.NET

**E-Trading System (ERS)**

# INTRODUCTION

## SETUP CHECKLIST

**Minimum System Requirements**

* Intel Pentium 4 and above Windows 2007, 2008 and 2010
* Memory 4 GB
* Internet Explorer 8.0 or higher
* SQL Server 2012 or aboveclient and access to SQL Server 2012 or above server
* Visual Studio 2019 or onwards

## INSTRUCTIONS

* The code modules in the project should follow all the coding standards.
* Create a directory by your name in drive **<drive>**. In this directory, Store your Project here.
* You can refer to secured websites.
* You may also look up the help provided in the MSDN

# PROBLEM STATEMENT

## OBJECTIVE

Development of E-Trading System (ERS) using ASP.NET MVC, ASP.NET Web API & Entity Framework

**MODULE LIST**

**Modules:**

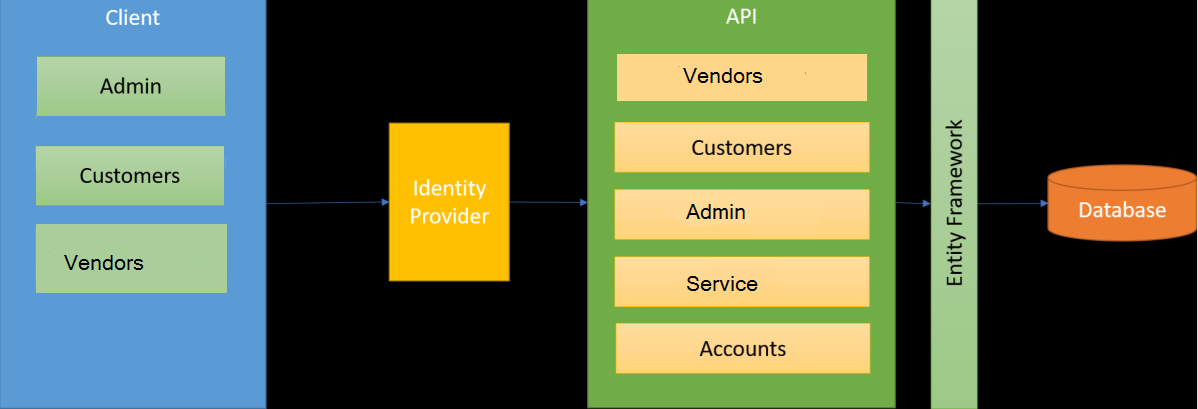
* Registration & Login
* Admin
* Customer
* Vendors
* Service
* Accounts
* Unit Testing

**REQUIREMENT**

You have to use:

* + ASP.NET MVC
  + Use Entity Framework instead of ADO.NET
  + Create Web API and host the Functionality
  + Unit Testing using Microsoft Test Framework

Create Client Application in ASP.NET MVC and Services using ASP.NET Web API



IMPLEMENTATION

**SUMMARY OF THE FUNCTIONALITY TO BE BUILT:**

The participants need to develop the E-Trading System by building the functionality incrementally

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Course** | **Duration** | **Functionality to be built** |
| **(in PDs)** |
| 1 | NET Framework 4.6 Entity Framework ASP.NET MVC ASP.NET Web API | 5 | Developing Presentation components (ASP.NET MVC, Entity Framework, ASP.NET Web API) |

**Abstract of the project**

The proposed **E-Trading System (ETS)** is a web based application which enables both the customers and vendors are logged into the application and view the showcased products and purchase them.

**E-Trading System (ETS)** system will have following features:

* Registration for Customers.
* Reports for Customers & Vendors
* Customers can check, different Products from Vendors.
* Buy and Selling of Products.
* Price alerts
* Customer account balance details.

Users of the System will be: **Admin, Customer,Vendor.**

**Role of Admin**

* Login to System
* Manage Customer & Vendors**.**
* Manage Business Category.
* He has the authority to add/delete users

**Role of Customer**

* Register & Login to System
* Check account balance.
* Can view & edit its own Profile.
* Can Search & view Products.
* Buy and Sell Products.

**Role of Vendors**

* Register & Login to System
* View and select Customers.
* Can view & edit its own Profile.
* Can change the price of Products.

**FUNCTIONAL COMPONENTS OF THE PROJECT**

Application Architecture:

Distributed web applications traditionally to be designed and built across three logical tiers:

* Database Access Layer (DAL)
* Business Logic Layer (BLL)
* Presentation Layer

The DAL refers to the database itself, the stored procedures, and the component that provides an interface to the database. The BLL refers to the component that encapsulates all the business logic of the application. And, the Presentation layer refers to the web application pages.

* Data access layer of 3-tier use ADO.NET data access using SQL stored procedures - All the database interaction would be performed using Data Access Component. Most common methods in Data Access Component would be –
  1. Create Connection to the Database
  2. Create Command Object
  3. Set Command Type to Stored Procedure
  4. Create and Populate Parameters
  5. Execute the Command
  6. Close the Connection