CarStoreApplication.java

```java
package com.cts.carstore;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.cts.carstore.skeletonvalidator.SkeletonValidator;

@SpringBootApplication
public class CarStoreApplication {

	public static void main(String[] args) {
		SpringApplication.run(CarStoreApplication.class, args);

		new SkeletonValidator();
	}

}
```

CarStoreController.java

```java
package com.cts.carstore.controller;

import java.util.ArrayList;
import java.util.List;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.validation.Validator;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;

import com.cts.carstore.exception.ApplicationException;
import com.cts.carstore.model.Car;
import com.cts.carstore.model.CarSearch;
import com.cts.carstore.service.CarStoreService;

@Controller
public class CarStoreController {

	@Autowired
	private CarStoreService service;

	@Autowired
	private Validator validator;
```

```
	public CarStoreController() {
		System.out.println("in default constr of controller");

	}

	@Autowired
	public CarStoreController(CarStoreService service, Validator validator)
{
		this.service = service;
		this.validator = validator;
	}

	@GetMapping("/showCarSearchForm")
	public String showCarSearchForm(@ModelAttribute("carSearch") CarSearch
carSearch) {
		// Add code here..
		carSearch = new CarSearch();
		return "carSearch"; // TODO, modify return value
	}

	@PostMapping("/getCarSearchResultPage")
	public String getCarSearchResultForm(@Valid @ModelAttribute("carSearch")
CarSearch carSearch, BindingResult result,
			ModelMap map) throws ApplicationException {

		// Add code here..
		if (result.hasErrors())
			return "carSearch"; // TODO, modify return value

		List<Car> carList = service.getCarSearchResult(carSearch);
		map.addAttribute("carList", carList);

		return "carSearchResult";
	}

	@ModelAttribute("cities")
	public List<String> populateCities() {
		List<String> cities = new ArrayList<String>();

		cities.add("Chennai");
		cities.add("Mumbai");
		cities.add("Delhi");
		cities.add("Bangalore");
		cities.add("Pune");
		cities.add("Kolkatta");

		return cities;
	}

	@ModelAttribute("brands")
	public List<String> populateBrands() {
		List<String> brands = new ArrayList<String>();

		brands.add("Maruti Suzuki");
```

```
		brands.add("Honda");
		brands.add("Mahindra");
		brands.add("Toyota");
		brands.add("Hundai");

		return brands;
	}

	@ModelAttribute("budgetUpto")
	public List<String> populateBudget() {
		List<String> budgetUpto = new ArrayList<String>();

		budgetUpto.add("Below 3 Lakh");
		budgetUpto.add("3 Lakh");
		budgetUpto.add("5 Lakh");
		budgetUpto.add("7 Lakh");
		budgetUpto.add("10 Lakh");

		return budgetUpto;
	}

	@ModelAttribute("fuelTypes")
	public List<String> populateFuelTypes() {
		List<String> fuelTypes = new ArrayList<String>();

		fuelTypes.add("Petrol");
		fuelTypes.add("Diesel");
		fuelTypes.add("CNG");

		return fuelTypes;
	}
}
```

ApplicationException.java

```
package com.cts.carstore.exception;


public class ApplicationException extends Exception {

	private static final long serialVersionUID = -9079454849611061074L;

	public String errorMessage;

}
```

ExceptionHandlerControllerAdvice.java

```
package com.cts.carstore.exception;



import java.time.LocalDateTime;

import javax.servlet.http.HttpServletRequest;

import org.springframework.http.HttpStatus;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.servlet.ModelAndView;



@ControllerAdvice
public class ExceptionHandlerControllerAdvice {

	@ExceptionHandler(ApplicationException.class)
	@ResponseStatus(value = HttpStatus.INTERNAL_SERVER_ERROR)
	public ModelAndView handleResourceNotFound(final ApplicationException
exception,
			final HttpServletRequest request, final Model model) {

		//Add code here..
		ModelAndView mv=new ModelAndView();
		mv.setViewName("error");
		mv.addObject("errorMessage","Low Budget-No car available below 3
lakh price");
		//add attribute error and timestamp
		mv.addObject("errorTime",LocalDateTime.now());

		return mv; //TODO, change return value
	}
}
```

