



ASP.NET MVC

Lab Book



Document Revision History

Date	Revision No.	Author	Summary of Changes
10-Jun-2011	1	Ajit Jog	Content Creation
10-Jun-2011	1	Karthikeyan Muthukrishnan	Review
08-Aug-2012	2.0	Ganesh Desai	Content Revamp
12-May-2016	3.0	Nachiket Inamdar	Rectified mistakes and removed Windows Application Development content.
	4.0	Sangeetha C	Incorporated C# 6.0 Features
23-Jan-2019	5.0	Shital Patil	Incorporated C# 7.0 Features



Table of Contents

Lab 1 : Simple Web Application using ASP.NET MVC	4
Lab 2 : Working with Model Validation.....	8
Lab 3 : Working with Web Optimization and AJAX	10

Lab 1 : Simple Web Application using ASP.NET MVC

Objective	This Lab will help you understand How to create Web Application using ASP.NET MVC
Time	120 Mins

1. You have been assigned the task of creating a web application to maintain employee details for an organization.

The details required are : employee id, employee name, designation and qualification details.

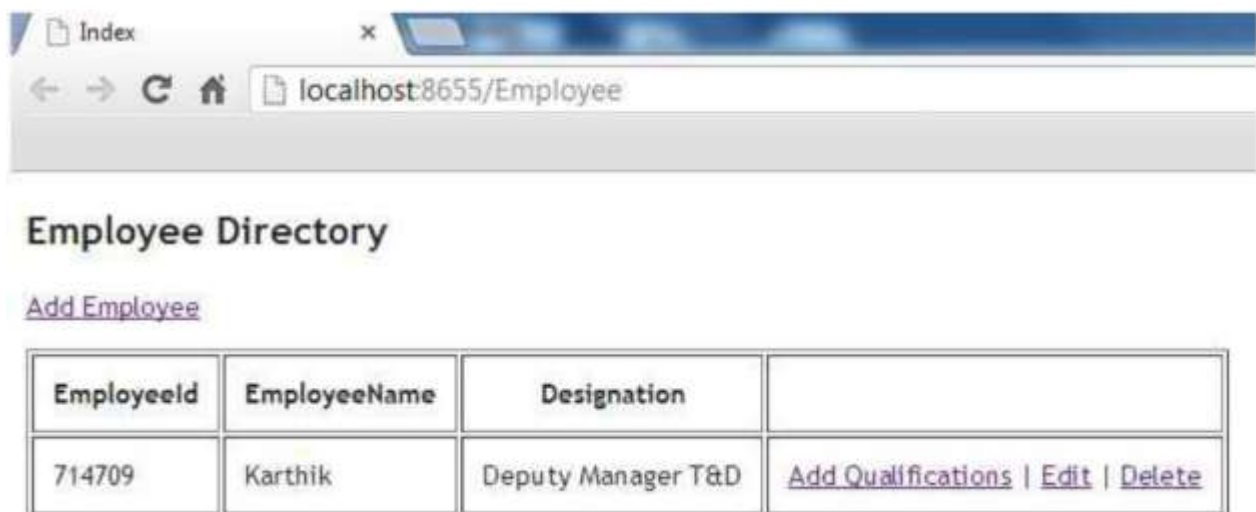
An employee can have more than one qualification detail, depending on whether the employee is a graduate or post-graduate.

Develop the web application using the screen shots provided to understand the flow of the application.

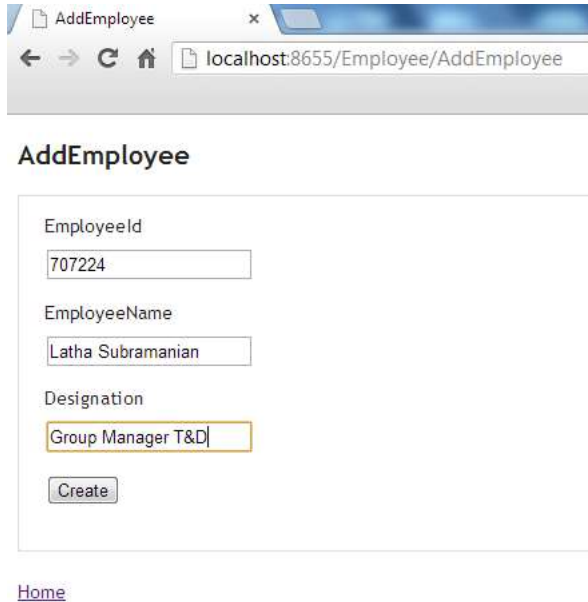
The application must use the URL patterns displayed in the snap shots.

