# **טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

**הגשת פרויקט**

# **להגשה עד 13.7.25 בשעה 23:59** יש לכלול בתחילת מסמך הפרויקט את דף השער הבא: **טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

**הגשת פרויקט**

**שם פרויקט <Micro-learn>**

**קוד פרויקט: < B21>**

**מס קבוצה** : **<8>**

|  |  |
| --- | --- |
| **שם חבר.ת הצוות** | **תז** |
| אוסאמה | 325468254 |
| בדר | 322394214 |
| נואר | 213911548 |
| ודיע | 212682637 |
| הלאל | 212131353 |
| ספא | 325223436 |

**תקציר הפרויקט –**

**הפרויקט MicroLearn הוא פלטפורמת מיקרו-למידה אינטראקטיבית, שנועדה לעזור לסטודנטים להשלים חומר לימודי במהירות באמצעות סרטונים ממוקדים בני עד 6 דקות ותקצירים מבוססי AI. המערכת מחולקת לשני סוגי משתמשים עיקריים:**

* **משתמשים רשומים יכולים לחפש נושאים, לצפות בסרטונים קצרים, לקרוא תקציר AI, לבצע חידון קצר על החומר ולצבור ניקוד. כל החיפושים, הסרטונים והחידונים נשמרים בפרופיל המשתמש לצורכי מעקב והתקדמות, כולל הצגת תגיות, ציונים ותגמולים (badges).**
* **משתמשים לא רשומים יכולים להשתמש בעמוד ה-Try לביצוע חיפוש בלבד, ולקבל תקציר וסרטונים – אך ללא אפשרות לחידון וללא פרופיל אישי.**

**בנוסף, קיימת מערכת ניהול עבור מנהל (Admin), הכוללת מחיקת משתמשים, צפייה בגרפים וסטטיסטיקות (כגון חיפושים לפי קטגוריות והנושאים הנפוצים ביותר), וניהול כלל המשתמשים במערכת.**

**מימוש – טכנולוגיות מרכזיות בפרויקט MicroLearn**

* **Backend**  
  Next.js API Routes (Node.js Serverless Functions)  
  • שימוש ב-Fetch פנימי לשמירת היסטוריית חיפושים, צפיות וביצועי חידונים  
   Gemini API להפקת הסברים, תקצירים ושאלות  
   YouTube Data API לקבלת סרטונים קצרים על פי נושא החיפוש
* **Frontend**  
  Next.js (React-based Framework)  
  React Hooks (useState, useEffect, useContext)  
   React Context – לניהול התחברות המשתמשים (AuthContext)  
   Dynamic Routing עם Next Router  
   Chart.js – להצגת גרפים בעמוד הניהול
* **מסד נתונים (Database)**  
   MongoDB – מסד נתונים בענן לניהול נתוני המשתמשים, החיפושים, הצפיות והחידונים. mongoose – ספרייה לניהול הגישה ל־MongoDB מתוך Node.js בצורה נוחה ובטוחה.
* **עיצוב (Styling)**  
  Tailwind CSS – לעיצוב רספונסיבי, נגיש ומודרני  
   Lucide-react – סט אייקונים אינטראקטיבי  
   Chart.js – להצגת תרשימים אינטראקטיביים בלוח הניהול

**קישור לתיקיית Git ציבורי (הסופית):**

<https://github.com/osamamadi/MicroLearn>

**קישור לתיקיית Git שבה שמורות כל הגרסאות של הקוד (test):**

<https://github.com/HelalAli31/micro-learn>

**קישור לאתר:**

[**https://micro-learn-67et.vercel.app/**](https://micro-learn-67et.vercel.app/)

1. עליכם להמשיך את בניית האתר לפי האלמנטים המתקדמים שלמדתם
2. יש למנות מהנדס מערכת בכל צוות, אשר יהיה אחראי על הגדרת והקצאת המשימות בתרגיל זה.  
   נא לרשום את שם הסטודנט בתרגיל זה. על מהנדס המערכת לכתוב כיצד נעשתה חלוקת העבודה מול הצוות, מה היו המשימות של כל חבר צוות, האם היה ממשק בין חברי הצוות, והאם המשימות מולאו:

**מהנדס המערכת: ודיע פראן**  
ודיע נבחר לשמש כמהנדס המערכת בתרגיל זה בזכות ניסיונו, הבנתו הטכנית בתחום הפיתוח, ויכולותיו הארגוניות והבינאישיות. חלוקת המשימות בוצעה בצורה שוויונית ובהתאם לחוזקות של כל אחד מחברי הצוות, תוך התחשבות בהעדפות האישיות של כל משתתף.  
לאורך העבודה התקיימה תקשורת פתוחה ורציפה בין כל חברי הצוות, כולל פגישות תיאום, דיונים טכניים ומעקב אחר ההתקדמות. כל משימה הוקצתה בצורה ברורה ונמסרה לביצוע בזמן. כאשר התעוררו אתגרים או בעיות, הם טופלו במשותף תוך שיתוף פעולה מלא עד למציאת פתרון.  
בסיום התרגיל, כל אחד מחברי הצוות השלים את חלקו כנדרש, והפרויקט הושלם בהצלחה ובאופן מסודר.

|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** |
| אוסאמה | -הכנת תיק משתמש | הכל |
| ודיע | שילוב כל החלקים של קובץ | הכל |
| בדר | הכנת וידאו+עדכונים useCase | הכל |
| נואר | -הכנת תיק משתמש | הכל |
| הלאל | הכנת דרישות+עדכון ארכיטקטורה | הכל |
| ספא | -הכנת תיק מתכנת | הכל |

2. **הציגו רשימת דרישות:**

**יש להתבסס על המבנה של asana (**[**https://asana.com/resources/software-requirement-document-template**](https://asana.com/resources/software-requirement-document-template)

* **דרישות פונקציונליות**
* **דרישות לא פונקציונליות (בנפרד, יש לסווג דרישות לא פונקציונליות לפי wikipedia NFR).**
* **דרישות ממשק חיצוניות.**

**Functional Requirements**

 The system allows users to search for academic topics using keywords.

 The system fetches short, high-quality videos (under 5 minutes) from YouTube based on the search query.

 The system displays the top 6 most relevant videos for each query.

 The system generate a short explanation paragraph based on the user’s search topic.

 The system allows users to read both the AI-generated summary and watch the related videos on the same page.

 The system allows users to search without needing to log in or create an account.

 The system allows the manager to view analytics about popular searches and most-viewed topics.

 The system allows users to take a quiz related to their search topic.

 The system allows the submission of a quiz only if all of the questions are answered.

 The system displays the score immediately after the quiz is submitted.

 The system saves the user’s search history, video viewing history, and quiz history in their profile.

 The system tracks user activity and quiz performance to calculate their learning progress.

 The system awards digital badges to users based on their search frequency, quiz scores, and learning consistency.

 The system updates the user’s badge automatically as their activity or performance improves.

 The system allows the admin to remove users from the platform.

 The system lets users edit their profile details.

**Non-Functional Requirements**

**Usability**

* The interface is simple, intuitive, and easy to navigate for students of all levels.
* Users can complete actions (search, learn, quiz, view history) in 1–2 clicks.
* Visual hierarchy ensures focus on the learning material with minimal distraction.

**Performance**

* Quizzes are loaded instantly and scored within 1 second after submission.
* All profile and badge updates are processed **within 1-2 seconds** after the triggering action.

**Interoperability**

* **The platform is compatible with major operating systems and devices, including desktops, tablets, and smartphones.**
* **The system supports all major browsers, including Chrome, Firefox, Safari, and Edge.**

**Reliability**

* In the case of high traffic, load balancing mechanisms ensure continued performance.
* Quizzes and search logs are autosaved to prevent data loss during usage.

**Availability**

* The platform maintains 99.9% uptime using cloud deployment via Vercel.
* Failover and rollback mechanisms are implemented to revert to the last stable version in case of errors or deployment issues.

**Maintainability**

* The system uses modular architecture and reusable components for easy maintenance.
* Codebase is documented and structured to allow fast debugging and scalable updates.
* Clear separation of concerns between frontend (UI), backend (API logic), and external services (YouTube/Gemini).

**Scalability**

The quiz system and ranking engine support horizontal scaling for future growth.

