1. Введение

Это веб-приложение разработано для публикации и управления статьями. Пользователи могут создавать, редактировать и комментировать посты, а также взаимодействовать друг с другом через комментарии. Приложение построено на основе клиент-серверной архитектуры с использованием PHP и MySQL для серверной части, и Bootstrap для стилизации интерфейса.

Основные функции:

* Регистрация и аутентификация пользователей.
* Создание, редактирование и удаление статей.
* Комментирование постов.

Технологический стек:

* Фронтенд: HTML, CSS, JavaScript, Bootstrap.
* Бэкенд: PHP 8.1, MySQL.
* Сервер: Apache.

2. Архитектура приложения

2.1 Общая схема

Приложение использует клиент-серверную архитектуру. Пользователи взаимодействуют с интерфейсом через браузер, отправляя запросы на сервер, который обрабатывает эти запросы с помощью PHP и MySQL, а затем возвращает данные для отображения на стороне клиента.

2.2 Взаимодействие компонентов

1. Пользователь взаимодействует с веб-интерфейсом для создания постов, комментариев или редактирования контента.
2. Запросы отправляются через формы с использованием методов HTTP (POST/GET).
3. PHP обрабатывает запросы и взаимодействует с MySQL для сохранения или получения данных.
4. Ответы возвращаются на клиент в виде HTML или перенаправления на другие страницы.

2.3 Основные модули

* vendor/connect.php: Модуль подключения к базе данных;
* index.php: Главная страница, отображение погоды и новостей;
* Login.php: Страница авторизации пользователя;
* Register.php: Страница регистрации пользователя;
* profile.php: Страница профиля пользователя;
* clause.php: Страница с опубликованными статьями;
* clause\_edit.php: Редактирование статей;
* 3. Разработка фронтенда

3.1 Установка окружения

Для разработки фронтенда вам потребуется локальный веб-сервер (например, OpenServer или XAMPP).

3.2 Структура проекта.

css/

bosstrap.css  
normalize.css  
reset.css  
style.css  
  
fonts/  
Roboto-Bold.ttf  
Roboto-Medium.ttf  
Roboto-Regular.ttf  
  
icons/  
icon1.svg  
  
img/  
img-main.png  
  
js/  
boostrap.bundle.js  
  
uploads/  
photo1.png  
  
vendor/  
connect.php  
helpers.php  
logout.php  
  
./  
index.php  
register.php  
login.php

##### profile.php comments.php clause.php clause\_edit.php weather\_cache.json 3.3 Взаимодействие с сервером через формы

Все формы отправляют данные на сервер с помощью стандартных HTTP-запросов (POST/GET). Пример формы для добавления комментария:

<form action="clause.php" method="POST" class="form\_\_clause">

<div class="form-group">

<label for="news\_category" class="form-label">Категория статьи</label>

<select name="news\_category" class="form-control" required>

<option value="" disabled selected>Выберите категорию</option>

<?php while ($category = mysqli\_fetch\_assoc($result\_categories)): ?>

<option value="<?= htmlspecialchars($category['id']) ?>"><?= htmlspecialchars($category['title']) ?></option>

<?php endwhile; ?> </select></div>

<div class="form-group">

<label for="news\_name" class="form-label">Название статьи</label>

<input type="text" class="form-control" name="news\_name" placeholder="Введите название статьи" required>

</div>

<div class="form-group">

<label for="news\_content" class="form-label">Содержание статьи</label>

<textarea class="form-control" name="news\_content" placeholder="Введите содержание статьи" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Отправить</button>

#### </form> 4. Разработка бэкенда

##### 4.1 Подключение к базе данных (connect.php)

Для подключения к базе данных используется MySQL с помощью PHP:

<?php

$conn = mysqli\_connect('localhost', 'root', '' , 'repeat');

if(!$conn){

die('connection failed');

}

?>

##### 4.2 Обработка запросов (Пример добавления комментария)

Пример добавления комментария пользователем:

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

if (isset($\_SESSION['user\_id'])) {

$userID = $\_SESSION['user\_id'];

$comment = trim($\_POST['comment']);

if (!empty($comment)) {

$sql = "INSERT INTO comments (post\_id, user\_id, comment, created\_at) VALUES ('$postID', '$userID', '$comment', NOW())";

if (mysqli\_query($conn, $sql)) {

$message\_success = "Комментарий успешно добавлен!";

} else {

$message\_error = "Ошибка при добавлении комментария: " . mysqli\_error($conn);

}

} else {

$message\_error = "Комментарий не может быть пустым!";

}

}

}

##### 4.3 Аутентификация

Регистрация и авторизация реализованы через обработку POST-запросов с проверкой данных пользователя в базе:

<?php

require 'vendor/connect.php';

require 'vendor/helpers.php';

session\_start();

$message\_error = '';

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

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

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

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

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

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

$role\_id\_user = 1;

if (strlen($password) < 6) {

$message\_error = 'Длина пароля должна быть более 6 символов';

}

// Проверка уникальности логина

$check\_login = "SELECT id FROM users WHERE login = '$login'";

$check\_login\_result = mysqli\_query($conn, $check\_login);

if (mysqli\_num\_rows($check\_login\_result) > 0) {

$message\_error .= " Данный пользователь уже зарегистрирован с этим логином";

}

// Проверка уникальности email

$check\_email = "SELECT id FROM users WHERE email = '$email'";

$check\_email\_result = mysqli\_query($conn, $check\_email);

if (mysqli\_num\_rows($check\_email\_result) > 0) {

$message\_error .= " Данный пользователь уже зарегистрирован с этим E-mail";

}

// Проверка уникальности телефона

$check\_phone = "SELECT id FROM users WHERE phone = '$phone'";

$check\_phone\_result = mysqli\_query($conn, $check\_phone);

if (mysqli\_num\_rows($check\_phone\_result) > 0) {

$message\_error .= " Данный пользователь уже зарегистрирован с этим телефоном";

}

// Если ошибок нет, вставляем данные

if (empty($message\_error)) {

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

$insert\_sql = "INSERT INTO users (login, full\_name, password, phone, email) VALUES ('$login', '$full\_name', '$hashed\_password', '$phone', '$email')";

if (mysqli\_query($conn, $insert\_sql)) {

header('location: login.php');

exit();

} else {

$message\_error = 'Ошибка при регистрации: ' . mysqli\_error($conn);

}

}

}

?>  
  
<?php

require 'vendor/connect.php';

require\_once 'vendor/helpers.php';

session\_start();

// Проверяем, вошёл ли пользователь в систему

if (isset($\_SESSION['user\_id'])) {

header('Location: profile.php');

exit();

}

$message\_error = '';

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

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

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

$sql = "SELECT \* FROM users WHERE login = '$login'";

$result = mysqli\_query($conn, $sql);

if ($result && mysqli\_num\_rows($result) > 0) {

$row = mysqli\_fetch\_assoc($result);

if (password\_verify($password, $row['password'])) {

$\_SESSION['user\_id'] = $row['id'];

header('Location: profile.php');

exit();

} else {

$message\_error = "Некорректные данные";

}

} else {

$message\_error = "Пользователь с таким логином не найден";

}

}

?>