

PHASE 4 :

USER AUTHENTICATION :

1. Set up a Node.js project: Make sure you have Node.js installed. Create a new project folder, navigate to it in your terminal, and run `npm init` to initialize a Node.js project. Install necessary dependencies:

2. Create a basic folder structure

3. Create a `config.js` file to store your configuration variables (database URI, JWT secret, etc.).

4. Set up the MongoDB database using Mongoose in your `server.js`:

5. Create a user model in `models/User.js`

6. Create registration and login routes in `routes/user.js` and `routes/auth.js`, respectively. Use bcrypt for password hashing and JWT for authentication.

7. Implement user registration and login functionality in the route files.

8. In the `auth.js` route, generate a JWT token upon successful login and send it to the client.

9. Protect your routes by verifying JWT tokens in your routes using middleware.

Implement shopping cart functionality, calculate the total, and enable a smooth checkout process.

```
// JavaScript for cart and checkout functionality
const addToCartButtons = document.querySelectorAll('.product button');
const cartItemsContainer = document.querySelector('.cart-items');

addToCartButtons.forEach(button => {
  button.addEventListener('click', () => {
    const product = button.parentElement;
    const productName = product.querySelector('h3').textContent;
    const productPrice = parseFloat(product.querySelector('p').textContent.slice(1));
    const cartItem = document.createElement('div');
    cartItem.classList.add('cart-item');
    cartItem.innerHTML = `
      <p>${productName}</p>
      <p>${productPrice.toFixed(2)}</p>
      <button class="remove-item">Remove</button>
    `;
    cartItemsContainer.appendChild(cartItem);

    // Calculate and display updated total
    const cartTotal = document.querySelector('#cart p');
    const currentTotal = parseFloat(cartTotal.textContent.slice(7));
    const newTotal = currentTotal + productPrice;
    cartTotal.textContent = `Total: ${newTotal.toFixed(2)}`;
  });
});

// Remove item from cart
cartItemsContainer.addEventListener('click', event => {
  if (event.target.classList.contains('remove-item')) {
    const cartItem = event.target.parentElement;
    const itemPrice = parseFloat(cartItem.querySelector('p:nth-child(2)').textContent);
    const cartTotal = document.querySelector('#cart p');
    const currentTotal = parseFloat(cartTotal.textContent.slice(7));
    const newTotal = currentTotal - itemPrice;
    cartTotal.textContent = `Total: ${newTotal.toFixed(2)}`;
    cartItem.remove();
  }
});
```

