Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и ФИО руководителя)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**КУРСОВАЯ РАБОТА**

|  |  |
| --- | --- |
| Тема | Разработка веб-приложения «Магазин растений» |
| По МДК 09.01 | Проектирование и разработка веб-приложений |
| Выполнил | Бас В. М. |
| Курс, группа | 4 курс, группа ИС-4 |
| Специальность | 09.02.07 Информационные системы и программирование |
| Научный руководитель | Мартыненко В.А. |

г. Белореченск

2024 г.

Содержание

[ВВЕДЕНИЕ 3](#_Toc161185609)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 4](#_Toc161185610)

[1.1 Анализ существующих решений 4](#_Toc161185611)

[1.2 Проектирование базы данных 5](#_Toc161185612)

[1.3 Цели и задачи выполняемой системы 5](#_Toc161185613)

[1.4 Метод реализации процесса проектирования 6](#_Toc161185614)

[1.5 Обоснование проектных решений 6](#_Toc161185615)

[2. ПРОЕКТНАЯ ЧАСТЬ 14](#_Toc161185616)

[2.1 Техническое задание 14](#_Toc161185617)

[2.2 Программное обеспечение информационной системы 14](#_Toc161185618)

[2.3 Руководство пользователя 15](#_Toc161185619)

[2.4 Руководство администратора 17](#_Toc161185620)

[ЗАКЛЮЧЕНИЕ 19](#_Toc161185621)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#_Toc161185622)

[ПРИЛОЖЕНИЕ 21](#_Toc161185623)

# ВВЕДЕНИЕ

В современном мире онлайн магазины растений становятся неотъемлемой частью культурного и декоративного потребления. Они не только предоставляют доступ к широкому спектру растений из разных стран и климатических зон, но и создают комфортную среду для выбора и покупки растений в удобное время и в любом месте, где есть интернет-соединение.

Объект исследование – целью данного исследования веб-приложение «Магазин растений», которое было разработано с использованием PHP, Bootstrap, CSS, HTML.

Цель исследования – целью данного исследования является разработка и анализ веб-приложения «Магазин растений», которое позволит упростить процесс поиска и покупи растений.

Методы исследования – для достижения поставленной цели исследования были использованы следующие методы:

Анализ существующих решений - был проведен анализ существующих веб-приложений, связанных с интернет магазнов для покупки и продажи растений, для выявления их преимуществ и недостатков.

Разработка прототипа - был разработан прототип веб-приложения, который был использован на целевой аудитории для получения данных о его эффективности.

Разработка и тестирование - была проведена разработка и тестирование приложения с использованием различных технологий и инструментов.

Анализ результатов - был проведен анализ результатов, полученных в ходе исследования, для выявления улучшения и возможного направлений дальнейшего развития.

# 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ существующих решений

Был проведен анализ существующих конкурентов. Это помогло понять, какие проблемы нужно решить в настоящее время. Для анализа были выбраны следующие конкуренты:

Flowwow (https://flowwow.com/):

* Описание: Flowwow является одним из крупнейших онлайн магазинов растений в России и странах СНГ. Он предоставляет доступ к обширному каталогу растений, цветов и т.д.   
  Преимущества:
* Обширный каталог растений и цветов.
* Разнообразные функции поиска и фильтрации.
* Поддержка различных платформ и устройств.
* Система рейтингов и отзывов пользователей.  
  Недостатки:
* Отсутствие возможности скачивания контента.

Букет столицы ([https://getbuket.ru/](https://www.ivi.ru/)):

* Описание: Букет столицы - это популярный магазин растений с широким выбором товаров, включая ростения, цветы, доступный для использования в России.  
  Преимущества:
* Большой выбор контента, включая российские и зарубежные растения.
* Поддержка мобильных приложений для удобного мониторинга на различных устройствах.
* Не всегда актуальный каталог новинок.

Procvetok (https://procvetok.ru/):  
Описание: Procvetok - это платформа для онлайн покупок растений и цветов, предоставляющая доступ к эксклюзивному контенту, включая товары для дома, сада и двора.  
Преимущества:

* Эксклюзивный контент, включая , включая товары для дома, сада и двора.
* Качественное фотограции ростений.
* Отсутствие рекламы (зависит от модели подписки).

Недостатки:

* Ограниченный выбор товара по сравнению с более крупными платформами.
* Возможные ограничения по доступности товара в некоторых регионах.

## 1.2 Проектирование базы данных

В предоставленной работе приведена спроектированная база данных для положения. Это включает в себя определение структуры базы данных, выбор подходящего типа данных и структуры данных, а также проектирование отношений между таблицами

## 1.3 Цели и задачи выполняемой системы

Основные цели системы:

* Разработать удобное веб-приложение для создания страниц для покупки и продажи растений.
* Реализовать функциональность регистрации и авторизации пользователей.

Задачами являются:

* Достичь поставленных целей;
* Реализовать функционал покупки и продажи растений.

## 1.4 Метод реализации процесса проектирования

При проектировании веб-приложения, которое использует базу данных для вывода данных, можно использовать следующий подход:

1) Анализ требований:

* Определение какие данные необходимо получить из базы данных.
* Определить как эти данные должны быть представлены в пользовательском интерфейсе.

2) Интеграция с базой данных:

* Использование PHP для создания запросов к базе данных и получения данных.
* Обработка ответов от базы данных.

3) Отображение данных:

* Используется HTML, CSS и Bootstrap для создания использования пользовательского интерфейса.
* Выделение полученных данных на страницу с использованием PHP.

## 1.5 Обоснование проектных решений

Использование определенного стека технологий – можно обосновать использование PHP, Bootstrap, CSS и HTML для разработки веб-приложения. это обеспечивает кросс-браузерную совместимость, легкость в изучении и использовании, а также широкую допустимость библиотек и фреймворков.

Использование определённой архитектуры – MVC(Model-View-Controller) для организации кода и его масштабируемости.

