PWHA Horse Show Shows Using Vue

Vuetify, Material, Axios, Express, MySQL

Requirements

Create a program that will:

- Keep track of all the shows, riders, horses and rider/horse pairs
- Keep track of all the Divisions and Classes held in each show and all the entrants in each class
- Compute the winner of each Division, Show and Season using the recorded placing for each pairing in each class in each show

Quick Demo

Two node windows

MySQL

Quick CRUD explanation

Back End

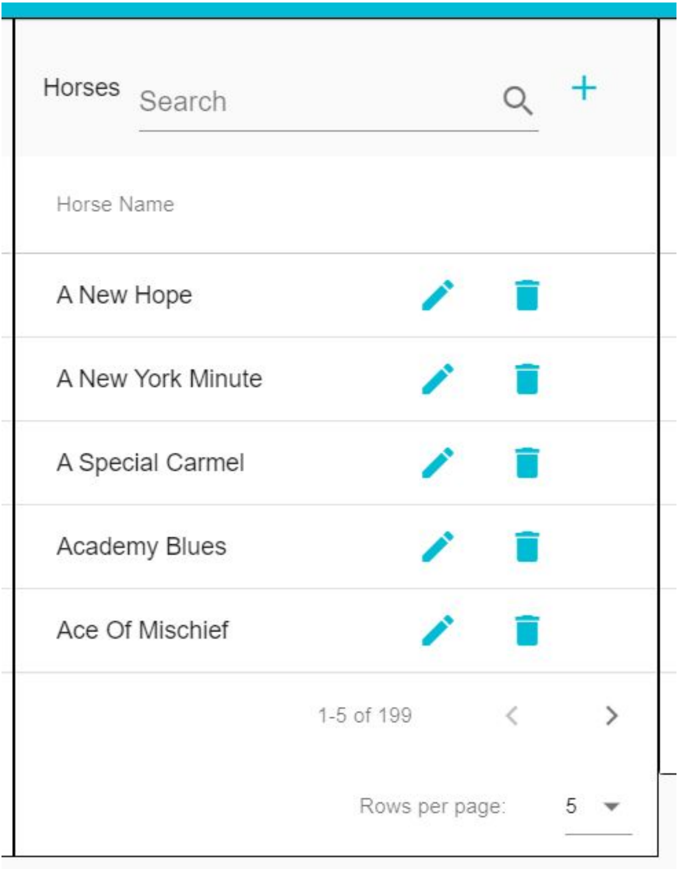
MySQL

Node with Express creating an API

```
var horses = require('./data_access/horses')
//express server
module.exports = {
 configure: function (app) {
   app.get('/getAllHorses',
      function (req, res) {
        horses.getAll(res)
   }) }  }
```

```
function Horses () {
 this.getAll = function (res) {
  connection.init()
  connection.acquire(function (err, con) {
  var query = 'select id, name from horses
order by name'
 con.query(query, function (err, result) {
  con.release()
  res.send(result)
     })   })  }
```

Front End -- Vuetify



Horses

Search

Horse Name

A New Hope

A New York Minute

A Special Carmel

Academy Blues

Ace Of Mischief

1-5 of 199

Rows per page: 5

```
<div class="col2">
 <v-card-title>Horses<v-spacer></v-spacer>
  <v-text-field append-icon="search" label="Search" single-line hide-details
          v-model="horseSearch"></v-text-field>
   <dialogHorseNew @horseNew="onHorseNew"></dialogHorseNew>
 </v-card-title>
 <v-data-table :headers="horsesHeaders" :items="horses" :search="horseSearch">
  <template slot="items" slot-scope="props">
   <td class="text-xs-left largerFont">{{ props.item.name }}</td>
   <td class="text-xs-right sm3 flex">
     <dialogHorseEdit :item="props.item" :index="props.item.id"
         @horseEdited="onHorseUpdated"></dialogHorseEdit>
     <dialogHorseDelete :message="{'message': 'Are you sure?',
         'item': props.item}" :index="props.item.id"
          @horseDeleted="onHorseDeleted"></dialogHorseDelete>
    </td>
 </template>
</v-data-table></div>
```

Getting/Using Data

Services: `import HorsesService from '@/services/HorsesService'`

```
data () {
  return {
    horseSearch: '',
    horses: [],
}
async mounted () {
  this.riders = (await RidersService.getAllRiders()).data
  this.horses = (await HorsesService.getAllHorses()).data
  this.pairs = (await PairsService.getAllPairs()).data
}
```

```
methods: {
 async onHorseNew () {
  this.horseSearch = ''
  this.horses = (await HorsesService.getAllHorses()).data
  this.$toasted.show('Horse Added', {type: 'success'})   },
 onHorseUpdated (value) {
  let pos = this.horses.map(function (e) {
       return e.id }).indexOf(value.id)
  this.$set(this.horses[pos], 'name', value.name)
  this.changePairsAfterEdit(this.pairs, 'horse', value.id,
value.name)
  this.$toasted.show('Horse Updated', {type: 'success'}) },
```

Material Models

Vuetify's model pop-ups were too complicated

```
<template><div id = 'editmodal'>
<md-dialog :md-active.sync="dialog" style="background-color: #ffffff;">
 <md-dialog-title>{{message.item.name}}<br><br>{{message.message}}</md-dialog-title>
 <md-dialog-content></md-dialog-content>
 <md-dialog-actions>
   <v-btn v-if="dispSaveBtn" class="cyan" dark @click="deleteRider()">Delete</v-btn>
   <v-btn class="cyan" dark @click="cancel()">Cancel</v-btn>
 </md-dialog-actions>
</md-dialog>
<!-- dialog button -->
<v-btn flat icon color="cyan" @click="openDialog()" title="Delete this horse"><v-icon>delete</v-icon></v-btn>
</div></template>
```

Material Models cont

```
async deleteRider () {
  this.closeDialog()
  await HorsesService.deleteHorse({id: this.index})
  this.$emit('horseDeleted', this.index)
}
<dialogHorseDelete :message="{'message': 'Are you sure you want to delete this
horse?', 'item': props.item}" :index="props.item.id" @horseDeleted="onHorseDeleted">
</dialogHorseDelete>
```

The Emit calls a method called onHorseDeleted which can carry parameters

Drag and Drop -- vue-smooth-dnd

Demo...

Contact...

https://github.com/strongbryan/horseShows.git

strongbryan@hotmail.com