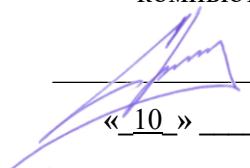


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии


СОГЛАСОВАНО

Канд. пед наук, доцент департамента
программной инженерии факультета
компьютерных наук

 С. А. Виденин
« 10 » мая 2023 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

 В. В. Шилов
« 11 » мая 2023 г.


WEB-ПРИЛОЖЕНИЕ «Tamak'»

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.09.09-01 81 01-1-ЛУ

Исполнитель:

 / Студентка группы БПИ217
/ Омирбекова Д.К. /
« 10 » мая 2023 г.

Москва 2023

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

УТВЕРЖДЕН

RU.17701729.09.09-01 81 01-1-ЛУ

WEB-ПРИЛОЖЕНИЕ «Tamak'»

Техническое задание

RU.17701729.09.09-01 81 01-1

Листов 18

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

Техническое задание – это основной документ, оговаривающий набор требований и порядок создания программного продукта, в соответствии с которым производится разработка программы, ее тестирование и приемка.

Данная пояснительная записка к Web-приложению «Татак'» содержит следующие разделы: «Введение», «Назначение и область применения», «Технические характеристики», «Ожидаемые технико-экономические показатели» и приложения.

В разделе «Введение» представлены наименование и краткое наименование программы, а также документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» представлено эксплуатационное и функциональное назначение программы, а также краткая характеристика области применения.

В разделе «Технические характеристики» объясняется постановка задачи на разработку программы, а именно клиентская и серверная часть приложения, а также описывается алгоритм и функционирование программы, в который входят: архитектура приложения, схема базы данных, описание алгоритма хеширования паролей, валидации входных данных.

В разделе «Ожидаемые технико-экономические показатели» представлены экономические преимущества приложения в сравнении с аналогами, а также ожидаемая потребность.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101–77 Виды программ и программных документов.
2. ГОСТ 19.102-77 Стадии разработки.
3. ГОСТ 19.103-77 Обозначения программ и программных документов.
4. ГОСТ 19.104-78 Основные надписи.
5. ГОСТ 19.105-78 Общие требования к программным документам.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом.
7. ГОСТ 19.201-79 Пояснительная записка. Требования к содержанию и оформлению.

Изменения к данному Пояснительной записке оформляются согласно:

8. ГОСТ 19.603-78
9. ГОСТ 19.604-78

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	2
1. ВВЕДЕНИЕ	5
1.1. Наименование программы	5
1.2. Краткая характеристика области применения	5
1.3. Документ, на основании которого ведется разработка	5
1.4. Наименование темы разработки.....	5
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....	6
2.1. Назначение программы	6
2.1.1. Функциональное назначение	6
2.1.2. Эксплуатационное назначение	6
2.2. Краткая характеристика области применения	7
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	8
3.1. Постановка задачи на разработку программы	8
3.1.1. Клиентская часть:	8
3.2. Описание алгоритма и функционирования программы	9
3.3. Описание и обоснование выбора метода организации входных и выходных данных	11
3.4. Описание и обоснование выбора состава технических и программных средств	11
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	13
4.1. Предполагаемая потребность	13
4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами	13
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	14
ТЕРМИНОЛОГИЯ	16
ОПИСАНИЕ БАЗЫ ДАННЫХ	17
ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ	19
ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ	22
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	27

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы на русском языке – «Web-приложение «Татак'»».

Наименование программы на английском языке – «Web application «Татак'»».

Краткое наименование программы – «Татак'».

1.2. Краткая характеристика области применения

«Татак'» - Web-приложение предназначено для преждевременного заказа еды в предприятиях общественного питания, располагающихся в корпусах НИУ ВШЭ. Основными клиентами сервиса будут являться студенты и сотрудники НИУ ВШЭ.

1.3. Документ, на основании которого ведется разработка

Программа выполнена в рамках темы курсовой работы, в соответствии с учебным планом подготовки бакалавров (Национальный исследовательский университет – Высшая школа экономики, факультет компьютерных наук), по направлению 09.03.04 «Программная инженерия».

1.4. Наименование темы разработки

Наименование темы разработки - «Web-приложение «Татак'»».

Наименование темы разработки на английском языке – «Web application «Татак'»».

