СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc72949750)

[1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУВЕБ САЙТА 4](#_Toc72949751)

[2 ОБЩАЯ ЧАСТЬ 6](#_Toc72949752)

[2.1Выбор и обоснование средств реализации базы данных 6](#_Toc72949753)

[2.2 Выбор и обоснование средств разработки программного продукта 7](#_Toc72949754)

[2.3 Анализ прототипов выбранной ИС 11](#_Toc72949755)

[3.СПЕЦИАЛЬНАЯ ЧАСТЬ 13](#_Toc72949756)

[3.1 Разработка алгоритма и проектирование структуры базы данных 13](#_Toc72949757)

[3.2 Разработка программного продукта 15](#_Toc72949758)

[3.3 Руководство пользователя 34](#_Toc72949759)

[ЗАКЛЮЧЕНИЕ 40](#_Toc42525560)

[СПИСОК ЛИТЕРАТУРЫ 42](#_Toc72949695)

[ПРИЛОЖЕНИЕ 43](#_Toc72949696)

ВВЕДЕНИЕ

Целью данного курсового проекта является, разработка веб приложения, реализующего функционал ежедневника с использованием, программного языка PHP JS CSS HTML и СУБД MY SQL OPEN SERVER

Разрабатываемый веб сайт поможет эффективно использовать рабочее, и свободное время, а также хранение, выведение на экран, запланированных событий пользователю. В современном мире, огромное количество планов, целей, зада, которые необходимо решить в этот же день, либо через годы. Запомнить всё невозможно, а данный продукт с использованием информационных веб технологий будет особенно актуален. Выдвигается гипотеза, что программный продукт, реализующий функционал ежедневник с использованием PHP JS CSS и СУБД MY SQL OPEN SERVER будет полезным для использования в повседневной жизни. Используя этот программный продукт, поможет сконцентрировать ваше внимание на достижения поставленных задач и целей. В результате, личностный рост, и успехи в трудовой деятельность.

Объектом в данной работе является программный веб продукт, реализующий функционал ежедневника.

Предметом является возможность оптимизации процессов, связанных с веб обработкой информации, регистрации незарегистрированных, и авторизации в системе уже зарегистрированных пользователей.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Разработать техническое задание;
* Проанализировать прототипы;
* Разработать веб-приложение;
* Провести тестирование и отладить веб-приложение;
* Разработать руководство пользователя

Разрабатываемое веб-приложение сможет применяться в повседневной жизни, для достижений поставленных целей и реализации запланированных задач в своё время, используя его с максимальной эффективностью.

# 1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУВЕБ САЙТА

**Разработка веб-приложение «Ежедневник»**

Прототипом данной системы можно считать интернет ресурс: https://worktek.ru/individual WorkTek – pr. 1qazQ2wsx#**,** имеющий основную функциональную часть разрабатываемого приложения.

**Требования к функциональным характеристикам следующие:**

1. Модальное окно регистрации должна содержать поля для ввода фамилии, имени, отчества (при наличии, данное поле не обязательно), пароль и подтверждение пароля, телефон или email, дата рождения, пол. При регистрации происходит проверка на соответствие паролей, а также реализовать проверку телефона или email. Механизм защиты от авто регистрации реализовать с помощью captha. Также в базу данных заносится информация о дате регистрации в системе.
2. В модальном окне авторизации пользователь вводит логин пароль.
3. Главная страница сайта должна иметь разделение на «шапку», «контент» и «подвал». В шапке необходимо разместить логотип, название организации и строку поиска, а также необходимо расположить горизонтальное меню навигации.
4. В основной части сайта необходимо расположить карусель с информацией об актуальности планирования целей для успешной реализации поставленных задач, ниже окно прогноз погоды далее видео для ознакомления с работой программы «Ежедневник».
5. В подвале сайта необходимо расположить реквизиты, ссылки на социальные сети и служебную информацию.
6. Не авторизованному пользователю необходимо предусмотреть возможность ознакомления с информационными блоками веб-приложения, и видео руководство, реализовать возможность авторизоваться.
7. После авторизации пользователю должен быть доступен личный кабинет, в котором необходимо предусмотреть возможность редактирования личных данных и смену пароля.
8. Авторизованный пользователь должен иметь возможность заполнять календарь задачами на день, на квартал, месяц, годы. В дальнейшем производить заметки, корректировки, добавление и удаление своих целей, задач и планов, возможных комментариев.
9. Должна быть реализована возможность выводить данные событий, за определённый промежуток времени, в таблицу.
10. Авторизованный пользователь должен иметь возможность оставить отзыв, который сохраняется в базе данных и отображается на листе веб-приложения, а также удалять его.

**Требования к информационному обеспечению следующие:**

К веб-приложения должна быть предоставлена документация, содержащая техническое описание продукта и руководство пользователя.

**Требования к лингвистическому обеспечению следующие:**

Веб-приложение должен иметь интерфейс и документацию на русском языке.

**Требования к программному обеспечению следующие:**

Реализация серверной части приложения должна быть организована на языке PHP, визуальная часть реализуется с применением CSS, HTML, PHPи JS. В качестве СУБД использовать MySQL.

**Требования к техническому обеспечению следующие:**

Разрабатываемое веб-приложение должно корректно отображаться на ПК, планшетах и мобильных устройствах. Браузером для тестирования предусмотреть Chrome и Opera.

# 2 ОБЩАЯ ЧАСТЬ

## 2.1 Выбор и обоснование средств реализации базы данных

Неотъемлемой частью разработки приложения и его адекватного функционирования, хранение, а также возврата необходимых данных, работала СУБД My SQL

База данных My SQL это упорядоченная информация, связанная между собой определёнными отношениями. Представлена она в виде таблиц, в которых и лежит вся эта информация.

**СУБД**Microsoft SQL – это система управления базами данных от компании Microsoft. Она очень популярна в корпоративном секторе, особенно в крупных компаниях. И это не просто СУБД – это целый комплекс приложений, позволяющий хранить и модифицировать данные, анализировать их, осуществлять безопасность этих данных и многое другое.

**MySQL является решением для малых и средних приложений. Входит в состав серверов Open Server, WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы**

MS SQL Server не предназначен непосредственно для разработки пользовательских приложений, а выполняет функции управления базой данных. Сервер имеет средства удаленного администрирования и управления операциями, организованные на базе объектноориентированной распределенной среды управления.

**MySQL отличатся хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP.**

**Немаловажным фактором является ее бесплатность. MySQL распространяется на условиях общей лицензии GNU (GPL, GNU Public License).**

## 2.2 Выбор и обоснование средств разработки программного продукта

При проектировании приложения были использованы такие программные средства как Visual Studio Code, Open Server, и Opera Browser.

При написании приложения был выбран язык программирования PHP.

Приложение на РНР, использующее для хранения информации базу данных (в частности MySQL) всегда работает быстрее приложения, построенного на файлах, также значительно увеличивает скорость и качество, передачи данных в базу MySQL и загрузка страницы сайта. С помощью метода передачи данных AJAX jQuery-3.6.0.min информация возвращается на страницу сайта без её перезагрузки и мерцания контента на страницы сайта.

При выполнении данной работы была выбрана интегрированная среда разработки Visual Studio Code.

Visual Studio Code — это редактор исходного кода. Он поддерживает ряд языков программирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности. Многие возможности Visual Studio Code недоступны через графический интерфейс, зачастую они используются через палитру команд или JSON файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш.

Visual Studio также позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа.

Open Server Panel — это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий.

Программный комплекс имеет богатый набор серверного программного обеспечения, удобный, многофункциональный продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а также для предоставления веб-сервисов в локальных сетях. В браузере Opera предусмотрены средства управления для людей с нарушениями зрения или ограниченными двигательными возможностями: есть голосовое управление и возможность подключения «экранных дикторов». С помощью различных установок пользователь может настроить интерфейс по своему вкусу. Масштабирование страниц позволяет пропорционально увеличивать или уменьшать все элементы документа: текст, изображения, флеш-анимацию, создавая эффект приближения и удаления страницы. Масштаб можно изменять в пределах от 20 % до 1000 %. Пользователь может настроить шрифты и цвета для веб-страниц, и даже преодолеть настройки стилей CSS. Это может быть полезным, например, для страниц с неудачным дизайном: когда шрифты трудночитаемы или контраст между цветом шрифта и фоном недостаточен. В настоящее время Opera поддерживает многие веб-стандарты, включая редакции CSS 3, HTML 5, XHTML 1.1, HTML Basic, XHTML Mobile Profile, XHTML+Voice, WAI-ARIA, WML 2.0, XSLT, XPath, XSL-FO, ECMAScript 3(JavaScript), DOM 2, XMLHttpRequest, HTTP 1.1, Юникод, SVG 1.1 Basic, SVG 1.1 Tiny, DIF89a, JPEG, WebExtensions. Браузер полностью совместим с графическими форматами APNG, PNG.

Так же для изменения стилевых параметров некоторых элементов сайта, был использован формальный язык разметки CSS.

Использование CSS при создании любого Интернет-проекта открывает доступ к следующим преимуществам:

* Возможности создания более чистого кода, который быстрее загружается, легче поддерживается и лучше оптимизирован для роботов поисковых систем.
* Доступ к более модульному коду, который создает правила стиля для множества страниц, облегчает управление Интернет-ресурсом и имеет единообразный дизайн.
* Точности контроля за отображением и позиционированием всех элементов страниц вашего сайта.

К сожалению, у CSS есть и недостаток, заключающийся в некорректности браузерами тех или иных интерпретаций инструкций каскадных стилевых таблиц. Это касается верстки Интернет-страницы и взаимного расположения элементов. Для того, чтобы сократить время на продумывание и разработку единого стиля на CSS было решено использовать готовое решение, Bootstrap 5.

Bootstrap - интуитивно простой и в тоже время мощный интерфейсный Фреймворк, повышающий скорость и облегчающий разработку Web-приложений. Преимущества, которые даёт Фреймворк Bootstrap при разработке на его основе frontend части сайтов и интерфейсов админок:

* высокая скорость создания качественной адаптивной вёрстки даже начинающими веб-разработчиками (достигается это благодаря использованию готовых компонентов, созданных профессионалами);
* кроссбраузерность и кроссплатформенность (корректное отображение и работа сайта во всех поддерживаемых этим Фреймворком браузерах и операционных системах);
* наличие большого количество готовых хорошо продуманных компонентов, протестированных огромным сообществом веб-разработчиков на различных устройствах;
* возможность настройки под свой проект, достигается это посредством изменения SCSS переменных и использования Bootstrap миксинов (можно изменить количество колонок, цвета, радиуса скруглений углов элементов, отступы между колонками и многое другое);
* низкий порог вхождения; для работы с Фреймворком не обязательно иметь глубокие знания по HTML, CSS, JavaScript и jQuery (достаточно знать только основы вышеперечисленных технологий);
* наличие хорошо продуманного дизайна компонентов и согласованности (в Bootstrap все компоненты выполнены в едином стиле);
* наличие огромного сообщества, большого количества статей, рецептов и видеоматериалов; всё это при желании поможет не только хорошо разобраться в Фреймворке, но и найти ответы практически на любые вопросы.

Для разработки данного сайта был выбран стандартизированный язык разметки документов во Всемирной паутине HTML. Большинство веб-страниц реализованы с помощью смежных технологий AJAX, плюсы использования этой технологии:

* Позволяет сократить трафик при работе с веб-приложением благодаря тому, что вместо загрузки всей страницы достаточно загрузить только изменившуюся часть или вообще только получить/передать набор данных в формате JSON.
* С помощью AJAX веб-приложение может запрашивать только тот контент, который необходимо обновить, что значительно сокращает использование полосы пропускания и время загрузки.
* Использование асинхронных запросов позволяет веб-браузеру клиента UI быть более интерактивным и быстро реагировать на входные данные, а разделы страниц также могут быть перезагружены индивидуально.
* Пользователи могут воспринимать приложение как более быстрое или отзывчивое, даже если оно не изменилось на стороне сервера.
* Использование AJAX может уменьшить количество подключений к серверу.
* Состояние может поддерживаться на протяжении всего веб-сайта. Переменные JAVASCRIPT будут сохраняться, поскольку главная страница контейнера не нуждается в перезагрузке.

Недостатки:  
Самым важным недостатком JSON является то, что формат очень трудно читается для людей, и что, конечно, каждая запятая, цитата и скобка должны быть точно в правильном месте. В то время как это также верно для XML, беспорядок сложного синтаксиса JSON, такой как}}]} в конце фрагмента данных, может напугать новичков и привести к сложной отладке.

## 2.3 Анализ прототипов выбранной ИС

При проектировании приложения реализующего функционал ежедневника прототипом стали Web приложение «Онлайн органайзер планировщик ежедневник»

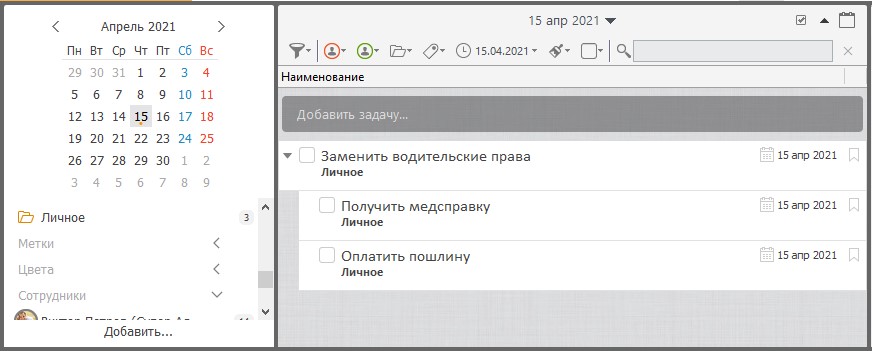
Ознакомится и скачать это приложение можно по ссылке: https://www.leadertask.ru/content/view/481 Очень удобная возможность, состояние списка дел на любой день в будущем. Есть задачи, которые нужно выполнить через продолжительный период времени. Например, обновить водительское удостоверение. 

Рисунок 1 – «интерфейс приложения».

Прикрепление файлов: фотографий, документов, медиа файлов. Такая возможность пригодиться для того, чтобы в одном месте видеть всю информацию о задаче.

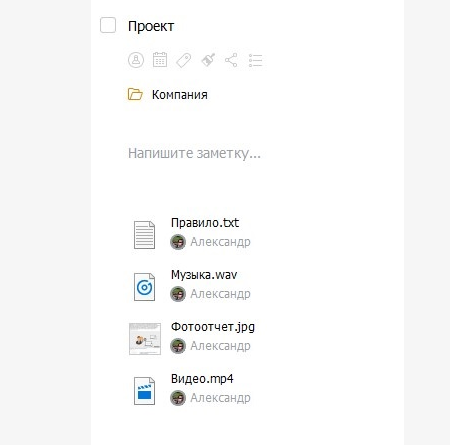
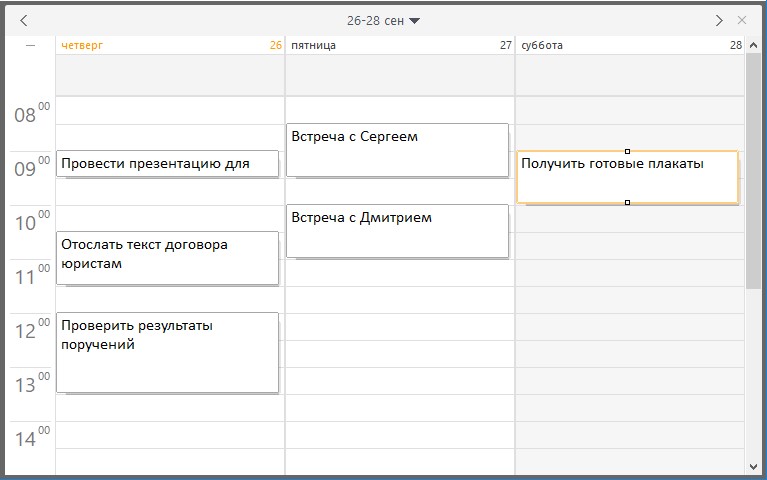


Рисунок 2 – «прикреплённые файлы к определённым событиям».

Составление почасового расписания, если в программе реализована временная школа, как в бумажном ежедневнике, то вы сможете создать наглядное почасовое расписание на день. Представлено на рисунке 3.

Рисунок 3 – «почасовое расписание на день».

Некоторые дела должны быть привязаны к конкретному времени. Яркий пример – встречи. Программа не позволит вам опоздать, если за несколько минут до начала покажет напоминание.

# 3.СПЕЦИАЛЬНАЯ ЧАСТЬ

## 3.1 Разработка алгоритма и проектирование структуры базы данных

База данных веб-приложение «Ежедневник» была создана на основании задания, представленного в приложении А. Функционал ежедневника сформирован на платформе MYSQL OPEN-Server. Графическая и физическая модель таблиц, базы данных, применялся режим конструктора MYSQL, где и были сформированы свойства полей всех таблиц базы данных программного веб-приложения. Данные представленного веб продукта «Ежедневник» расположены в пяти таблицах, четыре таблицы связаны между собой, обладающие особой структурой:

* «Users» (Пользователи)
* «information» (Отзывы)
* «fosad» (Центральная страница)- без связи
* «everybay» (Задачи ежедневные)
* «category» (Категории)

Таблица«Users» (Данные о пользователе),  ключевое поле «id\_user» (Идентификатор пользователя) – int, следующие поля таблицы: «iast\_name» (Имя пользователя) – varchar(50), «first\_name» (Фамилия ) - varchar(50), «patronymic» (Отчество) – varchar(30) , «email» (Электронный адрес) – varchar(30), «phone» (телефон) – varchar(30) , «Password» (Пароль) - varchar(50), «date\_of\_birth» (Дата рождения) –date, «registration\_date» (Дата регистрации) – datetime, «sex» (Пол) – bit(1).

Следящая таблица «information» (Отзывы) имеет ключевое поле id –int(11), а так-же связь с таблицей «users» по полю id\_users – int(11), следующее поля «name\_inf» (Имя отзыва) – varchar(200), «infomat» (Текст отзфва) – text, «foto» (Фото) – varchar(300), «email» (Электронный адрес пользователя) – varchar(50).

Таблица «fosad» (Центральная страница), имеет ключевое поле «id\_fosad» - int(11), связей нет. Следующие поля: «text\_heading» (Заголовок текста слайда карусели) – varchar(50), «foto» (Фото на карусель) – varchar(300).

Таблица «everybay» (Задачи на день), ключевое поле «id\_ever» - int(11), «id\_user» - int(11) это поле имеет связь с таблицей «users» по одноимённому полю, «id\_categ» - int(11), это поле имеет связь с таблицей «category» (Категории), «task» (Задачи) - text, «my\_data» (Дата и время начало задачи) - datetime, «itogi» (Исполнена задача либо нет) - bit(1).

Таблица «category» (Категории), имеет ключевое поле «id\_categ» (Категория) – int(11), обладает одноимённой связью с таблицей «everyba», с полем «nave\_categ» (Наименование категории) – varchar(20).

Целостность данных *–* это значит, что данные будут надёжно хранится в базе данных. СУБД автоматически поддерживает целостность данных с помощью установленных связей между таблицами. Таблицы базы данных связаны связями «один ко многим»; для каждой из связей установлена поддержка целостности данных, каскадные обновление и удаление. Схема связей таблиц представлена в соответствии с рисунком 4.

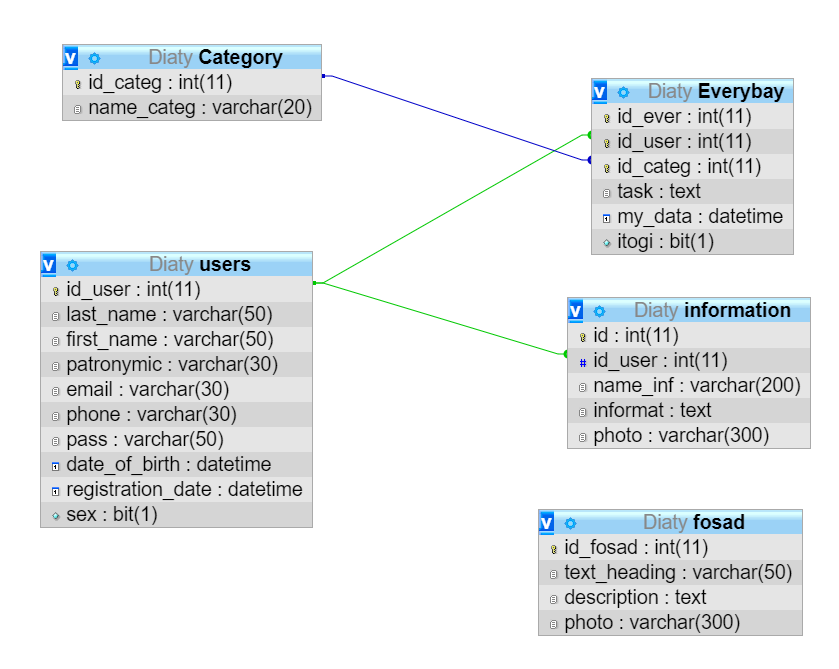


Рисунок 4 – Схема связей таблиц

**3.2 Разработка программного продукта**

При разработке данного программного продукта было реализовано несколько подсистем. Первая из них – это система авторизации.

В ней реализована проверка на существование пользователя, проверка на правильность введенного пароля, так же реализована возможность перейти на форму регистрации. Код файла программы представлен на листинге 1.

Листинг 1 - Код файла

<?php

session\_start();

include("connect.php");

$login=$\_POST['login'];

$pass=$\_POST['pass'];

$s="select id\_user from users where (phone='$login' and pass='$pass')

or (email='$login' and pass='$pass')";

$res=mysqli\_query($link, $s);

$n=mysqli\_num\_rows($res);

$mass=array("status"=> "", "mess"  => "", "login"=>"");

if ($n==0){

   $mass['status']=0;

   $mass['mess']="Неверный логин или пароль";

}

else{

    $\_SESSION['login']=$login;

    $row=mysqli\_fetch\_array($res);

    $\_SESSION['id\_user']=$row[0];

    $mass['status']=1;

    $mass['login']=$login;

    $mass['mess']="Добро пожаловать,".$login;

}

echo json\_encode($mass);

?>

Код JS авторизация

let sub\_autorization=document.getElementById("sub\_autorization");

sub\_autorization.onclick=function(){

    let login=document.getElementById("login").value;

    let pass=document.getElementById("pass").value;

    $.ajax({

        url: 'autorization\_bd.php',

        method: 'post',

        dataType: 'html',

        data: {login: login, pass:pass} ,

        success: function(data){

            let mass\_data=JSON.parse(data);

            let message\_autorization=document.getElementById("message\_autorization");

            message\_autorization.style.display="block";

            message\_autorization.innerHTML=mass\_data['mess'];

            if (mass\_data['status']==1){

                let authorized=document.getElementById("authorized");

                let unauthorized=document.getElementById("unauthorized");

                let btn=document.getElementById("btn");

                btn.style.display="block";

                authorized.style.display="block";

                unauthorized.style.display="none";

                let login\_user=document.getElementById("login\_user");

                login\_user.innerHTML=mass\_data['login'];

                $(".modal").modal("hide");

            }

        }

    });

};

Вторая из подсистем – это регистрация. Возможность регистрации пользователя реализована из меню администратора и при запуске пользователем программы. При заполнении полей определённой длины и определёнными символами происходит добавление нового пользователя в базу данных. Также в нём происходит проверка на пустоту полей и существование логина (login) в базе данных. Код файла программы представлен на листинге 2.

Листинг 2 - Код файла

<?php

include("connect.php");

$last\_name=$\_POST['last\_name'];

$first\_name=$\_POST['first\_name'];

$patronymic=$\_POST['patronymic'];

$phone=$\_POST['phone'];

$email=$\_POST['email'];

$pass1=$\_POST['pass1'];

$date\_of\_brith=$\_POST['date\_of\_brith'];

$sex=$\_POST['sex'];

$s="";

if ($email!="" && $phone!="")

    $s="SELECT \* FROM `users` WHERE `email`='$email' OR`phone`='$phone'";

else if ($email==""){

    $s="SELECT \* FROM `users` WHERE `phone`='$phone'";

}

else{

    $s="SELECT \* FROM `users` WHERE `email`='$email'";

}

$r=mysqli\_query($link,$s);

$n=mysqli\_num\_rows($r);

if ($n!=0){

    echo "Данный email или телефон уже зарегистрирован";

   echo $s;

}

else{

    $s1="INSERT INTO `users`( `last\_name`, `first\_name`, `patronymic`,

    `email`, `phone`, `pass`, `date\_of\_birth`, `registration\_date`, `sex`)

     VALUES ('$last\_name','$first\_name','$patronymic',

    '$email','$phone','$pass1','$date\_of\_brith', CURRENT\_DATE(),$sex)";

    $r1=mysqli\_query($link,$s1);

        echo "Пользователь успешно зарегистрирован";

}

?>

Код JS регистрация

function registration(){

    var pass1=document.getElementById("pass1").value;

    var pass2=document.getElementById("pass2").value;

    var email=document.getElementById("email").value;

    var phone=document.getElementById("phone").value;

    var last\_name=document.getElementById("last\_name").value;

    var first\_name=document.getElementById("first\_name").value;

    var patronymic=document.getElementById("patronymic").value;

    var date\_of\_brith=document.getElementById("date\_of\_birth").value;

    var sex=document.getElementById("sex").value;

    var message\_registration2=document.getElementById("message\_registration2");

    var mess="";

    if (pass1!=pass2){

        mess+="<p>Пароли не совпадают!</p>";

    }

    email=email.trim();

    phone=phone.trim();

    if (email.length==0 && phone.length==0){

        mess+="<p>Необходимо заполнить Email или телефон!</p>";

    }

    if (mess.length!=0){

        message\_registration2.style.display="block";

        message\_registration2.innerHTML=mess;

    }

    else{

    $.ajax({

        url: 'registration\_bd.php',

        method: 'post',

        dataType: 'html',

        data: {last\_name: last\_name, first\_name:first\_name,patronymic:patronymic,

            date\_of\_brith:date\_of\_brith,sex:sex,email:email,phone:phone, pass1:pass1} ,

        success: function(data){

            message\_registration2.style.display="block";

            message\_registration2.innerHTML=data;

            if (data=="Пользователь успешно зарегистрирован"){

                var form\_registration=document.getElementById("form\_registration");

                form\_registration.style.display="none";

            }

            return false;

        }

    });

}

   return false;

}

В личном кабинете пользователя реализована возможность редактировать свои данные. Код файлов продемонстрирован на листинге 3.

Листинг 3 – Код файла

<?php

include("header.php");

?>

<link href="Css/style(t).css" rel="stylesheet">

<form>

  <div class="container">

    <div class="modal-body">

      <div>

      <p class="my\_message" id="message\_updata\_task"></p>

        <div id="form\_load\_tascke">

          <label class="form-check-label">задача</label>

          <input type="text" id="task" class="form-control">

          <label class="form-check-label">дата и время</label>

          <div class="form-group row">

              <div class="col-xs-10">

                <input class="form-control" type="datetime-local" value="2021-00-00T00:00:00" id="my\_data">

              </div>

            </div>

            <div>

            <label class="form-check-label" disabled>Выполненно/нет</label>

            <td><input type="checkbox" id="inlineCheckbox1" value="option1" ><br>

            </div>

          <label class="form-check-label" disabled>Категория</label>

          <div id="select\_status\_id">

              <select class="form-select" id="name\_categ">

                <option value="1">mbmnn</option>

                <option value="2">mbmnn</option>

                <option value="3">mbmnn</option>

                <option value="4">mbmnn</option>

                <option value="5">mbmnn</option>

              </select>

        </div>

      </div>

      <br>

      <div>

       <input type="button" class="btn btn-primary" value="Редактировать событие" id="submit\_update\_task" >

       <br>

      </div>

    </div>

  </form>

    <script src="jsj/diary\_tb.js"></script>

    <script src="jsj/update\_task.js"></script>

<?php

include("footer.php");

?>

Код JS редактирование данных пользователя

  load\_user\_date();

let message\_new\_pass=document.getElementById("message\_new\_pass");

  message\_new\_pass.style.display="none";

function load\_user\_date(){

    $.ajax({

    url: 'load\_user\_date.php',

    method: 'post',

    dataType: 'html',

    data: "" ,

    success: function(data){

        let user\_data=JSON.parse(data);

        let last\_name\_update=document.getElementById("last\_name\_update");

        last\_name\_update.value=user\_data["last\_name"];

        let first\_name\_update=document.getElementById("first\_name\_update");

        first\_name\_update.value=user\_data["first\_name"];

        let patronymic\_update=document.getElementById("patronymic\_update");

        patronymic\_update.value=user\_data["patronymic"];

        let phone\_update=document.getElementById("phone\_update");

        phone\_update.value=user\_data["phone"];

        let email\_update=document.getElementById("email\_update");

        email\_update.value=user\_data["email"];

        let date\_of\_brith\_update=document.getElementById("date\_of\_brith\_update");

        var d=user\_data["date\_of\_birth"];

        var dateVal = new Date(user\_data["date\_of\_birth"]);

        var day = dateVal.getDate().toString().padStart(2, "0");

        var month = (1 + dateVal.getMonth()).toString().padStart(2, "0");

        var inputDate = dateVal.getFullYear() + "-" + (month) + "-" + (day) ;

        date\_of\_brith\_update.value = inputDate;

let sex\_update=document.getElementById("sex\_update");

        sex\_update.value=user\_data["sex"];

    }

        });

}

let update\_status=document.getElementById("update\_status");

update\_status.onclick=function(){

last\_name\_update.disabled=!(last\_name\_update.disabled);

first\_name\_update.disabled=!(first\_name\_update.disabled);

patronymic\_update.disabled=!(patronymic\_update.disabled);

phone\_update.disabled=!(phone\_update.disabled);

email\_update.disabled=!(email\_update.disabled);

date\_of\_brith\_update.disabled=!(date\_of\_brith\_update.disabled);

sex\_update.disabled=!(sex\_update.disabled);

let submit\_update\_user\_date=document.getElementById('submit\_update\_user\_date');

submit\_update\_user\_date.disabled=!(submit\_update\_user\_date.disabled);

if(sex\_update.disabled==true){

update\_status.value="Редактировать личные данные";

load\_user\_date();

}

else{

update\_status.value="Отменить редактирование";

}

}

function update\_user\_date(){

let message\_update\_user\_date=document.getElementById("message\_update\_user\_date");

let mess="";

let em=email\_update.value.trim();

let ph=phone\_update.value.trim();

if (em.length==0 && ph.length==0){

mess+="<p>Необходимо заполнить Email или телефон!</p>";

}

if (mess.length!=0){

message\_update\_user\_date.style.display="block";

message\_update\_user\_date.innerHTML= mess;

}

else{

last\_name\_update.disabled=!(last\_name\_update.disabled);

first\_name\_update.disabled=!(first\_name\_update.disabled);

patronymic\_update.disabled=!(patronymic\_update.disabled);

phone\_update.disabled=!(phone\_update.disabled);

email\_update.disabled=!(email\_update.disabled);

date\_of\_brith\_update.disabled=!(date\_of\_brith\_update.disabled);

sex\_update.disabled=!(sex\_update.disabled);

let submit\_update\_user\_date=document.getElementById('submit\_update\_user\_date');

submit\_update\_user\_date.disabled=!(submit\_update\_user\_date.disabled);

$.ajax({

url: 'update\_user\_data.php',

method: 'post',

dataType: 'html',

data: {last\_name: last\_name\_update.value,

first\_name:first\_name\_update.value,

patronymic:patronymic\_update.value,

date\_of\_brith:date\_of\_brith\_update.value,

sex:sex\_update.value,

email:email\_update.value,

phone:phone\_update.value} ,

success: function(data){

let mas=JSON.parse(data);

if (mas['status']==1){

update\_status.onclick();

message\_update\_user\_date.style.display="block";

message\_update\_user\_date.innerHTML=mas['mess'];

let login\_user=document.getElementById("login\_user");

login\_user.innerHTML=mas['login'];

}

else{

message\_update\_user\_date.style.display="block";

message\_update\_user\_date.innerHTML=mas['mess'];

}

return false;

}

});

}

return false;

}

Так же, в личном кабинете пользователя реализована возможность менять пароль. Код программы представлен на листинге 4.

Листинг 4 – Код файла

<?php

session\_start();

include("connect.php");

$id=$\_SESSION['id\_user'];

$oldpass=$\_POST['oldpass'];

$newpass=$\_POST['newpass'];

$s="SELECT \* FROM `users` WHERE `id\_user`=$id AND `pass`=$oldpass";

$res=mysqli\_query($link, $s);

$n=mysqli\_num\_rows($res);

if ($n==0){

    echo 0;

}

else{

    $s1="UPDATE `users` SET `pass`=$newpass WHERE `id\_user`=$id";

    $r1=mysqli\_query($link,$s1);

    echo 1;

}

?>

Код JS – Смена пароля

function new\_password(){

    message\_new\_pass.style.display="none";

    message\_new\_pass.innerHTML="";

    let oldpass=document.getElementById("old\_pass").value;

    let newpass=document.getElementById("newpass").value;

    let newpass2=document.getElementById("newpass2").value;

    if (newpass!=newpass2){

        message\_new\_pass.style.display="block";

        message\_new\_pass.innerHTML="Новые пароли не совпадают";

    }

    else{

$.ajax({

url: 'update\_user\_pass.php',

method: 'post',

dataType: 'html',

data: {oldpass:oldpass,newpass:newpass} ,

success: function(data){

if(data=="0"){

message\_new\_pass.style.display="block";

message\_new\_pass.innerHTML="Не удалось сменить пароль";

}

else{

message\_new\_pass.style.display="block";

message\_new\_pass.innerHTML="Пароль успешно изменен";

}

return false;

}

});

}

return false;

}

Страница ежедневник, содержит три программы: 1 таблица куда выводятся, цели и задачи, дата и время, назначенные на эту цель, флаг исполнена либо нет, категория которой соответствует задача, кнопка удаления, кнопка редактирования задачи. Код программы представлен на листинге 5.

Листинг 5 – Таблица ежедневника

<?php

session\_start();

include("connect.php");

$id=$\_POST["id"];

$id\_user=$\_SESSION['id\_user'];

/\* $s=""; \*/

$s="SELECT \* FROM `Everybay` WHERE `id\_ever`=$id";

$res=mysqli\_query($link, $s);

$row=mysqli\_fetch\_array($res);

echo json\_encode($row);

?>

Код JS – Таблица ежедневник

function load\_table() {

  $.ajax({

    url: 'load\_table.php',

    method: 'post',

    dataType: 'html',

    data: "",

    success: function (data) {

      let d = JSON.parse(data);

      let s = "<table class='table' id='task\_history'>" +

        "<thead>" +

        "<tr>" +

        "<th scope='col'>№</th>" +

        "<th scope='col'>задача</th>" +

        "<th scope='col'>дата и время</th>" +

        "<th scope='col'>выполнено/нет</th>" +

        "<th scope='col'>категория</th>" +

        "<th scope='col'>кнопки действия</th>" +

        "</tr>" +

        "</thead>" +

        "<tbody id='tableE'>";

      for (let i = 0; i < d.length; i++) {

        if (d[i][5] == 0) {

          s += "<tr>" +

            "<th scope='row'>" + d[i][0] + "</th>" +

            "<td>" + d[i][3] + "</td>" +

            "<td>" + d[i][4] + "</td>" +

            "<td><input type='checkbox' id='inlineCheckbox1' value='option1' disabled><br></td>" +

            "<td>" + d[i][7] + "</td>" +

            "<td><button class='btn btn-primary' type='button' id='cnopDl' onclick='delet\_task(" + d[i][0] + ");'>Удалить событие</button>" +

            "<a href='update\_task\_data.php?id=" + d[i][0] + "' class='btn btn-primary' type='button' id='update' >Редактировать</button>" +

            "</td>" +

            "</tr>";

        }

        else {

          s += "<tr>" +

            "<th scope='row'>" + d[i][0] + "</th>" +

            "<td>" + d[i][3] + "</td>" +

            "<td>" + d[i][4] + "</td>" +

            "<td><input type='checkbox' id='inlineCheckbox1' value='option1' checked disabled><br></td>" +

            "<td>" + d[i][7] + "</td>" +

            "<td><button class='btn btn-primary' type='button' id='cnopDl' onclick='delet\_task(" + d[i][0] + ");'>Удалить событие</button>" +

            "<a href='update\_task\_data.php?id=" + d[i][0] + "' class='btn btn-primary' type='button' id='update' >Редактировать</button>" +

            "</td>" +

            "</tr>";

        }

      }

      s += "</tbody> </table>"

      let history\_table = document.getElementById("history\_table");

      history\_table.innerHTML = s;

    }

  });

}

Кнопка удаления задачи, представлена на листинге 6.

Листинг 6

<?php

include("connect.php");

$id\_task=$\_POST['id\_task'];

   $s="DELETE FROM `Everybay` WHERE `id\_ever`=$id\_task";

   $rez=mysqli\_query($link, $s);

?>

Код JS – удаление задачи

function delet\_task(id\_task) {

  $.ajax({

    url: 'delete\_tasck.php',

    method: 'post',

    dataType: 'html',

    data: { id\_task: id\_task },

    success: function (data) {

      load\_table();

    }

  });

}

Кнопка редактирования задачи. Представлена листинг 7.

Листинг 7 – Редактирование задачи

<?php

session\_start();

include("connect.php");

$task=$\_POST['task'];

 $my\_data=$\_POST['my\_data'];

 $itogi=$\_POST['itogi'];

 $id\_user=$\_SESSION['id\_user'];

 $name\_categ=$\_POST['name\_categ'];

$id=$\_POST["id"];

 $s="UPDATE `Everybay` SET `id\_categ`=$name\_categ,`task`='$task',`my\_data`='$my\_data',`itogi`=$itogi WHERE `id\_ever`=$id";

 mysqli\_query($link,$s);

 echo $s;

?>

Код JS – Кнопка редактирование

function update\_tas\_bd() {

var task = document.getElementById("task").value;

var my\_data = document.getElementById("my\_data").value;

var name\_categ = document.getElementById("kat").value;

var ch=document.getElementById("inlineCheckbox1");

let itogi=0;

if (ch.checked ==true){

itogi=1;

}

$.ajax({

url: 'update\_tas\_bd.php',

method: 'post',

dataType: 'html',

data: {task:task, my\_data: my\_data, name\_categ: name\_categ, id:id, itogi:itogi},

success: function (data) {

}

});

}

let submit\_update\_task = document.getElementById("submit\_update\_task");

submit\_update\_task.onclick = function () {

update\_tas\_bd();

document.location.href = "diary.php";

}

Добавления задачи, реализовано с помощью программного кода, представленного на листинге 8.

Листинг 8 – Добавление задачи

<?php

 session\_start();

 include("connect.php");

 $task=$\_POST['task'];

 $my\_data=$\_POST['my\_data'];

 $id\_user=$\_SESSION['id\_user'];

 $name\_categ=$\_POST['name\_categ'];

 $s="INSERT INTO `Everybay`( `id\_user`, `id\_categ`, `task`, `my\_data`, `itogi`)

 VALUES ($id\_user, $name\_categ, '$task', '$my\_data', 0)";

  $rez=mysqli\_query($link,$s);

  echo "Задача успешно добавлена!";

  //echo $s;

  ?>

Код JS – Добавление задачи

function add\_task() {

var message\_add\_taskc = document.getElementById("message\_add\_taskc");

var task = document.getElementById("task").value;

var my\_data = document.getElementById("my\_data").value;

var name\_categ = document.getElementById("kat").value;

if (task == 0 || my\_data == 0) {

message\_add\_taskc.style.display = "none";

message\_add\_notask.style.display = "block";

message\_add\_notask.innerHTML = "Добавте задачу и дату, время!";

}

else {

message\_add\_notask.style.display = "none";

$.ajax({

url: 'add\_task.php',

method: 'post',

dataType: 'html',

data: { task: task, my\_data: my\_data, name\_categ: name\_categ },

success: function (data) {

load\_table();

}

–––  });

document.getElementById("task").value = "";

document.getElementById("my\_data").value = "";

document.getElementById("kat").value = "";

message\_add\_taskc.style.display = "block";

message\_add\_taskc.innerHTML = "Задача успешно добавлена!";

}

}

Так же, на странице ежедневник предоставлена для пользователя возможность выводить данные о задачи, за определённый промежуток времени, в таблицу с помощью программного кода, представленного на листинге 9.

Листинг 9 – Выбор задач за определённый промежуток времени

<?php

 session\_start();

 include("connect.php");

 $id\_user=$\_SESSION['id\_user'];

 $start\_data=$\_POST['start\_data'];

 $finish\_data=$\_POST['finish\_data'];

 $name\_categ=$\_POST['name\_categ'];

 $s="SELECT \* FROM `everybay`INNER JOIN Category ON Category.id\_categ=Everybay.id\_categ WHERE (`my\_data`>='$start\_data' AND my\_data<='$finish\_data') AND`id\_user`=$id\_user";

  $rez=mysqli\_query($link,$s);

  $mas=array();

 $n=mysqli\_num\_rows($rez);

 for($i=0;$i<$n;$i++){

    $row=mysqli\_fetch\_row($rez);

    $mas[$i]=$row;

 }

  echo  json\_encode($mas);

?>

Код JS – Вывод задач за определённый промежуток времени

add\_time();

load\_table();

function  add\_time(){

let start\_data= document.getElementById("start\_data").value;

let finish\_data=document.getElementById("finish\_data").value;

$.ajax({

url: 'add\_period\_of\_time.php',

method: 'post',

dataType: 'html',

data: { start\_data: start\_data ,finish\_data:finish\_data },

success: function (data) {

let d1 = JSON.parse(data);

let s = "<table class='table'>" +

"<thead>" +

"<tr>" +

"<th scope='col'>№</th>" +

"<th scope='col'>задача</th>" +

"<th scope='col'>дата и время</th>" +

"<th scope='col'>выполнено/нет</th>" +

"<th scope='col'>категория</th>" +

"</tr>" +

"</thead>" +

"<tbody id='tableE'>";

for (let i = 0; i < d1.length; i++) {

s += "<tr>" +

"<th scope='row'>" + d1[i][0] + "</th>" +

"<td>" + d1[i][3] + "</td>" +

"<td>" + d1[i][4] + "</td>";

if(d1[i][5]!=1){

s+= "<td>Не выполнено</td>" ;}

else{

s+= "<td>Выполнено</td>" ;}

s+="<td>" + d1[i][7] + "</td>" + "</tr>";

}

s += "</tbody> </table>"

let task\_history = document.getElementById("task\_history");

task\_history.innerHTML = s;

}

});

}

function tab\_time(){

if(start\_data!="" && finish\_data!==""){

load\_table();

start\_data.value="";

finish\_data.value="";

}

};

Следующая страница «отзывы», попасть на которую можно инициировав кнопку «Оставить отзыв, или рекомендации для улучшения сайта» из личного кабинета пользователя. Код программы представлен на листинге 10.

Листинг 10 – Добавление отзыва, либо рекомендаций

<?php

 session\_start();

 include("connect.php");

$id\_user=$\_SESSION['id\_user'];

 $foto=$\_POST['foto'];

 $name\_inf=$\_POST['name\_inf'];

 $informat=$\_POST['informat'];

 $email2=$\_POST['email2'];

 $s="INSERT INTO `information`(`id\_user`, `name\_inf`, `informat`, `foto`, `email`) VALUES ($id\_user,'$name\_inf','$informat','$foto','$email2')";

  $rez=mysqli\_query($link,$s);

?>

Код JS - Добавление отзыва, либо рекомендаций

function add\_reviews() {

  var message\_reviews = document.getElementById("message\_reviews");

  var message\_no\_reviews = document.getElementById("message\_no\_reviews");

  var foto = document.getElementById("foto").value;

  var name\_inf = document.getElementById("name\_inf").value;

  var informat = document.getElementById("informat").value;

  var email2 = document.getElementById("email2").value;

  informat = informat.trim();

  foto = foto.trim();

  if (foto.length == 0 || informat.length == 0) {

    message\_reviews.style.display = "none";

    message\_no\_reviews.style.display = "block";

    message\_no\_reviews.innerHTML = "Нобходимо добавить фото и текст отзыва!!"

  }

  else {

    message\_no\_reviews.style.display = "none";

    $.ajax({

      url: 'form\_rexiews\_bd.php',

      method: 'post',

      dataType: 'html',

      data: { name\_inf: name\_inf, informat: informat, foto: foto, email2: email2 },

      success: function (data) {

        reviews\_bd();

      }

    });

    document.getElementById("foto").value = "";

    document.getElementById("name\_inf").value = "";

    document.getElementById("informat").value = "";

    document.getElementById("email2").value = "";

    message\_reviews.style.display = "block";

    message\_reviews.innerHTML = "Отзыв успешно добавлен!!"

  }

}

Визуализация, отзывов на листе реализован с помощью программного кода, представленного на листинге 11.

Листинг 11 - Визуализация, отзывов

<?php

include("connect.php");

session\_start();

$s="SELECT \* FROM `information` ORDER BY `id` DESC";

$res=mysqli\_query($link, $s);

 $mas=array();

$n=mysql\_num\_\_rows($res)

 for ($i=0; $i<$n;$i++){

    $row=mysqli\_fetch\_array($res);

     $mas[$i]= $row;

    }

  echo json\_encode($mas);

?>

Код JS – Вывод карточек отзывов на лис

function reviews\_bd() {

$.ajax({

url: 'reviews\_bd.php',

method: 'post',

dataType: 'html',

data: "",

success: function (data) {

//alert(data);

var z = JSON.parse(data);

var s = "";

for (var i = 0; i < z.length; i++) {

s+="<div class='col'>"+

"<svg class='bd-placeholder-img card-img-top' width='2%' height='2%' xmlns='http://www.w3.org/2000/svg' aria-label=' preserveAspectRatio='xMidYMid slice role='img'><img src='img/info\_img/" + z[i]['foto'] + "'></title><rect width='1%' height='1%' fill='#55595c'></rect><text x='1%' y='1%' fill='#eceeef' dy='.3em'>Ваш коментарий (отзыв)</text></svg>"+

"<div class=card-body'>"+

"<h3 class='card-text'>" + z[i]['name\_inf'] + "</h3>"+

"<div class=card-body'>"+

"<h5 class='card-text'>" + z[i]['informat'] + "</h5>"+

"<div class=card-body'>"+

"<p class='card-text'>" + z[i]['email'] + "</p>"+

"<div class='d-flex justify-content-between align-items-center'>"+

"<div class='btn-group'>"+

"<button class='btn btn-primary' type='button' id='del\_revi' onclick='reviews\_delet(" + z[i][0] + ");'>Удалить отзыв</button>"+

"<p><a class='btn btn-secondary' id='ret' href='index.php' role='button'>Вернуться на главную страницу &raquo;</a></p>"+

"</div>"+

"</div>"+

"</div>"+

"</div>"+

"</div>"+

"</div>";

}

let detalis\_nformation = document.getElementById("teviews");

detalis\_nformation.innerHTML = s;

}

});

}

Кнопка удаление отзыва реализована с помощью программного кода на листинге 12.

Листинг 12 – Кнопка удаления

<?php

include("connect.php");

session\_start();

$id\_user=$\_SESSION['id\_user'];

$id=$\_POST['id'];

$s="DELETE FROM `information` WHERE `id`=$id AND `id\_user`=$id\_user";

mysqli\_query($link,$s);

?>

Код JS – Кнопка удаления

function reviews\_delet(id) {  let message\_del\_reviews = document.getElementById("message\_del\_reviews");

   message\_del\_reviews.style.display = "none";

  message\_del\_reviews.innerHTML = "Возможно удалять только свой отзыв!";

  $.ajax({

    url: 'reviews\_delet\_bd.php',

    method: 'post',

    dataType: 'html',

    data: { id: id },

    success: function (data) {

      message\_del\_reviews.style.display = "block";

      reviews\_bd();

    }

  });

}

## 3.3 Руководство пользователя

При запуске Web – приложения «Ежедневник» открывается главная страница сайта, где представлено вашему вниманию «Карусель» меняющиеся иллюстрации с текстом. Ниже реализованы цифровые часы, справа под часами, окно с прогнозом погоды, слева предложен к ознакомлению с сайтом, видео ролик, как пользовать веб-приложением. Ниже расположены три актуальных отзыва. В подвале сайта находятся ссылки на соц-сети.

Доступ к ежедневнику, личному кабинету и к странице отзывы, открывается авторизованному пользователю. В шапке сайта находится кнопка «Авторизоваться» кликнув по ней, открывается модальное окно, заполнение которого предоставляет возможность полноценно пользоваться веб-приложения. Иллюстрация, представ лена на рисунке 5.

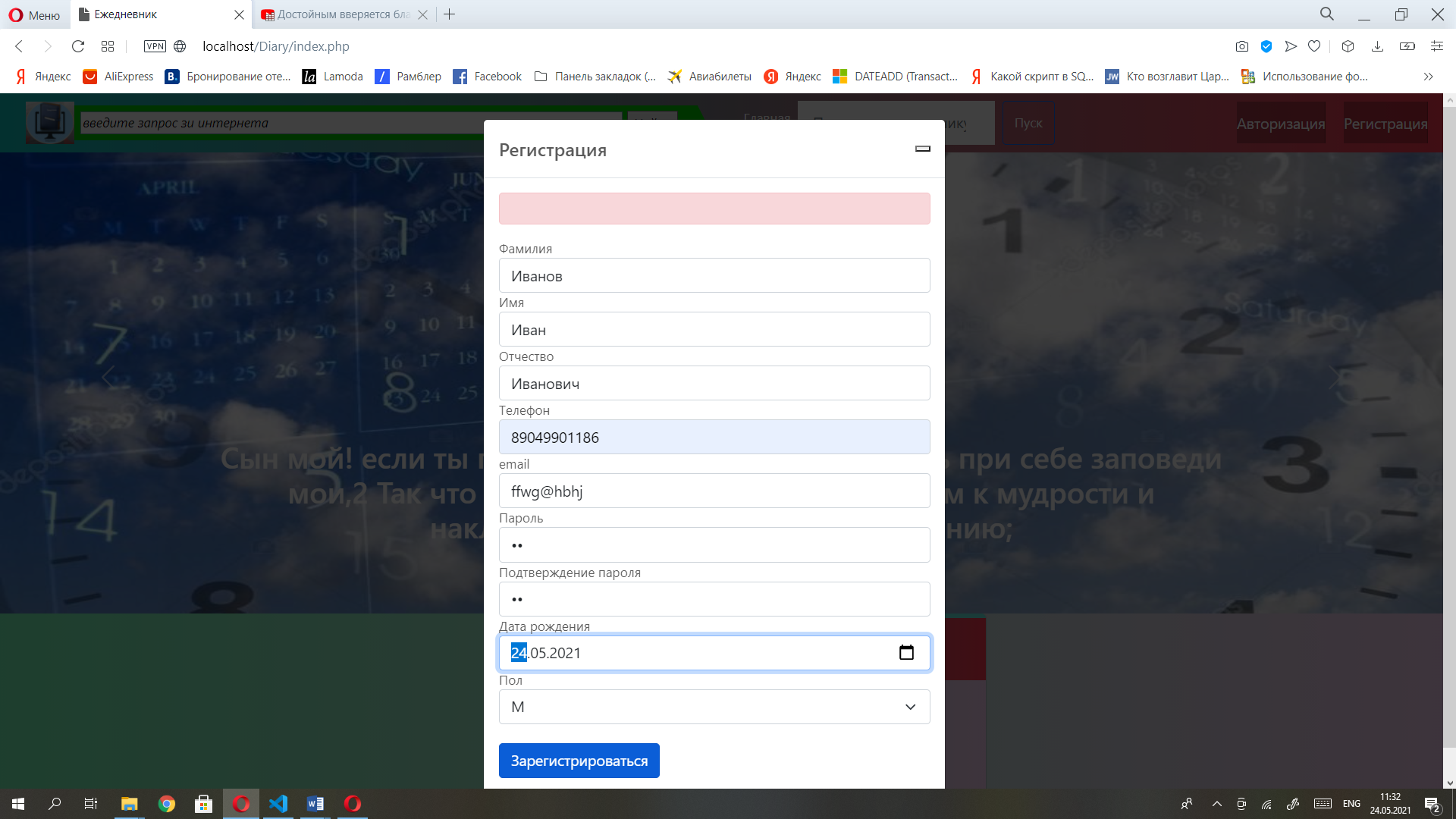


Рисунок 5 - Форма регистрации

Зарегистрированный пользователь может авторизоваться, кликнув по кнопке «Авторизоваться» которая находится рядом с кнопкой «Регистрироваться» , в модальном окне заполните поле «Логин» - это то что вы занесли в поле Телефон, при регистрации. Второе поле пароль, который вы продуммали и занесли при регистрации, образец на рисунке 12.

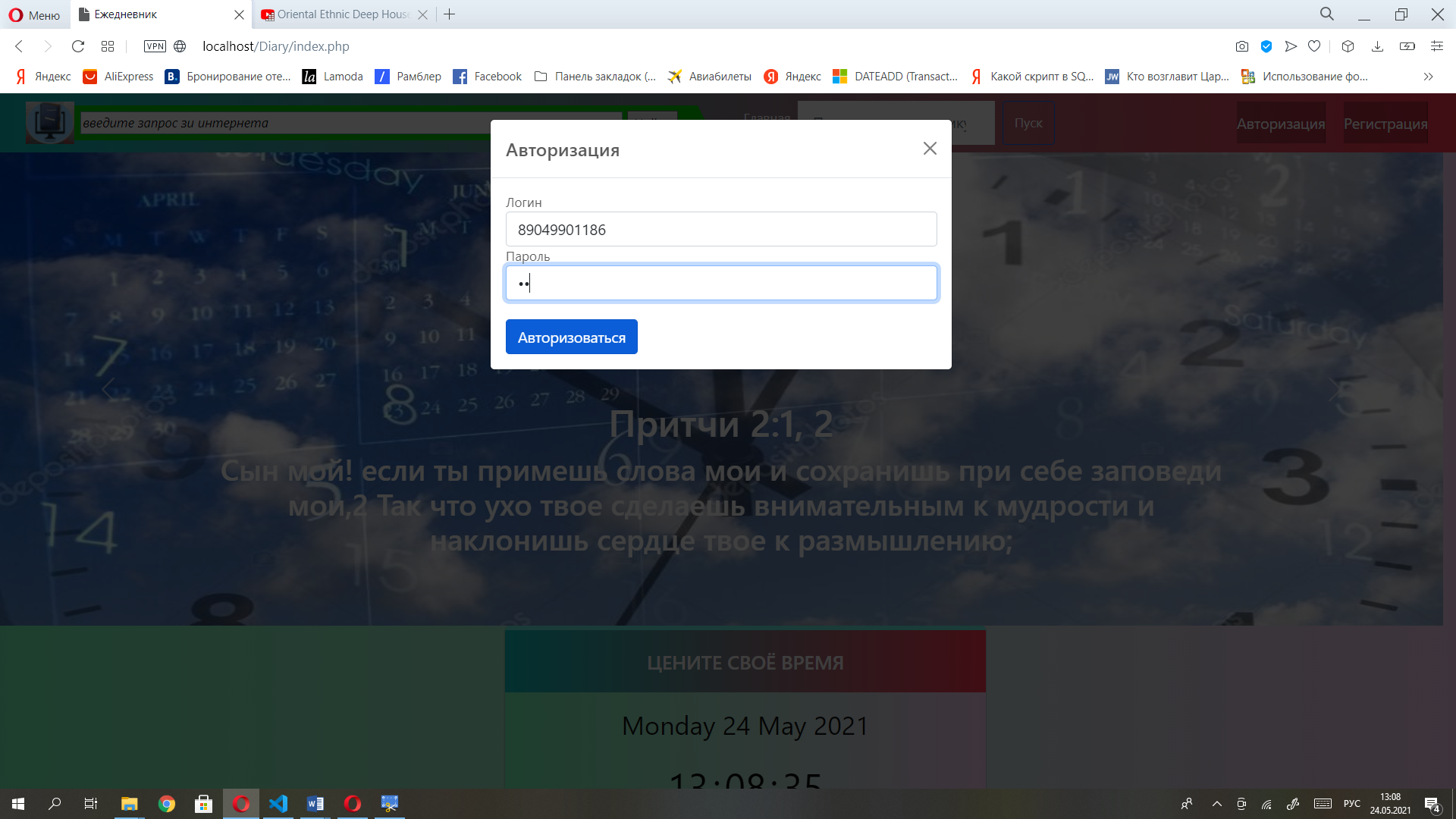


Рисунок 12-Авторизация

При авторизации предоставляется возможность проихводить действия в личном кабинете, занасить свои цели, задачи, союытия , переход на эти страницы осуществляется спомощю кнопок: «Ежедневник», и клике на ваш логин, с право в шапке сайта. Изображено на рисунке 13- Шапка сайта с сылками на страници: «Ежедневник», «Личный кабинет», «Выход из сессии», «Вернуться на главную страницу»,

