**Resume Builder Web Application**

**Increment-02**

**Team Leader:** Ravi Krishna Teja Dachepalli (16300464)

**Team members :**

1.Jayanth Golla

2.Ravi Krishna Teja Dachepalli

3.Anudeep Thippireddy

4.Manoj Desetty

**Project Description (Background Story):**

For any working individual in any business, creating a resume is a time-consuming effort. It must be kept short, straightforward, and current with work experience, and it must be updated on a regular basis.

This project will guide you through the steps necessary to create a resume builder using ReactJS and NodeJS. Implementing the project will provide you the satisfaction of being able to generate it on your own while also assisting working pros.

The main objective is to develop a web application that will auto-generate a nice and properly formatted Resume from the information filled up in a form.

Here the resume contains ,

* Contact Information.
* Career Objective.
* Skills and Abilities
* Work experience
* Educational qualifications

**Project Workflow:**



There are different stages involved in creating this web application which gives us some good resume pdf documents as output through the process that can be followed to build your resume-builder using ReactJS and NodeJS.

**Stage 1:**

The first step is installing react on your system using and creating new react applications using various tools . Basically this step is installing all the required environments for the project.

The first step is setting up an environment which is required for building an application. We start it by installing a platform i.e Webstorm or Visual Studio. For our project we required installing NodeJs and Angular.

**Stage 2:**

In this stage, we need to build components which take the User input and store them in a form which was used for building a good resume.

Resume Builder contains the following columns as different input fields requested from the user. All these columns together form a data set that is required to build a resume of the user.

**Table showing basic user inputs and their data type**

| Input | Data type of the input | User Reply |
| --- | --- | --- |
| Image | .jp or .jpeg |  |
| First Name | text |  |
| Middle Name | text |  |
| Last Name | text |  |
| About (< 50 words) | text |  |
| Date of Birth | date |  |
| Address | text |  |
| Email Id | text |  |
| Contact | number |  |
| Highest Degree | text |  |
| Date of Highest Degree | date |  |
| Score of Highest Degree | number |  |
| Experience | number |  |
| Technical Skills | Text |  |
| Certifications(if any) | Text or URL |  |
| Previous Worked Company | text |  |
| Publications (if any) | Text or URL |  |
| Personal Blog | Text or URL |  |
| Social Media | Text or URL |  |
| Awards | Text |  |
| Leadership Skills | Text |  |

**Stage 3:**  
Here, the output of the stage 2 is styled by using Styling components like Bootstrap and CSS components.

**Stage 4:**

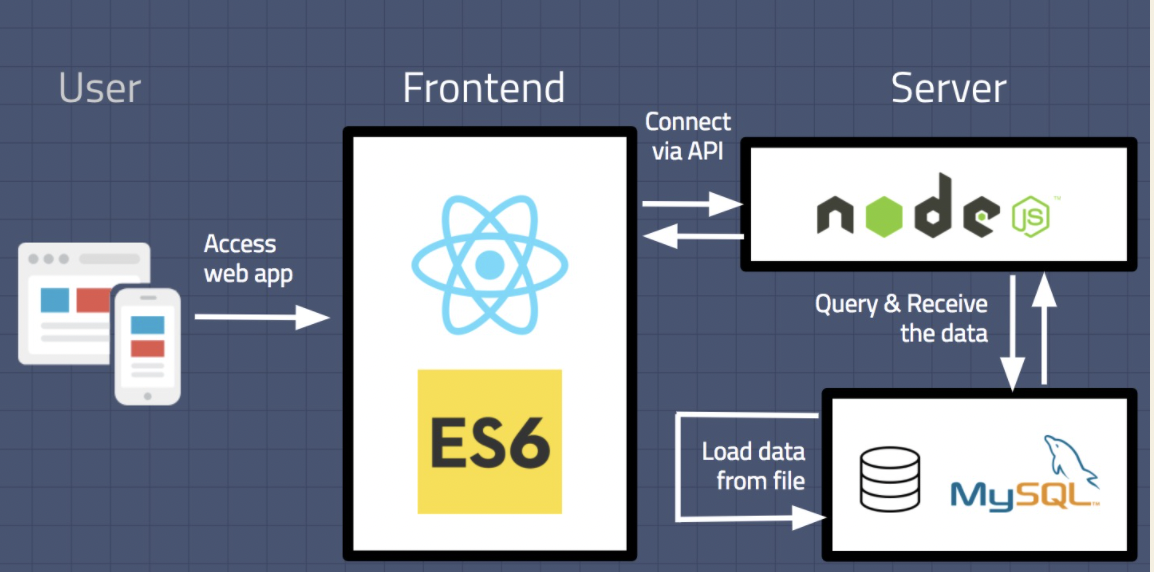
The styled components are passed to the server and stored in an user selected template.