Система MVC архитектуры, представленных на рисунке 1.1:

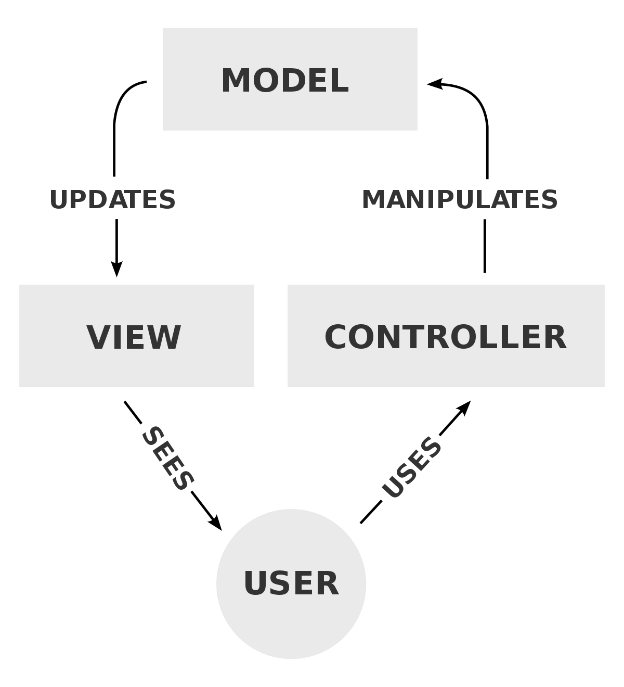


Рисунок 1 – MVC архитектура

Выбор определенного способа хранения данных – MySQL, для хранения информации о пользователях, товарах и заказов.

Требования к системе представлены в таблице 1.1.

Таблица 1.1 – Требования к системе учебного курса

|  |  |  |
| --- | --- | --- |
| Атрибуты | Метки | Требования |
| 1 | 2 | 3 |
| Удобство | | |
| Простота использования | Среднее время, необходимое  пользователю, чтобы найти  необходимый элемент в программе | Поиск отдельного элемента интерфейса не должен занимать более 1 секунды |
| Привлекательность | Соответствие интерфейса  требованиям | 99% элементов должны быть работоспособны для  пользователя |
| Обучаемость | Показатель, затрачиваемый  пользователями на обучение | Пользователь должен изучить работу ПО за первые 3-5 минут |

Продолжение таблицы 1.1 – Требования к системе учебного курса

|  |  |  |
| --- | --- | --- |
| Производительность | | |
| Временная эффективность | Время выполнения компонента программы | - Приложение должно  обслуживать любую функцию не дольше 1 сек  (не включая задержки в сети); |
| Эффективность использования ресурсов | Объемы ресурсов требуемых для выполнения задач | - Постоянная память,  используемая программой – не более 10МБ;  - Приложение должно потреблять не более 3МБ  памяти на каждый неактивный сеанс с  пользователем; |
| 1 | 2 | 3 |
| Переносимость | | |
| Удобство установки | Легкость установки | Необходима установка:  apache, mysql, admin php |
| Способность к сосуществованию | Способность ПО сосуществовать с другими  программами в общем окружении | Программа не должна  занимать большие объемы памяти устройства (максимум 3МБ) |
| Удобство замены другого ПО данным | Совместимость структур данных | - Должна поддерживать  внедрение новых версий;  - Система должна быть применима вместо других  программных систем для решения тех же задач в определенном окружении. |

Требования к численности и квалификации персонала системы и режиму его работы:

В системе существуют 2 группы пользователей: Пользователь и Администратор. Пользователи выполняют функцию: просмотра товара, и продажи товара.

Администратор имеет возможность использовать полный функционал CRUD (Создание\Просмотра\Редактирование\Удаление) различных данных из БД

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне пользователя.

В таблице 1.2 представлены требования к надежности.

Таблица 1.2 – Требования к надёжности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Метрики | | Требования | |
| 1 | | 2 | | 3 |
| Надёжность | | | | |
| Устойчивость к отказу | Коэффициент аварийных | | - Не менее 90% ошибок | |
|  | отказов | | в программе должны  обрабатываться без  экстренного завершения;  - Обеспечение бесперебойного питание  активного сетевого  оборудования; | |
| Способность к восстановлению | Отношение количества числа успешного восстановление работы системы к ее аварийному завершению за определённый промежуток времени | | - Восстановление  работоспособности  системы не должно превышать 5 минут; | |
| Уровень зрелости | Среднее время работы без сбоев | | - Система должна  работать 24 часа в сутки;  - Система должна оповещать  - Система должна оповещать пользователя при некорректных действиях в системе;  - Данные, которые вводит пользователь, должны сохраняться. | |

Средняя доступность приложения должна составлять не менее 99%

Среднее время между сбоями — это среднее время, за которое компонент или модуль может выполнять свои функции без перерыва.

Измеряется от начала работы до момента следующего сбоя.

Среднее, временя работы без сбоев должно составлять не менее 1500 часов.

В таблице 1.3 представлен перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Таблица 1.3 – Перечень и критерии отказов для каждой функции

|  |  |
| --- | --- |
| Функции | Критерии отказа |
| 1 | 2 |
| Отображение данных | Отсутствие соединения с базой данных |
| Добавление | Отсутствие соединения с базой данных |
| Редактирование | Отсутствие соединения с базой данных |
| Удаление | Отсутствие соединения с базой данных |

Требования к функциям web-приложения, выполняемым программным обеспечением

Имя: Вход пользователя

Описание: Пользователь заходит на главную страницу

Действующий субъект: Пользователь

Предусловие: пользователь зашел на сайт для покупки растений Сценарий:

* Пользователь зарегистрировался на сайте
* Авторизировался под своими учетными данными
* Зашел на главную страницу
* Нашел нужный товар

Расширения: в случае не верно введенных данных при авторизации будет возникать ошибка.

Перед вводом системы в действие необходимо:

– интернет-соединение;

– обеспечить каждого специалиста доступом к web-приложению;

– иметь данные для доступа к админ-панели;

– база данных MySQL.

Разрабатываемая система требует в своей архитектуре наличие БД-сервера со свободной реляционной системой управления – MySQL.

База данных информационной системы должна состоять из следующих связанных таблиц:

– basket– таблица, содержащая данные о товарах в корзине.

– delivery– таблица, содержащая информацию о доставках.

* product– таблица содержащая информацию о товарах
* user– таблица, содержащая информацию о пользователях.

Данные о пользователях представлены в таблице basket 1.4.

Таблица 1.4 – Данные о корзине

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 11 | Идентификатор товара в корзине | Первичный ключ |
| user\_id | int | 11 | Пользователь | NOT NULL |
| product\_id | int | 11 | Товар | NOT NULL |

Данные о доставках должны размещаться в таблице delivery 1.5.

