



All About Me

A brief introduction about my personality and skills

By Marie Behzadi mariebehzadi@gmail.com

Who am I?

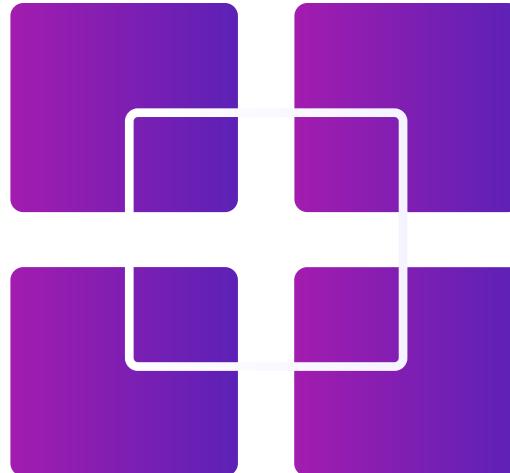
Hello! My name is Maryam Behzadi but you can call me Marie. It's an honor to have this meeting.

-  I'm a **GNU/Linux user** and I've been practically living in the heaven of programmers for almost 8 years.
-  I have more than **10 years of Experience** in Software Engineering.
-  I'm a **Full-Stack developer** and have a lot of experience developing desktop/web applications.
-  I have a **BSc degree in Software Engineering** and a **MSc degree in Artificial Intelligence**
-  I believe in **Teamwork** because It allows to use the strengths of others in addition to my own strengths.
-  I am very **passionate** about what I do and I really care about **self-development** and **self-improvement**
-  I really like to **Learn new things** and expand my knowledge through every project.
-  I'm a very **Organized Person** and this personality aspect has direct effect on my **Structured Coding** process.
-  I started to learn programming in high school and I was one of **The Computer Olympiad** winners in Fars.
-  Outside of work I keep myself **Fit and Active** and I visit the gym a few times a week.
-  I live in **Shiraz, Fars, Iran** and I'm happily married to another developer.

My Skills and Experiences

INTERMEDIATE TO HIGH-LEVEL
RELATED TO THIS SPECIFIC PROJECT

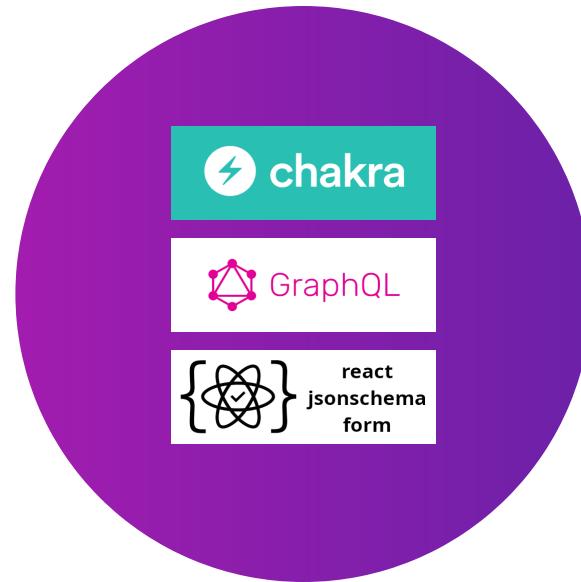
- **Git**
- **VSCode**
- **React.js**
- **RESTful API**
- **Docker**
- **CI/CD**
- **JSON Schema**
- **TypeScript**
- **MongoDB**
- **SQLite**
- **Bootstrap** and **Material UI**



What will I learn?

