Business Anwendungen

Prüfungsleistung: Online Shop und Dokumentation

Prüfer: Dr. Christian Krauss
BA, WS 2018/2019, 5. SEMESTER

Melina Valtinke, 929464
Jonas Kahnwald, 929552
Fachhochschule Kiel

Inhaltsverzeichnis

Inhalt

1. Einleitung

1.1 Projektvorgaben

Das Projekt für das fünfte Semester für das Fach Businessanwendungen ist ein Online-Shop, welcher auf die Grundlagen von dem ersten bis zum vierten Semester aufbaut. Darunter befinden sich die IT Grundlagen, die Grundlagen der Webseitenerstellung sowie die dynamische Webseitenerstellung und Echtzeitanwendungen im Internet. Das Semesterprojekt legt den Schwerpunkt auf die funktionale Umsetzung und Programmierung der Anwendung, weniger um die Schönheit des Online-Shops. Eine CMS-Verwendung oder Frameworks sind dabei nicht erlaubt, lediglich HTML, CSS, php, MySQL und JavaScript. Dennoch sind jQuery, Bootstrap und Font Awesome gestattet.

Neben den grundlegenden Vorgaben des Online-Shops, soll dieser bestimmte Funktionen beinhalten. Diese werden folglich aufgelistet und passend zu dem Online-Shop erläutert und getestet.

2. Einzelne Entwicklungsschritte

2.1 Ideenfindung

Um einen Online-Shop zu erstellen, ohne auf variable Versandkosten zu stoßen, wurde ein Thema gewählt, welches die Produkte per Email versendet. Diese Orientierung war bereits von Beginn an da.

Nach intensivem Brainstorming wurde entschieden einen Online-Shop für Gaming Spiele zu erstellen. Der Schwerpunkt liegt dabei auf die Schlüssel (Keys), die dem User zugeschickt werden und nicht der Download des Spiels. Deshalb wurde der Name *locksmith*, Deutsch Schlosser/Schlüsseldienst, gewählt.

2.2 Ziele und Zielgruppen

Das Zielgruppenspektrum ist sehr breit gefächert. Die angebotenen Spiele reichen von Abenteuerspielen über Landwirtschafts-Simulatoren bis zu Rollenspielen. Heutzutage ist das Klischeebild von einem „*Zocker*“ längst zerbrochen, die Interessen an der Technologie ist von Kind bis zum Erwachsenen sehr weit fortgeschritten und damit auch das Interesse an Spielen auf dem

Computer. Deshalb kann eine bestimmte Zielgruppe, wie beispielsweise bei Onlineshops für Kleidung, schwierig bestimmt werden. Auch die angebotenen Spiele sollen gleich von der ersten Seite aufgelistet werden, da die User kein großes Drumherum und Schnick-Schnack auf einer Gaming-Seite brauchen. Dies zeichnet sich auch auf aktuellen Gaming-Shop-Seiten wider.
Weitere Ziele sind, einen simplen und übersichtlichen Shop mit sehr guter Sicherheit zu erstellen. Da diese Keys das Kernstück und das damit wertvollste am Shop sind, wird die Sicherheit großgeschrieben. Außerdem sollen Vorkehrungen getroffen werden, um Daten wie Passwörter oder IBAN sowie allgemein personenbezogene Daten zu sichern.

2.3 Designrecherche

Durch Eigenerfahrungen und Recherche auf weiteren Online-Shop-Seiten im Bereich der Computerspiele, ist ein schlichtes, dunkles Design aufgefallen. Gerne sind diese Seiten auch sehr überladen von Videos, Bildern, verschiedenen Schriftarten und Schriftgrößen. Ein solches Design bringt viel Unruhe. Leider ist dies schwer zu vermeiden, da diese Seiten nun mal aus interaktiver Werbung für die jeweils angebotenen Spiele bestehen.
Es wurde entschieden mit Bootstrap zu arbeiten. Dies war erlaubt und vereinfachte viele Anwendungen. Da die eigene Erfahrung mit Bootstrap fehlte, musste dies zuerst nachgeholt werden. Dafür wurde viel auf „*Youtube*“ und der eigenen *Bootstrap.org*-Seite recherchiert. Nach ein wenig Stunden konnte das Layout entwickelt werden.

2.4 Designentscheidungen

Der komplette weiß hinterlegte Hauptbereich befindet sich auf der Index-Seite. Das heißt, es werden keine neuen Seiten aufgerufen. Ein dunkles gefixtes, nicht wiederholendes Hintergrundbild mit wenig, aber starken Farben rundet die Seite ab. Zentral gesetzt befindet sich der Hauptbereich. inklusive aller angeforderten Funktionen. Farbe und Schrift sind neutral und nicht aufdringlich. Nur die Farben weiß(*#ffffff*), schwarz(*#000000*) und blau (*#007bff*) wurden gewählt. Die Schriftart ist „*Dosis, light, sans-serif*“ mit einer Schriftgröße von *15pt*. Die Headerschriften, also Überschriften sind *h2* und hat die Schriftgröße *23pt*. Die Überschrift *h3* besitzt die Schriftgröße vom Elternelement. Die Buttons ziehen bezüglich des gleichen Designs einen roten Faden durch die Seite. Da durch Bootstrap schnell ein einheitliches Bild entsteht und sich die Webseite wenig von anderen Seiten, die mit Bootstrap erstellt wurden, abhebt, wurden selbstständig in der CSS-Style Datei einige Änderungen vorgenommen.

Wie oben schon angesprochen, sind die Online-Games Seiten, wie beispielsweise Steam ziemlich aufwühlend. Verschiedene Bilder konnten wir nicht vermeiden, da jedes Spiel sein eignes Werbebild mit Titel und Preis haben soll. Dies wurde mit der Bootstrap Klasse *card* bestimmt. Damit die Spieleauflistung nebeneinander abläuft, ist der Style *float: left* mit einer festen Breite dazugekommen. Die feste Breite bestimmt nicht nur den Aufbau der *cards*, sondern auch den, der Seitenzahlen mit der Blätterfunktion. Diese besitzen ebenfalls ein *float: left*. Sobald keine feste Breite angegeben wird und ein responsives Design einleitet, verschieben sich die Seitenzahlen neben den Produkten. Deshalb wurde vorerst auf ein responsives Design verzichtet, da dies keine Pflichtaufgabe war.

2.5 Programmablaufplan Frontend

2.6 Programmablaufplan Backend

2.7 Benutzerdokumentation Frontend

Auf der Startseite wird man persönlich willkommen geheißen, wenn die Seite bereits besucht und ein eigenes Konto erstellt wurde. Als Erstbesucher gibt es die Möglichkeit sich ein neues Kundenkonto zu erstellen. Unmittelbar daneben wird der Warenkorb mit der Anzahl der im Warenkorb bestehenden Artikel angezeigt. Direkt unter dieser Funktion werden die Spiele aufgelistet. Am Ende befindet sich die Seitenzahlen mit der vorgegebenen Blätterfunktion. Falls der User sich noch nicht registriert hat, kann dieser keine Spiele kaufen. Die Möglichkeit den Warenkorb mit Artikeln zu füllen besteht trotzdem. Nach der Registrierung bleiben die gewählten Artikel im Warenkorb gespeichert. Wenn ein User sich registriert, muss dieser nach der Registration sich dennoch anmelden. Dafür wurde eine kleine Meldung eingebaut, sodass der User darauf aufmerksam gemacht wird.
Falls bei der Registrierung falsche Daten eingetragen wurden, können diese in Mein Bereich geändert werden.
Die Produkte sind direkt auf der Startseite abgebildet. Beim Klicken auf den Button mit dem Preis, wird der User zu der Einzelansicht des Produktes weitergeleitet. Dort wird das Spiel detailliert beschrieben, die vorhandene Anzahl des Artikels angegeben und in den Warenkorb gelegt werden.

Sobald der User mit dem Einkauf fertig ist, kann dieser über den Link „Warenkorb“ oben rechts sich in Richtung Bezahlvorgang bewegen.
Bevor der User die Bezahlmöglichkeit auswählt, werden alle im Warenkorb enthaltenen Artikel aufgelistet. Diese Artikel können nochmal bearbeitet, das heißt erhöht oder gelöscht werden. Rechts über den Button „Kasse“ gelingt der User zum Kaufabschluss. Auf dieser Seite sind die verschiedenen Zahlungsmethoden auswählbar, die ausgewählten Spiele mit der Gesamtsumme und die Akzeptierung der AGB´s. Nur, wenn diese akzeptiert werden, erscheint der Button „Jetzt zahlungspflichtig bestellen!“. Sobald dieser gedrückt wurde, erscheint eine kleine Bestätigung mit der Information zur Bestellung. Nach der Bestellung wird der Warenkorb wieder geleert. Auf der Seite „Mein Bereich“ können die vergangenen Bestellungen gesichtet werden. Das Ausdrucken dieser Informationen ist nicht möglich. Alle Bestellungs- und Rechnungsdaten werden dem User per Email zugesendet und nicht auf der Shop-Seite hinterlegt. Im *footer* befinden sich außerdem die Datenschutzerklärung und das Impressum. Der Admin hat neben den gleichen Funktionen wie ein User, die Möglichkeiten Artikel hinzuzufügen, diese zu bearbeiten oder zu löschen. Auch unter „*Mein Bereich*“ kann dieser alle Bestellungen von jedem User (*Nutzer_ID*) ansehen, wie auch seine Daten korrigieren. Am Ende der Startseite und der „Mein Bereich“ Seite befindet sich der Button zum Adminbereich. In dem Adminbereich können Spiele zugefügt, Artikel bearbeitet oder auch gelöscht werden. Jeder einzelne Artikel hat seinen eigenen Button für die Speicherung und die Löschung.

2.8 Benutzerdokumentation Backend

In dem *locksmith* Ordner befindet sich der komplette Online-Shop. Zuerst kommt der Ordner mit der Dokumentation. Der ist für den Shop nicht relevant. Der Ordner *webseite* dagegen sehr. Weiterhin beinhaltet ist die *index.php*, LICENS, die *locksmith.sql* Datenbank, die shop.css Datei und die shop.js Datei. In der *index.php* Datei spielt sich der komplette Shop ab. Diese Seite ist das Kernstück von *locksmith*. Eingebunden wird hier der Aufbau, welcher das Grundgerüst bildet und die JavaScript Datei. In der shop.css Datei befindet sich des gesamten Stylesheets, also das Design und Layout von *locksmith*. In dem *webseite* Ordner befindet sich der Ordner aussehen und der Ordner *modules*. Außerdem steht dort die *.htaccess* Datei und die *config.php* Datei. In der *config.php* wird die Datenbank eingebunden. Da bei der Arbeit mit dem Online Shop XAMPP benutzt wurde, muss der Benutzer auf „root“ und das Passwort auf „“ gesetzt sein. Im aussehen Ordner befindet sich der Unterordner vorlagen. In diesem sind die Schriftarten (*fonts* Ordner), das

HTML Gerüst (*html* Ordner) und der Images Ordner. In dem *html* Ordner befindet sich die *footer.php*, die *header.php*, *html_foot.php* und die *html_head.php* Dateien. Diese Dateien sind relevant für die im vorlagen Ordner befindende *aufbau.php* Datei. Dort werden alle Dateien inklusive der *controller.php* Datei eingebunden und ergeben im Endeffekt die fertige Shop Seite. Im Ordner *modules* wurde der *model view controller* angelegt. Dieser wurde jedoch nicht, wie üblicherweise benutzt. Im Ordner *controller* befindet sich die *controller.php* Datei, welche alle MySQL Befehle beinhaltet und somit das Gehirn des Online-Shop *locksmith* ist. Im *model* wurde keine Datei angelegt, diese befinden sich mit in der *controller.php*. Im *view* Ordner befinden sich alle sichtbaren Elemente für die Shop-Seite. Die einzelnen Seiten, Buttons, Formulare wie auch die Rest- und Seitenanzahl. Dort haben diese ihre Klassen und HTML tags und das damit verbundene Layout zugewiesen bekommen.

3. Ergebnisse

3.1 Testplan Frontend

Um sicher zu sein, ob die Pflichtanforderungen erfüllt wurden und dass der Shop für die Benutzer anwenderfreundlich und übersichtlich ist, wird ein Test durchgeführt. Zu Beginn mit einem User. Dieser soll sich Produkte bestellen und damit die Funktionalität und das Design bewerten. Dies wird in verschiedenen Browsertypen ausprobiert, Chrome und Firefox.

Zudem soll getestet werden, ob man ein Produkt ohne Anmeldung bestellen kann, die User Datenänderungen, die Anmeldefunktion bei richtigen und falschen Anmeldedaten, der Kassenzugang, der Userbereich mit den letzten Bestellungen, die richte Anzahl des Artikelbestandes sowie eine Kaufmöglichkeit erst nach Zustimmung der AGB´s.

Im Adminbereich wird der Artikelbereich genauer unter die Lupe genommen. Neben dem Hinzufügen von neuen Artikeln soll auch das Bearbeiten von den einzelnen Artikeln ermöglicht werden, wie auch eine Ansicht aller Bestellungen von jedem User.

Außerdem soll die Sicherheit getestet werden. Die erfolgt durch Versuchen, das Passwortfeld bei der Anmeldung aus zu kommentieren oder auch ein HTML Tag in eins der Formularfelder von der Nutzer- oder Artikeltabelle hinzuzufügen.

3.2 Testergebnisse Frontend

Die Shop-Seite funktioniert nur in dem Chrome Browser, bei Firefox werden nicht alle Designs geladen. Der Testuser empfindet die Seite als Übersichtlich und kann durch die Produktauflistung erahnen, worum es sich bei diesem Shop handelt. Auch die Buttons auf den Seiten haben sich von

selbst erklärt. Durch den *Hover*-Effekt wird drauf hingewiesen, dass durch eine Klickaktion etwas passiert. Da jeder Button logisch beschriftet ist, weiß der Testuser auch wohin er geleitet wird. Auf der Einzelartikelseite sind immer nur bestimmte Mengen des Produktes vorhanden. Der User kann zwar eine höhere Zahl eintippen, doch bekommt nach betätigen des Warenkorb Buttons eine Fehlermeldung. Bei Nichtanmeldung, gelingt es dem User Produkte in den Warenkorb zu legen und auch zur Kasse zu gehen, aber nicht den Abschließenden Kaufbutton zu betätigen. Dieser wird erst angezeigt, wenn der User registriert und eingeloggt ist. Eine Meldung gibt dies ebenfalls zur Kenntnis. Sobald der User eingeloggt ist, bleibt auch das in Warenkorb gelegte Spiel vorhanden. Das Passwort konnte durch verschiedene Anwendungen auch nicht auskommentiert werden. Dazu kommt auch die Fehlermeldung, dass die Emailadresse oder das Passwort nicht korrekt sind. Dies erschwert dem Hacker zu wissen, ob er nun die richtige Email oder das richtige Passwort oder gar nichts richtig angewendet hat. Bei einer Registrierung nützt es dem User nicht, HTML oder SQL Codes in die Formulare zu setzten. Durch das Absenden in Formularen kann also z.B. das <script>-Tag versendet werden, welches dann auch im Browser ausgeführt werden kann. Dadurch kann ein Angreifer auf Informationen zugreifen, die ihm sonst nicht zustehen, oder den HTML-Code der Seite verändern. Dies kann durch eine Maskierung der Eingaben, welche ein Nutzer abschicken kann (in diesem Fall z.B. ein Postinhalt etc.), verhindert werden. PHP liefert von Haus aus die Funktion htmlspecialchars() oder strip_tags() mit. Durch beide wird der abgesendete String unfähig, <script>-Tags in einem ausführbaren Zustand zu verschicken.

In dem Kassenvorgang ist es nicht möglich die Anzahl des Artikels über die bestehende Anzahl zu setzten. Bei der Kasse angekommen, wird der Testuser auf die Zahlungsmethoden aufmerksam gemacht. Dabei ist ein Fehler aufgefallen. Die Kreditkarte ist zwar ausgewählt, aber die Felder sind nicht auf *required* gesetzt. Wenn diese als Pflichtfelder gesetzt werden, müssen diese bei den anderen Bezahlmethoden ausgeblendet sein. Dies wurde daraufhin umgesetzt und hat die gleiche Funktion wie dem darunter stehenden Einverständnis der AGB´s. Erst nach setzten des Hackens wird der „Jetzt zahlungspflichtig bestellen“ Button eingeblendet. Nach Kaufabschluss erscheint ein kleiner Informationstext. Bei *locksmith* bekommt man die Rechnung, sowie den Key und den Link zum Download des Spiels verschlüsselt per Email zugeschickt. Deshalb kann der User direkt auf unserer Seite keine Rechnung ausdrucken. Bis zu dieser Seite stehen noch die Artikel im Warenkorb, sobald der User aber auf den Warenkorb oder zu anderen Seiten klickt, wird dieser wieder auf Null resettet. Unter „Mein Bereich“ werden alle getätigten Bestellungen des Users und die persönlichen Daten angezeigt. Alle Felder, bis auf das Anrede Feld, können bearbeitet und abgespeichert werden. Das Anrede Feld ist ein *selec*t Befehl und beinhaltet das von der

Registrierung *value*. Durch rechtsklick und untersuchen wird dies auch bestätigt. Leider wird dennoch, die erste *option* angezeigt. Diesen Fehler konnten wir nicht beheben.
Der Admin erhält nach Anmeldung den Zugriff auf den Adminbereich. Zu diesem gelingt man durch betätigen des unten rechtsstehenden Buttons. In den Adminbereich können Artikel hinzugefügt oder bearbeitet werden. In diesen Zeilen ist es wieder nicht möglich Zugriff auf den Code zu bekommen, durch auskommentieren oder setzten von HTML tags oder SQL Codes. Leider sind bei der Bearbeitung von Artikeln zwei Fehler aufgefallen. Es sollten in den Formularen alle Felder durch das *value* belegt sein. Doch bei den Bildern funktioniert dies nicht. Die in der Datenbank gespeicherten Bilder werden dort nicht gesetzt. Es wird in der Rubrik Titelbild lediglich kein Bild ausgewählt und sobald der Admin nur den Namen des Spiels ändert, wird die bestehende Bilddatei mit einer leeren Datei, also nicht ausgewählten Datei, ersetzt. Deshalb muss der Admin bei jeder Bearbeitung das passende Bild hochladen. Ein weiterer kleiner Fehler ist, dass die *textarea* den beinhalteten Text mit zwei bis drei Absätzen nach unten verschiebt und zentriert. Auf der Einzelartikelansicht wird dies jedoch nicht verschoben mit Absätzen angezeigt, sondern linksbündig und direkt unter dem Bild. Sobald ein Spiel bearbeitet wurde, muss dies vom Admin direkt vom dazugehörigen untenstehenden Button abgespeichert werden. Falls der Admin mehrere Spiele bearbeitet und nur den letzten Speichern Button betätigt, gehen die vor dem letzten Spiel bearbeiteten Datensätze verloren und alles bleibt beim Alten. Auf der Mein Bereich Seite vom Admin werden alle Bestellungen von jedem Nutzer angezeigt und ein Formular für die Bearbeitung der Admin Daten.

3.3 Testplan Backend

Im Backend findet der Test statt, ob die Daten wie Passwörter und IBAN Verschlüsselt in der Datenbank abgespeichert werden und ein sicheres Log-In gewährleistet wird. Zudem kommt der Versuch einzelne SQL-Abfragen einzuschleusen.

3.4 Testergebnisse Backend

Für eine sichere Webseite wurden ein paar Vorkehrungen getroffen, damit *locksmith* gegen Dinge wie *SQL-Injection* geschützt ist. Auch beim Login wurde der heutige Standard der Passwortverschlüsselung genutzt.
Für alle SQL-Abfragen in der Anwendung wurde die *PHP-Data-Obejcts-Erweiterung*, kurz PDO, benutzt. Dies ist eine konsistente Schnittstelle, um mit PHP auf die *locksmith.sql*-Datenbank zuzugreifen. Durch das vorbereiten der einzelnen SQL-Abfragen mit Platzhaltern anstelle von

Variablen mit anschließender Parameterbindung kann das Einschleusen von nicht gewolltem SQL-Code verhindert werden. Bei einer *SQL-Injection* kann eine bereits geschriebene SQL-Abfrage durch Nutzereingaben erweitert werden und so dem Nutzer Datenbankinformationen geben, die nicht für ihn bestimmt sind. Durch das soeben genannte Verfahren wird eben das verhindert und SQL-Code, der durch Formularfelder durch Angreifer übermittelt werden könnte, wird nicht mehr als solcher behandelt. Die Login-Funktion besitzen vorgefertigte PHP-Funktionen wie *passsword_hash()* und *password_verify().* Durch erstere wird der durch den Nutzer eingegebene Passwort-String ausreichend gut verschlüsselt und anschließend in der Datenbank gespeichert. Das heißt der User kirsten@lol.web mit dem Passwort 123456 hat ein 30-stelliges verschlüsseltes Passwort bestehend aus Sonderzeichen, Zahlen, Groß- und Kleinbuchstaben. Es beginnt mit *$2y* und endet mit *iiu*.
Dabei werden identische Passwörter jedes Mal durch die Zugabe eines zufällig generierten SALT-Wertes und der anschließenden md5-Verschlüsselung anders gespeichert, sodass auch bei einem Datenbank-Leak die Passwörter der Nutzer verschlüsselt bleiben.
Damit ein sicherer Login gewährleistet werden kann, wird bei jedem Login eine neue Session-ID erstellt, welche in der Form eines Cookies auf Client-Seite abgespeichert wird. Diese wird zufällig generiert und anschließend in der Datenbank gespeichert. Zugehörig ist immer die Information, ob der Nutzer, der die Session-ID verwendet, eingeloggt ist. Anhand dieser Information kann ich zu jedem Zeitpunkt wissen, ob ein Nutzer eingeloggt ist oder nicht und ein Angreifer kann mit der Session-ID nichts anfangen, da sie sich dauernd ändert.

4. Anwendungen

4.1 Installationsanleitung

Um den Shop aufrufen zu können, wird XAMPP benötigt. Als erstes wird der *locksmith* Ordner aus dem Onlineshop2 Ordner in den *htdocs*-Ordner von XAMPP verschoben. XAMPP wird gestartet, danach *localhost/phpmyadmin* im Browser öffnen und eine neue Datenbank namens *locksmith* anlegen. Nun die *locksmith.sql* aus dem Onlineshop Ordner Datei importieren. Alles speichern und XAMPP erneut starten. Der Onlineshop *locksmith* kann im Browser aufgerufen und verwendet werden.

4.2 Zugriffsorganisation

Die Admin Anmeldedaten lauten:

Email: admin@lol.de

Passwort: 123

Die User Anmeldedaten lauten:

Email: kirsten@web.de

Passwort: 123456

5. Quellen

Spielinformationen:

https://store.steampowered.com/search/?filter=topsellers

6. Bildquellen

Bilder der zugehörigen Spiele:

https://store.steampowered.com/search/?filter=topsellers