Таблица 1.5 – Данные о доставках.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | Int | 11 | Идентификатор | Первичный ключ |
| user\_id | Int | 11 | Пользователь | NOT NULL |
| product\_id | Int | 11 | Товар | NOT NULL |
| qtyBuy | Int | 11 | Количество товара | NOT NULL |
| fPrice | Int | 11 | Общая цена | NOT NULL |
| dost | text |  | Доставка или самовывоз | NOT NULL |

Данные о товарах должны размещаться в таблице product 1.6.

Таблица 1.6. – Данные о товарах.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| Id | Int | 11 | Индентификатор товара | Первичный ключ |
| title | Varchar | 255 | Название товара | NOT NULL |
| price | Int | 11 | Ценна | NOT NULL |
| description | varchar | 255 | Описание товара | NOT NULL |
| qty | Int | 11 | Количество | NOT NULL |
| user\_id | Int | 11 | Пользователь | NOT NULL |
| avatar | Varchar | 255 | Превью товара | NOT NULL |

Данные о пользователях должны размещаться в таблице user 1.7.

Таблица 1.7. – Таблица ползователей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| id | INT | 255 | Идентификатор | Превичный ключ |
| username | VARCHAR | 255 | Логин | NOT NULL |
| password | VARCHAR | 255 | Пароль | NOT NULL |
| email | VARCHAR | 255 | Майл | NOT NULL |
| number | VARCHAR | 255 | Номер телефона | NOT NULL |
| avatar | VARCHAR | 255 | Аватар | NOT NULL |
| role | int | 11 | роль | NOT NULL |

На рисунке 2 представлено окно создания товара.

Рисунок 2. – окно создания товара.

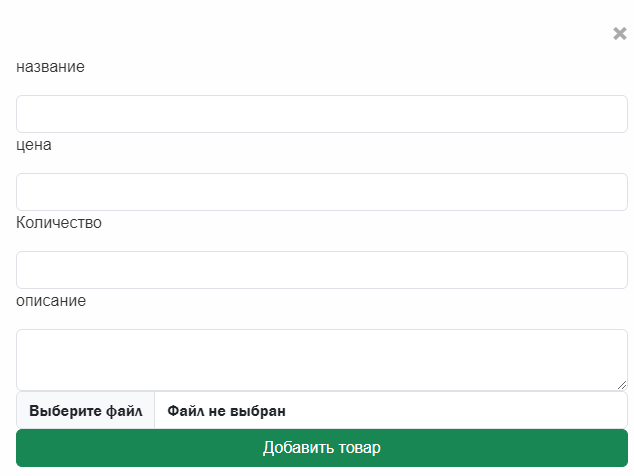


Рисунок 2. - в данном окне показывается форма создания товара.

На рисунке 3 представлен профиль пользователя.

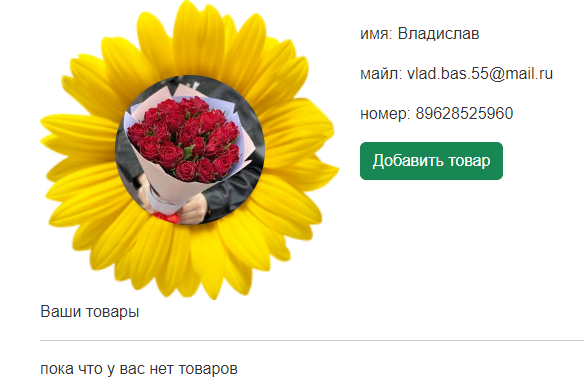


Рисунок 3. - профиль пользователя.

На рисунке 4 – представлена страница подробнее о товаре.

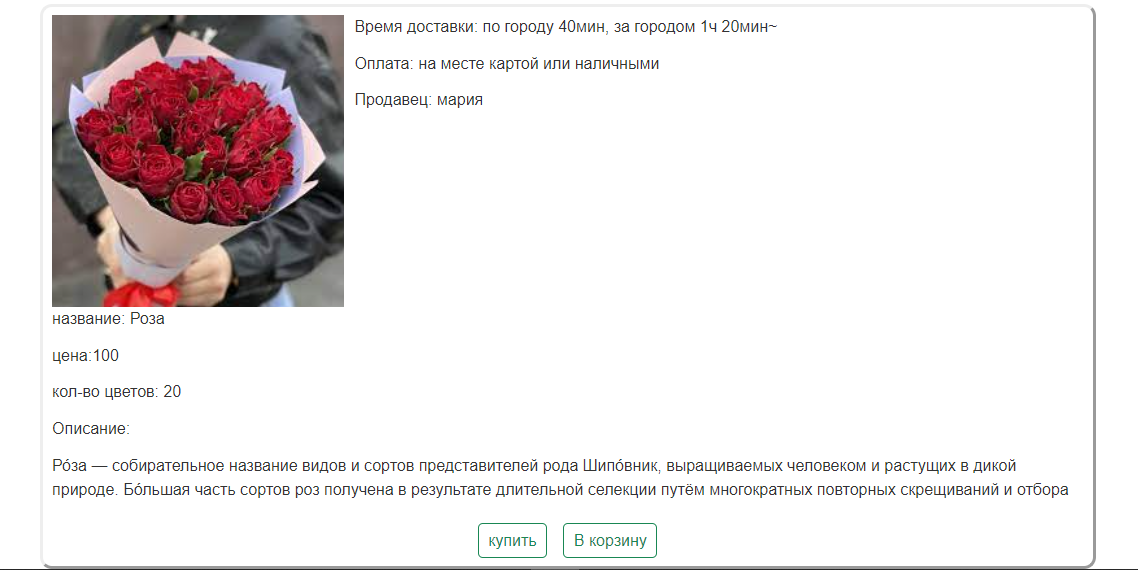


Рисунок 4. - страница подробнее о товаре.

На рисунке 5 – представлено страница покупки товара.



Рисунок 5 – страница покупки товара.

