



# Takeaway - Order list App

Firstly, I added **vue-resource** plugin to my project making web requests and handle responses, explicitly to send a GET request to the APIs that return JSON metadata to get the users and todos data.

Then, **Bootstrap** library was added so that I used the Bootstrap CSS classes for styling the user interface of my application. One of the reasons is that Bootstrap makes the entire process of creating responsive designs easy.

My UI design like below:



Users

Leanne Graham

Ervin Howell

Clementine Bauch

Patricia Lebsack

Chelsey Dietrich

Mrs. Dennis Schulist

Kurtis Weissnat

Nicholas Runoldsdottir V

Glenna Reichert

Clementina DuBuque

Leanne Graham User Details

Leanne Graham Todos

Name  
Leanne Graham

Email  
Sincere@april.biz

Address  
Kulas Light  
Apt. 556  
City: Gwenborough  
Zipcode: 92998-3874

Phone  
1-770-736-8031 x56442

Website  
hildegard.org

Company Name  
Romaguera-Crona

Company Tags  
Multi-layered client-server neural-net

Company Business  
harness real-time e-markets

Leanne Graham Todos Details

delectus aut autem

quis ut nam facilis et officia qui

fugiat veniam minus

et porro tempora

laboriosam mollitia et enim quasi adipisci quia provident illum

qui ullam ratione quibusdam voluptatem quia omnis

illo expedita consequatur quia in

quo adipisci enim quam ut ab



molestiae perspiciatis ipsa

illo est ratione doloremque quia maiores aut

vero rerum temporibus dolor

ipsa repellendus fugit nisi

et doloremque nulla



Select header background color

☐ Red

☐ Blue

☐ Orange

☐ Black

Select font family

☐ sans-serif

☐ Times

☐ monospace

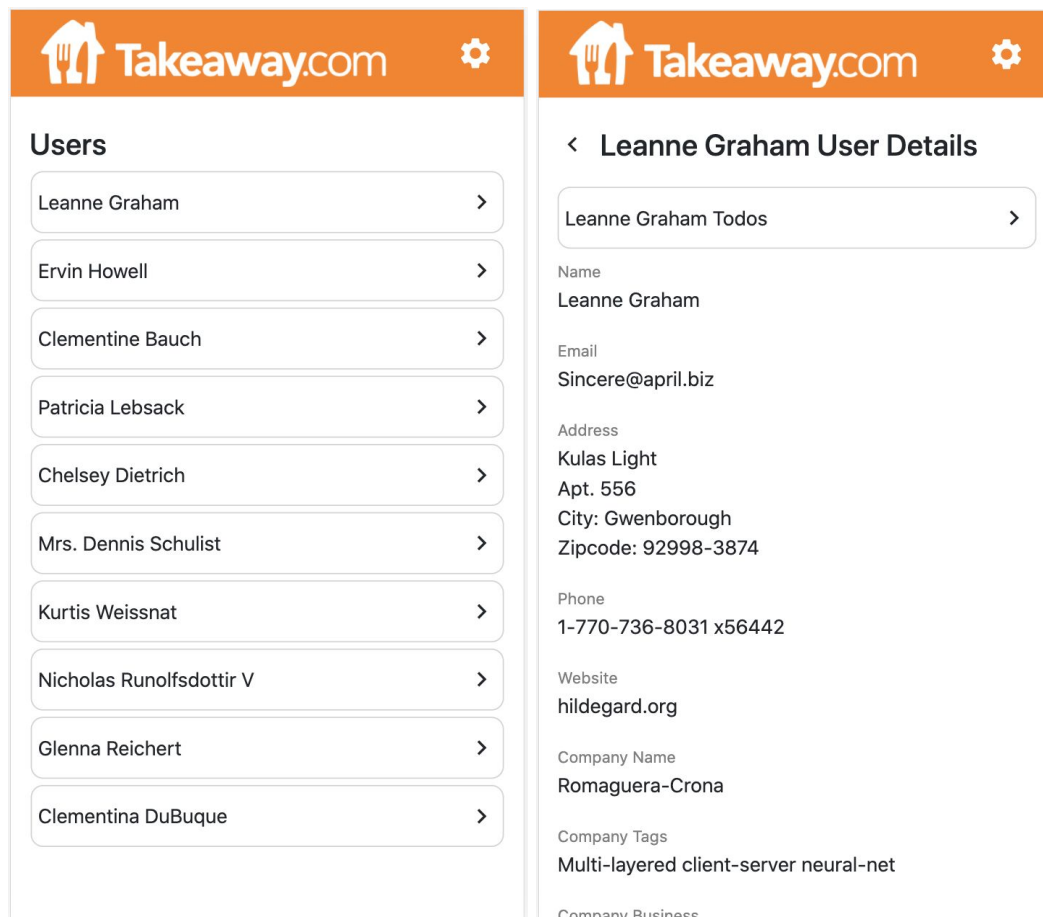
☐ Courier

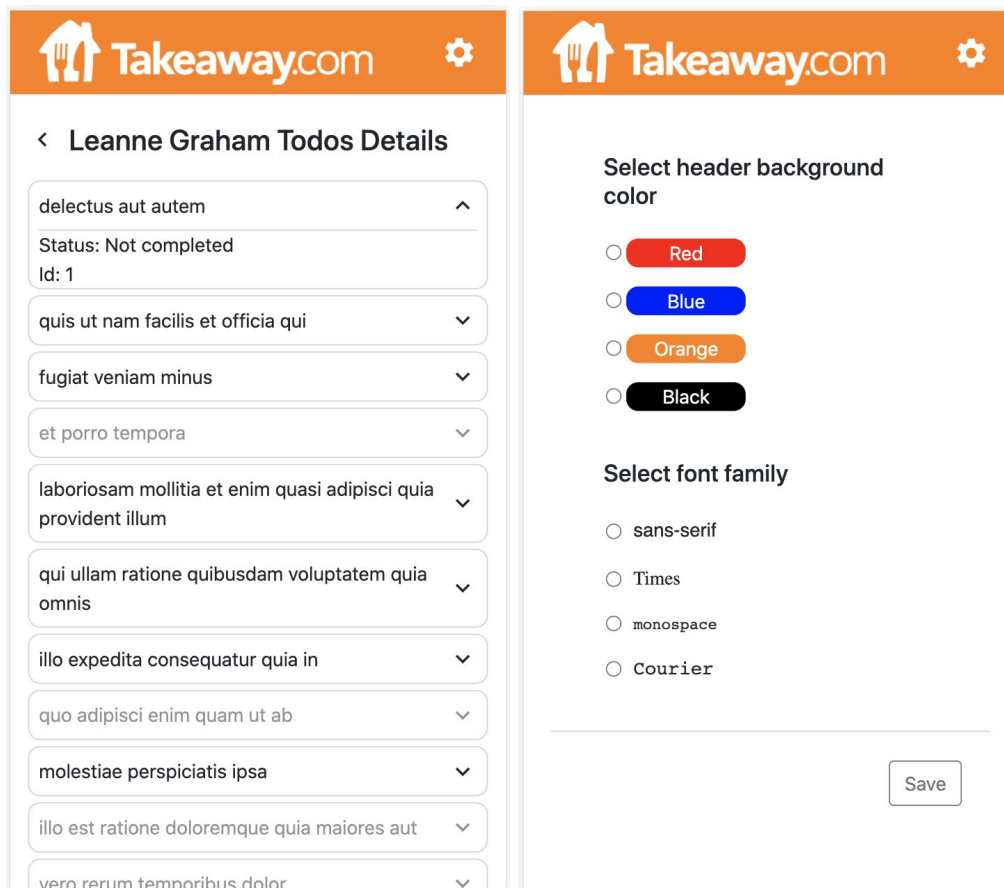
Save

After that, Components are implemented according to the UI design. To achieve the UI design, I created 4 components:

- **UsersList:** Firstly, the component gets the users list from the store and displays it. If the user clicks on any user, that user's user details is been displaying in the next column. To show that user details, it calls the UserDetails component. Moreover, if the user clicks on the user's todos, it calls the TodoDetails component to display the todos list.
- **UserDetails:** When this component created, it also gets the selected user's todos from the store. Then it displays the selected user's user details as well as the todos button. Lastly, it emits the todos data to the UsersList component.
- **TodoDetails:** This component gets the todos list as a prop from the UserList component. When the user clicks on a todo, it displays the details of the selected todo.
- **AppSettings:** In this component, the user can change the color of the header and the font family of the entire app.

Also, the mobile UI design of my app like below:





**i18n** package is for adapting the app or products to specific local languages. For now, I added only the English option, other languages can be added easily in the future. The main reason to add this package is that It makes the maintenance of the app easier and simpler.

Also, I added **Vuex(store)** for the state management. So, Vuex is a state management library, and it helps us deal with shared states. In my application, I put the API call method in the actions and I can dispatch different parameters from different components. After dispatching different parameters to change the URL and data, it will mutate the state values. Lastly, we will get these state values to use in the components.

For the testing, I used **Nightwatch.js** framework for end-2-end(functional) testing. I wrote E2E tests because they assure me that I have a working system. And also, It Can implicitly test many things at once. There are 3 test suites to check:

- Header is visible
- User-list, user details and todos details are visible
- Setting changes work

To run tests first go to project directory and write the below command to the terminal  
`npx vue-cli-service test:e2e --url http://localhost:8080/`