VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

VIT RESTAURANT REVIEWER SYSTEM

An J Component Report

submitted by

TEAM MEMBERS:

NAME	REG. NO.
SAYON PALIT	17BCE0157
ADITYA PRATAP SINGH	17BCE0810
SWATI SINGHVI	17BCE2037

in partial fulfilment for the award of the degree of

B. Tech

in

COMPUTER SCIENCE ENGINEERING

Under the guidance of

Faculty: Prof. Delhi Babu R

School of Computing Science and Engineering

OCTOBER 2018

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

DECLARATION

I hereby declare that the J Component report entitled "**VIT RESTAURANT REVIEWER SYSTEM**" submitted by me to Vellore Institute of Technology, Vellore-14 in partial fulfilment of the requirement for the award of the degree of **B.Tech** in **Computer science and Engineering** is a record of bonafide undertaken by me under the supervision of **Dr. R. Delhi Babu** I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

SAYON PALIT	ADITYA PRATAP SINGH	SWATI SINGHVI
17BCE0157	17BCE0810	17BCE2037

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	LIST OF FIGURES	5
1.	**INTRODUCTION**	5
	1.1 Abstract	6
	1.2 Requirements	6
	1.3 Website Components	7
	1.3.1 Front-End	
	1.3.2 Back-End	
	1.4 Schema	9
2.	**FUNCTIONALITY**	9
	2.0 Database implementation and server activation	
	2.1 Landing Page	
	2.2 Index Page	
	2.3 Registration Page	

	2.4 Login Page	
	2.5 View Page	
	2.7 Adding Restaurant Page	
	2.8 Adding Comment And Rating	
	2.9 Deleting/editing comment/restaurant	
3	**CONCLUSION**	14
	APPENDICES	14

LIST OF FIGURES

Figure 1: HTML code snippet

Figure 2: Front end and Back end tools

Figure 3: Schema Table

Figure 4: Initializing the server

Figure 5: Storage format of MongoDB

Figure 6: Landing Page

Figure 7: Index Page

Figure 8: Registration Page

Figure 9: Login Page

Figure 10: View Page

Figure 11: Adding Restaurant Page

Figure 12: Adding Comment and Rating

Figure 13: Deleting/editing comment/restaurant

1. INTRODUCTION

India is one of the fastest growing economies. As development increases day by day, so does the number of internet users in the country. With an omnipresent technology like the internet, the daily lives of people are becoming more and more digitized. Whether it is e-commerce websites like Amazon and Flipkart or social media platforms like Facebook, a lot of our daily tasks like shopping and networking are being done digitally. This means that businesses who are present on the internet have an inherent advantage over their competitors who are not using the internet to promote their products. Similarly, whenever a person wishes to buy a new product, or try out a new restaurant or hotel, they try to gather feedback and reviews about the commodity.

1.1 ABSTRACT

This project intends to provide an easy-to-use, aesthetically pleasing and secure platform to let people share reviews and experiences from various restaurants in their cities. Many such apps are already present like Zomato and Yelp, but those focus mostly on restaurants in big cities present in the United States and Europe.

Our website has all the functionalities of these apps and some more useful features such as geolocation of the restaurants. To facilitate the information provided to the user, a well-planned database has been designed which enables users to register and login to the website. In addition, registered users can also add restaurants to the database with various details like name , location, price and even images. Other registered users can comment on their experiences in the restaurants

present in the database and give a star rating out of 5. There are added features for editing or deleting the restaurants as well as comments, but only the database administrator or the author of that comment/restaurant can delete or modify it.

