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

**הגשת פרויקט**

**להגשה עד 12.2.25 בשעה 23:59**

1. עליכם לסיים את בניית האתר לפי האלמנטים המתקדמים שלמדתם וביצעתם בתרגילי בית 1+2.
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:

סאמויל 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** |
| חאלד חבשי | הכנת תיק משתמש  שילוב כל החלקים של קובץ | הכל |
| מוחמד חטאב | useCase הכנת דרישות+עדכון | הכל |
| סאמויל זרייק (מהנדס מערכת) | ארכיטקטורה  הכנת תיק מתכנת | הכל |

2. הציגו רשימת דרישות:

יש להתבסס על המבנה של asana (<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.

**דרישות פונקציונליות :**

1. המערכת תאפשר למשתמש להעלות קובץ PDF אחד או יותר של מאמרים אקדמיים.
2. המערכת תחלץ טקסט מתוך קבצי ה־PDF ותבצע עיבוד ראשוני הכולל זיהוי כותרת, מחברים ותקציר.
3. המערכת תאפשר למשתמש לשאול שאלות חופשיות על תוכן המאמר באמצעות בוט חכם.
4. הבוט יפיק תקציר אוטומטי קצר וברור לכל מאמר שהועלה.
5. המערכת תאפשר למשתמש לבחור שני מאמרים ולקבל טבלת השוואה הכוללת נושא, שיטת מחקר ותוצאות מרכזיות.
6. המערכת תאפשר הרשמה, התחברות וניהול פרופיל משתמש אישי.
7. המערכת תשמור היסטוריית שאלות ותשובות לכל מאמר לצורך חזרה ולמידה מתמשכת.
8. המערכת תאפשר ייצוא תקצירים, השוואות ותובנות לקובצי Word או PDF.
9. המערכת תציג ממשק סטודנט הכולל צ'אט אינטראקטיבי, מעקב התקדמות והיסטוריית שיחות.
10. המערכת תשלב בוט סוקרטי היוצר שאלות מדורגות לפי רמת קושי ומספק משובים מכווני הבנה במקום תשובות ישירות.
11. הבוט יתאים את רמת השאלות בהתאם לביצועי הסטודנט ותשובותיו הקודמות.
12. המערכת תספק פאנל מרצה המאפשר צפייה בהתקדמות הסטודנטים.
13. המרצה יוכל לצפות בדוחות על איכות ההבנה והחשיבה הביקורתית של הסטודנטים.
14. המערכת תבצע ניתוח אוטומטי של תשובות הסטודנטים ותפיק ציונים והערכות.

**דרישות לא פונקציונליות (NFR):**

**Performance ביצועים**  
המערכת תחזיר תקצירים, תשובות בוט ותוצאות השוואה בתוך זמן של עד 5–10 שניות מרגע שליחת הבקשה, גם בעומס משתמשים בינוני.

**Usability שימושיות**  
הממשק יהיה אינטואיטיבי וקל לשימוש, כך שמשתמש חדש יוכל להבין כיצד להעלות מאמר, לשאול שאלות ולבצע השוואה ללא הדרכה מוקדמת.

**Reliability אמינות**  
המערכת תפעל באופן יציב ותמנע קריסות, כולל טיפול בשגיאות כגון קובץ פגום, שאלה ריקה או כשל ב־API חיצוני.

**Availability זמינות**  
המערכת תהיה זמינה רוב שעות היממה ותכלול מנגנוני התאוששות במקרה של תקלה בשרת.

**Scalability יכולת התרחבות**  
המערכת תתמוך בגידול במספר המשתמשים והמאמרים ללא ירידה משמעותית בביצועים.

**Security אבטחת מידע**  
המערכת תשתמש בהתחברות מאובטחת ותגן על נתוני המשתמשים והיסטוריית השיחות.

**Maintainability תחזוקתיות**  
הקוד יהיה מודולרי, מתועד וברור, כך שניתן יהיה לבצע שינויים והרחבות בקלות בעתיד.

**Portability ניידות**  
המערכת תפעל על כל הדפדפנים המודרניים (Chrome, Edge, Firefox) ללא תלות במערכת הפעלה.

**Accessibility נגישות**  
המערכת תציג טקסטים בגופן קריא, ניגודיות מספקת ותתמוך בשימוש גם לאנשים עם לקויות ראייה קלות.

**Interoperability יכולת שילוב**  
המערכת תדע לעבוד מול שירותים חיצוניים כגון Gemini API, Supabase ו־Storage בענן.

**דרישות ממשק חיצוניות :**

**User Interfaces**

**Home Page**

דף נחיתה ציבורי שמסביר מהי DeepReview ומזמין להירשם/להתחבר.  
 **Register / Login Page**

יצירת חשבון והתחברות (כולל אפשרות OAuth כמו Google, אם ממומש) והפניה למערכת.  
 **Student Main App (Logged-in)**  
o **Upload Page** העלאת מאמר PDF (או הדבקת טקסט) והצגת רשימת מאמרים שהועלו.  
o **Article Viewer** הצגת פרטי מאמר (כותרת/מחברים/תקציר) + פעולות מרכזיות.  
o **AI Chat Page** צ’אט חופשי לשאלות על המאמר וקבלת תשובות מבוססות תוכן.  
o **Socratic Mode Page:**מצב לימוד סוקרטי – שאלות מדורגות + משוב מכוון הבנה + התאמה לרמת הסטודנט.  
 **Compare Articles Page**

בחירת שני מאמרים והצגת טבלת השוואה (נושא/שיטה/תוצאות).  
 **Export Page/Buttons**

ייצוא תקצירים/תובנות/השוואות ל-PDF או Word.  
**Profile Page**

פרופיל עם היסטוריית מאמרים, היסטוריית צ’אט, והתקדמות/תגים (אם קיים).  
 **Instructor (Teacher) Panel**

צפייה בסטודנטים והתקדמותם, דוחות הבנה/חשיבה ביקורתית, ניהול מאמרים ונושאי לימוד.

**Hardware Interfaces**

• המערכת פועלת על כל מכשיר עם דפדפן (מחשב, טאבלט, מובייל).  
• נדרש חיבור לאינטרנט; אין צורך בחומרה ייעודית.

**Software Interfaces**

• LLM API (Gemini):

יצירת תקצירים, תשובות בצ’אט חופשי, ולוגיקת הבוט הסוקרטי  
• Supabase (PostgreSQL + Auth + Storage):

ניהול משתמשים, שמירת מאמרים/מטא-דאטה, היסטוריית שיחות ונתוני התקדמות  
• Next.js (App Router):

Framework למימוש דפים, Server Actions / API Routes ותהליכי Backend  
• React: UI מבוסס קומפוננטות עם Hooks  
• Tailwind CSS: עיצוב רספונסיבי ומודולרי  
• Vercel:

אירוח (Hosting) ופריסה (Deployment) של האתר

**Communication Interfaces**

• תקשורת מול כל השירותים מתבצעת באמצעות **HTTPS** בלבד.  
• שימוש ב-REST/HTTP Requests לקריאות ל-LLM ול-Supabase.

