Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированной обработки информации (АОИ)

**FRONTEND-ПРИЛОЖЕНИЕ «СПИСОК ДЕЛ»**

Отчет о выполнении лабораторной работы

по дисциплине «Разработка интернет-приложений»

Студент гр. 429-3

\_\_\_\_\_\_\_\_\_\_ Бабец А. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2021 г.

Ассистент каф. АОИ

\_\_\_\_\_\_\_\_\_\_ Владимиров М. В.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2021 г.

Томск 2021

**Оглавление**

[1 Введение 3](#_Toc83149604)

[2 Основная часть 4](#_Toc83149605)

[2.1 Описание используемой среды программирования 4](#_Toc83149606)

[2.2 Описание метода решения задачи 4](#_Toc83149607)

[2.3 Описание фрагментов кода 5](#_Toc83149608)

[2.4 Описание функциональных возможностей разработанного приложения. 6](#_Toc83149609)

[3 Заключение 7](#_Toc83149611)

[Приложения 8](#_Toc83149612)

# Введение

Целью лабораторной работы является разработка frontend-приложения для каталога электроники.

Задачи лабораторной работы следующие:

* Выбор современного фреймворка;
* Изучение основ фреймворка;
* Реализовать frontend-приложение.

# Основная часть

## Описание стека технологий

Для разработки frontend-приложения было принято решение воспользоваться библиотекой ReactJS для Javascript. React – это инструмент для создания пользовательских интерфейсов. Его главная задача – обеспечение вывода на экран того, что можно видеть на веб-страницах. React значительно облегчает создание интерфейсов благодаря разбиению каждой страницы на небольшие фрагменты. Мы называем эти фрагменты компонентами.

Также писать код будем в Visual Studio Code, так как это бесплатная IDE с широким функционалом.

## Описание метода решения задачи

Для реализации приложения необходимо реализовать сами страницы, роутер, для перемещения по страницам.

## Описание фрагментов кода

Создадим React проект с помощью команды “npx create-react-app”.

Далее создадим папку «components» в ней будут храниться все компоненты.

Перейдём к описанию основных функций. Создадим форму.:

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

import TextField **from** '@material-ui/core/TextField'**;**

const TodoForm **=** **({** saveTodo **})** **=>** **{**

const **[value,** setValue**]** **=** useState**(**''**);**

**return** **(**

**<**form

onSubmit**={event** **=>** **{**

**event.**preventDefault**();**

saveTodo**(value);**

setValue**(**''**);**

**}}>**

**<**TextField

variant**=**"outlined"

placeholder**=**"Add todo"

margin**=**"normal"

onChange**={event** **=>** **{**

setValue**(event.**target**.value);**

**}}**

**value** **=** **{value}**

**/>**

**</**form**>**

**);**

**};**

export **default** TodoForm**;**

Создадим список дел:

import React **from** 'react'**;**

import List **from** '@material-ui/core/List'**;**

import ListItem **from** '@material-ui/core/ListItem'**;**

import ListItemSecondaryAction **from** '@material-ui/core/ListItemSecondaryAction'**;**

import ListItemText **from** '@material-ui/core/ListItemText'**;**

import Checkbox **from** '@material-ui/core/Checkbox'**;**

import IconButton **from** '@material-ui/core/IconButton'**;**

import DeleteIcon **from** '@material-ui/icons/Delete'**;**

const TodoList **=** **({** todos**,** deleteTodo **})** **=>** **(**

**<**List**>**

**{**todos**.**map**((**todo**,** index**)** **=>** **(**

**<**ListItem key**={**index**.**toString**()}** dense button**>**

**<**Checkbox tabIndex**={-**1**}** disableRipple **/>**

**<**ListItemText primary**={**todo**}** **/>**

**<**ListItemSecondaryAction**>**

**<**IconButton

aria**-**label**=**"Delete"

onClick**={()** **=>** **{**

deleteTodo**(**index**);**

**}}**

**>**

**<**DeleteIcon **/>**

**</**IconButton**>**

**</**ListItemSecondaryAction**>**

**</**ListItem**>**

**))}**

**</**List**>**

**);**

export **default** TodoList**;**

Листинг кода представлен в приложении Б.

