Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ

**ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ**

разработка ВЕБ-ПРИЛОЖЕНИЯ ОНЛАЙН-МАГАЗИН “Sushispot”

Курсовой проект по учебной дисциплине

«Технология разработки программного обеспечения»

по специальности 09.02.07 Разработчик веб и мультимедийных приложений

ПТК.КП 0902 11.000ПЗ

|  |
| --- |
| Руководители  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года |
| Студент группы 2994  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года |

Содержание

[Введение 4](#_Toc165654004)

[Актуальность: 4](#_Toc165654005)

[Объект и предмет исследования: 4](#_Toc165654006)

[Цель и задачи: 5](#_Toc165654007)

[Гипотеза: 5](#_Toc165654008)

[Для написания данной работы были использованы следующие методы: 5](#_Toc165654009)

[Научная новизна и практическая значимость исследуемой проблемы: 5](#_Toc165654010)

[1. Общая часть 7](#_Toc165654011)

[1.1 Постановка задачи 7](#_Toc165654012)

[1.1.1 Обоснование необходимости разработки 7](#_Toc165654013)

[1.1.2 Технико-математическое описание задачи 7](#_Toc165654014)

[1.1.3 Характеристика бизнес-процессов 8](#_Toc165654015)

[1.1.4 Требования к программе 9](#_Toc165654016)

[1.1.5 Цели и задачи 9](#_Toc165654017)

[1.1.6 Требования к программному обеспечению 10](#_Toc165654018)

[2. Специальная, практическая, опытно-экспериментальная часть 11](#_Toc165654019)

[2.1 Анализ задачи 11](#_Toc165654020)

[2.1.1 Информационное моделирование предметной области 11](#_Toc165654021)

[2.1.2 Проектирование пользовательского интерфейса 12](#_Toc165654022)

[2.2 Описание логической структуры 12](#_Toc165654023)

[2.2.1 Алгоритм программы 12](#_Toc165654024)

[2.2.1.1 Алгоритм страницы входа: 13](#_Toc165654025)

[2.2.1.2 Алгоритм страницы регистрации: 13](#_Toc165654026)

[2.2.1.3 Алгоритм главной страницы: 14](#_Toc165654027)

[2.2.1.4 Алгоритм страницы редактирования пользовательских данных: 15](#_Toc165654028)

[2.2.1.5 Алгоритм страницы корзины: 16](#_Toc165654029)

[2.2.2 Используемые методы 17](#_Toc165654030)

[2.3 Описание работы программы 21](#_Toc165654031)

[2.3.1 Общие сведения 21](#_Toc165654032)

[2.3.2 Функциональное назначение 21](#_Toc165654033)

[2.3.3 Связи с другими программами 22](#_Toc165654034)

[2.3.4 Входные данные 22](#_Toc165654035)

[2.3.5 Выходные данные 22](#_Toc165654036)

[2.3.6 Тестирование 23](#_Toc165654037)

[2.4 Руководство оператора 26](#_Toc165654038)

[2.4.1 Назначение программы 26](#_Toc165654039)

[2.4.2 Условия выполнения 26](#_Toc165654040)

[2.4.3 Выполнение программы 27](#_Toc165654041)

[2.4.3.1 Работа пользователя 27](#_Toc165654042)

[2.4.3.2 Работа администратора 27](#_Toc165654043)

# Введение

## Актуальность:

В современном мире онлайн-магазины становятся все более популярными, отвечая на запросы потребителей и предоставляя удобные и инновационные способы покупки и доставки продуктов и услуг. SushiSpot является примером такого онлайн-магазина, специализирующегося на заказе и доставке свежих суши.

В наше время люди все чаще предпочитают совершать покупки онлайн из-за своей занятости, удобства и широкого выбора. SushiSpot предлагает пользователям возможность заказать свои любимые виды суши прямо из дома или офиса, с минимальными усилиями и огромным выбором блюд.

Онлайн-магазины, подобные SushiSpot, становятся неотъемлемой частью современного образа жизни, обеспечивая клиентам удобство и выбор, которые раньше могли быть недоступны. Вместе с тем, такие платформы создают новые возможности для предпринимателей в области общественного питания, позволяя им достигать широкой аудитории и развивать свой бизнес в цифровой среде.

