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

**הגשת פרויקט**

**להגשה עד 29.8.24 בשעה 23:59**

**<G14> <B30> <Emotional Intelligence Trainer>**

|  |  |
| --- | --- |
| שם חבר.ת הצוות | ת"ז |
| מוראד עסאקלה | 315688069 |
| שאדי אבו ריש | 208658898 |
| אניל סלאמה | 316198522 |
| ראיד חדאד | 207075383 |

**תקציר הפרויקט :**   
הפרויקט שלנו הוא פלטפורמת Trainer Emotional Intelligence שנועדה לאבחן וגם לשפר את האינטליגנציה הרגשית של המשתמשים. הפלטפורמה מאפשרת למשתמשים להירשם, לבצע מבחן הערכה, ולקבל תוצאות מותאמת אישית בהתאם למבחן ההערכה ,והמשתמש גם מקבל תוכניות אימון לשיפור יכולותיהם הרגשיות. המערכת מציעה כלים להמשך התפתחות אישית, תוצאות מפורטות של המבחן, צפיה בהיסטורית תוצאות ומעקב אחר ההתקדמות ודף חדשות שקשור לכל זה ואכן אפשר ליצור איתנו קשר בדף ה- CONTACT US וזה מגיע למייל שלנו.

**מימוש-שמות הטכנולוגיות המרכזיות בכל אחד מהחלקים :**

* **Frontend**:

**React.js** לבניית ממשקי משתמש אינטראקטיביים.

**Tailwind CSS** לעיצוב מהיר ויעיל.

* **Backend**:

**Node.js** עם **Express.js** לבניית שרתים וממשקי API.

* **Database**:

**MongoDB** לניהול מידע גמיש ומשתמשים.

קישורים :

[**קישור לתיקיית Git ציבורי**](https://github.com/moradasakli/Emotional-AI-Trainer)

[**קישור לאתר**](https://client-iota-liard-59.vercel.app/)

[**קישור ל MTW**](https://www.morethanwallet.com/app/884)

**1. חברי הצוות וחלוקת משימות :**

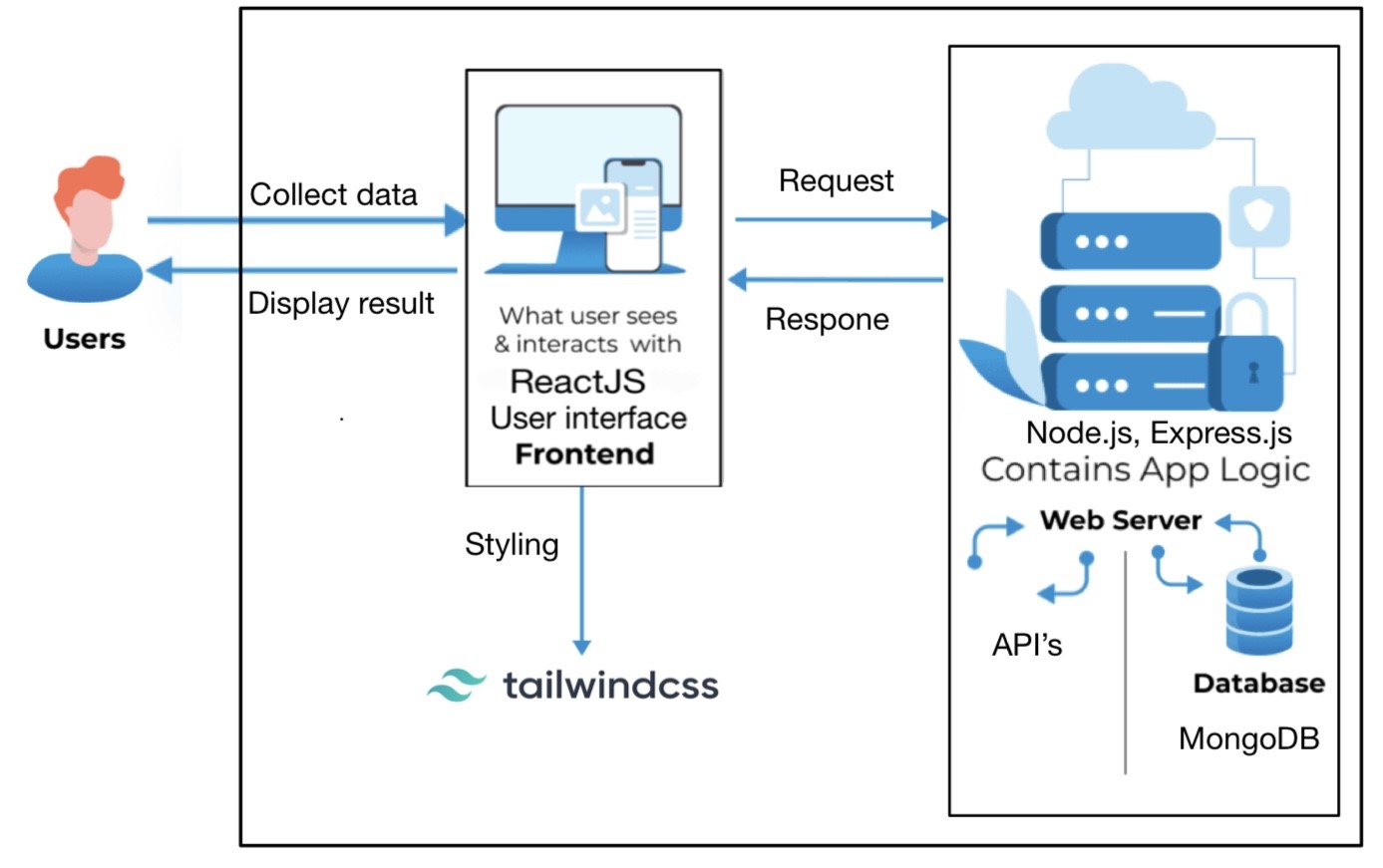
מהנדס מערכת – אניל סלאמה

|  |  |  |
| --- | --- | --- |
| שם חבר הצוות | משימות שהוקצו | משימות שהושלמו |
| מוראד עסאקלה | היה אחראי על הסדר של ה NAVBAR והניווט באתר בכללי, התחברות, רישום משתמש והיסטוריה עבור על משתמש ויצירת קשר במייל ע"י שימוש בשרת חיצוני (EMAILJS). | |  | | --- | | הושלם במלואו |  |  | | --- | |  | |
| שאדי אבוריש | היה אחראי על החלק של ה TRAINING , יצירת מגוון של תרגילים ע"י שליפת נתונים מקובץ JSON, היה לו גם תפקיד בעיצוב של דף הבית ובאתר בכללי. | |  | | --- | | הושלם במלואו |  |  | | --- | |  | |
| אניל סלאמה | היה אחראי על פיתוח דף המבחן (EXAM) והעיצוב שלו. ע"י שליפת נתונים מקובץ JSON והצגת התוצאות הרלוונטית בהתאם לניקוד המשתמש ועדכון במסד הנתונים להיסטוריית הניקוד של המשתמש. | |  | | --- | | הושלם במלואו |  |  | | --- | |  | |
| ראיד חדאד | היה אחראי על דף ה NEWS , בוצע ע"י שליפת מידע מ API חיצוני, בנוסף הוא ממש את פונקציונליות החיפוש בדף. | |  | | --- | | הושלם במלואו |  |  | | --- | |  | |

**2. רשימת הדרישות הפונקציונליות ו – הלא פונקציונאליות:**

|  |  |  |  |
| --- | --- | --- | --- |
| **מספר** | **דרישות** | **סוג** | **קטגוריה** |
| 1 | המערכת תאפשר הרשמת משתמשים חדשים . | FR |  |
| 2 | הנתונים הנשמרים הם : שם משתמש , מייל , סיסמה והציון ההתחלתי . | NFR | Data Integrity |
| 3 | המערכת תאפשר כניסה. | FR |  |
| 4 | המערכת תאפשר יציאה. | FR |  |
| 5 | הכניסה תתבצע על ידי הזנת משתמש וסיסמה. | NFR | Security |
| 6 | המערכת מאפשרת צפיה בחדשות. | FR |  |
| 7 | החדשות נשלפות מאתרים קשורים שמכילים חדשות בנושא רגשות. | NFR | Interoperability |
| 8 | המערכת מאפשרת חיפוש מידע. | FR |  |
| 9 | המערכת מאפשרת יצירת קשר איתנו. | FR |  |
| 10 | יצירת קשר נעשית על ידי שליחת הודעה למייל . | NFR | Usability |
| 11 | המערכת מעדכנת סטטוס עם אימוג'י. | FR |  |
| 12 | המערכת מאפשרת למשתמש ביצוע אבחון. | FR |  |
| 13 | שאלות האבחון מחולקות לפי קטגוריות שונות : Secure, Regulated, Valuable, Decided, Related. | NFR | Usability |
| 14 | המערכת מאפשרת ניווט בין השאלות. | FR |  |
| 15 | המערכת מחשבת ציון לכל קטגוריה בסוף האבחון ומחשבת ממוצע. | FR |  |
| 16 | המערכת מנתחת ומציגה את התוצאות בהתאם לציון הקטגוריה המינימלית. | FR |  |
| 17 | המערכת שומרת את התוצאה להיסטורית התוצאות של המשתמש. | FR |  |
| 18 | המערכת מאפשרת צפיה בהיסטורית התוצאות. | FR |  |
| 19 | התוצאות מוצגות למשתמש בצורת גרף. | NFR | Usability |
| 20 | התוצאות ממוינות לפי תאריכים. | NFR | Usability |
| 21 | המערכת מאפשרת מעבר למצב לילה (מעבר בין מצבים) | FR |  |
| 22 | האתר צריך להיות מותאם למכשירים שונים (סמארטפונים , טאבלטים , ... ) | NFR | Usability |
| 23 | האתר צריך להיות ידידותי למשתמש , נוח ויזואלית ומעוצב . | NFR | Usability |
| 24 | הביצועים של האתר צריכים להיות מהירים . | NFR | Performance |
| 25 | . האתר יהיה זמין 24/7. | NFR | Availability |
| 26 | קוד האתר יהיה מובנה ומתועד באופן שיאפשר עדכונים קלים, תיקוני באגים ושיפורים של תכונות. | NFR | Maintainability |
| 27 | המערכת תתמוך במספר הולך וגדל של משתמשים וגרפים מבלי לגרום לירידה בביצועים או בשימושיות.  ראש הטופס  תחתית הטופס | NFR | Scalability |

**2.1. ארכיטקטורת המערכת של האתר :**

****

**פירוט ארכיטקטורת המערכת:**

1. **Users  משתמשים:**

המשתמשים הם אלה שמבצעים פעולות באתר כמו הרשמה, התחברות, לקיחת מבחן אינטליגנציה רגשית, צפייה בתוצאות, והתחלת אימון אישי מותאם.

משתמשים שולחים בקשות לשרת ומקבלים ממנו תגובות המוצגות בצד הFrontend.

1. **ReactJS - User Interface Frontend:**

חלק ה-Frontend של האתר בנוי בעזרת ReactJS, שם המשתמש רואה ומבצע אינטראקציה עם ממשק המשתמש.

