

Hackathon Day 5

Testing, Error Handling, and Backend Integration Refinement

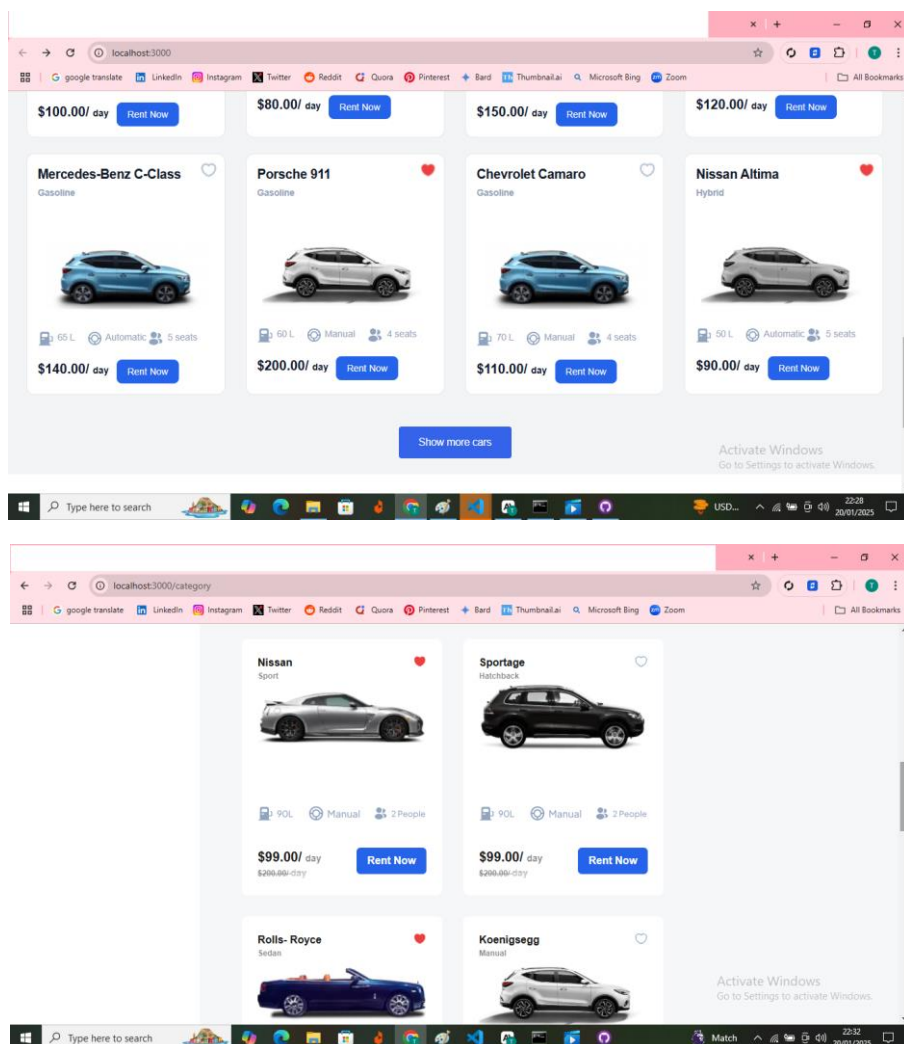
Objective:

The marketplace for deployment by ensuring all components are thoroughly tested, optimized for , and refined to handle customer traffic effectively.

1: Functional Testing of a Rental E-Commerce Platform

Product Listings:

- Display of rental items (images, descriptions, prices, availability).
- Test filtering and sorting options (e.g., category, price range, location).
- Validate search functionality with keywords.



Rental Booking Process

- Test date selection for availability.
- Verify accurate calculation for rental duration and additional fees.
- Test payment gateway integrations (e.g., Stripe, PayPal).
- Ensure rental confirmation notifications (email/SMS) are sent.