# 2. ПРОЕКТНАЯ ЧАСТЬ

## 2.1 Техническое задание

Административная часть сайта должна предоставлять возможность добавления, редактирования и удаления данных из базы данных

Веб-сайт должен соответствовать требованиям:

* единый стиль оформления;
* интуитивно понятное назначение элементов интерфейса;
* отображение на экране только тех возможностей, которые доступны конкретному пользователю;
* отображение на экране только необходимой для решения текущей прикладной задачи информации;

HTML— это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео.

Для реализации интерактивных элементов сервероной части должны использоваться языки php, javascript.

Объем flash-заставки не должен превышать 300 Kb.

CSS — язык описания внешнего вида HTML-документа. Это одна из базовых технологий в современном интернете. Практически ни один сайт не обходится без CSS, поэтому HTML и CSS действуют в единой связке.

Каскадные таблицы стилей работают с HTML, но это совершенно другой язык. HTML структурирует документ и упорядочивает информацию, а CSS взаимодействует с браузером, чтобы придать документу оформление.

## 2.2 Программное обеспечение информационной системы

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

Необходимое программное обеспечение для персонального компьютера:

- Операционная система: Windows/Linux/macOS

Сайт должен быть доступен для полнофункционального просмотра с помощью следующих браузеров:

– Opera 6.0 и выше;

– Google;

– Yandex.

JavaScript - предназначен для написания сценариев для активных HTML-страниц. Язык JavaScript не имеет никакого отношения к языку Java. Java разработан фирмой SUN. JavaScript - фирмой Netscape Communication Corporation. Первоначальное название - LiveScript. После завоевания языком Java всемирной известности LiveScript из коммерческих соображений переименовали в JavaScript.

2.3 Руководство пользователя

Авторизация и Регистрация: для доступа к профилю пользователя, продажи и прокупки товара необходимо зарегистрироваться и авторизироваться.

Если у пользователя нет аккаунта, он может создать его, нажав «Зарегистрироваться» и заполнив необходимую информацию, включая имя, email, пароль.

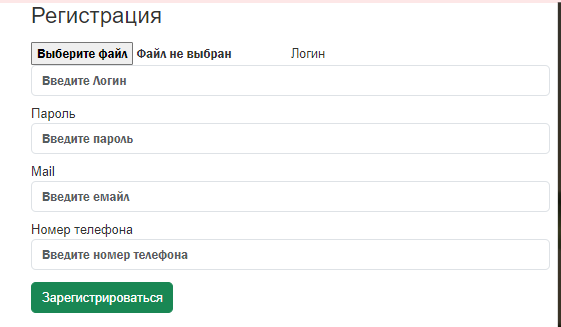


Рисунок 6. – окно регистрации.

После успешной авторизации или регистрации вы попадете на главную страницу.

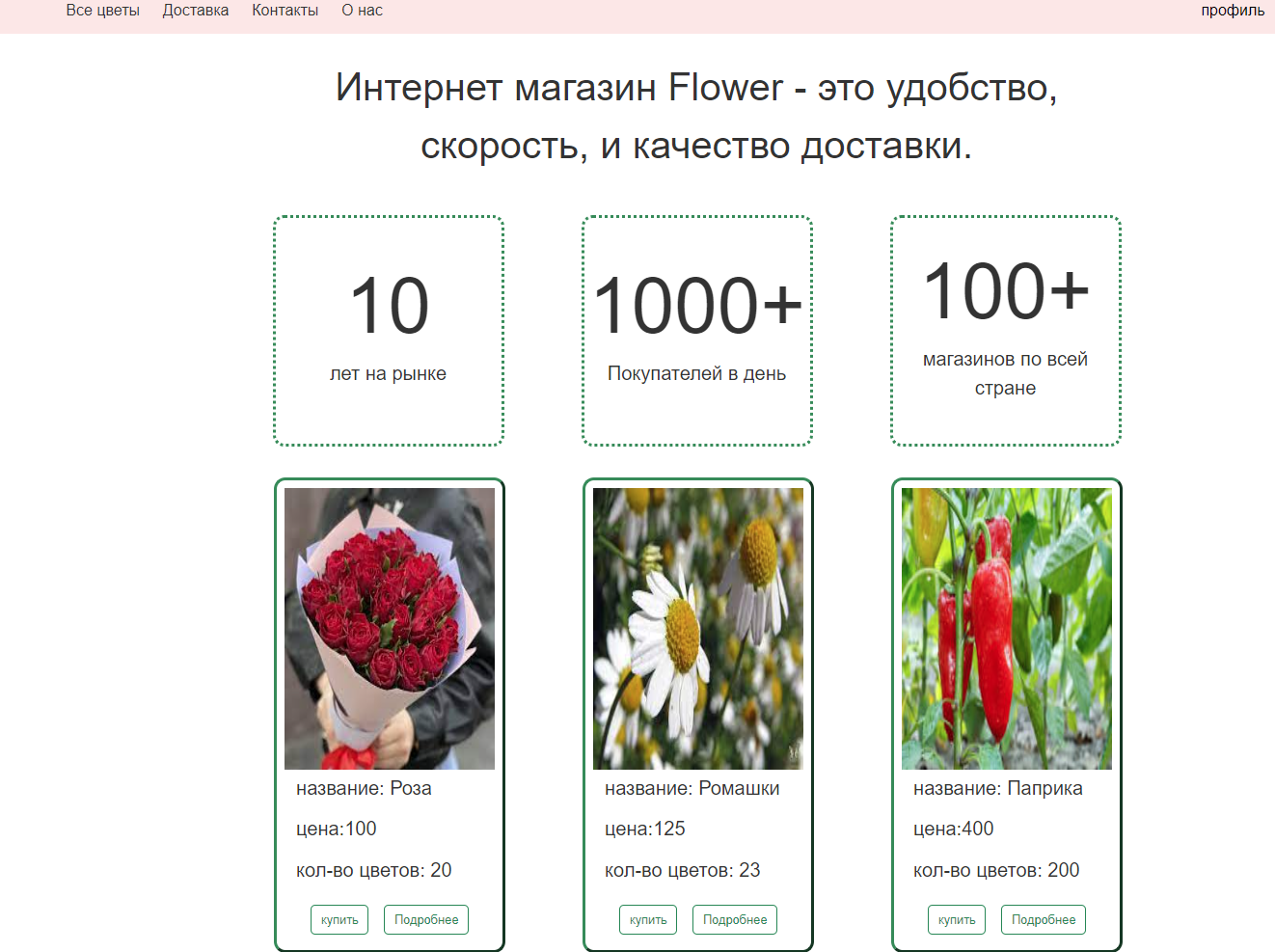


Рисунок 7. – Главная страница.

