

COMSM0104-Web Technologies Report

Kehan Du, Shunyi Zhao

May 27, 2020

1 Introduction

1.1 Project Introduction

Our team is comprised of Kehan Du (mz19460) and Shunyi Zhao (vt19049). The aim of our project is to construct the PC end official website of Deep KW Innovation Factory.

Deep KW Innovation Factory is the academic workshop of interdisciplinary at Huazhong University of Science and Technology, China. Since the establishment of Deep KW Innovation Factory in 2017, it has always upheld and practiced the concepts and methods of interdisciplinary, and insisted on exploring the question of social sciences by using natural science methods.

The central goal of this version is to construct of the official website framework and the dynamic realization of key pages and complete the initial launch of the official website. This version is just used to practice the skills of web technologies, and will not be used commercially. We have got the permission to develop this website.

Because of the dangerous circumstance, we use GitHub to work together.

1.2 Development Method

The following operations will instantly start the project:

- git clone <https://github.com/lele1307/DeepKW.git>
- cd Server
- npm install
- npm run start
- cd project
- npm install
- npm run dev

2 Page Design

The home page of this project is shown below:



Figure 1: Home page

3 Technology Evaluation

3.1 Estimation of marks

- A for HTML
- A for CSS
- A for JS
- A for PNG
- B for SVG
- B for Server
- B for Database
- B for dynamic pages
- 12 for Depth (out of 20)

3.2 Client Side

We chose a popular javascript framework, Vue.js (version 2.6.11), to develop these pages. Modularity and componentization are important features of Vue, allowing us to build complex applications with small, independent, and usually reusable components. And these components convenient for later maintenance and upgrade of the official website. We use webpack and npm manage the packages based on node.js, and information about these packages is shown in package.json. Works are placed in folder named `./src/components`. HTML, used as the template of every module, and CSS will control the layout and style of these modules.

3.2.1 HTML

HTML is an integral part of the Vue component demonstrated as `<template>` tag. Vue.js uses an HTML-based template syntax that allows programmers to declaratively bind the rendered DOM to the underlying Vue instance's data. The advantage of Vue is it compiles the templates into Virtual DOM render functions. Combined with the reactivity system, Vue is able to intelligently figure out the minimal number of components to re-render and apply the minimal amount of DOM manipulations when the app state changes.

In our project, only one index.html document is used as the entrance, which is accessed through index.js. Other pages are all broken down into components, which are stitched together like building blocks. Compared with basic .html document, Vue's template HTML reduces the replication of many HTML statements. At the same time, the grammar in the Vue framework makes the data no longer nested in HTML statements, making the project itself more flexible.

In this project, BaseNav.vue `project/src/components/Common/BaseNav.vue` is a good example. This is a common Navigation component, which uses list rendering to render the navigation name(data) into HTML, and completes the task with a brief HTML statement.

```

<template>
<div id="wholeNavi">
  <div id="dwLogo">
    
  </div>
  <ul class="nav-left-container">
    <li v-for="(item,index) in fatherList" :key="index" @click="getIndex($event,item.name)">
      <router-link :to="item.router">{{item.name}}</router-link>
      <ul class="nav-left-container-small" :class="item.name">
        <li v-for="(item,i) in list" @click="getIndex($event,item.name)">
          <router-link :to="item.router">{{item.name}}</router-link>
        </li>
      </ul>
    </li>
  </ul>
</div>
</template>

```

Figure 2: The HTML codes written in BaseNav.vue

3.2.2 CSS

CSS is also an integral part of the Vue component demonstrated as `<style>`. The advantage is that the scope of CSS can be limited by adding the "scoped" attribute.

Vue supports a variety of CSS, Less, Sass style expression mode, this project uses basic CSS for style editing. At the same time, we also carried out basic beautification of some HTML parts with the help of Bootstrap4. The use of Flexbox in the layout, this method effectively improves the website screen adaptability.

