SmartCourse — Intelligent Course Delivery Platform

EduCorp is building **SmartCourse**, an intelligent, large-scale learning platform designed to support modern digital education for universities, enterprises, and training academies. The company is experiencing rapid growth in enrolled learners and instructors, which has exposed several limitations in their current systems:

1. **Content publishing is slow and manual**, making it difficult for instructors to launch new courses and update existing ones.
2. **Students struggle to find relevant information**, as the platform lacks intelligent search, contextual assistance, and adaptive learning support.
3. **Course data, user progress, and analytics are scattered**, causing inconsistencies between reporting dashboards and the actual state of the platform.
4. **High user traffic** results in delays when processing large volumes of enrollments, notifications, and background tasks.
5. **Course interactions generate rich but underutilized data**, which is not being leveraged for recommendations or learning enhancement.

To address these challenges, EduCorp is commissioning a new backend for SmartCourse with the following **business goals**:

Business Goals & Vision

SmartCourse must provide:

1. **A robust course management system**
    - Instructors create courses, define modules, upload learning materials, and publish updates.
    - Students browse courses, enroll, track their learning progress, and interact with content.
2. **A scalable and reliable operations backbone**
    - Publishing a course triggers multiple internal processes such as indexing, content extraction, and preparation for intelligent search.
    - Enrollment triggers progress initialization, analytics updates, and notifications.
3. **Consistent and accurate learner data**
    - The state of enrollments, progress, completions, and certificates must be reliable, durable, and easy to query.
4. **Intelligent learning experiences**

    - Students must be able to ask questions about course material and receive context-aware answers.
    - Instructors must be able to auto-generate summaries, quizzes, and explanations to enhance teaching materials.
5. **Smooth real-time interactions**
    - Responses to students' questions may involve long-running reasoning or content generation.
    - Responses should not block the system or create delays for other users.
6. **A foundation that supports long-term scalability**
    - As the platform grows, SmartCourse must handle tens of thousands of learners concurrently, along with spikes during course launches or corporate training schedules.
    - Background workflows should run reliably even under heavy load.

Core Functional Requirements

1. Course & User Management

The system should support:

- Creation and updating of courses, modules, and learning assets
- User registration with appropriate roles (student/instructor/admin)
- Student enrollment into courses, including rules for:
    - Duplicate enrollments
    - Enrollment limits or prerequisites
    - Enrollment history

Each update or enrollment must ensure consistency across all parts of the system.

---

2. Content Publishing Workflow

When an instructor publishes or updates a course:

- The content must be analyzed and broken into components (modules, lessons, chunks).
- Relevant data should be stored in a structure that supports:
    - Fast retrieval
    - Search
    - Context-based queries
- The platform must mark the course as "ready" once all internal processing completes.
- Partial failures must not corrupt the publishing workflow.

---

3. Enrollment Workflow

When a student enrolls in a course:

- Their enrollment is recorded.
- Their progress tracking is initialized.
- Analytics records must be updated to reflect platform activity.
- Notifications may need to be triggered (e.g., “Welcome to the course”).

This workflow must handle:

- High volume
- Idempotency (avoiding duplicated enrollments or duplicated analytics updates)
- Backpressure handling
- Recovery from failures without losing user state

---

4. Intelligent Learning Assistant

SmartCourse must provide an AI-driven learning assistant with two capabilities:

A. Contextual Q&A

- Students ask questions related to course content.
- The system retrieves relevant course sections.
- The assistant generates a meaningful, contextually correct answer.

B. Content Enhancement for Instructors

- Instructors request summaries, objectives, or quiz questions.
- The assistant generates them based on course material.
- Responses may be long and should support incremental/streamed delivery.

The assistant must behave consistently even when dealing with incomplete or ambiguous questions.

---

5. Distributed & Event-Driven Behaviors

Certain actions in SmartCourse trigger multiple dependent tasks, such as:

- Content processing after publishing

- Updating analytics after enrollment
- Sending notifications
- Preparing course material for intelligent Q&A

These tasks should:

- Run independently from the main user flows
- Be traceable and recoverable
- Handle failures gracefully
- Avoid double-processing
- Support workload spikes

---

6. Analytics Metrics for SmartCourse

- Total Students – Number of active learners on the platform.
- Total Instructors – Number of active instructors managing courses.
- Total Courses Published – Count of courses that are currently available to students.
- New Enrollments Over Time – Number of student enrollments per day/week/month.
- Course Completion Rate – Percentage of students who complete a course after enrolling.
- Average Time to Complete a Course – Mean duration from enrollment to completion.
- Most Popular Courses – Courses with the highest number of enrollments or engagement.
- Average Courses per Student – Average number of courses each student is enrolled in.
- AI Assistant Usage – Number of questions asked, answered, and type of assistance used (contextual or generated).
- Failed Events / Workflow Issues – Count of failed background tasks, notifications, or events in the system.

---

7. System Observability & Reliability Expectations

SmartCourse must provide:

- Clear separation of responsibilities between components
- Monitoring and logging for all key flows
- Ability to diagnose failures in:
    - Course publishing
    - Enrollment progression
    - Intelligent assistant interactions
    - Background tasks

High consistency and accuracy across all data models is a strict requirement.

Tech Stack

- **Backend**
  - Python, FastAPI
  - PostgreSQL
  - NoSQL DB
  - Redis
  - Celery Workers with RabbitMQ
  - Kafka + Schema Registry
  - Temporal (workflows)
- **AI**
  - LangGraph
  - LLM Provider (OpenAI/Groq/Anthropic)
  - Vector DB (Choose the best acc to need
- **Observability & Monitoring**
  - Prometheus + Grafana
  - Jaeger
  - OpenTelemetry
- **DevOps**
  - Docker
  - Docker Compose

Expected Outcomes

By the end of the assignment, the backend system for SmartCourse should:

1. You must also create a **Product Requirements Document (PRD)** outlining:
   - Key use-cases
   - Functional and non-functional requirements
   - A proposed implementation timeline/milestones
   - Clear traceability between features, requirements, and deliverables

2. **Support all major course lifecycle operations**
   from creation to publishing to student engagement.

3. **Handle background operations reliably**
   ensuring that publishing, indexing, analytics, and notification pipelines function without data loss or duplication.

4. **Provide a seamless intelligent learning assistant**
   capable of answering student questions and enhancing instructor content.

5. **Scale to increasing loads**
   both in real-time API interactions and background workflows.

6. **Maintain high architectural quality**, evaluated on:
   - Clean code and maintainability
   - Proper application of design principles
   - Use of appropriate design patterns
   - Robust workflow orchestration
   - Stability under failure conditions
   - Readability of documentation and diagrams
   - Test coverage and test quality
7. Additionally, this exercise is being given to upskill you, meaning the goal is not only to complete the required milestones, but also to deeply learn and understand the core concepts of the architectural patterns, design principles, and the underlying tech stack involved.