# 1 Ziele

1. Statistikseite
   1. Erstellen und Einbinden der API-Funktion
   2. Suchmöglichkeiten und Filterung
   3. Anzeige der Statistik nach Filterung
2. Gästebuch
   1. Erstellen und einbinden
   2. Verbindung mit Datenbank
      1. Datensätze/Eingaben übergeben und in Datenbank speichern
      2. Datensätze zurückholen und anzeigen
3. Designanpassungen
   1. Navigation
   2. Fußzeile
   3. Content Bereich
   4. Header
   5. Tabellen
   6. Gästebuch
   7. Scrollbar
4. Responsive Webdesign anpassen
5. Erstellung des Banners
   1. Planung
   2. Skizzierung
   3. Erstellen eines Grobkonzepts
   4. Fertigstellung des Banners
6. LiveMap (Zusammenarbeit Andre Münstermann, API-Gruppe)
   1. Einbinden der Karte
   2. Anzeigen von statischen Werten
   3. Anzeigen mit den Werten der User
   4. Verlauf der User anzeigen
7. Kontaktformular
   1. Button, Textfelder erstellen
   2. Funktion für die Absendung erstellen
8. Inhalte
   1. Inhalte der Seiten ändern (Über das Team, Spiel, Startseite)
9. Google Fonts
   1. Einbinden

# 2 Umsetzung

### 2. 1 Lösungskonzept

2.1.1. Statistik

Für das Anfertigen der Statistikseite wurde die API-Funktion eingebunden, um die eingetragenen Daten aus der Datenbank zu erhalten und diese anzuzeigen.

<?php

//connection

$arr = $klasse -> getTopTenScoresForAllMinigamesByUsername($Username);

$arr = $klasse -> getTopTenScoresForAllMinigames();

//var\_dump($arr);

//var\_dump($arr->{'Zukunftsmeile'});

?>

Außerdem erstellten wir eine Möglichkeit, um die Auswahl der Statistiken nach den Spielernamen zu filtern und den Usern zu ermöglichen einen besseren Überblick über die Daten zubekommen, welche sie auch sehen möchten. In einem Dropdown-Menü hat der User einen Überblick über alle angemeldeten Spieler, wo der User dann ein Spieler auswählen kann und dann mit über den Button „Filtern“ die Ausgabe filtern kann. Es werden bei jeder Ausgabe, egal ob gefiltert oder nicht gefiltert nur die besten 10 Datensätze ausgegeben.

div class="tables"><?php

drawTable($arr->{'Zukunftsmeile'},'Zukunftsmeile');

drawTable($arr->{'HNF'},'HNF');

