Smart Trial Studio

Kashmira Chaudhari 20158

1. Introduction

1. Revolutionizing the way we shop for clothing: The Smart Trial Studio offers an innovative approach to shopping. Customers can effortlessly curate a trial cart with up to 5 items from various brands and styles, simplifying the selection process.
2. Seamless Integration with Retailers: Leveraging a network of online and offline retailers, customers can conveniently try on their chosen items at a nearby store's trial room within a 1-day window. This bridges the gap between online browsing and in-store trying.
3. Enhanced Shopping Experience: This service reduces the time and hassle of traditional shopping while enhancing the overall experience with personalized convenience and flexibility.
4. API Development Goals:
   • CRUD Operations: My goal is to develop API's that can perform different operations like Create, Read, Update, and Delete (CRUD) on the database.
   • Access Control:
   o READ Access: Available to both employees and customers.

   o CREATE, UPDATE, and DELETE Access: Restricted to store employees only.

2. Tech Stack:

I. **Sprinboot:** Java Spring Boot is an open-source tool that makes it easier to use Java-based frameworks to create microservices and web apps.

II. **Mongodb:** MongoDB is the document database designed to make it easy for developers to work with data in any form, and from any programming language.

III. **Java:** Java is a versatile, object-oriented programming language designed for building platform-independent, secure, and high-performance applications.

IV. **HTML:** HTML (HyperText Markup Language) is the standard language used to create and structure content on the web by defining elements like text, images, and links.

V. **JavaScript:** JavaScript is a dynamic, high-level programming language primarily used to create interactive and responsive behavior on web pages.

VI. **CSS:** CSS (Cascading Style Sheets) is a stylesheet language used to control the presentation, layout, and design of HTML elements on a web page.

3. Architecture Diagram:

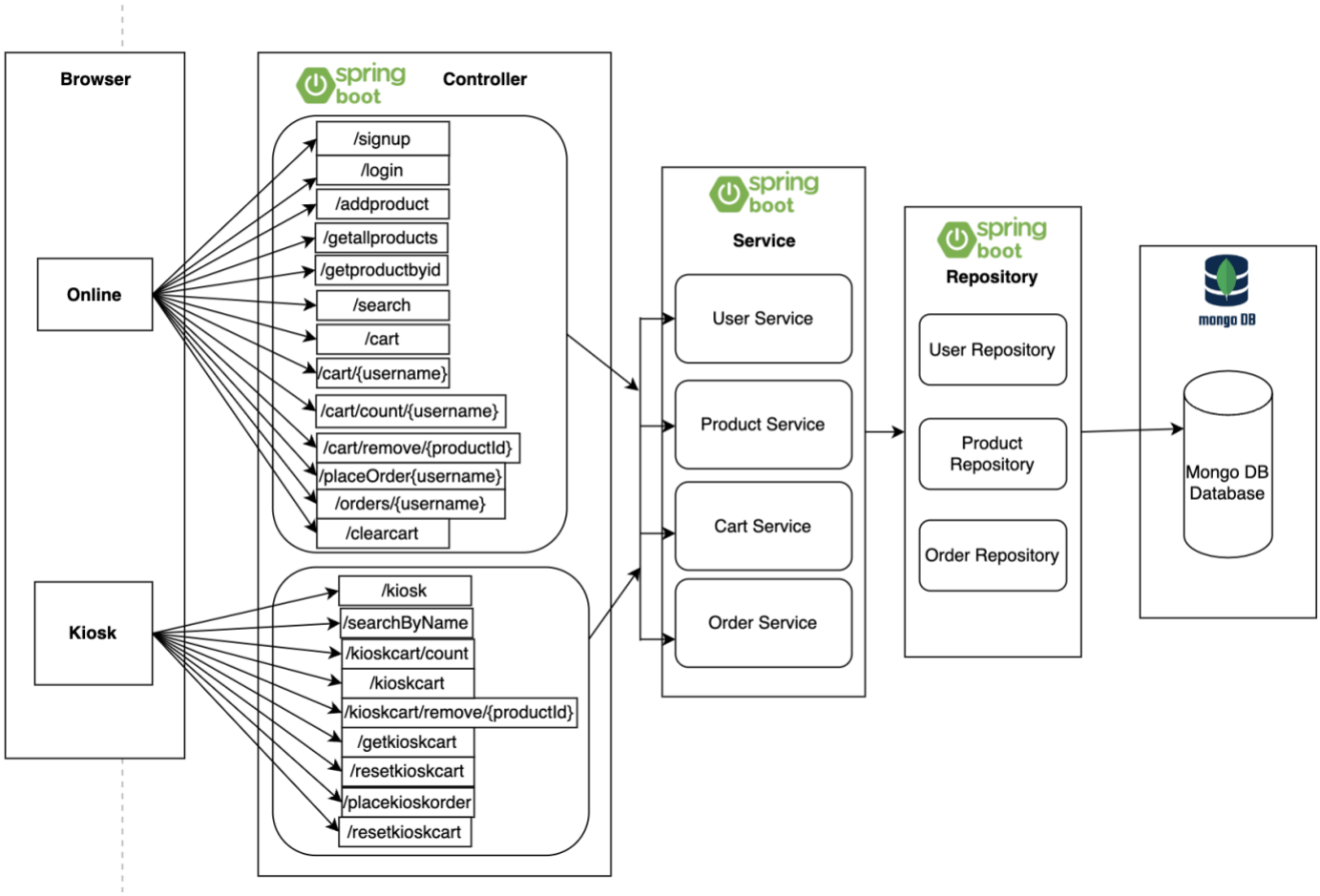


Fig: Architecture of Smart Trial Studio

4. Use Case for Smart Trial Studio

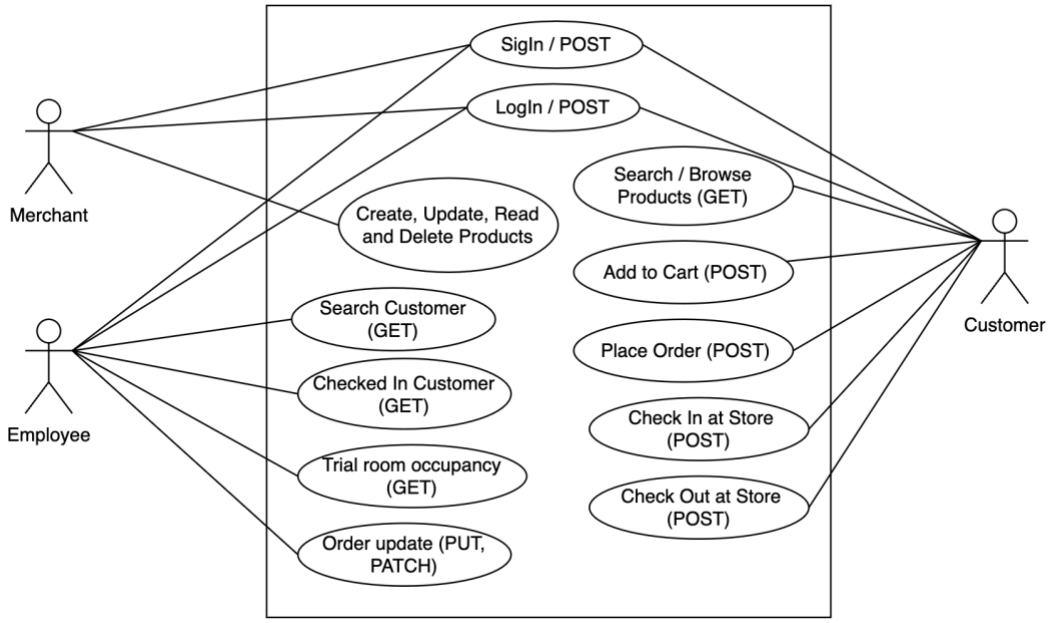


Fig: Smart Trial Studio

5. FLOW chart:

I. User/Customer Flowchart

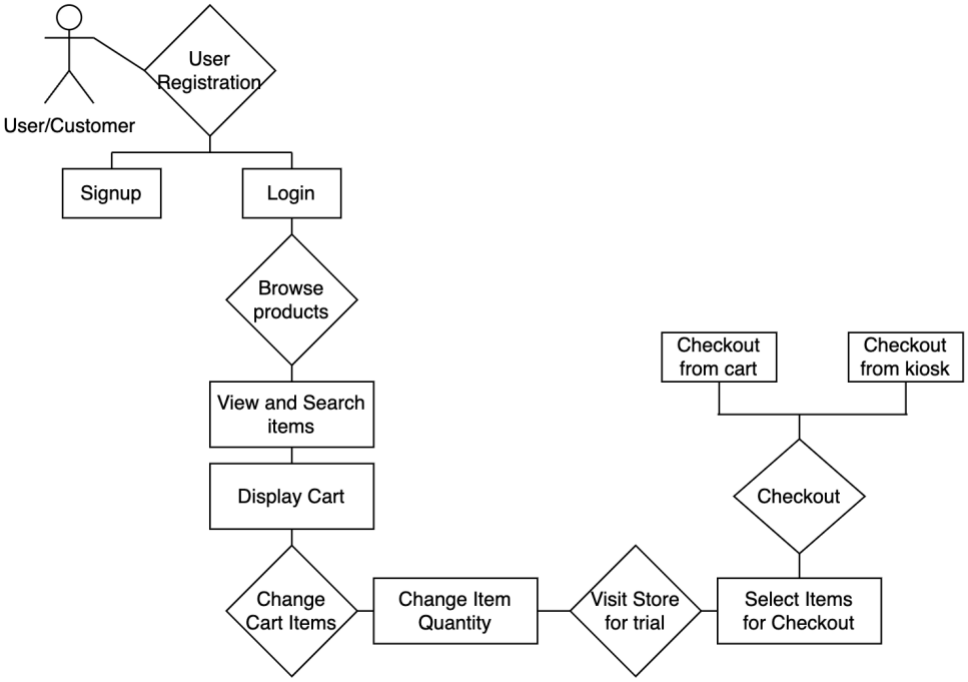


Fig: Flowchart for Customer

II. Store Flowchart:

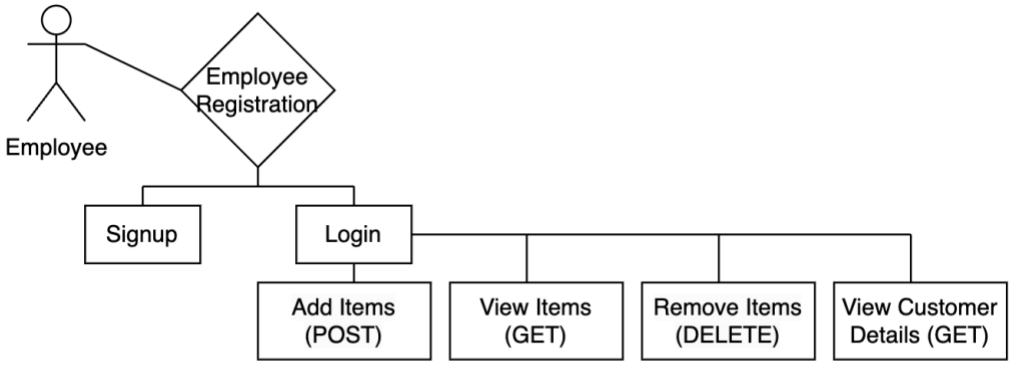
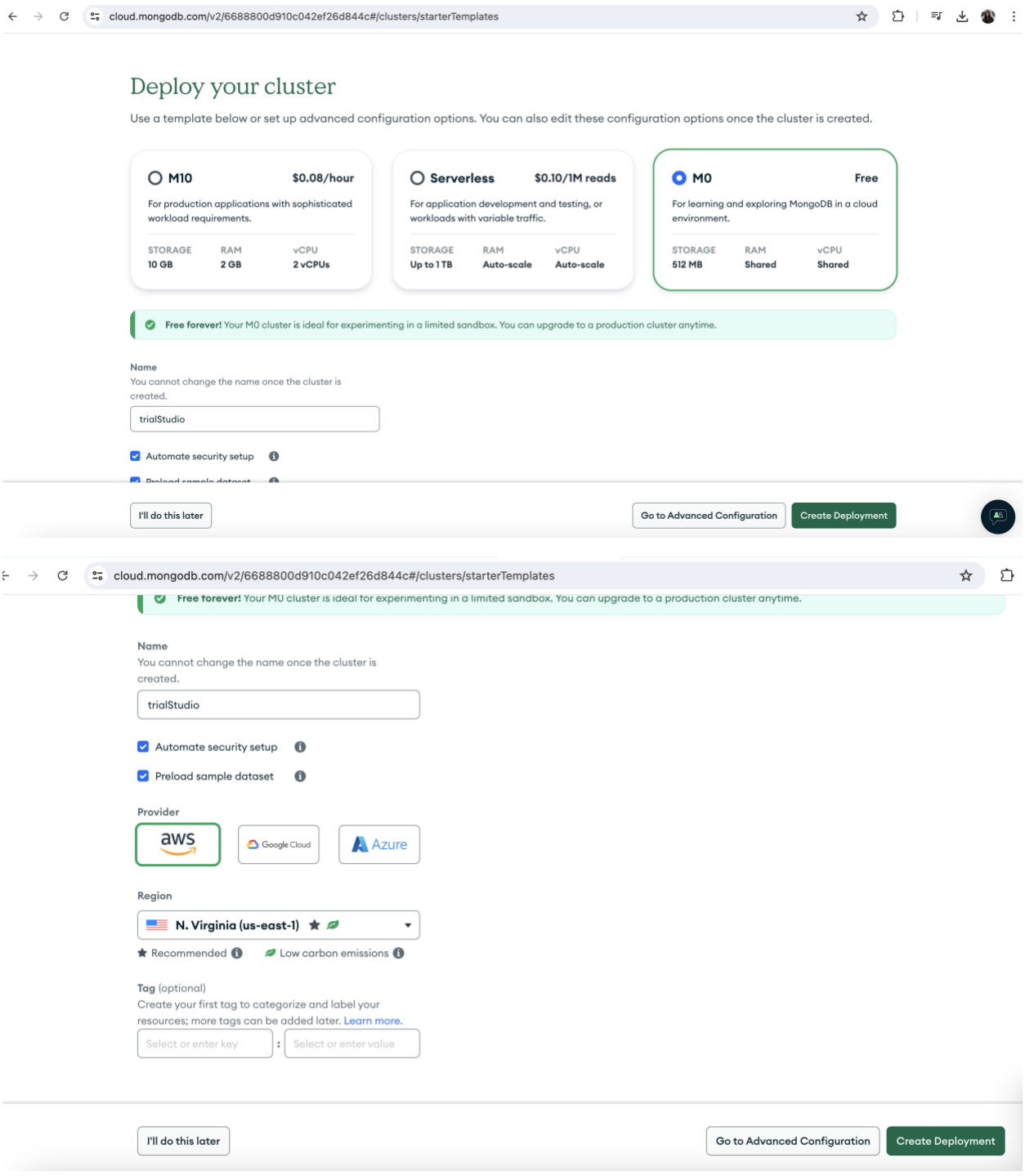
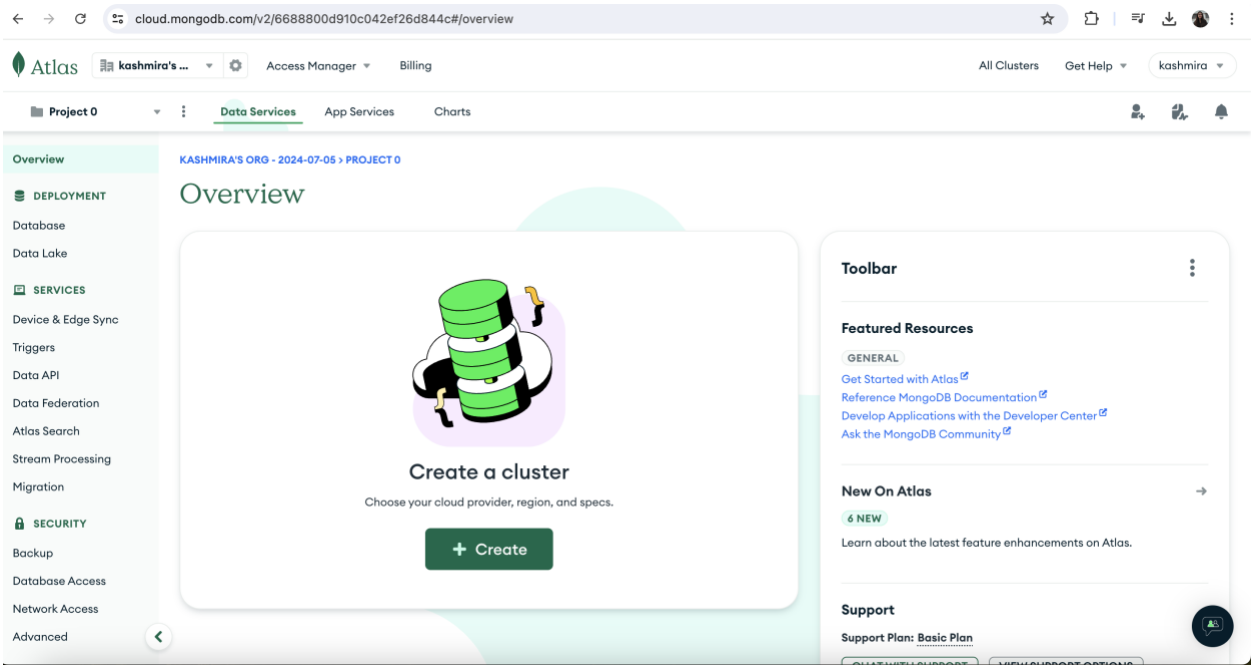


