**Links Sharing Application**

Document Properties

|  |  |
| --- | --- |
| Item | Details |
| Document | Architecture Design Document |
| Application | Links Sharing Application |
| Author | Mangesh Joshi |
| Version | V1.0 |
| Creation Date | 20/11/2020 |
| Last Updated | 20/11/2020 |
| Status | Completed |

Contents

Document Properties ………………………………………………………………………………………………………………. 1

[1. Introduction 3](#_Toc56807480)

[2. Architecture Diagram 3](#_Toc56807481)

[3. Use Case View 4](#_Toc56807482)

[4. Components 4](#_Toc56807483)

[5. Services 5](#_Toc56807484)

[6. Validations 5](#_Toc56807485)

[7. Tools and Libraries 5](#_Toc56807486)

[8. Unit Testing 6](#_Toc56807487)

[9. Responsive 7](#_Toc56807488)

[10. Accessibility 7](#_Toc56807489)

[11. Lighthouse 7](#_Toc56807490)

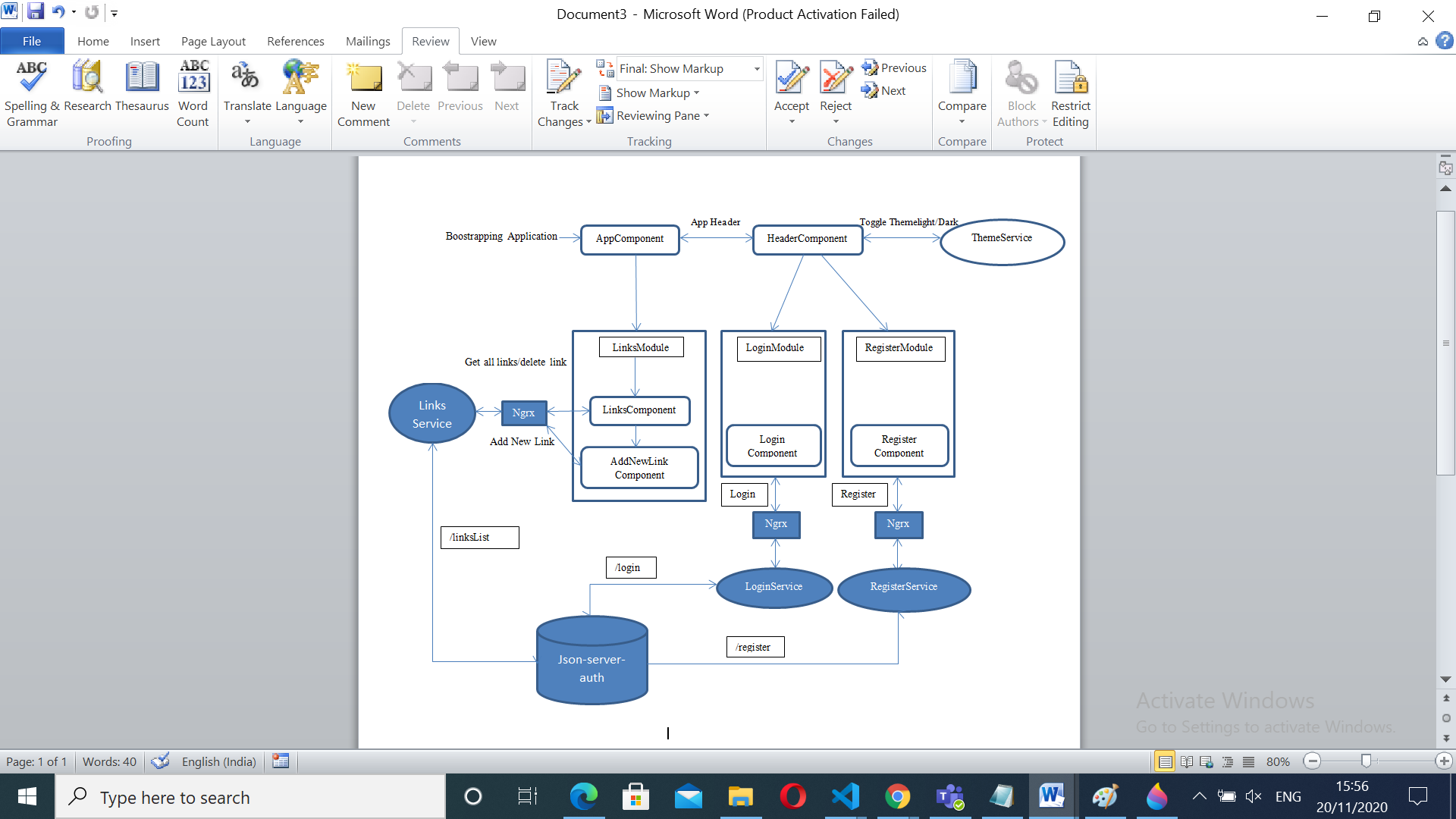
[12. Story Link 8](#_Toc56807491)

[13. Setup 8](#_Toc56807492)

# Introduction

This document provides a comprehensive architectural overview of the Links Sharing Application, using different architectural views to depict different aspects and functionalities of the application.

# Architecture Diagram



# Use Case View

The application use cases are -

* **Login –** It provides functionality to Login to Links Sharing Application.
* **Register –** It provides functionality to Register to Links Sharing Application.
* **Display Links List** **–** Displays the List of links.
* **Add Link –** It provides functionality to add new Link details. Also ‘Add’ button is disabled until the required fields are valid.
* **Edit Link** **–** It provides functionality to edit/update the existing Link details.
* **Delete Link** **–** Each Link listing will have a ‘Delete’ button, the respective Link can be deleted by clicking on it
* **Theme** – Can toggle between Light/Dark theme.

# Components

* **AppComponent** – This component is used for Application Bootstrapping
* **Header Component** – This component contains the application header, Login/ Register/ Theming buttons
* **LinksComponent** – Displays Links list and contains buttons to add/edit/delete a Link.
* **AddNewLinkComponent** – This component contains form and logic to add/edit single Link.
* **LoginComponent** – This component contains form and logic for Login functionality.
* **RegisterComponent** – This component contains form and logic for Register functionality.

Login/Register and CRUD operations on Link uses ngrx for state management.

# Services

* **AuthGuardService** – Implements CanActivate for routing user to dashboard page if not logged in.
* **LinkService** **–** This service has the logic on getting links list, add new Link details, edit/update Link details and delete entire link details. All these functionalities call their respective json-server API’s.
* **LoginService** – This service has logic for calling json-server-auth Login API.
* **RegisterService** – This service has logic for calling json-server-auth Login API.
* **ThemeService** – This service has logic to toggle between Light/Dark theme.

# Validations

* Application has covered the required validation for Login/Register forms.
* Application shows proper error message for Login/Register failure i.e. incorrect email/password, duplicate email registration, short password etc.

**Validation strategy:**

* Reactive form for validation in Login/Register
* Display error message received from service

# Tools and Libraries

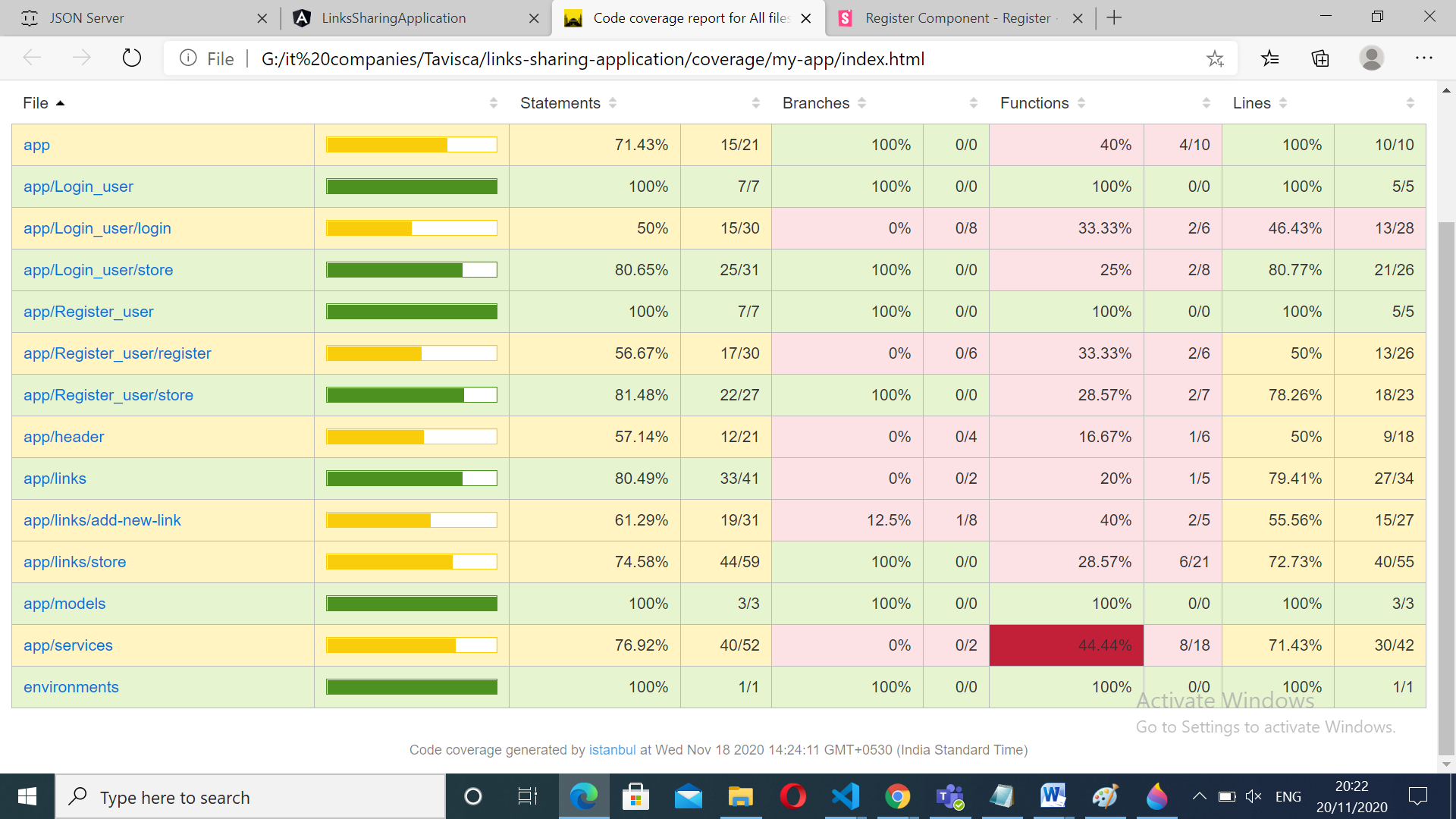
* Angular version – 11.0.0
* TypeScript - 4.0.2
* rxjs - 6.6.0
* ngrx/store – 10.0.1
* ngrx/entity – 10.0.1
* ngrx/effects – 10.0.1
* Jest – 26.6.3
* json-server-auth – 2.0.2
* @storyLink – 6.0.21

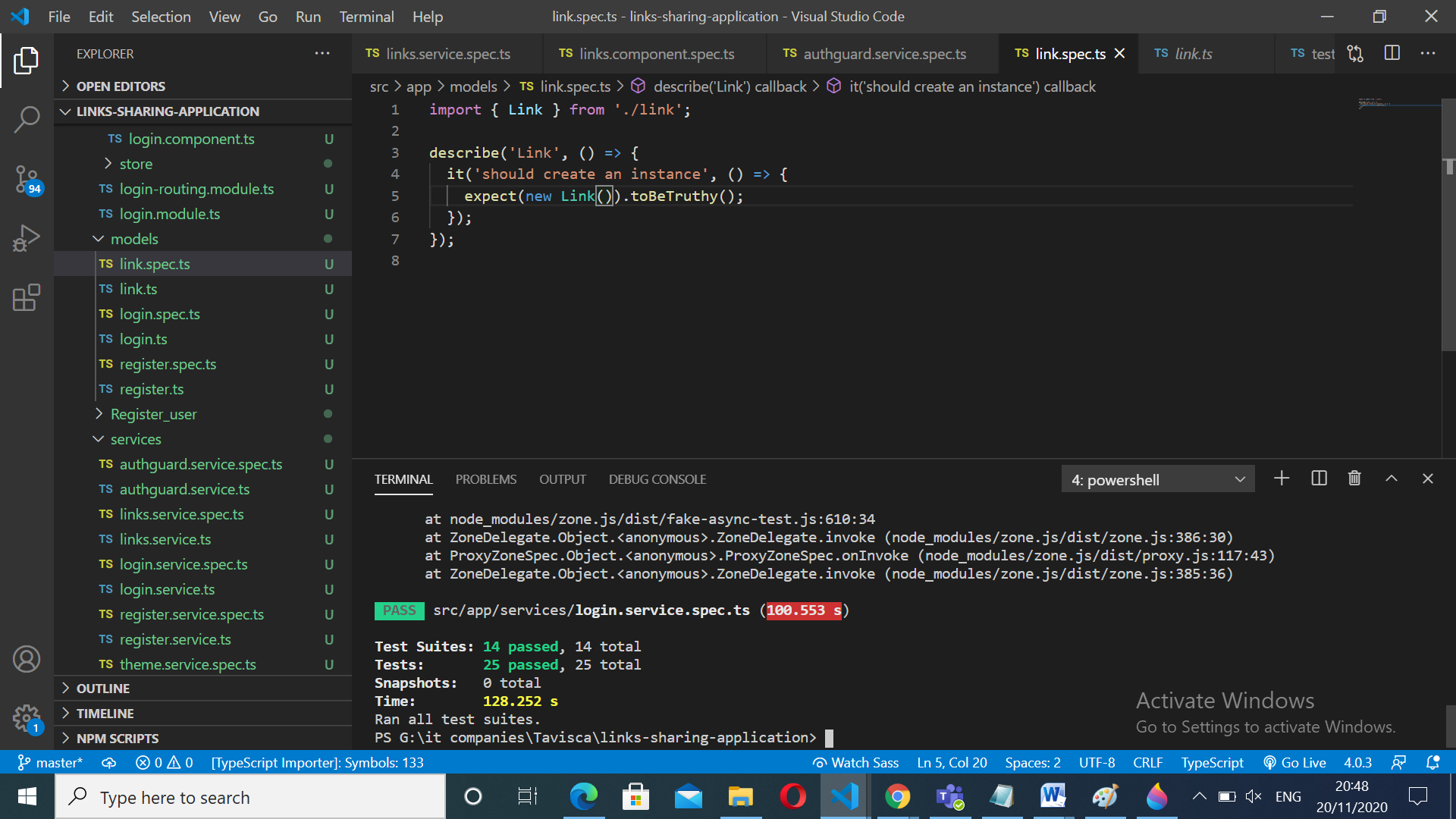
# Unit Testing

* Test cases for services: It includes test cases for the json-server API calls
* No of test cases: 10
* Component related test cases – 12
* Class related test cases – 3

**Total test cases: 25**

**Screenshot of unit test cases –**





# Responsive

* The application is responsive with all the device and desktop size.
* Used custom CSS for styling.

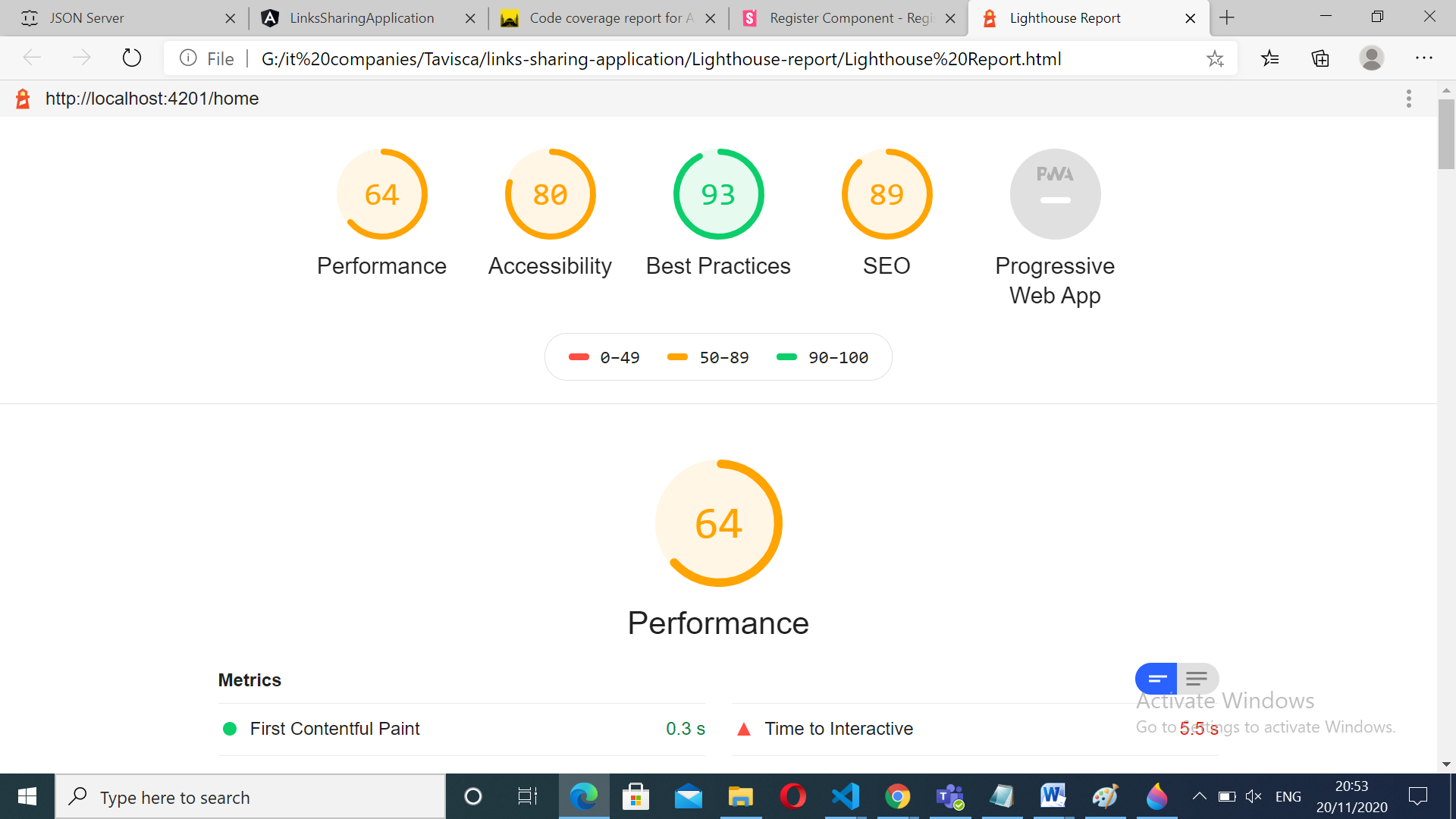
# Accessibility

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

User can run application via tab and enter.

# Lighthouse

Below is the Lighthouse report screenshot

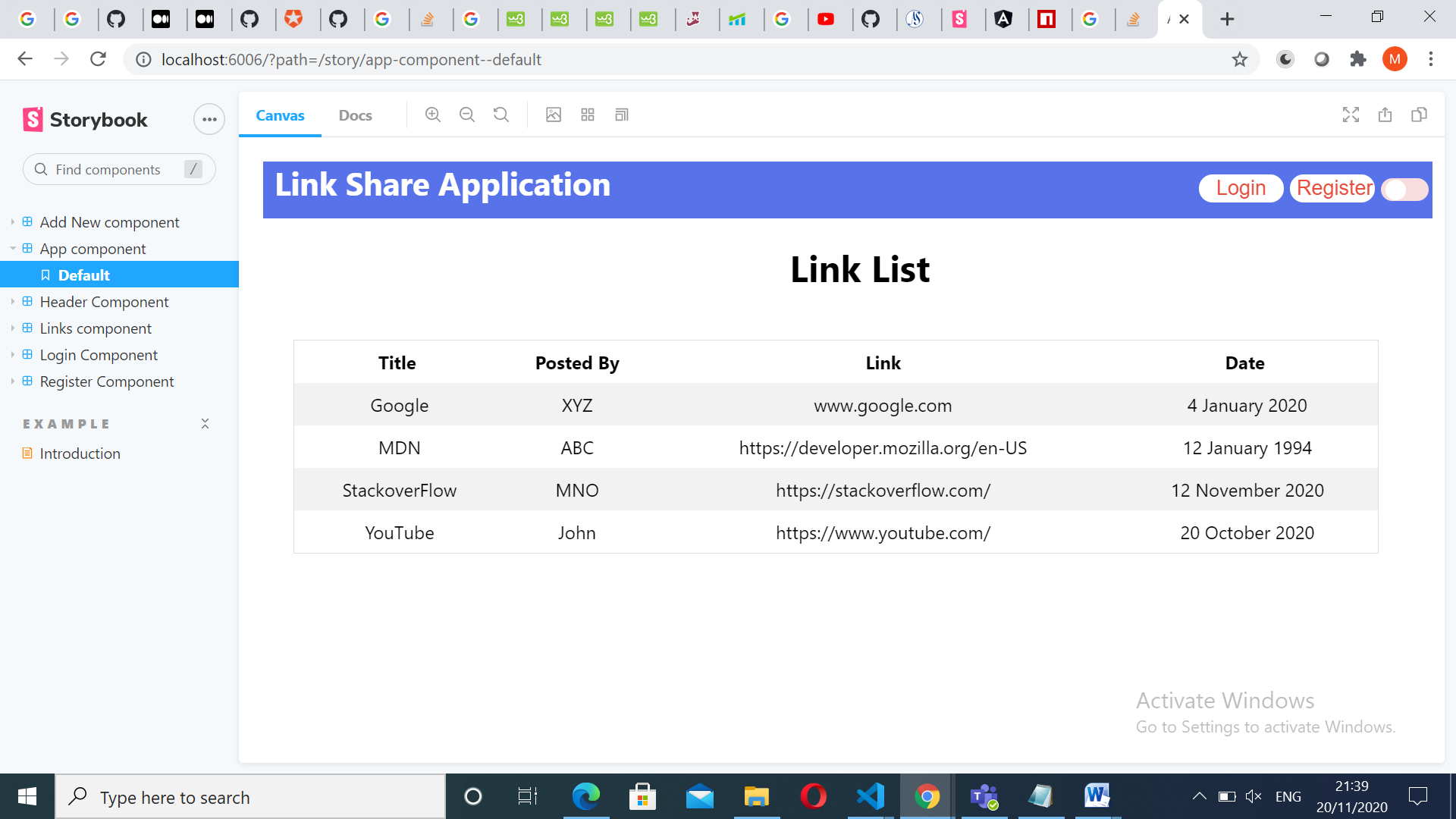


# Story Link

StoryLinks are written for below components:

1. AppComponent
2. HeaderComponent
3. LinksComponent
4. LoginComponent
5. RegisterComponent
6. AddNewLinkComponent

Below is the sample screenshot:



# Setup

1. Download and install node.js (version: 13.8.0)
2. Install npm and install @angular (version: 10.1.0)
3. Git clone [https://github.com/mangesh-joshi/Links-Sharing-Application-Tavisca](https://github.com/mangesh-joshi/Links-Sharing-Application-Tavisca%20)
4. Navigate to cloned folder & run **npm install**
5. To start application run **ng serve –open**
6. To start the json-server run **npm install -D json-server json-server-auth & then**

**json-server-auth --watch db.json.**

1. For testing use **npx jest**
2. For running storyLink, run **npm** **run** **storybook**