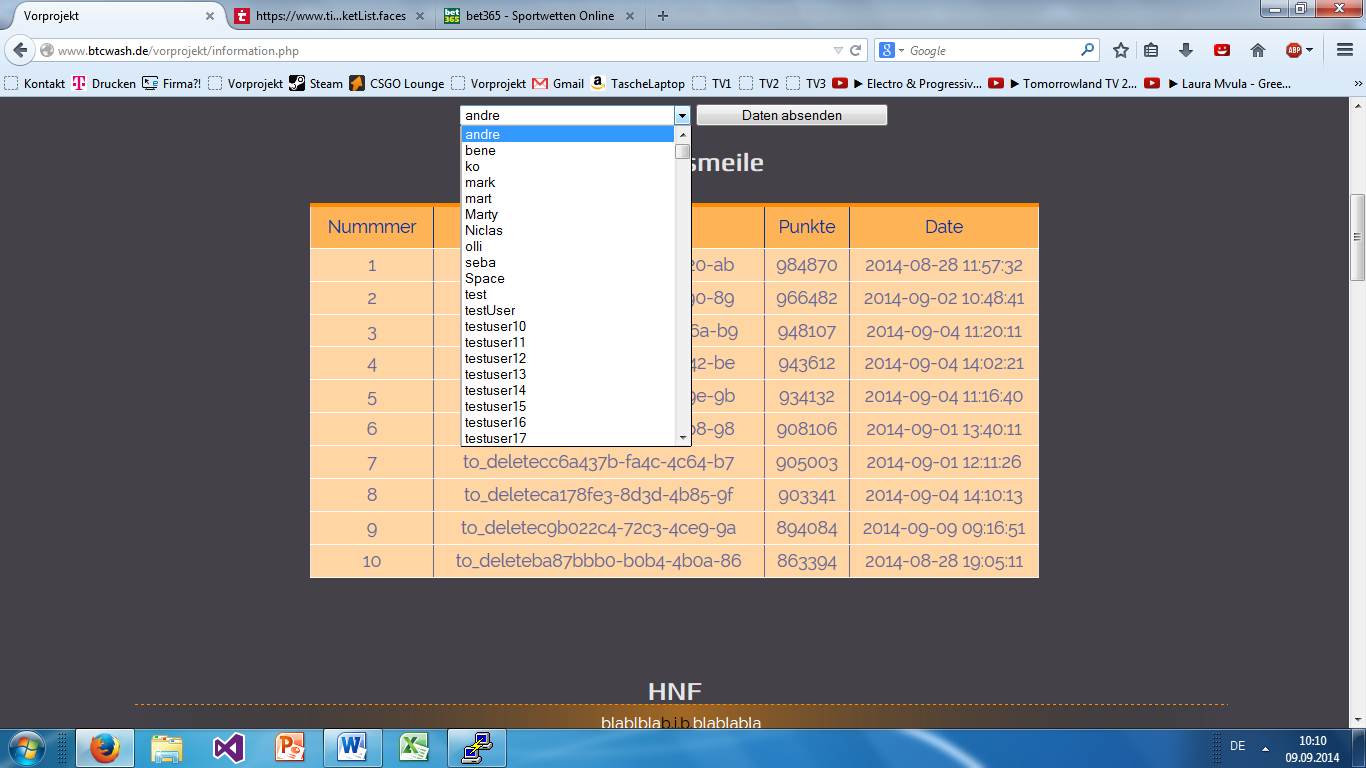
drawTable($arr->{'Fluss'},'Fluss');

drawTable($arr->{'Serverraum'},'Serverraum');

drawTable($arr->{'Wohnheim'},'Wohnheim');

