**Day 6: Deployment Preparation and Staging Environment Setup**

### **1. Selecting the Deployment Platform**

* Platform Chosen: Vercel
  + Why Vercel?
    - Vercel is well-known for its seamless integration with modern frontend frameworks.
    - It automatically deploys projects from connected Git repositories.
    - It supports features like preview deployments for every push and smooth production deployments.
  + Additional Note: You appreciated Vercel’s ease of use and its robust features which made it the preferred choice for your project deployment.

### **2. Creating the Repository and Pushing to GitHub**

* **Step 2.1: Initialize Your Local Repository**
  + Open your terminal or command prompt in your project’s root directory.

Run the following command:  
bash  
CopyEdit  
git init

* **Step 2.2: Create and Link a GitHub Repository**
  + On GitHub, create a new repository (e.g., your-repo-name).

Link your local repository to GitHub by running:  
bash  
CopyEdit  
git remote add origin https://github.com/your-username/your-repo-name.git

* **Step 2.3: Stage and Commit Your Files**

Stage all your project files:  
bash  
CopyEdit  
git add .

Commit the changes:  
bash  
CopyEdit  
git commit -m "Initial commit"

* **Step 2.4: Push to GitHub**

Push your code to GitHub:  
bash  
CopyEdit  
git push -u origin main

* + This process ensured that your project’s code was safely stored on GitHub and ready for deployment.

### **Uploading the Project to Vercel and Redirecting for Deployment**

* **Step 3.1: Logging Into Vercel**
  + Sign up for or log into my Vercel account by visiting the Vercel Dashboard.
* **Step 3.2: Importing Your GitHub Repository**
  + Click on **“New Project”** on myr Vercel dashboard.
  + Select GitHub as your Git provider.
  + Authorize Vercel to access my GitHub repositories
  + Locate and select the repository you previously pushed to GitHub.

### **Configure Environment Variables**

**1. Create a .env.local File:**

NEXT\_PUBLIC\_SANITY\_PROJECT\_ID=your\_project\_id

NEXT\_PUBLIC\_SANITY\_DATASET=production

SANITY\_API\_TOKEN

NEXT\_PUBLIC\_CLERK\_PUBLISHABLE\_KEY

CLERK\_SECRET\_KEY

**2. Upload Variables to Hosting Platform:**

* Using Vercel’s dashboard, the environment variables from the .env.local file were securely added to the project settings, ensuring they are available during the build and runtime phases.

### **Challenges Faced During the Deployment Process**

During deployment, you encountered challenges, particularly during the build phase:

* **Issue Encountered:**
  + The build process was hanging indefinitely.
  + Logs indicated that the process stalled during the server start-up phase.
* **Troubleshooting Steps Taken:**
  + **Reviewing Port Binding:**
    - Verified that the application was not explicitly binding to a specific port using methods like app.listen().
    - **Rationale:** Vercel manages the routing and port allocation internally; binding to a specific port can cause the build to hang.
  + **Checking Environment Variables:**
    - Ensured that all required environment variables were correctly set in the Vercel project settings.
    - **Rationale:** Missing or incorrect environment variables can lead to runtime or build issues.
  + **Analyzing Build Logs:**
    - Carefully examined the build logs provided by Vercel to pinpoint the exact stage where the process was stalling.
    - **Rationale:** Logs often reveal missing dependencies, configuration errors, or other issues affecting the build.
  + **Clearing the Build Cache:**
    - Considered clearing the build cache to eliminate potential issues with corrupted or outdated cache data.
    - **Rationale:** Sometimes cached data from previous builds can interfere with the current build process.
* **Outcome:**
  + After implementing these troubleshooting steps, the build issues were resolved, and the project was successfully deployed on Vercel.

### **Step 3: Deploy to Staging**

**1. Deploy Application:**

* **The application was deployed to a staging environment through Vercel.**
* **The deployment was triggered automatically upon linking the GitHub repository, utilizing the pre-configured build settings.**

**2. Validate Deployment:**

* **The build process was monitored via Vercel’s logs to ensure it completed without errors.**
* **Basic functionality was verified in the staging environment, confirming that the application loaded properly and that key features were operational.**