Профиль:

В личном кабинете есть возможность редактирования профиля, при нажатии на “Добавить” появляется модальное окно для заполнения данных.

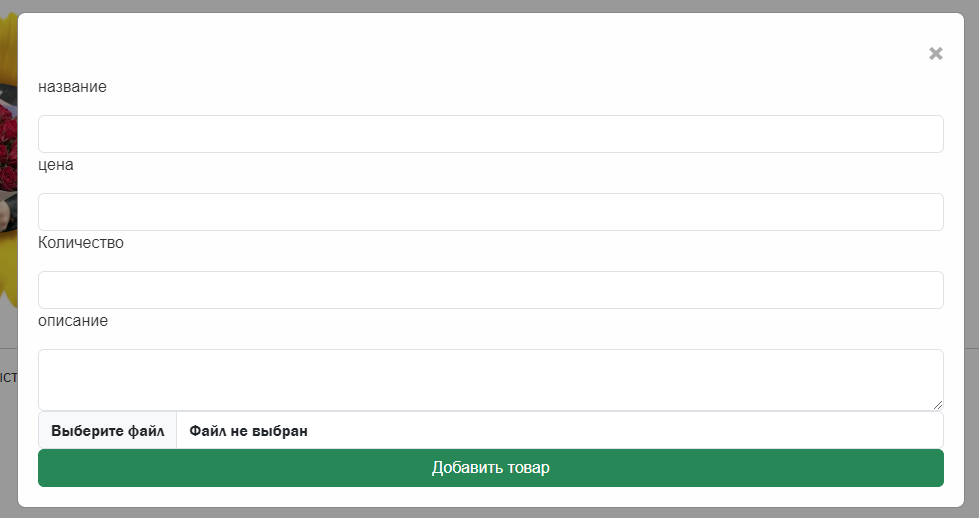


Рисунок 8. – окно добовление товара.

Выход из Аккаунта:

В правой части экрана будет кнопка “Выход” при нажатии на нее происходит ди авторизация из аккаунта.

2.4 Руководство администратора

Авторизация:

Как администратор, вы авторизуетесь так же, как и обычный пользователь, нажимая на «Авторизация» и вводя свой логин и пароль.

Роль администратора определена в базе данных как «admin».

Доступ к Админ-Панели:

После успешной авторизации под Администратором у вас появляется доступ к “Админ-панели”, там выводится вся информация о пользователях, и товарах, так же есть возможность удаления.

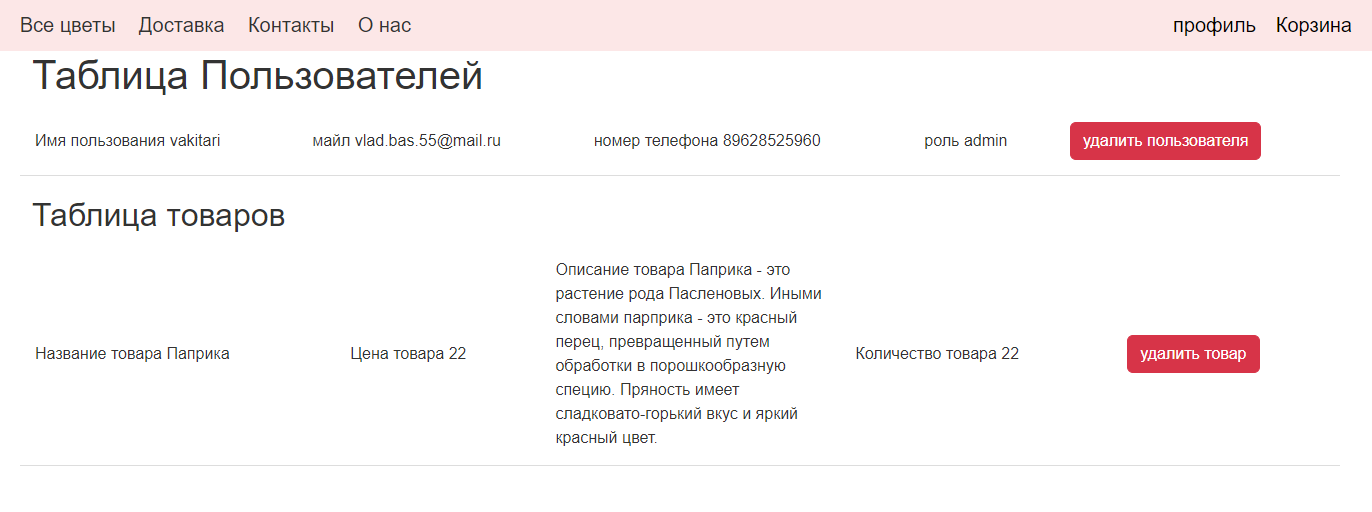


Рисунок 9. – Страница админ панели.

Нажмите на этот пункт, чтобы перейти в административную панель.

Обзор Пользователей:

В админ-панели отображается таблица со всеми пользователями, включая их данные.

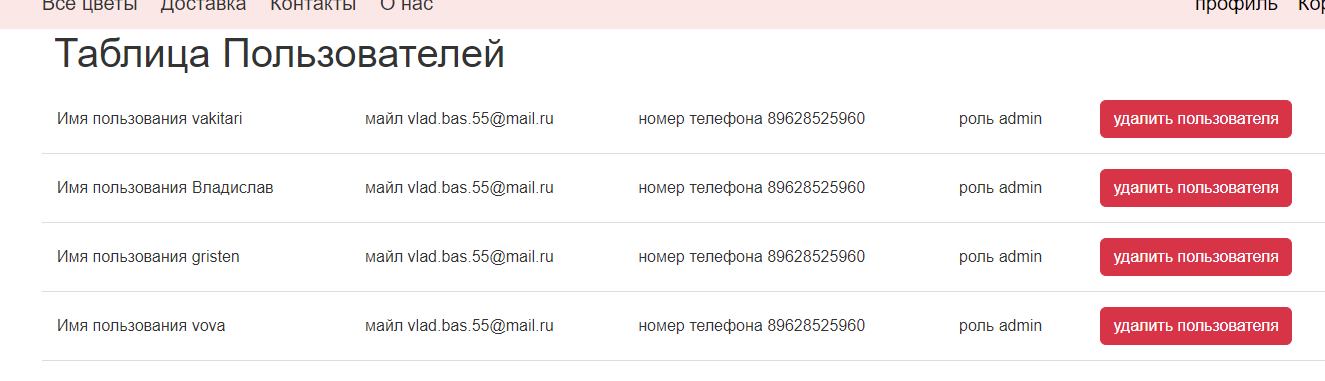


Рисунок 10. – Таблица пользователей.

