**תיאור פרויקט חלק ג'**

**מסמך באנגלית**

**The project is written in TypeScript. I read online how not to start a new project with npm and followed various documents and guides. To run the project, install the mySql server on the computer - a DB server that will run on the computer. Here is the link to install:**

<https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-web-community-8.0.31.0.msi>

**After installing the server, you need to run the following line to create a DB named web and to be able to connect to it, you need to set a username: root and a password: root as per the project requirements.**

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'root';

CREATE DATABASE web

**After we have a DB installed and working on the computer, we will install npm, nodejs to install the project libraries and we can run it:**

<https://nodejs.org/en/download/>

**In order to run the project you need to enter the project folder (through vsCode or open cmd in the project location) Now we will install the libraries from npm using the command: npm install. Now we will run the project with the command: npm start. Now the project is working and it should register blogs that it connected to the DB.**

**In order to use the API of the project, there are a number of addresses that can be accessed, for example if the project runs on port 3000 then each address will start with**: <http://localhost:3000/> ….

**First we will initialize the DB using this GET request - http://localhost:3000/db/init It should be noted that this reference will delete the entire DB and restart it with data according to the files in init-data.**

**There are 3 personalities in DB:**

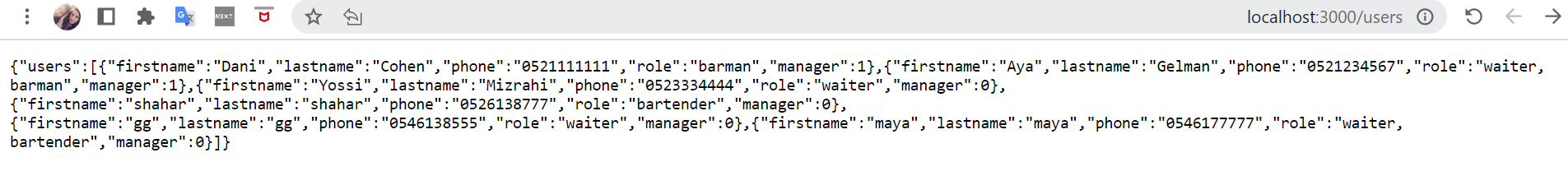
** users**

** Shifts that the employees submitted shifts**

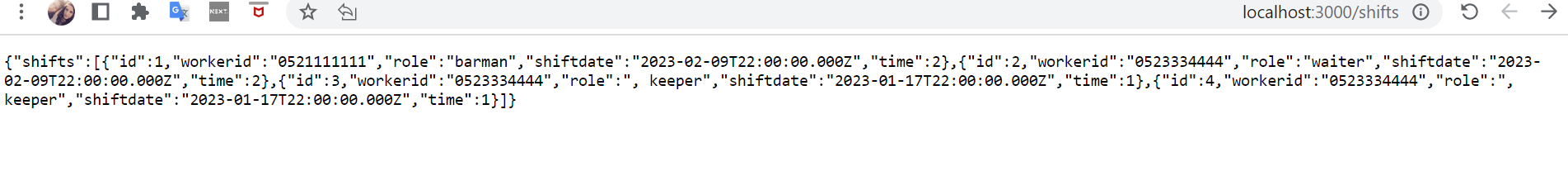
** requirements for shifts from the requirements manager**

**Anyone can be contacted with a GET request in order to get all the information about the person, or to filter according to certain parameters. Now I will detail each of the options and show examples.**

