

2019

Projekt: Carsharing



CARSHARING

NILS RIEGER, REMO WINKLER

Inhaltsverzeichnis

1	Einleitung.....	2
2	Anforderungsanalyse	2
3	Sicherheitsaspekte.....	2
4	Verwendete Programmiersprachen	3
5	Wichtige Funktionen.....	3
5.1	Registrierung	3
5.2	Anmeldung	4
5.3	Karte	5
6	Testfälle und Lösungen	6
7	Fazit.....	7

1 Einleitung

Diese Dokumentation zeigt auf, was Remo und Ich im Modul 133 bei Herrn Dennler zu tun hatten. Im Modul 133 ging es darum eine Webanwendung zu realisieren. Remo und Ich habe das Projekt Carsharing zugeteilt bekommen. Das Ziel dieses Projektes war es den Softwareentwicklungsprozess zu behandeln.

Die Webanwendung soll ein Portal werden um einfach und schnell Autos zu mieten, man soll auf einer Karte sehen können, wo sich die Autos befinden. Natürlich sieht man auch Details zu den jeweiligen Fahrzeugen. Ein weiterer wichtiger Aspekt ist die Verfügbarkeit, man muss natürlich sehen zu welchem Datum das Fahrzeug verfügbar ist oder ob dieses Fahrzeug dann bereits vermietet ist.

2 Anforderungsanalyse

Bei unserer Webanwendung sind folgende Anforderungen notwendig. Der Kunde möchte auf der Website nach Fahrzeugen suchen können. Nachdem er nach einem Fahrzeug gesucht hat möchte er gerne die Informationen über das jeweilige Fahrzeug nachlesen. Ausserdem sollte der Kunde sehen, ob das Fahrzeug verfügbar ist oder nicht. Nicht nur die Verfügbarkeit der Fahrzeuge soll angezeigt werden, sondern auch gleich der Standort des zu vermietenden Autos. Dies wird auf einer normalen Karte mit Standortpunkten abgebildet. Wenn sich der Kunde für die Mietung eines Fahrzeuges entscheidet, muss sich der Benutzer registrieren, ist der Benutzer bereits registriert gibt es eine normale Anmeldefunktion.

3 Sicherheitsaspekte

Die Sicherheit unsere Webanwendung ist ein unabdingbarer Aspekt der Entwicklung. Ein wichtiger Sicherheitsaspekt ist, dass keine falschen Daten vom Kunden bei der Registrierung eingegeben werden dürfen. Dies muss man abfangen und den Kunden auf die korrekte Eingabe hinweisen, ansonsten haben wir in unserem System verfälschte Daten die wir nicht verwenden können.

Ein weiterer wichtiger Punkt sind Angriffe. Es gibt unzählige Möglichkeiten Schwachstellen in einer Webanwendung zu finden und diese auszunutzen. Um sich davor zu schützen gibt es einige Massnahmen. Zum einen müssen alle Eingabedaten von der Anwendung auf dem Server auf Zulässigkeit überprüft werden. Das Gleiche gilt auch für Namens- und Formularvariablen, diese können auch beliebig manipuliert werden. Deshalb ist es besser auch eine Überprüfung der Zeichenketten zu machen. Noch ein weiterer Punkt ist die Länge, beziehungsweise die Grösse der Eingabedaten zu begrenzen.

4 Verwendete Programmiersprachen

Für unsere Webanwendung verwendeten wir hauptsächlich zwei Programmiersprachen, dass wären zum einen HTML und PHP, es kam jedoch auch JavaScript zum Einsatz.

Die Grundfunktionen wurden mit PHP oder HTML realisiert, das reichte auch aus bis auf die Kartenfunktion. Die Kartenfunktion mussten wir mithilfe von JavaScript realisieren.

Den Style der Seite haben wir ganz schlicht und einfach gehalten. Damit dies auch einfach für die Bearbeitung und allfällige Erweiterungen ist, haben wir mit unserem CSS alle Seiten ähnlich gehalten.

5 Wichtige Funktionen

5.1 Registrierung

Bei der Registrierung haben wir zuerst überprüft ob alle Felder ausgefüllt sind. Danach haben wir geprüft ob die Benutzereingaben auch den Anforderungen entsprechen. Anschließend haben wir alle Eingaben des Benutzers in Variablen gespeichert. Das Passwort wurde dann noch verschlüsselt damit es bei Sniffing Attacken nicht im Klartext sichtbar ist. Danach wird der SQL Befehl ausgeführt, tritt kein Fehler auf wird der Benutzer in die Datenbank eingetragen.

```
<?php
error_reporting(!E_ALL);
include '../scripts/connection.php';

if($_POST['pw'] == $_POST['pwl'])
{
    if(isset($_POST['registrieren'])) //prüfen ob Registrierung geklickt wurde
    {
        $vn = $_POST['vn']; //Vorname in Variabel Speichern
        $nn = $_POST['nn']; //Nachname in Variabel Speichern
        $tel = $_POST['tel']; //telefonnummer in Variabel Speichern
        $email = $_POST['email']; //Email in Variabel Speichern
        $bn = $_POST['bn']; //benutzername in variable speichern
        $pw = $_POST['pw']; //passwort in variable speichern

        $pwmd5 = md5($pw); //passwort md5 verschlüsseln

        $sql = "INSERT INTO `benutzer` (`Vorname`, `Nachname`, `Telefonnummer`, `Email`,

        $db->query($sql); //sql befehl ausführen

        $sql2 = "SELECT ID FROM benutzer WHERE Benutzername='$bn' && Passwort='$pwmd5'";

        $result = $db->query($sql2); //Datenbankabfrage um ID herauszufinden ausführen

        $dsatz = mysqli_fetch_assoc($result); //array aus abfrage bilden

        $bnid=md5($dsatz['ID']); //User ID in md5 verschlüsseln

        $ang = md5('NeinNeinNein'); //Parameter für Angemeldet

        header('Location: index.php?id='.$ang.'&lg='.$bnid.'');

    }
}
else
{
    echo 'Passwörter stimmen nicht überein, bitte überprüfen Sie Ihre Eingabe.';
}
?>
```

5.2 Anmeldung

Die Anmeldefunktion ist um sich anzumelden. Bei der Folgenden Funktion wird zuerst überprüft ob man auch wirklich auf Anmelden geklickt hat. Anschliessend werden der Benutzername und das Passwort in Variablen gespeichert, darauf wird eine Abfrage getätigt ob diese Benutzerdaten korrekt sind oder nicht. Auch wird darauf geachtet, dass keine SQL Codes in die SQL Abfrage mithilfe Textfelder geschrieben werden können. Sind die Benutzerdaten dann korrekt wird der Benutzer angemeldet. Sollte die Abfrage nicht korrekt sein wird der Benutzer dazu aufgefordert, seine Eingaben nochmals zu überprüfen. Anschliessend wird wieder eine Abfrage ausgeführt um zu überprüfen ob die Anmeldedaten nun korrekt sind.

```
<?php

include '../scripts/connection.php';

if(isset($_POST['anmelden'])) //prüfen ob Anmelden geklickt wurde
{
    $benutzer = $_POST['bn']; //benutzername in variable speichern
    $Passwort = $_POST['pw']; //passwort in variable speichern

    $Passwortmd5 = md5($Passwort); //passwort md5 verschlüsseln

    //-----Abfrage ob Benutzerdaten korrekt sind-----//

    $sql = "SELECT ID FROM benutzer WHERE Benutzername='$benutzer' && Passwort='$Passwortmd5'";

    $result = $db->query($sql); //Datenbankabfrage ausführen

    $dsatz = mysqli_fetch_assoc($result); //array aus abfrage bilden

    $bnid=md5($dsatz['ID']); //User ID in md5 verschlüsseln

    $sang = md5('NeinNeinNein'); //Parameter für Angemeldet

    if(!$result)
    {
        echo "Anmeldung Fehlgelassen. Bitte überprüfen sie Ihre eingaben und Versuchen sie es erneut";
    }
    else
    {
        header('Location: index.php?id='.$sang.'&lg='.$bnid.'');
    }
    //-----Ende Abfrage ob Benutzerdaten korrekt sind-----//
}

?>
```

5.3 Karte

```
<script type="text/javascript">
    function initialize() {
        if (GBrowserIsCompatible()) {
            var map = new GMap2(document.getElementById("map-container"));

            map.setCenter(new GLatLng(47.22309, 8.31395), 13, G_HYBRID_MAP);

            function createMarker(point, html) {
                var marker = new GMarker(point);

                GEvent.addListener(marker, 'click', function() {
                    marker.openInfoWindowHtml(html);
                });
                return marker;
            }

            var point = new GLatLng(51.21412, 6.77418);
            var marker = createMarker(point, 'Hier wohne ich');
            map.addOverlay(marker);

            map.addControl(new GLargeMapControl());

            // Kartentypen einblenden
            map.addControl(new GMapTypeControl());

            // Übersichtskarte einblenden
            map.addControl(new GOverviewMapControl());

            // Massstab einblenden
            map.addControl(new GScaleControl());

        }
    }
</script>
```

6 Testfälle und Lösungen

Testfall	Soll-Erwartung	Ist-Erwartung	Massnahme
User registrieren	User wird registriert und kann sich Anmelden	User wird erfasst und in DB geschrieben	-
Fahrzeugdetails anzeigen	Man sieht die Angaben zum Fahrzeug übersichtlich	Man sieht ein Bild des Fahrzeugs und die Angaben dazu	-
User löschen	User wird aus der Datenbank gelöscht und kann sich nicht wieder Anmelden	User wird von DB gelöscht und kann sich nicht mehr Anmelden	-
Auto auf Karte anzeigen	Man sieht wo sich das Auto befindet	Man sieht die Autos auf der Karte	-
Auto reservieren	Das Auto wird für das gewählte Datum nicht mehr angezeigt	Man kann das Auto reservieren	-
User kann keine falschen eingaben in die Reservierung schreiben	Geht nicht	Geht nicht	-

7 Fazit

Nils: Mein Persönliches Fazit zum Projekt fällt sehr positiv aus. Nachdem wir uns mit der Planung der Arbeit zu viel Zeit gelassen haben und gegen Ende unter Druck geraten sind, haben wir gemerkt, dass bei so einem grossen Projekt nicht alles auf den letzten Drücker gemacht werden kann. Bei der Entwicklung der Webanwendung hatten wir einige kleine Problem die wir aber immer lösen konnten mithilfe des Internets. Für ein allfällig nächstes Projekt würde ich die Arbeit viel genauer Unterteilen. So kann man Woche für Woche die Arbeit abarbeiten und kommt am Ende nicht unter Druck.

Remo: Mein Fazit ist auch sehr positiv. Jedoch ist mir auch aufgefallen das wir das Projekt am Anfang ein wenig unterschätzt haben. So sind wir bei der Programmierung ziemlich unter Zeitdruck gefallen. Auch sind wir auf kleinere Schwierigkeiten gestossen die den Zeitplan noch mehr durcheinandergebracht haben. Für das nächste Projekt werden wir dann mehr Zeit in die Umsetzung investieren. Aber im Ganzen muss ich sagen, dass es sehr viel spass gemacht hat und wir viel daraus lernen konnten.