

APPLICANT TRACKING SYSTEM

A project report submitted in partial fulfillment of the
requirement for the award of the degree

OF

Master of Computer Application (MCA)

A Project Report:

Under the guidance of:
(Dr. A.K. Mohapatra)

Submitted By:

Name: Uzma Naaz

Name: Reeya Roy

Name: Deepak Kumar

Roll No.: 21MC515147

Roll No.: 21MC515193

Roll No.: 21MC515201

Name: Wabhishek Anand

Roll No.: 21MC515219



DR. SHYAMA PRASAD MUKHERJEE UNIVERSITY, RANCHI

Month March Year 2023

ACKNOWLEDGEMENT

A task or project cannot be completed alone. It requires the effort of many individuals. I take this opportunity to thank all those who helped me complete this project.

I would like to express my sincere thanks to **Dr. Anita Kumari (Co-ordinator), Dept. of MCA, DSPMU, Ranchi** who helped a lot in offered various prompt suggestions during this project timely and properly finished.

I express my sincere gratitude to my guide **Dr. A.K. Mohapatra ,Dept. of MCA, DSPMU, Ranchi** for giving us the opportunity to undergo this project. I further thank his for lending a helping hand when it comes to solving my problems related to the project. This project would not have been possible without his valuable time and support.

I am also thanks to other **Faculty members of Dept. of MCA, DSPMU, Ranchi** for helping me in the preparation of the project.

I can't forget to thanks my **friends** who are very helpful to me directly or indirectly during development of this project.

Name:- Uzma Naaz

Roll No:- 21MCA515147

Name:- Reeya Roy

Roll No:- 21MCA515193

Name:- Deepak Kumar

Roll No:- 21MCA515201

Name:- Wabhishek Anand

Roll No:- 21MCA515219

DECLARATION

This is to declare that this project report titled **APPLICANT TRACKING SYSTEM** has been developed by me along with other project members for the minor project of final semester of Master of Computer Application (MCA) of Dr. Shyama Prasad Mukherjee University, Ranchi

I declare that this project is my own work by the best of my knowledge and belief. This project has never been submitted to any University or Institute for the partial fulfillment of the award of any degree or diploma.

Name:- Uzma Naaz

Roll No:- 21MCA515147

Name:- Reeya Roy

Roll No:- 21MCA515193

Name:- Deepak Kumar

Roll No:- 21MCA515201

Name:- Wabhishek Anand

Roll No:- 21MCA515219

BONAFIDE CERTIFICATE

This is to certify that this project titled **Applicant Tracking System** is the bonafide work of **UZMA NAAZ, 21MCA515147, REEYA ROY ,21MCA515193 , DEEPAK KUMAR , 21MCA515201 , WABHISHEK ANAND , 21MCA515219** 3rd semester student of Master of Computer Application (MCA) of Dr. Shyama Prasad Mukherjee University, Ranchi that carried out the project work under my supervision.

SIGNATURE

NAME:

(GUIDE)

EXAMINER'S CERTIFICATION

The project report titled **Applicant Tracking System** of **UZMA NAAZ 21MCA515147, REEYA ROY 21MCA515193 , DEEPAK KUMAR 21MCA515201 , WABHISHEK ANAND 21MCA515219** 3rd semester student of Master of Computer Application (MCA) of Dr. Shyama Prasad Mukherjee University, Ranchi.

Is approved and is acceptable in quality and form.

**Examiner - I
(Signature)**

Name:

Designation:

**Examiner -II
(Signature)**

Name:

Designation:

UNIVERSITY DEPARTMENT CERTIFICATE

This is to declare that this project titled **Applicant Tracking System** submitted in partial fulfillment of the requirements for the degree of Master of Computer Application (MCA) of Dr. Shyama Prasad Mukherjee University, Ranchi.

Has worked under my supervision and that guidance and that no part of this report has been submitted for the award of any other Degree, Diploma, fellowship or other similar titles or prizes and that the work has not been published in any Journal or Magazine.

Name:- Uzma Naaz

Roll No:- 21MCA515147

Name:- Reeya Roy

Roll No:- 21MCA515193

Name:- Deepak Kumar

Roll No:- 21MCA515201

Name:- Wabhishek Anand

Roll No:- 21MCA515219

Certified

TABLE OF CONTENT

S.no	Topic	Page no
1.	Introduction	1-2
2.	Modules	3
3.	Data dictionary	4-5
4.	Data Flow Diagram	6-7
5.	Tools description	8-12
6.	Er Diagram	13
7.	Design	14
8.	Class Diagram	15
9..	Coding	16-49
10.	Output screen	50-53
11.	Limitation	54-56
12.	Future scope	57-59
13.	References	60

INTRODUCTION

An ATS is a software application that helps you manage the entire recruitment process, right from receiving resumes to hiring employees. The software helps you automate the process of screening & shortlisting candidates, applicant evaluation, scheduling interviews, managing the hiring process, background verification, and completing new-hire paperwork. ATS helps in sorting through resumes, to determine which ones are the best fit for the open positions. Applicant tracking systems evaluates your resume by checking for keywords or skill sets that fit the job description. The main function of an ATS is to provide a central location and database for a company's recruitment efforts. ATSs are built to better assist management of resumes and applicant information. Data is either received from internal applications via the ATS front-end, located on the company website or is extracted from applicants on job boards. The majority of job and resume boards (LinkedIn.com, Monster.com,, Indeed.com) have partnerships with ATS software providers to provide integration, parsing support and ease of data migration from one system to another.

When an applicant submits their resume for a job of their liking and interest, the ATS captures the applicant information. Information such as the applicant skill sets, experience, educational background, contact information, resume and covering letter can be uploaded into the system database.

As the hiring process progresses, this information can be seamlessly transferred from system to system. Recruiters can review applicant information, shortlist and review if found suitable for requirements. Applicants can be sent automated messages on receiving their application, as well as information with regard to interview schedules, communication of details such as date, time and place of interview, and progress of the recruitment process. Online testing can also be conducted to screen applicants. The ATS also helps close the loop in the hiring process. Rejection letters can be shared, and if selected, the new hires can connect the information to payroll and HR managers for onboarding. The ATS thereby, cohesively streamlines the recruiting and application process with other networks and systems in the organization.

The ATS can also scan and scout potential recruits using social media networks, and help in the recruitment process. The Applicant Tracking system can store resumes and create a database. This way, when a suitable opportunity comes up in an organization, recruiters use keywords to search out applicants from the database. The ATS matches the keywords and ranks resumes in the search results according to relevance. Keyword searches could include skills, experience, qualifications, location, specific to a particular job opening. This works well for both the applicant and the hiring managers, as the system sieves the entire database, even looking at applicants who have applied a long while back.

Streamlining the Process

The ATS helps reduce errors in the recruitment process, saving time, money and effort. Risks involved in deletion of mails, or misplacing files connected to candidates are entirely eliminated. Certain systems even have provision for candidates to directly apply from their social network sites such as LinkedIn or Indeed. Vital information such as work history, education, specific qualifications, work references can be uploaded with ease. This information is automatically collected and organized digitally, making it easily available to all connected systems. HR managers, line managers, and hiring managers can seamlessly review candidates, reducing and even eliminating time involved in sorting, filing and collating the same information over and over. This helps job applicants, who no longer have to retype the same information for each application, making it easy to use and time saving. Why do employers use applicant tracking systems? Human errors such as deletion and misplacing files can be eliminated. The ATS help hiring managers stay organized, and on top of their game. The system streamlines the hiring process, handling both small and large volumes of applications. The ATS also helps to keep employers maintain connect with potential hires, by communicating with the applicants directly.

Tracking the process

The ATS also helps to build efficiency in the recruitment management process. The system allows employers to track how and where the candidate found a particular job posting. This helps to identify the most efficient means of job posting – be it a job board, the company website, referrals, consultants, or any other source. Percentage of employee applicants, as well as applicant to hire success rates can also be measured. This enables employers to focus and build in the venues where there has been most success, while reducing or improving efforts in others.

MODULES

1) User Authentication Module:

This module is responsible for handling the user authentication and authorization process. It provides a way for users to create an account, log in, and manage their personal information. The module may include features such as password reset, two-factor authentication, and role-based access control.

2) Resume Submission Module:

This module is responsible for handling the resume submission process. It provides a way for candidates to upload their resumes and other relevant information for job applications through the web application. The module may also include features such as resume formatting tools, file type validation, and resume preview options.

3) Resume Scoring Module:

This module is responsible for evaluating each resume based on job requirements and scoring each candidate based on their fit for the job. The algorithm could use a variety of factors to score each candidate, such as education level, years of experience, specific skills or certifications, and more. The module may also include features such as customizable scoring criteria, weightage adjustment, and real-time candidate ranking.

4) Feedback Module:

This module is responsible for providing feedback to candidates on areas where their experience or skills were lacking. The feedback can help candidates understand why they were not shortlisted for a particular job and provide suggestions on how they can improve their resume or job application. The module may include features such as customizable feedback templates, email notifications, and real-time feedback tracking.

5) Admin Dashboard Module:

This module is responsible for providing an interface for admins to manage the system. Admins can manage user accounts, job listings, and other system settings. The module may include features such as user role management, audit logs, and system analytics.

These modules work together to provide a comprehensive Applicant Tracking System that can help streamline recruitment and improve candidate experience.

