



Albert-Einstein-Gymnasium

Dokumentation

zu unserem Projekt im Projektkurs

Mathe-Physik-Informatik

Autoren: Jakob Fleischer
Robin Schlaak
Lars Schmalbach

Prüfer: Herr Thomas Bachran

Abgabedatum: 21.06.2017

I Umschreibung des Projekts

Jakob ...

II Inhaltsverzeichnis

I	Umschreibung des Projekts	I
II	Inhaltsverzeichnis	II
III	Abbildungsverzeichnis	IV
IV	Tabellenverzeichnis	V
V	Listing-Verzeichnis	VI
1	Kapitel: Herangehensweise	1
1.1	ursprüngliche Vorstellung und Konsequenzen	1
1.2	verspätete Planung	1
1.3	Zusammenarbeit und Kommunikation	1
2	Kapitel: Grundlagen	2
2.1	LaTeX	2
2.2	XAMPP	2
2.3	HTML	2
2.3.1	Frames	3
2.3.2	Tabellen	3
2.3.3	Formulare	4
2.3.4	JavaScript	5
2.4	PHP	5
2.5	Apache	6
2.6	Editoren und integrierte Entwicklungsumgebungen	6
2.6.1	Eclipse	6
2.6.2	Notepad ++	6
2.6.3	Texmaker	6
2.7	GitHub	6
2.8	Dropzone	7
2.9	Java-Servlet	7
2.9.1	Einrichtung in Eclipse	7
2.9.2	Vorteile	7
2.9.3	Nachteile	7
2.10	JSON	8
2.10.1	Erstellen	8
2.10.2	Auslesen	8
2.10.3	Vorteile	8
2.10.4	Nachteile	8
2.11	MYSQL	8
2.11.1	Vorteile	8
2.11.2	Datenbank-Entwurf	8
2.11.3	Verwaltung einer Datenbank mit PHP	8

3	Projekt	9
3.1	Datei-Verwaltungs-Programm	9
3.2	GUI	9
3.3	Engine I: Interface und Datei-System	9
3.4	Engine II: Interface und Datenbank	9
4	Fazit	10
4.1	10
4.2	10
4.3	10
5	Literatur	11
	Anhang	I

III Abbildungsverzeichnis

IV Tabellenverzeichnis

V Listing-Verzeichnis

Lst. 1 Beispiel für ein einfaches HTML-Dokument	2
Lst. 2 Beispiel für Frames in HTML	3
Lst. 3 Beispiel für Tabellen in HTML	4
Lst. 4 Beispiel für Formulare in HTML	5

1 Kapitel: Herangehensweise

Jakob

Bei der Planung und der Herangehensweise an unser Projekt sind wir vermehrt auf Probleme gestoßen, welche in diesem Kapitel dargestellt werden. ...

1.1 ursprüngliche Vorstellung und Konsequenzen

Lars

Unsere Unerfahrenheit in der Projektarbeit hat uns zu einem überstürzten und unüberlegten Projekteinstieg getrieben, bei dem wir uns in Themen eingearbeitet haben, ohne uns einen genauen Plan zu erstellen. ...

Zu spät haben wir gemerkt, dass unsere bisherige Arbeit weder zielführend, noch in irgendeiner Weise sinnvoll war, weshalb wir uns zusammengesetzt und intensiv beraten haben.

1.2 verspätete Planung

Jakob

...

1.3 Zusammenarbeit und Kommunikation

Jakob

....

2 Kapitel: Grundlagen

Jakob & Lars

Im zweiten Kapitel dieser Dokumentation wird genauer auf die Themen eingegangen, in welche wir uns im Laufe des Projektkurses hineingearbeitet haben. Es wird nicht nur auf für das Projekt relevante Techniken, sondern auch auf solche, in die wir uns zwar eingearbeitet haben, sie jedoch nicht verwendet haben, eingegangen.

2.1 L^AT_EX

Robin

...

2.2 XAMPP

Robin

...

