# **יש לכלול בתחילת מסמך הפרויקט את דף השער הבא:**

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

הגשת פרויקט

Be Healthy <B18> group 3

| שם חברת הצוות | תז |
| --- | --- |
| אמנה אבו סמרה | 324159367 |
| גנט אבראהים | 212269617 |
| זינב עבד אלגני | 323963405 |
| שאדי חגאזי | 324847110 |

The "Be Healthy" project includes a comprehensive metrics feature designed to help patients track and manage their health more effectively. Here’s a breakdown of the key components and functionalities of the metrics feature:

Functionalities :

* **Appointment Scheduling**: Book medical appointments with a choice of doctor, date, and reason for the visit.
* **View and Manage Appointments**: Access and manage existing appointments.
* **Access to Medical Test Results**: View and display test results in graphs
* **Online Store**: Purchase medications, vitamins, and dietary supplements.
* **Track Patient Medical History**: Monitor patients' medical history.

Technologies:

1. Frontend:

-React.js: For building user interfaces and handling the state of different components.

-React Router: For managing routing between different pages.

-Tailwind CSS: For styling the application with a focus on a responsive and modern UI.

- JavaScript: Core programming language for developing the frontend logic.

2. Backend:

-Firebase: For backend services like authentication, database, and hosting.

3. Database:

-Firestore (Firebase): For storing user data, appointments, health metrics, and other dynamic content.

4. Styling:

- CSS Files: Organized into different files for modular and maintainable styling (e.g., Contact.css, Features.css, Login.css).

- Tailwind CSS Classes: Used for a more efficient and responsive design.

<https://github.com/AmniAbo/Web_group3.git>

[Be Healthy (web-group3-iota.vercel.app)](https://web-group3-iota.vercel.app/)

[MoreThanWallet](https://www.morethanwallet.com/app/893)

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

הגשת פרויקט

להגשה עד 29.8.24 בשעה 23:59

1. עליכם להמשיך את בניית האתר לפי האלמנטים המתקדמים שלמדתם

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

| שם חבר הצוות | משימות שהוקצו | משימות שהושלמו |
| --- | --- | --- |
| אמנה אבו סמרה | מבנה פריסה (Deployment),הכנת הצגה, עדכון קומפוננטות , הערות בקוד | הושלם |
| גנט אבראהים | יצירת ארכיטקטורה , דרישות פונקציונלית ולא פונקציונלית | הושלם |
| זינב עבד אלגני | תיק לקוח ,דיאגרמת ארכיטקטורה ,דיאגרמת Use Case | הושלם |
| שאדי חגאזי | תיק מתכנת, יצירת ארכיטקטורה , מצב כהה | הושלם |

2. הציגו רשימת דרישות פונקציונליות ולא פונקציונליות (בנפרד, יש לסווג דרישות לא פונקציונליות לפי wikipedia NFR).

