

## CHAPTER 1

### INTRODUCTION

#### A. COMPANY PROFILE

##### ***PAMANTASAN NG LUNGSOD NG MAYNILA***

On 19 June 1965, the final bill entitled "An Act Authorizing the City of Manila to Establish and Operate the University of the City of Manila and for Other Purposes" was signed into law by President Diosdado P. Macapacal in a signing ceremony in Malacañang Palace witnessed by Atty. Primitivo de Leon, Mayor Villegas, Congressman Ramon Mitra Jr., and its main sponsor in the House of Representatives, Congressman Albert. The law was captioned as Republic Act No. 4196 which now serves as the University Charter.

On February 5, 1970, then Executive Secretary Ernesto M. Macea, by authority of then President Ferdinand Marcos, conveyed unto the City of Manila two (2) parcels of land owned by the Republic of the Philippines. Said real properties consist the area where PLM stands today. And as solemnly endeared and inscribed by every PLM member to his heart, the University formally opened the University College on Monday, 17 July 1967 with an initial enrollment of 556 outstanding and bright scholars carefully selected from the upper 10% of the various public high schools in Manila. Indeed, excellence has been a "birthright" of PLM. The PLM also established the Graduate College a year later, followed by the Institute for Extra-Mural Studies. And the rest was history in the making.

## B. WHAT IS THE SYSTEM ALL ABOUT

This System is about the Information Management of PFMO (Physical Facilities Management Office) of Pamantasan ng Lungsod ng Maynila. From adding, to editing, and deleting of Sections, Section Heads, Inventory, and Personnel tasks. It can also archive old blueprints, reports, memos, and other important documents needed for future reference. Report Generation is also a part of this system. It is generated from the information that has been added or changed from the system. Sections are the different sections that are under PFMO. Section Heads are the people who are in-charge of the different sections. Inventory holds the list, quantity, categories, and descriptions of the items or supplies under PFMO. Personnel tasks are the different tasks or duties that are assigned to the personnel of PFMO.

## C. STATEMENT OF THE PROBLEM

**Problem 1:** Slow and Tedious process of borrowing and returning of items and equipment.

**Problem 2:** No existing process of tracking of special project supplies.

**Problem 3:** Management receives no real time update from the custodians for each tools, equipment and supplies issued or returned.

## D. OBJECTIVES

### GENERAL OBJECTIVES

The main objectives of the study are to apply the principle and concepts to provide a solution to the existing problems of the company and provide an automated management system for the Pamantasan ng Lungsod ng Maynila, Physical Facilities and Management Office Inventory and Supplies Management and Tracking Section.

### SPECIFIC OBJECTIVES

The specific objectives of the study based on the problems stated:

1. To immediately check tools and equipment quantity in the inventory.
2. To directly track and monitor outgoing equipment, tools and supplies.
3. To update the Chief PFMO of received and issued equipment, tools and supplies from the three property sections (Special Projects, Quarterly Supplies, Equipment and Tools).

## E. SCOPE AND LIMITATIONS

This study is about the Management Information System of PFMO in Pamantasan ng Lungsod ng Maynila. The focus of this is only about the management of information such as the sections, inventory, personnel, archives and reports and all the other processes that are not covered and irrelevant to the system is out of our scope and out of our bounds to be discussed. Transactions such as borrowing and returning of items can be added upon request from PFMO chief.

**CHAPTER 2****SYSTEM ANALYSIS****A. DATA DICTIONARY****SectionTable**

<b>Field Data</b>	<b>Field Type</b>	<b>Length</b>	<b>Description</b>
<b>SectionID</b>	<b>Guid</b>	<b>32</b>	<b>Section Code</b>
<b>SectionName</b>	<b>Text</b>	<b>200</b>	<b>Name of Section</b>
<b>SectionType</b>	<b>Text</b>	<b>200</b>	<b>Type of Section</b>
<b>SectionDescription</b>	<b>Text</b>	<b>200</b>	<b>Description of Section</b>
<b>SectionDateCreated</b>	<b>DateTime</b>	<b>20</b>	<b>Date of the creation of Section</b>

**SectionHeadTable**

<b>Field Data</b>	<b>Field Type</b>	<b>Length</b>	<b>Description</b>
<b>SectionHeadID</b>	<b>Guid</b>	<b>32</b>	<b>Section Head Code</b>
<b>SectionHeadLastName</b>	<b>Text</b>	<b>200</b>	<b>Last name of Section Head</b>
<b>SectionHeadFirstName</b>	<b>Text</b>	<b>200</b>	<b>First name of Section Head</b>
<b>SectionHeadGender</b>	<b>Text</b>	<b>10</b>	<b>Gender of Section Head</b>
<b>SectionHeadDesignationID</b>	<b>Guid</b>	<b>32</b>	<b>Section Head Designation Code</b>
<b>SectionHeadRemarks</b>	<b>Text</b>	<b>200</b>	<b>Remarks of the Section Head</b>
<b>SectionID</b>	<b>Guid</b>	<b>32</b>	<b>Section Code</b>
<b>SectionHeadHireDate</b>	<b>DateTime</b>	<b>20</b>	<b>Hire date of the Section Head</b>
<b>SectionHeadContactNumber</b>	<b>Number</b>	<b>20</b>	<b>Contact number of Section Head</b>

**SectionTaskTable**

Field Data	Field Type	Length	Description
SectionTaskID	Guid	32	Section Task Code
SectionTaskTitle	Text	200	Task title of Section
SectionTaskDescription	Text	200	Task description of the Section
SectionID	Guid	32	Section code
SectionTaskStartDate	DateTime	20	Start Date of the Task of the Section
SectionTaskEndDate	DateTime	20	End Date of the Task of the Section

**SectionHeadDesignationTable**

Field Data	Field Type	Length	Description
SectionHeadDesignationID	Guid	32	Section Head Designation code
SectionHeadDesignationName	Text	200	Name of the Designation for the Section Head
SectionHeadDesignationDescription	Text	200	Description of the Designation for the Section Head

**PersonnelTable**

Field Data	Field Type	Length	Description
PersonnelID	Guid	32	Personnel code
PersonnelLastName	Text	200	Last name of Personnel
PersonnelFirstName	Text	200	First name of Personnel
PersonnelGender	Text	10	Gender of Personnel
PersonnelDesignationID	Guid	32	Personnel Designation code
PersonnelRemarks	Text	200	Remarks of the Personnel
SectionID	Guid	32	Section code
ProjectTeamID	Guid	32	Project Team code
PersonnelHireDate	DateTime	20	Hire date of the Personnel
PersonnelContactNumber	DateTime	20	Contact number of the personnel

**PersonnelDesignationTable**

Field Data	Field Type	Length	Description
PersonnelDesignationID	Guid	32	Personnel Designation code
PersonnelDesignationName	Text	200	Name of the Designation for the Personnel
PersonnelDesignationDescription	Text	200	Description of the Designation for the Personnel

**PersonnelTaskTable**

Field Data	Field Type	Length	Description
PersonnelTaskID	Guid	32	Personnel Task code
PersonnelTaskTitle	Text	200	Title of the Personnel task
PersonnelTaskDescription	Text	200	Description of the Personnel task
PersonnelID	Guid	32	Personnel code
PersonnelTaskStartDate	DateTime	20	Start date of the Personnel task
PersonnelTaskEndDate	DateTime	20	End date of the Personnel task

**ProjectTeamTable**

Field Data	Field Type	Length	Description
ProjectTeamID	Guid	32	Project Team code
ProjectTeamName	Text	200	Name of Project Team
ProjectTeamDescription	Text	200	Description of Project Team
ProjectID	Guid	32	Project code

**ProjectTable**

Field Data	Field Type	Length	Description
ProjectID	Guid	32	Project code
ProjectName	Text	200	Name of project
ProjectStatus	Text	200	Status of the project
ProjectCompletion	Integer	100	Completion of the project
ProjectStartDate	DateTime	20	Start date of the project
ProjectEndDate	DateTime	20	End date of the project

**SupplyTable**

Field Data	Field Type	Length	Description
SupplyID	Guid	32	Supply code
SupplyCode	Text	200	Code of the item supply
SupplyName	Text	200	Name of supply
SupplyCategoryID	Guid	32	Supply Category code
SupplyRemarks	Text	200	Remarks of the supply
SupplyStatus	Text	200	Status of the Supply
SupplyBalance	Integer	200	Remaining balance of the supply
SupplyUnit	Text	200	Unit of measure of the supply
SupplyReorderPoint	Integer	200	Reorder point of the supply
Target	Text	200	Target for the supply
ProjectID	Guid	32	Project code
SupplyExpiryDate	DateTime	20	Expiry date of the supply

**SupplyCategoryTable**

Field Data	Field Type	Length	Description
SupplyCategoryID	Guid	32	Supply category code
SupplyCategoryName	Text	200	Name of the supply category
SupplyCategoryDescription	Text	200	Description of the supply category

**EquipmentTable**

Field Data	Field Type	Length	Description
EquipmentID	Guid	32	Equipment code
EquipmentCode	Text	200	Code of the equipment
EquipmentName	Text	200	Name of equipment
EquipmentCategoryID	Guid	32	Equipment Category code
EquipmentRemarks	Text	200	Remarks of equipment
EquipmentStatus	Text	200	Status of the equipment
EquipmentBalance	Integer	200	Remaining balance of the item
EquipmentUnit	Text	200	Unit of measure of the item
EquipmentReorderPoint	Integer	200	Reorder point of the equipment
Target	Text	200	Target for equipment
SupplyExpiryDate	DateTime	20	Expiry supply date

**EquipmentCategoryTable**

Field Data	Field Type	Length	Description
EquipmentCategoryID	Guid	32	Equipment Category code
EquipmentCategoryName	Text	200	Category name of equipment
EquipmentCategoryDescription	Text	200	Description of equipment category

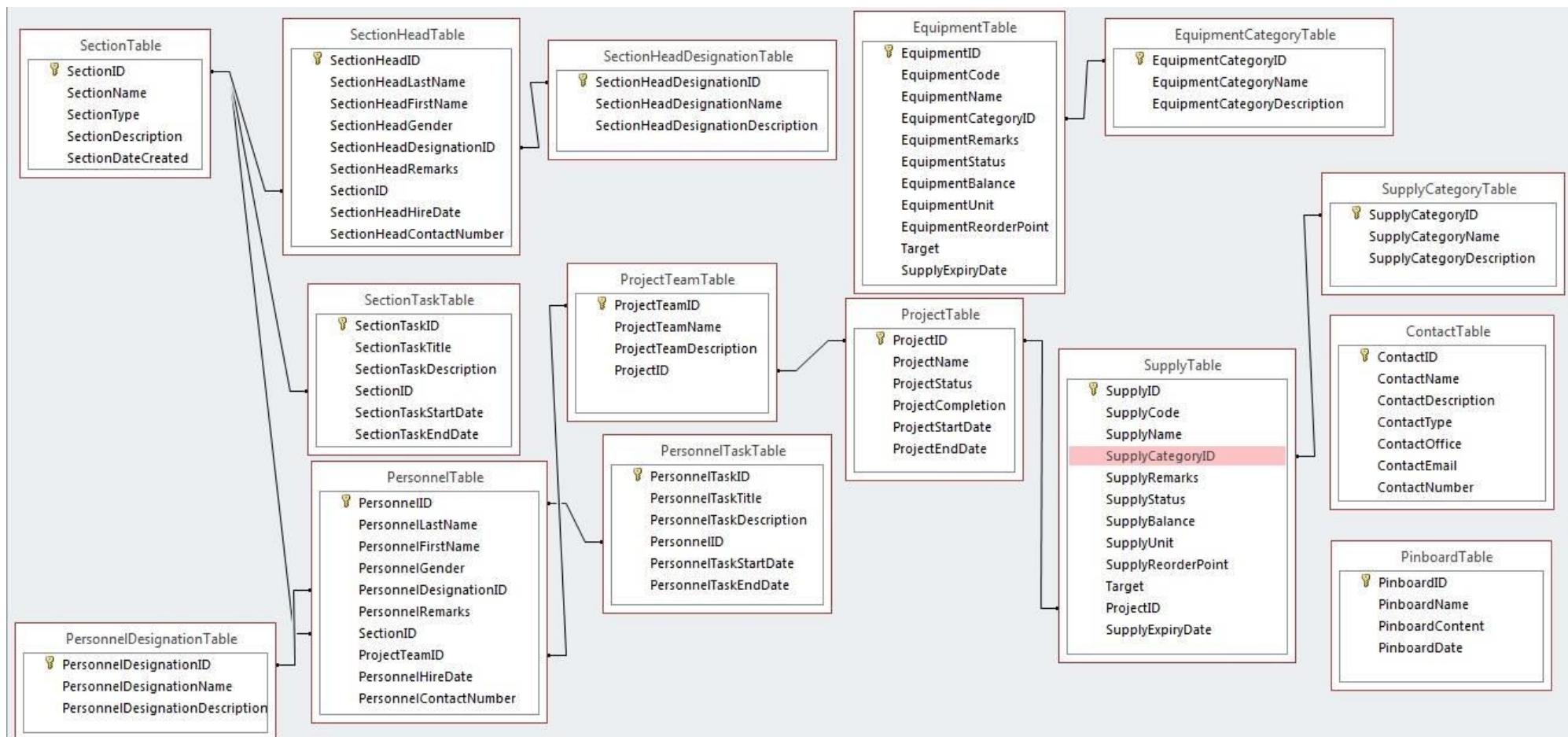
**ContactTable**

<b>Field Data</b>	<b>Field Type</b>	<b>Length</b>	<b>Description</b>
ContactID	Guid	32	Contact code
ContactName	Text	50	Name of the contact person
ContactDescription	Text	100	Description of the contact person
ContactType	Text	30	Type of contact person
ContactOffice	Text	50	Office of the contact person
ContactEmail	EmailAddress	50	Email address of the contact person
ContactNumber	PhoneNumber	20	Phone number of the contact person

**PinboardTable**

<b>Field Data</b>	<b>Field Type</b>	<b>Length</b>	<b>Description</b>
PinboardID	Guid	32	Pinboard code
PinboardName	Text	200	Name of the Pinboard
PinboardContent	Text	200	Content of the Pinboard
PinboardDate	DateTime	20	Date of the Pinboard

## B. RELATIONAL DATABASE



## C.NORMALIZATION

**NORMALIZATION-** Database normalization (or normalization) is the process of organizing the columns (attributes) and tables (relations) of a relational database to minimize data redundancy.

**UNF**

Section
Section Head
Section Task
Section Designation
Personnel
Personnel Task
Personnel Designation
Project
Project Team
Supply
Supply Category
Equipment
Equipment Category

**1NF**

Section
Section Head
Section Task
Section Designation
Personnel
Personnel Task
Personnel Designation
Project
Project Team
Supply
Supply Category
Equipment
Equipment Category

**2NF**

Section_Code
Section_Head
Section_Task
Section_Designation

Equipment
Equipment_Category

Section_Code
Personnel
Personnel_Task
Personnel_Designation

Project_Code
Project_Team
Project

Project_Code
Supply
Supply_Category

**3NF**

**Section\_TBL      Section\_Head\_TBL      Section\_Task\_TBL      Section\_Head\_Designation\_TBL**

ID
Name
Type
Description
Date Created

ID
Last_Name
First_Name
Gender
Designation_ID
Remarks
Section_ID
Hire_Date
Contact_No

ID
Title
Description
Start_Date
End_Date

ID
Name
Description

**Personnel\_TBL      Personnel\_Designation\_TBL      Personnel\_Task\_TBL      Project\_Team\_TBL**

ID
Last_Name
First_Name
Gender
Designation_ID
Remarks
Section_ID
Project_Team_ID
Hire_Date
Contact_No

ID
Name
Description

ID
Title
Description
Personnel_ID
Start_Date
End_Date

ID
Name
Description
Project_ID

**Project\_TBL**      **Supply\_TBL**      **Supply\_Category\_TBL**      **Equipment\_TBL**

ID
Name
Status
Completion
Start_Date
End_Date

ID
Code
Name
Category_ID
Remarks
Status
Balance
UNIT
ReOrder_Point
Target
Project_ID
Expiry_Date

ID
Name
Description

ID
Code
Name
Category_ID
Remarks
Status
Balance
UNIT
ReOrder_Point
Target
Supply_Expiry_Date

**Equipment\_Category\_TBL**

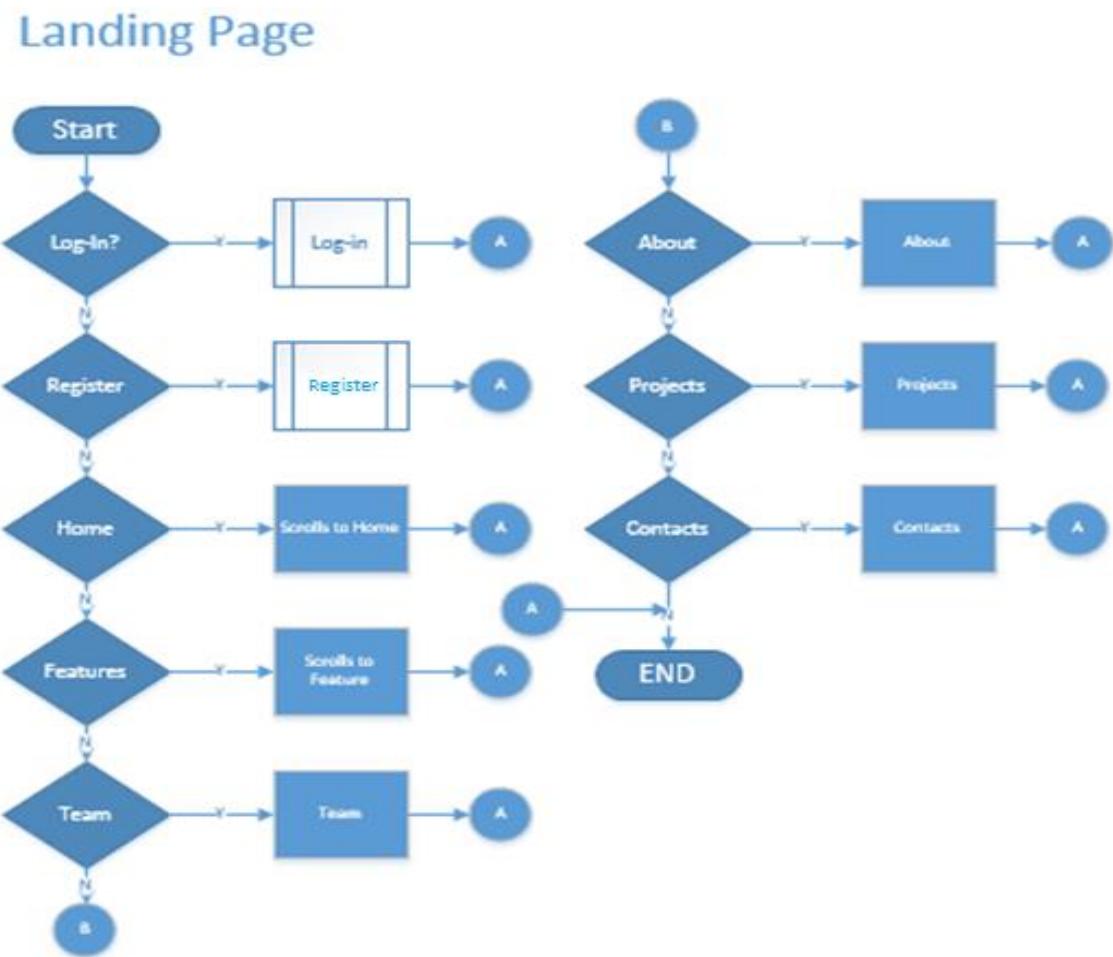
ID
Name
Description

**Contact\_TBL**

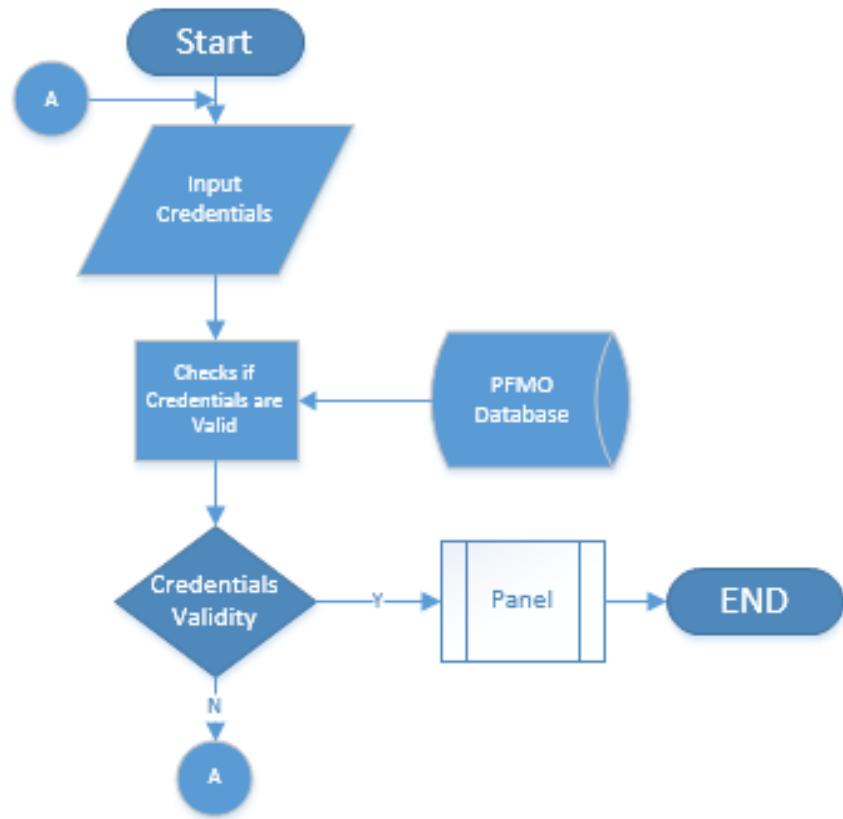
ID
Name
Description
Type
Office
Email
Contact_Number

