



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

# **RESTAURANT MANAGEMENT SYSTEM**

*submitted by*

**Navya RG (17BCE2416)**  
**Alisha Shah (17BCE0930)**

**Report submitted for the  
Final Project Review of**

**Course Code: CSE3002 – Internet and Web Programming**  
**Slot: E1**  
**Lab: L37 + L38**

**Professor: Archana T**

## **Contents:**

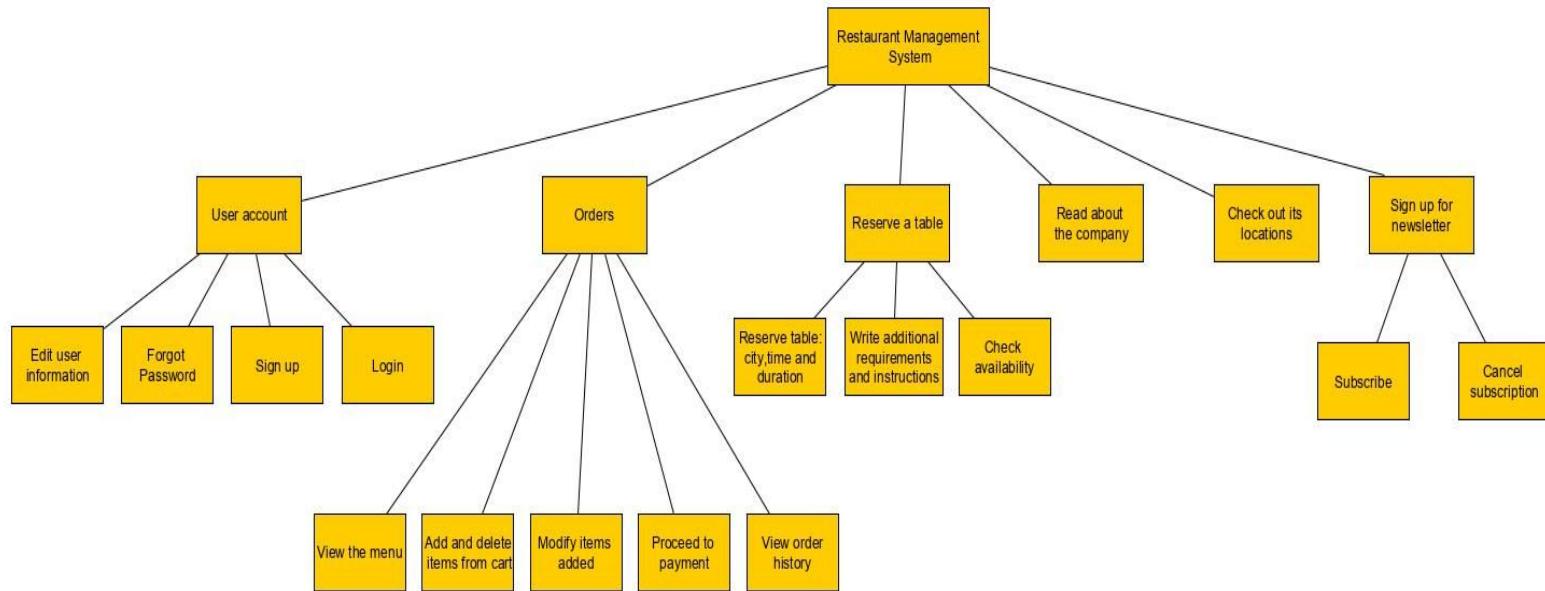
|                                     |           |
|-------------------------------------|-----------|
| <b>Introduction</b>                 | <b>3</b>  |
| <b>Work breakdown structure</b>     | <b>4</b>  |
| <b>Tools used for project</b>       | <b>4</b>  |
| <b>Database and schema</b>          | <b>7</b>  |
| <b>Screenshots of working model</b> | <b>12</b> |
| <b>Code snippets</b>                | <b>28</b> |
| <b>Conclusion</b>                   | <b>32</b> |

## **1. Introduction**

Our restaurant management system, The Golden Spoon is a web application which advertises the restaurant and also allows ordering food online. This website plays a major role in connecting the customers and service providers. In our case, the service provider is the restaurant. Customers are provided with the best services as they are allowed to order food at their own feasible situation. To avail all the features of the website, the customer has to sign up which is a simple 2 minute process. Signing up is made mandatory to avoid fake orders. The static pages that can be accessed by users without logging in are the homepage, about us, locations and menu. The homepage contains general information about the restaurant, about us contains details and history of the restaurant and locations contains the details about the 9 branches of the restaurant. The user can subscribe for newsletters before logging in. This functionality is also available after logging in. The menu page contains a list of sumptuous food items belonging to 5 categories- Soups & Salads, Appetizers, Main course, Beverages and Desserts along with a description of each item. The user can add items to the cart from this list only after logging in. These items can also be deleted from the cart making it easy for the customer to place an order with just a few clicks. After adding items to the cart, the customer can proceed to the payment process after the completion of which the items will be removed from the cart and can be viewed in the order history. After the completion of this process, the user is alerted with a message which says order is successfully placed. The customer can also edit his profile details like name, address, password etc. An additional feature incorporated in our project is the table reservation functionality. The user can also book a table at the restaurant according to their suitable time and number of people by checking the availability and vacancy.

This system minimizes the unnecessary time taken to go all the way to the physical location of the restaurant and increases convenience for the stakeholders. Our project aims to help customers order online and reserve a table according to their desires by delivering the above mentioned facilities. With this system, we are endeavouring to achieve easy ordering and reservation from home itself.

## 2. Work breakdown structure



## 3. Tools used for project

- Node.js:

Node.js is a server side scripting language which is used for programming the back end of our Restaurant Management System. It is an asynchronous scripting language which is based on javascript and built on Google Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js is light-weight and efficient as it uses a non-blocking, event-driven I/O model. It is well suited for applications that run on a cluster of distributed devices and real-time data-intensive applications.

Node.js packages used:

- Express:

There are many web application development frameworks available. One such framework is Express. It is a flexible and minimal Node.js web application development framework. It provides a robust set of features to develop mobile and web apps. Node based web applications can be developed rapidly using express.

- EJS:

One of the templating engines used by Node.js is EJS. It stands for Embedded Javascript Templating. A templating engine has two main functions- helps by providing the ability to use minimal code to create an HTML template and produce the final HTML code by dynamically injecting data into the HTML template at the client side. In our project, it is used for sending data from the backend (server) and accessing it in the front-end (client's browser). For example, it is used for displaying the menu which is fetched from the database.

- **Mongoose:**  
Mongoose is an ODM (Object Data Modeling) library for Node.js and MongoDB. Mongoose translates between the code objects and how these objects are represented in the database by providing schema validation and managing relationships between data. Mongoose also simplifies mongoDB commands and makes it developer friendly.
- **Nodemailer:**  
Nodemailer is a package used for sending mails with ease. It is licensed under the MIT license. Nodemailer is used for sending mails to users upon sign up and upon newsletter subscription.
- **Cookie-parser:**  
Cookies have various functionalities. They can be used for maintaining sessions and adding user-specific features in web apps. For parsing these cookies, a module called cookie-parser from npm acts as the middleware.
- **Express-session:**  
Express-session is the package used for creating and maintaining sessions in Node.js. In our project, sessions are used to enable users to login and enable certain functionalities like placing an order or reserving a table only after login.
- **Nodemon:**  
When any changes are made in the file, it can be automatically detected by nodemon. Nodemon makes node.js app development easier by automatically restarting the server when these changes are detected. Nodemon is a replacement wrapper for node , to use nodemon replace the word node on the command line when executing your script with nodemon. It can be used when the developer does not want to keep restarting the server whenever he makes a change in the code.
- **MongoDB:**  
MongoDB is a noSQL database which is based on documents and collections. It makes handling data in the database easier than using SQL. MongoDB Compass is a user interface for MongoDB. Traditionally, mongoDB has to be accessed from the command prompt, which is not very user friendly. MongoDB Compass allows the use of a mouse to click and avoid typing mongoDB commands.  
In our project, MongoDB is used as the database for storing the user details, orders, menu etc.
- **HTML:**  
HTML stands for Hypertext Markup Language. It is used for structuring the webpage and add content.
- **CSS:**  
CSS stands for Cascading Style Sheets. It is used to style the HTML pages by avoiding cluttered styling attributes within HTML codes.
- **JavaScript:**  
It is a client side scripting language used to perform arithmetic, logical and many more operations on the clients browser. Using DOM, HTML elements can be

accessed in JavaScript and can be changed dynamically. Without javascript, web pages cannot be dynamic. In our project, javascript is used for form validation, displaying the users name on the top right corner when he/she is logged in, to display items in cart, to display order history, to trigger events onclick etc.

- Sublime Text 3:

It is a text editor which was used in our project to develop the entire application. It supports HTML, CSS, JavaScript, Node.js etc.

