

Министерство образования и науки Республики Башкортостан
Государственное автономное профессиональное образовательное
учреждение
Уфимский колледж статистики, информатики и вычислительной техники

Отчет
по УП 11.01 Учебная практика
ПМ.11 Разработка, администрирование и защита баз данных

Руководитель проекта
_____ О.В.Фатхулова
«____» _____ 2025 г.

Студенты гр. 22П-2

_____ В.А.Бурмистров
Д.М.Казыханов
«____» _____ 2025 г.

2025

СОДЕРЖАНИЕ

лист

ВВЕДЕНИЕ	4
1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ	5
1.1 НАЗНАЧЕНИЕ РАЗРАБОТКИ	5
1.2 ТРЕБОВАНИЯ К ПРОГРАММЕ	5
1.3 ТРЕБОВАНИЯ К ФУНКЦИОНАЛЬНЫМ ХАРАКТЕРИСТИКАМ	5
1.4 ОПИСАНИЕ ВХОДНОЙ ИНФОРМАЦИИ	5
1.5 ОПИСАНИЕ ВЫХОДНОЙ ИНФОРМАЦИИ	5
2. РАЗРАБОТКА ПРОТОТИПА	7
3. СТРУКТУРА ПРОЕКТА	9
4. ОПИСАНИЕ ЗАПРОСОВ API	10
5. РЕЗУЛЬТАТ РАБОТЫ	30
ЗАКЛЮЧЕНИЕ	34

ВВЕДЕНИЕ

Современная клиент-ориентированная система требует эффективной автоматизации процессов оформления документов. Разрабатываемая информационная система призвана объединить на единой платформе клиентов и работников для оптимизации работы кафе.

Система предоставляет клиентам удобный сайт для просмотра новинок и возможность участия в балльной системе. Работники получают удобный инструмент, способный за несколько действий оформить нужный документ, добавить, редактировать или убрать новинки, а также регистрировать новых пользователей в систему.

Целью работы является проектирование и разработка базы данных и программного обеспечения для автоматизации заказов кафе.

В данном отчёте подробно рассматриваются реализованная архитектура проекта, ключевые функции готового продукта, использованные технологии, а также выводы о проделанной работе и потенциальных направления для дальнейшего развития.

1. Техническое задание

1.1 Назначение разработки

Разработка информационной системы для автоматизации процессов заказов клиентов. Цель системы, упростить работу сотрудникам и улучшить пропускную способность кафе.

1.2 Требования к программе

1.3 Требования к функциональным характеристикам

Требования к роли «Клиент»:

- Просмотр списка меню.
- Возможность авторизоваться в системе по номеру телефона.
- Возможность добавления предметов в корзину заказа и возможности оплатить полученный список.

Требования к роли «Бариста»:

- Возможность регистрировать клиента в систему по номеру телефона и ФИО.
- Возможность получения баллов клиента по номеру телефона.

Требования к роли «Менеджер»:

- Возможность добавления новых предметов в меню.
- Возможность регистрировать новых работников в систему.

1.4 Описание входной информации

Входными данными данного проекта являются: список работников, список клиентов, прайс лист меню, заказы, список заказа.

Таблица 1.4.1 – Описание входной информации

Наименование	Дата поступления	Откуда поступает
Прайс лист меню	При обновлении позиций	Из вне

1.5 Описание выходной информации

Выходная информация — это результат работы информационной системы или процесса, который может быть представлен в различных формах,

таких как документы, файлы, отчеты и т. д. Она представляет собой данные или документы, которые создаются информационной системой и передаются пользователям или другим системам для использования или дальнейшей обработки.