The image displays two screenshots of the MORENT car rental application. The top screenshot shows the 'payment' page, which is divided into 'Billing Info' and 'Rental Summary' sections. The 'Billing Info' section (Step 1 of 4) contains input fields for Name, Phone Number, Address, and Town/City. The 'Rental Summary' section shows a 'Nissan GT - R' car with a subtotal of \$80.00, tax of \$0, and a total rental price of \$80.00. The bottom screenshot shows the 'admin' page, which includes a sidebar with navigation options like 'Calendar', 'Preferences', and 'Main Menu'. The main content area displays a 'Nissan GT - R' car with a price of \$9751, a 'Pick-Up' section with location, date, and time selectors, a 'Drop-Off' section with similar selectors, and a 'Total Rental Price' of \$80.00. A 'Recent Transaction' table lists four transactions for the same car model on 28 and 29 July, each priced at \$80.00. The footer of the admin page includes links for 'About', 'Community', and 'Socials'.

payment

Billing Info
Please enter your billing info
Step 1 of 4

Name: Your name
Phone Number: Phone number
Address: Address
Town/City: Town or city

Rental Summary
Prices may change depending on the length of the rental and the price of your rental car.

Nissan GT - R
★★★★☆ 440+ Reviewer

Subtotal: \$80.00
Tax: \$0
Apply promo code:

Total Rental Price \$80.00
Overall price and includes rental discount

admin

Nissan GT - R
Sport Car #9751

Pick-Up
Locations: Select your city
Date: Select your date
Time: Select your time

Drop-Off
Locations: Select your city
Date: Select your date
Time: Select your time

Total Rental Price \$80.00
Overall price and includes rental discount

Recent Transaction

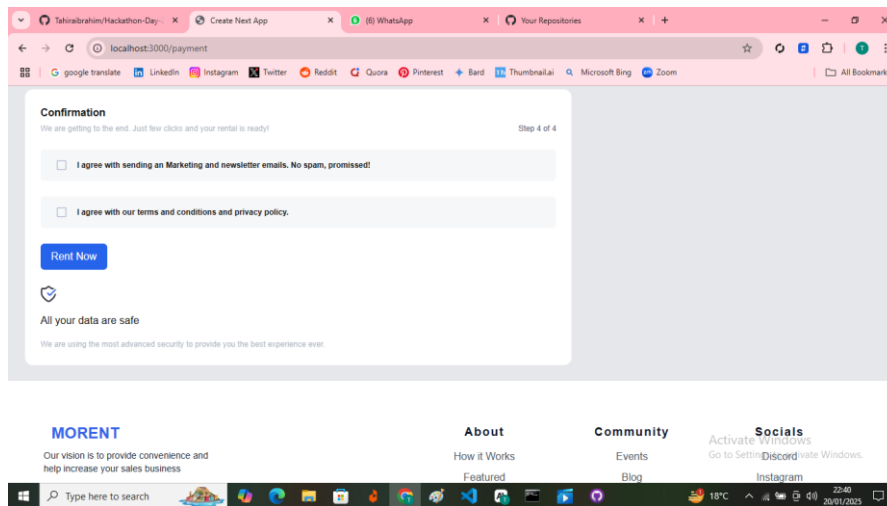
Car Model	Date	Price
Nissan GT - R Sport Car	28 July	\$80.00
Nissan GT - R Sport Car	29 July	\$80.00
Nissan GT - R Sport Car	29 July	\$80.00
Nissan GT - R Sport Car	29 July	\$80.00

MORENT
Our vision is to provide convenience and help increase your sales business

