Yunnan University

Big Data Final Project Group 03

MINHAZUL ISLAM 20193290764

RASHIK JAHANGIR 20193290734

YEASIN ARAFAT 20193290711

MASHIUR RAHMAN 20183290516

Abstract

In this project we will build a e-commerce website.Here we will use Big data technology.In this e-commerce site we will have a lots of features ,we will make login page,registration page,customer able to aces without registration when customer want to buy product they must have to register.Customer can add product to cart also delete option available here.In cart section customer can increase product number and total amount of money are also visible in cart section.From cart section customer able to create invoice here we will collect customer address phone number for delivery.Admin can add more product with category also picture ,delete product,product edit and update option are also available.Our website name is UNDER WORLD

Project Description

Technology used

1. **Front end**
   - **1.1. HTML**
   - **1.2. CSS**
   - **1.3. React**
2. **Back end**
   - **2.1. Node JS**
   - **2.2. Express JS**
   - **2.3. Nodemon**
   - **2.4. Body parser**
3. **Database**
   - **3.1. MongoDb(main database)**
   - **3.2. Redis(caching database)**

User Interface

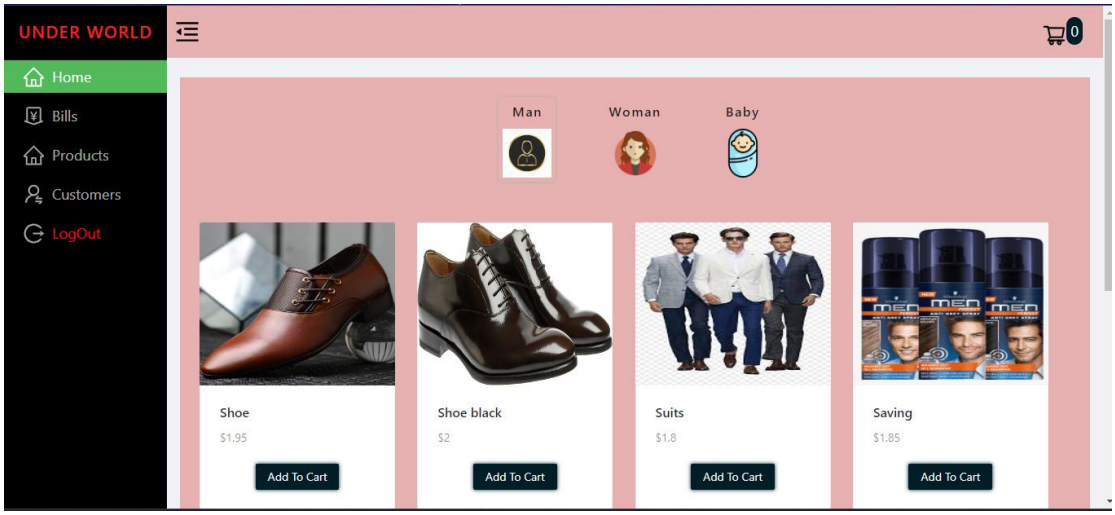
Dependencies: for the front end site

```
"dependencies": {
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.2.0",
    "@testing-library/user-event": "^13.5.0",
    "antd": "^4.20.6",
    "axios": "^0.27.2",
    "react": "^18.1.0",
    "react-dom": "^18.1.0",
    "react-redux": "^8.0.2",
    "react-router-dom": "^6.3.0",
    "react-scripts": "5.0.1",
    "react-to-print": "^2.14.7",
    "redux": "^4.2.0",
    "redux-devtools-extension": "^2.13.9",
    "redux-thunk": "^2.4.1",
    "web-vitals": "^2.1.4"
  },
```