**Security**

* All API keys (e.g., Gemini, YouTube) are stored in .env.local and not exposed to the frontend.
* User data (e.g., history, performance) is protected with role-based access (admin vs user).

**Accessibility**

* The interface supports keyboard navigation and screen-reader compatibility.
* Font sizes and contrast ratios are accessible to users with visual limitations.

**External interface requirements**

**User Interfaces**

* **Home Page**: Public landing page that explains what Micro-Learn is and invites users to try or register.
* **Try Page**: For visitors without login; allows search and displays Gemini summary + videos, but no quizzes or profile.
* **Register Page**: Enables account creation using Google login.
* **Main App (for logged-in users)**:
  + Search bar with AI summary + 6 short YouTube videos.
  + “Take Quiz” button for each search.
  + Profile page with history (search, quiz, video) and earned badges.
* **Admin Page**: Allows admins to view users and remove accounts.

**Hardware Interfaces**

* **Runs on any device with a browser (desktop, mobile, tablet).**
* **Requires internet connection; no special hardware needed.**

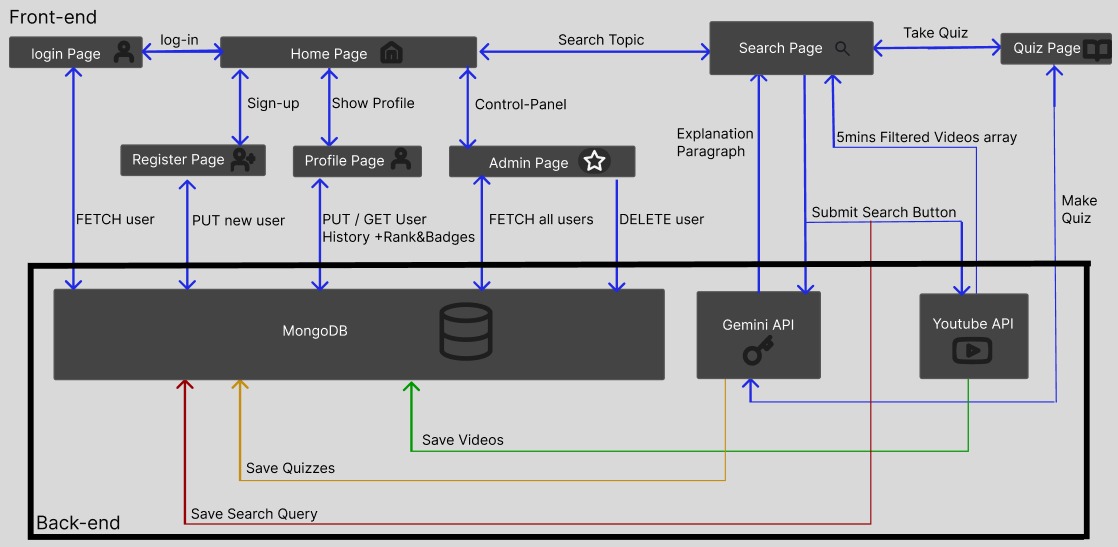
**Software Interfaces**

* **Gemini API: For AI-generated summaries.**
* **YouTube API: For fetching short videos.**
* **Next.js (App Router): Web framework for pages and API routes.**
* **Tailwind CSS: For responsive UI.**
* **Vercel: For cloud hosting.**

**Communication Interfaces**

* **Uses HTTPS requests for all API communication.**
* **Future plan: Add feedback form and email notifications.**

3. הציגו ארכיטקטורה מעודכנת של האתר (תרשים הכולל את האלמנטים המרכזיים).



4. הציגו דיאגרמת use case המתארת את השימוש באתר.

תמונה שמכילה טקסט, תרשים, צילום מסך, קו

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

5. יש להציג מבנה סופי של האתר שלכם:

**נדרשת בכל פרויקט פריסה מלאה (deploment) של הפרויקט! ב - vercel.  
לא יתקבלו הגשות של קבצים או אתר ב -localhost .**

**טכנולוגיות**:    להלן המרכיבים הטכנולוגיים המומלצים לשימוש בפרויקט:

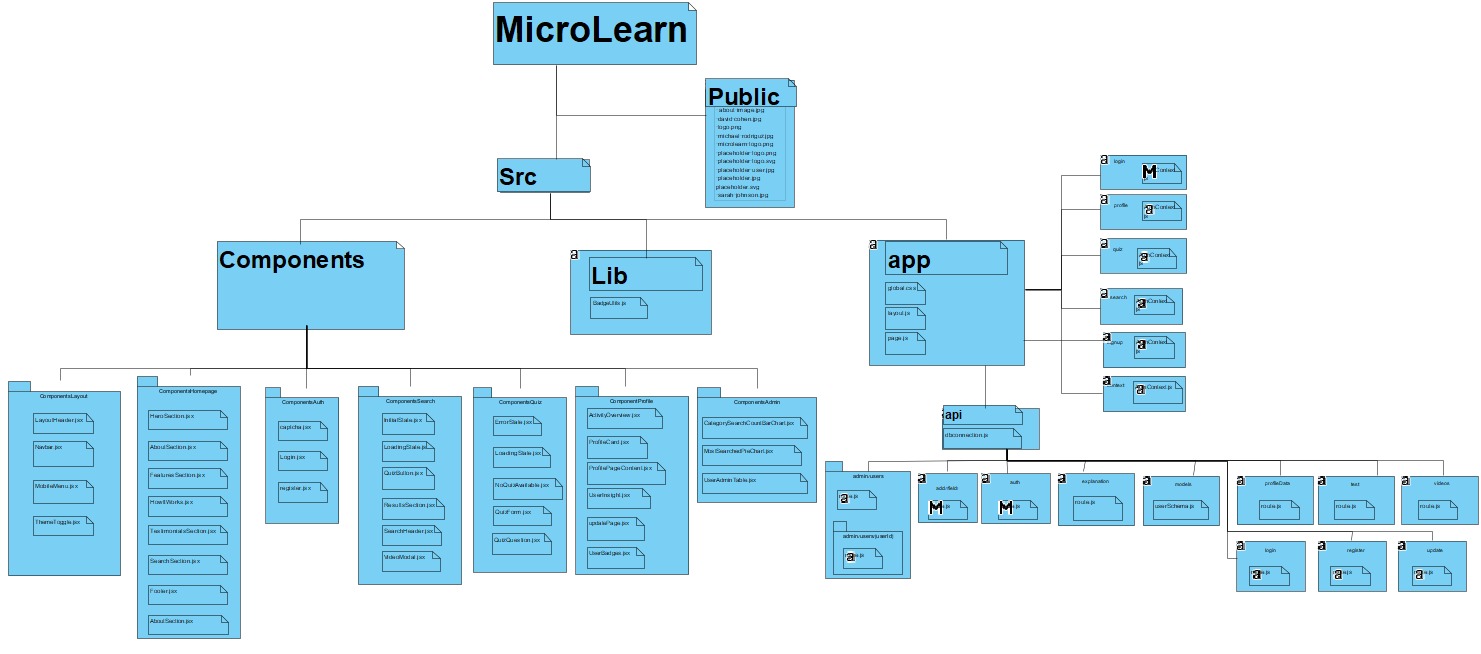
·           - front-end: React/Preact/Next with Tailwind

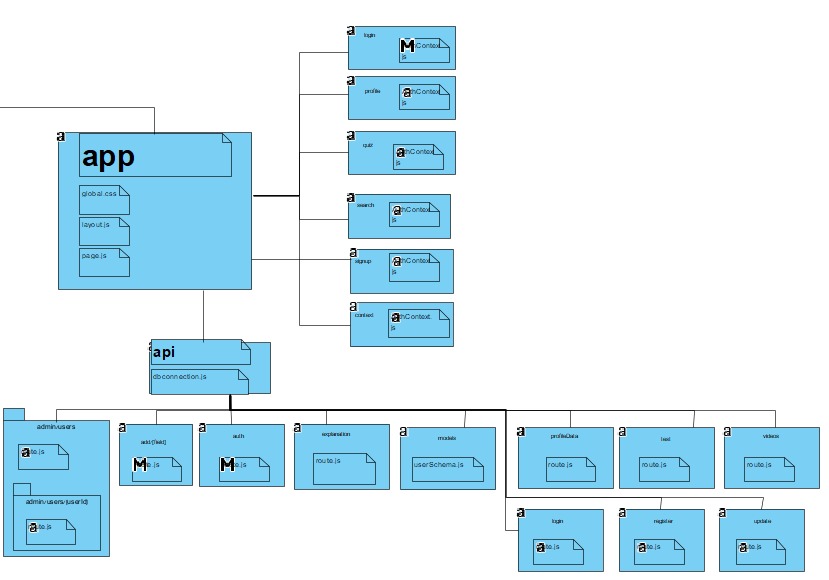
    - back-end:

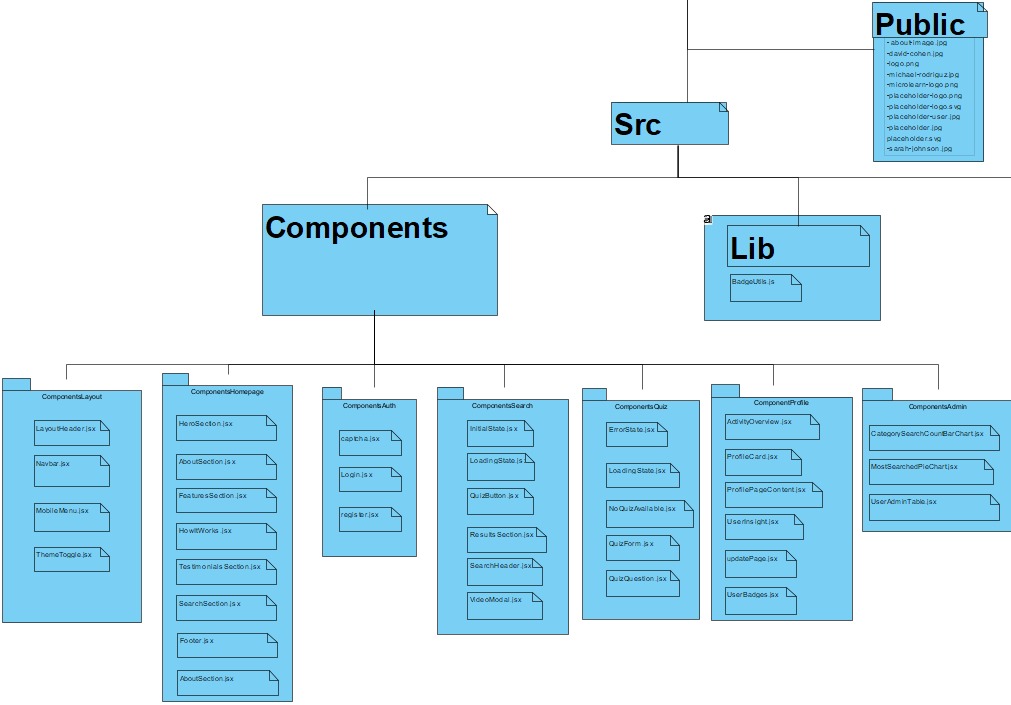
    - option 1 - remote services/APIs

    - option 2 - node js/express deployed on remote web server

**א. האתר ימומש ב -React/Preact/Next, וכן שימוש ב Tailwind - נא להציג דיאגרמה המתארת את התיקיות והקבצים השונים. יש לפרט את הקומפוננטות השונות.**

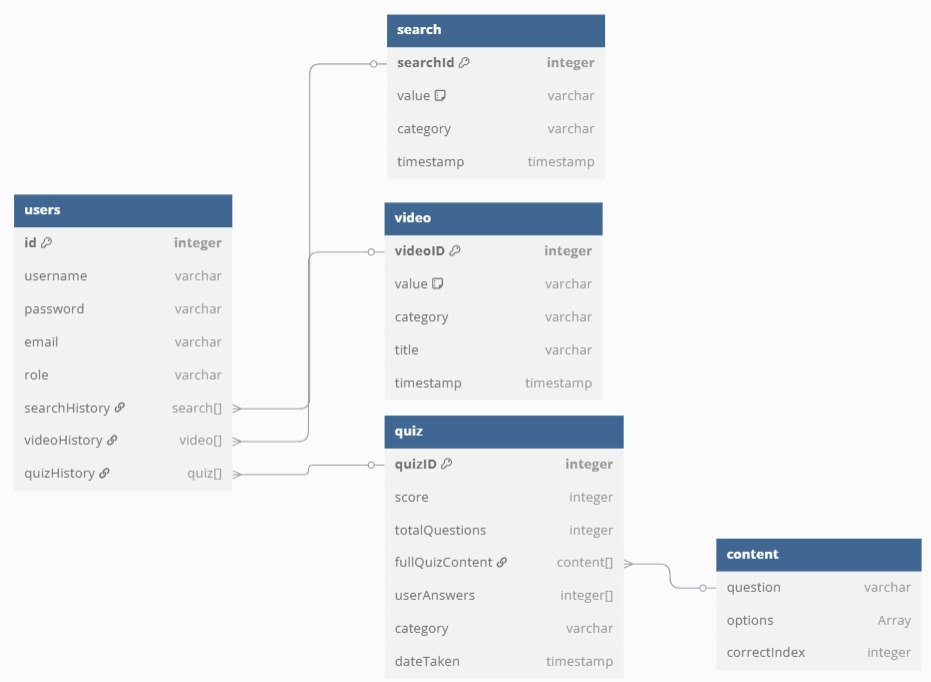
****

צד ימין:

צד שמאל

ב. יש לפרט את פריטי המידע - יש להשתמש במידע אמיתי ורלוונטי לפרויקט שלכם (בשליפה ממסד נתונים חיצוני או מ - API). יש להראות דיאגרמת מבנה DB.**.**

**המערכת משתמשת במסד נתונים הכולל טבלאות למשתמשים, היסטוריית חיפושים, סרטונים, מבחנים ותוכן שאלות. כל משתמש שומר את הפעילות האישית שלו, כולל חיפושי תוכן, סרטונים שצפה בהם, וחידונים שביצע. הנתונים נשמרים בצורה מקושרת, כך שניתן לעקוב אחר ההתקדמות האישית של כל משתמש, להציג תובנות מותאמות אישית ולהעניק תגמולים כמו תגים לפי ביצועים.**

****

.**6 יש להגיש תיק למתכנת:**

**א. יש לתעד ולהסביר את כל הפונקציות המרכזיות בקוד שלכם.**

**תיק מתכנת:**

**1.יש לתעד ולהסביר את כל הפונקציות המרכזיות בקוד שלכם**

**CategorySearchCountBarChart()**

תיאור: רכיב React זה אחראי על הצגת תרשים עמודות הממחיש את סך מספר החיפושים עבור קטגוריות שונות. הוא משתמש בספריית react-chartjs-2 ליצירת גרפים דינמיים, ומגדיר את צבעי העמודות, תוויות הציר וכותרות התרשים. הרכיב מותאם לרספונסיביות וכולל עיצוב מותאם אישית עבור הטיפים (tooltips).

קלט:

* labels (מערך מחרוזות): שמות הקטגוריות שיוצגו על ציר ה-X.
* data (מערך מספרים): ספירות החיפושים התואמות לכל קטגוריה.

פלט:

* תרשים עמודות רספונסיבי, או הודעת טקסט אם אין נתונים זמינים.

**MostSearchedPieChart()**

**תיאור:** רכיב React זה מציג **תרשים עוגה** שמראה את מונחי החיפוש הנפוצים ביותר במערכת. הוא מספק ייצוג חזותי קל להבנה של נתוני חיפוש פופולריים. הרכיב משתמש ב־react-chartjs-2 וכולל הגדרות לצבעי המקטעים, מיקום המקרא (legend) ואופן הצגת המידע בטיפים (tooltips).

**קלט:**

* labels (מערך מחרוזות): שמות מונחי החיפוש המובילים.
* data (מערך מספרים): ספירות החיפושים עבור כל מונח.

**פלט:**

* תרשים עוגה רספונסיבי, או הודעת טקסט אם אין נתונים זמינים.

**UserAdminTable()**

תיאור: רכיב React זה הוא טבלת ניהול משתמשים מקיפה המיועדת למנהלי מערכת. הוא מציג פרטים חשובים על משתמשים כמו שם, אימייל, תפקיד ונתוני חיפוש/הישגים. הטבלה מאפשרת מיון (לפי הישגים) וכוללת פונקציונליות למחיקת משתמשים, עם מודל אישור והודעות התראה על פעולות.

