VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

RESTAURANT MANAGEMENT SYSTEM

Report submitted for the
Final Project Review of

Course Code: CSE3002 – Internet and Web Programming
Slot: G1

Professor: Lydia Jane

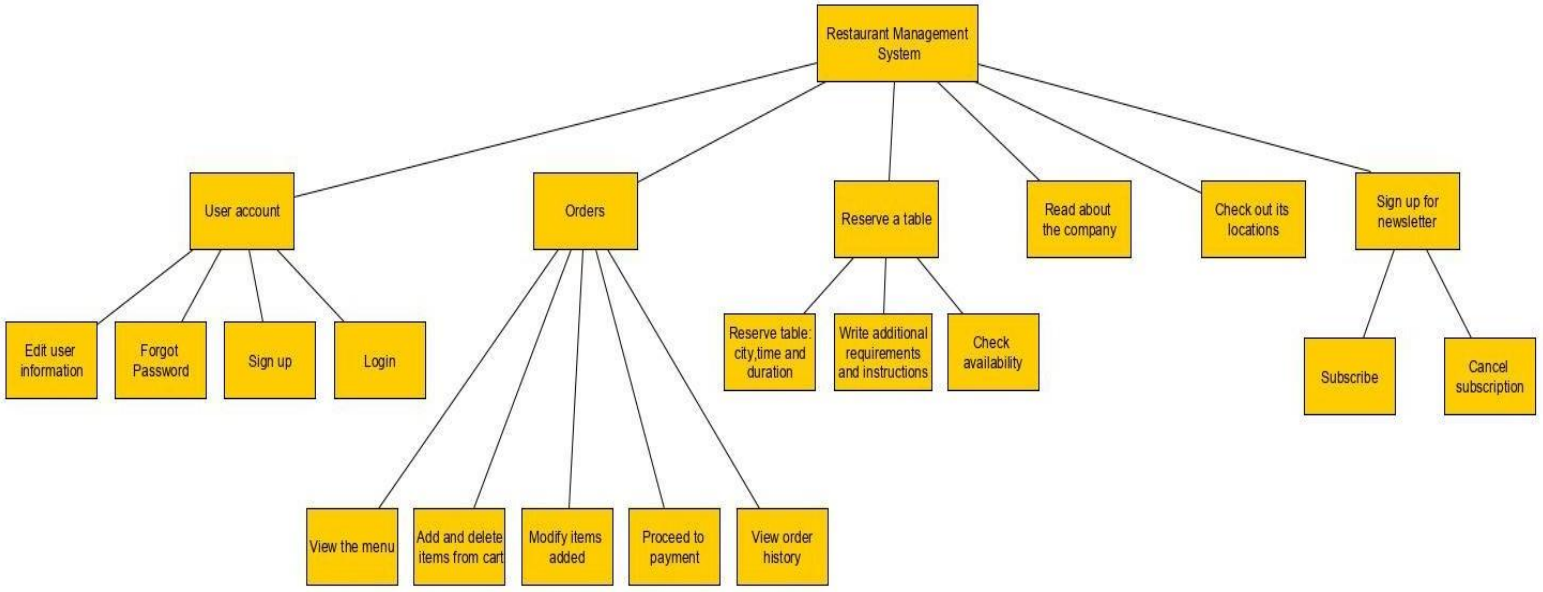
Contents:

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 is 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.

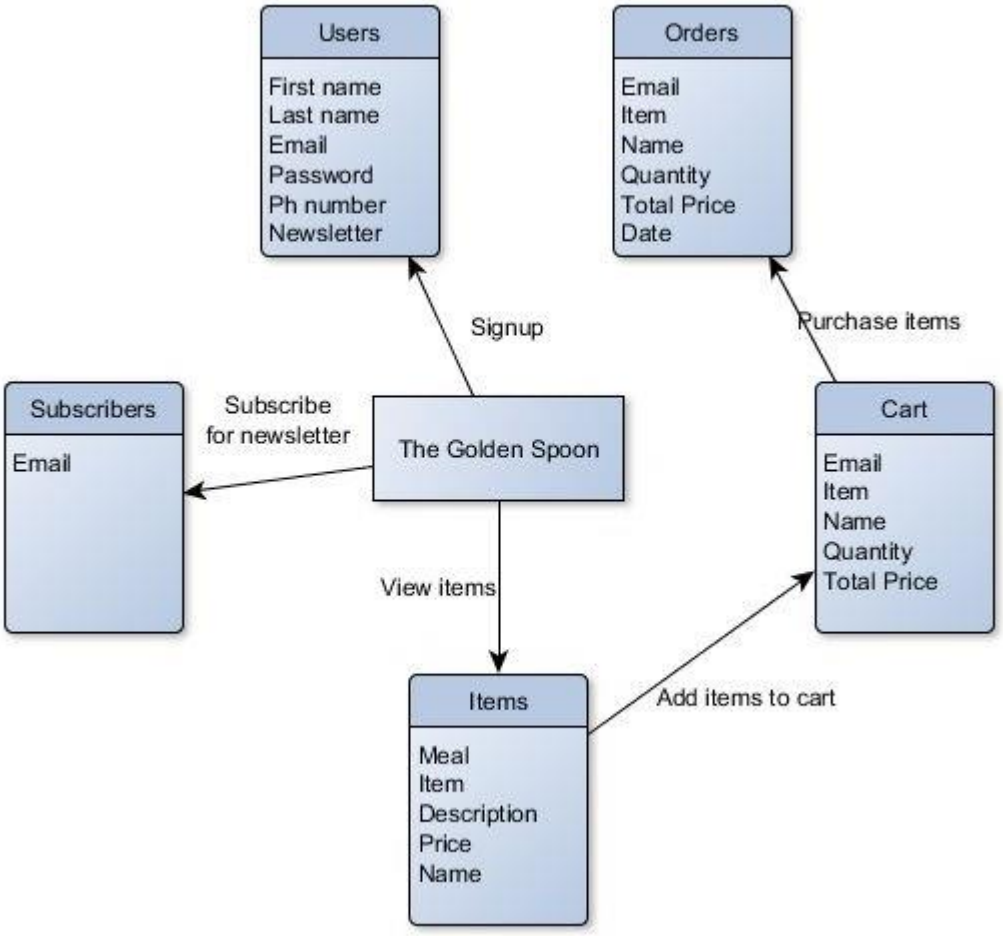
- JavaScript:
  It is a client side scripting language used to perform arithmetic, logical and many more operations on the clients browser. Using DOM, HTML elements can be

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



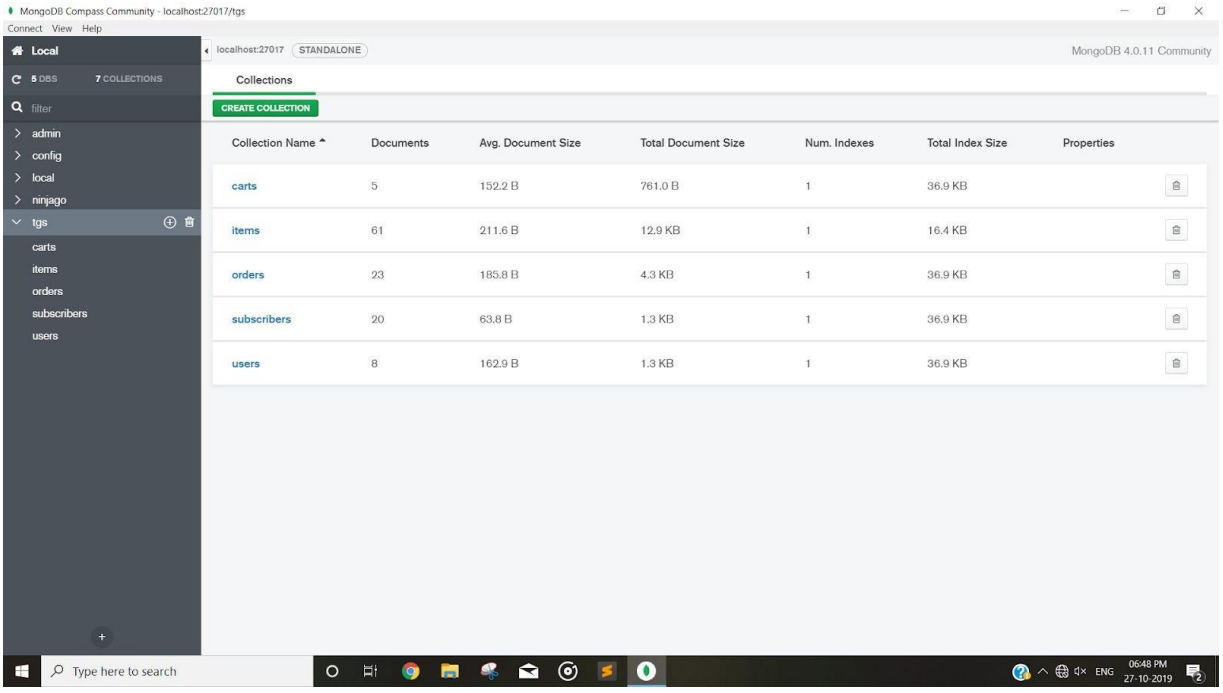
MongoDB is used for maintaining the database for The Golden Spoon. MongoDB is a noSQL database which is based on documents and collections. It makes handling data in the database easier than using SQL. MongoDB Compass is a user interface for MongoDB. Mongoose is a Node.js framework used for handling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

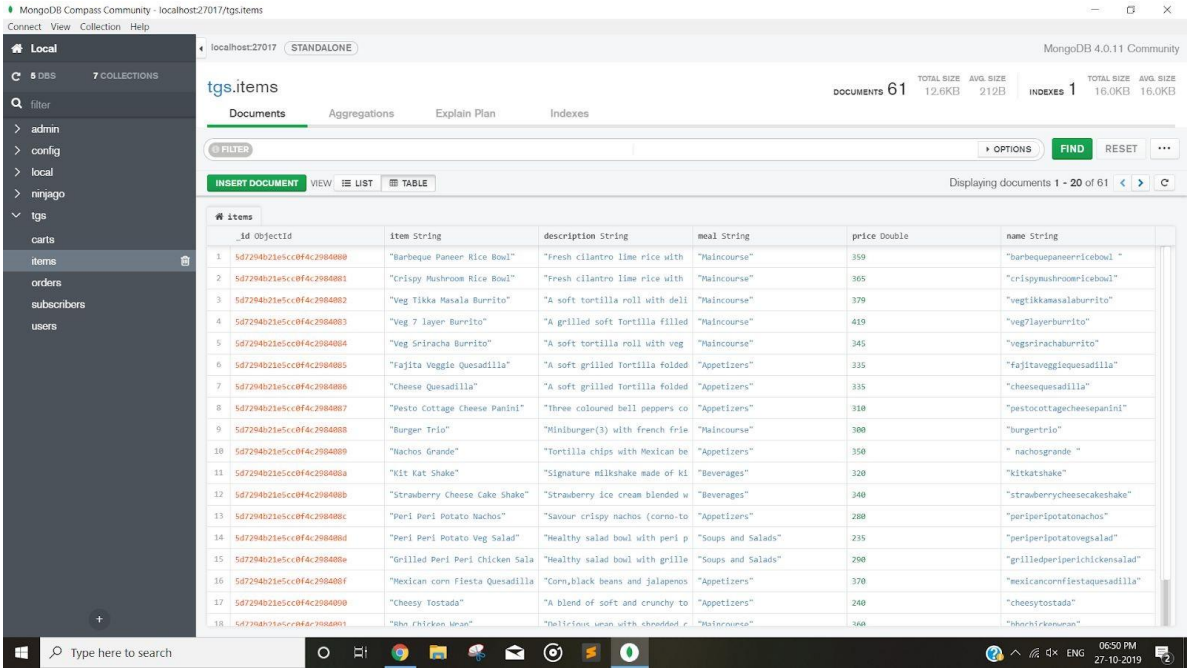
Given below is the schema used:

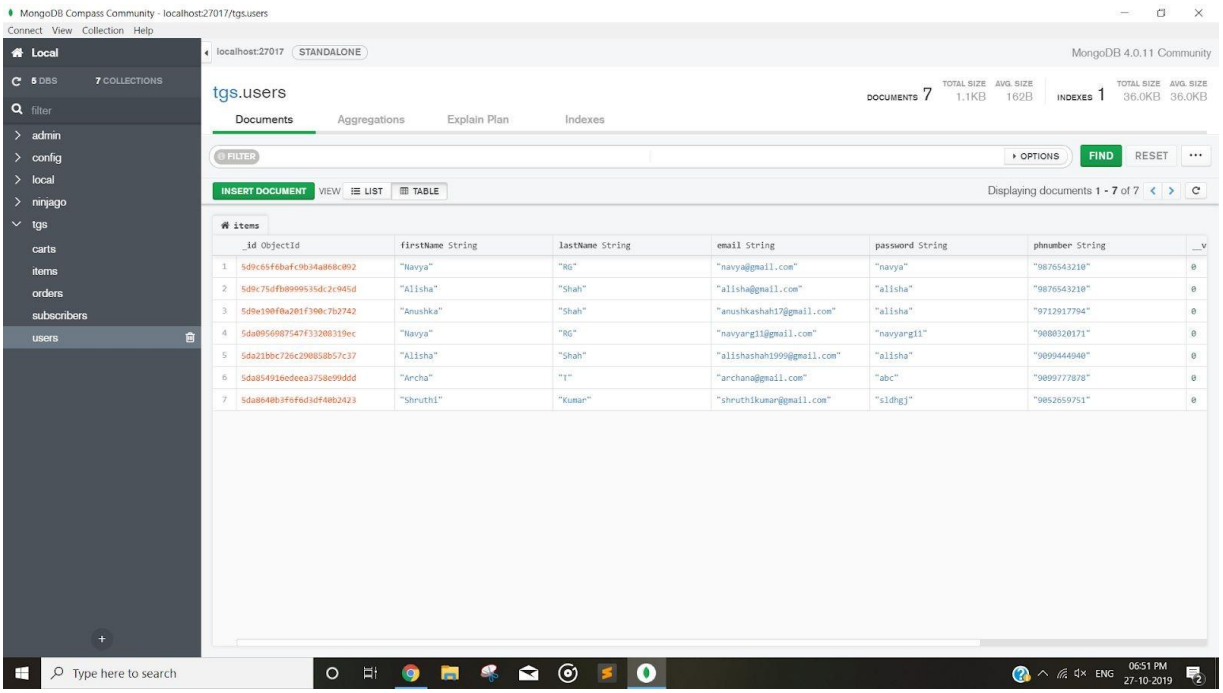
```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

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

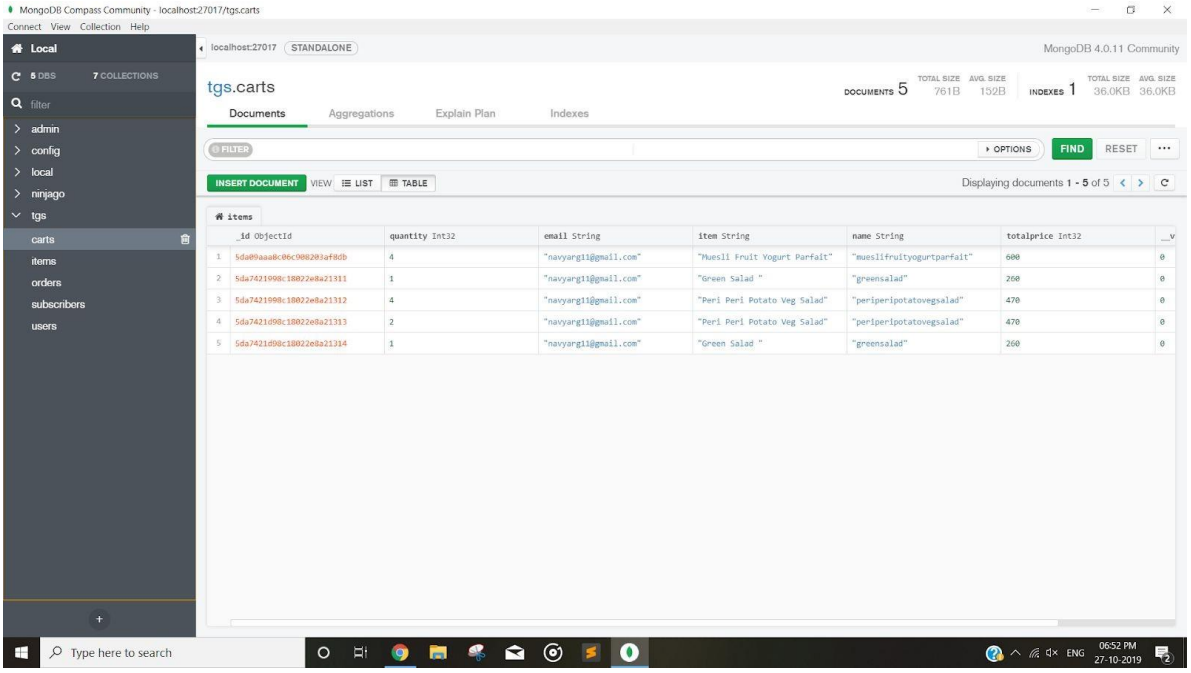
```
const Item = mongoose.model('items', ItemSchema);
module.exports.Item = Item;

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
});
```

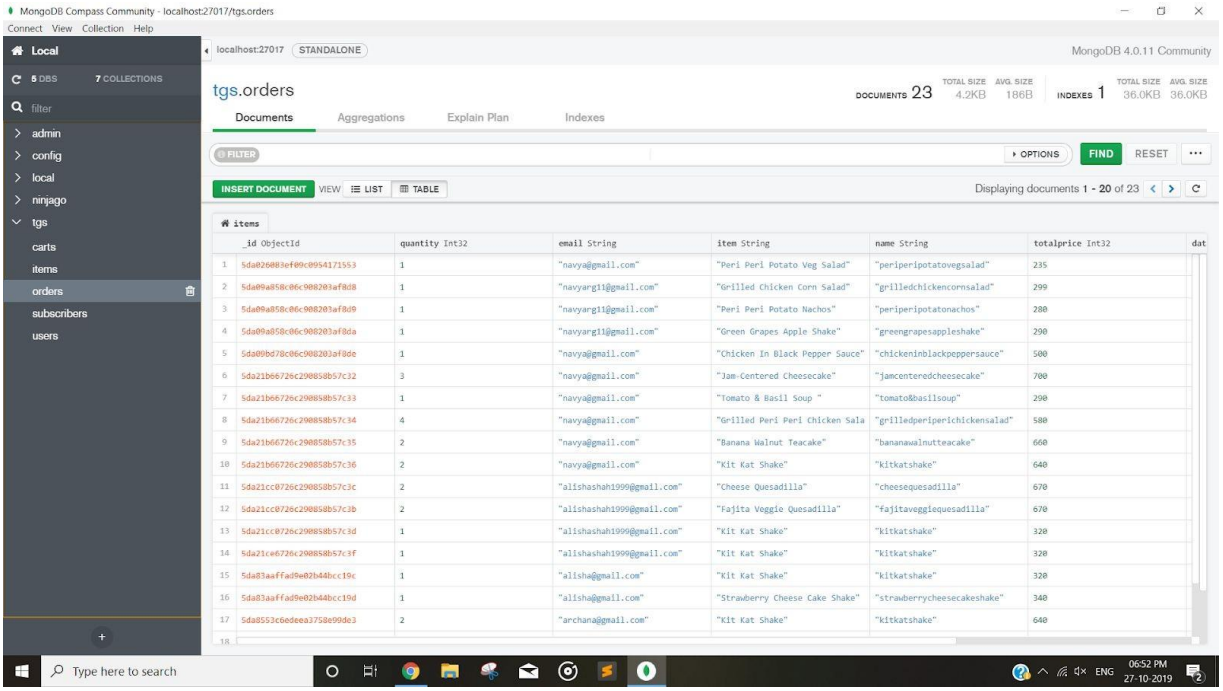
```
const Order = mongoose.model('orders', OrderSchema);
module.exports.Order = Order;
```

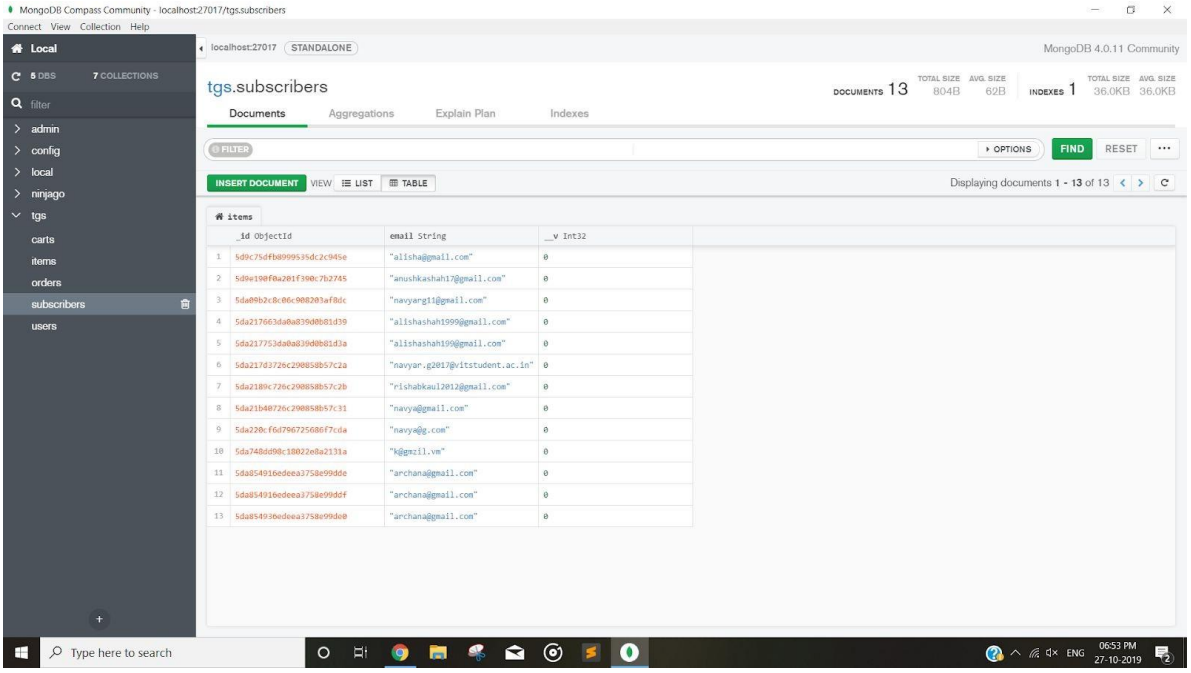




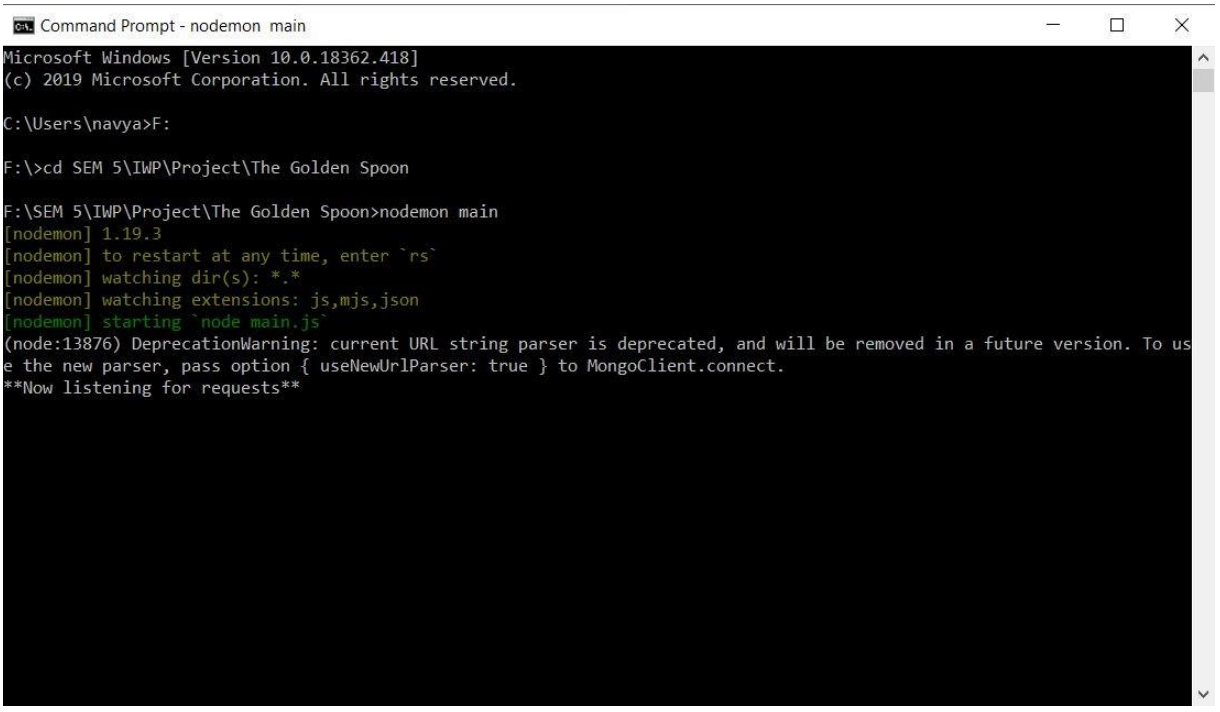

MongoDB Compass Community - localhost:27017/tgs.users
Connect View Collection Help
Local
5 DBS 7 COLLECTIONS
filter
admin
config
local
ninjago
tgs
carts
items
orders
subscribers
users
localhost:27017 STANDALONE
MongoDB 4.0.11 Community
tgs.users
DOCUMENTS 7 TOTAL SIZE 1.1KB AVG. SIZE 162B
INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB
Documents Aggregations Explain Plan Indexes
FILTER OPTIONS FIND RESET
INSERT DOCUMENT VIEW LIST TABLE
Displaying documents 1 - 7 of 7
items
_id ObjectId firstName String lastName String email String password String phnumber String _v
1 5d9c65f6bafc9b34a868c892 "Navya" "RG" "navya@gmail.com" "navya" "9876543210" 0
2 5d9c75dfb8999535dc2c945d "Alisha" "Shah" "alisha@gmail.com" "alisha" "9876543210" 0
3 5d9e190f0a201f390c7b2742 "Anushka" "Shah" "anushkashah17@gmail.com" "alisha" "9712917794" 0