?></div>

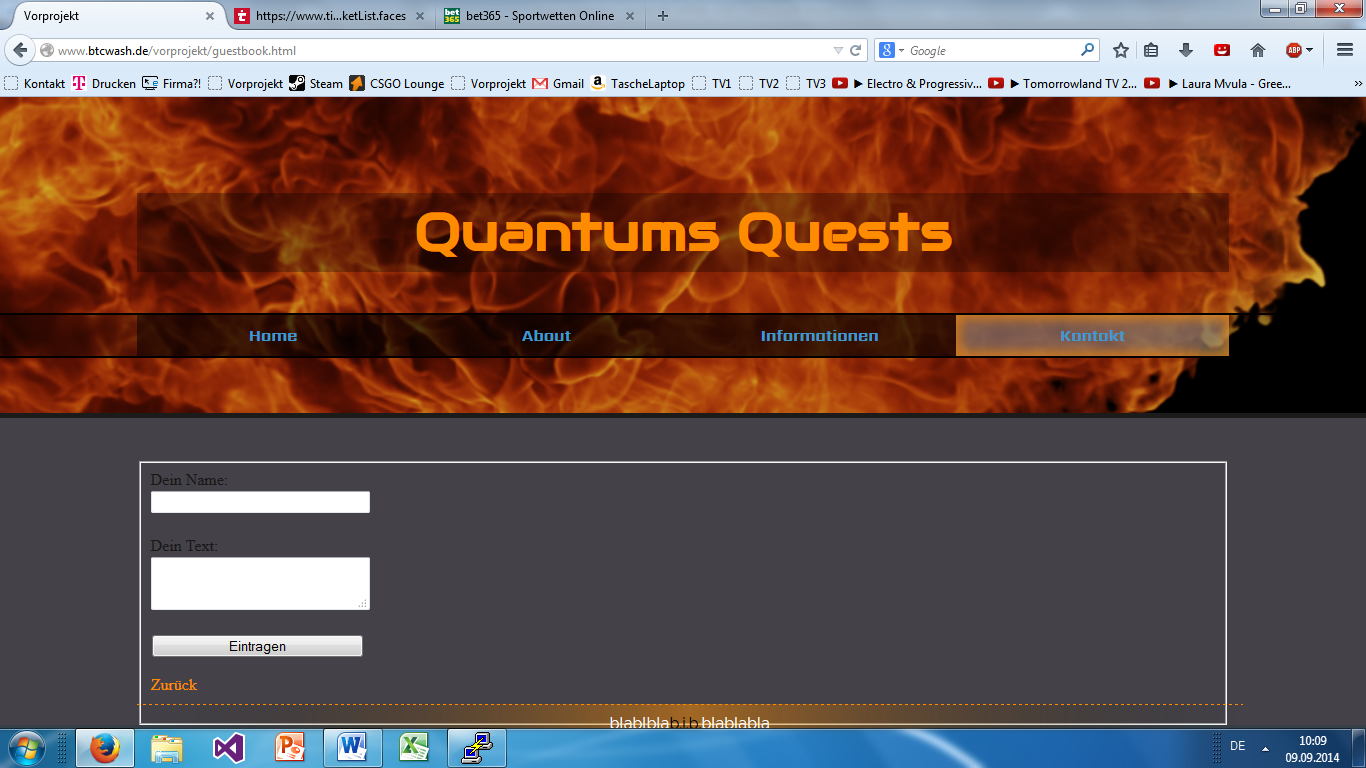
Hier werden die Tabellen für die einzelnen Spiele erstellt und Überschriften mitgegeben.





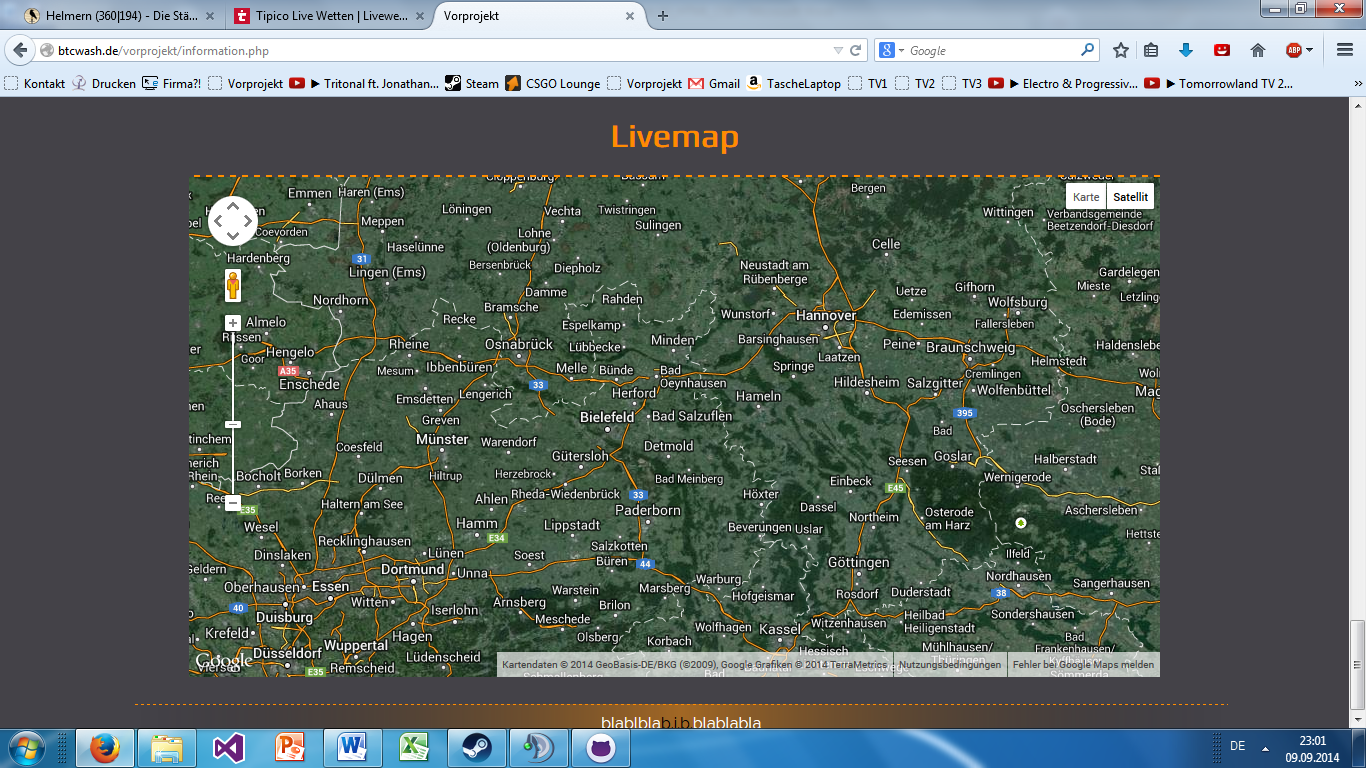
2.1.2. Gästebuch

Als erstes wurde das Gästebuch auf einem Server mit einer MySQL Datenbank zum Laufen gebracht, da die API-Funktionen zu dem Zeitpunkt noch nicht bereit waren. Damit wurden schon einmal die Einträge angezeigt und es konnten neue Einträge gemacht werden. Nachdem dann die API-Funktion fertig war, wurde diese dann eingebunden und es werden nun alle Einträge untereinander angezeigt mit dem Zeitpunkt, dem Namen des Autors und den geschriebenen Text. Mit dem Betätigen des Textes „Neuen Eintrag schreiben“ öffnet sich eine neue Seite, dort kann der User dann seinen Namen und seine Nachricht in zwei Textfeldern eingeben. Danach kann der User über einen Button „Eintragen“ den Eintrag abschicken und dieser wird dann zur Datenbank übermittelt. Außerdem kann der Spieler auch über den Text „Zurück“ das Eintragen abbrechen und er kommt zurück zur vorherigen Seite.



2.1.3. Designanpassungen

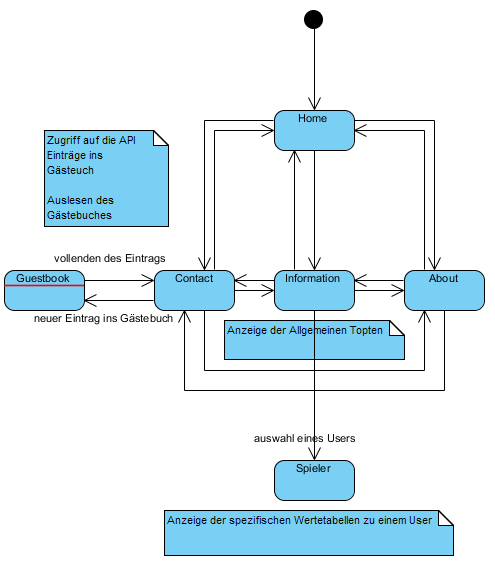
In diesem Schritt wurden alle möglichen Designs verbessert. Wir begannen mit der Navigation und nutzen Farben welche in den Designrichtlinien angegeben wurden. Danach wurde die Fußzeile noch optisch so angepasst, dass diese keinen zu großen Platz wegnimmt und optisch nicht so stark in den Vordergrund rückt wie zuvor. Dann wurde der Header mit einem neuen passendem Bild ausgestattet und die Tabellen und das Gästebuch wurden optisch ansprechender gestaltet, um dem User ein besseren Überblick zu schaffen. Zum Schluss erstellten wir noch eine farbige Scrollbar.