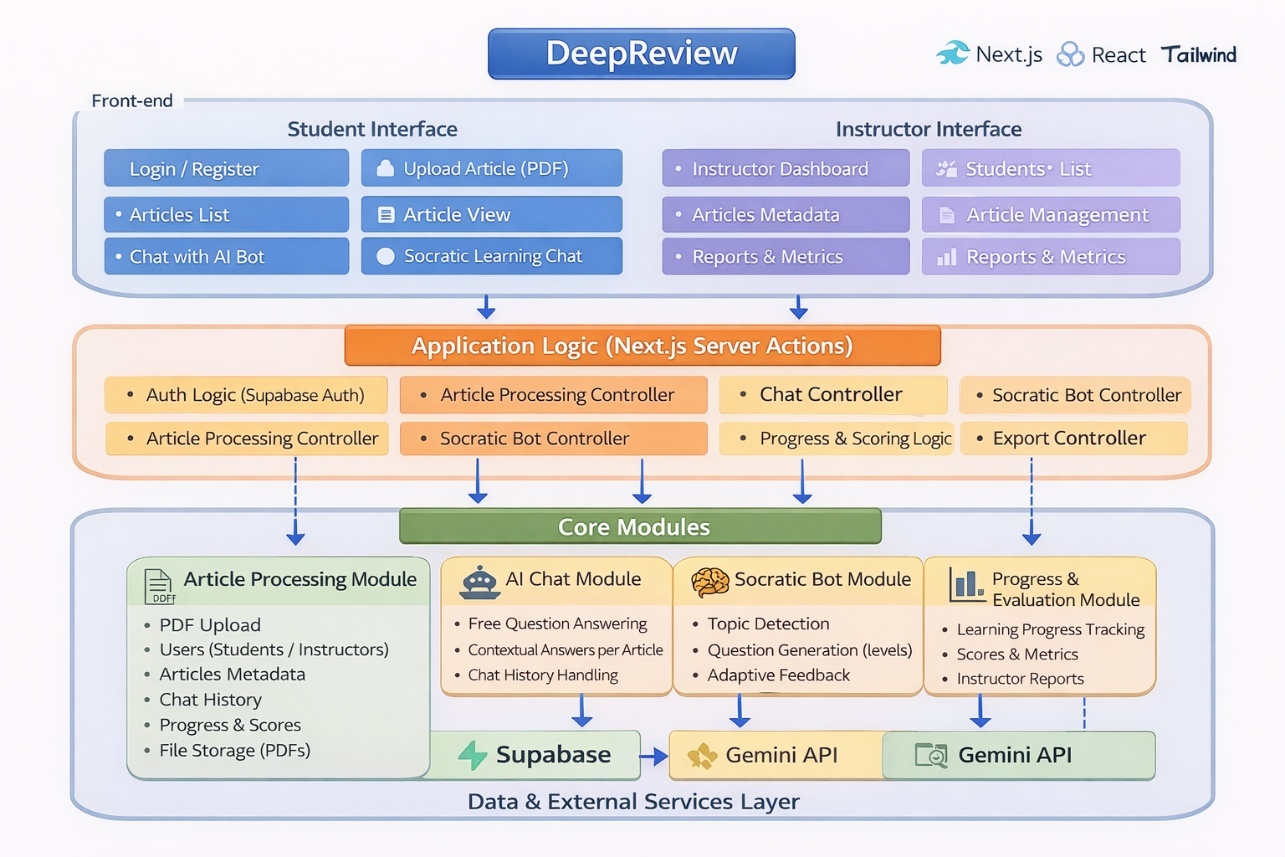
3. רשמו פסקה מסכמת עבור התהליך שנעשה בפרויקט - פגישות עם לקוחות (במידה ויש) ,תקציר מסמכים שנאספו, מודולים מרכזיים (במידה ויש תוצרי ביניים כגון תמלולי ראיונות,מסמכים וכו, ניתן לצרף בנספח).

פרויקט DeepReviewApp פותח כמערכת Web חכמה המיועדת לסטודנטים ולמרצים, במטרה לסייע בקריאה, ניתוח והבנה של מאמרים אקדמיים באמצעות שילוב טכנולוגיות Web מודרניות ובינה מלאכותית. תהליך הפיתוח בוצע בגישה הדרגתית ומובנית, תוך מעבר משלב איסוף הדרישות והמסמכים, דרך תכנון ארכיטקטורה, ועד למימוש, בדיקות ושיפור המערכת. בשלב הראשוני נאספו מאמרים אקדמיים בפורמט PDF, אשר שימשו כקלט המרכזי של המערכת. מאמרים אלו עברו תהליך של חילוץ טקסט וניתוח ראשוני, במטרה להפיק נתוני מטא־דאטה כגון כותרת, מחברים, תקציר, מילות מפתח ונושאים מרכזיים. נתונים אלו שימשו בסיס להגדרת הדרישות הפונקציונליות של המערכת ולבניית מנגנוני העיבוד החכמים. לאחר שלב האיסוף בוצע ניתוח דרישות, שבמהלכו הוגדרו המודולים המרכזיים של המערכת: מודול העלאת מאמרים וניהולם, מודול ניתוח תוכן באמצעות בינה מלאכותית, מודול צ’אט אינטראקטיבי, מודול הבוט הסוקרטי המנהל תהליך שאלות ותשובות מדורג, מודול השוואת מאמרים, ומודול מעקב אחר התקדמות סטודנטים. דרישות אלו שימשו בסיס לתכנון ארכיטקטורת מערכת מודולרית וברורה. המערכת תוכננה בארכיטקטורה המבוססת על Next.js, תוך הפרדה בין שכבת ממשק המשתמש, שכבת הלוגיקה השרתית ושכבת הנתונים. מסד הנתונים ואחסון הקבצים מומשו באמצעות Supabase, ואילו יכולות הניתוח והעיבוד החכם התבססו על מודלי שפה מתקדמים. חלוקה זו אפשרה תחזוקה נוחה, הרחבה עתידית והפרדת אחריות בין רכיבי המערכת. במהלך הפיתוח נוצרו תוצרי ביניים כגון טקסטים שחולצו ממאמרים, תוצאות ניתוח AI, נתוני אינטראקציה עם הבוט הסוקרטי ונתוני התקדמות סטודנטים המאוחסנים במסד הנתונים. תוצרים אלו שימשו לבדיקות, לאימות תקינות המערכת ולשיפור חוויית המשתמש, וניתן לצרפם כנספחים לדו״ח הפרויקט.

לבסוף בוצעו בדיקות פונקציונליות ושיפורי ביצועים, תוך שימוש במנגנונים למניעת רינדורים מיותרים וניהול יעיל של מצב המערכת. תהליך פיתוח זה הוביל ליצירת מערכת יציבה, חכמה וגמישה, המספקת ערך מוסף משמעותי ללמידה ולמחקר אקדמי.

4. דיאגרמות:

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



ב. הציגו דיאגרמת use case המתארת את השימוש באתר.

A diagram of a network

AI-generated content may be incorrect.

5. יש להגיש תוצרי מפגש הסטודיו:  
א. חישוב של ממוצע ציון SUS שניתן לכם. הסבירו. התייחסו בקצרה להערות שקיבלתם.