THROUGH THIS PROJECT

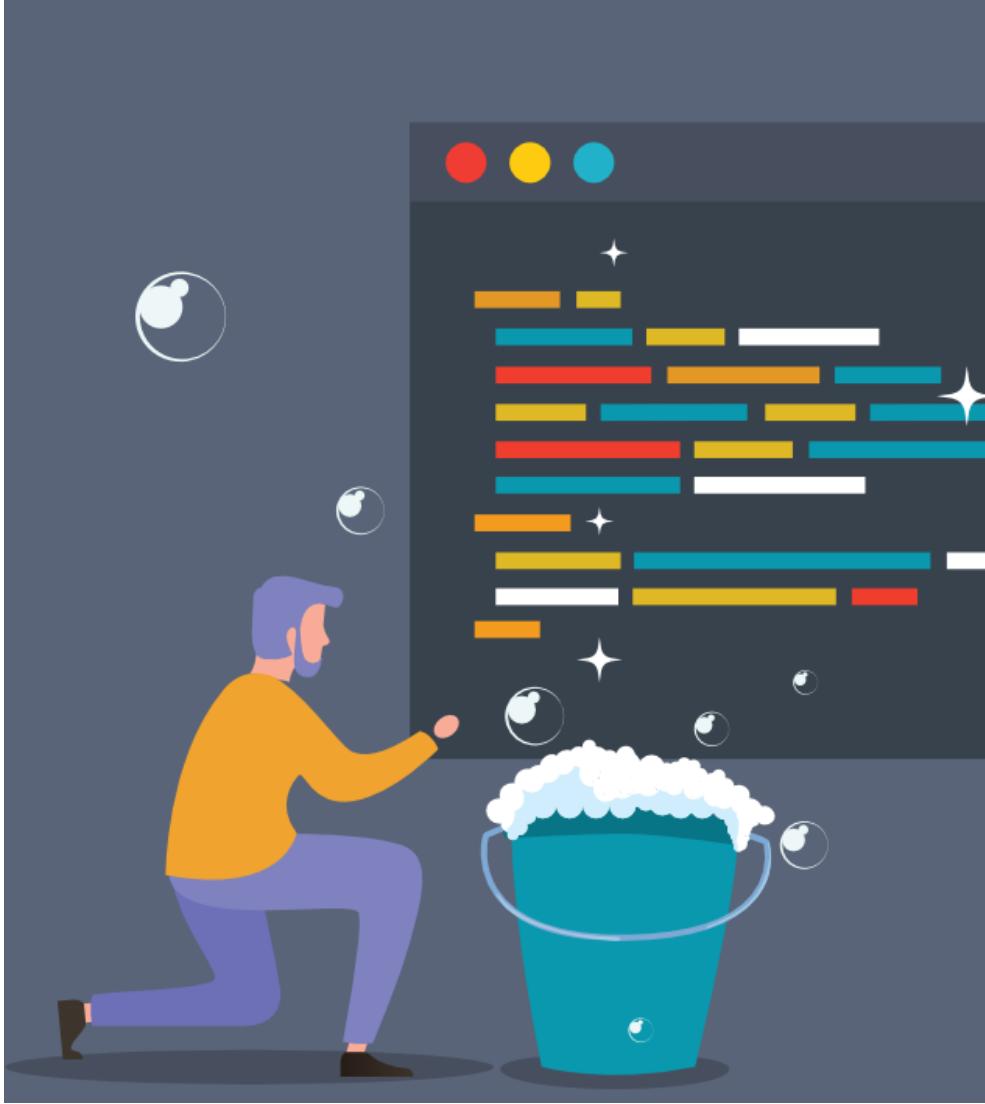
- **Chakra UI** - No previous experience
- **GraphQL** - basic knowledge
- **RJSF** - similar experience in Vue.js



Clean Code Techniques

As I mentioned before, I try to be very organized and clean when it comes to programming. It is absolutely necessary to define guidelines and code rules between multiple programmers in a team and it has a lot of pros to follow the same structure:

- Easier to Understand, Debug and Maintain
- Easier to Test
- Cost Savings
- Reusability
- Simplicity in Large Scale projects
- Avoid continuous Conflicts in Code
- Avoid continuous Conflicts between programmers
- Avoid Refactoring the same code over and over
- Happiness for Everyone!



Clean Code Techniques

1. FOLDER STRUCTURE

Structuring a project's folders in a manner that fits the project's size and intent is key to scalability and maintainability. There also is no single correct answer for what a perfect folder structure looks like. But we have to define a basic structure which seems to be the best solution for our project.

2. NAMING CONVENTIONS

This stage is very critical because we will be defining the best way to describe files, variables, functions and classes. We can even go further by using conventions which indicate variable scopes.