2.1.4. Responsive Webdesign anpassen

Durch das Gästebuch und die Tabellen musste das Responsive Webdesign nochmal überarbeitet werden, weil diese sich noch nicht angepasst haben.

Auch die Navigation wurde angepasst. Die Navigation ist auf jeder Seite der Homepage sichtbar und man kann dort immer alle anderen Seiten auswählen. Die Unterseite guestbook.html kann nur unter der contact.php aufgerufen werden.



2.1.5. Banner

Für den Header wurde ein Banner erstellt, welcher in einzelnen Schritten erstellt wurden. Zuerst wurde geplant, wie der Header aussehen könnte. Diese Planung wurde dann skizziert. Später wurde ein Grobkonzept entwickelt.



2.1.6. LiveMap

Für die LiveMap wurde mit Hilfe der Google API die Google Map angezeigt. Als letztes wurde mit Hilfe der Standorte der Spieler noch ein Verlauf der jeweiligen Spieler erstellt. Diese werden in verschiedenen Farben angezeigt so dass man die Laufbahnen von den einzelnen Spielen sehen kann.

function addMarker(location,title) //funktion die die User zur LiveMap hinzufügt

{

marker = new google.maps.Marker({

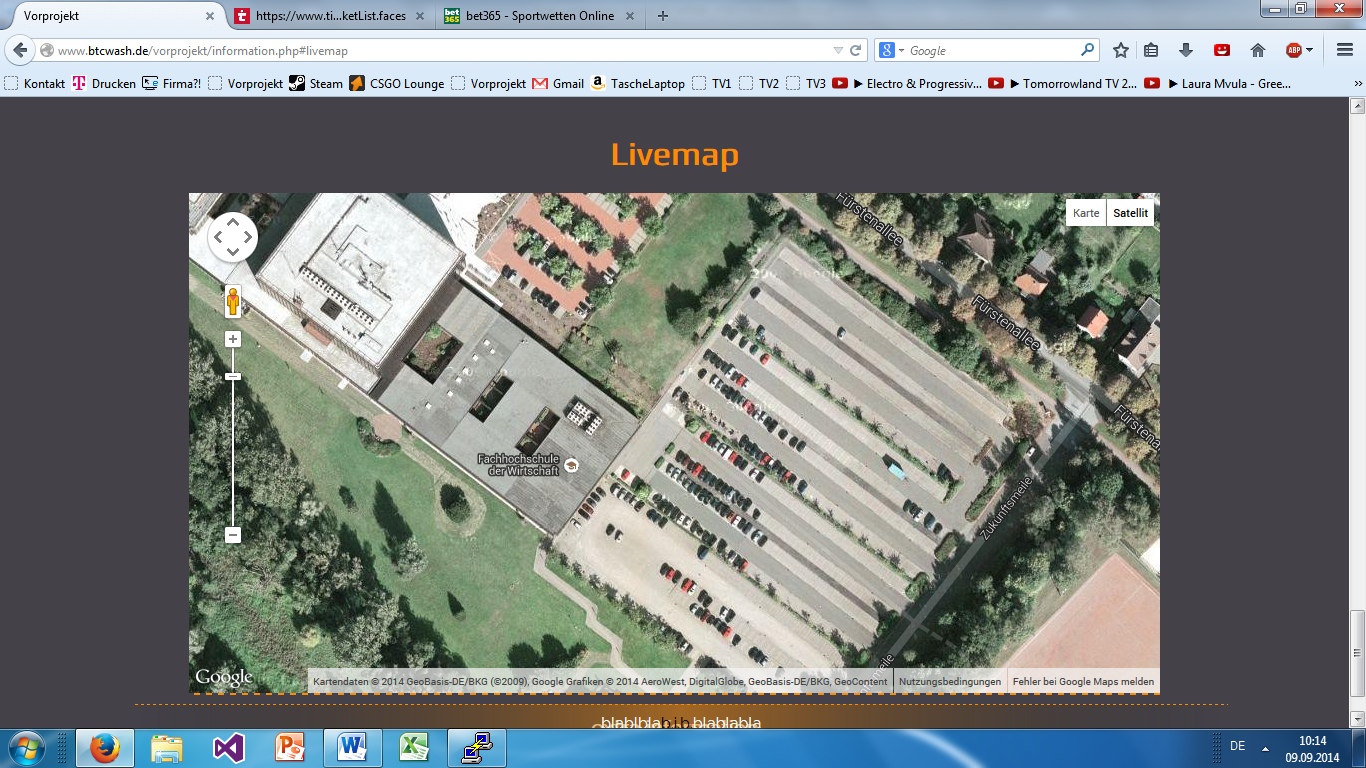
position: location,

title: title

});

markersArray.push(marker);

}



2.1.7. Kontaktformular

Der User kann über drei Textfelder seinen Namen, seine E-Mail und seine Nachricht eingeben. Über einen Button „Daten absenden“ wird der Inhalt der Felder abgeschickt und an die hinterlegte E-Mail verschickt. Diese E-Mail Adresse steht im MailSenden.php und kann dort auch geändert werden. Wird die Nachricht erfolgreich abgeschickt so lädt die Seite neu. Gibt der User keinen Inhalt in mindestens eines der Textfelder ein so gibt es eine Fehlermeldung und der User muss die Eingabe erneut tätigen.

