# Exercise: Intro to React.js

## Compose Plain JSX

As an introduction to React, convert the provided HTML and CSS files to a React application. **There is no need to use components yet**, just play around with the import/export functionality that JSX provides. Examine the provided files and implement it with JSX.

## JSX With Reusable Parts

Try to find any parts that of the markup from Task 1 that are similar and create parametrized functions returning HTML to reduce code repetition.

For instance, the majority of the HTML might be placed inside a variable named app, which you pass to ReactDOM.render(), while the contact-card of each user can be a function makeContact(data), that receives a user object and returns HTML based on the data. Import the list of contacts from the provided JSON file and iterate over the array, passing each object to the makeContact() function. Store the results in a variable and place that inside the HTML of app.

## Increment Counter

Create a simple app, which shows a number that is initially set to zero and a button. When the button is clicked, the displayed number is incremented by one. Do not use components, instead manually call ReactDOM.render() whenever the button is clicked and change of the value is required. Observe the DOM tree through your browser’s dev tools to see which parts of it are updated with every change. Try to add more text nodes and see if there’s a difference in the amount of updated text.

## Timer

Create a simple app, indicating the current time, which updates automatically every second. Do not use components, instead manually call ReactDOM.render() using a setInterval call. Observe the DOM tree through your browser’s dev tools to see which parts of it are updated with every change. Try to add more text nodes and see if there’s a difference in the amount of updated text.

## \* App Refactoring

Implement interactive functionality to the project, provided with the first task (Contact Book). When each of the contacts from the list on the left is clicked, display detailed information into the main area on the right. Do not use components, instead manually call ReactDOM.render() whenever a change in the page is required.