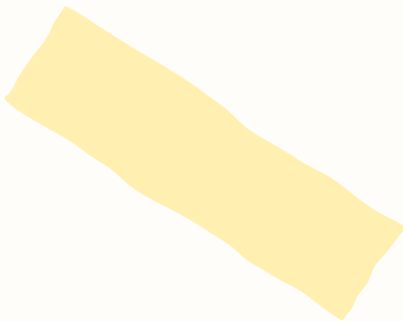


DAY 5 - TESTING AND BACKEND REFINEMENT - MORENT CAR RENTAL WEBSITE

Rental E-Commerce Website



Testing my website throughly

This document summarizes the testing efforts for the car rental marketplace, including functional testing, performance optimization, error handling, and security measures. The objective is to ensure all components work seamlessly before deployment.

Main Part Testing Reports

Testing Reports.csv

After checking my full website from component responsiveness, to api data fetching i have gathered these reports in a csv file named Testing Reports.csv

Test Case ID	Description	Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Verify car listings load from the Sanity Api	Open the homepage > Check if car listings are displayed > Match details with Sanity Api Data.	Car listings are displayed with correct details.	Cars displayed as expected.	Passed	High	
TC002	Verify search functionality	Enter "SUV" in the search bar > Check if results match search.	Only SUVs should be displayed in search results.	Only SUVs has been displayed as expected.	Passed	High	
TC003	Test fallback UI for missing car data	Disconnect internet > Refresh the homepage > Check for error message or fallback UI.	Fallback UI or a message should be displayed without errors	Fallback UI with Error message displayed.	Passed	Medium	-
TC004	Verify responsiveness on mobile devices	Open the website on a mobile device > Check layout and interactivity.	Website layout adjusts to mobile screens.	Layout is responsive.	Passed	Medium	Tested using Browser DevTool, an actual laptop and mobile.
TC005	Lighthouse performance test	Open Lighthouse in Chrome DevTools > Run performance analysis > Record score.	Performance score should be > 90.	Score: 98	Passed	Medium	Tested in incognito tab
TC006	Test cross-browser compatibility	Open website on Chrome, Firefox, Edge, and Safari > Verify functionality and layout.	Website works and displays consistently across browsers.	No Layout Shifts and no Inconsistency across browsers	Passed	Low	
TC007	Input validation for search bar	Enter "" in search bar > Verify no XSS vulnerability occurs.	No XSS or malicious execution happens.	Input sanitized successfully.	Passed	High	Tested manually.
TC008	API error handling for car details page	Disable API > Visit car details page > Check for error handling UI fallbacks.	Error message: "Please wait or check your internet!" has been displayed	Error handled gracefully.	Passed	High	-
TC009	Check entire layout and responsiveness	Go to all pages one by one > Check all components > ensure every UI Component & Page is user friendly	Every Component is user friendly and is aligned cleanly	User profile component is aligned on the left instead of right	Failed	Low	Align it on the right side to ensure UI reacts as expected

How i handled errors

I wrapped all fetches inside try-catch

```
const getData = async () => {
  let data;
  let users;
  try {
    data = await fetch(`${process.env.NEXT_PUBLIC_BASE_URL}/api/cars`);
    data = await data.json();
    users = await client.fetch(UsersQuery);
    setCarDetails(data);
    setUsers(users)
  } catch (error) {
    console.log(error);
  }
};
getData();
```

I used &&, ? : and || operators to handle UI Fallbacks

```
filteredCars.filter(car => car.tags && car.tags[0] + ' '.includes(tag)).slice(0, more).length ?
filteredCars.filter(car => car.tags && car.tags[0] + ' '.includes(tag)).slice(0, more).map((obj, key) => (
  <Card key={key} data={{
    slug: obj.slug,
    name: obj.name,
    price_per_day: obj.price_per_day,
    image: obj.image,
    type: obj.type,
    heart: obj.heart,
    original_price: obj.original_price,
    available: obj.available,
    fuel_capacity: obj.fuel_capacity,
    seating_capacity: obj.seating_capacity,
    tags: obj.tags,
    transmission: obj.transmission,
    reviews: obj.reviews,
    desc: obj.desc,
  }} users={users} />
) : <div className="flex items-center justify-center text-center text-xs opacity-50 h-[388px] min-w-[304px] rounded-lg bg-white
animate-pulse">Please wait or check you connection!</div>
```

Errors i got at build time

I got new errors that i never saw before after using sanity with nextjs API handlers GET, POST etc... 🤔

```
▲ Next.js 15.0.3
- Environments: .env.local

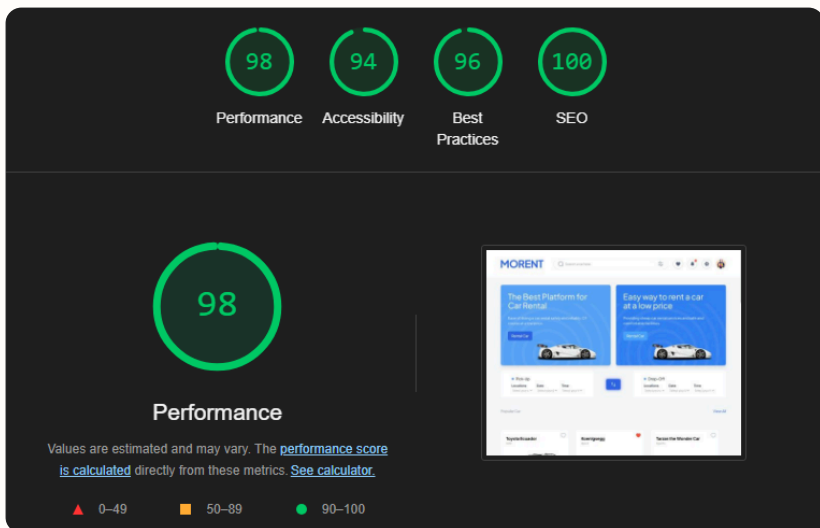
Creating an optimized production build ...
✓ Compiled successfully
✓ Linting and checking validity of types
✓ Collecting page data
  at async m (C:\Users\kk\Desktop\Morent - Way to Rent Cars\.next\server\app\details[...slug]\page.js:1:4312) {
    [cause]: [AggregateError: ] { code: 'ECONNREFUSED' }
  }
Error occurred prerendering page "/details/rolls-royce". Read more: https://next.js.org/docs/messages/prerender-error
TypeError: Cannot read properties of undefined (reading 'image')
  at m (C:\Users\kk\Desktop\Morent - Way to Rent Cars\.next\server\app\details[...slug]\page.js:1:8190)
  at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
Export encountered an error on /details/[...slug]/page: /details/rolls-royce, exiting the build.
✗ Static worker exited with code: 1 and signal: null
```

What i did to solve:

- I Separated **Development** Environment variables and **Production**.
- I ran “npm run build” locally in my laptop before deploying it on vercel.
- I fixed all linting errors and warnings.

Performance & Summary

I ran a report test on chrome lighthouse and my car renting website home page gave **98%**. **this test was in incognito tab**



Cross Platform Testing

- I tested my car rental website inside my own **laptop, big phone, small phone**.
- I tested it on **edge, chrome, opera, firefox**.
- I used tools like ChromeDevTool to check **errors, responsiveness, performance**.

Tested on Devices

LG V60, DELL Precision 5510, MOTOROLA Mini Phone

Findings:

1. Layout is responsive across most devices.
2. Minor alignment issues.

Fallback UI:

When no cars are available or API fails, a message is displayed:
"Please wait or check your internet!"

Repository Link: <https://github.com/DanielHashmi/Car-Rent-Website/tree/main/Documentation>