2022

FINAL PROJECT REPORT

ONLINE PIZZA ORDERING WEB-BASED APP

SUBMITED BY :

Inuri Gunathilaka

UNDER THE GUIDANCE OF:

Mr. Samitha Nanayakkara

A Project report On

Real-time Pizza Ordering System

Submitted by

Inuri Gunathilaka

Register No: AR/96541

Index No: AF/18/14575

Under The Guidance of:

Mr. Samitha Nanayakkara

(Lecturer, Dept. of ICT)

June 2022

ICTC 3101.3 – Web Application Development

Bachelor of Science (Hons) Information Technology

Department of Information and Communication Technology

Faculty of Humanities and Social Sciences

University of Sri Jayewardenepura

1. *ABSTRACT*

Real-time Pizza Ordering System is proposed here which simplifies the pizza ordering process. The proposed system shows a user interface and update the menu with all available options so that it eases the customer work. Customer can choose more than one item to make an order and can view order details before logging off. The order confirmation is sent to the customer. The order is placed in the queue and updated in the database and returned in real time. This system assists the staff to go through the orders in real time and process it efficiently with minimal errors.

Index Terms — A frontend (React) website, a Backend (NodeJS + ExpressJS) and a database (MongoDB)

.

Table of Contents

Contents

2. *Problem description*

i. Introduction

The "Online Pizza Ordering System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Online Food Ordering System, as described above, can lead to error free, secure, reliable and fast management system. Our System come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

ii. System specification

Functionalities provided by pizza ordering system are as follows:

- Provides the searching facilities based on various factors. Such as Food Item, Customer, Order, Confirm Order.
- System manage the payment details online for order details, confirm order details, food item.
- Manage the information of food items.
- Editing, adding and updating of records is improved which results in proper resource management of food item data.
- Manage the information of order.

Technical Specification provided by pizza ordering system are as follows:

The system is made up of three layers. At the top there is the GUI (Graphical User Interface) layer, the middle layer is the storage and query manager, the bottom layer is the underlying database.

- GUI - There are 2 separate GUI's. One is for the customers to create orders and one is for employees for processing orders and administration purposes.
- Storage and Query manager - The storage and query manager layer is responsible for information storage, retrieval, authorizations and error checking. This layer allows selecting, adding, updating, deleting entities and relations in the database by using different queries.
- Database layer - The database layer contains all the data of entities and their relationships.

iii. Requirements

Functional Requirements

1. **Customer**

A Customer who will be able to search Menu items on the platform listed by store and can order them online and get delivered at home through the platform.

- The user must be able to create a new order.
- The user must be able to customize a pizza by:
  - The user must be able to view a list of available ingredients.
  - The user must be able to add an ingredient to a custom pizza.
  - The user must be able to remove an ingredient from a custom pizza.
- The user must be able to add a custom pizza to an order.
- The user must be able to delete a custom pizza from an order.
- The user must be able to see the total price of an order
- The user must be able to add the name and address of the customer.
- The user must be able to clear the current order to start a new one.

2. **Administrators**

The system admin can list menu items on the platform, get orders from the customers, and get delivered them to customer's place through platform's delivery personals or through delivery person of himself.

- The administrator must be able to log in and out.
- The administrator must be able to add/delete/edit orders.
- The administrator must be able to add/delete/edit ingredients.
- The administrator must be able to add/delete/edit other users.
- The administrator must be able to view an order log.
- Only users with respective rights (administrators) must be able to use all these "Administrators" features.

Non-Functional Requirements

- The system will run as a database with a website as user interface. As performance requirement the system must be accessible 24 hours a day, seven days a week. Due to the nature of the system as an ordering website, the system must have a low response time, preferably shorter than second, with a maximum of five seconds. The exception is viewing order logs which could have a higher response time (of seconds) as the log increases in size over time.

3. *Results*

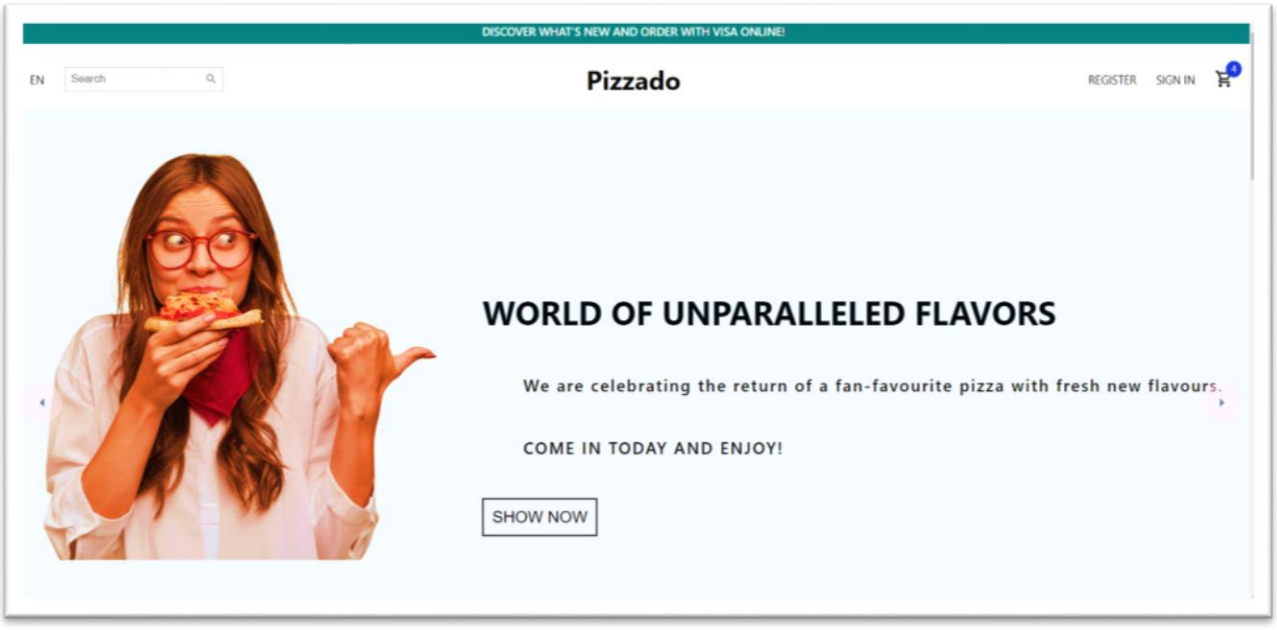
i. ***System overview including capabilities and GUI***

Implementation-Frontend

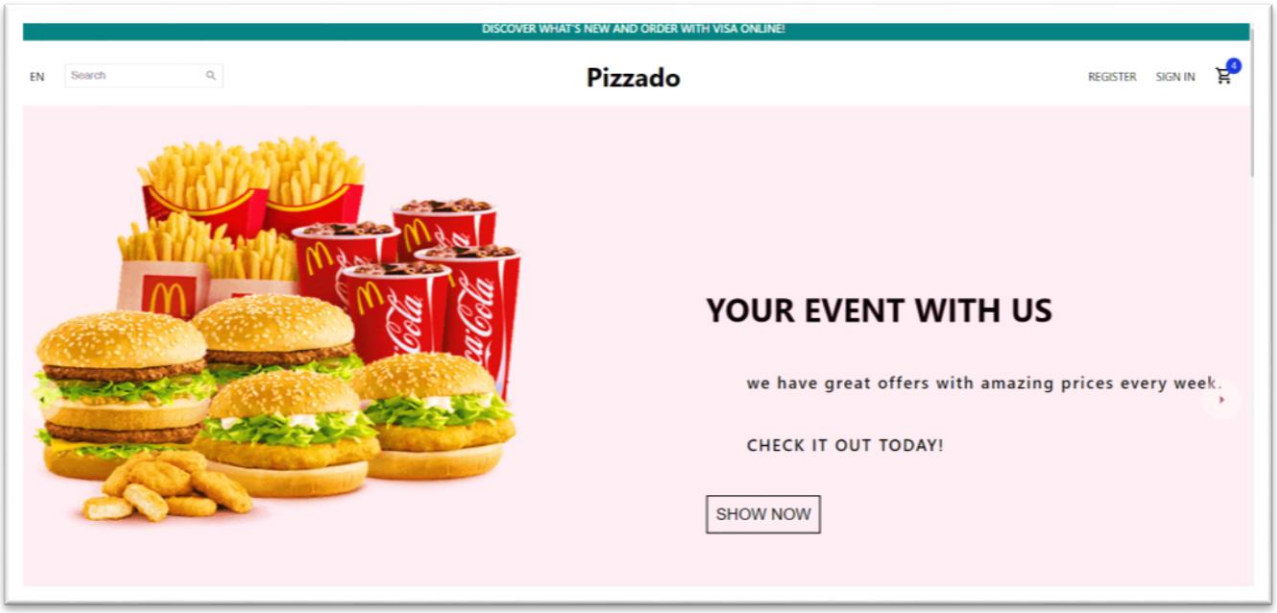
Home Page

This is homepage,which has created absolute slider without using any library. It has consist of categories, products and other components. This was created using functional reusable react components react hooks and start components. When I'm clicking right side and as you can see secound one. When clicking again the third one and it's going to first run again. Using animation make the page look better as well as timing function.