1.2 REQUIREMENTS

Software Requirements:

- Sublime Text
- MongoDB 5.0 and above
- Windows 7.0 and above or MACOS 10.0 and above
- Any web browser (Google Chrome/Internet Explorer/Mozilla Firefox)

Libraries & Frameworks

- Bootstrap
- NPM
- Mongoose
- Google Developers API

1.3 WEBSITE COMPONENTS

1.3.1 FRONT-END

We have used Node.JS and embedded JavaScript in the backend .So the frontend code is basic HTML styled with CSS and the Bootstrap template and the web-pages are made more dynamic using jQuery for DOM manipulation.

```
<html>
    <head>
        <title>VIT FOOD REVIEWER</title>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
        <link rel="stylesheet" href="/stylesheets/landing.css">
        <script src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.min.js" type="text/javascript"
        async></script>
    </head>
    <body>

    <div class="container">
        <% if(error && error.length > 0){ %>
            <div class="alert alert-danger" role="alert">
                <%= error %>
            </div>
        <% } %>
        <% if(success && success.length > 0){ %>
            <div class="alert alert-success" role="alert">
                <%= success %>
            </div>
        <% } %>
    </div>

    <div id="landing-header">
        <h1>Welcome to VIT Food Reviewer!</h1>
        <a href="/campgrounds" class="btn btn-lg btn-success">View All Restaurants</a>
    </div>

    <ul class="slideshow">
```

Fig 1 shows a code snippet that is plain HTML with embedded JS (code enclosed in these '<%' ,'%>' tags).These are used to make the website more interactive and responsive. Content on the website is arranged by with the help of the grid system by Bootstrap which divides the horizontal width of the screen into 12 individual columns.

1.3.2 BACK-END

The Javascript framework Node.JS has been used to produce the back-end.The routes in the website have been based on the RESTful standards for routing so that other developers can understand the website functionality easily for future improvements.

The requests of the users are handled in following way:

1.User requests for some webpage.

2.Request is sent to node.js server.

3.Server dynamically creates a webpage using data items from the database and a mixture of HTML, CSS and JavaScript.

4.Then this webpage is sent to the user.

The database used is MongoDB, an upcoming NoSQL database which was used because of its accompanying JavaScript library Mongoose which helped with the integration with rest of the tools. Also, in MongoDB many features such as JSON viewer are present which help in easy visualization of data

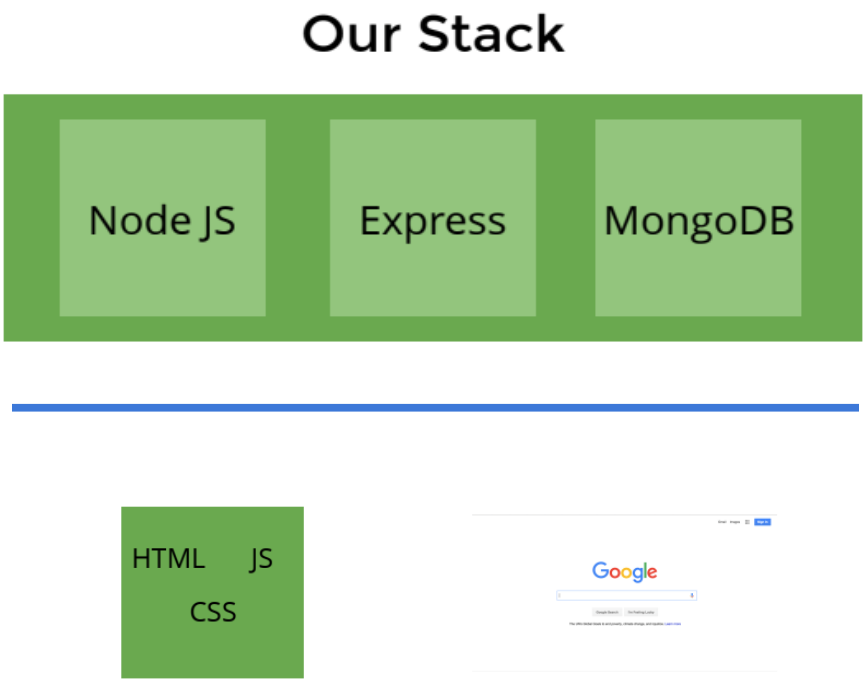


Fig2: The frontend and backend tools are separated by blue line.