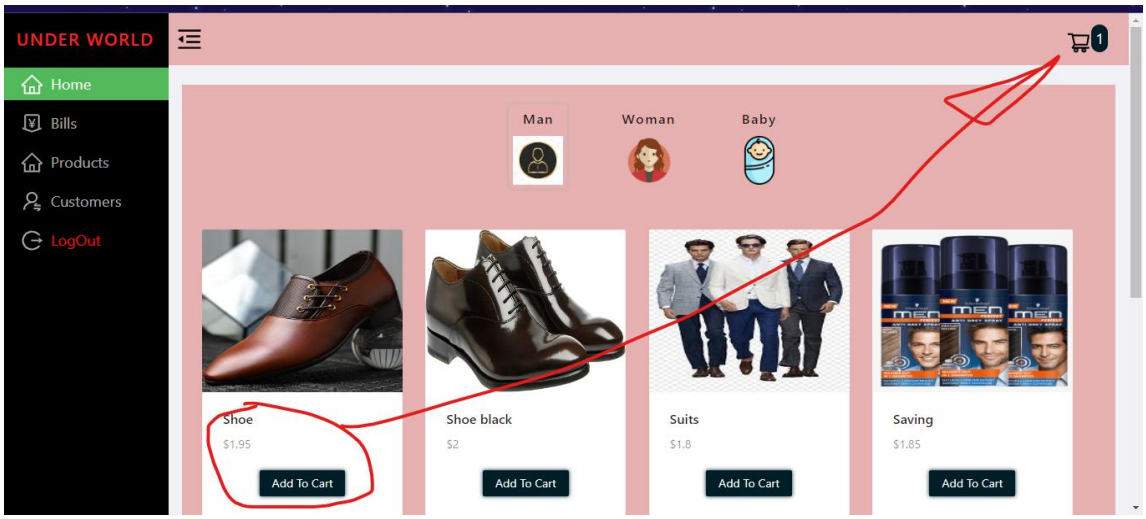
Registration: In this page customer able to create a account with their name .User also can chose a user id and password.

Login: In this section customer able to login on site with User ID and Password.Also customer can visit website with out creating a account because if they chose any product then customer can create a account.

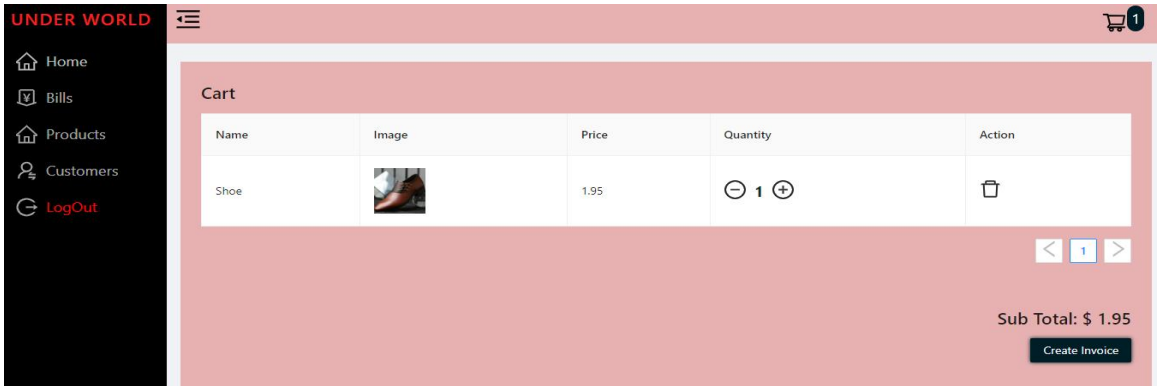
Home: It is the first page after login to website.In the left side user can find navigation bar where we have Home,Bills,Product,Customer and Logout option.In the center point we have three category Man ,Woman and Baby.when customer click any category user can see product according the category.