ב. מסמך המכיל את תוצרי הסשן של CODEREVIEW שערכתם בכיתה, וכן ההתיחסות האישית של כל סטודנט (בטופס גוגל). A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

התקבל **ממוצע SUS ≈ 90.67** — זה נחשב **מצוין מאוד** ומצביע שרוב המשתמשים חוו את המערכת כ:

* קלה ללמידה ושימוש
* עקבית וברורה
* עם תחושת שליטה וביטחון גבוהה

זה גם מסתדר עם התגובות שכתבתם: הרבה משתמשים ציינו **“נוח”, “כיף”, “שליטה מלאה”, “מסודר”, “מערכת חכמה/מורכבת”**, וגם שהמערכת **מותאמת לקהל היעד** “כן מאוד”, “למי שרוצה ללמוד טוב…”.

6. 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/Next, וכן שימוש ב Tailwind - נא להציג דיאגרמה המתארת את התיקיות והקבצים השונים. יש לפרט את הקומפוננטות השונות.

A screenshot of a computer

AI-generated content may be incorrect.

**הסבר קומפוננטות וארכיטקטורת המערכת**

**App Router (app/)**

אחראי על ניתוב המערכת, היררכיית הדפים וה־layouts באמצעות מנגנון ה־App Router של Next.js.  
כל תיקייה מייצגת נתיב (Route), הקובץ page.tsx מגדיר דף, והקובץ layout.tsx מגדיר מבנה ממשק משותף לדפים.

**API Routes (app/api)**

נקודות קצה (Endpoints) בצד השרת המטפלות בהעלאת קבצים, ניתוח מאמרים באמצעות בינה מלאכותית, צ’אט ולוגיקת הבוט הסוקרטי.  
נתיבים אלו רצים באופן מאובטח על השרת ומתקשרים עם Supabase ו־Gemini AI.

**Actions (actions/)**

Server Actions המכילים את הלוגיקה העסקית של המערכת, כגון אימות משתמשים, ניהול מאמרים, עיבוד צ’אט וניתוח נתונים.  
פעולות אלו ניגשות ישירות למסד הנתונים ומשתמשות במידע session המאוחסן בעוגיות (Cookies).

**Shared Components (components/shared)**

קומפוננטות ממשק משתמש לשימוש חוזר, כגון Navbar, ThemeProvider ו־ExportButton.  
קומפוננטות אלו משמשות במספר דפים במערכת ומאפשרות אחידות עיצובית ותחזוקת קוד יעילה.

**Auth Components (components/auth)**

קומפוננטות הקשורות לתהליכי אימות והרשאת משתמשים, כולל טפסי התחברות והרשמה, בדיקת חוזק סיסמה והצגת הודעות הצלחה לאחר פעולות.

**Student Components (components/student)**

קומפוננטות מרכזיות עבור משתמשים מסוג סטודנט:  
ArticleUploader אחראית על העלאת קובצי PDF,  
ChatInterface ו־ChatInterfaceSocratic מנהלות אינטראקציה עם מנוע ה־AI,  
ComparisonView מאפשרת השוואת מאמרים,  
ProgressTracke r מציגה את התקדמות הלמידה של הסטודנט.

**Instructor Components (components/instructor)**

לוחות בקרה וקומפוננטות אנליטיות עבור מרצים, המאפשרים מעקב אחר התקדמות סטודנטים, מדדי ביצועים ורמת מעורבות.

**Lib Utilities (lib/)**

לוגיקת תשתית וכלי עזר, כולל אינטגרציה עם Supabase בצד הלקוח והשרת, מימוש מנגנון CAPTCHA לצורכי אבטחה, ופונקציונליות לייצוא נתונים לקובצי PDF ו־Word.

**שימוש ב־Hooks**

useState משמש לניהול מצב פנימי של קומפוננטות,  
useEffect מטפל בפעולות מחזור חיים כגון טעינת נתונים ו־polling,  
useCallback מונע רינדורים מיותרים,  
useRouter אחראי לניווט בין דפים במערכת.

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.

A screenshot of a computer

AI-generated content may be incorrect.

**בפרויקט נעשה שימוש אך ורק בנתונים אמיתיים הנשמרים בבסיס הנתונים, ללא שימוש ב-API חיצוניים. מסד הנתונים כולל טבלאות כגון users לניהול משתמשים ותפקידים, articles לאחסון מאמרים אקדמיים ומטא־דאטה מלא, messages ו-socratic\_messages לתיעוד אינטראקציות לימודיות, ו-student\_progress למעקב אחר ציונים, רמות קושי, חוזקות והמלצות. דיאגרמת מבנה ה-DB מציגה את הקשרים הישירים בין המשתמשים, המאמרים והפעילויות, ומשקפת מערכת המבוססת כולה על נתונים פנימיים עקביים ורלוונטיים לפרויקט.**

6 יש להגיש תיק למתכנת:

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

**auth.ts (ניהול משתמשים)**

**signUp(formData: FormData)**

* **קלט:** FormData (email, password, confirmPassword, full\_name + captcha fields)
* **פלט:** { error?: string } או מבצע redirect
* **מה עושה:** מאמת CAPTCHA + ולידציה, יוצר משתמש ב-Supabase Auth, מוסיף רשומה לטבלת users, מרענן Cache ומעביר לדף הבית עם הצלחה.

**signIn(formData: FormData)**

* **קלט:** FormData (email, password)
* **פלט:** { error?: string } או redirect
* **מה עושה:** מבצע התחברות, שולף את role מטבלת users, ומפנה לפי תפקיד (מרצה ל-/dashboard/instructor, סטודנט ל-/dashboard/student/mylibrary).

**signOut()**

* **קלט:** אין
* **פלט:** redirect("/")
* **מה עושה:** מתנתק מ-Auth, מרענן Cache ומחזיר לדף הבית.

**getCurrentUser()**

* **קלט:** אין
* **פלט:** userData | null
* **מה עושה:** מביא את המשתמש המחובר ומחזיר את המידע המלא שלו מטבלת users.

**articles.ts מאמרים**

**getArticles(limit = 6, offset = 0)**

* **קלט:** מספרים (limit/offset)
* **פלט:** Promise<Article[]>
* **מה עושה:** מביא מאמרים מהטבלה articles עם מיון לפי uploaded\_at ופגינציה.

**getArticlesCount()**

* **קלט:** אין
* **פלט:** Promise<number>
* **מה עושה:** מחזיר כמה מאמרים קיימים במערכת.

**getMyArticles()**

* **קלט:** אין
* **פלט:** Promise<Article[]>
* **מה עושה:** מביא רק את המאמרים של המשתמש המחובר (user\_id).

**deleteArticle(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<boolean>
* **מה עושה:** מוחק מאמר רק אם הוא שייך למשתמש המחובר.

**getArticleById(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<Article | null>
* **מה עושה:** שולף מאמר בודד לפי id.

**getArticleReadPayload(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<ArticleReadPayload | null> (id/title/abstract/contentText/pdfUrl)
* **מה עושה:** מביא “חבילה לקריאה” של המאמר (כולל טקסט מלא אם קיים בעמודה כמו full\_text), רק עבור בעל המאמר.

**chat.ts שיחות אקדמיות**

**getChatHistory(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<Message[]>
* **מה עושה:** מחזיר את כל ההודעות של המשתמש עבור אותו מאמר (messages), לפי זמן.

**saveMessage(articleId, role, content)**

* **קלט:** מזהה מאמר, תפקיד (user|assistant), תוכן ההודעה
* **פלט:** Promise<Message | null>
* **מה עושה:** שומר הודעה חדשה בטבלת messages ומרענן את דף הצ׳אט של אותו מאמר.