קלט:

* users (מערך אובייקטי משתמשים): נתוני המשתמשים שיש להציג בטבלה.
* onUserDeleted (פונקציה): פונקציית קריאה חוזרת המופעלת לאחר מחיקת משתמש מוצלחת.

פלט:

* טבלת HTML מעוצבת עם נתוני המשתמשים, מודל אישור מחיקה, והודעות טוסט (toast notifications).

**SearchHeader()**

**תיאור:** רכיב זה מציג את כותרת החיפוש, כולל סמל חיפוש וכותרת בולטת. הוא כולל שדה קלט טקסט לחיפוש וטופס לשליחת שאילתות. הרכיב מעוצב עם סגנונות מודרניים וצבעים בולטים.

**קלט:**

* query (מחרוזת): ערך החיפוש הנוכחי.
* setQuery (פונקציה): פונקציה לעדכון ערך החיפוש.
* handleSubmit (פונקציה): פונקציה לשליחת הטופס.
* username (מחרוזת): שם המשתמש.

פלט: קטע בממשק המשתמש (UI) המכיל כותרת חיפוש, שדה קלט לחיפוש, וכפתור שליחה.

**()VideoModal**

תיאור: רכיב זה מציג מודל קופץ (modal) להצגת סרטון. הוא מציג את הסרטון בתוך iframe, יחד עם פרטים כמו כותרת, משך זמן ומספר צפיות. הרכיב מאפשר סגירה קלה של המודל.

קלט:

* selectedVideo (אובייקט): פרטי הסרטון שנבחר להצגה.
* closeVideoModal (פונקציה): פונקציה לסגירת המודל.

**פלט:** מודל קופץ בממשק המשתמש (UI) המכיל את נגן הווידאו ופרטי הסרטון.

**ResultsSection**

**תיאור:** רכיב זה מציג את תוצאות החיפוש, כולל הסבר טקסטואלי ורשימת סרטונים רלוונטיים. הוא מאפשר להציג/להסתיר את ההסבר המלא, ומציג תמונות מקדימות של הסרטונים עם פרטים כמו משך זמן ומקור. במקרה שלא נמצאו סרטונים, מוצגת הודעה מתאימה.

**קלט:**

* explanation (אובייקט): הסבר טקסטואלי לתוצאות החיפוש.
* summary (מחרוזת): תקציר של ההסבר.
* videos (מערך אובייקטים): רשימת סרטונים רלוונטיים.
* פונקציות ומשתנים נוספים לטיפול באירועים ובמצבים שונים.

פלט: קטע בממשק המשתמש (UI) המכיל את הסבר החיפוש, רשימת סרטונים, או הודעה המציינת שלא נמצאו סרטונים.

**QuizQuestion()**

תיאור: רכיב זה מציג שאלה בודדת בחידון יחד עם אפשרויות התשובה שלה. הוא מאפשר למשתמש לבחור תשובה אחת מבין האפשרויות. לאחר הגשת החידון, הרכיב יכול להציג את התשובה הנכונה. הוא גם מסמן תשובות שגויות ומציג את התשובה שבחר המשתמש.

קלט:

* quizItem (אובייקט): נתוני השאלה הספציפית.
* index (מספר): האינדקס של השאלה במערך החידון.
* selectedAnswer (מספר): אינדקס התשובה שנבחרה על ידי המשתמש.
* handleOptionChange (פונקציה): לטיפול בבחירת תשובה.
* showResults (בוליאני): האם להציג את התוצאות.

פלט: קטע בממשק המשתמש (UI) המציג שאלה בודדת בחידון.

**QuizForm()**

**תיאור:** רכיב זה מהווה את הטופס הראשי של השאלון. הוא מציג את כל שאלות החידון באמצעות רכיב QuizQuestion. הטופס מטפל בשינויים בתשובות המשתמש ומציג הודעות ולידציה. הוא מאפשר שליחת החידון או הצגת ציון ואפשרות לנסות שוב.

**קלט:**

* **quiz** (מערך אובייקטים): נתוני השאלות והאפשרויות של החידון.
* **selectedAnswers** (אובייקט/מערך): התשובות שבחר המשתמש.
* **handleOptionChange** (פונקציה): לטיפול בשינוי בחירת תשובה.
* **handleSubmitQuiz** (פונקציה): לטיפול בשליחת הטופס.
* **showResults** (בוליאני): האם להציג את תוצאות החידון.
* **validationMessage** (מחרוזת): הודעה המוצגת בוולידציה שנכשלה.
* **calculateScore** (פונקציה): פונקציה שמחשבת ומחזירה את הציון.

**פלט:** טופס אינטראקטיבי בממשק המשתמש (UI) המציג את החידון.

**ProfileCard()**

**תיאור**: רכיב זה מציג את כרטיס הפרופיל האישי של המשתמש. הוא מציג פרטים בסיסיים של המשתמש כמו שם משתמש וכתובת אימייל. הרכיב כולל כפתור "Edit Personal Information" המאפשר לערוך את הפרטים הללו. לחיצה על כפתור העריכה פותחת מודל עדכון נפרד.

**קלט:**

* **user** (אובייקט): אובייקט המשתמש הנוכחי עם פרטיו האישיים.
* **onUserUpdate** (פונקציה): פונקציית קריאה חוזרת שתופעל לאחר עדכון פרטי המשתמש בהצלחה.

**פלט:** כרטיס ממשק משתמש (UI) המציג את פרטי הפרופיל. פותח מודל UpdatePage לעריכת מידע.

**UpdatePage**

**תיאור:** רכיב זה מציג **מודל קופץ לעדכון פרטי הפרופיל** של המשתמש. הוא מאפשר למשתמש לעדכן את שם המשתמש, אימייל וסיסמה. המודל דורש הזנת סיסמה נוכחית לצורך אימות לפני שמירת שינויים. הוא מטפל בשליחת הנתונים לשרת ומציג הודעות שגיאה או הצלחה.

**קלט:**

* **onClose** (פונקציה): פונקציית קריאה חוזרת לסגירת המודל.
* **onUserUpdated** (פונקציה): פונקציית קריאה חוזרת המופעלת עם נתוני המשתמש המעודכנים.

**פלט:** מודל קופץ בממשק המשתמש (UI) המכיל טופס לעדכון פרטי פרופיל.

**UpdatePage()**

**תיאור:** רכיב זה מציג **מודל קופץ לעדכון פרטי הפרופיל** של המשתמש. הוא מאפשר למשתמש לעדכן את שם המשתמש, אימייל וסיסמה. המודל דורש הזנת סיסמה נוכחית לצורך אימות לפני שמירת שינויים. הוא מטפל בשליחת הנתונים לשרת ומציג הודעות שגיאה או הצלחה.

**קלט:**

* **onClose** (פונקציה): פונקציית קריאה חוזרת לסגירת המודל.
* **onUserUpdated** (פונקציה): פונקציית קריאה חוזרת המופעלת עם נתוני המשתמש המעודכנים.

**פלט:** מודל קופץ בממשק המשתמש (UI) המכיל טופס לעדכון פרטי פרופיל.

**UserBadges**

**תיאור**: רכיב זה מציג את התגים וההישגים של המשתמש. הוא משתמש בפונקציית getUserBadges חיצונית כדי לחשב אילו תגים נצברו. הרכיב מציג את התגים שנצברו ואת ההתקדמות לקראת התג הבא באמצעות סרגל התקדמות. הוא מאפשר הצגה או הסתרה של טבלה מפורטת עם כל יעדי התגים והתקדמות עבורם.

**קלט:**

user(אובייקט): אובייקט המשתמש, שממנו נלקח user.quizHistory לחישוב התגים.

**פלט**: קטע ממשק משתמש (UI) המציג את התגים שנצברו, סרגל התקדמות לתג הבא, וטבלת תגים מפורטת.

ב)

**2. את כל ה -API שהשתמשתם בהם**

**YouTube Data API v3 -**  
**תיאור:** חיפוש סרטונים ושאיבת סטטיסטיקות מתוך YouTube Data API v3.

**Google Generative Language API (Gemini) -**  
**תיאור:** יצירת הסברים, סיכומים וחידונים דינמיים באמצעות מודל השפה של גוגל.

