Цели практической работы

Научиться:

* подключаться к БД из PHP-скрипта,
* выбирать данные из БД для авторизации и страницы профиля пользователя.

Что входит в задание

1. Подключить к сайту БД и переработать форму авторизации на взаимодействие с БД.
2. Создать страницу профиля пользователя.
3. Добавить возможность обмена сообщениями между пользователями (по желанию).

Задание 1. Доработка авторизации

Что нужно сделать

Продолжаем доделывать и развивать сайт, над которым вы работали в предыдущем практическом задании.

Реализуйте авторизацию так, чтобы данные о логинах и паролях пользователей брались не из файлов, а из БД. Старые файлы с логинами и паролями удалите, а данные из них перенесите в БД. Для удобства проверки практической работы установите всем созданным в БД пользователям пароль 123456.

Советы и рекомендации

* При авторизации пользователя не делайте разные запросы в БД: один — для получения его пароля по логину, второй — для получения ID. Всегда старайтесь минимизировать число запросов к БД, выбирайте все нужные поля сразу. Запросы к БД — всегда узкое место в производительности приложения.
* Не применяйте старый механизм авторизации: неправильно будет выбрать всех пользователей, а потом искать по списку подходящего. SQL-запросы поддерживают фильтрацию в запросах.
* Не храните пароли в БД в открытом виде, зашифруйте их перед сохранением. Для шифрации подойдёт функция [password\_hash](https://www.php.net/manual/ru/function.password-hash.php" \t "_blank), а для сравнения шифрованного пароля с нешифрованным — [password\_verify](https://www.php.net/manual/ru/function.password-verify.php" \t "_blank).
* Всегда применяйте SQL-экранирование (или подготовленные запросы) перед подстановкой любых данных в SQL-запрос, вне зависимости от их источника: массив $\_POST, сессия или параметр функции.
* Отдавайте предпочтение работе с БД через [PDO](https://www.php.net/manual/ru/book.pdo.php), а не простыми функциями *mysqli\_\**.

Что оценивается

* В БД созданы пользователи.
* Авторизация осуществляется по данным из БД.
* Пароли пользователей хранятся в хешированном виде.
* Все данные, подставляемые в SQL-запросы, экранируются.

Задание 2. Разработка страницы профиля

Что нужно сделать

Добавьте новую страницу с профилем пользователя. Она должна располагаться по адресу **/profile/**. Доступ к этой странице имеют только авторизованные пользователи. Если неавторизованный пользователь попробует сюда зайти, он должен быть переадресован на форму авторизации.

На странице профиля введите Ф. И. О., email, телефон текущего авторизованного пользователя в виде списка, а также выведите названия всех групп, к которым он привязан.

Советы и рекомендации

* Не переусердствуйте, выбирая информацию для профиля. Не пытайтесь разные по типу данные склеить в один запрос. Выбирайте поля профиля одним запросом, а данные о группах — вторым.
* Всегда применяйте SQL-экранирование (или подготовленные запросы) перед подстановкой любых данных в SQL-запрос, вне зависимости от их источника: массив $\_POST, сессия или параметр функции.
* Отдавайте предпочтение работе с БД через [PDO](https://www.php.net/manual/ru/book.pdo.php), а не простыми функциями *mysqli\_\**.

Что оценивается

* Выбор информации о пользователе осуществляется одним запросом.
* Выбор всех групп пользователя так же реализован одним запросом.
* На странице /profile/ отображается профиль текущего авторизованного пользователя. Доступ к ней закрыт для неавторизованных пользователей.
* Все данные, подставляемые в SQL-запросы, экранируются.

Задание 3. Обмен сообщениями между пользователями (по желанию)

В качестве дополнительной практики выполните задание на взаимодействие с БД и формами на сайте. Необходимо добавить возможность пользователям обмениваться друг с другом личными сообщениями.

Что нужно сделать

**Страница со списком сообщений**

Создайте новый раздел на сайте для сообщений по адресу **/posts/**.

Любой зарегистрированный пользователь после авторизации на этой странице видит сообщение «Вы сможете отправлять сообщения после прохождения модерации».

Пользователь, имеющий право писать сообщения после авторизации на этой странице, видит два списка сообщений: непрочитанных и прочитанных. В этих списках в виде ссылки выводятся сообщения. В качестве текста ссылки используется заголовок сообщения и название раздела. Ссылка ведёт на детальную страницу просмотра сообщения.

Также на странице со списком отображается ссылка «Написать сообщение». При клике на неё выполняется переход на страницу добавления сообщения **/posts/add/**.

**Страница добавления сообщения**

На странице **/posts/add/** должна быть форма, состоящая из следующих полей:

* Заголовок — текстовое поле.
* Текст сообщения — текстовое поле.
* Пользователь (кому отправить сообщение) — выпадающий список, состоящий из списка всех пользователей, которые состоят в группе «Пользователь, имеющий право писать сообщения».
* Раздел сообщения — выбор доступных разделов в виде списка, состоящего из названий разделов.
* Кнопка «Отправить».

После нажатия на кнопку «Отправить» в случае успешной валидации (все поля обязательны для заполнения) сообщение добавляется в таблицу.

При возникновении ошибки валидации или ошибки добавления записи текст этой ошибки выводится обратно в форму, а также в форму подставляются введённые ранее данные.

**Детальная страница сообщения**

Адрес страницы — **/post/detail.php?id={id}**, где {id} — это идентификатор сообщения. Доступ к сообщению имеют только его автор и получатель.

На этой странице отображается следующая информация:

* заголовок и дата отправки сообщения,
* кто написал — Ф. И. О. и его email,
* полный текст сообщения.

Также выводится ссылка «Вернуться к списку сообщений».

После перехода на детальную страницу сообщения у него устанавливается признак, что сообщение прочитано.

Если сообщение с указанным идентификатором не найдено, то вместо информации о нём выводится сообщение «Сообщение не найдено», а статусом HTTP-ответа устанавливается 404.

Советы и рекомендации

* Не забывайте о SQL-экранировании и защите от SQL-инъекций.
* Список прочитанных и непрочитанных сообщений удобнее выбирать разными запросами, для этого создайте несколько вспомогательных функций.

Что оценивается

* Созданы страницы для просмотра и добавления сообщений.
* В форме добавления сообщения реализована валидация на обязательность заполнения всех полей.
* Добавление сообщений работает. После добавления сообщения оно, при последующем входе на страницу, появляется в списке непрочитанных у его получателя.
* Для каждого сообщения открывается детальная страница, доступная только для автора и получателя.
* После открытия этой страницы сообщение, при последующем входе на страницу со списком сообщений, перемещается в список прочитанных.
* Все данные, подставляемые в SQL-запросы, экранируются.
* При попытке зайти на страницу несуществующего сообщения выводится ошибка 404.