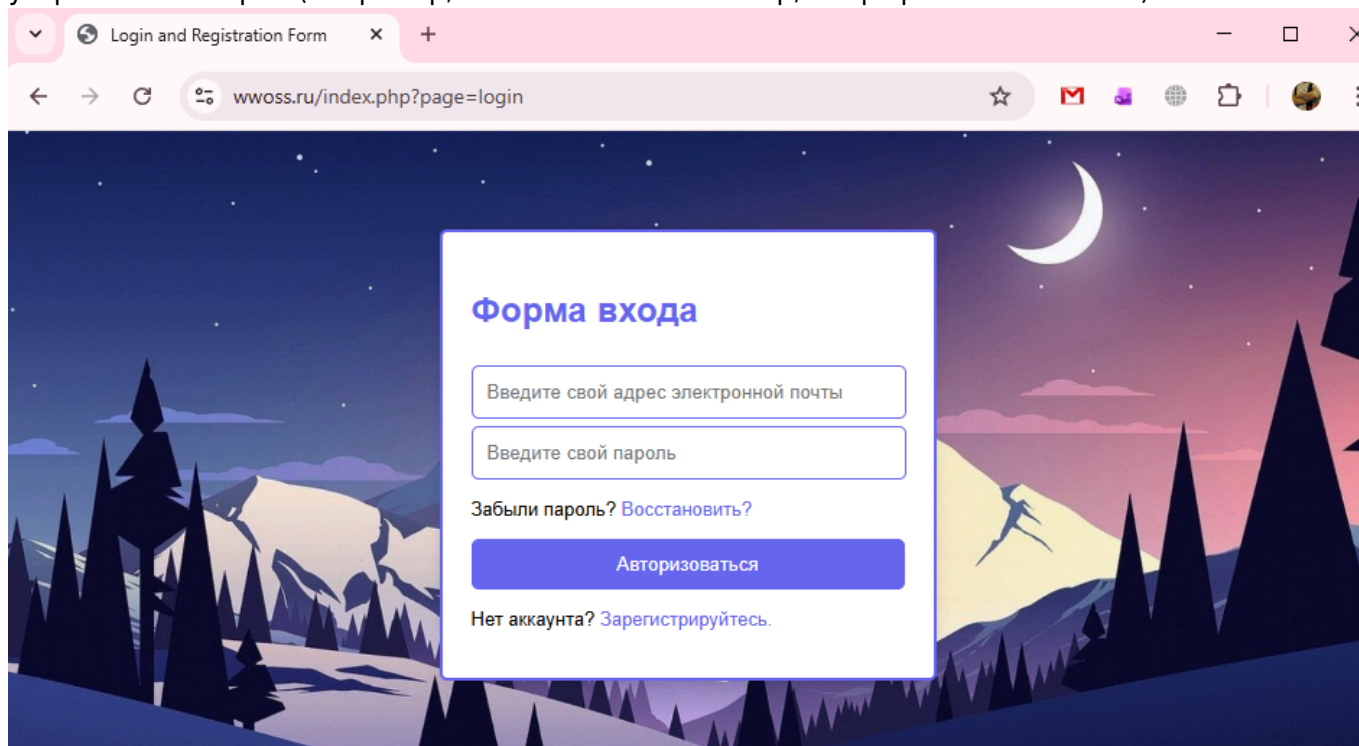


Простая адаптивная форма регистрации и авторизации

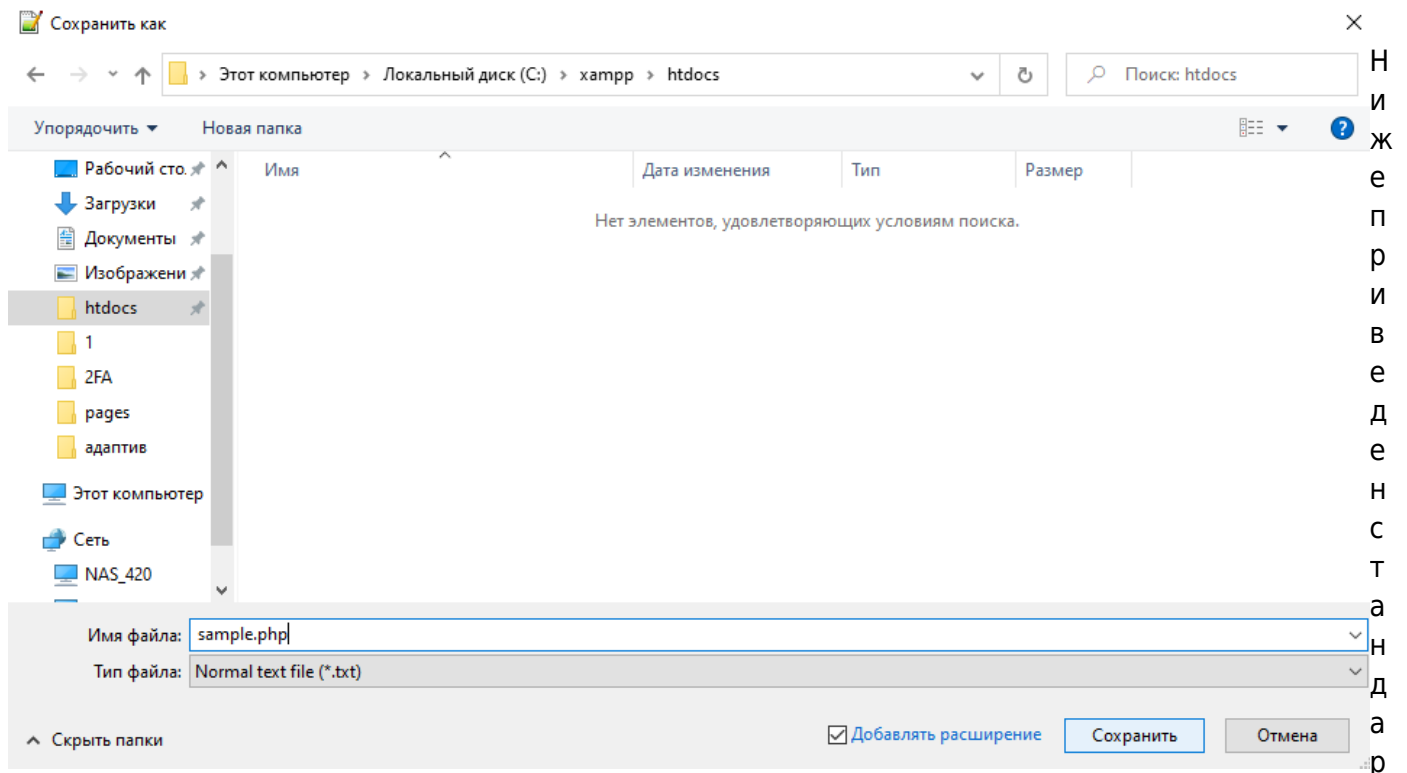
Данная форма написана на PHP и CSS. Адаптивность ей придает медиа-запрос `@media screen and (max-width: 480px)`. Это правило адаптивного дизайна, которое применяет определенные стили, когда ширина окна браузера составляет 480 пикселей или меньше, а устройство — экран (например, настольный компьютер,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



- 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



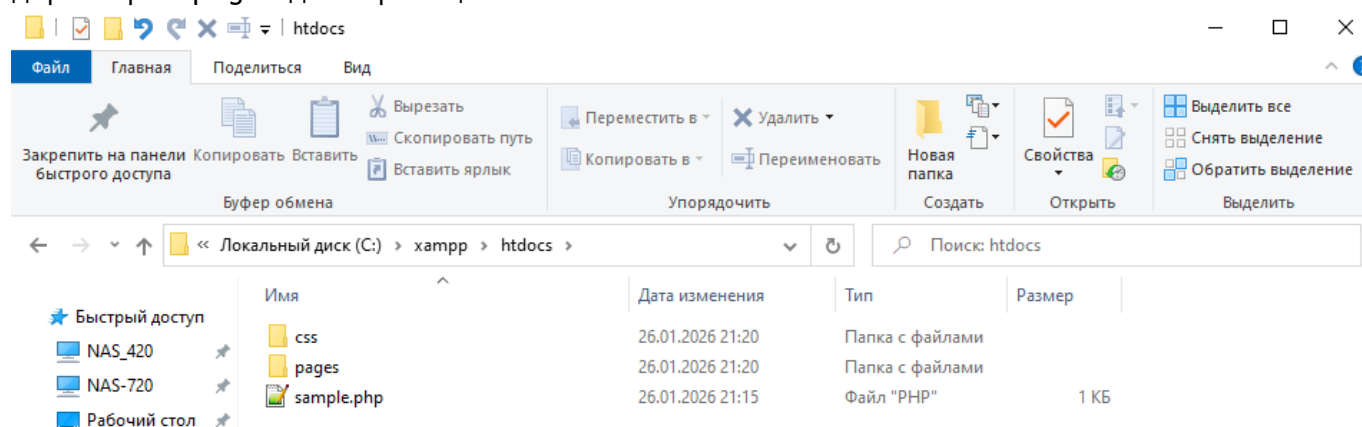
тнй код для разметки HTML5

[sample.php](#)