ReactJS מטפל בהצגת הדפים, בטפסים, ובאינטראקציות עם המשתמש כמו לקיחת מבחן האינטליגנציה הרגשית.

ה-Frontend גם אחראי על איסוף נתונים מהמשתמש (כמו תשובות למבחן) ושליחתם לשרת, וקבלת תגובות להצגה על המסך.

1. **TailwindCSS – Styling:**

כל העיצוב באתר מבוצעים באמצעות  TailwindCSS למעט דף של ה בוחן עוצב ב קובץ CSS חיצוני מספק כלים ליצירת עיצוב מותאם אישית ומהיר עבור כל הרכיבים באתר, דבר המאפשר יצירת ממשק משתמש מושך ומותאם אישית.

1. **Node.js & Express.js – Backend:**

ה-Backend של האתר בנוי על Node.js ו-Express.js, והוא אחראי על הלוגיקה של האפליקציה.

השרת מקבל בקשות מה Frontend כמו שליחת תשובות למבחן או בקשה לצפייה בתוצאות מעבד את הבקשות ומחזיר תגובות מתאימות.

ה-API's מנוהלות על ידי Express.js המאפשר תקשורת חלקה בין הFrontend ו-Backend.

1. **MongoDB – Database:**

MongoDB  הוא מסד הנתונים של האתר. הוא משמש לאחסון מידע כמו פרטי משתמשים, תשובות למבחן, תוצאות, ומידע נוסף שנדרש לניהול האתר.

ה-Backend (Node.js & Express.js) מבצע שאילתות למסד הנתונים כדי לקרוא ולכתוב נתונים, ומחזיר את המידע המבוקש ל-Frontend להצגה.

זרימת התהליך באתר:

Collect Data  איסוף נתונים:

המשתמש מקיים אינטראקציה עם ה-Frontend של האתר. לדוגמה, כאשר המשתמש לוקח את מבחן האינטליגנציה הרגשית, הנתונים נאספים על ידי ה Frontend.

Request  בקשה:

ה-Frontend שולח בקשה ל-Backend. לדוגמה, בקשה יכולה להיות שליחת תשובות המבחן לניתוח ב Backend

Response  תגובה:

ה-Backend מקבל את הבקשה, מעבד אותה (כולל גישה למסד הנתונים במידת הצורך), ומחזיר תגובה ל-Frontend עם המידע הנדרש.

Display Result  הצגת תוצאה:

ה-Frontend מקבל את התגובה מה-Backend ומציג את המידע למשתמש. לדוגמה, הצגת תוצאות המבחן או הצגת המלצות לאימון אישי מותאם.

קישור לכלים:

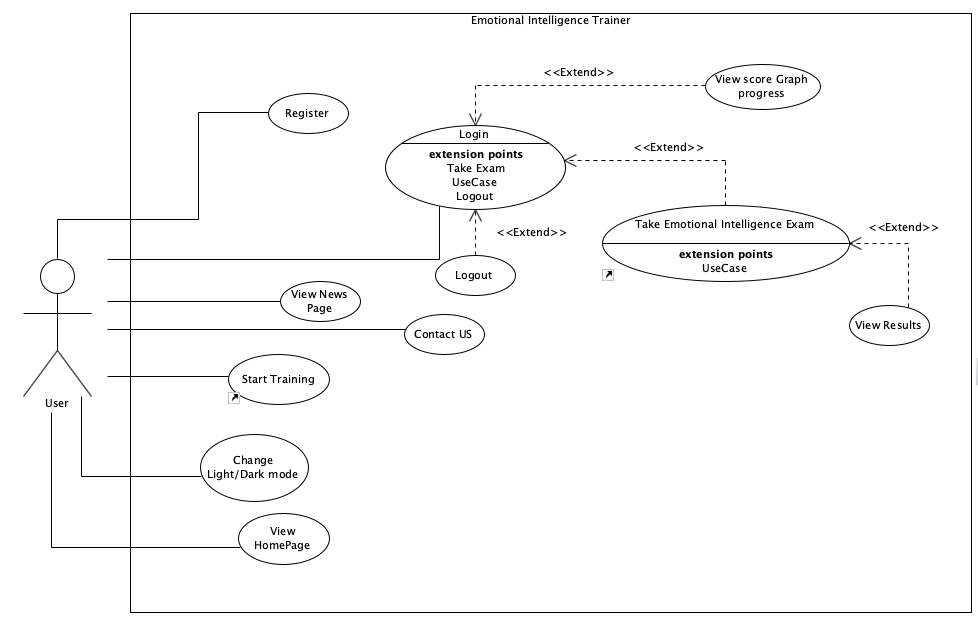
ReactJS  מספק את היכולת לבנות ממשק משתמש דינאמי ואינטראקטיבי.

TailwindCSS  מאפשרת לעצב את הממשק בצורה מהירה ויעילה.

Node.js & Express.js  מנהלים את כל הלוגיקה והתקשורת של ה Backend.

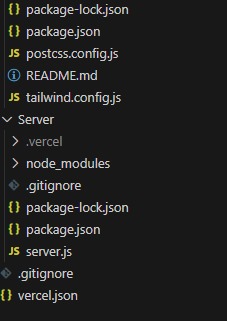
MongoDB  משמש לאחסון ואחזור מידע חיוני לאתר.

1. **דיאגרמת Use-Case :**

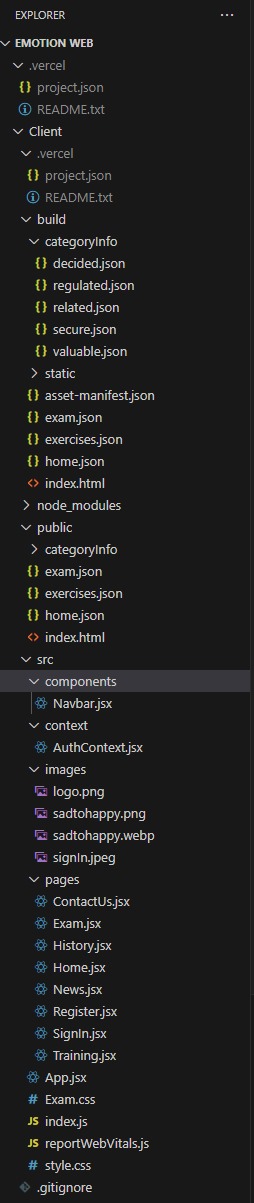
****

1. **מבנה סופי של האתר :**

**צד שרת :**



**צד לקוח :**



**4. א. דיאגרמה שמתארת התיקיות בפרויקט :**

**EMOTION TRAINER/**

**├── .vercel/**

**│   ├── project.json  # Vercel project configuration**

**│   └── README.txt    # Readme file with project-related information**

**├── Client/**

**│   ├── .vercel/**

**│   │   ├── project.json  # Vercel client project configuration**

**│   │   └── README.txt    # Readme file for the client**

**│   ├── build/**

**│   │   ├── categoryInfo/  # Directory containing JSON files for various categories**

**│   │   │   ├── decided.json  # JSON data for 'decided' category**

**│   │   │   ├── regulated.json  # JSON data for 'regulated' category**

**│   │   │   ├── related.json  # JSON data for 'related' category**

**│   │   │   ├── secure.json  # JSON data for 'secure' category**

**│   │   │   ├── valuable.json  # JSON data for 'valuable' category**

**│   │   └── static/**

**│   │       ├── asset-manifest.json  # Asset manifest file for the static build**

**│   │       ├── exam.json  # JSON data for exams**

**│   │       ├── exercises.json  # JSON data for exercises**

**│   │       ├── home.json  # JSON data for the home page**

**│   │       └── index.html  # Main HTML file for the build**

**│   └── public/**

**│       ├── categoryInfo/  # Public directory for category-related data**

**│       │   ├── exam.json  # Public JSON data for exams**

**│       │   ├── exercises.json  # Public JSON data for exercises**

**│       │   ├── home.json  # Public JSON data for the home page**

**│       │   └── index.html  # Public index HTML file**

**│   └── src/**

**│       ├── components/**

**│       │   ├── Navbar.jsx  # Navigation bar component**

**│       ├── context/**

**│       │   ├── AuthContext.jsx  # Authentication context for managing user state**

**│       ├── images/**

**│       │   ├── logo.png  # Logo image**

**│       │   ├── sadtohappy.png  # Image representing the emotional change**

**│       │   ├── sadtohappy.webp  # WebP version of the emotional change image**

**│       │   └── signIn.jpeg  # Image for the sign-in page**

**│       ├── pages/**

**│       │   ├── ContactUs.jsx  # Contact Us page component**

**│       │   ├── Exam.jsx  # Exam page component**

**│       │   ├── History.jsx  # History page component**

**│       │   ├── Home.jsx  # Home page component**

**│       │   ├── News.jsx  # News page component**

**│       │   ├── Register.jsx  # Registration page component**

**│       │   ├── SignIn.jsx  # Sign-in page component**

**│       │   ├── Training.jsx  # Training page component**

**│       ├── App.jsx  # Main application component**

**│       ├── Exam.css  # CSS file for styling the exam page**

**│       ├── index.js  # Main entry point for the React application**

**│       ├── reportWebVitals.js  # Script to measure web vitals**

**│       └── style.css  # General stylesheet for the application**

**├── Server/**

**│   ├── .vercel/**

**│   ├── node\_modules/**

**│   ├── .gitignore  # Git ignore file**

**│   ├── package-lock.json  # Lockfile for installed npm packages**

**│   ├── package.json  # Package configuration file**

**│   ├── server.js  # Main server-side script**

**│   └── vercel.json  # Vercel deployment configuration file**

**├── package-lock.json  # Lockfile for installed npm packages at the root**

**├── package.json  # Root package configuration file**

**├── postcss.config.js  # PostCSS configuration for TailwindCSS**

**├── README.md  # Project readme file**

**└── tailwind.config.js  # TailwindCSS configuration file**

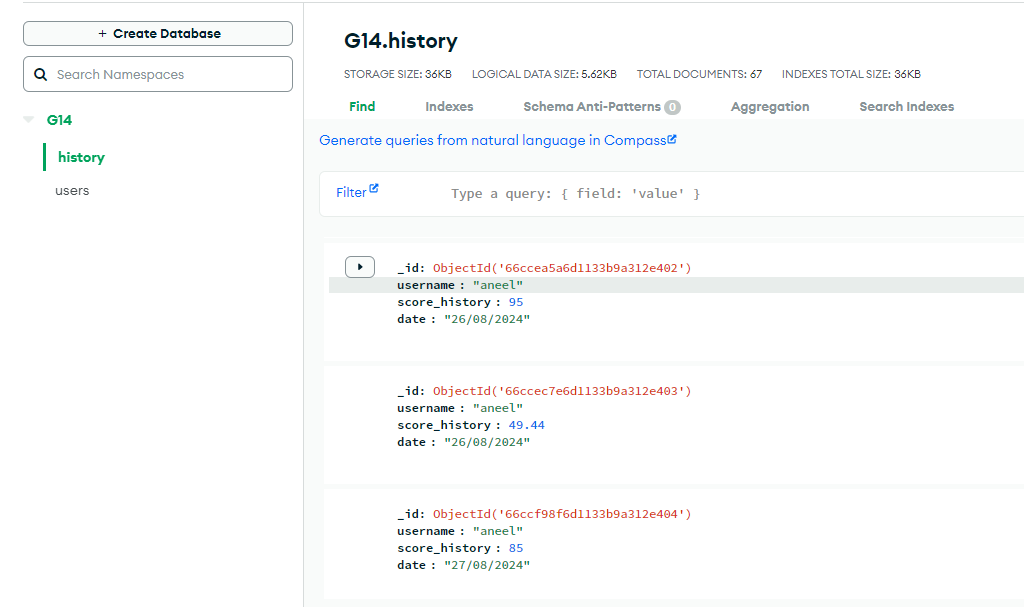
**4. ב. דיאגרמה שמתארת את מבנה ה DB :**

השתמשנו ב MongoDB כדי לאחסן את הנתונים , ושמרנו אותם בשתי קולקציות :

הראשונה בשם users לשמירת נתוני המשתמשים בעת יצירת חשבונות , עבור כל משתמש אנחנו שומרים : שם משתמש , מייל , סיסמה והציון שקיבל באבחון האחרון (0 בעת יצירת חשבון חדש) .