Original CSS can be reflected in groups.vue *project/src/components/About/groups.vue*. Finally achieving the predetermined goal of the design diagram.

3.2.3 JavaScript

The embodiment of Javascript on the client-side is the Vue.js framework. From the structural point of view, index.js is used as the entrance into the program, router.js is the route deployment of the subpage of the website, and other specific js methods are written in the `<script>` tag of each component.

There are some examples of specific JS methods:

- The pop-up effect of the About-Group page
- Hide and show the title of the navigation bar subset
- News / Works pages read data from the database in the light of requirements

In addition, we used the Webpack packaging tool to modularize and unify a large number of CSS, JS and other dependent files. To a certain extent, the page loading capacity has been optimized.

The entire Vue project does not rely on Vue CLI scaffolding, and the project configuration files are all original.

3.2.4 PNG

Some images used in this project are drawn with GIMP 2. One of the examples, textitable.png which is used in *Contact.vue*, is shown below in Figure 4.

Some background images are cited from internet.

Firstly, we added three text objects into a 300×400 file, and then typed in some useful information. Then the coverages of these text objects are combined to fix the positions of them. The result of this stage is shown in the left image of Figure 4. And then, three rectangular layers are added, the result is shown in the right image of Figure 4.

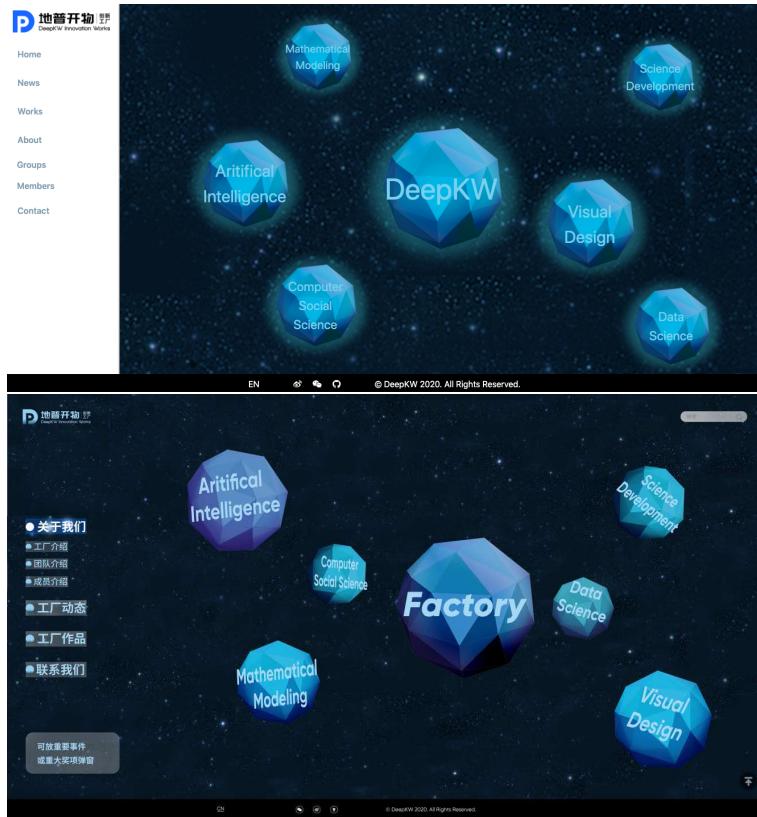


Figure 3: Design sketches and final layout



Figure 4: Left: the image after adding three text objects. Right: add three rectangles with 48.0 opaqueness.

3.2.5 SVG

We tried to use Inkscape as a tool for original SVG design.

- The arrow shape mainly reflected in Contact page *project/src/assets/img/Contact/arrow.svg*.
- LOGO of <title> tag is converted to ico format via svg.

In addition to the original SVG, the project also references the font-awsome icon font library. This font library can provide scalable vector icons. Its scalability can better adapt to the page size.