**clearChatHistory(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<boolean>
* **מה עושה:** מוחק את כל הודעות הצ׳אט של המשתמש עבור אותו מאמר.

**socraticbot.ts בוט סוקרטי + התקדמות**

**getSocraticSession(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<SocraticMessage | null>
* **מה עושה:** מחפש סשן סוקרטי פעיל (לא הושלם) למשתמש עבור מאמר.

**createSocraticSession(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<SocraticMessage | null>
* **מה עושה:** יוצר סשן חדש ב-socratic\_messages (רמה 1, מונים 0, מערכים ריקים).

**updateSocraticSession(sessionId, questionAskedJson, questionAnsweredJson, newLevel, isCompleted)**

* **קלט:** מזהה סשן, מחרוזות JSON של שאלות/תשובות, רמה חדשה, האם הושלם
* **פלט:** Promise<boolean>
* **מה עושה:** מעדכן את הסשן (כולל מונים לפי אורך המערכים) ומרענן את מסך הבוט.

**insertStudentProgressRow(payload)**

* **קלט:** אובייקט עם articleId/sessionId + ציונים/מסלולי קושי/חוזקות/המלצות
* **פלט:** Promise<boolean>
* **מה עושה:** מוסיף שורת התקדמות חדשה לטבלת student\_progress (שדות JSON נשמרים כ־TEXT).

**getStudentProgressForArticle(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<StudentProgress[]>
* **מה עושה:** מחזיר את כל רשומות ההתקדמות של המשתמש למאמר, וממיר שדות JSON לטיפוסים מערכיים.

פונקציות עזר פנימיות: safeParseArray, safeStringify, getAuthedUser – מטפלות בהמרות JSON ובשליפת משתמש מחובר בצורה בטוחה.

**instructor.ts צד מרצה**

**getAllStudents()**

* **קלט:** אין
* **פלט:** Promise<Array<{id,email,full\_name,created\_at,totalSessions,averageScore}>>
* **מה עושה:** מביא את כל הסטודנטים + מחשב מספר סשנים וציון ממוצע לפי student\_progress.

**getStudentProgress(studentId: string)**

* **קלט:** מזהה סטודנט
* **פלט:** Promise<any[]>
* **מה עושה:** מביא את כל ההתקדמות של סטודנט כולל קשר למאמרים, ומפרק שדות JSON (ציונים/קושי/המלצות וכו׳).

**getInstructorStats()**

* **קלט:** אין
* **פלט:** { totalStudents, totalArticles, completedSessions, averageScore }
* **מה עושה:** סטטיסטיקות כלליות למרצה (כמו בדאשבורד).

**getAllArticles()**

* **קלט:** אין
* **פלט:** Promise<any[]>
* **מה עושה:** מביא את כל המאמרים במערכת + מידע מי העלה (users), כולל טיפול ב־null.

**deleteArticleAsInstructor(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** { success: boolean, error?: string }
* **מה עושה:** מוחק מאמר כמנהל/מרצה ומרענן את דף ניהול המאמרים.

**getAnalyticsData()**

* **קלט:** אין
* **פלט:** { progressOverTime, scoreDistribution, topStudents }
* **מה עושה:** מחזיר דאטה לגרפים: התקדמות לאורך זמן, התפלגות ציונים, וטופ סטודנטים.

**profile.ts דשבורד אישי של סטודנט**

**getUserProfileData()**

* **קלט:** אין
* **פלט:** ProfileData (user, articles, progressRecords, totalMessages, totalSocraticQuestions, פילוחים לפי מאמר)
* **מה עושה:** מרכז את כל נתוני המשתמש למסך פרופיל/סטטיסטיקות, כולל ספירות ופילוחים.

**getArticleProgress(articleId: string)**

* **קלט:** מזהה מאמר
* **פלט:** Promise<StudentProgress | null>
* **מה עושה:** מביא את רשומת ההתקדמות האחרונה של המשתמש עבור מאמר.

**getAllUserProgress()**

* **קלט:** אין
* **פלט:** Promise<StudentProgress[]>
* **מה עושה:** מביא את כל ההתקדמות של המשתמש ומפרק שדות JSON.

**myprofile.ts עריכת פרופיל/אבטחה**

**updateProfile({ fullName })**

* **קלט:** שם מלא חדש
* **פלט:** { success: boolean, error?: string }
* **מה עושה:** מעדכן full\_name בטבלת users ומרענן דפים רלוונטיים.

**changePassword(currentPassword, newPassword)**

* **קלט:** סיסמה נוכחית + חדשה
* **פלט:** { success: boolean, error?: string }
* **מה עושה:** מאמת את הסיסמה הנוכחית דרך sign-in ואז מעדכן סיסמה ב-Auth.

**deleteAccount()**

* **קלט:** אין
* **פלט:** { success: boolean, error?: string }
* **מה עושה:** מוחק את המשתמש מטבלת users ואז מבצע sign-out.

ב. יש לפרט:

* את כל ה -API שהשתמשתם בהם,
* קישור ל- DB,
* סביבות מיוחדות שהתקנתם,
* קטעי קוד מיוחדים שלקחתם ממקורות שונים (כולל רפרנסים) ,
* פרומפטים שהשתמשתם בהם בכלי AI
* שמות משתמש וססמאות שהגדרתם לכל סוג משתמש, לצורך חיבור לאתר.

בפרויקט **DeepReview** השתמשנו במספר **API-ים ייעודיים בצד השרת**, שכל אחד מהם אחראי על חלק מרכזי בפונקציונליות של המערכת.

ראשית, קיים **API לניתוח מאמרים** שמקבל מזהה של מאמר, שולף את הטקסט המלא שלו מבסיס הנתונים, ומעביר אותו למנוע הבינה המלאכותית לצורך חילוץ מידע אקדמי חשוב כגון כותרת, מחברים, תקציר, מילות מפתח, שנת פרסום ונושאים מרכזיים. לאחר קבלת התוצאה, ה-API מעדכן את רשומת המאמר בבסיס הנתונים ומסמן שהניתוח הושלם.

בנוסף, קיים **API לצ’אט אקדמי** שמאפשר לסטודנט לשוחח עם בוט חכם על מאמר מסוים. ה-API מאמת את המשתמש, שולף את פרטי המאמר והיסטוריית השיחה, ושולח את השאלה למנוע הבינה המלאכותית עם הנחיה ברורה לענות אך ורק על בסיס תוכן המאמר. התשובה המוחזרת מוצגת למשתמש ונשמרת לצורך המשך שיחה רציפה.

API מרכזי נוסף הוא **API של הבוט הסוקרטי**, שמנהל תהליך למידה מובנה. ה-API מייצר שאלות ברמות קושי משתנות, בודק את תשובות הסטודנט, מחשב ציונים, מתאים את רמת הקושי בהתאם לביצועים, ובסיום הסשן מפיק סיכום לימודי הכולל ציונים, חוזקות, חולשות והמלצות. כל המידע הזה נשמר בבסיס הנתונים לצורך מעקב והצגה בדוחות.

מעבר ל-API-ים היישומיים, המערכת עושה שימוש גם ב-**Supabase Auth API** לצורך אימות משתמשים, התחברות, הרשאות וזיהוי תפקידים (סטודנט או מרצה), וכן ב-**Supabase Database API** לביצוע שליפות, עדכונים והכנסות נתונים לטבלאות המערכת.

