|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКО ФЕДЕРАЦИИ |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  (ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)») |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |  |
| --- | --- |
|  | ДОПУСТИТЬ К ЗАЩИТЕ |
|  | Заместитель директора по учебно-методической работе |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Вернер |
|  | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |
|  | ДОПУСКАЕТСЯ К ЗАЩИТЕ |
|  | Председатель ПЦК специальности 09.02.03 Программирование в компьютерных системах |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.И. Глускер |
|  | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

|  |
| --- |
| **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА** |
| **(ДИПЛОМНЫЙ ПРОЕКТ)** |
|  |
| на тему: Разработка веб-сайта кафе |
|  |
| студента группы 090203-9о-17/2  специальности 09.02.03 Программирование в компьютерных системах |
| Тешаева Рустама Мирзодавлатовича |

|  |  |  |
| --- | --- | --- |
| Студент |  | Р.М. Тешаев |
| Руководитель |  | Е.М. Дворникова |
| Консультанты: |  |  |
| Нормоконтроль |  | Е.А. Чеснокова |

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2021 г. | | |
| Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Председатель ГЭК |  | О.И. Белов |

Москва

2021

**СОДЕРЖАНИЕ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Разраб.

Р.М. Тешаев

Провер.

Е.М. Дворникова

Н. Контр.

Е.А. Чеснокова

Утверд.

.

Разработка веб-сайта кафе

Лит.

Листов

46

УКИТ

ВВЕДЕНИЕ 4

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 7

1.1 Анализ предметной области 7

1.1.1 Описание терминологии 9

1.2 Анализ и выбор инструментальных средств 11

2 ПРАКТИЧЕСКАЯ ЧАСТЬ 14

1.3 Требования к программному изделию 14

1.3.1 Наименование программного изделия 14

2.1.2 Эксплуатационное назначение 14

2.1.3 Условия эксплуатации 14

2.1.4 Требования к функциональным характеристикам 14

2.1.5 Требования к составу и параметрам технических средств 15

2.2 Проектирование программного изделия 15

2.2.1 Архитектура программного изделия 15

2.2.2 Диаграмма вариантов использования 17

2.2.3 Классическая концепция веб-сайта 17

2.2.4 Логическая модель базы данных 19

2.2.5 Нормализация таблиц 19

2.2.6 Описание таблиц 19

2.2.8 Другие объекты базы данных 21

2.3 Разработка программного изделия 21

2.3.1 Стадии разработки 21

2.3.1 Шаблон проектирования MVC 22

2.3.2 Интерфейс приложения 23

ДП

2.3.3 Защита программного изделия 24

2.4 Тестирование программного изделия 24

ЗАКЛЮЧЕНИЕ 29

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30

ПРИЛОЖЕНИЕ A 32

ПРИЛОЖЕНИЕ Б 34

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

# ВВЕДЕНИЕ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

4

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Данная выпускная квалификационная работа посвящена разработке веб-сайта кафе для продвижения и дальнейшего роста, данный сайт поможет начинающему предприятию выйти на новый уровень.

Актуальность темы заключается в создании веб-сайта для начинающего кафе, чтобы пользователи могли заказывать еду на дом, были знакомы с данным предприятием и следили за актуальной информацией на веб-сайте. Все это нужно для расширения базы клиентов и для дальнейшего роста, потому что чем больше доход тем лучше для заведения. Когда-то такого понятия, как «доставка еды» в принципе не было, людям приходилось посещать очно подобные заведения, потом были просто частные курьеры, но их работа не была привязана к какому-либо ПО, сейчас же 21 век, это век технологий, возможность доставки еды, должна быть у каждого начинающего предприятия подобного рода.

Объектом ВКР является изучение рынка заведений общественного питания, конкретно в данной теме, это кафе предприятия.

Предметом ВКР является процесс создания веб-сайта для роста предприятия и продвижения.

Цель проекта – разработать веб сайт кафе.

При разработке дипломного проекта были поставлены следующие задачи:

* проанализировать предметную область,
* спроектировать базу данных,
* создать базу данных,
* разработать веб-сайт,
* протестировать веб-сайт.

Методы исследования, которые использовались при разработке дипломного проекта:

* анализ,
* сравнение.

Круг рассматриваемых проблем:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

5

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

* изучение рынка кафе,
* способы разработки веб-сайтов,
* выбор оптимального предприятия.

Структура дипломного проекта:

* титульный лист,
* содержание,
* введение,
* теоретическая часть,
* практическая часть,
* заключение,
* список использованных источников,
* приложения.

Содержание включает в себя названия разделов и номера страниц этих разделов.

В теоретическую часть входят теоретические аспекты изучаемого объекта и предмета дипломной работы, обзор инструментальных средств.

В практическую часть входят описание разработки программного изделия, использованных методов разработки и паттернов проектирования, описание примененных методов тестирования, а также требования к программному изделию, описание и анализ применяем используемых библиотек, фреймворков, сервисов, описание и обоснование модели базы данных.

В заключении подводится итог проделанной работы. Раздел демонстрирует выводы о выполнении целей и задач, указанных в введении, и трудностях, возникших при выполнении работы.

В списке используемых источников приводится литература, которая использовалась при разработке приложения.

С практической точки рассмотрения и экономической веб-сайт оправдает себя повышением активности текущих клиентов и привлечет новых. Данный сайт будет понятен интуитивно любому пользователю и будет работать эффективно.

Кафе − это заведение общественного питания и отдыха, похожее на небольшой ресторан, но с ограниченным по сравнению с рестораном ассортиментом продукции, также, возможно, − с самообслуживанием.

Как следствие эффективность использования веб-сайта играет важную роль при создании любого бизнеса. Эффективность использования сайта выражается в различных факторах:

* возможность доставки товаров,
* актуальные новости об интересующих пользователей товаров,
* быстрый поиск.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

6

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

# 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

7

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

## Анализ предметной области

Дипломный проект посвящен разработке веб-сайта кафе для продвижения и дальнейшего роста, данный сайт поможет начинающему предприятию выйти на новый уровень. Постановка задачи: разработать веб-сайт кафе, чтобы пользователь видел на сайте каталоги еды, названия и описания блюд, контактные данные. На сайте должен быть реализован поиск, корзина, оформление заказа, обработка заказа. Это основная часть того что пользователь может сделать на сайте. Администратор должен хранить данные о еде в соответствующей БД, у него должна быть возможность просматривать, изменять, добавлять и удалять данные. Все это нужно для продвижения и дальнейшего роста начинающего заведения.

Программное изделие действительно является уникальным, по причине того, что в сети Интернет нет веб-сайта с таким же набором функций для начинающего предприятия общественного питания. Для сравнения веб-сайт https://buharacafe.ru/ там тоже можно посмотреть меню, описание блюд, можно оформить доставку еды, но есть один нюанс, это сайт-визитка, сайт-визитка подразумевает просто ознакомление с предприятием и с его услугами, на этом все, то есть пользователь не сможет заказать еду непосредственно через сам веб-сайт, он может это сделать, позвонив на горячую линию, которая указана в контактных данных, по-другому никак. Конкретно в данной дипломной работе ВКР, пользователь сможет заказать еду через сайт, то есть ему необязательно звонить на горячую линию, чтобы оформить заказ, он может это сделать сам. Ниже приведена главная страница данного веб-сайта (рисунок 1).

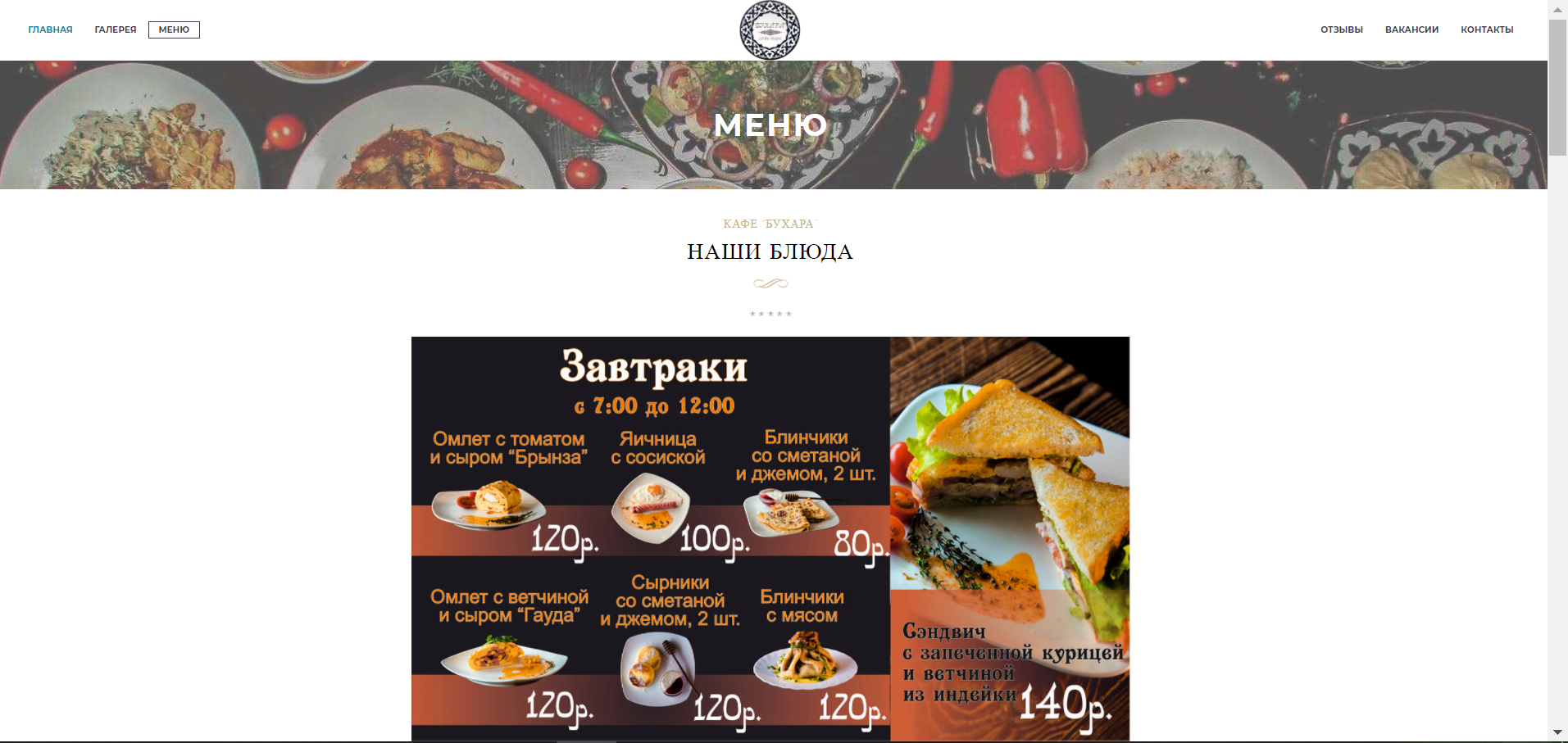


Рисунок 1 – Главная страница Кафе Бухара

Изм.