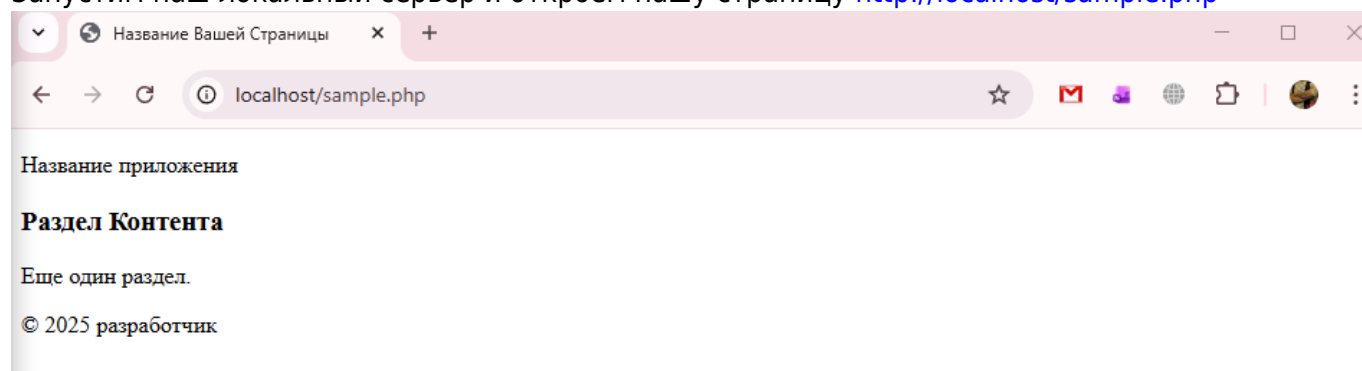
```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Название Вашей Страницы</title>
  <!-- Подключение CSS -->
  <link rel="stylesheet" href="/css/style.css">
</head>
<body>
  <header>
    <p>Название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

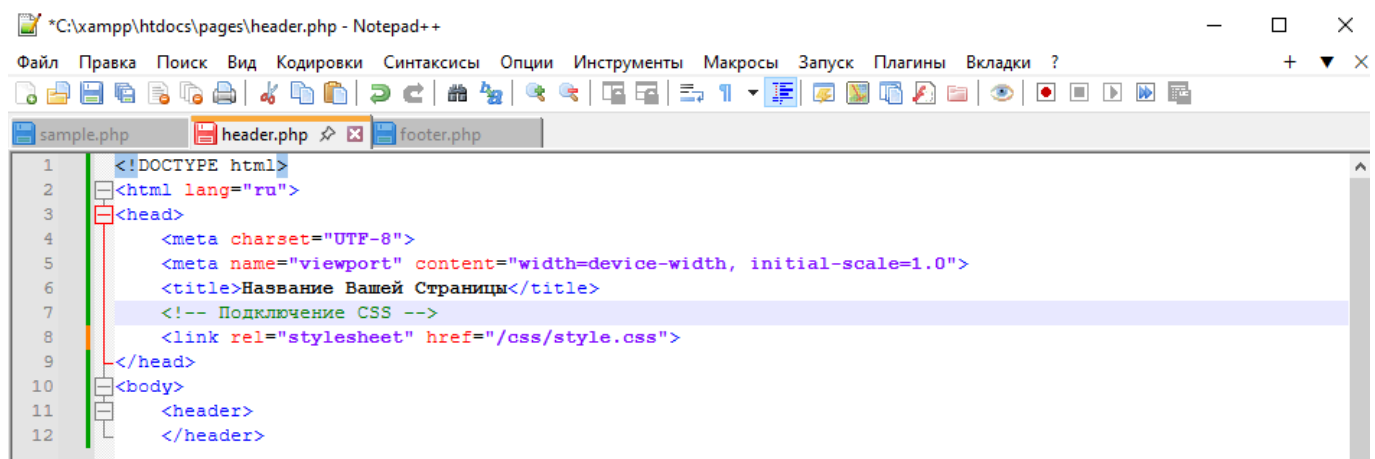
Блоки `<header></header>` и `<footer></footer>` мы скроем просто удалив из них контент

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

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

header.php

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

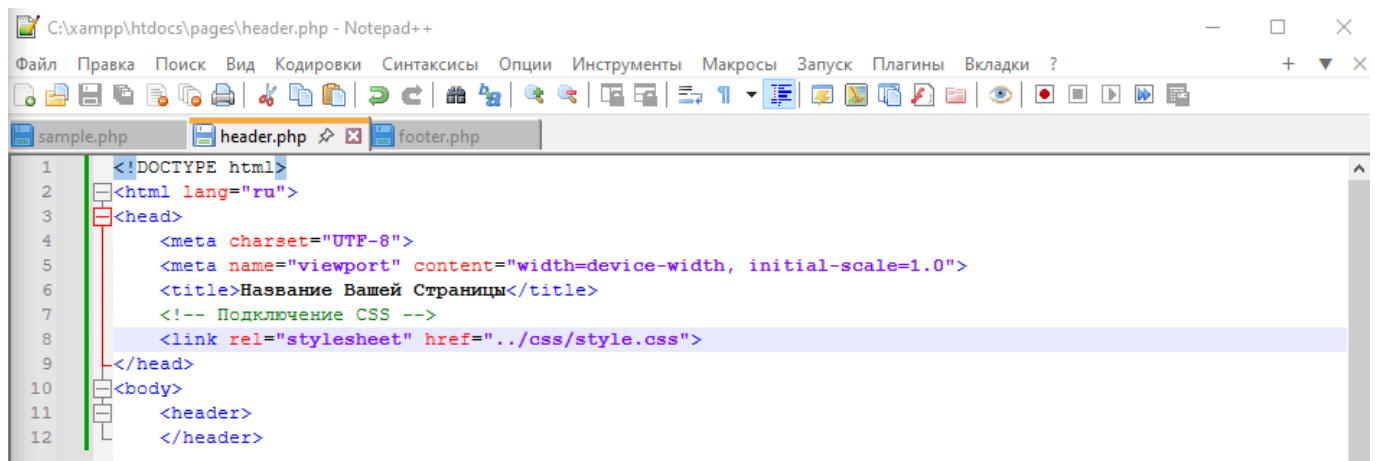


