Wernle Yannick

What is it about?

Code, how the website works internally

Documentation DEV JSG

Development Dokumentation

Table of Contents

[Versions 2](#_Toc132293315)

[Documentation 2](#_Toc132293316)

[Program 2](#_Toc132293317)

[Use 3](#_Toc132293318)

[Concept – What is this Website for? 3](#_Toc132293319)

[Structure 4](#_Toc132293320)

[General 4](#_Toc132293321)

[Extensions etc. used 4](#_Toc132293322)

[Overview 5](#_Toc132293323)

[Database 6](#_Toc132293324)

[Detailed: HTML 8](#_Toc132293325)

[General 8](#_Toc132293326)

[Detailed: Javascript 9](#_Toc132293327)

[Detailed: PHP 10](#_Toc132293328)

[Detailed: CSS 11](#_Toc132293329)

[Detailed: Additional 12](#_Toc132293330)

[FAQ 13](#_Toc132293331)

[Additional 14](#_Toc132293332)

[Things to improve 14](#_Toc132293333)

[How to do Versions: 16](#_Toc132293334)

[GitHub 17](#_Toc132293335)

[Contact 18](#_Toc132293336)

# Versions

## Documentation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Author | Notes | Status |
| 07.03.2023 | v. 0.0 | Wernle Yannick | Initial - Development | Draft |
| 13.04.2023 | v. 0.0 | Wernle Yannick | Small Modifications, such as Versions | Draft |

## Program

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Developer | Notes | Status |
| 07.03.2023 | v. .0.0 | Wernle Yannick | Initial - Development | Draft |
|  |  |  |  |  |

# Use

## Concept – What is this Website for?

This website was created out of the need for a better solution for counting the absences of members in rehearsals; Previously, everything was done by hand on a sheet of paper and that needs to change.

The concept is, that one or more members of the JSG can easily document, which members of the JSG were present in a rehearsal, and who not. This is primarily needed for the general assembly for distributing prizes, depending on who participated in the most rehearsals. If needed, more “services” could be added.

#### Structure

The website contains a start page with a login prompt, where the user needs to login to change or document everything.

After login, the user is directed to a dashboard with 4 options: Create new User for adding a new JSG-member, Document todays absences, Todays submitted absences and Absences for a Time Period (User can choose over which time).

# Structure

## General

### Extensions etc. used

The Website was created with a template from [www.w3schools.com](http://www.w3schools.com) and therefore uses some stylesheets from there:

* https://www.w3schools.com/w3css/4/w3.css
* https://fonts.googleapis.com/css?family=Lato
* https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css

Furthermore, bootstrap 5.3.0 is used:

* https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css
* https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js

As well as Jquery

* https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js

### Overview

I don’t really know how to document such a website and because of that, I’ll just explain the functions and for each file what it does.

In General, the website works as follows: The user opens the website in his browser and is directed to a login form (index.html) if submitted, a database-query will validate the login (“login.php”) and if correct will direct the user to the main dashboard (index1.html).

On this page, the navbar shows the available services:

* Home (index1.html)
  + This page is used as a Dashboard/Landingpage and contains the logout-button, with which you can log out of the website (logout.js).
* Document Absences (registrere.html),
  + The script “getMembersForAbsence.js” loads current users into a form (using “faamedlemer.php”), for each with an option, “Present” or “Absent”, counted as 0 and 1 for ease of calculating later.  
    The form submits the input data with the PHP-file “submitToDB.php”. After insertion, the user will be redirected to Todays Absences.
* Add new User (nyebrukere.html),
  + The Script “getMembers.js” loads all current users into a table (using “faamedlemer.php”). Below, there is a form creating new members: skapmedlem.php.  
    After Insertion, the User remains on this side, although it will refresh and show the new user.
* Todays Absences (utgave.html),
  + The Script “getAbsences.js” gets the Absences from the database (using “faadatafraidag.php” with Todays Date and loads them into a table.
* Absences over time (utgaveTidsperiode.html)
  + A Form will take the desired start and end-dates and submit a request to the db (faadata.php). This script loads the data and puts them in files into the page “FileOutput.html”.

## Database

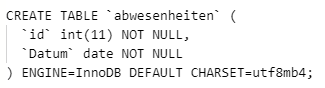
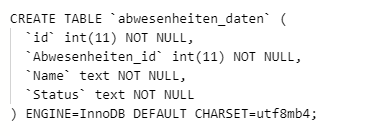
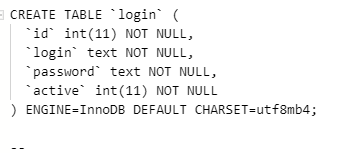
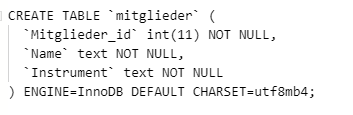
The Database was created with following parameters:

* phpMyAdmin (SQL Dump) version 5.2.0
* Host: 127.0.0.1
* Erstellungszeit: 05. Jan 2023 um 16:28
* Server-Version: 10.4.24-MariaDB
* PHP-Version: 7.4.29
* Database name: *jsg\_fravaer*

#### Table structure

The IDs are always Primary Keys and in the Case of the “Abwesenheiten\_daten”-Table, the ID of the “abwesenheiten”- and the “mitglieder”-table are foreign keys, referencing the data.

The IDs use auto-incrementation; they count up automatically with each record.

* Abwesenheiten
  + Here, the Date for Absences is saved along with and Id as a PK, to make it easier to change the date of absences
  + 
* Abwesenheiten\_daten
  + Here, the individual absences are saved; Along with FKs to the Absence-Table (“Abwesenheiten\_id”).
  + 
* Login
  + Here, the login is saved and compared to
  + 
* Mitglieder
  + Here the members are stored
  + 

## Detailed: HTML

### General

Currently, a script is executed on each html that checks for logged in cookies. Every site contains the same footer and navbar, which transforms to a Hamburger on small devices.

#### Index.html

Relevant on this site is only the login-form. It sends the Username and Password tipped in and sends it with the POST method on submit to “login.php”.

#### Index1.html

If the login is successful, the user is forwarded to this page. On here is nothing relevant; It’s the main page with maybe some text and the navbar for the other services, but nothing particularly unique.

#### Registrere.html

This page contains a table, in which the members are loaded into for documenting absences (“getMembersForAbsence.js”). This table is contained within a form, because the loaded in members can be set to either “Present” or “Absent”. On submit, the data is sent to “submitToDB.php”.

#### Nyebrukere.html

This page contains a table that loads the members into a table (“getMembers.js”) and below is a form for adding a new member; This is done via GET and with the page “skapmedlem.php”

#### Utgave.html

This page contains a table with todays absences, retrieved by “getAbsences.js”.

#### UtgvaeTidsperiode.html

This page contains a form for the desired timeperiod and this sends the data to “faadata.php”;

#### FileOutput.html

This page contains the path to the saved files with the queries from “UtgaveTidsperiode.php”.

## Detailed: Javascript

#### getAbsences.js

Gets data via «faadatafraidag.php», parses it into Javascript and appends it to the table, for each absence one row.

#### getMember.js

Gets the data via “faamedlemer.php”, parses it into Javascript and appends it into the table, for each member one row.

#### getMembersForAbsence.js

Gets the members from “faamedlemer.php”, parses it into Javascript and appends it to the table: For each member one row, each one containing a select with the options “Present” or “Absent”. For identifying the uniques, the variable name of each select is appended with an incrementing number.

#### isloggedin.js

Getting Cookies with plain Javascript and depending on which number, you’ll either be redirected to the login page or can continue.

#### Logout.js

Deletes the Cookie and redirects to login-form.

## Detailed: PHP

#### Faadata.php

Connects to DB and gets the records from within the timeperiod; Parses it to JSON and saves into files. Forward to “FileOutput.html”.

#### Faadatafraidag.php

Connects to DB and gets the records from today, converts to JSON and displays it for JS to read.

#### Faamedlemer.php

Connects to DB and gets the members, converts to JSON and displays it for JS to read, plus writes into a file.

#### Login.php

Gets Username and Password and checks for count of matching rows; If at least one matches, then the login is successful.

Sets cookie value corresponding to the success of the login.

#### Skapmedlem.php

Connects to DB and Inserts the new member as well as their instrument. Redirects to “nyebrukere.html”.

#### submitToDB.php

Connects to DB, checks if date already exists, and creates one if not. Get the members and insert each one with its status into the db.

Redirects to “utgave.html”.

## Detailed: CSS

#### Style.css

Changes very minor things, since practically everything is done with bootstrap:

Color change of selected HTML page in navbar is managed hear, plus some minor spacing things.

## Detailed: Additional

#### All\_Absences\_SortByName.csv

All Absences sorted by Name, within desired time period.

#### Absences\_AddedUp\_GroupedByName

All absences added up, grouped together by person.

# Additional

## Things to improve

##### Importance

* Needs to be fixed Immediately
* High
* Medium
* Low
* Nice to have

#### General

* Remove any remains of Norwegian words and alike (Mainly filenames)
* ~~Translation? (Done)~~
* Easter Eggs (for fun)
* Provider
* Revise Documentations ==> Diagram of Program
* ~~Change Wording for Absence (Done)~~

#### Database

* New field in Login (isAdmin) for admin functions
* Make “Name” field in «abwesenheiten\_daten» a foreign key referencing “mitglieder” table
* ~~Change Login in PHP-files (Done)~~
* ~~submitToDB.php => get Members via SQL and not file. (Done)~~
* ~~Checking if already exists (Done)~~

#### Functions and features

* Delete User (admin)
* Add User (admin)
* Edit Absence
* Optimize View of Absences
  + Table on website itself?

#### Behind the Scenes

* Editing History for verification
  + Date, Login, what type (“2023-03-07; Wernle\_y; Created new User”)
* Improve Login
* Improve Security
* Backup for the data
* ~~“Currently being changed” – Staging thingy (Done)~~
* Document documenting all Versions

#### ToDo before next major release

* Revise Documentation
* Document all Versions in Separate Document
* Provider
* Monolingual

<https://www.hostinger.com/tutorials/mysql/how-create-mysql-user-and-grant-permissions-command-line>

<https://stackoverflow.com/questions/97984/how-to-secure-database-passwords-in-php>

<https://www.hostinger.com/tutorials/mysql-show-users/#:~:text=Use%20the%20MySQL%20SHOW%20USERS%20Query,have%20been%20created%20in%20MySQL>.

<https://stackoverflow.com/questions/16674715/granting-permission-to-individual-fields-in-mysql>

<https://dev.mysql.com/doc/refman/8.0/en/drop-user.html#:~:text=The%20DROP%20USER%20statement%20removes,for%20the%20mysql%20system%20database>.

## How to do Versions:

Since this is my first program, I searched the internet and came up with the following sceme:

**vX.Y.Z\_a**

**X:**

X is used for major releases, that change something fundamental, that might not be backwards compatible.

**Y:**

Y is used for minor releases and new features.

**Z:**

Z is used for Bugfixes, Hotfix, Patches and alike

**a:**

a is used for indicating a change in the documentation which didn’t modify anything other than the documentation; For example when modifying the wording in the documentation or adding a paragraph.

**Increments:**

Each number is incremented normally and can only be reset, if the previous digit is modified (1.0.5 => 1.1.0 instead of 1.1.5).

The same applies to the documentation indicator – But it resets after each patch (1.1.0b => 1.1.1\_a instead of 1.1.1\_c)

Preferably, a Document with all Versions ever and descriptions each needs to be created for better oversight over all changes.

Reference I used:

[Best Practice: Software Versioning - Stack Overflow](https://stackoverflow.com/questions/2864448/best-practice-software-versioning)

<https://cpl.thalesgroup.com/software-monetization/software-versioning-basics>

[What is a Version? (computerhope.com)](https://www.computerhope.com/jargon/v/version.htm)

## GitHub

The program is saved in a Git-repository – currently set on private and only visible by the developers. To get access, please contact

[yativilli/JSG-Absences (github.com)](https://github.com/yativilli/JSG-Absences)

## Contact

#### Development:

*Yannick Wernle,*

*Remigerstrasse 31, 5234 Villigen*

[*yannick@wernle.net*](mailto:yannick@wernle.net)

*+41 077 464 47 80*

#### Management Board:

Current members:

[Vorstand – Jugendspiel Geissberg (jsgeissberg.ch)](http://www.jsgeissberg.ch/?page_id=56)

Contact:

[Kontakt – Jugendspiel Geissberg (jsgeissberg.ch)](http://www.jsgeissberg.ch/?page_id=60)

©Yannick Wernle, Jugendspiel Geissberg