

SELL YOUR FURNITURE APPLICATION

A PROJECT REPORT

SUBMITTED BY

NITIN Garg
(1628144)

**IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE
OF**



**CGC-COE COLLEGE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING.**

MAY-2020

ACKNOWLEDGEMENT

The internship opportunity I had with MINDTREE LTD. was a great chance for learning and professional growth. Therefore, I consider myself as a very fortuitous individual to be a part of such reputed organization. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this internship period.

I am also thankful to all the staff members of our department, who helped us directly or indirectly in our endeavour and shown keen interest in our project by providing their encouragement.

I would like to thank Mr. Amarendra Gopalakrishna , Mr. Deepak Nagaraj and the management of Mindtree ltd for their opportunity and enlightening us with other aspects of project.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives.

I want to give my thanks to my **COMPUTER SCIENCE DEPARTMENT** of CGC College of engineering for always supporting me. The knowledge which they have imparted in me allowed me to perform the actions in a smooth and efficient manner. Along with that their contribution towards self-development skills helped me a lot.

CERTIFICATE

This is to certify that the declaration statement made by the student is correct to the best of my knowledge and belief. He has completed his internship under my guidance and supervision. The present work is the result of his original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The internship is fit for the submission and partial fulfilment of the conditions for the award of B.Tech degree in Computer Science and Engineering from CGC College Of Engineering, Punjab.

PREFACE

A student gets theoretical knowledge from classroom and gets practical knowledge from industrial training. When these two aspects of theoretical knowledge and practical experience are together then a student is fully equipped to secure his best.

In conducting the project study in an industry, students get exposed and have knowledge of real situation in the work field and gains experience from them. The objective of the Industrial Training is to provide an opportunity to experience the practical aspect of the technology in an organization. It provides a chance to get the feel of the organization and its function.

CONTENTS

Title	Page No.
I. Introduction	
I.1. Mindtree Private Limited	6-8
II. Project	8
III. Technology	9-26
III.1. Overview	9
III.2. Snapshots and code	10-26
III.3 Unit testing	26-27
IV. Conclusion	27
V. Bibliography	28

I. INTRODUCTION

I.1. Mindtree Private Limited:

Mindtree is a global technology services company with a mission to engineer meaningful technology solutions to help businesses and societies flourish. Our culture is about collaboration, education and opportunity, and we are proud to be recognized as a great place to work. Mindtree is acquired on the 7th position in the list of top IT companies in India.



Mindtree deals in e-commerce, mobile applications, cloud computing, digital transformation, data analytics, enterprise application integration and enterprise resource planning, with more than 339 active clients and 43 offices in over 18 countries.

The works in Application Development and Maintenance, Data Analytics, Digital Services, Enterprise Application Integration and Business Process Management, Engineering R&D, Enterprise Application Services, Testing, and Infrastructure Management Services.

We Deliver Value to our Clients in Many Ways:

- Our experienced consultants understand current business challenges and how they manifest in a rapidly changing environment
- We put our clients' interests first and do not hesitate to offer candid opinions about solution designs or implementations
- Our proprietary process accelerators are born out of our diverse experience in implementing and integrating enterprise systems; these tools help compress the time required for our clients to realize tangible benefits
- We serve our smaller clients with the same dedication as our multi-billion dollar clients

- We are flexible about how we engage- we do not lock our clients into expensive, multi-year contracts

How We Are Different (and our clients agree!):

- Certainty of Outcome
- Better Anticipation, Better Implementation
- Trusted Advisors
- Cost-Effectiveness & Flexibility
- Knowledge Assets

Website: <https://www.mindtree.com>

Industries: Information Technology and Services

Company size: 20000-25000 employees

Headquarters: Bengaluru, India

Type: Privately Held

Founded: 1999

Revenue: US\$980 million

II.

PROJECT

Project named as SELL YOUR FURNITURE is a web based project. Through this website user can sell or buy furniture. For the server side coding I have used spring boot and for the client side I have used angular 8. For storing the data in the database I have used MySQL.

In this application user can post the furniture details which they want to sell and the buyer can buy that furniture by choosing payment through two modes. These two modes are payment through card and payment on delivery. In this application we have authenticate users because I have enable login and signup through Facebook and Google. It will help the buyer who are looking for varieties by sitting at home and will also help the seller to sell the furniture without any hesitation.

Spring Boot- Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.

Angular 8- It is used to create a single page application .In this I have also used the HTML, CSS and bootstrap. It works on Typescript language.

Bootstrap-I have used it to make my application attractive.

MySQL- I have used it as a local database for storing the data in the local database.

III.

TECHNOLOGY

III.1. OVERVIEW

With the help of Spring Boot framework of java I have completed the backend (server side) of my project and created some API's which has to hit by the user Interface.

With the help of angular 8 using html,css,typescript and bootstrap I have completed the frontend(client side) part of my project and have made it responsive using some media query. To make my website attractive I have also used some courosels and the sliders using bootstrap. To store the data in the local database I have used MySQL.

It is website named as SELL YOUR FURNITURE through which its user can sell or buy any kind of furniture. For selling or buying furniture user should signup first in our website. User can also signup through [Facebook](#) or [Google](#)

Here I have used the social login user for login through Facebook and google. Reactive forms from **reactiveformsmodule** is also used for the validations of the various fields.