## Описание функциональных возможностей разработанного приложения

Запустив приложение, перед нами появится окно с главной страницей, на которой есть поле ввода названия дела и список дел ниже.

При нажатии на клавишу «Enter» при введённом названии дела, оно добавится в список.

Скриншоты приложения представлен в приложении Б.

# Заключение

В практической работе были достигнуты поставленные задачи. Мы познакомились с библиотекой ReactJS и изучили её основы. Разработали проект для каталога электроники.

# Приложения

# Приложение А

# (обязательное)

# Листинг кода

index.js

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

import ReactDOM **from** 'react-dom'**;**

import Typography **from** '@material-ui/core/Typography'**;**

import './styles.css'**;**

import TodoForm **from** './TodoForm'**;**

import TodoList **from** './TodoList'**;**

const App **=** **()** **=>** **{**

const **[**todos**,** setTodos**]** **=** useState**([]);**

**return** **(**

**<**div className**=**"App"**>**

**<**Typography component**=**"h1" variant**=**"h2"**>**

Todos

**</**Typography**>**

**<**TodoForm

saveTodo**={**todoText **=>** **{**

const trimmedText **=** todoText**.**trim**();**

**if** **(**trimmedText**.**length **>** 0**)** **{**

setTodos**([...**todos**,** trimmedText**]);**

**}**

**}}**

**/>**

**<**TodoList

todos **=** **{**todos**}**

deleteTodo**={**todoIndex **=>** **{**

const newTodos **=** todos

**.**filter**((**\_**,** index**)** **=>** index **!==** todoIndex**)**

setTodos**(**newTodos**)**

**}}**

**/>**

**</**div**>**

**);**

**};**

const rootElement **=** document**.**getElementById**(**'root'**);**

ReactDOM**.**render**(<**App **/>,** rootElement**);**

TodoForm.js

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

import TextField **from** '@material-ui/core/TextField'**;**

const TodoForm **=** **({** saveTodo **})** **=>** **{**

const **[value,** setValue**]** **=** useState**(**''**);**

**return** **(**

**<**form

onSubmit**={event** **=>** **{**

**event.**preventDefault**();**

saveTodo**(value);**

setValue**(**''**);**

**}}>**

**<**TextField

variant**=**"outlined"

placeholder**=**"Add todo"

margin**=**"normal"

onChange**={event** **=>** **{**

setValue**(event.**target**.value);**

**}}**

**value** **=** **{value}**

**/>**

**</**form**>**

**);**

**};**

export **default** TodoForm**;**

TodoList.js

import React **from** 'react'**;**

import List **from** '@material-ui/core/List'**;**

import ListItem **from** '@material-ui/core/ListItem'**;**

import ListItemSecondaryAction **from** '@material-ui/core/ListItemSecondaryAction'**;**

import ListItemText **from** '@material-ui/core/ListItemText'**;**

import Checkbox **from** '@material-ui/core/Checkbox'**;**

import IconButton **from** '@material-ui/core/IconButton'**;**

import DeleteIcon **from** '@material-ui/icons/Delete'**;**

const TodoList **=** **({** todos**,** deleteTodo **})** **=>** **(**

**<**List**>**

**{**todos**.**map**((**todo**,** index**)** **=>** **(**

**<**ListItem key**={**index**.**toString**()}** dense button**>**

**<**Checkbox tabIndex**={-**1**}** disableRipple **/>**

**<**ListItemText primary**={**todo**}** **/>**

**<**ListItemSecondaryAction**>**

**<**IconButton

aria**-**label**=**"Delete"