Условное обозначение темы разработки – «Татак'».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение программы

2.1.1. Функциональное назначение

Сайт «Тамак'» предназначен для преждевременного заказа еды в предприятиях общественного питания, располагающихся в корпусах НИУ ВШЭ.

В web-приложении присутствует разделение на две роли пользователей, каждая из которых имеет следующий функционал:

Тип ресурса	Роль пользователя	
	Покупатель	Продавец
Заказ	создание чтение	создание чтение изменение удаление
Меню блюд	чтение	создание чтение изменение удаление
Личный кабинет	чтение изменение	чтение изменение
Список заказов		чтение изменение удаление

2.1.2. Эксплуатационное назначение

Пользователи веб-сервиса будут иметь возможность заранее осуществить заказ и выбрать доступное время для его получения. Данный сайт позволит сократить время ожидания для клиентов и поспособствует равномерному распределению загрузки работников сервисов питания, расположенных в зданиях НИУ ВШЭ.

Основными клиентами сервиса будут являться студенты и сотрудники НИУ ВШЭ, а также сервисы питания, расположенные в корпусах НИУ ВШЭ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2.2. Краткая характеристика области применения

Веб-сервис «Татак'» предназначен для студентов и сотрудников НИУ ВШЭ, которым необходимо заранее осуществлять заказы различной продукции организаций питания НИУ ВШЭ, а также для организаций питания НИУ ВШЭ, которые готовы внедрить в свой рабочий процесс возможность преждевременного заказа еды.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

3.1.1. Клиентская часть:

Данная часть веб-сервиса должна обеспечивать выполнение следующих задач:

3.1.1.1. В роли покупателя:

- 3.1.1.1.1. Осуществление регистрации нового пользователя;
- 3.1.1.1.2. Авторизация уже существующего пользователя по уникальной почте и подходящему паролю;
- 3.1.1.1.3. Просмотр и изменение данных об аккаунте, которые вводились при регистрации;
- 3.1.1.1.4. Просмотр списка зарегистрированных организаций питания в выбранной городе;
- 3.1.1.1.5. Просмотр меню выбранной организации питания;
- 3.1.1.1.6. Добавление конкретного блюда в корзину;
- 3.1.1.1.7. Просмотр и редактирование списка добавленных блюд в корзину;
- 3.1.1.1.8. Выбор доступного времени для заказа покупок;
- 3.1.1.1.9. Выход из авторизованного пользователя из аккаунта.

3.1.1.2. В роли предприятия:

- 3.1.1.2.1. Осуществление регистрации нового пользователя;
- 3.1.1.2.2. Авторизация уже существующего пользователя по уникальной почте и подходящему паролю;
- 3.1.1.2.3. Просмотр и изменение данных об аккаунте, которые вводились при регистрации;
- 3.1.1.2.4. Просмотр и редактирование выбранного блюда;
- 3.1.1.2.5. Добавление нового блюда в меню;
- 3.1.1.2.6. Изменение состояние блюда («есть в наличии» или «нет в наличии»);
- 3.1.1.2.7. Просмотр всех активных заказов;
- 3.1.1.2.8. Редактирование статуса активных заказов;
- 3.1.1.2.9. Добавление конкретного пользователя в черный список;
- 3.1.1.2.10. Выход из авторизованного пользователя из аккаунта.

3.1.2. Серверная часть:

Серверная часть веб-сервиса должна обеспечивать выполнение следующих задач:

- 3.1.2.1. Предоставление покупателям и предприятиям разные права доступа (права модератора и базовые права доступа);
- 3.1.2.2. Осуществление асинхронного обмена данными между серверной и клиентской частью приложения;
- 3.1.2.3. Осуществление асинхронной обработки запросов в режиме реального времени;
- 3.1.2.4. Осуществление поддержки всех функций в пунктах 3.1.1.1. и 3.1.1.2.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2. Описание алгоритма и функционирования программы

3.2.1. Архитектура приложения:

Веб-сервис был написан с использованием модели MVC (Model-View-Controller), благодаря которому логика приложения делится на три компонента: модель, представление и контроллер, которые взаимодействуют между собой, тем самым обеспечивая работу приложения. Данная модель помогает разделить приложение на блоки, в которых информация принимается от пользователя, обрабатывается и отправляется на сервер. Представления отображают интерфейс приложения, а также реагирует на действия пользователя на сайте. Полученная информация от действия пользователя отправляется в контроллер, где данные обрабатываются. Для структурирования данных и отображения их в базе данных используются модели, в которых описываются свойства и поля объекта.

