Inhaltsverzeichnis

[Projektziel und Umfang 2](#_Toc182490982)

[Kommunikation und Reporting 3](#_Toc182490983)

[Anleitung für die lokale Ansicht der Webseite 3](#_Toc182490984)

[Nutzerberechtigungen für die Spaghetti-Code Website 6](#_Toc182490985)

[Anmeldedaten in die Datenbank 7](#_Toc182490986)

[User Case: Entwicklung eines Webportals mit Registrierung und Bezahlfunktion 8](#_Toc182490987)

[Live-Schaltung einer Webseite auf einem Synology NAS 11](#_Toc182490988)

[Feedback 16](#_Toc182490989)

**Protokoll**

**Projektstart:** 7. November 2024  
**Projektende:** 14. November 2024  
**Projektteam:**

* Ron Kahihikolo
* Moayad Suleiman
* Muhammad Ibrahim Danno
* Viktor Betker
* Abdul Kader Zidan
* Thanh Nguyen

**Kunde/ Auftraggeber:** Danilo Arlt und Julian Martins

# Projektziel und Umfang

**Ziel des Projekts:**  
Die Erstellung einer Webseite ‚Spaghetti-Code‘ mit PHP, HTML und CSS zur Repräsentierung des eigenen Betriebs mit Onlinebestellfunktion.

**Projektumfang:**

Für das Spaghetti-Code Restaurant wird eine Webseite zur Verwaltung für Nutzer und Administratoren von Onlinebestellungen.

Das Integrieren von Bildern und die dazugehörigen optischen Anpassungen für das Interface der Seite sind erfordert. Die Bestell- und Registrierungsfunktion werden mit einer Datenbank verbunden. Bei der Anmeldung wird durch manuelle Eingabe der Daten in ein Register eingetragen. Dabei sind die eigenen Konten für die Endbenutzer individualisiert, sodass jeder einen Warenkorb durch Bestellungen erstellen kann.

Die Seite wird in mehreren aufgeteilt (Startseite, Menü, Kontakt, Impressum und Login) für eigene Prozesse oder zur Informationsbeschaffung.

Im Menü lassen sich die Produkte selektieren, der Preis wird im Warenkorb addiert und eine Bezahlfunktion ist oberflächlich verfügbar. Zusätzlich wird auf der Kontaktseite ein Formular für Kritik erstellt.

# Kommunikation und Reporting

Nachdem morgendlichen externen Meeting mit anderen Teams, folgte daraufhin das interne Daily-Scrum. Innerhalb des Daily Scrum’s wird die Analyse vom Vortag wahrgenommen und die Tagesplanung besprochen.

# Anleitung für die lokale Ansicht der Webseite

Ein Bild, das Text, Screenshot, Software, Display enthält.

Automatisch generierte Beschreibung

**Webserver einrichten:**   
Damit die HTML-, CSS- und PHP-Dateien korrekt angezeigt werden, benötigt man einen Webserver. Wie benutzen dafür das Programm „XAMPP“. Downloaden und installieren Sie „XAMPP“ (<https://www.apachefriends.org/>). Nach der Installation starten Sie die Anwendung als Administrator. Starten Sie den Apache-Server sowie den MySQL-Server.    
   
Legen Sie den Projekt-Ordner namens „Spaghetti-Code“ in das Verzeichnis „htdocs“ („xampp/htdocs“). 

Ein Bild, das Text, Software, Screenshot, Computersymbol enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung

**Datenbank einrichten:**   
Falls die Webseite Datenbankverbindungen nutzt, sind folgende Schritte notwendig:   
   
Öffnen Sie phpMyAdmin. Standardmäßig in einem beliebigen Browser erreichbar über:   
<http://localhost/phpmyadmin>   
   
Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

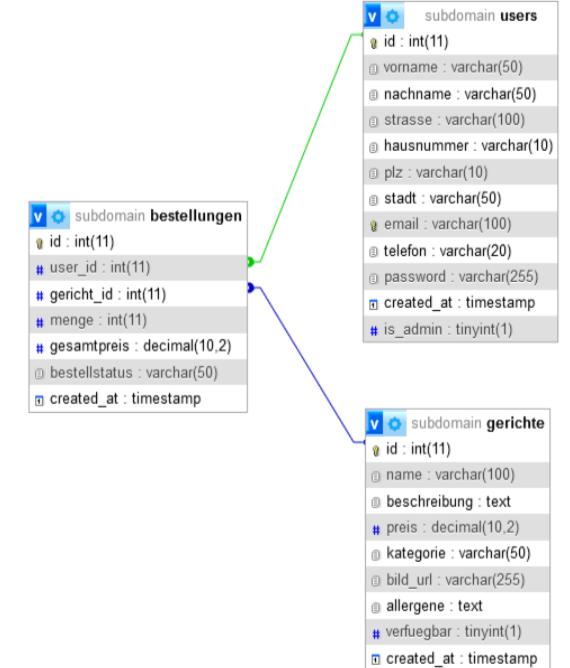
Automatisch generierte Beschreibung   
 

Erstellen Sie eine neue Datenbank namens „spaghetti\_code“.   
   
