

## خطة تنفيذ المشروع

تهدف هذه المنصة إلى توفير رحلة تعليمية سلسة، مما يسمح للطلاب بالتسجيل والالتحاق بالدورات ومتابعة تقدمهم. يتتوفر للمعلمين الأدوات الازمة لإنشاء محتوى مرجعي مسجل و بث مباشر للدورات وإدارته مع إمكانية الدردشة مع الطلاب والإجابة عن استفساراتهم ، بينما يقوم المسؤول عن النظام بالتحكم المركزي من خلال لوحة تحكم إدارية قوية يستطيع من خلالها إخراج تقارير و رفع دورات واضافة معلمين.

**الخدمات المطلوب توفرها في النظام:**

الهدف الأساسي	الخدمة
	<b>API Gateway</b>
	<b>Auth Service</b>
	<b>User Service</b>
	<b>Course Service</b>
	<b>Enrollment Service</b>
	<b>Content Service</b>
	<b>Streaming Service</b>
	<b>Chat Service</b>

	<b>Notification Service</b>
	<b>Payment Service</b>

Tech stack:

1- الواجهة الخلفية (**Backend**)

1. اللغة والمنصة:

**Node.js** •

2. إطار العمل:

(لبناء خدمات RESTapi/gRPC بشكل منظم) **Express.js** •

3. التصميم والبنية:

• تصميم قائم على الخدمات الصغيرة (**Microservices**)

(**Domain-Driven Design (DDD)** • تقسيم النظام إلى نطاقات خدمات مستقلة)

• كل خدمة تمتلك قاعدة بيانات مستقلة (Database per Service)

#### 4. بوابة API:

• AWS API Gateway, Kong, NGINX  
(التوحيد نقطة الدخول، وإدارة المصادقة، وتحديد المعدلات، والمراقبة)

#### 5. الاتصال بين الخدمات:

• REST أو gRPC (مبدئياً في احتمال إضافة اتصال قائم على الأحداث)

#### 6. الأمان:

• OAuth2 / JWT للتحقق من الهوية

• HTTPS لتسفير

• RBAC (التحكم في الوصول بناءً على الصلاحيات لكل مستخدم)

---

الواجهة الأمامية (Frontend)

#### 1. المكتبة الرئيسية:

• React.js

2. المعمارية:

## Component-Driven Architecture •

• أدوات: Vite

3. الانتقال بين صفحات الواجهة:

## React Router •

4. إدارة الحالة:

• في الأغلب بنستخدمo redux لي يجيء مع react او نستخدمo useContext على حسب

5. جلب البيانات من الـ : backend

(REST APIs) لـ Axios •

6. التنسيق والتصميم:

Tailwind CSS, styled-components •

- إمكانية استخدام نظام تصميم UI جاهز مثل shad cn

## الخطة الزمنية:

### المرحلة 1: تحليل المتطلبات والتخطيط (الأسبوع 1 - 2)

- جمع متطلبات المشروع من الأطراف المعنية (الطلاب، المعلمين، المسؤولين).

- رسم .class diagram, ERD, Use Cases,

- تحديد نطاق النظام والخدمات المطلوبة.

- وضع خطة العمل وجدولة المهام باستخدام أدوات مثل Trello أو Jira.

### المرحلة 2: التصميم المعماري (الأسبوع 3 - 4)

- إعداد البنية المعمارية للنظام باستخدام Microservices.

- تحديد قواعد البيانات لكل خدمة وتصميمها.

- تحديد هيكلة الـ api.

- تحديد الحماية للتصرิحات (JWT) وتحطيط RBAC.

### المرحلة 3: تطوير الخدمات الأساسية (الأسبوع 5 - 8)

- إنشاء API Gateway.

- تطوير Auth Service, User Service

- تطوير Course Service ,Enrollment Service .
- ربط الخدمات ببعضها عبر REST/gRPC.
- المرحلة 4: تطوير خدمات المحتوى (الأسبوع 9 - 11)
  - بناء Content Service لإدارة الفيديوهات.
  - إعداد Streaming Service للبث المباشر.
- بناء Chat Service باستخدام WebSockets.
- المرحلة 5: تطوير الخدمات المساعدة (الأسبوع 12 - 13)
  - بناء Notification Service (للإشعارات).
  - بناء Payment Service (توليد الكروت والتحقق منها).
  - بناء Reporting Service (توليد تقارير تفاعلية).
  - بناء Search Service (للبحث).
- المرحلة 6: واجهة المستخدم (الأسبوع 14 - 16)
  - بناء الواجهة بـ React.js .
  - استخدام React Router .
  - تنفيذ واجهة المعلم، الطالب، ومالك النظام.

الباقي الاختبارات والتحسين (كان خدمنا الفوقي كلة صح تقدر المرحلة هذي أسبوع)

المجموع الكلي: 17 أسبوع = حوالي 4 أشهر