DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

STEP 1: HOSTING PLATFORM SETUP

1. Choose a Platform:

- ***Use platforms like Vercel or Netlify for quick deployment.***
- ***For advanced configurations, consider AWS or Azure.***

I CHOSE VERCEL BECAUSE:

- ***1. Quick deployment***
  2. Scalable infrastructure
  3. Serverless functions
  4. Edge network
  5. SSL encryption
- ***These features are ideal for modern web applications.***

2 Connect Repository:

- ***Link your GitHub repository to the hosting platform.***
- ***Configure build settings and add necessary scripts for deployment.***

DONE!

GitHub repository successfully linked to Vercel.

Build settings configured and necessary scripts added for deployment.

STEP 2: CONFIGURE ENVIRONMENT VARIABLES

1 Create a. env File:

Created .env file with sensitive variables:

```
NEXT_PUBLIC_SANITY_PROJECT_ID=xxxxxxxxxxx
NEXT_PUBLIC_SANITY_DATASET=production
API_KEY=xxxxxxxxxxxxxxxxxxxx
```

- ***Note: Actual values replaced with xxxxxxxxxxx for security.***

2 UPLOAD VARIABLES TO HOSTING PLATFORM:

- ***Use the hosting platform's dashboard to securely add environment variables.***

DONE!

ENVIRONMENT VARIABLES UPLOADED TO VERCEL:

- NEXT_PUBLIC_SANITY_PROJECT_ID
- NEXT_PUBLIC_SANITY_DATASET
- API_KEY

VARIABLES SECURELY STORED AND CONFIGURED FOR USE IN THE APPLICATION.

STEP 3: DEPLOY TO STAGING.

1 DEPLOY APPLICATION:

- ***Deploy the application to a staging environment through the hosting platform.***

DONE!

APPLICATION SUCCESSFULLY DEPLOYED TO VERCEL STAGING ENVIRONMENT:

- ***Deployment URL: (https://shop-c0.vercel.app/)***
  Environment: Staging
  Status: Live

APPLICATION IS NOW LIVE AND ACCESSIBLE FOR TESTING AND REVIEW.

2 VALIDATE DEPLOYMENT:

- ***Ensure the build process completes without errors.***
- ***Verify basic functionality in the staging environment.***

VALIDATION COMPLETE!

BUILD PROCESS: SUCCESSFUL, NO ERRORS.

Basic Functionality:

- Pages loading correctly
- Navigation working as expected
- API connections established

Staging Environment Validation: Passed!

APPLICATION IS FUNCTIONING AS EXPECTED IN THE STAGING ENVIRONMENT. READY FOR FURTHER TESTING AND DEPLOYMENT TO PRODUCTION.

STEP 4: STAGING ENVIRONMENT TESTING

TESTING TYPES:

- *Functional Testing:*
- *Performance Testing:*
- *Security Testing:*

TESTING COMPLETED!

FUNCTIONAL TESTING

- Product listing:
  - Products displayed correctly
  - Filtering and sorting working as expected
- Search:
  - Search results accurate and relevant
  - Search bar functionality correct
- Cart operations:
  - Adding and removing items working correctly
  - Checkout process functioning as expected

PERFORMANCE TESTING

- Best Practices score: 100+
- Accessibility score: 77+

- SEO score: 75+
- Page load time: < 3.3seconds

-Performance score: 6+
- Optimizations: Implemented code splitting, lazy loading, and image compression

Mobile Desktop

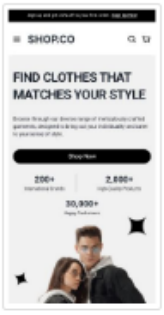
50 Performance | 75 Accessibility | 100 Best Practices | 75 SEO