```
*C:\xampp\htdocs\pages\header.php - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
sample.php header.php footer.php
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Название Вашей Страницы</title>
7     <!-- Подключение CSS -->
8     <link rel="stylesheet" href="/css/style.css">
9 </head>
10 <body>
11     <header>
12     </header>
```

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

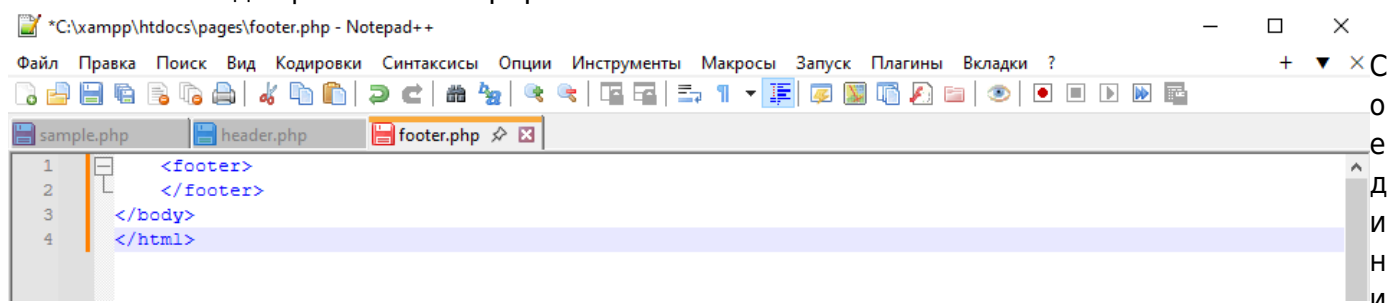
```
<!-- Подключение CSS -->
<link rel="stylesheet" href="../../css/style.css">
```