Use the HTML helper methods and the ViewBag to transfer data between views

When the user runs the application, the following page is displayed:

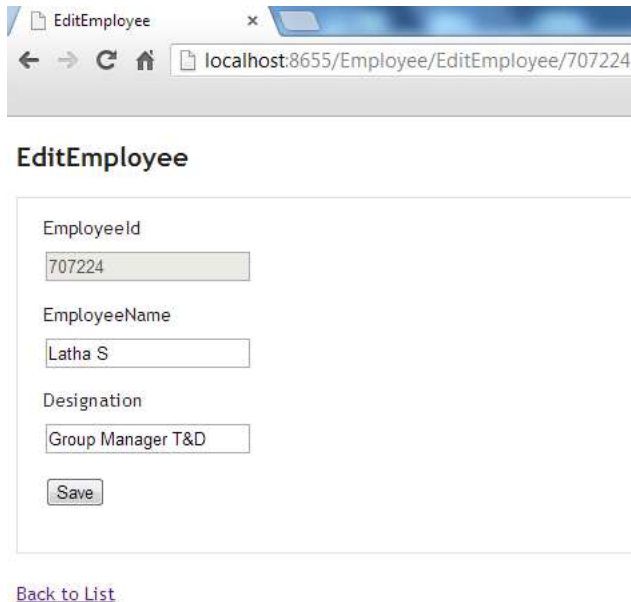


On clicking the “Add Employee” link, the user is redirected to the following page:



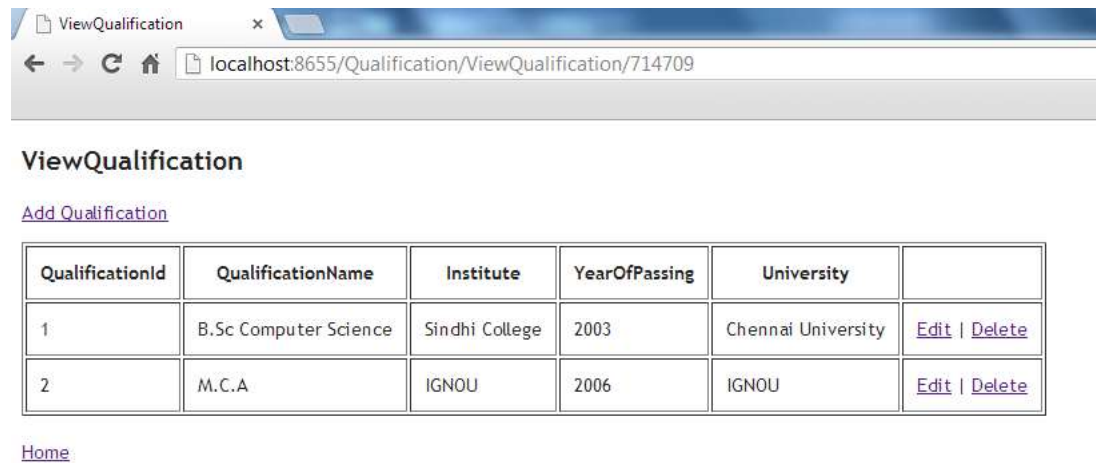
The screenshot shows a web browser window with the title "AddEmployee" and the URL "localhost:8655/Employee/AddEmployee". The page content includes a form with three text input fields: "EmployeeId" containing "707224", "EmployeeName" containing "Latha Subramanian", and "Designation" containing "Group Manager T&D". Below these fields is a "Create" button. At the bottom of the page is a link labeled "Home".

The following page shows the screen for editing an employee's details:



The screenshot shows a web browser window with the title "EditEmployee" and the URL "localhost:8655/Employee/EditEmployee/707224". The page content includes a form with three text input fields: "EmployeeId" containing "707224", "EmployeeName" containing "Latha S", and "Designation" containing "Group Manager T&D". Below these fields is a "Save" button. At the bottom of the page is a link labeled "Back to List".

When the user clicks on the “Add Qualification” link in the home page, the application redirects the user to the following page:



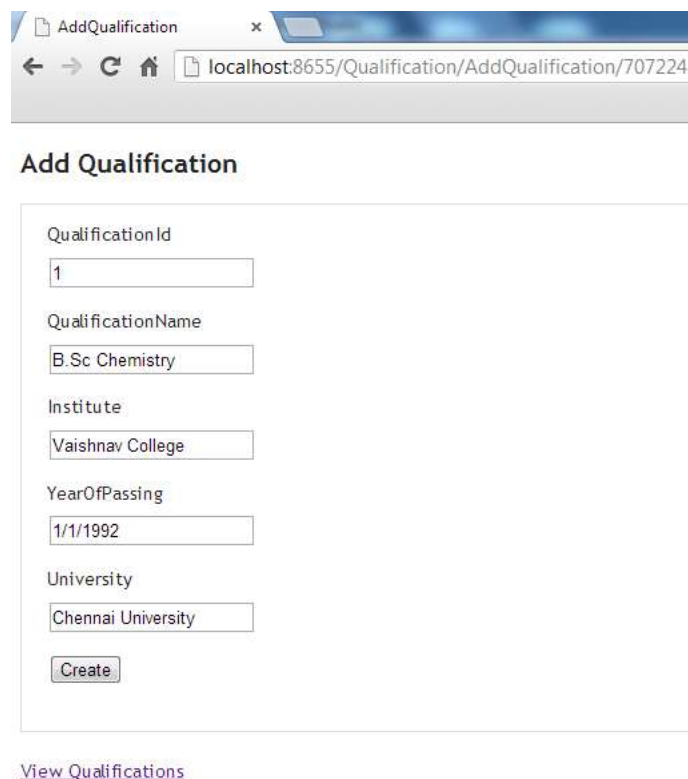
ViewQualification

[Add Qualification](#)

