

Joshua Hoshiko

Project 2

Project 2 Report

Changed files:

- General
 - **routes.rb** (General): Added new routes that redirect to the search (cars_search for example)
- Car:
 - **index.html.erb** (View): Created a Search box
 - **car.rb** (Model): Updated to have validations for uniqueness and non-blank model/VIN fields
 - **cars_controller.rb** (Controller): Added a search method
- Part:
 - **index.html.erb** (View): Created a Search box
 - **part.rb** (Model): Updated to have validations for uniqueness and non-blank name field
 - **parts_controller.rb** (Controller): Added a search method
- Make:
 - **index.html.erb** (View): Created a Search box
 - **make.rb** (Model): Updated to have validations for uniqueness and non-blank name/country fields

- **makes_controller.rb** (Controller): Added a search method
- **Testing:**
 - Fixtures: Edited fixtures to have attributes that better represent the items rather than the defaults
 - Controller testing: Added test that test the search function (successful search, unsuccessful search, successful search status)
 - Model testing: Added tests for all validations in all models

Samples:

Cars

Search for:

Model	Vin	Make	
Cross Runner	HG7245	Toyota	Show Edit Destroy
Modula	457274	Mercedes	Show Edit Destroy
Roller	9823879y25782457	Chevy	Show Edit Destroy

[New Car](#)

Cars

Search for:

Model	Vin	Make	
Cross Runner	HG7245	Toyota	Show Edit Destroy

[New Car](#)

New Car

2 errors prohibited this car from being saved:

- Model can't be blank
- Vin can't be blank

Model

Vin

Make

Toyota

Car

☐ Exhaust
☒ Turbo
☐ Shocks
☐ Muffler
☒ Axle
☐ Windshield Wipers

[Back](#)

Git Diffs: a5fd514 - 323ab60

```

--- a/Gemfile
+++ b/Gemfile
@@ -56,4 +56,5 @@ end

# Windows does not include zoneinfo files, so bundle the tzinfo-data gem
gem 'tzinfo-data', platforms: [:mingw, :mswin, :x64_mingw, :jruby]

-gem 'seed_dump'
\ No newline at end of file
+gem 'seed_dump'
+

diff --git a/app/controllers/cars_controller.rb b/app/controllers/cars_controller.rb
index 16ff1a2..62aeea6 100644
--- a/app/controllers/cars_controller.rb
+++ b/app/controllers/cars_controller.rb
@@ -27,13 +27,14 @@ class CarsController < ApplicationController

  # POST /cars.json

  def create

    @car = Car.new(car_params)
+   @parts = Part.all

    respond_to do |format|
      if @car.save
        format.html { redirect_to @car, notice: 'Car was successfully created.' }
        format.json { render :show, status: :created, location: @car }
      else
-       format.html { render :new }
+       format.html { render :new }

```

```

        format.json { render json: @car.errors, status: :unprocessable_entity }
      end
    end
@@ -53,6 +54,11 @@ class CarsController < ApplicationController
      end
    end

+ def search
+   @cars = Car.where("model like ?", "%#{params[:query]}%")
+   render :index
+ end
+
  # DELETE /cars/1
  # DELETE /cars/1.json
  def destroy
diff --git a/app/controllers/makes_controller.rb b/app/controllers/makes_controller.rb
index 19460d9..cc6a559 100644
--- a/app/controllers/makes_controller.rb
+++ b/app/controllers/makes_controller.rb
@@ -51,6 +51,11 @@ class MakesController < ApplicationController
      end
    end

+ def search
+   @makes = Make.where("name like ?", "%#{params[:query]}%")
+   render :index
+ end
+

