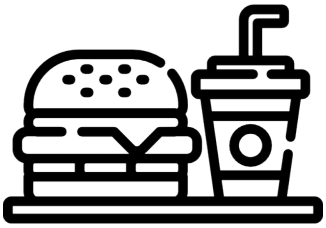
Project report.

CFS

Canteen Food Service.



Submitted by: Daksh Sharma
Roll No: IMT2021533

PROJECT PLAN.

(NOTE: Before running the app.js file please do following steps
1. Make a .env file and paste following api keys in it

```
SECRET=Thisisourlittlesecret
CLIENT_ID=409994263743-8o5astqgn6mtmibbv240cma6uf7rbrso.apps.googleusercontent.com
CLIENT_SECRET=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
```

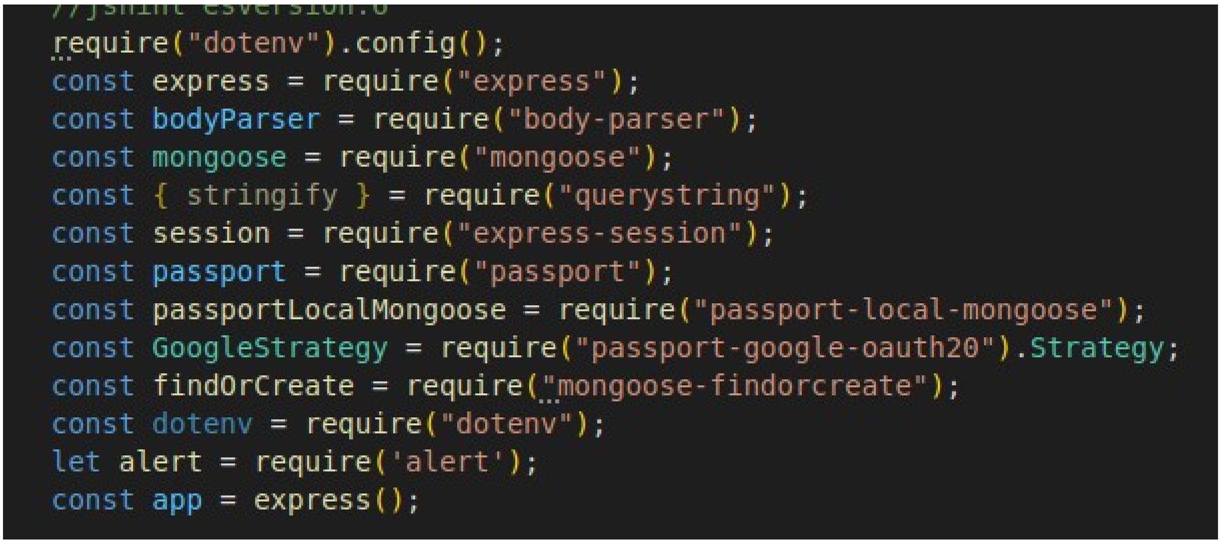
2. Run command npm install to install required modules.)

Major aspect of this project work has been to develop a proper food ordering website which can be used by students to order food from canteen while sitting in their room so as to reduce the waiting time .

I have used node js and mongodb databse to store data of users.
The details such as user info , item list, orders placed have been uploaded on mongodb cloud database.

(Note: before running app.js please do following steps:
1) make a file .env in the current directory and-
)

Modules used:-

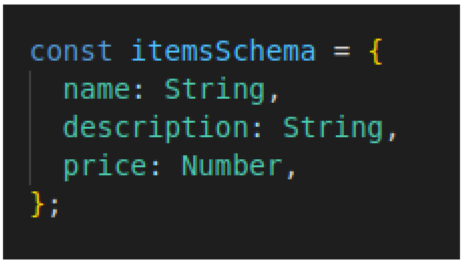


```
//jshint esversion:6
require("dotenv").config();
const express = require("express");
const bodyParser = require("body-parser");
const mongoose = require("mongoose");
const { stringify } = require("querystring");
const session = require("express-session");
const passport = require("passport");
const passportLocalMongoose = require("passport-local-mongoose");
const GoogleStrategy = require("passport-google-oauth20").Strategy;
const findOrCreate = require("mongoose-findorcreate");
const dotenv = require("dotenv");
let alert = require('alert');
const app = express();
```

Database used:-
MongoDB (cloud)

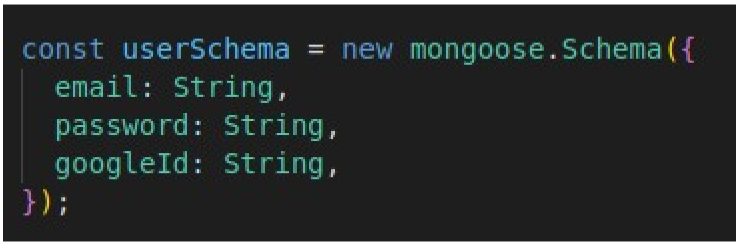
Collections:-
I have made 4 collectioins in my database.

