**System Architecture - Day 2**  
**Marketplace Technical Foundation for Software House**

**1. System Architecture Document Overview**

This document outlines the **system architecture** for the software house’s website, detailing interactions between the **frontend, Sanity CMS, and third-party APIs**. The architecture is designed to ensure scalability, maintainability, and seamless integration of multiple services.

**Components of the System:**

* **Frontend:** Developed using **Next.js with Tailwind CSS**, ensuring a responsive and dynamic user experience.
* **Sanity CMS:** Acts as a **backend content management system** for handling dynamic content like blogs, portfolios, and product information.
* **Third-Party APIs:** Integrates external services for functionalities such as **payment processing, live chat, CRM, analytics, and shipment tracking**.

**2. Roles of Components**

**Frontend (Next.js & Tailwind CSS):**

* Handles **user input** and **renders data** fetched from Sanity CMS or third-party APIs.
* Provides a **seamless and responsive** UI/UX.
* Implements **dynamic routing, server-side rendering (SSR), and static site generation (SSG)** for performance optimization.

**Sanity CMS (Content Management System):**

* Stores and manages **structured content** for the website.
* Acts as the **single source of truth** for blogs, portfolios, and dynamic sections.
* Provides an **intuitive dashboard** for content management.

**Third-Party APIs:**

* **Payment APIs (e.g., Stripe):** Securely processes online transactions.
* **Communication APIs (e.g., Twilio):** Enables real-time messaging and notifications.
* **Analytics APIs (e.g., Google Analytics):** Tracks user behavior and performance metrics.
* **CRM APIs:** Manages customer relationships and engagement.
* **Project Management APIs:** Supports collaboration and workflow tracking.
* **Shipment Tracking APIs:** Allows real-time tracking of orders and deliveries.

**3. System Workflow**

The system workflow outlines how various components interact:

* **User Interaction:** The user interacts with the frontend, browsing products, blogs, or services.
* **Data Fetching:** The frontend requests content from **Sanity CMS**.
* **User Actions:** If a user makes a purchase, the **payment API** handles the transaction.
* **Data Processing:** The **CRM API** logs customer interactions.
* **Analytics & Tracking:** The **analytics API** monitors user behavior, while the **shipment API** tracks deliveries.

**Workflow Diagram**

+-------------+ +------------+ +-------------+ +--------------+

| User | ---> | Frontend | ---> | Sanity CMS | ---> | Third-Party |

| Interaction | | (Next.js) | | (CMS) | | APIs |

+-------------+ +------------+ +-------------+ +--------------+

| | | |

v v v v

Browse Products Fetch Data Manage Content Process Payments,

Place Orders Render UI Store & Serve Data Track Shipments,

Track Orders Handle Input Update Listings Log Analytics



**4. Technical Roadmap**

**Milestones:**

* **Frontend Development**
* Design and implement the website using **Next.js and Tailwind CSS**.
* Integrate dynamic content fetching from **Sanity CMS**.
* **Sanity CMS Integration**
* Set up schemas for **users, projects, services, portfolios, blogs, FAQs, and careers**.
* Configure **Sanity dashboard** for seamless content management.
* **Third-Party API Integration**
* Integrate **payment processing APIs** (e.g., Stripe).
* Implement **communication and engagement APIs** (e.g., Twilio).
* Connect **CRM, scheduling, calendar, and project management APIs**.
* Enable **shipment tracking** for logistics support.
* **Testing and QA**
* Perform **unit testing** for API endpoints.
* Conduct **end-to-end testing** for full user workflows.

**5. Deployment Strategy**

* Deploy the website on **Vercel** for **scalability and performance**.
* Implement **continuous monitoring and optimization**.
* Ensure proper **documentation and testing reports** before release.

**6. Deliverables**

* A **fully functional, responsive** marketplace website.
* Integrated **CMS and third-party APIs**.
* **Comprehensive documentation** for future enhancements.

This system architecture ensures a **scalable, high-performing, and user-friendly** marketplace platform. 🚀