CSC 415/515 PROJECT 4
AJAX AND JSON PROJECT

PROF. GODFREY MUGANDA
DEPT OF COMPUTER SCIENCE

Write a web application that implements a website for an online used car dealership. The web application has two pages:

(1) An Administrator Page
(2) A Customer View Page

1. The Administrator Page

A web site administrator uses this page to upload information on a car available for sale. The page has a form that allows the administrator to post information on a car consisting of

(1) A short description for a car
(2) a file containing an image of the car
(3) a text file containing a full description of the car.

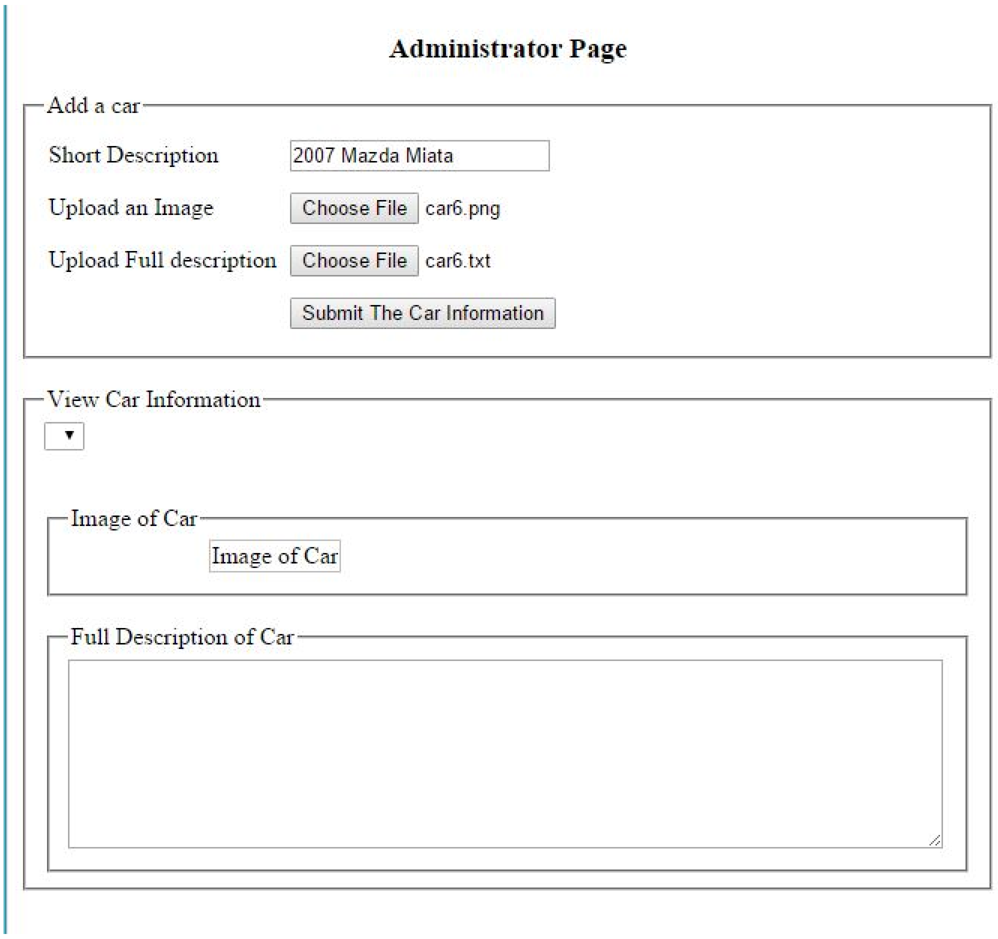
In addition to the form, the page has a series of HTML elements that allow the administrator to select any car among those already posted to the site and view an image of the car, the short description of the car, and full description of the car. This part must use AJAX and JSON to communicate with the server:

(1) Clicking on the `select` element results in an AJAX request being sent to the server. The response to the AJAX request is used to populate the options in the select element. In addition, completion of the document load will also trigger an AJAX call to populate the options of the `select` element.
(2) The `onchange` event on the `select` element results an AJAX call to fetch from the server the short description and the full description of the car just selected. In addition, an image of the newly selected car is displayed, but this image does not require and AJAX call: just set the `src` element of the image to the URL for the car's image.

