Cab Rent Estimator

Grade settings: Maximum grade: 100
Based on: Spring MVC METADATA
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 120 s **Maximum memory used**: 1.50 GiB **Maximum execution file size**: 1 GiB **Maximum number of processes**: 10000
Cab Rent Estimator

Ranbir owns a travel agency which provides Cab renting service on hourly basis within the city. For his customer's convenience, he needs an application which generates the estimated amount of cab rent based on the Cab Type and Duration in Hours.

Create a Spring MVC Spring Boot Application for developing a Cab Rent Estimator Application. Design a Cab Rent Estimator Page to choose an appropriate Cab Type **("Micro-NonAC", "Micro-AC", "Mini-NonAC", "Mini-AC", "Sedan-AC", "Luxury")** and enter the Duration in Hours.

On clicking Estimate Cab Rent button, the application should calculate the cab rent depending on the cab type chosen and the duration in hours which the cab needs to be rented. The customer has to then be redirected to estimatordesk.jsp page that displays the message **"Welcome to Ranbir Cabs. Your estimated cab rent is Rs. <<cab rent>>"**

Application Work Flow:

- EstimatorController is the Controller class.
- CabRentBean is the model class with three attributes cabType, rentPerHr and durationInHrs along with its getters and setters.
- CabRentService is the Service class which has a method called calculateCabRent that takes CabRentBean as its argument and returns double.
- This method needs to set the rentPerHr instance variable based on the Rent per Hour (in Rs) value given in the below mentioned table.
- Calculate the cabRent depending on the rentPerHr for the cabType chosen and the durationInHrs.
- cabRent should be returned as the double.

Cab Type	Rent per Hour (in Rs)
Micro-NonAC	500.0
Micro-AC	700.0
Mini-NonAC	800.0
Mini-AC	1000.0

Sedan-AC	1500.0
Luxury	2000.0

Initially, the customer should be routed via the EstimatorController's estimatorPage method to estimatorpage.jsp that allows user to choose the cabType and enter the duration in hours.

[Note: estimatorPage method has to be written inside the EstimatorController]

A method in the EstimatorController known as buildState should be annotated with the ModelAttribute **"cabList"**. This method should populate the cab types (Micro-NonAC, Micro-AC, Mini-NonAC, Mini-AC, Sedan-AC, Luxury) in the Map as key-value pair. Both key and value be the same cabType. (Example: Key- Micro-NonAC, Value- Micro-NonAC, Key- Micro-AC, Value- Micro-AC, etc) and then return the Map. This should be then used to autopopulate the cabType in the estimatorpage.jsp

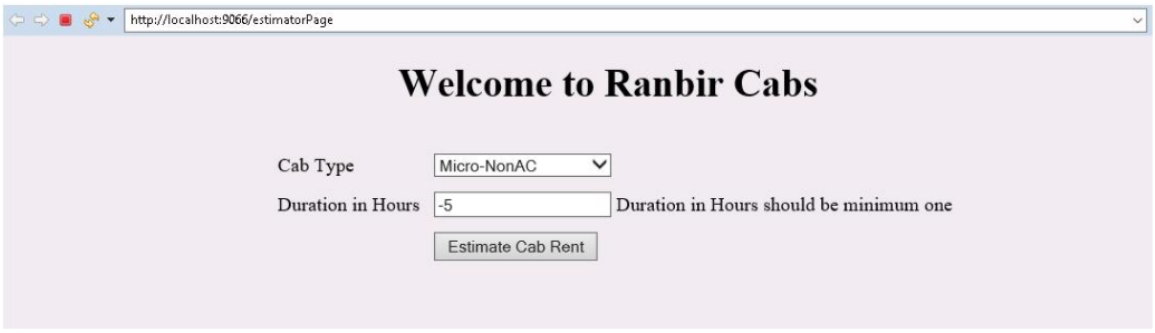
[Note: buildState method should be written inside the EstimatorController]

On clicking the Estimate Cab Rent button, the EstimatorController's calculateCabRent method should be called. This method takes three arguments - model attribute named "cab" which holds the form populated CabRentBean Object, BindingResult and the ModelMap.