About
How it Works

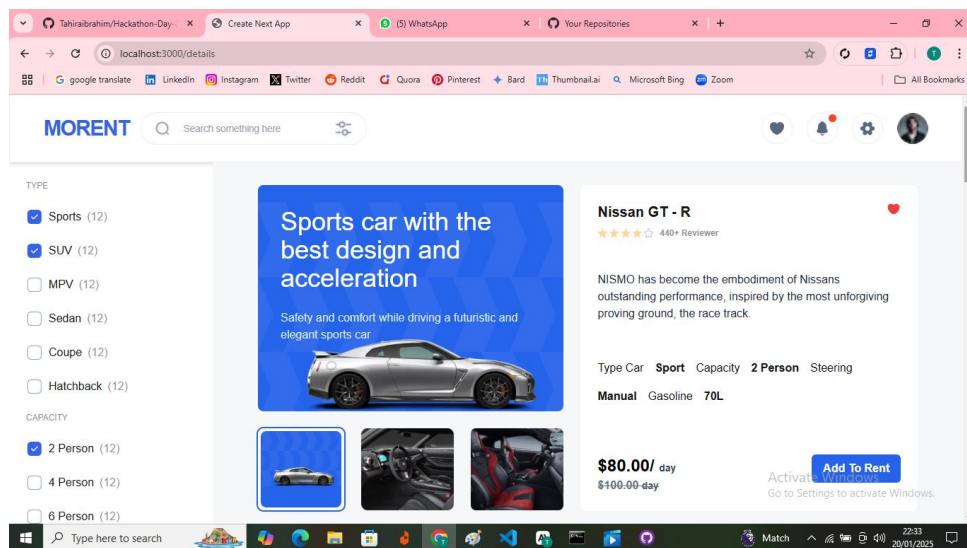
Community
Events

Socials
Go to Settings to activate Windows. Go to Settings to activate Windows. Go to Settings to activate Windows.



Cart Functionality:

- Add, update, and remove items in the cart.
- Verify cart retention after logging out and back in.

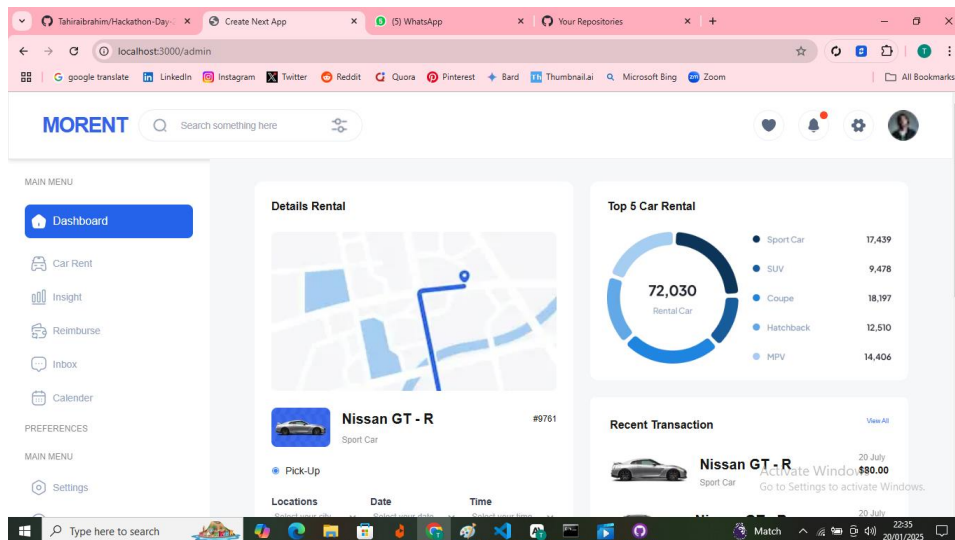


Dynamic Routing:

- Test navigation to individual product pages.
- Verify proper loading of product-specific data.

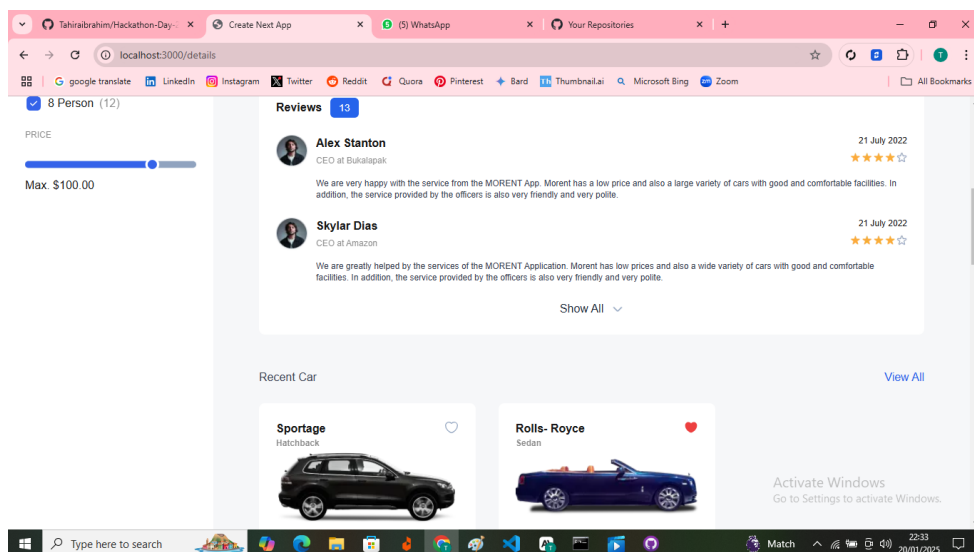
Admin Dashboard (if applicable):

- Add, update, and delete rental listings.
- View user bookings and manage inventory.