```

```

# DELETE /makes/1

# DELETE /makes/1.json

def destroy

diff --git a/app/controllers/parts_controller.rb b/app/controllers/parts_controller.rb
index d674130..305507b 100644
--- a/app/controllers/parts_controller.rb
+++ b/app/controllers/parts_controller.rb
@@ -28,6 +28,7 @@ class PartsController < ApplicationController
  # POST /parts.json
  def create
    @part = Part.new(part_params)
+   @cars = Car.all

    respond_to do |format|
      if @part.save
@@ -54,6 +55,11 @@ class PartsController < ApplicationController
    end
  end

+ def search
+   @parts = Part.where("name like ?", "%#{params[:query]}%")
+   render :index
+ end
+
  # DELETE /parts/1
  # DELETE /parts/1.json
  def destroy

diff --git a/app/models/car.rb b/app/models/car.rb

```

```

index b636ddc..ac88bd2 100644
--- a/app/models/car.rb
+++ b/app/models/car.rb
@@ -1,4 +1,7 @@
class Car < ApplicationRecord
  belongs_to :make

  has_and_belongs_to_many :parts
+
+ validates :model, presence: true, uniqueness: true
+ validates :vin, presence: true, uniqueness: true
end
diff --git a/app/models/make.rb b/app/models/make.rb
index b5d9750..e7cd633 100644
--- a/app/models/make.rb
+++ b/app/models/make.rb
@@ -1,3 +1,6 @@
class Make < ApplicationRecord
  has_many :cars
+
+ validates :name, presence: true, uniqueness: true
+ validates :country, presence: true, uniqueness: true
end
diff --git a/app/models/part.rb b/app/models/part.rb
index 72dc3f4..4872522 100644
--- a/app/models/part.rb
+++ b/app/models/part.rb
@@ -1,3 +1,5 @@
class Part < ApplicationRecord

```

```

    has_and_belongs_to_many :cars
+
+ validates :name, presence: true, uniqueness: true
end
diff --git a/app/views/cars/index.html.erb b/app/views/cars/index.html.erb
index b9019be..fa5c828 100644
--- a/app/views/cars/index.html.erb
+++ b/app/views/cars/index.html.erb
@@ -2,6 +2,12 @@

<h1>Cars</h1>

+<%= form_tag(search_cars_url, method: "get") do %>
+ <%= label_tag(:query, "Search for:") %>
+ <%= text_field_tag(:query) %>
+ <%= submit_tag("Search") %>
+<% end %>
+
<table>
  <thead>
    <tr>
diff --git a/app/views/makes/index.html.erb b/app/views/makes/index.html.erb
index f675780..078808a 100644
--- a/app/views/makes/index.html.erb
+++ b/app/views/makes/index.html.erb
@@ -2,6 +2,12 @@

<h1>Makes</h1>

