



**לכבוד : מרכז חרדי ירושלים 72235**

שלום רב,

הצעת הפרוייקט של הסטודנט **טירן עמנואל 203879952** במגמת הנדסת תוכנה-התמחות fullstack

Ayin Bright

אושרה כולל מספר תכניות שצורפו.

**חובה לצרף את ההצעה ואישור זה לספר הפרוייקט.**

בהצלחה

קדוש דרור , יועץ



---

# פרויקט גמר - הנדסאי תוכנה

תשפ"ג - תשפ"ד (2023-2024)

---



שם הפרויקט:



מגישים: עמנואל טירן | יהושע בלור

מנחה אקדמי: רונן שחר

קישור לפרויקט בגיטהאב <

### הצהרת הסטודנטים

אנו עמנואל טירן ת.ז. 203879952

יהושע בלוך ת.ז. 204019277

החתומים מטה, מצהירים בזאת כי כל פרויקט הגמר המוגש בספר זה, הינו פרי עבודתנו בלבד, על בסיס הנחייתו של המנחה ותוך הסתמכות על מקורות הידע והמידע האחרים המצויים בביבליוגרפיה המובאת בסיום ספר זה.

אנו מודעים לאחריות שהננו מקבלים על עצמנו ע"י חתימה על ההצהרה הזאת, שכל הנאמר בה הינה אמת ורק אמת.

---

חתימת התלמיד

---

חתימת התלמיד

אישור המנחה

הריני מאשר הגשת ספר פרויקט הגמר להערכה.

---

חתימת המנחה

## תוכן עניינים

4	1. מבוא לפרויקט
6	2. הצעת הפרויקט
18	3. פירוט דפי האתר
33	4. דפי האתר מותאם גם למובייל אישי
34	5. קוד הפרויקט
44	6. הסבר על מבנה הנתונים
49	7. סיכום ומסקנות

# מבוא

## סקירת פרוייקט:

הפרויקט שלנו הוא יישום אינטרנט חדשני שנבנה באמצעות Next.js, שנועד לעזור למשתמשים לשפר את ה"חזון" ( = ראייה ) שלהם באמצעות סדרה של תרגילים אינטראקטיביים, שגרות מותאמות אישית ומעקב אחר התקדמות. האפליקציה ממנפת את הכוח של רינדור בצד השרת ואינטראקטיביות בצד הלקוח כדי לספק חווית משתמש חלקה, מגיבה ומרתקת.

## קהל היעד:

- אנשים עם לקות ראייה קלה עד בינונית
- אנשים המעוניינים לשמור ולשפר את בריאות העיניים הכללית שלהם
- הורים המחפשים כלים לשיפור הראייה עבור ילדיהם בצורה מעניינת וחוויתית
- אנשים שאין להם את הכסף והזמן ללכת למכון כושר לראייה
- מי שלא רוצה לעשות לייזר לעיניים שלו מכל סיבה שהיא

## בעלי עניין פוטנציאליים:

- אופטומטריסטים ורופאי עיניים
- מרפאות טיפול בראייה
- ארגוני בריאות ואיכות חיים
- תוכניות בריאות ארגוניות

## תיאור המערכת:

### -בצד השרת :

- Next.js: מטפל בעיבוד צד השרת (SSR) ובנתיבי API:
- מסד נתונים: מאחסן פרופילי משתמשים, נתוני פעילות גופנית ומידע על התקדמות
  - מערכת אימות: מנהלת רישום משתמש, כניסה וטיפול בהפעלה

- מנוע יצירת תרגילים: יוצר שגרות אימונים מותאמות אישית המבוססות על נתוני משתמש
- מודול ניתוח התקדמות: מעבד נתוני ביצועי משתמש כדי להתאים את רמות הקושי
- הצפנת נתונים: מבטיחה אחסון ושידור מאובטח של מידע משתמש רגיש

### **תכונות עיקריות :**

- עיבוד בצד השרת לשיפור זמני טעינה ראשוניים ו - SEO-מסלולי API לטיפול בבקשות נתונים ועדכונים
- אימות משתמש מאובטח והרשאה
- יצירת תוכן דינמי המבוסס על פרופילי משתמשים והתקדמות

### **צד הלקוח כולל בתוכו:**

- רכיבי תגובה: בונה את ממשק המשתמש ומטפל באינטראקציות עם המשתמש –
- ניהול מצב: מנהל את מצב האפליקציה לחוויית משתמש חלקה
- מנוע עיבוד תרגיל: מציג תרגילי ראייה אינטראקטיביים
- עיצוב מגיב: מבטיח תאימות בין מכשירים וגדלים שונים של מסך תכונות עיקריות
- תרגילי ראייה אינטראקטיביים
- לוח מחוונים מותאם אישית המציג התקדמות ותרגילים מומלצים
- יכולת לא מקוונת לתרגילים נבחרים באמצעות עובדי שירות אינטגרציה ותקשורת

### **טכנולוגיות מפתח:**

- Next.js**-מגרת React לעיבוד וניתוב בצד השרת
- React**-ספריית JavaScript לבניית ממשקי משתמש
- Node.js**-זמן ריצה של JavaScript עבור לוגיקה בצד השרת
- מסד נתונים **MongoDB** לאחסון נתונים
- מודולי **CSS** וספריות ( framer-motion Animate TAILWIND ועוד )

## **הצעת הפרויקט**

## הצעה לפרויקט גמר

\* יש להדפיס את כל הנתונים הנדרשים

### א. פרטי הסטודנטים

שם הסטודנט	ת.ז. 9 ספרות	כתובת	טלפון נייד	תאריך סיום הלימודים
עמנואל טירן	203879952		0549631916	10.24
יהושוע בלוך	204019277		0587142427	10.24

שם המכללה: JBH - ירושלים

מסלול ההכשרה: הנדסאי תוכנה

מגמת לימוד: תוכנה מקום ביצוע הפרויקט: JBH - ירושלים

### ב. פרטי המנחה האישי

שם המנחה *	כתובת	טלפון נייד	תואר	מקום עבודה/תפקיד
רון שחר	רמת גן	054-3532852	MBA	

חתימת הסטודנטים

חתימת המנחה האישי

חתימת הגורם המקצועי מטעם מה"ט

### 1. שם הפרויקט

Ayin Bright – אתר לאבחון ושיפור הראייה



## 2. רקע

### 2.1. תיאור ורקע כללי

בעידן הדיגיטלי של היום, אנשים רבים חווים אי נוחות חזותית והידרדרות עקב חשיפה ממושכת למסכים ואורח חיים בישיבה VisionEnhance. נוצרה במטרה להעצים את המשתמשים לקחת אחריות על בריאות העיניים שלהם באמצעות תוכן מרתק ואינפורמטיבי.

### 2.2. מטרות המערכת

קדם תרגילי עיניים - מציע מגוון תרגילי עיניים אינטראקטיביים שמטרתם לשפר את המיקוד, הגמישות ובריאות העין הכללית. עודדו את המשתמשים לשלב את התרגילים האלה בשגרה היומיומית שלהם.

להפחית את עומס העיניים: - לשתף טיפים וטכניקות, למזעור עומס עיניים דיגיטלי שנגרם כתוצאה מזמן מסך ממושך, כגון כלל 20-20-20. - לספק הדרכה לגבי ארגונומיה נכונה לשימוש במחשב ובמכשיר נייד.

לתמוך בהרגלים בריאים: - ללמד את המשתמשים על השפעת התזונה על הראייה והצע שינויים תזונתיים לתמיכה בבריאות העיניים. - לקדם שינויים באורח החיים שיכולים לשפר את חדות הראייה, כגון פעילות גופנית סדירה ושינה מספקת.

## 3. סקירת מצב קיים בשוק

שוק שיפור הראייה וטיפול העיניים מתפתח במהירות, מונע על ידי צרכי צרכנים משתנים, התקדמות טכנולוגית ודגש הולך וגובר על בריאות מונעת. האתר שלנו יכול למצב את עצמו כמשאב בעל ערך בנוף זה על ידי הצעת פתרונות חדשניים, תוכן חינוכי כדי לענות על הדרישות של צרכנים מודעים לבריאות.

## 4. מה הפרויקט אמור לחדש או לשפר

- לפתח תוכניות מותאמות המבוססות על הערכות אישיות, העדפות ומעקב התקדמות. זה יכול לכלול משטרי פעילות לעיניים מותאמים אישית ותוכן מעשי שמסתגל לאורך זמן ככל שהמשתמשים משתפרים.

❓ ליצור תרגילים מגניבים וכלים אינטראקטיביים שהופכים את שיפור הראייה למהנה ומרתק. זה יכול לכלול

אתגרים, שאלונים ותגמולים עבור השלמת תרגילים או מודולים.

## 5. דרישות מערכת ופונקציונאליות

### 5.1. דרישות מערכת

- סביבת הטמעה ושימוש: מערכת מבוססת אינטרנט שתהיה נגישה מכל מכשיר עם חיבור לאינטרנט.
- שרידות וגיבוי נתונים: המערכת תתוכנן כך שתוכל להתמודד עם מספר משתמשים בו זמנית. נתוני המערכת יגובו על בסיס יומי כדי להבטיח שמירה ושחזור של מידע במקרה של תקלה
- ביצועים והתמודדות עם עומסים: המערכת תתוכנן עם דגש על ביצועים גבוהים וזמן תגובה מהיר

### 5.2. דרישות פונקציונאליות.

רישום משתמשים ומעקב:

- משתמשים צריכים להיות מסוגלים ליצור חשבונות, לנהל פרופילים ולעקוב אחר ההתקדמות שלהם לאורך זמן.
- ספריית תרגילים אינטראקטיביים עם הוראות ברורות ועזרים ויזואליים.

## 6. בעיות צפויות במהלך הפיתוח ופתרונות

### 6.1 תיאור הבעיות

-בעיה: האתר עשוי להיתקל בזמני טעינה איטיים, במיוחד עם תוכן כבד או תנועה רבה

. - פתרון: ביצוע אופטימיזציה של תמונות ונכסים, יישום טעינה עצלנית ושימוש ב-CDN כדי לשפר את מהירות האתר. עקוב באופן קבוע אחר מדדי ביצועים והתאם לפי הצורך.

-בעיה: טיפול בנתוני משתמש רגישים עלול להעלות בעיות אבטחה ופרטיות

- פתרון: הטמעת שיטות הצפנה חזקות, פתרונות אחסון נתונים מאובטחים ועמידה בתקנים הרלוונטיים להגנה על נתונים (למשל, GDPR). בצע בקביעות ביקורות ועדכוני אבטחה.

