data streams. Unlike traditional flowcharts, the nodes in a DFD can be activated or triggered. DFDs can also be expressed using informational notation and, similar to other system flowcharts, can be developed with varying levels of detail for deeper or more simplified analysis.

Key Symbols

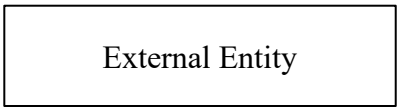
1. Process

A process is depicted using a circle, oval, or sometimes a rectangle with rounded edges. This symbol represents an operation that processes or transforms data. For instance, a process might perform actions like **"Total Price"** or **"Login."** It receives input data, processes it, and generates output data as a result.



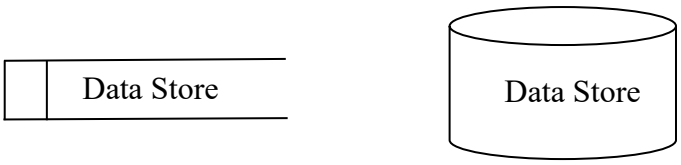
2. External Entity

An external entity, also known as a terminator, is shown using a square or rectangle. It represents a person, organization, or system that exists outside the boundaries of the system under study. These entities act as the sources from which data enters the system and the destinations to which data is sent. They do not carry out any processing within the system. Examples include **"User"** or **"Admin."**



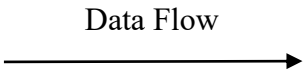
3. Data Store

A data store is depicted using a cylinder or two parallel horizontal lines. This element signifies a location where information is kept for future reference. It serves as a storage point for data, which could be in the form of a database, a document, or any other record system. Data can either be added to the store (data input) or retrieved from it (data output).



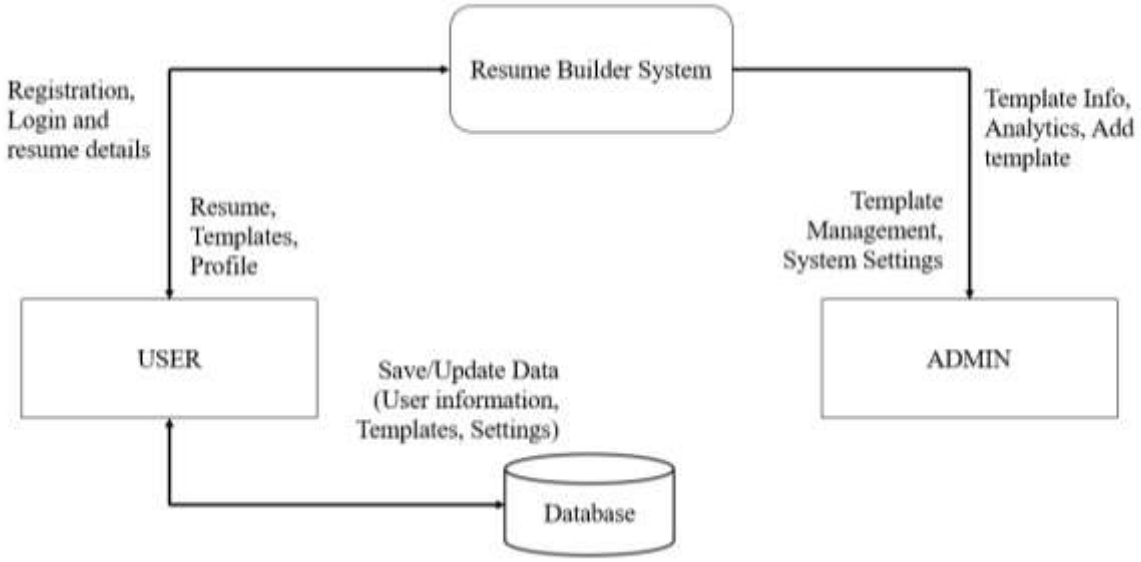
4. Data Flow

A data flow is illustrated with a labeled arrow that depicts the transfer of information between different components of the system. The label on the arrow identifies the type of data being transmitted. For instance, an arrow marked **"Order Information"** could indicate data moving from an **external source** (such as a **User**) to a **process** (like **Submit Order**).

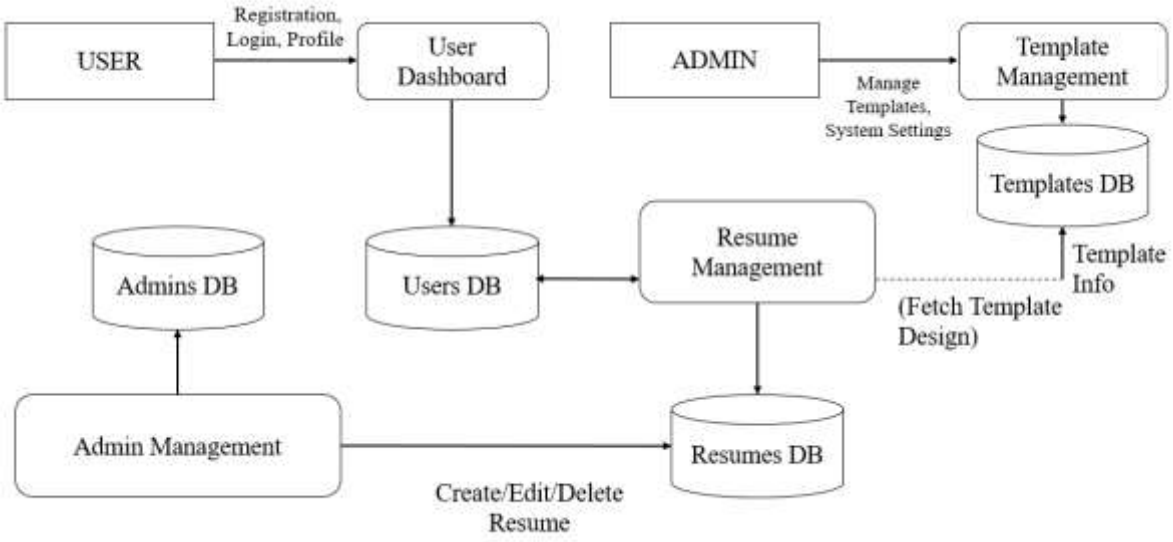


DATAFLOW DIAGRAM

LEVEL 0

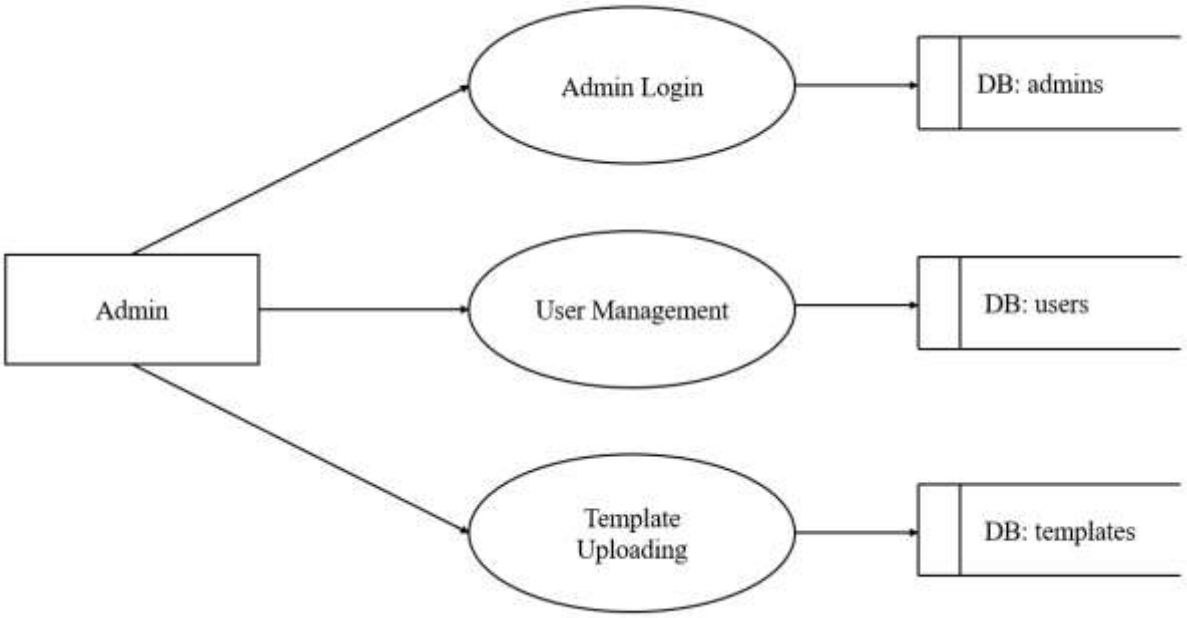


LEVEL 1



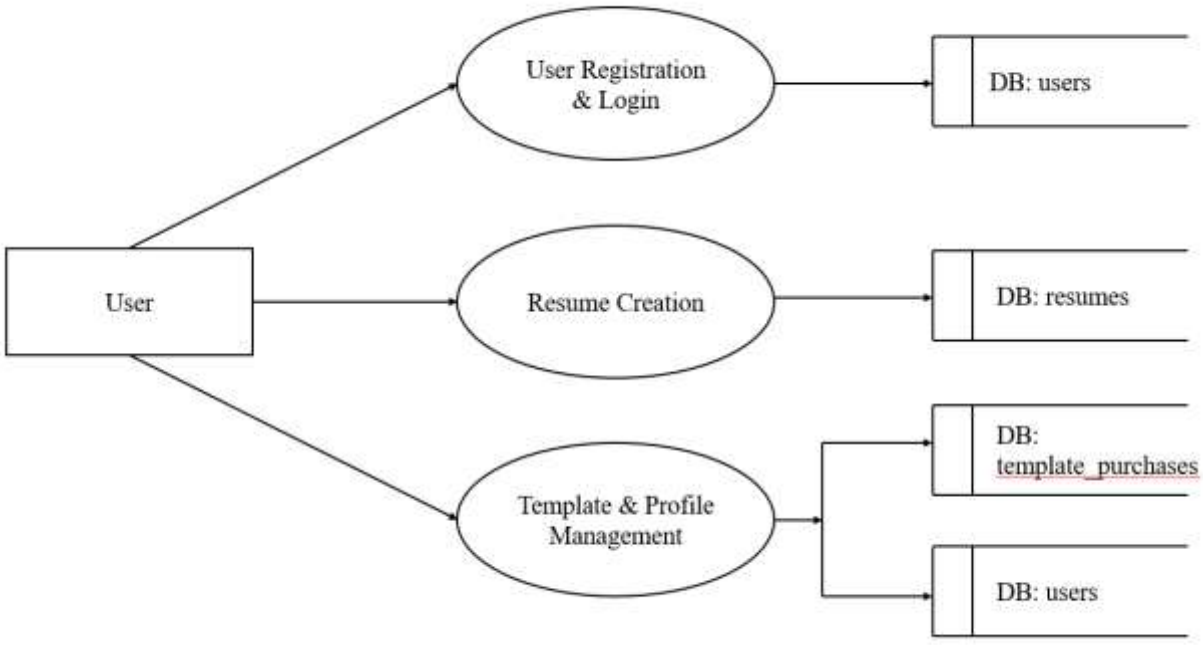
LEVEL 2

ADMIN



LEVEL 2

USER



4.3 DATABASE DESIGN

Table: users

Purpose: Stores information about people who register to use the resume builder.

Field	Type	Constraints	Description
id	int(11)	Primary Key, Auto Increment	Unique identifier for each user account.
username	varchar(50)	NOT NULL, UNIQUE	The user's chosen username for login.
email	varchar(100)	NOT NULL, UNIQUE	The user's email address, used for login and communication.
first_name	varchar(50)	NOT NULL	The user's first name.
last_name	varchar(50)	NOT NULL	The user's last name.
phone	varchar(20)	NULL	The user's phone number.
profile_image	varchar(255)	NULL	File path to the user's profile image.
password	varchar(255)	NOT NULL	The hashed password for the user's account.
created_at	timestamp	DEFAULT current_timestamp()	Timestamp of when the user account was created.

Table: admin

Purpose: Stores admin users who manage the system.

Column	Type	Constraints	Description
id	int(11)	Primary Key, Auto Increment	Unique identifier for each admin account.
username	varchar(50)	NOT NULL, UNIQUE	The admin's chosen username for login.
email	varchar(100)	NOT NULL, UNIQUE	The admin's email address.
password	varchar(255)	NOT NULL	The hashed password for the admin account.
created_at	timestamp	DEFAULT current_timestamp()	Timestamp of when the admin account was created.

Table: resumes

Purpose: Stores predefined resume templates that users can choose from.

Field	Type	Constraints	Description
id	int(11)	Primary Key, Auto Increment	A unique identifier for each resume entry.
user_id	int(11)	Foreign Key, NOT NULL, ON DELETE CASCADE	A foreign key referencing the ID in the users table, linking the resume to a specific user.
full_name	varchar(255)	NULL	The full name as it appears on the resume.
professional_title	varchar(255)	NULL	The user's professional title.
email	varchar(255)	NOT NULL	The email address associated with the resume.
phone	varchar(50)	NULL	The user's contact phone number.
website	varchar(255)	NULL	The user's personal website or portfolio link.
profile_image	varchar(255)	NULL	File path to the user's profile picture for the resume.
address	text	NULL	The user's physical address.
objective	text	NULL	The resume's career objective or summary.
education	text	NULL	The user's educational background.
experience	text	NULL	Details about the user's professional work experience.
achievements	text	NULL	A list of professional achievements.
certifications	text	NULL	A list of certifications the user has obtained.
skills	text	NULL	The user's skills.
projects	text	NULL	A list of personal or professional projects.
languages	text	NULL	The languages the user is proficient in.

references_text	text	NULL	A text field for professional references.
template_id	int(11)	DEFAULT 1	The ID of the template used for the resume.
title	varchar(100)	NULL	The title given to the resume by the user.
content	text	NULL	A text field that can store resume content, possibly in JSON format.
created_at	timestamp	DEFAULT current_timestamp()	Timestamp of when the resume was first created.
updated_at	timestamp	DEFAULT current_timestamp() ON UPDATE current_timestamp()	Timestamp of the last time the resume was updated.
linkedin	varchar(255)	NULL	The user's LinkedIn profile URL.
twitter	varchar(255)	NULL	The user's Twitter handle.
hobbies	text	NULL	A field for listing the user's hobbies.
template_type	varchar(50)	DEFAULT 'theme1'	The type of template used, such as 'theme1'.

Table: templates

Purpose: Stores the actual resumes created by users.

Field	Type	Constraints	Description
id	int(11)	Primary Key, Auto Increment	A unique identifier for each template.
name	varchar(255)	NOT NULL	The name of the template.
description	text	NULL	A description of the template's design and features.
category	varchar(100)	NULL	The category the template belongs to (e.g., 'Professional', 'Minimalist').
thumbnail	varchar(255)	NULL	The file path to the template's thumbnail image.

file_path	varchar(255)	NOT NULL	The file path to the main template file.
php_file	varchar(255)	NULL	The file path to the PHP file for the template.
html_file	varchar(255)	NULL	The file path to the HTML file for the template.
css_file	varchar(255)	NULL	The file path to the CSS file for the template.
css_content	text	NULL	The CSS content for the template.
js_file	varchar(255)	NULL	The file path to the JavaScript file for the template.
js_content	text	NULL	The JavaScript content for the template.
theme_file	varchar(255)	NULL	The file path to the theme file for the template.
is_ats_friendly	tinyint(1)	DEFAULT 1	A flag indicating if the template is ATS (Applicant Tracking System) friendly.
is_premium	tinyint(1)	DEFAULT 0	A flag indicating if the template is a premium (paid) template.
price	decimal(10,2)	DEFAULT 0.00	The price of the premium template.
deleted	tinyint(1)	DEFAULT 0	A flag to mark the template for soft deletion.
created_by	int(11)	Foreign Key, NULL, ON DELETE SET NULL	The ID of the user or admin who created the template.
created_at	timestamp	DEFAULT current_timestamp()	The timestamp of when the template was created.
updated_at	timestamp	DEFAULT current_timestamp() ON UPDATE current_timestamp()	The timestamp of the last time the template was updated.

Table: templates purchase

Purpose: This table is to record and track all transactions where a user buys a premium resume template.

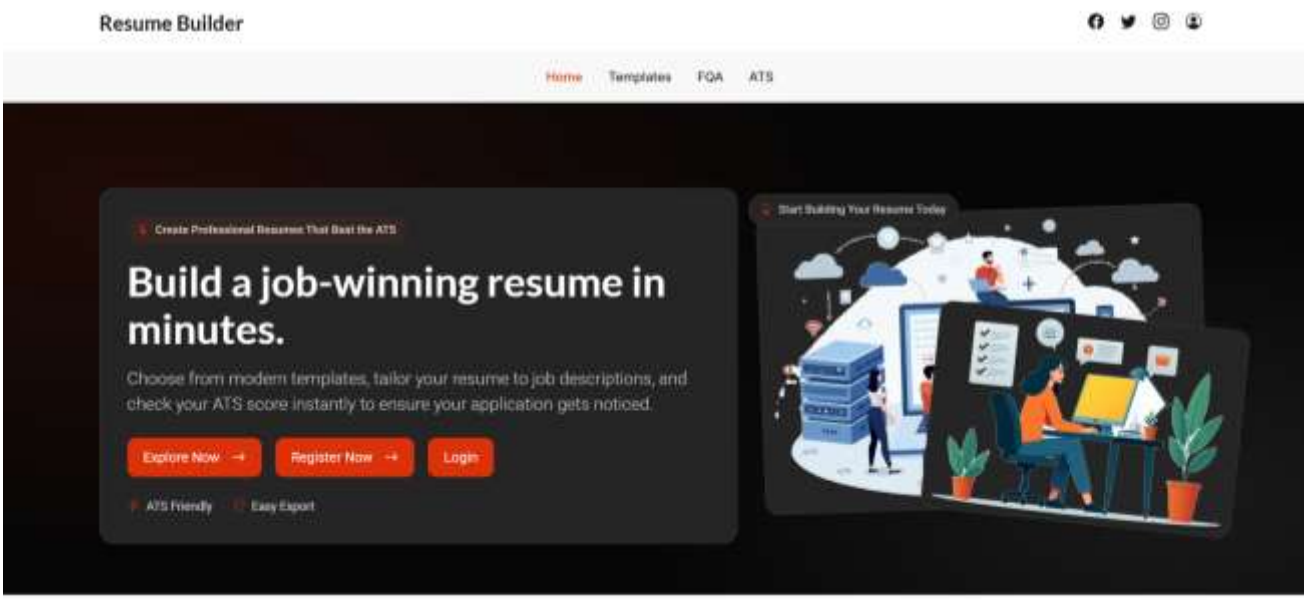
Field	Type	Constraints	Description
id	int(11)	Primary Key, Auto Increment	Unique identifier for each purchase transaction.
user_id	int(11)	Foreign Key, NOT NULL, ON DELETE CASCADE	Links the purchase to the user who made it.
template_id	int(11)	Foreign Key, NOT NULL, ON DELETE CASCADE	Links the purchase to the specific template that was bought.
amount	decimal(10,2)	NOT NULL	The price paid for the template.
payment_ method	varchar(50)	NOT NULL	The method used for the payment
transaction_id	varchar(100)	NOT NULL	The unique ID provided by the payment gateway for the transaction.
status	varchar(20)	NOT NULL, DEFAULT 'completed'	The status of the transaction (e.g., 'completed').
created_at	timestamp	NOT NULL, DEFAULT current_timestamp()	The timestamp of when the purchase was recorded.