Серверная часть содержит в себе три составляющие, в которых осуществляется вся работа: controller, service и repository. В controller поступают данные, которые ввел пользователь, там же они обрабатываются и передаются в service. Service является связующей точкой между controller и базой данных. Именно в этом слое происходит валидация данных. В случае прохождения валидации service обращается в repository. Repository необходим для обращения к базе данных.

Приложение писалось на языке программирования C# с использованием фреймворка ASP Net Core. Отображение интерфейса происходит благодаря представлениям (View). Разметка страницы писалась на HTML, стили заданы с помощью CSS, изменение состояния элементов страницы производится благодаря JavaScript. Для поддержания адаптивности и внедрения дополнительных стилей был использован Bootstrap для CSS и JS, а также для отправки запросов с frontend на backend использовался JQuery. Передача свойств и полей моделей в HTML код производилась благодаря макету Razor. Взаимодействие с базой данных осуществлялась через фреймворк Entity.

3.2.2. Схема базы данных приложения:

Для работы приложение использует локальную базу данных созданную в SQL Server Express LocalDB. Созданная база данных для приложения «Tamak'» содержит следующие таблицы:

3.2.2.1. User – хранится информация о пользователе

3.2.2.2. Products – хранится информация о продуктах

3.2.2.3. ShopCart – хранится информация о корзине покупок пользователя

3.2.2.4. Orders – хранится информация о покупках, занесенных в корзину покупок

Более детальное описание баз данных расположено в Приложении 3.

3.2.3. Алгоритм хеширования паролей:

Для хранения паролей под защитой используется алгоритм хеширования SHA256 из библиотеки System.Security.Cryptography.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2.4. Валидация входных данных:

Валидация данных, которые вводит пользователь, осуществляется с помощью аннотаций в моделях представления, а также благодаря осуществлению дополнительных проверок в контроллерах и на сервисах.

Валидация входных данных осуществляется при следующих действиях пользователя:

1. Вход пользователя в аккаунт:
 - a. Если пользователь не заполнит все поля при входе в аккаунт, то программа выведет сообщение об ошибке, в котором будет описано, какие именно поля не заполнены.
 - b. Если пользователь ввел неверную почту или пароль, то программа выведет сообщение о соответствующей ошибке.
2. Регистрация пользователя в приложении:
 - a. Если пользователь не заполнит все поля при регистрации нового аккаунта, то программа выведет сообщение об ошибке, в котором будет описано, какие именно поля не заполнены.
 - b. Если при заполнении полей для пароля и повторного пароля данные не совпадут, то программа выведет соответствующую ошибку.
 - c. Если пользователь попытается зарегистрировать аккаунт под уже зарегистрированной почтой, то программа выведет сообщение с соответствующей ошибкой.
3. Создание нового элемента меню:
 - a. Если пользователь не заполнит все поля при создании элемента меню, то программа выведет соответствующее сообщение об ошибке, в котором будет описано, какие именно поля не заполнены.
4. Редактирование данных в профиле:
 - a. Если пользователь оставит какое-нибудь поле в профиле пустым, то программа выведет сообщение об ошибке, в котором будет указано, какое из полей не заполнено.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.3. Описание и обоснование выбора метода организации входных и выходных данных

3.3.1. Описание метода организации входных и выходных данных

Входными данными веб-приложения являются данные, введенные пользователем при заполнении различных форм, представленных на сайте, или при выборе элементов в раскрывающемся списке.

При прохождении валидации данные отправляются на сервер в виде объекта класса, в котором заполнены поля, введенные пользователем. Если происходит операция изменения уже существующего объекта, то с помощью поля с id объекта, данные ищутся в базе данных, и редактируются. Если же создается новый элемент сайта, то вместе с ним создается и новый объект класса, который в итоге заносится в базу данных.

Взаимодействие между клиентской и серверной частью приложения осуществляется благодаря обмену данными через запрос-ответ.

С клиентской стороны поступает запрос, включающий в себя URL, который включает в себя метод, с помощью которого необходимо произвести действие и объект, с которым нужно произвести взаимодействие.

