

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ ДОПОЛНИТЕЛЬНОГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ ЦЕНТР
ПРОФЕССИОНАЛЬНЫХ КВАЛИФИКАЦИЙ И СОДЕЙСТВИЯ
ТРУДОУСТРОЙСТВУ «ПРОФЕССИОНАЛ»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к итоговой аттестационной работе на тему

**«Разработка web-ресурса с использованием технологий HTML, CSS,
JavaScript»**

на примере web-ресурса: <https://ade.p-host.in/Diploma/index.html>

слушателя **Зотова Дениса Владимировича** группы №: **209**

программы профессиональной переподготовки

«Frontend разработка»

Москва, 2023

Оглавление

ПОСТАНОВКА ЗАДАЧИ И ПЛАН РАБОТЫ.	3
ОСНОВНАЯ ЧАСТЬ.	4
НАЗНАЧЕНИЕ ВЕБ-РЕСУРСА.....	4
ОПИСАНИЕ ЭТАПОВ РАЗРАБОТКИ, ОПИСАНИЕ ФУНКЦИОНАЛА С ПРИЛОЖЕНИЕМ ЛИСТИНГОВ ИСХОДНОГО ПРОГРАММНОГО КОДА ОСНОВНОЙ ФУНКЦИИ, СТРУКТУРНЫХ МОДУЛЕЙ, ПРИМЕР КОДА-РАЗМЕТКИ	4
СПИСОК ЛИТЕРАТУРЫ.....	15

ПОСТАНОВКА ЗАДАЧИ И ПЛАН РАБОТЫ.

Задачей является создание веб-ресурса «**Моё портфолио**». Работа серверной части в данном проекте минимальна и представлена тремя формами: «Форма регистрации», «Форма авторизации» и «Личный кабинет». Проект со стороны клиента представлен формами: «Мой сайт-визитка», «Обо мне», «Калькулятор услуг», «Мои контакты для связи». Внешний вид сайта в виде ряда кнопок и ссылок с гиперссылками. Личный кабинет пользователя имеет простой вид. Попасть в личный кабинет возможно после успешной регистрации в форме «Форма регистрации» и авторизации в форме «Форма авторизации».

Задачи к наполнению сайта: наличие информации о частном предпринимателе, портфолио, контакты с схемой проезда к офису, каталог выполненных работ, личный кабинет.

План работ:

1. Подключение фреймворка Bootstrap.
2. Подключение фреймворка Yandex Maps.
3. Создание базы данных на хостинге.
4. Верстка страниц сайта.
5. Подключение страниц сайта к базе данных.
6. Связывание страниц между собой гиперссылками.

ОСНОВНАЯ ЧАСТЬ.

Назначение веб-ресурса.

Веб-ресурс предназначен для предоставления информации потенциальным клиентам о частном предпринимателе, ценовой политики по разработке Web-сайтов. Также для действующих клиентов разработан личный кабинет пользователя, в котором они могут посмотреть информацию о себе для обратной связи.

Описание этапов разработки, описание функционала с приложением листингов исходного программного кода основной функции, структурных модулей, пример кода-разметки

Этап разработки начинался с создания проекта с использованием фреймворка Bootstrap ver. 4.6 (далее “фреймворк”). Перед началом работы над проектом необходимо подключить фреймворк.

```
<link  
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css"  
rel="stylesheet"  
integrity="sha384-  
zCbkRUCUGaJDkqS1kPbPd7TveP5iyJE0EjAuZQTgFLD2ylzuqKfdKlfg/eSrtxUkn"  
crossorigin="anonymous"/>
```

Листинг 1.

Веб ресурс создавался с использованием системы сеток фреймворка “Container”, “Container-fluid”, а также дополнительных готовых компонентов фреймворка. Создается база данных на стороне хостинга для получения, записи данных пользователей.

В проект веб ресурса входят: 1 файл в формате html и 1 файл в формате css в корне проекта, 4 файла в формате php и 2 файла в формате html в папке проекта php, 3 файла в формате html и 1 файл в формате css в папке pages, 1 файл в формате jpg в папке img.

Страницы веб проекта адаптивны и могут открываться на разных устройствах с разным разрешением экрана.

При переходе по адресу <https://ade.p-host.in/Diploma/index.html> пользователь попадает на главную страницу сайта, в верхней части которой расположено меню для перехода на другие страницы веб ресурса.

```
<div class="container">
  <nav>
    <div class="logo">
      <a href="https://github.com/deniszotov1978" target="_blank">Моё
портфолио:</span></a>
    </div>
    <ul>
      <li><a href="https://ade.p-host.in/Diploma/index.html">ГЛАВНАЯ
СТРАНИЦА</a></li>
      <li><a href="https://ade.p-host.in/Diploma/pages/about.html"
target="_blank">ОБО МНЕ</a></li>
      <li><a href="https://ade.p-host.in/Diploma/pages/calc.html"
target="_blank">СТОИМОСТЬ УСЛУГ</a></li>
      <li><a href="https://ade.p-host.in/Diploma/pages/contact.html"
target="_blank">КОНТАКТЫ</a></li>
    </ul>
    <div class="buttons">
      <a href="https://ade.p-host.in/Diploma/php/auth.html" target="_blank"
class="login">Авторизация</a>
      <a href="https://ade.p-host.in/Diploma/php/register.html" target="_blank"
class="btn">Регистрация</a>
    </div>
  </nav>
  <div class="content">
    <h2>Привет!<br>Это я - начинающий Web-программист</h2>
    <p>Сделаю Ваш проект привлекательным и успешным.<br>Вы можете
посмотреть мои работы на GitHub.</p>
  </div>
  <div class="link">
    <a href="https://ade.p-host.in/Diploma/pages/contact.html" target="_blank"
class="hire">Найми меня</a>
  </div>
</div>
```

Листинг 2.

На главной странице расположены кнопки, ссылки на страницы данного проекта с информацией о частном предпринимателе, контактами и

временем работы. Добавлена интерактивная карта местоположения офиса компании.

```
<script
  type="text/javascript"
  charset="utf-8"
  async
  src="https://api-
maps.yandex.ru/services/constructor/1.0/js/?um=constructor%3A7e1fff857885ac925c6f
e1918b57172adc200ca3818cde496f9c63da7ff0a3b1&width=100%25&height=618&
lang=ru_RU&scroll=false"
></script>
```

Листинг 3.

При нажатии на кнопку «Моё портфолио:» открывается страница **GitHub** по адресу <https://github.com/deniszotov1978> появляется список выполненных работ.

Далее следует кнопка «Обо мне» с ссылкой на страницу с кратким описанием владельца сайта.

Следующая кнопка «Стоимость услуг» с ссылкой на страницу «Калькулятор услуг». Данный инструмент выполнен с применением HTML с отдельным файлом CSS и JavaScript.

```
<script>
  const result = document.querySelector('#result'),
        expression = document.querySelector('#expression'),
        num = document.querySelectorAll('.number:not(.equals)'),
        operation = document.querySelectorAll('.operation'),
        equals = document.querySelector('.equals'),
        clear = document.querySelector('#clear'),
        ce = document.querySelector('#ce');

  let ex = ''; // строка выражения для результата
  result.innerHTML = '0';

  function clickN() { // нажатие на цифру
    if(!ex || typeof(ex) === 'number' || ex === '0') {
      expression.innerHTML = this.id;
      ex = this.id;
    } else {
      expression.innerHTML += this.id;
      ex += this.id;
    }
  }
```

```

        result.innerHTML = ex.split(/\|\/|\*|\+|-|=\/).pop();
        checkLength(result.innerHTML);
    };

    function click0() { // нажатие на операцию
        if(!ex) {
            return;
        }
        ex = ex.toString().replace(/=/, '');
        if (ex.match(/\|\/|\*|\+|-|=\/)) {
            ex = eval(ex).toString();
        }
        expression.innerHTML = expression.innerHTML.replace(/=/, '') +
this.id;
        ex += this.id;
        result.innerHTML = this.id;
    };

    Array.from(num).forEach(function(element) { // назначьте соответствующую
функцию всем числам и операциям
        element.addEventListener('click', clickN);
    });

    Array.from(operation).forEach(function(element) {
        element.addEventListener('click', click0);
    });

    // очистить все по клику
    clear.addEventListener('click', () => {
        result.innerHTML = '';
        expression.innerHTML = '';
        ex = '';
    });

    // очистить последнюю запись по клику
    ce.addEventListener('click', () => {
        if (!expression.innerHTML.match(/=$/)) {

            expression.innerHTML = doCE(expression.innerHTML);
            ex = doCE(ex);
            result.innerHTML = 0;

            function doCE(arg) {
                arg = arg.split(/([\|\/|\*|\+|-|=])/g);
                arg.splice(-1, 1);
                return arg.join('');
            }
        }
    });

```

```

// рассчитать все это
equals.addEventListener('click', ()=> {
    if (!ex) {
        result.innerHTML = '0';
    } else {
        ex = eval(ex);
        expression.innerHTML += '=';
        result.innerHTML = trim12(ex);
    }
})

function checkLength(arg) { // если мы введем слишком длинное число
    if (arg.toString().length > 14) {
        expression.innerHTML = 'number too long'.toUpperCase();
        result.innerHTML = '0';
        ex = '0';
    }
}

function trim12(arg) { // если мы посчитаем слишком длинное число
    if (arg.toString().length > 14) {
        ex = parseFloat(arg.toPrecision(12));
        if (ex.toString().length > 14) {
            ex = ex.toExponential(9);
        };
        return ex;
    } else {
        return arg;
    }
}
</script>

```

Листинг 4.

В веб-ресурсе представлена краткая информация о компании и фактическом адресе офиса на странице «Мои контакты для связи». Попасть на эту страницу можно нажав на ссылку «Контакты» либо нажав на кнопку «Найми меня». На данной странице, кроме подключенного фреймворка, подключается Yandex Maps.


```

<script
  type="text/javascript"
  charset="utf-8"
  async
  src="https://api-
maps.yandex.ru/services/constructor/1.0/js/?um=constructor%3A7e1fff857885ac925c6f
e1918b57172adc200ca3818cde496f9c63da7ff0a3b1&width=100%25&height=618&
lang=ru_RU&scroll=false"
></script>

```

Листинг 5.

Последними кнопками являются «Авторизация» и «Регистрация». Прежде, чем авторизоваться в личном кабинете, необходимо пройти успешную регистрацию, нажав на кнопку «Регистрация». В открывшейся странице регистрации необходимо заполнить все поля и нажать кнопку «Зарегистрироваться». После этого пользователь может перейти на страницу авторизации, нажав кнопку «Авторизация». На странице «Форма авторизации» пользователь вводит e-mail и пароль. При успешном прохождении авторизации, пользователь попадает на страницу «Личный кабинет». На данной странице находится информация, заполненная пользователем при регистрации. Имеются 2 кнопки «Редактировать», напротив метки «Имя:» и «Фамилия:» при помощи которых ранее зарегистрированный пользователь может изменить свою Фамилию и Имя. Отправка исправленной формы на сервер происходит непосредственно в файле (листинг 6) при нажатии на кнопки «Сохранить» любой из меток. Исправленные данные сохраняются в базу данных на сервере хостинга. Пароль для базы данных при регистрации шифруется (6.1), при авторизации пароль дешифруется. Вся обработка данных происходят в файлах php.

```

<?php
session_start();
header('Content-Type: text/html; charset=utf-8');

$mysqli = mysqli_connect("localhost", "epwxohxw_0209db", "1qazxsw23",
"epwxohxw_0209db");

if ($mysqli == false) {

```

```

    print("error");
} else {
    $email = trim(mb_strtolower($_POST["email"]));
    $pass = trim($_POST["pass"]);

    $result = $mysqli->query("SELECT * FROM `user` WHERE `email`='$email'");
    $result = $result->fetch_assoc();
    $pass_hash = $result["pass"];

    if (password_verify($pass, $pass_hash)) {
        echo "success";
        $_SESSION["id"] = $result["id"];
        $_SESSION["name"] = $result["name"];
        $_SESSION["lastname"] = $result["lastname"];
        $_SESSION["email"] = $result["email"];
    } else {
        echo "rejected";
    }
}
}

```

Листинг 6.

```

<?php session_start(); ?>
<!DOCTYPE html>
<html lang="ru">
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css"
integrity="sha384-
zCbkRcUGaJdkqS1kPbPd7TveP5iyJE0EjAuZQTgFLD2ylzuqKfdK1fG/eSrtxUkn"
crossorigin="anonymous" />
<title>Личный кабинет</title>
<style>
    body {
        background-color: white;
        font-size: 1.2rem;
    }

    p {
        font-weight: bold;
    }

    span {
        margin-left: 0.5rem;
    }

    p>span:nth-child(1) {

```

```

        font-style: italic;
        color: black;
    }
    .btn-danger{
        background-color: #042548;
        color: white;
        border: 1px solid #F116FF;
    }
    .btn-warning{
        background-color: #F116FF;
        color: white;
        border: 1px solid #042548;
    }
    .btn-success{
        background-color: #042548;
        color: white;
        border: 1px solid #F116FF;
    }
    .mx-auto{
        color: red;
    }
</style>
</head>

<body>
    <div class="container mt-5">
        <div class="row">
            <div class="col-9 mx-auto">

                <p>Id: <span><?php echo $_SESSION["id"]; ?></span></p>

                <p>Имя:
                    <span><?= $_SESSION["name"]; ?></span>
                    <span class="edit-btn btn btn-
danger">Редактировать</span>
                    <span class="save-btn btn btn-warning" hidden
data-item="name">Сохранить</span>
                    <span class="cancel-btn btn btn-success"
hidden>Отменить</span></p>
                <p>Фамилия:
                    <span><?php echo $_SESSION["lastname"]; ?></span>
                    <span class="edit-btn btn btn-
danger">Редактировать</span>
                    <span class="save-btn btn btn-warning" hidden
data-item="lastname">Сохранить</span>
                    <span class="cancel-btn btn btn-success"
hidden>Отменить</span></p>
                <p>E-mail: <span><?php echo $_SESSION["email"];
?></span></p>

```

```

        </div>
    </div>
</div>

<script>
    let edit_buttons = document.querySelectorAll(".edit-btn");
    let save_buttons = document.querySelectorAll(".save-btn");
    let cancel_buttons = document.querySelectorAll(".cancel-btn");

    for (let i = 0; i < edit_buttons.length; i++) {
        let inputValue =
edit_buttons[i].previousElementSibling.innerText;

        edit_buttons[i].addEventListener("click", () => {
            edit_buttons[i].previousElementSibling.innerHTML =
`<input type="text" value="${inputValue}">`;
            save_buttons[i].hidden = false;
            cancel_buttons[i].hidden = false;
            edit_buttons[i].hidden = true;
        })

        cancel_buttons[i].addEventListener("click", () => {
            edit_buttons[i].previousElementSibling.innerText =
inputValue;

            save_buttons[i].hidden = true;
            cancel_buttons[i].hidden = true;
            edit_buttons[i].hidden = false;
        })

        save_buttons[i].addEventListener("click", async () => {
            let newInputValue =
edit_buttons[i].previousElementSibling.firstElementChild.value;

            edit_buttons[i].previousElementSibling.innerText =
newInputValue;

            save_buttons[i].hidden = true;
            cancel_buttons[i].hidden = true;
            edit_buttons[i].hidden = false;

            let formData = new FormData();
            formData.append("value", newInputValue);
            formData.append("item", save_buttons[i].dataset.item);

            let response = await fetch("lk_obr.php", {
                method: 'POST',
                body: formData
            });
        })
    }

```

```

    }
    </script>

    <script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js
" integrity="sha384-
fQybjgWLrvvRgtW6bF1B7jaZrFsaBXjsOMm/tb9LTS58ONXgqbR9W8oWht/amnpF"
crossorigin="anonymous"></script>
</body>

</html>

```

Листинг 6.1

Личный кабинет пользователя реализован с использованием HTML, PHP, JavaScript. На странице пользователя в поля ФИО приходит информация со стороны сервера, и подставляется в поля.

```

<?php
session_start();
header('Content-Type: text/html; charset=utf-8');

$mysqli = mysqli_connect("localhost", "epwxoxxw_0209db", "1qazxsw23",
"epwxoxxw_0209db");

if ($mysqli == false) {
    print("error");
} else {
    $inputValue = $_POST["value"];
    $item = $_POST["item"];
    $id = $_SESSION["id"];

    $mysqli->query("UPDATE `user` SET `{$item}`='{$inputValue}' WHERE `id`='{$id}'");
    $_SESSION[$item] = $inputValue;
}

```

Листинг 7.

```
<?php
header('Content-Type: text/html; charset=utf-8');

$mysqli = mysqli_connect("localhost", "epwxoxxw_0209db", "1qazxsw23",
"epwxoxxw_0209db");

if ($mysqli == false) {
    print("error");
} else {
    // print("—единение установлено успешно");

    $name = $_POST["name"];
    $lastname = $_POST["lastname"];
    $email = trim(mb_strtolower($_POST["email"]));
    $pass = trim($_POST["pass"]);
    $pass = password_hash($pass, PASSWORD_DEFAULT);

    $result = $mysqli->query("SELECT * FROM `user` WHERE `email`='$email'");
    //var_dump($result->num_rows);
    if ($result->num_rows != 0) {
        print("exist"); //“акой пользователь существует
    } else {
        print("ok"); //“акого пользователь не существует, можно регистрировать
        $mysqli->query("INSERT INTO `user`(`name`, `lastname`, `email`, `pass`)
VALUES ('$name', '$lastname', '$email', '$pass')");
    }
}
```

Листинг 8.

Исходные файлы проекта можно просмотреть на GitHub по ссылке <https://github.com/deniszotov1978/Diploma>.

Тестирование работоспособности веб ресурса проводилось не в автоматическом режиме, т.е. без написания тестов.

СПИСОК ЛИТЕРАТУРЫ

1. Электронные ресурсы.

1. Сайт

Документация по фреймворку Bootstrap на русском языке [Электронный ресурс]: офиц. сайт. URL: <https://bootstrap-4.ru/> (Дата обращения: 18.05.2022).

2. Сайт

Официальная документация Яндекс по подключению API [Электронный ресурс]: офиц. сайт. URL: <https://yandex.ru/dev/maps/jsapi/doc/2.1/quick-start/index.html> (Дата обращения: 18.05.2022).

3. Сайт

Электронный учебник по HTML и CSS [Электронный ресурс]: офиц. сайт. <http://htmlbook.ru/html5> (Дата обращения: 19.05.2022).

4. Сайт

Официальная документация Mozilla [Электронный ресурс]: офиц. сайт. URL: <https://developer.mozilla.org/ru/> (Дата обращения: 18.05.2022).