Лист

№ докум.

Подпись

Дата

Лист

8

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

В качестве второго примера можно рассмотреть https://kafe-mangal.ru/, здесь уже более обширный функционал, тут можно оформить доставку еды через сайт непосредственно, только здесь это делается медленнее чем в дипломном проекте, то есть сначала пользователю нужно зарегистрироваться, потом авторизоваться, это занимает какое-то время и не все любят тратить на это время. А дипломная работа на ВКР, подразумевает, что пользователь зашел на сайт, выбрал понравившийся товар и оформил доставку без авторизации/регистрации, пользователь не будет тратить время на это, у него будет возможность оформить заказ без учета авторизации/регистрации. Два фактора, которые отличают этот веб-сайт от всех остальных, в сети Интернет нет подобного сайта, именно с таким подходом. Ниже расположена главная страница данного веб-сайта (рисунок 2)

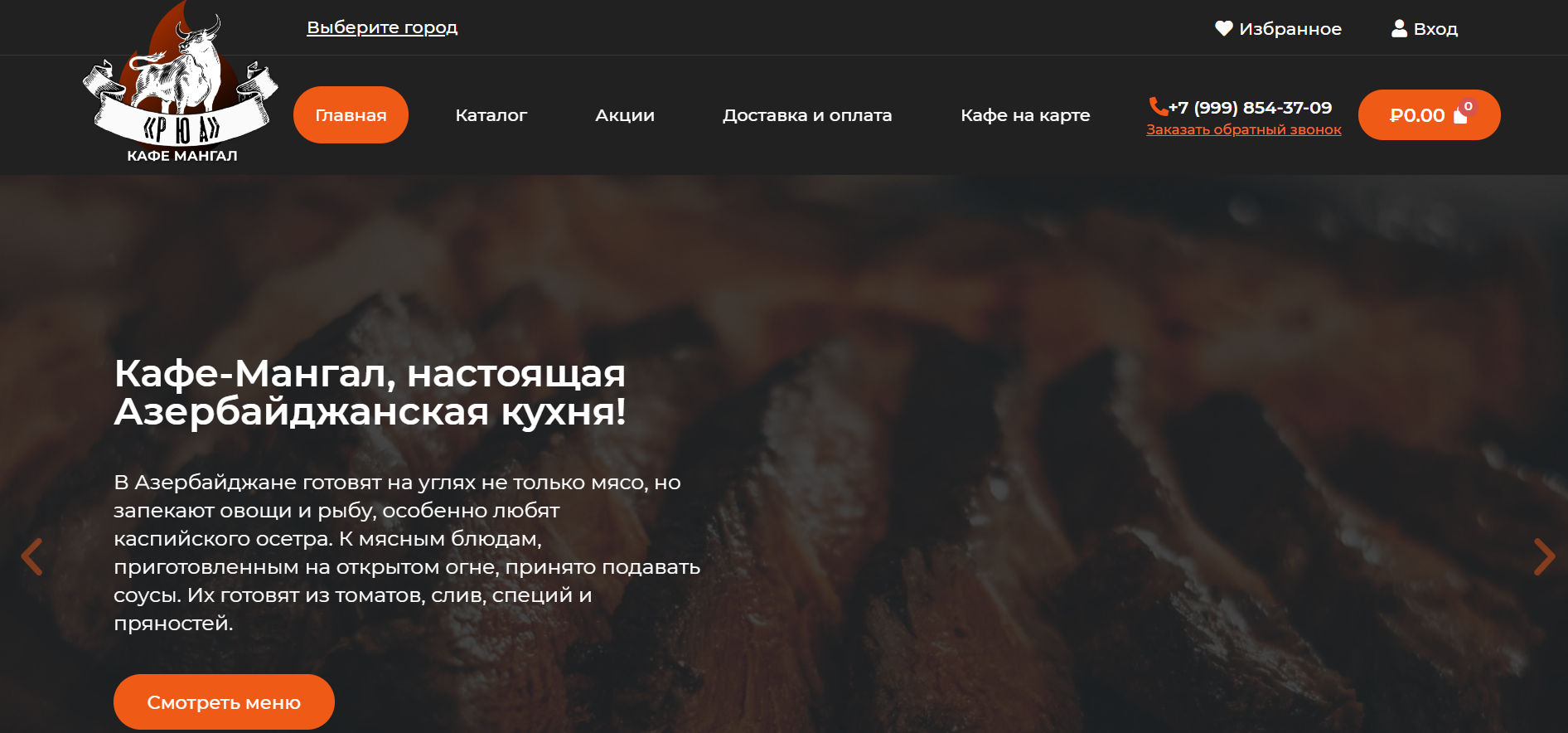


Рисунок 2 – Главная страница Кафе Мангал

Актуальность программного изделия заключается в том, что с каждым годом количество людей которые посещают подобные предприятия растет, как и само количество таких предприятий, но далеко не все предприятия успешны в финансовом плане, большинство из них разваливаются, потому что нет правильной бизнес-модели, слишком устаревшие подходы к этому бизнесу. Создание веб-сайта для подобного предприятия, это отличное решение для кафе, это нужно для развития предприятия и хороших продаж.

## Описание терминологии

Изм.

Лист

№ докум.

Подпись

Дата

Лист

9

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Программное изделие (ПИ) представляет собой программу, являющуюся продуктом промышленного производства.

Автоматизация – процесс замены ручного труда человека машинным. Целью автоматизации является увеличение эффективности за счет автоматизации однотипных, рутинных процессов.

Автоматизированная информационная система (АИС) – это совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией.

Язык программирования − это набор формальных правил, по которым пишут программы. Обычный язык нужен для общения людей, а язык программирования − для общения с компьютером.

База данных − это место для хранения данных. Используется в том числе в клиент-серверной архитектуре.

Система управления базами данных (СУБД) – это комлекс программно-языковых средств, позволяющих создать базы данных и управлять данными. Иными словами, СУБД – это набор программ, позволяющий организовывать, контролировать и администрировать базы данных.

Жизненный цикл базы данных (ЖЦБД) – это процесс проектирования, реализации и поддержки базы данных.

Трехзвенная архитектура – архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиента, сервера приложений и сервера баз данных.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

10

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Клиентский слой – интерфейс пользователя. Это может быть браузер, которому отправляются HTML-страницы. Главное, чтобы с его помощью пользователь мог отправлять запросы на сервер и обрабатывать его ответы.

Слой логики − сервер, на котором происходит обработка запросов/ответов. Часто его еще называют серверным слоем. Также здесь происходят все логические операции: математические расчеты, операции с данными, обращения к другим сервисам или хранилищам данных.

Слой данных – сервер баз данных: к нему обращается сервер. В этом слое сохраняется вся необходимая информация, которой пользуется приложение при работе.

Веб-интерфейс – это Веб-страница или совокупность веб-страниц, предоставляющая пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера. Веб-интерфейсы получили широкое распространение в связи с ростом популярности всемирной паутины и соответственно - повсеместного распространения веб-браузеров.

MVC описывает простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, приложение легче масштабируется, тестируется, сопровождается и реализуется. В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Модель − содержит бизнес-логику приложения и включает методы выборки, обработки и предоставления конкретных данных, что зачастую делает ее очень толстой, что вполне нормально.

Вид − используется для задания внешнего отображения данных, полученных из контроллера и модели.

Контроллер − связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение. Контроллер отвечает за обработку запросов пользователя. Контроллер не должен содержать SQL-запросов. Их лучше держать в моделях. Контроллер не должен содержать HTML и другой разметки, это лучше выносить в виды.

## Анализ и выбор инструментальных средств

Изм.

Лист

№ докум.

Подпись

Дата

Лист

11

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

В качестве СУБД для хранения всей информации был выбран Microsoft SQL Server.

Особенности:

* независимость от конкретных СУБД,
* переносимость с одной вычислительной системы на другую,
* наличие стандартов,
* реляционная основа,
* высокоуровневая структура.

Недостатки:

* допущение строк-дубликатов в таблицах и результатах выборок, что в рамках реляционной модели данных невозможно и недопустимо,
* поддержка неопределённых значений NULL, создающую фактически многозначную логику.

В качестве среды разработки был выбран Visual Studio, по причине ряда преимуществ:

* встроенный тестировщик и отладчик,
* работа с Git репозиторием,
* привычное использование.

Для разработки приложения был выбран язык программирования С# из-за ряда преимуществ:

* безопасный код,
* опыт работы с ним.

В качестве фреймворка был выбран ASP.NET MVC Framework, специальный фреймворк для создания веб-приложений, по причине ряда преимуществ:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

12

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

* кроссплатформенная поддержка,
* открытый исходный код,
* опыт работы с ним.

ASP.NET MVC Framework – этофреймворк для создания веб-приложений, который реализует шаблон Model−view-controller. В апреле 2009 года исходный код ASP.NET MVC был опубликован под лицензией Microsoft Public License.

В качестве среды разработки был выбран Visual Studio, по причине ряда преимуществ:

* встроенный тестировщик и отладчик,
* работа с Git репозиторием,
* привычное использование.

Хранилище GitHub выбрано в качестве контроля версий, по причине ряда преимуществ:

* опыт работы с ним,
* приятный интерфейс.

Для подключения приложения к базе данных использовался Entity Framework Core. Entity Framework Core – это технология для доступа к базам данных от Microsoft. Она позволяет взаимодействовать с СУБД с помощью сущностей (entity), то есть классов и объектов NET, а не таблиц базы данных. Это самый известный и функциональный ORM – инструмент в C#. ORM − это object-relational mapping − отображение данных на реальные объекты.

Библиотеки, которые использовались в проекте:

* JQuery − это кроссплатформенная и удобная JavaScript-библиотека, предназначенная для упрощения клиентского сценария HTML. JQuery − самая популярная в настоящее время библиотека JavaScript, которая устанавливается на 65% из 10 миллионов самых посещаемых сайтов в Интернете,
* Modernizr − библиотека JavaScript, которая определяет функции, доступные в браузере пользователя,
* Razor − это синтаксис разметки для внедрения серверного кода на веб-страницы. Razor синтаксис состоит из Razor разметки, C# и HTML. Файлы, содержащие Razor обычно, имеют расширение . cshtml,
* Bootstrap − свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения,
* Ninject − это мощный и одновременно легкий в использовании IoC контейнер. По простоте использования, а также понятности api этот контейнер чем-то напоминает Autofac,

Изм.

Лист

№ докум.

Подпись

Дата

Лист

13

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

* Moq – это простой и легковесный изоляционный фреймоврк (Isolation Framework), который построен на основе анонимных методов и деревьев выражений.

Стоит обратить внимание на IIS Express. IIS Express − это веб-сервер, созданный компанией Microsoft для использования в среде Windows. Простыми словами − это сервер, при помощи которого можно запускать сайты на компьютере.

# ПРАКТИЧЕСКАЯ ЧАСТЬ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

14

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

## 2.1 Требования к программному изделию

В данном разделе описана общая информация о программном изделии. Наименование программного изделия, эксплуатационное назначение, средства проектирования, архитектура приложения.