   
Importieren Sie die Datei „spaghetti\_code.sql“. Diese enthält die Struktir der Tabellen.   
   
   
   
Überprüfen Sie die Konfigurationsdatei für die Datenbankverbindung mit „config.env“ und passen Sie die Zugangsdaten an:   
   
```php   
   $host = 'localhost';   
   $db   = 'spaghetti\_code';   
   $user = 'root';   
   $pass = '';   
   ``` 

**Projektdateien bereitstellen:**   
Stellen Sie sicher, dass Ihr Projekt alle erforderlichen Dateien enthält.   
- HTML und CSS für das Frontend   
- PHP-Skripte für das Backend   
- Die .env-Datei mit den richtigen Einstellungen   
   
Diese sollen alle in das Webserver-Verzeichnis (z.B. „htdocs/Spaghetti-Code“). 

**URL aufrufen**   
Rufen Sie die Startseite im Browser auf:   
<http://localhost/spaghetti-code> 

# Nutzerberechtigungen für die Spaghetti-Code Website



Administrator:

* Nutzerlöschung und Bearbeitung

Nutzer:

* Bestellung
* Registrierung
* Einloggen
* Informationsbeschaffung
* Nutzerbearbeitung

# Anmeldedaten in die Datenbank

// SQL-Anweisung zum Einfügen des neuen Nutzers

            $sql = "INSERT INTO users (

                        vorname, nachname, strasse, hausnummer,

                        plz, stadt, email, telefon,

                        password, salt, created\_at

                    ) VALUES (

                        :vorname, :nachname, :strasse, :hausnummer,

                        :plz, :stadt, :email, :telefon,

                        :password, :salt, NOW()

                    )";

Dieser SQL-Code fügt einen neuen Benutzer in die Tabelle users ein. Die Daten für Vorname, Nachname, Adresse, E-Mail, Telefon, Passwort und Salt werden als Platzhalter (:vorname, :nachname, etc.) eingefügt. Der created\_at-Wert wird automatisch mit dem aktuellen Datum und der Uhrzeit durch die Funktion NOW() gesetzt. Der Code nutzt Platzhalter, um sicher Benutzereingaben einzufügen und schützt so vor SQL-Injektionen.

# 

# User Case: Entwicklung eines Webportals mit Registrierung und Bezahlfunktion

**1. Beginn des Projekts - 06. November 2024**

**Ziel:**  
Erstellung einer ersten HTML-Struktur als Übung für die Praxis.

* **Aktivität:**
  + Das Team bespricht die Aufgabenstellung und startet ein Brainstorming.
  + Basierend auf den Diskussionen wird eine Mindmap erstellt, die den Entwicklungsprozess skizziert.
  + Jedes Teammitglied erstellt eine eigene HTML-Datei als Übung.
  + Evaluierung der Arbeit wird für den nächsten Tag eingeplant.
  + **Herausforderung:** Die fehlende Routine mit HTML und CSS verzögert den Fortschritt, da einige Teammitglieder noch unsicher im Umgang mit diesen Technologien sind.

**2. Methodik und Plattform - 07. November 2024**

**Ziel:**  
Vorstellung der Methodik und Einführung in GitHub als Plattform zur Code-Verwaltung.

* **Aktivität:**
  + Viktor stellt die Methodik aus dem Vortag in der Sitzung mit den anderen Teams vor. Eindrücke anderer Teams werden gesammelt und in die eigene Vorgehensweise integriert.
  + In einer internen Daily-Scrum wird der Tagesablauf geplant und die Arbeit mit GitHub als Versionierungsplattform für das Projekt festgelegt.
  + **Herausforderung:** Fehlende Expertise in der Nutzung von GitHub erfordert eine intensive Einarbeitung. Ron übernimmt die Verwaltung von GitHub und führt das Team in die grundlegenden Funktionen der Plattform ein (Installation, Registrierung, Repository-Verwaltung).
  + **Ergebnis:** Das HTML-Konstrukt von Moayad wird als Grundlage für die Weiterentwicklung des Frontends gewählt.

