

2025년 상반기 K-디지털 트레이닝

# 프론트엔드 준비

[KB] IT's Your Life



#### ☑ 프로젝트 만들기

```
09 Vue+Spring₩scoula> npm init vue frontend
Vue.js – The Progressive JavaScript Framework
√ Add TypeScript? ... No / Yes
√ Add JSX Support? ... No / Yes
\checkmark Add Vue Router for Single Page Application development? ... No / Yes
√ Add Pinia for state management? ... No / Yes
√ Add Vitest for Unit Testing? ... No / Yes

√ Add an End-to-End Testing Solution? » No
√ Add ESLint for code quality? ... No / Yes
大〇己
Scaffolding project in ...09 Vue+Spring\scoula\forall frontend...
Done. Now run:
 cd frontend
 npm install
 npm run dev
```

#### ☑ 기본 라이브러리 추가

- o npm install
- o npm install bootstrap@5 axios moment

#### ☑ index.html (fontawesom CDN 추가)

```
<!DOCTYPE html>
<html lang="ko">
 <head>
  <meta charset="UTF-8" />
   <link rel="icon" href="/favicon.ico" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   link
    rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.6.0/css/all.min.css"
    integrity="sha512-
Kc323vGBEqzTmouAECnVceyQqyqdsSiqLQISBL29aUW4U/M7pSPA/gEUZQqv1cwx4OnYxTxve5UMg5GT6L4JJg=="
    crossorigin="anonymous"
    referrerpolicy="no-referrer"
  />
   <title>Scoula App</title>
 </head>
 <body>
   <div id="app"></div>
  <script type="module" src="/src/main.js"> </script>
 </body>
</html>
```

ERF STC

✓ Image: Assets

> IFF components

#### 프론트엔드 준비

### assets/main.css

 $@import\ url('https://fonts.googleap is.com/css2?family=Noto+Sans+KR:wght@100;200;300;400;500;600;700;800;900\&display=swap');$ 

```
* {
   font-family: 'Noto Sans KR';
}
```

#### O webfont 설정

■ <a href="https://noonnu.cc/">https://noonnu.cc/</a> 에서 noto 검색



### main.js

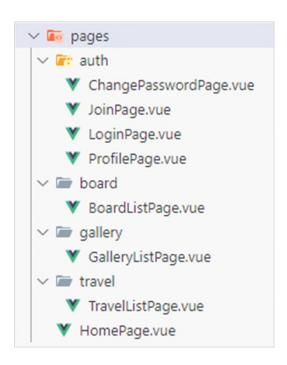
```
import ',/assets/main.css';
import 'bootstrap/dist/css/bootstrap.css';
import { createApp } from 'vue';
import { createPinia } from 'pinia';
import App from ',/App.vue';
import router from ',/router';

const app = createApp(App);
app.use(createPinia());
app.use(router);

app.mount('#app');
```

### App.vue

#### ☑ 기본 페이지 컴포넌트



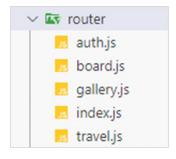
#### ○ 기존 views 폴더는 삭제

### pages/HomePage.vue

```
<script setup>
</script>
<template>
<h1>첫 번째 페이지</h1>
</template>
```

#### ○ 나머지 모든 페이지 → 타이틀만 구성

#### 🕜 라우팅 준비



#### ○ 기능별로 라우팅 설정을 모듈화

■ index: 기본 라우팅

auth: login, logout, join 라우팅

■ board: 게시글 관련 라우팅

■ travel: 추천 여행지 관련 라우팅

■ gallery: 갤러리 관련 라우팅

### router/auth.js

```
export default [
   path: '/auth/login',
   name: 'login',
   component: () => import('../pages/auth/LoginPage.vue'),
   path: '/auth/join',
   name: 'join',
   component: () => import('../pages/auth/JoinPage.vue'),
   path: '/auth/profile',
   name: 'profile',
   component: () => import('../pages/auth/ProfilePage.vue'),
   path: '/auth/changepassword',
   name: 'changepassword',
   component: () => import('../pages/auth/ChangePasswordPage.vue'),
```

### router/board.js

### router/travel.js

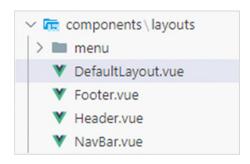
### router/gallery.js

### router/index.js

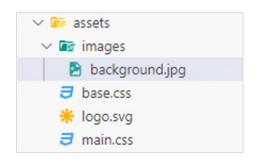
```
import { createRouter, createWebHistory } from 'vue-router';
import HomePage from '../pages/HomePage.vue';
import authRotes from './auth';
import boardRotes from './board';
import travelRoutes from './travel';
import galleryRoutes from './gallery';
const router = createRouter({
 history: createWebHistory(import.meta.env.BASE_URL),
 routes: [
   { path: '/', name: 'home', component: HomePage },
   ...authRotes.
   ...boardRotes,
   ...travelRoutes,
   ...galleryRoutes,
});
export default router;
```

