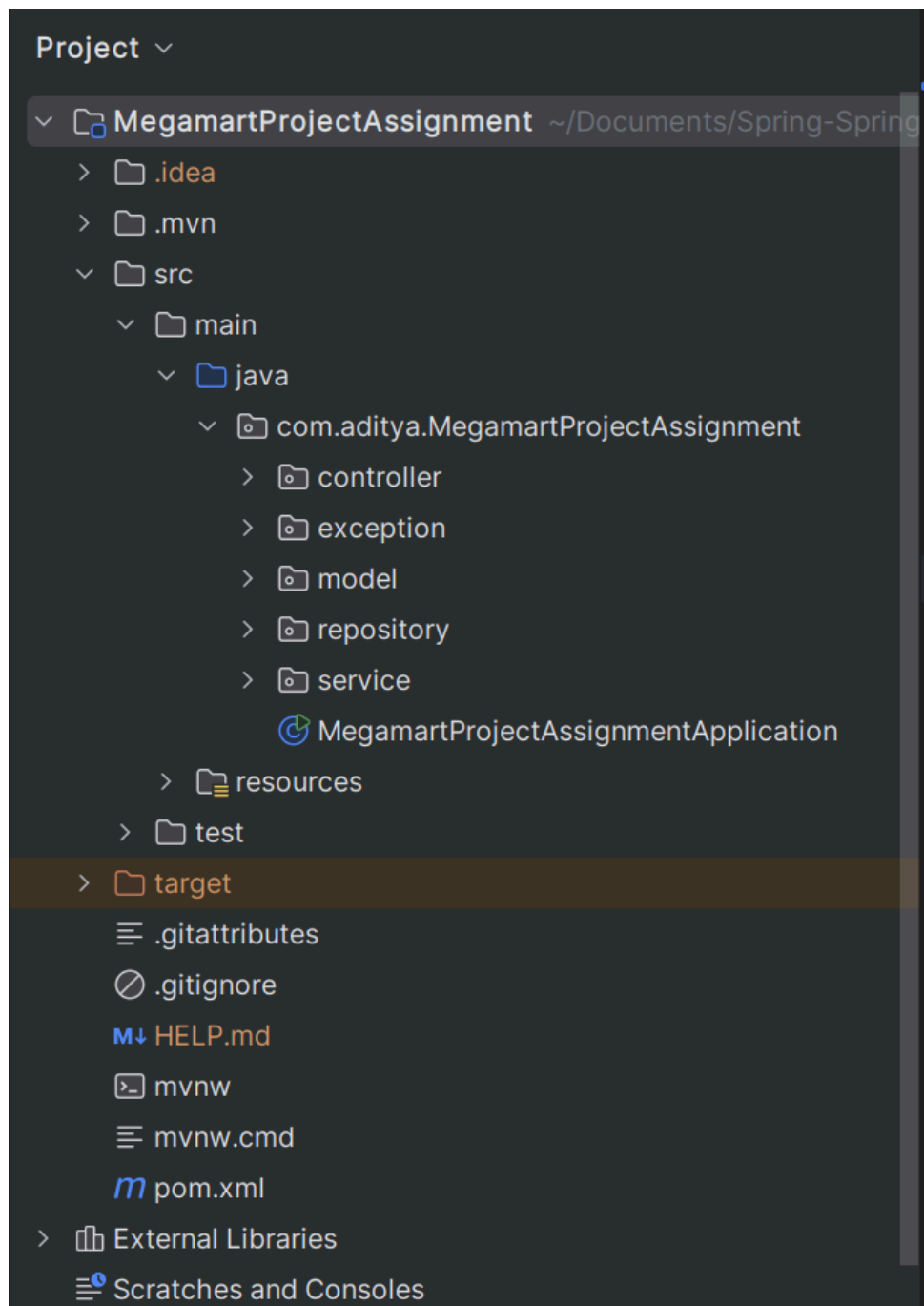


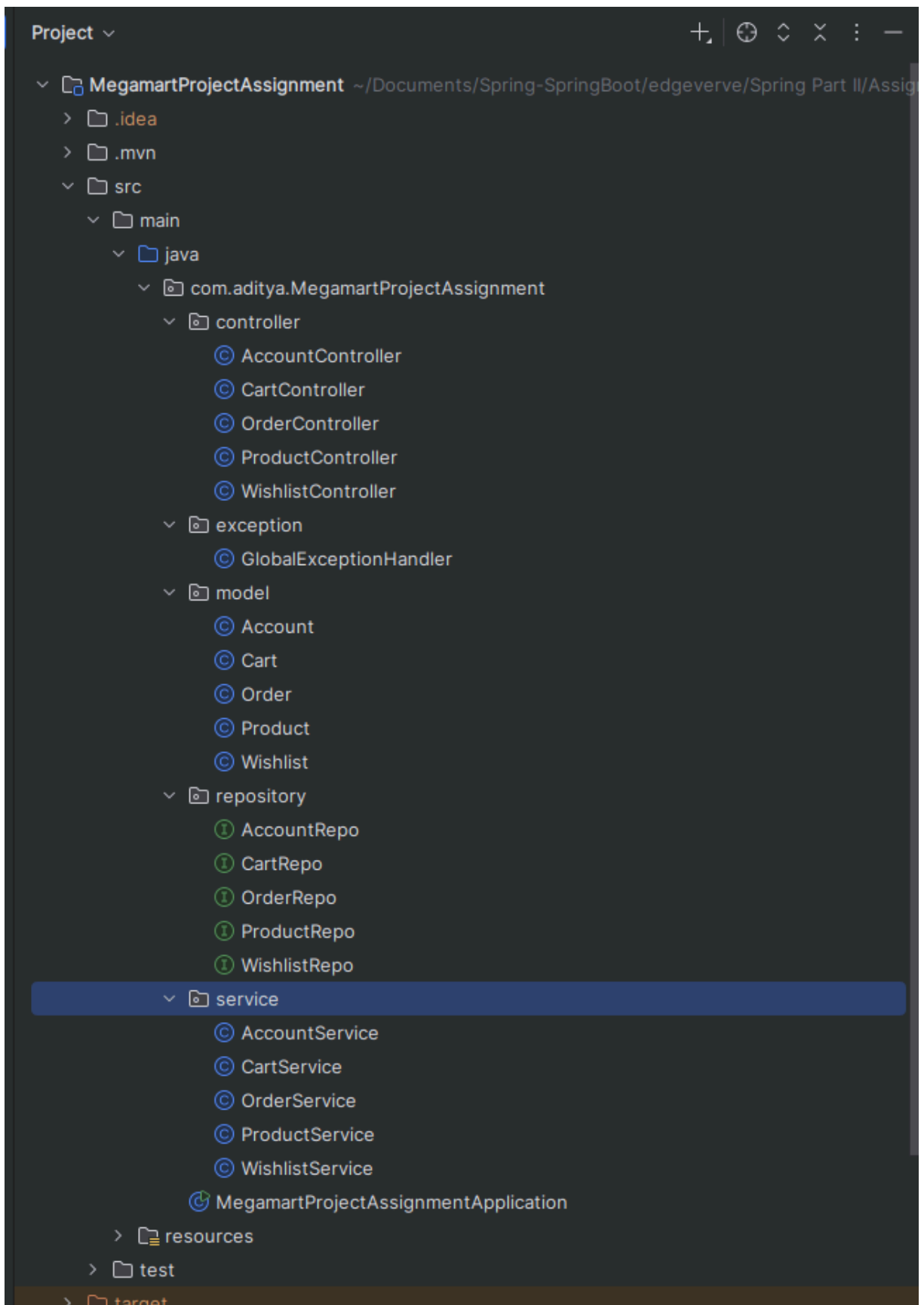
Assignment No. 3

Name : Aditya Pawar

USN: Aditya_72233061J

Folder Structure:





Application.properties

```
spring.application.name=MegamartProjectAssignment

spring.datasource.url=jdbc:postgresql://localhost:5432/megamartSpring
spring.datasource.username=postgres
spring.datasource.password=root
spring.datasource.driver-class-name=org.postgresql.Driver


spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.properties.hibernate.format_sql=true
```

MegamartProjectApplication.java

```
package com.aditya.MegamartProjectAssignment;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class MegamartProjectAssignmentApplication {

    public static void main(String[] args) {
        SpringApplication.run(MegamartProjectAssignmentApplication.class, args);
    }

}
```

Model Package

1. Account.java

```
package com.aditya.MegamartProjectAssignment.model;

import jakarta.persistence.*;
import jakarta.validation.constraints.Email;
import jakarta.validation.constraints.NotBlank;
import jakarta.validation.constraints.Size;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;

@AllArgsConstructor
@NoArgsConstructor
@ToString
@Data
@Entity
@Table(name = "account")
public class Account {
    @Id
    @NotBlank(message = "Username cannot be blank")
    private String name;

    @Email(message = "Invalid email format")
    @Column(nullable = false)
    @NotBlank(message = "Email cannot be blank")
    private String email;

    @NotBlank(message = "Password cannot be blank")
    @Size(min = 6, message = "Password must be at least 6 characters long")
```

```
private String password;

@Transient // not saved in DB, used during registration only
private String confirmPassword;
}
```

2. Cart.java

```
package com.aditya.MegamartProjectAssignment.model;

import jakarta.persistence.*;
import lombok.*;

@Entity
@Table(name = "cart")
public class Cart {
    @Id
    private String cartId;
    private String productName;
    private String sellerName;
    private int quantity;
    private Double cartOffer_price;
}
```

3. Order.java

```
package com.aditya.MegamartProjectAssignment.model;

import jakarta.persistence.*;
```

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;
```

```
import java.util.Date;
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
@Data
```

```
@ToString
```

```
@Entity
```

```
@Table(name = "orders")
```

```
public class Order {
```

```
    @Id
```

```
    private String orderId;
```

```
    private String displayName;
```

```
    private String category;
```

```
    private String seller_name;
```

```
    private Double price;
```

```
    private int quantity;
```

```
    private Double total_price;
```

```
    private Date ordered_date;
```

```
    private String order_status;
```

```
}
```

4. Product.java

```
package com.aditya.MegamartProjectAssignment.model;
```

```
import jakarta.persistence.*;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Data;
```

```

import lombok.NoArgsConstructor;
import lombok.ToString;

import java.math.BigInteger;

@AllArgsConstructor
@NoArgsConstructor
@Data
@Entity
@Table(name = "products")
public class Product {
    @Id
    private String productId;
    private String productName;
    private String description;
    private Double price;
    private Double discount;
    private Double delivery_charges;
    private float Avg_rating;
    private String seller_name;

    @Override
    public String toString() {
        return "\n{" +
            "\nproductId=" + productId +
            ", \nproductName='" + productName + '\'' +
            ", \ndescription='" + description + '\'' +
            ", \nprice=" + price +
            ", \ndiscount=" + discount +
            ", \ndelivery_charges=" + delivery_charges +
            ", \nAvg_rating=" + Avg_rating +
            ", \nseller_name='" + seller_name + '\'' + "\n" +
            '}' +
        '}';
    }
}

```


5. WishList.java

```
package com.aditya.MegamartProjectAssignment.model;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;

@Entity
@Table(name = "wishlist")
public class Wishlist {
    @Id
    private String wishId;
    private String displayName;
    private String shortDescription;
    private String category;
}
```

Repository Package

1. AccountRepo.java

```
package com.aditya.MegamartProjectAssignment.repository;

import com.aditya.MegamartProjectAssignment.model.Account;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
```

```
import java.util.Optional;

@Repository
public interface AccountRepo extends JpaRepository<Account, String> {
    Optional<Account> findByNameAndPassword(String name, String password);
    public Boolean existsByEmail(String email);
}
```

2. CartRepo.java

```
package com.aditya.MegamartProjectAssignment.repository;

import com.aditya.MegamartProjectAssignment.model.Cart;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface CartRepo extends JpaRepository<Cart, String> {
    List<Cart> findBySellerName(String sellerName);
    List<Cart> findByProductName(String productName);
}
```

3. OrderRepo

```
package com.aditya.MegamartProjectAssignment.repository;

import com.aditya.MegamartProjectAssignment.model.Order;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
```

```
public interface OrderRepo extends JpaRepository<Order, String> {  
}
```

4. ProductRepo.java

```
package com.aditya.MegamartProjectAssignment.repository;  
  
import com.aditya.MegamartProjectAssignment.model.Product;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Repository  
public interface ProductRepo extends JpaRepository<Product, String> {  
    List<Product> findByProductNameStartingWith(String prefix);  
}
```

5. WishlistRepo.java

```
package com.aditya.MegamartProjectAssignment.repository;  
  
import com.aditya.MegamartProjectAssignment.model.Wishlist;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface WishlistRepo extends JpaRepository<Wishlist, String> {  
}
```

Service Package

1. AccountService.java

```

package com.aditya.MegamartProjectAssignment.service;

import com.aditya.MegamartProjectAssignment.exception.GlobalExceptionHandl
import com.aditya.MegamartProjectAssignment.model.Account;
import com.aditya.MegamartProjectAssignment.repository.AccountRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class AccountService {

    @Autowired
    AccountRepo accountRepo;

    @Autowired
    GlobalExceptionHandler exceptionHandler;

    public String register(Account account){

        if (account.getName().isEmpty()) return "Provide Name";
        if (account.getEmail().isEmpty()) return "Provide Email";
        if (account.getPassword().isEmpty()) return "Provide Password";

        if(!account.getName().matches("^[a-zA-Z\\s]+$")) {
            return "Name can only contain letters and spaces";
        }

        if (!account.getEmail().matches("^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\. [a-
            return "Invalid Email Format";
        }

        if(accountRepo.existsByEmail(account.getEmail())){
            throw new IllegalArgumentException("Email already exists");
        }
    }
}

```

```

    }

    if(!account.getPassword().matches("^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)[A-Za-z0-9]{8,}$")) {
        return "Password must be at least 8 characters long, contain at least one
    }

    if(!account.getPassword().equals(account.getConfirmPassword())) {
        return "Password does not Match";
    }

    accountRepo.save(account);
    return "Account Registered Successfully...";
}

public String login(String name, String password){

    if (name.isEmpty()) return "Provide Name";
    if (password.isEmpty()) return "Provide Password";

    return accountRepo.findByNameAndPassword(name, password)
        .map(account → "Login successful")
        .orElse("Invalid username or password");

}

public List<Account> getAllAccounts() {
    return accountRepo.findAll();
}

public void deleteAccountByName(String name) {
    accountRepo.deleteById(name);
}
}

```

2. CartService.java

```
package com.aditya.MegamartProjectAssignment.service;

import com.aditya.MegamartProjectAssignment.model.Cart;
import com.aditya.MegamartProjectAssignment.repository.CartRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class CartService {

    @Autowired
    CartRepo cartRepo;

    public String addToCart(Cart cart){
        cartRepo.save(cart);
        return "Product Added to Cart Successfully!!!";
    }

    public List<Cart> getCartBySellerName(String sellerName){
        return cartRepo.findBySellerName(sellerName);
    }

    public List<Cart> getCartByProductName(String productName){
        return cartRepo.findByProductName(productName);
    }

    public List<Cart> getAllCartItems(){
        return cartRepo.findAll();
    }

    public Cart updateCartItem(String cartId, int quantity) {
        Cart cart = cartRepo.findById(cartId).orElseThrow(() ->
```

```

        new RuntimeException("Cart Id not found"));
        cart.setQuantity(quantity);
        cartRepo.save(cart);
        return cart;
    }

    public void removeFromCart(String cartId) {
        cartRepo.deleteById(cartId);
    }
}

```

3. OrderService.java

```

package com.aditya.MegamartProjectAssignment.service;

import com.aditya.MegamartProjectAssignment.model.Cart;
import com.aditya.MegamartProjectAssignment.model.Order;
import com.aditya.MegamartProjectAssignment.repository.CartRepo;
import com.aditya.MegamartProjectAssignment.repository.OrderRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.Date;
import java.util.List;
import java.util.Optional;

@Service
public class OrderService {

    @Autowired
    OrderRepo orderRepo;

    @Autowired
    CartRepo cartRepo;
}

```

```

public Order placeOrder(Order order){
    // checking if the cart items exist before placing the order
    Optional<Cart> optionalCart = cartRepo.findById(order.getOrderId());

    if (optionalCart.isPresent()) {
        order.setOrdered_date(new Date());
        order.setOrder_status("Pending");
        return orderRepo.save(order);
    }
    else {
        throw new IllegalArgumentException("Cart item not found for given
        Order ID: " + order.getOrderId());
    }
}

public List<Order> getAllOrders() {
    return orderRepo.findAll();
}

public boolean cancelOrder(String id) {
    Optional<Order> orderOptional = orderRepo.findById(id);
    if (orderOptional.isPresent()) {
        Order order = orderOptional.get();
        order.setOrder_status("Cancelled");
        orderRepo.save(order);
        return true;
    } else {
        return false; // Order not found
    }
}
}

```

4. ProductService.java


