**טכנולוגיות WEB – תיק פרויקט**

**קבוצה 12, פרויקט B29**

**חברי הצוות וחלוקת משימות:**

מהנדס מערכת – אנה גרמש

|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** |
| אנה גרמש –  337889125 | חלקות העבודה בין כל חברי הצוות, פיקוח עמידת בזמנים בביצוע משימות, אינטגרציה בין חברי הצוות, החלטה על הפונקציונליות של המערכת וניתוח דרישות, בדיקת עמידות המערכת בדרישות לפני ההגשה. | כל המשימות שתוכננו הושלמו. בניית המערכת נעשתה לפי הדרישות, מכילה פונקציונליות מלאה.  בוצעו תיקונים בהתאם להערות שקיבלנו בהצגת הפרויקט.  עבודה בצוות נעשה באווירה ידידותית עם חלוקת המשימות הוגנת שהתאימה לכל חברי הצוות. |
| אורן ברנובסקי - 311170450 | אחראי על בניית המערכת לפי הדרישות הוגדרו, אחראי על BUILD, תחזוקה והגשת הפרויקט לריפו/אתר. |
| מעיין עציוני - 211567342 | אחראית על DESIGN, יצירת תפריט, בחירת תבנית עיצוב, בדיקת המערכת תוך כדי העבודה - QA. |
| ליאור פניקשווילי - 319003943 | אחראי על בניית המערכת לפי הדרישות הוגדרו, אחראי על יצירתיות, פיתוח ויישום רעיונות של המערכת, דואג להביא פתרונות יצירתיים לבעיות שצפות תוך כדי העבודה. |

**דרישות:**

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Requirements | Type | NFR Category |
| 1 | The system will allow registration. | FR |  |
| 2 | The system will allow login. | FR |  |
| 3 | The system will allow logout. | FR |  |
| 4 | Login and logout will be done by entering email and password. | NFR | Execution Quality (Security) |
| 5 | The system will allow viewing daily progress. | FR |  |
| 6 | The system will allow viewing weekly progress. | FR |  |
| 7 | The system will allow viewing monthly progress. | FR |  |
| 8 | The system will contain habits. | FR |  |
| 10 | The system will allow adding/removing habits. | FR |  |
| 11 | The system will allow setting color of a habit. | FR |  |
| 13 | A habit contains a name. | NFR | Execution Quality (Usability) |
| 14 | The system will allow updating progress in a habit. | FR |  |
| 15 | The system will save user details. | FR |  |
| 16 | The user details that the system will save: user unique id, first name, last name, list of friends, list of hobbies. | NFR | Execution Quality (Security) |
| 17 | The system will allow adding friends. | FR |  |
| 18 | The system will display a list of friends. | FR |  |
| 19 | The system allows sharing a habit. | FR |  |
| 20 | Sharing the habit will be possible with friends. | NFR | Execution Quality (Usability) |
| 21 | The shared habit will show the progress of both friends. | NFR | Execution Quality (Usability) |

**תקציר הפרויקט**

המערכת שלנו עוסקת בניהול ומעקב אחר תחביבים בצורה חכמה ונוחה. המערכת כוללת דף נחיתה שמציג אחוזי הצלחה מהשבוע הקודם עבור כל תחביב, דף להוספת תחביבים ומעקב שבועי, דף חודשי המאפשר לראות התקדמות לאורך זמן ולבצע השוואה מול חברים ו דף הוספת חברים חדשים מתוך משתמשי המערכת. בעזרת ממשק מבוסס React ו-Tailwind CSS, בשילוב עם Node.js ו-Firebase לניהול הנתונים, המערכת מספקת חוויית משתמש חלקה ופשוטה לניהול יעיל של תחביבים ומטרות אישיות.

**רכיבי המערכת:**

1. **ממשק משתמש ידידותי**: המערכת מציעה למשתמשים ממשק פשוט ואינטואיטיבי המאפשר הרשמה והתחברות מהירה, ללא צורך בידע טכנולוגי מתקדם.
2. **התאמה אישית של תחביבים:** המשתמשים יכולים להתאים את התחביבים לפי הצרכים וההעדפות האישיות שלהם, כולל תחביבים כמו קריאה, ספורט, יצירה, ועוד.
3. **מעקב שבועי**: המערכת מאפשרת גישה לנתונים על תחביבים מהשבוע החולף, כולל אחוזי הצלחה והתקדמות.
4. **מעקב חודשי:** המשתמשים יכולים לצפות בהתקדמות החודשית שלהם ובמצב ההתקדמות בהשוואה לחברים.
5. **ניהול תחביבים ומעקב מתקדם**: המערכת מספקת כלים למעקב אחר תחביבים, קבלת תובנות והתקדמות בצורה חכמה. היא עוזרת למשתמשים לשפר את היעילות ולהתקדם בתחביבים שלהם בצורה ממוקדת ומדודה.
6. **ניהול חברים**: המערכת מספקת יכולת לנהל רשימת החברים של המשתמש, היא מאפשרת הוספה ומחיקה של החבר מרשימת החברים.

**פונקציונליות מרכזיות**

1. **הרשמה**: משתמשים חדשים יכולים להירשם למערכת באמצעות טופס פשוט שכולל פרטים בסיסיים. המערכת תיצור חשבון חדש.
2. **התחברות**: משתמשים רשומים יכולים להיכנס לחשבונם על ידי הזנת שם משתמש וסיסמה. המערכת תוודא את נכונות פרטי ההתחברות ותאפשר גישה לחשבון.
3. **דף נחיתה - מעקב יומי**: דף זה מציג את אחוז ההצלחות היומי, כולל גרפים וסטטיסטיקות להמחשת הצלחות בהשוואה לחברים.
4. **דף הוספת תחביבים ומעקב שבועי**: המערכת תאפשר למשתמשים להוסיף תחביבים חדשים, לעדכן את הפעולות שלהם בתחביבים הקיימים ולצפות בהתקדמות השבועית.
5. **דף חודשי**: דף המעקב החודשי מציג את ההתקדמות החודשית של כל תחביב, כולל אפשרות להשוות את ההתקדמות מול חברים (לתחביבים משותפים) ולהציג מגמות ארוכות טווח.
6. **דף הוספת חברים**: משתמש יכול למצוא, להוסיף או למחוק חבר מרשימת החברים שלו.
7. **התנתקות**: המשתמש יכול להתנתק מחשבונו בסיום השימוש במערכת. לאחר ההתנתקות, המערכת תסגור את הסשן הנוכחי ותעביר את המשתמש לדף הבית או לדף ההתחברות.

