

# Abschlussprojekt Webtechnologien SS20

Alexander Bazo, Martin Kocur

## Fan-Fiction-Explorer

### Aufgabenstellung

Im Rahmen Ihres Abschlussprojektes implementieren Sie eine Webanwendung, die der Erfassung und Darstellung eines Fan-Fiction Korpus dient. Mit Hilfe der Anwendung sollen NutzerInnen einen Überblick über die zur Verfügung gestellten Fan-Fiction Inhalte erhalten und diese bei Bedarf mit textuellen Kommentaren und Notizen ergänzen können. Zur Implementierung des Projekts setzen Sie die im Kurs erworbenen Fähigkeiten und die dort besprochenen Technologien ein. Das Projekt besteht aus zwei Teilen:

1. Der erste Teil liefert die Datenbasis für den zweiten Teil des Projekts. Hierfür sollen Sie mit Hilfe des Skripts *Fan-Fiction Data-Importer* eine Datenbank mit den Fan-Fiction Stories und den dazugehörigen Metadaten erstellen. Gestalten Sie ein geeignetes Datenbankschema und halten Sie sich an die im Kurs besprochenen Normalisierungsregeln.
2. Der zweite Teil des Projekts ist die eigentliche Webanwendung, in die Sie die Datenbank aus Teil 1 einbinden und die im Folgenden beschriebenen Anforderungen implementieren.

### Anforderungen Teil 1

Importieren Sie **min. 100** der in diesem Archiv bereitgestellten *Fan Fiction*-Geschichten in eine SQLite-Datenbank. Entwerfen Sie dazu zuerst ein passenden Datenbankschema, um die Geschichten vollumfänglich in der Datenbank zu repräsentieren. Beachten Sie die im Kurs besprochenen Normalisierungsregeln und denken Sie breit an dieser Stelle an die zusätzlichen Anforderungen, die sich aus Aufgabenteil 2 für das Schema ergeben (nutzergenerierte Kommentare und Notizen). Verwenden Sie für den automatischen Import das Skript, das wir Ihnen hier bereitstellen. **Diesen, in JavaScript bzw. für Node.js geschriebenen, Importer müssen Sie an einigen Stellen noch selbständig anpassen.** Der angepasste Skript ist Teil des Projekts und wird zusammen mit der Lösung aus Aufgabenteil 2 eingereicht.

### Anforderungen Teil 2

Das wesentliche Ziel des Projektes ist es, die bereitgestellten Daten über Ihre Datenbank in das System einzupflegen und im Anschluss für die NutzerInnen – aufbereitet – zugänglich zu machen. Die zu implementierende Anwendung erfüllt dabei zwei grundlegende Funktionen:

1. Die Informationen werden den NutzerInnen strukturiert zugänglich gemacht. NutzerInnen erhalten unter anderem einen Überblick über die Autoren, die Charaktere und den Inhalt der Geschichten.
2. Über eine entsprechende graphische Schnittstelle können NutzerInnen die Inhalte um zusätzliche Informationen ergänzen. So können Kommentare und Notizen zu den jeweiligen Fan-Fiction Inhalten hinzugefügt und in einer Datenbank persistiert werden.

Ihre Applikation benötigt mindestens drei Seiten:

1. Eine Startseite, die beim Start Ihrer Anwendung angezeigt wird.
2. Eine Übersichtsseite, bestehend aus einer Listenansicht, die alle Fan-Fiction Inhalte darstellt, und einer Detailansicht, die Informationen über die entsprechende Fan-Fiction Geschichte zur Verfügung stellt. Dabei sollen alle wichtigen Metadaten (z.B. Titel, Autor, Charaktere, Fangemeinde etc.) in der Detailansicht angezeigt werden. Ein Diagramm zeigt die Häufigkeitsverteilung der wichtigsten Wörter der entsprechenden Geschichte an.
3. Eine Lese- und Kommentarseite, die die entsprechende Fan-Fiction Geschichte darstellt und es den NutzerInnen ermöglicht, über ein Eingabefeld öffentliche Notizen und Kommentare zu der jeweiligen Geschichte zu verfassen. Bereiten Sie die Geschichten entsprechend auf, indem Sie z.B. "junk characters", unnötige Leerzeichen und sonstige Elemente, die den Textfluss stören, entfernen. Natürlich müssen Sie keine Rechtschreibfehler oder sonstige Fehler, die von den Autoren gemacht worden sind, verbessern. Zusätzlich zur Geschichte werden auch die nutzergenerierten Kommentare auf dieser Seite angezeigt. Außerdem können Sie zwischen den einzelnen Geschichten wechseln ohne dabei zur Übersichtsseite zurückspringen zu müssen.

Orientieren Sie sich für die Gestaltung der drei Seiten an den im Folgenden dargestellten Skizzen der Benutzeroberfläche.

