**HACKATHON DAY 6 DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP**

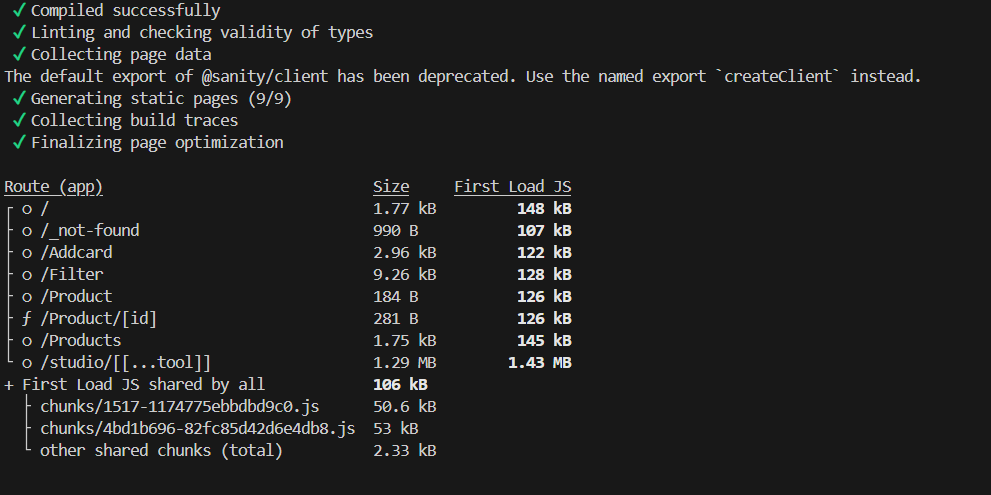
|  |
| --- |
| The project focuses on developing a [product/solution] that aims to address [specific problem or need]. Through collaboration and innovation, we have built a scalable solution utilizing [technologies used, e.g., web development, cloud services, AI]. With successful deployment in the staging environment, the project is poised for launch, offering [key benefits or impact]. Our team is ready to execute the final deployment and ensure a seamless transition to production. |

**HOSTING PLATFORM**

|  |
| --- |
| * The API hosted in vercel, chosen for its key advantages, including: * Global CDN: Enhances content delivery speed worldwide. * Image Optimization: Automatically, optimizes images for improve performance. * Free tire availability... offers caused – effective deployment solution for small to medium projects |

**Deployment Workflow**

To build locally, install dependencies using npm install or pip install -r requirements.txt and set up environment variables. Run the build with npm run build or python manage.py migrate, and resolve errors like missing packages with npm install <missing-package> or handle port conflicts using kill -9 $(lsof -t -i:PORT\_NUMBER). After resolving issues, test the build with npm test or pytest before staging or production deployment.



**Setting up a vercel Project**

To set up a Vercel project, create an account and link your Git repository. Vercel will automatically detect the framework and deploy your app with default settings. You can customize build commands, environment variables, and manage deployments via the dashboard or the Vercel CLI using commands like vercel to deploy and vercel env add to add environment variables. Once deployed, Vercel will handle both preview and production environments.

**Secure information of Environmental Variables**

Environmental variables (env variables) store sensitive information like API keys, database credentials, and configuration settings. To ensure security, keep these variables out of your source code by storing them in environment files (e.g., .env) or using environment management tools provided by hosting platforms (e.g., Vercel, AWS, Heroku). Avoid hardcoding sensitive information directly into the codebase, and use encryption for highly sensitive data when possible. Additionally, ensure the .env file is excluded from version control by adding it to .gitignore to prevent accidental exposure. For team projects, use secret management tools to securely share and access variables across different environments.