Creating a Simple React Todo List Application

Setup Steps

1. **Install Node.js and npm:** Ensure you have Node.js and npm installed on your machine. You can download them from Node.js official website.
2. **Create a New React Application:** Use the create-react-app tool to set up a new React application.

   ```
   npx create-react-app todo-list
   cd todo-list
   ```

3. **Install Dependencies:** For this simple todo list application, we'll use the base dependencies provided by create-react-app.
4. **Start the Development Server:** Start the React development server to see the application in action.

   ```
   npm start
   ```

Code Snippets

Step 1: Structure the Application We will create a simple todo list application with the following components:

- App: The main component that holds the state and renders the list and form.
- TodoItem: A component to display individual todo items.
- TodoForm: A component to handle adding new todos.

Step 2: Create the Components

App Component (src/App.js):

```
import React, { useState } from 'react';
import TodoForm from './TodoForm';
import TodoItem from './TodoItem';
import './App.css';

const App = () => {
  const [todos, setTodos] = useState([]);

  const addTodo = (text) => {
    const newTodos = [...todos, { text, completed: false }];
    setTodos(newTodos);
  };

  const toggleComplete = (index) => {
    const newTodos = todos.map((todo, i) => {
      if (i === index) {
        return { ...todo, completed: !todo.completed };
      }
      return todo;
    });
    setTodos(newTodos);
  };
```

```
  const removeTodo = (index) => {
    const newTodos = todos.filter((_, i) => i !== index);
    setTodos(newTodos);
  };

  return (
    <div className="App">
      <h1>Todo List</h1>
      <TodoForm addTodo={addTodo} />
      <div className="todo-list">
        {todos.map((todo, index) => (
          <TodoItem
            key={index}
            index={index}
            todo={todo}
            toggleComplete={toggleComplete}
            removeTodo={removeTodo}
          />
        ))}
      </div>
    </div>
  );
};

export default App;
```

TodoForm Component (src/TodoForm.js):

```
import React, { useState } from 'react';

const TodoForm = ({ addTodo }) => {
  const [value, setValue] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    if (!value) return;
    addTodo(value);
    setValue('');
  };

  return (
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        className="input"
        value={value}
        onChange={(e) => setValue(e.target.value)}
        placeholder="Add a new todo"
      />
      <button type="submit">Add</button>
    </form>
  );
};

export default TodoForm;
```

TodoItem Component (src/TodoItem.js):

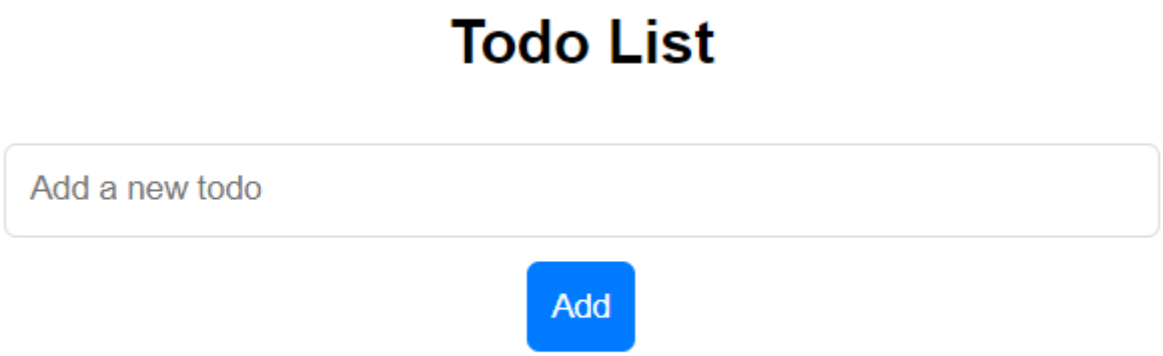
```
import React from 'react';

const TodoItem = ({ todo, index, toggleComplete, removeTodo }) => {
  return (
    <div className="todo">
      <span
        style={{ textDecoration: todo.completed ? 'line-through' : '' }}
        onClick={() => toggleComplete(index)}
      >
        {todo.text}
      </span>
      <button onClick={() => removeTodo(index)}>x</button>
    </div>
  );
};

export default TodoItem;
```

App CSS (src/App.css):