Каждая строка содержит кнопку «Удалить» для управления данными пользователя.

Функции Администрирования:

Удалить: Кнопка «Удалить» удаляет профиль пользователя.

Безопасность: Защита от входа обычных пользователей

# ЗАКЛЮЧЕНИЕ

При разработке приложения были достигнуты цели учебной практики, а именно: Разработан удобный интерфейс для сайта по выбору растений, Реализована функциональность регистрации и авторизации пользователей, Цель работы достигнута, а именно создано веб-приложения для сайта выбора растений. Особое внимание уделено удобству ввода и редактирования и просмотра данных.

В результате выполнения учебной практики, был создан онлайн магазин растений. Основная задача сайта — удобный сайт для выбора интересующих людей растений.

Это программное обеспечение включает в себя:

– язык программирования PHP, HTML, Bootstrap, CSS, JS;

– реляционная система управления БД MySQL;

Всё информационное наполнение сайта находится в БД.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. JavaScript, Веб-дизайн, Веб-программирование, Зарубежная компьютерная литература, Интернет Программирование, Создание сайтов Элизабет Робсон 200c.
2. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2020. - 464 c.
3. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт. - Москва: Наука, 2019. - 480 c.
4. Дронов, В. JavaScript в Web-дизайне / В. Дронов. - М.: СПб: БХВ, 2019. - 880 c.
5. Дронов, В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2019. - 480 c.
6. Дронов, В.А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов / В.А. Дронов. - М.: БХВ-Петербург, 2019. - 820 c.
7. Дронов, Владимир JavaScript и AJAX в Web-дизайне / Владимир Дронов. - Москва: Высшая школа, 2019. - 736 c.
8. Дронов, Владимир PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов / Владимир Дронов. - М.: "БХВ-Петербург", 2019. - 448 c.
9. Колисниченко, Д. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Колисниченко. - М.: БХВ-Петербург, 2021. - 560 c.
10. Колисниченко, Д.Н. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д.Н. Колисниченко. - М.: БХВ-Петербург, 2021. - 364 c.

# ПРИЛОЖЕНИЕ

Страница которая отвечает за header на каждой странице и отвечает за стили на всех страницах – start.php

<?php

/\*\*

\* @var \App\Kernel\View\View $view

\* @var \App\Kernel\Session\Session $session

\*/

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@400..700&display=swap" rel="stylesheet">

<title>Document</title>

<style>

body,

html {

height: 100%;

font-family: sans-serif;

color: #333333;

}

input,

textarea {

font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

}

table {

width: 100%;

border-collapse: collapse;

}

th,

td {

padding: 15px;

text-align: left;

border-bottom: 1px solid #ddd;

}

th {

background-color: #f2f2f2;

}

.panel-body label{

margin-top: 10px;

}

.about-page{

margin-top: 50px;

display: flex;

align-items: center;

flex-direction: column;

}

.contact-info {

margin-top: 30px;

}

.contact-info p {

margin-bottom: 10px;

}

.about-page p{

font-size: 25px;

display: flex;

flex-direction: column;

align-items: center;

text-align: center;

}

.about-page div p{

text-align: center;

width: 70%;

}

.about-page div{

display: flex;

flex-direction: column;

align-items: center;

font-size: 20px;

margin-top: 200px;

}

.about-page .hel-p{

display: flex;

flex-direction: column;

align-items: center;

font-size: 30px;

}

.card-pay form div {

display: flex;

width: 100%;

}

.card-pay form {

display: flex;

flex-wrap: wrap;

}

.time-delivery {

margin-top: 20px;

}

#but-form-control {

border: 2px double;

}

#test p {

margin-top: 20px;

}

.payment .btn {

border: revert;

border-color: green;

}

.payment {

width: 40%;

margin-top: 20px;

font-size: 20px;

}

.pay-page {

display: flex;

justify-content: space-between;

}

.prof-row {

display: flex;

margin-top: 1%;

}

.prof-inf {

margin-left: 20px;

margin-top: 2%;

}

.prof-container {

position: relative;

width: 300px;

height: 300px;

padding: 0;

margin: 0;

}

.overlay {

position: absolute;

top: 0%;

left: 1;

width: 100%;

height: 100%;

}

a{

color: #333; /\* Цвет текста \*/

text-decoration: none; /\* Убираем подчеркивание \*/

border-radius: 5px; /\* Скругление углов \*/

transition: background-color 0.3s ease; /\* Плавное изменение цвета фона \*/

}

a:hover{

background-color: #e0e0e0;

}

img{

height: 365px;

}

.overlay img {

position: absolute;

top: 75px;

left: 75px

.a\_buy {

display: flex;

align-items: flex-end

}

.buyInf {

margin-left: 1%;

display: flex;

flex-direction: column;

font-size: 20px;

}

.buyMini {

display: flex;

}

.buy-div {

border: outset;

border-radius: 15px;

margin-top: 15px;

padding-top: 10px;

font-size: 20px;

}

.buy-div a {

font-size: 20px;

}

.pError a {

margin-left: 7px;

}

.pError {

display: flex;

justify-content: center;

margin-top: 5%;

font-size: 25px;

}

.div-nav {

display: flex;

align-items: center;

.div-nav p {

padding: 0px;

margin: 0px;

padding: 1%;

}

.navbar-brand {

font-size: 50px;

font-family: "Dancing Script", cursive;

}

.prod-home {

display: inline-block;

margin-top: 20px;

margin-left: 50px;

display: flex;

align-items: flex-start;

flex-wrap: wrap;

}

.p\_b {

display: flex;

justify-content: center;

align-items: center

}

.p\_b a {

margin: 10px;

}

.pay-page .prod-div {

height: fit-content;