```

package com.aditya.MegamartProjectAssignment.service;

import com.aditya.MegamartProjectAssignment.model.Product;
import com.aditya.MegamartProjectAssignment.repository.ProductRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class ProductService {

    @Autowired
    ProductRepo productRepo;

    public Product addProduct(Product product){
        return productRepo.save(product);
    }

    public Product getProductById(String id){
        return productRepo.findById(id).orElseThrow(() → new RuntimeException
        ("Product Id not Found"));
    }

    public List<Product> getAllProducts(){
        return productRepo.findAll();
    }

    public List<Product> searchProducts(String prefix) {
        return productRepo.findByProductNameStartingWith(prefix);
    }
}

```

```
}
```

5. WishListService.java

```
package com.aditya.MegamartProjectAssignment.service;

import com.aditya.MegamartProjectAssignment.model.Wishlist;
import com.aditya.MegamartProjectAssignment.repository.WishlistRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.UUID;

@Service
public class WishlistService {

    @Autowired
    WishlistRepo wishlistRepo;

    public String addToWishList(Wishlist wishlist){
        try {
            wishlistRepo.save(wishlist);
        }
        catch (Exception e) {
            return "Error while adding product to WishList: " + e.getMessage();
        }

        return "Product Added to WishList Successfully!!!";
    }

    public String removeFromWishlist(String wishId){
        try {
            wishlistRepo.deleteById(wishId);
        }
    }
}
```

```

    }
    catch (Exception e) {
        return "Error while adding product to WishList: " + e.getMessage();
    }

    return "Product removed from WishList Successfully!!!";
}

public List<Wishlist> getAllWishListItems() {
    return wishlistRepo.findAll();
}
}

```

Controller Package

1. AccountController.java

```

package com.aditya.MegamartProjectAssignment.controller;

import com.aditya.MegamartProjectAssignment.model.Account;
import com.aditya.MegamartProjectAssignment.repository.AccountRepo;
import com.aditya.MegamartProjectAssignment.service.AccountService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;

@RestController
@RequestMapping("/api/account")
@CrossOrigin(origins = "http://localhost:3000")
public class AccountController {

```

```
@Autowired
AccountService accountService;
```

```
@Autowired
AccountRepo accountRepo;
```

```
@PostMapping("/register")
public ResponseEntity<String> register(@RequestBody Account account){
    String result = accountService.register(account);
    return ResponseEntity.ok(result);
}
```

```
@PostMapping("/login")
public ResponseEntity<String> login(@RequestParam String name,
                                   @RequestParam String password) {
    String result = accountService.login(name, password);
    return ResponseEntity.ok(result);
}
```

```
@GetMapping("/get")
public ResponseEntity<List<Account>> getAllAccounts(){
    List<Account> accounts = accountService.getAllAccounts();
    if (accounts.isEmpty()){
        return ResponseEntity.status(HttpStatus.NOT_FOUND).body(null);
    }
    else {
        return ResponseEntity.ok(accounts);
    }
}
```

```
@DeleteMapping("/delete")
public ResponseEntity<String> deleteAccount(@RequestParam String name){
    if (!accountRepo.existsById(name)){
        return ResponseEntity.status(HttpStatus.NOT_FOUND)
            .body("Account not found");
    }
}
```

```

    }
    accountService.deleteAccountByName(name);
    return ResponseEntity.ok("Account deleted Successfully");
}

}

```

2. CartController.java

```

package com.aditya.MegamartProjectAssignment.controller;

import com.aditya.MegamartProjectAssignment.model.Cart;
import com.aditya.MegamartProjectAssignment.service.CartService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/cart")
public class CartController {

    @Autowired
    CartService cartService;

    @PostMapping("/addToCart")
    public ResponseEntity<String> addToCart(@RequestBody Cart cart){
        String result = cartService.addToCart(cart);
        return ResponseEntity.ok(result + "\n" + cart.toString());
    }

    @GetMapping("/all")
    public ResponseEntity<List<Cart>> viewAllCartItems(){

```

```

        return ResponseEntity.ok(cartService.getAllCartItems());
    }

    @GetMapping("/product")
    public ResponseEntity<List<Cart>> viewByProduct(
        @RequestParam String productName) {
        return ResponseEntity.ok(cartService.getCartByProductName(productName));
    }

    @GetMapping("/seller")
    public ResponseEntity<List<Cart>> viewBySeller(
        @RequestParam String sellerName) {
        return ResponseEntity.ok(cartService.getCartBySellerName(sellerName));
    }

    @PutMapping("/{cartId}")
    public ResponseEntity<Cart> updateCartItem(@PathVariable String cartId,
        @RequestParam int quantity) {
        Cart updatedCart = cartService.updateCartItem(cartId, quantity);
        return ResponseEntity.ok(updatedCart);
    }

    @DeleteMapping("/{cartId}")
    public ResponseEntity<String> removeFromCart(@PathVariable String cartId) {
        cartService.removeFromCart(cartId);
        return ResponseEntity.ok("Deleted successfully");
    }
}

```

3. OrderController.java

```

package com.aditya.MegamartProjectAssignment.controller;

import com.aditya.MegamartProjectAssignment.model.Order;

```

```

import com.aditya.MegamartProjectAssignment.service.OrderService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/order")
public class OrderController {

    @Autowired
    OrderService orderService;

    @PostMapping("/placeOrder")
    public ResponseEntity<String> placeOrder(@RequestBody Order order){
        orderService.placeOrder(order);
        return ResponseEntity.ok("Order Placed Successfully...\n\n"+
            order.toString());
    }

    @GetMapping("/getAll")
    public ResponseEntity<List<Order>> getAllOrders() {
        List<Order> orders = orderService.getAllOrders();
        return ResponseEntity.ok(orders);
    }

    @DeleteMapping("/cancelOrder/{id}")
    public ResponseEntity<String> cancelOrder(@PathVariable String id){
        boolean isCancelled = orderService.cancelOrder(id);
        if (isCancelled) {
            return ResponseEntity.ok("Order with ID " + id +
                " has been cancelled successfully.");
        } else {
            return ResponseEntity.status(404).body("Order with ID " + id +
                " not found.");
        }
    }
}

```

```
    }  
  }  
  
}
```

4. ProductController.java

```
package com.aditya.MegamartProjectAssignment.controller;  
  
import com.aditya.MegamartProjectAssignment.model.Product;  
import com.aditya.MegamartProjectAssignment.repository.ProductRepo;  
import com.aditya.MegamartProjectAssignment.service.ProductService;  
import jakarta.persistence.GeneratedValue;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/api/product")  
public class ProductController {  
  
    @Autowired  
    ProductService productService;  
  
    @Autowired  
    ProductRepo productRepo;  
  
    @PostMapping("/add")  
    public String addProduct(@RequestBody Product product){  
        productService.addProduct(product);  
        return "Product Added Successfully...\n"+product.toString();  
    }  
}
```



```

    }

    @GetMapping("/search")
    public ResponseEntity<List<Product>> searchProducts(
        @RequestParam String prefix) {
        return ResponseEntity.ok(productService.searchProducts(prefix));
    }

    @GetMapping("/{id}")
    public Product getProduct(@PathVariable("id") String id){
        return productService.getProductById(id);
    }

    @GetMapping("/all")
    public List<Product> getAllProducts(){
        return productService.getAllProducts();
    }
}

```

5. WishlistController.java

```

package com.aditya.MegamartProjectAssignment.controller;

import com.aditya.MegamartProjectAssignment.model.Wishlist;
import com.aditya.MegamartProjectAssignment.service.WishlistService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/wishlist")

```

```

public class WishlistController {

    @Autowired
    WishlistService service;

    @PostMapping("/add")
    public ResponseEntity<String> addToWishlist(@RequestBody Wishlist wishlist) {
        String result = service.addToWishList(wishlist);
        return ResponseEntity.ok(result);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<String> deleteFromWishList(
        @PathVariable("id") String id){
        service.removeFromWishlist(id);
        return ResponseEntity.ok("WishList Item with id "+id+
            " is deleted Successfully!!!");
    }

    @GetMapping("/all")
    public ResponseEntity<List<Wishlist>> getWishlistItems() {
        return ResponseEntity.ok(service.getAllWishListItems());
    }
}

```

Exception Package

GlobalExceptionHandler.java

```

package com.aditya.MegamartProjectAssignment.exception;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;

```

```

import org.springframework.web.bind.annotation.ExceptionHandler;