CHAPTER 5

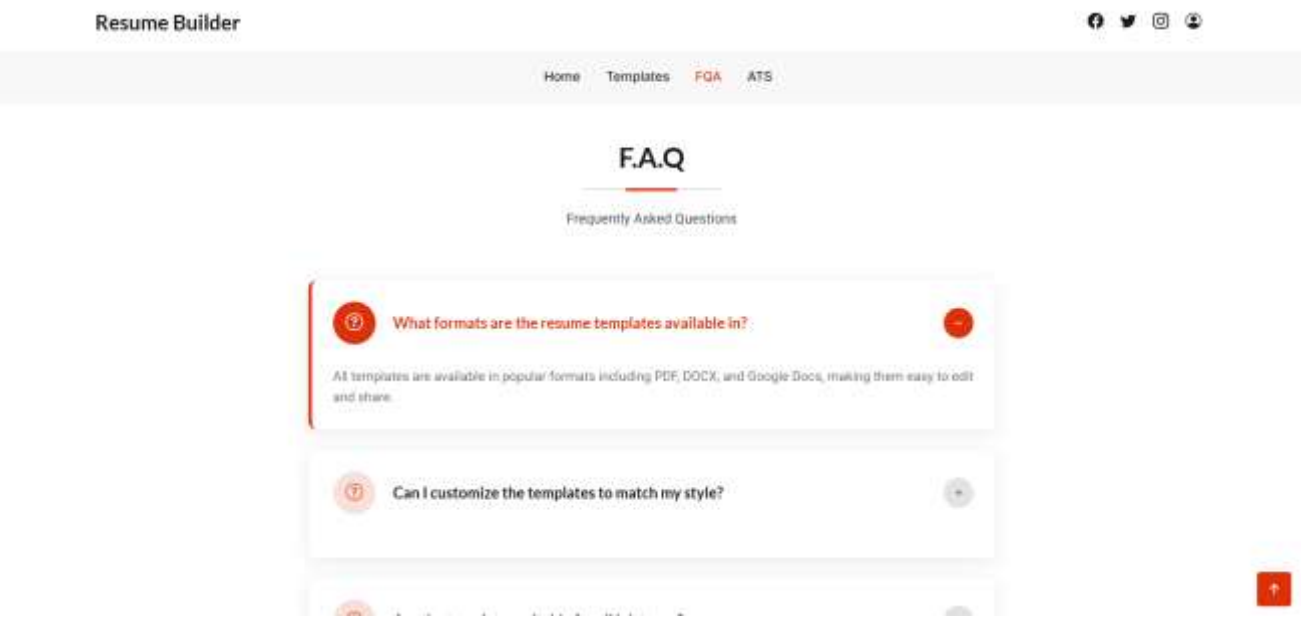
SYSTEM IMPLEMENTATION

This chapter explains the core functionalities of the Resume Builder System, with a specific focus on the features that support ATS optimization. The subsequent sections provide a detailed, step-by-step overview of the implemented features, from user registration to the final resume download and ATS score checker.

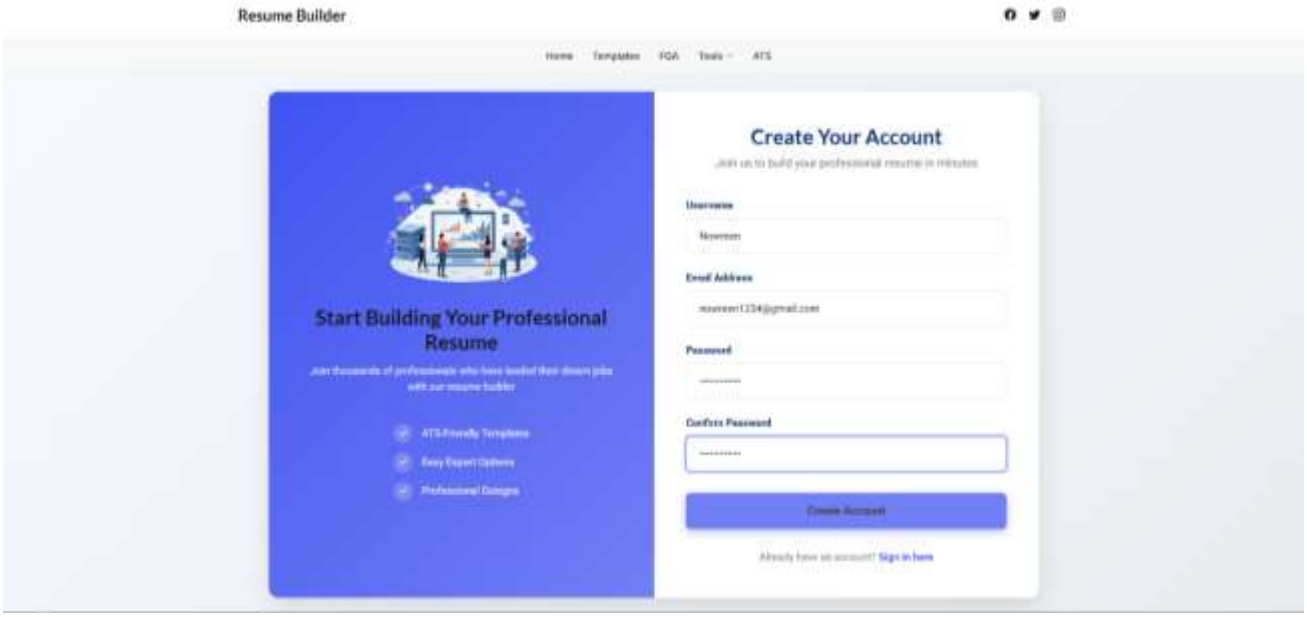
HOME PAGE: This is the main Home page of the Resume Builder platform. It highlights key features like professional templates and an ATS score checker to attract new users. The page provides quick access buttons for registration, login, and exploring templates.



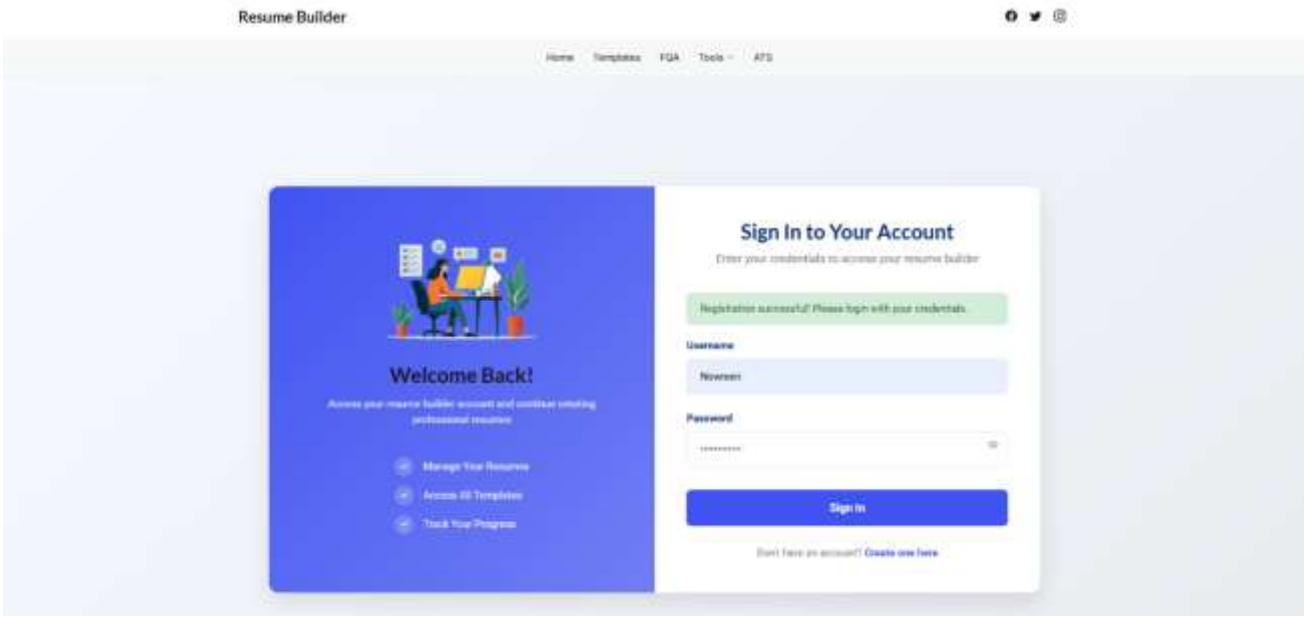
FAQ: When a user clicks on the **FAQ** button from the navigation bar, this page is displayed, providing answers to common questions about the platform. It features an accordion-style layout that allows users to expand each question to view its corresponding answer.



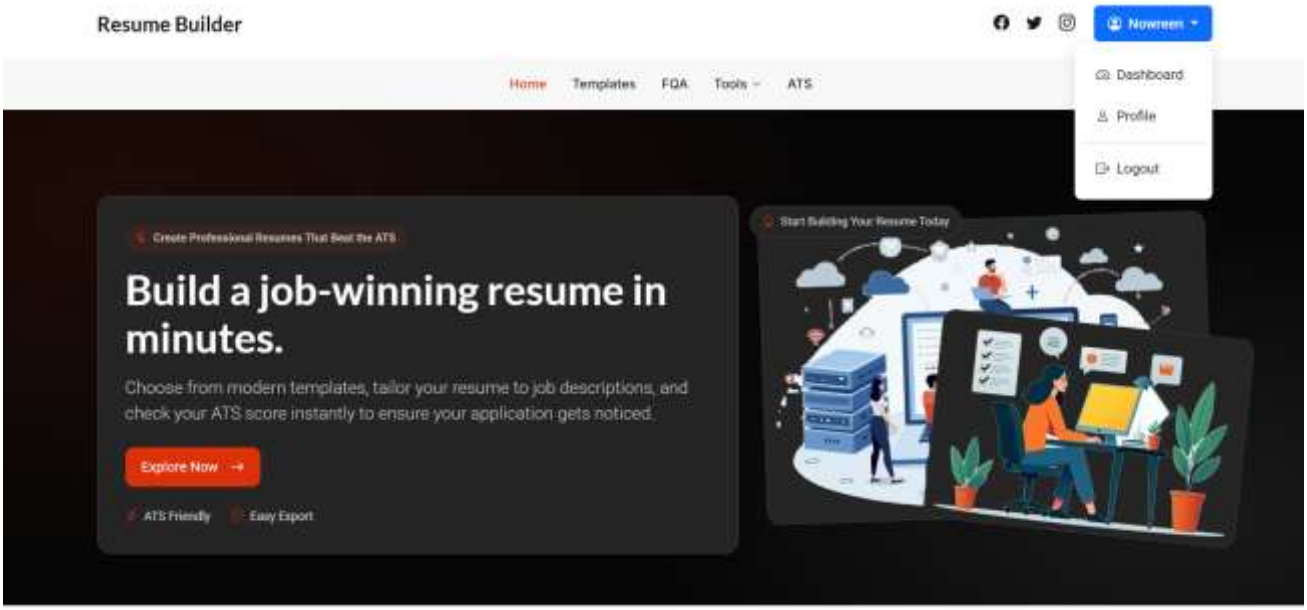
USER REGISTER PAGE: This page is for new users to create an account. It features a form where users can input their personal details, such as a username, email, and password, to sign up for the platform. Upon successful account creation, the user's details are stored in the database, and they are then redirected to the login page. This allows them to securely access their profile and begin creating their resume.



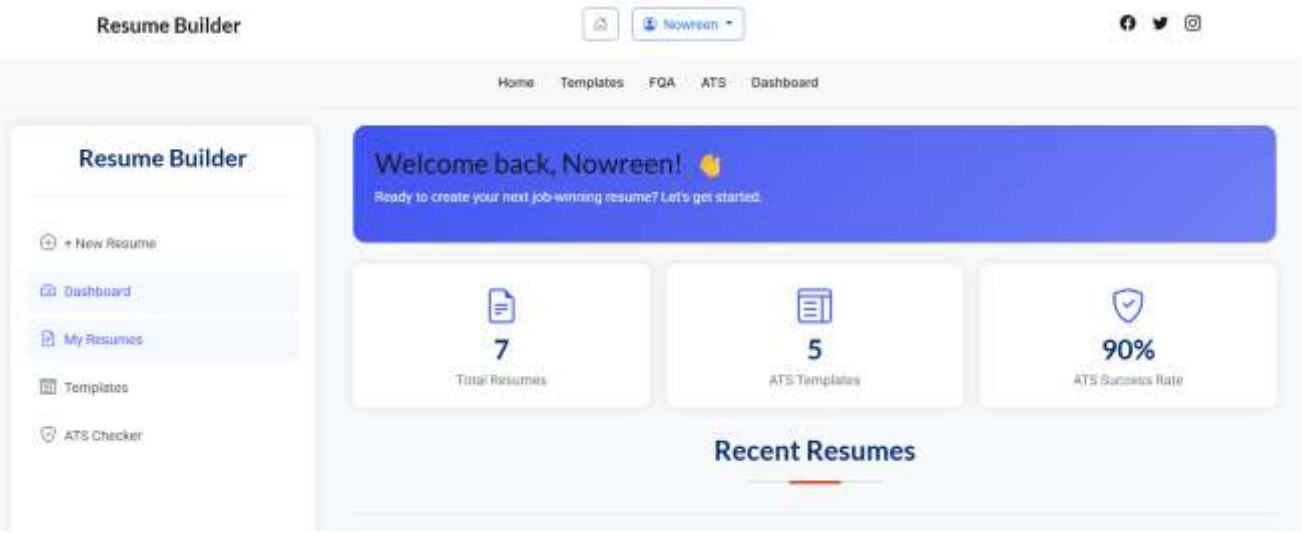
USER LOGIN PAGE: This is the secure login page for existing users. After a user successfully registers, a confirmation message, "Registration successful! Please log in with your credentials," is displayed. Users can enter their username and password to access their account and continue working on their resumes.



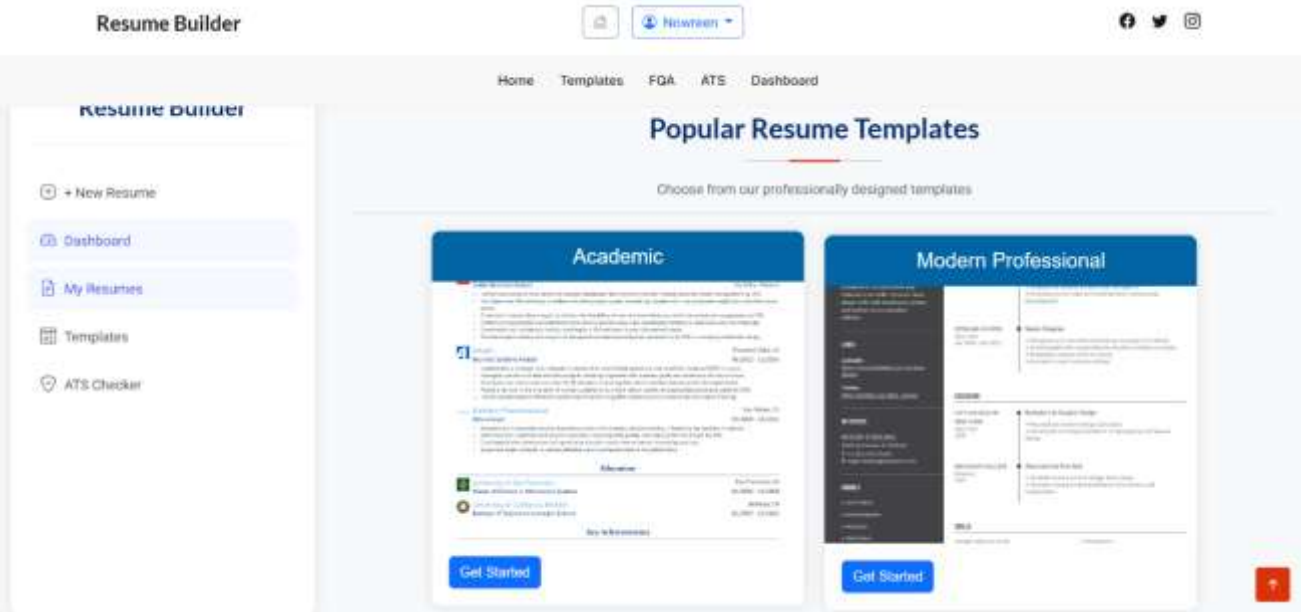
USER HOME: After logging in, this is the main home page for the authenticated user. The navigation bar now includes the user's name with a dropdown menu, providing quick access to their Dashboard and Profile, as well as a Logout option. The main content of the page remains the same, promoting the key features of the resume builder.