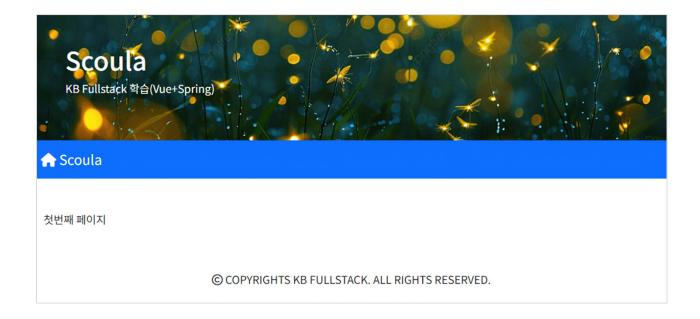
#### ○ 기본 레이아웃

- 기존 components/의 내용 모두 삭제
- 다음 구조로 컴포넌트 작성



#### ○ 헤더의 배경 이미지 준비





### **config/index.js**

```
export default {
                   // 메인 타이틀
 title: 'Scoula',
 subtitle: 'KB Fullstack 학습(Vue+Spring)', // 서브 타이틀
 menus: [ // 메인 메뉴 구성 정보
    title: '게시판',
    url: '/board/list',
    icon: 'fa-solid fa-paste',
    title: '여행',
    url: '/travel/list',
    icon: 'fa-solid fa-plane-departure',
    title: '갤러리',
    url: '/gallery/list',
    icon: 'fa-regular fa-images',
   },
```

### config/index.js

```
accoutMenus: { // 인증 관련 메뉴 정보
login: {
    url: '/auth/login',
    title: '로그인',
    icon: 'fa-solid fa-right-to-bracket',
    },

join: {
    url: '/auth/join',
    title: '희원가입',
    icon: 'fa-solid fa-user-plus',
    },
    },
},
```

### components/layouts/Header.vue

```
<script setup>
import config from '@/config';
</script>
<template>
 <div class="jumbotron p-5 bg-primary text-white">
  <h1>{{ config.title }}</h1>
  {{ config.subtitle }}
 </div>
</template>
<style scoped>
.jumbotron {
 background-image: url('@/assets/images/background.jpg');
 background-repeat: no-repeat;
 background-position: center;
 background-size: cover;
 color: white;
 padding: 2rem;
</style>
```

### components/layouts/NavBar.vue

### components/layouts/NavBar.vue

```
<template>
 <nav class="navbar navbar-expand-sm bg-primary navbar-dark">
  <div class="container-fluid">
    <router-link class="navbar-brand" to="/">
     <i class="fa-solid fa-house"></i>
     Scoula
    </router-link>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#collapsibleNavbar"
          @click="toggleNavShow">
     <span class="navbar-toggler-icon"></span>
    </button>
    <div :class="navClass" id="collapsibleNavbar">
       <!-- 추후 작업 예정 -->
    </div>
  </div>
 </nav>
</template>
<style></style>
```

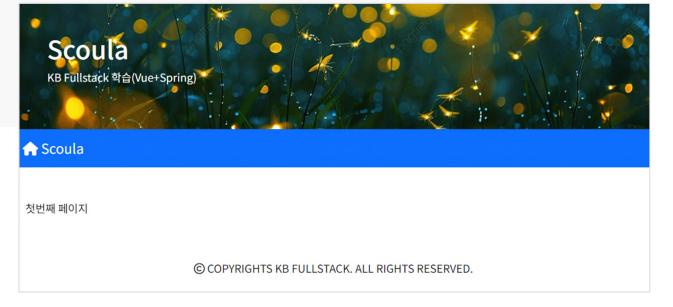
### components/layouts/Footer.vue

```
<script> </script>
<template>
  <div class="my-5 p-3 text-center">
        <i class="fa-regular fa-copyright"> </i>
        COPYRIGHTS KB FULLSTACK. ALL RIGHTS RESERVED.
        </div>
    </template>
```

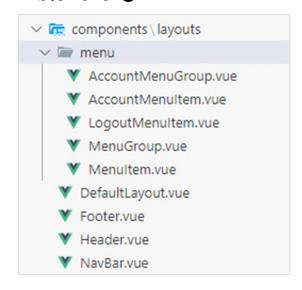
### components/layouts/DefaultLayout.vue

