

המכללה האקדימית בראודה כרמיאל

טכנולוגיות אינטרנט מתקדמות (WEB) – 61776

קבוצה 25 – A19 – TeamPower

מגישים:

אדהם אסעד - 208166249

ליאן עאמר – 323068700

הייא כבישי – 211582978

דניאלה קדמאני – 314701699

רונן זיאן - 318239563

<קישור לתיקיית גיט ציבורי>

https://github.com/RonenZeyan/A19-TeamManagement

<קישור לאתר>

https://a19-team-management-w1ax.vercel.app/

<קישור ל -MTW>

<https://www.morethanwallet.com/app/728>

**תוכן עניינים:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **תקציר ...................................................................................** | | | | **2** |
| **1.חלוקת הפרויקט .....................................................................** | | | | **3** |
| **2.דרישות המערכת .....................................................................** | | | | **5** |
|  | 2.1. הדרישות הפונקציונליות **.......................................................** | | | 5 |
|  | 2.2. הדרישות לא פונקציונליות **....................................................** | | | 6 |
| **3.דיאגרמות המערכת ................................................................** | | | | **7** |
|  | 3.1. ארכיטקטורה של האתר **........................................................** | | | 7 |
|  | 3.2. דיאגרמת Use case **............................................................** | | | 8 |
| **4.מבנה האתר............................................................................** | | | | **9** |
|  | 4.1.דיאגרמה המתארת את התיקיות והקבצים השונים **...........................** | | | 9 |
|  | 4.2.מסדי נתונים MongoDB**.............................................................** | | | 13 |
|  | | 4.2.1.שילוב MongoDB לאחסון וניהול נתונים**..............................** | 13 |
|  | | 4.2.2. דיאגרמת מבנה DB**........................................................** | 13 |
|  | | 4.2.3.אוספים **.....................................................................** | 14 |
| **5.תיק למתכנת ..........................................................................** | | | | **15** |
|  | 5.1.הפונקציות המרכזיות **......................................................** | | | 15 |
|  | 5.2.תשתיות, טכנולוגיות ומקורות חיצוניים שנעשה הבם שימוש**............** | | | 15 |
|  |  | 5.2.1.פירוט API **........................................................................** | | 15 |
|  |  | 5.2.2.קטעי קוד מיוחדים בצד לקוח **...............................................** | | 16 |
|  |  | 5.2.3.קטעי קוד מיוחדים בצד שרת **..............................................** | | 17 |
|  |  | 5.2.4.סביבות מיוחדות **...............................................** | | 17 |
| **6.תיק משתמש ..........................................................................** | | | | **18** |
|  | 6.1.המסך הראשי לאפליקציה **......................................................** | | | 18 |
|  | 6.2.מסך ההתחברות (login) **........................................................** | | | 18 |
|  | 6.3.מסך אודותינו (About Us) **........................................................** | | | 19 |
|  | 6.4.מסך צור קשר (Contavt Us) **........................................................** | | | 19 |
|  | 6.5.מסך שאלות ותשובות (FAQs) **..............................................** | | | 20 |
|  | 6.6.מסך למד עוד (Learn more) **........................................................** | | | 20 |
|  | 6.7.מסכי משתמש מסוד מנהל**........................................................** | | | 21 |
|  | 6.8.מסך מסכי משתמש מסוד עובד**........................................................** | | | 25 |
| **7.רפרנסים ..................................................................................** | | | | 28 |

**תקציר:**

TeamPower היא אפליקציית אינטרנט המציעה פלטפורמה לניהול משימות[[1]](#footnote-1), המאפשרת למנהל לפקח ולהקצות משימות ביעילות לעובדיו. האפליקציה מציעה מגוון פונקציונליות לייעל את זרימת העבודה ולטפח שיתוף פעולה בתוך הצוות כגון ניהול עובדים, ניהול משימות תקשורת. TeamPower נועד לספק חווית משתמש עקבית במכשירים שונים, מערכת הניווט האינטואיטיבית, הכוללת סרגל ניווט ייעודי, מאפשרת למשתמשים לחקור בצורה חלקה את תכונות האפליקציה.

**תכונות עיקריות:**

ניהול עובדים:

* מנהל מערכת יכול להוסיף, למחוק ולערוך פרופילי עובדים.
* לכל עובד מוקצית סיסמה ייחודית.

ניהול משימות:

* מנהל יכול להקצות משימות לעובדים ספציפיים, כולל תאריך יעד ופרטי משימות.
* עובדים יכולים לראות את המשימות שהוקצו להם, לקבל ולסמן משימות כמושלמות.
* יישום עוקב אחר מצב/ סטטוס המשימות (בהמתנה, בתהליך, הושלמו).

תקשורת:

* מנהל יכול לשלוח הודעות לכל העובדים.
* עובדים יכולים לראות את ההודעות שנשלחו על ידי המנהל.

ניהול פרופילים:

* העובדים יכולים לערוך את הפרופילים האישיים שלהם ולהעלות תמונת פרופיל

**טכנולוגיות מפתח:**

* React: היא ספריית JavaScript לבניית ממשקי משתמש עם ארכיטקטורה מבוססת רכיבים.
* Tailwind CSS: היא מסגרת CSS-ראשונה בשירותים המספקת קבוצה של מחלקות מוגדרות מראש לבנייה מהירה של ממשקי משתמש רספונסיביים ומותאמים אישית ללא צורך בכתיבת קוד CSS נרחב.
* MongoDB: מסד נתונים לאחסון נתוני יישומים.
* **חלוקת הפרויקט:** אחראי מערכת, אחראי הגדרת והקצאת המשימות אדהם אסעד, חלקנו את העבודה ביננו באופן הבא:

|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** |
| אדהם אסעד | * קבלת החלטה ותכנון משותף כיצד האתר יעבוד, יראה ויצירת הדפים. | הושלם |
| * הקצאת משימות וחלוקת עבודה + בדיקת התקדמות במשימות. | הושלם |
| * דיאגרמת קבצים ותיקיות: להציג תרשים היררכי של תיקיות הפרויקט והקבצים השונים | הושלם |
| * חיבור בסיס נתונים באמצעות Mongodb ובניית שאילתות ולקשר אותו לאפליקציה | הושלם |
| * תיקון ארכיקטורת אתר וחלוקת הקוד | הושלם |
| * תיעוד הקוד: לתעד את הפונקציות המרכזיות בקוד בצד השרת | הושלם |
| ליאן עאמר | * קבלת החלטה ותכנון משותף כיצד האתר יעבוד, יראה ויצירת הדפים. | הושלם |
| * עיצוב המבנה והפריסה של האתר לפונקציונליות מיטבית. | הושלם |
| * דיאגרמת מבנה בסיס הנתונים: דיאגרמת מבנה של בסיס הנתונים עם הטבלאות והקשרים ביניהן | הושלם |
| * בניית, עיצוב והוספת דף בית לאתר ב - React | הושלם |
| * תיקון עיצוב האתר | הושלם |
| הייא כבישי | * קבלת החלטה ותכנון משותף כיצד האתר יעבוד, יראה ויצירת הדפים. | הושלם |
| * ארכיטקטורה של האתר - לזהות ולתאר אינטראקציות שונות של משתמשים ותגובות מערכת. | הושלם |
| * תיק משתמש: להכין דפי הסבר ותמונות להדגמת השימוש באתר (כמו: הרשמה, התחברות, וכו') | הושלם |
| * דרישות פונקציונליות - לאסוף ולרשום רשימה מפורטת של דרישות פונקציונליות לאתר   דרישות לא פונקציונליות - לסווג ולפרט דרישות לא פונקציונליות לקטגוריות שונות (ביצועים, אבטחה, שמישות וכו') | הושלם |
| * תיקון דיאגרת מבנה מערכת ארכיקטורה | הושלם |
| דניאלה קדמאני | * קבלת החלטה ותכנון משותף כיצד האתר יעבוד, יראה ויצירת הדפים. | הושלם |
| * יצירת דיאגרמת use case לתהליכים והמשתמשים של האתר. | הושלם |
| * פריטי המידע: לציין ממקור המידע ייאסף (API חיצוני ספציפי או מסד נתונים) | הושלם |
| * חיבור בסיס נתונים באמצעות Mongodb ובניית שאילתות ולקשר אותו לאפליקציה | הושלם |
| * תיקון ועדכון דיאגרת תיקיות | הושלם |
| רונן זיאן | * קבלת החלטה ותכנון משותף כיצד האתר יעבוד ויראה | הושלם |
| * שלב Tailwind CSS לעיצוב משופר ותגובתיות. | הושלם |
| * העברת דפי האתר ב - React | הושלם |
| * תיעוד הקוד: לתעד את הפונקציות המרכזיות בקוד בצד הלקוח | הושלם |
| * חיבור בסיס נתונים באמצעות Mongodb ובניית שאילתות ולקשר אותו לאפליקציה | הושלם |
| * תיקון הפונקציונליות של המערכת | הושלם |

* **דרישות המערכת:**
  1. **הדרישות הפונקציונליות FR**

1. The system allows users to switch between dark and light modes based on their preferences.
2. The system allows users to log in using valid email and password authentication and remain logged in after successful login.
3. The system allows users to update their user profile and upload a profile picture.
4. The system allows the administrator to add, edit, and delete employee information.
5. The system allows the administrator to view a table of all employees along with relevant personal details.
6. The system allows the administrator to add or delete tasks for specific employees.
7. The system allows for searching and filtering content in tables by selecting search fields.
8. The system allows the administrator to view a table of all tasks in the system along with their current status (pending, in progress, completed).
9. The system allows the administrator to send messages to all employees.
10. The system allows employees to view a table of their assigned tasks.
11. The system allows employees to receive and approve new tasks assigned to them.
12. The system allows employees to update the status of their tasks (in progress, completed, pending approval).
13. The system allows employees to view messages sent by the administrator.
    1. **הדרישות לא פונקציונליות NFR**

1) Usability:

- An intuitive, accessible, and consistent user interface for all users and devices (desktop, mobile, tablet).

- An easy-to-use navigation system with a clear and task-focused navigation bar.

- Error messages, alerts, and instructions are clear, concise and in simple language.

2) Performance:

- Responsive, fast and efficient user interface with minimal loading times.

3) Security:

- User data, tasks and other sensitive information are secured and encrypted against unauthorized access.

- Role-based authentication and access control mechanisms for users.

4) Reliability:

- High availability with minimal downtime for maintenance.

- The system guarantees integrity, confidentiality and reliability of sensitive data such as employee details and tasks.

5) Maintenance:

- Modular architecture that allows easy maintenance and updates of specific components.

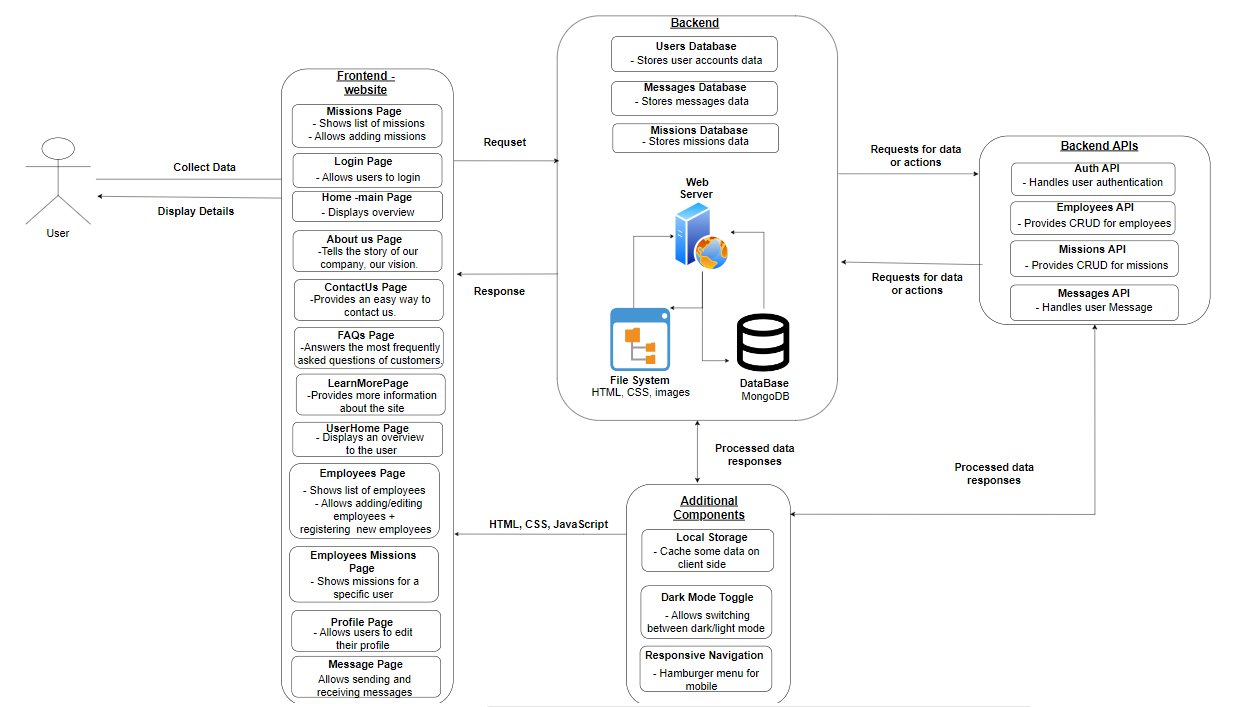
- Well-documented code with comprehensive technical documentation and compliance with leading coding standards.