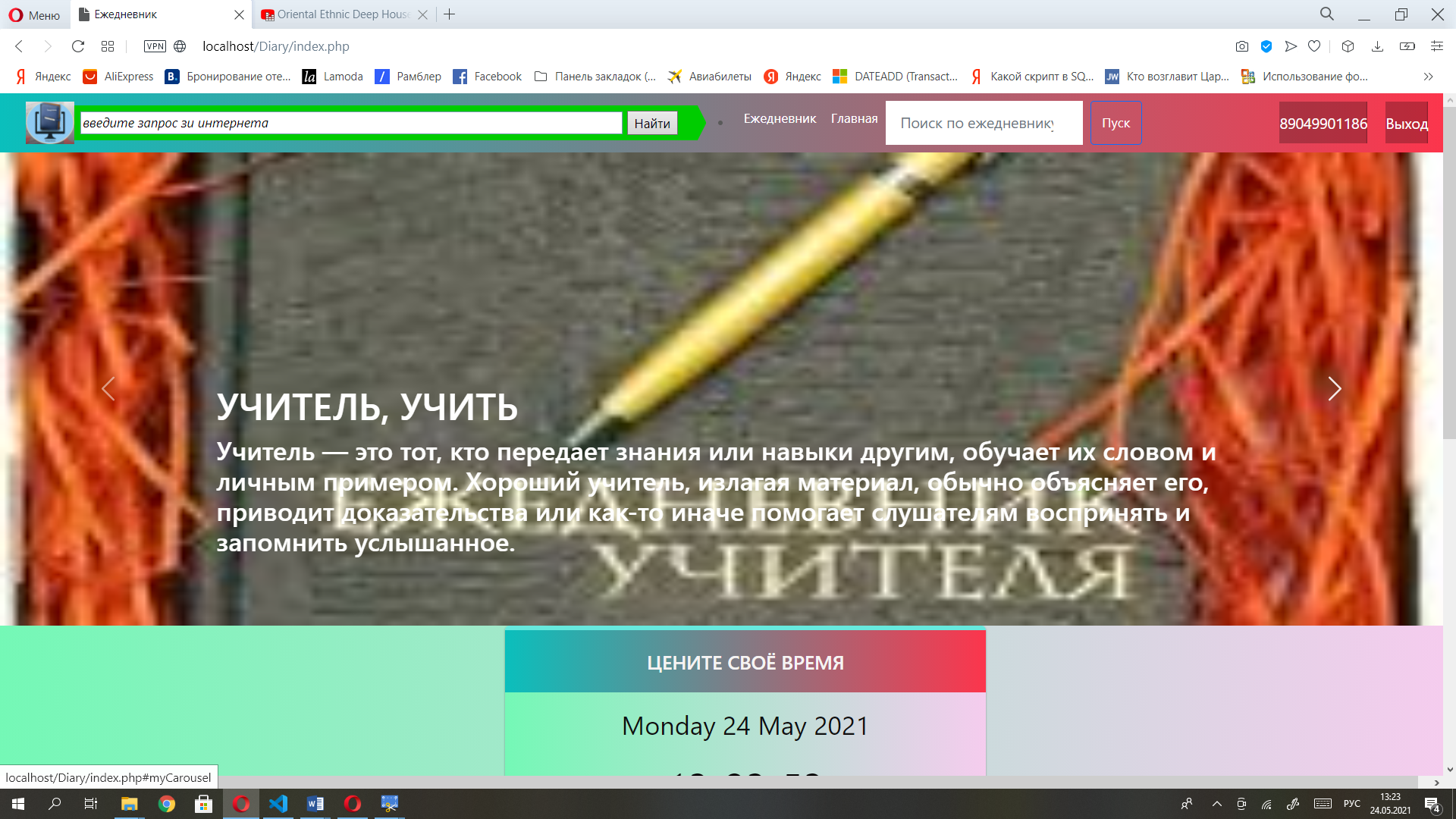


Рисунок 13- Шапка сайта с сылками на страници

### Страница ежедневника открывается при клике на ссылку «Ежедневник» в шапке веб – приложаения. Добавляем события на форме «Добавляем цели, задачи, события!!!» Заполняем поля: задачи, дату и время, когда вы планируете её реализовать и категорию, к какой вы её определите, запускаем программу кнопкой «Добавить событие» рисунок 14, 15.

### 

### Рисунок 14 –Создание задачи

### 

### Рисунок 15 – Цель добавлена в базу данных и отображена в Вышей таблице

### Реализована возможность редактировать свои цели, события. При клике на соответствующею кнопку, показано на рисунке 16

### 

### Рисунок 16 – Кнопка редактирования

### Открывается страница с формой, там производится редакция своей записи, при клике по кнопке «редактировать», данные сохраняются в БД. Форма на рисунке 17.

### 

### Рисунок 17 – Форма редактирование событий

### На этой страницы «Ежедневник» сформирована возможность выбора событий за определённый, пользователем, промежуток времени и вывод этих данных в соответствующею таблицу «Ваша таблица целей, задач, событий». Код программы представлен на листинге 18. Стрелки показывают на два поля ввода промежутка времени, с какой даты и времени по кое, необходимо вывести событие в таблицу «Ваши цели, задачи, события». Выбрав необходимый промежуток времени, с помощью выпадающего календаря, кликаем по кнопки «Вывод» Кнопка «Назад», возвращает все ваши события в таблицу «Ваши цели, задачи, события».

### Рисунок 18 – События за промежуток времени

### Следующий лист, веб-приложения, личный кабинет пользователя. В нём находятся поля формы с вашими данными 3ст. Кликнув по кнопке с надписью: «Редактировать личные данные» 1ст., она поменяет своё имя на «Отменить редактирование» 1ст. и форма с полями редактирования станет активной можно производить редактирование. Изменённые данные сохраняем, кликая по кнопке «Сохранить изменения» 2ст. Также можно отменить редактирование 1ст. Рисунок 19.

### 

### Рисунок 19-Редактирование личных данных

### Возможно сменить пароль, кликнув по кнопке «Сменить пароль», открывается модальное окно где меняете пароль. На рисунке 20.

### 

### Рисунок 20 – Модальное окно смена пароля

### С помощью кнопки «Оставить отзыв, или рекомендации по улучшению сайта», осуществляется переход на страницу отзывы.

### Страница «Отзывы» реализует потребность пользователя, оставить свой отзыв, комментарий и совет. Разработана возможность ознакомиться с другими комментариями. А также удалять свои комментарии. Изображено на рисунке 21.

### 

### Рисунок 21 – Форма отзывов

### Заполняем форму, кликаем по кнопке «Отправить отзыв» и он отображается на этой странице. Можно удалить только свой отзыв. С этой страницы можно также, перейти на главную страницу кликнув по кнопки «Вернуться на главную страницу». 1ст. поиск по ежедневнику, осуществляется только на странице ежедневник, поиск запланированных событий, 2ст.поиск в интернете. Представлено на рисунке 22

### Рисунок 22 – Страница отзывы.

# ЗАКЛЮЧЕНИЕ

### В рамках курсовой работы, был разработан программный веб продукт «Ежедневник», с использованием СБД MY SQL. При реализации программного продукта было проведено исследование предметной области, проведен анализ прототипов программного продукта, изучена среда разработки VS Code, разработан интерфейс Web-проекта с применением интерфейсного Фреймворка Bootstrap и формального языка описания внешнего вида документа - CSS. Так же стоит отметить, что обмен данными, страниц с базой данных MY SQL осуществлялся с помощью метода AJAX. Этот метод, обмена данными без перезагрузки страницы, был изучен и успешно применён. Все страницы выполнены в одном стиле, разработано руководство пользователя, проведено тестирование и отладка.

### В результате разработанной курсовой работы, поставленные цели и задачи выполнены. Полезное веб – приложение «Ежедневник» готов к использованию. Он поможет пользователю эффективно использовать рабочее и свободное время, используя возможности этого приложения, сохранять, и демонстрировать, свои намеченные задачи, цели, события и комментарии.

### При разработке программного продукта возникали трудности отказ в работе платформы OPE SERVER, эта проблема была решена с помощью установке другого сервера.

### В процессе разработке программного продукта и написание курсовой работы, были получены полезные навыки: программирования на языках HTML, PHP, JS, также понимание структуры написания тэгов кода и js скриптов и как реализуется связи между файлами php, css, js, и внешними библиотеками, понимание логики обмена данными страницы и базы данных.

В дальнейшем планируется провести более детальную обработку ошибок, интерфейса и добавить возможности для функционирования администратора и расширять полезный функционал для пользователя. Цели достигнуты, задачи выполнены.

.

# СПИСОК ЛИТЕРАТУРЫ

**Основные источники:**

1.     Советов Б.Я. Базы данных [Текст]: учебник для СПО /Б. Я. Советов, В.В. Цехановский, В.Д.Чертовский .- 2 изд.- Москва : Юрайт, 2017. – 463 с.-(Профессиональное образование). (Гриф).

2.     Казанский А.А. Программирование на visual c # 2013[Текст] : учебное пособие для СПО / А.А. Казанский.- Москва : Юрайт, 2017. - 191 с. –(Профессиональное образование). (Гриф).

**Дополнительные источники:**

1.     Организационное и правовое обеспечение информационной безопасности [Текст]: учебник и практикум для СПО / под ред. Т. А.Поляковой ,А. А., Стрельцова. -  Москва: Юрайт, 2017. – 325 с. –(Профессиональное образование).

2.     Зуб А.Т. Управление проектами [Текст]: учебник и практикум для СПО / А.Т. Зуб. – Москва: Юрайт, 2017. -  422 с. –(Профессиональное образование).(Гриф).

# 

# ПРИЛОЖЕНИЕ