Day 6: Deployment and Environment Configuration

On **Day 6**, we successfully deployed the Comforty Marketplace on **Vercel**, a powerful platform for hosting Next.js projects. This document outlines the steps taken to deploy the project and securely manage sensitive data using environment variables.

Deployment on Vercel 🚀

1. Prepare the Project for Deployment

Before deployment, we ensured the project was ready for production:

- **Optimized Code**: Cleaned up the codebase, removed unused code, and ensured all necessary configurations were in place.
- **Build Scripts**: Verified that the Next.js `build` script (`npm run build`) executed without errors.

2. Set Up Vercel Account

- Logged into Vercel using a GitHub account for seamless integration.
- Created a new project by importing the repository from GitHub.

3. Configure Project Settings

- Selected the appropriate **framework preset** (Next.js) during the setup.
- Verified the **default build command** (`npm run build`) and **output directory** (`.next`).

4. Deploy the Project

- Clicked on the "Deploy" button, and within seconds, the project was live at a Vercel-provided URL.

Managing Sensitive Data with `.env` Files 🔐

1. Why Use `.env` Files?

Environment variables are used to store sensitive data like API keys, database credentials, and configuration secrets. Using `.env` files ensures that sensitive data is **not hardcoded** into the application and remains secure.

2. Setting Up `.env` File

1. Created a `.env.local` file in the root of the project.
2. Added all sensitive data as **key-value pairs** in the file:

   ```
   NEXT_PUBLIC_SANITY_PROJECT_ID=your_sanity_project_id
   NEXT_PUBLIC_SANITY_DATASET=your_dataset_name
   NEXT_PUBLIC_SANITY_API_TOKEN=your_api_token
   NEXT_PUBLIC_SHIPMENT_API_KEY=your_shipment_api_key
   NEXT_PUBLIC_PAYMENT_GATEWAY_KEY=your_payment_key
   ```

   > **Note**: Environment variable keys prefixed with `NEXT_PUBLIC_` are exposed to the client-side. Keys without this prefix remain server-side.

3. Accessing Environment Variables in the Code

Environment variables were accessed in the project using `process.env`:

```
const projectId = process.env.NEXT_PUBLIC_SANITY_PROJECT_ID;
const dataset = process.env.NEXT_PUBLIC_SANITY_DATASET;
const apiKey = process.env.NEXT_PUBLIC_SHIPMENT_API_KEY;
```

4. Adding `.env` to `.gitignore`

To prevent sensitive data from being pushed to the repository, we updated `.gitignore`:

```
 Environment Variables

.env.local
.env
.env.*
```

5. Configuring Environment Variables on Vercel

- Navigated to the **Environment Variables** section in the Vercel dashboard.
- Added each variable manually with its corresponding value.
- Verified that the variables were correctly injected into the deployment.

Cross-Origin Resource Sharing (CORS) Configuration 🌐

To allow the frontend to securely communicate with Sanity CMS:

1. Configured **CORS settings** in Sanity CMS to whitelist the Vercel domain.
    - Added the Vercel production URL (e.g., `https://comforty.vercel.app`) to the list of allowed origins.

Testing and Finalization ☑

- **Testing**: After deployment, tested all pages, API routes, and features to ensure they worked as expected.
- **Performance**: Verified the site performance using Lighthouse, achieving 100% performance.
- **Environment Variables**: Confirmed that sensitive data was correctly injected and functioning without exposure.

Key Takeaways

- Deploying on **Vercel** simplifies hosting and scaling Next.js projects.
- Using `.env` files ensures sensitive data is securely managed and protected from being exposed in the codebase.
- Proper **CORS configuration** is essential for smooth integration with external services like Sanity CMS.

Deployment Checklist

- **Deployment Preparation:** ✔
- **Staging Environment Testing:** ✔
- **Documentation:** ✔
- **Form Submission:** ✔
- **Final Review:** ✔