Таким образом, SushiSpot является примером успешного онлайн-магазина, который отвечает на запросы современных потребителей и предлагает инновационные решения для заказа и доставки суши, делая процесс покупки более удобным и доступным для всех.

## Объект и предмет исследования:

Объектом исследования данной работы является онлайн-магазин SushiSpot, специализирующийся на доставке суши. Предметом исследования являются различные аспекты функционирования SushiSpot, такие как его техническая структура, пользовательский опыт, маркетинговые стратегии, безопасность данных и перспективы развития.

## Цель и задачи:

Цель данной курсовой работы - разработка сайта SushiSpot на базе фреймворка Laravel. Для достижения этой цели были сформулированы следующие задачи:

* Изучить основные принципы работы фреймворка Laravel.
* Разработать возможность для пользователей регистрацию, вход на сайте, а также добавление продуктов в корзину.

## Гипотеза:

Гипотеза работы заключается в том, что создание онлайн-магазина SushiSpot на основе фреймворка Laravel является эффективным и востребованным решением для предоставления услуги заказа и доставки суши. Предполагается, что SushiSpot предоставляет удобный и доступный способ для пользователей заказывать свои любимые блюда суши, обеспечивая широкий выбор, простоту использования и высокое качество обслуживания. Гипотеза также предполагает, что SushiSpot будет успешно конкурировать на рынке общественного питания, привлекая и удерживая клиентов за счет своей функциональности, качества продукции и эффективных маркетинговых стратегий.

## Для написания данной работы были использованы следующие методы:

* Изучение документаций
* Анализ существующих решений в данной области
* Проектирование и разработка сайта
* Тестирование и анализ результатов

## Научная новизна и практическая значимость исследуемой проблемы:

Научная новизна данной работы заключается в разработке онлайн-магазина SushiSpot на основе фреймворка Laravel. Практическая значимость этого проекта заключается в создании удобной и доступной платформы для заказа и доставки суши, которая предоставляет пользователям широкий выбор блюд и обеспечивает простоту процесса заказа. Создание SushiSpot позволит пользователям удовлетворить свои потребности во вкусной и качественной еде, а также предоставит новые возможности для развития бизнеса в сфере общественного питания.

1. Общая часть
   1. Постановка задачи

### Обоснование необходимости разработки

В современном мире все больше людей нуждается в удобных и эффективных онлайн-платформах для заказа и доставки пищи, особенно в контексте растущей популярности услуг доставки еды. С учетом этой тенденции, разработка онлайн-магазина SushiSpot становится необходимой для удовлетворения потребностей клиентов в качественной еде, предоставленной с удобством и доступностью.

Создание SushiSpot обосновано рядом факторов:

* Увеличение спроса на услуги онлайн-заказа и доставки пищи: с каждым годом все больше людей предпочитают заказывать еду онлайн из-за своей удобства и экономии времени. SushiSpot предоставит клиентам возможность легко выбирать и заказывать блюда из широкого ассортимента суши.
* Растущий интерес к японской кухне: суши и роллы становятся все более популярными во многих странах мира, и создание специализированного онлайн-магазина суши может привлечь широкую аудиторию.
* Возможность предоставить лучший выбор и качество: SushiSpot позволит клиентам выбирать из разнообразного меню суши, обеспечивая высокое качество и свежесть продуктов.

Таким образом, создание SushiSpot будет отвечать растущему спросу на онлайн-покупки и доставку пищи, предоставляя клиентам возможность наслаждаться высококачественными блюдами суши с минимальными усилиями.

* + 1. Технико-математическое описание задачи

Для реализации проекта SushiSpot было решено использовать фреймворк Laravel, который обеспечивает эффективное развертывание и управление веб-приложениями. Laravel предоставляет широкий набор инструментов для работы с маршрутизацией, контроллерами, моделями и представлениями, что обеспечивает гибкость и расширяемость разрабатываемого приложения.

Для хранения данных о меню, заказах и пользователях в проекте SushiSpot будет использоваться реляционная база данных MySQL. Это позволит эффективно хранить и управлять информацией о продуктах, клиентах и заказах.

Для разработки пользовательского интерфейса SushiSpot будут использоваться языки HTML, CSS и JavaScript. Использование JavaScript позволит реализовать интерактивные элементы, такие как динамическое обновление содержимого страницы без перезагрузки и валидация форм.