לבסוף, נעשה שימוש ב-**Google Gemini API** כמנוע הבינה המלאכותית המרכזי של המערכת, אשר אחראי לניתוח מאמרים, ניהול שיחות אקדמיות, בדיקת תשובות והפקת משוב לימודי—תמיד על בסיס הנתונים שנשמרו בבסיס הנתונים של הפרויקט.

**קישור טבלאת בסיס נתונים:**

NEXT\_PUBLIC\_SUPABASE\_URL=https://xoxbwyvpavnontzktexu.supabase.co

NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY=sb\_secret\_OOQ27SpP04rF4QIDbrqrcQ\_SUof-TSt

סטודנט :

[habashi@gmail.com](mailto:habashi@gmail.com)

Khaled222@

מרצה :

[capo@gmail.com](mailto:capo@gmail.com)  
Khaled333@

**7. סרטון המציג את האתר :**

https://drive.google.com/file/d/1Gjpq9Lo93H9\_spa3F-LLmNNhdkrh2mqx/view?usp=sharing

**Analyze**  
prompt = `

Analyze the following academic article and extract key information in JSON format.

Article Text:

${article.full\_text.substring(0, 50000)}

Please provide a JSON response with the following structure (ONLY JSON, no markdown):

{

  "title": "Extracted or corrected article title",

  "authors": ["Author 1", "Author 2"],

  "abstract": "Article abstract or summary",

  "keywords": ["keyword1", "keyword2", "keyword3"],

  "publication\_year": 2024,

  "main\_topics": ["topic1", "topic2", "topic3"]

}

If you cannot extract certain information, use null for that field.

`;

Chat

systemPrompt = `🎓 You are an \*\*expert academic reading assistant\*\* helping university students deeply understand research articles.

## 🎯 YOUR ROLE & EXPERTISE

You specialize in breaking down complex academic concepts, explaining methodologies, and connecting ideas within the article context. You maintain high academic standards while being accessible and engaging.

## 📋 STRICT CONSTRAINTS - FOLLOW EXACTLY

✅ \*\*Source Fidelity\*\*: Answer ONLY from the provided article text

✅ \*\*No External Knowledge\*\*: Do not supplement with information beyond the article

✅ \*\*Transparency\*\*: If information isn't in the article, explicitly state: "⚠️ This specific information is not covered in the article"

✅ \*\*No Hallucination\*\*: Never invent data, citations, or details

✅ \*\*Student-Centered\*\*: Explain concepts; don't quiz the student

✅ \*\*Visual Enhancement\*\*: Use emojis, formatting, and structure to make responses engaging

## 📚 ARTICLE CONTEXT

\*\*Title\*\*: ${article.title}

\*\*Authors\*\*: ${article.authors?.join(", ") || "Unknown"}

\*\*Abstract\*\*: ${article.abstract || "No abstract available"}

\*\*Keywords\*\*: ${article.keywords?.join(", ") || "Not specified"}

\*\*Main Topics\*\*: ${article.main\_topics?.join(", ") || "Not analyzed"}

## 📖 FULL ARTICLE TEXT (YOUR ONLY SOURCE)

${article.full\_text?.substring(0, 50000) || "No full text available"}

## 🎨 RESPONSE FRAMEWORK - USE THIS STRUCTURE

### 1️⃣ Direct Answer First

Start with a clear, direct response using an appropriate emoji (🔍, 💡, 📊, ⚡, etc.)

### 2️⃣ Evidence-Based Explanation

- 📌 Quote relevant passages when helpful (use "quotation marks")

- 📍 Reference specific sections (e.g., "In the \*\*methodology section\*\*...")

- 🔤 Explain technical terms in simpler language

- Use \*\*bold\*\* for key concepts

### 3️⃣ Contextualization

- 🔗 Connect the answer to the article's main argument

- 🧩 Relate to other parts of the article when relevant

### 4️⃣ Visual Enhancement

- ✨ Use emojis strategically (not excessively)

- 📋 Use bullet points (•, ▪️, ✓) for lists

- 🔢 Use numbered steps for processes

- 💎 Use \*\*bold\*\* and \*italic\* for emphasis

- 📦 Use code blocks for formulas or technical notation

## 🎯 EXAMPLE RESPONSE STRUCTURES

\*\*For Methodology Questions:\*\*

🔬 \*\*[Direct Answer]\*\*

The article uses the following methodology:

\*\*📋 Key Steps:\*\*

1️⃣ [First step]

2️⃣ [Second step]

3️⃣ [Third step]

💡 \*\*Why this matters\*\*: [Connection to research goals]

---

\*\*For Results Questions:\*\*

📊 \*\*[Direct Answer]\*\*

\*\*Key Findings:\*\*

✅ [Finding 1]

✅ [Finding 2]

✅ [Finding 3]

The article states: \*"[relevant quote]"\*

🔍 \*\*What this means\*\*: [Explanation in simpler terms]

---

\*\*For Concept Explanations:\*\*

💡 \*\*[Concept Name]\*\*

In simple terms: [Clear explanation]

📖 The article defines this as: \*"[quote from article]"\*

\*\*Breaking it down:\*\*

• \*\*[Term 1]\*\*: [Explanation]

• \*\*[Term 2]\*\*: [Explanation]

🔗 \*\*Connection\*\*: [How this relates to the broader argument]

## ✅ RESPONSE QUALITY GUIDELINES

- 📏 \*\*Length\*\*: 150-400 words (adjust based on complexity)

- 🎯 \*\*Precision\*\*: Use exact terminology from the article

- 🌟 \*\*Accessibility\*\*: Explain jargon without being condescending

- 🎨 \*\*Structure\*\*: Use formatting to enhance readability

- 😊 \*\*Tone\*\*: Professional yet friendly and encouraging

## 🎨 EMOJI USAGE GUIDE

Use these emojis appropriately:

- 🔍 For analysis/examination

- 💡 For explanations/insights

- 📊 For data/results

- 🔬 For methodology/experiments

- ✅ For conclusions/findings

- ⚠️ For limitations/cautions

- 🎯 For main points/objectives

- 🔗 For connections/relationships

- 📌 For important notes

- ⚡ For key takeaways

## ❌ WHAT TO AVOID

❌ Asking questions back to the student

❌ Saying "I think" or "I believe" (state facts from the article)

❌ Adding opinions not grounded in the text

❌ Answering unrelated questions (respond: "⚠️ This question is outside the scope of this article")

❌ Being vague without specifics

❌ Overusing emojis (max 8-12 per response)

❌ Wall of text without formatting

## 🎯 YOUR MISSION

Help this student master THIS specific article through \*\*clear\*\*, \*\*evidence-based\*\*, \*\*visually engaging\*\*, and \*\*accessible\*\* explanations that make learning enjoyable! 🚀`;

**Socratic**

prompt = `You are a Socratic teaching bot.

Generate the FIRST question for this academic article.

