**Contents**

[**Section 1 (Overview)** 2](#_Toc524344347)

[1- Training program will begin with installation for SQL server 2014 and visual studio in addition to configuration for IIS. 2](#_Toc524344348)

[2- Making a small research and a small example using html and JS to fill data in a table. 2](#_Toc524344349)

[3- Overview about Angular JS and .net MVC. 2](#_Toc524344350)

[4- Learn how to use ng-app and ng-controller in Angular and how to bind data to HTML. 2](#_Toc524344351)

[5- Learn how to use Controllers in C#. 2](#_Toc524344352)

[6- Make an example of getting some data from memory using C# Controller by Rest API and present them into a grid using ng-repeat (angular js). 2](#_Toc524344353)

[Ex: 2](#_Toc524344354)

[7- Implement the full flow of getting data from SQL database by creating store procedure and using ADO.Net in C# code. 2](#_Toc524344355)

[8- Review the basic flow and giving some enhancement notes. 2](#_Toc524344356)

[9- Create new class library for data access layer and adding new class with generic method to get data from database. 2](#_Toc524344357)

[**Section 2 (Management pages with editor)** 3](#_Toc524344358)

[**Task 1**: Implement Students page that includes a grid having two columns (Student Id, Student Name). 3](#_Toc524344359)

[**Task 2**: 3](#_Toc524344360)

[**Task 3**: Implement add new student and edit existing. 3](#_Toc524344361)

[**Task 4**: Implement Rooms Page 3](#_Toc524344362)

[**Section 3 (Enums and Selectors)** 4](#_Toc524344363)

[**Task 5**: Implement Room Selector in Student editor page. 4](#_Toc524344364)

[**Task 6**: Implement Gender selector 4](#_Toc524344365)

[**Section 4: (Directives & Promises)** 5](#_Toc524344366)

[**Task 8**: Create waitMultiplePrpmises 5](#_Toc524344367)

[**Task 9**: Create Loader Directive as Attribute 5](#_Toc524344368)

[**Section 5: (Abstraction and Best practices)** 5](#_Toc524344369)

[**Task 11**: Implement payment method as abstract 5](#_Toc524344370)

[**Section 6: (Exam 1 Drilldown Question)** 6](#_Toc524344371)

[**Section 7: (Exam 2 Question & Documentation)** 6](#_Toc524344372)

Training Plan

# **Section 1 (Overview)**

## Training program will begin with installation for SQL server 2014 and visual studio in addition to configuration for IIS.

## Making a small research and a small example using html and JS to fill data in a table.

## Overview about Angular JS and .net MVC.

## Learn how to use ng-app and ng-controller in Angular and how to bind data to HTML.

## Learn how to use Controllers in C#.

## Make an example of getting some data from memory using C# Controller by Rest API and present them into a grid using ng-repeat (angular js).

## Ex:

|  |  |
| --- | --- |
| **ID** | **Name** |
| 1 | Samer |
| 2 | Wissam |

## Implement the full flow of getting data from SQL database by creating store procedure and using ADO.Net in C# code.

## Review the basic flow and giving some enhancement notes.

## Create new class library for data access layer and adding new class with generic method to get data from database.

Note: This Method should be generic in terms of T and it should take three parameters.

1. SP Name.
2. SP Parameters.
3. Mapper to map result.

List<T> ExcuteSPItems <T> (string spName, List<SqlParameter> sqlParameters, Func<IDataReader, T> mapper)

**At The end of this section review notes should be given for the code done above.**

# **Section 2 (Management pages with editor)**

## **Task 1**: Implement Students page that includes a grid having two columns (Student Id, Student Name).

Steps:

1. Create student table with ID and Name.
2. Create Student Manager to retrieve data from database.
3. Show the results in the student grid.

## **Task 2**:

1. Implement pagination for student grid by sending some information to store procedure to make the paging works well using server side calls.
2. Adding a text box and a small search button that allow user to search for the name in the student grid. This filtering should be implemented and handled server side and the grid will show any student contains the entered string in the text box.

## **Task 3**: Implement add new student and edit existing.

1. Learn how to implement a popup modal and how to pass parameters from main page to popup.
2. Add a text box in the modal with save button to add new student.
3. Add an edit button in the main page to edit an existing student by sent the studentId to the modal and get the student entity from database to show existing name in the text box.
4. When click save the editor should know that it should update the student entity.