@ControllerAdvice
public class GlobalExceptionHandler {

    @ExceptionHandler(IllegalArgumentException.class)
    public ResponseEntity<String> handleIllegalArgumentException
    (IllegalArgumentException ex){
        return new ResponseEntity<>("Invalid input → "+
        ex.getMessage(), HttpStatus.BAD_REQUEST);
    }

    @ExceptionHandler(NullPointerException.class)
    public ResponseEntity<String> handleNullPointerException
    (NullPointerException ex) {
        return new ResponseEntity<>("Something went wrong (null value).", HttpStatus.INTERNAL_SERVER_ERROR);
    }

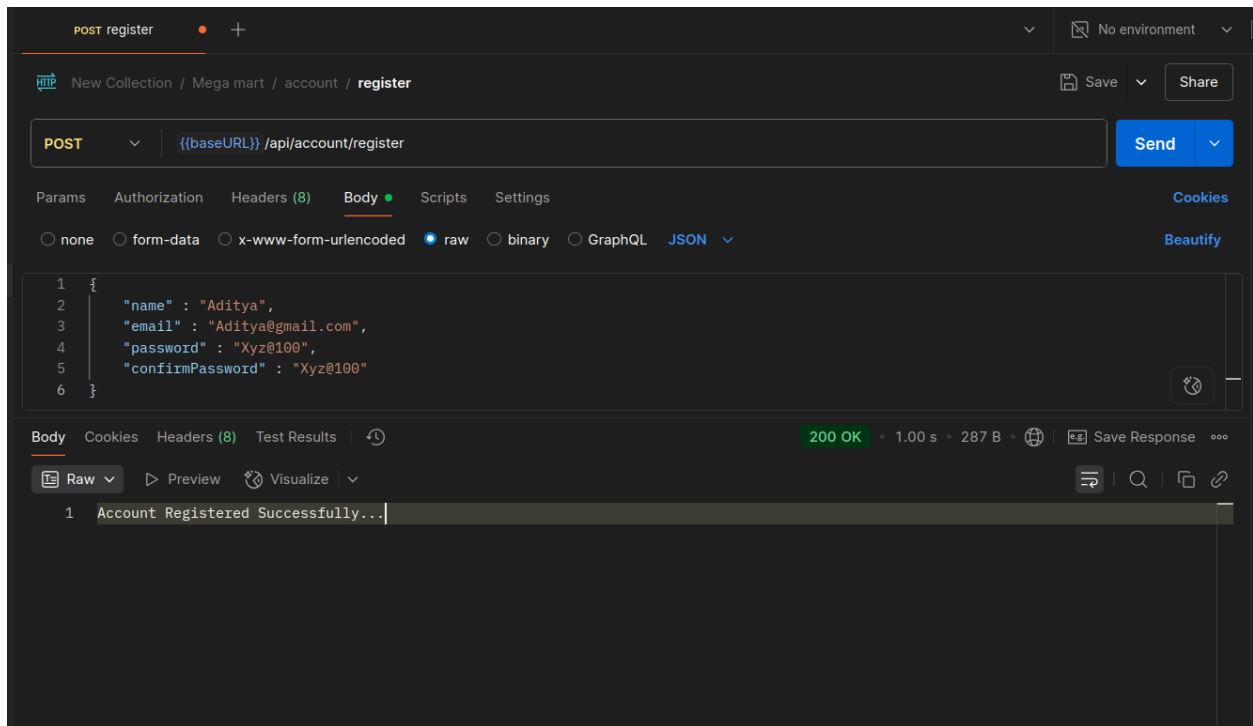
    // Handle any other generic exception
    @ExceptionHandler(Exception.class)
    public ResponseEntity<String> handleException(Exception ex) {
        return new ResponseEntity<>("Unexpected Error: " + ex.getMessage(), HttpStatus.INTERNAL_SERVER_ERROR);
    }
}

```

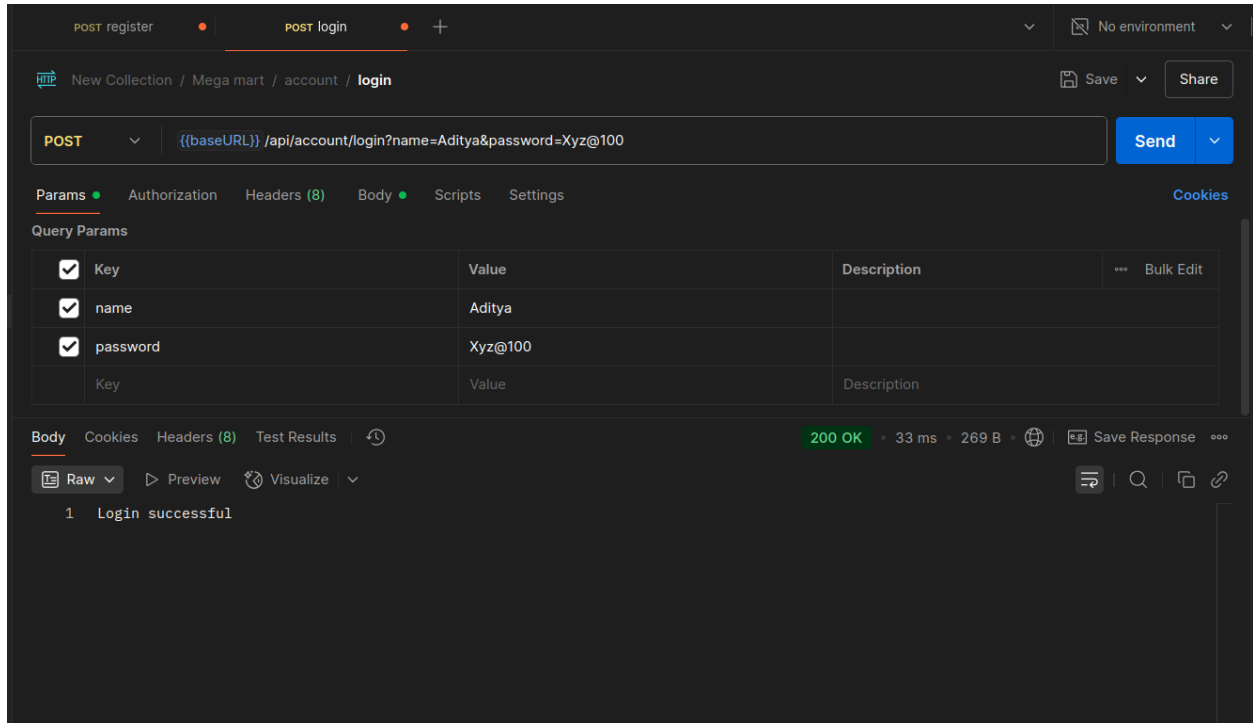
Output:

A. Account —>

1. Register



2. Login



Validations:

1. incorrect name

```
1 {
2   "name" : "Aditya~/[]",
3   "email" : "Aditya@gmail.com",
4   "password" : "Xyz@100",
5   "confirmPassword" : "Xyz@100"
6 }
```

Body Cookies Headers (8) Test Results

Raw Preview Visualize

1 Name can only contain letters and spaces

2. Wrong confirm password