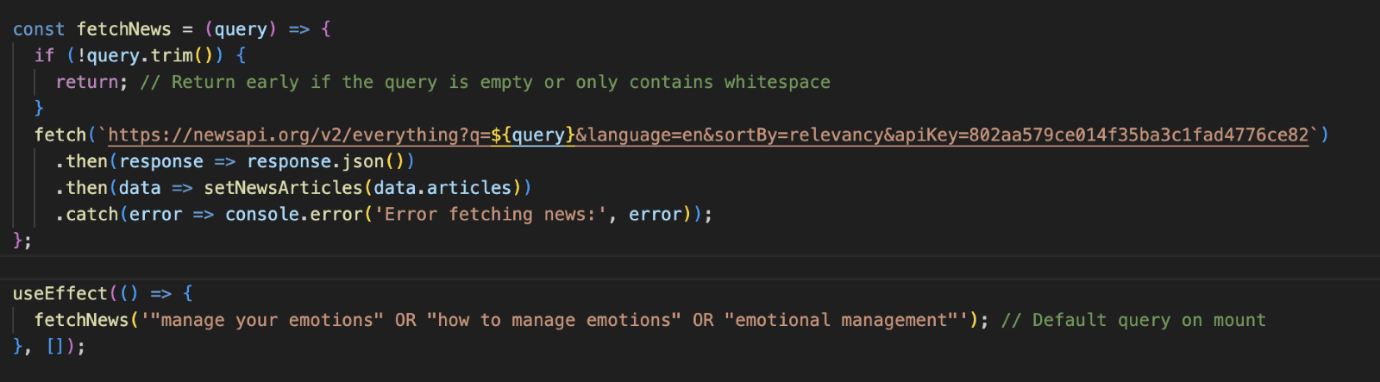
קולקציה שניה בשם history שבה שומרים את ציוני המבחנים בכל פעם שמשתמש מבצע אבחון , כל אובייקט מכיל את הנתונים הבאים : שם משתמש , ציון , ותאריך .



**5.תיק מתכנת :**

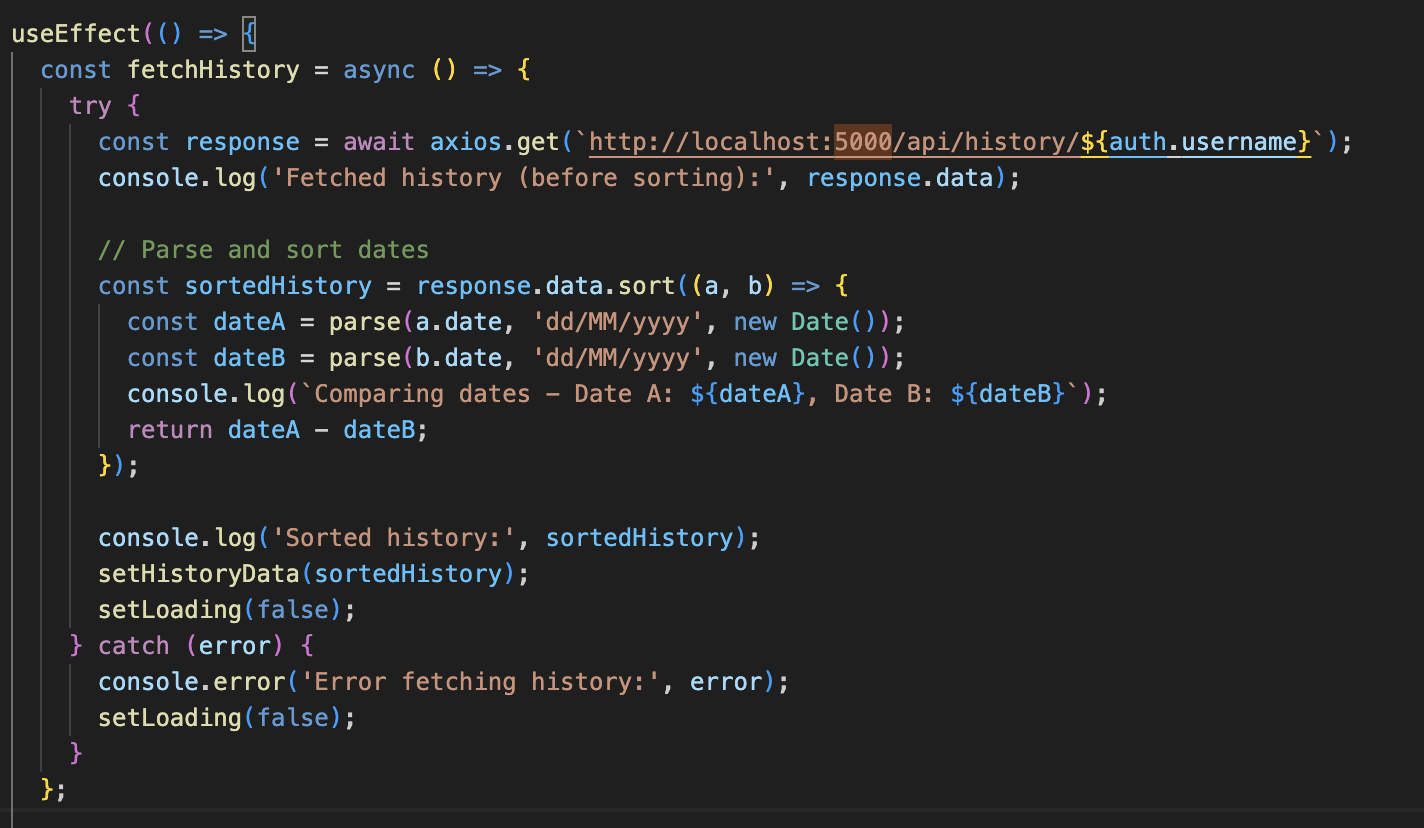
**חלק א' פונקציות מרכזיות בקוד :**

**News.jsx**

****

הפונקציה fetchNews מאחזרת מאמרי חדשות הקשורים לניהול רגשות על ידי שאילת API של חדשות עם מונחי חיפוש ספציפיים, ומעדכנת את מצב הרכיב עם המאמרים שהתקבלו. היא פועלת אוטומטית כשהרכיב נטען, תוך שימוש בשאילתת חיפוש ברירת מחדל כדי להציג חדשות רלוונטיות כברירת מחדל.

**History.jsx**

****

הפונקציה fetchHistory מאחזרת את נתוני ההיסטוריה של המשתמש מ-API, ממיינת אותם לפי תאריך, ומעדכנת את מצב הרכיב עם הנתונים הממוינים. היא מנהלת את מצב הטעינה ומציגה ביומן כל שגיאה שמתרחשת במהלך תהליך האחזור.

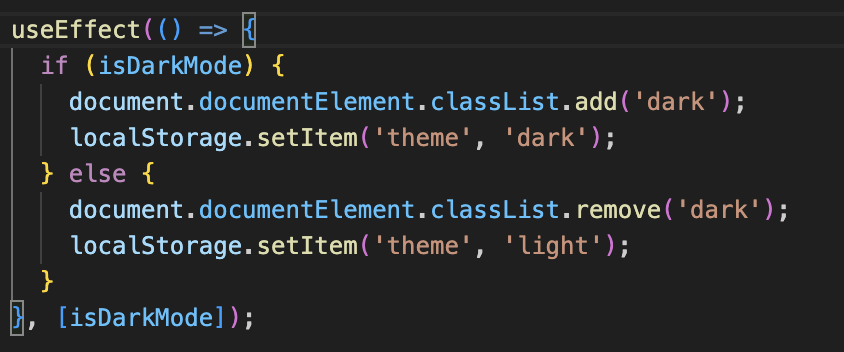
**Exams.jsx**   
****

קטע הקוד הזה משתמש ברכיב <Helmet> כדי להגדיר את כותרת מסמך ה-HTML ל-"EmotionAI - Exam" ולכלול אייקון מותאם אישית (favicon), שנקבע לפי תמונת הלוגו, בגודל של 16x16 פיקסלים.

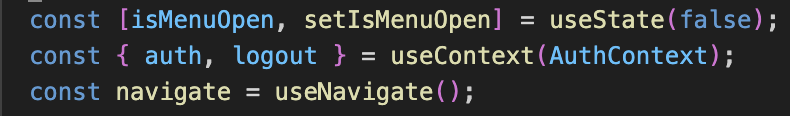
****

קוד זה מחשב את אחוזי הציון עבור כל קטגוריה על ידי סיכום תשובות המשתמש לשאלות באותה קטגוריה וחלוקה בציון המרבי האפשרי. לאחר מכן, הוא שומר את האחוזים הללו, מעוגלים לשתי ספרות אחרי הנקודה, באובייקט שנקרא categoryScores.

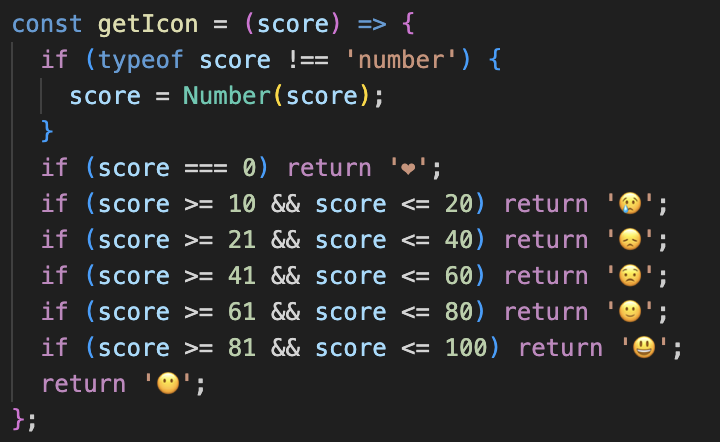
**Navbar.jsx**

****

ההוק useEffect מחליף בין ערכות נושא כהות ובהירות על ידי הוספה או הסרה של המחלקה 'dark' מהאלמנט הראשי של המסמך בהתאם למצב isDarkMode. הוא גם שומר את העדפת המשתמש לערכת הנושא ('כהה' או 'בהירה') באחסון המקומי (local storage).

