Finals prep

Sunday, 26 February 2023 3:09 pm

JQUERY

RegularExpression=(@"^\d{3}-\d{3}-\d{4}$"

```
Public Class StudentModel{

[Required(Error Message="You need to enter the name")]
[StringLength(20, Minlength=7, ErrorMessage="Name must be between 7 to 20 characters")]
Public string Name {get;set;}


[Required]
[EmailAddress]
Public string Email {get;set;}

Public string Phone {get; set;}



}
```

Now creating a form in html

```
<form id ="studentForm">

<input type ="text" id="name" Name="name" />

<input type="email" id="email" Name="Email" />


<label asp-for="phone"></label>
<input asp-for="phone" type="number" id="phone"/>
<span asp-validation-for ="phone" class="text-danger" ></span>

<button type="button" onclick="submitButton()>Submit </button>
```

In script
Merthod1

```
Function submitButton()
{

Var myData= $("#studentForm").serialize();    //this is jquery

$.ajax({
Type: 'post',
Url : '/Home/CreateStudent',

Data:myData

// now making success and error function

Success: function(result)
{
Console.log(we are in success function);
}

Error: function(result)
{
Console.log(we are in error function);
}


})

}
```

Method2

```
Function submitBtn()
{

Var myData={
Name: $('#name').val(),
Email:$('#email').val(),


}


$.ajax(

{

Type:'post',
contentType:applicatoin/json,
Data:Json.stringify(myData),
Url:'/Student/addstudent',
Success:function(result)
{
Console.log(in success);
}
Error:function(result)
{
```

```
Fr.student.where(s => s.Email==email).select(s=> s.Name).tostring();

List<string>name= new List<string();
Foreach(var e in emails)
{
  string myName= fr.Student.where(s => s.email==e).Select(s => s.Name).first().toString();

  name.add(myName);
Console.log(in success);
}
Error:function(result)
{
Console.log(in error function);
}

}
)
}
```

```
{
  string myName= fr.Student.where(s => s.email==e).Select(s => s.Name).first().toString();

  name.add(myName);

}
Retrn
```

FILE UPLOADING WITH AJAX

```
< INPUT TYPE=' FILE' ID = 'POSTEDFILE'  MULTUIPLE/>
<BUTTON TYPE='BUTTON' ONCLICK = ' UPLOADFILE()'>
```

```
FUNCTION UPLOADFILE()
{

VAR INPUT= $(' #POSTEDFILE');
VAR FILE= INPUT.FILES;  // WE GET ALL FILES

VAR FORMDATA = NEW FORMDATA();

FOR( INT I=0; I < FILE.LENGTH ; I++)
{
        FORMDATA.APPEND(" FILE" , FILE[I]);

}

$.AJAX(
{

TYPE:'POST',
CONTENTYPE:FALSE,
PROCESSDATA: FALSE;
URL:/HOME/UPLOADFILE,
DATA: FORMDATA,

SUCCESS:FUNCTION(RESULT){}
ERROR : FUNCTION(RESULT){}



}

)


}
```

```
[http post]
PUBLIC IACTIONRESULT UPLOADFILE(LIST<IFORMFILE>files)
{

String wwwpath= this.enviroment.webrootpath;
String path=Path.combine(wwwpath, 'UPLOADS');

If(!Directory.exist(path)
{
Directory.CreateDirectory(path);
}

Foreach(file in files)
{
Var fileName= path.getfilename(file.filename);
Var fileNameWithPath= Path.combine(path,fileName);

Using (FileStream stream = newFileStream(pathwithfileName, fileMode.Create))
{
        file.copyto(stream);
}

}
Return json(true);

}
```

DEPENDANCY INJECTION

Create an interface class inside interface folder. Make sure your interface folder is inside modsl folder

```
Public interface Iemployee
{
   public List<Employees> GetEmployees();


}
```

```
Public class EmployeeRepo: Iemployee
{

Public List<Employees> GetEmployees()
{
     froshContext fc= new froshContext();

      return fc.Employees.Get().toList<Employees>();

}


}
```

```
Public class EmployeeController: Controller
{



         Private readonly Iemployee empRepo;

         Public EmployeeController(Iemployee e)
         {
                  empRepo=e;
         }

         Public IActionResult GetEmployees()
         {
                  Return View(empRepo.GetEmployees());
         }
}
```

In program.cs file

```
Builder.services.Addscoped<Iemployeee, EmployeeRepo>();
```