You may select any one project from the following optional projects, but not other non-related projects. The app can be Window form application, Asp.net Webform application, or Asp.net MVC application. You need to decide to save your data in flat files or a database.

The project will worth 40% as stated in the syllabus: 20% for the basic CRUD tasks, 20% for the class designation (being object oriented), architecture, deployment, coding details (exception handling, layout, comments etc.), and other features. Ensure that your names are clearly displayed on the first page of your app. It could include up to 3 students in a team. More than 3 in one team would get much lower scores.

Please submit your project files in a zipped file to the Turnitin Assignment on the black board site. If you deploy your project on Internet, I will verify and reward higher scores. The teams that can do the presentation and demonstrate your codes will also get better scores. Or a separated report file will also help.

**Copying projects will get 0 score.**

## Final Project 1- Used Vehicle Dealership

A Used Vehicle Dealership has asked you to develop an Application using C# that manages vehicles for their lot. The specific details of their request are listed below:

- Maintain a 4 primary lists:
  - Makes - Manages a list of vehicle Makes sold on the lots (Chevy, Ford, Volvo etc.)
  - Models - Manages a list of Models to choose from (F150, 1500, Trailblazer, etc.)
  - Vehicle Types - Manages a list of types of vehicle (4-door, Pick Up, Hatchback, 2-door etc.)
  - Vehicles on the lot - Manages a list of vehicles (sold and not sold) on the lot

The following table describes some of the features of the primary Objects in your applications:

| **Object Name / Properties** | **Description** | **Comments** |
|---|---|---|
| **Vehicle** | | |
| VehicleId | The unique identifier of the Vehicle | |
| MakeId | The Foreign Key related to the Make table | |
| ModelId | The Foreign Key related to the Model table | |
| Year | Year the vehicle was manufactured | An unsigned number |

| Object Name / Properties | Description | Comments |
|---|---|---|
| Price | Retail price of the vehicle | Displayed as currency |
| Cost | The cost of the vehicle when purchased | Displayed as currency |
| SoldDate | The date the vehicle was sold | In addition to this field, a calculated field name will be added to your Object that displays whet vehicle has been sold, this calculation is based null value in SoldDate field. If there is a value t vehicle has been sold. |
| **Make** | | |
| MakeId | The unique identifier of the Make | |
| Name | Name of the Make of the Vehicle (i.e. Chevy, Ford) | |
| **VehicleModel** | | |
| ModelId | The unique identifier of the Model | |
| Engine Size | The size of the engine in the vehicle | |
| Number of Doors | The number of doors on the vehicle | |
| Colour | The colour of the vehicle (Black, Red etc.) | |
| VehicleTypeId | The Foreign Key related to the VehicleType table | |
| **VehicleType** | | |
| VehicleTypeId | The unique identifier of the Vehicle Type | |
| Name | The name of the Vehicle Type | Compact, Sedan, Truck, etc. |

All Objects will inherit the following common features:

- CreateDate
- EditDate

You may choose to follow the Linked Resources videos to assist you to build your application.

Lesson Links

- MVC Web Application, Part 1 - http://youtu.be/oOzagc-6fyc
- MVC Web Application, Part 2 - http://youtu.be/oNUPGXIQFL4
- MVC Web Application, Part 3 - http://youtu.be/kJ8EUNPrQeQ
- MVC Web Application, Part 4 - http://youtu.be/G376fhOlQtY
- MVC Web Application, Part 5 - http://youtu.be/WBHw7FNEOkc

**Final Project 2 - Grid Game**

Develop a game that one player jumps in a maze grid of 4*4 (or more if you would like to try). This is testing your basic OO knowledge and skills. Please try to think about what fields, properties or methods should be implemented within the classes.

To start the game, a new user needs to input the name. The user should be able to save the game and resume the game to start next time. The game should show instructions (arrows to move etc.) and basic status such as the current score or blood (or health value) on labels. The grid of terrains should be generated randomly to start the game.

You need to have ideas of what type of traps and pitfalls, and/or what is the winner point.

Create classes such as: a player, a terrain, a play (session)

**Player** class member examples: fields of Name, Health, Strength, Methods of MoveTo (int direction) etc. The player can be shown as a simple icon or even a red point.

**Terrain** class members: ground (land, fire, water…), Coordinates. The objects could be shown as blocks of different colors.

**Play** class: organize the Players and Terrain classes, to allow the players save the progress and restart the game. You can save the data in a database or just flat files.

Resources links:

How to: Draw a Filled Rectangle on a Windows Form
https://docs.microsoft.com/en-us/dotnet/framework/winforms/advanced/how-to-draw-a-filled-rectangle-on-a-windows-form

How to: Draw an Existing Bitmap to the Screen
https://docs.microsoft.com/en-us/dotnet/framework/winforms/advanced/how-to-draw-an-existing-bitmap-to-the-screen

https://scratch.mit.edu/projects/77278452/