**Aufbau:**

-Da die Funktionen und Struktur für das Hinzufügen eines neuen Buches und des Editierens sehr ähnlich sind und der Seiteninhalt von übersichtlicher Länge ist, habe ich beide Funktionen in eine einzelne Komponente gepackt, um redundante Änderungen klein zu halten. Bei potenziell wachsenden Projekten wäre es sicherlich sinnvoll gewesen, das Ganze in Subkomponenten Aufzuteilen. Hier habe ich es der Einfachheit halber so gelassen und den jeweiligen Modus über einen Router-Metaparameter abgefragt.

-Der Aufbau der URLs ist sicher Geschmackssache. Ich persönlich hielt es für nicht unsinnvoll für jedes Buch, inklusive Duplikate eine eigene URL zu haben und mir außerdem dadurch direkt die BuchID mitzugeben. Dies hat den Vorteil, dass vorhandene Bücher bei manueller Navigation über die URL automatisch schon korrekt angezeigt werden, auch wenn dies ein seltener Anwendungsfall ist und im aktuellen Projekt die manuelle Navigation auch nicht komplett abgesichert ist.

-Als Filterfunktion habe ich eine Suchleiste hinzugefügt, die während der Eingabe die gefilterte Buchliste aktualisiert. Ich habe dies simplerweise als Suche nach Substrings in Autor und Titel der Bücher implementiert.

**Daten:**

-Für das Speichern der Buchdaten habe ich den localStorage verwendet. Das entpuppte sich als nicht ganz optimal, da die Buchdaten dort nicht als Array gespeichert werden konnten. Für größere Datenmengen wäre es sicher sinnvoller gewesen, die einzelnen Bücher mit eigenem Key abzuspeichern. In diesem Fall wäre das sicherlich ohne Probleme möglich gewesen. Ich habe mich dafür entschieden, alle Bücher serialisiert abzuspeichern. Bei jeder Operation, die auf den Buchdaten durchgeführt wird, die Daten wieder zu serialisieren ist ein unsinniger Aufwand von einer Optimierungswarte aus, aber den localStorage bei Löschen und Hinzufügen selbst zu managen und den Storage mit einzelnen Keys für die Bücher zu füllen, wenn potentiell noch andere Objekte darin gespeichert werden können, die nichts mit den Buchdaten zu tun haben fühlte sich für mich sehr unsauber an. Letzten Endes macht es bei den Datenmengen, die in diesem Projekt realistisch per Hand eingetragen werden keinen großen Unterschied und bei einem echten Anwendungsfall hätte ich mich für eine gänzlich andere Art der Datenspeicherung entschieden, daher habe ich hier die simpelste Variante benutzt.