Выходная информация представлена в виде выходных документов. Описание выходных документов представлено в таблице 1.5.1

Таблица 1.5.1 – Описание выходных документов

Наименование документа	Периодичность выдачи	Кол-во экземпляров	Куда передаётся
Чек	После оплаты заказа	1	Клиенту
Сms сообщение с кодом для использования в бальной системе	По необходимости	1	Клиенту

2. Разработка прототипа

Веб приложение состоит из нескольких страниц:

- “Каталог”. На странице находится меню. Интерфейс страницы показан на рисунке 2.1. При нажатии на иконку пользователя, появляется окно входа.
- Внешний вид формы отображена на рисунке 2.1.

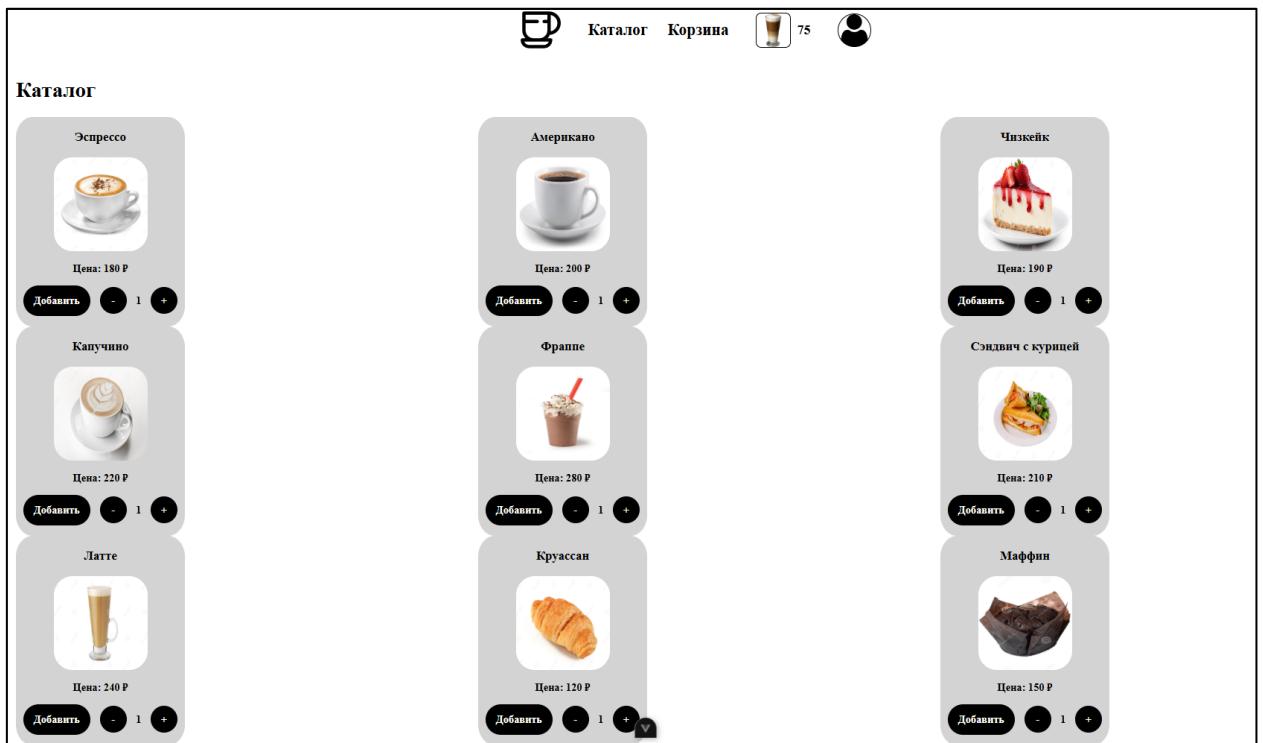


Рисунок 2.1 – Каталог меню

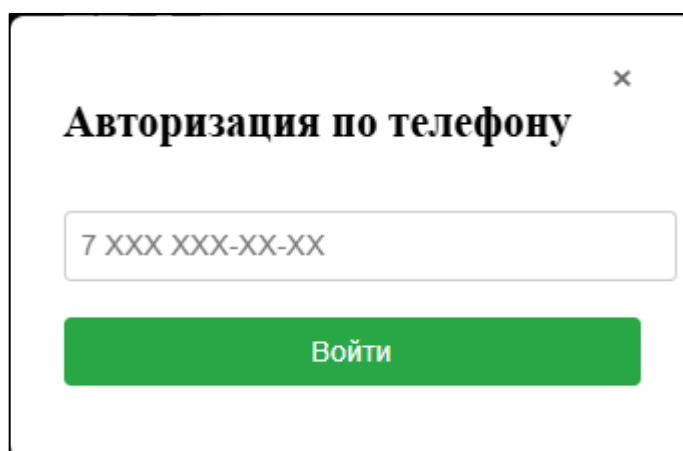


Рисунок 2.2 – Окно входа

- “Корзина”. На странице указан список клиента, с указанием предмета и его количества. В низу страницы указывается итоговая цена за все предметы.

При желании, клиент может удалить позицию из этого списка. Внешний вид страницы показан на рисунке 2.3.

Корзина

Чизкейк Кол-во: 3 Цена за шт: 190 ₽ Итого: 570 ₽ Удалить

Круассан Кол-во: 2 Цена за шт: 120 ₽ Итого: 240 ₽ Удалить

Эспрессо Кол-во: 5 Цена за шт: 180 ₽ Итого: 900 ₽ Удалить

Общая сумма: 1710 ₽

Оплатить заказ

Рисунок 2.3 – Корзина

3. Структура проекта

Проект имеет классическую клиент-серверную архитектуру, разделённую на три основных модуля: WPF-приложение, сайт и сервер.

- Сайт реализован с использованием фреймворка Vue.js. Отвечает за пользовательский интерфейс, отображения данных и взаимодействия с клиентом.
- WPF-приложение реализовано на платформе Windows и представляет собой административный клиент, представляющий интерфейс для управления системой, включая расширенные функции модерации и администрирования.
- Серверная часть построена на платформе ASP.NET Core, написанный на языке C#. Представляет собой REST API для обработки логики и работы с базой данных, обслуживая оба клиентских приложения.

