**Awesome Job! 900/900**

**This assignment is an individual assignment, collaboration is prohibited.**

**This assignment is similar to a HOT, where the Instructor will not provide assistance with coding.**

**\*\*\* Please make sure that you do the code along using this video. This will help with setting-up the Entity Framework, Database, implementing CRUD, etc…** [**https://www.youtube.com/watch?v=i9ILQK8cIRw**](https://www.youtube.com/watch?v=i9ILQK8cIRw)

**You may use your textbook, class handouts, lecture notes, and labs.**

**All code submitted must be your original work.**

* For this assignment you have the freedom to choose your own project with **instructor approval.**
* Your chosen project and feature list must be approved by your instructor.
* Make sure to spend adequate time polishing your app and fixing bugs. Focus on having an app that works well, looks nice, is user-friendly, and has all the important features.
* Consult with your instructor on a regular basis to get input on professionalism and polish.
* There are a few different deliverables required for this project.
  + Project Proposal Document (**final version**) due **Wednesday Dec 4th by midnight -** 100 points
    - Must use the provided template.
    - All requested information must be filled out.
    - Must include screen mockups for all features
  + Project Presentation due **Wednesday Dec 4th by midnight -** 300 points
    - Slides must be pre-approved by instructor due **Sunday Dec 1st by midnight**
    - Project Background
    - Explanation of Business Need
    - Purpose of Project
    - Brief Overview of Features
    - Brief Overview of Proposal
    - Live Demo of Website
  + Finished Project uploaded to GitHub by **Wednesday Dec 4th by midnight -** 900 Points
    - Code must be committed and pushed to git repo by **Wednesday Dec 4th by midnight** for grading and student must confirm that instructor can download the code
      * ~~Add jimk-developer as a Collaborator~~
* **Program must implement at least one custom class and object per UML diagram \_\_\_\_\_/200pts**
* Objects must be implemented as a foundational part of your solution
* Classes must demonstrate proper encapsulation per the UML diagram
* Must demonstrate one of the following class relationships
  + Inheritance, Aggregation or Composition
* **Solution must implement programming concepts including**

**variables, custom methods, branching and loops \_\_\_\_\_ / 100pts**

* **Solution must use arrays or collections to store and manipulate data\_\_\_\_\_ / 100pts**
* Must use a List<>
* **Solution must implement multiple windows\_\_\_\_\_ / 100pts**
* Data must be sent from one window to another
* **Key Features Implemented from Final Project Proposal: \_\_\_\_\_ / 100pts**
* Must use CRUD
* <Your features as stated in your final project proposal>
* <Your features as stated in your final project proposal>
* <Your features as stated in your final project proposal>
* **Implement Exception Handling for the following scenarios: \_\_\_\_\_ / 100pts**
* **Data Validation on CRUD operations**
* **Database errors - Try/Catch**
* **Microsoft SQL Server Database \_\_\_\_\_/100pts**
* At least one table of data operations.

**Create**

* Application allows user to Insert entities.

**Read (Search)**

* Application allows user to Search for entities.

**Update**

* Application allows user to Update entities.

**Delete**

* Application allows user to Delete entities.

**Professionalism and Polish \_\_\_\_\_/ 100pts**

* **Most valuable client features have been prioritized and are implemented in a bug-free and usable fashion.**
* Application does not contain any **typos or misspelled words.**
* All error messages are displayed in a user friendly manner.
* Use the main screen as a menu of sorts to guide users towards the primary functions of the application.