**Pinboard\_TBL**

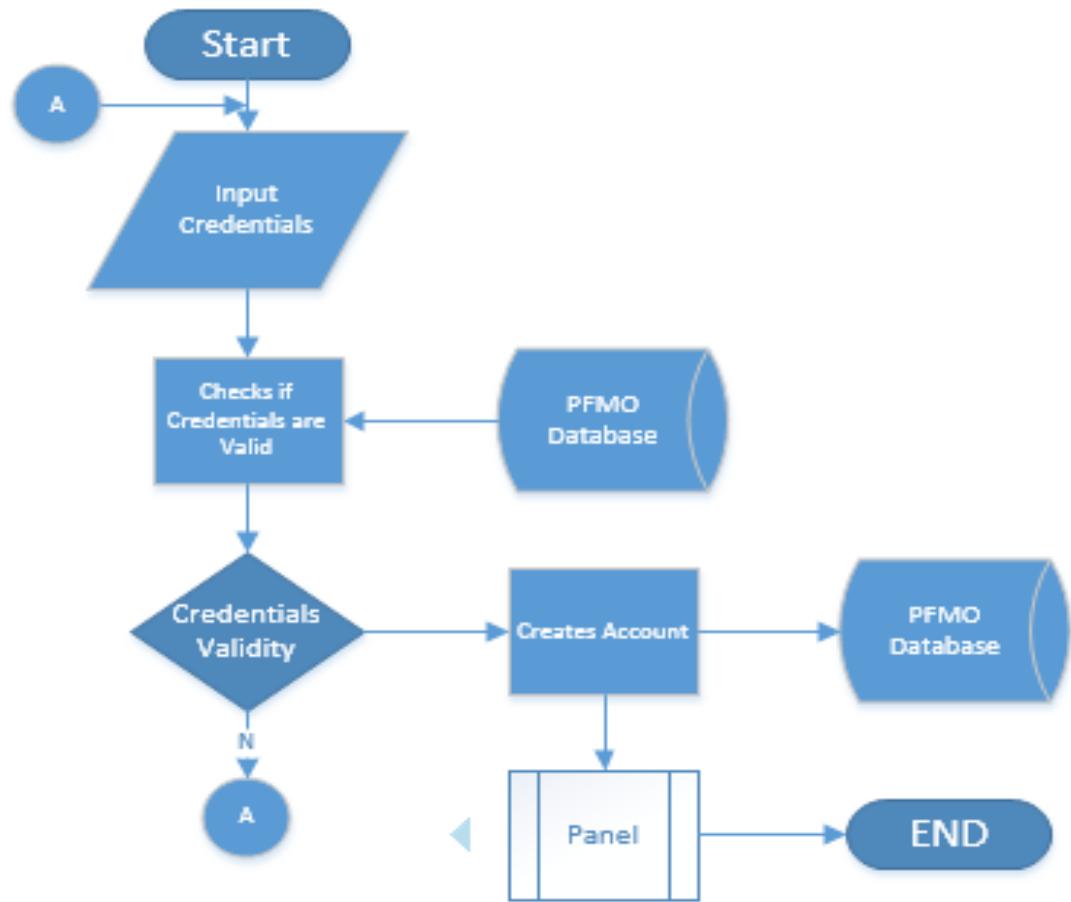
ID
Name
Content
Date

**CHAPTER 3****GENERAL DESIGN****A. System Flowchart**

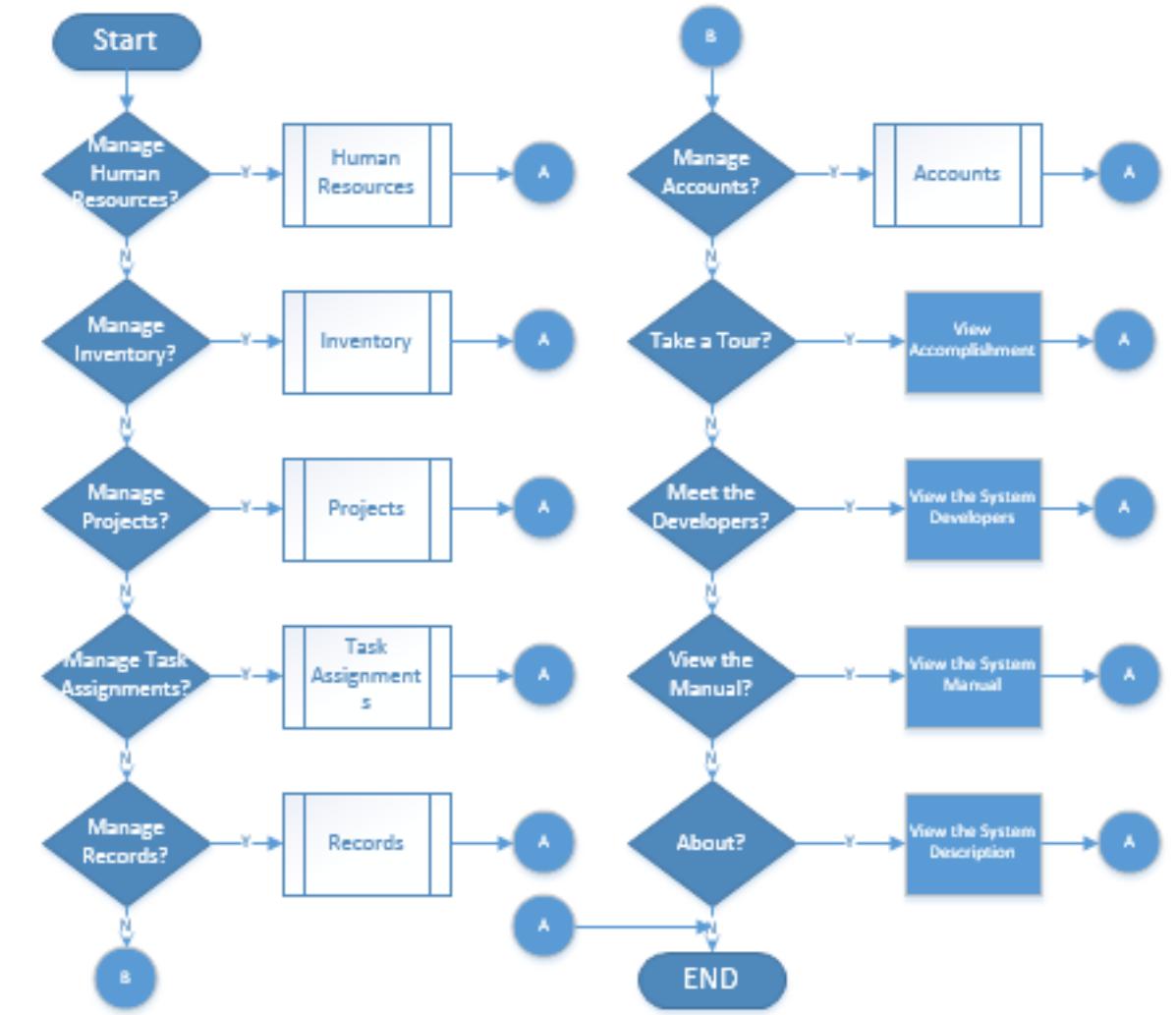
# Log-In



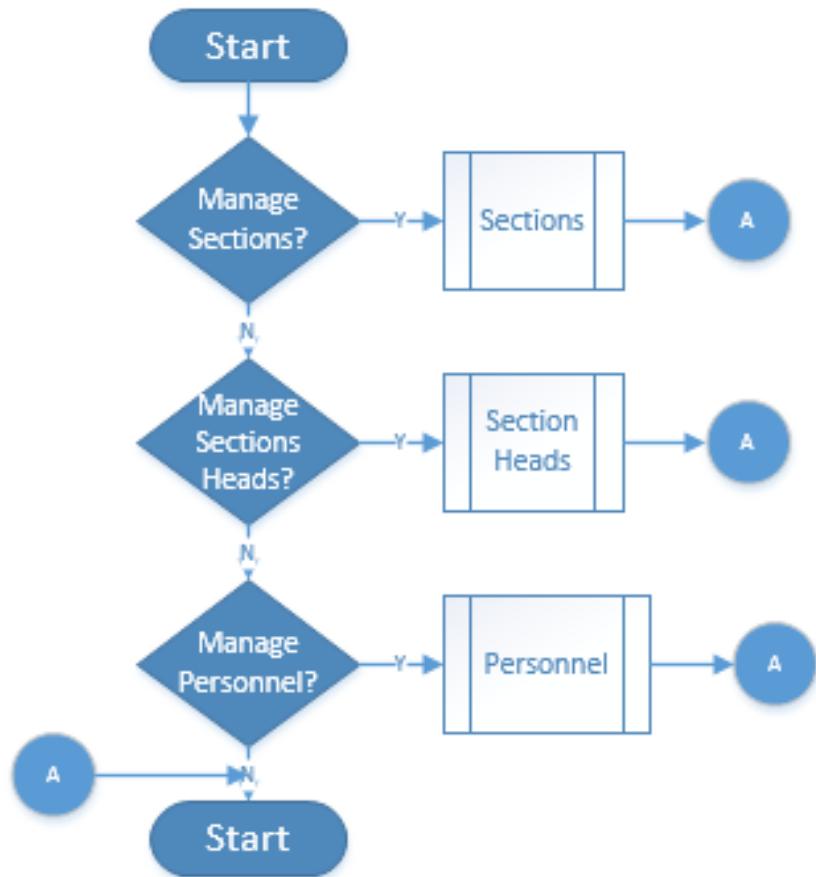
# Register



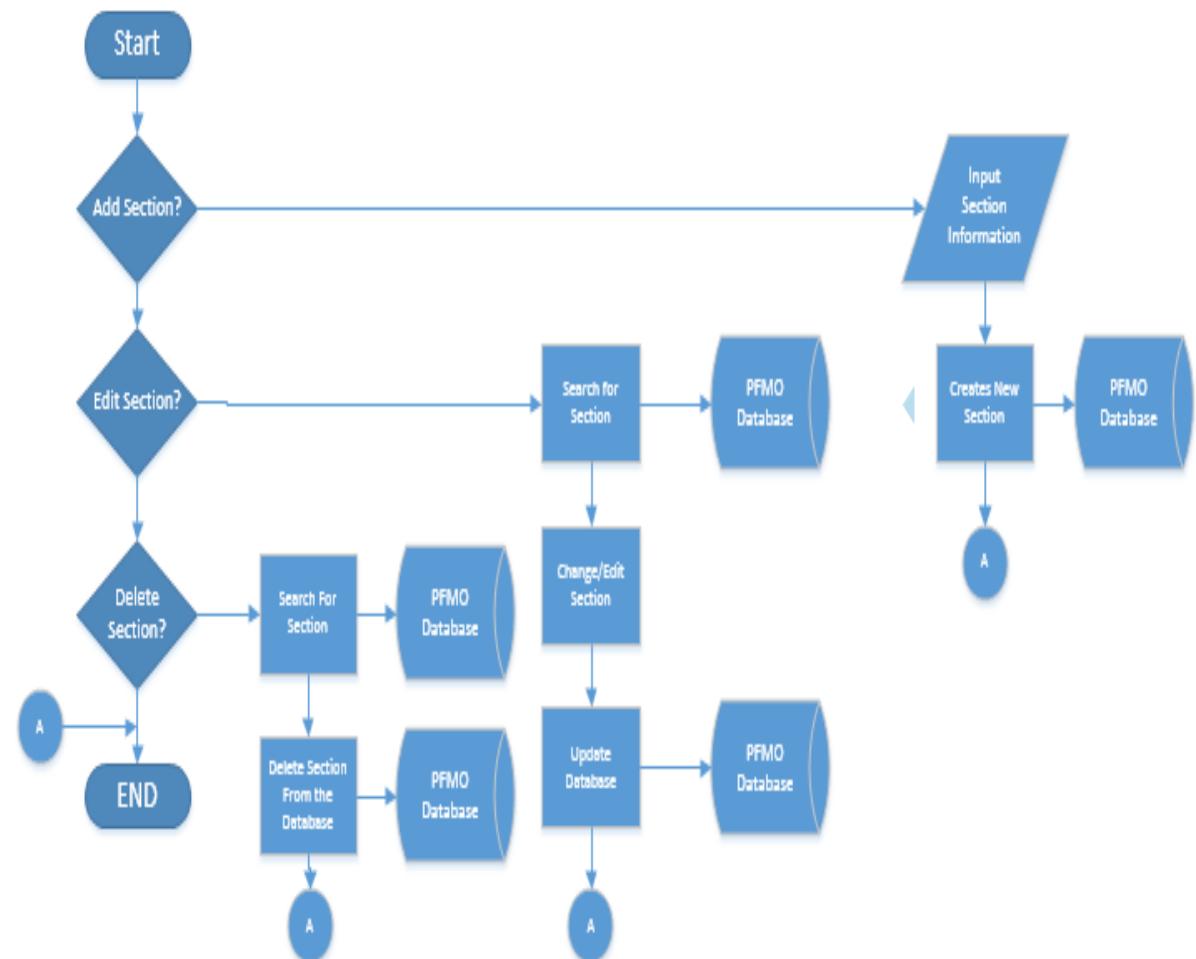
# Panel



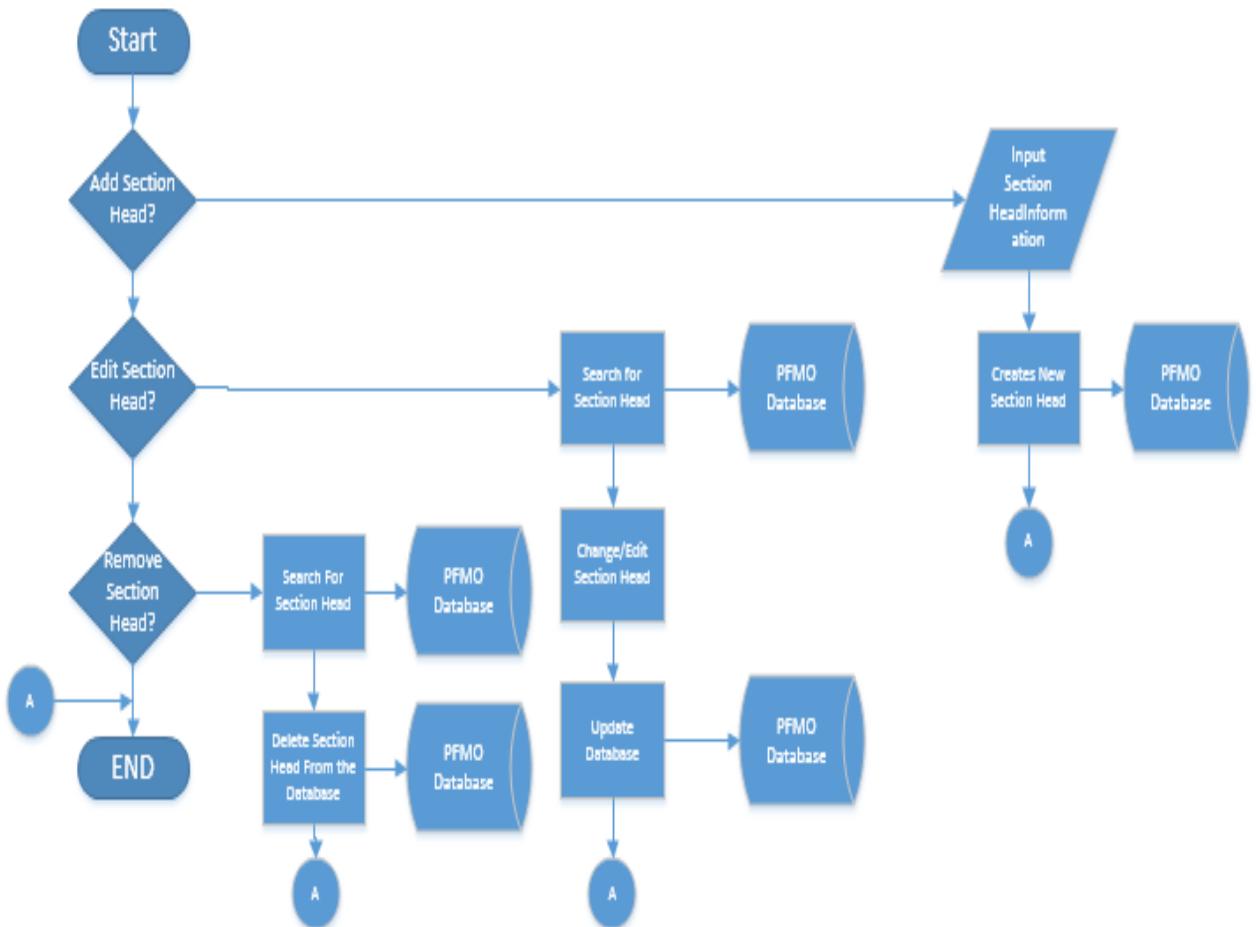
# Human Resources



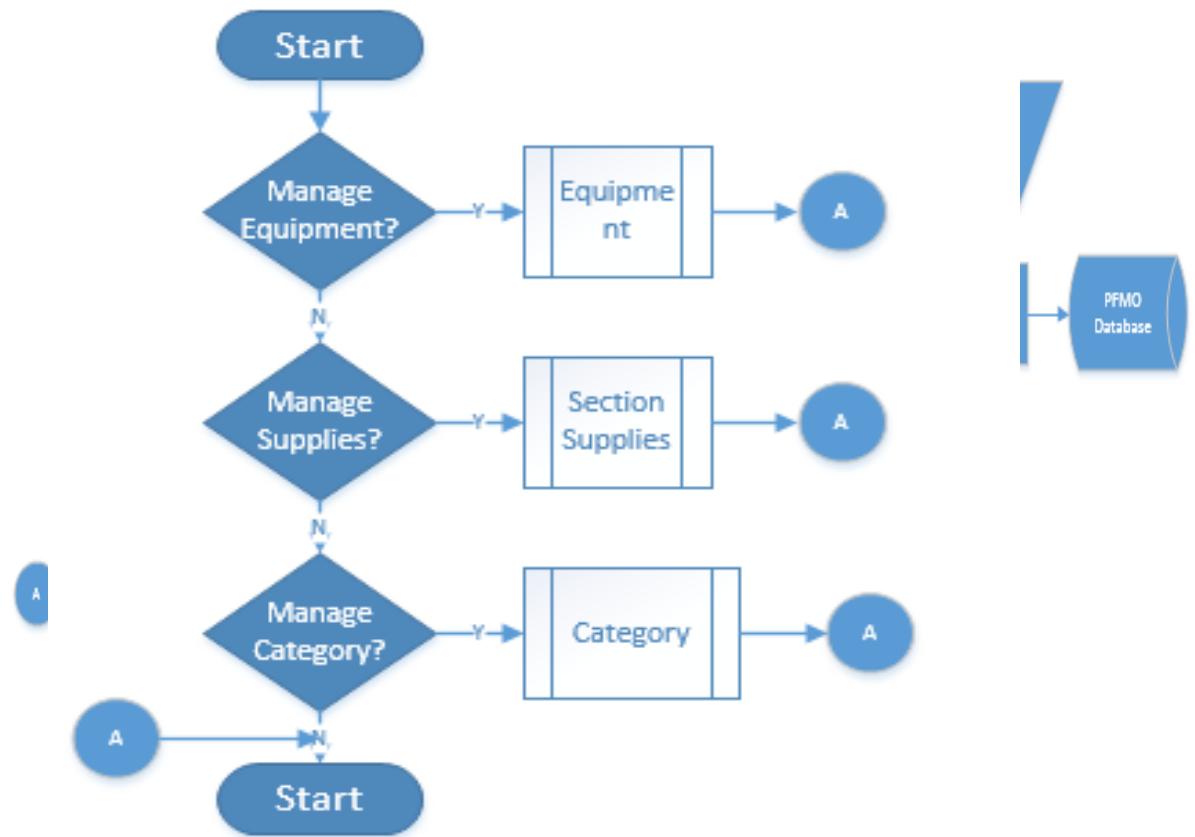
## Sections



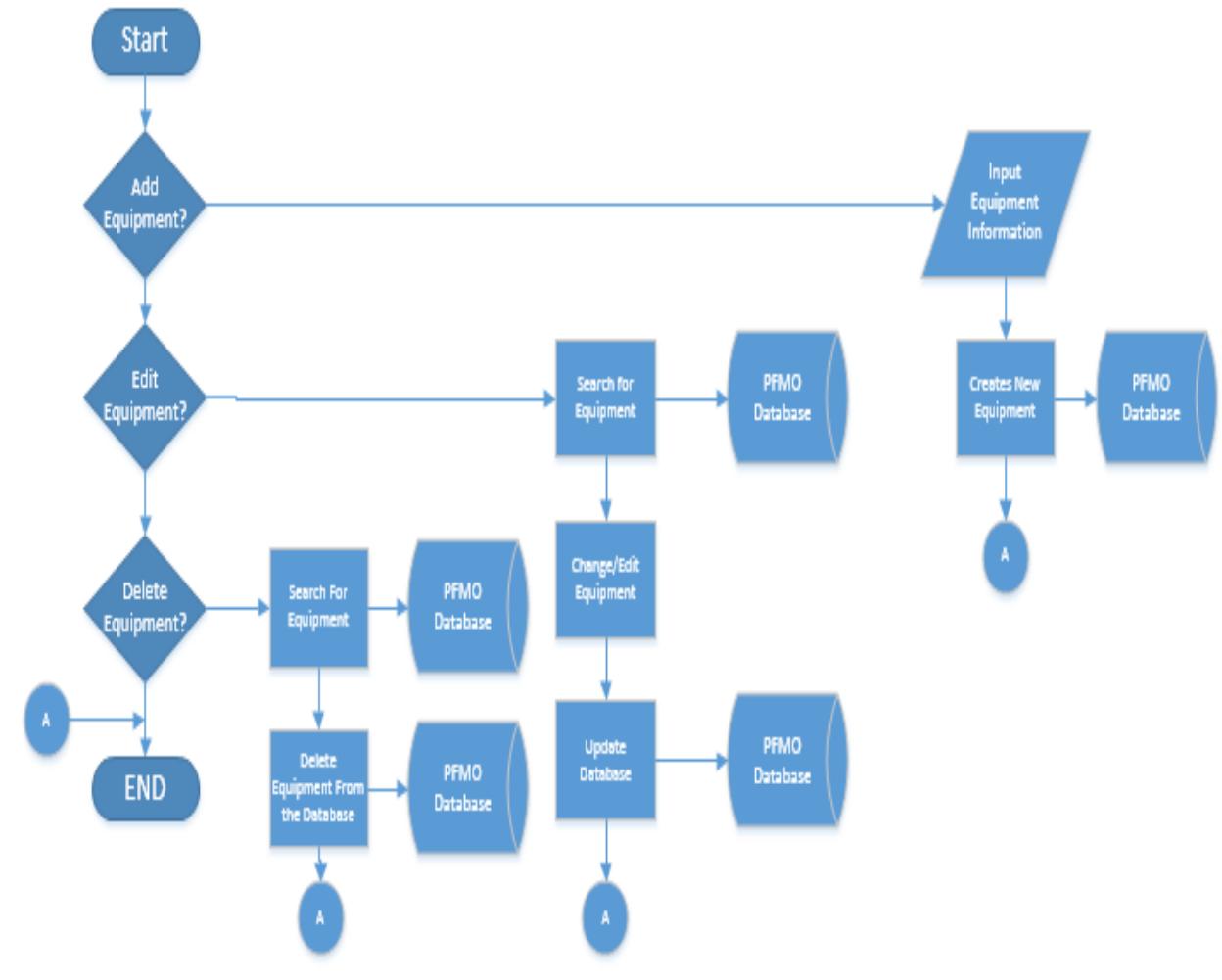
## Section Heads



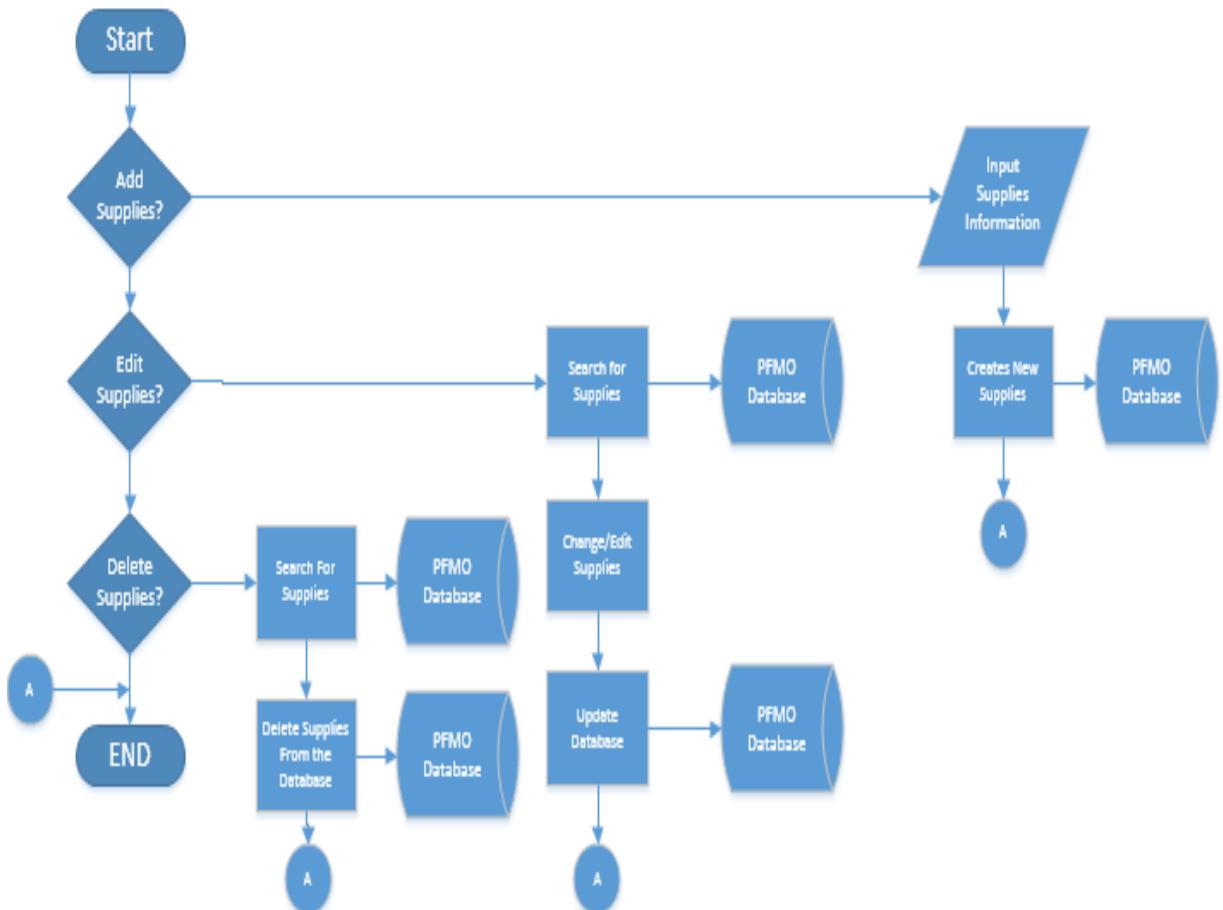
# Inventory



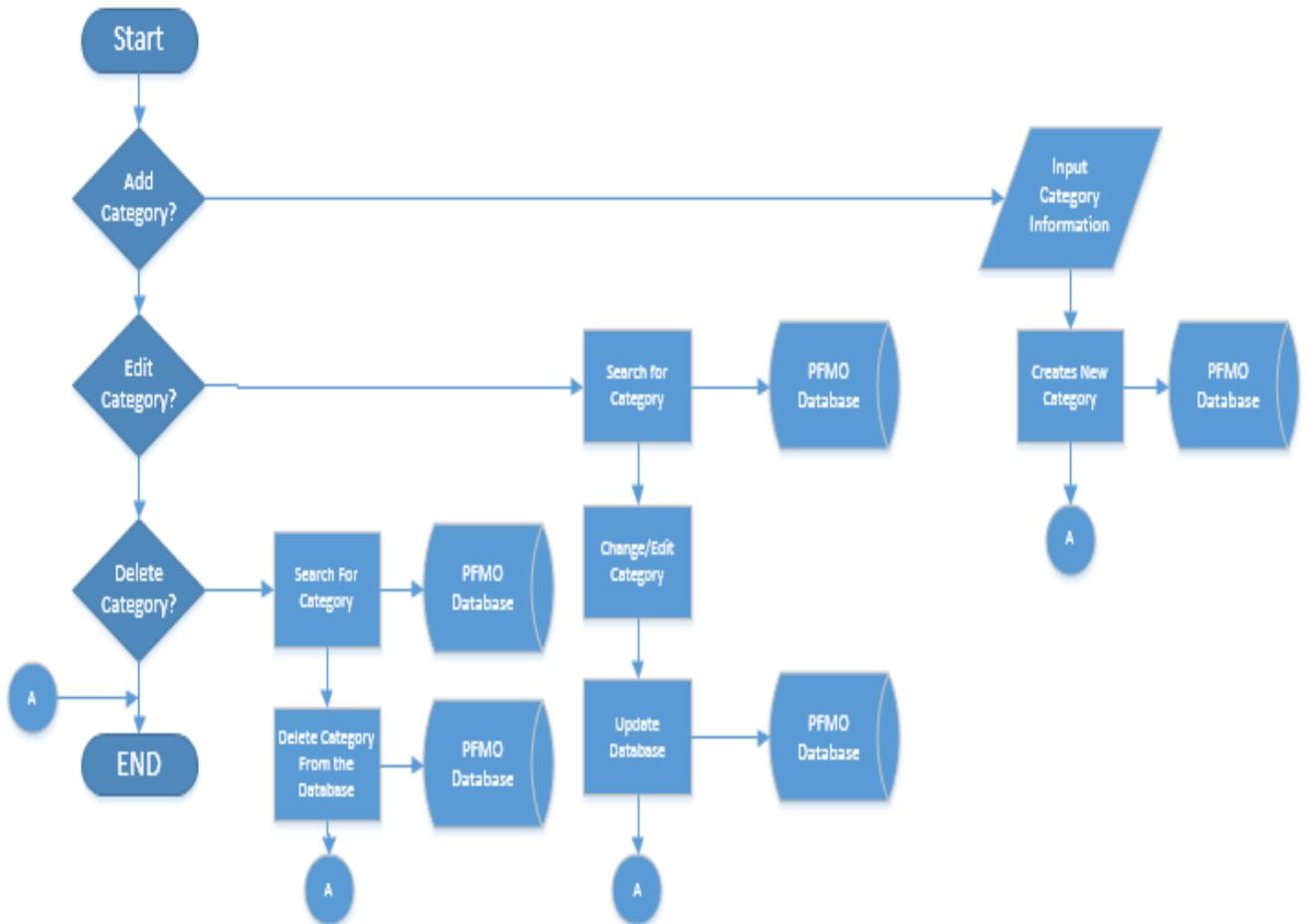
# Equipment



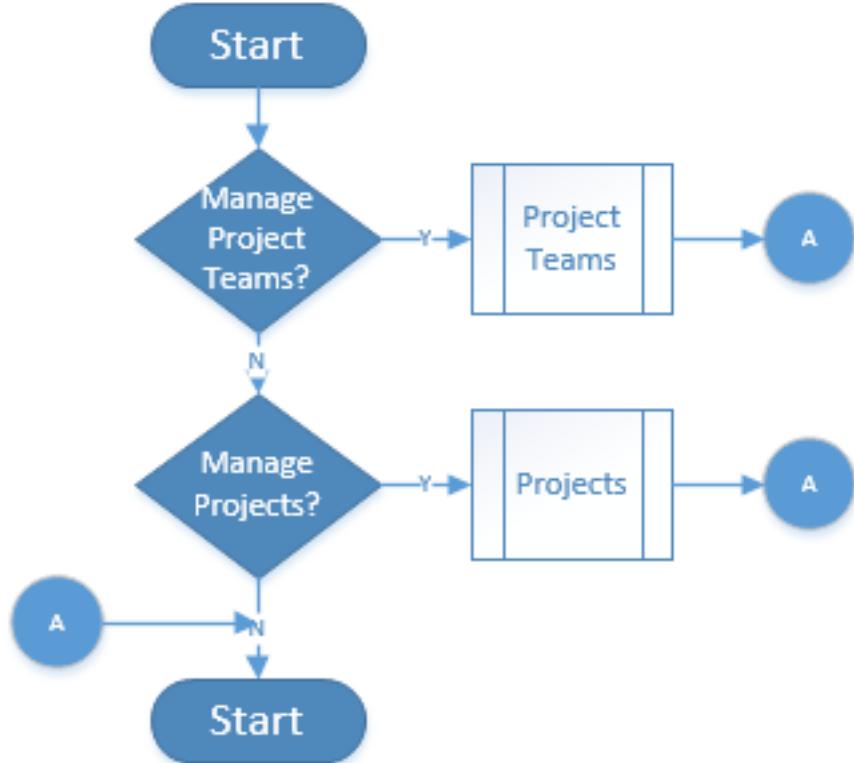
## Supplies



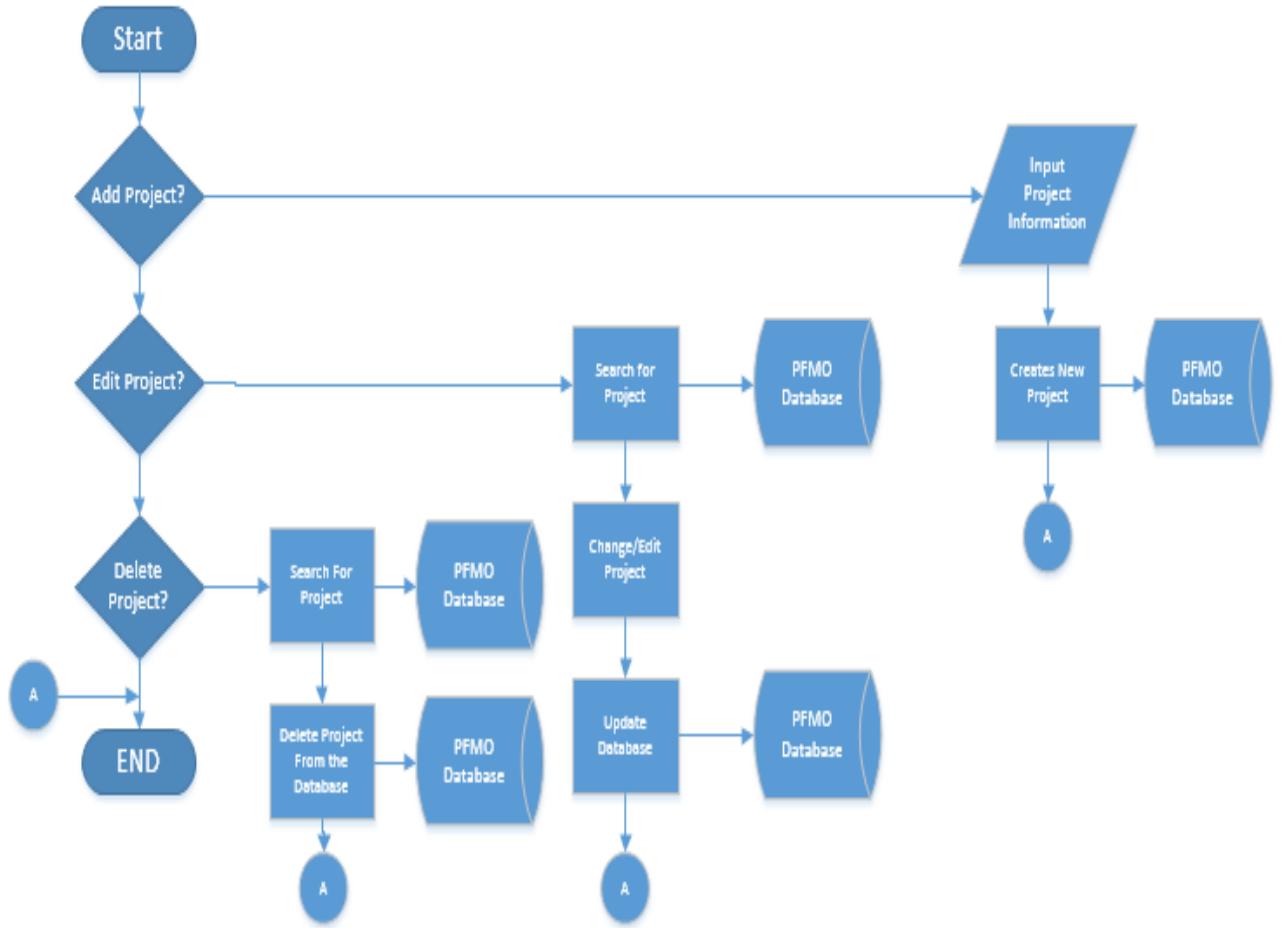
## Category



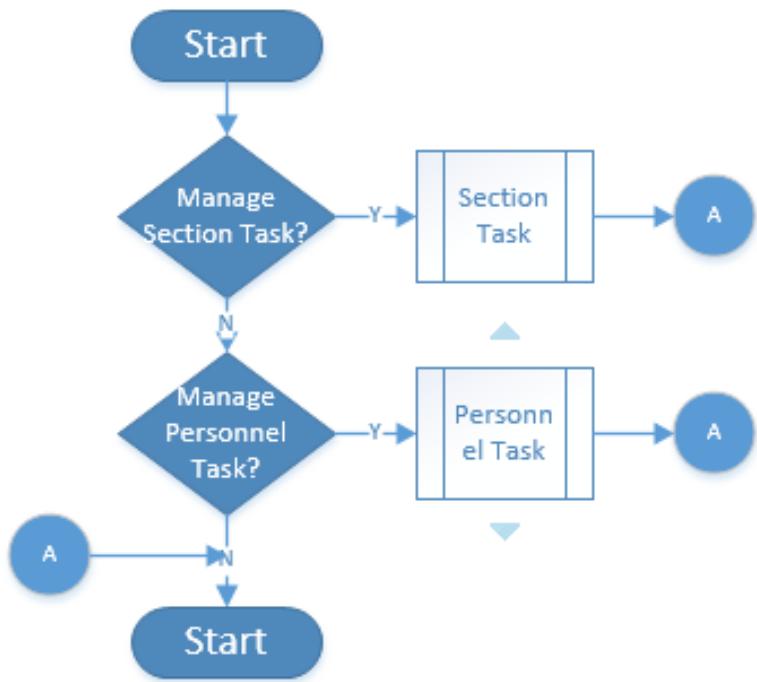
# Projects



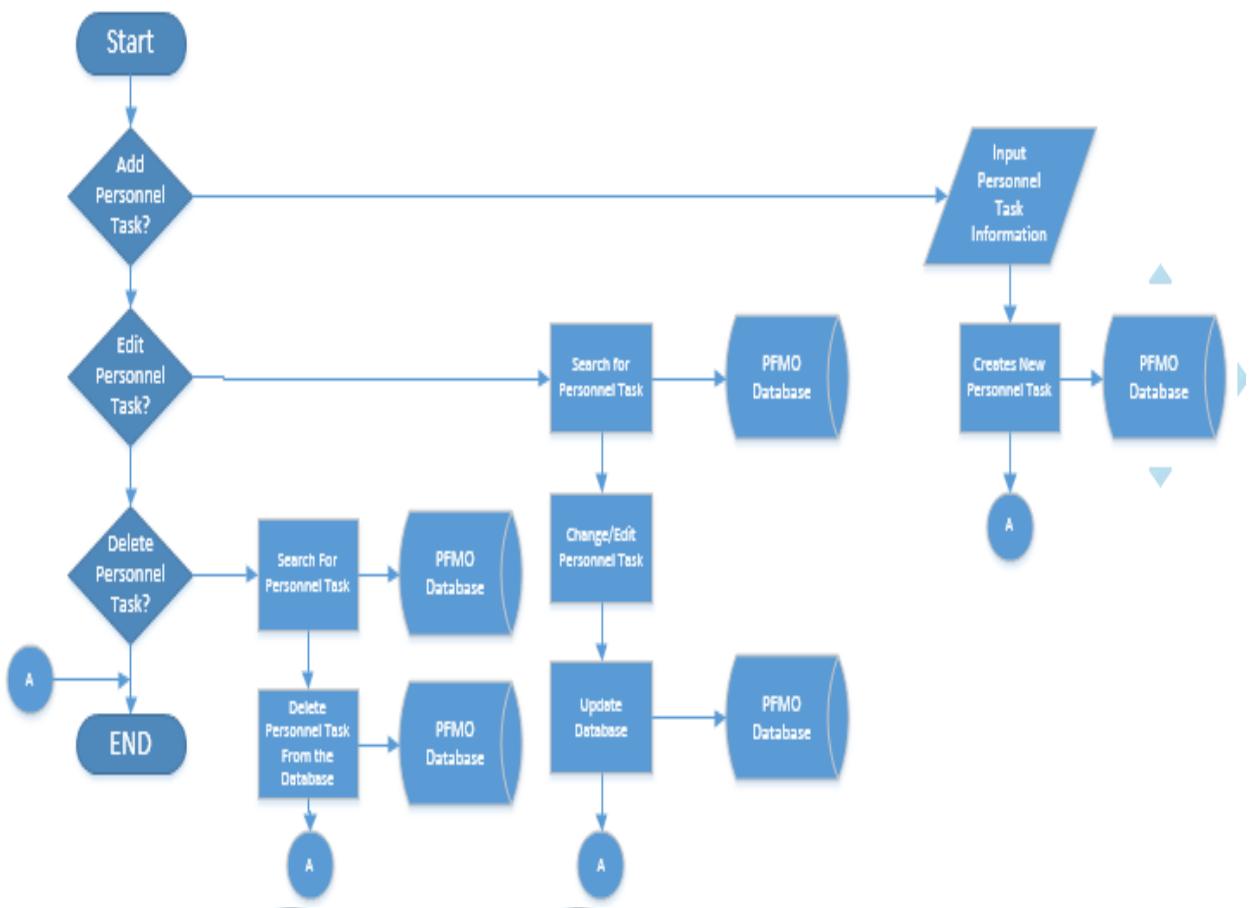
## Projects



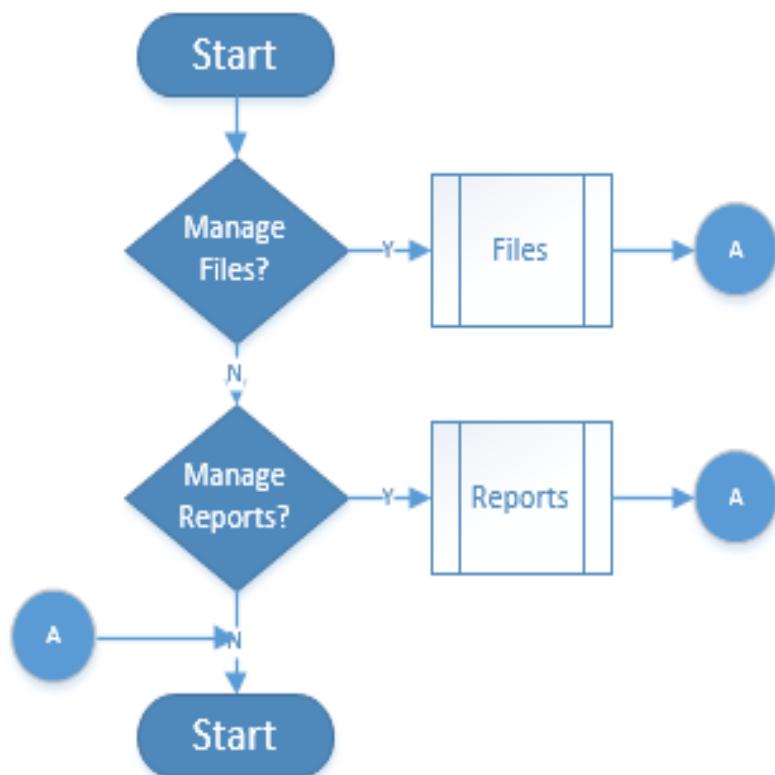
# Task Assignments



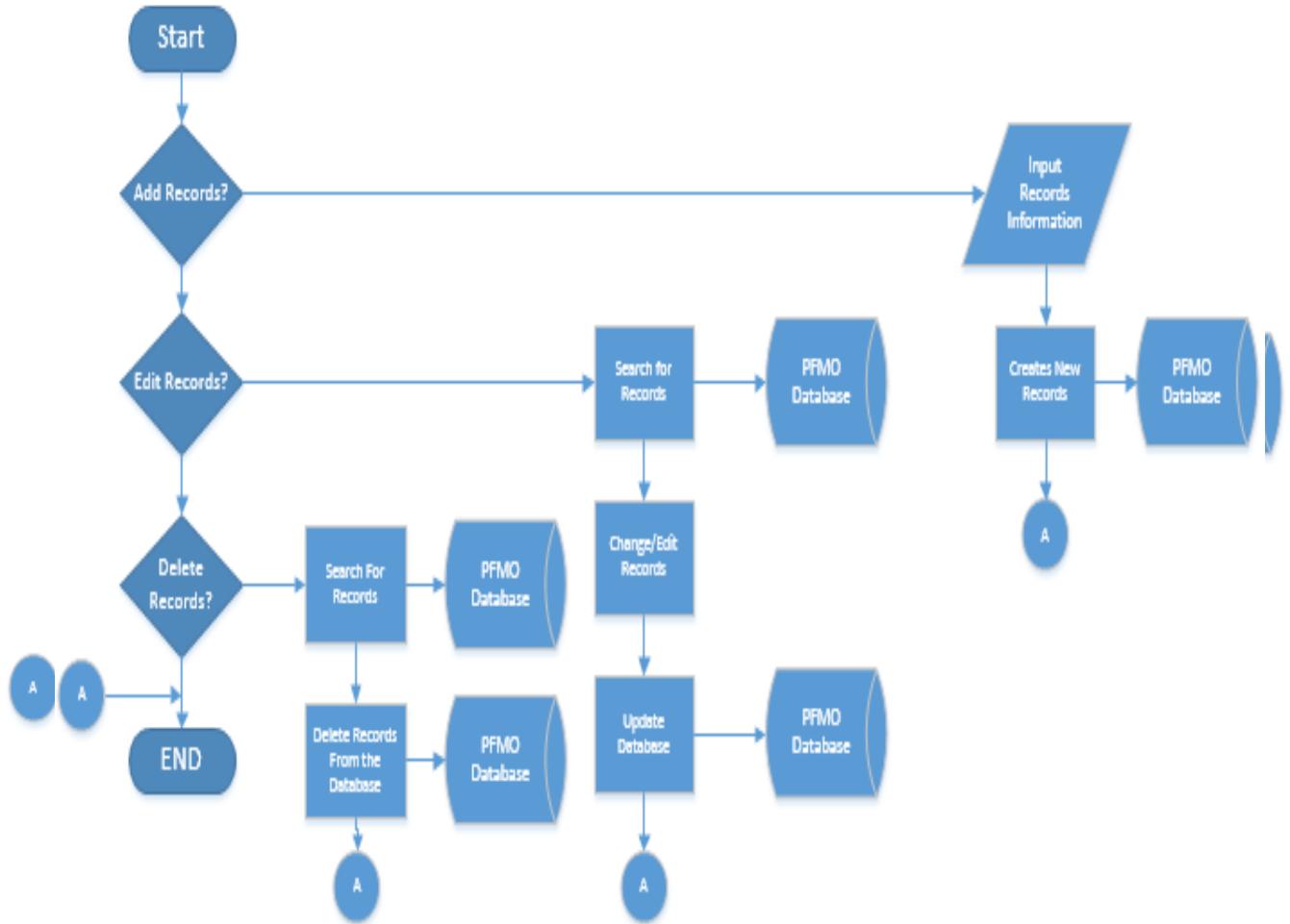
## Personnel Task



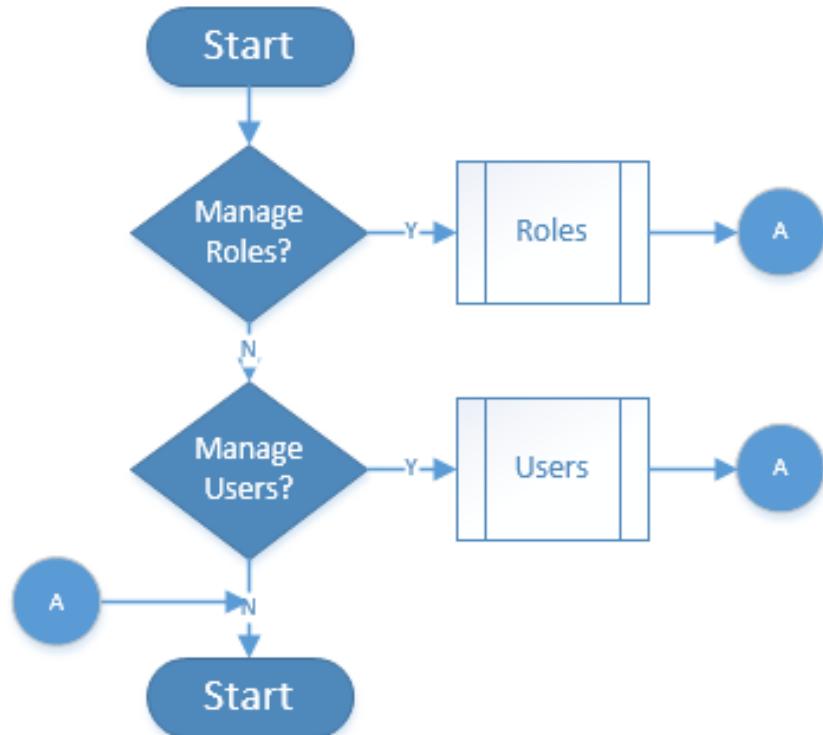
# Records



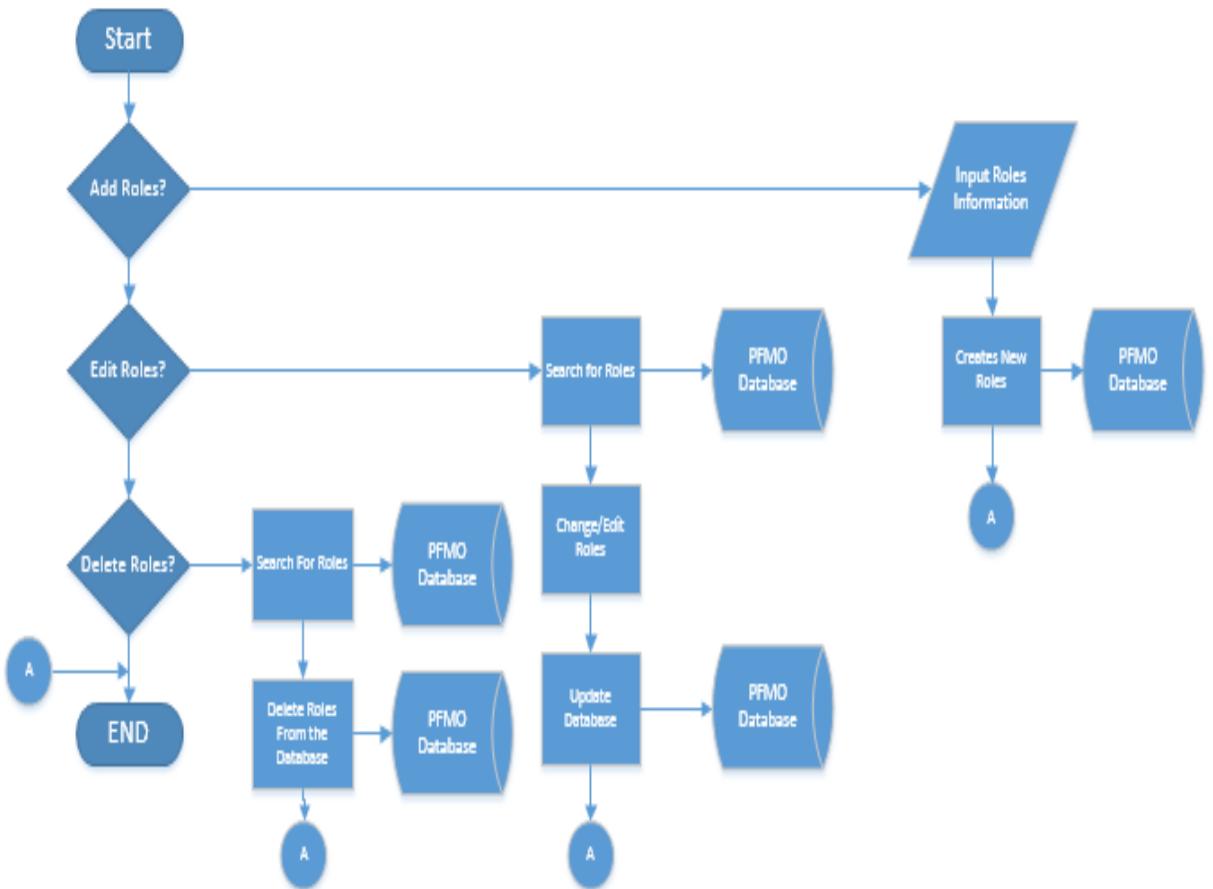
## Records

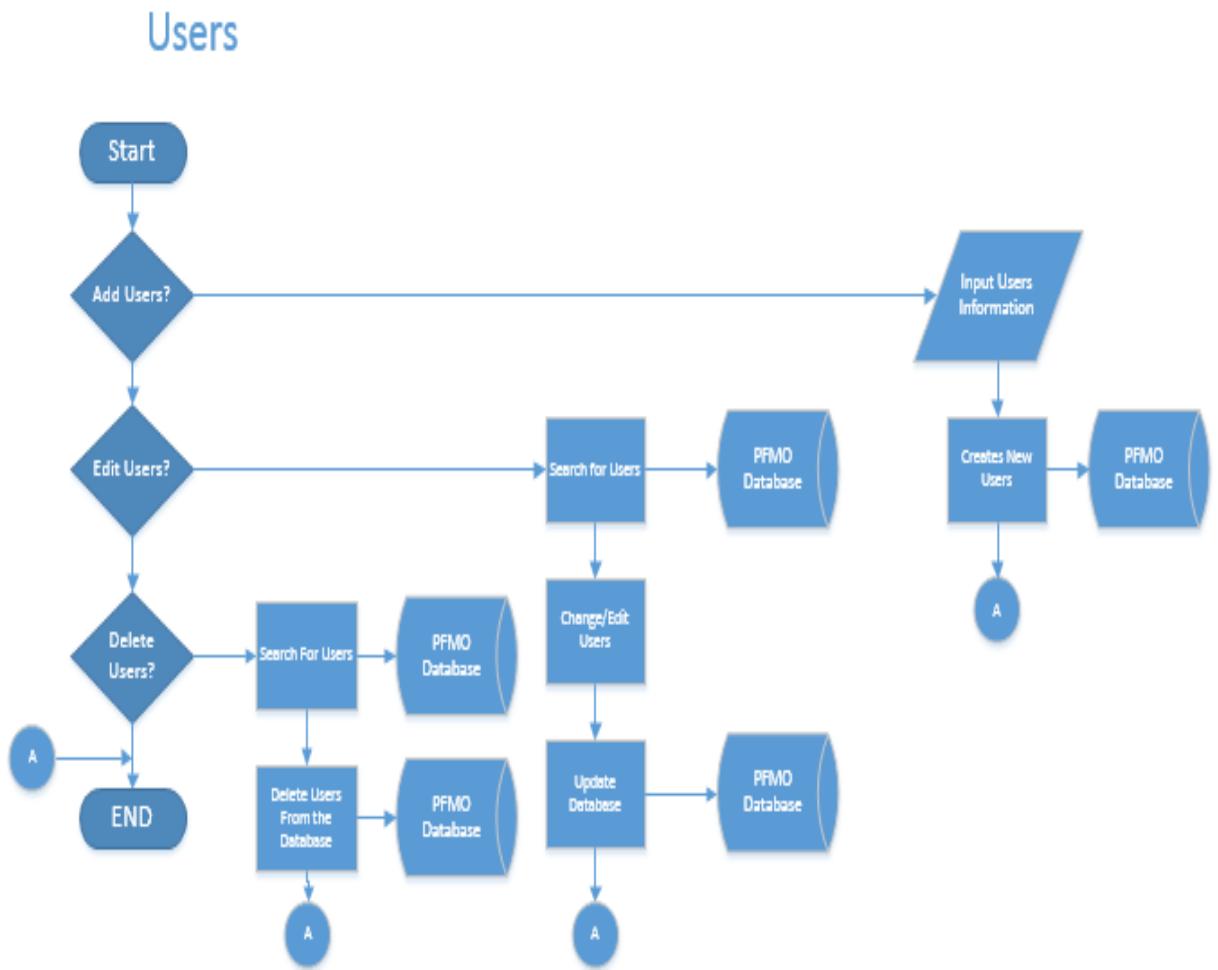


# Account

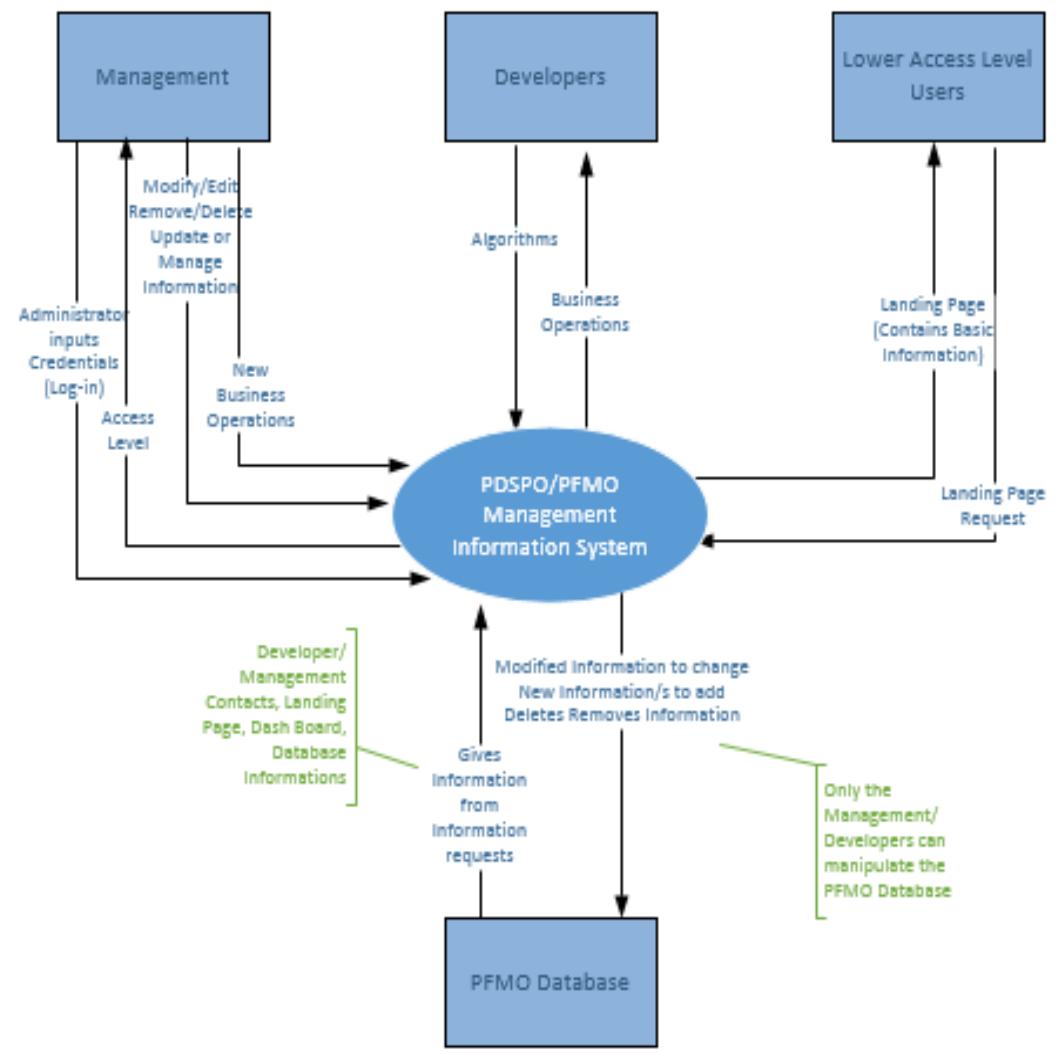


## Roles

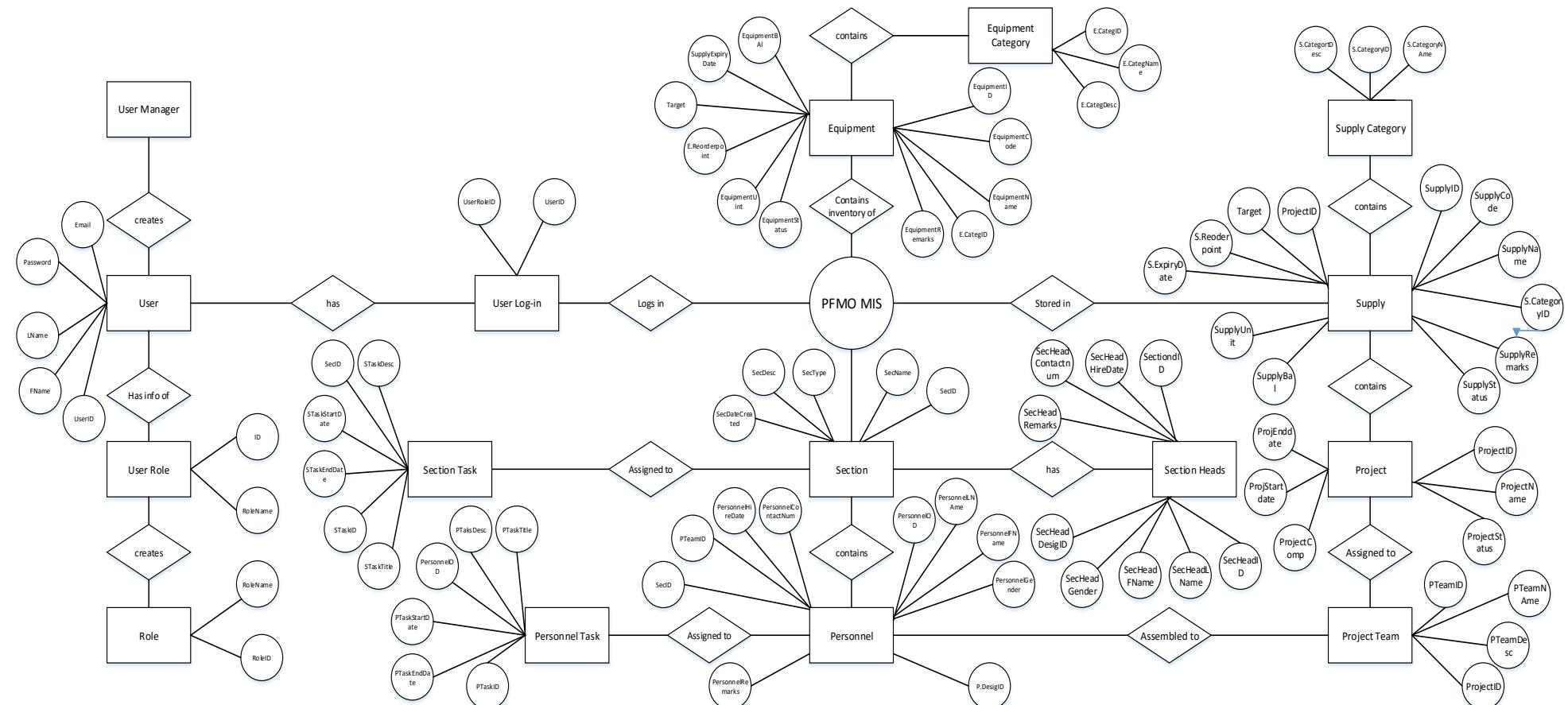




## B. Context Diagram

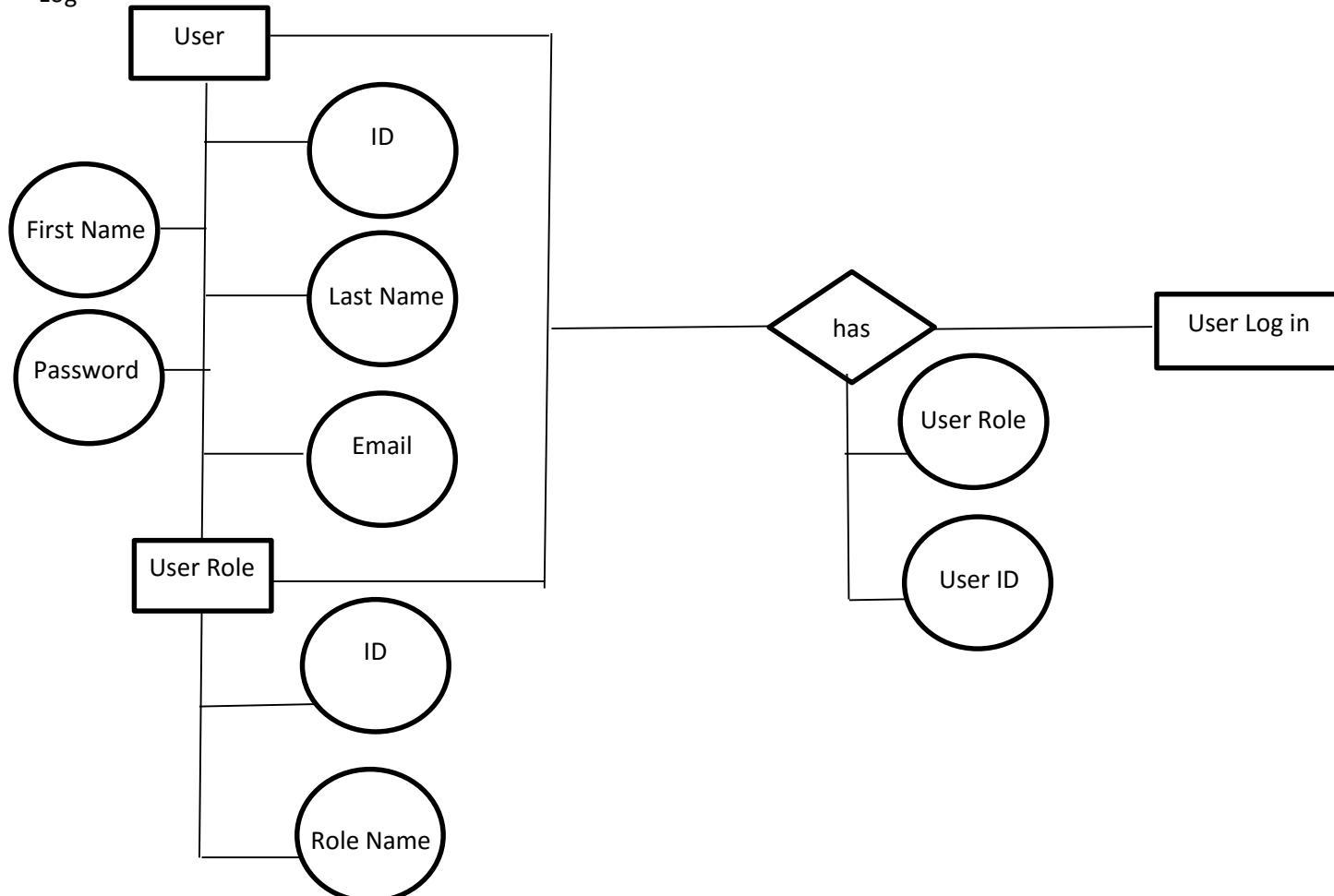


## C. Entity Relationship Diagram (AS A WHOLE)

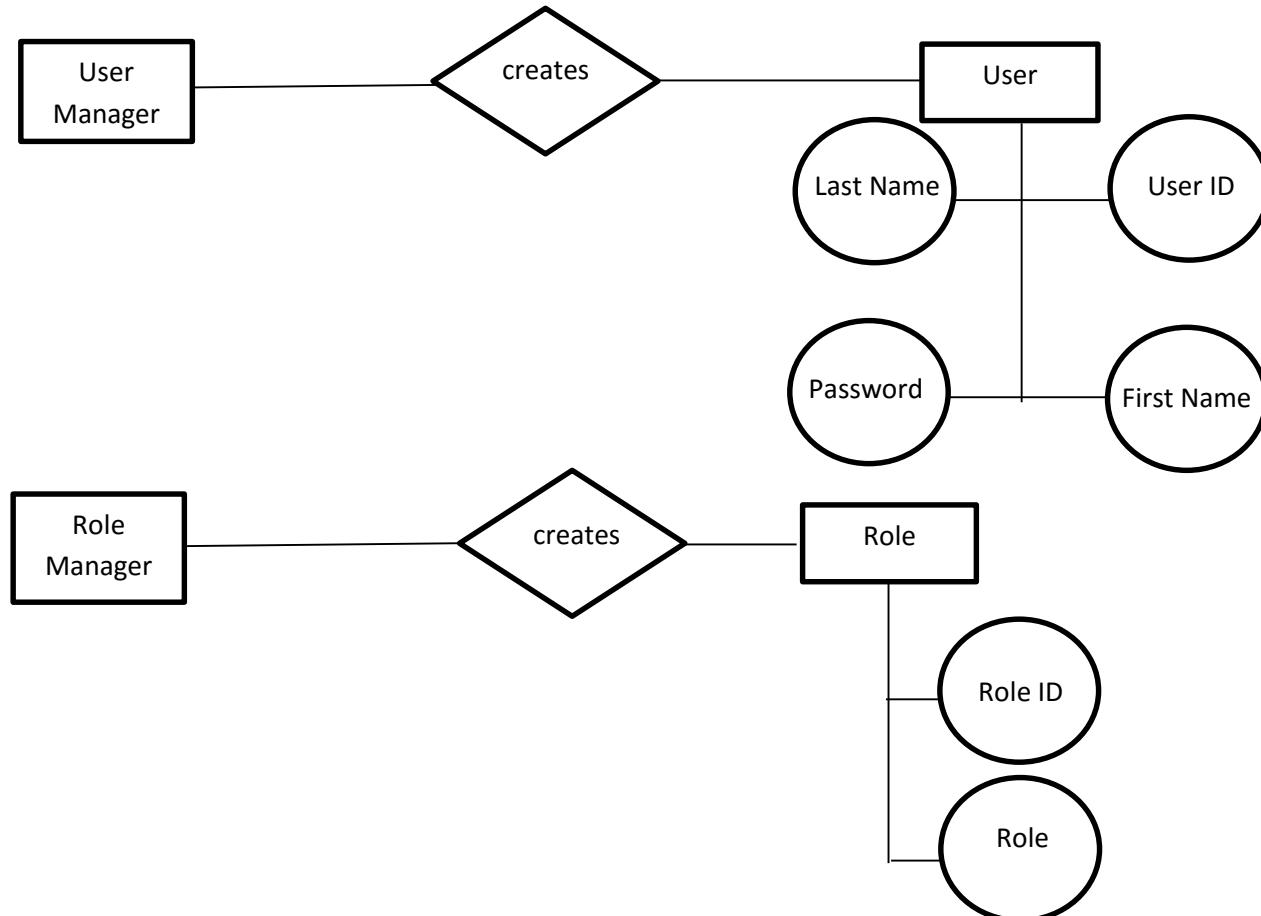


## Entity Relationship Diagram (PER TRANSACTION)

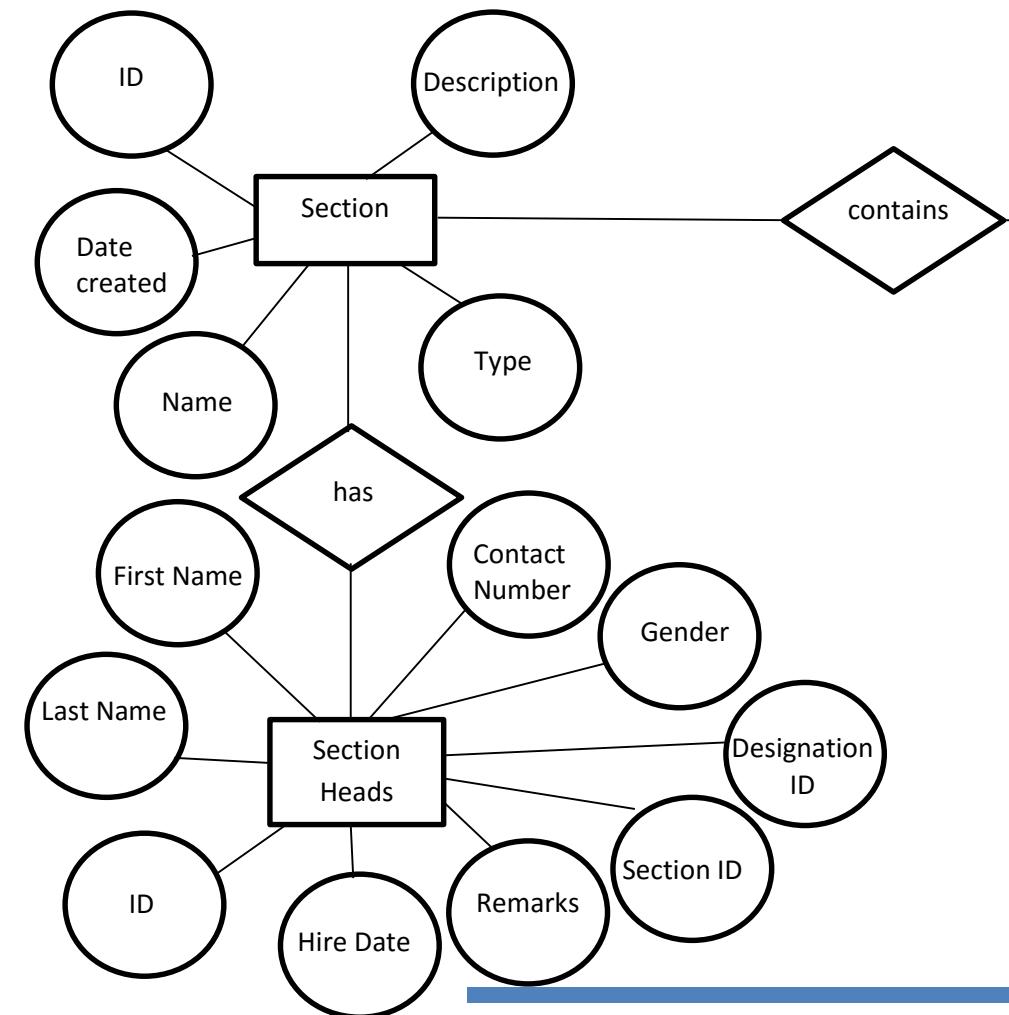
- Log



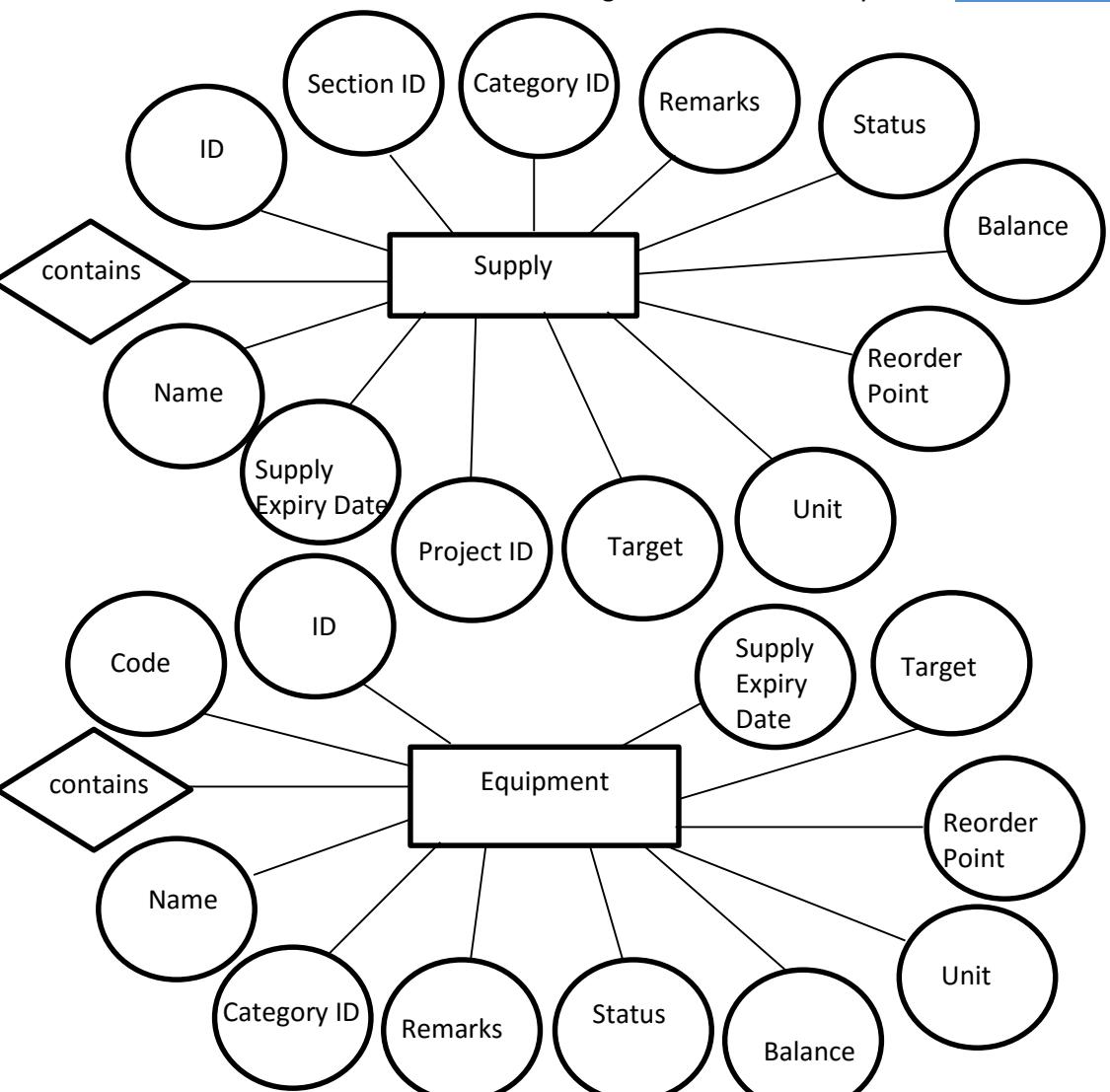
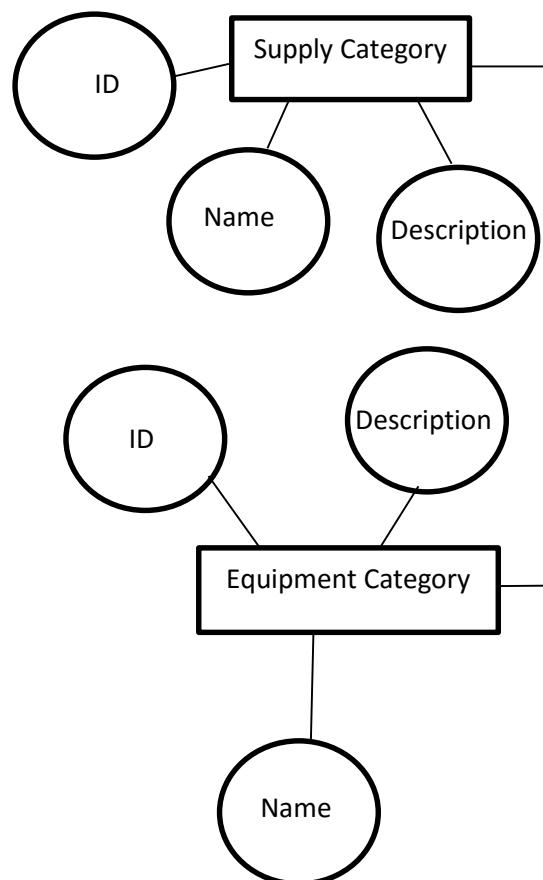
- Register

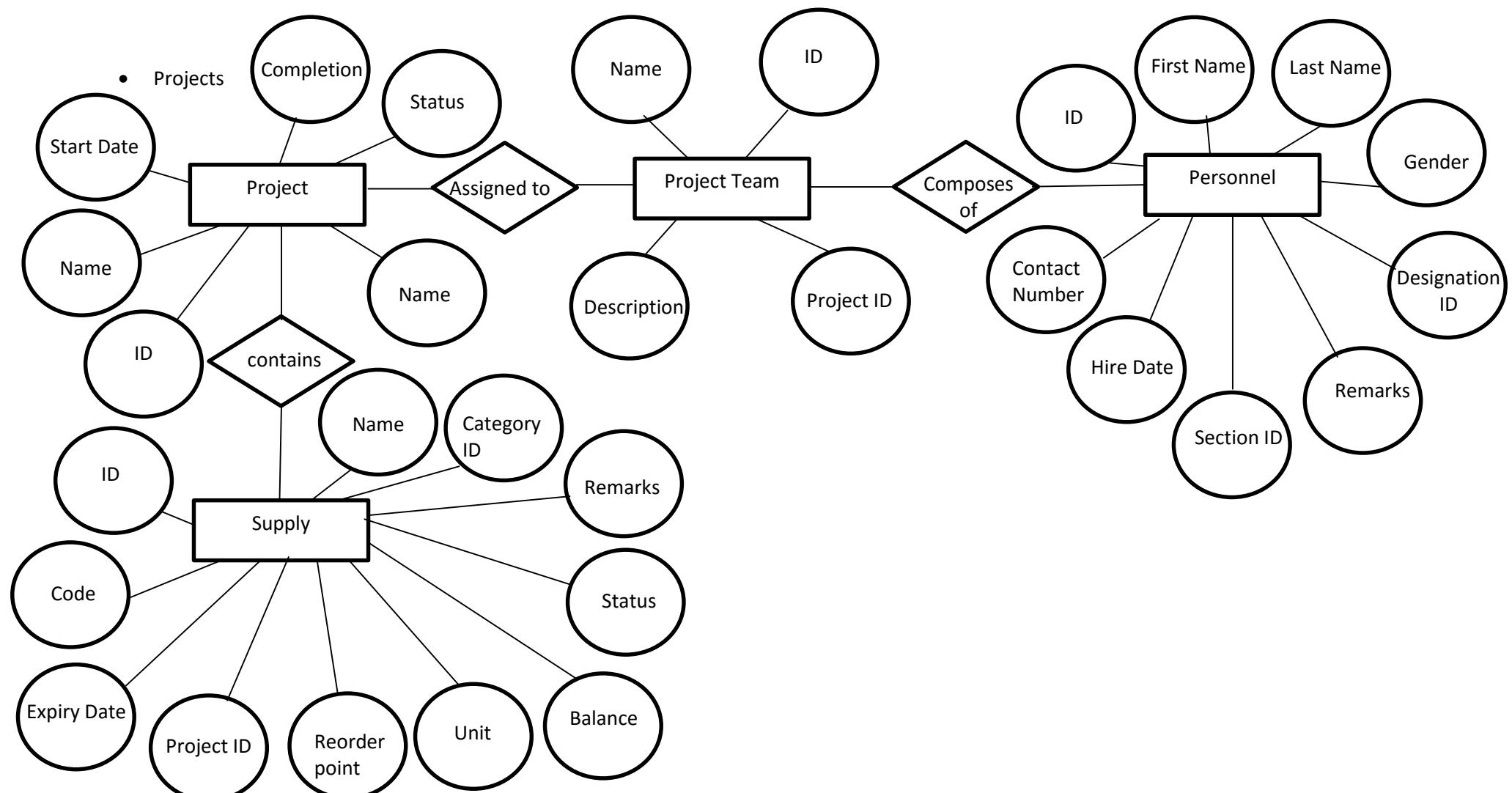


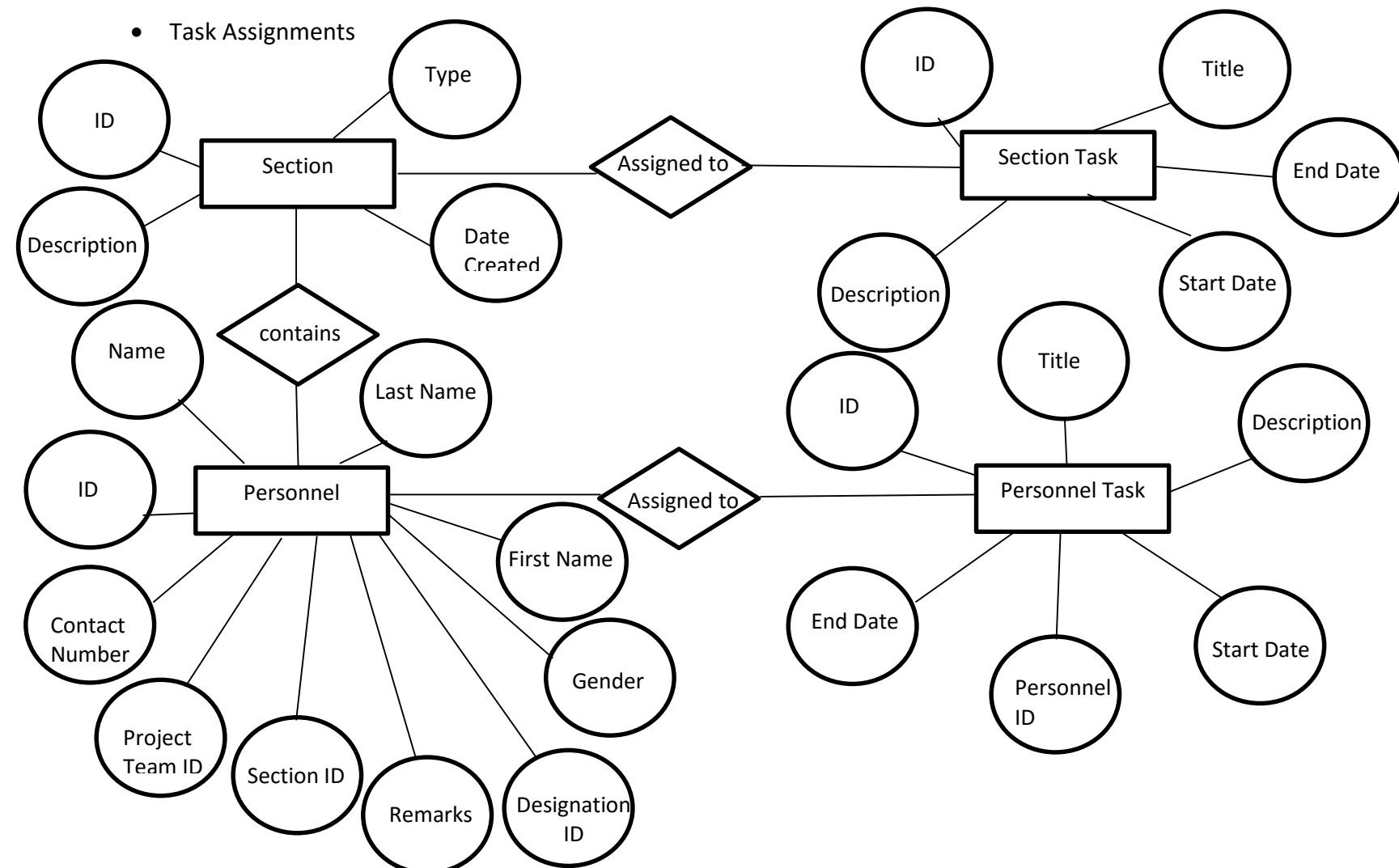
- Human Resources



- Inventory







## CHAPTER 4

### Graphical User Interface (GUI)

#### Landing Page

**Physical Facilities Management Office.**  
MANAGEMENT INFORMATION SYSTEM

**FULL RESPONSIVE**  
Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod.  
Nulla id dolor id nibh ultricies vehicula ut id elit. Morbi leo risus.  
[Details >](#)

**LESS/SASS FILES**  
Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod.  
Nulla id dolor id nibh ultricies vehicula ut id elit. Morbi leo risus.  
[Details >](#)

**6 CHARTS LIBRARY**  
Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod.  
Nulla id dolor id nibh ultricies vehicula ut id elit. Morbi leo risus.  
[Details >](#)

**ADVANCED FORMS**  
Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod.  
Nulla id dolor id nibh ultricies vehicula ut id elit. Morbi leo risus.  
[Details >](#)

#### Log-in Page

Physical Facilities Management Office  
Management Information System

Log In: Start your Session.

Email

Password

**Log In**

[Forgot password?](#)  
[Do not have an account?](#)  
[Create an account](#)

Pamantasan ng Lungsod ng Maynila © 2015-2016  
localhost:50218

## Register Page

Physical Facilities Management Office  
Create Account. See it in action.

Email  
First Name  
Last Name  
Password  
Confirm Password

**Register**

Already have an account?  
[Log In](#)

Pamantasan ng Lungsod ng Maynila © 2015-2016

## Home Page

Search for something...

Hello Alexander Paul Quint! [Logout](#)

Alexander Paul Quint  
Dashboard

- Human Resources
- Inventory
- Projects
- Task Assignments
- Records
- Accounts
- Take a Tour!** [START](#)
- Developers
- Manual
- About

Okay, Let's Start. Time to Manage.

**Click to Start**

Copyright Pamantasan ng Lungsod ng Maynila © 2014-2015

Physical Facilities Management Office

## I. Human Resources

### Manage Sections

Section Name	Section Type	Section Description	Date Created	Action
Administrative Support Staff Section			2015-03-09	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Electrical Section	Equipment	Electrical Related	2015-03-09	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
General Services Section	Equipment	All-around Services	2015-03-09	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Grounds Maintenance and Sanitation Section			2015-03-09	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Mechanical Section	Projects	Mechanical Related	2015-03-09	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>

### Manage Section Heads

Section Name	Designation	Last Name	First Name	Gender	Remarks	Hire Date	Contact Number	Action
Electrical Section	Supervising Administrative Officer	Diallo	Dennis Edward	Male		2012-09-05	8102391684	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Mechanical Section	Senior Building Electrician A	Rivas	Leandro	Male		2014-07-08	8293456781	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
General Services Section	Construction Maintenance General Foreman	Fadera	Stiven	Male		2013-01-14	9068847634	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>

### Manage Personnel

Designation	Project Team	Section Name	Last Name	First Name	Gender	Remarks	Hire Date	Contact Number	Action
Carpenter I	EveryJuan	General Services Section	Aveilla	Laurence	Male		2014-01-15	9293462451	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Aircon Technician I	Bomboompow	Mechanical Section	Oncor	Emilio	Male		2015-04-16	9278953462	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Electrician I	EveryJuan	Electrical Section	Ramos	Cesar	Male		2013-05-06	9234532122	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>

## Manage Designation

### Personnel Designation

Personnel Designations

Designation	Description	
Aeron Technician I	Mechanical	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Carpenter I	General Services	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Electrician I	Electrical	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>

### Section Head Designation

Section Head Designations

Designation	Description	
Construction Maintenance General Foreman	General Services Section	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Senior Building Electrician A	Mechanical Section	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
Supervising Administrative Officer	Electrical Section	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>

## II. Inventory

### Manage Categories

#### Equipment Category

Category Title	Category Description	Actions
Carpentry Supplies	Carpentry	[Details] [Edit] [Delete]
Electrical Supplies	Electrical	[Details] [Edit] [Delete]
Plumbing Supplies	Plumbing	[Details] [Edit] [Delete]

#### Supply Category

Category Title	Category Description	Actions
Aircon Supplies	Aircon	[Details] [Edit] [Delete]
Painting and Varnishing Supplies	Paints and Varnish	[Details] [Edit] [Delete]

## Manage Equipment

Category Title	Equipment Code	Equipment Name	Remarks	Status	Balance	Unit of Measure	Reorder Point	Target	Expiry Date
Carpentry Supplies	CAR001	CWN 3/4"		30	15		2016-12-22		
Carpentry Supplies	CAR018	Sticwell Glue		5	3		2017-09-07		
Electrical Supplies	E8922	Stranded Wire No.12 black		15	10		2016-11-02		

## Manage Supply

Project Title	Category Title	Supply Code	Supply Name	Remarks	Status	Balance	Unit of Measure	Reorder Point	Target	Expiry Date
GCA Building Maintenance	Aircon Supplies	AIR007	WD-40 (400cc)		15	5		2017-09-07		
GCA Building Maintenance	Aircon Supplies	AIR002	Bronze Rod		6	6		2016-03-24		

### III. Projects

#### Manage Teams

Project Title	Project Team	Project Team Description	Actions
GV 301 Renovation	EveryJuan	General	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
GCA Building Maintenance	Boomboompow	General	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>

#### Manage Projects

Project Title	Project Status	Project Completion	Start Date	End Date	Actions
GCA Building Maintenance	On-Going	20	2016-03-04	2016-04-04	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>
GV 301 Renovation	On-Going	11	2016-01-16	2016-05-05	<a href="#">Details</a> <a href="#">Edit</a> <a href="#">Delete</a>

## IV. Task Assignments

### Section Tasks

The screenshot shows the PFMO Management Information System interface. The title bar reads "PFMO Management | Index". The URL in the address bar is "localhost:50210/SectionTasks". The top navigation bar includes a search bar and a user profile with the name "Hello Alexander Paul Quintill". Below the navigation is a sidebar with links: Human Resources, Inventory, Projects, Task Assignments, Records, Accounts, Take a Tour!, Developers, Manual, and About. The main content area is titled "Index" and "SectionTask". It displays a table titled "List of SectionTask" with two rows:

Section Name	Section Task Title	Section Task Description	Start Date	End Date	Details	Edit	Delete
Electrical Section	Light Changing	Changing of lights	March 21 2016	March 21 2016	<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
General Services Section	Elevator Cleaning	Cleaning of Elevator	March 5 2016	March 5 2016	<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

At the bottom of the page, there is a footer with the text "Copyright Pamantasan ng Lungsod ng Maynila © 2014-2015" and "Physical Facilities Management Office". The system status bar at the bottom right shows "96:07 AM" and "16/03/2016".

### Personnel Task

The screenshot shows the PFMO Management Information System interface. The title bar reads "PFMO Management | Index". The URL in the address bar is "localhost:50210/PersonnelTasks". The top navigation bar includes a search bar and a user profile with the name "Hello Alexander Paul Quintill". Below the navigation is a sidebar with links: Human Resources, Inventory, Projects, Task Assignments, Records, Accounts, Take a Tour!, Developers, Manual, and About. The main content area is titled "Index" and "PersonnelTask". It displays a table titled "List of PersonnelTask" with two rows:

PersonnelFullName	Personnel Task Title	Personnel Task Description	Start Date	End Date	Details	Edit	Delete
Ramos, Cesar	Electrical Maintenance				<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Averilla, Laurence	Electrical Maintenance				<a href="#">Details</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

At the bottom of the page, there is a footer with the text "Waiting for localhost..." and "Physical Facilities Management Office". The system status bar at the bottom right shows "96:10 AM" and "16/03/2016".