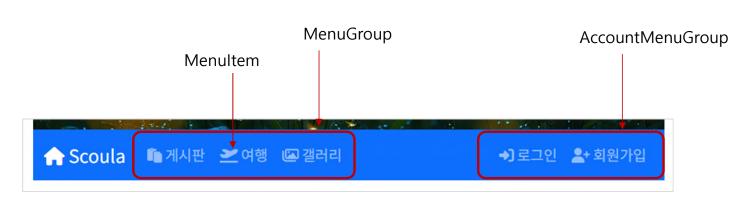
```
<script setup>
import Header from './Header.vue';
import NavBar from './NavBar.vue';
import Footer from './Footer.vue';
</script>
<template>
 <div class="container">
   <Header />
  <NavBar />
   <div class="content my-5 px-3">
    <slot></slot>
  </div>
  <Footer />
 </div>
</template>
```

### App.vue



#### ☑ 메뉴의 구성







### components/layouts/menu/MenuItem.vue

```
<script setup>
const props = defineProps({
  menu: { Type: Object, required: true },
});

</script>

<template>
  class="nav-item">
        <router-link class="nav-link" :to="menu.url">
        <i :class="menu.icon"> </i>
        {{ menu.tite }}
        </router-link>

        </template>
        <style> </style>
```

### components/layouts/menu/MenuGroup.vue

```
<script setup>
import Menultem from '/Menultem.vue';

const props = defineProps({
    menus: { Type: Array, required: true },
});

</script>

<template>

    <Menultem v-for="menu in menus" :menu="menu" />

</template>
```

### components/layouts/menu/AccountMenuItem.vue

### components/layouts/menu/LogoutMenuItem.vue

```
<script setup>
import { useRouter } from 'vue-router';
const router = useRouter();
const logout = (e) => {
 // 로그아웃
 router.push('/');
};
</script>
<template>
 <a href="#" class="nav-link" @click.prevent="logout">
   <i class="fa-solid fa-right-from-bracket"></i>
  로그아웃
 </a>
</template>
```

### components/layouts/menu/AccountMenuGroup.vue

```
<script setup>
import { computed } from 'vue';
import Menultem from ',/Menultem.vue';
import AccountMenultem from ',/AccountMenultem.vue';
import LogoutMenultem from ',/LogoutMenultem.vue';
import config from '@/config';

const { login, join } = config.accoutMenus;

const islogin = computed(() => false); // 임시: 로그인하지 않음
const username = computed(() => ''); // 임시: 사용자명 없음
</script>
```

### components/layouts/menu/AccountMenuGroup.vue

```
<template>

<template v-if="islogin">
<template v-if="islogin">
<AccountMenuItem :username="username" />
<LogoutMenuItem />
</template>
<template>
<template v-else>
<MenuItem :menu="login" />
<MenuItem :menu="join" />
</template>
</template>
</template>
</template>
```

### components/layouts/NavBar.vue

```
<script setup>
import { reactive, computed } from 'vue';
import config from '@/config';
import MenuGroup from './menu/MenuGroup.vue';
import AccountMenuGroup from './menu/AccountMenuGroup.vue';
</script>
<template>
 <nav class="navbar navbar-expand-sm bg-primary navbar-dark">
   <div class="container-fluid">
    <div :class="navClass" id="collapsibleNavbar">
     <!-- 추후 작업 예정 -->
     <MenuGroup :menus="config.menus" />
     <AccountMenuGroup />
    </div>
   </div>
 </nav>
</template>
<style></style>
```

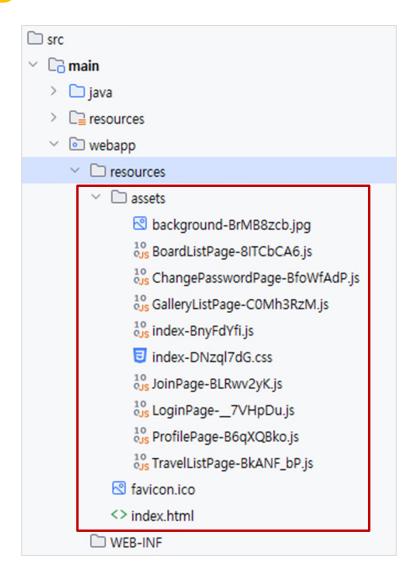
- api 연결을 위한 proxy 서버 설정 및 배포 설정
  - o 백엔드 서버로 요청을 포워딩할 proxy 설정
  - o npm run build 명령시 결과 생성 폴더를 백엔드의 resources 폴더로 지정

### vite.config.js

```
import { fileURLToPath, URL } from 'node:url';
export default defineConfig({
 plugins: [vue()],
 resolve: {
   alias: {
     '@': fileURLToPath(new URL('./src', import.meta.url)),
   },
 server: {
   proxy: {
     '/api': {
      target: 'http://localhost:8080',
 build: {
   outDir: 'C:/KB_Fullstack/09_Spring+Vue/scoula/backend/src/main/webapp/resources',
});
```

#### onpm run build

o backend의 resources 폴더에 결과물 생김



#### ☑ 백엔드 기동

#### http://localhost:8080/