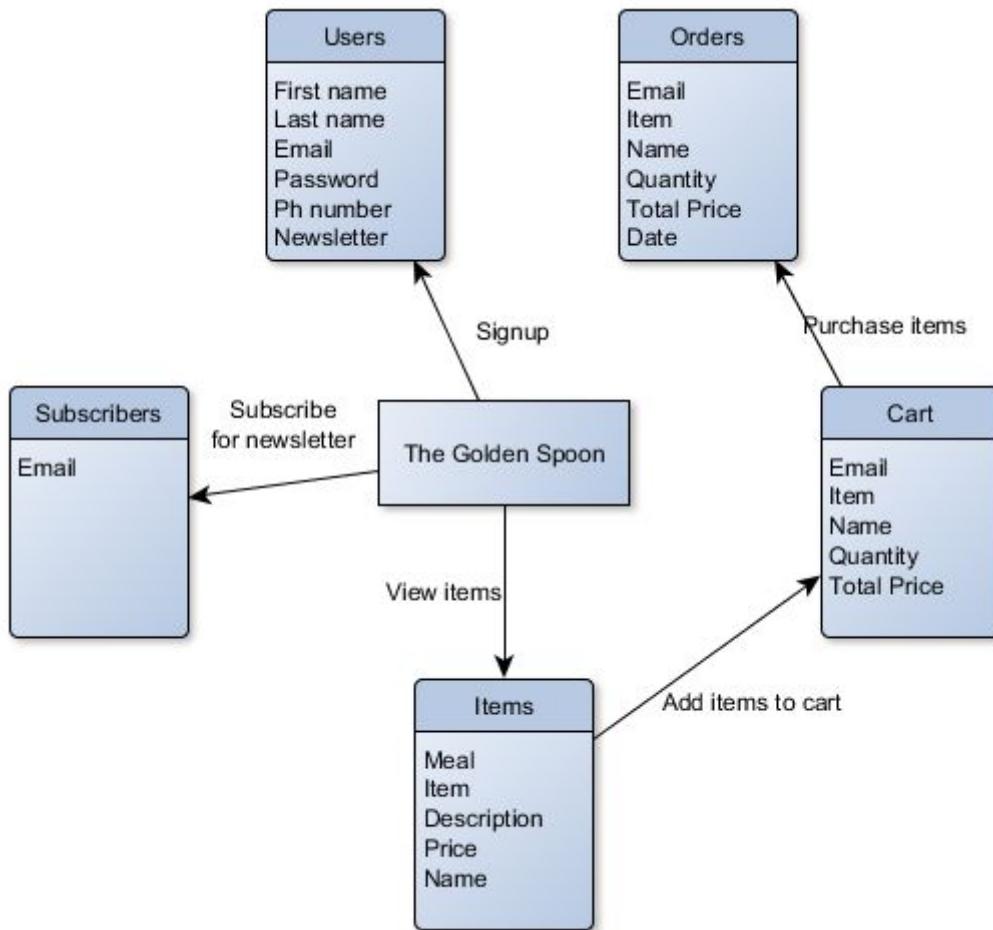
- Google Chrome:

Google Chrome is the web browser used for displaying the web pages. Chrome's developer tools is an inbuilt tools in chrome which can be used for debugging and editing. Live editing of the code is possible with developer tools. It also allows mobile view for developers to see and edit the way web pages look in mobile phones and tablets.

- Git:

Git is an open source version control software. It helps in keeping track of the changes made in each version. If the developer wants to roll back to the previous version, i.e, remove all the recent changes that he made, he can do it with just a command on git. This information can also be stored on GitHub which is a version control application on the web. It allows other users to view our projects and also lets us collaborate with other developers. Using Git in our project made it easier for us to work simultaneously on the project and avoiding inconsistencies.

#### 4. Database and schema



MongoDB is used for maintaining the database for The Golden Spoon. MongoDB is a noSQL database which is based on documents and collections. It makes handling data in the database easier than using SQL. MongoDB Compass is a user interface for MongoDB. Mongoose is a Node.js framework used for handling MongoDB commands in an easier way. The schema can be standardized using mongoose.

Given below is the schema used:

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const ItemSchema = new Schema({
  meal: String,
  item: String,
  description: String,
  price: Number,
  name: String
});
```

```
const Item = mongoose.model('items', ItemSchema);
module.exports.Item = Item;

const UserSchema = new Schema({
    firstName: String,
    lastName: String,
    email: String,
    password: String,
    phnumber: String,
    newsletter: Boolean
});

const User = mongoose.model('user', UserSchema);
module.exports.User = User;

const SubscriberSchema = new Schema({
    email: String
});

const Subscriber = mongoose.model('subscribers', SubscriberSchema);
module.exports.Subscriber = Subscriber;

const CartSchema = new Schema({
    email: String,
    item: String,
    name: String,
    quantity: Number,
    totalprice: Number
});

const Cart = mongoose.model('cart', CartSchema);
module.exports.Cart = Cart;

const OrderSchema = new Schema({
    email: String,
    item: String,
    name: String,
    quantity: Number,
    totalprice: Number,
    date: String
});
```

