

МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И ИННОВАЦИОННОЙ ПОЛИТИКИ
НОВОСИБИРСКОЙ ОБЛАСТИ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
НОВОСИБИРСКОЙ ОБЛАСТИ
«НОВОСИБИРСКИЙ КОЛЛЕДЖ ЭЛЕКТРОНИКИ И ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКИ»
(ГБПОУ НСО «НКЭ и ВТ»)

Курсовой проект

По дисциплине: «Разработка программных модулей»

На тему: «Интернет магазин электроники Cyber»

По специальности: 09.02.07 Информационные системы и
программирование

Выполнил студент:

9ис-340к Кудашев Никита Максимович

Проверил преподаватель:

Щерба Евгений Андреевич

«___» _____ 20__ г.

_____ (оценка)

Новосибирск, 2022 г.

Содержание

Актуальность исследования.....	3
Цели и задачи исследования.....	3
Объект исследования.....	4
Предмет исследования.....	4
Практическая значимость исследования.....	4
ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	5
1.1. Framework Symfony.....	5
1.2. Использование phpMyAdmin	7
1.3 Язык программирования PHP.....	9
1.4. Шаблонизатор Twig	11
1.5. Компилятор PhpStorm 2022.....	12
ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ	14
2.1 Начало Backend разработки	14
2.2 Подготовка к разработке сайтов на PhpStorm 2022.....	15
2.3 Начальный цикл разработки сайта в PhpStorm 2022.....	16
ЗАКЛЮЧЕНИЕ	27
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	28
ПРИЛОЖЕНИЕ.....	29

ВВЕДЕНИЕ

Актуальность исследования.

Тема является актуальной в связи с высоким ростом спроса на Back-end разработку. В наше время для любой крупной компании требуется сайт его собственно и создают Frontend и Backend разработчики. Для разработки сайтов используется движок (одним из самых популярных является wordpress) или же framework (Сейчас очень востребован php symfony)

Цели и задачи исследования.

Целью работы является разработка сайта магазина электроники.
В связи с поставленной целью в работе решаются следующие задачи:

- изучить Framework Symfony;
- углублённое изучение Php и MariaDB;
- провести окончательную сборку проекта.

Объект исследования

Объектом исследования являются сайты и их разработка

Предмет исследования.

Предметом исследования является Backend составляющая сайта на компиляторе PhpStorm 2022.

В первой главе «Начало Backend разработки» рассмотрена программа в которой будет продвигаться работа и необходимые для неё плагины.

Вторая глава «Подготовка к разработке сайтов на PhpStorm 2022» включает в себя этапы подготовки файлов для начала разработки, перечень языков которые могут понадобиться для разработки.

Третья глава содержит в себе главные моменты в разработке: перенос frontend на framework, контроллеры страниц, база данных, авторизация и регистрация пользователя, настройка роутов.

Практическая значимость исследования.

Заключается в проявлении интереса к углублённому изучению языков программирования таких как php и framework symfony в следствии возможности реализовывать свои идеи путём самостоятельного создания сайтов.

ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Framework Symfony

Symfony — свободный фреймворк написанный на PHP. Symfony предлагает быструю разработку и управление веб-приложениями, позволяет легко решать рутинные задачи веб-программиста. Работает только с PHP 5 и выше. Имеет поддержку множества баз данных (MySQL, PostgreSQL, SQLite или любая PDO - совместимая СУБД). Информации о реляционной базе данных в проекте должна быть связана с объектной моделью. Это можно сделать про помощи ORM инструмента. Symfony представляется с двумя из них: Propel и Doctrine.

Элементы HTML переносятся в шаблонизатор Twig, он создаёт каталог /templates в корне приложения, где будут размещаться все шаблоны. Twig даёт возможность разбития страницы по блокам и возможность наследовать любые части в другой шаблон. Блоки Twig выделяются тегами, записанными с использованием фигурных скобок и знаков процента. Такие теги, как `{% extends index.twig %}` и `{% block main %}`, позволяют наследовать элементы с другого шаблона и сделать разметку по блокам где наследование определённого элементов не нужно. Другие теги, такие как `{{ }}`, помогают сделать корректные пути для ссылок и медиафайлов. Браузеры не отображают Twig-теги, но используют HTML-теги заключённые в блоках для интерпретации содержимого страницы.

Текстовые документы, содержащие разметку на шаблонизаторе Twig (такие документы традиционно имеют расширение `.html.twig`), обрабатываются специальными приложениями, которые отображают документ

в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет-обозревателями», обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер. Наиболее популярными на сегодняшний день браузерами являются Google Chrome, Mozilla Firefox, Opera, Internet Explorer и Safari

```
<!DOCTYPE html><html lang="ru">
  <head>
    { % block header % }
    <meta charset="UTF-8">
    <title>HTML Document</title>
  { % endblock % }
  </head>
  <body>
    { % block main % }
    <p>
      <b>
        Этот текст будет полужирным, <i>а этот — ещё и курсивным</i>.
      </b>
    </p>
  { % endblock % }
  </body>
  <footer>
    { % block footer % }

  { % endblock % }
  </footer>
  <p>
    Информация подвала
  </P>
</html>
```

1.2 PhpMyAdmin

PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд. Приложение распространяется под лицензией GNU General Public License и поэтому многие другие разработчики интегрируют его в свои разработки, например XAMPP, Denwer, AppServ, Open Server.

Использование PhpMyAdmin

PhpMyAdmin используется создателями сайтов для управления базами данных MySQL и MariaDB. Он представляет пользователям простой способ создавать, редактировать и удалять таблицы и поля; добавлять, удалять и изменять данные; выполнять операторы SQL и управлять привилегиями пользователей.

История создания и развития PhpMyAdmin

Проект phpMyAdmin появился на свет 9 сентября 1998 года, когда Тобиас Ратшиллер выпустил версию 0.9.0. За прошедшие полтора десятилетия phpMyAdmin превратился в один из основных инструментов для администрирования MySQL и других MySQL - подобных баз данных.

К созданию phpMyAdmin причастны 669 разработчиков, а основная группа разработчиков возросла с 1 до 9 человек.

За годы жизни на сайте SourceForge.net на трёх специализированных форумах пользователи оставили 36 359 сообщений.

PhpMyAdmin неоднократно удостоивался различных наград, в том числе совсем недавно получил титул “MySQL-приложение 2013 года” на церемонии MySQL Community Awards.

1.3 Язык программирования Php

Php — это интерпретируемый язык программирования общего назначения, работающий на серверной стороне. Это означает, что когда посетитель запрашивает страницу сайта, сервер получает запрос, передаёт управление интерпретатору PHP, он выполняет все необходимые операции, отдаёт готовый код серверу и только потом сервер отправляет готовую страницу браузеру.

PHP изначально создавался специально для web-разработки. Основным преимуществом PHP является возможность добавлять PHP-код непосредственно в обычные HTML документы.

Основные архитектурные черты: динамическая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса, наличие анонимных функций.

Возможности языка

Главная область применения PHP - написание скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять все то, что выполняет любая другая программа CGI, например, обрабатывать данные форм, генерировать динамические страницы или отсылать и принимать cookies. Но PHP способен выполнять намного больше.

- Создание скриптов для выполнения на стороне сервера;
- Создание скриптов для выполнения в командной строке;
- Создание оконных приложений, выполняющихся на стороне клиента;
- Формирование изображений, файлов PDF, Flash роликов;
- Поддержка широкого круга баз данных.

Минусы:

- слабая возможность работы с глобальными исключениями;
- У приложений на RНР более низкая защищенность, чем с использованием других языков.

1.4 Шаблонизатор Twig

Twig — PHP шаблонизатор, который компилирует шаблоны. Он позволяет сделать код шаблонов чище и понятнее и таким образом сократить трудозатраты на обновление. Он расширяем, можно переопределить поведение почти всего движка, включая и ядро. Twig работает в качестве движка шаблонов как на стороне фронтэнда так и на стороне бэкэнда проекта.

История создания

Twig — PHP шаблонизатор с открытым исходным кодом, написанный на языке PHP в 2008 году и берёт своё начало от движков шаблонов Jinja и Django. Был создан Армином Ронахером, идею которого развивает и поддерживает посей день Фабьен Потенсье, ведущий разработчик и идеолог фреймворка Symfony.

Когда он приступил к поиску PHP-шаблонизатора, он сосредоточился на библиотеках, копирующих поведение Django. Спустя несколько часов «гугления» был найден Twig. Его автором является Armin Ronacher, известный по проекту Jinja (шаблонизатор для Python). Испытывая крайнее уважение к Армину за его замечательную работу над Jinja. Twig скорее похож на Jinja, чем на Django, как описано в реализации.

Взглянув на код, он сразу понял что это то, что он ищущ. Главное отличие от Calypso состоит в том, что Twig компилирует шаблоны в обычный PHP-код. Он начал использовать эту библиотеку и в конце этой недели спросил у Армина, не желает ли он дать своему проекту новую жизнь. Его ответ был полон энтузиазма, и я приступил к изучению кода. Его версия сильно отличается от версии Армина, но «лексер» и «парсер» сохранены почти в исходном виде.

Фабьен потратил всего несколько дней на работу над кодом, но я уже горжусь результатом и думаю что пора показывать библиотеку публично. Первое что он сделал - написал документацию и сделал простой сайт. Осталось

ещё много работы: закончить документацию, добавить тесты и RHPDoc; но код уже целостен и функционален:

Встроенное наследование шаблонов (шаблоны компилируются как классы)

Автоматическое экранирование (отсутствие дополнительного времени на запуск — все делается во время компиляции)

Сверх безопасный режим «песочницы» (список допустимых тегов, фильтров и методов которые разрешены в шаблоне)

Расширяемость: вы можете переписывать все что угодно, даже функции ядра, написав расширение; также можно манипулировать AST перед компиляцией. Используя эти возможности, вы можете создать даже свой собственный DSL-язык, ориентированный на ваше приложение.

Несмотря на то, что Twig самый функциональный шаблонизатор, он ещё и самый быстрый.

Особенности

- Высокая скорость работы;
- Наследование шаблонов (возможность определить базовый и дочерние шаблоны);
- Понятный и компактный синтаксис;
- Оптимизированный php код;
- Безопасный режим «песочницы» (пользователи имеют ограниченный набор тегов, фильтров и методов объектов, определённых разработчиком);
- Возможность создания и использования макросов.
- Полностью документирован (и API тоже);
- Простая отладка (при синтаксических ошибках шаблонизатор выведет сообщение об имени файла с ошибкой и номером строки);

1.4 Компилятор PhpStorm 2022

PhpStorm — это среда разработки, которая по-настоящему понимает ваш код. IDE обеспечивает первоклассную поддержку PHP 5.3–8.1, мгновенно предотвращает ошибки, предоставляет точное автодополнение и безопасные рефакторинги, а также возможность редактирования кода на HTML, CSS и JavaScript.

Возможности:

- Поддержка PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2, 7.4 и 8.0, включая генераторы, сопрограммы, пространства имён, замыкания, типаж, синтаксис коротких массивов.
- Автодополнение кода финализирует классы, методы, имена переменных, ключевые слова PHP, а также широко используемые имена полей и переменных в зависимости от их типа.
- Поддержка стандартов оформления кода (PSR1/PSR12, Drupal, Symfony6, Zend).
- Поддержка PHPDoc. PhpStorm предоставляет соответствующее автодополнение кода, основанное на аннотациях `@property`, `@method` и `@var`.
- Детектор дублируемого кода.
- PHP Code Sniffer (phpcs), проверяющий код на лету.
- Рефакторинги (Переименование, Введение переменной/константы/поля, Встраивание переменной).
- Поддержка редактирования шаблонов Smarty (подсвечивание синтаксических ошибок, автодополнение функций и атрибутов Smarty, автоматическая вставка парных скобок, кавычек и закрывающих тегов и др.).

- MVC представление для Symfony2 и Yii фреймворков.
- Распознавание кода, запакованного в PHAR-архивы.

Поддержка фронтенд-технологий

В PhpStorm вы можете работать с самыми современными технологиями: HTML 5, CSS, Sass, Less, Stylus, CoffeeScript, TypeScript, Emmet и JavaScript. При этом будут доступны рефакторинг, отладка и юнит-тестирование. Благодаря функции Live Edit все изменения можно тут же посмотреть в браузере.

PhpStorm может быть оснащён менеджером пакетов, Composer - это пакетный менеджер уровня приложений для языка программирования PHP, который предоставляет средства по управлению зависимостями в PHP-приложении.

ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Начало Backend разработки

Любая разработка начинается с создания папки где будут храниться наши файлы. Для начала разработки нам потребуется установить фреймворк symfony, менеджер пакетов composer и компилятор в котором мы откроем нашу папку с файлами. Моя разработка проводилась в таком компиляторе как PhpStorm 2022.

Для комфортабельной работы в компиляторах нужно произвести установку плагинов которые облегчат нашу работу с созданием нужных файлов, баз данных. Плагины которые использовались мной:

"Doctrine",

"Make-bundle",

"SonataAdminBundle",

"SonataUserBundle",

"EscapeWSSEAuthenticationBundle".

К примеру один из самых известных плагинов для backend разработки это Doctrine он позволяет создавать базу данных и делать запросы на собственном объектно-ориентированном диалекте SQL.

2.2 Подготовка к разработке сайтов на PhpStorm

Для того чтобы начать работу с бэкэндом сайта не обязательно быть профессиональным разработчиком, вам лишь достаточно иметь базовые знания php, twig, запросов к базе данных, ведь чем больше объем ваших знаний тем

более функциональный и оптимизированный сайт у вас получится и тем больше вы сможете реализовать интересных элементов.

Для разработки интересных элементов вам потребуется знания таких языков программирования как JavaScript , JQuery , React , PHP. К примеру при помощи JQuery мы можем создавать вывод списка категорий из базы данных и форму регистрации.

Так же вам обязательно нужно будет разобраться в устройстве шаблонов и так же использовать как можно меньше кода для каждого шаблона чтобы сайт не нагружался при посещении его пользователем.

2.3 Начальный цикл разработки сайта в PhpStorm

Начнём с того что нам следует создать контроллеры основных страниц которые мы будем использовать в качестве шаблонов и сможем настроить роуты.

Любая разработка сайта начинается с разметки блоков главной станицы. Создав разметку по блокам на основной странице мы можем наследовать её свойства на другие страницы и контроллеры.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="style.css">
```



```

</head>
<body>
  <header>
    {% block header %}

    {% endblock %}
  </header>
  <main>
    {% block main %}

    {% endblock %}
  </main>
  <footer>
    {% block footer %}

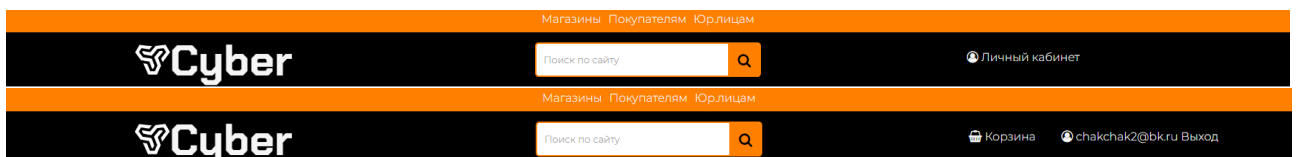
    {% endblock %}
  </footer>
</body>
</html>

```

Сделав разьебиение по блокам, мы можем приступить к натяжке frontend части. После натяжки фронтенда делаем контроллеры страниц, настройку роутов. Создав функцию формы авторизации, корзина будет показываться только авторизованным пользователям.

Header нашего сайта

```
<header>
```



```

<div >
    <div id="verh_stroka">
        <div class="nadpisi_pervayaverh_stroka">
            <a id="magaz" style="cursor: pointer;" href="">Магазины</a>
            <a id="pokupateli" href="">Покупателям</a>
            <a id="urli" href="">Юр.лицам</a>
        </div>
    </div>
</div>
<div>
    <div id="vtoraya_verh_stroka">
        <div class="logo">
            
            <a id="LoGo" href="">Cyber</a>
        </div>
        <div id="d11">
            <div class="d1">
                <form>
                    <input type="text" placeholder="Поиск по сайту">
                    <button type="submit"></button>
                </form>
            </div>
        </div>
        <div id="Vtoraya_verh_stroka_nadpisi">
            <i class="fa fa-shopping-basket" style="" aria-
hidden="true"></i><a id="Korzina" href="php/Korzina.php">&nbsp;Корзина</a>
            <i class="fa fa-user-circle-o" style="" aria-
hidden="true"></i><a href="" id="lichKab">&nbsp;Личный кабинет</a>

```

```

<div class="pop_up" id="pop_up">
<div class="pop_up_container">
  <div class="pop_up_body" id="pop_up_body">
    <form method="post" action="{{ path('app_login') }}">
      {% if error %}
        <div class="alert alert-danger">{{ error.messageKey|trans(error.messageData,
'security') }}</div>
      {% endif %}
      <div class="input_okna">
        <input placeholder="Логин или e-mail" type="email" value="{{ last_username }}"
name="email" id="inputEmail" class="form-control" autocomplete="email" required autofocus>

        <input placeholder="Пароль" type="password" name="password" id="inputPassword"
class="form-control" autocomplete="current-password" required>
      </div>
      <input type="hidden" name="_csrf_token"
        value="{{ csrf_token('authenticate') }}"
      >