## CHAPTER 5

### SOURCE CODE

#### PFMO.Context

##### *Identity\_DB.cs*

```

using System.Data.Entity;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace PFMO.Contexts
{
    // You can add profile data for the user by adding
    more properties to your ApplicationUser class,
    please visit
    http://go.microsoft.com/fwlink/?LinkId=317594 to
    learn more.
    public class ApplicationUser : IdentityUser
    {

        public string FirstName { get; set; }
        public string LastName { get; set; }
        public async Task<ClaimsIdentity>
GenerateUserIdentityAsync(UserManager<ApplicationUser> manager)
        {
            // Note the authenticationType must match
            the one defined in
            CookieAuthenticationOptions.AuthenticationType
            var userIdentity = await
            manager.CreateIdentityAsync(this,
            DefaultAuthenticationTypes.ApplicationCookie);
            // Add custom user claims here
            return userIdentity;
        }
    }

    public class Identity_DB :
    IdentityDbContext<ApplicationUser>
    {
        public Identity_DB()
        : base("PFMO")
        {
        }

        public static Identity_DB Create()
        {
    
```

```
        return new Identity_DB();
    }
}
```

```
}
```

#### PFMO.Context

##### *IdentityModels.cs*

```

using System.Data.Entity;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace PFMO.Contexts
{
    // You can add profile data for the user by adding
    more properties to your ApplicationUser class,
    please visit
    http://go.microsoft.com/fwlink/?LinkId=317594 to
    learn more.
    public class ApplicationUser : IdentityUser
    {
        public async Task<ClaimsIdentity>
GenerateUserIdentityAsync(UserManager<ApplicationUser> manager)
        {
            // Note the authenticationType must match
            the one defined in
            CookieAuthenticationOptions.AuthenticationType
            var userIdentity = await
            manager.CreateIdentityAsync(this,
            DefaultAuthenticationTypes.ApplicationCookie);
            // Add custom user claims here
            return userIdentity;
        }
    }
}
```

```

public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
{
    public ApplicationDbContext()
        : base("DefaultConnection")
    {
    }

    public static ApplicationDbContext Create()
    {
        return new ApplicationDbContext();
    }
}

public DbSet<ProjectTeam> ProjectTeams { get; set; }
public DbSet<Section> Sections { get; set; }
public DbSet<SectionHead> SectionHeads { get; set; }
public DbSet<SectionHeadDesignation> SectionHeadDesignations { get; set; }
public DbSet<SectionTask> SectionTasks { get; set; }
public DbSet<Supply> Supplies { get; set; }
public DbSet<SupplyCategory> SupplyCategories { get; set; }

#endregion

}
}

```

**PFMO.Context***PFMO\_DB.cs*

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.Entity;
using System.Security.Claims;
using PFMO.Entities;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace PFMO.Contexts
{
    public class PFMO_DB : DbContext
    {
        public PFMO_DB()
            : base("PFMO")
        {
        }

        #region DbSets
        public DbSet<Contact> Contacts { get; set; }
        public DbSet<Equipment> Equipments { get; set; }
        public DbSet<EquipmentCategory> EquipmentCategories { get; set; }
        public DbSet<Personnel> Personnels { get; set; }
        public DbSet<PersonnelDesignation> PersonnelDesignations { get; set; }
        public DbSet<PersonnelTask> PersonnelTasks { get; set; }
        public DbSet<Pinboard> Pinboards { get; set; }
        public DbSet<Project> Projects { get; set; }
    }
}

```

**PFMO.Entities***Contacts.cs*

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PFMO.Entities
{
    public class Contact
    {
        public Guid ContactID { get; set; }

        [Display(Name = "Contact Name")]
        [DataType(DataType.Text)]
        [StringLength(50, ErrorMessage = "The Contact Name Value cannot exceed 17 characters. ")]
        public string ContactName { get; set; }

        [Display(Name = "Contact Address")]
        [DataType(DataType.Text)]
        [StringLength(100, ErrorMessage = "The Contact Name Value cannot exceed 100 characters. ")]
        public string ContactDescription { get; set; }
    }
}

```

```

[Display(Name = "Contact Type")]
[DataType(DataType.Text)]
[StringLength(30, ErrorMessage = "The Contact
Name Value cannot exceed 30 characters. ")]
public string ContactType { get; set; }

[Display(Name = "Contact Office")]
[DataType(DataType.Text)]
[StringLength(50, ErrorMessage = "The Contact
Name Value cannot exceed 50 characters. ")]
public string ContactOffice { get; set; }

[Display(Name = "Contact Email")]
[DataType(DataType.EmailAddress)]
public string ContactEmail { get; set; }

[Display(Name = "Contact Number")]
[DataType(DataType.PhoneNumber)]
public string ContactNumber { get; set; }
}

}

[Display(Name = "Category")]
public Guid EquipmentCategoryID { get; set; }
public virtual EquipmentCategory
EquipmentCategory { get; set; }

[Display(Name = "Remarks")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string EquipmentRemarks { get; set; }

[Display(Name = "Status")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string EquipmentStatus { get; set; }

[Display(Name = "Balance")]
public int EquipmentBalance { get; set; }

```

**PFMO.Entities***Equipment.cs*

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using
System.ComponentModel.DataAnnotations.Schema
;
using System.Data.Entity;

namespace PFMO.Entities
{
    public class Equipment
    {
        public Guid EquipmentID { get; set; }

        [Required]
        [Display(Name = "Equipment Code")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 50 characters. ")]
        public string EquipmentCode { get; set; }

        [Required]
        [Display(Name = "Equipment Name")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string EquipmentName { get; set; }
    }
}

```

```

[Display(Name = "Unit of Measure")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string EquipmentUnit { get; set; }

[Display(Name = "Reorder Point")]
public int EquipmentReorderPoint { get; set; }

[Display(Name = "Target")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string Target { get; set; }

[Display(Name = "Expiry Date")]
[DataType(DataType.DateTime)]
[DisplayFormat(DataFormatString = "{0:yyyy-
MM-dd}", ApplyFormatInEditMode = true)]
public DateTime EquipmentExpiryDate { get; set; }

}
}

```

**PFMO.Entities**  
*EquipmentCategory.cs*

using System;

```

using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace PFMO.Entities
{
    public class EquipmentCategory
    {
        public Guid EquipmentCategoryID { get; set; }

        [Required]
        [Display(Name = "Category Title")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string EquipmentCategoryName { get;
set; }

        [Display(Name = "Category Description")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string EquipmentCategoryDescription { get;
set; }

        public virtual ICollection<Equipment>
Equipments { get; set; }
    }
}

```

```

[Display(Name = "Last Name")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string PersonnelLastName { get; set; }

[Required]
[Display(Name = "First Name")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string PersonnelFirstName { get; set; }

[Display(Name = "Gender")]
[DataType(DataType.Text)]
[StringLength(10, ErrorMessage = "Value
cannot exceed 10 characters. ")]
public string PersonnelGender { get; set; }

[Display(Name = "Designation")]
public Guid PersonnelDesignationID { get; set; }
public virtual PersonnelDesignation
PersonnelDesignation { get; set; }

[Display(Name = "Remarks")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string PersonnelRemarks { get; set; }

```

### **PFMO.Entities**

*Personnel.cs*

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using
System.ComponentModel.DataAnnotations.Schema
;

namespace PFMO.Entities
{
    public class Personnel
    {
        public Guid PersonnelID { get; set; }

        public string PersonnelFullName
        {
            get { return PersonnelLastName + ", " +
PersonnelFirstName; }
        }

        [Required]

```

```

[Display(Name = "Section Assigned")]
public Guid SectionID { get; set; }
public virtual Section Section { get; set; }

[Display(Name = "Project Team Assigned")]
public Guid ProjectTeamID { get; set; }
public virtual ProjectTeam ProjectTeam { get;
set; }

public virtual ICollection<PersonnelTask>
PersonnelTasks { get; set; }

[Display(Name = "Hire Date")]
[DataType(DataType.DateTime)]
[DisplayFormat(DataFormatString = "{0:yyyy-
MM-dd}", ApplyFormatInEditMode = true)]
public DateTime PersonnelHireDate { get; set; }

[Display(Name = "Contact Number")]
[DataType(DataType.PhoneNumber)]
public long PersonnelContactNumber { get; set;
}

}
}

```

```
}
```

**PFMO.Entities***PersonnelDesignation.cs*

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PFMO.Entities
{
    public class PersonnelDesignation
    {
        public Guid PersonnelDesignationID { get; set; }

        [Required]
        [Display(Name = "Designation")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string PersonnelDesignationName { get; set; }

        [Display(Name = "Description")]
        [DataType(DataType.Text)]
        [StringLength(500, ErrorMessage = "Value
cannot exceed 500 characters. ")]
        public string PersonnelDesignationDescription { get; set; }

        public virtual ICollection<Personnel> Personnel
        { get; set; }
    }
}
```

**PFMO.Entities**  
*PersonnelTask.cs*

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
;

namespace PFMO.Entities
{
    public class PersonnelTask
    {
        public Guid PersonnelTaskID { get; set; }
```

[Required]

[Display(Name = "Personnel Task Title")]

[DataType(DataType.Text)]

[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]

public string PersonnelTaskTitle { get; set; }

[Display(Name = "Personnel Task Description")]

[DataType(DataType.Text)]

[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]

public string PersonnelTaskDescription { get; set; }

[Display(Name = "Personnel Assigned")]

public Guid PersonnelID { get; set; }

public virtual Personnel Personnel { get; set; }

[Display(Name = "Start Date")]

[DataType(DataType.Text)]

[DisplayFormat(DataFormatString = "{0:yyyy-
MM-dd}", ApplyFormatInEditMode = true)]

public string PersonnelTaskStartDate { get; set; }

[Display(Name = "End Date")]

[DataType(DataType.Text)]

[DisplayFormat(DataFormatString = "{0:yyyy-
MM-dd}", ApplyFormatInEditMode = true)]

public string PersonnelTaskEndDate { get; set; }

}

**PFMO.Entities***Pinboard.cs*

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

namespace PFMO.Entities

{

public class Pinboard

{

public Guid PinboardID { get; set; }

[Required]

[Display(Name = "Note Title")]

```

[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string PinboardName { get; set; }

[Display(Name = "Note Content")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
public string PinboardContent { get; set; }

[Display(Name = "Date")]
[DataType(DataType.DateTime)]
[DisplayFormat(DataFormatString = "{0:yyyy-
MM-dd}", ApplyFormatInEditMode = true)]
public DateTime PinboardDate { get; set; }
}
}

```

#### PFMO.Entities Projects.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace PFMO.Entities
{
    public class Project
    {
        public Guid ProjectID { get; set; }

        [Required]
        [Display(Name = "Project Title")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string ProjectName { get; set; }

        [Display(Name = "Project Status")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string ProjectStatus { get; set; }

        [Display(Name = "Project Completion")]
        [Range(0, 100)]
        public int ProjectCompletion { get; set; }

        [Display(Name = "Start Date")]
        [DataType(DataType.DateTime)]
        [DisplayFormat(DataFormatString = "{0:yyyy-
MM-dd}", ApplyFormatInEditMode = true)]
    }
}

```

```

public DateTime ProjectStartDate { get; set; }

[Display(Name = "End Date")]
[DataType(DataType.DateTime)]
[DisplayFormat(DataFormatString = "{0:yyyy-
MM-dd}", ApplyFormatInEditMode = true)]
public DateTime ProjectEndDate { get; set; }

public virtual ICollection<ProjectTeam>
ProjectTeams { get; set; }
public virtual ICollection<Supply> Supplies { get;
set; }
}

```

#### PFMO.Entities ProjectTeam.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace PFMO.Entities
{
    public class ProjectTeam
    {
        public Guid ProjectTeamID { get; set; }

        [Required]
        [Display(Name = "Project Team")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string ProjectTeamName { get; set; }

        [Display(Name = "Project Team Description")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string ProjectTeamDescription { get; set; }

        public virtual ICollection<Personnel>
Personnels { get; set; }

        [Display(Name = "Assigned Project")]
        public Guid ProjectID { get; set; }
        public virtual Project Project { get; set; }
    }
}

```

```
}
```

PFMO.Entities  
Section.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace PFMO.Entities
{
    public class Section
    {
        public Guid SectionID { get; set; }

        [Required]
        [Display(Name = "Section Name")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string SectionName { get; set; }

        public virtual ICollection<SectionHead>
SectionHeads { get; set; }

        [Display(Name = "Section Type")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string SectionType { get; set; }

        [Display(Name = "Section Description")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string SectionDescription { get; set; }

        public virtual ICollection<Personnel>
Personnels { get; set; }

        public virtual ICollection<SectionTask>
SectionTasks { get; set; }

        [Display(Name = "Date Created")]
        [DataType(DataType.DateTime)]
        [DisplayFormat(DataFormatString = "{0:yyyy-
MM-dd}", ApplyFormatInEditMode = true)]
        public DateTime SectionDateCreated { get; set;
}
    }
}
```

**PFMO.Entities**  
**Sectionhead.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace PFMO.Entities
{
    public class SectionHead
    {
        public Guid SectionHeadID { get; set; }

        public string SectionHeadFullName
        {
            get { return SectionHeadLastName + ", " +
SectionHeadFirstName; }
        }

        [Required]
        [Display(Name = "Last Name")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string SectionHeadLastName { get; set; }

        [Required]
        [Display(Name = "First Name")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
        public string SectionHeadFirstName { get; set; }

        [Display(Name = "Gender")]
        [DataType(DataType.Text)]
        [StringLength(10, ErrorMessage = "Value
cannot exceed 10 characters. ")]
        public string SectionHeadGender { get; set; }

        [Display(Name = "Designation")]
        public Guid SectionHeadDesignationID { get;
set; }

        public virtual SectionHeadDesignation
SectionHeadDesignation { get; set; }

        [Display(Name = "Remarks")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value
cannot exceed 200 characters. ")]
    }
}
```

```

public string SectionHeadRemarks { get; set; }

[Display(Name = "Section Assigned")]
public Guid SectionID { get; set; }
public virtual Section Section { get; set; }

[Display(Name = "Hire Date")]
[DataType(DataType.DateTime)]
[DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
public DateTime SectionHeadHireDate { get; set; }

[Display(Name = "Contact Number")]
[DataType(DataType.PhoneNumber)]
public long SectionHeadContactNumber { get; set; }
}
}

```

#### **PFMO.Entities SectionHeadDesignation.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PFMO.Entities
{
    public class SectionHeadDesignation
    {
        public Guid SectionHeadDesignationID { get; set; }

        [Required]
        [Display(Name = "Designation")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
        public string SectionHeadDesignationName { get; set; }

        [Display(Name = "Description")]
        [DataType(DataType.Text)]
        [StringLength(500, ErrorMessage = "Value cannot exceed 500 characters. ")]
        public string SectionHeadDesignationDescription { get; set; }
    }
}

```

```

public virtual ICollection<SectionHead>
SectionHead { get; set; }
}
}

```

#### **PFMO.Entities SectionTask.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
;

namespace PFMO.Entities
{
    public class SectionTask
    {
        public Guid SectionTaskID { get; set; }

        [Required]
        [Display(Name = "Section Task Title")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
        public string SectionTaskTitle { get; set; }

        [Display(Name = "Section Task Description")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
        public string SectionTaskDescription { get; set; }

        [Display(Name = "Section Assigned")]
        public Guid SectionID { get; set; }
        public virtual Section Section { get; set; }

        [Display(Name = "Start Date")]
        [DataType(DataType.Text)]
        [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
        public string SectionTaskStartDate { get; set; }

        [Display(Name = "End Date")]
        [DataType(DataType.Text)]
        [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
        public string SectionTaskEndDate { get; set; }
    }
}

```

```

PFMO.Entities
Supply.cs

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations.Schema.Schema;

namespace PFMO.Entities
{
    public class Supply
    {
        public Guid SupplyID { get; set; }

        [Required]
        [Display(Name = "Supply Code")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 50 characters. ")]
        public string SupplyCode { get; set; }

        [Required]
        [Display(Name = "Supply Name")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
        public string SupplyName { get; set; }

        [Display(Name = "Category")]
        public Guid SupplyCategoryID { get; set; }
        public virtual SupplyCategory SupplyCategory { get; set; }

        [Display(Name = "Remarks")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
        public string SupplyRemarks { get; set; }

        [Display(Name = "Status")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
        public string SupplyStatus { get; set; }

        [Display(Name = "Balance")]
        public int SupplyBalance { get; set; }

        [Display(Name = "Unit of Measure")]
    }
}

[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
public string SupplyUnit { get; set; }

[Display(Name = "Reorder Point")]
public int SupplyReorderPoint { get; set; }

[Display(Name = "Target")]
[DataType(DataType.Text)]
[StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
public string Target { get; set; }

[Display(Name = "Project")]
public Guid ProjectID { get; set; }
public virtual Project Project { get; set; }

[Display(Name = "Expiry Date")]
[DataType(DataType.DateTime)]
[DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
public DateTime SupplyExpiryDate { get; set; }
}

PFMO.Entities
SupplyCategory.cs

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace PFMO.Entities
{
    public class SupplyCategory
    {
        public Guid SupplyCategoryID { get; set; }

        [Required]
        [Display(Name = "Category Title")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
        public string SupplyCategoryName { get; set; }

        [Display(Name = "Category Description")]
        [DataType(DataType.Text)]
        [StringLength(200, ErrorMessage = "Value cannot exceed 200 characters. ")]
        public string SupplyCategoryDescription { get; set; }
    }
}

```

```

        public virtual ICollection<Supply> Supply { get;
set; }
    }
}

{
    record.ContactID = Guid.NewGuid();
    _db.Contacts.Add(record);
}
_db.SaveChanges();
}

//Delete()
public void Delete(Contact record)
{
    _db.Contacts.Remove(record);
    _db.SaveChanges();
}

public void Dispose(bool disposing)
{
    if (disposing)
    {
        _db.Dispose();
    }
}
}

PFMO.Logic
EquipmentCategoriesLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class EquipmentCategoriesLogic : IPFMO<EquipmentCategory>
    {

        //Initialization
        private PFMO_DB _db;
        public EquipmentCategoriesLogic(PFMO_DB db)
        {
            _db = db;
        }

        // GetList()
        public List<EquipmentCategory> GetList()
        {
            return _db.EquipmentCategories.OrderBy(x =>
x.EquipmentCategoryName).ToList();
        }

        // Find()
        public EquipmentCategory Find(Guid id)
        {
            return _db.EquipmentCategories.Find(id);
        }

        //Save()
        public void Save(EquipmentCategory record)
        {
            if (record.ContactID != Guid.Empty)
            {
                _db.Entry(record).State =
EntityState.Modified;
            }
            else
        }

        {
            record.ContactID = Guid.NewGuid();
            _db.Contacts.Add(record);
        }
        _db.SaveChanges();
    }
}

public Task<List<EquipmentCategory>>
ToListAsync()
{
    return _db.EquipmentCategories.OrderBy(x => x.EquipmentCategoryName).ToListAsync();
}

```

```

        }

        public Task<EquipmentCategory>
FindAsync(Guid id)
{
    return
_db.EquipmentCategories.FindAsync(id);
}

        public Task
SaveChangesAsync(EquipmentCategory record)
{
    if (record.EquipmentCategoryID !=
Guid.Empty)
    {
        _db.Entry(record).State =
EntityState.Modified;
    }
    else
    {
        record.EquipmentCategoryID =
Guid.NewGuid();
        _db.EquipmentCategories.Add(record);
    }
    return _db.SaveChangesAsync();
}

        public Task Remove(EquipmentCategory
record)
{
    _db.EquipmentCategories.Remove(record);
    return _db.SaveChangesAsync();
}

        public void Dispose(bool disposing)
{
    if (disposing)
    {
        _db.Dispose();
    }
}
}

```

### PFMO.Logic EquipmentLogic.cs

```

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class EquipmentsLogic :
IPFMOAsync<Equipment>
{
    //Initialization
    private PFMO_DB _db;
    public EquipmentsLogic(PFMO_DB db)
    {
        _db = db;
    }

    public void Dispose(bool disposing)
{
    if (disposing)
    {
        _db.Dispose();
    }
}

    public Task<Equipment> FindAsync(Guid id)
{
    return _db.Equipments.FindAsync(id);
}

    public Task Remove(Equipment record)
{
    _db.Equipments.Remove(record);
    return _db.SaveChangesAsync();
}

    public Task SaveChangesAsync(Equipment
record)
{
    if (record.EquipmentID != Guid.Empty)
    {
        _db.Entry(record).State =
EntityState.Modified;
    }
    else
    {
        record.EquipmentID = Guid.NewGuid();
        _db.Equipments.Add(record);
    }
    return _db.SaveChangesAsync();
}

    public Task<List<Equipment>> ToListAsync()
{

```

```

        return _db.Equipments.OrderBy(x =>
x.EquipmentCode).ToListAsync();
    }
}
}

PFMO.Logic
PersonnelDesignationLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class PersonnelDesignationsLogic : IPFMOAsync<PersonnelDesignation>
    {
        //Initialization
        private PFMO_DB _db;
        public PersonnelDesignationsLogic(PFMO_DB db)
        {
            _db = db;
        }

        public void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
        }

        public Task<PersonnelDesignation> FindAsync(Guid id)
        {
            return
            _db.PersonnelDesignations.FindAsync(id);
        }

        public Task Remove(PersonnelDesignation record)
        {
            _db.PersonnelDesignations.Remove(record);
            return _db.SaveChangesAsync();
        }

        }
    }

    public Task
SaveChangesAsync(PersonnelDesignation record)
{
    if (record.PersonnelDesignationID != Guid.Empty)
    {
        _db.Entry(record).State =
EntityState.Modified;
    }
    else
    {
        record.PersonnelDesignationID =
Guid.NewGuid();
        _db.PersonnelDesignations.Add(record);
    }
    return _db.SaveChangesAsync();
}

public Task<List<PersonnelDesignation>>
ToListAsync()
{
    return _db.PersonnelDesignations.OrderBy(x => x.PersonnelDesignationName).ToListAsync();
}
}
}

PFMO.Logic
PersonnelLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class PersonnelsLogic : IPFMOAsync<Personnel>
    {
        //Initialization
        private PFMO_DB _db;
        public PersonnelsLogic(PFMO_DB db)
        {
            _db = db;
        }

        }
    }
}

```

```

public void Dispose(bool disposing)
{
    if (disposing)
    {
        _db.Dispose();
    }
}

public Task<Personnel> FindAsync(Guid id)
{
    return _db.Personnels.FindAsync(id);
}

public Task Remove(Personnel record)
{
    _db.Personnels.Remove(record);
    return _db.SaveChangesAsync();
}

public Task SaveChangesAsync(Personnel record)
{
    if (record.PersonnelID != Guid.Empty)
    {
        _db.Entry(record).State =
EntityState.Modified;
    }
    else
    {
        record.PersonnelID = Guid.NewGuid();
        _db.Personnels.Add(record);
    }
    return _db.SaveChangesAsync();
}

public Task<List<Personnel>> ToListAsync()
{
    return _db.Personnels.OrderBy(x =>
x.PersonnelLastName).ToListAsync();
}
}

PFMO.Logic
PersonnelTaskLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;

```

```

using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class PersonnelTasksLogic :
IPFMOAsync<PersonnelTask>
    {
        //Initialization
        private PFMO_DB _db;
        public PersonnelTasksLogic(PFMO_DB db)
        {
            _db = db;
        }

        public void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
        }

        public Task<PersonnelTask> FindAsync(Guid id)
        {
            return _db.PersonnelTasks.FindAsync(id);
        }

        public Task Remove(PersonnelTask record)
        {
            _db.PersonnelTasks.Remove(record);
            return _db.SaveChangesAsync();
        }

        public Task SaveChangesAsync(PersonnelTask record)
        {
            if (record.PersonnelTaskID != Guid.Empty)
            {
                _db.Entry(record).State =
EntityState.Modified;
            }
            else
            {
                record.PersonnelTaskID = Guid.NewGuid();
                _db.PersonnelTasks.Add(record);
            }
            return _db.SaveChangesAsync();
        }

        public Task<List<PersonnelTask>> ToListAsync()
        {
            return _db.PersonnelTasks.OrderBy(x =>
x.PersonnelTaskStartDate).ToListAsync();
        }
    }
}
```

```

PFMO.Logic
PinboardLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class PinboardsLogic : IPFMOAsync<Pinboard>
    {
        //Initialization
        private PFMO_DB _db;
        public PinboardsLogic(PFMO_DB db)
        {
            _db = db;
        }

        public void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
        }

        public Task<Pinboard> FindAsync(Guid id)
        {
            return _db.Pinboards.FindAsync(id);
        }

        public Task Remove(Pinboard record)
        {
            _db.Pinboards.Remove(record);
            return _db.SaveChangesAsync();
        }

        public Task SaveChangesAsync(Pinboard
record)
        {
            if (record.PinboardID != Guid.Empty)
            {
                _db.Entry(record).State =
EntityState.Modified;
            }
        }
    }
}

else
{
    record.PinboardID = Guid.NewGuid();
    _db.Pinboards.Add(record);
}
return _db.SaveChangesAsync();
}

public Task<List<Pinboard>> ToListAsync()
{
    return _db.Pinboards.OrderBy(x =>
x.PinboardDate).ToListAsync();
}
}

}

PFMO.Logic
ProjectLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class ProjectsLogic : IPFMOAsync<Proj
{
    //Initialization
    private PFMO_DB _db;
    public ProjectsLogic(PFMO_DB db)
    {
        _db = db;
    }

    public void Dispose(bool disposing)
    {
        if (disposing)
        {
            _db.Dispose();
        }
    }

    public Task<Project> FindAsync(Guid id)
    {
        return _db.Projects.FindAsync(id);
    }

    public Task SaveChangesAsync(Project
record)
    {
        if (record.ProjectID != Guid.Empty)
        {
            _db.Entry(record).State =
EntityState.Modified;
        }
    }
}
}
```

```

        public Task Remove(Project record)
        {
            _db.Projects.Remove(record);
            return _db.SaveChangesAsync();
        }

        public Task SaveChangesAsync(Project record)
        {
            if (record.ProjectID != Guid.Empty)
            {
                _db.Entry(record).State =
                    EntityState.Modified;
            }
            else
            {
                record.ProjectID = Guid.NewGuid();
                _db.Projects.Add(record);
            }
            return _db.SaveChangesAsync();
        }

        public Task<List<Project>> ToListAsync()
        {
            return _db.Projects.OrderByDescending(x =>
x.ProjectCompletion).ToListAsync();
        }
    }
}

PFMO.Logic
ProjectTeamLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class ProjectTeamsLogic :
IPFMOAsync<ProjectTeam>
    {
        //Initialization
        private PFMO_DB _db;
        public ProjectTeamsLogic(PFMO_DB db)
        {
            _db = db;
        }

        public Task Remove(ProjectTeam record)
        {
            _db.ProjectTeams.Remove(record);
            return _db.SaveChangesAsync();
        }

        public void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
        }

        public Task<ProjectTeam> FindAsync(Guid id)
        {
            return _db.ProjectTeams.FindAsync(id);
        }

        public Task Remove(ProjectTeam record)
        {
            _db.ProjectTeams.Remove(record);
            return _db.SaveChangesAsync();
        }

        public Task SaveChangesAsync(ProjectTeam record)
        {
            if (record.ProjectTeamID != Guid.Empty)
            {
                _db.Entry(record).State =
                    EntityState.Modified;
            }
            else
            {
                record.ProjectTeamID = Guid.NewGuid();
                _db.ProjectTeams.Add(record);
            }
            return _db.SaveChangesAsync();
        }

        public Task<List<ProjectTeam>> ToListAsync()
        {
            return _db.ProjectTeams.OrderBy(x =>
x.ProjectTeamName).ToListAsync();
        }
    }
}

PFMO.Logic
SectionHeadDesignationLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```

using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class SectionHeadDesignationsLogic : IPFMOAsync<SectionHeadDesignation>
    {
        //Initialization
        private PFMO_DB _db;
        public SectionHeadDesignationsLogic(PFMO_DB db)
        {
            _db = db;
        }

        public void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
        }

        public Task<SectionHeadDesignation> FindAsync(Guid id)
        {
            return _db.SectionHeadDesignations.FindAsync(id);
        }

        public Task Remove(SectionHeadDesignation record)
        {
            _db.SectionHeadDesignations.Remove(record);
            return _db.SaveChangesAsync();
        }

        public Task
        SaveChangesAsync(SectionHeadDesignation record)
        {
            if (record.SectionHeadDesignationID != Guid.Empty)
            {
                _db.Entry(record).State =
                    EntityState.Modified;
            }
            else
            {
                record.SectionHeadDesignationID =
                    Guid.NewGuid();
            }
            _db.SectionHeadDesignations.Add(record);
        }
        return _db.SaveChangesAsync();
    }

    public Task<List<SectionHeadDesignation>>
    ToListAsync()
    {
        return _db.SectionHeadDesignations.OrderBy(x =>
            x.SectionHeadDesignationName).ToListAsync();
    }
}

PFMO.Logic  

SectionHeadLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class SectionHeadsLogic : IPFMOAsync<SectionHead>
    {
        //Initialization
        private PFMO_DB _db;
        public SectionHeadsLogic(PFMO_DB db)
        {
            _db = db;
        }

        public void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
        }

        public Task<SectionHead> FindAsync(Guid id)
        {
            return _db.SectionHeads.FindAsync(id);
        }
}

```

```

public Task Remove(SectionHead record)
{
    _db.SectionHeads.Remove(record);
    return _db.SaveChangesAsync();
}

public Task SaveChangesAsync(SectionHead
record)
{
    if (record.SectionHeadID != Guid.Empty)
    {
        _db.Entry(record).State =
EntityState.Modified;
    }
    else
    {
        record.SectionHeadID = Guid.NewGuid();
        _db.SectionHeads.Add(record);
    }
    return _db.SaveChangesAsync();
}

public Task<List<SectionHead>> ToListAsync()
{
    return _db.SectionHeads.OrderBy(x =>
x.SectionHeadLastName).ToListAsync();
}
}
}

```

### PFMO.Logic SectionLogic.cs

```

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class SectionsLogic : IPFMOAsync<Section>
    {
        //Initialization
        private PFMO_DB _db;
        public SectionsLogic(PFMO_DB db)
        {
            _db = db;
        }
    }
}

```

```

public void Dispose(bool disposing)
{
    if (disposing)
    {
        _db.Dispose();
    }
}

public Task<Section> FindAsync(Guid id)
{
    return _db.Sections.FindAsync(id);
}

public Task Remove(Section record)
{
    _db.Sections.Remove(record);
    return _db.SaveChangesAsync();
}

public Task SaveChangesAsync(Section record)
{
    if (record.SectionID != Guid.Empty)
    {
        _db.Entry(record).State =
EntityState.Modified;
    }
    else
    {
        record.SectionID = Guid.NewGuid();
        _db.Sections.Add(record);
    }
    return _db.SaveChangesAsync();
}

public Task<List<Section>> ToListAsync()
{
    return _db.Sections.OrderBy(x =>
x.SectionName).ToListAsync();
}
}

```

### PFMO.Logic SectionTaskLogic.cs

```

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;

```

```

using System.Data.Entity;

namespace PFMO.Logic
{
    public class SectionTasksLogic : IPFMOAsync<SectionTask>
    {
        //Initialization
        private PFMO_DB _db;
        public SectionTasksLogic(PFMO_DB db)
        {
            _db = db;
        }

        public void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
        }

        public Task<SectionTask> FindAsync(Guid id)
        {
            return _db.SectionTasks.FindAsync(id);
        }

        public Task Remove(SectionTask record)
        {
            _db.SectionTasks.Remove(record);
            return _db.SaveChangesAsync();
        }

        public Task SaveChangesAsync(SectionTask record)
        {
            if (record.SectionTaskID != Guid.Empty)
            {
                _db.Entry(record).State =
                    EntityState.Modified;
            }
            else
            {
                record.SectionTaskID = Guid.NewGuid();
                _db.SectionTasks.Add(record);
            }
            return _db.SaveChangesAsync();
        }

        public Task<List<SectionTask>> ToListAsync()
        {
            return _db.SectionTasks.OrderBy(x =>
                x.SectionTaskTitle).ToListAsync();
        }
    }
}

```

### PFMO.Logic SuppliesLogic.cs

```

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class SuppliesLogic : IPFMOAsync<Supply>
    {
        //Initialization
        private PFMO_DB _db;
        public SuppliesLogic(PFMO_DB db)
        {
            _db = db;
        }

        public void Dispose(bool disposing)
        {
            if (disposing)
            {
                _db.Dispose();
            }
        }

        public Task<Supply> FindAsync(Guid id)
        {
            return _db.Supplies.FindAsync(id);
        }

        public Task Remove(Supply record)
        {
            _db.Supplies.Remove(record);
            return _db.SaveChangesAsync();
        }

        public Task SaveChangesAsync(Supply record)
        {
            if (record.SupplyID != Guid.Empty)
            {
                _db.Entry(record).State =
                    EntityState.Modified;
            }
            else

```

```

    {
        record.SupplyID = Guid.NewGuid();
        _db.Supplies.Add(record);
    }
    return _db.SaveChangesAsync();
}

public Task<List<Supply>> ToListAsync()
{
    return _db.Supplies.OrderBy(x =>
x.SupplyCode).ToListAsync();
}
}

PFMO.Logic
SupplyCategoryLogic.cs

using PFMO.Logic.Interface;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using PFMO.Entities;
using PFMO.Contexts;
using System.Data.Entity;

namespace PFMO.Logic
{
    public class SupplyCategoriesLogic :
IPFMOAsync<SupplyCategory>
{
    //Initialization
    private PFMO_DB _db;
    public SupplyCategoriesLogic(PFMO_DB db)
    {
        _db = db;
    }

    public void Dispose(bool disposing)
    {
        if (disposing)
        {
            _db.Dispose();
        }
    }

    public Task<SupplyCategory> FindAsync(Guid
id)
    {
        return _db.SupplyCategories.FindAsync(id);
    }
}

public Task Remove(SupplyCategory record)
{
    _db.SupplyCategories.Remove(record);
    return _db.SaveChangesAsync();
}

public Task SaveChangesAsync(SupplyCategory
record)
{
    if (record.SupplyCategoryID != Guid.Empty)
    {
        _db.Entry(record).State =
EntityState.Modified;
    }
    else
    {
        record.SupplyCategoryID =
Guid.NewGuid();
        _db.SupplyCategories.Add(record);
    }
    return _db.SaveChangesAsync();
}

public Task<List<SupplyCategory>>
ToListAsync()
{
    return _db.SupplyCategories.OrderBy(x =>
x.SupplyCategoryName).ToListAsync();
}
}

PFMO.LogicInterface
IPFMO.cs

using PFMO.Entities;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PFMO.Logic.Interface
{
    public interface IPFMO<T> where T : class
    {
        List<T> GetList();
        T Find(Guid id);
        void Save(T record);
        void Delete(T record);
        void Dispose(bool disposing);
    }
}

```

```

PFMO.LogicInterface
IPFMOAsync.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PFMO.Logic.Interface
{
    public interface IPFMOAsync<T> where T : class
    {
        Task<List<T>> ToListAsync();
        Task<T> FindAsync(Guid id);
        Task SaveChangesAsync(T record);
        Task Remove(T record);
        void Dispose(bool disposing);
    }
}

PFMO.Web
Global.asax.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace PFMO.Web
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();

            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}

PFMO.Web
Startup.cs

using Microsoft.Owin;
using Owin;

[assembly:
OwinStartupAttribute(typeof(PFMO.Web.Startup))]
namespace PFMO.Web
{
    public partial class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            ConfigureAuth(app);
        }
    }
}

PFMO.Web
Views(Account)
ExternalLoginListpartial.cshtml

@model PFMO.Web.Models.ExternalLoginListViewModel
@using Microsoft.Owin.Security

<h4>Use another service to log in.</h4>
<hr />
 @{
    var loginProviders =
        Context.GetOwinContext().Authentication.GetExternalAuthenticationTypes();
    if (loginProviders.Count() == 0) {
        <div>
            <p>
                There are no external authentication services configured. See <a href="http://go.microsoft.com/fwlink/?LinkId=403804">this article</a>
                for details on setting up this ASP.NET application to support logging in via external services.
            </p>
        </div>
    } else {
        using (Html.BeginForm("ExternalLogin", "Account", new { ReturnUrl = Model.ReturnUrl })) {
            @Html.AntiForgeryToken()
            <div id="socialLoginList">
                <p>

```

```
        @foreach (AuthenticationDescription p
in loginProviders) {
            <button type="submit" class="btn btn-
default" id="@p.AuthenticationType"
name="provider" value="@p.AuthenticationType"
title="Log in using your @p.Caption
account">@p.AuthenticationType</button>
        }
    </p>
</div>
}
}
```

## PFMO.Web

### Views(Accounts)

#### ConfirmEmail.cshtml

```
@{  
    ViewBag.Title = "Confirm Email";  
}  
  
<h2>@ViewBag.Title.</h2>  
<div>  
    <p>  
        Thank you for confirming your email. Please  
        @Html.ActionLink("Click here to Log in", "Login",  
        "Account", routeValues: null, htmlAttributes: new {  
            id = "loginLink" })  
        </p>  
    </div>
```

## PFMO.Web Views(Accounts) ExternalLoginConfirmation.cshtml

```
@model PFMO.Web.Models.ExternalLoginConfirmationView  
Model  
{  
    ViewBag.Title = "Register";  
}  
<h2>@ViewBag.Title.</h2>  
<h3>Associate your @ViewBag.LoginProvider  
account.</h3>  
  
@using  
(Html.BeginForm("ExternalLoginConfirmation",  
"Account", new { ReturnUrl = ViewBag.ReturnUrl },  
FormMethod.Post, new { @class = "form-  
horizontal", role = "form" }))  
{  
    @Html.AntiForgeryToken()  
}
```

```
<h4>Association Form</h4>
<hr />
@Html.ValidationSummary(true, "", new { @class =
="text-danger" })
<p class="text-info">
    You've successfully authenticated with
    <strong>@ViewBag.LoginProvider</strong>.
    Please enter a user name for this site below and
    click the Register button to finish
    logging in.
</p>
<div class="form-group">
    @Html.LabelFor(m => m.Email, new { @class =
"col-md-2 control-label" })
    <div class="col-md-10">
        @Html.TextBoxFor(m => m.Email, new {
            @class = "form-control" })
        @Html.ValidationMessageFor(m => m.Email,
        "", new { @class = "text-danger" })
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" class="btn btn-default"
        value="Register" />
    </div>
</div>
}

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

## PFMO.Web Views(Accounts) ExternalLoginFailure.cshtml

```
@{
    ViewBag.Title = "Login Failure";
}

<hgroup>
    <h2>@ViewBag.Title.</h2>
    <h3 class="text-danger">Unsuccessful login with
        service.</h3>
</hgroup>
```

PFMO.Web  
Views(Accounts)  
ForgotPassword.cshtml

```
@model PFMO.Web.Models.ForgotPasswordViewModel
@{
```

```

ViewBag.Title = "Forgot your password?";
Layout = null;
}

<!DOCTYPE html>
<html>
<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Icon -->
    <link rel="icon" href="~/Content/favicon.png">
    <title>PFMO Management | @ViewBag.Title</title>
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,300,600,700" rel='stylesheet' type='text/css'>
        <!-- Add jQuery Style direct - used for jQGrid plugin -->
        <link href="@Url.Content("~/Scripts/plugins/jquery-ui/jquery-ui.css")" rel="stylesheet" type="text/css" />

        <!-- Primary Inspinia style -->
        @Styles.Render("~/Content/css")
        @Styles.Render("~/font-awesome/css")
</head>

<body class="gray-bg">

    <div class="passwordBox animated fadeInDown">
        <div class="row">
            <div class="col-md-12">
                <div class="ibox-content">
                    <h2 class="font-bold">Forgot password</h2>
                    <p>Enter your email address and your password will be reset and emailed to you.</p>
                    <div class="row">
                        <div class="col-lg-12">
                            @using (Html.BeginForm("ForgotPassword", "Account", FormMethod.Post, new { @class = "m-t", role = "form" }))
                            {
                                @Html.AntiForgeryToken()
                                <form class="m-t" role="form">
                                    @Html.ValidationSummary("", new { @class = "text-danger" })
                                    <div class="form-group">
                                        @Html.TextBoxFor(m => m.Email, new { @class = "form-control", @type = "email", @placeholder = "Email", @required = "" })
                                    </div>
                                    <div class="col-md-offset-2 col-md-10">
                                        <button type="submit" class="btn btn-primary block full-width m-b" value="Email Link">Send New Password</button>
                                    </div>
                                </form>
                            }
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <hr />
        <div class="row">
            <div class="col-md-6">
                Copyright P.L.M.
            </div>
            <div class="col-md-6 text-right">
                <small>© 2015-2016</small>
            </div>
        </div>
        <div class="row Scripts">
            @section Scripts {
                @Scripts.Render("~/bundles/jqueryval")
            }
        </div>
    </div>
</body>
</html>

```

**PFMO.Web**  
**Views(Accounts)**  
**ForgotPasswordConfirmation.cshtml**

```

@{
    ViewBag.Title = "Forgot Password Confirmation";
    Layout = null;
}

<!DOCTYPE html>
<html>
<head>

```

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Icon -->
<link rel="icon" href="~/Content/favicon.png">
<title>PFMO Management | @ ViewBag.Title</title>
<link href='https://fonts.googleapis.com/css?family=Open+Sans:400,300,600,700' rel='stylesheet' type='text/css'>
    <!-- Add jQuery Style direct - used for jQGrid plugin -->
    <link href="@Url.Content("~/Scripts/plugins/jquery-ui/jquery-ui.css")" rel="stylesheet" type="text/css" />

    <!-- Primary Inspinia style -->
    @Styles.Render("~/Content/css")
    @Styles.Render("~/font-awesome/css")
</head>

<body class="gray-bg">

    <div class="passwordBox animated fadeInDown">
        <div class="row">
            <div class="col-md-12">
                <div class="ibox-content">
                    <h2 class="font-bold">Forgot password</h2>
                    <p>
                        Please check your email to reset your password.
                    </p>
                    <a href="@Url.Action("Login", "Account")" class="btn btn-primary">Login</a> <a href="@Url.Action("Register", "Account")" class="btn btn-primary">Register</a>
                </div>
            </div>
        </div>
    </div>

    <@section Scripts {
        @Scripts.Render("~/bundles/jqueryval")
    }

</body>
</html>

```

PFMO.Web  
Views(Accounts)

## Login.cshtml

```

@using PFMO.Web.Models
@model LoginViewModel
 @{
     ViewBag.Title = "Login";
     Layout = null;
 }

<!DOCTYPE html>
<html>
<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Icon -->
    <link rel="icon" href="~/Content/favicon.png">
    <title>PFMO Management | Login</title>

    <link href="~/Content/bootstrap.min.css" rel="stylesheet">
    <link href="~/Fonts/font-awesome/css/font-awesome.css" rel="stylesheet">

    <link href="~/Content/animate.css" rel="stylesheet">
    <link href="~/Content/style.css" rel="stylesheet">

</head>

<body class="gray-bg">

    <div class="middle-box text-center loginscreen animated flip">
        <div>

            <section id="loginForm">
                @using (Html.BeginForm("Login", "Account", new { ReturnUrl = ViewBag.ReturnUrl }, FormMethod.Post, new { @class = "form-horizontal", role = "form" }))
                {
                    @Html.AntiForgeryToken()
                    <div>
                        <h1 align="center" class="logo-name"><a href="@Url.Action("Index", "Home")"></a></h1>
                    </div>
                    <h3>Physical Facilities Management Office</h3>
                    <p>
                        Management Information System
                    </p>
                }
            </section>
        </div>
    </div>

```

```

<p>Log In. Start your Session.</p>
<form class="m-t" role="form"
action="#">
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
        <div>
            @Html.TextBoxFor(m => m.Email, new { @type = "email", @class = "form-control", @placeholder = "Email", @required = "" })
            @Html.ValidationMessageFor(m => m.Email, "", new { @class = "text-danger" })
        </div>
    </div>
    <div class="form-group">
        <div>
            @Html.PasswordFor(m => m.Password, new { @type = "password", @class = "form-control", @placeholder = "Password", @required = "" })
            @Html.ValidationMessageFor(m => m.Password, "", new { @class = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div>
            <input type="submit" value="Log In" class="btn btn-primary block full-width m-b" />
        </div>
    </div>

    <a href="@Url.Action("Forgot your password?", "ForgotPassword")"><small>Forgot password?</small></a>
    <p class="text-muted text-center"><small>Do not have an account?</small></p>
    <a class="btn btn-sm btn-white btn-block" href="@Url.Action("Register", "Account")">Create an account</a>

    /* Enable this once you have account confirmation enabled for password reset functionality
    <p>
        @Html.ActionLink("Forgot your password?", "ForgotPassword")
    </p>*@

    </form>
    <p class="m-t"> <small>Pamantasan ng Lungsod ng Maynila © 2015-2016</small>
</p>
}
</section>

@*<div>
<section id="socialLoginForm">
    @Html.Partial("_ExternalLoginsListPartial", new ExternalLoginListViewModel { ReturnUrl = ViewBag.ReturnUrl })
</section>
</div>*@
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

</body>
</html>

```

**PFMO.Web**  
**Views(Accounts)**  
**Register.cshtml**

```

@model PFMO.Web.Models.RegisterViewModel
 @{
    ViewBag.Title = "Register";
    Layout = null;
}

<!DOCTYPE html>
<html>
<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Icon -->
    <link rel="icon" href="~/Content/favicon.png">
    <title>PFMO Management | Register</title>

    <link href="~/Content/bootstrap.min.css" rel="stylesheet">
    <link href="~/Fonts/font-awesome/css/font-awesome.css" rel="stylesheet">

    <link href="~/Content/animate.css" rel="stylesheet">
    <link href="~/Content/style.css" rel="stylesheet">

```

```

</head>

<body class="gray-bg">

    <div class="middle-box text-center loginscreen animated flip">
        <div>

            @using (Html.BeginForm("Register",
"Account", FormMethod.Post, new { @class =
"form-horizontal", role = "form" }))
            {
                @Html.AntiForgeryToken()
                <div>
                    <h1 align="top"><a href="@Url.Action("Index", "land")"></a></h1>
                    </div>
                    <h3>Physical Facilities Management Office</h3>
                    <p>Create Account. See it in action.</p>
                    <form class="m-t" role="form" action="#">

                        @Html.ValidationSummary("", new {
                            @class = "text-danger" })

                        <div class="form-group">
                            <div>
                                @Html.TextBoxFor(m => m.Email,
new { @type = "email", @class = "form-control",
@placeholder = "Email", @required = "" })
                            </div>
                        </div>

                        <div class="form-group">
                            <div>
                                @Html.TextBoxFor(m =>
m.FirstName, new { @type = "text", @class = "form-control",
@placeholder = "First Name", @required = "" })
                            </div>
                        </div>

                        <div class="form-group">
                            <div>
                                @Html.TextBoxFor(m =>
m.LastName, new { @type = "text", @class = "form-control",
@placeholder = "Last Name", @required = "" })
                            </div>
                        </div>

                        <div class="form-group">
                            <div>
                                @Html.PasswordFor(m =>
m.Password, new { @type = "password", @class =
"form-control", @placeholder = "Password",
@required = "" })
                            </div>
                        </div>

                        <div class="form-group">
                            <div>
                                @Html.PasswordFor(m =>
m.ConfirmPassword, new { @type = "password",
@class = "form-control", @placeholder = "Confirm Password",
@required = "" })
                            </div>
                        </div>

                        <div class="form-group">
                            <div>
                                <button type="submit" class="btn btn-primary block full-width m-b" value="Register">Register</button>
                            </div>
                        </div>

                        <p class="text-muted text-center"><small>Already have an account?</small></p>
                        <a class="btn btn-sm btn-white btn-block" href="@Url.Action("Login", "Account")">Log In</a>
                    </form>
                    <p class="m-t"><small>Pamantasan ng Lungsod ng Maynila © 2015-2016</small></p>
                </div>
            </div>
        </div>
    </div>
</body>
</html>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(Accounts)
ResetPassword.cshtml

@model PFMO.Web.Models.ResetPasswordViewModel
 @{
    ViewBag.Title = "Reset password";
}

```

```

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("ResetPassword",
"Account", FormMethod.Post, new { @class =
"form-horizontal", role = "form" }))
{
    @Html.AntiForgeryToken()
    <h4>Reset your password.</h4>
    <hr />
    @Html.ValidationSummary("", new { @class =
"text-danger" })
    @Html.HiddenFor(model => model.Code)
    <div class="form-group">
        @Html.LabelFor(m => m.Email, new { @class =
"col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Email, new {
@class = "form-control" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(m => m.Password, new {
@class = "col-md-2 control-label" })
            <div class="col-md-10">
                @Html.PasswordFor(m => m.Password, new {
@class = "form-control" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(m => m.ConfirmPassword,
new { @class = "col-md-2 control-label" })
            <div class="col-md-10">
                @Html.PasswordFor(m =>
m.ConfirmPassword, new { @class = "form-control" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" class="btn btn-default"
value="Reset" />
            </div>
        </div>
    </div>
}

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(Accounts)**  
**ResetPasswordConfirmation.cshtml**

```

 @{
    ViewBag.Title = "Reset password confirmation";
}

<hgroup class="title">
    <h1>@ViewBag.Title.</h1>
</hgroup>
<div>
    <p>
        Your password has been reset. Please
        @Html.ActionLink("click here to log in", "Login",
"Account", routeValues: null, htmlAttributes: new {
id = "loginLink" })
    </p>
</div>

```

**PFMO.Web**  
**Views(Accounts)**  
**SendCodes.cshtml**

```

@model PFMO.Web.Models.SendCodeViewModel
 @{
    ViewBag.Title = "Send";
}

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("SendCode", "Account",
new { ReturnUrl = Model.ReturnUrl },
FormMethod.Post, new { @class = "form-
horizontal", role = "form" })) {
    @Html.AntiForgeryToken()
    @Html.Hidden("rememberMe",
@Model.RememberMe)
    <h4>Send verification code</h4>
    <hr />
    <div class="row">
        <div class="col-md-8">
            Select Two-Factor Authentication Provider:
            @Html.DropDownListFor(model =>
model.SelectedProvider, Model.Providers)
            <input type="submit" value="Submit"
class="btn btn-default" />
        </div>
    </div>
}

```

```

@section Scripts {

```

```

    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(Accounts)**  
**VerifyCode.cshtml**

```

@model PFMO.Web.Models.VerifyCodeViewModel
 @{
    ViewBag.Title = "Verify";
}

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("VerifyCode", "Account",
new { ReturnUrl = Model.ReturnUrl },
FormMethod.Post, new { @class = "form-
horizontal", role = "form" })) {
    @Html.AntiForgeryToken()
    @Html.Hidden("provider", @Model.Provider)
    @Html.Hidden("rememberMe",
@Model.RememberMe)
    <h4>Enter verification code</h4>
    <hr />
    @Html.ValidationSummary("", new { @class =
"text-danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.Code, new { @class =
"col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Code, new {
                @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <div class="checkbox">
                @Html.CheckBoxFor(m =>
m.RememberBrowser)
                @Html.LabelFor(m =>
m.RememberBrowser)
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" class="btn btn-default"
value="Submit" />
        </div>
    </div>
}

```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

}

PFMO.Web  
Views(Calendar)  
Index.cshtml

```

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-lg-8">
        <h2>Calendar</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Dashboard</a>
            </li>
            <li class="active">
                <strong>Calendar</strong>
            </li>
        </ol>
    </div>
</div>

```

```

<div class="wrapper wrapper-content">
    <div class="row animated fadeInDown">
        <div class="col-lg-3">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Draggable Events</h5>
                    <div class="ibox-tools">
                        <a class="collapse-link">
                            <i class="fa fa-chevron-up"></i>
                        </a>
                        <a class="dropdown-toggle" data-
toggle="dropdown" href="#">
                            <i class="fa fa-wrench"></i>
                        </a>
                    <ul class="dropdown-menu dropdown-
user">
                        <li>
                            <a href="#">Config option 1</a>
                        </li>
                        <li>
                            <a href="#">Config option 2</a>
                        </li>
                    </ul>
                <a class="close-link">

```

```

<i class="fa fa-times"></i>
</a>
</div>
</div>
<div class="ibox-content">
    <div id='external-events'>
        <div class='external-event navy-bg'>Attend Meeting</div>
        <div class='external-event navy-bg'>Update Record</div>
        <div class='external-event navy-bg'>Submit Report</div>
        <div class='external-event navy-bg'>Evaluate Deliverable</div>
        <div class='external-event navy-bg'>Manage Account</div>
        <div class='external-event navy-bg'>Schedule Meeting</div>
        <div class='external-event navy-bg'>Review Report</div>
        <div class='external-event navy-bg'>Activate Account</div>
        <div class='external-event navy-bg'>Change Password</div>
        <div class='external-event navy-bg'>Upload File</div>
        <div class='external-event navy-bg'>Validate Account</div>
        <div class='external-event navy-bg'>Check Inventory</div>
        <div class='external-event navy-bg'>Manage Project</div>
        <p class="m-t">
            <input type='checkbox' id='drop-remove' class="i-checks" checked /> <label for='drop-remove'>remove after drop</label>
        </p>
    </div>
</div>
<div class="ibox float-e-margins">
    <div class="ibox-title">
        <h5>Create Event</h5>
        <div class="ibox-tools">
            <a class="collapse-link">
                <i class="fa fa-chevron-up"></i>
            </a>
            <a class="dropdown-toggle" data-toggle="dropdown" href="#">
                <i class="fa fa-wrench"></i>
            </a>
            <ul class="dropdown-menu dropdown-user">
                <li>
                    <a href="#">Config option 1</a>
                </li>
                <li>
                    <a href="#">Config option 2</a>
                </li>
            </ul>
            <a class="close-link">
                <i class="fa fa-times"></i>
            </a>
        </div>
    </div>
    <div class="ibox-content">
        <div class="btn-group" style="width: 100%; margin-bottom: 10px;">
            <button type="button" id="color-chooser-btn" class="btn btn-info btn-block dropdown-toggle" data-toggle="dropdown">Color <span class="caret"></span></button>
        </div><!-- /btn-group -->
        <div class="input-group">
            <input id="new-event" type="text" class="form-control" placeholder="Event Title">
            <div class="input-group-btn">
                <button id="add-new-event" type="button" class="btn btn-primary btn-flat">Add</button>
            </div><!-- /btn-group -->
            </div><!-- /input-group -->
        </div>
        <div class="col-lg-9">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Striped Table </h5>
                    <div class="ibox-tools">
                        <a class="collapse-link">
                            <i class="fa fa-chevron-up"></i>
                        </a>
                        <a class="dropdown-toggle" data-toggle="dropdown" href="#">
                            <i class="fa fa-wrench"></i>
                        </a>
                        <ul class="dropdown-menu dropdown-user">
                            <li>
                                <a href="#">Config option 1</a>
                            </li>
                            <li>
                                <a href="#">Config option 2</a>
                            </li>
                        </ul>
                        <a class="close-link">
                            <i class="fa fa-times"></i>
                        </a>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

</div>
<div class="ibox-content">
    <div id="calendar"></div>
</div>
</div>
</div>
</div>
</div>

PFMO.Web
Views(Contact)
Create.cshtml

@model PFMO.Entities.Contact

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Contacts</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Dashboard</a>
            </li>
            <li>
                <a href="#">Contact</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Create Contact</h5>
                </div>

```

```

<div class="ibox-content">
    @using
    (Html.BeginForm())
    {
        @Html.AntiForgeryToken()

        <div class="form-horizontal">
            @Html.ValidationSummary(true)

            <div class="form-group">
                <div class="form-group">
                    @Html.LabelFor(model =>
model.ContactName, new { @class = "control-label col-md-2" })
                <div class="col-md-9">
                    @Html.EditorFor(model =>
model.ContactName)

                    @Html.ValidationMessageFor(model =>
model.ContactName)
                </div>
            </div>

            <div class="form-group">
                @Html.LabelFor(model =>
model.ContactDescription, new { @class = "control-label col-md-2" })
                <div class="col-md-9">
                    @Html.EditorFor(model =>
model.ContactDescription)

                    @Html.ValidationMessageFor(model =>
model.ContactDescription)
                </div>
            </div>

            <div class="form-group">
                @Html.LabelFor(model =>
model.ContactType, new { @class = "control-label col-md-2" })
                <div class="col-md-9">
                    @Html.EditorFor(model =>
model.ContactType)

                    @Html.ValidationMessageFor(model =>
model.ContactType)
                </div>
            </div>

            <div class="form-group">
                @Html.LabelFor(model =>
model.ContactOffice, new { @class = "control-label col-md-2" })
                <div class="col-md-9">

```

```

@Html.EditorFor(model =>
model.ContactOffice)

@Html.ValidationMessageFor(model =>
model.ContactOffice)
    </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.ContactEmail, new { @class = "control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.ContactEmail)

@Html.ValidationMessageFor(model =>
model.ContactEmail)
    </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.ContactNumber, new { @class = "control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.ContactNumber)

@Html.ValidationMessageFor(model =>
model.ContactNumber)
    </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Create" class="btn btn-primary" />
            @Html.ActionLink("Cancel", "Index", null, new { @class = "btn btn-white" })
        </div>
    </div>

    </div>
}

</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(Contacts)**  
**Delete.cshtml**

```

@model PFMO.Entities.Contact

{@
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Contacts</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Dashboard</a>
            </li>
            <li>
                <a href="#">Contact</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Contact</h5>
                </div>
                <div class="ibox-content">
                    <h4>
                        Are you sure you want to
                        delete this?
                    </h4>
                    <dl class="dl-horizontal">
                        <dt>
```

```

        @Html.DisplayNameFor(model =>
model.ContactName)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.ContactName)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.ContactDescription)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.ContactDescription)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.ContactType)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.ContactType)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.ContactOffice)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.ContactOffice)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.ContactEmail)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.ContactEmail)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.ContactNumber)
</dt>

<dd>

```

```

        @Html.DisplayFor(model =>
model.ContactNumber)
</dd>

</dl>

@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()
    <div class="form-actions no-color">
        <input type="submit" value="Delete" class="btn btn-primary" />
        @Html.ActionLink("Back to List", "Index", null, new { @class = "btn btn-white" })
    </div>
}

</div>
</div>
</div>
</div>

```

PFMO.Web  
Views(Contact)  
Details.cshtml

```

@model PFMO.Entities.Contact

 @{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Contacts</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Dashboard</a>
            </li>
            <li>
                <a href="#">Contact</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">

```

```

        <strong>Details</strong>
    </li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Details Contact</h5>
                </div>
                <div class="ibox-content">
                    <dl
class="dl-horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.ContactName)
                        </dt>

                        <dd>
                            @Html.DisplayFor(model =>
model.ContactName)
                        </dd>

                        <dt>
                            @Html.DisplayNameFor(model =>
model.ContactDescription)
                        </dt>

                        <dd>
                            @Html.DisplayFor(model =>
model.ContactDescription)
                        </dd>

                        <dt>
                            @Html.DisplayNameFor(model =>
model.ContactType)
                        </dt>

                        <dd>
                            @Html.DisplayFor(model =>
model.ContactType)
                        </dd>

                        <dt>
                            @Html.DisplayNameFor(model =>
model.ContactOffice)
                        </dt>

```

@Html.ActionLink("Edit", "Edit", new { id = Model.ContactID }, new { @class = "btn btn-primary"})  
@Html.ActionLink("Back to List", "Index", null, new { @class = "btn btn-white"})

```

                    </dd>
                </dl>
            </div>
        </div>
    </div>
</div>

```

**PFMO.Web**  
**Views(Accounts)**  
**Edit.cshtml**

```

@model PFMO.Entities.Contact
 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Contacts</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Dashboard</a>
            </li>
            <li>
                <a href="#">Contacts</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Contact</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()

                        <div class="form-horizontal">
                            @Html.ValidationSummary(true)
                            @Html.HiddenFor(model => model.ContactID)

                            <div class="form-group">
                                @Html.LabelFor(model =>
model.ContactName, new { @class = "control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.ContactName)
                                    @Html.ValidationMessageFor(model =>
model.ContactName)
                            </div>
                        </div>
                        <div class="form-group">
                            @Html.LabelFor(model =>
model.ContactDescription, new { @class = "control-label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.ContactDescription)
                                @Html.ValidationMessageFor(model =>
model.ContactDescription)
                            </div>
                        </div>
                        <div class="form-group">
                            @Html.LabelFor(model =>
model.ContactType, new { @class = "control-label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.ContactType)
                                @Html.ValidationMessageFor(model =>
model.ContactType)
                            </div>
                        </div>
                        <div class="form-group">
                            @Html.LabelFor(model =>
model.ContactOffice, new { @class = "control-label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.ContactOffice)
                                @Html.ValidationMessageFor(model =>
model.ContactOffice)
                            </div>
                        </div>
                        <div class="form-group">
                            @Html.LabelFor(model =>
model.ContactEmail, new { @class = "control-label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.ContactEmail)
                                @Html.ValidationMessageFor(model =>
model.ContactEmail)
                            </div>
                        </div>
                        <div class="form-group">
                            @Html.LabelFor(model =>
model.ContactNumber, new { @class = "control-label col-md-2" })
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```



```

        </div>
    </div>

    </div>
    </div>
}
</div>
</div>

@Html.ActionLink("Change View",
"Index", "Contacts", new { @class = "btn btn-primary btn-xs" })
@Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-xs" })
</div>
</div>

<div class="ibox-content">

    <input type="text" class="form-control
input-sm m-b-xs" id="filter" placeholder="Search
Contact">
    <table class="footable table table-hover"
data-filter="#filter data-page-size="5">
        <thead>
            <tr>
                <th>
                    @Html.DisplayNameFor(model
=> model.ContactName)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.ContactDescription)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.ContactType)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.ContactOffice)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.ContactEmail)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.ContactNumber)
                </th>
            <th></th>
        </thead>
        <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>
                        @Html.DisplayFor(modelItem
=> item.ContactName)
                    </td>
                    <td>

```

```

        @Html.DisplayFor(modelItem
=> item.ContactDescription)
    </td>
    <td>
        @Html.DisplayFor(modelItem
=> item.ContactType)
    </td>
    <td>
        @Html.DisplayFor(modelItem
=> item.ContactOffice)
    </td>
    <td>
        @Html.DisplayFor(modelItem
=> item.ContactEmail)
    </td>
    <td>
        @Html.DisplayFor(modelItem
=> item.ContactNumber)
    </td>
    <td>
        @Html.ActionLink("Details",
"Details", new { id = item.ContactID }, new { @class =
"btn btn-primary btn-sm" })
        @Html.ActionLink("Edit",
>Edit", new { id = item.ContactID }, new { @class =
"btn btn-white btn-sm" })
        @Html.ActionLink("Delete",
"Delete", new { id = item.ContactID }, new { @class =
"btn btn-white btn-sm" })
    </td>
</tr>
}
</tbody>

<tfoot>
<tr>
<td colspan="7">
    <ul class="pagination pull-
right"></ul>
    </td>
</tr>
</tfoot>

</table>
</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views/Desktop**  
**Index.cshtml**

```

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row border-bottom white-bg
dashboard-header">

<div class="col-sm-3">
    <h2>Desktop</h2>
    <small>Physical Facilities Management Office -
MIS</small>
    <ul class="list-group clear-list m-t">
        <li class="list-group-item fist-item">
            <span class="pull-right">
                09:00 pm
            </span>
            <span class="label label-success">1</span>
        Please contact me
        </li>
        <li class="list-group-item">
            <span class="pull-right">
                10:16 am
            </span>
            <span class="label label-info">2</span>
        Sign a contract
        </li>
        <li class="list-group-item">
            <span class="pull-right">
                08:22 pm
            </span>
            <span class="label label-primary">3</span> Open new shop
        </li>
        <li class="list-group-item">
            <span class="pull-right">
                11:06 pm
            </span>
            <span class="label label-default">4</span>
        Call back to Sylvia
        </li>
        <li class="list-group-item">
            <span class="pull-right">
                12:00 am
            </span>
            <span class="label label-primary">5</span> Write a letter to Sandra
        </li>
    </ul>
</div>

```