6) Compatibility:

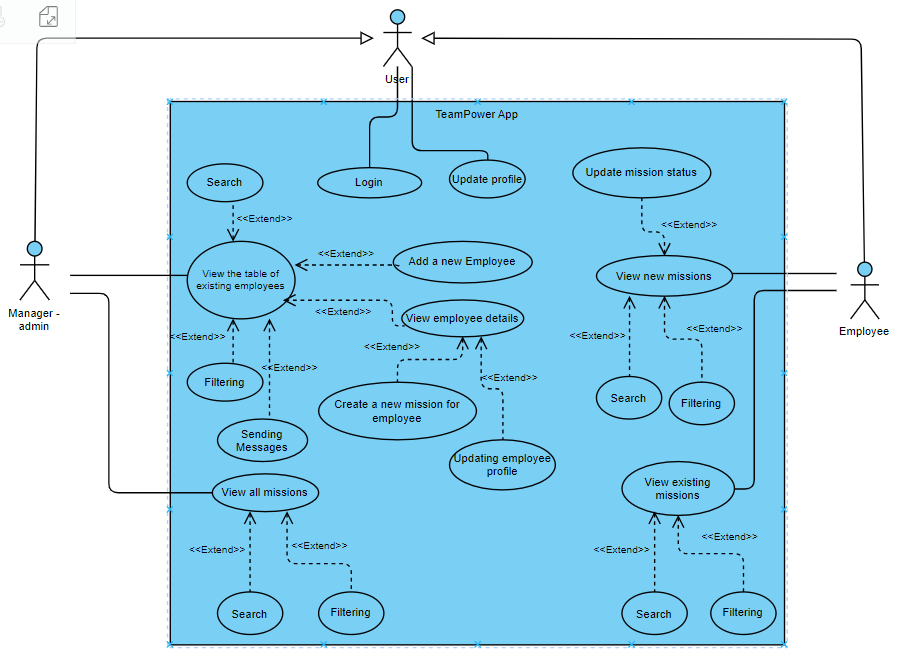
- The system is compatible with current web browsers and various devices (desktop, mobile, tablet).

- Responsive interface adapted to different screen sizes.

* **דיאגרמות המערכת** 
  1. ארכיטקטורה של האתר (תרשים הכולל את האלמנטים המרכזיים)



* 1. דיאגרמת use case



* 1. דיאגרמה המתארת את התיקיות והקבצים השונים.

─ 📁finalProjectA19

│ ├─ 📁client

│ │ ├─ 📁public

│ │ │ ├─📁Images

│ │ │ │ ├─ 📄finishStatus.png

│ │ │ │ ├─ 📄HomeBG.jpg

│ │ │ │ ├─ 📄iconFemale.png

│ │ │ │ ├─ 📄iconMan.png

│ │ │ │ ├─ 📄loginBG.jpeg

│ │ │ │ ├─ 📄Logo.png

│ │ │ │ └─ 📄UserHomeBG.jpeg

│ │ │ ├─ 📄index.html

│ │ │ ├─ 📄manifest.json

│ │ │ └─ 📄robots.txt

│ │ ├─ 📁src

│ │ │ ├─📁components

│ │ │ │ ├─📁Auth

│ │ │ │ │ └─ 📄AuthContext.js

│ │ │ │ ├─📁DarkMode

│ │ │ │ │ ├─ 📄DarkMode.css

│ │ │ │ │ └─ 📄DarkMode.js

│ │ │ │ ├─📁EditUser

│ │ │ │ │ └─ 📄EditableField.js

│ │ │ │ ├─📁Footer

│ │ │ │ │ ├─ 📄FooterNew.css

│ │ │ │ │ └─ 📄FooterNew.js

│ │ │ │ ├─📁Modals

│ │ │ │ │ ├─ 📄NewEmployeeModal.css

│ │ │ │ │ └─ 📄NewMissionModal.js

│ │ │ │ ├─📁Navbar

│ │ │ │ │ └─ 📄Navbar.js

│ │ │ │ ├─📁Search

│ │ │ │ │ └─ 📄Search.js

│ │ │ │ ├─📁Table

│ │ │ │ │ ├─ 📄Cell.js

│ │ │ │ │ ├─ 📄Row.js

│ │ │ │ │ └─ 📄Table.js

│ │ │ ├─📁DisplayMSGAndMissions

│ │ │ │ ├─📄DisplayMSG.js

│ │ │ │ └─📄DisplayUserMissions.js

│ │ │ ├─📁pages

│ │ │ │ ├─📁AdminPages

│ │ │ │ │ ├─ 📄AdminHomePage.js

│ │ │ │ │ ├─ 📄CreateNewMessage.js

│ │ │ │ │ ├─ 📄EditUserByAdminPage.js

│ │ │ │ │ ├─ 📄EmployeesPage.js

│ │ │ │ │ ├─ 📄MissionsPage.js

│ │ │ │ │ └─ 📄UserMissionsPage.js

│ │ │ │ ├─📁FooterPages

│ │ │ │ │ ├─ 📄AboutUs.js

│ │ │ │ │ ├─ 📄ContactUs.js

│ │ │ │ │ └─ 📄FAQs.js

│ │ │ │ ├─📁StartPages

│ │ │ │ │ ├─ 📄HomePage.js

│ │ │ │ │ ├─ 📄LearnMorePage.js

│ │ │ │ │ └─ 📄LoginPage.js

│ │ │ │ ├─📁UsersPages

│ │ │ │ │ ├─ 📄MessagePage.js

│ │ │ │ │ ├─ 📄ProfilePage.js

│ │ │ │ │ ├─ 📄UserExistingMissionsPage.js

│ │ │ │ │ ├─ 📄UserHomePage.js

│ │ │ │ │ └─ 📄UserNewMissionsPage.js

│ │ │ ├─📁utils

│ │ │ │ └─📄JWTutils.js

│ │ │ ├─📄App.css

│ │ │ ├─ 📄App.js

│ │ │ ├─ 📄App.test.js

│ │ │ ├─ 📄index.css

│ │ │ ├─📄index.js

│ │ │ ├─ 📄logo.svg

│ │ │ ├─ 📄reportWebVitals.js

│ │ │ └─ 📄setupTests.js

│ │ ├─📄.gitignore

│ │ ├─📄package-lock.json

│ │ ├─📄package.json

│ │ ├─📄README.md

│ │ └─📄tailwind.config.js

│ ├─ 📁server

│ │ ├─ 📁controllers

│ │ │ ├─📄Admincontroller.js

│ │ │ └─📄AllUserscontroller.js

│ │ ├─ 📁middleware

│ │ │ └─📄uploadimage.js

│ │ ├─ 📁models

│ │ │ ├─📄Missions.js

│ │ │ ├─📄MSG.js

│ │ │ └─📄User.js

│ │ ├─ 📁public

│ │ │ ├─📁Images

│ │ │ │ ├─ 📄iconFemale.png

│ │ │ │ └─ 📄iconMan.png

│ │ ├─ 📁routes

│ │ │ ├─📄AdminRoute.js

│ │ │ └─📄AllUsersRoute.js

│ │ ├─📄.env

│ │ ├─📄.gitignore

│ │ ├─📄index.js

│ │ ├─📄package-lock.json

│ │ ├─📄package.json

│ │ └─📄wercel.json

* 1. מסדי נתונים MongoDB
     1. שילוב MongoDB לאחסון וניהול נתונים

המערכת משתלבת עם MongoDB המספק פתרון מבוסס ענן (Cloud) שבו ניתן לאחסן מידע באופן מרכזי ולנהל אותו בצורה קלה ואבטחה. תהליך האינטגרציה כולל את המרכיבים הבאים:

* חיבור של מסדי נתונים:

- בתוך קובץ `.env` נמצא DATABASE\_URL אשר יוצר את החיבור למופע MongoDB. קובץ `.env` שומרים בתוכו משתנים קבועים שהם רגישים ולכן, שמרנו בתוכו את כתובת ה URL בשביל להגן על המידע מפני פריסות ולספק הבטחת מידע.

- הוא מגדיר את 'PORT' עבור השרת להאזנה בו, כמו גם את המחרוזת 'DATABASE\_URL' שמציינת את פרטי החיבור עבור מסד הנתונים של MongoDB.

- `DATABASE\_URL` כולל את הפרוטוקול, שם המשתמש והסיסמה, כתובת השרת ואפשרויות חיבור נוספות הנדרשות ליצירת חיבור מאובטח למופע MongoDB.

* דגמי נתונים – Models:

- ספריית `models/` מכילה את דגמי הנתונים שהוגדרו באמצעות Mongoose, ספריית Object Data Modeling (ODM) עבור MongoDB ו-Node.js.

- מודלים אלו מייצגים את מבנה הנתונים המאוחסנים באוספים של MongoDB ומגדירים את הסכמות לכל אוסף.

- המודלים כוללים את `User.js` (למידע על המשתמש), `Missions.js` (עבור משימות או משימות שהוקצו), ו-`MSG.js` (עבור הודעות שנשלחו בתוך המערכת).

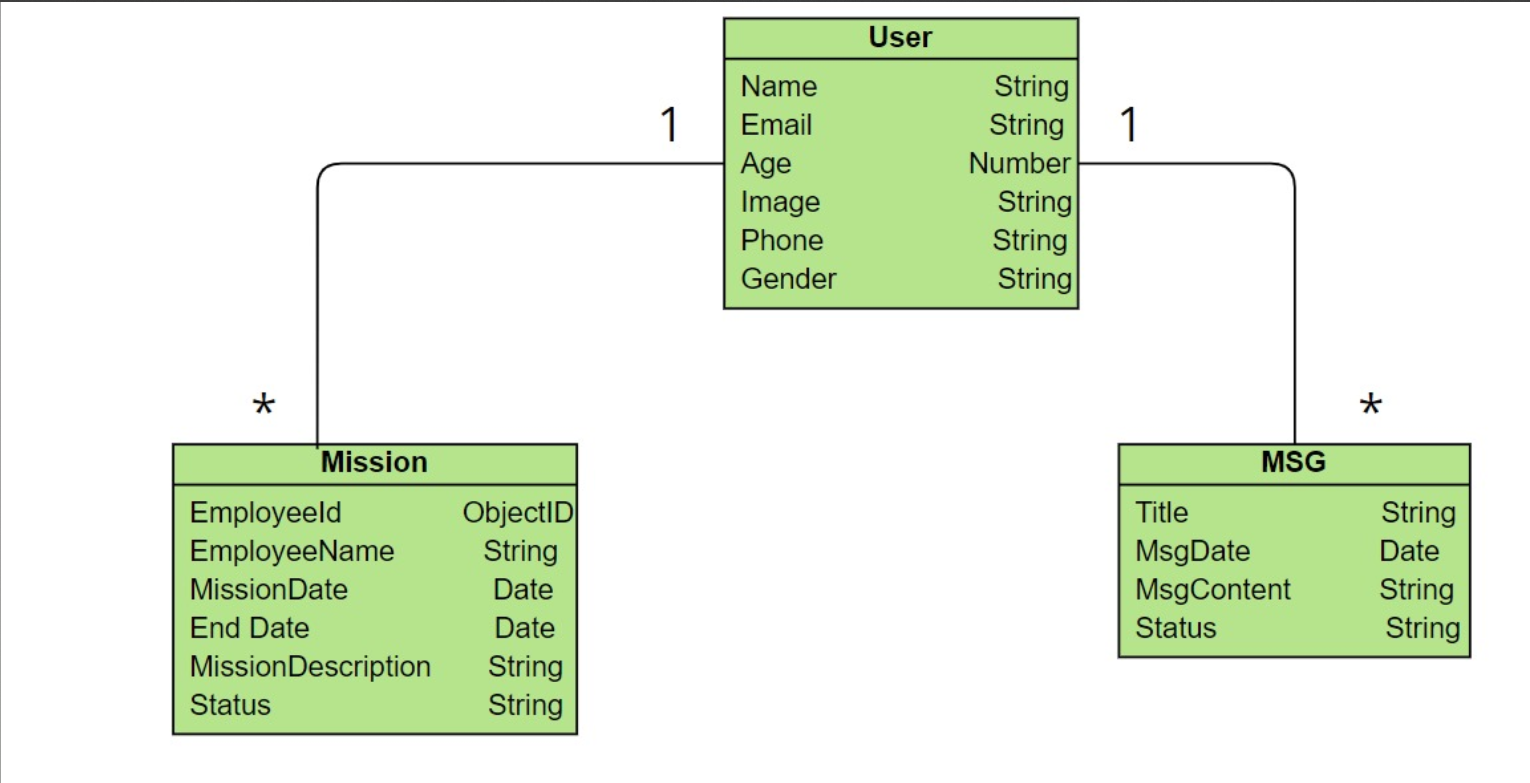
* שכבת גישה לנתונים:

- ספריית `בקרים/` - ' / controllers ' מכילה את רכיבי DAL האחראים לאינטראקציה עם מסד הנתונים של MongoDB.