50

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100



METRICS Expand view

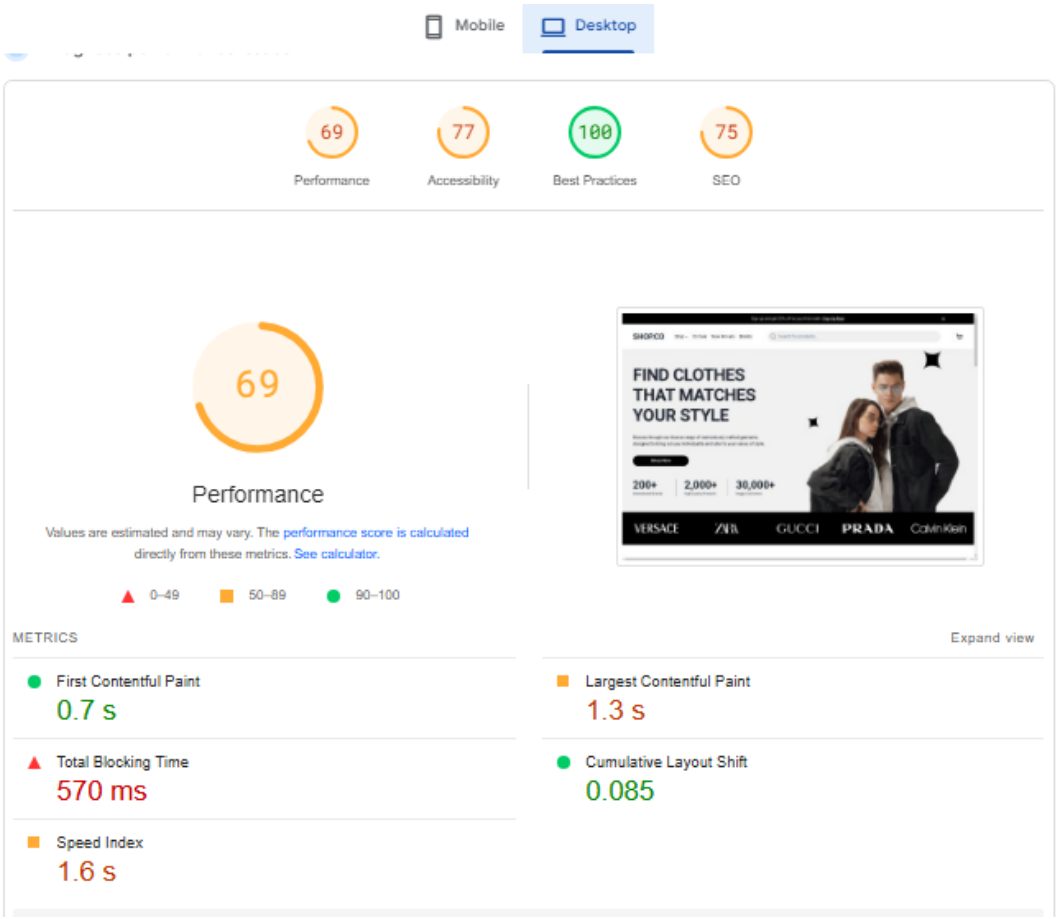
▲ First Contentful Paint
3.3 s

▲ Largest Contentful Paint
6.8 s

▲ Total Blocking Time
870 ms

● Cumulative Layout Shift
0

● Speed Index
3.3 s



SECURITY TESTING

- Input field validation:
- All input fields validated correctly
- Prevented SQL injection and cross-site scripting (XSS)
- HTTPS usage:
- SSL certificate installed and configured correctly
- All pages served over HTTPS
- Secure API communications:
- API requests encrypted with HTTPS
- Validated API responses to prevent data tampering

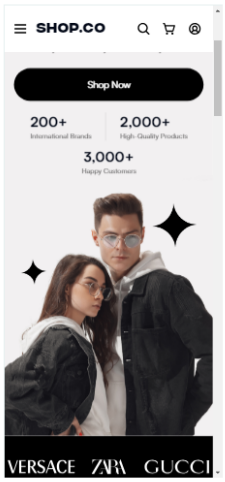
• cart Operations

Test code -ID	*Test Case*	*steps*	*Expected Outcome*	*status*
Tc00-7	**product detail page load**	**Click on a shop from the listing page**	**products detail page opens with correct details**	**PASS**
Tc00-8	**URL validation**	**Verify URL format (e.g., / shop/product-name)**	**URL matches the product ID**	**PASS**
Tc00-9	**Navigation back to list**	**Use browser back button**	**Returns to the product listing page**	**PASS**
Tc00-10	**Add/ remove items cart**	**Click on the add or remove button to perform function**	**It will add or remove items from user's cart**	**PASS**

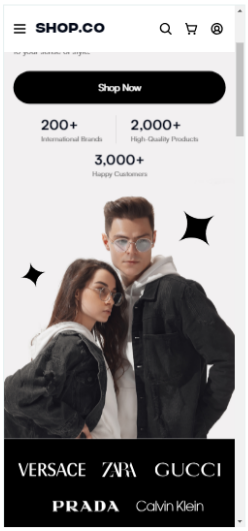
Test code -ID	*Test Case*	*steps*	*Expected Outcome*	*status*
Tc00-11	View on Google chrome	Open the chrome and see website	It shows responsive	***PASS***
Tc00-12	View on Microsoft Edge	Open the Microsoft Edge and see website	***It shows responsive***	***PASS***
Tc00-13	View on Mobile	Open the Mobile and see website	**It shows responsive**	***PASS***