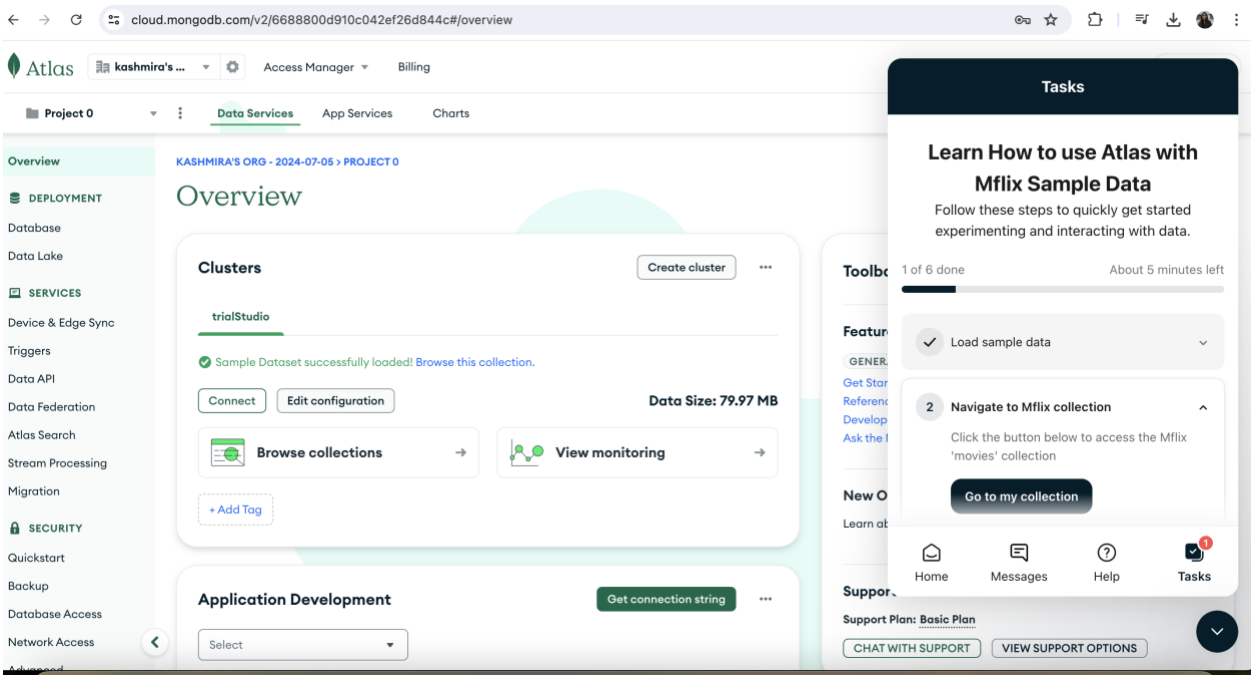
Fig: Flowchart for Store Employee

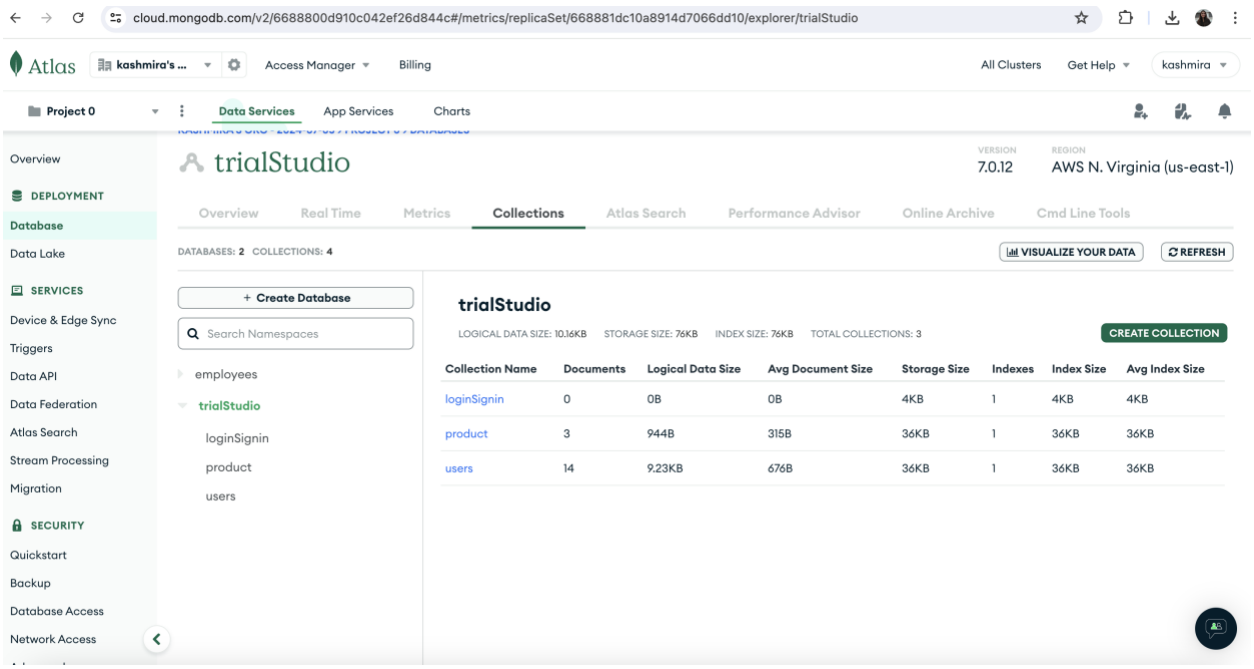
6. Setting up MongoDB:



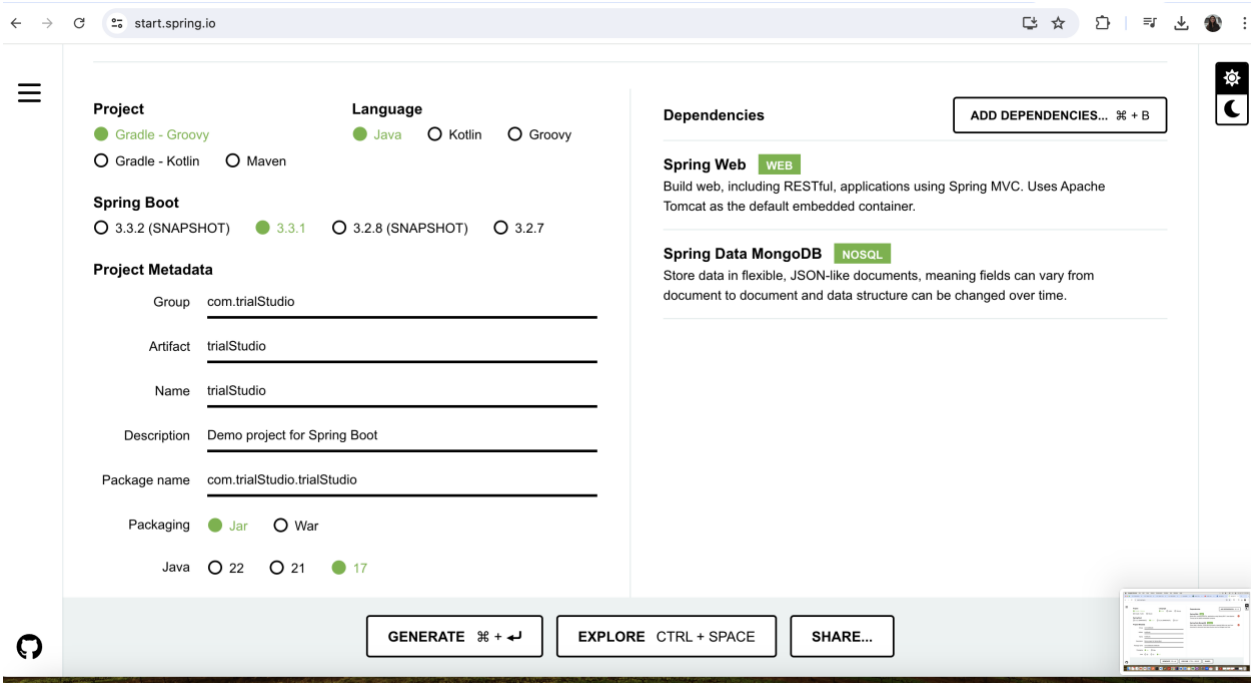


7. Created a database named “trialStudio”

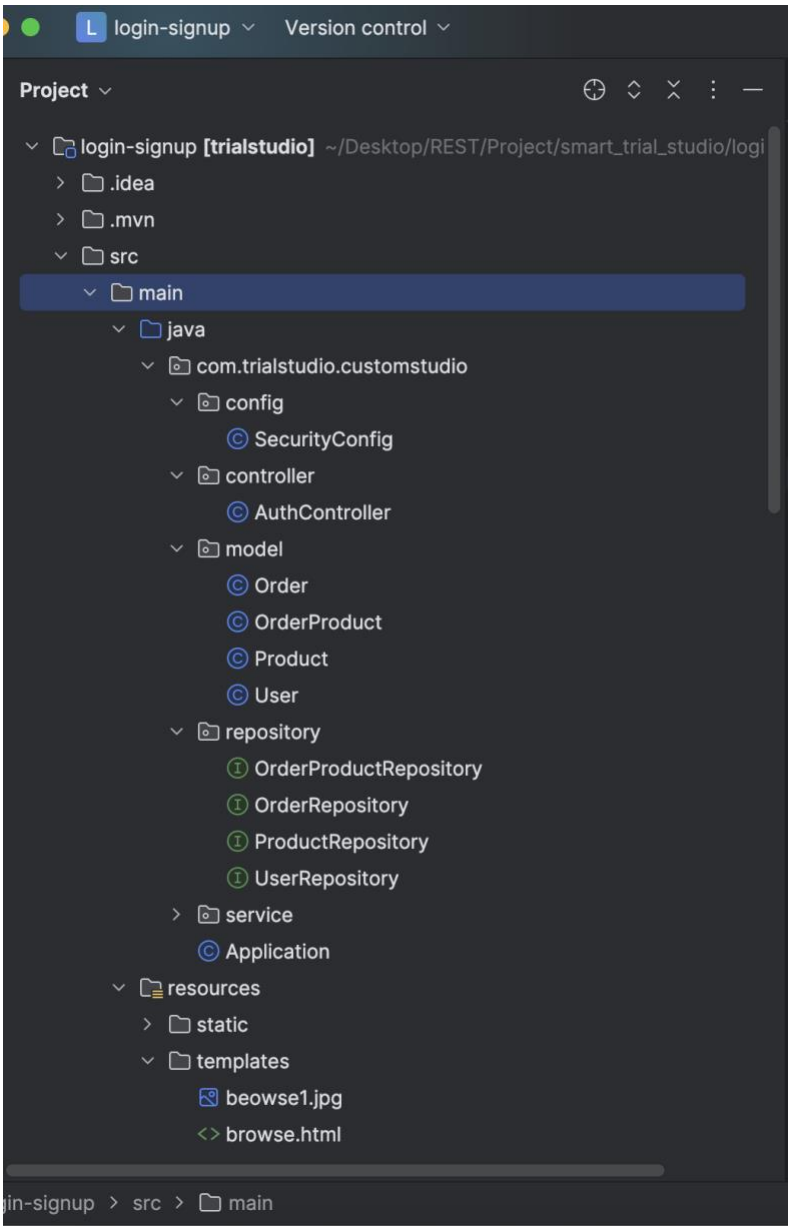


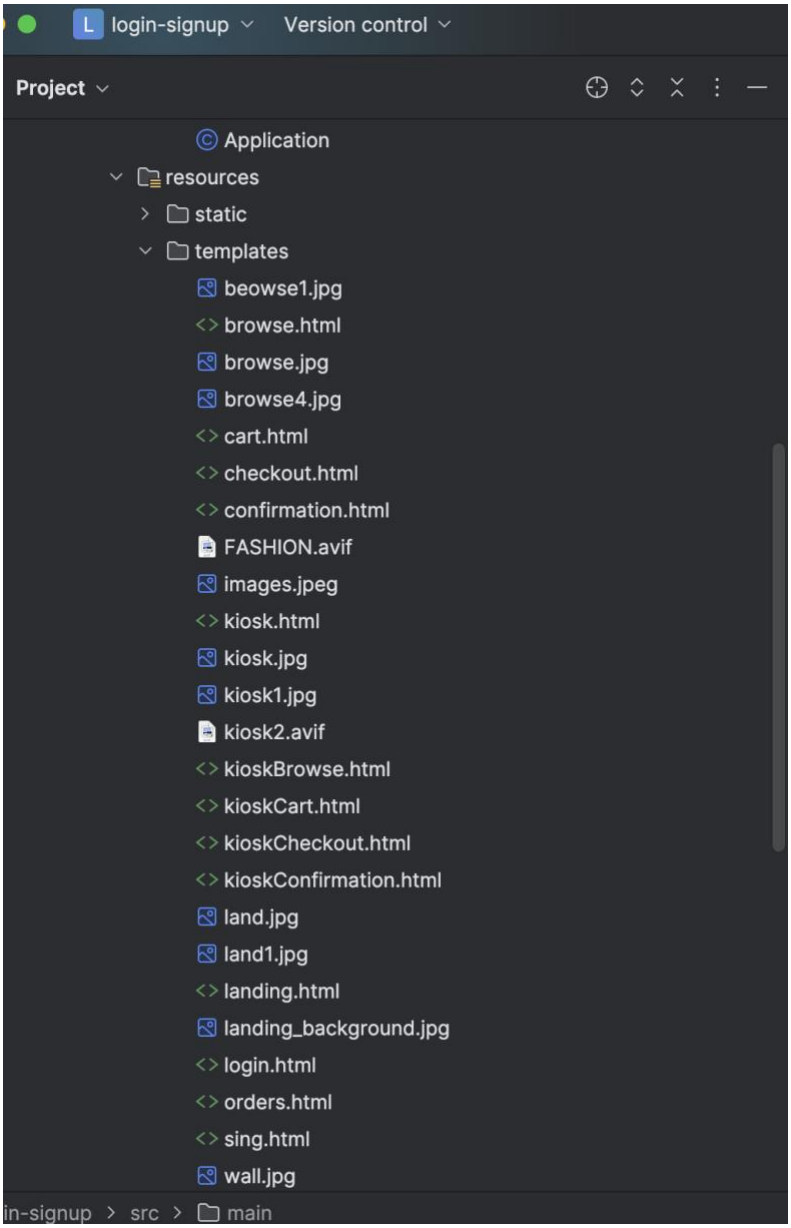


8. **Create a project using** https://start.spring.io/



9. Project Structure:



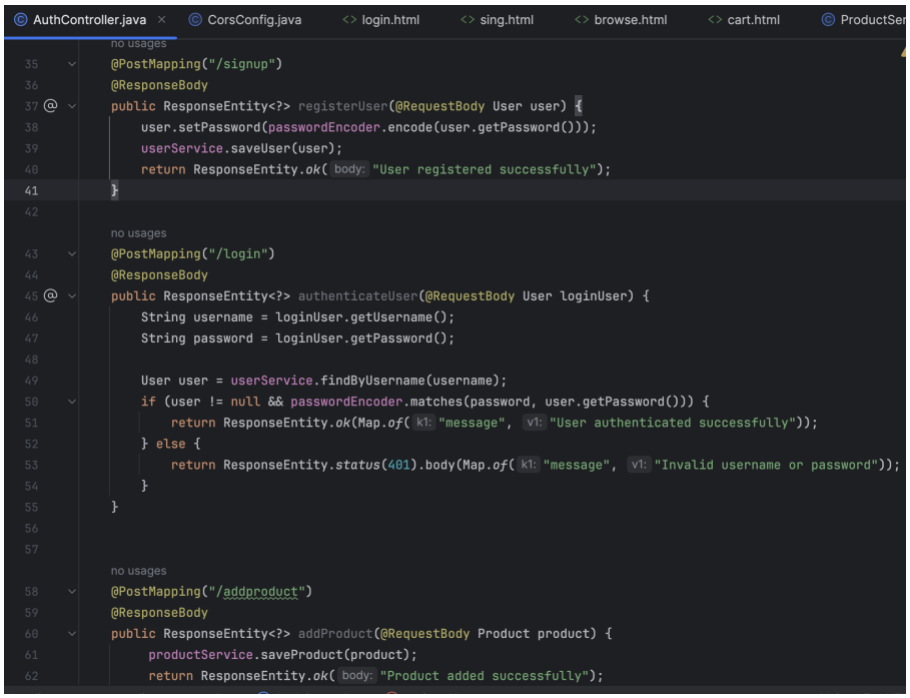


10. Created below listed API End points:

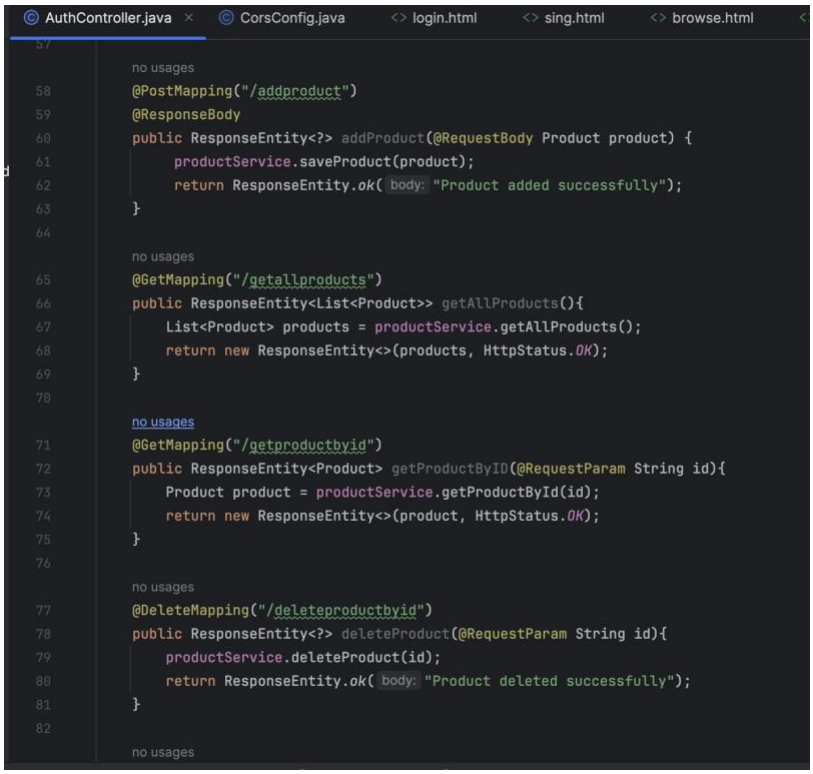
1. @PostMapping("/signup") => api/auth/signup
2. @PostMapping("/login") => api/auth/login
3. @PostMapping("/addproduct") => /api/auth/addproduct
4. @GetMapping("/getallproducts") => /api/auth/getallproducts
5. @GetMapping("/getproductbyid") => /api/auth/getproductbyid
6. @DeleteMapping("/deleteproductbyid") => /api/auth/deleteproductbyid

7. @GetMapping("/search") => /api/auth/search
8. @PutMapping("/{id}") => /api/auth/{id}
9. @PatchMapping("/{id}") => /api/auth/{id}
10. @GetMapping("/login")
11. @GetMapping("/signup")
12. @PostMapping("/cart")
13. @GetMapping("/cart/{username}")
14. @GetMapping("/cart/count/{username}")
15. @GetMapping("/kiosk/count")
16. @GetMapping("/searchByName")
17. @GetMapping("/searchByName")
18. @PostMapping("/kioskcart")
19. @DeleteMapping("/kioskcart/remove/{productId}")
20. @GetMapping("/getkioskcart")
21. @GetMapping("/kioskcart/count")
22. @GetMapping("/resetkioskcart")
23. @PostMapping("/placeOrder/{username}")
24. @GetMapping("orders/{username}")
25. @PostMapping("/placekioskorder")
26. @PostMapping("/clearcart")
27. @PostMapping("/resetkioskcart")

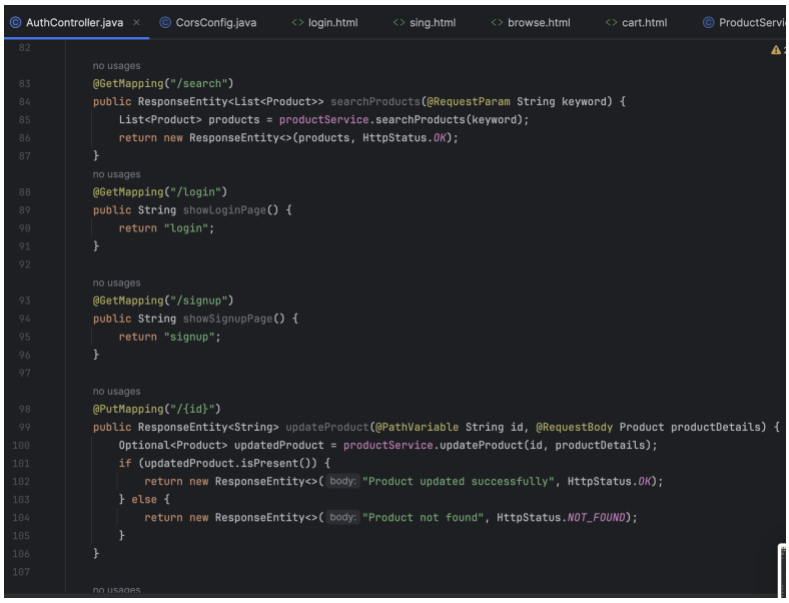
11. AuthController code to write API's:



```
AuthController.java | CorsConfig.java | login.html | sing.html | browse.html | cart.html | ProductSer

no usages
@PostMapping("/signup")
@ResponseBody
public ResponseEntity<?> registerUser(@RequestBody User user) {
    user.setPassword(passwordEncoder.encode(user.getPassword()));
    userService.saveUser(user);
    return ResponseEntity.ok( body: "User registered successfully");
}

no usages
@PostMapping("/login")
@ResponseBody
public ResponseEntity<?> authenticateUser(@RequestBody User loginUser) {
    String username = loginUser.getUsername();
    String password = loginUser.getPassword();

    User user = userService.findByUsername(username);
    if (user != null && passwordEncoder.matches(password, user.getPassword())) {
        return ResponseEntity.ok(Map.of( k1: "message",  v1: "User authenticated successfully"));
    } else {
        return ResponseEntity.status(401).body(Map.of( k1: "message",  v1: "Invalid username or password"));
    }
}


no usages
@PostMapping("/addproduct")
@ResponseBody
public ResponseEntity<?> addProduct(@RequestBody Product product) {
     productService.saveProduct(product);
     return ResponseEntity.ok( body: "Product added successfully");
```



```
AuthController.java | CorsConfig.java | login.html | sing.html | browse.html

no usages
@PostMapping("/addproduct")
@ResponseBody
public ResponseEntity<?> addProduct(@RequestBody Product product) {
     productService.saveProduct(product);
     return ResponseEntity.ok( body: "Product added successfully");
}

no usages
@GetMapping("/getallproducts")
public ResponseEntity<List<Product>> getAllProducts(){
    List<Product> products = productService.getAllProducts();
    return new ResponseEntity<>(products, HttpStatus.OK);
}

no usages
@GetMapping("/getproductbyid")
public ResponseEntity<Product> getProductByID(@RequestParam String id){
    Product product = productService.getProductById(id);
    return new ResponseEntity<>(product, HttpStatus.OK);
}

no usages
@DeleteMapping("/deleteproductbyid")
public ResponseEntity<?> deleteProduct(@RequestParam String id){
    productService.deleteProduct(id);
    return ResponseEntity.ok( body: "Product deleted successfully");
}

no usages
```




```
no usages
@PostMapping("/cart")  // To add the product into the cart
@ResponseBody
public ResponseEntity<String> addToCart(@RequestBody Map<String, String> request) {
    String username = request.get("username");
    String productId = request.get("productId");

    // Implement logic to add product to cart for the user
    // Example:
    Optional<User> user = userService.addProductToCart(username, productId);
    if (user.isPresent()) {
        return ResponseEntity.ok( body: "Product added to cart");
    } else {
        return ResponseEntity.status(HttpStatus.BAD_REQUEST).body("User not found");
    }
}

no usages
@PostMapping("/kiosk")  // To add the product into the KIOSK
@ResponseBody
public ResponseEntity<String> addToKiosk(@RequestBody Map<String, String> request) {
    String username = request.get("username");
    String productId = request.get("productId");

    // Implement logic to add product to cart for the user
    // Example:
    Optional<User> user = userService.addProductToKiosk(username, productId);
    if (user.isPresent()) {
        return ResponseEntity.ok( body: "Product added to Kiosk");
    } else {
```

```
    no usages
    @DeleteMapping("/cart/remove/{productId}")
    @ResponseBody
    public ResponseEntity<String> removeFromCart(@PathVariable String productId, @RequestParam String username) {
        boolean isRemoved = userService.removeProductFromCart(username, productId);
        if (isRemoved) {
            return ResponseEntity.ok( body: "Product removed from cart");
        } else {
            return ResponseEntity.status(HttpStatus.BAD_REQUEST).body("Product or user not found");
        }
    }

    no usages
    @GetMapping("/cart/{username}")
    public ResponseEntity<List<Product>> getCartItems(@PathVariable String username) {
        //String username = getUsernameFromSecurityContext();
        Optional<List<Product>> cartItems = userService.getCartItems(username);
        if (cartItems.isPresent()) {
            return ResponseEntity.ok(cartItems.get());
        } else {
            return ResponseEntity.status(400).body(null);
        }
    }
```

```
    //Cart count api for browse.html page
    no usages
    @GetMapping("/cart/count/{username}")
    public ResponseEntity<Integer> getCartItemCount(@PathVariable String username) {
        int itemCount = userService.getCartItemCount(username);
        return ResponseEntity.ok(itemCount);
    }

//    Kiosk API's


    no usages
    @GetMapping("/kiosk/count")
    public ResponseEntity<Integer> getKioskItemCount() {
        // Assuming you have a method in CartService that gets the count of items in the kiosk cart
        int itemCount = cartService.getCartItemCount();
        return ResponseEntity.ok(itemCount);
    }

    no usages
    @GetMapping("/searchByName")
    @ResponseBody
    public ResponseEntity<List<Product>> searchProductsByName(@RequestParam String keyword) {
        List<Product> products = productService.searchProductsByName(keyword);
        return new ResponseEntity<>(products, HttpStatus.OK);
    }
```

```java
@PostMapping("/kioskcart")
@ResponseBody
public ResponseEntity<String> addKioskToCart(@RequestBody Map<String, String> request) {
    cartService.addToCart(request.get("productId"));
    return ResponseEntity.ok( body: "Product added to cart");
}

no usages
@DeleteMapping("/kioskcart/remove/{productId}")
@ResponseBody
public ResponseEntity<String> removeFromCart(@PathVariable String productId) {
    cartService.removeFromCart(productId);
    return ResponseEntity.ok( body: "Product removed from cart");
}

no usages
@GetMapping("/getkioskcart")
public ResponseEntity<List<Product>> getKioskCartItems() {
    List<Product> cartItems = cartService.getCartItems();
    return new ResponseEntity<>(cartItems, HttpStatus.OK);
}

no usages
@GetMapping("/kioskcart/count")
public ResponseEntity<Integer> getCartItemCount() {
    int itemCount = cartService.getCartItemCount();
    return ResponseEntity.ok(itemCount);
}
```

