МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

Кафедра автоматизированных и вычислительных систем

КУРСОВОЙ ПРОЕКТ

по дисциплине: «Разработка приложений в Visual Studio»

Тема: «Разработка веб-приложения Kanban»

Расчетно-пояснительная записка

Разработал студент гр. бВМ-201 А. А. Чёлник

Подпись, дата Инициалы, фамилия

Руководитель В. Н. Черников

Подпись, дата Инициалы, фамилия

Нормоконтролер В. Н. Черников

Подпись, дата Инициалы, фамилия

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГТУ», ВГТУ)

Кафедра автоматизированных и вычислительных систем

# ЗАДАНИЕ

# на курсовой проект

по дисциплине «Разработка приложений в Visual Studio»

Тема проекта: «Разработка веб-приложения Kanban»

Студент группы бВМ-201: Чёлник Александр Александрович

Технические условия: Visual Studio Code, С#, .NET, ReactJS, TypeScript

Содержание и объем работы: N страниц, N рисунков

Сроки выполнения этапов: 1-й этап – 15.02.2023; 2-й этап – 12.03.2023

Срок защиты курсового проекта:

Руководитель В. Н. Черников

Подпись, дата Инициалы, фамилия

Задание принял студент А. А. Чёлник

Подпись, дата Инициалы, фамилия

# Замечания руководителя

# Содержание

Задание [на курсовой проект 2](#_Toc162295509)

[Замечания руководителя 3](#_Toc162295510)

[Содержание 4](#_Toc162295511)

[Введение 5](#_Toc162295512)

[1 Теоретическая часть 6](#_Toc162295513)

[1.1 Visual Studio Code 6](#_Toc162295514)

[1.2 Платформа разработки .NET 7](#_Toc162295515)

[1.3 Язык программирования C# 8](#_Toc162295516)

[1.4 Язык программирования TypeScript 10](#_Toc162295517)

[2 Практическая часть 12](#_Toc162295518)

[2.1 12](#_Toc162295519)

[2.2 Backend-часть 13](#_Toc162295520)

[2.3 Frontend-часть 13](#_Toc162295521)

[2.4 Описание эндпоинтов 16](#_Toc162295522)

[Заключение 17](#_Toc162295523)

[Список литературы 18](#_Toc162295524)

# Введение

Visual Studio Code (VS Code) – это мощный инструмент, разработанный компанией Microsoft, который предоставляет разработчикам все необходимые средства для создания различных программных продуктов. Он является одним из наиболее популярных среди разработчиков благодаря своей гибкости, богатым возможностям расширения и простоте использования.

В ходе курсового проекта мы углубимся в изучение возможностей Visual Studio Code и применим полученные знания для разработки реальных приложений. Мы овладеем основными функциями VS Code, такими как редактирование и отладка кода, интеграция с системами контроля версий, работа с плагинами и расширениями.

Целью данного проекта является практическое применение полученных знаний и навыков в разработке приложений с использованием Visual Studio Code. Мы изучим лучшие практики и подходы к разработке приложений, а также освоим процесс совместной работы разработчиков в рамках одной интегрированной среды.

В результате мы ожидаем получить полноценное приложение, разработанное с использованием Visual Studio Code, которое демонстрирует наши навыки в создании качественного программного продукта. Этот проект будет являться отличным доказательством нашей способности применять теоретические знания на практике.

В процессе выполнения курсового проекта мы будем активно применять практические навыки, работать в команде, решать возникающие проблемы, а также анализировать и оптимизировать наш код. Мы также будем придерживаться принципов чистого кода и эффективной организации проекта.

Наш курсовой проект по разработке приложений в Visual Studio Code станет незаменимым опытом, который поможет нам развиться как профессиональных разработчиков и открыть новые горизонты в области разработки программного обеспечения.

# 1 Теоретическая часть

## 1.1 Visual Studio Code

Visual Studio Code (VS Code) – это бесплатная интегрированная среда разработки (IDE), разработанная компанией Microsoft. Она предоставляет разработчикам инструменты для создания разнообразных программных продуктов. VS Code стал очень популярным среди разработчиков благодаря своей мощности, гибкости и удобству использования.

Одной из главных ролей VS Code в современной разработке ПО является облегчение процесса разработки и повышение производительности разработчиков. Благодаря широкому набору инструментов и поддержке различных языков программирования, VS Code позволяет разрабатывать приложения на различных платформах, включая веб-разработку, мобильную разработку и разработку для облачных сервисов.