**MongoDB Atlas** -  
**תיאור:** מסד נתונים בענן לשמירת כל הנתונים (משתמשים, פרופילים, היסטוריות וכו').

**3.קישור ל-DB )מסד נתונים(**

<mongodb+srv://viwer:justviwer@cluster0.kghsp.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0>

**4. סביבות מיוחדות שהתקנו:**

* **Vercel**  
  סביבת פריסה ודיחוס אוטומטי (CI/CD) עבור אפליקציית Next.js בענן.

**כתובת -** [**https://micro-learn-67et.vercel.app/**](https://micro-learn-67et.vercel.app/)

* **Node.js (v18+) עם npm**

סביבת ריצה ל־JavaScript/TypeScript ושימוש ב־CLI להתקנת התלויות והרצת הסקריפטים.

* **Next.js (App Router)**

מסגרת React עם תמיכה ב־SSR/SSG, ניהול דפים תחת src/app ויצירת API פנימי.

* **MongoDB Atlas (עם Mongoose)**

מסד נתונים בענן ו־ORM שמנהל את החיבור (MONGODB\_URI) והסקימות בצד השרת.

**5.קטעי קוד מיוחדים שלקחנו ממקורות שונים:**

* **Google Gemini Content Generator (src/app/api/explanation/route.js)** **עוטף Google Generative Language API**

**מקור:**   
 Google Generative Language API docs (gemini-1.5-flash:generateContent)

* **YouTube Data API Wrapper (src/app/api/videos/route.js) עוטף YouTube v3**

**מקור:**YouTube Data API v3 docs (Search & Videos endpoints)

* **Mongoose User Schema (src/app/api/models/userSchema.js) – הגדרת סכימת משתמש  
  מקור:**

מדריך “Defining Schemas” בדוקומנטציה של Mongoose.

* **AuthContext (src/app/context/AuthContext.js)** **ניהול אימות משתמש באפליקציה**

**מקור:**

מבוסס על React Context API בדוקומנטציה הרשמית, עם התאמות לNext.js.

**.6 פרומפטים שהשתמשתם בהם בכלי AI**

השתמשנו **ב ChatGPT** כדי לשפר את הקוד שלנו מבחינת עיצוב , פונקציונאליות , לפתור כל מיני באגים..

**"**תוסיף לוגיקה בשרת שתבדוק אם הערך שנשלח ל־videoHistory כולל title, value, timestamp ו־category – ואם לא, תחזיר שגיאה**"**

**"**תתקן לי את השגיאה שאומרת ש־params.field צריך להיות await – אבל בלי באמת לעשות await, כי זה לא נכון**"**

"תעשה לי טיפול כללי בשגיאות ב־catch כך שיציג גם את שם השדה מתוך context"

**7. שמות משתמש וססמאות שהגדרתם לכל סוג משתמש, לצורך חיבור לאתר**

**ADMIN :**

**username**: osama

**password**: o

**USER:**

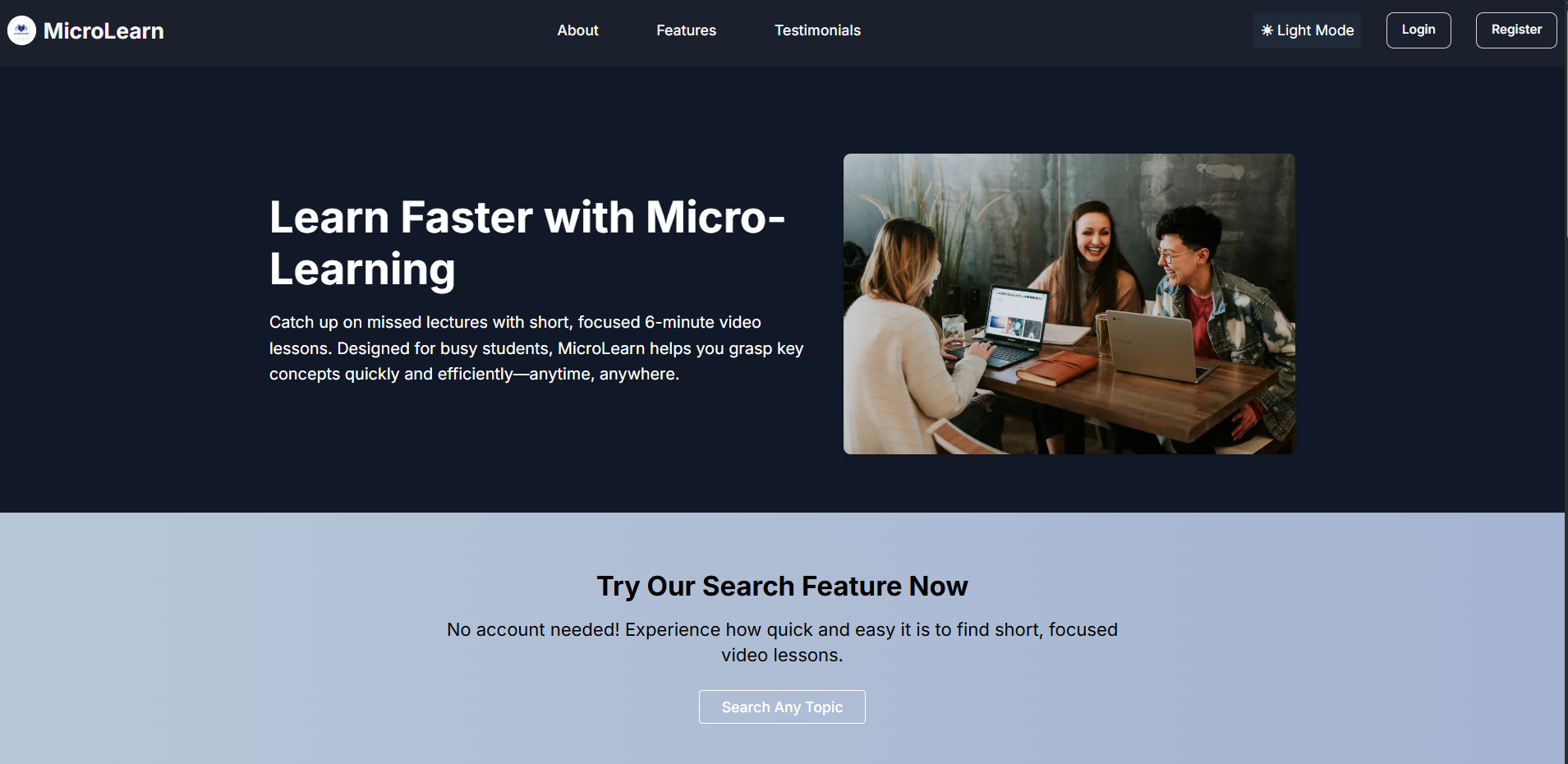
**username:** bader

**pass:** b

**7. יש להגיש תיק משתמש הכולל צילומי מסך והסברים כיצד להשתמש באתר שלכם. נא לכלול סרטון של כדקה-דקה וחצי , שיווקי, המציג את האתר**.