2.3 HTML

Lars

HTML ist die Abkürzung für „Hypertext Markup Language“, was zu deutsch „Hypertext-Auszeichnungssprache“ heißt. Es ist eine Programmiersprache, mit der man den Aufbau von Internetseiten bestimmt. Solche HTML-Dokumente stellen die Grundlage für das World Wide Web dar und werden von Browsern dargestellt.[1][2] Sie bestehen in der Regel aus drei Teilen.[3]

```
1 <html>
2   <head>
3     <meta charset="UTF-8">
4     <title> Titel</title>
5   </head>
6   <body>
7     Sichtbarer Text auf der Webseite.
8   </body>
9 </html>
```

Listing 1: Beispiel für ein einfaches HTML-Dokument

Der erste Teil eines üblichen HTML-Dokuments ist die Dokumenttyp-Deklaration. In ihr werden Angaben zur verwendeten HTML-Version gegeben. Im „head“, welcher den zweiten Teil darstellt, werden Kopfdaten, wie zum Beispiel der Titel der Seite oder andere,

für den menschlichen Betrachter der Webseite zunächst nicht sichtbare, Informationen zur korrekten Darstellung des sichtbaren Teils der Webseite, angegeben.[4] Anzuzeigende Inhalte werden in den „body“, den dritten Teil, geschrieben. Hier werden also sämtliche Texte, Verweise, Grafiken und so weiter eingefügt, die auf der Webseite sichtbar sein sollen.[3]

In unserem Fall stellt HTML die Grundlage für die optische Gestaltung des GUIs dar.

2.3.1 Frames

Eine hilfreiche Technik, die auch bei uns ihren Einsatz gefunden hat, heißt Frames. Diese Technik wurde 1996 von Netscape eingeführt. Mit ihr kann man mehrere Dateien gleichzeitig auf dem Bildschirm anzeigen lassen.[5] Sie wurde jedoch im Oktober 2014 mit HTML5[6] aus dem Standard entfernt, da sie entscheidende Nachteile aufweist. Aufgrund des Verwendungszweckes unserer Webseite benutzen wir Frames, obwohl empfohlen wird, Server-seitig andere Techniken zum Auslagern von Teilen der Seite zu benutzen.[5] Die stärksten Argumente waren, die simple Handhabung und die guten Gestaltungsmöglichkeiten mit dieser Technik.

Im Folgenden Beispiel wird ein sogenanntes Frameset dargestellt, bei dem der Bildschirm, mit dem Attribut „rows“ in zwei Zeilen aufgeteilt wird, wobei die obere 20% der Pixel einnimmt und die untere den Rest, also 80%. Alternativ kann man den Bildschirm auch in Spalten aufteilen, dies geschieht mit dem Attribut „cols“. Des Weiteren wird mit dem Attribut „border“ die Breite des Randes zwischen den jeweiligen Frames angegeben.

```
1 <html>
2   <head>
3     <title>Titel</title>
4   </head>
5   <frameset rows="20%,*" border="1">
6     <frame src="Quelle1.html">
7     <frame src="Quelle2.html">
8   </frameset>
9 </html>
```

Listing 2: Beispiel für Frames in HTML

2.3.2 Tabellen

Tabellen in HTML [2.3] bieten gute, einfache und vielseitige Möglichkeiten, Internetseiten zu strukturieren. Sie wurden im Januar 1997 mit HTML 3.2 ins Standardrepertoire von HTML aufgenommen.[2]

Das Folgende Beispiel beinhaltet eine Tabelle mit zwei Zeilen und zwei Spalten. Tabellen

in HTML werden Reihe für Reihe definiert. Eine Reihe beginnt mit „tr“ und wird mit „/tr“ beendet. Auf die selbe Art werden auch Tabelleneinträge, also die Spalten in den Zeilen, definiert.

```
1 <html>
2   <head>
3     <meta charset="UTF-8">
4     <title> Titel</title>
5   </head>
6   <body>
7     <table>
8       <tr>
9         <td>
10           Oben links
11         </td>
12         <td>
13           Oben rechts
14         </td>
15       </tr>
16       <tr>
17         <td>
18           Unten links
19         </td>
20         <td>
21           Unten rechts
22         </td>
23       </tr>
24     </table>
25   </body>
26 </html>
```

Listing 3: Beispiel für Tabellen in HTML

2.3.3 Formulare

Formulare sind ein Element von HTML [2.3], das es ermöglicht Daten zu erfassen und über das Hypertext Transfer Protocol per XMLHttpRequest, HTTP-GET oder HTTP-POST zur Verarbeitung an einen Server zu senden.[7] In unserem Programm kam letzteres zum Einsatz. Man kann in HTML zwar Formulare definieren und erstellen, für eine Verarbeitung und Auswertung der Eingaben ist jedoch eine andere Programmiersprache, wie zum Beispiel Javascript [2.3.4] oder PHP [2.4] nötig.[11]

In unserem Fall haben wir Formulare in Form von Anmeldeformularen, Suchfiltern und auch als Möglichkeit zum Hochladen von Dateien eingesetzt.

