# Semesterarbeit Webarchitekturen: Thema -> Next.JS

Ersteller: Lorant Gulyas

### Vorwort

Diese Dokumentation hat zum Ziel die Installation einer kleinen Todo Liste als Webseite zu begleiten, Hierbei werden alle wichtigen Schritte beschrieben, um an ende eine funktionierende Anwendug mit Datenbank Anbindung zu haben. Diese Dokumentation wurde auf einem MacOS Betriebssystem erstellt. Die Befehle können auf einer Linux system leicht abweichen. Die Logik der Befehle ändert sich jedoch nicht.

Die Finale Seite Seite kann man unter folgenden URL sehen:

Todo Liste für Web Architekturen

Den dazugehörigen Source Code findet man auf Github:

**Github** 

## Technische Vorraussetzungen in der Entwicklungsumgebung

### Betriebssystem

Es wird eine Linux oder Mac OS X Umgebung benötigt. Sofern man auf einem Mac oder Linux arbeitet sind alle notwendigen Tools nativ verfügbar.

Sofern Windows verwendet wird, muss entweder eine Virtuelle Maschine mit einer Linux Instanz installiert werden, oder ein WSL System in der Shell angelegt werden. Unter folgenden Links findet man eine Anleitung zum installieren einer Virtuellen Maschine sowie WSL Subsystem.

Ubuntu als Virtuelle Maschine installieren

WSL auf Windows installieren

#### Node:

eine Instanz des Node Framework muss am System installiert sein.

Um zu prüfen, ob wir Node installiert haben, können wir den folgenden Befehle absetzen.

#### node --version

Sofern Node bereits installiert ist, erhalten sie eine Information übe die Version. Diese sollte mindestens mit einer 18 beginnen. Sollte Node nicht installiert sein, wird eine Fehlermeldung erscheinen, dass der Befehl nicht erkannt wurde.

in diesem Fall muss Node erst installiert werden. Das kann man etwas über Homebrew bei MacOS machen.

#### brew install node

Bei Linux kann man die Pakete großteils mit Paketmanagern installieren,

zB.

yum install nodejs14

oder

pacman -S node npm

#### VsCode:

Der Code Editor ist unabdingbar, um den Source Code der Seite zu schreiben. Das Tool ist Gratis und kann unter folgenden URI heruntergeladen werden.

#### Vs Code Editor

Einfach installieren, und danach starten. Im IDE der Sotware hat man die Möglichkeit Plugins zu installieren, sowie Accounts zu verknüpfen. Wir werden diese Funktion nutzen, um VsCode mit Github zu verbinden, um das Updaten der App zu erleichtern. Links unten im Editor Fenster kann man ein Avatar sehen, übe den man sich auf Github anmelden kann, sofern das entsprechende Plugin installiert wurde.

Verwendete Plugins:

- GitHub Pull Request
- Git History
- GitLens
- Git Graph

## Benötigte Accounts

#### Github:

Dieser Acount wird die Schaltzentrale für unsere Versionsverwaltung. Wir werden unsere App auf Github laden, von wo es sich mit unseren Web Server synchronisiert wird. Hierdurch ist es möglich Änderungen direkt von unseren Editor bis auf die Webseite per Konpfdruck einzuspielen.

Um uns mit Vscode verbinden zu können, müssen wir uns zuerst registrieren.

### Github Login Seite

Hier einfach rechts oben auf Sign Up klicken und ein neues Konto anlegen. Eine Empfehlung ist es ein schon bestehenden Account zu verwenden. Hierzu bietet sich z.B. ein Google Account.



#### Vercel:

Zusätzlich zu Github benötigen wir auch einen Account auf Vercel. Das gute ist, dass man hier direkt seinen Github Zugang verwenden kann. Einfach beim registrieren Github wählen, und die Verbindung bestätigen.



Sobald diese beiden Accounts angelegt wurden, kann auch schon mit der Erstellung der Webseite begonnen werden.

#### MongoDB:

Als letztes benötigen wir noch einen Account bei MongoDB, das ist eine Dokumentenbasierte Datenbank, auf der wir uns eigene Strukturen ganz leicht anlegen können, damit wir diese später über die Webseite abrufen können. Hier haben wir auch die Möglichkeit, und einen kostenlosen Account anzulegen. Auch hier haben wir die Möglichkeit uns mit unseren Google Account anzumelden.





Durch diese Massnahmen haben wir eine ganz einfache Möglichkeit direkt die Datenbank anzulegen, mit der wir später auf der Plattform arbeiten.

Wir können auch direkt einen Zugang für unsere spätere Schnittstelle definieren. Hierfür wird ein Benutzer angegeben, und definiert, welche Zugriffsrechte dieser bekommt. Zusätzlich können wir IP Adressen definieren, über welche der Zugriff auf die Datenbank ermöglicht wird.

## Initialisierung eines neuen Projektes

Um ein neues Projekt zu starten müssen die folgenden Schritte berücksichtigt werden.

Starten wir die Virtuelle Maschine oder öffnen das WSL Terminal.

Wechseln wir in den Arbeitsordner, in dem das Projekt angelegt werden soll. Das kann z.B. der Ordner sein, in dem alle Github Projekte abliegen.

hier verwenden wir den folgenden Befehl:

#### cd "xx"/Github/

Wobei xx für die Ordnerstruktur steht, wo der Ordner abgelegt ist.

