

Day 5 :-

⇒ Test cases executed and their results :-

i) Test cases :-

a) Product listing page:-

⇒ Successfully fetching data from sanity and showing on UI.

⇒ With all the functionalities added.

⇒ No issues found.

b) Test API ..

⇒ Test api that given are dynamically rendered in the frontend.

⇒ Implement lazy loading and giving error if there is network problem as "Fail to fetch products."

c) Add to cart page:-

⇒ Functionality has added to add products in cart with increment and decrement options.

⇒ Also user can remove the specific product from the cart

⇒ No issue found.

⇒ Working perfectly.

d) Mobile responsiveness:-

⇒ The UI is fully responsive according to mobile and desktop.

⇒ Every component behaves different in UI UX ~~comp~~ whether its mobile screen or desktop.

	Test Cases	Description	Steps	Results	Status	Remarks
1	TC01	Product Listing page	SHOP>FOOD	Fetching data from sanity and showing it to UI	passed	no issue found
2	TC02	Test APIs	refresh page	implement lazy loading and errors	passed with good connection	issue if internet not working
3	TC03	add to cart page	Specific cart content	Proper increment decrement property with remove option	passed	no issue found

mobile responsiveness	check on inspect or different device	all the components are arranged according to screen size no issue found	passed	no issue found



Category:

All



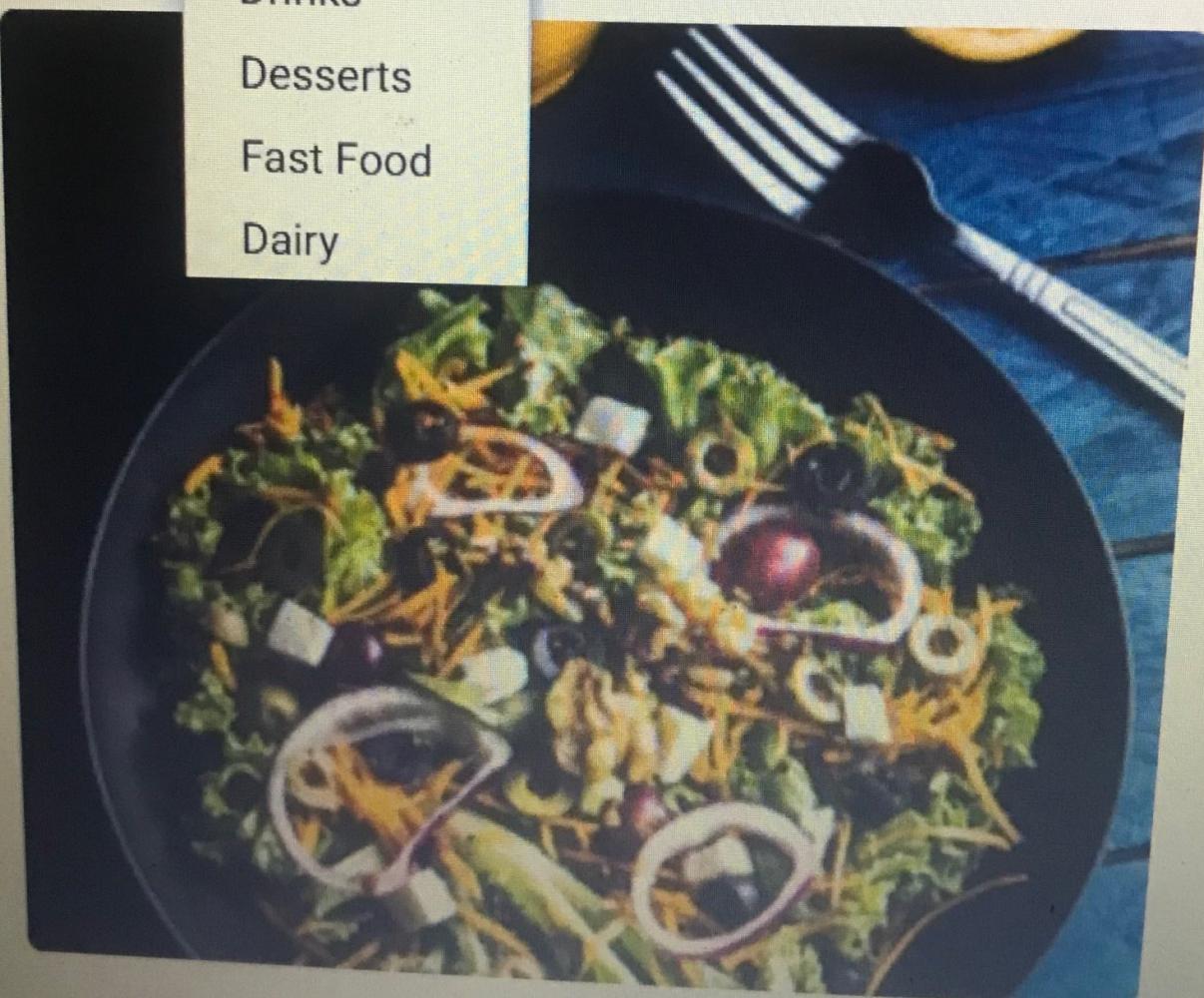
All

Drinks

Desserts

Fast Food

Dairy



Fresh Lime

\$ 38

Add to Cart

+

-



X



...