Для удобства разработки и отладки кода проекта SushiSpot будет использоваться интегрированная среда разработки Visual Studio, которая предоставляет мощные инструменты для написания, отладки и тестирования кода на языке PHP.

Таким образом, проект SushiSpot на фреймворке Laravel предполагает использование современных технологий разработки веб-приложений для обеспечения высокой производительности, удобства использования и функциональности онлайн-магазина суши.

Мой интернет-магазин будет иметь следующие функциональные возможности:

* регистрация и авторизация пользователей;
* добавление товаров в корзину;
* редактирование персональных данных.
  + 1. Характеристика бизнес-процессов

Основной целью проекта SushiSpot является предоставление клиентам удобной и эффективной платформы для заказа суши. Для достижения этой цели необходимо описать основные бизнес-процессы, которые включают в себя следующие шаги:

* регистрация на сайте: Клиент может зарегистрироваться на сайте, чтобы в последующие разы, когда он будет заказывать, его адрес подгружался из базы данных;
* подбор и заказ блюд: Клиент выбирает блюда из представленного меню на сайте SushiSpot, добавляет их в корзину и оформляет заказ.
  + 1. Требования к программе

Для разработки проекта SushiSpot выбран фреймворк Laravel и реляционная база данных MySQL. Для работы программы необходим компьютер с операционной системой Windows и установленным PHP, а также PHPMyAdmin для работы с базой данных.

Для фронтенд-разработки используются языки HTML, CSS и JavaScript. Отладка и разработка ведутся с помощью среды разработки Visual Studio.

* + 1. Цели и задачи

Целью данного проекта является разработка онлайн-магазина SushiSpot с использованием фреймворка Laravel и базы данных MySQL. Для достижения данной цели были поставлены следующие задачи:

* разработать дизайн веб-сайта, который будет привлекательным и удобным для пользователей;
* создать архитектуру веб-приложения с использованием фреймворка Laravel, обеспечивающую эффективную работу и масштабируемость приложения;
* разработать базу данных MySQL для хранения информации о меню, заказах и пользователях;
* реализовать функциональность сайта, позволяющую пользователям просматривать меню, делать заказы, оплачивать их и отслеживать статус доставки;
* провести тестирование веб-приложения и устранить возможные ошибки и недочеты.

Цель проекта SushiSpot заключается в создании удобного и функционального онлайн-магазина для заказа суши с доставкой. Для достижения этой цели необходимо разработать все необходимые компоненты веб-приложения, начиная от дизайна и архитектуры, заканчивая реализацией функциональности и тестированием.

* + 1. Требования к программному обеспечению

Требования к программному обеспечению для разработки онлайн-магазина SushiSpot:

1. Дизайн сайта должен быть удобным и привлекательным для пользователей, с акцентом на простоту навигации и визуальное представление блюд.
2. Сайт должен обеспечивать возможность пользователям просматривать меню, добавлять блюда в корзину и оформлять заказы.
3. Для пользователей должна быть доступна функция регистрации и создания личных профилей, где они могут просматривать историю заказов, управлять своими данными и сохранять адреса доставки.
4. Программное обеспечение должно обеспечивать высокую производительность и отзывчивость сайта, минимизируя время загрузки страниц и обработки запросов.
5. Специальная, практическая, опытно-экспериментальная часть
   1. Анализ задачи