Поддерживаются следующие типы запросов:

1. GET – этот запрос позволяет получить данные;
2. DELETE – этот запрос позволяет удалить данные;
3. POST – этот запрос позволяет создать объект с определенными данными;
4. PUT – этот запрос позволяет изменить существующие данные.

Выходными данными веб-приложения являются элементы сайта, которые формируются в ответ на клиентские запросы.

Для составления ответных выходных данных осуществляется обработка запроса, а также работа с базой данных. Если данные не проходят обработку, то приложение выбрасывает соответствующую ошибку.

3.3.2. Обоснование выбора метода организации входных и выходных данных

В качестве входных и выходных данных был выбран метод, описанный в пункте 3.3.1., поскольку он является простым и удобным в реализации.

3.4. Описание и обоснование выбора состава технических и программных средств

3.4.1. Описание состава технических и программных средств

Минимальные требования для корректной работы приложения:

3.4.1.1. Клиентская часть

- 3.4.1.1.1. Стабильный доступ к сети Интернет со скоростью не менее 60 Кб/с;
- 3.4.1.1.2. Процессор, имеющий не менее двух ядер и тактовую частоту, составляющую 1.2 ГГц;
- 3.4.1.1.3. Оперативная память должна составлять не менее 2 Гб;
- 3.4.1.1.4. Свободная память на жестком диске должна составлять не менее 3 Гб;
- 3.4.1.1.5. Браузер, имеющий поддержку HTML5.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.4.1.2. Серверная часть

3.4.1.2.1. Стабильный доступ к сети Интернет со скоростью 600 Кб/с;

3.4.1.2.2. Оперативная память должна составлять не менее 2 Гб;

3.4.1.2.3. Свободная память на жестком диске должна составлять не менее 50 Гб;

3.4.1.2.4. Возможность непрерывной работы.

3.4.1.2.5. Наличие таких фреймворков, как: bootstrap, entity, JQuery.

3.4.2. Обоснование выбора технических и программных средств**3.4.2.1. Клиентская часть**

Стабильный доступ к сети Интернет необходим с минимальной скоростью в 60 Кб/с необходим для комфортной эксплуатации приложения без перебоев и провисаний.

Минимальные характеристики процессора и оперативной памяти не должны быть ниже заявленных значений для избежания задержек и некорректной работы программы.

Минимальное количество свободного места на жестком диске составляет 3 Гб для возможности хранения кеша приложения.

Поддержка HTML5 необходима для корректного отображения интерфейса приложения.

3.4.2.2. Серверная часть

Стабильный доступ к сети Интернет необходим с минимальной скоростью в 600 Кб/с для возможности корректного поддержания передачи информации между пользователями приложения.

Минимальные характеристики процессора и оперативной памяти не должны быть ниже заявленных значений для избежания задержек и некорректной работы программы.

Работа сервера не должна прерываться для поддержания круглосуточного доступа пользователей к веб-приложению.

Наличие указанных фреймворков необходима для корректной работы приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Сайт «Татак'» предназначен для преждевременного заказа еды в предприятиях общественного питания, располагающихся в корпусах НИУ ВШЭ. Основными клиентами сервиса будут являться студенты и сотрудники НИУ ВШЭ, а также другие посетители университета. Пользователи веб-сервиса будут иметь возможность заранее осуществить заказ и выбрать доступное время для его получения. Данный сайт позволит сократить время ожидания для клиентов и поспособствует равномерному распределению загрузки работников сервисов питания, расположенных в зданиях НИУ ВШЭ

4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

На данный момент у веб-приложения «Татак'» существует ряд конкурентов вроде:

1. DODO
2. Шоколадница
3. Cofix
4. ВкусВилл
5. Яндекс еда
6. СНИСКО
7. Гурманика

и многих других, но ни один из конкурентов не имеет возможности осуществлять преждевременный заказ еды в стенах корпусов НИУ ВШЭ. Внедрение сервисов конкурентов в стены НИУ ВШЭ ввело бы множество трудностей при условии, если человек, не работающий или не обучающийся в НИУ ВШЭ осуществил заказ. Данное приложение за счет условия регистрации людей, имеющих корпоративную почту данного учебного заведения, предотвращает ряд проблем, с которыми могли бы столкнуться организации.

