DAY 2 PLANNING THE TECHNICAL FOUNDATION

Frontend:

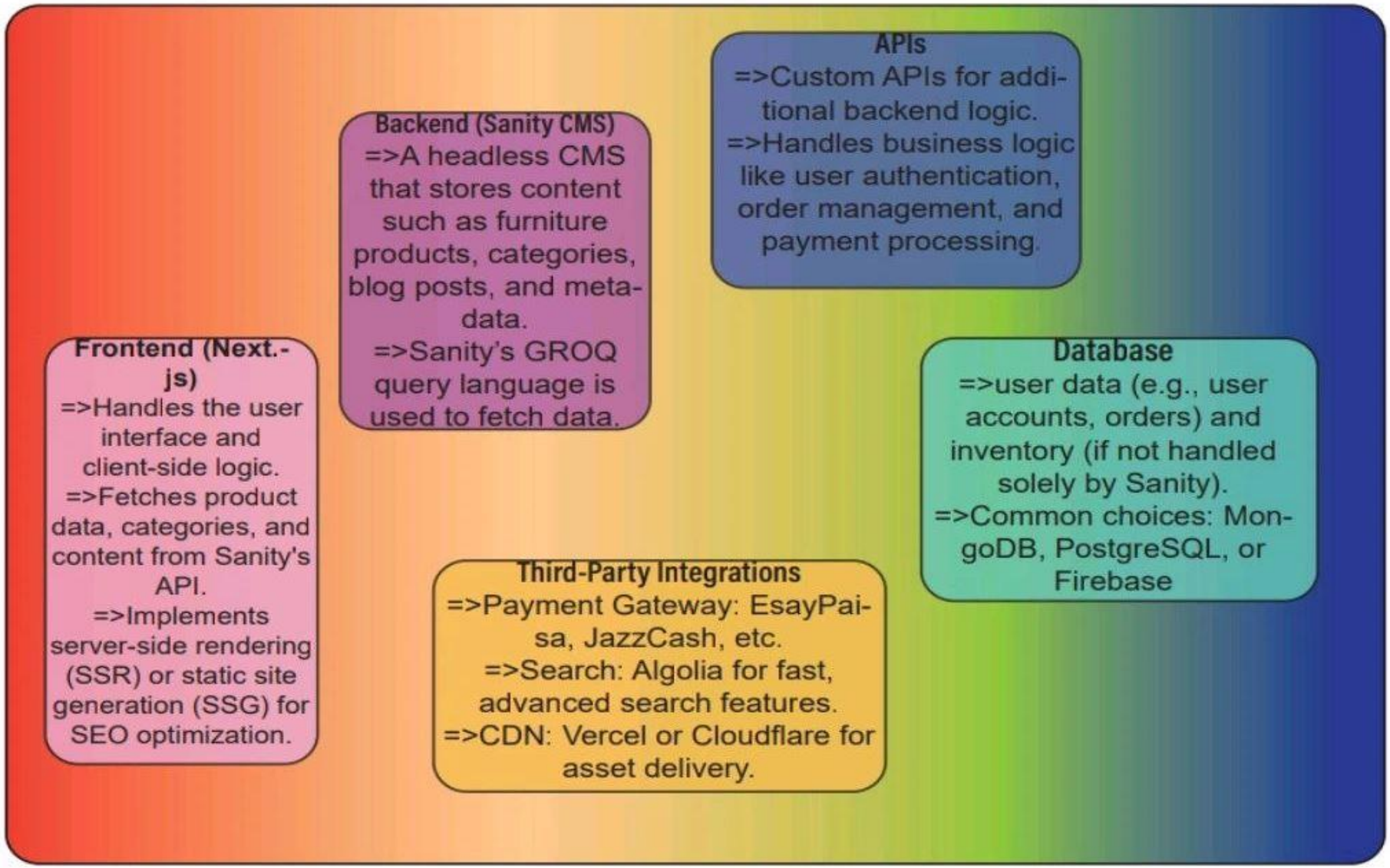
- **Framework:**
  - **React:** A popular JavaScript library for building dynamic and interactive user interfaces.
  - **Next.js:** A React framework that provides features like server-side rendering, static site generation, and file-based routing, which can improve performance and SEO.
- **CSS Libraries/Frameworks:**
  - **Tailwind CSS:** A utility-first CSS framework that allows for rapid development and customization.
  - **Material UI:** A popular UI component library that provides pre-built, customizable components.
- **State Management:**
  - **Redux Toolkit:** Simplifies Redux development with features like createSlice and createAsyncThunk.
  - **Zustand:** A lightweight and performant state management library.