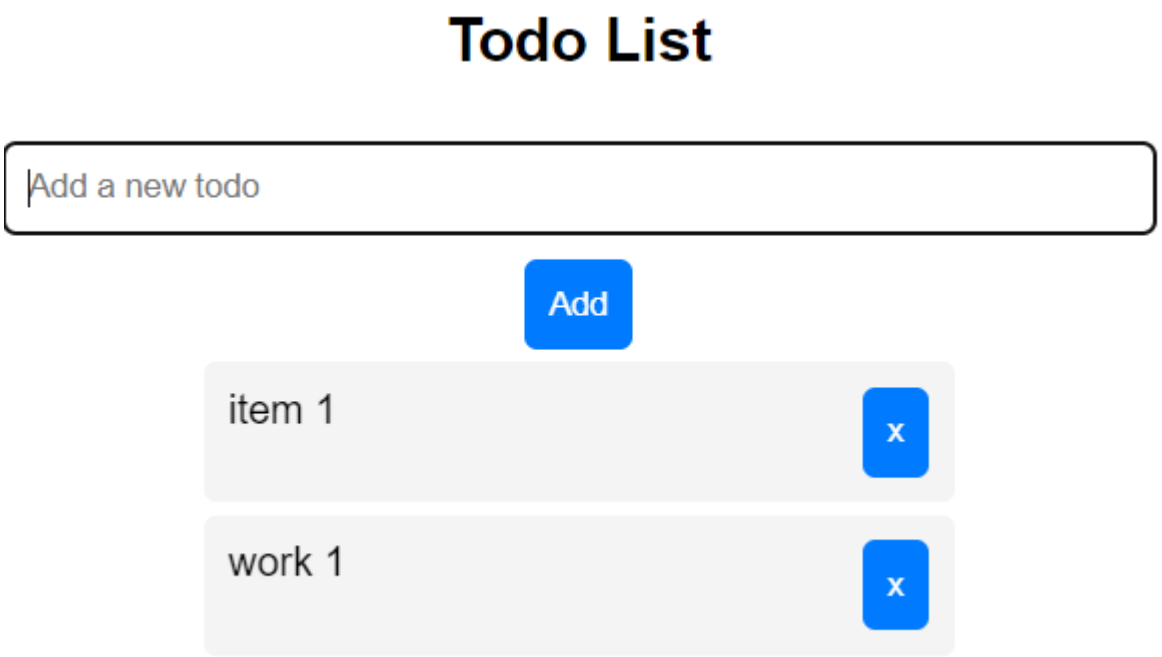
```
.App {
  text-align: center;
}
.todo-list {
  margin: 0 auto;
  width: 300px;
}
.todo {
  display: flex;
  justify-content: space-between;
  background: #f4f4f4;
  margin: 5px 0;
  padding: 10px;
  border-radius: 5px;
}
.todo span {
  cursor: pointer;
}
input {
  padding: 10px;
  margin: 10px 0;
  width: calc(100% - 24px);
  border: 1px solid #ddd;
  border-radius: 5px;
}
button {
  padding: 10px;
  border: none;
  background: #007bff;
  color: #fff;
  cursor: pointer;
  border-radius: 5px;
}
button:hover {
  background: #0056b3;
}
```

Screenshots

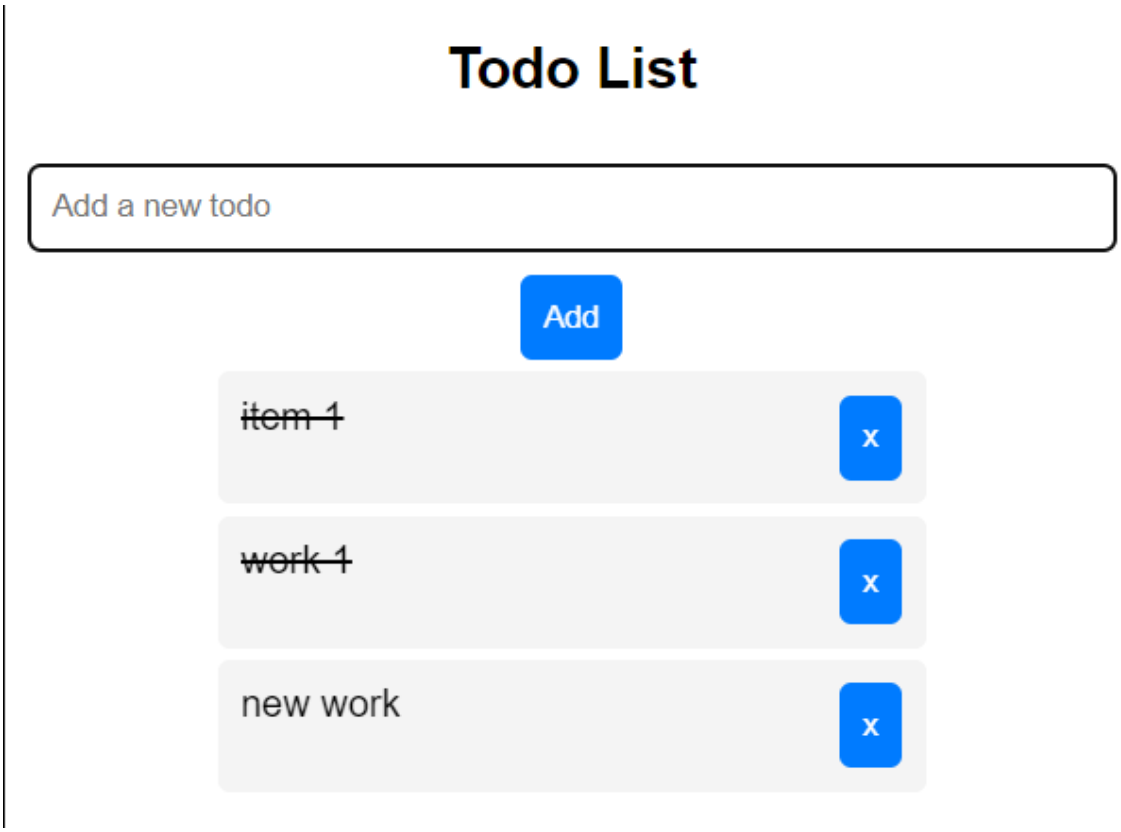
1. **Initial Todo List Page:**



2. **Adding a Todo Item:**



3. **Marking Todo Item as Complete:**



4. **Deleting Items:**