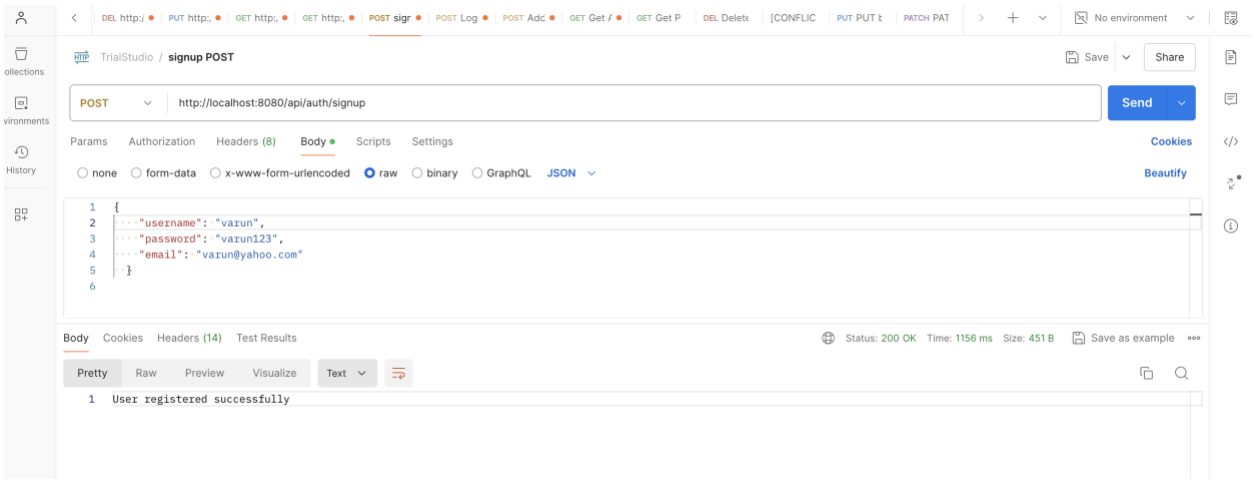
```java
no usages
@GetMapping("/resetkioskcart")
public ResponseEntity<String> clearKisokCart(){
    cartService.clearCart();
    return ResponseEntity.ok( body: "Cart has been reset");
}
```

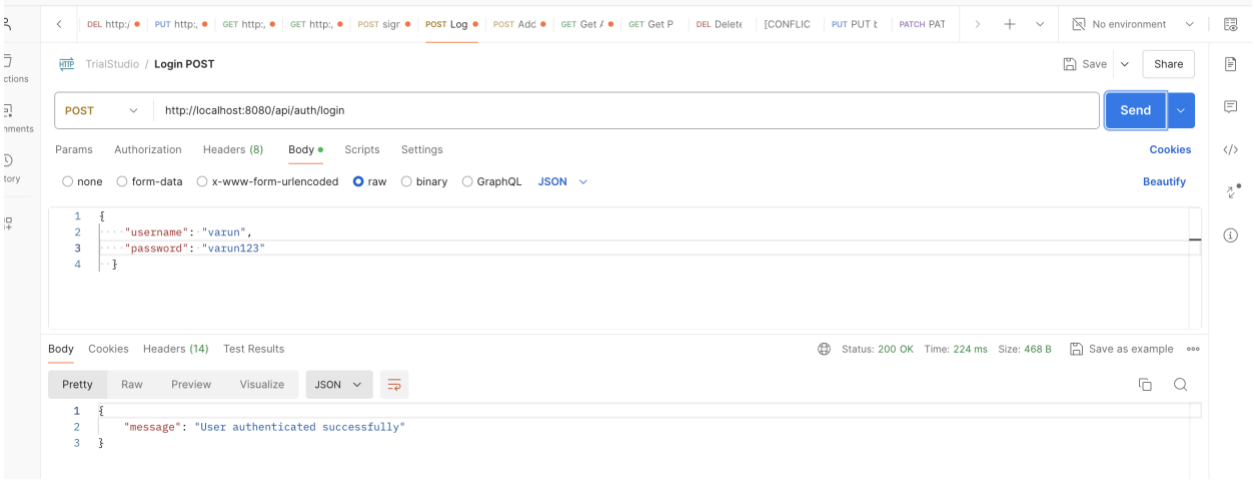
```java
no usages
@PostMapping("/placeOrder/{username}")
public  ResponseEntity<String> placeOrder(@PathVariable String username, @RequestBody Order order) {
    userService.placeOrder(username, order);
    //orderService.placeOrder(username, order);
    return ResponseEntity.ok( body: "Order Placed");
}
```

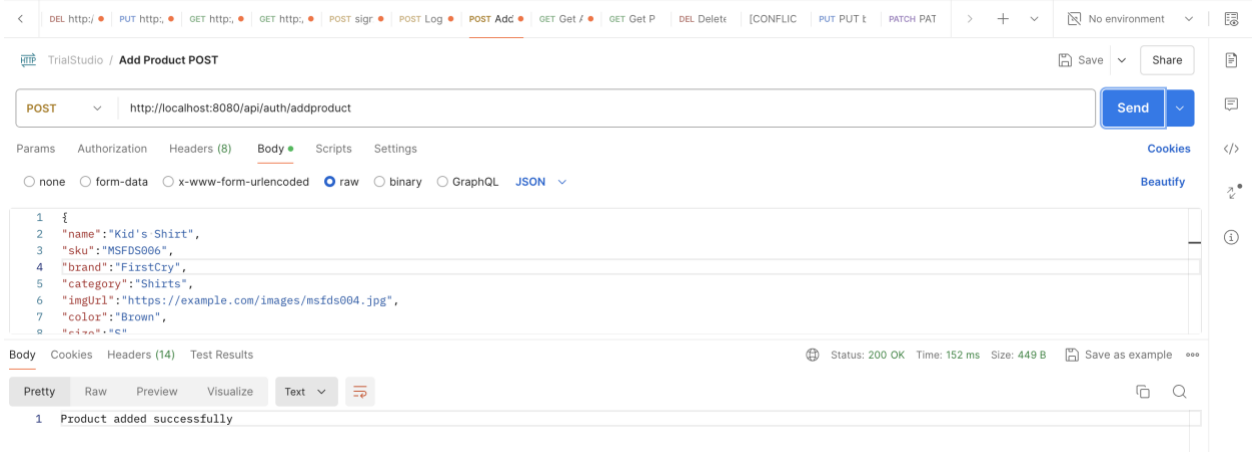
```
    no usages
    @GetMapping("orders/{username}")
    public ResponseEntity<List<Order>> getOrders(@PathVariable String username) {
        List<Order> orders = orderService.getOrdersByUsername(username);
        return ResponseEntity.ok(orders);
    }

    no usages
    @PostMapping("/placekioskorder")
    public ResponseEntity<String> placeOrder(@RequestBody Order order) {
        orderService.saveOrder(order); // Assuming saveOrder method is implemented in the OrderService
        return ResponseEntity.ok( body: "Order Placed Successfully");
    }

//   browser clear cart: confirmation page ---> browser page

    no usages
    @PostMapping("/clearcart")
    public ResponseEntity<Void> clearCart(@RequestBody Map<String, String> request) {
        String username = request.get("username");
        User user = userService.findByUsername(username);
        if (user != null) {
            user.getCart().clear();
            userService.save(user); // Save the user to update the cart in the database
            return ResponseEntity.ok().build();
        } else {
            return ResponseEntity.status(HttpStatus.NOT_FOUND).build();
        }
    }
```

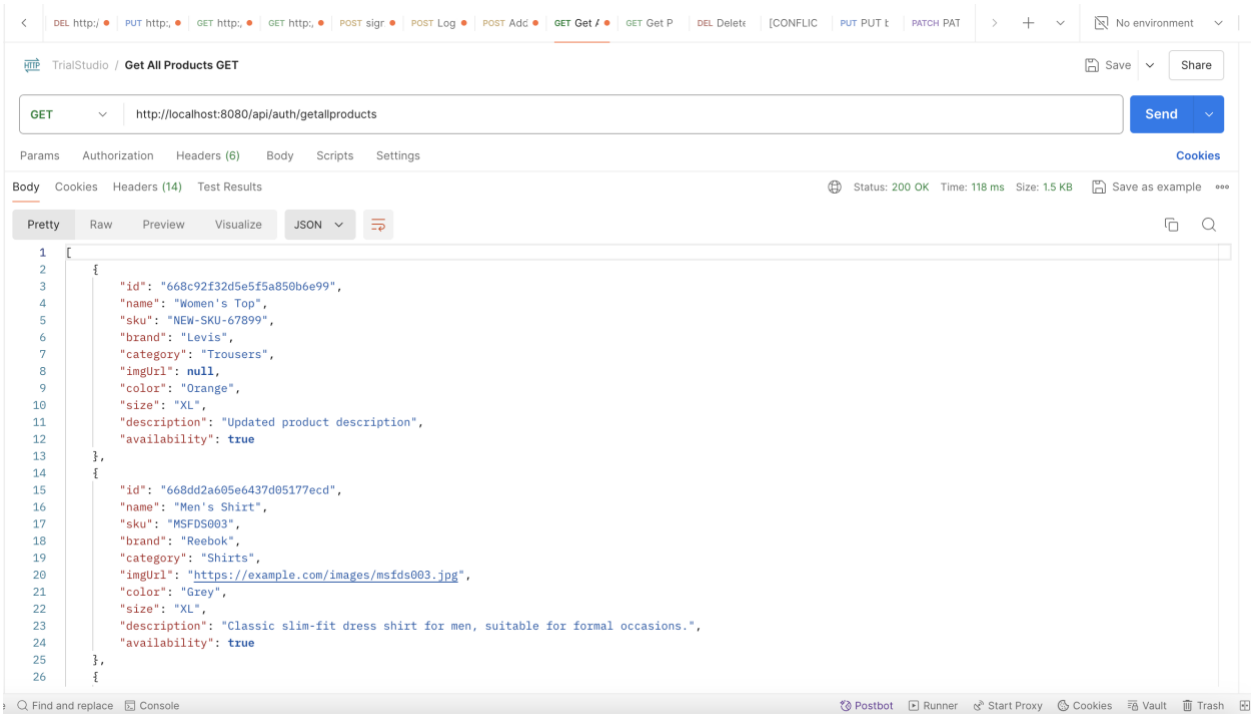
```
//    kiosk clear cart: kioskconfirmation page ---> kioskbrowser page
    no usages
    @PostMapping("/resetkioskcart")
    public ResponseEntity<Void> resetKioskCart() {
        cartService.clearCart(); // Assuming clearCart is a method in cartService that clears the kiosk cart
        return ResponseEntity.ok().build();
    }


}
```

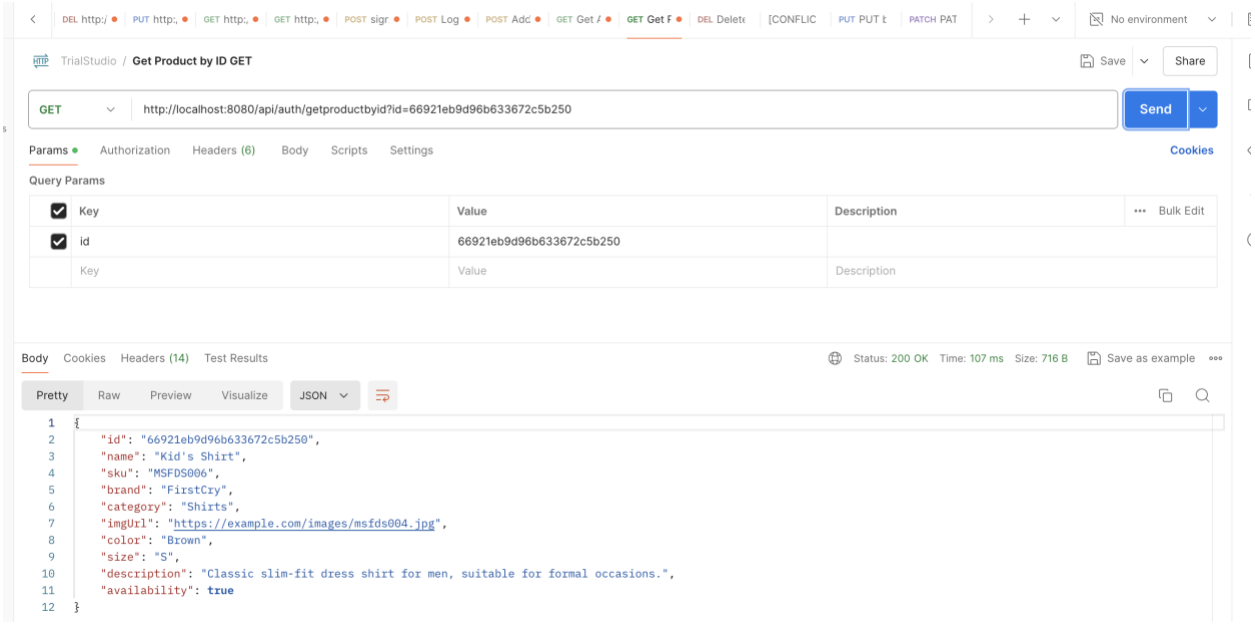
12. API response on Postman:

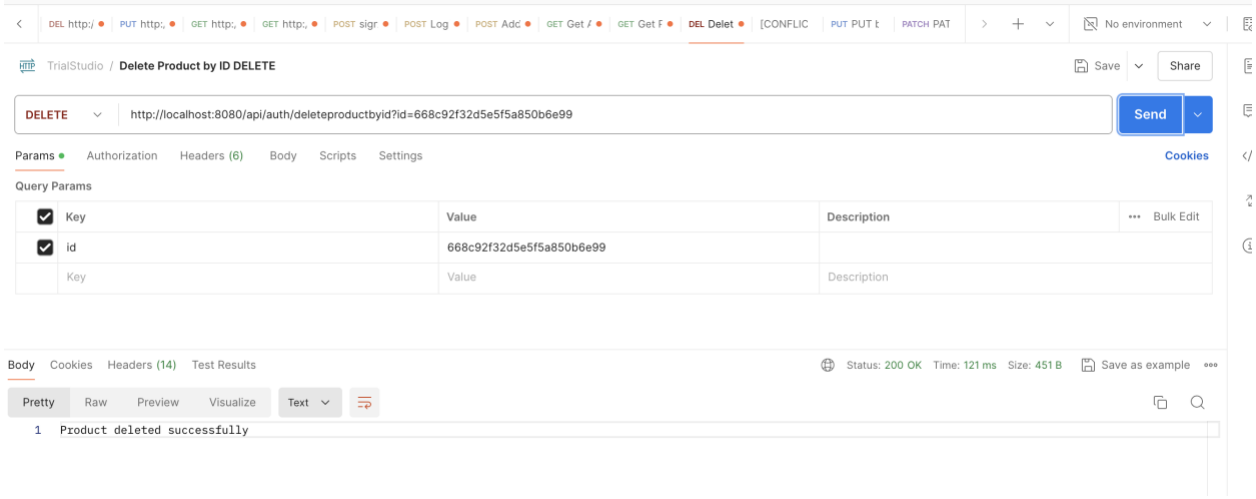

TrialStudio / signup POST
Save
Share
POST
http://localhost:8080/api/auth/signup
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"username": "varun",
"password": "varun123",
"email": "varun@yahoo.com"
}
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 1156 ms
Size: 451 B
Save as example
Pretty
Raw
Preview
Visualize
Text
User registered successfully