```

```
+<%= form_tag(search_makes_url, method: "get") do %>
+ <%= label_tag(:query, "Search for:") %>
+ <%= text_field_tag(:query) %>
+ <%= submit_tag("Search") %>
+<% end %>
```

```
+
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
diff --git a/app/views/parts/index.html.erb b/app/views/parts/index.html.erb
```

```
index bbc1cf9..3621ea3 100644
```

```
--- a/app/views/parts/index.html.erb
```

```
+++ b/app/views/parts/index.html.erb
```

```
@ @ -2,6 +2,12 @ @
```

```
<h1>Parts</h1>
```

```
+<%= form_tag(search_parts_url, method: "get") do %>
+ <%= label_tag(:query, "Search for:") %>
+ <%= text_field_tag(:query) %>
+ <%= submit_tag("Search") %>
+<% end %>
```

```
+
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
diff --git a/config/routes.rb b/config/routes.rb
```



```
index 76b1337..522110f 100644
```

```
--- a/config/routes.rb
```

```
+++ b/config/routes.rb
```

```
@@ -1,6 +1,18 @@
```

```
  Rails.application.routes.draw do
```

```
    - resources :parts
```

```
    - resources :makes
```

```
    - resources :cars
```

```
    + resources :parts do
```

```
      + collection do
```

```
        + get 'search'
```

```
      + end
```

```
    + end
```

```
    + resources :makes do
```

```
      + collection do
```

```
        + get 'search'
```

```
      + end
```

```
    + end
```

```
    + resources :cars do
```

```
      + collection do
```

```
        + get 'search'
```

```
      + end
```

```
    + end
```

```
    # For details on the DSL available within this file, see
    http://guides.rubyonrails.org/routing.html
```

```
  end
```

```
diff --git a/test/controllers/cars_controller_test.rb b/test/controllers/cars_controller_test.rb
```

```
index c2f842f..d0dbeea 100644
```

```
--- a/test/controllers/cars_controller_test.rb
```

```
+++ b/test/controllers/cars_controller_test.rb
```

```
@@ -45,4 +45,18 @@ class CarsControllerTest < ActionDispatch::IntegrationTest
```

```
    assert_redirected_to cars_url
```

```
  end
```

```
+
```

```
+ test "should find car from the fixture" do
```

```
+   assert Car.where("model like ?", "Cross Runner").length == 1
```

```
+ end
```

```
+
```

```
+ test "should not find car from the fixture" do
```

```
+   assert Car.where("model like ?", "Roller").length == 0
```

```
+ end
```

```
+
```

```
+ test "searches always return 200" do
```

```
+   get search_cars_url, params: { search: "Skipper" }
```

```
+   assert_equal 200, status
```

```
+ end
```

```
+
```

```
end
```

```
diff --git a/test/controllers/makes_controller_test.rb b/test/controllers/makes_controller_test.rb
```

```
index 52dfd39..0383146 100644
```

```
--- a/test/controllers/makes_controller_test.rb
```

```
+++ b/test/controllers/makes_controller_test.rb
```

```
@@ -45,4 +45,17 @@ class MakesControllerTest < ActionDispatch::IntegrationTest
```

```
    assert_redirected_to makes_url
```

```
  end
```

```

+
+ test "should find make from the fixture" do
+   assert Make.where("name like ?", "Toyota").length == 1
+ end
+
+ test "should not find make from the fixture" do
+   assert Make.where("model like ?", "Lamborghini").length == 0
+ end
+
+ test "searches always return 200" do
+   get search_makes_url, params: { search: "Mazda" }
+   assert_equal 200, status
+ end
end
diff --git a/test/controllers/parts_controller_test.rb b/test/controllers/parts_controller_test.rb
index 0191216..b857c97 100644
--- a/test/controllers/parts_controller_test.rb
+++ b/test/controllers/parts_controller_test.rb
@@ -45,4 +45,17 @@ class PartsControllerTest < ActionDispatch::IntegrationTest

    assert_redirected_to parts_url
  end
+
+ test "should find part from the fixture" do
+   assert Part.where("name like ?", "Exhaust").length == 1
+ end
+
+ test "should not find part from the fixture" do