```
<div class="col-sm-6">
  <div class="flot-chart dashboard-chart">
    <div class="flot-chart-content" id="flot-
      dashboard-chart"></div>
    </div>
    <div class="row text-left">
      <div class="col-xs-4">
        <div class="m-l-md">
          <span class="h4 font-bold m-t block">$406,100</span>
          <small class="text-muted m-b block">Sales marketing report</small>
        </div>
      </div>
      <div class="col-xs-4">
        <span class="h4 font-bold m-t block">$150,401</span>
        <small class="text-muted m-b block">Annual sales revenue</small>
      </div>
      <div class="col-xs-4">
        <span class="h4 font-bold m-t block">$16,822</span>
        <small class="text-muted m-b block">Half-year revenue margin</small>
      </div>
    </div>
  </div>
</div>
<div class="col-sm-3">
  <div class="statistic-box">
    <h4>Project Beta progress</h4>
    <p>You have two project with not compleated task.</p>
    </p>
    <div class="row text-center">
      <div class="col-lg-6">
        <canvas id="polarChart" width="80" height="80"></canvas>
        <h5>Kolter</h5>
      </div>
      <div class="col-lg-6">
        <canvas id="doughnutChart" width="78" height="78"></canvas>
        <h5>Maxtor</h5>
      </div>
    </div>
    <div class="m-t">
      <small>Lorem Ipsum is simply dummy text of the printing and typesetting industry.</small>
    </div>
  </div>
</div>
```

```
</div>
</div>
</div>

<div class="wrapper wrapper-content">
  <div class="row">
    <div class="col-lg-2">
      <div class="ibox float-e-margins">
        <div class="ibox-title">
          <span class="label label-success pull-right">Monthly</span>
          <h5>Views</h5>
        </div>
        <div class="ibox-content">
          <h1 class="no-margins">386,200</h1>
          <div class="stat-percent font-bold text-success">98% <i class="fa fa-bolt"></i></div>
          <small>Total views</small>
        </div>
      </div>
    </div>
    <div class="col-lg-2">
      <div class="ibox float-e-margins">
        <div class="ibox-title">
          <span class="label label-info pull-right">Annual</span>
          <h5>Orders</h5>
        </div>
        <div class="ibox-content">
          <h1 class="no-margins">80,800</h1>
          <div class="stat-percent font-bold text-info">20% <i class="fa fa-level-up"></i></div>
          <small>New orders</small>
        </div>
      </div>
    </div>
    <div class="col-lg-4">
      <div class="ibox float-e-margins">
        <div class="ibox-title">
          <span class="label label-primary pull-right">Today</span>
          <h5>Visits</h5>
        </div>
        <div class="ibox-content">
          <div class="row">
            <div class="col-md-6">
              <h1 class="no-margins">406,42</h1>
              <div class="font-bold text-navy">44% <i class="fa fa-level-up"></i></div>
              <small>Rapid pace</small>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        <h1 class="no-
margins">206,12</h1>
        <div class="font-bold text-
navy">22% <i class="fa fa-level-up"></i>
<small>Slow pace</small></div>
        </div>
    </div>

        </div>
    </div>
</div>
<div class="col-lg-4">
    <div class="ibox float-e-margins">
        <div class="ibox-title">
            <h5>Monthly income</h5>
        <div class="ibox-tools">
            <span class="label label-
primary">Updated 12.2015</span>
        </div>
    </div>
    <div class="ibox-content no-padding">
        <div class="flot-chart m-t-lg"
style="height: 55px;">
            <div class="flot-chart-content"
id="flot-chart1"></div>
        </div>
    </div>

        </div>
    </div>
</div>
<div class="row">
    <div class="col-lg-8">
        <div class="ibox float-e-margins">
            <div class="ibox-content">
                <div>
                    <span class="pull-right text-right">
                        <small>Average value of sales in the
past month in: <strong>United
states</strong></small>
                    <br />
                    All sales: 162,862
                </span>
                <h3 class="font-bold no-margins">
                    Half-year revenue margin
                </h3>
                <small>Sales marketing.</small>
            </div>
        <div class="m-t-sm">
            <div class="row">
                <div class="col-md-8">
                    <div>

```

Total orders in period

```

                <div id="lineChart"
height="114"></div>
            </div>
        </div>
        <div class="col-md-4">
            <ul class="stat-list m-t-lg">
                <li>
                    <h2 class="no-
margins">2,346</h2>
                    <small>Total orders in
period</small>
                    <div class="progress progress-
mini">
                        <div class="progress-bar"
style="width: 48%;"></div>
                    </div>
                </li>
                <li>
                    <h2 class="no-margins
">4,422</h2>
                    <small>Orders in last
month</small>
                    <div class="progress progress-
mini">
                        <div class="progress-bar"
style="width: 60%;"></div>
                    </div>
                </li>
            </ul>
        </div>
    </div>

```

Orders in last month

```

            <div class="m-t-md">
                <small class="pull-right">
                    <i class="fa fa-clock-o"></i>
                    Update on 16.07.2015
                </small>
                <small>
                    <strong>Analysis of sales:</strong>
The value has been changed over time, and last
month reached a level over $50,000.
                </small>
            </div>

```

Analysis of sales

```

                </div>
            </div>
        </div>
        <div class="col-lg-4">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <span class="label label-warning pull-
right">Data has changed</span>
                <h5>User activity</h5>

```

```

</div>
<div class="ibox-content">
<div class="row">
<div class="col-xs-4">
<small class="stats-label">Pages / Visit</small>
<h4>236 321.80</h4>
</div>

<div class="col-xs-4">
<small class="stats-label">% New Visits</small>
<h4>46.11%</h4>
</div>
<div class="col-xs-4">
<small class="stats-label">Last week</small>
<h4>432.021</h4>
</div>
</div>
<div class="ibox-content">
<div class="row">
<div class="col-xs-4">
<small class="stats-label">Pages / Visit</small>
<h4>643 321.10</h4>
</div>

<div class="col-xs-4">
<small class="stats-label">% New Visits</small>
<h4>92.43%</h4>
</div>
<div class="col-xs-4">
<small class="stats-label">Last week</small>
<h4>564.554</h4>
</div>
</div>
<div class="ibox-content">
<div class="row">
<div class="col-xs-4">
<small class="stats-label">Pages / Visit</small>
<h4>436 547.20</h4>
</div>

<div class="col-xs-4">
<small class="stats-label">% New Visits</small>
<h4>150.23%</h4>
</div>
<div class="col-xs-4">
<small class="stats-label">Last week</small>
<h4>124.990</h4>
</div>
</div>
</div>
</div>
</div>
</div>

```

PFMO.Web  
Views(EquipmentCategory)  
Create.cshtml

@model PFMO.Entities.EquipmentCategory

```

@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>

```

@div class="wrapper wrapper-content animated fadeInRight"><div class="row"><div class="col-lg-12">

```

<div class="ibox float-e-margins">
    <div class="ibox-title">
        <h5>Create Equipment Category</h5>
    </div>
    <div class="ibox-content">

        @using
        (Html.BeginForm()
        {
            @Html.AntiForgeryToken()

            <div class="form-horizontal">
                @Html.ValidationSummary(true)

                <div class="form-group">
                    <div class="form-group">
                        @Html.LabelFor(model =>
model.EquipmentCategoryName, new { @class =
"control-label col-md-2" })
                        <div class="col-md-9">
                            @Html.EditorFor(model =>
model.EquipmentCategoryName)
                            @Html.ValidationMessageFor(model =>
model.EquipmentCategoryName)
                        </div>
                    </div>
                </div>

                <div class="form-group">
                    @Html.LabelFor(model =>
model.EquipmentCategoryDescription, new {
@class = "control-label col-md-2" })
                    <div class="col-md-9">
                        @Html.EditorFor(model =>
model.EquipmentCategoryDescription)
                        @Html.ValidationMessageFor(model =>
model.EquipmentCategoryDescription)
                    </div>
                </div>

                <div class="form-group">
                    <div class="col-md-offset-2 col-md-10">
                        <input type="submit" value="Create"
class="btn btn-primary" />
                        @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
                    </div>
                </div>
            </div>
        }
    </div>
</div>

```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

@model PFMO.Entities.EquipmentCategory

 @{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeinRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Equipment Category</h5>
                </div>

```

```

<div class="ibox-content">
    <h4>Are you sure you want to
    delete this?</h4>
    <dl class="dl-
horizontal">
        <dt>
            @Html.DisplayNameFor(model =>
model.EquipmentCategoryName)
        </dt>

        <dd>
            @Html.DisplayFor(model =>
model.EquipmentCategoryName)
        </dd>

        <dt>
            @Html.DisplayNameFor(model =>
model.EquipmentCategoryDescription)
        </dt>

        <dd>
            @Html.DisplayFor(model =>
model.EquipmentCategoryDescription)
        </dd>
    </dl>

    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()

        <div class="form-actions no-color">
            <input type="submit" value="Delete"
class="btn btn-primary" />
            @Html.ActionLink("Back to List", "Index",
null, new { @class = "btn btn-white" })
        </div>
    }

    </div>
    </div>
</div>
</div>

```

PFMO.Web  
Views(EquipmentCategories)  
Details.cshtml

@model PFMO.Entities.EquipmentCategory

```

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Details</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Details</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Equipment Category Details</h5>
                </div>
                <div class="ibox-content">

                    <dl class="dl-horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.EquipmentCategoryName)
                        </dt>

                        <dd>
                            @Html.DisplayFor(model =>
model.EquipmentCategoryName)
                        </dd>

                        <dt>

```

```

        @Html.DisplayNameFor(model =>
model.EquipmentCategoryDescription)
    </dt>

    <dd>
        @Html.DisplayFor(model =>
model.EquipmentCategoryDescription)
    </dd>

</dl>

        @Html.ActionLink("Edit", "Edit", new { id =
= Model.EquipmentCategoryID }, new { @class =
"btn btn-primary"})
        @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

    </div>
    </div>
</div>
</div>

```

PFMO.Web  
Views(EquipmentCategory)  
Edit.cshtml

```

@model PFMO.Entities.EquipmentCategory

 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">

```

```

        <strong>Edit</strong>
    </li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Equipment Category</h5>
                </div>
                <div class="ibox-content">

```

```

@using
(Html.BeginForm())
{
    @Html.AntiForgeryToken()

<div class="form-horizontal">

    @Html.ValidationSummary(true)
    @Html.HiddenFor(model =>
model.EquipmentCategoryID)

    <div class="form-group">
        @Html.LabelFor(model =>
model.EquipmentCategoryName, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.EquipmentCategoryName)
            @Html.ValidationMessageFor(model =>
model.EquipmentCategoryName)
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.EquipmentCategoryDescription, new {
@class = "control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.EquipmentCategoryDescription)
            @Html.ValidationMessageFor(model =>
model.EquipmentCategoryDescription)
        </div>
    </div>

    <div class="form-group">

```

```

<div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Save"
    class="btn btn-primary" />

        @Html.ActionLink("Cancel", "Index", null,
    new { @class = "btn btn-white" })
    </div>
    </div>
    </div>
    </div>
}

</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(EquipmentCategory)**  
**Index.cshtml**

```

@model
IEnumerable<PFMO.Entities.EquipmentCategory>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Equipment Categories</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List",
"Index")</strong>
            </li>
        

```

```

        </div>
    </div>

PFMO.Web
Views(EquipmentCategory)
Tables.cshtml

@model
IEnumerable<PFMO.Entities.EquipmentCategory>

{@
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Equipment Categories</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List",
"Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Equipment Categories</h5>
                    <div class="ibox-tools">
                        @Html.ActionLink("Change View",
"Index", "EquipmentCategories", new { @class =
"btn btn-primary btn-xs" })
                        @Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-
xs" })
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

<div class="ibox-content">
    <input type="text" class="form-control input-sm m-b-xs" id="filter" placeholder="Search Category">
    <table class="footable table table-hover" data-filter="#filter data-page-size="5">
        <thead>
            <tr>
                <th>
                    @Html.DisplayNameFor(model => model.EquipmentCategoryName)
                </th>
                <th>
                    @Html.DisplayNameFor(model => model.EquipmentCategoryDescription)
                </th>
            </tr>
        </thead>
        <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>
                        @Html.DisplayFor(modelItem => item.EquipmentCategoryName)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.EquipmentCategoryDescription)
                    </td>
                    <td>
                        @Html.ActionLink("Details",
"Details", new { id = item.EquipmentCategoryID },
new { @class = "btn btn-primary btn-sm" })
                    </td>
                    <td>
                        @Html.ActionLink("Edit",
>Edit", new { id = item.EquipmentCategoryID }, new {
@class = "btn btn-white btn-sm" })
                    </td>
                    <td>
                        @Html.ActionLink("Delete",
"Delete", new { id = item.EquipmentCategoryID },
new { @class = "btn btn-white btn-sm" })
                    </td>
                </tr>
            }
        </tbody>
        <tfoot>
            <tr>
                <td colspan="7">
                    <ul class="pagination pull-right"></ul>
                </td>
            </tr>
        </tfoot>
    </table>
</div>

```





```

PFMO.Web
Views(Equipment)
Delete.cshtml

@model PFMO.Entities.Equipment

{@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Equipment</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Equipment</h5>
                </div>
                <div class="ibox-content">
                    <h4>
                        Are you sure you want to
                        delete this?
                    </h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.EquipmentCategory.EquipmentCategoryNam
e)
                        </dt>
                
```

<dd>  
 `@Html.DisplayFor(model =>`  
`model.EquipmentCategory.EquipmentCategoryNam`  
`e)`  
`</dd>`

<dt>  
 `@Html.DisplayNameFor(model =>`  
`model.EquipmentCode)`  
`</dt>`

<dd>  
 `@Html.DisplayFor(model =>`  
`model.EquipmentCode)`  
`</dd>`

<dt>  
 `@Html.DisplayNameFor(model =>`  
`model.EquipmentName)`  
`</dt>`

<dd>  
 `@Html.DisplayFor(model =>`  
`model.EquipmentName)`  
`</dd>`

<dt>  
 `@Html.DisplayNameFor(model =>`  
`model.EquipmentRemarks)`  
`</dt>`

<dd>  
 `@Html.DisplayFor(model =>`  
`model.EquipmentRemarks)`  
`</dd>`

<dt>  
 `@Html.DisplayNameFor(model =>`  
`model.EquipmentStatus)`  
`</dt>`

<dd>  
 `@Html.DisplayFor(model =>`  
`model.EquipmentStatus)`  
`</dd>`

<dt>  
 `@Html.DisplayNameFor(model =>`  
`model.EquipmentBalance)`  
`</dt>`

<dd>  
 `@Html.DisplayFor(model =>`  
`model.EquipmentBalance)`

```

</dd>
<dt>
    @Html.DisplayNameFor(model =>
model.EquipmentUnit)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.EquipmentUnit)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.EquipmentReorderPoint)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.EquipmentReorderPoint)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.Target)
</dt>

<dd>
    @Html.DisplayFor(model => model.Target)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SupplyExpiryDate)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SupplyExpiryDate)
</dd>

</dl>

@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Delete"
class="btn btn-primary" />
        @Html.ActionLink("Back to List", "Index",
null, new { @class = "btn btn-white" })
    </div>
}

```

PFMO.Web  
Views(Equipment)  
Details.cshtml

```

    </div>
    </div>
    </div>
    </div>
    </div>

    @model PFMO.Entities.Equipment

    @{
        ViewBag.Title = "Details";
        Layout = "~/Views/Shared/_Layout.cshtml";
    }

    <div class="row wrapper border-bottom white-bg
page-heading">
        <div class="col-sm-4">
            <h2>Details</h2>
            <ol class="breadcrumb">
                <li>
                    <a href="#">Home</a>
                </li>
                <li>
                    <a href="#">Panel</a>
                </li>
                <li>
                    <a href="#">Equipment</a>
                </li>
                <li>
                    @Html.ActionLink("List", "Index")
                </li>
                <li class="active">
                    <strong>Details</strong>
                </li>
            </ol>
        </div>
    </div>

```

```

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Equipment Details</h5>
                </div>

```

```

<div class="ibox-content">
    <dl
        class="dl-horizontal">
            <dt>
                @Html.DisplayNameFor(model =>
model.EquipmentCategory.EquipmentCategoryName)
            </dt>
            <dd>
                @Html.DisplayFor(model =>
model.EquipmentCategory.EquipmentCategoryName)
            </dd>
            <dt>
                @Html.DisplayNameFor(model =>
model.EquipmentCode)
            </dt>
            <dd>
                @Html.DisplayFor(model =>
model.EquipmentCode)
            </dd>
            <dt>
                @Html.DisplayNameFor(model =>
model.EquipmentName)
            </dt>
            <dd>
                @Html.DisplayFor(model =>
model.EquipmentName)
            </dd>
            <dt>
                @Html.DisplayNameFor(model =>
model.EquipmentRemarks)
            </dt>
            <dd>
                @Html.DisplayFor(model =>
model.EquipmentRemarks)
            </dd>
            <dt>
                @Html.DisplayNameFor(model =>
model.EquipmentStatus)
            </dt>
            <dd>
                @Html.DisplayFor(model =>
model.EquipmentStatus)
        </dl>
    </div>
    <dt>
        @Html.DisplayNameFor(model =>
model.EquipmentBalance)
    </dt>
    <dd>
        @Html.DisplayFor(model =>
model.EquipmentBalance)
    </dd>
    <dt>
        @Html.DisplayNameFor(model =>
model.EquipmentUnit)
    </dt>
    <dd>
        @Html.DisplayFor(model =>
model.EquipmentUnit)
    </dd>
    <dt>
        @Html.DisplayNameFor(model =>
model.EquipmentReorderPoint)
    </dt>
    <dd>
        @Html.DisplayFor(model =>
model.EquipmentReorderPoint)
    </dd>
    <dt>
        @Html.DisplayNameFor(model =>
model.Target)
    </dt>
    <dd>
        @Html.DisplayFor(model => model.Target)
    </dd>
    <dt>
        @Html.DisplayNameFor(model =>
model.SupplyExpiryDate)
    </dt>
    <dd>
        @Html.DisplayFor(model =>
model.SupplyExpiryDate)
    </dd>
</div>

```

```

        @Html.ActionLink("Edit", "Edit", new { id
= Model.EquipmentID }, new { @class = "btn btn-
primary"})
        @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

        </div>
    </div>
</div>
</div>
</div>
</div>

PFMO.Web
Views(Equipment)
Edit.cshtml

@model PFMO.Entities.EquipmentCategory

 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">

```

```

<div class="col-lg-12">
    <div class="ibox float-e-margins">
        <div class="ibox-title">
            <h5>Edit Equipment Category</h5>
        </div>
        <div class="ibox-content">

```

```

        @using
        (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        @Html.ValidationSummary(true)
        @Html.HiddenFor(model =>
model.EquipmentCategoryID)

        <div class="form-group">
            @Html.LabelFor(model =>
model.EquipmentCategoryName, new { @class =
"control-label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
model.EquipmentCategoryName)
                @Html.ValidationMessageFor(model =>
model.EquipmentCategoryName)
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
model.EquipmentCategoryDescription, new {
@class = "control-label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
model.EquipmentCategoryDescription)
                @Html.ValidationMessageFor(model =>
model.EquipmentCategoryDescription)
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save"
class="btn btn-primary" />

```

```

            @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
        </div>
    </div>
}

```

```

        </div>
    </div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(Equipment)
Index.cshtml

@model IEnumerable<PFMO.Entities.Equipment>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Equipment</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Equipment</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List", "Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Equipment</h5>
                </div>
                <div class="ibox-tools">
                    @Html.ActionLink("Create New", "Create", null, new { @class = "btn btn-primary btn-xs" })
                </div>
            </div>
        </div>
    </div>
</div>

<input type="text" class="form-control input-sm m-b-xs" id="filter" placeholder="Search Equipment">
<table class="footable table table-hover" data-filter="#filter" data-page-size="5">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentCategory.EquipmentCategoryName)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentCode)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentName)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentRemarks)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentStatus)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentBalance)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentUnit)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentReorderPoint)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Target)
            </th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>
                @Html.DisplayFor(model => model.EquipmentCategory.EquipmentCategoryName)
            </td>
            <td>
                @Html.DisplayFor(model => model.EquipmentCode)
            </td>
            <td>
                @Html.DisplayFor(model => model.EquipmentName)
            </td>
            <td>
                @Html.DisplayFor(model => model.EquipmentRemarks)
            </td>
            <td>
                @Html.DisplayFor(model => model.EquipmentStatus)
            </td>
            <td>
                @Html.DisplayFor(model => model.EquipmentBalance)
            </td>
            <td>
                @Html.DisplayFor(model => model.EquipmentUnit)
            </td>
            <td>
                @Html.DisplayFor(model => model.EquipmentReorderPoint)
            </td>
            <td>
                @Html.DisplayFor(model => model.Target)
            </td>
        </tr>
    </tbody>
</table>

```

```

</th>
<th>
    @Html.DisplayNameFor(model
=> model.SupplyExpiryDate)
</th>
<th></th>
</tr>
</thead>

<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem
=>
item.EquipmentCategory.EquipmentCategoryName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.EquipmentCode)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.EquipmentName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.EquipmentRemarks)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.EquipmentStatus)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.EquipmentBalance)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.EquipmentUnit)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.EquipmentReorderPoint)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.Target)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyExpiryDate)
            </td>
            <td>
                @Html.ActionLink("Details",
"Details", new { id = item.EquipmentID }, new {
@class = "btn btn-primary btn-sm" })
                @Html.ActionLink("Edit",
>Edit", new { id = item.EquipmentID }, new { @class
= "btn btn-white btn-sm" })
                @Html.ActionLink("Delete",
"Delete", new { id = item.EquipmentID }, new {
@class = "btn btn-white btn-sm" })
            </td>
        </tr>
    }
</tbody>

<tfoot>
    <tr>
        <td colspan="7">
            <ul class="pagination pull-
right"></ul>
        </td>
    </tr>
</tfoot>

</table>

</div>
</div>
</div>
</div>
</div>
</div>

```

PFMO.Web  
Views(Files)  
Index.cshtml

```

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

PFMO.Web  
Views(Land)  
Index.cshtml

```

@{
    Layout =
"~/Views/Shared/_LayoutLandingPage.cshtml";
}

```

PFMO.Web

```

Views/Home
Index.cshtml

@using Microsoft.AspNet.Identity
@{
    ViewBag.Title = "Home";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<!-- Page Header -->
@*<div class="row wrapper border-bottom white-
bg page-heading">
    <div class="col-lg-10">
        <h2>Welcome @ ViewData["FullName"].</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Startup</a>
            </li>
            <li>
                <strong>Index</strong>
            </li>
        </ol>
    </div>
    <div class="col-lg-2">

    </div>
</div>*@

<!-- Content -->
<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="text-center m-t-lg">
                <h1>
                    Okay, Let's Start. Time to Manage.
                </h1>
                <a href="@Url.Action("Index", "Desktop")"
class="btn btn-primary"><i class="fa fa-
play">&nbsp; Click to Start<i>&nbsp;</i></i></a>
                </div>
            </div>
        </div>
    </div>
</div>

<script>
    $(document).ready(function() {
        setTimeout(function() {
            toastr.options = {
                closeButton: true,
                progressBar: true,
                showMethod: 'slideDown',
                timeOut: 4000
            };
            toastr.success(@(ViewData["FullName"]),
'You are Successfully Logged In!');
        }, 1300);
    });
    var data1 = [
        [0,4],[1,8],[2,5],[3,10],[4,4],[5,16],[6,5],[7,11],[8,6],[9,11],[10,30],[11,10],[12,13],[13,4],[14,3],[15,3],[16,6]
];
    var data2 = [
        [0,1],[1,0],[2,2],[3,0],[4,1],[5,3],[6,1],[7,5],[8,2],[9,3],[10,2],[11,1],[12,0],[13,2],[14,8],[15,0],[16,0]
];
    $('#flot-dashboard-chart').length &&
    $.plot($('#flot-dashboard-chart'), [
        data1, data2
    ],
    {
        series: {
            lines: {
                show: false,
                fill: true
            },
            splines: {
                show: true,
                tension: 0.4,
                lineWidth: 1,
                fill: 0.4
            },
            points: {
                radius: 0,
                show: true
            },
            shadowSize: 2
        },
        grid: {
            hoverable: true,
            clickable: true,
            tickColor: "#d5d5d5",
            borderWidth: 1,
            color: '#d5d5d5'
        },
        colors: ["#1ab394", "#1C84C6"],
        xaxis: {
        },
        yaxis: {
            ticks: 4
        },
        tooltip: false
    });
})

```

```

        }
    );

var doughnutData = [
    {
        value: 300,
        color: "#a3e1d4",
        highlight: "#1ab394",
        label: "App"
    },
    {
        value: 50,
        color: "#dedede",
        highlight: "#1ab394",
        label: "Software"
    },
    {
        value: 100,
        color: "#A4CEE8",
        highlight: "#1ab394",
        label: "Laptop"
    }
];

var doughnutOptions = {
    segmentShowStroke: true,
    segmentStrokeColor: "#fff",
    segmentStrokeWidth: 2,
    percentageInnerCutout: 45, // This is 0 for
Pie charts
    animationSteps: 100,
    animationEasing: "easeOutBounce",
    animateRotate: true,
    animateScale: false
};

var ctx =
document.getElementById("doughnutChart").getContext("2d");
var DoughnutChart = new
Chart(ctx).Doughnut(doughnutData,
doughnutOptions);

var polarData = [
    {
        value: 300,
        color: "#a3e1d4",
        highlight: "#1ab394",
        label: "App"
    },
    {
        value: 140,
        color: "#dedede",
        highlight: "#1ab394",
        label: "Software"
    },
    {
        value: 200,
        color: "#A4CEE8",
        highlight: "#1ab394",
        label: "Laptop"
    }
];

var polarOptions = {
    scaleShowLabelBackdrop: true,
    scaleBackdropColor:
"rgba(255,255,255,0.75)",
    scaleBeginAtZero: true,
    scaleBackdropPaddingY: 1,
    scaleBackdropPaddingX: 1,
    scaleShowLine: true,
    segmentShowStroke: true,
    segmentStrokeColor: "#fff",
    segmentStrokeWidth: 2,
    animationSteps: 100,
    animationEasing: "easeOutBounce",
    animateRotate: true,
    animateScale: false
};

var ctx =
document.getElementById("polarChart").getContext("2d");
var Polarchart = new
Chart(ctx).PolarArea(polarData, polarOptions);

```

**PFMO.Web**  
**Views(Manage)**  
**AddPhoneNumber.cshtml**

```

@model PFMO.Web.Models.AddPhoneNumberViewModel
 @{
    ViewBag.Title = "Phone Number";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>@ViewBag.Title.</h2>

```

```

@using (Html.BeginForm("AddPhoneNumber",
"Manage", FormMethod.Post, new { @class =
"form-horizontal", role = "form" }))
{
    @Html.AntiForgeryToken()
    <h4>Add a phone number</h4>
    <hr />
    @Html.ValidationSummary("", new { @class =
"text-danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.Number, new { @class =
"col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Number, new {
                @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" class="btn btn-default"
value="Submit" />
        </div>
    </div>
}

```

@section Scripts {  
 @Scripts.Render("~/bundles/jqueryval")  
}

### PFMO.Web Views(Manage) ChangePassword.cshtml

```

@model
PFMO.Web.Models.ChangePasswordViewModel
 @{
    ViewBag.Title = "Change Password";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("ChangePassword",
"Manage", FormMethod.Post, new { @class =
"form-horizontal", role = "form" }))
{
    @Html.AntiForgeryToken()
    <h4>Change Password Form</h4>
    <hr />
    @Html.ValidationSummary("", new { @class =
"text-danger" })
    <div class="form-group">

```

```

        @Html.LabelFor(m => m.OldPassword, new {
            @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.PasswordFor(m => m.OldPassword,
new { @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.NewPassword, new {
            @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.PasswordFor(m => m.NewPassword,
new { @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.ConfirmPassword,
new { @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.PasswordFor(m =>
m.ConfirmPassword, new { @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Change
password" class="btn btn-default" />
        </div>
    </div>
}

```

@section Scripts {  
 @Scripts.Render("~/bundles/jqueryval")  
}

### PFMO.Web Views(Manage) Index.cshtml

```

@model PFMO.Web.Models.IndexViewModel
 @{
    ViewBag.Title = "Manage";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>@ViewBag.Title.</h2>

<p class="text-
success">@ViewBag.StatusMessage</p>
<div>
    <h4>Change your account settings</h4>
    <hr />
    <dl class="dl-horizontal">
```

```

<dt>Password:</dt>
<dd>
[
@if (Model.HasPassword)
{
    @Html.ActionLink("Change your
password", "ChangePassword")
}
else
{
    @Html.ActionLink("Create",
"SetPassword")
}
]
</dd>
<dt>External Logins:</dt>
<dd>
@Model.Logins.Count [
    @Html.ActionLink("Manage",
"ManageLogins") ]
</dd>
@*
    Phone Numbers can used as a second factor
of verification in a two-factor authentication
system.

    See <a
    href="http://go.microsoft.com/fwlink/?LinkId=4038
04">this article</a>
        for details on setting up this ASP.NET
application to support two-factor authentication
using SMS.

    Uncomment the following block after you
have set up two-factor authentication
*/
/*
<dt>Phone Number:</dt>
<dd>
    @(Model.PhoneNumber ?? "None") [
        @if (Model.PhoneNumber != null)
    {
        @Html.ActionLink("Change",
"AddPhoneNumber")
        @: &nbsp;|&nbsp;
        @Html.ActionLink("Remove",
"RemovePhoneNumber")
    }
    else
    {
        @Html.ActionLink("Add",
"AddPhoneNumber")
    }
]
</dd>
</dl>
</div>
*/

```

There are no two-factor authentication providers configured. See <a href="http://go.microsoft.com/fwlink/?LinkId=403804">this article</a> for details on setting up this ASP.NET application to support two-factor authentication.

@\*  
@if (Model.TwoFactor)
{
 using  
(Html.BeginForm("DisableTwoFactorAuthentication",
"Manage", FormMethod.Post, new { @class =
"form-horizontal", role = "form" }))
{
 @Html.AntiForgeryToken()
 <text>Enabled
 <input type="submit" value="Disable"
class="btn btn-link" />
 </text>
}
}
else
{
 using  
(Html.BeginForm("EnableTwoFactorAuthentication",
"Manage", FormMethod.Post, new { @class =
"form-horizontal", role = "form" }))
{
 @Html.AntiForgeryToken()
 <text>Disabled
 <input type="submit" value="Enable"
class="btn btn-link" />
 </text>
}
}
\*/@

### PFMO.Web Views(Manage) ManageLogins.cshtml

```

@model
PFMO.Web.Models.ManageLoginsViewModel
@using Microsoft.Owin.Security
 @{
    ViewBag.Title = "Manage your external logins";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```

    }
}

<h2>@ViewBag.Title.</h2>

<p class="text-success">@ViewBag.StatusMessage</p>
@{
    var loginProviders =
    Context.GetOwinContext().Authentication.GetExternalAuthenticationTypes();
    if (loginProviders.Count() == 0) {
        <div>
            <p>
                There are no external authentication services configured. See <a href="http://go.microsoft.com/fwlink/?LinkId=313242">this article</a> for details on setting up this ASP.NET application to support logging in via external services.
            </p>
        </div>
    }
    else
    {
        if (Model.CurrentLogins.Count > 0)
        {
            <h4>Registered Logins</h4>
            <table class="table">
                <tbody>
                    @foreach (var account in Model.CurrentLogins)
                    {
                        <tr>
                            <td>@account.LoginProvider</td>
                            <td>
                                @if
                                (ViewBag.ShowRemoveButton)
                                {
                                    using
                                    (Html.BeginForm("RemoveLogin", "Manage"))
                                    {
                                        @Html.AntiForgeryToken()
                                        <div>
                                            <@Html.Hidden("loginProvider", account.LoginProvider)>
                                            <@Html.Hidden("providerKey", account.ProviderKey)>
                                            <input type="submit" class="btn btn-default" value="Remove" title="Remove this @account.LoginProvider login from your account" />
                                        </div>
                                    }
                                }
                            </td>
                        </tr>
                    }
                </tbody>
            </table>
        }
    }
}
}

<table border="1">
    <thead>
        <tr>
            <th>Provider</th>
            <th>Actions</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var provider in Model.ExternalLogins)
        {
            <tr>
                <td>@provider.AuthenticationType</td>
                <td>
                    @:@nbsp;
                </td>
            </tr>
        }
    </tbody>
</table>

```

**PFMO.Web**  
**Views(Manage)**  
**SetPassword.cshtml**

```

    @model PFMO.Web.Models.SetPasswordViewModel
    @{
        ViewBag.Title = "Create Password";
        Layout = "~/Views/Shared/_Layout.cshtml";
    }

    <h2>@ViewBag.Title.</h2>
    <p class="text-info">
        You do not have a local username/password for this site. Add a local account so you can log in without an external login.
    </p>

```

```

@using (Html.BeginForm("SetPassword", "Manage",
FormMethod.Post, new { @class = "form-
horizontal", role = "form" }))
{
    @Html.AntiForgeryToken()

    <h4>Create Local Login</h4>
    <hr />
    @Html.ValidationSummary("", new { @class =
"text-danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.NewPassword, new {
@class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.PasswordFor(m => m.NewPassword,
new { @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.ConfirmPassword,
new { @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.PasswordFor(m =>
m.ConfirmPassword, new { @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Set password"
class="btn btn-default" />
        </div>
    </div>
}
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(Manage)**  
**VerifyPhoneNumber.cshtml**

```

@model PFMO.Web.Models.VerifyPhoneNumberViewModel
 @{
    ViewBag.Title = "Verify Phone Number";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>@ViewBag.Title.</h2>

```

```

@using (Html.BeginForm("VerifyPhoneNumber",
"Manage", FormMethod.Post, new { @class =
"form-horizontal", role = "form" }))
{
    @Html.AntiForgeryToken()
    @Html.Hidden("phoneNumber",
@Model.PhoneNumber)
    <h4>Enter verification code</h4>
    <h5>@ViewBag.Status</h5>
    <hr />
    @Html.ValidationSummary("", new { @class =
"text-danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.Code, new { @class =
"col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Code, new {
@class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" class="btn btn-default"
value="Submit" />
        </div>
    </div>
}

```

**@section Scripts {  
 @Scripts.Render("~/bundles/jqueryval")  
}**

**PFMO.Web**  
**Views(PersonnelDesignations)**  
**Creat.cshtml**

```

@model PFMO.Entities.PersonnelDesignation

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
        </ol>
    </div>

```

```

<li>
    <a href="#">Panel</a>
</li>
<li>
    <a href="#">Designations</a>
</li>
<li>
    @Html.ActionLink("List", "Index")
</li>
<li class="active">
    <strong>Create</strong>
</li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
    <h5>Create Personnel Designation</h5>
</div>
<div class="ibox-content">
    @using
    (Html.BeginForm())
    {
        @Html.AntiForgeryToken()
    }

    <div class="form-horizontal">
        @Html.ValidationSummary(true)

        <div class="form-group">
            <div class="form-group">
                @Html.LabelFor(model =>
model.PersonnelDesignationName, new { @class =
"control-label col-md-2" })
                <div class="col-md-9">
                    @Html.EditorFor(model =>
model.PersonnelDesignationName)
                    @Html.ValidationMessageFor(model =>
model.PersonnelDesignationName)
                </div>
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
model.PersonnelDesignationDescription, new {
@class = "control-label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
model.PersonnelDesignationDescription)
                @Html.ValidationMessageFor(model =>
model.PersonnelDesignationDescription)
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-4">
            <h2>Delete</h2>
            <ol class="breadcrumb">
                <li>
                    <a href="#">Home</a>
                </li>
                <li>
                    <a href="#">Panel</a>
                </li>
            </ol>
        </div>
        <div class="col-sm-8">
            @Html.ValidationSummary(true)
            <div class="form-group">
                <div class="col-sm-12">
                    <input type="submit" value="Create"
class="btn btn-primary" />
                </div>
            </div>
            <div class="form-group">
                <div class="col-sm-12">
                    @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
                </div>
            </div>
        </div>
    </div>
</div>

```



```

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Personnel Designation Details</h5>
                </div>
                <div class="ibox-content">

                    <dl
                    class="dl-horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.PersonnelDesignationName)
                        </dt>

                        <dd>
                            @Html.DisplayFor(model =>
model.PersonnelDesignationName)
                        </dd>

                        <dt>
                            @Html.DisplayNameFor(model =>
model.PersonnelDesignationDescription)
                        </dt>

                        <dd>
                            @Html.DisplayFor(model =>
model.PersonnelDesignationDescription)
                        </dd>

                    </dl>

                    @Html.ActionLink("Edit", "Edit", new { id =
Model.PersonnelDesignationID }, new { @class =
"btn btn-primary"})
                    @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

                </div>
            </div>
        </div>
    </div>
</div>

@model PFMO.Entities.PersonnelDesignation
@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Designations</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Personnel Designation</h5>
                </div>
                <div class="ibox-content">

                    @using
(Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">

        @Html.ValidationSummary(true)
        @Html.HiddenFor(model =>
model.PersonnelDesignationID)

        <div class="form-group">

```

```

        @Html.LabelFor(model =>
model.PersonnelDesignationName, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelDesignationName)
        @Html.ValidationMessageFor(model =>
model.PersonnelDesignationName)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PersonnelDesignationDescription, new {
@class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelDesignationDescription)
        @Html.ValidationMessageFor(model =>
model.PersonnelDesignationDescription)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Save"
class="btn btn-primary" />
    @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>
</div>
}

</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(PersonnelDesignation)**  
**Index.cshtml**

```

@model
IEnumerable<PFMO.Entities.PersonnelDesignation>

```

```

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Personnel Designations</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Designations</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List",
"Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>List of Personnel Designations</h5>
            <div class="ibox-tools">
                @Html.ActionLink("Change View",
"Table", "PersonnelDesignations", new { @class =
"btn btn-primary btn-xs" })
                @Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-
xs" })
            </div>
        </div>
    </div>
</div>
@foreach (var item in Model)
{
    <div class="col-lg-3">
        <div class="contact-box center-version">
            <a href="#">
                

```

```

<h3 class="m-b-0">
    <strong>@Html.DisplayFor(modelItem =>
    item.PersonnelDesignationName)</strong></h3>

    <div class="font-weight-bold">
        @Html.DisplayFor(modelItem =>
        item.PersonnelDesignationDescription)</div>
    </a>

    <div class="contact-box-footer">
        <div class="m-t-xs btn-group">
            @Html.ActionLink("Details",
            "Details", new { id = item.PersonnelDesignationID },
            new { @class = "btn btn-xs btn-white" })
            @Html.ActionLink("Edit", "Edit",
            new { id = item.PersonnelDesignationID }, new {
            @class = "btn btn-xs btn-white" })
            @Html.ActionLink("Delete",
            "Delete", new { id = item.PersonnelDesignationID },
            new { @class = "btn btn-xs btn-white" })
        </div>
        </div>
    </div>
}
</div>
</div>

```

**PFMO.Web**  
**Views(PersonnelDesignation)**  
**Tables.cshtml**

```

@model
IEnumerable<PFMO.Entities.PersonnelDesignation>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Personnel Designations</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>

```

```

                <a href="#">Designations</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List",
                "Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Personnel Designations</h5>
                    <div class="ibox-tools">
                        @Html.ActionLink("Change View",
                        "Index", "PersonnelDesignations", new { @class =
                        "btn btn-primary btn-xs" })
                        @Html.ActionLink("Create New",
                        "Create", null, new { @class = "btn btn-primary btn-
                        xs" })
                    </div>
                </div>
                <div class="ibox-content">
                    <input type="text" class="form-control
                    input-sm m-b-xs" id="filter" placeholder="Search
                    Designation">
                    <table class="footable table table-hover"
                    data-filter="#filter data-page-size="5">
                        <thead>
                            <tr>
                                <th>
                                    @Html.DisplayNameFor(model
                                    => model.PersonnelDesignationName)
                                </th>
                                <th>
                                    @Html.DisplayNameFor(model
                                    => model.PersonnelDesignationDescription)
                                </th>
                                <th></th>
                            </tr>
                        </thead>
                        <tbody>
                            @foreach (var item in Model)
                            {
                                <tr>
                                    <td>
                                        @Html.DisplayFor(modelItem
                                        => item.PersonnelDesignationName)

```

```

        </td>
        <td>
            @Html.DisplayFor(modelItem
=> item.PersonnelDesignationDescription)
        </td>
        <td>
            @Html.ActionLink("Details",
"Details", new { id = item.PersonnelDesignationID },
new { @class = "btn btn-primary btn-sm" })
            @Html.ActionLink("Edit",
>Edit", new { id = item.PersonnelDesignationID },
new { @class = "btn btn-white btn-sm" })
            @Html.ActionLink("Delete",
"Delete", new { id = item.PersonnelDesignationID },
new { @class = "btn btn-white btn-sm" })
        </td>
    </tr>
}
</tbody>

<tfoot>
<tr>
<td colspan="7">
    <ul class="pagination pull-right"></ul>
</td>
</tr>
</tfoot>

</table>
</div>
</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(Personnel)**  
**Create.cshtml**

```

@model PFMO.Entities.Personnel

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Create Personnel</h5>
                </div>
                <div class="ibox-content">

```

```

@using
(Html.BeginForm())
{
    @Html.AntiForgeryToken()

<div class="form-horizontal">
    @Html.ValidationSummary(true)

    <div class="form-group">
        <div class="form-group">
            @Html.LabelFor(model =>
model.PersonnelLastName, new { @class = "control-
label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
model.PersonnelLastName)
                @Html.ValidationMessageFor(model =>
model.PersonnelLastName)
            </div>
        </div>
    <div class="form-group">
        @Html.LabelFor(model =>
model.PersonnelFirstName, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">

```

```

        @Html.EditorFor(model =>
model.PersonnelFirstName)
        @Html.ValidationMessageFor(model =>
model.PersonnelFirstName)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PersonnelGender, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelGender)
        @Html.ValidationMessageFor(model =>
model.PersonnelGender)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Designation</label>
    @* @Html.LabelFor(model =>
model.PersonnelDesignationID,
"PersonnelDesignationID", new { @class = "control-label col-md-2" })*@
    <div class="col-md-9">
        @Html.DropDownList("PersonnelDesignationID",
String.Empty)
        @Html.ValidationMessageFor(model =>
model.PersonnelDesignationID)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PersonnelRemarks, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelRemarks)
        @Html.ValidationMessageFor(model =>
model.PersonnelRemarks)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Section</label>
    @* @Html.LabelFor(model =>
model.SectionID, "SectionID", new { @class =
"control-label col-md-2" })*@
    <div class="col-md-9">
        @Html.DropDownList("SectionID",
String.Empty)
        @Html.ValidationMessageFor(model =>
model.SectionID)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Project Team</label>
    @* @Html.LabelFor(model =>
model.ProjectTeamID, "ProjectTeamID", new {
@class = "control-label col-md-2" })*@
    <div class="col-md-9">
        @Html.DropDownList("ProjectTeamID",
String.Empty)
        @Html.ValidationMessageFor(model =>
model.ProjectTeamID)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PersonnelHireDate, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelHireDate)
        @Html.ValidationMessageFor(model =>
model.PersonnelHireDate)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PersonnelContactNumber, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelContactNumber)
        @Html.ValidationMessageFor(model =>
model.PersonnelContactNumber)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Create"
class="btn btn-primary" />
        @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>
</div>

```

```

        </div>
    </div>
}
</div>
</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(Personnels)
Delete.cshtml

@model PFMO.Entities.Personnel

 @{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>

```

```

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Personnel</h5>
                </div>
                <div class="ibox-content">
                    <h4
                        class="text-danger">Are you sure you want to
                        delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.PersonnelDesignation.PersonnelDesignation
Name)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.PersonnelDesignation.PersonnelDesignation
Name)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.ProjectTeam.ProjectTeamName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.ProjectTeam.ProjectTeamName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.Section.SectionName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.Section.SectionName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.PersonnelLastName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.PersonnelLastName)
                        </dd>

```

```

<dt>
    @Html.DisplayNameFor(model =>
model.PersonnelFirstName)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.PersonnelFirstName)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.PersonnelGender)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.PersonnelGender)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.PersonnelRemarks)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.PersonnelRemarks)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.PersonnelHireDate)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.PersonnelHireDate)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.PersonnelContactNumber)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.PersonnelContactNumber)
</dd>

</dl>

@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

```

```

<div class="form-actions no-color">
    <input type="submit" value="Delete" class="btn btn-primary" />
    @Html.ActionLink("Back to List", "Index", null, new { @class = "btn btn-white" })
</div>
}

</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(Personnels)**  
**Details.cshtml**

```

@model PFMO.Entities.Personnel

 @{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Details</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Details</strong>
            </li>
        </ol>
    </div>

```

```

<div class="wrapper wrapper-content animated
fadeInRight">
  <div class="row">
    <div class="col-lg-12">
      <div class="ibox float-e-margins">
        <div class="ibox-title">
          <h5>Personnel Details</h5>
        </div>
        <div class="ibox-content">

          <dl
            class="dl-horizontal">
            <dt>
              @Html.DisplayNameFor(model =>
model.PersonnelDesignation.PersonnelDesignation
Name)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.PersonnelDesignation.PersonnelDesignation
Name)
            </dd>

            <dt>
              @Html.DisplayNameFor(model =>
model.ProjectTeam.ProjectTeamName)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.ProjectTeam.ProjectTeamName)
            </dd>

            <dt>
              @Html.DisplayNameFor(model =>
model.Section.SectionName)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.Section.SectionName)
            </dd>

            <dt>
              @Html.DisplayNameFor(model =>
model.PersonnelLastName)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.PersonnelLastName)
            </dd>

            <dt>
              @Html.DisplayNameFor(model =>
model.PersonnelFirstName)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.PersonnelFirstName)
            </dd>

            <dt>
              @Html.DisplayNameFor(model =>
model.PersonnelGender)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.PersonnelGender)
            </dd>

            <dt>
              @Html.DisplayNameFor(model =>
model.PersonnelRemarks)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.PersonnelRemarks)
            </dd>

            <dt>
              @Html.DisplayNameFor(model =>
model.PersonnelHireDate)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.PersonnelHireDate)
            </dd>

            <dt>
              @Html.DisplayNameFor(model =>
model.PersonnelContactNumber)
            </dt>

            <dd>
              @Html.DisplayFor(model =>
model.PersonnelContactNumber)
            </dd>

          </dl>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        @Html.ActionLink("Edit", "Edit", new { id =
= Model.PersonnelID }, new { @class = "btn btn-
primary"})
        @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

        </div>
        </div>
        </div>
    </div>
</div>

```

**PFMO.Web**  
**Views(Personnels)**  
**Edit.cshtml**

```

@model PFMO.Entities.Personnel

 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

```

```

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Personnel</h5>
                </div>
                <div class="ibox-content">

                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()

                        <div class="form-horizontal">

                            @Html.ValidationSummary(true)
                            @Html.HiddenFor(model =>
model.PersonnelID)

                            <div class="form-group">
                                @Html.LabelFor(model =>
model.PersonnelLastName, new { @class = "control-
label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.PersonnelLastName)
                                    @Html.ValidationMessageFor(model =>
model.PersonnelLastName)
                                </div>
                            </div>

                            <div class="form-group">
                                @Html.LabelFor(model =>
model.PersonnelFirstName, new { @class =
"control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.PersonnelFirstName)
                                    @Html.ValidationMessageFor(model =>
model.PersonnelFirstName)
                                </div>
                            </div>

                            <div class="form-group">
                                @Html.LabelFor(model =>
model.PersonnelGender, new { @class = "control-
label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.PersonnelGender)
                                    @Html.ValidationMessageFor(model =>
model.PersonnelGender)
                                </div>
                            </div>

```

```

    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Designation</label>
        @*@Html.LabelFor(model =>
model.PersonnelDesignationID,
"PersonnelDesignationID", new { @class = "control-
label col-md-2" })*@
    <div class="col-md-9">

        @Html.DropDownList("PersonnelDesignationID",
String.Empty)
            @Html.ValidationMessageFor(model =>
model.PersonnelDesignationID)
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.PersonnelRemarks, new { @class = "control-
label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.PersonnelRemarks)
            @Html.ValidationMessageFor(model =>
model.PersonnelRemarks)
        </div>
    </div>

    <div class="form-group">
        <label class="control-label col-md-2">Section</label>
        @*@Html.LabelFor(model =>
model.SectionID, "SectionID", new { @class =
"control-label col-md-2" })*@
        <div class="col-md-9">
            @Html.DropDownList("SectionID",
String.Empty)
            @Html.ValidationMessageFor(model =>
model.SectionID)
        </div>
    </div>

    <div class="form-group">
        <label class="control-label col-md-2">Project
Team</label>
        @*@Html.LabelFor(model =>
model.ProjectTeamID, "ProjectTeamID", new {
@class = "control-label col-md-2" })*@
        <div class="col-md-9">
            @Html.DropDownList("ProjectTeamID",
String.Empty)
        </div>
    </div>

    <div class="form-group">
        @div class="col-md-2" style="text-align: right;">
            @Html.LabelFor(model =>
model.ProjectTeamID)
        </div>
        <div class="col-md-9">
            @div class="form-group">
                @Html.LabelFor(model =>
model.PersonnelHireDate, new { @class = "control-
label col-md-2" })
                <div class="col-md-9">
                    @Html.EditorFor(model =>
model.PersonnelHireDate)
                    @Html.ValidationMessageFor(model =>
model.PersonnelHireDate)
                </div>
            </div>

            <div class="form-group">
                @Html.LabelFor(model =>
model.PersonnelContactNumber, new { @class =
"control-label col-md-2" })
                <div class="col-md-9">
                    @Html.EditorFor(model =>
model.PersonnelContactNumber)
                    @Html.ValidationMessageFor(model =>
model.PersonnelContactNumber)
                </div>
            </div>

            <div class="form-group">
                <div class="col-md-offset-2 col-md-10">
                    <input type="submit" value="Save"
class="btn btn-primary" />
                </div>
            </div>

            @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
        </div>
        </div>
    </div>
}

</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(Personnels)**  
**Index.cshtml**

```
@model IEnumerable<PFMO.Entities.Personnel>

{@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Personnel</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel</a>
            </li>
            <li class="active">
                <strong>List</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>List of Personnels</h5>
            <div class="ibox-tools">
                @Html.ActionLink("Change View", "Table", "Personnels", new { @class = "btn btn-primary btn-xs" })
                @Html.ActionLink("Create New", "Create", null, new { @class = "btn btn-primary btn-xs" })
            </div>
        </div>
    </div>
    @foreach (var item in Model)
    {
        <div class="col-lg-3">
            <div class="contact-box center-version">
```

```
<a href="#">
    
</a>
<h3 class="m-b"><strong>@Html.DisplayFor(modelItem => item.PersonnelFullName)</strong></h3>
<div class="font-bold"><strong>@Html.DisplayFor(modelItem => item.PersonnelDesignation.PersonnelDesignationName)</strong></div>
<div class="contact-box-footer">
    <div class="m-t-xs btn-group">
        @Html.ActionLink("Details", "Details", new { id = item.SectionID }, new { @class = "btn btn-xs btn-white" })
        @Html.ActionLink("Edit", "Edit", new { id = item.SectionID }, new { @class = "btn btn-xs btn-white" })
        @Html.ActionLink("Delete", "Delete", new { id = item.SectionID }, new { @class = "btn btn-xs btn-white" })
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
```

**PFMO.Web**  
**Views(Personnels)**  
**Tables.cshtml**

```
@model IEnumerable<PFMO.Entities.Personnel>

{@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Personnel</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
```

```

</li>
<li>
    <a href="#">Panel</a>
</li>
<li>
    <a href="#">Personnel</a>
</li>
<li class="active">
    <strong>List</strong>
</li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Personnels</h5>
                <div class="ibox-tools">
                    @Html.ActionLink("Change View",
                    "Index", "Personnels", new { @class = "btn btn-primary btn-xs" })
                    @Html.ActionLink("Create New",
                    "Create", null, new { @class = "btn btn-primary btn-xs" })
                </div>
            </div>
            <div class="ibox-content">

                <input type="text" class="form-control
                input-sm m-b-xs" id="filter" placeholder="Search
                Personnel">
                <table class="footable table table-hover"
                data-filter="#filter data-page-size="5">

                    <thead>
                        <tr>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.PersonnelDesignation.PersonnelDesignation
                                Name)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.ProjectTeam.ProjectTeamName)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.Section.SectionName)
                            </th>
                    </thead>
                    <tbody>
                        <tr>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.PersonnelLastname)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.PersonnelFirstName)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.PersonnelGender)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.PersonnelRemarks)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.PersonnelHireDate)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                =>
                                model.PersonnelContactNumber)
                            </th>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>

```

```

<td>
    @Html.DisplayFor(modelItem
=> item.PersonnelGender)
</td>
<td>
    @Html.DisplayFor(modelItem
=> item.PersonnelRemarks)
</td>
<td>
    @Html.DisplayFor(modelItem
=> item.PersonnelHireDate)
</td>
<td>
    @Html.DisplayFor(modelItem
=> item.PersonnelContactNumber)
</td>
<td>
    @Html.ActionLink("Details",
"Details", new { id = item.PersonnelID }, new {
@class = "btn btn-primary btn-sm" })
    @Html.ActionLink("Edit",
>Edit", new { id = item.PersonnelID }, new { @class =
"btn btn-white btn-sm" })
    @Html.ActionLink("Delete",
"Delete", new { id = item.PersonnelID }, new {
@class = "btn btn-white btn-sm" })
</td>
</tr>
}
</tbody>

<tfoot>
<tr>
<td colspan="7">
    <ul class="pagination pull-
right"></ul>
</td>
</tr>
</tfoot>
</table>

</div>
</div>
</div>
</div>
</div>

```

@model PFMO.Entities.PersonnelTask

```

@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel Task</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>
```

@using

```

(Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        @Html.ValidationSummary(true)

        <div class="form-group">
            <div class="form-group">
```

PFMO.Web

Views(PersonnelTask)

Create.cshtml

```

        @Html.LabelFor(model =>
model.PersonnelTaskTitle, new { @class = "control-
label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelTaskTitle)
        @Html.ValidationMessageFor(model =>
model.PersonnelTaskTitle)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PersonnelTaskDescription, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelTaskDescription)
        @Html.ValidationMessageFor(model =>
model.PersonnelTaskDescription)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-
2">Section</label>
    <div class="col-md-9">
        @Html.DropDownList("PersonnelID",
String.Empty)
        @Html.ValidationMessageFor(model =>
model.PersonnelID)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PersonnelTaskStartDate, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelTaskStartDate)
        @Html.ValidationMessageFor(model =>
model.PersonnelTaskStartDate)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PersonnelTaskEndDate, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PersonnelTaskEndDate)
        @Html.ValidationMessageFor(model =>
model.PersonnelTaskEndDate)
    </div>
</div>

```

```

        </div>
    </div>

    <div class="form-group">
        <input type="submit" value="Create"
class="btn btn-primary" />
    </div>

```

```

        @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>

```

```

        </div>
    }
</div>
</div>
</div>

```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(PersonnelTask)**  
**Delete.cshtml**

```

@model PFMO.Entities.PersonnelTask

```

```

 @{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel Task</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
        

```

```

</li>
<li class="active">
    <strong>Delete</strong>
</li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Personnel Task</h5>
                </div>
                <div class="ibox-content">
                    <h4
class="text-danger">Are you sure you want to
delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.Personnel.PersonnelLastName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.Personnel.PersonnelLastName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.PersonnelTaskTitle)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.PersonnelTaskTitle)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.PersonnelTaskDescription)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.PersonnelTaskDescription)
                        </dd>
                    <dt>
                </div>
                </div>
                </div>
                </div>
                <dt>
                    @Html.DisplayNameFor(model =>
model.PersonnelTaskStartDate)
                </dt>
                <dd>
                    @Html.DisplayFor(model =>
model.PersonnelTaskStartDate)
                </dd>
                <dt>
                    @Html.DisplayNameFor(model =>
model.PersonnelTaskEndDate)
                </dt>
                <dd>
                    @Html.DisplayFor(model =>
model.PersonnelTaskEndDate)
                </dd>
            </div>
            <dl class="form-actions no-color">
                <input type="submit" value="Delete"
class="btn btn-primary" />
                @Html.ActionLink("Back to List", "Index",
null, new { @class = "btn btn-white" })
            </div>
        </div>
    </div>
</div>

```

**PFMO.Web**  
**Views(PersonnelTask)**  
**Details.cshtml**

```

@model PFMO.Entities.PersonnelTask
 @{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">

```

```

<h2>Details</h2>
<ol class="breadcrumb">
    <li>
        <a href="#">Home</a>
    </li>
    <li>
        <a href="#">Panel</a>
    </li>
    <li>
        <a href="#">Personnel Task</a>
    </li>
    <li>
        @Html.ActionLink("List", "Index")
    </li>
    <li class="active">
        <strong>Details</strong>
    </li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Personnel Task Details</h5>
                </div>
                <div class="ibox-content">
                    <dl
class="dl-horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.Personnel.PersonnelLastName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.Personnel.PersonnelLastName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.PersonnelTaskTitle)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.PersonnelTaskTitle)
                        </dd>
                    </dl>
                    <dt>
                        @Html.DisplayNameFor(model =>
model.PersonnelTaskDescription)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model =>
model.PersonnelTaskDescription)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model =>
model.PersonnelTaskStartDate)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model =>
model.PersonnelTaskStartDate)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model =>
model.PersonnelTaskEndDate)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model =>
model.PersonnelTaskEndDate)
                    </dd>
                </div>
                <div>
                    @Html.ActionLink("Edit", "Edit", new { id =
Model.PersonnelTaskID }, new { @class = "btn btn-primary"})
                    @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})
                </div>
            </div>
        </div>
    </div>
</div>

```

```

PFMO.Web
Views(PersonnelTask)
Edit.cshtml

@model PFMO.Entities.PersonnelTask

@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel Task</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Personnel Task</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()
                        <div class="form-horizontal">
                            @Html.ValidationSummary(true)
                            @Html.HiddenFor(model =>
                            model.PersonnelTaskID)

                            <div class="form-group">
                                @Html.LabelFor(model =>
                                model.PersonnelTaskTitle, new { @class = "control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
                                    model.PersonnelTaskTitle)
                                    @Html.ValidationMessageFor(model =>
                                    model.PersonnelTaskTitle)
                                </div>
                            </div>

                            <div class="form-group">
                                @Html.LabelFor(model =>
                                model.PersonnelTaskDescription, new { @class = "control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
                                    model.PersonnelTaskDescription)
                                    @Html.ValidationMessageFor(model =>
                                    model.PersonnelTaskDescription)
                                </div>
                            </div>

                            <div class="form-group">
                                <label class="control-label col-md-2">Personnel</label>
                                <div class="col-md-9">
                                    @Html.DropDownList("PersonnelID",
                                    String.Empty)
                                    @Html.ValidationMessageFor(model =>
                                    model.PersonnelID)
                                </div>
                            </div>

                            <div class="form-group">
                                @Html.LabelFor(model =>
                                model.PersonnelTaskStartDate, new { @class = "control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
                                    model.PersonnelTaskStartDate)
                                    @Html.ValidationMessageFor(model =>
                                    model.PersonnelTaskStartDate)
                                </div>
                            </div>

                            <div class="form-group">
                                @Html.LabelFor(model =>
                                model.PersonnelTaskEndDate, new { @class = "control-label col-md-2" })
                            </div>
                        </div>
                    }
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<div class="col-md-9">
    @Html.EditorFor(model =>
model.PersonnelTaskEndDate)
    @Html.ValidationMessageFor(model =>
model.PersonnelTaskEndDate)
</div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Save"
class="btn btn-primary" />

        @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(Personneltask)
Index.cshtml

@model
IEnumerable<PFMO.Entities.PersonnelTask>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Index</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel Task</a>
            </li>
            <li class="active">
                <strong>List</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>List of Personnel Tasks</h5>
            <div class="ibox-tools">
                @Html.ActionLink("Change View",
"Table", "PersonnelTasks", new { @class = "btn btn-
primary btn-xs" })
                @Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-
xs" })
            </div>
        </div>
        </div>
        </div>
        @foreach (var item in Model)
        {
            <div class="col-lg-3">
                <div class="contact-box center-version">
                    <a href="#">
                        