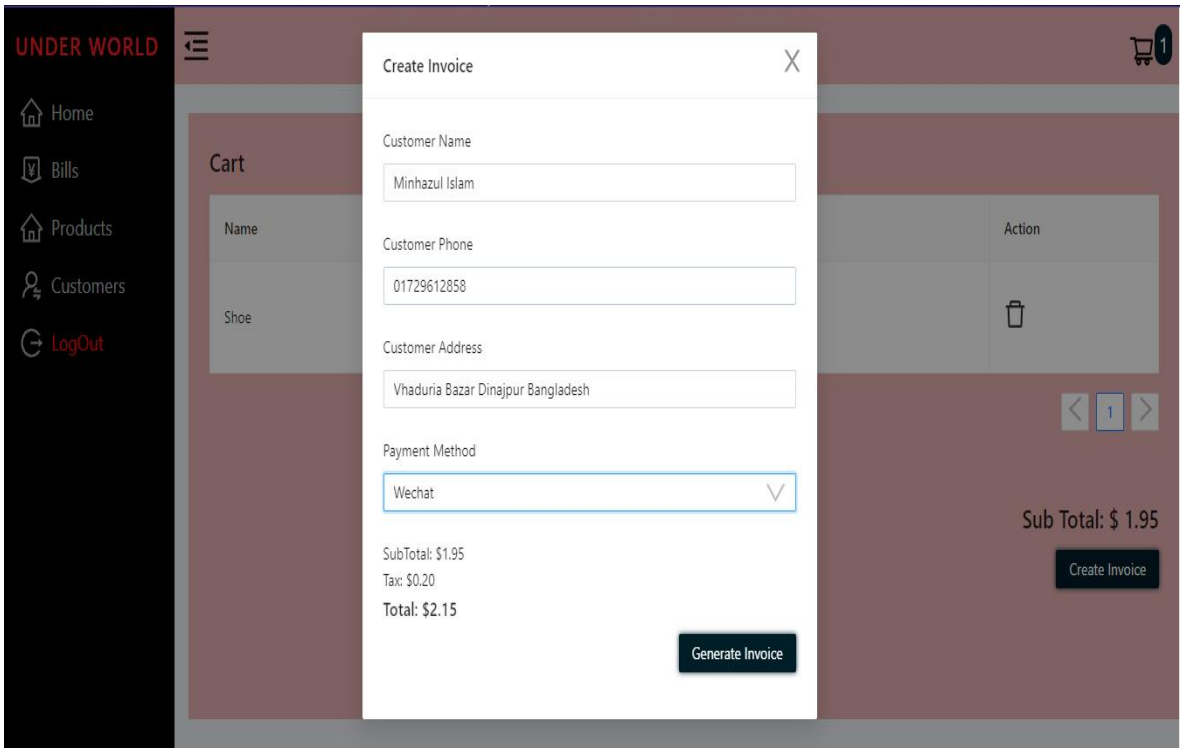
Add To Cart: When user click on Add to Cart product will add on the cart.



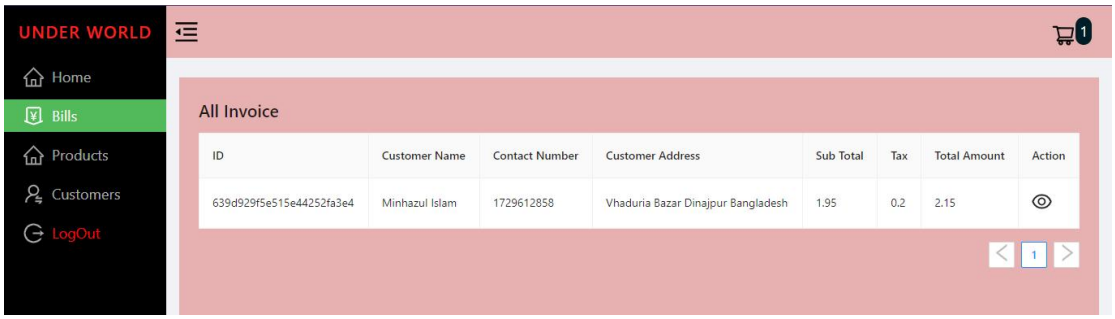
In cart section customer can increase or decrease quantity.Here also can see total amount of money and also create invoice.



Create Invoice: To create invoice we will collect customer name phone number customer address and also select payment method here customer can select payment option wechat/cash/card.Here we will count tax also.



