

Заключение

В рамках выполнения курсовой работы была разработана и реализована многоуровневая клиент-серверная система, моделирующая взаимодействие поставщиков и клиентов. Все поставленные цели достигнуты, ключевой функционал успешно внедрен.

Основные достижения:

- Построена гибкая многоуровневая архитектура с четким разделением ответственности между слоями.
- Реализован асинхронный HTTP-сервер на основе HttpListener для обработки запросов.
- Создан консольный клиент с интуитивно понятным интерактивным интерфейсом.
- Внедрена система персональных рейтингов поставщиков.
- Разработан алгоритм формирования корзины на основе индивидуальных предпочтений пользователей.
- Организовано взаимодействие с базой данных SQLite через паттерн «Репозиторий».
- Реализован модуль обмена сообщениями между клиентами и поставщиками.

Приобретенные умения и навыки:

- Углубленное понимание принципов клиент-серверной архитектуры.
- Практический опыт работы с протоколом HTTP и проектирования REST API.
- Навыки асинхронного программирования на C# с использованием async/await.
- Применение архитектурных шаблонов, таких как Repository и Service Layer.
- Разработка пользовательского интерфейса для консольных приложений.

- Проведение отладки и тестирования распределенных систем.

Направления дальнейшего развития:

- Создание веб-интерфейса для работы с существующим API.
- Интеграция системы оформления заказов и проведения платежей.
- Внедрение push-уведомлений о новых сообщениях.
- Добавление аналитических отчетов для поставщиков.
- Масштабирование системы до кластерной архитектуры.

Разработанное приложение демонстрирует современный подход к построению бизнес-ориентированных систем и может служить основой для реальных платформ электронной коммерции.

Список используемых источников

1. Microsoft Documentation. .NET 8 Documentation –
<https://docs.microsoft.com/en-us/dotnet/>.
2. SQLite Documentation [Электронный ресурс].
– <https://www.sqlite.org/docs.html>.
3. RFC 7230-7237: HTTP/1.1 Specification –
<https://tools.ietf.org/html/rfc7230>.
4. Metanit. Учебные материалы по C# и .NET –
<https://metanit.com/sharp/>.
5. Martin Fowler. Patterns of Enterprise Application Architecture. – Addison-Wesley, 2002. – 560 с.
6. Jeffrey Richter. CLR via C#. – Microsoft Press, 2012. – 896 с.
7. Microsoft Learn. ASP.NET Core Documentation –
<https://learn.microsoft.com/en-us/aspnet/core/>.
8. Stack Overflow. Сообщество разработчиков –
<https://stackoverflow.com/>.
9. GitHub. Исходный код проектов .NET – <https://github.com/dotnet>.
10. Newtonsoft.Json Documentation – <https://www.newtonsoft.com/json>.

Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по курсу «Разработка клиент-серверных приложений»
на тему «Сетевое клиент-серверное приложение обеспечивающее
выполнение удаленного задания»

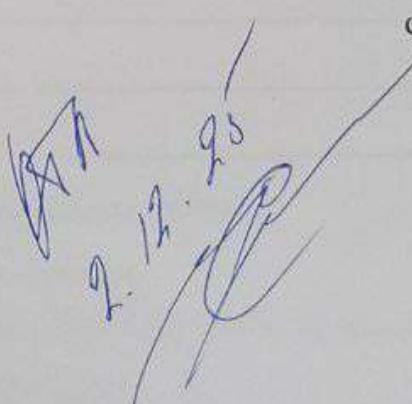
Выполнил:

студент группы 22ВВП1

Биляев Д. И.

Проверил:

к.т.н. Зинкин С. А.



A handwritten signature in blue ink, appearing to read 'Даниил Биляев' (Daniil Bilyayev), with some smaller numbers and letters written near it.

Пенза 2025

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

М.А. Митрохин

«15» 09 2025 г.

ЗАДАНИЕ

на курсовое проектирование по курсу

Разработка клиент-серверных приложений

Студенту Белиеву Дашичу Ильишу Группа 22ВВП_1

Тема проекта Сетевое клиент-серверное приложение

обеспечивающее выполнение удаленного задания

Исходные данные (технические требования) на проектирование

Сеть Microsoft TCP/IP

Стек протоколов TCP/IP

Протоколы TCP и UDP

Интерфейс WinSock2

На удаленном сервере размещена база данных

Дополнительные условия по указанию

преподавателя: Клиент-серверные системы

взаимодействие поставщиков и клиентов

Объем работы по курсу

1. Расчетная часть

Анализ граф-схемы алгоритма

Алгоритм работы

сервера

2. Графическая часть

UML-диаграммы

ГСА в Visio

3. Экспериментальная часть

Отладка сетевого приложения на сети

Отладка программного обеспечения и формирование контента сервера

Срок выполнения проекта по разделам

1 По графику

2 _____

3 _____

4 _____

5 _____

6 _____

7 _____

8 _____

Дата выдачи задания "15" _09_ 2025.

Дата защиты проекта По графику

Руководитель Зинкин С. А.

Задание получил " 15 " 09 2025 г.

Студент Риф Башеев Р. В.

Содержание

Введение	3
1. Постановка задачи	4
2. Выбор решения	5
3. Теоретическая часть задания	6
4. Описание алгоритма работы	7
5. Отладка и тестирование	9
6. Описание программы	10
7. Руководство пользователя	12
Заключение	14
Список используемых источников	16
Приложение А	17
Приложение Б	35
Приложение Б1	35
Приложение Б2	36
Приложение Б3	37
Приложение Б4	38
Приложение Б5	38
Приложение В	39

Введение

Модель распределения вычислений «клиент-сервер» является фундаментальной архитектурой при построении вычислительных систем. В рамках данной модели сервер предоставляет ресурсы и услуги, а клиенты осуществляют к ним доступ. Эта архитектура широко применяется в современной разработке для создания масштабируемых и отказоустойчивых приложений.

В рамках данной работы была разработана многоуровневая клиент-серверная система на платформе .NET 8, моделирующая взаимодействие между поставщиками и покупателями. Система поддерживает функционал управления товарами, рейтинговую систему, обмен сообщениями, а также интеллектуальное формирование корзины покупок на основе предпочтений пользователей.

Отличительными чертами проекта являются модульная архитектура, разделение на слои данных и бизнес-логики, а также реализация алгоритма формирования корзины с учетом индивидуальных оценок клиентов.