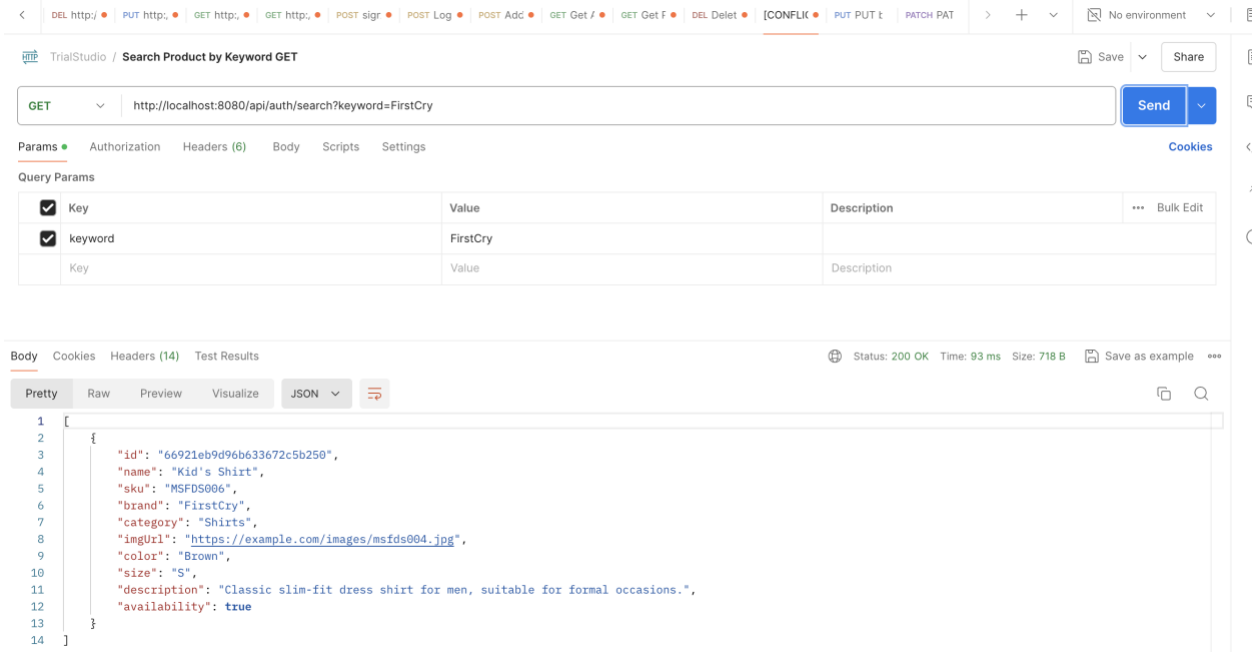

TrialStudio / Login POST
Save
Share
POST
http://localhost:8080/api/auth/login
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"username": "varun",
"password": "varun123"
}
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 224 ms
Size: 468 B
Save as example
Pretty
Raw
Preview
Visualize
JSON
{
"message": "User authenticated successfully"
}

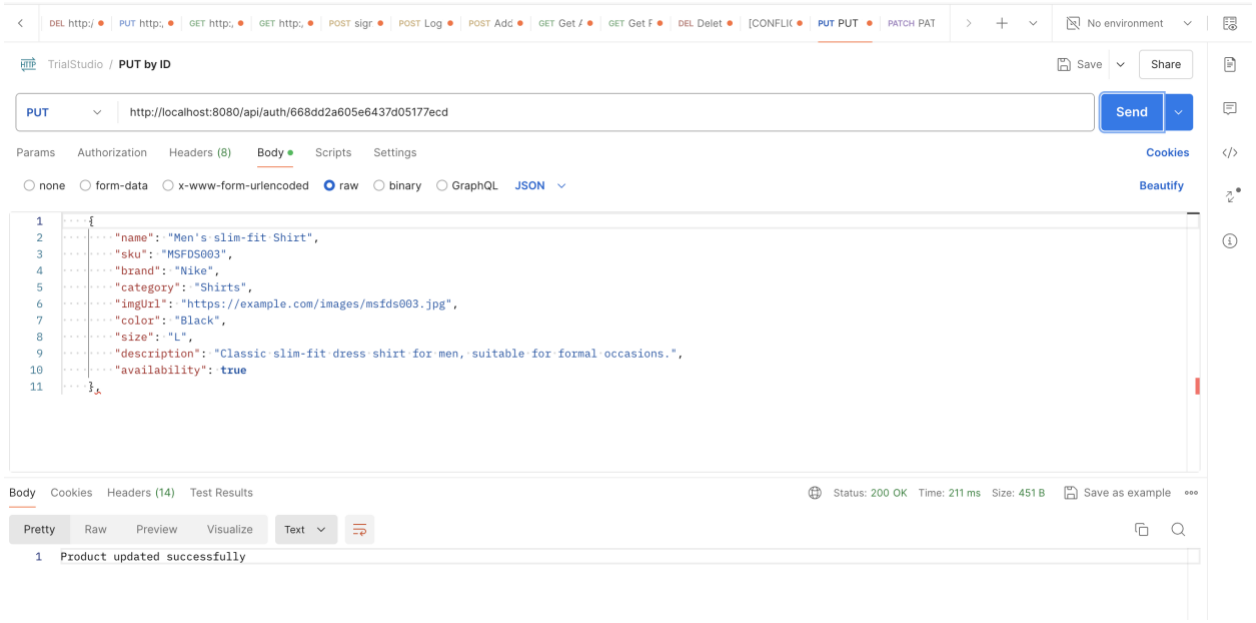

TrialStudio / Add Product POST
Save
Share
POST
http://localhost:8080/api/auth/addproduct
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name":"Kid's Shirt",
"sku":"MSFDS006",
"brand":"FirstCry",
"category":"Shirts",
"imgUrl":"https://example.com/images/msfds004.jpg",
"color":"Brown",
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 152 ms
Size: 449 B
Save as example
Pretty
Raw
Preview
Visualize
Text
Product added successfully

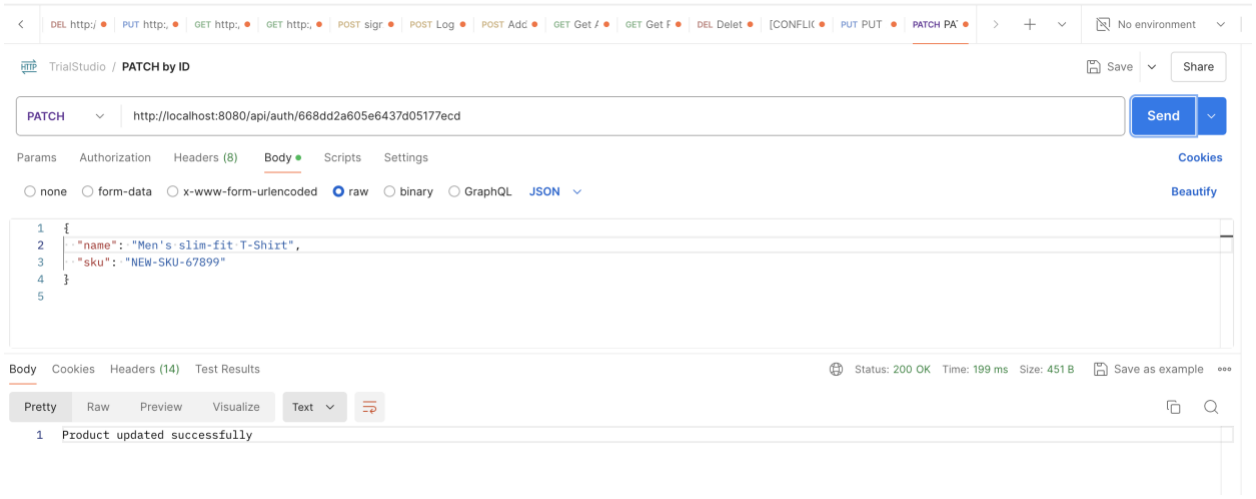

TrialStudio / Get All Products GET
Save
Share
GET
http://localhost:8080/api/auth/getallproducts
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 118 ms
Size: 1.5 KB
Save as example
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": "668c92f32d5e5f5a850b6e99",
"name": "Women's Top",
"sku": "NEW-SKU-67899",
"brand": "Levis",
"category": "Trousers",
"imgUrl": null,
"color": "Orange",
"size": "XL",
"description": "Updated product description",
"availability": true
},
{
"id": "668dd2a605e6437d05177ecd",
"name": "Men's Shirt",
"sku": "MSFDS003",
"brand": "Reebok",
"category": "Shirts",
"imgUrl": "https://example.com/images/msfds003.jpg",
"color": "Grey",
"size": "XL",
"description": "Classic slim-fit dress shirt for men, suitable for formal occasions.",
"availability": true
},
{
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash


TrialStudio / Get Product by ID GET
Save
Share
GET
http://localhost:8080/api/auth/getproductbyid?id=66921eb9d96b633672c5b250
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
id
66921eb9d96b633672c5b250
Key
Value
Description
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 107 ms
Size: 716 B
Save as example
Pretty
Raw
Preview
Visualize
JSON
{
"id": "66921eb9d96b633672c5b250",
"name": "Kid's Shirt",
"sku": "MSFDS006",
"brand": "FirstCry",
"category": "Shirts",
"imgUrl": "https://example.com/images/msfds004.jpg",
"color": "Brown",
"size": "S",
"description": "Classic slim-fit dress shirt for men, suitable for formal occasions.",
"availability": true
}


DEL http:/
PUT http:,
GET http:,
GET http:,
POST sigr
POST Log
POST Adc
GET Get /
GET Get F
DEL Delet
[CONFLIC
PUT PUT t
PATCH PAT
No environment
TrialStudio / Delete Product by ID DELETE
Save
Share
DELETE
http://localhost:8080/api/auth/deleteproductbyid?id=668c92f32d5e5f5a850b6e99
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
id
668c92f32d5e5f5a850b6e99
Key
Value
Description
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 121 ms
Size: 451 B
Save as example
Pretty
Raw
Preview
Visualize
Text
1 Product deleted successfully

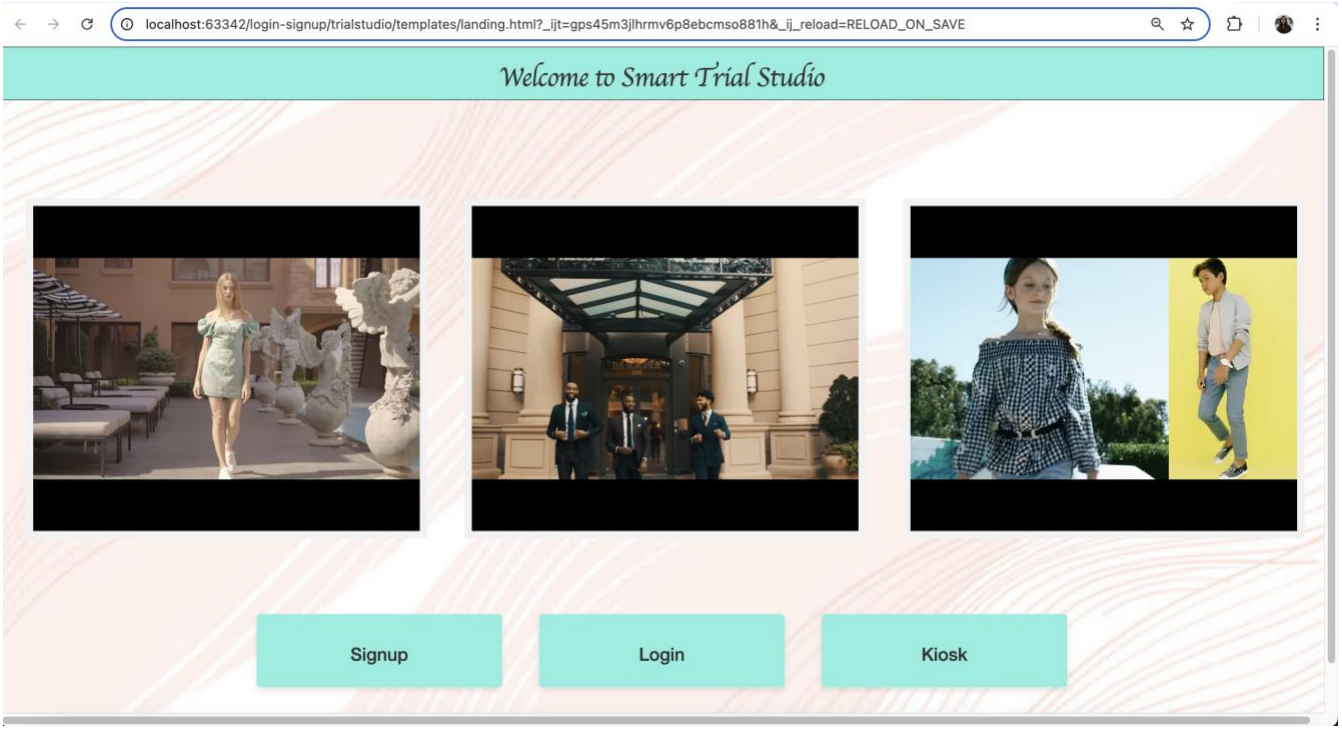

DEL http:/
PUT http:,
GET http:,
GET http:,
POST sigr
POST Log
POST Adc
GET Get /
GET Get F
DEL Delet
[CONFLI(
PUT PUT t
PATCH PAT
No environment
TrialStudio / Search Product by Keyword GET
Save
Share
GET
http://localhost:8080/api/auth/search?keyword=FirstCry
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
keyword
FirstCry
Key
Value
Description
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 93 ms
Size: 718 B
Save as example
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": "66921eb9d96b633672c5b250",
"name": "Kid's Shirt",
"sku": "MSFDS006",
"brand": "FirstCry",
"category": "Shirts",
"imgUrl": "https://example.com/images/msfds004.jpg",
"color": "Brown",
"size": "S",
"description": "Classic slim-fit dress shirt for men, suitable for formal occasions.",
"availability": true
}
]


TrialStudio / PUT by ID
Save
Share
PUT
http://localhost:8080/api/auth/668dd2a605e6437d05177ecd
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name": "Men's slim-fit Shirt",
"sku": "MSFDS003",
"brand": "Nike",
"category": "Shirts",
"imgUrl": "https://example.com/images/msfds003.jpg",
"color": "Black",
"size": "L",
"description": "Classic slim-fit dress shirt for men, suitable for formal occasions.",
"availability": true
}
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 211 ms
Size: 451 B
Save as example
Pretty
Raw
Preview
Visualize
Text
Product updated successfully

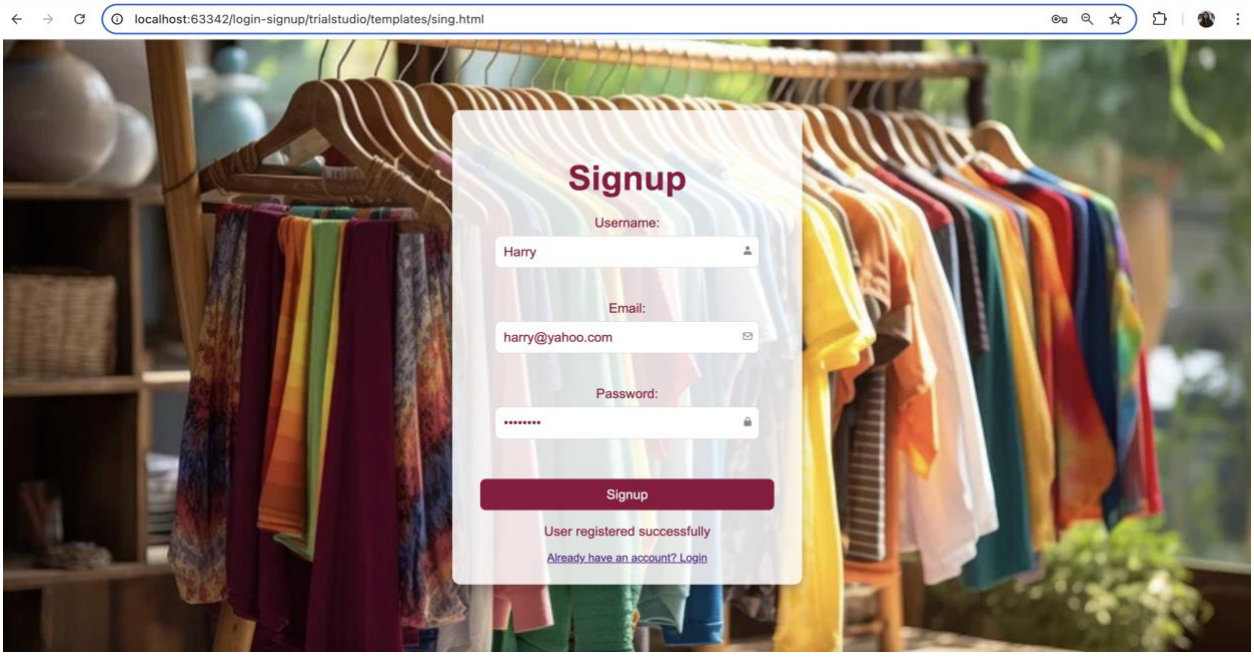

TrialStudio / PATCH by ID
Save
Share
PATCH
http://localhost:8080/api/auth/668dd2a605e6437d05177ecd
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name": "Men's slim-fit T-Shirt",
"sku": "NEW-SKU-67899"
}
Body
Cookies
Headers (14)
Test Results
Status: 200 OK
Time: 199 ms
Size: 451 B
Save as example
Pretty
Raw
Preview
Visualize
Text
Product updated successfully

13. Project Execution:

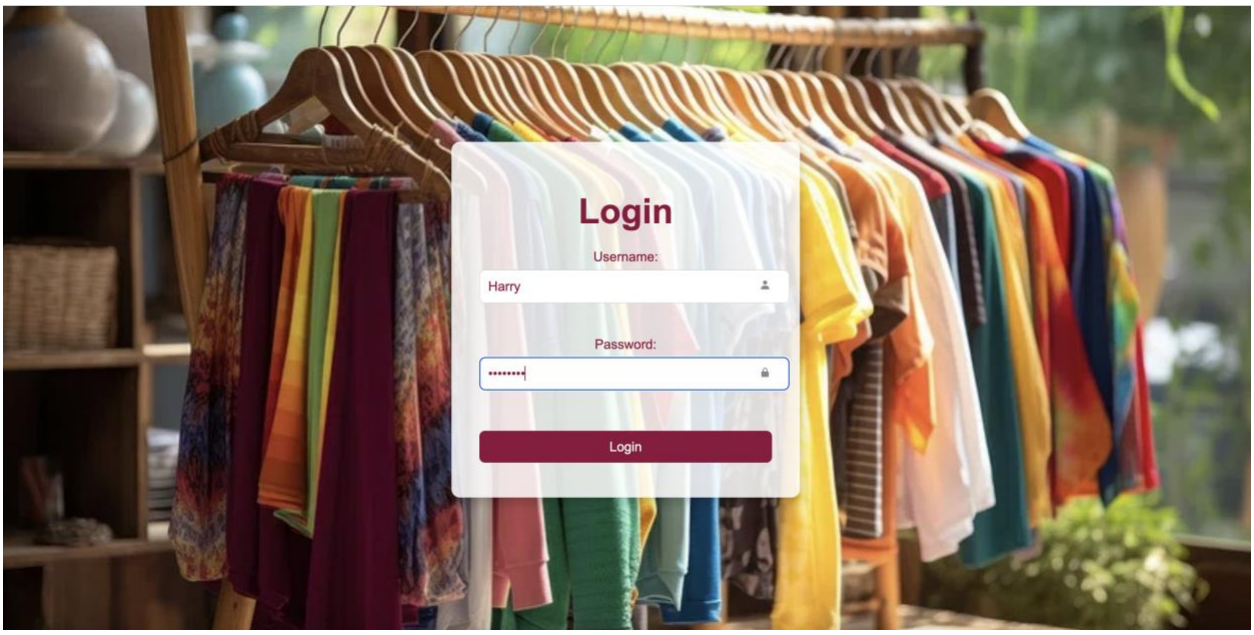
1. Landing Page:



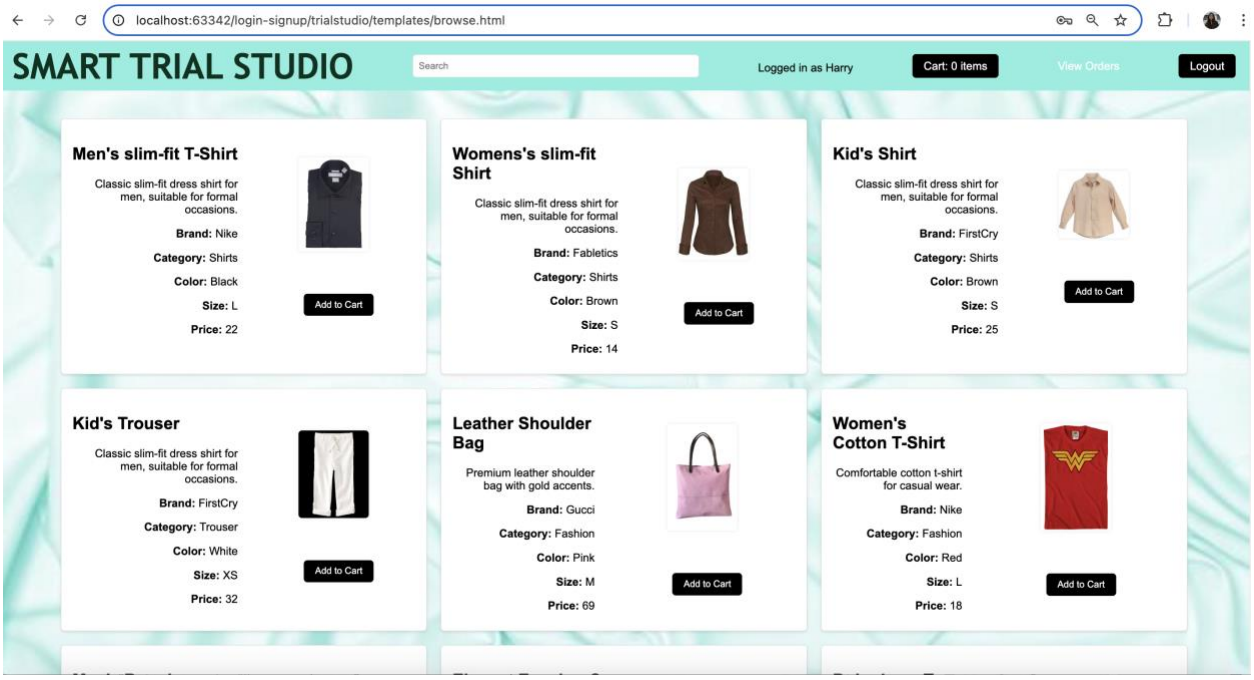
2. Sing Up page:

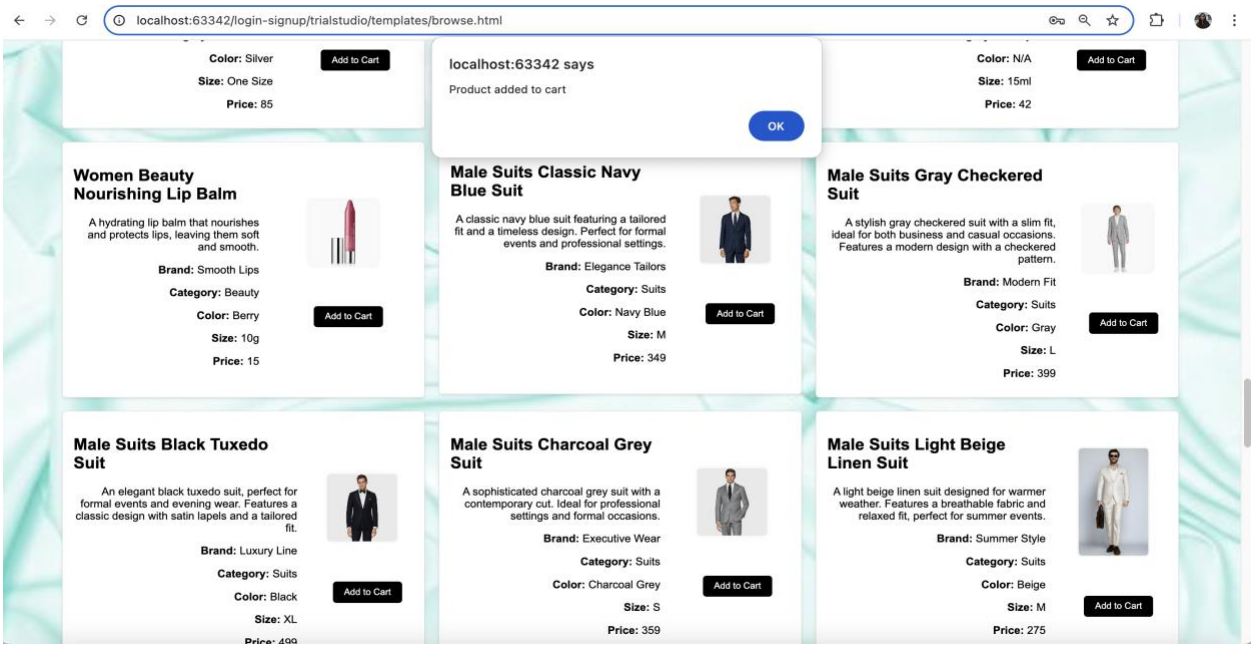


3. Login Page:

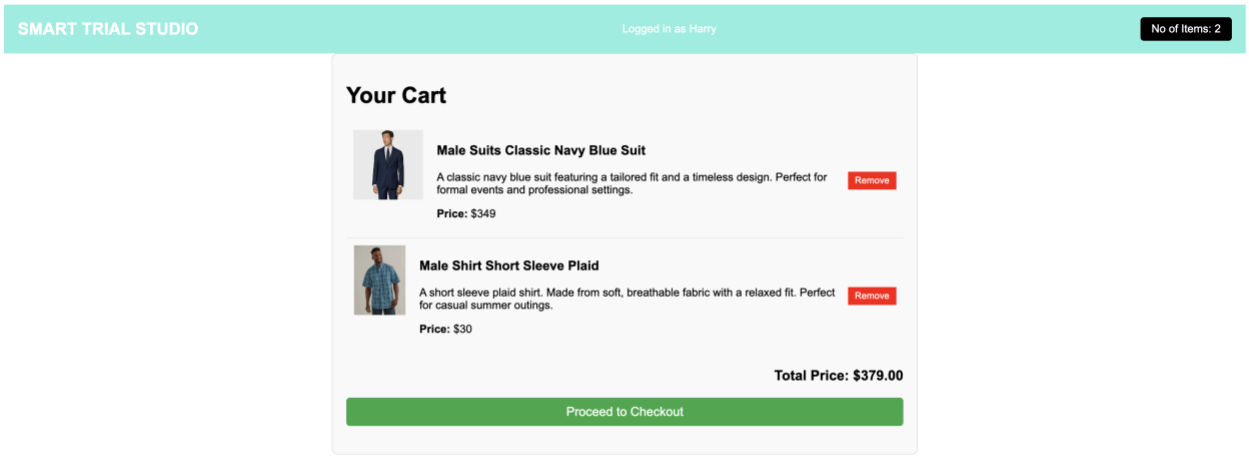


4. Browser page:

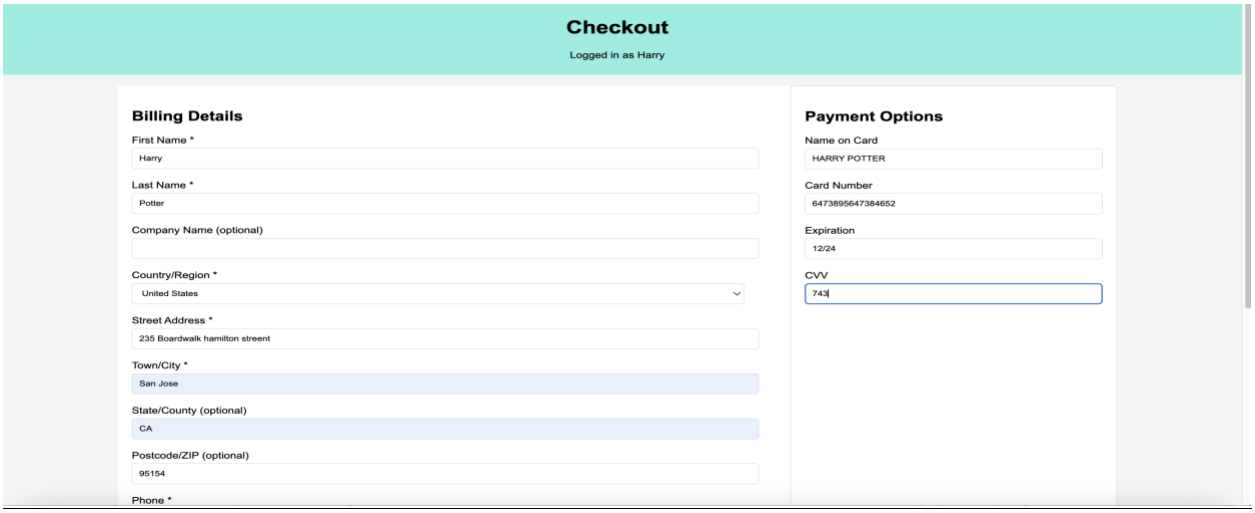


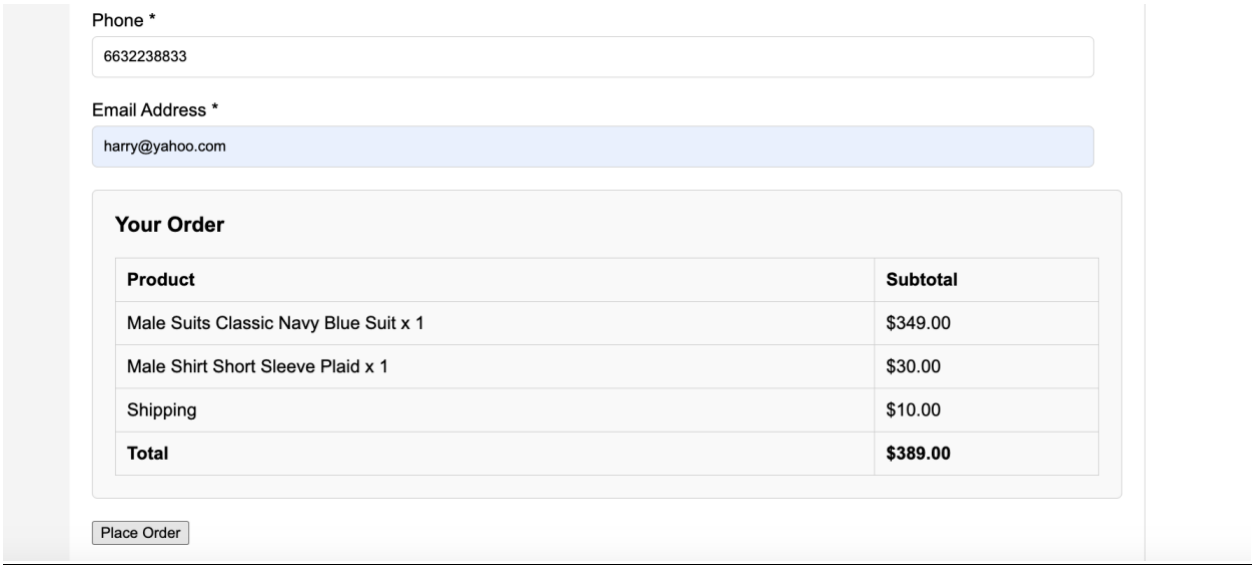

localhost:63342/login-signup/trialstudio/templates/browse.html
Color: Silver
Size: One Size
Price: 85
Add to Cart
localhost:63342 says
Product added to cart
OK
Color: N/A
Size: 15ml
Price: 42
Add to Cart
Women Beauty Nourishing Lip Balm
A hydrating lip balm that nourishes and protects lips, leaving them soft and smooth.
Brand: Smooth Lips
Category: Beauty
Color: Berry
Size: 10g
Price: 15
Add to Cart
Male Suits Classic Navy Blue Suit
A classic navy blue suit featuring a tailored fit and a timeless design. Perfect for formal events and professional settings.
Brand: Elegance Tailors
Category: Suits
Color: Navy Blue
Size: M
Price: 349
Add to Cart
Male Suits Gray Checkered Suit
A stylish gray checkered suit with a slim fit, ideal for both business and casual occasions. Features a modern design with a checkered pattern.
Brand: Modern Fit
Category: Suits
Color: Gray
Size: L
Price: 399
Add to Cart
Male Suits Black Tuxedo Suit
An elegant black tuxedo suit, perfect for formal events and evening wear. Features a classic design with satin lapels and a tailored fit.
Brand: Luxury Line
Category: Suits
Color: Black
Size: XL
Add to Cart
Male Suits Charcoal Grey Suit
A sophisticated charcoal grey suit with a contemporary cut. Ideal for professional settings and formal occasions.
Brand: Executive Wear
Category: Suits
Color: Charcoal Grey
Size: S
Price: 359
Add to Cart
Male Suits Light Beige Linen Suit
A light beige linen suit designed for warmer weather. Features a breathable fabric and relaxed fit, perfect for summer events.
Brand: Summer Style
Category: Suits
Color: Beige
Size: M
Price: 275
Add to Cart

5. Cart page

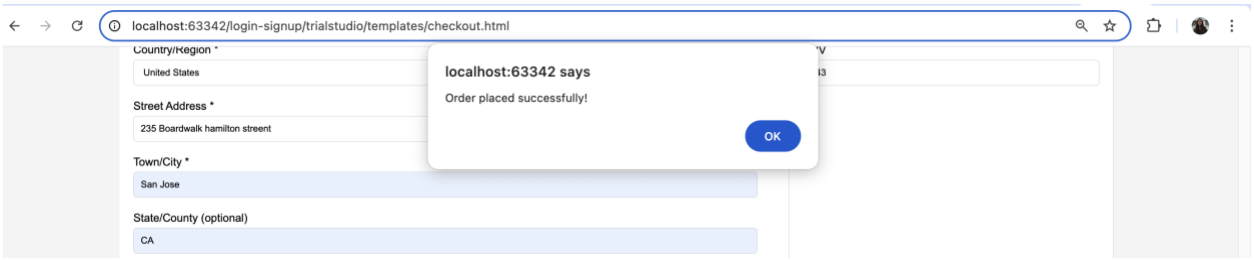

SMART TRIAL STUDIO
Logged in as Harry
No of Items: 2
Your Cart
Male Suits Classic Navy Blue Suit
A classic navy blue suit featuring a tailored fit and a timeless design. Perfect for formal events and professional settings.
Price: $349
Remove
Male Shirt Short Sleeve Plaid
A short sleeve plaid shirt. Made from soft, breathable fabric with a relaxed fit. Perfect for casual summer outings.
Price: $30
Remove
Total Price: $379.00
Proceed to Checkout

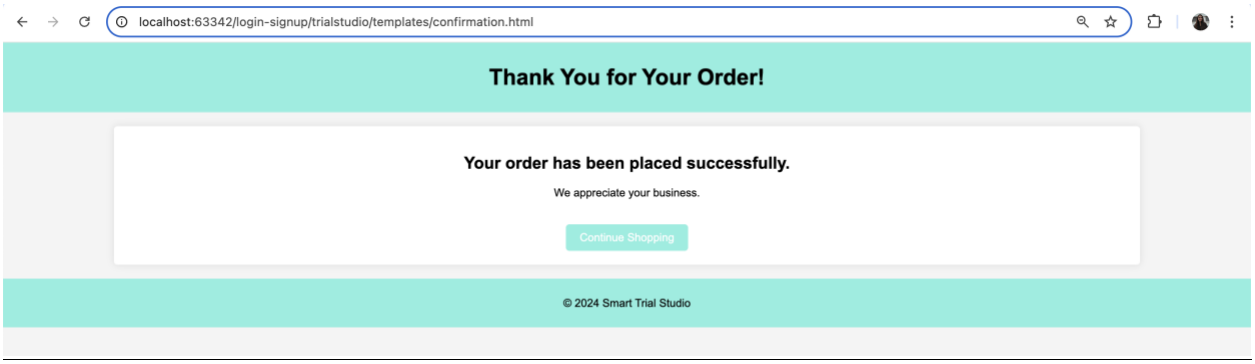
6. Checkout page:





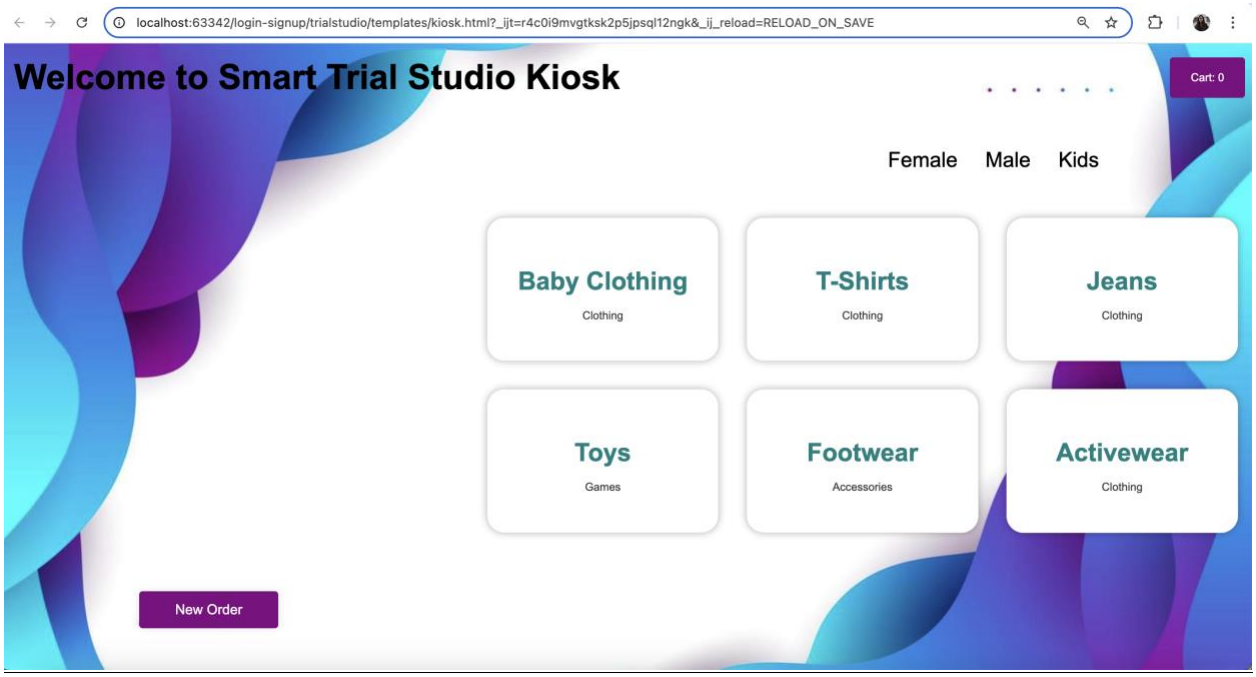
7. Order confirmation



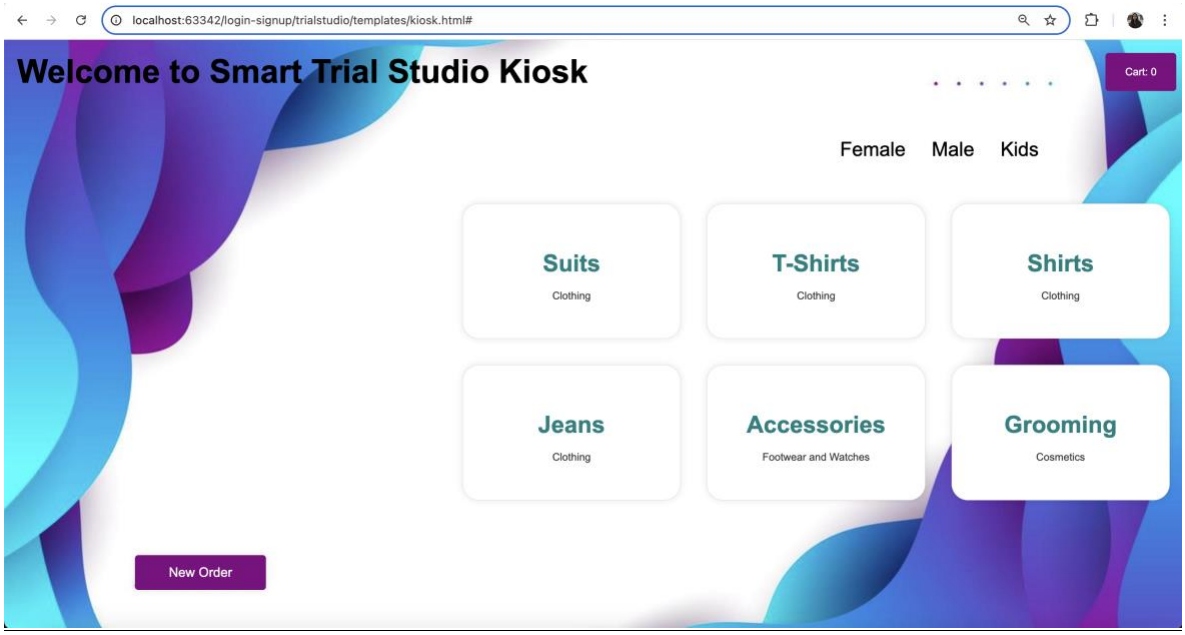


8. Kiosk Page

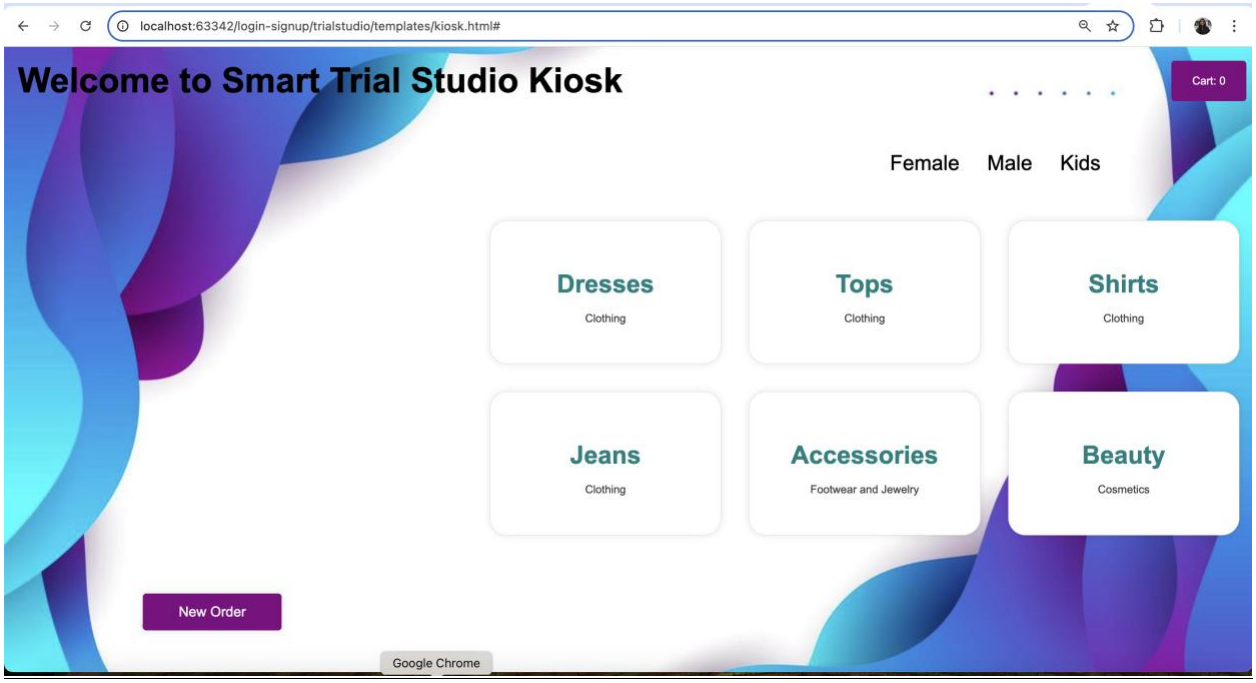
a) Kids:



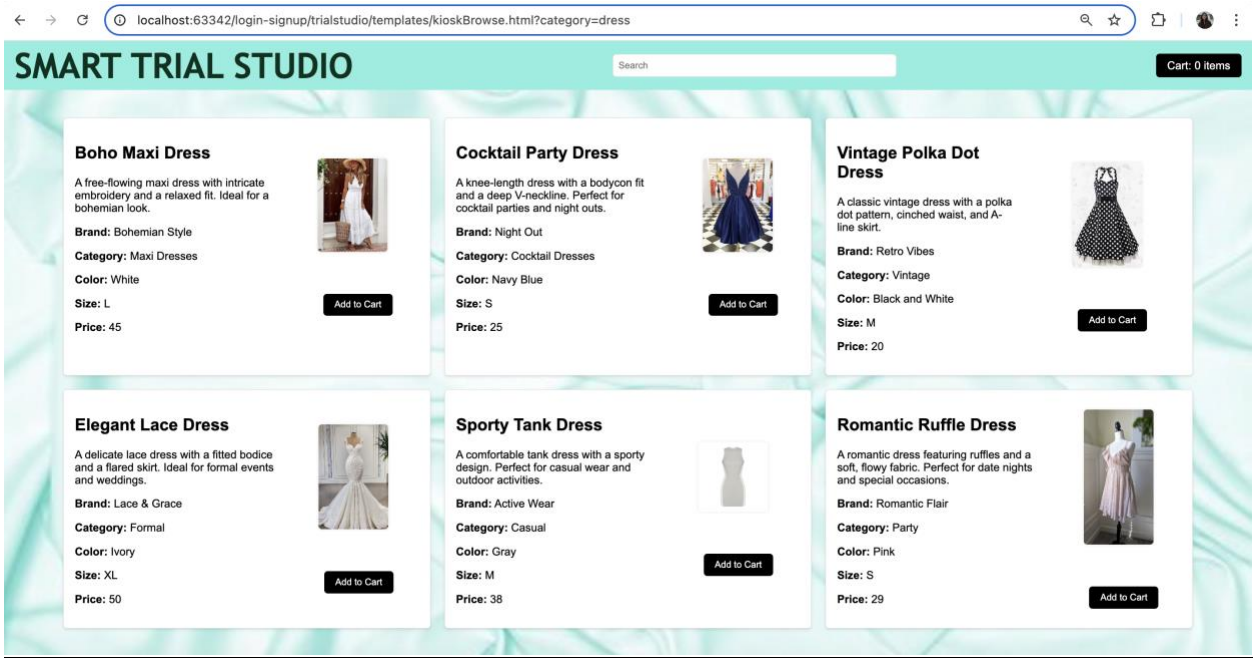
b) Male:



c) **Female:**



9. Browse products on kiosk:



localhost:63342/login-signup/trialstudio/templates/kioskBrowse.html?category=top

SMART TRIAL STUDIO

Search

Cart: 2 items

Boho Lace Top

A bohemian lace top with intricate detailing and a flowy silhouette, great for a relaxed, stylish look.