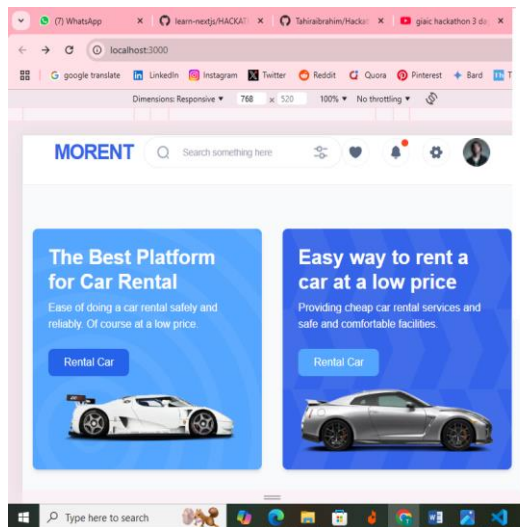
User Reviews and Ratings:

- Submit reviews and ratings for rented items.
- Validate review display on product pages.

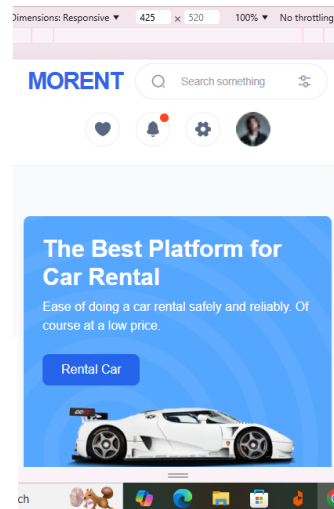


Responsive

Tablet



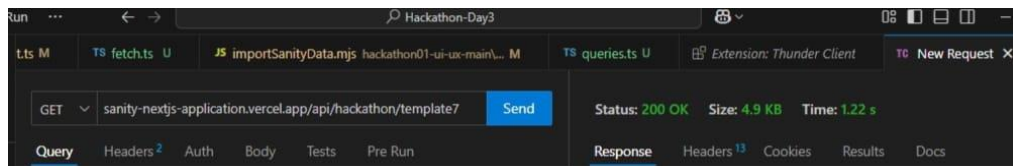
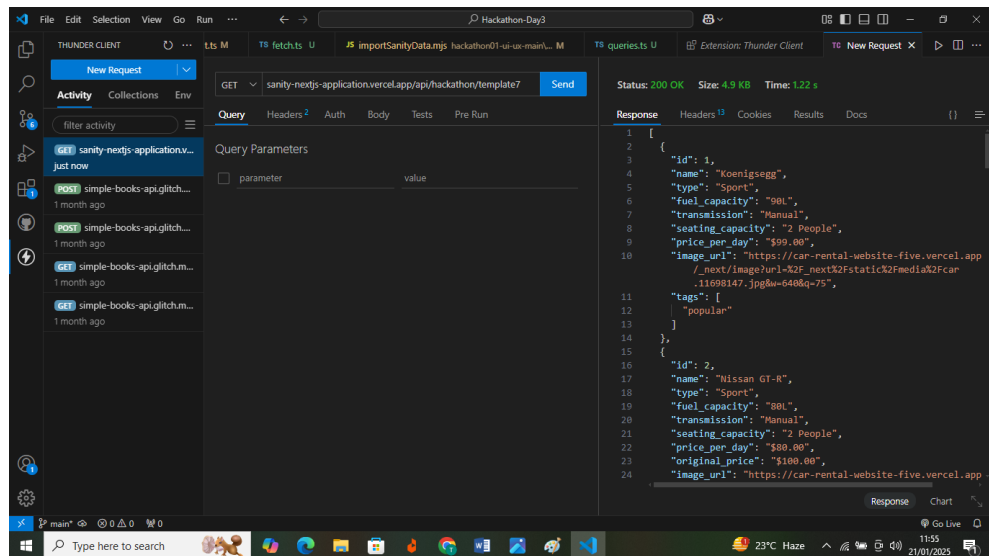
Mobile