****

קוד זה מנהל את המצב עבור תפריט ההמבורגר (isMenuOpen) ומביא מידע אימות (auth) ופונקציית יציאה מ-AuthContext. הוא גם מאפשר ניווט בין נתיבים באמצעות ההוק useNavigate.

****

פונקציית getIcon ממירה את הציון למספר (אם הוא לא כבר מספר) ומחזירה אימוג'י בהתאם לטווח הציונים: לב עבור 0, פנים עצובות עבור 10-20, ואימוג'ים נוספים עבור טווחים עד 100. אם הציון אינו נכלל באחד

מהטווחים המצוינים, הוא מחזיר באופן ברירת מחדל פנים נייטרליות.

**ContactUS.jsx   
**

פונקציית handleSubmit שולחת אימייל באמצעות EmailJS עם מזהי השירות, התבנית, והמשתמש המצוינים, יחד עם נתוני הטופס. היא מעדכנת את המצב ל-'SUCCESS' ומנקה את השדות אם האימייל נשלח בהצלחה, או מעדכנת את המצב ל-'FAILED' ומציגה את השגיאה אם השליחה נכשלה.

**חלק ב' פירוט של API :**

הפונקציה fetchNews:

אתה מבצע בקשה ל-API של חדשות כדי לאחזר מאמרים הקשורים לניהול רגשות. קריאת ה-API הזו מתבצעת כאשר הרכיב נטען ומביאה מאמרים בהתבסס על מונחי חיפוש מסוימים.

הפונקציה fetchHistory:

אתה מבצע בקשה ל-API של השרת שלך כדי לאחזר את נתוני ההיסטוריה של המשתמש. קריאת ה-API הזו מאחזרת נתונים מהשרת בהתבסס על שם המשתמש המאומת.

הפונקציה handleSubmit:

אתה משתמש בשירות EmailJS, שמתפקד כמו API, כדי לשלוח אימייל בהתבסס על מזהה השירות, מזהה התבנית, ומזהה המשתמש, יחד עם נתוני הטופס. למרות שזה לא API מסורתי, הוא עדיין מבצע בקשת HTTP חיצונית לשירות EmailJS כדי לשלוח את האימייל.

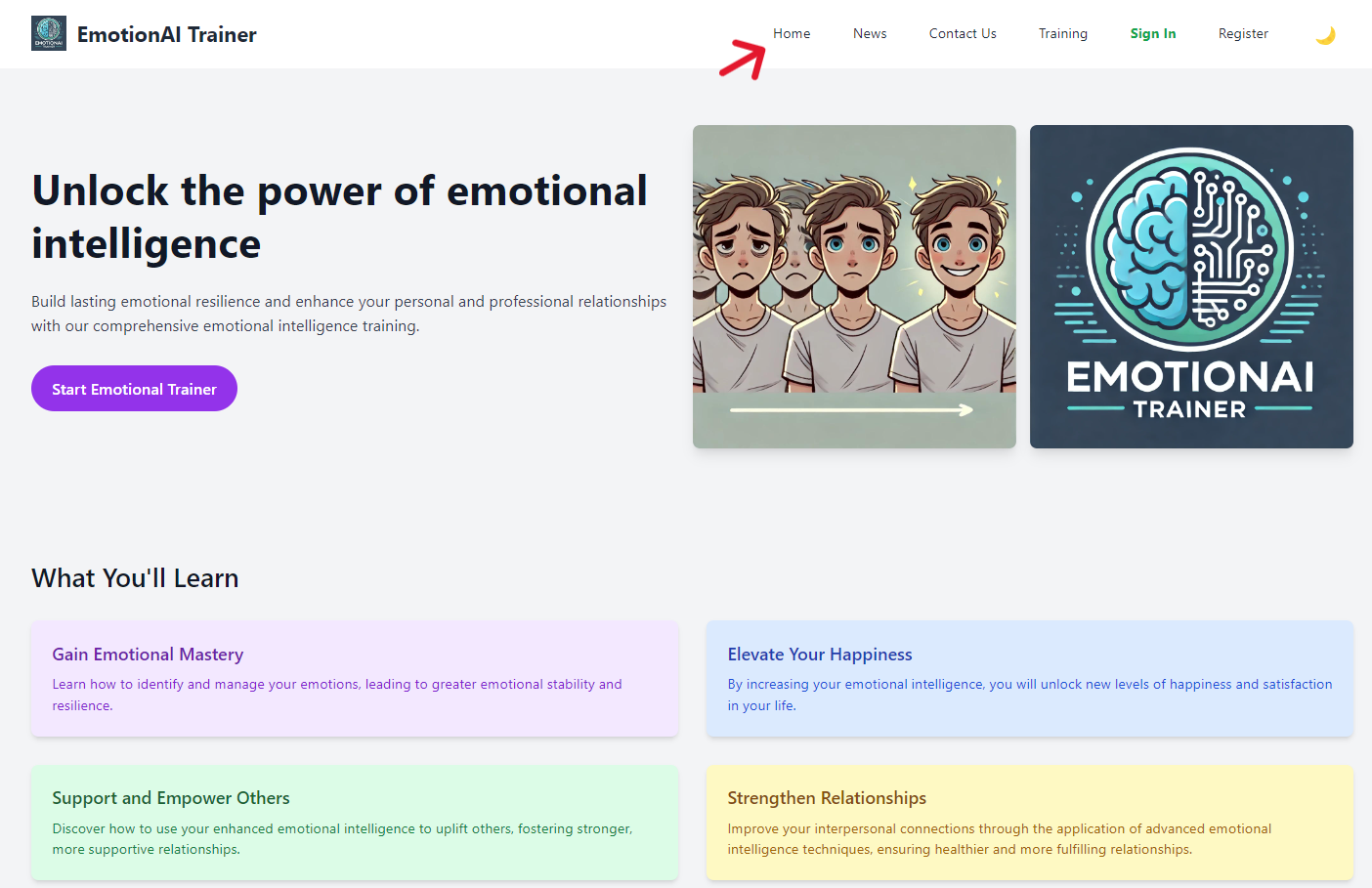
שליפת מידע על קטגוריה עם fetch:

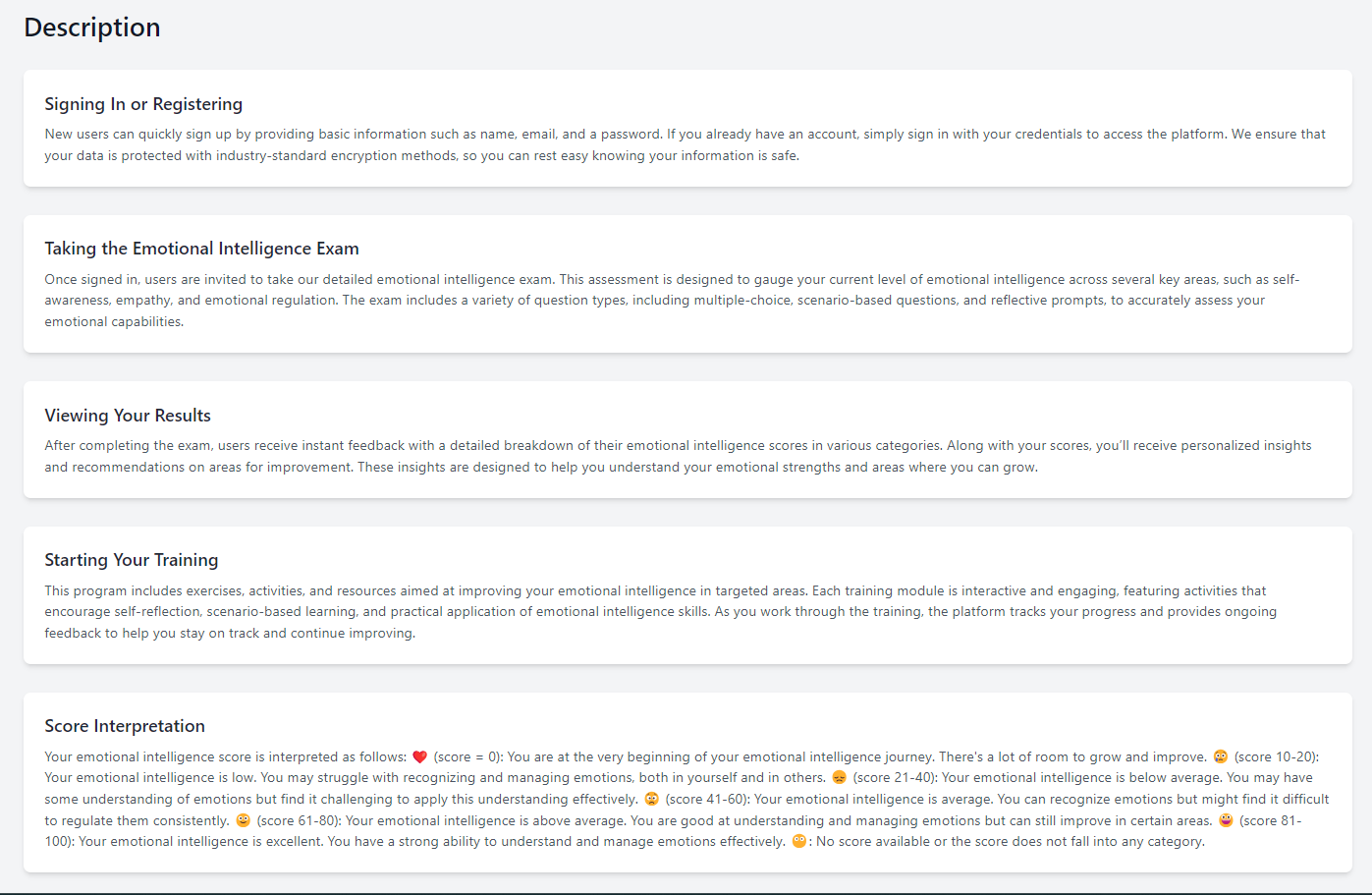
אתה מבצע קריאת API כדי לאחזר קובץ JSON המכיל מידע על הקטגוריה עם הציון הנמוך ביותר. זו בקשה דינמית לקבלת נתונים עבור הקטגוריה הנמוכה ביותר.

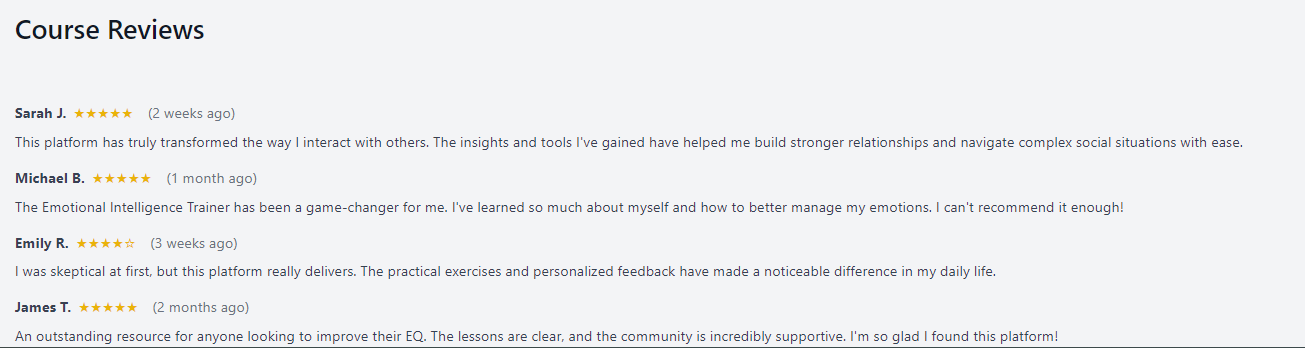
[קישור ל DB](https://cloud.mongodb.com/v2/66a62738bf7e440f01675d2c#/metrics/replicaSet/66a62800e265de4704c7bfb1/explorer/G14)

**6.תיק משתמש :**

**Home page**



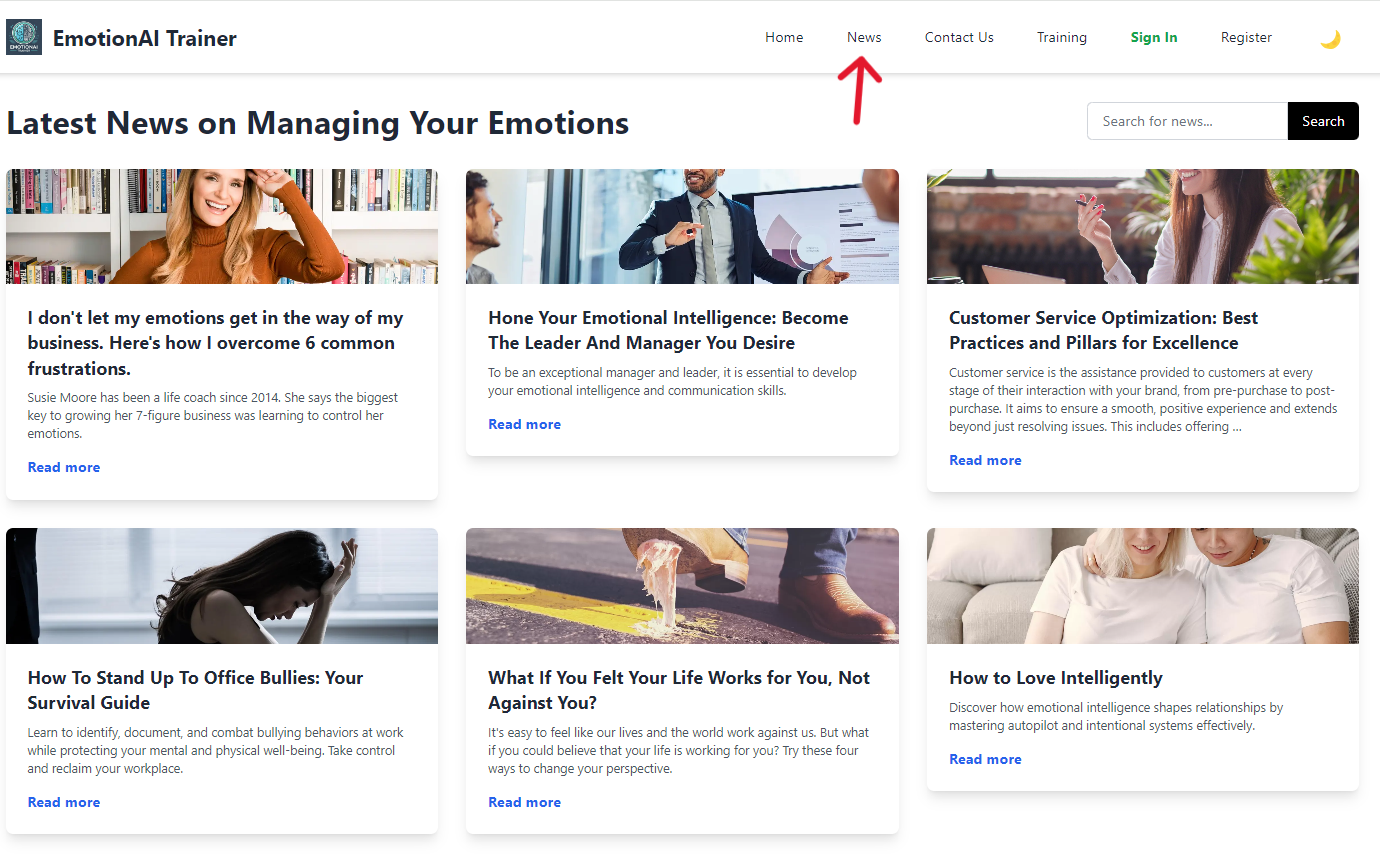




דף הבית של האתר , כאן אנחנו מציגים את האתר ואת התכנים שיש בו , הסברים כללים על התחום , תיאור כללי של האתר והסבר כיצד להתחיל באינטראקציה ועל אופן השימוש באתר בצורה האפקטיבית ביותר .

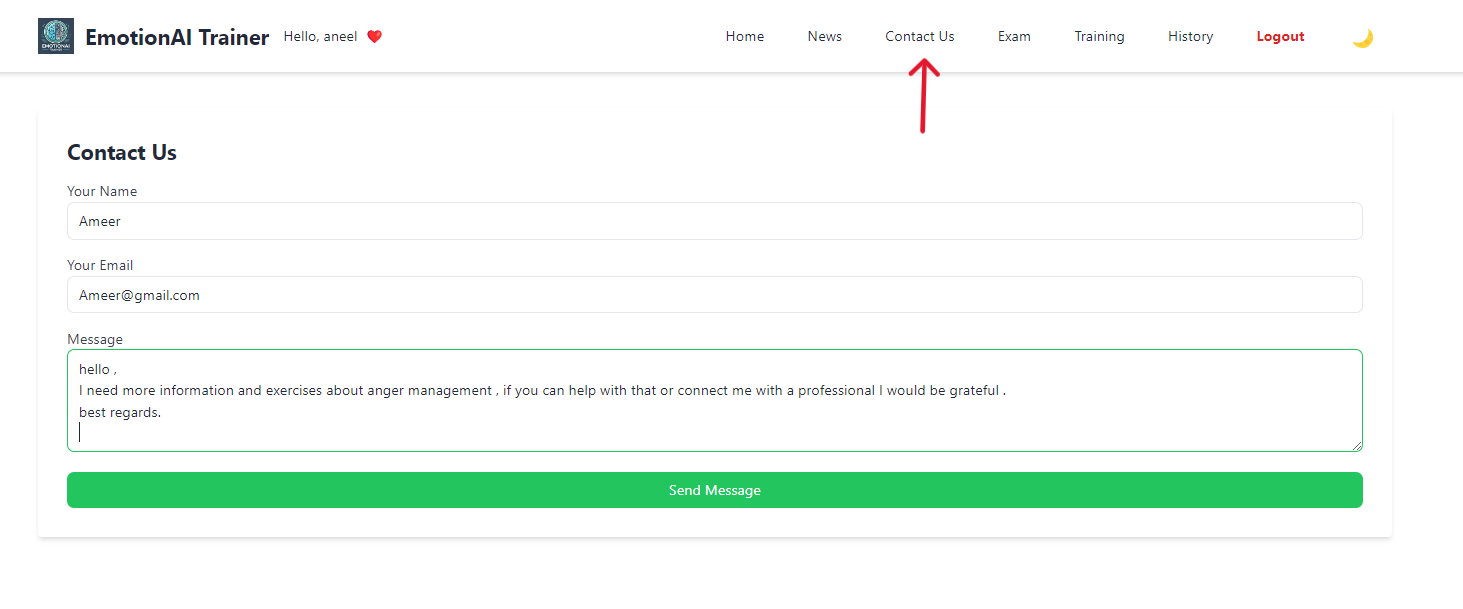
בנוסף ניתן לראות גם את חוות דעת ודירוגים של המשתמשים (לצורך הדגמה) .

**News page**



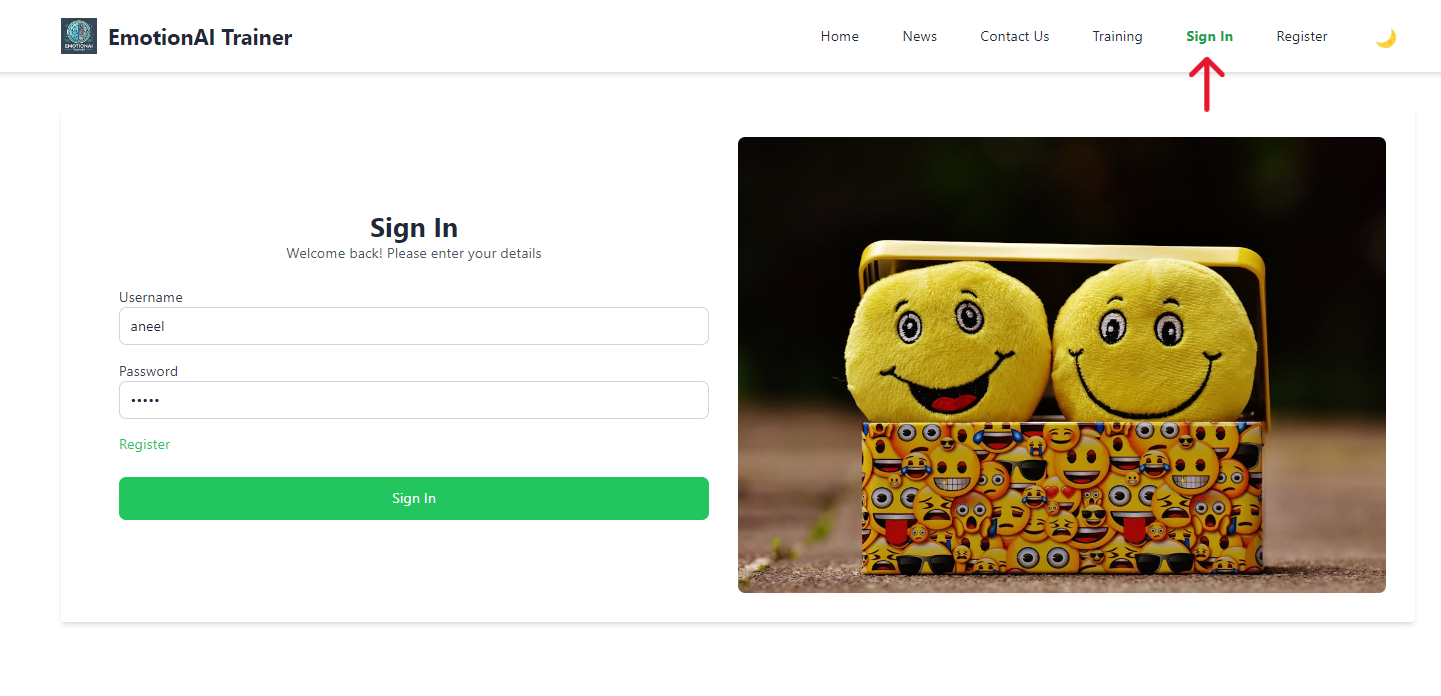
דף חדשות מעניינות בתחום האינטלגנציה הרגשית , עם האופציה לעשות חיפוש של נושאים ספיציפיים עליהם המשתמש מחפש מידע באינטרנט .

**Contact Us page**



דף יצירת קשר , כאן למשתמשים יש את האפשרות לשלוח הודעות ישירות למייל שלנו , שמיועד לטיפול בפניות ומתן שירות למשתמשים .

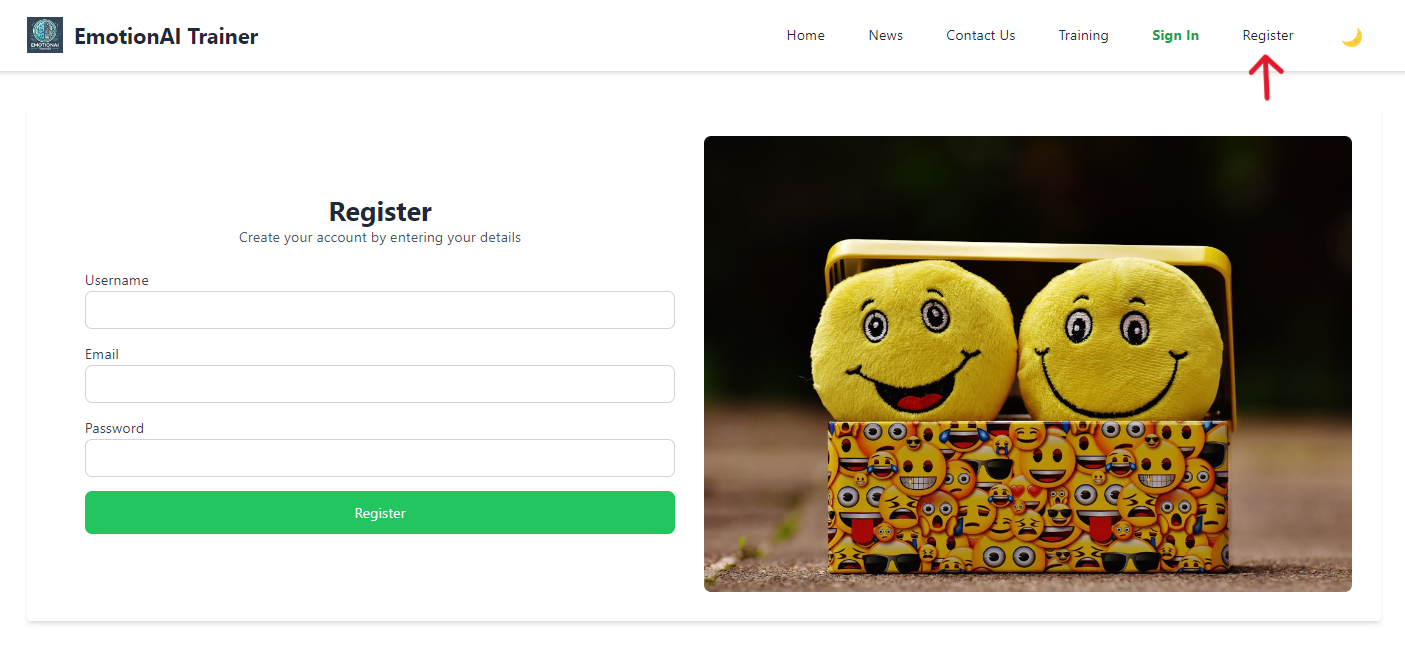
**Sign In page**



דף התחברות לאתר למשתמשים שכבר קיימים במערכת , כדי שיתאפשר למשתמשים לעשות את האבחון ולראות את התוצאות ואת דרך הטיפול שהם צריכים בהתאם לתוצאת האבחון שלהם , הם צריכים להיות מחוברים לאתר קודם לכן .

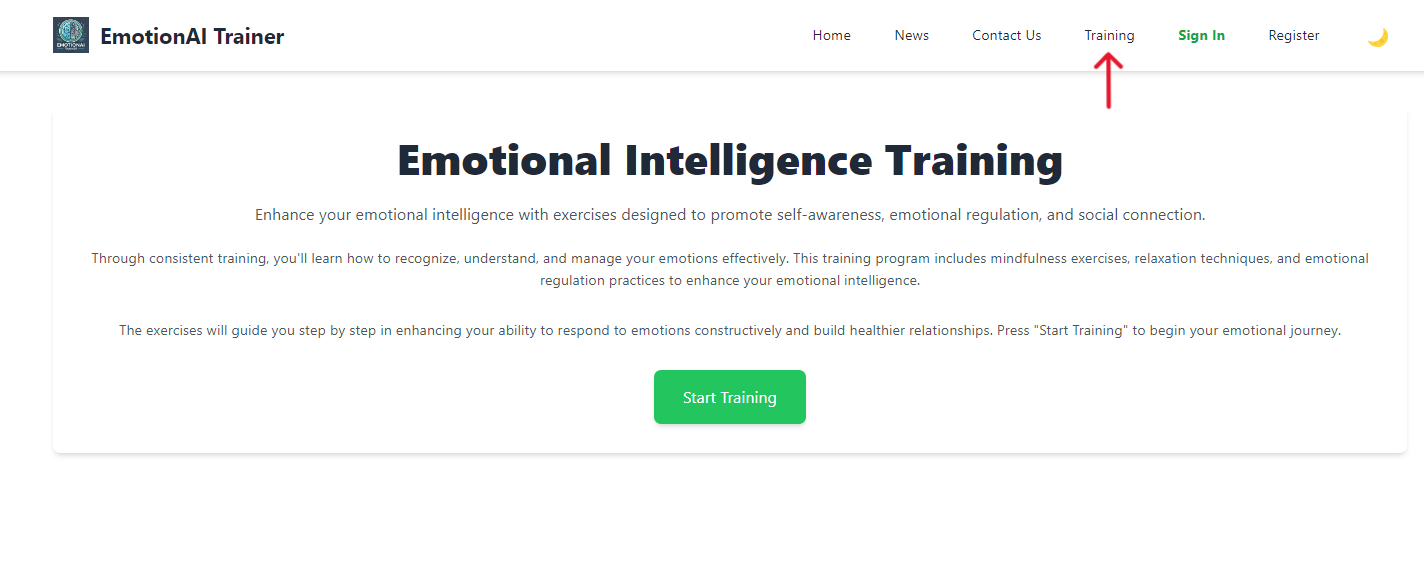
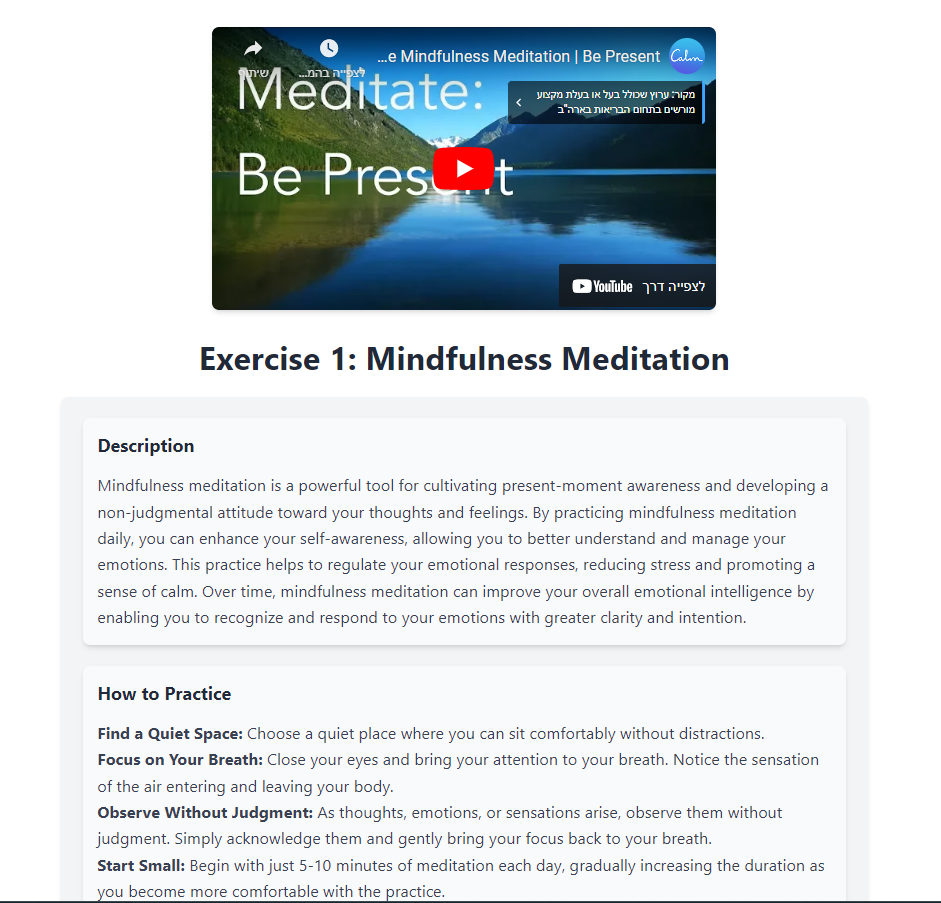
במידה ומשתמש לא רשום לאתר , יוכל להירשם על ידי לחיצה על Register , והאתר יפנה אותו לדף הרישום.

**Register page**



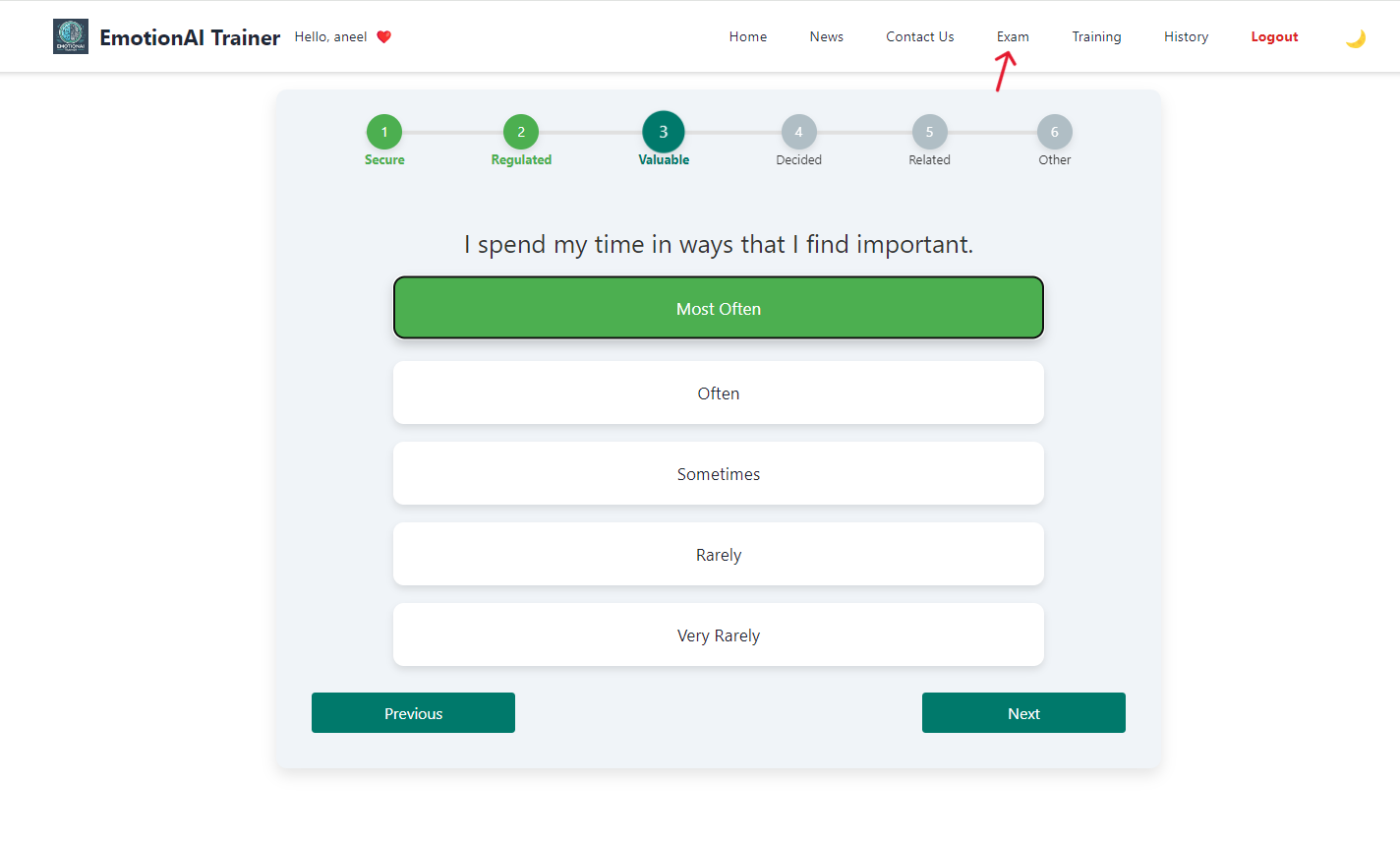
דף רישום משתמשים חדשים לאתר בו משתמשים יוצרים חשבונות חדשים ומזינים את פרטיהם למערכת .

**Training page**

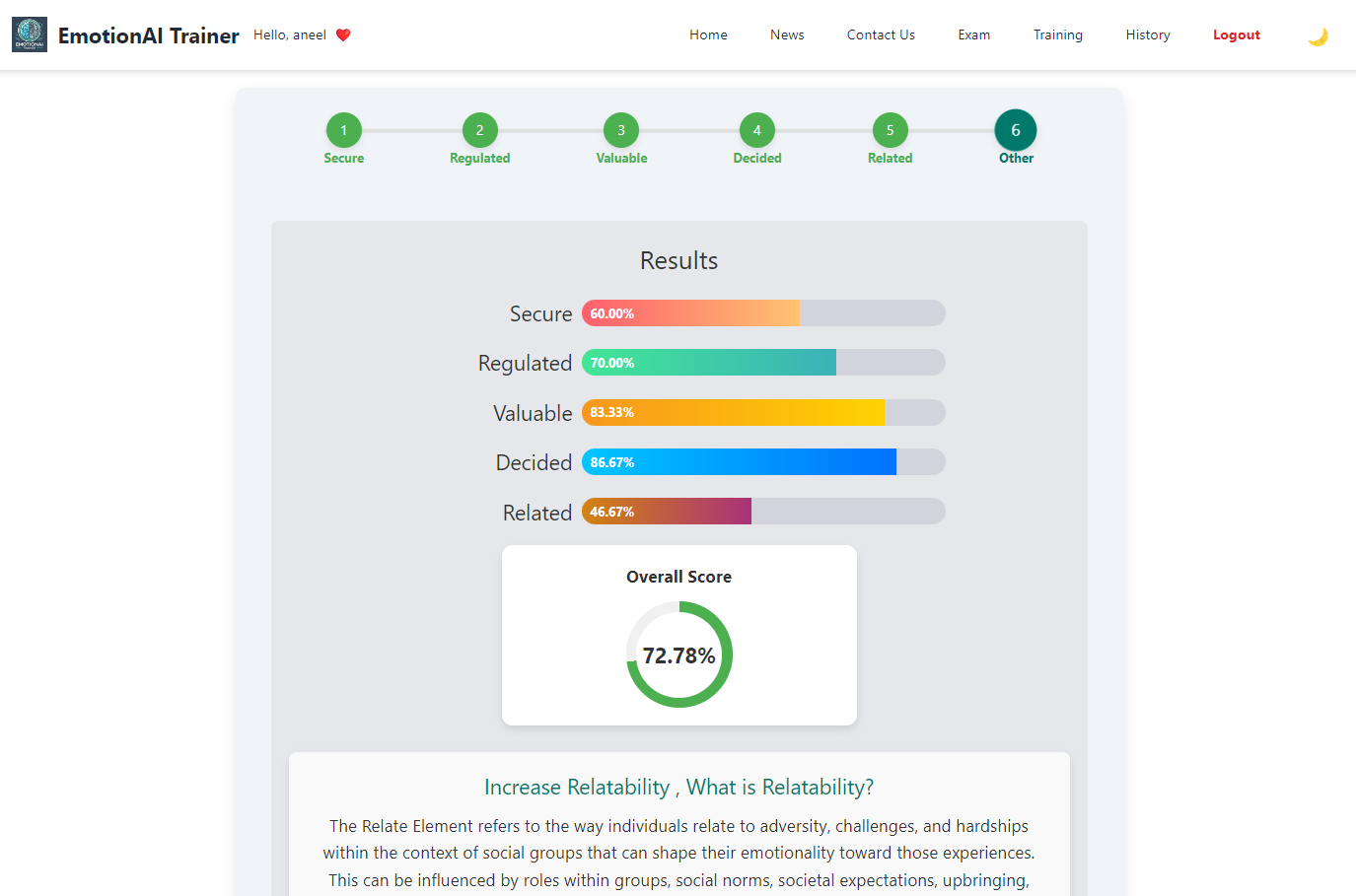
 

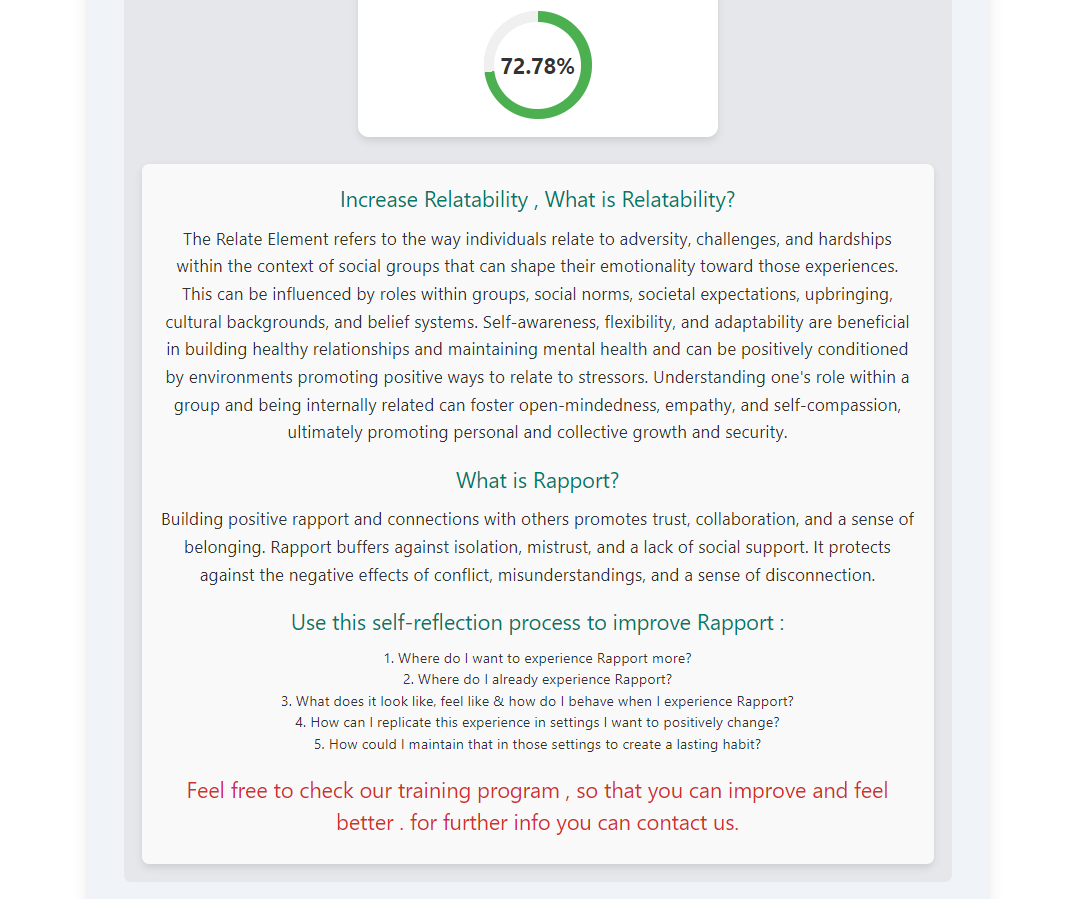
דף תרגילים להעצמה אישית עם הסברים מפורטים וקישורים לסרטוני הדרכה , דף זה גם זמין לשירותם של משתמשים אורחים ולא רשומים כדי להתרשם מהאתר ולקבל רעיון במה מדובר .

**Exam page**



דף האבחון של האתר , המשתמש עונה על שאלות לפי קטגוריות ( מקור השאלון הוא MindArch Health ) עם האפשרות לנווט בין השאלות , ובסוף יקבל את התוצאות של האבחון עם הסברים , הדרכות , ודרך לטפל באופן הבא :

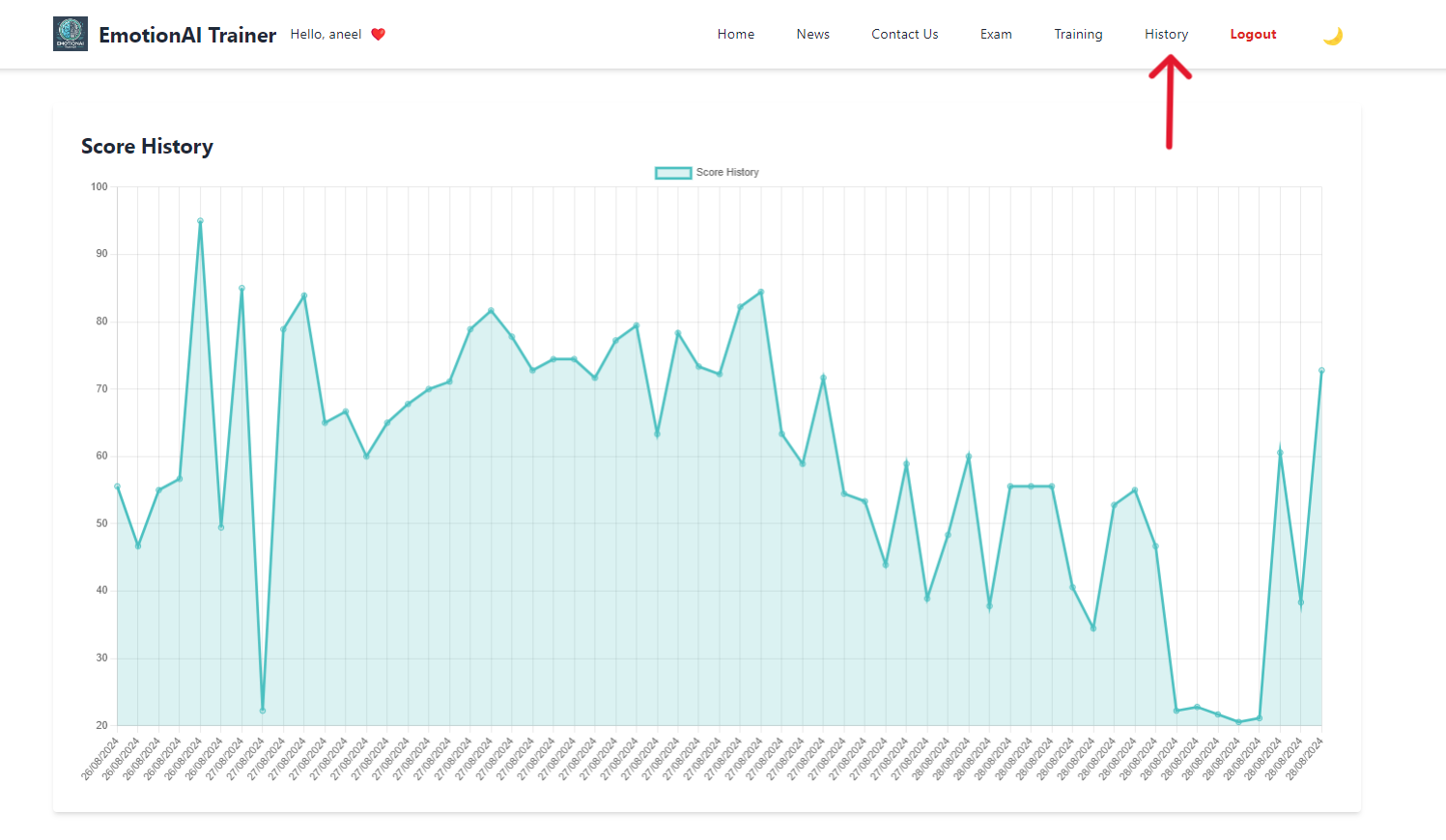




לאחר חישוב הציון בכל אחת מהקטגוריות מחשבים את הממוצע שלהם לציון הכללי . בהתאם לציון הנמוך ביותר אנחנו מציגים את המידע הרלוונטי , מה הוא צריך לשפר , איך לגשת לבעיה וכיצד לקבל רפלקציה לשיפור תכונות מסוימות שהמערכת חושפת בפניו וממליצה לו לעבוד עליהן בעקבות ניתוח התוצאות שקיבל .

את הממוצע של הציונים המערכת מחשבת ומוסיפה להיסטוריית התוצאות האישית שלו כך שהמשתמש יוכל לצפות בהן בהמשך .

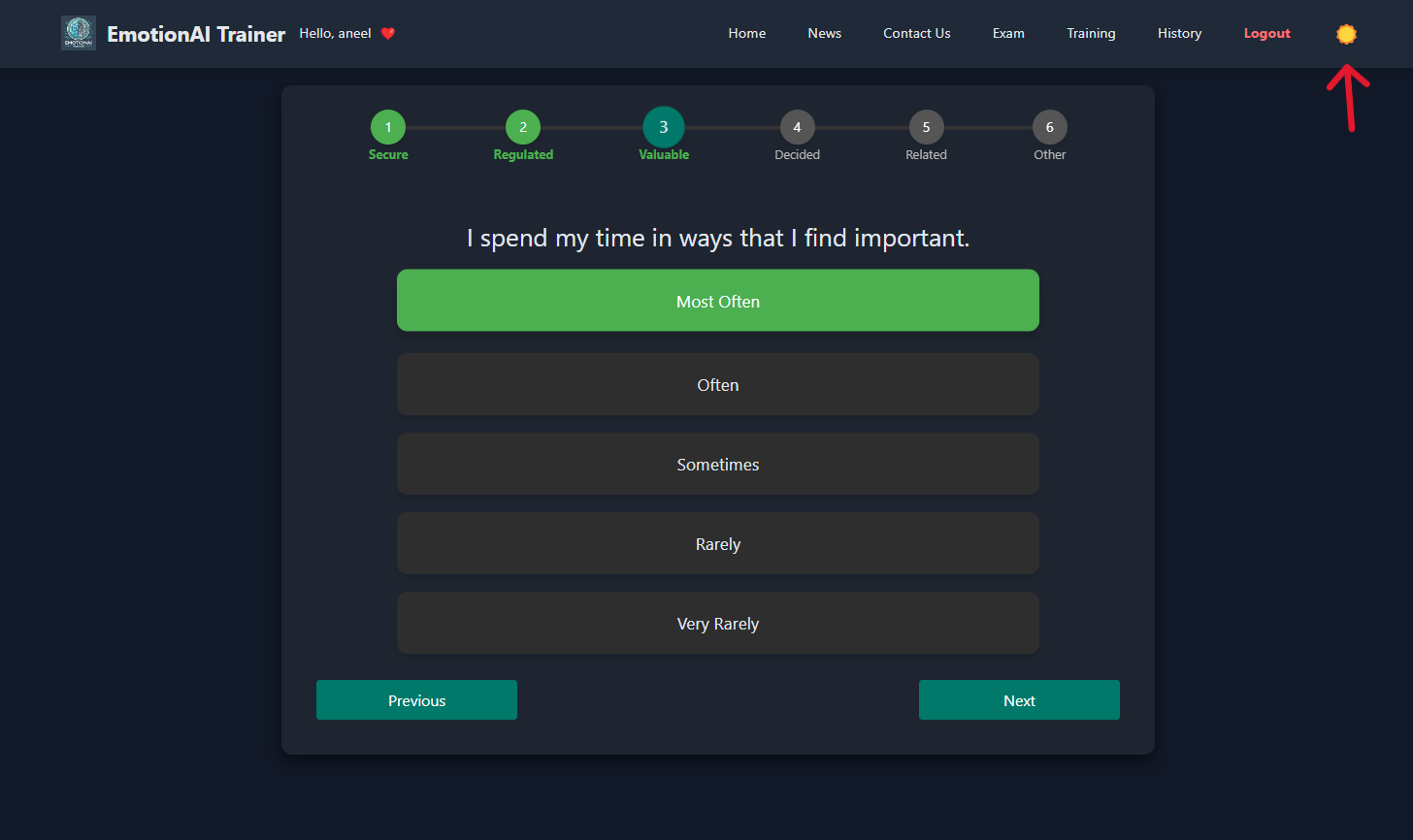
**History page**



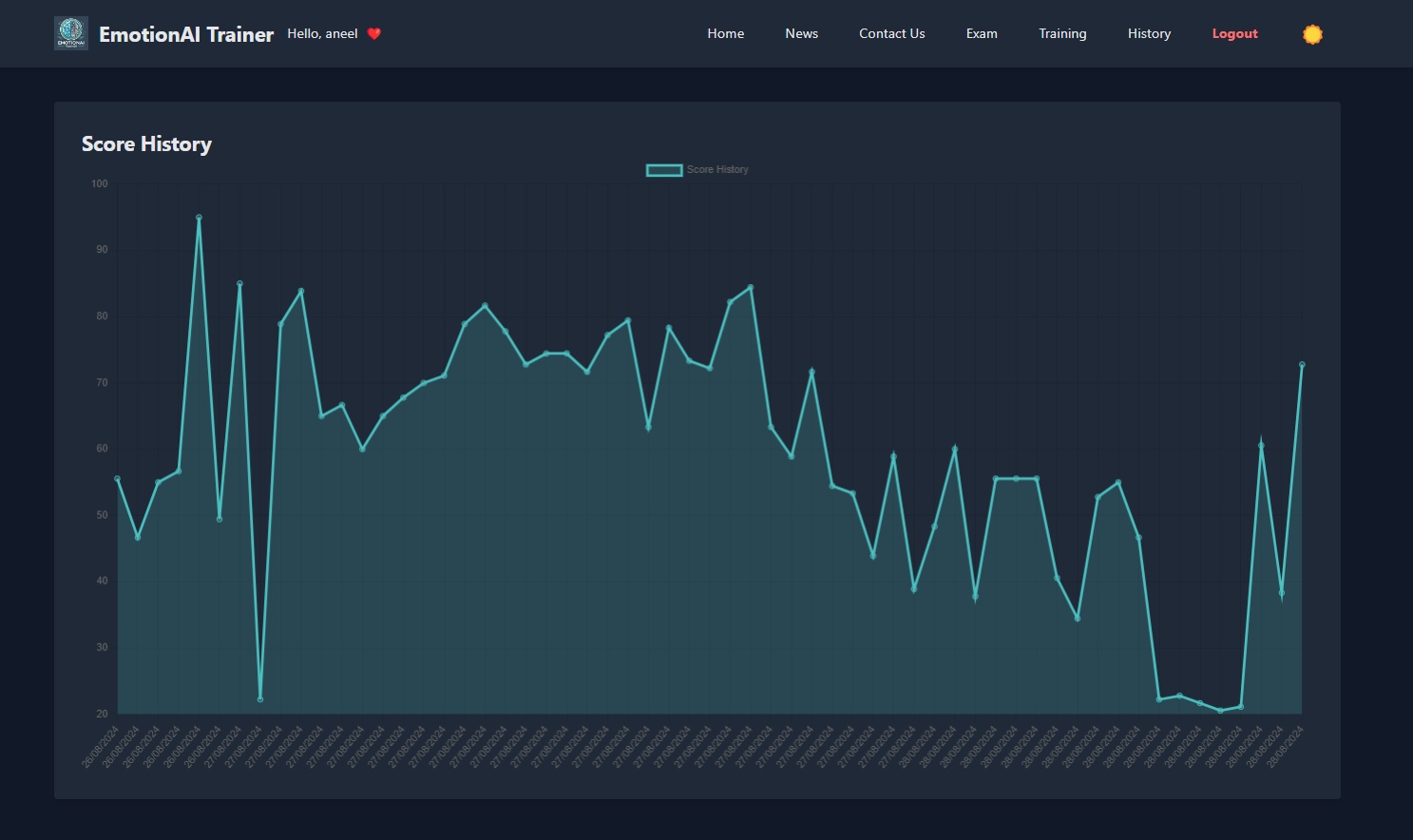
דף היסטורית התוצאות שבו המשתמש צופה בציונים שקיבל באבחונים השונים לפי תאריכים , כאן בעצם ניתן להסיק מסקנות ולקבל אינדיקציה על קצב ההתקדמות והשיפור שנעשה לאורך הזמן .

**Dark Mode**

ניתן להשתמש באתר גם ב "מצב לילה" ולעבור בין המצבים על ידי לחיצת כפתור , לבחירת המשתמש ולנוחיותו . דוגמאות להמחשה :







**באופן כללי :** האתר שלנו מאוד ידידותי ונוח לשימוש והדפים פשוט מסבירים את עצמם ואפשר לנווט ביניהם בקלות ויש חווית משתמש חיובית , ובמקרה של תקלה/שאלה כלשהי תמיד ניתן ליצור איתנו קשר .