sit 120 aSSEMENT 3: Developing a responsive web app

ASHLEY JAMES DOWIE

Contents

[Description 1](#_Toc83381695)

[Explanation 2](#_Toc83381696)

[Documentation and Basics 2](#_Toc83381697)

[Computed Properties and Watches 2](#_Toc83381698)

[Conditional Rendering 2](#_Toc83381699)

[List Rendering 3](#_Toc83381700)

[Event Handling 3](#_Toc83381701)

[Form Input Bindings 3](#_Toc83381702)

[Component Basics 3](#_Toc83381703)

[Components in Depth 4](#_Toc83381704)

[Component Registration 4](#_Toc83381705)

[Props 4](#_Toc83381706)

[Custom Events 4](#_Toc83381707)

[Transitions and Animations Reusability & Composition Tolling and Scaling Up 5](#_Toc83381708)

[Enter/Leave & List Transitions 5](#_Toc83381709)

[Single File Components 5](#_Toc83381710)

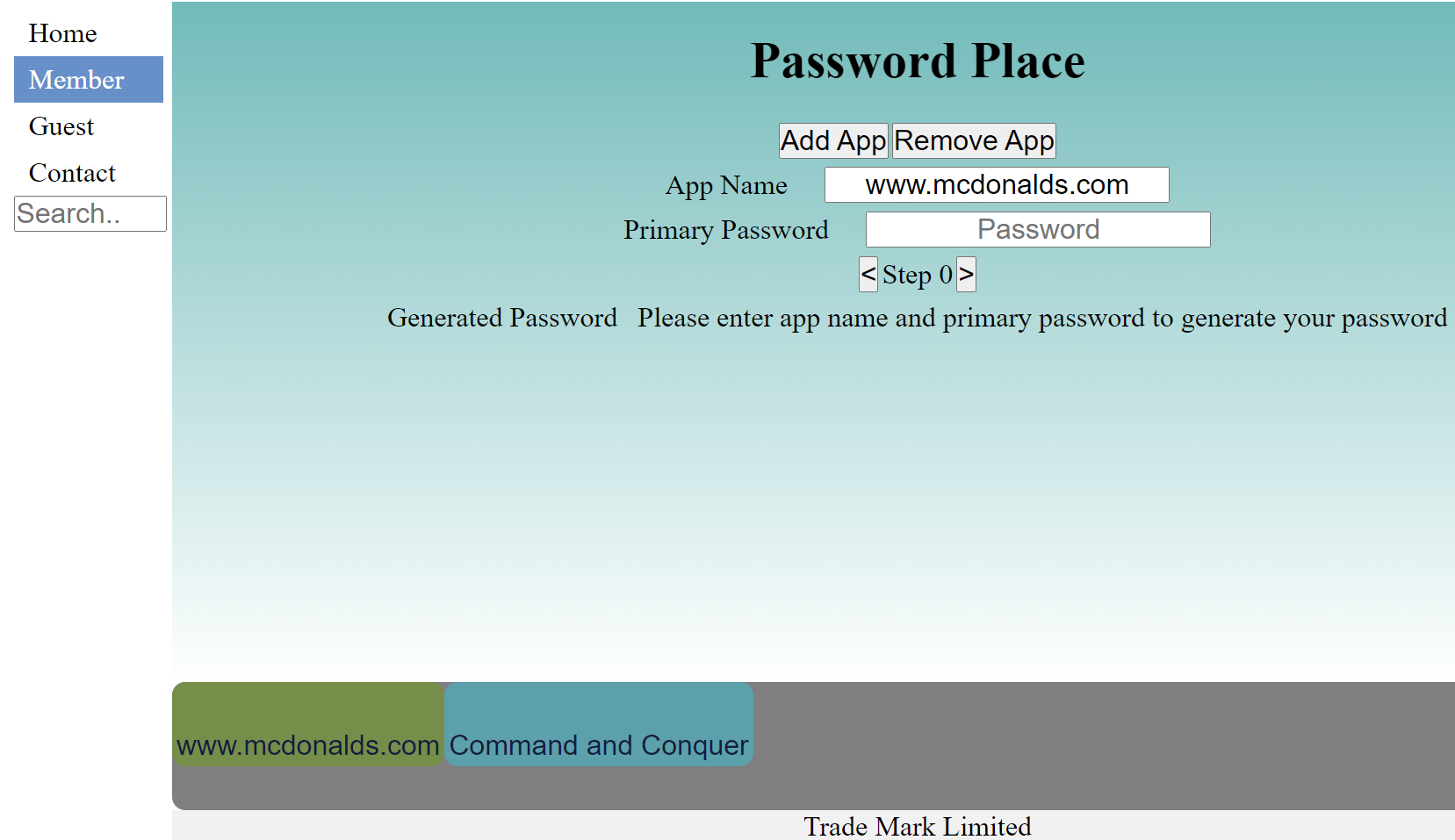
[Routing 5](#_Toc83381711)

# Description

Password Place is a password generation program. Password Place does not store passwords, but instead uses a deterministic function to generate passwords for different applications. Users simply need to provide the name of an application and a primary password. Password Place will then generate a password using a deterministic function that takes the primary password and application name as parameters.

