

47 페이지 마지막 줄 그림 설명 추가 -> 1.3.1 Vue 2 의 Optional API 애플리케이션 제작

탐색기	...
<ul style="list-style-type: none"> CH01 <ul style="list-style-type: none"> ch01_myapp1 <ul style="list-style-type: none"> node_modules public <ul style="list-style-type: none"> assets <ul style="list-style-type: none"> code.jpg react.jpg restful.jpg springboot.jpg tensor.jpg favicon.ico index.html src <ul style="list-style-type: none"> assets <ul style="list-style-type: none"> books.js logo.png components <ul style="list-style-type: none"> BookDetail.vue ServiceList.vue App.vue main.js .eslintrc.js babel.config.js jsconfig.json package-lock.json package.json README.md vue.config.js 	<p>프로젝트의 의존성 패키지들이 저장되는 디렉터리 (자동 생성) 정적 파일이 위치하는 디렉터리</p> <p>웹사이트 아이콘 메인 HTML 파일, Vue 앱이 이곳에 마운트됨</p> <p>소스코드가 위치하는 디렉터리 이미지, 폰트 등 정적 자원을 저장하는 디렉터리</p> <p>기본 제공되는 로고 이미지</p> <p>Vue 컴포넌트를 저장하는 디렉터리</p> <p>사용자가 작성한 Vue 컴포넌트</p> <p>루트 컴포넌트 (모든 Vue 컴포넌트는 이 컴포넌트 안에서 렌더링)</p> <p>Vue 인스턴스를 생성하고 App 컴포넌트를 마운트하는 엔트리 파일</p> <p>ESLint 설정 파일, 코드 스타일 및 규칙 설정</p> <p>Babel 설정 파일, 최신 자바스크립트 문법 변환에 사용 JavaScript 프로젝트에서 사용하는 설정 파일</p> <p>프로젝트의 의존성을 정확하게 관리하기 위한 파일 프로젝트의 의존성 및 스크립트를 정의하는 파일</p> <p>프로젝트 설명 및 사용법을 설명하는 파일</p> <p>Vue CLI 프로젝트에서 사용할 수 있는 프로젝트 설정 파일</p>

[그림 1-42] 프로젝트 파일 구조

146 페이지 15~17 번째 줄 명령 -> 3.5.1 프로젝트 제작

(2) 프로젝트 디렉터리로 이동 및 의존성 라이브러리 설치 -

tailwind CSS 프레임워크를 설치하고 tailwind.css 외부 스타일 파일을 초기화합니다.

```
D:\book\ch03>cd ch03_myapp1
D:\book\ch03\ch03_myapp1>npm install -D tailwindcss postcss autoprefixer
D:\book\ch03\ch03_myapp1>npx tailwindcss init
D:\book\ch03\ch03_myapp1>npm run serve
```

187 페이지 .eslint 내용 누락 -> 4.3.1 Vue 커스텀 이벤트 실습

(4) 프로젝트 애플리케이션 설정 src/main.js 코드 내용 아래

es6 문법 설정을 위한 .eslintrc 를 설정합니다.

.eslintrc

```
01 {
02   "extends": [
03     "plugin:vue/vue3-recommended"
04   ],
05   "rules": {
06     "vue/script-setup-uses-vars": "error"
07   }
08 }
```

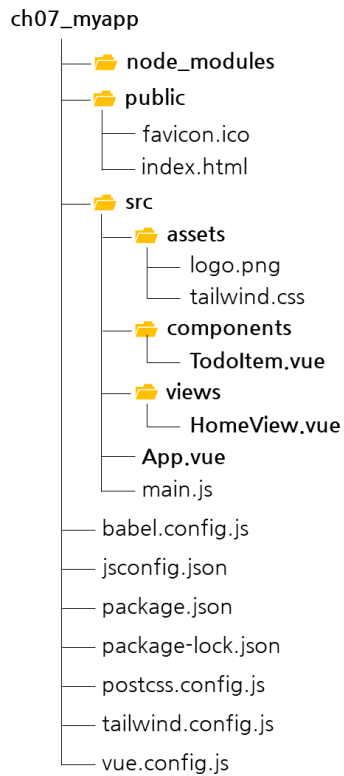
190 페이지 17 번째 줄 App.vue 내용 누락 -> (5) 컴포넌트 작성

최상위 컴포넌트인 App.vue 를 작성합니다.