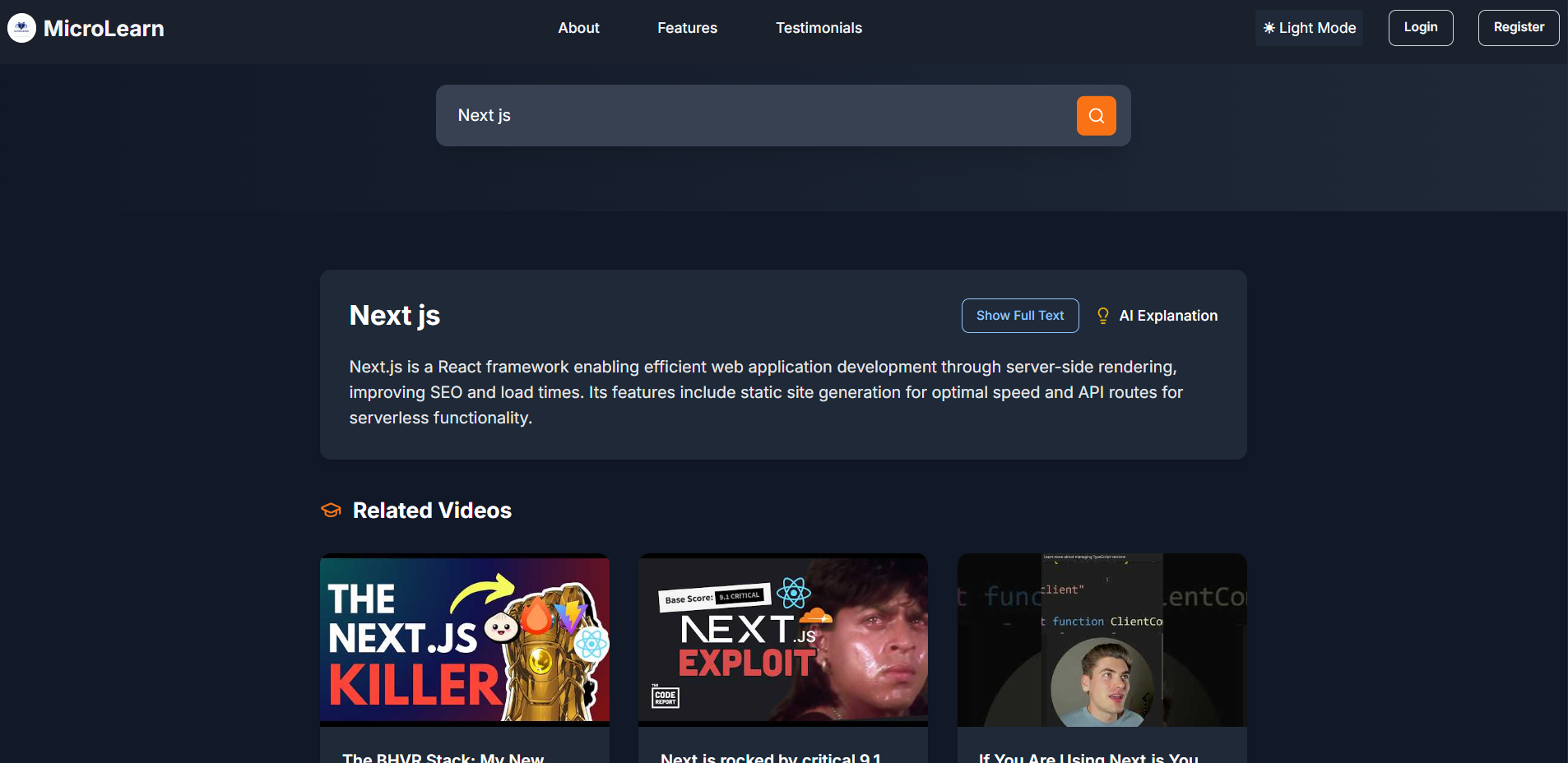
מסך הבית (Homepage):

בעת הכניסה לאתר, המשתמש מגיע לדף הבית של Micro-Learn, שבו מוצג הסבר קצר על מטרת הפלטפורמה ויתרונותיה. העמוד מעוצב בצורה ברורה ונעימה, עם כותרת המזמינה את המשתמש ללמוד בצורה מהירה ויעילה. בעמוד מופיעים מקטעים כמו "About MicroLearn", "Why MicroLearn Works" ו-"How It Works", המסבירים על השיטה, היתרונות, ושלבי השימוש בפלטפורמה. בנוסף, מופיעים המלצות של סטודנטים אמיתיים, המחזקים את הערך של המערכת. כפתור קריאה לפעולה מאפשר למשתמשים להתנסות במערכת דרך עמוד ה-**Try**, או להירשם דרך עמוד ההרשמה. בעיצוב מופיעים גם תמונות צבעוניות, התאמה למצבים כהה ובהיר, ורספונסיביות מלאה לכל סוגי המכשירים.



**מסך ההתנסות (Try Page)**

עמוד ההתנסות של Micro-Learn מיועד למשתמשים שאינם מחוברים לחשבון. בדף זה ניתן לבצע חיפוש על כל נושא אקדמי, ולקבל תוצאה מיידית הכוללת סיכום קצר שנוצר בעזרת בינה מלאכותית (Gemini AI) וכן עד שישה סרטונים קצרים מ-YouTube הממוקדים באותו נושא. המשתמשים יכולים לצפות בסרטונים ולקרוא את ההסבר, אך אינם יכולים לגשת לחידונים או לשמור את ההתקדמות שלהם. מטרת העמוד היא לאפשר לכל מבקר לנסות את המערכת בצורה מהירה ולחוות את יעילותה, ובסיום החיפוש מופיעה קריאה לפעולה המעודדת הרשמה לחשבון כדי ליהנות מהיכולות המלאות.

תמונה שמכילה טקסט, צילום מסך, תכונות מולטימדיה, תוכנה

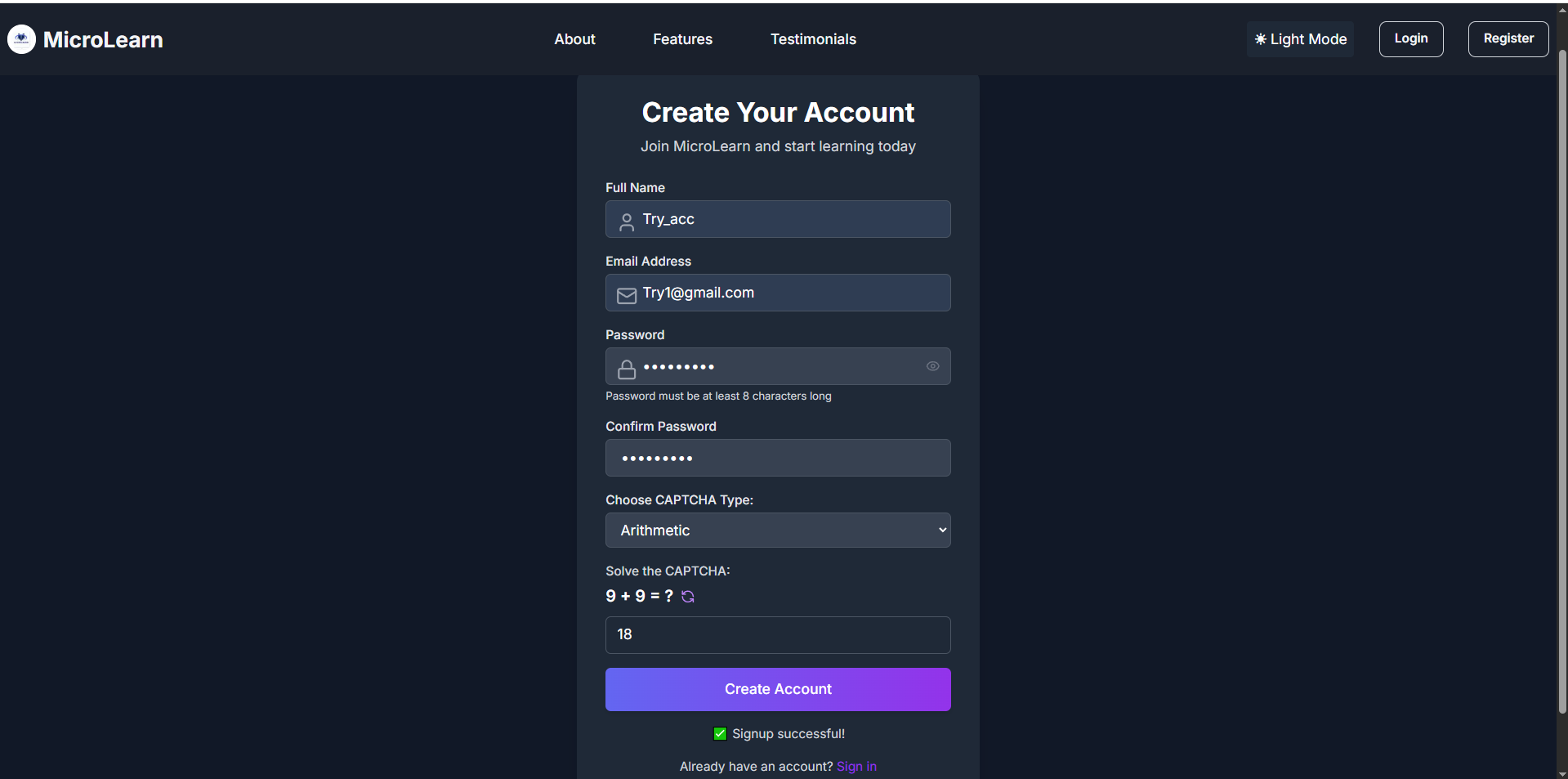
תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

