**CASE STUDY**

MEAN Stack – Project

**FOOD COURT***An Ecommerce Website*

*Single Page web Application*

Project Report

*Submitted By*

Gautami Mehta

*Under the guidance of*

Mr. Naga

**CONTENT**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Topic | Page |
| 1 | Introduction | 4 |
| 2 | Technology Stack | 5 |
| 3 | End To End Diagram | 6 |
| 4 | **Architecture Explanation** | 8 |
| 5 | **Flow Diagram** | 9 |
| 6 | **Project Explanation** | 10 |
| 7 | **Application Features** | 12 |
| 8 | **Website Images** | 13 |
| 9 | **Postman Testing** | 18 |
| 10 | **Conclusion** | 21 |
| 11 | **References** | 21 |

**ACKNOWLEDGEMENT**

This report would not have been possible without the essential and gracious support of **Mr. Naga** sir. His willingness to motivate me contributed tremendously to our report. I also would like to thank him for showing us some examples related to the topic. He encouraged me even when we had any sort of obstacles or difficulties and guided us through it. It couldn’t be possible without him. He inspired us and gave thoughtful ideas and innovative hints which paid dividends to my project.

Gautami Mehta

**INTRODUCTION**

The aim of the project was to build a responsive online application for restaurant which helps customers to order foods online. One of big problem while ordering online is trust. Social media websites play a vital role to attract customers and make trust on their mind.

Hence, Tokenization, Authorization and Authentication is integrated to the application. So, customers trust the quality and attract to order online.

To address these functionalities, a proper technology stack and versions have been chosen on the earlier works for automating the food ordering processes.

Food Court is built using the following

* Front-End - Angular – A free open-source web application framework.
* Back-End - Node Js – An opensource, backend JavaScript environment.

MongoDB – A source available for database program.

* Postman – Used for API testing.

This application will help restaurants may be large or small to optimise and have complete control over their business and customers. This application helps the restaurants to do all functionalities more accurately and faster way. It reduces the manual works and improves the efficiency of restaurants. The software helps food orders to maintain day to day records in the system. It’s very useful in keeping a proper record of the database.

**TECHNOLOGY STACK**

Front-End - Angular – A free open-source web application framework.

Back-End - Node Js – An opensource, backend JavaScript environment.

MongoDB – A source available for database program.

Postman – Used for API testing.

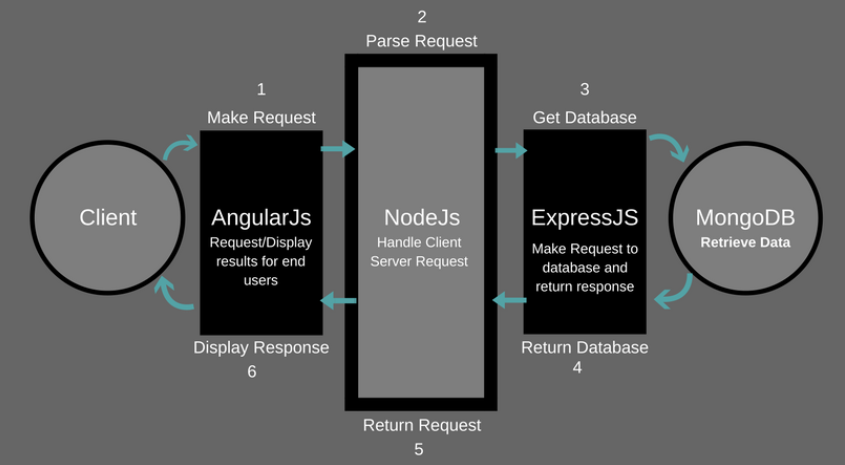
|  |  |  |
| --- | --- | --- |
| **Technology** | **Versions** | **Description** |
| angular/cli | 13.1.3 | The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications directly from a command shell. |
| node | 17.3.1 | Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. |
| mongoose | 6.1.10 | Mongoose is a library designed especially for node. js and won't work in the browser environment. In order to be able to connect the db - you need to establish node. js back-end server, for example express. js. |
| express | 4.17.2 | Express.js, or simply Express, is a back-end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. |
| jsonwebtoken | 8.5.1 | JSON Web Token is a proposed Internet standard for creating data with optional signature and/or optional encryption whose payload holds JSON that asserts some number of claims. The tokens are signed either using a private secret or a public/private key. |
| bootstrap | 5.1.3 | Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. |

**END TO END DIAGRAM**

The term MEAN is a free and open-source stack that refers to a collection of JavaScript-based technologies used to develop web applications. It can be also stated as an acquisition of JavaScript-based advances used to create website and web applications.

MEAN is an acronym for MongoDB (Data system), Express.JS (backend framework), Angular.JS (frontend framework) and Node.js (backend runtime environment). It is [full stack JavaScript](https://www.brainvire.com/mean-stack/). It uses a single coding language i.e., JavaScript.

The idea behind it is to solve the common issues with connecting those frameworks, build a robust framework to support daily development needs, and help developers use better practices while working with popular JavaScript components.



# MEAN Stack

**M= MongoDB** is a free and open-source cross-platform document-oriented database program. It is Classified as a NoSQL database program.

**E = Express.js**, a framework which supports and is used to host Node.js projects. It is a JavaScript web application framework used to build dynamic websites with efficient features.

**A = Angular.js**, yet another framework for building apps. It is used to provide features like the two-way data binding. It also provides maintenance, reusability, and testability.

**N = Node.js**, it is a runtime environment, which runs server-side web applications, i.e., it works on the back-end, away from the user’s eyes to fetch relevant data or perform operations on the same. It helps in building profoundly adaptable, flexible and concurrent applications in quick time.

**Back-end with Node.js**

Node.js is built for handling asynchronous I/O while JavaScript has an event loop built-in for the client-side. This makes Node.js fast compared to other environments. However, the event-driven/call back approach makes Node.js difficult to learn and debug.

Node.js includes modules such as Mongoose, which is a MongoDB object modelling, and the Express web application framework. Through Node modules, abstraction can be achieved, which reduces the overall complexity of the MEAN stack.

## Back-end with Express Framework

Express is a minimalist and unopinionated application framework for Node.js. It is a layer on top of Node.js that is feature-rich for web and mobile development without hiding any Node.js functionality.

## Front-end with Angular

Angular is a web development platform built in TypeScript that provides developers with robust tools for creating the client side of web applications.

It allows development of single-page web applications where content changes dynamically based on user behaviour and preferences. It features dependency injection to ensure that whenever a component is changed, other components related to it will be changed automatically.