3.3 Sever Side

3.3.1 Server

The development of Sever is also in node.js. We use *express* as the server framework. The main works of Server is placed in the path (Server/routes), including three .js file. The file submit.js is used to accept the information

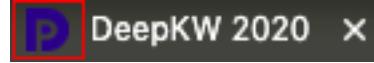


Figure 5: Logo of title



Figure 6: Svg examples

sent by the webpage Contact. This file use the function called `get()` to accept and send the response to the webpage. With the prepared statement of sql, this file could use a query to store these information into the database. Because the primary key of the table used in this file is not set to a auto-increment attribute, this file will also use a select query to count the number of rows in this table, to calculate the number which should be inserted.

The port number used in the sever side is set to 3000. Because some rules ban the direct communication between the client side and the server side, we use a feature of axios to send the information. This proxy is set in the file `./project/webpack.config.js`.

3.3.2 Database

We choose `sqlite3` to manage our databases. We use three tables, including candidata, news and work in this project.

The table candidata is used to store the data submitted by users. This table has six attributes, including id (primary key), name, gender, country, university and info. The table work has five attributes, including id, type, subject, title and content.

All queries used by the server files is written in prepared statements. We use these queries to get information from database and insert a new row into it. We also use some callback functions to ensure the programmes get the information in order.

3.3.3 Dynamic Page

A good example of dynamic page in our project is the *Works* page. We develop a method called `getAll()` in the file `project/src/components/Works/Selecter.vue`. This method is also used in the created part of Vue, which could get the data of all the works stored in databases when this webpage is initialized. The layout of this page is shown below in Figure 7.

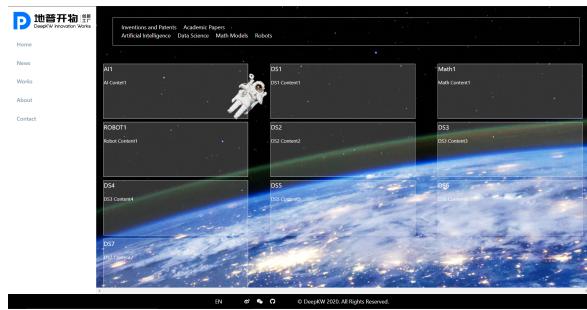


Figure 7: The layout of Works page when open it

Users could choose a type and a subject in the selection region in this page. If a user choose more than one option, only the first type and the first subject will be used to get information from server. If an option is chosen, the background of this option bacome white. The works of academic papers about data science is shown in Figure 8.

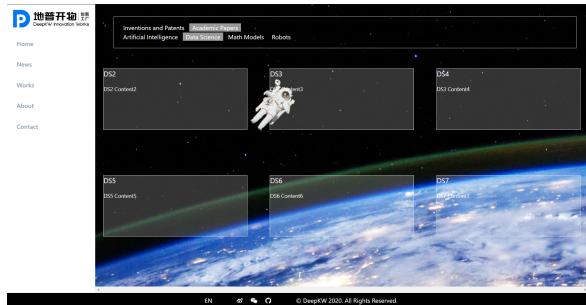


Figure 8: Works of academic papers about data science

A scroll bar is used to show more works.

4 Evaluation

4.1 Deficiencies

The construction of the website is basically completed, including the communication between the client and the server, and the configuration of the database. However, some styles and functions on the sub-webpage are still unfinished. For example, the About-Member page is just the basic layout, with simple JS interaction methods, but the formal style has not been completed yet and needs further improvement.

4.2 Teamwork

Our two-person team mainly uses Git for version control and GitHub for code merging and storage. Only in the circumstance of online communication, this tool greatly improves work efficiency. In the early stage of development, we agreed to use the Vue.js framework. At the same time, we have been learning by ourselves and helping each other. Although the final product still has a gap to launching, the first Web development experience is still impressive for us.