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

**הגשת פרויקט**

**Social network B21 group15**

|  |  |
| --- | --- |
| **שם חבר/ת הצוות** | **ת"ז** |
| קטרין פרימוס | 207410093 |
| רפאל אבסוב | 207187816 |
| ליטל חיבטוב | 208110973 |
| דביר חייט | 318400165 |

**תקציר הפרויקט** :  
מטרת האתר היא הצגת קשרים חברתיים.

כל אדם יכול להירשם למערכת לאחר מילוי הפרטיים הרלוונטיים.

לאחר ההרשמה ניתן להתחבר למערכת ע"י תעודת זהות וסיסמא, במידה ופרטי ההתחברות נכונים **המשתמש** יעבור לדף הבית.

בדף הבית יוכל לצפות בגרף הקשרים כאשר כל קו מחבר בין צמתים הינו סוג קשר המחבר, המשתמש יכול לסנן את הגרף לפי סוג הקשר שבו רוצה להתמקד.

בנוסף ניתן לעבור לפרופיל ובו יוצג למשתמש פרטיו האישיים וכל החברים אליו הוא מקושר. המשתמש יכול לשלוח מייל במידה וירצה.

בכל עת יכול המשתמש לחזור לגרף, לעבור בין מצב לילה ליום ולהתנתק מהמערכת.

משתמש המוגדר **כAdmin** יכול לעשות את כל הפונקציונליות שמשתמש רגיל יכול לעשות ובנוסף יכול להוסיף או למחוק תחביבים ומקומות עבודה ובנוסף למחוק מהמערכת משתמשים לא רצויים.

**הטכנולוגיות שבהן השתמשנו:**

Frontend מומש באמצעות react, ו Tailwind- **.**

הbackend מומש באמצעות Next.js.

הDB שבחרנו להשתמש הינו mongoDb.

**קישורים:**

* GIt repository - [Front](https://github.com/RafaelAbb/SocialNetwork) [Back](https://github.com/DvirHayat/WEB-Course-backend)
* [Vercel](https://social-network-bfr9.vercel.app/)
* [MoreThanWallet.com Gallery](https://www.morethanwallet.com/app/887)

1. מהנדס מערכת : ליטל חיבטוב

|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** |
| ליטל חיבטוב | עמוד אדמין, עמוד פרופיל, הגדרת דרישות, דיאגרמת פרונט | V |
| דביר חייט | סרבר, גרף , ארכיטקטורות,דיאגרמת בק | V |
| רפאל אבסוב | עמוד הרשמה , עמוד כניסה ,תיק מתכנת ,מבנה סרבר | V |
| קטרין פרימוס | עמוד מידע , תפריטים , darklight , usecase ת תיק משתמש | V |

היה ממשק בין חברי הצוות במהלך העבודה. כל אחד ידע מה השני עושה והיה תיאום בביצוע העבודה. אחת לכמה ימים הסברנו את השינויים שבוצעו אחד לשני.

2.**דרישות**:

דרישות פונקציונליות:

1. המערכת מאפשרת כניסה והתחברות.
2. המערכת מאפשרת צפייה בגרף.
3. המערכת מאפשרת צפייה בפרופילים של אנשים אחרים.
4. המערכת מאפשר הוספה והורדה של תחביבים ומקומות עבודה.
5. המערכת מאפשרת הסרה של משתמשים .
6. המערכת מאפשרת התממשקות מול MongoDB.

דרישות לא פונקציונליות:

1. הכניסה למערכת תתבצע באמצעות שם משתמש וסיסמה. (Privacy)

2. הקשרים החברתיים הם : תחביבים, מקום עבודה ומדינה. (Extensibility)