## Database with MongoDB

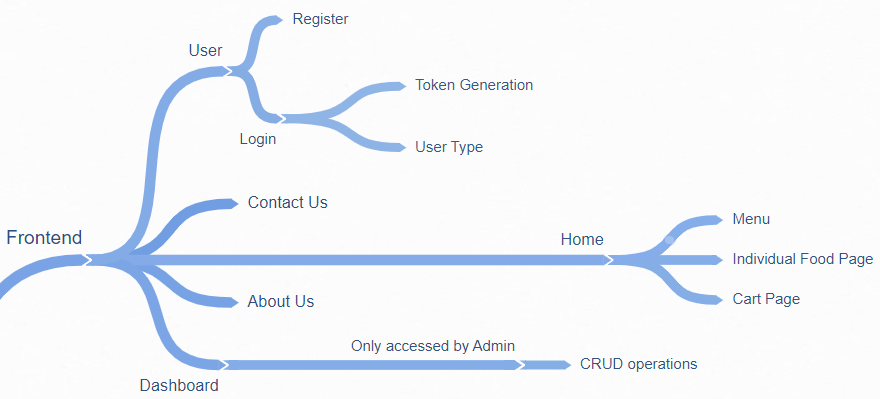
MongoDB is a NoSQL database which stores data in BJSON (Binary JavaScript Object Notation).

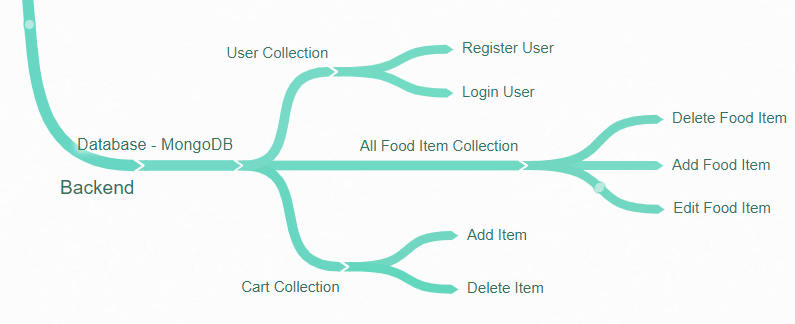
MongoDB became the de facto standard database for Node.js applications to fulfil the "JavaScript everywhere" paradigm using JSON (JavaScript Object Notation) to transmit data across different tiers (front-end, back-end, and the database).

**Architecture Explanation.**

* When a client makes any request, then it is firstly processed by the Angular.JS. It is a client-side JavaScript framework. It is a library written in JavaScript.
* After that, the Request from angular.js enters in the second section which is Node.JS. It is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side.
* After that Request enters in the third section that is Express. Js. Then the Express.js makes the request to the database i.e., MongoDB. After that MongoDB recovers the data and returns the response to the Express. Js.
* Then Express. Js returns the response received from Express. Js to the Node.JS and then Node.JS return it to the Angular.JS to display the result. This is the working of MEAN Stack architecture.

**FLOW DIAGRAM**

**Front-end Chart**

**Back-end Chart**

**PROJECT EXPLANATION**

Food Court is a **single page** web application build using **MEAN Stack**. The website lands up with a login page where the registered users can login and view the site where as for the users who are new to the site can register themselves. All the user data is stored with **security and authentication** in the mongo DB database.

The login form and the registration form both have their individual field **validations** by using Reactive forms, Form Group and Form control modules. The registered email id can’t be re-registered again as the backend takes care no email id is duplicated.

With login a **token** using json web token module and user type are generated depending on **roles like user or admin**. And each role users have different functionalities and attributes.

Once the user has logged in, he will enter the home page done using **lazy routing**. Here the user will see all the food items displayed. Along with the food items he can even see the description like the price, ratings and Origin and other different tags.

According to the preference of the user he can add products to the cart, change the quantity, visit the home page again and add more projects. Once the cart has some food items present, the user can proceed with payment option using **Google** **Payment Method** which is implemented in the frontend as backend required some valid user card details.

The user can also visit the About Us page to know more about the site and the Contact Us page which gives him a privilege of contacting us and sending us their feedbacks and remarks.

According to the role-based authentication the admin can do **CRUD** operations on the food items. Like he can add food item and their details like price, tags, origin also he can **upload image** in the backend. Along with adding he can even edit the existing food item and make changes according to his preference. Deleting option is also available for the admin, depending on his choice.

All the food items are being displayed in a table using **angular material** and some of the other features like **pagination** and **filtering** is also applied for displaying the data in a sorted and a clear manner.

The front end and back end are connected using the **mongoose** with mongo DB. **Express** and other modules have also been used to make the connection. Express is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. Different **collections** are made in the database like user database, product database and the cart database.

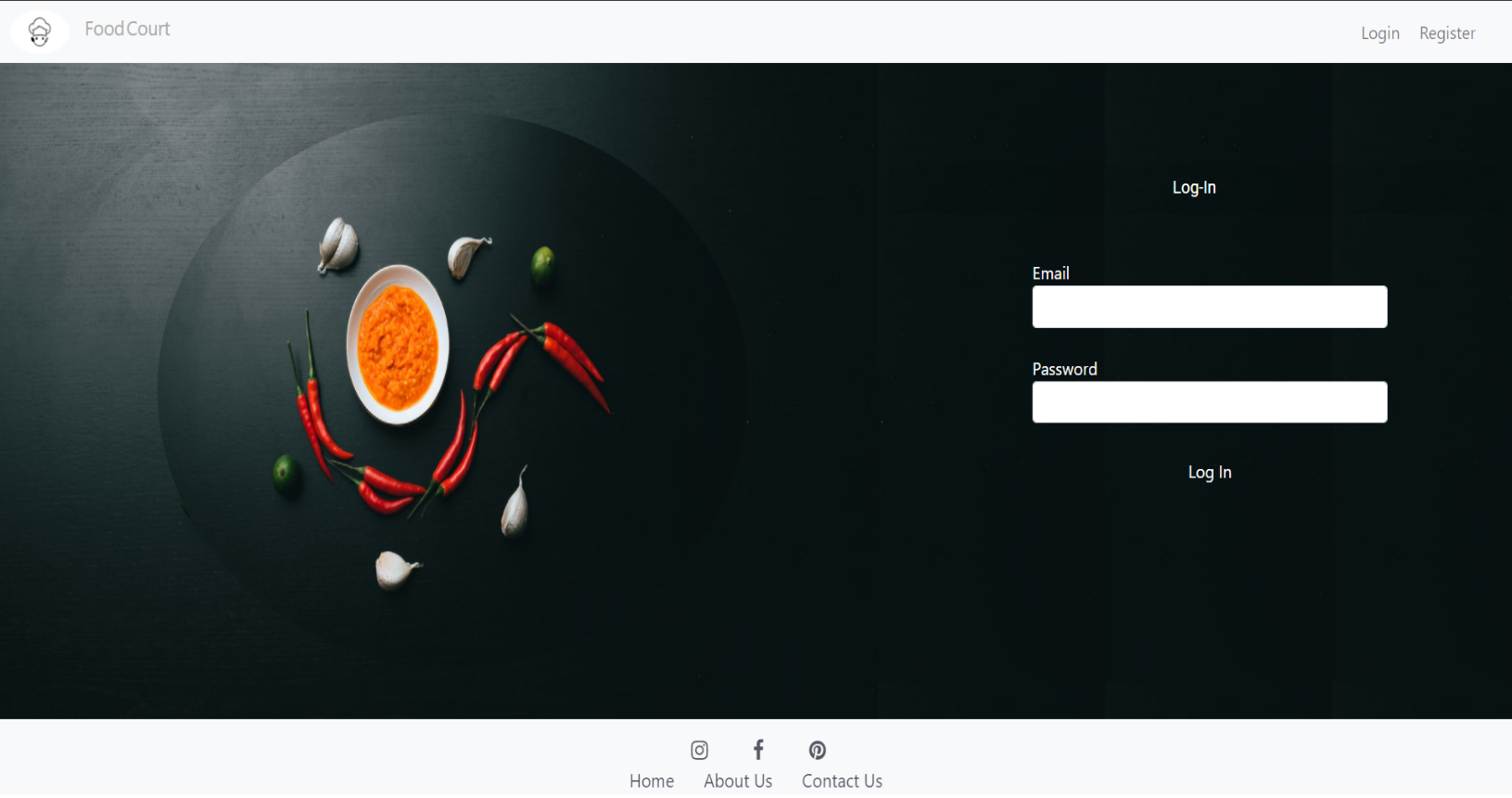
API – **Application Programming Interface** –are tested using postman software and all the request like get, post, delete and push are tested. And further in the document screen shorts have been attached for the overview.

