

IT5007: Software Engineering on Application Architecture
Tutorial-3: Hotel California Waitlist System Enhanced Version

Deadline: 3rd Oct. 23:59 PM

Points: 20

You have been nominated for the best employee of the year award for designing the waitlist system. However, the front desk feels that your waitlist system could use some improvements, with which you stand a great chance of winning the award. Your task is to improve the offline system to an online web application with the long-term goal of supporting better responsiveness. In the IT5007 module, you have learned that React is the best way to go about doing this. To win this award, you must provide two specific improvements:

(1) Make the system online with a server-client architecture, wherein the hotel front desk is the client who uses the browser and the server (for testing) is the docker container on your machine. [Hint: You can use the MERN stack setup discussed in class (Refer Chapter 2 of textbook)]. [7 points]

(2) Rewrite your front-end (user interface) code using React. Specifically, use React components to modularize your code. You can skip implementing logic that requires asynchronous events (e.g., button click). Instead, use a static array with a hard-coded list of waitlisted customers. [Hint: You can use the simple React component examples in Chapter 3 of textbook for reference.] [13 points]

The front-end requires the following changes:

- 1) Disable all functionalities that require a button press. (e.g., Just remove the buttons)
- 2) Create the following React components: displayHomepage, addCustomer, deleteCustomer, displayCustomers, displayFreeSlots. You can create more components if you need. For example, you may want a dummy top level component that contains the aforementioned components. You may also want a component for building the table. (refer Chapter 3 of textbook)
- 3) Unconditionally display all the components all the time (i.e., no button press).
- 4) Use hard-coded customer information. This means your addCustomer and deleteCustomer logic will not work yet. Also, the number of free slots will remain constant. You will code the rest of the logic in Tutorial 4.
- 5) Testing: You need at least 2-3 customer details (which you hard coded) to be displayed on the web application. There is nothing else to check at this point. But remember, "Rome wasn't built in a day." We will test the rest of the functionality in Tutorial 4.

Submission details:

- 1) Submission is through Luminus. The folder will be available soon.
- 2) Files: Submit just one zip file. Name the zip file with your matric number and name.
 - a. Submit your source files: html, js, css. Only one file each allowed. Justify if you need to add more files.
 - b. Submit a small one page report with 2 subheadings: HTML, JS, CSS. List all important features you have implemented under each category.
- 3) No bonus points for this tutorial!