onClick**={()** **=>** **{**

deleteTodo**(**index**);**

**}}**

**>**

**<**DeleteIcon **/>**

**</**IconButton**>**

**</**ListItemSecondaryAction**>**

**</**ListItem**>**

**))}**

**</**List**>**

**);**

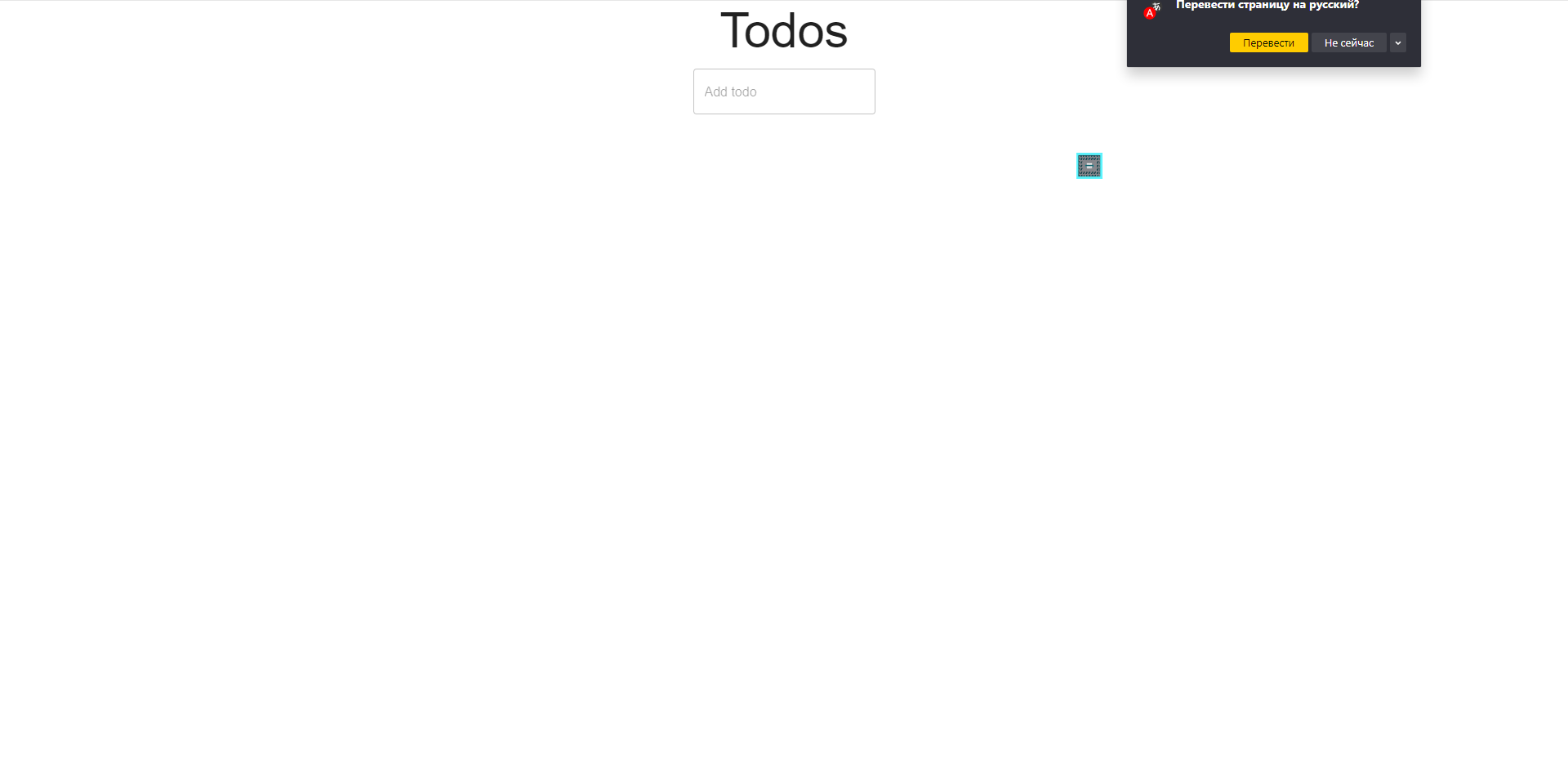
export **default** TodoList**;**

export default Admin;