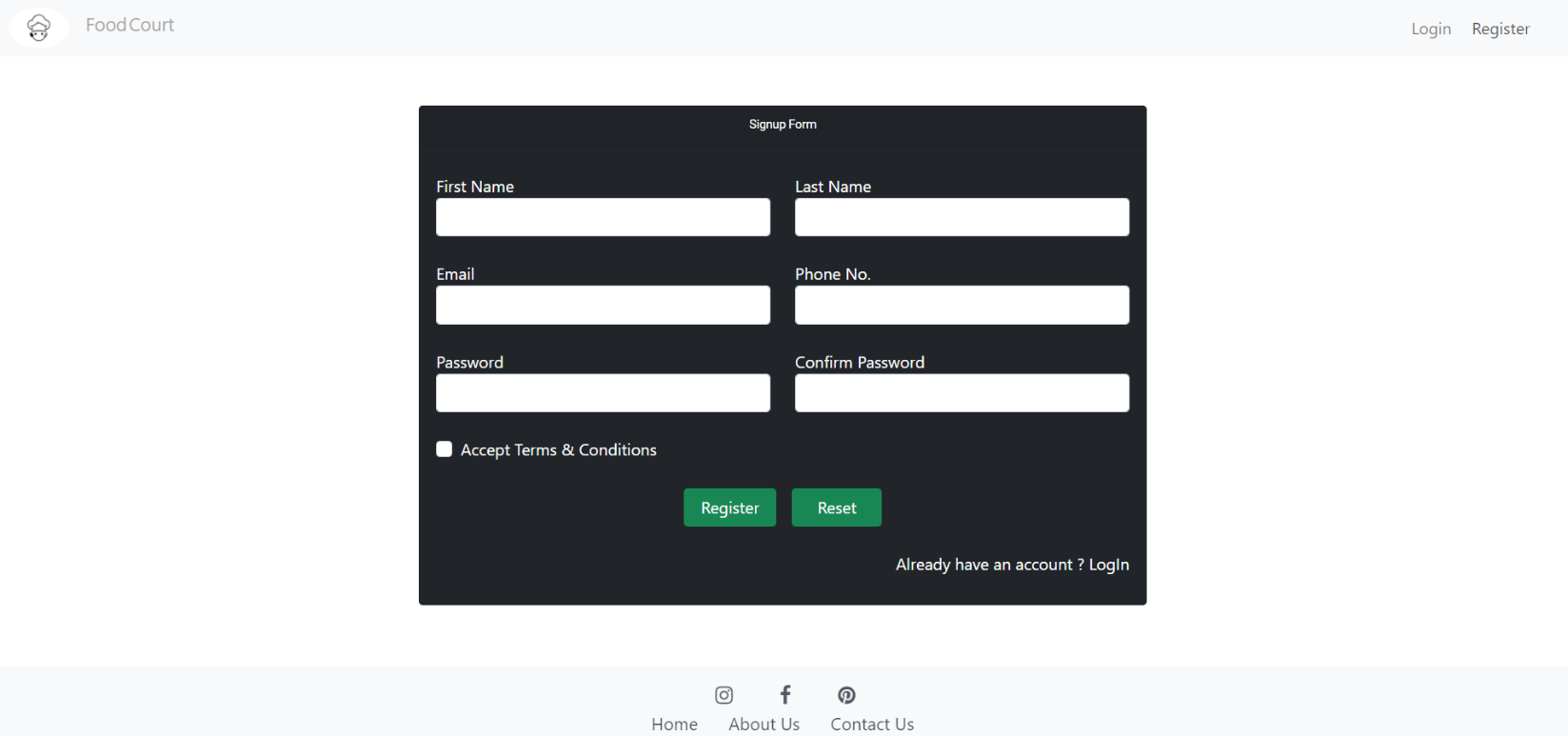
**FEATURES OF THE APPLICATION**

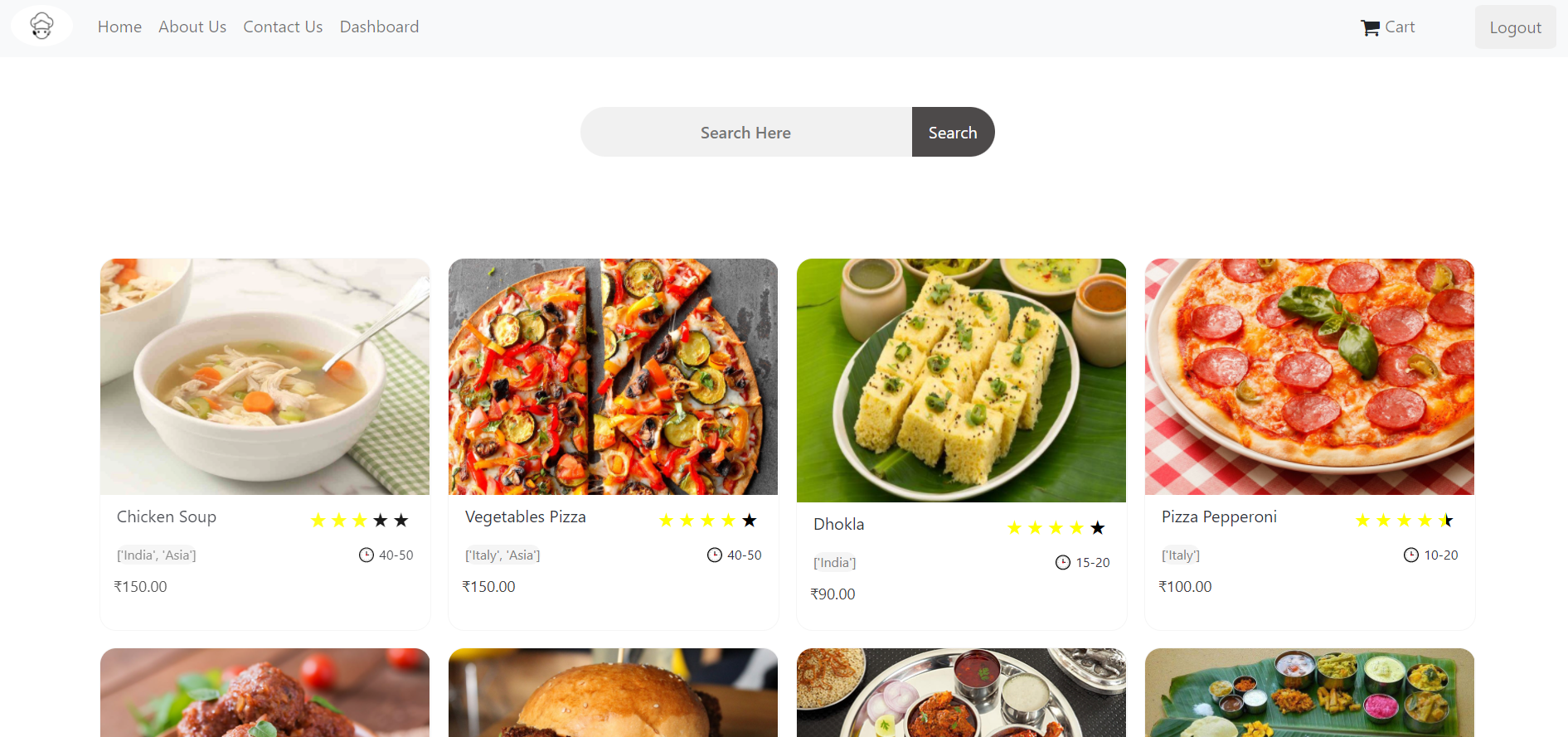
Some of the features of the Single Web application are listed below:

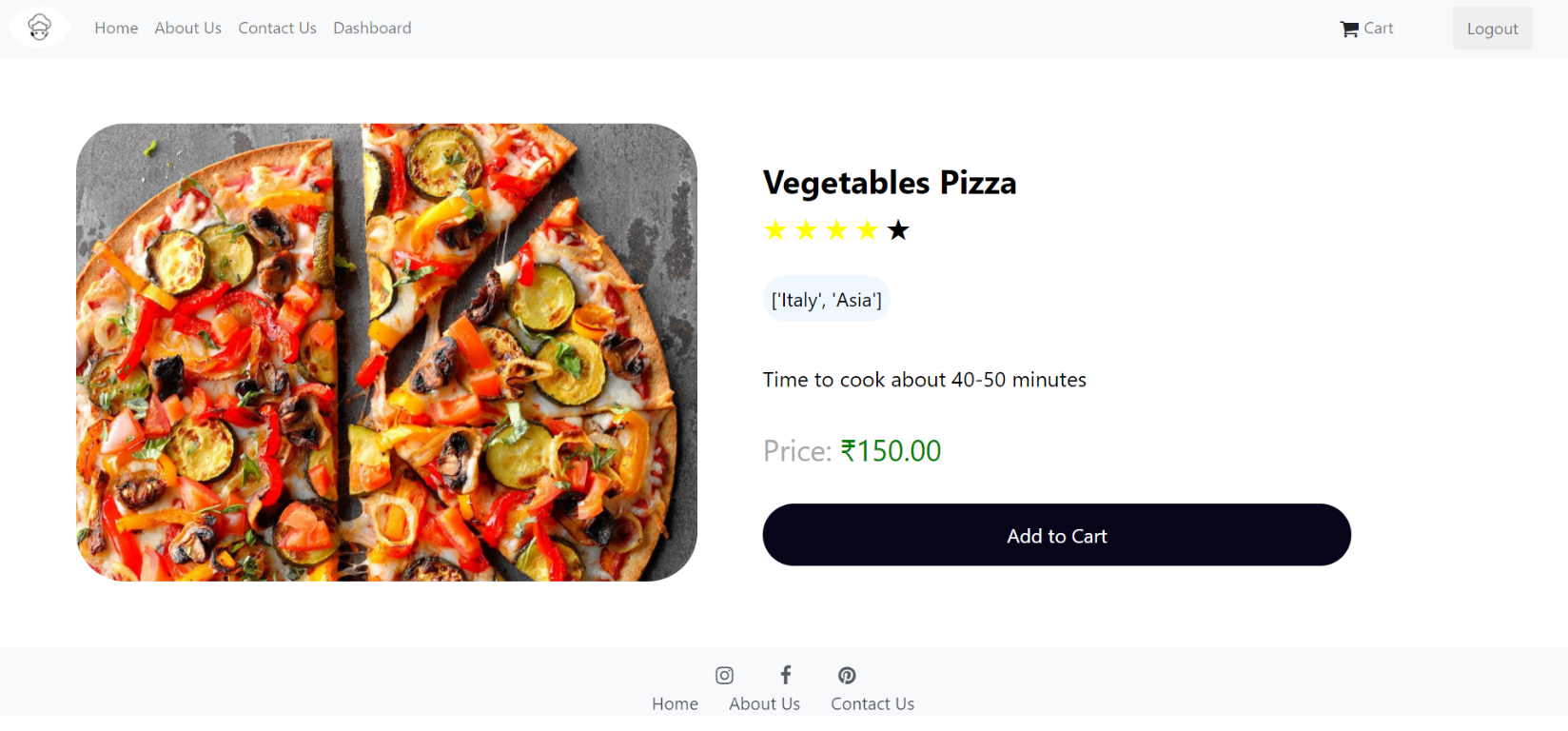
* Login based on user Registration using backend database.
* Token generation and Role based functionalities.
* Form Validations for all fields separately.
* Home Page For displaying items
* Search bar for searching of particular food items.
* Add to cart option for each food item.
* Option for variable quantity of a particular product and calculation of total price based on products in cart.
* Fake Google payment method is implemented.
* Option for adding, editing and deleting food items – CRUD operations - depending on user role.
* Image upload in backend database.
* About Us and Contact Us page for making site more user friendly.