```
C:\xampp\htdocs\pages\header.php - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
sample.php header.php footer.php
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Название Вашей Страницы</title>
7     <!-- Подключение CSS -->
8     <link rel="stylesheet" href="../../css/style.css">
9 </head>
10 <body>
11     <header>
12     </header>
```

footer.php

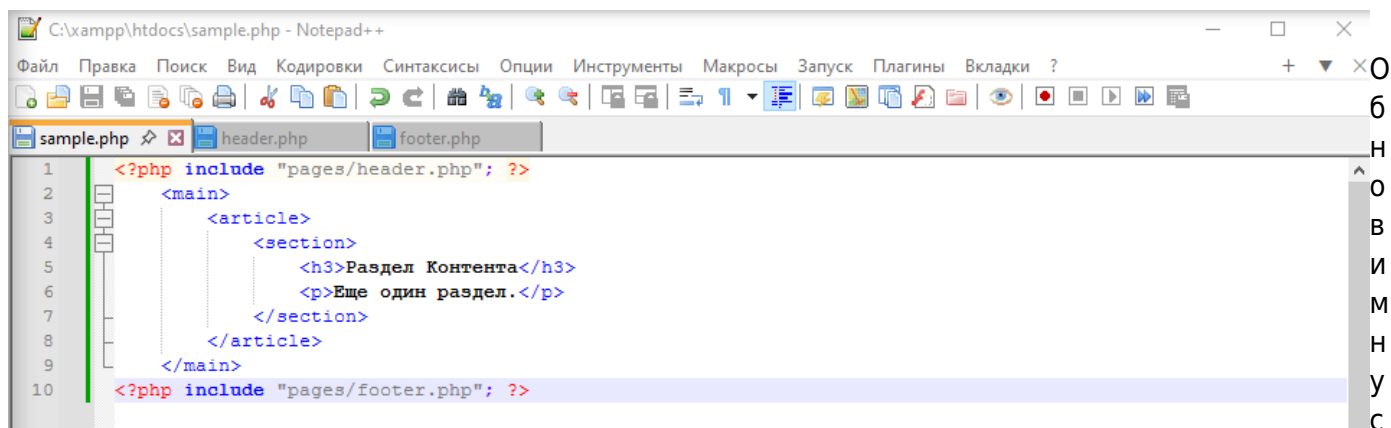
В директории `pages` создадим пустой файл `footer.php` и откроем его в блокноте. Удлим из нашего файла шаблона `sample.php` код от начала `<footer>` и до окончания блока `</html>` и вставим этот код в файл `footer.php`.



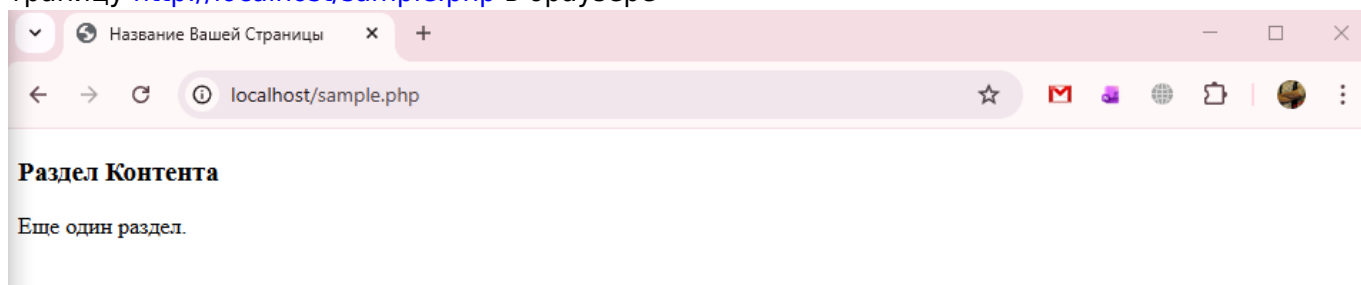
```
*C:\xampp\htdocs\pages\footer.php - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
sample.php header.php footer.php
1 <footer>
2 </footer>
3 </body>
4 </html>
```

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

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



страницу <http://localhost/sample.php> в браузере



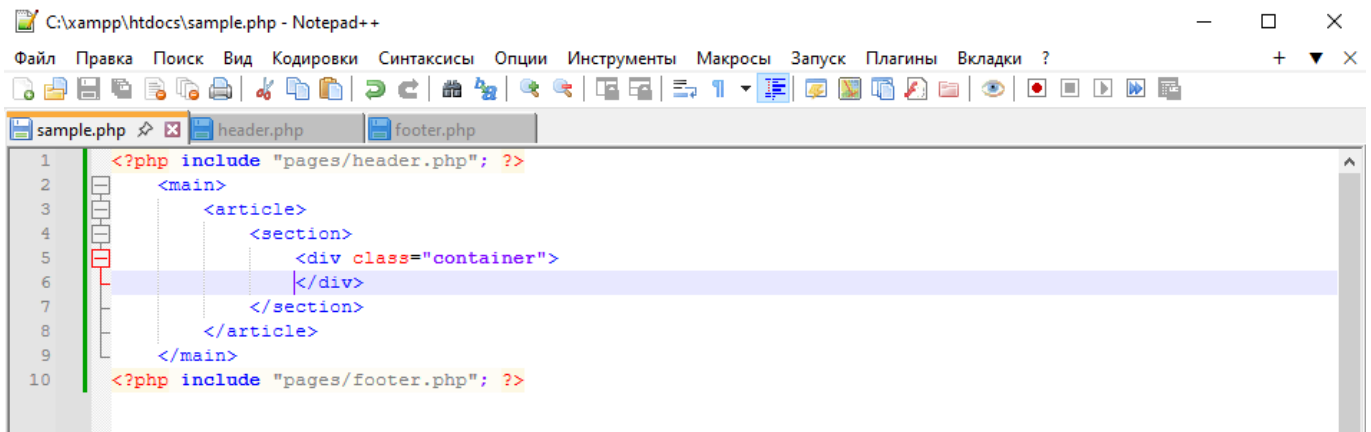
container

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 контейнер.



