

## Создать форму регистрации / авторизации

Подключаемся к базе MySQL в нашем терминале Линукса. Затем командой в терминале создаём нужную базу данных для информации по регистрации user\_access с кодировкой UTF-8. Создаём пользователя admin1, идентифицированного паролем. Даём этому пользователю привилегии в нашей базе user\_access на выборку записей, вставку записей и обновление:

```
sudo mysql -S '/opt/lampp/var/mysql/mysql.sock'  
create database user_access character set utf8 collate utf8_unicode_ci;  
use user_access;  
create user 'admin1'@'localhost' identified by '12345';  
grant select, insert, update on user_access.* to 'admin1'@'localhost';
```

БД будет состоять из двух таблиц: Registrations и Enters. В таблице Registrations будет храниться список успешно зарегистрированных пользователей. В таблице Enters будет храниться список всех успешных попыток входа в систему, авторизаций.

Таблица Registrations состоит из четырёх полей: Login1 (логин) (первичный ключ, уникальное поле), Password1 (хеш пароля), Email1 (электронный адрес) (уникальное поле, так как на один электронный адрес можно зарегистрировать только одного пользователя), Date1 (дата и время регистрации).

Таблица Enters состоит из трёх полей: Login1 (логин, который пользователь вводит для авторизации; здесь уже не уникальное поле, так как по данному логину можно успешно входить в систему многократно), Date2 (дата и время входа пользователя с данным логином), Rate2 (показатель того, что пользователь находится в системе: значение 0 — пользователь корректно вышел из системы (нажатием соответствующей кнопки), значение 1 — пользователь всё ещё авторизован в системе). Для всё ещё авторизованных в системе попытка входа приведёт к сообщению о том, что они уже авторизованы.

Такая реализация администратору базы данных (admin1) даёт возможность узнать, например, количество авторизованных пользователей в данный момент времени. Узнать можно запросом:

```
SELECT SUM(Rate2)  
FROM Enters;
```

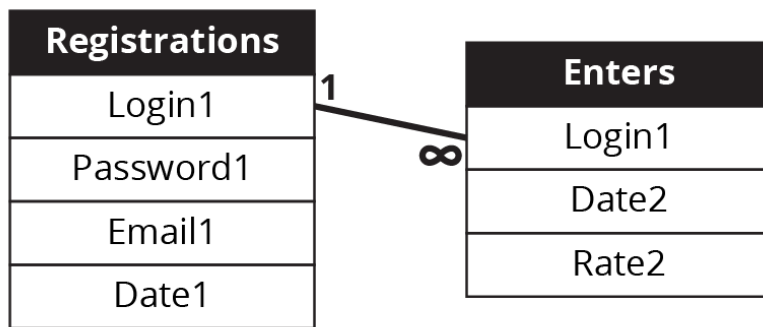
Date1 и Date2 возвращают время в московском часовом поясе, так как сервер базы находится в Москве. Чтобы обеспечить возврат именно московского времени, в файле php.ini нужно установить в значение Europe/Moscow параметр date.timezone.

Login1 таблицы Enters — это внешний ключ, связанный с Login1 таблицы Registrations. Связь таблицы Registrations и таблицы Enters — это связь «один ко многим». Неизбыточность таблицы Enters будет обеспечиваться первичным ключом по совокупности её обоих полей: Login1, Date2 (то есть не может один и тот же пользователь войти одновременно в систему более одного раза). Это означает, что после выхода из системы, пользователю необходимо подождать хотя бы 1 секунду, чтобы иметь возможность войти снова. Такое требование не только обеспечивает избыточность Enters, но и представляется логичным, потому что физически невозможно за одну секунду ввести логин и пароль авторизации.

Структура таблиц показана ниже.

На логин выделяем, например, 10 символов, на хеш пароля — 255 символов (сам пароль пусть будет от 6 до 10 символов), на электронный адрес — 30 символов, на дату/время — 20 символов.

Командой в терминале создаём таблицы:



Первая страница, с которой начинается работа с системой — файл form1.html. Этот файл ведёт в форму регистрации (registr.html) для тех пользователей, которые ещё незарегистрированы, и в форму авторизации (authoriz.html) для тех пользователей, которые уже прошли регистрацию и хотят войти в систему, авторизоваться. При этом в форме авторизации есть возможность восстановления логина и обновления пароля для тех пользователей, которые забыли свои регистрационные данные. Восстановление производится по зарегистрированному адресу электронной почты.

Введение данных в форму регистрации и отправка их на сервер (localhost) приводит прежде всего к проверке корректности введённых данных, в соответствии с теми описаниями, которые даны в форме регистрации. Кроме того проверяется отсутствие уже зарегистрированного пользователя с данными логином и паролем. Пароли могут повторяться у разных пользователей. Однако повторение у разных пользователей логинов и адресов e-mail не допускается.