                    <h3 class="m-b
xs"><strong>@Html.DisplayFor(modelItem =>
item.PersonnelTaskTitle)</strong></h3>
                    <div class="font-
bold">@Html.DisplayFor(modelItem =>
item.PersonnelTaskDescription)</div>
                    </a>
                <div class="contact-box-footer">
                    <div class="m-t-xs btn-group">
                        @Html.ActionLink("Details",
"Details", new { id = item.PersonnelTaskID }, new {
@class = "btn btn-xs btn-white" })
                    </div>
                </div>
            </div>
        }
    </div>
</div>

```

```

        @Html.ActionLink("Edit", "Edit",
new { id = item.PersonnelTaskID }, new { @class =
"btn btn-xs btn-white" })
        @Html.ActionLink("Delete",
"Delete", new { id = item.PersonnelTaskID }, new {
@class = "btn btn-xs btn-white" })
    </div>
    </div>
</div>
</div>
}
</div>
</div>

```

**PFMO.Web**  
**Views(PersonnelTask)**  
**Tables.cshtml**

```

@model
IEnumerable<PFMO.Entities.PersonnelTask>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Index</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Personnel Task</a>
            </li>
            <li class="active">
                <strong>List</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">

```

```

<div class="ibox-title">
    <h5>List of Personnel Tasks</h5>
    <div class="ibox-tools">
        @Html.ActionLink("Change View",
"Index", "PersonnelTasks", new { @class = "btn btn-primary btn-xs" })
        @Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-xs" })
    </div>
</div>
<div class="ibox-content">

    <input type="text" class="form-control"
input-sm m-b-xs" id="filter" placeholder="Search
Personnel Task">
    <table class="footable table table-hover"
data-filter=#filter data-page-size="5">
        <thead>
            <tr>
                <th>
                    @Html.DisplayNameFor(model
=> model.Personnel.PersonnelFullName)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.PersonnelTaskTitle)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.PersonnelTaskDescription)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.PersonnelTaskStartDate)
                </th>
                <th>
                    @Html.DisplayNameFor(model
=> model.PersonnelTaskEndDate)
                </th>
            </tr>
        </thead>
        <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>
                        @Html.DisplayFor(modelItem
=> item.Personnel.PersonnelFullName)
                    </td>
                    <td>
```

```

        @Html.DisplayFor(modelItem
=> item.PersonnelTaskTitle)
    </td>
    <td>
        @Html.DisplayFor(modelItem
=> item.PersonnelTaskDescription)
    </td>
    <td>
        @Html.DisplayFor(modelItem
=> item.PersonnelTaskStartDate)
    </td>
    <td>
        @Html.DisplayFor(modelItem
=> item.PersonnelTaskEndDate)
    </td>
    <td>
        @Html.ActionLink("Details",
"Details", new { id = item.PersonnelTaskID }, new {
@class = "btn btn-primary btn-sm" })
        @Html.ActionLink("Edit",
>Edit", new { id = item.PersonnelTaskID }, new {
@class = "btn btn-white btn-sm" })
        @Html.ActionLink("Delete",
"Delete", new { id = item.PersonnelTaskID }, new {
@class = "btn btn-white btn-sm" })
    </td>
</tr>
}
</tbody>

<tfoot>
<tr>
<td colspan="7">
<ul class="pagination pull-right"></ul>
</td>
</tr>
</tfoot>
</table>
</div>
</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(Pinboards)**  
**Create.cshtml**

```

@model PFMO.Entities.Pinboard
@{

```

```

ViewBag.Title = "Create";
Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-4">
<h2>Create</h2>
<ol class="breadcrumb">
<li>
<a href="#">Home</a>
</li>
<li>
<a href="#">Dashboard</a>
</li>
<li>
<a href="#">Pinboard</a>
</li>
<li>
<@Html.ActionLink("List", "Index")>
</li>
<li class="active">
<strong>Create</strong>
</li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>Create Note</h5>
</div>
<div class="ibox-content">
@using
(Html.BeginForm()
{
    @Html.AntiForgeryToken()

<div class="form-horizontal">
    @Html.ValidationSummary(true)

<div class="form-group">
<div class="form-group">
    @Html.LabelFor(model =>
model.PinboardName, new { @class = "control-label col-md-2" })
    <div class="col-md-9">

```

col-md-2" })

```

        @Html.EditorFor(model =>
model.PinboardName)
        @Html.ValidationMessageFor(model =>
model.PinboardName)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PinboardContent, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PinboardContent)
        @Html.ValidationMessageFor(model =>
model.PinboardContent)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.PinboardDate, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.PinboardDate)
        @Html.ValidationMessageFor(model =>
model.PinboardDate)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Create"
class="btn btn-primary" />
        @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>
</div>

        </div>
    }
</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

}

**PFMO.Web**  
**Views(Pinboards)**  
**Delete.cshtml**

```

@model PFMO.Entities.Pinboard

{@
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Dashboard</a>
            </li>
            <li>
                <a href="#">Pinboard</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Note</h5>
                </div>
                <div class="ibox-content">
                    <h4
class="text-danger">Are you sure you want to
delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>

```

```

        @Html.DisplayNameFor(model =>
model.PinboardName)
    </dt>

    <dd>
        @Html.DisplayFor(model =>
model.PinboardName)
    </dd>

    <dt>
        @Html.DisplayNameFor(model =>
model.PinboardContent)
    </dt>

    <dd>
        @Html.DisplayFor(model =>
model.PinboardContent)
    </dd>

    <dt>
        @Html.DisplayNameFor(model =>
model.PinboardDate)
    </dt>

    <dd>
        @Html.DisplayFor(model =>
model.PinboardDate)
    </dd>

</dl>

@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Delete"
class="btn btn-primary" />
        @Html.ActionLink("Back to List", "Index",
null, new { @class = "btn btn-white" })
    </div>
}

</div>
</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(Pinboards)**  
**Index.cshtml**

```

@model IEnumerable<PFMO.Entities.Pinboard>

{@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Pin Board</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Dashboard</a>
            </li>
            <li>
                <a href="#">Pinboard</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List",
"Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>List of Notes</h5>
            <div class="ibox-tools">
                @Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-xs" })
            </div>
        </div>
    </div>
    <div class="notes">
        @foreach (var item in Model)
        {
            <li>
                <div>
                    <small>@Html.DisplayFor(modelItem =>
item.PinboardDate)</small>

```

```

        <h4>@Html.DisplayFor(modelItem => item.PinboardName)</h4>
        <p>@Html.DisplayFor(modelItem => item.PinboardContent)</p>
        @Html.ActionLink(" ", "Delete", new { id = item.PinboardID }, new { @class = "fa fa-trash-o" })
    </div>
</li>
}
</ul>
</div>
</div>

```

**PFMO.Web**  
**Views(Projects)**  
**Create.cshtml**

```

@model PFMO.Entities.Project

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Projects</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>

```

```

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Create Project</h5>
                </div>
                <div class="ibox-content">

                    @using (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()

                        <div class="form-horizontal">
                            @Html.ValidationSummary(true)

                            <div class="form-group">
                                <div class="form-group">
                                    @Html.LabelFor(model => model.ProjectName, new { @class = "control-label col-md-2" })
                                    <div class="col-md-9">
                                        @Html.EditorFor(model => model.ProjectName)
                                        @Html.ValidationMessageFor(model => model.ProjectName)
                                    </div>
                                </div>
                            </div>

                            <div class="form-group">
                                @Html.LabelFor(model => model.ProjectStatus, new { @class = "control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model => model.ProjectStatus)
                                    @Html.ValidationMessageFor(model => model.ProjectStatus)
                                </div>
                            </div>

                            <div class="form-group">
                                @Html.LabelFor(model => model.ProjectCompletion, new { @class = "control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model => model.ProjectCompletion)
                                    @Html.ValidationMessageFor(model => model.ProjectCompletion)
                                </div>
                            </div>
                        </div>
                    }
                </div>
            </div>
        </div>
    </div>

```

```

<div class="form-group">
    @Html.LabelFor(model =>
model.ProjectStartDate, new { @class = "control-
label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.ProjectStartDate)
        @Html.ValidationMessageFor(model =>
model.ProjectStartDate)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.ProjectEndDate, new { @class = "control-
label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.ProjectEndDate)
        @Html.ValidationMessageFor(model =>
model.ProjectEndDate)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Create"
class="btn btn-primary" />
    @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>
</div>

</div>
}
</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}


```

**PFMO.Web**  
**Views(Projects)**  
**Delete.cshtml**

```

@model PFMO.Entities.Project

 @{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Projects</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Project</h5>
                </div>
                <div class="ibox-content">
                    <h4
class="text-danger">Are you sure you want to
delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.ProjectName)
                        </dt>

```



```

<div class="row">
    <div class="col-lg-12">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>Project Details</h5>
            </div>
            <div class="ibox-content">
                <dl
                    class="dl-horizontal">
                    <dt>
                        @Html.DisplayNameFor(model =>
                            model.ProjectName)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model =>
                            model.ProjectName)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model =>
                            model.ProjectStatus)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model =>
                            model.ProjectStatus)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model =>
                            model.ProjectCompletion)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model =>
                            model.ProjectCompletion)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model =>
                            model.ProjectStartDate)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model =>
                            model.ProjectStartDate)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model =>
                            model.ProjectEndDate)
                    </dt>
                </dl>
            </div>
        </div>
    </div>
</div>

```

**PFMO.Web**  
**Views(Projects)**  
**Edit.cshtml**

```

@model PFMO.Entities.Project

 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Projects</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>

```

```

        </div>
    </div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Project</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm()
                    {
                        @Html.AntiForgeryToken()

                        <div class="form-horizontal">

                            @Html.ValidationSummary(true)
                            @Html.HiddenFor(model => model.ProjectID)

                            <div class="form-group">
                                @Html.LabelFor(model =>
model.ProjectName, new { @class = "control-label
col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.ProjectName)
                                    @Html.ValidationMessageFor(model =>
model.ProjectName)
                                </div>
                            </div>
                        </div>

                        <div class="form-group">
                            @Html.LabelFor(model =>
model.ProjectStatus, new { @class = "control-label
col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.ProjectStatus)
                                @Html.ValidationMessageFor(model =>
model.ProjectStatus)
                            </div>
                        </div>

                        <div class="form-group">
                            @Html.LabelFor(model =>
model.ProjectCompletion, new { @class = "control-
label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.ProjectCompletion)
                                @Html.ValidationMessageFor(model =>
model.ProjectCompletion)
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(Projects)**  
**Index.cshtml**

```

@model IEnumerable<PFMO.Entities.Project>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Projects</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Projects</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List", "Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Project</h5>
                </div>
                <div class="ibox-tools">
                    @Html.ActionLink("Create New", "Create", null, new { @class = "btn btn-primary btn-xs" })
                </div>
            </div>
            <div class="ibox-content">

```

```

                <input type="text" class="form-control input-sm m-b-xs" id="filter" placeholder="Search Project">
                <table class="footable table table-hover" data-filter="#filter data-page-size="5">
                    <thead>
                        <tr>
                            <th>
                                @Html.DisplayNameFor(model => model.ProjectName)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.ProjectStatus)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.ProjectCompletion)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.ProjectStartDate)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.ProjectEndDate)
                            </th>
                            <th></th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach (var item in Model)
                        {
                            <tr>
                                <td>
                                    @Html.DisplayFor(modelItem => item.ProjectName)
                                </td>
                                <td>
                                    @Html.DisplayFor(modelItem => item.ProjectStatus)
                                </td>
                                <td>
                                    @Html.DisplayFor(modelItem => item.ProjectCompletion)
                                </td>
                                <td>
                                    @Html.DisplayFor(modelItem => item.ProjectStartDate)
                                </td>
                                <td>
                                    @Html.DisplayFor(modelItem => item.ProjectEndDate)
                                </td>
                                <td></td>
                            </tr>
                        }
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>

```

```

        <td>
            @Html.ActionLink("Details",
"Details", new { id = item.ProjectID }, new { @class =
"btn btn-primary btn-sm" })
            @Html.ActionLink("Edit",
>Edit", new { id = item.ProjectID }, new { @class =
"btn btn-white btn-sm" })
            @Html.ActionLink("Delete",
"Delete", new { id = item.ProjectID }, new { @class =
"btn btn-white btn-sm" })
        </td>
    </tr>
}
</tbody>

<tfoot>
<tr>
<td colspan="7">
    <ul class="pagination pull-
right"></ul>
</td>
</tr>
</tfoot>
</table>

</div>
</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(ProjectTeam)**  
**Create.cshtml**

```

@model PFMO.Entities.ProjectTeam

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
        </ol>
    </div>
    <div class="col-md-8">
        <ul class="list-group">
            <li>
                <a href="#">Project Team</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ul>
    </div>

```

```

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Create Project Team</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()
                    }
                    <div class="form-horizontal">
                        @Html.ValidationSummary(true)
                    <div class="form-group">
                        <div class="form-group">
                            @Html.LabelFor(model =>
model.ProjectTeamName, new { @class = "control-
label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.ProjectTeamName)
                                @Html.ValidationMessageFor(model =>
model.ProjectTeamName)
                            </div>
                        </div>
                    <div class="form-group">
                        @Html.LabelFor(model =>
model.ProjectTeamDescription, new { @class =
"control-label col-md-2" })
                        <div class="col-md-9">
                            @Html.EditorFor(model =>
model.ProjectTeamDescription)
                            @Html.ValidationMessageFor(model =>
model.ProjectTeamDescription)
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>

    <div class="form-group">
        <label class="control-label col-md-2">Assigned to Project</label>
        <div class="col-md-9">
            @Html.DropDownList("ProjectID",
String.Empty)
            @Html.ValidationMessageFor(model =>
model.ProjectID)
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Create"
class="btn btn-primary" />
            @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
        </div>
    </div>

        </div>
    }
</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(ProjectTeam)
Delete.cshtml

@model PFMO.Entities.ProjectTeam

 @{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Project Team</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Project Team</h5>
                </div>
                <div class="ibox-content">
                    <h4
class="text-danger">Are you sure you want to
delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.Project.ProjectName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.Project.ProjectName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.ProjectTeamName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.ProjectTeamName)
                        </dd>

```



```

<dd>
    @Html.DisplayFor(model =>
model.ProjectTeamDescription)
</dd>

</dl>

    @Html.ActionLink("Edit", "Edit", new { id =
Model.ProjectTeamID }, new { @class = "btn btn-
primary"})
    @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

</div>
</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(ProjectTeam)**  
**Edit.cshtml**

```

@model PFMO.Entities.ProjectTeam

 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-
heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Project Team</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>

```

```

</div>
</div>

```

```

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Project Team</h5>
                </div>
                <div class="ibox-content">

```

```

@using
(Html.BeginForm())
{
    @Html.AntiForgeryToken()

<div class="form-horizontal">

    @Html.ValidationSummary(true)
    @Html.HiddenFor(model =>
model.ProjectTeamID)

<div class="form-group">
    @Html.LabelFor(model =>
model.ProjectTeamName, new { @class = "control-
label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.ProjectTeamName)
        @Html.ValidationMessageFor(model =>
model.ProjectTeamName)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.ProjectTeamDescription, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.ProjectTeamDescription)
        @Html.ValidationMessageFor(model =>
model.ProjectTeamDescription)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-
2">Assigned to Project</label>
    <div class="col-md-9">
        @Html.DropDownList("ProjectID",
String.Empty)
    </div>

```



```

        <td>
            @Html.DisplayFor(modelItem
=> item.Project.ProjectName)
        </td>
        <td>
            @Html.DisplayFor(modelItem
=> item.ProjectTeamName)
        </td>
        <td>
            @Html.DisplayFor(modelItem
=> item.ProjectTeamDescription)
        </td>
        <td>
            @Html.ActionLink("Details",
"Details", new { id = item.ProjectTeamID }, new {
@class = "btn btn-primary btn-sm" })
            @Html.ActionLink("Edit",
"Edit", new { id = item.ProjectTeamID }, new {
@class = "btn btn-white btn-sm" })
            @Html.ActionLink("Delete",
"Delete", new { id = item.ProjectTeamID }, new {
@class = "btn btn-white btn-sm" })
        </td>
        </tr>
    }
</tbody>

<tfoot>
    <tr>
        <td colspan="7">
            <ul class="pagination pull-right"></ul>
        </td>
    </tr>
</tfoot>

</table>

        </div>
    </div>
</div>
</div>
</div>

PFMO.Web
Views(Reports)
Index.cshtml

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```

PFMO.Web
Views(SectionHeadDesignation)
Create.cshtml

@model PFMO.Entities.SectionHeadDesignation

{@
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Designations</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Create Section Head Designation</h5>
                </div>
                <div class="ibox-content">

```

```

                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()

```

```

                    <div class="form-horizontal">

```

```

@Html.ValidationSummary(true)

<div class="form-group">
<div class="form-group">
    @Html.LabelFor(model =>
model.SectionHeadDesignationName, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SectionHeadDesignationName)
        @Html.ValidationMessageFor(model =>
model.SectionHeadDesignationName)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SectionHeadDesignationDescription, new {
@class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SectionHeadDesignationDescription)
        @Html.ValidationMessageFor(model =>
model.SectionHeadDesignationDescription)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Create"
class="btn btn-primary" />
    @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>
</div>

</div>
}
</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

PFMO.Web
Views(SectionHeadDesignations)
Delete.cshtml

@model PFMO.Entities.SectionHeadDesignation

 @{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Designations</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Section Head
Designation</h5>
                </div>
                <div class="ibox-content">
                    <h4
class="text-danger">Are you sure you want to
delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionHeadDesignationName)

```

```

</dt>
<dd>
    @Html.DisplayFor(model =>
model.SectionHeadDesignationName)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionHeadDesignationDescription)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionHeadDesignationDescription)
</dd>

</dl>

@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Delete"
class="btn btn-primary" />
        @Html.ActionLink("Back to List", "Index",
null, new { @class = "btn btn-white" })
    </div>
}

</div>
</div>
</div>
</div>
</div>

PFMO.Web
Views(SectionHeadDesignation)
Details.cshtml

@model PFMO.Entities.SectionHeadDesignation

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Details</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Designations</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Details</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Section Head Designation Details</h5>
                </div>
                <div class="ibox-content">
                    <dl
class="dl-horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionHeadDesignationName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.SectionHeadDesignationName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionHeadDesignationDescription)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.SectionHeadDesignationDescription)
                        </dd>
                    </dl>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </dd>

    </dl>

    @Html.ActionLink("Edit", "Edit", new { id
= Model.SectionHeadDesignationID }, new { @class
= "btn btn-primary"})
    @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

        </div>
    </div>
</div>
</div>

```

**PFMO.Web**  
**Views(SectionHeadDesignations)**  
**Edit.cshtml**

```

@model PFMO.Entities.SectionHeadDesignation

 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Designations</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

```

```

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Section Head Designation</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm())
{
    @Html.AntiForgeryToken()

<div class="form-horizontal">
    @Html.ValidationSummary(true)
    @Html.HiddenFor(model =>
model.SectionHeadDesignationID)

        <div class="form-group">
            @Html.LabelFor(model =>
model.SectionHeadDesignationName, new { @class
= "control-label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
model.SectionHeadDesignationName)
                @Html.ValidationMessageFor(model =>
model.SectionHeadDesignationName)
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
model.SectionHeadDesignationDescription, new {
@class = "control-label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
model.SectionHeadDesignationDescription)
                @Html.ValidationMessageFor(model =>
model.SectionHeadDesignationDescription)
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save"
class="btn btn-primary" />
            @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })

```

```

        </div>
    </div>
</div>
}

        </div>
    </div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(SectionHeadDesignations)
Index.cshtml

@model
IEnumerable<PFMO.Entities.SectionHeadDesignatio
n>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Section Head Designations</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Designations</a>
            </li>
            <li class="active">
                <strong>List</strong>
            </li>
        </ol>
    </div>
</div>

```

```

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>List of Section Head
Designations</h5>
            <div class="ibox-tools">
                @Html.ActionLink("Change View",
"Table", "SectionHeadDesignations", new { @class =
"btn btn-primary btn-xs" })
                @Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-
xs" })
            </div>
        </div>
    </div>
    @foreach (var item in Model)
    {
        <div class="col-lg-3">
            <div class="contact-box center-version">
                <a href="#">
                    <img alt="image" class="img-circle"
src "~/Content/img/HeadDesignation.png">
                    <h3 class="m-b
xs"><strong>@Html.DisplayFor(modelItem =>
item.SectionHeadDesignationName)</strong></h3>
                    <div class="font-
bold">@Html.DisplayFor(modelItem =>
item.SectionHeadDesignationDescription)</div>
                </a>
            <div class="contact-box-footer">
                <div class="m-t-xs btn-group">
                    @Html.ActionLink("Details",
"Details", new { id = item.SectionHeadDesignationID
}, new { @class = "btn btn-xs btn-white" })
                    @Html.ActionLink("Edit", "Edit",
new { id = item.SectionHeadDesignationID }, new {
@class = "btn btn-xs btn-white" })
                    @Html.ActionLink("Delete",
"Delete", new { id = item.SectionHeadDesignationID
}, new { @class = "btn btn-xs btn-white" })
                </div>
            </div>
        </div>
    }
</div>

```

```

PFMO.Web
Views(SectionHeadDesignations)
Tables.cshtml

@model
IEnumerable<PFMO.Entities.SectionHeadDesignatio
n>

{@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Section Head Designations</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Designations</a>
            </li>
            <li class="active">
                <strong>List</strong>
            </li>
        </ol>
    </div>
    </div>

    <div class="wrapper wrapper-content animated
fadeInRight">
        <div class="row">
            <div class="col-lg-12">
                <div class="ibox float-e-margins">
                    <div class="ibox-title">
                        <h5>List of Section Head
Designations</h5>
                    <div class="ibox-tools">
                        @Html.ActionLink("Change View",
"Index", "SectionHeadDesignations", new { @class =
"btn btn-primary btn-xs" })
                        @Html.ActionLink("Create New",
>Create", null, new { @class = "btn btn-primary btn-
xs" })
                    </div>
                </div>
                <div class="ibox-content">

```

```

<input type="text" class="form-control
input-sm m-b-xs" id="filter" placeholder="Search
Designation">
<table class="footable table table-hover"
data-filter=#filter data-page-size="5">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model
=> model.SectionHeadDesignationName)
            </th>
            <th>
                @Html.DisplayNameFor(model
=> model.SectionHeadDesignationDescription)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem
=> item.SectionHeadDesignationName)
                </td>
                <td>
                    @Html.DisplayFor(modelItem
=> item.SectionHeadDesignationDescription)
                </td>
                <td>
                    @Html.ActionLink("Details",
"Details", new { id = item.SectionHeadDesignationID
}, new { @class = "btn btn-primary btn-sm" })
                    @Html.ActionLink("Edit",
>Edit", new { id = item.SectionHeadDesignationID },
new { @class = "btn btn-white btn-sm" })
                    @Html.ActionLink("Delete",
"Delete", new { id = item.SectionHeadDesignationID
}, new { @class = "btn btn-white btn-sm" })
                </td>
            </tr>
        }
    </tbody>
    <tfoot>
        <tr>
            <td colspan="7">
                <ul class="pagination pull-
right"></ul>
            </td>
        </tr>
    </tfoot>

```

```

        </tfoot>
    </table>
</div>
</div>
</div>
</div>
</div>
</div>

PFMO.Web
Views(SectionHeads)
Create.cshtml

@model PFMO.Entities.SectionHead

@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Heads</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Create Section Head</h5>
                </div>

```

```

<div class="ibox-content">
    @using (Html.BeginForm())
    {
        @Html.AntiForgeryToken()

        <div class="form-horizontal">
            @Html.ValidationSummary(true)

            <div class="form-group">
                <div class="form-group">
                    @Html.LabelFor(model =>
model.SectionHeadLastName, new { @class =
"control-label col-md-2" })
                    <div class="col-md-9">
                        @Html.EditorFor(model =>
model.SectionHeadLastName)
                        @Html.ValidationMessageFor(model =>
model.SectionHeadLastName)
                    </div>
                </div>
            <div class="form-group">
                @Html.LabelFor(model =>
model.SectionHeadFirstName, new { @class =
"control-label col-md-2" })
                <div class="col-md-9">
                    @Html.EditorFor(model =>
model.SectionHeadFirstName)
                    @Html.ValidationMessageFor(model =>
model.SectionHeadFirstName)
                </div>
            </div>
            <div class="form-group">
                @Html.LabelFor(model =>
model.SectionHeadGender, new { @class =
"control-label col-md-2" })
                <div class="col-md-9">
                    @Html.EditorFor(model =>
model.SectionHeadGender)
                    @Html.ValidationMessageFor(model =>
model.SectionHeadGender)
                </div>
            </div>
            <div class="form-group">
                <label class="control-label col-md-2">Designation</label>
                @* @Html.LabelFor(model =>
model.SectionHeadDesignationID,
"SectionHeadDesignationID", new { @class =
"control-label col-md-2" }) *@
                <div class="col-md-9">

```

```

        </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
model.SectionHeadDesignationID)
        </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
model.SectionHeadRemarks, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SectionHeadRemarks)
            @Html.ValidationMessageFor(model =>
model.SectionHeadRemarks)
        </div>
        </div>

        <div class="form-group">
            <label class="control-label col-md-
2">Section</label>
            @*@Html.LabelFor(model =>
model.SectionID, "SectionID", new { @class =
"control-label col-md-2" })*@
        <div class="col-md-9">
            @Html.DropDownList("SectionID",
String.Empty)
            @Html.ValidationMessageFor(model =>
model.SectionID)
        </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
model.SectionHeadHireDate, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SectionHeadHireDate)
            @Html.ValidationMessageFor(model =>
model.SectionHeadHireDate)
        </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
model.SectionHeadContactNumber, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SectionHeadContactNumber)
            @Html.ValidationMessageFor(model =>
model.SectionHeadContactNumber)
        </div>
        </div>
    
```

```

        </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create"
class="btn btn-primary" />
            </div>
        </div>

        @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
    </div>
</div>

        </div>
    }
    </div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(SectionHeads)
Delete.cshtml

@model PFMO.Entities.SectionHead

 @{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Heads</a>
            </li>
        </ol>
    

```

```

<li>
    @Html.ActionLink("List", "Index")
</li>
<li class="active">
    <strong>Delete</strong>
</li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Section Head</h5>
                </div>
                <div class="ibox-content">
                    <h4
class="text-danger">Are you sure you want to
delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.Section.SectionName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.Section.SectionName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionHeadDesignation.SectionHeadDesign
ationName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.SectionHeadDesignation.SectionHeadDesign
ationName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionHeadLastName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.SectionHeadLastName)
                        </dd>
                    </dl>
                </div>
            </div>
        </div>
    </div>
</div>
<dt>
    @Html.DisplayNameFor(model =>
model.SectionHeadFirstName)
</dt>
<dd>
    @Html.DisplayFor(model =>
model.SectionHeadFirstName)
</dd>
<dt>
    @Html.DisplayNameFor(model =>
model.SectionHeadGender)
</dt>
<dd>
    @Html.DisplayFor(model =>
model.SectionHeadGender)
</dd>
<dt>
    @Html.DisplayNameFor(model =>
model.SectionHeadRemarks)
</dt>
<dd>
    @Html.DisplayFor(model =>
model.SectionHeadRemarks)
</dd>
<dt>
    @Html.DisplayNameFor(model =>
model.SectionHeadHireDate)
</dt>
<dd>
    @Html.DisplayFor(model =>
model.SectionHeadHireDate)
</dd>
<dt>
    @Html.DisplayNameFor(model =>
model.SectionHeadContactNumber)
</dt>
<dd>
    @Html.DisplayFor(model =>
model.SectionHeadContactNumber)
</dd>
</dl>
<@using (Html.BeginForm()) {
```

```

@Html.AntiForgeryToken()



<input type="submit" value="Delete"
    class="btn btn-primary" />
    @Html.ActionLink("Back to List", "Index",
    null, new { @class = "btn btn-white" })
</div>
}

</div>
</div>
</div>
</div>
</div>

PFMO.Web
Views(SectionHeads)
Details.cshtml

@model PFMO.Entities.SectionHead

 @{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Section Head Details</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Heads</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Details</strong>
            </li>
        </ol>
    </div>
</div>


```

```

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Section Head Details</h5>
                </div>
                <div class="ibox-content">
                    <dl
                        class="dl-horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
                            model.Section.SectionName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
                            model.Section.SectionName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
                            model.SectionHeadDesignation.SectionHeadDesign
                            ationName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
                            model.SectionHeadDesignation.SectionHeadDesign
                            ationName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
                            model.SectionHeadLastname)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
                            model.SectionHeadLastname)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
                            model.SectionHeadFirstName)
                        </dt>
                        <dd>

```

```

        @Html.DisplayFor(model =>
model.SectionHeadFirstName)
    </dd>

    <dt>
        @Html.DisplayNameFor(model =>
model.SectionHeadGender)
    </dt>

    <dd>
        @Html.DisplayFor(model =>
model.SectionHeadGender)
    </dd>

    <dt>
        @Html.DisplayNameFor(model =>
model.SectionHeadRemarks)
    </dt>

    <dd>
        @Html.DisplayFor(model =>
model.SectionHeadRemarks)
    </dd>

    <dt>
        @Html.DisplayNameFor(model =>
model.SectionHeadHireDate)
    </dt>

    <dd>
        @Html.DisplayFor(model =>
model.SectionHeadHireDate)
    </dd>

    <dt>
        @Html.DisplayNameFor(model =>
model.SectionHeadContactNumber)
    </dt>

    <dd>
        @Html.DisplayFor(model =>
model.SectionHeadContactNumber)
    </dd>

</dl>

        @Html.ActionLink("Edit", "Edit", new { id =
= Model.SectionHeadID }, new { @class = "btn btn-
primary"})
        @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

    </div>
</div>

```

**PFMO.Web**  
**Views(SectionHeads)**  
**Edit.cshtml**

@model PFMO.Entities.SectionHead

```

@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-
heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Heads</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

```

<div class="wrapper wrapper-content animated fadeInRight">

```

<div class="row">
    <div class="col-lg-12">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>Edit Section Head</h5>
            </div>
            <div class="ibox-content">

```

```

@using
(Html.BeginForm())
{
    @Html.AntiForgeryToken()

<div class="form-horizontal">
    @Html.ValidationSummary(true)
    @Html.HiddenFor(model =>
model.SectionHeadID)

    <div class="form-group">
        @Html.LabelFor(model =>
model.SectionHeadLastName, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SectionHeadLastName)
            @Html.ValidationMessageFor(model =>
model.SectionHeadLastName)
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.SectionHeadFirstName, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SectionHeadFirstName)
            @Html.ValidationMessageFor(model =>
model.SectionHeadFirstName)
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.SectionHeadGender, new { @class = "control-
label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SectionHeadGender)
            @Html.ValidationMessageFor(model =>
model.SectionHeadGender)
        </div>
    </div>

    <div class="form-group">
        <label class = "control-label col-md-
2">Designation</label>
        @*@Html.LabelFor(model =>
model.SectionHeadDesignationID,
"SectionHeadDesignationID", new { @class =
"control-label col-md-2" })*@

```

```

<div class="col-md-9">
    @Html.DropDownList("SectionHeadDesignationID",
String.Empty)
    @Html.ValidationMessageFor(model =>
model.SectionHeadDesignationID)
</div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SectionHeadRemarks, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SectionHeadRemarks)
        @Html.ValidationMessageFor(model =>
model.SectionHeadRemarks)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-
2">Section</label>
    @*@Html.LabelFor(model =>
model.SectionID, "SectionID", new { @class =
"control-label col-md-2" })*@
    <div class="col-md-9">
        @Html.DropDownList("SectionID",
String.Empty)
        @Html.ValidationMessageFor(model =>
model.SectionID)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SectionHeadHireDate, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SectionHeadHireDate)
        @Html.ValidationMessageFor(model =>
model.SectionHeadHireDate)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SectionHeadContactNumber, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SectionHeadContactNumber)

```

```

        @Html.ValidationMessageFor(model =>
model.SectionHeadContactNumber)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Save"
class="btn btn-primary" />

        @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
    </div>
</div>
}

</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

PFMO.Web  
Views(SectionHeads)  
Index.cshtml

```

@model IEnumerable<PFMO.Entities.SectionHead>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Section Heads<br></h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Heads</a>
            </li>
        
```

```

</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>List of Section Heads</h5>
                <div class="ibox-tools">
                    @Html.ActionLink("Change View",
"Table", "SectionHeads", new { @class = "btn btn-primary btn-xs" })
                    @Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-xs" })
                </div>
            </div>
            <div class="contact-box center-version">
                <a href="#">
                    <img alt="image" class="img-circle"
src "~/Content/img/Head.jpg">
                    <h3 class="m-b">
                        @Html.DisplayFor(modelItem =>
item.SectionHeadFullName)</h3>
                    <div class="font-bold">
                        @Html.DisplayFor(modelItem =>
item.SectionHeadDesignation.SectionHeadDesignati
onName)</div>
                </a>
                <div class="contact-box-footer">
                    <div class="m-t-xs btn-group">
                        @Html.ActionLink("Details",
"Details", new { id = item.SectionID }, new { @class =
"btn btn-xs btn-white" })
                        @Html.ActionLink("Edit", "Edit",
new { id = item.SectionID }, new { @class = "btn btn-
xs btn-white" })
                    </div>
                
```

```

        @Html.ActionLink("Delete",
"Delete", new { id = item.SectionID }, new { @class =
"btn btn-xs btn-white" })
    </div>
</div>
</div>
</div>
}
</div>
</div>

PFMO.Web
Views(SectionHeads)
Tables.cshtml

@model IEnumerable<PFMO.Entities.SectionHead>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Section Heads<br></h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Heads</a>
            </li>
            <li class="active">
                <strong>List</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Section Heads</h5>
                <div class="ibox-tools">
                    @Html.ActionLink("Delete",
"Delete", new { id = item.SectionID }, new { @class =
"btn btn-xs btn-white" })
                </div>
                <div class="ibox-content">
                    @Html.ActionLink("Change View",
"Index", "SectionHeads", new { @class = "btn btn-primary btn-xs" })
                    @Html.ActionLink("Create New",
"Create", null, new { @class = "btn btn-primary btn-xs" })
                </div>
            </div>
            <div class="ibox-content">
                <input type="text" class="form-control" id="filter" placeholder="Search Section Head">
                <table class="footable table table-hover" data-filter="#filter" data-page-size="5">
                    <thead>
                        <tr>
                            <th>
                                @Html.DisplayNameFor(model => model.Section.SectionName)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.SectionHeadDesignation.SectionHeadDesignationName)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.SectionHeadLastName)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.SectionHeadFirstName)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.SectionHeadGender)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.SectionHeadRemarks)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.SectionHeadHireDate)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model => model.SectionHeadContactNumber)
                            </th>
                        </tr>
                    <tbody>
                        <tr>
                            <td>
                                @Html.DisplayFor(model => model.Section)
                            </td>
                            <td>
                                @Html.DisplayFor(model => model.SectionHeadDesignation)
                            </td>
                            <td>
                                @Html.DisplayFor(model => model.SectionHeadLast)
                            </td>
                            <td>
                                @Html.DisplayFor(model => model.SectionHeadFirst)
                            </td>
                            <td>
                                @Html.DisplayFor(model => model.SectionHeadGender)
                            </td>
                            <td>
                                @Html.DisplayFor(model => model.SectionHeadRemarks)
                            </td>
                            <td>
                                @Html.DisplayFor(model => model.SectionHeadHire)
                            </td>
                            <td>
                                @Html.DisplayFor(model => model.SectionHeadContact)
                            </td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>

```

```

<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem
=> item.Section.SectionName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=>
item.SectionHeadDesignation.SectionHeadDesignati
onName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SectionHeadLastName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SectionHeadFirstName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SectionHeadGender)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SectionHeadRemarks)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SectionHeadHireDate)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SectionHeadContactNumber)
            </td>
            <td>
                @Html.ActionLink("Details",
"Details", new { id = item.SectionHeadID }, new {
@class = "btn btn-primary btn-sm" })
                @Html.ActionLink("Edit",
>Edit", new { id = item.SectionHeadID }, new {
@class = "btn btn-white btn-sm" })
                @Html.ActionLink("Delete",
"Delete", new { id = item.SectionHeadID }, new {
@class = "btn btn-white btn-sm" })
            </td>
        </tr>
    }
</tbody>

<tfoot>
    <tr>
        <td colspan="7" style="text-align: right"></ul>
    </td>
    </tr>
</tfoot>

```

**PFMO.Web**  
**Views(Sections)**  
**Create.cshtml**

@model PFMO.Entities.Section

```

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Sections</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
        <li class="active">
            <strong>Create</strong>
        </li>
        </ol>
    </div>
</div>

```

```

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Create Section</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()
                    }
                    <div class="form-horizontal">
                        @Html.ValidationSummary(true)

                        <div class="form-group">
                            <div class="form-group">
                                @Html.LabelFor(model =>
model.SectionName, new { @class = "control-label
col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.SectionName)
                                    @Html.ValidationMessageFor(model =>
model.SectionName)
                                </div>
                            </div>
                        </div>

                        <div class="form-group">
                            @Html.LabelFor(model =>
model.SectionType, new { @class = "control-label
col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.SectionType)
                                @Html.ValidationMessageFor(model =>
model.SectionType)
                            </div>
                        </div>

                        <div class="form-group">
                            @Html.LabelFor(model =>
model.SectionDescription, new { @class = "control-
label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.SectionDescription)
                                @Html.ValidationMessageFor(model =>
model.SectionDescription)
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model =>
model.SectionDateCreated, new { @class = "control-
label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
model.SectionDateCreated)
                @Html.ValidationMessageFor(model =>
model.SectionDateCreated)
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create"
class="btn btn-primary" />
                @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
            </div>
        </div>
    </div>

```

@section Scripts {
 @Scripts.Render("~/bundles/jqueryval")
}

**PFMO.Web**  
**Views(Sections)**  
**Delete.cshtml**

---

@model PFMO.Entities.Section

@{
 ViewBag.Title = "Delete";
 Layout = "~/Views/Shared/\_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">

## PFMO.Web Views(Sections) Delete.cshtml

@model PFMO.Entities.Section

```
@{  
    ViewBag.Title = "Delete";  
    Layout = "~/Views/Shared/_Layout.cshtml";  
}  
  
<div class="row wrapper border-bottom white-bg page-heading">  
    <div class="col-sm-4">
```

```

<h2>Delete</h2>
<ol class="breadcrumb">
    <li>
        <a href="#">Home</a>
    </li>
    <li>
        <a href="#">Panel</a>
    </li>
    <li>
        <a href="#">Sections</a>
    </li>
    <li>
        @Html.ActionLink("List", "Index")
    </li>
    <li class="active">
        <strong>Delete</strong>
    </li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Section</h5>
                </div>
                <div class="ibox-content">
                    <h4
                        class="text-danger">Are you sure you want to
                        delete this?</h4>
                    <dl class="dl-
                        horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
                                model.SectionName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
                                model.SectionName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
                                model.SectionType)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
                                model.SectionType)
                        </dd>
                </div>
            </div>
        </div>
    </div>
</div>

<dt>
    @Html.DisplayNameFor(model =>
        model.SectionDescription)
</dt>
<dd>
    @Html.DisplayFor(model =>
        model.SectionDescription)
</dd>
<dt>
    @Html.DisplayNameFor(model =>
        model.SectionDateCreated)
</dt>
<dd>
    @Html.DisplayFor(model =>
        model.SectionDateCreated)
</dd>
</dl>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Delete"
            class="btn btn-primary" />
        @Html.ActionLink("Back to List", "Index",
            null, new { @class = "btn btn-white" })
    </div>
}

</div>
</div>
</div>
</div>

PFMO.Web
Views(Sections)
Details.cshtml

@model PFMO.Entities.Section

 @{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Details</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Sections</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Details</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Section Details</h5>
                </div>
                <div class="ibox-content">
                    <dl
                        class="dl-horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionName)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.SectionName)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionType)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.SectionType)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionDescription)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.SectionDescription)
                        </dd>
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SectionDateCreated)
                        </dt>
                        <dd>
                            @Html.DisplayFor(model =>
model.SectionDateCreated)
                        </dd>
                    </dl>
                    @Html.ActionLink("Edit", "Edit", new { id =
Model.SectionID }, new { @class = "btn btn-primary"})
                    @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})
                </div>
            </div>
        </div>
    </div>
</div>

```

**PFMO.Web**  
**Views(Sections)**  
**Edit.cshtml**

@model PFMO.Entities.Section

@{  
 ViewBag.Title = "Edit";  
 Layout = "~/Views/Shared/\_Layout.cshtml";  
}

<div class="row wrapper border-bottom white-bg page-heading">

```

<div class="col-sm-4">
    <h2>Edit</h2>
    <ol class="breadcrumb">
        <li>
            <a href="#">Home</a>
        </li>
        <li>
            <a href="#">Panel</a>
        </li>
        <li>
            <a href="#">Sections</a>
        </li>
        <li>
            @Html.ActionLink("List", "Index")
        </li>
        <li class="active">
            <strong>Edit</strong>
        </li>
    </ol>
</div>

</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Section</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()

                        <div class="form-horizontal">
                            @Html.ValidationSummary(true)
                            @Html.HiddenFor(model => model.SectionID)

                            <div class="form-group">
                                @Html.LabelFor(model =>
                                model.SectionName, new { @class = "control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
                                    model.SectionName)
                                    @Html.ValidationMessageFor(model =>
                                    model.SectionName)
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model =>
            model.SectionType, new { @class = "control-label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
                model.SectionType)
                @Html.ValidationMessageFor(model =>
                model.SectionType)
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
            model.SectionDescription, new { @class = "control-label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
                model.SectionDescription)
                @Html.ValidationMessageFor(model =>
                model.SectionDescription)
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model =>
            model.SectionDateCreated, new { @class = "control-label col-md-2" })
            <div class="col-md-9">
                @Html.EditorFor(model =>
                model.SectionDateCreated)
                @Html.ValidationMessageFor(model =>
                model.SectionDateCreated)
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save"
                class="btn btn-primary" />
            </div>
        </div>

        <div class="form-group">
            @Html.ActionLink("Cancel", "Index", null,
            new { @class = "btn btn-white" })
            </div>
        </div>
    </div>
</div>
</div>

```



```

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Sections</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Sections</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List", "Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Sections</h5>
                    <div class="ibox-tools">
                        @Html.ActionLink("Change View", "Index", "Sections", new { @class = "btn btn-primary btn-xs" })
                        @Html.ActionLink("Create New", "Create", null, new { @class = "btn btn-primary btn-xs" })
                    </div>
                </div>
                <div class="ibox-content">
                    <input type="text" class="form-control input-sm m-b-xs" id="filter" placeholder="Search Section">
                    <table class="footable table table-hover" data-filter="#filter data-page-size="5">
                        <thead>
                            <tr>
                                <th>
                                    @Html.DisplayNameFor(model => model.SectionName)
                                </th>
                                <th>

```

@Html.DisplayNameFor(model => model.SectionType)

```

                                    </th>
                                <th>
                                    @Html.DisplayNameFor(model => model.SectionDescription)
                                </th>
                                <th>
                                    @Html.DisplayNameFor(model => model.SectionDateCreated)
                                </th>
                            </tr>
                        </thead>
                        <tbody>
                            @foreach (var item in Model)
                            {
                                <tr>
                                    <td>
                                        @Html.DisplayFor(modelItem => item.SectionName)
                                    </td>
                                    <td>
                                        @Html.DisplayFor(modelItem => item.SectionType)
                                    </td>
                                    <td>
                                        @Html.DisplayFor(modelItem => item.SectionDescription)
                                    </td>
                                    <td>
                                        @Html.DisplayFor(modelItem => item.SectionDateCreated)
                                    </td>
                                </tr>
                            }
                        </tbody>
                    <tfoot>
                        <tr>
                            <td colspan="7">
                                <ul class="pagination pull-right"></ul>

```



```

</div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SectionTaskStartDate, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SectionTaskStartDate)
        @Html.ValidationMessageFor(model =>
model.SectionTaskStartDate)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SectionTaskEndDate, new { @class =
"control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SectionTaskEndDate)
        @Html.ValidationMessageFor(model =>
model.SectionTaskEndDate)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Create"
class="btn btn-primary" />
    @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>
</div>

</div>

}
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(SectionTasks)**  
**Delete.cshtml**

```

@model PFMO.Entities.SectionTask

{@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Task</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Section Task</h5>
                </div>
                <div class="ibox-content">
                    <h4
class="text-danger">Are you sure you want to
delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.Section.SectionName)
                        </dt>

```

```

<dd>
    @Html.DisplayFor(model =>
model.Section.SectionName)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionTaskTitle)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionTaskTitle)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionTaskDescription)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionTaskDescription)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionTaskStartDate)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionTaskStartDate)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionTaskEndDate)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionTaskEndDate)
</dd>

<dt>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()

        <div class="form-actions no-color">
            <input type="submit" value="Delete"
class="btn btn-primary" />
            @Html.ActionLink("Back to List", "Index",
null, new { @class = "btn btn-white" })
    
```

PFMO.Web  
Views(SectionTasks)  
Details.cshtml

```

        </div>
    }
}

</div>
</div>
</div>
</div>
</div>

@model PFMO.Entities.SectionTask
@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Details</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Task</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Details</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
```

```

<h5>Section Task Details</h5>
</div>
<div class="ibox-content">

<dl
class="dl-horizontal">
<dt>
    @Html.DisplayNameFor(model =>
model.Section.SectionName)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.Section.SectionName)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionTaskTitle)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionTaskTitle)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionTaskDescription)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionTaskDescription)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionTaskStartDate)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionTaskStartDate)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SectionTaskEndDate)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SectionTaskEndDate)
</dd>

```

```

</dd>
</dl>

@Html.ActionLink("Edit", "Edit", new { id =
= Model.SectionTaskID }, new { @class = "btn btn-
primary"})
@Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

</div>
</div>
</div>
</div>

PFMO.Web
Views(SectionTasks)
Edit.cshtml

@model PFMO.Entities.SectionTask

 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
<div class="col-sm-4">
    <h2>Edit</h2>
    <ol class="breadcrumb">
        <li>
            <a href="#">Home</a>
        </li>
        <li>
            <a href="#">Panel</a>
        </li>
        <li>
            <a href="#">Section Task</a>
        </li>
        <li>
            @Html.ActionLink("List", "Index")
        </li>
        <li class="active">
            <strong>Edit</strong>
        </li>
    </ol>
</div>
</div>

```

```

</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Section Task</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()

                        <div class="form-horizontal">
                            @Html.ValidationSummary(true)
                            @Html.HiddenFor(model =>
model.SectionTaskID)

                            <div class="form-group">
                                @Html.LabelFor(model =>
model.SectionTaskTitle, new { @class = "control-
label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.SectionTaskTitle)
                                    @Html.ValidationMessageFor(model =>
model.SectionTaskTitle)
                                </div>
                            </div>

                            <div class="form-group">
                                @Html.LabelFor(model =>
model.SectionTaskDescription, new { @class =
"control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.SectionTaskDescription)
                                    @Html.ValidationMessageFor(model =>
model.SectionTaskDescription)
                                </div>
                            </div>

                            <div class="form-group">
                                <label class="control-label col-md-
2">Section</label>
                                <div class="col-md-9">
                                    @Html.DropDownList("SectionID",
String.Empty)
                                    @Html.ValidationMessageFor(model =>
model.SectionID)
                                </div>
                            </div>
                        </div>
                    
```

```

                        </div>
                        <div class="form-group">
                            @Html.LabelFor(model =>
model.SectionTaskStartDate, new { @class =
"control-label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.SectionTaskStartDate)
                                @Html.ValidationMessageFor(model =>
model.SectionTaskStartDate)
                            </div>
                        </div>

                        <div class="form-group">
                            @Html.LabelFor(model =>
model.SectionTaskEndDate, new { @class =
"control-label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.SectionTaskEndDate)
                                @Html.ValidationMessageFor(model =>
model.SectionTaskEndDate)
                            </div>
                        </div>

                        <div class="form-group">
                            <div class="col-md-offset-2 col-md-10">
                                <input type="submit" value="Save"
class="btn btn-primary" />
                            </div>
                        </div>
                    
```

```

                    @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
                </div>
            </div>
        }
    
```

```

            </div>
            </div>
            </div>
            </div>
        
```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

**PFMO.Web**  
**Views(SectionTasks)**  
**Index.cshtml**

```
@model IEnumerable<PFMO.Entities.SectionTask>

{@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Index</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Section Task</a>
            </li>
            <li class="active">
                <strong>List</strong>
            </li>
        </ol>
    </div>
</div>
```

```
<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>List of Section Tasks</h5>
                <div class="ibox-tools">
                    @Html.ActionLink("Change View", "Table", "SectionTasks", new { @class = "btn btn-primary btn-xs" })
                    @Html.ActionLink("Create New", "Create", null, new { @class = "btn btn-primary btn-xs" })
                </div>
            </div>
            <div class="col-lg-3">
                <div class="contact-box center-version">
                    <a href="#">
```

```


<h3 class="m-b"><strong>@Html.DisplayFor(modelItem => item.SectionTaskTitle)</strong></h3>

<div class="font-bold">@Html.DisplayFor(modelItem => item.SectionTaskDescription)</div>

</a>

<div class="contact-box-footer">
    <div class="m-t-xs btn-group">
        @Html.ActionLink("Details", "Details", new { id = item.SectionTaskID }, new { @class = "btn btn-xs btn-white" })
        @Html.ActionLink("Edit", "Edit", "Delete", new { id = item.SectionTaskID }, new { @class = "btn btn-xs btn-white" })
        @Html.ActionLink("Delete", "Delete", new { id = item.SectionTaskID }, new { @class = "btn btn-xs btn-white" })
    </div>
</div>
</div>
</div>
</div>
</div>
```

**PFMO.Web**  
**Views(SectionTasks)**  
**Tables.cshtml**

```
@model IEnumerable<PFMO.Entities.SectionTask>

{@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Index</h2>
        <ol class="breadcrumb">
            <li>
```

```

<a href="#">Home</a>
</li>
<li>
    <a href="#">Panel</a>
</li>
<li>
    <a href="#">Section Task</a>
</li>
<li class="active">
    <strong>List</strong>
</li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Section Tasks</h5>
                <div class="ibox-tools">
                    @Html.ActionLink("Change View",
                    "Index", "SectionTasks", new { @class = "btn btn-primary btn-xs" })
                    @Html.ActionLink("Create New",
                    "Create", null, new { @class = "btn btn-primary btn-xs" })
                </div>
            </div>
            <div class="ibox-content">

                <input type="text" class="form-control"
                input-sm m-b-xs" id="filter" placeholder="Search
                Personnel Task">
                <table class="footable table table-hover"
                data-filter="#filter data-page-size="5">

                    <thead>
                        <tr>
                            <th>
                                @Html.DisplayNameFor(model
                                => model.Section.SectionName)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                => model.SectionTaskTitle)
                            </th>
                            <th>
                                @Html.DisplayNameFor(model
                                => model.SectionTaskDescription)
                            </th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>
                                @Html.DisplayFor(modelItem
                                => item.SectionTaskStart Date)
                            </td>
                            <td>
                                @Html.DisplayFor(modelItem
                                => item.SectionTaskEnd Date)
                            </td>
                            <td>
                                @Html.ActionLink("Details",
                                "Details", new { id = item.SectionTaskID }, new {
                                @class = "btn btn-primary btn-sm" })
                                @Html.ActionLink("Edit",
                                "Edit", new { id = item.SectionTaskID }, new { @class
                                = "btn btn-white btn-sm" })
                                @Html.ActionLink("Delete",
                                "Delete", new { id = item.SectionTaskID }, new {
                                @class = "btn btn-white btn-sm" })
                            </td>
                        </tr>
                    </tbody>
                    <tfoot>
                        <tr>
                            <td colspan="3">

```

```

                    <ul class="pagination pull-right"></ul>
                </td>
            </tr>
        </tfoot>

    </table>
</div>
</div>
</div>
</div>
</div>
</div>

PFMO.Web
Views(Supplies)
Create.cshtml

@model PFMO.Entities.Supply

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Supplies</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">

```

```

<div class="ibox float-e-margins">
    <div class="ibox-title">
        <h5>Create Supply</h5>
    </div>
    <div class="ibox-content">

        @using (Html.BeginForm())
        {
            @Html.AntiForgeryToken()

            <div class="form-horizontal">
                @Html.ValidationSummary(true)

                <div class="form-group">
                    <div class="form-group">
                        @Html.LabelFor(model =>
model.SupplyCode, new { @class = "control-label col-md-2" })
                    <div class="col-md-9">
                        @Html.EditorFor(model =>
model.SupplyCode)

                        @Html.ValidationMessageFor(model =>
model.SupplyCode)
                    </div>
                </div>

                <div class="form-group">
                    @Html.LabelFor(model =>
model.SupplyName, new { @class = "control-label col-md-2" })
                    <div class="col-md-9">
                        @Html.EditorFor(model =>
model.SupplyName)

                        @Html.ValidationMessageFor(model =>
model.SupplyName)
                    </div>
                </div>

                <div class="form-group">
                    <label class="control-label col-md-2">Category</label>
                    <div class="col-md-9">

```

```

@Html.DropDownList("SupplyCategoryID",
String.Empty)

@Html.ValidationMessageFor(model =>
model.SupplyCategoryID)
                </div>
            </div>

            <div class="form-group">

```

```

        @Html.LabelFor(model =>
model.SupplyRemarks, new { @class = "control-
label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyRemarks)

    @Html.ValidationMessageFor(model =>
model.SupplyRemarks)
    </div>
</div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.SupplyStatus, new { @class = "control-label
col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyStatus)

    @Html.ValidationMessageFor(model =>
model.SupplyStatus)
    </div>
</div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.SupplyBalance, new { @class = "control-label
col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyBalance)

    @Html.ValidationMessageFor(model =>
model.SupplyBalance)
    </div>
</div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.SupplyUnit, new { @class = "control-label col-
md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyUnit)

    @Html.ValidationMessageFor(model =>
model.SupplyUnit)
    </div>
</div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.SupplyReorderPoint, new { @class = "control-
label col-md-2" })

```

```

<div class="col-md-9">
    @Html.EditorFor(model =>
model.SupplyReorderPoint)

    @Html.ValidationMessageFor(model =>
model.SupplyReorderPoint)
</div>
</div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.Target, new { @class = "control-label col-md-
2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.Target)

    @Html.ValidationMessageFor(model =>
model.Target)
    </div>
</div>

    <div class="form-group">
        <label class="control-label col-
md-2">Assigned to Project</label>
    <div class="col-md-9">
        @Html.DropDownList("ProjectID", String.Empty)

    @Html.ValidationMessageFor(model =>
model.ProjectID)
    </div>
</div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.SupplyExpiryDate, new { @class = "control-
label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyExpiryDate)

    @Html.ValidationMessageFor(model =>
model.SupplyExpiryDate)
    </div>
</div>

    <div class="form-group">
        <div class="col-md-offset-2 col-
md-10">
            <input type="submit"
value="Create" class="btn btn-primary" />
            @Html.ActionLink("Cancel",
"Index", null, new { @class = "btn btn-white" })
        </div>

```



```

</dt>
<dd>
    @Html.DisplayFor(model =>
model.SupplyRemarks)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SupplyStatus)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SupplyStatus)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SupplyBalance)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SupplyBalance)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SupplyUnit)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SupplyUnit)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SupplyReorderPoint)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SupplyReorderPoint)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.Target)
</dt>

<dd>
    @Html.DisplayFor(model => model.Target)
</dd>

```

```

<dt>
    @Html.DisplayNameFor(model =>
model.SupplyExpiryDate)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SupplyExpiryDate)
</dd>

</dl>

@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Delete" class="btn btn-primary" />
        @Html.ActionLink("Back to List", "Index", null, new { @class = "btn btn-white" })
    </div>
}

</div>
</div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(Supplies)**  
**Details.cshtml**

```

@model PFMO.Entities.Supply

 @{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Details</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
        </ol>
    </div>

```

```

</li>
<li>
    <a href="#">Supplies</a>
</li>
<li>
    @Html.ActionLink("List", "Index")
</li>
<li class="active">
    <strong>Details</strong>
</li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
<div class="row">
    <div class="col-lg-12">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>Supply Details</h5>
            </div>
            <div class="ibox-content">

                <dl
                    class="dl-horizontal">
                    <dt>
                        @Html.DisplayNameFor(model =>
model.Project.ProjectName)
                    </dt>

                    <dd>
                        @Html.DisplayFor(model =>
model.Project.ProjectName)
                    </dd>

                    <dt>
                        @Html.DisplayNameFor(model =>
model.SupplyCategory.SupplyCategoryName)
                    </dt>

                    <dd>
                        @Html.DisplayFor(model =>
model.SupplyCategory.SupplyCategoryName)
                    </dd>

                    <dt>
                        @Html.DisplayNameFor(model =>
model.SupplyCode)
                    </dt>

                    <dd>
                        @Html.DisplayFor(model =>
model.SupplyCode)
                    </dd>

                    <dt>
                        @Html.DisplayNameFor(model =>
model.SupplyName)
                    </dt>

                    <dd>
                        @Html.DisplayFor(model =>
model.SupplyName)
                    </dd>

                    <dt>
                        @Html.DisplayNameFor(model =>
model.SupplyRemarks)
                    </dt>

                    <dd>
                        @Html.DisplayFor(model =>
model.SupplyRemarks)
                    </dd>

                    <dt>
                        @Html.DisplayNameFor(model =>
model.SupplyStatus)
                    </dt>

                    <dd>
                        @Html.DisplayFor(model =>
model.SupplyStatus)
                    </dd>

                    <dt>
                        @Html.DisplayNameFor(model =>
model.SupplyBalance)
                    </dt>

                    <dd>
                        @Html.DisplayFor(model =>
model.SupplyBalance)
                    </dd>

                    <dt>
                        @Html.DisplayNameFor(model =>
model.SupplyUnit)
                    </dt>

                    <dd>
                        @Html.DisplayFor(model =>
model.SupplyUnit)
                    </dd>
                </dl>
            </div>
        </div>
    </div>
</div>

```

```

        <dt>
            @Html.DisplayNameFor(model =>
model.SupplyReorderPoint)
        </dt>

        <dd>
            @Html.DisplayFor(model =>
model.SupplyReorderPoint)
        </dd>

        <dt>
            @Html.DisplayNameFor(model =>
model.Target)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Target)
        </dd>

        <dt>
            @Html.DisplayNameFor(model =>
model.SupplyExpiryDate)
        </dt>

        <dd>
            @Html.DisplayFor(model =>
model.SupplyExpiryDate)
        </dd>

    </dl>

    @Html.ActionLink("Edit", "Edit", new { id =
= Model.SupplyID }, new { @class = "btn btn-
primary"})
    @Html.ActionLink("Back to List", "Index", null,
new { @class = "btn btn-white"})

        </div>
    </div>
</div>
</div>
</div>

```

**PFMO.Web**  
**Views(Supplies)**  
**Edit.cshtml**

```

@model PFMO.Entities.Supply

 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```

        <div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Supplies</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Supply</h5>
                </div>
                <div class="ibox-content">

```

```

@using
(Html.BeginForm())
{
    @Html.AntiForgeryToken()
<div class="form-horizontal">

    @Html.ValidationSummary(true)
    @Html.HiddenFor(model => model.SupplyID)

    <div class="form-group">
        @Html.LabelFor(model =>
model.SupplyCode, new { @class = "control-label
col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SupplyCode)

```

```

        @Html.ValidationMessageFor(model =>
model.SupplyCode)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SupplyName, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyName)
        @Html.ValidationMessageFor(model =>
model.SupplyName)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Category</label>
    <div class="col-md-9">
        @Html.DropDownList("SupplyCategoryID",
String.Empty)
        @Html.ValidationMessageFor(model =>
model.SupplyCategoryID)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SupplyRemarks, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyRemarks)
        @Html.ValidationMessageFor(model =>
model.SupplyRemarks)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SupplyStatus, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyStatus)
        @Html.ValidationMessageFor(model =>
model.SupplyStatus)
    </div>
</div>

<div class="form-group">

