E-DAC May ***2021 E-DAC May 2021***

MEALS ON WHEELS



–

**Guided By : Snehal somwanshi**

Submitted By :

**Group No: 04**

* **Ganesh Gogate 210543081027**
* **Geetanjali Kale 210543081028**
* **Mahesh Nazare 210543081053**
* **Prashant Nalawade 210543081080**

**Table of Contents**

**1. Introduction 2**

1.1 Document Purpose **2**

1.2 Project Background

1.3 Aim & Objectives **2**

**2. Business Requirements Overview**……………………………………………………………………………………**3**

**3.Functional Requirements** 4

3.1 Customer Module **5**

3.2 Service Provider Module **6**

3.3 Admin Module **7**

**4.** **Non-Functional Requirement………………………………………………………………8**

**5. Use Case Diagram 8**

5.1 Admin **9**

5.2 Customer **10**

5.3 Service Provider **11**

**6. Database Design 12**

1 Admin **12**

2 Customer **13**

3 Order Table **13**

4 Order Details…………………………………………………………………………………… **14**

5 Menu Table **14**

6 Service Provider **14**

**7. E-R Diagram………………………………………………………………………………………15**

**8. Snapshots 25**

**9. Conclusion 26**

**List of Figures**

**Use Case Diagrams 8**

Fig 1 Admin 8

Fig 2 Customer 9

Fig 3 Service Provider 10

Fig 4 ER Diagram 15

**CHAPTER 1. MEALS ON WHEELS**

**Introduction**

* 1. **Document Purpose:**

The purpose of this document is to provide in-depth description of design and implementation details of the system, as well as descriptions of all available functionality and plans for evolution. In addition, user manuals and trouble-shooting tips have been included for all three components to give the reader a clear idea of intended typical use cases for the system.

* 1. **Project Background :**

In the existing system, user needs to waste a lot of time and physical efforts in finding the good Tiffin service provider and it also helps Tiffin service providers to known about their customers and it is hard for Tiffin service provider to manage their orders and also getting new customers.

In the present scenario people have to physically visit the tiffin service provider and have to make payment through cash mode most of the times due to unawareness of advanced technologies at certain places.

In this method time as well as physical work is required, among which time is something that no one has in ample amount.

Also from the owner’s point of view maintaining data record and the accounts in physical file is cumbersome and tedious work to do. And also it is full of risk as anyone can access it and modify the data.

* 1. **Aim & Objectives:**

“Meals on wheels” is an application which helps customers to get Tiffin from our platform by selecting whichever Tiffin service they like and can easily order their lunch or dinner and Tiffin will be delivered at their doorstep. Tiffin providers can start their business by registering on our platform, they can also advertise their business, connect with customers, and maintain their order history records.



**CHAPTER 2.**

**2. Business Requirements Overview:**

* Meals on Wheels is the public web application.
* Meals on Wheels will be opened to the global, but in the phase 1, the main target is in the India.
* There are mainly two types of user. One is the Customers and other is Tiffinwala.
* Customers can search for tiffin in their area .
* Meals on Wheels provides the functions which connect the tiffinwala and the customers efficiently.
* Meals on Wheels could be maintained by Administrator.

# CHAPTER 3.

# 3. Functional Requirements Overview:

“Meals on Wheels” System consists of Four modules described as below.

* **Customer Module**
* **Tiffine Provider Module**
* **Login**
* **Admin Module**

## Customer:

* + - Customer can see list of Tiffin service provider and his availability and contacts details.
    - On selection of a particular Tiffin service provider, customer will be able to see current session menu from Tiffin service provider.
    - Customer will be able to update Tiffin as per his requirements whether to add extra roti, add sweets.
    - After selected required services, customer can visit the payment page and add payment details.
    - After successful payment, user will get the receipt and will get notification about his payment.

## 3.2 Tiffin service provider:

## 

* Tiffin service provider can add new menu (through menu Form).
  + Tiffin service provider can see his own menu with the availability status and also can update his menu (if required).
* Tiffin service provider can check the list of orders under the Orders history section. List will have following details –
  + Date of order

* + Order Id
  + Customer details
  + Payment status
  + Service (pending / delivered)



Fig.Tiffin service provider

## For Tiffin Service provider, another list form is provided.

**MENU LIST FORM -**

After tiffin service provider login, they are redirected to menu page.

Where they will see all the Tiffin service providers and their menu. Menu has certain items like below:

* Service Name
* Date
* Session
* Roti
* Rice
* Sabji
* Dessert
* Drinks



Fig. Tiffin Menu

* 1. **LOGIN**

All, Customers, Tiffin service provider, Admin can login through the same Login Page.

Login requires –

* + 1. Email Id
    2. Password

After successfully validating their credentials, they will be redirected to their respective Home Pages.

In case of Unsuccessful Login, they will be given error message on the login screen.



3.4 Admin Module :

* This System should provide all functionality to admin how to handle the System.
* Admin has controls of all the functionalities of customer and Tiffin Provider.
* Admin will provide Login credentials to Customer and Tiffin Provider.

**CHAPTER 4.**

**4. Non-Functional Requirement:**

* Because the design patterns of the Meals on Wheels System are pretty much standard for a web application, the non-functional requirements of the system are very straightforward.
* Although written using Google Web Toolkit, the application is cross-compiled to Java Script, along with a Spring Boot backend, all of which are supported by any reasonably well maintained web server. All of the application data is stored in a MySQL database.
* The website should use professional design, look and feel and color scheme.
* Users will have no limitations for accessing the application through Internet. The portal being an internet application, it is difficult specify exact number of visitor or users. Hence we will target the system to support between 5 and 10 million users on launch of phase 1.
* Being a public website, the site must follow general usability guidelines for menus, navigation, colors, links and other actions provided on the screens.
* The system should be designed in such a manner that user will be able to complete tasks in minimum number of steps.