### Информационное моделирование предметной области

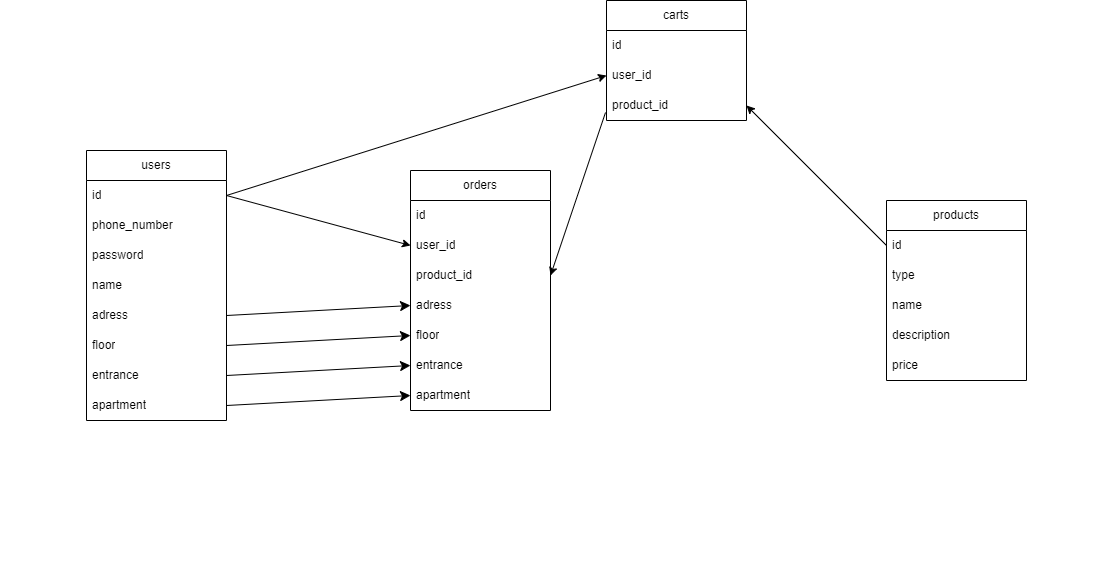
SushiSpot – это онлайн-магазин для заказа блюд японской кухни. Для хранения данных используется PHPMyAdmin. ER-модель предметной области SushiSpot состоит из следующих таблиц:

Users: содержит информацию о зарегистрированных пользователях, включая их идентификатор, имя пользователя, номер телефона, время и дата регистрации пользователя, пароль, а также адрес.

Orders: содержит информацию о заказах, включая их идентификатор, идентификатор пользователя, который сделал заказ, идентификатор(ы) продуктов в заказе и адрес доставки.

Carts: содержит идентификатор корзины, идентификатор пользователя, собиравшего корзину и индентификатор(ы) продуктов в корзине.

Products: содержит идентификатор продукта, тип (категория), название, описание и цену.

ER-диаграмма:  


* + 1. Проектирование пользовательского интерфейса

Для обеспечения удобства использования и эффективного взаимодействия пользователей с онлайн-магазином SushiSpot необходимо разработать интуитивно понятный пользовательский интерфейс. В этом разделе мы рассмотрим основные экраны и элементы пользовательского интерфейса, которые должны присутствовать на каждом из них.

Основные экраны:

Главная страница сайта:

* Хедер с логотипом, кнопками входа/регистрации;
* Меню навигации;
* Приветственным сообщением;
* Список товаров.

Страница корзины:

* Хедер с логотипом, кнопками «Редактировать профиль» и «Корзина»;;
* Список добавленных в корзину товаров;
* Общая стоимость заказа;
* Форма оформления заказа с полями для ввода контактной информации и ввода адреса доставки.

Страница редактирования профиля:

* Хедер с логотипом, кнопками «Редактировать профиль» и «Корзина»;
* Информация о пользователе (имя, контактная информация и адрес пользователя);
* Возможность редактировать профиль;
  1. Описание логической структуры
     1. Алгоритм программы

Пользователь при заходе на сайт попадает на главную страницу, где можно увидеть приветствие и меню онлайн-магазина, а также пройти авторизацию. После того, как пользователь перейдёт на страницу авторизации для него будет представлена форма с номером телефона и паролем.

* + - 1. Алгоритм страницы входа:

1. Создается контроллер AuthController, который содержит два метода: index() для отображения страницы входа и auth() для аутентификации пользователя.
2. Метод index() возвращает вид страницы входа auth.login.
3. Метод auth() получает данные из запроса, проводит их валидацию с использованием правил из класса LoginRequest.
4. Если данные не прошли валидацию, возвращается обратно на страницу входа с сообщением об ошибке.
5. Если данные валидны, производится попытка аутентификации пользователя с использованием метода attempt() класса Auth.
6. Если аутентификация не прошла успешно, возвращается обратно на страницу входа с сообщением об ошибке.
7. Если аутентификация прошла успешно, сессия пользователя перезагружается, пользователь аутентифицируется и перенаправляется на главную страницу сайта.
8. HTML-разметка страницы входа содержит форму, в которой пользователь вводит свой phone\_number и пароль для аутентификации.
9. Используется CSRF-защита с помощью директивы @csrf.
10. Если происходят ошибки валидации, они выводятся пользователю.
11. При отправке формы пользователь отправляется на маршрут login, который обрабатывается методом auth() контроллера AuthController.
    * + 1. Алгоритм страницы регистрации:
12. Создается контроллер RegisterController, который содержит методы index() для отображения страницы регистрации и reg() для обработки запроса на регистрацию.
13. Метод index() возвращает вид страницы регистрации auth.register.
14. При отправке формы регистрации данные передаются методом POST на маршрут register.
15. В методе reg() происходит валидация полученных данных из запроса с использованием класса RegisterRequest.
16. Если данные проходят валидацию, создается новый экземпляр модели User с помощью метода create(). В этом экземпляре указываются поля name, phone\_number и password, взятые из данных запроса. Пароль хэшируется с помощью метода Hash::make().
17. После успешного создания пользователя происходит редирект на страницу входа login.
    * + 1. Алгоритм главной страницы:
18. DOCTYPE и HTML-корневой элемент: объявляется DOCTYPE для указания типа документа и открывается HTML-элемент, задается его языковый атрибут.
19. Мета-теги: устанавливаются метатеги для указания кодировки символов и масштабирования страницы для различных устройств.
20. Ссылки на ресурсы: происходит предварительное подключение к серверам Google Fonts для загрузки шрифта "Baloo Bhaina 2", а также подключение каскадной таблицы стилей (CSS) и иконки сайта.
21. Заголовок страницы: устанавливается заголовок страницы - "SushiSpot".
22. Тело страницы (body):

* Включение шапки и подвала страницы: используется директива @include для включения компонентов шапки и подвала страницы из внешних файлов.
* Основное изображение с информацией: размещается основное изображение с заголовком и описанием.
* Навигация по категориям продуктов: Представлена навигационная панель с ссылками на различные категории продуктов: "Наборы", "Роллы", "Запечённые роллы", "Суши", "Пицца", "Супы", "Напитки", "Десерты".
* Рекомендации по продуктам: Представлены различные категории продуктов с использованием цикла foreach для вывода информации о продуктах в каждой категории. В каждой категории отображается название категории, а также карточки продуктов с изображением, названием, ценой и возможностью добавления в корзину (если пользователь аутентифицирован).
  + - 1. Алгоритм страницы редактирования пользовательских данных:

1. DOCTYPE и HTML-корневой элемент: объявляется DOCTYPE для указания типа документа и открывается HTML-элемент, задается его языковый атрибут.
2. Мета-теги: устанавливаются метатеги для указания кодировки символов и масштабирования страницы для различных устройств.
3. Ссылки на ресурсы: происходит предварительное подключение к серверам Google Fonts для загрузки шрифта "Baloo Bhaina 2", а также подключение каскадной таблицы стилей (CSS) и иконки сайта.
4. Заголовок страницы: устанавливается заголовок страницы - "SushiSpot".
5. Тело страницы (body):

* Включение шапки и подвала страницы: используется директива @include для включения компонента шапки из внешнего файла.
* Форма редактирования данных пользователя: Представлена форма, отправляющая данные на маршрут обновления пользователя. Форма содержит поля для ввода номера телефона, адреса, квартиры/офиса, подъезда и этажа. Поля уже заполнены текущими данными пользователя.
* Кнопка подтверждения изменений: Пользователь может подтвердить внесенные изменения, нажав на кнопку "Подтвердить".
* Кнопка выхода из учетной записи: Представлена ссылка, при нажатии на которую осуществляется выход из учетной записи. Для этого используется JavaScript-обработчик, который предотвращает стандартное действие ссылки и отправляет запрос на выход из учетной записи через скрытую форму.
* Скрытая форма для выхода из учетной записи: присутствует форма, которая отправляет запрос на выход из учетной записи на маршрут "logout".
  + - 1. Алгоритм страницы корзины:

1. DOCTYPE и HTML-корневой элемент: объявляется DOCTYPE для указания типа документа и открывается HTML-элемент. Задается языковый атрибут для корректного отображения контента.
2. Мета-теги: устанавливаются метатеги для указания кодировки символов и масштабирования страницы для различных устройств.
3. Ссылки на ресурсы: происходит предварительное подключение к серверам Google Fonts для загрузки шрифта "Baloo Bhaina 2". Подключается каскадная таблица стилей (CSS) и иконка сайта для установки вкладки в браузере.
4. Заголовок страницы: устанавливается заголовок страницы - "SushiSpot".
5. Тело страницы (body):

* Включение компонента шапки и подвала страницы с помощью директивы @include.
* Представлена форма редактирования данных пользователя, отправляющая данные на маршрут обновления пользователя.
* Форма содержит поля для ввода номера телефона, адреса, квартиры/офиса, подъезда и этажа, заполненные текущими данными пользователя.
* Присутствует кнопка подтверждения изменений для сохранения внесенных изменений.
* Пользователь может выйти из учетной записи, нажав на кнопку "Выход". Для этого используется JavaScript-обработчик, который предотвращает стандартное действие ссылки и отправляет запрос на выход из учетной записи через скрытую форму.
* Присутствует скрытая форма для выхода из учетной записи, отправляющая запрос на маршрут "logout".
  + 1. Используемые методы

UserController:

1. edit($id):

* Этот метод отвечает за отображение формы редактирования данных пользователя.
* Получает пользователя с указанным идентификатором ($id) с помощью метода findOrFail() модели User.
* Передает полученного пользователя в представление "public.edit" с помощью метода view(), используя компактный массив.

1. update(Request $request, $id):

* Этот метод отвечает за обновление данных пользователя.
* Получает данные из формы, отправленной пользователем, с помощью объекта Request ($request).
* Находит пользователя с указанным идентификатором ($id) с помощью метода findOrFail() модели User.
* Обновляет данные пользователя с помощью метода update(), передавая полученные данные ($request->all()).
* После успешного обновления перенаправляет пользователя на страницу редактирования своих данных с сообщением об успешном обновлении.

ProductController:

1. show():

* Этот метод отвечает за отображение всех продуктов.
* Получает все продукты из модели Product с помощью метода all().
* Передает полученные продукты в представление "home" с помощью метода view(), используя компактный массив.
* Представление "home" может содержать список продуктов для отображения пользователю.

HeaderController:

1. show():

* Этот метод отображает информацию в шапке страницы.
* Получает текущего пользователя с помощью метода Auth::user(), который возвращает аутентифицированного пользователя или null, если пользователь не аутентифицирован.
* Передает информацию о пользователе в представление "layouts.app" с помощью метода view(), используя компактный массив.
* Представление "layouts.app" может содержать компоненты, такие как меню навигации, профиль пользователя или другую информацию, зависящую от состояния аутентификации.

AuthController:

1. index():

* Этот метод отображает форму входа.
* Возвращает представление для страницы входа.

1. auth(LoginRequest $request):

* Этот метод обрабатывает запрос на аутентификацию пользователя.
* Принимает экземпляр класса LoginRequest $request, который содержит данные, введенные пользователем в форме входа.
* Валидирует данные запроса с помощью правил, определенных в классе LoginRequest.
* Пытается аутентифицировать пользователя с использованием метода Auth::attempt(), передавая ему данные из запроса.
* Если аутентификация успешна, сессия обновляется и пользователь аутентифицируется.
* После успешной аутентификации пользователь перенаправляется на главную страницу.
* В случае неудачной аутентификации пользователю предлагается вернуться на страницу входа с сообщением об ошибке.

1. logout():

* Этот метод отвечает за выход пользователя из системы.
* Вызывает метод Auth::logout() для выхода пользователя из сеанса.
* После выхода пользователь перенаправляется на главную страницу.

RegisterController:

1. index():

* Этот метод отображает форму регистрации.
* Возвращает представление для страницы регистрации.

1. reg(RegisterRequest $request):

* Этот метод обрабатывает данные, отправленные при регистрации нового пользователя.
* Принимает экземпляр класса RegisterRequest $request, который содержит данные, введенные пользователем в форме регистрации.
* Валидирует данные запроса с помощью правил, определенных в классе RegisterRequest.
* Создает новую запись пользователя в базе данных, используя полученные данные.
* Пароль пользователя хешируется с помощью функции Hash::make(), чтобы обеспечить безопасное хранение.
* После успешной регистрации пользователя перенаправляет его на страницу входа.

