# React, A JavaScript library for building user interfaces

# Introduction:

In this course we are going to get to know the basics of React. We will start with simple one file applications and proceed to more challenging projects, eventually creating a fully working Tic-Tac-Toe game with a couple of fun additions. This course is **not** aimed towards learning CSS, so the UI side is completely in your control (excluding the tic-tac-toe game), feel free to make it look the way you want.

[Here](link) is a demo of the project and the applications we’re going to learn to create

(Jos saa hostattua ton valmiin projektin jossain nii tähän linkki siihen että sais vähän ideaa mitä lähetään opetteleen)

In this course we’re going to learn about the following features:

* [onClick function](https://www.javatpoint.com/javascript-onclick-event) with React
* [React Router](https://v5.reactrouter.com/web/guides/quick-start)
* The React-hooks: [useState](https://reactjs.org/docs/hooks-state.html) & [useEffect](https://reactjs.org/docs/hooks-effect.html)
* [React props](https://reactjs.org/docs/components-and-props.html)
* JavaScript [map](https://reactjs.org/docs/lists-and-keys.html) function to create React elements from arrays

What is React?

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “**components**”. Many major websites are built with React. In fact, it was created and deployed by **Facebook** in 2011. Other big websites that you might have heard of use React are **Instagram** and **Netflix**. React was created with a single focus: to create components for web applications. A React component can be anything in your web application like a **Button**, **Text**, **Label**, or a **Grid**.

Why use React?

React is remarkably **flexible** and after learning it, you can use it on a vast variety of platforms. React makes it painless to create interactive UIs. Design simple views for each state in your application and React will efficiently update and render **just the right components** when your data changes.

## Installing and launching React project

**DISCLAIMER!**

React does not work without node (and the **npm package manager** included in node). If you don’t already have it on your computer, go [here](https://phoenixnap.com/kb/install-node-js-npm-on-windows) to see the instructions on how to install it. You can’t continue the steps before you do.

Link to download node: <https://nodejs.org/en/download/>

### Step 1: Create the root folder

**Create a folder** on your system that you want to install your project and open that folder.

### Step 2: Create your React app

Open command prompt on your root folder and enter the following command:

npx create-react-app my-app

You can name the project something different if you want, just change “**my-app**” to something else of your choosing.

This will take a minute.

### Step 3: Launch your project

Move to your **project folder**

cd my-app

And launch your project with the following command:

npm run start

**You will use the command above every time you want to launch the developer server!**

The project will open on port 3000 by default (if not changed or the port already occupied). Your browser should **automatically open** the project on your default browser.

If not, go to the address: <http://localhost:3000>.

You should see this view:

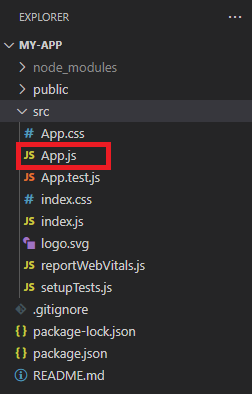
Graphical user interface

Description automatically generated with medium confidence

### Step 4: Clearing the App.js file to our needs

You can see the project has quite few folders and files. Right now, we will just have to focus on the **src** folder and the **App.js** file. The App.js will work as a “**base**” to our project. We will **render our components** there, but more on that on the first exercise.

Now, open the file in the editor you will be using (the examples are written with [Visual Studio Code](https://code.visualstudio.com/)):



It will look like this:

import logo from './logo.svg';

import './App.css';

function App() {

  return (

    <div className="App">

      <header className="App-header">

        <img src={logo} className="App-logo" alt="logo" />

        <p>

          Edit <code>src/App.js</code> and save to reload.

        </p>

        <a

          className="App-link"

          href="https://reactjs.org"

          target="\_blank"

          rel="noopener noreferrer"

        >

          Learn React

        </a>

      </header>

    </div>

  );

}

export default App;

You can scrap everything and **replace it with the following code**.

import React from "react";

function App() {

    return (

        <div>

        </div>

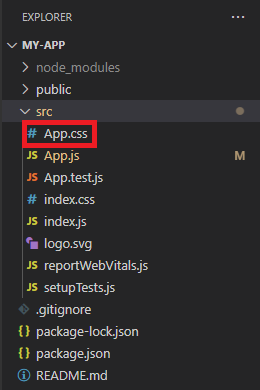
    );

}

export default App;

**DISCLAIMER!**

If you want to style your project, go to the App.css file and scrap everything from there.



And add the CSS import back to your App.js file under the React import:

import React from "react";

import './App.css';

Style the project to your liking as we continue.

Now you should see nothing but a blank page in your browser. This is where we start building our own project, in the first exercise. **See you there!** 😊