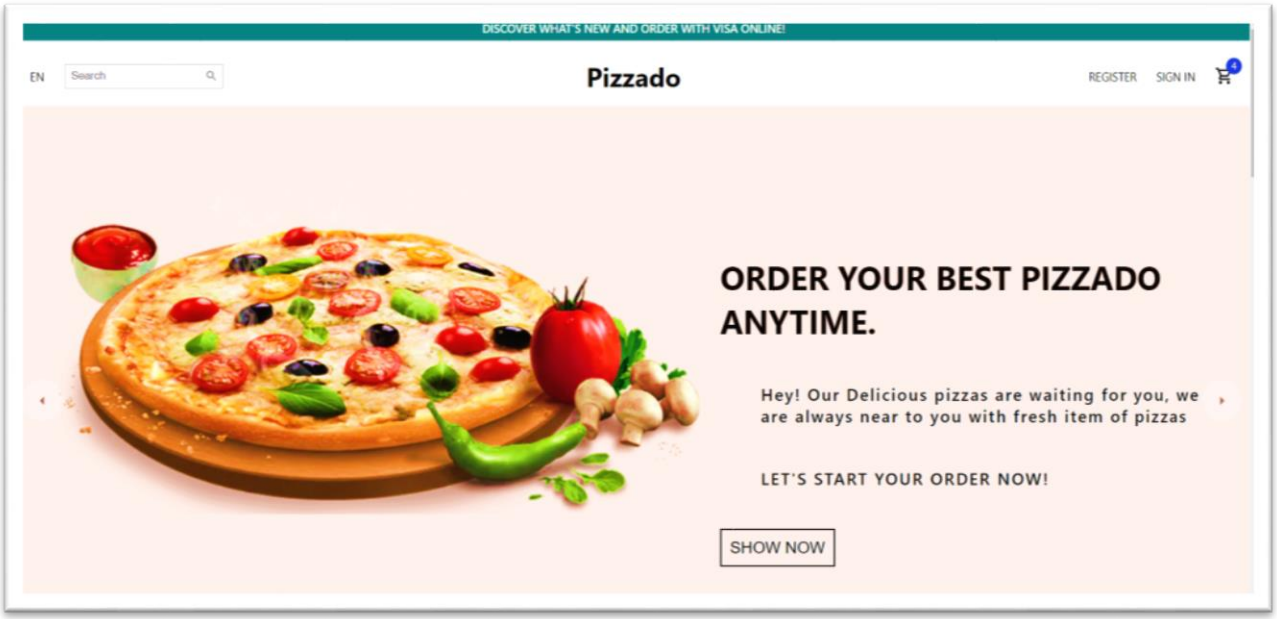
1st Slide

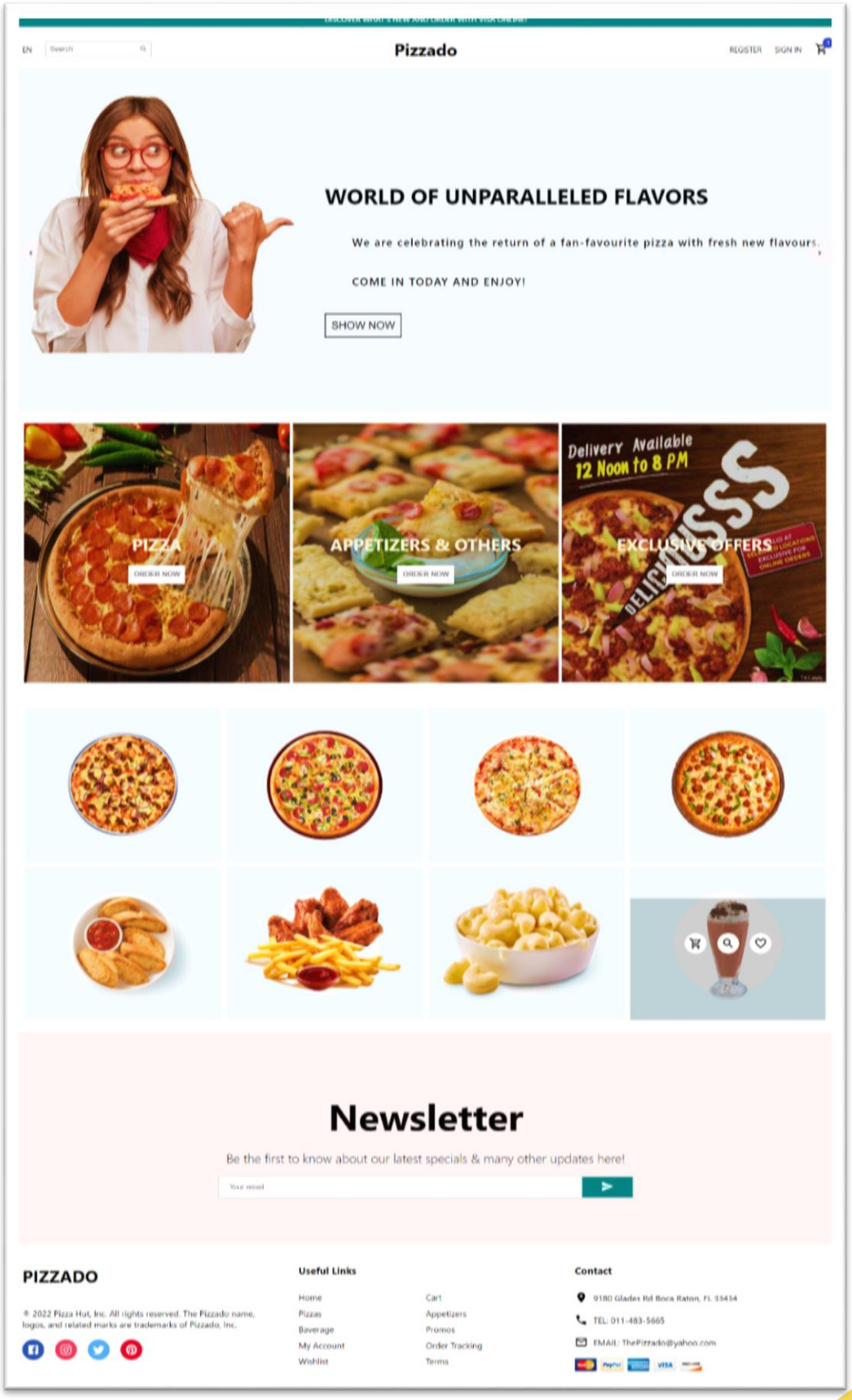


2nd Slide

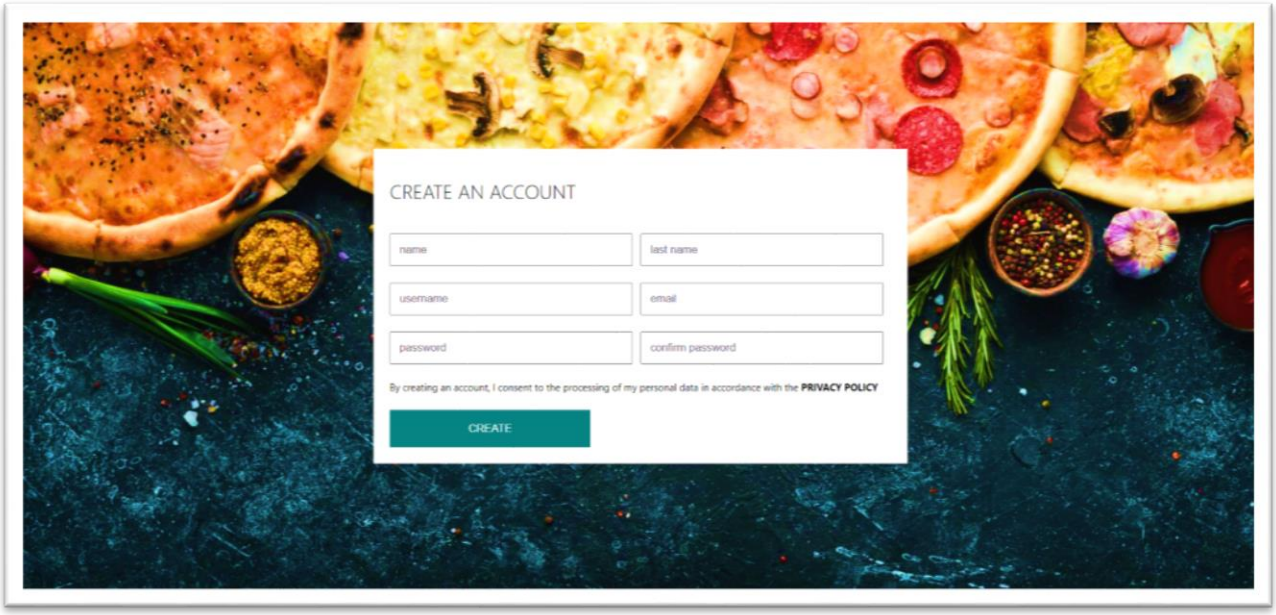


3rd Slide

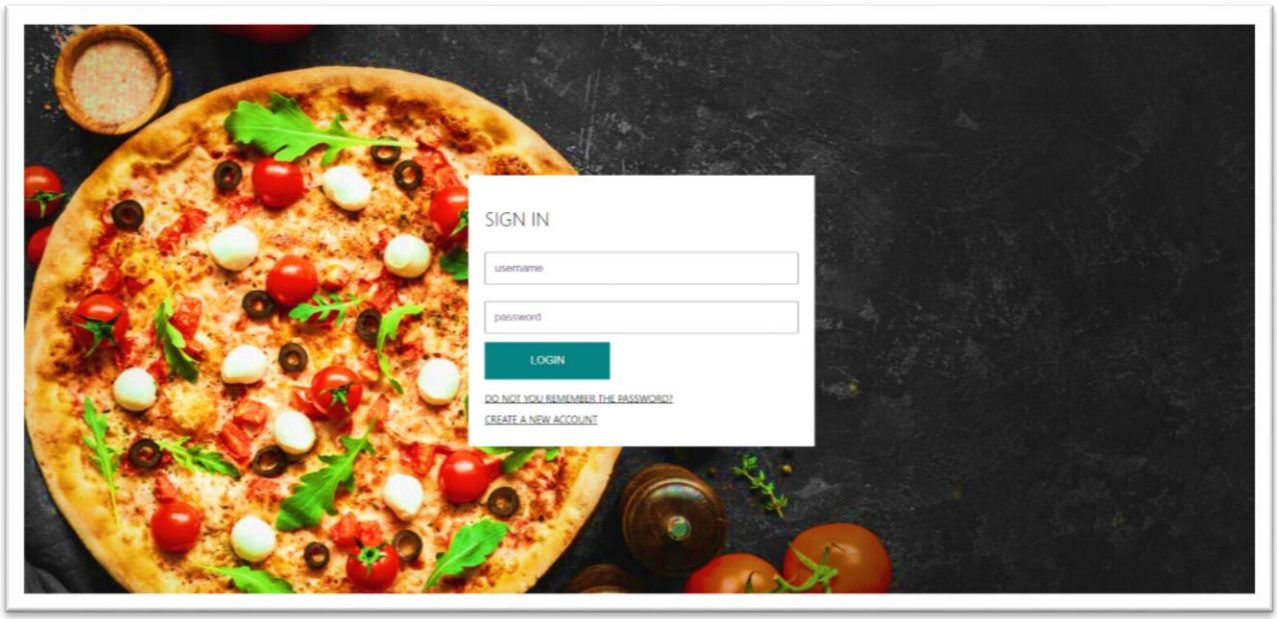



EN
Search
Pizzado
REGISTER
SIGN IN
WORLD OF UNPARALLELED FLAVORS
We are celebrating the return of a fan-favourite pizza with fresh new flavours.
COME IN TODAY AND ENJOY!
SHOW NOW
PIZZA
ORDER NOW
APPETIZERS & OTHERS
ORDER NOW
Delivery Available
12 Noon to 8 PM
EXCLUSIVE OFFERS
ORDER NOW
Newsletter
Be the first to know about our latest specials & many other updates here!
PIZZADO
© 2022 Pizza Hut, Inc. All rights reserved. The Pizzado name, logos, and related marks are trademarks of Pizzado, Inc.
Useful Links
Home
Pizzas
Beverage
My Account
Wishlist
Cart
Appetizers
Promos
Order Tracking
Terms
Contact
9180 Glades Rd Boca Raton, FL 33434
TEL: 011-483-5665
EMAIL: ThePizzado@yahoo.com