Связь между клиентскими модулями и сервером осуществляется через HTTP-запросы через API. Модульная схема проекта представлена на рисунке 3.1.

Рисунок 3.1 – Модульная схема

4. Описание запросов API

Разработанное API обеспечивает полный цикл взаимодействия с системой через эндпоинты. Взаимодействие построено вокруг основных сущностей: Пользователей и заказов. Основные контроллеры описаны в таблицах 4.1-4.3.

Таблица 4.1 – Описание запросов API контроллера администрирования

AdminController				
Метод	Curl (входные данные)	URL (Строка запроса)	Response body (возвращаемый результат)	Код результата
Login	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Admin/login' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "name": "Давид", "role": "Бариста", "password": "Canlet30" }'</pre>	https://localhost:7088/api/Admin/login	{ "name": "Давид", "role": "Бариста", "password": "Canlet30" }	200
AddWorker	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Admin/addWorker' \</pre>	https://localhost:7088/	{ "workerId": 32, }	200

	<pre>-H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "workerName": "Давид", "workerRole": "Бариста", "workerPassword": "Canlet31" }'</pre>	api/Admin/addWorker	<pre>"workerName": "Давид", "workerRole": "Бариста", "workerPassword": "\$2a\$11\$oqnhvDgLMq0RsJA0rFHmZu0fhIO/zfFPb8Tindcq3m0sKJZAHBiGW", "orders": []}</pre>	
--	---	---------------------	---	--

Продолжение таблицы 4.1

1	2	3	4	5
AddItem	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Admin/addItem' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "itemId": 0, "itemValue": 100, "iitemName": "Курник", "itemType": "Закуска",</pre>	https://localhost:7088/api/Admin/addItem	<pre>{ "itemId": 13, "itemValue": 100, "iitemName": "Курник", "itemType": "Закуска", "itemPrice": 50, "itemImagePath": "/Resources/images/Maffin.jpg", "orderLists": []}</pre>	200