```

```

        @Html.LabelFor(model =>
model.SupplyBalance, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyBalance)
        @Html.ValidationMessageFor(model =>
model.SupplyBalance)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.SupplyUnit,
new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyUnit)
        @Html.ValidationMessageFor(model =>
model.SupplyUnit)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model =>
model.SupplyReorderPoint, new { @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyReorderPoint)
        @Html.ValidationMessageFor(model =>
model.SupplyReorderPoint)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Target, new
{ @class = "control-label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model => model.Target)
        @Html.ValidationMessageFor(model =>
model.Target)
    </div>
</div>

<div class="form-group">
    <label class="control-label col-md-2">Assigned to Project</label>
    <div class="col-md-9">
        @Html.DropDownList("ProjectID",
String.Empty)
        @Html.ValidationMessageFor(model =>
model.ProjectID)
    </div>
</div>

```

```

<div class="form-group">
    @Html.LabelFor(model =>
model.SupplyExpiryDate, new { @class = "control-
label col-md-2" })
    <div class="col-md-9">
        @Html.EditorFor(model =>
model.SupplyExpiryDate)
        @Html.ValidationMessageFor(model =>
model.SupplyExpiryDate)
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Save"
class="btn btn-primary" />

        @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
    </div>
</div>
</div>
}

</div>
</div>
</div>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(Supplies)
Index.cshtml

@model IEnumerable<PFMO.Entities.Supply>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Supplies</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Home</a>
            </li>
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Supplies</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List",
"Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Supply</h5>
                </div>
                <div class="ibox-tools">
                    @Html.ActionLink("Create New", "Create",
null, new { @class = "btn btn-primary btn-xs" })
                </div>
                </div>
                <div class="ibox-content">
                    <input type="text" class="form-control
input-sm m-b-xs" id="filter" placeholder="Search
Equipment">
                    <table class="footable table table-hover"
data-filter=#filter data-page-size="5">
                        <thead>
                            <tr>
                                <th>
                                    @Html.DisplayNameFor(model
=> model.Project.ProjectName)
                                </th>
                                <th>
                                    @Html.DisplayNameFor(model
=> model.SupplyCategory.SupplyCategoryName)
                                </th>
                                <th>
                                    @Html.DisplayNameFor(model
=> model.SupplyCode)
                                </th>
                            </tr>
                        <tbody>
                            <tr>
                                <td>
                                    <strong>@Html.ActionLink("Edit",
"Edit", new { id = item.Id })</strong>
                                </td>
                                <td>
                                    <strong>@Html.ActionLink("Details",
"Details", new { id = item.Id })</strong>
                                </td>
                                <td>
                                    <strong>@Html.ActionLink("Delete",
"Delete", new { id = item.Id })</strong>
                                </td>
                            </tr>
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        @Html.DisplayNameFor(model
=> model.SupplyName)
        </th>
        <th>
            @Html.DisplayNameFor(model
=> model.SupplyRemarks)
        </th>
        <th>
            @Html.DisplayNameFor(model
=> model.SupplyStatus)
        </th>
        <th>
            @Html.DisplayNameFor(model
=> model.SupplyBalance)
        </th>
        <th>
            @Html.DisplayNameFor(model
=> model.SupplyUnit)
        </th>
        <th>
            @Html.DisplayNameFor(model
=> model.SupplyReorderPoint)
        </th>
        <th>
            @Html.DisplayNameFor(model
=> model.Target)
        </th>
        <th>
            @Html.DisplayNameFor(model
=> model.SupplyExpiryDate)
        </th>
        <th></th>
    </tr>
</thead>

<tbody>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem
=> item.Project.ProjectName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyCategory.SupplyCategoryName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyCode)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyName)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyRemarks)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyStatus)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyBalance)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyUnit)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyReorderPoint)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.Target)
            </td>
            <td>
                @Html.DisplayFor(modelItem
=> item.SupplyExpiryDate)
            </td>
            <td>
                @Html.ActionLink("Details",
"Details", new { id = item.SupplyID }, new { @class =
"btn btn-primary btn-sm" })
            </td>
            <td>
                @Html.ActionLink("Edit",
>Edit", new { id = item.SupplyID }, new { @class =
"btn btn-white btn-sm" })
            </td>
            <td>
                @Html.ActionLink("Delete",
"Delete", new { id = item.SupplyID }, new { @class =
"btn btn-white btn-sm" })
            </td>
        </tr>
    }
</tbody>

<tfoot>
    <tr>
        <td colspan="7">
            <ul class="pagination pull-
right"></ul>
        </td>
    </tr>
</tfoot>

</table>
</div>

```

```

        </div>
    </div>
</div>
</div>

PFMO.Web
Views(SupplyCategories)
Create.cshtml

@model PFMO.Entities.SupplyCategory

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Create</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Create</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Create Supply Category</h5>
                </div>
                <div class="ibox-content">
                    @using
                    (Html.BeginForm())
                    {
                        @Html.AntiForgeryToken()
                
```

```

                    <div class="form-horizontal">
                        @Html.ValidationSummary(true)

                        <div class="form-group">
                            <div class="form-group">
                                @Html.LabelFor(model =>
model.SupplyCategoryName, new { @class =
"control-label col-md-2" })
                                <div class="col-md-9">
                                    @Html.EditorFor(model =>
model.SupplyCategoryName)
                                    @Html.ValidationMessageFor(model =>
model.SupplyCategoryName)
                                </div>
                            </div>
                        <div class="form-group">
                            @Html.LabelFor(model =>
model.SupplyCategoryDescription, new { @class =
"control-label col-md-2" })
                            <div class="col-md-9">
                                @Html.EditorFor(model =>
model.SupplyCategoryDescription)
                                @Html.ValidationMessageFor(model =>
model.SupplyCategoryDescription)
                            </div>
                        </div>
                        <div class="form-group">
                            <div class="col-md-offset-2 col-md-10">
                                <input type="submit" value="Create"
class="btn btn-primary" />
                            @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
                            </div>
                        </div>
                    </div>
                
```

```

                </div>
            </div>
        </div>
    </div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

## PFMO.Web Views(SupplyCategories) Delete.cshtml

```
@model PFMO.Entities.SupplyCategory

{@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Delete</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Delete</strong>
            </li>
        </ol>
    </div>
</div>
```

```
<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Delete Supply Category</h5>
                </div>
                <div class="ibox-content">
                    <h4
class="text-danger">Are you sure you want to
delete this?</h4>
                    <dl class="dl-
horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SupplyCategoryName)
                        </dt>
                        <dd>
```

```
    @Html.DisplayFor(model =>
model.SupplyCategoryName)
</dd>

<dt>
    @Html.DisplayNameFor(model =>
model.SupplyCategoryDescription)
</dt>

<dd>
    @Html.DisplayFor(model =>
model.SupplyCategoryDescription)
</dd>

</dl>

@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Delete" class="btn btn-primary" />
        @Html.ActionLink("Back to List", "Index",
null, new { @class = "btn btn-white" })
    </div>
}

</div>
</div>
</div>
</div>
</div>
```

**PFMO.Web**  
**Views(SupplyCategories)**  
**Details.cshtml**

@model PFMO.Entities.SupplyCategory

```
@{  
    ViewBag.Title = "Details";  
    Layout = "~/Views/Shared/_Layout.cshtml";  
}
```

## Details

- [Panel](#)

```

</li>
</li>
    <a href="#">Categories</a>
</li>
<li>
    @Html.ActionLink("List", "Index")
</li>
<li class="active">
    <strong>Details</strong>
</li>
</ol>
</div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Supply Category Details</h5>
                </div>
                <div class="ibox-content">

                    <dl class="dl-horizontal">
                        <dt>
                            @Html.DisplayNameFor(model =>
model.SupplyCategoryName)
                        </dt>

                        <dd>
                            @Html.DisplayFor(model =>
model.SupplyCategoryName)
                        </dd>

                        <dt>
                            @Html.DisplayNameFor(model =>
model.SupplyCategoryDescription)
                        </dt>

                        <dd>
                            @Html.DisplayFor(model =>
model.SupplyCategoryDescription)
                        </dd>

                    </dl>
                </div>
            </div>
        </div>
    </div>
</div>

@model PFMO.Entities.SupplyCategory
 @{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Edit</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li>
                @Html.ActionLink("List", "Index")
            </li>
            <li class="active">
                <strong>Edit</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>Edit Supply Category</h5>
                </div>
                <div class="ibox-content">

```

```

@using
(Html.BeginForm())
{
    @Html.AntiForgeryToken()

<div class="form-horizontal">

    @Html.ValidationSummary(true)
    @Html.HiddenFor(model =>
model.SupplyCategoryId)

    <div class="form-group">
        @Html.LabelFor(model =>
model.SupplyCategoryName, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SupplyCategoryName)
            @Html.ValidationMessageFor(model =>
model.SupplyCategoryName)
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model =>
model.SupplyCategoryDescription, new { @class =
"control-label col-md-2" })
        <div class="col-md-9">
            @Html.EditorFor(model =>
model.SupplyCategoryDescription)
            @Html.ValidationMessageFor(model =>
model.SupplyCategoryDescription)
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Save"
class="btn btn-primary" />

            @Html.ActionLink("Cancel", "Index", null,
new { @class = "btn btn-white" })
        </div>
    </div>
}
</div>
</div>
</div>
</div>

```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

PFMO.Web
Views(SupplyCategories)
Index.cshtml

@model
IEnumerable<PFMO.Entities.SupplyCategory>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-sm-4">
        <h2>Supply Categories</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List",
"Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="ibox float-e-margins">
            <div class="ibox-title">
                <h5>List of Supply Categories</h5>
                <div class="ibox-tools">
                    @Html.ActionLink("Change View",
"Table", "SupplyCategories", new { @class =
"btn
btn-primary btn-xs" })
                    @Html.ActionLink("Create New",
"Create", null, new { @class = "btn
btn-primary
btn-xs" })
                </div>
            </div>
            <div class="ibox-content">
                @foreach (var item in Model)
                {
                    <div class="col-lg-3">

```

```

<div class="contact-box center-version">
    <a href="#">
        
        <h3 class="m-b-10"><strong>@Html.DisplayFor(modelItem =>
            item.SupplyCategoryName)</strong></h3>
        <div class="font-bold"><strong>@Html.DisplayFor(modelItem =>
            item.SupplyCategoryDescription)</strong></div>
    </a>
    <div class="contact-box-footer">
        <div class="m-t-xs btn-group">
            @Html.ActionLink("Details",
                "Details", new { id = item.SupplyCategoryID }, new {
                    @class = "btn btn-xs btn-white" })
            @Html.ActionLink("Edit", "Edit",
                new { id = item.SupplyCategoryID }, new { @class =
                    "btn btn-xs btn-white" })
            @Html.ActionLink("Delete",
                "Delete", new { id = item.SupplyCategoryID }, new {
                    @class = "btn btn-xs btn-white" })
        </div>
    </div>
    </div>
</div>

PFMO.Web
Views(SupplyCategories)
Tables.cshtml

@model
IEnumerable<PFMO.Entities.SupplyCategory>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="row wrapper border-bottom white-bg page-heading">
    <div class="col-sm-4">
        <h2>Supply Categories</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">Panel</a>
            </li>
            <li>
                <a href="#">Categories</a>
            </li>
            <li class="active">
                <strong>@Html.ActionLink("List",
                    "Index")</strong>
            </li>
        </ol>
    </div>
</div>

<div class="wrapper wrapper-content animated
fadeInRight">
    <div class="row">
        <div class="col-lg-12">
            <div class="ibox float-e-margins">
                <div class="ibox-title">
                    <h5>List of Supply Category</h5>
                    <div class="ibox-tools">
                        @Html.ActionLink("Change View",
                            "Index", "SupplyCategories", new { @class =
                                "btn btn-primary btn-xs" })
                        @Html.ActionLink("Create New",
                            "Create", null, new { @class =
                                "btn btn-primary btn-xs" })
                    </div>
                </div>
                <div class="ibox-content">
                    <input type="text" class="form-control
                    input-sm m-b-xs" id="filter" placeholder="Search
                    Category">
                    <table class="footable table table-hover"
                        data-filter="#filter" data-page-size="5">
                        <thead>
                            <tr>
                                <th>
                                    @Html.DisplayNameFor(model =>
                                        model.SupplyCategoryName)
                                </th>
                                <th>
                                    @Html.DisplayNameFor(model =>
                                        model.SupplyCategoryDescription)
                                </th>
                            </tr>
                        </thead>
                        <tbody>
                            @foreach (var item in Model)
                            {
                                <tr>
                                    <td>

```

```

        @Html.DisplayFor(modelItem
=> item.SupplyCategoryName)
    </td>
    <td>
        @Html.DisplayFor(modelItem
=> item.SupplyCategoryDescription)
    </td>
    <td>
        @Html.ActionLink("Details",
"Details", new { id = item.SupplyCategoryID }, new {
@class = "btn btn-primary btn-sm" })
        @Html.ActionLink("Edit",
>Edit", new { id = item.SupplyCategoryID }, new {
@class = "btn btn-white btn-sm" })
        @Html.ActionLink("Delete",
"Delete", new { id = item.SupplyCategoryID }, new {
@class = "btn btn-white btn-sm" })
    </td>
</tr>
}
</tbody>

<tfoot>
<tr>
<td colspan="7">
<ul class="pagination pull-right"></ul>
</td>
</tr>
</tfoot>

```

--

#### PFMO.Web Views(Shared) \_DocumentViewerPartial.cshtml

```

@Html.DevExpress().DocumentViewer(settings =>
{
    // The following settings are required for a Report
    // Viewer.
    settings.Name = "documentViewer1";
    // Callback and export route values specify
    // corresponding controllers and their actions.
    // These settings are required as well.
    settings.CallbackRouteValues = new { Controller =
"Home", Action = "DocumentViewerPartial" };

```

```

settings.ExportRouteValues = new { Controller =
"Home", Action = "ExportDocumentViewer" };

// Specify the report service URI and the name of
// the report to display.
settings.SetRemoteSourceSettings(s =>
{
    s.ServiceUri =
@"http://localhost:53813//ReportService1.svc";
    s.ReportTypeName =
"WebApplication1.XtraReport1, WebApplication1";
});
}).GetHtml()

```

#### PFMO.Web Views(Shared) \_Footer.cshtml

```

<div class="footer fixed">
<div class="pull-right">
    Physical Facilities Management Office
</div>
<div>
    <strong>Copyright</strong> Pamantasan ng
Lungsod ng Maynila © 2014-2015
</div>
</div>

```

#### PFMO.Web Views(Shared) \_LandingPageCarousel

```

<div id="inSlider" class="carousel carousel-fade"
data-ride="carousel">
<div class="carousel-inner" role="listbox">
<div class="item">
<div class="container">
<div class="carousel-caption blank">
<h2>Physical Facilities Management
Office</h2>
<p>Management Information
System</p>
<p>
<a class="btn btn-lg btn-primary"
href="@Url.Action("Index", "Home")"
role="button">LOG IN</a>
<a class="caption-link" href="#">
Start your Session.</a>
</p>
</div>
</div>
<!-- Set background for slide in css -->

```



```

<div class="col-lg-3 col-lg-offset-3">
    <address>
        <strong><span
class="navy">Pamantasan ng Lungsod ng
Maynila</span></strong><br />
        795 Folsom Ave, Suite 600<br />
        San Francisco, CA 94107<br />
        <abbr title="Phone">P:</abbr> (123)
    456-7890
    </address>
</div>
<div class="col-lg-4">
    <p class="text-color">
        Consectetur adipisicing elit. Aut eaque,
        totam corporis laboriosam veritatis quis ad
        persiciatis, totam corporis laboriosam veritatis,
        consectetur adipisicing elit quos non quis ad
        persiciatis, totam corporis ea,
    </p>
    </div>
</div>
<div class="row">
    <div class="col-lg-12 text-center">
        <a href="mailto:test@email.com"
class="btn btn-primary">Send us mail</a>
        <p class="m-t-sm">
            Or follow us on social media
        </p>
        <ul class="list-inline social-icon">
            <li>
                <a href="#"><i class="fa fa-
twitter"></i></a>
            </li>
            <li>
                <a href="#"><i class="fa fa-
facebook"></i></a>
            </li>
            <li>
                <a href="#"><i class="fa fa-
linkedin"></i></a>
            </li>
        </ul>
    </div>
    <div class="row">
        <div class="col-lg-8 col-lg-offset-2 text-center
m-t-lg m-b-lg">
            <p><strong>&copy; 2015 Pamantasan ng
Lungsod ng Maynila</strong><br />consectetur
adipisicing elit. Aut eaque, laboriosam veritatis,
quos non quis ad persiciatis, totam corporis ea,
alias ut unde.</p>
        </div>
    </div>
</div>

```

```

</section>

PFMO.Web
Views(Shared)
_LandingPageEvenMoreFeatures.cshtml

<section class="features">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="navy-line"></div>
                <h1>Even more great feautres</h1>
                <p>Donec sed odio dui. Etiam porta sem
malesuada magna mollis euismod. </p>
            </div>
        </div>
        <div class="row features-block">
            <div class="col-lg-3 features-text wow
fadeInLeft">
                <small>INSPINIA</small>
                <h2>Perfectly designed </h2>
                <p>INSPINIA Admin Theme is a premium
admin dashboard template with flat design concept.
It is fully responsive admin dashboard template built
with Bootstrap 3+ Framework, HTML5 and CSS3,
Media query. It has a huge collection of reusable UI
components and integrated with latest jQuery
plugins.</p>
                <a href="" class="btn btn-primary">Learn
more</a>
            </div>
            <div class="col-lg-6 text-right m-t-n-lg wow
zoomIn">
                <img
src "~/Content/LandingPage/img/iphone.jpg"
class="img-responsive" alt="dashboard">
            </div>
            <div class="col-lg-3 features-text text-right
wow fadeInRight">
                <small>INSPINIA</small>
                <h2>Perfectly designed </h2>
                <p>INSPINIA Admin Theme is a premium
admin dashboard template with flat design concept.
It is fully responsive admin dashboard template built
with Bootstrap 3+ Framework, HTML5 and CSS3,
Media query. It has a huge collection of reusable UI
components and integrated with latest jQuery
plugins.</p>
                <a href="" class="btn btn-primary">Learn
more</a>
            </div>
        </div>
    </div>
</section>
```

**PFMO.Web**  
**Views(Shared)**  
**\_LandingPageFeatures.cshtml**

```
<section class="container features">
    <div class="row">
        <div class="col-lg-12 text-center">
            <div class="navy-line"></div>
            <h1>Pamantasan ng Lungsod ng Maynila<br/>
            /> <span class="navy">University of the City of
            Manila</span> </h1>
            <p>Gen. Luna cor. Muralla St., Intramuros,
            Manila, Metro Manila, Philippines</p>
        </div>
    </div>
    <div class="row">
        <div class="col-md-3 text-center wow
        fadeInLeft">
            <div>
                <i class="fa fa-mobile features-icon"></i>
                <h2>Full responsive</h2>
                <p>Donec sed odio dui. Etiam porta sem
                malesuada magna mollis euismod. Nullam id dolor
                id nibh ultricies vehicula ut id elit. Morbi leo
                risus.</p>
            </div>
            <div class="m-t-lg">
                <i class="fa fa-bar-chart features-
                icon"></i>
                <h2>6 Charts Library</h2>
                <p>Donec sed odio dui. Etiam porta sem
                malesuada magna mollis euismod. Nullam id dolor
                id nibh ultricies vehicula ut id elit. Morbi leo
                risus.</p>
            </div>
        </div>
        <div class="col-md-6 text-center wow
        zoomIn">
            
        </div>
        <div class="col-md-3 text-center wow
        fadeInRight">
            <div>
                <i class="fa fa-envelope features-
                icon"></i>
                <h2>Mail pages</h2>
                <p>Donec sed odio dui. Etiam porta sem
                malesuada magna mollis euismod. Nullam id dolor
                id nibh ultricies vehicula ut id elit. Morbi leo
                risus.</p>
            </div>
            <div class="m-t-lg">
```

```
<i class="fa fa-google features-icon"></i>
<h2>AngularJS version</h2>
<p>Donec sed odio dui. Etiam porta sem
malesuada magna mollis euismod. Nullam id dolor
id nibh ultricies vehicula ut id elit. Morbi leo
risus.</p>
</div>
</div>
</div>
<div class="row">
    <div class="col-lg-12 text-center">
        <div class="navy-line"></div>
        <h1>Premier Scholar's University</h1>
        <p>Gen. Luna cor. Muralla St., Intramuros,
        Manila, Metro Manila, Philippines</p>
    </div>
</div>
<div class="row features-block">
    <div class="col-lg-6 features-text wow
    fadeInLeft">
        <small>Pamantasan ng Lungsod ng
        Maynila</small>
        <h2>The City's Finest </h2>
        <p>INSPINIA Admin Theme is a premium
        admin dashboard template with flat design concept.
        It is fully responsive admin dashboard template built
        with Bootstrap 3+ Framework, HTML5 and CSS3,
        Media query. It has a huge collection of reusable UI
        components and integrated with latest jQuery
        plugins.</p>
        <a href="http://www.plm.edu.ph" class="btn
        btn-primary">Learn more</a>
    </div>
    <div class="col-lg-6 text-right wow
    fadeInRight">
        
    </div>
</div>
</section>
```

**PFMO.Web**  
**Views(Shared)**  
**\_LandingPageMore&MoreGreaterFeatures.cshtml**

```
<section class="features">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="navy-line"></div>
                <h1>More and more extra great
                feautres</h1>
                <p>Donec sed odio dui. Etiam porta sem
                malesuada magna mollis euismod. </p>
            </div>
        </div>
    </div>
</section>
```

```

        </div>
    </div>
    <div class="row">
        <div class="col-lg-5 col-lg-offset-1 features-text">
            <small>INSPINIA</small>
            <h2>Perfectly designed </h2>
            <i class="fa fa-bar-chart big-icon pull-right"></i>
            <p>INSPINIA Admin Theme is a premium admin dashboard template with flat design concept. It is fully responsive admin dashboard template built with Bootstrap 3+ Framework, HTML5 and CSS3, Media query. It has a huge collection of reusable UI components and integrated with.</p>
        </div>
        <div class="col-lg-5 features-text">
            <small>INSPINIA</small>
            <h2>Perfectly designed </h2>
            <i class="fa fa-bolt big-icon pull-right"></i>
            <p>INSPINIA Admin Theme is a premium admin dashboard template with flat design concept. It is fully responsive admin dashboard template built with Bootstrap 3+ Framework, HTML5 and CSS3, Media query. It has a huge collection of reusable UI components and integrated with.</p>
        </div>
        <div class="row">
            <div class="col-lg-5 col-lg-offset-1 features-text">
                <small>INSPINIA</small>
                <h2>Perfectly designed </h2>
                <i class="fa fa-clock-o big-icon pull-right"></i>
                <p>INSPINIA Admin Theme is a premium admin dashboard template with flat design concept. It is fully responsive admin dashboard template built with Bootstrap 3+ Framework, HTML5 and CSS3, Media query. It has a huge collection of reusable UI components and integrated with.</p>
            </div>
            <div class="col-lg-5 features-text">
                <small>INSPINIA</small>
                <h2>Perfectly designed </h2>
                <i class="fa fa-users big-icon pull-right"></i>
                <p>INSPINIA Admin Theme is a premium admin dashboard template with flat design concept. It is fully responsive admin dashboard template built with Bootstrap 3+ Framework, HTML5 and CSS3, Media query. It has a huge collection of reusable UI components and integrated with.</p>
            </div>
        </div>
    </div>

```

</div>

</section>

**PFMO.Web**

**Views(Shared)**

**\_LandingPagePricing.cshtml**

<section id="pricing" class="pricing">

<div class="container">

<div class="row m-b-lg">

<div class="col-lg-12 text-center">

<div class="navy-line"></div>

<h1>App Pricing</h1>

<p>Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod.</p>

</div>

</div>

<div class="row">

<div class="col-lg-4 wow zoomIn">

<ul class="pricing-plan list-unstyled">

<li class="pricing-title">

Basic

</li>

<li class="pricing-desc">

Lorem ipsum dolor sit amet, illum fastidii dissentias quo ne. Sea ne sint animal iisque, nam an soluta sensibus.

</li>

<li class="pricing-price">

\$16</span> / month

</li>

<li>

Dashboards

</li>

<li>

Projects view

</li>

<li>

Contacts

</li>

<li>

Calendar

</li>

<li>

AngularJs

</li>

<li>

<a class="btn btn-primary btn-xs" href="#">Signup</a>

</li>

</ul>

</div>

<div class="col-lg-4 wow zoomIn">

```

<ul class="pricing-plan list-unstyled selected">
    <li class="pricing-title">
        Standard
    </li>
    <li class="pricing-desc">
        Lorem ipsum dolor sit amet, illum fastidii dissentias quo ne. Sea ne sint animal iisque, nam an soluta sensibus.
    </li>
    <li class="pricing-price">
        <span>$22</span> / month
    </li>
    <li>
        Dashboards
    </li>
    <li>
        Projects view
    </li>
    <li>
        Contacts
    </li>
    <li>
        Calendar
    </li>
    <li>
        AngularJs
    </li>
    <li>
        <a class="btn btn-primary btn-xs" href="#">Signup</a>
    </li>
</ul>
</div>

<div class="col-lg-4 wow zoomIn">
    <ul class="pricing-plan list-unstyled">
        <li class="pricing-title">
            Premium
        </li>
        <li class="pricing-desc">
            Lorem ipsum dolor sit amet, illum fastidii dissentias quo ne. Sea ne sint animal iisque, nam an soluta sensibus.
        </li>
        <li class="pricing-price">
            <span>$160</span> / month
        </li>
        <li>
            Dashboards
        </li>
        <li>

```

Projects view  
 </li>  
 <li>  
 Contacts  
 </li>  
 <li>  
 Calendar  
 </li>  
 <li>  
 AngularJs  
 </li>  
 <li>  
 <a class="btn btn-primary btn-xs" href="#">Signup</a>  
 </li>  
</ul>  
</div>  
<div class="row m-t-lg">  
<div class="col-lg-8 col-lg-offset-2 text-center m-t-lg">  
 <p>\*Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. <span class="navy">Various versions</span> have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).</p>  
</div>  
</div>  
</div>

### PFMO.Web Views(Shared) \_LandingPageServices.cshtml

```

<section id="features" class="container services">
    <div class="row">
        <div class="col-sm-3">
            <h2>Full responsive</h2>
            <p>Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod. Nullam id dolor id nibh ultricies vehicula ut id elit. Morbi leo risus.</p>
            <p><a class="navy-link" href="#" role="button">Details &raquo;</a></p>
        </div>
        <div class="col-sm-3">
            <h2>LESS/SASS Files</h2>
            <p>Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod. Nullam id dolor

```

```

id nibh ultricies vehicula ut id elit. Morbi leo
risus.</p>
<p><a class="navy-link" href="#" role="button">Details &raquo;</a></p>
</div>
<div class="col-sm-3">
<h2>6 Charts Library</h2>
<p>Donec sed odio dui. Etiam porta sem
malesuada magna mollis euismod. Nullam id dolor
id nibh ultricies vehicula ut id elit. Morbi leo
risus.</p>
<p><a class="navy-link" href="#" role="button">Details &raquo;</a></p>
</div>
<div class="col-sm-3">
<h2>Advanced Forms</h2>
<p>Donec sed odio dui. Etiam porta sem
malesuada magna mollis euismod. Nullam id dolor
id nibh ultricies vehicula ut id elit. Morbi leo
risus.</p>
<p><a class="navy-link" href="#" role="button">Details &raquo;</a></p>
</div>
</div>
</section>

```

**PFMO.Web**  
**Views(Shared)**  
**\_LandingPageTeam.cshtml**

```

<section id="team" class="gray-section team">
<div class="container">
<div class="row m-b-lg">
<div class="col-lg-12 text-center">
<div class="navy-line"></div>
<h1>Management Information
System</h1>
<p>Physical Facilities Management
Office</p>
</div>
<div class="row">
<div class="col-sm-4 wow fadelnLeft">
<div class="team-member">

<h4><span
class="navy">Personnel</span> Account</h4>
<p>Lorem ipsum dolor sit amet, illum
fastidii dissentias quo ne. Sea ne sint animal iisque,
nam an soluta sensibus. </p>
@*<ul class="list-inline social-icon">

```

```

<li>
<a href="#"><i class="fa fa-
twitter"></i></a>
</li>
<li>
<a href="#"><i class="fa fa-
facebook"></i></a>
</li>
<li>
<a href="#"><i class="fa fa-
linkedin"></i></a>
</li>
</ul>*@
</div>
</div>
<div class="col-sm-4">
<div class="team-member wow zoomIn">

<h4><span
class="navy">Manager</span> Account</h4>
<p>Lorem ipsum dolor sit amet, illum
fastidii dissentias quo ne. Sea ne sint animal iisque,
nam an soluta sensibus.</p>
@*<ul class="list-inline social-icon">
<li>
<a href="#"><i class="fa fa-
twitter"></i></a>
</li>
<li>
<a href="#"><i class="fa fa-
facebook"></i></a>
</li>
<li>
<a href="#"><i class="fa fa-
linkedin"></i></a>
</li>
</ul>*@
</div>
</div>
<div class="col-sm-4 wow fadelnRight">
<div class="team-member">

<h4><span class="navy">Section
Head</span> Account</h4>
<p>Lorem ipsum dolor sit amet, illum
fastidii dissentias quo ne. Sea ne sint animal iisque,
nam an soluta sensibus.</p>
@*<ul class="list-inline social-icon">
<li>
<a href="#"><i class="fa fa-
twitter"></i></a>

```

```

</li>
<li>
    <a href="#"><i class="fa fa-facebook"></i></a>
</li>
<li>
    <a href="#"><i class="fa fa-linkedin"></i></a>
</li>
</ul>*@
</div>
</div>
</div>
<div class="row">
    <div class="col-lg-8 col-lg-offset-2 text-center m-t-lg m-b-lg">
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aut eaque, laboriosam veritatis, quos non quis ad perspiciatis, totam corporis ea, alias ut unde.</p>
    </div>
</div>
</div>
</section>

PFMO.Web
Views(Shared)
_LandingPageTestimonials.cshtml

I <section id="testimonials" class="navy-section testimonials" style="margin-top: 0">

    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center wow zoomIn">
                <i class="fa fa-comment big-icon"></i>
                <h1>
                    What our users say
                </h1>
                <div class="testimonials-text">
                    <i>"Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like)."</i>
                </div>
                <small>
                    <strong>12.02.2014 - Andy Smith</strong>
                </small>
            </div>
        </div>
    </div>
</section>

PFMO.Web
Views(Shared)
_LandingPageTimeline.cshtml

<section class="timeline gray-section">
    <div class="container">
        <div class="row">
            <div class="col-lg-12 text-center">
                <div class="navy-line"></div>
                <h1>Our workflow</h1>
                <p>Donec sed odio dui. Etiam porta sem malesuada magna mollis euismod.</p>
            </div>
        </div>
        <div class="row features-block">
            <div class="col-lg-12">
                <div id="vertical-timeline" class="vertical-container light-timeline center-orientation">
                    <div class="vertical-timeline-block">
                        <div class="vertical-timeline-icon navy-bg">
                            <i class="fa fa-briefcase"></i>
                        </div>
                        <div class="vertical-timeline-content">
                            <h2>Meeting</h2>
                            <p>
                                Conference on the sales results for the previous year. Monica please examine sales trends in marketing and products. Below please find the current status of the sale.
                            </p>
                            <a href="#" class="btn btn-xs btn-primary">More info</a>
                            <span class="vertical-date"> Today <br /> <small>Dec 24</small> </span>
                        </div>
                    </div>
                    <div class="vertical-timeline-block">
                        <div class="vertical-timeline-icon navy-bg">
                            <i class="fa fa-file-text"></i>
                        </div>
                        <div class="vertical-timeline-content">
                            <h2>Decision</h2>
                            <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>

```

```


Ipsum has been the industry's standard dummy text ever since.</p>
    <a href="#" class="btn btn-xs btn-primary"> More info</a>
        <span class="vertical-date">
Tomorrow <br /> <small>Dec 26</small> </span>
            </div>
        </div>

        <div class="vertical-timeline-block">
            <div class="vertical-timeline-icon navy-bg">
                <i class="fa fa-cogs"></i>
            </div>

            <div class="vertical-timeline-content">
                <h2>Implementation</h2>
                <p>Go to shop and find some products. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's. </p>
                <a href="#" class="btn btn-xs btn-primary"> More info</a>
                    <span class="vertical-date">
Monday <br /> <small>Jan 02</small> </span>
                        </div>
                    </div>

                </div>
            </div>
        </div>
    </div>
</div>


```

### **PFMO.Web Views(Shared) \_LandingPageTopNavBar.cshtml**

```

<div class="navbar-wrapper">
    <nav class="navbar navbar-default navbar-fixed-top" role="navigation">
        <div class="container">
            <div class="navbar-header page-scroll">
                <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
                    <span class="sr-only">Toggle navigation</span>

```

```

                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">Physical Facilities Management Office</a>
        </div>
        <div id="navbar" class="navbar-collapse collapse">
            <ul class="nav navbar-nav navbar-right">
                <li><a href="@Url.Action("Register", "Account")">Register</a></li>
                <li><a class="page-scroll" href="#page-top">Home</a></li>
                <li><a class="page-scroll" href="#features">University</a></li>
                <li><a class="page-scroll" href="#team">System</a></li>
                <li><a class="page-scroll" href="#testimonials">Features</a></li>
                <li><a class="page-scroll" href="#contact">Contact</a></li>
            </ul>
        </div>
    </div>

```

### **PFMO.Web Views(Shared) \_Layout.cshtml**

```

@using PFMO.Web.Models
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <!-- Icon -->
        <link rel="icon" href="~/Content/favicon.png" />
        <title>PFMO Management | @ ViewBag.Title</title>
        <link href='https://fonts.googleapis.com/css?family=Open+Sans:400,300,600,700' rel='stylesheet' type='text/css' />

```

```

<!-- Add local styles, mostly for plugins css file -->
@if (IsSectionDefined("Styles"))
{@RenderSection("Styles", required: false)}

<!-- Add jQuery Style direct - used for jQGrid
plugin -->
<link
href="@Url.Content("~/Scripts/plugins/jquery-
ui/jquery-ui.css")" rel="stylesheet" type="text/css"
/>

<!-- Primary Inspinia style -->
@Styles.Render("~/Content/css")
@Styles.Render("~/font-awesome/css")

<!-- Toastr style -->
@Styles.Render("~/plugins/toastr")
@Styles.Render("~/plugins/toastrStyles")

<!-- Bootstrap Tour -->
@Styles.Render("~/plugins/tourStyles")
@Styles.Render("~/plugins/tour")

<!-- FooTable -->
@Styles.Render("~/plugins/footable")
@Styles.Render("~/plugins/footableStyles")

<!-- Sweet Alert -->
@Styles.Render("~/plugins/sweetalert")

<!-- Morris -->
<link
href("~/Content/pcss/plugins/morris/morris-
0.4.3.min.css" rel="stylesheet"

<!-- Gritter -->
<link
href "~/Content/plugins/gritter/jquery.gritter.css"
rel="stylesheet"

<!-- Calendar -->
<link
href "~/Content/css/plugins/iCheck/custom.css"
rel="stylesheet"
<link
href "~/Content/css/plugins/fullcalendar/fullcalend
ar.css" rel="stylesheet"
<link
href "~/Content/css/plugins/fullcalendar/fullcalend
ar.print.css" rel='stylesheet' media='print'

</head>

<body class="fixed-sidebar fixed-nav fixed-nav-basic
md-skin">

```

```

<!-- Skin configuration box -->
@Html.Partial("_SkinConfig")

<!-- PageClass give you ability to specify custom
style for specific view based on action -->

<div id="wrapper"
@*class="@Html.PageClass"*@>

<!-- Navigation -->
@Html.Partial("_Navigation")

<!-- Page wrapper -->
<div id="page-wrapper" class="gray-bg
@ViewBag.SpecialClass">

<!-- Top Navbar -->
@Html.Partial("_TopNavbar")

<!-- Main view -->
@RenderBody()

<!-- Footer -->
@Html.Partial("_Footer")

</div>
<!-- End page wrapper-->

</div>
<!-- End wrapper-->


```

```

<!-- Section for main scripts render -->
<script
src "~/Content/js/plugins/fullcalendar/moment.mi
n.js"></script>
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@Scripts.Render("~/plugins/metsiMenu")
@Scripts.Render("~/plugins/pace")
@Scripts.Render("~/plugins/slimScroll")
@Scripts.Render("~/bundles/inspinia")

<!-- Skin config script -->
@Scripts.Render("~/bundles/skinConfig")

<!-- Handler for local scripts -->
@RenderSection("scripts", required: false)

<!-- Bootstrap Tour -->
@Scripts.Render("~/plugins/tourStyles")
@Scripts.Render("~/plugins/tour")
<script>

$(document).ready(function (){


```

```

// Instance the tour
var tour = new Tour({
  steps: [
    {
      element: "#step1",
      title: "Title of my step",
      content: "Content of my step",
      placement: "top",
      backdrop: true,
      backdropContainer: '#wrapper',
      onShown: function (tour) {
        $('body').addClass('tour-open')
      },
      onHidden: function (tour) {
        $('body').removeClass('tour-close')
      }
    },
    {
      element: "#step2",
      title: "Title of my step",
      content: "Content of my step",
      placement: "top",
      backdrop: true,
      backdropContainer: '#wrapper',
      onShown: function (tour){
        $('body').addClass('tour-open')
      },
      onHidden: function (tour){
        $('body').removeClass('tour-close')
      }
    },
    {
      element: "#step3",
      title: "Title of my step",
      content: "Content of my step",
      placement: "top",
      backdrop: true,
      backdropContainer: '#wrapper',
      onShown: function (tour) {
        $('body').addClass('tour-open')
      },
      onHidden: function (tour) {
        $('body').removeClass('tour-close')
      }
    },
    {
      element: "#step4",
      title: "Title of my step",
      content: "Content of my step",
      placement: "top",
      backdrop: true,
      backdropContainer: '#wrapper',
      onShown: function (tour) {
        $('body').addClass('tour-open')
      }
    }
  ],
  onHidden: function (tour) {
    $('body').removeClass('tour-close')
  }
},
{
  element: "#step5",
  title: "Title of my step",
  content: "Content of my step",
  placement: "top",
  backdrop: true,
  backdropContainer: '#wrapper',
  onShown: function (tour) {
    $('body').addClass('tour-open')
  },
  onHidden: function (tour) {
    $('body').removeClass('tour-close')
  }
}
]);
// Initialize the tour
tour.init();

$('.startTour').click(function(){
  tour.restart();
  // Start the tour
  // tour.start();
});

});
```

</script>

@Scripts.Render("~/plugins/tourStyles")

<!-- FooTable -->

@Scripts.Render("~/plugins/footable")

<!-- Page-Level Scripts -->

<script>

\$(document).ready(function() {

\$('.footable').footable();

\$('.footable2').footable();

});

</script>

<!-- Sweet alert -->

@Scripts.Render("~/plugins/sweetalert")

<script>

\$(document).ready(function () {

```

$('.basic').click(function(){
    swal({
        title: "Welcome in Alerts",
        text: "Lorem Ipsum is simply dummy text of
the printing and typesetting industry."
    });
});

$('.success').click(function(){
    swal({
        title: "Success!",
        text: "Transaction completed!",
        type: "success"
    });
});

$('.warning').click(function () {
    swal({
        title: "Are you sure?",
        text: "You will not be able to recover this
file!",
        type: "warning",
        showCancelButton: true,
        confirmButtonColor: "#DD6B55",
        confirmButtonText: "Yes, delete it!",
        closeOnConfirm: false
    }, function () {
        swal("Deleted!", "Your file has been
deleted.", "success");
    });
});

$('.warning2').click(function () {
    swal({
        title: "Are you sure?",
        text: "Your will not be able to recover
this file!",
        type: "warning",
        showCancelButton: true,
        confirmButtonColor: "#DD6B55",
        confirmButtonText: "Yes, delete it!",
        cancelButtonText: "No, cancel!",
        closeOnConfirm: false,
        closeOnCancel: false },
    function (isConfirm) {
        if (isConfirm) {
            swal("Deleted!", "Your file has been
deleted.", "success");
        } else {
            swal("Cancelled", "Your file is safe
:", "error");
        }
    });
});
});
```

```

});
```

```

</script>
```

```

<!-- slimScroll -->
@Scripts.Render("~/plugins/slimScroll")
```

```

<!-- jQuery UI custom -->
<script src "~/Content/js/jquery-
ui.custom.min.js"></script>
```

```

<!-- iCheck -->
<script
src "~/Content/js/plugins/iCheck/icheck.min.js"></
script>
```

```

<!-- jQuery UI custom -->
<script src "~/Content/js/jquery-
ui.custom.min.js"></script>
```

```

<!-- Full Calendar -->
<script
src "~/Content/js/plugins/fullcalendar/fullcalendar.
min.js"></script>
<script>
$(function () {
```

```

/* initialize the external events
-----*/
function ini_events(ele) {
    ele.each(function () {
```

```

        // create an Event Object
(http://arshaw.com/fullcalendar/docs/event\_data/Event\_Object/)
        // it doesn't need to have a start or end
        var eventObject = {
            title: $.trim($(this).text()) // use the
element's text as the event title
        };

        // store the Event Object in the DOM
        // element so we can get to it later
        $(this).data('eventObject', eventObject);

        // make the event draggable using
        // jQuery UI
        $(this).draggable({
            zIndex: 1070,
            revert: true, // will cause the event to
go back to its
```

```

        revertDuration: 0 // original position
after the drag
    });

}

ini_events($('#external-events div.external-
event'));

/* initialize the calendar
-----
----*/
//Date for the calendar events (dummy data)
var date = new Date();
var d = date.getDate(),
m = date.getMonth(),
y = date.getFullYear();
$('#calendar').fullCalendar({
    header: {
        left: 'prev,next today',
        center: 'title',
        right: 'month,agendaWeek,agendaDay'
    },
    buttonText: {
        today: 'today',
        month: 'month',
        week: 'week',
        day: 'day'
    },
    //Random default events
    events: [],
    editable: true,
    droppable: true, // this allows things to be
dropped onto the calendar !!!
    drop: function (date, allDay) { // this
function is called when something is dropped

        // retrieve the dropped element's stored
Event Object
        var originalEventObject =
$(this).data('eventObject');

        // we need to copy it, so that multiple
events don't have a reference to the same object
        var copiedEventObject = $.extend({}, 
originalEventObject);

        // assign it the date that was reported
        copiedEventObject.start = date;
        copiedEventObject.allDay = allDay;
        copiedEventObject.backgroundColor =
$(this).css("background-color");
        copiedEventObject.borderColor =
$(this).css("border-color");
    }
});
ini_events($('div.event'));
});
```

```

        // render the event on the calendar
        // the last 'true' argument determines if
the event "sticks"
(http://arshaw.com/fullcalendar/docs/event\_rendering/renderEvent/)
        $('#calendar').fullCalendar('renderEvent',
copiedEventObject, true);

        // is the "remove after drop" checkbox
checked?
        if ($('#drop-remove').is(':checked')) {
            // if so, remove the element from the
"Draggable Events" list
            $(this).remove();
        }

    });
}

/* ADDING EVENTS */
var currColor = "#3c8dbc"; //Red by default
//Color chooser button
var colorChooser = $("#color-chooser-btn");
$("#color-chooser > li > a").click(function (e) {
    e.preventDefault();
    //Save color
    currColor = $(this).css("color");
    //Add color effect to button
    $('#add-new-event').css({ "background-
color": currColor, "border-color": currColor });
});
$("#add-new-event").click(function (e) {
    e.preventDefault();
    //Get value and make sure it is not null
    var val = $("#new-event").val();
    if (val.length == 0) {
        return;
    }

    //Create events
    var event = $("<div />");
    event.css({ "background-color": currColor,
"border-color": currColor, "color": "#fff" })
.addClass("external-event");
    event.html(val);
    $('#external-events').prepend(event);

    //Add draggable funtionality
    ini_events(event);

    //Remove event from text input
    $("#new-event").val("");
});
```

```

    });
}
</script>
```

```

<!-- Flot -->
<script
src="~/Content/js/plugins/flot/jquery.flot.js"></script>
<script
src="~/Content/js/plugins/flot/jquery.flot.tooltip.min.js"></script>
<script
src="~/Content/js/plugins/flot/jquery.flot.spline.js"
></script>
<script
src="~/Content/js/plugins/flot/jquery.flot.resize.js"
></script>
<script
src="~/Content/js/plugins/flot/jquery.flot.pie.js"></script>

<!-- ChartJS-->
<script
src="~/Content/js/plugins/chartJs/Chart.min.js"></script>

<!-- Peity -->
<script
src="~/Content/js/plugins/peity/jquery.peity.min.js"
></script>
<script src="~/Content/js/demo/peity-
demo.js"></script>
<script>
$(document).ready(function() {

    var d1 = [[1262304000000, 6],
[1264982400000, 3057], [1267401600000, 20434],
[1270080000000, 31982], [1272672000000, 26602],
[1275350400000, 27826], [1277942400000, 24302],
[1280620800000, 24237], [1283299200000, 21004],
[1285891200000, 12144], [1288569600000, 10577],
[1291161600000, 10295]];
    var d2 = [[1262304000000, 5],
[1264982400000, 200], [1267401600000, 1605],
[1270080000000, 6129], [1272672000000, 11643],
[1275350400000, 19055], [1277942400000, 30062],
[1280620800000, 39197], [1283299200000, 37000],
[1285891200000, 27000], [1288569600000, 21000],
[1291161600000, 17000]];

    var data1 = [
        { label: "Data 1", data: d1, color:
'#17a084'},
        { label: "Data 2", data: d2, color: '#127e68'
    }
];
    $.plot($("#flot-chart1"), data1, {
        axis: {
            tickDecimals: 0
        },
        series: {
            lines: {
                show: true,
                fill: true,
                fillColor: {
                    colors: [{ opacity: 1
}, {
                        opacity: 1
                    }]
                },
                points: {
                    width: 0.1,
                    show: false
                },
                grid: {
                    show: false,
                    borderWidth: 0
                },
                legend: {
                    show: false,
                }
            });
        var lineData = {
            labels: ["January", "February", "March",
"April", "May", "June", "July"],
            datasets: [
                {
                    label: "Example dataset",
                    fillColor: "rgba(220,220,220,0.5)",
                    strokeColor: "rgba(220,220,220,1)",
                    pointColor: "rgba(220,220,220,1)",
                    pointStrokeColor: "#fff",
                    pointHighlightFill: "#fff",
                    pointHighlightStroke:
"rgba(220,220,220,1)",
                    data: [65, 59, 40, 51, 36, 25, 40]
                },
                {
                    label: "Example dataset",
                    fillColor: "rgba(26,179,148,0.5)",
                    strokeColor: "rgba(26,179,148,0.7)",
                    pointColor: "rgba(26,179,148,1)",
                    pointStrokeColor: "#fff",
                    pointHighlightFill: "#fff",
                    pointHighlightStroke:
"rgba(26,179,148,1)",
                    data: [48, 48, 60, 39, 56, 37, 30]
                }
            ]
        }
    }
}
)

```

```

        ]
    };

    var lineOptions = {
        scaleShowGridLines: true,
        scaleGridLineColor: "rgba(0,0,0,.05)",
        scaleGridLineWidth: 1,
        bezierCurve: true,
        bezierCurveTension: 0.4,
        pointDot: true,
        pointDotRadius: 4,
        pointDotStrokeWidth: 1,
        pointHitDetectionRadius: 20,
        datasetStroke: true,
        datasetStrokeWidth: 2,
        datasetFill: true,
        responsive: true,
    };

    var ctx =
document.getElementById("lineChart").getContext(
"2d");
    var myNewChart = new
Chart(ctx).Line(lineData, lineOptions);

    });
</script>

<!-- Flot -->
<script
src="~/Content/js/plugins/flot/jquery.flot.js"></scri
pt>
<script
src="~/Content/js/plugins/flot/jquery.flot.tooltip.mi
n.js"></script>
<script
src="~/Content/js/plugins/flot/jquery.flot.spline.js"
></script>
<script
src="~/Content/js/plugins/flot/jquery.flot.resize.js"
></script>
<script
src="~/Content/js/plugins/flot/jquery.flot.pie.js"></
script>

<!-- Sparkline -->
<script
src="~/Content/js/plugins/sparkline/jquery.sparklin
e.min.js"></script>

<!-- Sparkline demo data -->
<script src="~/Content/js/demo/sparkline-
demo.js"></script>
<!-- ChartJS-->
<script
src="~/Content/js/plugins/chartJs/Chart.min.js"></s
cript>

<!-- GITTER -->
<script
src="~/Content/js/plugins/gritter/jquery.gritter.min.
js"></script>

<!-- Toastr -->
<script
src="~/Content/js/plugins/toastr/toastr.min.js"></s
cript>
<script>
$(document).ready(function() {
    setTimeout(function() {
        toastr.options = {
            //closeButton: true,
            //progressBar: true,
            //showMethod: 'slideDown',
            //timeOut: 4000
        };
        //toastr.success('Responsive Admin
Theme', 'Welcome to INSPINIA');
    }, 1300);
}

var data1 = [
[0,4],[1,8],[2,5],[3,10],[4,4],[5,16],[6,5],[7,11],[8,6],[9,11],[10,30],[11,10],[12,13],[13,4],[14,3],[15,3],[16,6]
];
var data2 = [
[0,1],[1,0],[2,2],[3,0],[4,1],[5,3],[6,1],[7,5],[8,2],[9,3],[10,2],[11,1],[12,0],[13,2],[14,8],[15,0],[16,0]
];
$("#flot-dashboard-chart").length &&
$.plot($("#flot-dashboard-chart"), [
    data1, data2
],
{
    series: {
        lines: {
            show: false,
            fill: true
        },
        splines: {
            show: true,
            tension: 0.4,
            lineWidth: 1,
            fill: 0.4
        }
    }
});

```

```

        },
        points: {
            radius: 0,
            show: true
        },
        shadowSize: 2
    },
    grid: {
        hoverable: true,
        clickable: true,
        tickColor: "#d5d5d5",
        borderWidth: 1,
        color: '#d5d5d5'
    },
    colors: ["#1ab394", "#1C84C6"],
    xaxis:{},
    yaxis: {
        ticks: 4
    },
    tooltip: false
}
);

var doughnutData = [
{
    value: 300,
    color: "#a3e1d4",
    highlight: "#1ab394",
    label: "App"
},
{
    value: 50,
    color: "#dedede",
    highlight: "#1ab394",
    label: "Software"
},
{
    value: 100,
    color: "#A4CEE8",
    highlight: "#1ab394",
    label: "Laptop"
}
];

var doughnutOptions = {
    segmentShowStroke: true,
    segmentStrokeColor: "#fff",
    segmentStrokeWidth: 2,
    percentageInnerCutout: 45, // This is 0 for
Pie charts
    animationSteps: 100,
    animationEasing: "easeOutBounce",
    animateRotate: true,
    animateScale: false
};

};

var ctx =
document.getElementById("doughnutChart").getContext("2d");
var DoughnutChart = new
Chart(ctx).Doughnut(doughnutData,
doughnutOptions);

var polarData = [
{
    value: 300,
    color: "#a3e1d4",
    highlight: "#1ab394",
    label: "App"
},
{
    value: 140,
    color: "#dedede",
    highlight: "#1ab394",
    label: "Software"
},
{
    value: 200,
    color: "#A4CEE8",
    highlight: "#1ab394",
    label: "Laptop"
}
];

var polarOptions = {
    scaleShowLabelBackdrop: true,
    scaleBackdropColor:
"rgba(255,255,255,0.75)",
    scaleBeginAtZero: true,
    scaleBackdropPaddingY: 1,
    scaleBackdropPaddingX: 1,
    scaleShowLine: true,
    segmentShowStroke: true,
    segmentStrokeColor: "#fff",
    segmentStrokeWidth: 2,
    animationSteps: 100,
    animationEasing: "easeOutBounce",
    animateRotate: true,
    animateScale: false
};

var ctx =
document.getElementById("polarChart").getContext("2d");
var Polarchart = new
Chart(ctx).PolarArea(polarData, polarOptions);

});
</script>

```

```

</body>
</html>

PFMO.Web
Views(Shared)
_Layout_2.cshtml

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8"/>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible"
content="IE=edge">
<title>INSPINIA | @ViewBag.Title</title>
<link
href='http://fonts.googleapis.com/css?family=Open
+Sans:400,300,600,700' rel='stylesheet'
type='text/css'>
<!-- Add local styles, mostly for plugins css file -->
@if (IsSectionDefined("Styles"))
{@RenderSection("Styles", required: false)}

<!-- Add jQuery Style direct - used for jqGrid
plugin -->
<link
href="@Url.Content("~/Scripts/plugins/jquery-
ui/jquery-ui.css")" rel="stylesheet" type="text/css"
/>

<!-- Primary Inspinia style -->
@Styles.Render("~/font-awesome/css")
@Styles.Render("~/Content/css")
</head>
<body class="top-navigation">

<!-- Skin configuration box -->
@Html.Partial("_SkinConfig")

<!-- Wrapper-->
<!-- PageClass give you ability to specify custom
style for specific view based on action -->
<div id="wrapper" class="@Html.PageClass()">

<!-- Page wraper -->
<div id="page-wrapper" class="gray-bg
@ ViewBag.SpecialClass">

<!-- Top Navbar -->
@Html.Partial("_TopNavbar2")
```

```

<!-- Main view -->
@RenderBody()

<!-- Footer -->
@Html.Partial("_Footer")

</div>
<!-- End page wrapper-->

</div>
<!-- End wrapper-->

<!-- Section for main scripts render -->
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@Scripts.Render("~/plugins/slimScroll")
@Scripts.Render("~/bundles/inspinia")

<!-- Skin config script - only for demo purpose-->
@Scripts.Render("~/bundles/skinConfig")

<!-- Handler for local scripts -->
@RenderSection("scripts", required: false)
</body>
</html>
```

**PFMO.Web**
**Views(Shared)**
**\_Layout\_3.cshtml**

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8"/>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible"
content="IE=edge">
<title>INSPINIA | @ViewBag.Title</title>
<link
href='http://fonts.googleapis.com/css?family=Open
+Sans:400,300,600,700' rel='stylesheet'
type='text/css'>
<!-- Add local styles, mostly for plugins css file -->
@if (IsSectionDefined("Styles"))
{@RenderSection("Styles", required: false)}

```

<!-- Add jQuery Style direct - used for jqGrid
plugin -->
<link
href="@Url.Content("~/Scripts/plugins/jquery-
ui/jquery-ui.css")" rel="stylesheet" type="text/css"
/>
```

```

ui/jquery-ui.css")" rel="stylesheet" type="text/css"
/>


    <!-- Primary Inspinia style -->
    @Styles.Render("~/font-awesome/css")
    @Styles.Render("~/Content/css")
</head>
<body class="fixed-sidebar no-skin-config full-height-layout">

    <!-- Wrapper-->
    <!-- PageClass give you ability to specify custom
style for specific view based on action -->
    <div id="wrapper" class="@Html.PageClass()">

        <!-- Navigation -->
        @Html.Partial("_Navigation")

        <!-- Page wraper -->
        <div id="page-wrapper" class="gray-bg
@ ViewBag.SpecialClass">

            <!-- Top Navbar -->
            @Html.Partial("_TopNavbar")

            <!-- Main view -->
            @RenderBody()

            <!-- Footer -->
            @Html.Partial("_Footer")

        </div>
        <!-- End page wrapper-->

        <!-- Right Sidebar -->
        @Html.Partial("_RightSidebar")

    </div>
    <!-- End wrapper-->

    <!-- Section for main scripts render -->
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @Scripts.Render("~/plugins/slimScroll")
    @Scripts.Render("~/bundles/inspinia")

    <!-- Skin config script - only for demo purpose-->
    @Scripts.Render("~/bundles/skinConfig")

    <!-- Handler for local scripts -->
    @RenderSection("scripts", required: false)
</body>
</html>

```

```

PFMO.Web
Views(Shared)
_Layout_4.cshtml

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8"/>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible"
content="IE=edge">
<title>INSPINIA | @ViewBag.Title</title>
<link
href='http://fonts.googleapis.com/css?family=Open
+Sans:400,300,600,700' rel='stylesheet'
type='text/css'>
<!-- Add local styles, mostly for plugins css file -->
@if (IsSectionDefined("Styles"))
    {@RenderSection("Styles", required: false)}

<!-- Add jQuery Style direct - used for jqGrid
plugin -->
<link
href="@Url.Content("~/Scripts/plugins/jquery-
ui/jquery-ui.css")" rel="stylesheet" type="text/css"
/>

<!-- Primary Inspinia style -->
@Styles.Render("~/font-awesome/css")
@Styles.Render("~/Content/css")
</head>
<body class="canvas-menu">

    <!-- Wrapper-->
    <!-- PageClass give you ability to specify custom
style for specific view based on action -->
    <div id="wrapper" class="@Html.PageClass()">

        <!-- Navigation -->
        @Html.Partial("_NavigationOffCanvas")

        <!-- Page wraper -->
        <div id="page-wrapper" class="gray-bg
@ ViewBag.SpecialClass">

            <!-- Top Navbar -->
            @Html.Partial("_TopNavbar")

            <!-- Main view -->
            @RenderBody()

            <!-- Footer -->

```

```

    @Html.Partial("_Footer")

</div>
<!-- End page wrapper--&gt;

<!-- Right Sidebar --&gt;
@Html.Partial("_RightSidebar")

&lt;/div&gt;
<!-- End wrapper--&gt;
<!-- Section for main scripts render --&gt;
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@Scripts.Render("~/plugins/slimScroll")
@Scripts.Render("~/bundles/inspinia")

<!-- Skin config script - only for demo purpose--&gt;
@Scripts.Render("~/bundles/skinConfig")

<!-- Handler for local scripts --&gt;
@RenderSection("scripts", required: false)

&lt;script&gt;
    $('body.canvas-menu .sidebar-collapse').slimScroll({
        height: '100%',
        railOpacity: 0.9
    });
&lt;/script&gt;

&lt;/body&gt;
&lt;/html&gt;

<b>PFMO.Web
Views(Shared)
_LayoutLandingPage.cshtml

<!DOCTYPE html>
<!--#region HTML-->
<html lang="en">

<!--#region HTML HEAD-->
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <!-- Icon -->
    <link rel="icon" href="~/Content/favicon.png">

```

```

<title>Physical Facilities Management Office</title>

<!-- Bootstrap core CSS --&gt;
&lt;link href="~/Content/LandingPage/css/bootstrap.min.css" rel="stylesheet"&gt;

<!-- Animation CSS --&gt;
&lt;link href="~/Content/LandingPage/css/animate.min.css" rel="stylesheet"&gt;

&lt;link href="~/Content/LandingPage/font-awesome/css/font-awesome.min.css" rel="stylesheet"&gt;

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries --&gt;
&lt;!--[if lt IE 9]&gt;
&lt;script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"&gt;&lt;/script&gt;
&lt;script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"&gt;&lt;/script&gt;
&lt;![endif]--&gt;
<!-- Custom styles for this template --&gt;
&lt;link href="~/Content/LandingPage/css/style.css" rel="stylesheet"&gt;
&lt;/head&gt;
&lt;!--#endregion --&gt;

&lt;!--#region HTML BODY--&gt;
&lt;body id="page-top"&gt;

    @RenderBody()

    &lt;!--#region TOP NAVBAR--&gt;
    @Html.Partial("_LandingPageTopNavBar")
    &lt;!--#endregion --&gt;

    &lt;!--#region CAROUSEL--&gt;
    @Html.Partial("_LandingPageCarousel")
    &lt;!--#endregion --&gt;

    &lt;!--#region SERVICES--&gt;
    @Html.Partial("_LandingPageServices")
    &lt;!--#endregion --&gt;

    &lt;!--#region FEATURES--&gt;
    @Html.Partial("_LandingPageFeatures")
</pre>

```

```

<!--#endregion -->

<!--#region TEAM-->
@Html.Partial("_LandingPageTeam")
<!--#endregion -->

@*<!--#region EVEN MORE FEATURES-->
@Html.Partial("_LandingPageEvenMoreFeatures")
<!--#endregion -->*@

@*<!--#region TIMELINE-->
@Html.Partial("_LandingPageTimeline")
<!--#endregion -->*@

<!--#region TESTIMONIALS-->
@Html.Partial("_LandingPageTestimonials")
<!--#endregion -->

<!--#region COMMENTS-->
@Html.Partial("_LandingPageComments")
<!--#endregion -->

<!--#region MORE AND MORE GREATER
FEATURES-->

@Html.Partial("_LandingPageMore&MoreGreaterFeatures")
<!--#endregion -->

@*<!--#region PRICING-->
@Html.Partial("_LandingPagePricing")
<!--#endregion -->*@

<!--#region CONTACT-->
@Html.Partial("_LandingPageContact")
<!--#endregion-->

<script src="~/Content/LandingPage/js/jquery-
2.1.1.js"></script>
<script
src="~/Content/LandingPage/js/pace.min.js"><scri
pt>
<script
src="~/Content/LandingPage/js/bootstrap.min.js"><
/script>

