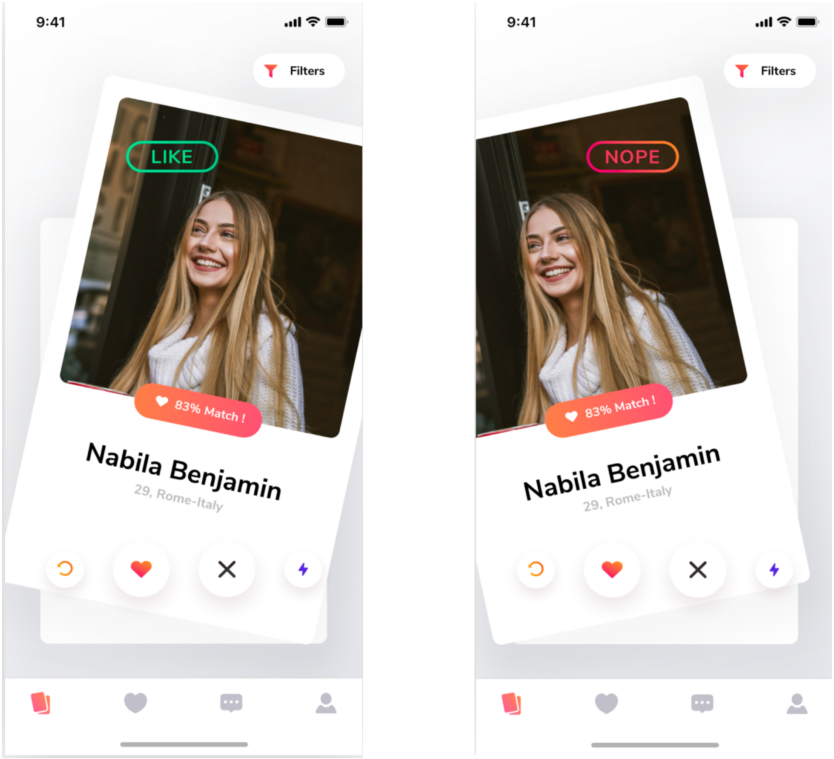
Jean-Loïc De Jaeger
jl.dejaeger@gmail.com

Tinder clone
Projet Service Web

Année 2024

Objective

Develop a Tinder clone with a frontend built in React and a profile provider web service using Flask.



Requirements

Frontend (React)

— Create a React application implementing the Tinder swiping screen
— Display user profiles with the ability to swipe left (dislike) or right (like).
— Choose existing React components to implement the UI faster
— Use CSS for styling the app

Backend (Flask)

— Develop a Flask web service to manage user profiles.
— Implement an API endpoint to retrieve n number of profile by API call.
— No algorithm needs to be implemented. You can use any public API to retrieve users with their photos and profile information
— No database is required for this project
— No authentication needed for this web service

Tips

The API service can use the randomuser API to generate user data.
API Url : https ://randomuser.me/api

The profile descriptions are a bit more tricky to generate. It can be obtained by doing several templates and replace some words dynamically such as the name, age, city, etc...

Bonus

— Implement the match service. Another React screen need to be implemented. The Flask endpoint should add a like_user_profile (true or false) in the response object (JSON)
— Implement other screens with router for the navigation : https ://reactnavigation.org/

Guidelines

1. Set up the React project using Create React App.
2. Create the Flask backend with the necessary endpoints.
3. Connect the React frontend to the Flask backend using Axios or Fetch API.
4. Test the application thoroughly to ensure all functionalities work as expected.

Submission

Submit the project as a GitHub repository link, including a README file with instructions on how to run the application locally.

Have fun and happy coding !