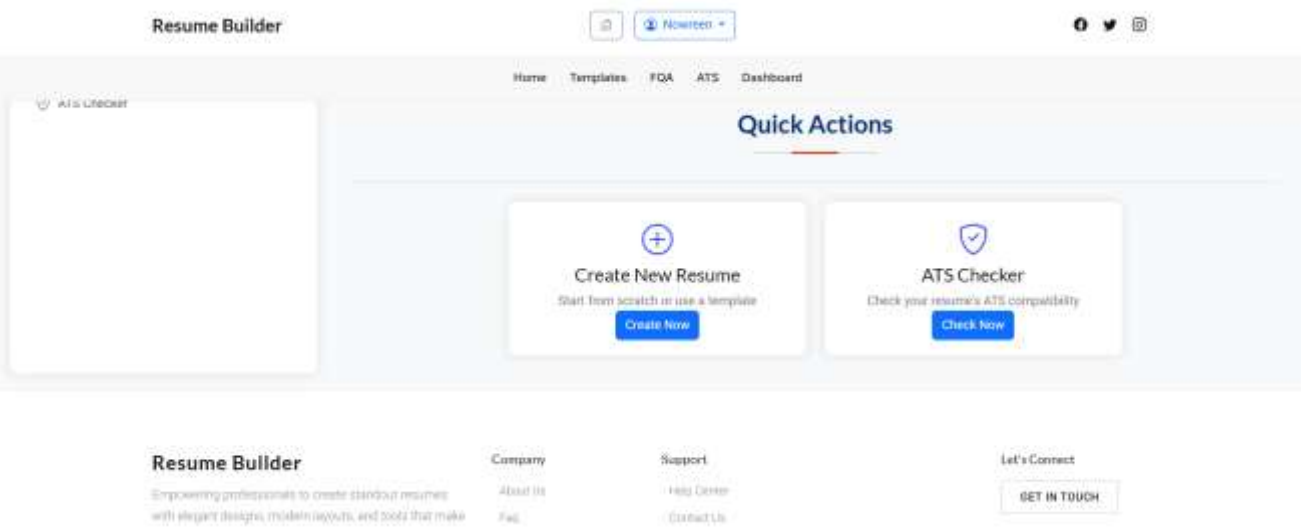
USER DASHBOARD: This is the user's main dashboard after they log in, serving as a central hub for all their activities. The left-hand menu provides quick navigation to create a new resume, view their existing resumes, browse templates, and access the ATS checker. The main section displays a high-level summary of their account, including the total number of resumes, ATS templates available, and their overall ATS success rate.



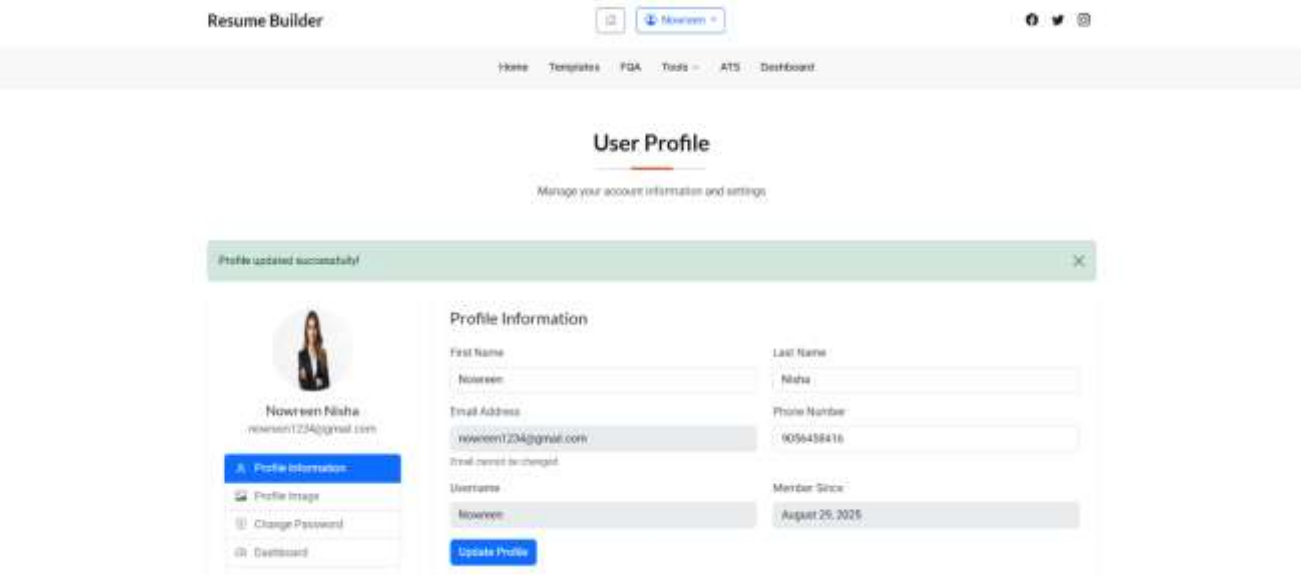
USER DASHBOARD: When a user scrolls down on the User Dashboard page, they are presented with a section titled "Popular Resume Templates." This section showcases a collection of professionally designed templates, such as "Academic" and "Modern Professional," that users can choose from to begin creating their resume.



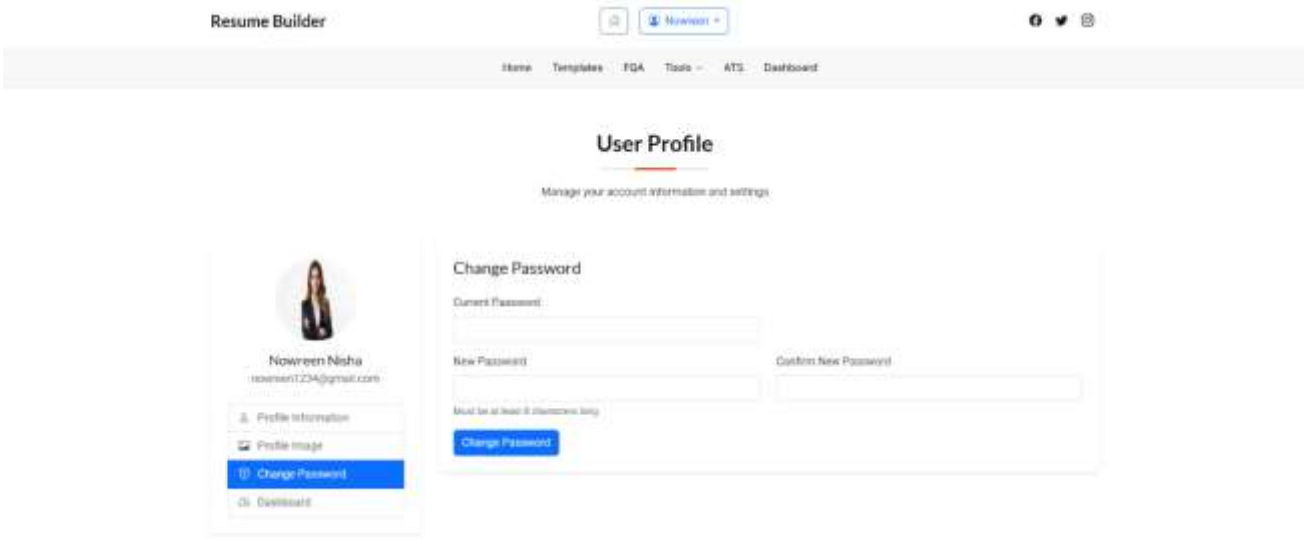
After a user scrolls past the popular resume templates on their dashboard, they will find the "Quick Actions" section. This area provides two prominent buttons: one to Create New Resume, which allows them to start building a resume from scratch or a template, and another for the ATS Checker, which lets them check their resume's compatibility with Applicant Tracking Systems.



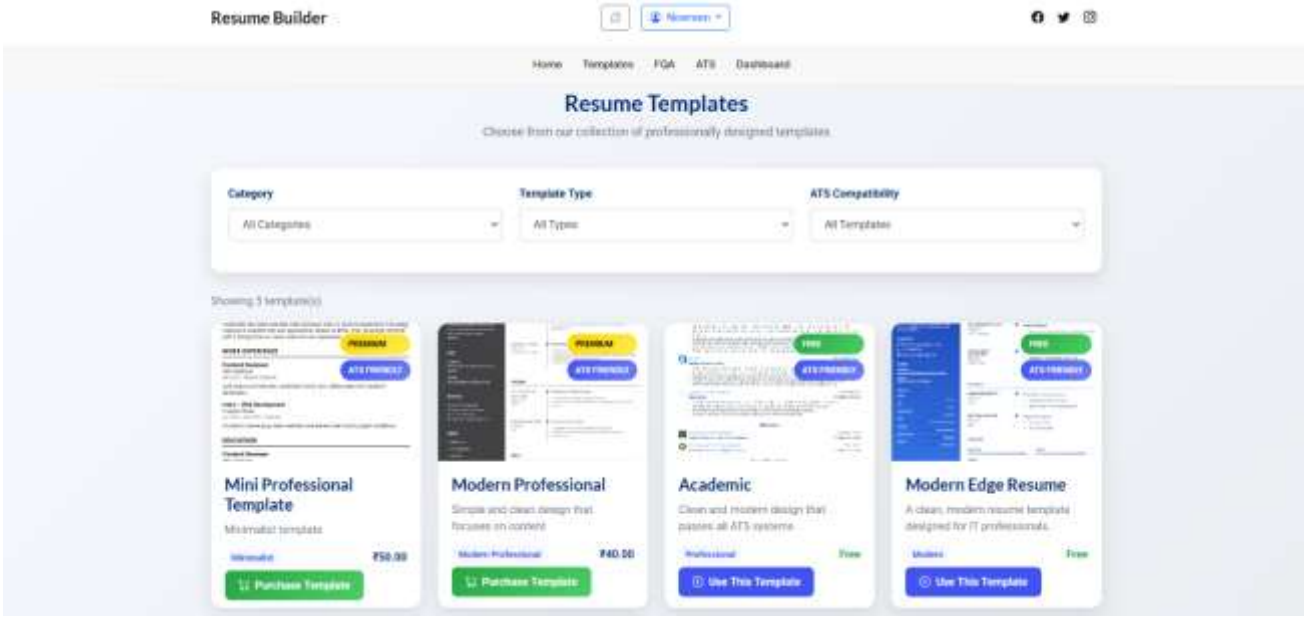
UPDATE USER PROFILE: This page allows the user to manage their account information and settings. The user can view and edit their personal details, such as **first name, last name, phone number, and username**. A profile image can also be uploaded, and the password can be changed through the side menu. Once changes are saved, a green banner confirming **"Profile updated successfully!"** is displayed. The user's email cannot be changed from this page.



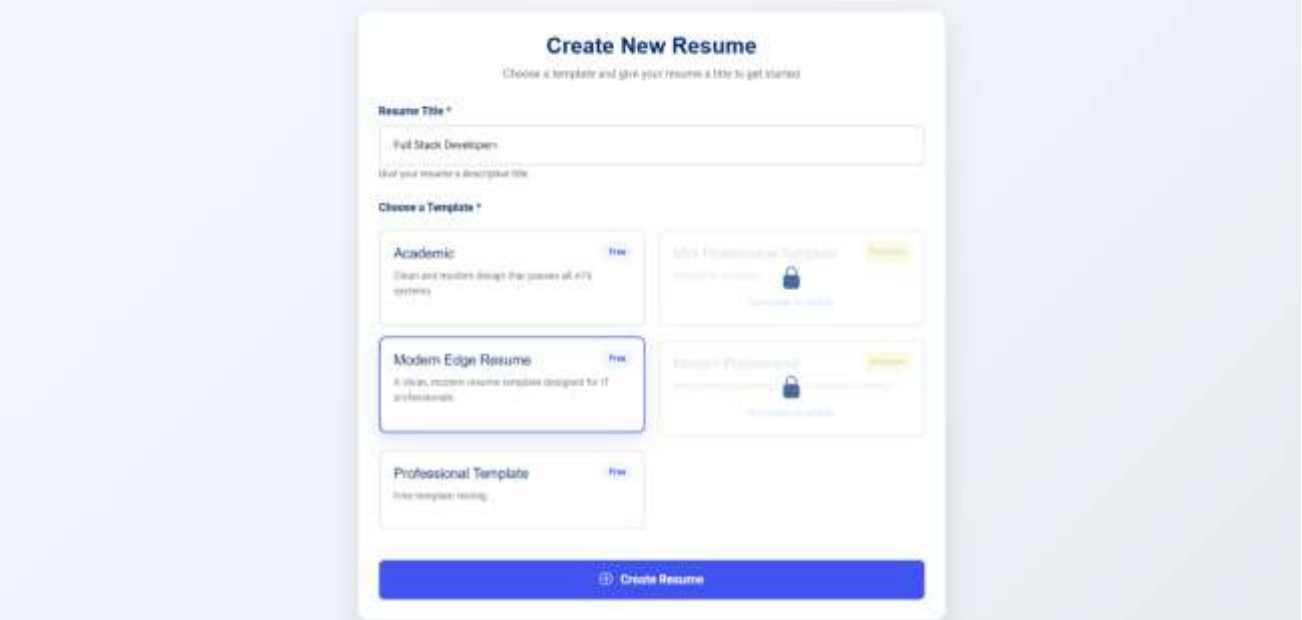
CHANGE PASSWORD: This section of the User Profile page allows users to change their account password. The user must enter their **Current Password**, followed by a **New Password** and a confirmation of the new password. A note specifies that the new password must be at least 8 characters long to ensure security.



TEMPLATES: When a user clicks on the "Templates" link in the navigation bar, they are redirected to this page, which showcases a collection of professionally designed resume templates. Users can browse through a variety of templates, filter them by category, type, and ATS compatibility, and see which ones are free and which are premium. The page also allows users to either **"Purchase Template"** for premium options or **"Use This Template"** for free ones.

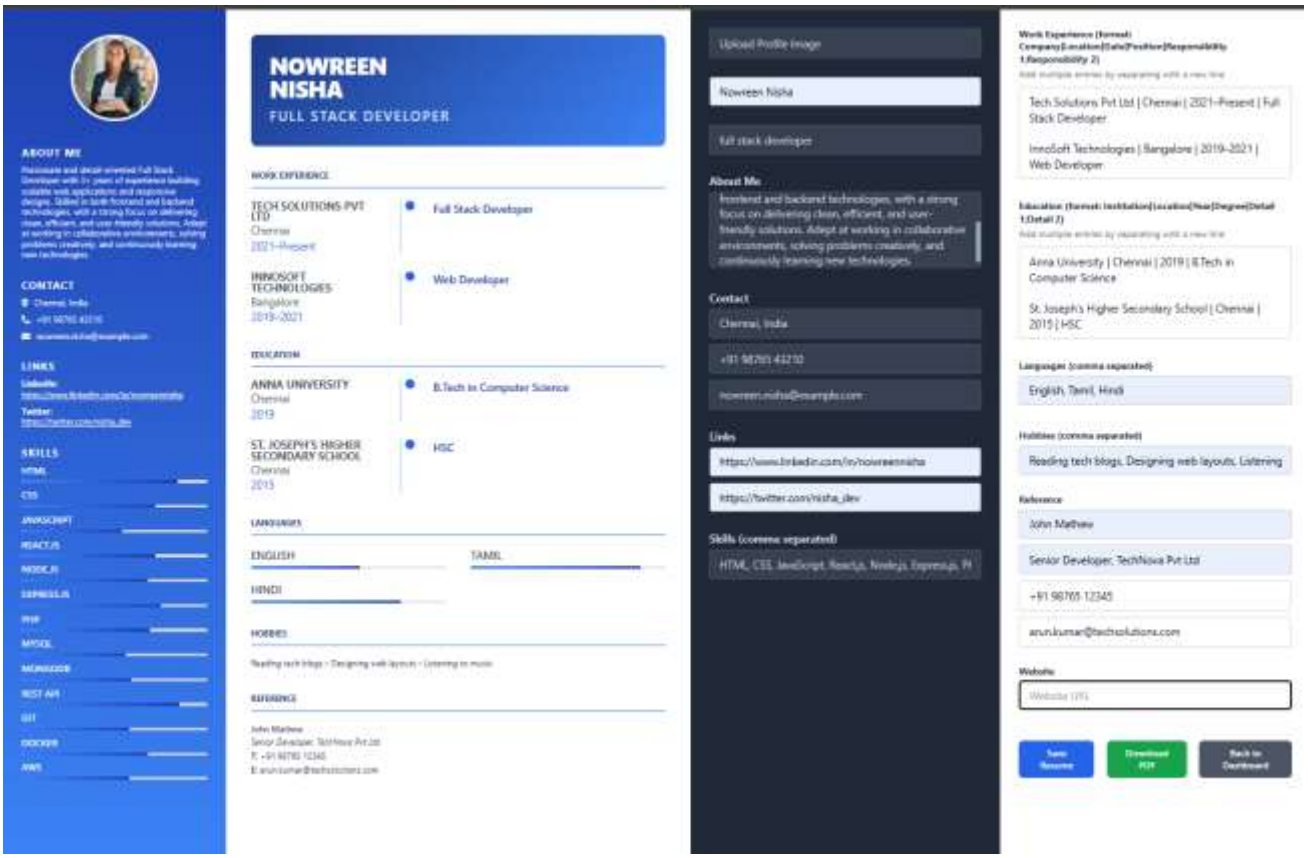


CREATING RESUME: When a user selects a template by clicking "Use This Template," they are redirected to the "Create New Resume" page. On this page, they can give their resume a title and select a template. The page displays a mix of free and premium templates, with premium options locked and requiring a purchase to unlock. After providing a title and choosing a template, the user clicks "Create Resume" to proceed to the next step.

