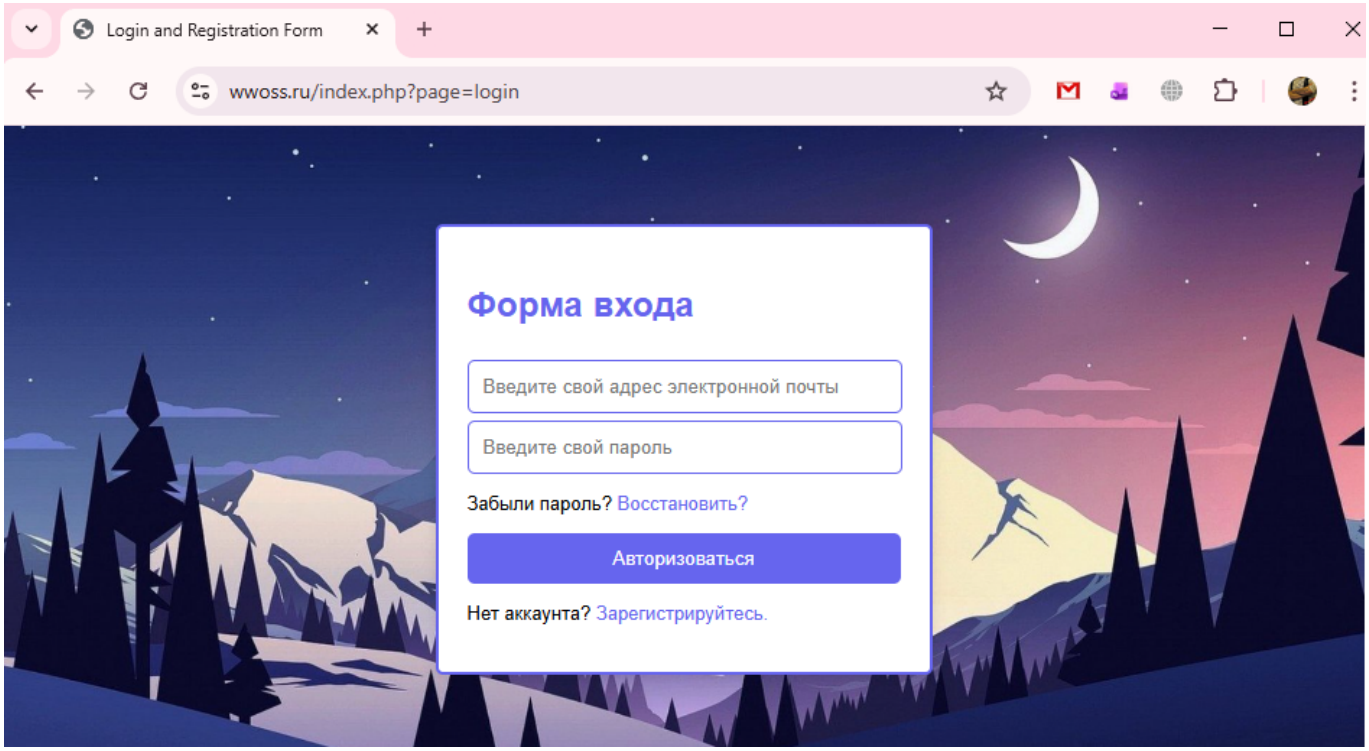
Простая адаптивная форма регистрации и авторизации

Данная форма написана на PHP и CSS. Адаптивность ей придает медиа-запрос `@media screen and (max-width: 480px)` Это правило адаптивного дизайна, которое применяет определенные стили, когда ширина окна браузера составляет 480 пикселей или меньше , а устройство — экран (например, настольный компьютер, смартфон или планшет).

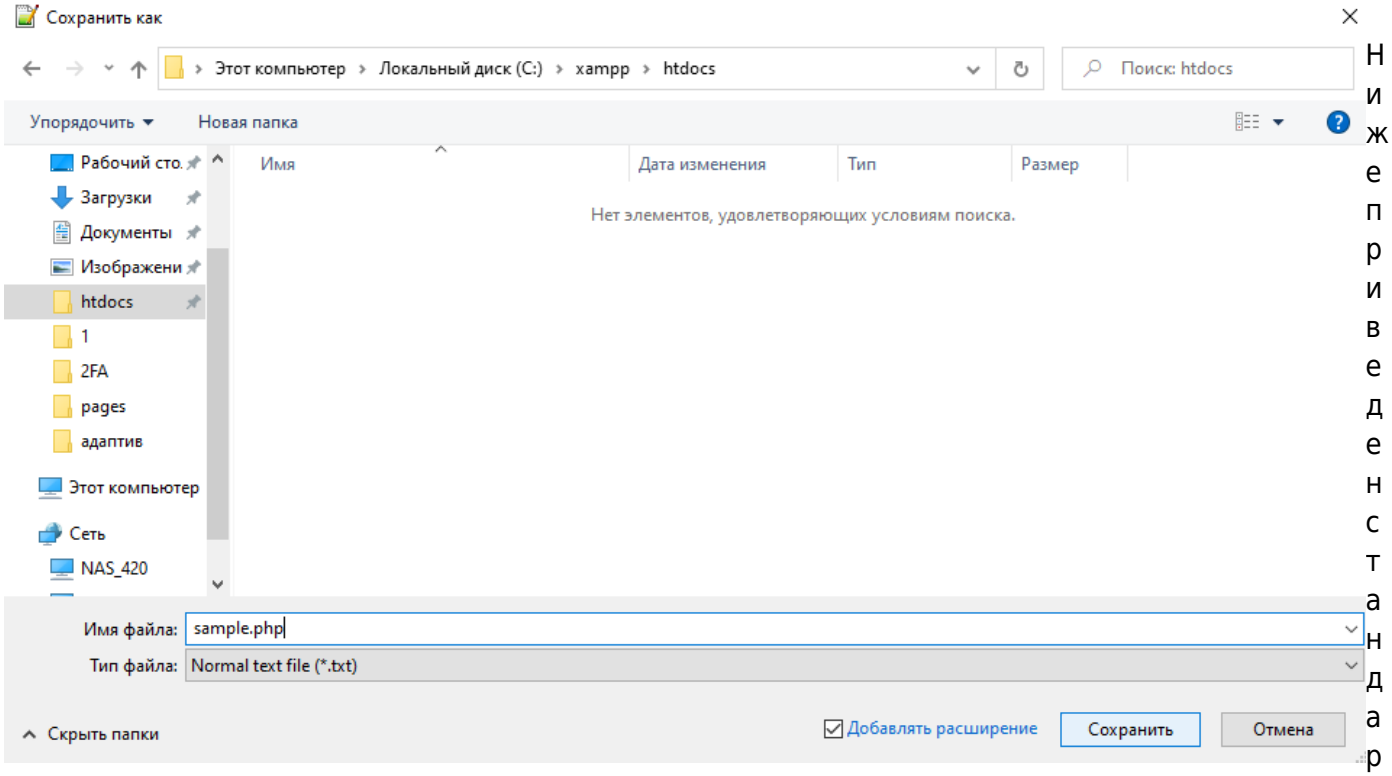
- Windows 10 Pro
- XAMPP - полностью бесплатный и простой в установке дистрибутив Apache, содержащий MariaDB, PHP и Perl. Работая в среде MS Windows, его использование регулируется GNU General Public License.
- Google Chrome — браузер, разрабатываемый на основе свободного браузера Chromium и движка Blink.
- Notepad++ - бесплатный текстовой редактор исходного кода. Работая в среде MS Windows, его использование регулируется GNU General Public License.



- Главная страница (auth)
- Форма авторизации (Login Form)
- Форма регистрации (Registration Form)
- Форма подтверждения электронной почты (Email Confirmation Form)
- Форма сброса пароля (Password Reset Form)
- Форма 2х уровневой авторизации (2-Level Authorization Form)
- Выход из системы (logout)

Sample (шаблон)

В блокноте создадим простой шаблон php и сохраним его в корне на сервере как sample.php



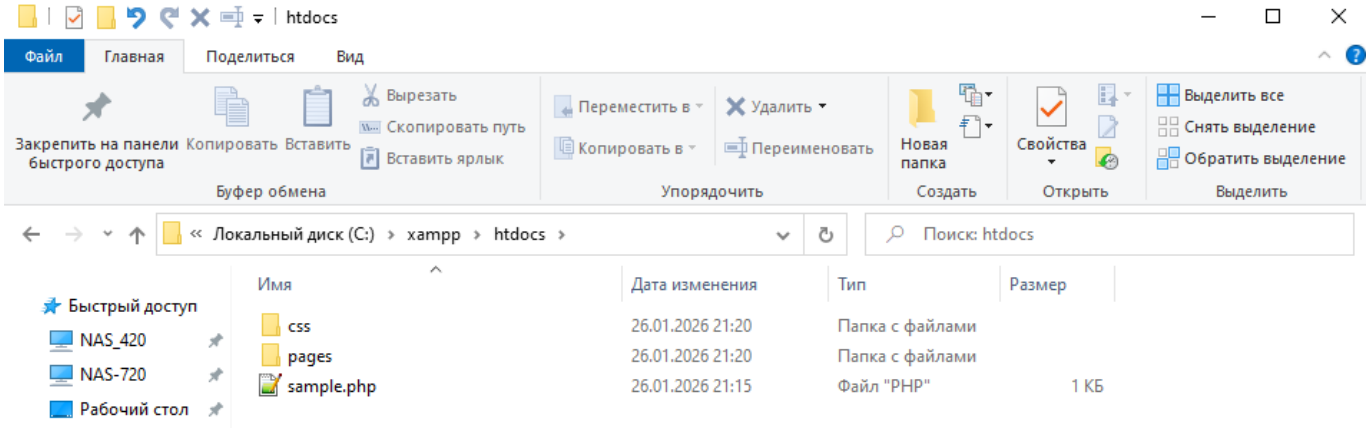
Ниже приведен стандартный код для разметки HTML5

sample.php

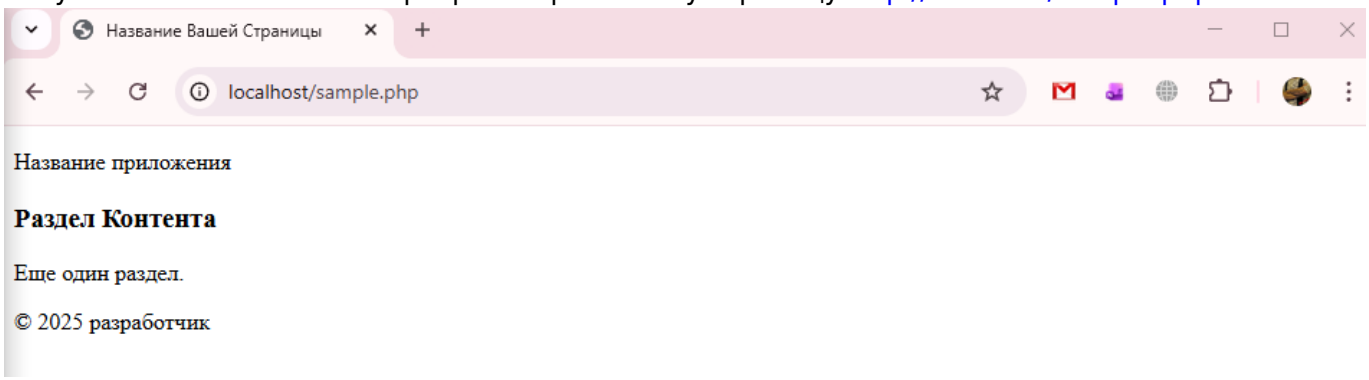
```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Название Вашей Страницы</title>
    <!-- Подключение CSS -->
    <link rel="stylesheet" href="/css/style.css">
</head>
<body>
    <header>
    <p>Название приложения</p>
    </header>
    <main>
        <article>
            <section>
                <h3>Раздел Контента</h3>
                <p>Еще один раздел.</p>
            </section>
        </article>
```

```
    </main>
    <footer>
        <p>&copy; 2025 разработчик</p>
    </footer>
</body>
</html>
```

В этой же директории каталог `css` для файлов стиля и фонового изображения, а так-же директорию `pages` для страниц



Запустим наш локальный сервер и откроем нашу страницу http://localhost/sample.php



блоки header и footer

Блоки <header></header> и <footer></footer> мы скроем просто удалив из них контент

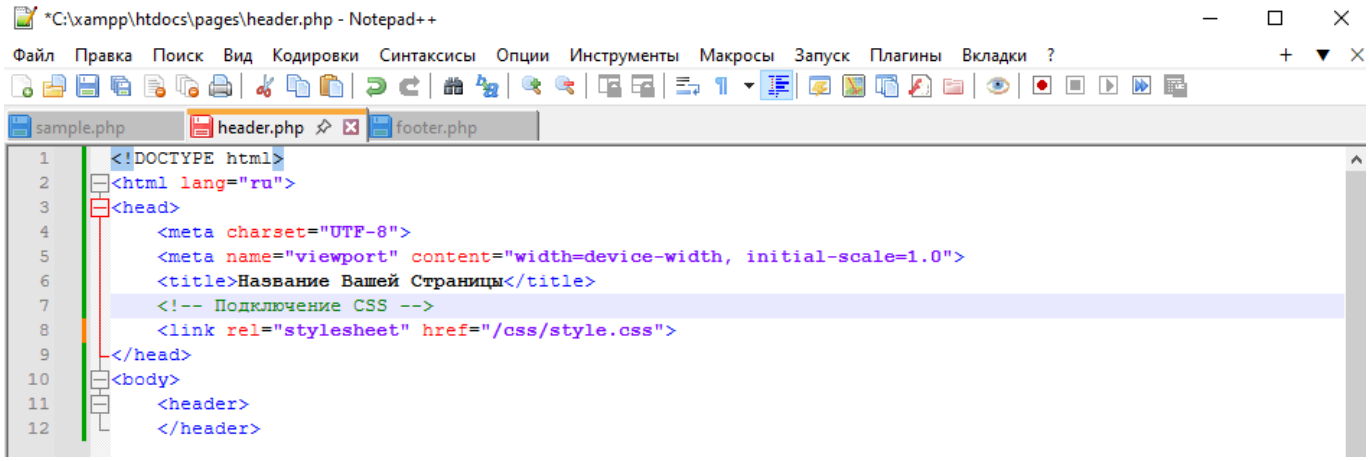
```
<p>Название приложения</p>
<p>&copy; 2025 разработчик</p>
```

т.к. это будет действитель простая форма, а они будут необходимы только если форме регистрации и авторизации будет неоходим общий стиль со страницей сайта.

header.php

В директории `pages` создадим пустой файл `header.php` и откроем его в блокноте. Удилим из

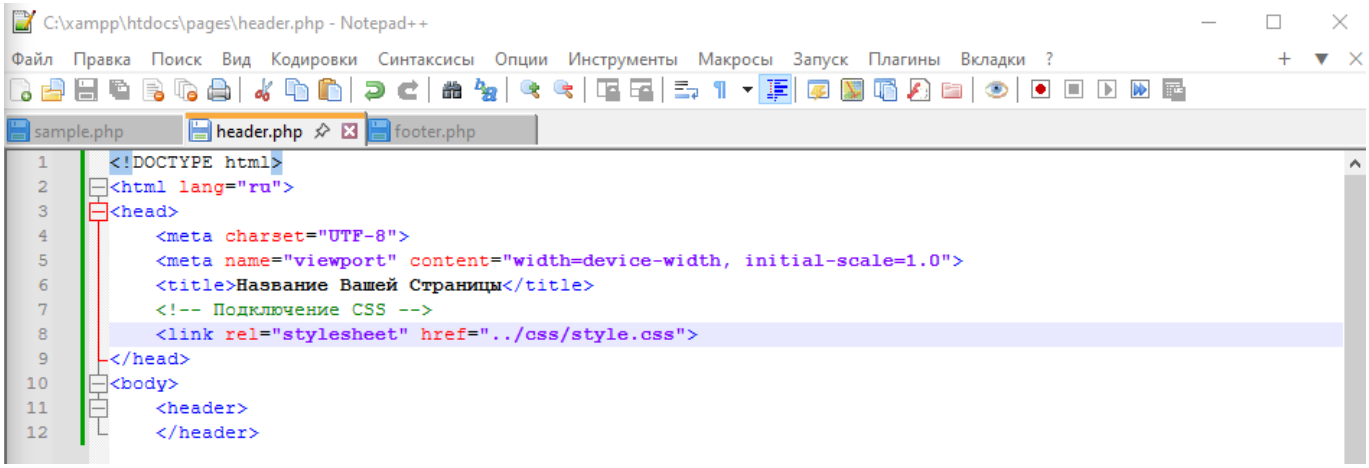
нашего файла шаблона `sample.php` код от начала %<!DOCTYPE html> и до окончания блока </header> и вставим этот код в файл `header.php`.



Так как файл `header.php` будет располагаться в директории `pages`, то в строке подключения файла внешнего стиля `style.css` добавим две точки `..` изменив тем самым путь к файлу.

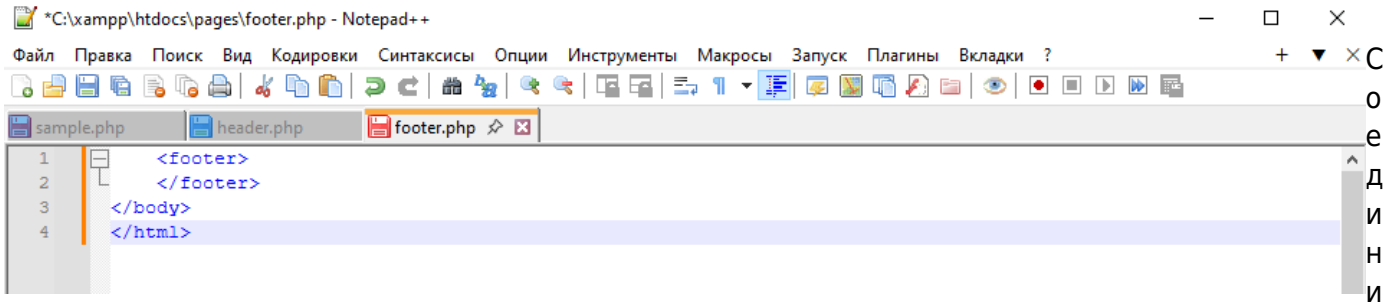
html

```
    <!-- Подключение CSS -->
    <link rel="stylesheet" href="../css/style.css">
```



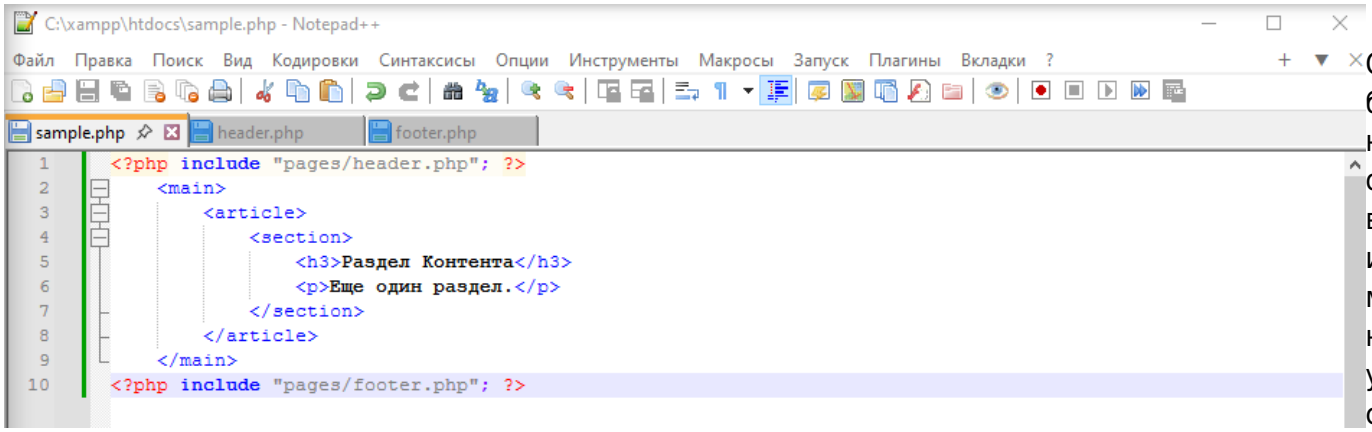
footer.php

В директории `pages` создадим пустой файл `footer.php` и откроем его в блокноте. Удилим из нашего файла шаблона `sample.php` код от начала %<footer> и до окончания блока </html> и вставим этот код в файл `footer.php`.

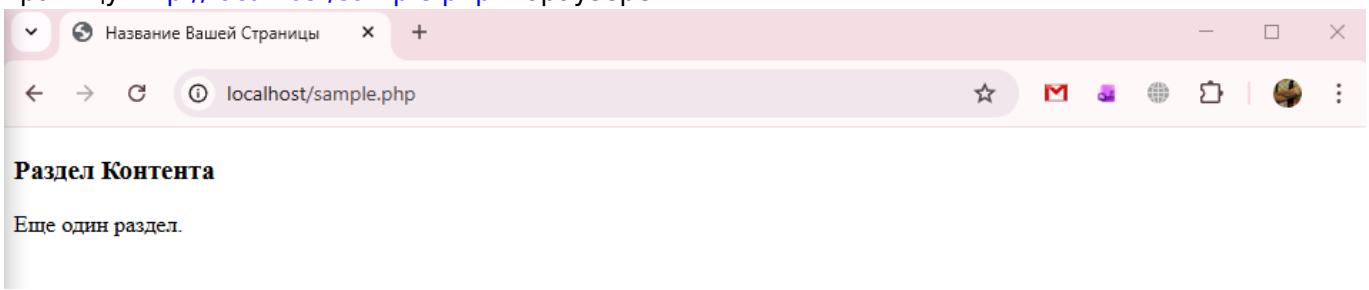


Соединим оставшуюся часть кода нашего шаблона `sample.php` с помощью инструкция `include` для подключения внешних файлов с header.php и footer.php

```
<?php include "pages/header.php"; ?>
и
<?php include "pages/footer.php"; ?>
```



Обновим страницу http://localhost/sample.php в браузере



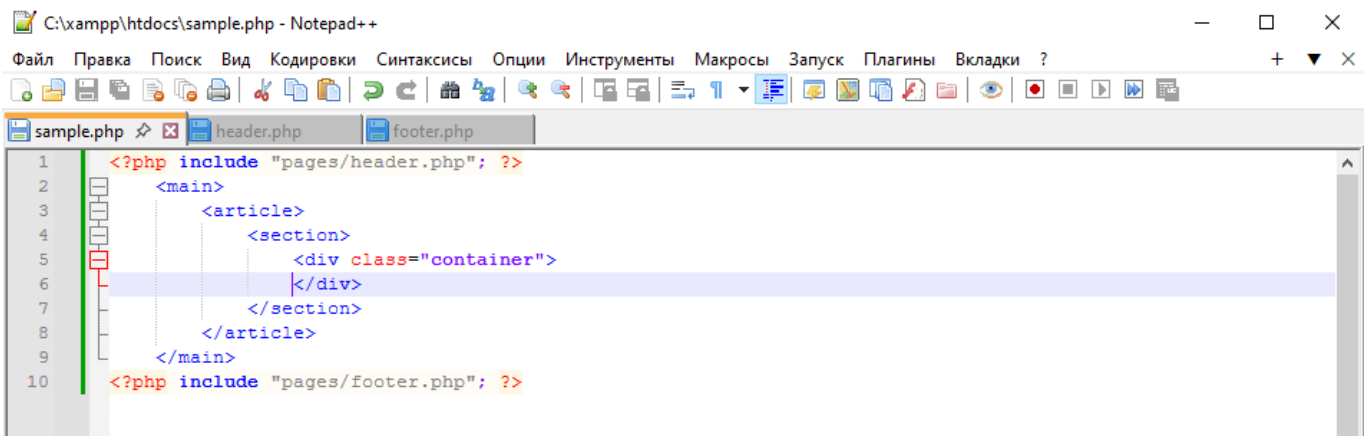
container

Класс HTML `container` это главное окно нашей формы в блоке `<section>`

HTML

```
<div class="container">
</div>
```

Вставим его вместо нашей заглушки (заголовка и параграфа)



style.css

В директории css создадим пустой файл style.css и откроем его в блокноте, задав стили для нашего контейнера. Свойства стиля указаны в комментариях кода:

style.css

```
/*Стиль контейнера*/
.container {
    background: white; /*цвет контейнера*/
    padding: 20px; /* Отступы по  20px  со всех сторон */
    border-radius: 5px; /*толщина рамки*/
    border: 2px solid #6665ee; /*цвет и углы рамки*/
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /*добавляет тень к элементу*/
    width: 300px; /*ширина контейнера*/
}
```

Обновим страницу браузера и увидим наш контейнет.



перейдем к фону страницы и стилю элемента body. В данном примере мы используем изображение для фона нашей формы. Его размер может быть любым (в нашем случае это 1918x1224px).

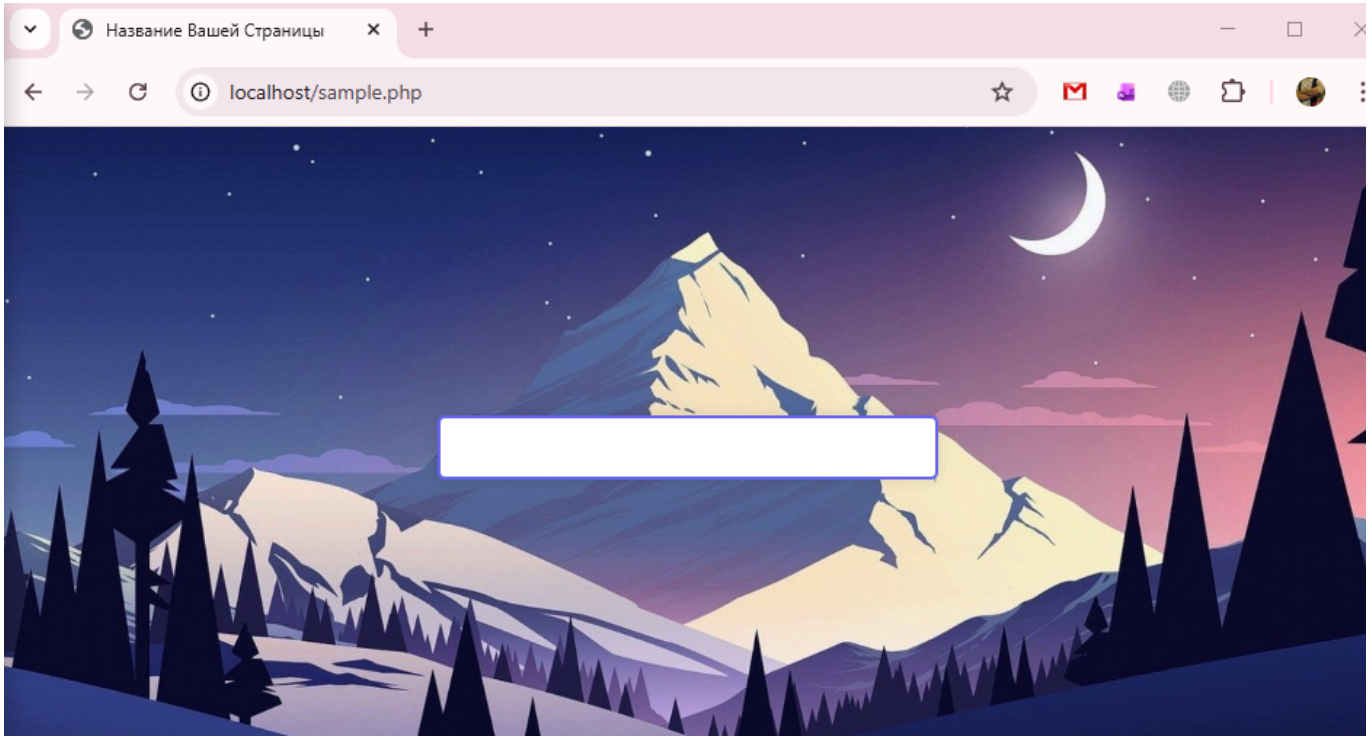


Сохраним это изображение в директории `css` нашего проекта и перейдем к стилю элемента `body`, прописав нижеприведенный код в начале файла `style.css`

style.css

```
/*Стиль контейнера*/
body {
    font-family: Arial, sans-serif;
    /*Управление поведением фоновых изображений*/
    background-image: url("fone_forms.jpg");
    background-size: cover; /* Растягивает и обрезает */
    background-position: center; /* Центрирует */
    background-repeat: no-repeat; /* Мы не повторяемся */
    display: flex; /* Коробка становится гибкой */
    align-items: center; /* Вертикальное выравнивание */
    justify-content: center; /* Горизонтальное выравнивание */
    max-width: 100%;/* Растянуть по ширине контейнера */
    height: 100vh; /* Высота всего экрана */
    object-fit: cover; /* Обрезка с заполнением */
    object-position: center; /* Центрирование */
}
```

и снова обновил страницу браузера



Чтобы все у нас было в едином внешнем виде, добавим правила для стилей ссылок, параграфов и заголовков

style.css

```
/*Стиль ссылок*/
a {
    text-decoration: none;/*Ссылки без подчеркивания*/
    color: #6665ee;   /* цвет текста для всей страницы */
}

/*Стиль заголовка*/
H2 {
    color: #6665ee;   /* цвет текста заголовка */
}

/*Стиль параграфа*/
p {
    font-family: Arial, sans-serif;      /* цвет текста заголовка */
    font-size: small;    /*задает размер шрифта*/
}
```

Заголовки, ссылки, параграфы и формы

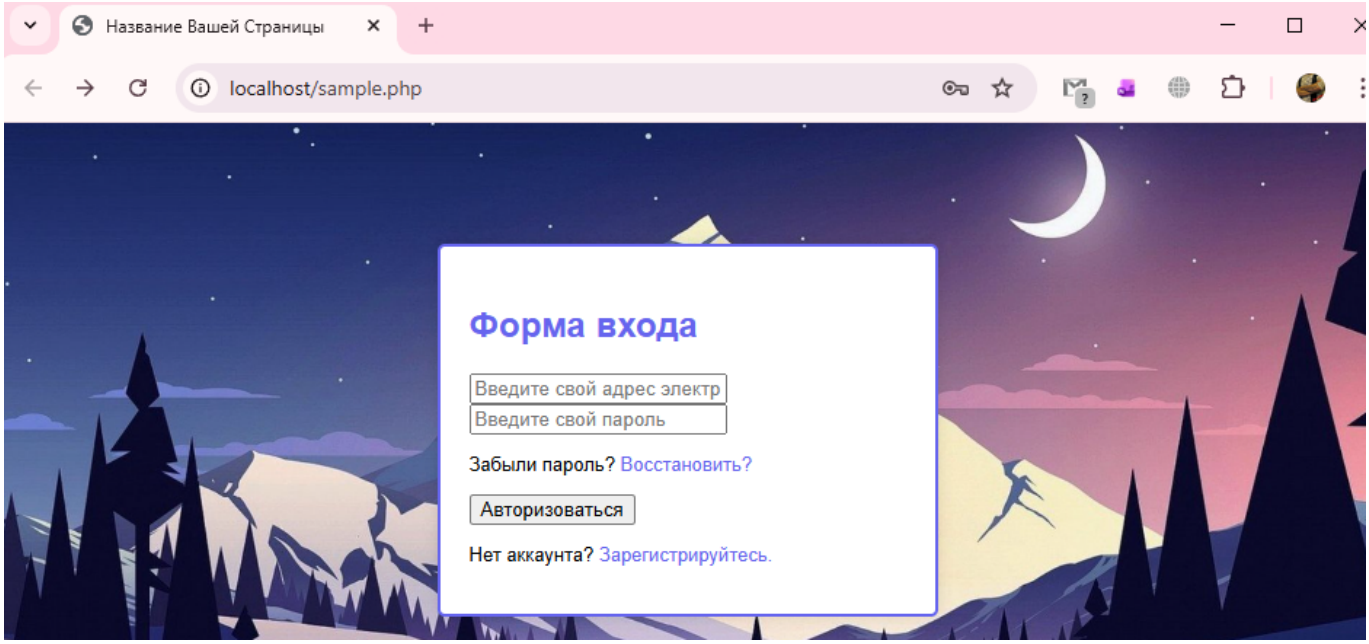
Вставим в наш `container` код формы с элементами заголовка, ввода, параграфа и кнопки

html

```
<form method="post" action="#">
    <h2>Форма входа</h2>
    <input type="email" id="email" required placeholder="Введите свой адрес электронной почты">
    <input type="password" id="password" required placeholder="Введите свой пароль">
    <p>Забыли пароль? <a href="index.php?page=reset">Восстановить?</a></p>
    <button type="submit">Авторизоваться</button>
    <p>Нет аккаунта? <a href="index.php?page=register">Зарегистрируйтесь.</a></p>
</form>
```

- method="post" - отправка данных формы на сервер.
- action="#" - значение параметра означает, что данные формы отправляются на ту же страницу

Приведенный выше код - это просто заглушка, которая показывает нам пример приведения стиля заголовка, параграфа и ссылки (отсутствие подчеркивания). Позже мы поменяем адреса этой заглушки



Формы и кнопки

Создадим правила для стилей кнопки и форм ввода. для этого пропишем в файле стилей `style.css` следующий код

style.css

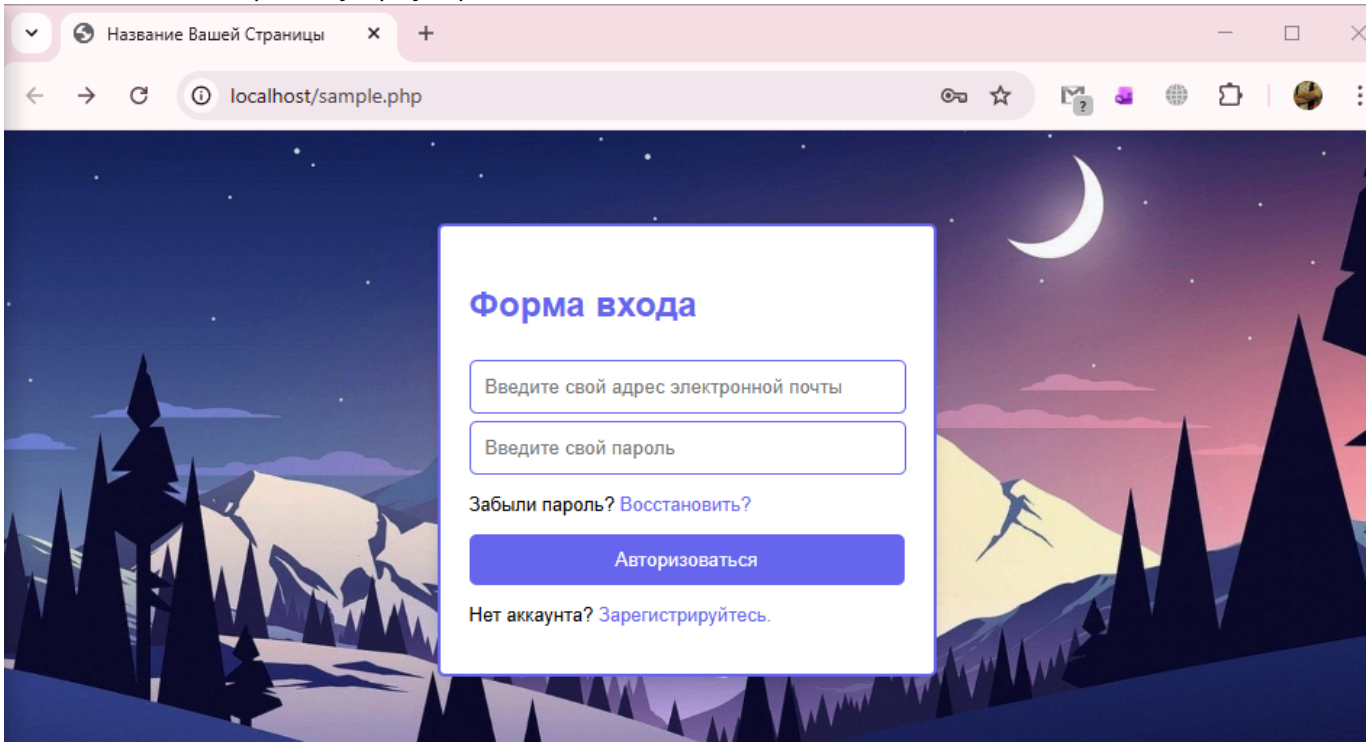
```
/*Стиль поля ввода*/
input {
    width: 93%;
```

```
    padding: 10px;
    margin: 5px 0 0 0; /*Отступ только сверху*/
    border: 1px solid #6665ee;
    border-radius: 5px;
}

/*Стиль кнопки*/
button {
    width: 100%;
    padding: 10px;
    background-color: #6665ee;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background-color: #5757d1;/*Изменение цвета кнопки при наведении курсора*/
}
```

Снова обновим страницу браузера



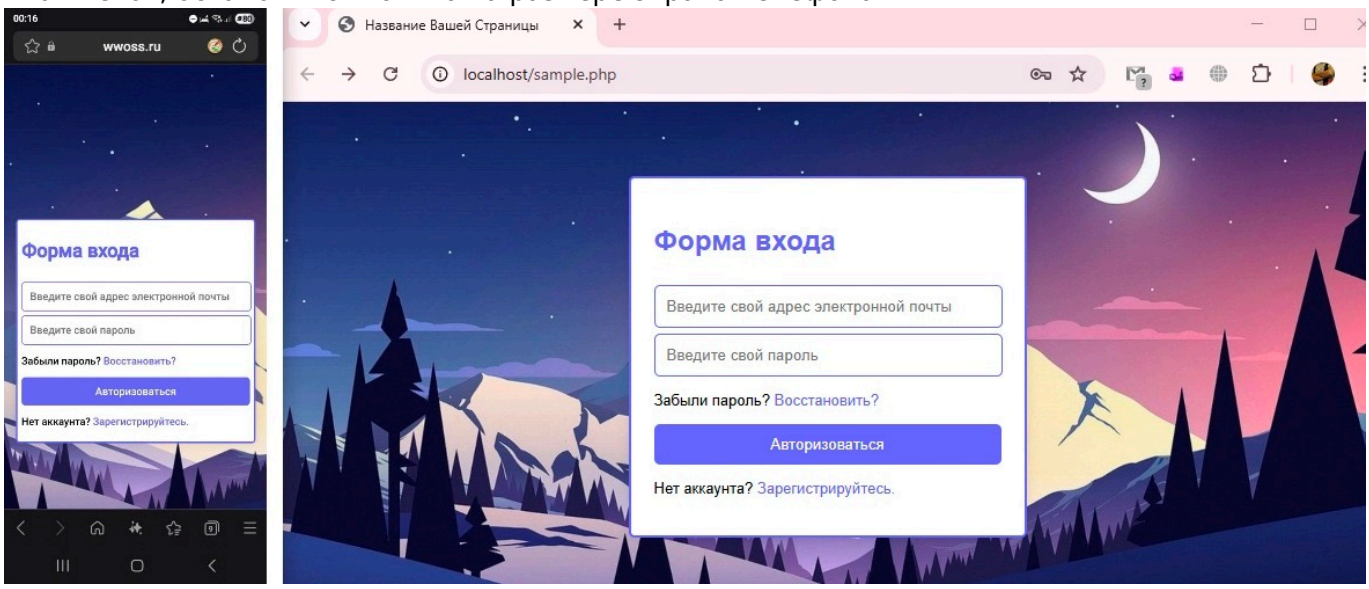
И последнее, что мы пропишем в файле `style.css`, это медиа-запрос, используемый для адаптивного веб-дизайна.

style.css

```
@media screen and (max-width: 480px) {
  /* Стили для планшетов и телефонов */
```

```
    .container { width: 90%; padding: 5px; }
}
```

Так как форма у нас имеет ширину всего 300px, то нет смысла писать медиа запрос для планшетов, остановимся только на размере экрана телефона.



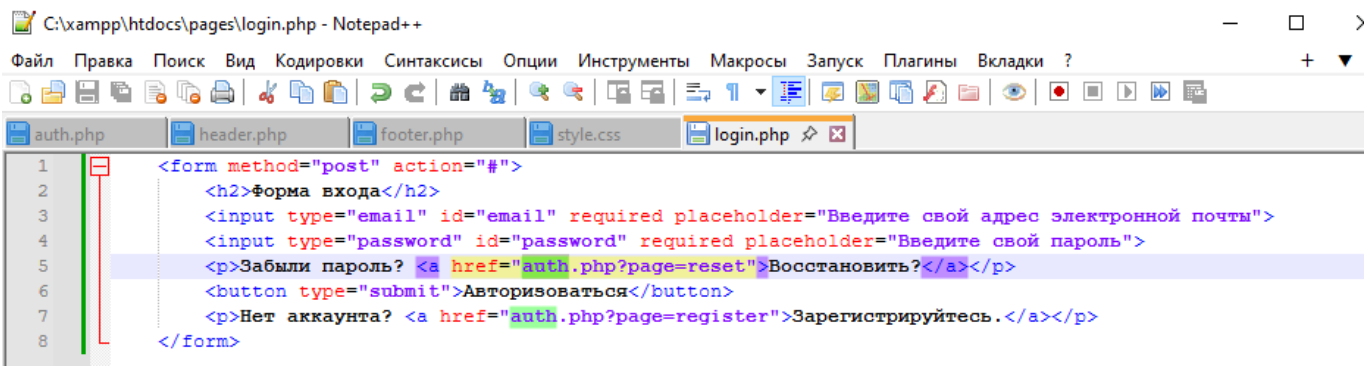
Динамическое подключение страниц

Итак, пришло время создание динамически подключаемых страниц.

- auth.php - Главная страница (auth)
- login.php - Форма авторизации (Login Form)
- register.php - Форма регистрации (Registration Form)
- confirm.php - Форма подтверждения электронной почты (Email Confirmation Form)
- reset.php - Форма сброса пароля (Password Reset Form)
- 2fa.php - Форма 2х уровневой авторизации (2-Level Authorization Form)
- logout.php - Выход из системы (logout)

login.php

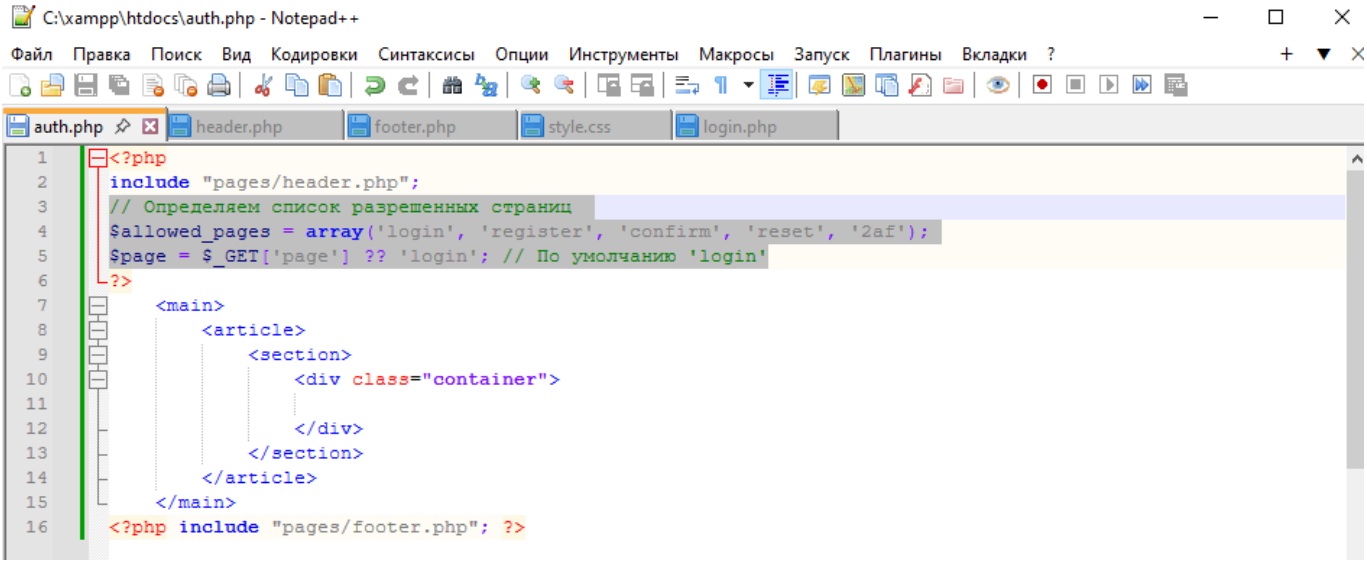
Переименуем, находящийся в корне директории наш шаблон `sample.php` в `auth.php`, а так же создадим файл `login.php` в директории `pages`.
Из файла `auth.php` удалим весь код из и запишем это в файл `login.php`, при этом изменив адреса ссылок на `auth.php`



Мы вынесли форму авторизации во внешний файл, подключим его. Для этого в файле `auth.php` пропишем код

auth.php

```
// Определяем список разрешенных страниц
$allowed_pages = array('login', 'register', 'confirm', 'reset', '2fa');
$page = $_GET['page'] ?? 'login'; // По умолчанию 'login'
```

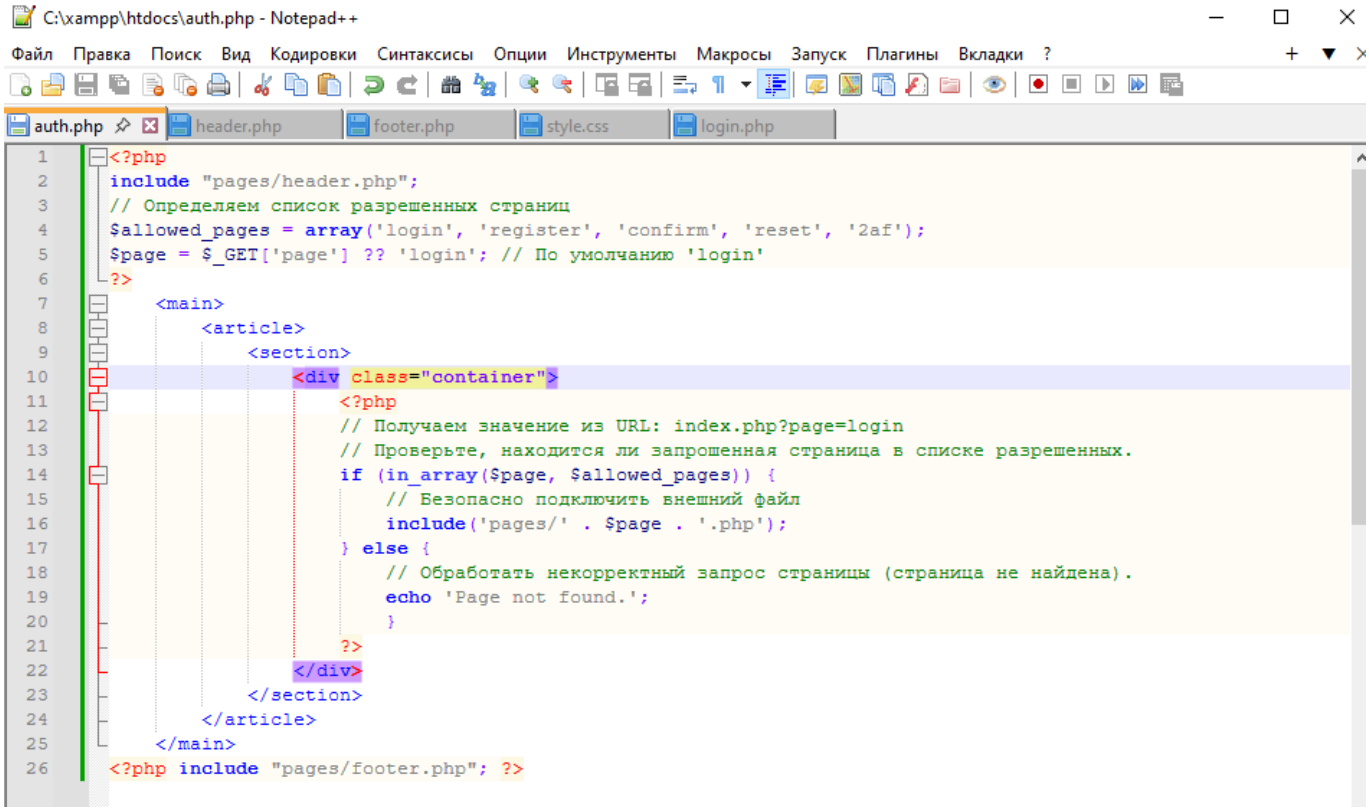


В наш блок контейнера вставляем следующий код

auth.php

```
<?php
// Получаем значение из URL: auth.php?page=login
// Проверьте, находится ли запрошенная страница в списке разрешенных.
if (in_array($page, $allowed_pages)) {
    // Безопасно подключить внешний файл
    include('pages/' . $page . '.php');
} else {
    // Обработать некорректный запрос страницы (страница не найдена).
    echo 'Page not found.';
}
```

```
?>
```



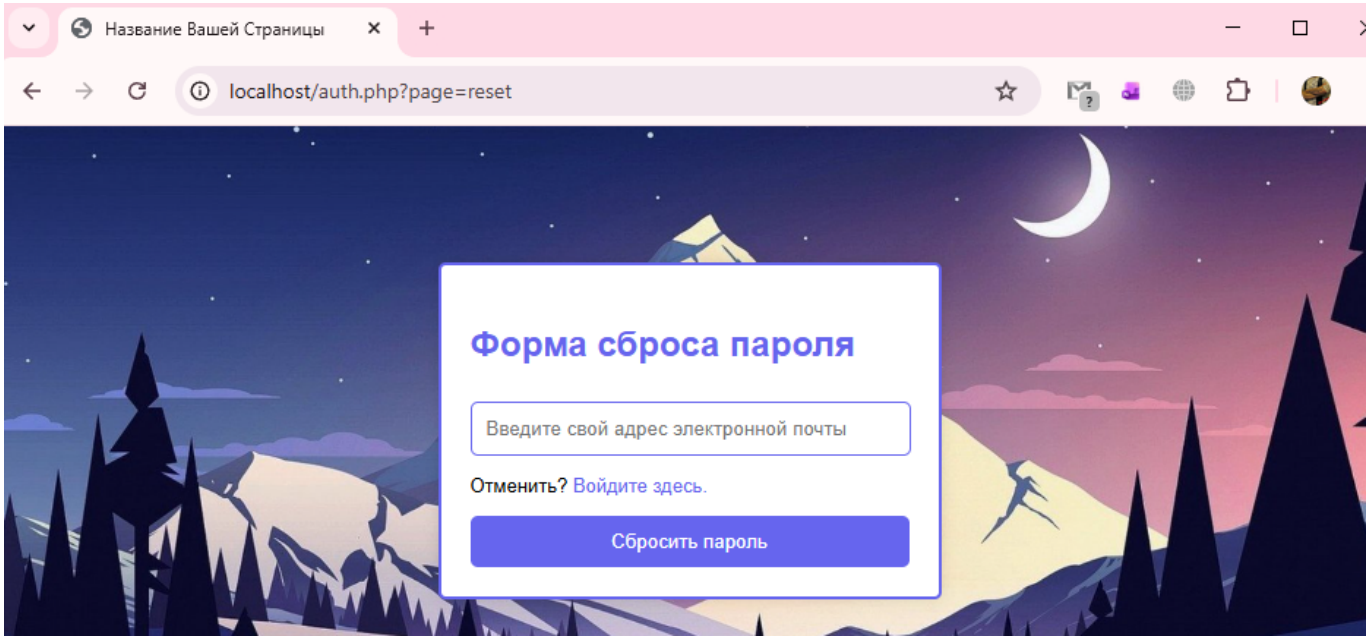
reset.php

Создадим файл `reset.php` в директории `pages` и запишем в него следующий код

reset.php

```
<form method="post" action="#">
    <h2>Форма сброса пароля</h2>
    <input type="email" id="email" required placeholder="Введите свой адрес электронной почты">
    <p>Отменить? <a href="auth.php?page=login">Войдите здесь.</a></p>
    <button type="submit">Сбросить пароль</button>
</form>
```

Обновим страница браузера и нажмем на ссылку «Восстановить?»



Ссылка «Отменить? Войдите здесь.» вернет на форму «login»

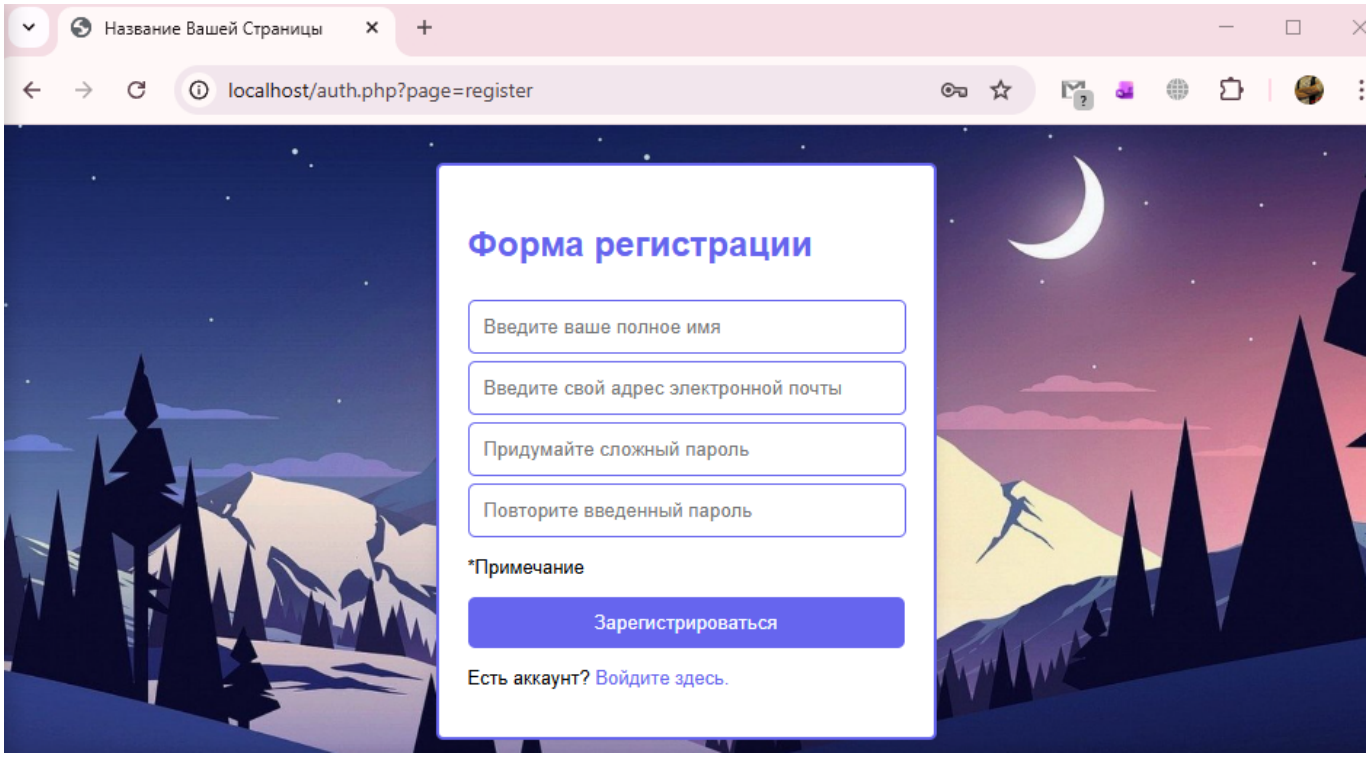
register.php

Создадим файл формы регистрации `register.php` в директории `pages` и запишем в него следующий код

register.php

```
<form method="post" action="#">
    <h2>Форма регистрации</h2>
    <input type="text" id="user_login" name="user_login" required
placeholder="Введите ваше полное имя">
    <input type="email" id="user_email" name="user_email" required
placeholder="Введите свой адрес электронной почты">
    <input type="password" id="user_pass" name="user_pass" required
placeholder="Придумайте сложный пароль">
    <input type="password" id="confirmPassword" name="confirmPassword"
required placeholder="Повторите введенный пароль">
    <p title="Пароль должен содержать не менее 8 символов и содержать не менее:
одной строчной буквы, одной заглавной буквы, одной цифры и одного
символа.">*Примечание</p>
    <button type="submit">Зарегистрироваться</button>
    <p>Есть аккаунт? <a href="auth.php?page=login" >Войдите здесь.</a></p>
</form>
```

Обновим страница браузера и нажмем на ссылку «Зарегистрируйтесь.»



Ссылка «Есть аккаунт? Войдите здесь.» вернет на форму «login»

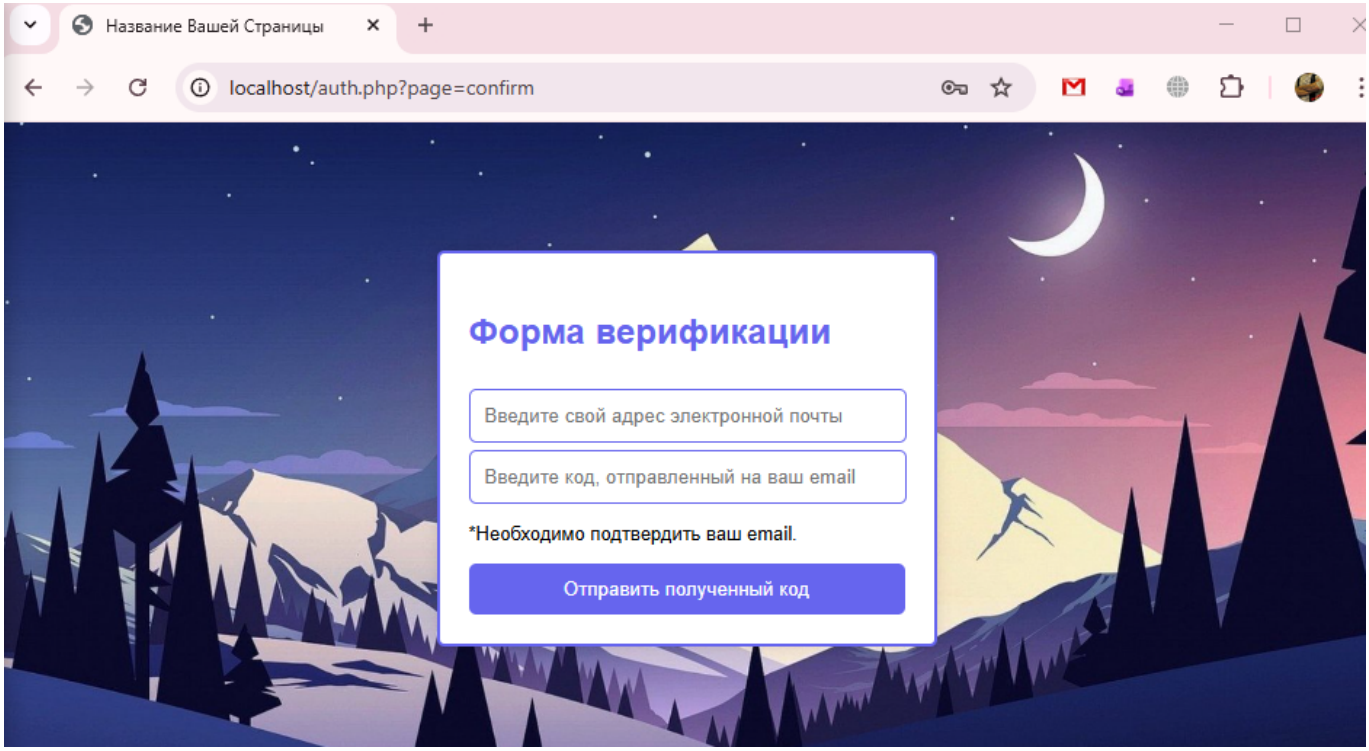
confirm.php

Создадим файл формы верификации `confirm.php` в директории `pages` и запишем в него следующий код

confirm.php

```
<form method="post" action="#">
    <h2>Форма верификации</h2>
    <input type="email" id="user_email" name="user_email" required
placeholder="Введите свой адрес электронной почты">
    <input type="password" id="user_activation_email_key"
name="user_activation_email_key" required placeholder="Введите код,
отправленный на ваш email">
    <p>*Необходимо подтвердить ваш email.</p>
    <button type="submit">Отправить полученный код</button>
</form>
```

На эту форму не кнопки (ссылки) перехода, она откроется после формы регистрации. Так же эта форма будет открываться, если пользователь выбирет 2х факторную защиту аккаунта по электронной почте.



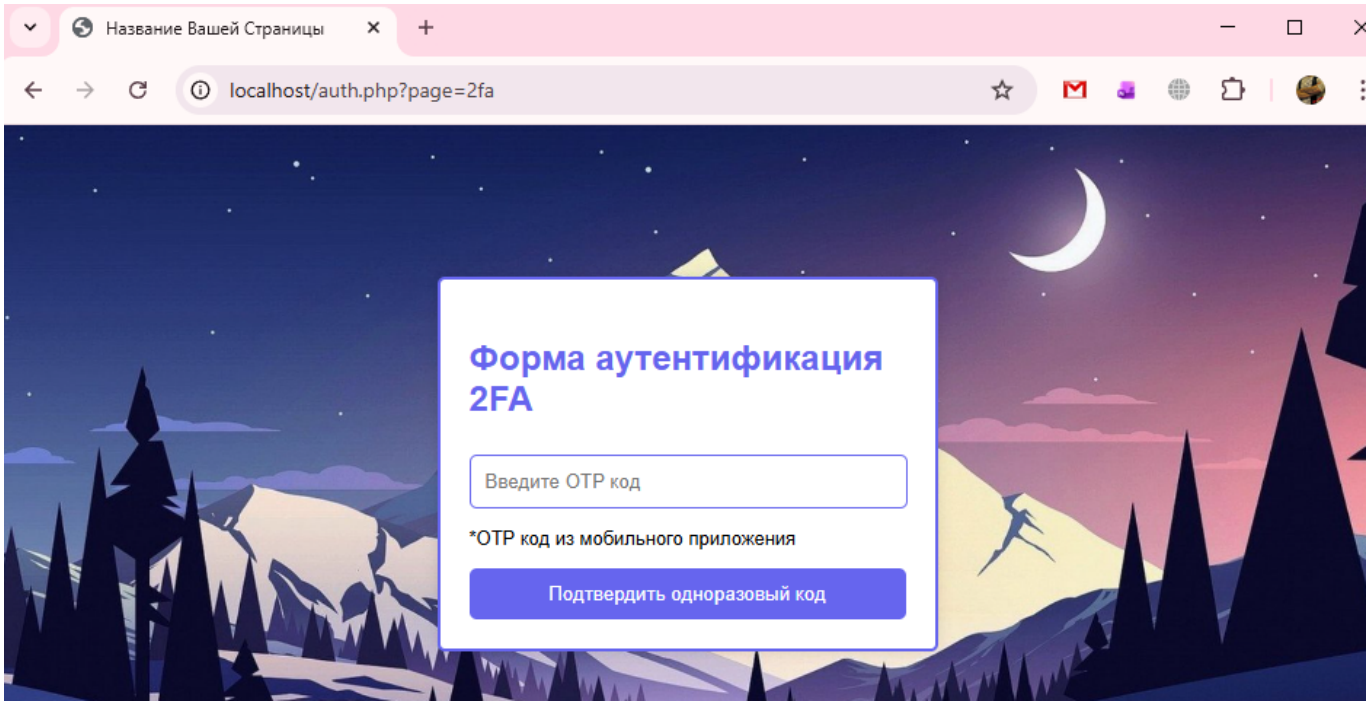
2fa.php

Создадим файл формы 2х факторной аутентификации через мобильное приложение `2fa.php` в директории `pages` и запишем в него следующий код

index.php

```
<form method="post" action="#">
    <h2>Форма аутентификация 2FA</h2>
    <input type="number" id="user_activation_2fa"
name="user_activation_2fa" pattern="\d{6}" required placeholder="Введите
OTP код">
    <p>*OTP код из мобильного приложения </p>
    <button type="submit">Подтвердить одноразовый код</button>
</form>
```

На эту форму нет кнопки (ссылки) перехода, она откроется после формы аутентификации, если пользователь выбирет 2х факторную защиту аккаунта в мобильном приложении.



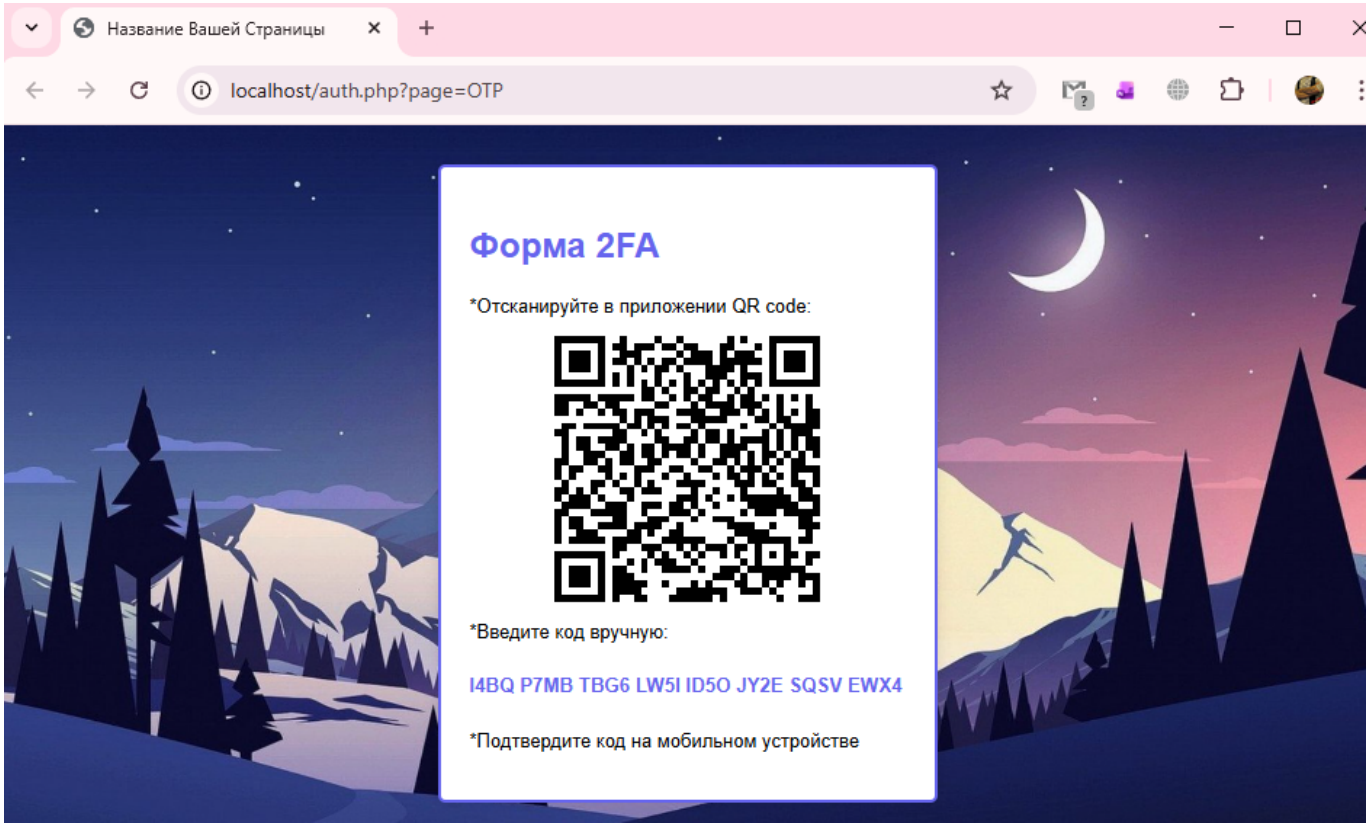
OTP.php

Создадим файл формы получения кода 2х факторной аутентификации через мобильное приложение `OTP.php` в директории `pages` и запишем в него следующий код-заглушку

OTP.php

```
<form method="post" action="#">
        <h2>Форма 2FA</h2>
        <p>*Отсканируйте в приложении QR code: </p>
    <!-- echo $tfa->getQRCodeImageAsDataUri('Demo', $secret);-->
    <img src="css/qr_code.png" />
    <p>*Введите код вручную: </p>
    <h5>I4BQ P7MB TBG6 LW5I ID5O JY2E SQSV EWX4</h5>
    <p>*Подтвердите код на мобильном устройстве </p>
</form>
```

Эта форма единожды будет доступна при включении пользователем двухфакторной авторизации после регистрации.



logout.php

Последний файл `logout.php` не доступен для просмотра и будет служить для кода выхода из системы.

Дополнения и Файлы

- См.так же
- Блочная адаптивная форма регистрации и авторизации
- Блочная полная адаптивная форма регистрации и авторизации
    - Скачать архив с примерами

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=software:development:demo:adaptive_registration_form_sample

Last update: **2026/03/16 18:47**