MongoDB\_CURD.docx

CRUD Operations for Home Cook Meal Platform

# Users

## Create

db.users.insertOne({  
 \_id: "u011",  
 name: "Meera Joseph",  
 email: "meera@example.com",  
 password: "hashed\_meera",  
 role: "cook",  
 profile: {  
 photo: "meera.jpg",  
 bio: "Healthy South Indian meals",  
 cuisines: ["South Indian"],  
 certified: true  
 },  
 address: { city: "Ernakulam", pincode: "682001" }  
});

## Read

db.users.find({ role: "cook" });

## Update

db.users.updateOne({ \_id: "u001" }, { $set: { "profile.bio": "Updated South Indian chef bio" } });

## Delete

db.users.deleteOne({ \_id: "u011" });

# Meals

## Create

db.meals.insertOne({  
 \_id: "m011",  
 cookId: "u001",  
 name: "Ragi Dosa",  
 price: 75,  
 type: "Breakfast",  
 cuisine: "South Indian",  
 diet: "Gluten-Free",  
 ingredients: ["ragi", "salt", "water"],  
 available: true  
});

## Read

db.meals.find({ cookId: "u001" });

## Update

db.meals.updateOne({ \_id: "m002" }, { $set: { price: 95 } });

## Delete

db.meals.deleteOne({ \_id: "m011" });

# Orders

## Create

db.orders.insertOne({  
 \_id: "o011",  
 userId: "u003",  
 cookId: "u004",  
 items: [{ mealId: "m007", quantity: 2 }],  
 orderDate: new Date("2025-07-23T18:00:00Z"),  
 status: "Pending",  
 total: 200  
});

## Read

db.orders.find({ userId: "u003" });

## Update

db.orders.updateOne({ \_id: "o011" }, { $set: { status: "Delivered" } });

## Delete

db.orders.deleteOne({ \_id: "o011" });

# Payments

## Create

db.payments.insertOne({  
 \_id: "p011",  
 orderId: "o011",  
 userId: "u003",  
 amount: 200,  
 method: "UPI",  
 status: "Success",  
 date: new Date("2025-07-23T18:15:00Z")  
});

## Read

db.payments.find({ userId: "u003" });

## Update

db.payments.updateOne({ \_id: "p011" }, { $set: { status: "Refunded" } });

## Delete

db.payments.deleteOne({ \_id: "p011" });

# Reviews

## Create

db.reviews.insertOne({  
 \_id: "r011",  
 mealId: "m003",  
 userId: "u005",  
 rating: 5,  
 comment: "Amazing keto option!",  
 date: new Date("2025-07-23T20:00:00Z")  
});

## Read

db.reviews.find({ mealId: "m003" });

## Update

db.reviews.updateOne({ \_id: "r011" }, { $set: { rating: 4 } });

## Delete

db.reviews.deleteOne({ \_id: "r011" });

# Tickets

## Create

db.tickets.insertOne({  
 \_id: "t011",  
 userId: "u003",  
 issue: "Can't apply discount coupon",  
 status: "open",  
 messages: [  
 { from: "user", text: "Coupon not applying during checkout", timestamp: new Date("2025-07-23T11:30:00Z") }  
 ]  
});

## Read

db.tickets.find({ status: "open" });

## Update

db.tickets.updateOne({ \_id: "t011" }, { $set: { status: "closed" } });

## Delete

db.tickets.deleteOne({ \_id: "t011" });