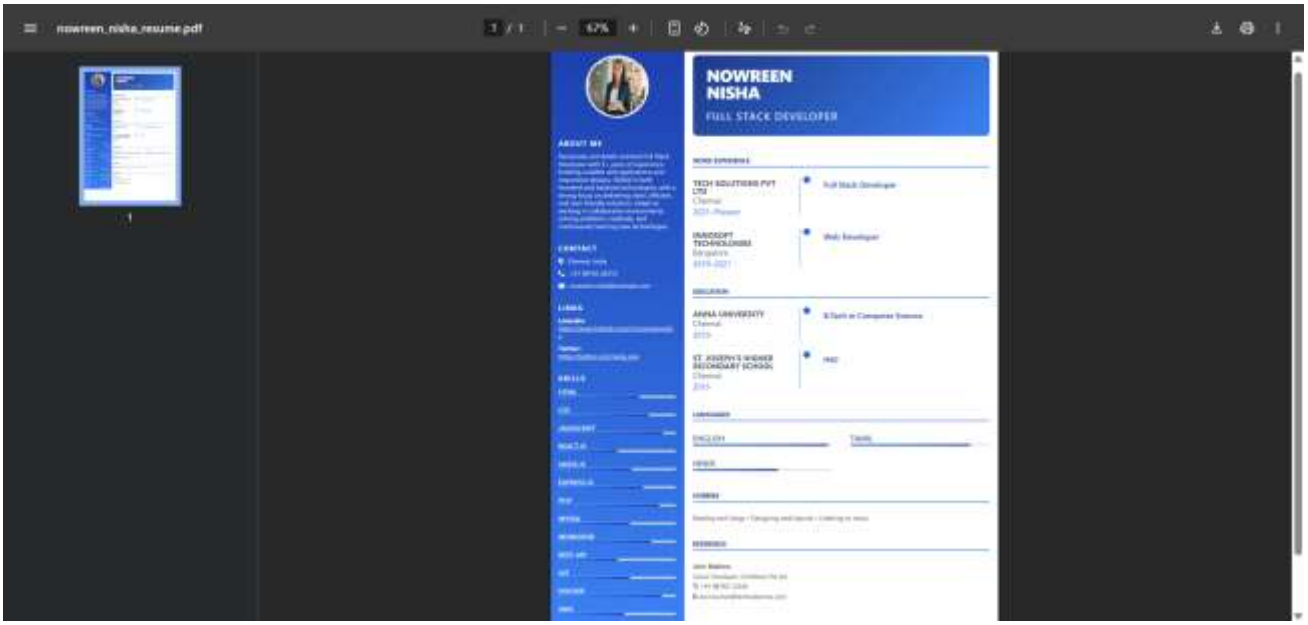


RESUME CREATION: Free Template (Selected Modern Edge Resume)

This is the main **editing interface** for the resume. On the left, a **live preview** of the resume template shows the user exactly how the content will appear. On the right, a **form** allows the user to input and edit all the necessary details, such as personal information, work experience, education, skills, and links. Any changes made in the form are immediately reflected in the live preview on the left. The user can save, download, or return to the dashboard from this page.

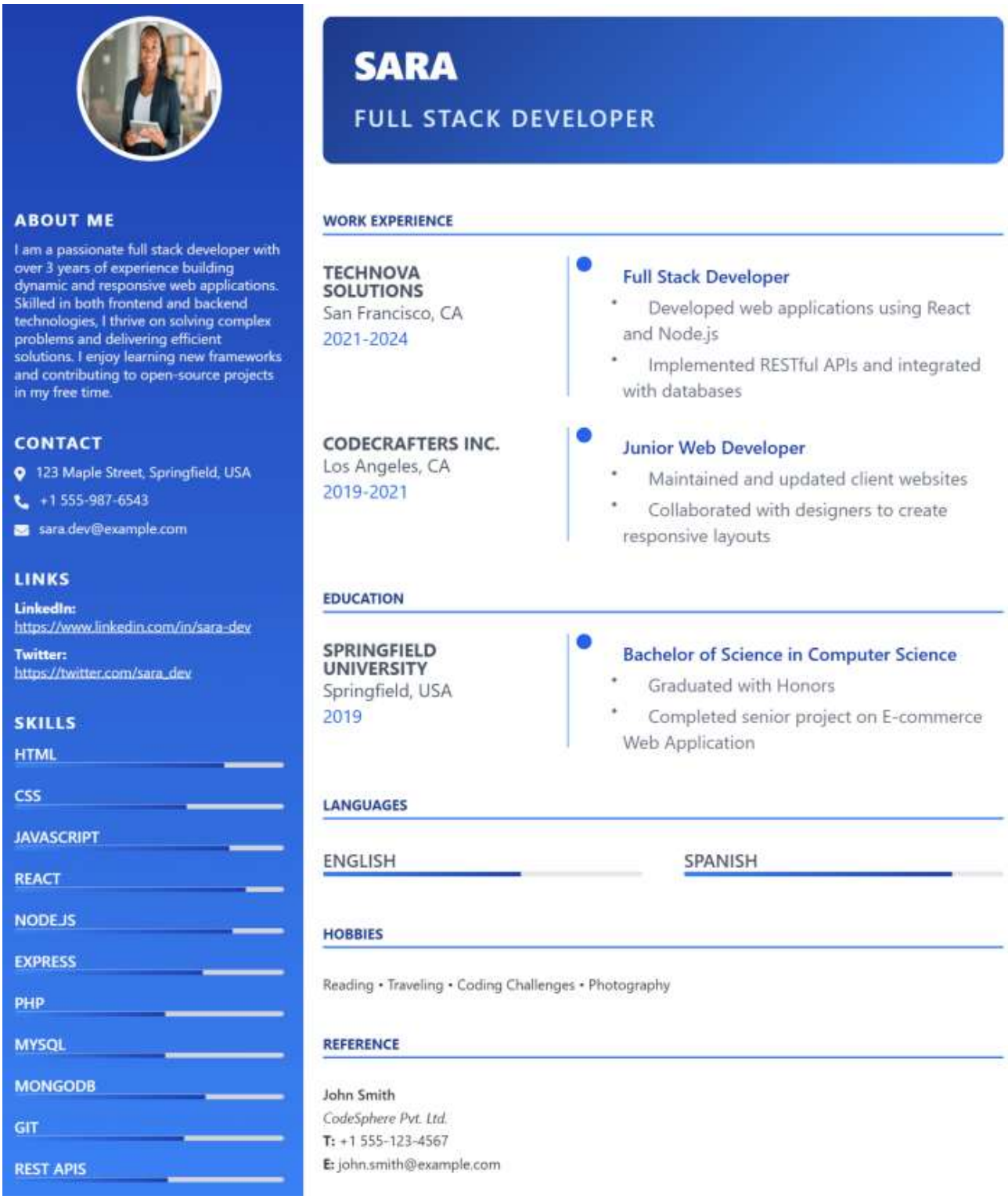


DOWNLOADED RESUME: This page displays the final output of the resume creation process in a format ready for download, such as a PDF file. The page shows a full-screen preview of the completed resume, which includes all the personal details, experience, education, skills, and contact information entered by the user. The user can then download the resume to their device to use for job applications.



DOWNLOADED PDF TEMPLATE: This is the user's final resume, downloaded as a PDF, after being created on the platform. The PDF represents various design options available on the site, ranging from a modern two-column layout to a clean, classic single-column format, confirming that users can build and download professional resumes tailored to different styles and needs.

TEMPLATE 1:



SARA

FULL STACK DEVELOPER

ABOUT ME

I am a passionate full stack developer with over 3 years of experience building dynamic and responsive web applications. Skilled in both frontend and backend technologies, I thrive on solving complex problems and delivering efficient solutions. I enjoy learning new frameworks and contributing to open-source projects in my free time.

CONTACT

123 Maple Street, Springfield, USA

+1 555-987-6543

sara.dev@example.com

LINKS

LinkedIn:
https://www.linkedin.com/in/sara-dev

Twitter:
https://twitter.com/sara_dev

SKILLS

HTML

CSS

JAVASCRIPT

REACT

NODE.JS

EXPRESS

PHP

MYSQL

MONGODB

GIT

REST APIS

WORK EXPERIENCE

TECHNOVA SOLUTIONS
San Francisco, CA
2021-2024

Full Stack Developer

- Developed web applications using React and Node.js
- Implemented RESTful APIs and integrated with databases

CODECRAFTERS INC.
Los Angeles, CA
2019-2021

Junior Web Developer

- Maintained and updated client websites
- Collaborated with designers to create responsive layouts

EDUCATION

SPRINGFIELD UNIVERSITY
Springfield, USA
2019

Bachelor of Science in Computer Science

- Graduated with Honors
- Completed senior project on E-commerce Web Application

LANGUAGES

ENGLISH

SPANISH

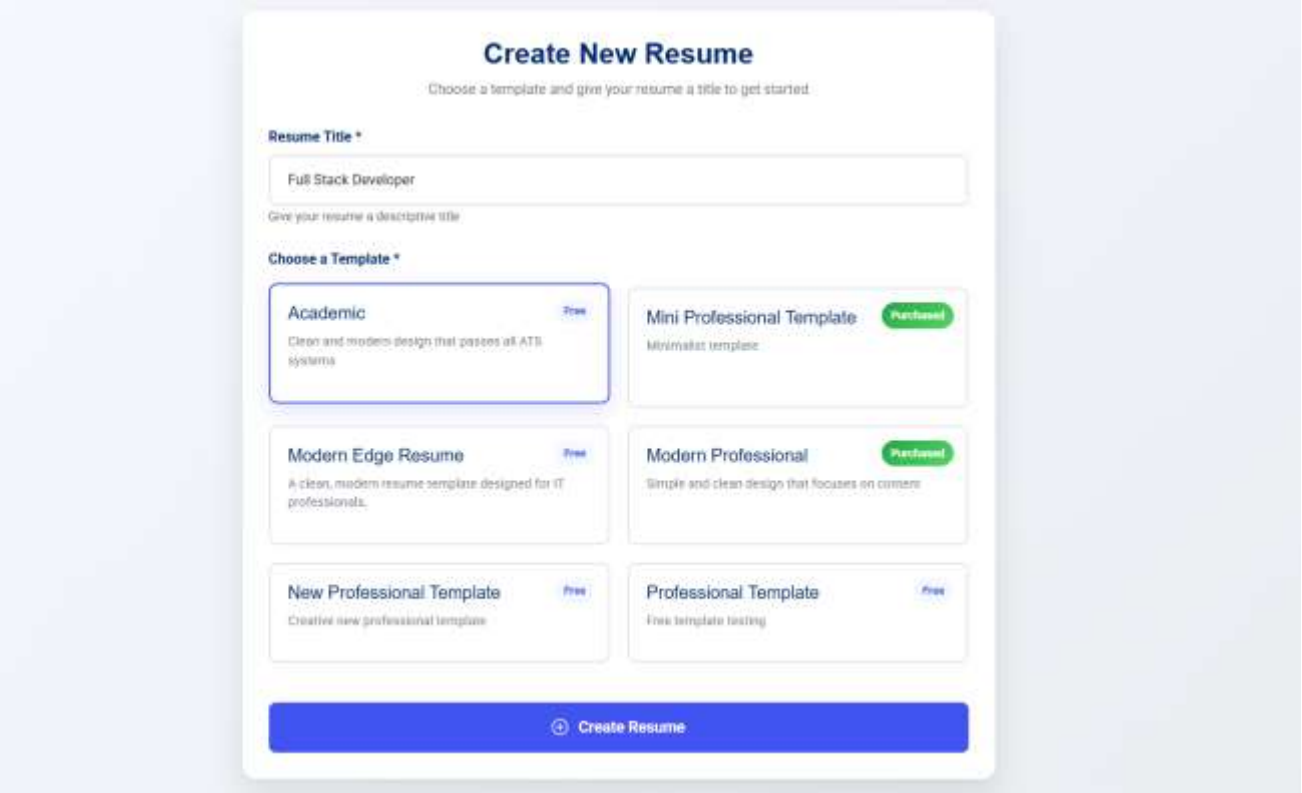
HOBBIES

Reading • Traveling • Coding Challenges • Photography

REFERENCE

John Smith
CodeSphere Pvt. Ltd.
T: +1 555-123-4567
E: john.smith@example.com

TEMPLATE 2: (Free Template) First, choose a resume title and select a template from the available options. Click "Create Resume" to proceed.



Next, the user will be taken to an editing page. Here, the user can fill in their personal information, summary, experience, education, skills, and certificates.

DOWNLOADED PDF TEMPLATE: Finally, the user can view their completed resume. This is the polished version, ready for printing.

Nowreen Nisha

Full Stack Developer

nisha23@gmail.com • https://www.linkedin.com/in/nowreennisha • Mumbai, Maharashtra

SUMMARY

Dedicated and detail-oriented Software Developer with strong experience in building responsive web applications and writing clean, maintainable code. Skilled in full-stack development, debugging, and optimizing performance. Passionate about learning emerging technologies, collaborating on innovative projects, and delivering solutions that enhance user experience.

EXPERIENCE

TechNova Solutions
Tirunelveli
Mern Stack Developer
01/2020 – Present
Developed and maintained dynamic MERN stack applications
Built RESTful APIs and integrated them with front-end components to ensure seamless data flow.
Collaborated with UI/UX designers to create user-friendly and responsive interfaces.
Improved page load time by 30% through code optimization and caching strategies.
Conducted version control management with Git/GitHub and participated in Agile sprints for iterative development.

EDUCATION

Christ University
Master of Science in Information Systems
Tirunelveli
06/2014 – 05/2016

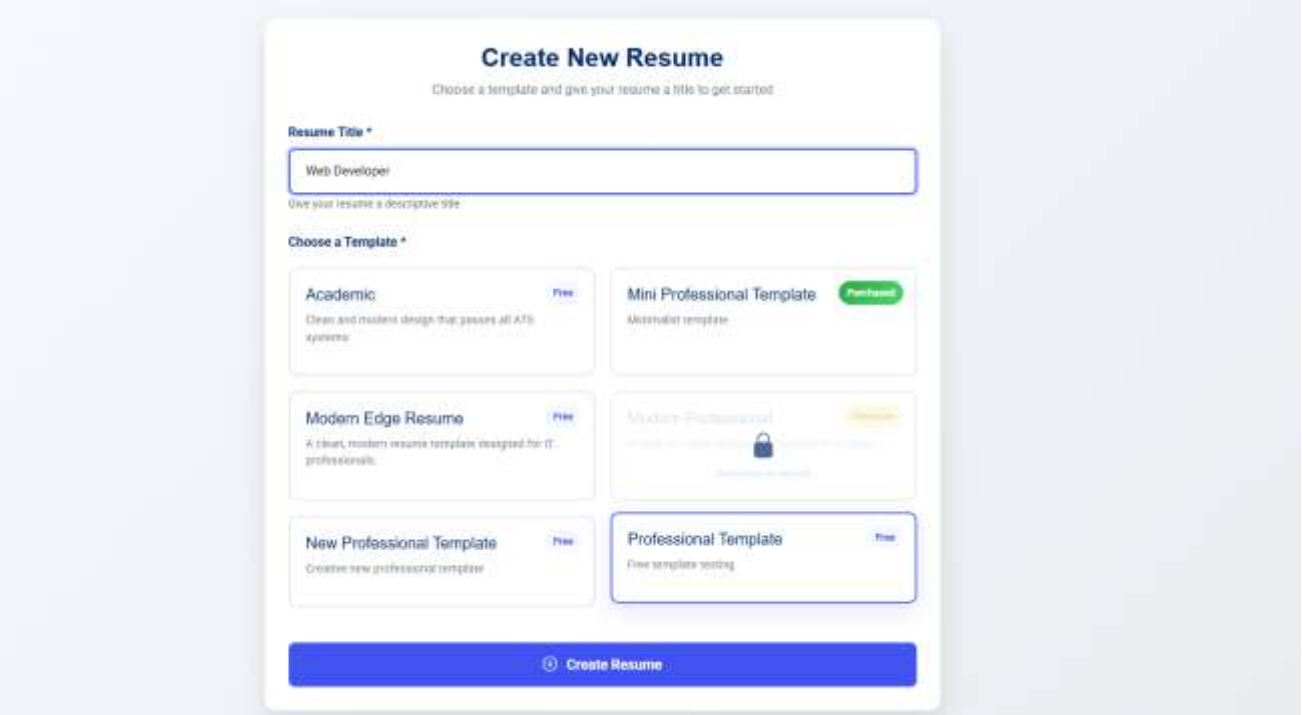
SKILLS

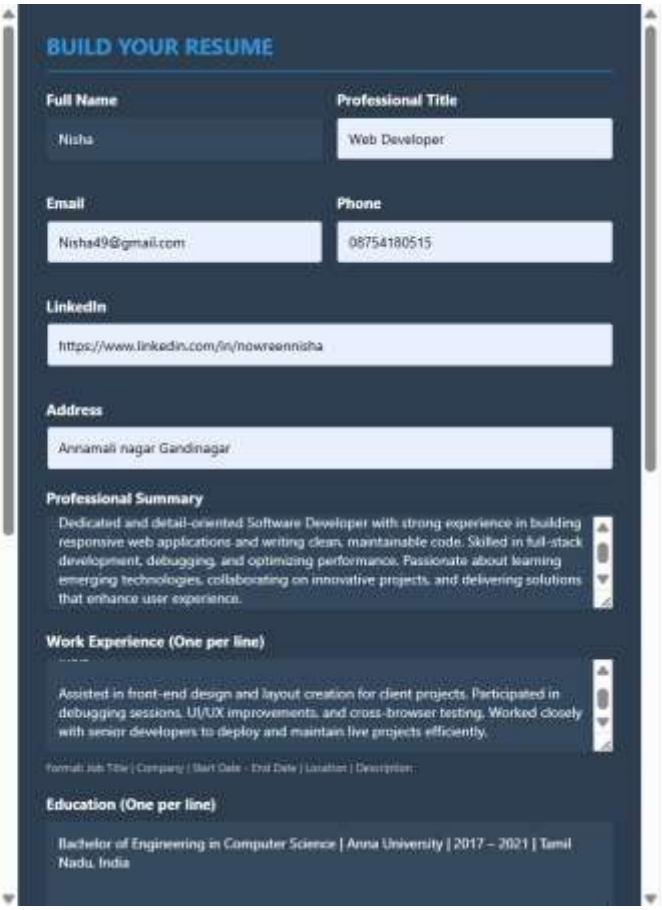
- Frontend: HTML, CSS, JavaScript, React.js, Redux
- Backend: Node.js, Express.js, PHP
- Database: MongoDB, MySQL
- Tools & Platforms: Git, GitHub, VS Code, Postman
- Other: RESTful APIs, Agile Methodologies, UI/UX Collaboration, Problem Solving

CERTIFICATIONS

- Full Stack Web Development – Coursera
- JavaScript Algorithms and Data Structures – freeCodeCamp
- Responsive Web Design – Udemy

TEMPLATE 3: (Free Template)





DOWNLOADED PDF TEMPLATE:

Nisha

Web Developer

Nisha49@gmail.com https://www.linkedin.com/in/nowreennisha

Annamali nagar Gandinagar

SUMMARY

Dedicated and detail-oriented Software Developer with strong experience in building responsive web applications and writing clean, maintainable code. Skilled in full-stack development, debugging, and optimizing performance. Passionate about learning emerging technologies, collaborating on innovative projects, and delivering solutions that enhance user experience.