CartController:

1. addToCart(Request $request, Product $product):

* Этот метод добавляет товар в корзину покупок.
* Принимает объект запроса $request и модель продукта $product в качестве аргументов.
* Получает идентификатор пользователя из аутентифицированной сессии.
* Создает запись корзины в базе данных, связывая текущего пользователя с выбранным продуктом.
* После добавления товара в корзину перенаправляет пользователя на предыдущую страницу.

1. showListAndAdress($id):

* Этот метод отображает список товаров в корзине и информацию об адресе доставки.
* Принимает идентификатор пользователя $id в качестве аргумента.
* Получает информацию о текущем пользователе из аутентифицированной сессии.
* Получает список товаров в корзине для указанного пользователя и подгружает связанные данные о продуктах.
* Возвращает представление, отображающее список товаров и адрес доставки.

1. remove($id):

* Этот метод удаляет товар из корзины покупок.
* Принимает идентификатор товара в корзине $id в качестве аргумента.
* Находит запись корзины по указанному идентификатору.
* Если запись найдена, она удаляется из базы данных.
* После удаления товара из корзины перенаправляет пользователя на страницу корзины с сообщением об успешном удалении или ошибкой, если товар не был найден в корзине.
  1. Описание работы программы

### Общие сведения

Название программы: SushiSpot

Краткое описание: SushiSpot — это веб-приложение для заказа еды, специализирующееся на японской кухне. Пользователи могут зарегистрироваться, войти в систему, просматривать меню, оформлять заказы и управлять своими данными.

* + 1. Функциональное назначение

SushiSpot предоставляет следующие функциональности:

* Регистрация и аутентификация пользователей: пользователи могут создать свою учетную запись в системе и войти в нее, используя свои учетные данные.
* Просмотр меню и продуктов: пользователи могут просматривать доступные блюда и напитки на главной странице сайта.
* Редактирование пользовательских данных: зарегистрированные пользователи могут редактировать свои данные, такие как номер телефона и адрес доставки.
* Оформление заказа: пользователи могут выбирать продукты из меню, добавлять их в корзину и оформлять заказ, указывая адрес доставки и другие детали.
* Просмотр рекомендаций: на главной странице отображаются рекомендуемые продукты для пользователя.
  + 1. Связи с другими программами

SushiSpot использует следующие связи с другими программами:

* PHPMyAdmin: SushiSpot использует PHPMyAdmin в качестве административного интерфейса для управления базой данных MySQL. PHPMyAdmin предоставляет инструменты для создания, изменения и удаления таблиц, выполнения запросов к базе данных и администрирования пользователей.
  + 1. Входные данные

Данные, которые подаются на вход в программу SushiSpot, включают:

* Регистрационные данные: Адрес электронной почты и пароль, введенные пользователем при создании учетной записи. Эти данные используются для аутентификации пользователя и обеспечения безопасного доступа к его учетной записи.
* Данные о продуктах: Название, описание, тип (наборы, роллы, напитки и т. д.) и цена каждого продукта. Эти данные используются для отображения меню и предоставления информации о доступных блюдах пользователю.
* Данные пользователя: Номер телефона, адрес доставки, информация о квартире/офисе, подъезде и этаже. Пользователь может вводить и обновлять эти данные для оформления заказа и обеспечения точной доставки.
  + 1. Выходные данные

Данные, которые выводятся из программы SushiSpot, включают:

* Информация о продуктах: Название, тип и цена каждого продукта отображается для пользователей на главной странице и страницах с категориями. Это позволяет пользователям просматривать доступные блюда и выбирать то, что им хочется заказать.
* Состояние сессии пользователя: Пользователь может видеть свое текущее состояние сеанса, например, авторизован ли он или нет. Это отображается через соответствующие элементы интерфейса, например, кнопки "Войти" или "Выйти", чтобы пользователь мог управлять своими действиями в приложении.
  + 1. Тестирование

