

Project - Unit 5

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** (created in Unit 4) with the following features:

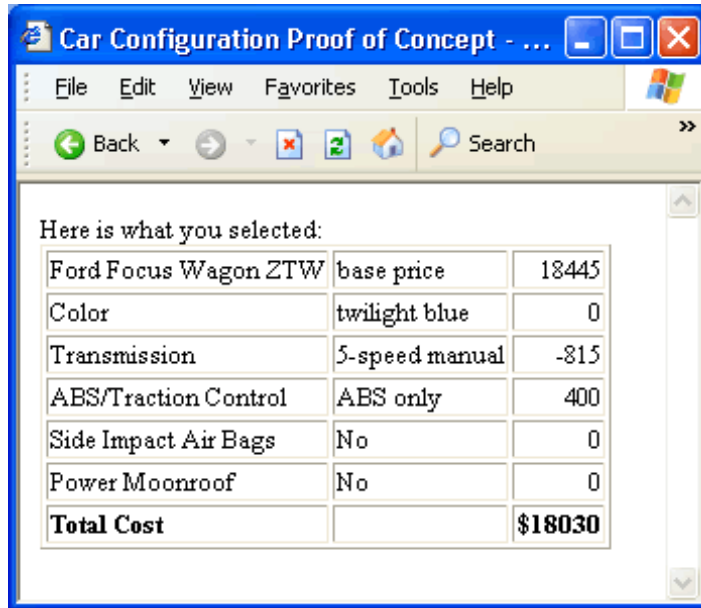
1. **Show list** of available models in a drop down box.
2. Upon selection of a Make/Model, the OptionSet and all related values load as shown below:

The screenshot shows a web browser window with the title "Basic Car Choice - Microsoft Internet Explorer". The browser's address bar is empty, and the main content area displays a form titled "Basic Car Choice". The form consists of several rows, each with a label and a dropdown menu. The labels are "Make/Model:", "Color:", "Transmission:", "ABS/Traction Control:", "Side Impact Air Bags:", and "Power Moonroof:". The dropdown menus contain the following values: "Ford Focus Wagon ZTW", "Fort Knox Gold Clearcoat Metallic", "Automatic", "Neither", "Yes", and "Yes". A "Done" button is located at the bottom right of the form.

Make/Model:	Ford Focus Wagon ZTW
Color:	Fort Knox Gold Clearcoat Metallic
Transmission:	Automatic
ABS/Traction Control:	Neither
Side Impact Air Bags:	Yes
Power Moonroof:	Yes

Done

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 Unit 4), to **get the list** of available models.
2. Create **another Servlet**, which can interact with the Client (created in Unit 4), 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 Unit 4).

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 for Unit 5.
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.