      <button class="btn btn-lg btn-primary" type="submit">
        Войти
      </button>
      <div class="registracia"><p class="net_akaunta">Нет аккаунта? <a id="registr"
href="">зарегистрироваться</a></p></div>

    </form>

    <p class="voiti">Войти в аккаунт</p>

    <div class="pop_up_close" id="pop_up_close">&#10006</div>
  </div>
  <div class="Pop_up_body" id="pop_up_registr_body">
    <p>Регистрация</p>

```

```

    {% if registrationForm is defined %}

    {{ form_start(registrationForm) }}
    <div class="reg_input">
    {{ form_row(registrationForm.email, {'attr':{'placeholder':'Email'}}) }}

    {{ form_row(registrationForm.plainPassword, {'attr':{'placeholder':'Пароль'}}) }}
    </div>
        <div class="agt1">
            <div class="reg_p">
                <p>Подтвердить</p>
            </div>

    {{ form_row(registrationForm.agreeTerms, {'attr':{'class':'agt'}}) }}
        </div>
        <div class="button_reg">
            <button type="submit" class="btn">Регистрация</button>
        </div>
    {{ form_end(registrationForm) }}
    {% endif %}

    <div class="Pop_up_close" id="Pop_up_close">&#10006</div>
</div>
</div>
</div>

<script src="{{ {'js/ForLichkab.js'} }}"></script>

</div>
</header>

```

Код JavaScript для POPUP окна

```

    const openPopUp = document.getElementById('LichKab');
    const closePopUp = document.getElementById('pop_up_close');
    const popUpMain = document.getElementById('pop_up');
    const popUp = document.getElementById('pop_up_body');
    const otcritie = document.getElementById('registr');
    const zacritie = document.getElementById('Pop_up_close');
    const popupRegistr = document.getElementById('pop_up_registr_body');

    openPopUp.addEventListener('click', function (e){
        e.preventDefault();
        popUpMain.classList.add('active');
        popUp.classList.add('active');
        $('.Pop_up_body').hide();
        $('.pop_up_body').show();
    })

    otcritie.addEventListener('click', function (a){
        a.preventDefault();
        popupRegistr.classList.add('active');
        $('.Pop_up_body').show();
        $('.pop_up_body').hide();
        popUp.classList.remove('active');

    })

    closePopUp.addEventListener('click' ,() => {
        popUp.classList.remove('active');
        $('.Pop_up_body').hide();
        $('.pop_up_body').hide();
        popUpMain.classList.remove('active');
    })

    zacritie.addEventListener('click' ,() => {
        popupRegistr.classList.remove('active');

```

```
$('.Pop_up_body').hide();  
$('.pop_up_body').hide();  
popUpMain.classList.remove('active');  
})
```

Далее проверяем работоспособность нашего PopUp окна.

Окно авторизации (PopUp окно)

×

Войти в акаунт

Логин или e-mail

Окно регистрации

×

Регистрация

Логин или e-mail

Пароль

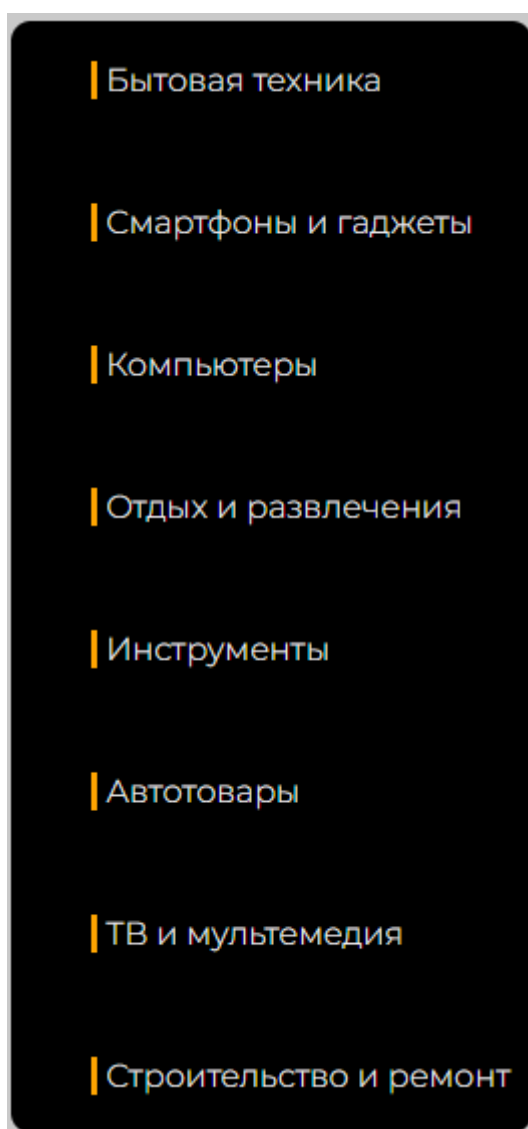
Повторите пароль

Зарегистрироваться

Создание главной части сайта.

Главная часть сайта отображается в коде внутри тега `main`(и блока внутри тега `main`),внутри него мы и пишем код нужных нам элементов.

Начнём пожалуй чуть ли не с главного элемента нашей основной части нашего сайта это меню категорий. Меню категорий сделано путём вывода неупорядоченного списка `ul-li` из базы данных с помощью цикла.



```

        <ul class="clearfix">
        {% for catalog in cataloglist %}

        <li class="betxim">
            <a href="{{ '/catalog/' }}{{ catalog.id }}" class="product">&nbsp;{{ catalog.name }}</a>

        </li>

        {% endfor %}

    </ul>

```

Естественно внутри всей структуры у нас стоит тег `<a>` к которому мы приписываем такое свойство как `href= “ “`, указываем настроенный роут и массив который отправляется на страницу с контроллера где происходит запрос к базе данных.

Создание Контроллера.

Контроллер отвечает за главные элементы страницы, такие как переходы и запросы к базе данных.


```

class PageController extends AbstractController
{
    #[Route('/', name: 'home_page')]
    public function index(
        ManagerRegistry $doctrine,
        AuthenticationUtils $authenticationUtils,
        Request $request,
        UserPasswordHasherInterface $userPasswordHasher,
        UserAuthenticatorInterface $userAuthenticator,
        LoginAuthenticator $authenticator,
        EntityManagerInterface $entityManager
    ): Response
    {
        return $this->render( view: 'base.html.twig', [
            'cataloglist'=>$cataloglist,

            'last_username' => $lastUsername,
            'error' => $error,
            'registrationForm' => $form->createView()
        ]);
    }
}

```

Здесь у нас происходит обработка главной страницы, пути к ней и отправка данных/функций.

```

if ($form->isSubmitted() && $form->isValid()) {
    // encode the plain password
    $user->setPassword(
        $userPasswordHasher->hashPassword(
            $user,
            $form->get('plainPassword')->getData()
        )
    );

    $entityManager->persist($user);
    $entityManager->flush();
    // do anything else you need here, like send an email

    return $userAuthenticator->authenticateUser(
        $user,
        $authenticator,
        $request
    );
}

// get the login error if there is one
$error = $authenticationUtils->getLastAuthenticationError();

// last username entered by the user
$lastUsername = $authenticationUtils->getLastUsername();

$cataloglist = $doctrine->getRepository( persistentObject: Catalog::class)->findBy(['parent_id'=>0]);

```

Формирование запроса в базу данных и проверка для формы регистрации авторизован пользователь или нет - в зависимости от этого на страницу будут отправлены данные которые позволят скрывать корзину или показывать её и отображение почты авторизованного пользователя.

ЗАКЛЮЧЕНИЕ

При выполнении данной работы разработан сайт, а также описан сам процесс разработки. Данный сайт является незаконченным продуктом, но при регулярной поддержке и доработке можно сделать его конкурентоспособным на рынке.

Также можно использовать любой из элементов данного сайта в любом другом проекте при полной доработке, или же выложить их на форум коих огромное множество, чтобы поделиться с сообществом или же для получения прибыли. Были рассмотрены основные аспекты разработки сайтов в целом и на symfony и php в частности. Компилятор на котором создавался проект показал себя как очень гибкий, удобный и многофункциональный инструмент для сайтов, не требующий установки дополнительного программного обеспечения, при этом значительных недостатков при работе с платформой замечено не было.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1) SensioLabs - “Fabien potencier. Symfony 6 быстрый старт”, документация, 2022.
- 2) Дмитрий Быкадоров "Один год с Symfony", Наука и Техника, 2016.
- 3) Робин Никсон - "Создаём динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5", Питер, 2019.
- 4) Веллинг, Томсон - "Разработка веб-приложений с помощью PHP и MySQL", Питер, 2017.
- 5) Дэвид Скляр, Адам Тразтенберг - “ PHP. Рецепты программирования ”, 2015.

ПРИЛОЖЕНИЕ