## Наименование программного изделия

Веб-сайт кафе.

## 2.1.2 Эксплуатационное назначение

Программа предназначена для использования неограниченным кругом лиц.

## Условия эксплуатации

Минимальное количество персонала – один человек, владеющий русским языком, обладающий практическим навыком по использованию веб- браузера.

## Требования к функциональным характеристикам

Приложение должно обеспечивать выполнение следующих функций:

* переход между страницами на сайте,
* поиск блюд,
* добавление товара в корзину,
* оформление заказа,
* адаптивный дизайн,
* CRUD для администратора,
* обеспечение защиты сайта.

## 2.1.5 Требования к составу и параметрам технических средств

Рекомендуемые системные требования к компьютеру:

* процессор Intel Core i3 и остальные процессоры,
* оперативная память не менее 80мб,
* HDD или SSD-диск объемом не менее 500мб,
* клавиатуру,
* монитор с разрешением не менее 1366x768 пикселей.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

15

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

На компьютере должен быть установлены:

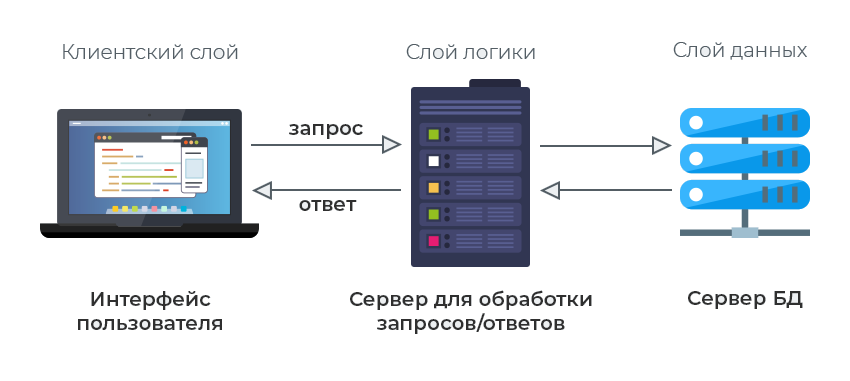
* Microsoft SQL Server,
* Visual Studio,
* IIS Express,
* копия операционный системы Windows (желательно Windows 10),
* поддержка NET.Framework,
* любой веб-браузер.

## Проектирование программного изделия

В данном разделе описаны концепция и архитектура программного изделия, использованные методы и алгоритмы при разработке, а также диаграммы UML.

## 2.2.1 Архитектура программного изделия

Архитектура информационной системы – трехзвенная. Трехзвенная архитектур − архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиента, сервера приложений и сервера баз данных (рисунок 3).



Изм.

Лист

№ докум.

Подпись

Дата

Лист

16

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Рисунок 3 – Трехзвенная архитектура

Преимущества:

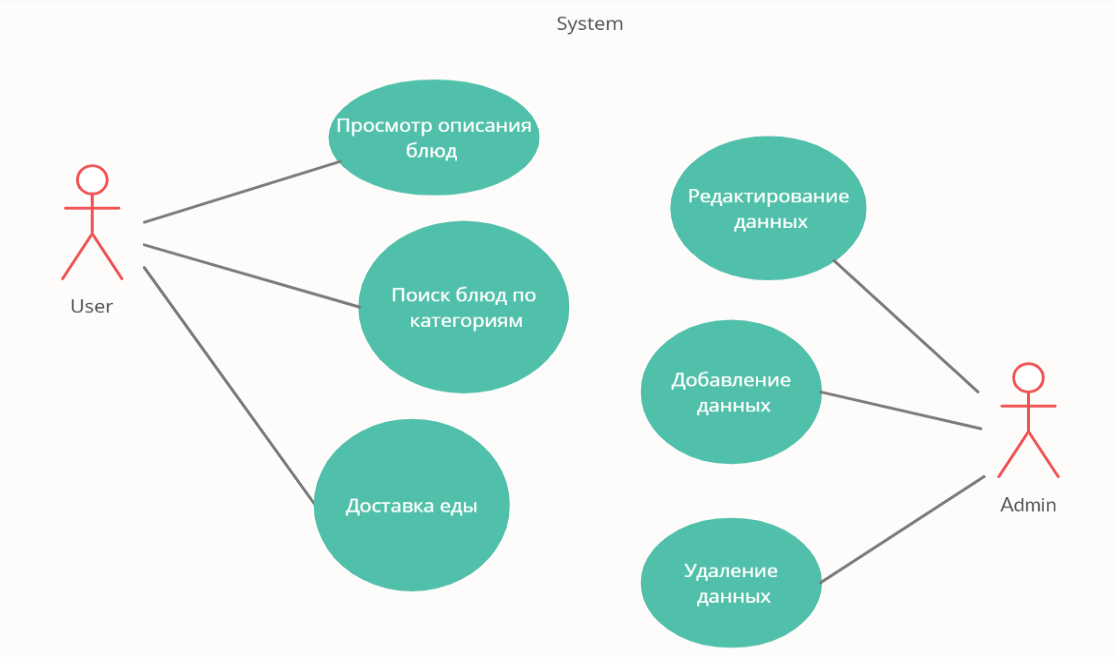
* возможность модифицировать данные перед отправкой клиенту,
* возможность расширить наше приложение на несколько серверов,
* высокая надежность,
* низкие требования к скорости канала между терминалом и сервером приложений,
* низкие требования к производительности и техническим характеристикам терминалов, как следствие снижение их стоимости.

Недостатки:

* более высокая сложность создания приложений,
* сложнее в разворачивании и администрировании,
* высокие требования к производительности серверов приложений и сервера базы данных, и высокая стоимость серверного оборудования,
* высокие требования к скорости канала между сервером базы данных и серверами приложений.

## 2.2.2 Диаграмма вариантов использования

На данной диаграмме показаны две роли. Роль пользователя и роль администратора. Ниже приведена диаграмма деятельности (рисунок 4).



Изм.

Лист

№ докум.

Подпись

Дата

Лист

17

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Рисунок 4 – Диаграмма вариантов использования

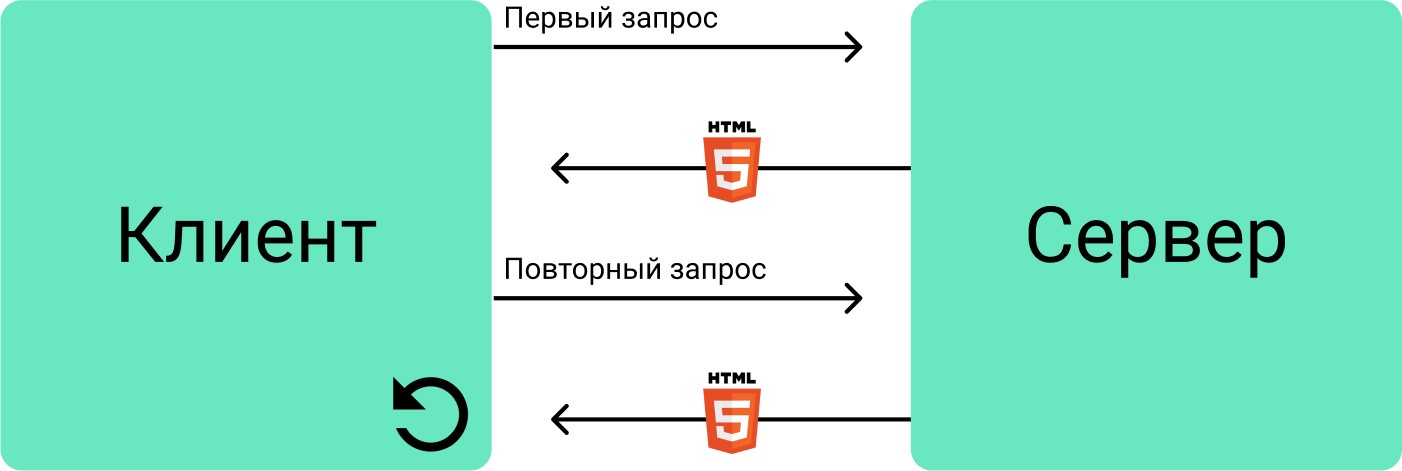
## Классическая концепция веб-сайта

Программное изделие использует классическую концепцию веб-сайта. При классической концепции каждая страница отправляет запрос на сервер и полностью обновляет все данные.

Основные элементы данной концепции:

* Backend API. Предназначен получения данных из базы данных,
* роутинг. Обеспечивает навигацию между страницами веб-сайта,
* шаблонизатор. Предназначен для повторного использования модулей веб-сайта.

Ниже приведена классическая концепция веб-сайта (рисунок 5).



Изм.

Лист

№ докум.

Подпись

Дата

Лист

18

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Рисунок 5 – Классическая концепция веб-сайта

Стоит обратить внимание на концепцию одностраничного приложения. Данная концепция пришла на замену классической концепции. Одностраничные приложения позволяют имитировать работу десктоп приложений. Архитектура устроена таким образом, что при переходе на новую страницу, обновляется только часть контента. С учетом всего перечисленного, нет необходимости повторно загружать одни и те же элементы.

Ниже приведена концепция одностраничного приложения (рисунок 6).

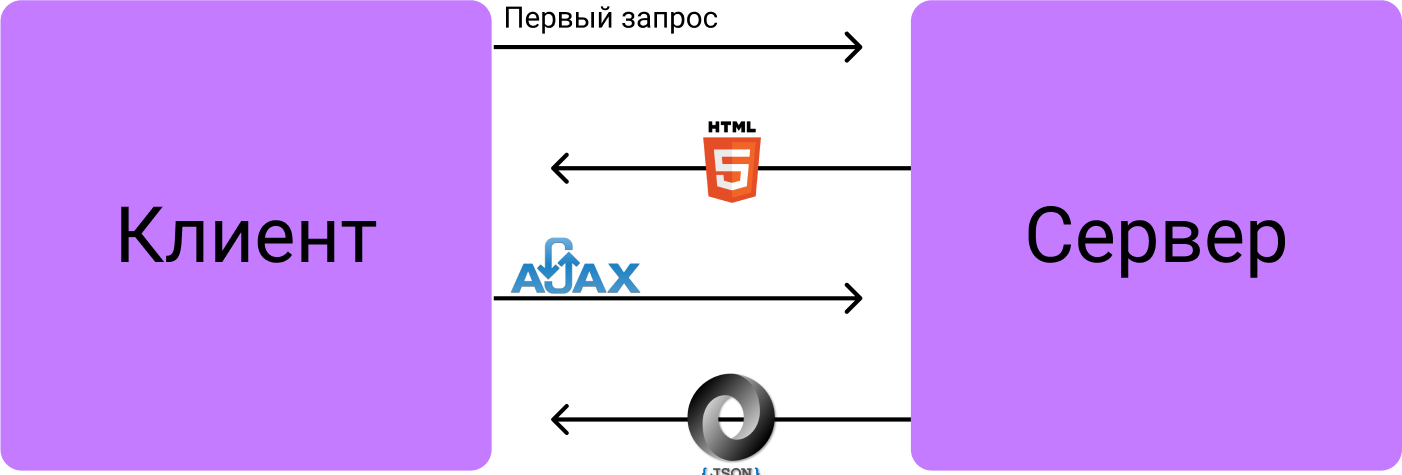
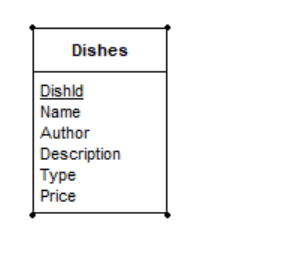


Рисунок 6 – Концепция одностраничного приложения

Каждая концепция имеет свои преимущества и недостатки и хорошо подходит для определенного типа проекта. Одностраничные приложения отличаются своей скоростью и возможностью на базе готового кода разработать мобильное приложение. Классическая концепция подходит для создания крупных интернет-магазинов, бизнес сайтов, каталогов.