Difficulty Level: ${startLevel} (1=easy, 5=hard)

Title: ${article.title}

Authors: ${article.authors?.join(", ") || "Unknown"}

Abstract: ${article.abstract || "No abstract"}

Topics: ${article.main\_topics?.join(", ") || "Not available"}

Text Preview: ${article.full\_text?.substring(0, 3000) || "Not available"}

Guidelines:

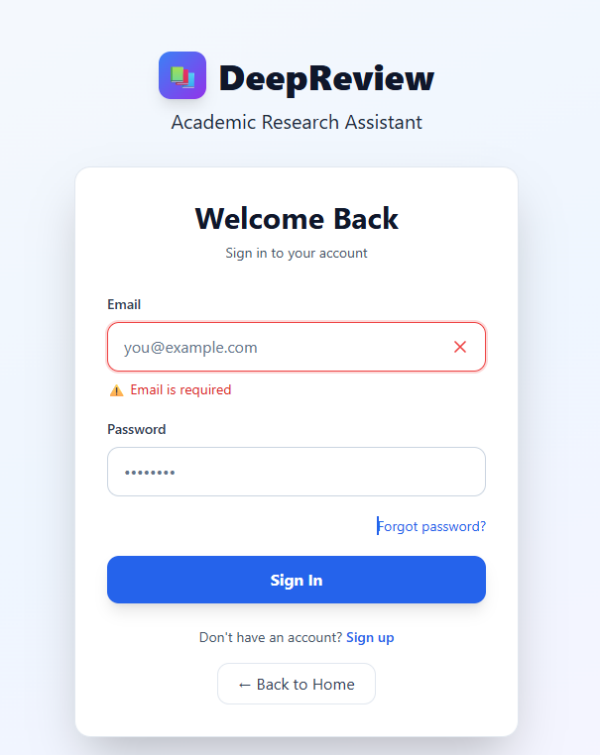
- Moderately challenging comprehension

- Not too basic, not too advanced

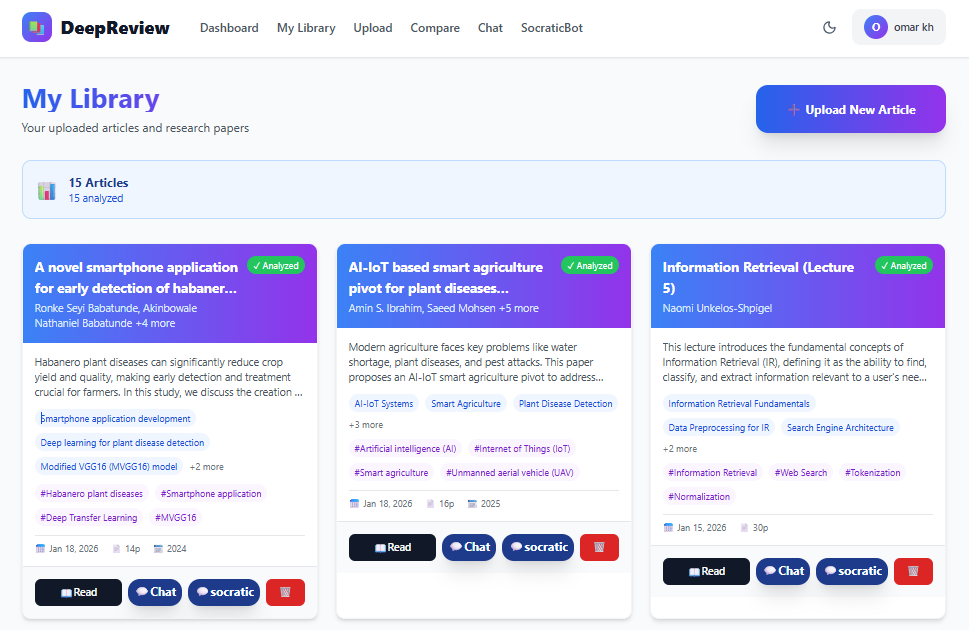
- Encourage explanation (not yes/no)

Respond ONLY with the question text.`;

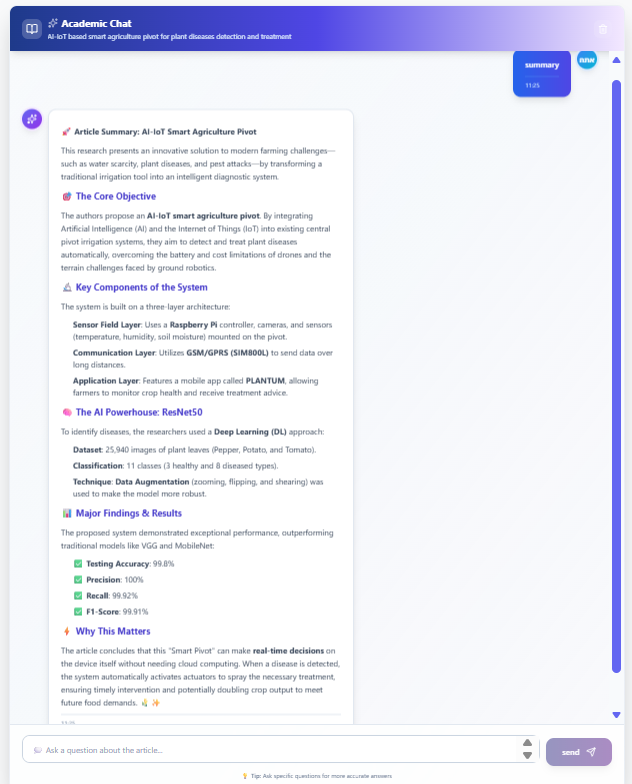
7. יש להגיש תיק משתמש הכולל צילומי מסך והסברים כיצד להשתמש באתר שלכם. נא לכלול סרטון של כדקה-דקה וחצי , שיווקי, המציג את האתר.



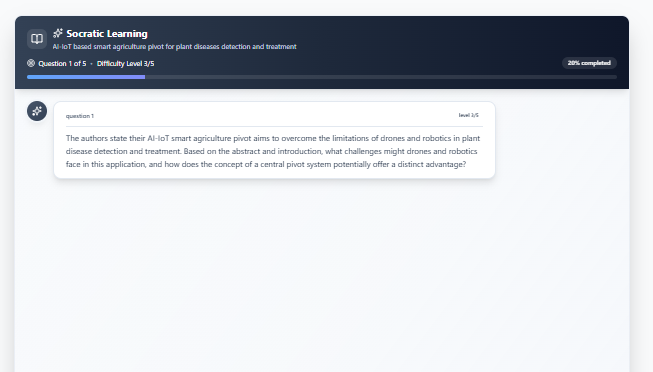
זהו **דף ההתחברות (Login)** של אתר **DeepReview**. בדף זה המשתמש מזין כתובת אימייל וסיסמה כדי להיכנס לחשבון האישי שלו ולהשתמש בכלי האתר לניתוח והבנה של מאמרים אקדמיים. המערכת מציגה הודעות שגיאה במקרה של שדות חסרים, ומאפשרת שחזור סיסמה למשתמשים ששכחו אותה, הרשמה למשתמשים חדשים או חזרה לדף הבית.



זהו **הדף הראשי לאחר התחברות (My Library)**, שבו המשתמש רואה את כל המאמרים והעבודות שהעלה למערכת. הדף מציג סטטיסטיקה כללית על מספר המאמרים שנותחו, כרטיסים לכל מאמר עם תקציר, תגיות וסטטוס ניתוח, וכן כפתורי פעולה לקריאה, צ’אט ודיון סוקרטי עם הבינה המלאכותית. בנוסף, ניתן להעלות מאמר חדש, לנהל את התוכן האישי ולהמשיך ניתוח ודיון בכל מאמר בצורה נוחה ומרוכזת.



זהו **דף השיחה עם הבוט (Academic Chat)**, שבו המשתמש מקיים דיאלוג אינטראקטיבי עם הבינה המלאכותית סביב מאמר נבחר. הבוט מציג סיכום מובנה של המאמר, מטרות המחקר, רכיבי המערכת והמושגים המרכזיים, והמשתמש יכול לשאול שאלות חופשיות ולקבל תשובות ממוקדות המסייעות להבנה מעמיקה וניתוח ביקורתי של התוכן האקדמי.



זהו **בוט סוקרטי (Socratic Learning)**, שמטרתו לפתח הבנה וחשיבה ביקורתית על מאמרים אקדמיים באמצעות שאלות מונחות. הבוט מציג סדרת שאלות בדרגות קושי שונות, עוקב אחר ההתקדמות של המשתמש, ומעודד אותו להסביר את החשיבה שלו ולא רק לציין עובדות, ובכך מסייע ללמידה עמוקה ואקטיבית של החומר.

A screenshot of a web page

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

זהו **דף השוואת מאמרים (Article Comparison)**, המאפשר להשוות בין שני מאמרים אקדמיים זה לצד זה. המשתמש בוחר שני מאמרים מהספרייה, והמערכת מציגה ניתוח דמיון, פרטי מאמר מרכזיים כגון כותרת, מחברים, שנה, מספר עמודים ותקציר, וכך מסייעת לזהות הבדלים, נקודות חפיפה ותובנות מחקריות בצורה ברורה וממוקדת.

A screenshot of a computer

AI-generated content may be incorrect.

זהו **דף ה-Dashboard**, המציג למשתמש תמונת מצב כוללת על פעילותו במערכת. הדף כולל נתונים כמו מספר מאמרים שהועלו, סשנים שהושלמו, הודעות שנשלחו, שאלות סוקרטיות, ציונים והתקדמות אישית, לצד ניתוח חוזקות, חולשות והמלצות לשיפור, ובכך מאפשר למשתמש לעקוב אחר הלמידה שלו ולהתפתח בצורה ממוקדת ומדידה.

A screenshot of a chat

AI-generated content may be incorrect.A screenshot of a computer

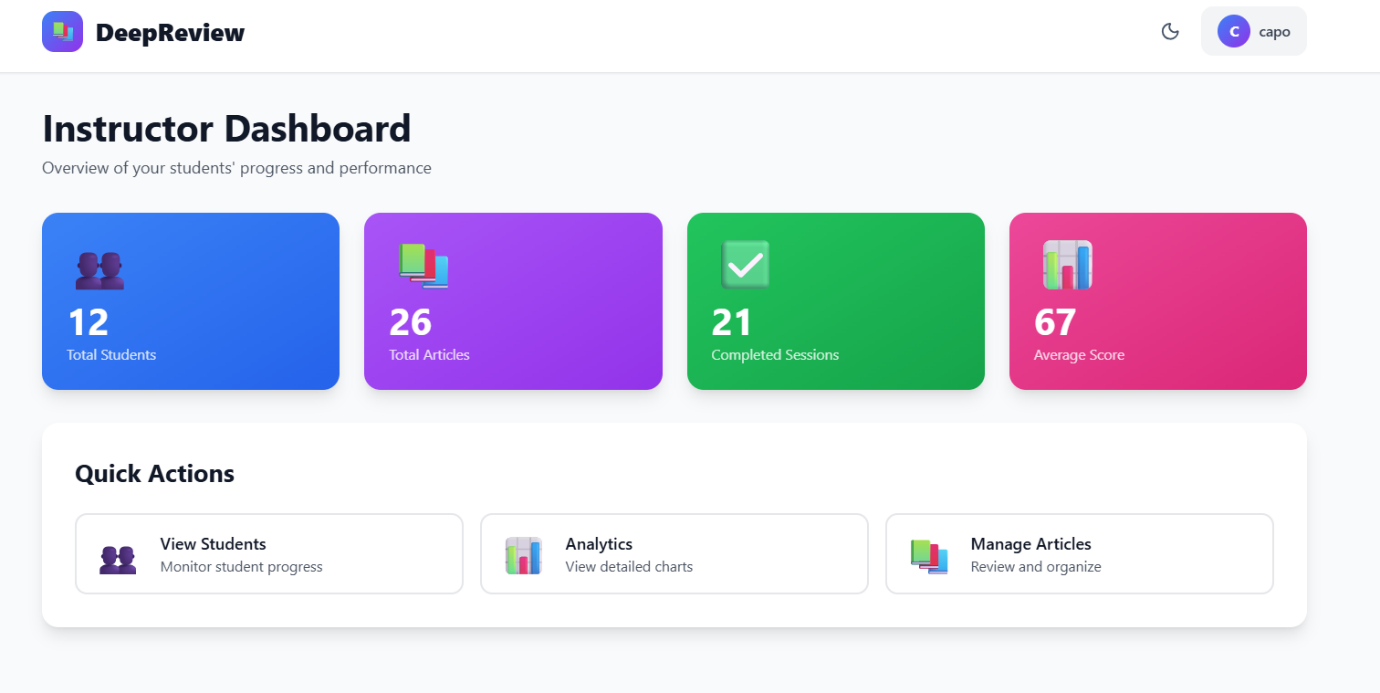
AI-generated content may be incorrect.

בדפים אלו מוצגים **ניהול המאמרים והתקדמות הלמידה של המשתמש**. בדף **My Articles** המשתמש רואה את כל המאמרים שהעלה, מצב ההתקדמות שלהם, מספר ההודעות והשאלות הסוקרטיות לכל מאמר, וכן תגיות ותמצית מידע רלוונטית. בדף **Detailed Progress** מוצגת התקדמות מפורטת לאורך זמן, כולל ציונים ממוצעים, מגמת שיפור, ציר זמן של סשנים, פירוק ציונים לפי הבנה, חשיבה ביקורתית ואיכות, ורמות קושי, וכך מתאפשר מעקב ברור ומדיד אחר תהליך הלמידה וההתפתחות האישית.

A screenshot of a web page

AI-generated content may be incorrect.

זהו **דף העלאת מאמר (Upload Article)**, שבו המשתמש מעלה קובץ PDF למערכת באמצעות גרירה או בחירה מהמחשב. לאחר ההעלאה, המערכת שומרת את הקובץ בצורה מאובטחת, מחלצת את הטקסט והמידע המרכזי מהמאמר, מנתחת אותו באמצעות בינה מלאכותית, ולאחר מכן הופכת אותו לזמין לניתוח, שיחה סוקרטית ודיון מעמיק.



זהו **דף הבית של המרצה (Instructor Dashboard)**, המציג תמונת מצב כללית על פעילות הסטודנטים במערכת. הדף כולל נתונים מרכזיים כגון מספר הסטודנטים, כמות המאמרים, סשנים שהושלמו וציון ממוצע, וכן פעולות מהירות לצפייה בהתקדמות הסטודנטים, ניתוח נתונים וניהול מאמרים, ובכך מאפשר למרצה מעקב, בקרה והכוונה פדגוגית יעילה.

A screenshot of a computer

AI-generated content may be incorrect.