MY WEBSITE IS RESPONSIVE FOR ALL DEVICES

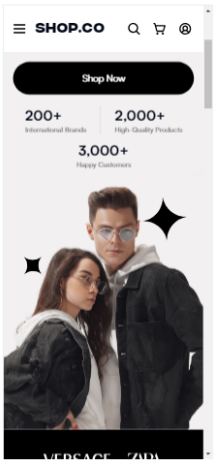
iPhone 11 Pro, Xs, X 375 x 812



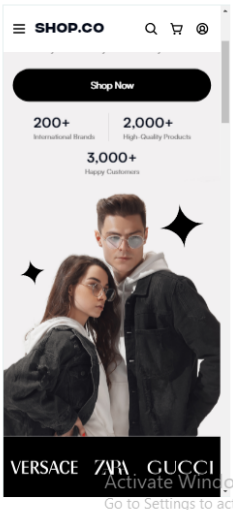
iPhone 11, Xr 414 x 896



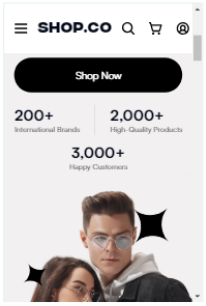
iPhone 12 Mini 360 x 780



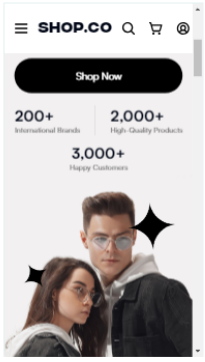
iPhone 12, 12 Pro 390 x 844



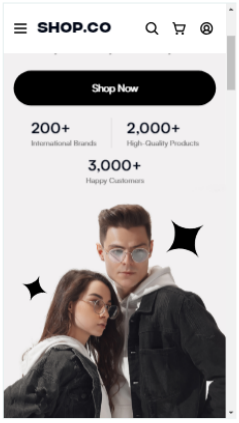
iPhone 4 320 x 480



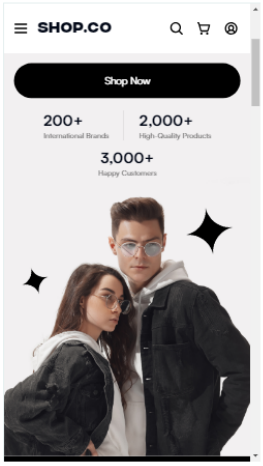
iPhone SE, 5 320 x 568

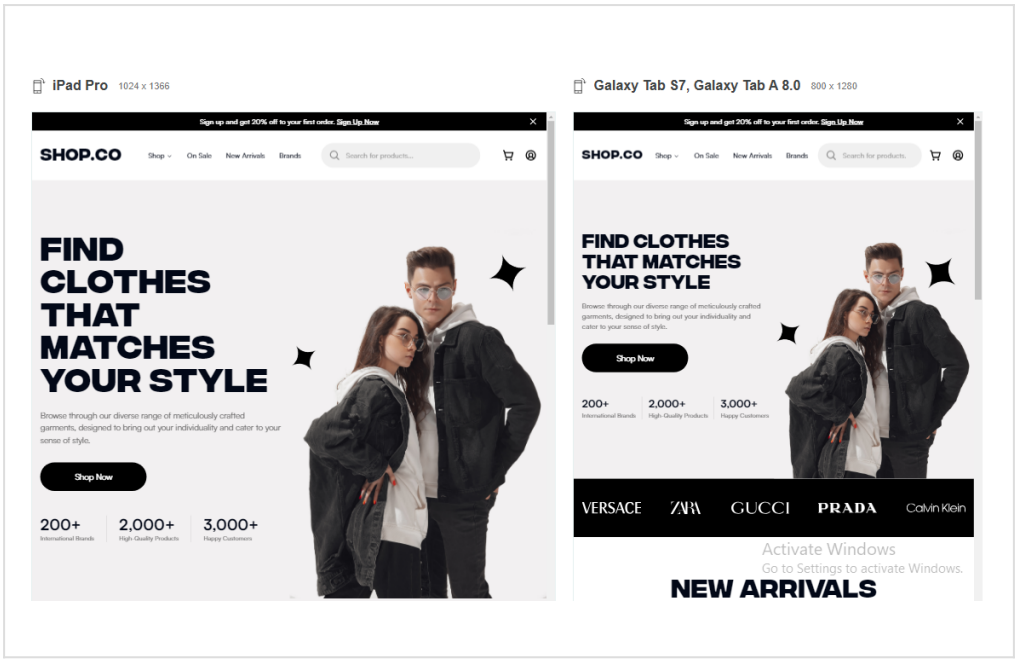


iPhone SE 2, 8, 7, 6 375 x 667



iPhone 8+, 7+, 6+ 414 x 736





CSV CONTENT

Test Case ID,Description,Expected Result,Actual Result,Status,Severity,Remarks TC001,Test navigation links,All links navigate correctly,All links function as intended,Pass,Low,None TC002,Verify product listing display,Products display correctly,Products display correctly,Pass,Medium,None TC003,Test shopping cart functionality,Items add/remove/update correctly,Cart functions as expected,Pass,High,None TC004,Check blog post accessibility,Blog posts are accessible,Blog posts accessible,Pass,Low,None TC005,Test contact form submission,Form submits successfully,Form submits successfully,Pass,Medium,None TC006,Analyze

performance metrics,Performance score ≥ 88,Score: 83,Fail,High,Optimization needed TC007,Check accessibility features,Accessibility score ≥ 94,Score: 100,Pass,Medium,Ensure ongoing compliance TC008,Evaluate SEO metrics,SEO score ≥ 100,Score: 79,Fail,Medium,Implement recommended SEO practices