```
1 {
2   "name" : "Aditya",
3   "email" : "Aditya@gmail.com",
4   "password" : "Xyz@100",
5   "confirmPassword" : "Xyz__100"
6 }
```

Body Cookies Headers (8) Test Results

Raw Preview Visualize

1 Password does not Match

3. Invalid username

Params Authorization Headers (8) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	name	Adityaaaa
<input checked="" type="checkbox"/>	password	Xyz@100
	Key	Value

Body Cookies Headers (8) Test Results

Raw Preview Visualize

1 Invalid username or password

4. Invalid Password

Params Authorization Headers (8) Body Scripts Settings

Query Params

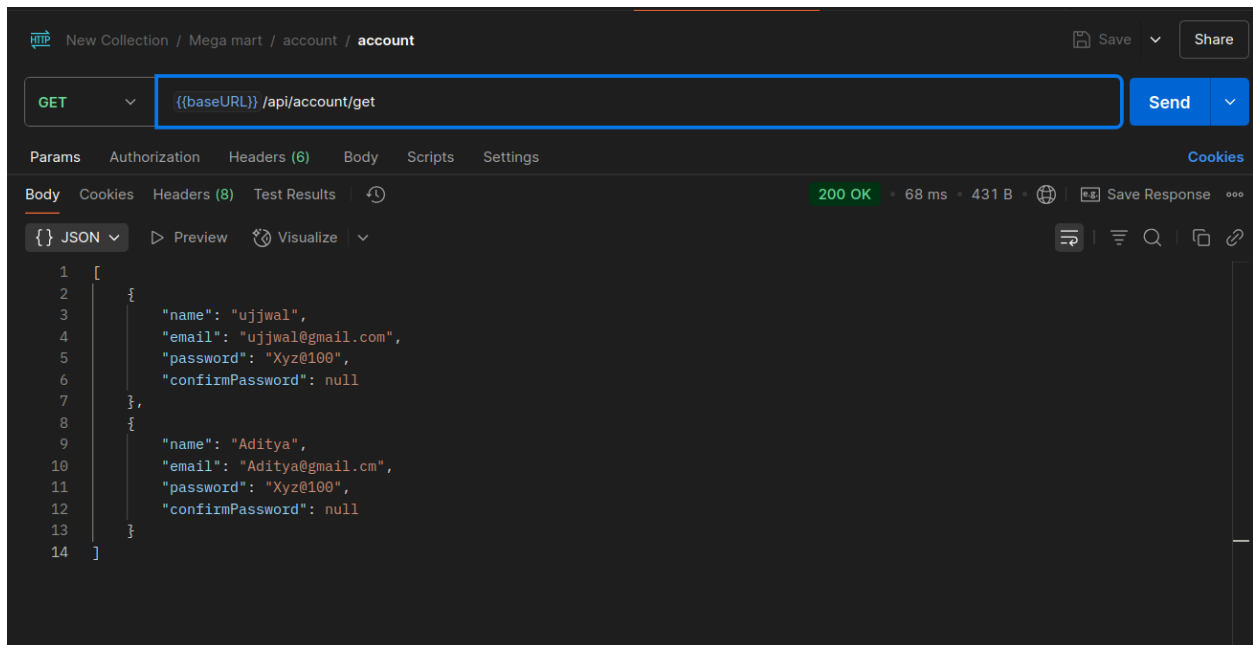
<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	name	Aditya
<input checked="" type="checkbox"/>	password	Xyz@100@@
	Key	Value

Body Cookies Headers (8) Test Results

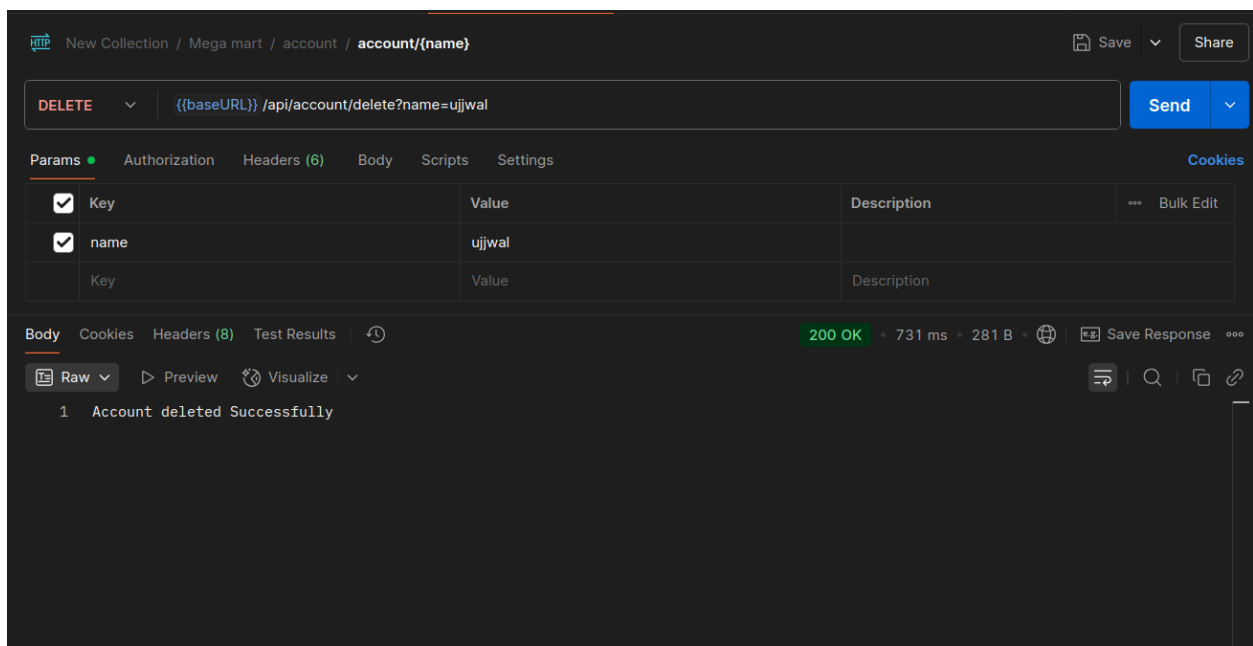
Raw Preview Visualize

1 Invalid username or password