- רכיבים אלו משתמשים במודלים המוגדרים של Mongoose לביצוע פעולות יצירה, קריאה, עדכון ומחיקה (CRUD) על הנתונים המאוחסנים באוספים של MongoDB.

- רכיבי ה-DAL כוללים `Admincontroller.js` (לניהול משתמשים ומשימות על ידי מנהלים) ו`AllUserscontroller.js` (לניהול משתמשים רגיל).

* + 1. דיאגרמת מבנה DB



* + 1. אוספים :
* משתמש – User

מאחסן מידע אישי על משתמשי המערכת (המנהל והעובדים) עם שדות למטרות אימות והרשאה.

שדות:

- שם: השם המלא של המשתמש (מחרוזת).

- אימייל: כתובת הדוא"ל של המשתמש (מחרוזת).

- גיל: גיל המשתמש (מספר).

- תמונה: כתובת אתר או נתיב לתמונת הפרופיל של המשתמש (מחרוזת).

- טלפון: מספר הטלפון של המשתמש (מחרוזת).

- מגדר: מגדר המשתמש (מחרוזת).

* משימה - Mission

מאחסן מידע על משימות או משימות שהוקצו לעובדים.

שדות:

- EmployeeId: ObjectID המפנה לאוסף המשתמשים (ObjectID).

- EmployeeName: שם העובד (מחרוזת).

- MissionDate: תאריך שבו הוקצתה המשימה (תאריך).

- תאריך סיום: מועד אחרון למשימה (תאריך).

- MissionDescription: פרטים או תיאור של המשימה (מחרוזת).

- סטטוס: סטטוס נוכחי של המשימה (מחרוזת).

* MSG

מאחסן הודעות שנשלחו בתוך האפליקציה.

שדות:

- כותרת: נושא או כותרת ההודעה (מחרוזת).

- MsgDate: תאריך שליחת ההודעה (תאריך).

- MsgContent: תוכן או גוף ההודעה (מחרוזת).

- סטטוס: סטטוס ההודעה (מחרוזת).

1. **תיק למתכנת**
   1. הפונקציות המרכזיות

* אימות משתמש, פונקציה זו כוללת אימות האימייל וסיסמת משתמש מול API חיצוני והנפקת אסימוני JWT לניהול הפעלה. זה מבטיח שמשתמשים יכולים להיכנס בצורה מאובטחת ולקיים אינטראקציה עם האפליקציה בהתאם להרשאות שלהם.
* ניהול עובדים, המערכת מאפשרת ניהול מלא של כוח העבודה בארגון (מאפשר פעולות CRUD) , הכולל יצירה, קריאה, עדכון ומחיקה של רשומות עובדים. ניתן להוסיף עובדים חדשים, לעדכן את פרטיהם האישיים ולהסיר עובדים שעזבו את הארגון.
* ניהול משימות, בדומה לניהול עובדים, המערכת מספקת כלים לניהול משימות ומעקב אחר ביצועי העובדים. ניתן ליצור משימות חדשות, להקצותן לעובדים ספציפיים ולמחוק רשומות משימות. העובדים יכולים לאשר קבלת משימה חדשה, לסמן משימה כ"בתהליך" ולאשר את השלמתה.
* ניהול הודעות, הפונקציה הזו מאפשרת למנהל לשלוח הודעות כלליות לכלל העובדים באמצעות המערכת. העובדים יכולים לצפות בהודעות אלה ולהתעדכן בחדשות וההנחיות הארגוניות.
  1. תשתיות, טכנולוגיות ומקורות חיצוניים שנעשה בהם שימוש
     1. פירוט API :
* Auth API:
* POST /login- מאמת משתמש עם דואר אלקטרוני וסיסמה, יוצר ומחזיר אסימון JWT.
* Employees API:
* GET /users- מאחזר את כל המשתמשים מלבד משתמש המנהל.
* POST /add\_employee - מוסיף עובד חדש עם שם, דוא"ל, טלפון ומין שסופקו.
* GET /userData/:id - מאחזר נתוני משתמש לפי מזהה משתמש.
* POST /editUserData/:id - עדכון נתוני משתמש (כולל תמונת פרופיל) לפי מזהה משתמש.
* DELETE /users/delete/:id - מוחק משתמש והמשימות המשויכות לו לפי מזהה משתמש.
* Missions API:
* GET /allMissions- מאחזר את כל המשימות.
* GET /users/userMissions/:id- מאחזר משימות המשויכות למשתמש לפי מזהה משתמש.
* POST /add\_mission - מוסיף משימה חדשה עם תאריך, תיאור, שעה ומזהה משתמש מסופקים.
* DELETE /userMissions/delete/:id - מוחק משימה לפי מזהה משימה.
* GET /newUserMissions/:id - מאחזר משימות חדשות (ממתינות) עבור משתמש לפי מזהה משתמש.
* GET /ExistingUserMissions/:id - מאחזר משימות קיימות (ממתינות) עבור משתמש לפי מזהה משתמש.
* - POST /acceptMission/:id מעדכן את סטטוס המשימה ל"IN PROCESS" לפי מזהה המשימה.
* POST /finishMission/:id - מעדכן את סטטוס המשימה ל"הושלמה" לפי מזהה המשימה.
* GET /numMissionsMessages/:id - מאחזר את מספר משימות ההמתנה למשתמש לפי מזהה משתמש.
* Messages API:
* GET /get\_messages- מאחזר את כל ההודעות.
* GET /numMessages/:id - מאחזר את מספר ההודעות הממתינות למשתמש לפי מזהה משתמש.
  + 1. קטעי קוד מיוחדים בצד לקוח ( client)
* Local Storage

הוא תכונת API של דפדפני האינטרנט שמאפשרת לאפליקציות האינטרנט לשמור נתונים במקום מקומי בדפדפן. נתונים שמורים ב-Local Storage יישארו זמינים לאפליקציה גם אחרי נסיגת הדפדפן או רענון הדף.

לדוגמה, מצב כהה או בהיר כאשר המשתמש לוחץ על כפתור darkmode נכנס משתנה לlocal storage שאומר אם זה dark או llight ו נשמר כאשר עובדעם דרך דפים

* Reuse

יצרנו תבניות נפרדות לרכיבים שונים (טבלה, שורה, תא) ואז שילבנו אותן יחד, מה שמאפשר לנו לחסוך בכתיבת קוד חוזר ולשמור על קוד מסודר ונקי יותר. לדוגמה, בתבנית של הטבלה (Table) אנו משתמשים בתבנית של השורה (Row) ומעבירים אליה את הנתונים של כל שורה בטבלה. בתבנית של השורה, כמו כן, אתם משתמשים בתבנית של התא (Cell) ומעבירים אליה את הנתונים של כל תא בשורה.

* Single page application

באמצעות BrowserRouter, אנו משנים את הרכיבים שמוצגים בממשק המשתמש בהתאם לנתיב שנבחר, במקום להחליף את הדף כולו בכל שינוי.

לדוגמה, רכיב ה-App מכיל את BrowserRouter מנהל את ניווט הדפים באפליקציה ומשווה את ה-URL לרשימת הנתיבים המוגדרים, מציב את הרכיב המתאים בתוך ה-Route שנבחר. בכך, האפליקציה מאפשרת ניווט דינמי בין הדפים, מבלי לטעון מחדש את הדף כמו בדפדפן רגיל.

* השימוש ב- ` axios `

ספרייה שמאפשרת לבצע בקשות HTTP לשרת בצורה נוחה ונקייה יותר מאשר בשיטות מסורתיות. לדוגמה, בבקשות POST, אנו משתמשים ב־`axios.post()` לשליחת מידע לשרת ועדכון המשאבים. ה־URL שמועבר מציין את כתובת המשאב בשרת

* השימוש ב- `async` ו־`await`

מנגנון שמאפשר להפעיל פונקציות אסינכרוניות באופן יותר קריא וקל לניהול, תוך שמירה על רצף ההפעלה הנכון.

לדוגמה, בפונקציה handleSubmit, השימוש ב־`async` בהגדרת פונקציה מאפשר שימוש ב־`await` להמתנה עד לסיום פעולה מסוימת

* השימוש ב- ` JWT `

סוג של טוקן מאובטח ומוצפן המשמש לאימות משתמשים ושמירת פרטים עליהם, שניתן להעביר בין הלקוח לשרת בצורה יעילה ומאובטחת.

לדוגמה, הפונקציה משתמשת בספריית jwtDecode כדי לפענח את ה-token שנשלח מהשרת ולהשיג ממנו מידע על המשתמש. אם יש שגיאה בפענוח, הפונקציה מדווחת על כך ומחזירה ערך `null`.

* השימוש ב- ` Hooks`

יצרנו Hooks מותאמים אישית לניהול המצב והקשר של אימות המשתמש, מה שמאפשר לנו לשתף מידע זה בין רכיבים שונים באפליקציה.

* + 1. קטעי קוד מיוחדים בצד השרת server))
* השימוש ב- ` Multer `

ספרייה המאפשרת לנו לנהל את תהליך העלאת הקבצים לשרת, כולל הגבלות על סוגי קבצים, גודל וכמות הקבצים.

דוגמה, הפונקציה destination מגדירה את מיקום השמירה של הקובץ בצורה תלת מימדית, ולאחר הגדרה תקינה של התיקייה, היא קוראת ל-cb כדי להמשיך בתהליך.

* + 1. סביבות מיוחדות

1. סביבת פיתוח:

עבור פיתוח היישום, נעשה שימוש בתוכנת Visual Studio Code - סביבת פיתוח קוד פתוח ויעילה. בחירה זו נבעה מהיתרונות הרבים שהיא מציעה, ובפרט האפשרות להתקין הרחבות נוספות התורמות לחוויית הפיתוח, כגון כלים לבקרת גרסאות וכלי עזר לעבודה עם מסדי נתונים וטכנולוגיות שונות. כמו כן, VS Code מספקת מסוף משולב (integrated terminal) הנוח לשימוש ומקל על התפעול השוטף במהלך הפיתוח.

1. מסד נתונים:

לצורך אחסון והפצת הנתונים של היישום, נעשה שימוש בשירות מונגו די-בי (MongoDB) - מסד נתונים מבוזר המבוסס על ענן. הבחירה במונגו די-בי נבעה מהיתרונות הרבים שהוא מספק, כגון התמיכה בקנה מידה גדול, זמינות גבוהה, ופריסה גלובלית לשיפור הביצועים והגמישות. מונגו די-בי הוא פתרון מוביל לאפליקציות אינטרנט מודרניות ויישומים המחייבים עמידות גבוהה ועיבוד מקבילי של נתונים.

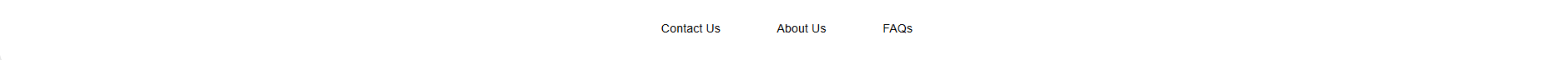
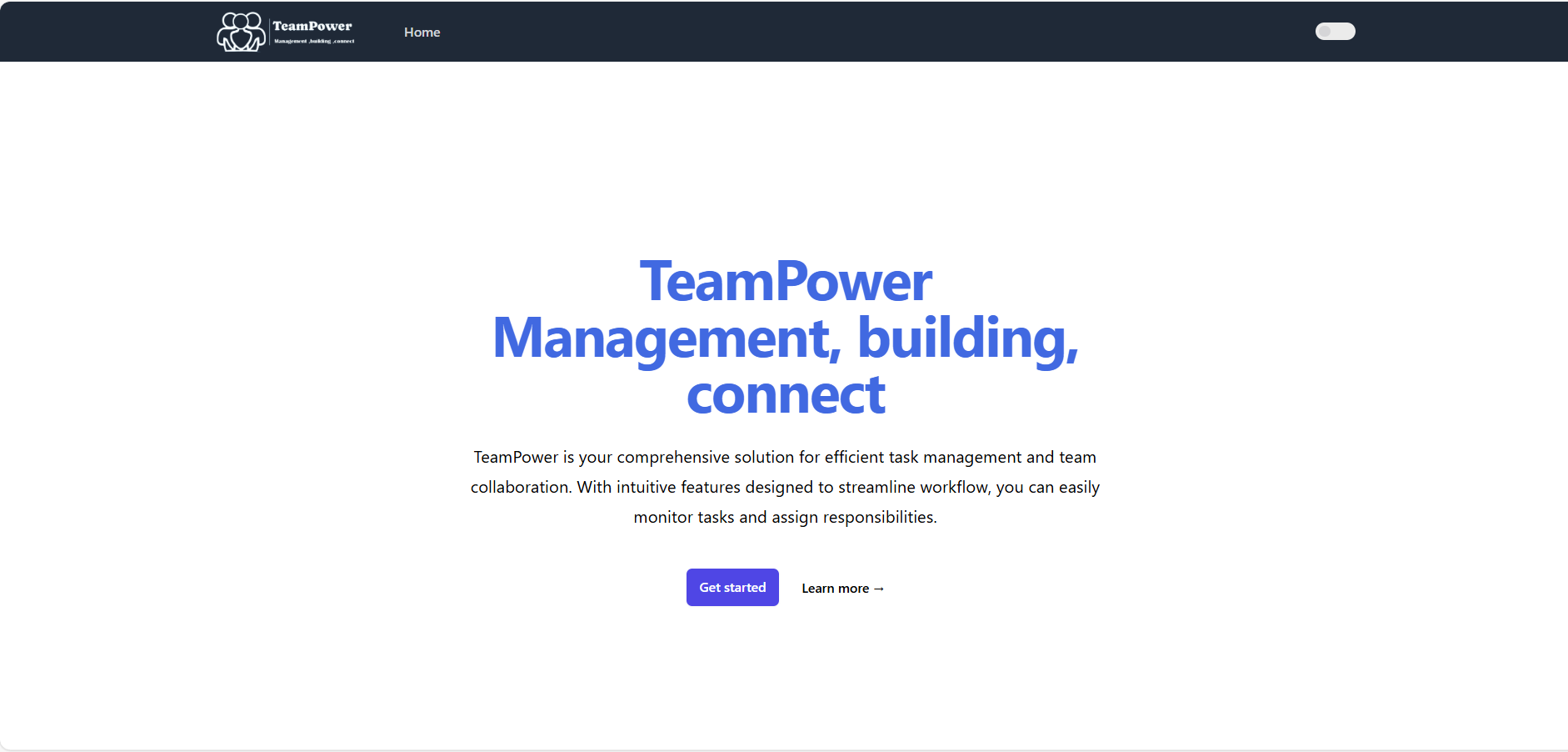
1. **תיק משתמש:**