Register Page

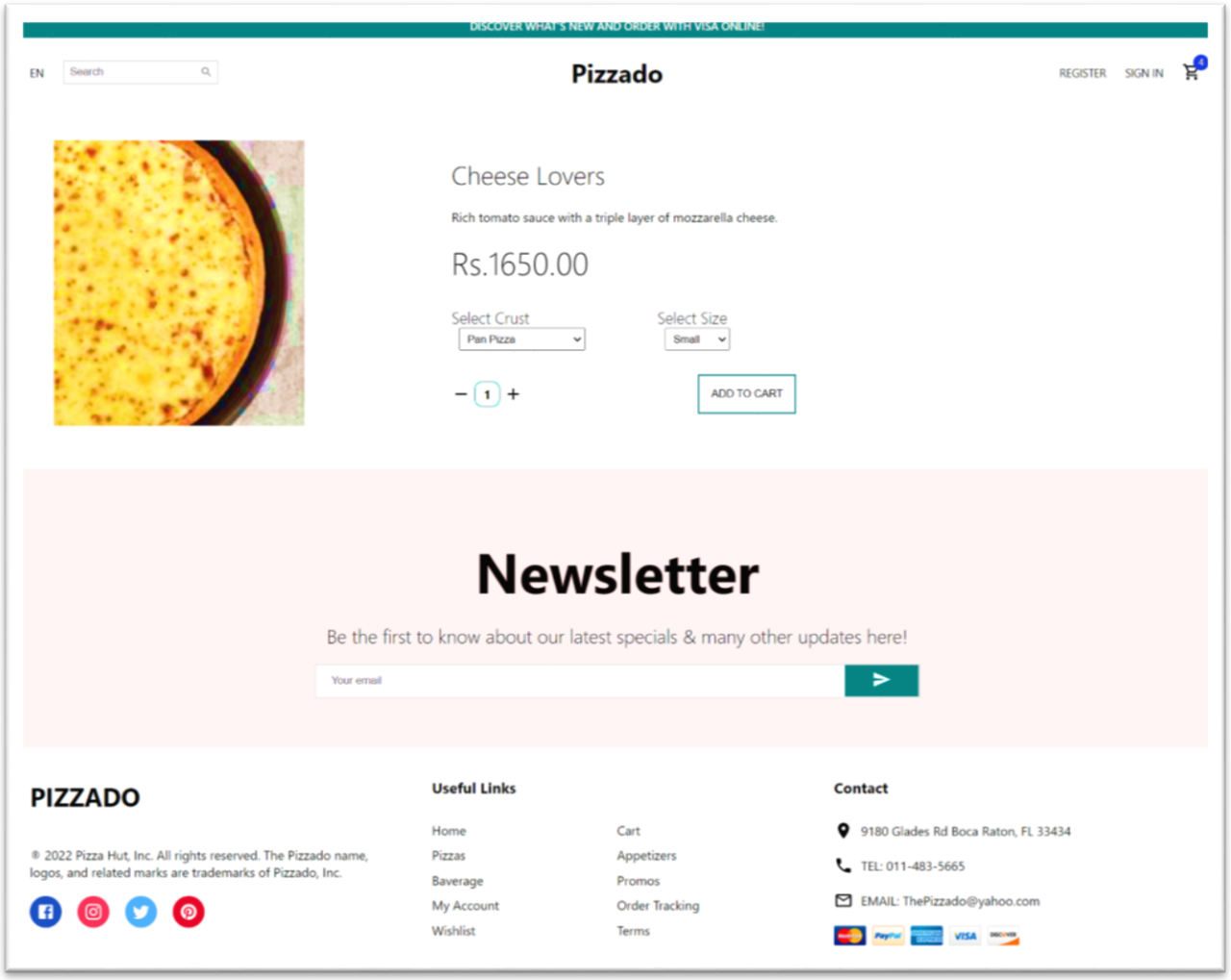


Login Page

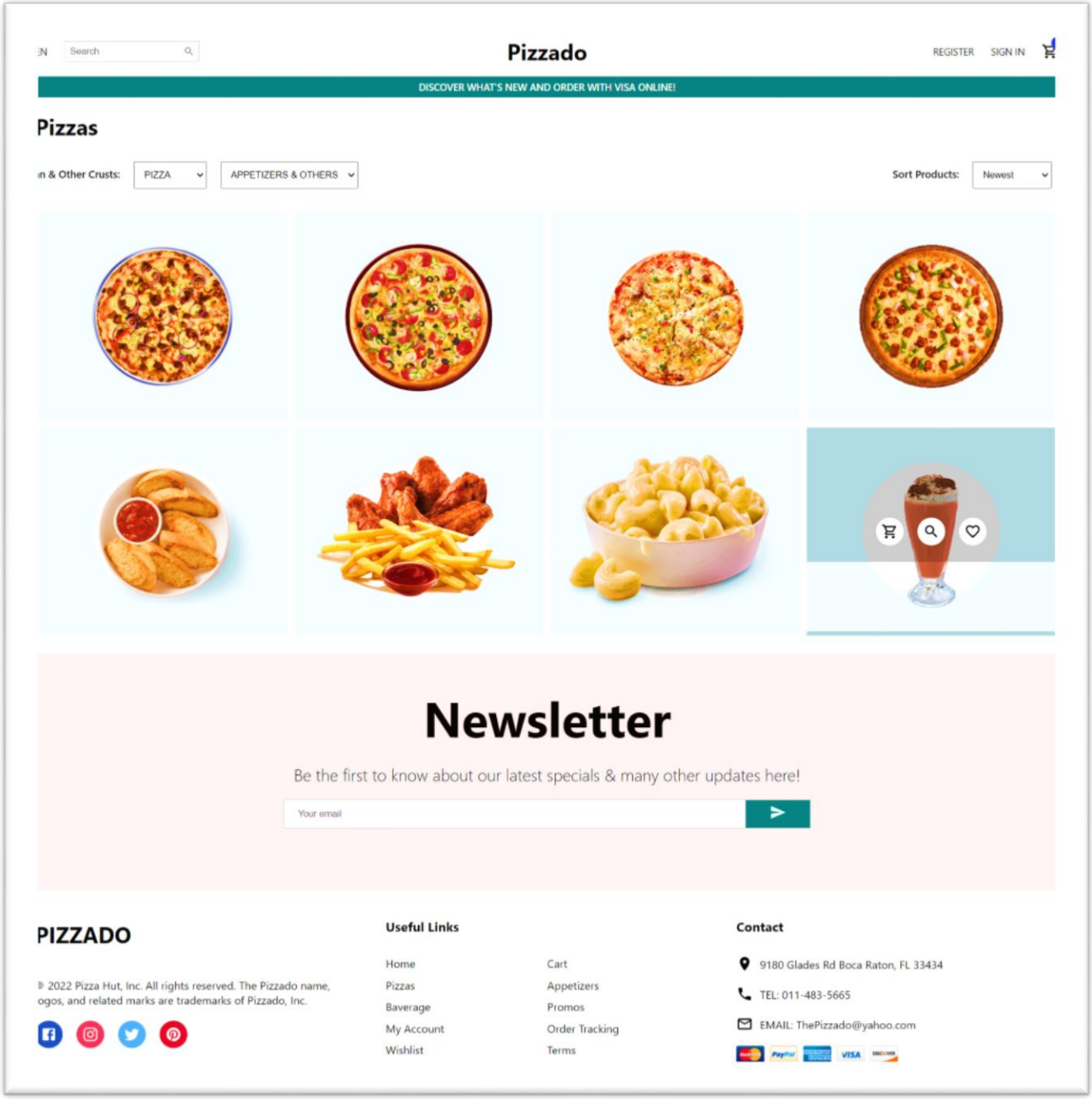


Single Product Page

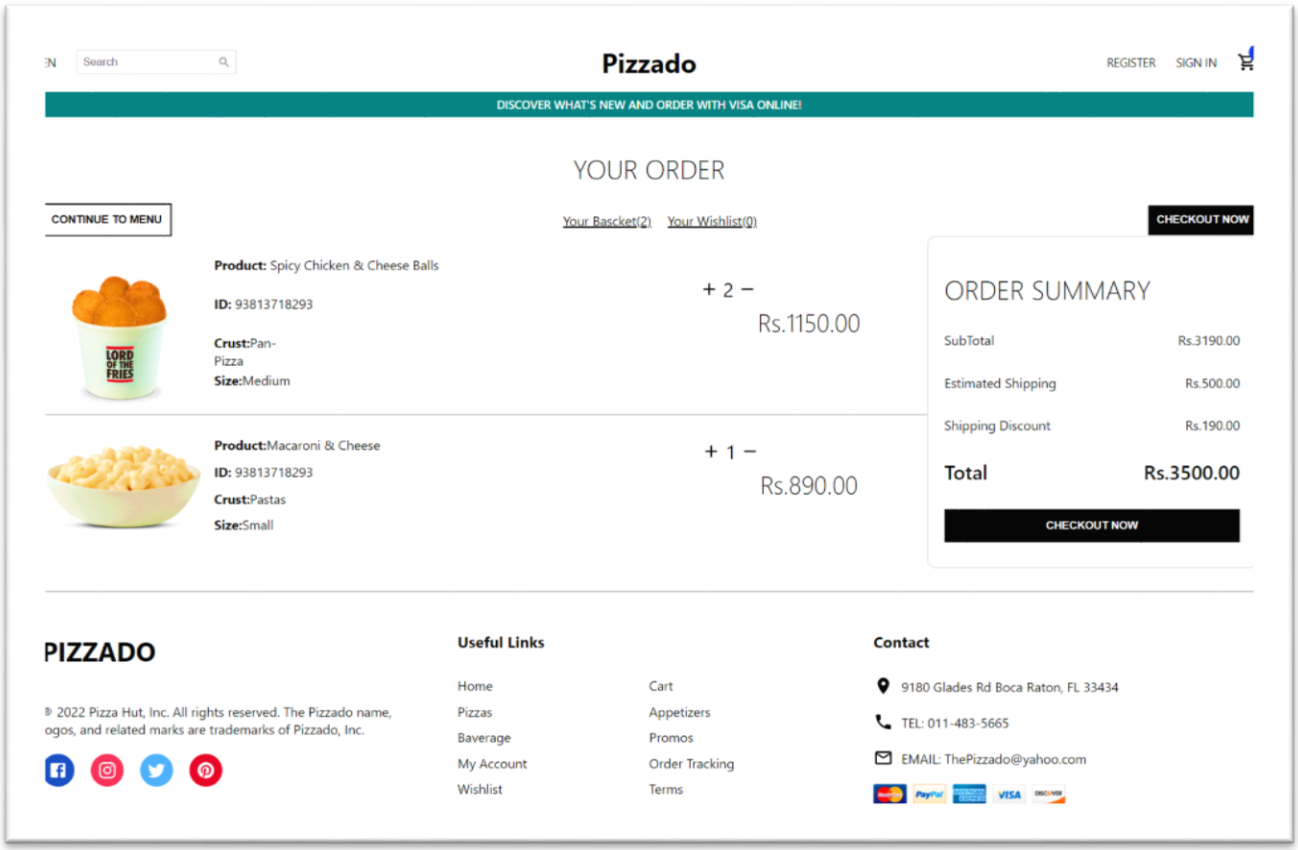
In here left side will show our product image and right side will be the information, purchase button.



