

Übung 3 – Planung Software

(Maximilian Kaltenreiner – if23b013)

Projektbeschreibung

Die Entwicklung einer Web-basierten Highscore-Listen-Applikation (Single-Page-Applikation). Diese Applikation umfasst einen Login- und Registrierungsmechanismus sowie das Einsehen einer Highscore-Liste für ein fiktives Spiel. Dabei sollen nur eingeloggte Benutzer die Berechtigung haben, die Highscore-Liste aufzurufen. Aufgeteilt wird die Applikation in Frontend, Backend und einer Datenbank.

Verwendete Technologien

- **Frontend:**
 - **Angular:**
 - Framework zur Erstellung der Single-Page-Applikation.
 - Erstellung der Web-Page mittels Angular Components (Login, Registration, Highscores)
 - **Angular Material:**
 - Library für benutzerfreundliche und optisch ansprechende UI-Komponenten.
 - Verwendung von Forms, Navigationsmenüs, Cards, Icons, etc.
- **Backend:**
 - **Node.js:**
 - Plattform für die Entwicklung der Web-Applikation.
 - **Express.js:**
 - Web-Framework für Node.js, um das Backend der Web-Applikation zu entwickeln.
- **Datenbank:**
 - **MongoDB:**
 - NoSQL-Datenbank zur Speicherung der Highscore-Liste und der Benutzerdaten.

Die Komponenten

1. Login-Komponente:

- **Frontend:**
 - Formulareingabe für E-Mail und Passwort
 - Validierung der Eingabedaten (gültiges E-Mail-Format, Passwörter mindestens 8 Zeichen lang)
 - Kommunikation mit dem Backend zur Authentifizierung
- **Backend:**
 - Endpunkt zur Annahme der Login-Daten und Authentifizierung des Benutzers
 - Token-Generierung zur Sitzungshandhabung

2. Register-Komponente:

- **Frontend:**
 - Formular zur Eingabe von E-Mail, Passwort, Firma, Straße, Stadt und PLZ
 - Validierung der Eingabedaten (gültiges E-Mail-Format, Passwörter sind mindestens 8 Zeichen lang, Validierung des Passwortes durch erneute Eingabe und vergleichen der Eingaben, Firma wird als „FH Technikum Wien“ vorgegeben und darf nicht verändert werden, E-Mail und Passwort sind Pflichtfelder)
 - Kommunikation mit dem Backend zur Registrierung des Benutzers
- **Backend:**
 - Endpunkt zur Annahme der Registrierungsdaten
 - Überprüfung der Eingabedaten (ist bereits ein User mit der übergebenen E-Mail-Adresse vorhanden)
 - Speicherung der Benutzerdaten in der MongoDB-Datenbank

3. Highscore-Komponente:

- **Frontend:**
 - Tabelle zur Anzeige der Highscores (nur für eingeloggte Benutzer sichtbar)
- **Backend:**
 - Endpunkt zur Abfrage der Highscore-Daten (nur für authentifizierte Anfragen)
 - Speicherung und Verwaltung der Highscore-Daten in der MongoDB-Datenbank

Entwicklungsprozess

1. Initial Setup:

- Einrichtung des Angular-Projekts und Installation von Angular Material

2. Frontend-Entwicklung:

- Erstellung der Login- und Register-Komponenten mit Angular
- Erstellung eines Navigationsmenüs (/login, /register, /highscores)
- Implementierung der Highscore-Komponente und Integration mit dem Backend

3. Backend-Entwicklung:

- Einrichtung des Node.js-Servers mit Express.js
- Konfiguration der MongoDB-Datenbank und Verbindung über Mongoose mit Express.js
- Implementierung der Authentifizierungs- und Registrierungslogik mit Express.js
- Implementierung der Highscore-API-Endpunkte
- Sicherstellung der Kommunikation zwischen Angular-Frontend und Express-Backend

4. Testen der Applikation:

- Überprüfung der Funktionalitäten der Applikation

Kommunikationsfluss und Ressourcenverwaltung

Frontend-Backend-Kommunikation

• Nutzung von HTTP-Requests (GET, POST):

- **Login:** POST-Request zur Übermittlung von E-Mail und Passwort an das Backend zur Authentifizierung.
- **Registration:** POST-Request zur Übermittlung der Registrierungsdaten an das Backend zur Erstellung eines neuen Benutzers.
- **Highscores:** GET-Request zum Abrufen der Highscore-Daten, nur für authentifizierte Benutzer.

• Tokens für die Authentifizierung und Sitzungshandhabung:

- Bei erfolgreichem Login wird ein Authentifizierungs-Token generiert und an den Client zurückgesendet.
- Dieser Token wird für nachfolgende Anfragen an geschützte Endpunkte im HTTP-Header mitgesendet.
- Das Backend validiert den Token, um sicherzustellen, dass die Anfrage von einem authentifizierten Benutzer stammt.

Ressourcenverwaltung

- **Speicherung in MongoDB:**
 - **Benutzerdaten:** Speicherung von E-Mail, Passwort, Firma, Straße, Stadt und PLZ.
 - **Highscores:** Speicherung der Benutzer-Highscores mit Referenzen auf die Benutzer.
- **Validierung der Benutzereingaben:**
 - **Frontend (Angular Forms):** Validierung von E-Mail-Format, Passwortlänge, Passwort-Wiederholung und festen Firmennamen.
 - **Backend:** Zusätzliche Validierung der Daten und Prüfung, ob die E-Mail-Adresse bereits existiert, sowie Token-Validierung bei der Abfrage der Highscores.

Beschreibung der Abläufe

- **Login:**
 - User -> Browser : Open Login Page
 - Browser -> Server : Send Login Request (Email, Password)
 - Server -> Database: Validate Credentials
 - Database -> Server : Return User Data/Token
 - Server -> Browser : Send Auth Token
 - Browser -> User : Display Success/Error Message
- **Registration:**
 - User -> Browser : Open Registration Page
 - Browser -> Server : Send Registration Data (Email, Passwort, Firma, Straße, Stadt, PLZ)
 - Server -> Database : Check if Email Already Exists
 - Database -> Server : Return Existence Status
 - Server -> Database : Save New User Data (if email does not exist)
 - Database -> Server : Return Confirmation
 - Server -> Browser : Send Registration Confirmation
 - Browser -> User : Display Success/Error Message
- **Highscores:**
 - User -> Browser : Request Highscore Page
 - Browser -> Server : Send Highscore Request (with Auth Token)
 - Server -> Database : Validate Auth Token
 - Database -> Server : Return Validation Status
 - Server -> Database : Fetch Highscore Data (if token is valid)
 - Database -> Server : Return Highscore Data
 - Server -> Browser : Send Highscore Data
 - Browser -> User : Display Highscore List