

# IPA 2022

08.03.2022

Abschlussarbeit  
Dashboard für Firmentools

# Inhaltsverzeichnis

<b>Vorwort .....</b>	<b>4</b>
<b>Vortschrittsicherung .....</b>	<b>4</b>
<b>Aufgabenstellung .....</b>	<b>5</b>
Ausgangslage .....	5
Detaillierte Aufgabenstellung .....	5
Mittel und Methoden .....	6
Vorkenntnisse .....	6
Vorarbeiten .....	6
Neue Lerninhalte .....	6
Arbeiten in den letzten 6 Monaten.....	6
Individuelle Bewertungskriterien .....	6
I1: Systematik der Lösungsfindung/Lösungsvorschläge : .....	6
I2: Entwurf, Design (Programmierung): .....	6
I3: Plausibilisierung der Benutzer-Eingaben .....	7
I4: Codierung, Fehlerbehandlung .....	7
I5: Codingstyle - lesbarer Code .....	8
I6: GUI Design (Fokus: Benutzerfreundlichkeit über die ganze Applikation) .....	8
I7: GUI Design (Fokus: eine Maske/Bildschirm-/Webseite).....	9
<b>Hilfestellung.....</b>	<b>9</b>
<b>Zeitplan .....</b>	<b>10</b>
<b>Arbeitsjournal.....</b>	<b>11</b>
Tag 01, Montag, 07.03.2022 .....	11
Tag 02, Dienstag, 08.03.2022 .....	11
Tag 03, Mittwoch, 09.03.2022 .....	12
Tag 04, Donnerstag, 10.03.2022 .....	12
Tag 05, Montag, 14.03.2022 .....	13
Tag 06, Dienstag, 15.03.2022 .....	13
Tag 07, Mittwoch, 16.03.2022 .....	14
Tag 08, Donnerstag, 17.03.2022 .....	15
Tag 09, Montag, 21.03.2022 .....	15
Tag 10, Dienstag, 22.03.2022 .....	15
<b>Projekt.....</b>	<b>16</b>
Kurzfassung.....	16
Ausgangslage .....	16
Umsetzung .....	16
Ergebnis .....	16
Einleitung.....	17

Informieren: .....	17
Planen: .....	17
Entscheiden: .....	17
Realisieren: .....	17
Kontrollieren: .....	17
Auswerten: .....	17
Informieren .....	17
Ziele der Aufgabenstellung .....	17
Vorgaben .....	18
Fragen .....	18
Planen .....	18
Realisierungskonzept .....	18
Applikationsentwurf .....	19
Datenbank modell .....	20
Funktionsmodelle .....	21
Testfälle .....	27
Entscheiden .....	27
Varianten .....	28
Entscheid .....	28
Realisieren .....	29
Datenbank .....	29
Login Funktion .....	29
Registrieren Funktion .....	30
Tool hinzufügen Funktion .....	32
Tool bearbeiten Funktion .....	35
Tool löschen Funktion .....	37
Bild hochladen Funktion .....	38
Dashboard .....	39
Layout & Design .....	41
Plausible Eingaben & Benutzerfreundlichkeit .....	43
Redundanz entfernt .....	43
Fehlerprüfung und Lösungsanwendung .....	44
Kontrollieren .....	46
Testprotokoll .....	46
Testbericht .....	46
Reflexion .....	48
<b>Glossar .....</b>	<b>49</b>
<b>Abbildungsverzeichnis .....</b>	<b>50</b>

<b>Quellenverzeichnis.....</b>	<b>51</b>
<b>Anhang.....</b>	<b>51</b>
Quellcode .....	52
.php Seiten.....	52
.css Seiten.....	62

## Vorwort

Dies ist die Dokumentation der IPA von Shannon Neil Schürch bei der Firma ALPEIN Software SWISS AG. In dieser Dokumentation ist der genaue Ablauf der Arbeit beschrieben und das Vorgehen dahinter.

Die Dokumentation wird in zwei Teile gegliedert.

Im ersten Teil wird die detaillierte Aufgabenstellung erläutert und ich fasse die Aufgabe in meinen eigenen Worten zusammen.










Im zweiten Teil beschreibe ich dann wie das Projekt abgelaufen ist, welche Probleme aufgetreten sind und wie das Projekt getestet wurde.

## Fortschrittssicherung

Ich habe für die IPA ein neues Repository auf GitHub erstellt und werde den gesamten Projektordner mit Code und Doku mindestens einmal am Tag um 17:00 Uhr pushen.

Meine Commits werden immer mit einer erklärenden Notiz ergänzt, die den Fortschritt durchsichtiger gestalten soll.

Ausserdem befindet sich in der Repository ein Unterordner vorherige Versionen wo ich jeden Tag gegen Feierabend die aktuelle Dokumentation reinkopiere und umbenenne, damit ich immer auf die vorherigen Versionen zugreifen kann.

	IPA-2022-ALPEIN_Software_SWISS_AG_v1...	07.03.2022 16:43	Microsoft Word D...	193 KB
	IPA-2022-ALPEIN_Software_SWISS_AG_v2...	08.03.2022 16:40	Microsoft Word D...	194 KB
	IPA-2022-ALPEIN_Software_SWISS_AG_v3...	09.03.2022 16:48	Microsoft Word D...	197 KB
	IPA-2022-ALPEIN_Software_SWISS_AG_v4...	10.03.2022 17:33	Microsoft Word D...	315 KB
	IPA-2022-ALPEIN_Software_SWISS_AG_v5...	14.03.2022 16:43	Microsoft Word D...	495 KB
	IPA-2022-ALPEIN_Software_SWISS_AG_v6...	15.03.2022 16:39	Microsoft Word D...	534 KB
	IPA-2022-ALPEIN_Software_SWISS_AG_v7...	16.03.2022 16:35	Microsoft Word D...	538 KB
	IPA-2022-ALPEIN_Software_SWISS_AG_v8...	17.03.2022 16:55	Microsoft Word D...	627 KB
	IPA-2022-ALPEIN_Software_SWISS_AG_v9...	21.03.2022 16:33	Microsoft Word D...	638 KB

# Aufgabenstellung

## Beschreibung

Ein Dashboard, indem man Firmeninterne Tools erstellen, löschen und bearbeiten kann. In PHP oder typo3

## Ausgangslage

Shannon interessiert sich sehr für die Webentwicklung mit PHP. Er konnte in den letzten Monaten schon einige Erfahrungen darüber sammeln. Daher haben wir uns entschieden, dass er für seine IPA eine kleine Web-Applikation entwickeln soll.

## Detaillierte Aufgabenstellung

Das Ziel der IPA ist es ein Dashboard zu entwickeln in dem wir firmeninterne Tools verwalten können. Es soll möglich sein, Tools hinzuzufügen, zu bearbeiten und Sie löschen zu können. Das Projekt soll ganz einfach in einem Text-Editor und auf dem Localhost entwickelt werden. Er wird dafür Sublime Text 3 und den Apache Server von XAMPP verwenden, um das ganze Projekt bei sich lokal auf dem Rechner abrufen zu können.

Jedes Tool soll mit einem dazu passenden Bild und einem Link versehen werden können. Bei den Bildern ist es wichtig zu beachten, dass die Bilder nicht in einer Datenbank, sondern direkt im assest Ordner abgespeichert und von da dynamisch abgerufen werden können. Ausserdem soll ein Login System mit einer Registrierung Funktion vorhanden sein. Wobei wichtig ist, dass nur verschlüsselte Passwörter in der Datenbank landen.

### Ergänzung zur verdeutlichung:

Der Kandidat soll einen Entwurf der Applikation entwerfen. In dem ersichtlich ist, welche Funktionen für das Projekt benötigt werden. Ausserdem soll er ein Nassi-Shneiderman-Diagramm zu jeder Funktion erstellen, in dem ersichtlich ist, wie die Funktionen agieren.

Die Tools bestehen aus drei Elementen, ein Bild, ein Titel und ein Link zu dem Tool. Und noch ein Erstellungs- und Bearbeitungszeitpunkt für den Verlauf

Es gibt zwei Datenbanken, einmal für die Benutzerdaten in der nur verschlüsselte Passwörter gespeichert werden und eine für die Tools selber. Shannon soll eine Lösung entwickeln, bei jeder Abfrage der Tools die passenden Bilder abzurufen.

Bei den Tools handelt es sich um Websites wie zum Beispiel Pipedrive oder unser Ticket-system. Der Link dazu soll das Tool einfach aufrufen.

Es ist geplant, dass jeder Benutzer eigene Tools ablegen kann, aber vor allem soll er lernen verschlüsselte Benutzerdaten abzulegen. Da das grösste Produkt unserer Firma, das PassSecurium ein Passwortmanager ist und ein grosser Verkaufspunkt davon ist, dass wir als Betreiber keine Einsicht in die Nutzerdaten haben.

Es soll ausserdem ein Verlauf eingebaut werden, in dem man einsehen kann, wann welche Tools bearbeitet, gelöscht oder hinzugefügt wurden.

Eingaben werden beim Anmelden und beim Registrieren gemacht. Dazu gehört jeweils ein Benutzername bzw. eine E-Mail und ein Passwort. Dazu soll überprüft werden, ob es sich um eine realistische E-Mail handelt und ob das Passwort sicher genug ist.

**Die besonderen Knacknüsse:**

Die Bilder der Tools müssen in einem Assets-Ordner abgelegt werden und dürfen nicht in einer Datenbank gespeichert werden. Grund dafür ist, dass wir sehen wollen, wie Shannon mit einer neuen Situation umgehen kann und welche Lösung er dafür entwickelt.

Die Bilder, Namen und Links der Tools sollen bearbeitet werden können. Also muss bei Änderung des Bildes nicht nur ein Eintrag in die Datenbank gemacht werden, sondern das passende Bild im Assets Ordner auch ersetzt und neu hinterlegt werden.

Ausserdem dürfen in der Benutzerdatenbank keine Passwörter ersichtlich sein, sondern nur eine verschlüsselte Version davon. Shannon musste noch nie dafür sorgen, dass Benutzerdaten sicher abgespeichert werden und dieses Projekt eignet sich perfekt, da es sich um ein firmeninternes Dashboard handelt.

## Mittel und Methoden

PHP, HTML und SQL Hardware: Laptop (auf dem die gesamte Applikation läuft.)

## Vorkenntnisse

Shannon hat in seiner Freizeit schon einige ähnliche Applikationen zum Spass entwickelt und kennt sich daher schon recht gut mit PHP aus. Er kennt den Apache Server und das MySQL System von XAMPP bestens von zu Hause und von der Schule.

## Vorarbeiten

Er hat als Vorarbeit XAMPP auf den Laptop installiert und getestet. Dazu hat er eine einfache Webseite mit MySQL Anbindung auf dem Apache Server aufgesetzt. Um zu überprüfen, ob der Laptop für das Projekt einsatzbereit ist.

## Neue Lerninhalte

Obwohl er schon oft Login Systeme entwickelt hat, musste er noch nie verschlüsselte Daten auf dem Server speichern. Er wird lernen, wie man Passwörter so verschlüsselt, dass auch ein Administrator nur mit grossen Schwierigkeiten an das Passwort kommt.

## Arbeiten in den letzten 6 Monaten

Webdesign mit PHP und HTML, first- und secondlevel Support für unsere Produkte PassSecurium und SwissSecurium.

## Individuelle Bewertungskriterien

### I1: Systematik der Lösungsfindung/Lösungsvorschläge :

**Bezeichnung:** Systematik der Lösungsfindung/Lösungsvorschläge.

**Definition (Leitfrage):** Ist es nachvollziehbar, warum gerade diese Lösung gewählt wurde (was waren die Kriterien)?

**Gütestufe 3:** Die Lösung ist sauber dargelegt. Weil mit Alternativen verglichen wird, können die Qualität und die Vorzüge der vorgeschlagenen Lösung nachvollzogen werden.

**Gütestufe 2:** Die Lösung ist teilweise dargelegt. Alternativen sind erwähnt.

**Gütestufe 1:** Nur wenig Information, warum gerade diese Lösung bevorzugt wurde. Keine Vergleiche mit Alternativen dokumentiert.

**Gütestufe 0:** Die Lösungsfindung ist nicht ersichtlich.

## I2: Entwurf, Design (Programmierung):

**Bezeichnung:** Entwurf, Design (Programmierung)

**Definition (Leitfrage):** Kann der Kandidat einen Systementwurf vorlegen, der das Problem angemessen/verständlich darstellt?

Wird in der Aufgabenstellung genauer definiert.

**Gütestufe 3:** Die Teilsysteme sind passend gewählt und die Schnittstellen sind korrekt. Das System wird einem Aussenstehenden damit verständlich gemacht.

**Gütestufe 2:** Der Systementwurf abstrahiert das System nur teilweise. Das System ist für einen Aussenstehenden weitgehend verständlich gemacht.

**Gütestufe 1:** Der Systementwurf müsste/könnte den Problemraum präziser beschreiben. Es ist schwierig, das System aufgrund des Entwurfes zu verstehen.

**Gütestufe 0:** Es existiert kein Systementwurf (obwohl als notwendig oder sinnvoll erachtet).

## I3: Plausibilisierung der Benutzer-Eingaben

**Bezeichnung:** Plausibilisierung der Benutzer-Eingaben

**Definition (Leitfrage):** Werden die Eingaben des Benutzers überprüft?

**Gütestufe 3:** Alle Eingabefelder werden überprüft. Es ist eindeutig gekennzeichnet, welche Felder Pflichtfelder sind. Für den Benutzer ist ersichtlich, welche Wertebereiche zulässig sind. Findet die Plausibilisierung eine Fehleingabe, so wird der Benutzer mit konkreten Hinweisen geführt, das entsprechende Feld wird aktiviert.

**Gütestufe 2:** Plausibilisierung findet statt, Feedback an Benutzer ist mangelhaft/nicht eindeutig/unvollständig. Nur korrekte Daten werden übermittelt.

**Gütestufe 1:** Eingaben werden plausibilisiert, aber bei Fehlern oder fehlenden Eingaben sind die bisher gemachten Eingaben verloren oder die fehlerhaften Eingaben werden trotzdem übermittelt. Oder: es werden nicht alle Eingaben ueberprueft, welche ueberprueft werden sollten.

**Gütestufe 0:** Es findet keine Plausibilisierung statt.

## I4: Codierung, Fehlerbehandlung

**Bezeichnung:** Codierung: Fehlerbehandlung

**Definition (Leitfrage):** Fehlerbehandlung: Werden mögliche Fehler mit den entsprechenden Mitteln erkannt und behandelt?

**Gütestufe 3:** Es wurde eine Strategie zur Fehlerbehandlung eingeführt und im ganzen Code konsistent verwendet.

**Gütestufe 2:** Die Fehlerbehandlung ist lückenhaft und/oder die Fehler bleiben nach der Erkennung unbehandelt.

**Gütestufe 1:** Die Fehlerbehandlung wurde oft vergessen. Code streckenweise ohne Fehlertests (wo sie als notwendig erachtet würden).

**Gütestufe 0:** Fehlerbehandlung nicht oder nur sehr rudimentär vorhanden.



## I5: Codingstyle - lesbarer Code

**Bezeichnung:** Codingstyle - lesbarer Code

**Definition (Leitfrage):** Ist der Code lesbar geschrieben, gut gegliedert und ist die Namensgebung gut gewählt?

**Gütestufe 3:** Die Namensgebung entspricht den Vorgaben oder ist einfach gut gewählt. Die Struktur des Codes ist ebenfalls gemäss möglicher Richtlinien oder einfach übersichtlich gemacht. Es ist eine gewisse Einheit zu sehen in der Art und Weise, wie der Code strukturiert ist (d.h. es ist überall etwa gleich gemacht).

**Gütestufe 2:** Die Namensgebung ist ab und zu ungeschickt gewählt, Vorgaben sind teilweise berücksichtigt. Die Codestruktur ist uneinheitlich (so dass Lesbarkeit leidet). Die Namensgebung ist ab und zu ungeschickt gewählt, Vorgaben sind teilweise berücksichtigt. Die Codestruktur ist uneinheitlich (so dass Lesbarkeit leidet).

**Gütestufe 1:** Die Namensgebung ist öfters verwirrend oder unpräzise. Dem Code fehlt es an einigen Stellen an klarer Struktur. Vorgaben sind nicht berücksichtigt.

**Gütestufe 0:** Die Namensgebung ist verwirrend oder unpräzise. Der Code ist schlecht lesbar.

## I6: GUI Design (Fokus: Benutzerfreundlichkeit über die ganze Applikation)

**Bezeichnung:** GUI Design (Fokus: Benutzerfreundlichkeit über die ganze Applikation).

**Definition (Leitfrage):** Ist die Applikation als Ganzes benutzerfreundlich?

1. Masken in der richtigen Reihenfolge (Applikation bildet den Prozess/Workflow richtig ab).
2. Ist das Design durchgängig (gleiche Elemente am gleichen Platz)
3. Kann der Benutzer bei Fehleingaben zurück navigieren (oder ist für den Benutzer transparent dargestellt warum dies nicht möglich ist)?

**Gütestufe 3:** Alle 3 Punkte sind erfüllt.

**Gütestufe 2:** 2 Punkte erfüllt

**Gütestufe 1:** 1 Punkt erfüllt

**Gütestufe 0:** Kein Punkt ist erfüllt

## 17: GUI Design (Fokus: eine Maske/Bildschirm-/Webseite)

### Bezeichnung

GUI Design (Fokus: eine Maske/Bildschirm-/Webseite)

**Definition (Leitfrage)** Sind die Design-Grundlagen eingehalten? 1. gängige Design-Grundlagen (Schriftgrösse, Farbe) 2. übersichtlich und ergonomisch strukturiert (Anordnung der Felder und Schaltflächen). 3. sinnvolle Defaults, Felder vorausgefüllt wo möglich, Anwender muss keine redundanten Daten erfassen.

**Gütestufe 3** Alle drei Punkte sind gut oder sehr gut erfüllt.

**Gütestufe 2** Nur zwei Punkte sind gut oder sehr gut erfüllt.

**Gütestufe 1** Nur ein Punkt ist gut oder sehr gut erfüllt.

**Gütestufe 0** Das GUI ist nicht brauchbar.

## Hilfestellung

Am fünften Tag meiner IPA, dem 14.03.2022 habe ich meinen VF gefragt:

Ich hätte meine erste Frage an dich als VF meiner Arbeit: Was könnte ich einbauen, damit mein Projekt von sql Angriffen geschützt ist?

Daraufhin hat er mir 2 links geschickt, die ich mir gerne ansehen werde:

<https://www.nginx.com/products/nginx/modules/nginx-waf/> und  
<https://www.linuxlinks.com/naxsi-nginx-anti-xss-sql-injection/>

Ich habe mich allerdings dazu entschieden, dass ich so einen Schutz erst nach der IPA einbauen werde, da ich mich vorerst auf die für die Bewertung relevanten Dinge konzentrieren will.

22.03.2022

# Zeitplan

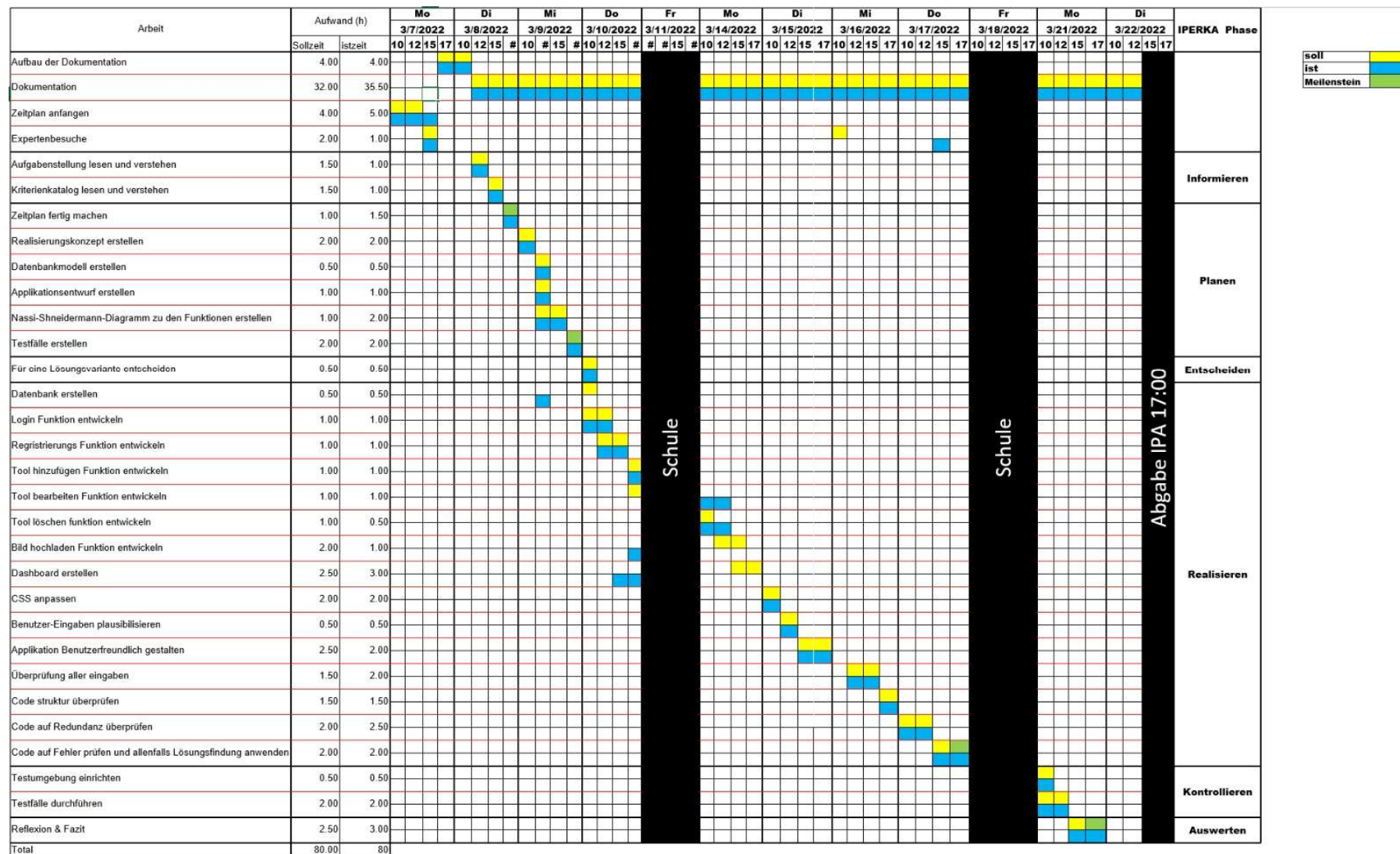


Abbildung 1: Zeitplan

# Arbeitsjournal

## Tag 01, Montag, 07.03.2022

<b>Arbeitsschritte</b>	Zeitplan erstellt. Dokumentation aufgebaut.
<b>Erreichte Ziele</b>	Zeitplan ohne soll ist Vergleich fertiggebracht.
<b>Probleme</b>	Ich habe für den Zeitplan mehr Zeit benötigt als ich erwartet hatte.
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	Eine extra Stunde am Zeitplan, dafür eine Stunde weniger an der Doku.
<b>Reflexion</b>	Ich war sehr nervös, aber nachdem ich den ersten Tag überstanden habe gehe ich mit grosser Motivation auf den zweiten Tag zu.

Abbildung 2: Arbeitsjournal vom 07.03.2022

## Tag 02, Dienstag, 08.03.2022

<b>Arbeitsschritte</b>	Zeitplan mit soll ist Vergleich fertig machen Dokumentation weiter aufgebaut Aufgabenstellung studiert Bewertungskriterien studiert
<b>Erreichte Ziele</b>	Zeitplan mit soll ist Vergleich fertig. Inhaltsverzeichnis fertig. Vorwort verfasst. Arbeitsjournal erstellt. Aufgabe verstanden. Bewertungskriterien verstanden.
<b>Probleme</b>	Ich habe heute meinem HEX den Zeitplan zugeschickt und dachte aus irgend einem Grund, dass ich ihn falsch angesprochen hätte. Daraufhin habe ich mich per Mail gerechtfertigt und erst danach bemerkt, dass meine ursprüngliche Mail fehlerfrei war.
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	-
<b>Reflexion</b>	Heute bin ich meiner Meinung nach weit gekommen. Die gesamte Struktur wird sich nicht mehr ändern. Also kann ich jetzt Stück für Stück die Doku ausfüllen und sie auch für zukünftige Kandidaten als Vorlage sichern. Das mit dem Zeitplan war ein wenig peinlich.

Abbildung 3: Arbeitsjournal vom 08.03.2022

## Tag 03, Mittwoch, 09.03.2022

<b>Arbeitsschritte</b>	<b>Realisierungskonzept</b> <b>Datenbankmodell</b> <b>Applikationsentwurf</b> <b>Funktionsdiagramme</b> <b>Testfälle erstellt</b> <b>Datenbank erstellt</b>
<b>Erreichte Ziele</b>	Realisierungskonzept Datenbankmodell Applikationsentwurf Funktionsdiagramme Testfälle erstellt Datenbank erstellt
<b>Probleme</b>	-
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	Ich habe heute schon früher als geplant die Datenbank erstellt, weil ich es für die einfachste Möglichkeit gehalten habe das Datenbankmodell zu erstellen
<b>Reflexion</b>	Ich bin froh, dass ich heute für die Datenbank Zeit gefunden habe. Jetzt kann ich morgen früh an der Doku arbeiten

Abbildung 4: Arbeitsjournal vom 09.03.2022

## Tag 04, Donnerstag, 10.03.2022

<b>Arbeitsschritte</b>	<b>Für Lösungsvariante entscheiden</b> <b>Login Funktion entwickeln</b> <b>Registrierung Funktion entwickeln</b> <b>Tool hinzufügen Funktion entwickeln</b> <b>Bild hochladen Funktion entwickeln</b> <b>Dashboard erstellen</b>
<b>Erreichte Ziele</b>	Für Lösungsvariante entscheiden Login Funktion entwickeln Registrierungs Funktion entwickeln Tool hinzufügen Funktion entwickeln Bild hochladen Funktion entwickeln Dashboard erstellen
<b>Probleme</b>	-
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	Bild hochladen Funktion entwickelt Dashboard erstellt
<b>Reflexion</b>	Heute lief alles bestens und ich bin gut vorangekommen. Nachdem ich die Tools hinzufügen Funktion entwickelt habe, habe ich mich entschieden, die Tools bearbeiten und löschen Funktion zu verschieben und das Dashboard vorzuziehen. Weil es für mich viel einfacher ist, die Tools bearbeiten und Tool löschen Funktion zu entwickeln, wenn ich weiss, wie die Tools im Dashboard dargestellt werden. Ausserdem hat sich herausgestellt, dass die Bild hochladen Funktion sehr einfach zu entwickeln ist. Daher habe ich sie direkt zusammen mit dem Tool hinzufügen Funktion erstellt.

Abbildung 5: Arbeitsjournal vom 10.03.2022

## Tag 05, Montag, 14.03.2022

<b>Arbeitsschritte</b>	<b>Tool bearbeiten Funktion entwickeln</b> <b>Tool löschen Funktion entwickeln</b>
<b>Erreichte Ziele</b>	Tool bearbeiten Funktion Tool löschen Funktion
<b>Probleme</b>	-
<b>Hilfestellung</b>	Ich habe meine VF gefragt, ob er mir einen Tipp geben kann, wie ich die Applikation sicherer gestalten kann. Er hat mir daraufhin den Tipp gegeben, mich über nginx schlau zu machen, was ich gerne machen werde.
<b>Ausserplanmässige Arbeiten</b>	Tool bearbeiten und Tool löschen Funktion erst nach Dashboard erstellen
<b>Reflexion</b>	Heute bin ich vor allem mit der Doku vorangekommen. Ausserdem hatte ich recht mit der Annahme, dass es einfacher wäre, die beiden Funktionen erst zu entwickeln, wenn das Dashboard steht. Im Rückblick war das wohl schlecht geplant.

Abbildung 6: Arbeitsjournal vom 14.03.2022

## Tag 06, Dienstag, 15.03.2022

<b>Arbeitsschritte</b>	<b>CSS anpassen</b> <b>Benutzereingaben plausibilisieren</b> <b>Applikation benutzerfreundlich gestalten</b>
<b>Erreichte Ziele</b>	CSS ist angepasst, die ganze Applikation ist jetzt hübsch. Benutzereingaben machen Sinn und Benutzer werden auf Fehler hingewiesen Alle Eingaben machen Sinn, es gibt immer die Möglichkeit abzubrechen bzw. zurück zu dem Dashboard zu gelangen
<b>Probleme</b>	Heute als ich am Feierabend committen wollte habe ich einen Error gekriegt und musste sourcetree neu installieren. Nach der Neuinstallation hat meine Repository auf dem Rechner nicht funktioniert, daher habe ich sie in einen neuen Projektordner geklont und arbeite fortan aus dem neuen Ordner.
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	20 min sourcetree reparieren und sicherstellen das keine Daten verloren gegangen sind.
<b>Reflexion</b>	Das ganze nimmt immer mehr Gestalt an, was mich sehr zu sehen freut. Heute war gut bis auf den kleinen Hertz intakt als ich von sourcetree mit lauter Errors begrüsst wurde. Jetzt bin ich aber wieder beruhigt, da das Problem schnell gelöst war. Ich bin mir nicht sicher, was mit sourcetree schiefgelaufen ist, aber ich hoffe, es kommt nicht wieder vor.

Abbildung 7: Arbeitsjournal vom 15.03.2022

## Tag 07, Mittwoch, 16.03.2022

<b>Arbeitsschritte</b>	<b>Überprüfung aller Eingaben</b> <b>Code Struktur überprüfen</b> <b>Code durchkommentiert</b>
<b>Erreichte Ziele</b>	Alle Eingaben werden nun hinterfragt Der code ist ordentlich und kommentiert
<b>Probleme</b>	-
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	-
<b>Reflexion</b>	Der Code ist nun fast fertig, morgen werde ich noch redundante Teile rauslöschen und den Code auf Fehler überprüfen.

Abbildung 8: Arbeitsjournal vom 16.03.2022

## Tag 08, Donnerstag, 17.03.2022

<b>Arbeitsschritte</b>	<b>Code auf Redundanz überprüfen</b> <b>Code auf Fehler prüfen und allenfalls Lösungsfindung anwenden</b>
<b>Erreichte Ziele</b>	Redundanz entfernt Fehlerprüfung und Lösungsanwendung
<b>Probleme</b>	-
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	-
<b>Reflexion</b>	Ich habe viel code gefunden der überflüssig ist vor allem in dem Tool bearbeiten und der Tool erstellen Funktion. Des Weiteren habe ich einen Fehler gefunden, den ich an mehreren Stellen gelöst habe. Am Montag muss ich noch testen und danach allenfalls gefundene Probleme lösen, aber danach ist der Code fertig.

Abbildung 9: Arbeitsjournal vom 17.03.2022

## Tag 09, Montag, 21.03.2022

<b>Arbeitsschritte</b>	<b>Testumgebung eingerichtet</b> <b>Testfälle durchgeführt</b> <b>Reflexion &amp; Fazit verfasst</b>
<b>Erreichte Ziele</b>	Ich konnte alle meine Tests durchführen und konnte sogar ein Problem beheben. Reflexion & Fazit verfasst
<b>Probleme</b>	-
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	-
<b>Reflexion</b>	Der Code steht und ich bin sehr stolz auf das Ergebnis. Jetzt kann ich morgen der Doku den letzten Schliff verpassen.

Abbildung 10: Arbeitsjournal vom 21.03.2022

## Tag 10, Dienstag, 22.03.2022

<b>Arbeitsschritte</b>	<b>Reflexion und Fazit</b> <b>Korrekturen</b> <b>Feinschliff</b>
<b>Erreichte Ziele</b>	Ich habe Teil a und b der IPA abgeschlossen
<b>Probleme</b>	-
<b>Hilfestellung</b>	-
<b>Ausserplanmässige Arbeiten</b>	-
<b>Reflexion</b>	Ich bin sehr froh, dass ich alles geschafft habe und bin stolz auf das Ergebnis

Abbildung 11: Arbeitsjournal vom 22.03.2022



# Projekt

## Kurzfassung

Diese Zusammenfassung richtet sich an Leser mit Fachwissen in der Informatik und vermittelt eine erste Übersicht, welche zur Erleichterung dienen soll, um die Arbeit und deren Inhalt verständlicher zu machen.

## Ausgangslage

Zurzeit sind alle Tools, die unsere Mitarbeiter verwenden, überall verstreut.

Neu soll es einen Single point of entry geben der Übersicht über all die Tools bietet und es den Mitarbeitern erlaubt die Tools selber zu verwalten.

Dabei ist es wichtig, dass jeder Mitarbeiter seinen eigenen Bereich hat und er seine Tools selber erstellen und verwalten kann.

## Umsetzung

Das Dashboard für Firmentools wird als eine Webapplikation realisiert, die eine Datenbank mit zwei Tabellen verwendet, um die Tools und Nutzer zu verwalten.

Damit jeder Mitarbeiter seinen eigenen Bereich hat, gibt es ein Login- und Registrierungs-System. Die Autorisierung wird innerhalb der Applikation geregelt.

Es gibt einen Verlauf, indem jeder Mitarbeiter seine Interaktionen mit seinen Tools nachverfolgen kann.

## Ergebnis

Jeder Mitarbeiter hat seinen eigenen Bereich und kann auch nicht in die Bereiche anderer User einsehen.

Die Datenbank hat drei Tabellen, eine für die Benutzer und eine für die Tools, wie im Auftrag beschrieben, aber ich habe noch eine dritte Tabelle nur für den Verlauf eingebaut, weil mir keine anständige Möglichkeit eingefallen ist, den Verlauf mit nur zwei Tabellen zu realisieren.

Ein Tool hat einen Namen, der bis zu 100 Zeichen lang sein darf, einen Link, der ebenfalls bis 100 Zeichen lang sein darf und optional ein Bild, das automatisch zu einem PNG umgewandelt wird, um die Ausgabe im Dashboard zu vereinfachen.

Jedes Attribut der Tools kann unabhängig von den anderen Attributen bearbeitet werden.

Wenn ein Tool hinzugefügt, gelöscht oder verändert wird, gibt es einen Eintrag in den Verlauf des Mitarbeiters, den er selbst unter Verlauf einsehen kann.

## Einleitung

In diesem Kapitel beschäftige ich mich mit dem Aufbau des Projektes. Ich habe dieses Projekt nach der IPERKA-Methode aufgebaut, um eine gegliederte Struktur durch das ganze Projekt aufrechtzuerhalten. Unter erkläre ich kurz die 6 Schritte oder auch «Phasen», die dafür notwendig sind.

### Informieren:

Den Auftrag gut lesen und sichergehen, dass man ihn zu 100 % versteht. Sicherstellen, dass man über all die Informationen verfügt, die man benötigt, um den Auftrag korrekt auszuführen.

### Planen:

Um einen Auftrag richtig auszuführen ist es wichtig, dass man gut vorausplant. Es sollte eine Zeitschätzung und ein Soll Zeit Modell erstellt werden. Ausserdem ist es am besten, wenn man, falls möglich, mehrere Lösungsansätze für eventuelle Schwierigkeiten präsentieren kann.

### Entscheiden:

Die verschiedenen Lösungswege werden miteinander verglichen und es wird sich für die beste Lösung entschieden.

### Realisieren:

Wenn man gut geplant hat und sich für einen Projektablauf entschieden hat, kann man nun mit der Realisierung beginnen. Dabei werden alle Arbeitsschritte genau protokolliert und die ist Zeiten werden in den Zeitplan eingetragen.

### Kontrollieren:

Sobald die Entwicklung abgeschlossen ist, wird anhand des bei der Planung erstellten Testkonzeptes getestet. Wobei negative Ergebnisse direkt korrigiert und dokumentiert werden.

### Auswerten:

Zuletzt gibt es eine Reflexion der Arbeit, in der man analysiert, was alles besonders gut oder schlecht gelaufen ist und was man in zukünftigen Projekten anders umsetzen würde.

## Informieren

Um mich richtig über den Auftrag zu informieren habe ich ihn mehrmals gut durchgelesen bis ich mir sicher war, dass ich alle Erwartungen verstehe. Ausserdem habe ich mir den Leitfaden der QV 2022 heruntergeladen und dafür gesorgt, dass ich den Inhalt verstehe. Zu guter Letzt habe ich noch die Bewertungskriterien verinnerlicht und mir eine Version ausgedruckt, um immer schnell einen Blick darauf werfen zu können.

## Ziele der Aufgabenstellung

Das Ziel der IPA ist es ein Dashboard zu entwickeln in dem wir firmeninterne Tools verwalten können. Es soll möglich sein, Tools hinzuzufügen, zu bearbeiten und Sie löschen zu können. Jedes Tool soll mit einem dazu passenden Bild und einem Link versehen werden können. Ausserdem soll ein Login System mit einer Registrierung Funktion vorhanden sein.

## Vorgaben

Die Bilder zu den Tools sollen in einem Assetsordner gespeichert werden.

Es landen keine unverschlüsselten Passwörter in der Datenbank.

Es gibt zwei Datenbanken, einmal für die Benutzerdaten in der nur verschlüsselte Passwörter gespeichert werden und eine für die Tools selber. Shannon soll eine Lösung entwickeln, bei jeder Abfrage der Tools die passenden Bilder abzurufen. Fragen

Es soll ausserdem ein Verlauf eingebaut werden, in dem man einsehen kann, wann welche Tools bearbeitet, gelöscht oder hinzugefügt wurden

Eingaben werden beim Anmelden und beim Registrieren gemacht. Dazu gehört jeweils ein Benutzername bzw. eine E-Mail und ein Passwort. Dazu soll überprüft werden, ob es sich um eine realistische E-Mail handelt und ob das Passwort sicher genug ist.

## Fragen

Ich habe meinen VF gefragt, was ich einbauen könnte, um die Webapplikation ein wenig sicherer zu gestalten. Er hat mir dazu zwei links über nginx gesendet, die ich mir gerne ansehen werde. Ich habe mich jedoch dazu entschieden, vorerst darauf zu verzichten, damit ich mich auf die für die Bewertung relevanten Themen konzentrieren kann.

## Planen

In der zweiten Phase der IPERKA-Methode habe ich ein Realisierungskonzept und einen Applikationsentwurf erstellt, der einem Aussenstehenden ersichtlich machen soll, wie die Webapplikation aufgebaut ist. Danach habe ich ein Datenbankmodell entworfen, nach dem ich dann die Datenbank aufbauen werde. Anschliessend habe ich noch NassiSchneidermann-Diagramme zu jeder Funktion entworfen und ein Testkonzept erstellt.

## Realisierungskonzept

Es wird ein neuer Projektordner Dashboard erstellt, indem alle Dateien, die für die Arbeit benötigt werden, abgelegt sind.

In dem Ordner befinden sich drei Unterordner Assets, Doku und vorherige Versionen.

Im Assets Ordner werden alle Bilder der Tools und die für das Design gespeichert. Ausserdem befinden sich im Assets Ordner noch das Skript zur Erstellung von dashboarddb. Im Ordner Doku befindet sich die Dokumentation, der Zeitplan und alle anderen Dateien, die für die Abgabe benötigt werden.

Die Datenbank dashboarddb wird wie im Datenbank-Modell abgebildet implementiert. Der Code zur Erstellung der DB wird in einer SQL Datei exportiert und auch im Assets Ordner abgelegt.

Für nicht eingeloggte Benutzer dürfen keine Informationen ersichtlich sein. Wenn ohne Anmeldung auf eine der Dateien aufgerufen wird, soll man direkt in ein Login gezwungen werden. Daher wird der Login Status auf allen wichtigen Dateien direkt am Anfang überprüft.

## Applikationsentwurf

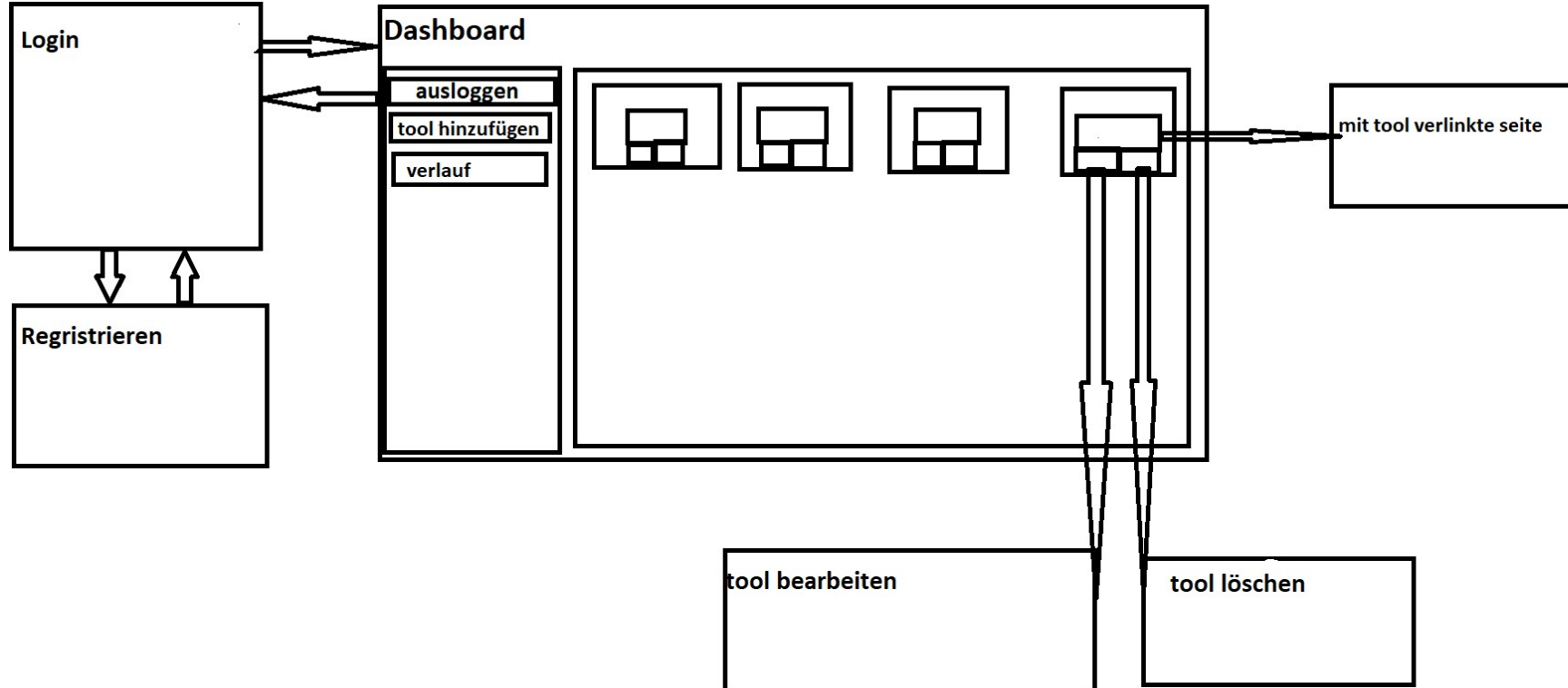


Abbildung 12: Grober Applikationsentwurf

## Datenbank Modell

Die Tabelle User speichert die Benutzer und beinhaltet nur verschlüsselte Passwörter. uid ist ein Primärschlüssel, der mit den anderen beiden Tabellen verbunden ist.

In der Tabelle tools werden die Tools der Benutzer abgespeichert, um sie dynamisch auf dem Dashboard darstellen zu können.

In der Tabelle history werden alle Änderungen an der Tabelle tools festgehalten, damit ein Benutzer Übersicht behalten kann.

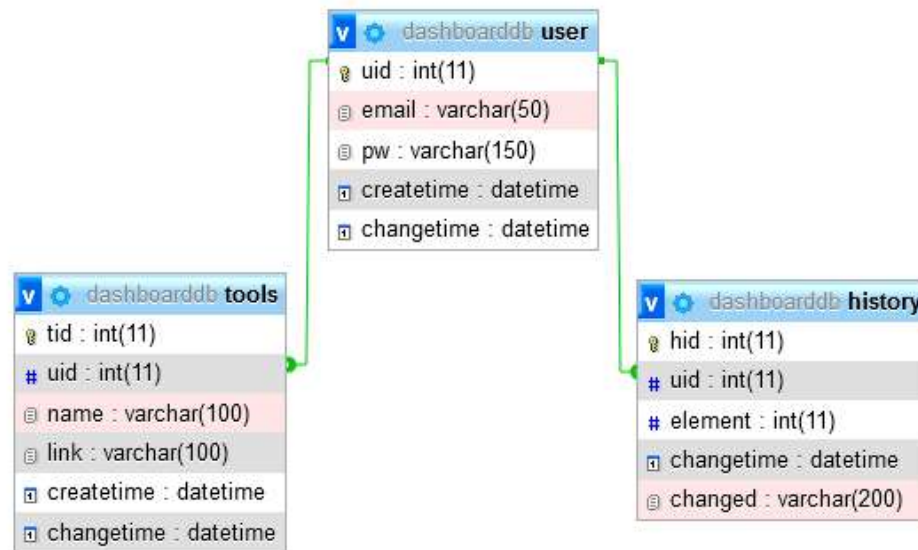


Abbildung 13: Datenbank modell

## Funktionsmodelle

### Login Funktion

#### Login

Eingabe: email, passwort		
Verbinden zu Dashboard DB		
ergebnis = alles aus tabelle user von Dashboard DB wo eingetragene email = eingegebene email		
wennn ergebnis einen inhalt hat		
T		F
saltz = erstelldatum von ergebnis verschlüsseltes passwort = md5 mit eingegebenen passwort und saltz		Ausgabe: "Email ist noch nicht registriert"
wenn verschlüsseltes passwort = passwort aus ergebnis		authentikation = nein
T	F	
authentikation = ja	Ausgabe: "Falsches Passwort"	
User daten in Session variabel speichern auf startseite weiterleiten	authentikation = nein	

Abbildung 14: Login funktion NassiSchneidermann

## Registrieren Funktion

### Registrieren

Eingabe: email, passwort, passwort1		
Verbinden zu DashboardDB		
ergebnis = alles aus tabelle user von Dashboard DB wo eingetragene email = eingegebene email		
T	wennn ergebnis einen inhalt hat	
	F	
Ausgabe: "Email bereits registriert"	wenn passwort = passwort1	
	T	F
	Zeit = aktuelle Zeit	Ausgabe: "Passwörter stimmen nicht überein"
	in tabelle user von DashboardDB neuen user (email, erstellt, geändert) hinzufügen mit den werten (email, Zeit, Zeit)	
	zeitholen = der letzte eintrag aus taballe user in DashboardDB saltz = erstellungsdatum von zeitholen verschlüsseltes Passwort = md5 mit passwort und saltz	
	in tabelle user von DashboardDB user updaten (passwort) mit den werten (verschlüsseltes Passwort) wo eingetragene email = emial	
	authentikation = ja	
	auf startseite weiterleiten	

Abbildung 15: Registrieren funktion NassiSchnediermann

*Ausloggen Funktion***ausloggen**

wen Session Variabel authentication einen inhalt hat	
T	F
Session Variabel authentication = false	auf login seite weiterleiten
auf login seite weiterleiten	

*Abbildung 16: Ausloggen Funktion NassiSchneidermann*



## Tool erstellen Funktion

### Tool erstellen

Eingabe: Name, Link, Bild(optional)		
Verbinden zu Dashboard DB		
prüfen ob "https://" im Link aus Eingabe vorhanden ist		
T		F
in Tabelle user von Dahboard DB neues tool (name = Name aus eingabe, link = Link aus eingabe, uid = uid aus Session Variabel, createtime = Zeit	in Tabelle user von Dahboard DB neues tool (name = Name aus eingabe, link = https://+Link aus eingabe, uid = uid aus Session Variabel, createtime = Zeit	
wen Bild aus Eingabe einen Inhalt hat		
T		F
Ergebnis = Daten von dem zuletzt erstellten tool in Dashboard DB		auf Startseite weiterleiten
eintrag in den Verlauf das "name aus Ergebnis tool erstellt wurde"		
Bild unter "/assets+Ergebnis+.png" speichern		
auf Startseite weiterleiten		

Abbildung 17: Tool erstellen Funktion NassiSchneidermann

## Tool bearbeiten Funktion

### Tool bearbeiten

id = id die im Link weitergeleitet wird name = name der im Link weitergeleitet wird link = link der im Link weitergeleitet wird Zeit = aktuelle Zeit			
Eingabe: Name(optional), Link(optional), Bid(optional)			
T			F
zu Dashboard DB verbinden			auf Startseite weiterleiten
prüfen ob "https://" im eingegebenen Link vorhanden ist			
T	F		
tool mit tid = id updaten name = name aus Eingabe, link = Link aus Eingabe, changetime = Zeit		tool mit tid = id updaten name = name aus Eingabe, link = https://+Link aus Eingabe, changetime = Zeit	
wen Bild einen Inhalt hat		wen Bild einen Inhalt hat	
T	F	T	F
Ergebnis = Daten von dem zuletzt veränderten tool in Dashboard DB	auf Startseite weiterleiten	Ergebnis = Daten von dem zuletzt veränderten tool in Dashboard DB	auf Startseite weiterleiten
Bild unter "/assets+id aus Ergebnis+.png" speichern		Bild unter "/assets+id aus Ergebnis+.png" speichern	
Eintag in Verlaf das name aus Ergebnis tool verändert wurde		Eintag in Verlaf das name aus Ergebnis tool verändert wurde	
auf Startseite weiterleiten		auf Startseite weiterleiten	

Abbildung 18: Tool bearbeiten Funktion NassiSchneidermann

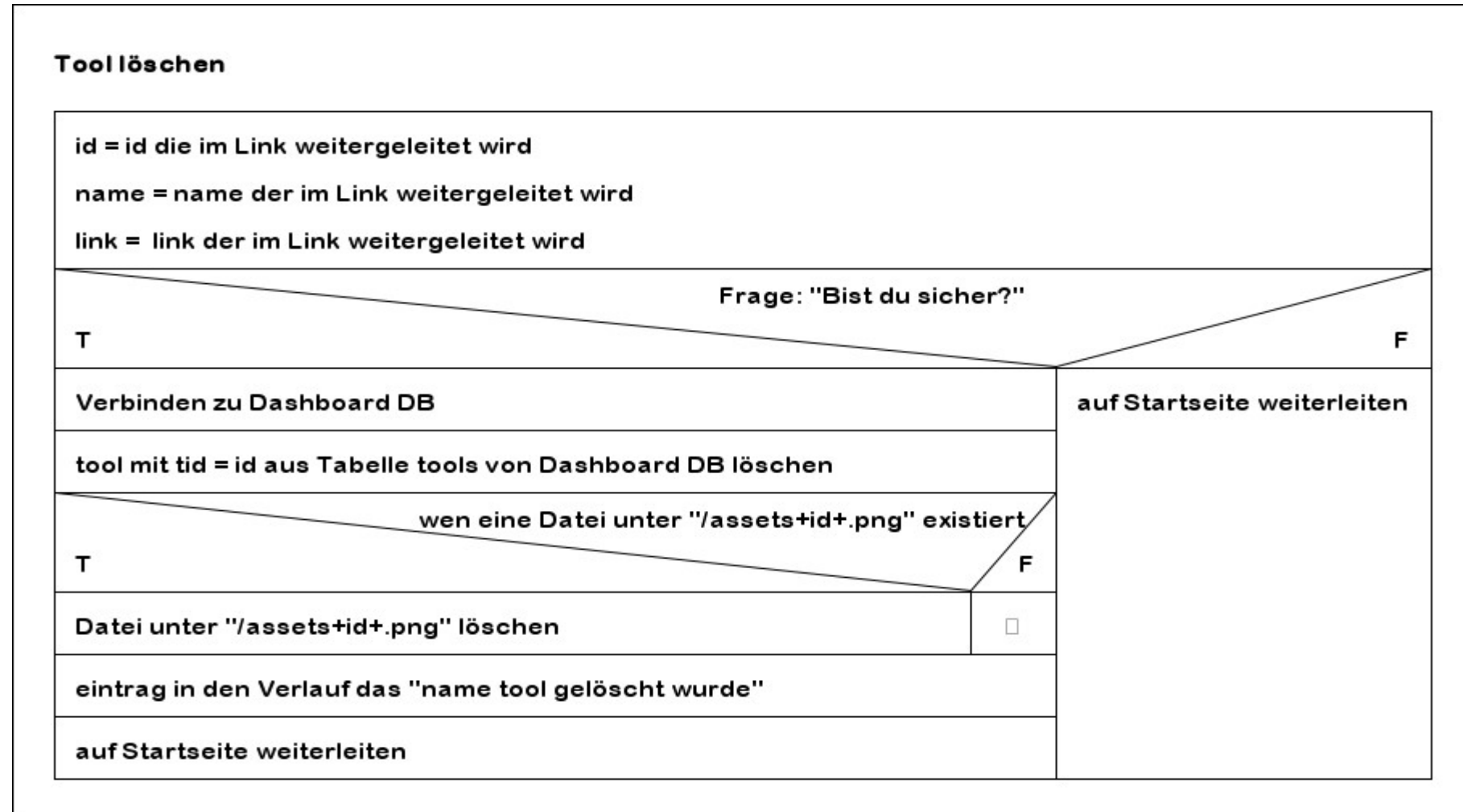
*Tool löschen Funktion*

Abbildung 19: Tool löschen Funktion NassiSchneidermann

## Testfälle

Test	Erwartetes Ergebnis	Test Nr.
Ohne Login versuchen auf index.php zu gelangen durch das ändern der URL zu localhost/Dashboard/index.php	Sofortige weiterleitung auf login.php	1
Ohne Login versuchen auf del.php zu gelanden durch das ändern der URL zu localhost/Dashoard/del.php	Sofortige weiterleitung auf login.php	2
Ohne Login versuchen auf edit.php zu gelanden durch das ändern der URL zu localhost/Dashoard/edit.php	Sofortige weiterleitung auf login.php	3
Ohne Login versuchen auf create.php zu gelanden durch das ändern der URL zu localhost/Dashoard/create.php	Sofrotige weiterleitung auf login.php	4
In der tool erstellen funktion aufabbrechen drücken	Weiterleitung zum Dashboard	5
Tool erstellen ohne namen aber mit link und Bild	Meldung das Name ein Pflichtfeld ist	6
Tool erstellen ohne link aber mit namen und Bild	Meldung das Link ein Pflichtfeld ist	7
Tool erstellen ohne ein Bild aber mit namen und link	Tool wird ohne Bild erstellt, weiterletung auf dashboard	8
Tool mit allen atributen erstellen	Tool wird mit eingegebenen Daten und ausgewähltem Bild erstellt	9
Ind er tool bearbeiten funktion auf abbrechen drücken	Zurück auf das Dashboard weiterleiten	10
Tool bearbeiten ohne namen aber mit link und Bild	Tool wird mit dem vorherigen Namen und neuem link und neuem Bild upgedated	11
Tool bearbeiten ohne link aber mit namen und Bild	Tool wird mit dem vorherigen Link und neuen nameun und neuen Bild upgedated	12
Tool bearbeiten ohne Bild aber mit namen und link	Tool name und link werden upgedated und das Bild verbleibt unberührt	13
In der tool löschen funktion auf abbrechen drücken	Zurück auf Startseite weiterleiten	14
In der tool löschen funktion auf Ja drücken	Tool wird aus der Datenbank gelöscht und das dazugehörige Bild wird falls vorhandenen gelöscht	15
Auf ausloggen drücken	Auhtentication wird ungültig und auf login seite weiterleiten	16
Etwas unter Email im login eingeben das nicht ein @ und einen . beinhaltet	Meldung das das keine gültige Email ist	17
Eine noch nicht registrierte Email im login eingeben	Meldung das Email noch nicht registriert ist	18
Falsches passwort mt registrierter email im login eingeben	Meldung das Passwort ungültig ist	19
Richtiges passwort mit registrierter email im login eingeben	Authentikation wird gültig weiterleitung auf Startseite	20
Im Login auf Registrieren drücken	Weiterleitung auf registrieren seite	21
Bereits registrierte email bei registrierung eingeben	Meldung das die Email noch nicht registriert ist	22
Etwas unter Email bei der registrierung eingeben das nicht ein @ und einen . beinhaltet	Meldung dass das keine gültige Email ist	23
Zwei verschiedene passwörter bei der Registrierung angeben	Meldung dass die passwörter nicht übereinstimmen	24
Nicht registrierte Email und zwei übereinstimmende passwörter bei Registrierung angeben	Neuer user wird erstellt login erfolgt und weiterleitung auf Startseite	25

Abbildung 20: Testfälle

## Entscheiden

Es gibt bei dem Projekt einige Entscheidungen zu treffen. Vor allem bei dem Layout gibt es verschiedene Varianten, die ich unten auflisten werde.

### Varianten

#### *Ausloggen Button*

1. Rechts oben im Kopf der Applikation.
2. Als einer der Links im Sidenav.
3. Der ausloggen Button erscheint, wenn man mit der Maus über den Benutzernamen schwebt.

#### *Anzahl der Tools, die im Dashboard pro Reihe dargestellt werden.*

1. Drei
2. Vier
3. Fünf

#### *Akzentfarben*

1. Hellblau mit gelb
2. Ein verwaschenes Hellblau mit Pastell gelb
3. Blau mit Hellblau
4. Blau mit rot

## Entscheid

#### *Ausloggen Button*

Ich habe mich dazu entschieden, die zweite Variante zu verwenden, da ich nicht plane oben rechts im Kopf der Seite irgendetwas darzustellen. Die dritte Variante wäre für mich auch eine gute Option gewesen, aber ich habe mir gedacht desto einfacher die Bedienung der Applikation desto besser.

#### *Anzahl der Tools, die im Dashboard pro Reihe dargestellt werden.*

Ich habe mich dazu entschieden pro Reihe vier Tools darzustellen, da ich die Grösse der Tools dann am ansprechendsten finde.

#### *Akzentfarben*

Ich habe mich für die zweite Version entschieden, da ich denke, dass dann schwarzer Text und weisser Text gut lesbar sein werden.

Hier sind die beiden Farbcodes, für die ich mich entschieden habe.

Verwaschenes Hellblau: #e8f9fb;

Pastell Gelb: #ffffd1;

## Realisieren

### Datenbank

Ich habe die Datenbank wie in meinem Datenbankmodell geplant erstellt.

### Login Funktion

Die Logikfunktion gibt die Authentifikation zurück. Nach der Eingabe wird geprüft, ob die E-Mail registriert ist oder nicht. Falls die E-Mail registriert ist, verwende ich die Erstellungszeit zur Verschlüsselung des Passwortes. Anschliessend wird das verschlüsselte Passwort mit dem unter der E-Mail registrierten Passwortes verglichen, falls Sie übereinstimmen wird die Authentifikation gültig, die Userdaten werden in der Browser Session abgelegt und man wird auf das Dashboard weitergeleitet.

Hier ist die Funktion Schritt für Schritt mit dem dazu gehörenden Code dargestellt:

Zur Eingabe verwende ich dieses Formular:

```
<form class="f" method="POST">
  <label for="email" class="label">Email:*</label>
  <input type="email" name="email" class="input" required>
  <br>
  <label for="pw" class="label">Passwort:*</label>
  <input type="password" name="pw" class="input" required>
  <br>
  <input type="submit" name="login" class="senden" value="login">
  <br>
  <br>
  <a href="reg.php" class="abbrechen">Noch nicht registriert?</a>
</form>
```

Falls die E-Mail registriert ist, wird das Erstellungsdatum des Users als Salt für die MD5 Verschlüsselung des eingegebenen Passwortes verwendet.(Siehe Quelle: MD5)

```
// get all data where entered email = registerd email
$result = $mysqli -> query("SELECT * FROM user WHERE email='".$$_POST['email']."'");

// if email is registred
if($row = mysqli_fetch_assoc($result))
{
  // enript password with time of creation
  $salt = $row['createtime'];
  $spw = md5($salt . $_POST['pw']);
}
```

Danach überprüfe ich, ob das verschlüsselte Passwort mit dem registrierten Passwort übereinstimmt. Falls die Passwörter übereinstimmen, wird die Authentifikation gültig und die Userdaten werden in der Session gespeichert.

```
// if encrypted password matches registerd password set authentication to true and save user data to session
if ($row['pw'] == $spw)
{
  $_SESSION['authentication'] = "true";
  $_SESSION['uid'] = $row['uid'];
  $_SESSION['uemail'] = $row['email'];
  header("location: index.php");
}
else
// if they dont match authentication becomes false and output error
{
  $_SESSION['authentication'] = "false";
  echo"<div class='error'>
```

## Registrieren Funktion

Die Registrieren-Funktion kann neue User erstellen und gibt die Authentifikation zurück. Nach der Eingabe wird geprüft, ob die E-Mail schon registriert ist. Falls ja, wird der entsprechende Error ausgegeben. Falls die E-Mail noch nicht registriert ist, werden die beiden eingegebenen Passwörter miteinander abgeglichen. Wenn sie übereinstimmen, wird ein neuer User, ohne das Passwort in der Datenbank erstellt, um einen Erstellungszeitpunkt zu generieren. Anschliessend wird das eingegebene Passwort mit der eben generierten Erstellungszeit verschlüsselt und dem neuen User hinzugefügt. Zum Schluss wird die Authentifikation dann gültig, die neuen Userdaten werden in der Browser Session abgelegt und man wird auf das Dashboard weitergeleitet.

Hier ist die Funktion Schritt für Schritt mit dem dazu gehörenden Code dargestellt:

Zur Eingabe verwende ich dieses Formular:

```
<form class="f" method="POST">
  <div class="elabel">Email:*</div>
  <input type="Email" name="email" class="email">
  <br>
  <div class="pwlabel">Passwort:*</div>
  <input type="password" name="pw" class="pw">
  <br>
  <div class="pwlabel">Passwort wiederholen:*</div>
  <input type="password" name="pw1" class="pw">
  <br>
  <input type="submit" name="senden" class="senden" value="senden">
  <br>
  <br>
  <a href="login.php" class="abbrechen">Bereits registriert?</a>
</form>
```

Zunächst überprüfe ich, ob die E-Mail bereits registriert ist, sonst gibt es einen Error.

```
// check if email is registerd
$check = $mysqli -> query("SELECT * FROM user WHERE email='".$$_POST['email']."'");
if($row = mysqli_fetch_assoc($check))
{
  echo"<div class='error'>
  Email bereits registriert
  </div>";
}
```

Überprüfen, ob die beiden Passwörter aus der Eingabe übereinstimmen

```
// check if the password entered match
elseif($_POST['pw']!= $_POST['pw1'])
{
```

Wenn die E-Mail noch nicht registriert ist, die Passwörter übereinstimmen und das Passwort länger als 8 Zeichen lang ist, wird ein neuer User erstellt. (Siehe Quelle: Stinglength)

```
elseif($_POST['pw']==$_POST['pw1'])
{
    //check if pw is shorter then 8 characters
    if(strlen($_POST['pw1'])<8)
    {
        echo"<div class='error'>
        Ein Sicheres Passwort sollte länger als 8 Zeichen sein
        </div>";
    }

    else
    {
        // get current time
        $time = date("y.m.d H.i.s");
        session_start();

        // add new user with entered values
        $mysqli->query("INSERT INTO `user` (`email`, `createtime`, `changetime`) VALUES
        ('".$_POST['email'].",".$time.", ".$time.")");
    }
}
```

Anschließend wird das eingegebene Passwort mit der Erstellungszeit des Users als Salt für die MD5 Verschlüsselung verwendet und der neue User wird mit dem verschlüsselten Passwort upgedatet und man wird eingeloggt.

```
$gt = $mysqli->query("SELECT * from user ORDER BY createtime DESC LIMIT 1");
$gggt = $gt->fetch_assoc();

// encrypt entered password using changetime
$salt = $gggt['changetime'];
$spw = md5($salt . $_POST['pw']);

// update new user with encrypted password
$mysqli->query("UPDATE `user` SET `pw`='".$spw.'" WHERE email='".$_POST['email']."'");

// return authentication true and save user data in browser session then redirect to dashboard
$_SESSION['authentication'] = "true";
$_SESSION['uid'] = $gggt['uid'];
$_SESSION['uemail'] = $gggt['email'];
header("location: index.php");
}
```

Wenn die Passwörter nicht übereinstimmen gibt es einen Error

```
// if passwords do not match error
else
{
    echo"<div class='error'>
    Passwörter stimmen nicht überein
    </div>";
}
```



## Tool hinzufügen Funktion

Mit der Tool hinzufügen Funktion kann ein User seine eigenen Tools erstellen. Nach der Eingabe wird überprüft, ob der eingegebene Name und der eingegebene Link unter 100 Zeichen lang sind. Falls ja, wird überprüft, ob im Link der String «https://» vorhanden ist. Falls ja, wird das neue Tool mit den eingegebenen Daten erstellt. Falls nein wird vor dem link noch ein «https://» angehängt und das Tool wird dann erstellt. Anschliessend wird geprüft, ob im Formular ein Bild mitgesendet wurde, falls ja, wird es mit einem neuen Namen korrespondierend zu der Tool ID im Assets Ordner abgelegt. Zum Schluss gibt es einen Eintrag in den Verlauf das ein Tool erstellt wurde und man wird auf das Dashboard weitergeleitet.

Hier ist die Funktion Schritt für Schritt mit dem dazu gehörenden Code dargestellt:

Zur Eingabe verwende ich dieses Formular:

```
<form class="f" method="POST" enctype="multipart/form-data">
  <label for="name" class="label">Name:</label>
  <input type="text" name="name" class="input" required>
  <br>
  <label for="text" class="label">Link:</label>
  <input type="text" name="link" class="input" required>
  <br>
  <div class="label">Bild:</div>
  <input type="file" class="fileinput" name="fileToUpload" accept=".jpg, .jpeg, .png">
  <br>
  <input type="submit" name="senden" class="senden" value="senden">
  <br>
  <a class="ab" href="index.php">Abbrechen</a>
</form>
```

Nach der Eingabe überprüfe ich, ob der Link und Name über 100 Zeichen lang ist.

```
// if the user presses senden
if (isset($_POST['senden'])) {
  // check if name is more then 100 caracters
  if(strlen($_POST['name'])>100)
  {
    // error if length is more then 100 caracters
    echo"<div class='error'>

    Der Name ist zu lange maximal 100 Zeichen
    </div>";
  }
  // check if link is more then 100 caracters
  elseif (strlen($_POST['link'])>100) {
    // error if length is more then 100 caracters
    echo"<div class='error'>

    Der Link ist zu lange maximal 100 Zeichen
    </div>";
  }
  // if link and name is under 100 caracters
  else{
```

Falls beides nicht zustimmt, überprüfe ich, ob im Link «https://» vorhanden ist und erstelle das Tool je Ergebnis mit angehängtem «https://» oder ohne.

```
// declare what string to search in link
$word = "https://";
$mystring = $_POST['link'];

// Test if string contains the word
if(strpos($mystring, $word) !== false)
{
    // if the link contains the word create new tool with entered attributes
    $mysqli->query("INSERT INTO `tools`(`name`,`link`,`uid`,`createtime`) VALUES ('" . $_POST['name'] . "','" .
$_POST['link'] . "','" . $_SESSION['uid'] . "','" . $time . "')");
}
else
{
    // if the link does not contain the word create new tool with the entered attributes but add "https://" in front of the link
    $mysqli->query("INSERT INTO `tools`(`name`,`link`,`uid`,`createtime`) VALUES ('" . $_POST['name'] . "','" . 'https://' .
$_POST['link'] . "','" . $_SESSION['uid'] . "','" . $time . "')");
}
```

Anschliessend überprüfe ich, ob ein Bild im Formular ausgewählt wurde.

```
if($_FILES['fileToUpload']['size'] > 0) {
```

Wen kein Bild ausgewählt wurde, wird man wieder auf das Dashboard verwiesen. Falls ein Bild ausgewählt wurde, wird bestimmt, wo das Bild hochgeladen werden soll.

```
// declare the target director where the file will get uploaded
$target_dir = "assets/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
```

Danach wird überprüft, ob es sich um ein echtes Bild handelt oder nicht.

```
// get the filetype of the uploaded file
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

// Check if image file is a actual image or fake image
$check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
if($check !== false) {
    echo "File is an image - " . $check["mime"] . ".";
    $uploadOk = 1;
} else {
    echo "File is not an image.";
    $uploadOk = 0;
}
```

Wenn es sich um ein Bild handelt, wird ein neuer Name für das Bild generiert und das Bild wird hochgeladen.

```
// create new file name with tool id
$temp = explode(".", $_FILES["fileToUpload"]["name"]);
$newfilename = $target_dir . $m . "." . "png";

// uplad file
if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $newfilename)) {

}

// if uplad fails display error
else
{
    echo"<div class='error'>
    Ein error ist aufgetreten
    </div>";
}
```

Zum Schluss gibt es noch einen Eintrag in den Verlauf und man wird zurück auf das Dashboard verwiesen, wo sich das neue Tool jetzt befindet.

```
// make entry in history that tool was created and redirect to dashboard
$mysqli -> query("INSERT INTO 'history'('uid', 'element', 'changetime', 'changed') VALUES
('".$_SESSION['uid']."','".$row['tid']."','".$time."','".$row['name']." tool erstellt')");
header("location: index.php");
```

## Tool bearbeiten Funktion

Mit der Tool bearbeiten Funktion kann ein User seine eigenen Tools bearbeiten. Nach der Eingabe

Hier ist die Funktion Schritt für Schritt mit dem dazu gehörenden Code dargestellt:

Zur Eingabe verwende ich dieses Formular:

```
// display from with registerd values
echo"
    <form class='f' method='POST' enctype='multipart/form-data'>
        <label for='name' class='label'>Name:</label>
        <input type='text' name='name' class='input' required value='".$_GET['name']."'>
        <br>
        <label for='link' class='label'>Link:</label>
        <input type='text' name='link' class='input' required value='".$_GET['link']."'>
        <br>
        <div class='label'>Bild:</div>
        <input type='file' class='fileinput' name='fileToUpload' accept='.jpg, .jpeg, .png'>
        <br>
        <input type='submit' name='senden' class='senden' value='senden'>
        <br>
    <a href='index.php' class='abbrechen'>Abbrechen</a>
    </form>
"
```

Nach der Eingabe überprüfe ich, ob der Link und Name über 100 Zeichen lang ist.

```
// if the user presses senden
    if(isset($_POST['senden']))
    {
if(strlen($_POST['name'])>100)
{
    // error if length is more then 100 characters
    echo"<div class='error'>

    Der Name ist zu lange maximal 100 Zeichen
    </div>";
}
// check if link is more then 100 characters
elseif (strlen($_POST['link'])>100) {
    // error if length is more then 100 characters
    echo"<div class='error'>

    Der Link ist zu lange maximal 100 Zeichen
    </div>";
}
// if link and name is under 100 characters
else{
```

Falls beides nicht zustimmt, überprüfe ich, ob im link «https://» vorhanden ist und update das Tool je Ergebnis mit angehängtem «https://» oder ohne.

```
// declare what string to search in link
$word = "https://";
$mystring = $_POST['link'];

// Test if string contains the word
if(strpos($mystring, $word) !== false){

// if the link contains the string update tool with entered values
    $mysqli -> query("UPDATE `tools` SET
`name`='".$_POST['name']."', `link`='".$_POST['link']."', `changetime`='".$time.'" WHERE tid='".$_GET['id']."'");

}
// if link does not contain the string
else
{
    // change tool with entered values and "https://" added in front of link
    $result = $mysqli -> query("UPDATE `tools` SET
`name`='".$_POST['name']."', `link`='https://".$_POST['link']."', `changetime`='".$time.'" WHERE tid='".$_GET['id']."'");
}
```

Anschliessend überprüfe ich, ob ein Bild im Formular ausgewählt wurde.

```
if($_FILES['fileToUpload']['size'] > 0) {
```

Wenn kein Bild ausgewählt wurde, wird man wieder auf das Dashboard verwiesen. Falls ein Bild ausgewählt wurde, wird bestimmt, wo das Bild hochgeladen werden soll.

```
// declare the target director where the file will get uploaded
$target_dir = "assets/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
```

Danach wird überprüft, ob es sich um ein echtes Bild handelt oder nicht.

```
// get the filetype of the uploaded file
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

// Check if image file is a actual image or fake image
$check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
if($check !== false) {
    echo "File is an image - " . $check["mime"] . " ";
    $uploadOk = 1;
} else {
    echo "File is not an image.";
    $uploadOk = 0;
}
```

Wenn es sich um ein Bild handelt, wird ein neuer Name für das Bild generiert und das Bild wird hochgeladen

```
// create new file name with tool id
$temp = explode(".", $_FILES["fileToUpload"]["name"]);
$newfilename = $target_dir . $m . "." . "png";

// uplad file
if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $newfilename)) {

}

// if uplad fails display error
else
{
    echo"<div class='error'>
    Ein error ist aufgetreten
    </div>";
}
```

Zum Schluss gibt es noch einen Eintrag in den Verlauf und man wird zurück auf das Dashboard verwiesen, wo sich das Tool befindet.

```
// make entry in history that tool was created and redirect to dashboard
$mysqli->query("INSERT INTO 'history' ( 'uid', 'element', 'changetime', 'changed' ) VALUES
('".$_SESSION['uid']."','".$row['tid']."','".$time."','".$row['name']." tool bearbeiten')");
header("location: index.php");
```

## Tool löschen Funktion

Die Tool löschen Funktion ermöglicht es einem User seine Tools zu löschen und dabei auch gleichzeitig das dazugehörige Bild aus dem Assetsordner zu entfernen.

Hier ist die Funktion Schritt für Schritt mit dem dazu gehörenden Code dargestellt:

Dieses Formular verwende ich zur Bestätigung:

```
<form method="POST" class="f">

    <label class="labeö">Bist du sicher?</label>
    <br>
    <input class="senden" type="submit" name="ja" id="ja" value="ja">

    <input class="senden" type="submit" name="abbrechen" id="abbrechen" value="abbrechen">

</form>
```

Falls der User abbrechen drückt, wird er auf das Dashboard zurück geleitet.

```
// if the user presses abbrechen redirect to Dashboard
elseif (isset($_POST['abbrechen'])) {

    header( "Location: index.php" );

}
```

Falls er ja drückt, wird das Tool aus der Datenbank gelöscht.

```
// if the user presses ja
if (isset($_POST['ja'])) {

    // delete tool with tool id from link
    $mysqli->query("DELETE FROM `tools` WHERE tid='".$_GET['id']."'");
```

Anschliessend wird überprüft, ob es zu dem Tool ein Bild im Assets Ordner gibt.

```
// check if tool had a img
$check = file_exists("assets/".$_GET['id'].".png");
```

Falls ja, wird das Bild aus dem Assets-Ordner entfernt.

```
// if the file exists delete file
if ($check==true) {

    unlink("assets/".$_GET['id'].".png");

}
```

Zum Schluss wird noch ein Eintrag in den Verlauf gemacht, dass ein Tool gelöscht wurde und man wird auf das Dashboard weitergeleitet, wo das Tool jetzt nicht mehr vorhanden ist.

```
// enty in history that tool was deleted then redirect to dashboard
$mysqli->query("INSERT INTO `history`(`uid`, `element`, `changetime`, `changed`) VALUES ('".$_SESSION['uid']."','".$_GET['id']."','".$time."','".$_GET['name']." tool gelöscht')");
header("location: index.php");
```

## Bild hochladen Funktion

Die Bild hochladen Funktion ermittelt die Tool-ID und ändert den Dateinamen zu Tool-ID.png, um die Bilder den Tools zuordnen zu können. Zunächst wird überprüft, ob im Formular ein Bild ausgewählt wurde, falls ja, wird dann überprüft, ob es sich um ein richtiges Bild handelt oder nicht. Falls es sich um ein echtes Bild handelt, wird die Datei mit dem neuen Namen im Assets Ordner abgelegt.

Hier ist die Funktion Schritt für Schritt mit dem dazu gehörenden Code dargestellt:

Zuerst überprüfe ich, ob im Formular eine Datei ausgewählt wurde.

```
// check if a file was submitted in the form
if($_FILES['fileToUpload']['size'] > 0) {
```

Danach deklarieren ich die Variabel, die für den neuen Dateinamen verwendet wird.

```
// declare and initialize variable with the tool id to change the name of the uploaded file
$rn=$_row['tid'];
```

Anschliessend bestimme ich wo die Datei hochgeladen werden soll.

```
// declare the target director wher the file will get uploaded
$target_dir = "assets/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
```

Zunächst überprüfe ich, ob es sich um ein echtes Bild handelt.

```
// get the filetype of the uploaded file
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

// Check if image file is a actual image or fake image
$check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
if($check !== false) {
    echo "File is an image - " . $check["mime"] . ".";
    $uploadOk = 1;
} else {
    echo "File is not an image.";
    $uploadOk = 0;
}
```

Falls ja, wird ein neuer Name für die Datei mit der Tool-ID erstellt und die Datei wird hochgeladen.

```
// create new file name with tool id
$temp = explode(".", $_FILES["fileToUpload"]["name"]);
$newfilename = $target_dir . $rn . "." . "png";

// uplad file
if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $newfilename)) {
```

## Dashboard

Im Dashboard werden alle Tools eines Users dargestellt. Es wird für jedes fünfte Tool eine neue Reihe generiert, sodass immer vier in einer Reihe dargestellt werden. Es wird überprüft, ob ein Tool ein dazu passendes Bild abgelegt hat und falls nicht, wird das Tool mit einem stand Art Bild dargestellt.

Hier ist die Funktion Schritt für Schritt mit dem dazu gehörenden Code dargestellt:

Die erste Reihe wird ausserhalb des PHP Codes geöffnet.

```
<div class="dashboard">

    <div class="row">

        <?php
```

Dann wird im PHP eine Abfrage formuliert, die alle Tools des Users ausgibt

```
// get all data from tool with uid from the logged in user
$result = $mysqli -> query("SELECT * FROM tools WHERE uid='".$$_SESSION['uid']."'");
```

Anschliessend öffne ich eine while schleife, die für jeden Eintrag aus der abfrage einen Durchlauf macht.

```
// for every tool with the uid from logged in user
while ($row = $result->fetch_assoc()) {
```



In der While befindet sich dann eine if die jedes Mal wenn mein Counter 4 erreicht, die Reihe schliesst und eine neue öffnet und dann den Counter wieder auf 0 stellt. Der Counter wird ausserhalb der While deklariert und auf 0 gesetzt.

```
// if counter is equal to 4 end row and start new one
if ($i==4)
{
    echo"</div><div class='row'>";
    // reset the counter
    $i=0;
}
// counter + 1
$i++;
```

Zum Schluss wird dann je nachdem, ob ein Bild existiert oder nicht eine der beiden Echos ausgegeben:

```
// if the file exists
if($check==true)
{
    // display tool
    echo"
    <div class='tool'>

    <div class='name'>
        ".$row['name'].
    </div>

    <div class='pictures'>
        <a class='imglink' href='".$row['link']."' target='_blank'><img class='img' src='assets/".$row['tid'].".png'></a>
    </div>
    <div class='buttons'>
        <a class='blink' href='edit.php?id=".$row['tid']."&name=".$row['name']."&link=".$row['link']."'><img class='buttonimg'
src='assets/edit.png'></a>
        <a class='blink' href='del.php?id=".$row['tid']."&name=".$row['name']."&link=".$row['link']."'><img class='buttonimg'
src='assets/delete.png'></a>
    </div>
    </div>
    ";
}
// display tool with default img
else
{
    echo"
    <div class='tool'>

    <div class='name'>
        ".$row['name'].
    </div>

    <div class='pictures'>
        <a class='imglink' href='".$row['link']."' target='_blank'><img class='img' src='assets/def.png'></a>
    </div>
```

```
<div class='buttons'>

    <a class='blink' href='edit.php?id=".$row["tid"]."&name=".$row["name"]."&link=".$row["link"]."'"><img class='buttonimg'
src='assets/edit.png'></a>

    <a class='blink' href='del.php?id=".$row["tid"]."&name=".$row["name"]."&link=".$row["link"]."'"><img class='buttonimg'
src='assets/delete.png'></a>

</div>

</div>

";
}
```

## Layout & Design

Es gibt für die Webapplikation 2 verschiedene Layouts, einmal für die Index-Seite(Dashboard) und für den Verlauf und ein zweites für all die Funktionen, wobei der Kopf für beide Layouts gleich bleibt.

Hier ist das Layout für den Index und Verlauf.

```
<div class="container">  
  <div class="head">  
    <div class="logo">  
      <a href="index.php"></a>  
    </div>  
    <div class="title">  
      <?php  
        echo"&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&$_SESSION['uemail']"."s Dashboard";  
      ?>  
    </div>  
  </div>  
  <div class="sidenav">  
    <nav>  
      <br>  
      <div class="navlinkwrap"><a class="navlink" href="logout.php">&nbsp;&nbsp;Ausloggen</a></div>  
      <div class="navlinkwrap"><a class="navlink" href="create.php">&nbsp;&nbsp;Tool hinzufügen</a></div>  
      <div class="navlinkwrap"><a class="navlink" href="history.php">&nbsp;&nbsp;Verlauf</a></div>  
    </nav>  
  </div>  
  <div class="dashboard">  
    </div>
```

Hier ist das Layout für die Funktionen.

[illegible]

## Plausible Eingaben & Benutzerfreundlichkeit

Ich habe dafür gesorgt, dass alle Pflichtfelder eine Markierung haben. Davon gibt es in der Applikation 7 Stück. E-Mail und Passwort auf der Login Seite, E-Mail, Passwort und Passwort wiederholen auf der Registrieren-Seite, Tool Name und Link auf der Tool hinzufügen Seite. Alle anderen Eingaben sind optional und somit auch nicht markiert. Für jeden Fehler, den ein User bei den Eingaben machen kann, wird ein dazu passender Error ausgegeben, der klarmacht, was verändert werden muss. Ausserdem gibt es immer die Möglichkeit die Funktion abubrechen bzw. zurück zum Dashboard zu gelangen oder bei der Login- und Registrierungsfunktion zu dem jeweiligen Gegenstück zu gelangen.

Hier sind die möglichen Errors der Funktionen aufgelistet:

### *login.php*

Failed to connect to MySQL

Email noch nicht registriert

Inkorrektes Passwort

### *reg.php*

Failed to connect to MySQL

Email bereits registriert

Ein Sicheres Passwort sollte länger als 8 Zeichen sein

Passwörter stimmen nicht überein

### *create.php*

Failed to connect to MySQL

Der Name ist zu lange maximal 100 Zeichen

Der Link ist zu lange maximal 100 Zeichen

Ein error ist aufgetrete

### *edit.php*

Failed to connect to MySQL

Der Name ist zu lange maximal 100 Zeichen

Der Link ist zu lange maximal 100 Zeichen

Ein error ist aufgetreten

### *del.php*

Failed to connect to MySQL

### *index.php*

Failed to connect to MySQL

### *histpry.php*

Failed to connect to MySQL

## Redundanz entfernt

Wie erwartet blieben meine Funktionen nicht vor redundanten code verschont.

Mir ist bei der Überprüfung ist mir aufgefallen, dass bei der Tool erstellen Funktion und der Tool bearbeiten Funktion jeweils die Bild hochladen Funktion doppelt vorhanden war. Vorher wurde, nachdem geprüft wurde ob «https://» im Link vorhanden ist zweimal derselbe code ausgeführt, mit dem kleinen Unterschied das je nachdem ob «https://» im link vorhanden ist eine andere SQL-Query durchgeführt wurde. Jetzt wird alles ausser der SQL-Query nach dem if durchgeführt, was bedeutet, dass ich die Funktion fast halbieren konnte.

Vorher:

```
// Test if string contains the word
if(strpos($mystring, $word) !== false)
{
    // if the link contains the word create new tool with entered attributes
    $mysqli->query("INSERT INTO `tools`(`name`, `link`, `uid`, `createtime`) VALUES ('" . $_POST['name'] . "','" . $_POST['link'] . "','" . $_SESSION['uid'] . "','" . $time . "')");

    // get all data from the last tool created
    $result = $mysqli->query("SELECT * FROM tools ORDER BY tid DESC LIMIT 1;");
    $row = $result->fetch_assoc();

    // check if a file was submitted in the form
    if($_FILES['fileToUpload']['size'] > 0)
    {
    }

    else
    {
        // if the link does not contain the word create new tool with the entered attributes but add "https://" in front of the link
        $mysqli->query("INSERT INTO `tools`(`name`, `link`, `uid`, `createtime`) VALUES ('" . $_POST['name'] . "','" . 'https://' . $_POST['link'] . "','" . $_SESSION['uid'] . "','" . $time . "')");

        // get all data from the last tool created
        $result = $mysqli->query("SELECT * FROM tools ORDER BY tid DESC LIMIT 1;");
        $row = $result->fetch_assoc();

        // check if a file was submitted in the form
        if($_FILES['fileToUpload']['size'] > 0)
        {
        }
    }
}
```

Nacher:

```
// Test if string contains the word
if(strpos($mystring, $word) !== false)
{
    // if the link contains the word create new tool with entered attributes
    $mysqli->query("INSERT INTO `tools`(`name`, `link`, `uid`, `createtime`) VALUES ('" . $_POST['name'] . "','" . $_POST['link'] . "','" . $_SESSION['uid'] . "','" . $time . "')");
}

else
{
    // if the link does not contain the word create new tool with the entered attributes but add "https://" in front of the link
    $mysqli->query("INSERT INTO `tools`(`name`, `link`, `uid`, `createtime`) VALUES ('" . $_POST['name'] . "','" . 'https://' . $_POST['link'] . "','" . $_SESSION['uid'] . "','" . $time . "')");
}

// get all data from the last tool created
$result = $mysqli->query("SELECT * FROM tools ORDER BY tid DESC LIMIT 1;");
$row = $result->fetch_assoc();

if($_FILES['fileToUpload']['size'] > 0)
{
}
```

## Fehlerprüfung und Lösungsanwendung

Bei der Fehlerprüfung habe ich bemerkt, dass ich bei der Tool bearbeiten Funktion die falsche variabel verwendet habe, um an die Tool-ID des zuletzt veränderten Tools zu gelangen. Das hatte zur Folge, dass das Bild immer mit der Tool-ID des zuletzt erstellten Tools hochgeladen wird.

Vorher:

```
// get all data from last changed tool  
$result = $mysqli -> query("SELECT * FROM tools ORDER BY createtime DESC LIMIT 1;");
```

Nacher:

```
// get all data from last changed tool  
$result = $mysqli -> query("SELECT * FROM tools ORDER BY changetime DESC LIMIT 1;");
```

Ausserdem ist mir ein Fehler aufgefallen, der sich durch jedes Formular der Website gezogen hat. Jede meiner «abbrechen» Buttons war ein Input Type submit und haben damit das Formular ausgeführt. Ich habe dies in login.php, reg.php, create.php und edit.php wie unten gezeigt angepasst.

Vorher:

```
<br>  
<input class="senden" type="submit" name="ja" value="ja">  
<input class="senden" type="submit" name="abbrechen" id="abbrechen" value="abbrechen">  
</form>
```

Nacher:

```
<input type="submit" name="senden" class="senden" value="senden">  
<br>  
<br>  
<a href="login.php" class="abbrechen">Bereits registriert?</a>  
</form>
```

## Kontrollieren

### Testprotokoll

Um die Tests durchzuführen habe ich Chrome beendet und neu gestartet, um die Session zu beenden und in phpmyadmin die Datenbank dashboarddb von allen Datensätzen befreit, damit unvollständige Datensätze aus der Entwicklung nicht zu unerwarteten Fehlern führen.

Test	Erwartetes Ergebnis	Ergebnis	Bemerkung	Test Nr.	Zeitpunkt und testende Person
Ohne Login versuchen auf index.php zu gelangen durch das Ändern der URL zu localhost/Dashboard/index.php	Sofortige Weiterleitung auf login.php	Sofortige Weiterleitung auf login.php	-	1	21.03.2022 08:45 Uhr Shannon Neil Schürch
Ohne Login versuchen auf del.php zu gelangen durch das Ändern der URL zu localhost/Dashoard/del.php	Sofortige Weiterleitung auf login.php	Sofortige Weiterleitung auf login.php	-	2	21.03.2022 08:55 Uhr Shannon Neil Schürch
Ohne Login versuchen auf edit.php zu gelangen durch das Ändern der URL zu localhost/Dashoard/edit.php	Sofortige Weiterleitung auf login.php	Sofortige Weiterleitung auf login.php	-	3	21.03.2022 08:57 Uhr Shannon Neil Schürch
Ohne Login versuchen auf create.php zu gelangen durch das Ändern der URL zu localhost/Dashoard/create.php	Sofortige Weiterleitung auf login.php	Sofortige Weiterleitung auf login.php	-	4	21.03.2022 08:58 Uhr Shannon Neil Schürch
In der Tool erstellen Funktion auf abbrechen drücken	Weiterleitung zum Dashboard	Weiterleitung zum Dashboard	-	5	21.03.2022 09:01 Uhr Shannon Neil Schürch
Tool erstellen ohne Namen aber mit link und Bild	Meldung, dass Name ein Pflichtfeld ist	Meldung, dass Name ein Pflichtfeld ist	-	6	21.03.2022 09:04 Uhr Shannon Neil Schürch
Tool erstellen ohne link aber mit Namen und Bild	Meldung das Link ein Pflichtfeld ist	Meldung das Link ein Pflichtfeld ist		7	21.03.2022 09:05 Uhr Shannon Neil Schürch
Tool erstellen ohne ein Bild aber mit Namen und link	Tool wird ohne Bild erstellt, Weiterleitung auf Dashboard	Tool wird ohne Bild erstellt, Weiterleitung auf Dashboard	Im Dashboard werden dann wie geplant anstatt der Bilder die Standardbilder dargestellt	8	21.03.2022 09:08 Uhr Shannon Neil Schürch
Tool mit allen Attributen erstellen	Tool wird mit eingegebenen Daten und ausgewähltem Bild erstellt	Tool wird mit eingegebenen Daten und ausgewähltem Bild erstellt	-	9	21.03.2022 09:10 Uhr Shannon Neil Schürch

<b>In der Tool bearbeiten Funktion auf abbrechen drücken</b>	Zurück auf das Dashboard weiterleiten	Zurück auf das Dashboard weiterleiten	Ohne die Fehleranwendung wäre dieser Test negativ ausgefallen	10	21.03.2022 09:12 Uhr Shannon Neil Schürch
<b>Tool bearbeiten ohne Namen aber mit link und Bild</b>	Tool wird mit dem vorherigen Namen und neuem Link und neuem Bild upgedatet	Tool wird mit dem vorherigen Namen und neuem Link und neuem Bild upgedatet	-	11	21.03.2022 09:15 Uhr Shannon Neil Schürch
<b>Tool bearbeiten ohne link aber mit Namen und Bild</b>	Tool wird mit dem vorherigen Link und neuen Namen und neuen Bild upgedatet	Tool wird mit dem vorherigen Link und neuen Namen und neuen Bild upgedatet	-	12	21.03.2022 09:20 Uhr Shannon Neil Schürch
<b>Tool bearbeiten ohne Bild aber mit Namen und link</b>	Tool Name und link werden upgedatet und das Bild verbleibt unberührt	Tool Name und link werden upgedatet und das Bild verbleibt unberührt	-	13	21.03.2022 09:25: Uhr Shannon Neil Schürch
<b>In der Tool löschen Funktion auf abbrechen drücken</b>	Zurück auf Startseite weiterleiten	Zurück auf Startseite weiterleiten	Ohne die Fehleranwendung wäre dieser Test negativ ausgefallen	14	21.03.2022 09:28 Uhr Shannon Neil Schürch
<b>In der Tool löschen Funktion auf Ja drücken</b>	Tool wird aus der Datenbank gelöscht und das dazugehörige Bild wird falls vorhanden gelöscht	Tool wird aus der Datenbank gelöscht und das dazugehörige Bild wird falls vorhanden gelöscht	-	15	21.03.2022 09:33 Uhr Shannon Neil Schürch
<b>Auf ausloggen drücken</b>	Authentifikation wird ungültig und auf Login Seite weiterleiten	Authentifikation wird nicht ungültig und auf Login Seite weiterleiten	.	16	21.03.2022 09:40 Uhr Shannon Neil Schürch
<b>Etwas unter E-Mail im Login eingeben das nicht ein @ und einen . beinhaltet</b>	Meldung das, dass keine gültige E-Mail ist	Meldung das, dass keine gültige E-Mail ist	-	17	21.03.2022 09:43 Uhr Shannon Neil Schürch
<b>Eine noch nicht registrierte E-Mail im Login eingeben</b>	Meldung das E-Mail noch nicht registriert ist	Meldung das E-Mail noch nicht registriert ist	-	18	21.03.2022 09:45 Uhr Shannon Neil Schürch



<b>Falsches Passwort mit registrierter E-Mail im Login eingeben</b>	Meldung das Passwort ungültig ist	Meldung das Passwort ungültig ist	-	19	21.03.2022 09:50 Uhr Shannon Neil Schürch
<b>Richtiges Passwort mit registrierter E-Mail im Login eingeben</b>	Authentifikation wird gültig Weiterleitung auf Startseite	Authentifikation wird gültig Weiterleitung auf Startseite	-	20	21.03.2022 09:51 Uhr Shannon Neil Schürch
<b>Im Login auf Registrieren drücken</b>	Weiterleitung auf registrieren Seite	Weiterleitung auf registrieren Seite	Ohne die Fehleranwendung wäre dieser Test negativ ausgefallen	21	21.03.2022 09:53 Uhr Shannon Neil Schürch
<b>Bereits registrierte E-Mail bei Registrierung eingeben</b>	Meldung, dass die E-Mail noch nicht registriert ist	Meldung, dass die E-Mail noch nicht registriert ist		22	21.03.2022 09:55 Uhr Shannon Neil Schürch
<b>Etwas unter E-Mail bei der Registrierung eingeben, das nicht ein @ und einen . beinhaltet</b>	Meldung, dass das keine gültige E-Mail ist	Meldung, dass das keine gültige E-Mail ist		23	21.03.2022 09:58 Uhr Shannon Neil Schürch
<b>Zwei verschiedene Passwörter bei der Registrierung angeben</b>	Meldung, dass die Passwörter nicht übereinstimmen	Meldung, dass die Passwörter nicht übereinstimmen	-	24	21.03.2022 10:00 Uhr Shannon Neil Schürch
<b>Nicht registrierte E-Mail und zwei übereinstimmende Passwörter bei Registrierung angeben</b>	Neuer User wird erstellt, Login erfolgt und Weiterleitung auf Startseite	Neuer User wird erstellt, Login erfolgt und Weiterleitung auf Startseite	-	25	21.03.2022 10:02 Uhr Shannon Neil Schürch

Abbildung 21: Testprotokoll

## Testbericht

Dank der Fehlerbehandlung verliefen alle Tests positiv. Die Applikation verhält sich genau wie erwartet. Alles hat sofort geladen und die ganze Erfahrung hat sich allgemein sehr flüssig angefühlt.

## Reflexion

In der ersten Phase der IPERKA-Mehode dem Informieren verlief alles reibungslos.

Das Planen verlief auch ohne Komplikationen, ich habe allerdings bemerkt, dass die Dokumentation mehr Aufwand verlangt als ich ursprünglich erwartet habe.

Beim Realisieren bin ich ein wenig vom Plan abgewichen, hatte aber keine grösseren Komplikationen.

Das Kontrollieren verlief reibungslos, da ich beim Realisieren schon sämtliche Fehler ausmerzen konnte.

In zukünftigen Projekten werde ich mehr Zeit für die Dokumentation einplanen, da ich den dafür benötigten Aufwand unterschätzt habe. Zum Glück hatte ich deswegen niemals Zeitnot und konnte rechtzeitig mit allem fertig werden.

Im nächsten Projekt werde ich die Planung genauer gestalten. In diesem Projekt musste ich von meinem Plan abweichen und ich habe vor, dass dies in Zukunft nicht mehr passiert.

Alles in allem bin ich mit dem Ergebnis sehr zufrieden und ich bin glücklich darüber, dass ich ein Projekt von der Wiege bis ins Grab selbständig realisieren durfte. Ich habe einiges, vor allem im Bereich der Planung, gelernt.

## Glossar

**PHP SESSION:** Eine PHP SESSION ist ein Array, in der verschiedene Session-Variablen gespeichert werden können.

**SESSION VARIABLE:** Eine Session variabel ist eine Variabel der PHP Session und ermöglicht es Informationen auf allen offenen Seiten eines Browsers zu verwenden. Ich speichere darin unter anderem, wer gerade angemeldet ist und ob die Authentifikation erfolgt ist.

## Abbildungsverzeichnis

Abbildung 1: Zeitplan.....	10
Abbildung 2: Arbeitsjournal vom 07.03.2022 .....	11
Abbildung 3: Arbeitsjournal vom 08.03.2022 .....	11
Abbildung 4: Arbeitsjournal vom 09.03.2022 .....	12
Abbildung 5: Arbeitsjournal vom 10.03.2022 .....	13
Abbildung 6: Arbeitsjournal vom 14.03.2022 .....	13
Abbildung 7: Arbeitsjournal vom 15.03.2022 .....	14
Abbildung 8: Arbeitsjournal vom 16.03.2022 .....	14
Abbildung 9: Arbeitsjournal vom 17.03.2022 .....	15
Abbildung 10: Arbeitsjournal vom 21.03.2022 .....	15
Abbildung 11: Arbeitsjournal vom 22.03.2022 .....	15
Abbildung 12: Grober Applikationsentwurf.....	19
Abbildung 13: Datenbank modell.....	20
Abbildung 14: Login funktion NassiSchneidermann.....	21
Abbildung 15: Registrieren funktion NassiSchneidermann .....	22
Abbildung 16: Ausloggen Funktion NassiSchneidermann.....	23
Abbildung 17: Tool erstellen Funktion NassiSchneidermann .....	24
Abbildung 18: Tool bearbeiten Funktion NassiSchneidermann.....	25
Abbildung 19: Tool löschen Funktion NassiSchneidermann .....	26
Abbildung 20: Testfälle .....	27
Abbildung 21: Testprotokoll .....	<b>Error! Bookmark not defined.</b>

# Quellenverzeichnis

## **Aufgabenstellung / Bewertungskriterien:**

<https://2022.pkorg.ch/overview>

## **Verpflichtende Termine / Leitfaden:**

<https://2022.pkorg.ch/dokument/1033/view>

## **MD5:**

[https://www.w3schools.com/php/func\\_string\\_md5.asp](https://www.w3schools.com/php/func_string_md5.asp)

<https://infosecscout.com/md5-salt-hash/>

## **String Length:**

[https://www.w3schools.com/php/func\\_string\\_strlen.asp](https://www.w3schools.com/php/func_string_strlen.asp)

## **Andere PHP und HTML Elemente:**

[www.w3schools.com](http://www.w3schools.com)





```

<div class="elabel">Email:*</div>
<input type="Email" name="email" class="email">
<br>
<div class="pwlabel">Passwort:*</div>
<input type="password" name="pw" class="pw">
<br>
<div class="pwlabel">Passwort wiederholen:*</div>
<input type="password" name="pw1" class="pw">
<br>
<input type="submit" name="senden" class="senden" value="senden">
<br>
<br>
<a href="login.php" class="abbrechen">Bereits registriert?</a>
</form>

<?php
// funtion to register adds new users and returns authentication and user data
if (isset($_POST['senden'])) {

    // connect to DB
    include "config.php";

    $mysqli = new mysqli($host, $user, $dbpw, $db);

    if ($mysqli -> connect_errno) {
        echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
        exit();
    }

    // check if email is registerd
    $check = $mysqli -> query("SELECT * FROM user WHERE email='".$_POST['email']."'");
    if($row = mysqli_fetch_assoc($check))
    {
        echo "<div class='error'>
        Email bereits registriert
        </div>";
    }

    // check if the password entered match
    elseif($_POST['pw']==$_POST['pw1'])
    {
        //check if pw is shorter then 8 characters
        if(strlen($_POST['pw1'])<8)
        {
            echo "<div class='error'>
            Ein Sicheres Passwort sollte länger als 8 Zeichen sein
            </div>";
        }

        else
        {
            // get current time
            $time = date("y.m.d H.i.s");
            session_start();

            // add new user with entered values
            $mysqli -> query("INSERT INTO `user` (`email`, `createtime`, `changetime`) VALUES
            ('".$_POST['email']."','".$time."','".$time."')");

            $gt = $mysqli -> query("SELECT * from user ORDER BY createtime DESC LIMIT 1");
            $ggt = $gt->fetch_assoc();

            // encrypt entered password using changetime
            $salt = $ggt['changetime'];
            $spw = md5($salt . $_POST['pw']);

            // update new user with encrypted password
            $mysqli -> query("UPDATE `user` SET `pw`='".$_spw.'" WHERE email='".$_POST['email']."'");

            // return authentication ture and save user data in browser session then redirect to dashboard
            $_SESSION['authentication'] = "true";
            $_SESSION['uid'] = $ggt['uid'];
            $_SESSION['uemail'] = $ggt['email'];
            header("location: index.php");
        }
    }
}

```

```

        // if passwords do not match error
        else
        {
            echo"<div class='error'>
                Passwörter stimmen nicht überein
            </div>";
        }
    }

    // if user presses login redirect to login
    elseif (isset($_POST['login']))
    {
        header("location: login.php");
    }

    ?>

</div>
</div>

</div>

</body>
</html>

```

### create.php

```

<!DOCTYPE html>
<html>

<?php

    // give history access to the sql config file
    include "config.php";

    // create new mysqli object
    $mysqli = new mysqli($host, $user, $dbpw, $db);

    // start the browser session
    session_start();

    // check if user is logged in
    if (!isset($_SESSION['authentication']))
    {
        header("location: index.php");
    }

    // check if authentication is valid
    elseif ($_SESSION['authentication'] !== "true")
    {
        header("location: login.php");
    }

    // double check that a user is logged in correctly
    if (!isset($_SESSION['uid']))
    {
        header("location: login.php");
    }

    // get the current time
    $time = date("y.m.d H.i.s");

    ?>

<head>
    <link rel="stylesheet" type="text/css" href="create.css">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Erstellen</title>
</head>

```





```

        else
        {
            // if the link does not contain the word create new tool with the entered attributes but add "https://" in front of the link
            $mysqli -> query("INSERT INTO `tools` (`name`, `link`, `uid`, `createtime`) VALUES ('" . $_POST['name'] .
            ", 'https://" . $_POST['link'] . "', '" . $_SESSION['uid'] . "', '" . $time . "')");
        }

        // get all data from the last tool created
        $result = $mysqli -> query("SELECT * FROM tools ORDER BY tid DESC LIMIT 1;");
        $row = $result->fetch_assoc();

        // check if a file was submitted in the form
        if($_FILES['fileToUpload']['size'] > 0)
        {
            // declare and initialize variable with the tool id to change the name of the uploaded file
            $rn=$row['tid'];

            // declare the target director where the file will get uploaded
            $target_dir = "assets/";
            $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
            $uploadOk = 1;

            // get the filetype of the uploaded file
            $imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

            // Check if image file is a actual image or fake image
            $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
            if($check !== false)
            {
                echo "File is an image - " . $check["mime"] . ".";
                $uploadOk = 1;
            }

            else
            {
                echo "File is not an image.";
                $uploadOk = 0;
            }

            // create new file name with tool id
            $temp = explode(".", $_FILES["fileToUpload"]["name"]);
            $newfilename = $target_dir . $rn . "." . $temp[1];

            // upload file
            if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $newfilename))
            {
            }

            // if upload fails display error
            else
            {
                echo"<div class='error'>
                Ein error ist aufgetreten
                </div>";
            }
        }

        // make entry in history that tool was created and redirect to dashboard
        $mysqli -> query("INSERT INTO `history` (`uid`, `element`, `changetime`, `changed`) VALUES
        ('" . $_SESSION['uid'] . "', '" . $row['tid'] . "', '" . $time . "', '" . $row['name'] . " tool erstellt')");
        header("location: index.php");
    }
}

?>

</div>

</div>

</body>

</html>

```

*edit.php*

```

<!DOCTYPE html>
<html>

    <?php

        // give history access to the sql config file
        include "config.php";

        // create new mysqli object
        $mysqli = new mysqli($host, $user, $dbpw, $db);

        // start the browser session
        session_start();

        // check if user is logged in
        if (!isset($_SESSION['authentication']))
        {
            header("location: index.php");
        }

        // check if authentication is valid
        elseif ($_SESSION['authentication'] !== "true")
        {
            header("location: login.php");
        }

        // double check that a user is logged in correctly
        if (!isset($_SESSION['uid']))
        {
            header("location: login.php");
        }

        // get current time
        $time = date("y.m.d H.i.s");

    ?>

<head>
    <link rel="stylesheet" type="text/css" href="edit.css">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bearbeiten</title>
</head>

<body>
    <div class="container">
        <div class="head">
            <div class="logo">
                <a href="index.php"></a>
            </div>

            <div class="title">
                <?php
                    echo "".$_GET['name']. " bearbeiten";
                ?>
            </div>

        </div>

        <div class="createwindow">
            <div class="fwrap">
                <?php
                    // display form with registered values
                    echo "
                    <form class='f' method='POST' enctype='multipart/form-data'>
                    <label for='name' class='label'>Name:</label>
                    <input type='text' name='name' class='input' required value='".$$_GET['name']."'>
                    <br>
                    <label for='text' class='label'>Link:</label>
                    <input type='text' name='link' class='input' required value='".$$_GET['link']."'>
                    <br>
                    <div class='label'>Bild:</div>
                    "
                ?>
            </div>
        </div>
    </div>

```

```

        <input type='file' class='fileinput' name='fileToUpload' accept='.jpg, .jpeg, .png'>
        <br>
        <input type='submit' name='senden' class='senden' value='senden'>
        <br>
    <br>
    <a href='index.php' class='abbrechen'>Abbrechen</a>
    </form>
    ";
    ?>

    <?php

    // if the user presses senden
    if(isset($_POST['senden']))
    {
        if(strlen($_POST['name'])>100)
        {
            // error if length is more then 100 caracters
            echo "<div class='error'>
Der Name ist zu lange maximal 100 Zeichen
</div>";
        }

        // check if link is more then 100 caracters
        elseif (strlen($_POST['link'])>100)
        {
            // error if length is more then 100 caracters
            echo "<div class='error'>
Der Link ist zu lange maximal 100 Zeichen
</div>";
        }

        // if link and name is under 100 caracters
        else
        {
            // connect to DB
            if ($mysqli -> connect_errno)
            {
                echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
                exit();
            }

            // declare what string to search in link
            $word = "https://";
            $mystring = $_POST['link'];

            // Test if string contains the word
            if(strpos($mystring, $word) !== false)
            {
                // if the link contains the string update tool with entered values
                $mysqli -> query("UPDATE `tools` SET
'name'='".$_POST['name']."', 'link'='".$_POST['link']."', 'changetime'='".$time.'" WHERE tid='".$_GET['id']."'");
            }
            // if link does not contain the string
            else
            {
                // change tool with entered values and "https://" added in front of link
                $result = $mysqli -> query("UPDATE `tools` SET
'name'='".$_POST['name']."', 'link'='https://' . $_POST['link'], 'changetime'='".$time.'" WHERE tid='".$_GET['id']."'");
            }

            // get all data from last changed tool
            $result = $mysqli -> query("SELECT * FROM tools ORDER BY changetime DESC LIMIT 1;");
            $row = $result->fetch_assoc();

            // declare variable with tool id for new file name
            $rn=$row['tid'];

            // check if a file was submitted in the form
            if($_FILES['fileToUpload']['size'] > 0)
            {
                // declare and initialize variable with the tool id to change the name of the uploaded file
                $rn=$row['tid'];

                // declare the target director where the file will get uploaded
                $target_dir = "assets/";

```

```

        $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
        $uploadOk = 1;

        // get the filetype of the uploaded file
        $imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

        // Check if image file is a actual image or fake image
        $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
        if($check !== false)
        {
            echo "File is an image - " . $check["mime"] . ".";
            $uploadOk = 1;
        }

        else
        {
            echo "File is not an image.";
            $uploadOk = 0;
        }

        // create new file name with tool id
        $temp = explode(".", $_FILES["fileToUpload"]["name"]);
        $newfilename = $target_dir . $temp[0] . ".png";

        // uplad file
        if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $newfilename))
        {
            // if uplad fails display error
            else
            {
                echo "<div class='error'>
                Ein error ist aufgetreten
                </div>";
            }
        }

        // entry in history that tool was changed
        $mysqli->query("INSERT INTO `history`(`uid`, `element`, `changetime`, `changed`) VALUES
        (".$_SESSION['uid'].",".$row['tid'].",".$time.",".$row['name'].`tool bearbeitet`)");
        header("location: index.php");
    }

}

// if user presse abrechen redirect to dashboard
elseif (isset($_POST['abbrechen']))
{
    header("location: index.php");
}

?>

</div>
</div>

</body>

</html>

```

## del.php

```

<!DOCTYPE html>

<html>

<?php

// give history access to the sql config file
include "config.php";

// create new mysqli object
$mysqli = new mysqli($host, $user, $dbpw, $db);

```

```

// start the browser session
session_start();

// check if user is logged in
if (!isset($_SESSION['authentication']))
{
    header("location: index.php");
}

// check if authentication is valid
elseif ($_SESSION['authentication'] !== "true")
{
    header("location: login.php");
}

// double check that a user is logged in correctly
if (!isset($_SESSION['uid']))
{
    header("location: login.php");
}

// get current time
$time = date("y.m.d H.i.s");

if ($mysqli -> connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
    exit();
}

?>

<head>
<link rel="stylesheet" type="text/css" href="del.css">
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Löschen</title>
</head>
<body>
<div class="container">
    <div class="head">
        <div class="logo">
            <a href="index.php"></a>

        </div>

        <div class="title">
            <?php
                echo "$_GET['name']. löschen";
            ?>
        </div>

    </div>

    <div class="createwindow">
        <div class="fwrap"></div>
        <form method="POST" class="f">
            <label class="labeö">Bist du sicher?</label>
            <br>
            <input class="senden" type="submit" name="ja" value="ja">
            <input class="senden" type="submit" name="abbrechen" id="abbrechen" value="abbrechen">
        </form>
    </div>

</div>

<?php
// if the user presses ja
if (isset($_POST['ja']))
{
    // delete tool with tool id from link
    $mysqli -> query("DELETE FROM `tools` WHERE tid='".$_GET['id']."'");

    // check if tool had a img
    $check = file_exists("assets/".$_GET['id'].".png");

    // if the file exists delete file

```

```

    if ($check==true)
    {
        unlink("assets/".$_GET['id'].".png");
    }

    // entry in history that tool was deleted then redirect to dashboard
    $mysqli->query("INSERT INTO `history` (`uid`, `element`, `changetime`, `changed`) VALUES
    (\"".$_SESSION['uid'].\"".$_GET['id'].\"".$_time.\"".$_GET['name'].\" tool gelöscht\");
    header("location: index.php");
}

// if the user presses abrechen redirect to Dashboard
elseif (isset($_POST['abrechen']))
{
    header("Location: index.php");
}

?>

</body>

</html>

```

## logout.php

```

<?php
session_start();
if (isset($_SESSION['authentication'])) {
    $_SESSION['authentication'] = "false";
    session_destroy();
    header("location: login.php");
}
else
{
    header("location: login.php");
}

?>

```

## config.php

```

<?php

$host = "localhost";
$user = "root";
$dbpw = "";
$db = "dashboarddb";

?>

```

## .css Seiten

### login.css

```

html, body
{
    width: 99vw;
    height: 99vh;
    overflow: hidden;
    margin-right: auto;
    margin-left: auto;
    font-family: sans-serif;
}

.head
{
    height: 5vw;
    background-color: #e8f9fb;
    width: 100%;
}

```

```
}  
  
.logo  
{  
  margin-top: 0.35%;  
  margin-bottom: 0.35%;  
  margin-left: 1%;  
  float: left;  
  width: 15%;  
  height: 85%;  
  text-align: left;  
}  
  
.title  
{  
  margin-top: 0.35%;  
  margin-bottom: 0.35%;  
  margin-left: -3%;  
  text-align: left;  
  float: left;  
  width: 50%;  
  height: 85%;  
  font-size: 2.5vw;  
  line-height: 225%;  
}  
  
.alpeinlogo  
{  
  width: 60%;  
  height: 100%;  
  max-height: 5vw;  
}  
  
.loginwindow  
{  
  height: 95vw;  
}  
  
.fwrap  
{  
  text-align: center;  
  margin-top: 12%;  
}  
  
.label  
{  
  display: block;  
  font-size: 1.5vw;  
  text-align: center;  
  width: 100%;  
}  
  
.input  
{  
  display: block;  
  width: 20vw;  
  height: 1vw;  
  margin-top: 0.1%;  
  text-align: center;  
  font-size: 1vw;  
  margin-left: auto;  
  margin-right: auto;  
}  
  
.senden  
{  
  cursor: pointer;  
  font-size: 1vw;  
  width: 45%;  
  height: 2vw;  
  margin-top: 1%;  
  background-color: #e8f9fb;  
  border-radius: 5%;  
  text-align: center;  
  border-style: none;
```



```
border: 0.5px solid black;
}

.senden:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}

.abbrechen
{
  cursor: pointer;
  font-size: 1vw;
  color: black;
  padding: 0.5vw;
  width: 45%;
  height: 2vw;
  margin-top: 1%;
  background-color: #e8f9fb;
  border-radius: 5%;
  text-align: center;
  border-style: none;
  border: 0.5px solid black;
}

.abbrechen:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}

.f
{
  text-align: center;
  width: 25%;
  margin: auto;
}

.error
{
  background-color: #faa0a0;
  border-top: 2px solid red;
  color: white;
  font-weight: bold;
  width: 20%;
  margin: auto;
  margin-top: 2%;
  padding: 1.5%;
}

.inputwrap
{
  height: 10%;
}
```

## reg.css

```
html, body
{
  width: 99vw;
  height: 99vh;
  overflow: hidden;
  margin-right: auto;
  margin-left: auto;
  font-family: sans-serif;
}

.head
```

```
{
  height: 5vw;
  background-color: #e8f9fb;
  width: 100%;
}

.logo
{
  margin-top: 0.35%;
  margin-bottom: 0.35%;
  margin-left: 1%;
  float: left;
  width: 15%;
  height: 85%;
  text-align: left;
}

.title
{
  margin-top: 0.35%;
  margin-bottom: 0.35%;
  margin-left: -3%;
  text-align: left;
  float: left;
  width: 50%;
  height: 85%;
  font-size: 2.5vw;
  line-height: 225%;
}

.alpeinlogo
{
  width: 60%;
  height: 100%;
  max-height: 5vw;
}

.loginwindow
{
  height: 95vw;
}

.fwrap
{
  text-align: center;
  margin-top: 12%;
  height: 10%;
}

.elabel
{
  font-size: 1.5vw;
  text-align: center;
  width: 100%;
}

.email
{
  width: 20vw;
  height: 1vw;
  margin-top: 0.1%;
  margin-bottom: 0.5%;
  text-align: left;
}

.pwlabel
{
  font-size: 1.5vw;
  text-align: center;
  width: 100%;
}

.pw
{
  width: 20vw;
```

```
height: 1vw;
margin-top: 0.1%;
margin-bottom: 0.5%;
text-align: left;
}

.senden
{
  cursor: pointer;
  font-size: 1vw;
  width: 40%;
  height: 2vw;
  margin-top: 1%;
  background-color: #e8f9fb;
  border-radius: 5%;
  text-align: center;
  border-style: none;
  border: 0.5px solid black;
}

.senden:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}

.abbrechen
{
  cursor: pointer;
  font-size: 1vw;
  color: black;
  padding: 0.5vw;
  width: 45%;
  height: 2vw;
  margin-top: 1%;
  background-color: #e8f9fb;
  border-radius: 5%;
  text-align: center;
  border-style: none;
  border: 0.5px solid black;
}

.abbrechen:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}

.f
{
  width: 25%;
  margin: auto;
}

.error
{
  background-color: #faa0a0;
  border-top: 2px solid red;
  color: white;
  font-weight: bold;
  width: 20%;
  margin: auto;
  margin-top: 2%;
  padding: 1.5%;
}
```

## create.css

```
html, body
{
  width: 99vw;
  height: 99vh;
  overflow: hidden;
  margin-right: auto;
  margin-left: auto;
  font-family: sans-serif;
}

.head
{
  height: 5vw;
  background-color: #e8f9fb;
  width: 100%;
}

.logo
{
  margin-top: 0.35%;
  margin-bottom: 0.35%;
  margin-left: 1%;
  float: left;
  width: 15%;
  height: 85%;
  text-align: left;
}

.title
{
  margin-top: 0.35%;
  margin-bottom: 0.35%;
  margin-left: -3%;
  text-align: left;
  float: left;
  width: 50%;
  height: 85%;
  font-size: 2.5vw;
  line-height: 225%;
}

.alpeinlogo
{
  width: 60%;
  height: 100%;
  max-height: 5vw;
}

.createwindow
{
  height: 95vh;
}

.fwrap
{
  text-align: center;
  margin-top: 12%;
}

.f
{
  width: 25%;
  margin: auto;
}

.label
{
  display: block;
  font-size: 1.5vw;
  text-align: center;
  width: 100%;
}
```

```
.input
{
  display: block;
  width: 20vw;
  height: 1vw;
  margin-top: 0.1%;
  text-align: center;
  font-size: 1vw;
  margin-left: auto;
  margin-right: auto;
}

.senden
{
  cursor: pointer;
  font-size: 1vw;
  width: 45%;
  height: 2vw;
  margin-top: 1%;
  background-color: #e8f9fb;
  border-radius: 5%;
  text-align: center;
  border-style: none;
  border: 0.5px solid black;
}

.ab
{
  vertical-align: middle ;
  text-decoration: none;
  cursor: pointer;
  font-size: 1vw;
  color: black;
  width: 45%;
  height: 2vw;
  margin-top: 1%;
  background-color: #e8f9fb;
  border-radius: 5%;
  text-align: center;
  border-style: none;
  border: 0.5px solid black;
  display: inline-block;
}

.ab:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}

.senden:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}

.fileinput
{
  border: 1px solid #ccc;
  display: inline-block;
  padding: 1% 2%;
  cursor: pointer;
  width: 60%;
  font-size: 0.5vw;
}

.error
{
  background-color: #faa0a0;
```

```
border-top: 2px solid red;
color: white;
font-weight: bold;
width: 20%;
margin: auto;
margin-top: 2%;
padding: 1.5%;
}
```

## edit.css

```
html, body
{
  width: 99vw;
  height: 99vh;
  overflow: hidden;
  margin-right: auto;
  margin-left: auto;
  font-family: sans-serif;
}

.head
{
  height: 5vw;
  background-color: #e8f9fb;
  width: 100%;
}

.logo
{
  margin-top: 0.35%;
  margin-bottom: 0.35%;
  margin-left: 1%;
  float: left;
  width: 15%;
  height: 85%;
  text-align: left;
}

.title
{
  margin-top: 0.35%;
  margin-bottom: 0.35%;
  margin-left: -3%;
  text-align: left;
  float: left;
  width: 50%;
  height: 85%;
  font-size: 2.5vw;
  line-height: 225%;
}

.alpeinlogo
{
  width: 60%;
  height: 100%;
  max-height: 5vw;
}

.createwindow
{
  height: 95vh;
}

.fwrap
{
  text-align: center;
  margin-top: 12%;
}

.f
{
  width: 25%;
}
```

```
margin: auto;
}

.label
{
  display: block;
  font-size: 1.5vw;
  text-align: center;
  width: 100%;
}

.input
{
  display: block;
  width: 20vw;
  height: 1vw;
  margin-top: 0.1%;
  text-align: center;
  font-size: 1vw;
  margin-left: auto;
  margin-right: auto;
}

.senden
{
  cursor: pointer;
  font-size: 1vw;
  width: 45%;
  height: 2vw;
  margin-top: 1%;
  background-color: #e8f9fb;
  border-radius: 5%;
  text-align: center;
  border-style: none;
  border: 0.5px solid black;
}

.abbrechen
{
  vertical-align: middle ;
  text-decoration: none;
  cursor: pointer;
  font-size: 1vw;
  color: black;
  width: 45%;
  height: 2vw;
  margin-top: 1%;
  background-color: #e8f9fb;
  border-radius: 5%;
  text-align: center;
  border-style: none;
  border: 0.5px solid black;
  display: inline-block;
}

.abbrechen:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}

.senden:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}

.fileinput
{
  border: 1px solid #ccc;
  display: inline-block;
```

```
padding: 1% 2%;
cursor: pointer;
width: 60%;
font-size: 0.5vw;
}
.error
{
background-color: #faa0a0;
border-top: 2px solid red;
color: white;
font-weight: bold;
width: 20%;
margin: auto;
margin-top: 2%;
padding: 1.5%;
}
```

## del.css

```
html, body
{
width: 99vw;
height: 99vh;
overflow: hidden;
margin-right: auto;
margin-left: auto;
font-family: sans-serif;
}

.head
{
height: 5vw;
background-color: #e8f9fb;
width: 100%;
}

.logo
{
margin-top: 0.35%;
margin-bottom: 0.35%;
margin-left: 1%;
float: left;
width: 15%;
height: 85%;
text-align: left;
}

.title
{
margin-top: 0.35%;
margin-bottom: 0.35%;
margin-left: -3%;
text-align: left;
float: left;
width: 50%;
height: 85%;
font-size: 2.5vw;
line-height: 225%;
}

.alpeinlogo
{
width: 60%;
height: 100%;
max-height: 5vw;
}

.createwindow
{
height: 95vh;
}
```



```
.fwrap
{
  text-align: center;
  margin-top: 12%;
}

.f
{
  width: 25%;
  margin: auto;
  text-align: center;
}

.label
{
  display: block;
  font-size: 1.5vw;
  text-align: center;
  width: 100%;
}

.senden
{
  cursor: pointer;
  font-size: 1vw;
  width: 45%;
  height: 2vw;
  margin-top: 1%;
  background-color: #e8f9fb;
  border-radius: 5%;
  text-align: center;
  border-style: none;
  border: 0.5px solid black;
}

.senden:hover
{
  font-size: 1vw;
  width: 50%;
  height: 2vw;
  background-color: #ffffd1;
  text-align: center;
}
```