4 5da0956987547f33208319ec "Navya" "RG" "navyarg11@gmail.com" "navyarg11" "9080320171" 0
5 5da21bbc726c290858b57c37 "Alisha" "Shah" "alishashah1999@gmail.com" "alisha" "9099444940" 0
6 5da854916edeea3758e99ddd "Archa" "T" "archana@gmail.com" "abc" "9099777878" 0
7 5da8648b3f6f6d3df40b2423 "Shruthi" "Kumar" "shruthikumar@gmail.com" "sldhgj" "9052659751" 0
Type here to search
ENG 06:51 PM 27-10-2019

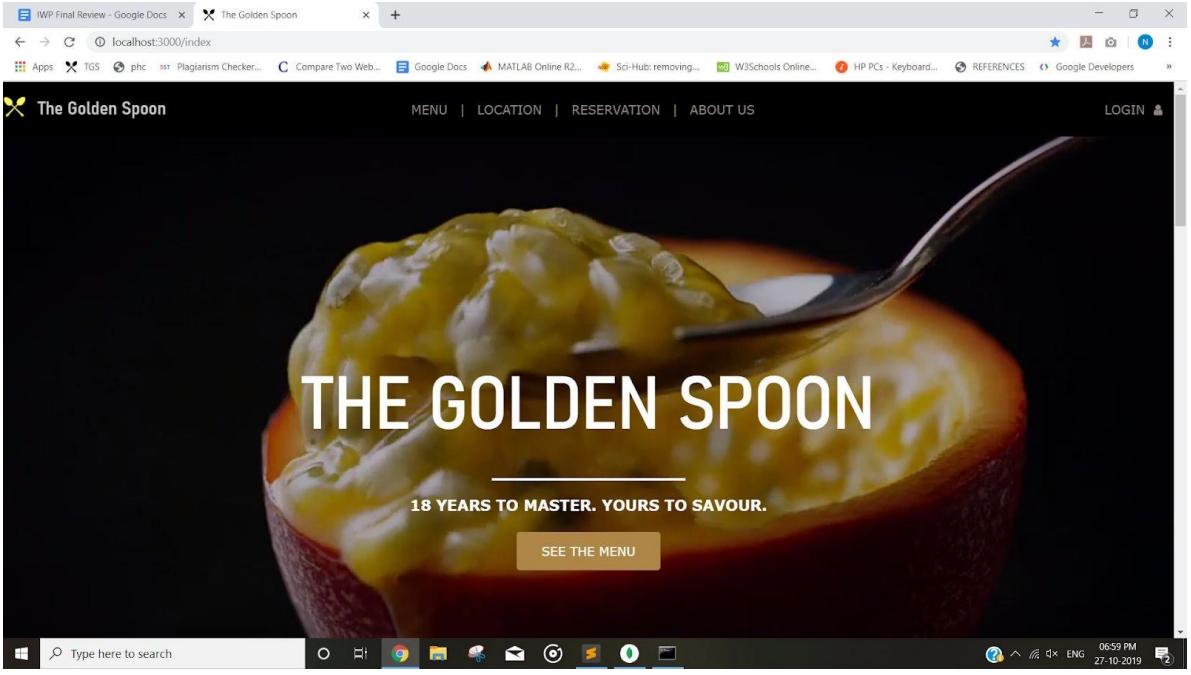

MongoDB Compass Community - localhost:27017/tgs.carts
Connect View Collection Help
Local
5 DBS 7 COLLECTIONS
filter
admin
config
local
ninjago
tgs
carts
items
orders
subscribers
users
localhost:27017 STANDALONE
MongoDB 4.0.11 Community
tgs.carts
DOCUMENTS 5 TOTAL SIZE 761B AVG. SIZE 152B
INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB
Documents Aggregations Explain Plan Indexes
FILTER OPTIONS FIND RESET
INSERT DOCUMENT VIEW LIST TABLE
Displaying documents 1 - 5 of 5
items
_id ObjectId quantity Int32 email String item String name String totalprice Int32 _v
1 5da89aaa8c06c908203af8db 4 "navyarg11@gmail.com" "Muesli Fruit Yogurt Parfait" "mueslifruityogurtparfait" 600 0
2 5da7421998c18022e8a21311 1 "navyarg11@gmail.com" "Green Salad " "greensalad" 260 0
3 5da7421998c18022e8a21312 4 "navyarg11@gmail.com" "Peri Peri Potato Veg Salad" "periperipotatovegsalad" 470 0
4 5da7421d98c18022e8a21313 2 "navyarg11@gmail.com" "Peri Peri Potato Veg Salad" "periperipotatovegsalad" 470 0
5 5da7421d98c18022e8a21314 1 "navyarg11@gmail.com" "Green Salad " "greensalad" 260 0
Type here to search
ENG 06:52 PM 27-10-2019


MongoDB Compass Community - localhost:27017/tgs.orders
Connect View Collection Help
Local
5 DBS 7 COLLECTIONS
filter
admin
config
local
ninjago
tgs
carts
items
orders
subscribers
users
localhost:27017 STANDALONE
MongoDB 4.0.11 Community
tgs.orders
DOCUMENTS 23 TOTAL SIZE 4.2KB AVG. SIZE 186B
INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB
Documents Aggregations Explain Plan Indexes
FILTER OPTIONS FIND RESET
INSERT DOCUMENT VIEW LIST TABLE
Displaying documents 1 - 20 of 23
items
_id ObjectId quantity Int32 email String item String name String totalprice Int32
1 5da026083ef09c0954171553 1 "navya@gmail.com" "Peri Peri Potato Veg Salad" "periperipotatovegsalad" 235
2 5da09a858c06c908203af8d8 1 "navyarg11@gmail.com" "Grilled Chicken Corn Salad" "grilledchickencornsalad" 299
3 5da09a858c06c908203af8d9 1 "navyarg11@gmail.com" "Peri Peri Potato Nachos" "periperipotatonachos" 280
4 5da09a858c06c908203af8da 1 "navyarg11@gmail.com" "Green Grapes Apple Shake" "greengrapesappleshake" 290