3. Get All Accounts

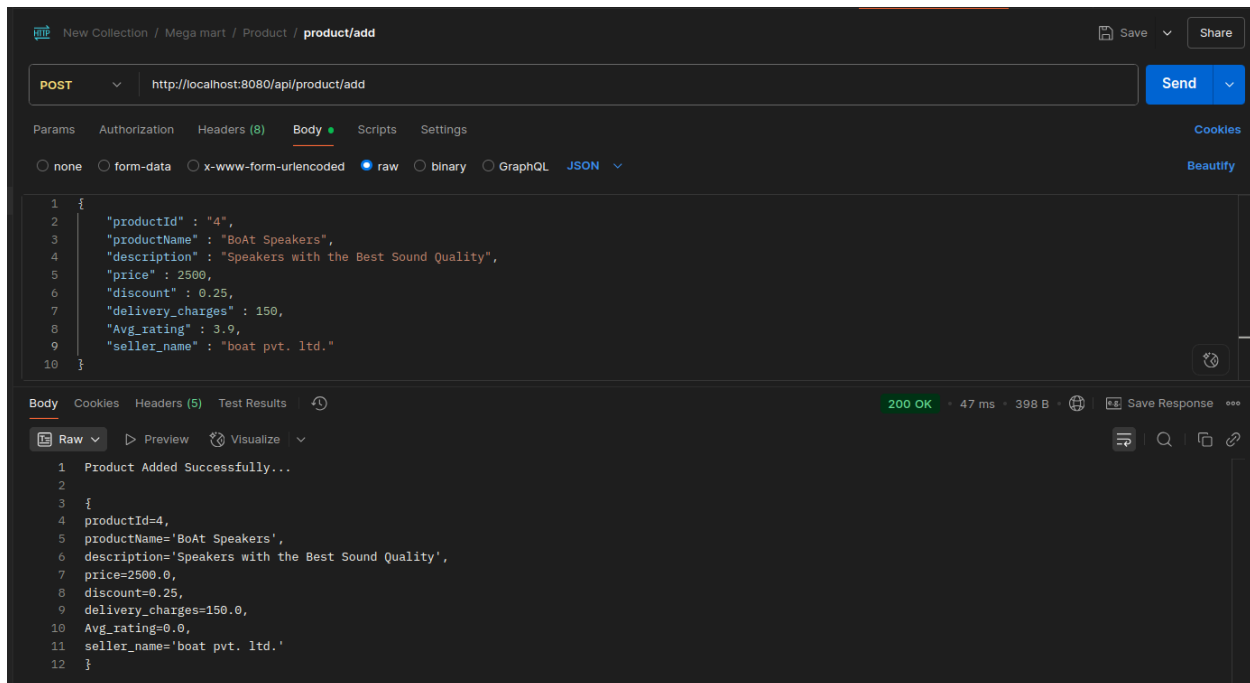


4. Delete Account

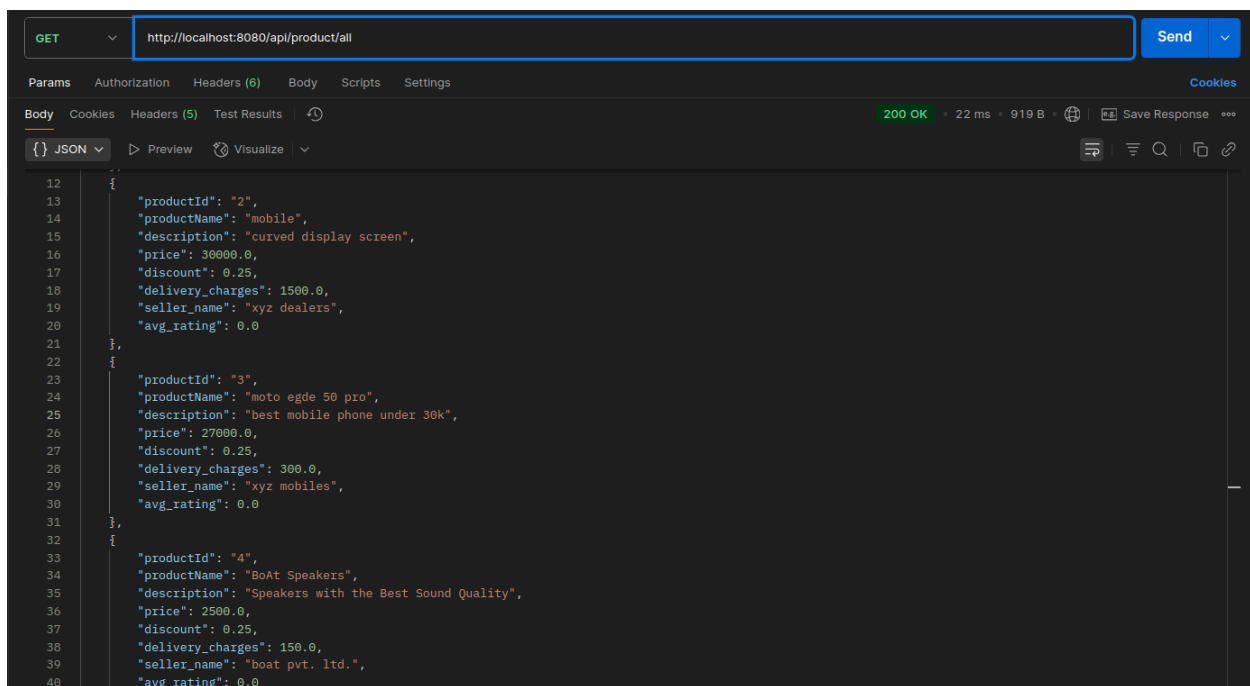


B. Product —>

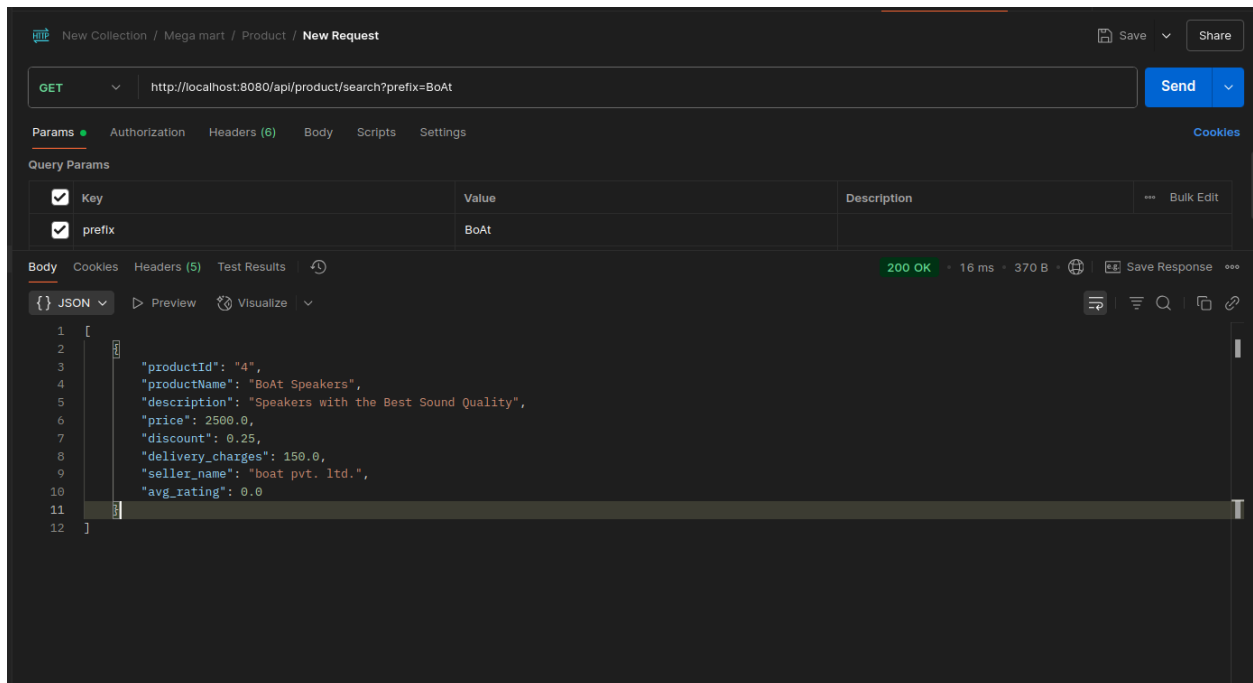
1. Add product



2. view all products

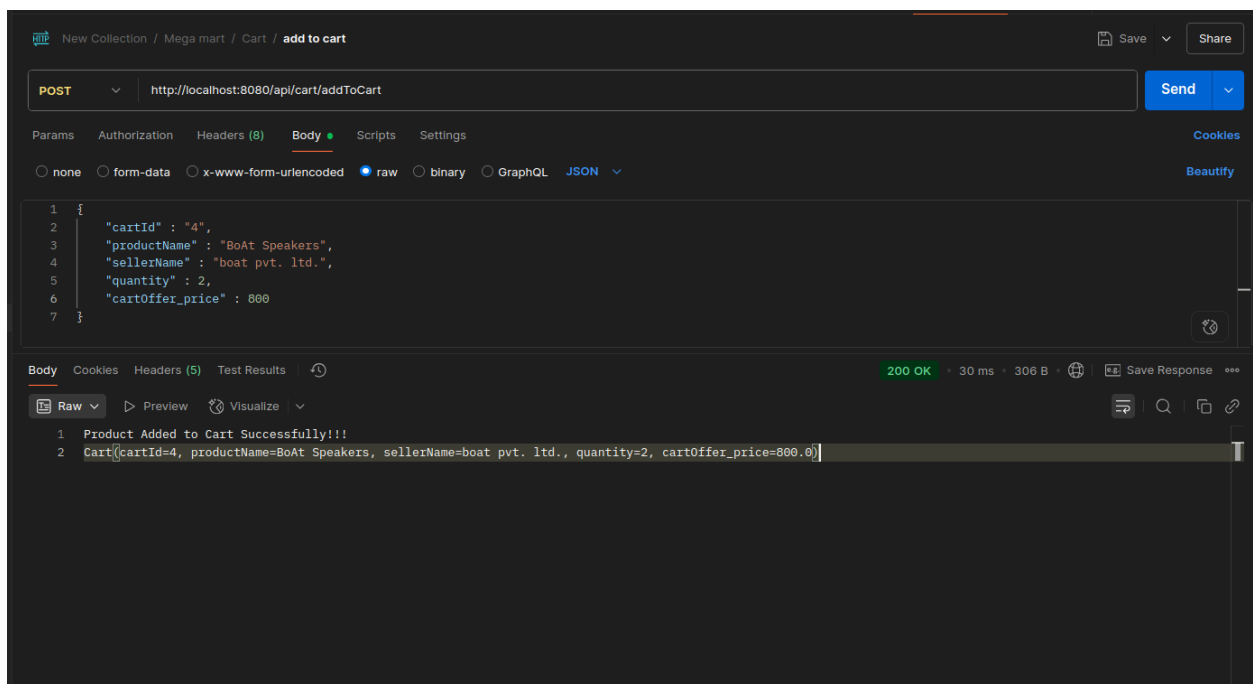


3. Search product

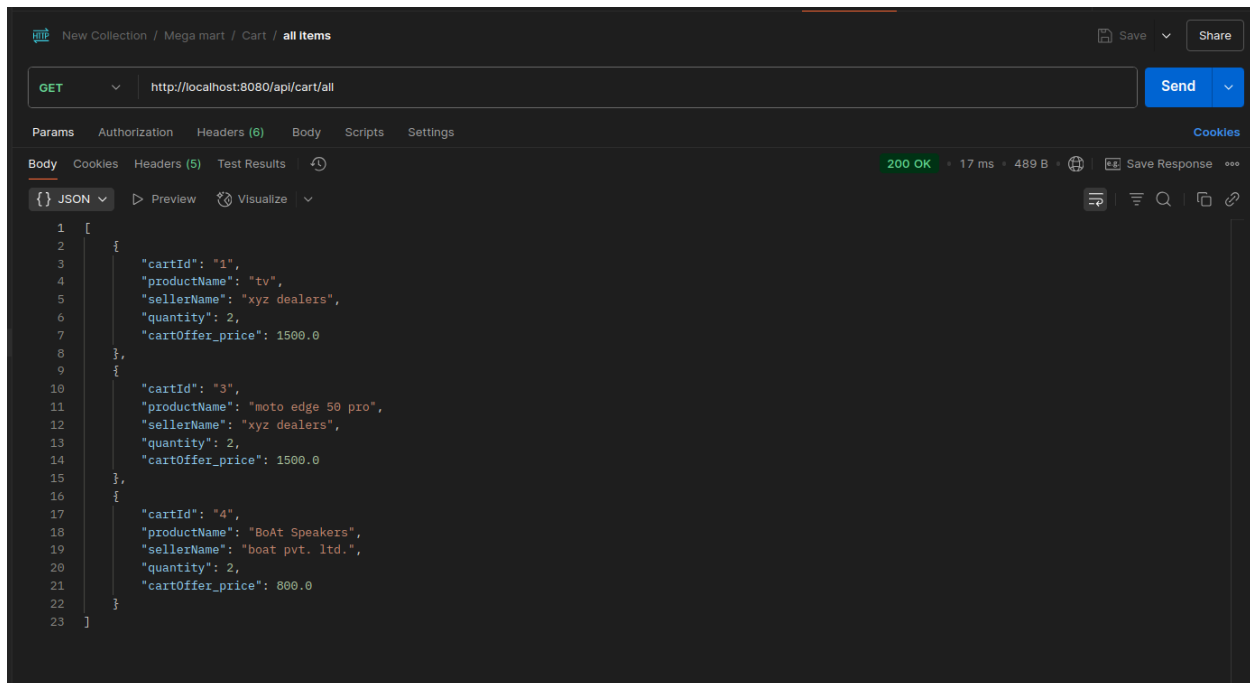


C. Cart →

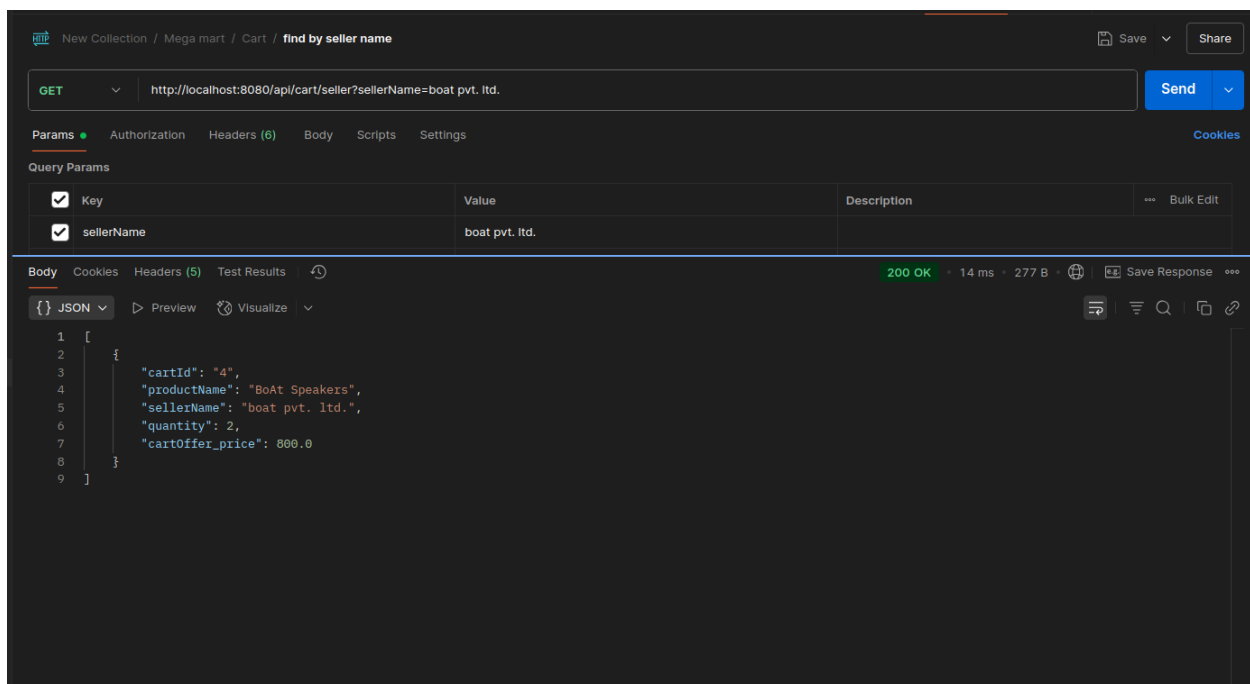
1. Add to Cart



2. view all cart items



3. search by dealer name



4. import java.util.UUID;search by product name

New Collection / Mega mart / Cart / **New Request** Save Share

GET ▼ Send ▼

Params ● Authorization Headers (6) Body Scripts Settings Cookies

<input checked="" type="checkbox"/>	Key	Value	Description	⋮ Bulk Edit
<input checked="" type="checkbox"/>	productName	BoAt Speakers		
	Key	Value	Description	

Body Cookies Headers (5) Test Results 🕒 200 OK 15 ms 277 B 🌐 📄 Save Response ⋮

{} JSON ▼ ▶ Preview 🔗 Visualize ▼