- This method should calculate the cabRent by invoking the calculateCabRent method of the CabRentService.

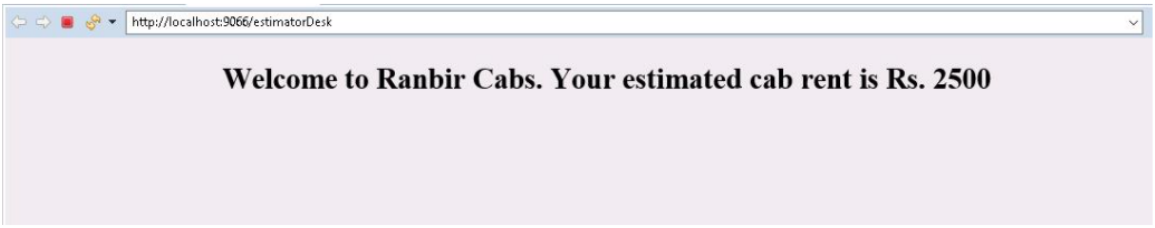
Screen designs and Expected Output:

If the durationInHrs entered by the customer is less than 1 then an error message **"Duration in Hours should be minimum one"** has to be displayed in the estimatorpage.jsp. To do this validation use appropriate annotation above **durationInHrs** attribute in **CabRentBean** model class.



- Redirect the user to **estimatordesk.jsp** page with a message **"Welcome to Ranbir Cabs. Your estimated cab rent is Rs. <<cab rent>>"**

estimatordesk.jsp



Design Constraints:

UI Design Constraints:

estimatorpage.jsp		
Component	**ID**	**Constraints**
Select	cabType	Should be auto populated using the model attribute written above the buildState method inside the EstimatorController. Do not hard code the values
Textbox	durationInHrs	*Should not be less than 1*
submit	submit (Name)	-

Note: In **estimatordesk.jsp**, the Result has to be rendered in the **<h2>** tag

Component Specification

Controller

EstimatorController			
AttributeName	**AttributeType**	**Access Specifier**	**Constraints**
cabRentService	CabRentService	private	Use annotation to Autowire

EstimatorController			
Method Name	**Method Argument name:type**	**Return type**	**RequestMapping URL & Request Method**
estimatorPage	modelAttribute "cab":CabRentBean	String	/estimatorPage & GET
calculateCabRent	modelAttribute "cab":CabRentBean, result:BindingResult, model:ModelMap	String	/estimatorDesk & POST
buildState		Map <String,String>	Should be annotated with ModelAttribute with name "cabList"

Service:

CabRentService		
Method Name	**Method Argument name:type**	**Return type**
calculateCabRent	cabRentBean:CabRentBean	double

Model:

CabRentBean	
AttributeName	**AttributeType**
cabType	String
rentPerHr	double
durationInHrs	int

Overall Design Constraints:

- EstimatorController should be inside the package com.controller
- CabRentService should be inside the package com.service
- CabRentBean should be in the package com.model
- Use appropriate annotation to configure CabRentService as a Service
- Use appropriate annotation to configure EstimatorController as a Controller
- CabRentService should be autowired inside the EstimatorController.
- Use annotations to implement the business Validation as specified in the screen shot [That is, when the duration in hours is less than 1 then error message **"Duration in Hours should be minimum one"** should be rendered in the UI]

- Do not change the property name given in the application.properties files, you can change the value and you can include additional property if needed.
- In the pom.xml you are provided with all the dependencies needed for developing the application.
- You will not be evaluated based on the UI design (layout, color, formatting, etc.). You are free to have a basic UI with all the required UI components (input fields, buttons, labels, etc.). Expected components with the id alone should be designed as per the requirement.
- Adhere to the design specifications mentioned in the case study.

- **Do not change or delete the class/method names or return types which are provided to you as a part of the base code skeleton.**
- Please make sure that your code does not have any compilation errors while submitting your case study solution.