



Interface Design and Development

Pass Task 10.1: Creating a Single Page Application Using Vue CLI

Overview

- Purpose:** Learn how to create single page applications using Vue CLI.
- Task:** Use Vue CLI to create a single page application.
- Time:** This task should be completed in your lab class and submitted for feedback before the start of week 12.
- Resources:**
- Lecture notes

Submission Details

You must submit the following files:

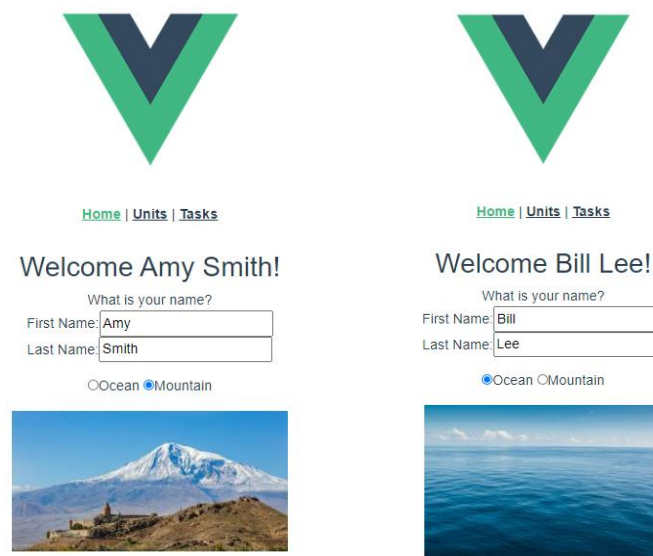
- Source code (html and/or javascript files).
- Screenshot of the webpage

Make sure that your task has the following in your submission:

- Code must be HTML5 standard and observe indentation.
- Your webpage uses the VueJS and Bootstrap framework appropriately.

Instructions

1. Install nodejs from <https://nodejs.org/en/download/>
2. Install Vue CLI, Create a new project named “my-study”. (Reference: lecture slides).
3. Run your project and view it from <http://localhost:8080/> (Reference: lecture slides).
4. Add router for your project. Run your project and make sure it is correct. (Reference: lecture slides).
5. Create three simple single file components Home.vue, Tasks.vue and Units.vue. Inside the components, you can just display a <h2> and a <p>. Modify the router created by Vue CLI to have three items: Home, Tasks and Units. Make them link to the Home, Tasks and Units components. (Reference: lecture slides). Test your app, make sure it is working.
6. Improve the Home.vue component:
 - Include two inputs for user’s first name and last name.
 - Display welcome message based on user’s name.
 - Show an image based on user’s selection on two radio buttons, such as mountain or ocean.



7. Improve the Units.vue component to display a list of units stored in a json file. Your Units.vue file may look like this:

```
<template>

    <!-- use a table to display all the units -->

</template>
```

```


<script>
import Units from "../assets/units.json";    // your json file is under assets
export default {
  name: 'Units',
  data: function(){
    return {units:Units}
  }
}
</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>

</style>

```

Your units page may look like this (you can use different styles):



[Home](#) | [Units](#) | [Tasks](#)

Units

| Code | Description | cp | Type |
|----------|---|------|----------------------|
| ICT10001 | Problem Solving with ICT | 12.5 | Core |
| COS10005 | Web Development | 12.5 | Core |
| INF10002 | Database Analysis and Design | 12.5 | Core |
| COS10009 | Introduction to Programming | 12.5 | Core |
| COS20001 | User-Centred Design | 12.5 | Software Development |
| TNE10005 | Network Administration | 12.5 | Software Development |
| COS20016 | Operating System Configuration | 12.5 | Software Development |
| INF20012 | Enterprise Systems | 12.5 | Systems Analysis |
| ACC10007 | Financial Information for Decision Making | 12.5 | Systems Analysis |
| INF20003 | Requirements Analysis and Modelling | 12.5 | Systems Analysis |
| ACC20014 | Management Decision Making | 12.5 | Systems Analysis |
| INF30001 | Systems Acquisition & Implementation Management | 12.5 | Systems Analysis |

8. Improve the Tasks.vue component: make it similar to 5.1 P (you can use your 5.1P component).
9. Test your app and **submit to Canvas** with screenshots.