REACT-FOLDER-STRUCTURE

```
> node_modules
> public
< src
  > components
  > constants
  > contexts
  < features
    < authentication
      TS auth-button.tsx
      TS auth-form.tsx
      TS index.tsx
      TS slice.ts
      TS utils.ts
    > hooks
    > layouts
    > pages
    > services
    > styles
    > types
    > utils
    TS store.ts
  & .gitignore
  { package-lock.json
  { package.json
  i README.md
  tsconfig.json
```

Clean Code Techniques

3. CODE FORMATTING

Sometimes software developers write code that is bunched up and smashed together, leaving absolutely no blank spaces. As a result, the code is really hard to read. Solid walls of text aren't very inviting or readable.

When we work on a large-scale project and the number of developers are more than just one person, It's necessary to define formatting rules to avoid constant changes and conflicts in git.

I recommend the following DevTools with the same rule sets to avoid conflicts between them:

- o [ESLint](#)
- o [Prettier](#)

TRY IT ONLINE

INSTALL PRETTIER

What is Prettier?

Why?

- * You press save and code is formatted
- * No need to discuss style in code review
- * Saves you time and energy
- * And more »

An opinionated code formatter
Supports many languages
Integrates with most editors
Has few options

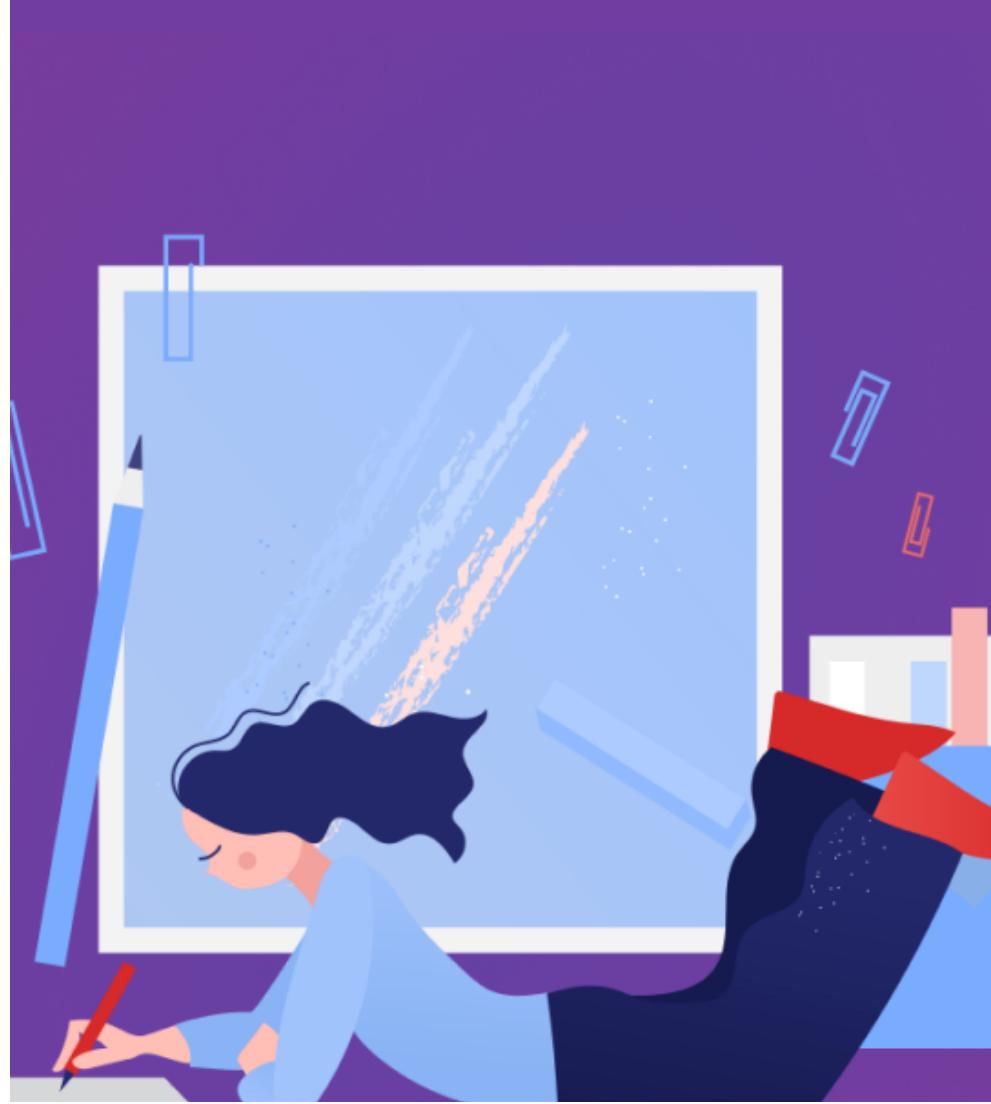
Clean Code Techniques

4. DOCUMENTATIONS

