Part 1

Test suite (test cases are ordered by business priority from top to bottom)

- Verity the restaurants list is returned without error
- Verify the restaurants returned in correct order (ex. the default order is best match)
- Verify each restaurant (restaurant card) has correct information (ex. restaurant name, customer logo, default logo, reviews, type, time, delivery charge, min fee etc.)
- Verify clicking on one restaurant in the list redirects to restaurant subpage (or restaurant card is clickable)
- Verify the restaurants in returned list all have same postal code (can be verified by backend test)
- Verify the restaurants order changes when user changes the sorting preference
- Verify if any one of the restaurants returns error it doesn't fail the whole page
- Verify when one restaurant's information is updated, the list gets updated with the new information (with or without refresh depends on frontend tech)
- Verify in case there is no matching restaurant in this postal code area, display a graceful error message
- Verify when searching time changes the returned list includes "closed for delivery list" section
- Verify restaurant card display information according to user's language settings
- Verify the number of returned restaurant is as expected
- Verify restaurants with other postal code than searching postal code don't show up in the returned list
- Verify scrolling down the page to the end, the page can display/load restaurant cards

Smoke testing suite:

- Verity the restaurants list is returned without error
- Verify each restaurant (restaurant card) has correct information (ex. restaurant name, customer logo, default logo, reviews, type, time, delivery charge, min fee etc.)
- Verify clicking on one restaurant in the list redirects to restaurant subpage (or restaurant card is clickable)
- Verify the restaurants in returned list all have same postal code (can be verified by backend test)
- Verify if any one of the restaurants returns error it doesn't fail the whole page
- Verify restaurant card display information according to user's language settings
- Verify the number of returned restaurant is as expected

Part 2

Reason to automate the tests:

Tests that are tedious to check by manual while by automation is simple, ex get restaurants list returns 200;
Tests can be reusable, ex restaurant card information checking;
Tests can be covered thoroughly when automation, ex check postal codes from restaurants list are the same as user input

Automaton tests:

- Verity the restaurants list is returned without error
    - Given user input a valid postal code
    - When user clicks on search button
    - Then backend response with 200

- Verify the restaurants returned in correct order (ex. the default order is best match)
    - Given user input a valid postal code
    - When user clicks on search button
    - And the returned restaurant list is returned in matching point from high to low

- Verify scrolling down the page to the end, the page can display restaurants card
    - Given user input a valid postal code and clicks on search button
    - When user scroll down several times on the restaurants list
    - Then user can see restaurant card displayed on the page

- Verify clicking on one restaurant in the list redirects to restaurant subpage (or restaurant card is clickable)
    - Given user input a valid postal code and clicks on search button
    - When user clicks on the first restaurant card
    - And it redirects to this restaurant page
    - Then the page contains restaurant name as in the card

- Verify each restaurant (restaurant card) has correct information (ex. restaurant name, customer logo, default logo, reviews, type, time, delivery charge, min fee etc.)
    - Given user search on a valid postal code
    - When user gets information on the restaurant card
    - Then user sees restaurant name
    - And user sees restaurant logo (custom one or maybe a default logo)
    - And user sees restaurant review stars and number of reviews
    - And user sees restaurant type
    - And user sees delivery time, charge and mim fee information

- Verify the restaurants in returned list all have same postal code (can be verified by backend test)
    - Given user search on a valid postal code
    - When user gets restaurant list
    - Then all restaurant postal code is the same as user input

- Verify when one restaurant's information is updated, the list gets updated with the new information (with or without refresh depends on frontend tech)
    - Given one restaurant information is update
    - When user search for a valid postal code include this restaurant
    - Then user should see the updated information from this restaurant card

- Verify in case there is no matching restaurant in this postal code area, display a graceful error message
    - Given there is no restaurants in this area
    - When user search for the postal code
    - Then it displays error message on the page (or an error message from the backend)

- Verify when searching time changes the returned list includes "closed for delivery list" section
    - Given searching time is 9 pm local time
    - When user search for a valid postal code
    - Then use sees closed for delivery list section
    - And there are restaurant cards in this section

- Verify the number of returned restaurant is as expected
    - Given user search on a valid postal code
    - When user gets restaurant list
    - Then the returned restaurant number is as expected

- Verify restaurants with other postal code than searching postal code don't show up in the returned list
    - Given user search on a valid postal code
    - When user gets restaurant list
    - Then no restaurants with other postal code appears in the list