

# Search Function Technical Document:

We need to be able to search students by major or graduation date.

## Feature to select major: Student Model

In your student model add the highlighted code.

app/models/student.rb

VALID\_MAJORS is a constant array that stores the list of accepted majors.

The validation ensures that the major selected by the user is included in this list

```
VALID_MAJORS = ["Computer Engineering BS", "Computer Information Systems BS",  
  "Computer Science BS", "Cybersecurity Major", "Data Science and Machine Learning  
Major"]
```

```
validates :major, inclusion: { in: VALID_MAJORS, message: "%{value} is not a valid major" }
```

## Views/Form:

The following code will go into your app/views/students/\_form.html.erb file.

- form.select :major creates a dropdown menu with options.
- options\_for\_select(Student::VALID\_MAJORS) automatically generates options for each major defined in the VALID\_MAJORS constant.
- { prompt: "Select a Major" } adds a placeholder to prompt the user to choose a major.

```
<div>  
  <%= form.label :major, style: "display: block" %>  
  <%= form.select :major, options_for_select(Student::VALID_MAJORS), { prompt: "Select a  
Major" }, class: "form-control" %>  
</div>
```

## Student Controller:

In your app/controllers/students\_controller.rb. There is also the code here to filter by graduation date.

- Params[:search]: Data is passed from the form field as a hash to know what attribute and value to search and will be stored in @search\_params
- || {} If no search criteria is provided in search parameters then Params[:search]: assigns an empty hash {}

- Once a search is submitted through the form then check if search is present for major and will use ORM of active record to query the database `@students.where(major: @search_params[:major])`

```
def index
  @search_params = params[:search] || {} # Ensure @search_params is a hash

  # Start with an empty relation, so no students are shown initially
  @students = Student.none

  # Display students only if search parameters are present or "Show All" is clicked
  if @search_params.present?
    @students = Student.all

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

    # Filter by graduation date
    if @search_params[:graduation_date_filter].present? &&
      @search_params[:graduation_date].present?
      date = Date.parse(@search_params[:graduation_date])
      if @search_params[:graduation_date_filter] == 'before'
        @students = @students.where('graduation_date < ?', date)
      elsif @search_params[:graduation_date_filter] == 'after'
        @students = @students.where('graduation_date > ?', date)
      end
    end
  elsif params[:show_all] # If "Show All" is clicked
    @students = Student.all
  end
end
```

## Views/Form:

In your Views create a new partial search form fragment `app/views/students/_search_form.html.erb` and copy this code into that file.

```
<%= form_with(url: students_path, method: :get, local: true, class: "form-inline") do |form| %>
<div>
  <%= 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>
```

```
<div>
```

```
<%= form.submit "Search", class: "btn btn-primary" %>
```

```
</div>
```

```
<% end %>
```

After your search partial has been created go into your index.html.erb file and render the search app/views/students/index.html.erb where the parameters are passed to dynamically create form

- **Form\_with** is a helper method that creates the form HTML and adds the correct URL and method for form submission.
- The form must specify where to submit request by using url: students\_path that is a part of the Rails routing
- method: :get: The http method must be set

```
<%= render 'search_form', search_params: @search_params %>
```