**נתוני תשתית וכלי פיתוח**

* **סביבת עבודה:**(Frontend) - Web using React
* **אחסון ומסד נתונים:**(Backend) – Firebase
* **ספריות ניווט**:(Backend) - Express Router
* **עיצוב:** (Frontend) - CSS and Tailwind
* **כלי למעקב גרסאות:** git

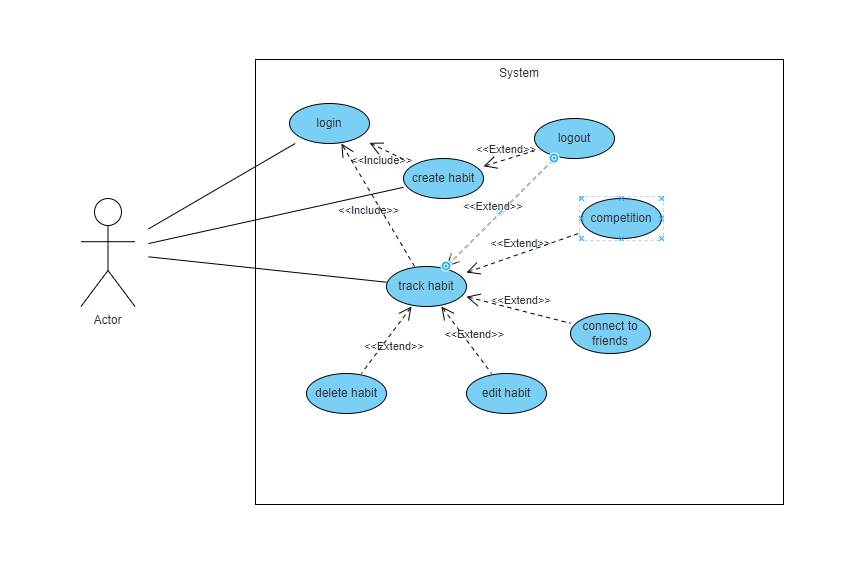
**קישורי מוצר**

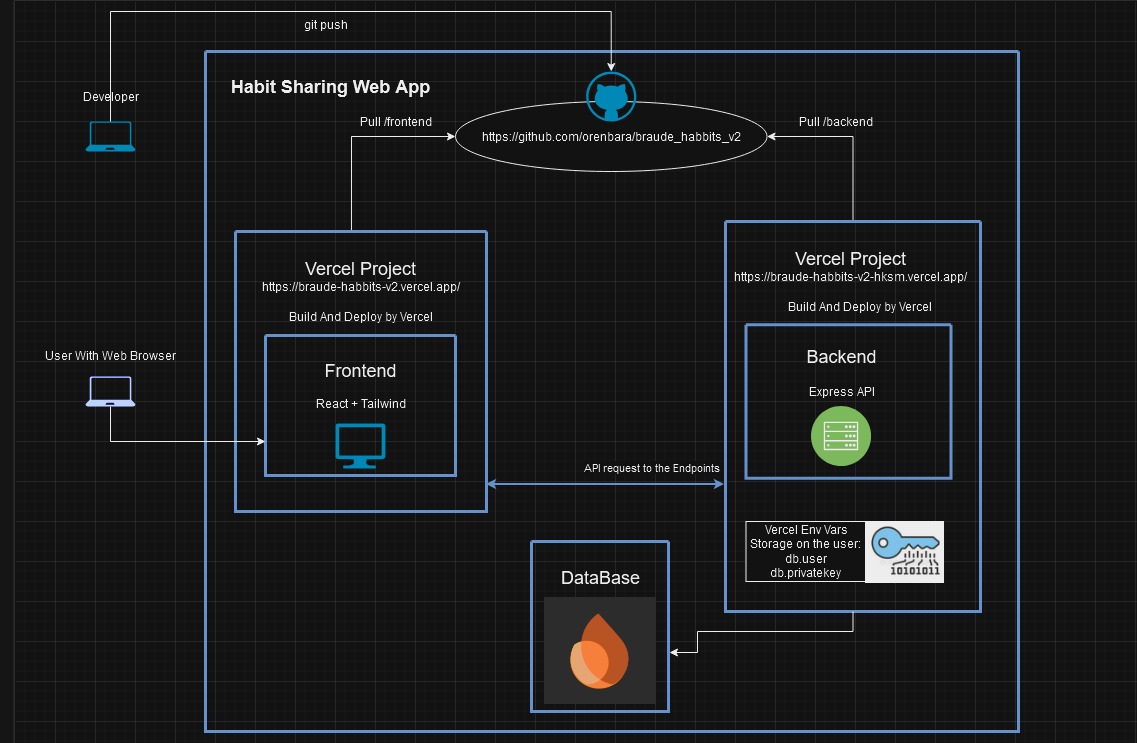
|  |
| --- |
| [קישור לתיקיית גיט ציבורי](https://github.com/orenbara/braude_habbits_v2/tree/oren) |
| [קישור לאתר](https://braude-habbits-v2.vercel.app/) |
| [MTW](https://www.morethanwallet.com/app/871) |

**ממשק בין חברי הצוות:**

במהלך הפרויקט, הצוות קיים תקשורת באמצעות פגישות ב-Discord וקבוצת WhatsApp. חילקנו את המשימות בין חברי הצוות בצורה מאורגנת והשלמנו את כל המשימות בהצלחה באמצעות שיתוף פעולה מתמשך ותקשורת אפקטיבית.