פתרון טכנולוגי נבחר:

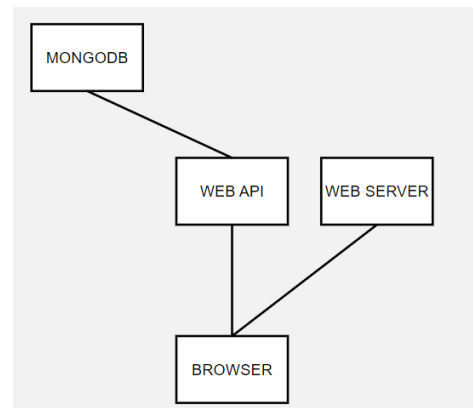
## 7.1 טופולוגית הפתרון

המערכת תיושם ברשת האינטרנט וכוללת את המרכיבים הבאים:

- שרת אינטרנט: לאחסון ולהרצת האפליקציה.
- ממשק משתמש בצד הלקוח: הנגיש מדפדפנים מודרניים ומותאם למכשירים שונים.
- מסד נתונים לאחסון מידע על משתמשים, תרגילים ומעקב.
- חיבור אינטרנטי: לשם גישה למערכת מכל מקום.

## 7.3 שפות הפיתוח

- React, next.js, JavaScript: לשם פיתוח בצד הלקוח והשרת בשל פופולריות, ביצועים וגמישות.
- HTML לשם בניית מבנה התוכן והשלד של הדפים באתר, תוך שמירה על תקני אינטרנט נגישים ומותאמים לכלל הדפדפנים.
- CSS לעיצוב ממשק המשתמש בשל יכולתו להקל על תחזוקת הקוד ולספק אפשרויות עיצוב מתקדמות.
- MONGODB לשם ניהול ואחסון מידע במסד הנתונים הרלציוני.



## 7.4 תיאור הארכיטקטורה הנבחרת

הארכיטקטורה היא Microservices המאפשרת חלוקה למודולים עצמאיים לשם פיתוח, תחזוקה ועדכון קלים יותר. כל שירות יתפקד באופן עצמאי, מה שמקל על שדרוגים ותיקונים מבלי להשפיע על המערכת כולה.

## 7.5 חלוקה לתכניות ומודולים

- ממשק משתמש: כולל את כל הממשקים הוויזואליים והשירותים הקשורים אליו.

- **API** של שרת: כולל את כל ה Endpoints-ליצירת, קריאה, עדכון ומחיקת נתונים.
- מסד נתונים: כולל את כל המודולים הקשורים לניהול הנתונים במערכת.

## 7.6 סביבת השרת

שירות ענן: הסביבה תופעל על שרתים וירטואליים בענן, כגון AWS או Azure לשם גמישות וסקלabilיות. השימוש בשירותי ענן יאפשר גמישות בהתאמה לעומסים משתנים והבטחת זמינות גבוהה של השירות.

## 7.7 ממשק המשתמש

הממשק יכולול אפשרויות שונים, מידע על הלקוח ומעקב באוסף התרגילים שהאתר נודע אליו.

מבוסס על React לשם פיתוח ממשק המשתמש בצד הלקוח, מאפשר פיתוח דינמי ואינטראקטיבי עם אפשרויות לשימוש חוזר בקומפוננטות ולתחזוקה קלה של הקוד.

## 7.8 ממשקים למערכות אחרות API

חיבור לAPI חיצוני להטמעת השעה הרלוונטית של המשתמש.

## 7.9 שימוש בחבילות תוכנה

JWT לאימות ואבטחת משתמשים באמצעות עוגיות מוצפנות.

## 8. שימוש במבני נתונים וארגון קבצים

### 8.1 מבני נתונים

- משתמש (**User**): שם משתמש, כתובת דוא"ל, סיסמא (מוצפנת), חולשות של הראייה, - שדה מזהה ייחודי (ID), ראייה, טשטוש, ועיוורון צבעים

### 8.2 שיטת האיחסון

- מסד נתונים MONGODB :
  - איחסון בענן:
- התכנון כולל שימוש באחסון בענן לצורך גיבוי ושמירת נתונים קריטיים, כגון גיבוי מסד הנתונים, ומידע רגיש אחר.

### 8.3 מנגנוני התאוששות מנפילה או קריסה

התאוששות מנפילה/קריסה: גיבוי תקופתי יומי של מסד הנתונים ושמירתם במיקום מאובטח בענן.

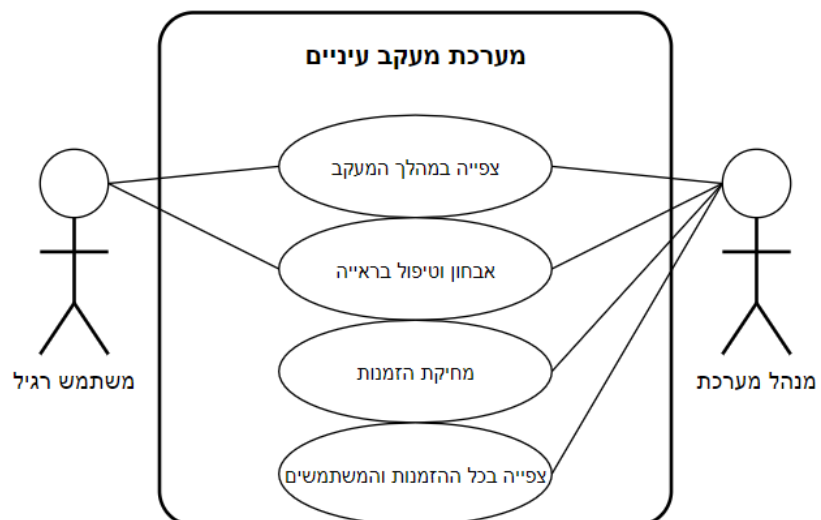
תהליך שחזור מתועד.

## 9. רשימת מערכת מרכזיים

### 9.1 תרשים מקרה שימוש (Use Case Diagram)

התרשים מתאר את האינטראקציות בין המשתמשים לבין המערכת ואת ההרשאות השונות:

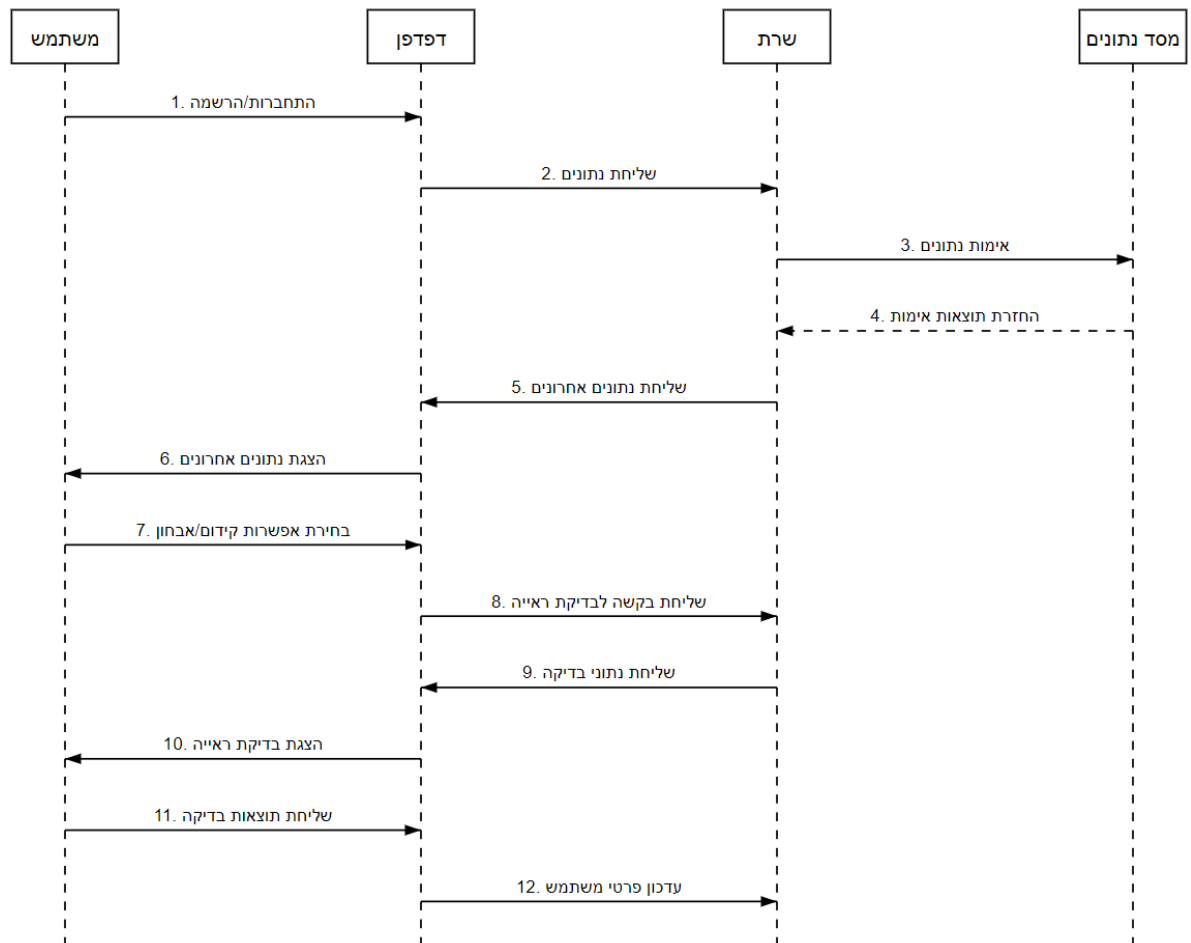
- משתמש רגיל: יכול לצפות בכל מהלך המעקב שעליו, לאבחן לשפר ולתרגל את הראייה שלו.
- מנהל מערכת: יכול לבצע את כל הפעולות של המשתמש הרגיל והקבלה, בנוסף למחיקת הזמנות, וצפייה ברשימת כל ההזמנות וכל המשתמשים.