```

1  [
2    {
3      "cartId": "4",
4      "productName": "BoAt Speakers",
5      "sellerName": "boat pvt. ltd.",
6      "quantity": 2,
7      "cartOffer_price": 800.0
8    }
9  ]

```

5. Update Cart details

New Collection / Mega mart / Cart / **New Request** Save Share

PUT ▼ Send ▼

Params ● Authorization Headers (7) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	⋮ Bulk Edit
<input checked="" type="checkbox"/>	quantity	4		
	Key	Value	Description	

Body Cookies Headers (5) Test Results 🕒 200 OK 417 ms 275 B 🌐 📄 Save Response ⋮

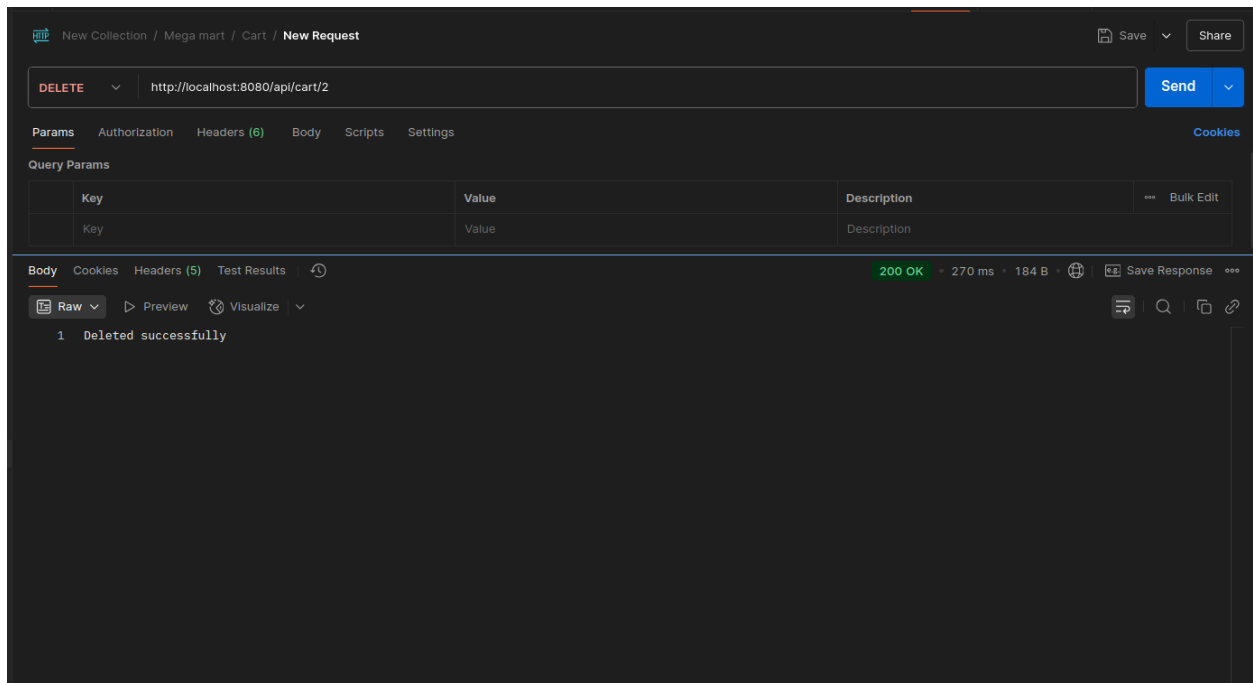
{} JSON ▼ ▶ Preview 🔗 Visualize ▼