**הערה** כפתור מצב כהה ובהיא נמצא בכל דף באתר שלנו.

* 1. המסך הראשי לאפליקציה:

המסך הזה היינו המסך הראשי לאפליקציה, בו אנו מציגים בקצרה את מהות האתר.

מופיע בדף כפתור מצב כהה או בהיר, אודותינו, צור קשר, שאלות ותשובות נפוצות, למד יותר על האתר וכפתור התחל.



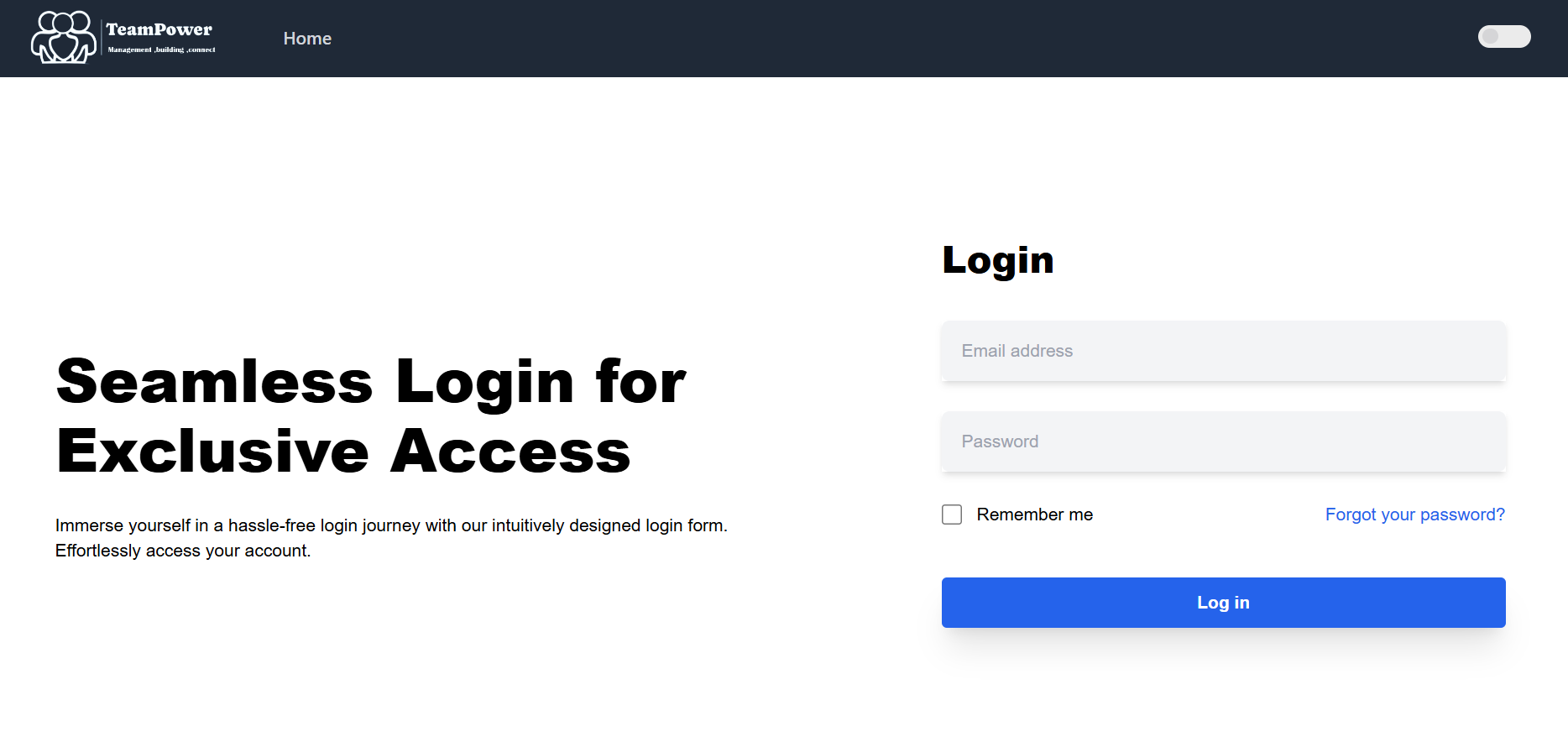
* 1. מסך ההתחברות (login)

במסך הזה, המשתמש שלנו יכול להתחבר, ישנם שני סוג משתמש, הסוג הראשון הוא המנהל (admin) והוא היחידי שמתחבר עם האימייל והסיסמה שלו, והסוג השני הוא עובד כאשר הוא עובד תחת אותו מנהל שמתחבר עם האימייל שלו והסיסמה שהמנהל נותן לו ( סיסמה ייחודית לו).

בעת מילוי הפרטים ולחיצה על כפתור התחבר יפתח מול המשתמש שלנו מסך הפעילויות המתאים לפי סוג המשתמש.

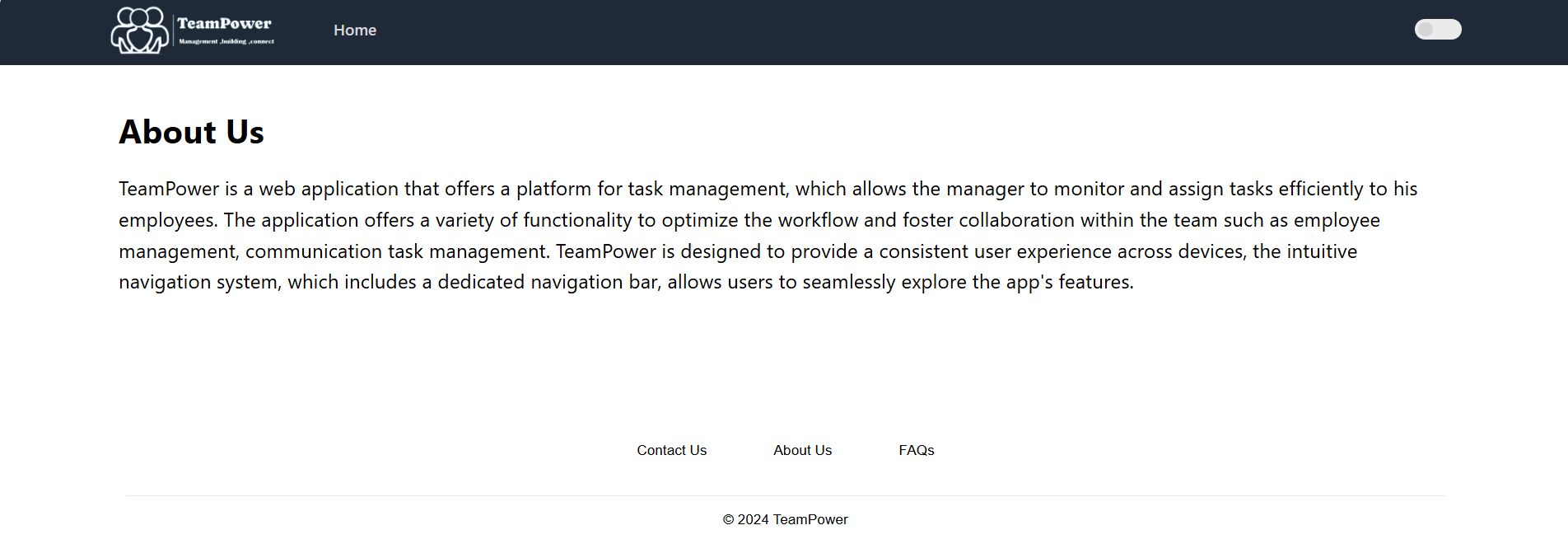
טבלת משתמשים קיימים:

|  |  |  |  |
| --- | --- | --- | --- |
| סיסמה | אימייל | סוג משתמש | שם |
| admin | admin@gmail.com | מנהל | admin |
| ronen | ronen@gmail.com | עובד | ronen |
| adham | adham@gmail.com | עובד | adham |
| daniella | daniella@gmail.com | עובדת | daniella |
| layan | layan@gmail.com | עובדת | layan |
| haya | haya@gmail.com | עובדת | haya |



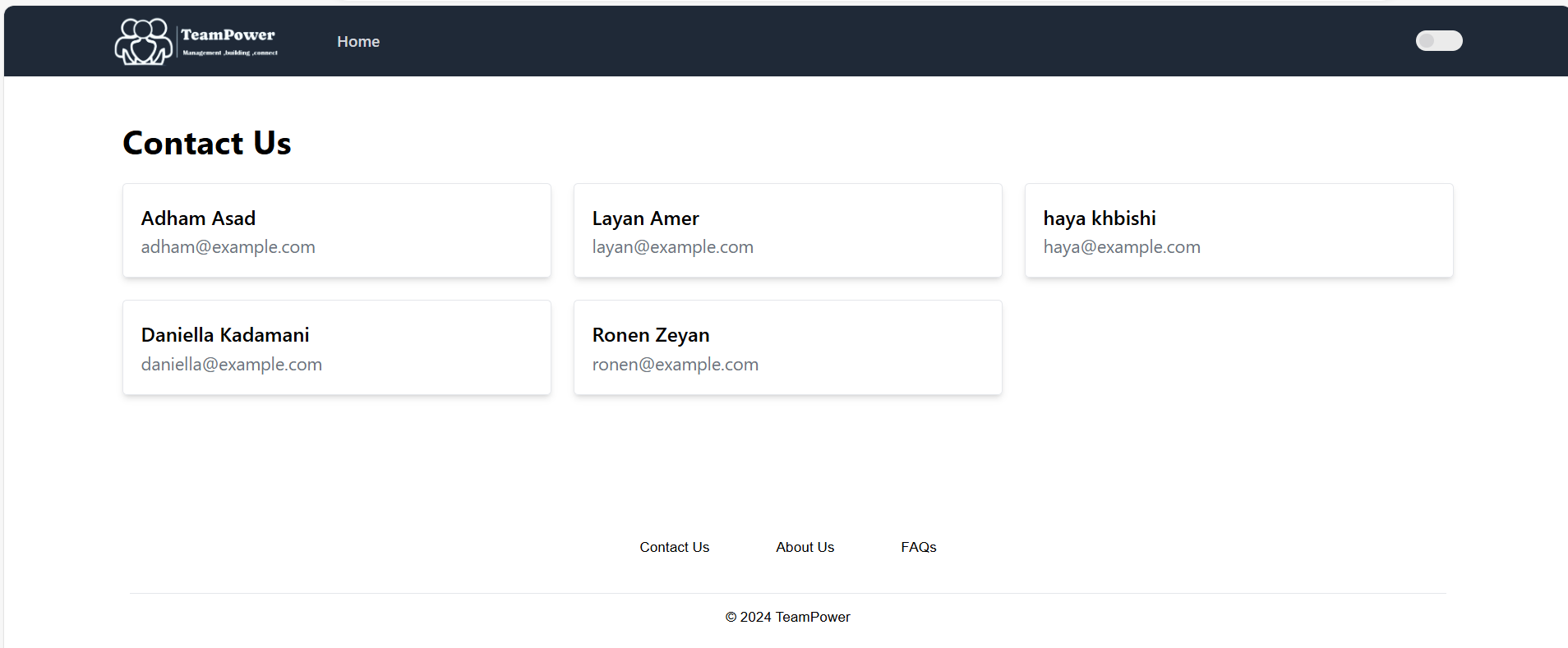
* 1. מסך אודותינו (About Us)