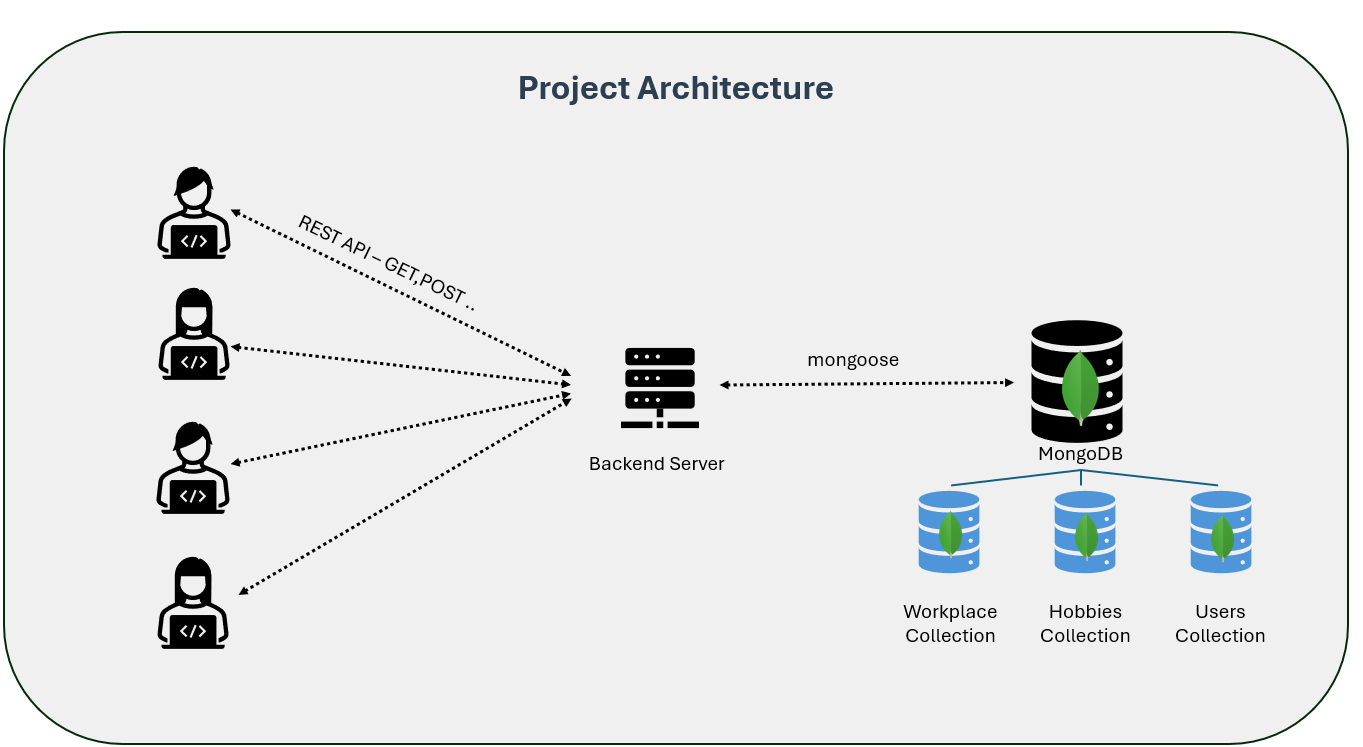
3. הפרטים שנשמרים ע"י במסד הנתונים הם : תעודת זהות סיסמה , שם ושם משפחה , תחביב , מקום עבודה , מדינה ומין. (Data integrity)

4. המערכת צריכה להיות מסוגלת להתרחב כדי לתמוך במספר הולך וגדל של משתמשים ללא ירידה משמעותית בביצועים. (Adaptability)

5. הממשק למערכת יהיה אינטואיטיבי וקל לפי תבניות ממשק מוכרות.( Usability)

6. הגבלת גישה לפעולות תהיה לפי תפקיד (משתמש רגיל או אדמין).( Accessibility)

7. המערכת צריכה להיות תואמת למכשירים ניידים ורזולוציות. (Platform Compatibility)

**ארכיטקטורת המערכת:**ארכיטקטורת האתר מופיע באופן מפורט בסעיף 4.

3. use case :

