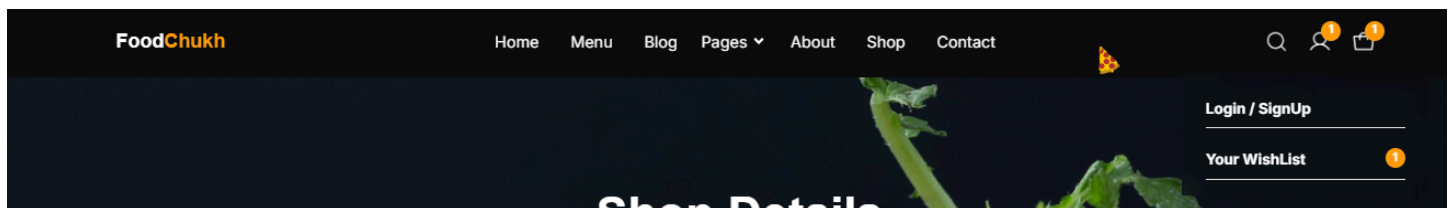


Day 5 - Testing and Backend Refinement - Food Chukh

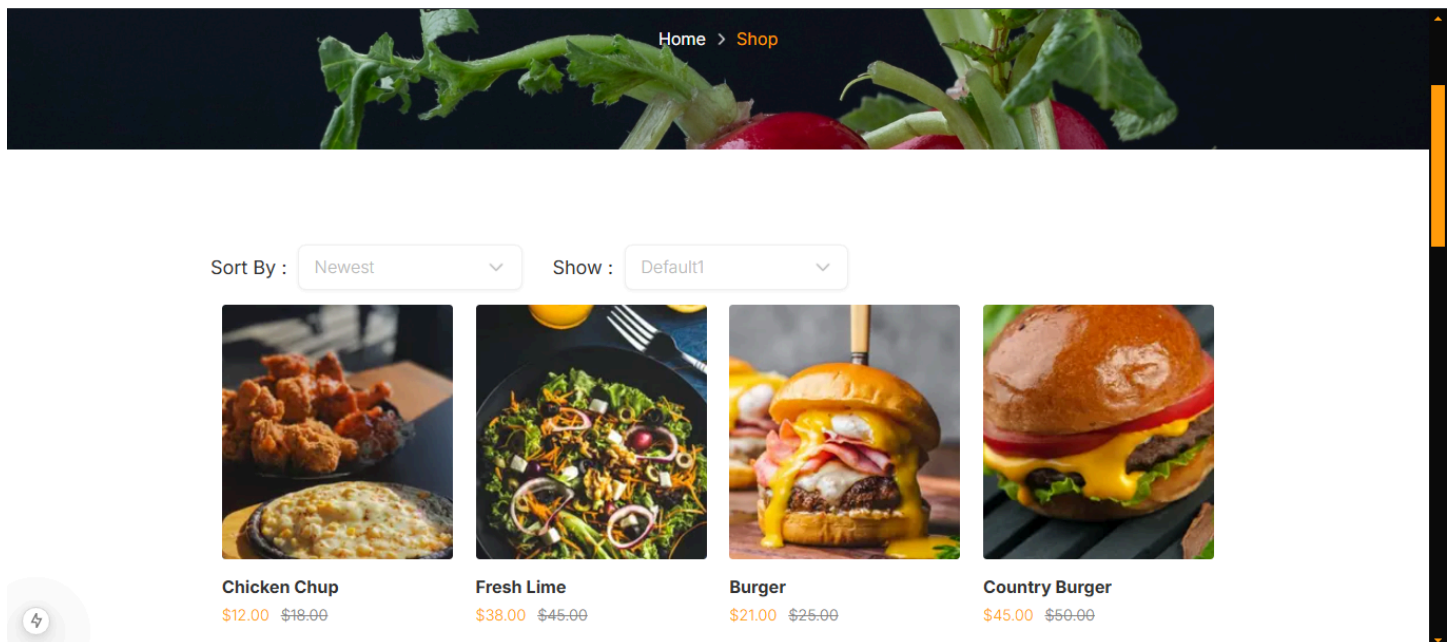
- **Functional Deliverables:**

Screenshots:

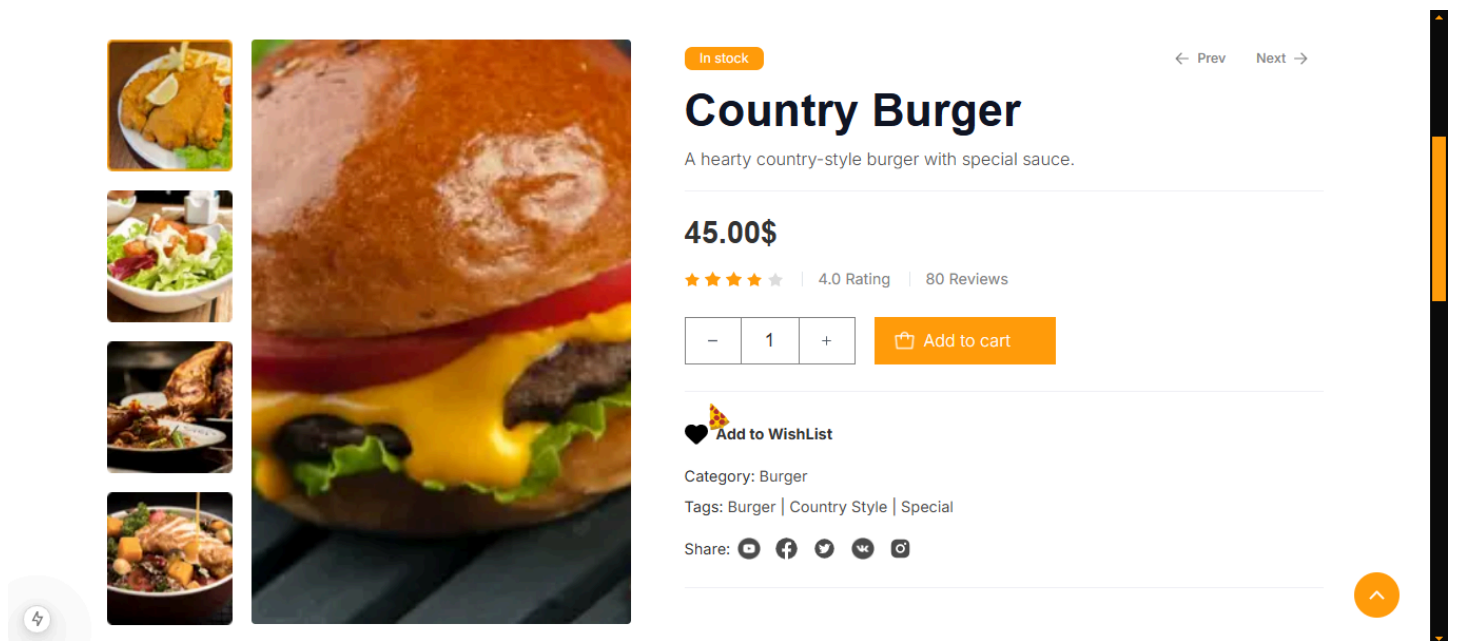
1.Header with Notificaitons:



2.Product List:

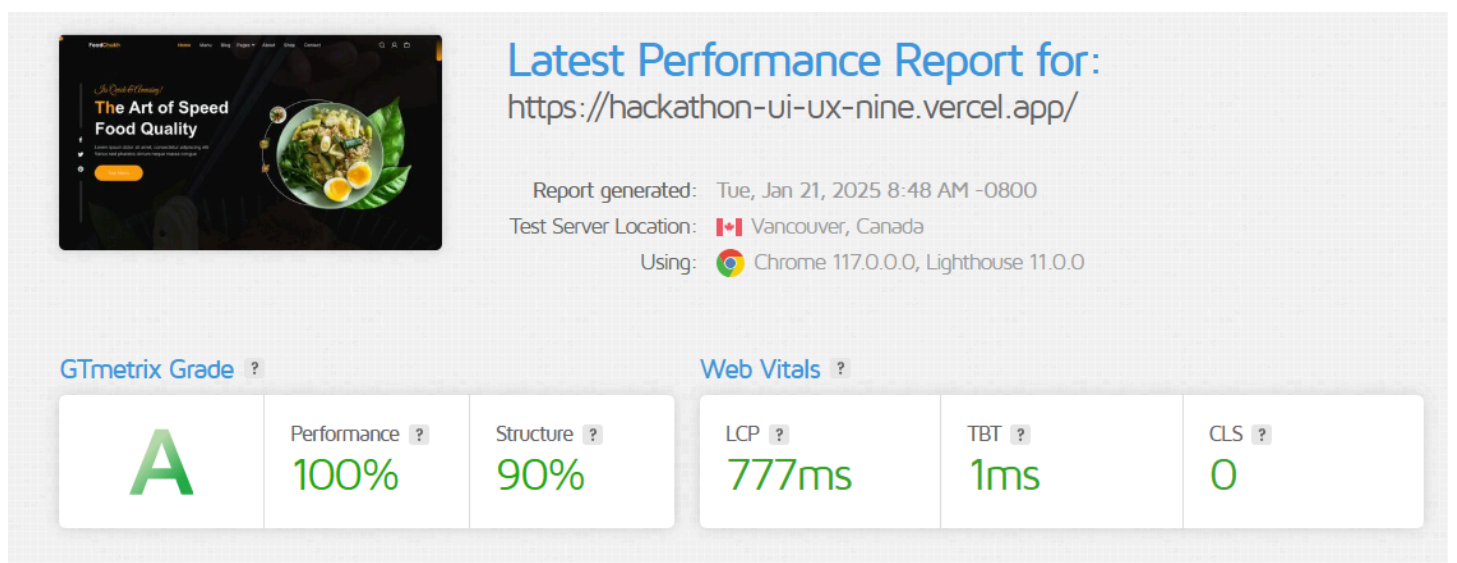


3.Each Product Page:



for more details view Day 4 - Functional Dynamic Components - Food Chukh

- GTMetrix Report:



Performance:

Performance Metrics

The following metrics are generated using Lighthouse Performance data.

Metric details ☐ OFF

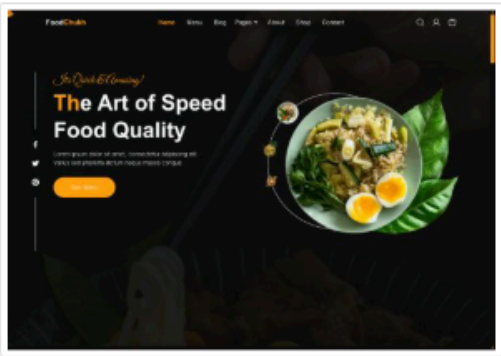
First Contentful Paint ?	Good - Nothing to do here 393ms	Time to Interactive ?	Good - Nothing to do here 444ms
Speed Index ?	Good - Nothing to do here 706ms	Total Blocking Time ?	Good - Nothing to do here 1ms
Largest Contentful Paint ?	Good - Nothing to do here 777ms	Cumulative Layout Shift ?	Good - Nothing to do here 0



Performance

Values are estimated and may vary. The [performance score](#) is calculated directly from these metrics. [See calculator.](#)

▲ 0–49 ■ 50–89 ● 90–100



- **Testing Report:**

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity	Remarks
TC001	Initial Website Load Time	1. Opened PostMan 2. Enter URL https://hackathon-ui-ux-nine.vercel.app 3. Send a GET request	The website should load in under 1-3 seconds	The website loads in 2.58s	Passed	Low	All steps executed successfully
TC002	Getting all Products	1. add /shop in the URL 2. selected GET request 3. Send a GET request	The Products should display after fetching from sanity	The products displayed!	Passed	Low	Works as expected
TC003	Getting Each Products Page	1. add /shop/[id] in the URL 2. selected GET request 3. Send a GET request	Each Product Details page should show!	The Details shown Successfully	Passed	Low	Works as expected
TC004	Getting Chefs Page	1. add /ourChefs in the URL 2. selected GET request 3. Send a GET request	All chefs should Show after fetching data from sanity!	The Details shown Successfully	Passed	Low	Works as expected
TC005	Checking Search Functionality	1. Click on Search Icon 2. Add Text on Search Bar	The Products should show Based on the text provided!	The Products are showing Successfully	Passed	Low	Works as expected
TC006	Checking Custom Cursor	1. Check Custom Cursor on different Browsers	Custom Cursor & pointer should show!	Not properly Showing on Safari browser! Resolved by updating code for custom cursor	Resolved	Medium	Working as expected after update
TC007	Checking Responsiveness across browser	1. Opening & changing layout on different browsers	Website Should Be responsive!	Website is responsive across different browsers	Passed	Low	Works as expected

		including: Chrome, Edge					
TC008	Checking For Internet Issue Handling	1.Disconnect ed the internet 2.Opened Website on local host	Website Should give a notification showing That User is Offline & fetch the data after the user comes back online!	Working good and handling the no internet issue gracefully!	Passed	Low	Works as expected

• Documentation:

Test Cases Executed and Results

All 8 test cases were executed successfully. 7 test cases passed without any issues, while 1 case (custom cursor functionality) required a resolution which was applied successfully.

Performance Optimization Steps

- Improved custom cursor implementation for better cross-browser compatibility.
- Ensured optimized loading times by testing with Postman and achieving a consistent load time of under 3 seconds.

Security Measures Implemented

- Verified secure API endpoints.
- Ensured data fetched from sanity is sanitized and protected.
- Used .env file to store sensitive information
- sanitize the input fields like comment box etc
- Tested for no internet connectivity to gracefully handle offline scenarios.

Prepared By: ~ Aliyan Jabbar