[Poster Describe Research or Software Development Project]

iCare

21

Dor Azaria (206333452) | Dolev Abuhazira (205776321) | Egor Kononov (342426608)

Instructed by Alon Fliess

Contribution/Project Goal

Make iCare the destination to connect and discover babysitters and nannies and also to offer new opportunities for babysitters around the country.

Introduction

iCare is an application, a community and a place to find babysitters according to parents' priorities. Parents can avoid the stress of finding a reliable Sitter and the hassle of last-minute runs. Nannies can post a job for free detailing childcare, school help, or even tutoring needs. Filter search results based on criteria like experience and skills and many more!

Methods/algorithms/Alternatives or Design Considerations

In order to develop this web application, there were two major possible approaches: **multiple-page application** or **single-page application**. Multiple-page applications are the traditional way of developing web applications and, for some time, were the only way to develop for the web. Single-page applications are a newer way of development that offloads all of the rendering work (front-end) to the browser while the server (back-end) simply manages connections and requests to the database.

Selected Approach

Single-page application was the selected approach, because it fully separates the front-end from the back-end, enabling the development to be done by two teams, working apart from each other in parallel. It also makes it easier to debug the application (something that is significantly harder in the traditional approach of multiple-page application).

Solution Description (Algorithms, Modulation, Patterns, Infrastructure, UI, Functionality)

This web application uses the client-server pattern, where the browser is the client using fetch requests to the server in order to retrieve data or post data.
The server is simply managing the underlying database according to the accesses done by the clients. The UI is rendered fully by the front-end using "*react*" and "react-router-dom" libraries. The back-end is done by *Django* models projected onto a simple and portable *SQLite3* database.

Visit Us

Scan QR-Code for full Instructions (GitHub) ⟶