дем 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).

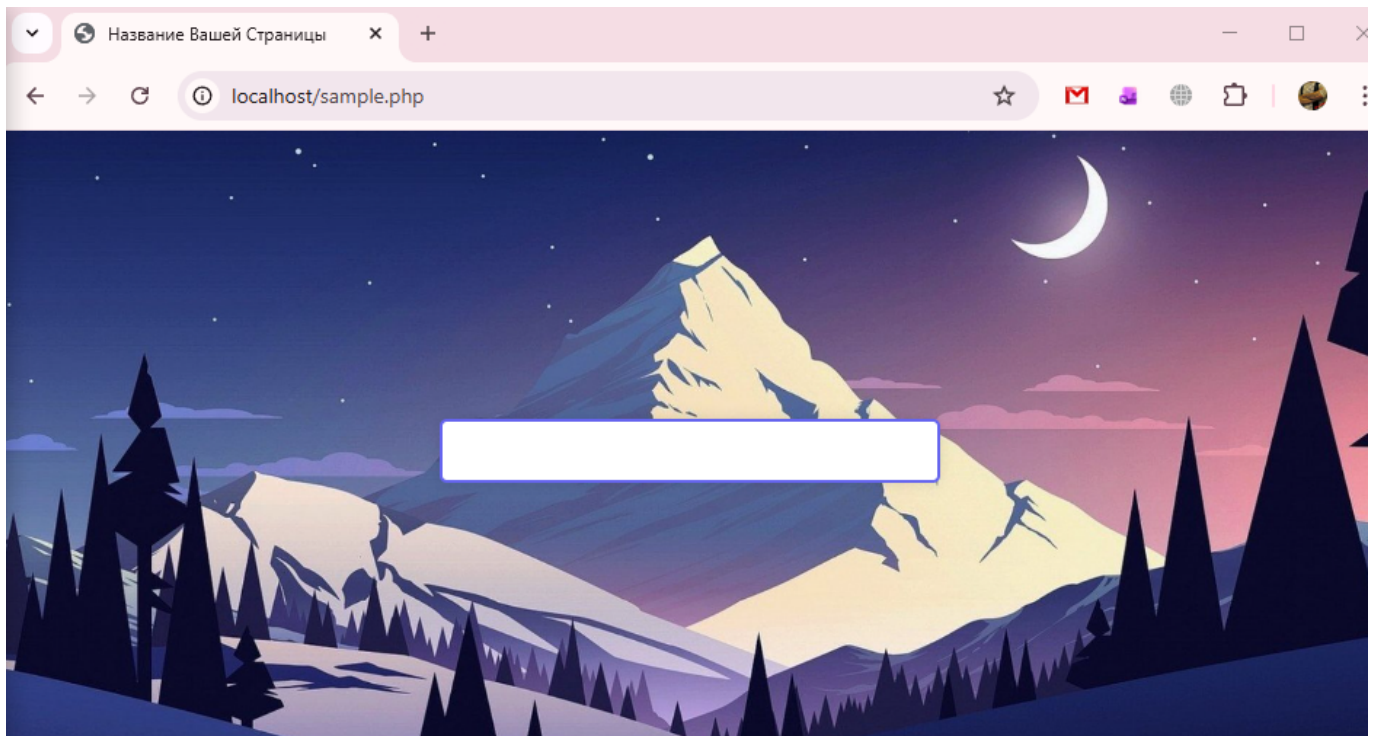


Сохраним это изображение в директории `css` нашего проекта и перейдем к стилю элемента `body`, прописав нижеприведенный код в начале файла `style.css`

[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

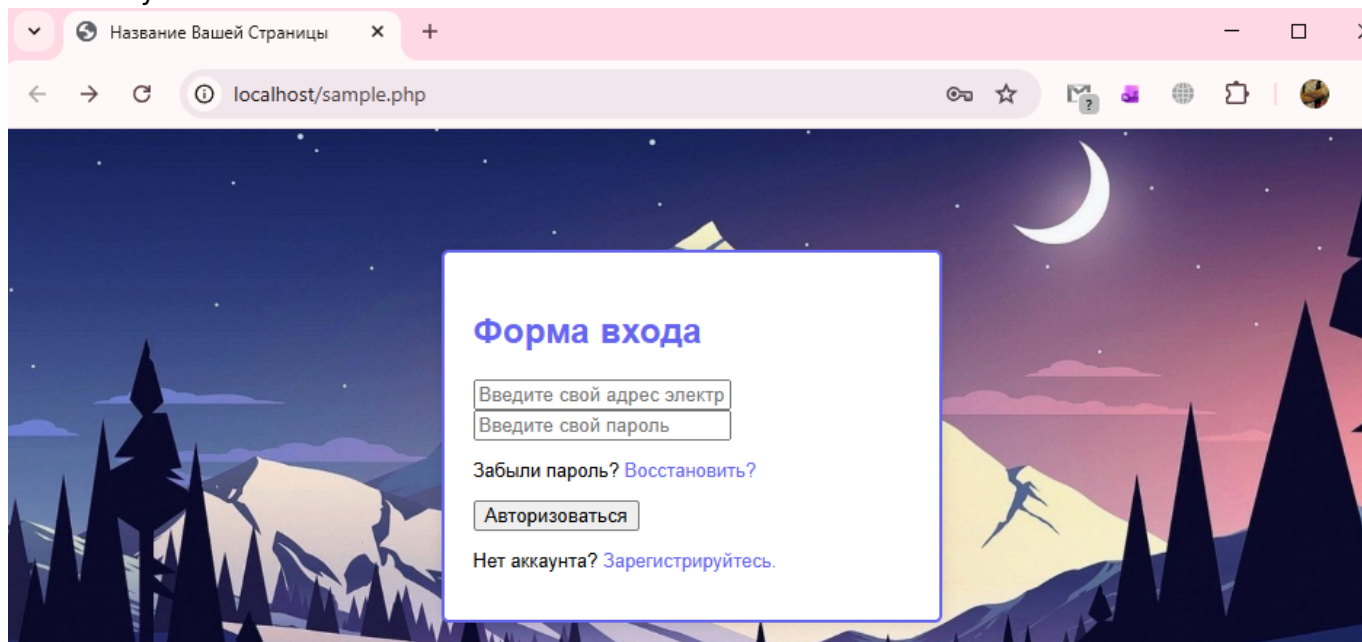
Вставим в наш container код формы с элементами заголовка, ввода, параграфа и кнопки

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

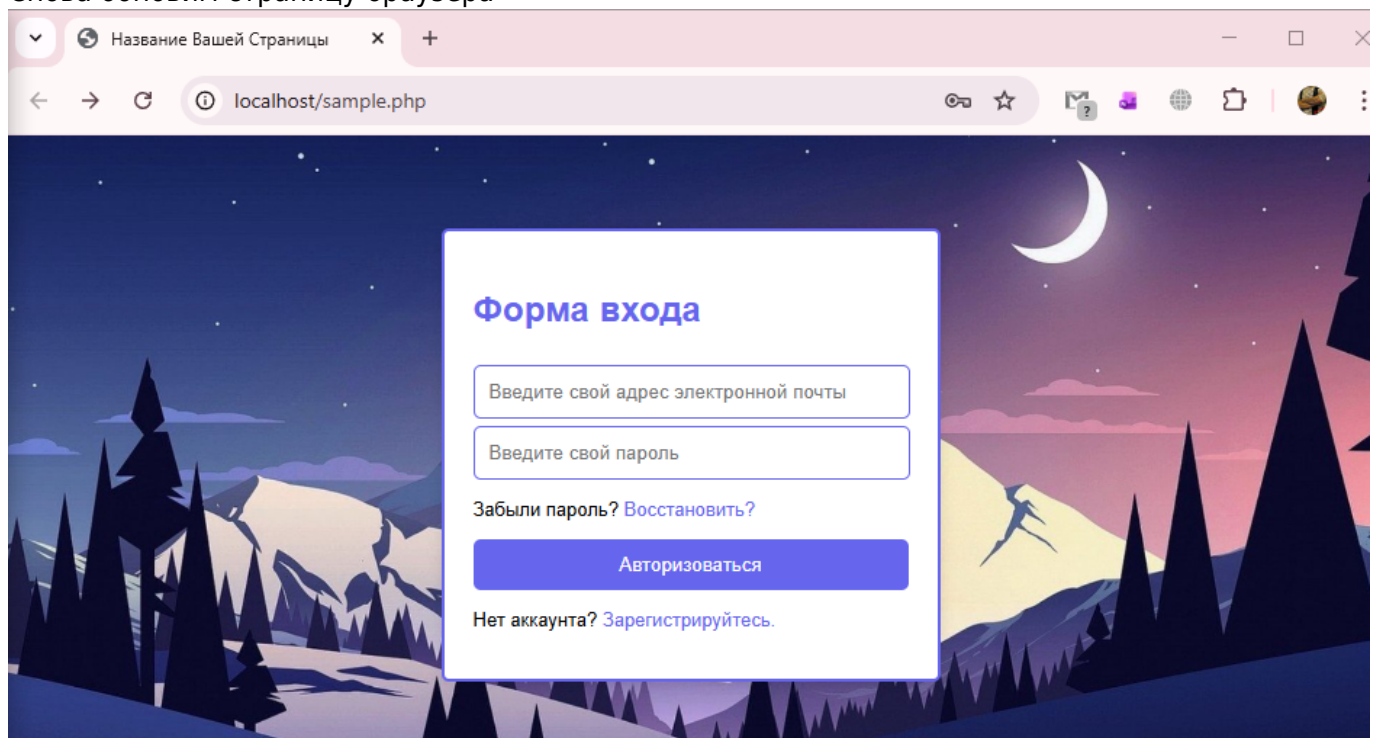
```
/*Стиль поля ввода*/
input {
  width: 93%;
```