In dem folgenden Beispiel ist die Implementation eines Formulars in HTML dargestellt. Zu sehen ist ein Formular, welches ein Label, also den Text „Suchbegriff“ enthält. Die Zugehörigkeit des darauf folgenden Eingabefelds zum Label wird durch das Attribut „name“ festgelegt. Das zweite „input“ Statement erstellt den Bestätigungsknopf, den man drücken muss, um das Formular abzusenden. Beim Absenden des Formulars wird die Datei oder die Internetseite aufgerufen, die im Kopf des Formulars unter dem Attribut „action“ steht.

```
1 <html>
2   <head>
3     <title>
4       Titel
5     </title>
6   </head>
7   <body>
8     <form action="action.php">
9       <label for="begriff">Suchbegriff</label>
10      <input type="text" name="begriff">
11
12      <input type="submit" name="Submit" value="Suchen">
13    </form>
14  </body>
15 </html>
```

Listing 4: Beispiel für Formulare in HTML

2.3.4 JavaScript

Bei JavaScript handelt es sich um eine interpretierende Programmier- beziehungsweise Skriptsprache, die 1995 von Netscape entwickelt wurde.[8][9] JavaScript ist sehr verbreitet, da sich in allen modernen Browsern Interpreter für die Sprache befinden. Es wird hauptsächlich Client-seitig verwendet und ermöglicht es, dynamischen Einfluss auf Webseiten zu nehmen.[10] Für unser Programm kam die Sprache besonders häufig aufgrund des, durch sie ermöglichten, einfachen Umgangs mit Variablen und Funktionen zum Einsatz.

2.4 PHP

Robin

...

2.5 Apache

Lars

...

2.6 Editoren und integrierte Entwicklungsumgebungen

Lars

Editoren werden zum Schreiben von Texten, wie zum Quellcodes, benutzt. gute Editoren helfen das Programmieren zu vereinfachen, indem sie gewisse Schlüsselwörter, sowie Befehle, farblich hervorheben, eine Autovervollständigung, eine Such- und Ersetzfunktion und indem sie den Quellcode automatisch einrücken, sowie eine Schnittstelle für Plugins darstellen.[13]

Neben Editoren gibt es auch integrierte Entwicklungsumgebungen. Diese bestehen aus einer Sammlung an Computerprogrammen, mit denen es möglich ist Software ohne die Verwendung vieler einzelner Programme zu entwickeln. Durch sie werden nicht nur Tippfehler verhindert, sondern auch Arbeitsschritte und somit Zeit bei der Softwareentwicklung gespart.[12][14] Bei der Entwicklung unseres Programms kamen die Editoren und integrierten Entwicklungsumgebungen Eclipse [2.6.1] (genau genommen Eclipse Neon IDE), Notepad++ [2.6.2] und Texmaker [2.6.3] zum Einsatz.

2.6.1 Eclipse

Robin

...

2.6.2 Notepad ++

Robin

...

2.6.3 Texmaker

Robin

...

2.7 GitHub

Jakob

Durch die Arbeit mit GitHub, wird das gemeinsame, nicht zwingend parallele, Arbeiten möglich. GitHub verwaltet die Quellcodes, welche die Benutzer hochladen so, dass jeder, der über Zugriff auf das Projekt verfügt, dieses weiter führen kann. -Lars

2.8 Dropzone

Lars

Ursprünglich beschreibt der Begriff „Dropzone“ einen geheimen Speicherort für maschinell gestohlene Daten, wie zum Beispiel Passwörter und Kontodaten.[15]

In unserem Fall ist die Dropzone jedoch ein Script, welches in JavaScript [2.3.4] geschrieben wurde. Dieses dient zur einfachen Gestaltung des Datei-Uploads mittels HTML. Es soll die optische Anpassung des Eingabefelds vereinfachen. Aufgrund ihrer Komplexität, welche sich leider erst nach und nach herauskristallisierte, haben wir uns schlussendlich gegen die Dropzone und für einen herkömmlichen Datei-Upload mittels HTML-Formular [2.3.3] entschieden.

2.9 Java-Servlet

Jakob

Im Folgenden werde ich erläutern, wie man ein Servlet in Eclipse über einen Apache Tomcat Server erstellen kann.

2.9.1 Einrichtung in Eclipse

Vorausgesetzt für eine erfolgreiche Einrichtung ist die bereits erfolgte Installation von Eclipse Neo in der Java EE Version und Apache Tomcat v 9.0. Nachdem diese erfolgt ist, ist es nun möglich in Eclipse einen neuen Server einzurichten, wobei man unter dem Ordner Apache die installierte Tomcat Version vorfindet, welche hier als Servertyp verwendet wird. Nachdem dies geschehen ist, richten sie ein Dynamic Web Project ein und innerhalb desselben erstellen sie im Java Resources/src Ordner ein neues Servlet, indem sie nun programmieren können.