```

```
+ assert Part.where("model like ?", "Heated Seats").length == 0
```

```
+ end
```

```
+
```

```
+ test "searches always return 200" do
```

```
+   get search_parts_url, params: { search: "Spark Plugs" }
```

```
+   assert_equal 200, status
```

```
+ end
```

```
end
```

```
diff --git a/test/fixtures/cars.yml b/test/fixtures/cars.yml
```

```
index ed8fd7d..a94aff2 100644
```

```
--- a/test/fixtures/cars.yml
```

```
+++ b/test/fixtures/cars.yml
```

```
@@ -1,11 +1,11 @@
```

```
# Read about fixtures at http://api.rubyonrails.org/classes/ActiveRecord/FixtureSet.html
```

```
one:
```

```
- model: MyString
```

```
- vin: MyString
```

```
- make: one
```

```
+ model: Cross Runner
```

```
+ vin: 1234
```

```
+ make: Toyota
```

```
two:
```

```
- model: MyString
```

```
- vin: MyString
```

```
- make: two
```

```
+ model: Skipper
```

```
+ vin: 4321
```

```
+ make: Mazda
```

```
diff --git a/test/fixtures/makes.yml b/test/fixtures/makes.yml
```

```
index d8d63f7..c0fb096 100644
```

```
--- a/test/fixtures/makes.yml
```

```
+++ b/test/fixtures/makes.yml
```

```
@@ -1,9 +1,9 @@
```

```
# Read about fixtures at http://api.rubyonrails.org/classes/ActiveRecord/FixtureSet.html
```

```
one:
```

```
- name: MyString
```

```
- country: MyString
```

```
+ name: Toyota
```

```
+ country: United States
```

```
two:
```

```
- name: MyString
```

```
- country: MyString
```

```
+ name: Mazda
```

```
+ country: Japan
```

```
diff --git a/test/fixtures/parts.yml b/test/fixtures/parts.yml
```

```
index 56066c6..2766515 100644
```

```
--- a/test/fixtures/parts.yml
```

```
+++ b/test/fixtures/parts.yml
```

```
@@ -1,7 +1,7 @@
```

```
# Read about fixtures at http://api.rubyonrails.org/classes/ActiveRecord/FixtureSet.html
```

```
one:
```

```
- name: MyString
+ name: Exhaust
```

```
two:
```

```
- name: MyString
+ name: Spark Plugs
```

```
diff --git a/test/models/car_test.rb b/test/models/car_test.rb
```

```
index 39bdaec..7e17d8f 100644
```

```
--- a/test/models/car_test.rb
```

```
+++ b/test/models/car_test.rb
```

```
@@ -1,7 +1,25 @@
```

```
require 'test_helper'
```

```
class CarTest < ActiveSupport::TestCase
```

```
- # test "the truth" do
```

```
- #   assert true
```

```
- # end
```

```
+ test "for empty model" do
```

```
+   s = Car.create({:model => "", :vin => "1234"})
```

```
+   refute s.valid?
```

```
+   refute s.save
```

```
+ end
```

```
+
```

```
+ test "for empty vin" do
```

```
+   s = Car.create({:model => "Car", :vin => ""})
```

```
+   refute s.valid?
```

```
+   refute s.save
```

```
+ end
```

```

+
+ test "for non-unique car" do
+   s = Car.create({:model => "Car", :vin => "1234"})
+   r = Car.create({:model => "Car", :vin => "1234"})
+   assert s.valid?
+   assert s.save
+   refute r.valid?
+   refute r.save
+ end
+
end
diff --git a/test/models/make_test.rb b/test/models/make_test.rb
index 8162f47..e3ef7e7 100644
--- a/test/models/make_test.rb
+++ b/test/models/make_test.rb
@@ -1,7 +1,24 @@
require 'test_helper'

class MakeTest < ActiveSupport::TestCase
- # test "the truth" do
- #   assert true
- # end
+ test "for empty name" do
+   s = Make.create({:name => "", :country => "United States"})
+   refute s.valid?
+   refute s.save
+ end
+

```

```

+ test "for empty country" do
+   s = Make.create({:name => "Saab", :country => ""})
+   refute s.valid?
+   refute s.save
+ end
+
+ test "for non-unique make" do
+   s = Make.create({:name => "Saab", :country => "Sweden"})
+   r = Make.create({:name => "Saab", :country => "Sweden"})
+   assert s.valid?
+   assert s.save
+   refute r.valid?
+   refute r.save
+ end
end
diff --git a/test/models/part_test.rb b/test/models/part_test.rb
index bba3103..b8693c5 100644
--- a/test/models/part_test.rb
+++ b/test/models/part_test.rb
@@ -1,7 +1,19 @@
require 'test_helper'

class PartTest < ActiveSupport::TestCase
- # test "the truth" do
- #   assert true
- # end
+ test "for empty name" do
+   s = Car.create({:name => ""})

```



```
+ refute s.valid?  
+ refute s.save  
+ end  
  
+  
+ test "for non-unique part" do  
+   s = Part.create({:name => "Exhaust"})  
+   r = Part.create({:name => "Exhaust"})  
+   assert s.valid?  
+   assert s.save  
+   refute r.valid?  
+   refute r.save  
+ end  
  
+  
end
```