Таким образом, преимуществами разработки являются: предоставление среды с необходимой аудиторией для предприятий общественного питания НИУ ВШЭ, а также небольшие бюджетные и ресурсные затраты на разработку.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1) ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 10) ASP.Net Documentation [Электронный ресурс] / Microsoft. URL: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0> (Дата обращения: 01.05.2023, режим доступа: свободный).
- 11) C# Documentation [Электронный ресурс] / Microsoft. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (Дата обращения: 01.05.2023, режим доступа: свободный).
- 12) Entity framework Documentation [Электронный ресурс] / Microsoft. URL: <https://learn.microsoft.com/en-us/ef/core/get-started/overview/first-app?tabs=netcore-cli> (Дата обращения: 01.05.2023, режим доступа: свободный).
- 13) Руководство по HTML5 и CSS3 [Электронный ресурс]. URL: <https://metanit.com/web/html5/> (Дата обращения: 01.05.2023, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 14) JQuery Documentation [Электронный ресурс]. URL: <https://api.jquery.com/> (Дата обращения: 01.05.2023, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2

ТЕРМИНОЛОГИЯ

1. **База данных** – структура хранения данных, которая задает определенные правила для способа хранения информации, а также методы взаимодействия данных между собой.
2. **Веб-приложение/Веб-сервис** – программное обеспечение, функционирующее в сети Интернет.
3. **Покупатель** – в контексте данного веб-приложения покупателем является пользователь, который намерен купить продукты питания в предприятиях питания НИУ ВШЭ через предоставленный веб-сервис.
4. **Предприятие** – в контексте данного веб-приложения предприятием является пользователь, который намерен продавать продукцию предприятий питания НИУ ВШЭ через предоставленный веб-сервис.
5. **Хеширование** – процесс, производящийся в хеш-функции.
6. **Хеш-функция** – это функция, которая производит шифрование входных данных путем преобразования поступившей строки в битовую строку.
7. **Bootstrap** – фреймворк языка разметки HTML, который упрощает использование стилей CSS.
8. **JQuery** – библиотека языка Java Script, которая упрощает взаимодействие Java Script и HTML, а также предоставляет удобный API для работы с AJAX.
9. **Entity** – фреймворк предназначенный для упрощенного взаимодействия с базой данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 3

ОПИСАНИЕ БАЗЫ ДАННЫХ

Таблица 3.1 – описание User

Поле	Тип данных	Назначение
Id	bigint	Идентификатор пользователя в таблице
Email	nvarchar(100)	Почтовый адрес пользователя
Password	nvarchar(MAX)	Захешированный пароль пользователя
Name	nvarchar(100)	Имя или название предприятия пользователя
City	int	Основной город пользователя
Campus	int	Основной корпус НИУ ВШЭ пользователя
Role	int	Роль пользователя
ProductId	bigint	Идентификатор продукта конкретного пользователя (нужен для формирования меню предприятий)

Таблица 3.2 – описание Products

Поле	Тип данных	Назначение
Id	bigint	Идентификатор продукта в таблице
Name	nvarchar(MAX)	Название продукта
Description	nvarchar(MAX)	Описание продукта
Img	nvarchar(MAX)	Фото продукта
Price	decimal(18, 2)	Цена продукта
Avaliable	bit	Флаг, показывающий наличие продукта
CategoryId	int	Идентификатор категории продукта
Category	int	Категория продукта
Avatar	varbinary(MAX)	Фото продукта в бинарном представлении

Таблица 3.3 – описание Orders

Поле	Тип данных	Назначение
Id	bigint	Идентификатор заказа в таблице
CardId	bigint	Идентификатор продукта в таблице
Time	int	Время заказа
ShopId	bigint	Идентификатор предприятия, в котором сделан заказ в таблице
UserId	bigint	Идентификатор пользователя, который сделал заказ в таблице
ShopCartId	bigint	Идентификатор корзины покупок в таблице

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 3.4 – описание ShopCarts

Поле	Тип данных	Назначение
UserId	bigint	Идентификатор пользователя, которому принадлежит корзина в таблице

