**Books Sharing Application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Books Sharing Application |
| Author | Pooja Shelar |
| Version | V1.0 |
| Creation Date | 09/17/2020 |
| Last Updated | 09/17/2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc45989188)

[**1.** **Introduction** 3](#_Toc45989189)

[**2.** **Architecture Diagram** 3](#_Toc45989190)

[**3.** **Use-Case View** 4](#_Toc45989191)

[**4.** **Flow Chart** 5](#_Toc45989192)

[**5.** **Components** 6](#_Toc45989193)

[**6.** **Services** 6](#_Toc45989194)

[**7.** **Validations** 6](#_Toc45989195)

[**8.** **Tools and Libraries** 7](#_Toc45989196)

[**9.** **Unit Testing** 7](#_Toc45989197)

[**10.** **Responsive** 8](#_Toc45989198)

[**11.** **Accessibility** 8](#_Toc45989199)

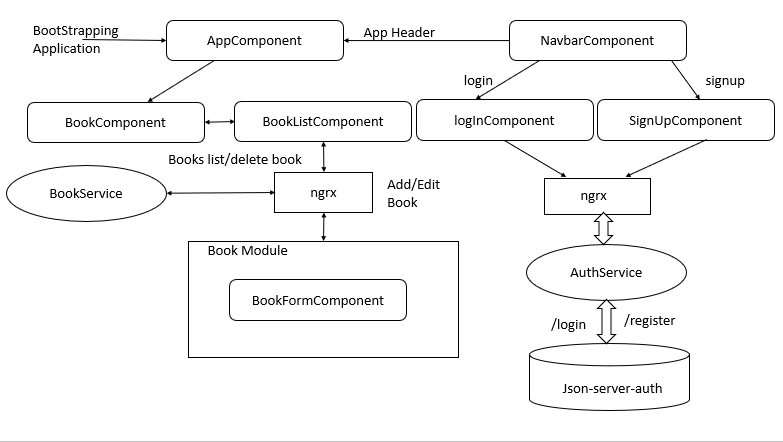
[**12.** **List other enhancement (at least top 3)** 9](#_Toc45989200)

[**13.** **Setup** 9](#_Toc45989201)

# **Introduction**

This document provides a comprehensive architectural overview of the application, using number of different architectural views to depict different aspects of the application.

# **Architecture Diagram**



# 

# **3. Use-Case View**

The application use cases are –

* **Login User** – It provide functionality to logging in user also having email and password validation with it.
* **Log out** – It provide functionality to Logout user
* **Signing up**  – It provide functionality to register new user also having email and password validation with it.
* **Book list** – It displays list of books
* **Add book** –After adding book, book entry added inside a list
* **Edit book** – Each book in the list will have edit button to edit the book.
* **Delete List** – Each book in the list will have a delete icon, the book can be deleted by clicking on it, it also confirms about the deletion of book, when user confirm by ok the book will be deleted

These use cases are initiated by the user.

**Features -**

|  |  |
| --- | --- |
| **Features Name** | **Uses/ Provide in Application** |
| NgRx | For Login, Sign up, CRUD |
| Jest for testing | For Testing Application |
| Accessibility | Use For all Application |
| Web components | Use for Component |
| Storybook | Implemented for every Component |
| Dynamic component loading | Use for Books Module |
| Json-server | Used for Application |
| Lighthouse | Show Application Perfoemance |

# **Components**

* AppComponent – This component is used for Application Bootstrapping
* BookComponent– Home Page of application that show book list.
* BookListComponent –This Component shows the list and edit and delete feature and button for adding new book .
* BookFormComponent – This component contain form for add and edit book.
* NavbarComponent – This component contains header information like Title, Login/Logout and Signup, change theme buttons.
* LogInComponent– This component contains form for user login.
* SignUpComponent– This component contains form for user registration.

# **Services**

There are two services –

1. BookService – This service generally having logic about list in the application and this contain code for getting only book by Id and edit book details, add new book and delete book logic.
2. AuthService – This service generally having logic about user logIn and Signup api.

# **Validations**

1) Application covered the required validation for User logIn, SignUp

**Validation strategy:**

- Reactive form for validation

- There is corresponding html template for form elements

- Required attribute true for individual form element, this can be extended to have in a separate service file for re-usability.

