Andrej Schreiner, David Stefanov, Daniel Wall, Justin Schiller

Dokumentation für das Django-Projekt: Bibliotheksverwaltung

1. Einleitung:

Das vorliegende Dokument dient als umfassende Dokumentation für das Django-Projekt zur Bibliotheksverwaltung. Das Projekt wurde entwickelt, um eine Software zur Verwaltung der Ausleihe von Büchern, Filmen und Spielen zu schaffen, die es Benutzern ermöglicht, eine detaillierte Übersicht über verfügbare und ausgeliehene Medien zu erhalten. Die Software bietet Funktionen wie Katalogisierung, Ausleih-Management, Benutzerprofile, Benutzer-Bewertungen und eine grafische Oberfläche für Neuigkeiten und Filterung.

2. Projektumfeld und Begründung:

Die Idee hinter dem Projekt entstand aus dem Bedarf, einen effizienten Manager von Bibliotheksmedien bereitzustellen. Mit dem ständig wachsenden Bedarf an digitalen Ressourcen war es wichtig, eine Software zu entwickeln, die es Benutzern ermöglicht, schnell und einfach auf Bibliotheksmedien zugreifen zu können. Aufgrund der Katalogisierung wurde dieser Dienst so übersichtlich wie möglich gehalten. Die Bibliotheksverwaltung bietet bequeme Möglichkeiten, ein eigenes Benutzerprofil zu führen, den Bestand zu verwalten, ausgeliehene Medien zu überwachen und Benutzern ein personalisiertes Erlebnis zu bieten.

3. Soll-Zustand:

Mit der Realisierung des Projektes sollen diverse Funktionsweisen gegeben sein. Hierzu sollte das Projekt über eine Katalogisierungsfunktion verfügen, mit welcher die Nutzer Filterkriterien, wie Titel, Autor, Genre oder Sprache anwenden können. Des Weiteren benötigen die Nutzer jeweils Benutzerprofile um Medien ausleihen oder Rückgabedaten verwalten zu können. Um die Benutzerfreundlichkeit zu gewährleisten, muss die Plattform eine intuitive Benutzeroberfläche bieten, mit welcher die neuesten Medien angezeigt und gefiltert werden können. Nützlich wäre zudem die Visualisierung der Mediencover um die Plattform anschaulich zu gestalten. Als letzte unabdingbare Anforderung sollten Medien reservierbar sein, dies simuliert eine reale Bibliothek und ermöglicht Medien vorzubestellen, wenn diese derzeit nicht verfügbar sind.

Zusätzlich können optionale Features umgesetzt werden, welche das Benutzererlebnis verbessern können. Dazu zählt das Bewertungssystem für Medien, welches Benutzern das Recht erteilt, Bewertungen zu vergeben und möglicherweise Rezensionen zu verfassen. Außerdem wäre ein Empfehlungssystem eine willkommene Funktion, da somit der Benutzer auf diverse Medien aufmerksam gemacht werden kann, womit idealerweise die Kundenbindung gestärkt wird. Dies kann auf Basis der Benutzerbewertungen oder der Ausleihhistorie erfolgen, wodurch die Empfehlungen je Nutzer personalisiert werden.

4. Ist-Zustand

Programm Schnittstelle: Die Programmierschnittstelle (API) des Django-Projekts besteht aus einer Reihe von URLs, die verschiedene Funktionen der Bibliotheksverwaltung bereitstellen. Die wichtigsten Endpunkte umfassen:

- /: Startseite des Projekts
- /lib_app/books: Liste aller Bücher
- /lib_app/book/<int:book_id>/: Detailansicht eines bestimmten Buches
- /lib_app/authors: Liste aller Autoren
- /lib_app/author/<int:author_id>/: Detailansicht eines bestimmten Autors
- /lib_app/register/: Registrierungsseite für Benutzer
- /lib_app/login/: Anmeldeseite für Benutzer
- /lib_app/logout/: Abmeldeseite für Benutzer
- Weitere Endpunkte für Benutzerprofile, Ausleih-Management, Reservierungen usw.

5. Genutzte Ressourcen und Einschränkungen:

Das Django-Projekt nutzt d**as** Django-Webframework für die Backend-Entwicklung und HTML/CSS/JavaScript für die Frontend-Entwicklung. Es verwendet auch eine SQLite-Datenbank zur Speicherung von Benutzerinformationen, Medieninformationen und Ausleihhistorie. Einschränkungen können sich aus den Ressourcen des Hosting-Servers, der Datenbankgröße und der Benutzerlast ergeben.

6. Implementierungsphase

1. lib_app/ **Verzeichnis:**
    - Dieses Verzeichnis enthält die Hauptanwendung des Django-Projekts.
    - models.py: Definiert die Datenbankmodelle für Bücher, Autoren und andere relevante Informationen.
    - views.py: Enthält die View-Funktionen, die die Anwendungslogik steuern und die Daten an die Benutzeroberfläche übergeben.
    - urls.py: Definiert die URL-Routen für die verschiedenen Ansichten und Endpunkte der Anwendung.
    - templates/ Verzeichnis: Enthält die HTML-Vorlagen für die Benutzeroberfläche der Anwendung.
2. Bibliothek/ **Verzeichnis:**
    - Dieses Verzeichnis enthält die Einstellungen und Konfigurationen für das gesamte Django-Projekt.
    - settings.py: Enthält die Projektkonfigurationen wie Datenbankkonfigurationen, berechtigte Hosts, statische Dateipfade usw.
3. manage.py**:**

- Das Hauptskript zum Ausführen von Verwaltungsaufgaben und Befehlen für das Django-Projekt.

4. **Weitere wichtige Dateien:**

   - requirements.txt: Enthält eine Liste der Python-Pakete und deren Versionen, die für das Projekt benötigt werden.
   - README.md: Eine Readme-Datei, die eine Einführung in das Projekt und Anweisungen zur Installation und Verwendung enthält.

Zusätzlich wurden Testszenarien verfasst, wodurch die Funktionsweise des Projektes in diversen Testfällen untersucht werden kann und demnach der Fehlerprävention nachgegangen werden kann. Durch die Analyse des Codes in den genannten Dateien können Entwickler ein besseres Verständnis für die Struktur und Funktionalität des Django-Projektes zur Bibliotheksverwaltung gewinnen und bei Bedarf Anpassungen oder Erweiterungen vornehmen.

7. Fazit:

Das Django-Projekt zur Bibliotheksverwaltung bietet eine umfassende Lösung zur Verwaltung von Bibliotheksmedien, die es Benutzern ermöglicht, einfach auf Medien zuzugreifen, diese auszuleihen und zu verwalten. Durch die Integration von Suchfunktionen, Benutzerprofilen, Ausleih-Management und weiteren Funktionen wird eine benutzerfreundliche Plattform geschaffen, die den Anforderungen moderner Bibliotheken gerecht wird. Dennoch bietet dieses Projekt eine weitreichende Skalierbarkeit, somit kann die Bibliotheksverwaltung zukünftig um diverse Funktionen erweitert werden.