.prod-div {

padding: 10px;

border: outset;

border-color: seagreen;

width: 300px;

margin-left: 50px;

margin-right: 50px;

margin-top: 20px;

display: inline-block;

flex-direction: column;

border-radius: 15px;

.modal {

display: none; /\* Скрываем модальное окно по умолчанию \*/

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0,0,0,0.4);

}

.modal-content {

background-color: #fefefe;

margin: 9% auto;

padding: 20px;

border: 1px solid #888;

width: 50%;

}

.close {

color: #aaa;

float: right;

font-size: 28px;

font-weight: bold;

.cose:hover,

.close:focus {

color: black;

text-decoration: none;

cursor: pointer;

}

.prod-div img {

width: 100%;

}

.pp {

margin-bottom: 0rem;

}

.prod-div p {

margin-top: 5px;

font-size: 25px;

padding-left: 15px;

}

.home-inf {

display: flex;

justify-content: center;

margin-top: 20px;

}

.inf-div {

border: dotted;

border-color: seagreen;

text-align: center;

height: 300px;

width: 300px;

margin-left: 50px;

margin-right: 50px;

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

border-radius: 15px;

}

.inf-div h2 {

font-size: 100px;

}

.inf-div p {

font-size: 25px;

}

.d-flex p {

margin-top: 10px;

font-size: 20px

}

.home-div {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

height: 200px;

border-radius: 15px;

margin-top: 15px;

}

.home-div p {

text-align: center;

font-size: 50px;

width: 80%;

}

.text {

display: flex;

flex-direction: row;

}

.imgReg img {

width: 100%;

height: 100%;

font-family: "Dancing Script", cursive;

}

.imgReg::before {

content: "Регистрация позволит нам отслеживать ваши покупки и лучше разбираться в ваших вкусах, спасибо что выбираете Flower. ";

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

color: white;

font-size: 24px;

text-align: center;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.7);

box-sizing: border-box;

padding-top: 50%;

padding-left: 10%;

padding-right: 10%;

}

.imgReg {

position: relative;

margin-left: 10px;

}

.imgLog img {

width: 100%;

height: 100%;

}

.imgLog::before {

content: "Авторизуйтесь на нашем сайте для того чтобы иметь возможность продавать свои товары и мы бы могли подбирать цветы специально для вас ";

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

color: white;

font-size: 24px;

text-align: center;

width: 100%;

height: 100%;

background: rgba(0, 0, 0, 0.7);

box-sizing: border-box;

padding-top: 50%;

padding-left: 10%;

padding-right: 10%;

}

.imgLog {

position: relative;

margin-left: 10px;

}

</style>

</head>

<body>

<nav class="navbar bg-body-tertiary">

<div class="container-fluid">

<a class="navbar-brand" href="/bas/home">Flower</a>

<form class="d-flex" role="search">

<?php if (!$session->has('user\_id')) {?>

<p><a href="/bas/register" class="link-success link-offset-2 link-underline-opacity-25 link-underline-opacity-100-hover me-2">Зарегистрироваться</a></p>

<p><a href="/bas/login" class="link-success link-offset-2 link-underline-opacity-25 link-underline-opacity-100-hover me-2">Войти</a></p>

<?php } else { ?> <a href="/bas/logout" style="margin-right: 100px" class="link-success link-offset-2 link-underline-opacity-25 link-underline-opacity-100-hover me-2">выйти</a> <?php } ?>

</form>

</div>

</nav>

<div class="div-nav" style="background-color: #FDE7E7 ;">

<div style="width: 100%; display: flex; margin-left:15%;font-size:20px">

<p><a href="/bas/home">Все цветы</a></p>

<?php if ($session->has('user\_id')) {?>

<p><a href="/bas/delivery">Доставка</a></p>

<?php } ?>

<p><a href="/bas/contact">Контакты</a></p>

<p><a href="/bas/about">О нас</a></p>

</div>

<?php

if ($session->has('user\_id')) {?>

<label style="margin-right:15%;font-size:20px;display:flex;">

<a href="/bas/profile" style="color: black;"> профиль</a>

<a href="/bas/basket" style="margin-left: 20px;color: black;"> Корзина</a>

</label><?php }?>

</div>

</div>

<div class="container">

<div class="row">

Страница покупки товара – buyFlow.php

<?php

/\*\*

\* @var \App\Kernel\View\View $view

\* @var \App\Kernel\Session\Session $session

\* @var \App\Kernel\Auth\Auth $auth

\* @var \App\Kernel\DataBase\DataBase $db

\*/

$user = $auth->user();

$product;

$view->component('start');

?>

<div class="pay-page">

<div class="payment">

<form action="/bas/deliveryBuy" method="post">

<p>Доставка или заберете сами?</p>

<button id="button1" type="button" class="btn">Самовывоз</button>

<button id="button2" type="button" class="btn">Доставка</button>

<div id="test" style="display: none;">

<p>Тогда укажите адрес доставки:</p>

<div class="input-group input-group-sm mb-3">

<span class="input-group-text" id="inputGroup-sizing-sm">Адрес</span>

<input style="border-color: cadetblue;" name="dost" type="text" class="form-control" aria-label="Sizing example input" aria-describedby="inputGroup-sizing-sm">

</div>

</div>

<div class="time-delivery">

<p>Время доставки: 20:21</p>

</div>

<input type="text" placeholder="Количество" name="qtyBuy" class="input-group" oninput="calculatePrice()">

<div class="card-pay">

<p>Заполните данные карты</p>

<input type="text" id="myInput" placeholder="Номер карты" class="input-group">

<input type="text" placeholder="Имя владельца" class="input-group">

<div>

<input type="text" id="myInput3" placeholder="Срок действия" class="input-group">