- **Items:-** To store the items available in canteen at present time. For default purpose i have added only 4 items in my items.



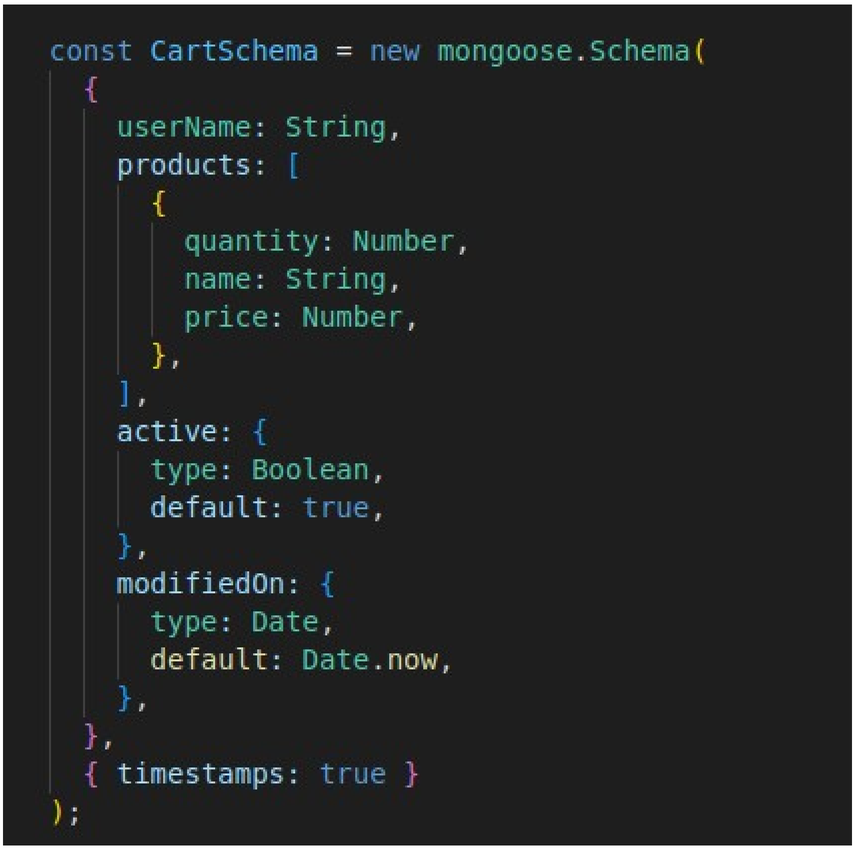
```
const itemsSchema = {
  name: String,
  description: String,
  price: Number,
};
```

- **User:-** To store the information of the registered users.



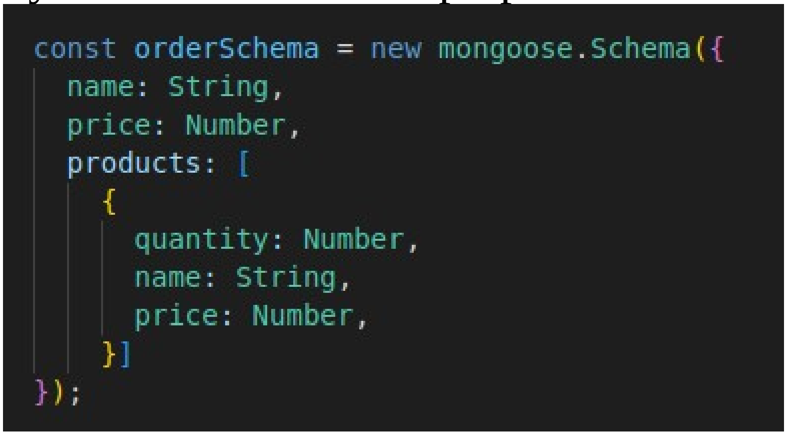
```
const userSchema = new mongoose.Schema({
  email: String,
  password: String,
  googleId: String,
});
```

- **Cart:-** A different cart for each user to store the items user wants to order.



```
const CartSchema = new mongoose.Schema(
  {
    userName: String,
    products: [
      {
        quantity: Number,
        name: String,
        price: Number,
      },
    ],
    active: {
      type: Boolean,
      default: true,
    },
    modifiedOn: {
      type: Date,
      default: Date.now,
    },
  },
  { timestamps: true }
);
```