```

```

<script
src="~/Content/LandingPage/js/classie.js"></script>
<script
src="~/Content/LandingPage/js/cbpAnimatedHeade
r.js"></script>
<script
src="~/Content/LandingPage/js-wow.min.js"></scri
pt>
<script
src="~/Content/LandingPage/js-inspinia.js"></script
>

</body>
<!--#endregion -->

</html>
<!--#endregion -->
```

**PFMO.Web**  
**Views(Shared)**  
**\_LoginPartial.cshtml**

```

@using Microsoft.AspNet.Identity
@if (Request.IsAuthenticated)
{
    <ul class="nav navbar-top-links navbar-right">
        <li><a target="_blank" class="btn btn-primary"
href="http://www.outlook.com/plm.edu.ph"><i
class="fa fa-envelope-o"></i>Outlook</a></li>
    </ul>

    using (Html.BeginForm("LogOff", "Account",
FormMethod.Post, new { id = "logoutForm", @class
= "navbar-right" }))
    {
        @Html.AntiForgeryToken()

        <ul class="nav navbar-top-links navbar-right">
            <li>
                @Html.ActionLink("Hello " +
(ViewData["FullName"]) + "!", "Index", "Manage",
routeValues: null, htmlAttributes: new { title =
"Manage" })
            </li>
            <li><a
href="javascript:document.getElementById('logoutF
orm').submit()"><i class="fa fa-sign-
out"></i>Logout</a></li>
        </ul>
    }
    else
    {
        <ul class="nav navbar-top-links navbar-right">
```

```

<li>@Html.ActionLink("Register", "Register",
"Account", routeValues: null, htmlAttributes: new {
id = "registerLink" })</li>
<li>@Html.ActionLink("Log in", "Login",
"Account", routeValues: null, htmlAttributes: new {
id = "loginLink" })</li>
</ul>
}

PFMO.Web
Views(Shared)
_Navigation

@using Microsoft.AspNet.Identity

<nav class="navbar-default navbar-static-side"
role="navigation">
<div class="sidebar-collapse">
<ul class="nav metismenu" id="side-menu">
<li class="nav-header">
<div class="dropdown profile-element">
<span>

</span>
<a data-toggle="dropdown"
class="dropdown-toggle" href="#"
bold">@(ViewData["FullName"])</strong>
</span> <span class="text-muted
text-xs block">Dashboard<b
class="caret"></b></span>
</span>
</a>
<ul class="dropdown-menu animated
fadeInRight m-t-xs">
<li><a href="@Url.Action("Index",
"Desktop")">Desktop</a></li>
<li class="divider"></li>
<li><a href="@Url.Action("Index",
"Pinboards")">Pinboard</a></li>
<li class="divider"></li>
<li><a href="@Url.Action("Index",
"Contacts")">Contacts</a></li>
<li class="divider"></li>
<li><a href="@Url.Action("Index",
"Calendar")">Calendar</a></li>
</ul>
</div>
<div class="logo-element">
PM
</div>
</li>

```

```

@*HUMAN RESOURCES MANAGEMENT*@ 
<li>
<a href="#"><i class="fa fa-sitemap"></i>
<span class="nav-label">Human Resources
</span><span class="fa arrow"></span></a>
<ul class="nav nav-second-level collapse">
<li>
<a href="@Url.Action("Index",
"Sections")">Manage Sections</a>
</li>
<li>
<a href="@Url.Action("Index",
"SectionHeads")">Manage Heads</a>
</li>
<li>
<a href="@Url.Action("Index",
"Personnels")">Manage Personnels</a>
</li>
<li>
<a href="#">Manage Designations
<span class="fa arrow"></span></a>
<ul class="nav nav-third-level">
<li>
<a href="@Url.Action("Index",
"SectionHeadDesignations")">Section Head</a>
</li>
<li>
<a href="@Url.Action("Index",
"PersonnelDesignations")">Personnel</a>
</li>
</ul>
</li>
</ul>
</li>

```

```

@*INVENTORY MANAGEMENT*@ 
<li>
<a href="#"><i class="fa fa-shopping-
cart"></i> <span class="nav-label">Inventory
</span><span class="fa arrow"></span></a>
<ul class="nav nav-second-level collapse">
<li>
<a href="#">Manage Categories <span
class="fa arrow"></span></a>
<ul class="nav nav-third-level">
<li>

```

```

                <a href="@Url.Action("Index",
"EquipmentCategories")">Equipment
Categories</a>
            </li>
            <li>
                <a href="@Url.Action("Index",
"SupplyCategories")">Supply Categories</a>
            </li>
            </ul>
        </li>
        <li>
            <a href="@Url.Action("Index",
"Equipments")">Manage Equipments</a>
        </li>
        <li>
            <a href="@Url.Action("Index",
"Supplies")">Manage Supplies</a>
        </li>
        </ul>
    </li>

    @*PROJECT MANAGEMENT*@
    <li>
        <a href="#"><i class="fa fa-pie-chart"></i>
<span class="nav-label">Projects </span><span
class="fa arrow"></span></a>
        <ul class="nav nav-second-level collapse">
            <li>
                <a href="@Url.Action("Index",
"ProjectTeams")">Manage Teams</a>
            </li>
            <li>
                <a href="@Url.Action("Index",
"Projects")">Manage Projects</a>
            </li>
            @*<li>
                <a href="#">Manage Transactions
<span class="fa arrow"></span></a>
                <ul class="nav nav-third-level">
                    <li>
                        <a href="">Issue Supply</a>
                    </li>
                    <li>
                        <a href="#">Lend Supply</a>
                    </li>
                </ul>
            </li>*@
            </ul>
        </li>

        @*TASKS MANAGEMENT*@
        <li>
            <a href="#"><i class="fa fa-bell"></i> <span
class="nav-label">Task Assignments </span><span
class="fa arrow"></span></a>
            <ul class="nav nav-second-level collapse">
                <li>
                    <a href="#">Section Tasks</a>
                </li>
                <li>
                    <a href="#">Personnel Tasks</a>
                </li>
            </ul>
        </li>

        @*RECORDS MANAGEMENT*@
        <li>
            <a href="#"><i class="fa fa-folder-
open"></i> <span class="nav-label">Records
</span><span class="fa arrow"></span></a>
            <ul class="nav nav-second-level collapse">
                <li>
                    <a href="#">Manage Reports</a>
                </li>
                <li>
                    <a href="#">Manage Files</a>
                </li>
            </ul>
        </li>

        @*USER ACCOUNTS*@
        <li>
            <a href="#"><i class="fa fa-users"></i>
<span class="nav-label">Accounts </span><span
class="fa arrow"></span></a>
            <ul class="nav nav-second-level collapse">
                <li>
                    <a href="#">Manage User Roles</a>
                </li>
                @*<li>
                    <a href="#">Manage Roles</a>
                </li>*@
                </ul>
        </li>

        @*SUPPORT*@
        <li class="special_link">
            <a target="_blank"
href="@Url.Action("Index", "Home")"><i class="fa
fa-space-shuttle"></i> <span class="nav-label">Take
a Tour!</span> <span class="label label-warning
pull-right">START</span></a>
        </li>
        <li class="special_link">
            <a target="_blank" href="#"><i class="fa
fa-pencil-square-o"></i> <span class="nav-
label">Alerts</span></a>
        </li>
    
```

```

<li>
    <a href="#"><i class="fa fa-file-text"></i>
<span class="nav-label">Manual</span></a>
</li>
<li>
    <a target="_blank" href="#"><i class="fa fa-star"></i> <span class="nav-label">Developers</span></a>
</li>
<li>
    <a href="#"><i class="fa fa-question-circle"></i> <span class="nav-label">About</span></a>
</li>
</ul>
</div>
</nav>

PFMO.Web
Views(Shared)
_NavigationOffCanvas.cshtml

<nav class="navbar-default navbar-static-side" role="navigation">
    <div class="sidebar-collapse">
        <a class="close-canvas-menu"><i class="fa fa-times"></i></a>
        <ul class="nav" id="side-menu">
            <li class="nav-header">
                <div class="dropdown profile-element">
                    <span>
                        
                    </span>
                    <a data-toggle="dropdown" href="#">
                        <span class="clear">
                            <span class="block m-t-xs">
                                <strong class="font-bold">David Williams</strong>
                            </span> <span class="text-muted text-xs block">Art Director <b>class="caret"</b></span>
                            </span>
                        </span>
                    </a>
                    <ul class="dropdown-menu animated fadeInRight m-t-xs">
                        <li><a href="@Url.Action("Profile", "AppViews")">Profile</a></li>
                        <li><a href="@Url.Action("Contacts", "AppViews")">Contacts</a></li>
                        <li><a href="@Url.Action("Inbox", "Mailbox")">Mailbox</a></li>
                        <li class="divider"></li>
                        <li><a href="@Url.Action("Logout", "Pages")">Logout</a></li>
                    </ul>
                </div>
            </li>
            <li class="@Html.IsSelected(controller: "Dashboards")">
                <a href="@Url.Action("Dashboard_1", "Dashboards")"><i class="fa fa-th-large"></i> <span class="nav-label">Dashboards</span> <span class="fa arrow"></span></a>
                <ul class="nav nav-second-level">
                    <li class="@Html.IsSelected(action: "Dashboard_1")"><a href="@Url.Action("Dashboard_1", "Dashboards")">Dashboard v.1</a></li>
                    <li class="@Html.IsSelected(action: "Dashboard_2")"><a href="@Url.Action("Dashboard_2", "Dashboards")">Dashboard v.2</a></li>
                    <li class="@Html.IsSelected(action: "Dashboard_3")"><a href="@Url.Action("Dashboard_3", "Dashboards")">Dashboard v.3</a></li>
                    <li class="@Html.IsSelected(action: "Dashboard_4_1")"><a href="@Url.Action("Dashboard_4_1", "Dashboards")">Dashboard v.4</a></li>
                </ul>
            </li>
            <li class="@Html.IsSelected(controller: "Layouts")">
                <a href="@Url.Action("Index", "Layouts")"><i class="fa fa-diamond"></i> <span class="nav-label">Layouts</span> <span class="label label-primary pull-right">NEW</span></a>
            </li>
            <li class="@Html.IsSelected(controller: "Graphs")">
                <a href="#"><i class="fa fa-bar-chart-o"></i> <span class="nav-label">Graphs</span> <span class="fa arrow"></span></a>
                <ul class="nav nav-second-level">
                    <li class="@Html.IsSelected(action: "Flot")"><a href="@Url.Action("Flot", "Graphs")">Flot Charts</a></li>
                    <li class="@Html.IsSelected(action: "Morris")"><a href="@Url.Action("Morris", "Graphs")">Morris.js Charts</a></li>
                </ul>
            </li>
        </ul>
    </div>
</nav>

```

```

<li class="@Html.IsSelected(action:
"Rickshaw")"><a href="@Url.Action("Rickshaw",
"Graphs")">Rickshaw Charts</a></li>
    <li class="@Html.IsSelected(action:
"Chartjs")"><a href="@Url.Action("Chartjs",
"Graphs")">Chart.js</a></li>
        <li class="@Html.IsSelected(action:
"Peity")"><a href="@Url.Action("Peity",
"Graphs")">Peity Charts</a></li>
            <li class="@Html.IsSelected(action:
"Sparkline")"><a href="@Url.Action("Sparkline",
"Graphs")">Sparkline Charts</a></li>
                </ul>
            </li>
        <li class="@Html.IsSelected(controller:
"Mailbox")">
            <a href="#"><i class="fa fa-envelope"></i>
            <span class="nav-label">Mailbox </span><span
            class="label label-warning pull-
            right">16/24</span></a>
            <ul class="nav nav-second-level">
                <li class="@Html.IsSelected(action:
"Inbox")"><a href="@Url.Action("Inbox",
"Mailbox")">Inbox</a></li>
                    <li class="@Html.IsSelected(action:
"EmailView")"><a href="@Url.Action("EmailView",
"Mailbox")">Email view</a></li>
                        <li class="@Html.IsSelected(action:
"ComposeEmail")"><a
href="@Url.Action("ComposeEmail",
"Mailbox")">Compose email</a></li>
                            <li class="@Html.IsSelected(action:
"EmailTemplates")"><a
href="@Url.Action("EmailTemplates",
"Mailbox")">Email templates</a></li>
                                </ul>
                            </li>
                        <li class="@Html.IsSelected(controller:
"Widgets")">
                            <a href="@Url.Action("Index",
"Widgets")"><i class="fa fa-flask"></i> <span
                            class="nav-label">Widgets</span> </a>
                        </li>
                    <li class="@Html.IsSelected(controller:
"Forms")">
                        <a href="#"><i class="fa fa-edit"></i>
                        <span class="nav-label">Forms</span><span
                        class="fa arrow"></span></a>
                        <ul class="nav nav-second-level">
                            <li class="@Html.IsSelected(action:
"BasicFroms")"><a href="@Url.Action("BasicFroms",
"Forms")">Basic form</a></li>
                                <li class="@Html.IsSelected(action:
"Advanced")"><a href="@Url.Action("Advanced",
"Forms")">Advanced Plugins</a></li>

```

```

<li class="@Html.IsSelected(action:
"Wizard")"><a href="@Url.Action("Wizard",
"Forms")">Wizard</a></li>
    <li class="@Html.IsSelected(action:
"FileUpload")"><a href="@Url.Action("FileUpload",
"Forms")">File Upload</a></li>
        <li class="@Html.IsSelected(action:
"TextEditor")"><a href="@Url.Action("TextEditor",
"Forms")">Text Editor</a></li>
            </ul>
        </li>
    <li class="@Html.IsSelected(controller:
"AppViews")">
        <a href="#"><i class="fa fa-desktop"></i>
        <span class="nav-label">App Views</span> <span
        class="pull-right label label-
        primary">SPECIAL</span></a>
        <ul class="nav nav-second-level">
            <li class="@Html.IsSelected(action:
"Contacts")"><a href="@Url.Action("Contacts",
"AppViews")">Contacts</a></li>
                <li class="@Html.IsSelected(action:
"Profile")"><a href="@Url.Action("Profile",
"AppViews")">Profile</a></li>
                    <li class="@Html.IsSelected(action:
"Projects")"><a href="@Url.Action("Projects",
"AppViews")">Projects</a></li>
                        <li class="@Html.IsSelected(action:
"ProjectDetail")"><a
href="@Url.Action("ProjectDetail",
"AppViews")">Project detail</a></li>
                            <li class="@Html.IsSelected(action:
"TeamsBoard")"><a
href="@Url.Action("TeamsBoard",
"AppViews")">Teams board</a></li>
                        <li class="@Html.IsSelected(action:
"Clients")"><a href="@Url.Action("Clients",
"AppViews")">Clients</a></li>
                            <li class="@Html.IsSelected(action:
"OutlookView")"><a
href="@Url.Action("OutlookView",
"AppViews")">Outlook view</a></li>
                                <li class="@Html.IsSelected(action:
"FileManager")"><a
href="@Url.Action("FileManager",
"AppViews")">File manager</a></li>
                                    <li class="@Html.IsSelected(action:
"Calendar")"><a href="@Url.Action("Calendar",
"AppViews")">Calendar</a></li>
                                        <li class="@Html.IsSelected(action:
"IssueTracker")"><a
href="@Url.Action("IssueTracker",
"AppViews")">Issue tracker</a></li>

```

```

<li class="@Html.IsSelected(action:
"Blog")"><a href="@Url.Action("Blog",
"AppViews")">Blog</a></li>
    <li class="@Html.IsSelected(action:
"Article")"><a href="@Url.Action("Article",
"AppViews")">Article</a></li>
        <li class="@Html.IsSelected(action:
"FAQ")"><a href="@Url.Action("FAQ",
"AppViews")">FAQ</a></li>
            <li class="@Html.IsSelected(action:
"Timeline")"><a href="@Url.Action("Timeline",
"AppViews")">Timeline</a></li>
                <li class="@Html.IsSelected(action:
"PinBoard")"><a href="@Url.Action("PinBoard",
"AppViews")">Pin board</a></li>
                    </ul>
                </li>
            <li class="@Html.IsSelected(controller:
"Pages")">
                <a href="#"><i class="fa fa-files-o"></i>
<span class="nav-label">Other Pages</span><span
class="fa arrow"></span></a>
                <ul class="nav nav-second-level">
                    <li class="@Html.IsSelected(action:
"SearchResults")"><a href="@Url.Action("SearchResults",
"Pages")">Search results</a></li>
                        <li class="@Html.IsSelected(action:
"LockScreen")"><a href="@Url.Action("LockScreen",
"Pages")">Lockscreen</a></li>
                            <li class="@Html.IsSelected(action:
"Invoice")"><a href="@Url.Action("Invoice",
"Pages")">Invoice</a></li>
                                <li class="@Html.IsSelected(action:
"Login")"><a href="@Url.Action("Login",
"Pages")">Login</a></li>
                                    <li class="@Html.IsSelected(action:
"Login_2")"><a href="@Url.Action("Login_2",
"Pages")">Login v.2</a></li>
                                        <li class="@Html.IsSelected(action:
"ForgotPassword")"><a href="@Url.Action("ForgotPassword",
"Pages")">Forgot password</a></li>
                                            <li class="@Html.IsSelected(action:
"Register")"><a href="@Url.Action("Register",
"Pages")">Register</a></li>
                                                <li class="@Html.IsSelected(action:
"NotNotFoundError")"><a href="@Url.Action("NotFoundError",
"Pages")">404 Page</a></li>
                                                    <li class="@Html.IsSelected(action:
"InternalServerError")"><a href="@Url.Action("InternalServerError",
"Pages")">500 Page</a></li>

```

```

<li class="@Html.IsSelected(action:
"EmptyPage")"><a href="@Url.Action("EmptyPage",
"Pages")">Empty page</a></li>
</ul>
</li>
<li class="@Html.IsSelected(controller:
"Miscellaneous")">
    <a href="#"><i class="fa fa-globe"></i>
<span class="nav-label">Miscellaneous</span><span class="label
label-info pull-right">NEW</span></a>
    <ul class="nav nav-second-level">
        <li class="@Html.IsSelected(action:
"Notification")"><a href="@Url.Action("Notification",
"Miscellaneous")">Notification</a></li>
            <li class="@Html.IsSelected(action:
"NestedList")"><a href="@Url.Action("NestedList",
"Miscellaneous")">Nested list</a></li>
                <li class="@Html.IsSelected(action:
"AgileBoard")"><a href="@Url.Action("AgileBoard",
"Miscellaneous")">Agile board</a></li>
                    <li class="@Html.IsSelected(action:
"Timeline_second_version")"><a href="@Url.Action("Timeline_second_version",
"Miscellaneous")">Timeline v.2</a></li>
                        <li class="@Html.IsSelected(action:
"Diff")"><a href="@Url.Action("Diff",
"Miscellaneous")">Diff</a></li>
                            <li class="@Html.IsSelected(action:
"IdleTimer")"><a href="@Url.Action("IdleTimer",
"Miscellaneous")">Idle timer</a></li>
                                <li class="@Html.IsSelected(action:
"Spinners")"><a href="@Url.Action("Spinners",
"Miscellaneous")">Spinners</a></li>
                                    <li class="@Html.IsSelected(action:
"LiveFavicon")"><a href="@Url.Action("LiveFavicon",
"Miscellaneous")">Live favicon</a></li>
                                        <li class="@Html.IsSelected(action:
"Forum_view")"><a href="@Url.Action("Forum_view",
"Miscellaneous")">Forum view</a></li>
                                            <li class="@Html.IsSelected(action:
"GoogleMaps")"><a href="@Url.Action("GoogleMaps",
"Miscellaneous")">Google maps</a></li>
                                                <li class="@Html.IsSelected(action:
"CodeEditor")"><a href="@Url.Action("CodeEditor",
"Miscellaneous")">Code editor</a></li>
                                                    <li class="@Html.IsSelected(action:
"ModalWindow")"><a href="@Url.Action("ModalWindow",
"Miscellaneous")">Modal window</a></li>

```

```

<li class="@Html.IsSelected(action:
"Validation")"><a href="@Url.Action("Validation",
"Miscellaneous")">Validation</a></li>
    <li class="@Html.IsSelected(action:
"Tree_view")"><a href="@Url.Action("Tree_view",
"Miscellaneous")">Tree view</a></li>
        <li class="@Html.IsSelected(action:
"Chat_view")"><a href="@Url.Action("Chat_view",
"Miscellaneous")">Chat view</a></li>
            </ul>
        </li>
    <li class="@Html.IsSelected(controller:
"UIElements")">
        <a href="#"><i class="fa fa-flask"></i>
        <span class="nav-label">UI Elements</span><span
        class="fa arrow"></span></a>
        <ul class="nav nav-second-level">
            <li class="@Html.IsSelected(action:
"Typography")"><a
            href="@Url.Action("Typography",
"UIElements")">Typography</a></li>
                <li class="@Html.IsSelected(action:
"Icons")"><a href="@Url.Action("Icons",
"UIElements")">Icons</a></li>
                    <li class="@Html.IsSelected(action:
"DraggablePanels")"><a
                    href="@Url.Action("DraggablePanels",
"UIElements")">Draggable Panels</a></li>
                        <li class="@Html.IsSelected(action:
"Buttons")"><a href="@Url.Action("Buttons",
"UIElements")">Buttons</a></li>
                            <li class="@Html.IsSelected(action:
"Video")"><a href="@Url.Action("Video",
"UIElements")">Video</a></li>
                                <li class="@Html.IsSelected(action:
"TablesPanels")"><a
                                href="@Url.Action("TablesPanels",
"UIElements")">Tabs & Panels</a></li>
                                    <li class="@Html.IsSelected(action:
"NotificationsToolips")"><a
                                    href="@Url.Action("NotificationsToolips",
"UIElements")">Notifications & Tooltips</a></li>
                                        <li class="@Html.IsSelected(action:
"BadgesLabelsProgress")"><a
                                        href="@Url.Action("BadgesLabelsProgress",
"UIElements")">Badges, Labels, Progress</a></li>
                                            </ul>
                                        </li>
                                    <li class="@Html.IsSelected(controller:
"GridOptions")">
                                        <a href="@Url.Action("Index",
"GridOptions")"><i class="fa fa-laptop"></i> <span
                                        class="nav-label">Grid options</span></a>
                                    </li>

```

```

<li class="@Html.IsSelected(controller:
"Tables")">
    <a href="#"><i class="fa fa-table"></i>
    <span class="nav-label">Tables</span><span
    class="fa arrow"></span></a>
    <ul class="nav nav-second-level">
        <li class="@Html.IsSelected(action:
"StaticTables")"><a
        href="@Url.Action("StaticTables", "Tables")">Static
        Tables</a></li>
            <li class="@Html.IsSelected(action:
"DataTables")"><a href="@Url.Action("DataTables",
"Tables")">Data Tables</a></li>
                <li class="@Html.IsSelected(action:
"jqGrid")"><a href="@Url.Action("jqGrid",
"Tables")">jqGrid</a></li>
                    </ul>
                </li>
            <li class="@Html.IsSelected(controller:
"Gallery")">
                <a href="#"><i class="fa fa-picture-o"></i>
                <span class="nav-label">Gallery</span><span
                class="fa arrow"></span></a>
                <ul class="nav nav-second-level">
                    <li class="@Html.IsSelected(action:
"BasicGallery")"><a
                    href="@Url.Action("BasicGallery",
"Gallery")">Lightbox Gallery</a></li>
                        <li class="@Html.IsSelected(action:
"BootstrapCarusela")"><a
                        href="@Url.Action("BootstrapCarusela",
"Gallery")">Bootstrap Carusela</a></li>

```

```

                </ul>
            </li>
        <li class="@Html.IsSelected(controller:
"Menu Levels")">
            <a href="#"><i class="fa fa-sitemap"></i>
            <span class="nav-label">Menu Levels </span><span
            class="fa arrow"></span></a>
            <ul class="nav nav-second-level">
                <li>
                    <a href="#">Third Level <span
                    class="fa arrow"></span></a>
                    <ul class="nav nav-third-level">
                        <li>
                            <a href="#">Third Level Item</a>
                        </li>
                        <li>
                            <a href="#">Third Level Item</a>
                        </li>
                        <li>
                            <a href="#">Third Level Item</a>
                        </li>

```

```

                    </ul>
                </li>
            </li>

```

```

</li>
<li><a href="#">Second Level
Item</a></li>
<li>
    <a href="#">Second Level Item</a>
</li>
<li>
    <a href="#">Second Level Item</a>
</li>
</ul>
</li>
<li class="@Html.IsSelected(controller:
"CSSAnimations")">
    <a href="@Url.Action("Index",
"CSSAnimations")"><i class="fa fa-magic"></i>
<span class="nav-label">CSS Animations
</span><span class="label label-info pull-
right">62</span></a>
</li>
<li class="special_link
@Html.IsSelected(controller: " Scaffolding")">
    <a href="@Url.Action("Index",
"Scaffolding")"><i class="fa fa-database"></i> <span
class="nav-label">Scaffolding Example</span></a>
</li>
<li class="special_link
@Html.IsSelected(controller: " Json")">
    <a href="@Url.Action("Index", "Json")"><i
class="fa fa-plug"></i> <span class="nav-
label">JSON Examples</span></a>
</li>
</ul>
</div>
</nav>

```

**PFMO.Web**  
**Views(Shared)**  
**\_PageHeader.cshtml**

```

<div class="row wrapper border-bottom white-bg
page-heading">
    <div class="col-lg-10">
        <h2>@ViewBag.Title</h2>
        <ol class="breadcrumb">
            <li>
                <a href="#">@ViewBag.Message1</a>
            </li>
            <li>
                <a href="#">@ViewBag.Message2</a>
            </li>
            <li>
                <a href="#">@ViewBag.Message3</a>
            </li>
            <li>

```

```

<strong>@ViewBag.Message4</strong>
</li>
</ol>
</div>
<div class="col-lg-2">
</div>
</div>

```

PFMO.Web  
Views(Shared)  
\_RightSidebar.cshtml

```

<div id="right-sidebar">
    <div class="sidebar-container">
        <ul class="nav nav-tabs navs-3">
            <li class="active">
                <a data-toggle="tab" href="#tab-1">
                    Notes
                </a>
            </li>
            <li>
                <a data-toggle="tab" href="#tab-2">
                    Projects
                </a>
            </li>
            <li class="">
                <a data-toggle="tab" href="#tab-3">
                    <i class="fa fa-gear"></i>
                </a>
            </li>
        </ul>
        <div class="tab-content">
            <div id="tab-1" class="tab-pane active">
                <div class="sidebar-title">
                    <h3><i class="fa fa-comments-o"></i>
                    Latest Notes</h3>
                    <small><i class="fa fa-tim"></i> You have
                    10 new message.</small>
                </div>
                <div>
                    <div class="sidebar-message">
                        <a href="#">
                            <div class="pull-left text-center">
                                

```

```

<div class="m-t-xs">
    <i class="fa fa-star text-
warning"></i>
    <i class="fa fa-star text-
warning"></i>
    </div>
</div>
<div class="media-body">

    There are many variations of
    passages of Lorem Ipsum available.
    <br>
    <small class="text-muted">Today
4:21 pm</small>
    </div>
</a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">
        The point of using Lorem Ipsum is
        that it has a more-or-less normal.
        <br>
        <small class="text-
muted">Yesterday 2:45 pm</small>
        </div>
    </a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        <div class="m-t-xs">
            <i class="fa fa-star text-
warning"></i>
            <i class="fa fa-star text-
warning"></i>
            <i class="fa fa-star text-
warning"></i>
            </div>
        </div>
    <div class="media-body">
        Evolved over the years,
        sometimes by accident, sometimes on purpose
        (injected humour and the like).
        <br>
        <small class="text-
muted">Yesterday 1:10 pm</small>
        </div>
    </a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">
        Lorem Ipsum, you need to be sure
        there isn't anything embarrassing hidden in the
        <br>
        <small class="text-
muted">Monday 8:37 pm</small>
        </div>
    </a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">
        All the Lorem Ipsum generators on
        the Internet tend to repeat.
        <br>
        <small class="text-muted">Today
4:21 pm</small>
        </div>
    </a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">
        Renaissance. The first line of
        Lorem Ipsum, "Lorem ipsum dolor sit amet..",
        comes from a line in section 1.10.32.
        <br>
        <small class="text-
muted">Yesterday 2:45 pm</small>
        </div>
    </a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">

```

```
<div class="m-t-xs">
    <i class="fa fa-star text-
warning"></i>
    <i class="fa fa-star text-
warning"></i>
    <i class="fa fa-star text-
warning"></i>
        </div>
    </div>
<div class="media-body">
    The standard chunk of Lorem
    Ipsum used since the 1500s is reproduced below.
    <br>
    <small class="text-
muted">Yesterday 1:10 pm</small>
    </div>
</a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">
        Uncover many web sites still in
        their infancy. Various versions have.
        <br>
        <small class="text-
muted">Monday 8:37 pm</small>
        </div>
    </a>
    </div>
</div>

</div>

<div id="tab-2" class="tab-pane">

    <div class="sidebar-title">
        <h3><i class="fa fa-cube"></i> Latest
        projects</h3>
        <small><i class="fa fa-tim"></i> You have
        14 projects. 10 not completed.</small>
    </div>

    <ul class="sidebar-list">
        <li>
            <a href="#">
                <div class="small pull-right m-t-xs">9
hours ago</div>
                <h4>Business valuation</h4>
                It is a long established fact that a
                reader will be distracted.
            </a>
        </li>
    </ul>
</div>

    <div class="small pull-right m-t-xs">22%
hours ago</div>
    <h4>Contract with Company</h4>
    Many desktop publishing packages
    and web page editors.

    <div class="small pull-right m-t-xs">9
hours ago</div>
    <h4>Meeting</h4>
    By the readable content of a page
    when looking at its layout.

    <div class="small pull-right m-t-xs">9
hours ago</div>
    <h4>The generated</h4>
    <!--<div class="small pull-right m-t-
xs">9 hours ago</div>-->
    There are many variations of
    passages of Lorem Ipsum available.
    <div class="small">Completion with:
    22%</div>
</div>
```

```

<div class="small text-muted m-t-xs">Project end: 4:00 pm - 12.06.2014</div>
</a>
</li>
<li>
<a href="#">
<div class="small pull-right m-t-xs">9 hours ago</div>
<h4>Business valuation</h4>
It is a long established fact that a reader will be distracted.

<div class="small">Completion with: 22%</div>
<div class="progress progress-mini">
<div style="width: 22%;" class="progress-bar progress-bar-warning"></div>
</div>
<div class="small text-muted m-t-xs">Project end: 4:00 pm - 12.06.2014</div>
</a>
</li>
<li>
<a href="#">
<div class="small pull-right m-t-xs">9 hours ago</div>
<h4>Contract with Company </h4>
Many desktop publishing packages and web page editors.

<div class="small">Completion with: 48%</div>
<div class="progress progress-mini">
<div style="width: 48%;" class="progress-bar"></div>
</div>
</a>
</li>
<li>
<a href="#">
<div class="small pull-right m-t-xs">9 hours ago</div>
<h4>Meeting</h4>
By the readable content of a page when looking at its layout.

<div class="small">Completion with: 14%</div>
<div class="progress progress-mini">
<div style="width: 14%;" class="progress-bar progress-bar-info"></div>
</div>
</a>
</li>
<li>

```

```

<a href="#">
<span class="label label-primary pull-right">NEW</span>
<h4>The generated</h4>
<!--<div class="small pull-right m-t-xs">9 hours ago</div>-->
There are many variations of passages of Lorem Ipsum available.
<div class="small">Completion with: 22%</div>
<div class="small text-muted m-t-xs">Project end: 4:00 pm - 12.06.2014</div>
</a>
</li>
</ul>
</div>
<div id="tab-3" class="tab-pane">
<div class="sidebar-title">
<h3><i class="fa fa-gears"></i> Settings</h3>
<small><i class="fa fa-tim"></i> You have 14 projects. 10 not completed.</small>
</div>
<div class="setings-item">
<span>
Show notifications
</span>
<div class="switch">
<div class="onoffswitch">
<input type="checkbox" name="collapsemenu" class="onoffswitch-checkbox" id="example" for="example">
<label class="onoffswitch-label" for="example">
<span class="onoffswitch-inner"></span>
<span class="onoffswitch-switch"></span>
</label>
</div>
</div>
<div class="setings-item">
<span>
Disable Chat
</span>
<div class="switch">
<div class="onoffswitch">

```

```

        <input type="checkbox"
name="collapsemenu" checked class="onoffswitch-
checkbox" id="example2">
        <label class="onoffswitch-label"
for="example2">
            <span class="onoffswitch-
inner"></span>
            <span class="onoffswitch-
switch"></span>
        </label>
    </div>
</div>
<div class="setings-item">
    <span>
        Enable history
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox"
name="collapsemenu" class="onoffswitch-
checkbox" id="example3">
            <label class="onoffswitch-label"
for="example3">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="setings-item">
    <span>
        Show charts
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox"
name="collapsemenu" class="onoffswitch-
checkbox" id="example4">
            <label class="onoffswitch-label"
for="example4">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="setings-item">
    <span>
        Offline users
    </span>

```

```

        <div class="switch">
            <div class="onoffswitch">
                <input type="checkbox" checked
name="collapsemenu" class="onoffswitch-
checkbox" id="example5">
                <label class="onoffswitch-label"
for="example5">
                    <span class="onoffswitch-
inner"></span>
                    <span class="onoffswitch-
switch"></span>
                </label>
            </div>
        </div>
</div>
<div class="setings-item">
    <span>
        Global search
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox" checked
name="collapsemenu" class="onoffswitch-
checkbox" id="example6">
            <label class="onoffswitch-label"
for="example6">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="setings-item">
    <span>
        Update everyday
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox"
name="collapsemenu" class="onoffswitch-
checkbox" id="example7">
            <label class="onoffswitch-label"
for="example7">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="sidebar-content">

```

```

<h4>Settings</h4>
<div class="small">
    I believe that. Lorem Ipsum is simply
    dummy text of the printing and typesetting
    industry.

    And typesetting industry. Lorem Ipsum
    has been the industry's standard dummy text ever
    since the 1500s.

    Over the years, sometimes by
    accident, sometimes on purpose (injected humour
    and the like).
</div>
</div>

</div>
</div>

</div>

</div>

```

PFMO.Web  
Views(Shared)  
\_RightSidebar.cshtml

```

<div id="right-sidebar">
    <div class="sidebar-container">

        <ul class="nav nav-tabs navs-3">

            <li class="active">
                <a data-toggle="tab" href="#tab-1">
                    Notes
                </a>
            </li>
            <li>
                <a data-toggle="tab" href="#tab-2">
                    Projects
                </a>
            </li>
            <li class="">
                <a data-toggle="tab" href="#tab-3">
                    <i class="fa fa-gear"></i>
                </a>
            </li>
        </ul>

        <div class="tab-content">

            <div id="tab-1" class="tab-pane active">

```

```

<div class="sidebar-title">
    <h3><i class="fa fa-comments-o"></i>
    Latest Notes</h3>
    <small><i class="fa fa-tim"></i> You have
    10 new message.</small>
</div>

<div class="media-body">
    There are many variations of
    passages of Lorem Ipsum available.
    <br>
    <small class="text-muted">Today
    4:21 pm</small>
</div>
</a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            

```

The point of using Lorem Ipsum is  
that it has a more-or-less normal.  
<br>  
<small class="text-  
muted">Yesterday 2:45 pm</small>

```

</div>
</a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            

```

```

</div>
</div>
<div class="media-body">
    The point of using Lorem Ipsum is
    that it has a more-or-less normal.  
<br>  
<small class="text-  
muted">Yesterday 2:45 pm</small>
</div>
</a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            

```

```

</div>

```

```

        <i class="fa fa-star text-
warning"></i>
        <i class="fa fa-star text-
warning"></i>
        <i class="fa fa-star text-
warning"></i>
    </div>
</div>
<div class="media-body">
    Mevolved over the years,
    sometimes by accident, sometimes on purpose
    (injected humour and the like).
    <br>
    <small class="text-
muted">Yesterday 1:10 pm</small>
</div>
</a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">
        Lorem Ipsum, you need to be sure
        there isn't anything embarrassing hidden in the
        <br>
        <small class="text-
muted">Monday 8:37 pm</small>
    </div>
    </a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">
        All the Lorem Ipsum generators on
        the Internet tend to repeat.
        <br>
        <small class="text-muted">Today
4:21 pm</small>
    </div>
    </a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    </a>
</div>
<div class="media-body">
    Renaissance. The first line of
    Lorem Ipsum, "Lorem ipsum dolor sit amet..",
    comes from a line in section 1.10.32.
    <br>
    <small class="text-
muted">Yesterday 2:45 pm</small>
</div>
</a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="m-t-xs">
        <i class="fa fa-star text-
warning"></i>
        <i class="fa fa-star text-
warning"></i>
        <i class="fa fa-star text-
warning"></i>
    </div>
</div>
<div class="media-body">
    The standard chunk of Lorem
    Ipsum used since the 1500s is reproduced below.
    <br>
    <small class="text-
muted">Yesterday 1:10 pm</small>
    </div>
    </a>
</div>
<div class="sidebar-message">
    <a href="#">
        <div class="pull-left text-center">
            
        </div>
    <div class="media-body">
        Uncover many web sites still in
        their infancy. Various versions have.
        <br>
        <small class="text-
muted">Monday 8:37 pm</small>
    </div>
    </a>
</div>
</div>
</div>
<div id="tab-2" class="tab-pane">

```

```

<div class="sidebar-title">
    <h3><i class="fa fa-cube"></i> Latest
    projects</h3>
    <small><i class="fa fa-tim"></i> You have
    14 projects. 10 not completed.</small>
</div>

<ul class="sidebar-list">
    <li>
        <a href="#">
            <div class="small pull-right m-t-xs">9
            hours ago</div>
            <h4>Business valuation</h4>
            It is a long established fact that a
            reader will be distracted.

            <div class="small">Completion with:
            22%</div>
            <div class="progress progress-mini">
                <div style="width: 22%;">
                    class="progress-bar progress-bar-warning"></div>
            </div>
            <div class="small text-muted m-t-
            xs">Project end: 4:00 pm - 12.06.2014</div>
        </a>
    </li>
    <li>
        <a href="#">
            <div class="small pull-right m-t-xs">9
            hours ago</div>
            <h4>Contract with Company </h4>
            Many desktop publishing packages
            and web page editors.

            <div class="small">Completion with:
            48%</div>
            <div class="progress progress-mini">
                <div style="width: 48%;">
                    class="progress-bar"></div>
            </div>
        </a>
    </li>
    <li>
        <a href="#">
            <div class="small pull-right m-t-xs">9
            hours ago</div>
            <h4>Meeting</h4>
            By the readable content of a page
            when looking at its layout.

            <div class="small">Completion with:
            14%</div>
            <div class="progress progress-mini">

```

class="progress-bar progress-bar-info"></div>

</div>

</a>

</li>

<li>

<a href="#">

<span class="label label-primary
 pull-right">NEW</span>

<h4>The generated</h4>

<!--<div class="small pull-right m-t-
 xs">9 hours ago</div-->

There are many variations of
 passages of Lorem Ipsum available.

<div class="small">Completion with:
 22%</div>

<div class="small text-muted m-t-
 xs">Project end: 4:00 pm - 12.06.2014</div>

</a>

</li>

<li>

<a href="#">

<div class="small pull-right m-t-xs">9
 hours ago</div>

<h4>Business valuation</h4>

It is a long established fact that a
 reader will be distracted.

 <div class="small">Completion with:
 22%</div>

<div class="progress progress-mini">

<div style="width: 22%;">

class="progress-bar progress-bar-warning"></div>

</div>

<div class="small text-muted m-t-
 xs">Project end: 4:00 pm - 12.06.2014</div>

</a>

</li>

<li>

<a href="#">

<div class="small pull-right m-t-xs">9
 hours ago</div>

<h4>Contract with Company </h4>

Many desktop publishing packages
 and web page editors.

 <div class="small">Completion with:
 48%</div>

<div class="progress progress-mini">

<div style="width: 48%;">

class="progress-bar"></div>

</div>

</a>

</li>

<li>

<a href="#">

<div class="small pull-right m-t-xs">9
 hours ago</div>

<h4>Meeting</h4>

By the readable content of a page
 when looking at its layout.

 <div class="small">Completion with:
 14%</div>

<div class="progress progress-mini">

```

<a href="#">
    <div class="small pull-right m-t-xs">9
hours ago</div>
    <h4>Meeting</h4>
    By the readable content of a page
when looking at its layout.

    <div class="small">Completion with:
14%</div>
    <div class="progress progress-mini">
        <div style="width: 14%;">
        <div class="progress-bar progress-bar-info"></div>
    </div>
</a>
</li>
<li>
    <a href="#">
        <span class="label label-primary
pull-right">NEW</span>
        <h4>The generated</h4>
        <!--<div class="small pull-right m-t-
xs">9 hours ago</div>-->
        There are many variations of
passages of Lorem Ipsum available.
        <div class="small">Completion with:
22%</div>
        <div class="small text-muted m-t-
xs">Project end: 4:00 pm - 12.06.2014</div>
        </a>
    </li>
</ul>
</div>

<div id="tab-3" class="tab-pane">

    <div class="sidebar-title">
        <h3><i class="fa fa-gears"></i>
Settings</h3>
        <small><i class="fa fa-tim"></i> You have
14 projects. 10 not completed.</small>
    </div>

    <div class="setings-item">
        <span>
            Show notifications
        </span>
        <div class="switch">
            <div class="onoffswitch">
                <input type="checkbox"
name="collapsemenu" class="onoffswitch-
checkbox" id="example">
                <label class="onoffswitch-label"
for="example">

```

```

<span class="onoffswitch-
inner"></span>
<span class="onoffswitch-
switch"></span>
</label>
</div>
</div>
<div class="setings-item">
    <span>
        Disable Chat
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox"
name="collapsemenu" checked class="onoffswitch-
checkbox" id="example2">
            <label class="onoffswitch-label"
for="example2">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="setings-item">
    <span>
        Enable history
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox"
name="collapsemenu" class="onoffswitch-
checkbox" id="example3">
            <label class="onoffswitch-label"
for="example3">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="setings-item">
    <span>
        Show charts
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox"
name="collapsemenu" class="onoffswitch-
checkbox" id="example4">

```

```

        <label class="onoffswitch-label"
for="example4">
            <span class="onoffswitch-
inner"></span>
            <span class="onoffswitch-
switch"></span>
        </label>
    </div>
</div>
<div class="setings-item">
    <span>
        Offline users
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox" checked
name="collapsemenu" class="onoffswitch-
checkbox" id="example5">
            <label class="onoffswitch-label"
for="example5">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="setings-item">
    <span>
        Global search
    </span>
    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox" checked
name="collapsemenu" class="onoffswitch-
checkbox" id="example6">
            <label class="onoffswitch-label"
for="example6">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="setings-item">
    <span>
        Update everyday
    </span>
    <div class="switch">
        <div class="onoffswitch">

```

I believe that. Lorem Ipsum is simply dummy text of the printing and typesetting industry.

And typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

Over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

**PFMO.Web**  
**Views(Shared)**  
**\_SkinConfig.cshtml**

```

            <input type="checkbox" name="collapsemenu" class="onoffswitch-
checkbox" id="example7">
            <label class="onoffswitch-label"
for="example7">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="switch">
```

```

<div class="onoffswitch">
    <input type="checkbox"
name="collapsemenu" class="onoffswitch-
checkbox" id="collapsemenu">
        <label class="onoffswitch-label"
for="collapsemenu">
            <span class="onoffswitch-
inner"></span>
            <span class="onoffswitch-
switch"></span>
        </label>
    </div>
</div>
<div class="setings-item">
    <span>
        Fixed sidebar
    </span>

    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox"
name="fixedsidebar" class="onoffswitch-checkbox"
id="fixedsidebar">
                <label class="onoffswitch-label"
for="fixedsidebar">
                    <span class="onoffswitch-
inner"></span>
                    <span class="onoffswitch-
switch"></span>
                </label>
            </div>
        </div>
    </div>
    <div class="setings-item">
        <span>
            Top navbar
        </span>

        <div class="switch">
            <div class="onoffswitch">
                <input type="checkbox"
name="fixednavbar" class="onoffswitch-checkbox"
id="fixednavbar">
                    <label class="onoffswitch-label"
for="fixednavbar">
                        <span class="onoffswitch-
inner"></span>
                        <span class="onoffswitch-
switch"></span>
                    </label>
                </div>
            </div>
        </div>
        <div class="setings-item">
            <span>
                Fixed footer
            </span>
        </div>
    </div>
</div>

```

```

<span>
    Top navbar v.2
    <br />
    <small>*Primary layout</small>
</span>

<div class="switch">
    <div class="onoffswitch">
        <input type="checkbox"
name="fixednavbar2" class="onoffswitch-checkbox"
id="fixednavbar2">
            <label class="onoffswitch-label"
for="fixednavbar2">
                <span class="onoffswitch-
inner"></span>
                <span class="onoffswitch-
switch"></span>
            </label>
        </div>
    </div>
</div>
<div class="setings-item">
    <span>
        Boxed layout
    </span>

    <div class="switch">
        <div class="onoffswitch">
            <input type="checkbox"
name="boxedlayout" class="onoffswitch-checkbox"
id="boxedlayout">
                <label class="onoffswitch-label"
for="boxedlayout">
                    <span class="onoffswitch-
inner"></span>
                    <span class="onoffswitch-
switch"></span>
                </label>
            </div>
        </div>
    </div>
    <div class="setings-item">
        <span>
            Fixed footer
        </span>
    </div>
</div>

```

```

        <span class="onoffswitch-switch"></span>
    </label>
</div>
</div>
</div>

<div class="title">Skins</div>
<div class="setings-item default-skin">
    <span class="skin-name ">
        <a href="#" class="s-skin-0">
            Default
        </a>
    </span>
</div>
<div class="setings-item blue-skin">
    <span class="skin-name ">
        <a href="#" class="s-skin-1">
            Blue light
        </a>
    </span>
</div>
<div class="setings-item yellow-skin">
    <span class="skin-name ">
        <a href="#" class="s-skin-3">
            Yellow/Purple
        </a>
    </span>
</div>
<div class="setings-item ultra-skin">
    <span class="skin-name ">
        <a href="#" class="md-skin">
            Material Design
        </a>
    </span>
</div>
</div>
</div>
</div>

PFMO.Web
Views(Shared)
_TopNavbar.cshtml

<div class="row border-bottom">
    <nav class="navbar navbar-fixed-top"
        role="navigation" style="margin-bottom: 0">
        <div class="navbar-header">
            <a class="navbar-minimize minimize-style-2 btn btn-primary" href="#"><i class="fa fa-bars"></i></a>
            <form role="search" class="navbar-form-custom" method="post" name="cse"
                id="searchbox_demo"
                action="https://www.google.com/cse">

```

```

        <div class="form-group">
            <input type="text" placeholder="Search for something..." class="form-control" name="top-search" id="top-search">
        </div>
    </form>

@*<form name="cse" id="searchbox_demo"
action="https://www.google.com/cse">
    <input type="hidden" name="cref" value="" />
    <input type="hidden" name="ie" value="utf-8" />
    <input type="hidden" name="hl" value="" />
    <input name="q" type="text" size="40" />
    <input type="submit" name="sa" value="Search" />
</form>
<script type="text/javascript"
src="https%3A%2F%2Fcse.google.com%2Fcse/tools/onthefly?form=searchbox_demo&lang="></script>
*@

</div>
<@Html.Partial("_LoginPartial")>
</nav>
</div>

PFMO.Web
Views(Shared)
_TopNavbar2.cshtml

<div class="row border-bottom white-bg">
    <nav class="navbar navbar-static-top"
        role="navigation">
        <div class="navbar-header">
            <button aria-controls="navbar" aria-expanded="false" data-target="#navbar" data-toggle="collapse" class="navbar-toggle collapsed" type="button">
                <i class="fa fa-reorder"></i>
            </button>
            <a href="#" class="navbar-brand">Inspinia</a>
        </div>
        <div class="navbar-collapse collapse" id="navbar">
            <ul class="nav navbar-nav">
                <li class="active">
                    <a aria-expanded="false" role="button" href="@Url.Action("Index", "Layouts")"> Back to main Layout page</a>

```

```

        </li>
        <li class="dropdown">
            <a aria-expanded="false" role="button"
            href="#" class="dropdown-toggle" data-
            toggle="dropdown"> Menu item <span
            class="caret"></span></a>
            <ul role="menu" class="dropdown-
            menu">
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
            </ul>
        </li>
        <li class="dropdown">
            <a aria-expanded="false" role="button"
            href="#" class="dropdown-toggle" data-
            toggle="dropdown"> Menu item <span
            class="caret"></span></a>
            <ul role="menu" class="dropdown-
            menu">
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
            </ul>
        </li>
        <li class="dropdown">
            <a aria-expanded="false" role="button"
            href="#" class="dropdown-toggle" data-
            toggle="dropdown"> Menu item <span
            class="caret"></span></a>
            <ul role="menu" class="dropdown-
            menu">
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
            </ul>
        </li>
        <li class="dropdown">
            <a aria-expanded="false" role="button"
            href="#" class="dropdown-toggle" data-
            toggle="dropdown"> Menu item <span
            class="caret"></span></a>
            <ul role="menu" class="dropdown-
            menu">
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
            </ul>
        </li>
        <li class="dropdown">
            <a aria-expanded="false" role="button"
            href="#" class="dropdown-toggle" data-
            toggle="dropdown"> Menu item <span
            class="caret"></span></a>
            <ul role="menu" class="dropdown-
            menu">
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
                <li><a href="">Menu item</a></li>
            </ul>
        </li>
    </ul>

```

PFMO.Web  
Views(Shared)  
Error.cshtml

```

@model System.Web.Mvc.HandleErrorInfo

 @{
    ViewBag.Title = "Error";
}

<h1 class="text-danger">Error.</h1>
<h2 class="text-danger">An error occurred while
processing your request.</h2>

```

PFMO.Web  
Views(Shared)  
Lockout.cshtml

```

@model System.Web.Mvc.HandleErrorInfo

 @{
    ViewBag.Title = "Locked Out";
}

<hgroup>
    <h1 class="text-danger">Locked out.</h1>
    <h2 class="text-danger">This account has been
locked out, please try again later.</h2>
</hgroup>

```

PFMO.Web  
Controllers  
AccountController.cs

```

using System;
using System.Globalization;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;
using System.Web;
using System.Web.Mvc;
using Microsoft.AspNet.Identity;

```

```

using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using PFMO.Entities;
using PFMO.Contexts;
using PFMO.Web.Models;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class AccountController : ApplicationBaseController
    {
        private ApplicationSignInManager _signInManager;
        private ApplicationUserManager _userManager;

        public AccountController()
        {
        }

        public AccountController(ApplicationUserManager userManager, ApplicationSignInManager signInManager )
        {
            UserManager = userManager;
            SignInManager = signInManager;
        }

        public ApplicationSignInManager SignInManager
        {
            get
            {
                return _signInManager ??
                    HttpContext.GetOwinContext().Get<ApplicationSignInManager>();
            }
            private set
            {
                _signInManager = value;
            }
        }

        public ApplicationUserManager UserManager
        {
            get
            {
                return _userManager ??
                    HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>();
            }
            private set
            {
                _userManager = value;
            }
        }
    }

    _userManager = value;
    }

}

// GET: /Account/Login
[AllowAnonymous]
public ActionResult Login(string returnUrl)
{
    if (User.Identity.IsAuthenticated)
    {
        //redirect to some other page
        return RedirectToRoute("Home", "Index");
    }

    //ViewBag.ReturnUrl = returnUrl;
    //return View();
}

return View();
//ViewBag.ReturnUrl = returnUrl;
//return View();
}

// POST: /Account/Login
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // This doesn't count login failures towards
    account lockout
    // To enable password failures to trigger
    account lockout, change to shouldLockout: true
    var result = await
    SignInManager.PasswordSignInAsync(model.Email,
    model.Password, model.RememberMe,
    shouldLockout: false);
    switch (result)
    {
        case SignInStatus.Success:
            return RedirectToLocal(returnUrl);
        case SignInStatus.LockedOut:
            return View("Lockout");
        case SignInStatus.RequiresVerification:
            return RedirectToAction("SendCode",
            new { ReturnUrl = returnUrl, RememberMe =
            model.RememberMe });
        case SignInStatus.Failure:
    }
}

```

```

        default:
            ModelState.AddModelError("", "Invalid
login attempt.");
            return View(model);
        }

    //}

    // GET: /Account/VerifyCode
    [AllowAnonymous]
    public async Task<ActionResult>
VerifyCode(string provider, string returnUrl, bool
rememberMe)
{
    // Require that the user has already logged in
via username/password or external login
    if (!await
SignInManager.Has Been VerifiedAsync())
    {
        return View("Error");
    }
    return View(new VerifyCodeViewModel {
Provider = provider, ReturnUrl = returnUrl,
RememberMe = rememberMe });
}

// POST: /Account/VerifyCode
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
VerifyCode(VerifyCodeViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // The following code protects for brute force
attacks against the two factor codes.
    // If a user enters incorrect codes for a
specified amount of time then the user account
        // will be locked out for a specified amount of
time.
    // You can configure the account lockout
settings in IdentityConfig
    var result = await
SignInManager.TwoFactorSignInAsync(model.Provid
er, model.Code, isPersistent: model.RememberMe,
rememberBrowser: model.RememberBrowser);
    switch (result)
    {
        case SignInStatus.Success:
            return
RedirectToLocal(model.ReturnUrl);
        case SignInStatus.LockedOut:
            return View("Lockout");
        case SignInStatus.Failure:
        default:
            ModelState.AddModelError("", "Invalid
code.");
            return View(model);
    }

}

// GET: /Account/Register
[AllowAnonymous]
public ActionResult Register()
{
    return View();
}

// POST: /Account/Register
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser
        {
            UserName = model.Email,
            Email = model.Email,
            FirstName = model.FirstName,
            LastName = model.LastName
        };
        var result = await
userManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            await SignInManager.SignInAsync(user,
isPersistent:false, rememberBrowser:false);

            // For more information on how to
enable account confirmation and password reset
please visit
http://go.microsoft.com/fwlink/?LinkId=320771
            // Send an email with this link
            // string code = await
userManager.GenerateEmailConfirmationTokenAsy
nc(user.Id);
            // var callbackUrl =
Url.Action("ConfirmEmail", "Account", new { userId
});
        }
    }
}

```

```

= user.Id, code = code }, protocol:
Request.Url.Scheme);
    // await
UserManager.SendEmailAsync(user.Id, "Confirm
your account", "Please confirm your account by
clicking <a href=\"" + callbackUrl + "\">here</a>");

        return RedirectToAction("Index",
"Home");
    }
    AddErrors(result);
}

// If we got this far, something failed,
redisplay form
return View(model);
}

//
// GET: /Account/ConfirmEmail
[AllowAnonymous]
public async Task<ActionResult>
ConfirmEmail(string userId, string code)
{
    if (userId == null || code == null)
    {
        return View("Error");
    }
    var result = await
UserManager.ConfirmEmailAsync(userId, code);
    return View(result.Succeeded ?
"ConfirmEmail" : "Error");
}

//
// GET: /Account/ForgotPassword
[AllowAnonymous]
public ActionResult ForgotPassword()
{
    return View();
}

//
// POST: /Account/ForgotPassword
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
ForgotPassword(ForgotPasswordViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = await
UserManager.FindByNameAsync(model.Email);
if (user == null || !(await
UserManager.IsEmailConfirmedAsync(user.Id)))
{
    // Don't reveal that the user does not
exist or is not confirmed
    return
View("ForgotPasswordConfirmation");
}

// For more information on how to enable
account confirmation and password reset please
visit
http://go.microsoft.com/fwlink/?LinkId=320771
// Send an email with this link
// string code = await
UserManager.GeneratePasswordResetTokenAsync(
user.Id);
// var callbackUrl =
Url.Action("ResetPassword", "Account", new {
userId = user.Id, code = code }, protocol:
Request.Url.Scheme);
    // await
UserManager.SendEmailAsync(user.Id, "Reset
Password", "Please reset your password by clicking
<a href=\"" + callbackUrl + "\">here</a>");
    // return
RedirectToAction("ForgotPasswordConfirmation",
"Account");
}

// If we got this far, something failed,
redisplay form
return View(model);
}

//
// GET: /Account/ForgotPasswordConfirmation
[AllowAnonymous]
public ActionResult
ForgotPasswordConfirmation()
{
    return View();
}

//
// GET: /Account/ResetPassword
[AllowAnonymous]
public ActionResult ResetPassword(string code)
{
    return code == null ? View("Error") : View();
}

//
// POST: /Account/ResetPassword
[HttpPost]

```

```

[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
ResetPassword(ResetPasswordViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }
    var user = await
UserManager.FindByNameAsync(model.Email);
    if (user == null)
    {
        // Don't reveal that the user does not exist
        return
RedirectToAction("ResetPasswordConfirmation",
"Account");
    }
    var result = await
UserManager.ResetPasswordAsync(user.Id,
model.Code, model.Password);
    if (result.Succeeded)
    {
        return
RedirectToAction("ResetPasswordConfirmation",
"Account");
    }
    AddErrors(result);
    return View();
}

// 
// GET: /Account/ResetPasswordConfirmation
[AllowAnonymous]
public ActionResult
ResetPasswordConfirmation()
{
    return View();
}

// 
// POST: /Account/ExternalLogin
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult ExternalLogin(string
provider, string returnUrl)
{
    // Request a redirect to the external login
    provider
    return new ChallengeResult(provider,
Url.Action("ExternalLoginCallback", "Account", new
{ ReturnUrl = returnUrl }));
}

// 
// GET: /Account/SendCode
[AllowAnonymous]
public async Task<ActionResult>
SendCode(string returnUrl, bool rememberMe)
{
    var userId = await
SignInManager.GetVerifiedUserIdAsync();
    if (userId == null)
    {
        return View("Error");
    }
    var userFactors = await
UserManager.GetValidTwoFactorProvidersAsync(us
erId);
    var factorOptions =
userFactors.Select(purpose => new SelectListItem {
Text = purpose, Value = purpose }).ToList();
    return View(new SendCodeViewModel {
Providers = factorOptions, ReturnUrl = returnUrl,
RememberMe = rememberMe });
}

// 
// POST: /Account/SendCode
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
SendCode(SendCodeViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View();
    }
    // Generate the token and send it
    if (!await
SignInManager.SendTwoFactorCodeAsync(model.Se
lectedProvider))
    {
        return View("Error");
    }
    return RedirectToAction("VerifyCode", new {
Provider = model.SelectedProvider, ReturnUrl =
model.ReturnUrl, RememberMe =
model.RememberMe });
}

// 
// GET: /Account/ExternalLoginCallback
[AllowAnonymous]
public async Task<ActionResult>
ExternalLoginCallback(string returnUrl)
{
}