## Логическая модель базы данных

Ниже приведена схема логической модели базы данных, выполненная в ER Assistant (рисунок 7).



Изм.

Лист

№ докум.

Подпись

Дата

Лист

19

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Рисунок 7 – Логическая модель базы данных

## Нормализация таблиц

Таблица приведена к третьей нормальной форме.

Первая нормальная форма − отношение называется нормализованным, если все его атрибуты просты, далее – неделимы.

Вторая нормальная форма – отношение находится во второй нормальной форме, если оно соответствует требованиям первой нормальной формы и каждый не ключевой атрибут функционально зависит от первичного ключа.

Третья нормальная форма − отношение находится в третьей нормальной форме, если оно соответствует требованиям второй нормальной формы и каждый не ключевой атрибут не транзитивно зависит от первичного ключа.

## Описание таблиц

В данной таблице представлено описание таблицы базы данных (таблица 1).

Таблица 1 – Базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Описание | Тип | Примечание |
| DishID | Номер блюда | int | PK(Первичный ключ) |
| Name | Название блюда | nvarchar(100) |  |
| Author | Наименование кухни | nvarchar(100) |  |
| Description | Описание | nvarchar(500) |  |
| Type | Категория | nvarchar(500) |  |
| Price | Цена | decimal(16, 2) |  |

* + 1. **Физическая модель базы данных**

База данных спроектирована средствами Microsoft SQL Server. Ниже приведена физическая модель базы данных (рисунок 8).

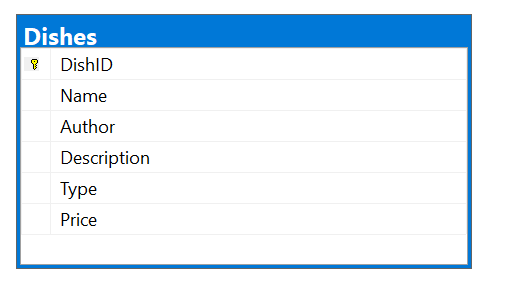
****

Рисунок 8 – Физическая модель базы данных

Изм.

Лист

№ докум.

Подпись

Дата

Лист

20

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

## Другие объекты базы данных

Ниже изображено представление таблиц (рисунок 9)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

21

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

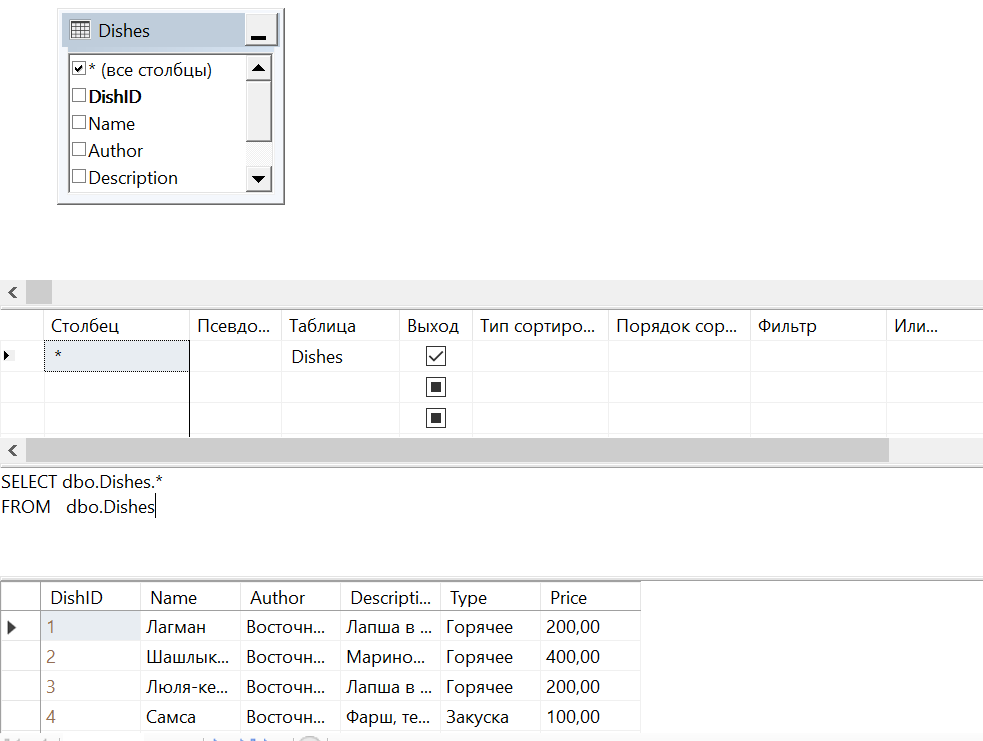


Рисунок 9 – Другие объекты базы данных

## Разработка программного изделия

Данный раздел описывает стадии и методы разработки, а также паттерны, используемые в процессе разработки и интерфейс приложения.

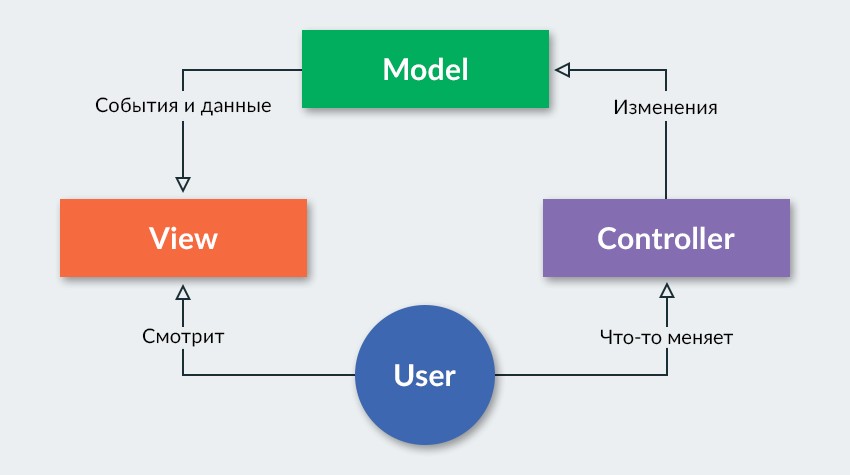
## 2.3.1 Стадии разработки

Разработка программного изделия осуществляется в четыре стадии:

* составление технического задания,
* проектирование,
* реализация,
* тестирование.

## 2.3.1 Шаблон проектирования MVC

Шаблон проектирования MVC разделяет внутреннюю структуру приложения на три компонента (рисунок 10). Шаблон проектирования MVC применялся только для серверной части приложения.



Изм.

Лист

№ докум.

Подпись

Дата

Лист

22

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Рисунок 10 – Шаблон проектирования MVC

MVC описывает простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, приложение легче масштабируется, тестируется, сопровождается и реализуется. В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Преимущества:

* единая концепция системы,
* облегчается поддержка и тестирование кода,
* удобство выводить разные представления для разных типов устройств.

Недостатки:

* необходимость использования большого количества ресурсов,
* усложнен механизм разделения программы на модули,
* усложнен процесс расширения функционала.

## 2.3.2 Интерфейс приложения

Каждая страница веб-сайта состоит из отдельных компонент. Взаимодействие между этими компонентами происходит через общее состояние приложение. Шаблон каждой страницы состоит из следующих компонент (рисунок 14):

* шапка,
* боковая панель,
* основной контент,
* подвал.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

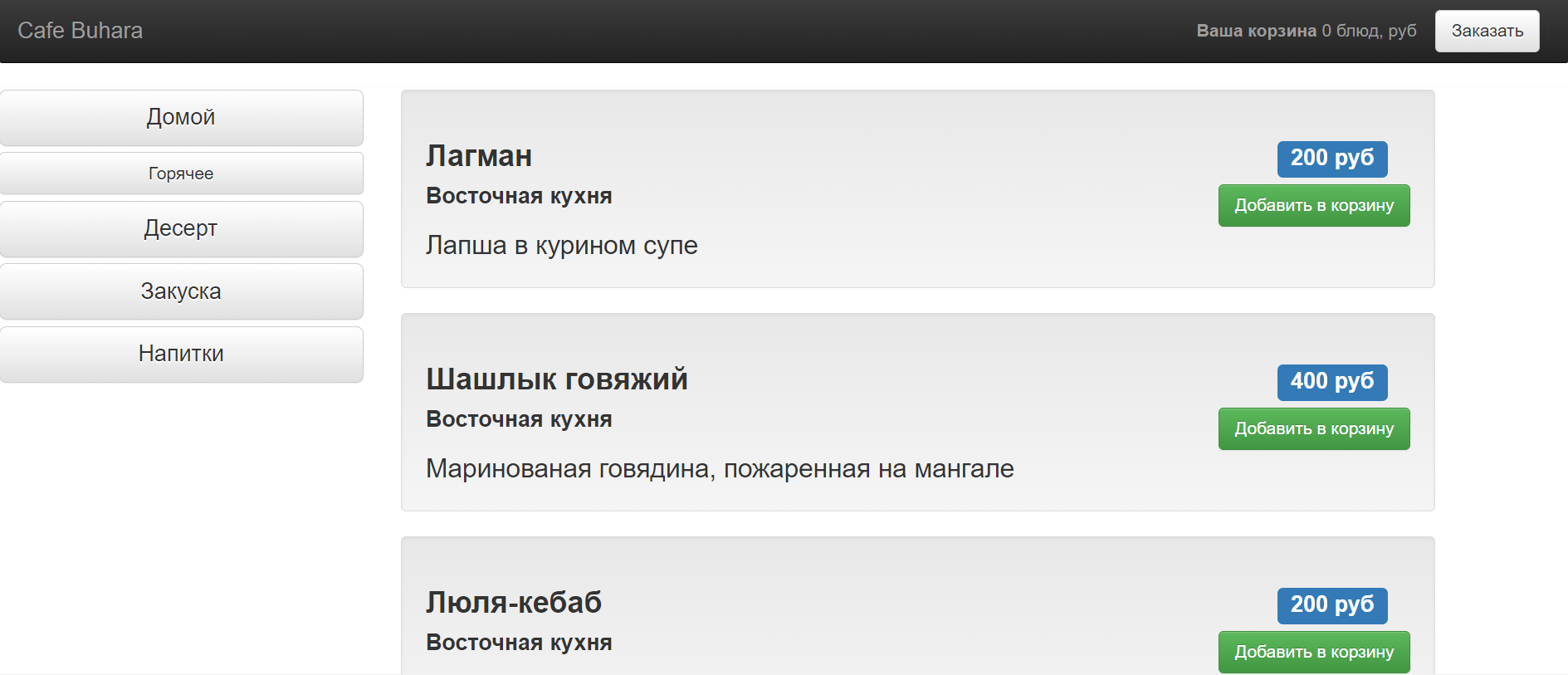
23

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП



Рисунок 11 – Шаблон сайта

Ниже представлена главная страница веб-сайта (рисунок 12).



