Deployment Strategy and Environment Variable Configuration Report:

1. Deployment Strategy Planning

Objective: To choose an appropriate hosting platform and ensure seamless integration with backend services like Sanity CMS and third-party APIs.

Steps:

1. **Choose a Hosting Platform:**
    - **Vercel (Recommended):** Ideal for modern frameworks like Next.js and React. It provides automatic build and deployment pipelines, making it developer-friendly.
    - **Netlify:** Suitable for static sites with support for serverless functions. Provides free-tier options for small projects.
    - **AWS/Azure:** Best for applications requiring advanced configurations, scalability, and custom setups. These platforms offer fine-grained control but require expertise to manage.
2. **Finalize Backend Integration:**
    - **Sanity CMS:**
        - Configure the CMS for dynamic content management.
        - Use its APIs for fetching and updating data in the application.
    - **Third-Party APIs:**
        - Identify required APIs (e.g., payment gateways, geolocation services).
        - Ensure API authentication keys are securely handled.

Action Points:

- Select a hosting platform that aligns with project requirements.
- Test backend service integrations locally before deploying to staging.

---

2. Environment Variable Configuration

Objective: To securely manage sensitive data like API keys, database credentials, and other critical configurations.

Steps:

1. **Create and Use `.env` Files Locally:**
    - Store sensitive data in a `.env` file at the root of the project.

Example of `.env` file:
NEXT_PUBLIC_API_URL=https://api.example.com
DATABASE_PASSWORD=super secret password
STRIPE_SECRET_KEY=sk_test_xxxx

Add `.env` files to `.gitignore` to prevent them from being pushed to version control.
.env

2. **Configure Environment Variables in the Hosting Platform:**
    - Go to the hosting platform's project settings.
    - Add environment variables manually in the platform's dashboard.
        - **Vercel:**
            - Navigate to Project Settings > Environment Variables.
            - Add variables like:
                - `NEXT_PUBLIC_API_URL`
                - `DATABASE_PASSWORD`
        - **Netlify:**
            - Navigate to Site Settings > Build & Deploy > Environment.
            - Add variables similarly.

Best Practices:

- Use `NEXT_PUBLIC_` prefix for variables that need to be accessible on the client side (e.g., public APIs).
- Rotate sensitive keys periodically to ensure security.

---

3. Maintain Clear Processes Across Environments

Objective: To follow a systematic approach for moving changes across different environments (DEV → SIT → UAT → PROD).

Steps:

1. **Development (DEV):**
    - **Purpose:** Code is written and tested locally.
    - **Process:**
        - Write and test new features or fixes.
        - Use local `.env` files for configuration.

        - Perform unit testing to ensure functionality.
2. **System Integration Testing (SIT):**
    - **Purpose:** Validate interactions between different systems or modules.
    - **Process:**
        - Deploy the application to a SIT environment.
        - Test API integrations, database connections, and other services.
        - Log and resolve any system-level issues.
3. **User Acceptance Testing (UAT):**
    - **Purpose:** Allow stakeholders and end-users to validate requirements.
    - **Process:**
        - Deploy to the UAT environment.
        - Perform testing based on user scenarios and scripts.
        - Collect feedback and make necessary adjustments.
4. **Production (PROD):**
    - **Purpose:** The live, customer-facing environment.
    - **Process:**
        - Deploy the application after successful UAT.
        - Monitor logs for real-time issues.
        - Ensure performance optimization and high availability.

Best Practices:

- Automate deployments using CI/CD pipelines for consistent and error-free transitions.
- Maintain a rollback plan in case of critical failures.

Summary:

- **Deployment Strategy:**
    - Choose Vercel for ease of use or other platforms like Netlify/AWS for specific needs.
    - Ensure smooth integration with backend services.
- **Environment Variable Configuration:**
    - Use `.env` files locally for sensitive data.
    - Configure hosting platforms to securely manage environment variables.
- **Clear Process Maintenance:**
    - Systematically move changes across environments (DEV → SIT → UAT → PROD).

- Perform thorough testing at each stage.

This Reports Made By:

- **Name:** Aziza Siddiqui
- **Student ID:** 00200937
- **Institute:** GIAIC