EVOLVING PIRANHA CMS FOR "CONTENTS'R'US"

Background

Contents'r'us is a technology provider specializing in both headless and traditional Content Management Systems (CMS). Over the past several years, they've built a robust client base by delivering flexible, easy-to-integrate solutions for businesses of all sizes. The cornerstone of their offering is a solution derived from the open-source **Piranha CMS**.

However, market conditions and client demands have shifted. Some clients now seek multi-tenant setups to streamline operating costs, others require advanced editorial workflows to manage content lifecycles more effectively, and still others want an event-driven architecture for integrating with a growing ecosystem of external services. Above all, **performance** and **scalability** have emerged as critical drivers: the system must handle large spikes in usage without degrading the user experience.

Your team has been contracted by Contents'r'us to **re-envision** portions of this platform. The assignment is to **analyze**, **redesign**, **and implement** architectural changes based on one of three possible scenarios, each reflecting strategic choices and long-term goals for the company.

Scenarios

Below are three future directions under consideration. In every scenario, **performance** and **scalability** remain top priorities. Each scenario also has its own **Vision & Strategic Goals** to guide decisions.

Scenario 1: Multi-Tenant Expansion

Vision & Strategic Goals

- **Vision**: Offer a robust, cost-effective **SaaS** model enabling multiple distinct clients (tenants) to operate securely on a single shared platform.
- Strategic Goals:
 - 1. Achieve strict data isolation between tenants to foster trust and meet compliance needs.
 - 2. Simplify tenant onboarding so new clients can self-provision in minimal time.

3. Ensure the platform scales horizontally to accommodate a growing customer base without compromising performance.

Scenario 2: Editorial Workflows & Content Approvals

Vision & Strategic Goals

- **Vision**: Become a leader in editorially driven content management by offering advanced **approval workflows** and governance features for large teams.
- Strategic Goals:
 - 1. Introduce multi-stage content states and permissions, giving organizations fine-grained control over publishing.
 - 2. Streamline the handoff from creators to reviewers, legal teams, and final approvers, in custom tailored flows.
 - 3. Maintain high performance even with large editorial teams simultaneously creating and publishing content.

Scenario 3: Event-Driven Extensions & Secure Integrations

Vision & Strategic Goals

- Vision: Transform the CMS into a modern, asynchronous platform that supports inbound and outbound data flows, integrates easily with external services, and leverages secure messaging.
- Strategic Goals:
 - Introduce a custom publish/subscribe model for domain events (e.g., content creation, updates, deletions) - where the admins may define if a given model will be published and/or receive information through subscription.
 - Support inbound events from authorized external publishers, allowing third parties to trigger CMS actions. have a way to configure the external publishers
 - 3. Ensure on all message flows (inbound and outbound) authenticity and integrity, and provide a clear setup process for end users to configure keys, endpoints, and permissions.

Assignment Structure

You will deliver **two main parts**: (1) Documentation & Design, and (2) Implementation & Demonstration. Select **one** scenario and proceed accordingly, keeping in mind the company's overarching emphasis on **performance**, **scalability**, **and reliability**.

Part 1: Documentation & Design

1. System Analysis & Project Vision

- Outline how the existing Piranha CMS aligns with (or diverges from) your chosen scenario's objectives.
- Summarize the **Vision & Strategic Goals** for your scenario, highlighting the key quality attributes you consider vital.

2. Choose an Architectural Design Methodology

- Select a design methodology or framework (e.g., Architecture-Centric Design Method, TOGAF ADM, Attribute-Driven Design).
- Document why you chose it and how you will apply it throughout your architectural decisions.

3. Identify and Model Core Domains

- Utilize Domain-Driven Design (DDD) techniques to identify the core domains relevant to your scenario.
- You may wish to model **User Management**, **Betting Transactions**, **Live Event Updates**, or other domains if they are pertinent to your approach.
- Emphasize loose coupling and high cohesion, ensuring each domain is clearly defined and interacts with others in a robust manner.

4. Address Cross-Cutting Concerns

- Determine which **cross-cutting concerns** (e.g., security, logging, error handling) are relevant to your chosen scenario.
- Incorporate strategies for managing these concerns seamlessly within the new architecture.

5. Proposed Architecture & Roadmap

- Present an overview of the envisioned system structure, indicating how you plan to incorporate scenario-specific features.
- Provide **diagrams** or **high-level descriptions** that communicate your intended design, but refrain from detailing specific implementation steps.

 Outline the two or three key features you aim to deliver as part of this architectural evolution.

Part 2: Implementation & Demonstration

1. Refactoring & Feature Development

- Update the Piranha CMS codebase to realize the features identified in your documentation.
- Implement any additional modules, APIs, or data models required for your chosen scenario (multi-tenancy, editorial workflows, or event-driven integrations).

2. Performance & Scalability Considerations

- Demonstrate how the system can handle increased load or complex usage scenarios without degrading service.
- Show evidence of **scalability** measures: for example, caching, partitioning strategies, asynchronous processing, or resource auto-scaling.

3. Final Configuration & Security Setup

- Provide a simple configuration approach so **end users** (e.g., tenant admins or system integrators) can manage keys, endpoints, and permissions relevant to your scenario.

4. Demonstration of the Evolved System

- Prove that your changes function correctly by showcasing the **new features**.
- Illustrate how these modifications address the Vision & Strategic Goals you outlined.
- The nature of your demonstration is flexible; any format that clearly communicates business value and architectural improvements is acceptable.

5. Repository & Traceability

- Maintain a **version-controlled repository** reflecting all notable changes, commits, and updates.
- Include any **build/run instructions** and notes on configuration so that others can reproduce your environment.

Useful links

Piranha CMS: https://github.com/PiranhaCMS/

Deadline Dates

Delivery Part 1: Documentation + presentation: 6th and 7th May Delivery Part 2: Code + report + presentation: 3rd and 4th June

Recap on evaluation

Final Exam: 30% => Regular exam season Assignment 1: 10% => eshop app (done)

Final Assignment - Part 1: 10% => Documentation & Design

Final Assignment - Part 2: 50% => Implementation & Demonstration