Bills: In the bills section from navigation bar.Here customer can see All invoice number.All invoice has a unique id.



Hereafter click on Action customer can invoice in details.And also able to download and print option

ID	Customer Name	Contact Number	Customer Address	Sub Total	Tax	Total Amount	Action
639d929f5e515e44252fa3e4	Minhazul Islam	1729612858	Vhaduria Bazar Dinajpur Bangladesh	1.95	0.2	2.15	

Invoice Details

UNDER WORLD

Number:*8801780203857
Address:Dhaka,Bangladesh

Customer Name:	Minhazul Islam
Customer Phone:	1729612858
Customer Address:	Vhaduria Bazar Dinajpur Bangladesh
Date Order:	2022-12-17
Total Amount:	$2.15

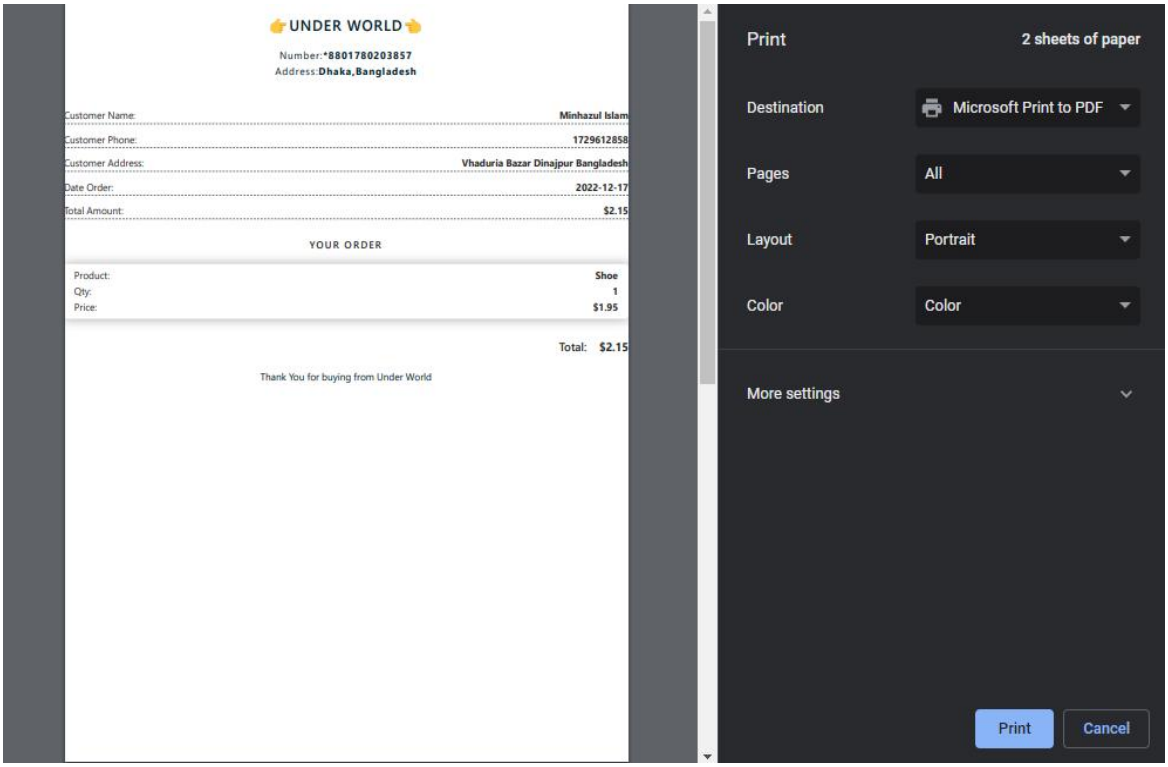
YOUR ORDER

Product:	Shoe
Qty:	1
Price:	$1.95

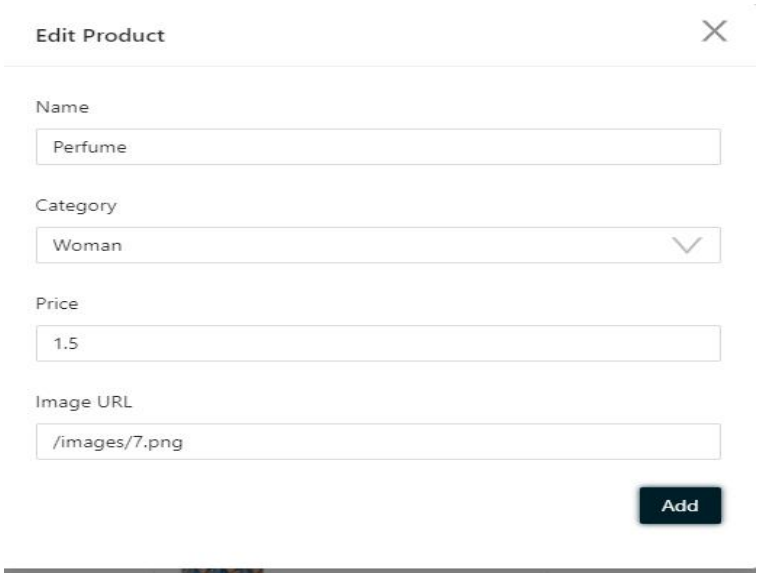
Total: $2.15

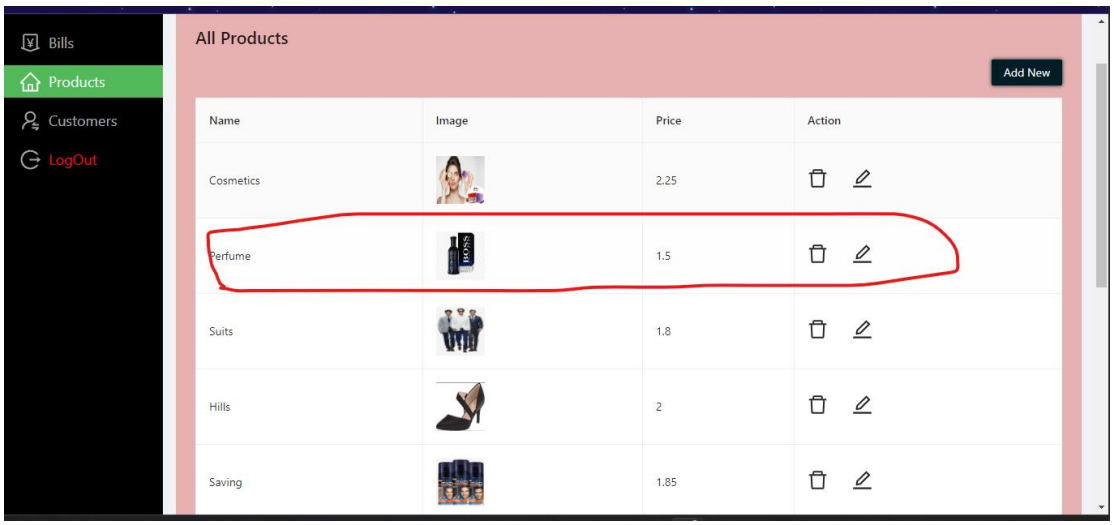
Thank You for buying from Under World

Generate Invoice

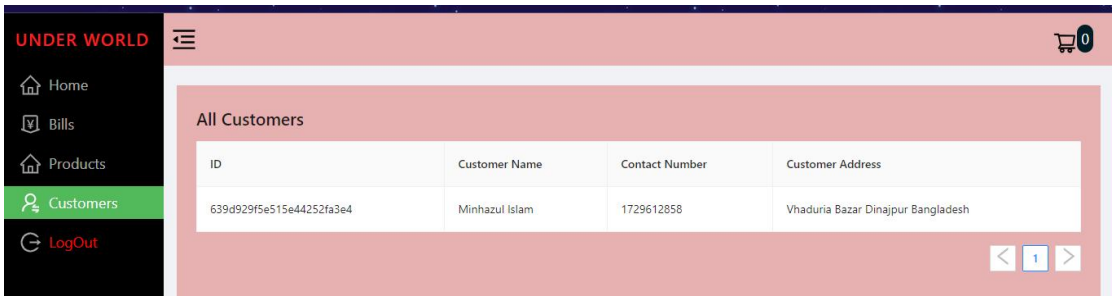


Products: In this part admin can add new product,update information also delete feature available.