Изм.

Лист

№ докум.

Подпись

Дата

Лист

24

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Рисунок 12 – Главная страница веб-сайта

## Защита программного изделия

Для защиты информационной системы будут использоваться следующие методы:

* ограничение прав доступа к объектам базы данных,
* хеширование паролей в таблицах базы данных.

## Тестирование программного изделия

На данной таблице представлены функции приложения и их тестирование.

Таблица 2 – Методы проверки требований к приложению

|  |  |  |
| --- | --- | --- |
| Действие | Результат | Ожидаемый результат |
| 1 | 2 | 3 |
| Запуск веб−сайта | Загружается главная страница | Загружается главная страница |
| Поиск блюд по категориям | По нужной категории подгружаются блюда из БД | По нужной категории подгружаются блюда из БД |
| Кнопка «Добавить в корзину» | Выбранные блюда, добавляются в корзину и автоматически высчитывается сумма заказа | Выбранные блюда, добавляются в корзину и автоматически высчитывается сумма заказа |
| Кнопка «Удалить» | Удаляет ненужные товары из корзины | Удаляет ненужные товары из корзины |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «Оформить заказ» | Переключает пользователя на форму заполнения доставки еды | Переключает пользователя на форму заполнения доставки |
| Кнопка «Обработать заказ» | После нажатия на данную кнопку, заказ обрабатывается и пользователю приходит благодарность за то, что он оформил заказ и что заказ будет доставлен в ближайшее время | После нажатия на данную кнопку, заказ обрабатывается и пользователю приходит благодарность за то, что он оформил заказ и что заказ будет доставлен в ближайшее время |
| CRUD | У администратора будет соответствующая админ панель, через которую он сможет добавлять, просматривать, редактировать и удалять данные. Все изменения данных будут храниться в соответствующей БД | У администратора будет соответствующая админ панель, через которую он сможет добавлять, просматривать, редактировать и удалять данные. Все изменения данных будут храниться в соответствующей БД |

Тестирование веб-сайта происходит самыми разными способами. На сегодняшний день применяются следующие виды тестирования:

Изм.

Лист

№ докум.

Подпись

Дата

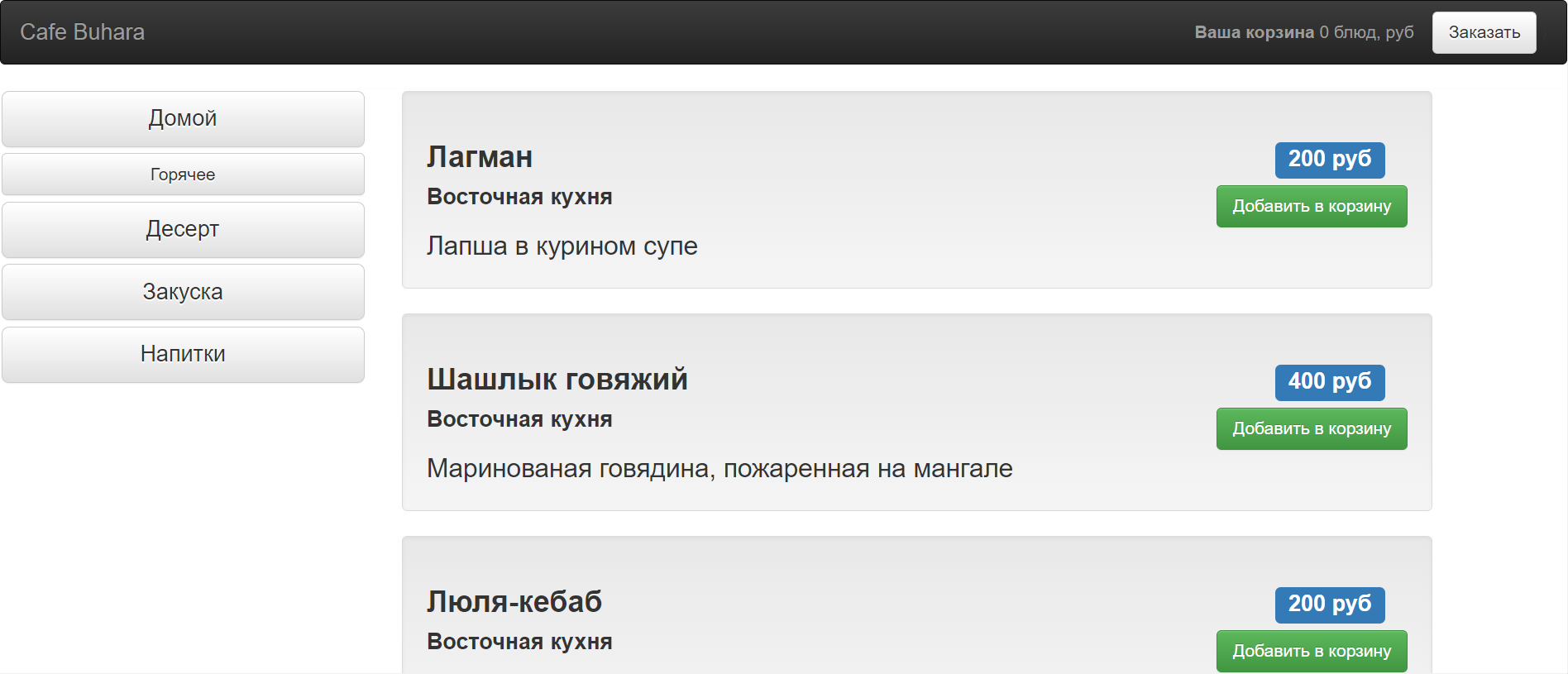
Лист

25

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

* функциональное тестирование. Тестирование веб-сайта на работоспособность основных функций. Публикация успешно добавляется в базу данных, все страницы веб-сайта корректно загружаются, после добавление публикации происходит автоматический расчет определенных характеристик,
* тестирование пользовательского интерфейса. Данный вид тестирования предполагает соответствие на стандарты графических интерфейсов,
* тестирование безопасности. Проверка возможностей сайта для авторизованных и неавторизованных пользователей. Неавторизованный пользователь не может создать и комментировать публикации. При переходе по несуществующей странице происходит перенаправление на главную страницу,
* тестирование юзабилити сайта. Проверка веб-сайта на удобство и практичность в использовании,
* тестирование базы данных. Проверка целостности данных, представлений, схем, таблиц, индексов и ключей. Все запросы к базе данных работают корректно, данные добавляются и извлекаются должным образом.

Ниже приведены функции веб-сайта (рисунок 13 – 17).



Изм.

Лист

№ докум.

Подпись

Дата

Лист

26

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Рисунок 13 – Запуск веб-сайта, загружается главная страница