## **Hinweise zum Vorgehen**

### **Skizzen der Benutzeroberfläche**

Die folgenden Skizzen beschreiben die wesentlichen Bestandteile der Benutzeroberfläche. Bei der Implementierung der Anwendung müssen Sie sich an diesen Vorgaben orientieren. Bitte beachten Sie dabei jedoch, dass die Skizzen lediglich die Struktur und den Inhalt der Benutzeroberfläche beschreiben, nicht aber deren finales oder vollständiges Aussehen.

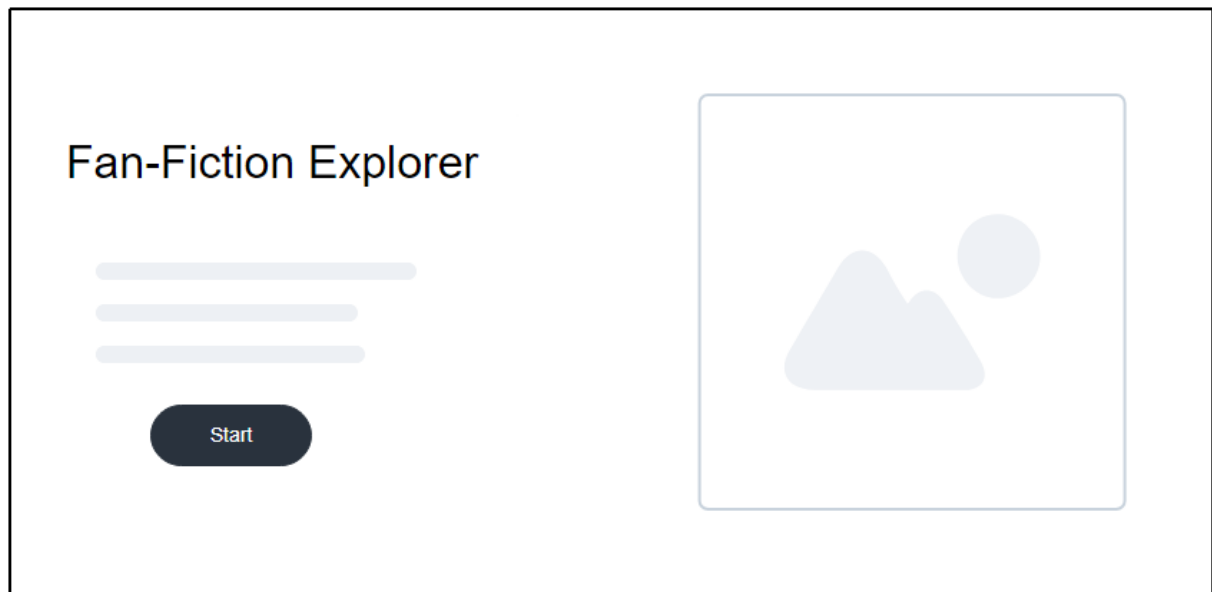


Abbildung 1: Startseite

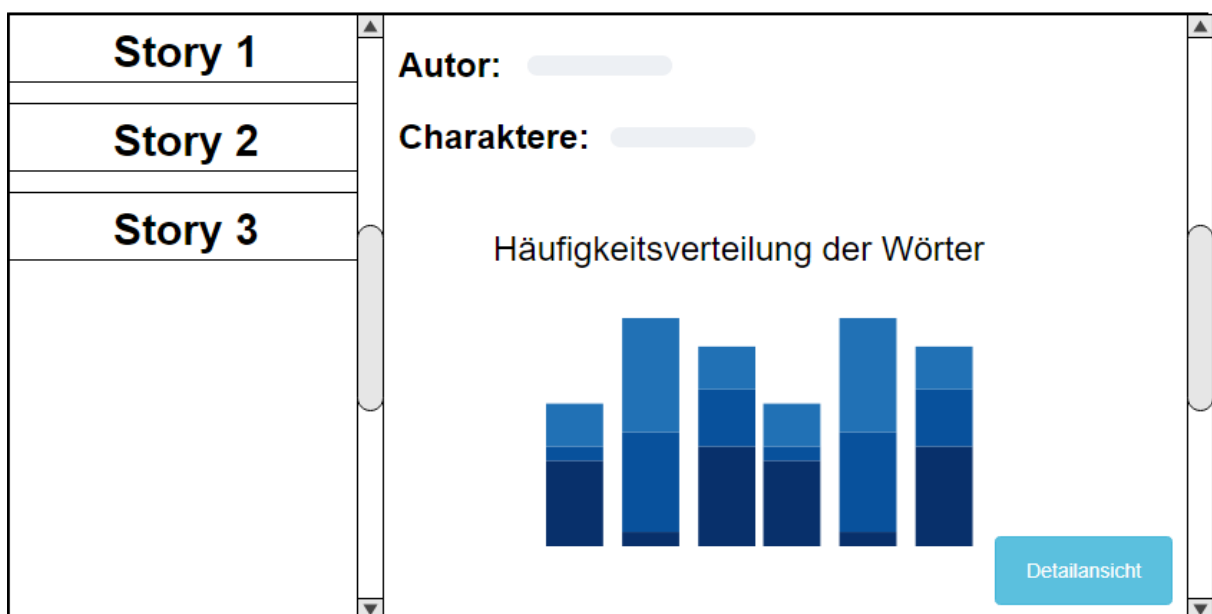


Abbildung 2: Übersichtsseite



Abbildung 3: Lese- und Kommentarseite

### Denkbare Benutzerinteraktion

1. Über die Startseite können NutzerInnen über einen Button-Klick zur Übersichtsseite gelangen.
2. Über einen Klick auf die Listenelemente erhalten NutzerInnen die entsprechenden Informationen zu der dazugehörigen Fan-Fiction Geschichte.
3. Über einen Klick auf einen Button springen NutzerInnen zur Lese- und Kommentarseite, die Eingabefelder für Kommentare und Notizen enthält. Zusätzliche Buttons ermöglichen einen Wechsel der Geschichten.

### Technischer Rahmen

Die Implementierung der Anwendung erfolgt auf der im Kurs gelernten Technologien. Verwenden Sie das Ihnen zur Verfügung gestellte *Starter-Paket*, um die Webanwendung (Teil 2 des Projekts) zu implementieren. Das Starter-Paket enthält die Ihnen bekannten node.js Module (`index.js`, `Database.js`) sowie Module, die für die clientseitigen Funktionalitäten der Anwendung zuständig sind (`app.js`, `DatabaseClient.js`). Nutzen sie die bestehenden Module und entwickeln sie diese weiter, und erweitern Sie die Anwendung darüber hinaus auch um zusätzliche Module, um Ihre Applikation modularisiert aufzubauen.

Die zu persistierenden Daten werden in einer SQLite-Datenbank gespeichert. Zur Gestaltung der Benutzeroberfläche verwenden Sie HTML & CSS. Die graphische Aufbereitung auf der Übersichtsseite können Sie mit der im Kurs besprochenen Javascript-Bibliothek *Chart.js* gestalten.

## Projektblauf

### Vorbereitung

- Laden Sie den Quellcode des Starter-Pakets über diesen Link herunter.
- Führen Sie im Projektverzeichnis, in einer Kommandozeile, den Befehl `npm install` aus, um alle notwendigen Abhängigkeiten zu installieren.
- Erstellen Sie einen Ordner `data` im Projektverzeichnis und kopieren Sie die von Ihnen generierte Datenbank aus Teil 1 dort hinein.

### Hilfestellung und Feedback

Sollten Sie während der Arbeit an Ihrem Projekt Fragen haben oder Hilfestellungen benötigen, können Sie uns per Mail kontaktieren. Zusätzlich dazu bieten wir Ihnen folgende Termine an, um Ihnen die Möglichkeit zu geben, den Start und die Finalisierung des Projekts zu besprechen. Wir werden für beide Zeiträume Terminumfragen verschicken und mit Ihnen einen Zeitslot festlegen.

**Zwischen 3. und 6. August:** Projekt-Kickoff zum Klären offener Fragen und zum Abstimmen des Vorgehens mit Alexander Bazo.

**Zwischen 7. und 11. September:** Finale Besprechung des aktuellen Stands vor der Abgabe mit Martin Kocur.

### Abgabekriterien

Laden Sie Ihr gesamtes Projekt bis spätestens 30.09.2020 (23:59 Uhr) als zip-komprimierten Ordner auf GRIPS hoch. Denken Sie daran, dass Sie auch mehrfach eine Lösung einreichen können und die vorherige Abgabe damit überschreiben. Bitte reichen Sie bei der Abgabe die erstellte und befüllte Datenbank mit ein. Fügen Sie der Abgabe **nicht** den gesamten JSON-Korpus hinzu. Ihr Abgabe sieht wie folgt aus:

- Teil 1: den Fan-Fiction-Importer mit Ihrer Implementierung.
- Teil 2: Ihre Fan-Fiction-Explorer Webanwendung inkl. Datenbank.

Der Name der Zip-Datei ergibt sich aus dem Präfix `Projekt_WT_SS20`, ihrem Vor- und Nachnamen jeweils getrennt durch `_`.

Beispiel: **Projekt\_WT\_SS20\_Max\_Mustermann.zip**

### Bewertungskriterien

Wesentliches Kriterium zur Bewertung Ihres Projektes ist die Umsetzung aller beschriebenen Funktionen. Ihre Anwendung muss für die definierten Anforderungen einen ernst gemeinten, erkennbaren Lösungsvorschlag beinhalten. Zusätzlich bewerten wir den Aufwand und die Qualität, die bei der Umsetzung der Funktionen erkennbar sind. Dazu gehört die Verwendbarkeit, Fehlertoleranz und ästhetische Qualität der Benutzeroberfläche sowie die Qualität (Struktur) des Datenbankschemas.

*Wir wünschen Ihnen viel Erfolg!*