תמונה שמכילה טקסט, תרשים, קו, צילום מסך

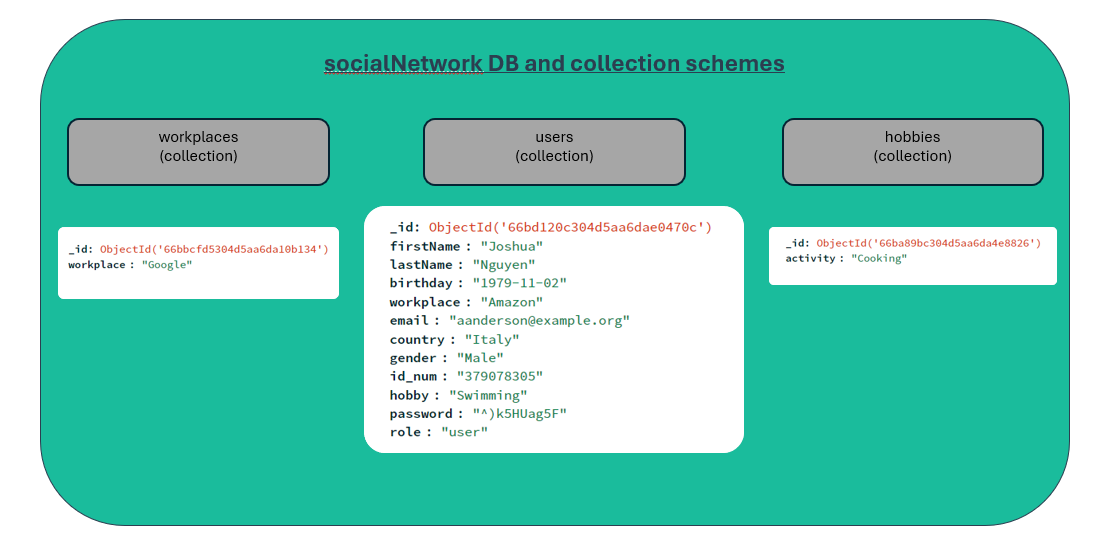
התיאור נוצר באופן אוטומטי

**4. יש להציג מבנה סופי של האתר שלכם:**

A screenshot of a computer program

Description automatically generatedתמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטיfrontend מומש באמצעות react, ו Tailwind- **.** הbackend מומש באמצעות Next.js ו mongoDb.



**5. תיק מתכנת**:  
ניתן למצא גרסה איכותית יותר של תיקי המתכנת בתיקיות הgit של הפרויקט.

This web application connects people based on their work, country, and favorite hobby. Users can view their connection graph and send emails to their connections. Using this platform, they can really get to know people with common interests.

#### Table of Contents

* Project Overview
* Technologies Used
* Dependencies
* Installation and Setup
* Running the Project
* Code Structure (Frontend)
* Code Structure (Backend)
* Main Files
* API Endpoints
* Usage
* Example Code
* Additional help

#### Project Overview

This application provides a platform for users to connect with others who share similar professional and personal interests. Users can visualize their connections, manage their profiles, and communicate with their network.

**Technologies Used**

A screenshot of a computer

Description automatically generated

#### Dependencies

List of dependencies used in this project:

**Frontend:**

* React
* Tailwind CSS

**Backend:**

* Node.js
* Next.js
* MongoDB

#### Installation and Setup

* + - 1. **Ensure you have the required dependencies and npm installed.**
      2. **Clone the frontend repository** to your local machine by running the command:  
          git clone <https://github.com/RafaelAbb/SocialNetwork.git>.
      3. **Clone the backend repository** to your local machine by running the following commands: git clone https://github.com/DvirHayat/WEB-Course-backend  
         cd next-backend to navigate into the backend directory.
      4. **Navigate to the frontend project directory** by running the command:   
         cd socialnetwork/frontend.
      5. **Install the required dependencies** by running the command:   
         npm install.
      6. **Repeat the previous step** for the backend directory by running the following commands:   
         cd socialnetwork/WEB-Course-backend   
         npm install

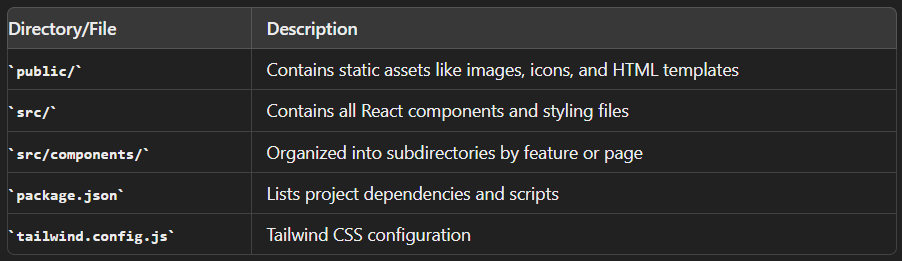
#### Set up environment variables for backend:

Create a .env.local file in the root directory and add your MongoDB connection string:  
MONGODB\_URI=your-mongodb-uri  
or change db\_URI in lib/mongodb.js

#### Running the Project

* **To run the project in development mode**, use the command: npm run dev.
* **To create a production build**, use the command: npm run build.
* **To start the production server**, use the command: npm start.
* **Ensure you are in the correct directory** by navigating to either the frontend directory with cd socialnetwork/frontend or the backend directory with cd socialnetwork/WEB-Course-backend.

