

Prepared By: **Urooj Sadiq**

DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

Overview

Today, I write the steps required to prepare for deploying my **SHOP.COM** e-commerce website, which specializes in clothing for men, women, and kids. It includes setting up a staging environment and conducting detailed tests before the final production deployment. The purpose is to ensure that the website is fully functional, free of bugs, and ready for deployment on live servers to provide a seamless shopping experience for customers.

Deployment and Finalization Process

1. Code Review & Final Adjustments

- **Code Quality:** Conducted a thorough review to ensure the code is clean, maintainable, and well-documented. Removed unused imports, console logs, and dependencies.
- **Version Control:** Managed version tracking through Git, creating a release branch for deployment.
- **Environment Variables:** Verified that API keys and sensitive credentials are stored securely in `.env` files, avoiding hardcoding.
- **Asset Optimization:** Minified JavaScript, CSS, and images. Removed any unused assets to reduce load times.
- **Cross-Browser Testing:** Tested compatibility on major browsers like Chrome, Firefox, and Safari to ensure a consistent experience.

2. Staging Environment Setup

- **Replication:** Created a staging environment mirroring the production setup for accurate testing.
- **Release Tagging:** Tagged versions in Git for clear version control and rollback if necessary.

3. Deploying to Staging

- **Application Build:** Compiled the application using `npm run build` to generate production-ready code.
- **Deployment:** Uploaded code to the staging server using Git.

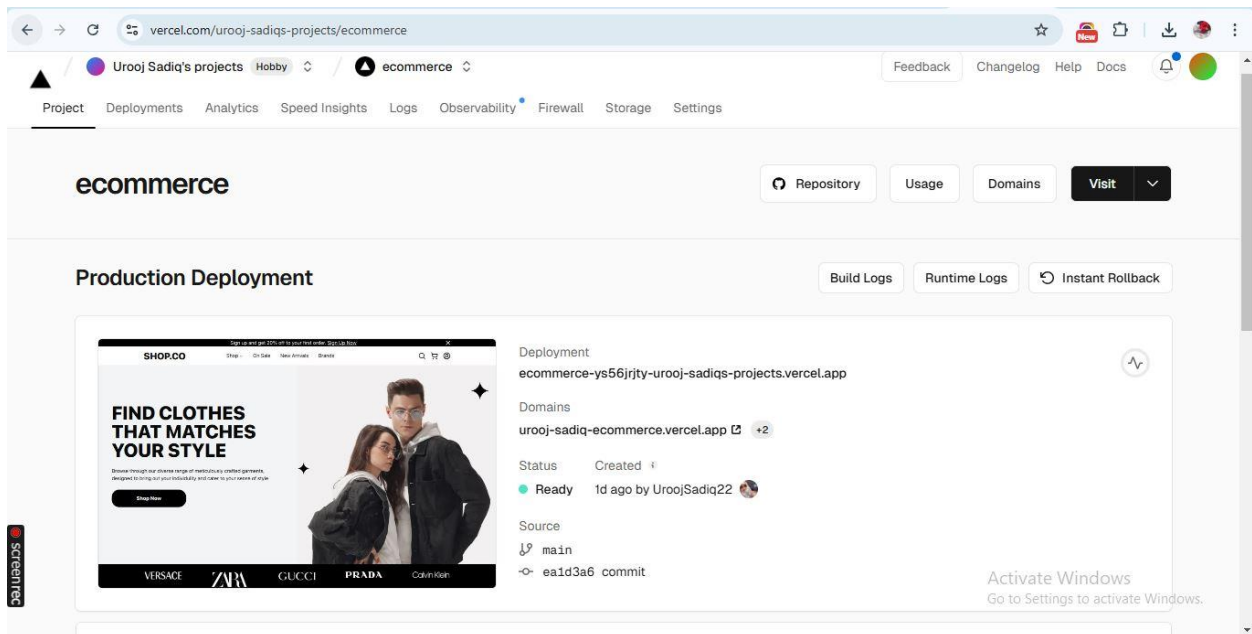
- **Database Migration:** Applied all schema changes to the staging database, ensuring data integrity.

4. Testing Before Final Deployment

- **Functional Testing:** Verified features such as filter, search, add to cart etc
- **Non-Functional Testing:** Assessed performance, security, and usability. Ensured responsiveness across devices and browsers.

5. Final Deployment to Production

- **Data Backup:** Backed up existing production data to avoid data loss.
- **Production Deployment:** Deployed the application using the same workflow as staging to maintain consistency.
- **Monitoring:** Integrated tools to track performance, errors, and user behavior during and after deployment.
- **User Notifications:** Released notes to inform users of new features and updates.



Conclusion:

In conclusion, launching SHOP.COM was the result of careful planning, thorough testing, and step-by-step implementation. From optimizing the code and testing features to preparing the staging environment and monitoring performance, everything was done to ensure the website runs smoothly. With a focus on scalability, security, and user-friendliness, SHOP.COM is ready to handle a growing audience.