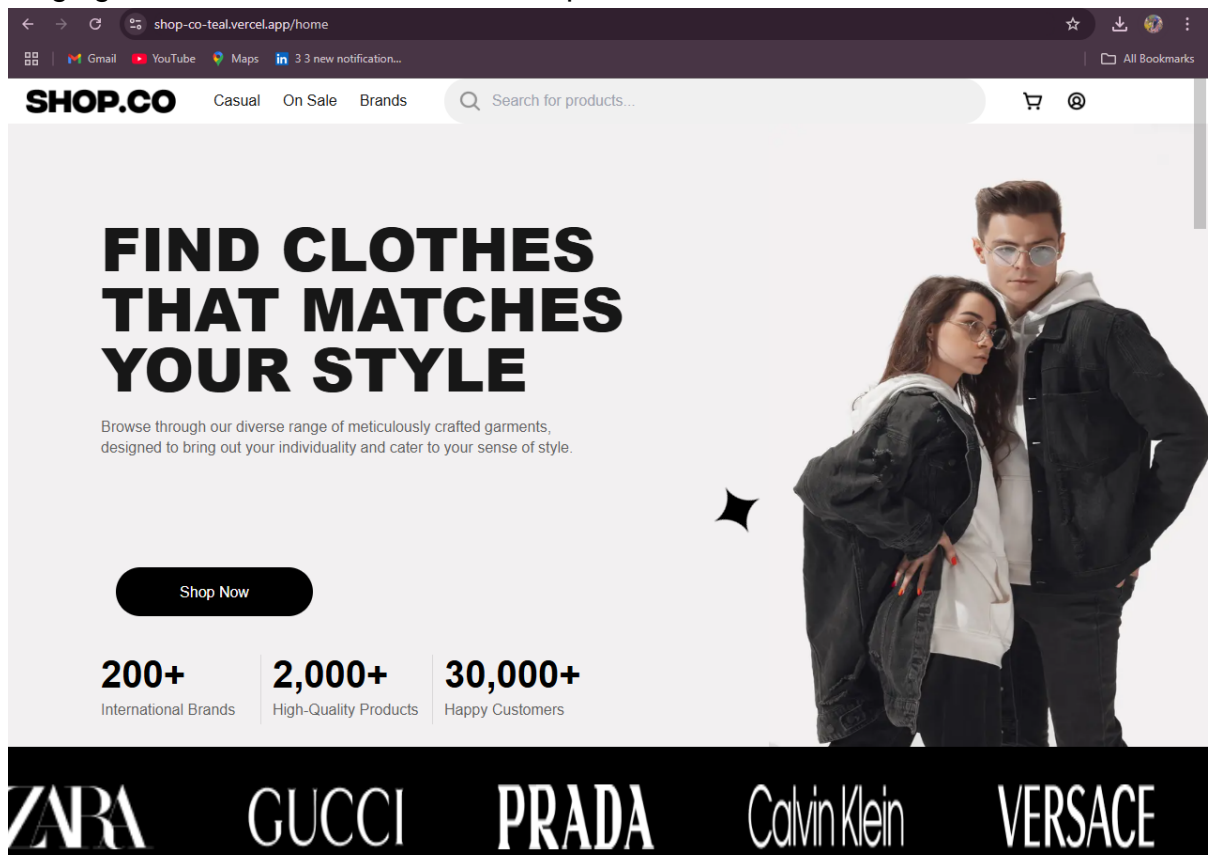


Day 6 - Deployment Preparation and Staging Environment Setup - [SHOP.CO]

1. Functional Deliverables

Screenshots or Screen Recordings:

- **Staging Environment Deployed Link:** Description: Demonstrates the deployed staging environment accessible via the provided link.



- **GitHub Repository Setup:** Description: Displays the GitHub repository structure with organized folders and files, including documents and reports.
- **Performance Testing Results:** Description: Good results of Lighthouse/GTmetrix for performance metrics.
- **README.md File:** Description: Highlights the README file summarizing project activities and folder structure.

2. Code Deliverables

Key Deployment Code Snippets:

- **GitHub Repository Structure:**
 - /documents folder: Contains Day 1 to Day 6 documents.
 - /public folder: Stores assets like images, performance reports, and test case reports.
 - /components folder: Contains reusable components such as Header, Footer, ProductList.
 - README.md: Summarizes project details.

3. Documentation

Steps Taken to Prepare Deployment:

1. **Staging Environment:**
 - a. Deployed the application to Vercel for a production-like environment.
 - b. Configured .env files for staging-specific settings.
2. **Testing:**
 - a. Conducted end-to-end testing to verify functionality in staging.
 - b. Used Postman to validate API endpoints.
3. **Performance Testing:**
 - a. Ran Lighthouse audits to assess page speed, accessibility, and SEO.
 - b. Generated reports and included them in the repository.
4. **Documentation Preparation:**
 - a. Compiled documents from Day 1 to Day 6.
 - b. Created CSV reports for test cases with details on passed and failed scenarios.

Challenges Faced and Solutions Implemented:

- **Challenge 1:** Deployment issues due to environment variable mismatch.
 - **Solution:** Ensured correct variable setup in Vercel and .env files.
- **Challenge 2:** Failed API calls in staging due to CORS errors.
 - **Solution:** Adjusted API configurations to allow requests from the staging domain.

Best Practices Followed During Deployment:

1. **Version Control:** Used Git for tracking changes and managing branches.
2. **Documentation:** Maintained clear and organized documentation to streamline submission.
3. **Testing:** Thoroughly tested staging environment before submission.

4. Submission Checklist

- **Staging Environment:** ✓
- **GitHub Repository:** ✓
- **Test Case Report:** ✓
- **Performance Testing Results:** ✓
- **Final Review:** ✓