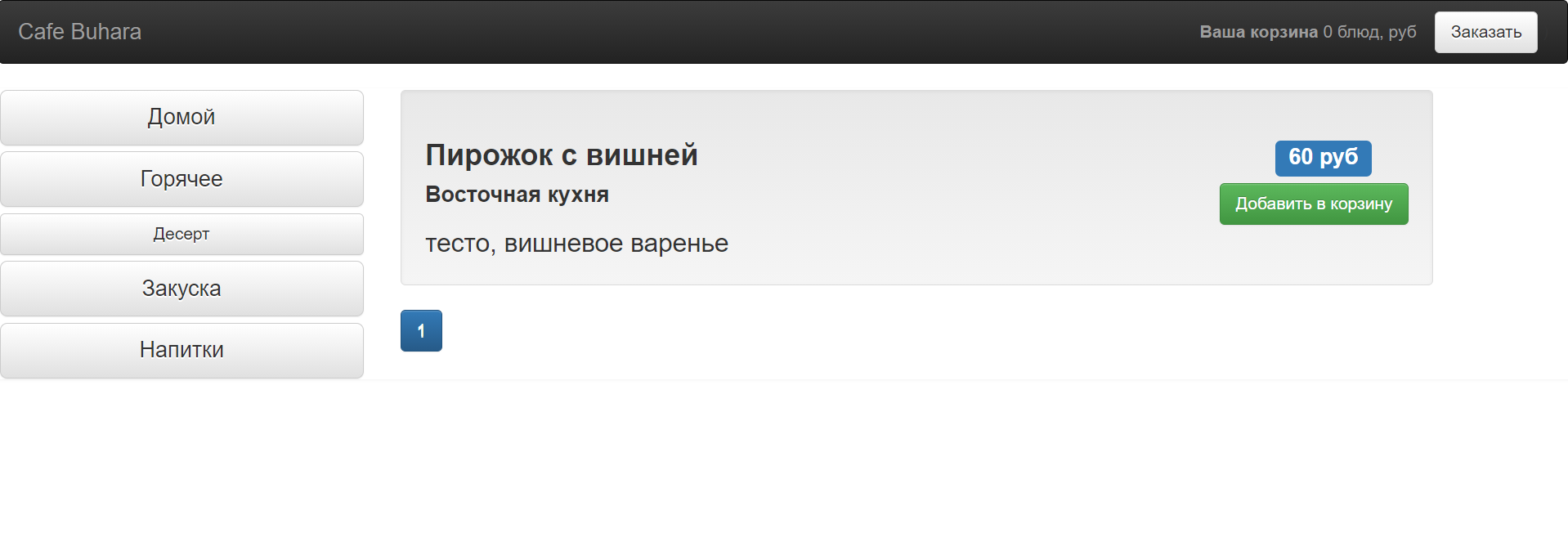


Рисунок 14 – Поиск блюд по категориям

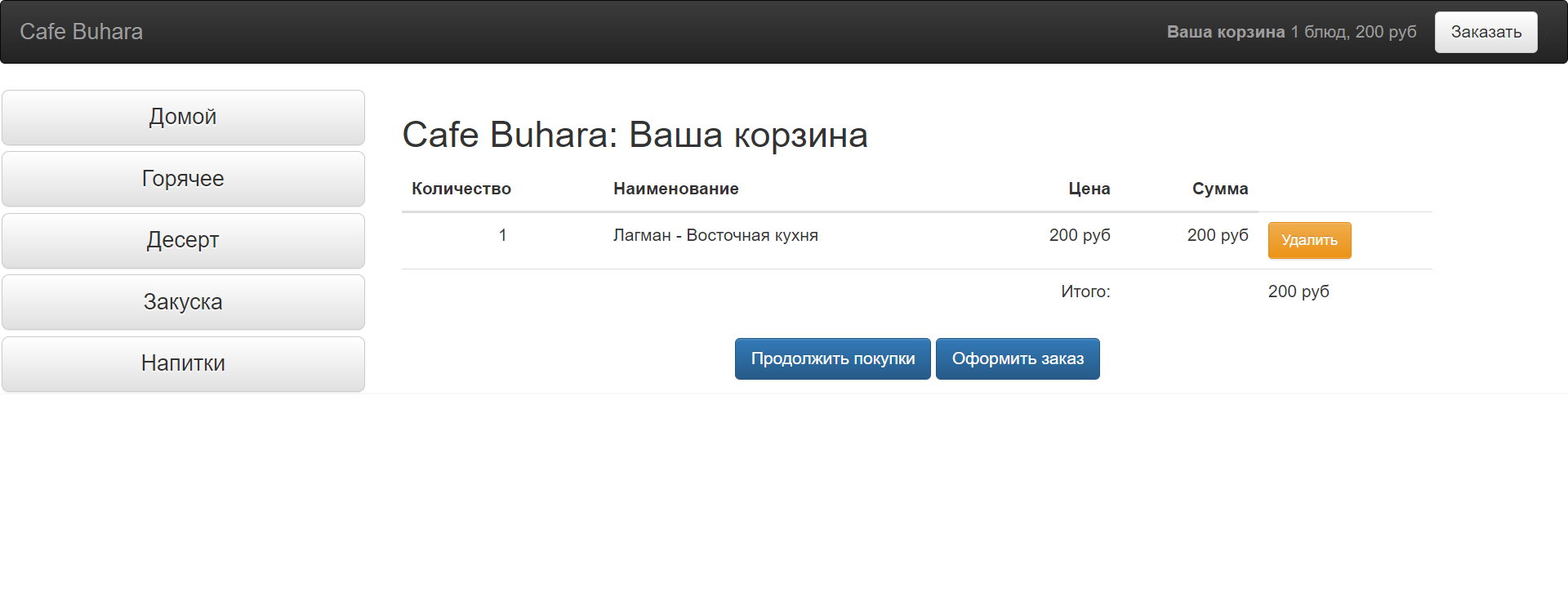
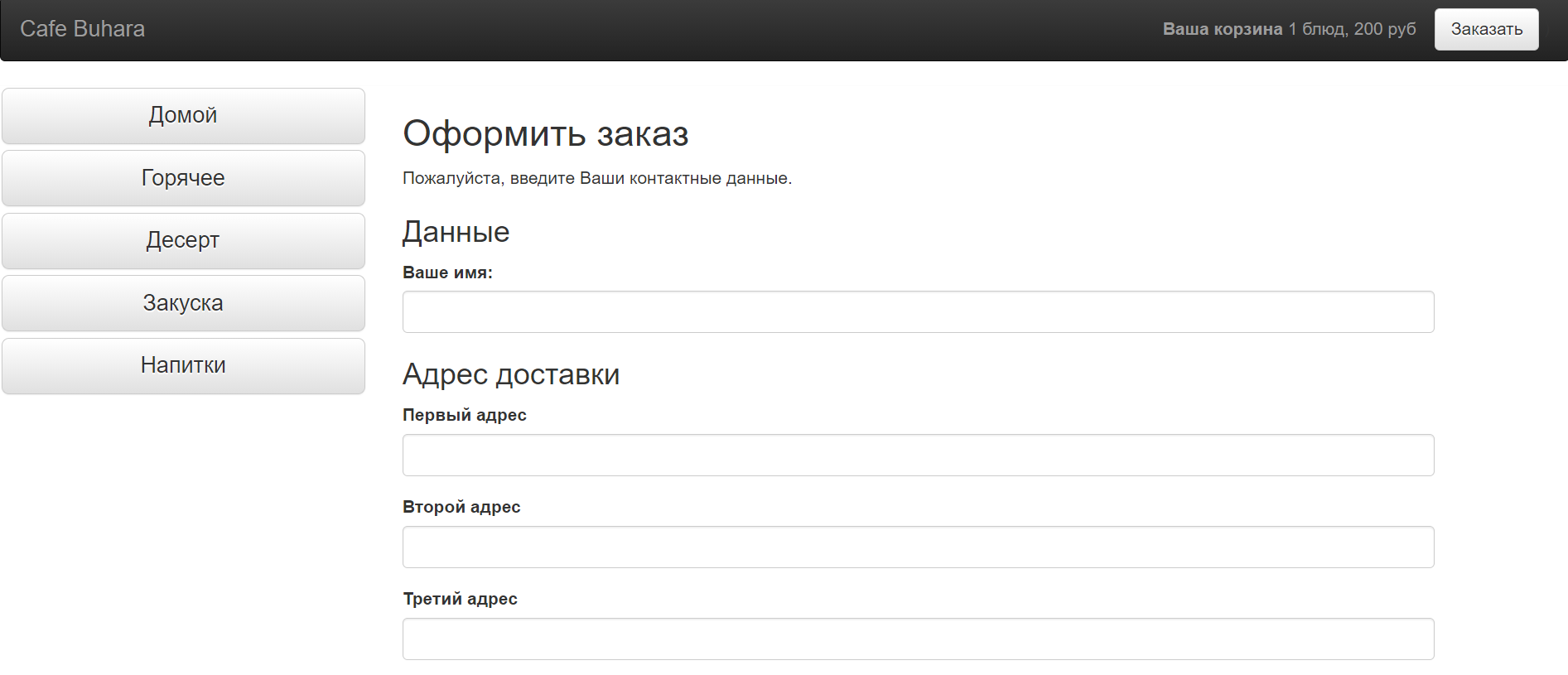


Рисунок 15 – Корзина покупок, добавляет нужные товары в корзину и высчитывает автоматически сумму заказа



Изм.

Лист

№ докум.

Подпись

Дата

Лист

27

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Рисунок 16 – Форма заполнения доставки еды и обработка заказа

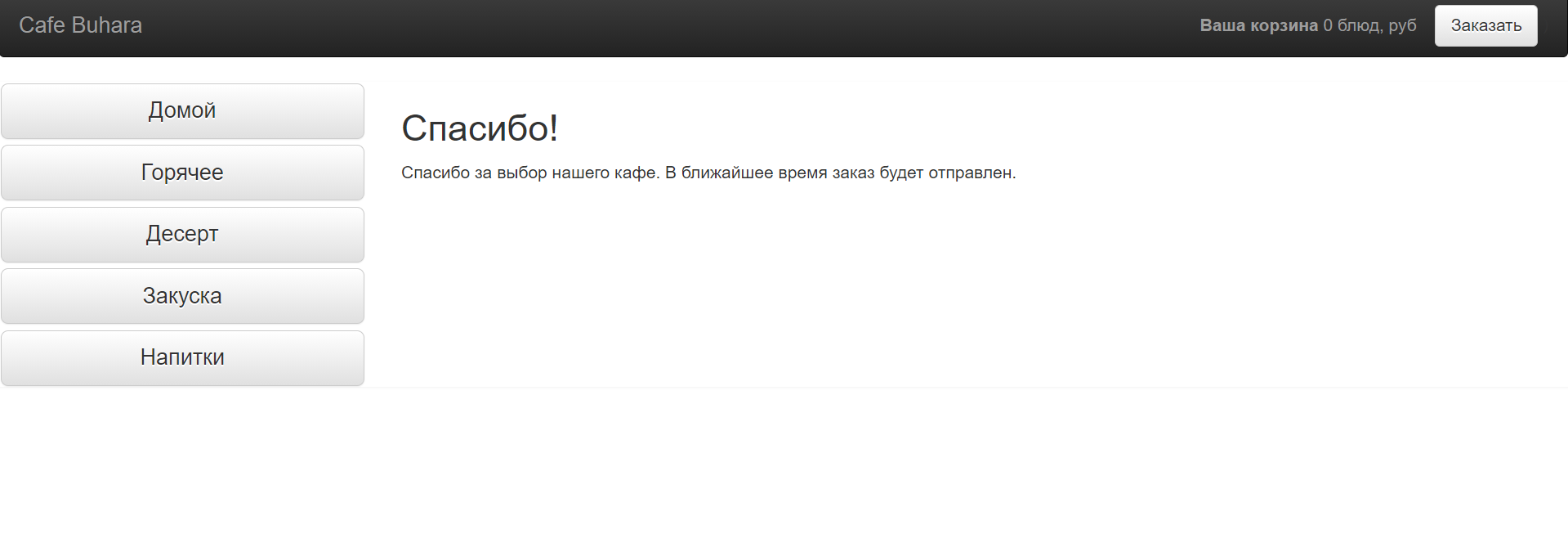


Рисунок 17 – После обработки заказа, пользователь получает благодарность за оформление заказа.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

28

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

# ЗАКЛЮЧЕНИЕ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

29

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

Использование данного веб-сайта предполагает ознакомление пользователей с блюдами данного предприятия, чтобы пользователи могли заказывать еду на дом и чтобы они могли следить за актуальной информацией на сайте. Все это нужно для расширения базы клиентов и для дальнейшего роста, потому что чем больше доход тем лучше для заведения. Когда-то такого понятия, как «доставка еды» в принципе не было, людям приходилось посещать очно подобные заведения, потом были просто частные курьеры, но их работа не была привязана к какому-либо ПО, сейчас же 21 век, это век технологий, возможность доставки еды, должна быть у каждого начинающего предприятия подобного рода.

По итогу выпускной квалификационной работы был создан веб-сайт, который позволяет расширить базу клиентов у начинающего предприятия и для дальнейшего роста, потому что чем больше доход тем лучше для заведения.

В процессе выполнения дипломного проекта была спроектирована база данных, разработана логическая модель в ER Assistant, UML-диаграмма вариантов использования, физическая модель в СУБД Microsoft SQL, спроектирован дизайн веб-сайта, разработан сам веб-сайт, произведено тестирование, а также подготовлена презентация и оформлена документация, удовлетворяющая стандартам ГОСТ.