Use the folder of images of cars delivered to you for the *Carsentration* project. Uploaded images will need to be stored in and `images` sub-folder in the `web` folder of the web application. Uploaded description files should be stored in the `WEB-INF` sub-folder of the web folder of the web application. Hints will be given in lecture as to how to do this.

Here are a few screen shots of the application, followed by the HTML code for the Administrator page (this will also be the welcome page).

The first shot shows data entry, just before posting to the server:


Administrator Page

Add a car

Short Description	2007 Mazda Miata
Upload an Image	Choose File car6.png
Upload Full description	Choose File car6.txt
	Submit The Car Information

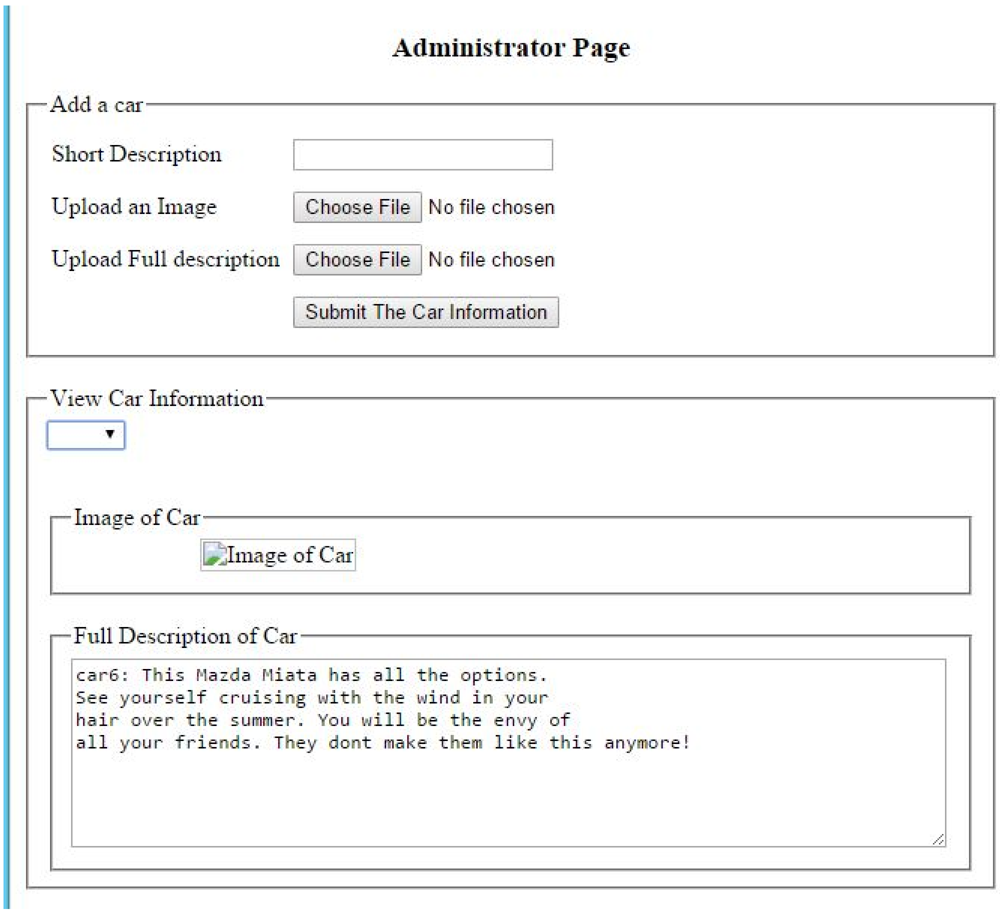
View Car Information

Image of Car

Image of Car

Full Description of Car

The second screen shot is after data for one car has been posted, and the administrator uses the `select` element to view the only available car. You can see the contents the uploaded file. Unfortunately, the image of the car disappears when I attempt to do a screen capture. A proper demonstration of how the web application should work will be given in class.



```
car6: This Mazda Miata has all the options.
See yourself cruising with the wind in your
hair over the summer. You will be the envy of
all your friends. They dont make them like this anymore!
```

Here is the HTML document that defines the user interface for the Administrator page.

```
<!DOCTYPE html>

<html>
    <head>
        <title>NCC Car Emporium Administrator Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
            body {width:650px;}
            input {margin:5px;}
            h3 {text-align: center;}
        </style>
        <script src="customer.js">
        </script>

    </head>
    <body>
        <h3>Administrator Page</h3>

        <form method = "POST" action="CarInfoServlet" enctype="multipart/form-data">
        <fieldset>
            <legend> Add a car</legend>
            <table>
                 <tr>
                    <td>Short Description </td>
                    <td><input type="text" id = "shortDescription" name =
                               "shortDescription" value="" />
                    </td>
                </tr>
                <tr>
                    <td>Upload an Image </td>
                    <td><input type="file" id = "imageFile" name="imageFile"  />
                    </td>
                </tr>
                <tr>
                    <td>Upload Full description </td>
                    <td> <input type="file" id = "descriptionFile"
                                name = "descriptionFile"  />
                    </td>
                </tr>
                 <tr>
