COMSM0104: Web Technologies 2019
Final Assignment Report

Tao Xu

si19010@bristol.ac.uk

Yinan Yang

ff19085@bristol.ac.uk

June 6, 2020

Abstract

Our two-person team consists of Tao Xu (si19010) and Yinan Yang (ff19085). Due to environmental influences, we used a remote collaboration model via GitHub to co-develop this project.

We have created a website that generates resumes. Our website is based on Vue's front-end technical architecture, taking advantage of Vue's MVVM, the Model-View-ViewModel, we have done it to componentize the front-end development.

***Keywords:** Vue; SQLite.*

1 Introduction

We have created a website that generates resumes called Simple Resume Maker. The website provides basic user registration and login functionality. Once logged in, the user can edit the resume template provided on the website on the web page and download a .pdf version of the resume.

We try to simulate the user experience of editing documents online so that what the user sees is what they get. What would otherwise be a cumbersome formatting process is made easier with different CSS. This addresses the initial point we made in designing the product, which was to make things easier.

In building this site, we used the VUE framework, which is a progressive framework for building user interfaces.

> Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.
> —— Official development documentation from Vue

1.1 Project setup

npm install

1.2 Compiles and hot-reloads for development

npm run serve

1.3 Compiles and minifies for production

npm run build

1.4 Lints and fixes files

npm run lint

1.5 Start the server

npm start

1.6 Start the server

See Configuration Reference.

2 Self Evalutation

2.1 Estimation of marks

- A for HTML

- A for CSS

- A for JS

- A for PNG

- A for SVG

- A for Server

- A for Database

- A for Dynamic pages

2.2 Client Side

2.2.1 HTML

2.2.2 CSS

2.2.3 JS

2.2.4 PNG

2.2.5 SVG

In the use of SVG, we have used a variety of ways to construct SVG images to take full advantage of his benefits. We even created SVG animation on the home page. We will describe this in detail below.

- Basic SVG images

  We used tools like Inkscape to draw simple SVG portraits, and since the team has the skills to operate adobe Kit experience, so we are light on SVG production.
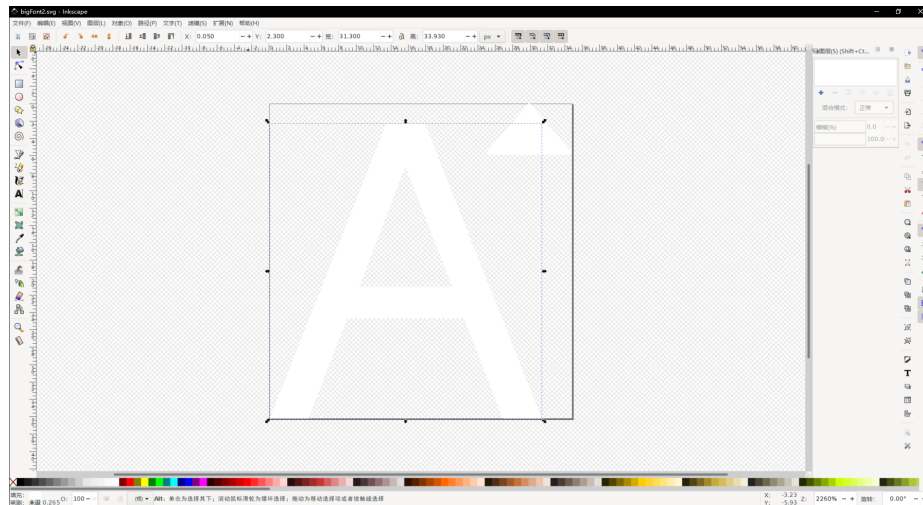

Figure 1: Inkscape

We used this basic graphical drawing to create 12 buttons, 3 of which are embedded in the page, while the remaining nine are used as Individual components are independent of the elements in the $src/components/button$. We take advantage of the object-oriented component design of the Vue components so that each module is easy to maintain and update later.

Figure 2: buttonBar

Figure 3: smallFontButton

- SVG-based css animation

  We were not satisfied with making basic SVG graphics. We created four SVG an-

imations with CSS animation effects. They are the start button on the home page, the continue and new buttons on the user-profile page, and the download button inside CVMaker. The most complex one is the download button, which activates the animation by changing the button's class when clicked.
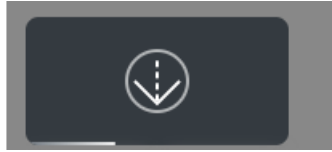

Figure 4: downloadButton animation

The download animation is divided into four parts, the first is the flashing of the outer ring, the second is the downward movement of the vertical line in the middle, and the third is the download of The middle arrow pattern becomes a checkmark when finished, and the fourth is the download progress bar at the bottom.


```
90    .button.downloading circle{
91        animation: 1.5s linear blink infinite;
92    }
93
94    .button.downloaded .arrow-top{
95        animation: 1s linear arrowTransform forwards;
96    }
97
98    .button.downloaded .checkmark{
99        opacity: 1;
100       stroke-dasharray: 100 100;
101       stroke-dashoffset: 100;
102       animation: 1s linear checkmarkTransform forwards 0.5s;
103   }
104
105   .button.downloaded .middle-line{
106       transition: 0.3s linear;
107       opacity: 0;
108   }
```

Figure 5: downloadButton animation

- svg animation based on vue-lottie
  Of course, doing this will not satisfy our ambition to try the coolest animations. So we introduced the vue-lottie open-source package, which is based on the lottie. Vue-lottie project vue architecture lottie can be interpreted as an SVG animation interpreter, and he supports the use of SVG in adobe After Effects exports complex animations to a JSON file and then self-rendering through the front-end of the web page to get cool effects.

  We've made a dynamic animation on the home page to highlight our theme, which we're sure you've seen.We save the exported animation JSON file that we send to AE in the $src/view/index/assets/animation$ folder.
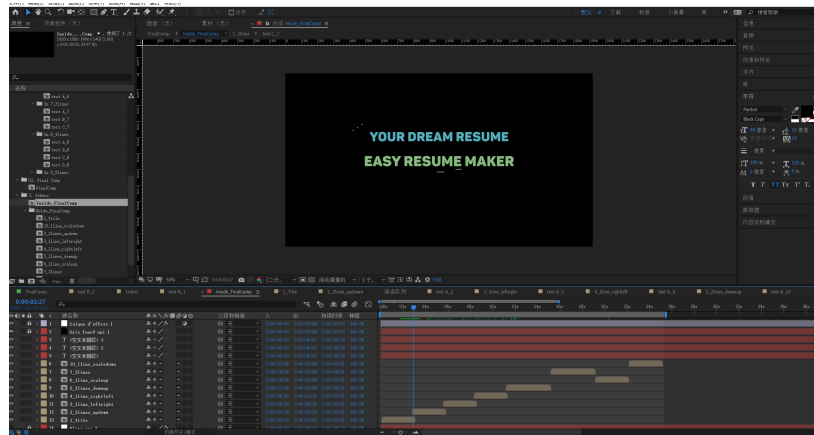
Figure 6: After Effect

Since this lottie tool is so new, we think we have made it pretty far ahead of the curve in terms of SVG usage.

2.3 Server Side

2.3.1 Server

2.3.2 Database

2.3.3 Dynamic pages

3 Working practices of the group