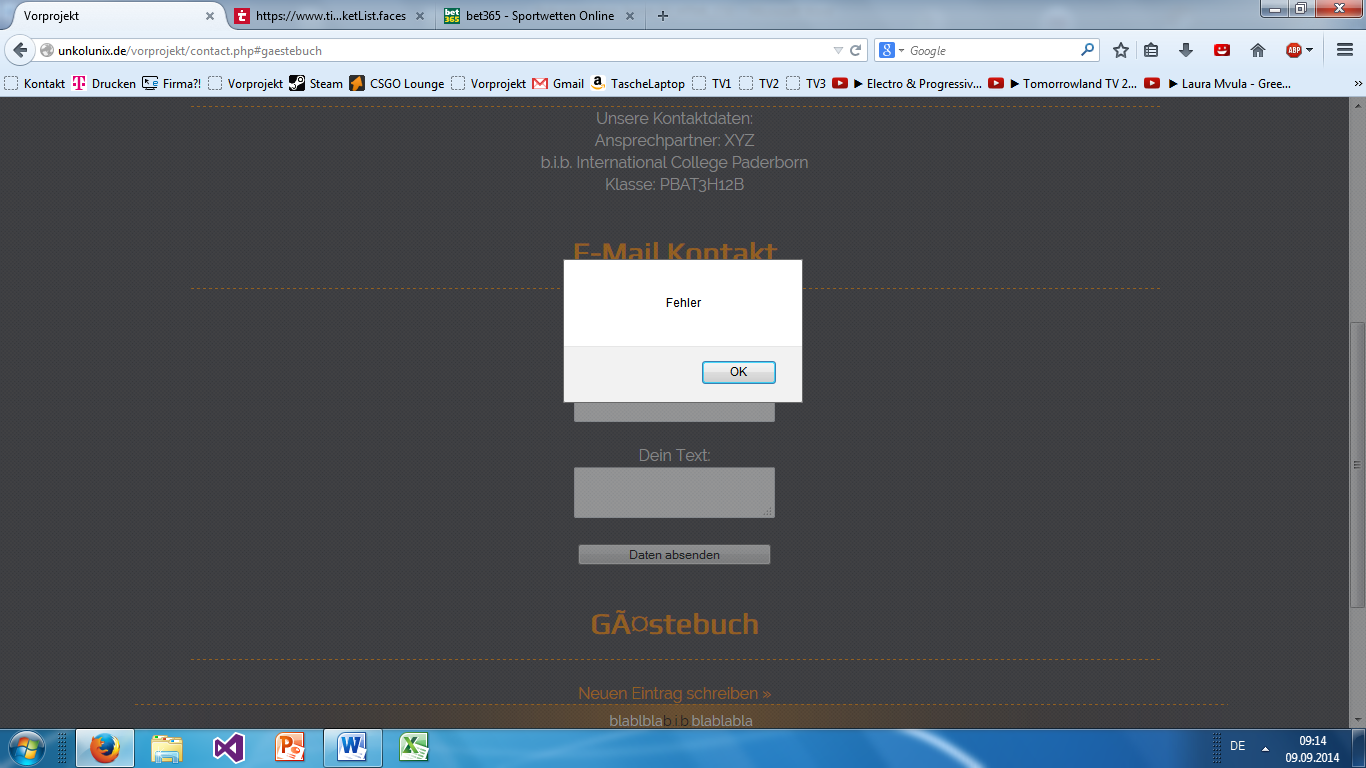
<?PHP

$text = "Kontaktaufname von: ".$\_POST[name]."\n Seine Nachricht lautet:".$\_POST[nachricht]."\n\n Antworten können Sie hier:".$\_POST[email];

mail("a@gmx.de","Email per Formular",$text);

header('Location: http://unkolunix.de/vorprojekt/contact.php'); exit;

?>



2.1.8. Inhalte

Die Inhalte auf der Startseite und auf der Seite „About“ wurden noch angepasst. Die Texte wurden selbst geschrieben.

2.1.9. Google Fonts

**WAS HAST DU GEMACHT????**

**CODE????**

**VORHER- NACHER???**

### 2.2 Probleme

-PHP funktioniert nicht lokal, deswegen muss auf dem Server getestet werden. Der Server musste gewechselt werden, da auf dem alten kein neueres PHP lief und somit keine Verbindung mit der REST-API hergestellt werden konnte.

-Die Fonts aus der Google Font API machten Probleme, da sie nicht geladen werden konnten, kurz vor Schluss konnte dies aber noch behoben

-Das Design wurde zwei bis drei Mal grundlegend geändert, da wir damit noch nicht zufrieden waren.