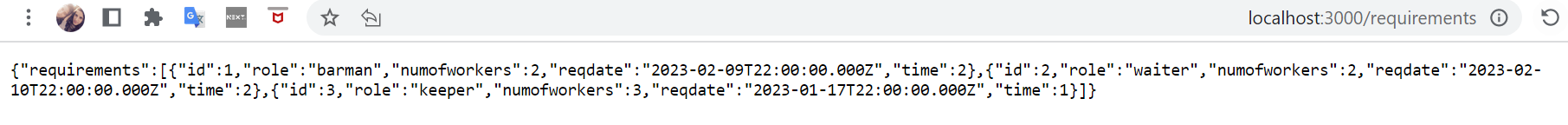
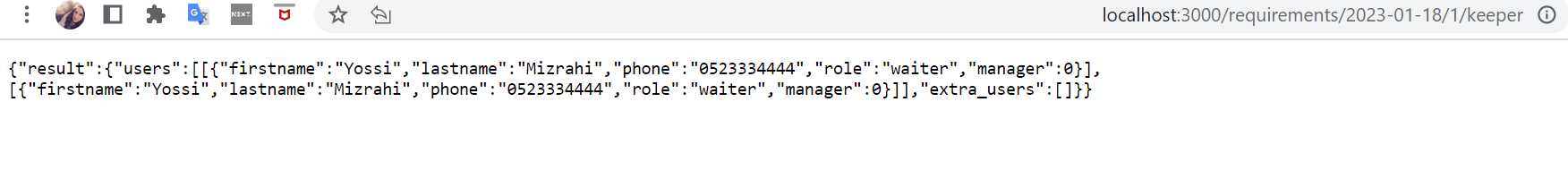
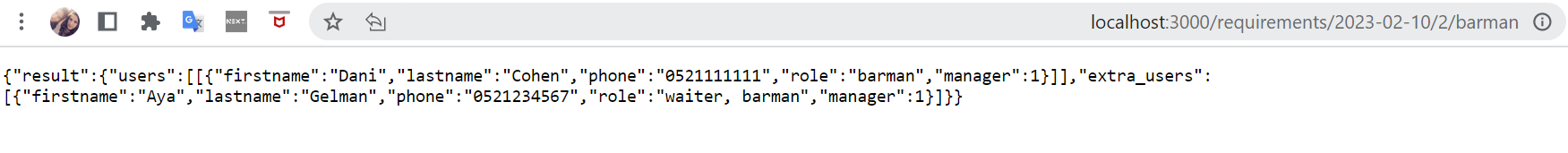
**USERS**

<http://localhost:3000/users> => get all users  
/users/0521234567 => get user by phone number (for example – 0521234567)  
/users/role/barman => get user by role (for example – barman)  
(POST API) /users => add new user

**SHIFTS**

<http://localhost:3000/shifts> => get all shifts  
/shifts/2023-02-17/2/barman => get shifts by date, time, role  
(POST API) /shifts => add new shift

**REQUIREMENTS**

/requirements => get all requirements  
  
/requirements/2023-02-17/2/barman => get users from shift for this requirement and extra users if not enough users for this requirement  
  
(POST API) /requirements => add new requirement

**מסמך בעברית**

הפרוייקט כתוב בTypeScript. קראתי באינטרנט איך לאתחל פרויקט חדש עם npm ועקבתי לפי מסמכים ומדריכים שונים. כדי להפעיל את הפרויקט נדרשות התקנות של mySql server על המחשב – שרת DB שירוץ על המחשב. להלן הקישור להתקנה:

https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-web-community-8.0.31.0.msi

אחרי שמתקינים את השרת צריך להריץ את השורה הבאה כדי ליצור DB בשם web ושיוכלו להתחבר אליו צריך להגדיר שם משתמש: root וסיסמא: root כמו בדרישות הפרויקט.

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'root';

CREATE DATABASE web

אחרי שיש לנו DB מותקן ועובד במחשב, נתקין npm, nodejs כדי להתקין את ספריות הפרויקט ונוכל להפעיל אותו:

https://nodejs.org/en/download/

על מנת להריץ את הפרויקט צריך להיכנס לתיקיית הפרויקט (דרך vsCode או לפתוח cmd במיקום הפרויקט) כעת נתקין את הספריות מnpm באמצעות הפקודה: npm install . כעת נריץ את הפרויקט בפקודה: npm start. כעת הפרויקט עובד והוא צריך לרשום בלוגים שהוא התחבר לDB.

על מנת להשתמש בAPI של הפרויקט, יש מס' כתובות שניתן לגשת אליהן, למשל אם הפרויקט רץ בפורט 3000 אז כל כתובת תתחיל ב: <http://localhost:3000/>......

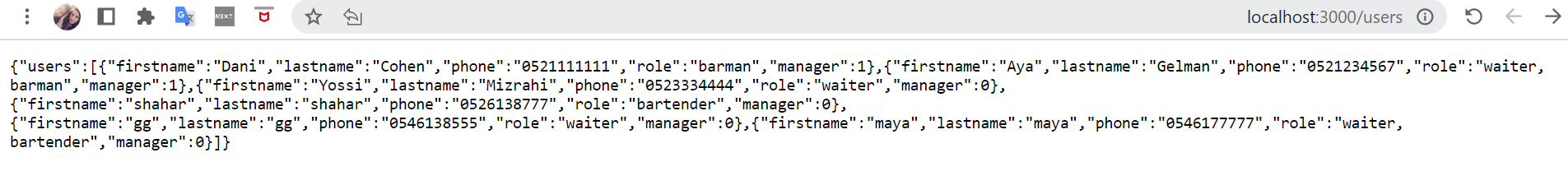
ראשית נאתחל את הDB באמצעות שימוש בבקשת GET הזו – http://localhost:3000/db/init   
יש לשים לב שהפניה הזו תמחק את כל הDB ותאתחל אותו מחדש עם נתונים לפי הקבצים בinit-data.