5 5da09bd78c06c908203af8de 1 "navya@gmail.com" "Chicken In Black Pepper Sauce" "chickeninblackpeppersauce" 500
6 5da21b66726c290858b57c32 3 "navya@gmail.com" "Jam-Centered Cheesecake" "jamcenteredcheesecake" 700
7 5da21b66726c290858b57c33 1 "navya@gmail.com" "Tomato & Basil Soup " "tomato&basilsoup" 290
8 5da21b66726c290858b57c34 4 "navya@gmail.com" "Grilled Peri Peri Chicken Sala" "grilledperiperichickensalad" 580
9 5da21b66726c290858b57c35 2 "navya@gmail.com" "Banana Walnut Teacake" "bananawalnutteacake" 660
10 5da21b66726c290858b57c36 2 "navya@gmail.com" "Kit Kat Shake" "kitkatshake" 640
11 5da21cc0726c290858b57c3c 2 "alishashah1999@gmail.com" "Cheese Quesadilla" "cheesequesadilla" 670
12 5da21cc0726c290858b57c3b 2 "alishashah1999@gmail.com" "Fajita Veggie Quesadilla" "fajitaveggiequesadilla" 670
13 5da21cc0726c290858b57c3d 1 "alishashah1999@gmail.com" "Kit Kat Shake" "kitkatshake" 320
14 5da21ce6726c290858b57c3f 1 "alishashah1999@gmail.com" "Kit Kat Shake" "kitkatshake" 320
15 5da83aaffad9e02b44bcc19c 1 "alisha@gmail.com" "Kit Kat Shake" "kitkatshake" 320
16 5da83aaffad9e02b44bcc19d 1 "alisha@gmail.com" "Strawberry Cheese Cake Shake" "strawberrycheesecakeshake" 340
17 5da8553c6edeea3758e99de3 2 "archana@gmail.com" "Kit Kat Shake" "kitkatshake" 640
Type here to search
ENG 06:52 PM 27-10-2019

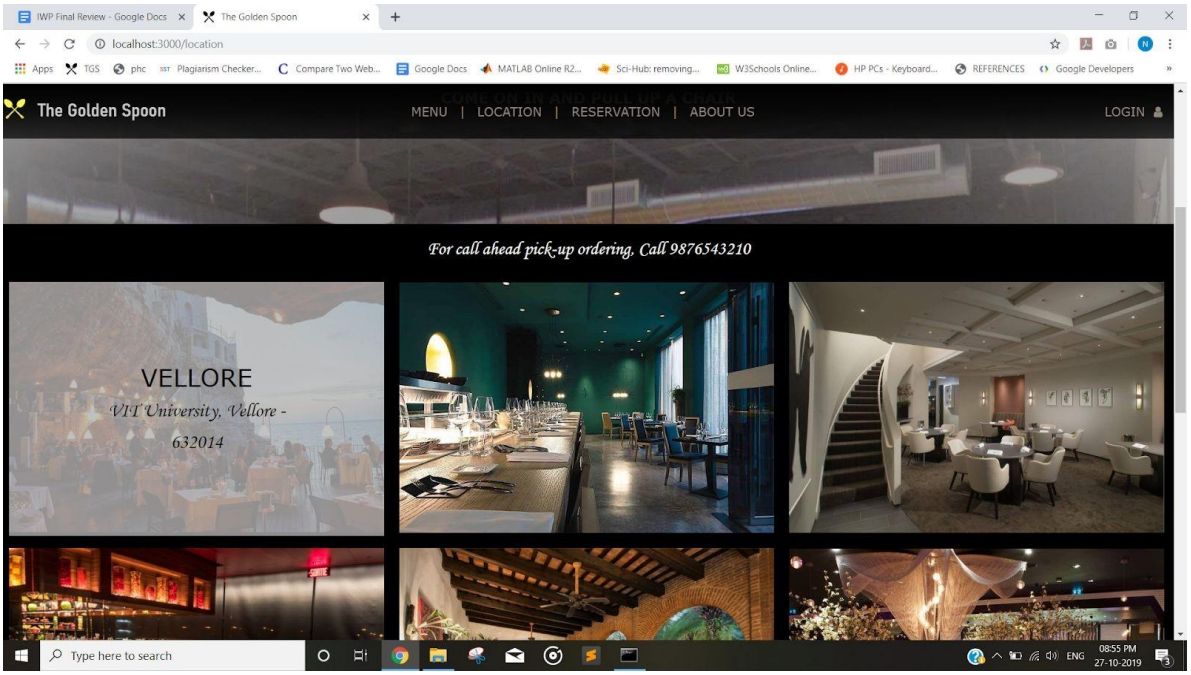

MongoDB Compass Community - localhost:27017/tgs.subscribers
Connect View Collection Help
Local
5 DBS 7 COLLECTIONS
filter
admin
config
local
ninjago
tgs
carts
items
orders
subscribers
users
localhost:27017 STANDALONE
MongoDB 4.0.11 Community
tgs.subscribers
DOCUMENTS 13 TOTAL SIZE 804B AVG. SIZE 62B
INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB
Documents Aggregations Explain Plan Indexes
FILTER OPTIONS FIND RESET
INSERT DOCUMENT VIEW LIST TABLE
Displaying documents 1 - 13 of 13
items
_id ObjectId email String __v Int32
1 5d9c75dfb8999535dc2c945e "alisha@gmail.com" 0
2 5d9e190f0a201f300c7b2745 "anushkashah17@gmail.com" 0
3 5da09b2c8c06c908203af8dc "navyarg11@gmail.com" 0
4 5da217663da0a839d0b81d39 "alishashah1999@gmail.com" 0
5 5da217753da0a839d0b81d3a "alishashah199@gmail.com" 0
6 5da217d3726c290858b57c2a "navyar.g2017@vitstudent.ac.in" 0
7 5da2189c726c290858b57c2b "rishabkaul2012@gmail.com" 0
8 5da21b40726c290858b57c31 "navya@gmail.com" 0
9 5da220cf6d796725686f7cda "navya@g.com" 0
10 5da748dd98c18022e8a2131a "k@gmail.vm" 0
11 5da854916edeea3758e99dde "archana@gmail.com" 0
12 5da854916edeea3758e99ddf "archana@gmail.com" 0
