

# Dokumentation



**Teammitglied: Philipp Stöhr**

**Rollen:** Scrum Master

Schriftführer

**Aufgaben:** Protokoll erstellen

Use-Case-Diagramm entwerfen

Logo designen & animieren

Login implementieren

- Registrierung
- Login / Logout
- Passwort zurücksetzen

## **Gliederung**

0.	Gliederung .....	2
1.	Einführung ins Projekt.....	3
1.1.	Projektüberblick.....	3
1.2.	Projektorganisation .....	4
2.	Rolle des Scrum Masters im Projekt .....	5
2.1.	Aufgaben eines Scrum Masters & Umsetzung im Projekt.....	5
2.2.	Entwurf einer Protokoll-Vorlage .....	6
3.	Entwurf des Use-Case-Diagramms „Build A Quiz“ .....	7
3.1.	Modellierung von Use-Case-Diagrammen .....	7
3.2.	Modellierung des Use-Cases „Build A Quiz“ .....	8
4.	Erstellen eines Logos für „Build A Quiz“ .....	9
4.1.	Design des Logos.....	9
4.2.	Animation des Logos .....	10
5.	Implementierung der Login-Möglichkeit .....	11
5.1.	Anforderungsanalyse.....	11
5.2.	Implementierung der Registrierungs-Seite.....	12
5.3.	Implementierung des Logins und Logouts .....	14
5.4.	Implementierung der Änderung des Passworts .....	16
6.	Anhang .....	17
6.1.	Abbildungsverzeichnis.....	17
6.2.	Protokoll-Vorlage.....	18

## **1. Einführung ins Projekt**

Das Projekt wurde im Rahmen des Praktikums Software Entwicklung zur Vorlesung Rechnernetze I entwickelt. Ziel dieses Projekts war es ein größeres Softwareprojekt erfolgreich im Team zu bearbeiten und somit einen Einblick in die professionelle Softwareentwicklung zu erhalten.

### **1.1. Projektüberblick**

Das Team verständigte sich darauf eine Webseite mit Datenbankanbindung zu erstellen. Hierfür sollten die Web-Technologien HTML, CSS und PHP genutzt werden. Das Thema der zu entwickelnden Webseite lautet „Build A Quiz“ (kurz: BAQ). Die Webseite soll die Möglichkeit bieten eigene Quiz zu erstellen und zu verwalten. Hierfür ist es notwendig eine Nutzerverwaltung zu implementieren. Außerdem sollen die Nutzer sich eine Bestenliste anzeigen lassen können. Abschließend ist es für eine Webseite wichtig, eine Kontaktmöglichkeit und ein Impressum zur Verfügung zu erstellen sowie über die Datenschutzrichtlinien bei der Verarbeitung personenbezogener Daten zu informieren. Entsprechend lassen sich die Aufgaben folgendermaßen zusammenfassen:

- Nutzerverwaltung implementieren:  
Möglichkeit zur Registrierung, Login und Logout
- Quiz Builder implementieren:  
Möglichkeit zum Erstellen von eigenen Quiz

- Lets Quiz implementieren:  
Möglichkeit zum Spielen der Quiz
- Bestenliste implementieren:  
Darstellung einer Rangliste
- Impressum und Datenschutzerklärung implementieren:  
Möglichkeit zur Kontaktaufnahme und zur Einsichtnahme  
in die Verarbeitung der personenbezogenen Daten

## **1.2. Projektorganisation**

Das Projekt wurde mittels Scrum, einer agilen Methode des Projektmanagements, umgesetzt. Der Vorteil hierbei liegt in der iterativen Vorgehensweise, wodurch man gut die Kontrolle behält und gegebenenfalls flexibel auf Änderungen reagieren kann. Im Rahmen von Scrum gibt es verschiedene Rollen, die im BAQ-Team folgendermaßen verteilt waren:

Business Owner & Product Owner: Kursleiter Stefan Müller

Scrum Master: Teammitglied Philipp Stöhr

Entwicklerteam: Theresa Alt, Peter Leonov, Daniel Rohrwild, Philipp Stöhr, Florian Vogl

Weiterhin gilt es einige relevante Scrum-Meetings zu planen und vom Team eigenständig zu organisieren. So wurde festgelegt, dass die Sprints 14 Tage umfassen und zu Beginn jeweils ein Sprint Planning abgehalten wird. Im Planning werden die anstehenden Aufgaben für den kommenden Sprint aufgeplant. Hierfür ist ein korrekt priorisiertes Backlog notwendig. Es wurden Daily Meetings zur Abstimmung der einzelnen Arbeiten zwischen den Teammitgliedern eingeführt.

## **2. Rolle des Scrum Masters im Projekt**

Im Rahmen des ersten Gruppentreffens wurde festgelegt, dass Philipp Stöhr als Scrum Master des Teams fungieren sollte. Der Scrum Master bedient zudem die Rolle des Schriftführers für die Protokollierung der Ergebnisse von Besprechungen.

### **2.1. Aufgaben eines Scrum Masters & Umsetzung im Projekt**

Der Scrum Master trägt die Verantwortung dafür, dass das Team möglichst effektiv und ungestört arbeiten kann. Er sorgt dafür, dass Hindernisse, sog. Impediments, beseitigt werden und eskaliert diese, wenn nötig, bis ins obere Management.

Weiterhin achtet der Scrum Master darauf, dass die Scrum Regeln eingehalten und verstanden werden und dass das Scrum-Team gut als Team zusammenarbeitet und funktioniert. Er moderiert sowohl die Meetings als auch weitere Treffen außerhalb der geplanten Meetings. Außerdem ist er verantwortlich für die Erstellung von Meeting-Protokollen, um die Ergebnisse schriftlich festzuhalten.

Ein gutes zwischenmenschliches Gespür und kommunikative Fähigkeiten sind als Scrum Master ein großer Vorteil, um die Mitglieder zusammenzuhalten. Indem die Zusammenarbeit optimiert wird, wird auch das durch das Team generierte Ergebnis optimiert.

Innerhalb des BAQ-Teams war die Rolle des Scrum Masters denkbar leicht umzusetzen, da das Team bereits bestens

aufeinander eingespielt war und gut zusammengearbeitet hat. Die Meetings wurden regelmäßig durchgeführt, wobei lediglich im Daily ab und zu darauf zu achten war, dass die Zeit eingehalten und das Meeting nicht zu sehr in die Länge gezogen wird. Insgesamt gab es kaum Störungen, sodass der Kursleiter dafür nicht hinzugezogen werden musste.

## **2.2. Entwurf einer Protokoll-Vorlage**

Um die Protokolle möglichst effizient und immer nach demselben Schema anzugehen, wurde zunächst eine Vorlage entworfen. Hierfür wurde recherchiert, welche Möglichkeiten der Protokollierung es gibt und wie man ein Protokollformular aufbauen kann.

Als wichtigste Eckdaten mussten zu Beginn Datum, Zeitpunkt und Ort der Besprechung festgehalten werden. Weiterhin sollte die nächste Besprechung angegeben werden, um eine lückenlose Darstellung möglichst einfach zu gestalten. Hierfür dient zudem die Protokoll-Nummer. Es empfiehlt sich weiterhin den Schriftführer des Protokolls aufzuführen, sodass klar ist, an wen man sich bei Fragen wenden kann.

Darauffolgend müssen alle Teilnehmer der Besprechung aufgeführt werden, um die Anwesenheit festzustellen. Um festzuhalten, dass alle Teilnehmer mit dem Protokoll einverstanden sind, ist es wichtig, dass alle Teilnehmer bestätigen das Protokoll gelesen und verstanden zu haben.

Anschließend geht es darum die inhaltlichen Aspekte zu erfassen. Dies wurde der Übersichtlichkeit wegen in Form einer Tabelle umgesetzt. Die Tabelle besteht aus den folgenden vier Spalten: Stichwort, Informationen, Wer?, Bis wann?. Dadurch kann man sehr schnell erfassen, wer welche Aufgabe bis zu welchem Zeitpunkt zu erledigen hat.

Die Protokoll-Vorlage befindet sich im Anhang dieser Dokumentation und wurde bei jeder Besprechung genutzt.

### **3. Entwurf des Use-Case-Diagramms „Build A Quiz“**

Bevor mit der tatsächlichen Implementierung einer Software begonnen werden kann, ist eine objektorientierte Analyse notwendig. Dazu gehört unter anderem die Modellierung von Use-Case-Diagrammen, die einen bestimmten Vorgang im System darstellen.

#### **3.1. Modellierung von Use-Case-Diagrammen**

Use-Case-Diagramme haben als Ziel, die Anforderungen an die geplante Software zu definieren. Ein Use-Case beschreibt dabei den Ablauf zwischen Mensch und Maschine aus Sicht des Benutzers. Es sollen dabei nicht die inneren Abläufe des Use-Cases beschrieben werden, sondern die Funktionalitäten für den Nutzer. Für die Darstellung von Use-Cases gibt es eine bestimmte Syntax. Die für den hier aufgeführten Use-Case notwendigen Grundlagen, werden im Folgenden kurz aufgeführt:

Akteure sind die Anwender des Systems und werden als Strichmännchen dargestellt. Das System ist die Software, welches die Form eines Rechtecks hat und die wesentlichen Anforderungen enthält. Mittels einer durchgezogenen Linie vom Akteur löst dieser einen Anwendungsfall aus, wobei die Tätigkeit als Beschriftung eingefügt wird. Beziehungen zwischen Use-Cases werden durch gestrichelte Pfeile mit der Beschriftung <<include>> dargestellt.

### 3.2. Modellierung des Use-Cases „Build A Quiz“

Im Rahmen der Projektplanung wurde der Use Case „Build A Quiz“ modelliert, welcher in Abbildung 1 abgebildet ist:

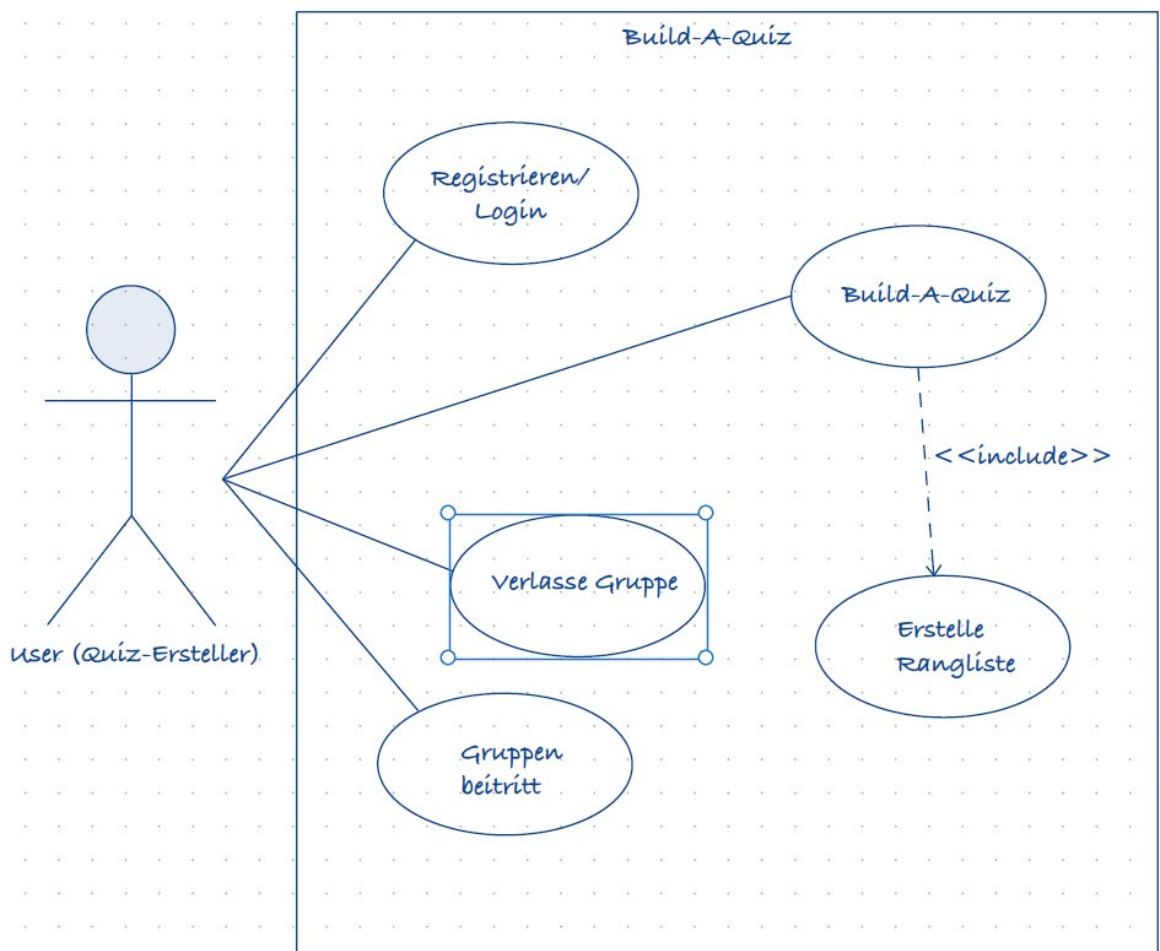


Abbildung 1: Use-Case-Diagramm "Build A Quiz"



## **4. Erstellen eines Logos für „Build A Quiz“**

Ein Logo ist ein wesentlicher Bestandteil des visuellen Erscheinungsbilds und dient der Repräsentation. Um das Produkt „Build A Quiz“ entsprechend anziehend und interessant darzustellen, sollte ein Logo entworfen werden, das zu der Quiz-Webseite passt.

### **4.1. Design des Logos**

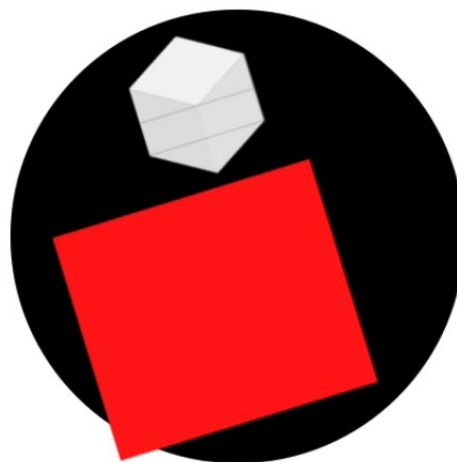
Für das Design des Logos galt es einige wesentliche Regeln einzuhalten. So sollte das Logo zum einen einfach verständlich und einprägsam sein. Zum anderen sollte das Produkt unverwechselbar erkenntlich dargestellt werden. Als auffallende Farben wurden rot und türkis gewählt, die sich innerhalb der Webseite stets wiederfinden. In Anspielung auf den Titel „Build A Quiz“ erschien es logisch, als Symbol ein Fragezeichen zu wählen. Dieses wurde auf einen dreidimensionalen Hintergrund gelegt, um die Vielfalt der Quizfragen darzustellen. Auch der dreidimensionale Schrifteffekt des Fragezeichens, der in den Farben der Webseite, rot und türkis, gehalten war, spielt auf die Breite der Quizfragen an. Im Hinblick auf die Einprägsamkeit wurde auf weitere Elemente verzichtet und somit das KISS-Prinzip (Keep It Simple Stupid) angewandt. Zur Visualisierung der unendlichen Möglichkeiten an Quizfragen wurde das gesamte Logo vor den Hintergrund eines Kreises gelegt. Das finale Logo ist im Folgenden in Abbildung 2 zu sehen:



*Abbildung 2: Logo für "Build A Quiz"*

#### **4.2. Animation des Logos**

Das Logo von „Build A Quiz“ sollte zudem als Eyecatcher dienen. Um diesen Effekt zu erzielen wurde das Logo zum Leben erweckt, indem eine schlichte Animation hinzugefügt wurde. Im Sinne des KISS-Prinzips sollte hier jedoch nicht zu viel Ablenkung entstehen. Daher wird die Schrift durch einen roten Balken ins Bild eingefügt, welcher zunächst die Aufmerksamkeit auf sich zieht. (vgl. Abbildung 3)



*Abbildung 3: Teil 1 der Animation des Logos*

Durch den etwas späteren Einblend-Effekt für das Fragezeichen wird noch eine leicht geheimnisvolle Atmosphäre geschaffen, die den Nutzer neugierig auf die zu beantwortenden Quizfragen machen soll. (vgl. Abbildung 4)



*Abbildung 4: Teil 2 der Animation des Logos*

## **5. Implementierung der Login-Möglichkeit**

Die Webseite „Build A Quiz“ dient den Nutzern dazu Quiz zu erstellen und zu spielen. Zudem ist es möglich Bestenlisten für die einzelnen Quiz zu generieren und anzuzeigen. Um die Ergebnisse einzelnen Nutzer zuordnen zu können, ist es notwendig eine Nutzerverwaltung zu implementieren.

### **5.1. Anforderungsanalyse**

Die Nutzerdaten müssen in der Datenbank in einer Tabelle gespeichert werden. Es ist wichtig, sich die Datenbankstruktur vorab zu überlegen, um beispielsweise Redundanz zu vermeiden.

Für den Login in die Quiz-Webseite werden lediglich ein Nutzernamen sowie ein Passwort für den jeweiligen Nutzer benötigt. Diese Spalten werden in der Datenbank daher als VARCHAR und mit der Eigenschaft not null angelegt.

Weiterhin wird bei der Registrierung eines neuen Nutzers automatisch eine ID vergeben. Die Spalte ID muss daher in der Datenbank als unique, not null und auto-increment angelegt werden. Die ID dient innerhalb der Datenbank als Primary Key, durch den der Nutzer eindeutig identifiziert werden kann.

Bei der Implementierung der Login-Funktion fiel auf, dass die Verbindung zur Datenbank noch nicht in eine eigene config-Datei ausgelagert war. Um Fehler und Redundanz zu vermeiden wurde die mysqli\_connect-Funktion mit den entsprechenden Parametern in die Datei config\_db.php ausgelagert. Die Überprüfung der Verbindung wird ebenfalls in der Datei durchgeführt und es wird gegebenenfalls eine passende Fehlermeldung ausgegeben. So kann nun die Datei config\_db.php einfach per require\_once in den jeweiligen Dateien eingebunden werden, wenn benötigt.

## **5.2. Implementierung der Registrierungs-Seite**

Die Nutzer müssen sich zunächst registrieren, damit sie der Webseite bekannt sind. Dies erfolgt über die Seite register.php, in welcher über ein Formular die relevanten Daten abgefragt werden: Username, Password und Confirmation Password.

Das zugehörige PHP-Skript auf der Seite prüft zunächst die Eingaben ab:

- Username:

Sollte der Nutzernamen leer sein oder ungültige Zeichen enthalten, so wird eine entsprechende Fehlermeldung ausgegeben.

Weiterhin wird geprüft, ob der Nutzernamen bereits in der Datenbank vorhanden ist. Falls ja, wird ebenfalls eine entsprechende Fehlermeldung ausgegeben.

- Password:

Auch bei der Passworteingabe wird überprüft, ob das Feld leer gelassen wurde und gegebenenfalls eine entsprechende Fehlermeldung ausgegeben.

Sollte das Passwort kürzer als sechs Zeichen sein, erhält der Nutzer die Rückmeldung ein längeres Passwort zu wählen.

- Confirmation Password:

Das Passwort zur Bestätigung darf ebenfalls nicht leer sein, ansonsten erhält der Nutzer eine entsprechende Rückmeldung.

Weiterhin wird geprüft, dass das Confirmation Password mit dem Passwort übereinstimmt. Falls nicht wird ausgegeben, dass die Passwörter nicht matchen.

Wenn alle Checks erfolgreich absolviert wurden, kann der neue Nutzer in der Datenbank angelegt werden. Dies geschieht wie in Abbildung 5 zu sehen über so genannte prepared Statements, um SQL injections in die Datenbank zu vermeiden:

```
// Prepare an insert statement
$sql = "INSERT INTO users (username, password) VALUES (?, ?)";

if($stmt = mysqli_prepare($link, $sql)){
    // Bind variables to the prepared statement as parameters
    mysqli_stmt_bind_param($stmt, "ss", $param_username, $param_password);

    // Set parameters
    $param_username = $username;
    $param_password = password_hash($password, $algo: PASSWORD_DEFAULT); // Creates a password hash

    // Attempt to execute the prepared statement
    if(mysqli_stmt_execute($stmt)){
        // Redirect to login page
        header( header: "location: login.php");
    } else{
        echo "Oops! Something went wrong. Please try again later.";
    }

    // Close statement
    mysqli_stmt_close($stmt);
}
```

Abbildung 5: PHP-Skript für die Registrierung

Es ist wichtig, dass Passwörter nicht im Klartext in der Datenbank gespeichert werden, da sie sonst von jedem lesbar sind. Bei einem Angriff auf die Datenbank wären somit sensible Nutzerdaten im Umlauf. Das Hashen der Passwörter wurde hier über die `password_hash()` Funktion von PHP umgesetzt, wodurch in der Datenbank nur kryptische Zeichenfolgen eingetragen werden. Diese Zeichenfolgen können nicht zum lesbaren Passwort zurück konvertiert werden.

### 5.3. Implementierung des Logins und Logouts

Neben der Möglichkeit zur Registrierung können auf der Login-Seite über ein Formular Nutzernamen und Passwort zum Einloggen eingegeben werden.

Das PHP-Skript überprüft zunächst, ob in diese Felder etwas eingegeben wurde. Falls ja, werden diese Daten mit den Daten

der Datenbank abgeglichen (vgl. Abbildung 5). Hierfür werden über ein prepared Statement das gehashte Passwort und der Nutzernamen, der zum eingegebenen Nutzernamen passt, abgerufen, sofern der Nutzernamen in der Datenbank existiert. Über die PHP-Funktion `password_verify()` wird überprüft, ob das eingegebene Passwort mit dem gespeicherten Passwort-Hash übereinstimmt. Wenn dies der Fall ist, sind die Credentials gültig und der Nutzer wird eingeloggt:

Es wird eine Session gestartet und die relevanten Keys werden in der `$_SESSION`-Variable gesetzt. Der Nutzer wird außerdem zur Seite `welcome.php` weitergeleitet. Sollte eine der Überprüfungen fehlschlagen, erhält der Nutzer eine entsprechende Fehlermeldung.

```
// Attempt to execute the prepared statement
if(mysqli_stmt_execute($stmt)){
    // Store result
    mysqli_stmt_store_result($stmt);

    // Check if username exists, if yes then verify password
    if(mysqli_stmt_num_rows($stmt) == 1){
        // Bind result variables
        mysqli_stmt_bind_result($stmt, &$id, &$username, $hashed_password);
        if(mysqli_stmt_fetch($stmt)){
            if(password_verify($password, $hashed_password)){
                // Password is correct, so start a new session
                session_start();

                // Store data in session variables
                $_SESSION["loggedin"] = true;
                $_SESSION["id"] = $id;
                $_SESSION["username"] = $username;

                // Redirect user to welcome page
                header("location: welcome.php");
            } else{
                // Password is not valid, display a generic error message
                $login_err = "Invalid password.";
            }
        }
    } else{
        // Username doesn't exist, display a generic error message
        $login_err = "Invalid username.";
    }
}
```

Abbildung 6: PHP-Skript für den Login

## 5.4. Implementierung der Änderung des Passworts

Die Möglichkeit zum Ändern des Passworts erhält der Nutzer, wenn er eingeloggt ist. Hierfür wird überprüft, ob in der `$_SESSION`-Variable der Key `loggedin` auf `true` gesetzt ist. Falls ja, erhält der Nutzer ein Formular, in das er sein neues Passwort zwei Mal eingeben kann. Analog zur Registrierungs-Seite wird im Folgenden überprüft, ob ein Passwort eingegeben wurde, ob es mind. sechs Zeichen enthält und ob das Bestätigungs-Passwort mit dem Passwort übereinstimmt. Wenn das Passwort alle Vorgaben erfüllt, kann die Datenbank geupdatet werden. Dies geschieht wieder über ein prepared Statement und mittels `password_hash()`-Funktion.

```
if(empty($new_password_err) && empty($confirm_password_err)){  
    // Prepare an update statement  
    $sql = "UPDATE users SET password = ? WHERE id = ?";  
  
    if($stmt = mysqli_prepare($link, $sql)){  
        // Bind variables to the prepared statement as parameters  
        mysqli_stmt_bind_param($stmt, "si", &$var1: $param_password, &...: $param_id);  
  
        // Set parameters  
        $param_password = password_hash($new_password, algo: PASSWORD_DEFAULT);  
        $param_id = $_SESSION["id"];  
  
        // Attempt to execute the prepared statement  
        if(mysqli_stmt_execute($stmt)){  
            // Password updated successfully. Destroy the session, and redirect to login page  
            session_destroy();  
            header( header: "location: index.php");  
            exit();  
        } else{  
            echo "Oops! Something went wrong. Please try again later.";  
        }  
  
        // Close statement  
        mysqli_stmt_close($stmt);  
    }  
}
```

Abbildung 7: PHP-Skript zur Änderung des Passworts



## **6. Anhang**

Im Anhang dieser Dokumentation sind alle relevanten Dokumente und Verzeichnisse aufgeführt.

### **6.1. Abbildungsverzeichnis**

Abbildung 1: Use-Case-Diagramm "Build A Quiz"	8
Abbildung 2: Logo für "Build A Quiz"	10
Abbildung 3: Teil 1 der Animation des Logos	10
Abbildung 4: Teil 2 der Animation des Logos	11
Abbildung 5: PHP-Skript für die Registrierung	14
Abbildung 6: PHP-Skript für den Login	15
Abbildung 7: PHP-Skript zur Änderung des Passworts	16

## 6.2. Protokoll-Vorlage

### Teambesprechungs-Protokoll

Protokoll-Nr.:	Datum:		Nächste Besprechung:	
NA	Zeit: von	bis	Uhr	Zeit: von
				bis
	Ort:		Ort:	
Protokoll: Frau/Herr				
	Anwesend		Protokoll gelesen, verstanden (Datum, Unterschrift)	
Alt, Theresa				
Leonov, Peter				
Rohrwild, Daniel				
Stöhr, Philipp (Scrum Master)				
Vogl, Florian				

Stichwort	Informationen	Wer?	Bis wann?

Version: 1.2	Erstellung: Rohrwild, Daniel - 13.10.2021	QM Prüfung: Stöhr, Philipp - 14.10.2021	Freigabe: Stöhr, Philipp - 14.10.2021	Kenn-Nr. BAQ- 001	Seite 1 von 3
-----------------	--	--	--	----------------------	------------------