קיימות 3 אישיות ב DB:

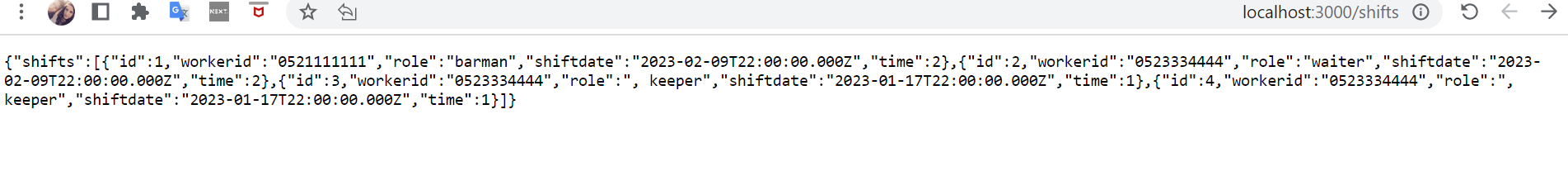
* משתמשים users
* משמרות שהעובדים הגישו shifts
* דרישות למשמרות מהמנהל requirements

לכל אחד ניתן לפנות בבקשה מסוג GET על מנת לקבל את כל המידע על האישות, או לפלטר לפי פרטרים מסויימים. כעת אפרט על כל אחת מהאופציות ואציג דוגמאות.

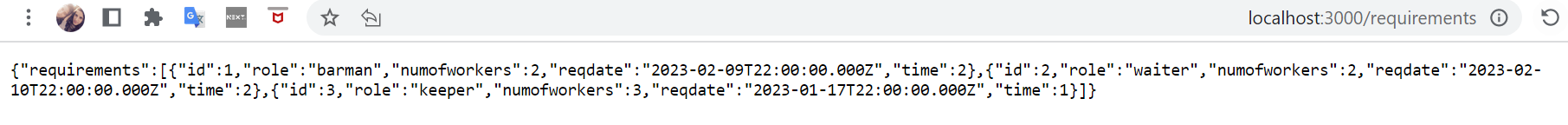
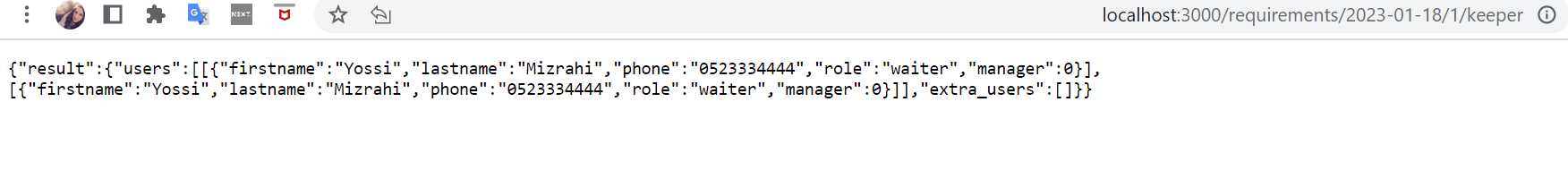
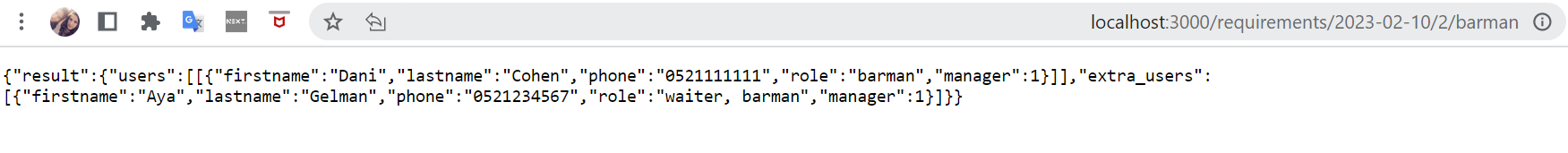
**USERS**

<http://localhost:3000/users> => get all users  
/users/0521234567 => get user by phone number (for example – 0521234567)  
/users/role/barman => get user by role (for example – barman)  
(POST API) /users => add new user

**SHIFTS**

<http://localhost:3000/shifts> => get all shifts  
/shifts/2023-02-17/2/barman => get shifts by date, time, role  
(POST API) /shifts => add new shift

**REQUIREMENTS**

/requirements => get all requirements  
  
/requirements/2023-02-17/2/barman => get users from shift for this requirement and extra users if not enough users for this requirement  
  
(POST API) /requirements => add new requirement