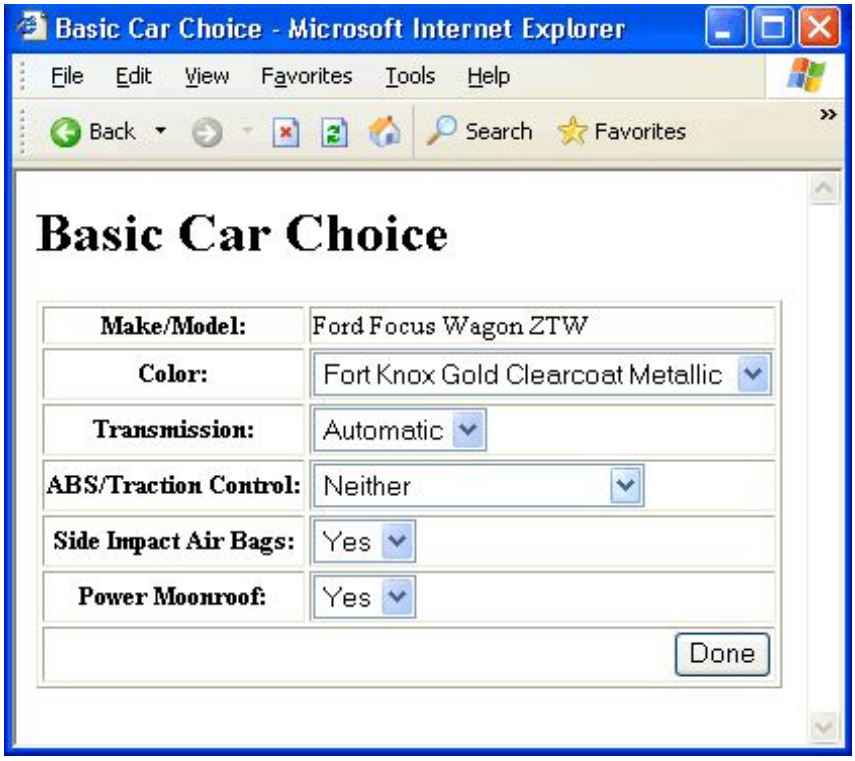
Assignment 6 - 40 points

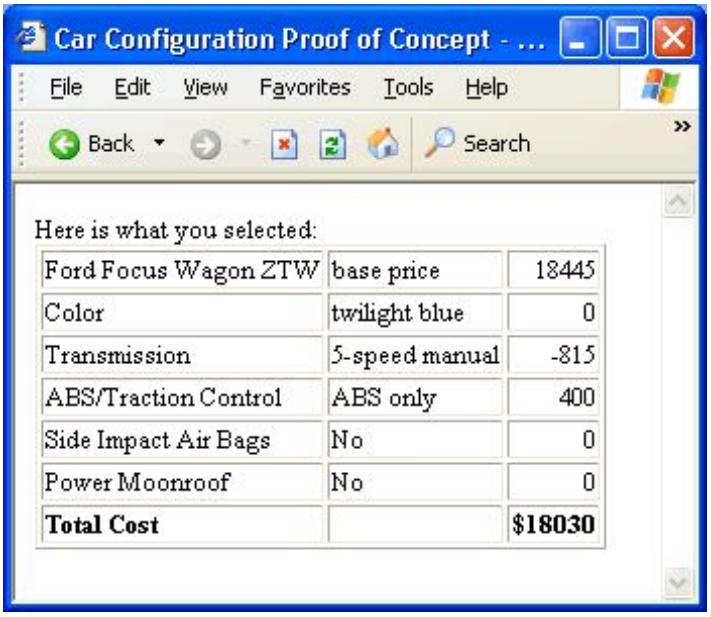
We now have a client-server system, where the end users (mostly non-programmers), can easily add new car models to the system and overwrite each other's data entry.

We now need to extend this system, so that it can be used over the web. You will need to extend the functionality of Client with the following features:

1. Show list of available models in a drop down box.
2. Upon selection of a Make/Model, the OptionSet and all related values load as shown below:



3. User selects an option and clicks on "Done".
4. User is directed to a page, which displays selected choices and total price for a vehicle.



Please use these screen shots as a design reference, on which to model the interface of your own page.

Our first goal is to develop the layout of the forms page, as well as to set up choices for users to select. Then, we need to update the page, to reflect choices made on the forms page.

Also, as you begin development, be sure to use the same configuration options that you used in previous unit. Specifically, I'm referring to the car choices like moonroof, color, transmission, brakes, and airbags.

Technical Concepts Applied

You have to design this unit with following technical requirements in mind:

1. Create a Servlet, which interacts with the Client (created in last unit), to get the list of available models.
2. Create another Servlet, which can interact with the Client (created in last unit), to get the data for the list of available OptionSets.
3. Create a JSP, which shows the OptionSets and prints the selected choices with total vehicle cost.

Please keep in mind that data displayed is dynamic (i.e. read from LinkedHashMap from the Server created in last unit).

Grading your Submission

1. Program Specification/Correctness (25 points)
    a. No errors, program always works correctly and meets the specification(s).

    b. The code could be reused as a whole or each routine could be reused.
    c. Design is reusable and extensible.
    d. Interfaces and/or abstract classes are applied between Servlets and Client Server interaction.
    e. Code is adequately tested and test runs are shown.

2. Readability (5 points)
    a. No errors, code is clean, understandable and well-organized.
    b. Code has been packaged and authored based on Java coding standards.

3. Documentation (5 points)
    a. The documentation is well written and clearly explains the functionality implemented by the code..
    b. Detailed class diagram is provided.

4. Code Efficiency (5 points)
    a. No errors; code uses the best approach in every case. The code is extremely efficient, readable and understandable.

Reclaiming lost points

If you lost points in Assignments 1 through 5 you should:

1. Modify your project and fix issues that were reported to you (i.e. for which points were taken off).
2. Create a change log that reflects following:
    a. What were you asked to change?
    b. How many points were taken off?
    c. What changes were made in what files?
    d. Show test cases for changes made (if applicable).
    e. In your opinion, how many points should be added back?

Lessons Learned - 50 points (Required) -

Please document the following:

- Things or concepts you learnt, – 50 points.
- Well-organized list of Design lessons learned from Project by participating in discussion board.