## CHAPTER 5.

## 5. Use-Case Diagram:

**5.1 Admin:**



Fig. Use-Case Diagram for Admin

5.2 Customer:



Fig. Use-Case Diagram for Customer

5.3 Servicce Provider:

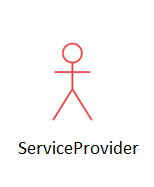


Fig. Use-Case Diagram for Service Provider

**CHAPTER 6.**

**6. Database Design:**

**Admin:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Loginid | Integer | NO | Primary key | Null |  |
| Username | Varchar(45) | NO |  | Null |  |
| password | integer | NO |  | Null |  |
| role | Varchar(45) | NO |  | Null |  |
|  |  |  |  |  |  |

Customer:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Custid | Integer | NO | Primary key | Null |  |
| Name | Varchar(45) | NO |  | Null |  |
| Address | Varchar(100) | NO |  | Null |  |
| email | Varchar(45) | NO |  | Null |  |
| mobile | Integer | NO |  | Null |  |
| birthdate | Integer | NO |  | Null |  |
| Loginid Fk | Integer | NO | Foreign key | Null |  |

Order Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Order Id | Integer | NO | Primary key | Null |  |
| Custid | Integer | NO | Foreign Key | Null |  |
| Serviceproviderid | Integer | NO | Foreign Key | Null |  |
| Grosstotalamnt | Integer | NO |  | Null |  |
| Order date | Integer | NO |  | Null |  |
| Payment mode | Double | NO |  | Null |  |
| Transaction id | Integer | NO |  | Null |  |
| Status |  | NO |  |  |  |

Order Details:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| OrderDetail Id | Integer | NO | Primary Key | Null |  |
| Orderid | Integer | NO |  | Null |  |
| menu id | Integer | NO |  | Null |  |
| quantity | Integer | NO |  | Null |  |
| Total amount | Double | NO |  | Null |  |

Menu Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Menuid | Integer | NO | Primary key | Null |  |
| Serviceproviderid | Integer | NO | Forign key | Null |  |
| Name | Varchar(45) | NO |  | Null |  |
| Amount | Double | NO |  | Null |  |
| rate | Integer | NO |  | Null |  |
| Date | Date Time | NO |  | Null |  |

ServiceProvider:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Serviceproviderid | Integer | NO | Primary key | Null |  |
| Name | Varchar(45) | NO |  | Null |  |
| Address | Varchar(45) | NO |  | Null |  |
| email | Varchar(45) | NO |  | Null |  |
| mobile | Varchar(45) | NO |  | Null |  |
| Foodlicenceno | Integer | NO |  | Null |  |
| Loginid FK | Integer | NO | Foreign key | Null |  |
|  |  |  |  |  |  |

**CHAPTER 7**

7. ER-Diagram:

**Order Details**

**Admin Login Table**

**Customer Login**

**Order Table**

**Customer**

**Login**

**Menu Table**

**Service Provider**

**Dao**

**Spring Boot**

**JSP**

**React JS**

System Architecture:

**First Tier**

**Second Tier** **Server UI**

**Server**

**Processing**

**Third Tier**

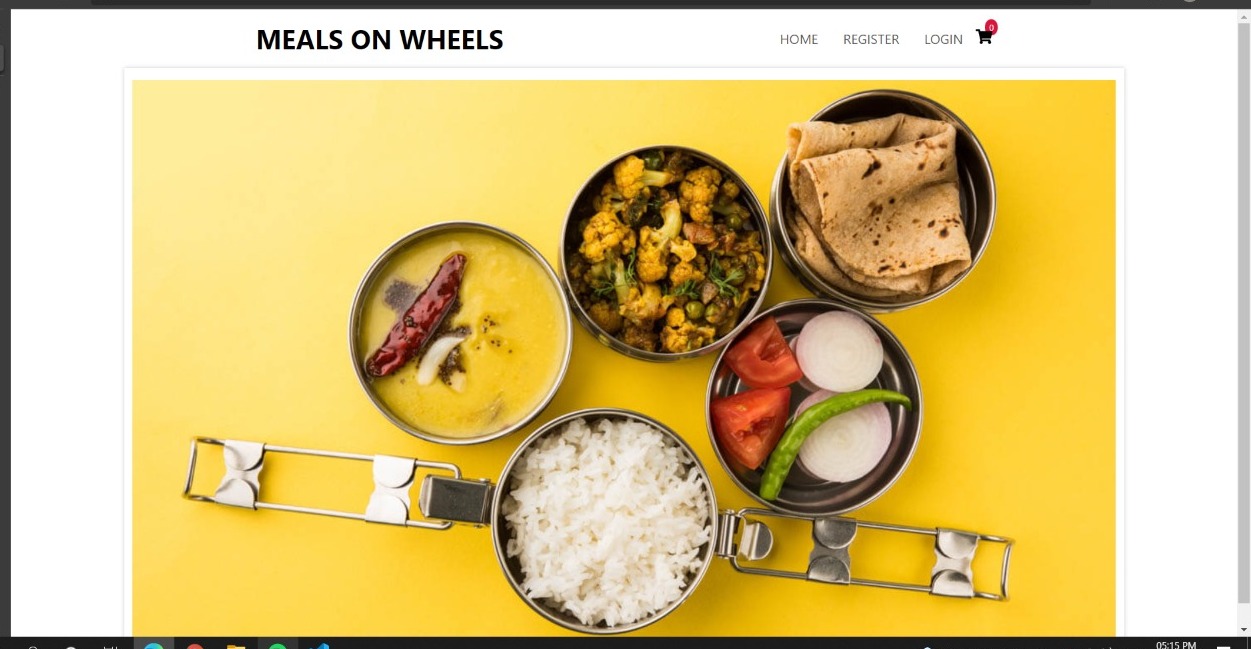
**MySQL**

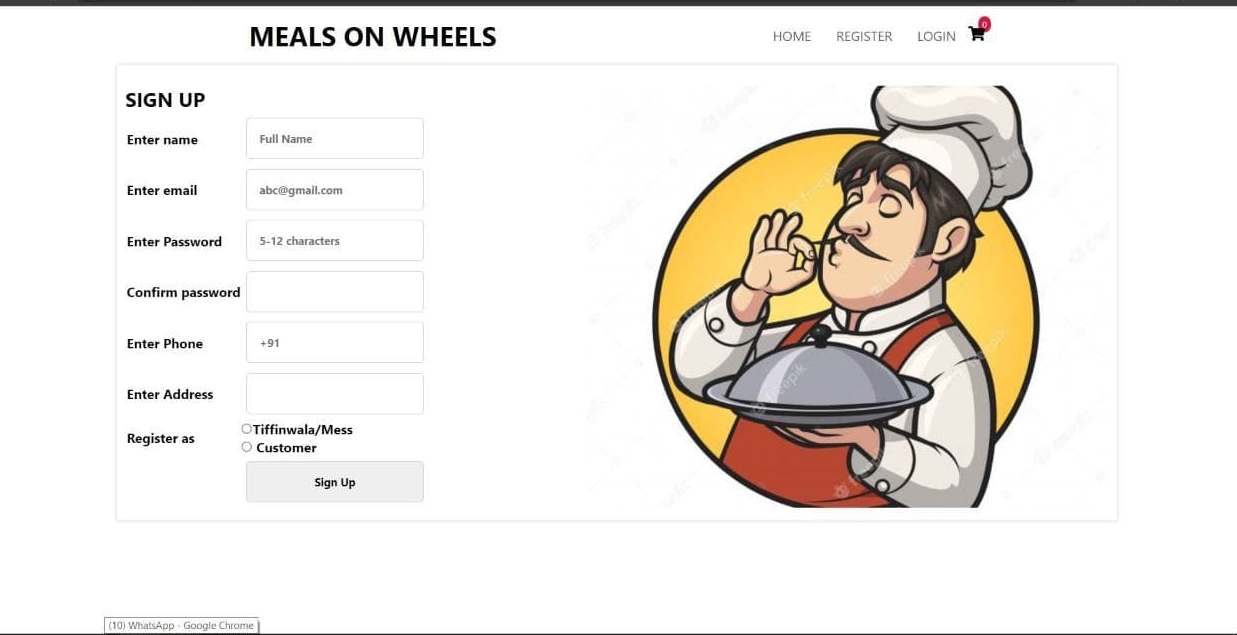
**Database**

**CHAPTER 8**

**8. Snapshots:**

**8.1 Home Page:**

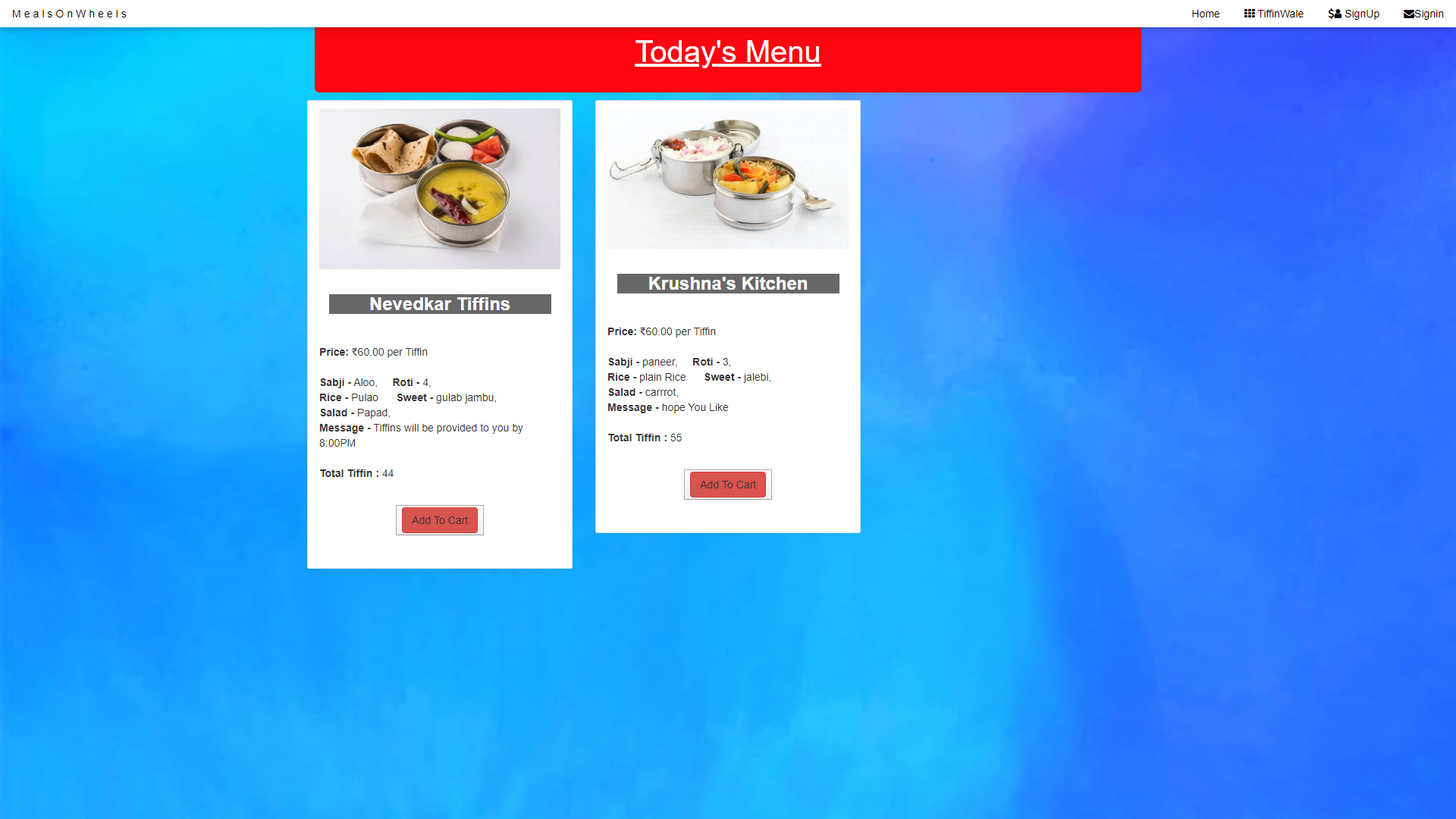


Registration Page :

Login Page:

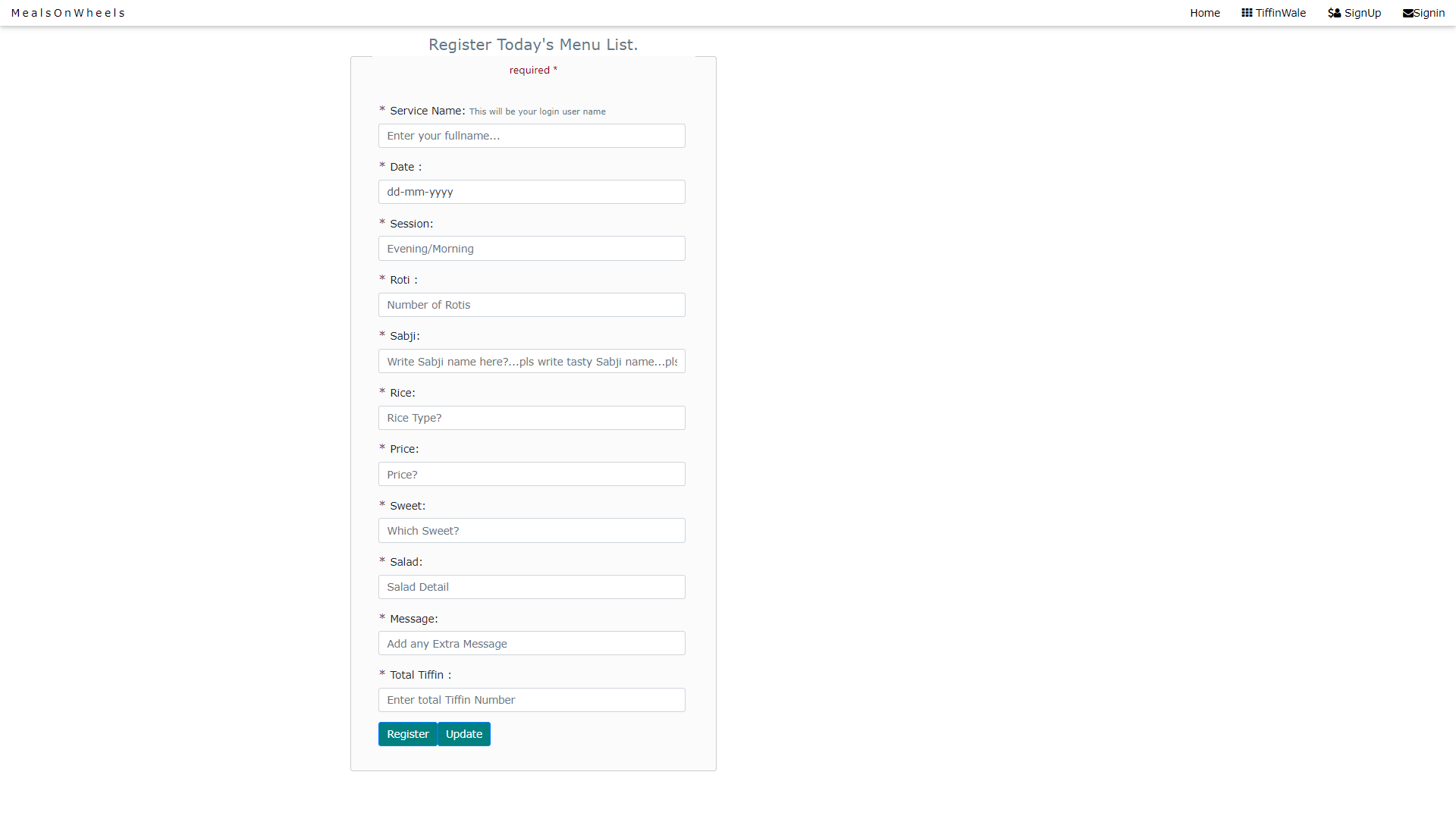
If user is customer it will be redirect to the menulist page

Menulist page :

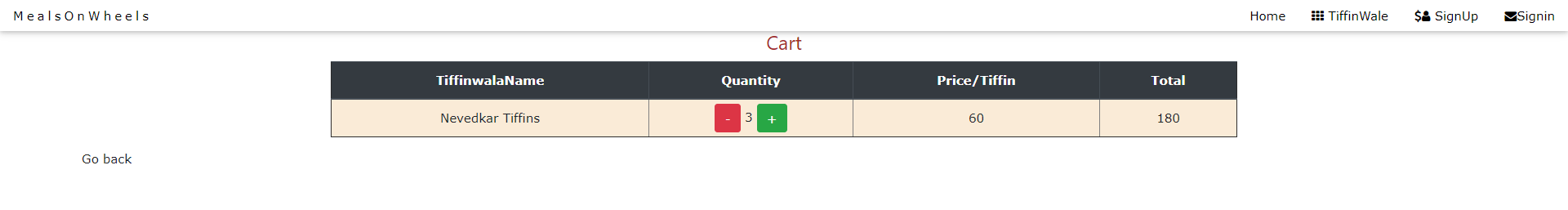


If user is Tiifinwala it will be redirect to the insert menu page

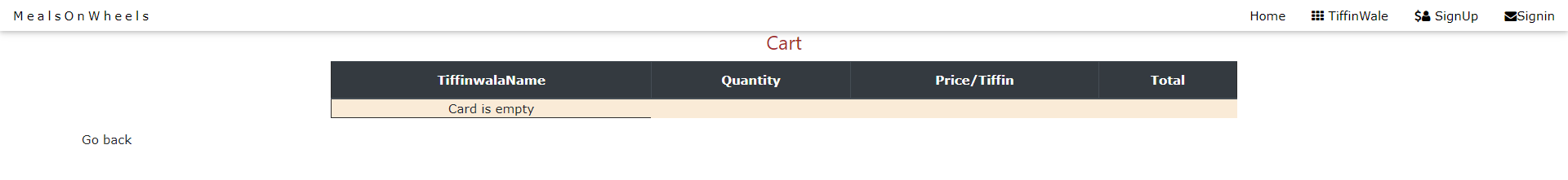
Menu page:



Customer can increase the quantity of orders according to his number of tiffins total amount to be paid will be shown --