## 9.2 תרשים רצף (Sequence Diagram)

תרשים רצף קריאות פונקציות מרכזיות בלוגיקה העיסוקית המרכזית של הפרוייקט מתאר את רצף הקריאות בפונקציות שונות בזמן ביצוע פעולות עיקריות כמו ביצוע הזמנה חדשה:

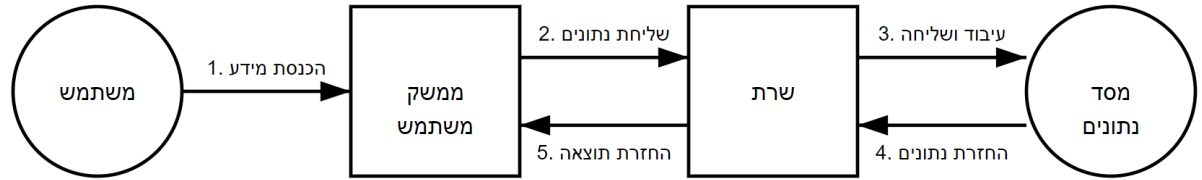
1. משתמש נרשם או מתחבר לאתר.
2. השרת מקבל את הבקשה ומבצע validation של הנתונים.
3. האתר מוביל את המשתמש לנתונים האחרונים שלו.
4. משם הוא בוחר את האפשרות שבו הוא רוצה לקדם או לאבחן.
5. המשתמש חווה גלישה מרהיבה ובודק את הראייה שלו.
6. אם יש למשתמש טעות בראייה – ניזון בערכת ומעדכן את הפרטים של המשתמש.



## 9.3 תרשים זרימת נתונים (Data Flow Diagram)

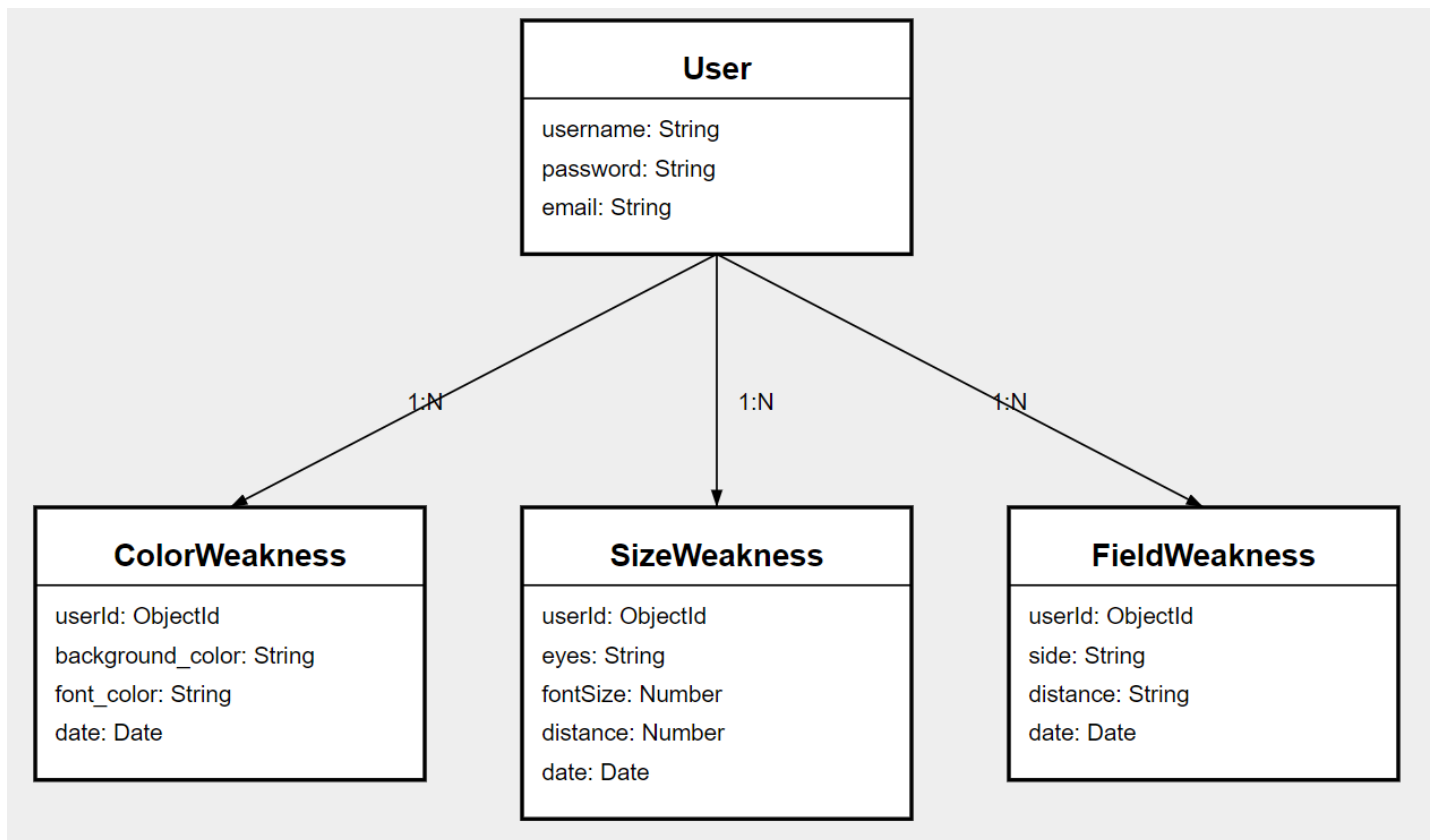
תרשים זרימת הנתונים מציג את הזרימה של נתונים בתוך המערכת:

1. משתמש מכניס מידע לממשק המשתמש (Client-Side).
2. הנתונים נשלחים לשרת (Server-Side).
3. השרת מבצע עיבוד נתונים ושולח אותם למסד הנתונים.
4. מסד הנתונים מחזיר את הנתונים המבוקשים לשרת.
5. השרת מעבד את הנתונים ומחזיר את התוצאה לממשק המשתמש.



## 9.4 Entity-Relationship Diagram

- תרשים יחסי הגומלין והקשרים בין הטבלאות והיישויות השונות במערכת:



## 10 תיאור המרכיב האלגוריתמי – חישובי

### 1. רישום משתמש ואימות

-בעיה: משתמשים צריכים ליצור חשבונות ולגשת לנתונים שלהם בצורה מאובטחת.  
-פתרון: הטמעת אלגוריתמים לרישום משתמש המאמתים סיסמאות קלט ו hash-השתמשות באלגוריתמי אימות כדי לאמת אישורים במהלך הכניסה.

### 2. אחסון וניהול נתונים

-בעיה: אחסון ואחזור ביעילות נתוני משתמש, כולל חולשות ראייה.  
-פתרון: השתמשות בפעולות CRUD לניהול נתוני משתמש במסד נתונים. ליצור קשרים בין משתמשים וחולשותיהם באמצעות מפתחות זרים.

### 3. הערכת אבחון ראייה

-בעיה: הערכת חולשות הראייה של המשתמשים על סמך הקלט שלהם.  
-פתרון: פיתוח אלגוריתמי אבחון המנתחים את תגובות המשתמשים לבדיקות ראייה.

### 4. המלצות מותאמות אישית

-בעיה: סיפוק תרגילים מותאמים כדי לעזור למשתמשים לשפר את הראייה שלהם.  
-פתרון: יצירת מנוע המלצות שיוצר תרגילים מותאמים אישית המבוססים על נתוני משתמשים ומשוב, תוך חידוד הצעות לאורך זמן.

### 6. ממשק משתמש וניסיון

-בעיה: הקפדה על ממשק אינטואיטיבי וידידותי למשתמש.  
-פתרון: הטמעת אלגוריתמים של מעקב אינטראקציות לאיסוף ניתוחים וטעינה דינמית של תוכן בהתבסס על העדפות המשתמש לחוויה מותאמת אישית.

### 7. דיווח וניתוח

-בעיה: סיפוק תובנות לגבי בריאות הראייה וההתקדמות של המשתמשים.  
-פתרון: פיתוח אלגוריתמים לניתוח נתונים כדי לעקוב אחר מגמות לאורך זמן ולהפיק טבלאות המשקפים את ההתקדמות של המשתמשים ותחומים לשיפור.

## סיכום

האתר כולל אלגוריתמים לניהול משתמשים, טיפול בנתונים, הערכת חזון, המלצות מותאמות אישית, עיצוב ממשק משתמש, אמצעי אבטחה ודיווח. כל הרכיבים עובדים יחד כדי לספק פלטפורמה מקיפה ומאובטחת לאבחון ושיפור הראייה.

## 11. תיאור/התייחסות לנושאי אבטחת מידע

- גישה לחשבונות משתמשים על ידי שימוש בשם משתמש וסיסמא



- גישה לחשבון מנהל על ידי חלוקה לרמות הרשאה
- הגנה מפני מתקפות MONGO Injection בעזרת מסנן קלט משתמש.
- שמירת נתוני משתמש מוצפנים בטוקן בתוך עוגיות בדפדפן
- הענקת תוקף קצר לטוקנים
- הצפנת סיסמאות בבסיס הנתונים

## 12. משאבים הנדרשים לפרויקט:

### 1. שעות עבודה:

- תכנון כ-10 שעות
- פיתוח אתר: כ-200 שעות
- עיצוב ממשק משתמש: כ-40 שעות
- בדיקות ותחזוקה: כ-60 שעות

### • ציוד נדרש:

- מחשבי פיתוח עם חיבור אינטרנט
- שרת לפיתוח והטמעה

### 3. תוכנות נדרשות:

- Visual Studio Code
- Node.js, React, NEXT.JS
- Mongo
- github desktop

### 4. ידע נדרש ללימוד:

- אבטחת מידע באתרי אינטרנט
- ספריית MUI
- ספריית TAILWIND
- ספריית framer-motion

### 5. ספרות ומקורות מידע:

- GOOGLE
- דוקומנטציה של MUI
- STACK OVERFLOW

## 13. תכנית עבודה ושלבים למימוש הפרויקט

## **שלב 1: ניתוח דרישות ותכנון מפורט**

- איסוף דרישות מהלקוח
- ניתוח דרישות והגדרת מקרי שימוש - Use Cases
- תכנון מפורט של ממשק המשתמש, מסכים ותזרים עבודה
- תכנון מפורט של מבנה מסד הנתונים
- תכנון ארכיטקטורת המערכת
- תכנון חלוקת עבודה בצוות ולו"ז מפורט

## **שלב 2: פיתוח ראשוני**

- הקמה ראשונית של השרתים וסביבת הפיתוח
- פיתוח מוקדם של מסך כניסה והרשמה
- יצירת מסד נתונים ראשוני עם טבלאות בסיסיות
- פיתוח בסיסי של ממשק ניהול תוכן פשוט

