App Implementation :

You will be building a small **React application** that allows a **parking warden to manage a car parking lot**.
The application will manage **parking space allocation**, **parking charge calculation** and **parking space deallocation**.

Functional requirement :

1. To start with the user will be asked to enter the number of spaces in the parking lot.
2. The app will then proceed to draw the parking spaces available in the parking lot - each parking space will show a unique parking space number on it.
3. The user should be able to add a new car registration and parking time when a car arrives.
4. The app should allocate a random space if available.
5. If no lot is empty then show a toast message that parking is full.
6. On exit of the car, the user should be able to press the parking space to open a new page.
7. This modal should display the calculated parking charges and also a button to deallocate the parking space once payment has been taken.
8. The new page should have a back button in case the user changes their mind.
9. Parking charges are calculated as follows : First 2 hours : $10.
10. $10 will be charged for every consecutive hour then after.
11. When the payment taken button is clicked, Make a POST request to
    https://httpstat.us/200 with the body
    { "car-registration":<car_registration_string>,"charge":
    <charge_amount_numbers> }

    Example: { "car-registration": "TU68 0BB", "charge": 20 }

Technical Requirements :

1. Written and typed with**TypeScript**.
2. Use a state management library (**Redux**, **React.Context**)

3. Styled and formatted appropriately so that the application is easy to use.
4. Use the **https:// material-ui.com/** component library for your components.
5. Write **JEST unit tests** that provide **80% or more test coverage** of your components.
6. No payment gateway is required - the button to indicate 'Payment taken' is enough.
7. Good coding practices are required - your code quality will be reviewed.
8. The following id attributes need to be added to your components (**Mandatory**).

Context	Element	id
Parking lot creation	Number of spcaes Text Input	parking-create-text-input
Parking lot creation	Submit button	parking-create-submit-button
Parking lot drawing	Each empty parking space	parking-drawing-space-<space_number>
Parking lot drawing	Each parking space's unique number	parking-drawing-space-number <space_number>
Parking lot drawing	Car Registration Text Input	parking-drawing-registration-input
Parking lot drawing	Car Registration Submit button	parking-drawing-add-car-button
Parking lot drawing	Each parking space with car registered	parking-drawing-registered-<space-number>
Car deregister page	Car registration	deregister-car-registration
Car deregister page	Time spent in parking lot	deregister-time-spent
Car deregister page	Parking charge	deregister-charge
Car deregister page	Payment taken button	deregister-payment-button
Car deregister page	Back button	deregister-back-button