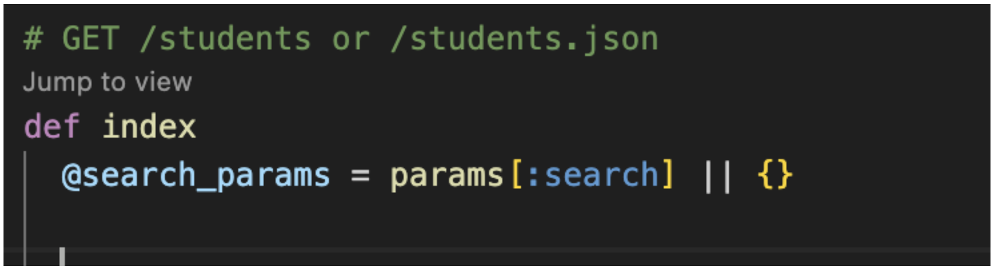
How to Implement a Search Function into your Ruby on Rails Project

Table of Contents:

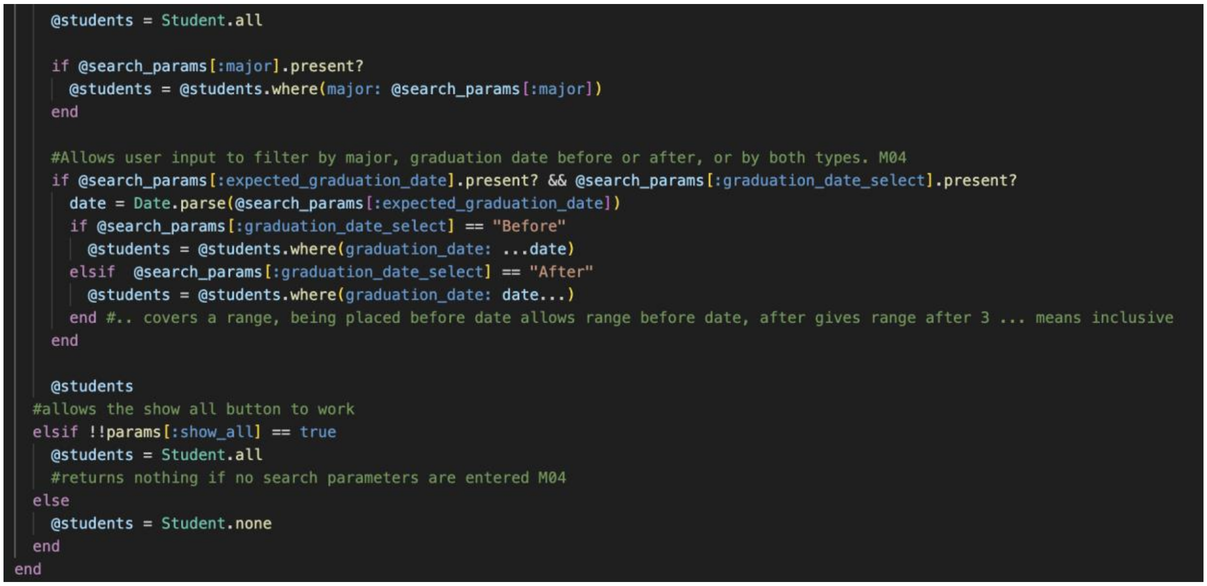
Controller Methods to be able to search your database

- This method will go into the index of your controller.



```
# GET /students or /students.json
Jump to view
def index
  @search_params = params[:search] || {}
```

- The first thing to do is to define your @search_params, these will be based off the params you set up earlier in your project.
- Then you will implement your logic to search the database. The methodology I chose was to start with all students, and filter little by little until we have the results we want left to show.

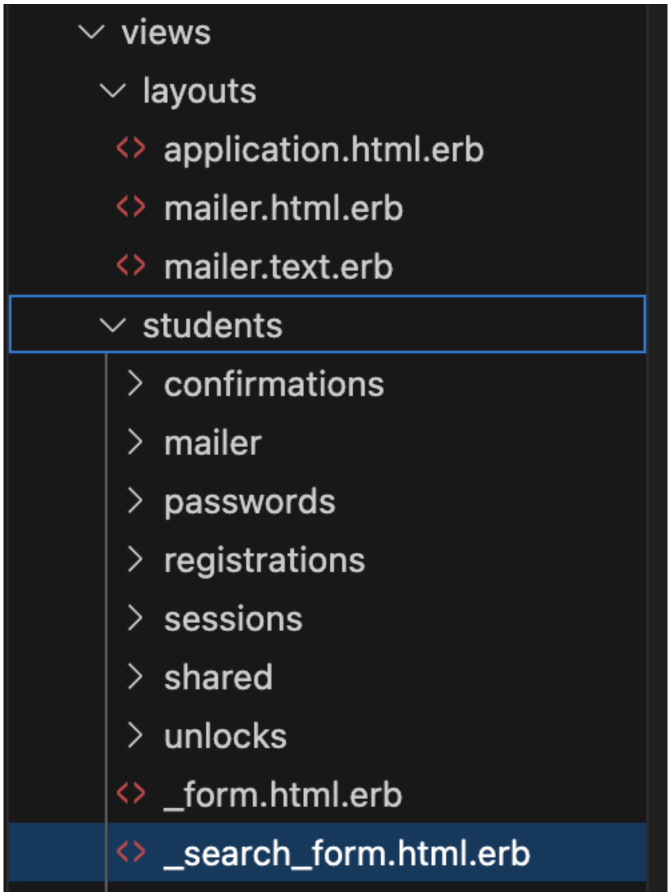


```
    @students = Student.all

    if @search_params[:major].present?
      @students = @students.where(major: @search_params[:major])
    end

    #Allows user input to filter by major, graduation date before or after, or by both types. M04
    if @search_params[:expected_graduation_date].present? && @search_params[:graduation_date_select].present?
      date = Date.parse(@search_params[:expected_graduation_date])
      if @search_params[:graduation_date_select] == "Before"
        @students = @students.where(graduation_date: ...date)
      elsif  @search_params[:graduation_date_select] == "After"
        @students = @students.where(graduation_date: date...)
      end #.. covers a range, being placed before date allows range before date, after gives range after 3 ... means inclusive
    end

    @students
  #allows the show all button to work
  elsif !!params[:show_all] == true
    @students = Student.all
    #returns nothing if no search parameters are entered M04
  else
    @students = Student.none
  end
end
```