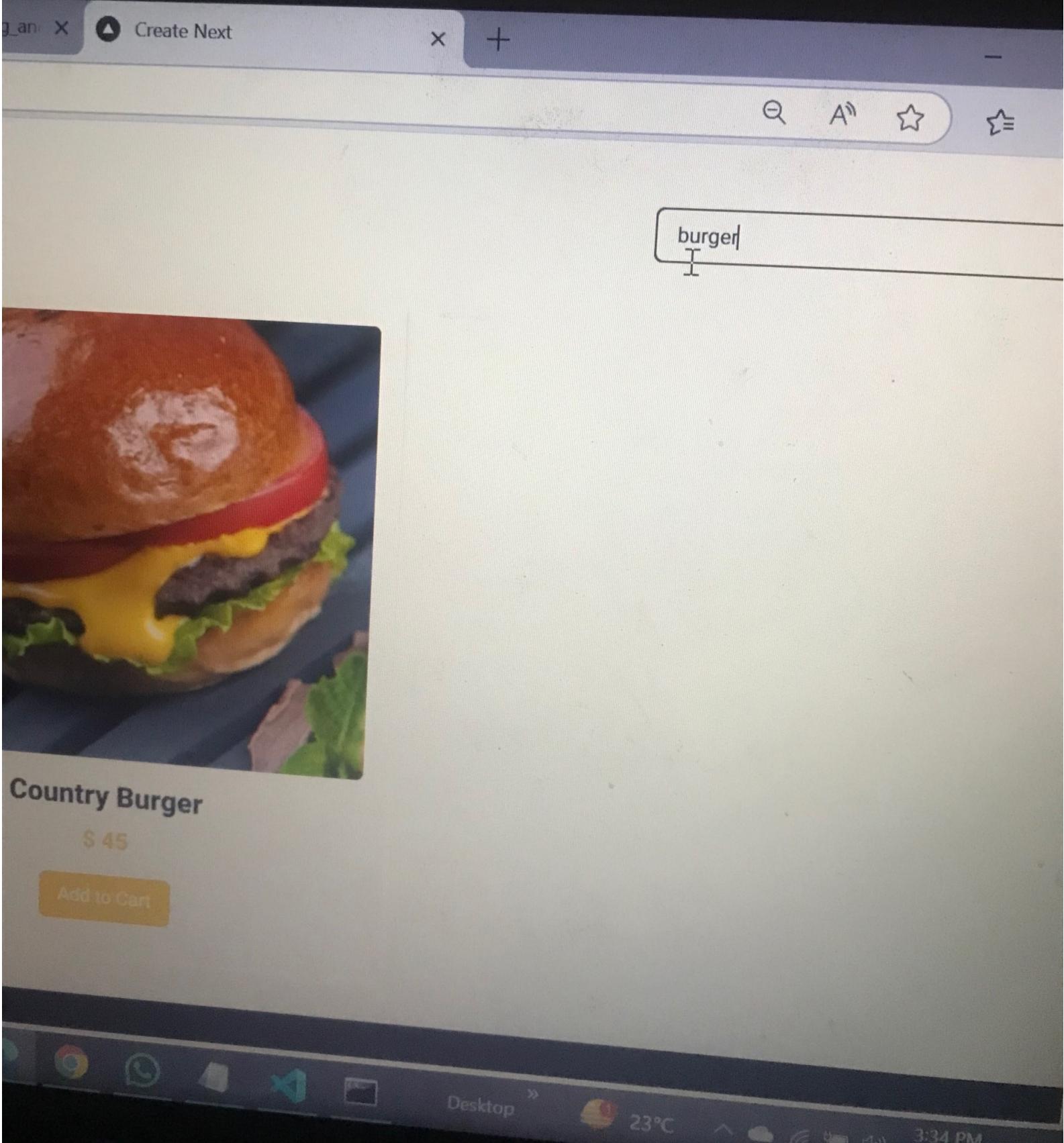


Search products...



Burger

\$ 21



Our Shop

Home > [Shop](#)

icecream

I

No products found for the search query.

```
const orderData = {  
  phone,  
  address,  
  cart,  
};  
  
try {  
  const response = await fetch('/api/checkout', {  
    method: 'POST',  
    headers: {  
      'Content-Type': 'application/json',  
    },  
    body: JSON.stringify(orderData),  
  });  
  
  if (response.ok) {  
    alert('Order placed successfully!');  
    setCart([]); // Clear the cart after successful order  
    setCheckoutVisible(false); // Close the checkout modal  
  } else {  
    alert('Something went wrong, please try again.');  
  }  
} catch (error) {  
  console.error(error);  
  alert('Error placing the order');  
}
```

```
export default async function handler(req: { method: string; body: { r
if (req.method === 'POST') {
  const { name, phone, address, cart } = req.body;
  // Process the checkout (e.g., save the order to a database)
  console.log('Order received:', { name, phone, address, cart });

  // Respond with success message
  res.status(200).json({ message: 'Order placed successfully!' });
} else {
  res.status(405).json({ message: 'Method Not Allowed' });
}
}
```