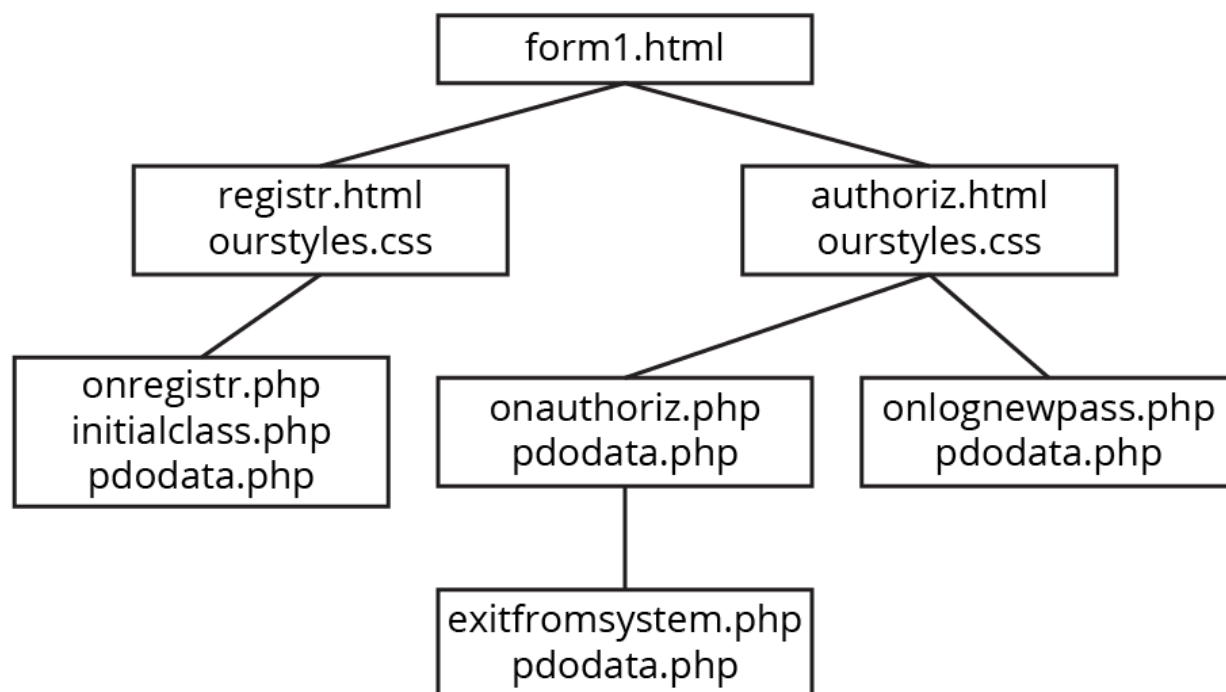
Аналогично этому корректность введённых данных проверяется на сервере при введении данных в форму авторизации или в форму восстановления логина и обновления пароля.

В случае, если данные введены в какую-либо из этих форм некорректно, выдаётся соответствующее сообщение, и программа (PHP) завершает работу. В этом случае можно вернуться назад и повторить ввод данных.

Основные обрабатываемые ошибки:

1. введение пароля или адреса e-mail меньше или больше заданных величин (меньше 6 или больше 10 символов для пароля, больше 30 символов для электронного адреса);
2. несовпадение паролей при регистрации;
3. наличие в электронном адресе более, чем одного символа собаки (@) или отсутствие такого символа вообще;
4. начало или окончание адреса с символа собаки или точки;
5. наличие двух точек, стоящих рядом в адресе;
6. начало некоторого элемента электронного адреса с дефиса;
7. начало логина с цифры;
8. наличие в пароле только цифр или только латинских букв;
9. наличие в логине или в пароле символов, отличающихся от латинских букв или цифр;
10. наличие в электронном адресе символов, отличающихся от латинских букв, цифр, точек, дефиса и подчёркивания (не считая символа собаки);
11. попытка зарегистрировать снова уже зарегистрированный логин или уже зарегистрированный адрес электронной почты;
12. попытка войти в систему, авторизоваться, с незарегистрированными логином или адресом электронной почты;
13. попытка авторизоваться с неправильным паролем;
14. попытка авторизоваться второй раз, не выйдя корректно из системы;
15. попытка обновить пароль для незарегистрированного адреса электронной почты;
16. попытка обновить пароль для чего-либо, что заведомо не является корректным электронным адресом (не содержит символа собаки).

Структура файлов программ:



Примеры регистраций в базе:

Логин	Пароль	Электронный адрес
abraka23	dabra45	moy@site.2.com.ru
nektO16H	4954O73	nekot.oryj@adress.net
sasha87	sasha87	sasha87@gmail.com
natasha	etoparol10	kate@yandex.ru
platochek	5Tetradka	myshka.koshka@zveri.net

Экспортируем сформированные таблицы базы.

Для экспорта заходим в rhpmuadmin, выбираем базу данных user\_access. Справа идём на вкладку Export. Метод экспорта — Quick, формат экспорта — SQL. Нажимаем Go. Сформировался файл user\_access.sql, содержащий таблицы базы.