Hackathon 3: Day_6

DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

Table of Contents

1. Introduction

1.1 Purpose

This document outlines the deployment preparation and staging environment setup for *Comforty Furnitures*, ensuring a structured and efficient transition from development to production.

1.2 Scope

This guide covers hosting selection, environment configuration, staging setup, testing protocols, performance optimization, and final deployment readiness.

2. Deployment Strategy

2.1 Hosting Platform

After evaluating various options, *Vercel* was selected due to its seamless Next.js integration, automated deployments, and scalability.

2.2 Version Control

- **GitHub** is used for source code management.
- A **branching strategy** is implemented with:
  - main branch for production.
  - develop branch for staging.
  - Feature branches for isolated development.

2.3 Deployment Workflow

1. Developers push changes to the feature branch.
2. Changes are merged into develop for staging testing.
3. After testing, successful changes are merged into main and deployed to production.

---

3. Environment Configuration

3.1 Environment Variables

- .env.local file used for storing sensitive information such as API keys and database URIs.
- Separate configurations for **development**, **staging**, and **production**.

3.2 Configuration Management

- **Vercel Environment Variables** configured per environment.
- **Secret Management** for API keys and sensitive credentials.

4. Staging Environment Setup

4.1 Infrastructure

- Dedicated **staging environment** deployed on Vercel using the develop branch.
- Databases, APIs, and third-party integrations mirrored for accurate testing.

4.2 Continuous Deployment

- **Vercel CI/CD** configured for automated deployments.
- PR-based deployments for isolated feature testing.
- Rollback mechanism enabled for instant recovery.

---

5. Testing in Staging Environment

5.1 Functional Testing

- End-to-end testing conducted to validate all functionalities.
- UI/UX verification across multiple devices and browsers.
- Authentication and authorization flow tested.

5.2 Performance Testing

- **Lighthouse** and **GTmetrix** used for performance audits.
- Load and stress tests conducted to assess scalability.

5.3 Security Testing

- API security validation.
- Authentication vulnerability checks.

Secure data transmission ensured with HTTPS enforcement.

6. Performance Optimization

6.1 Asset Optimization

- Image compression using Next.js **Image Optimization API**.
- Browser caching enabled.

6.2 Lazy Loading

- Implemented for images, videos, and non-critical scripts.
- Reduces initial page load time, enhancing user experience.

6.3 Code Splitting

- Dynamic imports in Next.js to improve performance.
- Optimized bundle size for faster page rendering.

---

7. Final Deployment Checklist

7.1 Security Review

- **Cross-Origin Resource Sharing (CORS)** policies validated.
- SSL certificates checked and enforced.
- OWASP security guidelines followed.

7.2 Backup Procedures

- Daily backups configured for database and assets.
- Disaster recovery plan documented.

7.3 Monitoring Setup

- **Google Analytics** and **Vercel Analytics** integrated.
- **Uptime Monitoring** configured for real-time alerts.

---

8. Conclusion

A well-defined deployment strategy and structured staging environment ensure a smooth transition of *Comforty Furnitures* from development to production. By implementing best practices in performance optimization, security, and continuous integration, the platform is equipped for a stable and scalable release.