Одним из главных преимуществ VS Code является его расширяемость. Он имеет огромное сообщество разработчиков, которые создают расширения для расширения его функциональности. Это позволяет адаптировать среду разработки под индивидуальные потребности каждого разработчика и повышает его эффективность. Расширения включают поддержку специализированных языков, интеграцию с системами контроля версий, отладчики, сниппеты и многое другое.

VS Code предлагает широкий спектр функций, которые повышают удобство использования. Он обладает отзывчивым интерфейсом с приятным дизайном и интуитивно понятным пользовательским интерфейсом. Он также поддерживает различные режимы работы, включая темную и светлую темы оформления. VS Code обладает мощным поиском и заменой текста, авто-завершением кода, разделением экрана и многочисленными другими функциями, которые делают его удобным для ежедневного использования.

Тем не менее, у VS Code есть и некоторые недостатки. Например, поскольку он предоставляет гибкость и адаптируемость, его конфигурация может быть сложной для некоторых начинающих пользователей. Кроме того, при работе с большими проектами некоторые пользователи могут заметить, что VS Code становится медленным и требует больше ресурсов компьютера.

В целом, Visual Studio Code является мощной и гибкой средой разработки ПО, которая предлагает разработчикам богатый набор инструментов и функций для создания программных продуктов. Благодаря его популярности и платформенной независимости, он стал предпочтительным выбором для многих разработчиков по всему миру и активно развивается благодаря своему активному сообществу.

## 1.2 Платформа разработки .NET

.NET (dotnet) представляет собой популярную платформу разработки, созданную компанией Microsoft, которая является ключевым инструментом для разработки программного обеспечения на языке программирования C#. .NET предоставляет набор инструментов и библиотек для эффективного создания многофункциональных и надежных приложений.

Роль .NET в современной разработке ПО на языке C# является основополагающей. С его помощью разработчики могут создавать широкий спектр приложений – от веб-приложений и мобильных приложений до десктопных приложений и облачных сервисов. Благодаря мощным инструментам .NET, разработчики могут эффективно использовать концепции объектно-ориентированного программирования, управление памятью и многопоточность, что обеспечивает высокую производительность и масштабируемость приложений.

Один из главных плюсов .NET – это его мультиплатформенность. С помощью платформы .NET можно разрабатывать приложения, которые могут работать на различных операционных системах, таких как Windows, macOS и Linux. Это упрощает процесс разработки и делает приложение доступным для более широкой аудитории.

.NET предлагает обширную библиотеку классов, которая содержит готовые компоненты и инструменты для разработки приложений. Это значительно ускоряет процесс разработки, поскольку разработчики могут использовать готовый код вместо того, чтобы писать его с нуля. Библиотека классов .NET также обеспечивает надежность и безопасность, предоставляя различные механизмы для обработки ошибок, контроля доступа и шифрования данных.

Еще одним преимуществом .NET является его интеграция с другими технологиями Microsoft. Например, .NET может легко интегрироваться с базами данных, такими как Microsoft SQL Server, и обеспечивать эффективную работу с данными. Он также интегрируется с платформой облачных вычислений Azure, что обеспечивает масштабируемость, гибкость и безопасность при развертывании приложений в облаке.

Однако .NET также имеет некоторые недостатки. Одним из них является то, что его процесс компиляции может занимать больше времени по сравнению с некоторыми другими платформами разработки. Это может оказывать влияние на производительность разработки и время запуска приложения. Также важно отметить, что некоторые возможности .NET доступны только на определенных операционных системах, что может ограничивать выбор платформы для развертывания.

В целом, .NET является мощной и гибкой платформой разработки, которая предлагает разработчикам все необходимые инструменты и ресурсы для создания качественного программного обеспечения на языке C#. Его множество преимуществ, таких как мультиплатформенность, обширная библиотека классов и интеграция с другими технологиями Microsoft, делают его одним из популярных выборов для разработки ПО.

## 1.3 Язык программирования C#

C# (C Sharp) – это современный, многоцелевой язык программирования, разработанный компанией Microsoft. Он является одним из ключевых языков для создания приложений на платформе .NET и охватывает различные сферы разработки, включая веб, мобильные, десктопные и игровые приложения. Роль C# в современной разработке ПО является важной и все более значимой.

