

**Rezume: A Web-based resume builder system for university
students**

by

Ayush Arora

**Project submitted in partial fulfillment of the requirements for the degree of
Master of Science in Information Sciences and Technology**

Rochester Institute of Technology

B. Thomas Golisano College

of

Computing and Information Sciences

Department of Information Sciences and Technologies

5th December 2021

Rochester Institute of Technology
B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in Information Sciences and Technology

Project Approval Form

Student Name: Ayush Arora

Project Title: Rezume - A Web-based resume builder system for university

 Students

Project Committee

Name

Signature

Date

.....
 Chair

.....
 Committee Member

.....
 Committee Member

Dedication

To my dearest uncle Mr. Darshan Lal Arora

Acknowledgments

This project would not have been possible without the unconditional support of many, and I would like to take this opportunity to give my thanks to the ones who made it possible. First, I would like to express my sincere thanks to Prof. Dan Bogaard for his valuable support and guidance throughout. His invaluable expertise in the domains of the project and methodologies helped me push myself into a highly creative zone. His valuable suggestions on how to make the functionalities for this project helped me move in the right direction and complete my work. I would also like to thank Prof. John Paul Takats for serving on my capstone committee and taking out the time to give me continual feedback on the project over the semester. His persistence helped me remain consistent and aim to achieve what I thought would be impossible in a short duration of time. Furthermore, this work would not have been possible without the love and understanding of my friends and family. Their continuous support made me want to be a better version of myself and create something valuable every day. This journey would not have been possible without them.

Table of contents

1. Abstract	01
2. Introduction	02
2.1 Background	02
2.2 Motivation	03
2.3 Contribution	04
3. Literature Review	05
3.1 Existing Systems	05
3.2 ATS Friendly Resumes	05
3.3 Understanding Features	06
4. Methodology	07
4.1 Architecture	08
3.2 Tools & Technologies	09
5. Results	17
6. Future Work	31
7. Conclusions	33
8. References	35

Table of figures

1. MVC Architecture.	09
2. MERN Stack Architecture.	09
3. Working of React Redux	12
4. Resume Template in JSON	14
5. REST API Model	16
6. Landing Page.	17
7. Register & Login Form	17
8. User Dashboard	18
9. Setup Profile.	19
10. View Profile Components.	19
11. Change Font Functionality.	20
12. Change Font Size Functionality.	20
13. Change Color Functionality	20
14. Template Selector.	21
15. Visibility Component.	21
16. More Visibility Component.	22
17. Order Changer	22
18. Resume Template 1	23
19. Resume Template 2	24

	6
20. Resume Template 3	25
21. Download & Load Data Functionality.	26
22. Paper Size Functionality	26
23. Local Storage & Sign Out Functionality	27
24. Functionality State Chart in Redux	27
25. Admin Dashboard	28
26. Custom Section Schema	28
27. Admin Form Modal	29
28. Toggle Admin	29
29. Form Modal States	30

1. Abstract

Project - **Rezume: A Web-based resume builder system for university students**

Student's Name - **Ayush Arora**

Supervising Professor: **Prof. Dan Bogaard**

A resume is created by a job applicant to showcase their qualifications for a position. It contains the summary of the applicant's experience and education making them a good fit in the company for the job opening. For new graduates, listing these qualifications becomes a strenuous job as the amount of practical knowledge they hold is limited. Research suggests that about 75% of the resumes don't ever make the threshold of the company and these numbers are even higher for fresh graduates[5]. Other common issues with resume writing include the organization of the content, having the correct margins and spacing, knowing what sections to be included or excluded, etc. Creating and formatting different resumes for different job roles is a strenuous job. The proposed website aims to help the students draft and rework their resumes based on a simple profile-based setup which will help them save invaluable time and provide them with standard and well-designed resumes to help make a compelling impact. The website supports the concept of JSON Resume, an open-source initiative to create JSON-based standards for resumes, and uses the schema of data in JSON format to create a minimalistic resume for the user[13]. This further helps to create generalized resume checkers that can easily and efficiently extract data from the resumes.

2. Introduction

2.1 Background

Resumes are written with the intent to emphasize the candidate's strengths and to show the employer how they would be a good match for a job opening. Employers generally look for the candidate's job history and their advancements in the work environment. A candidate can get a job interview and make a good first impression solely based on an impressively designed resume. There are a lot of agencies as well as websites that help students and other candidates draft resumes for a significant price. The websites focused on providing templates for the resumes leave the hard work of designing the resumes for the candidates. The candidates end up spending hours updating and personalizing the resumes as per the job requirements instead of focusing on preparing themselves for the job.

Larger companies use Applicant Tracking Systems (ATS) that screen a large number of applications and filter out the best candidates for the human recruiters to view[5]. This means when a student applies for a job their resumes don't reach the recruiter at all. Hence, designing an ATS-friendly resume is necessary, but this is seldom known to the candidates. The formatting of the resume plays an important role within the ATS systems as these softwares follow a standard set of rules to extract data from a resume.

On other hand, a lot of students lack the practical knowledge in the process of designing a resume. This with the constant pressure to portray your set of required skills in the best possible way in a single piece of paper is a difficult process.

2.2 Motivation

To solve the above-mentioned problem and to enable the students to easily generate resumes using an automated process is a necessity. Students generally learn to format their resumes using word processors or take on learning LaTeX to code their resumes. This manual method of designing resumes requires brute force methodology and the students are generally left with unembellished resumes. There are multiple websites on the internet that provide similar functionalities but do always come at a price. Each resume costs about \$100 on such portals which is something a student would rarely spend for designing resumes.

JSON Resumes is an initiative by a group of developers to bring standardization to the resume-making process. It uses a predefined format of the resume in JSON. JSON is a lightweight data interchangeable format that is easy to use, read and write by humans. Furthermore, JSON is built up of attribute-value pairs and arrays making them super easy to store and transmit in human-readable formats. Every developer working in the field are familiar with JSON formatted data and hence using it with an easy to use user interface for everyone helps achieve the goal of standardizing the CV fields to improve compatibility between tools like ATS checkers or resume parsers, ease of conversion, and above all ease of formatting the data into pre-designed formats. JSON when compared to the XML format is briefly encoded and thus JSON parsers require less space-time complexity. JSON also is built for faster transmission of data whereas XML does a lot more.

Intending to further spread the amazing initiative of JSON Resumes to non-developers who aren't familiar with JSON, a profile-based approach is preeminent. The user would fill a set of forms to create a profile and ultimately generate standardized resumes with a few clicks. This enables them to generate resumes compatible with ATS checkers using resume templates available with minimalist design, color, and logical layouts. This further makes it easier for students to recreate their resumes without having to format them repeatedly.

2.3 Contribution

The proposed project focuses on achieving a blend of all the above features with a focus on a user-friendly interface, a minimalistic design, and responsiveness. It would host a number of resume templates custom curated for new graduates. Each student will be able to build a free profile that holds data replicating their Social Media profiles. The stored data could be edited on the go and create quick and personalized resumes for the students efficiently. The generator allows the user to play around with the visibility and the order of the sections and subsections of the resume to make a one-page resume.

The motive is to use the user-created profiles and store that data into a standard Resume format in JSON and generate resumes for the users within minutes. The profile will be editable and will be converted into a resume based on the templates selected by the user within seconds. This means the users will save a lot of precious time spent on aligning their texts, handling spacing and margins within MS Word, or even writing code on LaTeX.

A user will be able to add features like personal information including links to their LinkedIn profiles or online portfolios, a summary of their qualifications and skills, their education, the experience they have in the respective field. The provided resume templates will follow several rules like the four-quadrant rule, which will use columns to represent information, and other design and typographic rules. Users would also have an option to select from a variety of colors and a set of standard fonts to portray their true selves via the resume.

The website aims at providing a professional-looking resume with minimal effort. To accomplish this further the user can also use data saved in JSON files from their previous resumes or even download the JSON format of the resume data to use them in other templates later on. This gives them the flexibility to generate resumes and customize them according to their specific needs.

3. LITERATURE REVIEW

3.1 Studying the existing systems

The resume builder systems on the web were explored and their best features have been added to the application. The biggest problem in most of these systems is that they endure a high cost every time a user builds a resume. The services do come with expert advice to improve the resume but do cost about hundreds of dollars. There are a few systems that are free of cost like the one I. Wu [1] describes the Pro Resume that focuses only on infographic resumes. Also, the website has only a single template but gives some design flexibility to the user. Similarly, Ingale [2] in her proposal of the Resume Portal focuses on an integrated system where the user can apply for job vacancies posted on the portal. This portal covers a lot of bases like building resumes and getting jobs with a lack of research and focusing on vastly used Applicant Tracking Systems that companies already use. It also assumes the companies will be posting their job requirements on the portal which makes the process of generating resumes coupled to the requirements. The one thing that can be useful is creating links to user resumes that can be shared by the user instead of uploading a PDF file everywhere. Other systems propose resume generators with multilingual capabilities or the use of AR markers to visualize various parts of printed resumes on the web.

3.2 Research on “ How to make ATS Friendly Resumes”

The Ladders eye-tracking study suggested that a recruiter spends about 7.4 seconds on a resume hence it is very important to focus on having a clean-looking resume, having short declarative statements, and most importantly having the standard headings and subheading without any extraordinary design experiments[10]. Thus, the details presented by Risavy in the Resume Research Literature[3] serve a very important role in the development of every template for the project. The valuable tips shared by Clift in the “How to beat ATS” [4] also list a checklist that should be followed while making resumes.

The design tips shared on the [7] RIT’s Career Services and Co-op and Purdue’s Resume Design [8] article also provide important insight on the checklist created previously.