Code Documentation is very important when facing a complicated and large project. I have created two types of documentations for my projects so far: documentation inside the code and supporting documentation about the code.

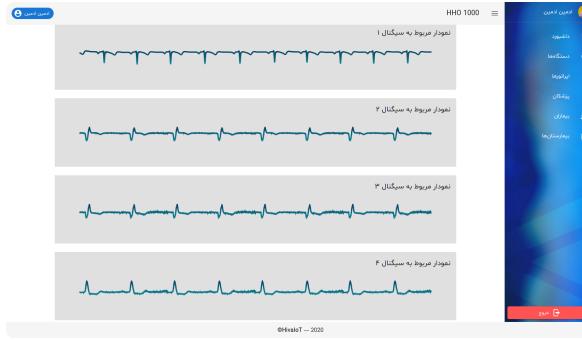
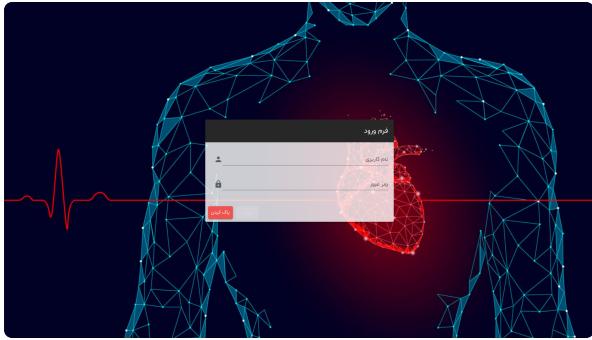
As the projects grows bigger and ages, going back and understanding previous codes can be overwhelming and time-consuming. So it's really necessary to provide additional information on each section of the code to increase and improve maintaining process.

I recommend the following techniques in order to provide code documentations:



Previous Work

1. HEART HOLTER MONITOR PROJECT



کد بیمار	نام و نام خانوادگی	آمار	جنسیت	شماره ثابت	آدرس	نام و نام خانوادگی	آمار	جنسیت	شماره ثابت	آدرس	نام و نام خانوادگی	آمار	جنسیت	شماره ثابت	آدرس
1	علی رضایی	۰۹۸۷۶۵۴۳۲۱	مهمان	۱۳۶۸/۰۸/۱۲	مشاور	علی رضایی	۰۹۱۷۱۱۱۲۳۳	مهمان	۰۹۱۷۱۱۱۲۳۳	مشاور	علی رضایی	۰۹۸۷۶۵۴۳۲۱	مهمان	۱۳۶۸/۰۸/۱۲	مشاور
۱	علی رضایی	۰۹۸۷۶۵۴۳۲۱	مهمان	۱۳۶۸/۰۸/۱۲	مشاور	علی رضایی	۰۹۱۷۱۱۱۲۳۳	مهمان	۰۹۱۷۱۱۱۲۳۳	مشاور	علی رضایی	۰۹۸۷۶۵۴۳۲۱	مهمان	۱۳۶۸/۰۸/۱۲	مشاور
۱	علی رضایی	۰۹۸۷۶۵۴۳۲۱	مهمان	۱۳۶۸/۰۸/۱۲	مشاور	علی رضایی	۰۹۱۷۱۱۱۲۳۳	مهمان	۰۹۱۷۱۱۱۲۳۳	مشاور	علی رضایی	۰۹۸۷۶۵۴۳۲۱	مهمان	۱۳۶۸/۰۸/۱۲	مشاور
۱	علی رضایی	۰۹۸۷۶۵۴۳۲۱	مهمان	۱۳۶۸/۰۸/۱۲	مشاور	علی رضایی	۰۹۱۷۱۱۱۲۳۳	مهمان	۰۹۱۷۱۱۱۲۳۳	مشاور	علی رضایی	۰۹۸۷۶۵۴۳۲۱	مهمان	۱۳۶۸/۰۸/۱۲	مشاور