#### Code Structure (Frontend)



#### Code Structure (Backend)

A screenshot of a computer

Description automatically generated

#### Main Files

##### Frontend Files

* **Menu.jsx** implements the sidebar or dropdown menu for navigation. This component is often used to toggle between different sections or features within the application.
* **DarkLight.jsx** is a component that enables users to toggle between dark and light themes. It manages the theme state and applies the appropriate styles across the application.
* **profilePage/ProfilePage.jsx** is the main component for displaying and managing user profiles. It includes functionality for viewing user details, editing profiles if the user is an admin, and contacting other connections.
* **homePage/GraphComponent.jsx** renders the connection graph on the homepage. It visualizes user connections based on shared interests like work, country, and hobbies.
* **homePage/GraphOptions.jsx** provides filtering and customization options for the connection graph, allowing users to adjust settings of the graph.
* **registerPage/RegisterForm.jsx** handles user registration by collecting information such as work, country, and hobbies. It manages the form submission process and communicates with the backend to create a new user account.
* **signInPage/SignIn.jsx** manages the user authentication process. This component includes the login form where users provide their credentials (email and password) to access their accounts. It interacts with the backend to validate user credentials and handle sessions.
* **AdminPage/AdminPage.jsx** is the main component for administrative tasks within the application. It provides a dashboard interface where admins can manage users, hobbies, and workplaces. Admins can perform tasks like deleting users, adding new hobbies, and more.
* **tailwind.config.js** is the configuration file for Tailwind CSS, allowing customization of themes, extension of the framework with additional utilities, and integration of plugins specific to the project’s needs.

##### Backend Files

* **login.js** handles user login by verifying the user ID and password. The file connects to the MongoDB database, authenticates users, and returns user details if the credentials are valid.
* **register.js** manages user registration. It handles POST requests to register new users, ensuring no duplicate emails or IDs are added to the database, and GET requests to provide a list of available hobbies and workplaces for the registration process.
* **userUtil.js** provides utility functions to fetch user connections and details. It allows users to find connections based on shared fields such as workplace, country, and hobby.
* **adminUtil.js** contains administrative functions, allowing admins to fetch all users via GET requests and delete users by email via DELETE requests.
* **hobbyUtil.js** manages hobbies within the application. It handles GET requests to retrieve all hobbies, PUT requests to add a new hobby, and DELETE requests to remove a hobby by name (only through admin).
* **workplaceUtil.js** manages workplaces within the application. It handles GET requests to retrieve all workplaces, PUT requests to add a new workplace, and DELETE requests to remove a workplace by name (only through admin).

#### API Endpoints

Here are some key API endpoints used in the application:

* **/api/login**
  + **Method:** POST
  + **Description:** Authenticates a user with their id\_num and password.
  + **Request Body:** JSON object containing the user's ID number and password.
  + **Response:** Returns user details if authentication is successful (200 OK). If the credentials are invalid, it returns a 401 Unauthorized response.
* **/api/register**
  + **Method:** POST
  + **Description:** Registers a new user.
  + **Request Body:** JSON object containing user details such as first name, last name, email, ID number, country, workplace, hobby, gender, password, and birthday.
  + **Response:** Returns 201 Created if the user is registered successfully. If any required fields are missing or the email/ID already exists, it returns 400 Bad Request.
* **/api/userUtil**
  + **Method:** GET
  + **Description:** Retrieves user connections based on shared attributes.
  + **Request Parameters:** ID number is required, with optional parameters for workplace, country, and hobby.
  + **Response:** Returns an array of users with shared attributes (200 OK).
* **/api/adminUtil**
  + **Method:** GET
  + **Description:** Fetches all non-admin users in the database.
  + **Response:** Returns an array of users (200 OK).
  + **Method:** DELETE
  + **Description:** Deletes a user by email.
  + **Request Body:** JSON object containing the user's email.
  + **Response:** Returns 200 OK if the user is deleted successfully. If the user is not found, it returns 404 Not Found.
  + **Method:** PATCH
  + **Description:** Deletes all users from the database (for debugging purposes).
  + **Response:** Returns 200 OK if all users are deleted successfully.