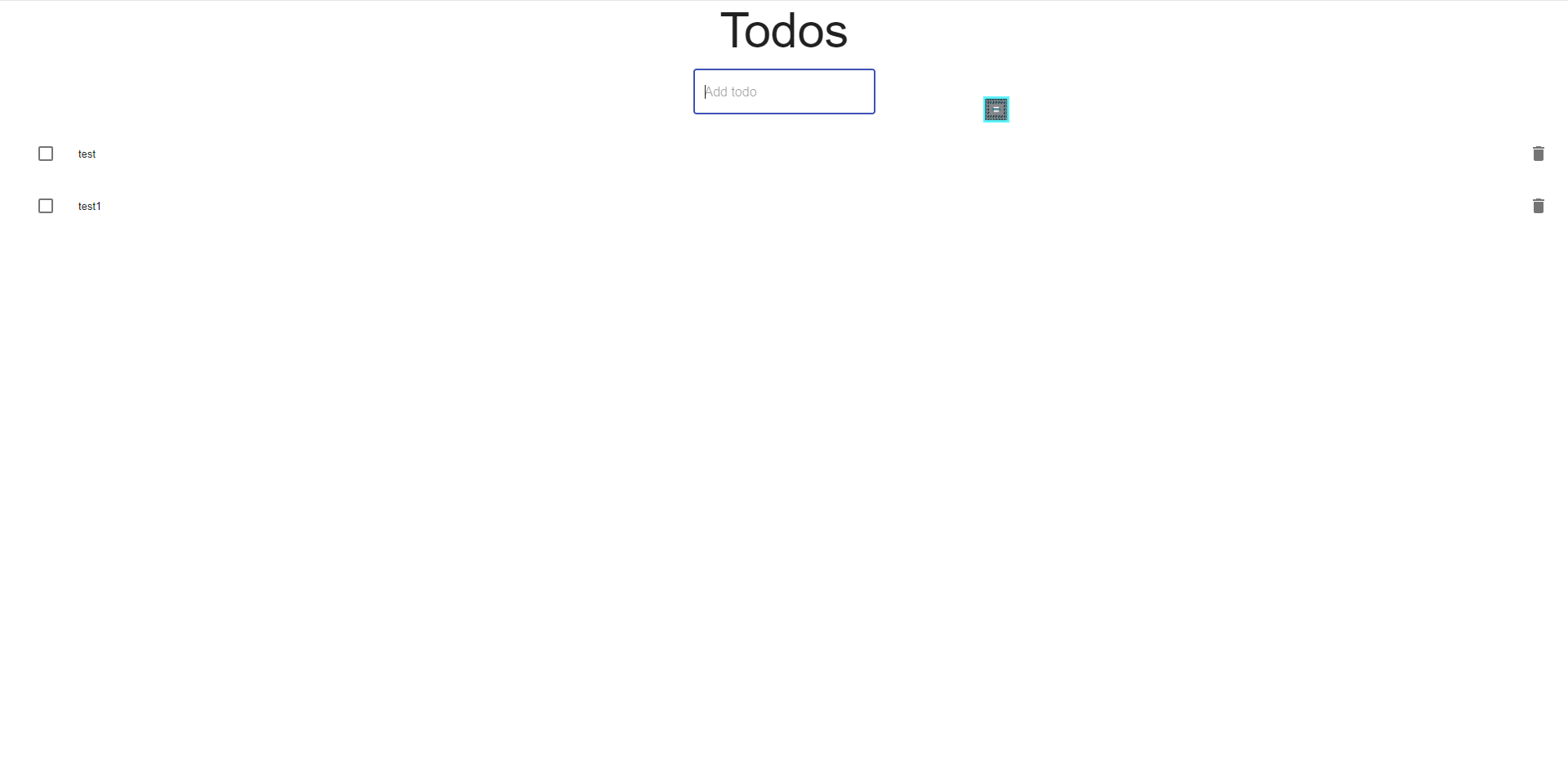
# Приложение Б

# (обязательное)

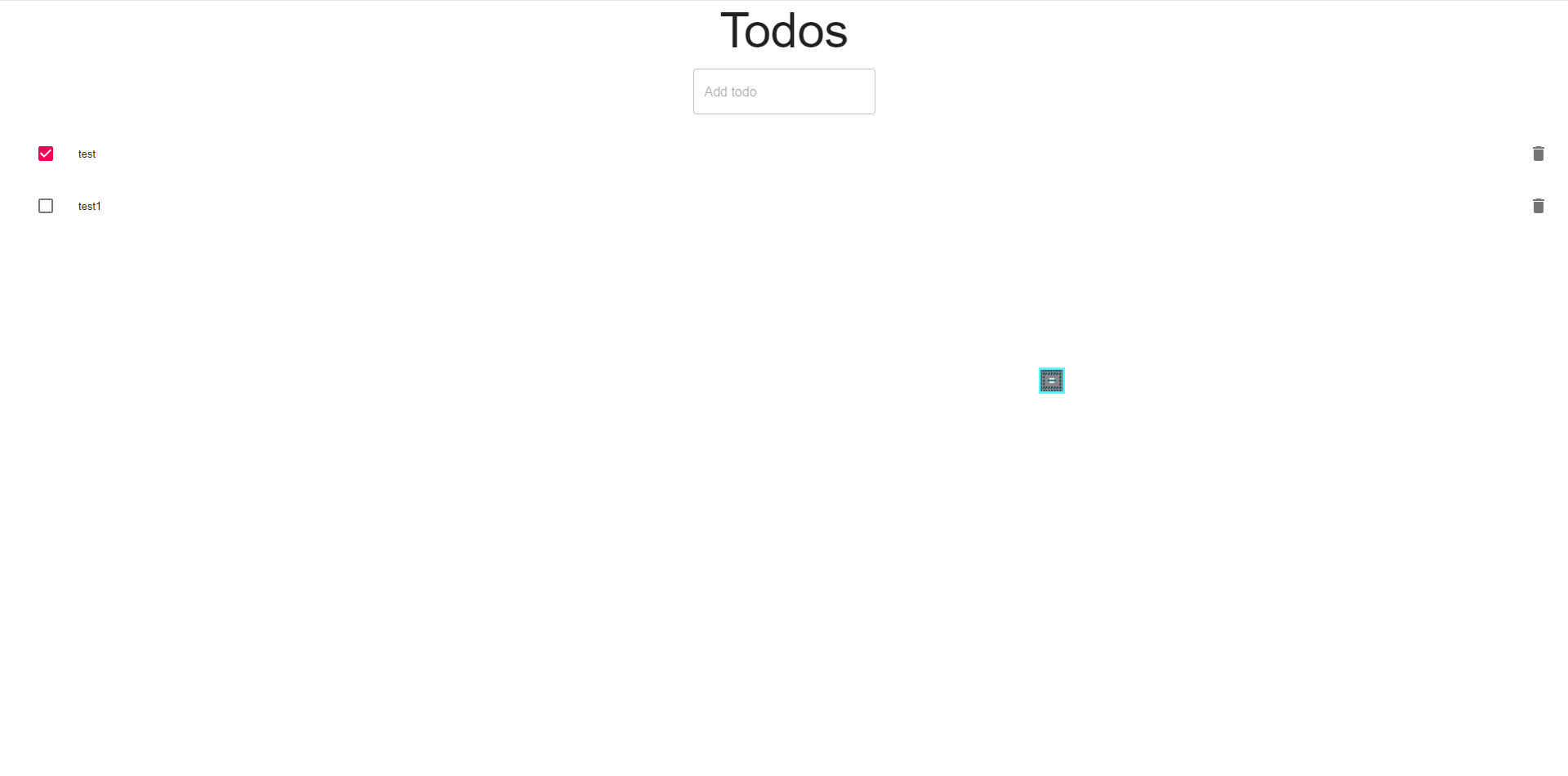
# Начальное окно приложения



# Проверка функционала приложения



# Проверка функционала приложения



# Проверка функционала приложения

