## Implemented functionalities

#### Header

1. Has to contain a title and a logo.
2. Has to be visible on all pages.
3. Has to be stuck on the top.

#### Footer

1. Has to be visible on all pages.
2. Has to be stuck to the bottom.

Menu

1. The selected menu has to be marked.

#### User From page

The purpose of this page is to collect and save user data.

1. Fields:
   * First name: required
   * Last name: required
   * email address: required, has to contain name part, @, domain, dot, and top-level domain (at least two characters)
   * Phone number: required, has to start with ‘+’ character
   * Birthday: Accepted format mm/dd/yyyy with leading zeros
   * About: no restrictions
   * avatar picture: max 3MB, jpg,jpeg, and png formats
2. The user has to be capable of deleting a selected picture.
3. Upon clicking the Submit button, the user has to be redirected to the User Details page.
4. Submit button has to save user details.

#### User Details page

1. All given details have to be visible.
2. Details have to be in separate lines.
3. Avatar picture has to be visible, if it is not available then a placeholder image.
4. Details have to be session persistent.

## Technical Details

#### Data storage:

The application stores the details in the localstorage.   
Pros: Fast and does not need a backend.   
Cons: Users can edit and delete from dev tools. Later when the app is connected to the backend has to keep up to date. It can become async with the backend. The picture has to be converted to base64 to store it as a string. Lack of space so the picture size has to be limited.

Alternative solution: I would recommend storing the data on the backend, and generating a session ID to track the user.

#### Styling:

Currently, the app uses basic css, and the pages use MaterialUI.   
The specification asked not to use CSS frameworks for some parts of the app so it is based on basic css. For the pages I choose Material UI because among the frameworks I am the most familiar with it. Furthermore, I like the technique of how Material UI helps build the application style.

For alternatives, any css framework can be used or it can be done without it as well. In my opinion, these tools lead to faster development, easier maintenance, and make redesigning or rebranding much more simple. In the case of Material UI a well-written theme-base can be used in several projects.

#### Routing:

The project uses ‘react-router-dom’. I do not have that much experience with SPA routing, so I did some research and chose this because it looked simple and straightforward.

#### Project structure:

