

خطة تنفيذ المشروع

تهدف هذه المنصة إلى توفير رحلة تعليمية سلسلة، مما يسمح للطلاب بالتسجيل والالتحاق بالدورات ومتابعة تقدمهم. يتوفر للمعلمين الأدوات اللازمة لإنشاء محتوى مرئي 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 و اضافة معلمين.

الخدمات المطلوب توفرها في النظام:

الهدف الأساسي	الخدمة
	API Gateway
	Auth Service
	User Service
	Course Service
	Enrollment Service
	Content Service
	Streaming Service
	Chat Service

	Notification Service
	Payment Service

Tech stack:

1- الواجهة الخلفية (Backend)

1. اللغة والمنصة:

• Node.js

2. إطار العمل:

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

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** أو
(لتوحيد 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** لتشفير
- **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 الاغلب بنستخدمو **useContext** لي يجيه مع **react** او نستخدمو **redux** على حسب

5. جلب البيانات من الـ **backend** :

• **Axios** (REST APIs)

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 و class diagram, ERD, Use Cases.
- تحديد نطاق النظام والخدمات المطلوبة.

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

المرحلة 2: التصميم المعماري (الأسبوع 3 - 4)

- إعداد البنية المعمارية للنظام باستخدام Microservices.
- تحديد قواعد البيانات لكل خدمة و تصميمها.
- تحديد هيكل الـ api.

- تحديد الحماية للتصاريحات (JWT) وتخطيط 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)

- Notification Service (للاشعارات).

- Payment Service (توليد الكروت والتحقق منها).

- Reporting Service (توليد تقارير تفاعلية).

- 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 تقعد المرحلة هذي أسبوع)
المجموع الكلي: 17 أسبوع = حوالي 4 أشهر