Deployment Documentation

Project Overview

This project is a modern e-commerce website specializing in furniture, built using Next.js for the frontend. The platform provides a seamless shopping experience across devices, including desktops, laptops, tablets, and mobile phones. The backend utilizes Sanity CMS for data storage, ensuring a robust and scalable solution for managing product details, categories, and other content. Key features include:

- User-friendly interface for browsing and purchasing furniture.
- Secure user authentication and data handling.
- Optimized performance for fast loading times and smooth navigation.

Deployment Procedure

1. Prerequisites

Hardware Requirements

The application is optimized for use on:

- Desktops and laptops with standard specifications.
- Tablets and smartphones with modern browsers.

Software Requirements

- **Frontend Framework**: Next.js (React-based framework)
- **Backend**: Sanity CMS for data storage and management
- **Password Encryption**: `bcryptjs` library for secure password handling
- **Dependencies**:
    - Node.js
    - npm (or yarn) for package management

2. Setup Instructions

Development Environment

1. Clone the repository:
2. `git clone [repository_url]`
3. Install dependencies:
4. `npm install`

5. Configure environment variables:
   - `NEXT_PUBLIC_SANITY_PROJECT_ID=[value]`
   - `NEXT_PUBLIC_SANITY_DATASET=[value]`
   - `NEXT_PUBLIC_SANITY_API_TOKEN=[value]`

Deployment Environment

1. Prepare the server with required software and libraries.
2. Configure CI/CD pipeline if applicable.
3. Build and deploy the application:
4. `npm run build`
5. `npm run start`

Performance Reports

1. Key Metrics

Metric	Value
Response Time	Average
Throughput	Average
Resource Utilization	CPU: Average, Memory: Average

2. Benchmark Results

- **Test 1**: Load testing with 1000 concurrent users
  - Results: Average response time of 300ms, no errors encountered
- **Test 2**: Database query performance
  - Results: Average query execution time of 50ms

3. Optimization Suggestions

- Optimize images for faster loading times.
- Implement caching mechanisms for frequently accessed data.

Test Case Reports

1. Overview

The testing approach includes unit tests, integration tests, and end-to-end tests to ensure all components function as expected.

2. Test Cases

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Tested By	Assigned To	Remarks
TC0001	Validation Category Menu in Navi from Sanity CMS	Open website > Display Category	Parking page should show with out showing error when sanity cms not connected.	Nav Bar Working correctly	Passed	High	Ather	Shoaib	Handled gracefully
TC0002	Validate product listing page	Open Product Page > Verify Product	Products displayed correctly	Products displayed correctly	Passed	Medium	Ather	Shoaib	Handled gracefully
TC0003	Validate Product Detail Page	Open Product Detail Page > Verify Product Detail Page	Product Detail Page Displayed correctly	Product detail Displayed Correctly	Passed	Medium	Ather	Shoaib	Handled gracefully
TC0004	Check Cart Functionality	Add Product to cart > Verify Cart Contents	Cart updates with added product	Text Alignment needed and + - Funcation needed		High	Ather	Shoaib	Work but need Some text algnment and need to add Funcation of + or - Qty
TC0004	Check Cart Functionality	Add Product to cart > Verify Cart Contents	Cart updates with added product	Cart updates with added product	Passed	High	Ather	Shoaib	Work as expected
TC0005	Check Cart Functionality	Remove from Cart > validation and operational functions	Remove any items from cart	cart remove functionality as expected	Passed	High	Ather	Shoaib	Work as expected
TC0006	Search any product and category	Search Product or categories	Product searching functionality is working	Product searching functionality is working	Passed	Medium	Ather	Shoaib	Work as expected
TC0007	Ensure responsiveness on mobile	resize browser window > check layout	layout adjust properly to screen size	Nav bar is showing 2 X buttons		Medium	Ather	Shoaib	Nav bar is showing 2 X buttons
TC0007	Ensure responsiveness on mobile	resize browser window > check layout	layout adjust properly to screen size	layout adjust properly to screen size	Passed	Medium	Ather	Shoaib	Work as expected
TC0008	Test Api error handling	Disconnect API > Refresh Page	Show fallback ui with error message	Error message shown	Passed	Medium	Shoaib	Shoaib	Handled gracefully

3. Bug Reports

Test Case	Test Case Description	Test Steps	Expected Result	Actual Result	Stat	Severity L	Tested	Assigned	Remarks
TC0004	Check Cart Functionality	Add Product to cart > Verify Cart Contents	Cart updates with added product	Text Alignement needed and + - Funcation needed		High	Ather	Shoaib	Work but need Some text algnment and need to add Funcation of + or - Qty
TC0007	Ensure responsiveness on mobile	resize browser window > check layout	layout adjust properly to secure size	Nav bar is showing 2 X buttons		Medium	Ather	Shoaib	Nav bar is showing 2 X buttons
TC0009	Home page Popular products and	Home Page > New Ceramics	It should be functional	not functional, Its Static and fixed		Medium	Shoaib	Shoaib	In Process
TC0010	Company name and searchicon in	Home Page > Company Name and Search Icon	UI Should responsive in all platform.	Overlapping in tab		Medium	Shoaib	Shoaib	In Process
TC0011	Signup not functional	Signup or login	Sign up should be functional and user/customer can login.	not functional, Its Static and fixed		Medium	Shoaib	Shoaib	In Process
TC0012	Order tracking is missing	Place Order > order tracking	when the order is placed, order tracking is necessary.	Order tracking is missing		Medium	Shoaib	Shoaib	In Process
TC0013	Intimation while add to cart and o	Add to Cart > Checkout	Intimation should be add in website in add to cart and place order.	Intimation is missing		low	Shoaib	Shoaib	In Process

Post-Deployment Steps

1. Monitor application performance.
2. Address any critical bugs or issues identified during deployment.
3. Regularly back up data and configurations.

Conclusion

The deployment process ensures the furniture e-commerce website operates smoothly across all devices. With its modern tech stack and secure backend, the platform is designed to provide a reliable and engaging user experience. Regular monitoring and maintenance will keep the system optimized for performance.

Appendices

- **Glossary**: Define technical terms.
- **References**: Include links to external tools or documentation.