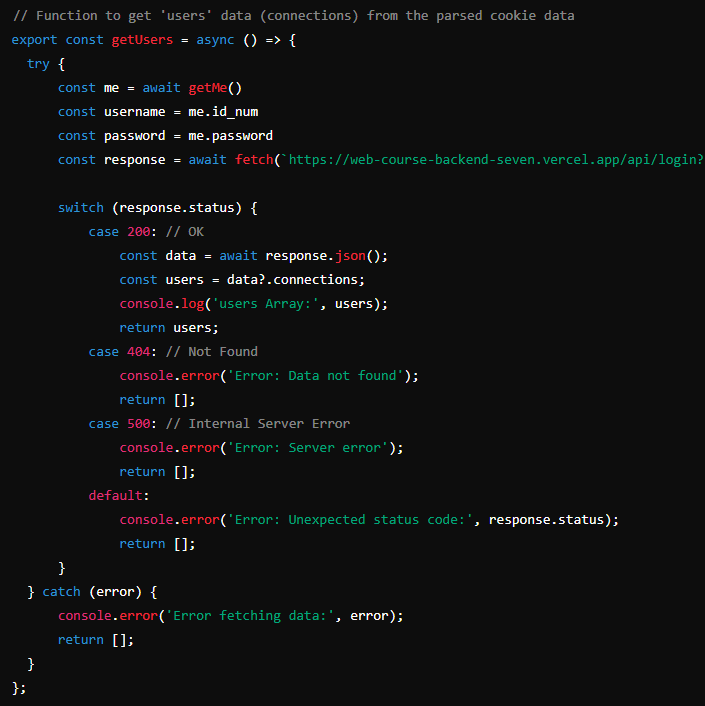
##### Hobby Endpoints

* **/api/hobbyUtil**
  + **Method: GET**
  + **Description: Fetches all hobbies.**
  + **Response: Returns an array of hobbies (200 OK).**
  + **Method: PUT**
  + **Description: Adds a new hobby.**
  + **Request Body: JSON object containing the hobby name.**
  + **Response: Returns 201 Created if the hobby is added successfully. If the hobby name is missing, it returns 400 Bad Request.**
  + **Method: DELETE**
  + **Description: Deletes a hobby by name.**
  + **Request Body: JSON object containing the hobby name.**
  + **Response: Returns 200 OK if the hobby is deleted successfully. If the hobby is not found, it returns 404 Not Found.**

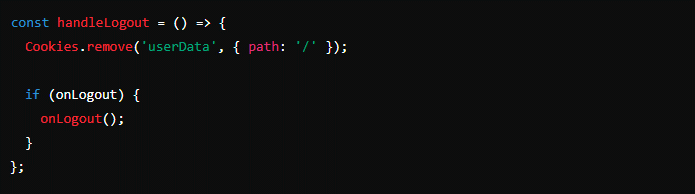
##### Workplace Endpoints

* **/api/workplaceUtil**
  + **Method: GET**
  + **Description: Fetches all workplaces.**
  + **Response: Returns an array of workplaces (200 OK).**
  + **Method: PUT**
  + **Description: Adds a new workplace.**
  + **Request Body: JSON object containing the workplace name.**
  + **Response: Returns 201 Created if the workplace is added successfully. If the workplace name is missing, it returns 400 Bad Request.**
  + **Method: DELETE**
  + **Description: Deletes a workplace by name.**
  + **Request Body: JSON object containing the workplace name.**
  + **Response: Returns 200 OK if the workplace is deleted successfully. If the workplace is not found, it returns 404 Not Found.**
* **next.config.js is used to configure Next.js to rewrite API routes to the backend URL. This is essential for routing requests during development or in production.**