                    <td> </td>
                    <td> <input type="submit" value ="Submit The Car Information"  />
                    </td>
                </tr>
            </table>
        </fieldset>
    </form>
    <br/>
    <fieldset>
    <legend> View Car Information </legend>
    <select id="availableCars" name="availableCars">
        <option></option>
        <option></option>
    </select>

    <br/>
     <br/>
     <br/>

    <fieldset>
    <legend> Image of Car </legend>
    <div style="width:400px;border:1px;">
        <div style="width:280px;  text-align:center;">
           <img id = "carImage" alt="Image of Car"  />
           <br/>
           <span id ="shortDescription">  </span>
        </div>
    </div>
    </fieldset>
    <br/>

    <fieldset>
```

```
    <legend> Full Description of Car </legend>
    <div style="width: 280px;  text-align:center;">
        <textarea id="fullDescriptionTextArea" rows="8" cols="80">
        </textarea>
        <br/>

    </div>
    </fieldset>

    </fieldset>
    <div id = "debug">

    </div>
    </body>
</html>
```

2. The Customer View Page

The customer view page is the same as the lower half of the Administrator page. It uses AJAX to allow the customer to select a car from the list of available cars and view its short description, full description, and image.

3. Additional Details

The names (keys) of the various cars will `car0` through `car27`. The full description text files will have the same names as the car, but with a `.txt` extension appended.

4. The Role Of JSON

The server should return JSON objects in response to AJAX call.

(1) When asked for the list of currently available cars to populate the select options list, the server will return an array of strings (names of the cars) in JSON form.
(2) When asked for the short description and full description of a given car, the server will return an array of two strings in JSON form. The string in position 0 of the array will be the short description, while the string in position 1 will be a single string that represents the contents of the description file.

5. Business / Domain Classes

Use the following class to represent information for a single car:

```
public class CarInfo
{
    public final String shortDescription;
    public final String carName;  // serves as prefix for the file name
    public CarInfo(String carName, String shortDescription)
    {
        this.shortDescription = shortDescription;
        this.carName = carName;
    }
    @Override
    public String toString()
    {
        String str =
        String.format("[car name : %s;  short description: %s]",
                      carName, shortDescription);
        return str;
    }
}
```

6. Due Date

This is due Monday of Week 9.