1.4 SCHEMA

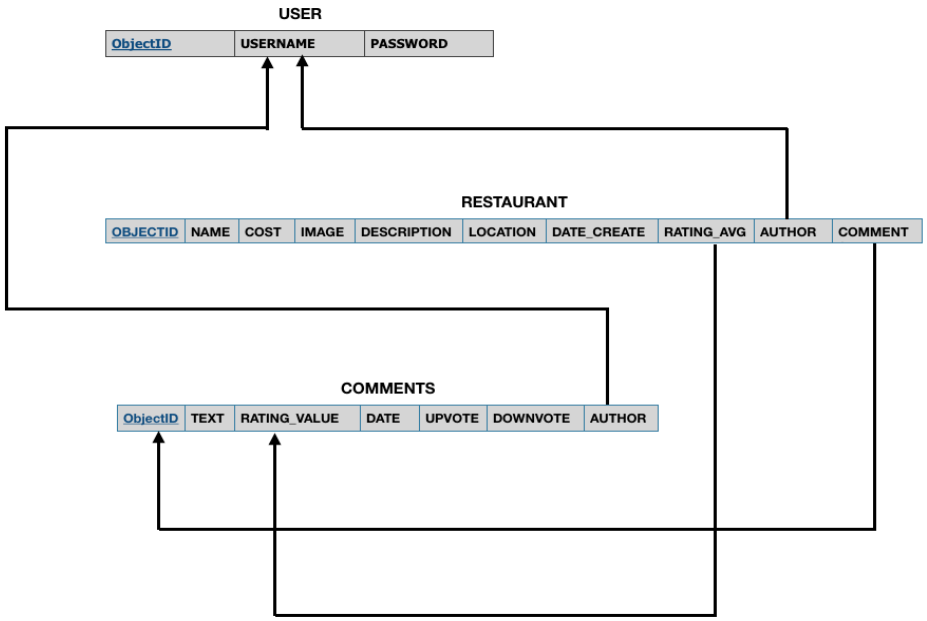
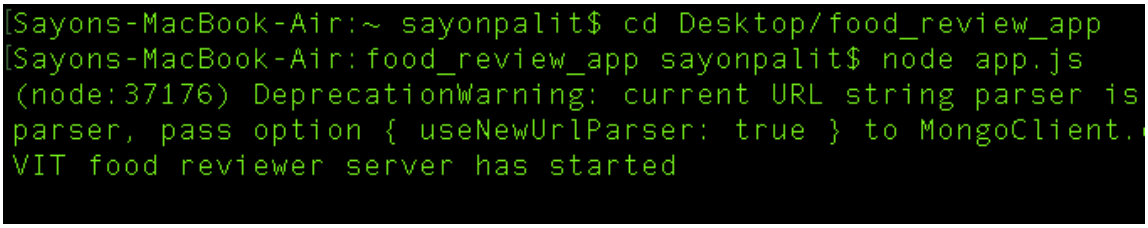


fig3 shows the schema table

2. FUNCTIONALITY

The functionalities of the project from start to finish are listed along with their implementation.

2.0 DATABASE IMPLEMENTATION AND SERVER ACTIVATION



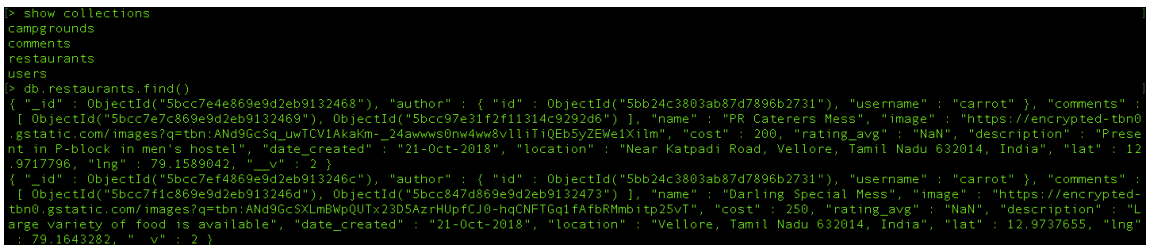


Fig4 and 5 show the initializing of the server and also the storage format in the MongoDB database. Using commands show dbs and show collections and then db. collection_name. find() we can see all the items of that collection.