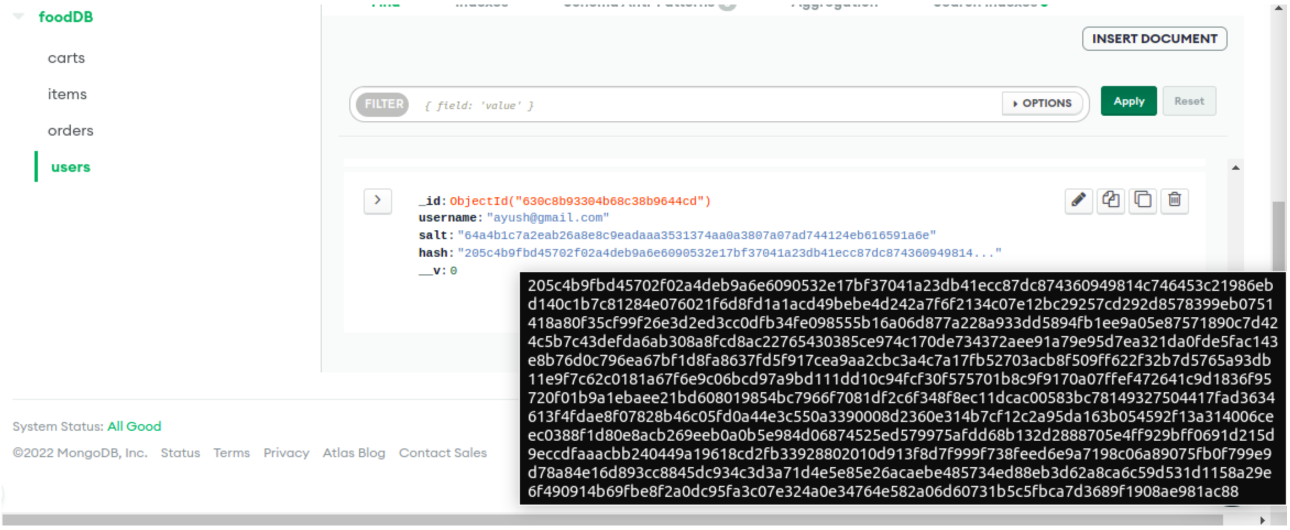
- **Order:-** To store the orders placed by different users this can be used by canteen workers to prepare orders.



```
const orderSchema = new mongoose.Schema({
  name: String,
  price: Number,
  products: [
    {
      quantity: Number,
      name: String,
      price: Number,
    }]
});
```

Security:-

For security purpose I have used passport module to store the password of user in hashed form. The techniques used are salting and hashing to make the password secure.
I have also used google signin option. This is also a secure login method for my website.



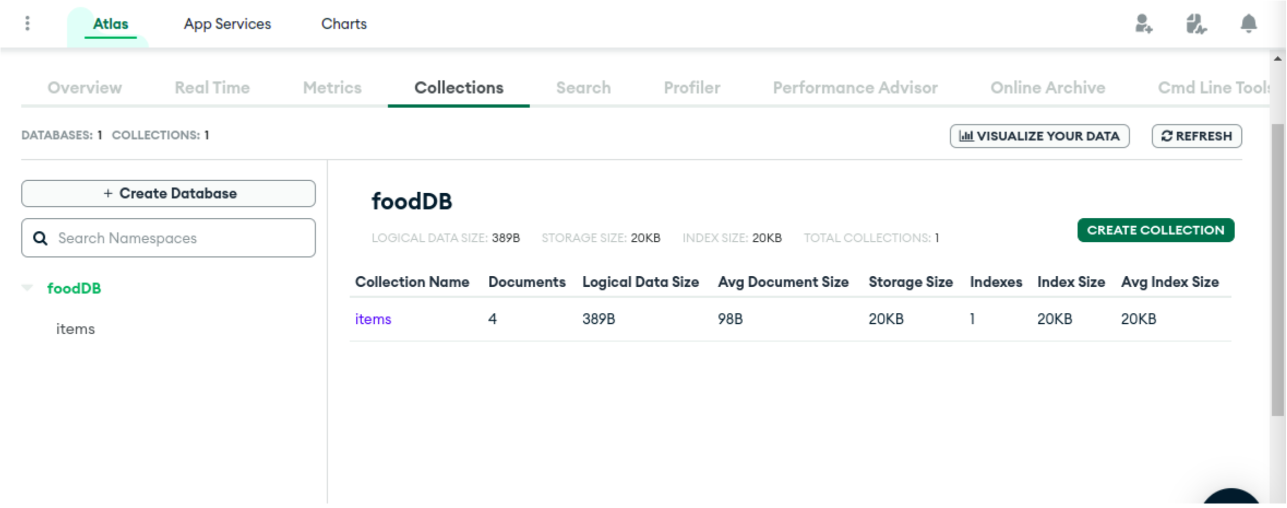
```
_id: ObjectId("630c8b93304b68c38b9644cd")
username: "ayush@gmail.com"
salt: "64a4b1c7a2eab26a8e8c9eadaaa3531374aa0a3807a07ad744124eb616591a6e"
hash: "205c4b9fbd45702f02a4deb9a6e6090532e17bf37041a23db41ecc87dc874360949814..."
__v: 0
```

```
205c4b9fbd45702f02a4deb9a6e6090532e17bf37041a23db41ecc87dc874360949814c746453c21986eb
d140c1b7c81284e076021f6d8fd1a1acd49bebe4d242a7f6f2134c07e12bc29257cd292d8578399eb0751
418a80f35cf99f26e3d2ed3cc0dfb34fe098555b16a06d877a228a933dd5894fb1ee9a05e87571890c7d42
4c5b7c43defda6ab308a8fcd8ac22765430385ce974c170de734372aee91a79e95d7ea321da0fde5fac143
e8b76d0c796ea67bf1d8fa8637fd5f917cea9aa2cbc3a4c7a17fb52703acb8f509ff622f32b7d5765a93db
11e9f7c62c0181a67f6e9c06bcd97a9bd111dd10c94fcf30f575701b8c9f9170a07ffef472641c9d1836f95
720f01b9a1ebaee21bd608019854bc7966f7081df2c6f348f8ec11dcac00583bc78149327504417fad3634
613f4fdae8f07828b46c05fd0a44e3c550a3390008d2360e314b7cf12c2a95da163b054592f13a314006ce
ec0388f1d80e8acb269eeb0a0b5e984d06874525ed579975afdd68b132d2888705e4ff929bff0691d215d
9eccdfaaacbb240449a19618cd2fb33928802010d913f8d7f999f738feed6e9a7198c06a89075fb0f799e9
d78a84e16d893cc8845dc934c3d3a71d4e5e85e26acaebe485734ed88eb3d62a8ca6c59d531d1158a29e
6f490914b69fbe8f2a0dc95fa3c07e324a0e34764e582a06d60731b5c5fbca7d3689f1908ae981ac88
```

Express:-
I have made use of express module to display different pages and list of items such as view items in cart or displaying purchased items along with total cost of order, date and time of placing order.

Working of my website:

1. Initially my database has 4 default items and no users, no cart and no orders.



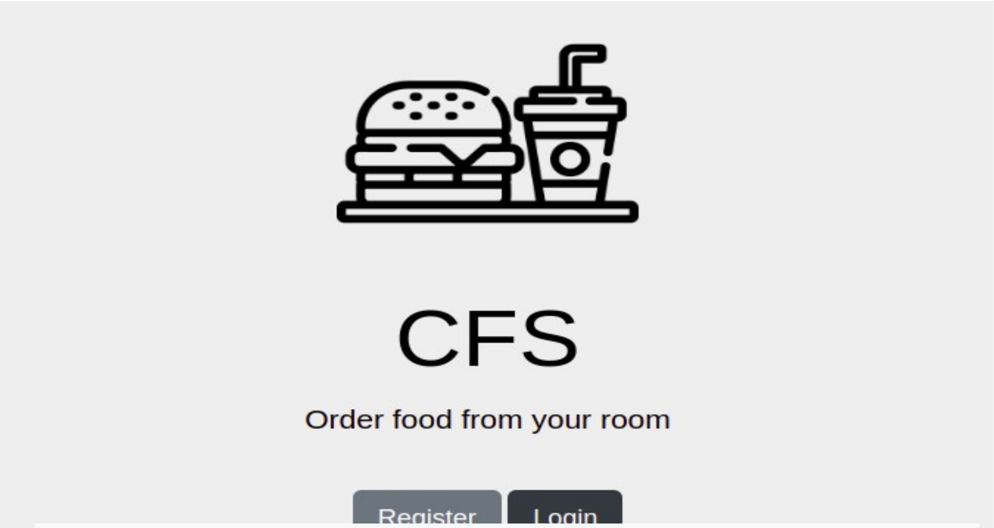
DATABASES: 1 COLLECTIONS: 1

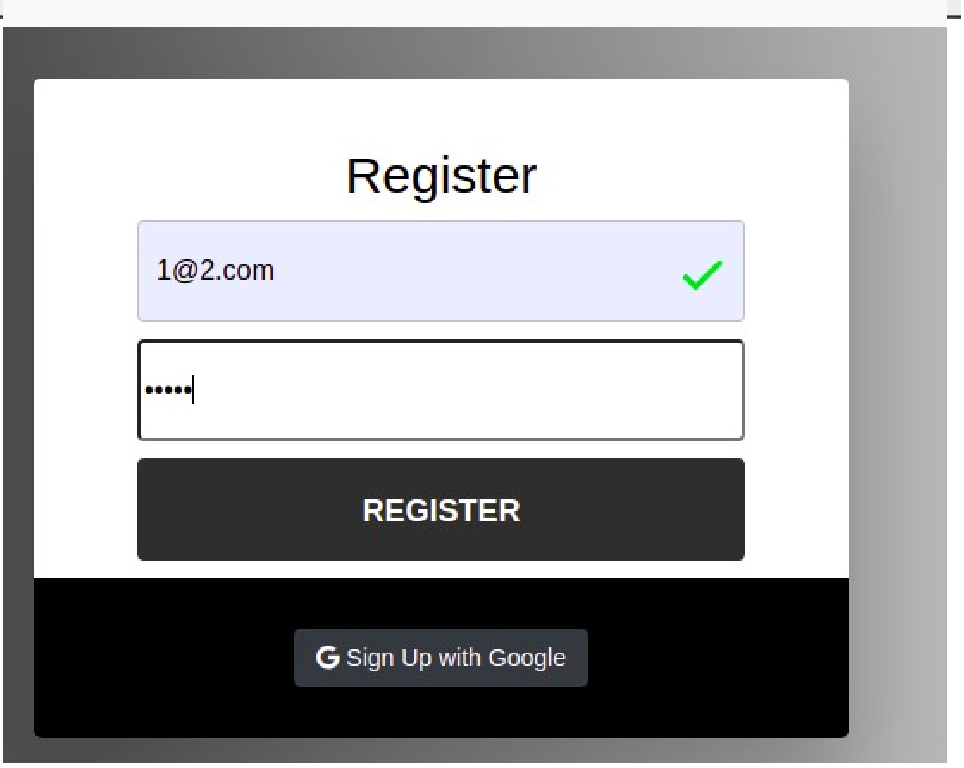
foodDB

LOGICAL DATA SIZE: 389B STORAGE SIZE: 20KB INDEX SIZE: 20KB TOTAL COLLECTIONS: 1

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
items	4	389B	98B	20KB	1	20KB	20KB

2. From Home page you can register new user or login with previous username and password.

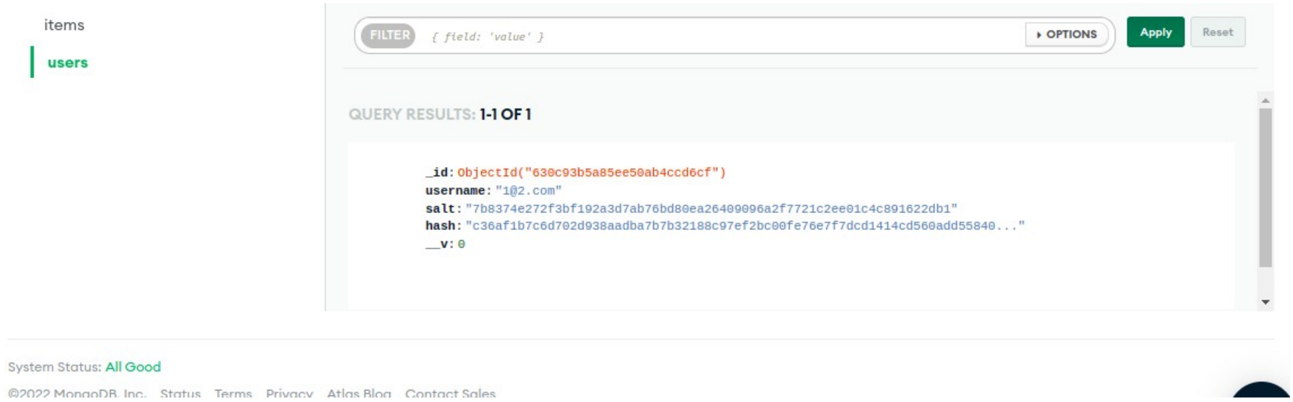




1@2.com

•••••

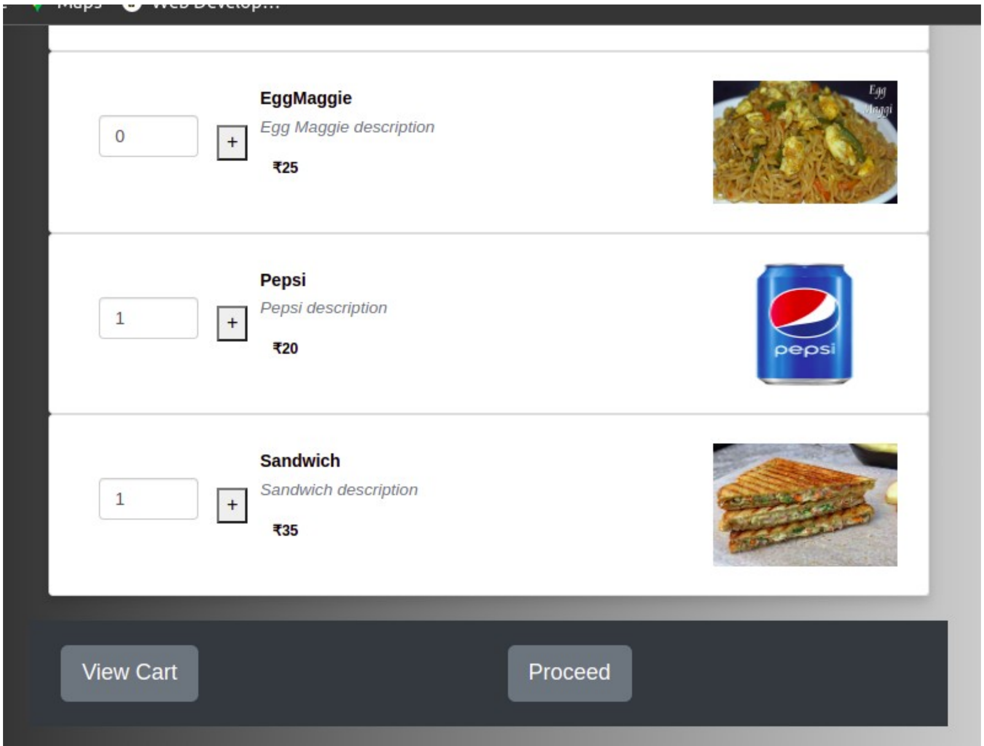
3. After we register new user we can see the details of our new user in our cloud database.



QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId("630c93b5a85ee50ab4ccd6cf")
username: "1@2.com"
salt: "7b8374e272f3bf192a3d7ab76bd80ea26409096a2f7721c2ee01c4c891622db1"
hash: "c36af1b7c6d702d938aadba7b7b32188c97ef2bc00fe76e7f7dcd1414cd560add55840..."
__v: 0
```

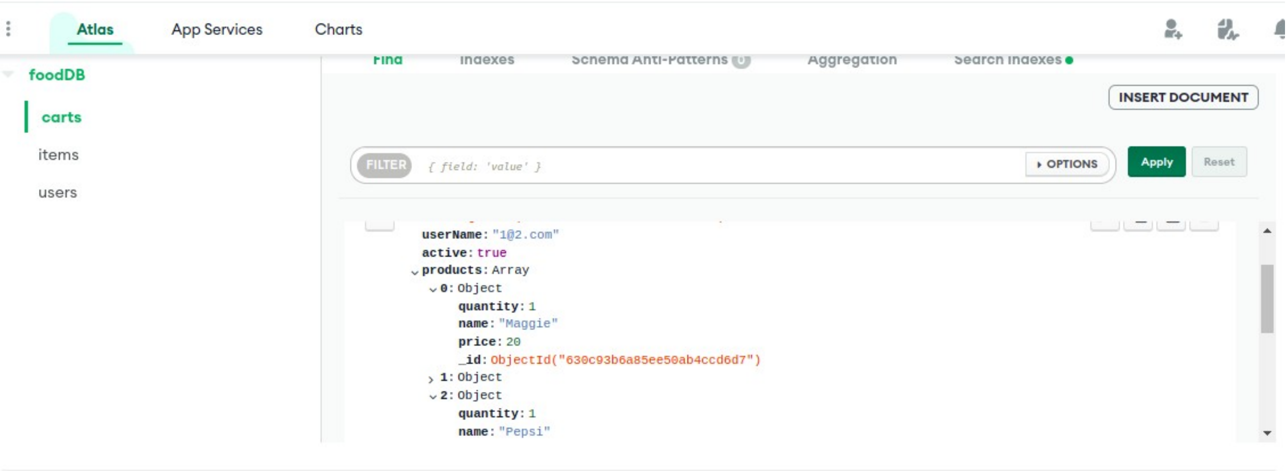
4. We can add items in cart by selecting quantity and clicking on + button to add the items in cart. Every user has a different cart assigned to him after registering on website.



0 EggMaggie
Egg Maggie description
₹25

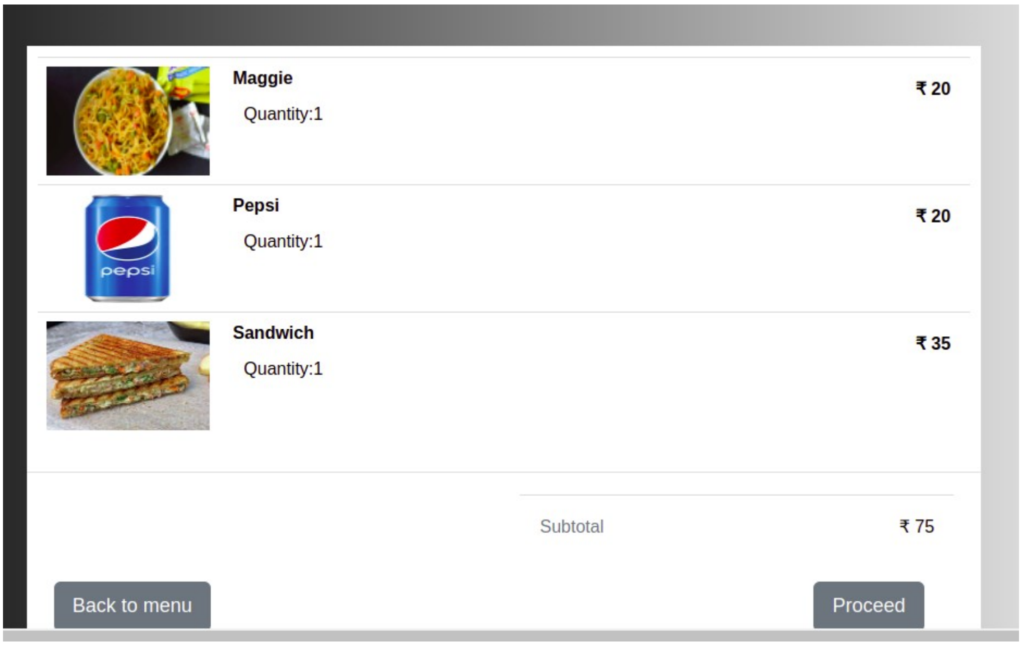
1 Pepsi
Pepsi description
₹20

1 Sandwich
Sandwich description
₹35



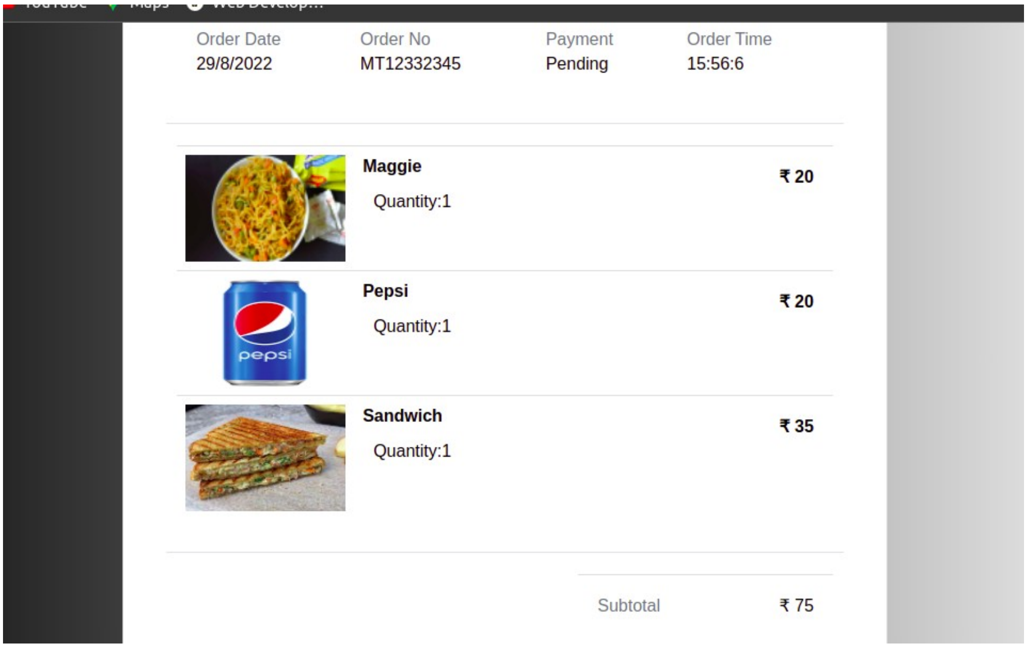
```
userName: "1@2.com"
active: true
⌄products: Array
  ⌄0: Object
      quantity: 1
      name: "Maggie"
      price: 20
      _id: ObjectId("630c93b6a85ee50ab4ccd6d7")
  ›1: Object
  ⌄2: Object
      quantity: 1
      name: "Pepsi"
```

5. User can view cart or place order.



Maggie Quantity:1		₹ 20
Pepsi Quantity:1		₹ 20
Sandwich Quantity:1		₹ 35
	Subtotal	₹ 75

6. By clicking on Proceed button order will be placed for user and new order will be entered in database.

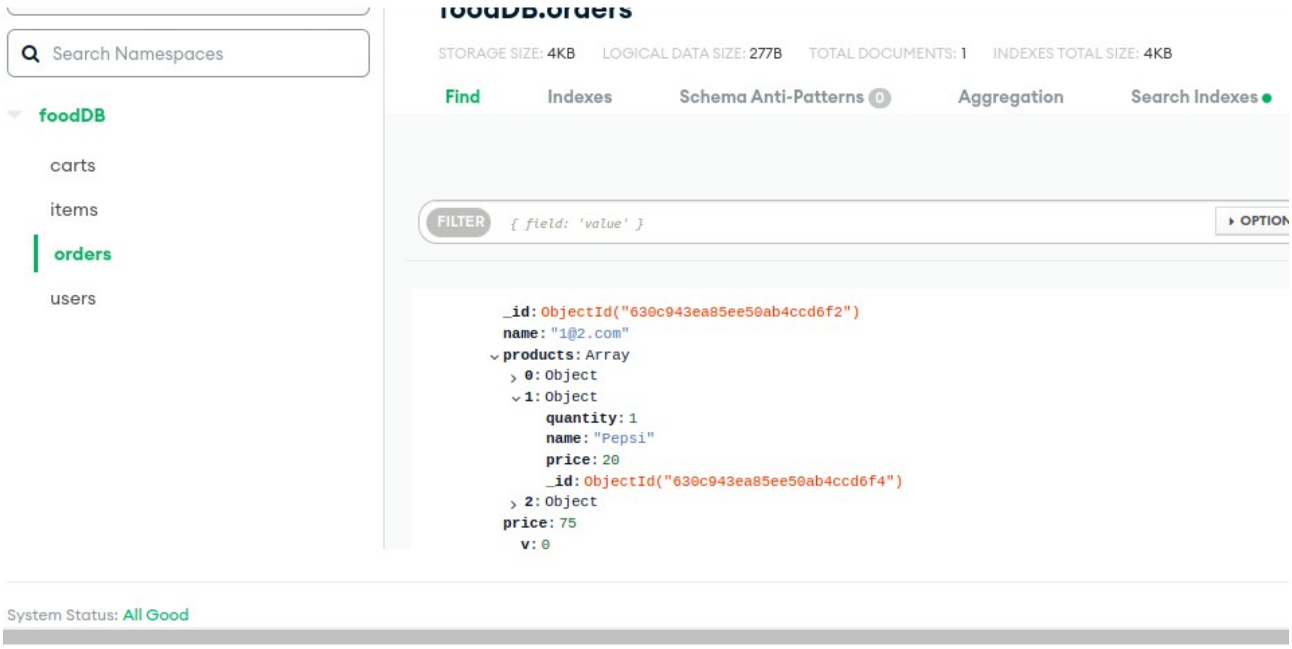


Order Date	Order No	Payment	Order Time
29/8/2022	MT12332345	Pending	15:56:6

Item	Quantity	Price
Maggie	Quantity:1	₹ 20
Pepsi	Quantity:1	₹ 20
Sandwich	Quantity:1	₹ 35

Subtotal ₹ 75

7. In Database you can see placed order.



foodDB.orders

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 277B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 4KB

```
  _id: ObjectId("630c943ea85ee50ab4ccd6f2")
  name: "1@2.com"
⌄ products: Array
  › 0: Object
  ⌄ 1: Object
      quantity: 1
      name: "Pepsi"
      price: 20
      _id: ObjectId("630c943ea85ee50ab4ccd6f4")
  › 2: Object
  price: 75
    v: 0
```

System Status: All Good