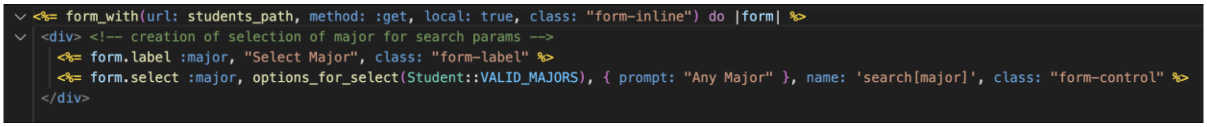
- Here is the logic I used to get my search function to work.
    - It starts by pulling all students.
    - It then checks to see if the major has been selected as a parameter, and if so will filter results accordingly.
        - You must parse the selected graduation date to get the range to function properly.
    - Then it will check to see the graduation date and range selection is filled out as well.
    - Pay particular attention to the comments about the range, and what is inclusive or exclusionary for the range.
- After the end of the logic for my search function, I added the ability to show all students without filtering any results. This can be a handy function to add.
- The end of my search logic ends with the else no search params are given, no students will be shown.

Accessing your Search Function via Views

- Now that you have implemented the logic to your search function, you need to be able to access the logic in your app.
- For this, you are going to start by creating a new file under views/students, the _search_form.html.erb.

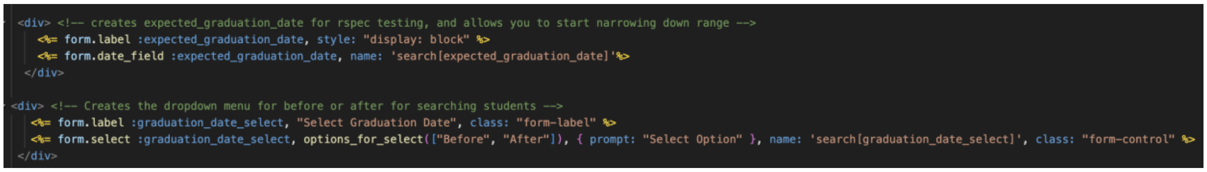
- 

```
views
  layouts
    application.html.erb
    mailer.html.erb
    mailer.text.erb
  students
    confirmations
    mailer
    passwords
    registrations
    sessions
    shared
    unlocks
    _form.html.erb
    _search_form.html.erb
```

- From here, you will add the necessary html to be able to use the search function on your webapp.
- 

```
<%= form_with(url: students_path, method: :get, local: true, class: "form-inline") do |form| %>
<div> <!-- creation of selection of major for search params -->
  <%= form.label :major, "Select Major", class: "form-label" %>
  <%= form.select :major, options_for_select(Student::VALID_MAJORS), { prompt: "Any Major" }, name: 'search[major]', class: "form-control" %>
</div>
```

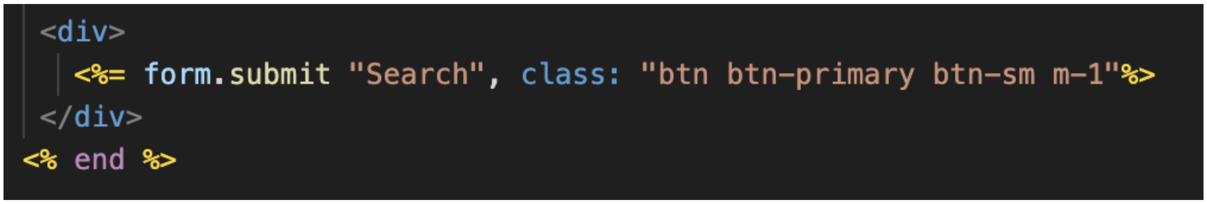
- This creates a form using the students_path route.
- The first <div> section creates the major selection drop down menu.
    a. The difference between this is the form.select and the options_for_select portion that create a drop down menu.
- The next two <div> sections will allow you to select a graduation date and if you want results before that date or after.



```
 <div> <!-- creates expected_graduation_date for rspec testing, and allows you to start narrowing down range -->
   <%= form.label :expected_graduation_date, style: "display: block" %>
   <%= form.date_field :expected_graduation_date, name: 'search[expected_graduation_date]'%>
 </div>

<div> <!-- Creates the dropdown menu for before or after for searching students -->
  <%= form.label :graduation_date_select, "Select Graduation Date", class: "form-label" %>
  <%= form.select :graduation_date_select, options_for_select(["Before", "After"]), { prompt: "Select Option" }, name: 'search[graduation_date_select]', class: "form-control" %>
</div>
```

- The selection for range is very similar to the major selector because it is also a dropdown menu.

- However the expected_graduation_date is a date_field, allowing you to select a date off a calender.
- The last thing needed is to implement a search bar.



```
  <div>
    <%= form.submit "Search", class: "btn btn-primary btn-sm m-1"%>
  </div>
<% end %>
```

- This is a form.submit, showing you are submitting the given parameters to search your database.
- For further information about the class: “btn btn-primary" things, please see UI and Bootstrap documentation.