Customers: In this section able to see how many customer.Every customer has unique id and also can see contact number and address also.



Back-end

Dependencies:

```
{
  "name": "pos",
  "type": "module",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "server": "nodemon server.js",
    "frontend": "npm start --prefix frontend",
    "dev": "concurrently \"npm run server\" \"npm run frontend\"",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Minhazul Islam",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.0",
    "color": "^4.2.3",
    "concurrently": "^7.2.1",
    "cors": "^2.8.5",
    "dotenv": "^16.0.1",
    "express": "^4.18.1",
    "mongoose": "^6.3.4",
    "morgan": "^1.10.0",
    "redis": "^4.5.1"
  },
  "devDependencies": {
    "nodemon": "^2.0.16"
  }
}
```

Server Description

server.js is the main page from this page most of the function will control. At first I import necessary dependencies.

```
import express from 'express';
import bodyParser from 'body-parser';
import cors from 'cors';
import morgan from 'morgan';
import dotenv from 'dotenv';
import mongoose from 'mongoose';
import productRouter from './routes/productsRoutes.js';
import userRouter from './routes/userRoutes.js';
import billsRouter from './routes/billsRoutes.js';
//require('colors');
```

Now I create port for this project I used 5000

```
const PORT = process.env.PORT || 5000;

//Listen
app.listen(PORT, () => {
    console.log(`Serve at running on the port: http://localhost:${PORT}`);
} )
```

Then I connect this apps with mongodb data base

```
mongoose.connect(process.env.MONGODB_URI).then(() => {
    console.log("Connected to DB");
}).catch((err) => {
    console.log(err.message);
});
```

From this page I called middleware and route

```
//middlewares
app.use(cors());
app.use(express.json());

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: false}))
app.use(morgan("dev"));

//routes
app.use('/api/products/', productRouter);
app.use('/api/users/', userRouter);
app.use('/api/bills/', billsRouter);
```

- In this project I used Redis As cache server . Fist time when I run server server collect data from main database mongodb.But when second time when call same api this time data come from redis database.

- At firs api will check this data is available in redis if data in redis then api collect from redis if data are not available in redis then api collect data from mongodb .

- After running server if admin add any product this time I insert data in redis and also mongodb.
- We will cache data in redis for 60 minutes,That mens redis data timeout 60.

```javascript
import Product from "../models/productModel.js";

import redis from 'redis';

export const client = redis.createClient();


//for add or fetch
export const getProductController = async (req, res) => {
    try {
        const products = await Product.find();
        res.status(200).send(products);
    } catch(error) {
        console.log(error);
    }
}

export const redis_post = (req, res, next) => {
    client.get('newProducts', (err, redis_data) => {
        if (err) {
            throw err;
            next();
        } else if (redis_data) {
            redis.send(JSON.parse(redis_data));
        } else {
            next();
        }
})
}

//for add
export const addProductController = async (req, res) => {

    try {

        const newProducts = new Product(req.body);
        await newProducts.save();
        res.status(200).send("Products Created Successfully!");
        redis.setex('newProducts',60,JSON.stringify(post));

    } catch(error) {
        console.log(error);
    }

}
```

```javascript
import express from "express";
import { addProductController, deleteProductController, getProductController, redis_post, updateProductController } from "../controllers/productController.js";

const productRouter = express.Router();

productRouter.get("/getproducts", redis_post,getProductController);//getCachData,

productRouter.post("/addproducts",addProductController);

productRouter.put("/updateproducts", updateProductController);

productRouter.post("/deleteproducts", deleteProductController);

export default productRouter;
```