EXPERIENCE

Web Developer | TechNova Solutions | Jan 2023 – Present | Chennai, India

Developed dynamic and user-friendly web applications using HTML, CSS, JavaScript, and React.js. Collaborated with backend teams to integrate RESTful APIs and enhance overall functionality. Maintained and optimized existing codebases for better performance and scalability.

Junior Web Developer | CodeWave Technologies | Jul 2021 – Dec 2022 | Coimbatore, India

Assisted in front-end design and layout creation for client projects. Participated in debugging sessions, UI/UX improvements, and cross-browser testing. Worked closely with senior developers to deploy and maintain live projects efficiently.

EDUCATION

Bachelor of Engineering in Computer Science | Anna University | 2017 – 2021 | Tamil Nadu, India

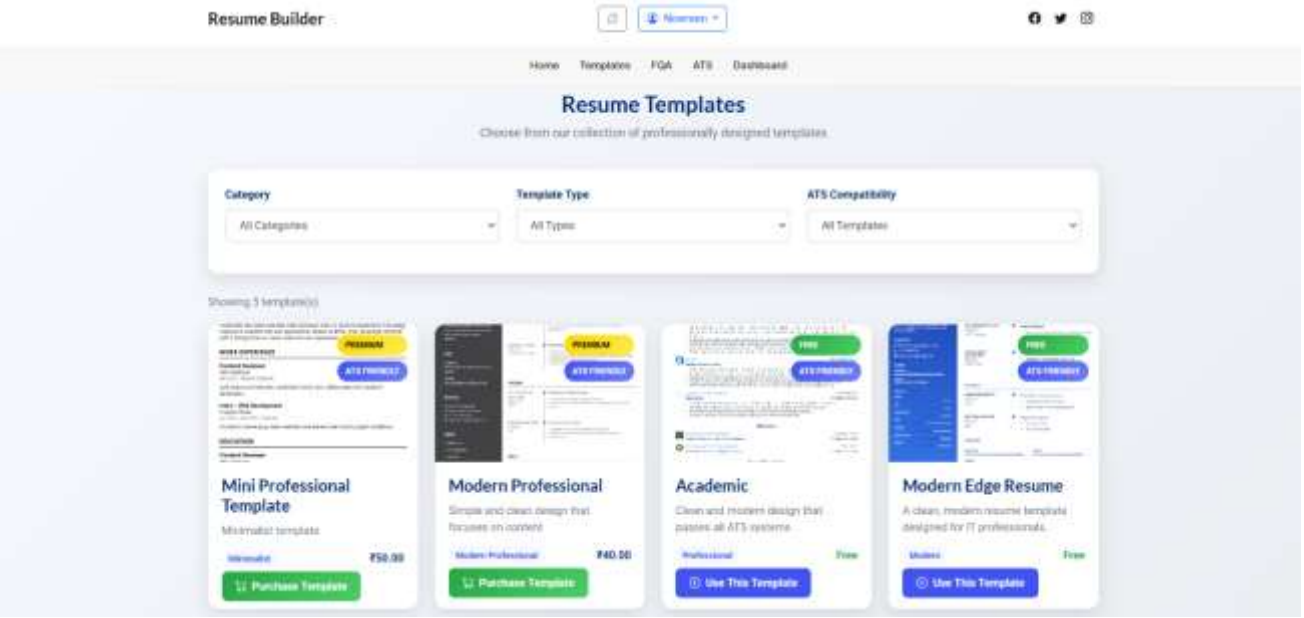
LANGUAGES

English ○○○○○

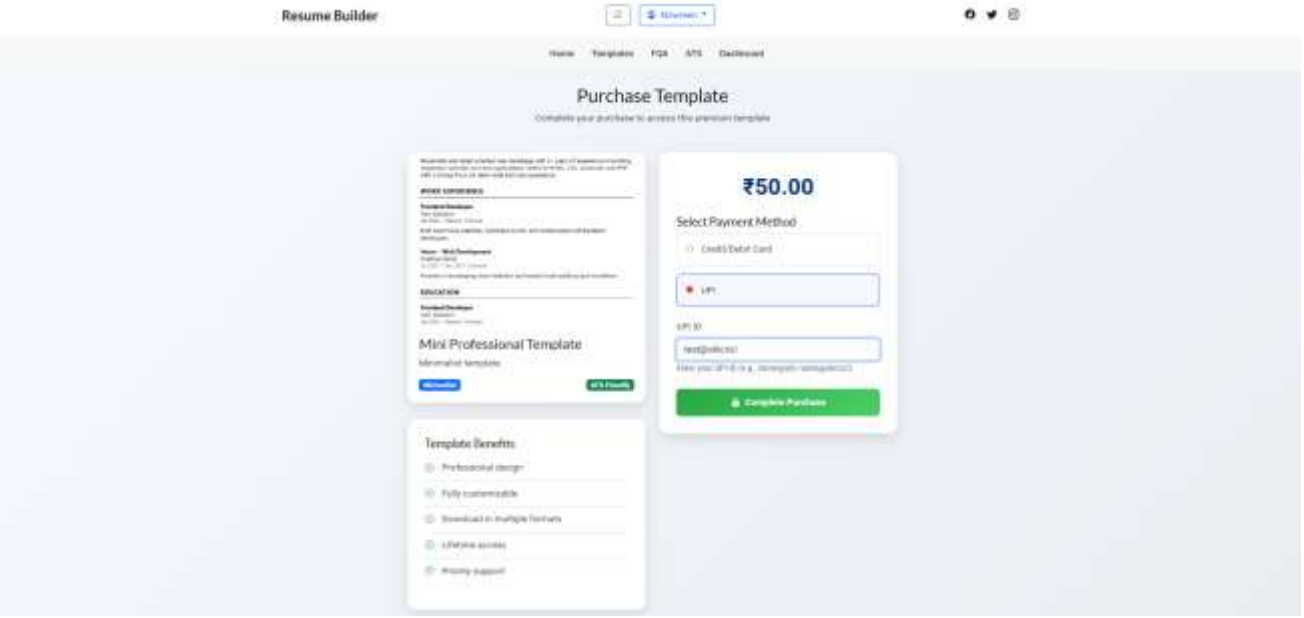
SKILLS

- HTML, CSS, JavaScript
- React.js, Node.js, Express.js
- PHP, MySQL, MongoDB
- Git & GitHub

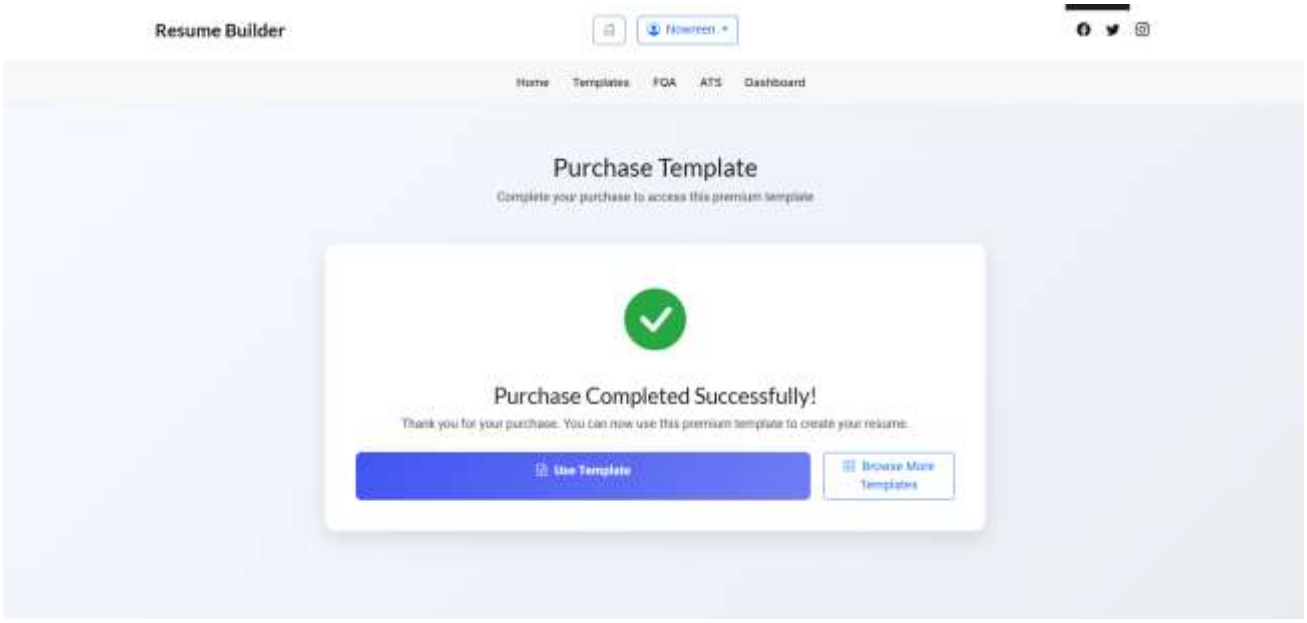
TEMPLATES: To purchase the template user needs to click the "Purchase Template" button for the "Mini Professional Template" or "Modern Professional" templates, which have a price listed on them.



PURCHASING TEMPLATES: This page is for users to purchase a premium template. It displays a preview of the selected template, such as the **"Mini Professional Template,"** and lists the benefits of purchasing it, like **"Lifetime access"** and **"Priority support."** The user can choose a payment method, such as UPI or Credit/Debit Card, and then complete the purchase. The price of the template is clearly displayed on the page.

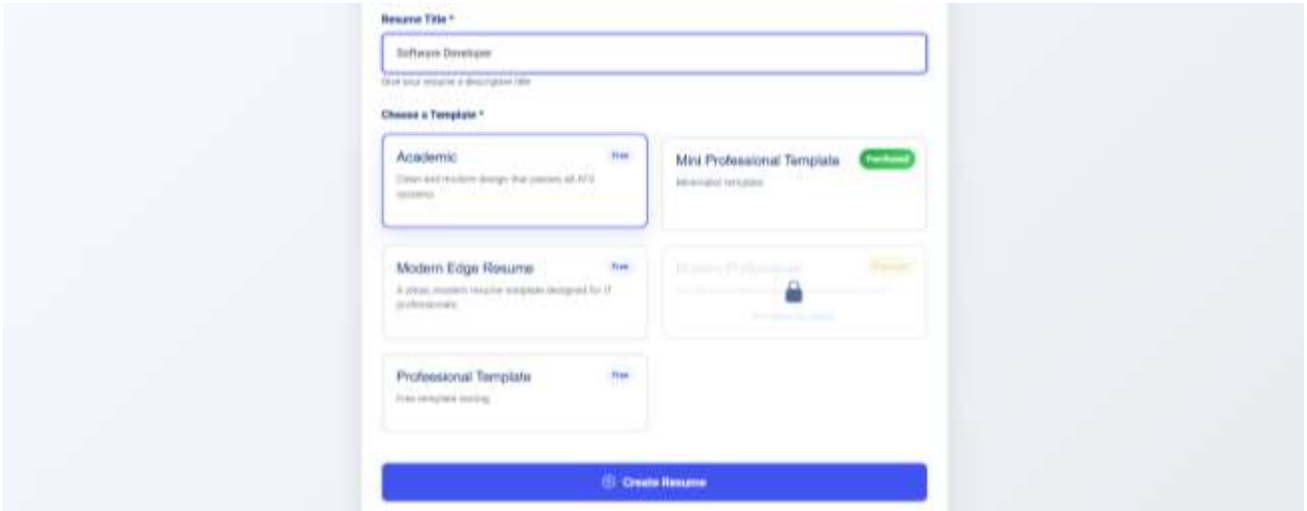


After a user completes a purchase, the system displays a confirmation page with a "Purchase Completed Successfully!" message. This page thanks the user for their purchase and provides two options: "Use Template," which redirects them to the resume creation process with the new template, and "Browse More Templates", which takes them back to the templates page.



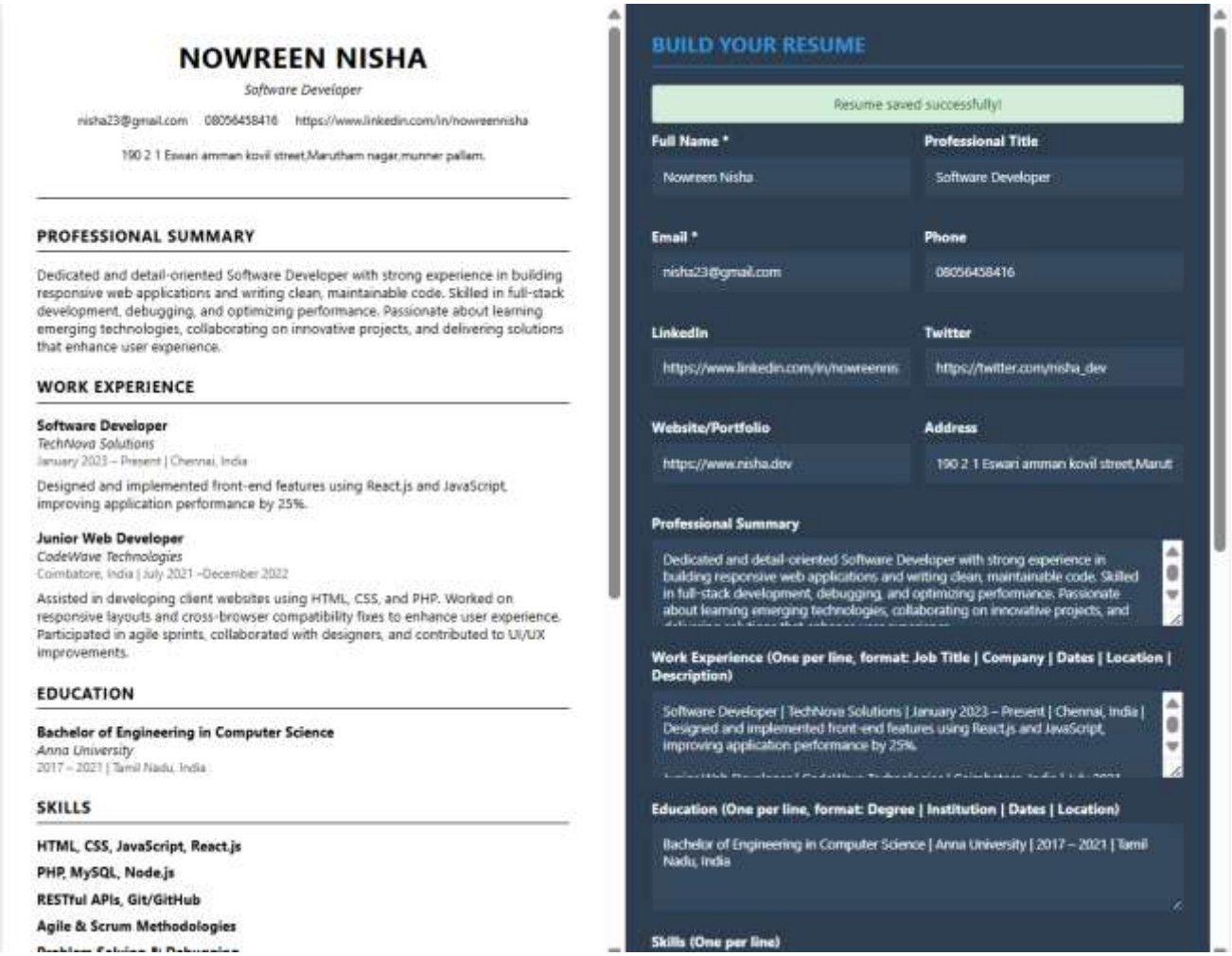
Resume Creation: Using a Purchased Template

This page allows the user to start creating a new resume after they've purchased a premium template. It displays a "Purchase Complete" message on the previously locked template, indicating it's now available for use. The user can give their resume a title, select the purchased template, and then click "Create Resume" to proceed to the editing interface.



TEMPLATE 4: Mini Professional Template:

PAID TEMPLATE: Here, the user can input their professional information, such as their personal summary, work experience, education, and skills. The page is split, allowing them to see their information and a preview of the resume simultaneously.



DOWNLOADED PDF TEMPLATE: The user's downloaded resume is a polished and professional document ready for sharing. It contains all the information they entered, including their professional summary, work experience, education, and skills. The layout is clean and easy to read, ensuring that hiring managers can quickly find the information they need.

NOWREEN NISHA

Software Developer

nisha23@gmail.com 08056458416 https://www.linkedin.com/in/nowreennisha

190 2 1 Eswari amman kovil street,Marutham nagar,munner pallam.

PROFESSIONAL SUMMARY

Dedicated and detail-oriented Software Developer with strong experience in building responsive web applications and writing clean, maintainable code. Skilled in full-stack development, debugging, and optimizing performance. Passionate about learning emerging technologies, collaborating on innovative projects, and delivering solutions that enhance user experience.

WORK EXPERIENCE

Software Developer
TechNova Solutions
January 2023 – Present | Chennai, India

Designed and implemented front-end features using React.js and JavaScript, improving application performance by 25%.

Junior Web Developer
CodeWave Technologies
Coimbatore, India | July 2021 –December 2022

Assisted in developing client websites using HTML, CSS, and PHP. Worked on responsive layouts and cross-browser compatibility fixes to enhance user experience. Participated in agile sprints, collaborated with designers, and contributed to UI/UX improvements.

EDUCATION

Bachelor of Engineering in Computer Science
Anna University
2017 – 2021 | Tamil Nadu, India

SKILLS

HTML, CSS, JavaScript, React.js

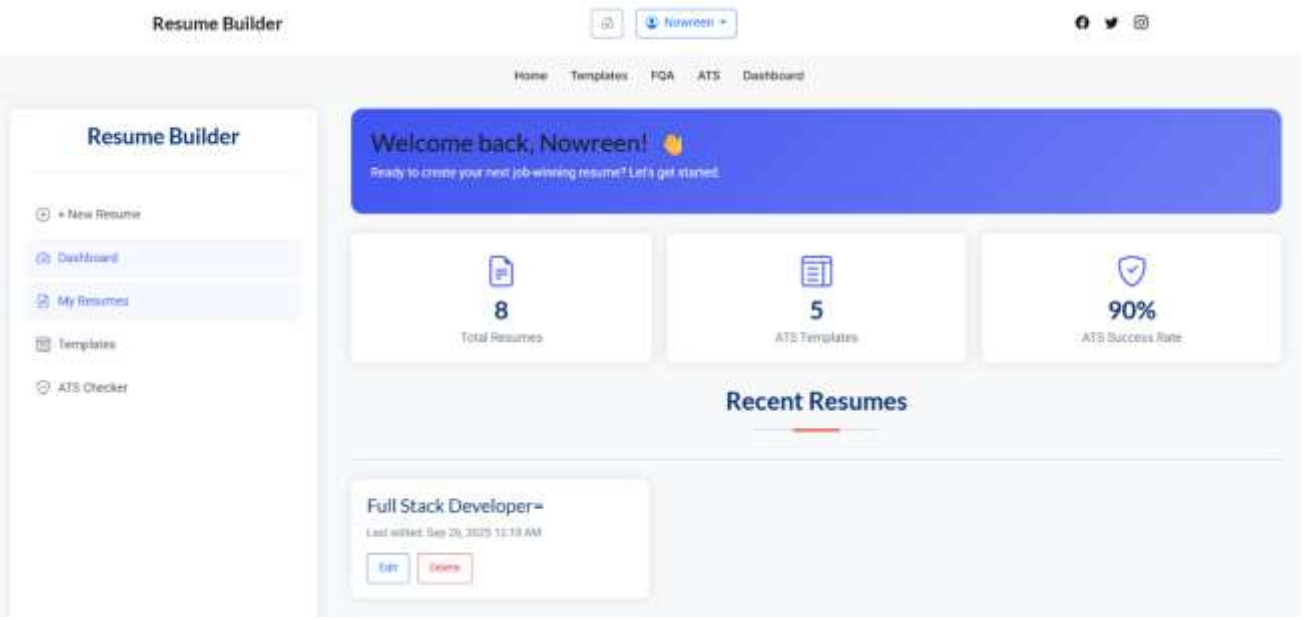
PHP, MySQL, Node.js

RESTful APIs, Git/GitHub

Agile & Scrum Methodologies

Problem Solving & Debugging

USER DASHBOARD AFTER RESUME CREATION: After creating a resume, the user's dashboard is updated to reflect their new resume and activity. The main section now shows an increased **"Total Resumes"** count and a new entry under **"Recent Resumes"**. This new entry displays the resume's title, a timestamp of the last edit, and buttons to **Edit** or **Delete** the resume.

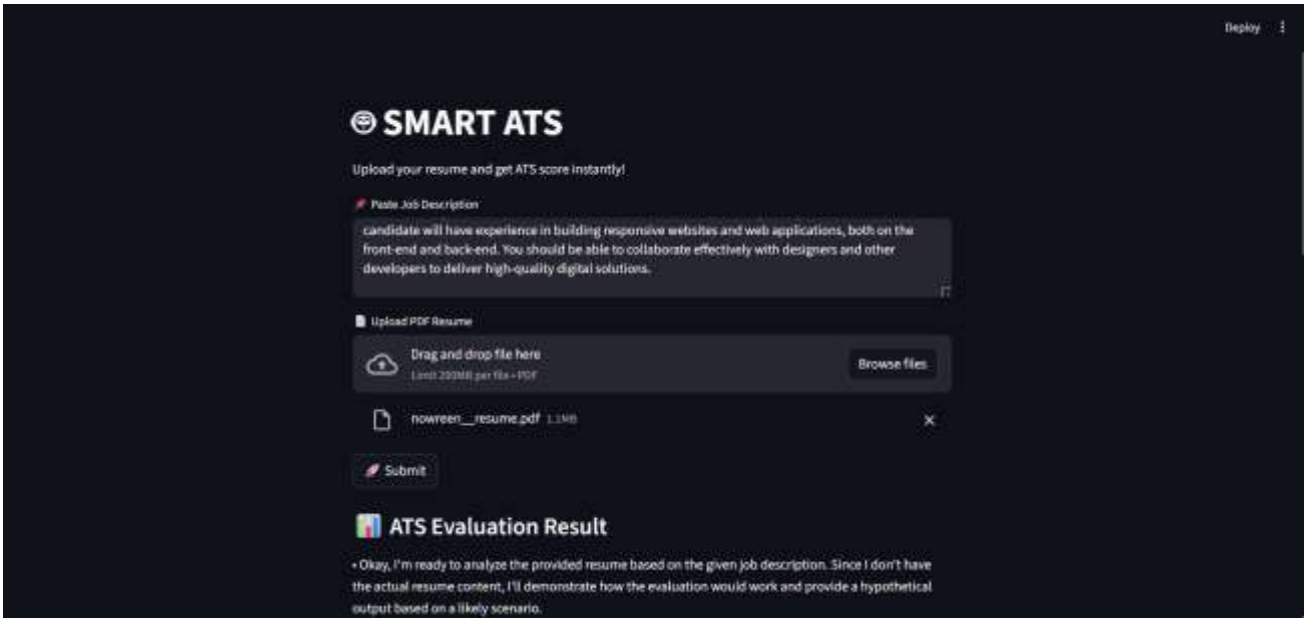


ATS CHECKER: This page tells about the tool that will help users optimize their resumes for Applicant Tracking Systems.

ATS Checker Functionality: The tool allows users to upload a PDF resume and paste a job description. The system then performs a comprehensive evaluation to help optimize the resume for ATS and hiring managers.

PDF Parsing

The **PyPDF2 library** is used to extract text from the uploaded PDF resume. This step is crucial as it converts the unstructured PDF data into a readable text format that the AI can analyze.

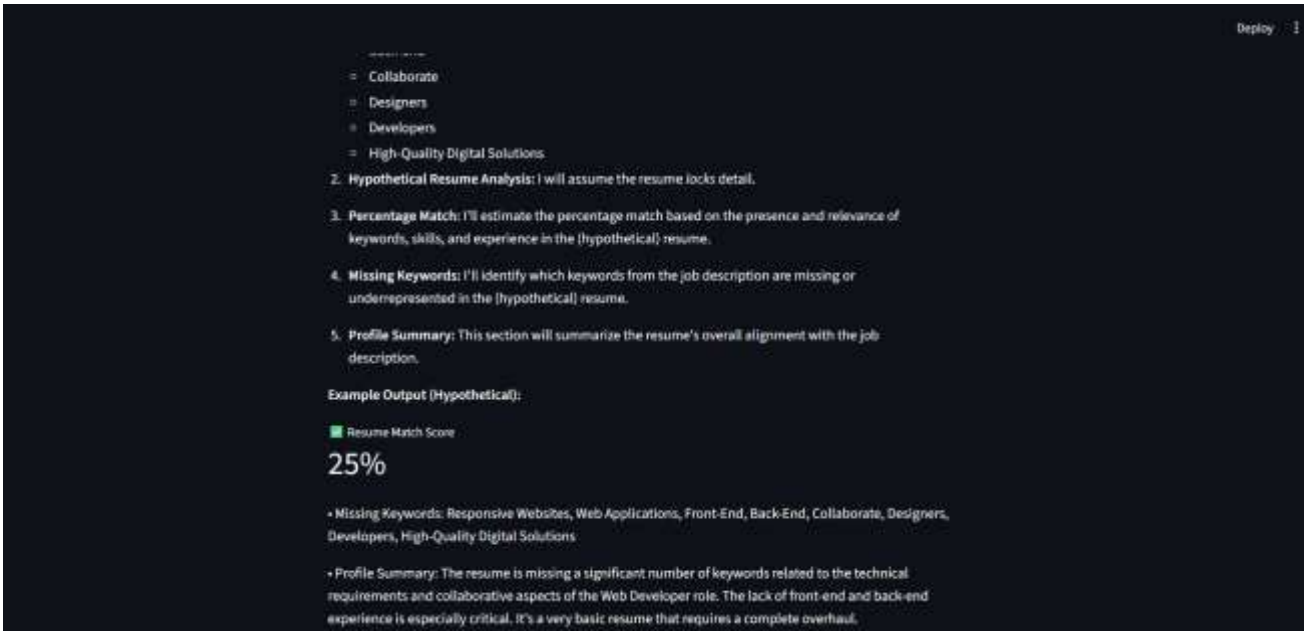


Resume Analysis with Gemini

Once the text is extracted, it is sent to **Google's Gemini API** along with the job description. The Gemini model, prompted to act as an experienced ATS, analyzes the content to provide an evaluation. The system delivers a compatibility score, lists missing keywords, and gives a summary of the candidate's profile. The code is configured to use the gemini-1.5-pro model.

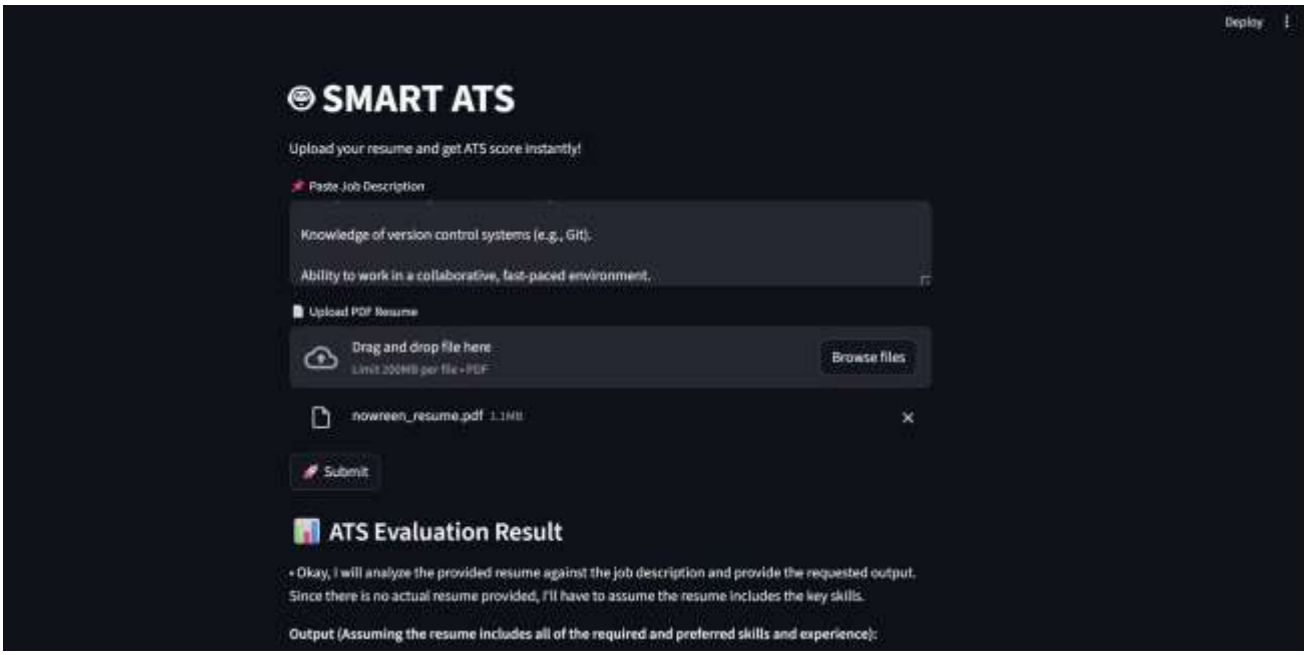
PERCENTAGE: 25%

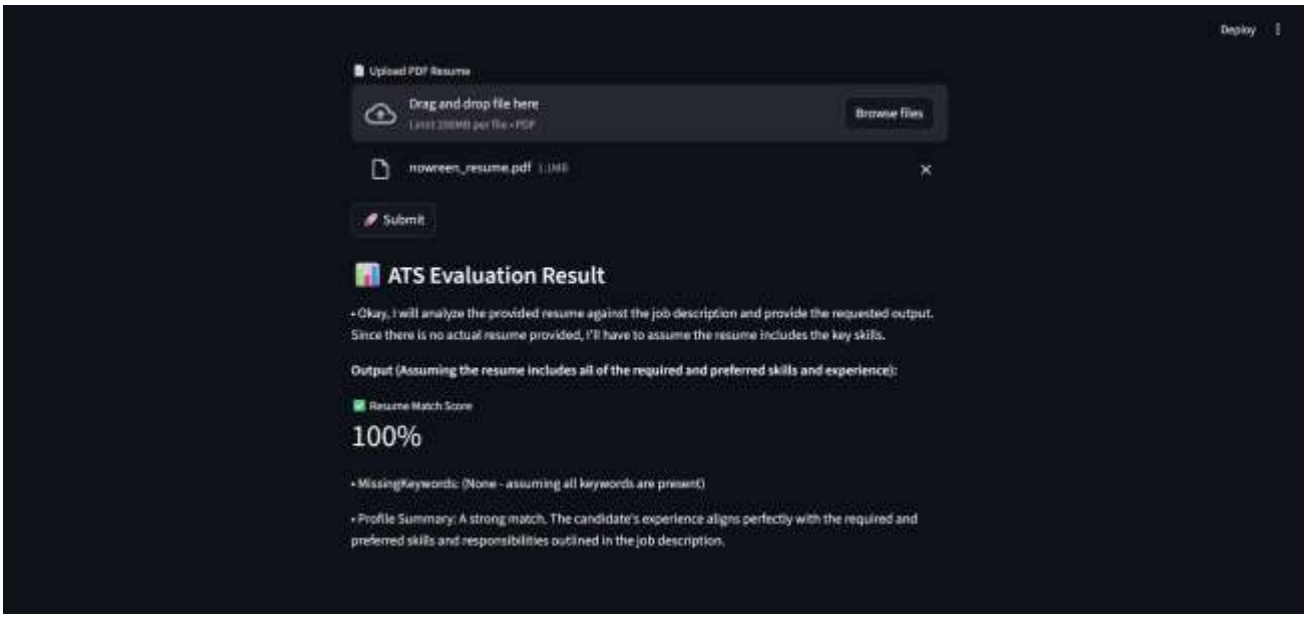
This page shows the results of a low-scoring ATS check. The **"Resume Match Score"** is displayed as **25%**. The results highlight key areas for improvement, specifically listing **"Missing Keywords"** such as "Responsive Websites" and "Web Applications". The **"Profile Summary"** section explains that the resume is missing critical keywords and lacks a strong match to the technical and collaborative requirements of the job description, suggesting the resume requires a complete overhaul to be effective.



RESUME 2: ATS Checker Result: 100%

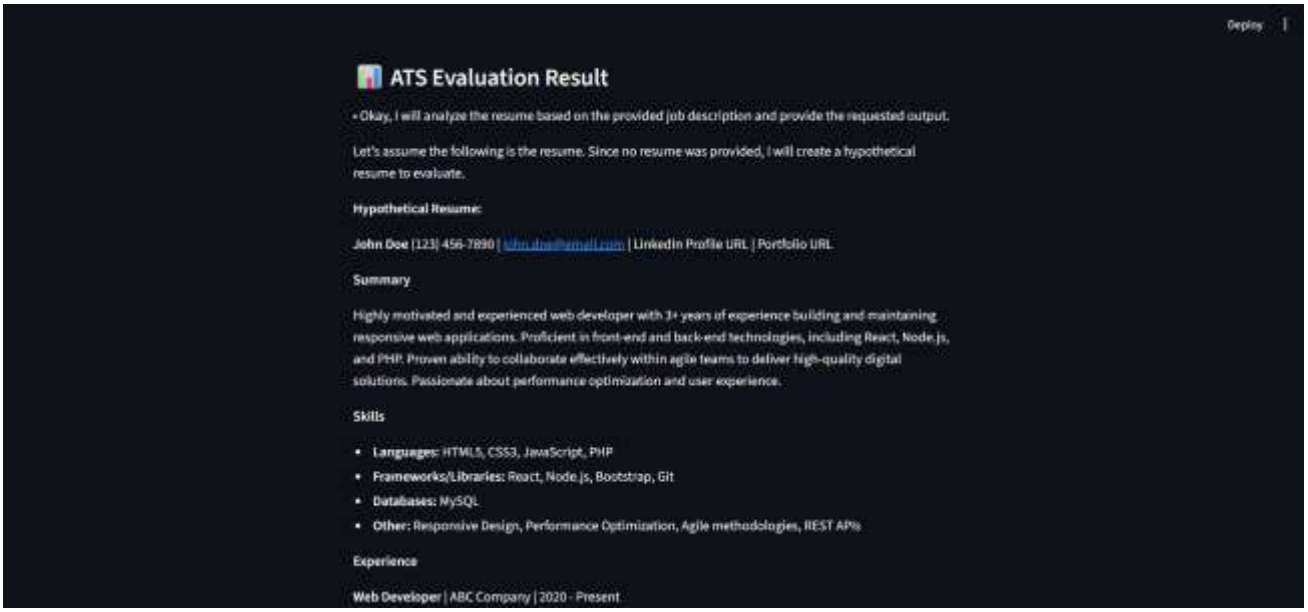
This page shows the results of a perfectly optimized resume for the ATS. The **"Resume Match Score"** is displayed as **100%**, indicating a perfect compatibility score. The results show no missing keywords, and the **"Profile Summary"** confirms a strong match between the candidate's experience and the requirements outlined in the job description, proving the resume is fully optimized for Applicant Tracking Systems.

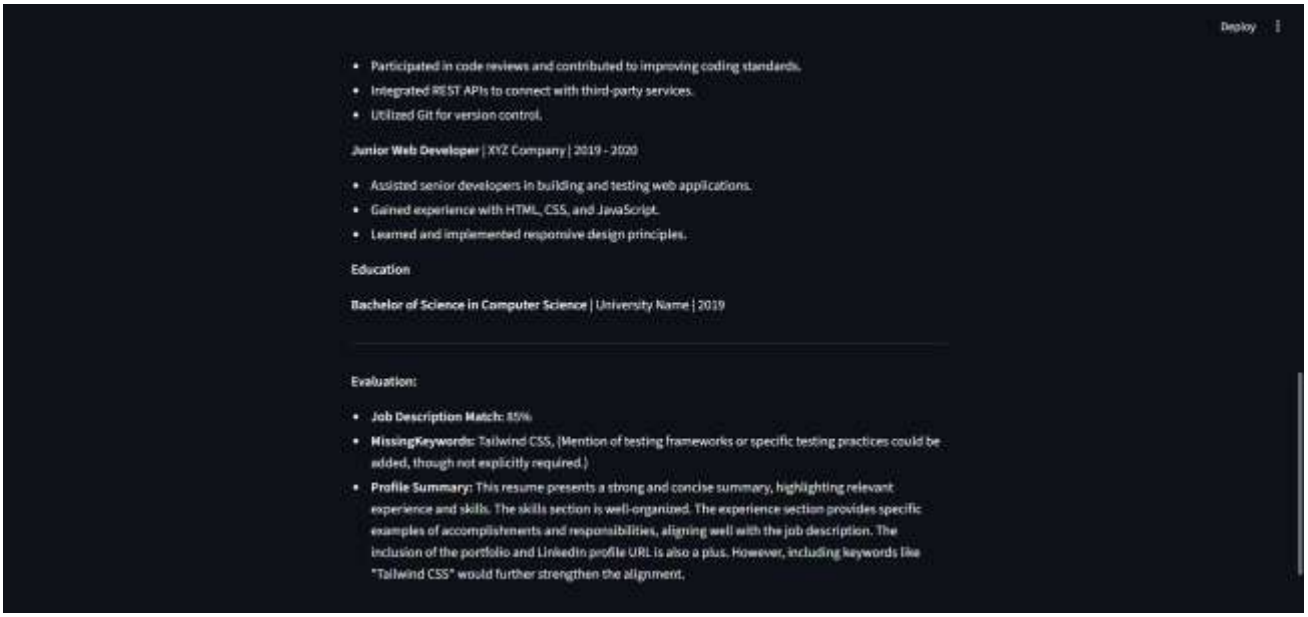




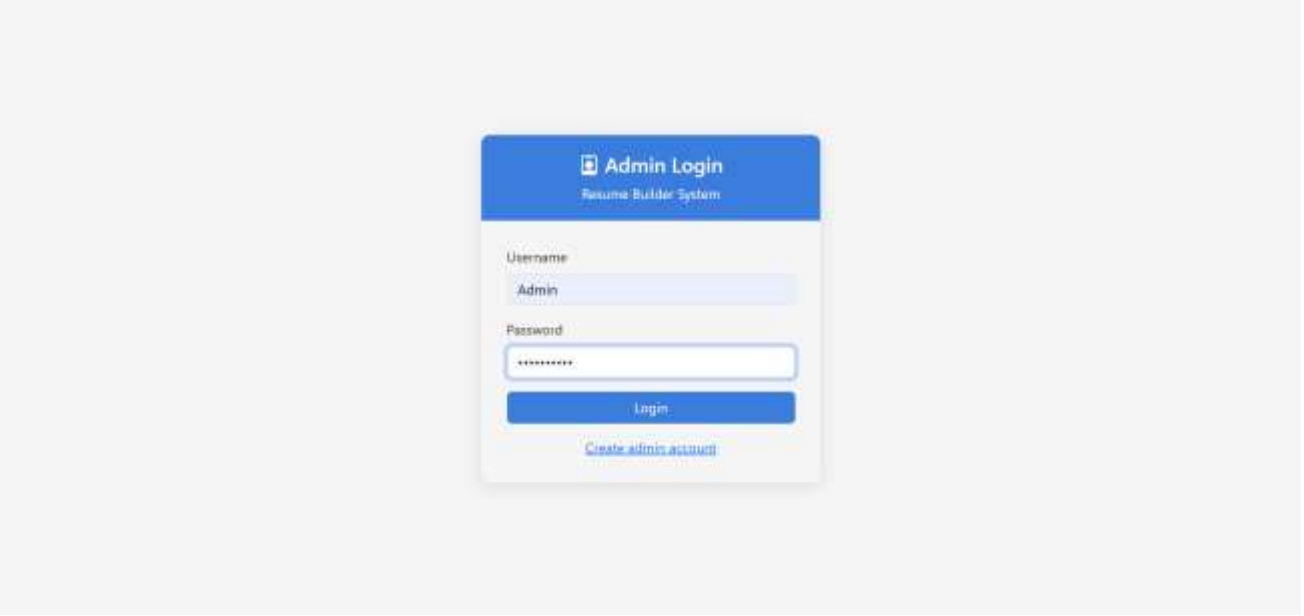
RESUME 3: ATS Checker Result: 85%

This page shows the results of a high-scoring but not perfect ATS check. The **"Job Description Match"** is **85%**, indicating the resume is well-structured and aligns closely with the job description. The **"Profile Summary"** confirms a strong, well-organized resume, but the **"Missing Keywords"** section highlights minor areas for improvement, such as mentioning specific testing frameworks or "Tailwind CSS," which would further strengthen the match.

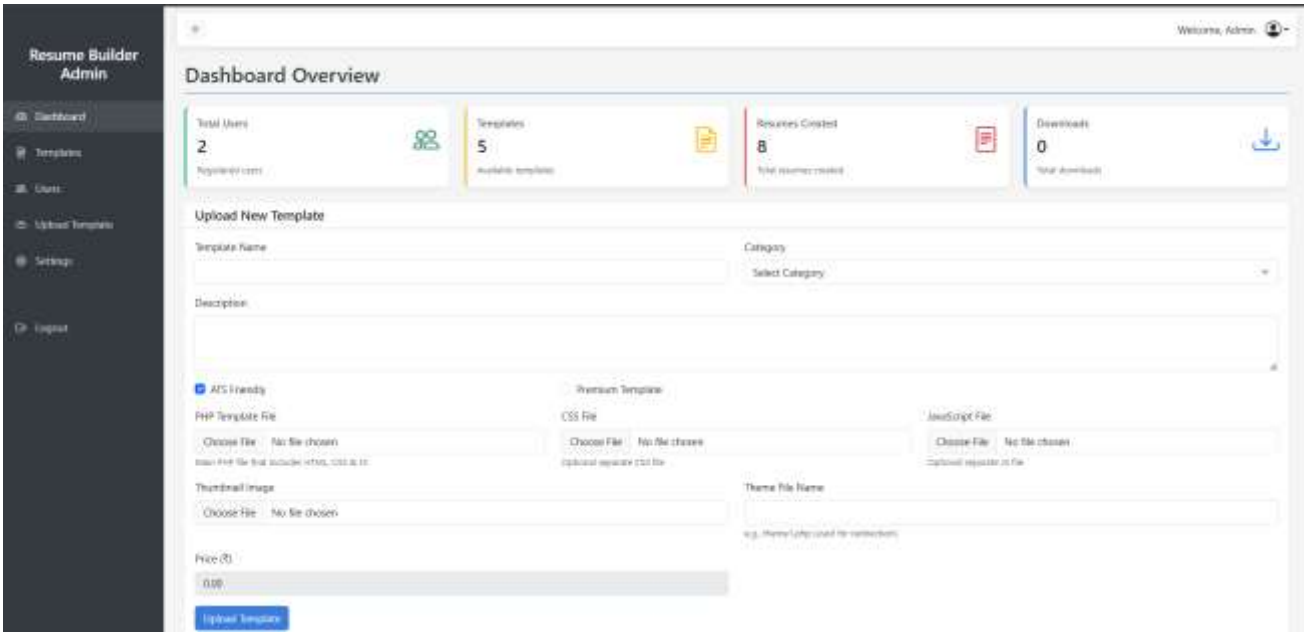




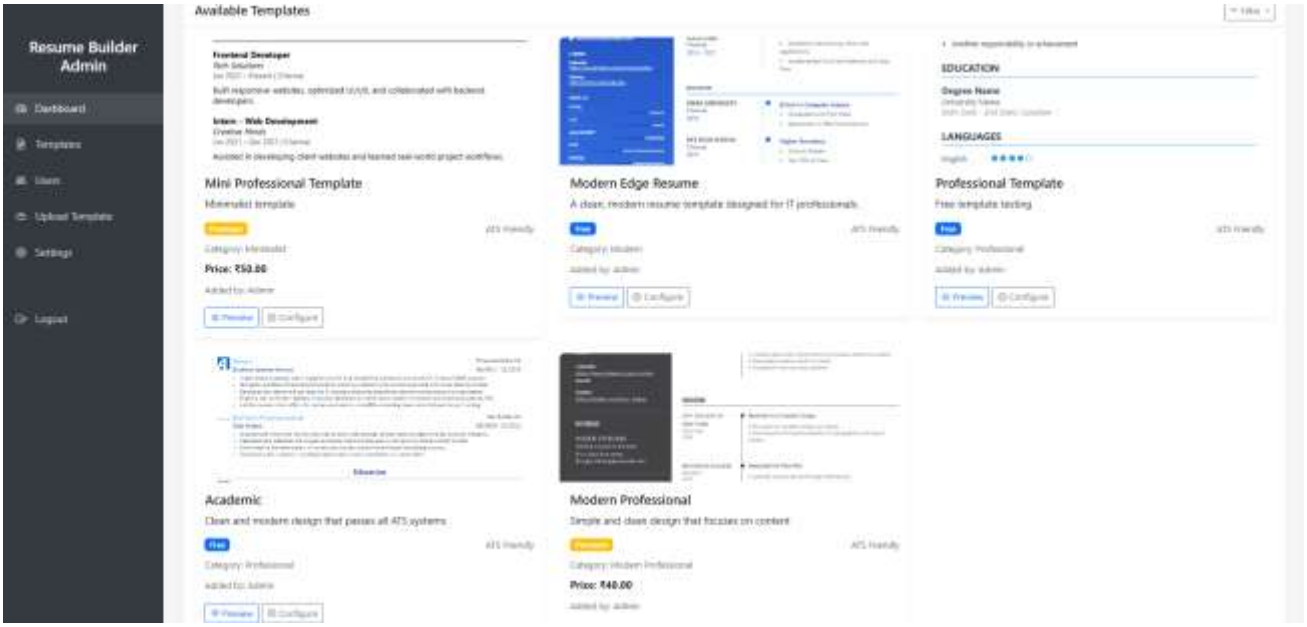
ADMIN LOGIN: This is the secure login page for the administrator of the Resume Builder System. It requires the admin to enter a **username** and **password** to gain access to the backend management system. The page also includes a link to create an admin account.



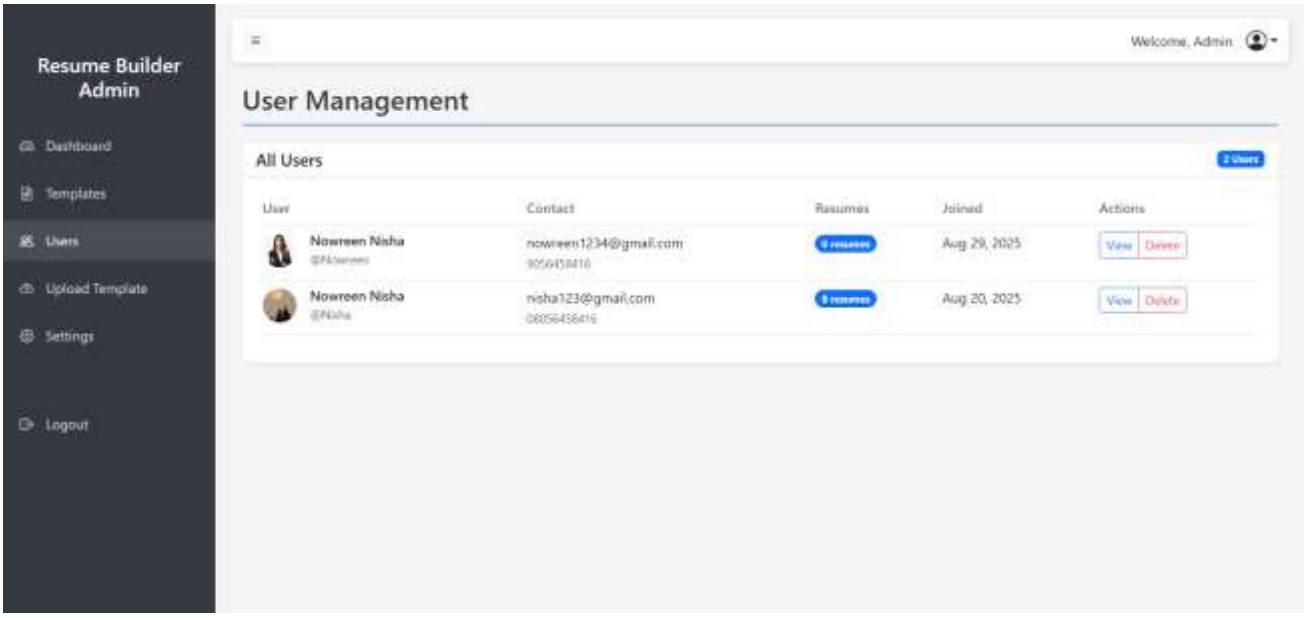
Admin Dashboard: This is the main control panel for the administrator of the Resume Builder System. The dashboard provides a high-level overview of key metrics, including the **Total Users**, **Available Templates**, **Resumes Created**, and **Total Downloads**. The left-hand menu provides quick navigation to manage templates, users, and system settings, while the main area allows the admin to upload new templates by filling out a form with details such as template name, description, and category.



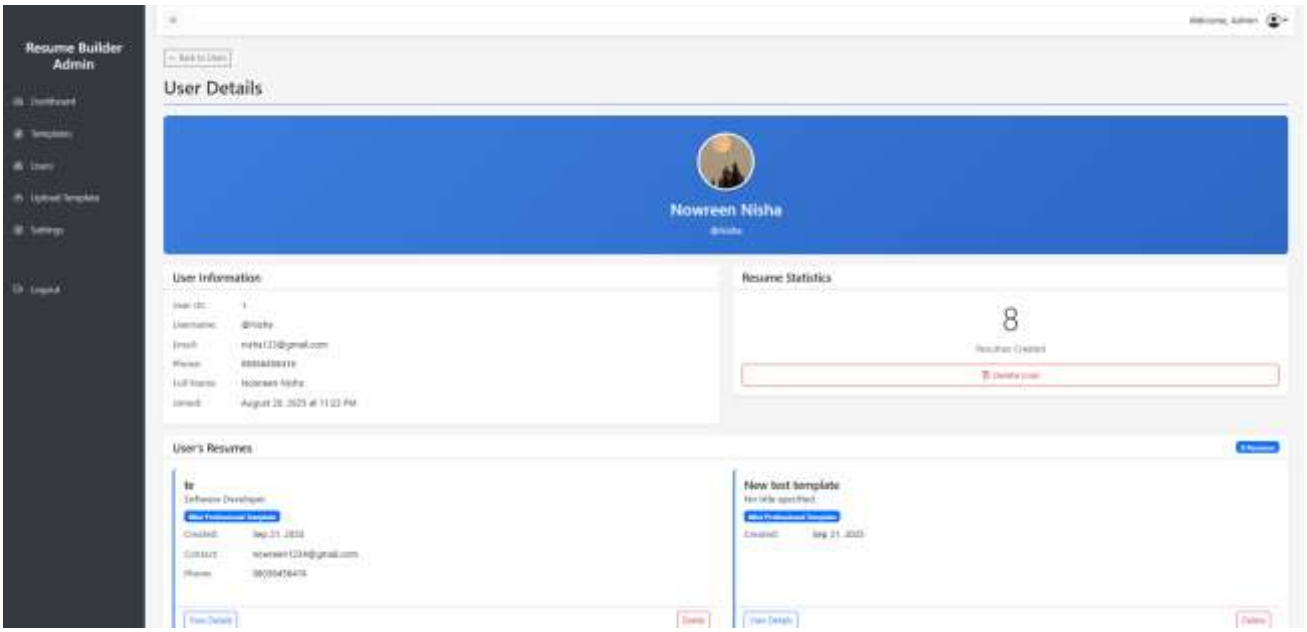
Upon scrolling down the Admin Dashboard, the administrator can view a list of all available templates. This page provides a clear overview of each template, including its name, a brief description, category, and pricing (free or premium). The admin can also see which templates are **"ATS Friendly."** For each template, there are options to **preview** and **configure** it, allowing for detailed management of the templates uploaded by the admin.