	<pre> "itemPrice": 50, "itemImagePath": "/Resources/images/Maffin.jpg" } </pre>			
UpdateItem	<pre> curl -X 'POST' \ 'https://localhost:7088/api/Admin/updateItem' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "itemId": 13, "itemValue": 100, "iitemName": "Хотдог", "itemType": "Закуска", "itemPrice": 130, "itemImagePath": "/Resources/images/Maffin.jpg"}' </pre>	<pre> https://localhost:7088/api/Admin/upda teItem </pre>	<pre> { "itemId": 13, "itemValue": 100, "iitemName": "Хотдог", "itemType": "Закуска", "itemPrice": 130, "itemImagePath": "/Resources/images/Maffin.jpg", "orderLists": [] } </pre>	200

Продолжение таблицы 4.1

1	2	3	4	5
---	---	---	---	---

UpdateWorker	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Admin/updateWorker' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "workerId": 32, "workerName": "Донни", "workerRole": "Бариста" }'</pre>	https://localhost:7088/api/Admin/updateWorker	<pre>{ "workerId": 32, "workerName": "Донни", "workerRole": "Бариста", "workerPassword": null, "orders": [] }</pre>	200
AddCustomer	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Admin/addCustomer' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "name": "Усман", "phone": "834595431853", "points": 0 }'</pre>	https://localhost:7088/api/Admin/addCustomer	<pre>{ "customerId": 19, "customerPhone": "834595431853", "customerName": "Усман", "customerPoints": 0, "orders": [] }</pre>	200

Продолжение таблицы 4.1

1	2	3	4	5
GetAllCustomers	<pre>curl -X 'GET' \ 'https://localhost:7088/api/Admin/CustomersAll' \ -H 'accept: */*'</pre>	<pre>https://localhost:7088/api/Admin/CustomersAll</pre>	<pre>[{ "customerId": 1, "customerPhone": "89123456789", "customerName": "Алексей", "customerPoints": 150, "orders": [] }]</pre>	200
GetAllWorkers	<pre>curl -X 'GET' \ 'https://localhost:7088/api/Admin/WorkersAll' \ -H 'accept: */*'</pre>	<pre>https://localhost:7088/api/Admin/WorkersAll</pre>	<pre>[{ "workerId": 1, "workerName": "Ольга", "workerRole": "Менеджер", }]</pre>	200

		<pre>"workerPassword": "\$2a\$11\$TI3fKQML3Aoa062jd SC.t.P1tuG9qziI4B6llaU7tuhYS g8zcZld.", "orders": [] }</pre>	
--	--	--	--

Продолжение таблицы 4.1

1	2	3	4	5
GetAllOrder s	<pre>curl -X 'GET' \ 'https://localhost:7088/api/Admin/OrdersAll' \ -H 'accept: */*' </pre>	https://localhost:70 88/ api/Admin/Orders All	[{ "orderId": 1, "orderDate": "2024-01- 15T00:00:00", "workerId": 2, "customerId": 1, "customer": null, "orderLists": [], "worker": null }]	200
DeleteCusto mer	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Admin/deleteCustomer' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{' </pre>	https://localhost:70 88/ api/Admin/deleteC ustomer	{ "customerId": 19, "customerPhone": "834595431853", "customerName": "Усман", }	200

	<pre> "customerId": 19, "customerPhone": "834595431853", "customerName": "Усман", "customerPoints": 0 }'</pre>		<pre> "customerPoints": 0, "orders": [] }</pre>	
--	---	--	--	--

Продолжение таблицы 4.1

1	2	3	4	5
DeleteWorker	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Admin/deleteWorker' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "workerId": 32, "workerName": "Донни", "workerRole": "Бариста" }'</pre>	https://localhost:7088/api/Admin/deleteWorker	{ "workerId": 32, "workerName": "Донни", "workerRole": "Бариста", "workerPassword": null, "orders": [] }	200