<input type="text" id="myInput4" placeholder="CVV/CVC" class="input-group">

</div>

</div>

<p style="margin-top: 20px;">доставка: бесплатная</p>

<input type="hidden" name="finP">

Итого:<p id="finPrice"></p>

<input type="hidden" name="product\_id" value="<?php echo $product['id']; ?>" >

<input type="hidden" name="user\_id" value="<?php echo $user->id(); ?>">

<button type="submit" >Оплатить</button>

</form>

</div>

<div class="prod-div">

<img src="<?php echo $storage->url($product['avatar']); ?>" alt="12312">

<p>название: <?php echo $product['title'] ?></p>

<p>цена:<?php echo $product['price'] ?></p>

<p>кол-во цветов: <?php echo $product['qty'] ?></p>

</div>

</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script>

$(document).ready(function() {

$("#button1").click(function() {

var test = document.getElementById("test");

var button1 = document.getElementById("button1");

var button2 = document.getElementById("button2");

button1.style.background = (button1.style.backgroundColor === 'green') ? 'white' : 'green';

button1.style.color = (button1.style.color === 'white') ? 'black' : 'white';

button2.style.background = (button2.style.backgroundColor === 'green') ? 'white' : 'white';

button2.style.color = (button2.style.color === 'white') ? 'black' : 'black';

test.style.display = (test.style.display === 'block') ? 'none' : 'none';

});

$("#button2").click(function() {

var test = document.getElementById("test");

var button1 = document.getElementById("button1");

var button2 = document.getElementById("button2");

button2.style.background = (button2.style.backgroundColor === 'green') ? 'white' : 'green';

button2.style.color = (button2.style.color === 'white') ? 'black' : 'white';

button1.style.background = (button1.style.backgroundColor === 'green') ? 'white' : 'white';

button1.style.color = (button1.style.color === 'white') ? 'black' : 'black';

test.style.display = (test.style.display === 'none') ? 'block' : 'none';

})

document.getElementById('myInput').addEventListener('input', function() {

if (this.value.length == 4 || this.value.length == 9 || this.value.length == 14) {

this.value += ' ';

}

if (this.value.length > 19) {

this.value = this.value.slice(0, 19);

}

});

document.getElementById('myInput').addEventListener('keydown', function(event) {

if (event.key === 'Backspace') {

var input = this;

if (this.value.length == 5 || this.value.length == 10 || this.value.length == 15) {

input.value = input.value.slice(0, -1);

}

}

});

document.getElementById('myInput3').addEventListener('input', function() {

if (this.value.length == 2) {

this.value += '/';

}

if (this.value.length > 5) {

this.value = this.value.slice(0, 5);

}

});

document.getElementById('myInput3').addEventListener('keydown', function(event) {

if (event.key === 'Backspace') {

var input = this;

if (this.value.length == 3 ) {

input.value = input.value.slice(0, -1);

}

}

});

document.getElementById('myInput4').addEventListener('input', function() {

if (this.value.length > 3) {

this.value = this.value.slice(0, 3);

}

});

})

function calculatePrice() {

var qtyBuy = document.querySelector('[name="qtyBuy"]').value;

var finPrice = document.querySelector('#finPrice');

var finP = document.querySelector('[name="finP"]');

if (qtyBuy > <?php echo $product['qty'] ?>) {

alert('Число не может быть больше ' + <?php echo $product['qty'] ?>);

document.querySelector('[name="qtyBuy"]').value = <?php echo $product['qty'] ?>;

}

if (qtyBuy === '') {

finP.value = '';

finPrice.textContent = '';

} else {

finP.value = qtyBuy \* <?php echo $product['price'] ?>;

finPrice.textContent = qtyBuy \* <?php echo $product['price'] ?>;

}

}

</script>

<?php

$view->component('end')

Страница на которой отображатся все данные пользователя «Профиль» - profile.php

<?php

/\*\*

\* @var \App\Kernel\View\View $view

\* @var \App\Kernel\Session\Session $session

\* @var \App\Kernel\Auth\Auth $auth

\* @var \App\Kernel\DataBase\DataBase $db

\* @var \App\Kernel\Storage\Storage $storage

\*/

$user = $auth->user();

$view->component('start')

?>

<div class="prof-row">

<div class="prof-container">

<img style="width: 100%; height: 100%;" src="views/components/img/sunflower.png" alt="12312">

<div class="overlay">

<img style="width: 50%; height: 50%;border-radius:100px;" src="<?php echo $storage->url($user->avatar()); ?>" alt="12312">

</div>

</div>

<div class="prof-inf">

<p>имя: <?php echo $user->username(); ?></p>

<p>майл: <?php echo $user->email(); ?></p>

<p>номер: <?php echo $user->number(); ?></p>

<button type="button" id="registerBut" class="btn btn-success">Добавить товар</button>

</div>

</div>

<div class="prof-product">

Товары марии

<hr>

<?php

if (!$db->first('product', ['user\_id' => $user->id()])) { ?>

<p>пока что мария не выставляла товар</p>

<?php } else {

foreach ($db->get('product', ['user\_id' => $user->id()]) as $value) { ?>

<div><img src="<?php echo $storage->url($value['avatar']); ?>" style="height: 50px;margin-bottom:20px" alt="123"><a href="/bas/buyFlow/?id=<?php echo $value['id'];

?>"> Название: </a><?php echo $value['title']; ?></div>

<?php }

}

?>

</div>

<div id="registerM" class="modal" >

<!-- Форма регистрации -->

<div id="registerForm" class="modal-content">

<div> <span class="close">&times;</span></div>

<form style="display: flex; flex-direction: column;" method="post" action="/bas/addFlow" enctype="multipart/form-data">

<input class="form-control" type="hidden" name="user\_id" value="<?php echo $user->id(); ?>">

<p>название</p>

<input class="form-control" type="text" name="title">

<p>цена</p>

<input class="form-control" type="text" name="price">

<p>Количество</p>

<input class="form-control" type="text" name="qty">

<p>описание</p>

<textarea class="form-control" type="text" name="description"></textarea>

<input class="form-control" type="file" name="avatar">

<button type="submit" class="btn btn-success">Добавить товар</button>

</form>

</div>

</div>

<script>

// Получаем элементы кнопки и модального окна

var registerButtons = document.getElementById("registerBut");

var registerModal = document.getElementById("registerM");

var closeButton = document.getElementsByClassName("close");

// Когда пользователь нажимает на кнопку, открываем модальное окно

registerButtons.onclick = function() {

registerModal.style.display = "block";

console.log('pid')

};

// Когда пользователь нажимает на кнопку закрытия, закрываем модальное окно

closeButton.onclick = function() {

registerModal.style.display = "none";

}

// Когда пользователь щелкает вне модального окна, закрываем его

window.onclick = function(event) {

if (event.target == registerModal) {

registerModal.style.display = "none";

}

}

</script>

<?php

$view->component('end')

?>