**Angular Home Test**

Build a clinic appointment system according to the following structure:

* **login page**
* **Appointment page** - **top**: menu, logo and user details.
* **Appointment page** - **Right menu**: Select the data to make an appointment
* **Appointment page - Left side Appointment list:** will contain the queue list**.**

Details of the structure:

1. login page:
   1. Username form
   2. Appointment Search Button
   3. Clicking on Search Appointment will take the username to the "Appointment Page" where the Appointments of that user and his personal details will be displayed.
2. Appointment page - top:
   1. Patient details, aligned to the right
   2. The date and time of entry, aligned to the left.
3. Appointment page - Right menu:
   1. Form that will contain: Date and time of the appointment, name of the attending doctor (selection from the list of doctors), additional information.
   2. save appointment button, that adds the appointment to the list and then clears the form.
   3. clears the form button.
4. Appointment page - Left side Appointment list
   1. List of appointments for the same patient. Identification will be done by username after moving from the login page to the appointments page. Bonus: The appointment list will appear in chronological order of date (the queue closest to the current date will appear first).
   2. Each Appointment contains:
      1. All details of the appointment - name of the doctor, date and time, and additional information.
      2. Delete button - clicking on it will delete the appointment
      3. *Bonus - edit button - clicking on it will load the details in the menu on the right and the "create appointment" button will change to "edit appointment" (of course after clicking the details will be updated in the specific appointment).*

additional instructions

* Note: Part of the application infrastructure is already built (the router, the basic components, and part of the design), please use this infrastructure to implement the system according to the instructions above.
* Think of building the system as part of something bigger, design the entities that will fit modularly to other pages that may be attached in the future.

Instructions for installing the startup project and running a server side:

* <https://github.com/bmcsites/EH-TEST>
* Make sure Compatibility with the working environmen, node js and Angular CLI corresponding to Angular 11 must be verified.
* Go to the project directory and after installing the node models run in a separate window: npm run fake: server
* Make sure that the project should be run without errors.
* Inside the project there is an Elad-Fake-Server.postman\_collection.json file, this file is a postman file that can be imported and get all the API lists related to the project.
* When finished, zip the project library (without node modules) and resubmit the test. You have 24 hours to finish your homework. Good luck !!!.