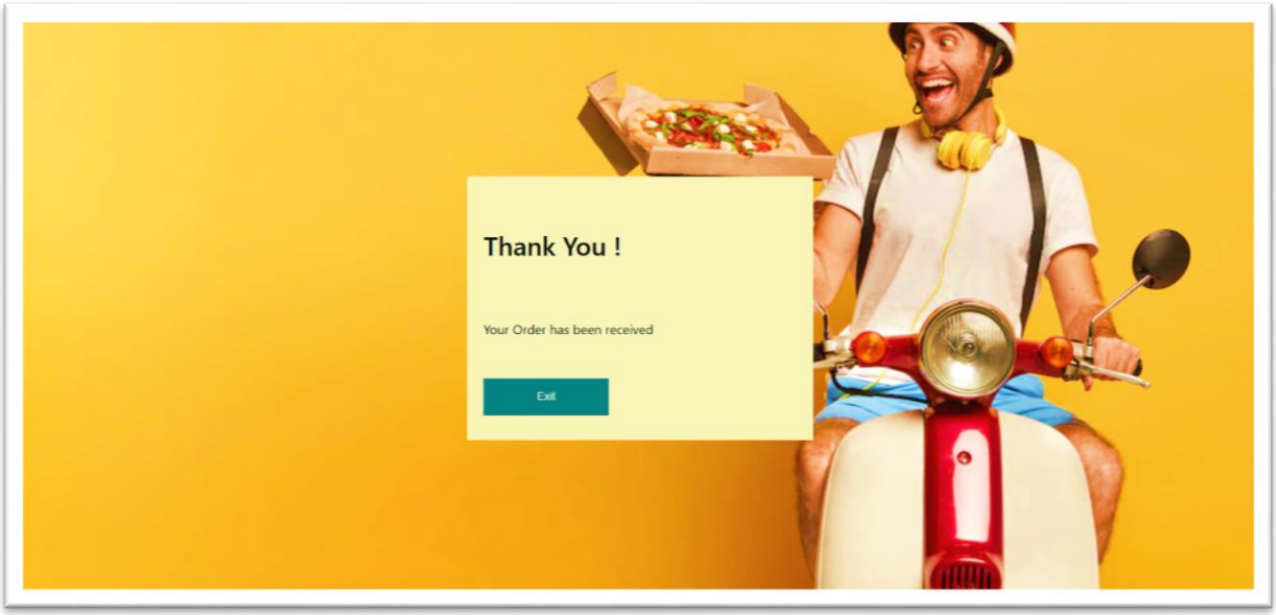
Product List



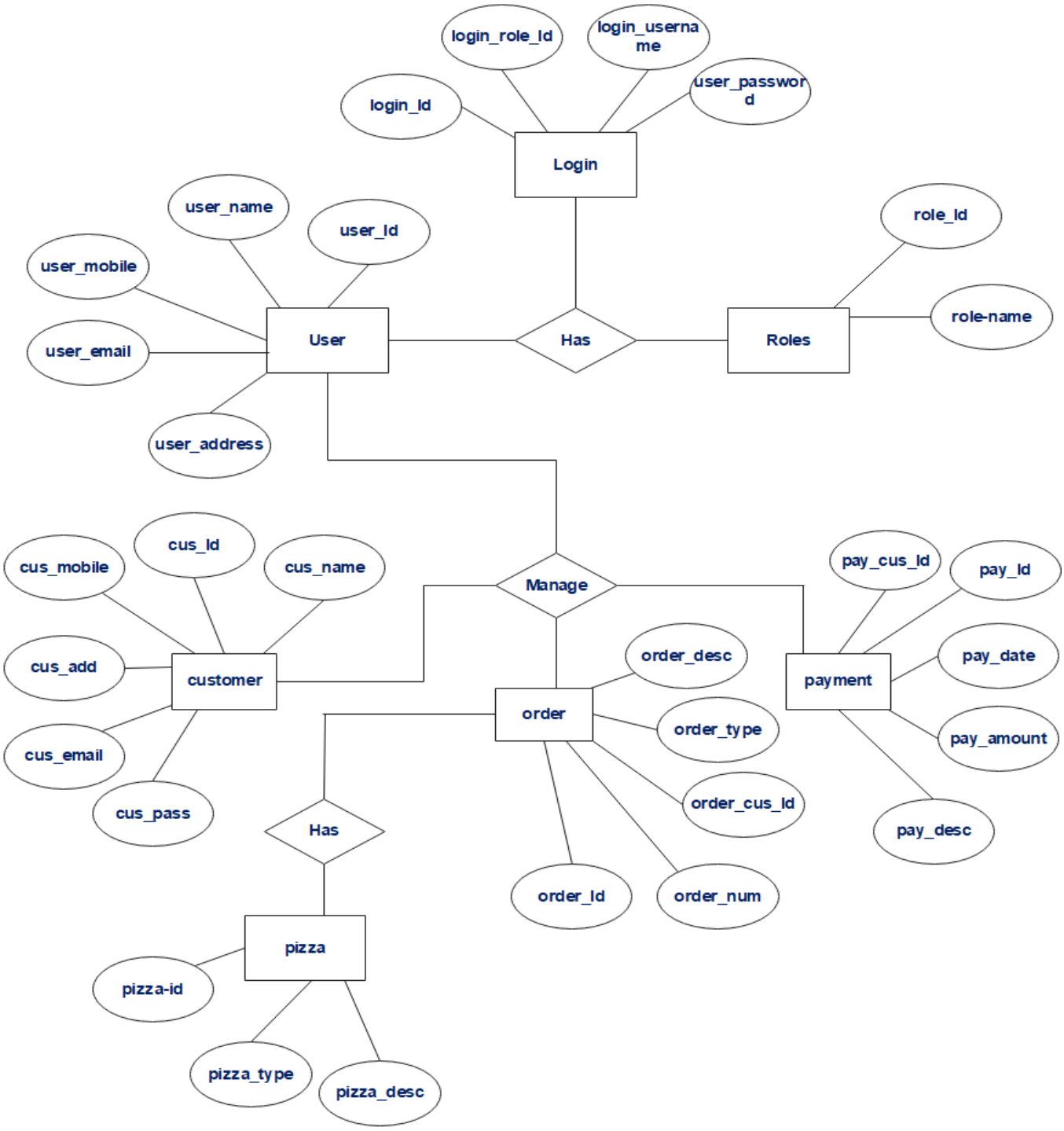
Shopping Cart



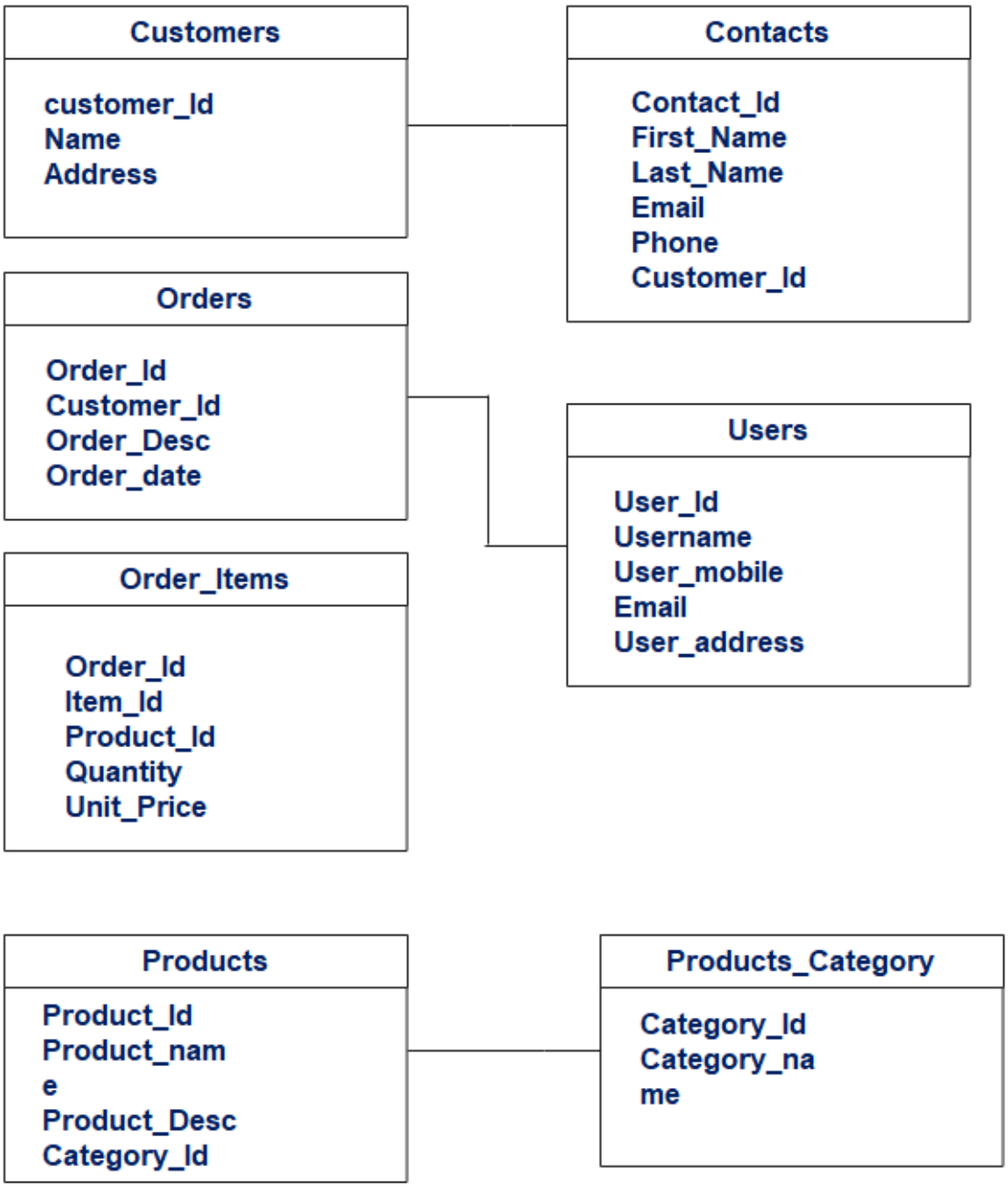
Checkout confirm Message



ii. Database Structure



ER Diagram for Pizza Ordering System



Relational Schema for Pizza Ordering System

iii. ***A detailed explanation of each API implementations and respective Endpoints***

This is a Full Stack (MERN) Pizza Delivery Application developed using React and React for Frontend , Node JS for Runtime environment , Express JS for Backend Routing and Mongo DB for Database.

In this course implement the following features in the Pizza Delivery Application,

- **Working with Redux**
- **Add to cart Feature**
- **Update Quantity in cart**
- **Delete Products from cart**
- **User Authentication**
- **Common Payment Gateway Integration**
- **Paying amount with stripe**
- **Placing Orders**
- **Store orders in database**
- **Retrieve Orders to user profile**
- **User Dashboard**
- **Admin Dashboard**
- **Manage Users , Products , Orders in Admin Panel**
- **Protected routes for admin panel**

Sign up/Login

The admin will be able to sign up/login to their account. A admin details consists of admin Name, Email, Address, City, State, Country, Contact Number ,Password. If a user already has an account then he/she can enter the credentials as email id and password. If the credentials are incorrect, users will be redirected to log in page and a message will be displayed that, username or password is not valid.

Dashboard

Profile — The System Name, photo, Address (City, State, Country), Location from google search suggest API, Restaurant Contact Number, Delivery postal code areas, Edit Details.

Status (Open / Closed) — This status will be managed by the restaurant owner and the same will reflect in the Customer's application. For example, if the status is set "Closed" by the owner, it will show "Closed Now" to the customer. If 'closed now' status, the restaurant will not receive any orders. If the status is set "Open" by the admin, it will show "Open Now" to the customer. If 'Open now' status, the restaurant will receive any orders.

Manage Orders

Active orders — The orders will have the details as Search by keyword, Customer Name, Customer Address, Customer Contact Number, Dish Name, Crust Type, Quantity, Select order status (Confirm, Order in making and order picked up), Order Status (Confirm / Order in making / Order picked up by the Delivery person/ Delivered / Cancelled. These status except ‘delivered’ will be managed by the system. As soon as the ‘order is in making’ a request is sent to the nearby Delivery person’s app. The one who accepts, his/her GPS location will be tracked), Delivery person Name, Delivery person Contact Number and Total Amount. The Delivery person name, Delivery person contact number will be visible only after the nearby Delivery person has accepted the request.

Completed orders — The orders will have the details as Search by keyword, Order id, Customer Name, Customer Address, Customer Contact Number, Delivery person Name, Delivery person Contact Number, Dish Name, Crust Type, Quantity, Status (delivered) and Total Amount.

Received rating — The Customer name, order id, order date, Star rating (out of five).

Notification — Received an order, Received rating, Customer cancelled the order

Account setting — The user will be able to change password and manage notification.

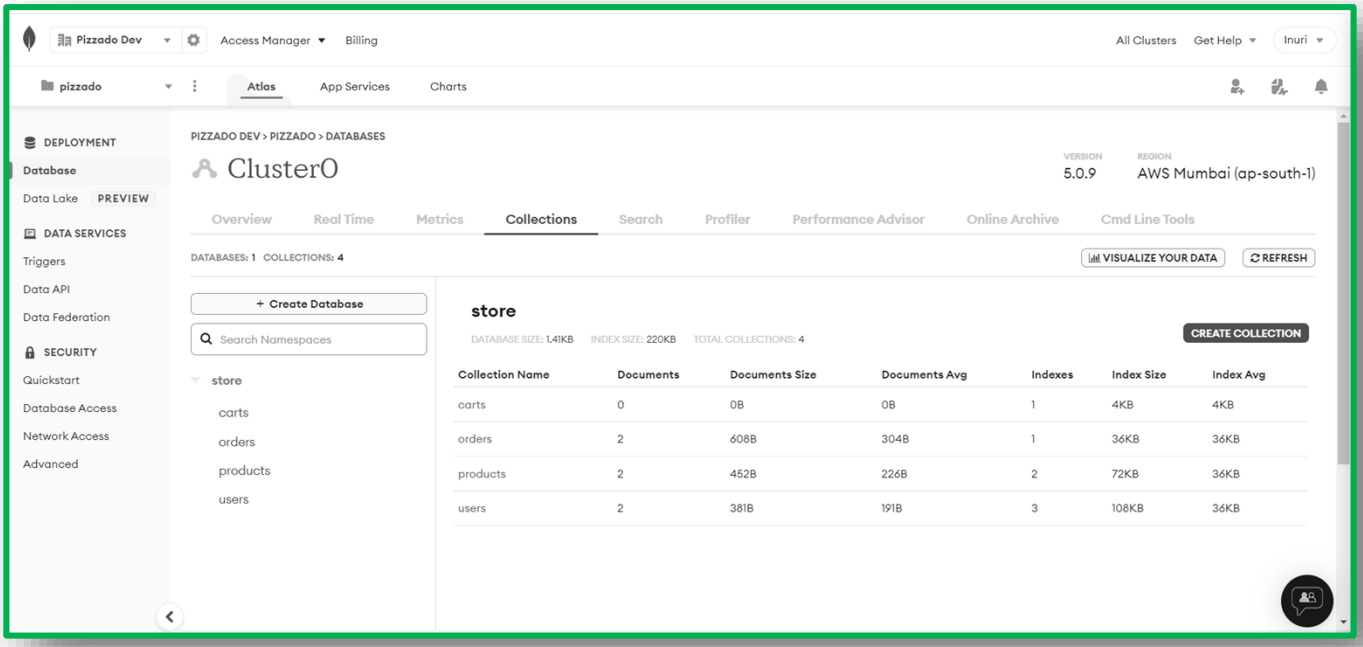
Logout — The user can logout to there account.