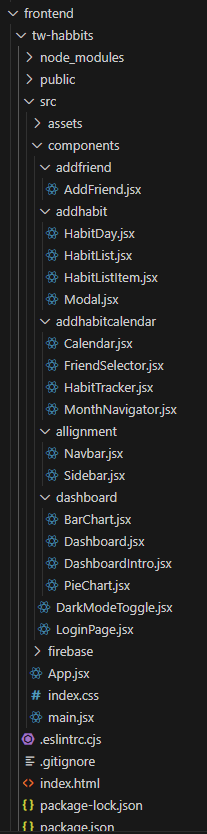
**תרשים USE CASE :**

**ארכיטקטורה של האתר:**



**הצגת מבנה סופי של האתר:**

**צד לקוח:**

**└── frontend**

**└── tw-habbits**

**├── README.md**

**├── index.html**

**├── src**

**│ ├── App.jsx**

**│ ├── assets**

**│ │ └── react.svg**

**│ ├── components**

**│ │ ├── DarkModeToggle.jsx**

**│ │ ├── LoginPage.jsx**

**│ │ ├── addfriend**

**│ │ │ └── AddFriend.jsx**

**│ │ ├── addhabit**

**│ │ │ ├── HabitDay.jsx**

**│ │ │ ├── HabitList.jsx**

**│ │ │ ├── HabitListItem.jsx**

**│ │ │ └── Modal.jsx**

**│ │ ├── addhabitcalendar**

**│ │ │ ├── Calendar.jsx**

**│ │ │ ├── FriendSelector.jsx**

**│ │ │ ├── HabitTracker.jsx**

**│ │ │ └── MonthNavigator.jsx**

**│ │ ├── allignment**

**│ │ │ ├── Navbar.jsx**

**│ │ │ └── Sidebar.jsx**

**│ │ └── dashboard**

**│ │ ├── BarChart.jsx**

**│ │ ├── Dashboard.jsx**

 **│ │ ├── DashboardIntro.jsx**

**│ │ └── PieChart.jsx**

**│ ├── firebase**

**│ │ └── config.js**

**│ ├── index.css**

**│ └── main.jsx**

**components** - מכיל רכיבי ממשק משתמש ניתנים לשימוש חוזר המשמשים בכל האפליקציה.

**addfriend**– מכיל רכיב ממשק משתמש הקשור להוספת חברים.

**addhabit** – מכיל רכיבי ממשק משתמש הקשורים להוספת תחביב חדש.

**addhabitcalendar** – מכיל רכיבי ממשק משתמש הקשורים לניהול תחביבים בצורה חודשית.

**alignment** – מכיל רכיבי ממשק משתמש אשר קיימים בכל הדפים וחלק מהמבנה הכללי של האתר.

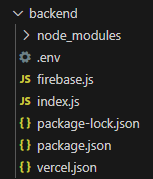
**dashboard** – מכיל רכיבי ממשק משתמש הקשורים למסך DASHBOARD.

**DarkModeToggle** - רכיב ממשק משתמש אחראי על dark/light mode.

**LoginPage** – רכיב ממשק משתמש אחראי על התחברות ורישום משתמשים.

Firebase –אחסון המידע של הפרויקט.

**צד שרת:**

├── backend

│ ├── firebase.js

│ ├── index.js

firebase – קובץ הגדרות של database

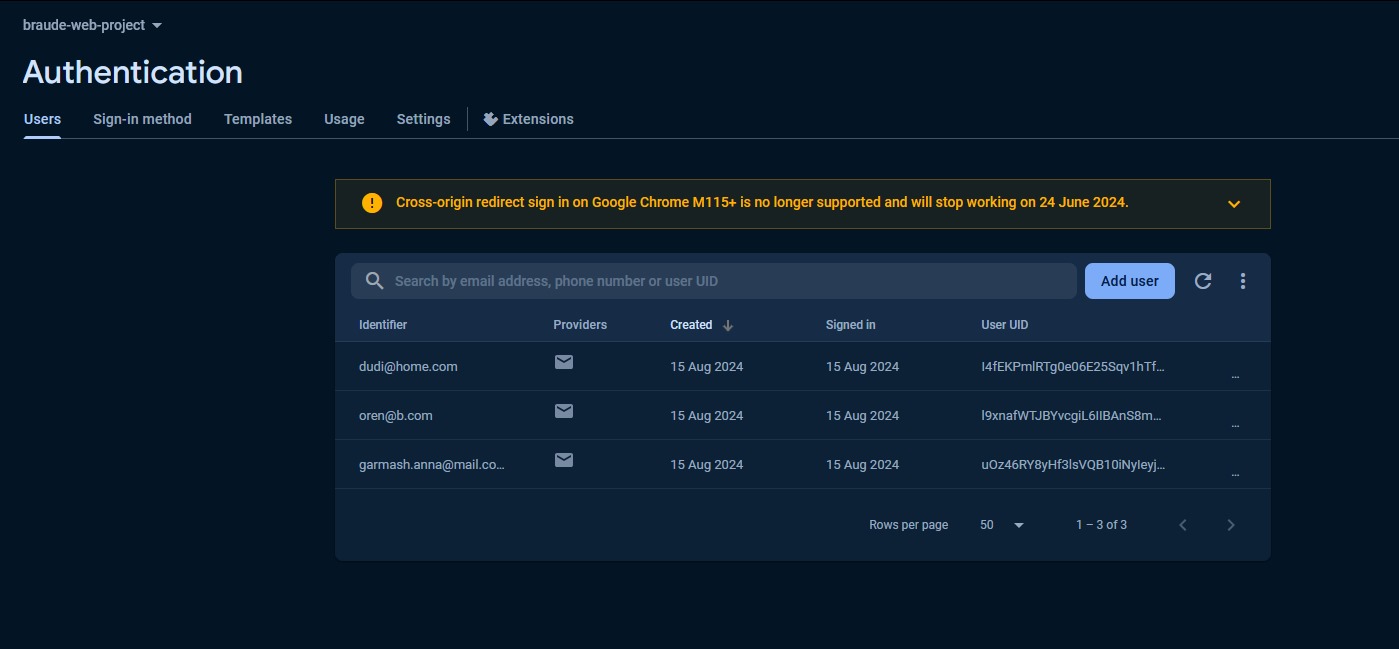
index – קובץ ניהול של API צד שרת כולל כל ENDPOINTS

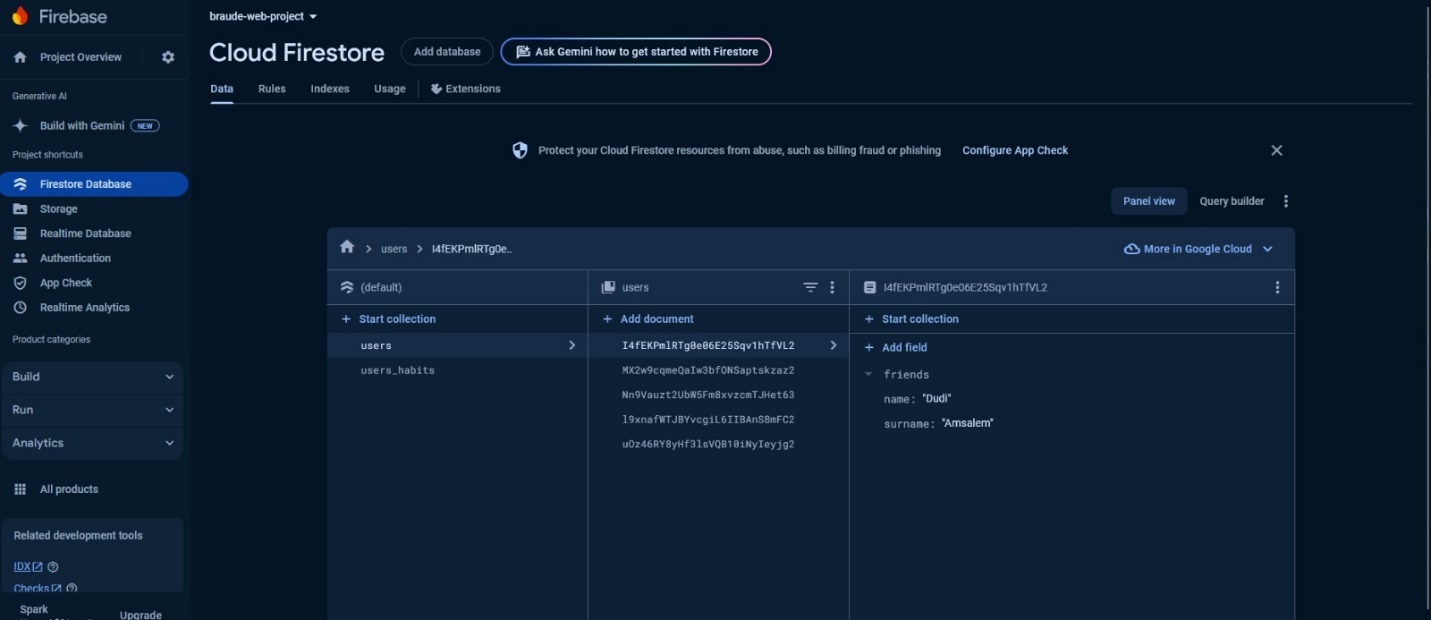
**תצוגת מבנה ה DB:**

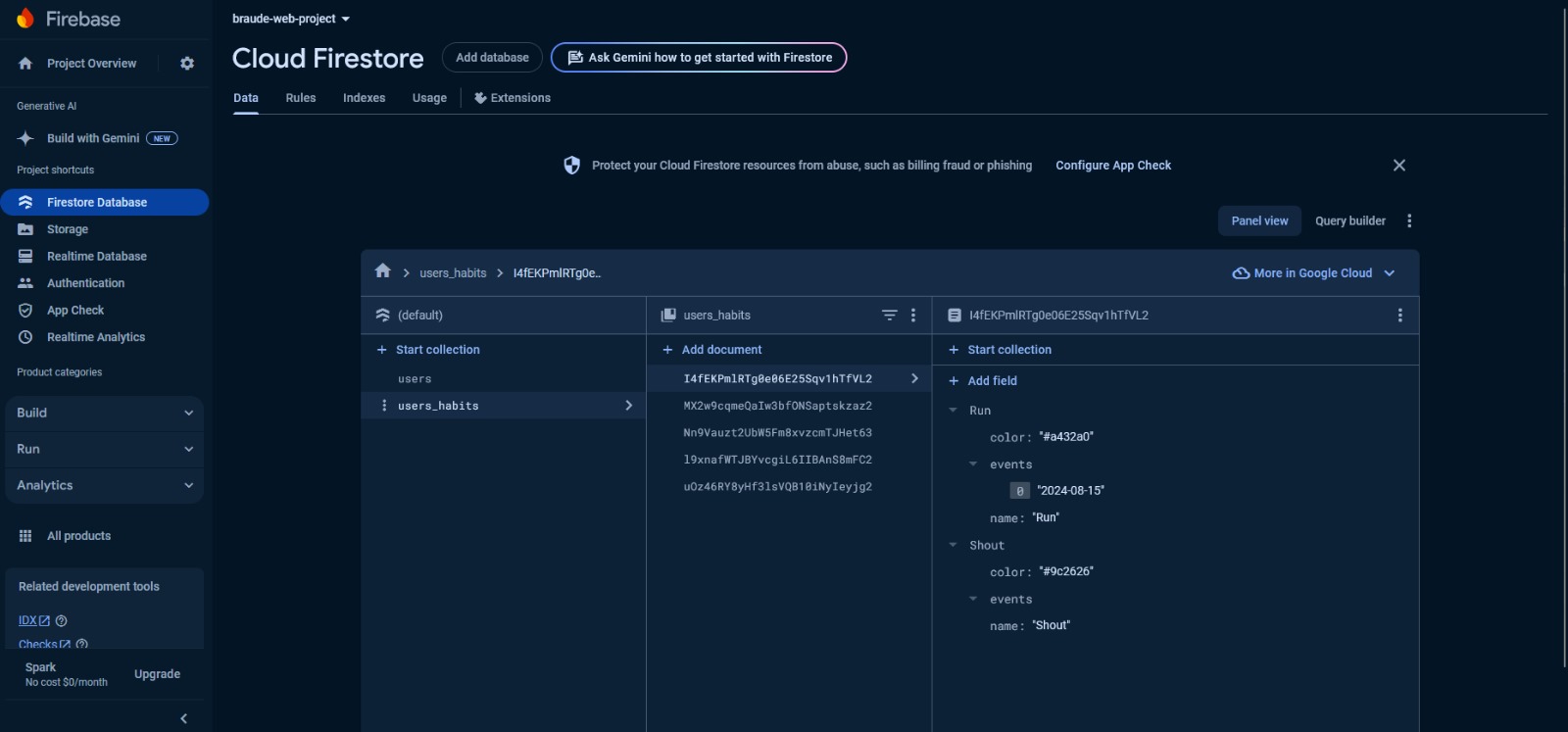
בפרויקט השתמשנו ב-firebase עבור אחסון הנתונים.

Firebase Authentication:

בפרויקט שלנו אנחנו עושים שימוש במוצר Authentication של Firebase. המוצר מספק כלים של התחברות ויצירת משתמשים לטובת אימות בלבד לאפליקציות frontend ואנחנו משתמשים בכלים אלו ללא תשלום.Firebase Authentication מאפשר יצירה של משתמשים מפלטפורמות שונות, ואנחנו בחרנו במשתמשים שמבצעים אימות עם Email and Password. לכל משתמש יש UID איחודי ואנחנו עושים בו שימוש לאחר אימות המשתמש במערכת.







**תיק למתכנת :**

**App.jsx:**

  useEffect(() => {

    const storedLoginState = localStorage.getItem("isLoggedIn");

    if (storedLoginState === "true") {

      setIsLoggedIn(true);

    }

    const storedDarkMode = localStorage.getItem("isDarkMode");

    if (storedDarkMode === "true") {

      setIsDarkMode(true);

    }

  }, []);

מבטיח שסטטוס ההתחברות של המשתמש והעדפת המצב dark/light mode ישמרו.

const renderView = () => {

פונקציה זו מרכזית לניווט של האפליקציה, ומבטיחה שהרכיב הנכון תוצג כאשר המשתמש אינטראקטיבי עם סרגל הצד.

  if (!isLoggedIn) {

    return (

      <LoginPage

        onLogin={handleLogin}

        setIsDarkMode={toggleDarkMode}

        isDarkMode={isDarkMode}

      />

    );

  }

מבטיח שמשתמשים בלתי מורשים לא יוכלו לגשת לתוכן המרכזי של האפליקציה ויועברו לעמוד ההתחברות.

**LoginPage.jsx:**

  const handleSignUp = async (e) => {

רושמת משתמש חדש במערכת ושולחת את פרטיו ל-DB.

  const handleSubmit = async (e) => {

מבצעת תהליך התחברות של משתמש קיים דרך Firebase Authentication ושומרת את מזהה המשתמש.

**DarkModeToggle.jsx:**

  const toggleMode = () => {

מחליף בין מצב dark mode למצב light mode ומעדכן את הגדרות ב-localStorage.

**Dashboard.jsx:**

    const calcPercentage = () => {

מחשב את אחוז ההצלחה היומי של המשתמש לפי התקדמות היומית.

      const friendsPercents = () => {

מביא נתונים על אחוזי ההצלחה היומי של חברים.

**HabitTracker.jsx:**

  const fetchHabits = (id) => {

  const fetchFriends = () => {

מושך את תחביבים של המשתמש ואת נתוני החברים לפי מזהה המשתמש או מזהה החבר הנבחר.

  const prevMonth = () => {

  const nextMonth = () => {

  <MonthNavigator currentMonth={currentMonth} prevMonth={prevMonth} nextMonth={nextMonth} />

מנהל את הניווט בין חודשים בלוח השנה.

**Calendar Interactions**: מציג ומאפשר עריכה של לוח השנה של המשתמש, מציג את לוח השנה של חבר במצב לקריאה בלבד.

**Friend Selector**: תפריט נפתח לבחירת חבר, שמציג את לוח השנה שלו בהתאמה להרגל הנבחר.

**HabitListItem.jsx:**

    const switchActive = (index) => {

משנה את סטטוס הפעילות של יום, מעדכנת את הסטטוס בשרת ומוסיפה או מסירה את האירוע מהמסד נתונים.

    const days = useMemo(

מחשב ושומר את רשימת הימים של השבוע הנוכחי.

**HabitList.jsx:**

    const fetchHabits = () => {

פונקציה שמבצעת בקשת fetch לשרת כדי למשוך את רשימת ההרגלים של המשתמש ומעדכנת את ה-state.

    const handleHabitSubmit = (newHabit) => {

פונקציה שמוסיפה תחביב חדש לרשימה וסוגרת את החלונית, ומבצעת משיכת התחביבים המעודכנים.

    const handleDeleteHabit = (index) => {

פונקציה שמוחקת תחביב מהרשימה על בסיס אינדקס ושולחת בקשת מחיקה לשרת.

**Add Habit Modal**: שימוש ב-Modal לקבלת קלט מהמשתמש (כותרת וצבע) ולניהול הוספת הרגל חדש.

**AddFriend.jsx:**

  const [searchQuery, setSearchQuery] = useState('');

מנהל את הקלט לחיפוש לסינון חברים.

  const fetchUsers = async () => {

מביא ומעדכן את רשימת כל המשתמשים לצורך הוספת חברים.

  const handleAddRemoveFriend = async (friend) => {

מוסיף או מסיר חבר מרשימת החברים מעדכן את מצב myFriends.

  const filteredFriends = friendsList

מסנן את רשימת החברים לפי השאילתה חוץ ממשתמש הנוכחי.

**פירוט של API:**

* **GET /add\_user:**

פונקציה: מוסיף משתמש חדש לקולקציית users ב-Firestore.

פרמטרים: id (מזהה המשתמש), name (שם פרטי), surname (שם משפחה).

תשובות: מחזיר הצלחה אם המשתמש נוסף בהצלחה, או שגיאה אם חסרים פרמטרים או אם התהליך נכשל.

* **GET /add\_habit:**

פונקציה: מוסיף תחביב חדש למשתמש בקולקציית users\_habits ב-Firestore.

פרמטרים: id (מזהה המשתמש), habitName (שם התחביב), color (צבע ההרגל).

תשובות: מחזיר הצלחה אם התחביב נוסף בהצלחה או שגיאה אם חסרים פרמטרים או אם התהליך נכשל.

* **GET /add\_event\_to\_habit:**

פונקציה: מוסיף אירוע (תאריך ביצוע תחביב) חדש לתחביב מסוים של המשתמש.

פרמטרים: id (מזהה המשתמש), habitName (שם התחביב), habitEvent (תאריך האירוע).

תשובות: מחזיר הצלחה אם האירוע נוסף בהצלחה או שגיאה אם חסרים פרמטרים או אם התהליך נכשל.

* **GET /delete\_habit:**

פונקציה: מוחק הרגל מסוים של המשתמש מקולקציית users\_habits.

פרמטרים: id (מזהה המשתמש), habitName (שם התחביב).

תשובות: מחזיר הצלחה אם התחביב נמחק בהצלחה או שגיאה אם חסרים פרמטרים, התחביב לא נמצא או התהליך נכשל.

* **GET /delete\_event\_from\_habit:**

פונקציה: מוחק אירוע מסוים מהתחביב של המשתמש.

פרמטרים: id (מזהה המשתמש), habitName (שם התחביב), eventDate (תאריך האירוע).

תשובות: מחזיר הצלחה אם האירוע נמחק בהצלחה או שגיאה אם חסרים פרמטרים, האירוע לא נמצא או התהליך נכשל.

* **GET /get\_user\_personal\_data:**

פונקציה: מקבל את הנתונים האישיים של משתמש מסוים.

פרמטרים: id (מזהה המשתמש).

תשובות: מחזיר את נתוני המשתמש או שגיאה אם המשתמש לא נמצא או אם יש בעיה בתהליך.

* **GET /get\_user\_habits:**

פונקציה: מקבל את רשימת התחביבים של משתמש מסוים.

פרמטרים: id (מזהה המשתמש).

תשובות: מחזיר את רשימת התחביבים של המשתמש או שגיאה אם המשתמש לא נמצא או אם יש בעיה בתהליך.

* **GET /get\_specific\_habit:**

פונקציה: מקבל את פרטי התחביב הספציפי של משתמש.

פרמטרים: id (מזהה המשתמש), habitName (שם ההרגל)

תשובות: מחזיר את פרטי התחביב או שגיאה אם ההרגל או המשתמש לא נמצאו.

* **GET /get\_all\_users:**

פונקציה: מקבל את רשימת כל המשתמשים בקולקציית users.

תשובות: מחזיר את רשימת המשתמשים או שגיאה אם לא נמצאו משתמשים או אם יש בעיה בתהליך.

* **GET /add\_friend:**

פונקציה: מוסיף חבר לרשימת החברים של המשתמש.

פרמטרים: userId (מזהה המשתמש), friendId (מזהה החבר).

תשובות: מחזיר הצלחה אם החבר נוסף בהצלחה או שגיאה אם יש בעיות בתהליך.

* **GET /remove\_friend:**

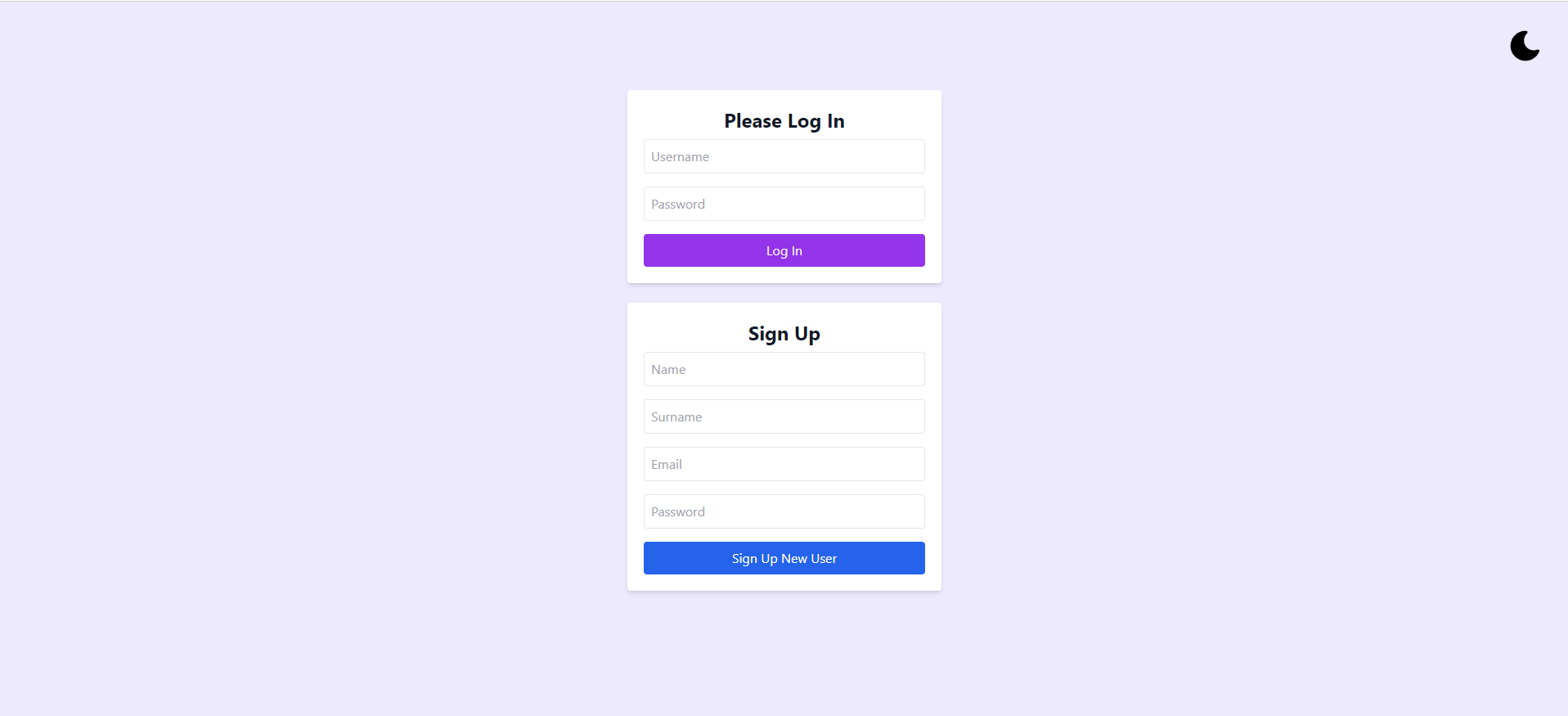
פונקציה: מסיר חבר מרשימת החברים של המשתמש.

פרמטרים: userId (מזהה המשתמש), friendId (מזהה החבר).

תשובות: מחזיר הצלחה אם החבר הוסר בהצלחה או שגיאה אם החבר לא נמצא או אם יש בעיה בתהליך.

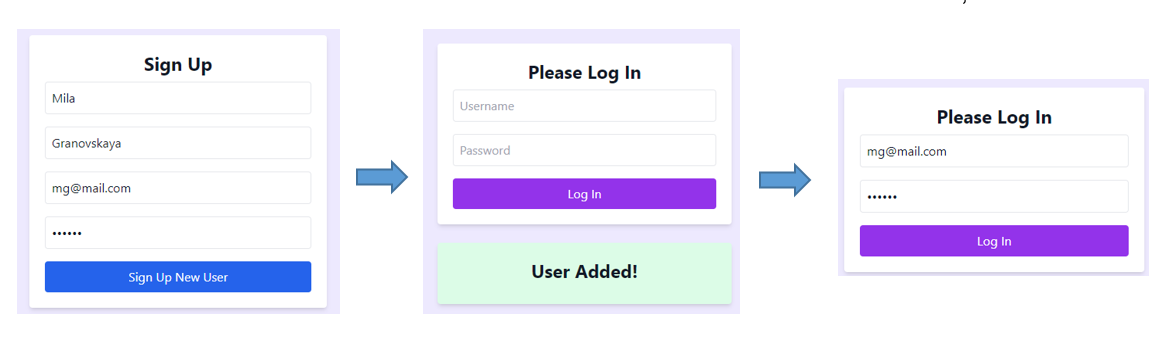
**תיק למשתמש :**

**עמוד התחברות והרשמה:**



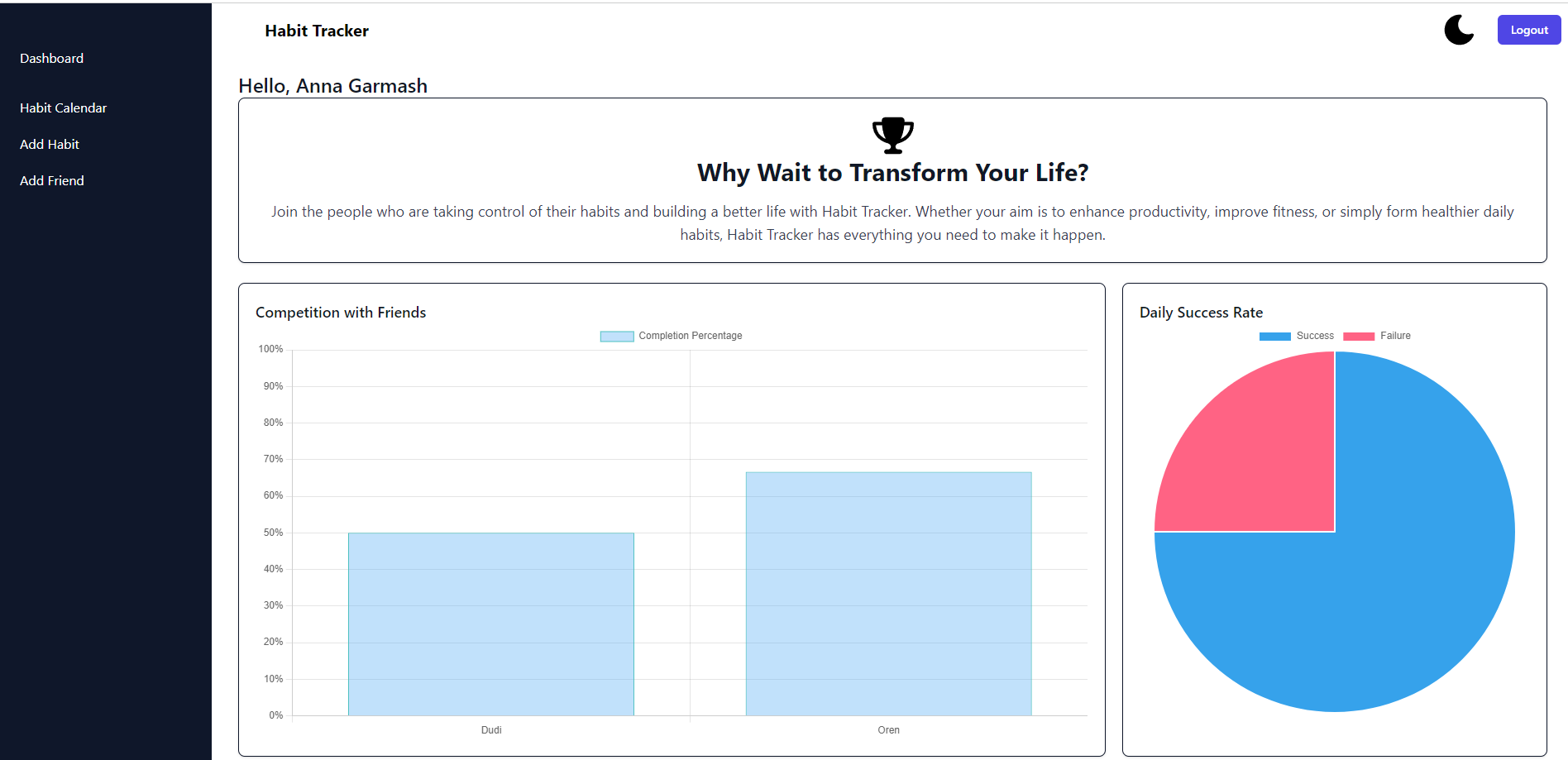
המסך הראשון שנפתח למשתמש יהיה מסך LOGIN. ניתן להתחבר עם משתמש קיים או ליצור משתמש חדש.

על מנת ליצור משתמש חדש יש ללחוץ על הכפתור “Sing Up New User” ולהכניס שם פרטי, שם משפחה, מייל וסיסמה.

לצורך הבדיקה ניתן ליצור משתמש חדש להתחבר עם המשתמש שיצרתם.

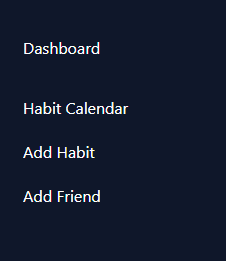
בלחיצה על כפתור “Log in” המשתמש מועבר למסך הראשי DASHBOARD.

**Dashboard**:

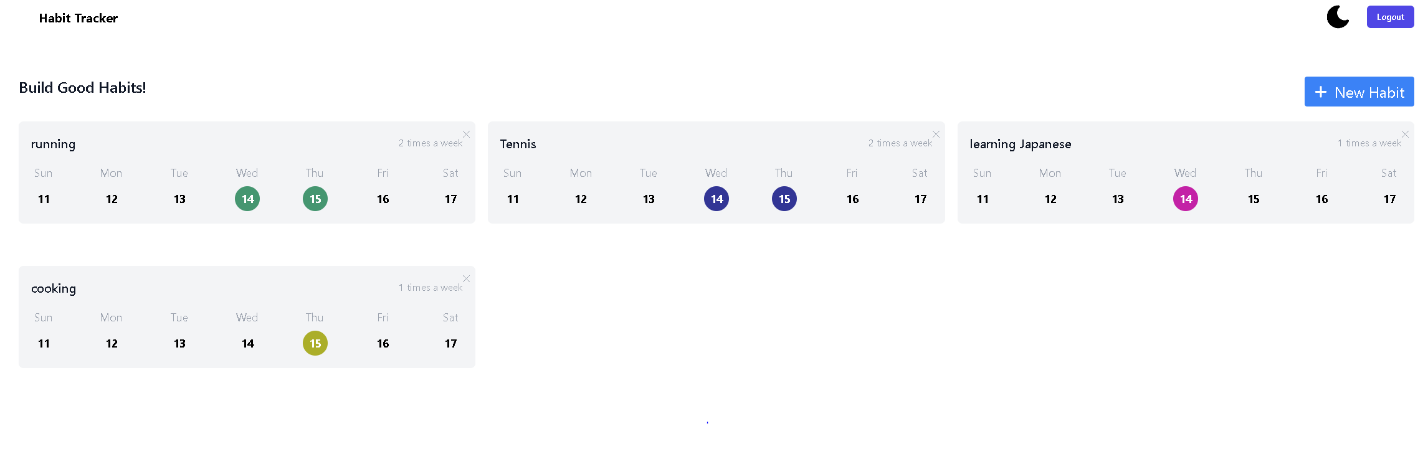


בעמוד Dashboard מוצג הסבר קצר על האתר עצמו, הצגת נתוני הצלחה יומיים למשתמש ונתוני הצלחה יומיים של החברים שלו.

בכל עת שימוש באתר, אפשר ללחוץ על אחד הכפתורים ב-Sidebar ולעבור לדפים האחרים.



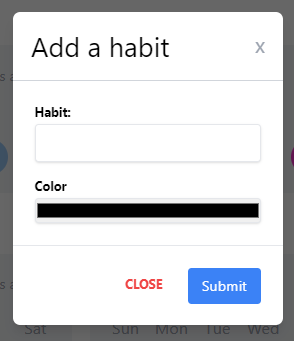
**עמוד מעקב שבועי והוספת/הסרת תחביבים Add Habit :**



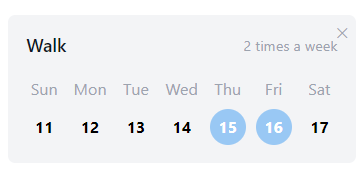
בעמוד זה ניתן לעקוב אחרי התקדמות שבועית, להוסיף תחביבים חדשים, למחוק תחביבים קיימים, לסמן אירועים בתחביבים שבוצעו או לבטל אירועים שסומנו בטעות.

על מנת לסמן את התאריך יש ללחוץ על התאריך המבוקש והוא יצבע בצבע שבחרתם. על מנת לבטל את הסימון, יש ללחוץ עוד פעם על התאריך.

כדי ליצור תחביב חדש יש ללחוץ על כפתור “New habit”. ניתן להכניס שם של תחביב ולבחור את הצבע הרצוי. עם לחיצת הכפתור “Submit” יבצר תחביב חדש, עם לחיצת הכפתור “Close” החלון ייסגר ללא שינוי.



למען מחיקת התחביב הקיים, יש ללחוץ על X בפינה הימנית למעלה.



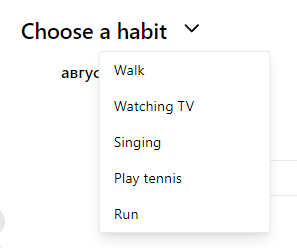
**עמוד מעקב חודשי והשוואה מול חברים Habit Calendar:**

A white background with black text

Description automatically generated

בעמוד זה ניתן לעקוב אחרי התקדמות חודשית בתחביב הנבחר, לראות את רשימת החברים ולהסתכל על התקדמותם.

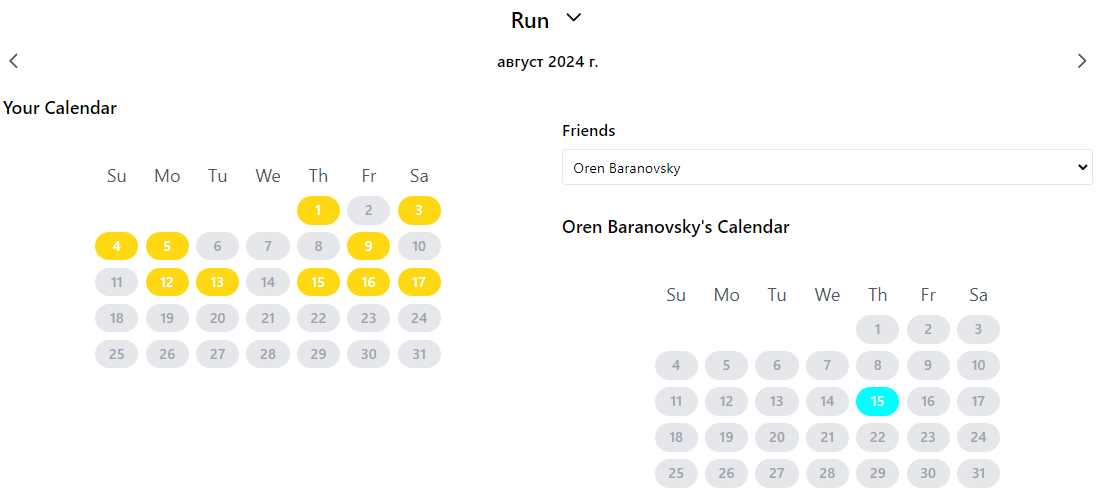
כדי לבחור תחביב, יש ללחוץ על החץ ליד “Choose a habit”, תפתח רשימה נפתחת עם כל התחביבים של המשתמש. באותו אופן ניתן גם לבור את החבר.



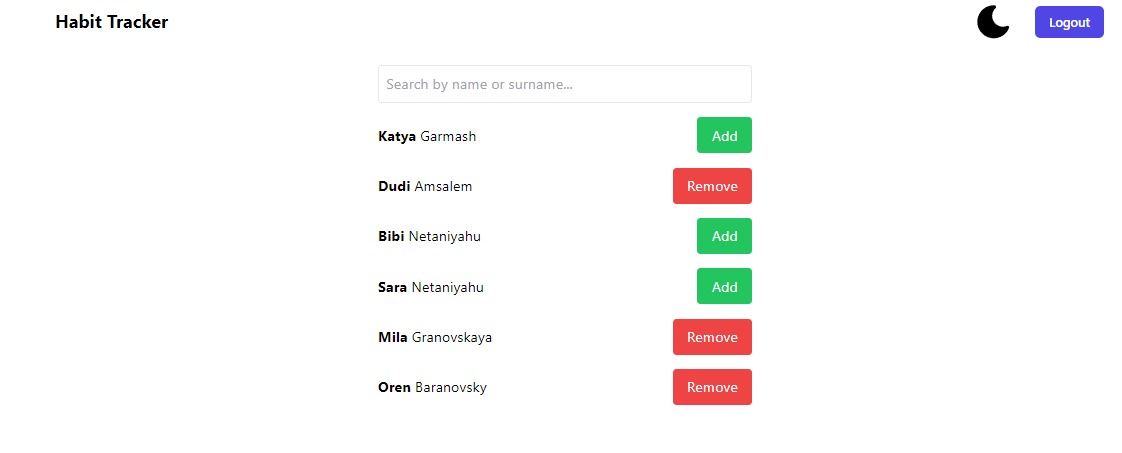


אם התחביב הנבחר לא קיים אצל חבר כלשהו, תופיע הודעה מתאימה שמודיעה על כך.

אם התחביב הנבחר קיים אצל החבר, יופיע יומן חודשי שלו ובו ניתן לצפות בהתקדמות החודשית של החבר בתחביב הנ"ל. ביומן האישי ניתן לבצע שינוי באירועים אך היומן של החבר הוא לצפייה בלבד ולא ניתן לשינוי.



**עמוד הוספת או הסרת חברים Add Friend:**



בעמוד זה מוצגים כל המשתמשים הרשומים במערכת. אם המשתמש כבר נמצא ברשימת החברים של המשתמש המחובר, נראה כפתור “Remove” ליד השם. אם המשתמש עדיין לא ברשימת החברים, נראה כפתור “Add”.

בכל עת שימוש באתר אפשר להוסיף או להסיר חברים וכך לראות את השינוי גם ב-Dashboard.

בעמוד קיימת אופציה חיפוש חבר. משתמש יכול להתחיל להקליד שם פרטי, שם משפחה או אפילו אותיות הקיימות בשם ושדה החיפוש יתחיל להתאים תצוגה לפי שאילתת החיפוש.