תמונה שמכילה טקסט, פני אדם, צילום מסך, אדם

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

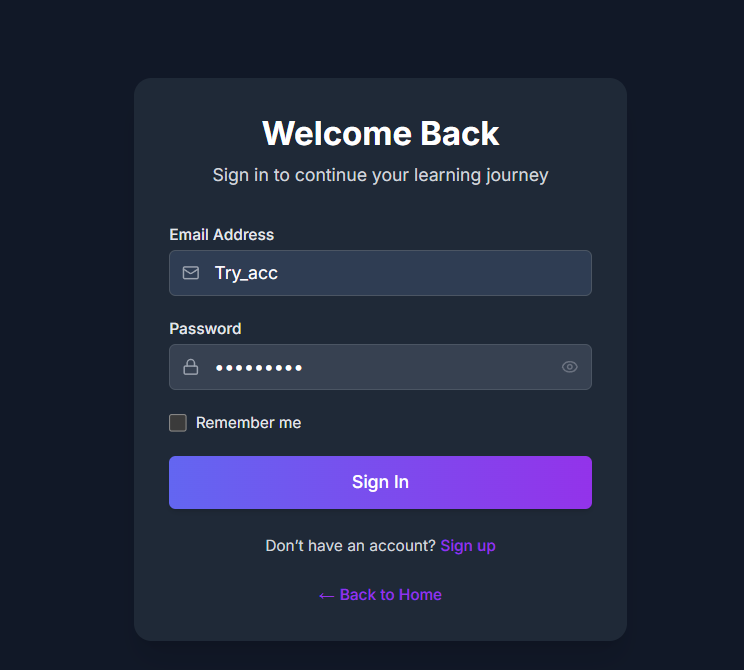
**מסך ההרשמה (Register Page)**

עמוד ההרשמה של Micro-Learn מאפשר למשתמשים חדשים ליצור חשבון בקלות ובצורה מאובטחת. בעמוד זה המשתמש מתבקש להזין שם משתמש, כתובת דוא"ל, סיסמה, ואימות סיסמה. בנוסף, לפני השלמת ההרשמה, יש לעבור אימות CAPTCHA – המשתמש יכול לבחור בין חישוב חשבוני פשוט או בחירת תמונות לפי נושא (כגון כלבים, מכוניות וכו'). לאחר אימות תקין, ניתן ללחוץ על כפתור "Create Account" וליצור חשבון חדש במערכת. במידה ויש שגיאה, מופיעה הודעה מתאימה. אם ההרשמה הצליחה, תופיע הודעת הצלחה, ובתחתית המסך מוצע קישור למשתמשים שכבר רשומים כדי לעבור למסך הכניסה. המסך מעוצב בצורה מודרנית, תומך במצבים כהים ובהירים, ומותאם לניידים ולמחשבים.



**מסך הכניסה (Login Page)**

מסך הכניסה מאפשר למשתמשים להתחבר באמצעות כתובת דוא"ל וסיסמה בצורה מאובטחת.  
לאחר ההתחברות, המשתמש מועבר לעמוד הפרופיל או לעמוד הניהול בהתאם לתפקידו.



**Admin panel**

**מסך הניהול מאפשר למנהל לצפות בכל המשתמשים במערכת, לראות את כמות החיפושים שלהם, ההישגים והקטגוריה הנפוצה שהם חיפשו.** בנוסף, המנהל יכול למחוק משתמשים בלחיצת כפתור ולבחון גרפים דינמיים שמציגים את הקטגוריות המבוקשות ביותר ואת התפלגות החיפושים לפי נושאים. המסך כולל גם התראות לפעולות ניהול וממשק גרפי נוח וידידותי.

תמונה שמכילה טקסט, צילום מסך, תוכנה, תכונות מולטימדיה

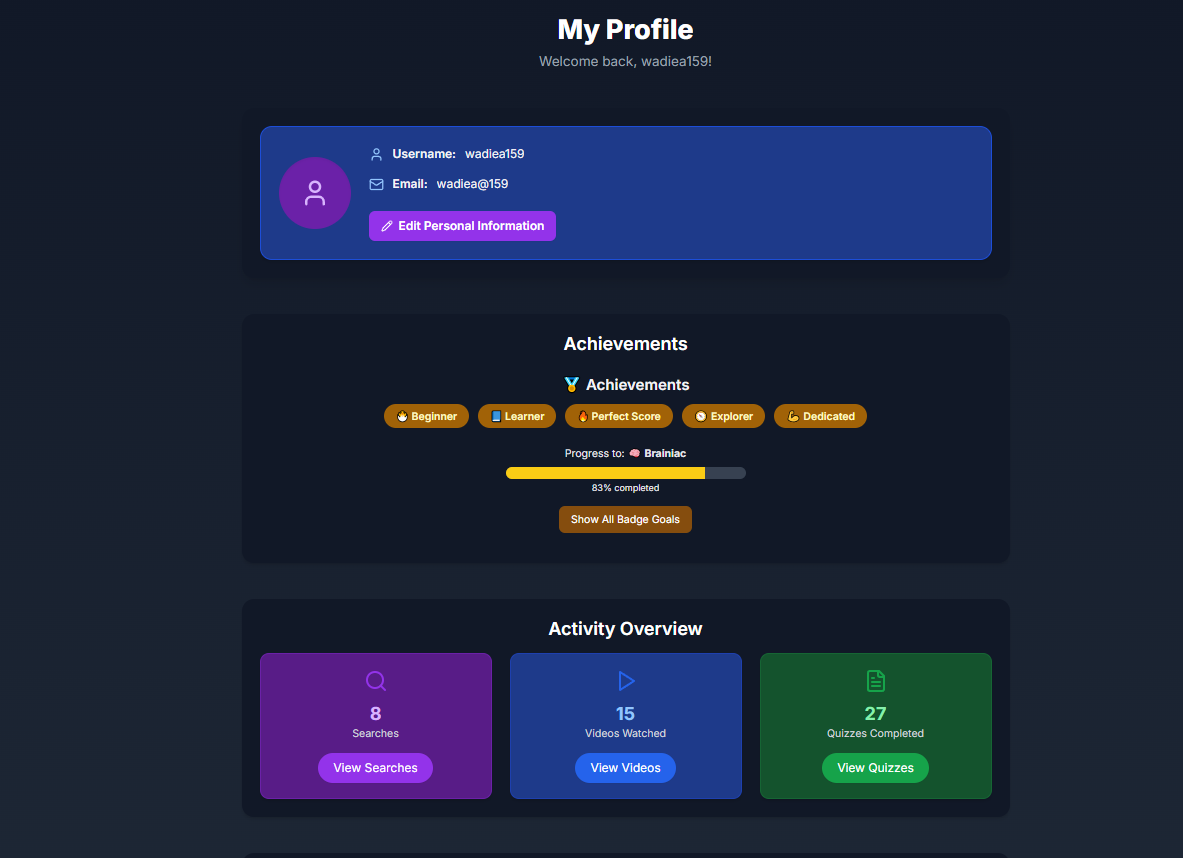
תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.



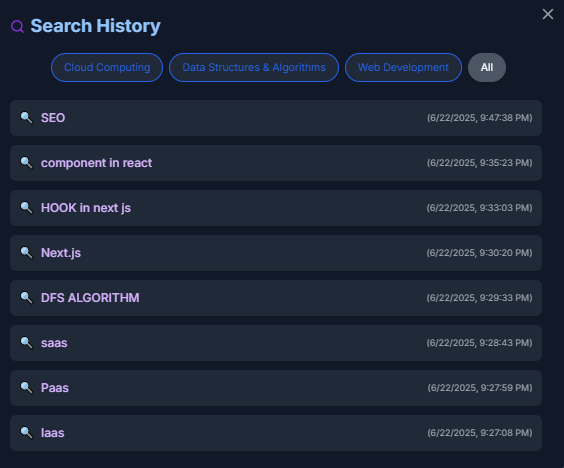


**Profile page**

עמוד הפרופיל של Micro-Learn מציג למשתמש מידע אישי וסטטיסטיקות על פעילות הלמידה שלו במערכת. בעמוד זה המשתמש יכול לצפות בפרטים כמו שם המשתמש, כתובת הדוא"ל וההישגים (תגים) שצבר במהלך השימוש. כמו כן, מוצגים סיכומים וסטטיסטיקות על ההתקדמות בלמידה, כגון מספר החיפושים, צפייה בסרטונים והשלמת מבחנים.  
המשתמש יכול לעדכן את פרטיו האישיים בלחיצה על כפתור העריכה, אשר פותח חלון לעריכת שם משתמש, דוא"ל וסיסמה. לפני שמירת השינויים יש לאשר את הסיסמה הנוכחית לשם אבטחה. במידה והשינויים נשמרים בהצלחה, פרטי המשתמש מתעדכנים מיידית ומוצגת הודעת הצלחה. במקרה של שגיאה, תופיע הודעה מתאימה.  
עמוד הפרופיל מעוצב כך שיתמוך גם במצב כהה ויאפשר למשתמש לעקוב אחר ההתקדמות וההישגים שלו בצורה ברורה ונוחה.