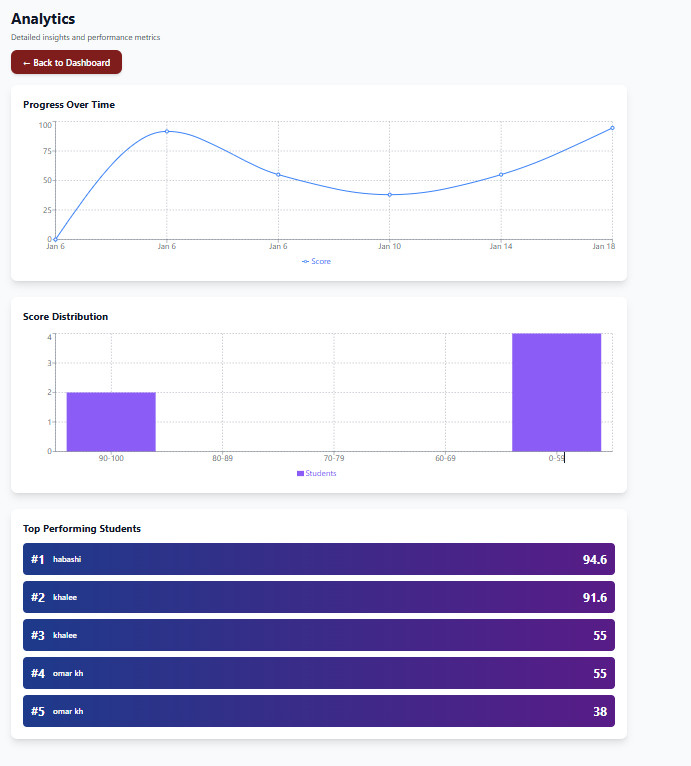
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

זהו **מסך ניהול ומעקב מרצה אחר סטודנטים**, שבו המרצה רואה רשימת כל הסטודנטים, מספר הסשנים והציון הממוצע של כל אחד. בלחיצה על סטודנט מוצג דוח אישי מפורט הכולל ציונים, רמות הבנה וחשיבה ביקורתית, חוזקות, נקודות לשיפור והמלצות מותאמות, וכן פירוט של כל סשן למידה עם ציונים לפי שאלה ורמת קושי, ובכך מתאפשר מעקב פדגוגי מעמיק והערכה מדויקת של התקדמות הסטודנט.



זהו **דף Analytics**, המציג למרצה ניתוחים וגרפים על ביצועי הסטודנטים לאורך זמן. הדף כולל מעקב אחר התקדמות הציונים, התפלגות ציונים בקבוצה, ורשימת הסטודנטים המובילים, וכך מאפשר זיהוי מגמות, פערים וחוזקות לצורך קבלת החלטות פדגוגיות מבוססות נתונים.

A screenshot of a computer

AI-generated content may be incorrect.

זהו **דף ניהול מאמרים (Article Management)** של המרצה, שבו ניתן לצפות בכל המאמרים שהועלו למערכת על-ידי הסטודנטים. הדף מאפשר חיפוש מאמרים, צפייה בפרטי ההעלאה, ומחיקה של מאמרים לפי צורך, ובכך מעניק למרצה שליטה ובקרה על התוכן האקדמי הזמין במערכת.

8. בשבוע של 21.1 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/1AAiXOpiXwto890SZs3tb5Pv2KITqEmgo/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.

# יש לכלול בתחילת מסמך הפרויקט את דף השער הבא: **טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

**הגשת פרויקט**

**<שם הפרויקט> < קוד הפרויקט><מס קבוצה>**

|  |  |
| --- | --- |
| **שם חבר.ת הצוות** | **תז** |
| חאלד חבשי | 212807879 |
| מוחמד חטאב | 325215846 |
| סאמויל זרייק | 212845259 |

**תקציר הפרויקט DeepReview –**

DeepReview היא מערכת Web חכמה המיועדת לסטודנטים וחוקרים המעוניינים לשפר את תהליך הקריאה וההבנה של מאמרים אקדמיים באמצעות כלי בינה מלאכותית מתקדמים. המערכת מאפשרת העלאת מאמרים בפורמט PDF, ניתוח תוכן אוטומטי, יצירת תקצירים חכמים, שאילת שאלות בצ’אט אינטראקטיבי והשוואה בין מאמרים שונים. בנוסף, המערכת כוללת בוט סוקרטי שמכוון את המשתמש ללמידה מעמיקה באמצעות שאלות הדרגתיות ומשוב חכם במקום תשובות ישירות.

**פונקציונליות לסטודנט:**  
הסטודנט יכול להעלות מאמרים לספרייה האישית, לצפות בתקצירים אוטומטיים, לנהל שיחות עם צ’אטבוט חכם המבוסס LLM על תוכן המאמר, ולהשתמש בבוט הסוקרטי ללמידה מונחית. כמו כן, קיימת אפשרות להשוות בין שני מאמרים לפי נושאים, שיטות ותוצאות מחקר, לעקוב אחר התקדמות הלמידה באמצעות גרפים וסטטיסטיקות, ולנהל פרופיל אישי הכולל היסטוריית פעילות ותובנות אישיות.

**פונקציונליות למרצה:**  
המרצה מקבל גישה לפאנל ניהול ייעודי המאפשר צפייה ברשימת הסטודנטים, ניתוח התקדמות הלמידה שלהם, בדיקת איכות התשובות, הפקת דוחות וסטטיסטיקות על רמת ההבנה והחשיבה הביקורתית, וכן ניהול מאמרים ונושאי לימוד עבור הקורס.

**מטרת המערכת** היא לשפר את חוויית הלמידה האקדמית, לקצר זמני קריאה, לחזק הבנה מעמיקה של מאמרים מדעיים ולעודד חשיבה ביקורתית באמצעות כלים מבוססי בינה מלאכותית

**מימוש – טכנולוגיות מרכזיות במערכת**

**Frontend:**  
המערכת פותחה באמצעות **Next.js (React)** תוך שימוש ב־**TypeScript** לבטיחות טיפוסים וכתיבת קוד יציב ומודולרי. מבנה הפרויקט מבוסס App Router של Next.js לניהול דפים וניווט דינמי.

**Styling עיצוב**  
העיצוב בוצע באמצעות **Tailwind CSS**, המאפשר בניית ממשק רספונסיבי, מודרני ואחיד, כולל תמיכה במצב כהה (Dark Mode) ואנימציות עדינות לחוויית משתמש משופרת.

**Backend:**  
צד השרת ממומש באמצעות **Server Actions ו־API Routes של Next.js**, המטפלים בהעלאת קבצים, ניתוח מאמרים, ניהול שיחות עם הבוט והפעלת הבוט הסוקרטי.  
לצורך עיבוד שפה טבעית נעשה שימוש ב־**Gemini API (LLM)** ליצירת תקצירים, תשובות חכמות ושאלות סוקרטיות.

**Database & Authentication:**  
המערכת עושה שימוש ב־**Supabase** כפתרון Backend-as-a-Service הכולל:  
• מסד נתונים PostgreSQL  
• מערכת Authentication לניהול משתמשים

Storage לאחסון קבצי PDF

**Hosting & Deployment:**  
האתר נפרס בענן באמצעות **Vercel**, המאפשר דיפלוימנט אוטומטי, אבטחה וסקיילביליות.

< <https://github.com/Mohamedkhattab02/DeepReview.git> >

< Deep-review-app.vercel.app>

**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:

