Melly Fashion – E-Commerce Website

Group 04

AF/20/16733	M.L.H.Raveena
AF/20/16778	L.D.D.I.Jayathilake
AF/20/16373	T.K.Ranasinghe
AF/20/16359	W.L.G.N.Kaushalya

BSc (Hons.) in Information Technology

ICT 3113 – Web Application Development

Course Director – Mr. Samitha Nanayakkara

Department of Information and Communication Technology
Faculty of Humanities and Social Sciences
University of Sri Jayewardenepura

20.08.2024

Acknowledgement

It is indeed with great pleasure and an immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to our Course Director Mr. Samitha Nanayakkara for the facilities provided to accomplish this ICT 3113 - Web Application Development final project. He has been a constant source of inspiration for us and we are very deeply thankful to him for his support and valuable advice. We are extremely grateful to our Department of Information Technology staff members, Lab technicians and Non-academic staff members for their extreme help throughout our project. Finally, we express our heart full thanks to all of our friends who helped us complete this project.

Table of Contents

List of Figures

List of Code Snippets

1. Executive Summary

Melly Fashion is a strong and active business focused on various aspects of the clothing trading market that occurs on the Internet. The platform was built with lots of attention to details with the help of the latest web technologies including TypeScript, React, Nodejs, Express.js and MongoDB to create a nimble outlet and one that delivers the ultimate user experience.

User Authentication: It also has an integrated user identification system for customers, which enables them to register, log in to their accounts, and personalize their respective profiles. To completely prohibit user data leakage and guarantee its safety and confidentiality, this authentication system complies with requirements like Auth.

Product Browsing and Search: Customers can access a vast range of clothing products and navigate through them with ease and efficiency.

Cart Management: The cart setup enables the addition, deletion, and modification of products by the users.

Order Placement: Melly Fashion helps users to place orders in a proper sequence where a customer can view the order details, select delivery options and make payments through payment gateways. The moment a user places an order, he or she is able to monitor progress from when the order is being processed to when it is delivered and, therefore, there is trust in the platform.

Backend APIs: This is the back end of the platform, developed using Nodejs, also offer a set of RESTful APIs to implement basic features and functionalities. These are among the key requests like getting product data, handling sessions, placing orders, and implementing payment gateways. The APIs are intended to have scalable and efficient interaction between the frontend and the backend.

Frontend: The user interface is implemented using React and TypeScript to provide a modern, responsive, and type-safe environment for the user.

The UI/UX design of the interface is simple as well as efficient, copying the simplicity of outstanding e-commerce models power the backend, offer the solid and scalable server environment to manage the requests, store the users' data and process the orders.

It also uses MongoDB, a NoSQL database, to store as product details, user profile and order databases, to enhance flexibility of data input.

Deployment: The Melly Fashion blog uses Heroku for backend hosting and Netlify for frontend hosting. The backend host link is added to the frontend before deployment. The deployment strategy follows Continuous Integration and Continuous Deployment (CI/CD), allowing the application to be updated and modified frequently through the development of new features. This setup ensures reliability, scalability, and security in a modern cloud environment.

2. Introduction

This study is regarding Melly Fashion which is in the clothing retailing business, and is an e-commerce solutions provider designed for the industry. This website was created with the needs of today’s fast-paced online shopping, where usability and stylish design co-exist to give customers the best experience possible.

2.1. Project Overview

Melly Fashion is specifically designed as an e-commerce site for the clothing retail industry to offer elegant and convenient online shopping experience. It takes advantage of state of the art web technologies to provide the contemporary online consumers with goods that meets their expectations with regards to functionality and aesthetics.

TypeScript is used when type-safe contractions of scripts are needed in JavaScript, which means better code quality and less bugs on the development stage.

React is responsible for front-end development and supports developing dynamic and responsive interfaces for enhanced user experience.

Node.js and **Express.js** are instrumental in functioning on the server side and offers a strong foundation for the pivotal aspects of server request handling and-API management at scale.

MongoDB stands for the response to database needs, providing freedom and high efficiency while working with all the product data, user info, and transaction histories.

It is crucial to make Melly Fashion merchant website friendly to the consumer regardless of the device, whether it is a desktop, portable, or a mobile device. This often includes how the development environment was established, how other important tools and technologies were integrated into the platform, and the critical decisions made during the development process of the platform.

2.2. Scope

The scope of development of the Melly Fashion is extensive as it seeks to address most of the core areas of an e-commerce firm. It is divided into several major sections each of which addresses essential features that contribute to the overall shopping process.

2.2.1. Authentication

Identity verification remains a core component of the platform where users have to prove they are legitimate or distinct before they have personal access.

Sign In: It is convenient for users to log in to their accounts using their account details while ensuring security. To sign-in the user it takes minimal time and uses standard encryption form to protect the user data.

Sign Up: New users can register a free account with the necessary profile information, including the email address, password, and other basic data. It is not difficult to sign up and the process is quite straightforward which contributes to the ease of onboarding.

2.2.2. Product Details

The product management system, therefore, plays a crucial role within Melly Fashion as it provides implement for display of the clothing items for sale.

Product Listing: All the products with the details such as product names, prices, descriptions, and images are categorized and shown on the platform. The listing is highly interactive and glamorous in such a way that the user can target any specific area they are looking for easily.

Product Specifications: Every product has its own page that contains the description of an item, availability and other pertinent specifications. A plus to these descriptions is that high-quality images inform the user about the product of interest.

2.2.3. Cart Functionality

The cart function is the key to the shopping experience since it enables the user to select products, review them before purchasing them. The cart features include:

Add to Cart: Cart management is possible from the product list page or the product detail page for the user. This action allows the user to preserve the products on the basket for next purchase so that it can easily shop again.

Edit Cart: Cart detail: Here, users have the option of editing the contents of the cart; this means that they can increase the quantity of the product, reduce the quantity of the product or even delete the product if they have changed their mind. This feature is designed to be flexible and can guarantee that the clients can modify their commands before paying for the items.

Delete Cart: Plainly, if the users no longer have the intention of proceeding with any items that they have chosen, they can delete everything in the cart for one click. The Feature is convenient for people who want to begin their shopping session from the very beginning.

2.2.4. Order Functionality

Carrying out the user's purchase orders is important in order and inventory management. This section includes:

Place Order: Once people are happy with the contents of their cart, they can go ahead to check out on an item. This function completes the purchase process and moves articles from a cart to an order in progress. Customers are required to review and input their shipping information as well as their payment option before checking out.

Order Details: Once an order has been placed users are able to view the specific details of the order made which includes, the items ordered, the cost of the items, the shipping details, and delivery dates. This makes users to be aware of there orders and also helps to develop trust between the company and the users.

Order History: The users have an option to view previous orders and check the details of orders that have been made in the past and the current status of the order that has been made. This feature can be valuable for clients who intend to order some products again or track the delivery of the recently purchased goods.

2.2.5. Payment Gateway

Another one is the payment gateway integration that has to be done in order to provide safe and accessible shopping. The platform supports multiple payment methods, including:The platform supports multiple payment methods, including:

PayPal: The option is provided to allow users to pay using their PayPal accounts, which is easy to use and very much secure for the users- hence safe for any transactions to be made.

Stripe: For secure credit and debit card processing, it uses integration with one of the fastest-growing payment platforms, the Stripe. Stripe is popular for being secure and easy to integrate for processing payments and customers love it for that.

Credit Card: Direct payment through credit cards without third-party payment gateways.

Test Pay Option: In the development stage, test payment option is there so the payment processes can be tested without actually coming into operation. This feature is useful when one has developed a new payment gateway for calculation and confirmation whether it will operate efficiently or not when the website goes live.

2.2.6. User Profile Section

Melly Fashion also has the User Profile section where one is able to edit her or his personal information and change settings. Key features of the user profile section include:Key features of the user profile section include:

Edit Name, Email, and Password:

Some of the modifications that can be easily made include the user’s name, his or her email address and password. This is important in keeping the current and updated contact information and ensuring their accounts’ security.

3. Motivation

3.1. Purpose and Motivation

Melly Fashion was developed because many small and medium-sized clothing vendors struggle to adopt e-commerce services.

The platform was developed with the intention to have a competitive advantage in a market which is currently inundated with large e-commerce giants.

This aimed at providing a smarter way of shopping for the end-user and technology solutions within the supply chain for the retailers.

3.2. Market Challenges and Target Audience

Most small and medium-sized merchants face challenges when it comes to designing and managing an online store.

They are typically not well equipped to challenge the dominance of the market-leading e-commerce platforms with regard to their user interfaces and technological underpinnings.

3.3. Technological Approach

Tech Stack:

The platform was developed using TypeScript, React, Nodejs, MongoDB, HTML, CSS, JavaScript, Bootstrap, and Vite.

Visual Appeal and Performance:

The use of these technologies makes the website's graphic and user interface appealing while at the same time making it fast to load and highly stable in performance.

3.4. Key Features

The platform must be able to accommodate increased numbers of products, users, and orders in a way that does not slow or hinder the business. Flexibility means that the platform is able to accommodate different loads and business requirements.

Continuity of the Shopping Experience: The sum of the values obtained in each step of the group of factors is then determined and compared to the maximum possible score for the entire group of factors from each client.

Created mainly to establish a user-friendly and functional environment for online shopping with fast navigation, brief response time, and safe payment methods.

All these conditions result into customer satisfaction hence customer repeat business and hence customer loyalty.

3.5. Team Motivation and Development

- The project forced the team to search and use the latest updates in tech stacks such as React, Node.
- Limitations were focused on enhancing the usability with the conception to become more users oriented and aligned with the changes of the current technological era and trends.

4. Problem and Objectives

4.1. Problem Statement

It should be noted that small and middle-performing clothing stores have considerable difficulties in making a serious appearance on the Internet. These are mainly due to inadequate technical skills, shortage of resources, and quick changes on the web Front. These businesses lack strong and good e-commerce platform they cannot cope with the top-end conventional large scale retailers who already have efficient online systems. Hence, they fail to capitalize on the ever-rising online customer market that has emerged as an important factor in retailing in the current world economy. Some of the main challenges the retailers face are to build websites that are not only practical but also visually attractive and easy to navigate. Given that some of these features include product display, shopping cart, user authentication, order processing, and secure payment, which are fundamental to an e-commerce site, means investment in human capital and technologies is a challenge that must be overcome. Further, even if a website is created, launching it on a viable and efficient host proves to be another challenge, which also affects their capacity to sustain a firm and polished web image.

4.2. Objectives

4.2.1. Develop a Responsive and Visually Appealing Website:

The goal is to design a website that is visually appealing and accommodates multipurpose connectivity with devices while not compromising user experience. The design has to be related to the brand so that customers would interact with it and decide to buy something.

4.2.2. Implement Essential E-Commerce Features:

As a result, in order to compete, the platform must contain all the necessary features of a modern e-commerce website.

- **Product Listing:** An online store, where users can come, and explore a list of products and services offered by the company Filtered/Searched.
- **Shopping Cart**: A working shopping cart where customers can select products and adjust the contents of their cart before making the purchase.
- **User Authentication:** Robust registration and login systems to safeguard the user information and enable targeted shopping experiences.
- **Order Management:** Easy-to-use platform where users can place orders, check their history, and track their orders' delivery.

4.2.3. Integrate a Secure Payment Gateway:

Security is a significant factor in e-commerce since the business processes involved require protection of customer information such as their credit card information. The goal is to incorporate a payment system that accepts multiple payment types and guarantees the security of all transactions to establish customers' trust.

4.2.4. Deploy the Website on a Reliable Hosting Platform:

The last goal is to make sure that the website is hosted on a platform that gives reliability, scalability and high performance. The selected hosting should be capable of supporting traffic rush, load time, and uptime that would afford the client and shoppers seamless shopping experience.

In this way, Melly Fashion will allow small and medium-sized retailers to have an efficient, competitive site to grow their business and sell more products online.

5. System Overview

Melly Fashion is an all-in-one full-fledged e-commerce web application that aims to provide an optimal solution for online shopping. It encompasses tools of the modern world to design and implement high-performing systems with scalable structures and satisfying customers, as well as fulfilling the needs of the business.

5.1. Frontend

The frontend of Melly Fashion is built using React with TypeScript incorporated for type safety so that bugs could be easily avoided and the code could be maintained with ease. The application has a user-friendly layout for navigation and it was developed with the help of Bootstrap that adapt the appearance of the platform to any devices, be it a computer, tablet, or a phone. This helps to have a fast creation of the quality of the user interface to meet the needs of the users with the help of the technologies mentioned above.

5.2. Backend

The backend is Node.js and Express.js, which offers a flexible framework for dealing with the server-side processes. It processes the API, handles the user login and does the necessary interaction with the database. Unlike the frontend, the backend has essential roles in the application, including querying the product catalog, handling the user actions that include adding products to the cart, and order handling. Node.js and Express.js makes sure that loads of traffic can be comfortably handled on the backend making the platform both fast and reliable.

5.3. Database

For the purpose of storing data on products, customers, and orders, Melly Fashion employs a database called MongoDB. MongoDB has no problem with the schema since the flexibility of the NoSQL databases is advantageous in the case of an e-commerce platform that experiences different data forms and structures. The database shows high efficiency in data collection, and this helps the application to provide optimal service to users as imposed by the online shopping activities. Moreover, the scalability of MongoDB as a database means that as a business changes and develops, the database does not have to be reworked to keep up with the demands.

5.4. Deployment

The application uses Heroku for backend hosting and Netlify for frontend hosting. The backend URI is linked to the frontend before the frontend is deployed to Netlify. This setup helps provide a perfect interaction between the client-side and server-side components thus complementing a perfect user experience.

In terms of the backend, Heroku offers a stable and scalable environment so that the Melly Fashion platform could maintain its availability and performance at different levels of user engagement. It enables the effortless installation and administration of the backend while having the ability to increase resources if required.

Here, the frontend is hosted on Netlify that offers CD among other features for hosting and delivery to ensure a fast and efficient load time. Capterra in particular highlights that the support that comes with Netlify allows for frequent and easy update of new changes on the Melly Fashion platform.

You can access Melly Fashion through - https://mellyfashion.netlify.app

5.5. GUI Overview

Also, the user interface of the Melly Fashion is friendly which is known as Graphical User Interface (GUI) that makes it easier for the users to find their way through the website.

- **Home Page**: From the homepage, users can select products of their choice which is heared filtered and sorted considering some specified factors. This page looks more eye-popping so the user would not have a hard time looking for the product or concepts to be found in this site.

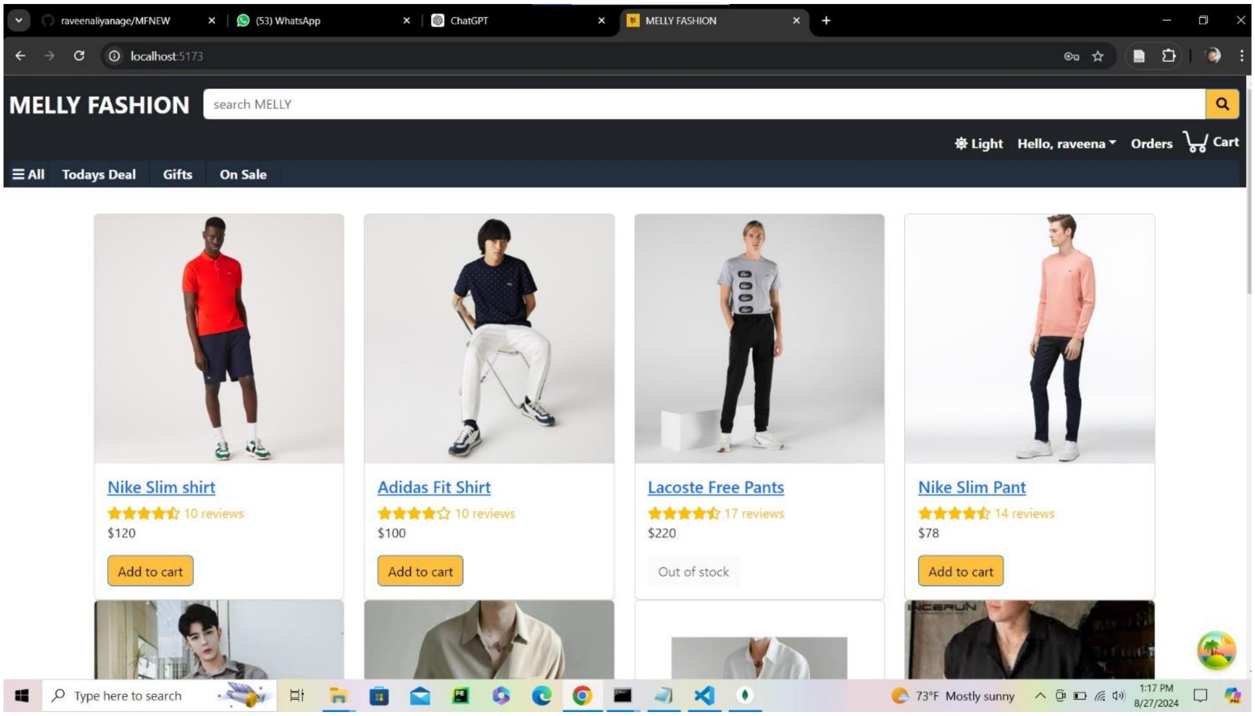


Figure 1 - Home Page GUI

- **Product Page**: Every product is described in the special page and contains the descriptions, the rating, and the reviews from the buyers. What makes this page effective is the fact that all the information put forward is aimed at assisting the users in making wise purchase choices.

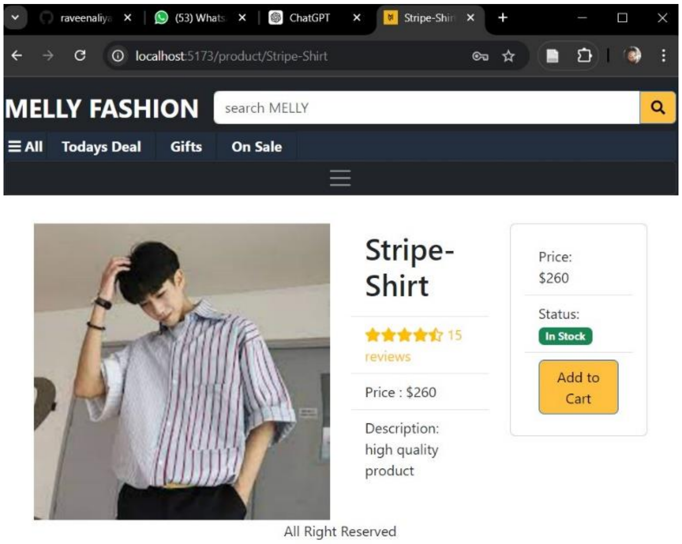


Figure 2 - Product Page GUI

- **Cart Page**: The cart page helps the users to see and also control the items they wish to buy. It consist of features for changing quantity, deleting items, and Check out. The cart page is created to be simple and streamlined; helping to make the buying process as easy as possible.

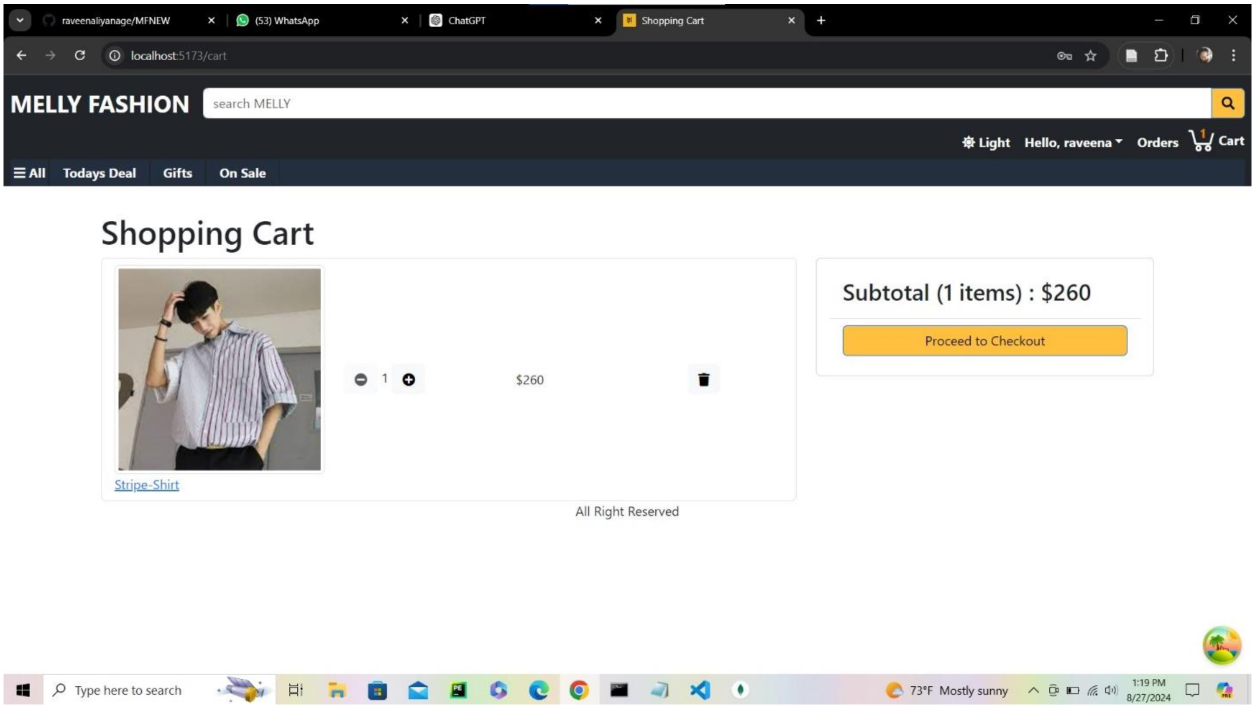


Figure 3 - Cart Page GUI

- **Signin/Signup Pages**: These pages deal with the process of user login that allows the users to register or sign in to their account. The authentication is secure and safe hence protecting the user data and is developed to give an efficient and effective login process.

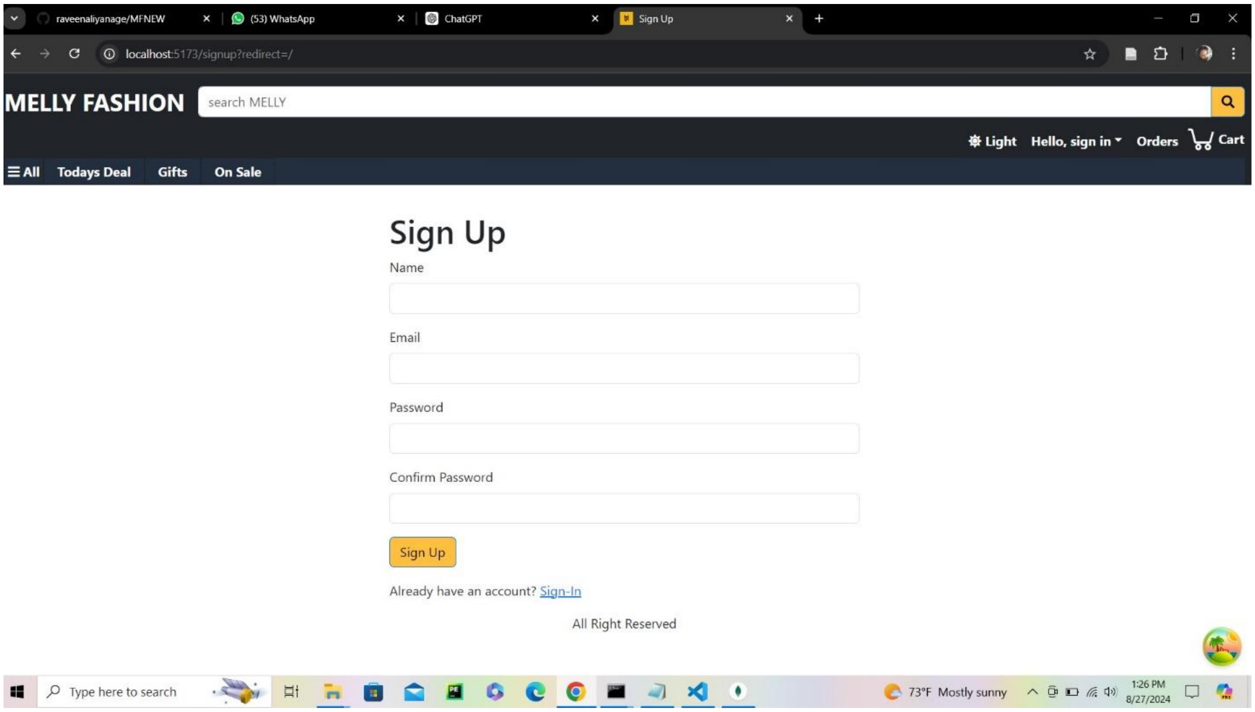
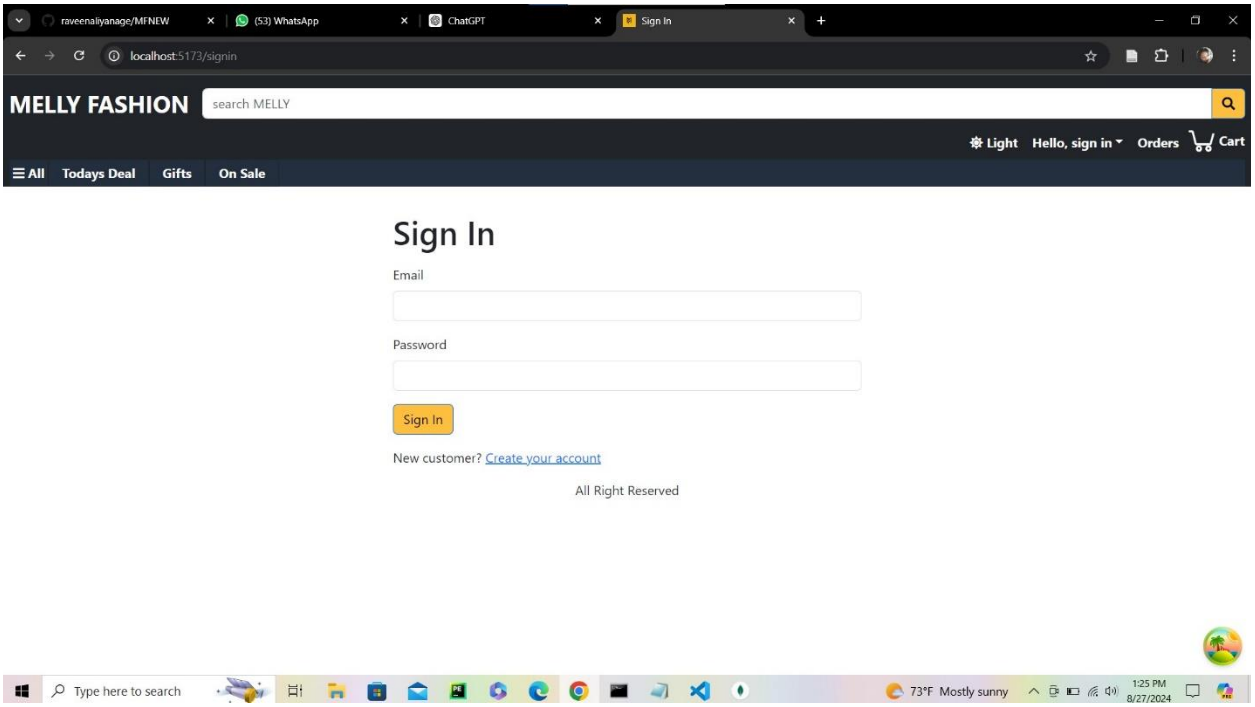


Figure 4 - Sign Up Page GUI



Sign In

Email

Password

Sign In

New customer? Create your account

All Right Reserved

Figure 5 - Sign In Page

- **Shipping Page**: This page contains the details about product shipping like customer full name, address, city, postal code, country etc.

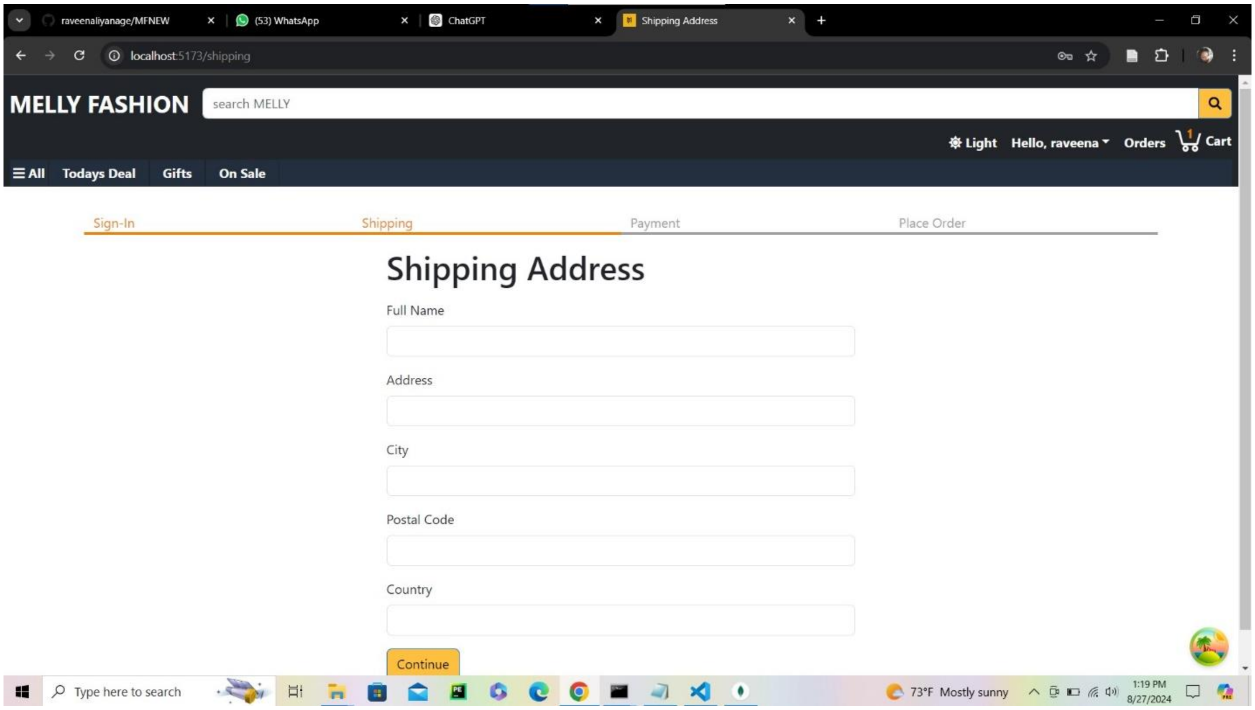
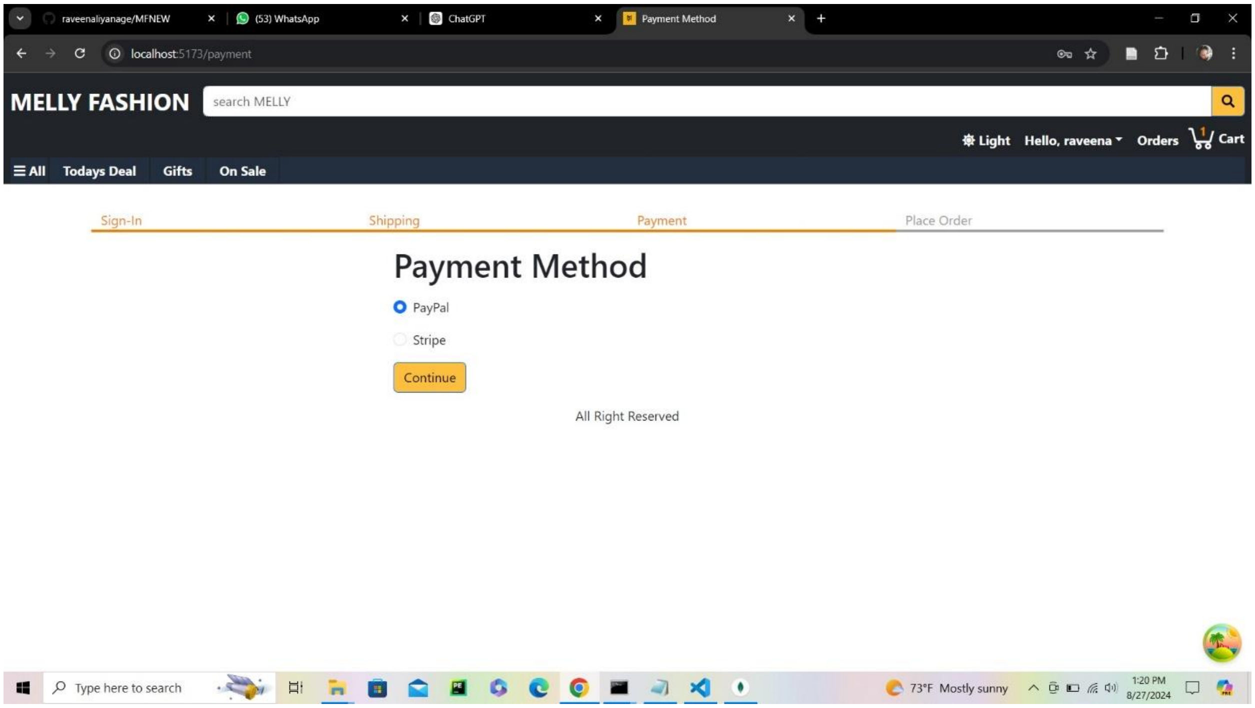


Figure 6 - Shipping Page GUI

- **Payment Pages**: This payment pages contain the Payment Gateway and place payment pages. In payment gateway user can choose PayPal or Stripe according to their choice. In place payment they can do their payment using a Paypal account or credit card. There is a Test pay option create to enhance development mode.



Payment Method

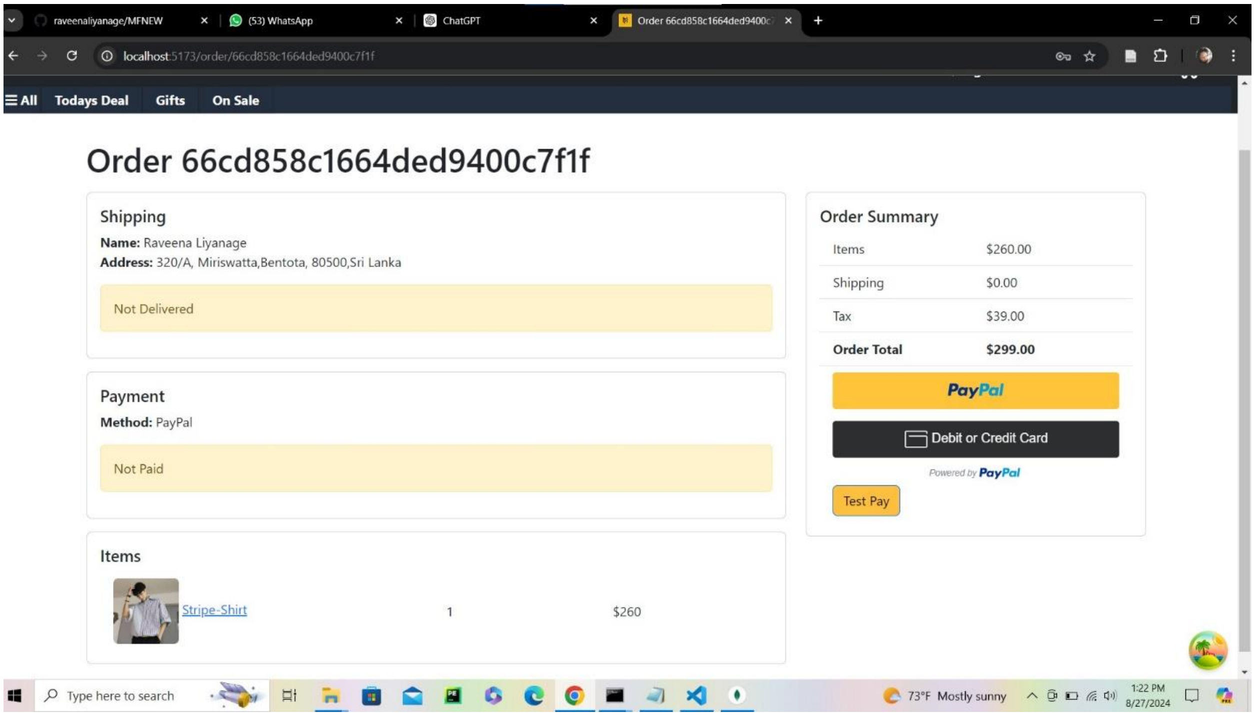
● PayPal

○ Stripe

Continue

All Right Reserved

Figure 7 - Payment Gateway GUI



Order 66cd858c1664ded9400c7f1f

Shipping

Name: Raveena Liyanage
Address: 320/A, Miriswatta,Bentota, 80500,Sri Lanka

Not Delivered

Payment

Method: PayPal

Not Paid

Items

Stripe-Shirt	1	$260

Order Summary

Items	$260.00
Shipping	$0.00
Tax	$39.00
Order Total	**$299.00**

PayPal

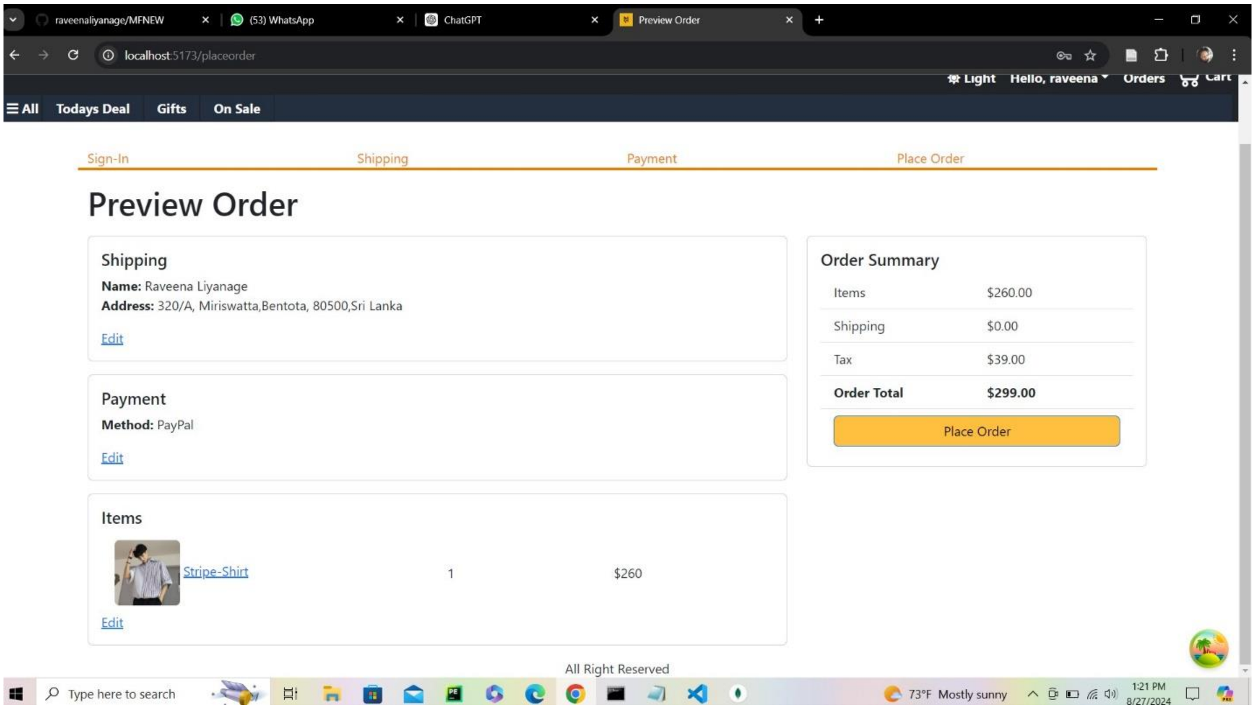
Debit or Credit Card

Powered by PayPal

Test Pay

Figure 8 - Payment Page GUI

- **Order Pages**: They also offer an opportunity for users to track down their past orders, methods of payment and shipment information. A number of them are aimed at aiding users to manage their orders in the simplest way possible to enable tracking and managing of orders placed. In place order page if user place a order they direct to payment page.



