

Nutzerverwaltungssoftware: Dokumentation

1 Übersicht der im Projekt vorhandenen Dateien

\Data\www\

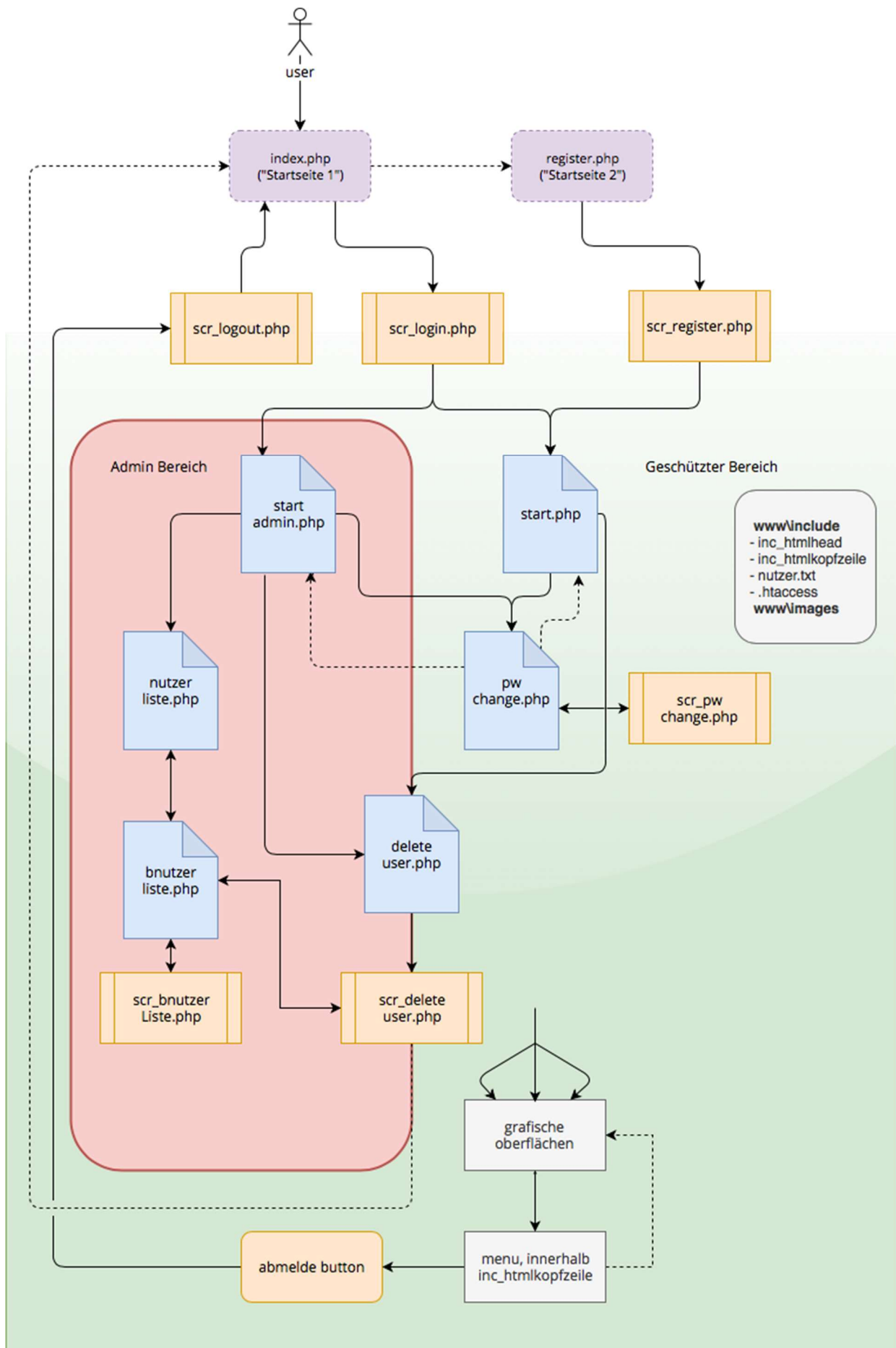
Name	Änderungsdatum	Typ	Größe
images	03.10.2019 19:09	Dateiordner	
include	07.10.2019 22:40	Dateiordner	
bnutzerListe.php	07.10.2019 22:40	PHP-Datei	2 KB
deleteuser.php	07.10.2019 22:40	PHP-Datei	3 KB
design.css	07.10.2019 22:40	Kaskadierendes St...	5 KB
index.php	07.10.2019 22:40	PHP-Datei	2 KB
nutzerListe.php	07.10.2019 22:40	PHP-Datei	2 KB
pwchange.php	07.10.2019 22:40	PHP-Datei	3 KB
README.md	07.10.2019 22:40	MD-Datei	1 KB
register.php	07.10.2019 22:40	PHP-Datei	2 KB
scr_bnutzerListe.php	07.10.2019 22:40	PHP-Datei	1 KB
scr_deleteuser.php	07.10.2019 22:40	PHP-Datei	1 KB
scr_login.php	07.10.2019 22:40	PHP-Datei	2 KB
scr_logout.php	07.10.2019 22:40	PHP-Datei	1 KB
scr_nutzerLoeschen.php	07.10.2019 22:40	PHP-Datei	1 KB
scr_pwchange.php	07.10.2019 22:40	PHP-Datei	2 KB
scr_register.php	07.10.2019 22:40	PHP-Datei	2 KB
start.php	07.10.2019 22:40	PHP-Datei	2 KB
startadmin.php	07.10.2019 22:40	PHP-Datei	2 KB