13 5da854936edeea3758e99de0 "archana@gmail.com" 0
Type here to search
ENG 06:53 PM 27-10-2019

5. Screenshots of working model






The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
LOGIN
PEACE
LOVE
4 RIVERS
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


IWP Final Review - Google Docs
The Golden Spoon
localhost:3000/location
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
LOGIN
For call ahead pick-up ordering, Call 9876543210
VELLORE
VIT University, Vellore - 632014
Type here to search
ENG
08:55 PM
27-10-2019

The Golden Spoon

OUR BEGINiNGS

IT ALL STARTED IN A MRAOE

"TGS had its humble beginnings. Just me, a smoker, andan age-of fhOWnge roofing 6ris§et."

John Rivers never set out to creale the most successful BBQ chain in Florida. He's not z classically trained chef or graduate of culinary schr'ol. ii was ie Tesas ttat he found the tu'o loves of his life: his wife and the delicate an ct brisket. And it's heen those very two

John spent 20 yrars in the healthcare industry before retiring as president of a bi11ion•dollar coaipaoy. Duziog that tiznu, be had the ugportunity o irsuuI the couatry, buaulg his lash ano talent for brishet perfection. in he soaked up nt flavors end praoices of 'cut fare countryxride, John's dream of de-regioealizing BBQ was born.

FUN FACT:

// /* **In short, it represents our family, John, Monica (wife), Pared (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 !**

*"his*eras never supposed to be a restaurant. W/ &e nition o Spi meant Haig what eat most natural when cam ort was in order ee&tkose in need."

wFs SmnkeI\one beyan witl› the l8unch oiche "i{arbecue fvtinistry" in z0n4, 4en John

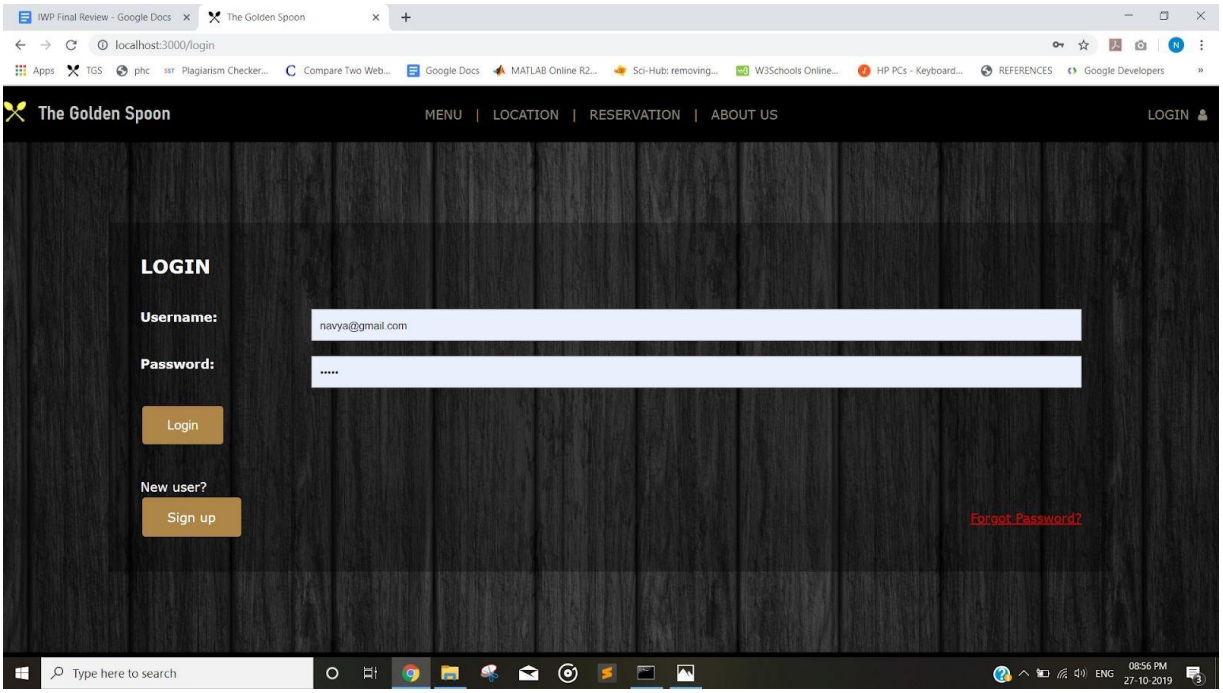
cancer. ThiS one event resulted in a passion for supporting local schools, clnirchm and charitable organizations, and a few years later, smoking thousands of pounds of meai out of a gray just wouldn '1 cut it anymore.

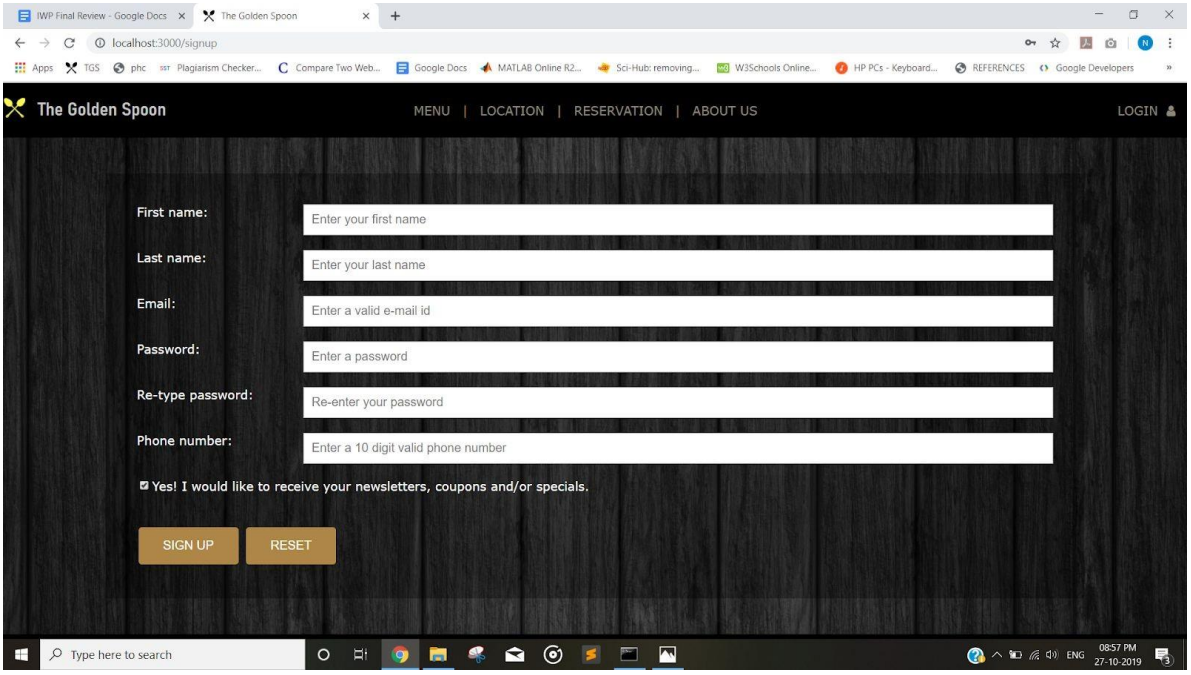
After triels, tribiilntioas, and a whet* lot of bwnt ends. We finally flipped thet "Hot Brisket Now" sign on in £Rtoher of 2009. Within the first hour, a line had frirmed nut tLe **door...** and then amund tne cornen

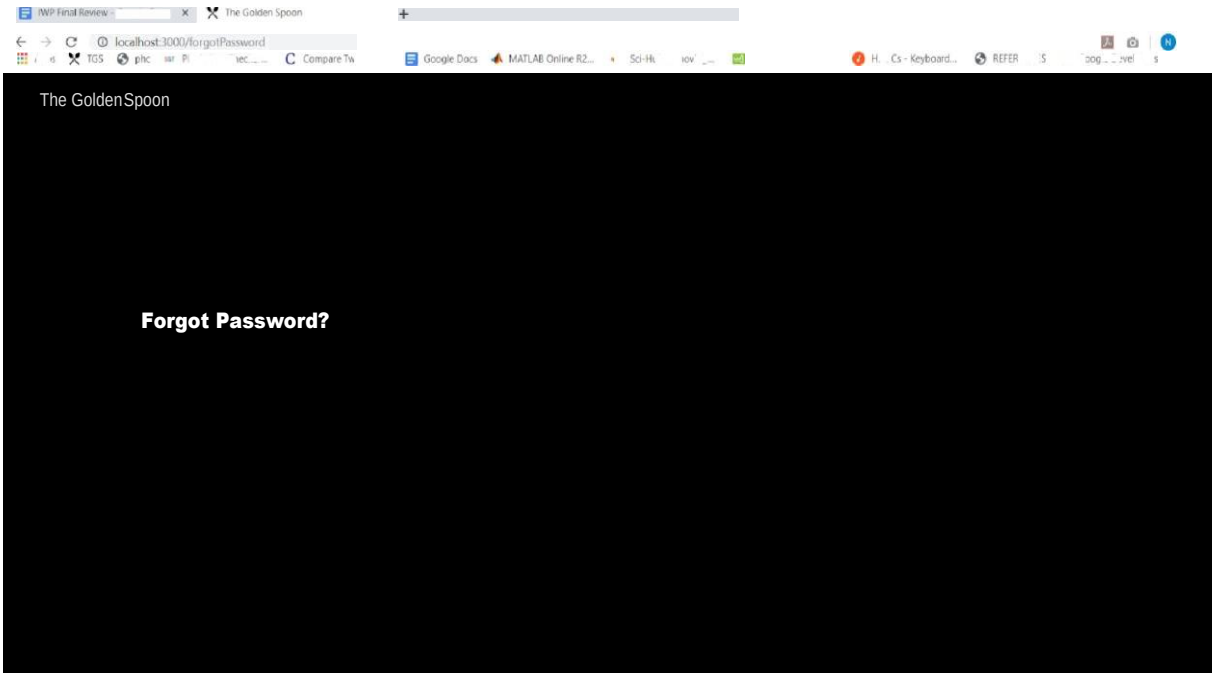
Today, » e hash several locations across the state of Florida, with plans for continued expansion and most tlsuusands 1,000 arctazing tiilLs on our IGS team. Musl importantly, the Barbecue iviinisuy remains otir timndetion end focus, and will until the cows come

SIGN UP FOR OUR NEWSLETTER!