```
const Order = mongoose.model('orders', OrderSchema);
module.exports.Order = Order;
```

MongoDB Compass Community - localhost:27017/tgs

Connect View Help

Local

5 DBS 7 COLLECTIONS

filter

> admin  
> config  
> local  
> ninjago

tgs

+ 🗑

localhost:27017 STANDALONE

MongoDB 4.0.11 Community

Collections

CREATE COLLECTION

| Collection Name | Documents | Avg. Document Size | Total Document Size | Num. Indexes | Total Index Size | Properties |
|-----------------|-----------|--------------------|---------------------|--------------|------------------|------------|
| carts           | 5         | 152.2 B            | 761.0 B             | 1            | 36.9 KB          | ⚙️         |
| items           | 61        | 211.6 B            | 12.9 KB             | 1            | 16.4 KB          | ⚙️         |
| orders          | 23        | 185.8 B            | 4.3 KB              | 1            | 36.9 KB          | ⚙️         |
| subscribers     | 20        | 63.8 B             | 1.3 KB              | 1            | 36.9 KB          | ⚙️         |
| users           | 8         | 162.9 B            | 1.3 KB              | 1            | 36.9 KB          | ⚙️         |

Type here to search

O H Chrome Edge Mail Camera Task View 06:48 PM ENG 27.10.2019

MongoDB Compass Community - localhost:27017/tgs.items

Connect View Collection Help

Local

5 DBs 7 Collections

filter

> admin  
> config  
> local  
> ninjago  
v tgs  
  cars  
  items  
  orders  
  subscribers  
  users

localhost:27017 STANDALONE

tgs.items

DOCUMENTS 61 TOTAL SIZE 12.6KB AVG. SIZE 212B INDEXES 1 TOTAL SIZE 16.0KB AVG. SIZE 16.0KB

Documents Aggregations Explain Plan Indexes

FILTER

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents 1 - 20 of 61

| #  | item                     | _id | Objectid | item String                      | description String                  | meal String        | price Double | name String                    |
|----|--------------------------|-----|----------|----------------------------------|-------------------------------------|--------------------|--------------|--------------------------------|
| 1  | 5d7294b21e5cc0f4c2984888 |     |          | "Barbeque Paneer Rice Bowl"      | "Fresh cilantro lime rice with "    | "Maincourse"       | 359          | "barbequepannericebowl "       |
| 2  | 5d7294b21e5cc0f4c2984881 |     |          | "Crispy Mushroom Rice Bowl"      | "Fresh cilantro lime rice with "    | "Maincourse"       | 365          | "crispymushroomricebowl"       |
| 3  | 5d7294b21e5cc0f4c2984882 |     |          | "Veg Tikka Masala Burrito"       | "A soft tortilla roll with deli "   | "Maincourse"       | 379          | "vegikkamasalaburrito"         |
| 4  | 5d7294b21e5cc0f4c2984883 |     |          | "Veg 7 layer Burrito"            | "A grilling soft Tortilla filled "  | "Maincourse"       | 419          | "veg7layerburrito"             |
| 5  | 5d7294b21e5cc0f4c2984884 |     |          | "Veg Sriracha Burrito"           | "A soft tortilla roll with veg "    | "Maincourse"       | 345          | "vegsrirachaburrito"           |
| 6  | 5d7294b21e5cc0f4c2984885 |     |          | "Fajita Veggie Quesadilla"       | "A soft grilled Tortilla folded "   | "Appetizers"       | 335          | "fajitaveggiequesadilla"       |
| 7  | 5d7294b21e5cc0f4c2984886 |     |          | "Cheese Quesadilla"              | "A soft grilled Tortilla folded "   | "Appetizers"       | 335          | "cheesequesadilla"             |
| 8  | 5d7294b21e5cc0f4c2984887 |     |          | "Pesto Cottage Cheese Panini"    | "Three coloured bell peppers co     | "Appetizers"       | 310          | "pestocottagecheesepanini"     |
| 9  | 5d7294b21e5cc0f4c2984888 |     |          | "Burger Trio"                    | "Miniburger(3) with french frie     | "Maincourse"       | 300          | "burgertrio"                   |
| 10 | 5d7294b21e5cc0f4c2984889 |     |          | "Nachos Grande"                  | "Tortilla chips with Mexican be     | "Appetizers"       | 350          | "nachosgrande "                |
| 11 | 5d7294b21e5cc0f4c298488a |     |          | "Kit Kat Shake"                  | "Signature milkshake made of ki     | "Beverages"        | 320          | "kitkatshake"                  |
| 12 | 5d7294b21e5cc0f4c298488b |     |          | "Strawberry Cheese Cake Shake"   | "Strawberry ice cream blended w     | "Beverages"        | 340          | "strawberrycheesecakeshake"    |
| 13 | 5d7294b21e5cc0f4c298488c |     |          | "Peri Peri Potato Nachos"        | "Savory crispy nachos (como-to:     | "Appetizers"       | 280          | "periperipotatonachos"         |
| 14 | 5d7294b21e5cc0f4c298488d |     |          | "Peri Peri Potato Veg Salad"     | "Healthy salad bowl with peri p     | "Soups and Salads" | 235          | "periperipotatoveg salad"      |
| 15 | 5d7294b21e5cc0f4c298488e |     |          | "Grilled Peri Peri Chicken Sala  | "Healthy salad bowl with peri pille | "Soups and Salads" | 290          | "grilledperiperi chickensalad" |
| 16 | 5d7294b21e5cc0f4c298488f |     |          | "Mexican corn Fiesta Quesadilla" | "Corn,black beans and jalapenos "   | "Appetizers"       | 370          | "mexicancornfiestaquesadilla"  |
| 17 | 5d7294b21e5cc0f4c2984890 |     |          | "Cheesy Tostada"                 | "A blend of soft and crunchy to     | "Appetizers"       | 240          | "cheesytostada"                |
| 18 | 5d7294b21e5cc0f4c2984891 |     |          | "Rhn Chicken Wran"               | "Delicious wrap with shredded c     | "Maincourse"       | 340          | "rhncrchnickenwrap"            |

Type here to search

O Hi ! S C 605 PM ENG 27-10-2019

MongoDB Compass Community - localhost:27017/tgs.users

Connect View Collection Help

**Local**

6 DBS 7 COLLECTIONS

**tgs.users**

Documents Aggregations Explain Plan Indexes

**INSERT DOCUMENT** VIEW LIST TABLE

Displaying documents 1 - 7 of 7

| # | Items | _id                       | ObjectID  | firstName | String                     | lastName    | String      | email        | String | password | String | phnumber | String |
|---|-------|---------------------------|-----------|-----------|----------------------------|-------------|-------------|--------------|--------|----------|--------|----------|--------|
| 1 |       | 5d9e65f6fbfc9b9e34ab8c092 | "Navya"   | "R6"      | "navya@gmail.com"          | "navya"     | "navya"     | "9876543210" |        |          |        |          |        |
| 2 |       | 5d9e75dfb8999955dc2c945d  | "Alisha"  | "Shah"    | "alisha@gmail.com"         | "alisha"    | "alisha"    | "9876543210" |        |          |        |          |        |
| 3 |       | 5d9e190f0a2ef1f9b7b21742  | "Anushka" | "Shah"    | "anushkashah17@gmail.com"  | "alisha"    | "alisha"    | "9712917794" |        |          |        |          |        |
| 4 |       | 5da0956987547f332088319ec | "Navya"   | "R6"      | "navyarg11@gmail.com"      | "navyarg11" | "navyarg11" | "9880320171" |        |          |        |          |        |
| 5 |       | 5da21bbc726c29085ab57c37  | "Alisha"  | "Shah"    | "alishashah1999@gmail.com" | "alisha"    | "alisha"    | "9899444040" |        |          |        |          |        |
| 6 |       | 5da854916ededa3758e99ddd  | "Archa"   | "T"       | "archana@gmail.com"        | "abc"       | "abc"       | "9899777878" |        |          |        |          |        |
| 7 |       | 5da8649b3f6f6d3df40b2423  | "Shruthi" | "Kumar"   | "shruthikumar@gmail.com"   | "sldhgj"    | "sldhgj"    | "9852659751" |        |          |        |          |        |

MongoDB Compass Community - localhost:27017/tgs.carts

Connect View Collection Help

**Local**

6 DBS 7 COLLECTIONS

**tgs.Carts**

Documents Aggregations Explain Plan Indexes

**INSERT DOCUMENT** VIEW LIST TABLE

Displaying documents 1 - 5 of 5

| # | Items | _id                      | ObjectID | quantity | Int32                 | email                         | String                     | item | String | name | String | totalprice | Int32 |
|---|-------|--------------------------|----------|----------|-----------------------|-------------------------------|----------------------------|------|--------|------|--------|------------|-------|
| 1 |       | 5da09aa08c06c900293af80b | 4        |          | "navyarg11@gmail.com" | "Muesli Fruit Yogurt Parfait" | "mueslifruityogurtparfait" |      |        |      |        | 688        |       |
| 2 |       | 5da7421998c18802e8a21311 | 1        |          | "navyarg11@gmail.com" | "Green Salad "                | "greensalad"               |      |        |      |        | 268        |       |
| 3 |       | 5da7421998c18802e8a21312 | 4        |          | "navyarg11@gmail.com" | "Peri Peri Potato Veg Salad"  | "periperipotatoveg salad"  |      |        |      |        | 470        |       |
| 4 |       | 5da7421d98c18802e8a21313 | 2        |          | "navyarg11@gmail.com" | "Peri Peri Potato Veg Salad"  | "periperipotatoveg salad"  |      |        |      |        | 470        |       |
| 5 |       | 5da7421d98c18802e8a21314 | 1        |          | "navyarg11@gmail.com" | "Green Salad "                | "greensalad"               |      |        |      |        | 268        |       |

MongoDB Compass Community - localhost:27017/tgs.orders

Connect View Collection Help

Local

6 DBS 7 COLLECTIONS

filter

- > admin
- > config
- > local
- > ninago
- > tgs
  - carts
  - items
  - orders
  - subscribers
  - users

tgs.orders

Documents Aggregations Explain Plan Indexes

INSERT DOCUMENT VIEW LIST TABLE

DOCUMENTS 23 TOTAL SIZE 4.2KB AVG. SIZE 186B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0B

FIND RESET ...

Displaying documents 1 - 20 of 23

| #  | items | _id                       | ObjectID | quantity | Int32 | email                      | String | item                            | String | name                          | String | totalprice | Int32 | date |
|----|-------|---------------------------|----------|----------|-------|----------------------------|--------|---------------------------------|--------|-------------------------------|--------|------------|-------|------|
| 1  |       | 5da020988fb00c0954171553  |          | 1        |       | "navyarg1@gmail.com"       |        | "Peri Peri Potato Veg Salad"    |        | "periperiopotatovegsalad"     |        | 235        |       |      |
| 2  |       | 5da09a458c06c908203af8d8  |          | 1        |       | "navyarg1@gmail.com"       |        | "Grilled Chicken Corn Salad"    |        | "grilledchickencornsalad"     |        | 299        |       |      |
| 3  |       | 5da09a458c06c908203af8d9  |          | 1        |       | "navyarg1@gmail.com"       |        | "Peri Peri Potato Nachos"       |        | "periperiopotatonachos"       |        | 288        |       |      |
| 4  |       | 5da09a458c06c908203af8d4  |          | 1        |       | "navyarg1@gmail.com"       |        | "Green Grapes Apple Shake"      |        | "greengrapesappleshake"       |        | 290        |       |      |
| 5  |       | 5da09a458c06c908203af8d6  |          | 1        |       | "navyarg1@gmail.com"       |        | "Chicken In Black Pepper Sauce" |        | "chickeninblackpeppersauce"   |        | 500        |       |      |
| 6  |       | 5da21b66726c290885b57c12  |          | 3        |       | "navyarg1@gmail.com"       |        | "Jam-Centred Cheesecake"        |        | "jamcenteredcheesecake"       |        | 700        |       |      |
| 7  |       | 5da21b66726c290885b57c13  |          | 1        |       | "navyarg1@gmail.com"       |        | "Tomato & Basil Soup "          |        | "tomato&basilsoup"            |        | 298        |       |      |
| 8  |       | 5da21b66726c290885b57c34  |          | 4        |       | "navyarg1@gmail.com"       |        | "Grilled Peri Peri Chicken Sali |        | "grilledperiperiichicensalad" |        | 588        |       |      |
| 9  |       | 5da21b66726c290885b57c35  |          | 2        |       | "navyarg1@gmail.com"       |        | "Banana Walnut Teacake"         |        | "bananawalnutteacake"         |        | 668        |       |      |
| 10 |       | 5da21b66726c290885b57c36  |          | 2        |       | "navyarg1@gmail.com"       |        | "Kit Kat Shake"                 |        | "kitkatshake"                 |        | 648        |       |      |
| 11 |       | 5da21c1c0726c290885b57c3c |          | 2        |       | "alishashah1999@gmail.com" |        | "Cheese Quesadilla"             |        | "cheesequesadilla"            |        | 670        |       |      |
| 12 |       | 5da21c1c0726c290885b57c1b |          | 2        |       | "alishashah1999@gmail.com" |        | "Fajita Veggie Quesadilla"      |        | "fajitaveggiquesadilla"       |        | 670        |       |      |
| 13 |       | 5da21c1c0726c290885b57c3d |          | 1        |       | "alishashah1999@gmail.com" |        | "Kit Kat Shake"                 |        | "kitkatshake"                 |        | 320        |       |      |
| 14 |       | 5da21c1c0726c290885b57c1f |          | 1        |       | "alishashah1999@gmail.com" |        | "Kit Kat Shake"                 |        | "kitkatshake"                 |        | 320        |       |      |
| 15 |       | 5da83aa4ffad9e02d44bc19c  |          | 1        |       | "alisharg1@gmail.com"      |        | "Kit Kat Shake"                 |        | "kitkatshake"                 |        | 320        |       |      |
| 16 |       | 5da83aa4ffad9e02d44bc19d  |          | 1        |       | "alisharg1@gmail.com"      |        | "Strawberry Cheese Cake Shake"  |        | "strawberrycheesecakeshake"   |        | 340        |       |      |
| 17 |       | 5da853c0edeeaa3758e99dd3  |          | 2        |       | "archana@gmail.com"        |        | "Kit Kat Shake"                 |        | "kitkatshake"                 |        | 640        |       |      |
| 18 |       |                           |          |          |       |                            |        |                                 |        |                               |        |            |       |      |

06:52 PM 27-10-2019

MongoDB Compass Community - localhost:27017/tgs.subscribers

Connect View Collection Help

Local

6 DBS 7 COLLECTIONS

filter

- > admin
- > config
- > local
- > ninago
- > tgs
  - carts
  - items
  - orders
  - subscribers
  - users

tgs.subscribers

Documents Aggregations Explain Plan Indexes

INSERT DOCUMENT VIEW LIST TABLE

DOCUMENTS 13 TOTAL SIZE 804B AVG. SIZE 62B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0B

FIND RESET ...

Displaying documents 1 - 13 of 13

| #  | items | _id                       | ObjectID | email                           | String | _v | Int32 |
|----|-------|---------------------------|----------|---------------------------------|--------|----|-------|
| 1  |       | 5d9c75dfb899955dc2c945e   |          | "alisha@gmail.com"              |        | 0  |       |
| 2  |       | 5d9e19bf0a2ef198b7b2745   |          | "anuskashah17@gmail.com"        |        | 0  |       |
| 3  |       | 5da09b12c8c06c908203af8dc |          | "navyarg1@gmail.com"            |        | 0  |       |
| 4  |       | 5da21753da0a030db8081d19  |          | "alishashah1999@gmail.com"      |        | 0  |       |
| 5  |       | 5da21753da0a030db8081d1a  |          | "alishashah1999@gmail.com"      |        | 0  |       |
| 6  |       | 5da21743726c290885b57c2a  |          | "navyar_g2017@ritstudent.ac.in" |        | 0  |       |
| 7  |       | 5da2199c726c290885b57c2b  |          | "rishabhkaul2021@gmail.com"     |        | 0  |       |
| 8  |       | 5da21b448726c290885b57c31 |          | "navyarg1@gmail.com"            |        | 0  |       |
| 9  |       | 5da22b_f67976726568b77cda |          | "navyarg1@gmail.com"            |        | 0  |       |
| 10 |       | 5da46d99c18022eb2131a     |          | "kg@gmail.vm"                   |        | 0  |       |
| 11 |       | 5da854916dedea3758e99dd6  |          | "archana@gmail.com"             |        | 0  |       |
| 12 |       | 5da854916dedea3758e99dd7  |          | "archana@gmail.com"             |        | 0  |       |
| 13 |       | 5da85493bedeeaa3758e99dd8 |          | "archana@gmail.com"             |        | 0  |       |

06:53 PM 27-10-2019

## 5. Screenshots of working model

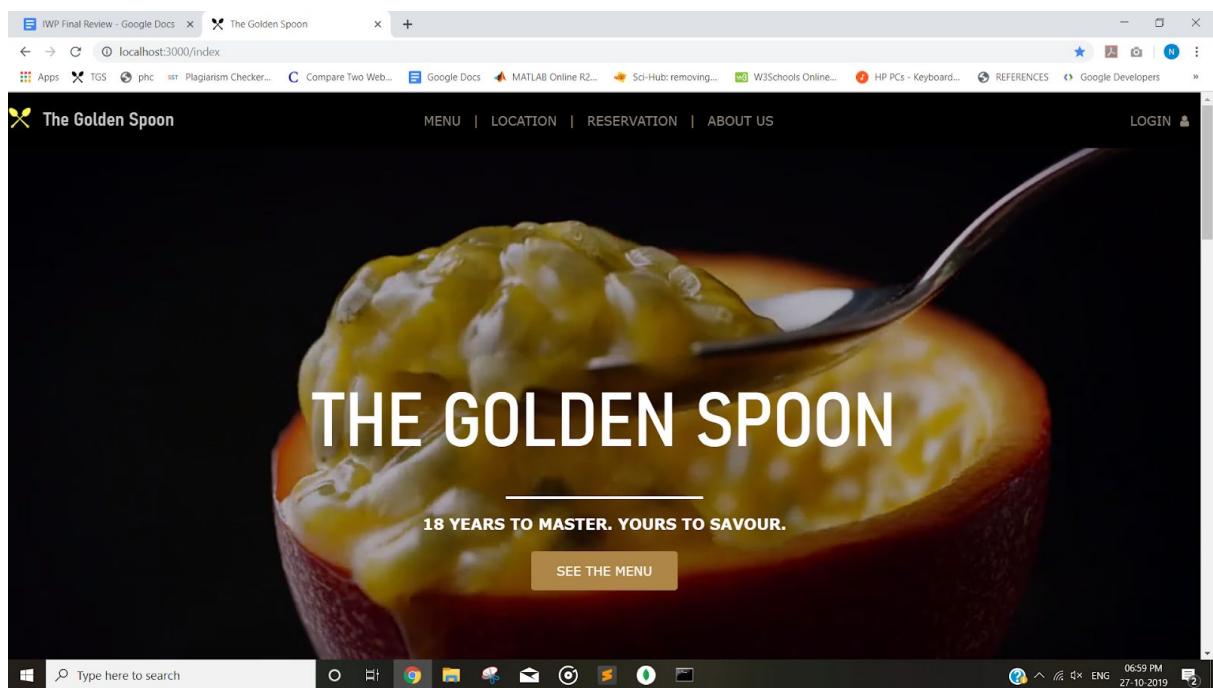
```
Command Prompt - nodemon main
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\navya>F:
F:>cd SEM 5\IWP\Project\The Golden Spoon