Во время разработки дипломного проекта были закреплены навыки работы с ASP.NET MVC Framework, с языком программирования C# и JavaScript, с CSS-фреймворком Bootstrap, языками разметки HTML и CSS. Также с технологией, которая позволяет связать наше приложение с базой данных Entity Framework и непосредственно были закреплены навыки работы с СУБД Microsoft SQL Server, в работе с представлениями и SQL-запросами для создания объектов в базе данных. Перспективой разработки данного веб-сайта является добавление новых функций, улучшение дизайна сайта, разработка мобильного приложения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

30

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

1. ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам (с Изменением N 1).
2. ГОСТ 19.106-78 Единая система программной документации (ЕСПД). Требования к программным документам, выполненным печатным способом (с Изменением N 1).
3. ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.
4. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
5. ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
6. ГОСТ 34.603-92 Информационная технология. Виды испытаний автоматизированных систем.
7. Сайт вопросов и ответов «Хабр» [Электронный ресурс]. – 2021. – URL: https://qna.habr.com/ (дата обращения 16.05.2021).
8. Система управления проектами и версиями кода «GitHub» [Электронный ресурс]. – 2021. – URL: https://github.com/ (дата обращения 18.05.2021).
9. Bootstrap [Электронный ресурс]. — 2021. — URL: https://bootstrap-4.ru/docs/4.5/getting-started/introduction/ (дата обращения 20.05.2021).
10. Рукововодства по SQL Server [Электронный ресурс]. – 2021. – URL: https://docs.microsoft.com/ru-ru/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver15/ (дата обращения 21.05.2021).
11. Руководства по ASP.NET MVC Framework [Электронный ресурс]. – 2021. – URL: https://metanit.com/sharp/mvc5/ (дата обращения 24.05.2021).
12. Сайт вопросов и ответов «Stack Overflow» [Электронный ресурс]. –2021. – URL: https://ru.stackoverflow.com/ (дата обращения 26.05.2021).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

31

ФГБОУ ВО «МГУТУ им. К Г Разумовского (ПКУ)» 09.02.03-090203-9о-17/2-20-2021 - ДП

1. Руководство по программированию на C# [Электронный ресурс]. – 2021. – URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/ (дата обращения 29.05.2021).
2. Документация по Entity Framework. [Электронный ресурс]. − 2021. − URL: https://docs.microsoft.com/ru-ru/ef/ (дата обращения 01.06.2021).
3. Справочник по JavaScript. [Электронный ресурс]. − 2021. − URL: https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference (дата обращения 03.06.2021).
4. IoC-контейнер Ninject. [Электронный ресурс]. − 2021. − URL: https://metanit.com/sharp/mvc5/21.2.php (дата обращения 05.06.2021).
5. Фреймворк Moq. [Электронный ресурс]. − 2021. − URL: https://metanit.com/sharp/mvc5/18.5.php (дата обращения 07.06.2021).
6. Modernizr: библиотека обнаружений фуункций для HTML5/CSS3. [Электронный ресурс]. − 2021. − URL: https://modernizr.com/ (дата обращения 09.06.2021).
7. 10 советов как пользоваться хостингом. [Электронный ресурс]. − 2021. − URL: https://host4.biz/ru/blog/kak-pravilno-polzovatsya-hostingom-10-poleznyh-sovetov (дата обращения 11.06.2021).
8. Виртуальный хостинг. [Электронный ресурс]. − 2021. − URL: http://gotw.ru/?yclid=3331175529723246817 (дата обращения 13.06.2021).

# ПРИЛОЖЕНИЕ A

**SQL скрипты**

USE [master]

GO

/\*\*\*\*\*\* Object: Database [Diplom] Script Date: 09.06.2021 17:52:12 \*\*\*\*\*\*/

CREATE DATABASE [Diplom]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Diplom', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Diplom.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'Diplom\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Diplom\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT

GO

ALTER DATABASE [Diplom] SET COMPATIBILITY\_LEVEL = 150

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [Diplom].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [Diplom] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [Diplom] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [Diplom] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [Diplom] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [Diplom] SET ARITHABORT OFF

GO

ALTER DATABASE [Diplom] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [Diplom] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [Diplom] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [Diplom] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [Diplom] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [Diplom] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [Diplom] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [Diplom] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [Diplom] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [Diplom] SET DISABLE\_BROKER

GO

ALTER DATABASE [Diplom] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [Diplom] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [Diplom] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [Diplom] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [Diplom] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [Diplom] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [Diplom] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [Diplom] SET RECOVERY SIMPLE

GO

ALTER DATABASE [Diplom] SET MULTI\_USER

GO

ALTER DATABASE [Diplom] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [Diplom] SET DB\_CHAINING OFF

GO

ALTER DATABASE [Diplom] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [Diplom] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [Diplom] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [Diplom] SET ACCELERATED\_DATABASE\_RECOVERY = OFF

GO

ALTER DATABASE [Diplom] SET QUERY\_STORE = OFF

GO

USE [Diplom]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dishes] Script Date: 09.06.2021 17:52:12 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dishes](

[DishID] [int] IDENTITY(1,1) NOT NULL,

[Name] [nvarchar](100) NOT NULL,

[Author] [nvarchar](100) NOT NULL,

[Description] [nvarchar](500) NOT NULL,

[Type] [nvarchar](500) NOT NULL,

[Price] [decimal](16, 2) NOT NULL,

CONSTRAINT [PK\_\_Dishes\_\_18834F709E48427D] PRIMARY KEY CLUSTERED

(

[DishID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

USE [master]

GO

ALTER DATABASE [Diplom] SET READ\_WRITE

GO

# ПРИЛОЖЕНИЕ Б

**Код приложения**

//Маршрутизация

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

namespace WebUI

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

null,

"",

new { controller = "Dishes", action = "List", type = (string)null, page = 1 }

);

routes.MapRoute(

name: null,

url: "Page{page}",

defaults: new { controller = "Dishes", action = "List", type = (string)null },

constraints: new { page = @"\d+" }

);

routes.MapRoute(

null,

"{type}",

new { controller = "Dishes", action = "List", page = 1}

);

routes.MapRoute(

null,

"{type}/Page{page}",

new {controller = "Dishes", action = "List" },

new {page = @"\d+"}

);

routes.MapRoute(

null,

"{controller}/{action}"

);

}

}

}

//репозиторий

using Domain.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domain.Abstract

{

public interface IDishRepository

{

IEnumerable<Dish> Dishes { get; }

}

}

// для работы с классами

using Domain.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domain.Abstract

{

public interface IProcessorOrder

{

void ProcessOrder(Bascket bascket, DeliveryDetails deliveryDetails);

}

}

//для работы с репозиторием

using Domain.Abstract;

using Domain.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domain.Concreate

{

public class EDDishRepository : IDishRepository

{

EFDbContext context = new EFDbContext();

public IEnumerable<Dish> Dishes

{

get { return context.Dishes; }

}

}

}

//DbContext

using Domain.Entities;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domain.Concreate

{

public class EFDbContext : DbContext

{

public DbSet<Dish> Dishes { get; set; }

}

}

// классы которые описывает оформление и обработку заказов

using Domain.Abstract;

using Domain.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Mail;

namespace Domain.Concreate

{

public class EmailSettings

{

public string AddressToMail = "orders@example.com";

public string AddressFromMail = "cafebuhara@example.com";

public bool SslUse = true;

public string Username = "MynameUser";

public string Password = "MyPassword";

public string NameServer = "smtp.example.com";

public int PortServer = 587;

public bool WriteAsFile = true;

public string LocationFile = @"c:\cafe\_buhara\_emails";

}

public class ProcessorOrderEmail : IProcessorOrder

{

private EmailSettings emailSettings;

public ProcessorOrderEmail (EmailSettings settings)

{

emailSettings = settings;

}

public void ProcessOrder(Entities.Bascket bascket, DeliveryDetails deliveryDetails)

{

using (var smtpClient = new SmtpClient())

{

smtpClient.EnableSsl = emailSettings.SslUse;

smtpClient.Host = emailSettings.NameServer;

smtpClient.Port = emailSettings.PortServer;

smtpClient.UseDefaultCredentials = false;

smtpClient.Credentials = new NetworkCredential(emailSettings.Username, emailSettings.Password);

if (emailSettings.WriteAsFile)

{

smtpClient.DeliveryMethod = SmtpDeliveryMethod.SpecifiedPickupDirectory;

smtpClient.PickupDirectoryLocation = emailSettings.LocationFile;

smtpClient.EnableSsl = false;

}

StringBuilder body = new StringBuilder()

.AppendLine("Новый заказ обработан")

.AppendLine("---")

.AppendLine("Товары:");

foreach (var line in bascket.Lines)

{

var totalsub = line.Dish.Price \* line.Quantity;

body.AppendFormat("{0} x {1} (итого: {2:c}", line.Quantity, line.Dish.Name, totalsub);

}

body.AppendFormat("Общая стоимость: {0:c}", bascket.CalculateTotalValue())

.AppendLine("---")

.AppendLine("Доставка:")

.AppendLine(deliveryDetails.Name)

.AppendLine(deliveryDetails.Address1)

.AppendLine(deliveryDetails.Address2 ?? "")

.AppendLine(deliveryDetails.Address3 ?? "")

.AppendLine(deliveryDetails.City)

.AppendLine(deliveryDetails.Country)

.AppendLine("---")

.AppendFormat("Подарочная упаковка: {0}", deliveryDetails.GiftPackaging ? "Да" : "Нет");

MailMessage mailMessage = new MailMessage(

emailSettings.AddressFromMail,

emailSettings.AddressToMail,

"Новый заказ отправлен!",

body.ToString()

);

if (emailSettings.WriteAsFile)

{

mailMessage.BodyEncoding = Encoding.UTF8;

}

smtpClient.Send(mailMessage);

}

}

}

}

//класс корзина

using Domain.Abstract;

using Domain.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Mail;

namespace Domain.Concreate

{

public class EmailSettings

{

public string AddressToMail = "orders@example.com";

public string AddressFromMail = "cafebuhara@example.com";

public bool SslUse = true;

public string Username = "MynameUser";

public string Password = "MyPassword";

public string NameServer = "smtp.example.com";

public int PortServer = 587;

public bool WriteAsFile = true;

public string LocationFile = @"c:\cafe\_buhara\_emails";

}

public class ProcessorOrderEmail : IProcessorOrder

{

private EmailSettings emailSettings;

public ProcessorOrderEmail (EmailSettings settings)

{

emailSettings = settings;

}

public void ProcessOrder(Entities.Bascket bascket, DeliveryDetails deliveryDetails)

{

using (var smtpClient = new SmtpClient())

{

smtpClient.EnableSsl = emailSettings.SslUse;

smtpClient.Host = emailSettings.NameServer;

smtpClient.Port = emailSettings.PortServer;

smtpClient.UseDefaultCredentials = false;

smtpClient.Credentials = new NetworkCredential(emailSettings.Username, emailSettings.Password);

if (emailSettings.WriteAsFile)

{

smtpClient.DeliveryMethod = SmtpDeliveryMethod.SpecifiedPickupDirectory;

smtpClient.PickupDirectoryLocation = emailSettings.LocationFile;

smtpClient.EnableSsl = false;

}

StringBuilder body = new StringBuilder()

.AppendLine("Новый заказ обработан")

.AppendLine("---")

.AppendLine("Товары:");

foreach (var line in bascket.Lines)

{

var totalsub = line.Dish.Price \* line.Quantity;

body.AppendFormat("{0} x {1} (итого: {2:c}", line.Quantity, line.Dish.Name, totalsub);

}

body.AppendFormat("Общая стоимость: {0:c}", bascket.CalculateTotalValue())

.AppendLine("---")

.AppendLine("Доставка:")

.AppendLine(deliveryDetails.Name)

.AppendLine(deliveryDetails.Address1)

.AppendLine(deliveryDetails.Address2 ?? "")

.AppendLine(deliveryDetails.Address3 ?? "")

.AppendLine(deliveryDetails.City)

.AppendLine(deliveryDetails.Country)