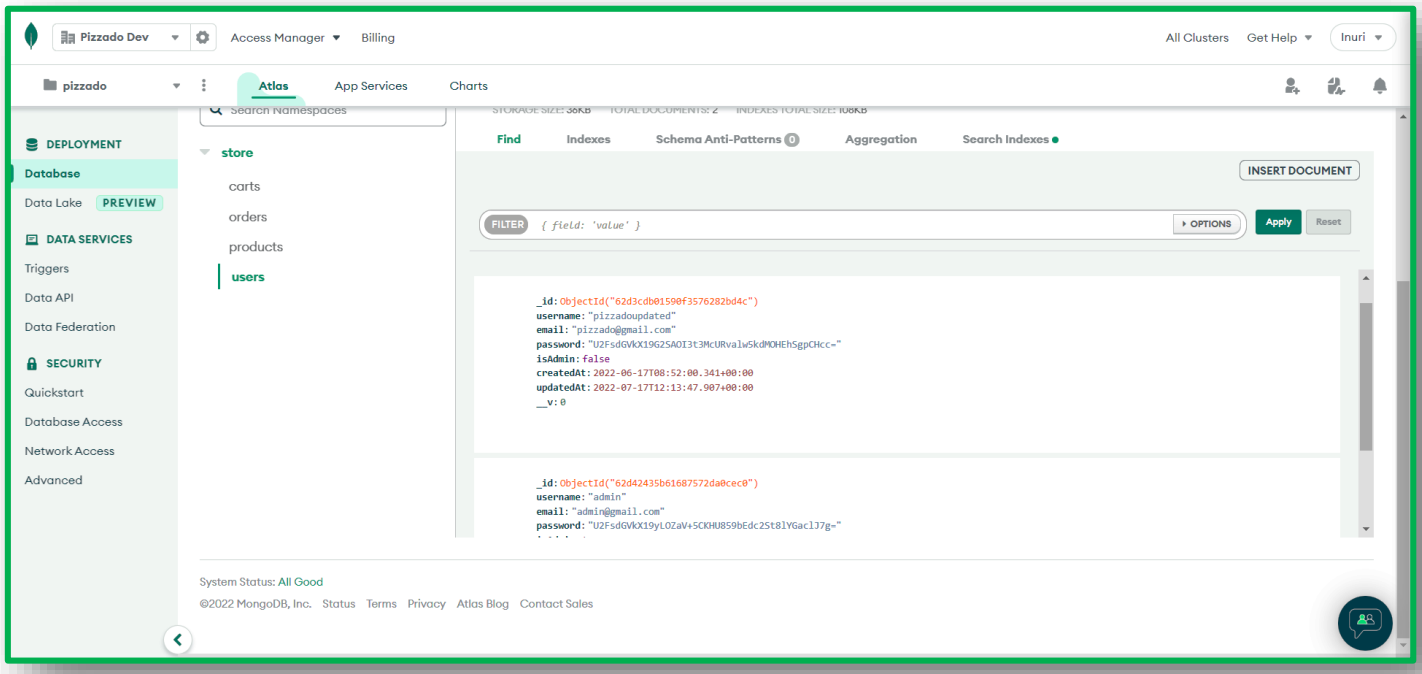
Backend

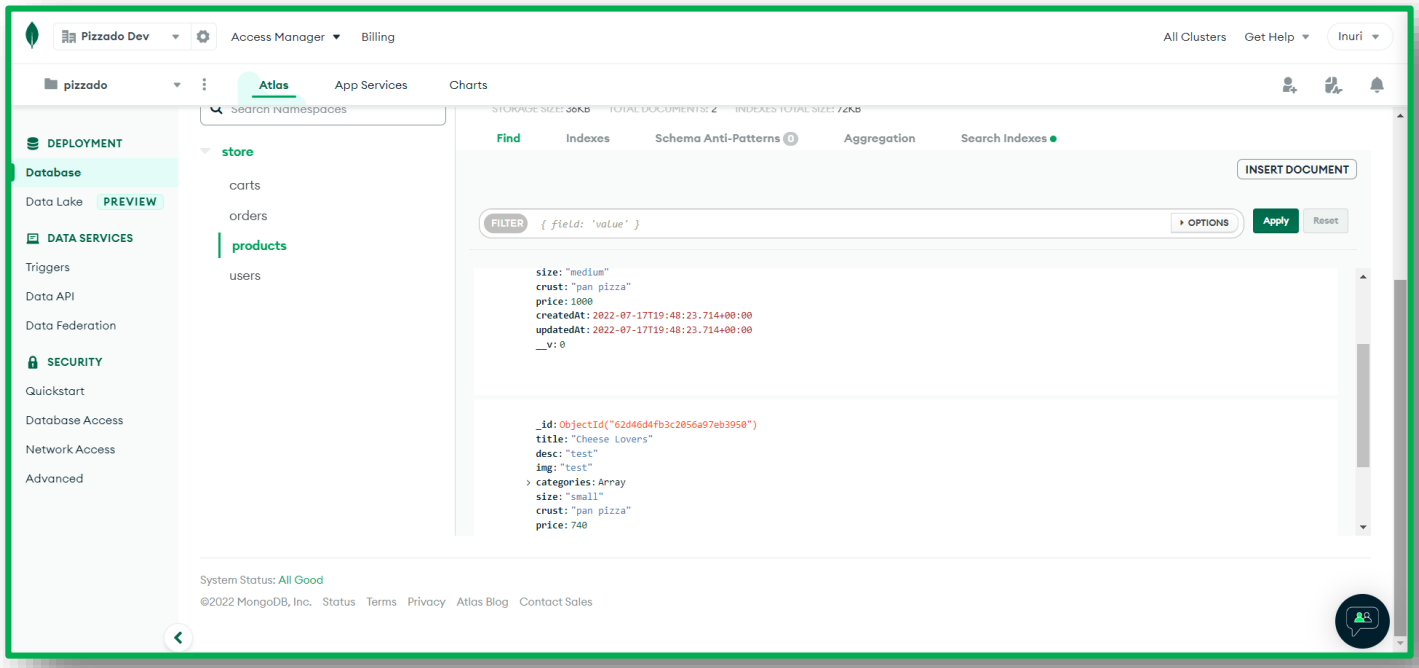
Database

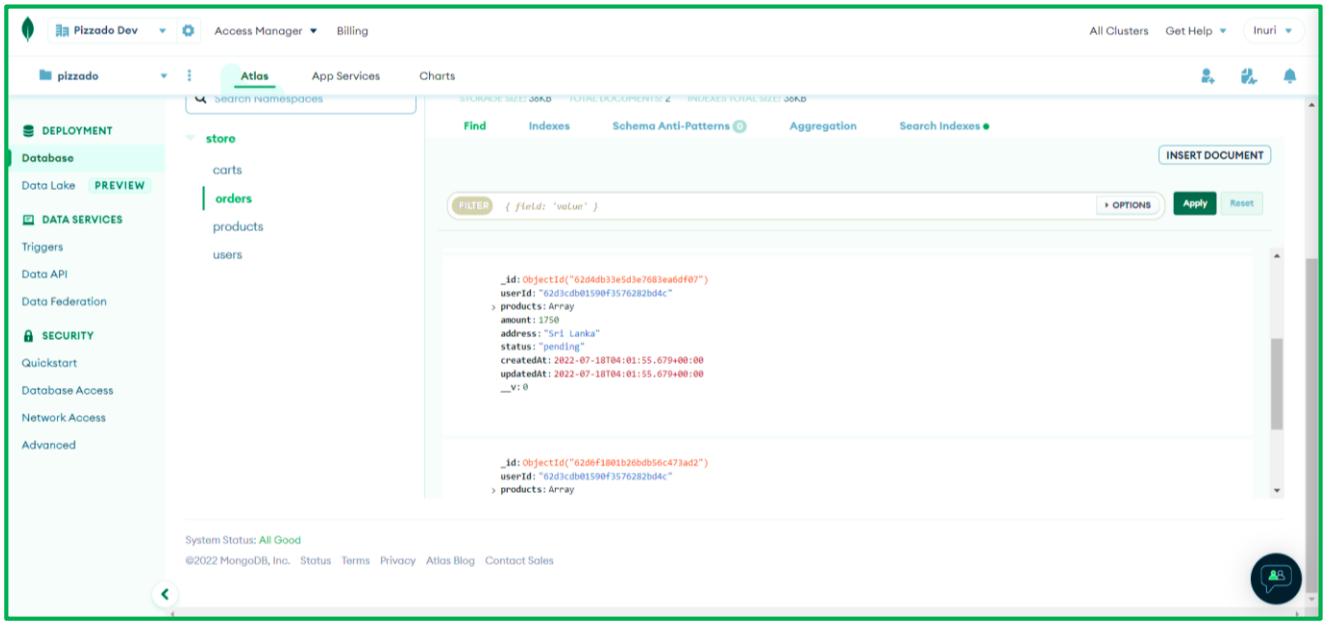
Create a new cluster with preferable credentials. After setting up the cluster I have connect

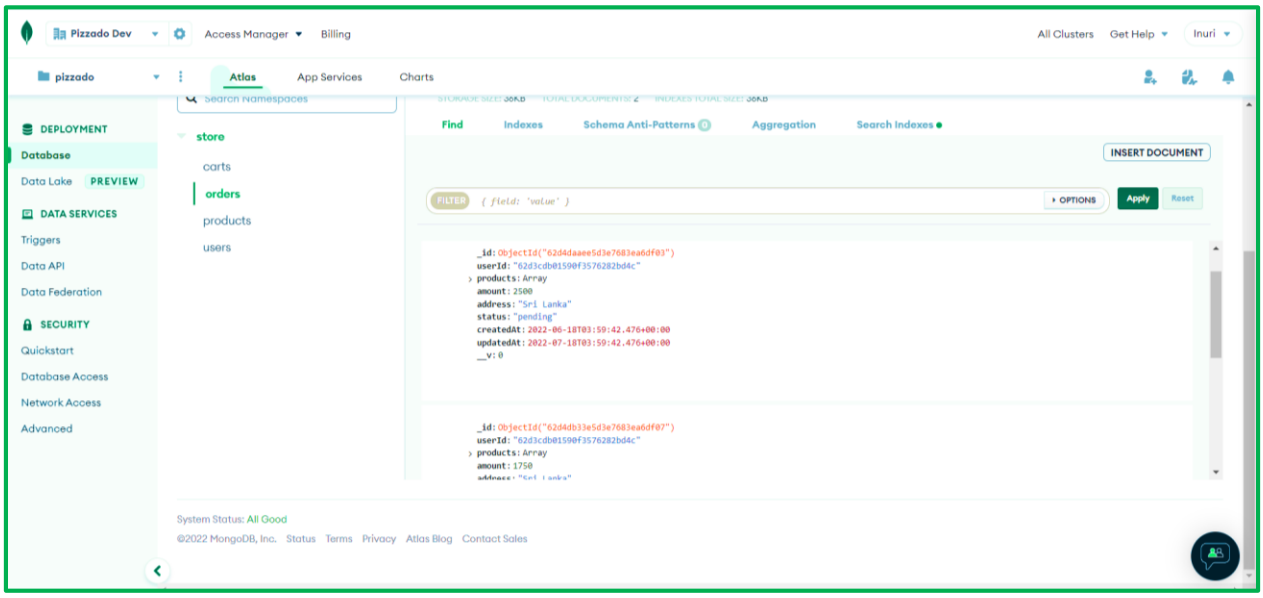
Pizzado-Online Pizza Delivery Application / API to the database (store).



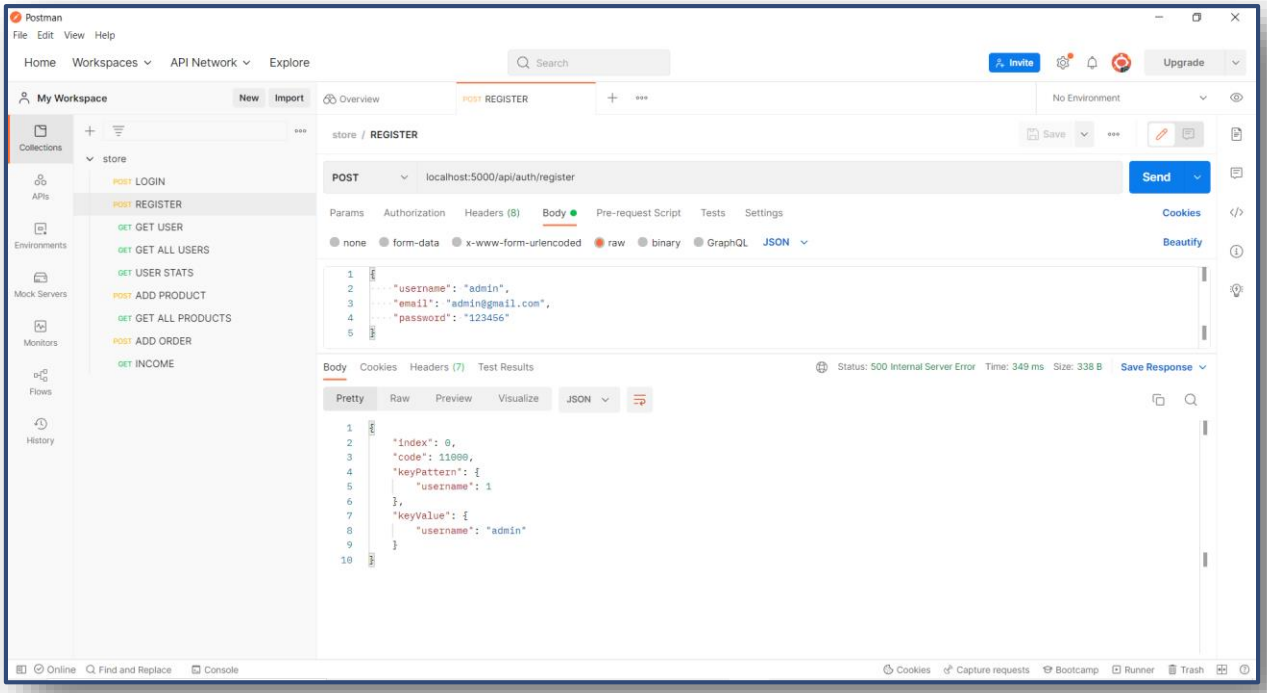

Pizzado Dev
Access Manager
Billing
All Clusters
Get Help
Inuri
pizzado
Atlas
App Services
Charts
DEPLOYMENT
Database
Data Lake
PREVIEW
DATA SERVICES
Triggers
Data API
Data Federation
SECURITY
Quickstart
Database Access
Network Access
Advanced
store
carts
orders
products
users
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
_id: ObjectId("62d3cdb01590f3576282bd4c")
username: "pizzadoupdated"
email: "pizzado@gmail.com"
password: "U2FsdGVkX19G2SAOI3t3McURvalw5kdMOHEhSgpCHcc="
isAdmin: false
createdAt: 2022-06-17T08:52:00.341+00:00
updatedAt: 2022-07-17T12:13:47.907+00:00
__v: 0
_id: ObjectId("62d42435b61687572da0cec0")
username: "admin"
email: "admin@gmail.com"
password: "U2FsdGVkX19yLOZaV+5CKHU859bEdc2St8lYGaclJ7g="
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