Одним из главных преимуществ C# является его простота и понятность. C# основан на языке программирования C++, однако он приносит множество улучшений и сокращений, делая его более доступным для разработчиков. Синтаксис C# легко читается и понимается, что упрощает создание и поддержку кода. Это особенно полезно для новичков в программировании.

C# обладает солидной объектно-ориентированной моделью, что позволяет разработчикам строить модульные, гибкие и расширяемые приложения. Он поддерживает основные принципы объектно-ориентированного программирования, такие как наследование, инкапсуляция и полиморфизм. Это способствует повторному использованию кода, обеспечивает высокую абстракцию и упрощает разработку сложных систем.

Еще одним преимуществом C# является его интеграция с платформой .NET. Язык C# и платформа .NET тесно связаны друг с другом, что обеспечивает богатый набор функциональности и инструментов для разработчиков. Например, C# имеет доступ к обширной библиотеке классов .NET, которая предоставляет готовые решения для многих задач — от работы с базами данных и сетевыми протоколами до обработки графики и мультимедиа. Это значительно ускоряет процесс разработки и позволяет разработчикам фокусироваться на бизнес-логике приложения.

Язык C# обладает сильной типизацией, что обеспечивает высокую надежность и безопасность кода. Благодаря строгой проверке типов на этапе компиляции, ошибки обнаруживаются на ранних этапах разработки, что позволяет предотвратить множество проблем во время выполнения программы. Также C# поддерживает управление памятью с помощью сборки мусора, что упрощает процесс управления ресурсами и предотвращает утечки памяти.

Вместе с преимуществами C# также есть и некоторые недостатки. Один из них – это ограничение C# на использование только на платформе .NET, что может ограничить выбор платформы для развертывания приложения. Также следует отметить, что C# может быть менее производительным по сравнению с языками программирования, написанными на низкоуровневом или близком к металлу коде.

В целом, C# является мощным, универсальным и популярным языком программирования, который играет важную роль в современной разработке ПО. Его простой синтаксис, объектно-ориентированная модель, интеграция с платформой .NET и сильная типизация делают его выбором многих разработчиков. C# открывает двери для создания разнообразных приложений и позволяет разработчикам реализовывать свои идеи с высокой производительностью и надежностью.

## 1.4 Язык программирования TypeScript

TypeScript – это язык программирования, который является надмножеством JavaScript. Это означает, что любой допустимый код JavaScript также является допустимым кодом TypeScript. Однако TypeScript добавляет дополнительные функции и возможности, которые повышают эффективность и безопасность разработки.

Роль TypeScript в современной разработке веб-приложений трудно переоценить. Он широко используется в индустрии и пользуется огромной популярностью среди разработчиков. Одна из основных причин – это поддержка статической типизации. TypeScript позволяет проверять типы данных на этапе компиляции, что позволяет выявлять ошибки и обнаруживать потенциальные проблемы в коде на ранних этапах разработки. Это существенно упрощает отладку и снижает количество ошибок в продакшн-коде.

Одной из ключевых особенностей TypeScript является использование строгого синтаксиса и полноценной объектной модели программирования. Благодаря этому, разработчику становится легче организовывать большие проекты, разделять код на модули, определять интерфейсы и структуры данных. Кроме того, TypeScript расширяет язык JavaScript с помощью функциональностей, таких как классы, наследование, дженерики, декораторы и др.

TypeScript также обладает большой экосистемой библиотек и инструментов разработки. Он интегрируется с такими популярными фреймворками, как Angular, React и Vue.js, предоставляя разработчикам мощные инструменты для создания масштабируемых и поддерживаемых веб-приложений.

Однако, как и любой другой язык программирования, у TypeScript есть и некоторые недостатки. Во-первых, TypeScript требует перед компиляцией преобразования валидного кода JavaScript, что может быть непривычным для разработчиков, привыкших к динамической типизации. Это может вызвать некоторое снижение производительности в процессе разработки.

Еще одним минусом TypeScript является расширение синтаксиса JavaScript, что может привести к некоторой сложности в понимании и обучении новых разработчиков. Некоторые JavaScript-разработчики могут считать, что TypeScrcipt является излишним сложным и избегать его использования.

В итоге, TypeScript является мощным и популярным языком программирования, который играет важную роль в современной разработке веб-приложений. Он обеспечивает статическую типизацию, улучшенную структурируемость кода и совместимость с широким спектром инструментов и фреймворков. Несмотря на некоторые недостатки, TypeScript является предпочтительным выбором для многих разработчиков, стремящихся к повышению производительности и надежности своих веб-приложений.