3.3 Exploration of features to improve the user experience

The overall functioning of an ATS is studied and the application proposed by Tiwari and et al in “Applicant Tracking System”[5] focuses on Natural Language Processing to only detect job-related words. Using the knowledge we have about the design of resume templates, a rubric of an improved system can be drafted. This would be done by including the factors listed and a higher preference would be given to the spellings and grammar of the Resume.

4. METHODOLOGY

The main idea behind the project is for the user to have a customizable resume-making experience with the ability to create, modify and delete the sections and subsections of the resume from the home screen itself. The JSON Resume standard is used to comply with a data format that can be used in the application by the users to create their resumes in a regulated manner. This format is discussed in detail in the upcoming sections. Further to make the visual aspect of the resume a number of functionalities are added so that users can generate the best possible resume for their needs. This is done by adding functionalities like changing the primary color in the template, updating the font family of the resume, size of the resume among A4, letter, and legal. Further, the user can either download the resume in the pdf format and also download the data in the JSON format that can be used elsewhere.

A user will have to register to a secure portal to access all these features and then log in every time they want to use it. The user can either be a general user or an admin that will have more functionalities to add sections, new colors, and font faces to the application. The authentication system for the website is built on the principles of JSON web tokens and a role field within the user defines if they are an admin or not. The user authentication is handled in a highly secure way as the user data being stored on the cloud is sensitive.

The user interface of the website is kept minimalistic with essential components combined together in an esthetic manner to reflect professionalism. The screen shows the resume the user is currently working on and all the components are stored within the slider menu. This reduces the number of unnecessary user interactions making the website easy to use.

4.1 Architecture

The application follows the Model View Controller architecture paradigm to promote organized programming; the functionality, logic, and user interface are all built separately to handle specific development aspects of the application.

Model - The model is the store of the application and handles the data and data-related logic of the application. This project uses a NoSQL database MongoDBs cloud implementation. The model is connected directly to the database so all the modifications to the data are done in the model component. The controller communicates with the model and requests the data and this request is further executed by the model to either retrieve, modify or add data from the database.

View - The view is what the user actually sees and how they interact with the application. The user can interact with the data and other business logic via the view. The view is built up using components using react and the dynamic data from the controller. The data actually comes from the Model but the controller acts as an interface between the view and the model. The user performs actions using buttons or other interactions on the view.

Controller - The controller is the main man that enables the interaction between the model and the view. At the backend, the controller just tells the model what needs to be done and then the data logic is handled by the model. Further the controller processes all the data received and instructs the view on how to represent the data. The controller also processes all the user inputs and interactions and then decides and informs both the view and the model what happens next. The controllers in the application are Express.js. Redux is also used as store management within JavaScript that acts as the data store for the UI layer.

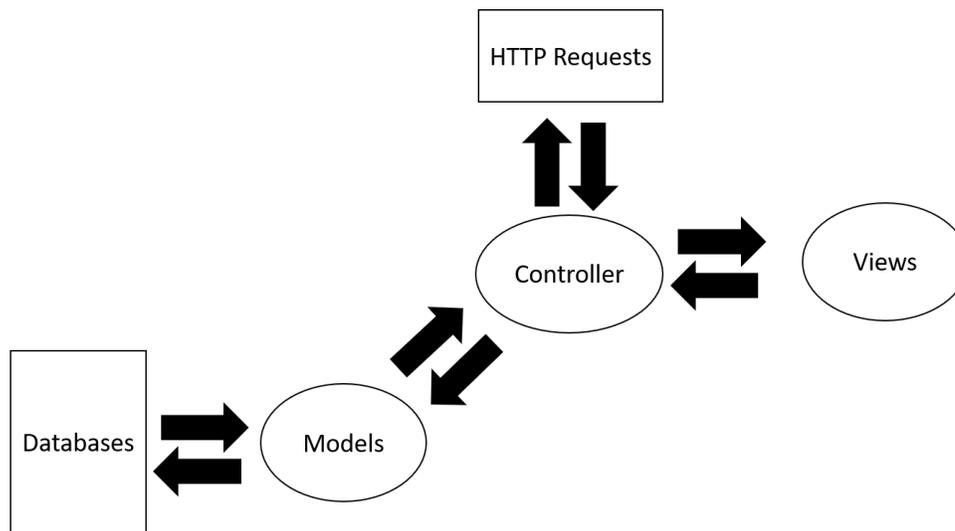


Figure 1: MVC Architecture

4.2 Tools and Technologies used

The project is built on the MERN stack which is used for faster and easier deployment of full-stack applications.

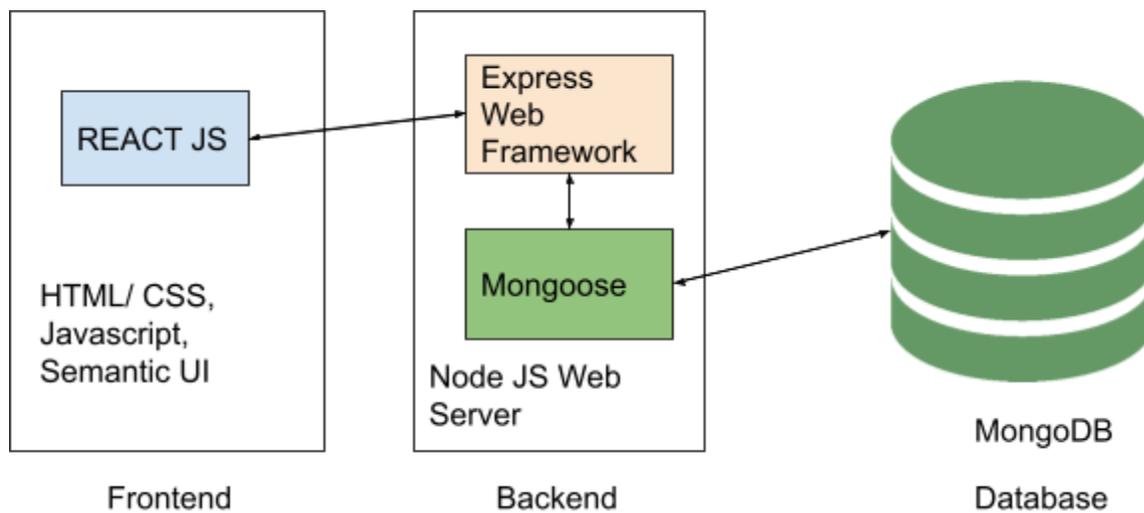


Figure 2: MERN Stack Architecture

MONGO DB - MongoDB is a NoSQL database management system that leverages a JSON-style storage format BSON. BSON is also known as Binary JSON is an extension of JSON that encodes type and length information thus making its parsing faster. This further allows MongoDB to store binary data and dates and to efficiently index, map, and nest data in

support of complex query operations and expressions. The biggest advantage of using NoSQL databases is that the structure of data stored isn't enforced hence it reduces the application development process and reduces the complexity of the deployments.

MongoDB is integrated into a Node.js application using a library called Mongoose. Mongoose is a data modeling library that manages relationships between data and previous schema validation. Mongoose helps in modeling the JSON objects to the representation of the objects in MongoDB. The major reason for choosing Mongoose over native MongoDB for this project was because Mongoose provides an abstraction layer on the top eliminating the use of named collections. The default values for document properties and data validation are also handled by models in mongoose. Further functions can be attached to the Mongoose models that chain together to form queries that are more flexible and easy to use. MongoDB cloud database lets development and production of websites without creating database servers and allow remote access to the database. Mongoose makes it possible for the declared schemas to directly read the data from the database instead of having to write code to do that separately.

EXPRESS - Express is a lightweight Node.js framework that helps in writing the back-end code for a web application. It helps users write robust APIs and web servers in a simple, flexible, minimalist, and scalable manner. A middleware is a piece of software that provides an integration between the application and a low-level API. Express is based on a Node.js middleware module Connect that extends the built-in server functionality. Express also supports other series of middlewares that make coding shorter and easier when compared to core Node.js. One of the most important reasons for selecting Express for this project is the reusability of the code to perform essential server functions like parsing of the payload, cookies, storage session, and selection of the route pattern. Further Express is efficient and fast with a great community of web developers willing to help each other. Express is also asynchronous and hence doesn't affect the functionality of other components within a website.

REACT - A JavaScript framework is a set of JavaScript code segments to perform routine programming tasks. React is a widely used JS framework used to build user interfaces as a tree of small pieces called components. Components can be simplified as independent and reusable segments of code that are written in JSX or JavaScript XML. JSX is a syntax extension that comes with the full power of JS and allows writing HTML within React.

The reason for selecting React for this project is that it is really easy to make interactive User Interfaces as the components are developed for different states and then React renders and updates the right components when the data changes. The virtual dom model created by React is very flexible and the user can code the whole pages on rerenders when a change occurs but in reality, only renders the components that change making it really fast. Finally React has a great set of supporting libraries that it supports making it possible to complete complex tasks easily.

Node.js - It is the most widely used run-time environment for JavaScript that is inclusive of everything that is needed for the execution of a program. Node.js was created so as to run JavaScript code outside from a browser into a standalone application. This extension made it possible to build websites with high functionalities like real-time chat applications, video streaming platforms, collaboration tools, the internet of things, and even machine learning applications instead of just plain interactive websites. Node.js runs on the V8 JS runtime engine that converts the written code into low-level machine code that is then run by the computer.

It is used to develop server-side applications based on an event-driven, non-blocking Input/Output model. Input and output can either be generated by the user interaction with the website, reading, writing from a local machine or generated from the HTTP requests from an API. An event-driven non-blocking I/O means that every execution in Node.js is triggered by an event request that does not wait for the previous I/O operations to finish making it possible to run multiple requests in parallel. This results in the elimination of the need for multi-threading making Node.js lightweight and efficient.

The application built is highly dependent on the transaction of data to build a fast, scalable application and not on any computation on the data. Node.js isn't very efficient to build applications with high computational complexity thus making it a good choice for the development of this application.