Wenn nun aber die Fehlermeldung „cannot resolve“ im Bezug auf die vorgegebenen import-Zeilen auftaucht müssen sie außerdem noch die servlet-api.jar downloaden und diese dann in den Eigenschaften ihres Dynamic Web Projects unter Java Build Path in der Kategorie Libraries mit der Funktion „Add External JARs“ hinzufügen.

...

2.9.2 Vorteile

...

2.9.3 Nachteile

...

2.10 JSON

Jakob

Ein weiteres Thema, in welches ich mich eingearbeitet habe war die Erstellung und das Auslesen von Dateien des Formates JSON mit Eclipse. ...

2.10.1 Erstellen

...

2.10.2 Auslesen

...

2.10.3 Vorteile

...

2.10.4 Nachteile

...

2.11 MYSQL

Jakob

Nachdem obige Thematiken angesichts unseres erneuerten Projektplans irrelevant geworden waren, habe ich mich mit MYSQL beschäftigt. ...

2.11.1 Vorteile

...

2.11.2 Datenbank-Entwurf

Im Folgenden möchte ich auf das Entwerfen einer MYSQL-Datenbank eingehen. ...

2.11.3 Verwaltung einer Datenbank mit PHP

Um eine Datenbank in unser Projekt zweckmäßig einzurichten, musste ich mich in eine MYSQL-Einbindung mittels PHP einlesen, welche ich nun darstellen möchte. ...

3 Projekt

Lars

In diesem Kapitel wird nun unser Projekt dargestellt, indem zunächst das Programm im Gesamten beschrieben wird. Daraufhin wird die Arbeit von Lars am GUI, die Arbeit von Robin am Interface und Datei-System und zuletzt die Arbeit von Jakob am Interface und an der Datenbank dargestellt. ...

3.1 Datei-Verwaltungs-Programm

Lars

Darstellung Ergebnis...

3.2 GUI

Lars

Darstellung GUI...

3.3 Engine I: Interface und Datei-System

Robin

Darstellung Interface und File-System...

3.4 Engine II: Interface und Datenbank

Jakob

Darstellung Interface und Datenbank...

4 Fazit

Im Folgenden wird jeder von uns ein Fazit geben, indem er darstellt, was er aus dem Projekt gelernt hat, was ihm gefallen hat, Sachen die er gut findet und so weiter.

4.1

Lars

...

4.2

Robin

...

4.3

Jakob

...

5 Literatur

- [1] <https://wiki.selfhtml.org/wiki/HTML>,
Zugriff: 15.06.2017
- [2] https://de.wikipedia.org/wiki/Hypertext_Markup_Language,
Zugriff: 15.06.2017
- [3] https://wiki.selfhtml.org/wiki/HTML/Dokumentstruktur_und_Aufbau,
Zugriff: 15.06.2017
- [4] <https://wiki.selfhtml.org/wiki/HTML/Kopfdaten>,
Zugriff: 15.06.2017
- [5] <https://wiki.selfhtml.org/wiki/HTML/Frames>,
Zugriff: 15.06.2017
- [6] <https://de.wikipedia.org/wiki/HTML5>,
Zugriff: 15.06.2017
- [7] <https://de.wikipedia.org/wiki/Webformular>,
Zugriff: 15.06.2017
- [8] <http://www.searchenterprisesoftware.de/definition/JavaScript>,
Zugriff: 15.06.2017
- [9] <https://de.wikipedia.org/wiki/JavaScript>,
Zugriff: 15.06.2017
- [10] <http://www.itwissen.info/JavaScript-JavaScript-JS.html>,
Zugriff: 15.06.2017
- [11] <https://wiki.selfhtml.org/wiki/HTML/Formulare/Form>,
Zugriff: 15.06.2017
- [12] https://de.wikipedia.org/wiki/Integrierte_Entwicklungsumgebung,
Zugriff: 16.06.2017
- [13] <https://de.wikipedia.org/wiki/Texteditor>,
Zugriff: 16.06.2017
- [14] <https://de.wikipedia.org/wiki/Medienbruch>,
Zugriff: 16.06.2017
- [15] <https://de.wikipedia.org/wiki/Dropzone>,
Zugriff: 16.06.2017

Anhang

Erklärung

Hiermit versichere ich, dass ich meine Dokumentation selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Datum:

.....

(Unterschrift)