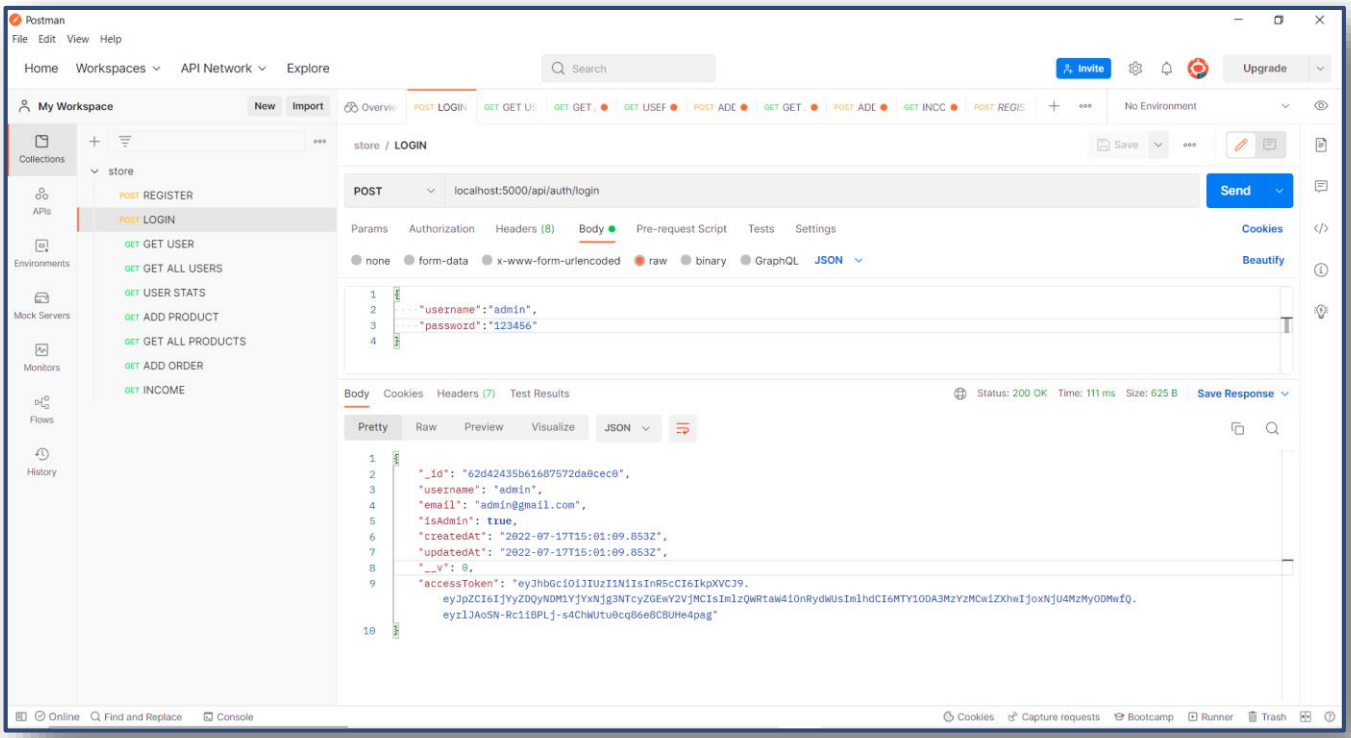

Pizzado Dev
Access Manager
Billing
All Clusters
Get Help
Inuri
pizzado
Atlas
App Services
Charts
DEPLOYMENT
Database
Data Lake
PREVIEW
DATA SERVICES
Triggers
Data API
Data Federation
SECURITY
Quickstart
Database Access
Network Access
Advanced
store
carts
orders
products
users
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
size: "medium"
crust: "pan pizza"
price: 1000
createdAt: 2022-07-17T19:48:23.714+00:00
updatedAt: 2022-07-17T19:48:23.714+00:00
__v: 0
_id: ObjectId("62d46d4fb3c2056a97eb3950")
title: "Cheese Lovers"
desc: "test"
img: "test"
categories: Array
size: "small"
crust: "pan pizza"
price: 740
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

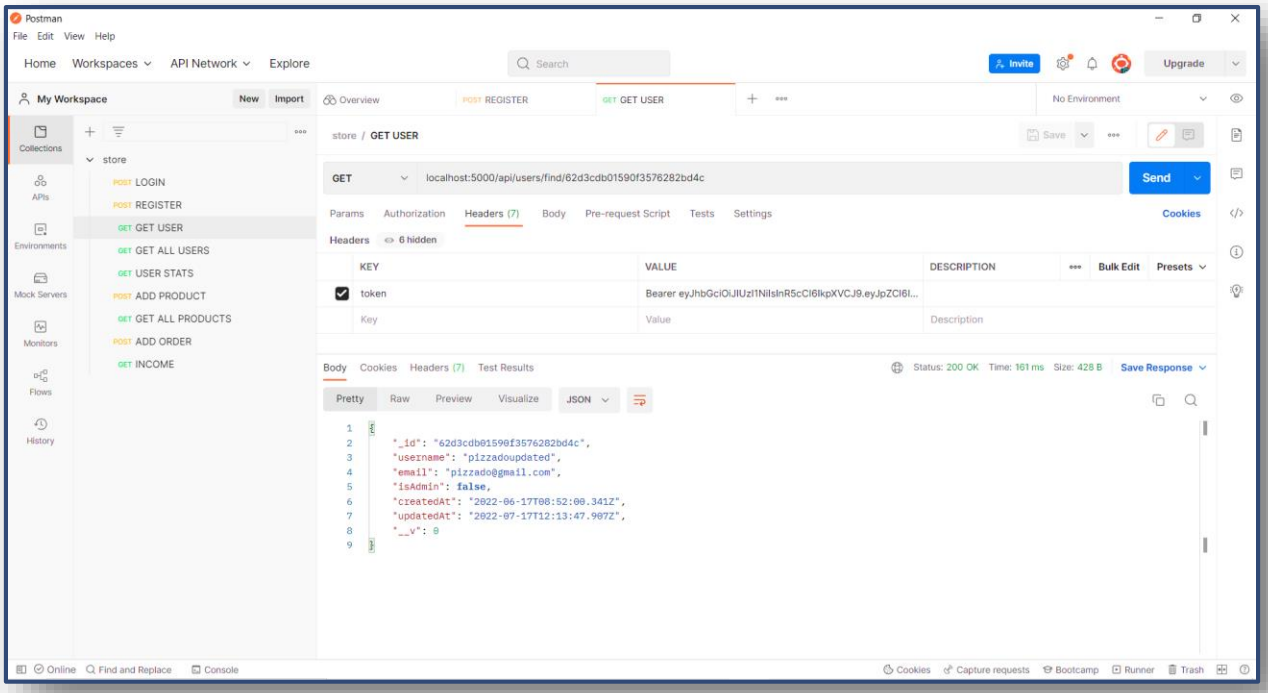

Pizzado Dev
Access Manager
Billing
All Clusters
Get Help
Inuri
pizzado
Atlas
App Services
Charts
DEPLOYMENT
Database
Data Lake
PREVIEW
DATA SERVICES
Triggers
Data API
Data Federation
SECURITY
Quickstart
Database Access
Network Access
Advanced
store
carts
orders
products
users
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
_id: ObjectId("62d4db33e5d3e7683ea6df07")
userId: "62d3cdb01590f3576282bd4c"
products: Array
amount: 1750
address: "Sri Lanka"
status: "pending"
createdAt: 2022-07-18T04:01:55.679+00:00
updatedAt: 2022-07-18T04:01:55.679+00:00
__v: 0
_id: ObjectId("62d6f1001b26bdb56c473ad2")
userId: "62d3cdb01590f3576282bd4c"
products: Array
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

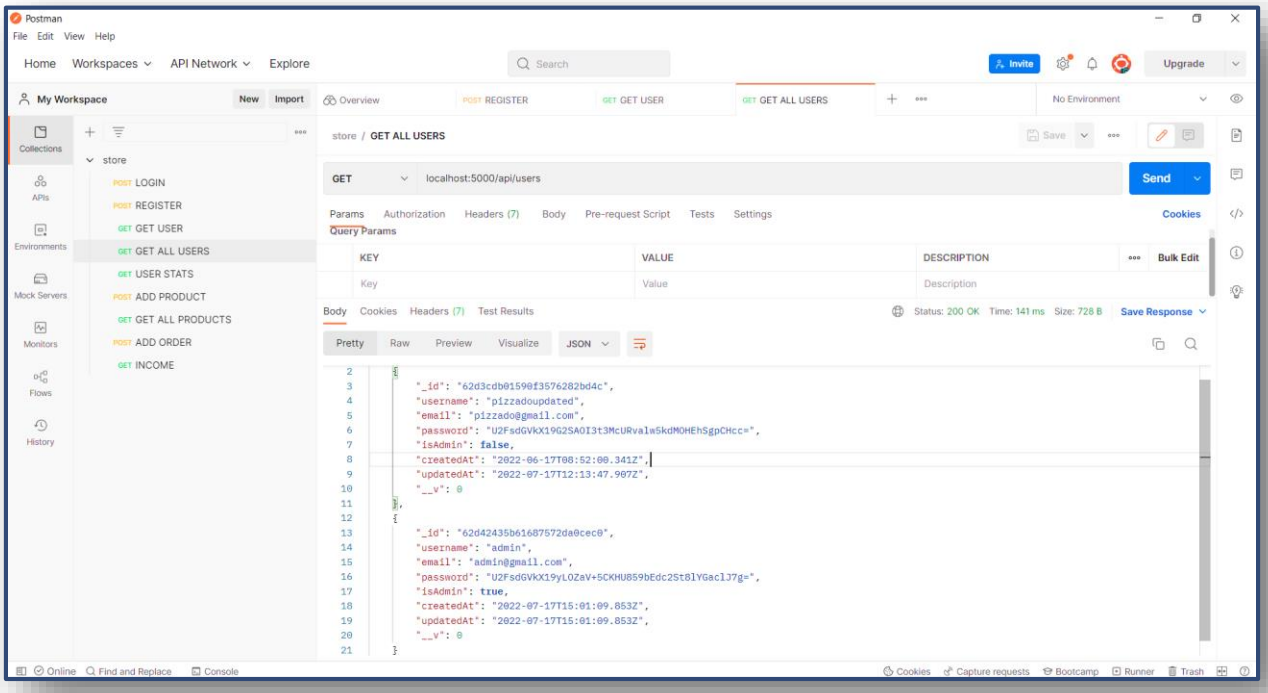

Pizzado Dev
Access Manager
Billing
All Clusters
Get Help
Inuri
pizzado
Atlas
App Services
Charts
DEPLOYMENT
Database
Data Lake
PREVIEW
DATA SERVICES
Triggers
Data API
Data Federation
SECURITY
Quickstart
Database Access
Network Access
Advanced
store
carts
orders
products
users
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
_id: ObjectId("62d4daaee5d3e7683ea6df03")
userId: "62d3cdb01590f3576282bd4c"
products: Array
amount: 2500
address: "Sri Lanka"
status: "pending"
createdAt: 2022-06-18T03:59:42.476+00:00
updatedAt: 2022-07-18T03:59:42.476+00:00
__v: 0
_id: ObjectId("62d4db33e5d3e7683ea6df07")
userId: "62d3cdb01590f3576282bd4c"
products: Array
amount: 1750
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