## **שלב 3: פיתוח מלא של מודולים עיקריים**

- פיתוח מלא של מודול ניהול משתמשים
- פיתוח מלא של מודול ניהול חולשות הראייה
- השלמת מסד נתונים עם כל הטבלאות והקשרים

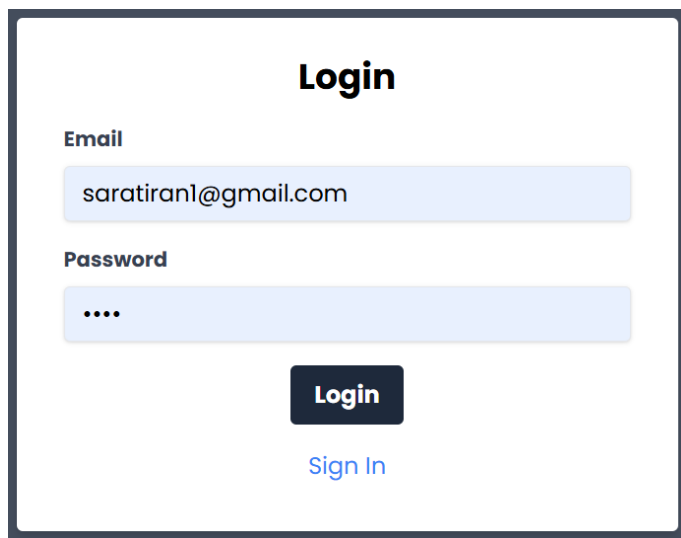
## **שלב 4: פיתוח ממשק משתמש מלא**

- עיצוב גרפי מלא של כל המסכים
- יישום רספונסיביות מלאה
- פיתוח ממשק משתמש מלא עם קישור למודולים
- בדיקות ממשק משתמש ותיקון באגים

## **שלב 5: בדיקות ושליפת באגים**

- בדיקות יחידה ואינטגרציה של כל המודולים
- בדיקות מערכת כולל מקרי קצה
- תיקון באגים וייצוב המערכת
- בדיקות ביצועים ועומסים

## פירוט דפי האתר



A login form titled "Login" with a dark blue border. It contains two input fields: "Email" with the text "saratiran1@gmail.com" and "Password" with four dots. Below the fields is a dark blue "Login" button and a blue "Sign In" link.

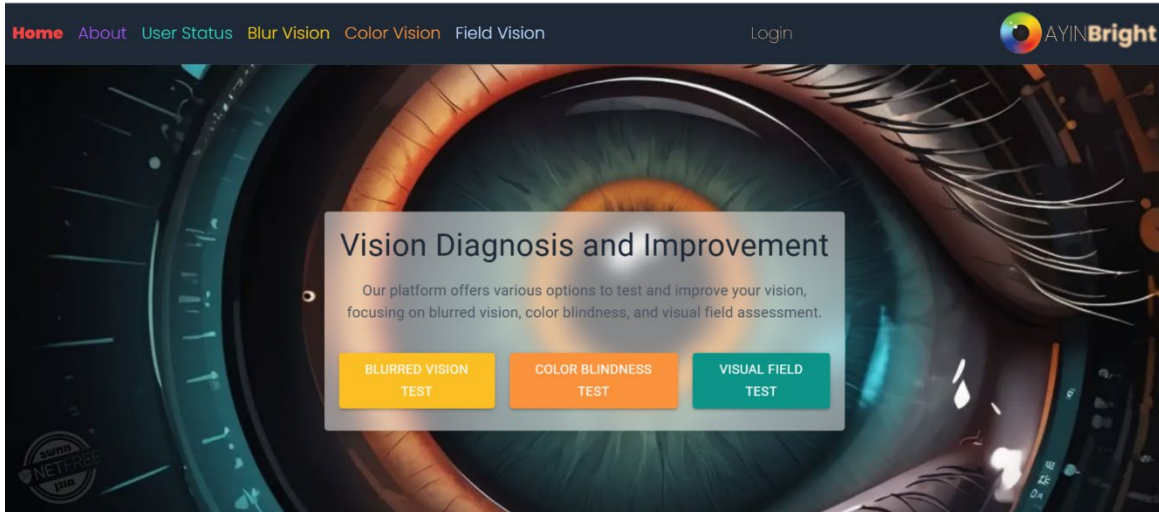
### תכונות דף הכניסה:

- מערכת אימות משתמש מאובטחת
- שדות דוא"ל/שם משתמש וסיסמה
- הצפנת נתונים לאבטחת המשתמש

### מִטְרָה:

- חווית משתמש מותאמת אישית
- גישה מאובטחת לכלי הערכת ראייה
- עקוב אחר ההתקדמות האישית
- אחסן היסטוריית בדיקות ראייה

## אלמנטים עיצוביים של דף הבית:



- מבנה ניווט אינטואיטיבי עם היררכיה ברורה
- ערכת צבעים אסטרטגית (טורקיז, כתום, צהוב) המשקפת מקצועיות רפואית
- תמונת גיבור שובת עין המדגישה מיקוד טיפול בראייה
- פריסה רספונסיבית המבטיחה תאימות לנייד

### תכונות עיקריות:

- גישה ישירה ל-3 מבחני ראייה ראשוניים
- נקה כפתורי קריאה לפעולה
- ממשק ידידותי למשתמש
- עיצוב נגיש למשתמשים לקויי ראייה

### חוויית משתמש:

- נתיב ניווט פשוט לפונקציות הליבה
- הבנה מיידית של מטרת האתר
- גישה מהירה לבדיקות ראייה חיוניות
- אווירה מקצועית אך מסבירת פנים

### יישום טכני:

- מסגרת עיצוב רספונסיבית
- טעינת תמונה אופטימלית

- תאימות בין דפדפנים
- עמידה בתקני נגישות

## ניהול פרופיל משתמש:

The screenshot shows a web application interface for user management. The browser address bar indicates the URL is localhost:3000/user. The application has a dark blue header with navigation links: Home, About, UserStatus, Blur Vision, Color Vision, Field Vision, Admin, and a Logout button. The main content area is titled 'User Details' and contains the following information:

- Username:** sara tiran
- Email:** saratiran1@gmail.com
- Color Weaknesses:**
  - Background: purple
  - Font: green
  - Date: 28.8.2024

Below this, there is another section with a background color of blue and a font color of orange.

- תצוגה מקיפה של מידע אישי
- אחסון מאובטח של נתוני זיהוי משתמש
- גישה נוחה לתוצאות בדיקה היסטוריות
- תצוגה מרוכזת של מצב בריאות הראייה

### תיעוד היסטוריית בדיקות:

- מעקב כרונולוגי של כל הערכות הראייה
- רישומים מפורטים של:
  - תוצאות בדיקת ראיית צבע
  - מדידות ראיית טשטוש
  - תוצאות בדיקת שדה ראייה
- ערכים עם חותמת תאריך למעקב אחר התקדמות

### ארגון נתונים:

- הצגה מובנית של נתוני בריאות הראייה

- סיווג ברור של סוגי בדיקות שונים
- פורמט קל לקריאה הן למשתמשים והן לספקי שירותי בריאות
- הדמיית מגמה היסטורית

#### **תכונות פרטיות ואבטחה:**

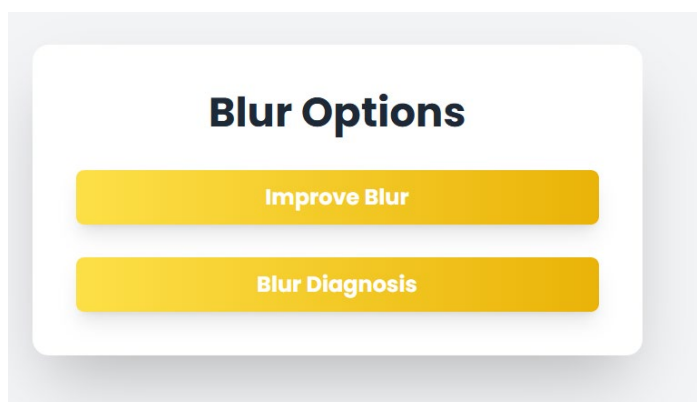
- תצוגת מידע אישי מוגן
- תאימות לאחסון נתונים מאובטח
- גישה מבוקרת לנתונים רפואיים רגישים

#### **יתרונות חווית משתמש:**

- גישה פשוטה להיסטוריה מלאה של בריאות הראייה
- מעקב קל אחר שינויים בראייה לאורך זמן
- סקירה מקיפה של מצב הראייה
- מעצים את המשתמשים במידע הבריאותי שלהם

## מודול בדיקת ראייה מטושטשת:

-גישת בדיקת מסך כפול לבדיקה ושיפור הראייה מקיפה  
-תהליך בחינה אינטראקטיבי מונחית משתמשים



### מסך 1 - מבחן זיהוי אותיות:

- הנחיות ברורות לבדיקת עין אחת
- מרחק בדיקה סטנדרטי (1 מטר)
- מערכת זיהוי אותיות מרובות בחירה
- פרוטוקולי סביבת בדיקה מבוקרים
- הערכת זיהוי תווים מדויקת

Cover your left eye  
And examine your right eye  
Tap the button corresponding to the character, 1 meter away

כ

א

צ

ש

כ

תוצאות בדיקת הראייה :

**The result of the test of  
the right eye**

הגודל המינימלי שבו הצלחת לראות הוא:

**14.6mm**

Try left eye





- מנגנון התאמת גודל גופן מדויק (נמדד במ"מ)
- יכולות שינוי גודל בזמן אמת
- מערכת זיהוי מילים אינטראקטיבית
- ממשק שליטה ידידותי למשתמש
- פונקציונליות של דיווח על שגיאות

#### מאפיינים טכניים:

- התאמת גודל גופן דינמית (דיוק 0.1 מ"מ)
- מסד נתונים של מילים מתחלפות
- אפשרויות בדיקה ספציפיות לעיניים
- מערכת מעקב ודיווח על שגיאות
- מנגנון משוב מיידי

כללים לבדיקת ראייה עם תרשים סגל  
באורך 1 מטר: הגדרה: - הקפידו על  
תאורה טובה ועקבית בחדר. - התקן את  
התרשים בגובה העיניים עבור האדם  
הנבדק. - מדוד וסמן בדיוק מטר אחד  
מהתרשים כדי שהנבדק יעמוד. נוהל  
בדיקה: - בדוק כל עין בנפרד. כסה את  
העין שאינה בודקת עם אטם או מדבקה.  
- התחל עם עין ימין, ולאחר מכן בדוק את  
עין שמאל. - בקשו מהנבדק לקרוא את  
הקו הקטן ביותר שהוא יכול לראות  
בבירור. ניקוד: - רשמו את השורה  
הקטנה ביותר שהנבדק יכול לקרוא  
במדויק (לפחות 3 מתוך 5 אותיות