דף זה מכיל מידע על הפרויקט שלנו ומה החזון שלנו



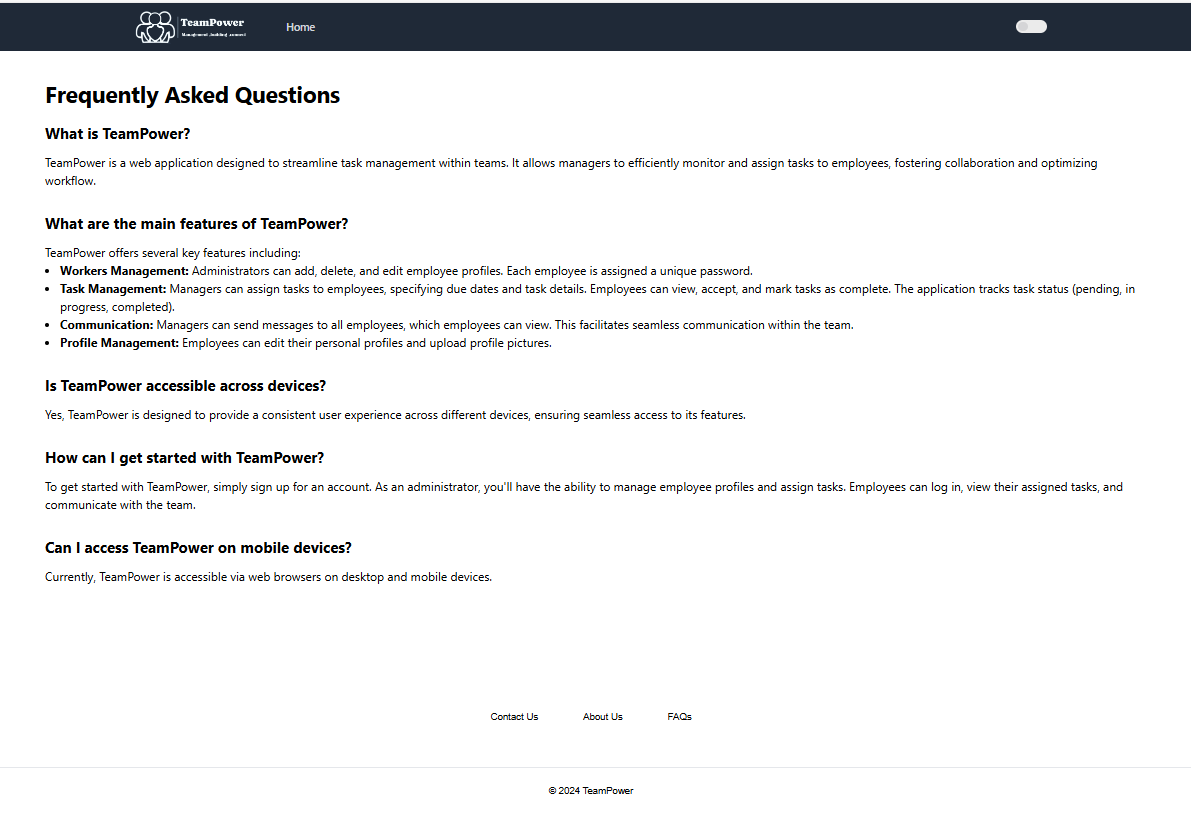
* 1. צור קשר (Contact Us)

בדף זה המשתמש יכול למצוא את כתובת המיילים אישיים שלנו וליצור איתנו קשר דרך התכבות במייל



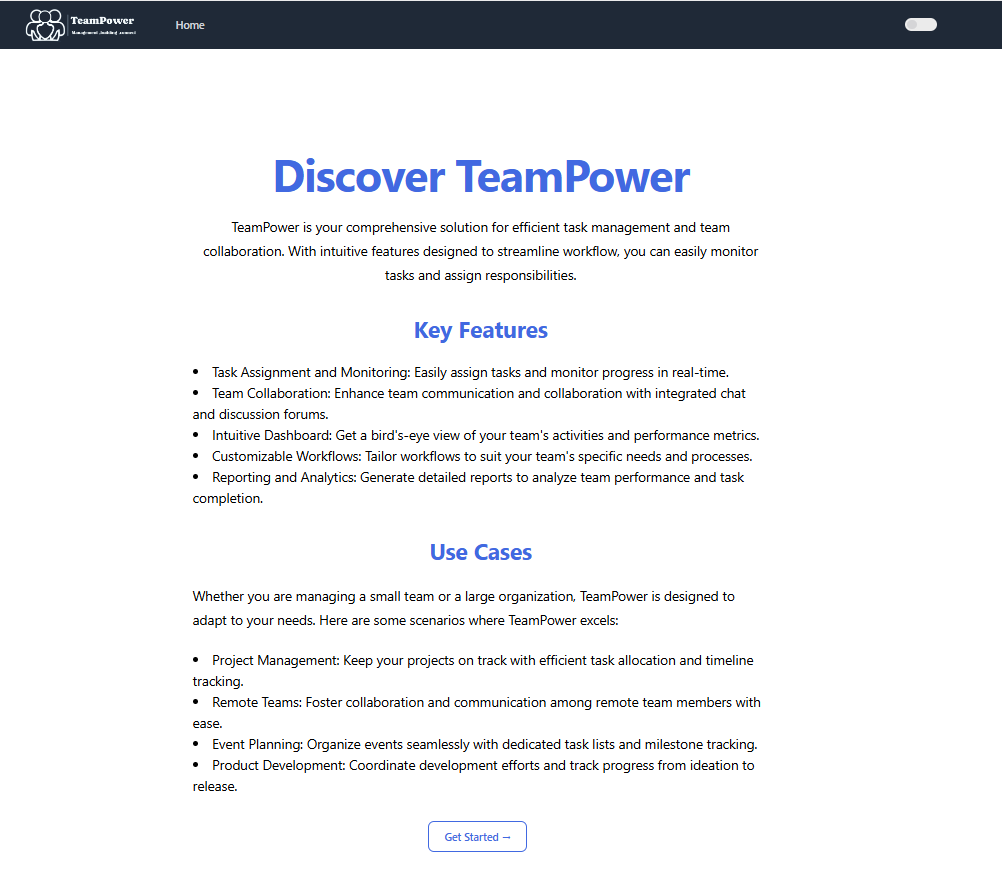
* 1. שאלות ותשובות (FAQs)

בדף זה נמצאים התשובות לשאלות הנפוצות בקרב המשתמשים שלנו



* 1. למד עוד (Learn more)

הדף הזה מכיל עוד מידע על האתר שלנו והפרויקט



* 1. מסכי משתמש מסוג מנהל:
* מסך הראשי למנהל

לאחר ההתחברות המנהל שלנו מגיע למסך הראשי יראה את כל הפעילות שהוא יכול לבציע באפליקציה שלנו:

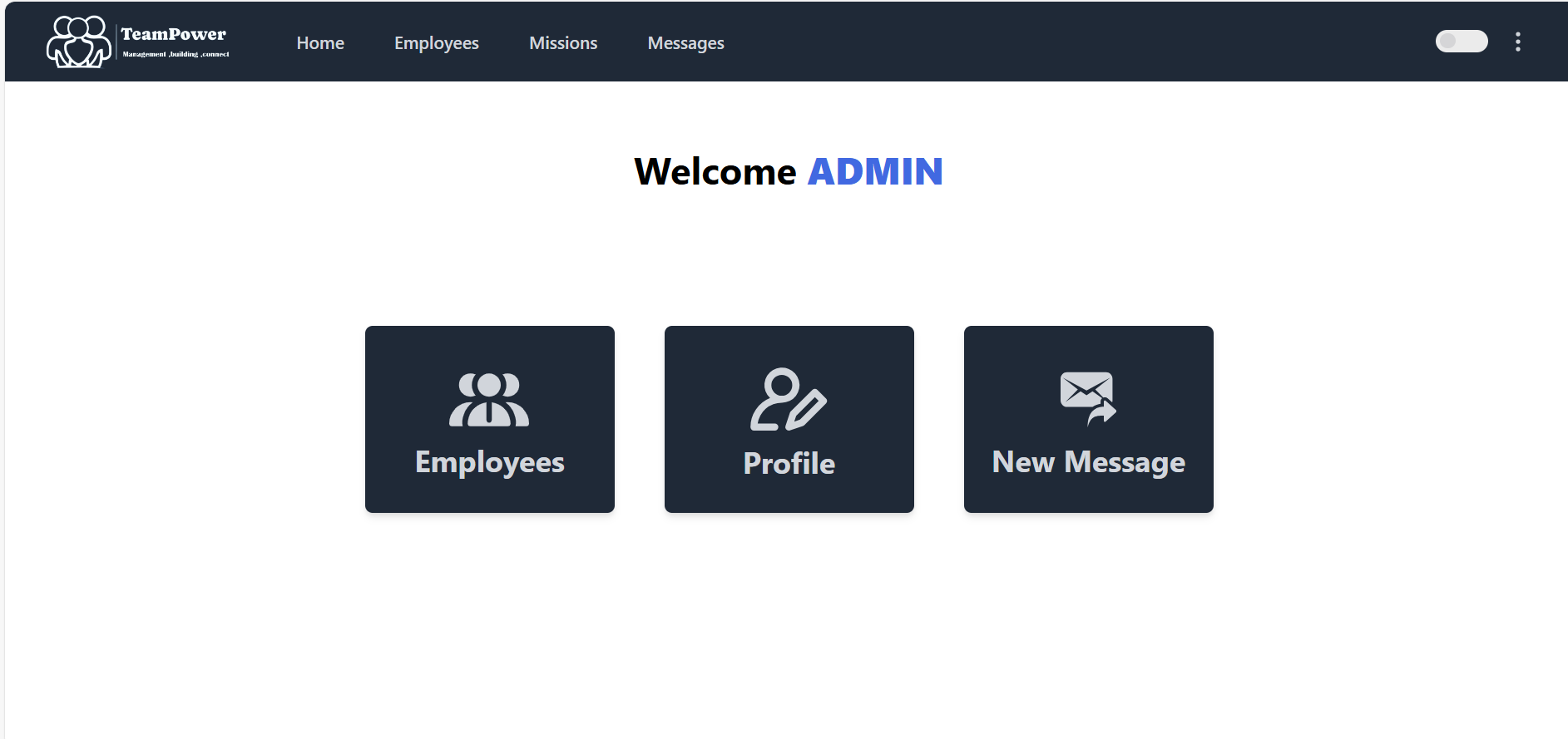
הוא יכול להפוך את המסך למצב כהה או בהיר.

הוא יכול לכנס לפרופיל האישי שלו ולערוך אותו כרצונו, הוא יכול לכנס גם דרך לחיצה על שלוש הנקודות ולבחור את האפשרות של פרופיל

הוא יכול להתנתק מהמערכת ברגע שהוא לוחץ על שלוש הנקודות ובוחר את אפשרות ההתנתקות

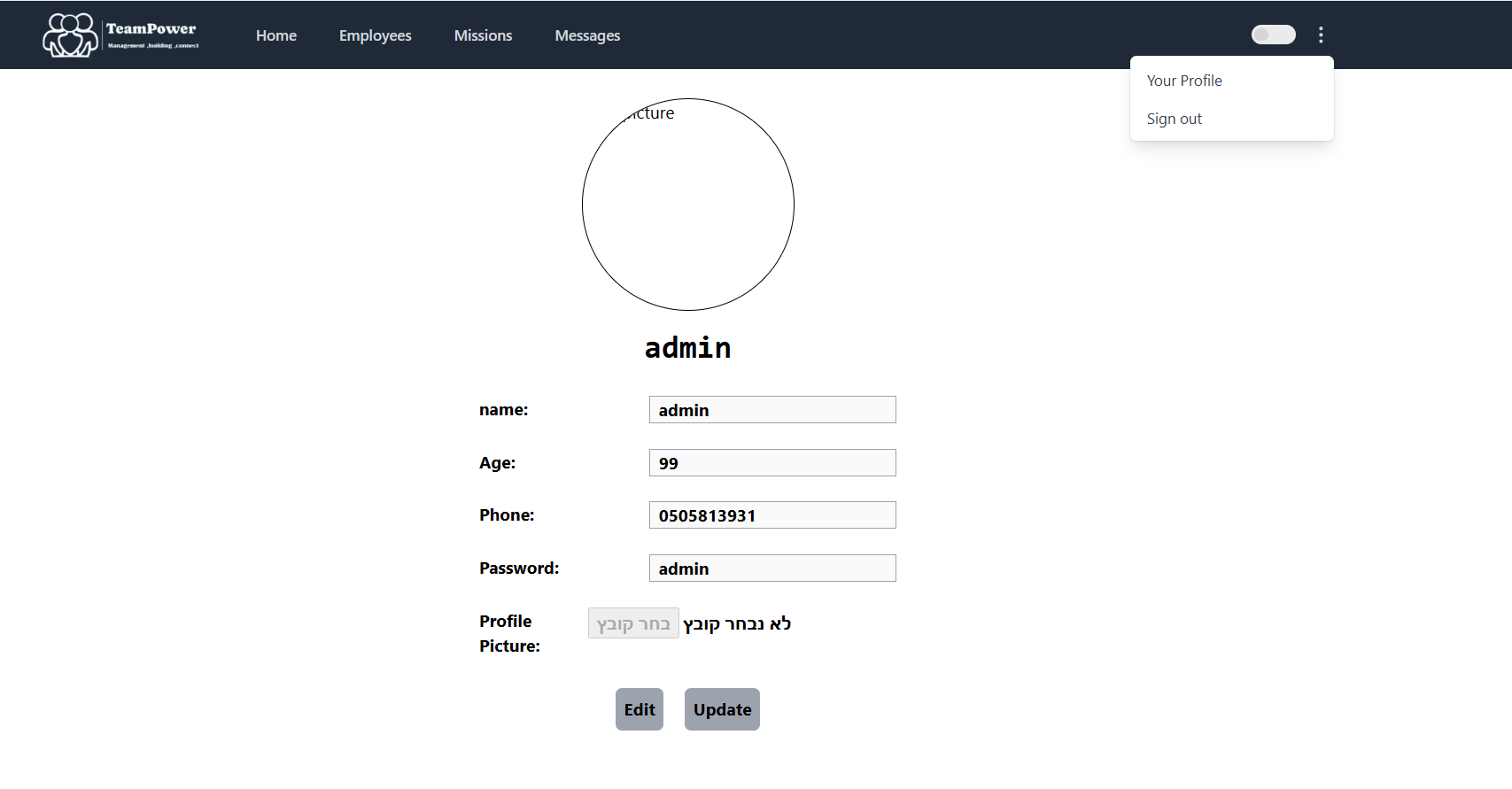
הוא יכול לכנס לדף ניהול העובדים ודף ניהול משימות ודף הודעות.