2: Thunder Client API:

sanity-nextjs-application.vercel.app/api/hackathon/template7

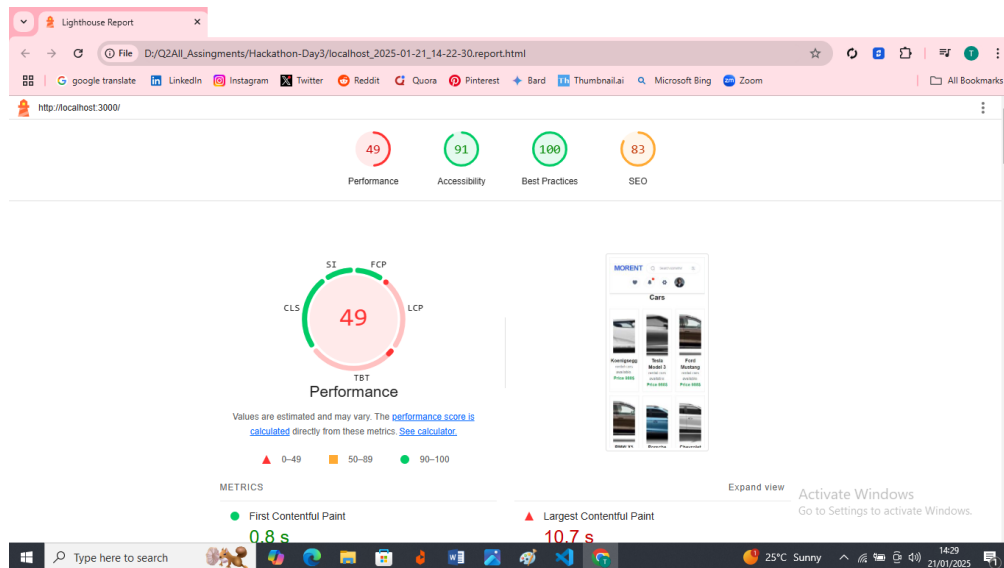
- My API is working perfectly, and my data is being displayed correctly.
- Additionally, proper error messages are showing in my API.
- However, there were some issues in the Day 4 project, but I have resolved them.
- Tool Thunder Client



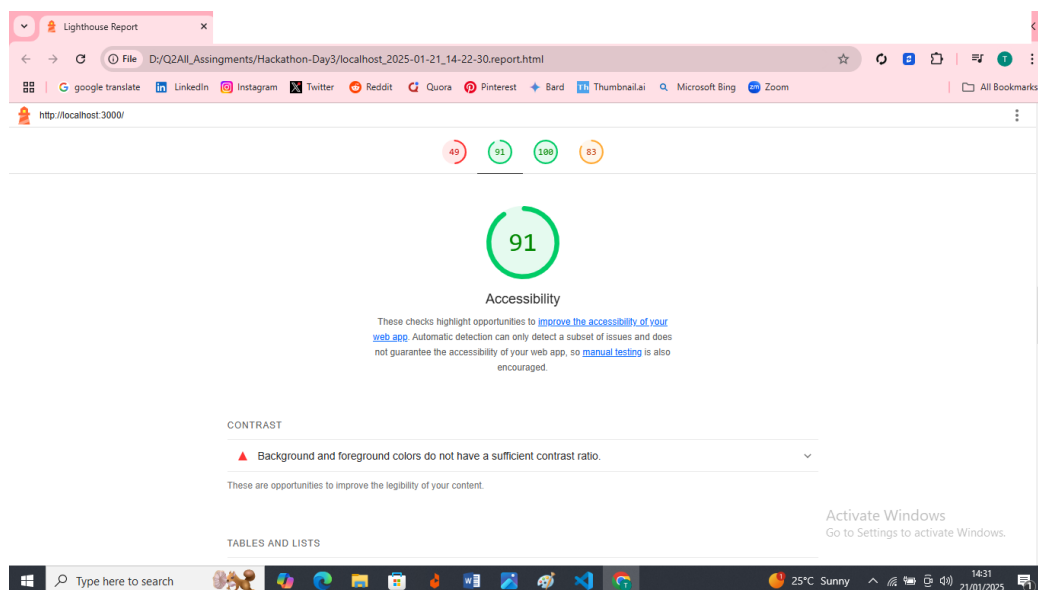
Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
2	TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	High	- No issues found
3	TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	- Handled gracefully
4	TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	High	- Works as expected
5	TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	High	- Test successful

