Task By Task Guide

TASK 1: Create a responsive visual layout

Create a basic HTML page including a <section> or a <div> containing the theater/cinema seat map. To create a responsive visual layout, you'll need to consider different screens the users may view the page on, the appearance of the seats grid, and the different states seats can be in (e.g. available, reserved etc.). On smaller screens, it will be hard to view the entire seat map: consider how you want to display the seat map once the screen size reaches a certain limit, for example you may divide the seat map into sectors/areas the users can zoom-in one sector at a time on smaller screens. Create a temporary visual layout for each different stage you have thought of, which can act as the design you will then generate in JavaScript.

TASK 2: Create the seat/area map structure in Javascript

Following the creation of your visual layout for desktop and mobile, it's time to make Javascript create these layouts dynamically. In order to do this, you should think about an appropriate data structure, for example in JSON, that can be used to create all the different layouts you have created (for example, a full seat map, seat areas, zoomed-in areas). You can either hard code the structure, or generate it using an algorithm. Consider adding by default a number of occupied seats, if you generate the JSON algorithmically consider using Math.random(). Make sure to include a unique ID for each seat.

TASK 3: Switch between the desktop seat map and the areas mobile layout

At this point, you will dynamically generate via JavaScript the visual layout you have previously created in the first task. Consider having two different functions that read your JSON data and write the appropriate DOM on the page, each function will be called depending on the current screen size. During this task and the next one, once you are able to dynamically output the same HTML that you had previously created on the page, you can delete the temporary static visual layout you have created earlier. Consider using window.matchMedia() to handle the different screen sizes and jQuery to update the DOM.

TASK 4: Create the seat map detail of each seat area

For the last step of the layout, you want to listen to clicks on a specific seat area, which triggers another function that shows the seat map of that specific seat area. You also want to remember to insert a button to go back to the seat area map.

TASK 5: Select a seat and send the data back to the server

The final piece of user interaction is the actual selection and booking of the seats. At this stage, create a function to handle the seat selection: the function will change the class of any clicked seat, using the CSS created in Task 1. Finally add a “reserve seat” button. The “reserve seat” button should be available when the seat map is visible on screen, so within the desktop layout or on mobile after the seat area has been expanded and individual seats are visible. Consider hiding or disabling the button if seats are not visible or no seat is selected, as an additional feature. When the button is clicked, data will be sent back to the server, and the booked row/seat numbers will be displayed back to the user. In this project, as you are only developing a front end user interface, simply display the information on screen.