- הוראות שימוש ברורות
- הודעות אזהרה לבדיקה תקינה
- תזכורות לחילופי עיניים
- מנגנוני מניעת שגיאות
- סביבת בדיקה נוחה

### איסוף נתונים:

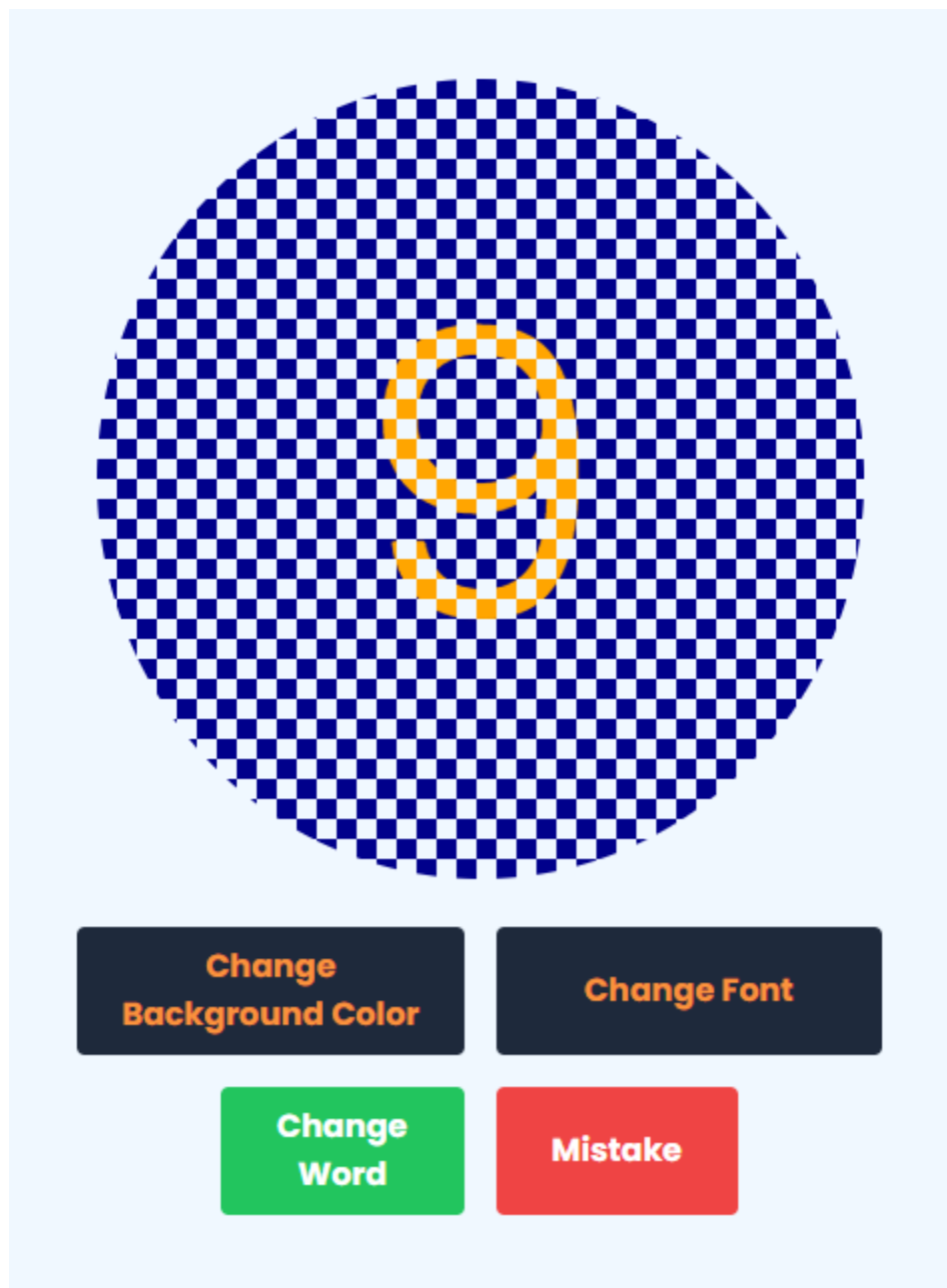
- מתעד גודל גופן מינימלי קריא
- עוקב אחר דיוק זיהוי
- מסמכים התקדמות בדיקות
- מאחסן נתוני ביצועים היסטוריים
- מאפשר ניתוח מגמות

### מתודולוגיית בדיקה:

- מבוסס על עקרונות אופטומטריים סטנדרטיים
- תנאי בדיקה מבוקרים
- התקדמות קושי רציף
- הערכת ראייה דו צדדית
- תיעוד תוצאות מקיף

## בדיקת ראיית צבע מבוססת אישיהרה

(רופא עיניים יפני, פיתח את האבחון ללקות ראייה בצבעים) :



- התאמה דיגיטלית למבחן אישיהרה הסטנדרטי
- בדיקת עיוורון צבעים ברמה מקצועית
- סביבת בדיקה אינטראקטיבית
- פרמטרי בדיקה הניתנים להתאמה אישית

### מבחן אלמנטים עיצוביים:

- כיול צבעים ברמת דיוק גבוהה
- דפוסי צבע מאומתים מדעית
- תצוגת מספר/אות דינמית
- רקעים נשלטי ניגודיות
- תנאי בדיקה מתוקננים

### יישום טכני:

- עיבוד דיגיטלי מדויק בצבע
- אקראית של דפוסי רקע
- אפשרויות וריאציה בסגנון גופן
- מערכת מעקב אחר שגיאות
- עיבוד תוצאות בזמן אמת

### יכולות בדיקה:

- איתור ליקויים שונים בראיית צבע:
- פרוטנופיה (עיוורון אדום)
- Deuteranopia (עיוורון ירוק)
- Tritanopia (עיוורון כחול)
- הערכת חומרה
- הערכת זיהוי דפוסים

### תכונות אינטראקטיביות:

- שינוי צבע רקע
- התאמה אישית של סגנון גופן
- שינויים בתוכן דינמי
- מערכת דיווח על שגיאות
- מנגנון משוב מיידי

### יישומים קליניים:

- הקרנת ראיית צבע מקצועית
- זיהוי מוקדם של ליקויים בצבע
- מעקב אחר התקדמות
- הערכת יעילות הטיפול
- תיעוד לתיעוד רפואי

### חווית משתמש:

- ממשק בדיקה אינטואיטיבי
- הוראות ברורות
- תנאי צפייה נוחים
- פרשנות קלה לתוצאה
- יכולת מעקב התקדמות

### אבטחת איכות:

- בדיקות כיול סדירות
- אימות דיוק הצבע
- אופטימיזציה של רזולוציית התצוגה
- תאימות בין מכשירים
- בדיקת בקורות סביבתיות

## תיעוד בדיקת שדה ויזואלי



### מטרה וסקירה

- מבחן השדה החזותי הוא מרכיב מכריע בחבילת הערכת הראייה של AYINBRIGHT, המיועדת ל:
- הערכת מידת הראייה ההיקפית של המשתמש
  - זיהוי נקודות עיוורות פוטנציאליות או ליקויי ראייה
  - לספק סקר ראשוני למצבים שונים הקשורים לראייה
  - אפשר ניטור עצמי של שינויים בראייה לאורך זמן

### יישום טכני

- מתודולוגיית בדיקה:
- מערכת תצוגת תווים דינמית
  - יצירת מספרים/אותיות אקראית
  - מיפוי מיקום משתנה
  - מערכת ניווט ותגובה נשלטת על ידי המשתמש

## רכיבי ממשק משתמש

### בקורות ניווט:

- בקורות תנועה לרוחב (שמאל/ימין)
- פונקציונליות בחירת צד
- אפשרות לשינוי תווים
- יכולת דיווח על שגיאות

### אלמנטים חזותיים:

- תצוגה בעלת ניגודיות גבוהה
- טיפוגרפיה ברורה וקריאה
- מיקום כפתורים אינטואיטיבי
- עיצוב ממוקד נגישות

### נוהל בדיקה

- תהליך שלב אחר שלב:
- א) תצוגת הוראות משתמש ראשונית
- ב) הצגת אופי בראייה מרכזית
- ג) בדיקות היקפיות פרוגרסיביות
- ד) איסוף תגובות משתמשים
- ה) אוסף תוצאות

תיעוד זה מספק סקירה מקיפה של רכיב בדיקת השדה החזותי תוך שמירה על התמקדות בתועלת הקלינית ובעיצוב ממוקד המשתמש.

## תיעוד פאנל WEBMASTER

Search by username or email				
Username	Email	Color Weaknesses	Size Weaknesses	Field Weaknesses
Bob	bob@example.com	NO COLOR WEAKNESSES	<div>HIDE SIZE WEAKNESSES</div> <div>Size Weakness 1 ^</div> <div>Font Size: 12</div> <div>Eye:</div> <div>Distance: 18</div> <div>Date: 11.11.2024</div> <div>DELETE</div>	NO FIELD WEAKNESSES
			Size Weakness 2 v	
Bob	bob@example.com	SHOW COLOR WEAKNESSES	SHOW SIZE WEAKNESSES	NO FIELD WEAKNESSES
Cllice	cllice@example.com	SHOW COLOR WEAKNESSES	SHOW SIZE WEAKNESSES	NO FIELD WEAKNESSES
Alice	alice@example.com	NO COLOR WEAKNESSES	SHOW SIZE WEAKNESSES	NO FIELD WEAKNESSES
Dob	dob@example.com	NO COLOR WEAKNESSES	SHOW SIZE WEAKNESSES	NO FIELD WEAKNESSES

Show More

### 1. תכונות ניהוליות

- ניהול מסדי נתונים של משתמשים
- מעקב אחר נתוני ראייה מקיף
- פונקציונליות חיפוש בזמן אמת
- מערכת תצוגת משתמש ניתנת להרחבה

### 2. יכולות ניהול נתונים

- קטגוריות מעקב:
- ליקויים בראיית צבע
- בעיות בתפיסת גודל
- מגבלות שדה ראייה
- שמירת נתונים היסטוריים



### מערכת חיפוש וסינון

- חיפוש שם משתמש/מייל
- סינון תוצאות דינמי
- גישה מיידית לנתונים
- מעקב אחר היסטוריית משתמשים