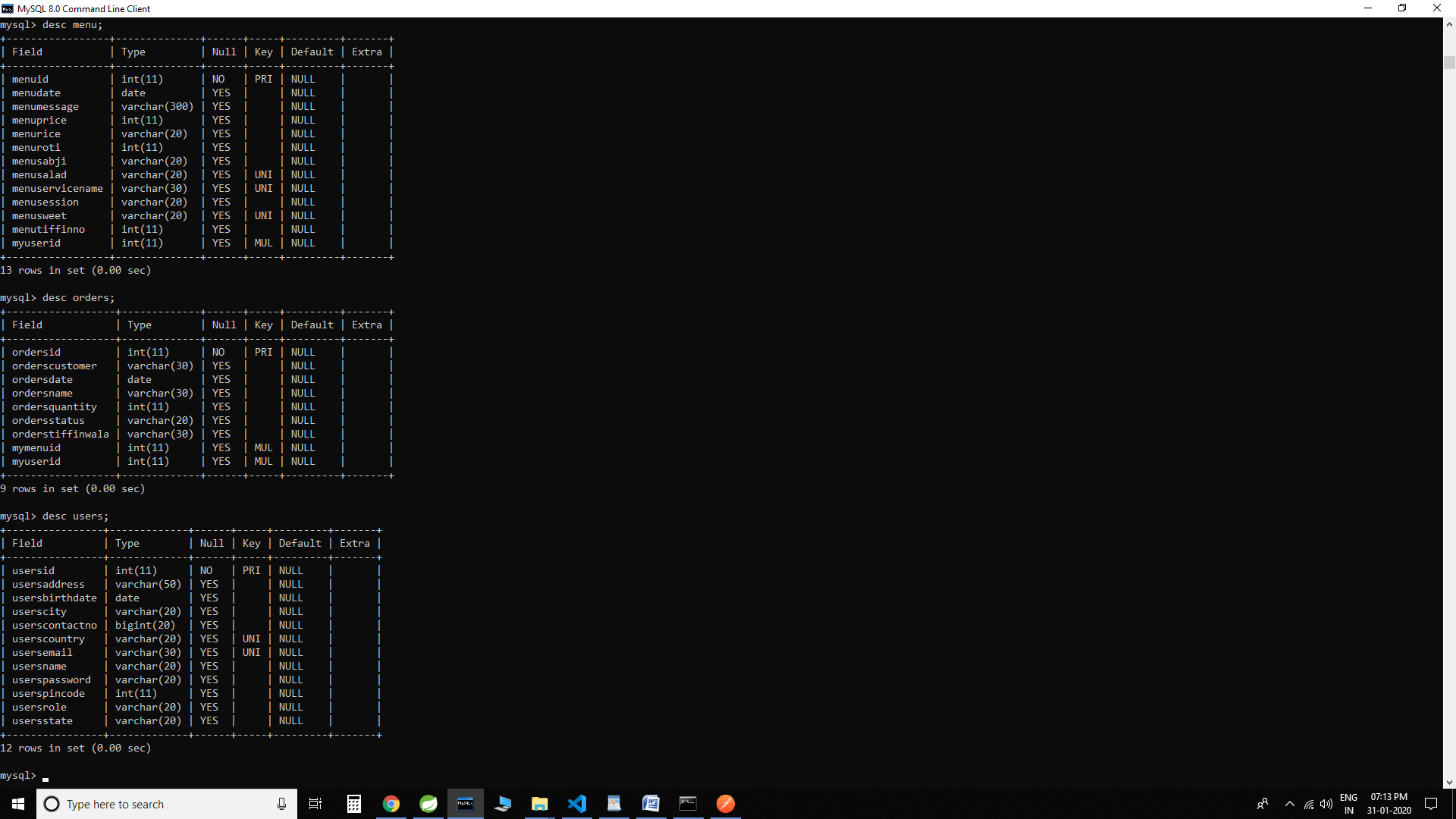


Customer can check status if cart is empty cart will show empty message

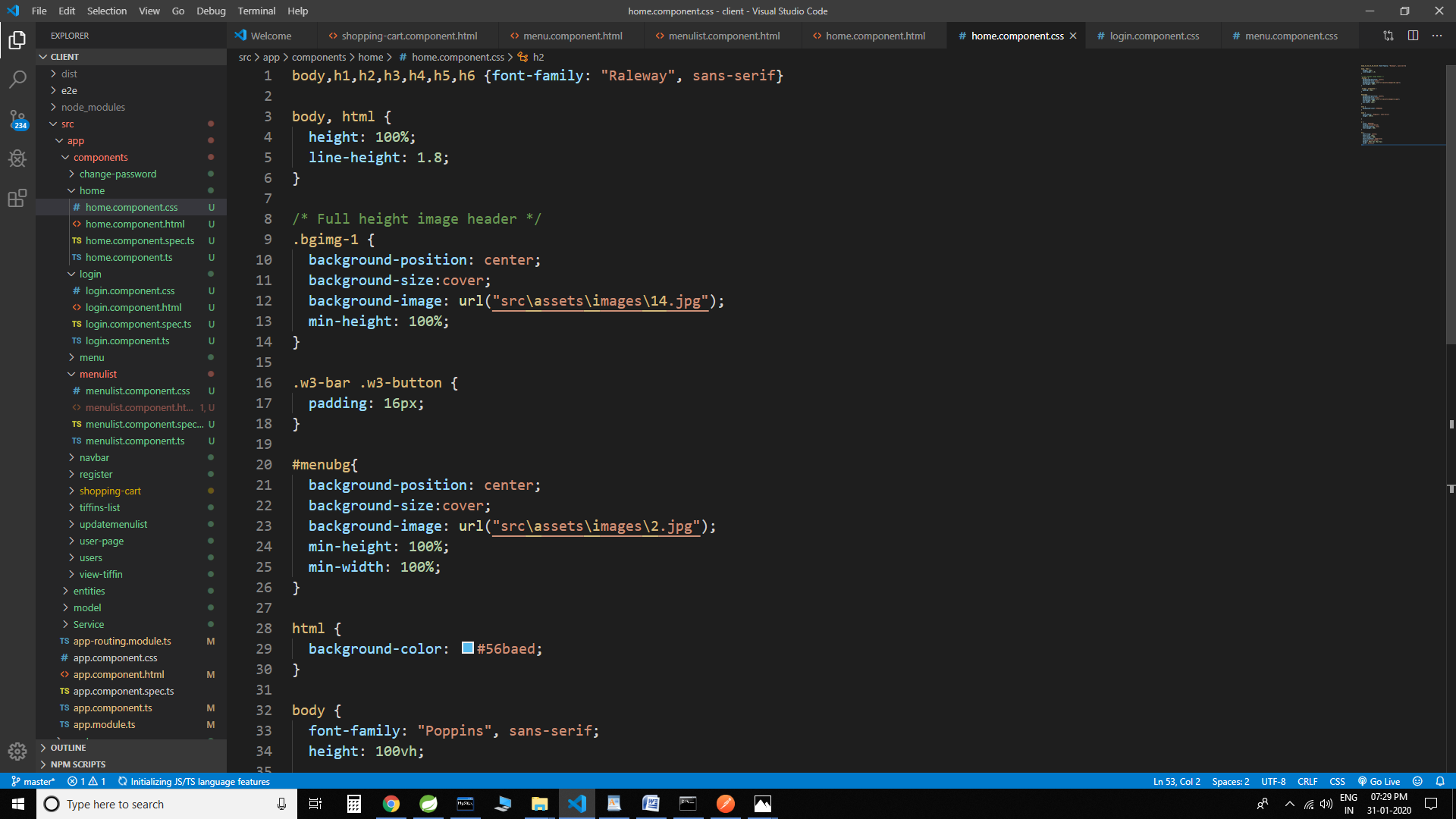


Database:

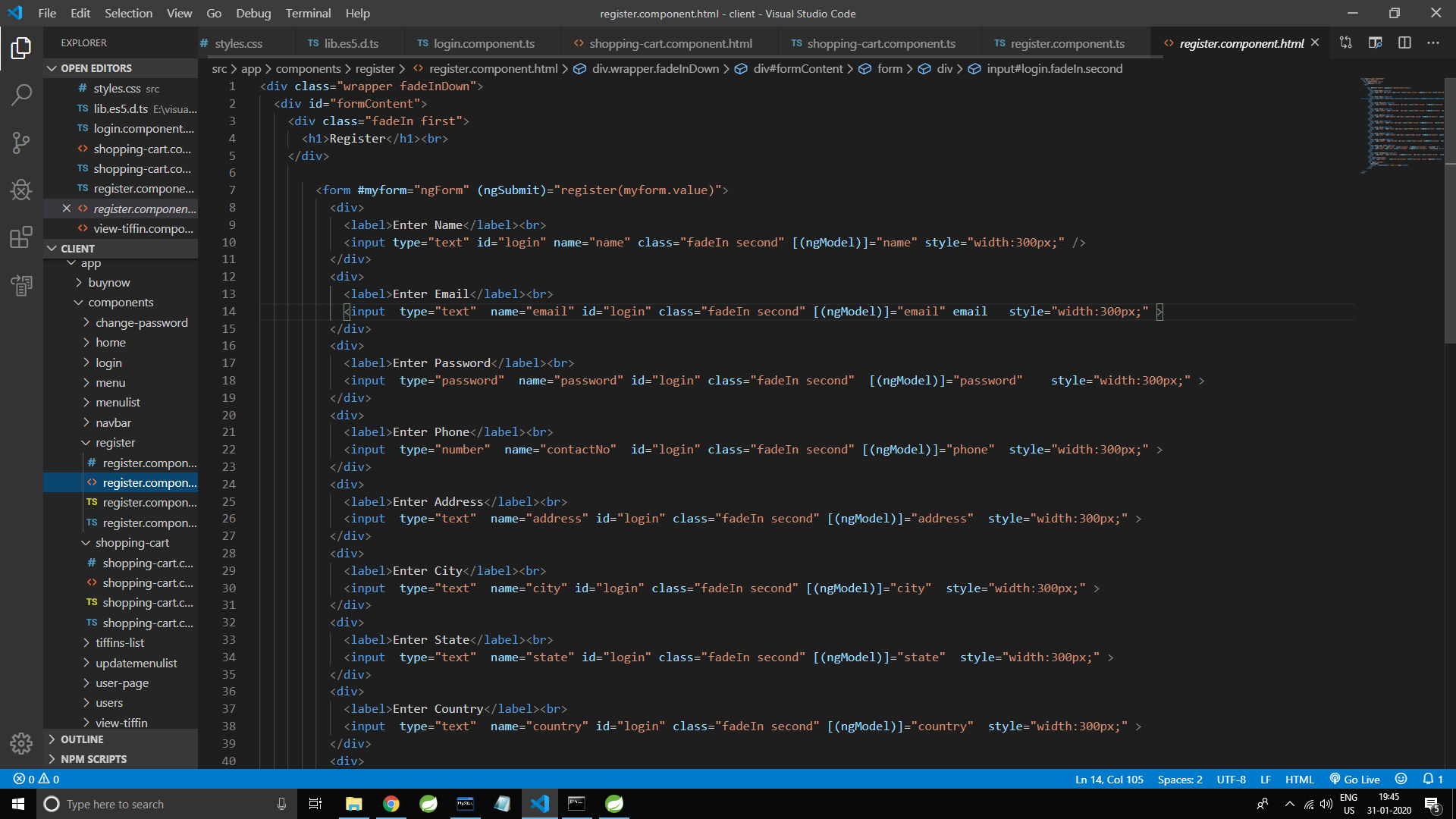
Different tables details which we use while implementing the project



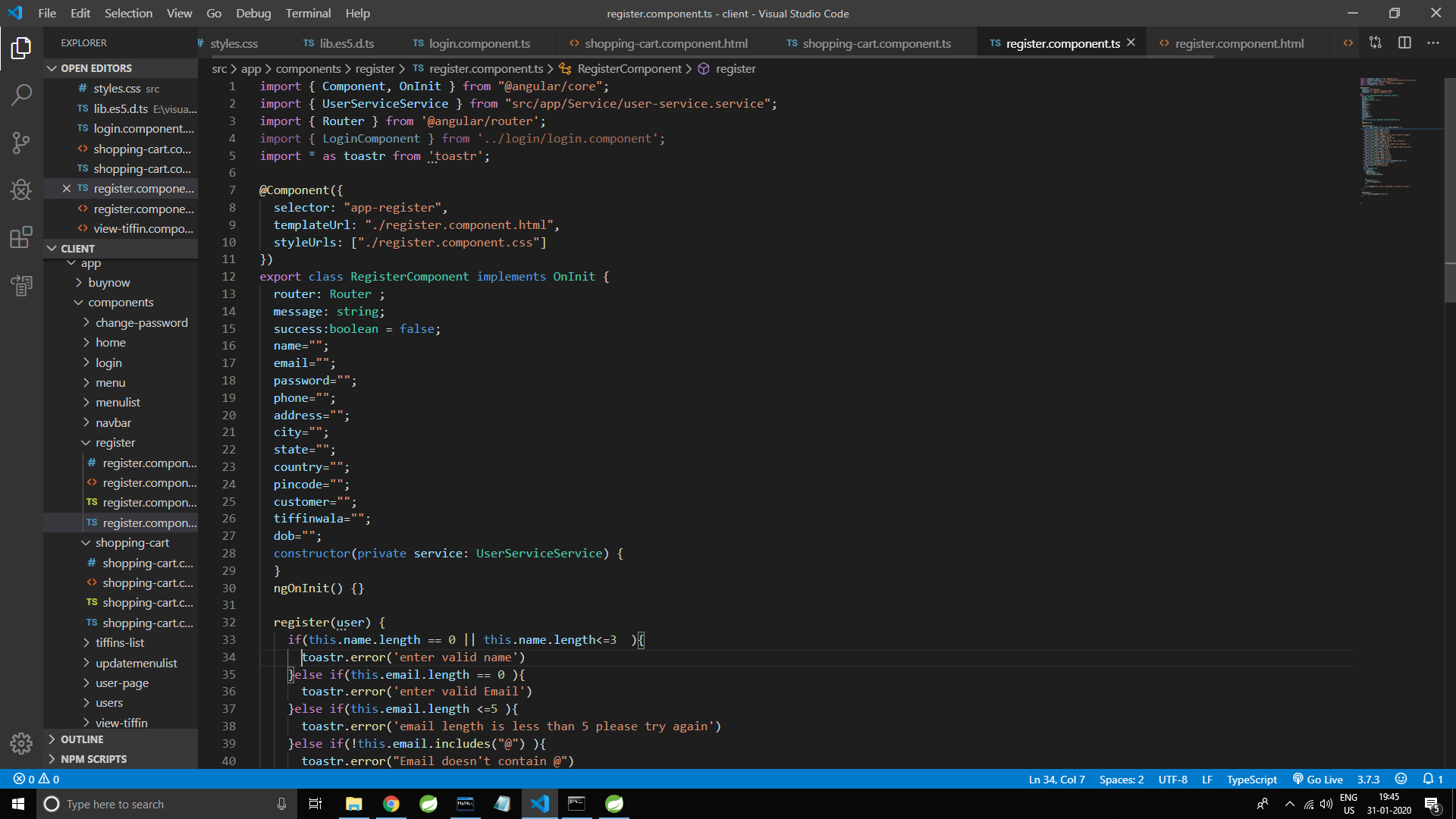
Coding: -CSS is the style sheet language for describing the presentation and design of web pages including colors, fonts, and layouts. It is mainly designed to enable the distinction between presentation and content, including colors, layouts, and fonts.



