IPA 2022

Abschlussarbeit  
Dashboard für Firmentools

08.03.2022

Shannon Neil Schuerch

ALPEIN Software SWISS AG  
Obergass 23  
8260 Stein am Rhein

Inhaltsverzeichnis

[Vorwort 0](#_Toc97729365)

[Vortschritssicherung 0](#_Toc97729366)

[Aufgabenstellung 0](#_Toc97729367)

[Beschreibung 0](#_Toc97729368)

[Ausgangslage 0](#_Toc97729369)

[Detailierte Aufgabenstellung 0](#_Toc97729370)

[Aufgabe in meinen Worten 1](#_Toc97729371)

[Mittel und Methoden 1](#_Toc97729372)

[Vorkentnisse 1](#_Toc97729373)

[Vorarbeiten 1](#_Toc97729374)

[Neue Lerninhalte 1](#_Toc97729375)

[Arbeiten in den letzten 6 Monaten 1](#_Toc97729376)

[Individuelle Bewertungskriterien 1](#_Toc97729377)

[I1: Systematik der Lösungsfindung/Lösungsvorschläge : 1](#_Toc97729378)

[I2: Entwurf, Design (Programmierung) 2](#_Toc97729379)

[I3: Plausibilisierung der Benutzer-Eingaben 3](#_Toc97729380)

[I4: Codierung, Fehlerbehandlung 3](#_Toc97729381)

[I5: Codingstyle - lesbarer Code 4](#_Toc97729382)

[I6: GUI Design (Fokus: Benutzerfreundlichkeit über die ganze Applikation) 4](#_Toc97729383)

[I7: GUI Design (Fokus: eine Maske/Bildschirm-/Webseite) 5](#_Toc97729384)

[Hilfestellung 5](#_Toc97729385)

[Zeitplan 5](#_Toc97729386)

[Arbeitsjournal 5](#_Toc97729387)

[Tag 01, Montag, 07.03.2022 5](#_Toc97729388)

[Tag 02, Dienstag, 08.03.2022 6](#_Toc97729389)

[Tag 03, Mittwoch, 09.03.2022 6](#_Toc97729390)

[Tag 04, Donnerstag, 10.03.2022 7](#_Toc97729391)

[Tag 05, Montag, 14.03.2022 7](#_Toc97729392)

[Tag 06, Dienstag, 15.03.2022 8](#_Toc97729393)

[Tag 07, Mittwoch, 16.03.2022 8](#_Toc97729394)

[Tag 08, Donnerstag, 17.03.2022 9](#_Toc97729395)

[Tag 09, Montag, 21.03.2022 9](#_Toc97729396)

[Tag 10, Dienstag, 22.03.2022 9](#_Toc97729397)

[Projekt 9](#_Toc97729398)

[Kurzfassung 9](#_Toc97729399)

[Ausgangslage 9](#_Toc97729400)

[Umsetzung 9](#_Toc97729401)

[Ergebnis 10](#_Toc97729402)

[Einleitung 10](#_Toc97729403)

[Informieren 11](#_Toc97729404)

[Ziele der Aufgabenstellung 11](#_Toc97729405)

[Vorgaben 11](#_Toc97729406)

[Fragen 11](#_Toc97729407)

[Planen 11](#_Toc97729408)

[Realisierungskonzep 11](#_Toc97729409)

[Datenbank modell 1](#_Toc97729410)

[Funktionsmodelle 0](#_Toc97729411)

[Testfälle 0](#_Toc97729412)

[Entscheiden 1](#_Toc97729413)

[Varianten 1](#_Toc97729414)

[Entscheind 1](#_Toc97729415)

[Realisieren 1](#_Toc97729416)

[Datenbank 2](#_Toc97729417)

[Login Funktion 2](#_Toc97729418)

[Regristrieren Funktion 3](#_Toc97729419)

[Tool hinzufügen Funktion 4](#_Toc97729420)

[Tool bearbeiten Funktion 7](#_Toc97729421)

[Tool löschen Funktion 7](#_Toc97729422)

[Bild hochladen Funktion 8](#_Toc97729423)

[Dashboard 8](#_Toc97729424)

[Layout & Design 8](#_Toc97729425)

[Plausible Eingaben & Benutzerfreundlichkeit 8](#_Toc97729426)

[Überprüfung aller Eingaben 8](#_Toc97729427)

[Redundanz entfernt 8](#_Toc97729428)

[Fehlerprüfung und Lösungsanwendung 8](#_Toc97729429)

[Kontrolieren 8](#_Toc97729430)

[Testprotokoll 8](#_Toc97729431)

[Testbericht 8](#_Toc97729432)

[Reflexion 8](#_Toc97729433)

[Glossar 8](#_Toc97729434)

[Abbildungsverzeichnis 8](#_Toc97729435)

[Quellenverzeichnis 8](#_Toc97729436)

[Anhang 8](#_Toc97729437)

[Anleitung 8](#_Toc97729438)

[SQL 9](#_Toc97729439)

[PHP 9](#_Toc97729440)

[HTML 9](#_Toc97729441)

[CSS 9](#_Toc97729442)

# 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 detailierte Aufgabenstellung erleutert 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.

# Vortschritssicherung

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 comits werden immer mit eine erklärende Notiz ergänz die den Fortschrit durchsichtiger gestallten soll.

# 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

## Aufgabe in meinen Worten

## Mittel und Methoden

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

## Vorkentnisse

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 zuhause 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 das 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**

**Güterstufe 1:**

**1 Punkt erfüllt**

**Gütestufe 0:**

**Kein Punkt ist erfüllt**

### **I7:** 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

# Zeitplan

# Arbeitsjournal

## Tag 01, Montag, 07.03.2022

|  |  |
| --- | --- |
| Arbeitsschrite | Zeitplan erstellt.  Dokumentation aufgebaut. |
| Erreichte Ziele | Zeitplan ohne soll ist vergleich fertig gebracht. |
| Probleme | Ich habe für den Zeitplan merh zeit benötigt als ich erwartet hatte. |
| Hilfestellung | - |
| Ausserplanmässige Arbeiten | Eine extra Stunde am Zeitplan dafür eine Stunde weniger an der Doku. |
| Reflexion | Ich war sehr nervös aber nachdem ich den ersten Tag überstanden habe gehe ich mit grosser Motivation auf den zweiten Tag zu. |

## Tag 02, Dienstag, 08.03.2022

|  |  |
| --- | --- |
| Arbeitsschrite | Zeitplan mit soll ist vergleich fertig machen  Dokumentation weiter aufgebaut  Aufgabenstellung studiert  Bewertungskriterien studiert |
| Erreichte Ziele | Zeitplan mit soll ist vergleich fetig.  Inhaltsverzeichniss fertig.  Vorwort verfasst.  Arbeitsjournal erstellt.  Aufgabe verstanden.  Bewertungskriterien verstanden. |
| Probleme | 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 bemekrt dass meine uhrsprüngliche mail fehlerfrei war. |
| Hilfestellung | - |
| Ausserplanmässige Arbeiten | - |
| Reflexion | Heute bin ich meiner Meinung nach weit gekommen. Die Gesamte struktur wird sich nicht merh ändern also kann ich jetzt stück für stück die Doku ausfüllen und Sie auch für Zukünftige Kanidaten als Vorlage sichern.  Das mit dem Zeitplan war ein wenig peinlich. |

## Tag 03, Mittwoch, 09.03.2022

|  |  |
| --- | --- |
| Arbeitsschrite | Realisierungskonzept  Datenbankmodell  Applikationsentwurf  Funktionsdiagramme  Testfälle erstellt  Datenbank erstellt |
| Erreichte Ziele | Realisierungskonzept  Datenbankmodell  Applikationsentwurf  Funktionsdiagramme  Testfälle erstellt  Datenbank erstellt |
| Probleme | - |
| Hilfestellung | - |
| Ausserplanmässige Arbeiten | Ich habe heute schon früher als geplant die Datenbank erstellt weil ich es für die einfachste möglichkeit gehalten habe das Datenbankmodell zu erstellen |
| Reflexion | Ich bin froh dass ich heute für die Datenbank Zeit gefunden habe. Jetzt kann ich Morgen früh an der Doku arbeiten |

## Tag 04, Donnerstag, 10.03.2022

|  |  |
| --- | --- |
| Arbeitsschrite | Für Lösungsvariante entscheiden  Login funktion entwickeln  Regristrierungs funktion entwickeln  Tool hinzufügen funktion entwickeln  Bild hochladen funktion entwickeln  Dashborad erstellen |
| Erreichte Ziele | Für Lösungsvariante entscheiden  Login funktion entwickeln  Regristrierungs funktion entwickeln  Tool hinzufügen funktion entwickeln  Bild hochladen funktion entwickeln  Dashborad erstellen |
| Probleme | - |
| Hilfestellung | - |
| Ausserplanmässige Arbeiten | Bild hochladen funktion entwickelt  Dashboard erstellt |
| Reflexion | Heute lief alles bestens und ich bin gut vorangekommen. Nachdem ich die Tool hinzufügen funktion entwickelt habe, habe ich mich entschieden die tool bearbeiten und löschen funktion zu verschieben und das Dashbord vorzuzihen. Weil es für mich viel einfacher ist die tool bearbeiten und tool löschen funktion zu entwickeln wenn ich weiss wie die tools im dashboard dargestellt werden. Ausserdem hat sich herausgestellt das die Bild hochladen funktion sehr einfach zu entwickeln ist daher habe ich sie direkt zusammen mit der tool hinzufügen funktion erstellt. |

## Tag 05, Montag, 14.03.2022

|  |  |
| --- | --- |
| Arbeitsschrite | Tool bearbeiten Funktion entwickeln  Tool löschen Funktion entwickeln |
| Erreichte Ziele | Tool bearbeiten Funktion  Tool löschen Funktion |
| Probleme | - |
| Hilfestellung | Ich habe meine VF gefrat ob er mir einen tipp geben kann wie ich die Applikation sicherer gestallten kann. Er hat mir daraufhin den tipp gegeben mich über nginx schlau zu machen was ich gerne machen werde. |
| Ausserplanmässige Arbeiten | Tool bearbeitn und Tool löschen funktion erst nach dashboard erstellen |
| Reflexion | Heute bin ich vorallem mit der Doku vorann gekommen. Ausserdem hatte ich recht mit der Annahme das es einfacher wäre die beiden Funktionen erst zu entwickeln wen das Dashboard steht. Im rückblick war das wohl schlecht geplant. |

## Tag 06, Dienstag, 15.03.2022

|  |  |
| --- | --- |
| Arbeitsschrite | CSS anpassen  Benutzer-Engaben plausibiliseren  Applikation Benutzerfreundlich gestalten |
| Erreichte Ziele | CSS ust 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 |
| Probleme | Heute als ich am Feierabend commiten wollte habe ich einen Error gekriegt und musste sourcetree neu installieren. Nach der Neuinstallation hat meine repository auf dem rechner nicht funktionert daher habe ich sie in einen neuen Projektordner geklont und arbeite fortann aus dem neuen Ordner. |
| Hilfestellung | - |
| Ausserplanmässige Arbeiten | 20 min sourcetree repaireren und sicherstellen das keine Daten verloren gegangen sind. |
| Reflexion | Das ganze nimmt immer mehr gestalt an was mich sehr zu sehen freut. Heute war gut bis auf den kleinen Hertz infakt als ich von sourcetree mit lauter errors begrüsst wurde. Jetzt bin ich aber wierder beruhigt da das Problem schnell gelöst war. Ich bin mir nicht sicher was mit sourcetree schief gelaufen ist aber ich hoffe es kommt nich wieder vor. |

## Tag 07, Mittwoch, 16.03.2022

|  |  |
| --- | --- |
| Arbeitsschrite | Überpüfung aller Eingaben  Code Struktur überprüfen  Code durchkommentiert |
| Erreichte Ziele | Alle eingaben werden nun hinterfragt  Der code ist ordentlich und kommentiert |
| Probleme | - |
| Hilfestellung | - |
| Ausserplanmässige Arbeiten | - |
| Reflexion | Der Code ist nun fast fertig morgen werde ich noch redundante teile rauslöschen und den code auf fehler überprüfen. |

## Tag 08, Donnerstag, 17.03.2022

## Tag 09, Montag, 21.03.2022

## Tag 10, Dienstag, 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

Zur Zeit sind alle Tools die unsere Mitarbeiter verwenden überall verstreut.

Neu soll es einen Singe 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 das jeder Mitarbeiter sienen eigenen Berecich 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 verwednet um die tools und nutzer zu verwalten.

Damit jeder Mitarbeiter seinen eigenen Bereich hat gibt es ein login- und regristrierungs Sytem. Die Authorisierung wird innerhalb der applikation geregelt.

Es gibt einen Verlauf indem jeder Mitarbeiter seine interaktionen mit seinen tools nachverfolgen 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 aufrecht zu erhalten. 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 das 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 estellt werden ausserdem ist es am besten wenn man falls möglich mehrere Lösungsansätze für eventuelle schwierigkeiten präsentieren kann.

### Entscheiden:

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

### Realisieren:

Wenn man gut geplan hat und sich für einen Projektablauf entschieden hat kann man nun mit der Realisierung beginnen. Dabei werden alle Arbeitschritte 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 Korrigiret und Dokumentiert werden.

### Auswerten:

Zuletzt gibt es eine Reflexion der Arbeit in der man analysier was alles besonders gut oder schlecht gelaufen ist und wass man in Zukünfitigen Projekten anders umsetzen würde.

## Informieren

Um mich richtig über den Auftrag zu informieren habe ich Ihn mehrmals gut durchgelesn bis ich mir sicher war dass ich alle erwartungen verstehe. Ausserdem habe ich mir den Leitfaden der QV 2022 heruntergeladen und dafür gesorft dass ich den Inhakt verstehe. Zu guter letzt habe ich noch die Bewertungskriterien verinerlicht und mir eine Version ausgeruckt um immer schnell einen Blick daraf 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 assets Ordner gespeicher werden.

Es landen keine unverschlüsselten Passwörter in der Datenbak.

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 Regristrieren 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 gefrag was ich einbeuen könnte um die Webapplikation ein wenig sicherrer zu gestalten. Er hat mir dazu zwei links über nginx gesendet die ich mir gerne ansehen werde.

## Planen

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

### Realisierungskonzep

Es wird ein Neuer Projektordner Dashboard erstellt indem alle Dateien die für die Arbeit benötigt werden abgeldeg sind.

In den Ordner befinden sich zwei Unterordner assets und Doku.

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 Ornder Doku befindet Sich die Dokumentation, der Zeitplan und alle andren Dateien die für die Abgabe benötigt werden.

Die Datenbank DashboardDB wird wie im Datenank Modell abgebildet implementiert. Der code zur erstellung der DB wird in eier 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 er Login Status auf allen wichtigen Datein direkt am Anfang überprüft.

### Applikationsentwurf

### Datenbank modell

### 

### Funktionsmodelle

#### Login Funktion

#### Registrieren Funktion

### Login Funktion

Die Login Funkton gibt die Authentikttion zurück. Nach der Eingabe wird geprüft ob die Emal schon regristiert ist oder nicht. Wenn die Email regristiiert ist verwende ich die Erstellungszeit zur vershlüsselung des Passwortes. Anschlisessend wird das Verschlüsslete Passwort mit dem unter der Email regristrierten Passwortes verglichen falls Sie übereinstimmen wird die Authentifikation gültig, die Userdaten werden in der Browser Session abgelegt und man wird auf das Dashoard weitergeleitet.

Hier ist die Funktion schritt fürschritt mit den 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">  
 <br>  
 <label for="pw" class="label">Passwort:</label>  
 <input type="password" name="pw" class="input">  
 <br>  
 <input type="submit" name="login" class="senden" value="login">  
 <br>  
 <input type="submit" name="reg" class="senden" value="Regristrieren">  
</form>

Wenn die Email regristriert ist wird das Erstellungsdatum des users für die Verschlüsselung des eingegebenen Passwortes verwendet

// get all data where enterd email = registerd emal  
$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 Regristrierten Passwort übereinstimmt.

// if encripted password matches registerd password set authentication to troo 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 macht authentication becomes false and output error  
{  
 $\_SESSION['authentication'] = "false";  
   
 echo"<div class='error'>  
  
 Inkorrektes Passwort  
 </div>";  
}

Falls die Passwörter Übereinstimmen wird die Authentifikation gültig und die Userdaten werden in der Session gespeichert.

### Regristrieren Funktion

Die Regristrieren Funktion kann neue User erstellen und gibt die Authentikation zurück. Nach der Eingabe wird geprüft ob die Email schon rgristireierst ist. Wenn ja wird der entschrechdende Error ausgegeben. Falls die Email noch nicht regristriert ist wer den die beiden eingegebenen Passwörter miteinander abgelichen. Falls sie übereinstimmen wird ein neuer User ohne das Passwort in der Datenbank erstellt um einen erstellungszeitpunk zu generieren. Anschliessend wird das eingegeben Passwot mit der eben generierten erstellungszeit verchlüsselt und dem neuen user hinzugefügt. Zum schluss wird die Authenfifikation dann gültig, die neuen Userdaten werden in der Browser Session abgelegt und man wird auf das Dashboard weitergeleteit.

Hier ist die Funktion schritt fürschritt mit den 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>  
 <input type="submit" name="login" class="senden" value="Noch nicht Regristriert?">  
</form>

Zunächst überprüfe ich ob die Email bereits regristriert 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>";   
}

Danach ob die beiden Passwörter aus der Eingabe übereinstimmen

// check if the password enterd match  
elseif($\_POST['pw']==$\_POST['pw1'])  
{

Wen die Email noch nicht regristriert ist und die Passwörter übereinstimmen wird die aktuelle Zeit in der Variabel time gespeichert und ein neuer User erstellt.

// get current time  
$time = date("y.m.d H.i.s");  
session\_start();  
  
// add new user with enterd values  
$mysqli -> query("INSERT INTO `user`(`email`, `createtime`, `changetime`) VALUES ('".$\_POST['email']."','".$time."','".$time."')");

Anschliessend wid das eingegebene Passwort mit der Erstellungszeit des users verschlüselt und der Neue user wird mit dem verschlüsselten Passwort upgedated

$gt = $mysqli -> query("SELECT \* from user ORDER BY createtime DESC LIMIT 1");  
$ggt = $gt->fetch\_assoc();  
  
// encript enterd password using changetime  
$salt = $ggt['changetime'];  
$spw = md5($salt . $\_POST['pw']);  
  
// upadate new user with encipted password  
$mysqli -> query("UPDATE `user` SET `pw`='".$spw."' WHERE email='".$\_POST['email']."'");

Zum schluss wird die Authentikation gültig, die Userdaten werden in der Session gespechert und man wird auf das Dashboard weitergeleitet.

// 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");

Wen die Passwörter nicht übereinstimmen gibt es einen Error

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

### Tool hinzufügen Funktion

Mit der Tool hinzugügen Funktion kann ein User seine Eigenen tools erstellen. Nach der Eingabe wird überprüft ob der eingegeben name und der eingegeben Link unter 100 zeichen lang sind. Falls ja wird überpüft ob im Link der String «https://» vorhanden ist. Falls ja wird das neue Tool mit den eingegeben daten Erstellt. Falls nein wird vor dem link noch ein «https://» angehäng und das Tool wird dann erstellt. Anschlissend 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 gitbt es einen einrag in den Verlauf das ein tool erstellt wurde und man wird auf das Dashboard weitergeleitet.

Hier ist die Funktion schritt fürschritt mit den 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{

Wenn beides nicht zustimmt überprüfe ich ob im link «https://» vorhanden ist und erstelle das Tool jeh ergebnis mit angehängtem «https://» oder ohne.

// declare what string to seach in link  
 $word = "https://";  
 $mystring = $\_POST['link'];  
  
 // Test if string contains the word   
 if(strpos($mystring, $word) !== false)  
 {  
 // if the link contains the word creatte new tool with enterd atributes  
 $mysqli -> query("INSERT INTO `tools`(`name`, `link`, `uid`, `createtime`) VALUES ('" . $\_POST['name'] . "','" . $\_POST['link'] . "','".$\_SESSION['uid']."','" . $time . "')");  
  
 }  
 else  
 { // if hte link does not contain the word create new tool with the enterd attributes but add "https://" infront 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 Vormular ausgewählt wurde.

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

Wenn kein Bild ausgewählt wurde wird man wirder auf das Dashborad verwiesen.

Falls ein Bild ausgewählt wurde wird bestimmt wo das bild 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;

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

// get the filetipe 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 = $target\_dir .$rn . "." . "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 estellt')");  
 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 fürschritt mit den 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='".$\_GET['name']."'>  
 <br>  
 <label for='text' class='label'>Link:</label>  
 <input type='text' name='link' class='input' required value='".$\_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>  
 <input type='submit' name='abbrechen' class='senden' value='Abbrechen'>  
  
 </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 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{

Wenn beides nicht zustimmt überprüfe ich ob im link «https://» vorhanden ist und update das Tool jeh ergebnis mit angehängtem «https://» oder ohne.

// declare what sring 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 upadate tool with enterd values  
 $mysqli -> query("UPDATE `tools` SET `name`='".$\_POST['name']."',`link`='".$\_POST['link']."',`changetime`='".$time."' WHERE tid='". $\_GET['id']."'");  
  
  
  
  
 }  
// if link does not contian the string  
 else  
 {  
 // change tool with enterd values and "https://" added infront 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 Vormular ausgewählt wurde.

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

Wenn kein Bild ausgewählt wurde wird man wirder auf das Dashborad verwiesen.

Falls ein Bild ausgewählt wurde wird bestimmt wo das bild 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;

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

// get the filetipe 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 = $target\_dir .$rn . "." . "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 glecihzeitig das dazugehörige bild aus dem Assets Ordner zu entfernen.

Hier ist die Funktion schritt fürschritt mit den 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 geletiet

// 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 wir überprüft ob es zu dem Tool ein Bild im Assets ordner gitb.

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

Falls ja wird das Bild aus dem Assets ordner empfernt

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

Zum schluss wird noch ein Eintrag in den Verlauf gemacht das ein tool gelöscht wurde und man wird auf das Dashboard weitergeleiet wo das Tool jetzt nicht mehr vorhenden 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 Toolid.png um die Bilder den Tools zuordnen zu können. Zunächst wird überprüft ob im Formular ein Bild ausgewächt wurde falls ja wird dann überprüft ob es sich um ein richtiges Bild hadelt oder nicht. Falls es sich um eine echtes Bild handelt wid die Datei mit dem neuen namen im Assets Ordner abgelegt.

Hier ist die Funktion schritt fürschritt mit den dazu gehörenden Code dargestellt:

Zuerst überprüfe ich ob im Formular eine Datei ausgewält wurde

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

Danach deklariere Ich die Variabel die für den neuen Dateinamen verwendet wird

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

Anschlissend besimme 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;

Zunächt überprüfe ich ob es sich um eine echtes Bild handelt

// get the filetipe 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 warden alle tools eines Users dargestellt dargestellt. Es wird für jedes fünfte tool eine neue Rheie generiert so dass immer vier in einer Rheie dargestellt werden. Es wird überprüft ob ein tool ein dazu passendes bild abgelegt hat und falls nicht wird das tool mit einem standart Bild dargestellt.

Hier ist die Funktion schritt fürschritt mit den dazu gehörenden Code dargestellt:

Die erste Rheie wird ausserhalb des php codes geöffnet

<div class="dashboard">  
  
 <div class="row">  
  
 <?php

Dann wird im php eine abrage vormuliert 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 wen mein counter 4 erreicht die rheie schliesst und eine neue öffnet und dann den counter wirder 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 Webapplikaton 2 verschiedene Layouts einmal für die index Seite(Dashboard) und für den Verlauf und ein zweites füt all die Funktionen wobei der Kopf für beide Layouts gleich bleibt.

Hier ist das Layout für den Index und Verlaf

<div class="container">  
  
 <div class="head">  
  
 <div class="logo">  
  
 <a href="index.php"><img class="alpeinlogo" src="assets/alpeinlogo.svg"></a>  
  
 </div>  
  
 <div class="title">  
 <?php   
 echo"&nbsp&nbsp&nbsp".$\_SESSION['uemail']."'s Dashboard";  
 ?>  
 </div>  
  
 </div>  
  
 <div class="sidenav">  
  
 <nav>  
  
 <br>  
  
 <div class="navlinkwrap"><a class="navlink" href="logout.php">&nbspAusloggen</a></div>  
  
 <div class="navlinkwrap"><a class="navlink" href="create.php">&nbspTool hinzufügen</a></div>  
  
 <div class="navlinkwrap"><a class="navlink" href="history.php">&nbspVerlauf</a></div>  
  
 </nav>  
  
 </div>  
  
 <div class="dashboard">

</div>

Hier ist das Layout für die Funktionen

<div class="container">  
   
 <div class="head">  
  
 <div class="logo">  
  
 <a href="index.php"><img class="alpeinlogo" src="assets/alpeinlogo.svg"></a>  
  
 </div>  
  
 <div class="title">  
 Login &nbsp&nbsp&nbsp  
 </div>  
  
 </div>  
  
  
 <div class="loginwindow">  
  
 <div class="fwrap">  
 <form class="f" method="POST">  
 </form>

</div>  
</div>

### Plausible Eingaben & Benutzerfreundlichkeit

Ich habe dafür gesogt dass alle Pflichtfleder eine Markierung haben davon gibt es in der Applikation 7 stück Email und Passwort auf de login Seite, Email Passwort und Passwort widerholen auf der Regristrieren Seite, tool Name und Link auf der tool hinzufügen seite. Alle anderen eingaben sind Optional und somit auch nich markiert. Für jeden Fehler den ein User bei den Eingaben machen kann wird ein dazu passender Error ausgegeben der Klar macht was verändert werden muss.

Hier sind Die Formulare mit Ihren verschiedenen Zuständen dargestellt.

Login Formular ohne eingaben:

Graphical user interface, application

Description automatically generated

Login Formular wenn eine unrealistische Email eingegben wird:  
Graphical user interface, application

Description automatically generated

Login Formular wenn eine noch nicht regristrierte Email eingegeben wird:

Graphical user interface, application

Description automatically generated

Login Formular wenn ein flasches Passwort eingegeben wird:

Graphical user interface, application

Description automatically generated

Regristriern Formular ohne eingabe:

Graphical user interface, text, application

Description automatically generated

Regristrieren Formular wenn eine unrealistische Email eingegben wird:

Graphical user interface, text, application, email

Description automatically generated

Regristirern Formular wenn eine bereis regristrierte Email eingegeben wird:

### Überprüfung aller Eingaben

### Redundanz entfernt

### Fehlerprüfung und Lösungsanwendung

## Kontrolieren

### Testprotokoll

### Testbericht

## Reflexion

# Glossar

# Abbildungsverzeichnis

# Quellenverzeichnis

# Anhang

## Anleitung

## SQL

## PHP

## HTML

## CSS