Technical Details

React.js + Golang + RESTful API + PostgreSQL

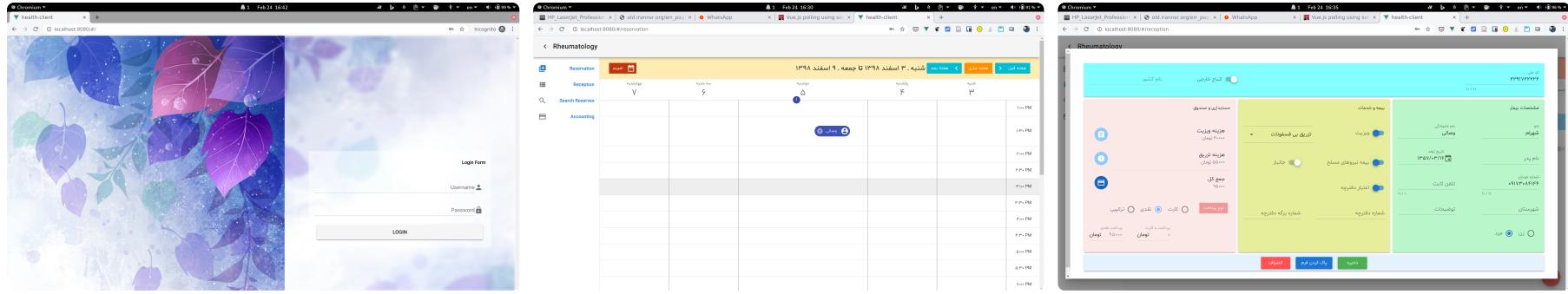
4 types of users and permissions + different dashboards

Generate Forms based on JSON Schemas using our custome components

Desktop / Web Application

Previous Work

2. RECEPTION APPLICATION



Technical Details

React.js + Golang + RESTful API + PostgreSQL

3 types of users and permissions + different dashboards

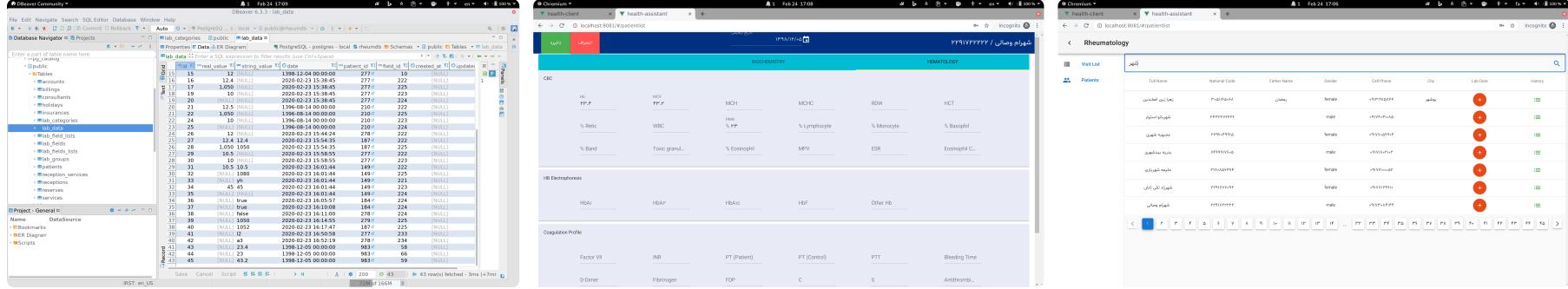
Design Custom Calendar filtered by day, week, month and year