**3. Weitere Arbeit an GitHub und Frontend - 08. November 2024**

**Ziel:**  
Fortschritte bei der Frontend-Entwicklung und Lösung von GitHub-Zugriffsproblemen.

* **Aktivität:**
  + Ibrahim stellt den Fortschritt der einzelnen Teammitglieder vor.
  + Weitere Schwierigkeiten beim Zugriff auf GitHub werden thematisiert, und durch Stephan können diese durch Eigenrecherche und gemeinsame Diskussion gelöst werden.
  + **Ergebnis:** Der Zugang zur GitHub-Plattform wird nun stabil gewährleistet. Ron verwendet seinen eigenen Server für das Projekt.
  + **Backend-Entwicklung:** Ibrahim beschäftigt sich mit der PHP-Integration, während die anderen Teammitglieder weiterhin am Frontend arbeiten, sodass die Webseite fast fertiggestellt ist.

**4. Backend-Entwicklung - 11. November 2024**

**Ziel:**  
Erstellung des Backend-Systems für die Registrierung und Optimierung der Frontend-Seite.

* **Aktivität:**
  + Ibrahim entwickelt die PHP-Skripte für die Benutzerregistrierung. Dabei werden die Eingaben des Nutzers in einer Datenbank gespeichert, wobei ein Filter verwendet wird, um schadhafter Code zu entfernen, und mit 'salt' & 'execute' vor möglichen Sicherheitslücken geschützt wird.
  + **Frontend-Optimierung:** Das Frontend wird zusätzlich optimiert, insbesondere die Darstellung des Warenkorbs.

**5. Datenbankintegration und Abschluss der Registrierung - 12. November 2024**

**Ziel:**  
Integration der Registrierung mit der Datenbank und Optimierung des Bestellprozesses.

* **Aktivität:**
  + Ibrahim schließt das Backend für die Benutzerregistrierung ab und integriert es mit dem Frontend.
  + Eine einfache Datenbank wird erstellt, die die Benutzeranmeldungen und Bestellinformationen speichert.
  + **Ergebnis:** Nach der Bestellausführung wird eine Bestellnummer generiert, die dem Nutzer als Referenz dient.

**6. Sicherheitsaspekte und Bezahlfunktion - 13. November 2024**

**Ziel:**  
Sicherstellung der Sicherheit und Integration der Bezahlfunktion.

* **Aktivität:**
  + **Sicherheitsoptimierung:** Die Zugangsdaten für das Projekt werden in einer .env-Datei gespeichert, um die Sicherheit zu erhöhen.
  + **Verbindung zur Datenbank:** Durch die Verwendung von include und config wird die Verbindung zum Online-Server und zur Datenbank sichergestellt. Ein zentraler Header für die Seite wird erstellt.
  + **Bezahlfunktion:** Parallel dazu wird die Bezahlfunktion oberflächlich abgeschlossen, wobei noch Arbeiten an der Optimierung der Bezahlprozesse zu erledigen sind.

**Ergebnis:**

Das Projekt hat in den letzten Tagen erhebliche Fortschritte gemacht. Das Frontend und das Backend sind mittlerweile größtenteils funktionsfähig. Die Registrierung ist abgeschlossen, und die Bezahlfunktion wurde teilweise implementiert. GitHub wird erfolgreich für die Codeverwaltung genutzt, und Sicherheitsaspekte, wie die Speicherung von Zugangsdaten und die Datenbankverbindung, wurden berücksichtigt.

**Ausblick:**

* Fertigstellung der Bezahlfunktion.
* Weitere Optimierung des Frontends und Backends
* Administratorenrechte mit Bestellungbearbeitung.

# Live-Schaltung einer Webseite auf einem Synology NAS

**Projekt Überblick**

Ziel des Projektes war es, eine Website zu erstellen, auf der Benutzer verschiedene Produkte bestellen, sich registrieren und ihre Bestellungen verfolgen können. Die technische Basis der Website wurde mit HTML, CSS, PHP und einer MariaDB-Datenbank geschaffen. Nach der erfolgreichen Entwicklung der lokalen Anwendung auf XAMPP (Localhost) war die nächste Herausforderung, die Website live zu schalten. Als Host wurde ein Synology NAS gewählt, das sich Ron befindet. Im Folgenden werden der Prozess der Live-Stellung sowie die wichtigsten Erfahrungen dokumentiert.