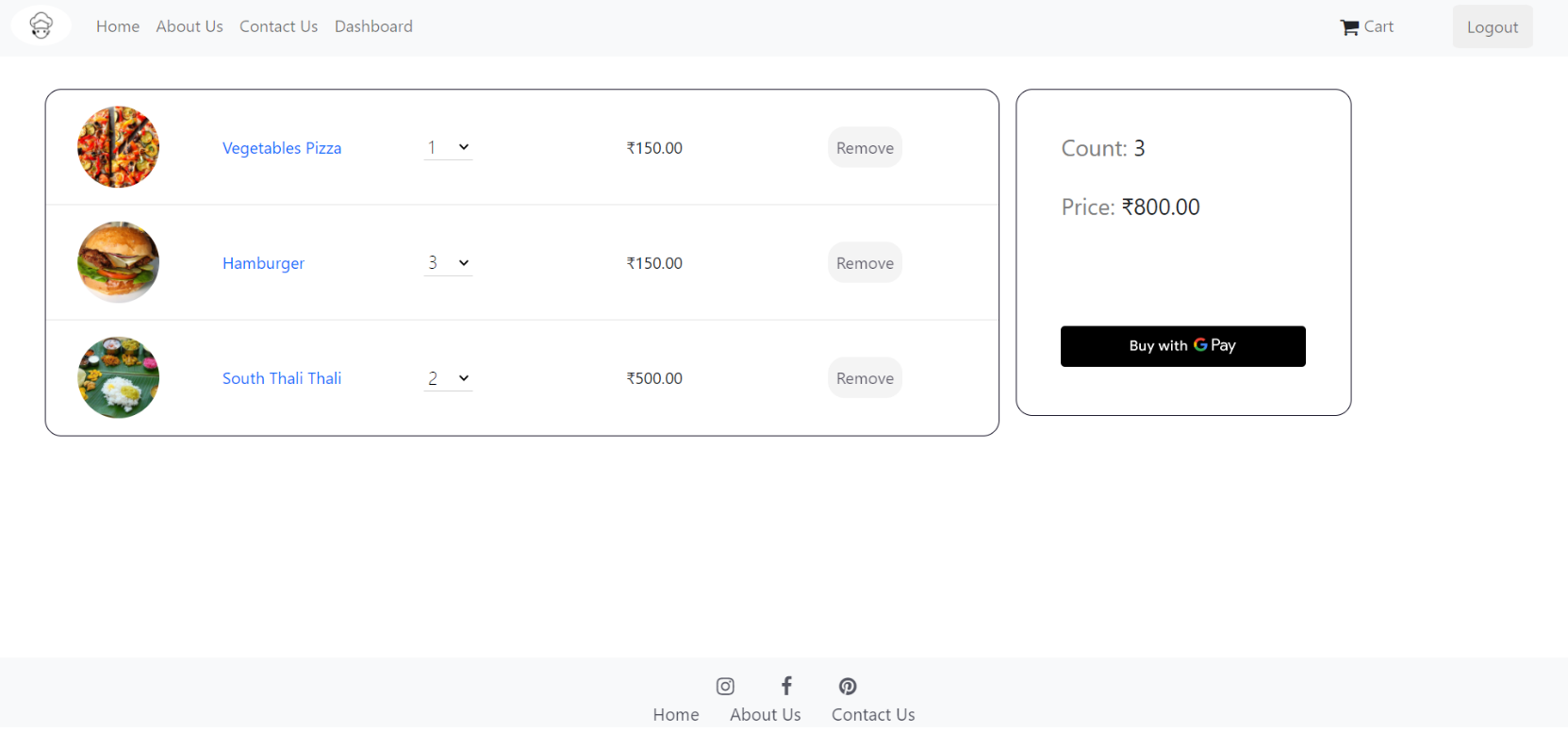
**WEBSITE IMAGES**

**Login page**

**Registration page**

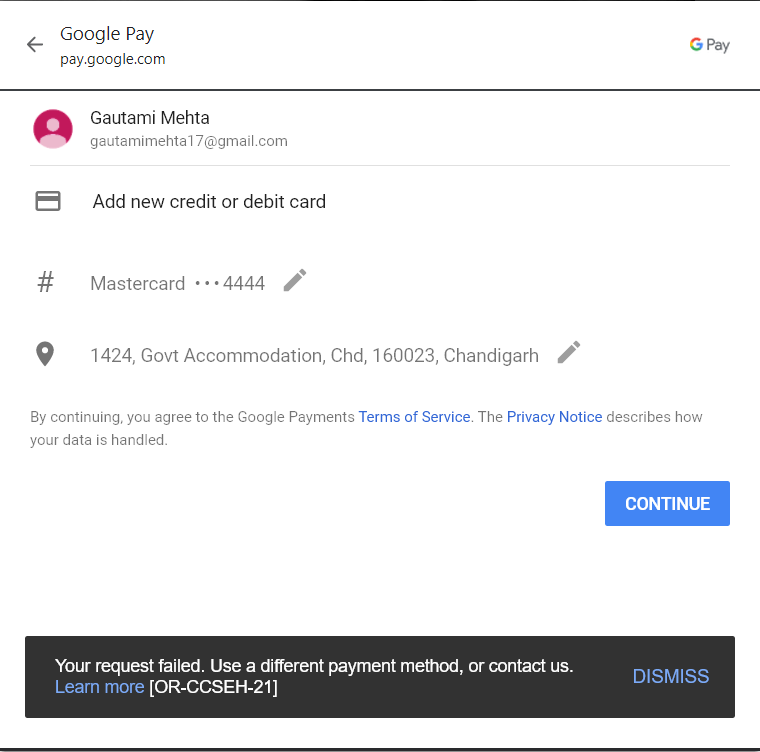
**Home Page**

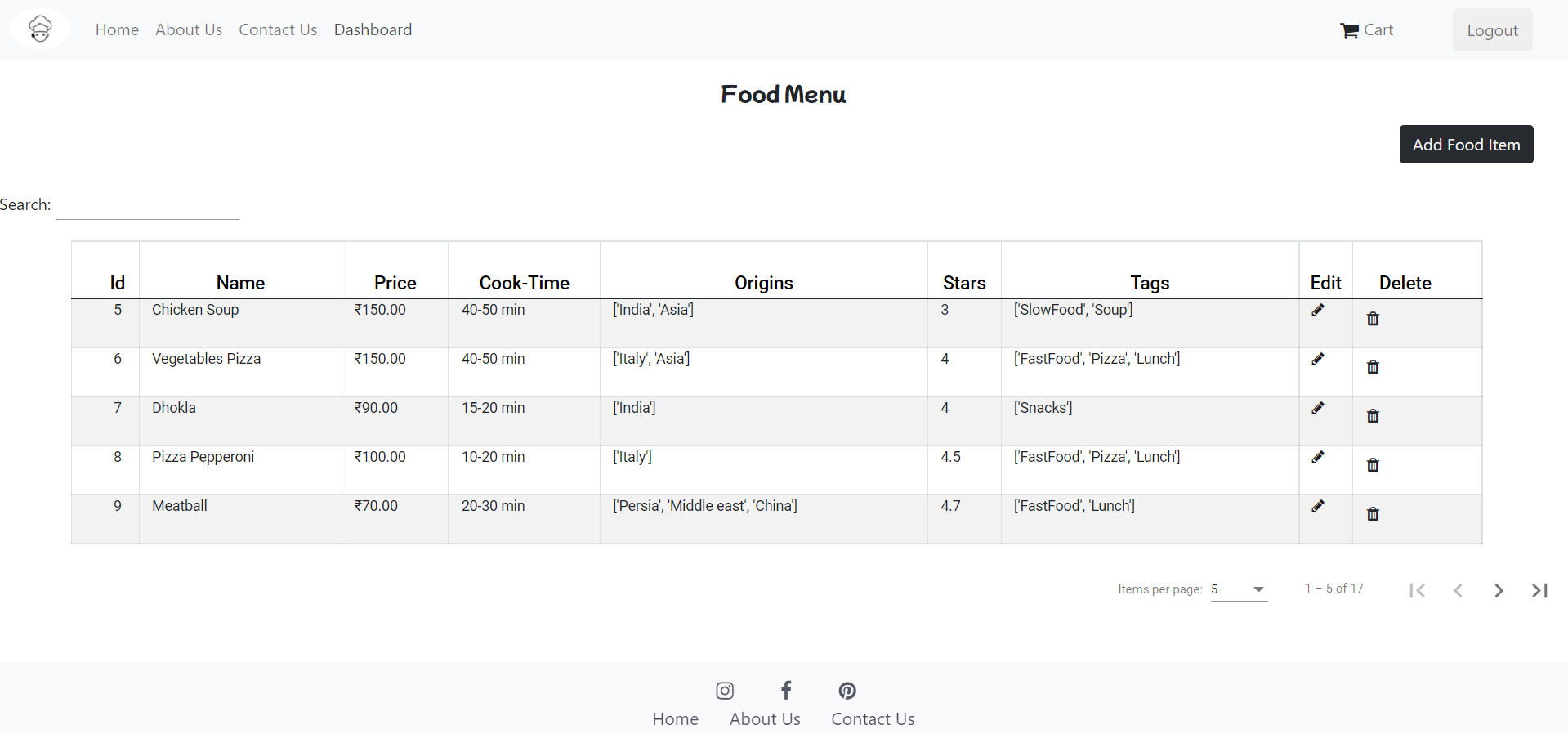
**Food Item Page**

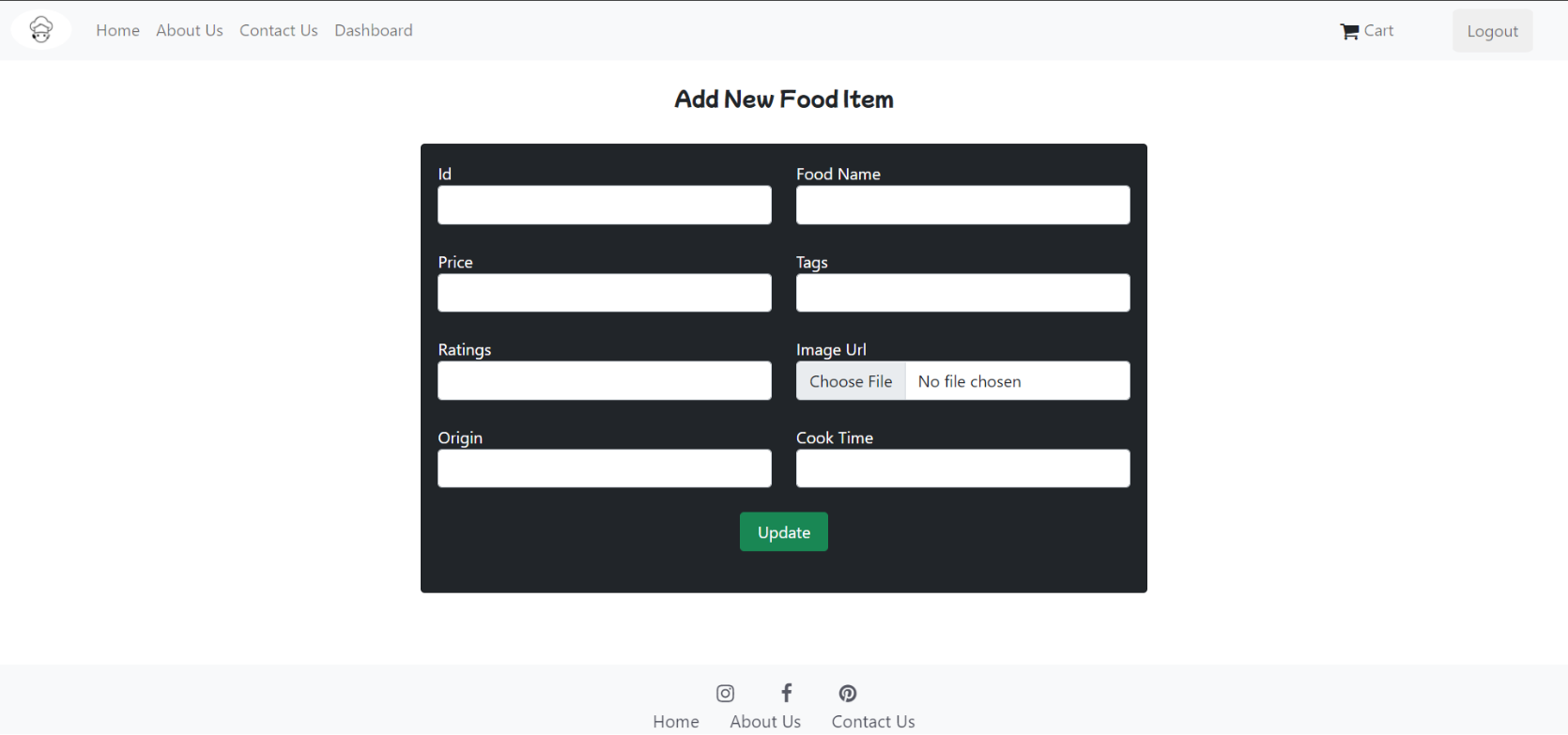
**Cart Page**

**Google Payment New Window With dummy data**

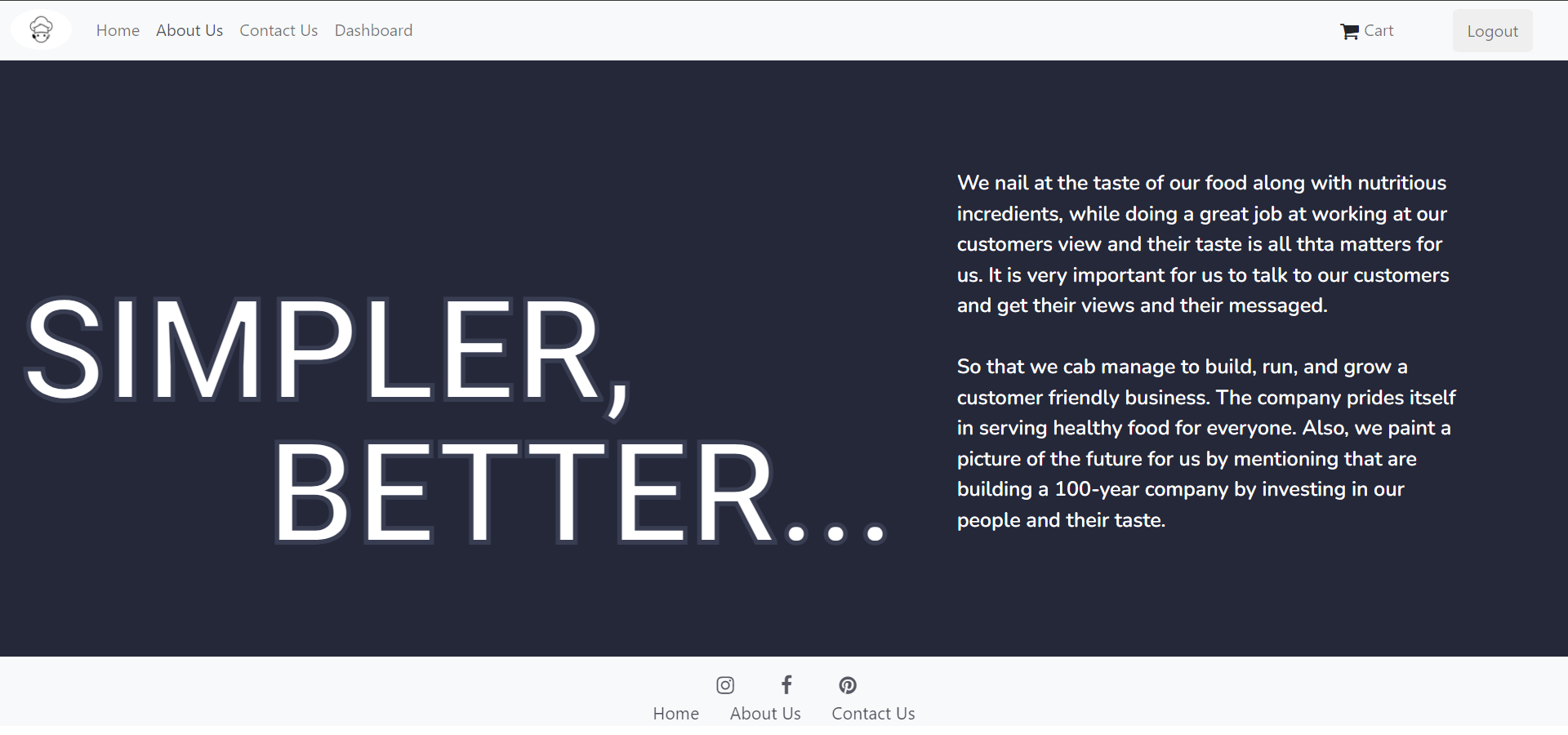




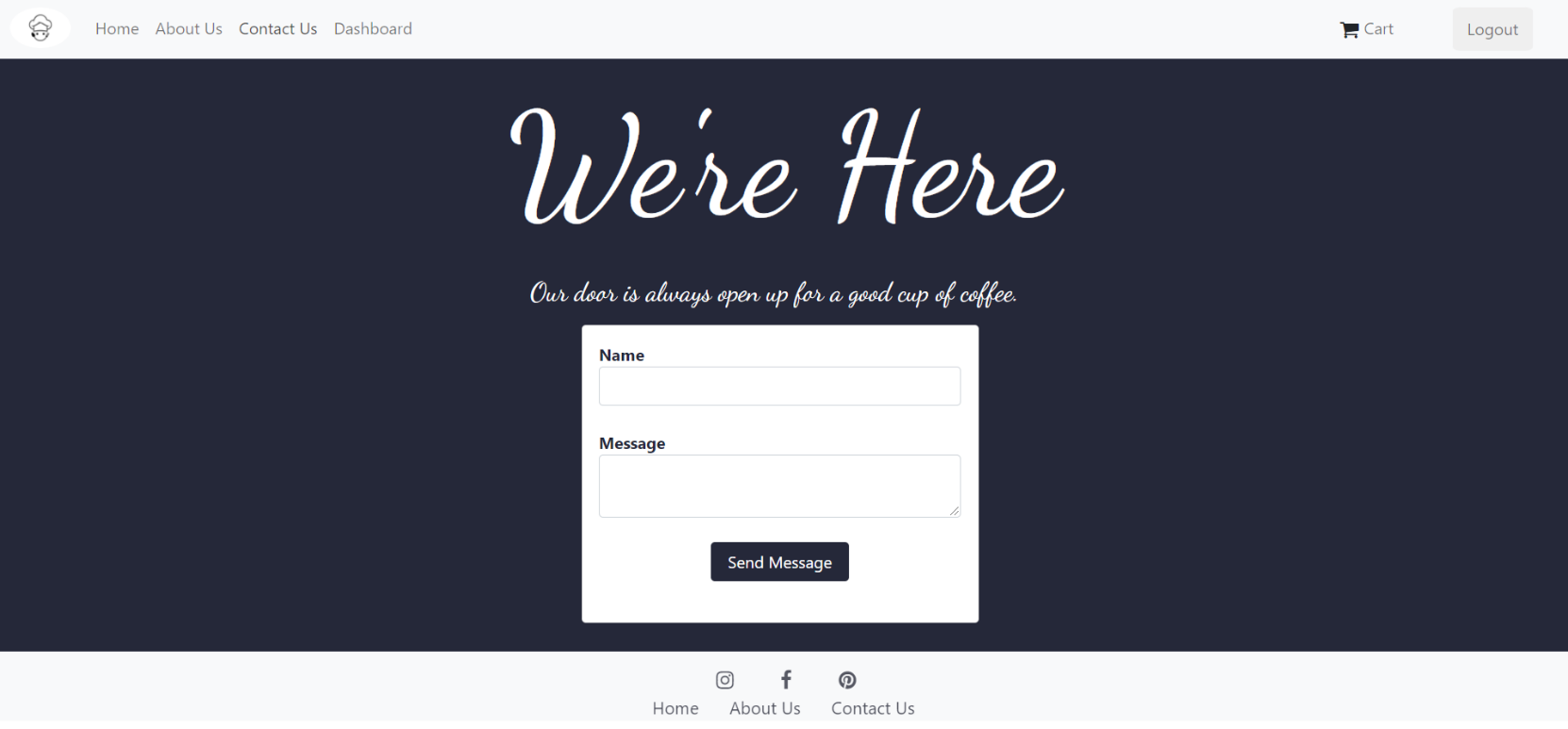
**Dashboard Page for Admin**

**Add new Food item**

**About Us Page**

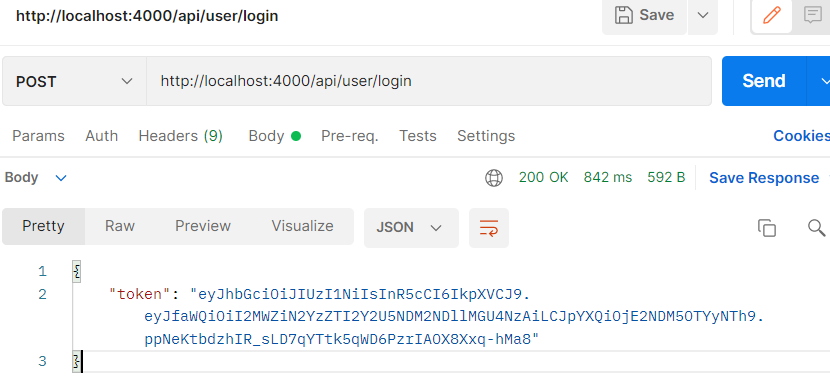


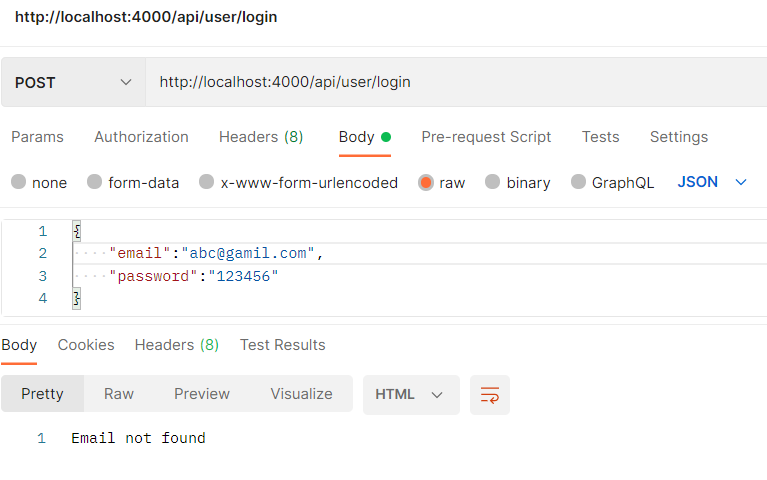
**Contact Us Page**

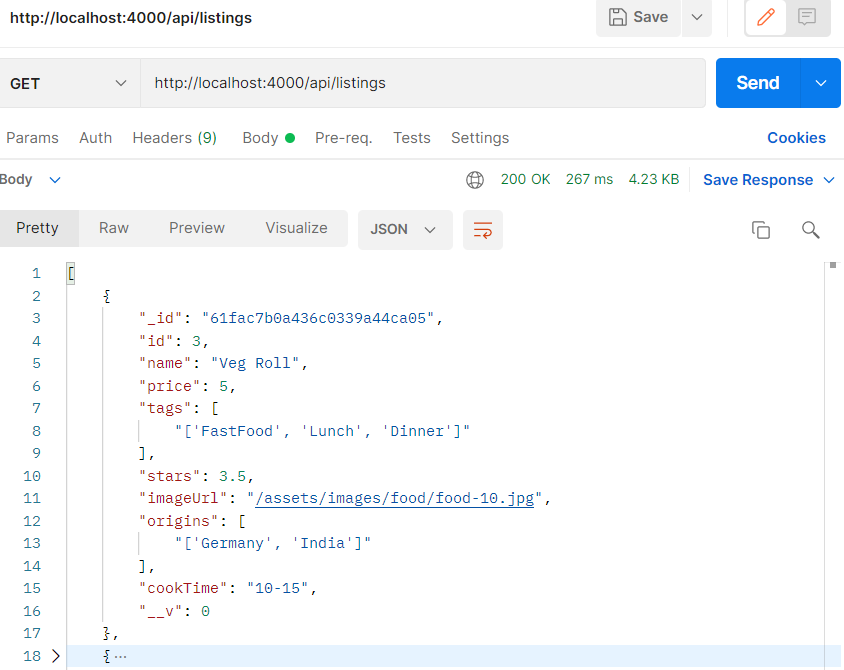


**POSTMAN TESTING**

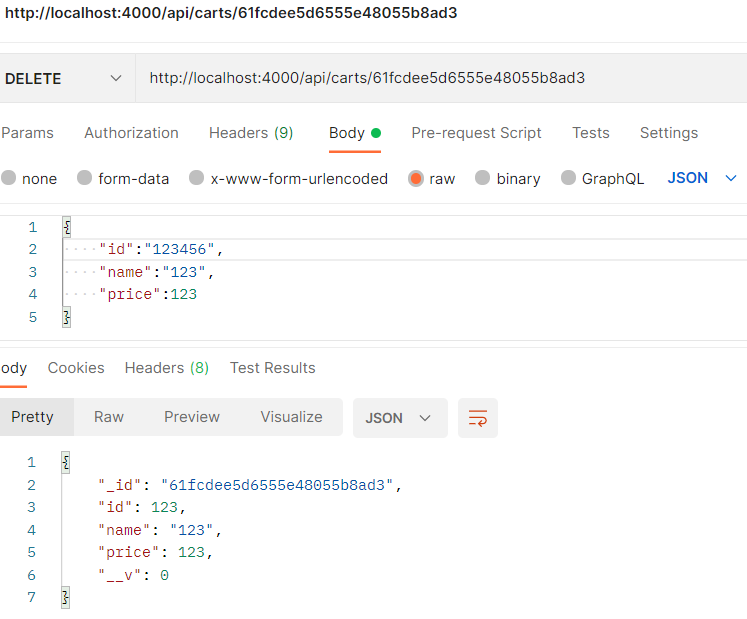
**Token generation**

****

**Only registered email ids can be used for login**

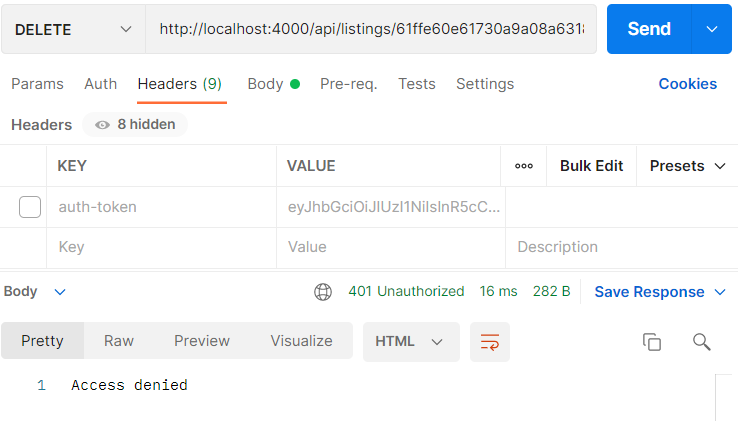
**All Product**

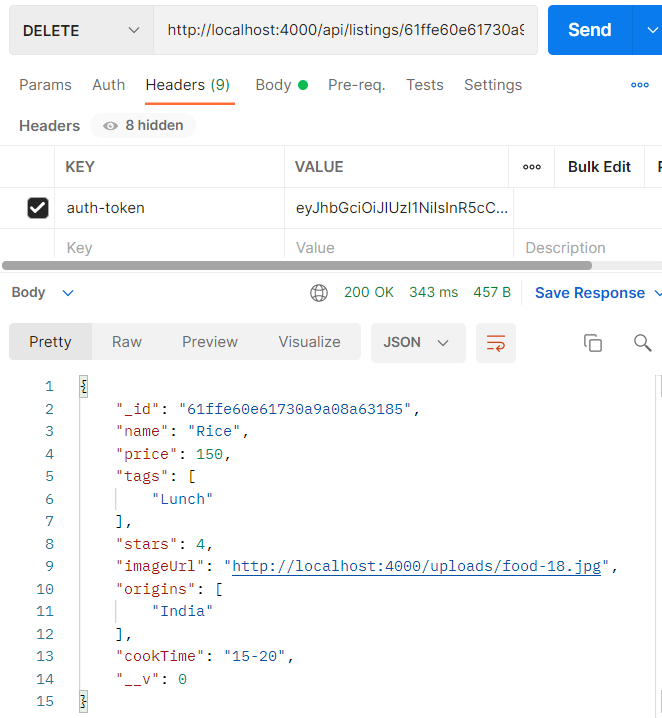
**Adding Products to Cart**

****

**Deleting product**

Access is denied because the token is not added



Token is added and product is deleted

**CONCLUSION**

This application will help restaurants may be large or small to optimise and have complete control over their business and customers. This application may help the restaurants to do all functionalities more accurately and faster way. It reduces the manual works and improves the efficiency of restaurants. The software helps food orders to maintain day to day records in the system. It’s very useful in keeping a proper record of the database.

**REFERENCES**

* <https://angular.io/docs>
* <https://docs.npmjs.com/>
* <https://docs.mongodb.com/>
* <https://www.w3schools.com/angular/angular_tables.asp>
* <https://www.section.io/engineering-education/nodejs-mongoosejs-mongodb/>
* <https://www.mongodb.com/blog/post/quick-start-nodejs-mongodb-how-to-get-connected-to-your-database>
* https://youtu.be/WfCJ3sHnLBM