**Marketplace Builder Hackathon - Day 7: Live Deployment and Post-Launch Practices**

Welcome to the **Marketplace Builder Hackathon 2025** project! This repository contains the code for the Marketplace Builder app developed during the hackathon. In this section, we’ll go through the steps required for **live deployment** and the **post-launch practices** to ensure your app is stable, scalable, and well-maintained after going live.

**Table of Contents**

1. [Overview](#overview)
2. [Pre-Deployment Checklist](#pre-deployment-checklist)
3. [Live Deployment Steps](#live-deployment-steps)
4. [Post-Launch Practices](#post-launch-practices)
5. [Best Practices](#best-practices)
6. [Troubleshooting](#troubleshooting)
7. [Resources](#resources)
8. [License](#license)

**Overview**

This repository is a part of the **Marketplace Builder Hackathon** where we’re building a scalable marketplace application using Next.js. This document provides a detailed guide to deploying the project live, monitoring it, and applying post-launch best practices to ensure the application performs optimally in production.

**Pre-Deployment Checklist**

Before you deploy the Marketplace Builder app to production, make sure you complete the following steps:

1. **Code Review & Testing**
   * Ensure all code is reviewed and passes all unit, integration, and end-to-end tests.
2. **Build the Application**
   * Run the following command to generate a production build:

bash

CopyEdit

npm run build

1. **Set Environment Variables**
   * Ensure that you’ve set up environment variables (such as API keys and database URIs) in the production environment.
2. **Version Control**
   * Make sure all changes are committed and pushed to your Git repository.
3. **Backup Plan**
   * Establish a rollback plan in case the deployment fails.

**Live Deployment Steps**

To deploy the Marketplace Builder app to a live production environment, follow these steps:

**1. Select a Hosting Provider**

Choose a hosting provider like Vercel, Netlify, or AWS to deploy the app. The hosting service should support Next.js and offer automatic scaling.

**2. Set Up the Hosting Environment**

* **DNS Configuration:** Ensure your domain is pointed to the hosting provider's servers.
* **SSL/TLS Setup:** Set up SSL certificates to ensure HTTPS is enabled for your domain.
* **Set Environment Variables:** Configure the production environment variables on the hosting platform.

**3. Continuous Integration / Continuous Deployment (CI/CD)**

* Set up CI/CD pipelines to automate the deployment process using services like GitHub Actions, GitLab CI, or the CI/CD tools provided by your hosting service.

**4. Deployment Command Example**

For **Vercel**, use the following command to deploy the app:

bash

CopyEdit

vercel --prod

**Post-Launch Practices**

After deploying your app, it's crucial to focus on the following practices to ensure it runs smoothly and is easily maintainable:

**1. Monitoring & Logging**

* Use tools like **Sentry**, **Datadog**, or **Vercel Analytics** to monitor errors and performance.
* Set up a centralized logging service such as **Loggly** or **ELK Stack** to capture logs.

**2. Performance Optimization**

* Implement caching strategies using CDNs or server-side caching to reduce load times and improve performance.
* Use profiling tools to identify and fix performance bottlenecks.

**3. Error Handling & Recovery**

* Implement error boundaries and graceful error handling in your app.
* Set up an incident response plan and automated failover mechanisms to quickly recover from any service failures.

**Best Practices**

* **Documentation:** Regularly update your deployment and operational documentation.
* **Security:** Continuously monitor for vulnerabilities and patch any security issues in dependencies or configuration.
* **Regular Audits:** Periodically audit the performance, security, and scalability of your app.
* **User Feedback:** Incorporate feedback from users to improve the app's functionality.

**Troubleshooting**

Here are some common issues you might encounter after deploying the app:

* **Deployment Fails:**  
  If your deployment fails, check the build logs for any error messages. Ensure that all environment variables are correctly set.
* **Performance Issues:**  
  Use profiling tools like **Lighthouse** or **WebPageTest** to find performance bottlenecks. Consider scaling resources or optimizing the codebase.
* **Downtime or Outages:**  
  If your app goes down, check your hosting provider’s dashboard for logs and diagnostics. Follow your incident response plan to restore service.

**Resources**

* Next.js Documentation
* Vercel Documentation
* [GitHub Documentation](https://docs.github.com/en/actions)
* Sanity Documentation

**License**

This project is licensed under the MIT License. See the LICENSE file for details.