Im Ordner führen wir den folgenden Befehl aus.

#### npx create-next-app weba-lorant-weba

Hierbei ist **npx** der Befehl, create-next-app der Parameter und "weba-uebung-lorant" der Projektname. Nachdem Ausführen des Befehls werden einige Fragen zur Art des Projektes gestellt. Hierbei wird z.B. abgefragt, ob das Projekt Typescript verwenden soll, Tailwind Css verwendet wird und ähnliches. Im folgenden Bild sind die passenden Einstellungen für das aktuelle Projekt abgebildet. Hier sieht man auch die Ausgabe des befehls in der Konsole.



Nachdem das Projekt erstellt wurde, kann auch schon das passende Git Repository erstellt werden. (Wir haben die erweiterte Version von Git verwendet. (Die erweiterung Git Flow ist hierbei aber Opional. Sie dient dazu verschiedene Stränge für die Entwicklung bereitzustellen.))

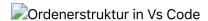
Zuerst mit cd "Projektname" in das entsprechende Verzeichnis wechseln und den folgenden Befehl ausführen.

### git flow init

Anbei die Ausgabe des Befehls, mit den möglichen Parametern, die man angeben kann.



Wenn alles erfolgreich geklappt hat, dann sollten folgende Dateien zu sehen sein, wenn man den neu angelegten Ordner mit Visual Studio Code öffnet.



Wenn wir dieses Bild sehen, können wir die Applikation zum ersten mal starten.

#### Start den Webserver

Um den Webserver zu starten müssen wir sicherstellen, dass wir im Projektordner sind. Das können wir ganz leicht mit: pwd prüfen.

Sollten wir wiedererwarten nicht im richtigen Ordner sein, bitte zuerst in den Projektordner wechseln.

Wenn das erledigt ist, dann kann man folgenden Befehl ausführen:

#### npm run dev

Hiermit wir eine Entwicklungsinstanz auf unseren Rechner gestartet. Wenn der Befehl ausgeführt wurde, bekommen wir eine Benachrichtigung, dass der Server auf http://localhost:3000 gestartet wurde. Und ggf. Fehlermeldungen, die eine Ausführung behindern.



Wenn alles richtig gemacht wurde, bekommen wir in der letzten Zeile die Nachricht, dass "compiled client and server succesfully"

## Wir haben es geschafft, wir haben unsere erste eigene Next.Js App angelegt

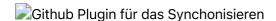
Wir können die Webseite öffnen, indem wir die Localhost Adresse in unseren Browser öffnen

Bevor wir mit den Anpassungen der Seite beginnen, sollten wir zuerst die Verbindung mit Vercel anlegen, damit später die Aktualisierung der App direkt aus VsCode passieren kann.

Da die genaue Verbindung von Github und Vs Code ein eigenes Kapitel darstellen würde, wird hier die Offizielle Anleitung verlinkt, wo genau beschrieben ist, welche Schritte man durchführen muss.

#### Github Verbindung mit Vs Code

Wenn alles gepasst hat, sollten wir auf der linken Seite, beim Punkt Quellencode Verwaltung eine Zahl sehen -> Diese zeigt an, wie viele Dateien mit dem Server zu synchronieren sind. Den menüpunkt anklicken, eine Nachricht verfassen, um für später das Update zuordnen zu können, und dann auf Commit klicken. Damit wurde das Paket mit Github Synchorniert.



nachdem Sync mit Github müssen wir das Projekt noch auf Vercel veröffentlichen. Sofern wir bei der Acount registrierung alles richtig gemacht haben, können wir das Github Projekt sehr leicht anlegen. Im Dashboard von Vercel klicken wir hierzu auf **Add New -> Project** im rechten Bereich der Bildschirms.

Vercel projekt anlegen

Auf dem nächsten Bildschirm erscheint eine Liste unserer Github Repositories, mit jeweils kleinen Symbolen, sofern der passende Typ gefunden wurde. Hier erscheint jetzt unser neues Projekt, mit einem kleinen (N) logo als Symbol für ein Next.JS System. Einfach auf Import neben dem Namen klicken, um das deployment zu starten.

Vercel projekt anlegen

Am folgeBildschirm können wir einige notwendige Details einstellen, und auch Environnement Variablen anlegen. Diese benötigen wir später auch für die Datenbank Anbindung.

Vercel projekt anlegen

Bevor wir weitermachen, sollten wir einen kurzen Blick auf die wichtigsten Komponenten des Projektes werfen.

- /src/pages/index.js --> Das ist unsere Startseite, die im Browser geöffnet wird, wenn wir die URL aufrufen.
- /src/styles/global.css --> Hier werden alle Designangaben getätigt, die sich Global auswirken sollen
- **package.json** --> Hier werden alle Komponenten gelistet, die in der Instanz laufen und benötigt werden.
- /public/ --> Hier sind alle Dateien, die frei zugänglich sein sollen.

Bevor wir die Index.js editieren werden, müssen wir erst Vorkehrungen treffen, um die Datenbank anbindung zu aktivieren. Hier gibt es bestimmte Routen, die eingehalten werden müssen, damit die Verbindung zur Datenbank auf funktionieren kann.