III.2 SNAPSHOTS AND CODE

Some of the snapshot of the websites are as follows:

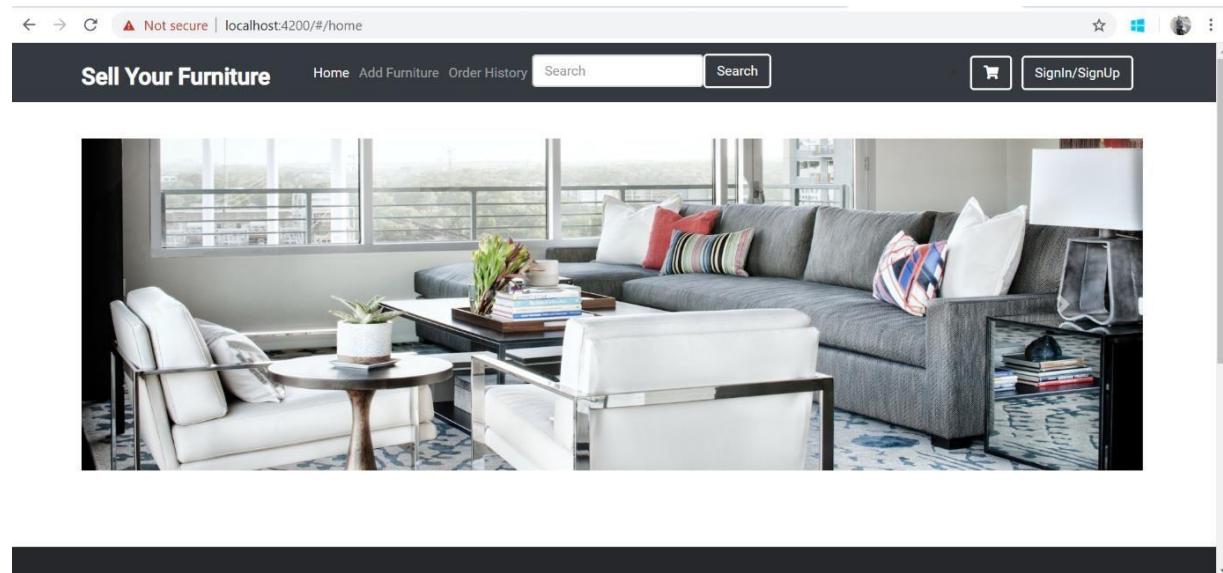


Fig:3.1

This is the home view of the application user should not be logged in for accessing the home page of the application but for the further selection user should be logged in.

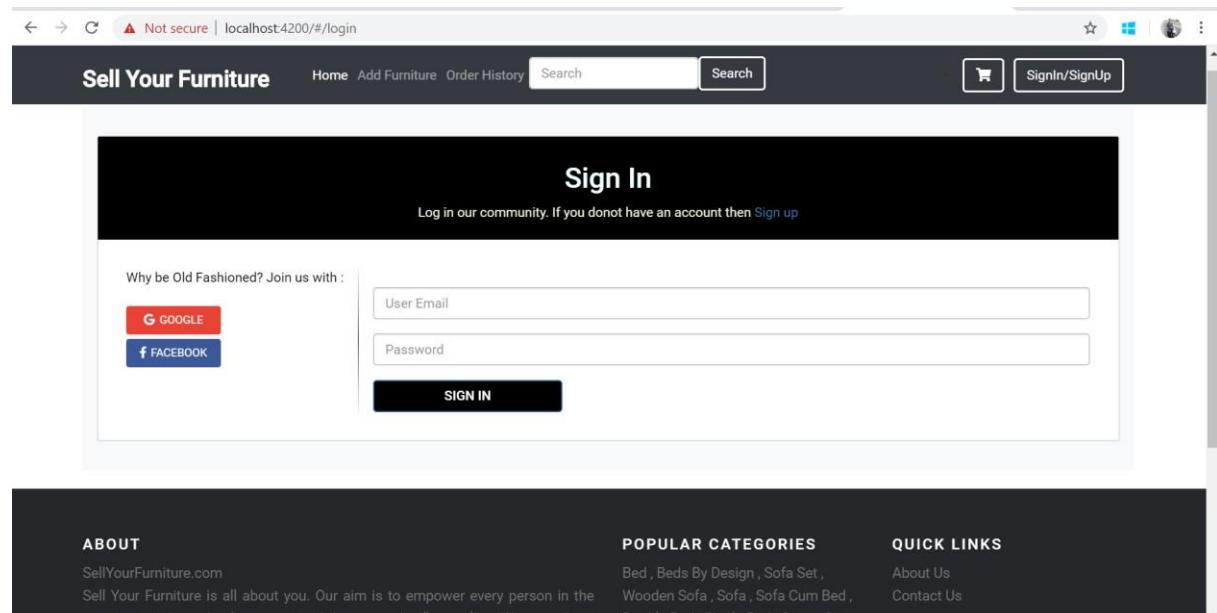


Fig: 3.2

This is the view of the login page where user should enter their credentials or can login through Facebook or Google.

The screenshot shows a web browser window with the URL `localhost:4200/#/signup`. The page has a dark header bar with the text "Sell Your Furniture". Below the header is a main content area with a black header containing the text "Create a new account". A sub-header below it says "Come join our community. If you already have an account? Sign in here.". To the left, there is a sidebar with the text "Why be Old Fashioned? Join us with" followed by two social media buttons: "G GOOGLE" and "f FACEBOOK". The main form area contains five input fields: "Enter first name", "Enter last name", "Enter email", "Enter password", and "Enter phone number". At the bottom of the form is a large orange "Submit" button.

