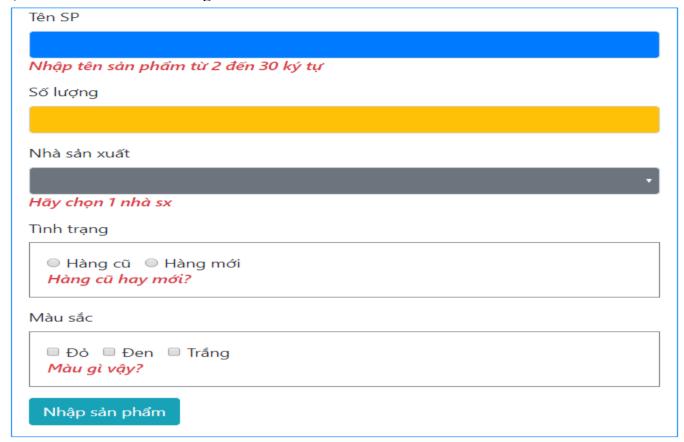
Lab 8: Sử dụng AngularJS

Bài 1: (6 diễm) Kiểm lỗi trong form

Tạo form và kiểm tra lỗi trong form



1. Chuẩn bị:

- Tao file bai1.html

```
<body>
<form name="frm" class="col-5 mx-auto border border-primary">
    <div class="form-group">
        <label>Tên SP</label>
        <input name="tensp" class="form-control bg-primary">
    <div class="form-group">
        <label>Số lượng</label>
        <input name="soluong" class="form-control bg-warning" >
    </div>
    <div class="form-group">
        <label>Nhà sản xuất</label>
        <select name="nsx" class="form-control bg-secondary text-white">
            <option value="0">Chon NSX</option>
            <option value="1">Samsung</option>
            <option value="2">Toshiba</option>
            <option value="3">HTC</option>
        </select>
    </div>
    <div class="form-group">
        <label>Tinh trang</label>
        <fieldset class="border border-secondary p-3">
            <input name="tinhtrang" type="radio" value="1"> Hàng cũ &nbsp;
            <input name="tinhtrang" type="radio" value="2"> Hàng mới
```

- Chèn thư viện angular và css bootstrap vào đầu trang

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular.min.js"> </
script>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
```

Tạo ứng dụng tên b3
 Thêm thuộc tính sau vào tag body

```
ng-app="b3"
```

Tạo controller tên ctr1
 Thêm thuộc tính sau vào tag body

```
ng-controller="ctr1"
```

- Code khởi tạo angularjs (để tạo app, gắn controller vào hàm xử lý)

```
<script>
    app= angular.module("b3",[]);
    app.controller("ctr1",xuly);
    function xuly($scope){
        //code xử lý
    }
</script>
```

2. Validate với tensp

- Bind giá trị của tensp vào scope :

Thêm thuộc tính sau vào tag input có name là tensp

```
ng-model="tensp"
```

Thêm quy tắc nhập liệu cho tensp: bắt buộc nhập, từ 2 đến 30 ký tự Bổ sung vào tag input có name là tensp các thuộc tính sau:

```
required minlength="2" maxlength="30"
```

- Hiên thông báo lỗi

Code sau tag input có name là tensp

```
<em ng-if="frm.tensp.$invalid" class="text-
danger h6">Nhập tên sản phẩm từ 2 đến 30 ký tự</em>
```

Test

3. Validate với soluong

- Bind giá tri của soluong vào scope :

Thêm thuộc tính sau vào tag input có name là soluong

ng-model="soluong"

- Thêm quy tắc nhập liệu cho soluong: kiểu số, nhỏ nhất là 1 Bổ sung vào tag input có name là soluong các thuộc tính sau:

```
type="number" min="1"
```

Hiện thông báo lỗi khi số lượng nhập ngoài giá trị 1-100
 Code sau tag input có name là soluong

```
<em ng-if="frm.soluong.$invalid || soluong>100" class="text-
danger h6">Nhập số lượng từ 1 đến 100
```

- Test

4. Validate với nhà sản xuất (selectbox)

- Bind giá trị của nsx vào scope:

Thêm thuộc tính sau vào tag select có name là nsx

```
ng-model="nsx"
```

- Hiện thông báo lỗi khi user không chọn mục nào hoặc chọn mục đầu tiên trong selectbox Code sau tag select có name là nsx

```
<em ng-if="!nsx || nsx==0" class="text-danger h6">Hãy chọn 1 nhà sx</em>
```

- Test

5. Validate với tinhtrang (radio button)

- Bind giá trị của tinhtrang vào scope:

Thêm thuộc tính sau vào 2 tag input có name là tinhtrang

```
ng-model="tinhtrang"
```

Hiện thông báo lỗi khi không chọn cái nào trong radio
 Code sau 2 tag input có name là tinhtrang

```
<em ng-if="!tinhtrang" class="text-danger h6"><br>Hàng cũ hay mới?</em>
```

- Test

6. Validate với màu sắc (check box)

- Bind giá trị của tinhtrang vào scope:

Thêm thuộc tính sau vào tag input có value là do

```
ng-model="maudo"
```

Thêm thuộc tính sau vào tag input có value là den

```
ng-model="mauden"
```

Thêm thuộc tính sau vào tag input có value là trang

```
ng-model="mautrang"
```

Hiện thông báo lỗi khi không chọn cái nào trong các checkbox
 Code sau 3 tag input có checkbox

```
<em ng-if="!maudo && !mauden && !mautrang " class="text-
danger h6"><br>>Màu gì vậy?
```

- Test

7. Thông báo thành công

Nếu user nhập đúng tất cả thì hiện dòng chữ **Bạn nhập dữ liệu đã chính xác.Cảm ơn nhé.** Đồng thời 5 giây sau chuyển tới trang web khác. (vd https://longnv.name.vn)

8. Định dạng

- Sinh viên định dạng trang web, form, các controler cho đẹp: width, height, border, shadow, margin, padding...
- Nhập thông tin sinh viên vào tiêu đề trang web: MaSV Họ Tên
- Định dạng css cho các field khi có lỗi : màn nền, border

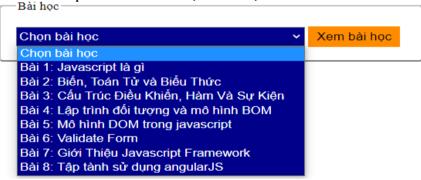
Bài 2: (4 điểm) Ôn tập

- 1. Tạo trang web với yêu cầu sau:
- Cho hiện đồng hồ ở giữa trang
- Phía dưới đồng hồ có 3 button Start, Stop, Xem bài học
- Nếu user nhắp nút Stop: đồng hồ dừng lại, user nhắp nút Start thì đồng hồ chạy tiếp
- Nếu user nhắp nút **Xem bài học** thì chuyển đến trang https://longnv.name.vn/lap-trinh-javascript
- Định dạng css
- 2. Tạo trang web có giao diện như sau:

Tạo button chứa chữ **Xem bài học** và 1 selectbox với các mục là các bài học của môn



- Khi nhắp selectbox sẽ hiện các mục như sau:



- Vào địa chỉ https://longnv.name.vn/lap-trinh-javascript, sẽ thấy các bài học có tiêu đề tương ứng với các mục Bài 1 đến Bài 8 ở hình trên.
- Trong sự kiện click của button **Xem bài học** gọi hàm **xembaihọc**()
- Định nghĩa hàm xembaihoc():
 Nếu user chưa chọn bài học (Bài 1 đến Bài 8) thì báo lỗi alert: Bạn chưa chọn bài học mà
 Còn nếu user đã chon 1 mục từ Bài 1 đến Bài 8 thì mở trang web tương ứng với bài học