Preview Order

Shipping

Name: Raveena Liyanage
Address: 320/A, Miriswatta,Bentota, 80500,Sri Lanka

Edit

Payment

Method: PayPal

Edit

Items

Stripe-Shirt	1	$260

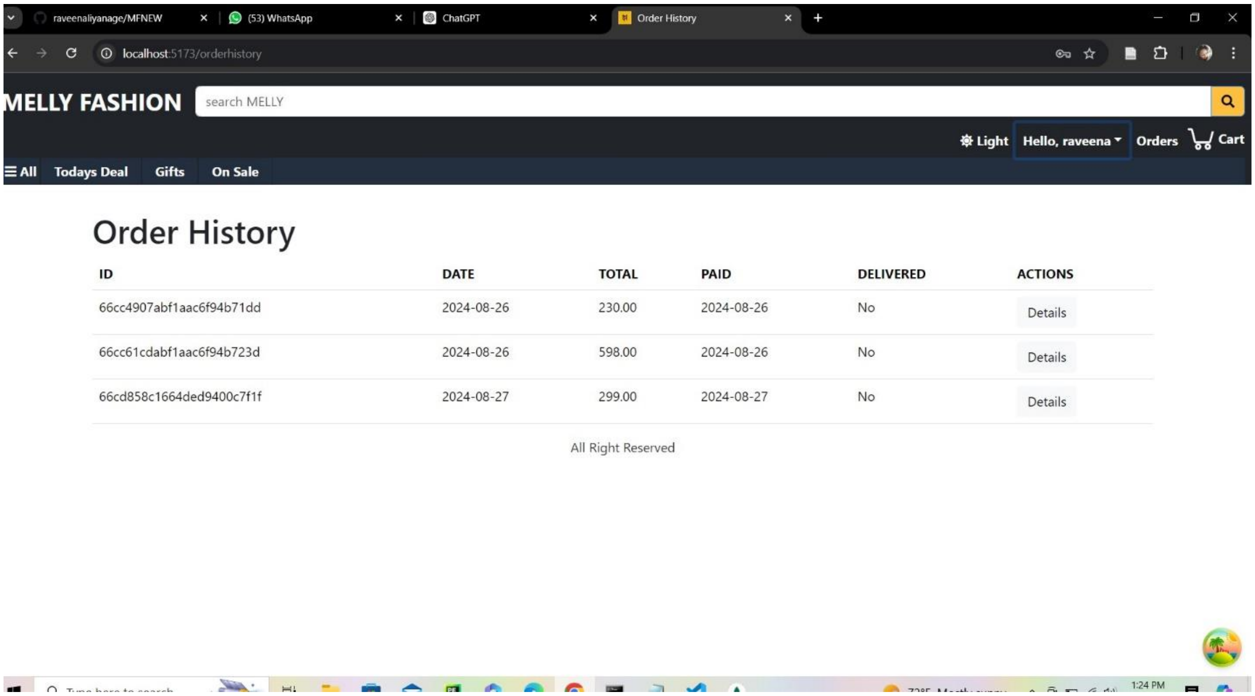
Edit

Order Summary

Items	$260.00
Shipping	$0.00
Tax	$39.00
Order Total	**$299.00**

All Right Reserved

Figure 9 - Place Order Page GUI



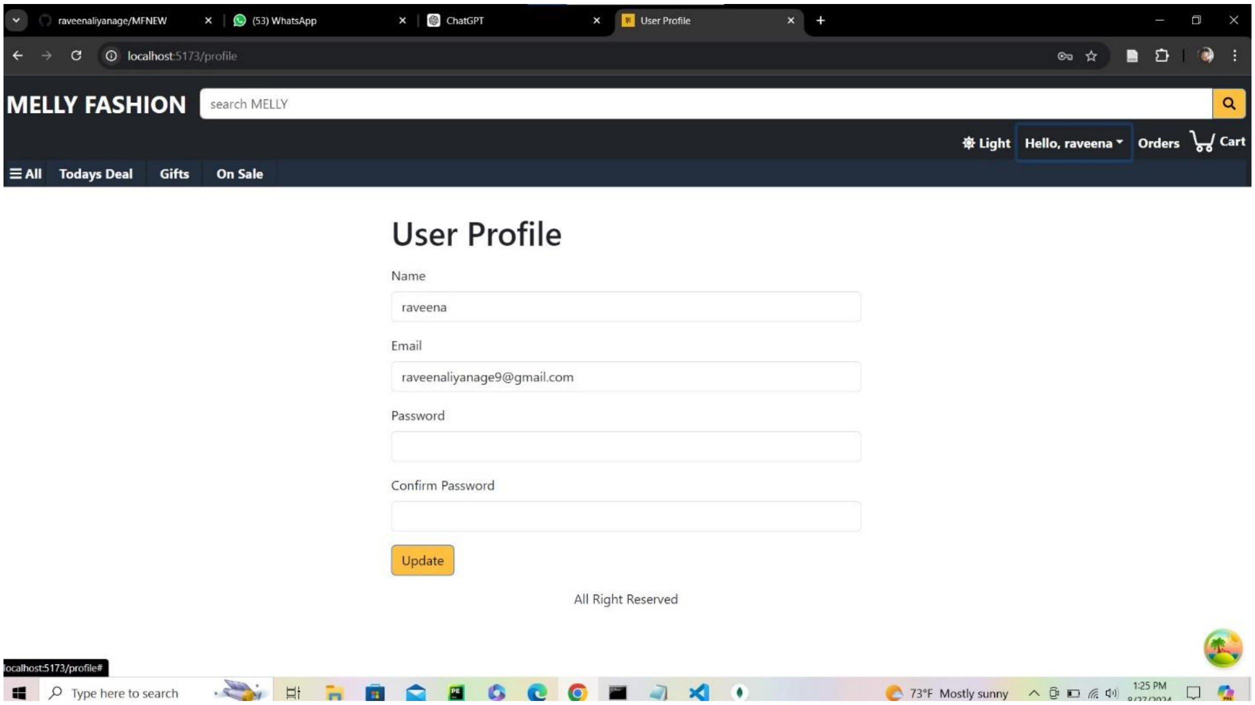
Order History

ID	DATE	TOTAL	PAID	DELIVERED	ACTIONS
66cc4907abf1aac6f94b71dd	2024-08-26	230.00	2024-08-26	No	Details
66cc61cdabf1aac6f94b723d	2024-08-26	598.00	2024-08-26	No	Details
66cd858c1664ded9400c7f1f	2024-08-27	299.00	2024-08-27	No	Details

All Right Reserved

Figure 10 - Order History Page GUI

- **User Profile Page**: The user profile page is the special page whereby the user is given an option to input, change or remove the details like contacts, addresses, and payment information. It also permits users to manage their accounts with the possibility to change password, email, and other parameters. For instance, on this very page, the user is offered options for managing one’s account data and configure one’s preferences in the platform for a more bespoke shopping experience.



User Profile

Name

raveena

Email

raveenaliyanage9@gmail.com

Password

Confirm Password

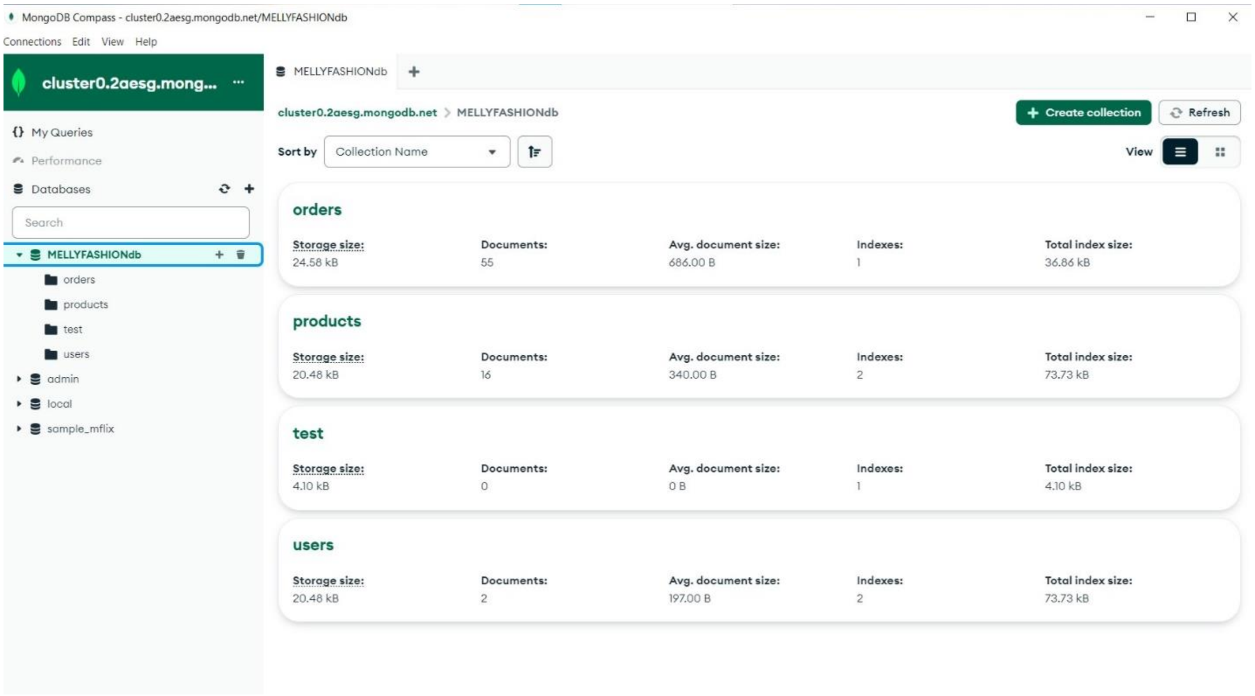
All Right Reserved

Figure 11 - User Profile Page GUI

Melly Fashion’s system architecture and GUI are carefully crafted to create a powerful, user-friendly platform that meets the needs of modern e-commerce, ensuring a smooth and enjoyable shopping experience for all users.

6. Database Structure

MongoDB Atlas is used to structure Melly Fashion’s database and the database is managed using MongoDB Compass. To facilitate the easy storage of data as well as the easy retrieval of the data, the database is made up of multiple collections that contain all unique types of data that are relevant in the management of the e-commerce platform. Each collection is created to deal with certain parts of the application: data accuracy, extensibility and data access considerations.



orders
Storage size: 24.58 kB
Documents: 55
Avg. document size: 686.00 B
Indexes: 1
Total index size: 36.86 kB

products
Storage size: 20.48 kB
Documents: 16
Avg. document size: 340.00 B
Indexes: 2
Total index size: 73.73 kB

test
Storage size: 4.10 kB
Documents: 0
Avg. document size: 0 B
Indexes: 1
Total index size: 4.10 kB

users
Storage size: 20.48 kB
Documents: 2
Avg. document size: 197.00 B
Indexes: 2
Total index size: 73.73 kB

Figure 12 - MongoDB Compass

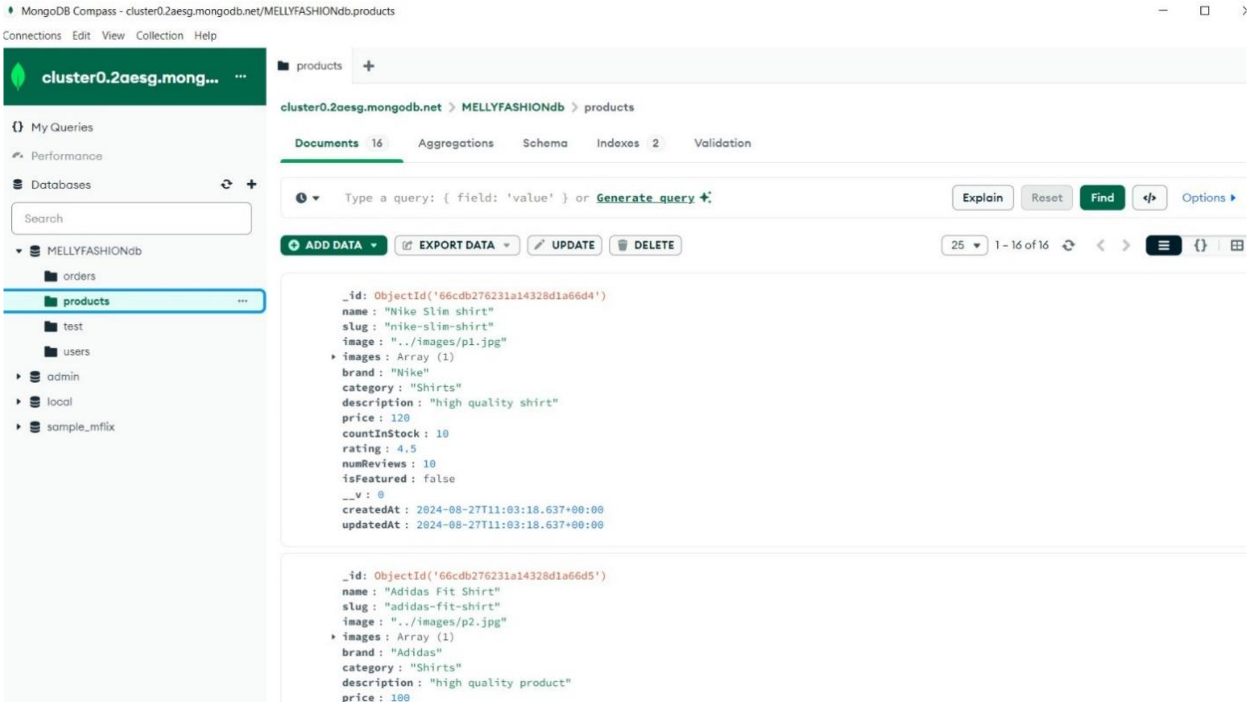
6.1. Collections

6.1.1. Users:

This collection presents all the user information such as login information, account details and even preferences. It includes all identifiable information, including username, the user’s email address, password in encoded form, and any other characteristic that may be useful for the identification of the user. The Users collection also handles token authentication for developing secure users’ sessions and it also assists the system to apply user profiling to enhance the shopping experience. Further, these suggestions can refer to a user’s order history, preferences, and even addresses.

6.1.2. Products:

The Products collection is the core of the e-commerce platform that keeps the comprehensive data about all the products presented in the Melly Fashion store. These are product name, description, price, quantity in stock and category among others. It also stores items such as images which are important when-displaying- products on the Internet. Every product document can contain the comments from the customers, which will help to understand how satisfied the clients are and which products are popular with them. It is intended to allow for intricate filtering and searching in the catalog so as to help users identify products of their desired type.



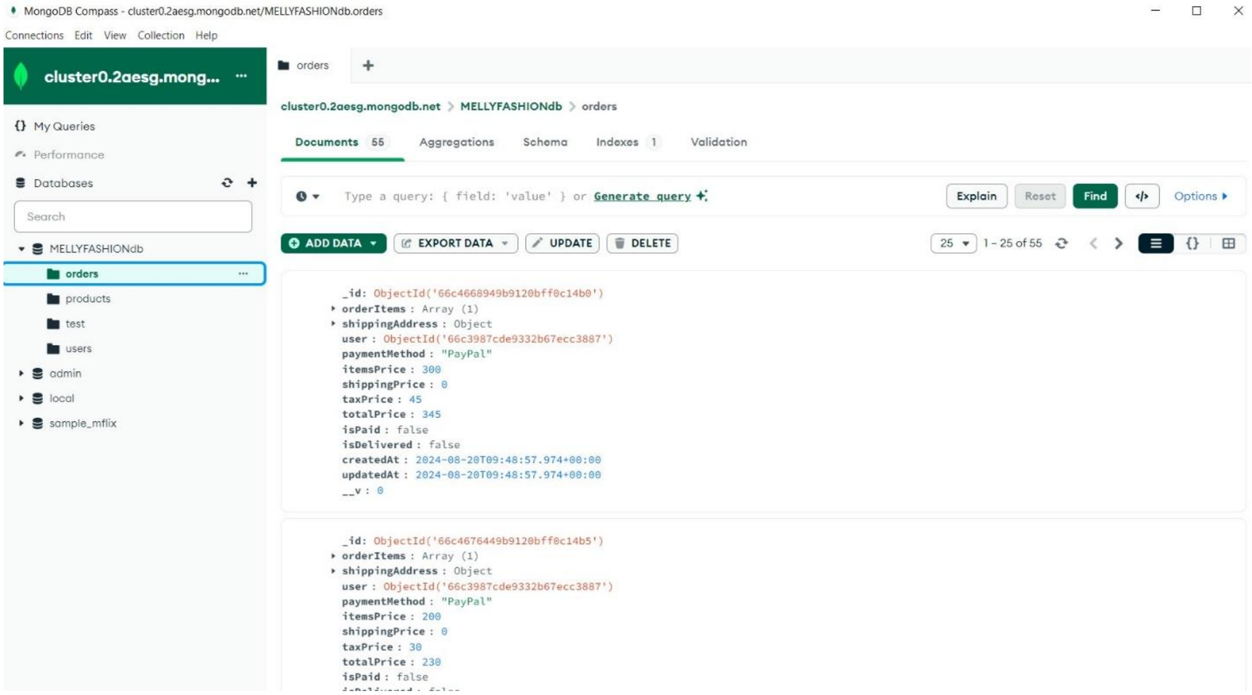
```
_id: ObjectId('66cdb276231a14328d1a66d4')
name : "Nike Slim shirt"
slug : "nike-slim-shirt"
image : "../images/p1.jpg"
▸ images : Array (1)
brand : "Nike"
category : "Shirts"
description : "high quality shirt"
price : 120
countInStock : 10
rating : 4.5
numReviews : 10
isFeatured : false
__v : 0
createdAt : 2024-08-27T11:03:18.637+00:00
updatedAt : 2024-08-27T11:03:18.637+00:00
```

```
_id: ObjectId('66cdb276231a14328d1a66d5')
name : "Adidas Fit Shirt"
slug : "adidas-fit-shirt"
image : "../images/p2.jpg"
▸ images : Array (1)
brand : "Adidas"
category : "Shirts"
description : "high quality product"
price : 100
```

Figure 13 - MongoDB Products

6.1.3. Orders:

The Orders collection encompasses all aspects of the customer's order ranging from the point of order placement to the delivery of the order. Also, it has records about all the orders: the items bought, the quantities, prices, status of payments, and the shipping details. It also registers different steps in an order, which may include processing, shipment, and delivery, to create awareness to the users regarding their obtained products. Ordersaverse is one of the collections that are significant for the management of orders, due to which it can receive and process orders from customers efficiently and at the right time.



```
  _id: ObjectId('66c4668949b9120bff0c14b0')
▸ orderItems : Array (1)
▸ shippingAddress : Object
  user : ObjectId('66c3987cde9332b67ecc3887')
  paymentMethod : "PayPal"
  itemsPrice : 300
  shippingPrice : 0
  taxPrice : 45
  totalPrice : 345
  isPaid : false
  isDelivered : false
  createdAt : 2024-08-20T09:48:57.974+00:00
  updatedAt : 2024-08-20T09:48:57.974+00:00
  __v : 0

  _id: ObjectId('66c4676449b9120bff0c14b5')
▸ orderItems : Array (1)
▸ shippingAddress : Object
  user : ObjectId('66c3987cde9332b67ecc3887')
  paymentMethod : "PayPal"
  itemsPrice : 200
  shippingPrice : 0
  taxPrice : 30
  totalPrice : 230
  isPaid : false
  isDelivered : false
```

Figure 14 - MongoDB Orders

6.2. Database Management

6.2.1. MongoDB Atlas:

MongoDB Atlas is a fully managed cloud database which hosts the database. Atlas facilitates automated scaling, backups and some security attributes so that the database is ready for handling high traffic and large amount of data with high reliability. But it also features good monitoring capabilities which allow a developer track how a database performs and fine-tune queries if necessary.

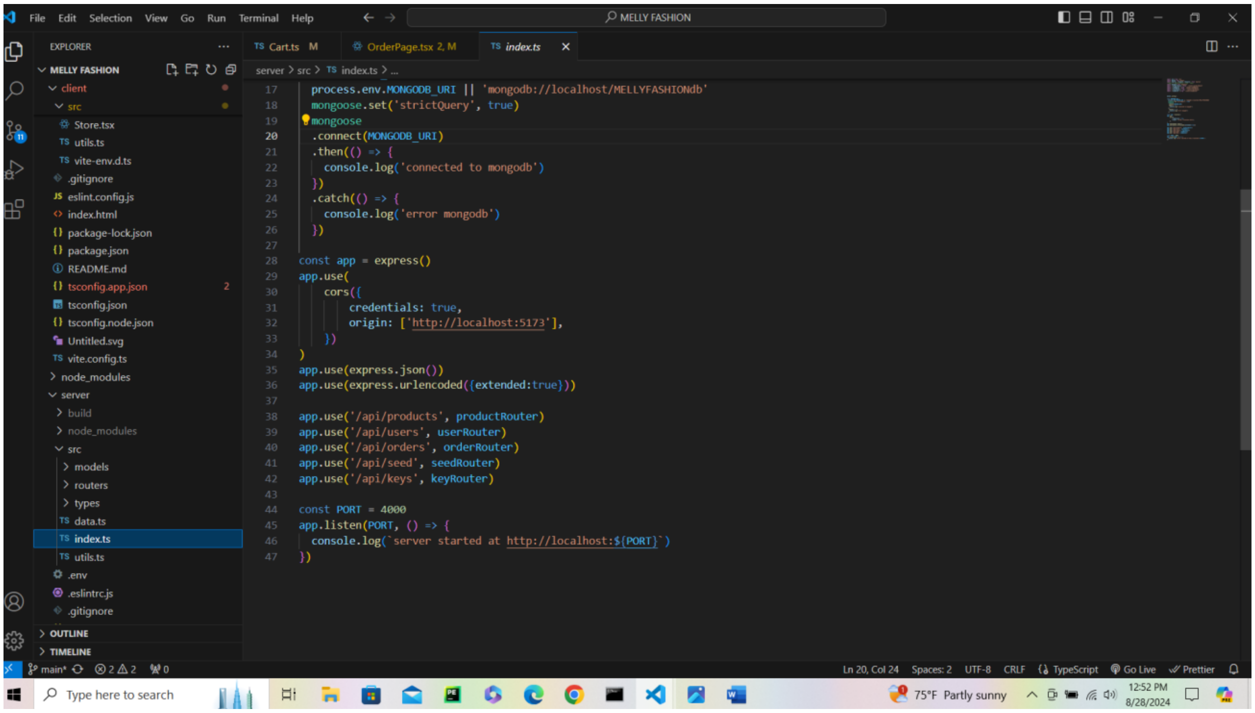
6.2.2. MongoDB Compass:

As for the tools which interact with the MongoDB, MongoDB Compass is the tool that provides convenient ways of management and interaction with the database. It helps in exploring data, querying, and visualizing the schema thus making it easier for the developers at Melly Fashion to deal with the complex data structures within the Melly Fashion platform. Compass also gives related information in data distribution and indexing, ensure that the application running in a good way as the scale grows up.

In general, the used database structure allows to satisfy the development requirements of the Melly Fashion as dynamically and as scalable as possible, addressing all the sides of the e-commerce process and the needs of the system’s growth in the future.

7. API Implementation

Several API routes are exist in Melly Fashion's backend because it possesses a variety of functionalities that require control. Each route is processed by a particular router that address definite types of requests; thus, the system remains modular, maintainable and scalalbe.



File Edit Selection View Go Run Terminal Help
MELLY FASHION
EXPLORER
TS Cart.ts M
OrderPage.tsx 2, M
TS index.ts
server > src > TS index.ts > ...

```
  process.env.MONGODB_URI || 'mongodb://localhost/MELLYFASHIONdb'
  mongoose.set('strictQuery', true)
  mongoose
  .connect(MONGODB_URI)
  .then(() => {
    console.log('connected to mongodb')
  })
  .catch(() => {
    console.log('error mongodb')
  })

const app = express()
app.use(
    cors({
        credentials: true,
        origin: ['http://localhost:5173'],
    })
)
app.use(express.json())
app.use(express.urlencoded({extended:true}))

app.use('/api/products', productRouter)
app.use('/api/users', userRouter)
app.use('/api/orders', orderRouter)
app.use('/api/seed', seedRouter)
app.use('/api/keys', keyRouter)

const PORT = 4000
app.listen(PORT, () => {
  console.log(`server started at http://localhost:${PORT}`)
})
```

Code Snippet 1 – API

7.1. /api/products (Product API)

Purpose:

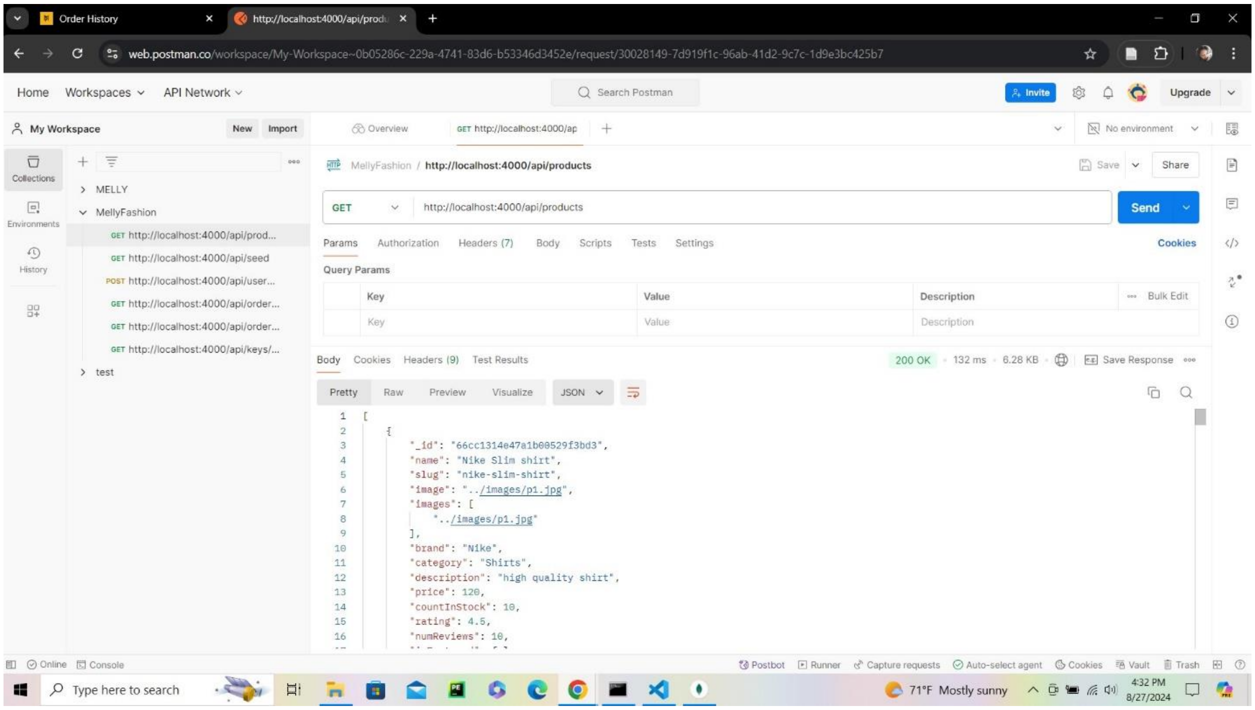
Supervises all activities that are in one way or another linked to products.

Endpoints:

GET /: I notice one that gets the list of all products. This is used to show products when customer gets to the homepage and when selecting certain items on the site.

Example Usage:

This API allows users to see what is currently available, browse for certain product and get the detailed description of the product.



GET http://localhost:4000/api/products

```
[
    {
        "_id": "66cc1314e47a1b00529f3bd3",
        "name": "Nike Slim shirt",
        "slug": "nike-slim-shirt",
        "image": "../images/p1.jpg",
        "images": [
            "../images/p1.jpg"
        ],
        "brand": "Nike",
        "category": "Shirts",
        "description": "high quality shirt",
        "price": 120,
        "countInStock": 10,
        "rating": 4.5,
        "numReviews": 10,
```

Figure 15 - API Products

7.2. /api/users (User API)

Purpose:

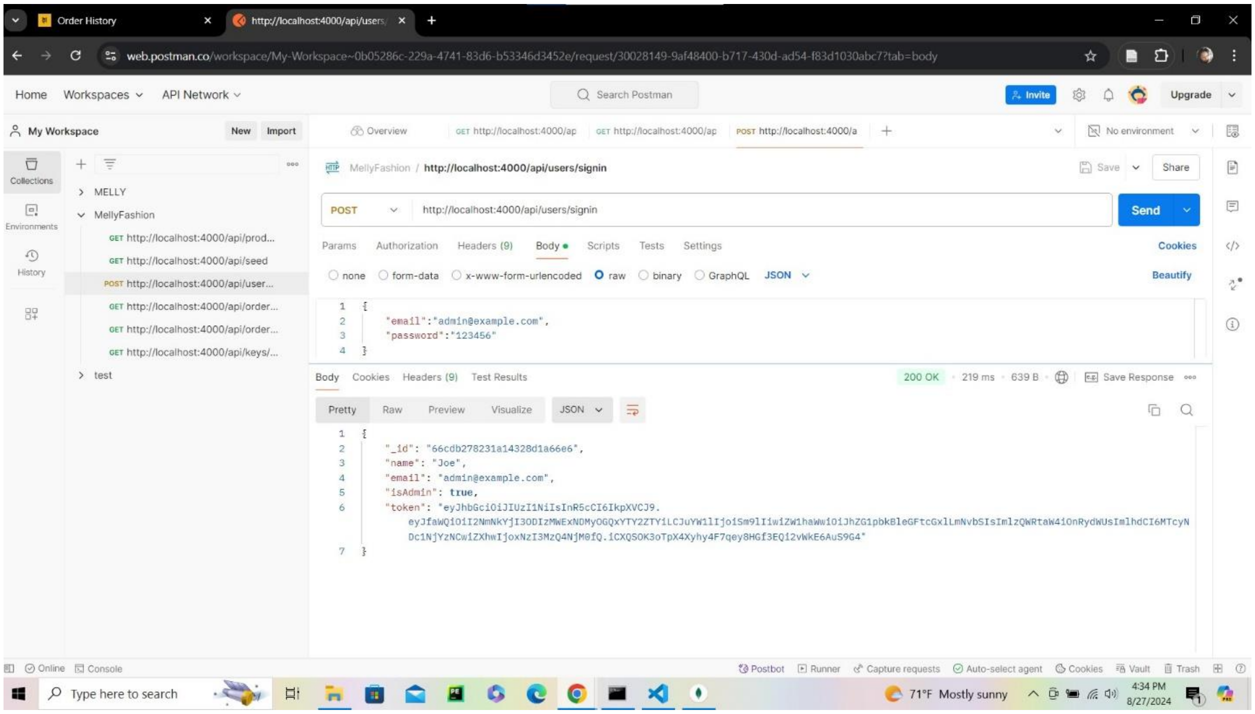
Is responsible with the sign in/up of users and also the management of the user's profile.

Endpoints:

POST /signin: The login function is used for the user's Authentication and also issues the token required for the session management.

Example Usage:

This API must allow the users to login, register, update their profiles and remain logged during their session.



MellyFashion / http://localhost:4000/api/users/signin

POST http://localhost:4000/api/users/signin

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "email":"admin@example.com",
    "password":"123456"
}
```

Body Cookies Headers (9) Test Results 200 OK 219 ms 639 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "_id": "66cdb278231a14328d1a66e6",
    "name": "Joe",
    "email": "admin@example.com",
    "isAdmin": true,
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJfaWQiOiI2NmNkYjI3ODIzMWExNDMyOGQxYTY2ZTYiLCJuYW1lIjoiSm9lIiwiZW1haWwiOiJhZG1pbkBleGFtcGxlLmNvbSIsImlzQWRtaW4iOnRydWUsImlhdCI6MTcyN
        Dc1NjYzNCwiZXhwIjoxNzI3MzQ4NjM0fQ.1CXQSOK3oTpX4Xyhy4F7qey8HGf3EQ12vWkE6AuS9G4"
}
```

Figure 16 - API Users

7.3. /api/orders (Order API)

Purpose:

Oversees the receipt and processing of customers' orders and payments.

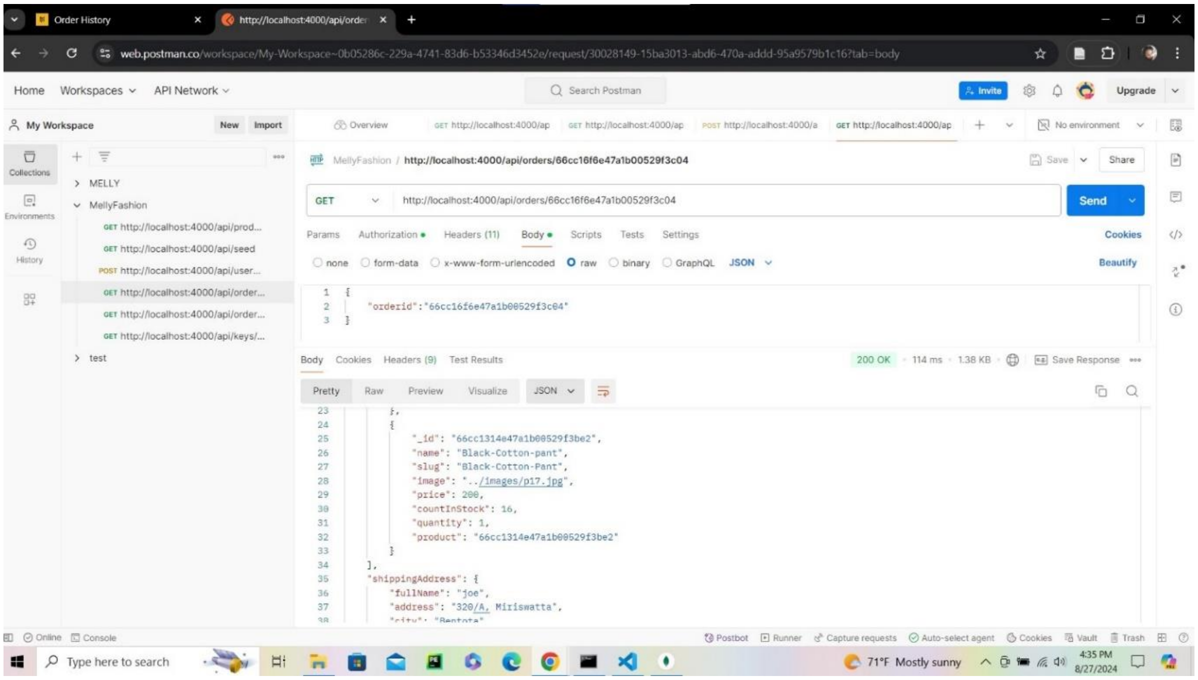
Endpoints:

GET /:id: This returns information of an order given the order ID, The current status of the order and confirmation of payment among others.

GET /mine: This operation returns basic information of all orders created by the user who is currently logged in the system.

Example Usage:

It also enables customers to place orders, track and manage their orders so that they can be in a position to purchase them in an efficient manner.



MellyFashion / http://localhost:4000/api/orders/66cc16f6e47a1b00529f3c04

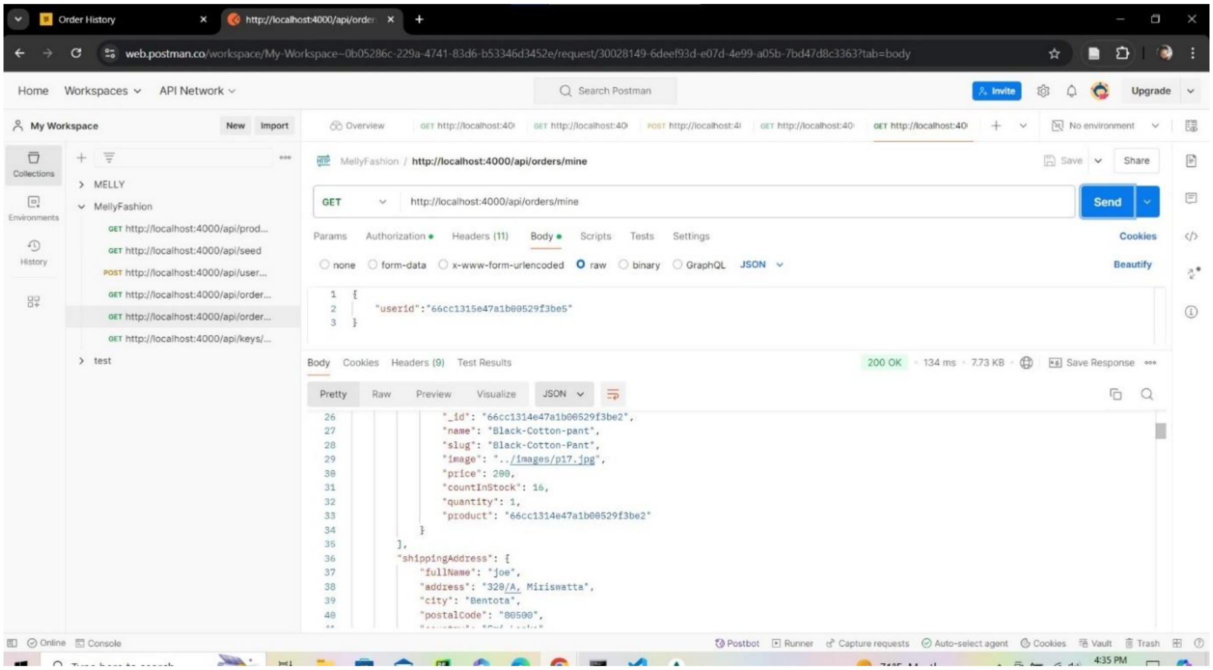
GET http://localhost:4000/api/orders/66cc16f6e47a1b00529f3c04

```
{
    "orderid":"66cc16f6e47a1b00529f3c04"
}
```

200 OK · 114 ms · 1.38 KB

```
            },
            {
                "_id": "66cc1314e47a1b00529f3be2",
                "name": "Black-Cotton-pant",
                "slug": "Black-Cotton-Pant",
                "image": "../images/p17.jpg",
                "price": 200,
                "countInStock": 16,
                "quantity": 1,
                "product": "66cc1314e47a1b00529f3be2"
            }
        ],
        "shippingAddress": {
            "fullName": "joe",
            "address": "320/A, Miriswatta",
            "city": "Bentota"
```

Figure 17 - API OrderID


Order History
web.postman.co/workspace/My-Workspace~0b05286c-229a-4741-83d6-b53346d3452e/request/30028149-6deef93d-e07d-4e99-a05b-7bd47d8c3363?tab=body
Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
History
MELLY
MellyFashion
GET http://localhost:4000/api/prod...
GET http://localhost:4000/api/seed
POST http://localhost:4000/api/user...
GET http://localhost:4000/api/order...
GET http://localhost:4000/api/order...
GET http://localhost:4000/api/keys/...
test
Overview
No environment
Save
Share
Send
Params
Authorization
Headers (11)
Body
Scripts
Tests
Settings
Cookies
Beautify
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON
Body
Cookies
Headers (9)
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Online
Console
Postbot
Runner
Capture requests
Auto-select agent
Cookies
Vault
Trash
4:35 PM

MellyFashion / http://localhost:4000/api/orders/mine

GET http://localhost:4000/api/orders/mine

```
{
    "userid":"66cc1315e47a1b00529f3be5"
}
```

200 OK · 134 ms · 7.73 KB

```
                "_id": "66cc1314e47a1b00529f3be2",
                "name": "Black-Cotton-pant",
                "slug": "Black-Cotton-Pant",
                "image": "../images/p17.jpg",
                "price": 200,
                "countInStock": 16,
                "quantity": 1,
                "product": "66cc1314e47a1b00529f3be2"
            }
        ],
        "shippingAddress": {
            "fullName": "joe",
            "address": "320/A, Miriswatta",
            "city": "Bentota",
            "postalCode": "80500",
```

Figure 18 - API Oeder-Mine

7.4. /api/seed (Seed API)

Purpose:

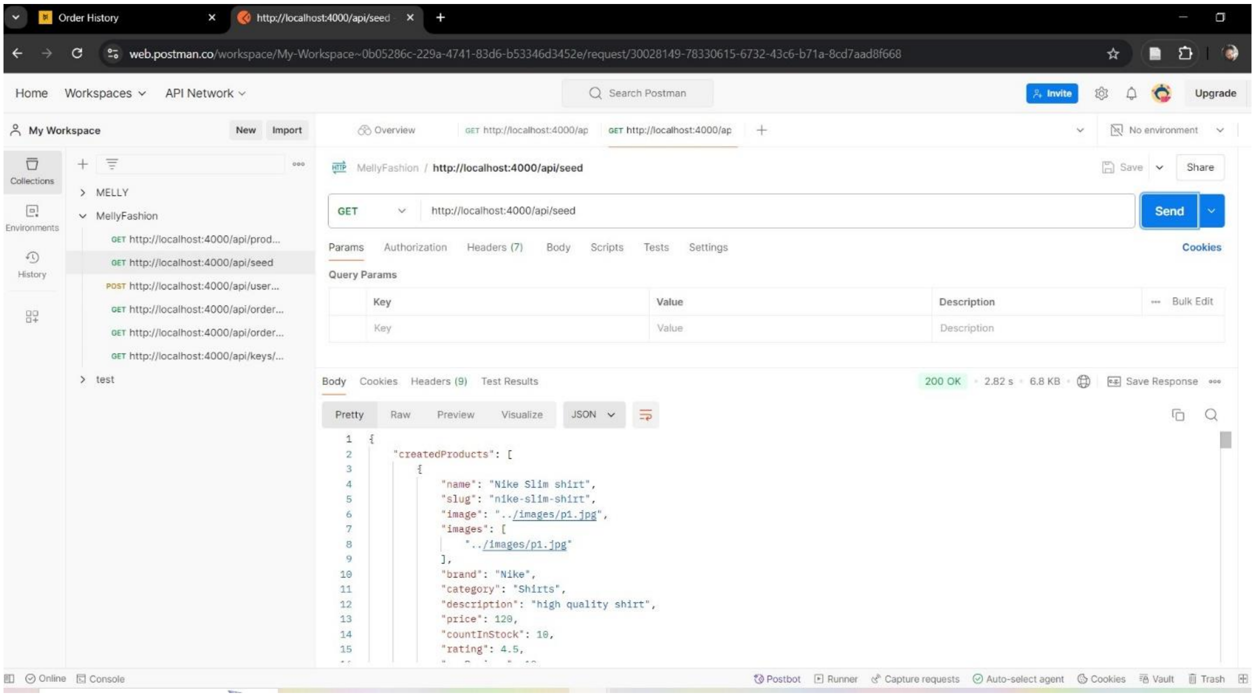
Creates a base of the database with a number of sample products and users, for instance.

Endpoints:

GET /: Initialises the database with development and test data in order to facilitate its testing. This is normally done during the construction phase and helps one to work on a dummy data to create a functional database.

Example Usage:

The seed API plays the role of assisting the developers easily feed the database with initial data, and thus reduce the raw effort of inputting data.



MellyFashion / http://localhost:4000/api/seed

GET http://localhost:4000/api/seed

200 OK · 2.82 s · 6.8 KB

```
{
    "createdProducts": [
        {
            "name": "Nike Slim shirt",
            "slug": "nike-slim-shirt",
            "image": "../images/p1.jpg",
            "images": [
                "../images/p1.jpg"
            ],
            "brand": "Nike",
            "category": "Shirts",
            "description": "high quality shirt",
            "price": 120,
            "countInStock": 10,
            "rating": 4.5,
```

Order History
http://localhost:4000/api/seed
web.postman.co/workspace/My-Workspace~0b05286c-229a-4741-83d6-b53346d3452e/request/30028149-78330615-6732-43c6-b71a-8cd7aad8f668
Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
Overview
GET http://localhost:4000/ap
GET http://localhost:4000/ap
No environment
Collections
Environments
History
MELLY
MellyFashion
GET http://localhost:4000/api/prod...
GET http://localhost:4000/api/seed
POST http://localhost:4000/api/user...
GET http://localhost:4000/api/order...
GET http://localhost:4000/api/order...
GET http://localhost:4000/api/keys/...
test
Save
Share
Send
Params
Authorization
Headers (7)
Body
Scripts
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (9)
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Online
Console
Postbot
Runner
Capture requests
Auto-select agent
Cookies
Vault
Trash

Figure 19 - API Seed

7.5. /api/keys (Key API)

Purpose:

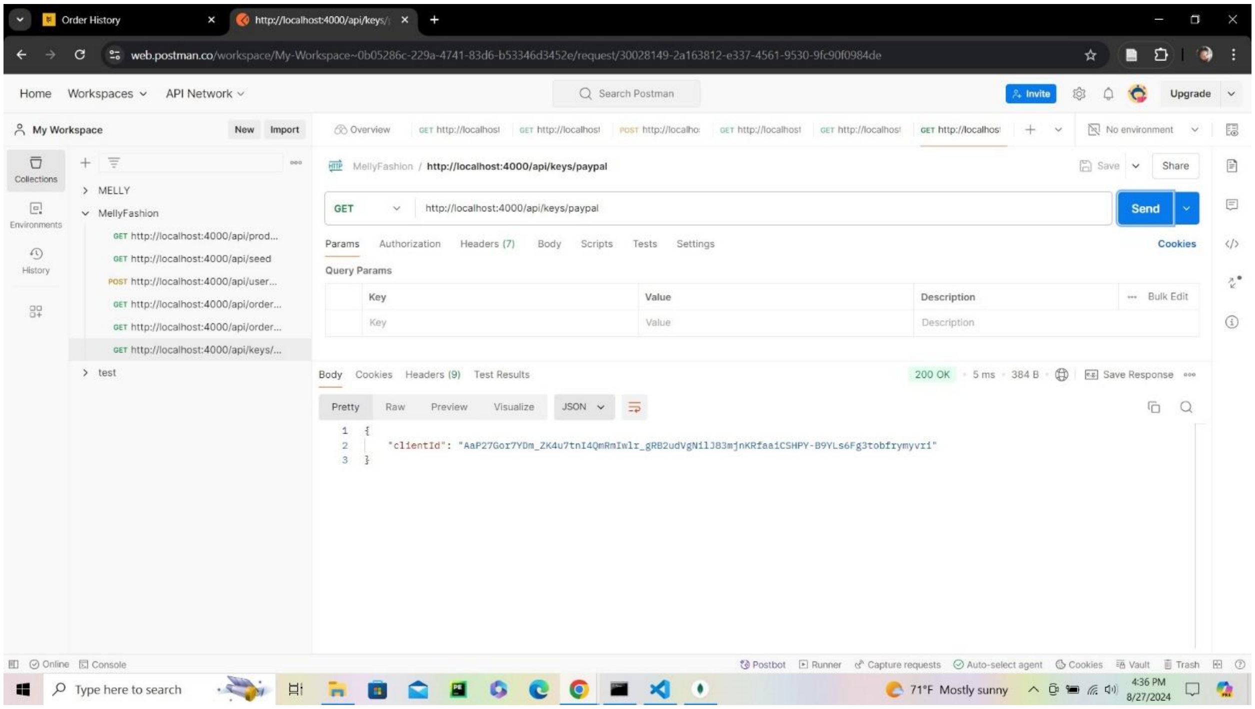
Responsible for the handling of API keys more so the keys that belong to third parties such as payment gateways.

Endpoints:

GET /paypal: Fetches the PayPal API key that PayPal uses for processing the transactions. This endpoint is required in order to implement PayPal payment gateway feature.

Example Usage:

This API makes it easy for the frontend to have all necessary keys in order to handle third party services such as PayPal to ensure payment solutions are adequately processed.



MellyFashion / http://localhost:4000/api/keys/paypal

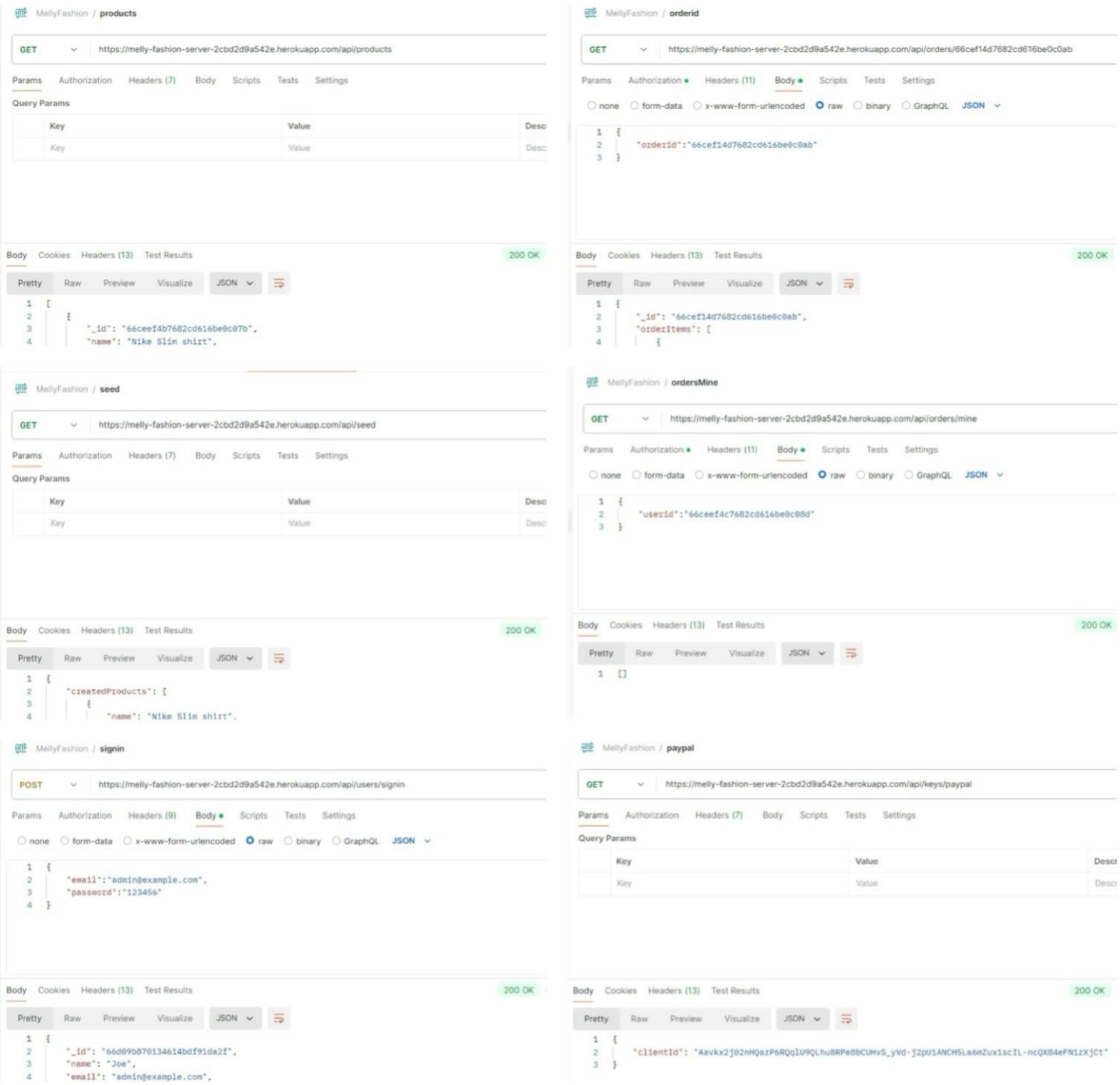
GET http://localhost:4000/api/keys/paypal

200 OK · 5 ms · 384 B

```
{
    "clientId": "AaP27Gor7YDm_ZK4u7tnI4QmRmIwlr_gRB2udVgNilJ83mjnKRfaaiCSHPY-B9YLs6Fg3tobfrymyvri"
}
```

Figure 20 - API-Key-Paypal

All of these APIs are unique in that they are designed to manage certain aspects of the e-commerce platform as whole in a unified and fully functioning Front End Shopping Experience. The APIs here are modular and this makes it easy to maintain and integrate new features to the application when they are added or modify the current ones.



MellyFashion / products

```
GET https://melly-fashion-server-2cbd2d9a542e.herokuapp.com/api/products
```

200 OK

```
[
    {
        "_id": "66ceef4b7682cd616be0c07b",
        "name": "Nike Slim shirt",
```

MellyFashion / orderid

```
GET https://melly-fashion-server-2cbd2d9a542e.herokuapp.com/api/orders/66cef14d7682cd616be0c0ab
```

```
{
    "orderid":"66cef14d7682cd616be0c0ab"
}
```

200 OK

```
{
    "_id": "66cef14d7682cd616be0c0ab",
    "orderItems": [
        {
```

MellyFashion / seed

```
GET https://melly-fashion-server-2cbd2d9a542e.herokuapp.com/api/seed
```

200 OK

```
{
    "createdProducts": [
        {
            "name": "Nike Slim shirt".
```

MellyFashion / ordersMine

```
GET https://melly-fashion-server-2cbd2d9a542e.herokuapp.com/api/orders/mine
```

```
{
    "userid":"66ceef4c7682cd616be0c08d"
}
```

200 OK

```
[]
```

MellyFashion / signin

```
POST https://melly-fashion-server-2cbd2d9a542e.herokuapp.com/api/users/signin
```

```
{
    "email":"admin@example.com",
    "password":"123456"
}
```

200 OK

```
{
    "_id": "66d09b870134614bdf91da2f",
    "name": "Joe",
    "email": "admin@example.com",
```

MellyFashion / paypal

```
GET https://melly-fashion-server-2cbd2d9a542e.herokuapp.com/api/keys/paypal
```

200 OK

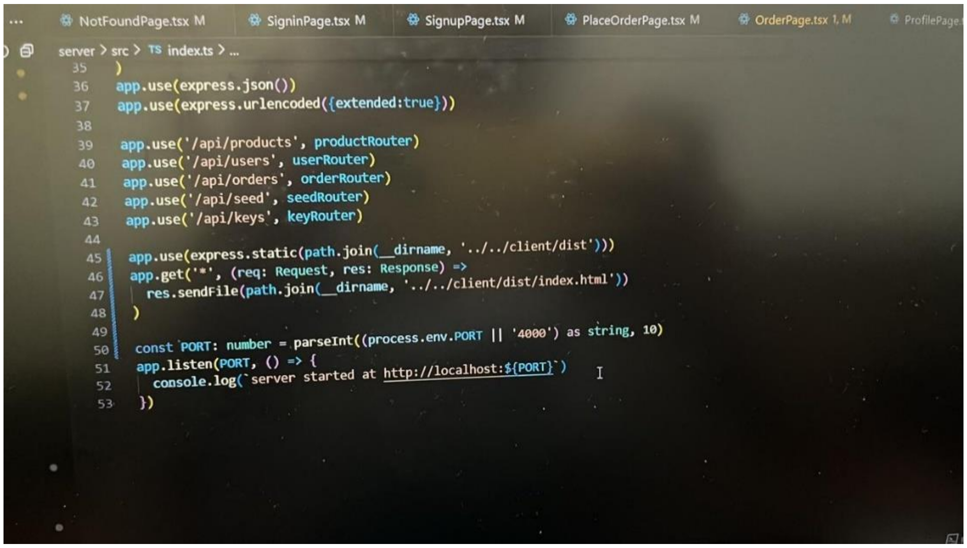
```
{
    "clientId": "Aavkx2j02nHQazP6RQqlU9QLhu8RPe8bCUHvS_yVd-j2pU1ANCH5La6mZux1scIL-ncQX84eFN1zXjCt"
}
```

Figure 21 - API check after deployment

8. Issues and Results

8.1. Issues

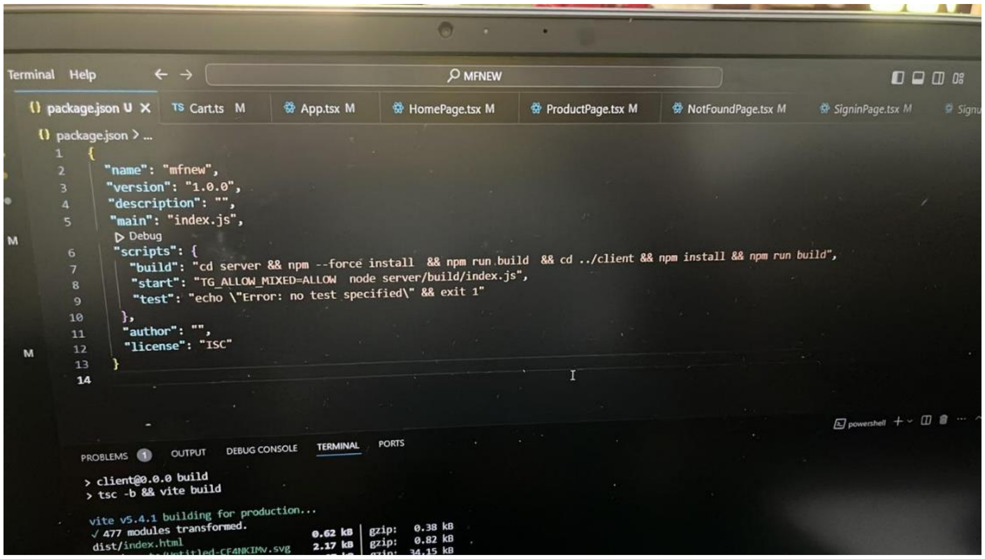
First we followed the steps below and tried to connect the client and server.



```
)
app.use(express.json())
app.use(express.urlencoded({extended:true}))

app.use('/api/products', productRouter)
app.use('/api/users', userRouter)
app.use('/api/orders', orderRouter)
app.use('/api/seed', seedRouter)
app.use('/api/keys', keyRouter)

app.use(express.static(path.join(__dirname, '../../client/dist')))
app.get('*', (req: Request, res: Response) =>
  res.sendFile(path.join(__dirname, '../../client/dist/index.html'))
)

const PORT: number = parseInt((process.env.PORT || '4000') as string, 10)
app.listen(PORT, () => {
  console.log(`server started at http://localhost:${PORT}`)
})
```

Code Snippet 2 - client + server

Then a packagege.json file was created on the root folder by giving ***npm init*** and according to Code Snippet 3 its scripts were set and ***npm run build*** was given and it was successful.

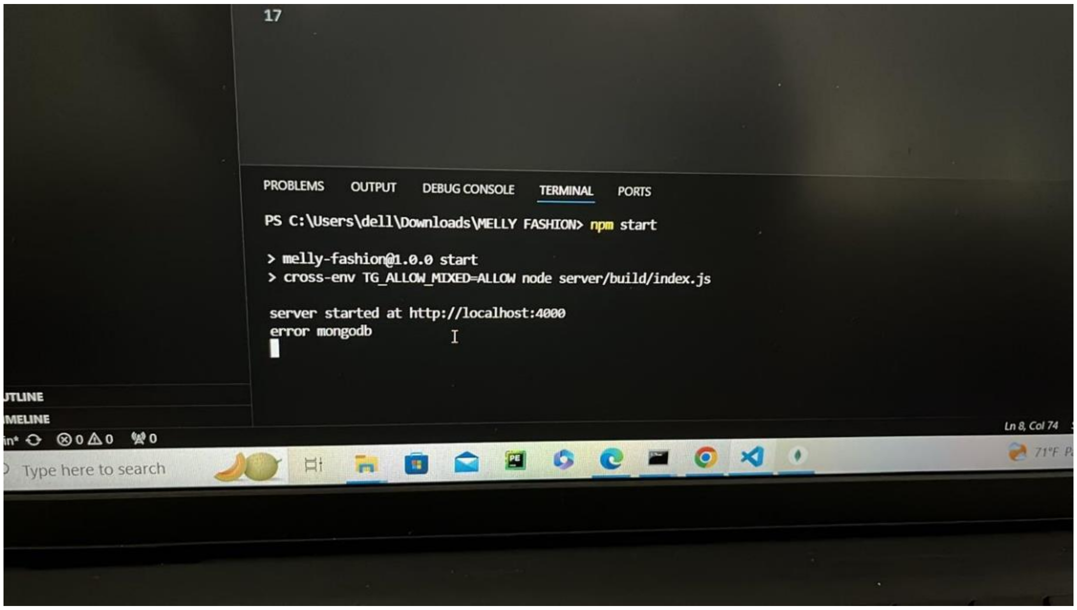


```
{
  "name": "mfnew",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "build": "cd server && npm --force install  && npm run build  && cd ../client && npm install && npm run build",
    "start": "TG_ALLOW_MIXED=ALLOW  node server/build/index.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

```
> client@0.0.0 build
> tsc -b && vite build

vite v5.4.1 building for production...
✓ 477 modules transformed.
dist/index.html                          0.62 kB │ gzip:  0.38 kB
                                         2.17 kB │ gzip:  0.82 kB
                                                 │ gzip: 34.15 kB
```

Code Snippet 3 - set scripts

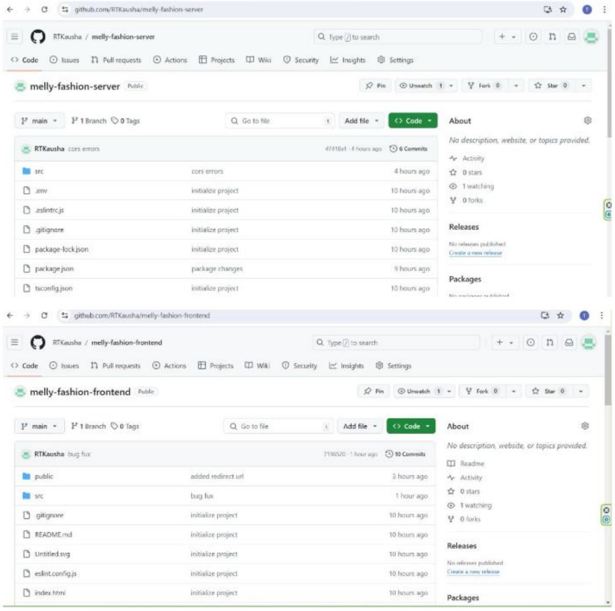
Then, when running npm start, the error mongodb was displayed and it was difficult to solve the problem, so we had to find another way to host this web app.



```
PS C:\Users\dell\Downloads\MELLY FASHION> npm start

> melly-fashion@1.0.0 start
> cross-env TG_ALLOW_MIXED=ALLOW node server/build/index.js

server started at http://localhost:4000
error mongodb
```

Code Snippet 4 - error mongodb

Accordingly, the server side was hosted in heroku and the obtained url was connected to the client folder and it was deployed in netlify.

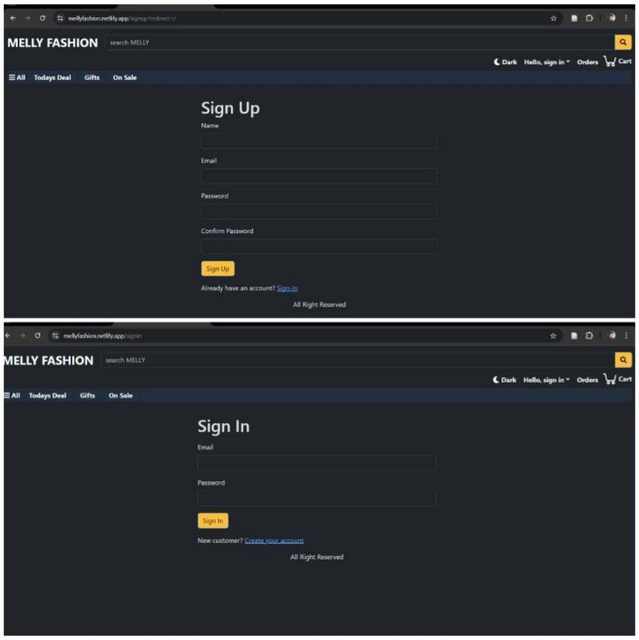


RTKausha cors errors	47418a1 · 4 hours ago	6 Commits
src	cors errors	4 hours ago
.env	initialize project	10 hours ago
.eslintrc.js	initialize project	10 hours ago
.gitignore	initialize project	10 hours ago
package-lock.json	initialize project	10 hours ago
package.json	package changes	9 hours ago
tsconfig.json	initialize project	10 hours ago

RTKausha bug fux	7196520 · 1 hour ago	10 Commits
public	added redirect url	3 hours ago
src	bug fux	1 hour ago
.gitignore	initialize project	10 hours ago
README.md	initialize project	10 hours ago
Untitled.svg	initialize project	10 hours ago
eslint.config.js	initialize project	10 hours ago
index.html	initialize project	10 hours ago

Figure 22 - new setup to host

8.2. Deployed Application View



Sign Up

Name

Email

Password

Confirm Password

Already have an account? Sign-In

All Right Reserved

Sign In

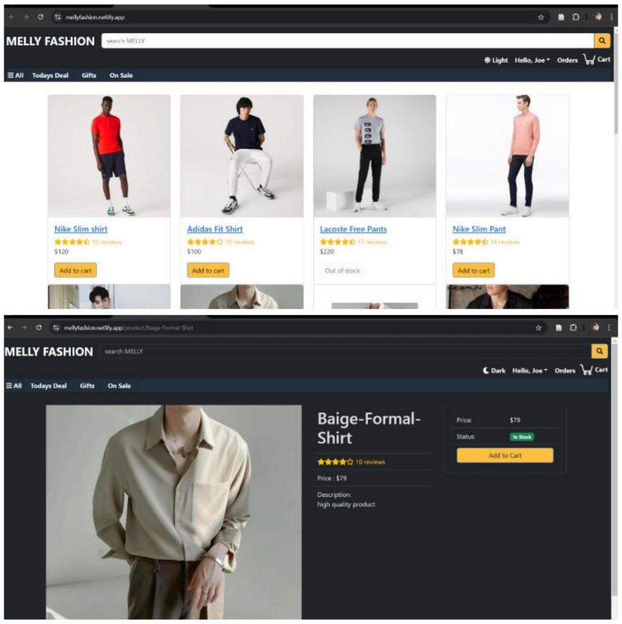
Email

Password

New customer? Create your account

All Right Reserved

Figure 23 - signin/ signup



Nike Slim shirt
10 reviews
$120
Add to cart

Adidas Fit Shirt
10 reviews
$100
Add to cart

Lacoste Free Pants
17 reviews
$220
Out of stock

Nike Slim Pant
14 reviews
$78
Add to cart

Baige-Formal-Shirt

10 reviews

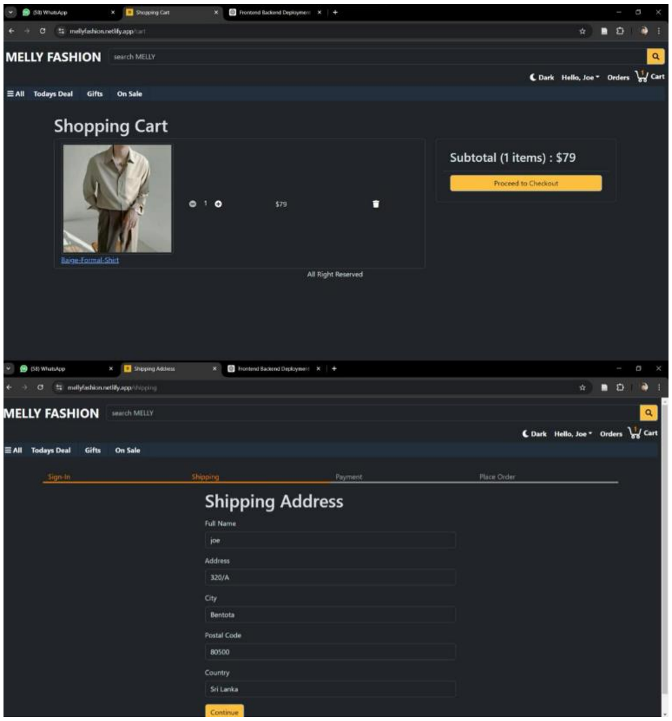
Price : $79

Description:
high quality product

Price: $79

Status: In Stock

Figure 24 - Home and Product Screens



Shopping Cart

Baige-Formal-Shirt	1	$79	

Subtotal (1 items) : $79

Shipping Address

Full Name: joe

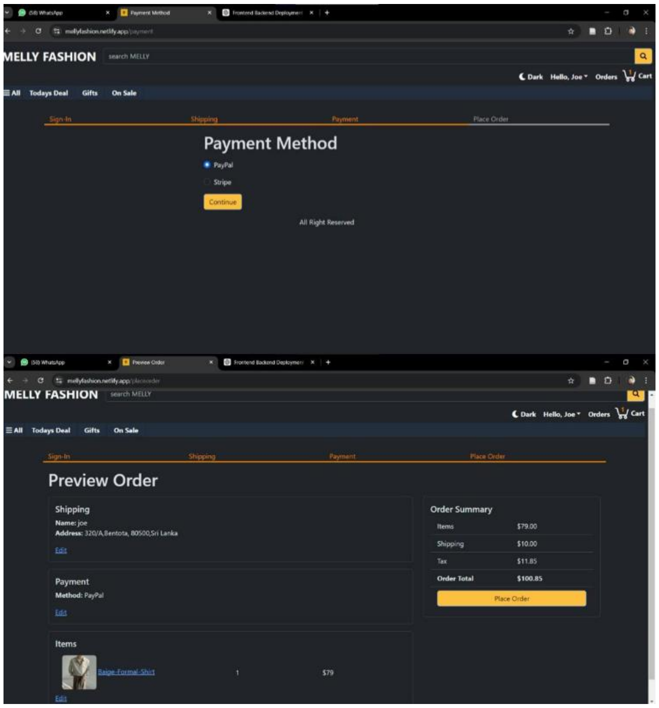
Address: 320/A

City: Bentota

Postal Code: 80500

Country: Sri Lanka

Figure 25 - Cart and Shipping Adress Screens



Payment Method

● PayPal

○ Stripe

Preview Order

Shipping

Name: joe

Address: 320/A,Bentota, 80500,Sri Lanka

Payment

Method: PayPal

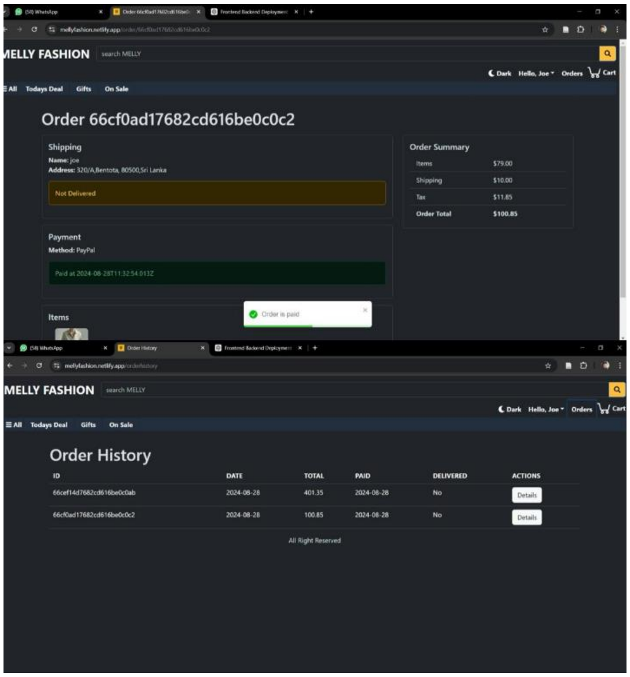
Items

Baige-Formal-Shirt	1	$79

Order Summary

Items	$79.00
Shipping	$10.00
Tax	$11.85
Order Total	**$100.85**

Figure 26 - Payment and Place Order Screens



Order 66cf0ad17682cd616be0c0c2

Shipping

Name: joe

Address: 320/A,Bentota, 80500,Sri Lanka

Not Delivered

Payment

Method: PayPal

Paid at 2024-08-28T11:32:54.013Z

Items

Order is paid

Order Summary

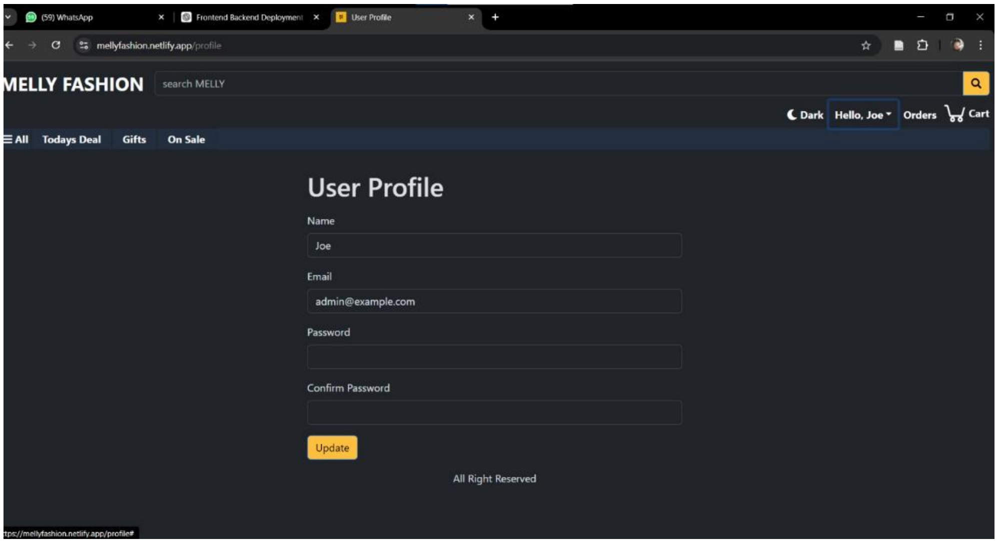
Items	$79.00
Shipping	$10.00
Tax	$11.85
Order Total	**$100.85**

Order History

ID	DATE	TOTAL	PAID	DELIVERED	ACTIONS
66cef14d7682cd616be0c0ab	2024-08-28	401.35	2024-08-28	No	Details
66cf0ad17682cd616be0c0c2	2024-08-28	100.85	2024-08-28	No	Details

All Right Reserved

Figure 27 - Order and Order History Screens



User Profile

Name

Joe

Email

admin@example.com

Password

Confirm Password

All Right Reserved

Figure 28 - User Profile Screens

Melly fashion is also mobile responsive and Windows and macOS support application.

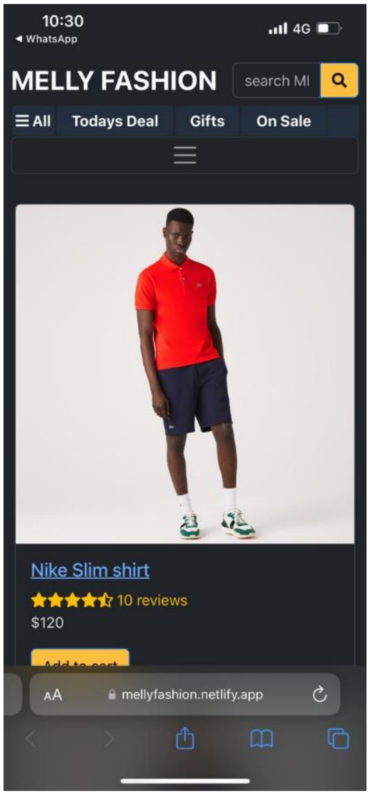


Figure 30 - Mobile view

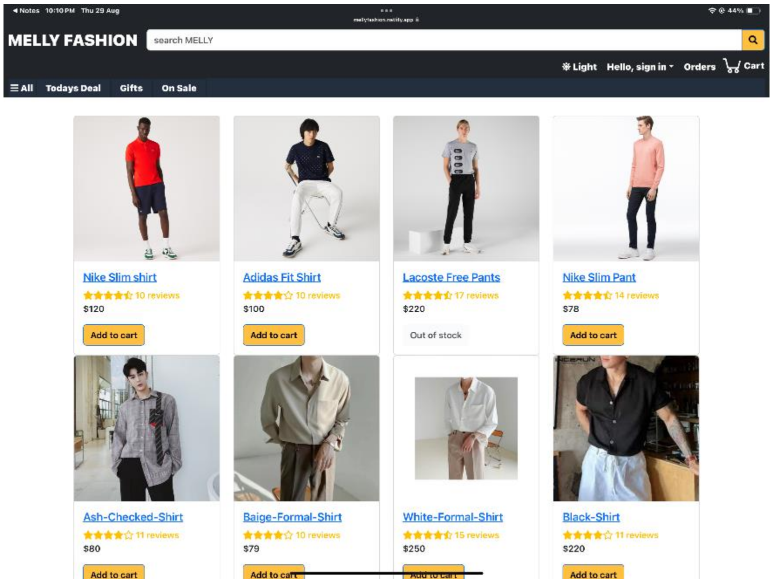


Figure 29 - macOS view

9. Conclusion / Summary

Melly Fashion is an advanced online shopping system that covers buyer’s journey from exploring products to making the final purchase. This platform is developed utilizing best practice modern technology stack like TypeScript and React on the front-end, Node. js to carry out all the backend works and MongoDB for the database. Together with the presented technologies, it delivers the fast, reactive, and intuitive UI and it is served by the server with a high loading capacity and suitable for fulfilling complex operations.

This paper presents the structure of this platform to provide the reader with an idea on how the system is designed to allow for scalability, security, and performance architectures. This document describes the structure of database; it explains the purpose of different collections addressing the user data, product information, orders, etc. The features of each API endpoint are described in details along with how they enable basic user login/update, product browse/add/edit, cart management, and order making.

In addition, the document expands on the deployment process on Backend in Heroku and Frontend in Netlify, in which it will be noted that the dependability of the hosting platform is critical to have a successful site since it may be suddenly unavailable or slow. Of course, reading this document gives developers a good idea of the nuances involved in working with Melly Fashion platform and can be regarded as a helpful advice on how to create and implement e-commerce systems.