If an application requires users to update their password, the User can adjust a third parameter called “Step”. Step is an integer value that can be incremented to handle password expiration. The step value can also be decremented to allow users to look through a history of passwords they have previously generated using Password Place.

Password Place offers a guest mode and a member mode. Guest mode is available to users who do not wish to create an account. Guest mode allows users to enter an application name and a primary password to generate a password for the given application. Member mode requires users to create an account and offers the same functionality as Guest mode but with additional features. Member mode allows users to use the Step feature to handle password expiration and allow users to cycle through a history of their passwords for the application. In member mode, users can add and remove applications from a list of applications that they wish to generate a password for. Generated passwords are still not stored in member mode, only the names of applications and the current step value for that application.



Selectable saved list of applications and associated step value.

Add / remove applications

Increment / decrement Step

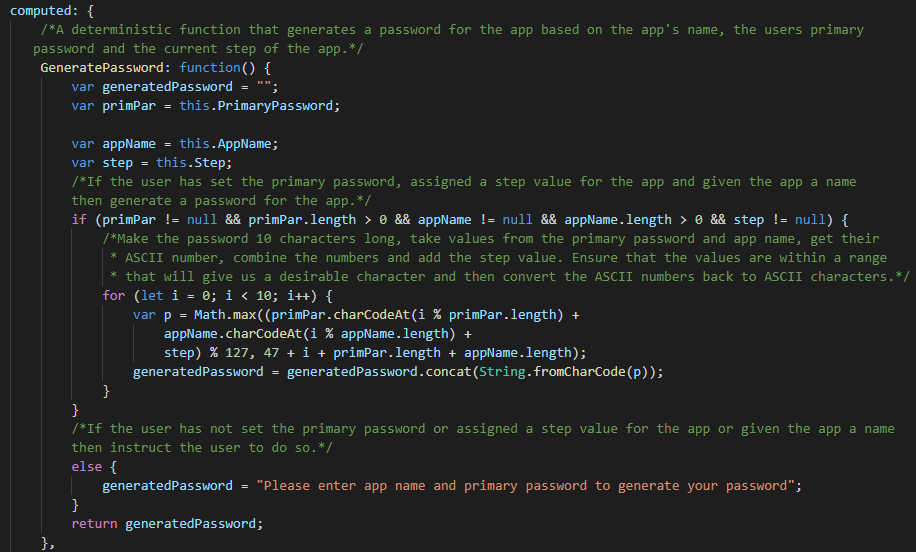
Enter app name and primary password

Generated password displayed here

# Explanation

## Documentation and Basics

### Computed Properties and Watches

Password Place generates a password using a deterministic function that uses application name, primary password, and step value as parameters. As the generated password is calculated based on other values, it made sense that the generated password would be a computed property that is part of the PasswordGenerator Vue component.   


### Conditional Rendering

Member mode and guest mode both present the user with a form that the user needs to fill out to generate a password. The only difference between the two forms is that in member mode there is the step adjustment section and the add and remove app buttons. It made sense to use the same MainForm Vue component for both modes and simply use v-if to only display member mode content if the user is in member mode.



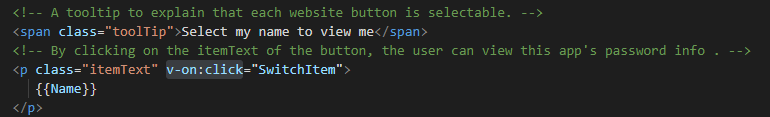
### List Rendering

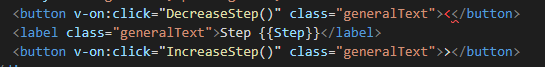
The member section provides a list of applications that the user can add to and remove from. Naturally the list is rendered in the application using v-for.



### Event Handling

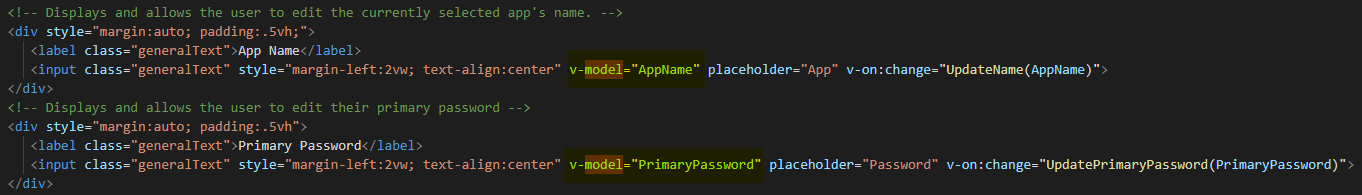
Users can change the currently selected application by clicking on an application from the list. Users are also able to increment / decrement the step value by clicking the < and > buttons. This is handled using v-on:click.