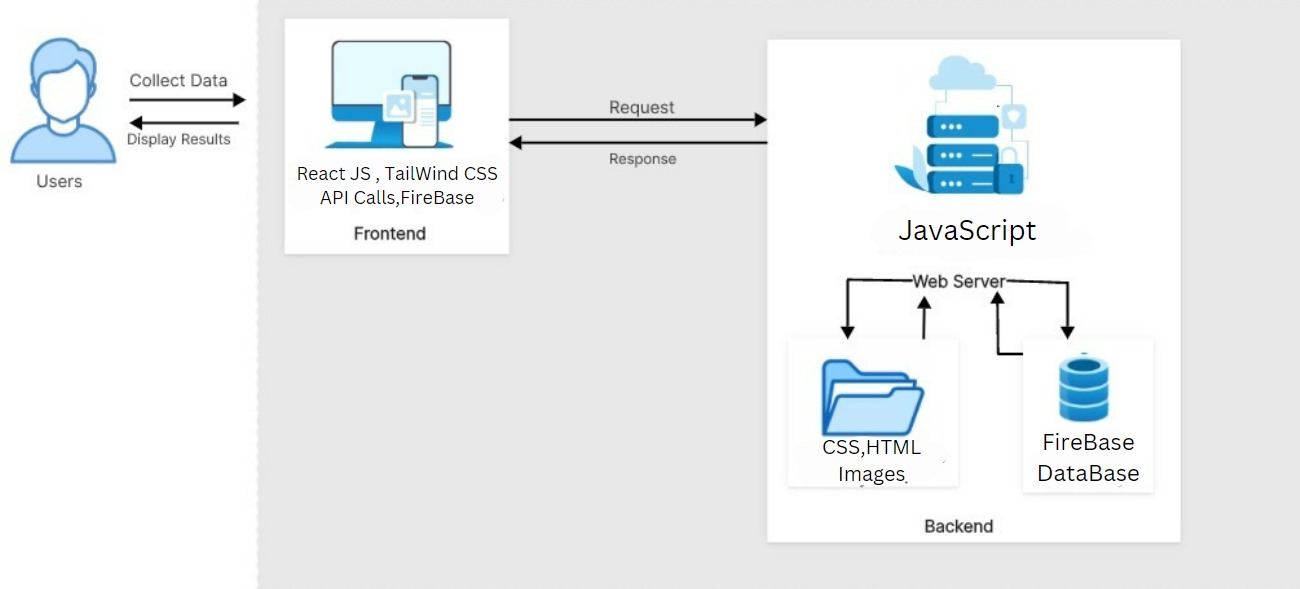
**דרישות פונקציונליות:**

| קטגוריה | דרישה פונקציונלית (מערכת מאפשרת…) |
| --- | --- |
| **התחברות משתמש** | * רישום משתמשים * התחברות לחשבון אישי * ניהול חשבון אישי * משתמש יוכל לשחזר את הסיסמה במקרה של שכחה באמצעות כתובת דוא"ל. |
| פרופיל משתמש וניהול נתונים | * ניהול המידע האישי כולל פרטי בריאות בסיסיים * עדכון תמונת פרופיל * עדכון הגדרות אישיות אחרות |
| מעקב מדדים בריאותיים | * מעקב וניהול מדדים בריאותיים * לראות היסטוריה של מדדים אלה על פני תקופות זמן שונות |
| צפייה בתוצאות רפואיות | * המשתמש יוכל לצפות בתוצאות הבדיקות הרפואיות שלו דרך האפליקציה. |
| הזמנת פגישות עם רופאים | * המשתמש יוכל להזמין פגישות עם רופאים ולבחור זמני פגישה. |
| תצוגת גרפים ויזואליזציה | * מאפשרים למשתמשים לראות שינויים לאורך זמן. * לבחור תקופות זמן שונות להצגת הנתונים |
| **מצב כהה**: | * המשתמש יוכל לעבור בין מצב כהה למצב בהיר באמצעות כפתור המתאים. |
| **הצגת תרופות וויטמינים:** | * האפליקציה תציג תרופות, ויטמינים ותוספי תזונה טבעיים עם שם, מחיר ותמונה. |
| **צפייה בפרטי תרופה:** | * המשתמש יוכל לצפות בפרטי התרופה בעת לחיצה על כפתור "View Details" בעמוד פרטי התרופה. |
| מטרות בריאותיות ומעקב | * הגדרת מטרות בריאותיות * מעקב אחר התקדמות המשתמש לעבר המטרות שהגדיר. |
| התראות מותאמות אישית | * קבלת התראות ותזכורות מותאמות אישית כאשר מדדים חורגים מהתחום המומלץ או כאשר יש צורך בעדכון נתונים. |
| יצירת דוחות מותאמים אישית | * יצירת דוחות מותאמים אישית על בסיס הנתונים הבריאותיים |
| שיתוף נתונים רפואיים | * אפשרות לשתף נתונים בריאותיים עם רופאים או אנשי מקצוע רפואיים ישירות דרך המערכת. |

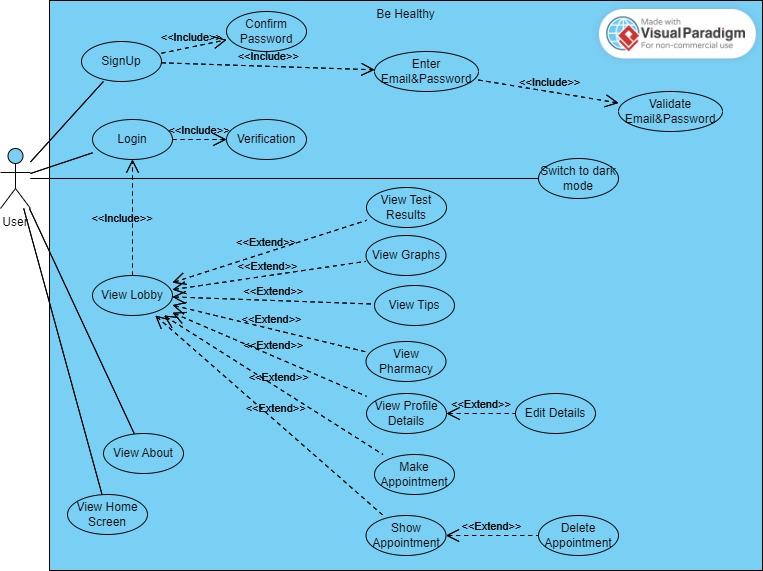
**דרישות לא פונקציונליות:**

| קטגוריה | דרישה לא פונקציונלית |
| --- | --- |
| ביצועים (Performance) | * זמן טעינת הדפים לא יעלה על 3 שניות. * המערכת תתמוך להזרמת מידע בזמן אמת ללא עיכובים משמעותיים. |
| שימושיות (Usability) | * המערכת תכיל מערכת חיפוש חכמה שתמצא תוכן רלוונטי במהירות ויעילות. * הממשק יהיה אינטואיטיבי וקל להבנה, עם מדריכים ועזרה בקו. * המשתמשים יוכלו להתאים אישית את התצוגה וההגדרות של המערכת. |
| אבטחת מידע (Security) | * המערכת תצפין את כל הנתונים האישיים של המשתמשים ותשמור עליהם בהתאם לדרישות רגולציה * המערכת תשתמש באלגוריתמים מוכיחים לאיתור ופתרון בעיות כדי להבטיח אמינות גבוהה. |
| זמינות (Availability) | * המערכת תהיה זמינה כל הזמן * המערכת תתמוך בשחזור נתונים עד לנקודת הזמן האחרונה במקרה של קריסת שרת |
| ניידות (Portability) | * המערכת תהיה תואמת גרסאות קודמות של דפדפנים והתוספים המובילים. |
| קנה מידה (Scalability) | * המערכת תתמוך בהרחבות ועדכונים קלים תוך שמירה על יציבות המערכת. |
| עמידות (Reliability) | * המערכת תוכל להתמודד עם קלט שגוי או לא צפוי מבלי לקרוס או להפסיק לפעול |
| התאמה אישית (Customizability) | * המשתמשים יוכלו להתאים אישית את התצוגה וההגדרות של המערכת בהתאם להעדפותיהם האישיות. |
| חוויית משתמש (User Experience) | * המערכת תספק חוויית משתמש חלקה ואחידה בכל הפלטפורמות (ווב, ניידים) ללא בעיות עיצוב או פונקציות לא מתואמות. |
| אחזור נתונים (Data Retrieval) | * המערכת תספק אפשרות חיפוש מהיר ומדויק של נתונים מתוך מאגרי מידע גדולים |
| תמיכה בריבוי משתמשים (Multi-user Support) | * המערכת תתמוך בפעולה בו זמנית של מספר משתמשים עם ניהול נכון של קונפליקטים ומשאבים. |
| תצוגת נתונים (Data Display) | * המערכת תספק אפשרויות תצוגה מגוונות לנתונים המוזנים, כולל גרפים, טבלאות ודוחות כדי להקל על הבנת המידע |
| הנחות נתונים (Data Presets) | * המערכת תספק אפשרות לשימוש בהנחות נתונים מוגדרות מראש כדי לייעל את תהליך ההזנה ולמנוע טעויות חוזרות. |
| שיתוף נתונים (Data Sharing) | * המערכת תתמוך ביכולת שיתוף נתונים עם משתמשים אחרים או מערכות חיצוניות בהתאם להרשאות שנקבעו. |
| סינון נתונים (Data Filtering) | * המערכת תאפשר למשתמשים לסנן את הנתונים המוזנים לפי קריטריונים שונים כדי לייעל את חווית השימוש. |
| ניהול שגיאות (Error Management) | * המערכת תספק הודעות שגיאה ברורות וידידותיות למשתמש במקרה של נתונים שגויים, עם הסברים כיצד לתקן את הבעיה. |
| יכולת עריכה (Edit Capability) | * המשתמשים יוכלו לערוך את הנתונים שהוזנו בכל שלב, עם שמירה אוטומטית של שינויים ועדכון המידע במאגר הנתונים. |
| אימות נתונים (Data Validation) | * המערכת תוודא שכל הנתונים המוזנים נכונים ומדויקים על ידי בדיקות נוספות כמו בדיקות פורמט, טווחים מקובלים, ולוגיקה פנימית. |

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



**3. הציגו דיאגרמת use case המתארת את השימוש באתר.**



4. יש להציג מבנה סופי של האתר שלכם:

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

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

· - front-end: React or Preact with Tailwind

- back-end:

- option 1 - remote services/APIs from morethanwallet.com (https://www.morethanwallet.com/appStore/apidoc)

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

א. האתר ימומש ב -react/preact, וכן שימוש ב Tailwind - נא להציג דיאגרמה המתארת את התיקיות והקבצים השונים. יש לפרט את הקומפוננטות השונות.

Final\_Project/

│

├── .gitignore

├── package-lock.json

├── package.json

├── tailwind.config.js

│

├── public/

│ ├── favicon.ico

│ ├── index.html

│ ├── logo192.png

│ ├── logo512.png

│ ├── manifest.json

│ └── robots.txt

│

└── src/

├── App.js

├── index.css

├── index.js

│

├── Dark\_Mode/

│ └── DarkModeContext.js

│

├── FireBase/

│ └── Firebase.js

│

├── Header\_classes/

│ ├── Contact/

│ │ └── Contact.js

│ │

│ ├── Css\_Files/

│ │ ├── Contact.css

│ │ ├── Features.css

│ │ ├── Login.css

│ │ └── Signup.css

│ │

│ ├── Features/

│ │ ├── FeatureItem.js

│ │ ├── Features.js

│ │ ├── Footer.js

│ │ ├── Header.js

│ │ └── MobileMenu.js

│

├── Lobby\_Classes/

├── Med\_PNG/

│ ├── iron.png

│ ├── omega\_3.png

│ ├── paracetamol.png

│ ├── turmeric.png

│ ├── vitamin\_c.png

│ ├── vitamin\_d.png

│ └── zinc.png

│

├── Profile/

│ ├── EditDetails.js

│ ├── FormField.js

│ ├── FormSection.js

│ ├── Header.js

│ └── ProfilePhoto.js

│

├── Tests/

│ ├── Filters.js

│ ├── generateTestResults.js

│ ├── Header.js

│ ├── TestResultItem.js

│ ├── TestResults.js

│ └── TestResultsList.js

│

└── Tips/

├── BMISection.js

├── Header.js

├── RecipesSection.js

├── SportsSection.js

└── Tips.js

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.

1. Appointments

Purpose: Stores information about scheduled medical appointments.

Structure: Each appointment includes details such as appointment type, date, doctor, medical center, and reason for the appointment.

2. Patients

Purpose: Contains test results for each patient.

Structure: Each patient has multiple test dates, and each test date includes various test types with associated parameters like test name and values.

3. Users

Purpose: Contains personal information about users.

Structure: Each user record includes details such as address, date of birth, email, family name, gender, height, ID number, and name.

Root

│

├── Appointments

│ ├── appointmentID\_1

│ │ ├── appointmentType: "Checkup"

│ │ ├── date: "2024-08-29"

│ │ ├── doctor: "Dr. Smith"

│ │ ├── medicalCenter: "Health Clinic"

│ │ └── reason: "Routine checkup"

│ └── appointmentID\_2

│ ├── appointmentType: "Consultation"

│ ├── date: "2024-09-01"

│ ├── doctor: "Dr. Doe"

│ ├── medicalCenter: "Downtown Clinic"

│ └── reason: "Specialist consultation"

│

├── Patients

│ ├── patientID\_1

│ │ └── testResults

│ │ ├── date\_1: "2024-08-01"

│ │ │ └── testTypes

│ │ │ ├── testType\_1

│ │ │ │ ├── testName: "Blood Sugar"

│ │ │ │ ├── parameter\_1: "Value\_1"

│ │ │ │ └── parameter\_2: "Value\_2"

│ │ │ ├── testType\_2

│ │ │ │ ├── testName: "Cholesterol"

│ │ │ │ ├── parameter\_1: "Value\_1"

│ │ │ │ └── parameter\_2: "Value\_2"

│ │ │ └── ... (up to 11 test types)

│ │ ├── date\_2: "2024-08-10"

│ │ │ └── testTypes

│ │ │ ├── testType\_1

│ │ │ │ ├── testName: "Blood Pressure"

│ │ │ │ ├── parameter\_1: "Systolic"

│ │ │ │ └── parameter\_2: "Diastolic"

│ │ │ └── ... (up to 11 test types)

│ │ └── ... (up to 10 dates)

│ └── patientID\_2

│ └── testResults

│ ├── date\_1: "2024-07-20"

│ │ └── testTypes

│ │ ├── testType\_1

│ │ │ ├── testName: "Hemoglobin"

│ │ │ ├── parameter\_1: "Value\_1"

│ │ │ └── parameter\_2: "Value\_2"

│ │ └── ... (up to 11 test types)

│ └── ... (up to 10 dates)

│

├── Users

│ ├── userID\_1

│ │ ├── address: "123 Main St"

│ │ ├── dateOfBirth: "1990-05-15"

│ │ ├── email: "john@example.com"

│ │ ├── familyName: "Doe"

│ │ ├── gender: "Male"

│ │ ├── height: 180

│ │ ├── idNumber: "123456789"

│ │ └── name: "John"

│ └── userID\_2

│ ├── address: "456 Elm St"

│ ├── dateOfBirth: "1985-11-22"

│ ├── email: "jane@example.com"

│ ├── familyName: "Smith"

│ ├── gender: "Female"

│ ├── height: 165

│ ├── idNumber: "987654321"

│ └── name: "Jane"

יש להגיש את כל התוצרים בסביבת MTW :

● GIt repository - קישור (הריפו שפתחתם לצוות)

● Git Pages - קישור

● MoreThanWallet.com Gallery

● **הגשה בתיבת ההגשה במוודל – "הגשת פרויקט**"

להזכירכם , יש לבצע פריסה (deployment) ב - morethanwallet.com App Gallery

תוך שימוש ב- vercel.

הסבר על הגשה זו ניתן למצוא ב:

https://www.morethanwallet.com/appStore/gettingStarted

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

הנחיות הגשה:

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

לא תתאפשר בדיקה מחודשת של העבודה עקב טעויות בעת ההגשה!

2 .יש להגיש את התרגיל בקבוצות שהוגדרו בקורס.

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

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

**שימו לב – יש לכלול בתחילת העבודה דף שער כמפורט למטה:**