### קטגוריות חולשה ניטור

- כל רשומת משתמש מציגה:
- מצב ראיית צבע
- נתוני תפיסת גודל
- תוצאות ראיית שדה
- דוחות חולשה מפורטים

### הצגת נתונים

- אלמנטים אינטראקטיביים:
- פרטי חולשה הניתנים להרחבה
- מחק פונקציונליות
- מעקב היסטורי
- ערכים עם חותמת תאריך

### תכונות אבטחה

- פרטי משתמש מוגן
- טיפול מאובטח בנתונים
- בקרת כניסה
- תאימות לפרטיות

### כלים לניהול משתמשים

- מעקב אחר משתמשים בודדים
- מעקב אחר התקדמות
- אפשרויות שינוי נתונים
- טיפול בנתונים בכמות גדולה

1. יישומים מקצועיים

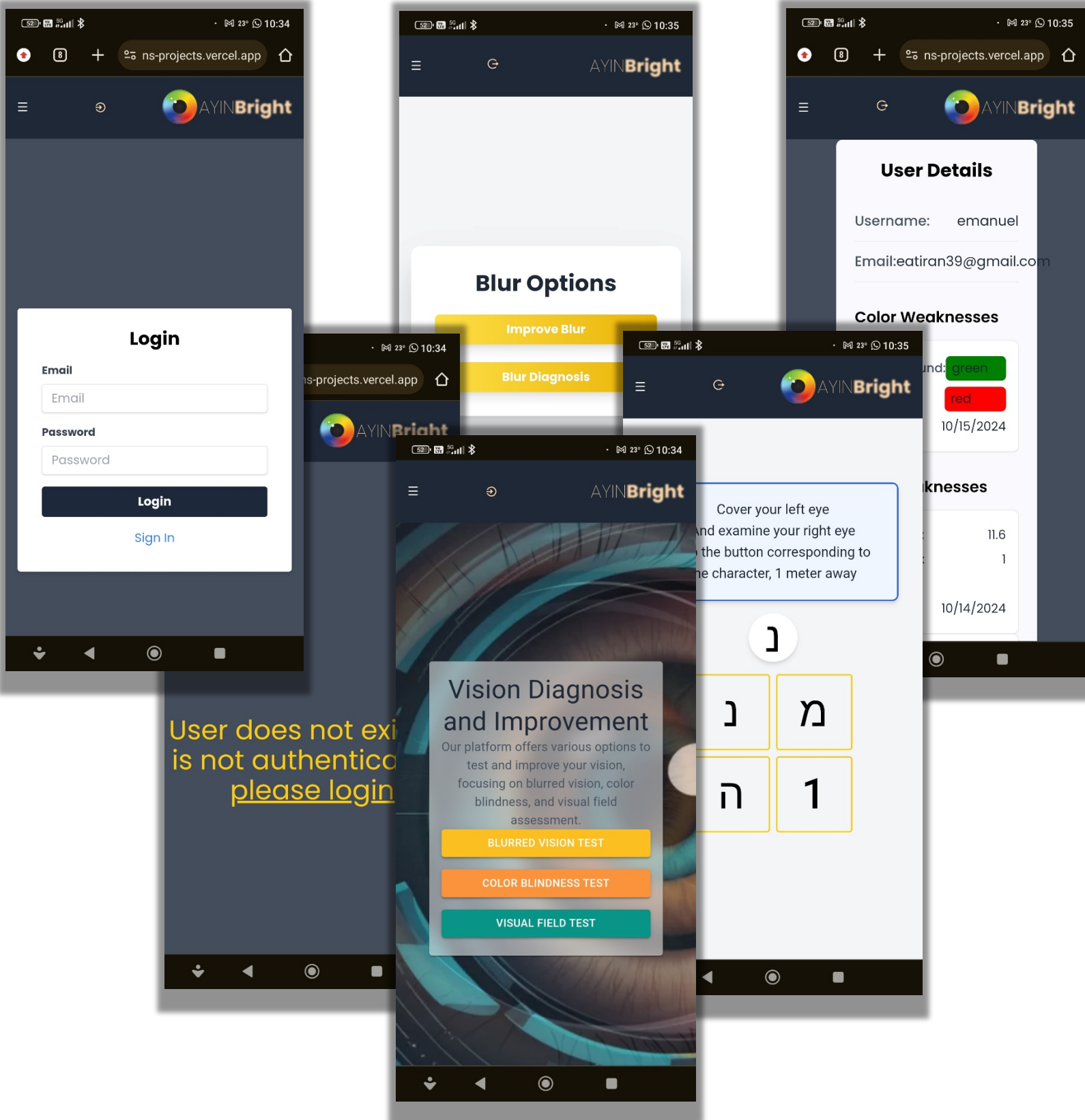
- מעקב אחר התקדמות המטופל
- תכנון טיפול
- ניתוח סטטיסטי
- זיהוי מגמה

2. יכולות דיווח

- דוחות משתמשים בודדים
- מעקב אחר התקדמות
- אפשרויות ייצוא נתונים

תיעוד זה מתאר את יכולות הניהול המקיפות תוך שמירה על התמקדות בתועלת קלינית ואבטחת מידע.

## דפי האתר מותאם גם למובייל אישי



## קוד הפרוייקט

### הקוד לדף הבית

```
import { connectToMongo } from '@server/connectToMongo'

import CurrentTime from '@components/CurrentTime'
import ImageCarousel from '@components/ImageCarousel'
import SighnIn from '@components/SighnIn'
import VisionImprovementHub from '@components/VisionImprovementHub'
import { authAction } from '@server/BL/actions/login.action'
import Login from '../components/Login'

Tabnine | Edit | Test | Explain | Document | Ask
export default async function Home() {

  await connectToMongo()

  const isUser = await authAction();

  return (
    <div>
      <ImageCarousel >
        <VisionImprovementHub />
      </ImageCarousel >
      {isUser ? (
        <Login />
      ) : (
        <SighnIn />
      )}
      <CurrentTime />
    </div>
  )
}
```

## הקוד להתחברות והרשמה

```
export default function Login() {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');

  const handleSubmit = async (event) => {
    event.preventDefault();
    const formData = new FormData(event.target);
    const response = await loginAction(formData);

    if (response.success) {
      setError('');
      let user = "Emanuel tiran";
      window.location.href = `/user`;
    } else {
      setError(response.message);
      alert(response.message);
    }
  };

  return (
    <div className="flex justify-center items-center min-h-screen p-4">
      <form className="w-full max-w-md" onSubmit={handleSubmit}>
        <div className="bg-white shadow-md rounded px-8 pt-6 pb-8 space-y-4">
          <h1 className="text-2xl font-bold mb-4 text-center">Login</h1>
          <div>
            <label className="block text-gray-700 text-sm font-bold mb-2" htmlFor="email">Email</label>
            <input
              name="email"
              className="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
              id="email"
              type="email"
              placeholder="Email"
              value={email}
              onChange={(e) => setEmail(e.target.value)}
              required
            />
          </div>
          <div>
            <label className="block text-gray-700 text-sm font-bold mb-2" htmlFor="password">Password</label>
            <input
              name="password"
              className="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
              id="password"
              type="password"
              placeholder="Password"
              value={password}
              onChange={(e) => setPassword(e.target.value)}
              required
            />
          </div>
          <div className="flex items-center justify-center">
            <button className="bg-slate-800 hover:bg-slate-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline w-full">
              Login
            </button>
          </div>
          <div className="text-center mt-4">
            <Link href="/signin" className="text-blue-500 hover:underline">Sign In</Link>
          </div>
        </form>
      </div>
    );
  }
}
```

## קוד לעמוד בחינת השדה ראייה

```
return (  
  <div  
    className={`flex items-center justify-center flex-col w-full mx-auto h-[84vh] bg-[aliceblue] rounded-[2%] border-2 border-[#bfdbfe] border-b border-purple-400 p-4`}  
    style={{  
      borderLeft: isLeft ? '8px solid #bfdbfe' : 'none',  
      borderRight: !isLeft ? '8px solid #bfdbfe' : 'none'  
    }}  
  >  
    <Button  
      variant="contained"  
      sx={{ ...  
    }}  
      onClick={() => setOpen(!open)}  
    >  
      Please read the details before use  
    </Button>  
    <div className="w-full" style={{ transform: `translateX(${distance}cm)`, transition: 'transform 0.3s ease-in-out' }}>...  
  </div>  
  <div className="flex flex-wrap gap-4 mb-4">  
    <Button  
      variant="contained"  
      sx={{ ...  
    }}  
      onClick={moveLeft}  
    >  
      Move Left  
    </Button>  
    <Button  
      variant="contained" ...  
      onClick={() => setIsLeft(!isLeft)}  
    >  
      Choose a Side  
    </Button>  
    <Button  
      variant="contained"  
      sx={{ ...  
    }}  
      onClick={moveRight}  
    >  
      Move Right  
    </Button>  
  </div>  
  <div className="flex flex-wrap gap-4 mb-4">  
    <Button  
      variant="contained"  
      sx={{ ...  
    }}  
      onClick={overWord}  
    >  
      Change Word  
    </Button>  
    <Button  
      variant="contained"  
      sx={{ ...  
    }}  
      onClick={handleMistake}  
    >  
      Mistake  
    </Button>  
  </div>  
  <Popup open={open} setOpen={setOpen} type={'field'} />  
</div>  
);
```

# הקוד לשיפור טשטוש הראיה

```
export default function Blur({ user, sizeUser }) {
  const overWord = () => {
  };

  const handleMistake = async () => {
    let taut = { fontSize: fontSize, distance: 1, eye: isLeftEye ? 'left' : 'right' };

    user.sizeWeaknesses.push(taut);
    const updatedWeaknesses = [...user.sizeWeaknesses, taut];

    // API route
    const response = await fetch('/api/updateUser', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        filter: { email: user.email },
        updateData: { sizeWeaknesses: updatedWeaknesses }
      })
    });

    if (response.ok) {
      const updatedUser = await response.json();
      console.log('User updated successfully:', updatedUser);
    } else {
      console.error('Failed to update user');
    }

    overWord();
  };

  return (
    <div className={` ${style.contain} border-b border-green-400 p-4`} >
      <Button
        variant="contained"
        style={{ ...
      }}
      onClick={() => setOpen(!open)}
    >
      Please read the details before use
    </Button>
    <p className={style.inContain}>Font size: {fontSize}mm</p>
    <p ...
    </p>
    <input
      type="number"
      value={fontSize}
      onChange={changeFontSize}
      className="border border-gray-300 rounded px-2 p-1 mb-4 text-center w-96"
      placeholder="Enter font size"
    />
    <div className="flex flex-wrap gap-4 mb-4">...
    </div>
    <div className="flex flex-wrap gap-4 mb-4">
      <button
        className="bg-green-500 hover:bg-green-700 text-white font-bold py-2 px-4 rounded flex-1"
        onClick={overWord}
      >
        Change Word
      </button>
      <button
        className="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-4 rounded flex-1"
        onClick={() => handleMistake()}
      >
        Mistake
      </button>
    </div>
    <Popup open={open} setOpen={setOpen} type={'blur'} />
  </div>
```