# 2 Практическая часть

## 2.1 Мероприятия перед разработкой

Разработка веб-приложения началась с продумывания концепции, дизайна, пользовательского опыта.

Первым делом было принято решение разбить веб-приложение на две страницы:

* Главная – на которой будет находится список проектов (название и описание проекта), а также action-меню, с помощью которого можно будет добавить новый проект;
* Детальная страница проект – на которой будет находится сама канбан-доска, в которой можно будет создавать колонки и карточки внутри них.

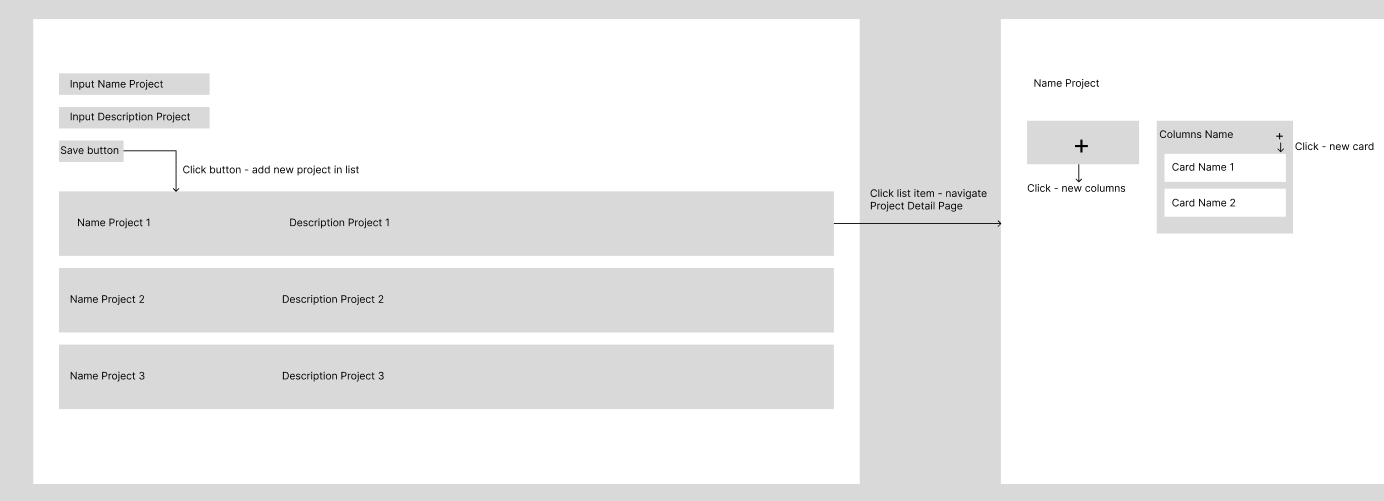


Рисунок 1 – UX веб-приложения

При заполнении двух инпутов с названием и описанием проекта становится доступна кнопка «Сохранить». При нажатии на эту кнопку в список проектов добавляется новый проект с введенными пользователем именем и описанием. Далее при клике на элемент списка открывается детальная страница проекта, на которой находится сам канбан (изначально пустой, доступна лишь кнопка «+» для создания колонки». При нажатии на кнопку «+» появляются два инпута для ввода названия и описания колонки, и кнопка сохранить при нажатии на которую в канбан добавляется новая колонка.

Внутри самой колонки можно создавать карточки, при нажатии на кнопку «+» возле названия колонки. Для колонки также можно задать название и описание.

Возможно создание неограниченного количества колонок и карточек в них.

После создания нужного числа колонок и карточек пользователь может взаимодействовать с ними: менять местами колонки (перемещать влево и вправо), менять позицию карточек в колонке (перемещать вверх и вниз), а также перемещать карточки из одной колонки в другую.

После любых действий произведенных в канбане (создание колонки, переименование колонки, удаление колонки, создание карточки, удалении карточки, изменение позиции колонки, изменение позиции карточки) появляется кнопка «Сохранить» для сохранения состояния канбана.