Таблица 4.2 – Описание запросов API сервисного контроллера

ServiceController				
Метод	Curl (входные данные)	URL (Строка запроса)	Response body (возвращаемый результат)	Код результата
Items	<pre>curl -X 'GET' \ 'https://localhost:7088/api/Service/items' \ -H 'accept: */*' </pre>	https://localhost:70 88/api/Service/item s	[{ "itemId": 0, "itemValue": 100, "itemName": "Краска", "itemType": "Закуска", "itemPrice": 50, "itemImagePath": "/Resources/images/Maffin.jpg" , "orderLists": [] }	200

]	
--	--	--	---	--

Продолжение таблицы 4.2

1	2	3	4	5
GetCategoryItems	<pre>curl -X 'GET' \ 'https://localhost:7088/api/Service/category_items?category=%D0%97%D0%B0%D0%BA%D1%83%D1%81%D0%BA%D0%B0' \ -H 'accept: */*' </pre>	<pre>https://localhost:7088/api/Service/category_items?category=%D0%97%D0%B0%D0%BA%D1%83%D1%81%D0%BA%D0%B0</pre>	<pre>[{ "itemId": 0, "itemValue": 100, "itemName": "Краса", "itemType": "Закуска", "itemPrice": 50, "itemImagePath": "/Resources/images/Maffin.jpg", "orderLists": [] }]</pre>	200

Продолжение таблицы 4.2

1	2	3	4	5
OrderAdd	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Service/order_add' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "idCustomer": 5, "idWorker": 16, "orderDate": "2025-10-27T05:01:06.173Z", "items": [{ "id": 7, "value": 2 }, { "id": 8, "value": 10 }], "point": 8 }'</pre>	<pre>https://localhost:7088/api/Service/order_add</pre>	<pre>{ "idCustomer": 5, "idWorker": 16, "orderDate": "2025-10-27T05:01:06.173Z", "items": [{ "id": 7, "value": 2 }, { "id": 8, "value": 10 }], "point": 8 }</pre>	200

Продолжение таблицы 4.2

1	2	3	4	5
ddItemKorz	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Service/AddItemKorz' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "idOrder": 2, "idCustomer": 3, "idItem": 2, "countItem": 4 }'</pre>	https://localhost:7088/api/Service/AddItemKorz	Предмет добавлен	200

Продолжение таблицы 4.2

ReduceItemKor z	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Service/ReduceItemKorz' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "idOrder": 2, "idCustomer": 3, "idItem": 2, "countItem": 4 }'</pre>	https://localhost:7088/api/Service/ReduceItemKorz	2	200
--------------------	--	---	---	-----

Продолжение таблицы 4.2

1	2	3	4	5
DeleteItemKorz	<pre>curl -X 'POST' \ 'https://localhost:7088/api/Service/DeleteItemKorz' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "idOrder": 2, "idCustomer": 3, "idItem": 2, "countItem": 4 }'</pre>	https://localhost:70 88/api/Service/Del eteItemKorz	2	200

Таблица 4.3 – Описание запросов API контроллера клиентов

UserController				
Метод	Curl (входные данные)	URL (Строка запроса)	Response body (возвращаемый результат)	Код результата
deleteCustomer	<pre>curl -X 'POST' \ 'https://localhost:7088/api/User/deleteCustomer' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "customerId": 20, "customerPhone": "string", "customerName": "string", "customerPoints": 0 }'</pre>	https://localhost:7088/api/User/deleteCustomer	{ "customerId": 20, "customerPhone": "string", "customerName": "string", "customerPoints": 0, "orders": [] }	200

Продолжение таблицы 4.3

1	2	3	4	5
Login	<pre>curl -X 'POST' \ 'https://localhost:7088/api/User/login' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '"898658433234"</pre>	https://localhost:70 88/api/User/login	{ "name": "string", "phone": "898658433234", "points": 0 }	200
ProfileUpdate	<pre>curl -X 'POST' \ 'https://localhost:7088/api/User/profile_update' \ -H 'accept: */*' \ -H 'Content-Type: application/json' \ -d '{ "name": "string", "phone": "65464565464", "points": 0 }'</pre>	https://localhost:70 88/api/User/profile _update	{ "customerId": 0, "customerPhone": "65464565464", "customerName": "string", "customerPoints": 0 }	200

5. Результат работы

Введение

Данное руководство содержит подробные инструкции и информацию, необходимую для самостоятельного и эффективного использования веб-сайта и мин приложения “Чай-не кофе”.