# **Tools and Libraries**

- Angular version: 10.0.6

- Node: 14.7.0

- TypesScript: 3.9

- ngrx/store: 10

- Rxjs: 6.5.5

- jest: 26.4.2

- tslint: 6.1.0

- typescript: 3.9.5

-concurrently – 5.3

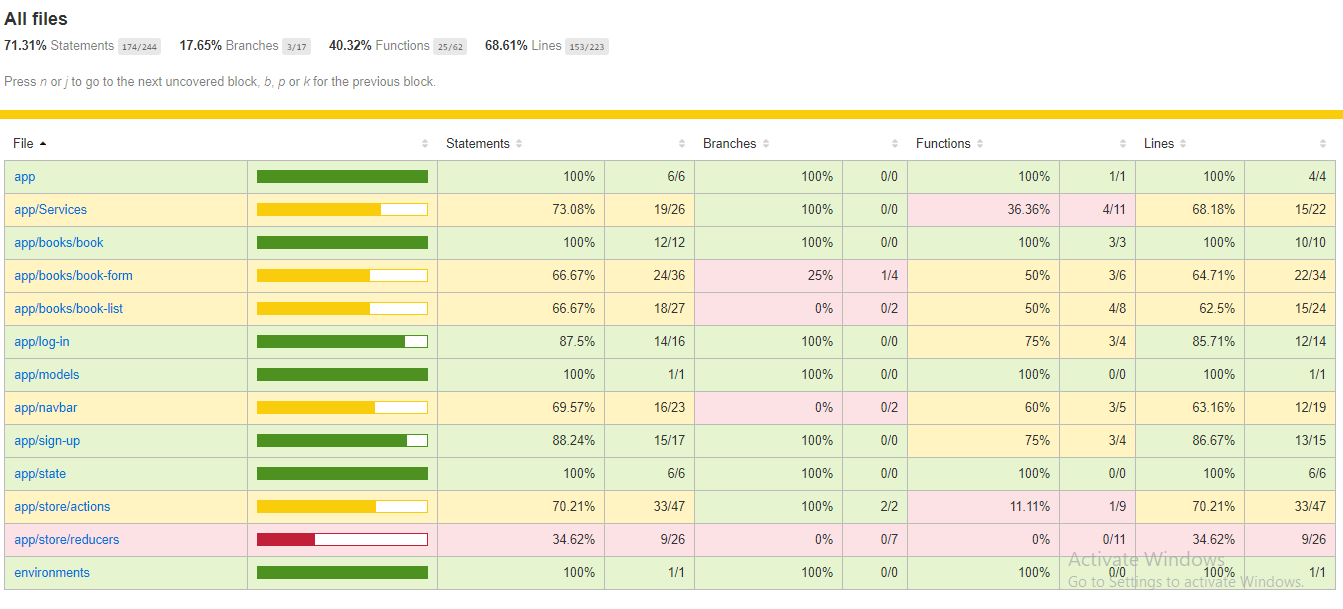
-json-server-auth- 2.0.2

# **Unit Testing**

Added Unit Test cases for services and components.

* + auth.service.ts – It contain the all logic like login and signup user
  + Each Component in Books sharing Application Consist their own test cases.

**Screenshot for unit test cases –**



# **Responsive**

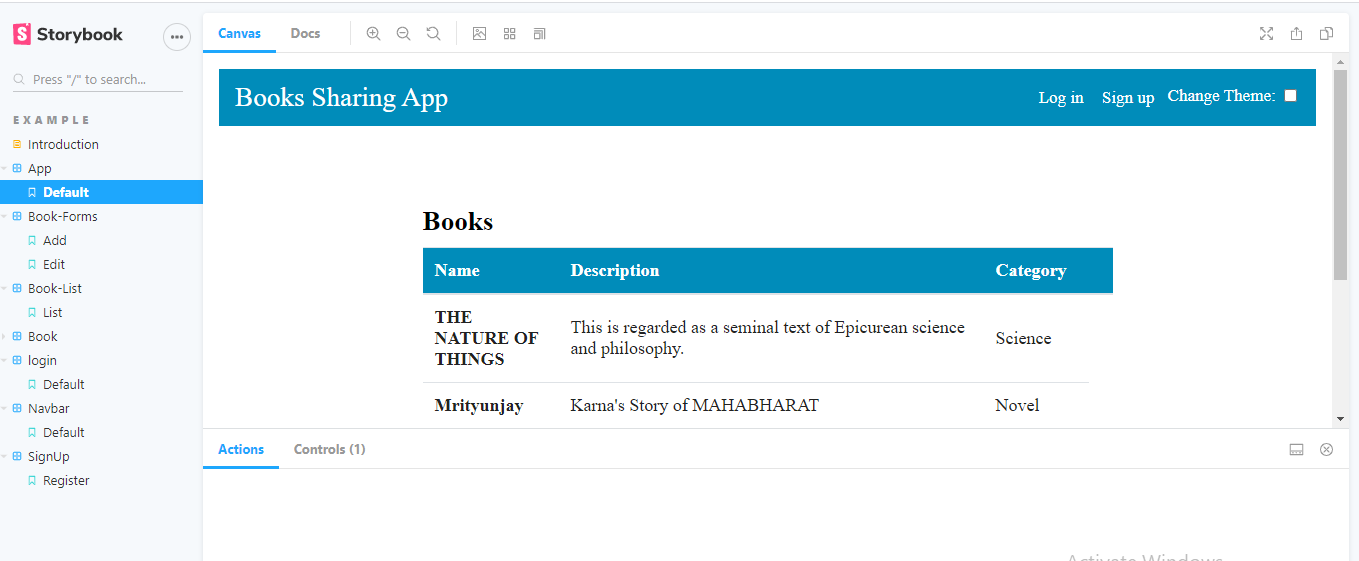
* The application is responsive with all the device and desktop size.
* No Horizontal scrollbar in application.