```
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



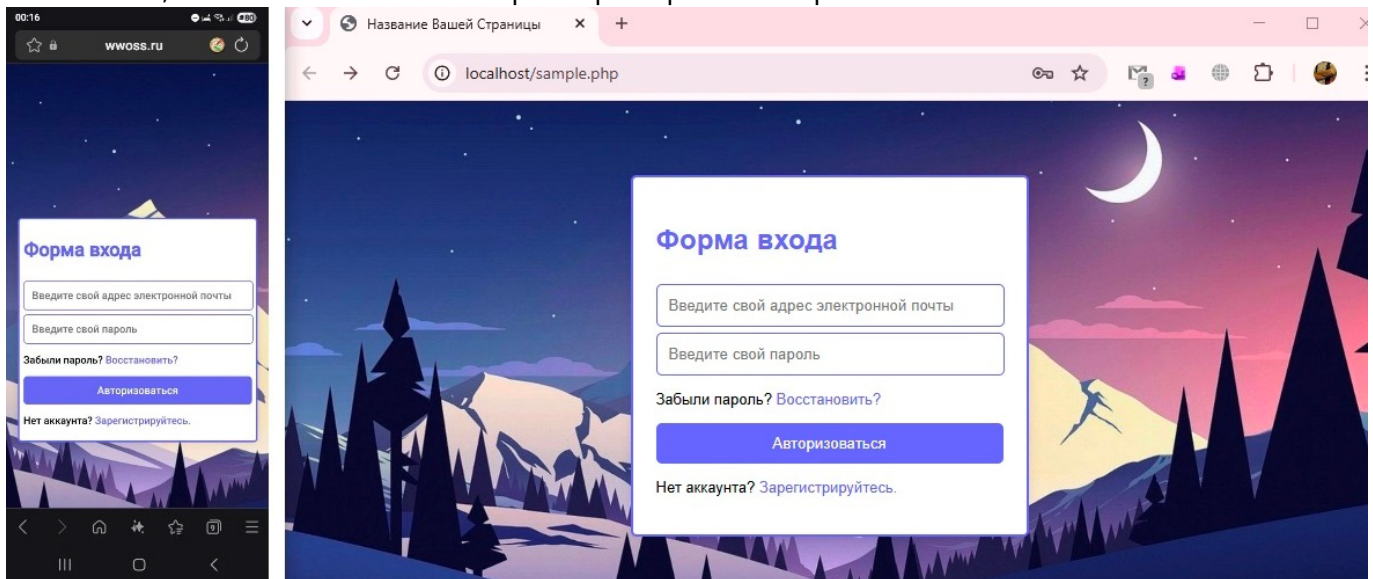
И последнее, что мы пропишем в файле `style.css`, это медиа-запрос, используемый для адаптивного веб-дизайна.

`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.



