

WebEngineering

Bearbeitete Aufgaben:

1. **Individuelles Nutzerlogin**
 - Nutzerlogin
 - Nutzerregistrierung
 - Zugriffskontrolle (Einteilung in rich und poor)
 - Speicherung von Nutzernamen und gehashtem Passwort in Datenbank (MongoDB Atlas)
2. **Abfrage von Wetterdaten**
 - Openweathermap API
 - Darstellung auf "Dashboard" - Nur für rich user sichtbar
3. **Text to speech, Vorlesen von Wikipedia-Daten**
 - Clientseitige Implementierung
 - Wikipedia-Daten nur für rich user sichtbar

JQuery habe ich an einigen Stellen genutzt, warhscheinlich aber nicht genug. Wenn ich es genutzt habe, dann nur, weil ich anders eine Menge Code hätte schreiben müssen. Für ein Beispiel siehe register.js.

Installation

1. Diesen Ordner herunterladen oder `git clone` in gewünschtes Directory
2. Gehe in übergeordneten Ordner 2518151
3. `cd Server` - Egal wie, hauptsache man befindet sich nach diesem Schritt im "Server"- Verezichnis
4. Im Terminal `npm install` eingeben
5. Im Terminal `node runServ.js` eingeben
6. Browser öffnen
7. `localhost:3000` aufrufen

Anmeldung

Um zu Testen, habe ich ein System eingeführt, in dem nur Nutzer mit dem Wort "rich" im Nutzernamen zugriff auf das "Dashboard", also den dynamischen Content haben. Zum Testen habe ich zwei Nutzer bereits angelegt:

poor Nutzer:

```
username: poor
password: pooruser
```

rich Nutzer:

```
username: rich
password: richuser
```

Struktur

Der Server startet via `runServ.js`. Für die Authentifikation und die Datenbankanbindung gibt es Utility-Files. Diese sind in `authUtil` gespeichert. `database.js` sorgt für eine Datenbankanbindung, `User.js` deklariert das Nutzerschema für die Datenbank und `signup.js` stellt Funktionen bereit, die beim Einloggen und Registrieren aufgerufen werden.

Die HTML-Seiten für die jeweilige Funktion liegen in `public/htm`. Alle HTML-Files, die auf keinen Fall ohne Registrierung an die Öffentlichkeit gelangen dürfen, liegen in `Server/htm`.

Erfahrungsbericht und Datenfluss

Alle erforderlichen Dokumente befinden sich im Ordner `PDFs`

Dateien im Überblick:

js/dashboard.js: Javascript für dynamischen Teil. Enthält Text-To-Speech und Wetterdatenanzeige

js/index.js: Javascript für Main-Page. Enthält u.a. Lösungen für Aufgaben während der Vorlesung.

js/login.js: Enthält Javascript für das Login. Im Grunde nur ein Form-Submit

js/register.js: Enthält Javascript für die Registrierung. Hier insbesondere noch double-checking vom Passwort.

Alle CSS-Stylesheets tragen den Namen der zugehörigen HTML-Datei. **Wichtig:** `login.htm` und `signup.htm` haben das gleiche Design. Somit gilt `signup.css` für beide!