REDUX

Redux is a lightweight JavaScript library that is used to centralize and manage the application state and can be used easily with any JavaScript framework. It is used to create processes and procedures to communicate with the store so that the components can read and update the store as and when required. The state is temporary storage that can be accessed by all the components in a structured way through the actions and the reducers.

React allows unidirectional data flow so it becomes difficult to maintain all the states at one place within the application and to transfer it when and where needed. Thus the sharing of data across components is easily enabled by using state management that creates a physical data structure that stores the state of the application that the components can access. Thus Redux becomes a very important aspect for this application as a lot of data coming from the API is stored here. Additionally, multiple functionalities like visibility, the font face, order of the sections, etc are stored in the store as states.

The actions on the current state of the application are performed by pure functions known as reducers returning the new state. An action is an object containing the payload of information that acts as the only source of information for the Redux store to be updated.

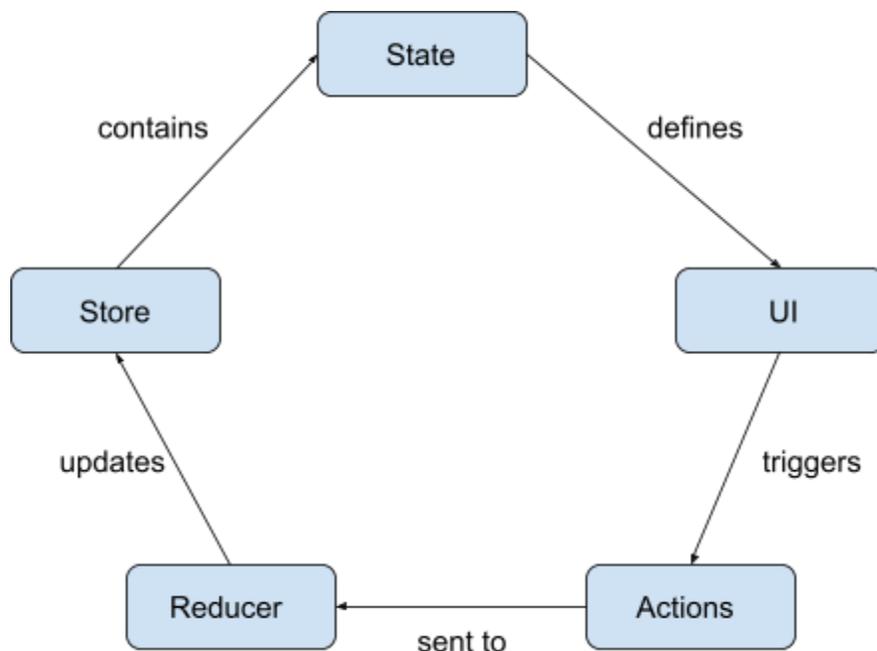


Figure 3: Working of React Redux

JSON - JavaScript Object Notation abbreviated as JSON is a lightweight schema-less, text-based representation of structured data. It supports the data-interchange format and is based on key-value pairs and ordered lists. It is parsed and generated by the machines easily and is supported in most programming languages either natively or by using libraries. JSON is a data-interchange format as it is a language-independent text format and two data structures

namely a key-value pair and lists are supported by every modern programming language. Thus JSON is commonly used to exchange data within web applications as with the advancements in JavaScript and JSON web developers were able to create websites that loaded content on a reload in the background and dynamically load it on the browser. This overcame the problem of reloading the whole page when any component was being loaded or updated in the traditional web applications making them faster and efficient. Other reasons for selecting JSON for this project were its fast transactions and parsing on the server-side, its cross-browser compatibility, its ability to store and share data of any size and type including multimedia.

Passport.js - The security of an application is one of the most important aspects of its successful implementation. The user data stored must be secured and only the user himself must be able to access the data. This can be achieved using authentication which consists of a series of tasks for its implementation. Authentication within this application is achieved using Passport.js middleware within Node.js. Passport helps in keeping the code modular and encapsulated by delegating all the functionalities to the application and taking complete responsibility for the authentication requests. Further passport has different authentication mechanisms packaged as individual strategies which help use them according to the requirements of the application without creating unnecessary dependencies. The ideal choice for this application is the passport strategy that uses JSON Web Tokens to implement Restful endpoints without the use of sessions.

JWT - JSON Web Tokens is a standard focused on the transmission of data in a verified and trusted manner. This uses digitally signed and verified JSON objects. JWTs are very commonly used in authentication because of their integrity and compactness. Once authenticated each API request is attached with a JWT providing user access based on the token. A JWT consists of three parts, the header that stores the type of the token and the algorithm used to sign it, the payload that stores the claims that store the information about the user, and user access, and finally the signature itself all separated by dots and encrypted using a secret key.

```
1 const Resume = {
2   profile: {
3     name: '',
4     email: '',
5     phone: '',
6     github: '',
7     linkedin: '',
8     website: '',
9     address: '',
10    city: '',
11    state: '',
12    zip: '',
13    country: '',
14  },
15  experience: [{
16    company: '',
17    city: '',
18    state: '',
19    position: '',
20    dateFrom: '',
21    dateTo: '',
22    primaryWorkBrief: '',
23    impact1: '',
24    impact2: '',
25    impact3: '',
26    impact4: '',
27    isVisible: ,
28  }],
29  education: [
30    {
31      school_name: '',
32      school_city: '',
33      school_state: '',
34      degree_name: '',
35      domain_name: '',
36      dateFrom: '',
37      dateTo: '',
38      isVisible: ,
39    },
40  ],
41  achievements: [
42    {
43      title: '',
44      description: '',
45      dateFrom: '',
46      dateTo: '',
47      isVisible: ,
48    },
49  ],
50  technicalSkills: [
51    {
52      category: '',
53      keywords: [{ name: '', level: }],
54      isVisible: ,
55    },
56  ],
57  projects: [
58    {
59      project_title: '',
60      dateFrom: '',
61      dateTo: '',
62      link: '',
63      skills_used: ['', ''],
64      details: ['', '', ''],
65      isVisible: ,
66    },
67  ],
68 ],
69 };
```

Figure 4: Resume Template in JSON

SEMANTIC UI - Semantic UI is a front-end development framework designed for theming that contains components that can be used to build responsive websites easily. It unpacks a variety of themes utilizing concise HTML, simplified CSS, and intuitive JavaScript and integrates perfectly with React, Angular, etc. It enables the users to design stunning designs by selecting their preferences from a ton of UI components. This helps reduce the file size and has lower load times with easy customizations that can be done on the go as it uses semantic class names. Semantic UI was chosen for this project because it works perfectly with React and provides the UI components needed with efficient and smaller code segments.

REST API - REST is an acronym for REpresentational State Transfer which is an architectural style commonly used in Web Services. RESTful API is built on six guiding principles consisting of a uniform interface, separation of the client and the server, statelessness, ability to reuse the response data, have a multi-layered system, and allow running the API code in form of applets or scripts. The transaction within a RESTful API is broken down into small modules that address a part of the transaction. Further, the calls within the REST API are stateless making it compatible with cloud-based applications as they can be easily redeployed if something fails, and can scale to accommodate load changes. The database being used in the application is a cloud-based implementation of a No SQL database thus making RESTful architecture the best choice for the development of this application. The HTTP methodologies used by the REST architecture that is used in the project are

GET - This request is used to retrieve a resource from the server.

PUT - This request is used to update an existing resource on the server.

POST - This request is used to create a new resource on the server.

DELETE - This request is used to delete a resource on the server.

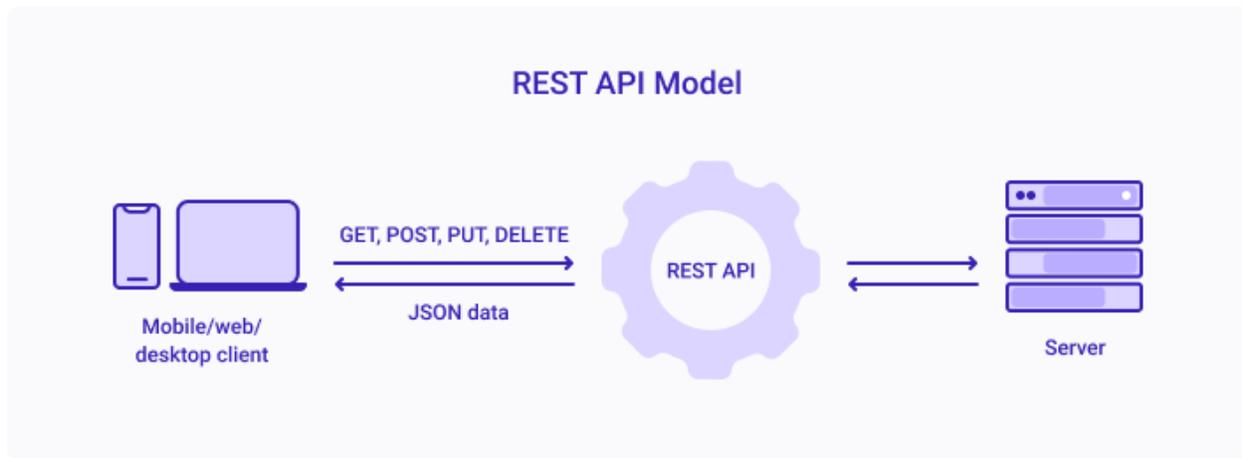


Figure 5: REST API Model

5. RESULTS

Rezume aims to help students at the university create customized resumes for their job hunts just with a few clicks and fills. On launching the URL of the website the user is taken to Rezume's landing page which has the option for the existing users to either register or the existing users to sign in to their profile.

