

# Urban Loom: Deployment Preparation And Environment Setup



## 1. Objectives:

This document aims to:

- Define a structured deployment process for Urban Loom.
- Ensure a seamless transition from development to production.
- Securely manage environment variables and configurations.
- Establish a systematic testing approach for staging deployment.
- Maintain well-documented records of testing and deployment.

## Introduction:

This document details the deployment strategy and staging environment setup for Urban Loom, ensuring a seamless transition to a production-ready state while adhering to industry best practices.

## 2. Deployment Strategy:

To achieve efficient deployment, the following strategy is implemented:

- Hosting Platform: Vercel (Preferred) / Netlify
- Repository Integration: GitHub repository with automated deployment
- Build & Deployment Configurations: Optimized for consistent staging builds
- Environment Variables Management: Securely handled through the hosting platform

## 3. Environment Setup:

### 3.1 Deployment Preparation Steps:

#### 1: Hosting Configuration:

- hosting provider (Vercel)
- Connect the GitHub repository for continuous integration
- Define deployment settings for a seamless build process

#### 2: Environment Variables Configuration:

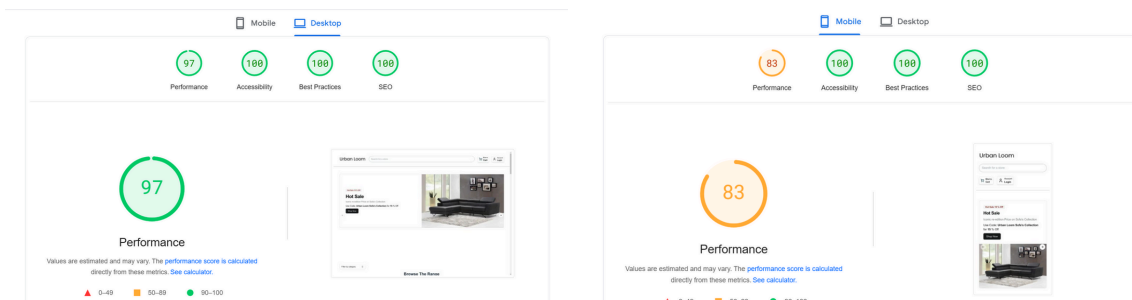
- Securely store API keys, Sanity credentials, and future third-party integrations
- Implement environment variables via the hosting platform

### 3: Staging Deployment:

- Push the latest code to the repository
- Trigger the staging build on the hosting platform
- Validate deployment success and perform initial checks

### 4: Staging Environment Testing:

- Performance Testing: Utilize tools such as Lighthouse to assess speed and responsiveness
- Security Testing: Ensure HTTPS implementation and data protection
- Responsiveness Testing: Verify layout adaptability across multiple devices



### 5: Testing Implementation:

- Functional Testing: Validate product listings, search functionalities, and cart operations.
- Performance Testing: Conduct tests using Lighthouse for speed and responsiveness.
- Security Testing: Ensure secure API communications, validate HTTPS usage, and prevent vulnerabilities.
- Test Case Documentation: Maintain structured CSV reports Like This:

1	A	B	C	D	E	F	G
2							
3	Test Case ID	Test Case Description	Expected Result	Actual Result	Status	Severity Level	Remarks
4	TC001	Validate product listing page	Products displayed correctly	Products displayed correctly	Passed	Medium	No issues found
5	TC002	Performance	Performance Score <90	Score 85	Passed	Medium	Improve needed
6	TC003	Accessibility	Accessability Score <90	Score 100	Passed	High	Test Successful
7	TC004	Seo	Seo score<90	Score 100	Passed	High	Test successful
8	TC005	Add to Cart Funtionality	Item Increase /Decrese/Add and Remove	Result as expected	Passed	High	Successfull funtionality
9	TC006	Best Practices	Practices Score <90	Score 100	Passed	High	Good

### 6: Documentation Management:

- Maintain an up-to-date README.md file with deployment procedures
- Organize project assets within a structured GitHub repository
- Record testing outcomes in a structured CSV format
- Submit a performance report generated via Lighthouse in the repository

#### **4. Expected Outcomes:**

- A fully functional and deployed staging environment
- Secure configuration of environment variables
- Comprehensive test case documentation and performance reports
- A well-organized GitHub repository with structured project documentation

#### **5. Deployment Checklist**

- ✓ Hosting platform successfully configured
- ✓ Environment variables securely implemented
- ✓ Staging deployment verified
- ✓ Functional, performance, and security testing completed
- ✓ Documentation reviewed and stored in GitHub

This document provides a structured framework for deploying Urban Loom



**Thanks for Viewing**

Ayesha Mughal