QualificationId	QualificationName	Institute	YearOfPassing	University	
1	B.Sc Computer Science	Sindhi College	2003	Chennai University	Edit Delete
2	M.C.A	IGNOU	2006	IGNOU	Edit Delete

[Home](#)

On clicking the “Add Qualification” link, the following page is displayed:



Add Qualification

QualificationId
1

QualificationName
B.Sc Chemistry

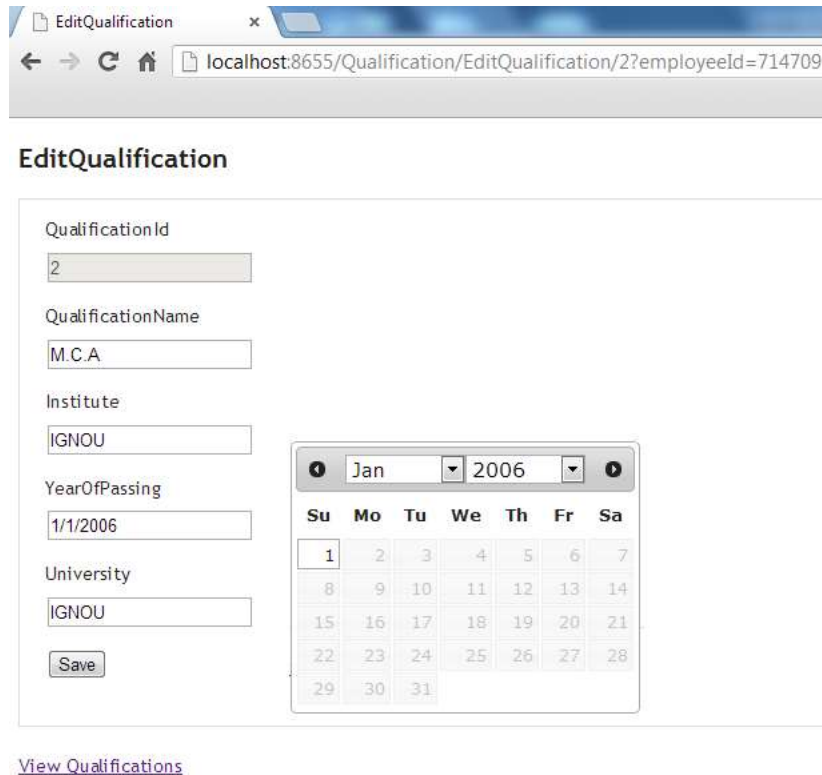
Institute
Vaishnav College

YearOfPassing
1/1/1992

University
Chennai University

[View Qualifications](#)

Existing qualification details can be modified using the following page:



EditQualification

QualificationId
2

QualificationName
M.C.A

Institute
IGNOU

YearOfPassing
1/1/2006

University
IGNOU

Calendar: Jan 2006

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

[View Qualifications](#)

Lab 2 : Working with Model Validation

Objective	This Lab will help you understand How to define Model Validation
Time	120 Mins

1. Create a web application using ASP.NET MVC for the administrator of an online shopping site. The application must enable the user to:
 - Add a Category
 - Add Products to the Category

Adding a category involves adding a category Id (integer) and name (string) which should accept minimum 3 alphabets and maximum 25 alphabets)

Adding a new product involves:

- Product Id as integer and should not exist in the data store.
- Product name as string and it should have minimum 2 characters and maximum 30 characters.
- Entry Date
- Category (must be populated from the available categories)
- Price should be integer and cannot be 0 or negative.

Use model validations and enable client side validations for the above application.

2. Add functionality to search for a product by entering the product name in a text box. The product name must be auto completed according to the names existing in the data source.
3. Create a Patient Information System using Asp.Net MVC and Entity Framework Code First Approach. Following details of the patient need to be stored into a database.
Patients:- PatientID, PatientName, Gender, Age, Contact, DoctorName
Implement the following functionality
 - a. Create a View to Add Patient Information



- b. Create a View to Search Patient by ID
- c. Create a View to Search Patient by Doctor name
- d. Create a View to Delete the patient based on ID

Lab 3 : Working with Web Optimization and AJAX

Objective	This Lab will help you understand How to work with Web Optimization Using AJAX
Time	Mins

1. Observe the web application folder created for task in Lab 2 Question 1. You will find many script files and css files used by the application for its default templates. Use fiddler tool to observe the time taken to download these files. Implement bundling and minification and observe the difference in the time taken and number of files downloaded.
2. Add a new functionality to the application created in Lab 2 Question 1. The application provides a view which displays a list of categories available in the shop. When the user clicks on a category, the details of the products available under the category are displayed. Use AJAX to display the product details.