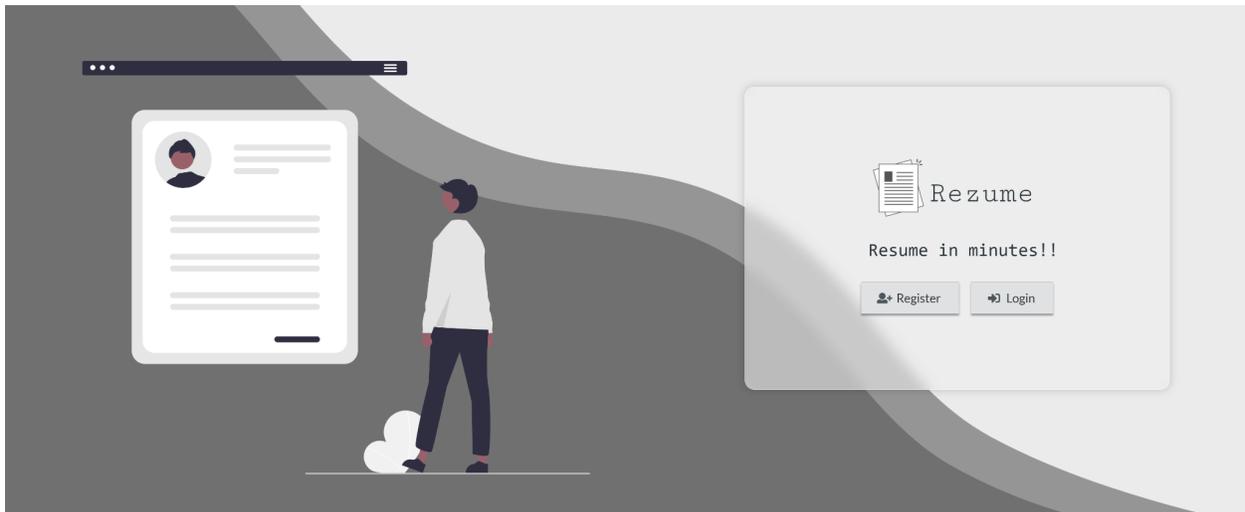


Figure 6: Landing Page

Two side-by-side screenshots of web forms. The left form is titled 'Register' and has fields for 'Username', 'Email', 'Password', and 'Confirm Password', with a 'Register' button and a link 'Already Registered Login'. The right form is titled 'Login' and has fields for 'Email' and 'Password', with a 'Login' button and a link 'Don't have an account? Register'.

Figure 7: Register & Login Form

Enter User Profile Details

Name Name	Email Address Email Address	Phone Number Phone Number
Github Github	Linkedin Linkedin	Website Website
Address Address	City City	State State
Zip Zip	Country Country	

Save And Continue

Figure 9: Setup Profile

The View Profile button takes the user to the profile dashboard where the user can view the data they filled in the forms in cards. Each card allows the user to add new data in the section, update the data within the selected field, and delete the field.

<p>Education </p> <hr/> <p>School Name: Rochester Institute of Technology School City: Rochester School State: NY Start Date: b End Date: b Degree Name: MS Domain Name: IST</p> <p> </p>	<p>Profile</p> <hr/> <p>Name: Ayush Arora Email: aaa@aaa.com Phone: 5859571899 Github: bh Linkedin: bh Website: bh Address: 476 Clay Road City: Rochester State: New York Zip: 14623 Country: United States</p> <p> </p>
---	--

Figure 10: View Profile Components

Next is a font selector dropdown menu that lists all the Fonts that have been added by the admin. The user can select a font and the change will be visible on the resume on the right-hand side instantly.



Figure 11: Change Font Functionality

Similarly, change in the font size and the color of the template can be viewed on the screen instantaneously on the screen.

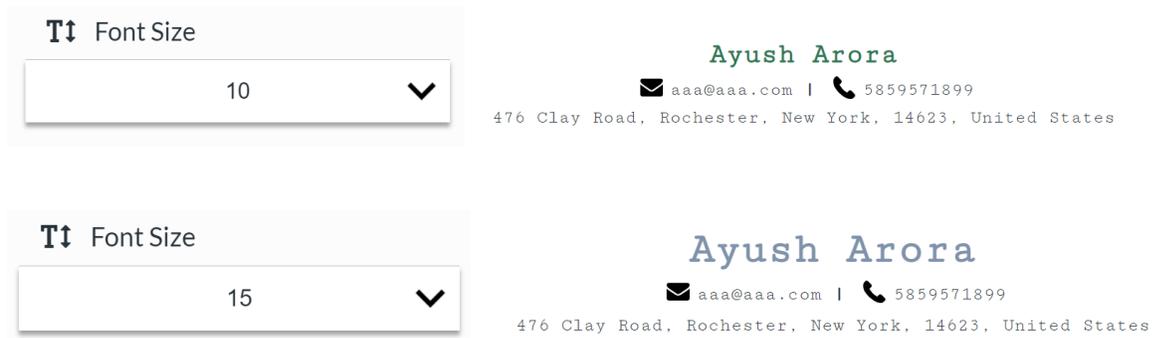


Figure 12: Change Font Size Functionality

The following image shows all the colors in the database being rendered. Clicking on the color will change the color of the design elements within the resume.



Figure 13: Change Color Functionality

The user can also toggle between the different templates that are available for the users. Currently, Rezume holds 3 professional-looking templates and the following figure shows the component which can be used to toggle between them.

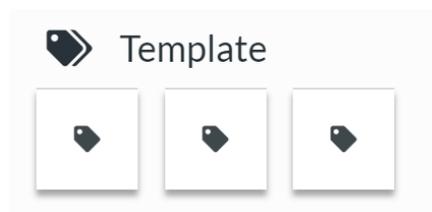


Figure 14: Template Selector

The next functionality is the Visibility of the sections on the resume. Each component that could be hidden is associated with a toggle switch that can either be switched on or off. Switching off a particular component will remove the component from the resume.

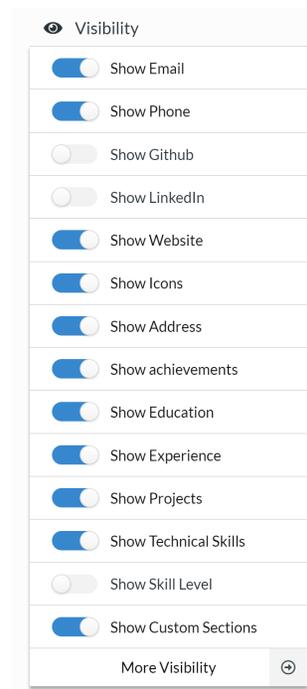


Figure 15: Visibility Component

The More Visibility component opens a modal that lists the data within the sections and can be toggled too in order to remove or add any data from the resume without actually deleting the data associated with it.

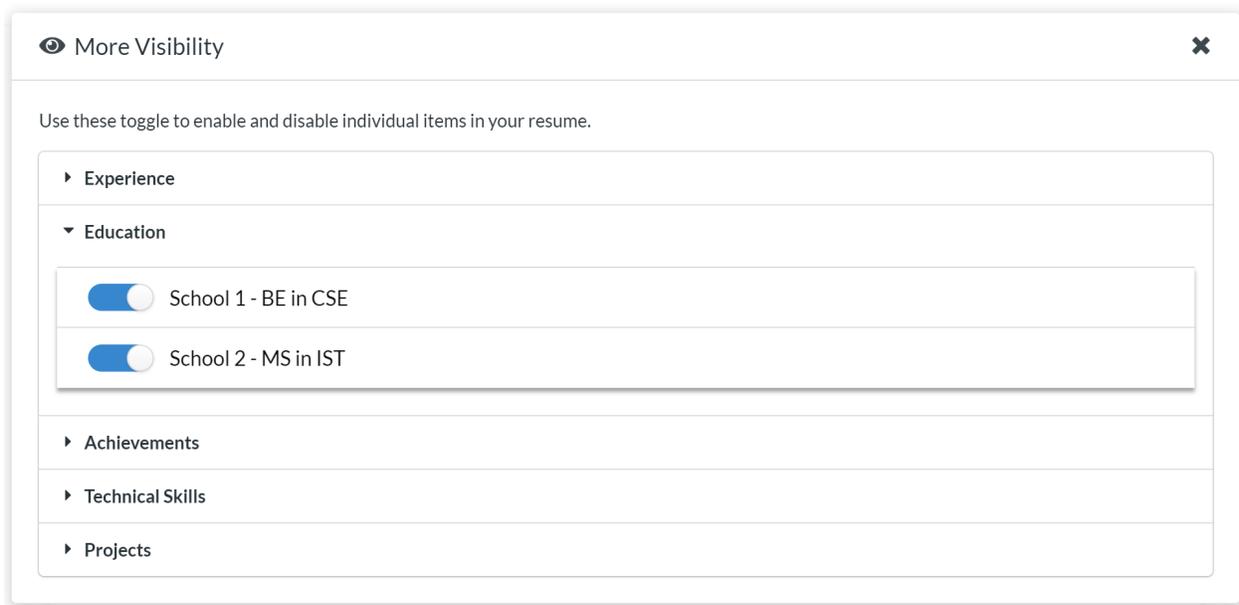


Figure 16: More Visibility Component

The next functionality on the sidebar is the section order changer that lets the user drag and drop the sections to set their order within the resume. The user can long click on the respective section and drag it upwards or downwards. This uses an array stored within the states that is then used to render the respective sections in the desired sequence.

