**Offers Management Application**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Architecture Design Document |
| Application | Offers Management Application |
| Author | Sneha Waje |
| Version | V1.0 |
| Creation Date | 07/09/2020 |
| Last Updated | 11/09/2020 |
| Status | Completed |

Contents

[Document Properties 1](#_Toc50748265)

[**1.** **Introduction** 3](#_Toc50748266)

[**2.** **Use-Case View** 3](#_Toc50748267)

[**3.** **Flow Chart** 4](#_Toc50748268)

[**4.** **Components** 5](#_Toc50748269)

[**5.** **Services** 5](#_Toc50748270)

[**6.** **Validations** 5](#_Toc50748271)

[**7.** **NgRX Store** 6](#_Toc50748272)

[**8.** **Tools and Libraries** 6](#_Toc50748273)

[**9.** **Unit Testing** 6](#_Toc50748274)

[**10.** **Responsive** 7](#_Toc50748275)

[**11.** **Accessibility** 7](#_Toc50748276)

[**12.** **Storybook** 7](#_Toc50748277)

[**13.** **Lighthouse** 8](#_Toc50748278)

[**14.** **Setup** 8](#_Toc50748279)

# **Introduction**

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

# **Use-Case View**

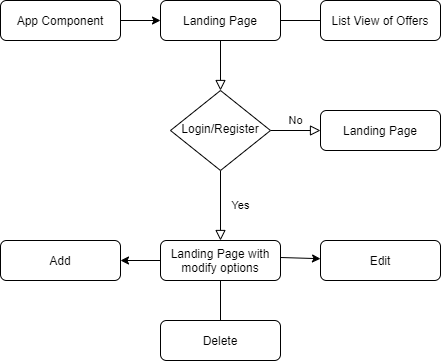
The application use cases are –

* **Login** – Allows user to login with credentials with provided form validations.
* **Register** – Allows user to register and login with credentials with provided form validations.
* **Show List of Offers** – Show list of all offers
* **Add Offer** – It provide functionality to add offer.
* **Edit Offer –** Any offer can be editable by clicking edit button provided in front of it by accessing ID of particular offer.
* **Delete Offer**– Any offer can be deleted by clicking delete button provided in front of it by accessing ID of particular offer.

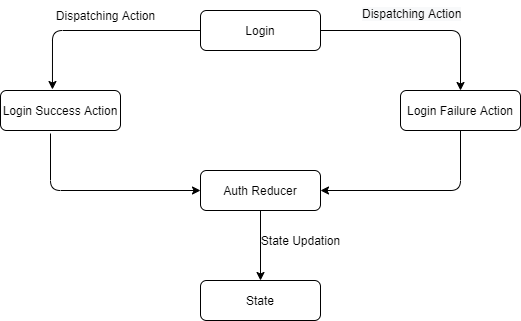
These use cases are initiated by the user.

# **Flow Chart**

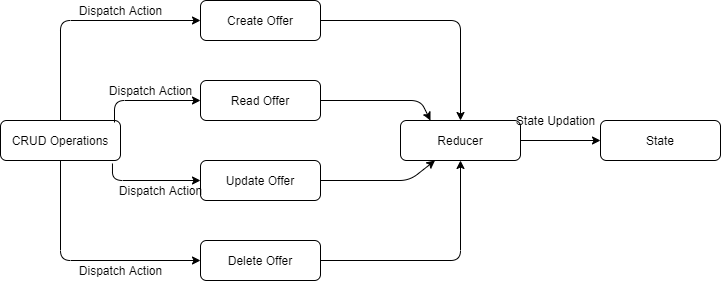
* **Basic Flow**



* **Login NgRX**



* **CRUD NgRX:**



# **Components**

* LandingComponent – This component is used for basic view of application with Login, Register, and list view option.
* LoginComponent – Login component allows a user to login and register with username and password and give access to modify list of offers.
* AddOfferComponent – This component contain form to add new offer to the list and edit offer; post to that updated list is shown.

# **Services**

There are two services –

1. AuthService – This service generally having logic about login and registration of a user.
2. OfferService – This service generally having logic about CRUD operations i.e. Create, Read, Update and Delete.

# **Validations**

1) Application covered the required validation for Username.

**Validation strategy:**

- Reactive form for validation

- There is corresponding html template for form elements

# **NgRX Store**

The application uses ngRx for state management for user and offers data.

# **Tools and Libraries**

- Angular version: 10.0

- TypesScript: 4.0.2

- Rxjs: 6.5.5

- Jest: 26.4.2 (This is a unit testing framework; it consists assertion and mocking support for writing test cases.)

- NgRX Store: 10.0.0

-tslint: 6.1.0

# **Unit Testing**

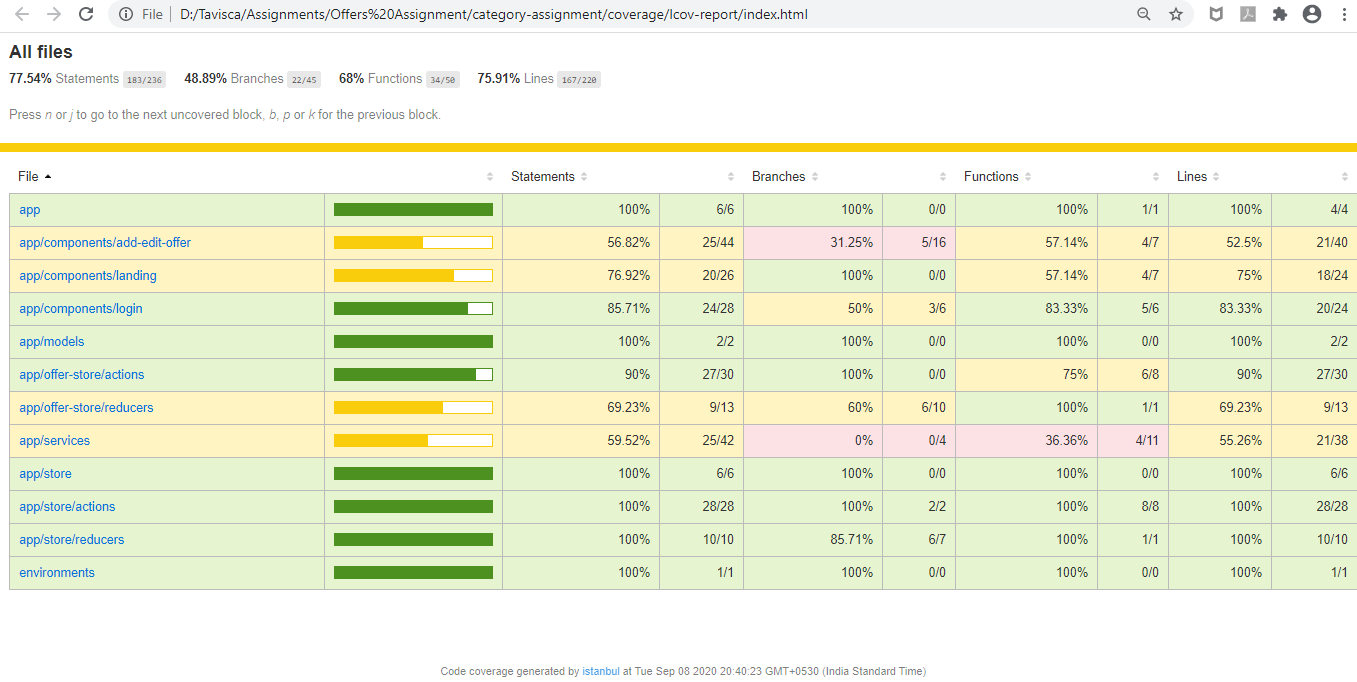
Added Unit Test cases for services.

It consists of two services –

* auth.service.spec.ts - It contain test cases for the all the logic like login and register user.
* offer.service.spec.ts - It contain test cases for the all the logic like add offer, update offer, delete offer and get list of offers.
* offer.action.spec.ts - It contain test cases for the all the logic like actions performed in offer components functionalities.
* offer.reducers.spec.ts - It contain test cases for the all the logic like actions performed in offer reducer functionalities.
* auth.action.spec.ts - It contain test cases for the all the logic like actions performed for login/register components functionalities.
* auth.reducers.spec.ts - It contain test cases for the all the logic like actions performed for login/register component functionalities.

**Total test cases: 35**

**Screenshot for unit test cases -**



# **Responsive**

* The application is responsive with all the device and desktop size.
* No Horizontal scrollbar in application.
* Used CSS grid for templating.

# **Accessibility**

For Accessibility, application having appropriate HTML tags that are preferred for accessibility

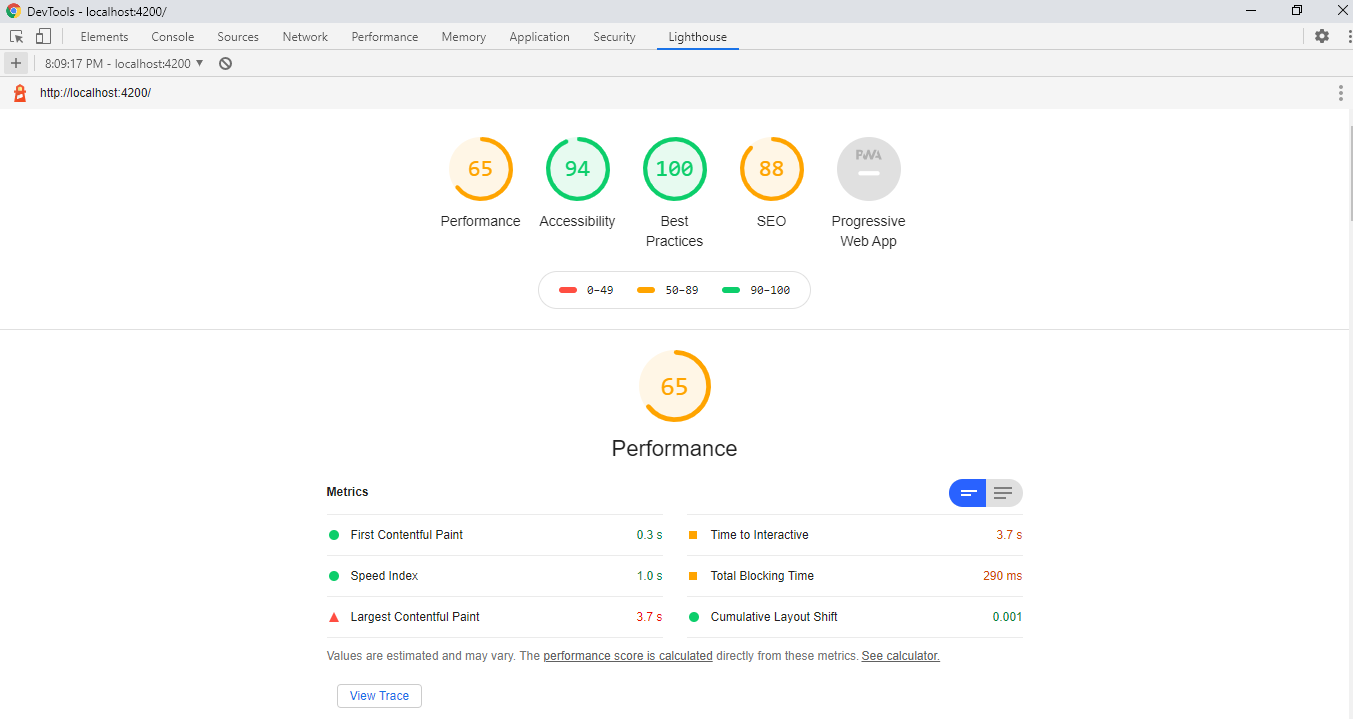
User can run application via keyboard and tab.

# **Storybook**

* I have installed and added basic setup for Srorybook.
* All components rendering fine with default views
* App Component works fine with fake json server

# **Lighthouse**

Desktop View



# **Setup**

1. Download and install node.js (version: 12.18.3)
2. Install npm(version:6.14.8) and install @angular (version: 10.0)
3. Git clone <https://github.com/snehawaje-1524/Tavisca-Assignment> origin master
4. Navigate to cloned folder & run npm install
5. To run API(JSON Server): npm run json:server
6. To run application: ng serve
7. For testing : npm run test-watch
8. For Storybook: npm run storybook