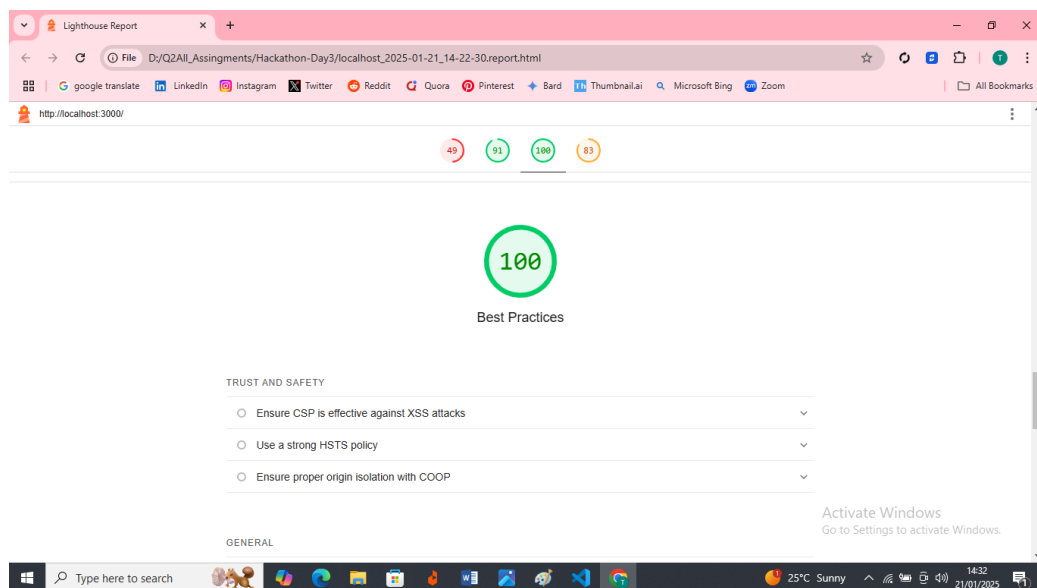
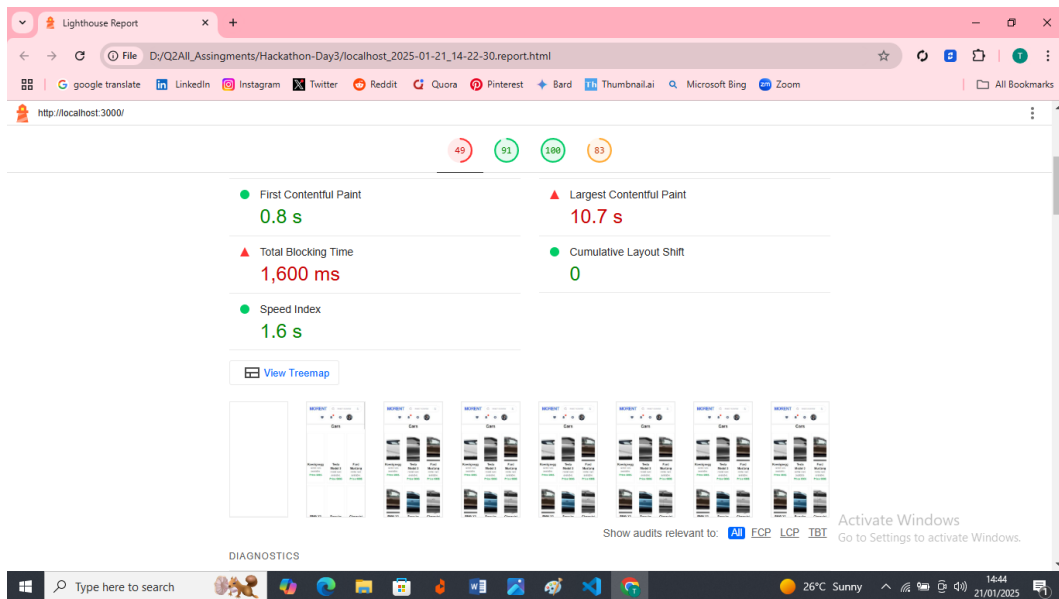
3: Performance Optimization

- npm install -g lighthouse
- lighthouse <http://localhost:3000> --view
- Values are estimated and may vary. The performance score is calculated directly from these metrics.