2.1 LANDING PAGE

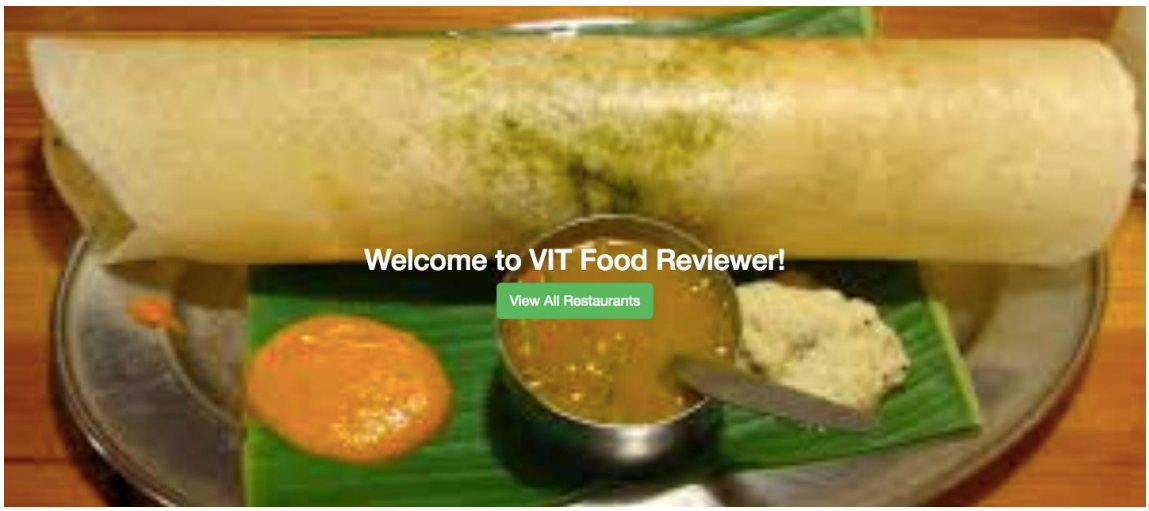


fig 6 shows the landing page which is a dynamic slideshow of food images.