F:\SEM 5\IWP\Project\The Golden Spoon>nodemon main
[nodemon] 1.19.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node main.js`

(node:13876) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.

**Now listening for requests**
```



# THE GOLDEN SPOON

18 YEARS TO MASTER. YOURS TO SAVOUR.

[SEE THE MENU](#)

*Redefining the art of culinary,  
one grain at a time.*

## LOCAL FLAVOUR

PEACE  
LOVE  
4 RIVERS.  
JOIN THE FAMILY

Family Owned. Locally Made. Community Focused.

[FIND US](#)

## IN THE BEGINNING

The story of TGS

[OUR HISTORY](#)

## LOYAL 4 LIFE

[Join our club](#)

[SIGN UP](#)

SIGN UP FOR OUR NEWSLETTER!

EMAIL\*

[SUBSCRIBE](#)

# OUR LOCATIONS

COME ON IN AND PULL UP A CHAIR.

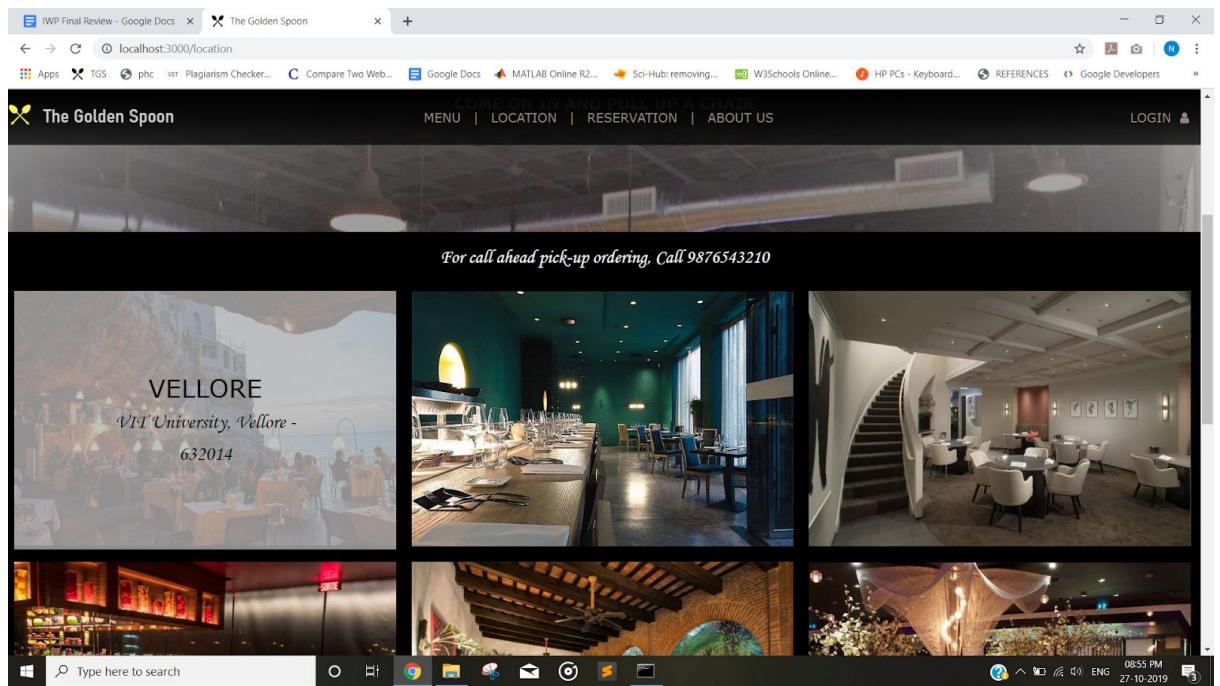
*For call ahead pick-up ordering, Call 9876543210*



SIGN UP FOR OUR NEWSLETTER!

EMAIL\*

SUBSCRIBE



# OUR BEGINNINGS

IT ALL STARTED IN A GARAGE



*"TGS had its humble beginnings. Just me, a smoker, and an age-old challenge—cooking brisket."*

John Rivers never set out to create the most successful BBQ chain in Florida. He's not a classically trained chef or graduate of culinary school. It was in Texas that he found the two loves of his life: his wife and the delicate art of brisket. And it's been those very two godsends that inspired TGS as we know it.

John spent 20 years in the healthcare industry before retiring as president of a billion-dollar company. During that time, he had the opportunity to travel the country, honing his taste and talent for brisket perfection. As he soaked up the flavors and practices of 'cue fare countrywide, John's dream of de-regionalizing BBQ was born.

## FUN FACT:

*Why TGS?* In short, it represents our family, John, Monica (wife), Jared (son) and Cameron (daughter). But they also appreciate the serendipitous double meaning with Genesis 2:10, where TGS branch from that which flows out of Eden. Now you know!

*"This was never supposed to be a restaurant. My definition of helping meant doing what felt most natural when comfort was in order—feed those in need."*

TGS Smokehouse began with the launch of the "Barbecue Ministry" in 2004, when John hosted a cookout fundraiser to support a local family whose young daughter was battling cancer. This one event resulted in a passion for supporting local schools, churches, and charitable organizations, and a few years later, smoking thousands of pounds of meat out of a garage just wouldn't cut it anymore.

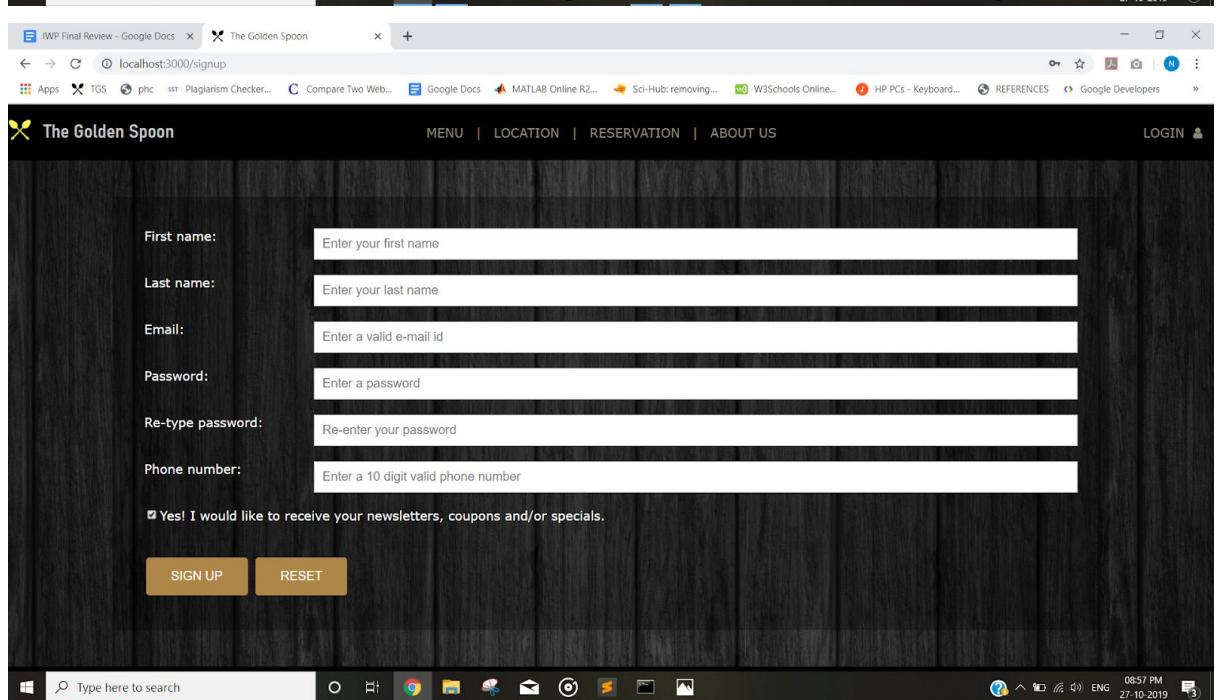
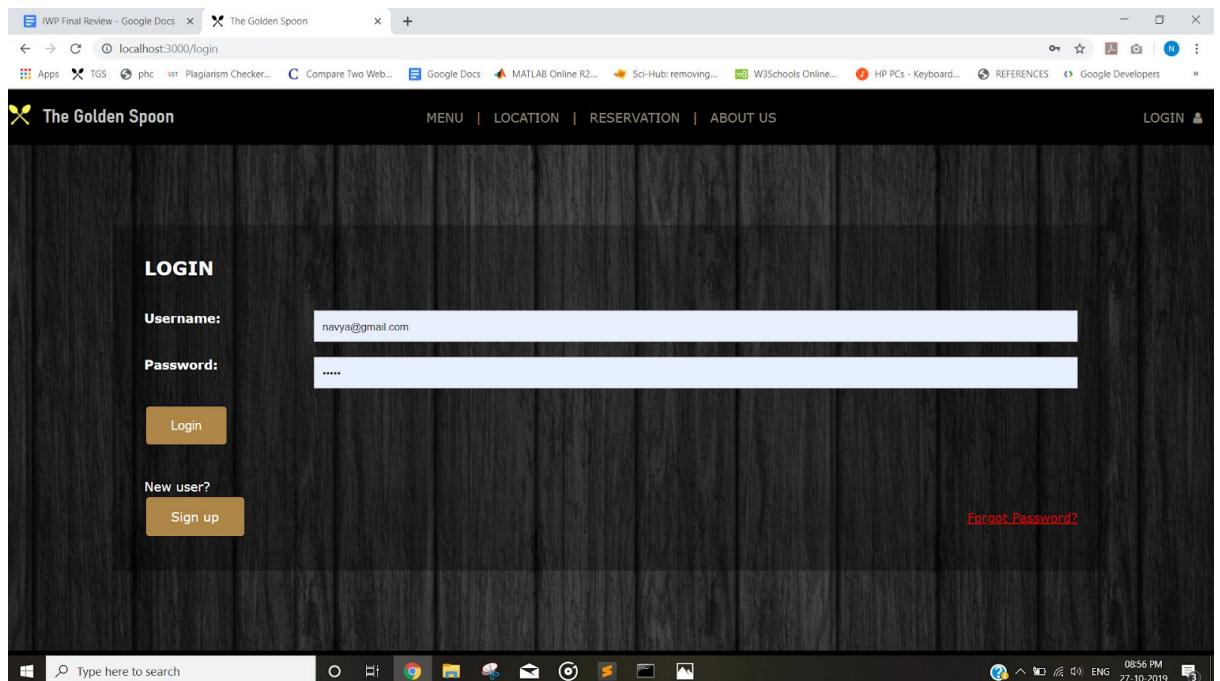
After trials, tribulations, and a whole lot of burnt ends, we finally flipped that "Hot Brisket Now" sign on October of 2009. Within the first hour, a line had formed out the door... and then around the corner.

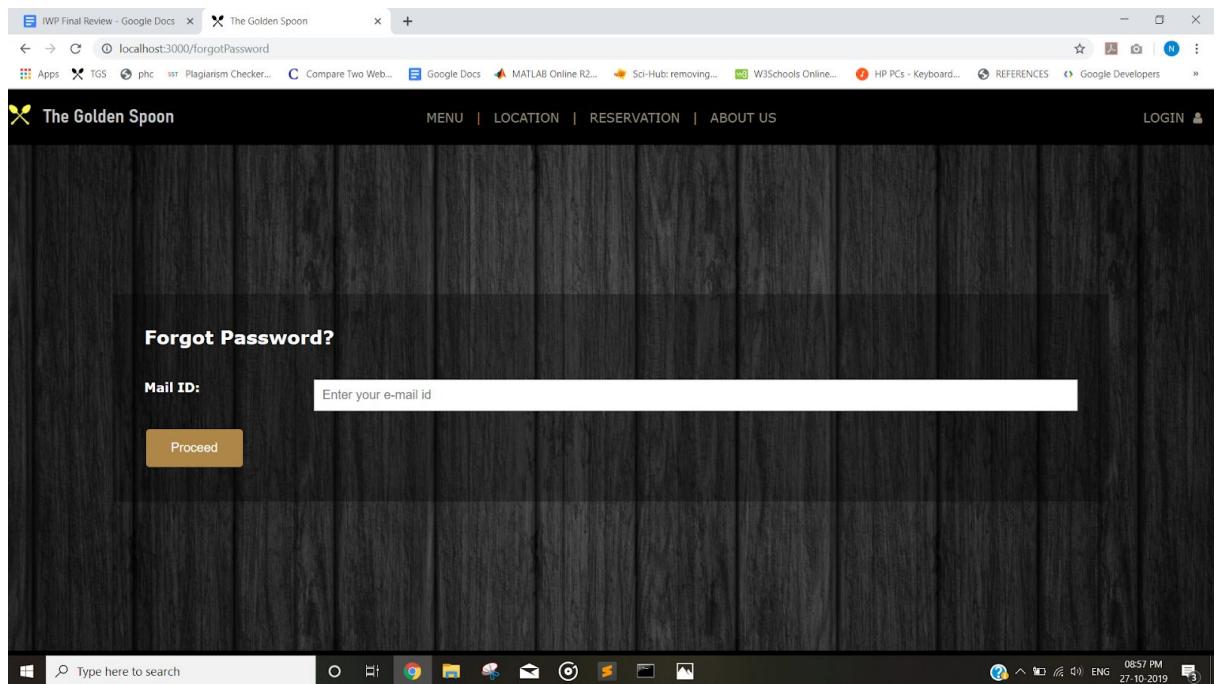
Today, we have several locations across the state of Florida, with plans for continued expansion and more thousands 1,000 amazing folks on our TGS team. Most importantly, the Barbecue Ministry remains our foundation and focus, and will until the cows come home.

SIGN UP FOR OUR NEWSLETTER!

EMAIL\*

SUBSCRIBE





# OUR MENU

MAY THE BELLY RULE THE MIND

*For call ahead pick-up ordering, Call 9876543210*

## SOUPS & SALADS



## APPETIZERS



## MAIN COURSE



## BEVERAGES



## DESSERTS



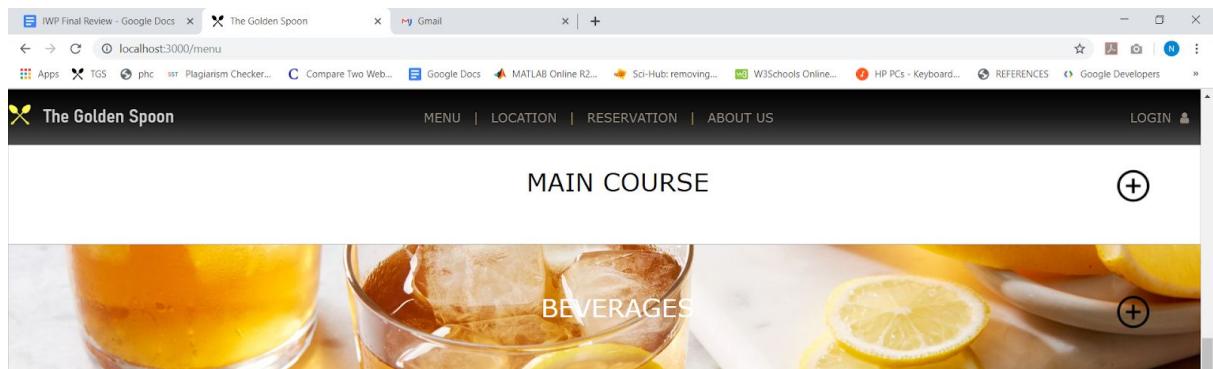
[ADD TO CART](#)

[RESET](#)

SIGN UP FOR OUR NEWSLETTER!

EMAIL\*

[SUBSCRIBE](#)



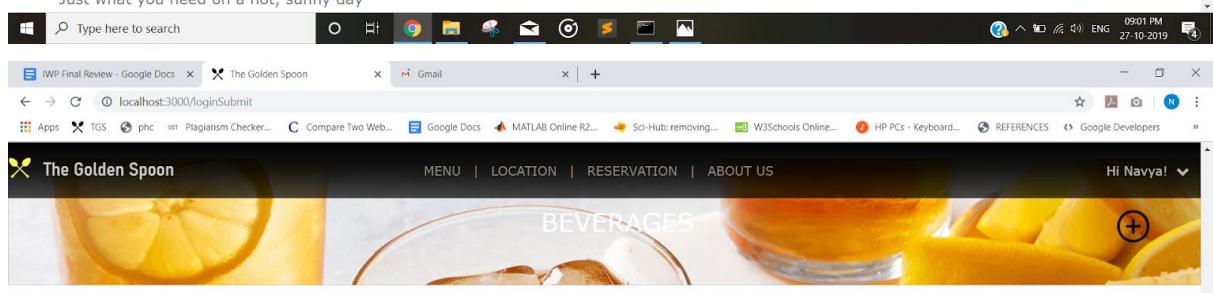
**Kit Kat Shake** - Signature milkshake made of kitkat topped with whipped cream and kitkat Rs. 320

**Strawberry Cheese Cake Shake** - Strawberry ice cream blended with kiri cheese and graham cracker powder Rs. 340

**Epigamia Greek Yogurt Smoothie - Strawberry** - Strawberry yogurt smoothie filled with almonds, topped with whip cream Rs. 390

**Epigamia Greek Yogurt Smoothie - Alphonso Mango** - Greek yogurt smoothie made of fresh alphonso mangoes Rs. 390

**Cold Coffee** - Just what you need on a hot, sunny day Rs. 240



**Kit Kat Shake** - Signature milkshake made of kitkat topped with whipped cream and kitkat Rs. 320 [ ]

**Strawberry Cheese Cake Shake** - Strawberry ice cream blended with kiri cheese and graham cracker powder Rs. 340 [ ]

**Epigamia Greek Yogurt Smoothie - Strawberry** - Strawberry yogurt smoothie filled with almonds, topped with whip cream Rs. 390 [ ]

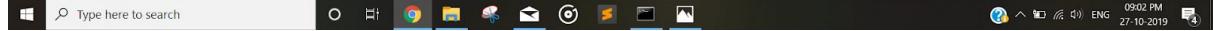
**Epigamia Greek Yogurt Smoothie - Alphonso Mango** - Greek yogurt smoothie made of fresh alphonso mangoes Rs. 390 [ ]

**Cold Coffee** - Just what you need on a hot, sunny day Rs. 240 [ 1 ]

**Oats Banana Date Smoothie** - Delicious blend of chopped dates and bananas with a scoop of rolled oats Rs. 250 [ 2 ]

**Peanut Butter Banana Shake** - Sticky peanut butter and chunks of banana blended into a yummy shake. Rs. 250 [ ]

**Avocado Banana Super Shake** - Rs. 260 [ ]



The Golden Spoon

Gmail

localhost:3000/add2cart

Items successfully added to cart

# OUR MENU

MAY THE BELLY RULE THE MIND

Type here to search

Temporary error 0

localhost:3000/cart

The Golden Spoon

MENU | LOCATION | RESERVATION | ABOUT US

Hi Navya!

~ITEMS IN CART~

| Item                      | Quantity | Price |
|---------------------------|----------|-------|
| Cold Coffee               | 1        | 240   |
| Oats Banana Date Smoothie | 2        | 500   |
| Total Cost:               | 740      |       |

Proceed to checkout

Type here to search

09:02 PM  
27-10-2019

IWP Final Review - Google Docs    The Golden Spoon    Temporary error 0

localhost:3000/payment

Apps TGS phc Plagiarism Checker... Compare Two Web... Google Docs MATLAB Online R2... Sci-Hub: removing... W3Schools Online... HP PCs - Keyboard... REFERENCES Google Developers

The Golden Spoon MENU | LOCATION | RESERVATION | ABOUT US Hi Navya!

Enter delivery address:

Full name:

Phone number:  10-digit mobile number without prefixes

Pincode:  6 digit [0-9] pincode

House No.:  Flat / House No. / Floor / Building

Locality:  Colony / Street / Locality

Landmark:  Landmark

City:

Credit Card   Debit Card   Net Banking   Cash on Delivery

Type here to search

IWP Final Review - Google Docs    The Golden Spoon    Temporary error 0

localhost:3000/creditcard

Apps TGS phc Plagiarism Checker... Compare Two Web... Google Docs MATLAB Online R2... Sci-Hub: removing... W3Schools Online... HP PCs - Keyboard... REFERENCES Google Developers

The Golden Spoon MENU | LOCATION | RESERVATION | ABOUT US Hi Navya!

Payment:    

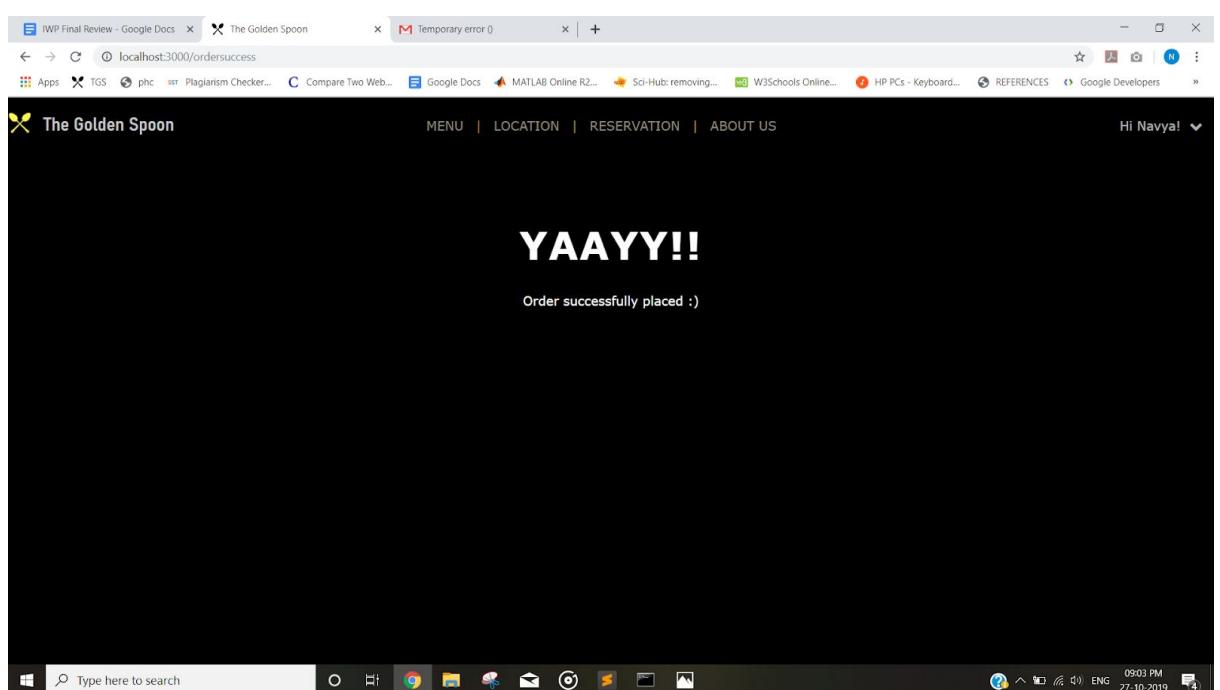
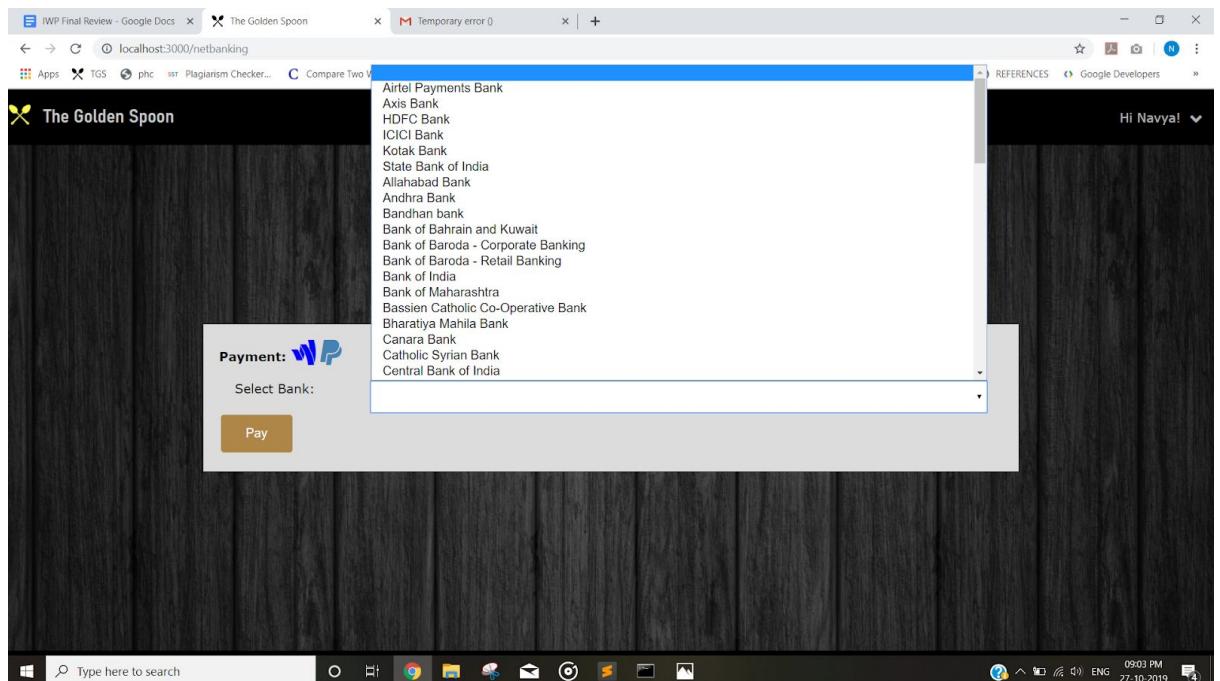
Name on card:  Card holder's name

Credit card number:  XXXX XXXX XXXX XXXX

Expiry:  mm/yy

CVV:  XXX

Pay



IWP Final Review - Google Docs   X The Golden Spoon   X Temporary error 0

localhost:3000/orders   Apps TGS phc Plagiarism Checker... Compare Two Web... Google Docs MATLAB Online R2... Sci-Hub: removing... W3Schools Online... HP PCs - Keyboard... REFERENCES Google Developers

The Golden Spoon MENU | LOCATION | RESERVATION | ABOUT US Hi Navya! ▾

~ORDER HISTORY~

| Item                            | Quantity | Price | Date                |
|---------------------------------|----------|-------|---------------------|
| Peri Peri Potato Veg Salad      | 1        | 235   | 2019-10-11 12:19:44 |
| Chicken In Black Pepper Sauce   | 1        | 500   | 2019-10-11 20:42:23 |
| Jam-Centred Cheesecake          | 3        | 700   | 2019-10-12 23:58:54 |
| Tomato & Basil Soup             | 1        | 290   | 2019-10-12 23:58:54 |
| Grilled Peri Peri Chicken Salad | 4        | 580   | 2019-10-12 23:58:54 |
| Banana Walnut Teacake           | 2        | 660   | 2019-10-12 23:58:54 |
| Kit Kat Shake                   | 2        | 640   | 2019-10-12 23:58:54 |
| Strawberry Cheese Cake Shake    | 1        | 340   | 2019-10-26 13:1:40  |
| Nachos Grande                   | 1        | 350   | 2019-10-27 19:7:2   |
| Cold Coffee                     | 1        | 240   | 2019-10-27 19:7:2   |
| Cold Coffee                     | 1        | 240   | 2019-10-27 21:3:46  |
| Oats Banana Date Smoothie       | 2        | 500   | 2019-10-27 21:3:46  |

Type here to search   09:04 PM ENG 27-10-2019

The Golden Spoon MENU | LOCATION | RESERVATION | ABOUT US Hi Navya! ▾

Note: All tables are 4 seater tables. Tables can be joined upon request.

City:

Time:

Duration:

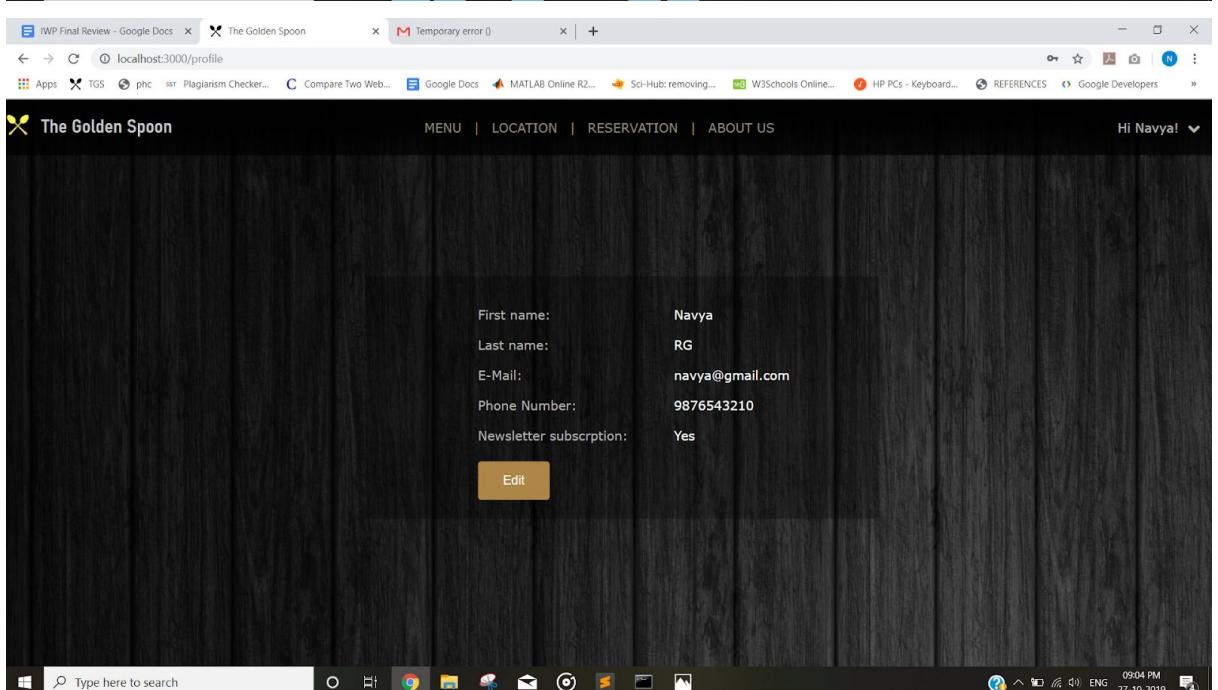
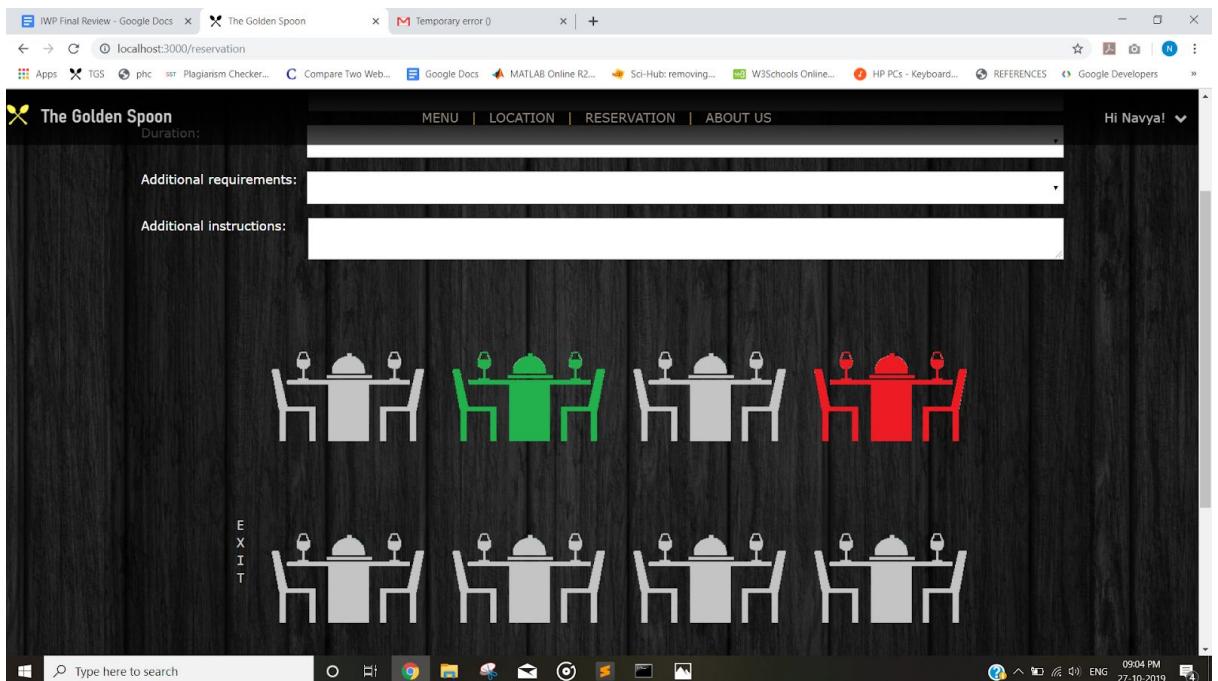
Additional requirements:

Additional instructions:





Reserve



The screenshot shows a web browser window with two tabs open. The active tab is titled 'The Golden Spoon' and displays a user profile update form. The form fields include 'First name' (Navya), 'Last name' (empty), 'Old Password' (empty), 'New Password' (empty), 'Re-type password' (empty), and 'Phone number' (9876543210). There is a checked checkbox for receiving newsletters and a 'UPDATE' button.

First name: Navya

Last name:

Old Password:

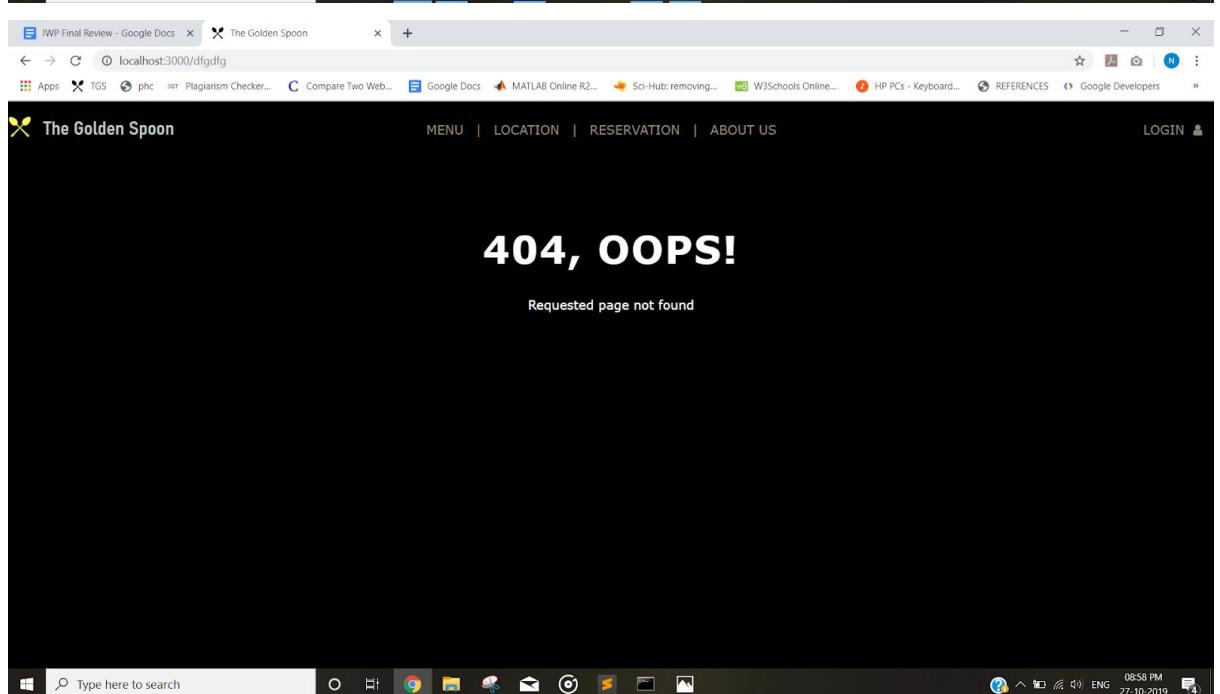
New Password:

Re-type password:

Phone number: 9876543210

Yes! I would like to receive your newsletters, coupons and/or specials.

UPDATE



Inbox\_Other - Gmail 2

+ New mail

Accounts

- Gmail nayarg5@gmail.com
- Gmail 2 nayarg11@gmail.com
- Outlook 2 nayarg11@outlook.com
- Gmail 3 nayarg2017@vitstudent.ac...

Folders

- Inbox
- More

tgs

Results All folders

21 October 2019

thegoldenspoon.tgs@gmail.com (no subject) Mon 11:39 AM Hello from python

11 October 2019

thegoldenspoon.tgs@gmail.com Forgot Password 11-10-2019 Your password is: nayarg11

thegoldenspoon.tgs@gmail.com Newsletter subscription 11-10-2019 Thank you for subscribing to our w

thegoldenspoon.tgs@gmail.com Newsletter subscription 11-10-2019 Thank you for subscribing to our w

thegoldenspoon.tgs@gmail.com Signup Successful 11-10-2019 Thank you for joining our family!

Can't see what you're looking for? Try using different keywords.

Reply Reply all Forward Archive Delete Set flag ...

**Signup Successful**

thegoldenspoon.tgs@gmail.com <thegoldenspoon.tgs@gmail.com>  
11-10-2019 08:15 PM  
To: nayarg11@gmail.com

**Thank you for joining our family!**

09:10 PM 27-10-2019

Inbox\_Other - Gmail 2

+ New mail

Accounts

- Gmail nayarg5@gmail.com
- Gmail 2 nayarg11@gmail.com
- Outlook 2 nayarg11@outlook.com
- Gmail 3 nayarg2017@vitstudent.ac...

Folders

- Inbox
- More

tgs

Results All folders

21 October 2019

thegoldenspoon.tgs@gmail.com (no subject) Mon 11:39 AM Hello from python

11 October 2019

thegoldenspoon.tgs@gmail.com Forgot Password 11-10-2019 Your password is: nayarg11

thegoldenspoon.tgs@gmail.com Newsletter subscription 11-10-2019 Thank you for subscribing to our w

thegoldenspoon.tgs@gmail.com Newsletter subscription 11-10-2019 Thank you for subscribing to our w

thegoldenspoon.tgs@gmail.com Signup Successful 11-10-2019 Thank you for joining our family!

Can't see what you're looking for? Try using different keywords.

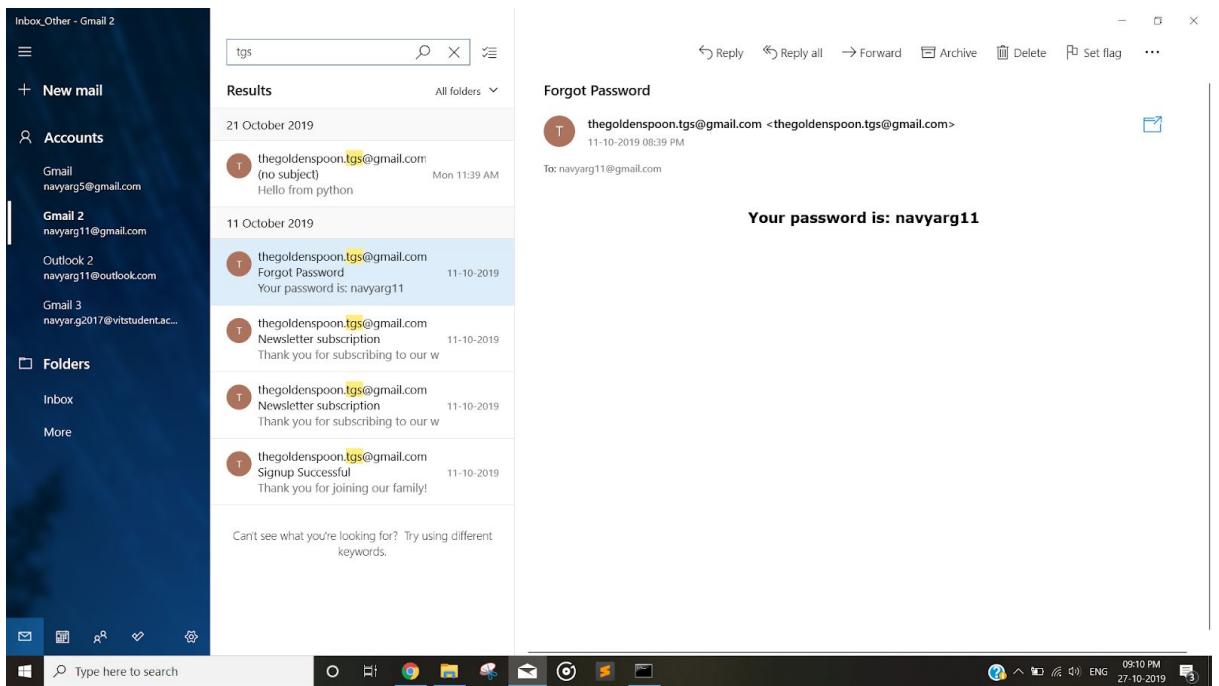
Reply Reply all Forward Archive Delete Set flag ...

**Newsletter subscription**

thegoldenspoon.tgs@gmail.com <thegoldenspoon.tgs@gmail.com>  
11-10-2019 08:15 PM  
To: nayarg11@gmail.com

**Thank you for subscribing to our weekly newsletter!**

09:10 PM 27-10-2019



## 6. Code snippets

Some of the libraries included are given below:

```

const express = require('express');
const bodyParser = require('body-parser');
const tgsController = require('./controllers/tgsController');
const mongoose = require('mongoose');
const models = require('./models/models');
const nodemailer = require('nodemailer');

const app = express();

const cookieParser = require('cookie-parser');
const session = require('express-session');

app.use(cookieParser());
app.use(session({
  secret: "SecretKey",
  resave: false,
  saveUninitialized: true
}));
```

```
var urlencodedParser = bodyParser.urlencoded({extended: false});
mongoose.connect('mongodb://localhost/tgs');
mongoose.Promise = global.Promise;

app.set('view engine', 'ejs');
app.use('/assets', express.static('./public/assets'));
app.use('/images', express.static('./public/images'));
app.use('/videos', express.static('./public/videos'));
app.use(bodyParser.json());
```

Enabling the server to listen for requests:

```
app.listen(3000, function(){
    console.log(**Now listening for requests**);
});
```

Examples of routing:

```
app.get('/forgotPassword', function(req, res){
    res.render('forgotPassword', {message: ""});
});
```

```
app.get('/menu', function(req, res){
    tgsController.fetchMenu(req, res);
});
```

```
app.get('/signup', function(req, res){
    res.render('signup');
});
```

```
app.get("*", function(req,res){
    res.render('404', {user: req.session.user});
});
```

```
app.post('/deleteFromCart', urlencodedParser, function(req, res){
    tgsController.deleteFromCart(req, res);
});
```

```
app.post('/updateMenu', urlencodedParser, function(req, res){
    var i = tgsController.updateMenu(req.body, res);
});
```

```
app.post('/signupSubmit', urlencodedParser, function(req, res){
```

```
tgsController.signupSubmit(req, res);
});

app.post('/loginSubmit', urlencodedParser, function(req, res){
    tgsController.loginSubmit(req, res);
});
```

#### Login and logout:

```
module.exports.loginSubmit = function(req, res){
    var data = req.body;
    models.User.findOne({email: data.username}, function (err, docs) {
        if(docs==undefined){
            res.render('login', {message: '**User does not exist**'});
        }
        else if(docs.password!=data.password){
            res.render('login', {message: '**Incorrect password**'});
        }
        else if(docs.password==data.password){
            req.session.user = data.username;
            console.log('After login: ',req.session);
            fetchMenu(req, res);
        }
    });
};

module.exports.logout = function(req, res){
    req.session.user = undefined;
    console.log('After logout: ', req.session);
    res.render('login', {message: '**Successfully logged out**'});
};
```

#### Sending mail:

```
var sendMail = function(receiver, sub='Newsletter subscription', msg="Thank you
for subscribing to our weekly newsletter!"){

    models.Subscriber.findOne({email: receiver}, function(err, docs){

        if(docs==undefined){
            models.Subscriber.create({email: receiver}).then(function(){});

        }
    });
}

var transporter = nodemailer.createTransport({
    service: 'gmail',
    auth: {
```

```

        user: 'thegoldenspoon.tgs@gmail.com',
        pass: 'XXXX'
    }
});

var mailOptions = {
    from: 'thegoldenspoon.tgs@gmail.com',
    to: receiver,
    subject: sub,
    html: '<h3 style="font-family: verdana; text-align: center;">' + msg + '</h3>'
};

transporter.sendMail(mailOptions, function(error, info){
    if (error) {
        console.log(error);
        return false;
    } else {
        console.log('Email sent: ' + info.response);
        console.log('Email sent to: ' + receiver);
        return true;
    }
});
};

module.exports.sendMail = sendMail;

```

Fetching the menu:

```

fetchMenu = function(req, res){
    models.Item.find({}).then(function(data){
        res.render('menu', {menu: data, user: req.session.user});
    });
};
module.exports.fetchMenu = fetchMenu;

```

Printing the menu:

```

<div class="menuHeading" id="m1" onclick="disp1()">
    <span class="meal">SOUPS & SALADS</span>
    
</div>
<div id="items1" style="display:none">
    <ul>
        <% for (item of menu){ %>

```

```

        if(item.meal=='Soups and Salads'){ %>
            <li><%=item.item%> - <input type="number"
name="<%=@item.name%" value="0" min="0" max="5" class="privilege"><span
class="menuItem">Rs.
<%=@item.price%></span><span><br><%=@item.description%></span></li>
                <%}>
            }%>
        </ul>
    </div>

```

#### Toolbar:

```

<div class="toolbar">
    <a href="/index" class="logoLink">
        
        <span class="tgslogo">&nbs The Golden Spoon</span>
    </a>
    <ul class="toolbarList">
        <li><a href="/menu" class="toolbarLink">MENU</a></li><span
class="pipe"> | </span>
        <li><a href="/location"
class="toolbarLink">LOCATION</a></li><span class="pipe"> | </span>
        <li><a href="/reservation"
class="toolbarLink">RESERVATION</a></li><span class="pipe"> | </span>
            <li><a href="/about" class="toolbarLink">ABOUT US</a></li>
        </ul>
        <ul class="toolbarList2">
            <li id="user"><a href="/login" class="toolbarLink">LOGIN &nbs<i
class="fa fa-user"></i></a></li>
        </ul>
    </div>

```

## 7. Conclusion

Restaurant Management of The Golden Spoon was achieved through a simple use of graphical interface menu options for ordering food and booking a table. This web application that was developed automated the day to day activity of a restaurant making it easier for the restaurant as well as the customers. This facility helps increase the ease of ordering and booking for customers without directly interacting with the restaurant.