Car.java

```
package com.cts.carstore.model;

public class Car {
```

```
	private String brandName;
	private String modelName;
	private double price;   //price in lakh
	private String fuelType;
	private double mileage;
	private int seatingCapacity;

	public Car() {
		// TODO Auto-generated constructor stub
	}

	public Car(String brandName, String modelName, double price, String
fuelType,double mileage, int seatingCapacity) {
		super();
		this.brandName = brandName;
		this.modelName = modelName;
		this.price = price;
		this.fuelType = fuelType;
		this.mileage = mileage;
		this.seatingCapacity = seatingCapacity;
	}

	public String getBrandName() {
		return brandName;
	}

	public void setBrandName(String brandName) {
		this.brandName = brandName;
	}

	public String getModelName() {
		return modelName;
	}

	public void setModelName(String modelName) {

		this.modelName = modelName;
	}

	public double getPrice() {
		return price;
	}

	public void setPrice(double price) {
		this.price = price;
	}

	public String getFuelType() {
		return fuelType;
	}

	public void setFuelType(String fuelType) {
		this.fuelType = fuelType;
	}
```

```
	public int getSeatingCapacity() {
		return seatingCapacity;
	}

	public void setSeatingCapacity(int seatingCapacity) {
		this.seatingCapacity = seatingCapacity;
	}

	public double getMileage() {
		return mileage;
	}

	public void setMileage(double mileage) {
		this.mileage = mileage;
	}




}
```

CarSearch.java

```
package com.cts.carstore.model;

import javax.validation.constraints.*;

public class CarSearch {
	// Use validation annotations as per the requirement
	@NotEmpty(message="Customer name is required")
	private String customerName;

	@NotEmpty(message="Gender is required")
	private String gender;

	@NotEmpty(message="Mobile number is required")
	@Pattern(regexp="[7-9]{1}[0-9]{9}",message="Mobile number should be 10
digits and starting with digit 7/8/9")
	private String mobileNumber;

	private String customerCity;
	private String brand;
	private String budgetUpto;
	private String fuelType;


	public CarSearch() {
		// TODO Auto-generated constructor stub
	}

	public String getCustomerName() {
```

```
		return customerName;
	}

	public void setCustomerName(String customerName) {
		this.customerName = customerName;
	}

	public String getGender() {
		return gender;
	}

	public void setGender(String gender) {
		this.gender = gender;
	}

	public String getMobileNumber() {

		return mobileNumber;
	}

	public void setMobileNumber(String mobileNumber) {
		this.mobileNumber = mobileNumber;
	}

	public String getCustomerCity() {
		return customerCity;
	}

	public void setCustomerCity(String customerCity) {
		this.customerCity = customerCity;
	}

	public String getBrand() {
		return brand;
	}

	public void setBrand(String brand) {
		this.brand = brand;
	}

	public String getBudgetUpto() {
		return budgetUpto;
	}

	public void setBudgetUpto(String budgetUpto) {
		this.budgetUpto = budgetUpto;
	}

	public String getFuelType() {
		return fuelType;
	}

	public void setFuelType(String fuelType) {

		this.fuelType = fuelType;
```

```
	}

}
```

ErrorResponse.java

```
package com.cts.carstore.model;


public class ErrorResponse {

	private String errorMessage;
	private String requestedURI;

	//add code here
}
```

