## ****פרויקט מתקדם: מערכת LMS לניהול למידה מקוונת (Learning Management System)****

✅ **פרויקט מתקדם הכולל ניהול משתמשים, הרשאות, קורסים, הרשמות, חומרי למידה, העלאת קבצים, חידונים ומערכת ציונים.**

## ****📂 מבנה הפרויקט****

LMS-System/

│── backend/

│ ├── models/

│ │ ├── User.js

│ │ ├── Course.js

│ │ ├── Enrollment.js

│ │ ├── Quiz.js

│ │ ├── Submission.js

│ ├── routes/

│ │ ├── authRoutes.js

│ │ ├── courseRoutes.js

│ │ ├── enrollmentRoutes.js

│ │ ├── quizRoutes.js

│ │ ├── submissionRoutes.js

│ ├── middleware/

│ │ ├── authMiddleware.js

│ ├── server.js

│ ├── .env

│── frontend/

│ ├── index.html

│ ├── login.html

│ ├── register.html

│ ├── dashboard.html

│ ├── courses.html

│ ├── course-detail.html

│ ├── quiz.html

│ ├── submissions.html

│ ├── js/

│ │ ├── auth.js

│ │ ├── courses.js

│ │ ├── dashboard.js

│ │ ├── quizzes.js

│ │ ├── submissions.js

│ ├── css/

│ ├── styles.css

│── package.json

# ****🔹 1️⃣ דרישות המערכת****

✅ **שלוש רמות משתמשים:**

* **Admin** – ניהול קורסים, חומרי לימוד, משתמשים וציונים.
* **Instructor** – יצירת קורסים, הוספת תכנים, יצירת חידונים ובדיקת הגשות.
* **Student** – הרשמה לקורסים, צפייה בשיעורים, מילוי חידונים והעלאת עבודות.

✅ **אימות משתמשים ואבטחה**:

* הרשאות לכל תפקיד.

✅ **ניהול קורסים וחומרי לימוד**:

* הוספת קורסים עם תכנים (וידאו, PDF, טקסט).
* מערכת הרשמה לקורסים.
* הצגת חומרים לפי סטטוס ההתקדמות של המשתמש.

✅ **מערכת חידונים והגשות**:

* יצירת חידונים לכל שיעור.
* תוצאות חידונים והגשת עבודות.

✅ **Dashboard לכל משתמש**:

* **Admin** – ניהול קורסים, סטטיסטיקות תלמידים.
* **Instructor** – צפייה ברשימת סטודנטים בקורסים.
* **Student** – רשימת קורסים פעילים והתקדמות לימודית.

✅ **מערכת תשלומים (אופציונלית)**:

* הרשמה לקורסים בחינם או בתשלום דרך Stripe.

✅ **API מתקדם עם MongoDB**:

* קורסים מנוהלים במסד נתונים עם סטטוס, מרצים ומשתמשים רשומים.
* נתונים מבוססי גרפים להצגת התקדמות סטודנטים.

# ****🔹 2️⃣ פונקציונליות Backend (Node.js + MongoDB)****

## ****📌 1. ניהול משתמשים (Auth)****

1. **רישום משתמשים (**POST /auth/register**)**
   * התחברות על בסיס email + password.
   * ברירת מחדל – student.
   * משתמש Admin יכול להגדיר משתמשים כמנהלים (Instructor).
2. **התחברות (**POST /auth/login**)**
   * אימות סיסמה מול מסד הנתונים.
   * החזרת JWT Token.
3. **הרשאות Middleware**
   * adminMiddleware.js – מאפשר רק ל- Admin לבצע שינויים בניהול מערכת.
   * instructorMiddleware.js – מאפשר רק למרצים להוסיף קורסים ולערוך אותם.

## ****📌 2. ניהול קורסים****

1. **יצירת קורס (**POST /courses**)**
   * רק Instructor או Admin יכולים ליצור קורסים.
   * title, description, instructorId, materials.
2. **הוספת תכנים לקורס (**POST /courses/:id/materials**)**
   * העלאת PDF / וידאו / טקסט לקורס.
3. **הצגת כל הקורסים (**GET /courses**)**
   * הצגת רשימת הקורסים לכל המשתמשים.
4. **רישום לקורס (**POST /courses/:id/enroll**)**
   * רק Student יכול להירשם לקורס.

## ****📌 3. מערכת חידונים והגשות****

1. **יצירת חידון (**POST /quizzes**)**
   * מרצה יכול ליצור חידון עם שאלות.
2. **הגשת חידון (**POST /quizzes/:id/submit**)**
   * Student שולח תשובות ונשמרות במסד.
3. **בדיקת ציונים (**GET /quizzes/:id/results**)**
   * Instructor יכול לבדוק הגשות.

# ****🔹 3️⃣ פונקציונליות Frontend (JavaScript, HTML, CSS)****

📌 **דפים עיקריים:**

1. index.html – עמוד ראשי עם רשימת קורסים.
2. register.html – רישום משתמשים.
3. login.html – התחברות.
4. dashboard.html – דשבורד אישי לפי סוג המשתמש.
5. courses.html – עמוד לכל הקורסים.
6. course-detail.html – עמוד לקורס ספציפי עם חומרים ולחצן הרשמה.
7. quiz.html – חידונים של הקורס.
8. submissions.html – הצגת הגשות ומילוי חידונים.

📌 **JS Scripts:**

* auth.js – ניהול התחברות ורישום.
* courses.js – טעינת קורסים מהשרת.
* dashboard.js – ניהול נתונים אישיים.
* quizzes.js – שליחת תשובות.
* submissions.js – צפייה בציונים.