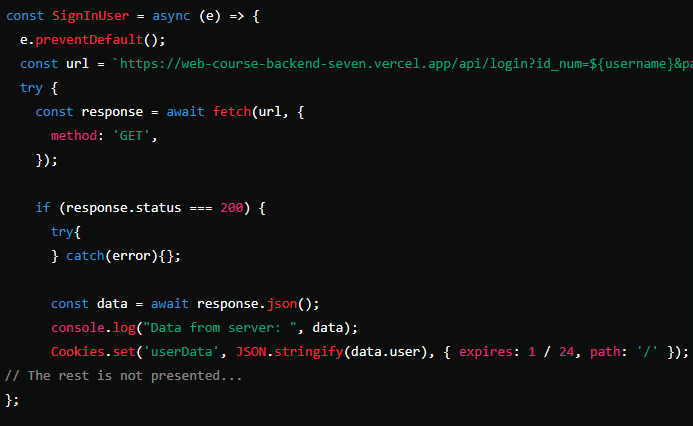
#### Example Code



Here is an example of getting connections using the server API:



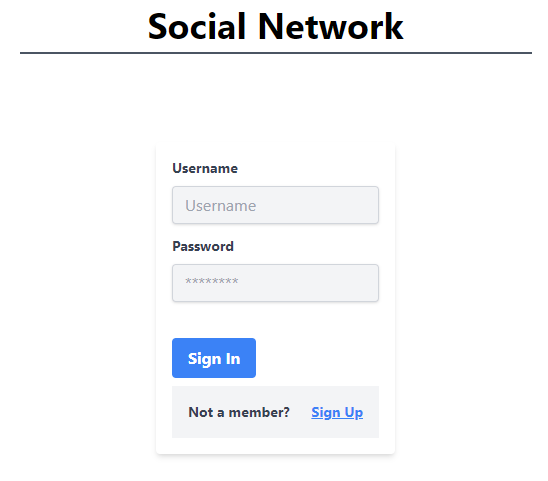
Here is an example of logging out and deleting the cookie of the logged user:



Here is an example of sign-in validation using the server API; after the user is validated, the user data is saved using cookies:

#### Additional Help

For more detailed guidelines, please refer to DvirTheking@braude.ac.il mail.

**6.תיק משתמש:**

1. כניסה לאתר :

הכניסה לאתר מתבצעת בעזרת שם משתמש (תעודת הזהות ), וסיסמה

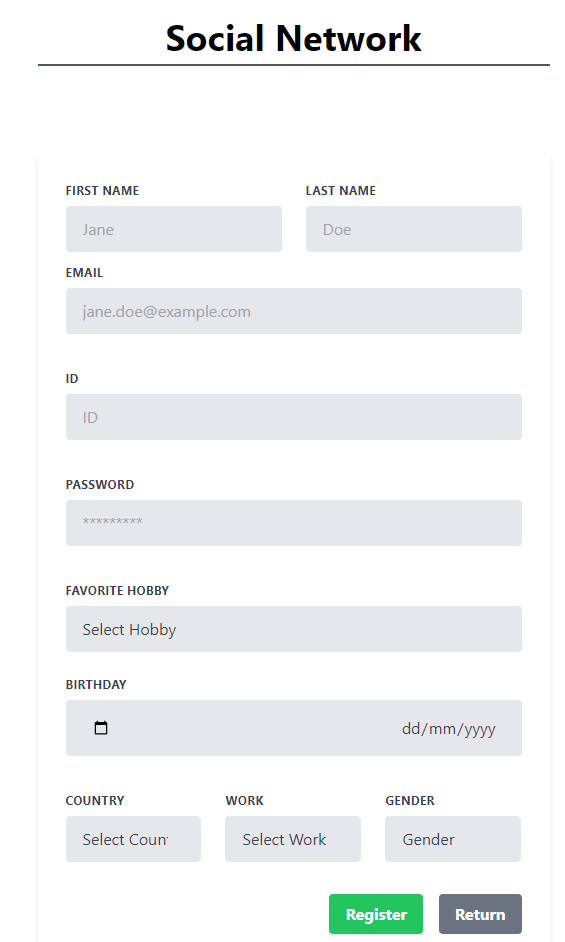
משתמשים לדוגמא :