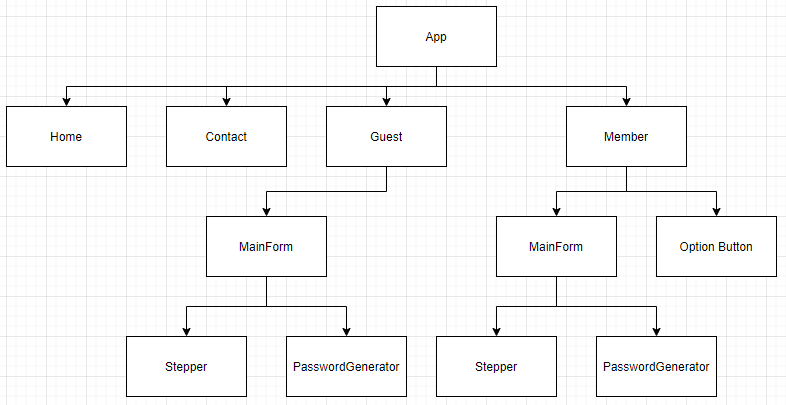
### Form Input Bindings

Users are required to enter an application name and primary password. The binding needed to be two way as the application name input element can also be updated when the user clicks on another application from the application list. As two-way binding was required, v-model was used to bind the input elements for primary password and application name.



### Component Basics

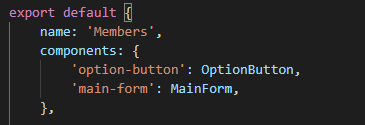
The application makes use of numerous components. The names of these components and their relationship to each other can be seen in the below component hierarchy diagram.



## Components in Depth

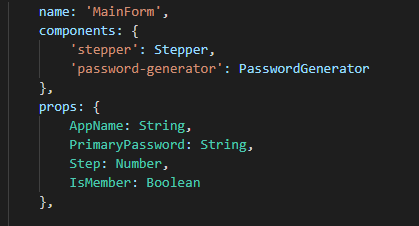
### Component Registration

All components are registered locally in this application. This will make the application easier to maintain if the application were to increase in scope in the future as a component dependency will be easier to determine. Below is an example of the member component locally declaring the OptionButton and MainForm components.



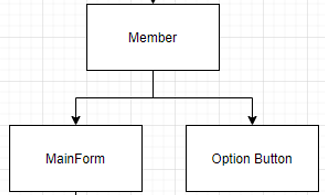
### Props

Props are used in the Mainform, OptionButton, Stepper and Password generator components. Below is an example of props used in the MainForm component.



### Custom Events

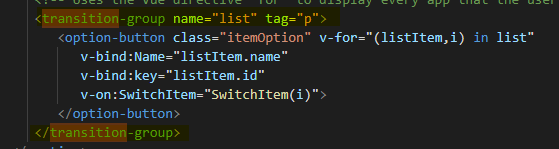
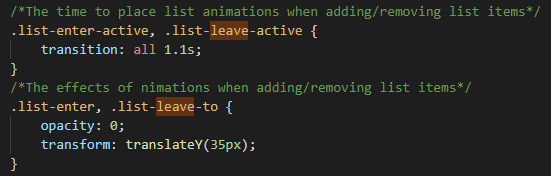
Custom events are used in this application to pass information from child components to their parent components. An example of how this is used is when the user selects an application name from the list, the main form needs to update to show the information of the currently selected app. The application achieves this by emitting a custom event from the OptionItem component to the Member component which updates the currentItem property, which is then passed as a prop to the MainForm component.



## Transitions and Animations Reusability & Composition Tolling and Scaling Up

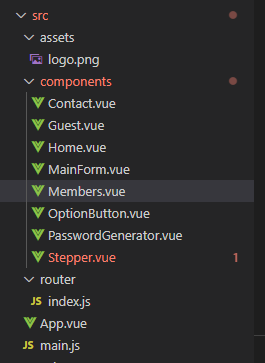
### Enter/Leave & List Transitions

List items have an animated transition that plays when they are removed / added to the list. Items travel upward and go from 0 to 100% opacity when added to the list. The opposite occurs when items are removed from the list.



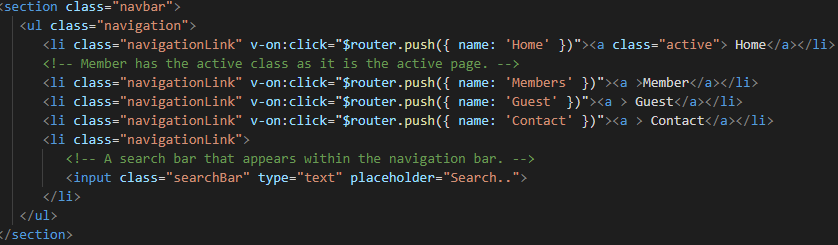
### Single File Components

Every component used within this application is contained within its own .vue file.



### Routing

Routing is used in this application to allow the user to travel between the four pages of the application. These pages are the member page, guest page, home page and contact page.



Please find zip file for application here: https://github.com/AshleyDowie/SIT120---Introduction-To-Responsive-Web-Apps/tree/main/Assignment%203