הוא יכול לכנס לדף שליחת הודעות ולשלוח הודעות לכלל העובדים שלו.



* מסך פרופיל

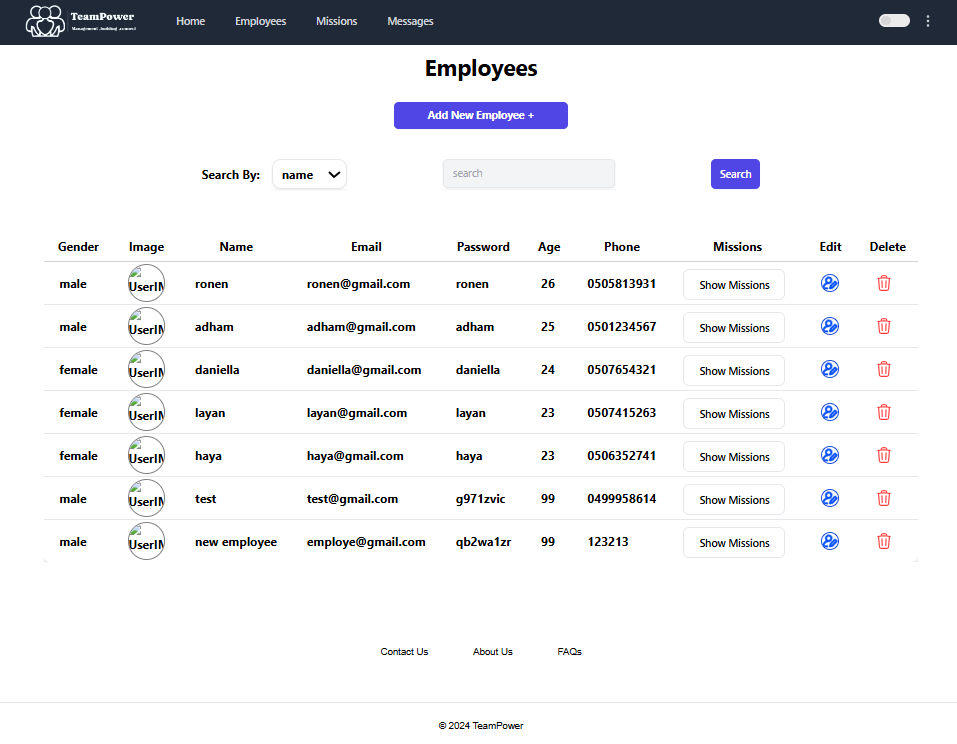
במסך זה המשתמש יוכל לערוך את הפרופיל האישי שלו, נוסף לכך הוא יכול להעלות תמונת פרופיל.



* מסך ניהול עובדים

המסך הזה מספק מספר פעולות לניהול עובדים. המנהל יוכל להוסיף עובד חדש לאחר הלחיצה על כפתור ההוספה ומילוי הפרטים הנדרשים, בנוסף לכך הוא יכול למחוק, לערוך פרטי עובד. כאשר המנהל ירצה לחפש עובד מסוים אשר נמצא בטבלה הוא יכול לבציע חיפוש על פי פרט הסינון.

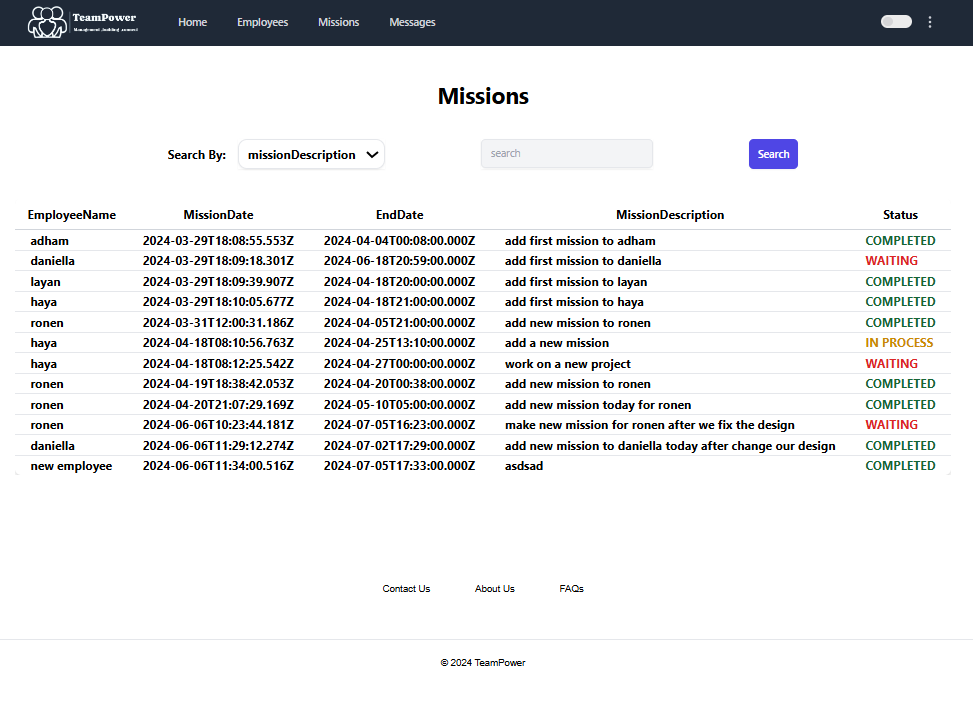
ועל ידי לחיצה על כפתור הראה משימות של עובד מסוים, הוא יכול לראות את כל המשימות שהוקצו לאותו עובד עם סטאטוס משימה ושמה יכול להקצות לעובד זה משימה חדשה או למחוק משימה.



* מסך ניהול משימות עובדים

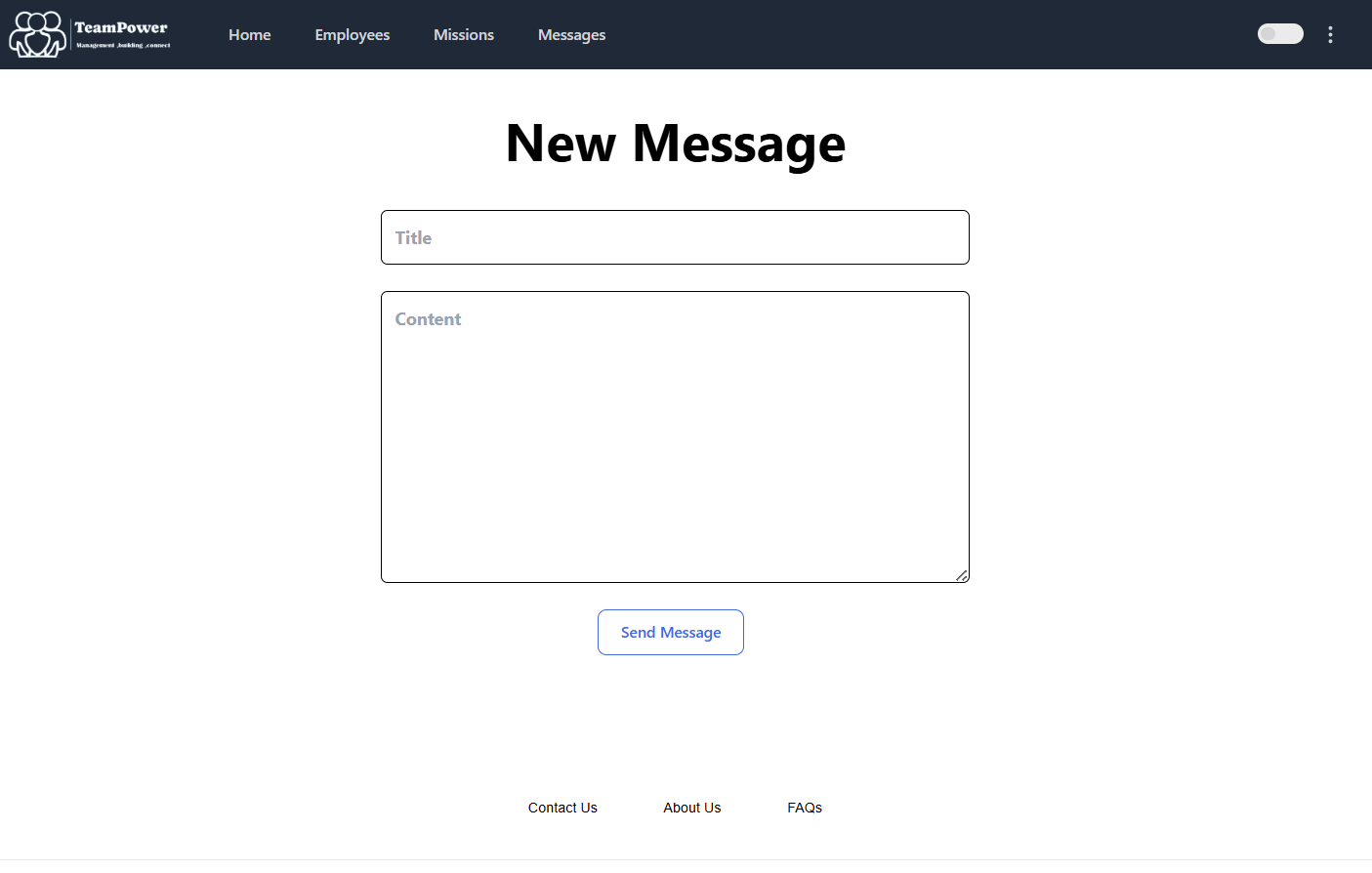
במסך הזה מופיעות כל המשימות שהמנהל הקצה לעובדים שלו, בטבלה מופיעים העובד שאחראי על ביצוע המשימה תאריך בו הקצה את המשימה, תאריך הסיום שהוא קבע לאותה משימה, תיאור המשימה והסטטוס של המשימה (בתהליך – העובד קבל את המשימה והיא בתהליך העבודה, הַמתָנָה – העובד עדין לא פתח את המשימה , הושלם – העובד סיים את המשימה וסמן אותה שהושלמה).

בנוסף לאפשרות החיפוש והסינון.



