

Day 7: Live Deployment and Post-Launch Practices

1. Live Deployment

1.1 Deploying to Vercel

For deploying the project to production, I used Vercel, a platform that integrates seamlessly with Next.js projects. Follow the steps below to deploy your application:

1. Connect GitHub to Vercel:
 - Go to Vercel (<https://vercel.com/>).
 - Log in to your account or create a new one.
 - Connect your GitHub repository by selecting GitHub under the New Project section.
2. Import the Project:
 - Once the GitHub account is connected, Vercel will list all your repositories. Choose the repository containing the Next.js project you want to deploy.
3. Configure Deployment Settings:
 - Vercel will automatically detect the framework (Next.js). However, ensure that the Build Command is set to 'next build' and Output Directory to 'out' (if you're using static export).
 - Set up Environment Variables for production if needed (e.g., API keys, database URLs).
4. Deploy the Project:
 - Once configuration is complete, click on Deploy.
 - The deployment process will take a few minutes. Once finished, you'll be provided with a live URL to access the deployed website.
5. Production URL:
 - Vercel assigns a unique URL (e.g., <https://your-project-name.vercel.app>) where the live website can be accessed.

1.2 Handling Domain Name

1. Connect Custom Domain:
 - After deployment, you can link a custom domain to the Vercel project. Navigate to the Vercel dashboard, select the project, and go to the Domains tab.
 - Enter your custom domain name and follow the instructions to point your domain to Vercel using the provided DNS records.

2. SSL Certificate:

- Vercel automatically provisions an SSL certificate for secure HTTPS traffic once the domain is connected, ensuring your website is served over a secure connection.

2. Post-Launch Practices

2.1 Monitoring and Debugging

1. Use Vercel Analytics:

- After deployment, I recommend using Vercel Analytics to monitor real-time data for my site, including performance, traffic, and server-side function errors.
- It provides insights into user interactions, page load times, and any performance bottlenecks.

2. Error Monitoring:

- Integrate tools like Sentry or LogRocket to capture and log frontend and backend errors.
- These tools help track any bugs that may arise after launch, ensuring faster resolution.

3. Server Logs:

- Check Vercel's logs for any errors in serverless functions or API calls. These logs can be accessed from the Vercel dashboard under the Functions section.

2.2 User Feedback and Optimization

1. User Feedback:

- Collect feedback through forms, surveys, or tools like Hotjar to understand user experience and areas that need improvement.
- Regularly check for bugs or UI issues that could hinder the user experience.

2. Performance Optimization:

- After launch, continuously monitor site speed and performance.
- Use tools like Lighthouse, Google PageSpeed Insights, or WebPageTest to analyze and optimize performance.

3. SEO Practices:

- Ensure all relevant SEO settings (meta tags, structured data) are configured correctly.
- Use tools like Google Search Console to monitor indexing and search performance.

2.3 Routine Updates and Maintenance

1. Version Control:

- Continue committing code updates to GitHub, which will automatically trigger Vercel's deployment process.

2. Content Updates:

- Regularly update content, including blog posts or any dynamic data coming from external sources like Sanity or other APIs.
- Plan a content calendar to keep the website fresh with new content.

3. Security Updates:

- Ensure that security patches and updates are applied to dependencies and packages. This can be done via npm audit or by monitoring Dependabot pull requests.

2.4 Scaling and Traffic Management

1. Scaling:

- Vercel provides auto-scaling, which ensures that the application can handle traffic spikes. Monitor the usage and consider scaling plans if necessary.

2. Caching:

- Use caching strategies (such as HTTP cache headers or service workers) to reduce server load and improve page load times.

3. Conclusion

Deploying the project on Vercel ensures a smooth and efficient transition to production. Post-launch practices like monitoring, optimization, and user feedback collection are crucial to maintain and improve the site's performance over time. By continually updating the application and addressing potential issues, I can ensure a seamless user experience for all visitors.