.AppendLine("---")

.AppendFormat("Подарочная упаковка: {0}", deliveryDetails.GiftPackaging ? "Да" : "Нет");

MailMessage mailMessage = new MailMessage(

emailSettings.AddressFromMail,

emailSettings.AddressToMail,

"Новый заказ отправлен!",

body.ToString()

);

if (emailSettings.WriteAsFile)

{

mailMessage.BodyEncoding = Encoding.UTF8;

}

smtpClient.Send(mailMessage);

}

}

}

}

//класс доставка еды, оформление заказа

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domain.Entities

{

public class DeliveryDetails

{

[Required(ErrorMessage = "Укажите Ваше имя")]

public string Name { get; set; }

[Required(ErrorMessage = "Адрес доставки")]

[Display(Name="Первый адрес")]

public string Address1 { get; set; }

[Display(Name = "Второй адрес")]

public string Address2 { get; set; }

[Display(Name = "Третий адрес")]

public string Address3 { get; set; }

[Required(ErrorMessage = "Укажите город")]

[Display(Name = "Город")]

public string City { get; set; }

[Required(ErrorMessage = "Укажите страну")]

[Display(Name = "Страна")]

public string Country { get; set; }

public bool GiftPackaging { get; set; }

}

}

//класс для работы с атрибутами из таблицы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domain.Entities

{

public class Dish

{

public int DishID { get; set; }

public string Name { get; set; }

public string Author { get; set; }

public string Description { get; set; }

public string Type { get; set; }

public decimal Price { get; set; }

}

}

//контроллер корзины

using Domain.Abstract;

using Domain.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using WebUI.Models;

namespace WebUI.Controllers

{

public class BascketController : Controller

{

private IDishRepository storage;

private IProcessorOrder processorOrder;

public BascketController(IDishRepository repo, IProcessorOrder processor)

{

storage = repo;

processorOrder = processor;

}

public ViewResult Index(Bascket bascket, string returnUrl)

{

return View(new BascketIndexViewModel

{

Bascket = bascket,

ReturnUrl = returnUrl

});

}

public RedirectToRouteResult AddToBascket(Bascket bascket, int dishID, string returnUrl)

{

Dish dish = storage.Dishes

.FirstOrDefault(x => x.DishID == dishID);

if (dish != null )

{

bascket.AddItem(dish, 1);

}

return RedirectToAction("Index", new { returnUrl });

}

public RedirectToRouteResult DeleteFromBascket(Bascket bascket, int dishID, string returnUrl)

{

Dish dish = storage.Dishes

.FirstOrDefault(x => x.DishID == dishID);

if ( dish != null )

{

bascket.DeleteLine(dish);

}

return RedirectToAction("Index", new { returnUrl });

}

public PartialViewResult Resume(Bascket bascket)

{

return PartialView(bascket);

}

public ViewResult Checkout()

{

return View(new DeliveryDetails());

}

[HttpPost]

public ViewResult Checkout(Bascket bascket, DeliveryDetails deliveryDetails)

{

if (bascket.Lines.Count() == 0)

{

ModelState.AddModelError("", "Извините, корзина пуста!");

}

if (ModelState.IsValid)

{

processorOrder.ProcessOrder(bascket, deliveryDetails);

bascket.Clear();

return View("Completed");

}

else

return View(new DeliveryDetails());

}

}

}

//Котроллер блюд

using Domain.Abstract;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using WebUI.Models;

namespace WebUI.Controllers

{

public class DishesController : Controller

{

// GET: Dishes

private IDishRepository repository;

public int pageSize = 4;

public DishesController(IDishRepository repo)

{

repository = repo;

}

public ViewResult List(string type, int page = 1)

{

DishesListViewModel model = new DishesListViewModel

{

Dishes = repository.Dishes

.Where(b => type == null || b.Type == type)

.OrderBy(dish => dish.DishID)

.Skip((page - 1) \* pageSize)

.Take(pageSize),

PagingInfo = new PagingInfo

{

CurrentPage = page,

ItemsPerPage = pageSize,

TotalItems = type == null ?

repository.Dishes.Count() :

repository.Dishes.Where(dish => dish.Type == type).Count()

},

CurrentType = type

};

return View(model);

}

}

}

//контроллер навигации

using Domain.Abstract;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace WebUI.Controllers

{

public class NavController : Controller

{

private IDishRepository repository;

public NavController(IDishRepository repo)

{

repository = repo;

}

public PartialViewResult Menu(string type = null)

{

ViewBag.SelectedType = type;

IEnumerable<string> types = repository.Dishes

.Select(dish => dish.Type)

.Distinct()

.OrderBy(x => x);

return PartialView(types);

}

}

}

//класс для работы со страницами

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Web;

using System.Web.Mvc;

using WebUI.Models;

namespace WebUI.HtmlHelpers

{

public static class PagingHelper

{

public static MvcHtmlString PageLinks(this HtmlHelper html, PagingInfo pagingInfo, Func<int, string> pageUrl)

{

StringBuilder result = new StringBuilder();

for (int i = 1; i <= pagingInfo.TotalPages; i++)

{

TagBuilder tag = new TagBuilder("a");

tag.MergeAttribute("href", pageUrl(i));

tag.InnerHtml = i.ToString();

if (i == pagingInfo.CurrentPage)

{

tag.AddCssClass("selected");

tag.AddCssClass("btn-primary");

}

tag.AddCssClass("btn btn-default");

result.Append(tag.ToString());

}

return MvcHtmlString.Create(result.ToString());

}

}

}

//представление 1

@using WebUI.Models

@model DishesListViewModel

@{

ViewBag.Title = "Блюда";

}

@foreach (var dish in Model.Dishes)

{

@Html.Partial("\_DishSummary", dish)

}

<div class="btn-group pull right">

@Html.PageLinks(Model.PagingInfo, x => Url.Action("List", new { page = x, type = Model.CurrentType }))

</div>

//представление 2

@model IEnumerable<string>

@Html.ActionLink("Домой", "List", "Dishes", null, new {@class = "btn btn-block btn-default btn-lg"})

@foreach (var link in Model)

{

@Html.RouteLink(link, new

{

controller = "Dishes",

action = "List",

type = link,

page = 1

}, new

{

@class = "btn btn-block btn-default btn-lg"

+ (link == ViewBag.SelectedType ? "btn-primary" : "")

})

}

//представление 3

@model Domain.Entities.Dish

<div class="well">

<div class="row">

<div class="col-xs-10">

<h3>

<strong>@Model.Name</strong>

</h3>

<h4>

<strong>@Model.Author</strong>

</h4>

</div>

<div class="col-xs-2">

<h3>

<strong id="price"class="pull right label label-primary">

@Model.Price.ToString("# руб")

</strong>

</h3>

@using (Html.BeginForm("AddToBascket", "Bascket"))

{

<div class="pull-right">

@Html.HiddenFor(x => x.DishID)

@Html.Hidden("returnUrl", Request.Url.PathAndQuery)

<input type="submit" class="btn btn-success" value="Добавить в корзину"/>

</div>

}

</div>

</div>

<span class="lead"> @Model.Description</span>

</div>

//представление 4

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="~/Content/bootstrap.min.css" rel="stylesheet" />

<link href="~/Content/bootstrap-theme.min.css" rel="stylesheet" />

<link href="~/Content/Site.css" rel="stylesheet" />

<title>@ViewBag.Title</title>

<style>

.navbar-right {

float: right !important;

margin-right: 15px;

margin-left: 15px;

}

</style>

</head>

<body>

<div class="navbar navbar-inverse" role="navigation">

<a class="navbar-brand" href="#">Cafe Buhara</a>

@Html.Action("Resume", "Bascket")

</div>

<div class="row panel">

<div id="types" class="col-xs-3">

@Html.Action("Menu", "Nav")

</div>

<div class="col-xs-8">

@RenderBody()

</div>

</div>

</body>

</html>

//представление 5

@model Domain.Entities.DeliveryDetails

@{

ViewBag.Title = "Cafe Buhara: форма заказа";

}

<h2>Оформить заказ</h2>

<p>Пожалуйста, введите Ваши контактные данные.</p>

@using (Html.BeginForm())

{

@Html.ValidationSummary();

<h3>Данные</h3>

<div class="form-group">

<label>Ваше имя:</label>

@Html.TextBoxFor(b => b.Name, new { @class = "form-control" })

</div>

<h3>Адрес доставки</h3>

foreach (var property in ViewData.ModelMetadata.Properties)

{

if (property.PropertyName != "Name" && property.PropertyName != "GiftPackaging")

{

<div class="form-group">

<label>

@(property.DisplayName ?? property.PropertyName)

</label>

@Html.TextBox(property.PropertyName, null, new { @class = "form-control" })

</div>

}

}

<h3>Опции</h3>

<div class="form-group">

<label>

@Html.EditorFor(b => b.GiftPackaging)

Использовать подарочную упаковку?

</label>

</div>

<div class="text-center">

<input class="btn btn-primary" type="submit" value="Обработать заказ" />

</div>

}

//представление 6

@model WebUI.Models.BascketIndexViewModel

@{

ViewBag.Title = "Cafe Buhara: Ваша корзина";

}

<h2>@ViewBag.Title</h2>

<table id="bascketTable" class="table">

<thead>

<tr>

<th>Количество</th>

<th>Наименование</th>

<th class="text-right">Цена</th>

<th class="text-right">Сумма</th>

</tr>

</thead>

<tbody>

@foreach (var line in Model.Bascket.Lines)

{

<tr>

<td class="text-center">@line.Quantity</td>

<td class="text-left">@line.Dish.Name - @line.Dish.Author</td>

<td class="text-right">@line.Dish.Price.ToString("# руб")</td>

<td class="text-right">@((line.Quantity \* line.Dish.Price).ToString("# руб"))</td>

<td>

@using (Html.BeginForm("RemoveFromCart", "Bascket"))

{

@Html.Hidden("DishID", line.Dish.DishID)

@Html.HiddenFor(x => x.ReturnUrl)

<input class="btn btn-sm btn-warning" type="submit" value="Удалить" />

}

</td>

</tr>

}

</tbody>

<tfoot>

<tr>

<td colspan="3" class="text-right">Итого:</td>

<td class="text-right"></td>

<td>@Model.Bascket.CalculateTotalValue().ToString("# руб")</td>

</tr>

</tfoot>

</table>

<div class="text-center">

<a class="btn btn-primary" href="@Model.ReturnUrl">Продолжить покупки</a>

@Html.ActionLink("Оформить заказ", "Checkout", null, new { @class = "btn btn-primary" })

</div>

//представление 7

@model Domain.Entities.Bascket

<div class="navbar-right">

@Html.ActionLink("Заказать", "Index", "Bascket",

new { returnUrl = Request.Url.PathAndQuery },

new { @class = "btn btn-default navbar-btn" })

)

</div>

<div class="navbar-text navbar-right">

<b>Ваша корзина</b>

@Model.Lines.Sum(x => x.Quantity) блюд,

@Model.CalculateTotalValue().ToString("# руб")

</div>