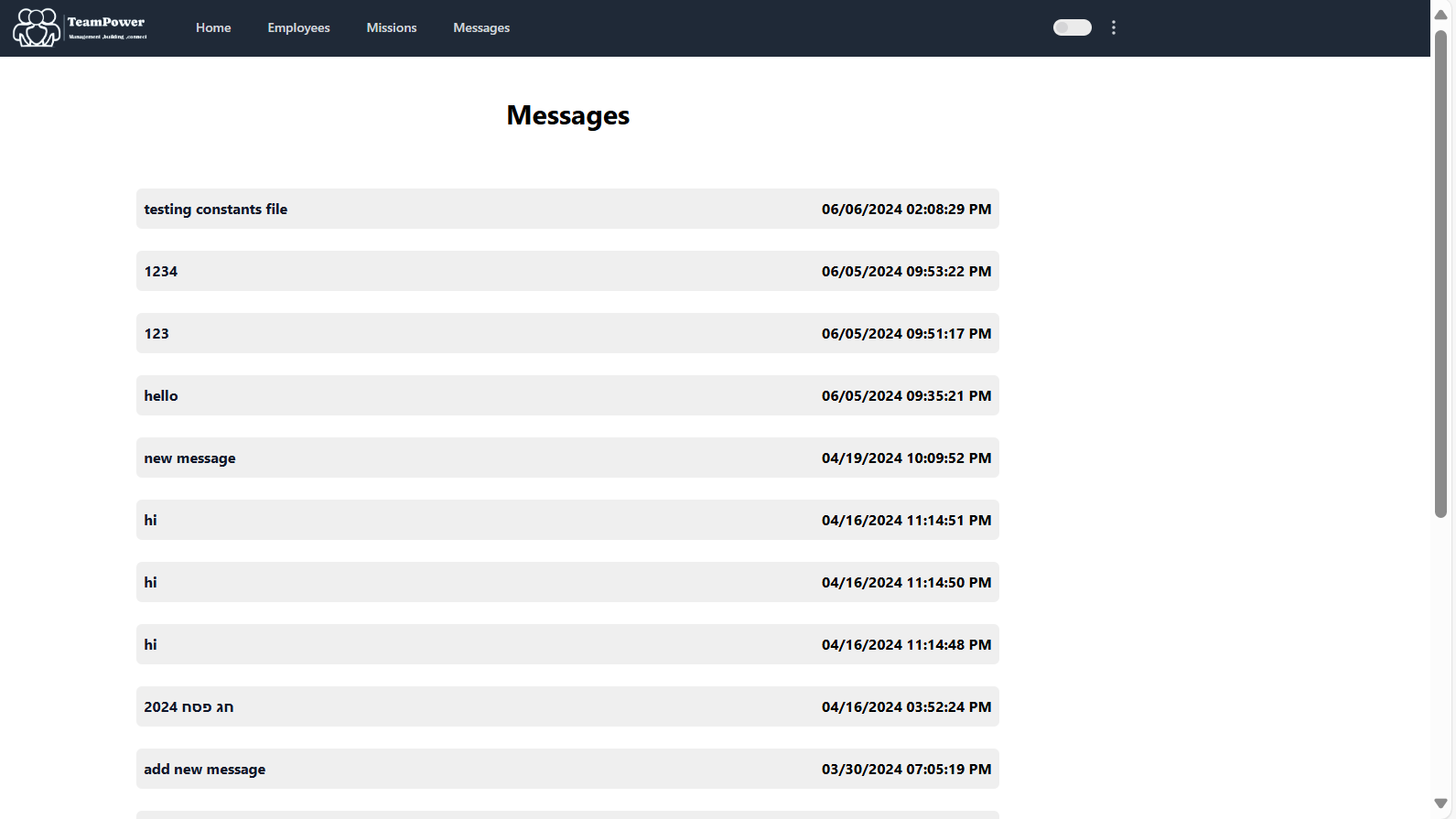
* מסך שליחת הודעות לעובדים

על המנהל למלאות את כל השדות, ולאחר מכן ללחוץ על כפתור שלח הודעה , ההודעה תופיע במסך של ההודעות בטבלה. והיא תשלח לכל העוב דים של המנהל.



* מסך הודעות

מסך זה מופיעות בו כל ההודעות ששלח לעובדים שלו. מופיע מולו הנושא של ההודעה עם התאריך והשעה של שלחת ההודעה. ברגע שהוא לוחץ על ההודעה התיאור שלה יפתח.



* 1. מסכי משתמש מסוג עובד:
* מסך הראשי לעובד

לאחר ההתחברות העובד שלנו מגיע למסך הראשי את כל הפעילות שהוא יכול לבציע באפליקציה שלנו. ברגע שהעובד נכנס הוא יכול לראות כמה הודעות חדשות יש לו וגם כמה משימות חדשות יש לו.

הפעילות שהוא יכול לבציע באפליקציה שלנו:

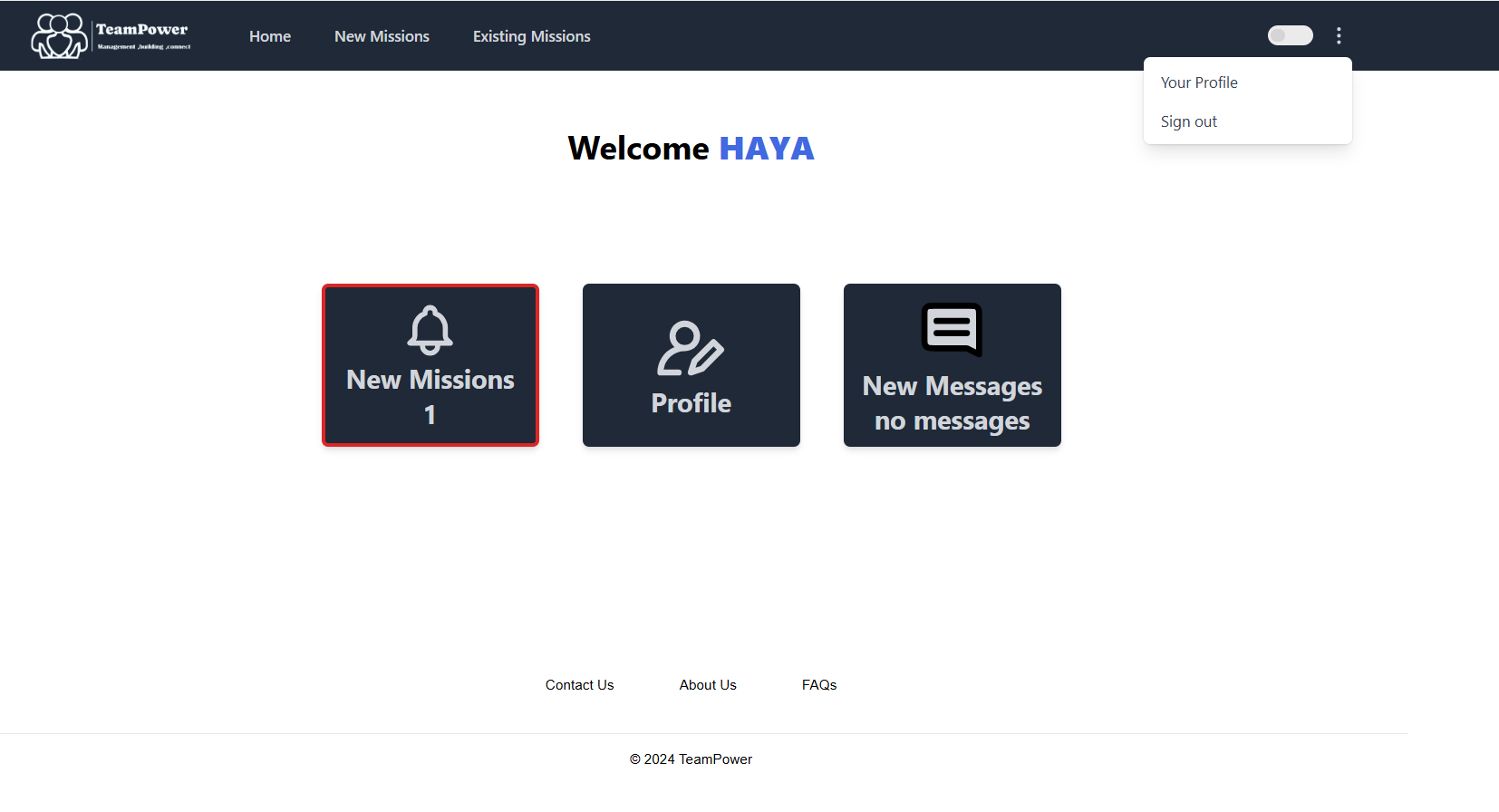
הוא יכול להפוך את המסך למצב כהה או בהיר.

הוא יכול לכנס לפרופיל האישי שלו ולערוך אותו כרצונו, הוא יכול לכנס גם דרך לחיצה על שלוש הנקודות ולבחור את האפשרות של פרופיל

הוא יכול להתנתק מהמערכת ברגע שהוא לוחץ על שלוש הנקודות ובוחר את אפשרות ההתנתקות

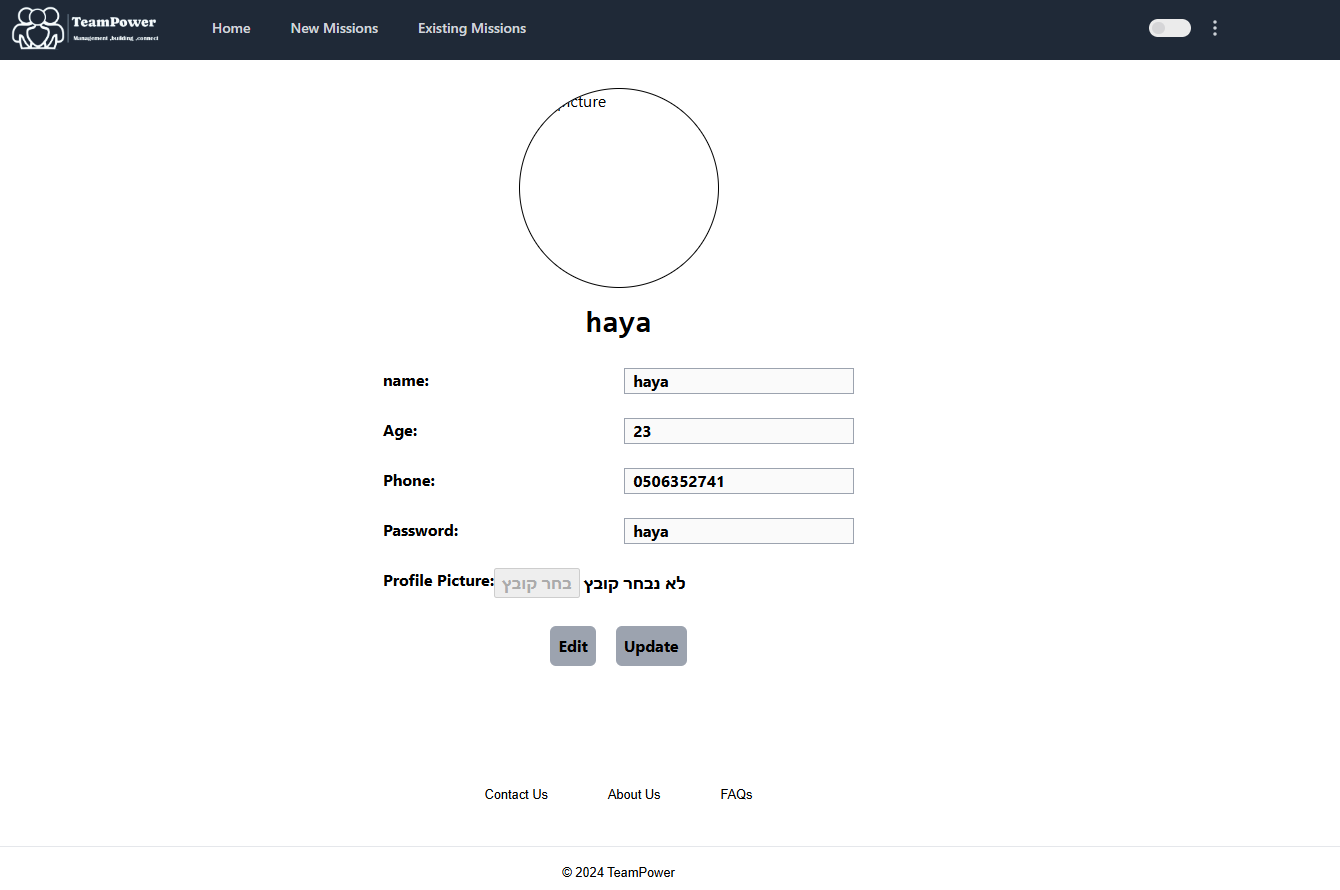
הוא יכול לכנס לדף משימות קיימות ודף המשימות החדשות.

הוא יכול לכנס לדף הודעות מנמהל.



* מסך פרופיל

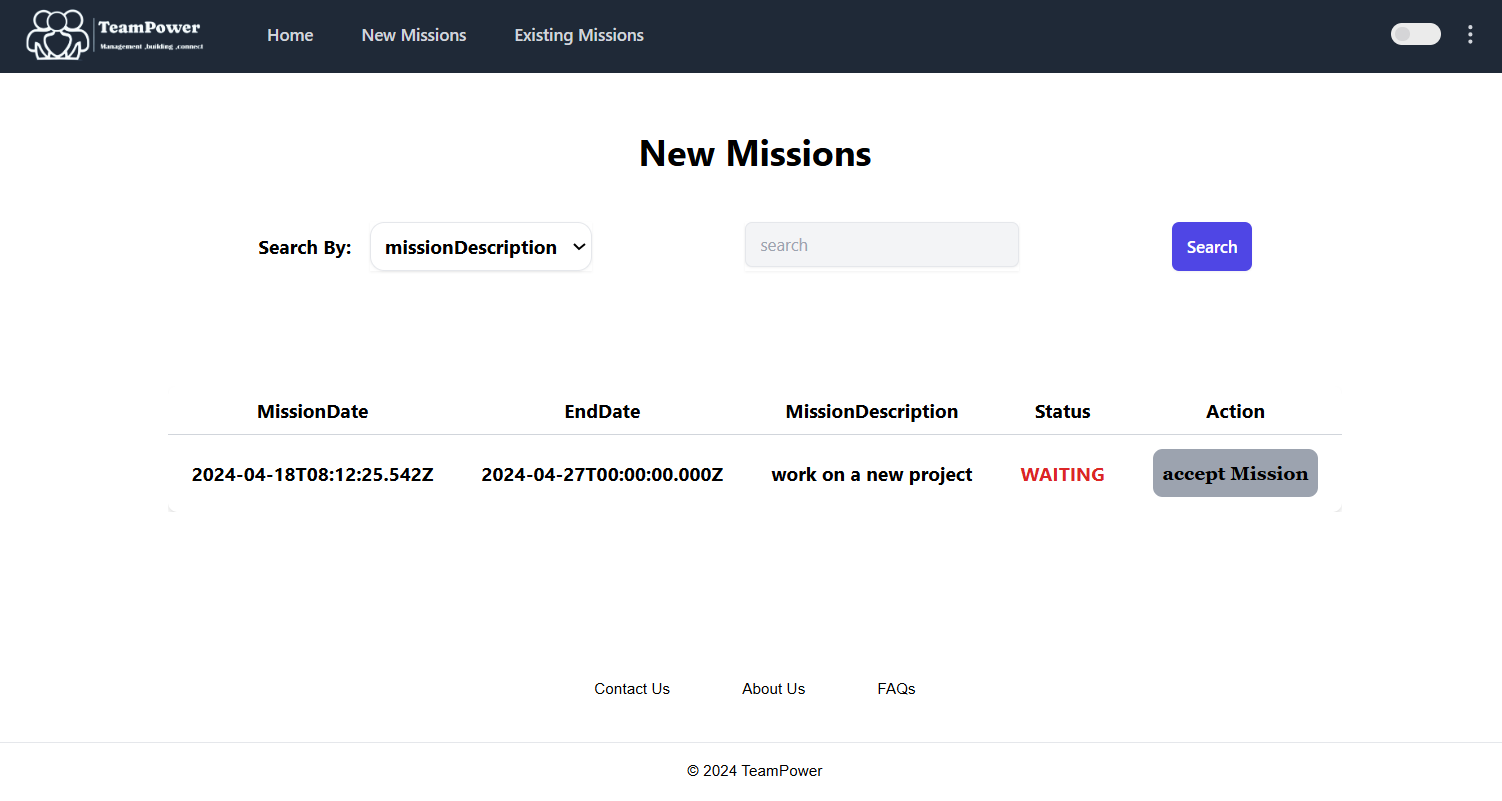
במסך זה המשתמש יוכל לערוך את הפרופיל האישי שלו, נוסף לכך הוא יכול להעלות תמונת פרופיל. [[2]](#footnote-2)



* מסך משימות חדשות

מסך זה מכיל את המשימות החדשות שהתקבלו מהנהל

מיוצגת מול העובד טבלה שמכילה את תאריך הקצאת המשימה, מועד הסיום שלה, תיאור של המשימה, סטטוס המשימה (הַמתָנָה – העובד עדין לא פתח את המשימה) וכפתור קבל משימה תופיע מול העובד ההתראה הבאה עליו לאשר את קבלת המשימה. בנוסף, יש לו את האפשרות לחפש משימה על ידי בחירת פרט הסינון (תיאור משימה), לאחר מכן ממלאים את שדה החיפוש ובעת הלחיצה כפתור החיפוש

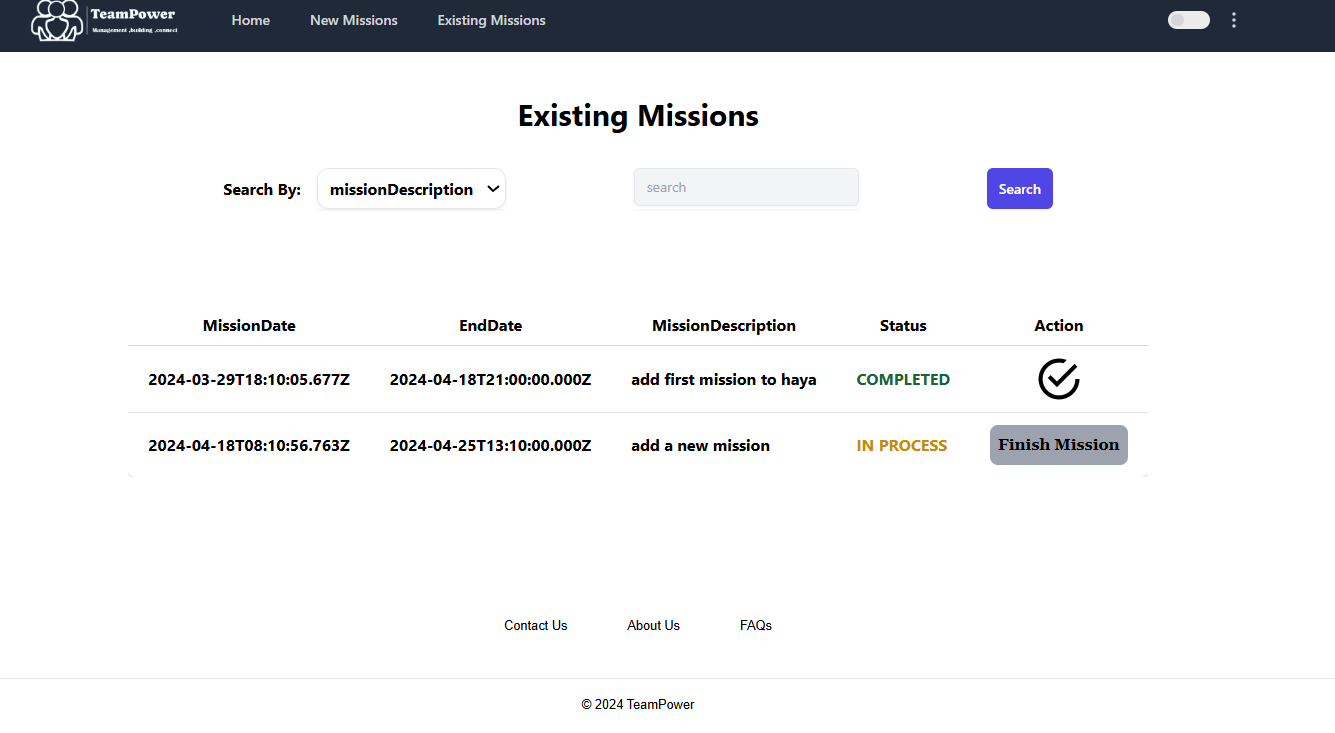


* מסך משימות קיימות בתהליך או שהושלמו

מסך זה מכיל את המשימות הקיימות בתהליך או שהושלמו

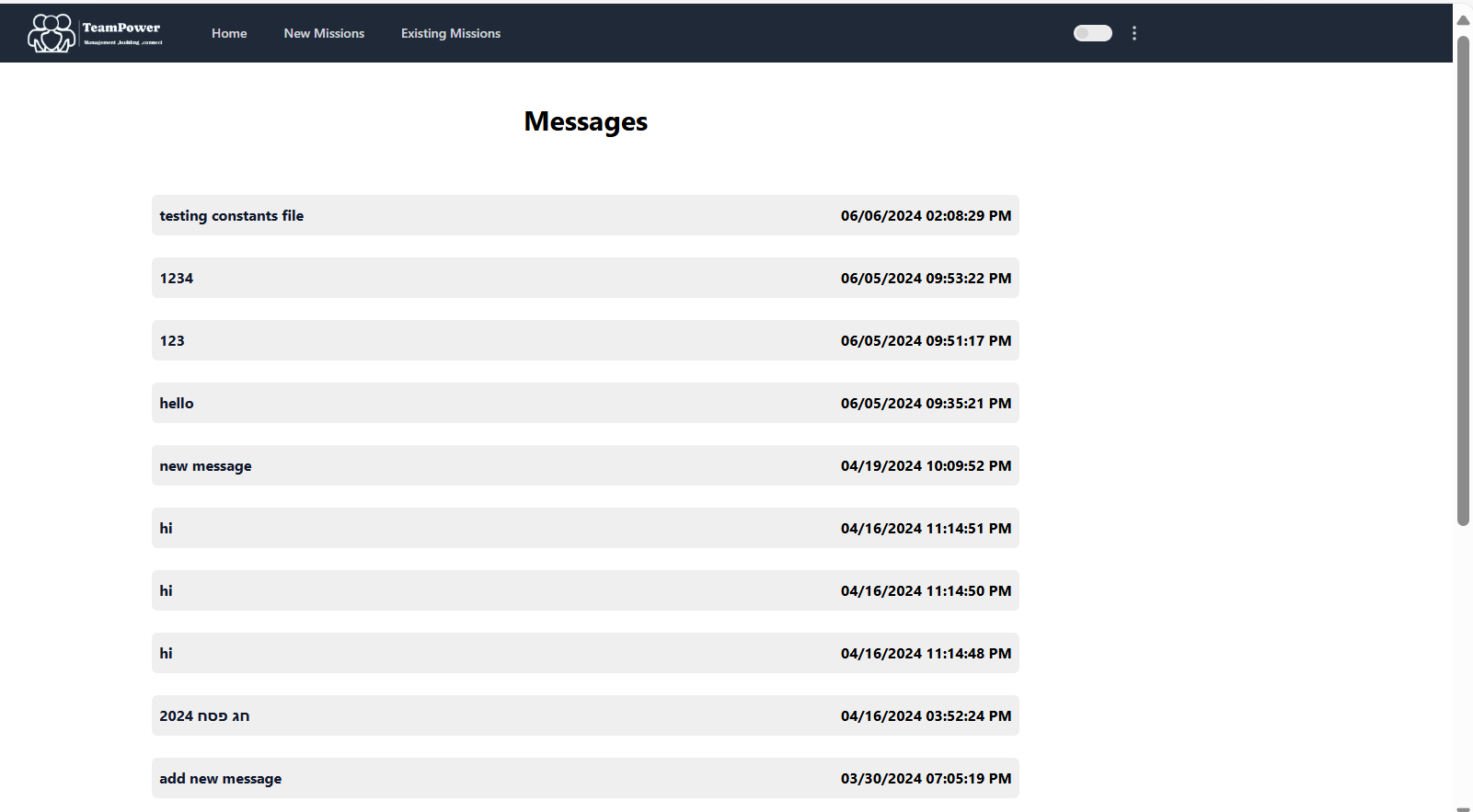
מיוצגת מול העובד טבלה שמכילה את תאריך הקצאת המשימה, מועד הסיום שלה, תיאור של המשימה, סטטוס המשימה (בתהליך – העובד קבל את המשימה והיא בתהליך העבודה או הושלם – העובד סיים את המשימה וסמן אותה שהושלמה) במקרה שהעובד סיים את המשימה עליו ללחוץ על כפתור סיים משימה, תופיע מול העובד ההתראה הבאה עליו לאשר את סיום המשימה.

בנוסף, יש לו את האפשרות לחפש משימה על ידי בחירת פרט הסינון (תיאור משימה), לאחר מכן ממלאים את שדה החיפוש ובעת הלחיצה כפתור החיפוש



* מסך הודעות

במסך זה מופיעות כל ההודעות שנשלחו על ידי המנהל. מופיע מולו הנושא של ההודעה עם התאריך והשעה של שלחת ההודעה. ברגע שהוא לוחץ על ההודעה התיאור שלה יפתח.



1. רפרנסים

* "React JS - React Tutorial for Beginners"

<https://www.youtube.com/watch?v=IKQQIYDfyPc>

* "Learn React JS - Full Course for Beginners - Tutorial 2022" <https://www.youtube.com/watch?v=ZVyIIyZJutM>

הסרטונים האלה סיפקו הדרכה מקיפה בנושאים שונים כמו יסודות React, ניהול מצב, ניתובים וכן הלאה, בליווי דוגמאות קוד.

* ChatGPT - נעזרנו לעיתים בכלי אינטליגנציה מלאכותית לסיוע בפתרון בעיות ותיקון באגים בקוד שכתבנו.

1. **הערה:** התחלנו את המשימה הראשונה על בסיס הקידום הרעיוני של האתר שלנו, המתמקד במשחקים וחדרים. אולם, לאחר קבלת פידבק והתייעצות עם המרצים, הוחלט לשנות את הנושא של הפרויקט ליצירת משימות לעובדים. על פי כך, אנו מפנים את כל המשאבים והעבודות הקודמות שלנו לנושא החדש שקיבלנו.( הנושא החדש אושר על ידי המרצה מר אלכס קסלמן ובמודעות ובאישורה של המרצה ד"ר נעמי אונקלוס-שפיגל) [↑](#footnote-ref-1)
2. **הערה -** עקב זה שהשתמשנו ב - vercel נכון להגדרות Vercel, הוא מתאים בעיקר לאפליקציות סטטיות ולכן לא מספק אפשרות להעלאת קבצים דינמיים כמו תמונות. בניגוד לכך, ב-local host המחשב האישי שלנו האפשרות עובדת ניתן לשמור ולהעלאת תמונות. [↑](#footnote-ref-2)