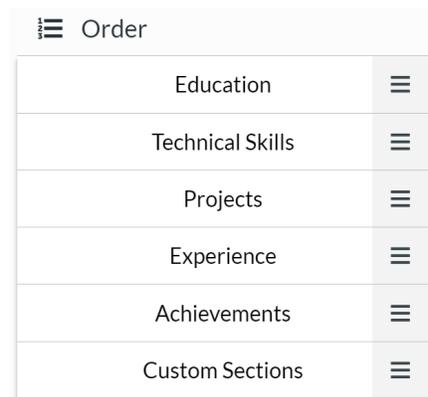


Figure 17: Order Changer

Next is the print resume button that lets users print the resume they have generated or just directly save it as a pdf using the Windows print dialog box.

John Doe

✉ email@domain.com | ☎ 123-456-7890 | 🌐 https://website.com
123 Main Street, City, ST, 00000, USA

Education

School 1 City, ST XX/XXXX - XX/XXXX
Degree 1 in Domain 1

School 2 City, ST XX/XXXX
Degree 2 in Domain 2

Skills

Development Languages

JavaScript, HTML, CSS

Technologies

MongoDB, Express, React, Node.js, Mocha, Passport, JWT, Chai, Redux, Git, GitHub, Gatsby

Custom Category

Item 1, Item 2, Item 3

Projects

Project 1 XX/XXXX
skill 1 skill 2
◦ Detail 1
◦ Detail 2
◦ http://projectLink.com

Project 2 XX/XXXX - XX/XXXX
skill 3 skill 4
◦ Detail 1
◦ Detail 2
◦ http://projectLink.com

Experience

Position 1 XX/XXXX
Experience 1, City, ST
◦ Brief description of your main tasks.
◦ Something awesome you did 1.
◦ Something awesome you did 2.

Position 2 XX/XXXX - XX/XXXX
Experience 2, City, ST
◦ Brief description of your main tasks.
◦ Something awesome you did 1.
◦ Something awesome you did 2.

Position 3
Experience 3, City, ST
◦ Brief description of your main tasks.
◦ Something awesome you did 1.
◦ Something awesome you did 2.

Achievements

Title 1 XX/XXXX
Achievements description 1

Title 2 XX/XXXX - XX/XXXX
Achievements description 2

Title 3 XX/XXXX
Achievements description 3

Figure 18: Resume Template 1

✉ email@domain.com | ☎ 123-456-7890
 🌐 https://website.com
 123 Main Street, City, ST, 00000, USA

John Doe

Education

School 1 City, ST XX/XXXX - XX/XXXX
Degree 1 in Domain 1

School 2 City, ST XX/XXXX
Degree 2 in Domain 2

Skills

Development Languages

JavaScript, HTML, CSS

Custom Category

Item 1, Item 2, Item 3

Technologies

MongoDB, Express, React, Node.js, Mocha, Passport, JWT, Chai, Redux, Git, GitHub, Gatsby

Projects

Project 1 XX/XXXX
skill 1 skill 2

- Detail 1
- Detail 2
- <http://projectLink.com>

Project 2 XX/XXXX - XX/XXXX
skill 3 skill 4

- Detail 1
- Detail 2
- <http://projectLink.com>

Experience

Position 1 XX/XXXX
Experience 1, City, ST

- Brief description of your main tasks.
- Something awesome you did 1.
- Something awesome you did 2.

Position 2 XX/XXXX - XX/XXXX
Experience 2, City, ST

- Brief description of your main tasks.
- Something awesome you did 1.
- Something awesome you did 2.

Position 3
Experience 3, City, ST

- Brief description of your main tasks.
- Something awesome you did 1.
- Something awesome you did 2.

Achievements

Title 1 XX/XXXX
 Achievements description 1

Title 2 XX/XXXX - XX/XXXX
 Achievements description 2

Title 3 XX/XXXX
 Achievements description 3

Figure 19: Resume Template 2

✉ email@domain.com | ☎ 123-456-7890 | 🌐 https://website.com
123 Main Street, City, ST, 00000, USA

JOHN DOE

EDUCATION

SCHOOL 1 CITY, ST XX/XXXX - XX/XXXX
Degree 1 in Domain 1

SCHOOL 2 CITY, ST XX/XXXX
Degree 2 in Domain 2

SKILLS

DEVELOPMENT LANGUAGES

JavaScript, HTML, CSS

TECHNOLOGIES

MongoDB, Express, React, Node.js, Mocha, Passport, JWT, Chai, Redux, Git, GitHub, Gatsby

CUSTOM CATEGORY

Item 1, Item 2, Item 3

PROJECTS

PROJECT 1 XX/XXXX
skill 1 skill 2
◦ Detail 1
◦ Detail 2
◦ <http://projectlink.com>

PROJECT 2 XX/XXXX - XX/XXXX
skill 3 skill 4
◦ Detail 1
◦ Detail 2
◦ <http://projectlink.com>

EXPERIENCE

POSITION 1 XX/XXXX
Experience 1, City, ST
◦ Brief description of your main tasks.
◦ Something awesome you did 1.
◦ Something awesome you did 2.

POSITION 2 XX/XXXX - XX/XXXX
Experience 2, City, ST
◦ Brief description of your main tasks.
◦ Something awesome you did 1.
◦ Something awesome you did 2.

POSITION 3
Experience 3, City, ST
◦ Brief description of your main tasks.
◦ Something awesome you did 1.
◦ Something awesome you did 2.

ACHIEVEMENTS

TITLE 1 XX/XXXX
Achievements description 1

TITLE 2 XX/XXXX - XX/XXXX
Achievements description 2

TITLE 3 XX/XXXX
Achievements description 3

Figure 20: Resume Template 3

The resume templates in the images above are developed in conjunction with the logical rules of design and typography. The features such as changing the font size and font are provided to let the user customize the resumes according to their preferences while only allowing a set of standard fonts on the portal. The templates are designed in a single-column data representation with clear and bold headings. This enables the ATC checkers to easily parse the resume enabling more hits.

The user can also download all the data linked to their resumes in the form of a JSON file and can be then uploaded on other templates that support JSON data to create awesome resumes. Similarly, JSON data manually coded by the users or from other JSON resume websites can also be uploaded on Rezume. The JSON format for the resume can be viewed by the user by clicking on the round file icon beside the Load from File button.



Figure 21: Download & Load Data Functionality

The user can also select the page size from the standard sizes they would prefer their resume to be generated in. Currently, Rezume supports Letter, A4, and Legal sizes.

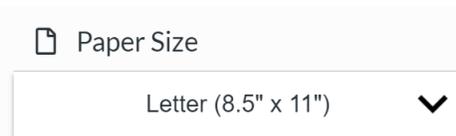


Figure 22: Paper Size Functionality

All the customizations made to the resume can be stored in the local storage and will be reloaded when the user visits the website next. This is optional for the user and can be toggled by the autosave switch. The delete icon next to it can be used to delete all the data stored in the local storage. This would also delete the tokens stored in order to keep the user signed in on their device. Finally, the user can directly sign out of the website and can keep their data stored by clicking on the Signout button.



Figure 23: Local Storage & Sign Out Functionality

The following picture shows the state chart of the tools used within the application. The tools state within Redux and react are responsible for handling all the above-mentioned functionalities.

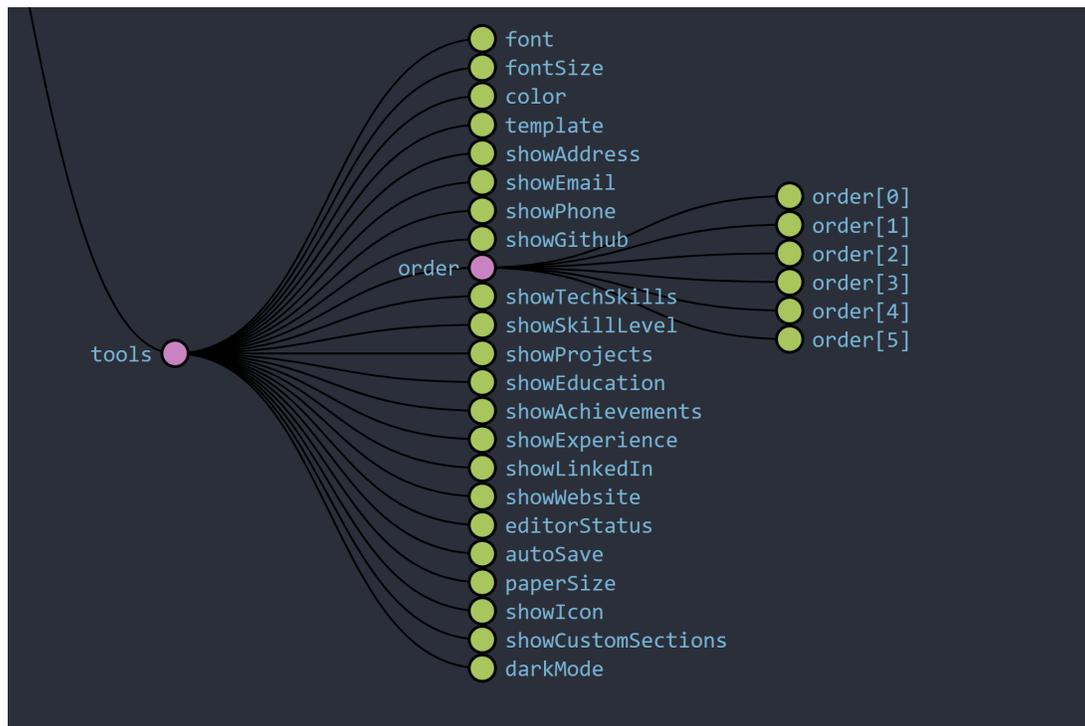


Figure 24: Functionality State Chart in Redux

The admin is directed to an admin dashboard directly and can use it to customize the Resume portal for the users. The following image shows the Admin Dashboard. Accessing this state data within components directly wouldnt be possible on React without passing the data throughout the components. React only allows direct passing of data from parent to child component and an unconventional way to pass it in the reverse direction using callbacks. Hence using a Redux state helps accessing the data at any components directly.