EMAIL* SUBSCRIBE


IWP Final Review - Google Docs
The Golden Spoon
localhost:3000/login
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
LOGIN
LOGIN
Username:
navya@gmail.com
Password:
Login
New user?
Sign up
Forgot Password?
Type here to search
ENG
08:56 PM
27-10-2019


IWP Final Review - Google Docs
The Golden Spoon
localhost:3000/signup
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
LOGIN
First name:
Enter your first name
Last name:
Enter your last name
Email:
Enter a valid e-mail id
Password:
Enter a password
Re-type password:
Re-enter your password
Phone number:
Enter a 10 digit valid phone number
Yes! I would like to receive your newsletters, coupons and/or specials.
SIGN UP
RESET
Type here to search
ENG
08:57 PM
27-10-2019


IWP Final Review
The Golden Spoon
localhost:3000/forgotPassword
TGS
phc
Compare Tw
Google Docs
MATLAB Online R2...
Sci-Hi
Cs - Keyboard...
REFER
The GoldenSpoon
Forgot Password?

SOU PS & SALADS

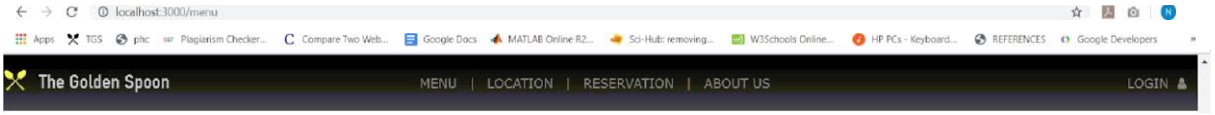
APPETIZE US

MAIN COURSE

BEVERAGES

DENSE RTS

'2IGN UP FOR O\JR NEWSLETTER !



MAIN COURSE 0

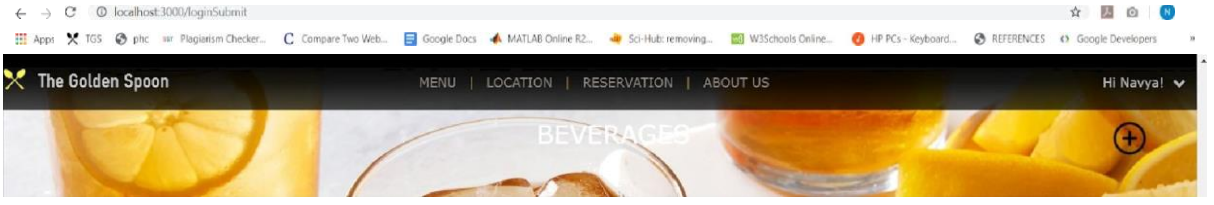
Kit Kat Shake - Rs. 320
Signature milkshake made of kitkat topped with whipped cream and kitkat

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

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

Epigamia Greek Yogurt Smoothie - Alphonso f•'lango Rs. 390
