

IsraelHayom Coding task

General Description

This project will be a combination of two sub-projects -

- Frontend- To be built using the NextJS framework, written in TypeScript, and using React. Use TailWind or Sass for styling.
- BackEnd- To be built using NestJS written in Typescript, and the data will be managed in MongoDB.

The frontend and backend should communicate using Get/Post HTTP requests. The project should be built with docker-compose and it should be easily deployed on any machine.

Frontend Task

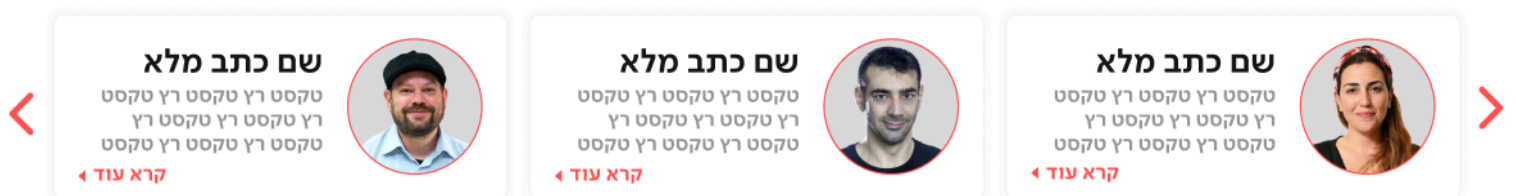
Build a frontend React component called 'writers-carousel', the component consists of a title, and a carousel of 'cards' showing a writer image, the name of the writer, and the title of the writer's latest articles. The component should be responsive and change based on the window width (i.e. desktop or mobile). Use SwiperJS for the carousel.

The data displayed in the component should be fetched from the backend.

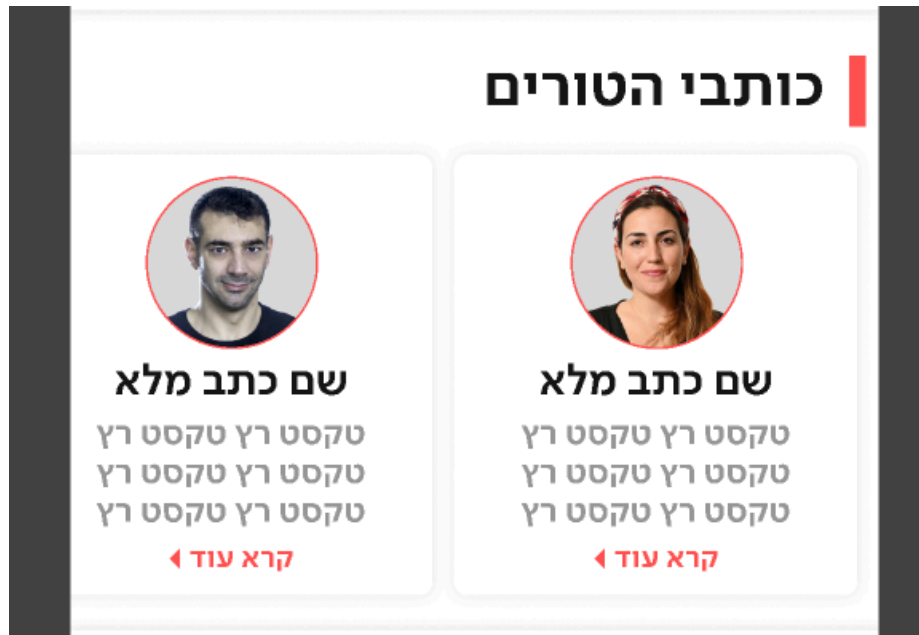
The design of the component should adhere as much as possible to the following two images:

- Desktop (width>600px)

כותבי הטורים



- Mobile (width <= 600px)



Design requirements:

- Font: Ariel
- Font size:
 - Big title: 28px
 - Writer name: 22px
 - post title: 16px
 - 'Read more': 14px
- Colors:
 - Reds: #FF5252
 - Black text: #181818
 - Gray text : #999999
- Carousel will scroll right to left
- Clicking on the writer name or 'read more' should redirect to a writer page
- Clicking on the article title should redirect to the article page
- The amount of cards displayed can vary depending on page width, it is fine if the last card is 'cut'
- Don't forget the red border around the image and box shadow around the cards

*The title and amount of writers displayed can be hard-coded in the frontend.

Backend Task

Build a basic NestJS based 'microservice', the service will handle the writers' data.

In MongoDB create two collections- 'posts', and 'writers'.

The 'writers' collection will contain -

- A unique number id
- Writer name
- Writer image url (for this task can be any image)
- Writer page url (for this task can be any example.com url)

The 'posts' collection should contain -

- A unique number id
- A post title
- A creation date-time
- A post url (for this task can be any example.com url)
- The ID of the writer that wrote that post

No need for more than 10 writers. To show the service is working correctly please create 3 posts per writer, each with a different creation time.

*You can use a script or ChatGPT to generate and import the data into the DB, you don't need to write it manually.

The backend service should expose an endpoint that is accessible by the frontend. Upon a GET request, the backend will pull a list of writers and posts from the DB, sort it, and serve it to the frontend (to be used by the component).

Each writer should be returned with their latest post, the writers should be ordered by who wrote a post most recently.