Fig: 3.3

This is the view of the signup page. If the user is not signed up before so he/she have to sign up before login. User can sign up through google or Facebook as well. Here every field is validated.

The screenshot shows a web browser window with the URL `localhost:4200/#/a`. The page has a dark header bar with the text "Sell Your Furniture". Below the header is a main content area with a white form titled "ADDRESS DETAILS". The form contains six input fields: "Street", "City", "District", "State", "Pincode", and "Country". At the bottom of the form is a large black "SUBMIT" button.

Fig: 3.4

This is address page where user should add the address for the delivery of the product.

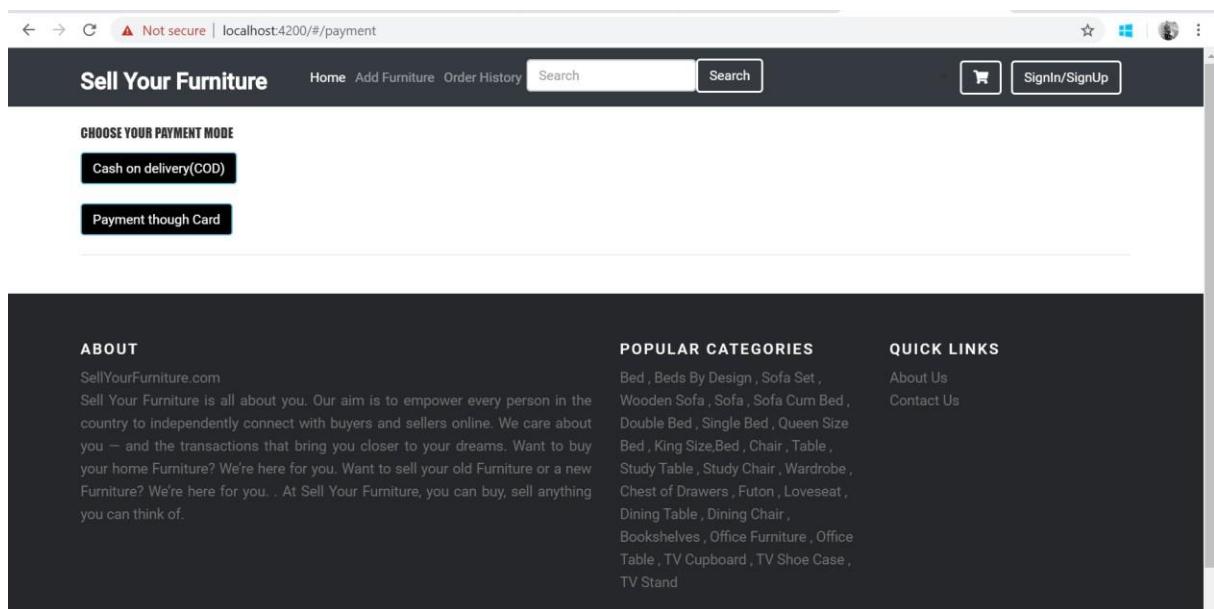


Fig: 3.5

This is the payment page where user can choose payment modes.

And some of the database images as are as follow:-

The screenshot shows a MySQL Workbench interface with a result grid titled 'Result Grid'. The grid displays a single row of data from a table. The columns are labeled: user_id, user_email, user_first_name, user_last_name, user_password, and user_phone_number. The data in the grid is as follows:

	user_id	user_email	user_first_name	user_last_name	user_password	user_phone_number
1	NULL	domain@o.com	prakash	singh	Pr@k@sh12	7888803530
			NULL	NULL	NULL	NULL

On the right side of the interface, there is a vertical toolbar with icons for 'Result Grid' (selected), 'Form Editor', and other tools.

Fig: 3.6

This is the image of the user table which is present in the local database.

The screenshot shows a MySQL Workbench interface with a result grid titled 'Result Grid'. The grid displays 15 rows of data from a table. The columns are labeled: address_id, address_city, address_country, address_district, address_pincode, address_state, and address_street. The data in the grid is as follows:

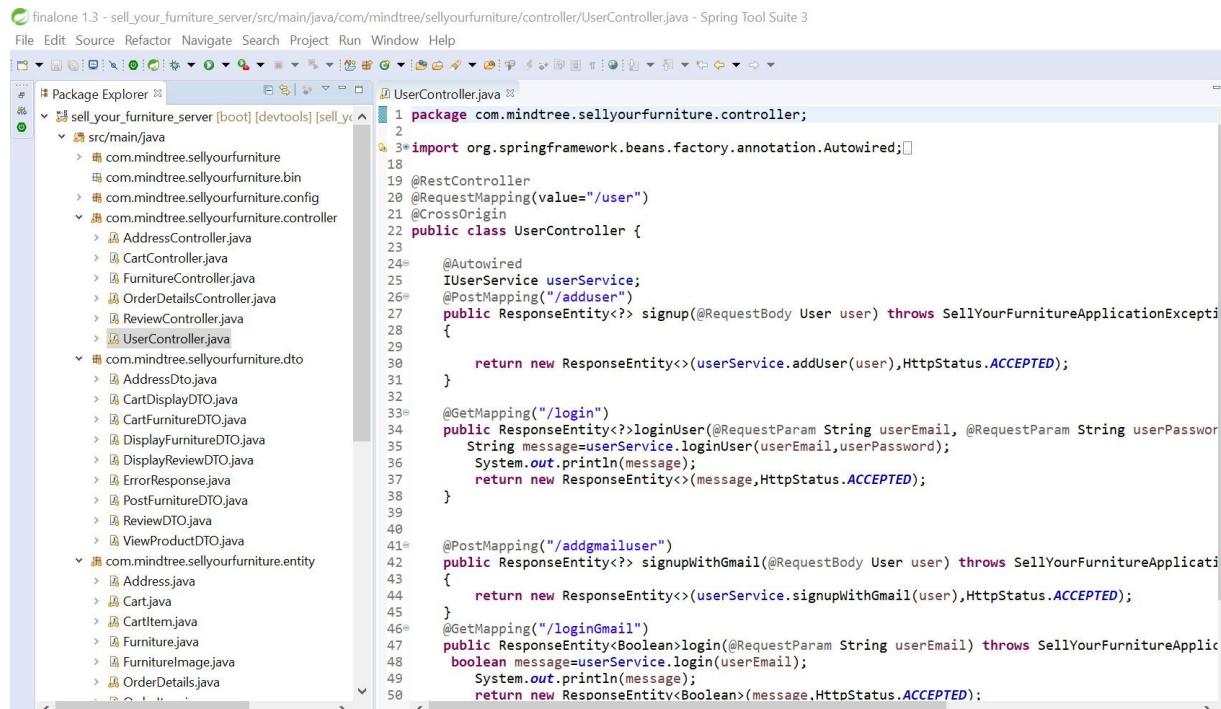
	address_id	address_city	address_country	address_district	address_pincode	address_state	address_street
2	asd	Saudi Arabia	iammu	121212	asd	sd wd	
3	asd	Saudi Arabia	asas	121212	asd	st	
4	asd	Saudi Arabia	dssd	232323	asd	st	
5	asd	Saudi Arabia	sadad	121212	asd	st	
6	asd	Saudi Arabia	o	121212	asd	st	
7	asd	Saudi Arabia	asd	345656	asd	st	
8	asd	Saudi Arabia	ss	121232	asd	st	
9	asd	Saudi Arabia	sad	121212	asd	st	
10	asd	Saudi Arabia	sds	121212	asd	st 1	
11	strina	strina	fnfk	strina	strina	strina	
12	strina	strina	strina	strina	strina	strina	
13	strina	strina	strina	strina	strina	strina	
14	strina	strina	strina	234356	strina	strina	
15	strina	strina	strina	723569	strina	strina	

On the right side of the interface, there is a vertical toolbar with icons for 'Result Grid' (selected), 'Form Editor', 'Field Types', and 'Query Details'.

Fig: 3.7

This is the address table where address of every user is stored.

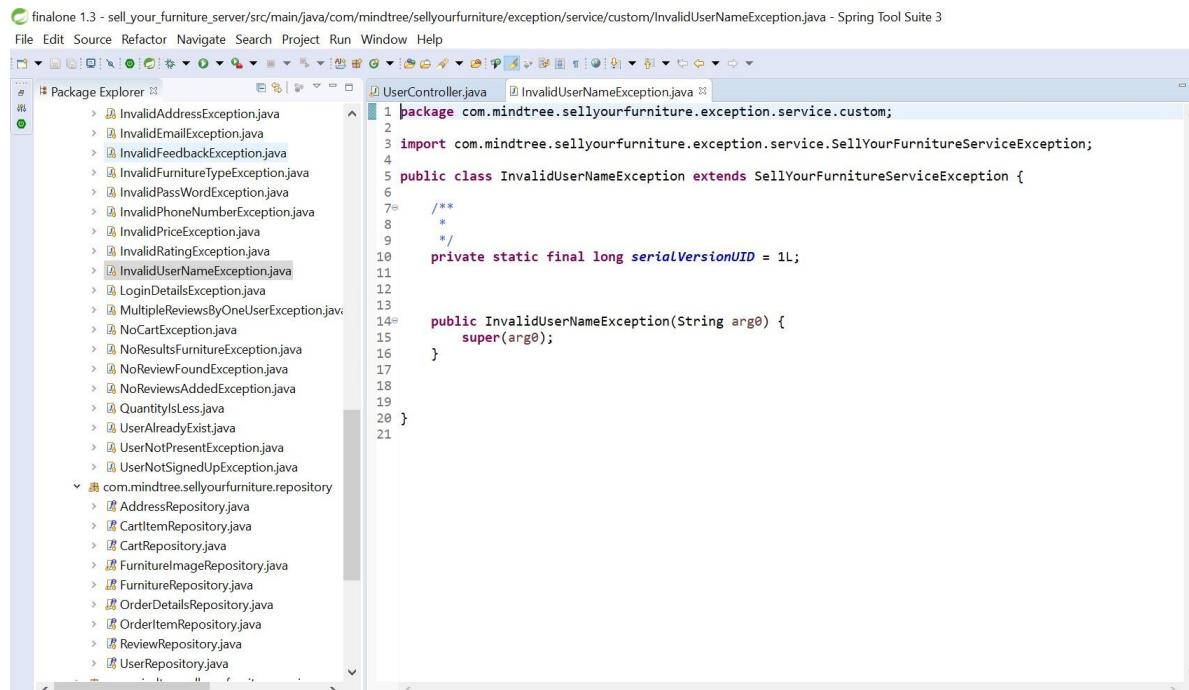
Some snapshot of the spring boot packages:



The screenshot shows the Spring Tool Suite interface with the 'UserController.java' file open in the editor. The code implements a REST controller for user management, utilizing annotations like @RestController, @RequestMapping, and @Autowired. It includes methods for signup, login, and adding a Gmail user.

```
1 package com.mindtree.sellyourfurniture.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @RestController
6 @RequestMapping(value="/user")
7 @CrossOrigin
8 public class UserController {
9
10    @Autowired
11    IUserService userService;
12
13    @PostMapping("/adduser")
14    public ResponseEntity<?> signup(@RequestBody User user) throws SellYourFurnitureApplicationException {
15
16        return new ResponseEntity<>(userService.addUser(user),HttpStatus.ACCEPTED);
17    }
18
19    @GetMapping("/login")
20    public ResponseEntity<?> loginUser(@RequestParam String userEmail, @RequestParam String userPassword) {
21        String message=userService.loginUser(userEmail,userPassword);
22        System.out.println(message);
23        return new ResponseEntity<>(message,HttpStatus.ACCEPTED);
24    }
25
26    @PostMapping("/addgmailuser")
27    public ResponseEntity<?> signupWithGmail(@RequestBody User user) throws SellYourFurnitureApplicationException {
28        return new ResponseEntity<>(userService.signupWithGmail(user),HttpStatus.ACCEPTED);
29    }
30
31    @GetMapping("/loginGmail")
32    public ResponseEntity<Boolean> login(@RequestParam String userEmail) throws SellYourFurnitureApplicationException {
33        boolean message=userService.login(userEmail);
34        System.out.println(message);
35        return new ResponseEntity<Boolean>(message,HttpStatus.ACCEPTED);
36    }
37
38 }
```

Fig: 3.8

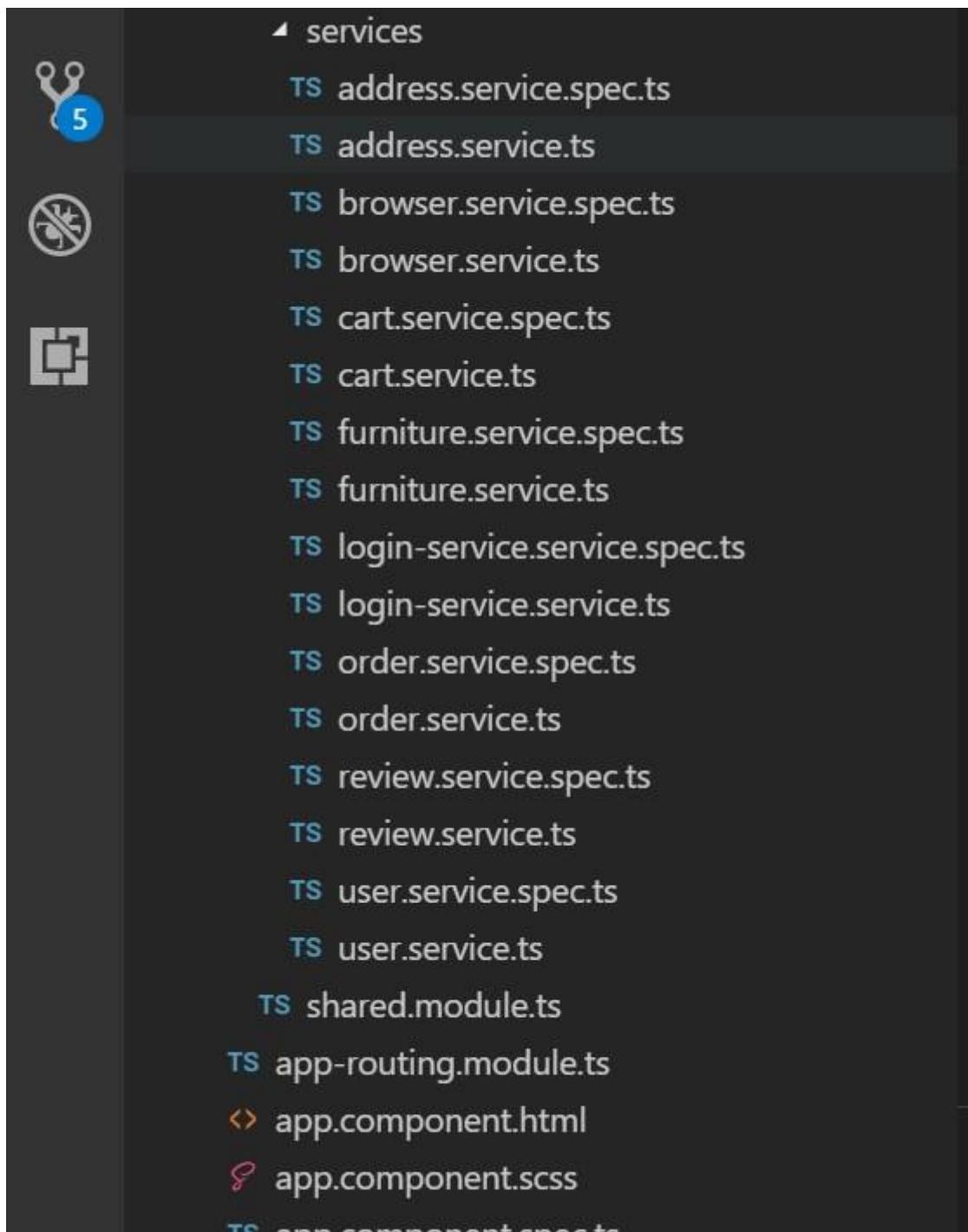


The screenshot shows the Spring Tool Suite interface with the 'InvalidUserNameException.java' file open in the editor. This class extends the 'SellYourFurnitureServiceException' and defines a constructor that takes a string argument.

```
1 package com.mindtree.sellyourfurniture.exception.service.custom;
2
3 import com.mindtree.sellyourfurniture.exception.service.SellYourFurnitureServiceException;
4
5 public class InvalidUserNameException extends SellYourFurnitureServiceException {
6
7    /**
8     *
9     */
10    private static final long serialVersionUID = 1L;
11
12
13
14    public InvalidUserNameException(String arg0) {
15        super(arg0);
16    }
17
18
19
20 }
```