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер тест-кейса | Описание | Шаги действий | Ожидаемый результат | Фактический результат | Статус тест-кейса |
| 1. | Тестирование модуля регистрации | 1. Зайти на главную страницу 2. Нажать на кнопку зарегистрироваться 3. Вписать свои данные для регистрации (имя, номер телефона и пароль) 4. Нажать на кнопку зарегистрироваться | Создастся поле в базе данных, и пользователь будет перенаправлен на страницу для входа | Поле в базе данных создалось, а пользователя перенаправило на страницу входа | Пройден |
| 2. | Тестирования модуля входа | 1. После регистрации пользователь перенаправлен на страницу для входа 2. Ввести данные для входа (номер телефона и пароль) 3. Пользователь будет перенаправлен на главную страницу | Пользователь войдёт в сессию и будет перенаправлен на главную страницу, где в хедере появится вместо «Зарегистрироваться» и «Войти» «Редактировать профиль» и «Корзина» | Пользователь успешно вошёл в сессию и был перенаправлен на главную страницу, где в хедере появились кнопки, доступные только зарегистрированным пользователям | Пройден |
| 3. | Тестирование добавления товара в корзину | 1. На главной странице, после входа в сессию пользователю доступно добавление товара в корзину 2. Нажать на кнопку «Добавить» под понравившимся товаром | После нажатия кнопки должно создаться поле в базе данных с корзиной пользователя и товаром | Поле в базе данных создалось и id товара было передано | Пройден |
| 4. | Проверка вывода товаров на странице корзины | 1. Товар добавлен 2. Перейти на страницу корзины | На странице корзины должен выводиться список добавленных товаров пользователем | Список товаров в корзине выводится | Пройден |
| 5. | Проверка работы страницы для редактирования профиля | 1. Войти в аккаунт 2. Зайти на страницу для редактирования профиля 3. Заменить или внести свои данные (номер телефона, имя, адрес) 4. Нажать на кнопку «Подтвердить» | После ввода данных и нажатия на кнопку «Подтвердить», в базе данных должны замениться или добавиться строки данных о пользователе | Изменения успешно были переданы в базу данных | Пройден |
| 6. | Проверка выхода пользователя из сессии | 1. Войти в аккаунт 2. Зайти в редактирование профиля 3. Нажать на кнопку выйти | После нажатия на кнопку «Выход», пользователя должно выкинуть из сессии и перенаправить на главную страницу | Пользователь вышел из сессии и был перенаправлен на главную страницу | Пройден |

* + 1. Вызов и загрузка

Программа SushiSpot запускается и загружается следующим образом:

* Установка зависимостей: перед запуском программы необходимо установить PHP-библиотеки напрямую с их официальных веб-сайтов или других источников, предоставляющих исходный код.
* Запуск локального сервера: после загрузки всех необходимых библиотек можно запустить локальный сервер для разработки и просмотра приложения. Используется встроенная команда Laravel «php artisan serve». Локальный сервер будет на определенном порте (обычно 8000) и предоставлять доступ к приложению через браузер.
  1. Руководство оператора
     1. Назначение программы

SushiSpot — это веб-приложение, предназначенное для заказа и доставки суши, роллов и других блюд японской кухни. Оно предназначено для широкого круга пользователей, включая любителей японской кухни, компании, которые хотят организовать обеды для сотрудников, и просто людей, которые предпочитают заказывать еду через интернет. SushiSpot обеспечивает удобство и простоту в заказе, а также широкий выбор блюд для удовлетворения различных вкусовых предпочтений.

* + 1. Условия выполнения

Для использования веб-сайта SushiSpot требуется компьютер с операционной системой, поддерживающей Visual Studio Code и предустановленный PHP.

* + 1. Выполнение программы
       1. Работа пользователя

Пользователь SushiSpot может выполнять следующие действия:

* Регистрироваться в системе, предоставляя необходимые учетные данные, такие как номер телефона, имя и пароль;
* Входить в систему с использованием зарегистрированного аккаунта;
* Просматривать меню ресторана с описанием блюд, ценами и фотографиями;
* Добавлять выбранные блюда в корзину для последующего оформления заказа;
* Оформлять заказ, указывая необходимую информацию для доставки;
* Редактировать личную информацию, такую как адрес доставки.
  + - 1. Работа администратора

Администратор программы SushiSpot имеет расширенные права доступа и может выполнять следующие действия:

* Управлять пользователями, включая регистрацию и удаление аккаунтов;
* Управлять продуктами, включая редактирование и удаление продуктов;