DATA DICTIONARY

User

Field name	Datatype	Description
Id	Int	Unique identifier for each user
Name	String	Name of the user
Email	String	Email address of the user
Password	String	Password for user's account
Createdat	Datetime	Date and time when account was created

Admin

Field name	Datatype	Description
Id	Int	Unique identifier for each admin
Name	String	Name of the admin
Email	String	Email address of the admin
Password	String	Password for admin's account
Createdat	Datetime	Date and time when account was created

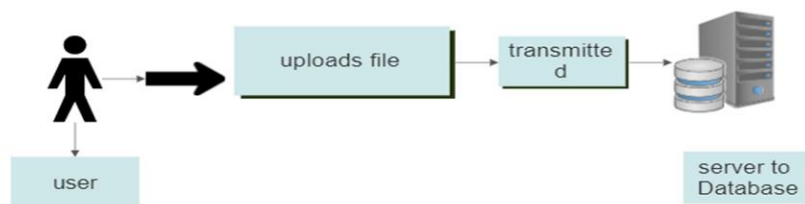
Resume

<u>Fieldname</u>	<u>Datatype</u>	<u>Description</u>
<u>Id</u>	<u>Int</u>	<u>Unique for each resume</u>
<u>UserId</u>	<u>Int</u>	<u>Id for user who uploaded resume</u>
<u>File</u>	<u>Binary</u>	<u>Binary data of upload resume</u>
<u>Filename</u>	<u>String</u>	<u>Mimetyoe of the uploaded resume</u>
<u>mimeType</u>	<u>String</u>	<u>Text extracted from resume</u>
<u>Parsedtext</u>	<u>Text</u>	<u>Text extracted from parsed tool</u>
<u>Score</u>	<u>Float</u>	<u>Score assigned</u>
<u>UploadedAt</u>	<u>datetime</u>	<u>Date and time when resume was uploaded</u>

Data flow Diagram(DFD)

Level 0 DFD:

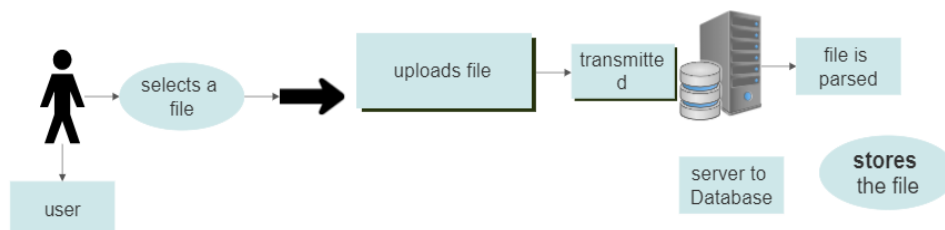
- User uploads a file
- File is transmitted to the server
- Server stores the file



DFD Level-0

Level 1 DFD:

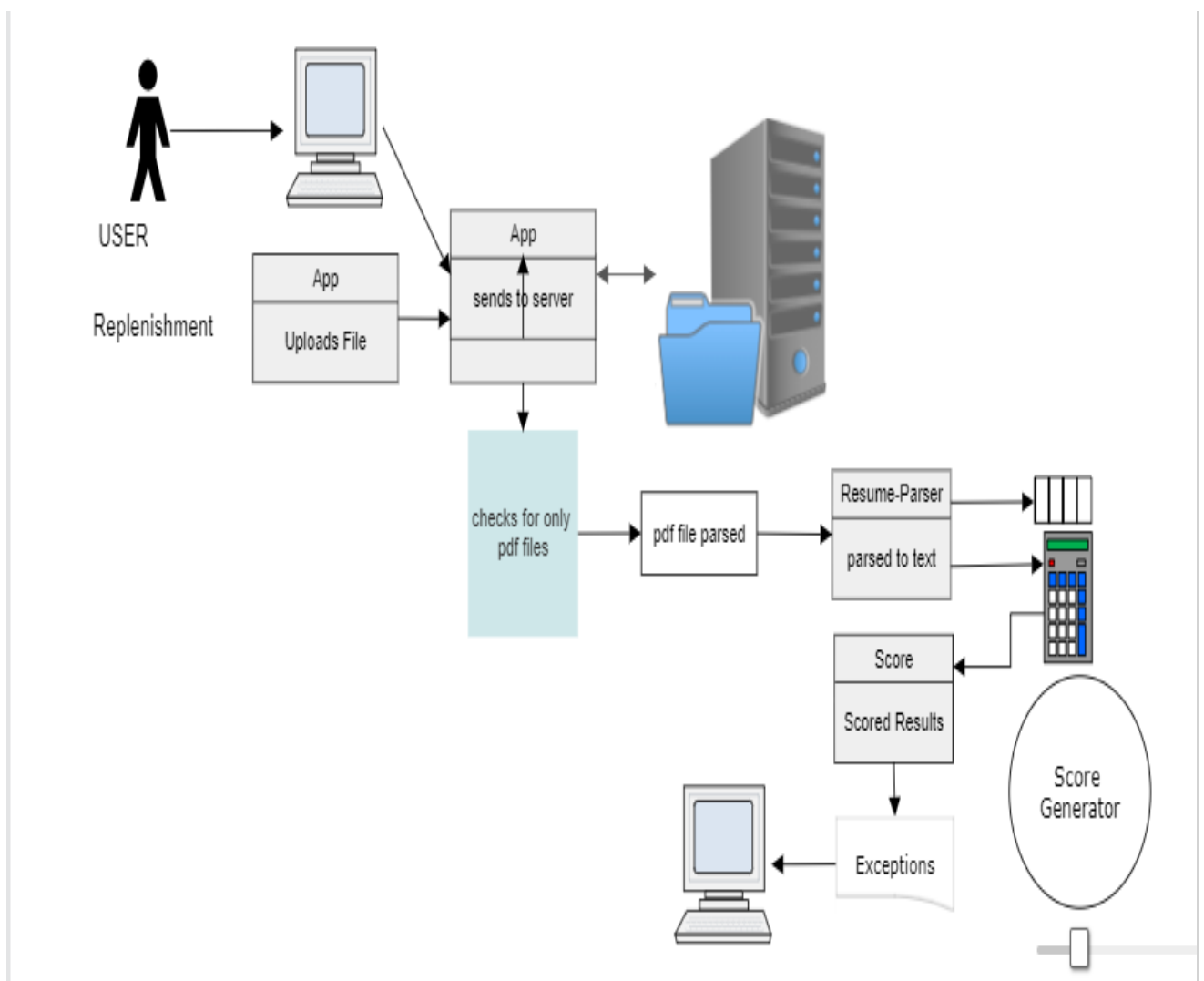
- User selects a file to upload
- User enters additional information (e.g. file name, description, tags)
- User submits the form
- Web application validates the input data
- Web application sends the file and metadata to the server
- Server stores the file and parse it to text



DFD Level-1

Level 2 DFD:

- Web application displays a form for file upload
- Web application validates user input and displays error messages if necessary
- Web application sends the file and metadata to the server
- Server stores the file and metadata in the appropriate location
- Server updates the database with file information
- The file is parsed to text for achieving scores.
- Scores are generated from the algorithm and checked from regex.



TOOLS DESCRIPTION

Bootstrap

Bootstrap is a popular front-end open-source framework used for creating responsive and mobile-first websites. It provides a set of pre-designed CSS and JavaScript components, such as buttons, forms, typography, navigation, modals, etc., that can be easily integrated into a web project. Bootstrap also supports customization, allowing developers to easily modify the pre-designed components to fit the specific needs of their projects. The framework is developed and maintained by Twitter and has a large community of developers contributing to its development and support. Some of the benefits of using Bootstrap include faster development, consistency across different devices and browsers, and easier maintenance of the website.

Css

CSS (Cascading Style Sheets) is a style sheet language used to describe the look and formatting of a document written in HTML, XHTML, or XML. It is used to control the presentation of web pages and makes it possible to separate content from presentation. With CSS, you can control the layout, colors, fonts, sizes, and other visual aspects of web pages.

CSS is a widely used technology on the web, and it provides a powerful and flexible way to style and design web pages. It offers a range of selectors, properties, and values that can be used to apply styles to specific elements on a web page. CSS can be applied inline within an HTML document, within the head section of an HTML document, or in a separate external file that is linked to the HTML document.

Html

HTML (Hypertext Markup Language) is a standard markup language used for creating web pages and web applications. It describes the structure of web pages using a set of tags and attributes that are used to define elements of a webpage such as text, images, videos, forms, and more. HTML is the backbone of the internet, and it is used to create and define the layout and structure of web pages.

HTML is a client-side language, meaning it is interpreted by the web browser rather than the server. This allows web developers to create dynamic and interactive web pages by using JavaScript and other client-side technologies in conjunction with HTML.

HTML is constantly evolving, with new versions being released periodically. The latest version of HTML is HTML5, which provides new features and improvements over previous versions. HTML5 includes support for multimedia elements, drag and drop, local storage, and more, making it easier to create rich and dynamic web applications.

Node.js

Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client's browser. Ryan Dahl developed it in 2009, and its latest iteration, was released in April 2021. Developers use Node.js to create server-side web applications, and it is perfect for data-intensive applications since it uses an asynchronous, event-driven model. Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop. Typically, behavior is defined through callbacks at the beginning of a script, and at the end a server is started through a blocking call like EventMachine run(). In Node.js, there is no such start-the-event-loop call. Node.js simply enters the event loop after executing the input script. Node.js exits the event loop when there are no

more callbacks to perform. This behavior is like browser JavaScript — the event loop is hidden from the user.

HTTP is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Node.js well suited for the foundation of a web library or framework.

Node.js being designed without threads doesn't mean you can't take advantage of multiple cores in your environment. Child processes can be spawned by using our `child_process.fork` API, and are designed to be easy to communicate with. Built upon that same interface is the cluster module, which allows you to share sockets between processes to enable load balancing over your cores

Express.js

Express.js or simply express is a back end web application framework for building RESTful APIs with Node.js , released as free and open source software under the MIT License. It is designed for building web application and APIs .It has been called the de facto standard server framework for Node .js. ExpressJS is a bridge between the front-end and back-end of a web application. It acts as the intermediary that processes data and information between the two, making it possible for users to interact with the application in real-time.

This framework also includes a range of helpful features, such as middleware, routing, and templating, that make it easier to develop and maintain a web application.

Technically speaking, ExpressJS is a free and open-source web application framework that is built on top of Node.js. It's designed to simplify the process of building and maintaining server-side applications. With ExpressJS, you can create dynamic and interactive websites, build RESTful APIs, and handle HTTP requests and responses in an organized and efficient manner.

ExpressJS is a great choice for beginners and experienced web developers alike. Its simple yet powerful architecture makes it easy to learn and use, while its robust set of features makes it suitable for creating complex web applications.

Visual Studio Code(Vs-Code)

Visual Studio Code (often referred to as VS Code) is a free and open-source source code editor developed by Microsoft. It is available for Windows, macOS, and Linux operating systems.

VS Code includes features like syntax highlighting, code completion, debugging, and Git integration. It also supports many programming languages and frameworks such as JavaScript, Python, Java, and React.

One of the biggest strengths of VS Code is its large and active extension marketplace. There are thousands of extensions available that can enhance the functionality of the editor, from adding support for specific programming languages to integrating with cloud services.

Overall, VS Code is a popular choice for developers due to its flexibility, ease of use, and large community of users and developers.

MongoDB

MongoDB is a document-oriented NoSQL database system that allows for high performance, high availability, and automatic scaling of enterprise-level databases. Unlike traditional relational database management systems, MongoDB stores data in flexible, JSON-like documents, making data integration, querying, and indexing easier.

MongoDB provides a number of features that make it an attractive option for modern software development projects. These features include:

1. Flexible data model:

MongoDB uses a flexible document model, which allows developers to store data in a format that closely mirrors their object-oriented programming languages. This makes it easier to work with data in code, and allows for more natural and intuitive data modelling

2. Scalability:

MongoDB's architecture is designed for horizontal scaling, which means that it can easily handle large amounts of data and high levels of traffic. This makes it an ideal choice for modern web applications and other large-scale data-intensive projects.

3. High availability:

MongoDB includes built-in replication and automatic failover, which means that it can provide high levels of availability even in the face of hardware or network failures.

4. Rich query language:

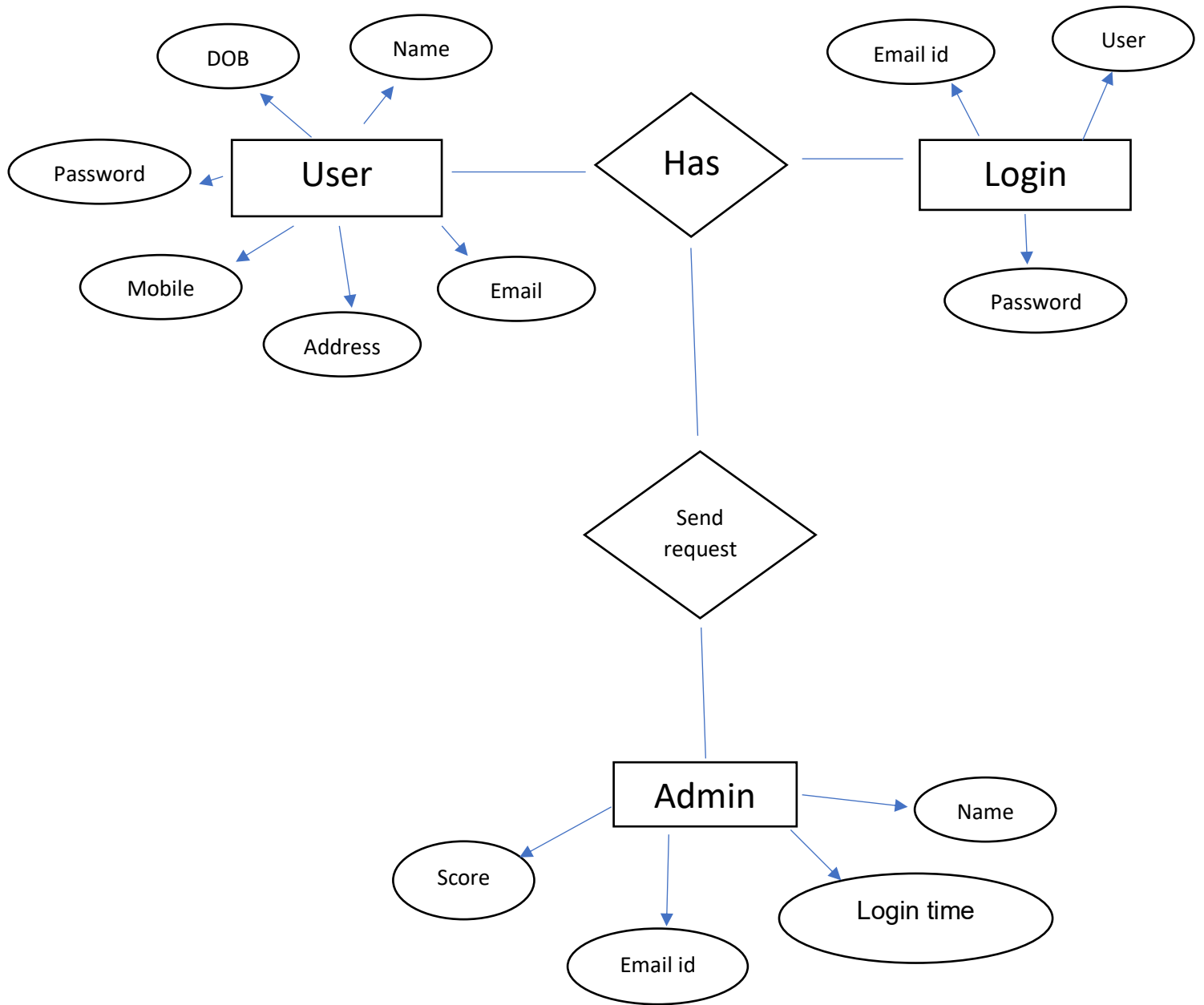
MongoDB provides a rich and powerful query language that supports a wide range of querying capabilities, including full-text search, geospatial queries, and more.

5. Open source:

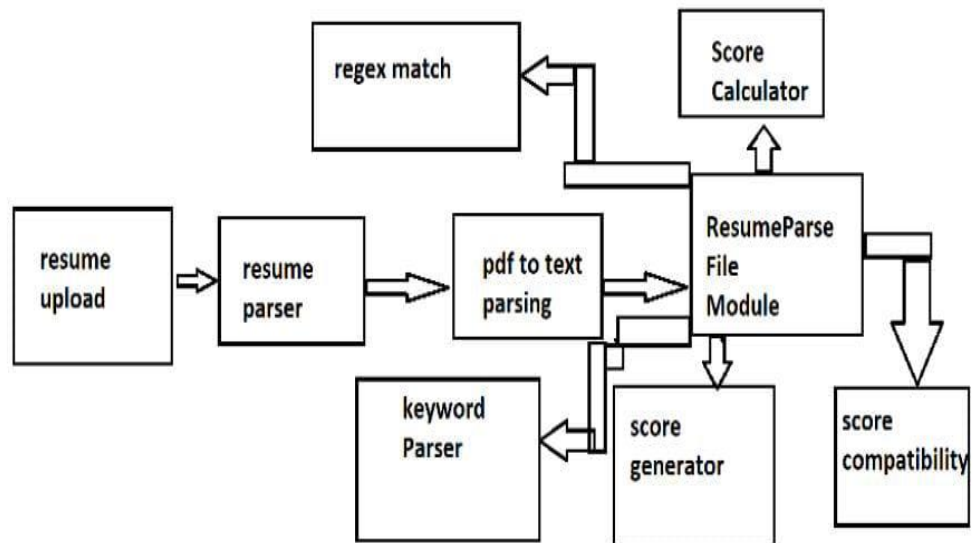
MongoDB is an open source database system, which means that it is free to use and can be customized and extended as needed. This makes it an ideal choice for startups and other organizations that need to keep costs low.

Overall, MongoDB is a powerful and flexible database system that provides a number of advantages over traditional relational databases. Its flexible data model, scalability, and high availability make it an ideal choice for modern software development projects.