CarStoreService.java

```
package com.cts.carstore.service;

import java.util.ArrayList;
import java.util.List;

import org.springframework.stereotype.Service;

import com.cts.carstore.exception.ApplicationException;
import com.cts.carstore.model.Car;
import com.cts.carstore.model.CarSearch;

@Service
public class CarStoreService {

	public List<Car> getCarSearchResult(CarSearch carSearch)
                                                        throws
ApplicationException {
		//Add code here..
		List<Car> list=new ArrayList<Car>();
		List<Car> available=buildCars();
		if(carSearch.getBudgetUpto().startsWith("Below"))
			throw new ApplicationException();

		int
budgetLimit=Integer.parseInt(carSearch.getBudgetUpto().substring(0,2).trim());
		for(Car car:available) {

	if(car.getBrandName().equalsIgnoreCase(carSearch.getBrand())&&

	car.getFuelType().equalsIgnoreCase(carSearch.getFuelType())
```

```
                                && car.getPrice()<=budgetLimit) {
                        list.add(car);
                    }
                }

                return list;  //TODO, modify this return value
        }


        // DO NOT CHANGE THIS METHOD
        //DO NOT CHANGE CODE WITHIN METHOD
        private List<Car> buildCars() {
                List<Car> cars = new ArrayList<Car>();
                // brand,modelName,price,fuelType,seatingCapacity
                Car car1 = new Car("Maruti Suzuki", "Swift", 5.20, "Petrol",
21.21, 5);
                Car car2 = new Car("Maruti Suzuki", "Alto", 3, "Petrol", 22.05,
5);
                Car car3 = new Car("Maruti Suzuki", "Ertiga LXi", 7.5, "Petrol",
19.01, 7);
                Car car4 = new Car("Maruti Suzuki", "Ertiga VXi", 8.9, "CNG",
26.2, 7);
                Car car5 = new Car("Maruti Suzuki", "WagonR", 4.5, "Petrol",
20.52, 5);
                Car car6 = new Car("Maruti Suzuki", "WagonR LXi", 5.20, "CNG",
32.52, 5);
                Car car7 = new Car("Honda", "Amaze", 6.20, "Petrol", 18.6, 5);
                Car car8 = new Car("Honda", "Amaze1.2E MT", 7.6, "Diesel", 24.7,
5);
                Car car9 = new Car("Honda", "Amaze1.5E MT", 8, "Diesel", 25.2, 5);
                Car car10 = new Car("Honda", "City", 9.30, "Petrol", 17.4, 5);
                Car car11 = new Car("Mahindra", "KUV100", 5.6, "Petrol", 15, 6);
                Car car12 = new Car("Toyota", "Yaris", 8.8, "Petrol", 18.1, 5);
                Car car13 = new Car("Hundai", "Aura", 7.30, "CNG", 20.5, 5);
                Car car14 = new Car("Hundai", "Creta", 9.9, "Petrol", 21.5, 5);
                cars.add(car1);
                cars.add(car2);
                cars.add(car3);
                cars.add(car4);
                cars.add(car5);
                cars.add(car6);
                cars.add(car7);
                cars.add(car8);
                cars.add(car9);
                cars.add(car10);
                cars.add(car11);
                cars.add(car12);
                cars.add(car13);
                cars.add(car14);

                return cars;
        }

}
```

SkeletonValidator.java

```
package com.cts.carstore.skeletonvalidator;

import java.lang.reflect.Method;
import java.util.logging.Level;
import java.util.logging.Logger;



public class SkeletonValidator {

	public SkeletonValidator() {
		validateClassName("com.cts.carstore.model.Car");
		validateClassName("com.cts.carstore.model.CarSearch");
		validateClassName("com.cts.carstore.model.ErrorResponse");
		validateClassName("com.cts.carstore.service.CarStoreService");

	validateClassName("com.cts.carstore.controller.CarStoreController");

	validateClassName("com.cts.carstore.exception.ApplicationException");

	validateClassName("com.cts.carstore.exception.ExceptionHandlerController
Advice");


		validateMethodSignature("getCarSearchResult:java.util.List",
				"com.cts.carstore.service.CarStoreService");

	validateMethodSignature("showCarSearchForm:java.lang.String,populateCiti
es:java.util.List,populateBrands:java.util.List,populateBudget:java.util.List,
populateFuelTypes:java.util.List",
				"com.cts.carstore.controller.CarStoreController");
	}

	private static final Logger LOG = Logger.getLogger("SkeletonValidator");

	protected final boolean validateClassName(String className) {

		boolean iscorrect = false;
		try {
			Class.forName(className);
			iscorrect = true;
			LOG.info("Class Name " + className + " is correct");

		} catch (ClassNotFoundException e) {

			LOG.log(Level.SEVERE, "You have changed either the " +
"class name/package. Use the correct package "
					+ "and class name as provided in the
skeleton");
```

```
		} catch (Exception e) {
			LOG.log(Level.SEVERE,
					"There is an error in validating the " +
"Class Name. Please manually verify that the "
							+ "Class name is same as
skeleton before uploading");
		}
		return iscorrect;

	}


	protected final void validateMethodSignature(String methodWithExcptn,
String className) {
		Class cls = null;
		try {

			String[] actualmethods = methodWithExcptn.split(",");
			boolean errorFlag = false;
			String[] methodSignature;
			String methodName = null;
			String returnType = null;

			for (String singleMethod : actualmethods) {
				boolean foundMethod = false;
				methodSignature = singleMethod.split(":");

				methodName = methodSignature[0];
				returnType = methodSignature[1];

				cls = Class.forName(className);
				Method[] methods = cls.getMethods();
				for (Method findMethod : methods) {
					if (methodName.equals(findMethod.getName()))
{
						foundMethod = true;

						if
(!(findMethod.getReturnType().getName().equals(returnType))) {

							errorFlag = true;
							LOG.log(Level.SEVERE, " You have
changed the " + "return type in '" + methodName
									+ "' method. Please
stick to the " + "skeleton provided");

						} else {
							LOG.info("Method signature of "
+ methodName + " is valid");
						}

					}
				}
				if (!foundMethod) {
```

```
                                    errorFlag = true;
                                    LOG.log(Level.SEVERE, " Unable to find the
given public method " + methodName
                                                + ". Do not change the " +
"given public method name. " + "Verify it with the skeleton");
                              }

                        }
                        if (!errorFlag) {
                              LOG.info("Method signature is valid");
                        }

                  } catch (Exception e) {
                        LOG.log(Level.SEVERE,
                                    " There is an error in validating the " +
"method structure. Please manually verify that the "
                                                + "Method signature is same as
the skeleton before uploading");
                  }
            }

}
```

Application Properties

```
server.port=3030
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
```

carSearch.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <br>
      <br>
      <!-- Add code here.. -->
      <h1 id="heading">Windsor Car Showroom</h1>

      <h3 style="align: center">Search Cars</h3>
      <sf:form name="form" action="getCarSearchResultPage"
            modelAttribute="carSearch" method="post">
            <sf:label path="customerName">Customer Name:</sf:label>
```

```
            <sf:input path="customerName" id="customerName" />
            <sf:errors path="customerName" cssStyle="color:red;" />
            <br />
            <sf:label path="mobileNumber">Mobile Number:</sf:label>
            <sf:input path="mobileNumber" id="mobileNumber" />
            <sf:errors path="mobileNumber" cssStyle="color:red;" />
            <br />
            <sf:label path="gender">Gender:</sf:label>
            <sf:radiobutton path="gender" value="Male" />Male
            <sf:radiobutton path="gender" value="Female" />Female
            <sf:errors path="gender" cssStyle="color:red;" />
            <br />
            <sf:label path="customerCity">Customer City:</sf:label>
            <sf:select path="customerCity" id="customerCity">
                  <sf:options items="${cities}" />
            </sf:select>
            <br />
            <sf:label path="brand">Brand:</sf:label>
            <sf:select path="brand" id="brand">
                  <sf:options items="${brands}" />
            </sf:select>
            <br />
            <sf:label path="fuelType">Fuel Type:</sf:label>
            <sf:select path="fuelType" id="fuelType">
                  <sf:options items="${fuelTypes}" />
            </sf:select>
            <br />
            <sf:label path="budgetUpto">Budget Upto:</sf:label>
            <sf:select path="budgetUpto" id="budgetUpto">
                  <sf:options items="${budgetUpto}" />
            </sf:select>
            <br />
            <input type="submit" value="CarSearch" />
            <input type="reset" value="Clear" />
      </sf:form>

</body>
</html>
```

carSearchResult.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
```

```
</head>
<body>

	<br>
	<br>
	<!--Add code here  -->
	<sf:form id="searchResult">
	<c:if test="${carList.isEmpty()==false}">
		<h3>Here are cars matching your search criteria: </h3>
		<table border="1">
			<tr>
				<th>Brand Name</th>
				<th>Model Name</th>
				<th>Price in lakh</th>
				<th>Fuel/Transmission</th>
				<th>Mileage</th>
				<th>Seating Capacity</th>
			</tr>
			<c:forEach items="${carList}" var="car">
				<tr>
					<td>${car.brandName}</td>
					<td>${car.modelName}</td>
					<td>${car.price}</td>
					<td>${car.fuelType}</td>
					<td>${car.mileage}</td>
					<td>${car.seatingCapacity}</td>
				</tr>
			</c:forEach>
		</table>
	</c:if>
	<c:if test="${carList.isEmpty()==true}">
		<h3 id="noResult">Sorry,No car available matching your
profile.</h3>
	</c:if>
	</sf:form>
	<br/>
	<a href="showCarSearchForm" id="searchCars">Search Cars</a>
</body>
</html>
```

Error.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1" isELIgnored="false" isErrorPage="true"%>

<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="sf" uri="http://www.springframework.org/tags/form"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<!DOCTYPE HTML>
```

```
<html>

<head>
<title>Car Search Errors</title>
</head>

<body>
       <!--Add code here..  -->
       <sf:form id="error">
       <h3>Unable to retrieve car information.Below are the error details:</h3>
       <h3>${errorMessage}</h3>
       <h3>${errorTime}</h3>
       </sf:form>
       <br/>
       <a href="showCarSearchForm" id="searchCars">Search Cars</a>
</body>
</html>
```