תמונה שמכילה טקסט, צילום מסך, גופן

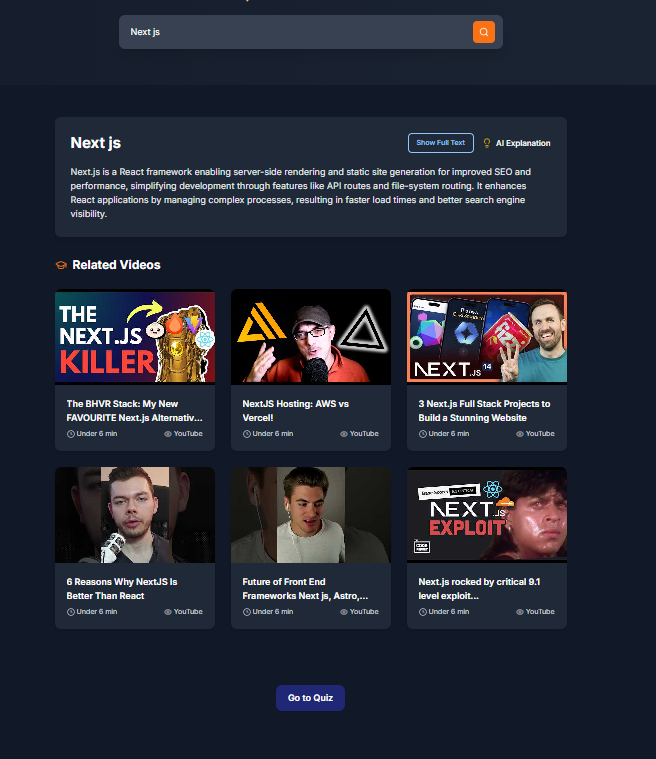
תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

תמונה שמכילה טקסט, צילום מסך, תוכנה

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**Search page(start learning)**:

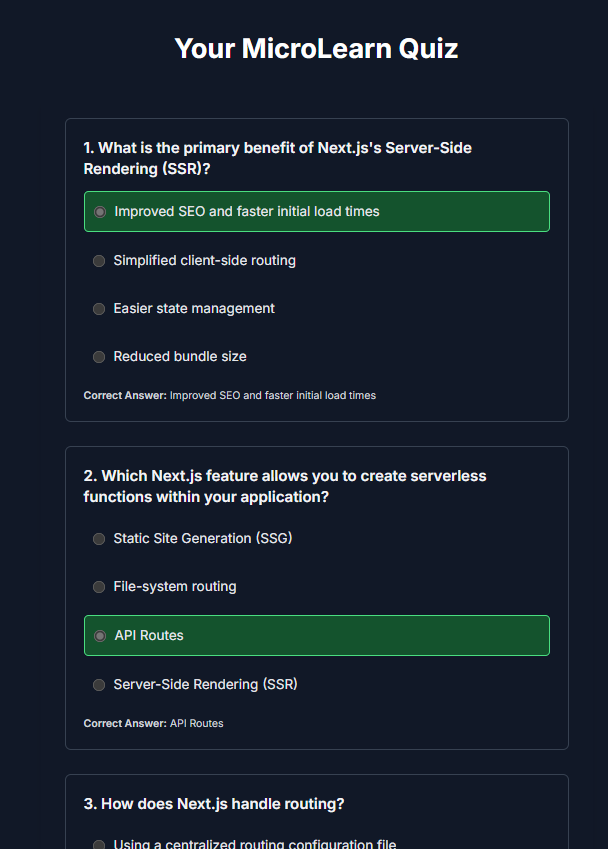
אותו הסבר כמו try page בנוסף יש את כפתור "go to quiz".



**:Quiz page**

**מסך המבחן (Quiz Page)  
עמוד המבחן של Micro-Learn מאפשר למשתמשים לענות על שאלות בצורה פשוטה ונוחה. כל שאלה מוצגת עם מספר אפשרויות בחירה, והמשתמש יכול לבחור את התשובה הנכונה על ידי סימון אחת מהאפשרויות. ניתן לשלוח את המבחן בלחיצה על כפתור "שלח מבחן".**

**לאחר השליחה, יוצגו תוצאות המבחן עם ניקוד מדויק, והמשתמש יוכל לראות אילו תשובות היו נכונות ואילו טעונות שיפור. במידה וירצה, הוא יכול ללחוץ על כפתור "נסה שוב" כדי לעבור את המבחן מחדש ולשפר את הציון שלו.**



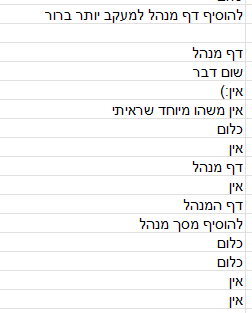
תמונה שמכילה טקסט, צילום מסך, גופן, תוכנה

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**וידאו**

[https://drive.google.com/file/d/1GYhuAKFqZcYBUT95J-0g8MwlTU38Tcn3/view](%20https:/drive.google.com/file/d/1GYhuAKFqZcYBUT95J-0g8MwlTU38Tcn3/view)

**התייחסות למשובים שקיבלנו בסטודיו:**



**בהתאם למשובים שקיבלנו, הוספנו דף מנהל ייעודי שמאפשר ניתוח ומעקב אחרי נתוני שימוש במערכת. בדף זה מוצגים גרפים סטטיסטיים הכוללים את הקטגוריות הנפוצות ביותר בחיפושים, המשתמשים הפעילים ביותר, וחלוקה לפי נושאים. כמו כן, ניתנה אפשרות למחיקת משתמשים על ידי המנהל לשם שמירה על סדר וניקיון במערכת.**

8. בשבוע של 22.6 תציגו את תוצרי הפרויקט שלכם. כל צוות יציג 15 דקות את המערכת, וכן אלמנטים מרכזיים של העבודה (סעיפים 1-4 של מסמך זה).  
יש להציג את הקוד והאתר - אין צורך להכין מצגת!  
בסיום ההצגה נשאל את חברי הצוות שאלות בנוגע להצגה ולנושאי הקורס.

מטרת ההצגה היא לתת משוב ומיקוד לקראת הצגת הפרויקט.

יש להשתבץ להצגה בקישור הבא, חובת ההצגה על כל חברי הצוות!

להזכירכם – **ציון אישי של 30%** יינתן לכל אחד מהמציגים לפי הצגה ומענה על שאלות:

<https://docs.google.com/spreadsheets/d/1xlO1uUu_Fk0H4Za0-uRry39XihSbcU7J/edit?gid=778935068#gid=778935068>

יש להגיש את כל התוצרים ב - GIT repository של הצוות , בתיקיית Project.

בנוסף, יש לוודא כי תיקיית ה- GIT ציבורית וכוללת את כל הקבצים של הפרויקט, כולל המסמכים, המצגת וכן הוראות הרצה.

**הנחיות הגשה:**

1.באחריותכם לוודא שהגשתם את כל התוצרים כנדרש ושהם הגיעו ליעדם.

לא תתאפשר בדיקה מחודשת של העבודה עקב טעויות בעת ההגשה!

2 .יש להגיש את התרגיל בקבוצות שהוגדרו בקורס.

3 .יש להגיש את התוצר הסופי של מסמך הפרויקט בפורמט WORD שייכלל בריפו של הפרויקט.

4.ניתן להפנות שאלות במייל לצוות הקורס, נא לשלוח שאלות לכלל הצוות . נא לציין את שם הקורס.

חלוקת ניקוד:

