PORTFOLIO

Eyal Lila - Web Developer

NutriScan

SUMMARY

NutriScan is an innovative full stack app, written in TypeScript and JavaScript. It is designed to help you make informed purchasing decisions based on your nutritional, dietary and environmental preferences. It does that by simply scanning products' barcode. It is easy to use, and is accessible as a web version and as mobile app. It is powered by cutting-edge technologies such as React, Node.js, and MongoDB, and is integrated with several features to make the user experience seamless.

### HOW TO RUN THE APP - web version and mobile app

- To access the web version, simply visit the following link: https://nutri-scan.netlify.app/
- To install NutriScan as mobile app, follow these steps:
    1. Visit the web version of NutriScan at the following link: https://nutri-scan.netlify.app/
    2. Click on the computer logo on the right side of the URL box:
    3. Click "Install".
- The server of the app is deployed to Render, a free website deployment service. Therefore, it needs a few minutes to "wake up" and respond to the first API call, for example when registering to the app.

FEATURES

- Scan or type products' barcode and see if they match your diet, environment, and nutrition preferences.
- View detailed information about each product, including ingredients and nutritional values.
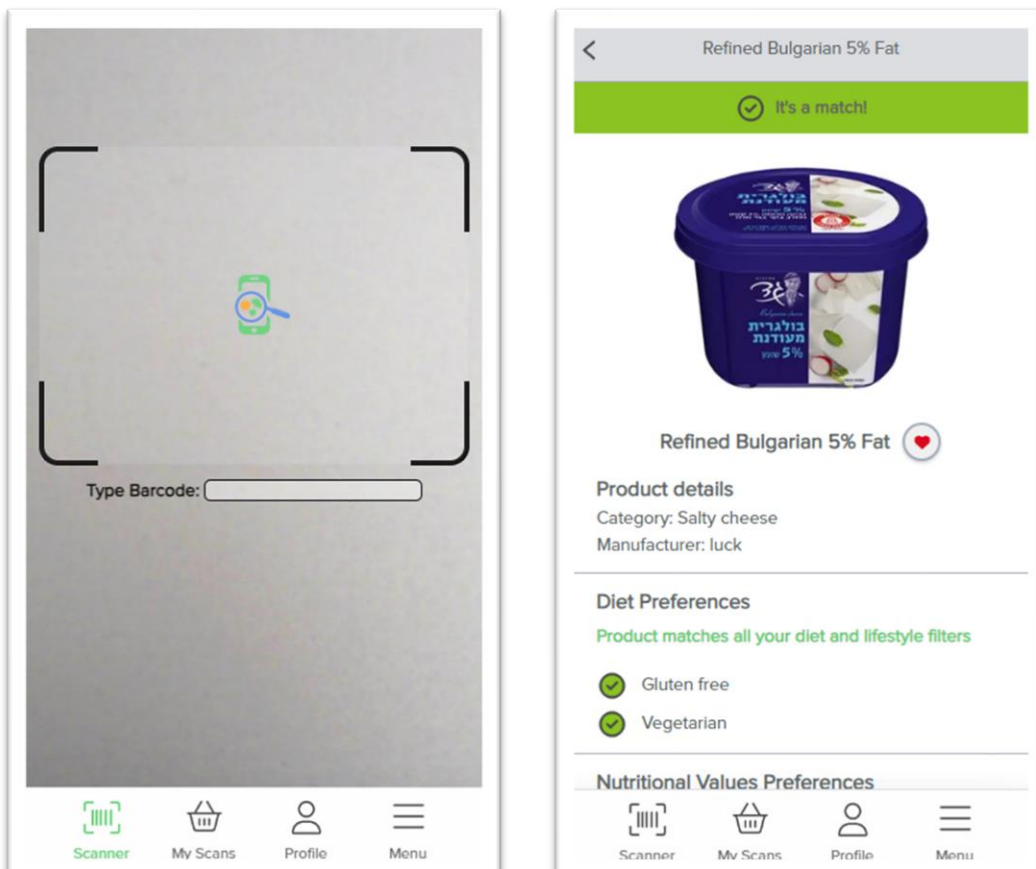
- Set your diet, environment, and nutrition preferences in the app.
- Search products from the products you scanned by name, barcode number or manufacturer.
- Use the liked products filter to show only products you liked.
- Create a user to have all your scans saved, or use without logging in and directly start scanning.
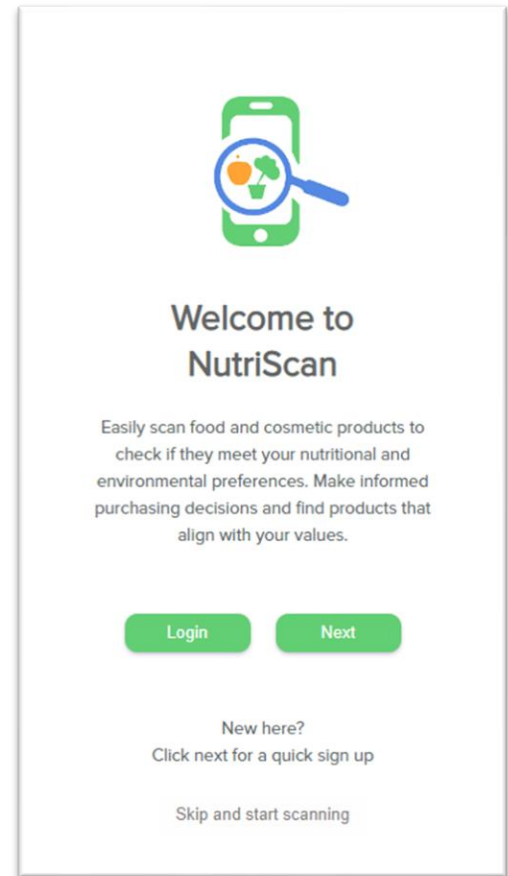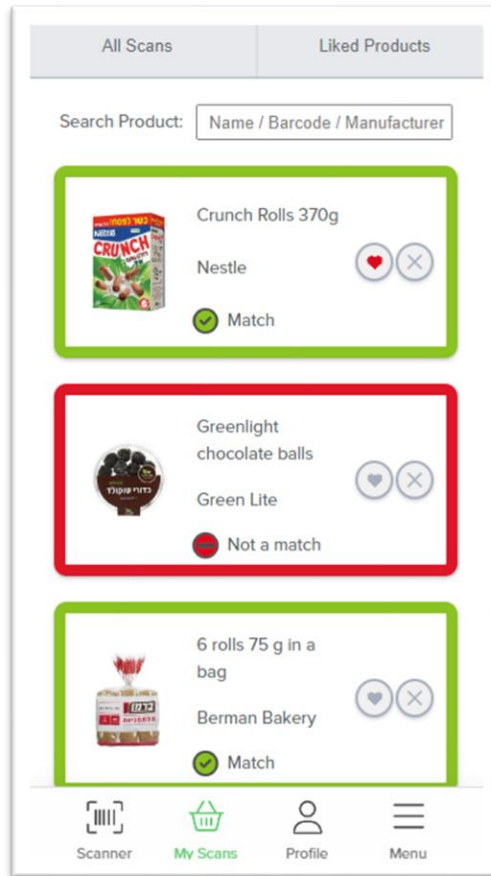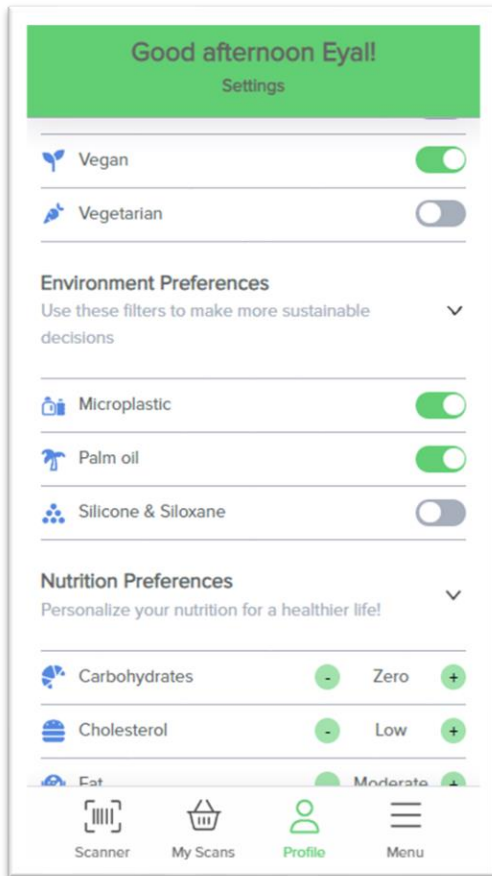
- Authentication with **JSON Web Tokens** (JWT), and use of **cookies**,
- Dynamically created **MongoDB** database, with products scraped from multiple supermarket chains' websites using **Puppeteer**.
- **Microsoft Azure API** integration for translation of products' information from Hebrew to English.
- Quagga barcode scanner integration.
- **Progressive web app**: use the app as a web version, or install it with just a click to use as a mobile app from every device.
- **Responsive** UI.

DEPLOYMENT SERVICES

- Server deployed to Render
- Client (UI) deployed to Netlify

TECHNOLOGIES

- React
- Node.js
- Express
- MongoDB
- JSON Web Tokens (JWT)
- Puppeteer (scraping)
- Microsoft Azure API (translation)
- Quagga barcode scanner
- Redux

PLANNED FEATURES

The app is continually improving, and already has several planned features in the pipeline. The following features are planned for the near future of NutriScan:

- Search from products, instead of scanning.
- In the product's page, a carousel of similar products.

CHALLENGES

The development of NutriScan presented several challenges, including:

- Creating the functions that determines if the product corresponds with the user's settings was the apps main challenge and the core of its logic.
- Scraping: writing a scraping function that scrolls the web page to load all the products, opens each product page and the sections within it. The function mimics user's behavior and has a few measures to handle fails while loading the information on the website.
- Dynamically creating the database, embedding translation and many functions that test the products' data, to be later tested against the user's preferences in the app.

Pokémon Role-Playing-Game

SUMMARY

This is a fun, React based RPG (role-playing game), written in JavaScript. It is inspired from the iconic TV series from the 90's, that millions around the world grew up with. This game allows you to tour through the Pokémon world, explore a variety of maps and fight many different Pokémons. And of course, to try and catch 'em all!

The game allows users to play with the Pokémons they catch, and fight dynamically generated opponents that are adapted to the users' level. To add a bit of a challenge, the game's difficulty increases as the game progresses.

The game is powered by React and Firebase Realtime Database. It is integrated with many features that make the user experience fun and smooth.
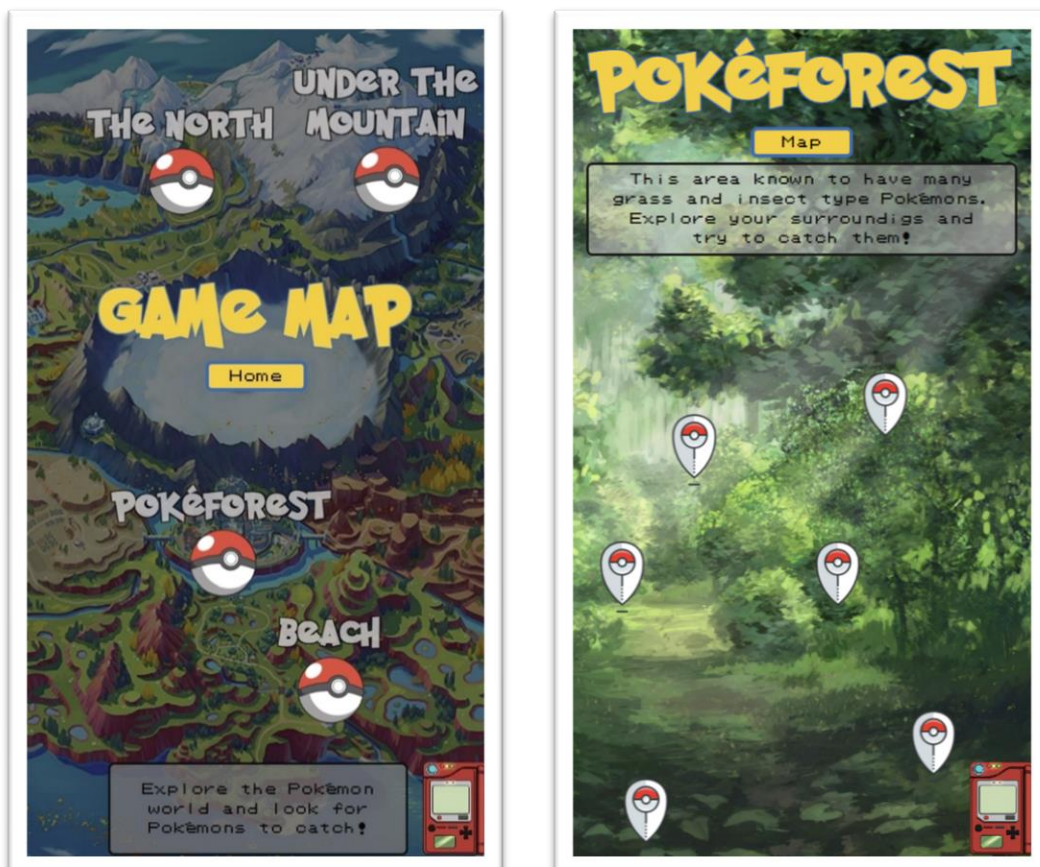
HOW TO RUN THE GAME

Simply visit the following link from a computer or any mobile device:

https://pokemon-rpg.netlify.app

FEATURES

- An interactive Pokémon world, consisting of several maps and many different opponents to fight.
- Dynamically updated maps - after a user caught a Pokémon, it will not ap pear again.

- Role playing battle with multiple action possibilities and unique animations.
- Dynamically generated opponents - adjusted to the users' level.
- Responsive UI, adapted to use from any device.
- Announcer: a message typing component that accompanies the user throughout the game, for a seamless user experience and a retro 90's design.
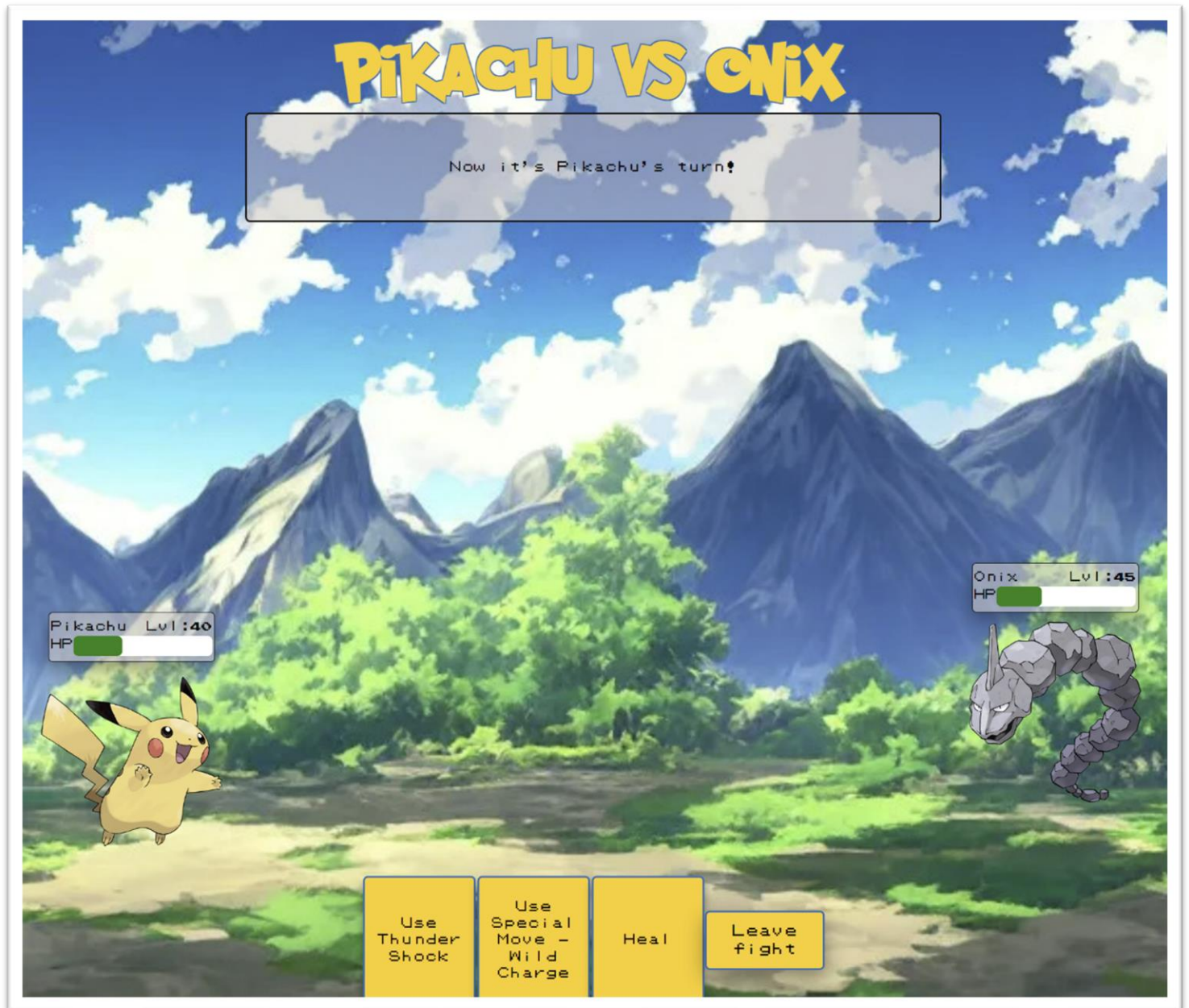
- Create a user or login to automatically save your progress.
- Play without login, for a direct and easy access to the game.
- Pokédex showing the user's Pokémons, which is synced with the user's data on Firebase Realtime Database.
- Play with the Pokémons you catch. The user is being updated in the Database and in the front-end as the game progresses, with zero loading time.

DEPLOYMENT SERVICES

- Deployed to Netlify

TECHNOLOGIES

- React
- Firebase Realtime Database

PLANNED FEATURES

The following features are planned for the near future of the game:

- Connect the game to PokéAPI, to randomly generate opponents from hundreds of Pokémons, so users can always fight different opponents (currently there are 27 different Pokémons in the game).
- Create a MongoDB database to support the game backend.
- User authentication with JWT.
- Adding sound to the game.

CHALLENGES

The development of the game presented several challenges, including:

- The battle logic: The battle is the core of the game, and it was important for me that it will have a seamless flow and readable code. For this I developed a custom hook, which controls the turns, the health of the players and the animations that accompany each stage of the battle sequence.
- Dynamically creating opponents that are adapted to the users' level.