```

```

        var loginInfo = await
AuthenticationManager.GetExternalLoginInfoAsync(
);
if (loginInfo == null)
{
    return RedirectToAction("Login");
}

// Sign in the user with this external login
provider if the user already has a login
var result = await
SignInManager.ExternalSignInAsync(loginInfo,
isPersistent: false);
switch (result)
{
    case SignInStatus.Success:
        return RedirectToLocal(returnUrl);
    case SignInStatus.LockedOut:
        return View("Lockout");
    case SignInStatus.RequiresVerification:
        return RedirectToAction("SendCode",
new { ReturnUrl = returnUrl, RememberMe = false
});
    case SignInStatus.Failure:
    default:
        // If the user does not have an account,
then prompt the user to create an account
        ViewBag.ReturnUrl = returnUrl;
        ViewBag.LoginProvider =
loginInfo.Login.LoginProvider;
        return
View("ExternalLoginConfirmation", new
ExternalLoginConfirmationViewModel { Email =
loginInfo.Email });
    }
}

// POST: /Account/ExternalLoginConfirmation
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
ExternalLoginConfirmation(ExternalLoginConfirmation
onViewModel model, string returnUrl)
{
    if (User.Identity.IsAuthenticated)
    {
        return RedirectToAction("Index",
"Manage");
    }

    if (ModelState.IsValid)
    {
        // Get the information about the user from
the external login provider
        var info = await
AuthenticationManager.GetExternalLoginInfoAsync(
);
        if (info == null)
        {
            return View("ExternalLoginFailure");
        }
        var user = new ApplicationUser { UserName =
model.Email, Email = model.Email };
        var result = await
userManager.CreateAsync(user);
        if (result.Succeeded)
        {
            result = await
userManager.AddLoginAsync(user.Id, info.Login);
            if (result.Succeeded)
            {
                await
signInManager.SignInAsync(user, isPersistent: false,
rememberBrowser: false);
                return RedirectToLocal(returnUrl);
            }
        }
        AddErrors(result);
    }

    ViewBag.ReturnUrl = returnUrl;
    return View(model);
}

// POST: /Account/LogOff
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult LogOff()
{
    authenticationManager.SignOut(DefaultAuthenticat
ionTypes.ApplicationCookie);
    return RedirectToAction("Index", "Home");
}

// GET: /Account/ExternalLoginFailure
[AllowAnonymous]
public ActionResult ExternalLoginFailure()
{
    return View();
}

protected override void Dispose(bool disposing)
{
    if (disposing)
}

```

```

{
    if (_userManager != null)
    {
        _userManager.Dispose();
        _userManager = null;
    }

    if (_signInManager != null)
    {
        _signInManager.Dispose();
        _signInManager = null;
    }
}

base.Dispose(disposing);
}

#region Helpers
// Used for XSRF protection when adding
external logins
private const string XsrfKey = "XsrfId";

private IAuthenticationManager
AuthenticationManager
{
    get
    {
        return
HttpContext.GetOwinContext().Authentication;
    }
}

private void AddErrors(IdentityResult result)
{
    foreach (var error in result.Errors)
    {
        ModelState.AddModelError("", error);
    }
}

private ActionResult RedirectToLocal(string
returnUrl)
{
    if (Url.IsLocalUrl(returnUrl))
    {
        return Redirect(returnUrl);
    }
    return RedirectToAction("Index", "Home");
}

internal class ChallengeResult :
HttpUnauthorizedResult
{
    public ChallengeResult(string provider, string
redirectUri)
        : this(provider, redirectUri, null)
    {
    }

    public ChallengeResult(string provider, string
redirectUri, string userId)
    {
        LoginProvider = provider;
        RedirectUri = redirectUri;
        UserId = userId;
    }

    public string LoginProvider { get; set; }
    public string RedirectUri { get; set; }
    public string UserId { get; set; }

    public override void
ExecuteResult(ControllerContext context)
{
    var properties = new
AuthenticationProperties { RedirectUri = RedirectUri
};

    if (UserId != null)
    {
        properties.Dictionary[XsrfKey] = UserId;
    }

    context.HttpContext.GetOwinContext().Authentication.Challenge(properties, LoginProvider);
}
}
#endregion
}
}

PFMO.Web  

Controllers  

ApplicationBaseController.cs

using PFMO.Contexts;
using PFMO.Web.Helpers;
using System.Linq;
using System.Web.Mvc;

namespace PFMO.Web.Controllers
{
    [RequireSSL]
    public class ApplicationBaseController : Controller
    {
        protected PFMO_DB db = new PFMO_DB();
        protected override void
OnActionExecuted(ActionExecutedContext
filterContext)
{
        if (User != null)
}
}
}

```

```

{
    var context = new Identity_DB();
    var username = User.Identity.Name;

    if (!string.IsNullOrEmpty(username))
    {
        var user =
            context.Users.SingleOrDefault(u => u.UserName ==
                username);
        string fullName = string.Concat(new
            string[] { user.FirstName, " ", user.LastName });
        ViewData.Add("FullName", fullName);
    }
    base.OnActionExecuted(filterContext);
}
public ApplicationController()
{
}
}

```

**PFMO.Web  
Controllers  
CalendarController.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace PFMO.Web.Controllers
{
    public class CalendarController : ApplicationController
    {
        // GET: Calendar
        public ActionResult Index()
        {
            return View();
        }
    }
}

```

**PFMO.Web  
Controllers  
ContactsController.cs**

```

using System;
using System.Net;
using System.Web.Mvc;
using PFMO.Entities;
using PFMO.Logic.Interface;

```

```

using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class ContactsController : ApplicationController
    {
        private IPFMO<Contact> _contacts;

        //Initialization
        public ContactsController()
        {
            var obj = new ContactsLogic(db);
            _contacts = obj;
        }

        // GET: /Contacts/
        public ActionResult Index()
        {
            return View(_contacts.GetList());
        }

        public ActionResult Table()
        {
            return View(_contacts.GetList());
        }

        // GET: /Contacts/Details/5
        //public ActionResult Details(Guid? id)
        //{
        //    if (id == null)
        //    {
        //        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        //    }
        //    Contact contact = _contacts.Find(id.Value);
        //    if (contact == null)
        //    {
        //        return HttpNotFound();
        //    }
        //    return View(contact);
        //}
    }
}

```

```

// GET: /Contacts/Create
public ActionResult Create()
{
    return View();
}

```

// POST: /Contacts/Create

# PFMO.Web Controllers DesktopController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace PFMO.Web.Controllers
{

```

```

public class DesktopController : ApplicationBaseController
{
    // GET: Desktop
    public ActionResult Index()
    {
        return View();
    }
}

PFMO.Web
Controllers
EquipmentCategoriesController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class EquipmentCategoriesController : ApplicationBaseController
    {
        private IPFMOAsync<EquipmentCategory> _equipmentCategory;

        //Initialization
        public EquipmentCategoriesController()
        {
            var obj = new EquipmentCategoriesLogic(db);
            _equipmentCategory = obj;
        }

        // GET: /EquipmentCategories/
        public async Task<ActionResult> Index()
        {
            return View(await _equipmentCategory.ToListAsync());
        }
        public async Task<ActionResult> Table()
        {
            return View(await _equipmentCategory.ToListAsync());
        }
    }
}

return View(await _equipmentCategory.ToListAsync());
}

// GET: /EquipmentCategories/Details/5
public async Task<ActionResult> Details(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    EquipmentCategory equipmentCategory =
    await _equipmentCategory.FindAsync(id.Value);
    if (equipmentCategory == null)
    {
        return HttpNotFound();
    }
    return View(equipmentCategory);
}

// GET: /EquipmentCategories/Create
public ActionResult Create()
{
    return View();
}

// POST: /EquipmentCategories/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Create([Bind(Include="EquipmentCategoryID,EquipmentCategoryName,EquipmentCategoryDescription")]
) EquipmentCategory equipmentCategory
{
    if (ModelState.IsValid)
    {
        await _equipmentCategory.SaveChangesAsync(equipmentCategory);
        return RedirectToAction("Index");
    }
    return View(equipmentCategory);
}

// GET: /EquipmentCategories/Edit/5
public async Task<ActionResult> Edit(Guid? id)
{
}

```

```

    {
        if (id == null)
        {
            return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest)
        ;
        }
        EquipmentCategory equipmentCategory =
        await _equipmentCategory.FindAsync(id.Value);
        if (equipmentCategory == null)
        {
            return HttpNotFound();
        }
        return View(equipmentCategory);
    }

    // POST: /EquipmentCategories/Edit/5
    // To protect from overposting attacks, please
    enable the specific properties you want to bind to,
    for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult>
        Edit([Bind(Include="EquipmentCategoryID,Equipment
        CategoryName,EquipmentCategoryDescription")])
        EquipmentCategory equipmentCategory)
    {
        if (ModelState.IsValid)
        {
            await
        _equipmentCategory.SaveChangesAsync(equipment
        Category);
            return RedirectToAction("Index");
        }
        return View(equipmentCategory);
    }

    // GET: /EquipmentCategories/Delete/5
    public async Task<ActionResult> Delete(Guid?
    id)
    {
        if (id == null)
        {
            return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest)
        ;
        }
        EquipmentCategory equipmentCategory =
        await _equipmentCategory.FindAsync(id.Value);
        if (equipmentCategory == null)
        {
            return HttpNotFound();
        }
        return View(equipmentCategory);
    }
}

return View(equipmentCategory);
}

// POST: /EquipmentCategories/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var equipmentCategory = await
    _equipmentCategory.FindAsync(id);
    await
    _equipmentCategory.Remove(equipmentCategory);
    return RedirectToAction("Index");
    //await
    _equipmentCategory.SaveChangesAsync(equipment
    Category);
}

protected override void Dispose(bool disposing)
{
    _equipmentCategory.Dispose(disposing);
    base.Dispose(disposing);
}
}

PFMO.Web
Controllers
EquipmentController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class EquipmentsController : ApplicationBaseController
    {
        private IPFMOAsync<Equipment> _equipment;

        //Initialization
        public EquipmentsController()

```

```

    {
        var obj = new EquipmentsLogic(db);
        _equipment = obj;
    }

    // GET: /Equipments/
    public async Task<ActionResult> Index()
    {
        return View(await
    _equipment.ToListAsync());
    }

    // GET: /Equipments/Details/5
    public async Task<ActionResult> Details(Guid?
id)
    {
        if (id == null)
        {
            return new
    HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Equipment equipment = await
    _equipment.FindAsync(id.Value);
        if (equipment == null)
        {
            return HttpNotFound();
        }
        return View(equipment);
    }

    // GET: /Equipments/Create
    public ActionResult Create()
    {
        ViewBag.EquipmentCategoryID = new
    SelectList(db.EquipmentCategories,
    "EquipmentCategoryID",
    "EquipmentCategoryName");
        //ViewBag.EquipmentCategoryID = new
    SelectList(_equipmentCategory.EquipmentCategorie
s, "EquipmentCategoryID",
    "EquipmentCategoryName");
        return View();
    }

    // POST: /Equipments/Create
    // To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
        // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Create([Bind(Include="EquipmentID,EquipmentCod
e,EquipmentName,EquipmentCategoryID,Equipmen
tRemarks,EquipmentStatus,EquipmentBalance,Equi
pmentUnit,EquipmentReorderPoint,Target,Equipme
ntExpiryDate")]) Equipment equipment)
    {
        if (ModelState.IsValid)
        {
            await
    _equipment.SaveChangesAsync(equipment);
            return RedirectToAction("Index");
        }
    }

    ViewBag.EquipmentCategoryID = new
    SelectList(db.EquipmentCategories,
    "EquipmentCategoryID",
    "EquipmentCategoryName",
    equipment.EquipmentCategoryID);
        return View(equipment);
    }

    // GET: /Equipments/Edit/5
    public async Task<ActionResult> Edit(Guid? id)
    {
        if (id == null)
        {
            return new
    HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Equipment equipment = await
    _equipment.FindAsync(id.Value);
        if (equipment == null)
        {
            return HttpNotFound();
        }
        ViewBag.EquipmentCategoryID = new
    SelectList(db.EquipmentCategories,
    "EquipmentCategoryID",
    "EquipmentCategoryName",
    equipment.EquipmentCategoryID);
        return View(equipment);
    }

    // POST: /Equipments/Edit/5
    // To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
        // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Edit([Bind(Include="EquipmentID,EquipmentCode,E
quipmentName,EquipmentCategoryID,EquipmentRe
marks,EquipmentStatus,EquipmentBalance,Equipm

```

```

entUnit,EquipmentReorderPoint,Target,Equipmentsx
piryDate")] Equipment equipment)
{
    if (ModelState.IsValid)
    {
        await
_equipment.SaveChangesAsync(equipment);
        return RedirectToAction("Index");
    }
    ViewBag.EquipmentCategoryID = new
SelectList(db.EquipmentCategories,
"EquipmentCategoryID",
"EquipmentCategoryName",
equipment.EquipmentCategoryID);
    return View(equipment);
}

// GET: /Equipments/Delete/5
public async Task<ActionResult> Delete(Guid?
id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Equipment equipment = await
_equipment.FindAsync(id.Value);
    if (equipment == null)
    {
        return HttpNotFound();
    }
    return View(equipment);
}

// POST: /Equipments/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var equipment = await
_equipment.FindAsync(id);
    await _equipment.Remove(equipment);
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    _equipment.Dispose(disposing);
    base.Dispose(disposing);
}
}

```

**PFMO.Web  
Controllers  
FilesController.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace PFMO.Web.Controllers
{
    public class FilesController : Controller
    {
        // GET: Files
        public ActionResult Index()
        {
            return View();
        }
    }
}

```

**PFMO.Web  
Controllers  
HomeController.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class HomeController : ApplicationBaseController
    {
        public ActionResult Index()
        {
            return View();
        }
    }
}

```

**PFMO.Web  
Controllers  
LandController.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Web;
using System.Web.Mvc;

namespace PFMO.Web.Controllers
{
    public class LandController : ApplicationBaseController
    {
        // GET: Land
        public ActionResult Index()
        {
            return View();
        }
    }
}

PFMO.Web
Controllers
ManageController.cs

using System;
using System.Linq;
using System.Threading.Tasks;
using System.Web;
using System.Web.Mvc;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using PFMO.Web.Models;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class ManageController : ApplicationBaseController
    {
        private ApplicationSignInManager _signInManager;
        private ApplicationUserManager _userManager;

        public ManageController()
        {
        }

        public ManageController(ApplicationUserManager userManager, ApplicationSignInManager signInManager)
        {
            UserManager = userManager;
            SignInManager = signInManager;
        }
    }
}

public ApplicationSignInManager
SignInManager
{
    get
    {
        return _signInManager ??
HttpContext.GetOwinContext().Get<ApplicationSignInManager>();
    }
    private set
    {
        _signInManager = value;
    }
}

public ApplicationUserManager UserManager
{
    get
    {
        return _userManager ??
HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>();
    }
    private set
    {
        _userManager = value;
    }
}

// 
// GET: /Manage/Index
public async Task<ActionResult>
Index(ManageMessageId? message)
{
    ViewBag.StatusMessage =
message ==
ManageMessageId.ChangePasswordSuccess ? "Your
password has been changed."
: message ==
ManageMessageId.SetPasswordSuccess ? "Your
password has been set."
: message ==
ManageMessageId.SetTwoFactorSuccess ? "Your
two-factor authentication provider has been set."
: message == ManageMessageId.Error ?
"An error has occurred."
: message ==
ManageMessageId.AddPhoneSuccess ? "Your phone
number was added."
: message ==
ManageMessageId.RemovePhoneSuccess ? "Your
phone number was removed."
: "";
}

var userId = User.Identity.GetUserId();

```

```

        var model = new IndexViewModel
        {
            HasPassword = HasPassword(),
            PhoneNumber = await
UserManager.GetPhoneNumberAsync(userId),
            TwoFactor = await
UserManager.GetTwoFactorEnabledAsync(userId),
            Logins = await
UserManager.GetLoginsAsync(userId),
            BrowserRemembered = await
AuthenticationManager.TwoFactorBrowserRememb
eredAsync(userId)
        };
        return View(model);
    }

    // POST: /Manage/RemoveLogin
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
RemoveLogin(string loginProvider, string
providerKey)
    {
        ManageMessageId? message;
        var result = await
UserManager.RemoveLoginAsync(User.Identity.Get
UserId(), new UserLoginInfo(loginProvider,
providerKey));
        if (result.Succeeded)
        {
            var user = await
UserManager.FindByIdAsync(User.Identity.GetUser
Id());
            if (user != null)
            {
                await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
            }
            message =
ManageMessageId.RemoveLoginSuccess;
        }
        else
        {
            message = ManageMessageId.Error;
        }
        return RedirectToAction("ManageLogins",
new { Message = message });
    }

    // GET: /Manage/AddPhoneNumber
    public ActionResult AddPhoneNumber()
    {
        return View();
    }

    // POST: /Manage/AddPhoneNumber
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
AddPhoneNumber(AddPhoneNumberViewModel
model)
    {
        if (!ModelState.IsValid)
        {
            return View(model);
        }
        // Generate the token and send it
        var code = await
UserManager.GenerateChangePhoneNumberToken
Async(User.Identity.GetUserId(), model.Number);
        if (UserManager.SmsService != null)
        {
            var message = new IdentityMessage
            {
                Destination = model.Number,
                Body = "Your security code is: " + code
            };
            await
UserManager.SmsService.SendAsync(message);
        }
        return
RedirectToAction("VerifyPhoneNumber", new {
PhoneNumber = model.Number });
    }

    // POST:
/Manage/EnableTwoFactorAuthentication
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
EnableTwoFactorAuthentication()
    {
        await
UserManager.SetTwoFactorEnabledAsync(User.Iden
tity.GetUserId(), true);
        var user = await
UserManager.FindByIdAsync(User.Identity.GetUser
Id());
        if (user != null)
        {
            await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
        }
        return RedirectToAction("Index", "Manage");
    }
}

```

```

// POST: /Manage/DisableTwoFactorAuthentication
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DisableTwoFactorAuthentication()
{
    await
UserManager.SetTwoFactorEnabledAsync(User.Identity.GetUserId(), false);
    var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
    if (user != null)
    {
        await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
    }
    return RedirectToAction("Index", "Manage");
}

// GET: /Manage/VerifyPhoneNumber
public async Task<ActionResult>
VerifyPhoneNumber(string phoneNumber)
{
    var code = await
UserManager.GenerateChangePhoneNumberToken
Async(User.Identity.GetUserId(), phoneNumber);
    // Send an SMS through the SMS provider to
verify the phone number
    return phoneNumber == null ? View("Error") :
View(new VerifyPhoneNumberViewModel {
PhoneNumber = phoneNumber });
}

// POST: /Manage/VerifyPhoneNumber
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
VerifyPhoneNumber(VerifyPhoneNumberViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }
    var result = await
UserManager.ChangePhoneNumberAsync(User.Identity.GetUserId(),
model.PhoneNumber,
model.Code);
    if (result.Succeeded)
    {
        var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
        if (user != null)
        {
            await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
        }
        return RedirectToAction("Index", new {
Message = ManageMessageId.AddPhoneSuccess });
    }
    // If we got this far, something failed,
redisplay form
    ModelState.AddModelError("", "Failed to
verify phone");
    return View(model);
}

// GET: /Manage/RemovePhoneNumber
public async Task<ActionResult>
RemovePhoneNumber()
{
    var result = await
UserManager.SetPhoneNumberAsync(User.Identity.GetUserId(),
null);
    if (!result.Succeeded)
    {
        return RedirectToAction("Index", new {
Message = ManageMessageId.Error });
    }
    var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
    if (user != null)
    {
        await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
    }
    return RedirectToAction("Index", new {
Message =
ManageMessageId.RemovePhoneSuccess });
}

// GET: /Manage/ChangePassword
public ActionResult ChangePassword()
{
    return View();
}

// POST: /Manage/ChangePassword
[HttpPost]
[ValidateAntiForgeryToken]

```

```

        public async Task<ActionResult>
ChangePassword(ChangePasswordViewModel
model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }
    var result = await
UserManager.ChangePasswordAsync(User.Identity.
GetUserId(), model.OldPassword,
model.NewPassword);
    if (result.Succeeded)
    {
        var user = await
userManager.FindByIdAsync(User.Identity.GetUserId());
        if (user != null)
        {
            await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
        }
        return RedirectToAction("Index", new {
Message =
ManageMessageId.ChangePasswordSuccess });
    }
    AddErrors(result);
    return View(model);
}

// 
// GET: /Manage/SetPassword
public ActionResult SetPassword()
{
    return View();
}

// 
// POST: /Manage/SetPassword
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
SetPassword(SetPasswordViewModel model)
{
    if (ModelState.IsValid)
    {
        var result = await
UserManager.AddPasswordAsync(User.Identity.GetUserId(),
model.NewPassword);
        if (result.Succeeded)
        {
            var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
            if (user != null)
                {
                    {
                        await
SignInManager.SignInAsync(user, isPersistent: false,
rememberBrowser: false);
                    }
                    return RedirectToAction("Index", new {
Message = ManageMessageId.SetPasswordSuccess });
                }
            AddErrors(result);
        }
    }
}

// 
// GET: /Manage/ManageLogins
public async Task<ActionResult>
ManageLogins(ManageMessageId? message)
{
    ViewBag.StatusMessage =
message ==
ManageMessageId.RemoveLoginSuccess ? "The
external login was removed."
: message == ManageMessageId.Error ?
"An error has occurred."
: "";
    var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
    if (user == null)
    {
        return View("Error");
    }
    var userLogins = await
UserManager.GetLoginsAsync(User.Identity.GetUserId());
    var otherLogins =
AuthenticationManager.GetExternalAuthenticationTypes().
Where(auth => userLogins.All(u =>
auth.AuthenticationType !=
u.LoginProvider)).ToList();
    ViewBag.ShowRemoveButton =
user.PasswordHash != null || userLogins.Count > 1;
    return View(new ManageLoginsViewModel
{
    CurrentLogins = userLogins,
    OtherLogins = otherLogins
});
}

// 
// POST: /Manage/LinkLogin

```

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult LinkLogin(string provider)
{
    // Request a redirect to the external login
    provider to link a login for the current user
    return new
    AccountController.ChallengeResult(provider,
    Url.Action("LinkLoginCallback", "Manage"),
    User.Identity.GetUserId());
}

// GET: /Manage/LinkLoginCallback
public async Task<ActionResult>
LinkLoginCallback()
{
    var loginInfo = await
    AuthenticationManager.GetExternalLoginInfoAsync(
    XsrfKey, User.Identity.GetUserId());
    if (loginInfo == null)
    {
        return RedirectToAction("ManageLogins",
    new { Message = ManageMessageId.Error });
    }
    var result = await
    UserManager.AddLoginAsync(User.Identity.GetUser
    Id(), loginInfo.Login);
    return result.Succeeded ?
    RedirectToAction("ManageLogins") :
    RedirectToAction("ManageLogins", new { Message =
    ManageMessageId.Error });
}

protected override void Dispose(bool disposing)
{
    if (disposing && _userManager != null)
    {
        _userManager.Dispose();
        _userManager = null;
    }

    base.Dispose(disposing);
}

#region Helpers
// Used for XSRF protection when adding
external logins
private const string XsrfKey = "XsrfId";

private IAuthenticationManager
AuthenticationManager
{
    get
    {
        return
        HttpContext.GetOwinContext().Authentication;
    }
}

private void AddErrors(IdentityResult result)
{
    foreach (var error in result.Errors)
    {
        ModelState.AddModelError("", error);
    }
}

private bool HasPassword()
{
    var user =
    UserManager.FindById(User.Identity.GetUserId());
    if (user != null)
    {
        return user.PasswordHash != null;
    }
    return false;
}

private bool HasPhoneNumber()
{
    var user =
    UserManager.FindById(User.Identity.GetUserId());
    if (user != null)
    {
        return user.PhoneNumber != null;
    }
    return false;
}

public enum ManageMessageId
{
    AddPhoneSuccess,
    ChangePasswordSuccess,
    SetTwoFactorSuccess,
    SetPasswordSuccess,
    RemoveLoginSuccess,
    RemovePhoneSuccess,
    Error
}

#endregion
}
}

PFMO.Web
Controllers
PersonalDesignationController.cs

using System;

```

```

using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class PersonnelDesignationsController : ApplicationBaseController
    {
        private IPFMOAsync<PersonnelDesignation> _personnelDesignation;
        public PersonnelDesignationsController()
        {
            var obj = new PersonnelDesignationsLogic(db);
            _personnelDesignation = obj;
        }

        // GET: /PersonnelDesignations/
        public async Task<ActionResult> Index()
        {
            return View(await _personnelDesignation.ToListAsync());
        }

        public async Task<ActionResult> Table()
        {
            return View(await _personnelDesignation.ToListAsync());
        }

        // GET: /PersonnelDesignations/Details/5
        public async Task<ActionResult> Details(Guid? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            PersonnelDesignation personnelDesignation =
                await _personnelDesignation.FindAsync(id.Value);
            if (personnelDesignation == null)
            {
                return HttpNotFound();
            }
            return View(personnelDesignation);
        }

        // POST: /PersonnelDesignations/Create
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to,
        // for
        // more details see
        // http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> Create([Bind(Include="PersonnelDesignationID,Pers
        onnelDesignationName,PersonnelDesignationDescri
        ption")]] PersonnelDesignation personnelDesignation)
        {
            if (ModelState.IsValid)
            {
                await _personnelDesignation.SaveChangesAsync(personn
                elDesignation);
                return RedirectToAction("Index");
            }
        }

        return View(personnelDesignation);
    }

    // GET: /PersonnelDesignations/Edit/5
    public async Task<ActionResult> Edit(Guid? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        PersonnelDesignation personnelDesignation =
            await _personnelDesignation.FindAsync(id.Value);
        if (personnelDesignation == null)
        {
            return HttpNotFound();
        }
        return View(personnelDesignation);
    }

    // POST: /PersonnelDesignations/Edit/5
}

```

```

// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Edit([Bind(Include="PersonnelDesignationID,Person
nelDesignationName,PersonnelDesignationDescripti
on")] PersonnelDesignation personnelDesignation)
{
    if (ModelState.IsValid)
    {
        await
_personnelDesignation.SaveChangesAsync(personn
elDesignation);
        return RedirectToAction("Index");
    }
    return View(personnelDesignation);
}

// GET: /PersonnelDesignations/Delete/5
public async Task<ActionResult> Delete(Guid?
id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    PersonnelDesignation personnelDesignation
= await _personnelDesignation.FindAsync(id.Value);
    if (personnelDesignation == null)
    {
        return HttpNotFound();
    }
    return View(personnelDesignation);
}

// POST: /PersonnelDesignations/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var personnelDesignation = await
_personnelDesignation.FindAsync(id);
    await
_personnelDesignation.Remove(personnelDesignati
on);
    return RedirectToAction("Index");
}

```

```

protected override void Dispose(bool disposing)
{
    _personnelDesignation.Dispose(disposing);
    base.Dispose(disposing);
}

PFMO.Web
Controllers
PersonnelsController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class PersonnelsController : ApplicationBaseController
    {
        private IPFMOAsync<Personnel> _personnel;
        //Initialization
        public PersonnelsController()
        {
            var obj = new PersonnelsLogic(db);
            _personnel = obj;
        }

        // GET: /Personnels/
        public async Task<ActionResult> Index()
        {
            return View(await _personnel.ToListAsync());
        }
        public async Task<ActionResult> Table()
        {
            return View(await _personnel.ToListAsync());
        }

        // GET: /Personnels/Details/5
        public async Task<ActionResult> Details(Guid?
id)
{
    if (id == null)

```

```

    {
        return new
    HttpStatusCodeResult(HttpStatusCode.BadRequest)
    ;
    }
    Personnel personnel = await
    _personnel.FindAsync(id.Value);
    if (personnel == null)
    {
        return HttpNotFound();
    }
    return View(personnel);
}

// GET: /Personnels/Create
public ActionResult Create()
{
    ViewBag.PersonnelDesignationID = new
    SelectList(db.PersonnelDesignations,
    "PersonnelDesignationID",
    "PersonnelDesignationName");
    ViewBag.ProjectTeamID = new
    SelectList(db.ProjectTeams, "ProjectTeamID",
    "ProjectTeamName");
    ViewBag.SectionID = new
    SelectList(db.Sections, "SectionID", "SectionName",
    personnel.SectionID);
    return View(personnel);
}

// POST: /Personnels/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Create([Bind(Include="PersonnelID,PersonnelLastName,
PersonnelFirstName,PersonnelGender,PersonnelDesignationID,
PersonnelRemarks,SectionID,ProjectTeamID,PersonnelHireDate,PersonnelContactNumber")]
Personnel personnel)
{
    if (ModelState.IsValid)
    {
        await
    _personnel.SaveChangesAsync(personnel);
        return RedirectToAction("Index");
    }

    ViewBag.PersonnelDesignationID = new
    SelectList(db.PersonnelDesignations,
    "PersonnelDesignationID",
    "PersonnelDesignationName",
    personnel.PersonnelDesignationID);
    ViewBag.ProjectTeamID = new
    SelectList(db.ProjectTeams, "ProjectTeamID",
    "ProjectTeamName", personnel.ProjectTeamID);
    ViewBag.SectionID = new
    SelectList(db.Sections, "SectionID", "SectionName",
    personnel.SectionID);
    return View(personnel);
}

// GET: /Personnels/Edit/5
public async Task<ActionResult> Edit(Guid? id)
{
    if (id == null)
    {
        return new
    HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Personnel personnel = await
    _personnel.FindAsync(id.Value);
    if (personnel == null)
    {
        return HttpNotFound();
    }
    ViewBag.PersonnelDesignationID = new
    SelectList(db.PersonnelDesignations,
    "PersonnelDesignationID",
    "PersonnelDesignationName",
    personnel.PersonnelDesignationID);
    ViewBag.ProjectTeamID = new
    SelectList(db.ProjectTeams, "ProjectTeamID",
    "ProjectTeamName", personnel.ProjectTeamID);
    ViewBag.SectionID = new
    SelectList(db.Sections, "SectionID", "SectionName",
    personnel.SectionID);
    return View(personnel);
}

// POST: /Personnels/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Edit([Bind(Include="PersonnelID,PersonnelLastName,
PersonnelFirstName,PersonnelGender,PersonnelDesignationID,
PersonnelRemarks,SectionID,ProjectTeamID,PersonnelHireDate,PersonnelContactNumber")]
Personnel personnel)
{
}

```

```

        if (ModelState.IsValid)
        {
            await
            _personnel.SaveChangesAsync(personnel);
            return RedirectToAction("Index");
        }
        ViewBag.PersonnelDesignationID = new
        SelectList(db.PersonnelDesignations,
        "PersonnelDesignationID",
        "PersonnelDesignationName",
        personnel.PersonnelDesignationID);
        ViewBag.ProjectTeamID = new
        SelectList(db.ProjectTeams, "ProjectTeamID",
        "ProjectTeamName", personnel.ProjectTeamID);
        ViewBag.SectionID = new
        SelectList(db.Sections, "SectionID", "SectionName",
        personnel.SectionID);
        return View(personnel);
    }

    // GET: /Personnels/Delete/5
    public async Task<ActionResult> Delete(Guid?
    id)
    {
        if (id == null)
        {
            return new
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Personnel personnel = await
        _personnel.FindAsync(id.Value);
        if (personnel == null)
        {
            return HttpNotFound();
        }
        return View(personnel);
    }

    // POST: /Personnels/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
    DeleteConfirmed(Guid id)
    {
        var personnel = await
        _personnel.FindAsync(id);
        await _personnel.Remove(personnel);
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        _personnel.Dispose(disposing);
        base.Dispose(disposing);
    }
}

```

**PFMO.Web  
Controllers  
PersonnelTaskController.cs**

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class PersonnelTasksController : ApplicationBaseController
    {
        private IPFMOAsync<PersonnelTask>
        _personnelTask;
        //Initialization
        public PersonnelTasksController()
        {
            var obj = new PersonnelTasksLogic(db);
            _personnelTask = obj;
        }

        // GET: /PersonnelTasks/
        public async Task<ActionResult> Index()
        {
            var personneltasks =
            db.PersonnelTasks.Include(p => p.Personnel);
            return View(await
            personneltasks.ToListAsync());
        }

        public async Task<ActionResult> Table()
        {
            var personneltasks =
            db.PersonnelTasks.Include(p => p.Personnel);
            return View(await
            personneltasks.ToListAsync());
        }

        // GET: /PersonnelTasks/Details/5

```

```

        public async Task<ActionResult> Details(Guid?
id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    PersonnelTask personnelTask = await
_personnelTask.FindAsync(id.Value);
    if (personnelTask == null)
    {
        return HttpNotFound();
    }
    return View(personnelTask);
}

// GET: /PersonnelTasks/Create
public ActionResult Create()
{
    ViewBag.PersonnelID = new
SelectList(db.Personnels, "PersonnelID",
"PersonnelFullName");
    return View();
}

// POST: /PersonnelTasks/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
    // more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Create([Bind(Include="PersonnelTaskID,PersonnelTa
skTitle,PersonnelTaskDescription,PersonnelID,Perso
nnelTaskStartDate,PersonnelTaskEndDate")]
PersonnelTask personnelTask)
{
    if (ModelState.IsValid)
    {
        await
_personnelTask.SaveChangesAsync(personnelTask);
        return RedirectToAction("Index");
    }

    ViewBag.PersonnelID = new
SelectList(db.Personnels, "PersonnelID",
"PersonnelFullName", personnelTask.PersonnelID);
    return View(personnelTask);
}

// GET: /PersonnelTasks/Edit/5
public async Task<ActionResult> Edit(Guid? id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    PersonnelTask personnelTask = await
_personnelTask.FindAsync(id.Value);
    if (personnelTask == null)
    {
        return HttpNotFound();
    }
    ViewBag.PersonnelID = new
SelectList(db.Personnels, "PersonnelID",
"PersonnelFullName", personnelTask.PersonnelID);
    return View(personnelTask);
}

// POST: /PersonnelTasks/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
    // more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Edit([Bind(Include="PersonnelTaskID,PersonnelTask
Title,PersonnelTaskDescription,PersonnelID,Perso
nnelTaskStartDate,PersonnelTaskEndDate")]
PersonnelTask personnelTask)
{
    if (ModelState.IsValid)
    {
        await
_personnelTask.SaveChangesAsync(personnelTask);
        return RedirectToAction("Index");
    }

    ViewBag.PersonnelID = new
SelectList(db.Personnels, "PersonnelID",
"PersonnelFullName", personnelTask.PersonnelID);
    return View(personnelTask);
}

// GET: /PersonnelTasks/Delete/5
public async Task<ActionResult> Delete(Guid?
id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
}

```

```

        }

        PersonnelTask personnelTask = await
    _personnelTask.FindAsync(id.Value);
        if (personnelTask == null)
        {
            return HttpNotFound();
        }
        return View(personnelTask);
    }

    // POST: /PersonnelTasks/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var personnelTask = await
    _personnelTask.FindAsync(id);
    await
    _personnelTask.Remove(personnelTask);
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    _personnelTask.Dispose(disposing);
    base.Dispose(disposing);
}
}

PFMO.Web
Controllers
PinboardsController.cs

using System;
using System.Threading.Tasks;
using System.Net;
using System.Web.Mvc;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class PinboardsController :
    ApplicationController
    {
        private IPFMOAsync<Pinboard> _pinboard;
        //Initialization
        public PinboardsController()
        {
            var obj = new PinboardsLogic(db);
            _pinboard = obj;
        }

        PersonnelTask personnelTask = await
    _personnelTask.FindAsync(id.Value);
        if (personnelTask == null)
        {
            return HttpNotFound();
        }
        return View(personnelTask);
    }

    // GET: /Pinboards/
    public async Task<ActionResult> Index()
    {
        return View(await _pinboard.ToListAsync());
    }

    // GET: /Pinboards/Details/5
    public async Task<ActionResult> Details(Guid?
id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Pinboard pinboard = await
    _pinboard.FindAsync(id.Value);
    if (pinboard == null)
    {
        return HttpNotFound();
    }
    return View(pinboard);
}

// GET: /Pinboards/Create
public ActionResult Create()
{
    return View();
}

// POST: /Pinboards/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Create([Bind(Include="PinboardID,PinboardName,Pi
nboardContent,PinboardDate")] Pinboard pinboard)
{
    if (ModelState.IsValid)
    {
        await
        _pinboard.SaveChangesAsync(pinboard);
        return RedirectToAction("Index");
    }

    return View(pinboard);
}
}

```

```

// GET: /Pinboards/Edit/5
public async Task<ActionResult> Edit(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Pinboard pinboard = await
    _pinboard.FindAsync(id.Value);
    if (pinboard == null)
    {
        return HttpNotFound();
    }
    return View(pinboard);
}

// POST: /Pinboards/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Edit([Bind(Include="PinboardID,PinboardName,Pinb
oardContent,PinboardDate")] Pinboard pinboard)
{
    if (ModelState.IsValid)
    {
        await
        _pinboard.SaveChangesAsync(pinboard);
        return RedirectToAction("Index");
    }
    return View(pinboard);
}

// GET: /Pinboards/Delete/5
public async Task<ActionResult> Delete(Guid?
id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Pinboard pinboard = await
    _pinboard.FindAsync(id.Value);
    if (pinboard == null)
    {
        return HttpNotFound();
    }
    return View(pinboard);
}
return View(pinboard);
}

// POST: /Pinboards/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var pinboard = await
    _pinboard.FindAsync(id);
    await _pinboard.Remove(pinboard);
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    _pinboard.Dispose(disposing);
    base.Dispose(disposing);
}
}

PFMO.Web
Controllers
ProjectsController.cs

using System;
using System.Threading.Tasks;
using System.Net;
using System.Web.Mvc;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class ProjectsController : ApplicationBaseController
    {
        private IPFMOAsync<Project> _project;
        //Initialization
        public ProjectsController()
        {
            var obj = new ProjectsLogic(db);
            _project = obj;
        }

        // GET: /Projects/
        public async Task<ActionResult> Index()
        {
            return View(await _project.ToListAsync());
        }
}

```

```

// GET: /Projects/Details/5
public async Task<ActionResult> Details(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Project project = await _project.FindAsync(id.Value);
    if (project == null)
    {
        return HttpNotFound();
    }
    return View(project);
}

// POST: /Projects/Edit/5
// To protect from overposting attacks, please
// enable the specific properties you want to bind to,
for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Edit([Bind(Include="ProjectID,ProjectName,ProjectS
tatus,ProjectCompletion,ProjectStartDate,ProjectEn
dDate")] Project project)
{
    if (ModelState.IsValid)
    {
        await _project.SaveChangesAsync(project);
        return RedirectToAction("Index");
    }
    return View(project);
}

// GET: /Projects/Delete/5
public async Task<ActionResult> Delete(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Project project = await _project.FindAsync(id.Value);
    if (project == null)
    {
        return HttpNotFound();
    }
    return View(project);
}

// POST: /Projects/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)

```

```

    {
        Project project = await
        _project.FindAsync(id);
        await _project.Remove(project);
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        _project.Dispose(disposing);
        base.Dispose(disposing);
    }
}

PFMO.Web
Controllers
ProjectTeamsController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class ProjectTeamsController : ApplicationBaseController
    {
        private IPFMOAsync<ProjectTeam>
        _projectTeam;
        //Initialization
        public ProjectTeamsController()
        {
            var obj = new ProjectTeamsLogic(db);
            _projectTeam = obj;
        }

        // GET: /ProjectTeams/
        public async Task<ActionResult> Index()
        {
            var projectTeams =
            db.ProjectTeams.Include(p => p.Project);
            return View(await
            projectTeams.ToListAsync());
        }

        // GET: /ProjectTeams/Details/5
        public async Task<ActionResult> Details(Guid?
        id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            ProjectTeam projectTeam = await
            _projectTeam.FindAsync(id.Value);
            if (projectTeam == null)
            {
                return HttpNotFound();
            }
            return View(projectTeam);
        }

        // GET: /ProjectTeams/Create
        public ActionResult Create()
        {
            ViewBag.ProjectID = new
            SelectList(db.Projects, "ProjectID", "ProjectName");
            return View();
        }

        // POST: /ProjectTeams/Create
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to,
        // for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult>
        Create([Bind(Include="ProjectTeamID,ProjectTeam
        Name,ProjectTeamDescription,ProjectID")]
        ProjectTeam projectTeam)
        {
            if (ModelState.IsValid)
            {
                await
                _projectTeam.SaveChangesAsync(projectTeam);
                return RedirectToAction("Index");
            }

            ViewBag.ProjectID = new
            SelectList(db.Projects, "ProjectID", "ProjectName",
            projectTeam.ProjectID);
            return View(projectTeam);
        }
    }
}

```

```

    }

    // GET: /ProjectTeams/Edit/5
    public async Task<ActionResult> Edit(Guid? id)
    {
        if (id == null)
        {
            return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
        ;
        }

        ProjectTeam projectTeam = await
        _projectTeam.FindAsync(id.Value);
        if (projectTeam == null)
        {
            return HttpNotFound();
        }
        ViewBag.ProjectID = new
        SelectList(db.Projects, "ProjectID", "ProjectName",
        projectTeam.ProjectID);
        return View(projectTeam);
    }

    // POST: /ProjectTeams/Edit/5
    // To protect from overposting attacks, please
    enable the specific properties you want to bind to,
    for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult>
        Edit([Bind(Include="ProjectTeamID,ProjectTeamNa
        me,ProjectTeamDescription,ProjectID")]
        ProjectTeam projectTeam)
        {
            if (ModelState.IsValid)
            {
                await
            _projectTeam.SaveChangesAsync(projectTeam);
                return RedirectToAction("Index");
            }
            ViewBag.ProjectID = new
            SelectList(db.Projects, "ProjectID", "ProjectName",
            projectTeam.ProjectID);
            return View(projectTeam);
        }

        // GET: /ProjectTeams/Delete/5
        public async Task<ActionResult> Delete(Guid?
        id)
        {
            if (id == null)
            {
                return new
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
            ;
            }

            ProjectTeam projectTeam = await
            _projectTeam.FindAsync(id.Value);
            if (projectTeam == null)
            {
                return HttpNotFound();
            }
            return View(projectTeam);
        }

        // POST: /ProjectTeams/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult>
        DeleteConfirmed(Guid id)
        {
            var projectTeam = await
            _projectTeam.FindAsync(id);
            await _projectTeam.Remove(projectTeam);
            return RedirectToAction("Index");
        }

        protected override void Dispose(bool disposing)
        {
            _projectTeam.Dispose(disposing);
            base.Dispose(disposing);
        }
    }
}

PFMO.Web
Controllers
ReportsController.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace PFMO.Web.Controllers
{
    public class ReportsController : Controller
    {
        // GET: Reports
        public ActionResult Index()
        {
            return View();
        }
    }
}

```

```

PFMO.Web
Controllers
SectionHeadDesignationController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class SectionHeadDesignationsController : ControllerBase
    {
        private IPFMOAsync<SectionHeadDesignation> _sectionHeadDesignation;
        //Initialization
        public SectionHeadDesignationsController()
        {
            var obj = new SectionHeadDesignationsLogic(db);
            _sectionHeadDesignation = obj;
        }

        // GET: /SectionHeadDesignations/
        public async Task<ActionResult> Index()
        {
            return View(await _sectionHeadDesignation.ToListAsync());
        }

        public async Task<ActionResult> Table()
        {
            return View(await _sectionHeadDesignation.ToListAsync());
        }

        // GET: /SectionHeadDesignations/Details/5
        public async Task<ActionResult> Details(Guid? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
        }
    }
}

}
SectionHeadDesignation
sectionHeadDesignation = await _sectionHeadDesignation.FindAsync(id.Value);
if (sectionHeadDesignation == null)
{
    return HttpNotFound();
}
return View(sectionHeadDesignation);
}

// GET: /SectionHeadDesignations/Create
public ActionResult Create()
{
    return View();
}

// POST: /SectionHeadDesignations/Create
// To protect from overposting attacks, please
// enable the specific properties you want to bind to,
// for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Create([Bind(Include="SectionHeadDesignationID,SectionHeadDesignationName,SectionHeadDesignationDescription")] SectionHeadDesignation sectionHeadDesignation)
{
    if (ModelState.IsValid)
    {
        await _sectionHeadDesignation.SaveChangesAsync(sectionHeadDesignation);
        return RedirectToAction("Index");
    }
    return View(sectionHeadDesignation);
}

// GET: /SectionHeadDesignations/Edit/5
public async Task<ActionResult> Edit(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SectionHeadDesignation
    sectionHeadDesignation = await _sectionHeadDesignation.FindAsync(id.Value);
    if (sectionHeadDesignation == null)

```

```

    {
        return HttpNotFound();
    }
    return View(sectionHeadDesignation);
}

// POST: /SectionHeadDesignations/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Edit([Bind(Include="SectionHeadDesignationID,Secti
onHeadDesignationName,SectionHeadDesignationD
escription")]) SectionHeadDesignation
sectionHeadDesignation)
{
    if (ModelState.IsValid)
    {
        await
_sectionHeadDesignation.SaveChangesAsync(sectio
nHeadDesignation);
        return RedirectToAction("Index");
    }
    return View(sectionHeadDesignation);
}

// GET: /SectionHeadDesignations/Delete/5
public async Task<ActionResult> Delete(Guid?
id)
{
    if (id == null)
    {
        return new
HttpResponseCode(HttpStatusCode.BadRequest);
    }
    SectionHeadDesignation
sectionHeadDesignation = await
_sectionHeadDesignation.FindAsync(id.Value);
    if (sectionHeadDesignation == null)
    {
        return HttpNotFound();
    }
    return View(sectionHeadDesignation);
}

// POST: /SectionHeadDesignations/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var sectionHeadDesignation = await
_sectionHeadDesignation.FindAsync(id);
    await
_sectionHeadDesignation.Remove(sectionHeadDesi
gnation);
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    _sectionHeadDesignation.Dispose(disposing);
    base.Dispose(disposing);
}
}
}

PFMO.Web
Controllers
SectionHeadsController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class SectionHeadsController : ApplicationBaseController
    {
        private IPFMOAsync<SectionHead>
_sectionHead;
        //Initialization
        public SectionHeadsController()
        {
            var obj = new SectionHeadsLogic(db);
            _sectionHead = obj;
        }

        // GET: /SectionHeads/
        public async Task<ActionResult> Index()
        {
}
}
}

```

```

        var sectionheads = db.SectionHeads.Include(s => s.Section).Include(s => s.SectionHeadDesignation);
        return View(await sectionheads.ToListAsync());
    }

    public async Task<ActionResult> Table()
    {
        var sectionheads = db.SectionHeads.Include(s => s.Section).Include(s => s.SectionHeadDesignation);
        return View(await sectionheads.ToListAsync());
    }

    // GET: /SectionHeads/Details/5
    public async Task<ActionResult> Details(Guid? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        SectionHead sectionHead = await _sectionHead.FindAsync(id.Value);
        if (sectionHead == null)
        {
            return HttpNotFound();
        }
        return View(sectionHead);
    }

    // GET: /SectionHeads/Create
    public ActionResult Create()
    {
        ViewBag.SectionID = new SelectList(db.Sections, "SectionID", "SectionName");
        ViewBag.SectionHeadDesignationID = new SelectList(db.SectionHeadDesignations, "SectionHeadDesignationID", "SectionHeadDesignationName");
        return View();
    }

    // POST: /SectionHeads/Create
    // To protect from overposting attacks, please enable the specific properties you want to bind to, for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]

```

```

        public async Task<ActionResult> Create([Bind(Include="SectionHeadID,SectionHeadLastName,SectionHeadFirstName,SectionHeadGender,SectionHeadDesignationID,SectionHeadRemarks,SectionID,SectionHeadHireDate,SectionHeadContactNumber")] SectionHead sectionHead)
    {
        if (ModelState.IsValid)
        {
            await _sectionHead.SaveChangesAsync(sectionHead);
            return RedirectToAction("Index");
        }
    }

    ViewBag.SectionID = new SelectList(db.Sections, "SectionID", "SectionName", sectionHead.SectionID);
    ViewBag.SectionHeadDesignationID = new SelectList(db.SectionHeadDesignations, "SectionHeadDesignationID", "SectionHeadDesignationName", sectionHead.SectionHeadDesignationID);
    return View(sectionHead);
}

// GET: /SectionHeads/Edit/5
public async Task<ActionResult> Edit(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SectionHead sectionHead = await _sectionHead.FindAsync(id.Value);
    if (sectionHead == null)
    {
        return HttpNotFound();
    }
    ViewBag.SectionID = new SelectList(db.Sections, "SectionID", "SectionName", sectionHead.SectionID);
    ViewBag.SectionHeadDesignationID = new SelectList(db.SectionHeadDesignations, "SectionHeadDesignationID", "SectionHeadDesignationName", sectionHead.SectionHeadDesignationID);
    return View(sectionHead);
}

// POST: /SectionHeads/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind to, for

```

```

    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Edit([Bind(Include="SectionHeadID,SectionHeadLastName,SectionHeadFirstName,SectionHeadGender,SectionHeadDesignationID,SectionHeadRemarks,SectionID,SectionHeadHireDate,SectionHeadContactNumber")]
SectionHead sectionHead)
{
    if (ModelState.IsValid)
    {
        await _sectionHead.SaveChangesAsync(sectionHead);
        return RedirectToAction("Index");
    }
    ViewBag.SectionID = new SelectList(db.Sections, "SectionID", "SectionName",
sectionHead.SectionID);
    ViewBag.SectionHeadDesignationID = new SelectList(db.SectionHeadDesignations,
"SectionHeadDesignationID",
"SectionHeadDesignationName",
sectionHead.SectionHeadDesignationID);
    return View(sectionHead);
}

// GET: /SectionHeads/Delete/5
public async Task<ActionResult> Delete(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SectionHead sectionHead = await _sectionHead.FindAsync(id.Value);
    if (sectionHead == null)
    {
        return HttpNotFound();
    }
    return View(sectionHead);
}

// POST: /SectionHeads/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult> DeleteConfirmed(Guid id)
{
    var sectionHead = await _sectionHead.FindAsync(id);
    await _sectionHead.Remove(sectionHead);
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    _sectionHead.Dispose(disposing);
    base.Dispose(disposing);
}
}

PFMO.Web
Controllers
SectionsController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class SectionsController : ApplicationController
    {
        private IPFMOAsync<Section> _section;
        //Initialization
        public SectionsController()
        {
            var obj = new SectionsLogic(db);
            _section = obj;
        }

        // GET: /Sections/
        public async Task<ActionResult> Index()
        {
            return View(await _section.ToListAsync());
        }

        public async Task<ActionResult> Table()
        {
            return View(await _section.ToListAsync());
        }
}

```

```

// GET: /Sections/Details/5
public async Task<ActionResult> Details(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Section section = await _section.FindAsync(id.Value);
    if (section == null)
    {
        return HttpNotFound();
    }
    return View(section);
}

// POST: /Sections/Edit/5
// To protect from overposting attacks, please
// enable the specific properties you want to bind to,
for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Edit([Bind(Include="SectionID,SectionName,Section
Type,SectionDescription,SectionDateCreated")]
Section section)
{
    if (ModelState.IsValid)
    {
        await _section.SaveChangesAsync();
        return RedirectToAction("Index");
    }
    return View(section);
}

// GET: /Sections/Delete/5
public async Task<ActionResult> Delete(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Section section = await _section.FindAsync(id.Value);
    if (section == null)
    {
        return HttpNotFound();
    }
    return View(section);
}

// POST: /Sections/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)

```

```

    {
        var section = await _section.FindAsync(id);
        await _section.Remove(section);
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        _section.Dispose(disposing);
        base.Dispose(disposing);
    }
}

PFMO.Web
Controllers
SectionTasksController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class SectionTasksController : ControllerBase
    {
        private IPFMOAsync<SectionTask> _sectionTask;
        //Initialization
        public SectionTasksController()
        {
            var obj = new SectionTasksLogic(db);
            _sectionTask = obj;
        }

        // GET: /SectionTasks/
        public async Task<ActionResult> Index()
        {
            var sectiontasks = db.SectionTasks.Include(s => s.Section);
            return View(await sectiontasks.ToListAsync());
        }
    }
}

public async Task<ActionResult> Table()
{
    var sectiontasks = db.SectionTasks.Include(s => s.Section);
    return View(await sectiontasks.ToListAsync());
}

// GET: /SectionTasks/Details/5
public async Task<ActionResult> Details(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SectionTask sectionTask = await _sectionTask.FindAsync(id.Value);
    if (sectionTask == null)
    {
        return HttpNotFound();
    }
    return View(sectionTask);
}

// GET: /SectionTasks/Create
public ActionResult Create()
{
    ViewBag.SectionID = new SelectList(db.Sections, "SectionID", "SectionName");
    return View();
}

// POST: /SectionTasks/Create
// To protect from overposting attacks, please
// enable the specific properties you want to bind to,
// for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Create([Bind(Include="SectionTaskID,SectionTaskTitle,SectionTaskDescription,SectionID,SectionTaskStartDate,SectionTaskEndDate")] SectionTask sectionTask)
{
    if (ModelState.IsValid)
    {
        await _sectionTask.SaveChangesAsync(sectionTask);
        return RedirectToAction("Index");
    }
}

```

```

ViewBag.SectionID = new
SelectList(db.Sections, "SectionID", "SectionName",
sectionTask.SectionID);
return View(sectionTask);
}

// GET: /SectionTasks/Edit/5
public async Task<ActionResult> Edit(Guid? id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SectionTask sectionTask = await
_sectionTask.FindAsync(id.Value);
if (sectionTask == null)
{
    return HttpNotFound();
}
ViewBag.SectionID = new
SelectList(db.Sections, "SectionID", "SectionName",
sectionTask.SectionID);
return View(sectionTask);
}

// POST: /SectionTasks/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Edit([Bind(Include="SectionTaskID,SectionTaskTitle,
SectionTaskDescription,SectionID,SectionTaskStartD
ate,SectionTaskEndDate")]) SectionTask sectionTask)
{
    if (ModelState.IsValid)
    {
        await
_sectionTask.SaveChangesAsync(sectionTask);
        return RedirectToAction("Index");
    }
    ViewBag.SectionID = new
SelectList(db.Sections, "SectionID", "SectionName",
sectionTask.SectionID);
    return View(sectionTask);
}

// GET: /SectionTasks/Delete/5

```

```

public async Task<ActionResult> Delete(Guid?
id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SectionTask sectionTask = await
_sectionTask.FindAsync(id.Value);
if (sectionTask == null)
{
    return HttpNotFound();
}
return View(sectionTask);
}

// POST: /SectionTasks/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var sectionTask = await
_sectionTask.FindAsync(id);
await _sectionTask.Remove(sectionTask);
return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    _sectionTask.Dispose(disposing);
base.Dispose(disposing);
}
}
}
```

### PFMO.Web Controllers SuppliesController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

```

```

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class SuppliesController : ApplicationBaseController
    {
        private IPFMOAsync<Supply> _supply;
        //Initialization
        public SuppliesController()
        {
            var obj = new SuppliesLogic(db);
            _supply = obj;
        }

        // GET: /Supplies/
        public async Task<ActionResult> Index()
        {
            var supplies = db.Supplies.Include(s =>
s.Project).Include(s => s.SupplyCategory);
            return View(await supplies.ToListAsync());
        }

        // GET: /Supplies/Details/5
        public async Task<ActionResult> Details(Guid? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Supply supply = await _supply.FindAsync(id.Value);
            if (supply == null)
            {
                return HttpNotFound();
            }
            return View(supply);
        }

        // GET: /Supplies/Create
        public ActionResult Create()
        {
            ViewBag.ProjectID = new SelectList(db.Projects, "ProjectID", "ProjectName");
            ViewBag.SupplyCategoryID = new SelectList(db.SupplyCategories, "SupplyCategoryID", "SupplyCategoryName");
            return View();
        }

        // POST: /Supplies/Create
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to,
        // for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> Create([Bind(Include="SupplyID,SupplyCode,Supply
Name,SupplyCategoryID,SupplyRemarks,SupplyStat
us,SupplyBalance,SupplyUnit,SupplyReorderPoint,T
arget,ProjectID,EquipmentxpiryDate")] Supply
supply)
        {
            if (ModelState.IsValid)
            {
                await _supply.SaveChangesAsync(supply);
                return RedirectToAction("Index");
            }

            ViewBag.ProjectID = new
SelectList(db.Projects, "ProjectID", "ProjectName",
supply.ProjectID);
            ViewBag.SupplyCategoryID = new
SelectList(db.SupplyCategories, "SupplyCategoryID",
"SupplyCategoryName", supply.SupplyCategoryID);
            return View(supply);
        }

        // GET: /Supplies/Edit/5
        public async Task<ActionResult> Edit(Guid? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Supply supply = await _supply.FindAsync(id.Value);
            if (supply == null)
            {
                return HttpNotFound();
            }
            ViewBag.ProjectID = new
SelectList(db.Projects, "ProjectID", "ProjectName",
supply.ProjectID);
            ViewBag.SupplyCategoryID = new
SelectList(db.SupplyCategories, "SupplyCategoryID",
"SupplyCategoryName", supply.SupplyCategoryID);
            return View(supply);
        }

        // POST: /Supplies/Edit/5
    }
}

```

```

// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
Edit([Bind(Include="SupplyID,SupplyCode,SupplyNa
me,SupplyCategoryID,SupplyRemarks,SupplyStatus,
SupplyBalance,SupplyUnit,SupplyReorderPoint,Targ
et,ProjectID,EquipmentxpiryDate")] Supply supply)
{
    if (ModelState.IsValid)
    {
        await _supply.SaveChangesAsync(supply);
        return RedirectToAction("Index");
    }
    ViewBag.ProjectID = new
SelectList(db.Projects, "ProjectID", "ProjectName",
supply.ProjectID);
    ViewBag.SupplyCategoryID = new
SelectList(db.SupplyCategories, "SupplyCategoryID",
"SupplyCategoryName", supply.SupplyCategoryID);
    return View(supply);
}

// GET: /Supplies/Delete/5
public async Task<ActionResult> Delete(Guid?
id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Supply supply = await
_supply.FindAsync(id.Value);
    if (supply == null)
    {
        return HttpNotFound();
    }
    return View(supply);
}

// POST: /Supplies/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var supply = await _supply.FindAsync(id);
    await _supply.Remove(supply);
    return RedirectToAction("Index");
}
}

protected override void Dispose(bool disposing)
{
    _supply.Dispose(disposing);
    base.Dispose(disposing);
}
}

PFMO.Web
Controllers
SupplyCategoriesController.cs

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Threading.Tasks;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PFMO.Contexts;
using PFMO.Entities;
using PFMO.Logic.Interface;
using PFMO.Logic;

namespace PFMO.Web.Controllers
{
    [Authorize]
    public class SupplyCategoriesController : ControllerBase
    {
        private IPFMOAsync<SupplyCategory>
_supplyCategory;
        //Initialization
        public SupplyCategoriesController()
        {
            var obj = new SupplyCategoriesLogic(db);
            _supplyCategory = obj;
        }

        // GET: /SupplyCategories/
        public async Task<ActionResult> Index()
        {
            return View(await
_supplyCategory.ToListAsync());
        }
        public async Task<ActionResult> Table()
        {
            return View(await
_supplyCategory.ToListAsync());
        }
    }
}

```

```

// GET: /SupplyCategories/Details/5
public async Task<ActionResult> Details(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SupplyCategory supplyCategory = await _supplyCategory.FindAsync(id.Value);
    if (supplyCategory == null)
    {
        return HttpNotFound();
    }
    return View(supplyCategory);
}

// GET: /SupplyCategories/Create
public ActionResult Create()
{
    return View();
}

// POST: /SupplyCategories/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Create([Bind(Include="SupplyCategoryId,SupplyCate
goryName,SupplyCategoryDescription")])
SupplyCategory supplyCategory
{
    if (ModelState.IsValid)
    {
        await _supplyCategory.SaveChangesAsync(supplyCategory
);
        return RedirectToAction("Index");
    }
    return View(supplyCategory);
}

// GET: /SupplyCategories/Edit/5
public async Task<ActionResult> Edit(Guid? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SupplyCategory supplyCategory = await _supplyCategory.FindAsync(id.Value);
    if (supplyCategory == null)
    {
        return HttpNotFound();
    }
    return View(supplyCategory);
}

// POST: /SupplyCategories/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to,
for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Edit([Bind(Include="SupplyCategoryId,SupplyCatego
ryName,SupplyCategoryDescription")])
SupplyCategory supplyCategory
{
    if (ModelState.IsValid)
    {
        await
_supplyCategory.SaveChangesAsync(supplyCategory
);
        return RedirectToAction("Index");
    }
    return View(supplyCategory);
}

// GET: /SupplyCategories/Delete/5
public async Task<ActionResult> Delete(Guid?
id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    SupplyCategory supplyCategory = await _supplyCategory.FindAsync(id.Value);
    if (supplyCategory == null)
    {
        return HttpNotFound();
    }
    return View(supplyCategory);
}

```

```
// POST: /SupplyCategories/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
DeleteConfirmed(Guid id)
{
    var supplyCategory = await
_supplyCategory.FindAsync(id);
    await
_supplyCategory.Remove(supplyCategory);
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    _supplyCategory.Dispose(disposing);
    base.Dispose(disposing);
}
```