2.2INDEX PAGE

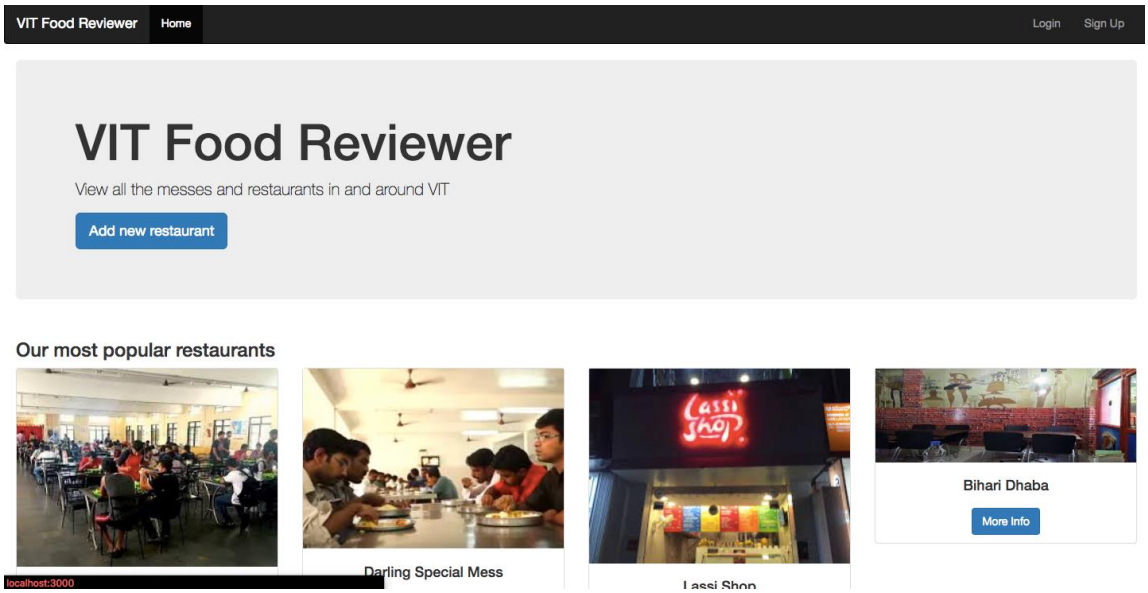


fig 7 shows the index page of the website where user can see all the restaurants in the database.

2.3 REGISTRATION PAGE

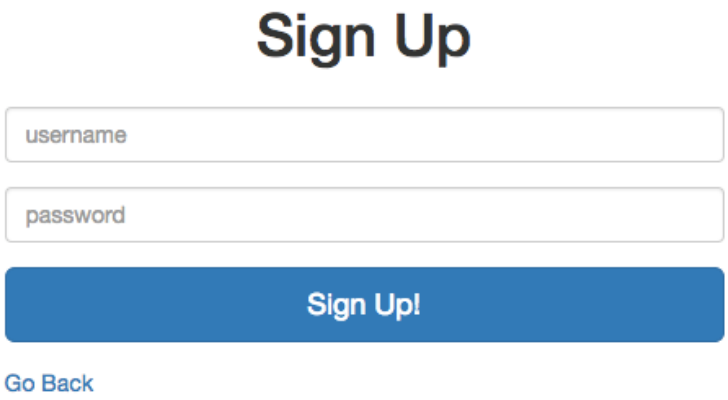


Fig8 lets the user signup to the website after they enter username and password which gets stored in database in users table

2.4 LOGIN PAGE

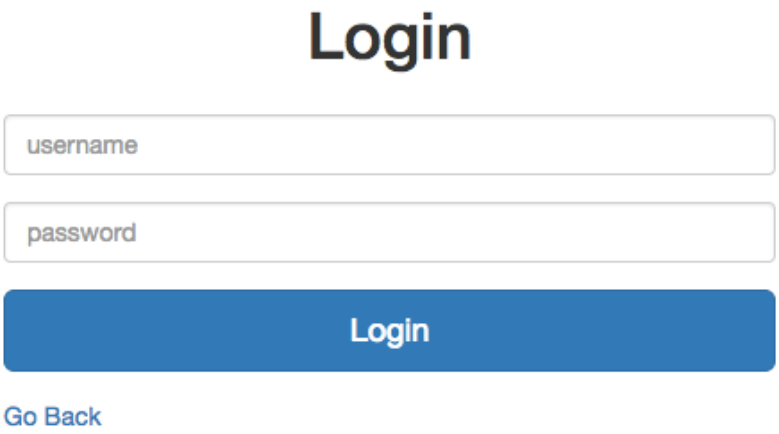


Fig9 lets the user slogin to the website after they enter existing username and password which gets matched in database in users table.

2.5 VIEW PAGE

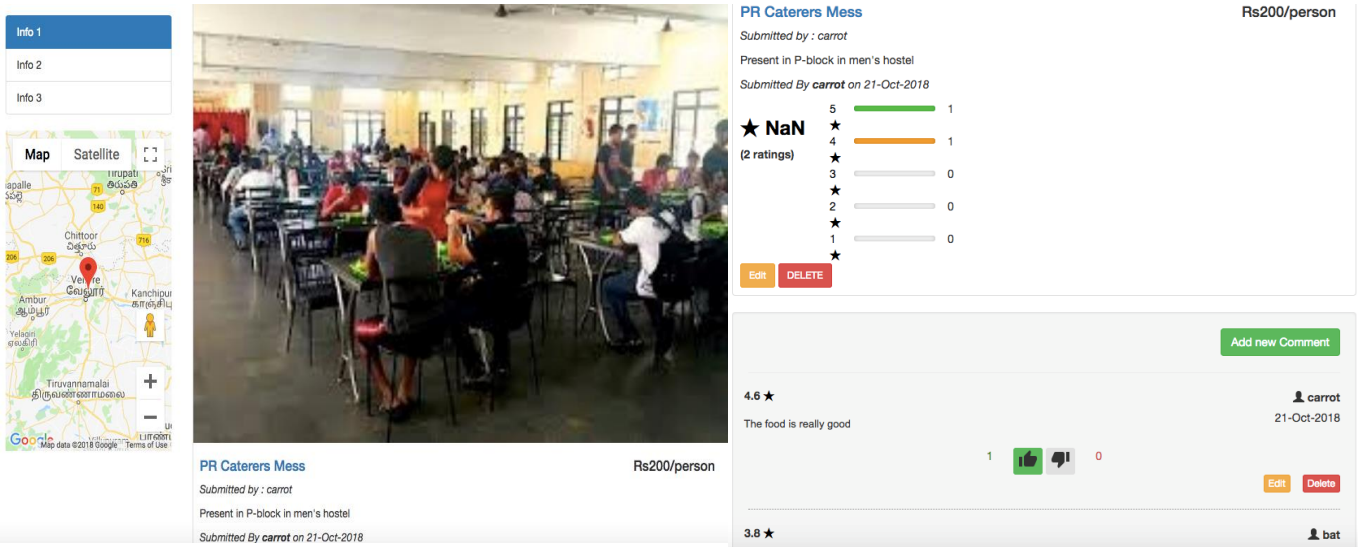


Fig10 shows the view page where one can see more details about the restaurant and add comments and ratings and also see the location of the restaurant.

2.6 ADDING RESTAURANT

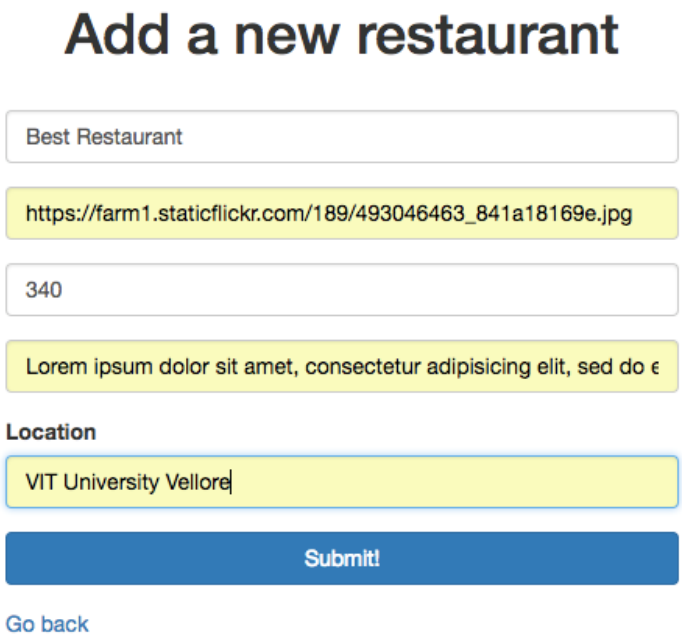


fig 11 shows the add restaurant page which takes in all the data from the user and stores it in the restaurants table.

2.7 ADDING COMMENT AND RATING

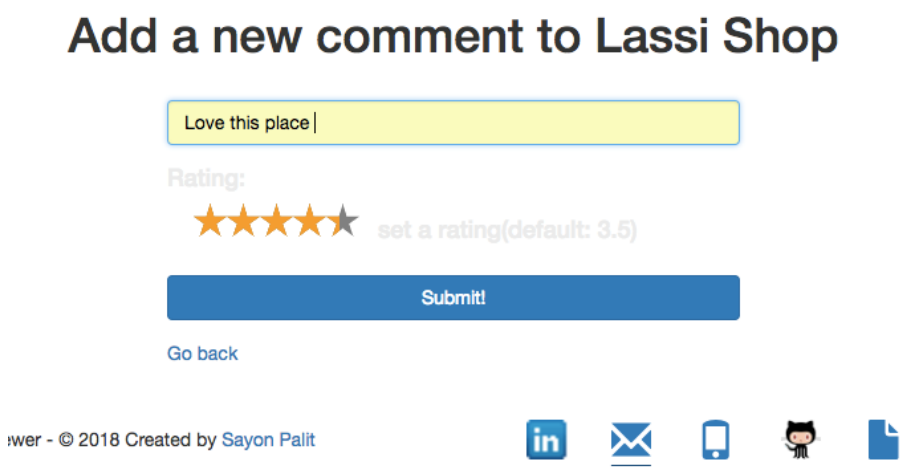


fig 12 shows the add new comment and ratings page and here users can write their views of a restaurant and give it a rating.

2.8 DELETING/EDITING COMMENT/RESTAURANT

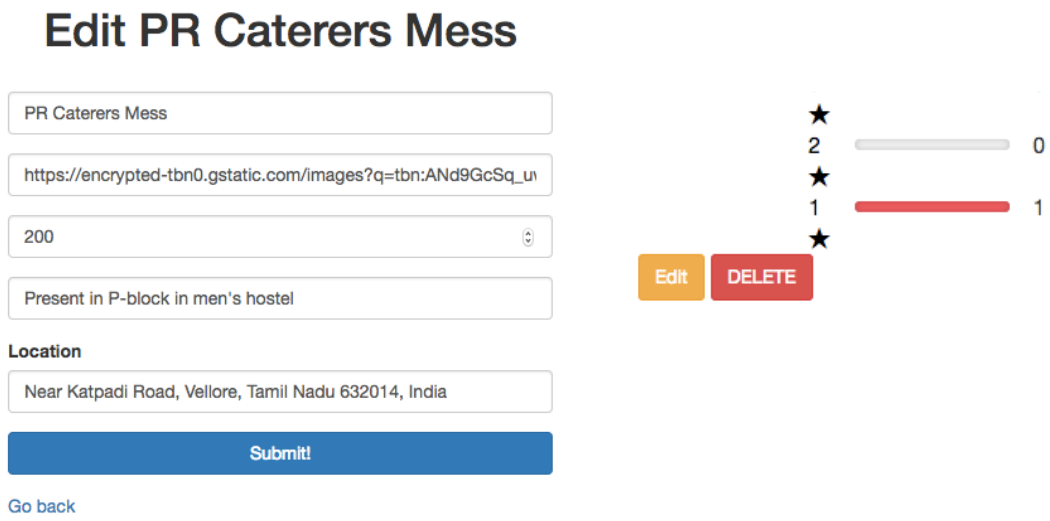


fig13 shows the edit and delete buttons which allow the user to manipulate the restaurant and comment data. But these buttons are only visible to the creator of that particular comment or restaurant.

3. CONCLUSION

This project deals with the primary goal of reviewing the restaurants and the user comments and details getting stored in a database. Adding a new restaurant feature also directly gets linked to the database. This way the admin has access to the critiques. Through backend and frontend, the site is both functional and attractive. As time evolves it is imperative to make everything digitized yet convenient and this project deals exactly with that. The future improvisation could be adding the “Online food ordering feature” or also the “Look for places on the basis of cuisine/proximity”. These are just certain advancements other than which the project is completely functional.

APPENDICES

MongoDB – MongoDB is an open source database management system that uses a document-oriented database model which supports various forms of data. It is used in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model. Instead of using tables and rows as in relational databases, the MongoDB architecture is made up of collections and documents.

Bootstrap - Bootstrap is a free and open source front end development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS, and JavaScript to facilitate the development of responsive, mobile-first sites and apps. Bootstrap includes user interface components, layouts and JS tools along with the framework for implementation. The software is available precompiled or as source code.

Mongoose DB - Mongoose is a MongoDB object modeling tool designed to work in an asynchronous environment.