Чай-не кофе – это информационная система для автоматизации ведения записей о заказах и посетителях.

Начало работы

Для работы с системой сперва надо убедится, что запущено серверное приложение по адресу `http://localhost:7088/swagger`. После успешного запуска API, запустите клиентское `Vue.js` приложение по адресу <http://localhost:5173/>

Руководство пользователя для клиента

Главная страница и навигация

При первом посещении сайта пользователь увидит каталог предметов, доступных для заказа в кафе. Страница сайта представлена на рисунке 3.1

В верхней части страницы находится навигация по сайту с кнопками «Каталог» и «Корзина». В самой правой части панели навигации, находится иконка. Нажав на неё, открывается окно авторизации по номеру телефона. После успешной авторизации, появится уведомление, приветствующее пользователя и в панели навигации будет отображено количество его баллов.

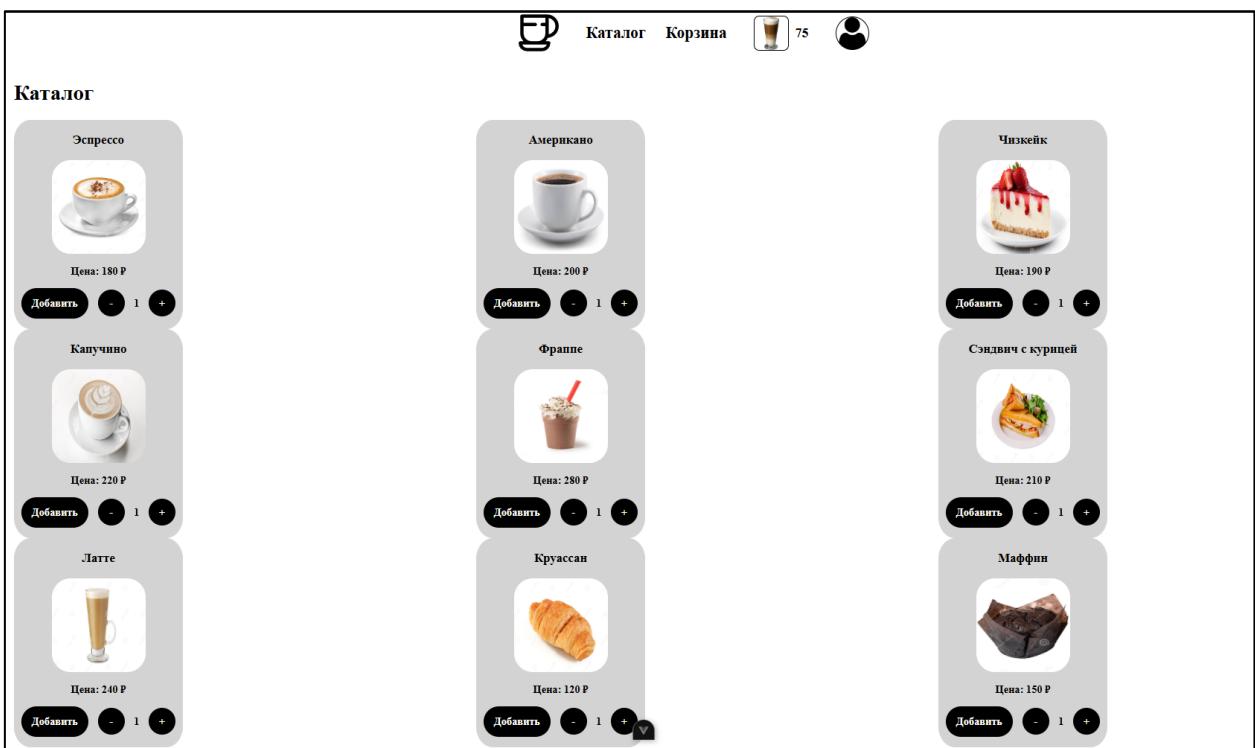


Рисунок 5.1 – Страница каталога

Регистрация и вход в систему

Чтобы получить возможность входа в систему, клиенту необходимо зарегистрировать себя через работника кафе. Для этого, клиенту надо сказать только своё Имя и номер телефона. Форма авторизации в систему показана на рисунке 3.2

1. Нажать на иконку профиля.
2. В открывшемся окне введите свой номер телефона.

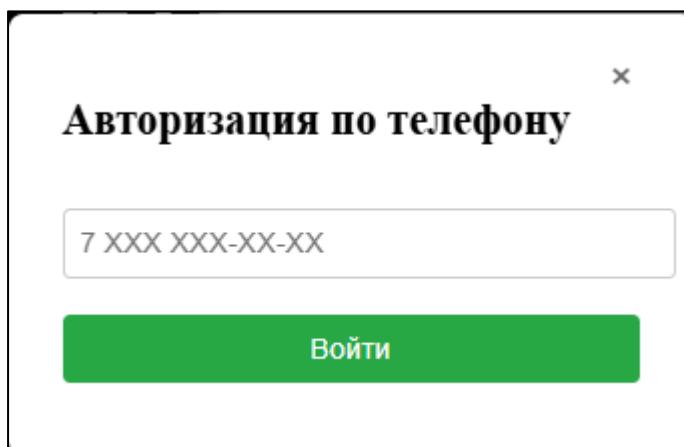


Рисунок 5.2 – Окно авторизации

Каталог и добавление еды в корзину

Для добавления желаемого предмета меню, нажмите на кнопку «Добавить» в карточке предмета. При желании можно нажать на кнопки «увеличения/уменьшения» рядом, для добавления большего количества данного предмета в корзину.