- These checks highlight opportunities to improve the accessibility of your web app.
- Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app,
- so manual testing is also encouraged.

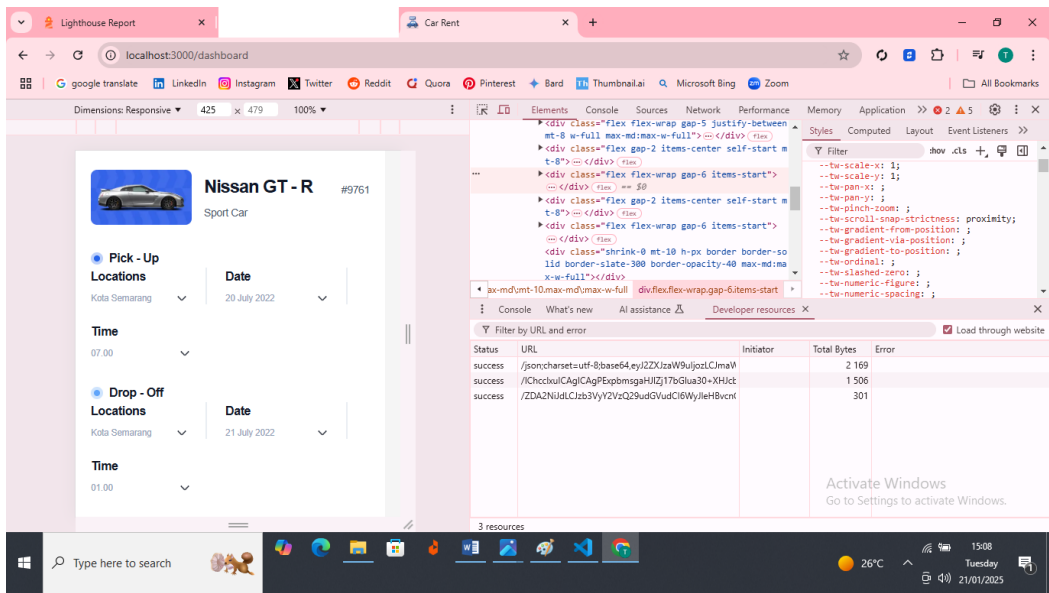




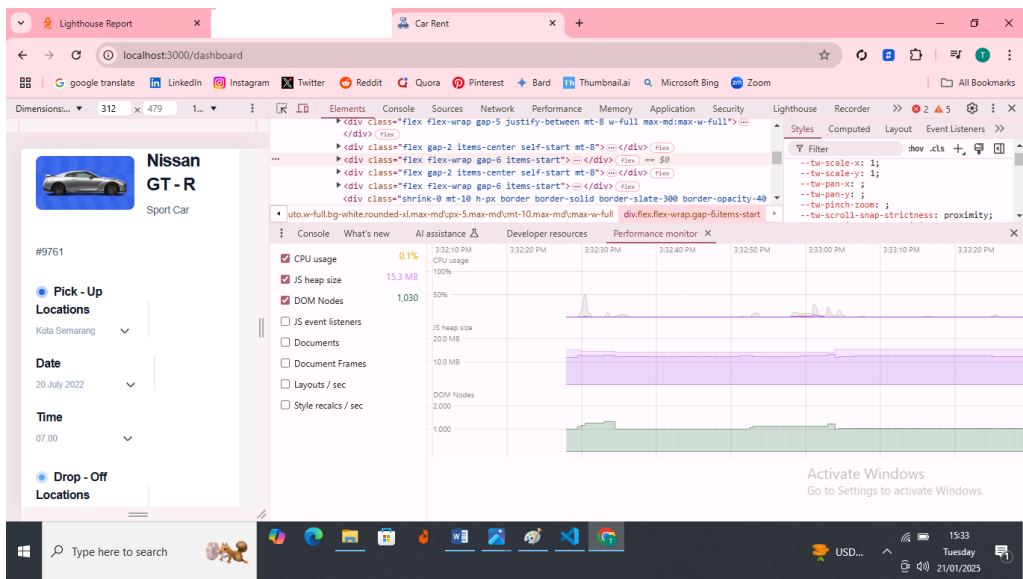
- These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on

Developer Resources:

- Go to more tools click on developer resources.



Performance Monitor



Security Overview