## 2.2 Backend-часть

## 2.3 Frontend-часть

В качестве инструментов для разработки frontend-части веб-приложения «Kanban» были выбраны:

* Библиотека React.js для создания пользовательского интерфейса сайта;
* Язык программирования TypeScript;
* Axios – библиотека JS, представляющая простой в применении интерфейс для выполнения HTTP-запросов в браузере на стороне сервера;
* Asseinfo/react-kanban – библиотека для React.js, с помощью которой можно создать примитивный канбан;
* Mobx – библиотека, дающая разработчикам инструмент для глобального использования переменных и методов между разными компонентами.

Первым делом в Visual Studio Code было создано React-приложение на TypeScript. После установки всех библиотек и зависимостей началась разработка приложения.

Первым делом были созданы страницы и настроен роутинг между ними. Далее был создан store с использованием mobx для глобального использования переменных, в котором были описаны все необходимые HTTP-запросы:

* fetchAll – получение полного списка проектов;
* fetchById – получение конкретного проекта по Id;
* create – создание проекта;
* update – сохранение изменений проекта;

Далее были созданы все необходимые модели:

1) BoardModel с полями:

* id: number;
* name: string;
* description: string;
* columns: Column[];

2) Columns с полями:

* id: number;
* title: string;
* cards: Card[];

3) Card с полями:

* id: number;
* title: string;
* description: string;

После чего были созданы простые компоненты, стили которых описывались с помощью предпроцессора SASS, а именно SCSS.

Далее в необходимых компонентах были использованы функции из store для описанных выше действий.

На главной странице при маунте компонента вызывается функция fetchAll для получения полного списка проектов и последующего мапинга их на странице. Здесь же вызывается функция create для создания нового проекта при нажатии на кнопку «Сохранить» после заполнения инпутов.

На детальной странице проекта при маунте компонента вызывается функция fetchById для получения конкретного проекта и записи данных в компонент Board из библиотеки Asseinfo/react-kanban. А также вызывается функция update для сохранения изменений произведенных в канбане.

Результат разработки веб-приложения «Kanban» представлено на рисунках ….

Рисунок … – Главная страница

Рисунок … – Создание нового проекта

Рисунок … – Детальная страница проекта

Рисунок … – Создание новой колонки

Рисунок … – Создание новой карточки

Рисунок … – Результат перемещения колонок и карточек между колонками

## 2.4 Описание эндпоинтов

# Заключение

В ходе выполнения курсового проекта по дисциплине «Разработка приложений в Visual Studio» мы успешно освоили основы разработки программных приложений с использованием мощной интегрированной среды разработки Visual Studio Code. Этот проект стал значимым этапом нашего обучения, позволяя нам применить полученные знания на практике и расширить наши навыки в разработке ПО.

Visual Studio Code является идеальной платформой для разработки приложений. Ее гибкость, удобство использования и широкие возможности расширения позволяют нам создавать высококачественные программные продукты на различных платформах. Мы овладели основными функциями VS Code.

Курсовой проект позволил нам не только попрактиковаться в разработке приложений, но и научиться эффективной работе в команде. Мы применяли найденные решения для решения задач, анализировали и оптимизировали наш код, следуя принципам чистого кода.

В результате нашего проекта мы создали полноценное приложение, которое демонстрирует наши навыки и достижения в разработке приложений с использованием Visual Studio Code.

Курсовой проект по дисциплине «Разработка приложений в Visual Studio» является важным шагом в нашей профессиональной карьере. Мы уверены, что благодаря этому проекту мы значительно расширили свои знания и навыки в разработке программного обеспечения. В дальнейшем мы сможем успешно применять эти навыки в реальных проектах и построить успешную карьеру в области разработки ПО.

# Список литературы

1 Прайс М. C# и .NET Core. Разработка и оптимизация / М. Прайс – СПб.: Питер, 2021. – 816 с.

2 Троелсен Э., Джепикс Ф. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. / Э. Троелсен, Ф. Джепикс.: Пер. с англ. – СПб.: ООО «Диалектика», 2018. – 1328 с.

3 Гриффитс И. Программируем на C# 8.0. Разработка приложений / И. Гриффитс. – СПб.: Питер, 2021. – 944 с.

4 Албахари Д., Албахари Б. C# 9.0. Карманный справочник / Д. Албахари, Б. Албахари / Пер. с англ. – СПб.: ООО «Диалектика», 2021. – 256 с.

5 Вандеркам Д. Эффективный TypeScript: 62 способа улучшить код / Д. Вандеркам. – СПб.: Питер, 2022. – 288 с.