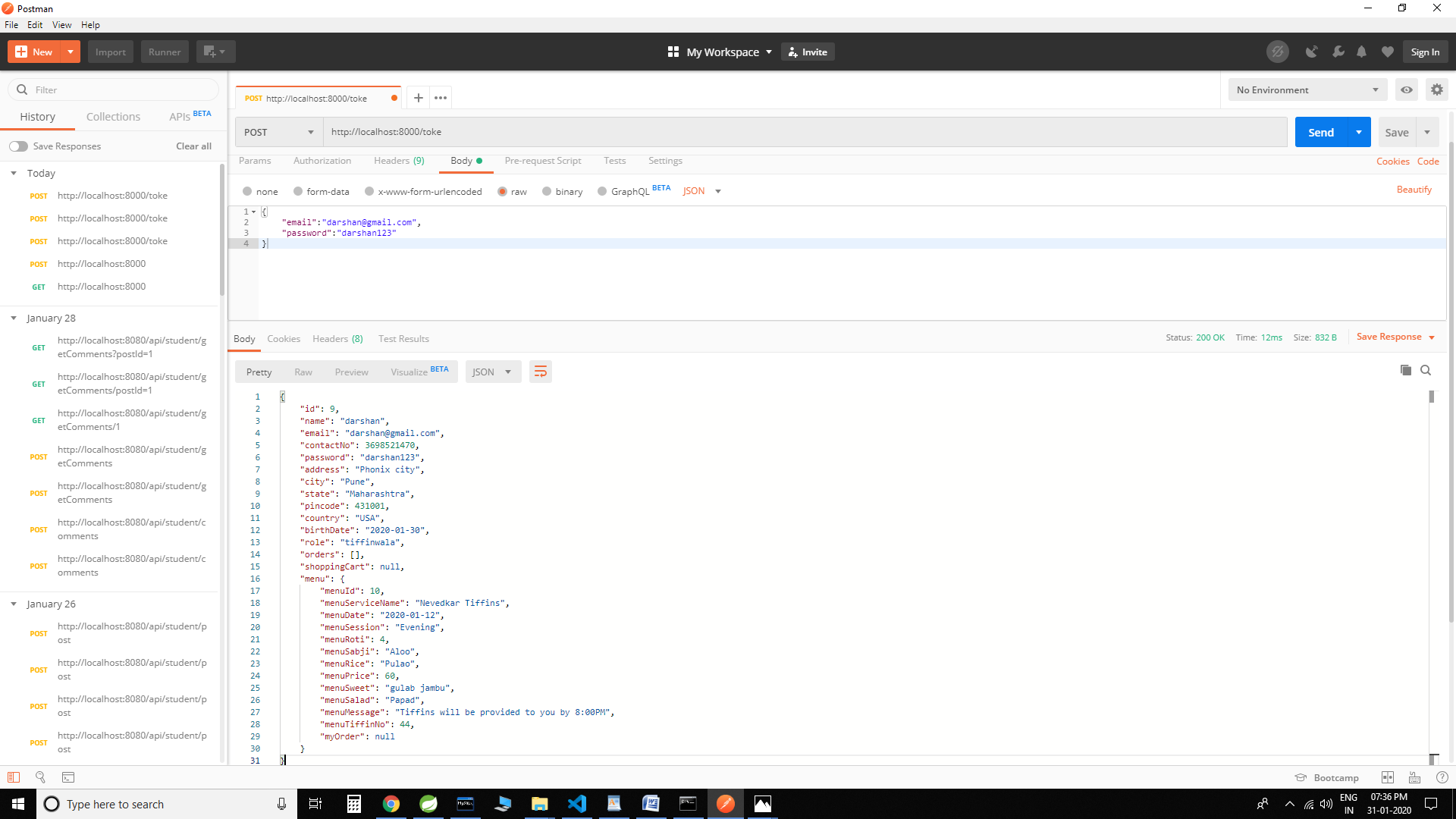
HTML code –use to create web pages we can use html to format text as titles and headings, to arrange graphics on a webpage, to link to different pages within a website, and to link to different websites.



TS code-- TypeScript is a superset of the JavaScript language that has a single open-source compiler and is developed mainly by a single vendor: Microsoft. The goal of TypeScript is to help catch mistakes early through a type system and to make JavaScript development more efficient.