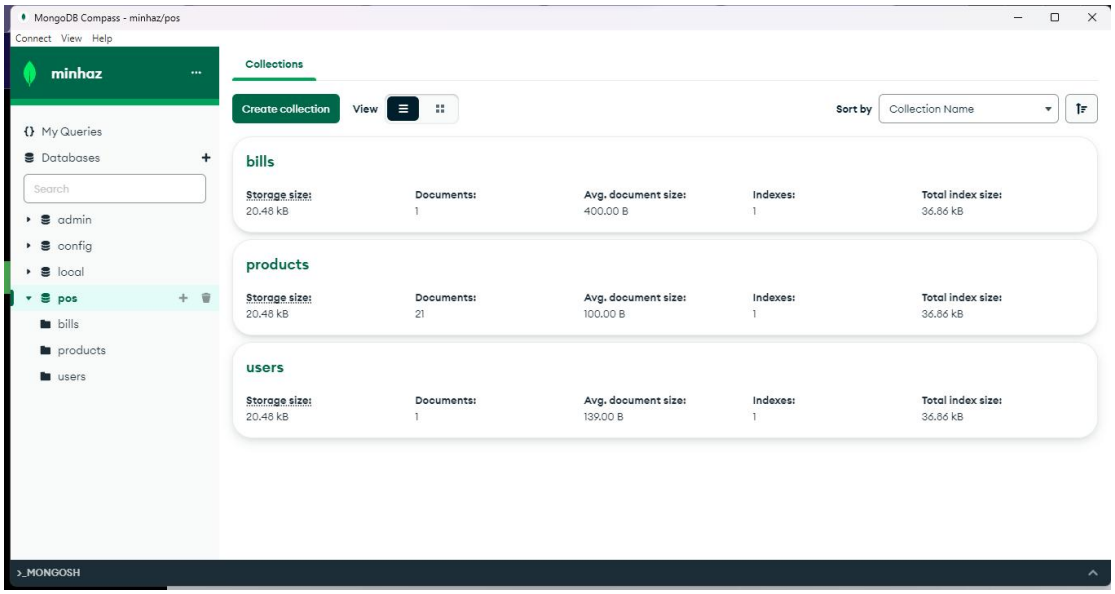
```javascript
export const redis_user_post = (req, res, next) => {
    client.post('newUser', (err, redis_data) => {
        if (err) {
            throw err;
            next();
        } else if (redis_data) {
            redis.send(JSON.parse(redis_data));
        } else {
            next();
        }
})
}

//for register
export const registerController = async (req, res) => {

    try {

        const newUser = new User({ ...req.body, verified: true});
        await newUser.save();
        res.status(200).send("New User Added Successfully!");
        redis.setex('newUser',60,JSON.stringify(post));

    } catch(error) {
        console.log(error);
    }

}
```