Full Accounting System

Desktop / Web Application

Previous Work

3. RHEUMATOLOGY APPLICATION



Technical Details

React.js + Golang + RESTful API + PostgreSQL

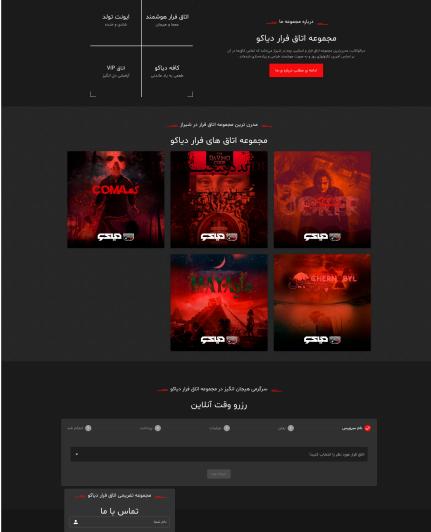
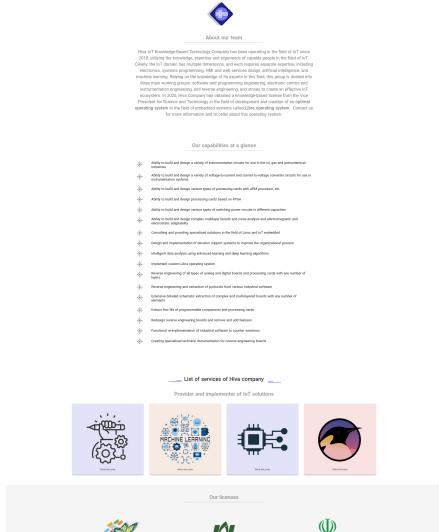
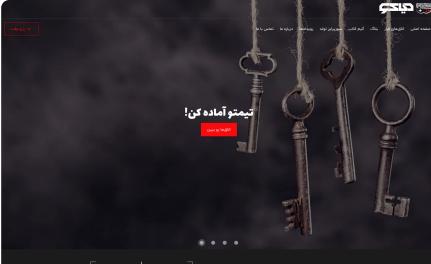
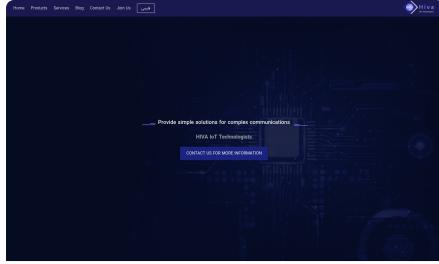
3 types of users and permissions + different dashboards

Generate dynamic Forms based on user's edit on form

Desktop / Web Application

Previous Work

4. SOME OF VUE.JS BASED PROJECTS



Bootstrap vs Material UI vs Chakra UI



MUI Core

About

MUI Core is a collection of React UI libraries for shipping new features faster. Start with Material UI, our fully-loaded component library, or bring your own design system to our production-ready components.

mui.com/

react, javascript, design-systems, typescript, material-design, react-components, hacktoberfest

Readme, MIT license, Code of conduct, 80k stars, 1.4k watching, 27.5k forks

Releases 365

v5.9.1 (Latest) 6 days ago

+ 364 releases



chakra

About

⚡ Simple, Modular & Accessible UI Components for your React Applications

chakra-ui.com

react, reactjs, uikit, a11y, react-components, accessible, wai-aria, ui-components, dark-mode, ui-library, chakra-ui

Readme, MIT license, Code of conduct, 27.8k stars, 198 watching, 2.4k forks

Releases 2,258

storybook-addon@2.0.10 (Latest) 11 days ago

+ 2,257 releases



bootstrap

About

Bootstrap components built with React

react-bootstrap.github.io/

react, javascript, bootstrap, typescript, react-components, hacktoberfest

Readme, MIT license, 20.9k stars, 432 watching, 3.4k forks

Releases 43

v2.4.0 (Latest) on May 13

+ 42 releases

Learn More about React.js

[official documentations](#) · [official tutorials](#) · [Awesome React](#)

[youtube](#) · [codewithmosh](#) · [GraphQL Intro](#) · [HowToGraphQL](#)

[RealReactApps](#) · [Important Talks](#) · [React Conferences](#)

Thank You

Marie Behzadi