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 4

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Таблица 4.1

Название	Назначение
AccountController	Класс, который обеспечивает связь между графическим интерфейсов и серверной связью. В нем происходит валидация данных, связанных с регистрацией и входом в аккаунт пользователя, а также отправка этих данных на сервер.
HomeController	Класс, который обеспечивает связь между графическим интерфейсов и серверной связью. В нем происходит валидация данных, связанных с данными, вводимыми на главной странице сайта, а также отправка этих данных на сервер.
ProductController	Класс, который обеспечивает связь между графическим интерфейсов и серверной связью. В нем происходит валидация данных, связанных с изменением данных продукта, а также отправка этих данных на сервер.
UserController	Класс, который обеспечивает связь между графическим интерфейсов и серверной связью. В нем происходит валидация данных, связанных с изменением данных во вкладке личного кабинета, а также отправка этих данных на сервер.
OrderController	Класс, который обеспечивает связь между графическим интерфейсов и серверной связью. В нем происходит валидация данных, связанных с заказом пользователя, а также отправка этих данных на сервер.
ShopCartController	Класс, который обеспечивает связь между графическим интерфейсов и серверной связью. В нем происходит валидация данных, связанных с добавлением продуктов в корзину, а также отправка этих данных на сервер.
Campus	Перечисление, в котором хранятся все существующие кампусы города.
Category	Перечисление, в котором хранятся всевозможные категории товаров.
City	Перечисление, в котором хранятся доступные города
Role	Перечисление, в котором хранятся роль покупателя и роль предприятия.
StatusCode	Перечисление, в котором описаны всевозможные ошибки.
EnumExtension	Расширение, позволяющее отобразить название элемента перечисления.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 4.1

HashPasswordHelper	Класс, в котором находится хеш-функция.
IBaseRepository	Интерфейс, который задает основные методы для взаимодействия с базой данных.
Product	Модель, описывающая свойства и поля продуктов.
User	Модель, описывающая свойства и поля пользователей.
ShopCart	Модель, описывающая свойства и поля корзины покупок.
ProductRepository	Класс, который взаимодействует с базой данных, используя модели продуктов.
UserRepository	Класс, который взаимодействует с базой данных, используя модели пользователей.
BaseResponse	Класс, который нужен для описания ошибок: код ошибки, описание произошедшей ошибки, а также модель, в которой произошла ошибка.
AppDBContent	Класс, в котором задаются настройки базы данных, а также начальные значения полей базы данных.
AccountService	Данный класс сервиса производит валидацию данных, пришедших из контроллера AccountController, а также приводит все данные к типу модели User. Благодаря этому классу происходит взаимодействие с базой данных.
ProductService	Данный класс сервиса производит валидацию данных, пришедших из контроллера ProductController, а также приводит все данные к типу модели Product. Благодаря этому классу происходит взаимодействие с базой данных.
OrderService	Данный класс сервиса производит валидацию данных, пришедших из контроллера OrderController, а также приводит все данные к типу модели Order. Благодаря этому классу происходит взаимодействие с базой данных.
ShopCartService	Данный класс сервиса производит валидацию данных, пришедших из контроллера ShopCartController, а также приводит все данные к типу модели ShopCart. Благодаря этому классу происходит взаимодействие с базой данных.
UserService	Данный класс сервиса производит валидацию данных, пришедших из контроллера UserController, а также приводит все данные к типу модели User. Благодаря этому классу происходит взаимодействие с базой данных.
IAccountService	Интерфейс, который задает поведение для AccountService.
IUserService	Интерфейс, который задает поведение для UserService.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 4.1

IOrderService	Интерфейс, который задает поведение для OrderService.
IShopCartService	Интерфейс, который задает поведение для ShopCartService.
HomeViewModel	Модель, в которую записываются все данные, введенные пользователем на главной странице.
LoginViewModel	Модель, в которую записываются все данные, введенные пользователем при входе в аккаунт.
ProductViewModel	Модель, в которую записываются все данные, введенные пользователем при изменении данных о продукте.
RegisterViewModel	Модель, в которую записываются все данные, введенные пользователем при регистрации.
UserViewModel	Модель, в которую записываются все данные, введенные пользователем при изменении данных о пользователе.
ShopCartViewModel	Модель, в которую записываются все данные, введенные пользователем при работе с корзиной покупок.
OrderViewModel	Модель, в которую записываются все данные, введенные пользователем при редактировании заказа.
Initializer	В данном классе происходит связывание моделей и репозитория, а также сервисов.
Program	Основной класс, в котором располагается метод Main.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 5