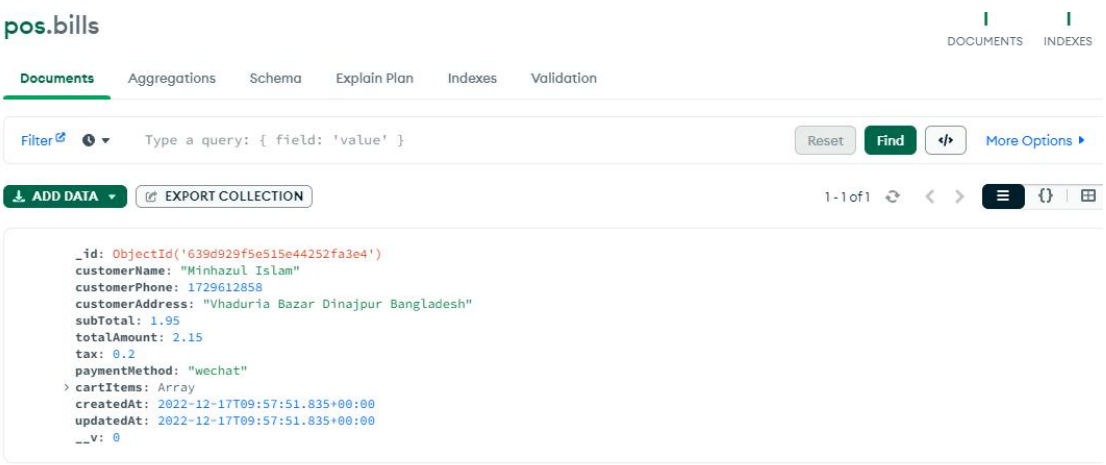
Database

For this project we used mongodb as main server.

Database url:mongodb+srv://minhaz602:<password>@cluster0.rj44c5b.mongodb.net/

In our mongodb database we create three collection bills,product and users.





pos.products

21 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT COLLECTION 1-20 of 21

products

	_id ObjectId	name String	category String	price Mixed	image String	updatedAt Date
1	ObjectId('6393bbbb6fe73ac4952…	"Cosmetic "	"woman"	2.25	"/images/12.png"	No field
2	ObjectId('6393bbbb6fe73ac4952…	"Baby Food"	"baby"	1.8	"/images/27.png"	No field
3	ObjectId('6393bbbb6fe73ac4952…	"Baby car"	"baby"	2.2	"/images/26.png"	No field
4	ObjectId('6393bbbb6fe73ac4952…	"Wills red"	"woman"	2.4	"/images/18.png"	No field
5	ObjectId('6393bbbb6fe73ac4952…	"Shoe"	"man"	1.95	"/images/8.png"	No field
6	ObjectId('6393bbbb6fe73ac4952…	"Baby Cloth"	"baby"	1.7	"/images/24.png"	No field
7	ObjectId('6393bbbb6fe73ac4952…	"Bag Red"	"woman"	2.5	"/images/19.png"	No field
8	ObjectId('6393bbbb6fe73ac4952…	"Perfume "	"woman"	1.75	"/images/6.png"	No field
9	ObjectId('6393bbbb6fe73ac4952…	"Johnson Shampoo "	"baby"	2.2	"/images/25.png"	No field
10	ObjectId('6393bbbb6fe73ac4952…	"Shoe black"	"man"	2	"/images/9.png"	No field
11	ObjectId('6393bbbb6fe73ac4952…	"Cosmetics "	"woman"	2.25	"/images/11.png"	No field
12	ObjectId('6393bbbb6fe73ac4952…	"Perfume "	"woman"	1.5	"/images/7.png"	2022-12-17T13:25:14.590+00:00
13	ObjectId('6393bbbb6fe73ac4952…	"Suits"	"man"	1.8	"/images/5.png"	No field
14	ObjectId('6393bbbb6fe73ac4952…	"Wills "	"woman"	2	"/images/16.png"	No field
15	ObjectId('6393bbbb6fe73ac4952…	"Saving"	"man"	1.85	"/images/2.png"	No field
16	ObjectId('6393bbbb6fe73ac4952…	"Pant"	"man"	2.5	"/images/4.png"	No field
17	ObjectId('6393bbbb6fe73ac4952…	"Watch"	"man"	1.95	"/images/3.png"	No field
18	ObjectId('6393bbbb6fe73ac4952…	"Cap"	"man"	2	"/images/10.png"	No field
19	ObjectId('6393bbbb6fe73ac4952…	"Side Bag"	"woman"	1.9	"/images/20.png"	No field
20	ObjectId('6393bbbb6fe73ac4952…	"Dipper "	"baby"	2.2	"/images/30.png"	No field

pos.users

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' }

ADD DATA EXPORT COLLECTION

```
_id: ObjectId('639b6e24d61bfa62b74d11d9')
name: "Minhazul Islam"
userId: "100"
password: "100"
verified: true
createdAt: 2022-12-15T18:57:40.381+00:00
updatedAt: 2022-12-15T18:57:40.381+00:00
__v: 0
```