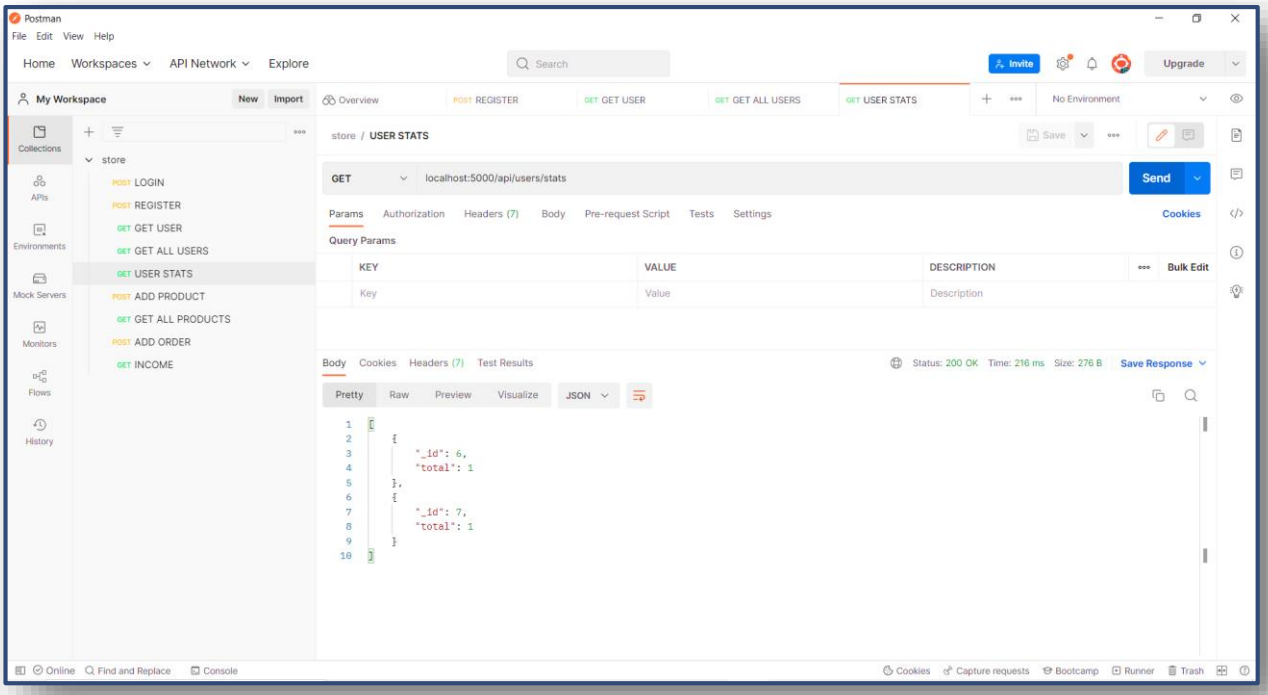
Postman

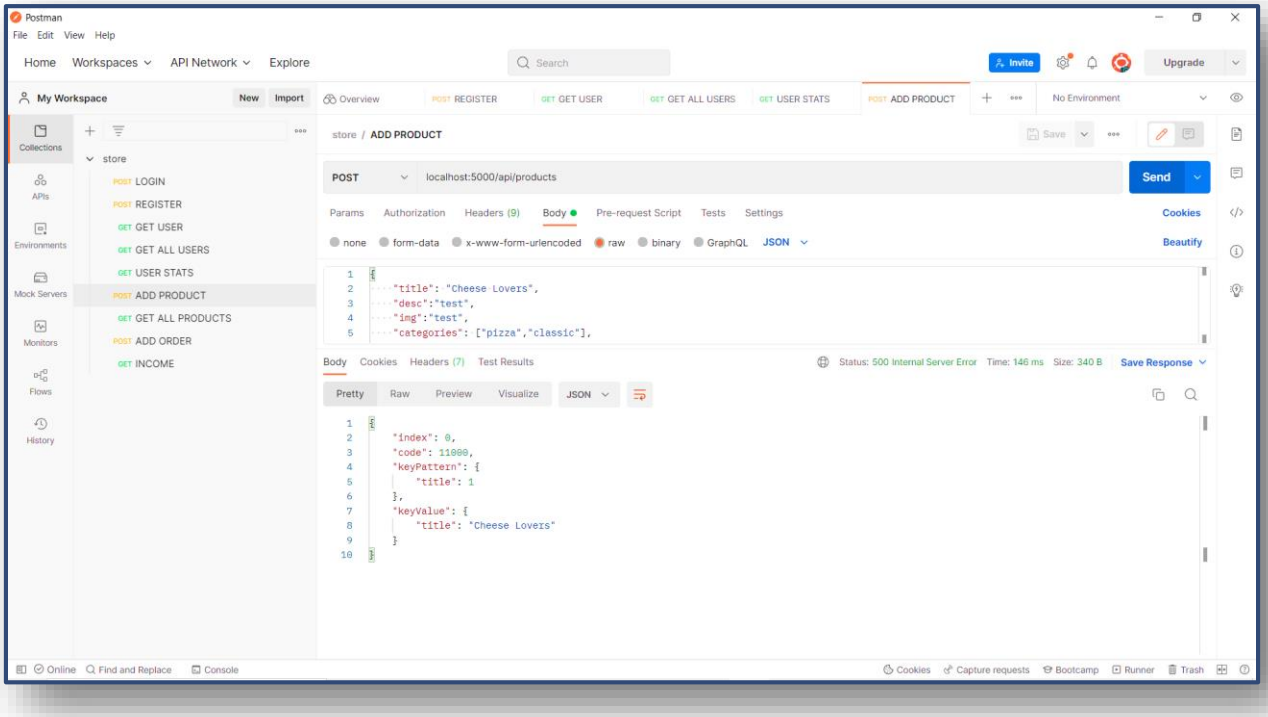

Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST LOGIN
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
POST ADD PRODUCT
GET GET ALL PRODUCTS
POST ADD ORDER
GET INCOME
store / REGISTER
POST
localhost:5000/api/auth/register
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"username": "admin",
"email": "admin@gmail.com",
"password": "123456"
}
Body Cookies Headers (7) Test Results
Status: 500 Internal Server Error Time: 349 ms Size: 338 B
Save Response
Pretty Raw Preview Visualize JSON
{
"index": 0,
"code": 11000,
"keyPattern": {
"username": 1
},
"keyValue": {
"username": "admin"
}
}
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash

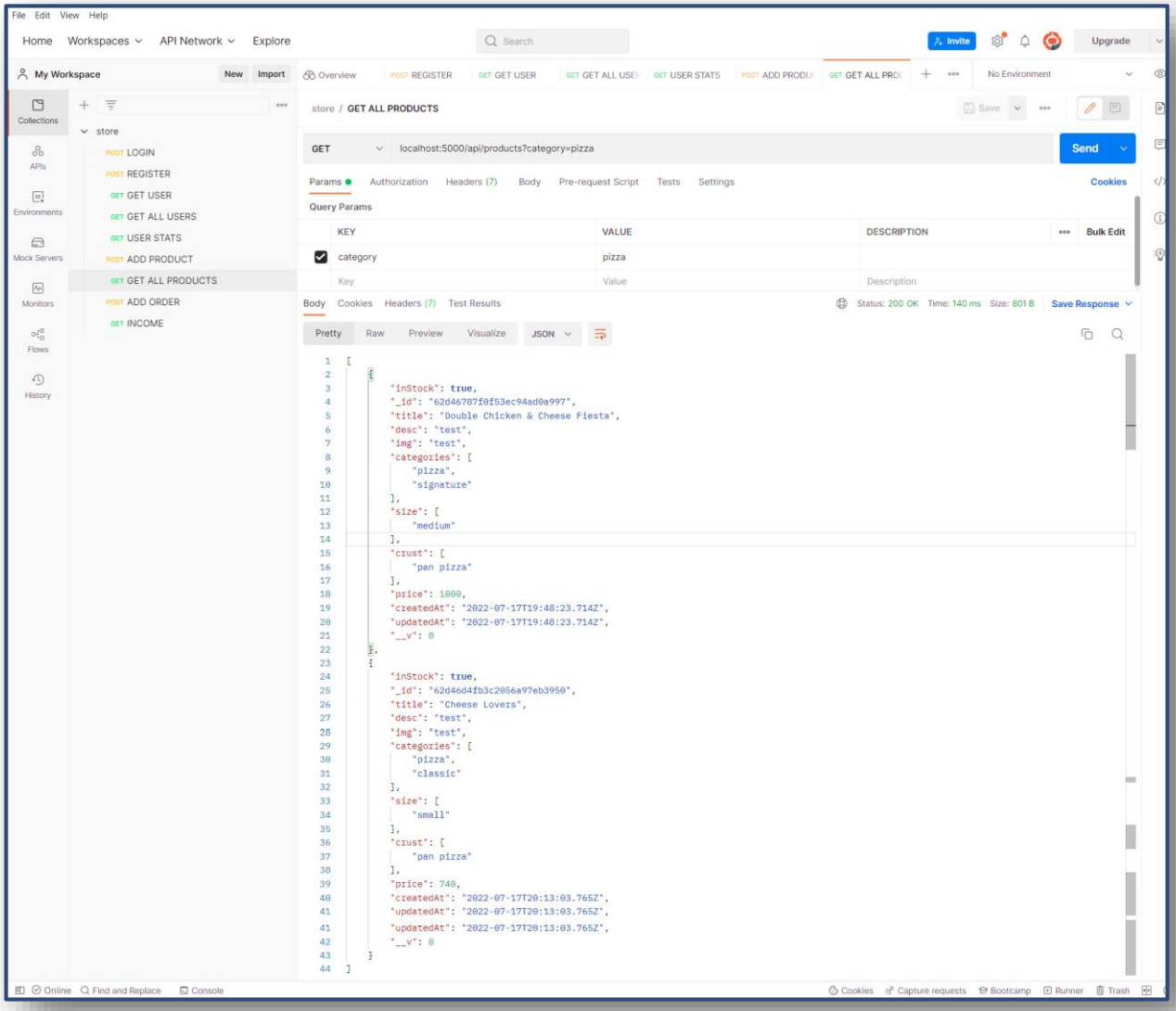

Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST REGISTER
POST LOGIN
GET GET USER
GET GET ALL USERS
GET USER STATS
GET ADD PRODUCT
GET GET ALL PRODUCTS
GET ADD ORDER
GET INCOME
store / LOGIN
No Environment
POST
localhost:5000/api/auth/login
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"username":"admin",
"password":"123456"
}
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 111 ms Size: 625 B
Save Response
Pretty Raw Preview Visualize JSON
{
"_id": "62d42435b61687572da0cec0",
"username": "admin",
"email": "admin@gmail.com",
"isAdmin": true,
"createdAt": "2022-07-17T15:01:09.853Z",
"updatedAt": "2022-07-17T15:01:09.853Z",
"__v": 0,
"accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJpZCI6IjYyZDQyNDM1YjYxNjg3NTcyZGEwY2VjMCIsImlzQWRtaW4iOnRydWUsImlhdCI6MTY1ODA3MzYzMCwiZXhwIjoxNjU4MzMyODMwfQ.
eyrlJAoSN-Rc1iBPLj-s4ChWUtu0cq86e8CBUHe4pag"
}
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash

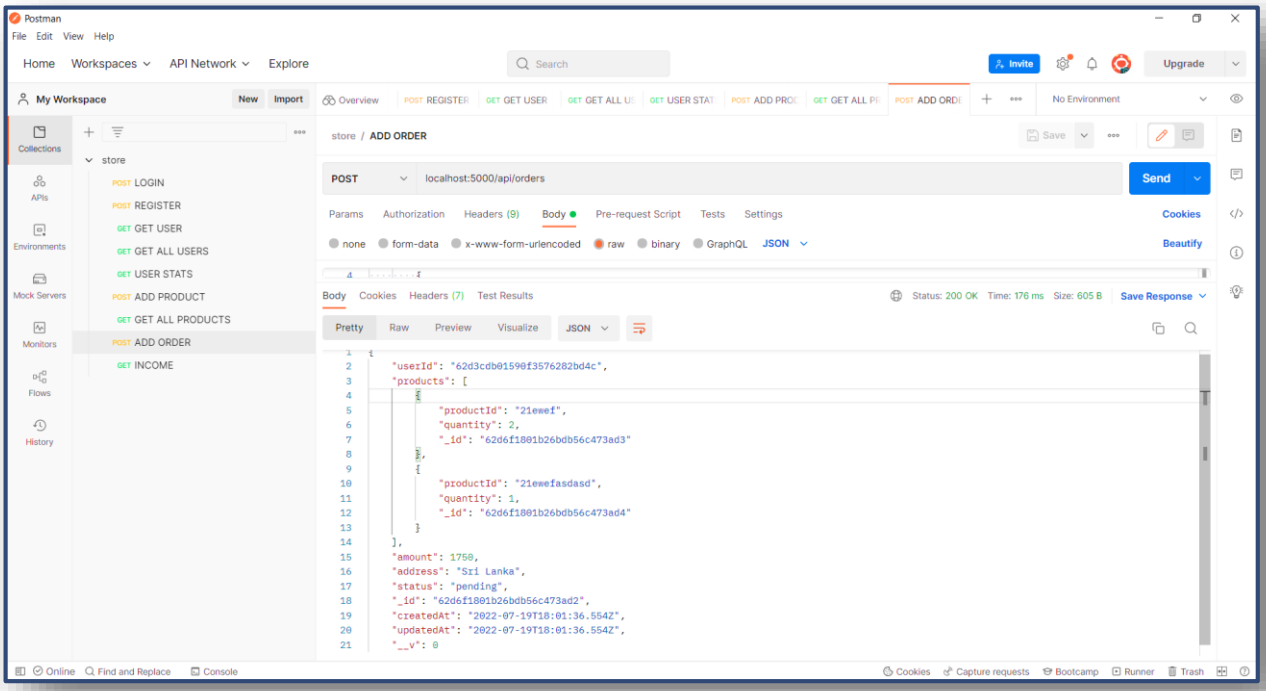

Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Overview
POST REGISTER
GET GET USER
No Environment
store / GET USER
Save
GET
localhost:5000/api/users/find/62d3cdb01590f3576282bd4c
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Cookies
Headers 6 hidden
KEY VALUE DESCRIPTION Bulk Edit Presets
token
Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6I...
Key
Value
Description
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 161 ms Size: 428 B
Save Response
Pretty Raw Preview Visualize JSON
{
"_id": "62d3cdb01590f3576282bd4c",
"username": "pizzadoupdated",
"email": "pizzado@gmail.com",
"isAdmin": false,
"createdAt": "2022-06-17T08:52:00.341Z",
"updatedAt": "2022-07-17T12:13:47.907Z",
"__v": 0
}
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST LOGIN
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
POST ADD PRODUCT
GET GET ALL PRODUCTS
POST ADD ORDER
GET INCOME
Online
Find and Replace
Console
Cookies
Capture requests
Bootcamp
Runner
Trash

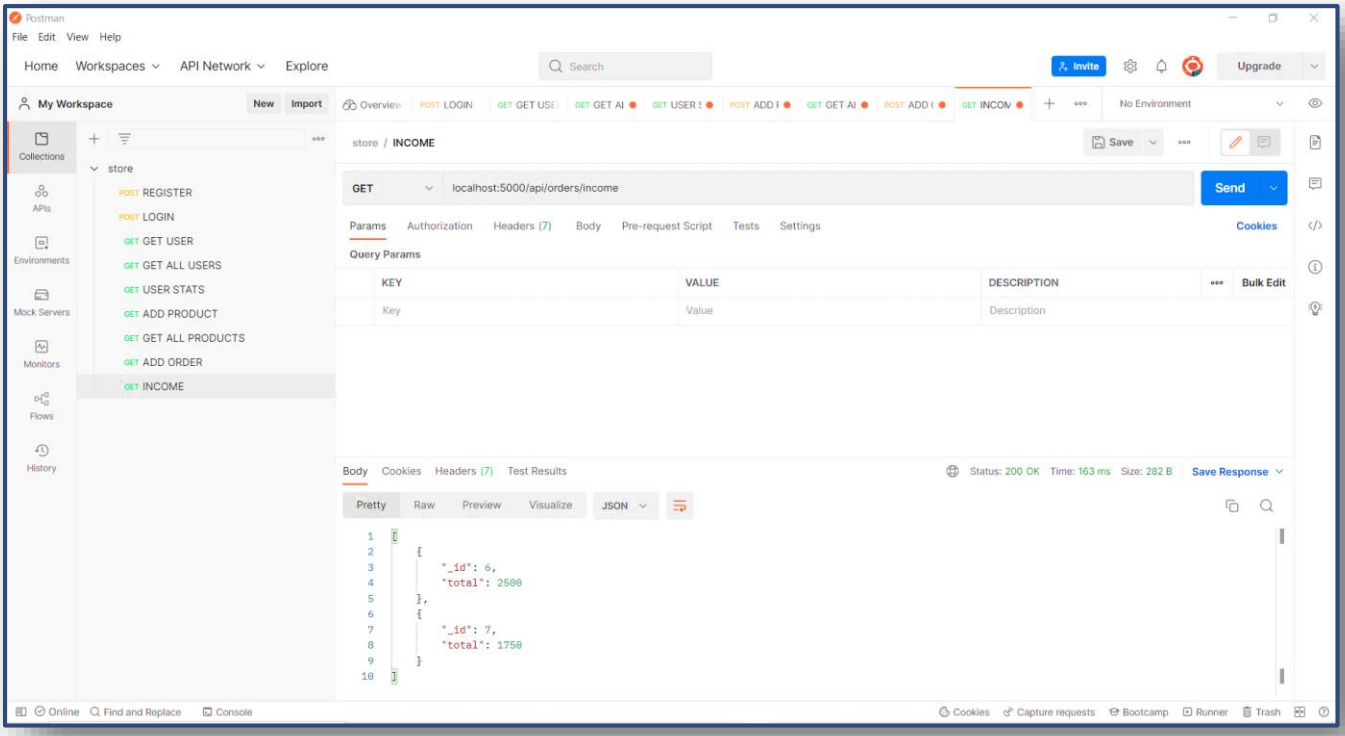

Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Overview
POST REGISTER
GET GET USER
GET GET ALL USERS
No Environment
store / GET ALL USERS
Save
GET
localhost:5000/api/users
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Key
Value
Description
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 141 ms Size: 728 B
Save Response
Pretty Raw Preview Visualize JSON
[
{
"_id": "62d3cdb01590f3576282bd4c",
"username": "pizzadoupdated",
"email": "pizzado@gmail.com",
"password": "U2FsdGVkX19G2SAOI3t3McURvalw5kdMOHEhSgpCHcc=",
"isAdmin": false,
"createdAt": "2022-06-17T08:52:00.341Z",
"updatedAt": "2022-07-17T12:13:47.907Z",
"__v": 0
},
{
"_id": "62d42435b61687572da0cec0",
"username": "admin",
"email": "admin@gmail.com",
"password": "U2FsdGVkX19yLOZaV+5CKHU859bEdc2St8lYGaclJ7g=",
"isAdmin": true,
"createdAt": "2022-07-17T15:01:09.853Z",
"updatedAt": "2022-07-17T15:01:09.853Z",
"__v": 0
}
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST LOGIN
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
POST ADD PRODUCT
GET GET ALL PRODUCTS
POST ADD ORDER
GET INCOME
Online
Find and Replace
Console
Cookies
Capture requests
Bootcamp
Runner
Trash


Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Overview
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST LOGIN
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
POST ADD PRODUCT
GET GET ALL PRODUCTS
POST ADD ORDER
GET INCOME
store / USER STATS
Save
GET
localhost:5000/api/users/stats
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 216 ms Size: 276 B
Save Response
Pretty Raw Preview Visualize JSON
[
{
"_id": 6,
"total": 1
},
{
"_id": 7,
"total": 1
}
]
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash


Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Overview
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
POST ADD PRODUCT
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST LOGIN
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
POST ADD PRODUCT
GET GET ALL PRODUCTS
POST ADD ORDER
GET INCOME
store / ADD PRODUCT
Save
POST
localhost:5000/api/products
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"title": "Cheese Lovers",
"desc":"test",
"img":"test",
"categories": ["pizza","classic"],
}
Body Cookies Headers (7) Test Results
Status: 500 Internal Server Error Time: 146 ms Size: 340 B
Save Response
Pretty Raw Preview Visualize JSON
{
"index": 0,
"code": 11000,
"keyPattern": {
"title": 1
},
"keyValue": {
"title": "Cheese Lovers"
}
}
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash


File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New
Import
Overview
POST REGISTER
GET GET USER
GET GET ALL USER
GET USER STATS
POST ADD PRODU
GET GET ALL PRO
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST LOGIN
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
POST ADD PRODUCT
GET GET ALL PRODUCTS
POST ADD ORDER
GET INCOME
store / GET ALL PRODUCTS
Save
GET
localhost:5000/api/products?category=pizza
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
category
pizza
Key
Value
Description
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 140 ms
Size: 801 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"inStock": true,
"_id": "62d46787f0f53ec94ad0a997",
"title": "Double Chicken & Cheese Fiesta",
"desc": "test",
"img": "test",
"categories": [
"pizza",
"signature"
],
"size": [
"medium"
],
"crust": [
"pan pizza"
],
"price": 1000,
"createdAt": "2022-07-17T19:48:23.714Z",
"updatedAt": "2022-07-17T19:48:23.714Z",
"__v": 0
},
{
"inStock": true,
"_id": "62d46d4fb3c2056a97eb3950",
"title": "Cheese Lovers",
"desc": "test",
"img": "test",
"categories": [
"pizza",
"classic"
],
"size": [
"small"
],
"crust": [
"pan pizza"
],
"price": 740,
"createdAt": "2022-07-17T20:13:03.765Z",
"updatedAt": "2022-07-17T20:13:03.765Z",
"__v": 0
}
]
Online
Find and Replace
Console
Cookies
Capture requests
Bootcamp
Runner
Trash


Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST LOGIN
POST REGISTER
GET GET USER
GET GET ALL USERS
GET USER STATS
POST ADD PRODUCT
GET GET ALL PRODUCTS
POST ADD ORDER
GET INCOME
store / ADD ORDER
POST
localhost:5000/api/orders
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 176 ms Size: 605 B
Save Response
Pretty Raw Preview Visualize JSON
{
"userId": "62d3cdb01590f3576282bd4c",
"products": [
{
"productId": "21ewef",
"quantity": 2,
"_id": "62d6f1801b26bdb56c473ad3"
},
{
"productId": "21ewefasdasd",
"quantity": 1,
"_id": "62d6f1801b26bdb56c473ad4"
}
],
"amount": 1750,
"address": "Sri Lanka",
"status": "pending",
"_id": "62d6f1801b26bdb56c473ad2",
"createdAt": "2022-07-19T18:01:36.554Z",
"updatedAt": "2022-07-19T18:01:36.554Z",
"__v": 0
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash


Postman
File Edit View Help
Home Workspaces API Network Explore
Search
Invite
Upgrade
My Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
store
POST REGISTER
POST LOGIN
GET GET USER
GET GET ALL USERS
GET USER STATS
GET ADD PRODUCT
GET GET ALL PRODUCTS
GET ADD ORDER
GET INCOME
store / INCOME
Save
GET
localhost:5000/api/orders/income
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Key Value Description
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 163 ms Size: 282 B
Save Response
Pretty Raw Preview Visualize JSON
[
{
"_id": 6,
"total": 2500
},
{
"_id": 7,
"total": 1750
}
]
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash

Summary

An online pizza ordering system is developed where the customers can make an order for the food and avoid the hassles of waiting for the order to be taken by the waiter. Using the application, the end users register online, read the E-menu card and select the food from the e-menu card to order food online. Once the customer selects the required food item the chef will be able to see the results on the screen and start processing the food. This application nullifies the need of a waiter or reduces the workload of the waiter. The advantage is that in a crowded restaurant there will be chances that the waiters are overloaded with orders and they are unable to meet the requirements of the customer in a satisfactory manner. Therefore by using this application, the users can directly place the order for food to the chef online.

The system also enables the restaurant to know the items available in real time and make changes to their food and beverage inventory based on the orders placed and the orders completed.

REFERENCES

[1]

"Build software better, together," *GitHub*. https://github.com/topics/pizza?l=javascript (accessed Jul. 19, 2022).

[2]
"#32 Adding New Pizza Functionality | Mern Project Tutorials (Hindi/Urdu)," *www.youtube.com*. https://www.youtube.com/watch?v=nv4vOKOcGiE (accessed Jul. 19, 2022).

[3]
"Stripe Login | Sign in to the Stripe Dashboard," *Stripe.com*, 2020. https://dashboard.stripe.com/test/apikeys

[4]

"crypto-js," *npm*, Dec. 14, 2016. https://www.npmjs.com/package/crypto-js