\Data\www\include

Name	Änderungsdatum	Typ	Größe
.htaccess	08.10.2019 09:42	HTACCESS-Datei	1 KB
inc_htmlhead	08.10.2019 09:42	Datei	1 KB
inc_htmlkopfzeile.php	15.10.2019 23:14	PHP-Datei	3 KB
nutzer.txt	15.10.2019 23:14	Textdokument	1 KB

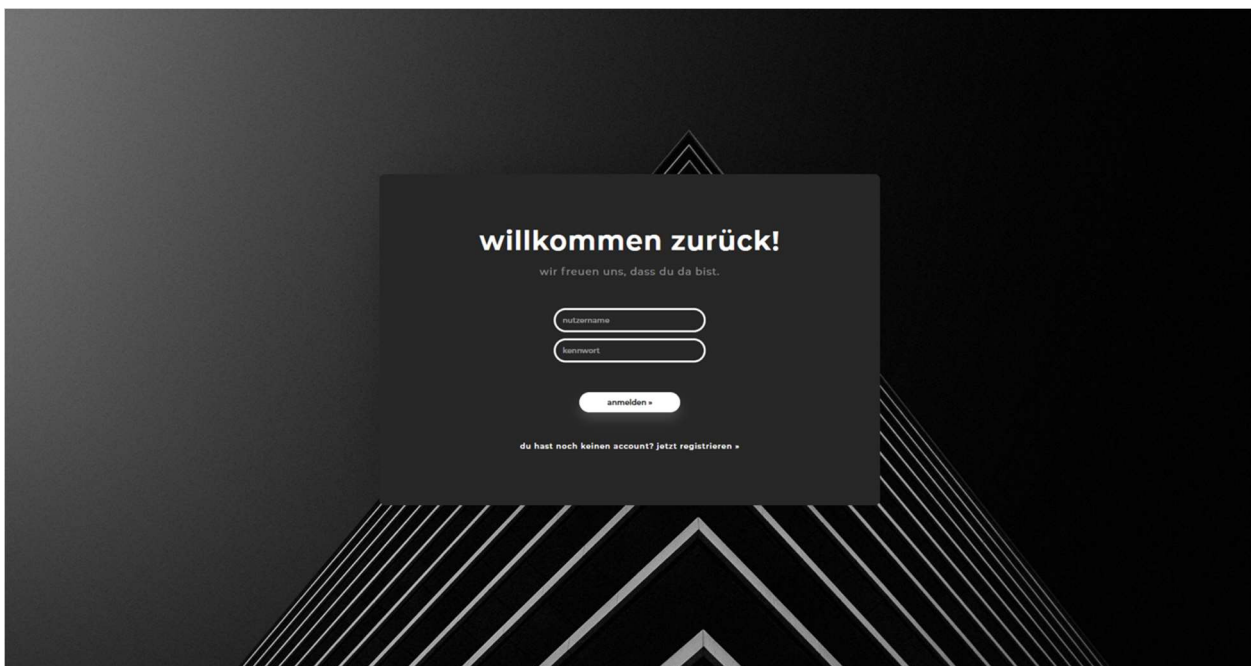
2 Übersicht des Zusammenhangs der PHP Skripte

Skripte heißen scr_name.php, grafische Oberflächen haben keinen Präfix.



3 Beschreibung der Webseitenfunktionen

Betritt ein Nutzer die Seite, ohne eine bestimmte Unterseite anzugeben, landet er auf *index.php*, welche gleichzeitig die Funktion der **Login-Seite** übernimmt. In den beiden Input-Feldern müssen Nutzernamen und Passwort eingegeben werden. (zum Testen: **admin:admin** für Adminzugang, **testuser:user** für Nutzerzugang). Sind die Angaben inkorrekt, wird ein Error angezeigt, andernfalls wird man eingeloggt. All dies passiert mit Hilfe des *scr_login.php* Skripts.

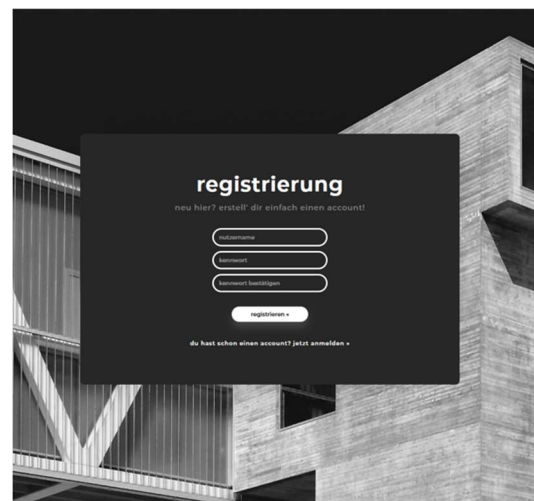


Login-Seite (*index.php*)

Die zweite Möglichkeit auf dieser Seite ist das Klicken auf „**registrieren**“ unten. Man wird auf die *register.php* Seite weitergeleitet.

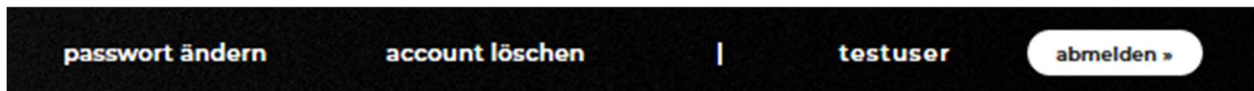
Dort hat ein neuer Nutzer die Möglichkeit, sich einen **neuen Account** zu erstellen. Dazu werden gewünschter Nutzernamen, das gewünschte Kennwort und dasselbe Kennwort nochmals zur Bestätigung eingegeben. Das *scr_register.php* Skript gleicht die Daten ab, erstellt einen neuen Nutzer und loggt diesen automatisch ein.

Sollte der Nutzernamen bereits vergeben sein, oder die Kennwörter nicht übereinstimmen, wird auch hier ein Error angezeigt und der Vorgang muss wiederholt werden.



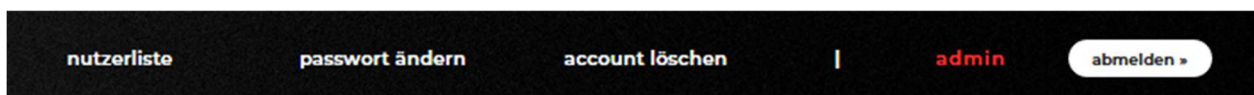
Loggt sich ein User ein, prüft das `scr_login.php` Skript, welche **Rechte** (dazu später mehr) bei dem sich einloggenden Nutzer hinterlegt sind. Ist er ein **normaler Nutzer**, wird er auf die Nutzerstartseite `start.php` weitergeleitet, hat er ein **Adminrecht**, wird er auf die Admin-Startseite `startadmin.php` weitergeleitet. Das Recht wird außerdem in einer Sessionvariable gespeichert, damit dieses jederzeit simpel abgefragt werden kann.

Auf den Startseiten befindet sich bisher nur eine universelle **Kopfzeile** (`inc_htmlkopfzeile.php`), die ein Menu mit verschiedenen Buttons beinhaltet.



Menü, das einem normalen Nutzer angezeigt wird

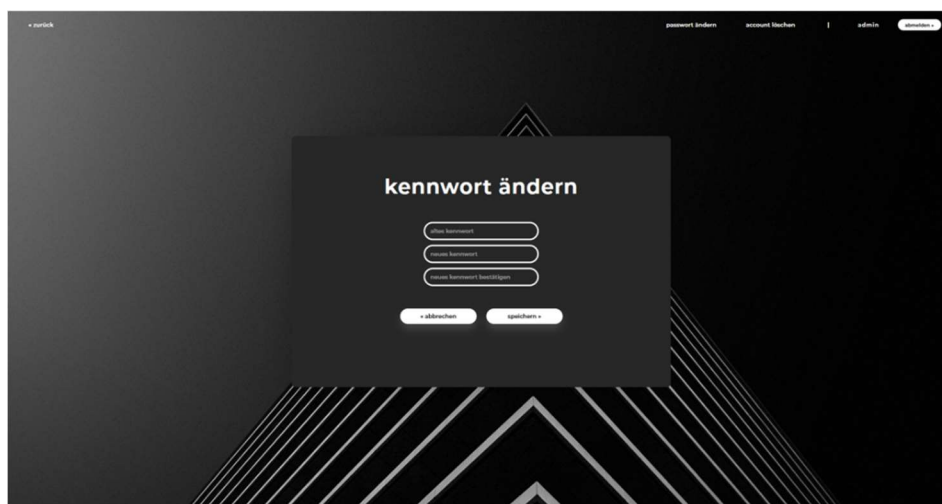
Diese Kopfzeile befindet sich auf jeder grafischen Oberfläche. Auch sie prüft, welches Recht der eingeloggte Nutzer hat. Ist er ein normaler Nutzer, hat dieser die Möglichkeit, sich über einen Button **abzumelden** (`scr_logout.php`), seinen **Account** zu **löschen** (`deleteuser.php`) oder sein **Passwort** zu **ändern** (`pwchange.php`). Gegebenenfalls gibt es außerdem einen **zurück**-Button oben links, mit dem der Nutzer einen Schritt zurückgehen kann.



Menü, das einem Admin angezeigt wird

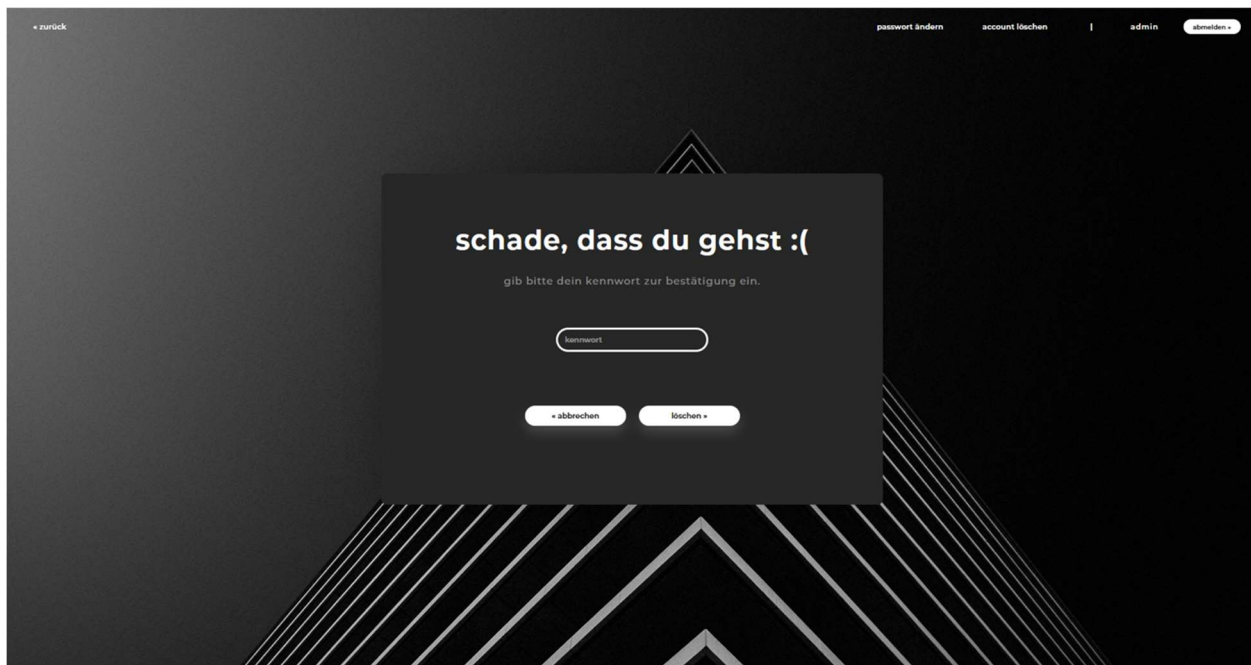
Ein Admin hat zu dem noch die Möglichkeit, die **Nutzerliste** aufzurufen. Außerdem wird sein Name neben dem Abmeldebutton rot angezeigt, wohingegen ein Nutzer einen weißen Anzeigenamen hat.

Die Seiten zum **Account löschen** und zur **Kennwortänderung** sind für beide Nutzergruppen gleich.



Ruft ein Nutzer die **Kennwort-Änderungs-Seite** auf, muss er hier zunächst sein aktuelles Kennwort eingeben, anschließend das gewünschte neue Kennwort und - wie bei der Registrierung - dasselbe Kennwort zur Überprüfung. Ist das aktuelle Kennwort inkorrekt oder stimmen die neuen Kennwörter nicht überein, werden auch hier jeweils verschiedene Fehlermeldungen angezeigt. Sind die Eingaben korrekt, wird eine Erfolgsmeldung angezeigt und der Nutzer kann oben links den zurück-Button nutzen. Auch ein Abbruch während des Vorgangs ist über den abbrechen-Button möglich.

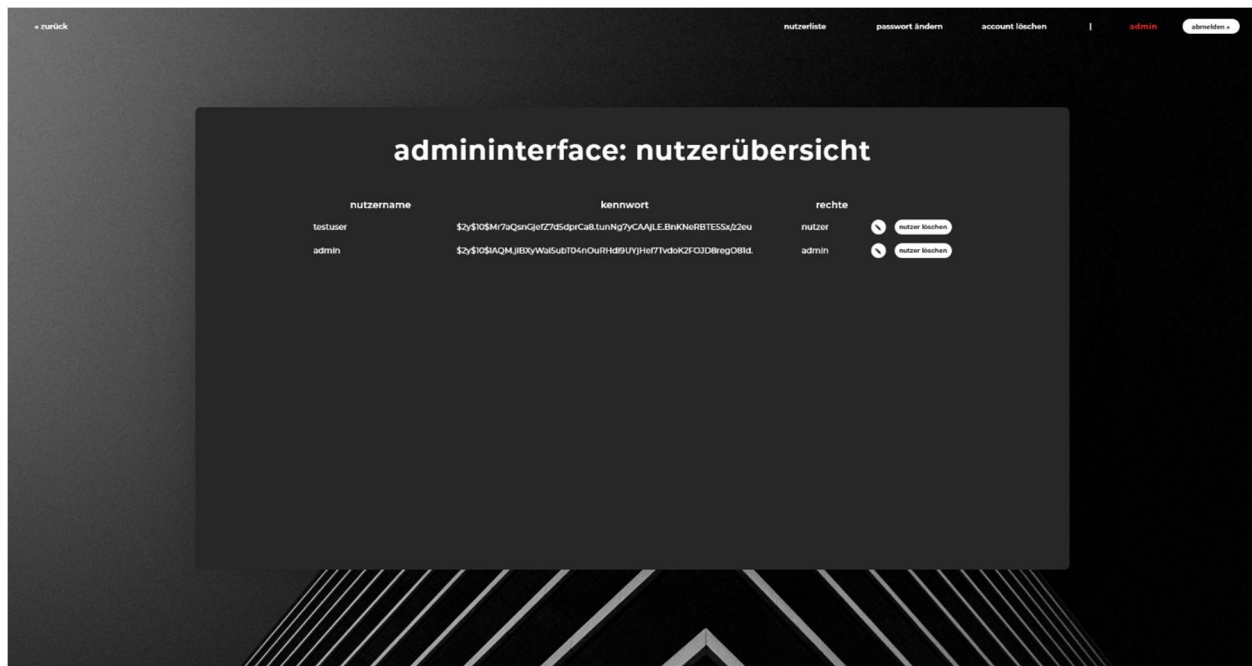
Möchte ein Nutzer seinen **Zugang löschen**, muss er hierzu sein aktuelles Kennwort zur Bestätigung eingeben. Bei erfolgreicher Eingabe wird der Account gelöscht und der Nutzer abgemeldet. Andernfalls wird auch hier eine Fehlermeldung angezeigt.



Seite zum Löschen des eigenen Accounts (*deleteuser.php*)

Das Nutzerlöschen-Skript (*scr_deleteUser.php*) unterscheidet allerdings, woher ein Nutzer kommt und welche Funktion genutzt werden soll. Das gleiche Skript wird nämlich auf der **Admin-Nutzerliste** genutzt.

Betrifft ein Admin diese Seite (*nutzerListe.php*), werden ihm zunächst nur **alle Nutzer** und die dazugehörigen Informationen aufgelistet. Er kann in der jeweiligen Nutzerzeile allerdings noch den **Stift** oder die **löschen-Funktion** nutzen. Der Lösch-Button führt auch hier wieder auf das oben genannte Lösch-Skript zurück, diesmal merkt das Skript allerdings, dass hier der Admin einen Nutzer löschen möchte, sodass der Nutzer hier zwar aus der Konfigurationsdatei (dazu später mehr) gelöscht wird, aber diesmal niemand ausgeloggt wird oder sonstiges. Löscht ein Admin sich selbst, wird er auf die normale *deleteuser.php* Seite zur Bestätigung weitergeleitet.



