

Graduation Date Filtering

User Goal: Being able to search for all students who graduate on, before, or after a specific date.

Part 1: Getting and searching by a specific date.

Part 2: Sorting by before or after a specific date.

Sorting By Graduation Date

In The View:

The following div and form elements should be added to `app\views\students_search_form_html.erb` under the Search By Major field. They implement a field where a date can be typed as mm / dd / yyyy, or manually selected using a built-in calendar option. The input value is then passed as `:graduation_date`.

```
<div>

  <%= form.label :graduation_date, "Select Graduation Date", class: "form-label" %>

  <%= form.date_field :graduation_date, value: params.dig(:search, :graduation_date), name:
'search[graduation_date]', class: "form-control" %>

</div>
```

```
<div>
  <%= form.label :graduation_date, "Select Graduation Date", class: "form-label" %>
  <%= form.date_field :graduation_date, value: params.dig(:search, :graduation_date), name: 'search[graduation_date]', class: "form-control" %>
</div>
```

In the Controller:

Implement the following if statement to check if the search query contains a graduation date or if the field was left blank.

```
if @search_params[:graduation_date].present?
end
```

Inside the if statement but before the end statement, insert the if/elsif/else sequence below. The statements will read the value of `:beforeafter` from the form (explained in the next section) to determine what comparison to make. Inside of each statement, `@students` is searched for students whose `graduation_date` attribute compares to the search form's `:graduation_date` value. The use of the `Date.parse` method and `"?"` instead of direct comparison are used as additional insurance of data integrity.

```
if @search_params[:beforeafter] == "Before"

  @students = @students.where("graduation_date <= ?",
Date.parse(@search_params[:graduation_date]))

elsif @search_params[:beforeafter] == "After"

  @students = @students.where("graduation_date >= ?",
Date.parse(@search_params[:graduation_date]))

else

  @students = @students.where("graduation_date == ?",
Date.parse(@search_params[:graduation_date]))

end
```

```

if @search_params[:graduation_date].present?
  # Print the selected date and the selected timeframe choice
  Rails.logger.info "#{Date.parse(@search_params[:graduation_date])}"
  Rails.logger.info "#{@search_params[:beforeafter]}"

  # Sorts by timeframe choice, or only the exact date if timeframe was left at the default.
  if @search_params[:beforeafter] == "Before"
    @students = @students.where("graduation_date <= ?", Date.parse(@search_params[:graduation_date]))
  elsif @search_params[:beforeafter] == "After"
    @students = @students.where("graduation_date >= ?", Date.parse(@search_params[:graduation_date]))
  else
    @students = @students.where("graduation_date == ?", Date.parse(@search_params[:graduation_date]))
  end
end
end

```

However, there is not yet anything to specifically handle a search where neither a major nor a graduation date was input (even if the timeframe is specified). Insert the following statement, which will show zero students as there are zero valid criteria.

```

  if (@search_params[:major].present? == false) && (@search_params[:graduation_date].present? == false)

    @students = @students.none

  end

```

```

  if (@search_params[:major].present? == false) && (@search_params[:graduation_date].present? == false)
    # If neither major nor date are supplied when search is pressed, this will show no students.
    @students = @students.none
  end

```

Sorting By Timeframe

In the View:

The following div and form elements should be added to `app\views\students_search_form.html.erb` right above or below the `graduation_date` form elements. They implement a dropdown where the default option is “Exact Date”, but other selectable options are “Before” and “After”. The chosen value is then passed as `:beforeafter`.

```

<div>

  <%= form.label :beforeafter, "Timeframe", class: "form-label" %>

  <%= form.select :beforeafter, options_for_select(["Before", "After"]), { prompt: "Exact Date" }, name: 'search[beforeafter]', class: "form-control" %>

</div>

```

```

<div>
  <%= form.label :beforeafter, "Timeframe", class: "form-label" %>
  <%= form.select :beforeafter, options_for_select(["Before", "After"]), { prompt: "Exact Date" }, name: 'search[beforeafter]', class: "form-control" %>
</div>

```

In the Controller:

If instructions from the prior section were followed, the `if/elsif/else` statements evaluating `:beforeafter` are already implemented and nothing else needs to be added inside the outer `if` statement. For clarity, the sequence checks if the user selected “Before”, then “After”, or does an “Exact” comparison if neither was

selected (because Exact is the default option). For a transcribed excerpt, see [Sorting By Graduation Date](#) under **In the Controller**.

```
if @search_params[:graduation_date].present?
  # Print the selected date and the selected timeframe choice
  Rails.logger.info "#{Date.parse(@search_params[:graduation_date])}"
  Rails.logger.info "#{@search_params[:beforeafter]}"

  # Sorts by timeframe choice, or only the exact date if timeframe was left at the default.
  if @search_params[:beforeafter] == "Before"
    | @students = @students.where("graduation_date <= ?", Date.parse(@search_params[:graduation_date]))
  elsif @search_params[:beforeafter] == "After"
    | @students = @students.where("graduation_date >= ?", Date.parse(@search_params[:graduation_date]))
  else
    | @students = @students.where("graduation_date == ?", Date.parse(@search_params[:graduation_date]))
  end
end
```