Brand: Bohemian Chic

Category: Blouses

Color: White

Size: L

Price: 45

Add to Cart

Classic woment top

A versatile teal top with a button-down front, made from lightweight, breathable fabric. Perfect for both professional and casual settings.

Brand: Elegant Essentials

Category: Top

Color: Teal

Size: M

Price: 35

Add to Cart

Casual Stripe Top

A casual top featuring bold stripes and a comfortable fit, perfect for everyday wear.

Brand: Urban Trend

Category: T-Shirts

Color: Sky Blue

Size: S

Price: 30

Add to Cart

Luxe Satin Top

A luxurious satin top with a sleek design and adjustable straps, ideal for elegant evening wear.

Brand: Elegance Wear

Category: Blouses

Color: Champagne

Size: M

Price: 45

Add to Cart

High Neck Top

A chic high-neck top made from soft cotton, ideal for layering or wearing alone for a sleek look.

Brand: Modern Classic

Category: T-Shirts

Color: Green

Size: XS

Price: 35

Add to Cart

Graphic Print Top

A trendy graphic print top with a bold design, perfect for adding a pop of style to casual outfits.

Brand: Trendsetter

Category: Graphic Tees

Color: Red

Size: M

Price: 28

Add to Cart

localhost:63342/login-signup/trialstudio/templates/kioskBrowse.html?category=grooming

SMART TRIAL STUDIO

localhost:63342 says

Product added to cart

OK

Cart: 2 items

Male Grooming Beard Oil

A nourishing beard oil designed to hydrate and condition your beard. Infused with natural oils and vitamins to promote healthy beard growth.

Brand: Beard Essentials

Category: Grooming

Color: Clear

Size: 30ml

Price: 18

Add to Cart

Cream

A rich and creamy shaving cream that provides a smooth shave. Enriched with moisturizers to protect and soothe the skin.

Brand: Smooth Shave

Category: Grooming

Color: White

Size: 100g

Price: 12

Add to Cart

Male Grooming Hair Styling Gel

A high-hold hair styling gel that keeps your hairstyle in place all day. Non-greasy and easy to wash out.

Brand: Urban Style

Category: Grooming

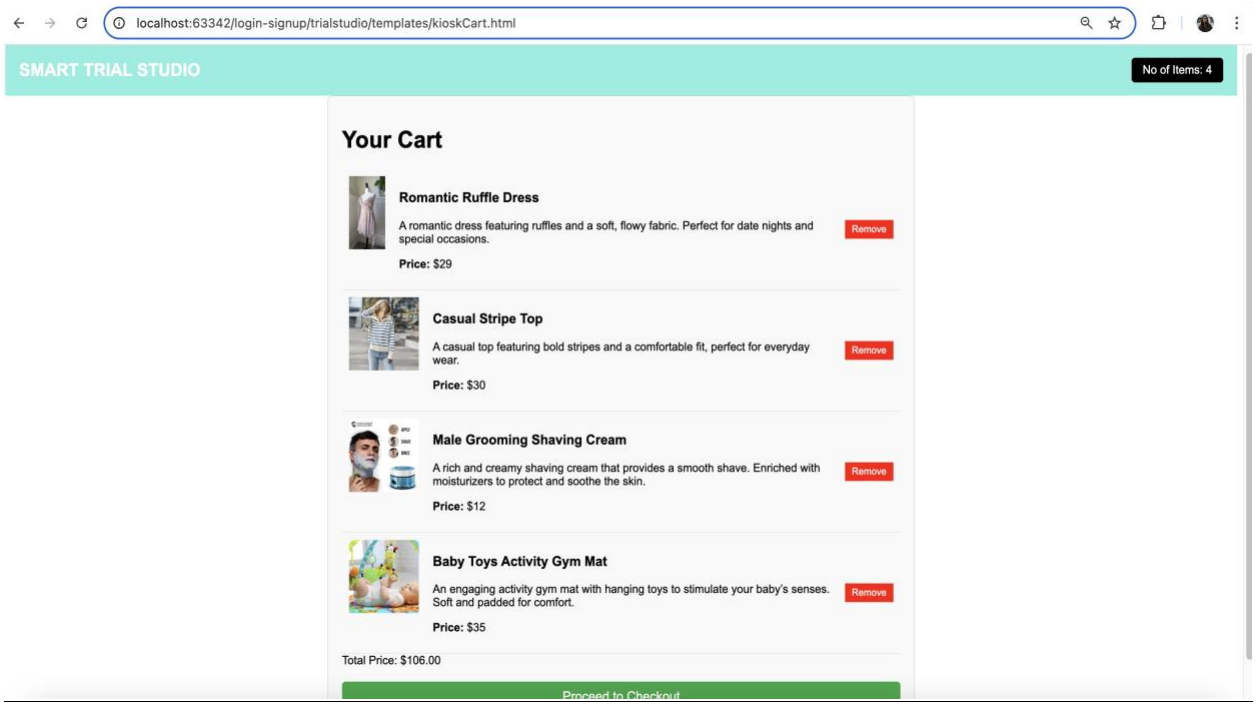
Color: Clear

Size: 150ml

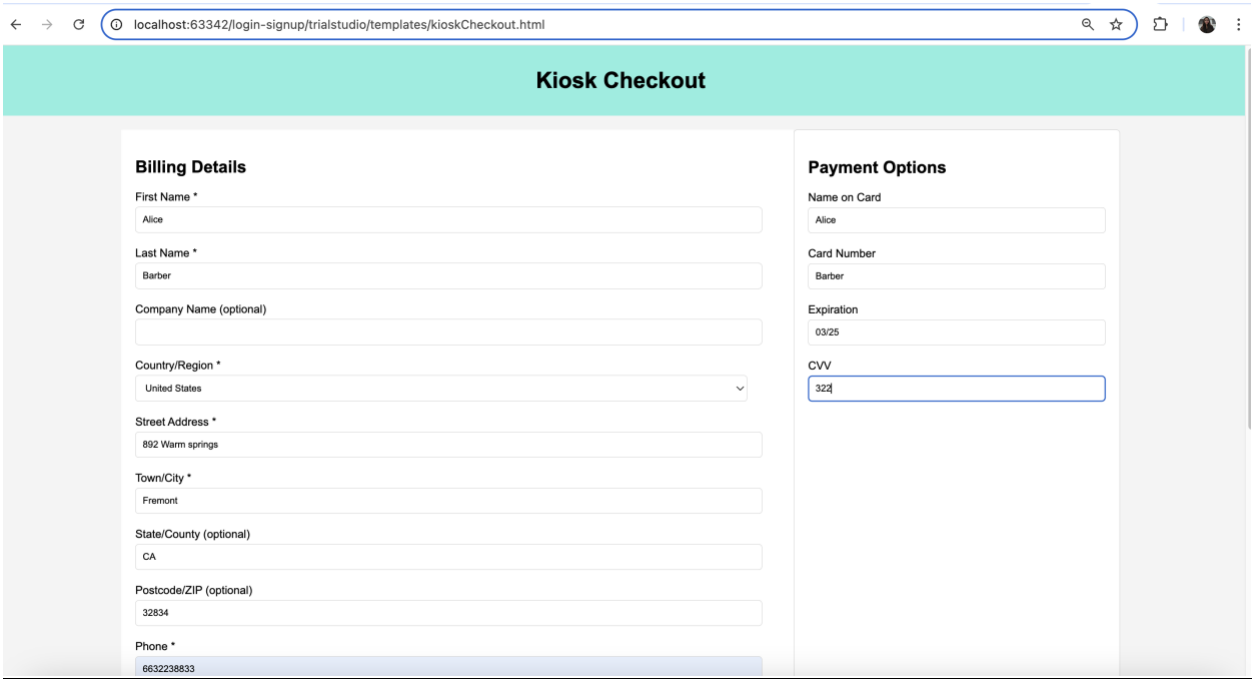
Price: 15

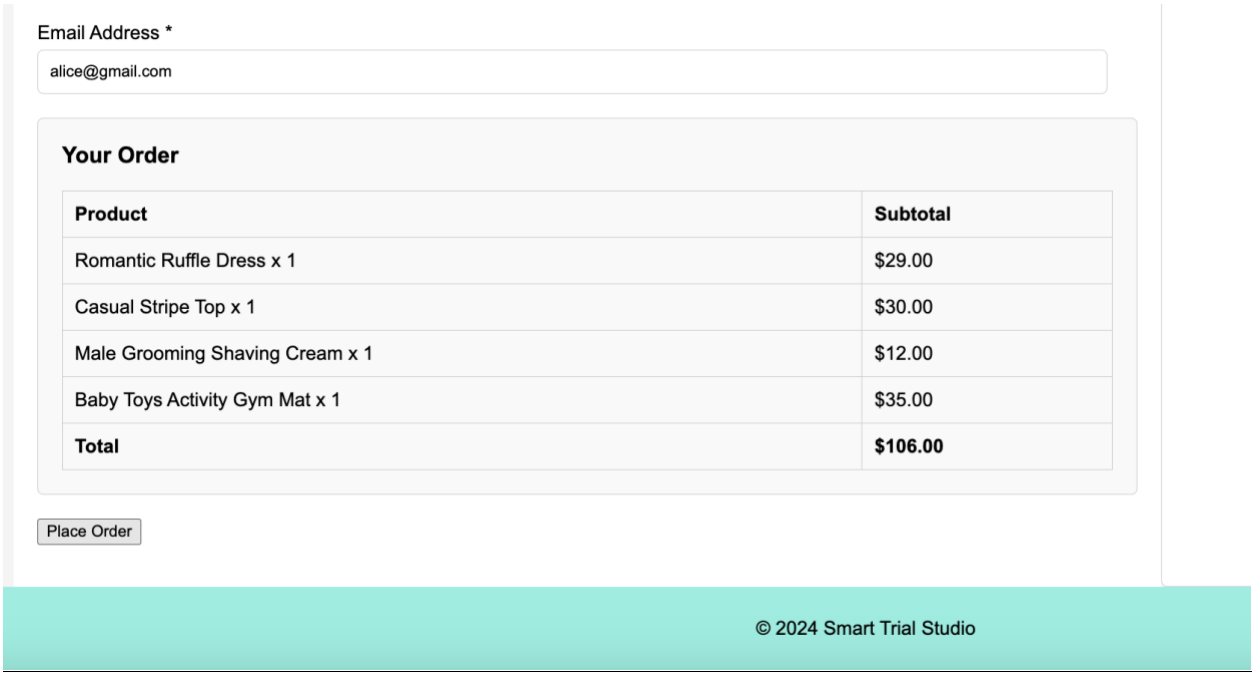
Add to Cart

10. Kiosk cart

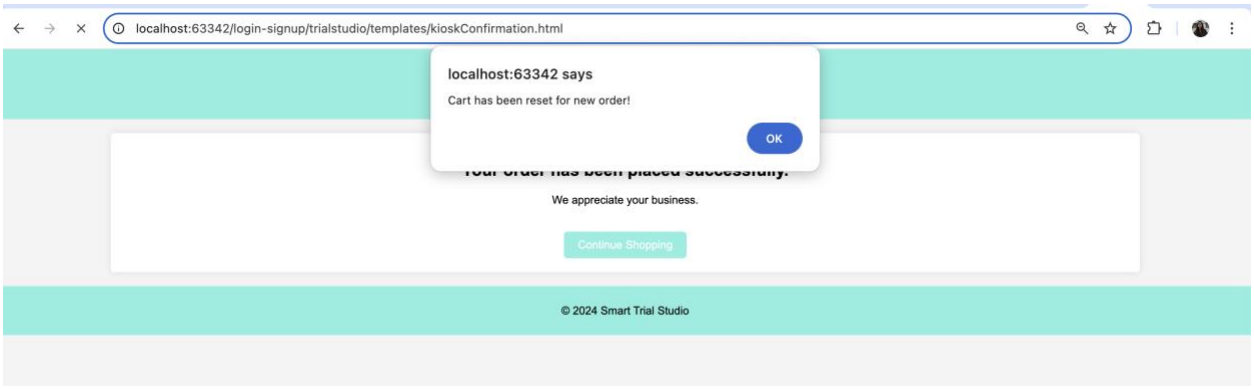


11. Kiosk Checkout





12. Order Confirmation:



14. Challenges:

A. **Mongodb connection:** I faced some issues when trying to connect to MongoDB, which is the database I used for the project. The connection kept failing due to problems like incorrect settings and outdated software versions. After some troubleshooting, I figured out the issue was with the network configuration and the database driver. Once I fixed these, the connection worked, and I could continue with the project.

B. **Security Configs:** I had trouble with the security settings when connecting to MongoDB. The database kept blocking my access because the security configurations were too strict. After adjusting the permissions and making sure my credentials were correct, I was finally able to connect and move forward with the project.

C. **API execution:** I ran into issues while creating the REST API for my project. The problem was that some of the endpoints weren't working correctly because of mistakes in how I set up the routes and handled requests. After debugging and fixing the code, the API started functioning as expected, and I could connect it to the rest of the application.

15. Completed Tasks:

Steps I followed to create the project:

1. Setting up MongoDB from https://www.mongodb.com/cloud/atlas/register
2. Created Spring boot project structure from https://start.spring.io/
3. Created API in AuthController.java
4. Used POSTMAN to test all the API's
5. Created web pages for Signup, Login, Browser and Cart, Checkout, Confirmation, Kiosk home page, kiosk browser page, Kiosk cart, Kiosk checkout, kiosk confirmation.
6. Added many functionalities to make the API and web pages execute smoothly.

PLEDGE:
I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else's work as my own.