## הקוד לאבחון שדה ראייה

```
const RandomCharacterGame = ({ user }) => {
  useEffect(() => {
    const updateUser = async () => {
      } catch (error) {
        console.error('Error updating user:', error);
      }
    }
  });

  updateUser();
}, [countFalse]);

return (
  <div className="flex flex-col items-center justify-center space-y-4 p-4 bg-gray-100 rounded-lg">
    <div className="bg-blue-50 border-2 border-blue-600 rounded-lg p-5 text-center font-sans text-gray-800 text-lg shadow-md transition-transf
      {`Cover your ${isLeftEye ? "right" : "left"} eye`} <br />
      {`And examine your ${isLeftEye ? "left" : "right"} eye`} <br />
      {`Tap the button corresponding to the character, 1 meter away`}
    </div>

    <div
      className="flex items-center justify-center w-16 h-16 bg-white rounded-full shadow-md"
      style={{ fontSize: `${size}mm` }}
    >
      {character}
    </div>

    <div className="grid grid-cols-2 gap-2">
      {!(countFalse === 3) ?
        buttons.map((char, index) => (
          <Button
            key={index}
            onClick={() => handleButtonClick(char, index)}
            variant="outlined"
            sx={{
              border: '2px solid #facc15',
              color: 'black', // Set text color to black
              fontSize: '14.6mm', // Adjust font size as needed
              padding: '10px 20px',
              width: '100px', // Width for square shape
              height: '100px',
            }}
            disabled={clickedIndex !== null}
            className={`${clickedIndex === index
              ? isCorrect
                ? 'bg-green-500 hover:bg-gray-600'
                : 'bg-red-500 hover:bg-red-600'
              : 'bg-white hover:bg-gray-200'}
            `}
          >
            {char}
          </Button>
        )) : <ResultTestBlur size={size} setSize={setSize} isLeftEye={isLeftEye} setIsLeftEye={setIsLeftEye} setCountFalse={setCountFalse} />
      }
    </div>
  </div>
);
};

export default RandomCharacterGame;
```



## קוד לאבחון עיוורון צבעים

```
const ColorChangerComp = ({ user, colorsUser }) => {
  const handleMistake = async () => {
    if (response.ok) {
      const updatedUser = await response.json();
      console.log('User updated successfully:', updatedUser);
    } else {
      console.error('Failed to update user');
    }
  }

  overWord();
};

const fontFileStyle = {
  backgroundImage:
    `linear-gradient(45deg, ${fontColor} 25%, transparent 25%, transparent 75%, ${fontColor} 75%, ${fontColor}),
    linear-gradient(45deg, ${fontColor} 25%, transparent 25%, transparent 75%, ${fontColor} 75%, ${fontColor})`, // יוצר אפקט פסיפס לזונט
};

const backgroundFileStyle = {
  backgroundImage:
    `linear-gradient(45deg, ${backgroundColor1} 25%, ${backgroundColor2} 25%, ${backgroundColor2} 75%, ${backgroundColor1} 75%, ${backgroundColor1}),
    linear-gradient(45deg, ${backgroundColor1} 25%, ${backgroundColor2} 25%, ${backgroundColor2} 75%, ${backgroundColor1} 75%, ${backgroundColor1})`, // יוצר אפקט פסיפס לרקע
};

return (
  <div className={`w-2/3 ${styles.contain} p-4`} >
    <div className={styles.inContain}>
      <div className={styles.rekaBackground} style={backgroundFileStyle}>
        <div className={styles.font} style={fontFileStyle}>
          {words[currentWordIndex]}
        </div>
      </div>
      <br />
      <div className="flex flex-wrap gap-4 mb-4">
        <button
          className="bg-slate-800 hover:bg-slate-700 text-orange-400 font-bold py-2 px-4 rounded flex-1"
          onClick={changeBackgroundColor}>
          Change Background Color
        </button>
        <button
          className="bg-slate-800 hover:bg-slate-700 text-orange-400 font-bold py-2 px-4 rounded flex-1"
          onClick={changeFontColor}>
          Change Font
        </button>
      </div>
      <div className="flex flex-wrap gap-4 mb-4">
        <button
          className="bg-green-500 hover:bg-green-700 text-white font-bold py-2 px-4 rounded flex-1"
          onClick={overWord}>
          Change Word
        </button>
        <button
          className="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-4 rounded flex-1"
          onClick={() => handleMistake()}>
          Mistake
        </button>
      </div>
    </div>
  </div>
);
};
```

## העמוד לניהול האתר

```
return (
  <div className="container mx-auto p-4 w-2/3 border-2 border-orange-200">
    <input
      type="text"
      placeholder="Search by username or email"
      value={searchQuery}
      onChange={handleSearch}
      className="mb-4 p-2 border border-gray-300 rounded w-full"
    />
    <table className="min-w-full bg-white shadow-md rounded-lg overflow-hidden">
      <thead className="bg-gray-200">
        <tr>
          <th onClick={() => requestSort('username')} className="cursor-pointer p-4 text-left">Username</th>
          <th onClick={() => requestSort('email')} className="cursor-pointer p-4 text-left">Email</th>
          <th className="p-4 text-left">Color Weaknesses</th>
          <th className="p-4 text-left">Size Weaknesses</th>
          <th className="p-4 text-left">Field Weaknesses</th>
        </tr>
      </thead>
      <tbody>
        {usersToDisplay.map((user, index) => (
          <tr key={index} className="border-t even:bg-gray-50 hover:bg-gray-100">
            <td className="p-4">{user.username}</td>
            <td className="p-4">
              <Link href={`mailto:${user.email}`}>{user.email}</Link>
            </td>
            <td className="p-4">
              <Weaknesses
                type="color"
                weaknesses={user.colorWeaknesses}
                email={user.email}
                deleteWeakness={deleteWeakness}
              />
            </td>
            <td className="p-4">
              <Weaknesses
                type="size"
                weaknesses={user.sizeWeaknesses}
                email={user.email}
                deleteWeakness={deleteWeakness}
              />
            </td>
            <td className="p-4">
              <Weaknesses
                type="field"
                weaknesses={user.fieldWeaknesses}
                email={user.email}
                deleteWeakness={deleteWeakness}
              />
            </td>
          </tr>
        ))}
      </tbody>
    </table>
    {visibleUsersCount < filteredUsers.length && (
      <button
        onClick={handleShowMore}
        className="mt-4 px-4 py-2 bg-orange-200 text-white rounded"
      >
        Show More
      </button>
    )}
  </div>
);
}
```

## הקוד לסטאטוס של המשתמש

```
export default function DetailUser({ simplifiedUser }) {
  return (
    <div className="mx-auto w-2/3 p-6 bg-gray-50 shadow-sm rounded-lg">
      <h2 className="text-2xl font-bold mb-6 text-center">User Details</h2>

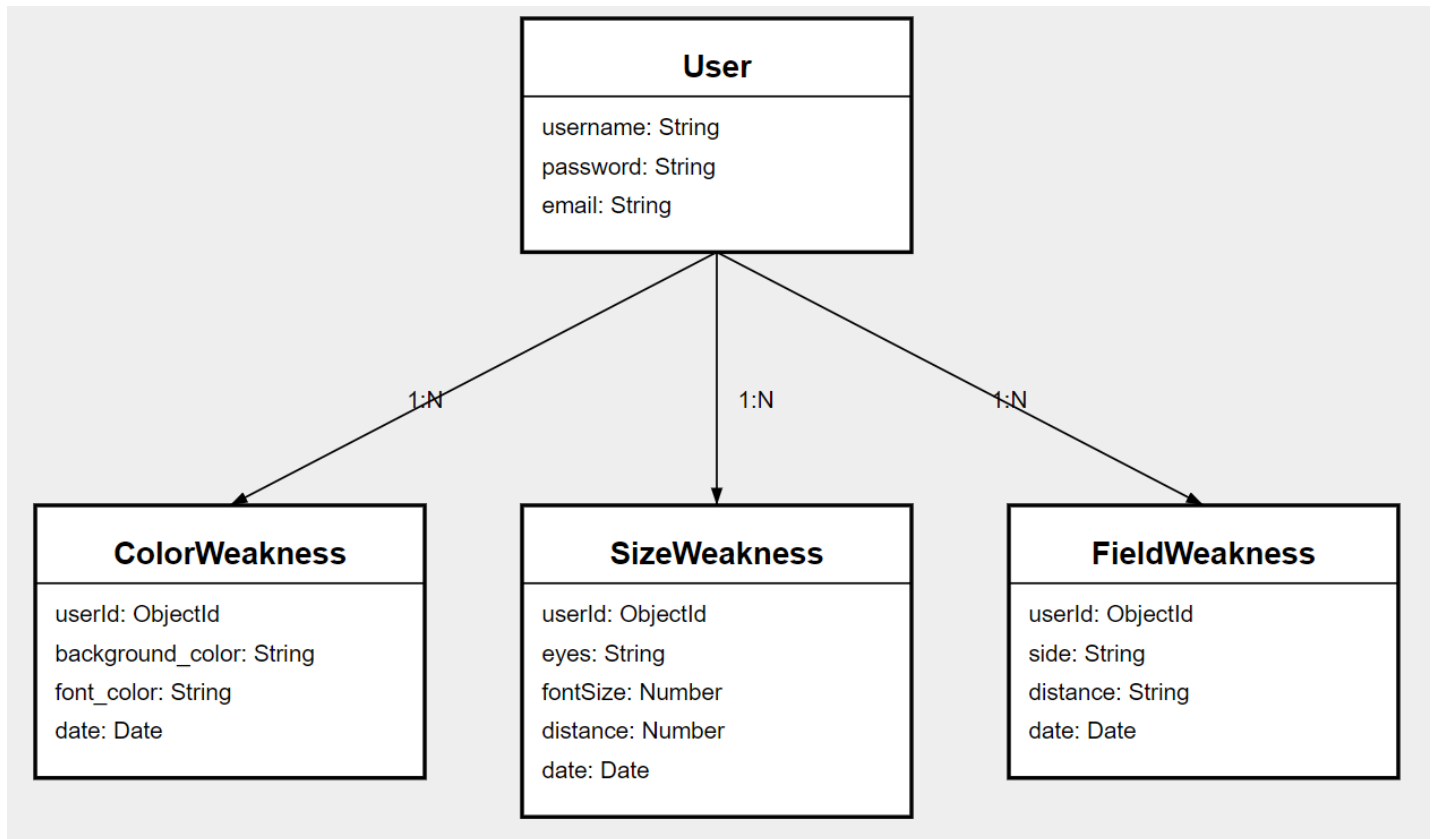
      <div className="mb-8">
        <p className="flex justify-between items-center border-b py-3 text-lg">
          <span className="font-medium text-gray-600">Username:</span>
          <span className="text-gray-800">{simplifiedUser.username}</span>
        </p>
        <p className="flex justify-between items-center border-b py-3 text-lg">
          <span className="font-medium text-gray-600">Email:</span>
          <span className="text-gray-800">{simplifiedUser.email}</span>
        </p>
      </div>

      <ColorDetails simplifiedUser={simplifiedUser} />

      <SizesDetails simplifiedUser={simplifiedUser} />

      <h3 className="text-xl font-semibold mb-4">Field Weaknesses</h3>
      <ul className="space-y-4">
        {simplifiedUser.fieldWeaknesses.length === 0 ? (
          <li className="border p-4 rounded-lg bg-white shadow-sm">
            <a href="#" className="text-red-500 underline">
              You don't have any diagnosis about your vision fields, Click here to explore field options
            </a>
          </li>
        ) : (
          simplifiedUser.fieldWeaknesses.map((weakness, index) => (
            <Link href={`field/${weakness.side}_${weakness.distance}`} key={index}>
              <li className="border p-4 rounded-lg bg-white shadow-sm">
                <div className="flex justify-between items-center py-1">
                  <span className="font-medium text-gray-600">Side:</span>
                  <span className="text-gray-800">{weakness.side}</span>
                </div>
                <div className="flex justify-between items-center py-1">
                  <span className="font-medium text-gray-600">Distance:</span>
                  <span className="text-gray-800">{weakness.distance}</span>
                </div>
                <div className="flex justify-between items-center py-1">
                  <span className="font-medium text-gray-600">Date:</span>
                  <span className="text-gray-800">
                    {new Date(weakness.date).toLocaleDateString()}
                  </span>
                </div>
              </li>
            </Link>
          ))
        )}
      </ul>
    </div>
  )
}
```