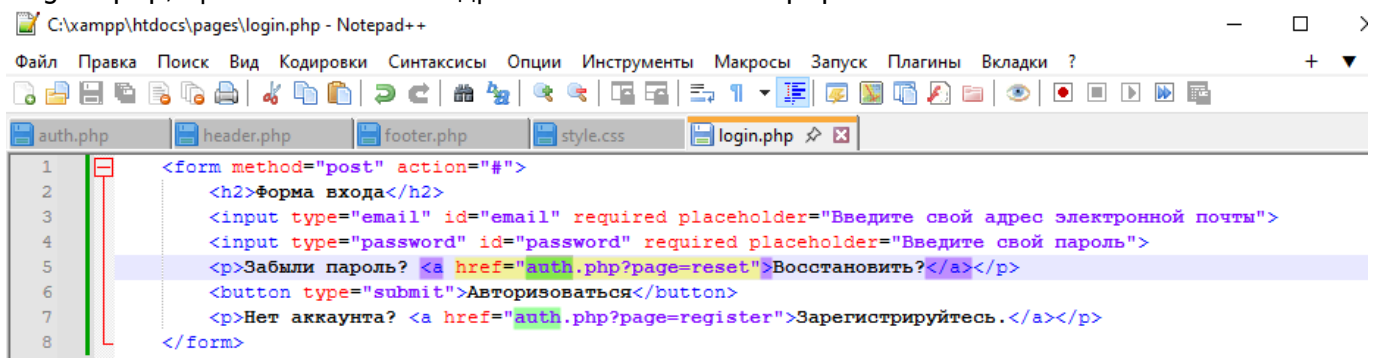
Динамическое подключение страниц

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)
- 2af.php - Форма 2х уровневой авторизации (2-Level Authorization Form)
- logout.php - Выход из системы (logout)

Переименуем, находящийся в корне директории наш шаблон sample.php в auth.php, а так же создадим файл login.php в директории pages.

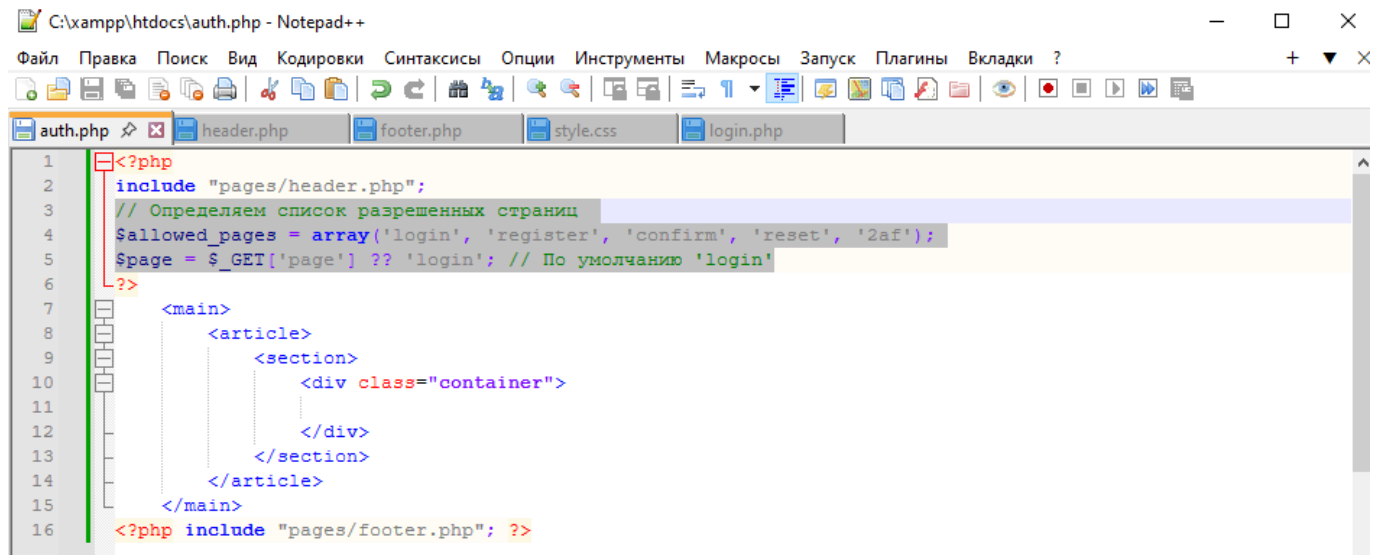
Из файла auth.php удалим весь код из `<div class="container">` и запишем это в файл login.php, при этом изменив адреса ссылок на auth.php



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', '2af');
$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.';
}
?>
```

```
C:\xampp\htdocs\auth.php - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
auth.php header.php footer.php style.css login.php
1 <?php
2 include "pages/header.php";
3 // Определяем список разрешенных страниц
4 $allowed_pages = array('login', 'register', 'confirm', 'reset', '2af');
5 $page = $_GET['page'] ?? 'login'; // По умолчанию 'login'
6 ?>
7
8 <main>
9     <article>
10         <section>
11             <div class="container">
12                 <?php
13                     // Получаем значение из URL: index.php?page=login
14                     // Проверьте, находится ли запрошенная страница в списке разрешенных.
15                     if (in_array($page, $allowed_pages)) {
16                         // Безопасно подключить внешний файл
17                         include('pages/' . $page . '.php');
18                     } else {
19                         // Обработать некорректный запрос страницы (страница не найдена).
20                         echo 'Page not found.';
21                     }
22                 ?>
23             </div>
24         </section>
25     </article>
26 </main>
27 <?php include "pages/footer.php"; ?>
```

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

Permanent link:
https://wwoss.ru/doku.php?id=software:development:demo:adaptive_registration_form_sample&rev=1769466284

Last update: 2026/01/27 01:24