Fig: 3.9

Some snapshots of the angular component:



The screenshot shows a file explorer interface with a dark theme. On the left, there are three icons: a wrench and gear icon with the number '5' (services), a circular icon with a crossed-out gear (app-routing.module.ts), and a square icon with rounded corners (app.component.ts). The main pane lists the following files:

- ▲ services
 - TS address.service.spec.ts
 - TS address.service.ts
 - TS browser.service.spec.ts
 - TS browser.service.ts
 - TS cart.service.spec.ts
 - TS cart.service.ts
 - TS furniture.service.spec.ts
 - TS furniture.service.ts
 - TS login-service.service.spec.ts
 - TS login-service.service.ts
 - TS order.service.spec.ts
 - TS order.service.ts
 - TS review.service.spec.ts
 - TS review.service.ts
 - TS user.service.spec.ts
 - TS user.service.ts
 - TS shared.module.ts
- TS app-routing.module.ts
- ◁ app.component.html
- ♀ app.component.scss
- TS app.component.spec.ts

Fig 3.10

```
▲ services
  TS address.service.spec.ts
  TS address.service.ts
  TS browser.service.spec.ts
  TS browser.service.ts
  TS cart.service.spec.ts
  TS cart.service.ts
  TS furniture.service.spec.ts
  TS furniture.service.ts
  TS login-service.service.spec.ts
  TS login-service.service.ts
  TS order.service.spec.ts
  TS order.service.ts
  TS review.service.spec.ts
  TS review.service.ts
  TS user.service.spec.ts
  TS user.service.ts
  TS shared.module.ts
TS app-routing.module.ts
▷ app.component.html
♀ app.component.scss
```

Fig: 3.11

```
▲ signin
  ◄ signin.component.html
  ♀ signin.component.scss
  TS signin.component.spec.ts
  TS signin.component.ts

  ▲ signup
    ◄ signup.component.html
    ♀ signup.component.scss
    TS signup.component.spec.ts
    TS signup.component.ts
    TS user-routing.module.ts
    TS user.module.ts

  ▲ shared
    ▲ models
      TS AddressDto.ts
      TS AddReviewDTO.ts
      TS AppConstants.ts
      TS displayFurnitureDTO.ts
      TS DisplayReview.ts
      TS furniture-cart-dto.spec.ts
      TS furniture-cart-dto.ts
      TS furniture-cart.spec.ts
```

Fig: 3.12

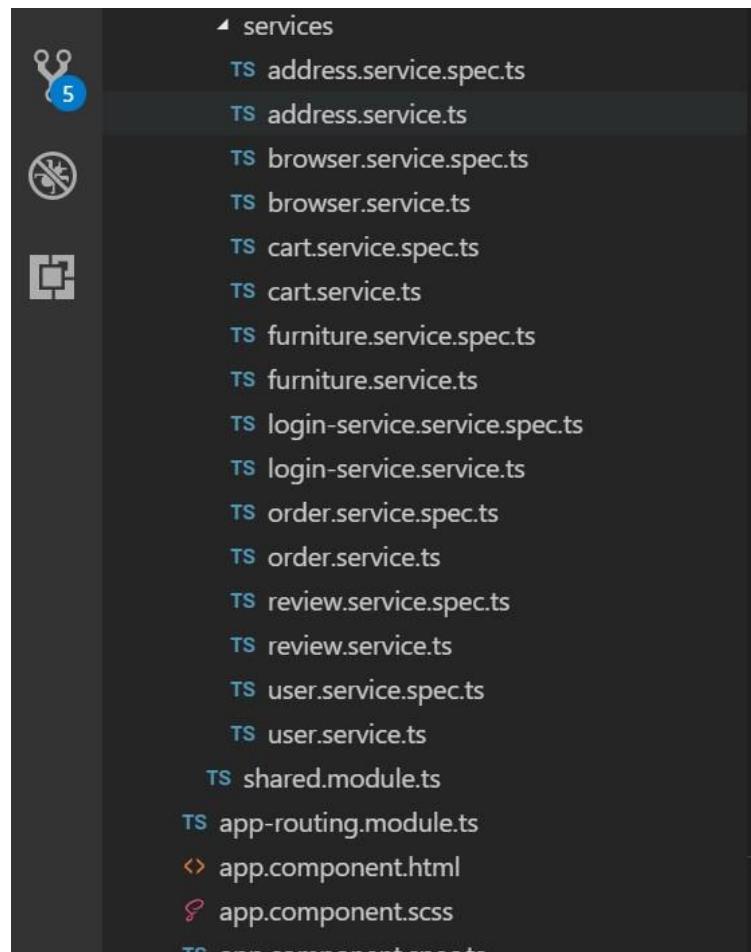


Fig: 3.13

RESPONSE ENTITY:-

In this application I have also used the response entity for the desired output.