# **Accessibility**

For Accessibility, application having appropriate HTML tags that are preferred for accessibility. User can run application via keyboard and tab.

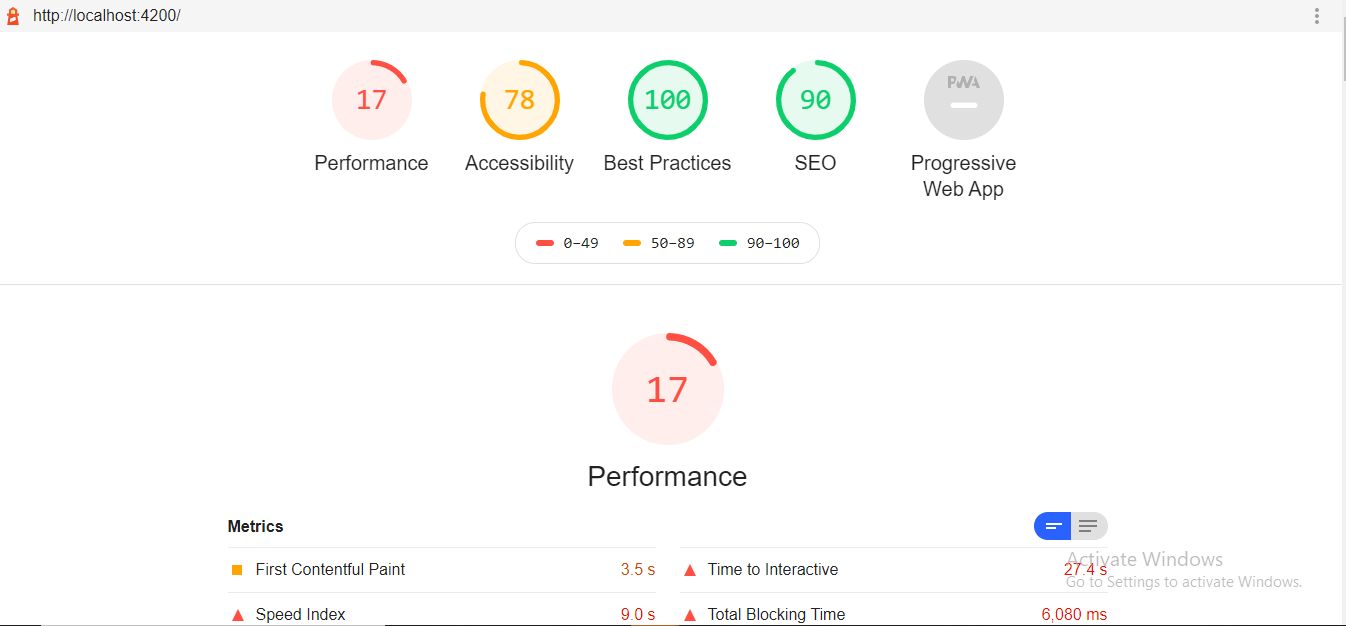
# **Storybook**

I have implemented basic storybook structure for all the components.



# **Lighthouse Performance –**

Below is the lighthouse performance of Books Sharing application.



# **Setup**

1. Download and install node.js (version: 12.16)
2. Install npm and install @angular (version: 9.0)
3. Git clone https://github.com/poojashelar/Tavisca-Assignment.git origin master
4. Navigate to cloned folder
5. And Run Command npm run start
6. For Jest testing use Below Command: -

npm run test-watch - Show the all test cases Passed and Fail.

**Story Book Set up** –

Run Command – npm run storybook