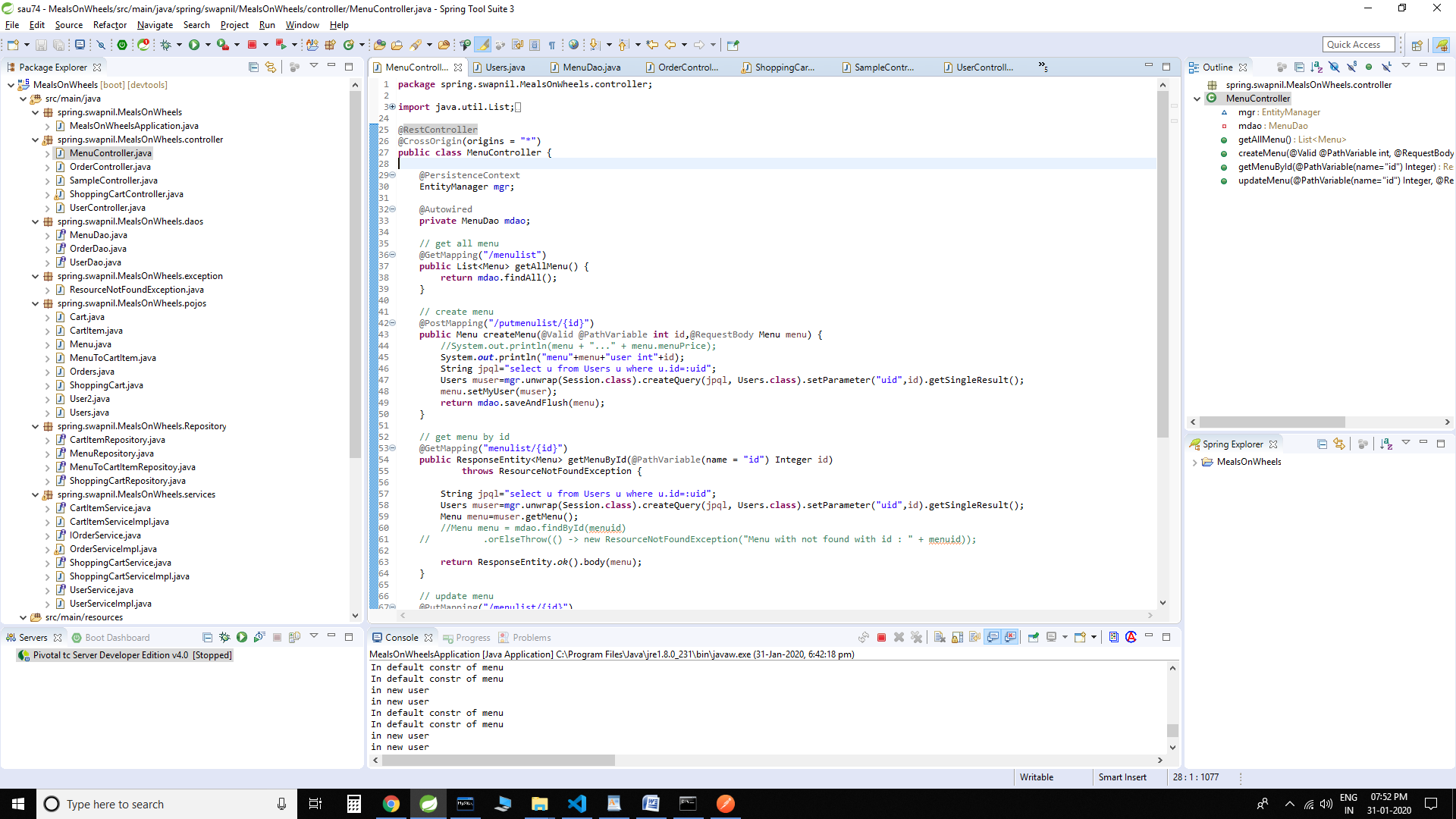


Postman testing Postman is a Google Chrome app for interacting with HTTP APIs. It presents you with a friendly GUI for constructing requests and reading responses. Postman is a great tool when trying to dissect RESTful APIs made by others or test ones you have made yourself. It offers a sleek user interface with which to make HTML requests, without the hassle of writing a bunch of code just to test an API's functionality.

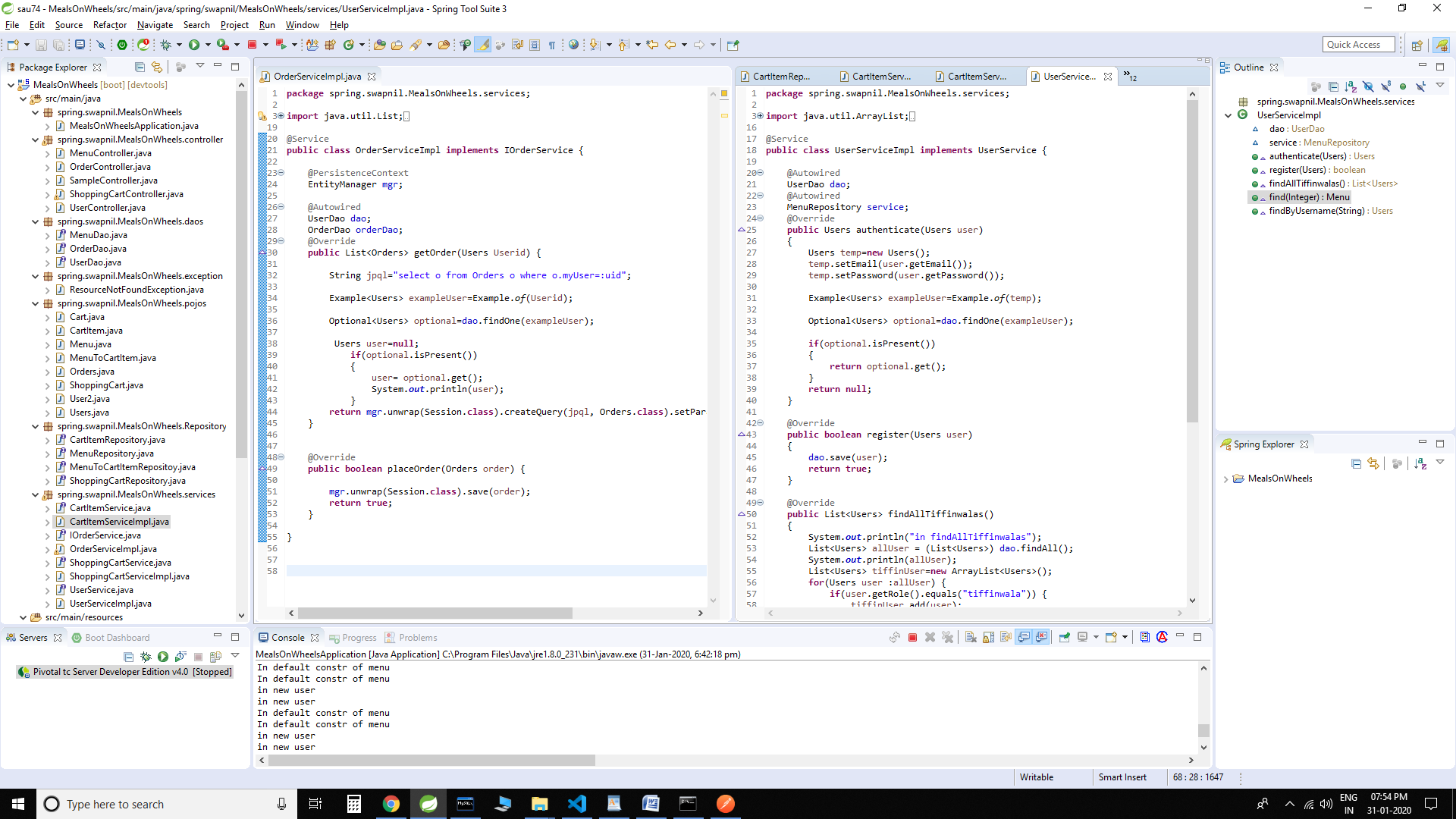


Java Code (using Spring Boot )

Controller



Service layer



Testing

To build up our project we used software testing process for executing a program with the intent of finding error that is uncovering errors in a program makes it a feasible task and also typing to find the errors (whose presence is assumed) in a program. As it is a destructive process.

Types of testing we use in our project

Here we just mentioned that how the testing is related to this software and in which way we have test the software? In our project we have used five types of testing this are listed below –

UNIT TESTING –

Unit testing where individual program units or object class are tested here by using this testing we have focus on testing functionality of the methods.

MODULE TESTINg –

Where this is the combination of unit program is called module. Here we tested unit program (5-6 programs ) is where the module program have dependency.

SUB SYSTEM TESTING –

Then we combined some module for the preliminary system testing in out project.

SYSTEM TESTING –

Where it Is combination of two or more sub system and then it is tested here we tested the entire system a per requirement.

ACEEPTANCE TESTING –

Normally this type of testing is done to verify if system meets the customer specified requirements. After submitting this project to the user then they tested and to determine whether to accept the application. It is the system of testing performed by the customer to determine where they should accept the delivery of system.

# CHAPTER 9

# 9. CONCLUSION AND FUTURE SCOPE:

* Enhance user interface by adding more user interactive features. Provide deals and promotional offer details to home page. Provide receipes of the week/day
* Allow customers to book Tiffin’s on monthly basis.
* Add different payment options such as PayTm, Cash, Gift Cards etc. Allow to save payment details for future use.
* Integrate GPS system in the application.
* Ways to analyze Tiffin service provider performance via customer rating.

Bibliography

<https://start.spring.io>/

<https://www.getpostman.com/>

<https://getbootstrap.com/>

<https://angular.io/>

[https://www.w3schools.com/](https://www.w3schools.com/js/)