```

1  {
2    "cartId": "4",
3    "productName": "BoAt Speakers",
4    "sellerName": "boat pvt. ltd.",
5    "quantity": 4,
6    "cartOffer_price": 800.0
7  }

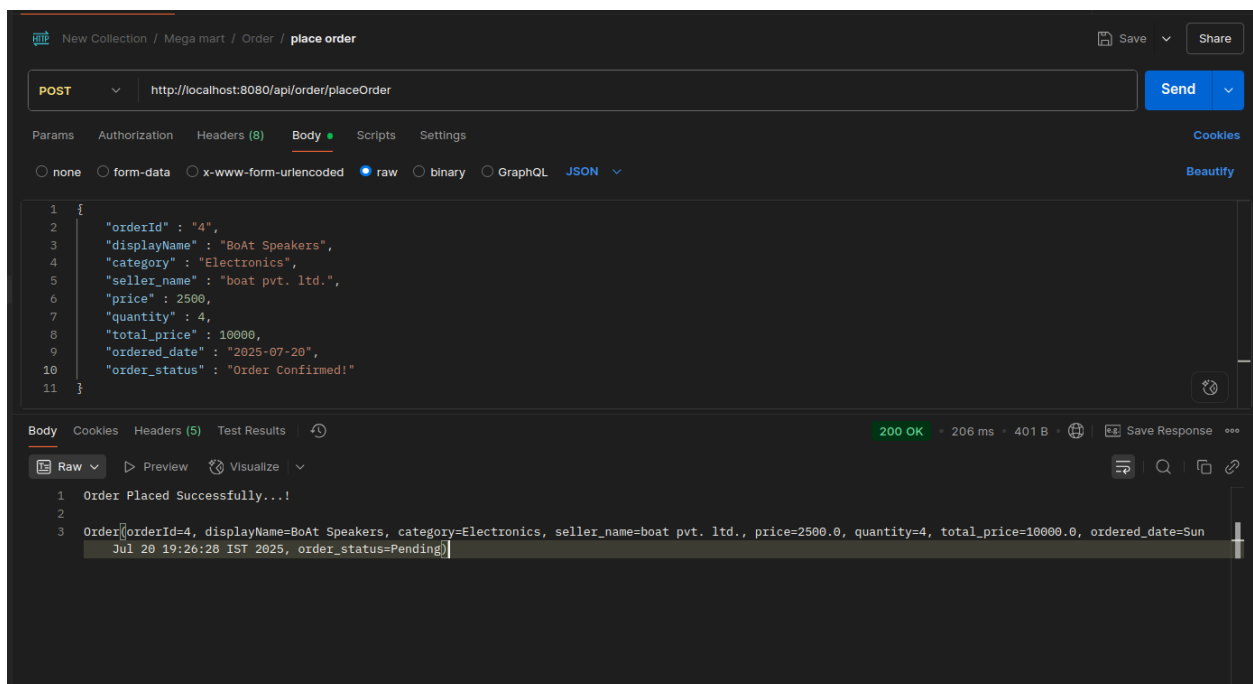
```

6. Delete Cart item



D. Order —>

1. place order



2. view placed orders

GET http://localhost:8080/api/order/getAll Send

Params Authorization Headers (6) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results 200 OK · 208 ms · 813 B Save Response

JSON Preview Visualize

```

9      "total_price": 120000.0,
10     "ordered_date": "2025-07-18T12:43:17.924+00:00",
11     "order_status": "Cancelled"
12   },
13   {
14     "orderId": "3",
15     "displayName": "moto edge 50 pro",
16     "category": "tv",
17     "seller_name": "xyz dealers",
18     "price": 60000.0,
19     "quantity": 2,
20     "total_price": 120000.0,
21     "ordered_date": "2025-07-19T04:58:43.758+00:00",
22     "order_status": "Cancelled"
23   },
24   {
25     "orderId": "4",
26     "displayName": "BoAt Speakers",
27     "category": "Electronics",
28     "seller_name": "boat pvt. ltd.",
29     "price": 2500.0,
30     "quantity": 4,
31     "total_price": 10000.0,
32     "ordered_date": "2025-07-20T13:56:28.063+00:00",
33     "order_status": "Pending"

```

3. cancel order

DELETE http://localhost:8080/api/order/cancelOrder/4 Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results 200 OK · 40 ms · 212 B Save Response

Raw Preview Visualize

```

1  Order with ID 4 has been cancelled successfully.

```

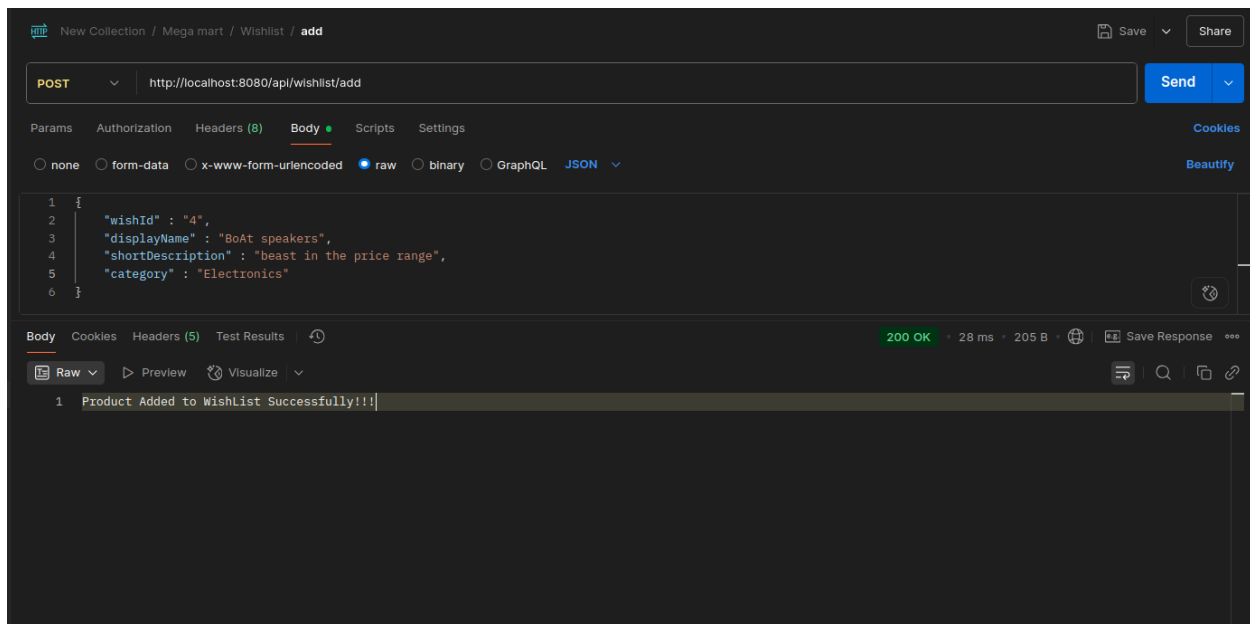
```

24     {
25         "orderId": "4",
26         "displayName": "BoAt Speakers",
27         "category": "Electronics",
28         "seller_name": "boat pvt. ltd.",
29         "price": 2500.0,
30         "quantity": 4,
31         "total_price": 10000.0,
32         "ordered_date": "2025-07-20T13:56:28.063+00:00",
33         "order_status": "Cancelled"
34     }
35 }

```

E. WishList —>

1. Add to Wishlist



2. View all wishlisted items

New Collection / Mega mart / Wishlist / **all** Save Share

GET ▼ http://localhost:8080/api/wishlist/all Send ▼

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 18 ms 362 B Save Response ...

{} JSON ▼ Preview Visualize ▼

```

1  [
2    {
3      "wishId": "1",
4      "displayName": "tv",
5      "shortDescription": "budget tv",
6      "category": "tv"
7    },
8    {
9      "wishId": "4",
10     "displayName": "BoAt speakers",
11     "shortDescription": "beast in the price range",
12     "category": "Electronics"
13   }
14 ]

```

3. remove from wishlist

New Collection / Mega mart / Wishlist / **by id** Save Share

DELETE ▼ http://localhost:8080/api/wishlist/4 Send ▼

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 33 ms 214 B Save Response ...

Raw ▼ Preview Visualize ▼

```

1  Wishlist Item with id 4 is deleted Successfully!!!

```

Thank You !