## הסבר על המבנה נתונים



### 1. ישות ראשית: משתמש

זוהי ישות הליבה המייצגת משתמש במערכת.

code

```
User {
  username: String
  password: String
  email: String
}
```

זוהי ישות האב שאליה מתייחסים ישויות אחרות באמצעות מערכת יחסים של אחד לרבים, כלומר למשתמש אחד יכול להיות מספר רשומות בכל אחת מהקטגוריות "חולשה".

## 2. ColorWeakness Entity

נראה שהישות הזו מטפלת בהגדרות נגישות הקשורות לצבע:

code

```
ColorWeakness {  
  ObjectId (Reference to User) :userId  
  String :background_color  
  String :font_color  
  Date :date  
}
```

מִטְרָה:

-מנהל העדפות נגישות הקשורות לצבע  
-שימושי למשתמשים עם עיוורון צבעים או ליקוי בראיית צבעים

## 3. ישותSizeWeakness

ישות זו מנהלת העדפות ראייה הקשורות לגודל:

```
SizeWeakness {  
  userId: ObjectId (Reference to User)  
  eyes: String  
  fontSize: Number  
  distance: Number  
  date: Date  
}
```

מִטְרָה:

-מטפל בצרכי נגישות הקשורים לגודל  
-שימושי למשתמשים עם:  
\*קוצר ראייה (קוצר ראייה)  
\*היפראופיה (רוחק ראייה)

\*בעיות חדות ראייה כלליות  
-השדה 'מרחק' מייצג מרחק צפייה אופטימלי  
-השדה 'עיניים' עשוי לאחסן מידע ספציפי על מצב העין

#### 4. ישות חולשת שדה

נראה כי ישות זו מטפלת במגבלות שדה הראייה:

```
FieldWeakness {  
  userId: ObjectId (Reference to User)  
  side: String  
  distance: String  
  date: Date  
}
```

מִטְרָה :

-ניהול צרכי נגישות הקשורים לשדה הראייה  
-שימושי למשתמשים עם:  
\*אובדן ראייה היקפית  
\*עיוורון חלקי  
\*חיתוכים בשדה הראייה  
\*מצבים כמו גלאוקומה

## **היבטים מרכזיים של המבנה:**

### **1. סוג מערכת היחסים:**

-קשר אחד לרבים (1:N) בין משתמש לכל סוג חולשה  
-למשתמש יכול להיות מספר רשומות בכל קטגוריה  
-כל רשומת חולשה שייכת בדיוק למשתמש אחד

### **2. מעקב זמני:**

-כל ישויות החולשה כוללות שדה 'תאריך'  
-זה מאפשר מעקב אחר שינויים לאורך זמן  
-שימושי לניטור התקדמות התנאים  
-מאפשר ניתוח היסטורי של צרכי המשתמש

### **3. גמישות:**

-גופים נפרדים מאפשרים ניהול עצמאי של סוגים שונים של צרכים ויזואליים  
-למשתמשים יכולים להיות כל שילוב של מגבלות אלו  
-קל להרחבה עם שדות נוספים במידת הצורך

### **4. דפוס משותף:**

-כל ישויות החולשה חולקות אלמנטים משותפים:

\*מזהה משתמש עבור מערכת יחסים

\*תאריך למעקב זמני

-מבנה עקבי מקל על התחזוקה

מקרי שימוש פוטנציאליים:

### **1. יישומי נגישות:**

-התאמת ממשק משתמש אוטומטית על סמך מגבלות המשתמש  
-הצגת תוכן מותאמת אישית  
-עיצוב ממשק אדפטיבי

### **2. מעקב רפואי:**

-מעקב אחר שינויים בתנאי הראייה

-מעקב אחר התקדמות ליקויי ראייה

מבנה זה מספק בסיס איתן לניהול סוגים שונים של צרכי נגישות חזותית תוך שמירה על עקרונות ארגון מסד נתונים טובים וגמישות להרחבות עתידיות.



## סיכום ומסקנות עבור מבנה הנתונים של האתר

תקציר:

AYINBRIGHT-הוא אתר שנועד לספק פתרונות נגישות למשתמשים עם ליקויי ראייה שונים. מבנה הנתונים בנוי סביב ניהול סוגים שונים של מגבלות ויזואליות והתאמה אישית של חווית המשתמש בהתאם.

סקירת מבנה הליבה:

-ישות משתמש מרכזית עם פרטי אימות בסיסיים  
-שלוש ישויות מיוחדות העוקבות אחר אתגרים ויזואליים שונים:

\*תפיסת צבע (ColorWeakness)

\*דרישות גודל/מרחק (SizeWeakness)

\*מגבלות שדה ראייה (FieldWeakness)

נקודות החוזק העיקריות של המבנה:

### 1. יכולות התאמה אישית

-מאפשר חוויות משתמש מותאמות במיוחד  
-יכול להתאים רכיבי ממשק בהתאם לצרכי המשתמש הספציפיים  
-תומך במספר פרופילים עבור מצבים שונים

### 2. מעקב רפואי

-מעקב זמני באמצעות שדות תאריכים  
-יכולת מעקב אחר שינויים בתנאי הראייה  
-פוטנציאל להפקת דוחות התקדמות

### 3. עיצוב מודולרי

-הפרדה ברורה של חששות  
-קל לתחזוקה והארכה  
-ניתן להרחבה עבור תכונות נוספות

#### 4. מיקוד נגישות

- בנוי במיוחד עבור נגישות הקשורה לראייה
- מבסה קטגוריות עיקריות של ליקויי ראייה
- תומך בפתרונות נגישות מקיפים

מסקנות:

##### 1. יישור מטרה

- ✓המבנה תומך ביעילות במשימתה של AYINBRIGHT לספק חוויות אינטרנט נגישות
- ✓מתאים לניהול צורכי נגישות חזותית מסוגים שונים
- ✓מאפשר התאמה אישית מפורטת של רכיבי ממשק משתמש

##### 2. יישום טכני

- ✓עיצוב מסד נתונים נקי ומנומל
- ✓יחסי אחד לרבים יעילים
- ✓הפרדה טובה בין חששות

##### 1. שיפורים פוטנציאליים:

- יכול להוסיף רמות חומרה עבור כל סוג מגבלה
  - עשוי להפיק תועלת מקביעות מוגדרות מראש של העדפות
  - יכול לכלול מידע רפואי מפורט יותר
  - עשוי להוסיף תמיכה עבור פרופילי מכשירים מרובים
2. ערך עסקי

- ✓תומך בתאימות לנגישות
- ✓מאפשר שיפורים מונעי נתונים
- ✓מקל על חוויות משתמש מותאמות אישית

5. מדרגיות עתידית

✓ מבנה יכול להכיל תכונות חדשות

✓ קל להרחבה לצרכי נגישות נוספים

✓ תומך באינטגרציה פוטנציאלית עם מערכות אחרות

המלצות לפיתוח עתידי:

### **הערכה כוללת:**

מבנה הנתונים מספק בסיס איתן לאתר ממוקד נגישות של AYINBRIGHT. הוא מאזן בהצלחה בין פשטות לפונקציונליות, תוך שמירה על יכולת הרחבה לצמיחה עתידית. המבנה תומך ביעילות במשימת הליבה של שיפור נגישות האינטרנט עבור משתמשים עם ליקויי ראייה.

העיצוב מדגים שיטות מודל טובות של מסדי נתונים תוך שמירה על התמקדות בפתרונות נגישות מעשיים. הוא ממוקם היטב כדי לתמוך הן בצרכים הנוכחיים והן בהרחבות עתידיות של פלטפורמת AYINBRIGHT.

מבנה זה מגדיר את AYINBRIGHT להצלחה במתן חוויות אינטרנט מותאמות אישית ונגישות תוך שמירה על הגמישות לצמוח ולהסתגל לצרכי המשתמש ולהתקדמות הטכנולוגית בתחום הנגישות.