שם משתמש : 321321321

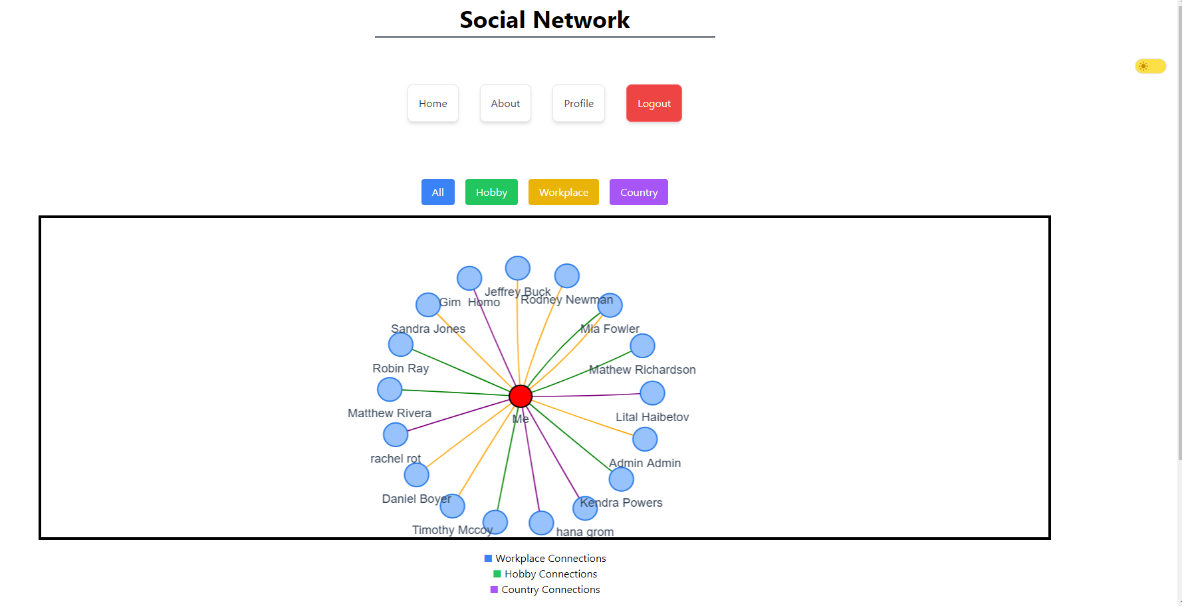
סיסמה : 321

שם משתמש : 888888888

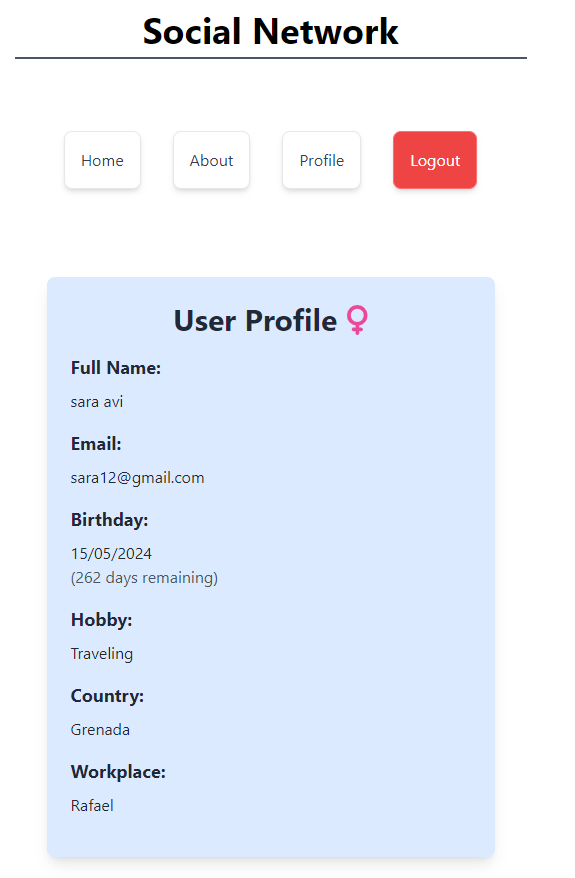
סיסמה : 888

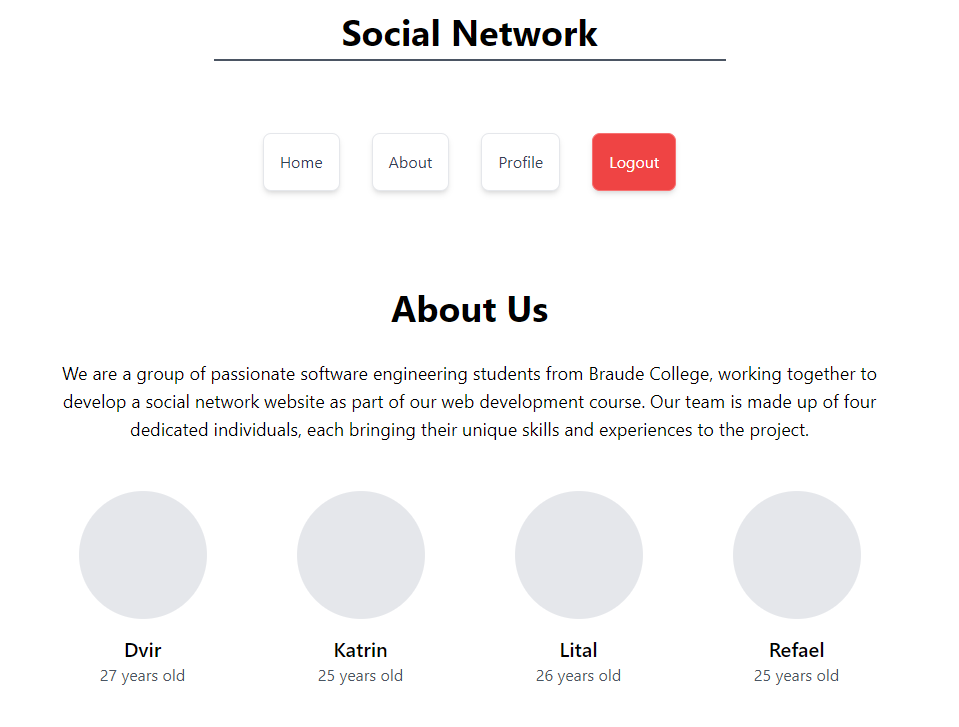
1. משתמשים לא רשומים נדרשים להירשם לאתר בעזרת כפתור הsign up
2. ההרשמה לאתר כוללת מספר שדות חובה ביניהם : שם ושם משפחה , תעודת זהות שתשמש כשם משתמש (חד חד ערכי) , סיסמה , תחביב/מקום עבודה /מדינה שנבחר ממספר אופציות , תאריך לידה ומין הנרשם.
3. לאחר הכניסה לאתר :

המסך הראשי מכיל את הגרף של הקשרים כאשר קשרים ממקום העבודה מיוצגים בעזרת קשתות צהובות , תחביבים בעזרת קשתות ירוקות ומדינה בעזרת קשתות סגולות.

ניתן לבצע סינון ולראות מי האנשים המחוברים עליי רק לפי תחביב (לדוגמא) ע"י לחיצה על כפתור הhobby

1. A screenshot of a phone

   Description automatically generatedהמסך הראשי גם מכיל את כפתור הפרופיל אשר בלחיצה עליו ניתן לראות את הפרופיל שלי הכולל את הפרטים של המשתמש , והפרופילים של האנשים שמחוברים אליי, ע"י מעבר מעל הבן אדם עם העכבר ניתן לשלוח לו מייל.
2. המסך הראשי כולל גם את כפתור הabout המכיל את המידע על היוצרים של האתר.



1. משתמש מנהל:

**שם משתמש : 111**

**סיסמה :123**

למשתמש מנהל נוסף גם כפתור הadmin כחלק מהתפריט.

A screenshot of a social network

Description automatically generated

1. כפתור האדמין מאפשר למשתמש להוסיף תחביבים/ מקומות עבודה ולמחוק אותם, התחביבים /מקומות העבודה שיתווספו יהיו מוצגים בcombo box בדף ההרשמה.
2. המשתמש מנהל יכול גם למחוק משתמשים מהמסד נתונים.

A white background with black lines

Description automatically generated with medium confidence

1. לבסוף , קיימת אופציית התנתקות מהמערכת

A screenshot of a social network

Description automatically generated

קישורים:

* GIt repository - [Front](https://github.com/RafaelAbb/SocialNetwork) [Back](https://github.com/DvirHayat/WEB-Course-backend)
* [Vercel](https://social-network-bfr9.vercel.app/)
* [MoreThanWallet.com Gallery](https://www.morethanwallet.com/app/887)