Greek yogurt smoothie made of fresh alphonso mangoes

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



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

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

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

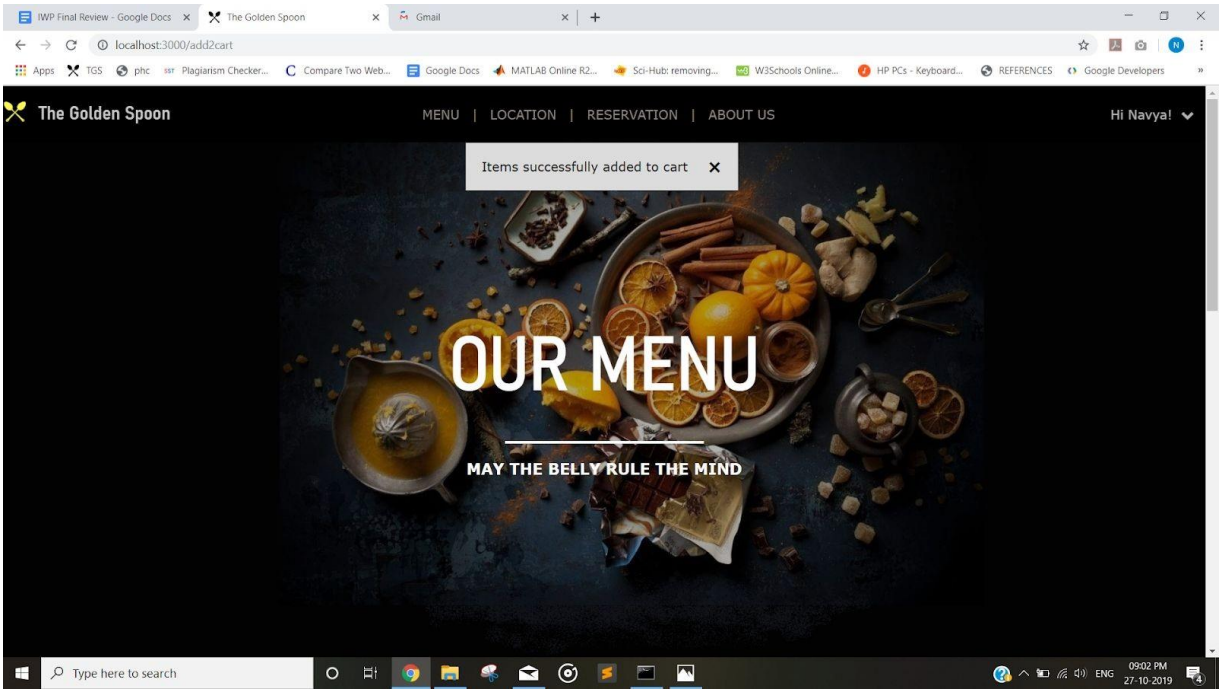
Epigamia Greek Yogurt Smoothie - Alphonso Mango - Rs. 390
Greek yogurt smoothie made of fresh alphonso mangoes

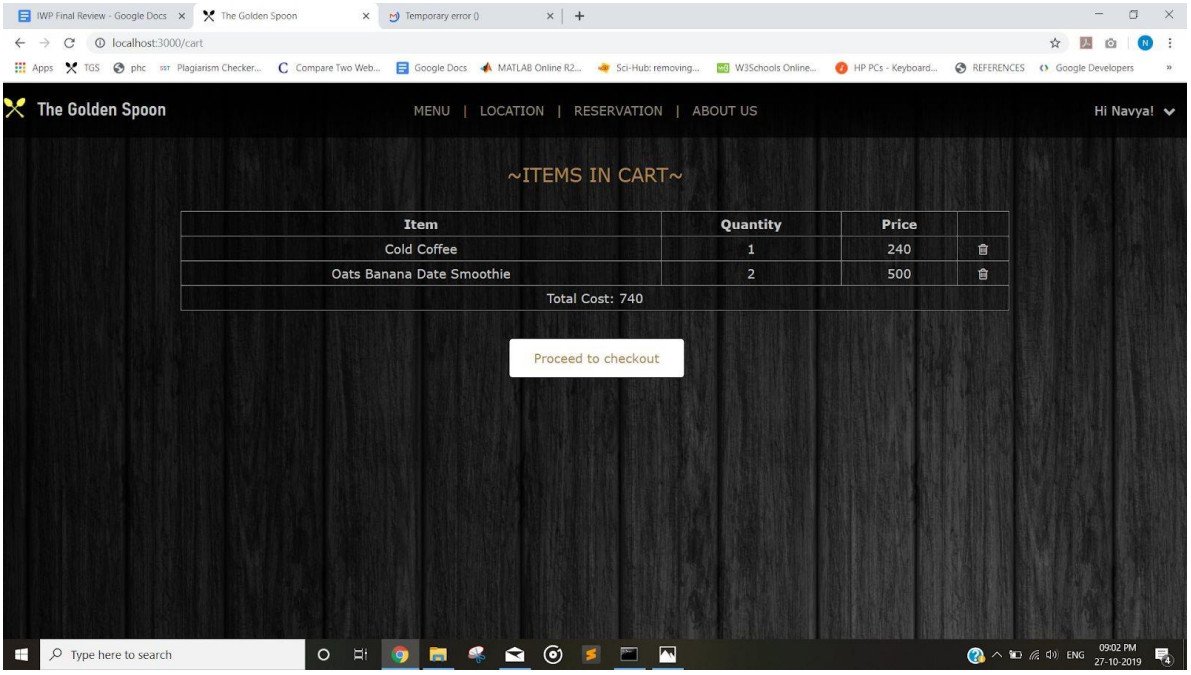
Cold Coffee Rs. 240
Just what you need on a hot, sunny day

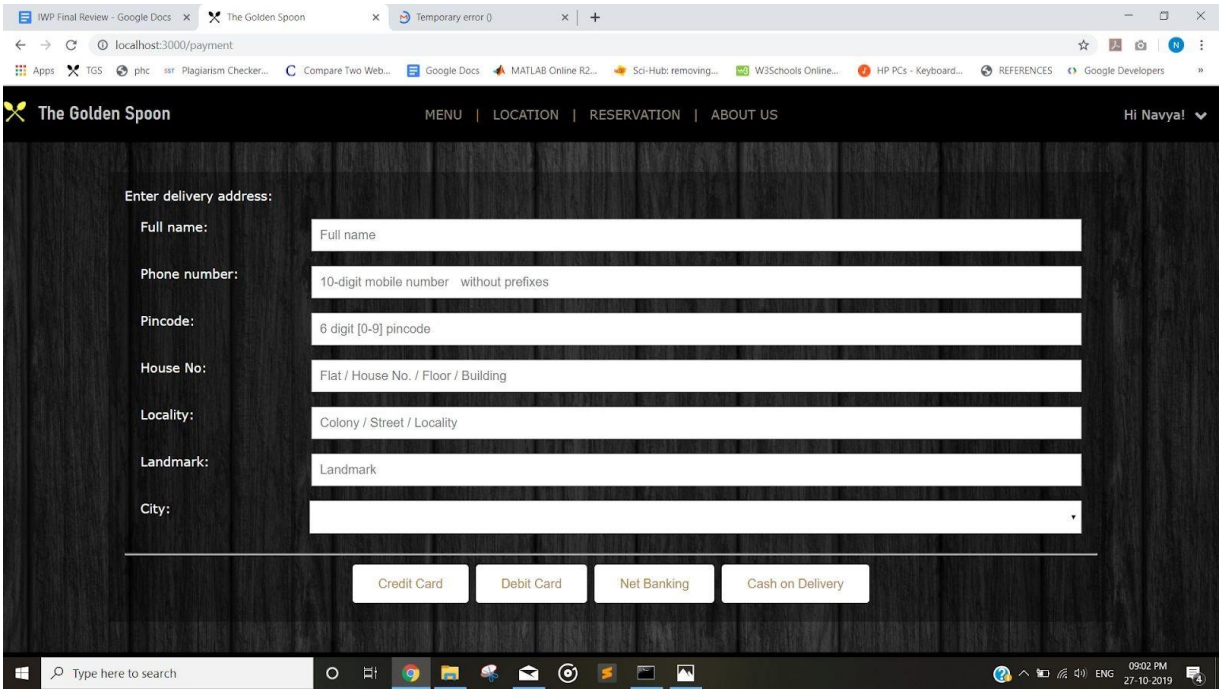
Oats Banana Date Smoothie - Rs. 250
Delicious blend of chopped dates and bananas with a scoop of rolled oats

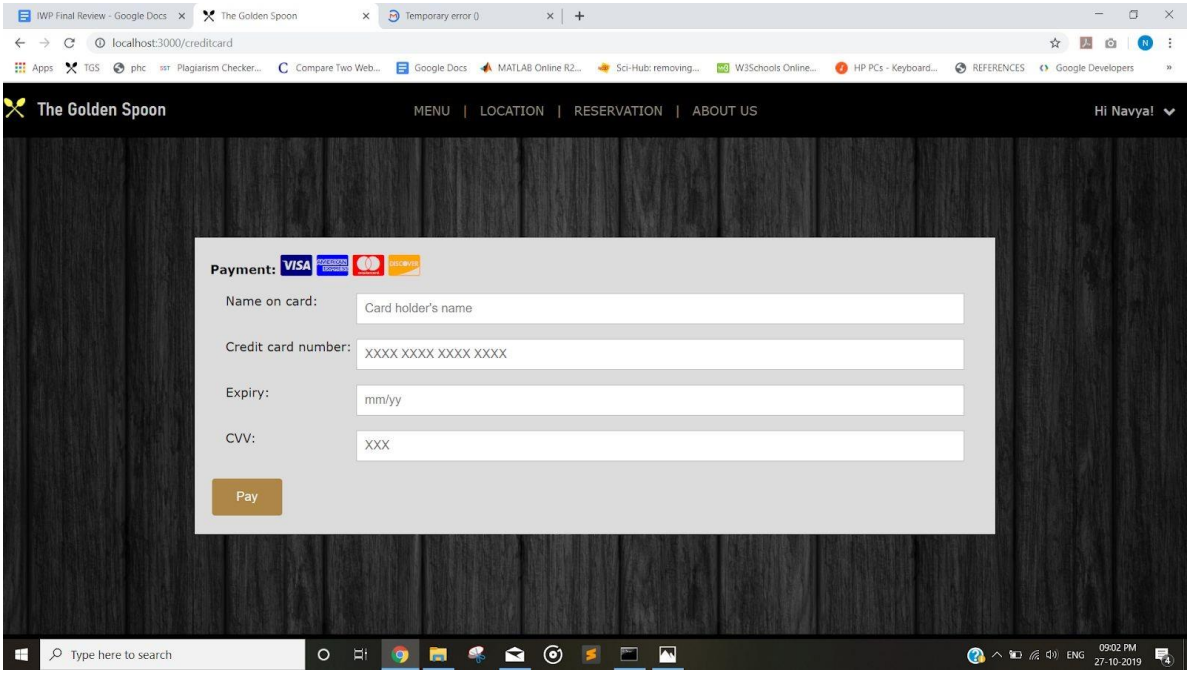
Peanut Butter Banana Shake - Rs.2E0
Sticky peanut butter and chunks of banana blended into a yummy shake.

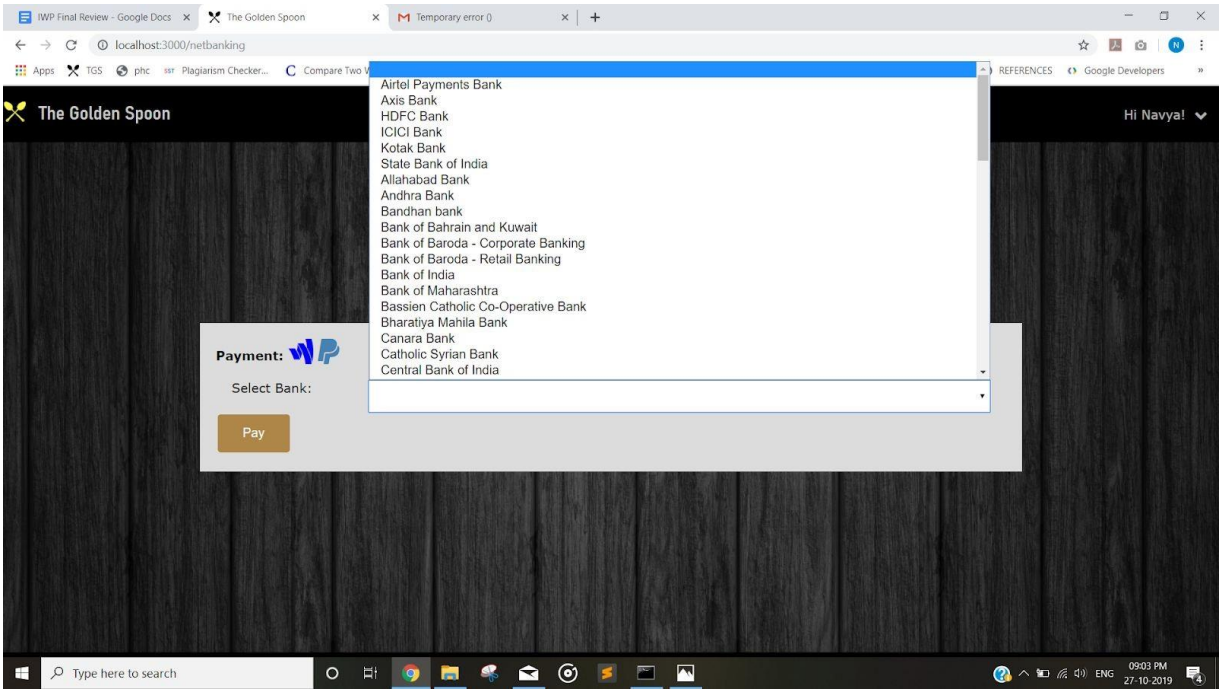
Avocado Banana Super Shake - Rs. 260

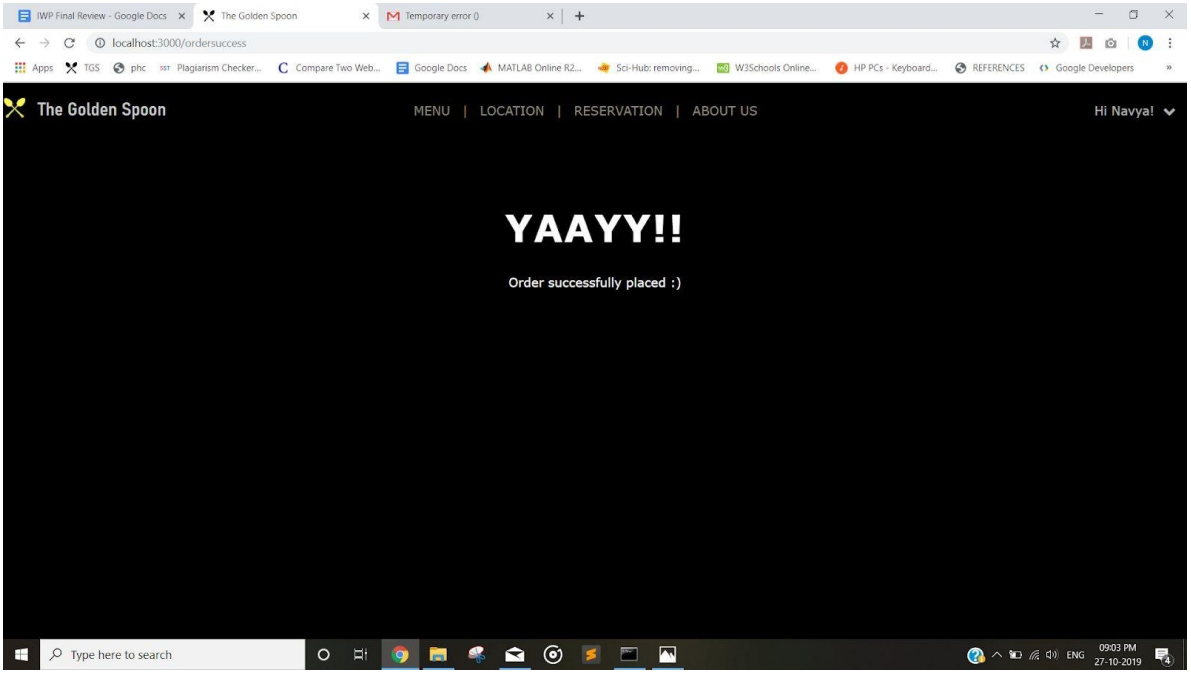

IWP Final Review - Google Docs
The Golden Spoon
Gmail
localhost:3000/add2cart
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
Hi Navya!
Items successfully added to cart
OUR MENU
MAY THE BELLY RULE THE MIND
Type here to search
ENG
09:02 PM
27-10-2019

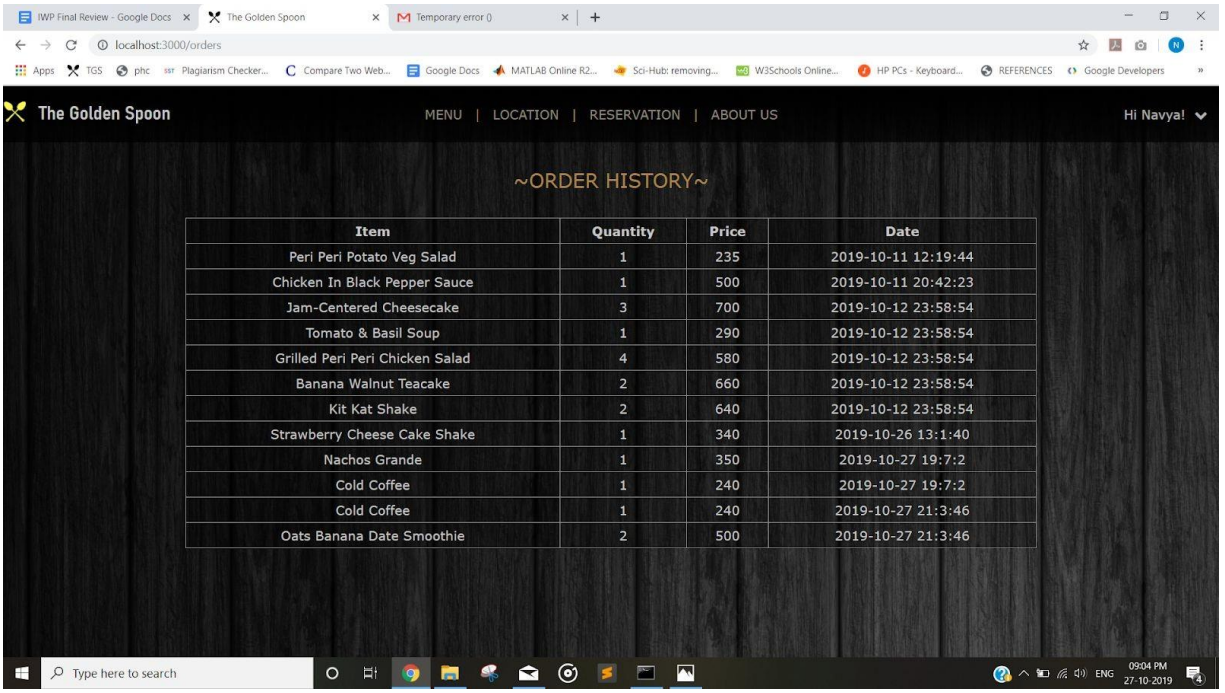

IWP Final Review - Google Docs
The Golden Spoon
Temporary error ()
localhost:3000/cart
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
Hi Navya!
~ITEMS IN CART~
Item
Quantity
Price
Cold Coffee
1
240
Oats Banana Date Smoothie
2
500
Total Cost: 740
Proceed to checkout
Type here to search
ENG
09:02 PM
27-10-2019

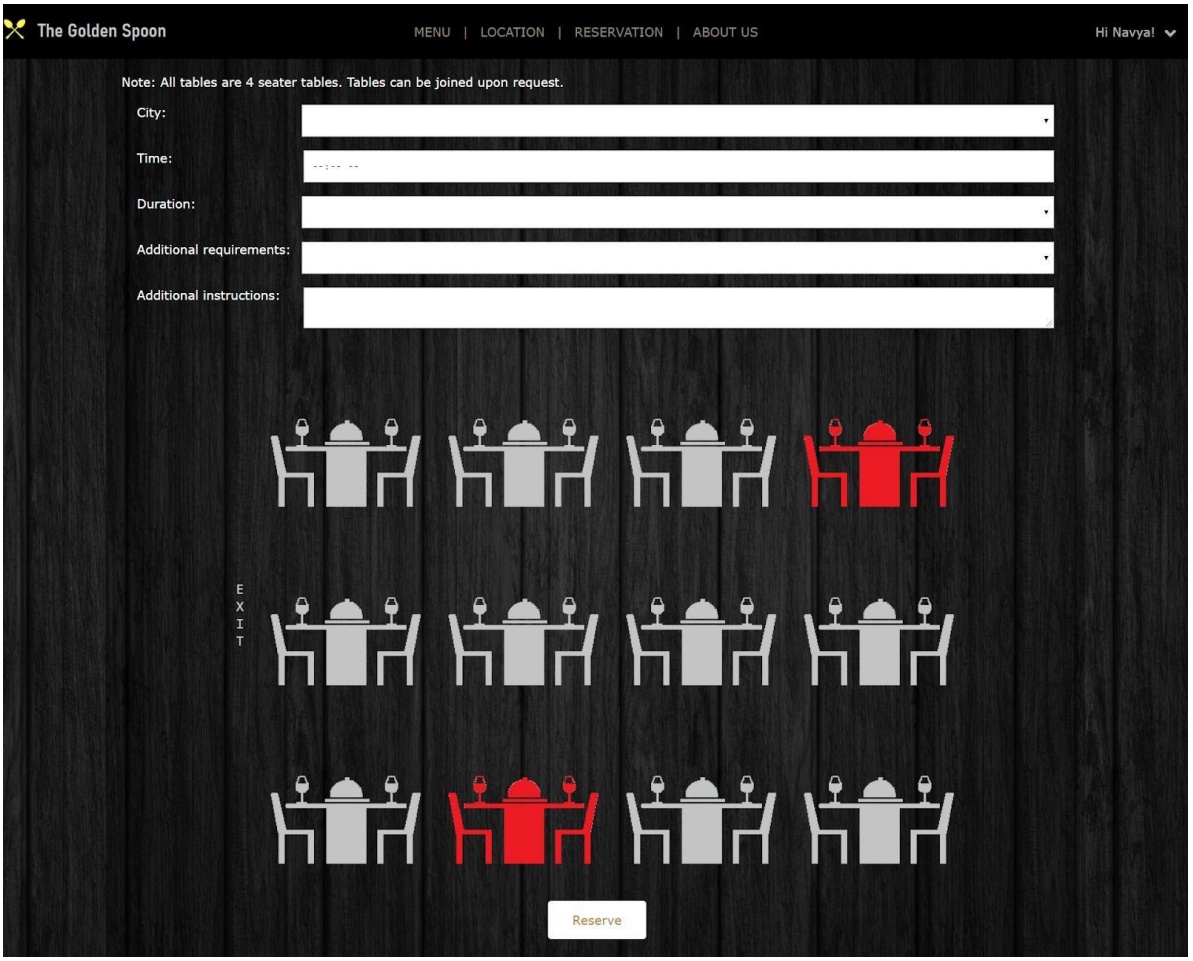

IWP Final Review - Google Docs
The Golden Spoon
Temporary error ()
localhost:3000/payment
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
Hi Navya!
Enter delivery address:
Full name:
Full name
Phone number:
10-digit mobile number without prefixes
Pincode:
6 digit [0-9] pincode
House No:
Flat / House No. / Floor / Building
Locality:
Colony / Street / Locality
Landmark:
Landmark
City:
Credit Card
Debit Card
Net Banking
Cash on Delivery
Type here to search
ENG
09:02 PM
27-10-2019

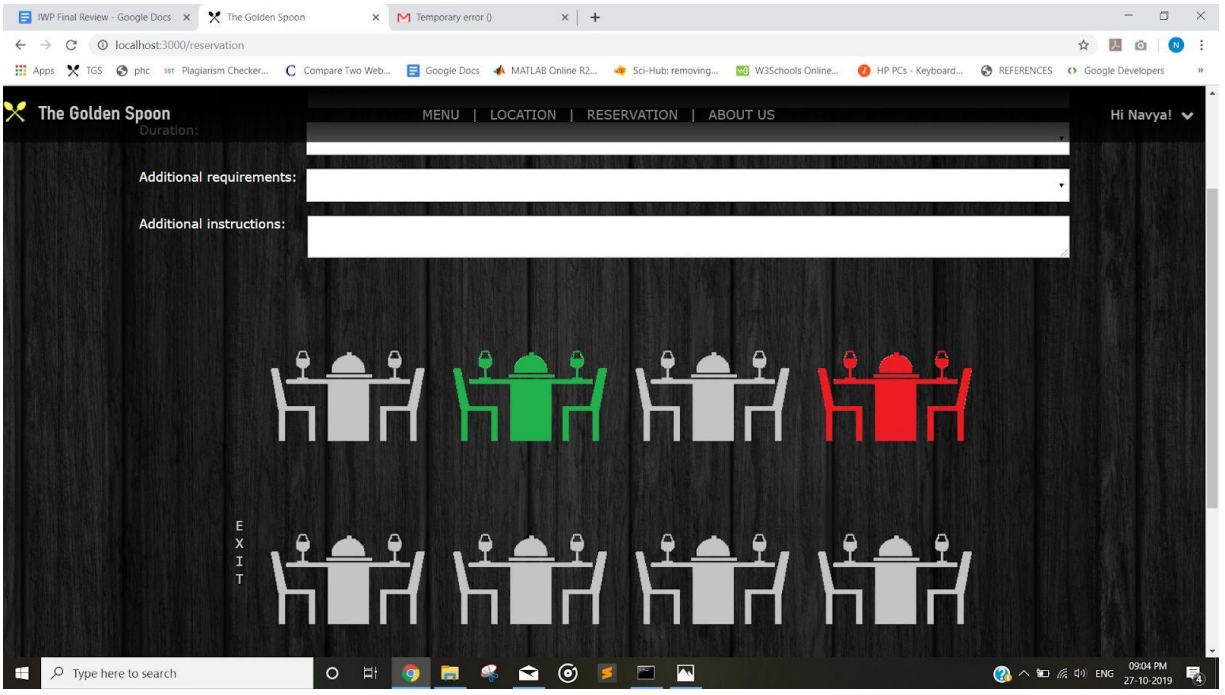

IWP Final Review - Google Docs
The Golden Spoon
Temporary error ()
localhost:3000/creditcard
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
Hi Navya!
Payment:
Name on card:
Card holder's name
Credit card number:
XXXX XXXX XXXX XXXX
Expiry:
mm/yy
CVV:
XXX
Pay
Type here to search
ENG
09:02 PM
27-10-2019

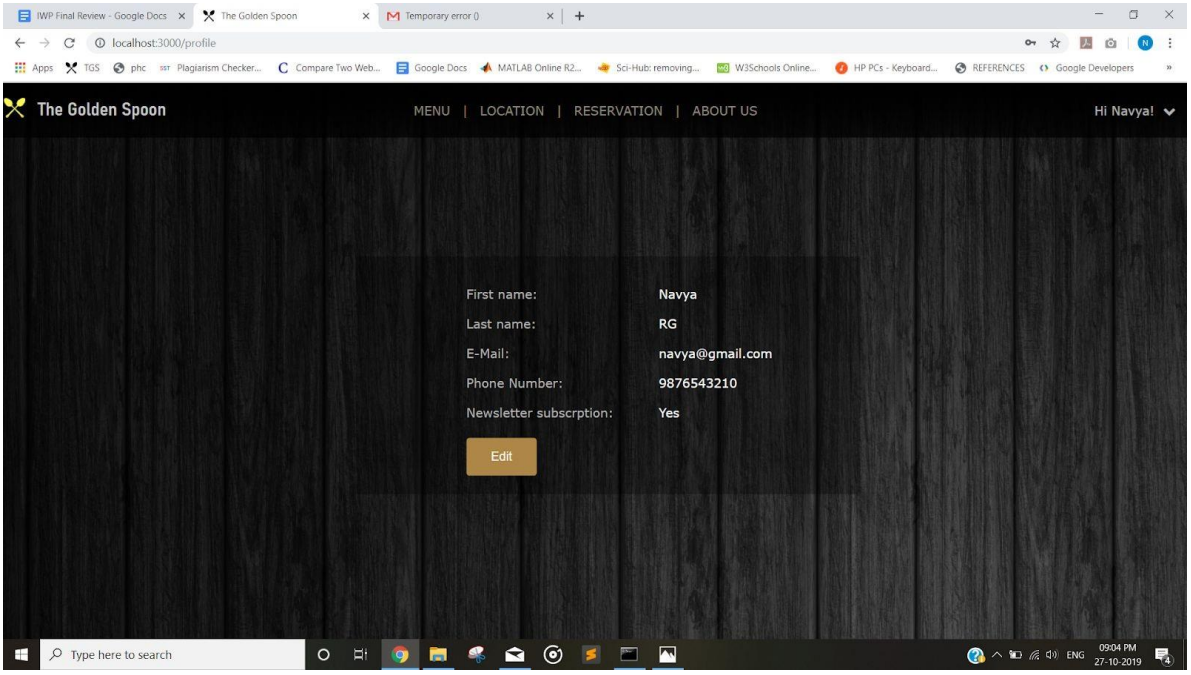

IWP Final Review - Google Docs
The Golden Spoon
Temporary error ()
localhost:3000/netbanking
The Golden Spoon
Hi Navya!
Airtel Payments Bank
Axis Bank
HDFC Bank
ICICI Bank
Kotak Bank
State Bank of India
Allahabad Bank
Andhra Bank
Bandhan bank
Bank of Bahrain and Kuwait
Bank of Baroda - Corporate Banking
Bank of Baroda - Retail Banking
Bank of India
Bank of Maharashtra
Bassien Catholic Co-Operative Bank
Bharatiya Mahila Bank
Canara Bank
Catholic Syrian Bank
Central Bank of India
Payment:
Select Bank:
Pay
Type here to search
ENG
09:03 PM
27-10-2019


IWP Final Review - Google Docs
The Golden Spoon
Temporary error ()
localhost:3000/ordersuccess
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
Hi Navya!
YAAYY!!
Order successfully placed :)
Type here to search
ENG
09:03 PM
27-10-2019


IWP Final Review - Google Docs
The Golden Spoon
Temporary error ()
localhost:3000/orders
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-Centered 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
ENG
09:04 PM
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:
EXIT
Reserve


IWP Final Review - Google Docs
The Golden Spoon
Temporary error ()
localhost:3000/reservation
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
Hi Navya!
Duration:
Additional requirements:
Additional instructions:
EXIT
Type here to search
ENG
09:04 PM
27-10-2019


IWP Final Review - Google Docs
The Golden Spoon
Temporary error ()
localhost:3000/profile
The Golden Spoon
MENU | LOCATION | RESERVATION | ABOUT US
Hi Navya!
First name: Navya
Last name: RG
E-Mail: navya@gmail.com
Phone Number: 9876543210
Newsletter subscrption: Yes
Edit
Type here to search
ENG
09:04 PM
27-10-2019

The Golden Spoon

First name:

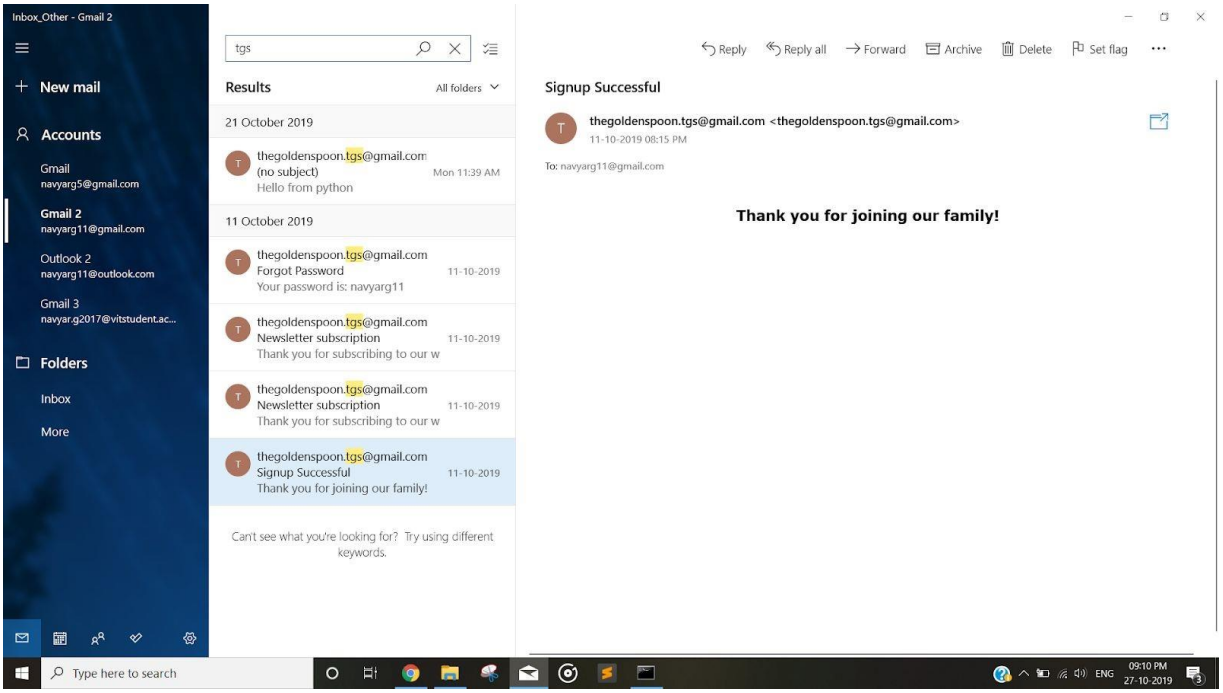
Old Password:

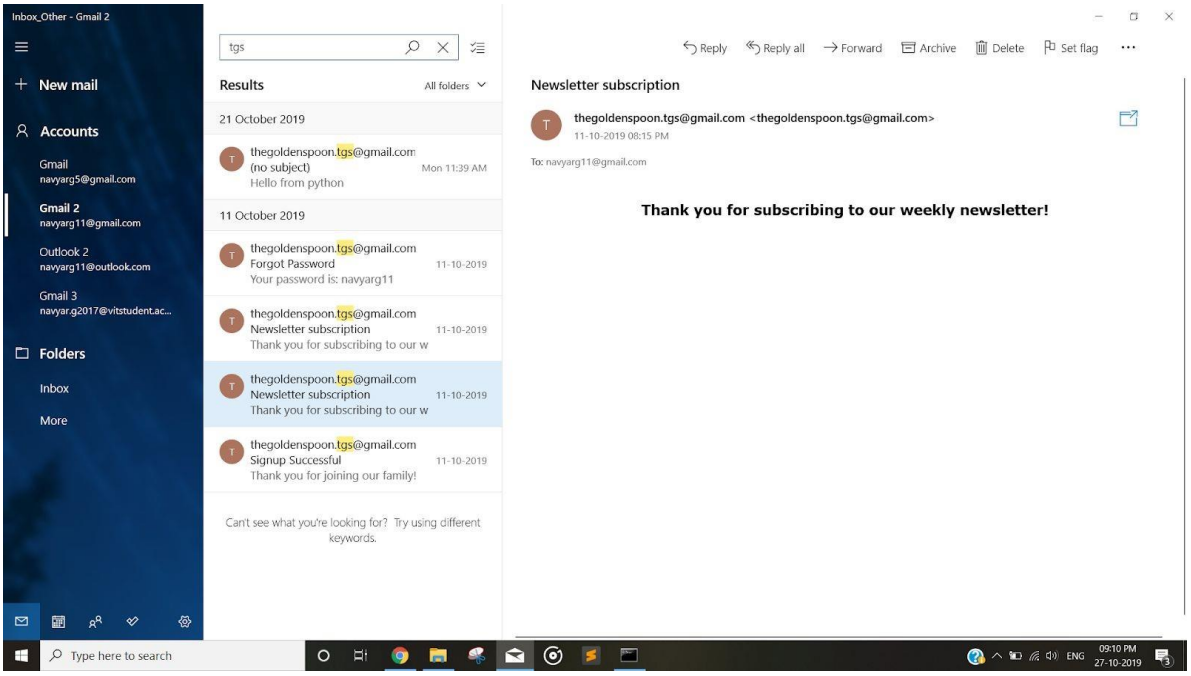
New Password:

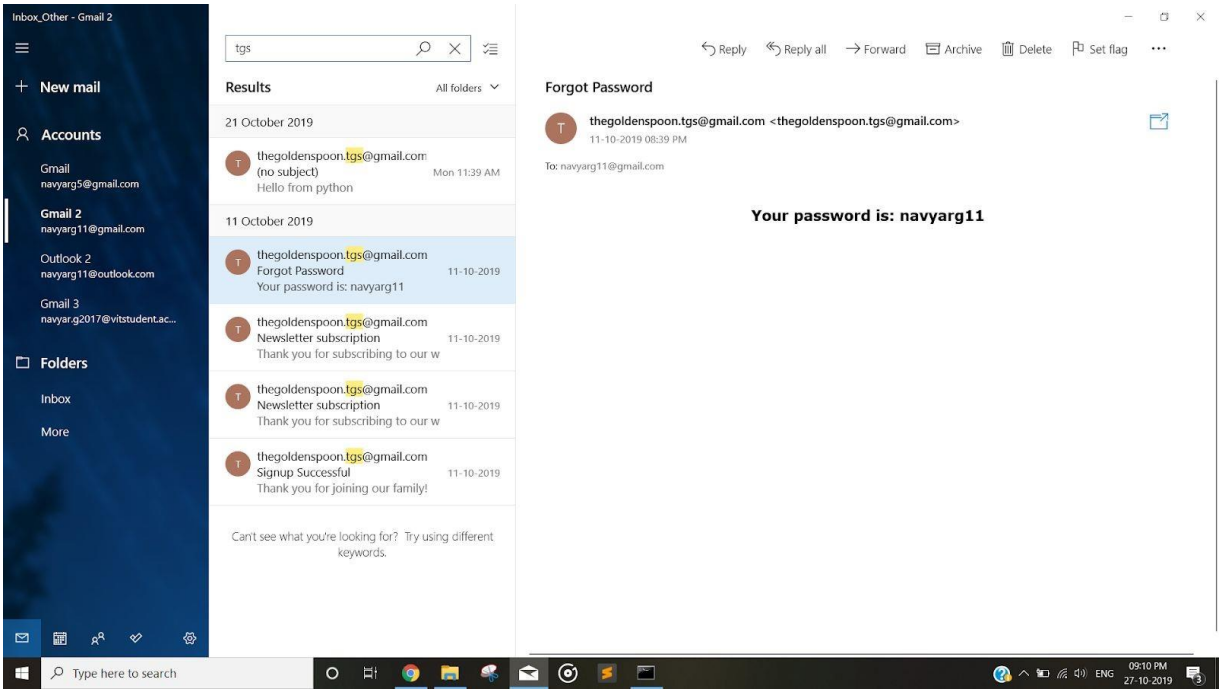
Yes! S would like to receive your newsletters, coupons and/or specials.

The Golden Spoon

404, OOPS!


Inbox_Other - Gmail 2
New mail
Accounts
Gmail
navyarg5@gmail.com
Gmail 2
navyarg11@gmail.com
Outlook 2
navyarg11@outlook.com
Gmail 3
navyar.g2017@vitstudent.ac...
Folders
Inbox
More
tgs
Results
All folders
21 October 2019
thegoldenspoon.tgs@gmail.com
(no subject)
Hello from python
Mon 11:39 AM
11 October 2019
thegoldenspoon.tgs@gmail.com
Forgot Password
Your password is: navyarg11
11-10-2019
thegoldenspoon.tgs@gmail.com
Newsletter subscription
Thank you for subscribing to our w
11-10-2019
thegoldenspoon.tgs@gmail.com
Newsletter subscription
Thank you for subscribing to our w
11-10-2019
thegoldenspoon.tgs@gmail.com
Signup Successful
Thank you for joining our family!
11-10-2019
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: navyarg11@gmail.com
Thank you for joining our family!
Type here to search
ENG
09:10 PM
27-10-2019


Inbox_Other - Gmail 2
New mail
Accounts
Gmail
navyarg5@gmail.com
Gmail 2
navyarg11@gmail.com
Outlook 2
navyarg11@outlook.com
Gmail 3
navyar.g2017@vitstudent.ac...
Folders
Inbox
More
tgs
Results
All folders
21 October 2019
thegoldenspoon.tgs@gmail.com
(no subject)
Hello from python
Mon 11:39 AM
11 October 2019
thegoldenspoon.tgs@gmail.com
Forgot Password
Your password is: navyarg11
11-10-2019
thegoldenspoon.tgs@gmail.com
Newsletter subscription
Thank you for subscribing to our w
11-10-2019
thegoldenspoon.tgs@gmail.com
Newsletter subscription
Thank you for subscribing to our w
11-10-2019
thegoldenspoon.tgs@gmail.com
Signup Successful
Thank you for joining our family!
11-10-2019
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: navyarg11@gmail.com
Thank you for subscribing to our weekly newsletter!
Type here to search
ENG
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>
                        <img src="images/plus.png" class="rotPlus">
                </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="" 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">
                <img class="toolbarImage" src="images/tgstrans2.png">
                <span class="tgslogo">  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  <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.