Рисунок 5.3 – Карточка товара

Корзина

Страница корзины представляет все выбранные предметы. Здесь пишется количество предметов, цена за штуку товара, и итоговая цена товара, являющийся произведением цены на количество предметов. В самой правой части находится кнопка удаления товара из корзины.

В низу страницы пишется итоговая цена за все предметы в корзине, и кнопка «оплатить заказ». После оплаты заказа, пользователь увидит уведомление об успешном оплате заказа.

Корзина

Чизкейк Кол-во: 1 Цена за шт: 190 ₽ Итого: 190 ₽ [Удалить](#)

Фраппе Кол-во: 3 Цена за шт: 280 ₽ Итого: 840 ₽ [Удалить](#)

Общая сумма: 1030 ₽

[Оплатить заказ](#)

Рисунок 5.4 – Страница корзины

ЗАКЛЮЧЕНИЕ

В ходе учебной практики была успешно выполнена задача по разработке полнофункционального веб-приложения “Чай-не кофе”, предназначенная для облегчения работоспособности работников кафе. Все цели и задачи, поставленные в техническом задании, были достигнуты.

В результате проделанной работы:

– разработан

Практическая значимость проекта заключается в облегчении работы сотрудникам кафе, а также возможность подключения системы баллов для клиентов. Таким образом, в рамках учебной практики были закреплены и применены на практике значения по веб-разработке, проектированию баз данных, клиент-серверному взаимодействию и проектной деятельности. Проект “Чай-не кофе” считается завершённым и готовым к дальнейшему развитию. В качестве перспективы развития платформы можно выделить интеграцию с платёжными системами, разработку мобильного приложения для более облегчённого доступа к приложению.