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ

Таблица 5.1 – Описание AccountController

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
Register		ActionResult	Необходим для отображения модели регистрации
Register	RegisterViewModel	Task<ActionResult>	Необходим для произведения валидации данных, введенных при попытке регистрации
Login		ActionResult	Необходим для отображения модели входа в аккаунт
Login	LoginViewModel	Task<ActionResult>	Необходим для произведения валидации данных, введенных при попытке входа в аккаунт
Logout		Task<ActionResult>	Необходим для выхода из аккаунта

Таблица 5.2 – Описание HomeController

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
IndexAsync		ActionResult	Возвращает все имеющиеся продукты
GetProduct	int	Task<ActionResult>	Возвращает конкретный продукт по id
Save	int	Task<ActionResult>	Сохраняет изменения у конкретного продукта по id
Save	ProductViewModel	Task<ActionResult>	Сохраняет изменения у конкретного продукта
ChangeAvaliable	ProductViewModel	Task<ActionResult>	Меняет доступность продукта (есть/нет в наличии)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 5.3 – Описание ProductController

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
GetProducts		ActionResult	Возвращает все имеющиеся продукты
GetProduct	int	Task<ActionResult>	Возвращает конкретный продукт по id
Save	int	Task<ActionResult>	Сохраняет изменения у конкретного продукта по id
Save	ProductViewModel	Task<ActionResult>	Сохраняет изменения у конкретного продукта
GetProduct	string	Task<ActionResult>	Возвращает конкретный продукт по названию
Delete	int	Task<ActionResult>	Удаляет конкретный продукт по id

Таблица 5.4 – Описание UserController

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
GetUsers		ActionResult	Возвращает всех зарегистрированных пользователей
GetUser		Task<ActionResult>	Возвращает авторизованного пользователя на данном устройстве

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 5.5 – Описание AccountService

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
Register	RegisterViewModel	Task<BaseResponse<ClaimIdentity>>	Необходим для произведения валидации данных, введенных при попытке регистрации, а также для создания нового пользователя в базе данных
Login	LoginViewModel	Task<BaseResponse<ClaimIdentity>>	Необходим для произведения валидации данных, введенных при попытке входа в аккаунт, а также поиске введенных данных в базе данных
Authenticate	User	ClaimIdentity	Авторизует пользователя

Таблица 5.6 – Описание ProductService

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
GetCategories		BaseResponse<Dictionary<long, string>>	Возвращает словарь, состоящий из кода ошибки и ее описания, а также все существующие категории
GetProduct	string	Task<BaseResponse<Dictionary<long, string>>>	Возвращает словарь, состоящий из кода ошибки и ее описания, а также продукт по его названию
GetProduct	int	Task<BaseResponse<Dictionary<long, string>>>	Возвращает словарь, состоящий из кода ошибки и ее описания, а также продукт по его id
Create	ProductViewModel	Task<IBaseResponse<Product>>	Создает продукт по модели представления
DeleteProduct	long	Task<IBaseResponse<bool>>	Удаляет продукт по id
Save	ProductViewModel	Task<IBaseResponse<Product>>	Сохраняет изменения в продукте
ChangeAvailable	ProductViewModel	Task<BaseResponse<Product>>	Меняет статус доступности продукта

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 5.7 – Описание UserService

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
GetUsers		Task<BaseResponse<IEnumerable<UserViewModel>>>	Возвращает всех зарегистрированных пользователей на сайте
GetUser	string	Task<BaseResponse<UserViewModel>>	Возвращает конкретного пользователя по email адресу

Таблица 5.8 – Описание_INITIALIZER

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
InitializeRepositories	IServiceCollection		Объединяет репозитории с соответствующими интерфейсами
GetUser	IServiceCollection		Объединяет сервисы соответствующими интерфейсами

Таблица 5.9 – Описание EnumExtension

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
GetDisplayName	Enum	String	Расширяет перечисления тем, что преобразует элемент перечисления в string, отображая заданное название элемента

Таблица 5.10 – Описание AppDbContext

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
OnModelCreating	ModelBuilder		Задаёт начальные настройки для баз данных и их поведения, инициализирует первичные значения таблиц (а именно их полей).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 5.11 – Описание Program

Методы			
Название	Аргументы	Возвращаемое значение	Назначение
Main			С данного метода начинается запуск приложения. Здесь задаются все необходимые начальные настройки веб-приложения, задаются связи и инициализируются начальные данные

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата