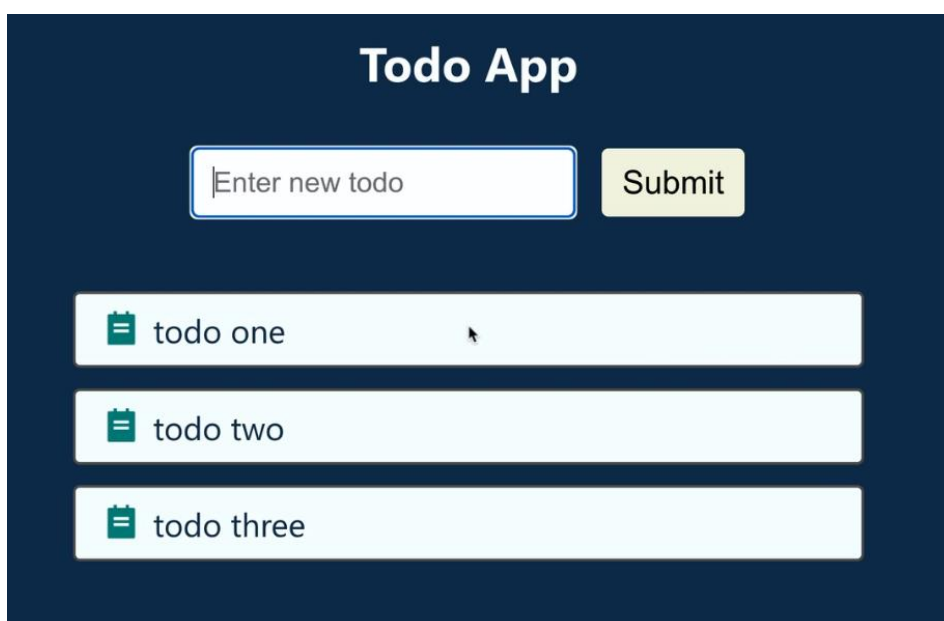
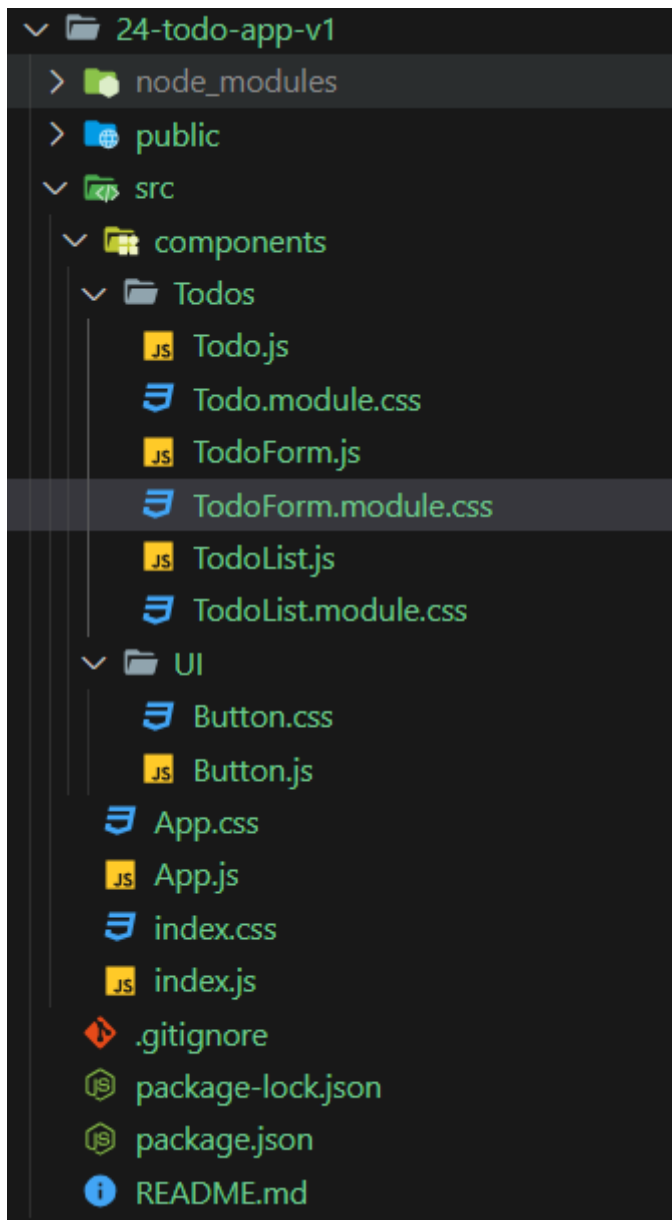


Первая версия проекта Todo App



Начнем проект с создания заготовок компонент и постепенно будем добавлять функционал.

Создадим форму и список задач. Тексты для задач пока поместим в массив строк. Пока без стиливого оформления.

```
JS Todo.js U X JS TodoList.js U
24-todo-app-v1 > src > components > Todos > JS Todo.js > ...
1  function Todo() {
2    return <h1>This is Todo</h1>;
3  }
4
5  export default Todo;
6
```

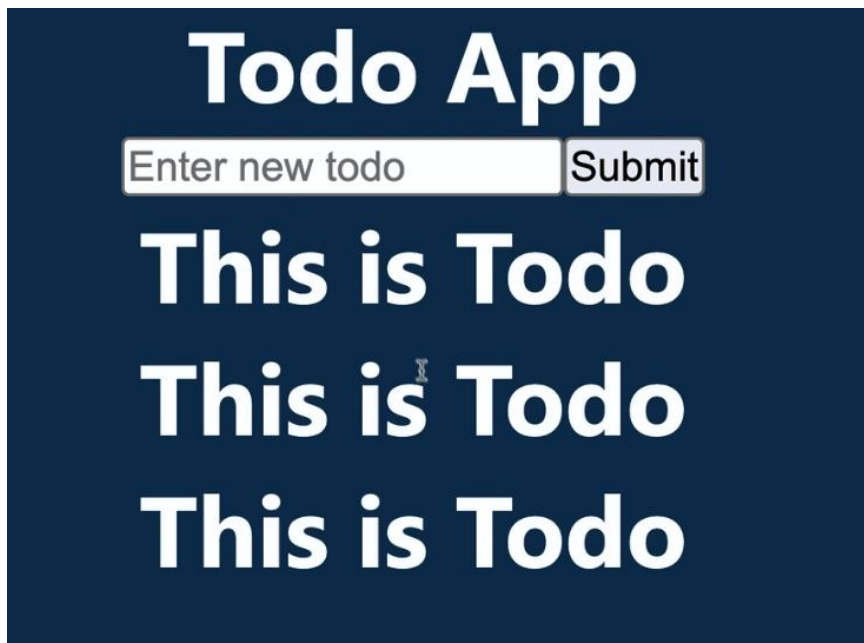
```
JS TodoList.js U X JS TodoForm.js U JS App.js U
24-todo-app-v1 > src > components > Todos > JS TodoList.js > [?] default
1  import Todo from './Todo';
2
3  function TodoList() {
4    return (
5      <>
6        <Todo />
7        <Todo />
8        <Todo />
9      </>
10   );
11 }
12
13 export default TodoList;
14
```

```
JS TodoList.js U JS TodoForm.js U X JS App.js U
24-todo-app-v1 > src > components > Todos > JS TodoForm.js > default
1 function TodoForm() {
2   return (
3     <form>
4       <input placeholder="Enter new todo" />
5       <button type="submit">Submit</button>
6     </form>
7   );
8 }
9
10 export default TodoForm
```

```
JS TodoList.js U JS TodoForm.js U JS App.js U X
24-todo-app-v1 > src > JS App.js > default
1 import TodoForm from './components/Todos/TodoForm';
2 import TodoList from './components/Todos/TodoList';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <h1>TodoApp</h1>
9       <TodoForm />
10      <TodoList />
11    </div>
12  );
13 }
14
15 export default App;
16
```

Создание заготовок закончили и переходим к стилизации

```
index.css U X
24-todo-app-v1 > src > index.css > ...
1  * {
2    margin: 0;
3    padding: 0;
4  }
5
6  body {
7    font-family: 'Segoe UI', 'Roboto', 'Oxygen',
8    'Ubuntu', 'Cantarell',
9    'Fira Sans', 'Droid Sans', 'Helvetica Neue',
10   sans-serif;
11    color: white;
12    background-color: #112d49;
```



Реализация базового функционала.

Сделаем так, чтобы форма работала, а ниже появлялись задачи. Для этого необходимо создать состояния, в котором мы будем хранить массив строк, т.е. каждая задача – это строка. Состояние разумно завести в App.js, т.к. в нем есть дочерние компоненты TodoForm и TodoList. В TodoForm мы будем добавлять новые компоненты, а в TodoList будем их отображать. Нам необходимо иметь возможность изменять состояние в TodoForm, т.е. иметь возможность вызывать setTodos. Кроме этого мы должны передавать список задач в компонент TodoList.

```
JS App.js M X JS TodoList.js M JS Todo.js M
24-todo-app-v1 > src > JS App.js > ...
1  import { useState } from 'react';
2  import TodoForm from './components/Todos/TodoForm';
3  import TodoList from './components/Todos/TodoList';
4  import './App.css';
5
6  function App() {
7    const [todos, setTodos] = useState([]);
8
9    return (
10     <div className="App">
11       <h1>TodoApp</h1>
12       <TodoForm />
13       <TodoList todos={todos} />
14     </div>
15   );
16 }
17
18 export default App;
```

```
JS App.js M JS TodoList.js M X JS Todo.js M
24-todo-app-v1 > src > components > Todos > JS TodoList.js > default
1  import Todo from './Todo';
2
3  function TodoList({ todos }) {
4    return todos.map((todo, index) => <Todo key={index} todo={todo} />);
5  }
6
7  export default TodoList;
```

```
JS App.js M JS TodoList.js M JS Todo.js M JS TodoForm.js
24-todo-app-v1 > src > components > Todos > JS Todo.js > ...
1 function Todo({ todo }) {
2   return <h3>{todo}</h3>;
3 }
4
5 export default Todo;
6
```

React App



Список задач пока пуст, но можно пока статически в массив добавить несколько задач:

```
JS App.js M X JS TodoList.js M JS Todo.js M JS TodoForm.js
24-todo-app-v1 > src > JS App.js > default
1 import { useState } from 'react';
2 import TodoForm from './components/Todos/TodoForm';
3 import TodoList from './components/Todos/TodoList';
4 import './App.css';
5
6 function App() {
7   const [todos, setTodos] = useState(['todo one', 'todo two']);
8
9   return (
10     <div className="App">
```

React App



Реализуем функционал добавления задач с помощью формы

```
JS App.js M X JS TodoList.js M JS Todo.js M JS TodoForm.js M
24-todo-app-v1 > src > JS App.js > ...
1  import { useState } from 'react';
2  import TodoForm from './components/Todos/TodoForm';
3  import TodoList from './components/Todos/TodoList';
4  import './App.css';
5
6  function App() {
7    const [todos, setTodos] = useState([]);
8    const addTodoHandler = (text) => {
9      setTodos([...todos, text]);
10   };
11
12   return (
13     <div className="App">
14       <h1>Todo App</h1>
15       <TodoForm addTodo={addTodoHandler} />
16       <TodoList todos={todos} />
17     </div>
18   );
19 }
20
21 export default App;
22
```

```
JS TodoForm.js M X JS App.js M JS TodoList.js M JS Todo.js M
24-todo-app-v1 > src > components > Todos > JS TodoForm.js > default
1  import { useState } from 'react';
2
3  function TodoForm({ addTodo }) {
4    const [text, setText] = useState('');
5
6    function onSubmitHandler(event) {
7      event.preventDefault();
8      addTodo(text);
9      setText('');
10   }
11
12   return (
13     <form onSubmit={onSubmitHandler}>
14       <input
15         placeholder="Enter new todo"
16         value={text}
17         onChange={(e) => setText(e.target.value)}
18       />
19       <button type="submit">Submit</button>
20     </form>
21   );
22 }
23
24 export default TodoForm;
```

React App