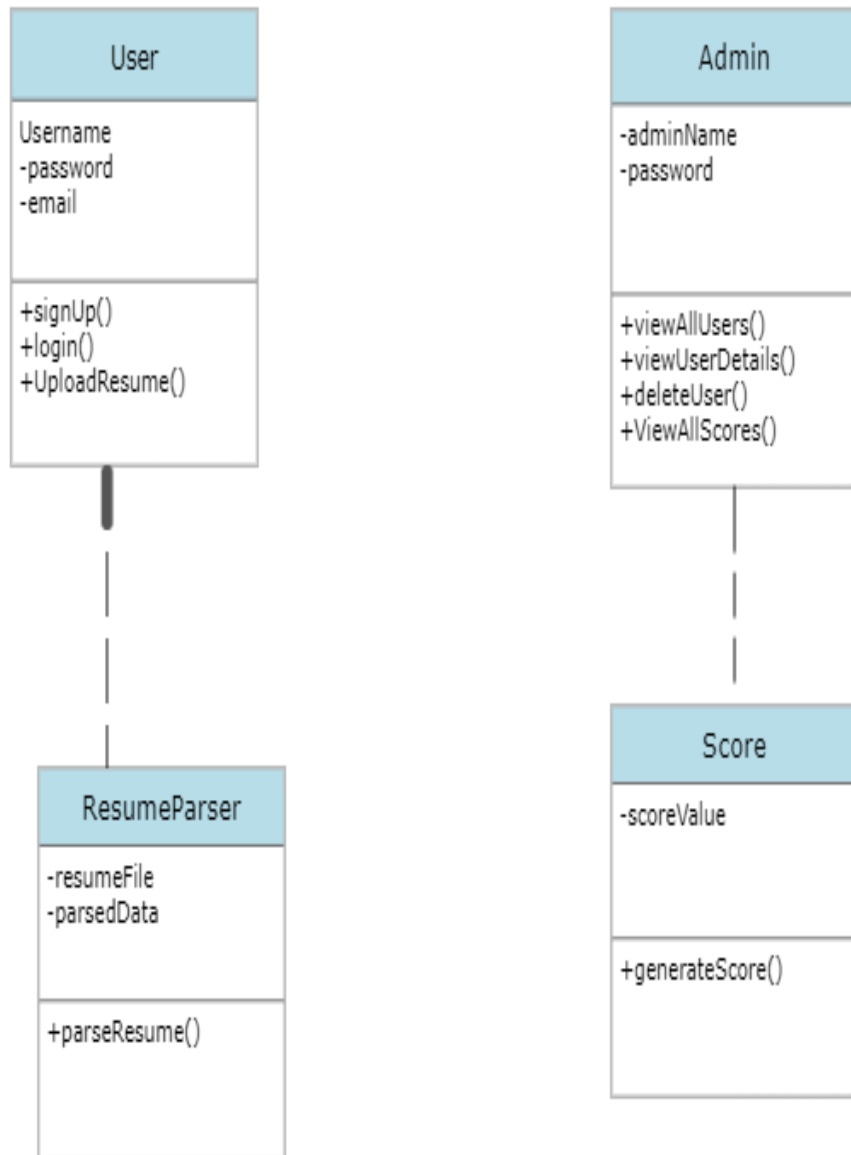
ER DAIGRAM



DESIGN



Class Diagram



CODING

1.INDEX.HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <meta http-equiv="X-UA-Compatible" content="ie=edge" />

  <title>Resume Checker - Your Next Job Interview Starts Here</title>

  <!-- Link to Tailwind CSS -->

  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.16/tailwind.min.css" />

</head>

<body>

  <!-- Navigation Bar -->

  <nav class="bg-white shadow-lg">

    <div class="container mx-auto px-6 py-3">

      <div class="flex items-center justify-between">

        <div>

          <a class="text-gray-800 font-bold text-xl" href="#">Resume
Checker</a>

        </div>

        <div class="flex items-center">

          <a class="text-gray-800 font-bold mx-3" href="#">Pricing</a>

        </div>

      </div>

    </div>

  </nav>


```

```

    <a class="text-gray-800 font-bold mx-3" href="#">Templates</a>

    <a class="text-gray-800 font-bold mx-3" href="#">Blog</a>

    <a class="text-gray-800 font-bold mx-3" href="signup.html">Sign up</a>

    <a class="bg-green-500 hover:bg-green-700 text-white font-bold py-2
px-4 rounded-full mx-3"

        href="login.html">Get Started</a>

    </div>

</div>

</div>

</nav>

<!-- Main Content -->

<div class="container mx-auto my-10 px-4">

    <h1 class="text-4xl font-bold text-gray-800 mb-10" style="text-align:center;

        font-weight: 700;box-sizing: border-box;display: block;

font-size: 2em;font-family: Rubik,Arial,Helvetica,serif;

letter-spacing: 0;

font-weight: 500;line-height: 36px!important;

margin: 0.67em 0;;font-size: 68px;max-width: 100%;margin-left: auto;

margin-right: auto; margin-block-start: 0.83em;display: block;margin-inline-start:

0px;

margin-inline-end: 0px;

margin-block-end: 0.83em;

line-height: 56px!important;">Is your resume good enough?

    </h1>

    <br><br>

```


<h3>

Our free and fast resume checker will give you the answer in just 5 minutes!

</h3>

<p class="text-xl font-bold text-gray-800 mb-10" style="font-family: Rubik,Arial,Helvetica,serif;

letter-spacing: 0;

font-weight: 500;">Check if your resume is a professional resume
 that stands out

in a sea of

applicants. Our scorer will
tell you where you lie and why your resume/cv are rejected.</p>

</div>

<div class="flex justify-center mt-8">

<form action="/upload" method="POST" enctype="multipart/form-data">

<label for="resume-upload"

class="py-2 px-4 bg-blue-500 hover:bg-blue-600 text-white font-semibold rounded-md shadow-md cursor-pointer"

id="upload-btn" type="submit">

Upload Resume

</label>

<input id="resume" type="file" class="hidden" name="resume" accept=".pdf,.doc,.docx">>

</div>

</form>

<div class="container mx-auto">

<h2 class="text-3xl font-bold mb-4">How we can help you</h2>

<div class="grid grid-cols-1 md:grid-cols-2 gap-4">

<div class="bg-white p-6 rounded-lg shadow-md">

<h3 class="text-xl font-bold mb-2">Make sure your resume stands out for the right reasons!</h3>

<p class="text-gray-700 text-lg leading-relaxed">With our expert review and editing services, we'll help

you highlight your strengths and accomplishments in a way that catches the eye of potential

employers.</p>

</div>

<div class="bg-white p-6 rounded-lg shadow-md">

<h3 class="text-xl font-bold mb-2">Eliminate embarrassing mistakes and typos</h3>

<p class="text-gray-700 text-lg leading-relaxed">Our website will review your resume

for grammar, spelling, and punctuation errors, as well as consistency and clarity. You can rest

assured that your resume will be error-free.</p>

</div>

<div class="bg-white p-6 rounded-lg shadow-md">

<h3 class="text-xl font-bold mb-2">Sell yourself effectively by strengthening your content</h3>

<p class="text-gray-700 text-lg leading-relaxed">We'll work with you to highlight your skills and

achievements in a way that grabs the attention of hiring managers. By optimizing your content,

you'll be able to showcase your qualifications and stand out from the competition.</p>

</div>

<div class="bg-white p-6 rounded-lg shadow-md">

<h3 class="text-xl font-bold mb-2">Improve your design and create a visually appealing resume</h3>

<p class="text-gray-700 text-lg leading-relaxed">Our team of designers can help you create a resume that

is visually appealing and easy to read. We'll ensure that your resume looks professional and modern,

while still highlighting your unique skills and qualifications.</p>

</div>

</div>

</div>

```

<br>

<br>

<br>

</div>

<div class="flex flex-wrap items-center">

  <div class="w-full lg:w-2/3">

    <h2 class="text-2xl font-bold mb-4">Make ATS Friendly Resume Tips:</h2>

    <p class="text-lg mb-4">

      <span class="text-green-500">&#10003;</span> Full ATS compatibility
check<br>

      We'll parse your resume and look for signs of ATS compatibility, such as
the presence of skills and

      other keywords, your contact data, file type, and length. Then, we'll give
you suggestions on how to

      improve your resume.

    </p>

    <p class="text-lg mb-4">

      <span class="text-green-500">&#10003;</span> Eliminate visual
mistakes<br>

      Is your resume too crowded or visually unappealing? Resume Checker
looks for design flaws and recommends

      a better template based on your job title.

    </p>

    <p class="text-lg mb-4">

      <span class="text-green-500">&#10003;</span> Remove
inconsistencies<br>

```

Designed to look for the finer details of your resume, Resume Checker will identify misspellings,

grammar and punctuation errors so you won't leave your application to chance.

</p>

<p class="text-lg mb-4">

⚠️ Repetition of phrases and cliches

Noticed a blatant word repetition in your resume only a couple of minutes after you sent the email?

Resume Checker is designed to look after and highlight the repetitive verbs, phrases and cliches we

programmatically reach for when we're lost for words.

</p>

<p class="text-lg mb-4">

⚠️ Filler, weak or vague wording

Words like "many", "much", "little", "a lot of" and their synonyms undermine the quality of your resume

as they don't provide the full picture of your impact. Resume Checker takes a stab at the

not-performance-oriented experience and reminds you to include an actionable result, preferably with a

number or metric achieved.

</p>

</div>

<div class="w-full lg:w-1/3 lg:order-first">

```

```

```
</div>
```

```
</div>
```

```
</div>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<hr>
```

```
<!-- Footer -->
```

```
<footer class="bg-gray-200 py-10">
```

```
<div class="container mx-auto px-4">
```

```
<p class="text-xl font-bold text-gray-800 mb-2">Connect With Us</p>
```

```
<div class="flex items-center justify-center my-4">
```

```
<a class="text-gray-800 font-bold mx-3" href="#">Facebook</a>
```

```
<a class="text-gray-800 font-bold mx-3" href="#">Twitter</a>
```

```
<a class="text-gray-800 font-bold mx-3" href="#">Instagram</a>
```

```
<a class="text-gray-800 font-bold mx-3" href="#">LinkedIn</a>
```

```
</div>
```

```
<p class="text-gray-600">Resume Checker &copy; 2023</p>
```

```
</div>
```

```
</footer>
```

```
<script>
```

```
window.addEventListener('load', function () {
```

```

const uploadBtn = document.getElementById('upload-btn');

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

    event.preventDefault();

    const file = fileInput.files[0];

    const formData = new FormData();

    formData.append('resume', file);

    fetch('/uploads', {

        method: 'POST',

        body: formData

    })

    .then(response => response.text())

    .then(data => {

        document.getElementById('score-container').innerHTML = data;

    })

    .catch(error => console.error(error));

});

});
</script>
</body>
</html>

```

2.LOGIN.HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>ATS Web Application</title>

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/tailwindcss@2.1.4/dist/tailwind.min.css" />

</head>

<body class="bg-gray-200">

  <body>

    <!-- Navigation Bar -->

    <nav class="bg-white shadow-lg">

      <div class="container mx-auto px-6 py-3">

        <div class="flex items-center justify-between">

          <div>

            <a class="text-gray-800 font-bold text-xl" href="index.html">Resume
Checker</a>

          </div>

          <div class="flex items-center">

            <a class="text-gray-800 font-bold mx-3" href="#">Pricing</a>

            <a class="text-gray-800 font-bold mx-3" href="#">Templates</a>

            <a class="text-gray-800 font-bold mx-3" href="#">Blog</a>

          </div>

        </div>

      </div>

    </nav>

  </body>

</body>
```



```

        <a class="text-gray-800 font-bold mx-3" href="signup.html">Sign
up</a>

        <a class="bg-green-500 hover:bg-green-700 text-white font-bold py-2
px-4 rounded-full mx-3"

        href="login.html">Get Started</a>

    </div>

</div>

</div>

</nav>

<br>

<br>

<hr>

<div class="flex justify-center items-center h-screen">

    <div class="w-full max-w-md">

        <form class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

            <div class="mb-4">

                <label class="block text-gray-700 font-bold mb-2" for="username">

                    Username

                </label>

                <input

                    class="shadow appearance-none border rounded w-full py-2 px-3
text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

                    id="username" type="text" placeholder="Username" />

            </div>

            <div class="mb-6">

```

```

<label class="block text-gray-700 font-bold mb-2" for="password">

    Password

</label>

<input

    class="shadow appearance-none border rounded w-full py-2 px-3
text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

    id="password" type="password" placeholder="*****" />

</div>

<div class="flex items-center justify-between">

    <button

        class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-
4 rounded focus:outline-none focus:shadow-outline"

        type="button">

            Sign In

    </button>

</div>

</form>

<form action="/upload" form enctype="multipart/form-data" method="post"

    class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

    <div class="mb-4">

        <input

            class="shadow appearance-none border rounded w-full py-2 px-3
text-gray-700 leading-tight focus:outline-none focus:shadow-outline"

            id="resume" name="resume" type="file" />

        </div>

```

```

<div class="flex items-center justify-between">

    <button id="upload-btn"

        class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-
4 rounded focus:outline-none focus:shadow-outline"

        type="submit">

            Upload

        </button>

    </div>

</form>

<br>

<footer class="bg-gray-200 py-10">

    <div class="container mx-auto px-4">

        <p class="text-xl font-bold text-gray-800 mb-2">Connect With Us</p>

        <div class="flex items-center justify-center my-4">

            <a class="text-gray-800 font-bold mx-3" href="#">Facebook</a>

            <a class="text-gray-800 font-bold mx-3" href="#">Twitter</a>

            <a class="text-gray-800 font-bold mx-3" href="#">Instagram</a>

            <a class="text-gray-800 font-bold mx-3" href="#">LinkedIn</a>

        </div>

        <p class="text-gray-600">Resume Checkmate &copy; 2023</p>

    </div>

</footer>

</div>

</div>

```

```
</body>

</html>
```

3. SIGNUP.HTML

```
<html>

<style>

body {

    background-image: url('img.jpg');

    background-repeat: no-repeat;

    background-attachment: fixed;

    background-size: cover;

}

.kbf{

    align:center;

}

</style>

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>ATS Web Application</title>

    <link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/tailwindcss@2.1.4/dist/tailwind.min.css" />

</head>
```

```

<nav class="bg-white shadow-lg">

  <div class="container mx-auto px-6 py-3">

    <div class="flex items-center justify-between">

      <div>

        <a class="text-gray-800 font-bold text-xl" href="index.html">Resume
Checker</a>

      </div>

      <div class="flex items-center">

        <a class="text-gray-800 font-bold mx-3" href="#">Pricing</a>

        <a class="text-gray-800 font-bold mx-3" href="#">Templates</a>

        <a class="text-gray-800 font-bold mx-3" href="#">Blog</a>

        <a class="text-gray-800 font-bold mx-3" href="signup.html">Sign up</a>

        <a class="bg-green-500 hover:bg-green-700 text-white font-bold py-2 px-4
rounded-full mx-3"

          href="login.html">Get Started</a>

      </div>

    </div>

  </div>

</nav>

```

```

<script>

function a()

{

  alert(f.value);

  document.getElementById("k4").src=f.value;

```

```

}

function valid()

{

var count=0;

if(sname.value=="")

{

document.getElementById("k1").innerHTML="FILL NAME FIRST";

return false;

}

else

document.getElementById("k1").innerHTML="";

n=parseInt(m.value);

for(i=n;i>0;i=parseInt(i/10))

count++;

if(count!=10)

{

document.getElementById("k2").innerHTML="INVALID MOBILE NO";

return false;

}

else

document.getElementById("k2").innerHTML="";

}

</script>

<div id="kbf">

```

```
<form name="kbf" method="Post" onsubmit="return valid()" action="save.php">
```

```
<center>
```

```
<h1 style="color:blanchedalmond">=====Sign up=====</h1>
```

```
<br><input type="text" placeholder="NAME" name="sname" id="sname"/><font  
id="k1" color="red"></font><br/><br>
```

```
<br><input type="text" placeholder="MOBILE NO" name="m" id="m"/><font id="k2"  
color="red"></font><br/><br>
```

```
<br><input type="text" placeholder="EMAIL ID" name="e" id="e"/><font id="k3"  
color="red"></font><br/><br>
```

```
Date of Birth:<br><input type="date" name="dob"/><br/><br>
```

```
<br><input type="Password" placeholder="password" name="pwd"  
id="pwd"/><br/><br>
```

```
<input type="submit" value="SAVE">
```

```
<input type="reset" value="RESET">
```

```
<font color="green">
```

```
<p>NOTE: Use lower and upper case letters with numbers and symbolic characters  
to increase strength of your password!<br>
```

```
<br>
```

```
Already a member?
```

```
<a href="login.html" >Sign In</a><br>
```

```
</center></font><br>
```

```
</form>
```

```
</div>
```

```

<footer class="bg-gray-200 py-10">

  <div class="container mx-auto px-4">

    <p class="text-xl font-bold text-gray-800 mb-2">Connect With Us</p>

    <div class="flex items-center justify-center my-4">

      <a class="text-gray-800 font-bold mx-3" href="#">Facebook</a>

      <a class="text-gray-800 font-bold mx-3" href="#">Twitter</a>

      <a class="text-gray-800 font-bold mx-3" href="#">Instagram</a>

      <a class="text-gray-800 font-bold mx-3" href="#">LinkedIn</a>

    </div>

    <p class="text-gray-600">Resume Checkmate &copy; 2023</p>

  </div>

</footer>

```

4.SCORE.HTML

```

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <meta http-equiv="X-UA-Compatible" content="ie=edge" />

  <title>ATS Score</title>

```



```

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/tailwindcss/dist/tailwind.min.css">

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/gauge.js/1.3.6/gauge.min.js"></script>

</head>

<body class="bg-gray-100 font-sans leading-normal tracking-normal">

    <nav class="bg-white shadow-lg">

        <div class="container mx-auto px-6 py-3">

            <div class="flex items-center justify-between">

                <div>

                    <a class="text-gray-800 font-bold text-xl" href="#">Resume
Checker</a>

                </div>

                <div class="flex items-center">

                    <a class="text-gray-800 font-bold mx-3" href="#">Pricing</a>

                    <a class="text-gray-800 font-bold mx-3" href="#">Templates</a>

                    <a class="text-gray-800 font-bold mx-3" href="#">Blog</a>

                    <a class="text-gray-800 font-bold mx-3" href="#">Sign up</a>

                    <a class="bg-green-500 hover:bg-green-700 text-white font-bold py-2
px-4 rounded-full mx-3"
href="login.html">Get Started</a>

                </div>

            </div>

        </div>

    </div>

```

</nav>

<div class="h-screen flex overflow-hidden bg-gray-100">

<div class="flex-1 flex flex-col overflow-hidden">

<main class="flex-1 overflow-x-hidden overflow-y-auto bg-gray-100">

<div class="container mx-auto px-6 py-8">

<div class="text-gray-800 font-bold text-xl mb-4">

Your ATS Score:

</div>

<div class="text-gray-800 font-bold text-3xl mb-4">

<%= score %>%

</div>

<div class="w-full bg-gray-200 rounded-full">

<div class="h-6 bg-blue-500 rounded-full" style="width: <%= score
%>%;"></div>

</div>

<% if (score < 50) { %>

<div class="text-red-600 font-bold mb-2">

Your resume needs improvement. Please check the highlighted
areas

below.

</div>

<div class="text-gray-600 text-sm mb-2">

Note: Personal information, such as gender, sex, photos, family
details, and date of birth, is not allowed.

</div>

<div class="text-gray-600 text-sm mb-2">

Note: Your resume must be at least 250-300 words and should contain sections on education, experience, and projects.

</div>

<% if (resumeTextErrors.length > 0) { %>

<div class="text-gray-600 text-sm mb-2">

The following issues were found in the text of your resume:

</div>

<ul class="list-disc list-inside">

<% for (const error of resumeTextErrors) { %>

<%= error %>

<% } %>

<% } %>

<div class="text-gray-600 text-sm mb-2">

The following issues were found in the PDF version of your resume:

</div>

<% if (pdfErrors.length > 0) { %>

<ul class="list-disc list-inside">

<% for (const error of pdfErrors) { %>


```

        <%= error %>

    </li>

    <% } %>

</ul>

<% } else { %>

    <div class="text-green-600 font-bold">

        Your resume PDF was parsed successfully.

    </div>

    <% } %>

    <div class="mt-4">

        <a href="/"

            class="bg-blue-500 hover:bg-blue-700 text-white
font-bold py-2 px-4 rounded">Upload

            Another Resume</a>

        </div>

</div>

<% } else { %>

    <div class="text-green-600 font-bold mb-2">

        Congratulations! Your resume is ATS-friendly.

    </div>

    <div class="text-gray-600 text-sm mb-2">

        Your resume has no personal pronouns, contains sections on
education,

        experience, and projects, and is at least 250-300 words.

    </div>

```

```
<div class="text-gray-600 text-sm mb-2">
```

Note: The date format in your resume is in reverse order, which is preferred by most ATS systems.

```
</div>
```

```
<div class="text-gray-600 text-sm mb-2">
```

Note: Your resume has no personal information, such as gender, sex, photos, family details, or date of birth.

```
</div>
```

```
<div class="text-gray-600 text-sm mb-2">
```

Note: Your resume has no diagrams and no grammatical errors.

```
</div>
```

```
<div class="mt-4">
```

```
  <a href="/" class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded">Upload
```

```
    Another Resume</a>
```

```
</div>
```

```
</div>
```

```
</main>
```

```
</div>
```

```
<div id="gauge-container"></div>
```

```
<footer class="bg-gray-200 py-10">
```

```
  <div class="container mx-auto px-4">
```

```
    <p class="text-xl font-bold text-gray-800 mb-2">Connect With Us</p>
```

```
    <div class="flex items-center justify-center my-4">
```

```
      <a class="text-gray-800 font-bold mx-3" href="#">Facebook</a>
```

```

    <a class="text-gray-800 font-bold mx-3" href="#">Twitter</a>

    <a class="text-gray-800 font-bold mx-3" href="#">Instagram</a>

    <a class="text-gray-800 font-bold mx-3" href="#">LinkedIn</a>

</div>

<p class="text-gray-600">Resume Checkmate &copy; 2023</p>

</div>

</footer>

<script>

    // Get the score value from the server

    const score = <%= score %>;

    // Create a new Gauge object

    const gauge = new Gauge({

        renderTo: 'gauge-container',

        value: score,

        min: 0,

        max: 100,

        gaugeWidthScale: 0.6,

        label: 'ATS Score'

    });

    // Set the options for the gauge

    gauge.setOptions({

```

```

angle: 0,

lineWidth: 0.44,

pointer: {

    length: 0.8,

    strokeWidth: 0.035,

    color: '#000000'

},

colorStart: '#6FADCF',

colorStop: '#8FC0DA',

strokeColor: '#E0E0E0',

generateGradient: true,

highDpiSupport: true,

staticLabels: {

    font: '10px sans-serif',

    labels: [0, 20, 40, 60, 80, 100],

    color: '#000000',

    fractionDigits: 0

},

staticZones: [

    { strokeStyle: '#6FADCF', min: 0, max: 40 },

    { strokeStyle: '#8FC0DA', min: 40, max: 60 },

    { strokeStyle: '#6FADCF', min: 60, max: 100 }

]

});

```

```
// Animate the gauge to the score value

gauge.animationSpeed = 32;

gauge.set(score);

</script>

</body>

</html>
```

5.APP.JS

```
const express = require('express');

const app = express();

const multer = require('multer');

const path = require('path');

const resume = require('./resume');

const ResumeParser = require('resume-parser');

const bodyParser = require("body-parser");

const File = require("./model/fileSchema");

app.use(

  bodyParser.urlencoded({

    extended: true,

  })
```



```

);

app.set('view engine', 'ejs');

app.set("views", path.join(__dirname, "views"));

app.use(express.static(path.join(__dirname, 'public')));

app.use(express.urlencoded({ extended: false }));


const upload = multer({

  storage: multer.diskStorage({

    destination: function (req, file, cb) {

      cb(null, 'public/uploads')

    },

    filename: function (req, file, cb) {

      cb(null, Date.now() + '-' + file.originalname)

    }

  })

});


// Route for handling file upload and parsing the resume

app.post('/upload', upload.single('resume'), async (req, res) => {

  try {

    if (!req.file) {

      throw new Error('Please upload a resume file');

    }

  }

```

```

    // Use the resume module to parse the uploaded resume file

    const filePath = path.join(__dirname, 'public/uploads', req.file.filename);

    const { atsScore, errors } = await ResumeParser.parseResume(filePath);

    res.render('views/score', { score: atsScore, errors: errors });

  } catch (error) {

    console.error(error);

    res.status(400).send(error.message);

  }

});

app.get('/views/score', (req, res) => {

  const atsScore = req.query.score;

  const errors = JSON.parse(decodeURIComponent(req.query.errors));

  res.render('score', { score: atsScore, errors: errors });

});

app.use('/resume', resume);

app.use("/", (req, res) => {

  res.status(200).render("index");

});

```

```
// Render the index.html file when the user accesses the root URL

app.get('/', (req, res) => {

  res.render('index');

});

const port = process.env.PORT || 3000;

app.listen(port, () => console.log(`Server started on port ${port}`));

module.exports = app
```

6. RESUME.JS

```
const ResumeParser = require('resume-parser');

const express = require('express');

const router = express.Router();

router.post('/', (req, res) => {

  const file = req.file;

  if (!file) {

    return res.status(400).send('No file uploaded');

  }

  if (file.mimetype !== 'application/pdf') {

    return res.status(400).send('File must be a PDF');

  }

}
```

```

ResumeParser.parseResume(file.path, (data) => {

  const fileText = data.text;

  // Initialize ATS score and criteria flags

  let atsScore = 100;

  let hasPersonalPronouns = false;

  let hasNonReverseDate = false;

  let hasPersonalInfo = false;

  let hasShortLength = false;

  let hasMissingSections = false;

  let hasDiagrams = false;

  if (fileText.match(/(\b\b\bme\b\bmy\b\bwe\b\bour\b)/i)) {

    atsScore -= 10;

    hasPersonalPronouns = true;

  }

  if (fileText.match(/(\d{1,2}\d{1,2}\d{4})/)) {

    atsScore -= 10;

    hasNonReverseDate = true;

  }

  if (fileText.match(/(gender|sex|marital status|photo|family details|date of
birth|dob)/i)) {

    atsScore -= 20;

    hasPersonalInfo = true;

  }

  if (fileText.split(/\s+/).length < 250) {

```

```

    atsScore -= 10;

    hasShortLength = true;
}

if (!fileText.match(/(education|experience|projects)/i)) {

    atsScore -= 20;

    hasMissingSections = true;
}

if (fileText.match(/(fig|table|diagram)/i)) {

    atsScore -= 10;

    hasDiagrams = true;
}

function scoreResume(fileText) {

    let atsScore = 100;

    const regex = /(he|she|his|her|they|them|their)\b/gi;

    if (fileText.match(regex)) {

        atsScore -= 10;

        fileText = highlightMatches(fileText, regex);
    }

    const dateRegex = /(\d{1,2})[\\.\-](\d{1,2})[\\.\-](\d{4}|\d{2})/g;

    if (fileText.match(dateRegex)) {

        atsScore -= 10;

        fileText = highlightMatches(fileText, dateRegex);
    }
}

```

```

    }

    const personalInfoRegex = /(sex|gender|marital status|photo|picture|family
details|date of birth)\b/gi;

    if (fileText.match(personalInfoRegex)) {

        atsScore -= 10;

        fileText = highlightMatches(fileText, personalInfoRegex);
    }

    if (fileText.split(/\s+/).length < 250) {

        atsScore -= 10;
    }

    const sectionsRegex = /(education|experience|projects)\b/gi;

    if (!fileText.match(sectionsRegex)) {

        atsScore -= 10;

        fileText = highlightMatches(fileText, /. /g);
    }

    if (fileText.match(/fig|table|diagram/i)) {

        atsScore -= 10;

        fileText = highlightMatches(fileText, /fig|table|diagram/gi);
    }

    if (fileText.match(/[\^x00-\x7F]/g)) {

        atsScore -= 10;

        fileText = highlightMatches(fileText, /[\^x00-\x7F]/g);
    }

    return {

        atsScore,

```

```

        highlightedResume: fileText
    };
}

function highlightMatches(text, regex) {
    const replacement = '<span class="text-red-500">$&</span>';
    return text.replace(regex, replacement);
}

router.post('/upload', upload.single('resume'), async (req, res) => {
    try {
        const fileText = await extractText(req.file.path);

        const { atsScore, highlightedResume } = scoreResume(fileText);

        res.render('score', { atsScore, highlightedResume });
    } catch (error) {
        console.error(error);

        res.render('error');
    }
});

res.render('score', {
    atsScore,
    hasPersonalPronouns,
    hasNonReverseDate,
    hasPersonalInfo,
    hasShortLength,
    hasMissingSections,

```

```
        hasDiagrams
    });

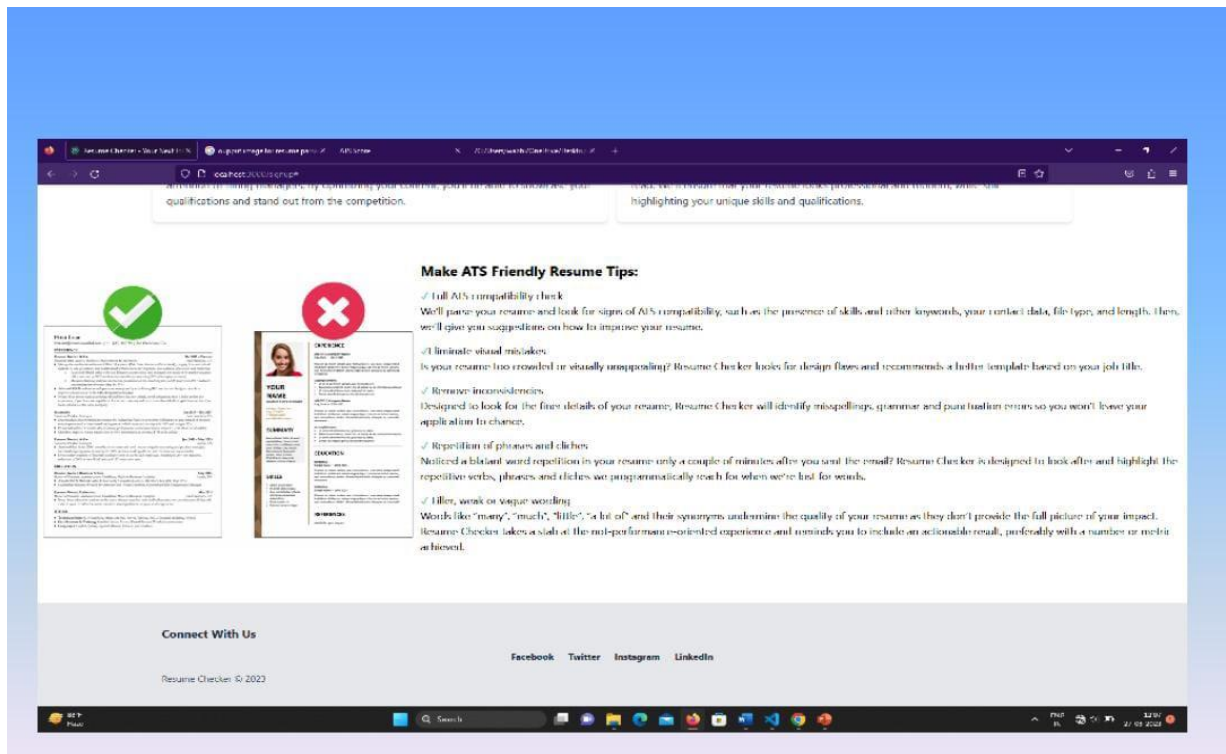
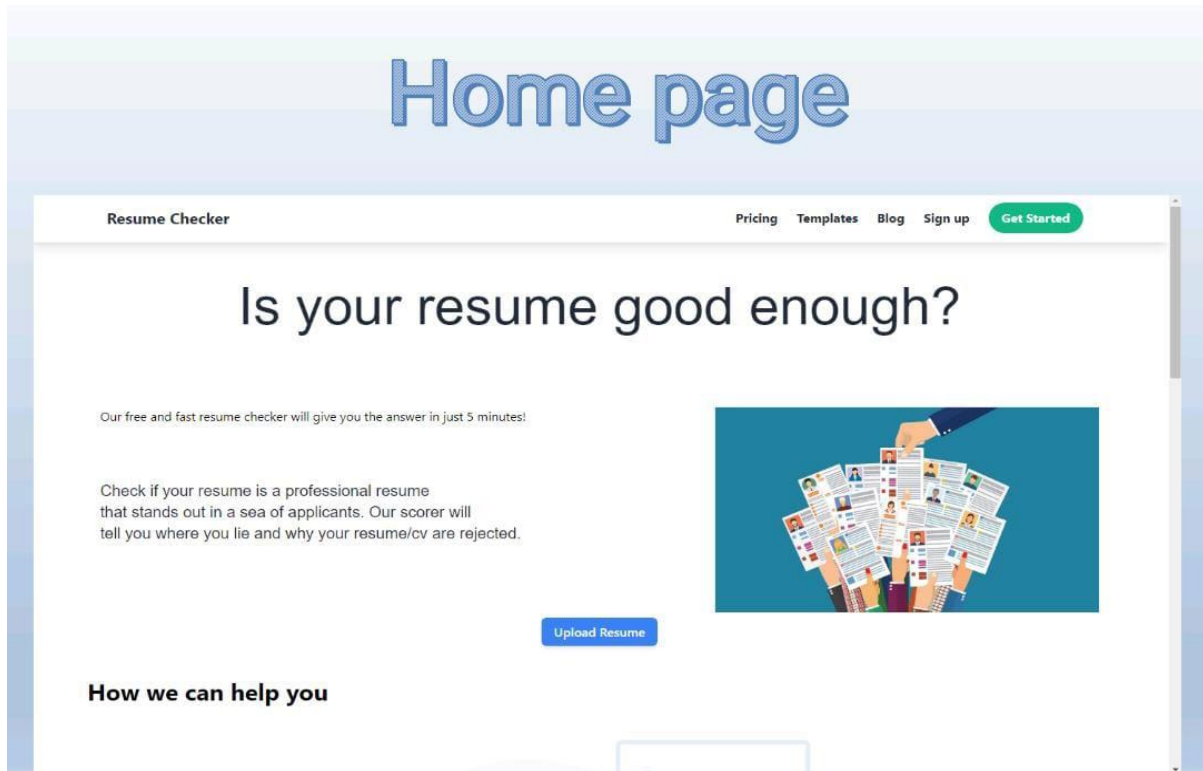
});

})

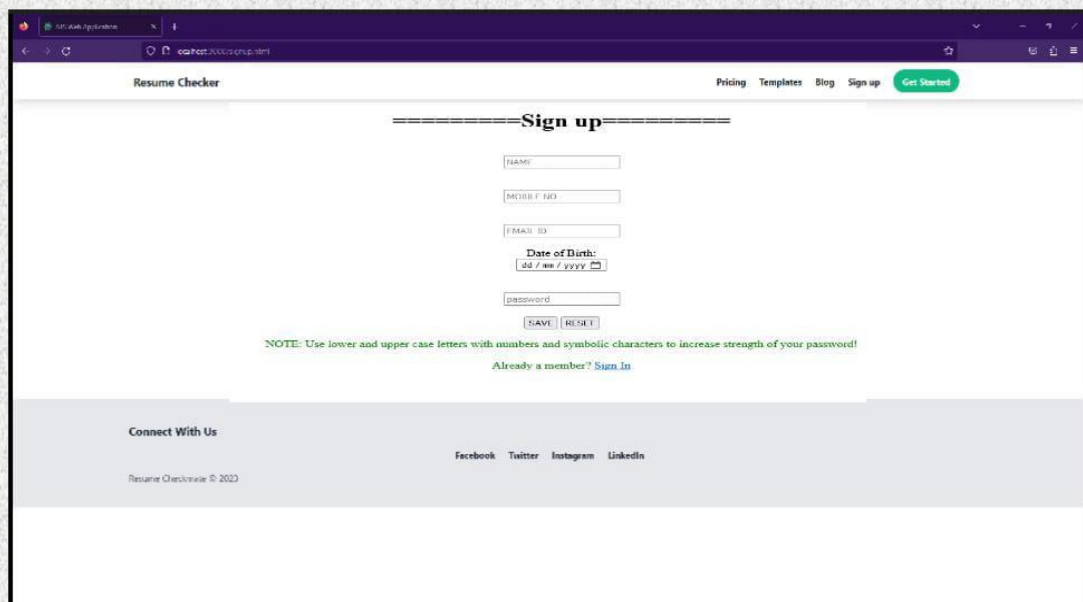
module.exports = router;
```


OUTPUT SCREEN

Home page



SIGN UP PAGE



The screenshot shows the 'Sign up' page of the 'Resume Checker' application. The page has a dark purple header with the site name 'Resume Checker' on the left and navigation links 'Pricing', 'Templates', 'Blog', 'Sign up', and a green 'Get Started' button on the right. The main content area is white and features a 'Sign up' title with a decorative underline. Below the title are input fields for 'NAME', 'MOBILE NO', 'EMAIL ID', 'Date of Birth' (with a date picker), and 'password'. There are 'SAVE' and 'RESET' buttons below the password field. A green note states: 'NOTE: Use lower and upper case letters with numbers and symbolic characters to increase strength of your password!'. Below the note is a link 'Already a member? Sign In'. At the bottom, there is a 'Connect With Us' section with links for Facebook, Twitter, Instagram, and LinkedIn, and a footer 'Resume Checker © 2023'.

Resume Checker

Pricing Templates Blog Sign up Get Started

Sign up

NAME

MOBILE NO

EMAIL ID

Date of Birth:

password

SAVE RESET

NOTE: Use lower and upper case letters with numbers and symbolic characters to increase strength of your password!

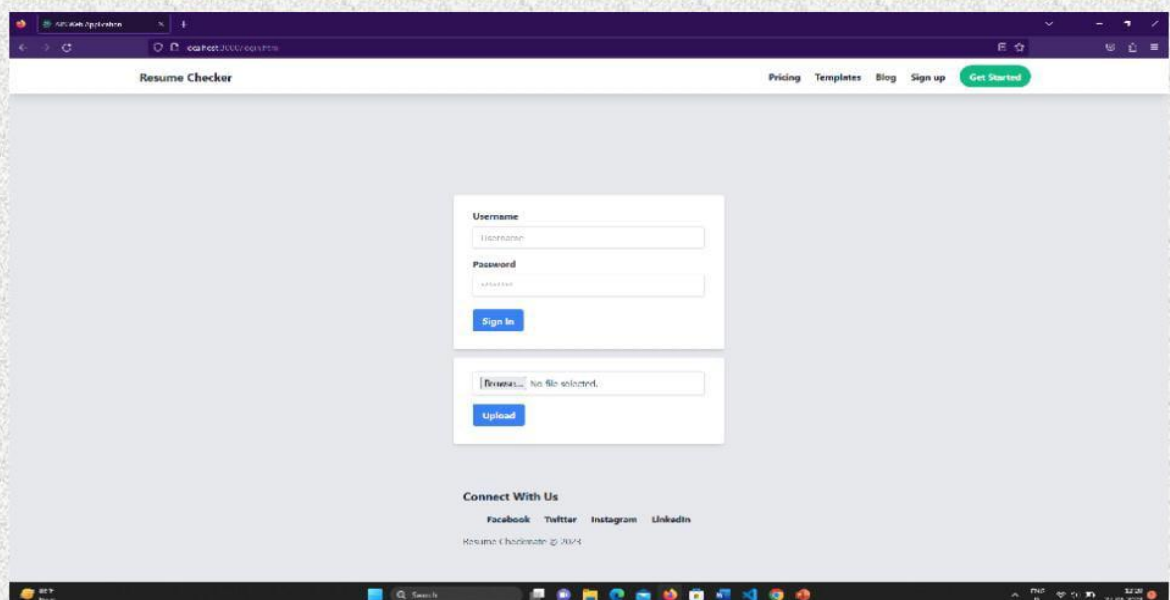
Already a member? [Sign In](#)

Connect With Us

Facebook Twitter Instagram LinkedIn

Resume Checker © 2023

LOG IN PAGE



The screenshot shows the 'Log In' page of the 'Resume Checker' application. The page has a dark purple header with the site name 'Resume Checker' on the left and navigation links 'Pricing', 'Templates', 'Blog', 'Sign up', and a green 'Get Started' button on the right. The main content area is light gray and features a 'Log In' title with a decorative underline. Below the title are input fields for 'Username' and 'Password', followed by a blue 'Sign In' button. Below the 'Sign In' button is a file upload section with a text input field containing 'Browse...' and 'no file selected.', and a blue 'Upload' button. At the bottom, there is a 'Connect With Us' section with links for Facebook, Twitter, Instagram, and LinkedIn, and a footer 'Resume Checker © 2023'.

Resume Checker

Pricing Templates Blog Sign up Get Started

Log In

Username

Password

Sign In

Browse... no file selected.

Upload

Connect With Us

Facebook Twitter Instagram LinkedIn

Resume Checker © 2023

Resume Analysis

Hello ROBERT SMITH

Your Basic info

Name: ROBERT SMITH

Email: info@wikresume.com

You are at intermediate level!

Skills Recommendation 💡

Skills that you have

Flask ✕ Opencv ✕ Tkinter ✕ Css ✕ Video ✕ System ✕
Information technology ✕ Database ✕ Writing ✕ R ✕ Nosql ✕ Openstack ✕
Teaching ✕ Real-time ✕ Html ✕ Machine learning ✕ Communication ✕
Tensorflow ✕ Aws ✕ Sql ✕ Documentation ✕ Analysis ✕ Algorithms ✕
Android ✕ Pandas ✕ English ✕ Matplotlib ✕ Python ✕ Windows ✕
Presentation ✕ Cloud ✕ Numpy ✕ Keras ✕ Linux ✕ Github ✕

[See our skills recommendation](#)

Our analysis says you are looking for Data Science Jobs.

Resume Tips & Ideas 💡

[-] According to our recommendation please add your career objective, it will give your career intension to the Recruiters.

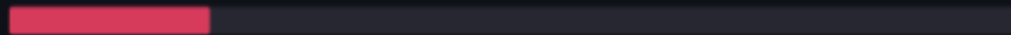
[-] According to our recommendation please add Declaration 📜. It will give the assurance that everything written on your resume is true and fully acknowledged by you

[+] Awesome! You have added your Hobbies 🎮

[-] According to our recommendation please add Achievements 🏆. It will show that you are capable for the required position.

[-] According to our recommendation please add Projects 🏠. It will show that you have done work related the required position or not.

Resume Score 📄



Your Resume Writing Score: 20

Note: This score is calculated based on the content that you have added in your Resume.

LIMITATIONS

Here are some limitations of this project:

1. **Language limitations:** The project is designed to analyze resumes that are in PDF format and in English language. It may not work accurately for resumes in other languages.
2. **ATS limitations:** The ATS scoring algorithm used in this project is a simple one and it may not be able to accurately evaluate the candidate's skills, experiences, and achievements. It may not be able to properly rank the resumes.
3. **Formatting limitations:** The project expects the resumes to be in a certain format, and it may not be able to parse the resumes that do not conform to the expected format.
4. **Dependency limitations:** The project uses third-party dependencies such as `resume-parser` and `express`. Any issues with these dependencies may cause issues with the project.
5. **User experience limitations:** The project does not provide a user-friendly interface for the users to upload their resumes and view the ATS scores. Users need to use API endpoints to interact with the project, which may not be convenient for some users.
6. **Security limitations:** The project does not have any security measures implemented to protect the uploaded resumes or user data. This could pose a security risk if the project is deployed on a public-facing server.

7. **Limited to PDF format:** This project is designed to parse only PDF format files, which limits the scope of its usage. It cannot parse other file formats like Microsoft Word, HTML, or plain text files.
8. **Accuracy of ATS score:** The accuracy of the ATS (Applicant Tracking System) score is dependent on the criteria used in the scoring algorithm. The current algorithm is based on some assumptions and rules that may not be accurate for all job descriptions or industries.
9. **Limited personal information detection:** The project only checks for some specific personal information like gender, marital status, and date of birth. It may not be able to detect other types of personal information like social security numbers, contact details, or addresses.
10. **Limited keyword matching:** The keyword matching used in the project is based on simple regular expressions. It may not be able to detect some variations of keywords.
11. **Limited scalability:** The project may not be scalable for large volumes of resumes. As the number of resumes increases, the processing time may also increase, which could affect the performance of the system.
12. **Dependence on machine learning models:** The project heavily relies on pre-trained machine learning models for natural language processing, which may not always provide accurate results. These models require large amounts of training data and may not be suitable for all types of language-related tasks.
13. **Sensitivity to input quality:** The quality of input data can have a significant impact on the accuracy of the output. Any noise or errors in the input data can result in incorrect or unreliable outputs.
14. **Limited scope of functionality:** The project is limited to a few specific language-related tasks, such as sentiment analysis, text classification, and text generation. It may not be suitable for more complex tasks, such as language translation or speech recognition.

15. **Resource-intensive:** The project requires significant computing resources, including powerful CPUs and GPUs, as well as large amounts of memory and storage. This can make it difficult to deploy and run on lower-end hardware or in resource-constrained environments.

16. **Privacy concerns:** The project relies on the processing of user data, which could raise privacy concerns. The storage and processing of user data must be handled carefully to ensure that it complies with relevant privacy regulations and best practices.

FUTURE SCOPE

The future scope of a project refers to the potential possibilities for growth and development beyond the current state of the project. It involves identifying opportunities to enhance the project's features, functionality, and impact to better serve the needs of users or stakeholders.

The future scope of a project is often considered during the planning and design phases, as it can have an impact on the project's overall direction and goals. It can involve a variety of factors, including new features, additional functionality, expanded user base, increased scalability, and improved performance.

1. **Multi-user support:** This project can be extended to support multiple users with different levels of access rights. Users can be categorized based on their roles and permissions can be assigned accordingly.
2. **Real-time collaboration:** Adding real-time collaboration feature can enhance the project by allowing multiple users to edit the same document simultaneously. This feature will be beneficial for teams that are working remotely or from different locations.
3. **Integration with cloud storage:** Integrating the project with cloud storage services such as Dropbox or Google Drive can make it more convenient for users to save and access their files from anywhere.
4. **Advanced document formatting options:** The project can be enhanced by adding more advanced document formatting options such as headers, footers, tables, page breaks, and other formatting options.
5. **Mobile application:** Developing a mobile application for the project can make it more accessible and convenient for users who prefer to work on their mobile devices.
6. **Customizable templates:** Adding customizable templates for common document types such as invoices, contracts, and reports can save users time and effort in formatting and creating these documents from scratch.
7. **Version control:** Implementing version control can allow users to track changes made to a document over time, restore previous versions of a document, and collaborate on a document without the risk of losing any important information.

8. **Text-to-speech functionality:** Adding text-to-speech functionality can make the project useful for users who prefer to listen to a document rather than read it. This feature can be especially useful for users with visual impairments or reading difficulties.
9. **Support for more file formats:** Currently, the project only supports parsing and scoring PDF resumes. However, many job seekers may also submit resumes in other file formats, such as Microsoft Word or plain text. Supporting these file formats can increase the usefulness and flexibility of the tool.
10. **Improved natural language processing:** While the tool currently uses basic regex patterns to identify potential issues with resumes, more advanced natural language processing (NLP) techniques could be implemented. For example, the tool could use named entity recognition to identify and flag personal information, or use machine learning algorithms to identify the most important sections of a resume.
11. **Customizable scoring criteria:** The current tool uses a fixed set of criteria to score resumes, but different employers and hiring managers may have their own specific criteria and priorities. Allowing users to customize the scoring criteria and weights can make the tool more relevant and useful to a wider range of users.
12. **Integration with applicant tracking systems (ATS):** Many larger companies use ATS software to manage and filter resumes submitted through online job portals. Integrating with these systems can make it easier for job seekers to submit resumes that meet the specific requirements of each employer, and increase the chances of their resumes being selected for review.
13. **Support for multiple languages:** The current tool is designed for resumes written in English, but many job seekers may be non-native English speakers or may submit resumes in other languages. Supporting multiple languages can expand the reach and accessibility of the tool.
Improved accuracy: The ATS scoring system could be improved by adding more criteria and algorithms, such as natural language processing and machine learning. This could result in more accurate and reliable scores.
14. **Integration with job portals:** The project could be integrated with popular job portals such as Indeed, Monster, and LinkedIn. This would enable job seekers

to upload their resumes directly from these platforms and receive scores and feedback.

15. **Customization:** Employers could use this project to create their own customized ATS algorithms that are tailored to their specific needs and requirements.
16. **Resume builder:** The project could be expanded to include a resume builder that helps job seekers create resumes that are optimized for ATS. This would include recommendations on formatting, content, and keyword usage.
17. **Multilingual support:** The project could be expanded to include multilingual support, allowing job seekers to upload resumes in languages other than English.
18. **Mobile app:** A mobile app version of the project could be developed, allowing users to upload resumes and receive scores on the go.
19. **Integration with HR software:** The project could be integrated with popular HR software platforms such as SAP SuccessFactors, Workday, and Oracle HCM Cloud. This would enable employers to use the ATS scoring system as part of their overall hiring process.

REFERENCES

Websites:-

1.<https://www.geeksforgeeks.org>

2.<https://www.javatpoint.com>

3.[codewithHarry](#)

4.[GeeksforGeeks](#)

5.[Tutorialspoint](#)

6.[smartDraw](#)