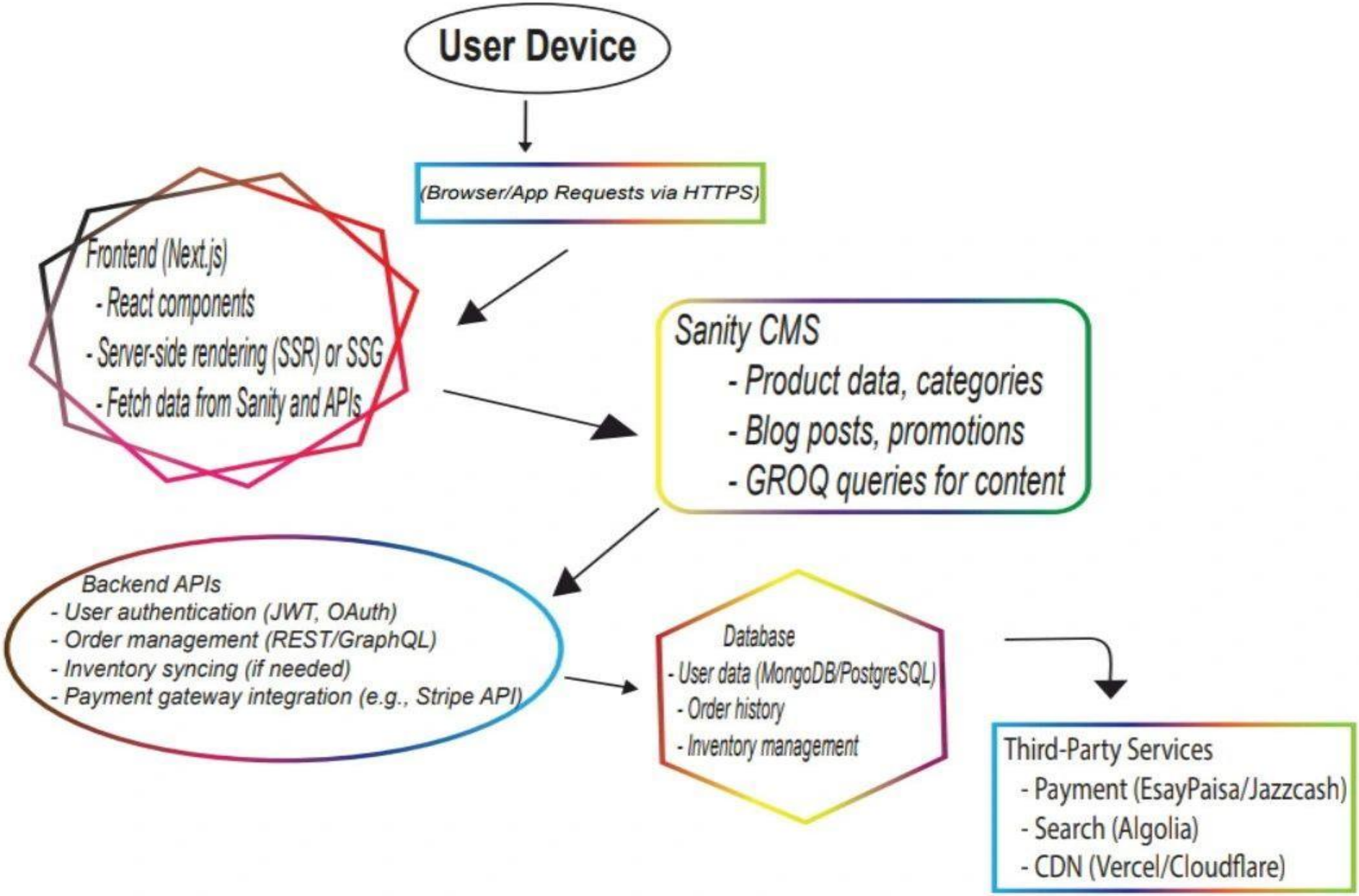
Backend (Sanity CMS):

- **Content Modeling:** Define content models (schemas) for products, categories, pages, blogs, etc. using Sanity's intuitive schema builder.
- **Data Fetching:** Utilize Sanity's GraphQL API or GROQ query language to fetch data for the frontend.
- **Content Editing:** Utilize Sanity's user-friendly content editor to create and manage website content.
- **Image and Media Handling:** Leverage Sanity's built-in image and media management features.
- **Version Control:** Utilize Sanity's version control features to track changes and revert to previous versions if needed.

Third-Party APIs:

- **Payment Gateways:** Integrate with payment gateways like Stripe, PayPal, or Square to process online payments securely.
- **Shipping & Logistics:** Integrate with shipping providers like FedEx, UPS, or USPS to calculate shipping costs and track orders.
- **Search:** Integrate with a search engine like Algolia or Meilisearch to provide fast and relevant search results.
- **Analytics:** Integrate with Google Analytics or other analytics platforms to track website traffic and user behavior.
- **Marketing & Advertising:** Integrate with marketing automation platforms or advertising networks to run targeted campaigns.


APIs
=>Custom APIs for additional backend logic.
=>Handles business logic like user authentication, order management, and payment processing.
Backend (Sanity CMS)
=>A headless CMS that stores content such as furniture products, categories, blog posts, and metadata.
=>Sanity's GROQ query language is used to fetch data.
Frontend (Next.-js)
=>Handles the user interface and client-side logic.
=>Fetches product data, categories, and content from Sanity's API.
=>Implements server-side rendering (SSR) or static site generation (SSG) for SEO optimization.
Database
=>user data (e.g., user accounts, orders) and inventory (if not handled solely by Sanity).
=>Common choices: MongoDB, PostgreSQL, or Firebase
Third-Party Integrations
=>Payment Gateway: EsayPaisa, JazzCash, etc.
=>Search: Algolia for fast, advanced search features.
=>CDN: Vercel or Cloudflare for asset delivery.


User Device
(Browser/App Requests via HTTPS)
Frontend (Next.js)
- React components
- Server-side rendering (SSR) or SSG
- Fetch data from Sanity and APIs
Sanity CMS
- Product data, categories
- Blog posts, promotions
- GROQ queries for content
Backend APIs
- User authentication (JWT, OAuth)
- Order management (REST/GraphQL)
- Inventory syncing (if needed)
- Payment gateway integration (e.g., Stripe API)
Database
- User data (MongoDB/PostgreSQL)
- Order history
- Inventory management
Third-Party Services
- Payment (EsayPaisa/Jazzcash)
- Search (Algolia)
- CDN (Vercel/Cloudflare)