## **Task 4**: Implement Rooms Page

Steps:

1. Create room table with ID and Name.
2. Create Room Manager to retrieve data from database.
3. Show the results in the room grid.

**At The end of this section review notes should be given for the code done above.**

**Some Notes should be satisfied:**

1. **Learn how to use services and separate JS Files to be more usable and clear.**
2. **Create class library for entities and managers to know more about levels in solution.**
3. **Use ng-view to make a single page application.**

# **Section 3 (Enums and Selectors)**

## **Task 5**: Implement Room Selector in Student editor page.

Steps:

1. Create new column in student table which is Room ID.
2. Create Room Selector and Map room data to info.
3. Use the selector in student editor to map student to specific room.
4. Map room name for student details to show it in student grid.
5. Create Room multi-select and add it to student management page to filter student in grid.

## **Task 6**: Implement Gender selector

1. Learn about enums in C# and constants in angular.
2. Implement gender selector that will include(Male, Female)
3. Add gender selector in student editor and save the gender into database.

**Task 7**: Implement Payment selector

1. Implement payment selector that will include(Cash, Credit Card)
2. Add payment selector in student editor and save the payment enum into database.

**At The end of this section review notes should be given for the code done above.**

# **Section 4: (Directives & Promises)**

## **Task 8**: Create waitMultiplePrpmises

1. Learn how to use promise in angular.
2. Create a common function that will take array of promises and wait them to finish then return only one promise.

## **Task 9**: Create Loader Directive as Attribute

1. Implement loader design to show it when getting data and loading the form.
2. Loader directive should be an attribute directive
3. It should take a Boolean value to show or hide.
4. Its role is to be shown when starting load the form and disappear when finish loading.

**Task 10**: Convert Selectors to Directives

1. Convert All Selectors to directives using best practices.
2. Explain our directive conventions that should be used in selectors to make sure that directives will load in a good way.
3. Explain API for directives and the two main methods load and get to communicate with directives using API and on-ready event function;
4. Explain how we should use the above approach to wait on directive promise to make sure it is loaded successfully.

**At The end of this section review notes should be given for the code done above.**

# **Section 5: (Abstraction and Best practices)**

## **Task 11**: Implement payment method as abstract

1. Explain the abstract concept and how it is used.
2. Explain how to use directive wrapper that will be given for trainees.
3. Explain the meaning of FQTN and how to use it.
4. Create extension table that includes (ID, Name , ConfigType, Settings)
5. Implement selective that will read the directives from extension table by config type and use directive wrapper to show the selected directive.
6. Explain how should we return the $type of the class to save the implementation of abstract class.
7. Save the data selected for abstract class and handle it in edit mode.
8. Implement two type of payment method
   1. Cash payment that includes (Amount).
   2. Credit card payment that includes (Type, Number , CNU, Expiration date)
9. Use GetDescription abstract method to show the description of payment in student grid.

**At The end of this section review notes should be given for the code done above.**

# **Section 6: (Exam 1 Drilldown Question)**

1. This exam will be consists of two dependent pages, for example: Company and Branches
2. The trainee should make the drill down concept by showing the companies in a grid and adding plus button on each row.
3. If we press the plus button a drill down shows the branches related for selected company.

**At The end of this section review notes should be given for the code done above.**

# **Section 7: (Exam 2 Question & Documentation)**

The final task is a small exam:

1. Create building page that have only ID and Name
2. Create building selector and use it in room editor to map room to building by adding building id in room table.
3. Add the building selector in student editor and it should satisfy the following:
   1. In Add Mode
      1. The building selector will be filled by all buildings
      2. The room selector will be empty
      3. When choosing a building the room selector will show all the rooms related to selected building.
   2. In Edit Mode
      1. The Selected building should be set selected.
      2. The room selected will show the rooms of selected building and will show the selected room.
      3. In this mode the load will be done for all controls at first load so this should be handled to prevent load any data after the load promises finished.

After finishing the final task, it is required from trainee to make a small document about everything he learned from this training plan and to add any feedback he/she want.