

Interface Design and Development

Task 3.1: String Test Web App with VueJS

Overview

Control flow in VueJS enables you to easily add conditions and loops to dynamically generate your web application view. In this task, you will create a small web application that uses conditions to dynamically output custom messages to users.

Purpose: Install and test the framework needed to get started.

Task: Create a web app that tests a user's name and displays a custom message.

Resources:

- Lecture notes #3
- VueJS <https://vuejs.org/>

Instructions

The first task includes the steps needed for you to install the framework you will need in this unit. You will then use the VueJS framework (version 3.x) to create interactive web page.

1. Download and install the framework you need to get started. Ensure that you have:

- Installed VueJS (<https://vuejs.org/>)

You should have the following files stored in appropriate folders

```
framework/
├── css/
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map    ← map is used for debugging
│   ├── ... ..
│   └── ... ..
└── js/
    ├── vue.min.js               (or vue.js)
    ├── bootstrap.min.js
    └── ... ..
```

2. If you don't already have one, make a directory (i.e., a 'folder') to store your framework.
3. Create a file named stringtest.html in your lab03 directory.
4. Implement a web application with the following logic:
 - It reads a name from the user, and displays back a message.
 - Check if the name entered is your name.
 - If the name is your name, output the message 'Awesome name!'
 - Otherwise output the 'not my name' message.

Tip: Remember to use == as the conditional operator, = is an assignment operator.

Note: Conditional operator is case sensitive. For example, Caslon is not equal to caslon

Hint: Use the lowercase filter to convert the value of a variable into lowercase before you compare.

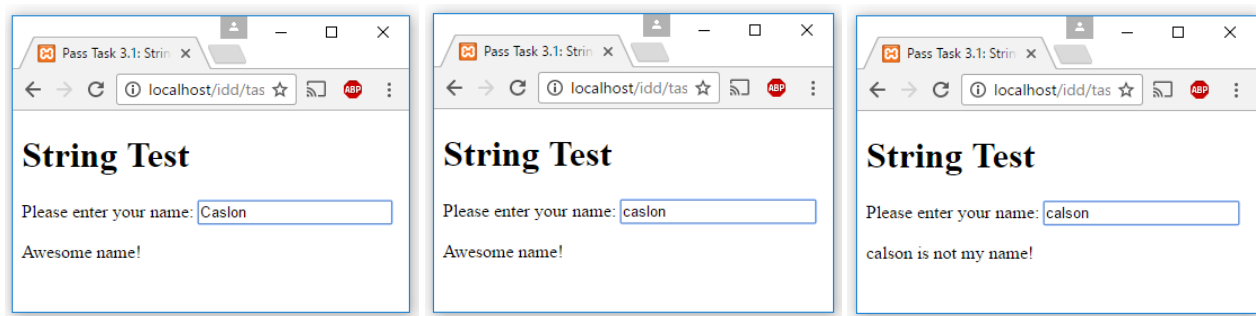


Figure 1: Screenshot of the web app with no Bootstrap mark up