Some of the examples of the usages of response entity is as in following code:

```
package com.mindtree.sellyourfurniture.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import com.mindtree.sellyourfurniture.entity.User;

import com.mindtree.sellyourfurniture.exception.SellYourFurnitureApplicationException;
```

```

import com.mindtree.sellyourfurniture.service.IUserService;

@Controller
@RequestMapping(value="/user")
@CrossOrigin
public class UserController {

    @Autowired
    IUserService userService;

    @PostMapping("/adduser")
    public ResponseEntity<?> signup(@RequestBody User user) throws SellYourFurnitureApplicationException {
        return new ResponseEntity<>(userService.addUser(user),HttpStatus.ACCEPTED);
    }

    @GetMapping("/login")
    public ResponseEntity<?> loginUser(@RequestParam String userEmail, @RequestParam String userPassword) throws SellYourFurnitureApplicationException {
        String message=userService.loginUser(userEmail,userPassword);
        System.out.println(message);
    }
}

```

```

        return new ResponseEntity<>(message,HttpStatus.ACCEPTED);

    }

    @PostMapping("/addgmailuser")

    public ResponseEntity<?> signupWithGmail(@RequestBody User user) throws
SellYourFurnitureApplicationException

    {

        Return new ResponseEntity<>(userService.signupWithGmail(user),HttpStatus.ACCEPTED);

    }

    @GetMapping("/loginGmail")

    public  ResponseEntity<Boolean>login(@RequestParam  String  userEmail)  throws
SellYourFurnitureApplicationException {

        Boolean message=userService.login(userEmail);

        System.out.println (message);

        return new ResponseEntity<Boolean>(message,HttpStatus.ACCEPTED);

    }

}

```

As the above code is of the controller package where I have used response entity. Like this only , I have use the response entity in every controller.

```
package com.mindtree.sellyourfurniture.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import com.mindtree.sellyourfurniture.dto.CartFurnitureDTO;
import com.mindtree.sellyourfurniture.service.ICartService;

@RestController
@RequestMapping("/cart")
```

```

    @CrossOrigin(origins = "*")

    public class CartController {

        @Autowired

        private ICartService cartService;

        @PostMapping("/addfurniture")

        public ResponseEntity<?> addFurnitureToCart(@RequestBody CartFurnitureDTO
        cartFurniture){

            return new ResponseEntity<>(cartService.addToCart(cartFurniture),HttpStatus.OK);

        }

        @PostMapping("/changequantity")

        public ResponseEntity<?> changeQuantityOfCart(@RequestBody CartFurnitureDTO
        cartFurniture){

            return new ResponseEntity<>(cartService.addToCart(cartFurniture),HttpStatus.OK);

        }

        @GetMapping("/getfurniture")

        public ResponseEntity<?> getFurnituresOfCart(@RequestParam("email") String
        email){

            return new ResponseEntity<>(cartService.getCartDetails(email),HttpStatus.OK);

        }
    }

```

```

    @PostMapping("/removecart")

    public ResponseEntity<?> deleteFurnitureOfCart(@RequestBody CartFurnitureDTO
    cartFurniture){

        return new
        ResponseEntity<>(cartService.deleteFurnitureFromCart(cartFurniture),HttpStatus.OK
        );

    }

    @GetMapping("/getfurnitureids")

    public ResponseEntity<?> allFurnitureIdOfCart(@RequestParam("email") String
    email){

        return new
        ResponseEntity<>(cartService.getFurnitureIdsOfCart(email),HttpStatus.OK);

    }

}

```

This is of the cart controller.

III.3 UNIT TESTING:

As I have also done the joint testing for the server side code and unit testing using jasmine and karma for the client side code.

From this testing I got the following output:

Frontend code coverage :

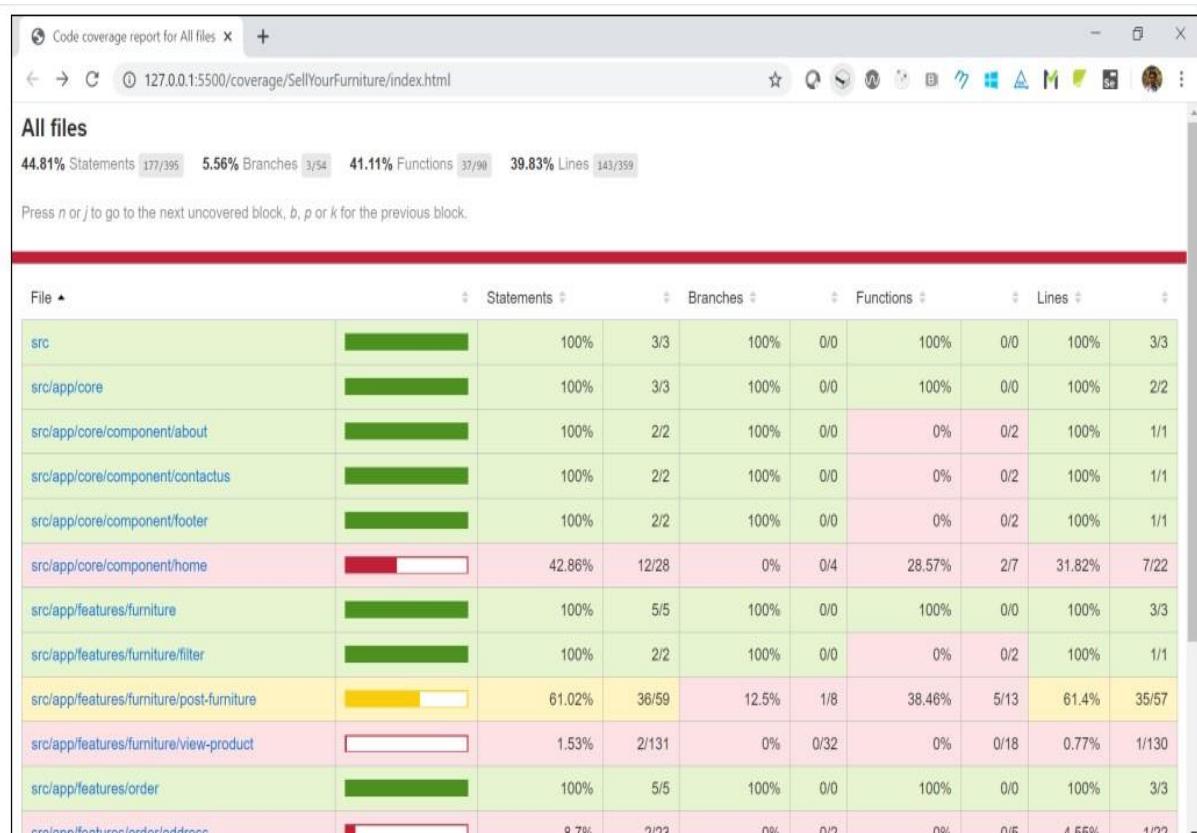


Fig: 3.14

As the required code coverage must be above 30%.so mine was 44.8%.

Backend Code Coverage:

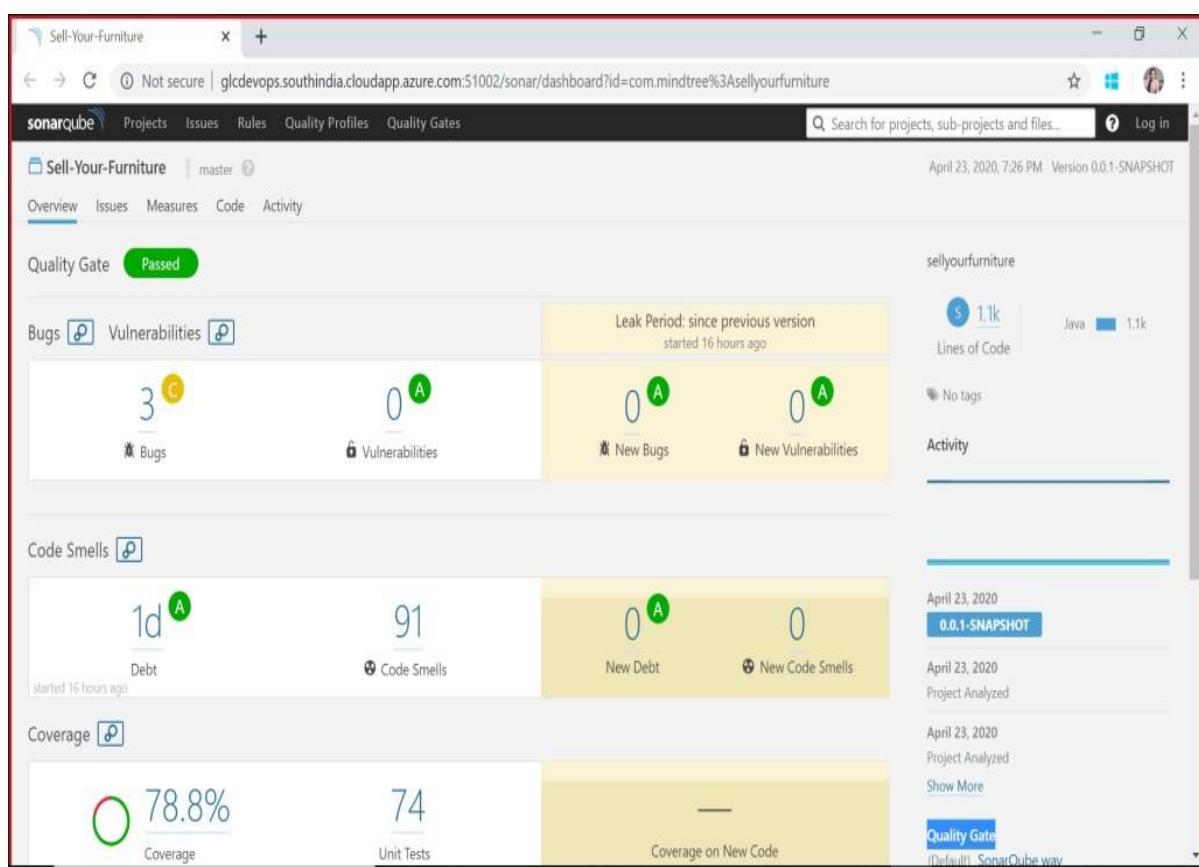


Fig: 3.15

As the backend code coverage must be 70 %. Mine was 78.8%.

IV. CONCLUSION

- As a person who want to sell his/her furniture can sell in on this website without roaming and searching for the buyer.
- So it will save the time for both of the seller as well as for the buyer.
- Buyer can also give reviews.
- Authenticated users.
- Time Saving

V. BIBLIOGRAPHY

1. <https://angular.io/>
2. https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm
3. <https://en.wikipedia.org/wiki/Mindtree>