**Stage 5:**

In this stage, the resume template is available in PDF format which can be easily downloaded by the user by clicking on a button.

**Database Design :**

Here we are using the MySQL store for all the user information and perform the data loading . Below picture shows the basic project architecture .

****

**Application working behind the scene :**

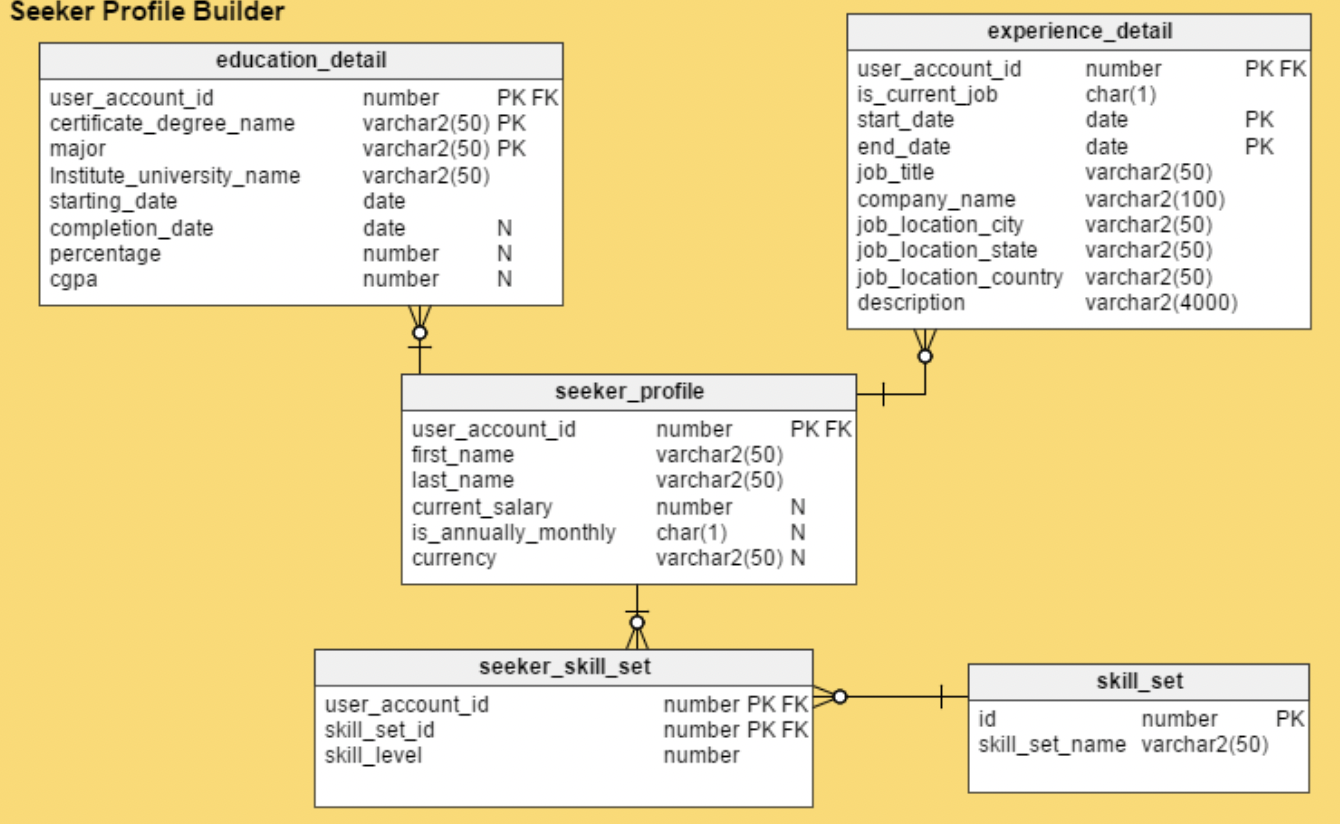
1. Users will visit the front end website built by react.
2. When user enters the data , the frontend will connect with the API

endpoints generated by NodeJS and stores the data in MySQL.

1. NodeJS will query the MySQL database and return the results to the frontend.
2. MySQL will load the data from the user input data into the database on a regular basis.
3. The result of the frontend is a well designed resume which will be available in PDF format and it can be downloaded on a single click.

**Data and its Flow :**

* Here the data is Users Input. Users input is taken as raw data and this data will be formatted to give a resume that can be used to apply to various jobs.
* The data is completely about the user and the user enters the data in the provided HTML input areas.
* This project doesn’t need any sampling as the user gives only the requested data.
* This data will be fresh for every user and this application will not be working with any global or previous data.
* This data doesn’t contain geographical information or any other factors like regional, national or global as the data is user specific.

****

**Figure displaying the provisional database design with examples**