**1. Vorbereitung der Synology NAS als Webserver**

Um die Synology NAS als Hosting-Plattform nutzen zu können, mussten die grundlegenden Web- und Datenbankdienste eingerichtet werden. Zunächst wurde Web-Station auf dem NAS installiert, wobei Apache 2.4 als Webserver und PHP 8.0 als Programmiersprache gewählt wurden. Für die Datenbankverwaltung wurde MariaDB 10 zusammen mit phpMyAdmin verwendet, das eine übersichtliche Verwaltung der Datenbankstruktur ermöglicht. Mit dieser Installation wurde die Grundlage für eine stabile Webumgebung auf dem NAS-System geschaffen.

Ein Bild, das Text, Software, Computersymbol, Betriebssystem enthält.

Automatisch generierte Beschreibung

**2. Domainanbindung und Zertifikatserstellung**

Da bereits eine Domain vorhanden war, wurde diese direkt eingebunden. Um den Zugriff auf die Website sicherer zu machen, wurde ein SSL-Zertifikat von Let's Encrypt installiert, so dass die Seite über HTTPS aufgerufen werden kann. Außerdem wurde eine DynDNS-Einstellung bei meinem Domain-Provider aktiviert, die sicherstellt, dass meine dynamische IP-Adresse immer mit der Domain synchronisiert bleibt. Diese Schritte ermöglichten eine stabile und verschlüsselte Verbindung zur Website von überall.

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

**3. Einrichtung der Datenbank und erste Datei-Uploads.**

Die Datenbank und die erforderlichen Benutzer wurden auf dem Synology NAS eingerichtet. Dazu wurden die Tabellen für die Benutzer- und Bestellinformationen in MariaDB erstellt und Sicherheitsvorkehrungen getroffen, wie z.B. die Änderung der Standardports und die Aktivierung der Zwei-Faktor-Authentifizierung (2FA) für kritische Administrationszugänge. Mit Hilfe von FileZilla wurden die ersten Webseiten hochgeladen, so dass die Basisversion der Website online verfügbar war. Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Zahl, Software enthält.

Automatisch generierte Beschreibung

**4. Implementierung weiterer Sicherheitsmaßnahmen**

Ein wichtiger Schritt zur Absicherung des Projektes war der Umgang mit den Datenbankzugangsdaten. Um eine Hardcodierung der Zugangsdaten zu vermeiden, wurde eine .env-Datei außerhalb des Web-Ordners erstellt, die über den Terminalzugang der Synology NAS als Root-Benutzer eingerichtet wurde. Diese Datei enthält alle sensiblen Zugangsdaten und wurde in das Projekt integriert. Um die .env-Datei in die PHP-Seiten einzubinden, wurde eine zentrale config.php erstellt, die die Umgebungsvariablen auf allen Seiten verfügbar macht. Jeder PHP-Code, der auf die Datenbank zugreift, kann diese Datei einfach per include-Befehl laden. Diese Konfiguration bietet eine zusätzliche Sicherheitsebene gegen unbefugten Zugriff auf sensible Daten.

.env Datei

DB\_HOST=localhost   
DB\_NAME=meine\_datenbank   
DB\_USER=mein\_benutzer   
DB\_PASS=mein\_passwort

Ein Bild, das Text, Screenshot, Software, Betriebssystem enthält.

Automatisch generierte Beschreibungconfig.php Datei

**### 5. Fertigstellung und Live-Schaltung der Website**

Nach intensiven Tests der Funktionalität - vom Login über den Bestellprozess bis hin zur Datenbankanbindung - wurde die Website offiziell live geschaltet. Die Seiten wurden in VS Code entwickelt und anschließend mit FileZilla auf den Server übertragen. Die Anwendung ist nun unter der Domain öffentlich zugänglich und erfüllt die Anforderungen des Projektes.

Ein Bild, das Text, Essen, Kochkunst, Fastfood enthält.

Automatisch generierte Beschreibung

# Feedback

Die Prozesse zur Erstellung einer Web-Seite und des Live-Stellens auf einem Synology NAS war ein wertvolles Lernprojekt und half dabei, grundlegende Webserver- und Sicherheitsprinzipien in der Praxis anzuwenden. Die Verwendung einer .env-Datei zur Sicherung der Zugangsdaten und die Konfiguration von DynDNS und HTTPS auf einem privaten NAS bieten eine gute Grundlage für zukünftige Projekte.

**Zusammenfassung**

**Positiv hervorzuheben:**

* Definierte Ziele und Projektstruktur.
* Regelmäßige Abstimmung und gute Kommunikation im Team.
* Sicherheit und Schutz der Daten durch Filter und Platzhalter.
* Strukturierte Anleitung zur lokalen Ansicht der Webseite.

Aufgrund mangelnder Erfahrung zeitintensiv. Optimierungen werden durch Erfahrungen erst generiert.