src/App.vue

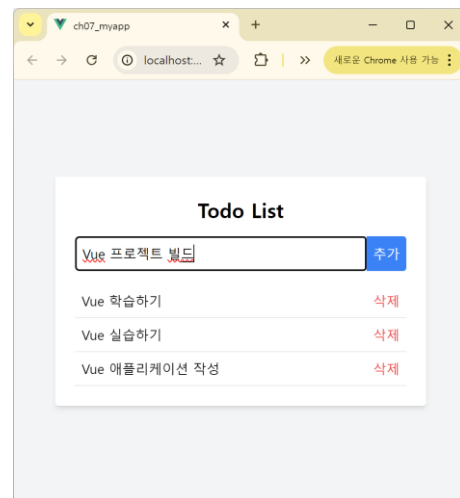
```
09 <template>
10   <div class="container">
11     <h1 class="center-align">Vue 3 Composition API Demo</h1>
12     <!-- Parent/Child Event with Custom Events -->
13     <ParentComponent />
14     <!-- SubParent/SubChild with v-model -->
15     <SubParentComponent />
16   </div>
17 </template>
18 <script setup>
19 import ParentComponent from './components/ParentComponent.vue';
20 import SubParentComponent from './components/SubParentComponent.vue';
21 </script>
22 <style>
23 .container {
24   margin-top: 50px;
25 }
26 </style>
```

369 페이지 그림 7-7 ch07_myapp 프로젝트 구조에서 일부 폴더명 잘못 표기 => HomeView.vue 의 위치가 assets 에서 views 로 변경



[그림 7-7] ch07_myapp 프로젝트 구조

HomeView.vue: TodoList 를 위한 메인 뷰입니다.
 TodoItem.vue: TodoList 에 각 항목이 출력되는 컴포넌트입니다.
 main.js: 애플리케이션의 진입 파일입니다.
 babel.config.js: 여러 버전의 Javascript 을 통일화하기 위한 Babel 설정 파일입니다.
 postcss.config.js: scss 나 less 와 같은 전처리기 여부를 설정합니다.
 tailwind.config.js: Tailwind 설정 파일입니다.



[그림 7-8] ch07_myapp 실행 화면

370 페이지 HomeView.vue 의 위치가 assets 에서 views 로 변경

HomeView.vue 는 TodoList 의 메인 뷰로서 여러 자식 컴포넌트를 포함할 수 있습니다.

src/views/HomeView.vue

```
01 <template>
02   <div>
03     <div class="flex mb-4">
04       <input v-model="newTodo" @keyup.enter="addTodo" type="text"
05         placeholder="새로운 Todo를 입력하세요" class="flex-grow p-2 border rounded-l" />
06       <button @click="addTodo" class="bg-blue-500 text-white p-2 rounded-r">
07         추가
08       </button>
09     </div>
10     <ul>
11       <TodoItem v-for="todo in todos" :key="todo.id" :todo="todo" @remove="removeTodo" />
12     </ul>
13   </div>
14 </template>
15 <script setup>
16 import { ref } from 'vue';
17 import TodoItem from '../components/TodoItem.vue';
18 const newTodo = ref('');
19 const todos = ref([]);
20 const addTodo = () => {
21   if (newTodo.value.trim() === '') return;
22   todos.value.push({ id: Date.now(), text: newTodo.value.trim(), });
23   newTodo.value = '';
24 };
25 const removeTodo = (id) => { todos.value = todos.value.filter((todo) => todo.id !== id); };
26 </script>
```

382 페이지 5 번째 줄 App.vue 내용 누락

(7) 애플리케이션 컴포넌트 작성

최상위 루트 컴포넌트인 App.vue 를 작성하여 라우터 내용을 출력합니다.

src/App.vue

```
01 <template>
02   <nav>
03     <router-link to="/">Home</router-link> |
04     <router-link to="/about">About</router-link>
05   </nav>
06   <router-view/>
07 </template>
08 <style>
09 #app {
10   font-family: Avenir, Helvetica, Arial, sans-serif;
11   -webkit-font-smoothing: antialiased;
12   -moz-osx-font-smoothing: grayscale;
13   text-align: center;
14   color: #2c3e50;
15 }
16 nav {
17   padding: 30px;
18 }
19 nav a, a {
20   font-weight: bold;
21   color: #2c3e50;
22   padding: 14px;
23 }
24 nav a.router-link-exact-active {
25   color: #42b983;
26 }
27 </style>
```

419 페이지 9 번째 줄의 코드인 <script setup> 아래에 import { defineProps } from 'vue';

이 교재에는 있으나 깃허브의 소스코드 ch08\ch08_myapp\src\views\UserProfile.vue 에는 없으니
깃허브에서 다운로드 받은 소스코드 ch08\ch08_myapp\src\views\UserProfile.vue 파일의 <script setup>
아래에 import { defineProps } from 'vue'; 를 추가해주세요