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

* connect.php: Модуль подключения к базе данных;
* index.php: Главная страница, отображение погоды и новостей;
* auto.php: Страница авторизации пользователя;
* reg.php: Страница регистрации пользователя;
* cabinet.php: Страница профиля пользователя;
* index.php: Страница с опубликованными статьями;
* update\_news.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.

boots/

bootstrap.bundle.js  
bootstrap.css

docs/

Рукводоство программиста.docx  
Руководство пользователя.docx  
  
fonts/  
Lato-Bold.ttf  
Lato-ExtraBold.ttf  
Lato-Medium.ttf  
Lato-Regular.ttf  
Lato-SemiBold.ttf  
RobotoSlab.ttf  
  
img/  
avatar1.png  
butterfly.svg  
gitara.svg  
Hamburger\_menu.svg  
like\_clear.svg  
Man.svg  
nigga.svg  
option.svg  
peisaj.svg  
Search.svg  
stateNY.svg  
Sun.svg  
  
News/  
auto.php  
cabinet.php  
cap.php  
connect.php  
create\_news.php  
delete.php  
get\_weather.php  
index.php  
logout.php  
post.php  
reg.php  
styles.css  
update\_news.php

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 class="box\_form" method="post">

                    <p class="regist">Добавить новость</p>

                    <div class="mb-3">

                        <label for="title" class="form-label">Заголовок</label>

                        <input type="text" name="title" class="form-control" id="title" required>

                    </div>

                    <div class="mb-3">

                        <label for="description" class="form-label">Описание новости</label>

                        <textarea name="description" class="form-control" id="description" rows="3" required></textarea>

                    </div>

                    <div class="mb-3">

                        <label for="category" class="form-label">Категория новости</label>

                        <select name="id\_category" class="form-select" id="category" required>

                            <option value="all">Все</option>

                            <?php foreach ($services as $service): ?>

                                <option value="<?= $service['id\_category'] ?>"

                                <?= isset($\_GET['name\_category']) && $\_GET['name\_category'] == $service['id\_category'] ? 'selected' : '' ?>><?= $service['name\_category'] ?></option>

                            <?php endforeach; ?>

                        </select>

                    </div>

                    <button type="submit" class="btn btn-primary">Добавить новость</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 = new mysqli("localhost", "root", "root", "news");

##### if ($conn->connect\_error) {

##### die("Connection failed: " . $conn->connect\_error);

##### }

##### ?>

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

Пример добавления комментария пользователем:

// Получение комментариев к посту

$commentsSql = "SELECT c.id\_comment, c.desc\_comment, c.id\_user, c.likes, u.nickname

FROM Comments c

JOIN Users u ON c.id\_user = u.id

WHERE c.id\_news = $id";

$commentsResult = $conn->query($commentsSql);

// Добавление комментария

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['comment'])) {

$comment = $conn->real\_escape\_string($\_POST['comment']);

$userId = $id; // Замените на ID текущего пользователя, если вы авторизованы

$insertSql = "INSERT INTO Comments (desc\_comment, id\_user, id\_news) VALUES ('$comment', $userId, $id)";

if ($conn->query($insertSql) === TRUE) {

header("Location: post.php?id=$id"); // Перезагрузка страницы для отображения нового комментария

exit();

}

}

##### $commentsSql = "SELECT c.id\_comment, c.desc\_comment, c.id\_user, c.likes, u.nickname

##### FROM Comments c

##### JOIN Users u ON c.id\_user = u.id

##### WHERE c.id\_news = $id";

##### $commentsResult = $conn->query($commentsSql);

##### // Обработка удаления комментария

##### if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['delete\_comment'])) {

##### $commentId = intval($\_POST['comment\_id']);

##### $deleteCommentSql = "DELETE FROM Comments WHERE id\_comment = $commentId";

##### $conn->query($deleteCommentSql);

##### header("Location: post.php?id=$id");

##### exit();

##### }

##### // Обработка редактирования комментария

##### if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['edit\_comment'])) {

##### $commentId = intval($\_POST['comment\_id']);

##### $editedComment = $conn->real\_escape\_string($\_POST['edited\_comment']);

##### $updateCommentSql = "UPDATE Comments SET desc\_comment = '$editedComment' WHERE id\_comment = $commentId";

##### $conn->query($updateCommentSql);

##### header("Location: post.php?id=$id");

##### exit();

##### }

##### 4.3 Аутентификация

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

<?php

require "connect.php";

require "cap.php";

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

    $login = $\_POST["login"];

    $password = $\_POST["pass"];

    $nickname = $\_POST["name"];

    $img = "avatar1.png";

    // Проверка на существование пользователя с таким же nickname

    $check\_double = mysqli\_query($conn, "SELECT \* FROM Users WHERE nickname = '$nickname'");

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

        echo "<script>

                alert('Пользователь с таким Никнеймом уже создан, попробуйте другой!');

                location.href = 'reg.php';

              </script>";

    } else {

        $sql\_users = "INSERT INTO Users (nickname, login, password, img)

                      VALUES ('$nickname', '$login', '$password', '$img')";

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

            echo "<script>

                    alert('Успешная Регистрация!');

                    location.href = 'auto.php';

                  </script>";

            exit();

        } else {

            echo "Ошибка регистрации: " . mysqli\_error($conn);

        }

    }

}

?>  
  
<?php

require "connect.php";

require "cap.php";

    if (isset($\_POST['login']) && isset($\_POST['pass'])) {

        $login = $\_POST["login"];

        $password = $\_POST["pass"];

        $queryUser = mysqli\_query($conn, "SELECT \* FROM Users WHERE login='$login' AND password='$password'");

    $user = mysqli\_fetch\_array($queryUser);

    $idUser = $user["id"];

    if(!empty($idUser)) {

            setcookie('idUser', $idUser, time() + 3600, "/");

            echo "<script>alert('Вы успешно вошли в аккаунт');

        location.href='cabinet.php';

        </script>";

    } else {

        echo "<script>alert('Данный пользователь не найден!');

        location.href='auto.php';

        </script>";

    }

    }

?>