-Da zwischendurch der Server geändert wurde, wurde teilweise an zwei verschiedenen “Orten“ gearbeitet. Dadurch kam es zu kleinen Mehrfachbearbeitungen die unnötig Zeit in Kauf genommen haben, aber dies hielt sich in Grenzen und war somit kein Problem.

### 2.3 Auskommentierter Quellecode der Lösung

Der Quellcode ist im Repository unter **HTML\_CSS/Sprint\_02** zu finden.

Der aktuellste Stand der Internetseite kann derzeit über [www.btcwash.de/vorprojekt](http://www.btcwash.de/vorprojekt) aufgerufen werden.

2.4 **Tests mit Ergebnissen**

Die Download Button wurden getestet und werden heruntergeladen

**MUSS NOCH**

# 3Fazit

**3.1**

- Die Funktionen des Gästebuches und der Statistiken funktioniert. In Zusammenarbeit mit der API-Gruppe konnte auch noch eine LiveMap angefertigt werden. Durch die mehrfache Überarbeitung am Design und am Responsive Webdesign waren wir immer zufriedener mit dem Aussehen der Homepage und konnten damit die Seite neben den Funktionen abrunden.

**3.3** **Welche Ideen/Arbeitsweisen haben sich als gut bewiesen und sind zu empfehlen?**

- Aufgabenteilung ist wichtig, jedoch sollte man neue Dinge (PHP) zusammen angehen, da man sich gut ergänzen kann und somit das neue schneller lernen kann.