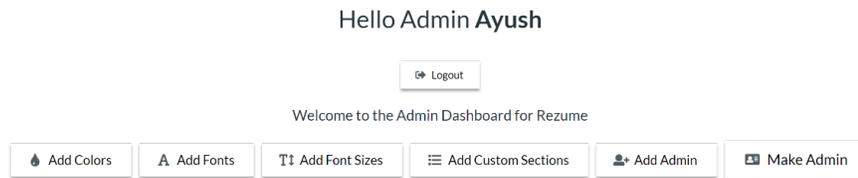


Figure 25: Admin Dashboard

The admin can add new colors in the Resume for the users to select for their resume templates using the Add Colors button. Similarly, Fonts and Font Sizes can be added by the admin which is rendered in the User dashboard to be used according to the user's needs. The Add Admin button lets a new admin be registered. Make Admin Button list all the users both admin and non-admin and the Admin can toggle their roles by submitting the form. Add Custom Section button lets the Admin add sections to the resumes which in turn focuses on increasing the scalability of the website. For every custom field created by the admin, a user can add an object with a title and description. The following figure shows the schema of the fields created for custom sections. The user field stores the detail of the user to whom the field belongs, customField is the detail of the custom field like Publication or Certification, and the title and description stores the title and description of the publication or certification.



Figure 26: Schema of the data linked to custom sections

Each of the buttons mentioned above opens a modal with a form like the one shown in the following picture. The font size form accepts an integer size, the colors form accepts a color in the RGB color format, the font form lists a dropdown of a range of fonts that could be selected.

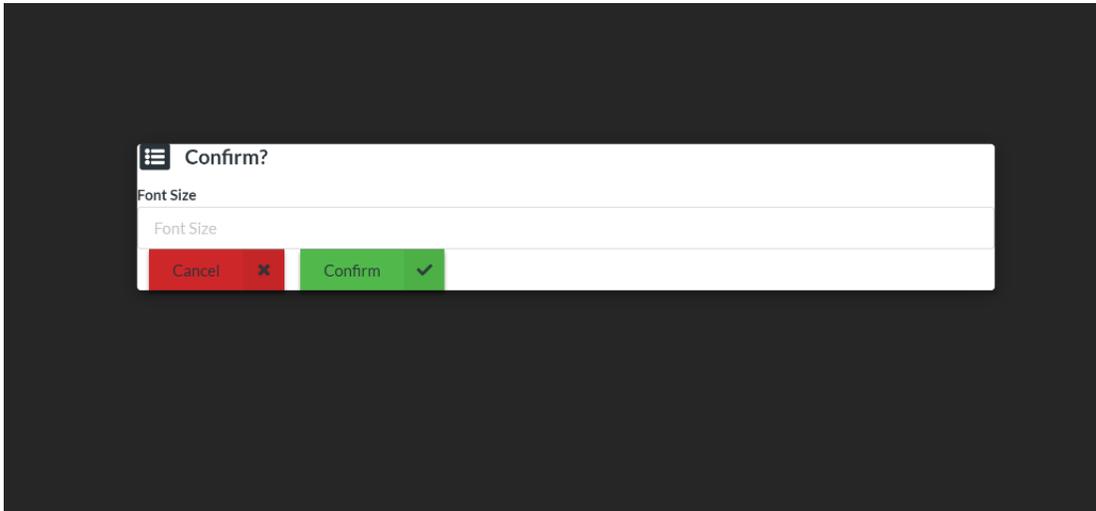


Figure 27: Admin Form Modal

The following image shows a snip of the Make Admin form. The select menu lists all the users registered on the website. The admin selects the user and submits the form to change the authorization of the respective user.

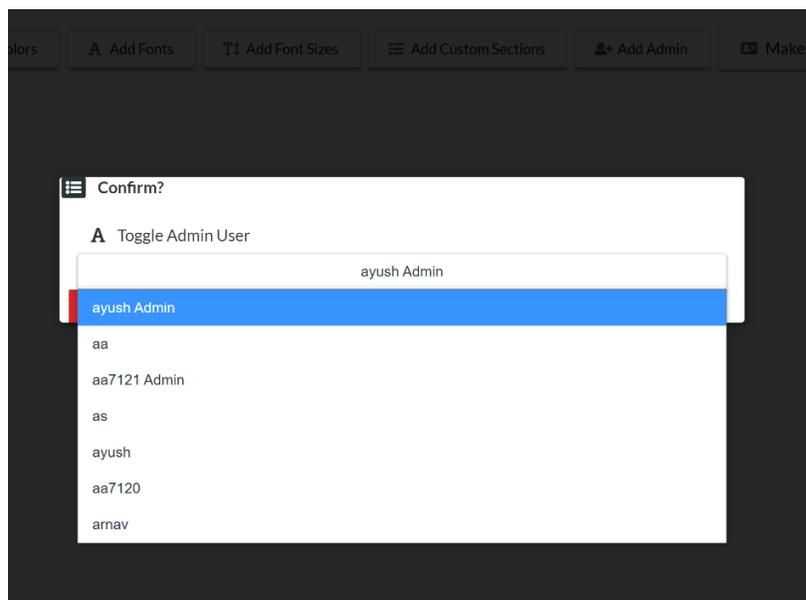


Figure 28: Toggle Admin

Each of these modals is created using the states within React. Each component has a separate state that is either true or false. The modal is set to pop up its respective state value is set to true. The value is set to true on the button click and false once the user either clicks cancel or confirm within the Form. The following figure shows the Fonts Modal as open and all other modals to be closed.

```
state
  adminUserModalOpen: false
  colorsModalOpen: false
  customFieldsModalOpen: false
  fontSizesModalOpen: false
  fontsModalOpen: true
  makeAdminModalOpen: false
```

Figure 29: Form Modal States

6. Future Works

The aim of this application is to help students across fields generate Resumes easily. Significant effort has been put in to make the application visually appealing and smooth. But having experts in UI/ UX design wireframe templates for the websites using the current trends in the industry would help make the website much better and user friendly. Further, this website needs thorough usability testing where representative users can try it and give their advice on how to make it better. This would also help understand how easy it is for users to navigate through the website, the complexity of the interface, how usable are the functionalities, etc. Rigorous application testing is also required before it can be hosted so as to discover any major bugs.

The requirements for this application were drawn out based on the research of the existing systems and by understanding the problems faced by the students during their resume creation process. There wasn't a large analysis to understand what features that the users might actually want with such a portal. Insight on that would be really helpful to expand the project further. There are certain features like letting the user save the resumes they built on the portal stored within their profile so that they can access them by easily signing in. Resume sharing functionality via email could also be a useful feature. Having the system generate cover letters in similar templates to the resume would be a useful feature that could be implemented in the future.

The application is built upon REST API with routes being protected using authentication and authorization middlewares. Rigorous testing and update of the API would be needed to remove any security concerns like Broken Access Control, Sensitive Data Exposure. Action must be taken to remove injection attacks. As this application is highly dependent on data being properly stored and accessed, proper client-side and server validation is necessary. Injecting the wrong format of data within the application might lead to opening access to unauthorized users and would be a major security concern. There are no complexity rules present in the password fields to secure the passwords from hit and trial errors. Also in the future providing the users with the power to log in and register using their social media handles like Facebook, LinkedIn, google, etc. Finally, a feature of forgot password is extremely important and can be implemented so that users can reassess their accounts.

A lot of effort has been put into making the application fast and efficient by reducing the number of API calls being made and using the web browser's ability to store data within itself. This can

still be further improved by researching adequate ways to get the tasks done in the best way possible. Work can also be put in the React components to make them faster and efficient. Currently, no tests were performed to measure the complexity and render time of the components in the front end.

This application gives a push towards making itself scalable by introducing custom sections that can be added by the admin. Currently, the functionalities linked to all the other sections weren't implemented because of the complexity of its design. More efforts into this module will mean it would be possible for the admins and then eventually users to add new sections to the resumes making it completely customizable.

This application currently only includes three templates that are professional-looking. A great future addition would be to have a lot more options for the user to work with. This would include some templates for students in creative fields helping them build resumes customized for their needs and taste.

7. Conclusions

This application serves the purpose to help students at universities to design and create their resumes with ease using forms and profiles that we all are so used to in terms of social media. The development of this application has taught me a lot about the application development process ranging from planning, requirements analysis to the deployment of the project. Planning and understanding of the requirements were some of the most challenging aspects of the project after the build phase itself. Resumes are a crucial part of who a student is professionally and having them curated to their needs is difficult. Thus a standard format is considered for the sake of this project so as to perfectly fit the needs of the companies job postings. Further, a lot of effort was put into developing forms that can be multiplied dynamically for the user to insert multiple entries within one section. A good amount of research and input from the Professors helped me tackle this problem easily.

The application uses a lot of technologies that I had some expertise on and some of which I learned focusing on this project. Developing the front end and its integration with the amount of data that comes in was a difficult task. Further implementing functionalities like reorganizing the sections within the resume, visibility of the sections and individual sections was the most challenging part. This utilized the core concept of React states, Redux and Hooks, and made me understand these concepts in a much better way. Developing the User Interface I had in my mind for each component including the functionalities was a very complicated task. This was thus done using the Semantic UI framework that keeps a large set of components pre-built which can be smoothly and efficiently integrated within React.

This project also helped me improve my research skills a lot as this is a relatively unexplored topic when it comes to scholarly articles and journals. Thus to better understand this project I had a lot of communication with the students around me to understand their point of view on this project and to plan the best way to help them out. The Internet became my best resource providing me with tonnes of information on the various aspects of the project from the design of the resume to how to make the website user-friendly. Further, this project made me realize that the journey of becoming a better software developer is a cycle of finding a solution to do a task, realizing you were wrong, and reiterating the whole process to bring out the best possible solution for the users. Overall this project has been a perfect portal for me to portray my

learnings from the Masters in Information Sciences and Technology program and also trains me better for my upcoming future endeavors as a software developer.

8. References

- [1] I. Wu, K. Wayne, S. Lakka, and T. Rai, “Pro-Resume: The Infographic Resume Builder,”
- [2] K. D. Ingale, “A Review Paper on Resume Portal,” p. 5, 2018.
- [3] S. Risavy, “The Resume Research Literature: Where Have We Been and Where Should We Go Next?,” *Journal of Educational and Developmental Psychology*, vol. 7, p. 169, Feb. 2017, DOI: [10.5539/jeep.v7n1p169](https://doi.org/10.5539/jeep.v7n1p169).
- [4] “How to Beat an Applicant Tracking System (ATS) with a 100% Pass Rate.” <https://www.visualcv.com/blog/how-to-beat-the-applicant-tracking-system/> (accessed Jul. 18, 2021).
- [5] “Resume Statistics [2021]: What Job Seekers Need To Know – Zippia.” <https://www.zippia.com/advice/resume-statistics/>.
- [6] A. Tiwari, S. Vaghela, R. Nagar, and M. Desai, “Applicant Tracking and Scoring System,” vol. 06, no. 04, p. 6, 2019.
- [7] C. Martin-Lacroux and A. Lacroux, “Do Employers Forgive Applicants’ Bad Spelling in Résumés?,” *Business and Professional Communication Quarterly*, vol. 80, no. 3, pp. 321–335, Sep. 2017, DOI: [10.1177/2329490616671310](https://doi.org/10.1177/2329490616671310).
- [8] “Cover Letters and Resumes | Career Services and Co-op | RIT.” <https://www.rit.edu/careerservices/students/job-search/cover-letters-and-resumes#resumes> (accessed Jul. 21, 2021).
- [9] P. W. Lab, “Résumé Design // Purdue Writing Lab,” *Purdue Writing Lab*. https://owl.purdue.edu/owl/job_search_writing/resumes_and_vitas/resume_design.html (accessed Jul. 21, 2021).
- [10] *You have 7.4 seconds to make an impression: How recruiters see your resume.* (n.d.). Ladders | Business News & Career Advice. Retrieved December 5, 2021, from <https://www.theladders.com:443/career-advice/you-only-get-6-seconds-of-fame-make-it-count>
- [11] Malewade, S. M., & Ekbote, A. (n.d.). Performance Optimization using MERN stack on Web Application. *International Journal of Engineering Research*, 10(06), 5.
- [12] *Getting Started with Redux | Redux.* (n.d.). Retrieved December 5, 2021, from <https://redux.js.org/introduction/getting-started>
- [13] *Getting Started—JSON Resume.* (n.d.). Retrieved December 5, 2021, from <https://jsonresume.org/getting-started>

APPENDIX

PROJECT PROPOSAL

ON

“Web-Based System to generate Resumes for University Students”

CAPSTONE PROJECT

Information Sciences & Technology

CHAIR

Prof. DAN BOGAARD

COMMITTEE MEMBER

Prof. JOHN-PAUL TAKATS

PROPOSAL BY

AYUSH ARORA

ABSTRACT

A resume is created by a job applicant to showcase their qualifications for a position. It contains the summary of the applicant's experience and education making them a good fit in the company for the job opening. For new graduates, listing these qualifications becomes a strenuous job as the amount of practical knowledge they hold is limited. Research suggests that about 75% of the

resumes don't ever make the threshold of the company and these numbers are even higher for fresh graduates[5]. Hence, a well-designed resume is necessary to help make a compelling impact. The proposed website aims to help the students draft and rework their resumes based on a simple profile-based setup which will help them save invaluable time and provide them with standard and well-designed resumes.

INTRODUCTION

Resumes are written with the intent to emphasize the candidate's strengths and to show the employer how they would be a good match for a job opening. Employers generally look for the candidate's job history and their advancements in the work environment. A candidate can get a job interview and make a good first impression solely based on an impressively designed resume. There are a lot of agencies as well as websites that help students and other candidates draft resumes for a significant price. The websites focused on providing templates for the resumes leave the hard work of designing the resumes for the candidates. A candidate spends about hours updating and personalizing the resumes as per the job requirements instead of focusing on the job itself.

Larger companies use Applicant Tracking Systems (ATS) that screen a large number of applications and filter out the best candidates for the human recruiters to view. This means when a student applies for a job their resumes don't reach the recruiter at all. Hence, designing an ATS-friendly resume is necessary, but this is seldom known to the candidates.

The proposed project focuses on providing well-designed resume templates specifically for new graduates. Each student will be able to build a free profile that holds data closely linked with their LinkedIn profiles. The stored data could be edited on the go and create quick and personalized resumes for the students quickly and efficiently. The project will also host other functionalities like grammar check and spell check to help students improve their resumes.

The motive is to store the user data in a format similar to social media. This profile will be editable and will be converted into a resume based on the templates selected by the user within seconds. This means the users will save a lot of precious time spent on aligning their texts, handling spacing and margins within MS Word, or even writing code on LaTeX.

A user will be able to add features like personal information including links to their LinkedIn profiles or online portfolios, a summary of their qualifications and skills, their education, the experience they have in the respective field. The provided resume templates will follow several rules like the four-quadrant rule, which will use columns to represent information, and other design and typographic rules. Users would also have an option to select from a variety of colors and a set of standard fonts to portray their true selves via the resume.

The user will also be able to create a cover letter from the content they pen down which will help them give an authentic touch to their application.

The user can also refer to do's and don'ts based on research for specific sections that they are adding to their application. This would help them reduce the common mistakes students make while making a resume. An integrated API to check to cover spell-checks as well the grammar of the resume and cover letter would be integrated. This would help them develop professional-looking and -feeling resumes.

PROBLEM STATEMENT

Developing resumes is a difficult and hectic job. It takes a major portion of the time a candidate slots for their job hunts. The other option is to spend hundreds of dollars on agencies or websites to get the job done which is rarely preferred. With the advent of all the job portals online the applicants, most of the time end up using the quick apply button without properly updating their resumes. This leads to a series of rejections and making it practically impossible to land a good job. Almost all the Fortune 500 companies use the Applicant Tracking Systems to sort out resumes in the initial stages. Due to the lack of knowledge about how the process works the job seekers seldom subconsciously forget some of the major steps like grammar and spelling checks within their resume. As stated in [7] these small mistakes coupled with unprofessional resume designs will result in no callbacks. The candidates that do the required research and are able to solve these problems end up not being able to recreate the templates in a professional manner.

GOALS AND OBJECTIVES

The purpose of this proposal is to present a web-based application that offers job seekers the opportunity to create and update their CVs with ease. The application stores the profile data of the user and creates resumes using designed templates. These templates are some of the most successful templates among ATS systems and HR. Other features like spellcheck and grammar check automate a tedious process and help the candidate stand out.

Goal 1 - Development of the Basic Application for the data transaction of the User Record.

Objective 1: Select the technologies best suited for the development of the project.

Objective 2: Create a database and the data models to store the desired data.

Objective 3: Develop and deploy the backend and create dummy data.

Objective 4: Design and code a user-friendly front end to handle basic data transactions.

Objective 5: Create a User Authentication system to keep the data secure and to link it to the users.

Goal 2 - Design the Resume Templates for the user to choose from.

Objective 1: Study the resume design principles to create effective resumes.

Objective 2: Understand the Applicant Tracking Systems and explore features that can be added.

Objective 3: Code the templates using the selected framework.

Goal 3 - Create the Admin and Researcher/ Faculty functionality.

Objective 1: Build the authorization for Admins and Professional Editors.

Objective 2: Create a Data API and dashboard for admin.

Objective 3: Develop the views for Researchers and faculty members to help create multiple page resumes that enlist their publications and conferences.

Goal 4 - Provide the other functionality to improve the user experience and generate better resumes.

Objective 1: Research features that would help users create better resumes.

Objective 2: Develop the features found in previous steps.

Objective 3: Integrate the features within the application.

Objective 4: Run extensive tests on the functionality and usability of the website and make appropriate updates.

LITERATURE REVIEW

1. Studying the existing systems

A huge amount of systems existing on the web were explored and their best features will be added to the software. The biggest problem in most of these systems is that they endure a high cost every time a user builds a resume. The services do come with expert advice to improve the resume but do cost about hundreds of dollars. There are a few systems that are free of cost like the one I. Wu [1] describes the Pro Resume that focuses only on infographic resumes. Also, the website has only a single template but gives some design flexibility to the user. Similarly, Ingale [2] in her proposal of the Resume Portal focuses on an integrated system where the user can apply for job vacancies posted on the portal. This portal covers a lot of bases like building resumes and getting jobs with a lack of research and focusing on vastly used Applicant Tracking Systems that companies already use. It also assumes the companies will be posting their job

requirements on the portal which makes the process of generating resumes coupled to the requirements. The one thing that can be useful is creating links to user resumes that can be shared by the user instead of uploading a PDF file everywhere. Other systems propose resume generators with multilingual capabilities or the use of AR markers to visualize various parts of printed resumes on the web.

2. Research on “ How to make ATS Friendly Resumes”

The Ladders eye-tracking study suggested that a recruiter spends about 7.4 seconds on a resume hence it is very important to focus on having a clean-looking resume, having short declarative statements, and most importantly having the standard headings and subheading without any extraordinary design experiments. Thus, the details presented by Risavy in the Resume Research Literature[3] serve a very important role in the development of every template for the project. The valuable tips shared by Clift in the “How to beat ATS” [4] also list a checklist that should be followed while making resumes.

The design tips shared on the [7] RIT’s Career Services and Co-op and Perdue’s Resume Design [8] article also provide important insight on the checklist created previously.

