

Day 6 - Deployment Preparation and Staging Environment Documentation

Quick Commerce Coffee Shop

Overview

Building upon the progress made in Days 3-5, where we successfully integrated Sanity CMS, implemented dynamic components, and conducted thorough testing, Day 6 focuses on formalizing our deployment preparation and staging environment setup. Our project is already hosted on Vercel with proper environment variable configuration, giving us a head start on today's objectives.

Current Project Status

Before proceeding with Day 6's tasks, here's a summary of our existing setup:

- Hosting Platform: Vercel (already configured)
- CMS Integration: Sanity CMS (fully integrated)
- Environment Variables: Configured in Vercel dashboard

Previous Achievements:

- Day 3: API Integration and data migration to Sanity CMS
- Day 4: Implementation of dynamic components (Cart, Reviews, Search, etc.)
- Day 5: Comprehensive testing and backend refinement

Deployment Environment Structure

Our project utilizes multiple environments to ensure reliable development and deployment:

1. Development (DEV)

- Local development environment
- Uses ``.env.local`` for environment variables
- Connected to Sanity development dataset

2. Staging (SIT/UAT)

- Pre-production environment on Vercel
- Uses production build with staging-specific variables

3. Production (PROD)

- Live production environment on Vercel
- Uses production build with production variables

Environment Variables Configuration

Current environment variables configured in Vercel:

```
NEXT_PUBLIC_SANITY_PROJECT_ID=<project_id>  
NEXT_PUBLIC_SANITY_DATASET=production  
NEXT_PUBLIC_SANITY_API_VERSION=2024-01-19  
SANITY_API_TOKEN="secret"
```

Deployment Preparation Steps

1. Pre-Deployment Checklist

- Verify all environment variables are properly set in Vercel
- Ensure all API endpoints are properly configured
- Confirm Sanity CMS schemas are up to date
- Check all dynamic components are functioning correctly
- Validate error handling for API failures
- Review security measures (API key protection, HTTPS)

2. Staging Environment Validation

- Deploy to staging environment
- Test all dynamic features in staging:
 - Product listing and filtering
 - Search functionality
 - Cart operations
 - Checkout process
 - Review system

- Verify environment variables are working correctly
- Test error handling and fallback states

3. Performance Optimization

- Implement lazy loading for images
- Enable automatic static optimization where applicable
- Verify proper code splitting
- Test loading states and transitions

4. Security Measures

- Verify API route protection
- Ensure sensitive data is not exposed in client-side code
- Validate CORS settings
- Check security headers

Deployment Process

Automatic Deployments

Our project utilizes Vercel's automatic deployment feature:

1. Push to `main` branch triggers production deployment
2. Push to `development` branch triggers staging deployment
3. Pull requests create preview deployments

Manual Deployment Process

If needed, manual deployment can be performed:

```
# Build the project locally
npm run build

# Deploy to Vercel
vercel --prod
```

Testing in Staging Environment

Functional Testing

- Complete end-to-end testing of user flows
- Verify all dynamic components work as expected

Test error states and recovery

Performance Testing

- Run Lighthouse audits
- Test load times and Core Web Vitals
- Verify mobile responsiveness

Security Testing

- Validate environment variable protection
- Test API endpoint security
- Verify data handling and storage

Documentation Updates

The project documentation has been updated to include:

- - Deployment procedures
- - Environment configuration
- - Testing protocols
- - Troubleshooting guides

Next Steps

1. Monitor staging environment for any issues
2. Plan for scalability improvements
3. Set up monitoring and alerting
4. Document maintenance procedures

Conclusion

With the deployment preparation and staging environment setup complete, our Quick Commerce Coffee Shop project is well-positioned for reliable and scalable operation. The existing Vercel hosting and Sanity CMS integration provide a solid foundation, while our comprehensive testing and documentation ensure maintainability and future development efficiency.