# REACT TODO



## Introduction

React is a popular JavaScript library used for building user interfaces. In this plan, we will be creating a TODO application using React.

Guide NodeJS environment installation and configuration

Guide GitHub repository creation

Guide Basic architecture of fronted applications - MVC

## Step-by-step Plan

**Task 1: Create a Project using create-react-app**

Initialize a project using [create-react-app tool](https://create-react-app.dev/) and investigate its structure. Add Github Action for project build and publishing on GitHub Pages (<https://github.com/marketplace/actions/deploy-react-to-github-pages>)

**Task 2: Create project markup**

Create the project markup using SCSS Modules.

**Task 3: Split the application into components - Atomic Design**

Explore Atomic Desing and splitting up the APP using it.

[Atomic Design by Brad Frost](https://atomicdesign.bradfrost.com/)

* [Atomic Design: Getting Started](https://medium.com/@WeAreMobile1st/atomic-design-getting-started-916bc81bad0e)
* [Atomic Design Principles & Methodology 101 | Adobe XD Ideas](https://xd.adobe.com/ideas/process/ui-design/atomic-design-principles-methodology-101/)
* [Principles of atomic design](https://medium.com/galaxy-ux-studio/principles-of-atomic-design-7b03a30c3cb6)
* [Atomic Design in React & React Native Applications](https://paulonteri.com/thoughts/atomic-design-react)
* <https://uxdesign.cc/atomic-design-2022-what-we-can-learn-from-eames-and-other-design-giants-4d8e2f579daa>

**Task 4: Adding interactivity with Redux Toolkit - State**

Install Redux Toolkit and add application state

**Task 5: Adding interactivity with Redux Toolkit - Actions Add/Check/Delete**

Implement business logic of the application

**Task 6: Adding interactivity with Redux Toolkit - Getters filters**

Implement filtration and elements counter logic

**Task 7: LocalStorage as Repository (Not redux persist)**

Add functionality to save data into browser storage

**Task 8: ESLint/Stylelint/Husky tools**

Add tools for static code analyze and fix errors (AirBnB)

**Task 9: Unit tests - snapshots, coverage**

Add testing tools and cover all components with snapshots. Explore how to cover the project with unit tests.

**Task 10: Unit tests - components**

Cover components with unit tests

**Task 11: Unit test - store**

Cover redux store/actions with unit tests

**Task 12: Unit test - repository**

Cover repository with unit tests

**Task 13: E2E tests Cypress**

For E2E tests add Cypress library and cover the application with E2E tests

Collect Feedback

Questions for self-checking:

What is React?

What is Virtual DOM and how does it work?

What is a prop?

What is the state in React?

What are the components in React?

What is the difference between class and functional components

What are the side effects of containing React loops?

What is JSX?

What is Flux?

What is Redux?

What is the state in Redux?

What is a pure function?

What are events in React?

How to create in React?

What do you know about React references?

What are High Order Components (HOC)?

What does Store mean in Redux?

What are the benefits of Redux?

What is React routing and why is it needed?

What is atomic design

What is Clean Code

What is redux-thunk

What are hooks in functional components?

How to work with input fields and forms in react?

What are modular CSS styles?

How to do conditional rendering in React?

React interview questions - <https://github.com/sudheerj/reactjs-interview-questions>

Action points:

* Replace screenshot with figma layout file
* React Router
* Separate project with bundlers (Vite and others)
* Separate project Auth (Session, JWT, OAuth)