

# **Lab Exercises**

## **Web Development Frameworks**



**zenika**  
ARCHITECTURE INFORMATIQUE

## Lab 2: React Todo App

---

In this lab, we will explore building a Todo app using React!

### Create a new React project

Install create-react-app :

```
npm install -g create-react-app
```

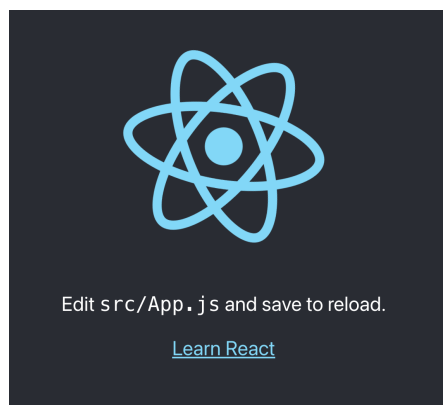
Then, use this module to create a client application

```
create-react-app react-todo-app
```

To run application:

```
cd react-todo-app  
npm start
```

You can now open the browser at `http://localhost:3000` to view the application. You should be greeted by this welcome screen.



Change one of the text strings in `App.js` to view hot-reload in action.

### Prepare the UI mockup

Open `App.css` in the `src` folder and paste the following css.

```
html,  
body {  
  margin: 0;  
  height: 100%;  
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,  
    Oxygen-Sans, Ubuntu, Cantarell, "Helvetica Neue", sans-serif;  
}  
  
#root {  
  height: 100%;  
}
```

```
main {
  display: flex;
  align-items: center;
  justify-content: center;
  background-color: #fefefe;
  background-image: linear-gradient(#fc6c48 0%, #ef5081 100%);
  background-repeat: no-repeat;
  background-size: cover;
  height: 100%;
}

.todo-wrapper {
  width: 400px;
  max-width: 100%;
  min-height: 500px;
  margin: 20px auto 40px;
  border: 1px solid #eee;
  border-radius: 4px;
  padding: 40px 20px;
  -webkit-box-shadow: 0 0 15px 0 rgba(0, 0, 0, 0.05);
  box-shadow: 0 0 15px 0 rgba(0, 0, 0, 0.05);
  background-color: #f4f7fc;
  overflow: hidden;
  position: relative;
}

.todo-title {
  font-size: 1.2em;
  color: #f65c65;
  font-weight: normal;
}

.btn,
input {
  line-height: 2em;
  border-radius: 3px;
  border: 0;
  display: inline-block;
  margin: 15px 0;
  padding: 0.2em 1em;
  font-size: 1em;
}

input[type="text"] {
  border: 1px solid #ddd;
  min-width: 70%;
  transition: all ease-in 0.25s;
}

input:focus {
  outline: none;
  border: 1px solid #a3b1ff;
}

.btn {
```

```

    text-align: center;
    font-weight: bold;
    cursor: pointer;
    border-width: 1px;
    border-style: solid;
}

.btn-add {
  background: #6664ff;
  border-color: #6664ff;
  pointer-events: visible;
  color: #fefefe;
  min-width: 12%;
  transition: all ease-in 0.25s;
  font-size: 2.2em;
  line-height: 0.5em;
  padding: 0.3em 0.3em;
  float: right;
}

ul.todo-list {
  padding: 0;
  margin-bottom: 30px;
}

ul.todo-list li {
  position: relative;
  list-style-type: none;
  display: block;
  margin: 10px 0;
  background: #e0e8f5;
  border-radius: 3px;
  padding: 12px;
  overflow: hidden;
}

```

Then in `src/App.js` file, paste the following code.

```

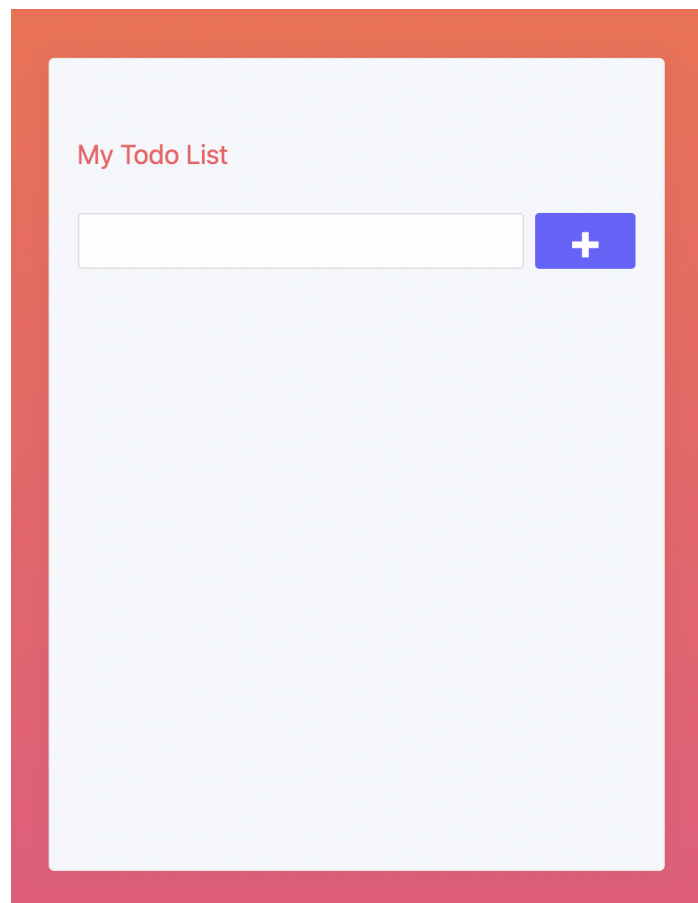
import './App.css';

function App() {
  return (
    <main>
      <section className="todo-wrapper">
        <h2 className="todo-title">My Todo List</h2>
        <input type="text" />
        <div className="btn btn-add">+</div>
        <ul className="todo-list">{/* <!-- list of todos here --> */}</ul>
      </section>
    </main>
  );
}

export default App;

```

The app should now look like this.



## Adding Todos

In the `App.js` file, we will declare two variables for input value and the todos list and the function methods to set the input values and add new todo into the `todos` variable.

```
import { useState } from "react";
import "./App.css";

function App() {
  const [inputValue, setInputValue] = useState("");
  const [todos, setTodos] = useState([]);

  function onInputChange(e) {
    setInputValue(e.target.value);
  }

  function addTodo() {
    setTodos([...todos, inputValue]);
    setInputValue("");
  }

  ...
  ...
  ...
}
```

Now our logic is ready. We can wire it into our render elements.

```

...
...
...
    <input
      type="text"
      value={inputValue}
      onChange={onInputChange}
      onKeyDown={onEnterPress}
    />
    <div className="btn btn-add" onClick={addTodo}>
      +
    </div>
...
...
...

```

Lastly, we will display the todo list in the `ul` tag.

```

<ul className="todo-list">
  {todos.map((todo, i) => (
    <li key={` ${todo} - ${i}`}>{todo}</li>
  ))}
</ul>

```

That's all! You can now insert new todo into the list!