Admininterface listet alle registrierten User auf (nutzerListe.php)

Klickt der Admin auf den **Stift**, wird er auf eine andere Seite (*bnutzerliste.php*) geleitet, die erstmal ähnlich aufgebaut ist und mit dem *scr_bnutzerliste.php* Skript arbeitet. Der Unterschied zur vorherigen Nutzerliste besteht darin, dass hier in der Zeile, in der der Stift angeklickt wurde, nun drei Felder vorhanden sind, in denen der **Nutzername**, das **Passwort** oder das **Recht** des jeweiligen Nutzers vom eingeloggten Admin **neu eingetragen** bzw. **verändert** werden können.

nutzername	kennwort	rechte
testuser	\$2y\$10\$Mr7aQsnGjefZ7d5dprCa8.tunNg7yCAAJLE.BnKNeRBTESSx/z2eu	nutzer
admin	kennwort zuweisen	admin

Nutzerzeile des Nutzers „admin“ wurde in den Bearbeitungsmodus versetzt

Nutzername und Kennwort sind beliebige Eingaben, im Rechte-Dropdown wird entweder „admin“ oder „nutzer“ ausgewählt.

4 Der Include Ordner

nutzer.txt

Die **Nutzerdaten** werden in einer Textdatei (*nutzer.txt*) gespeichert, die sich im include-Ordner befindet.

Für jeden Nutzer ist hier eine Zeile vorgesehen. In dieser Zeile befindet sich ein **Array**, in dem der **Name** [0], das gehashte, also **verschlüsselte Kennwort** [1] und das **Recht**, bzw. die Gruppe, also „admin“ oder „nutzer“ [2] eingetragen sind. Die Daten werden mit einem **Semikolon** als Feldtrennzeichen abgetrennt.

```
nutzer.txt
1  leo;$2y$10$ijMEc6ikLN1r9Lsx7uV8o.pChAM7Ke0vLFIZsBlxmg4tFWuCDMEq;admin
2  asf;$2y$10$Jgxn2zNbehMCyZtZGgQ3yerPVRyhxuZ8gPR0j3mZ6i2bqPbh5bJz6;nutzer
3  hans;$2y$10$5HzJ0KZMh07Cf1YAHKvHc0CSbYmvxawyakaixnWQRMm54CYGspB9q;nutzer
4  testuser;$2y$10$rp4ldeHnJZJQ1IJPYxU3Xe.R3SQWp6wGjyaJmJES4EjaJKTmbDpDq;nutzer|
```

nutzer.txt: nutzername ; verschlüsseltes kennwort ; gruppe

Diese Vorgehensweise der „Speicherung“ ist recht bequem und kann einfach in den anderen Skripten genutzt werden. Hierbei wird im Allgemeinen mit einer for-Schleife geschaut, welcher Nutzer gerade angemeldet ist und eine Aktion durchführen will. Dafür wird Zeile für Zeile abgeglichen, ob der Name des angemeldeten Nutzers (in der Session unter „name“ gespeichert) mit dem Eintrag im ersten Index des Arrays dieser Zeile übereinstimmt. Dann kann je nach Aktion geschaut werden, ob zum Beispiel das in einem Feld eingegebene Kennwort mit der zweiten Stelle im Array, also dem hinterlegten Kennwort, übereinstimmt, usw...

Das Format der Daten spielt eine wichtige Rolle, die jeweiligen Eingaben müssen **exakt** mit den gespeicherten Daten übereinstimmen. Klein-/Großschreibung sind also wichtig. Ein Nutzer kann diverse Zeichen verwenden, egal ob zum Beispiel mit Leerzeichen zwischen Buchstaben oder Großschreibung mitten im Wort, solange er es eben immer genauso eingibt. Dies hat den Vorteil, dass man auch seinen Nutzernamen schon spezieller festlegen kann, sodass auch dieser nicht ganz so einfach von Außenstehenden erraten werden kann.

inc_htmlhead.php

Dies ist der universelle htmlhead (nicht zu verwechseln mit unserer inc_htmlkopfzeile.php), der auf allen grafischen Seiten eingebunden wurde. Hier werden Seitentitel, das Favicon und die CSS-Datei (*design.css*) integriert, die Schriftart eingebunden und uft-8 konfiguriert.

.htaccess

Diese Konfigurationsdatei verhindert den Zugriff auf den include-Ordner, sollte jemand im Browser versuchen, den Ordner über die Suchleiste direkt aufzurufen. (localhost/include)