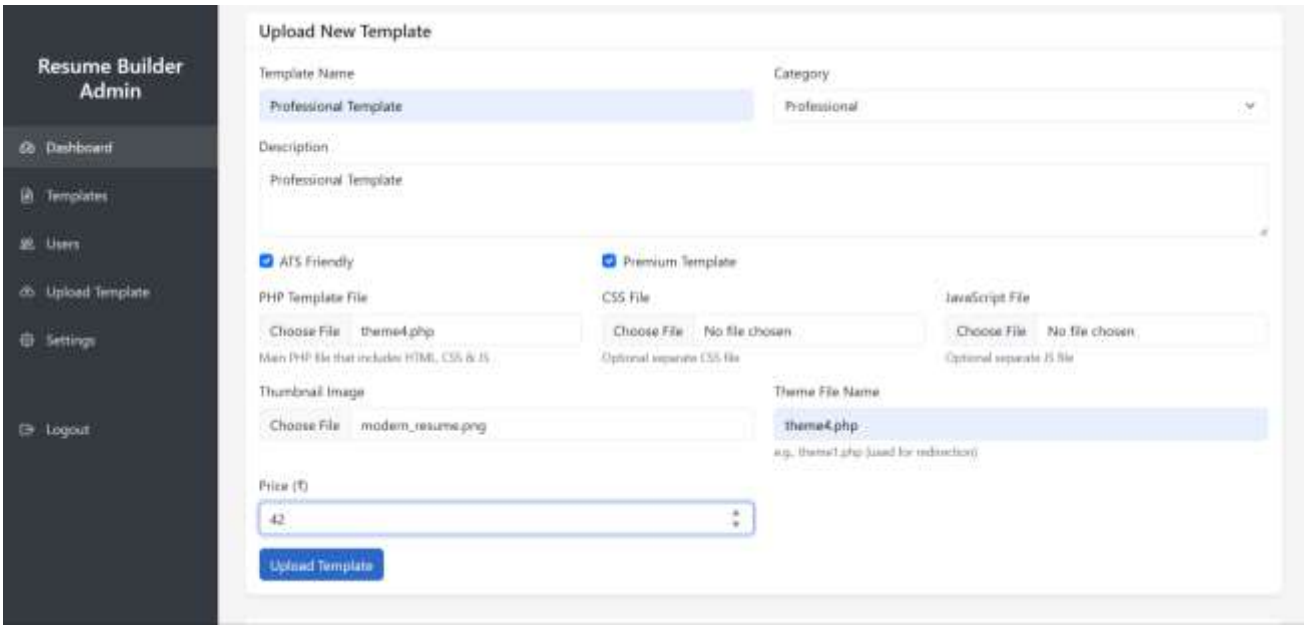
Admin Dashboard (User Management): This page allows the administrator to manage all user accounts. It provides a table showing a list of **"All Users"** on the platform. For each user, the admin can view their profile picture, username, email, contact information, the number of resumes they've created, and their join date. The admin has two key actions for each user: **"View"** to see more details and **"Delete"** to remove the account.



USER DETAILS: This page provides a detailed view of a specific user's account for the administrator. It displays User Information such as the user's ID, username, full name, email, and phone number. The admin can also see Resume Statistics, including the total number of resumes the user has created. Below these sections, a list of the user's Resumes is displayed, allowing the admin to view each resume's title, creation date, and contact info, with options to **"View Details"** or **"Delete"** each one. There is also a button to "Delete User" to completely remove the account.



Admin Dashboard (Uploading Templates): This page allows the administrator to add new resume templates to the system. The admin can fill out a form to provide all the necessary details for a new template. These details include the **Template Name**, **Category**, and **Description**. The admin can also specify if the template is **ATS-friendly** and if it is a **Premium Template**, in which case they can set a price. Finally, the admin uploads the template's files, such as the PHP, CSS, and JavaScript files, along with a thumbnail image.



CHAPTER 6

TESTING

Software testing is a critical phase in the development process used to identify and correct defects before deployment. The *Resume Builder Based on Applicant Tracking System* underwent multiple levels of testing to ensure the system's functionality, reliability, and performance met the expected standards. Testing confirmed that all modules — user, admin, and backend components — operated smoothly and provided accurate results.

The testing process was divided into three main categories: **Unit Testing**, **System Testing**, and **Validation Testing.**

6.1 Unit Testing

Unit testing was carried out during the development phase to verify the functionality of individual components and modules. Each unit or function was tested in isolation to ensure that it produced the correct output for a given input. The process involved testing both **front-end and back-end components,** such as user registration, login authentication, template uploads, PDF generation, and ATS score calculation.

AI-Based ATS Scoring Integration

The ATS Checker Module was enhanced using the Google AI Studio API, which analyzes resume content against job descriptions to calculate an ATS compatibility score. Unit testing focused on verifying the accuracy and reliability of API integration, ensuring that data sent to and received from the API was correctly processed and displayed in the user interface.
Testing activities included:

- Verified successful connection between the PHP backend and the Google AI Studio API using secure API keys.
- Tested JSON data exchange, ensuring that the resume text and job description were properly passed to the AI model for analysis.
- Checked that the ATS score, keyword suggestions, and analysis feedback returned from the API were accurately parsed and displayed on the user dashboard.

- Ensured the score values updated dynamically and reflected correctly in user reports and PDF exports.

The results confirmed that the AI-based ATS scoring worked precisely as intended, producing consistent and meaningful scores for diverse resume formats and job descriptions.

Key Units Tested

1. **User Registration & Login Module**
    - Verified proper storage of user data in the database.
    - Checked password encryption and session handling.
    - Ensured that invalid or duplicate email IDs triggered appropriate error messages.
2. **Resume Creation Module**
    - Tested form data validation for mandatory fields like name, education, and skills.
    - Ensured template data was correctly rendered in the PDF output.
    - Verified resume data saved and updated accurately in the database.
3. **ATS Checker Module**
    - Tested keyword matching functionality using sample job descriptions and resumes.
    - Ensured percentage scores reflected accurate keyword density and structure.
    - Checked Python script integration with backend through PHP execution.
4. **Template Management (Admin Panel)**
    - Verified upload, edit, and delete operations for resume templates.
    - Ensured that premium template pricing and status changes reflected correctly.
    - Checked template preview before activation.

Each unit test confirmed that the modules functioned independently without logical or syntax errors. Tools such as manual testing, debug logs, and browser console checks were used extensively during this phase.

6.2 System Testing

System testing was performed after integrating all modules to validate that the entire application functioned as a unified system. It focused on the interaction between modules and overall

system performance under normal and boundary conditions.

Objectives of System Testing

- To verify the correct integration of user and admin functionalities.
- To ensure smooth data flow between front-end and back-end layers.
- To confirm that all system features performed as per the defined requirements.

Testing Scenarios

1. **End-to-End Workflow**
    - A user registers, logs in, creates a resume, checks ATS compatibility, downloads the PDF, and logs out successfully.
    - Admin logs in, reviews templates, manages users, and monitors system activity.
2. **Database Connectivity**
    - Ensured all CRUD (Create, Read, Update, Delete) operations worked correctly.
    - Checked that foreign key constraints (users → resumes, templates) were properly maintained.
3. **Security Testing**
    - Verified session handling and restricted access for unauthorized users.
    - Tested protection against SQL injection and form tampering.
    - Checked encrypted storage of passwords.

4. **Performance Testing**
    - Tested the time taken to generate resumes and ATS scores.
    - Verified that the system remained responsive with multiple concurrent users.

System testing confirmed that the integrated modules worked seamlessly. The application demonstrated stable performance across various browsers (Chrome, Edge, and Firefox) and devices.

6.3 Validation Testing

Validation testing ensured that the system fulfilled all the functional and non-functional requirements specified during system analysis. It aimed to verify whether the *Resume Builder* truly met user expectations and project goals.

Validation Objectives

- To confirm that the system performs all the functionalities defined in the project requirements.
- To validate user interface consistency, accuracy, and ease of navigation.
- To ensure output accuracy, particularly in resume generation and ATS scoring.

Validation Activities

1. **Requirement Verification**
    - Each functional requirement (resume creation, ATS check, PDF export, etc.) was tested against expected outcomes.
    - Verified that all mandatory fields were validated before submission.
2. **Output Validation**
    - Compared generated PDFs for layout accuracy and keyword retention.
    - Verified that ATS scores matched expected results for given test resumes.
    - Checked that downloaded resumes were formatted and stored correctly.
3. **Error Handling Validation**
    - Tested system response to invalid data inputs and missing fields.
    - Verified proper error and confirmation messages were displayed to users.

The validation phase confirmed that the *ATS-based Resume Builder* performed accurately, met design objectives, and provided a user-friendly experience. The system successfully passed all validation criteria and was deemed ready for deployment.

CHAPTER 7
CONCLUSION

This project successfully developed an ATS-based resume builder that addresses the common challenges candidates face in today's job market. The system simplifies resume creation by providing a structured, form-based interface and generates professional, ATS-friendly PDFs. By integrating features such as template selection, live preview, and the ability to save user data, the system enhances the user experience and increases a candidate's visibility to recruiters. The inclusion of a dedicated ATS checker provides a unique and valuable tool for users to optimize their resumes for keyword relevance, thereby improving their chances of passing the initial screening phase. The modular design, with separate functionalities for users and administrators, ensures a secure, scalable, and easily maintainable platform. Overall, the project delivers a robust solution that not only simplifies resume building but also empowers job seekers to create "job-winning resumes" that are tailored for modern recruitment processes.

CHAPTER 8

REFERENCE

BOOK REFERENCES

1. **Robin Nixon**, "Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5," O'Reilly Media, 2018.

2. **Jon Duckett**, "HTML & CSS: Design and Build Websites," John Wiley & Sons, 2011.

3. **Wes McKinney**, "Python for Data Analysis," O'Reilly Media, 2017.

WEBSITE REFERENCES

- W3Schools
- Stack Overflow
- TutorialsPoint
- PHP Documentation
- MySQL Documentation
- Google AI Studio

CHAPTER 9

FUTURE SCOPE

1. AI-Powered Resume Suggestions

Integrate artificial intelligence or natural language processing (NLP) models to analyze user data and provide smart suggestions for improving resume content, such as adding relevant keywords, enhancing phrasing, and highlighting achievements based on the target job description.

2. Job Portal Integration

Expand the system by connecting it with job listing APIs or career platforms (like LinkedIn or Indeed), allowing users to directly apply for jobs using their generated ATS-friendly resumes within the platform.

3. Multilingual Resume Support

Add functionality to create and translate resumes in multiple languages, enabling users from different regions to prepare job applications that meet international recruitment standards.

4. Mobile Application Development

Develop a cross-platform mobile app version of the Resume Builder to provide users with the convenience of creating, editing, and downloading resumes on their smartphones anytime and anywhere.

CHAPTER 10

APPENDIX

10.1 SAMPLE SOURCE CODE

HOME

```
<?php
session_start();

// Check if the user is logged in
$isLoggedIn = isset($_SESSION['user_id']);
$username = $isLoggedIn ? $_SESSION['username'] : '';

// Logout functionality
if (isset($_GET['logout'])) {
session_destroy();
header("Location: index.php");
exit();
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0" name="viewport">
<title>Index - Resume Builder</title>
<meta name="description" content="">
<meta name="keywords" content="">

<!-- Favicons -->
<link href="assets/img/favicon.png" rel="icon">
<link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

<!-- Fonts -->
<link href="https://fonts.googleapis.com" rel="preconnect">
<link href="https://fonts.gstatic.com" rel="preconnect" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,
900;1,100;1,300;1,400;1,500;1,700;1,900&family=Lato:ital,wght@0,100;0,300;0,400;0,700;0,900;1,
100;1,300;1,400;1,700;1,900&family=Inter:wght@100;200;300;400;500;600;700;800;900&display=
swap" rel="stylesheet">

<!-- Vendor CSS Files -->
<link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
<link href="assets/vendor/aos/aos.css" rel="stylesheet">
```

```
<link href="assets/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
<link href="assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

<!-- Main CSS File -->
<link href="assets/css/main.css" rel="stylesheet">

<style>
/* Custom styles for the user dropdown */
.user-dropdown {
margin-left: 15px;
}
.user-dropdown .btn {
padding: 8px 15px;
border-radius: 6px;
font-weight: 500;
}
.user-dropdown .dropdown-menu {
border-radius: 8px;
box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
border: none;
padding: 10px;
}
.user-dropdown .dropdown-item {
padding: 10px 15px;
border-radius: 6px;
margin: 4px 0;
transition: background-color 0.2s;
}
.user-dropdown .dropdown-item:hover {
background-color: #f8f9fa;
}
.user-dropdown .dropdown-divider {
margin: 8px 0;
}
</style>

</head>

<body class="index-page">

<header id="header" class="header sticky-top">
<div class="container-fluid container-xl position-relative">

<div class="top-row d-flex align-items-center justify-content-between">
<a href="index.html" class="logo d-flex align-items-center">
<h1 class="sitename">Resume Builder</h1>
</a>

<div class="d-flex align-items-center">
<div class="social-links">
```

```
<a href="#" class="facebook"><i class="bi bi-facebook"></i></a>
<a href="#" class="twitter"><i class="bi bi-twitter"></i></a>
<a href="#" class="instagram"><i class="bi bi-instagram"></i></a>
<a href="admin/admin_register.php" class="admin"><i class="bi bi-person-circle"></i></a>
</div>

<!-- User Dropdown (shown when logged in) -->
<?php if ($isLoggedIn): ?>
<div class="user-dropdown">
<div class="dropdown">
<button class="btn btn-outline-primary dropdown-toggle" type="button" id="userDropdown" data-bs-toggle="dropdown" aria-expanded="false">
<i class="bi bi-person-circle me-1"></i> <?php echo htmlspecialchars($username); ?>
</button>
<ul class="dropdown-menu dropdown-menu-end" aria-labelledby="userDropdown">
<li><a class="dropdown-item" href="dashboard.php"><i class="bi bi-speedometer2 me-2"></i>Dashboard</a></li>
<li><a class="dropdown-item" href="profile.php"><i class="bi bi-person me-2"></i>Profile</a></li>
<li><hr class="dropdown-divider"></li>
<li><a class="dropdown-item" href="?logout=true"><i class="bi bi-box-arrow-right me-2"></i>Logout</a></li>
</ul>
</div>
</div>
<?php endif; ?>
</div>
</div>

</div>

<div class="nav-wrap">
<div class="container d-flex justify-content-center position-relative">
<nav id="navmenu" class="navmenu">
<ul>
<li><a href="#hero" class="active">Home</a></li>
<li><a href="templates.php">Templates</a></li>
<li><a href="#faq" class="active">FQA</a></li>
<li><a href="ats-checker.php">ATS</a></li>

</ul>
<i class="mobile-nav-toggle d-xl-none bi bi-list"></i>
</nav>
</div>
</div>

</header>

<main class="main">
```

```
<!-- Hero Section -->
<section id="hero" class="hero section dark-background">

<div class="container" data-aos="fade-up" data-aos-delay="100">

<div class="row align-items-center gy-5">
<div class="col-lg-7">
<div class="hero-card shadow-sm" data-aos="fade-right" data-aos-delay="150">
<div class="eyebrow d-inline-flex align-items-center mb-3">
<i class="bi bi-stars me-2"></i>
<span>Create Professional Resumes That Beat the ATS</span>
</div>
<div class="content">
<h2 class="display-5 fw-bold mb-3">Build a job-winning resume in minutes.</h2>
<p class="lead mb-4">Choose from modern templates, tailor your resume to job descriptions, and check your ATS score instantly to ensure your application gets noticed.</p>
<div class="d-flex flex-wrap gap-3">
<a href="#about" class="btn btn-primary-ghost">
<span>Explore Now</span>
<i class="bi bi-arrow-right ms-2"></i>
</a>
<?php if (!$isLoggedIn): ?>
<a href="register.php" class="btn btn-primary-ghost">
<span>Register Now</span>
<i class="bi bi-arrow-right ms-2"></i>
</a>
<a href="login.php" class="btn btn-primary-ghost">
<span>Login</span>
</a>
<?php endif; ?>
<footer id="footer" class="footer position-relative">

<div class="container">
<div class="row gy-5">

<div class="col-lg-4">
<div class="footer-brand">
<a href="index.html" class="logo d-flex align-items-center mb-3">
<span class="sitename">Resume Builder</span>
</a>
<p class="tagline">Empowering professionals to create standout resumes with elegant designs, modern layouts, and tools that make job applications faster, easier, and more effective.</p>

<div class="social-links mt-4">
<a href="#" aria-label="Facebook"><i class="bi bi-facebook"></i></a>
<a href="#" aria-label="Instagram"><i class="bi bi-instagram"></i></a>
<a href="#" aria-label="LinkedIn"><i class="bi bi-linkedin"></i></a>
</div>
</div>
</div>
```

```
<div class="col-lg-6">
<div class="footer-links-grid">
<div class="row">
<div class="col-6 col-md-4">
<h5>Company</h5>
<ul class="list-unstyled">
<li><a href="#">About Us</a></li>
<li><a href="#">Faq</a></li>
<li><a href="#">ATS</a></li>
<li><a href="#">Templates</a></li>
</ul>
</div>
<div class="col-6 col-md-4">
<h5>Support</h5>
<ul class="list-unstyled">
<li><a href="#">Help Center</a></li>
<li><a href="#">Contact Us</a></li>
<li><a href="#">Privacy Policy</a></li>
<li><a href="#">Terms of Service</a></li>
</ul>
</div>
</div>
</div>
</div>

<div class="col-lg-2">
<div class="footer-cta">
<h5>Let's Connect</h5>
<a href="contact.html" class="btn btn-outline">Get in Touch</a>
</div>
</div>

</div>
</div>

<div class="footer-bottom">
<div class="container">
<div class="row">
<div class="col-12">
<div class="footer-bottom-content">
<p class="mb-0">© <span class="sitename">Resume builder</span>. All rights reserved.</p>
<div class="credits">
<!-- All the links in the footer should remain intact. -->
<!-- You can delete the links only if you've purchased the pro version. -->
<!-- Licensing information: https://bootstrapmade.com/license/ -->
<!-- Purchase the pro version with working PHP/AJAX contact form: [buy-url] -->
Designed by <a href="#">Nisha</a>
</div>
</div>
```

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

</footer>

<!-- Scroll Top -->
<a href="#" id="scroll-top" class="scroll-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>

<!-- Preloader -->
<div id="preloader">
<div></div>
<div></div>
<div></div>
<div></div>
</div>
<!-- Main JS File -->
<script src="assets/js/main.js"></script>
</body>
</html>
```