3. Exploration of features to improve the user experience

The overall functioning of an ATS is studied and the application proposed by Tiwari and et al in “Applicant Tracking System”[5] focuses on Natural Language Processing to only detect job-related words. Using the knowledge we have about the design of resume templates, a rubric of an improved system can be drafted. This would be done by including the factors listed and a higher preference would be given to the spellings and grammar of the Resume.

METHODOLOGY

1. Research, Synthesise and Elect (Goal 1 Obj. 1, Goal 2 Obj. 1 & 2, Goal 3 Obj. 1)

Building a user-friendly full-stack website involves a lot of background research on the technologies and frameworks that would be used to build the application. This decision impacts the complexity and the friendliness of the project as a whole. Similarly, a prudent

analysis of the user experience needs to be developed using a tool like Figma. One of the major outcomes of this project will be a collection of ATS-friendly resume templates for people across various domains. An extensive study of the design of templates is accompanied by necessary features. The system would check for how friendly a resume is for an ATS machine and features that would make the overall experience of the user pleasant.

2. Development of the basic project stack (Goal 1 Obj. 2, 3, 4 & 5)

The project is developed using the Model View Controller (MVC) architecture. The model acts as the brain of the project and interacts with the database. The database being used for the application is a cloud-based non-relational MongoDB database. The model will be developed using Express.js which is a web application framework based on Node.js. The View is what the user actually sees and interacts with or the User Interface of the Application. The views will be developed using the javascript library React which makes the development process easier with the power to reuse the developed components. Finally, the Controller is the logic that glues the Model and the Views together. It deals with the transaction of data between the Models and the Views. The controller also performs other business functionalities like grammar check, spell check, authentication, etc. The following figure depicts the functioning of the application based on the MVC Architecture.

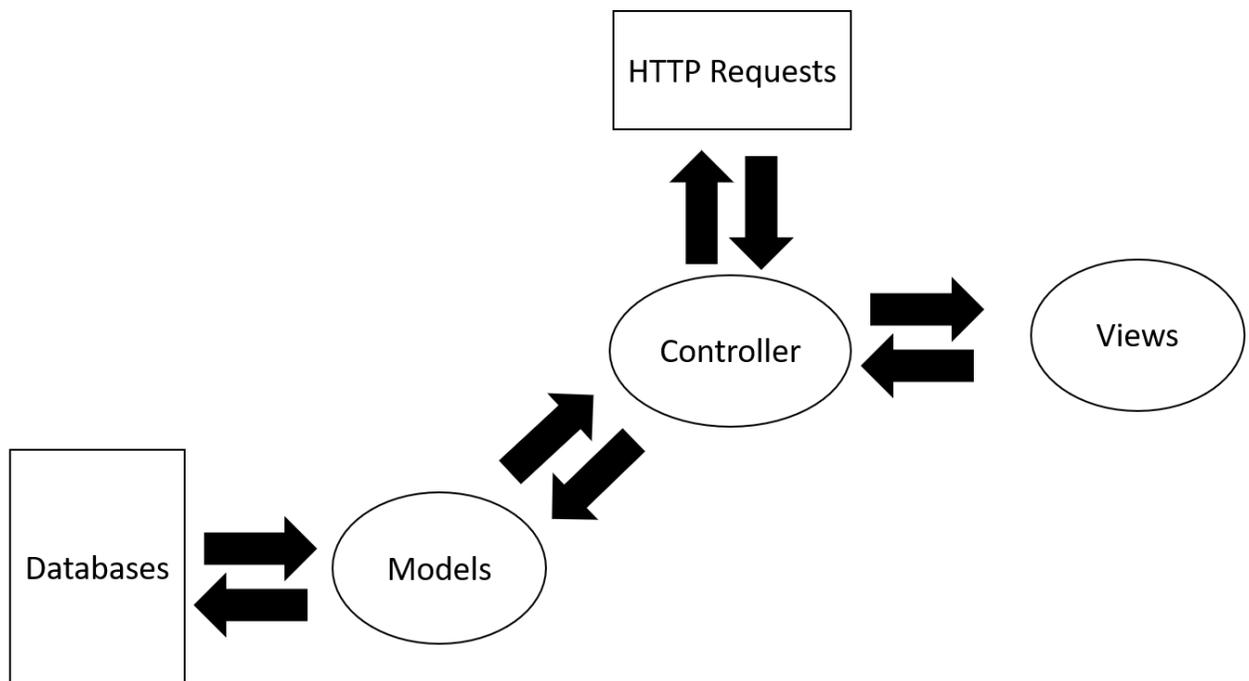


FIGURE: MVC ARCHITECTURE

Using the above-selected technologies a basic website would be developed that takes care of tasks like User Authentication, collection of the data by using forms, and programming

a basic data API. The API will be responsible for the interactions between the website and the database including actions like storing and retrieving the data. Each user's profile information is then displayed in the form of a JSON Resume. JSON Resume is a community-driven open-source initiative to create a JSON-based standard for resumes.

3. Designing and coding web-based resume templates and Integration (Goal 2)

Sample resume templates will be designed using the knowledge of the successful templates discovered in the previous stage. Once designed the templates will be coded into components of websites.

Further, the resume components will be integrated into the website and the templates will be tested on the existing user data.

4. Developing different roles such as Admin, Researcher/ Faculty (Goal 3)

Having different levels of authorization will allow people with different roles to perform different levels of functionalities. An admin can view the statistics related to the website and how the features within the website are helpful to the students.

The faculty members and researchers have different requirements while creating the resumes which enlists their publications and conferences. Functionality to create multiple pages resumes with templates would be provided for these users.

5. Adding the selected features into the program (Goal 4)

Next, the various features explored in the previous steps are developed. This includes functions like spelling and grammar checker, a rule-based chatbot, live preview, showing and removing certain sections of the resume. The resume will finally be downloadable into PDF format, sharable in form of links, etc.

The website will also include verb lists according to the job and the user can use those verbs right out of the box to improve the resume. The word list will be derived from trusted sources and would be ATS Friendly. Finally, a resume scorer will be able to score the resume based on the rules of the ATS resume systems.

LIMITATIONS

The project will only be able to host a limited number of templates because of time constraints. Also, all the details will solely be based on research from various credible sources but no expert will directly be available within the project.

DELIVERABLES

1. A Web-based Application that can be used by the students to make their resumes efficiently.
2. An Instruction Manual explaining the functionality of the application.
3. A report explaining in detail the importance of such a system and how the developed project is an upgrade to the preexisting systems free of cost.
4. A PowerPoint presentation explaining, in brief, the important details of the project.

TIMELINE

The plan is to develop this project in the Fall Semester 2021 towards the completion of the Capstone Project in MS in IST degree. The following is the basic timeline that will be followed for the completion of the project.

No.	Task	Start Date	End Date
1	Pre Proposal	06/15/2021	06/25/2021
2	Literature Review and Proposal	06/27/2021	08/07/2021
3	Research, Synthesize and Elect	07/15/2021	07/22/2021
4	Development of Basic Project Stack	07/22/2021	08/22/2021
5	Development of Resume Templates	08/23/2021	10/07/2021
6	Addition of user roles	10/01/2021	10/15/2021
7	Additional Features	10/15/2021	11/07/2021
8	Rough Draft of Project Report	10/20/2021	11/07/2021
9	Final Project Report and Presentation	11/07/2021	11/30/2021*
10	Submission		12/01/2021*

- * Tentative dates to be finalized later.

CONCLUSION

The job-seekers applying for jobs spend the majority of their time updating their resumes according to the job requirements. This task is especially tedious for graduates because of the limited amount of experience they have. Thus, the project proposes a web-based application that can help these students in their job hunt by automating the resume creation process free of cost. The users will be able to save their data in the form of a profile and this data can be updated every time they apply for a new job post. They also follow the rules required by the Applicant Tracking System for creating these resumes using the pre-designed templates.

REFERENCES

- [1] I. Wu, K. Wayne, S. Lakka, and T. Rai, “Pro-Resume: The Infographic Resume Builder,”
- [2] K. D. Ingale, “A Review Paper on Resume Portal,” p. 5, 2018.
- [3] S. Risavy, “The Resume Research Literature: Where Have We Been and Where Should We Go Next?,” *Journal of Educational and Developmental Psychology*, vol. 7, p. 169, Feb. 2017, DOI: [10.5539/jeep.v7n1p169](https://doi.org/10.5539/jeep.v7n1p169).
- [4] “How to Beat an Applicant Tracking System (ATS) with a 100% Pass Rate.” <https://www.visualcv.com/blog/how-to-beat-the-applicant-tracking-system/> (accessed Jul. 18, 2021).
- [5] “Resume Statistics [2021]: What Job Seekers Need To Know – Zippia.” <https://www.zippia.com/advice/resume-statistics/>.
- [6] A. Tiwari, S. Vaghela, R. Nagar, and M. Desai, “Applicant Tracking and Scoring System,” vol. 06, no. 04, p. 6, 2019.
- [7] C. Martin-Lacroux and A. Lacroux, “Do Employers Forgive Applicants’ Bad Spelling in Résumés?,” *Business and Professional Communication Quarterly*, vol. 80, no. 3, pp. 321–335, Sep. 2017, DOI: [10.1177/2329490616671310](https://doi.org/10.1177/2329490616671310).
- [8] “Cover Letters and Resumes | Career Services and Co-op | RIT.” <https://www.rit.edu/careerservices/students/job-search/cover-letters-and-resumes#resumes> (accessed Jul. 21, 2021).
- [9] P. W. Lab, “Résumé Design // Purdue Writing Lab,” *Purdue Writing Lab*. https://owl.purdue.edu/owl/job_search_writing/resumes_and_vitas/resume_design.html (accessed Jul. 21, 2021).