**Experiment: CRUD Operations for Product Database Using Mongoose**

**Aim**

To implement Create, Read, Update, and Delete (CRUD) operations on a Product Database using Mongoose (an ODM for MongoDB) in a Node.js application.

**Theory**

1. CRUD Operations
   * Create → Insert new product records into the database.
   * Read → Retrieve product details.
   * Update → Modify existing product records.
   * Delete → Remove product records.
2. Mongoose
   * Mongoose is an Object Data Modeling (ODM) library for MongoDB.
   * Provides schema-based modeling, validation, and easy-to-use query methods.
3. Schema & Model
   * Schema defines the structure of documents (fields, types, constraints).
   * Model is a compiled version of the schema, used for performing CRUD.

**Code :**   
  
const mongoose = require("mongoose");

// 1. Connect to MongoDB

mongoose.connect("mongodb://127.0.0.1:27017/productDB", {

useNewUrlParser: true,

useUnifiedTopology: true,

})

.then(() => console.log("✅ MongoDB Connected"))

.catch(err => console.error("❌ Connection Error:", err));

// 2. Define Schema

const productSchema = new mongoose.Schema({

name: { type: String, required: true },

price: { type: Number, required: true },

category: String,

inStock: { type: Boolean, default: true }

});

// 3. Create Model

const Product = mongoose.model("Product", productSchema);

// 4. CRUD Operations

// CREATE

async function createProduct() {

const product = new Product({

name: "Laptop",

price: 55000,

category: "Electronics",

inStock: true,

});

const result = await product.save();

console.log("Created:", result);

}

// READ

async function getProducts() {

const products = await Product.find();

console.log("All Products:", products);

}

// UPDATE

async function updateProduct(id) {

const result = await Product.findByIdAndUpdate(

id,

{ price: 60000, inStock: false },

{ new: true }

);

console.log("Updated:", result);

}

// DELETE

async function deleteProduct(id) {

const result = await Product.findByIdAndDelete(id);

console.log("Deleted:", result);

}

// Run functions

(async () => {

await createProduct();

await getProducts();

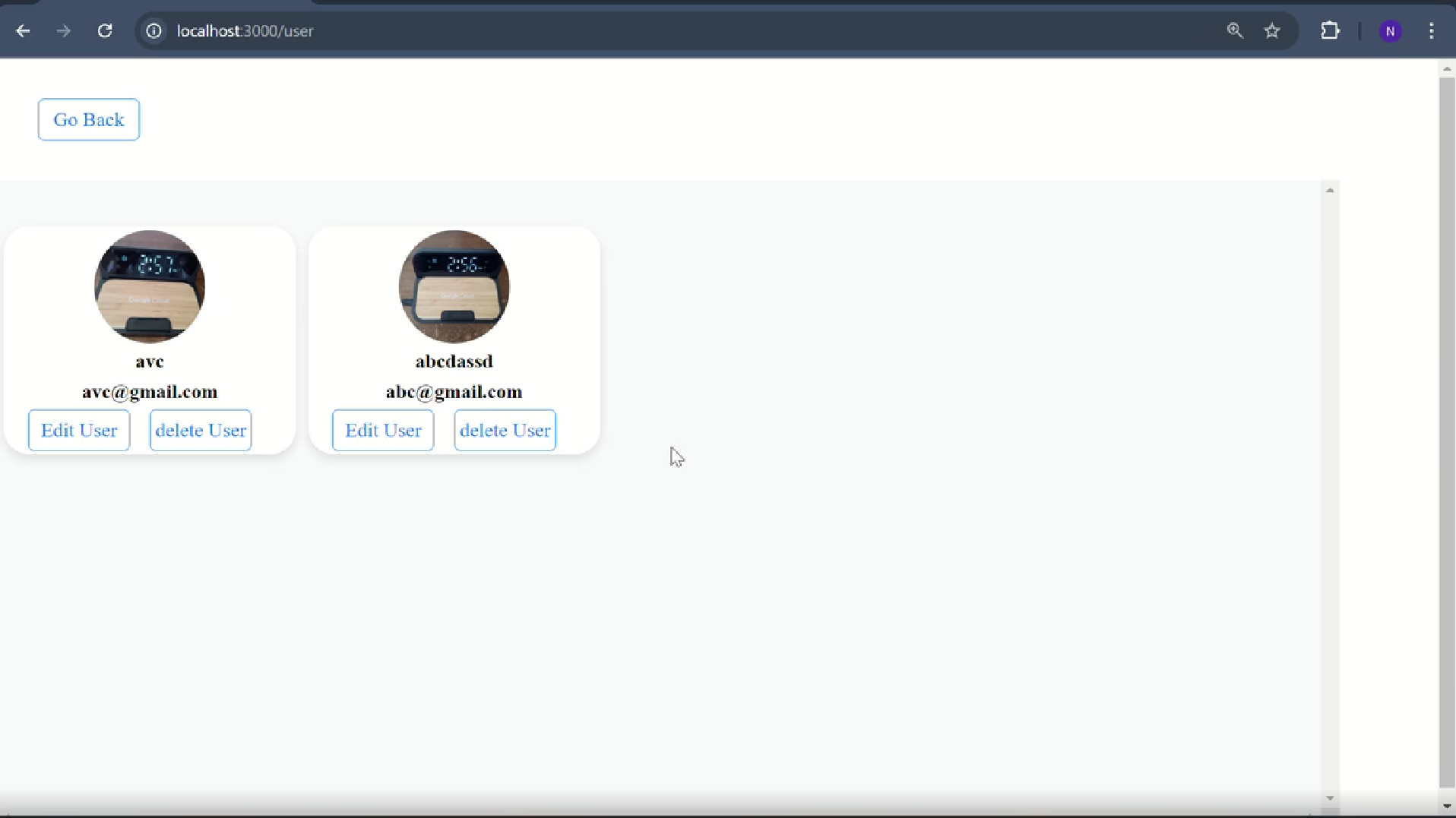
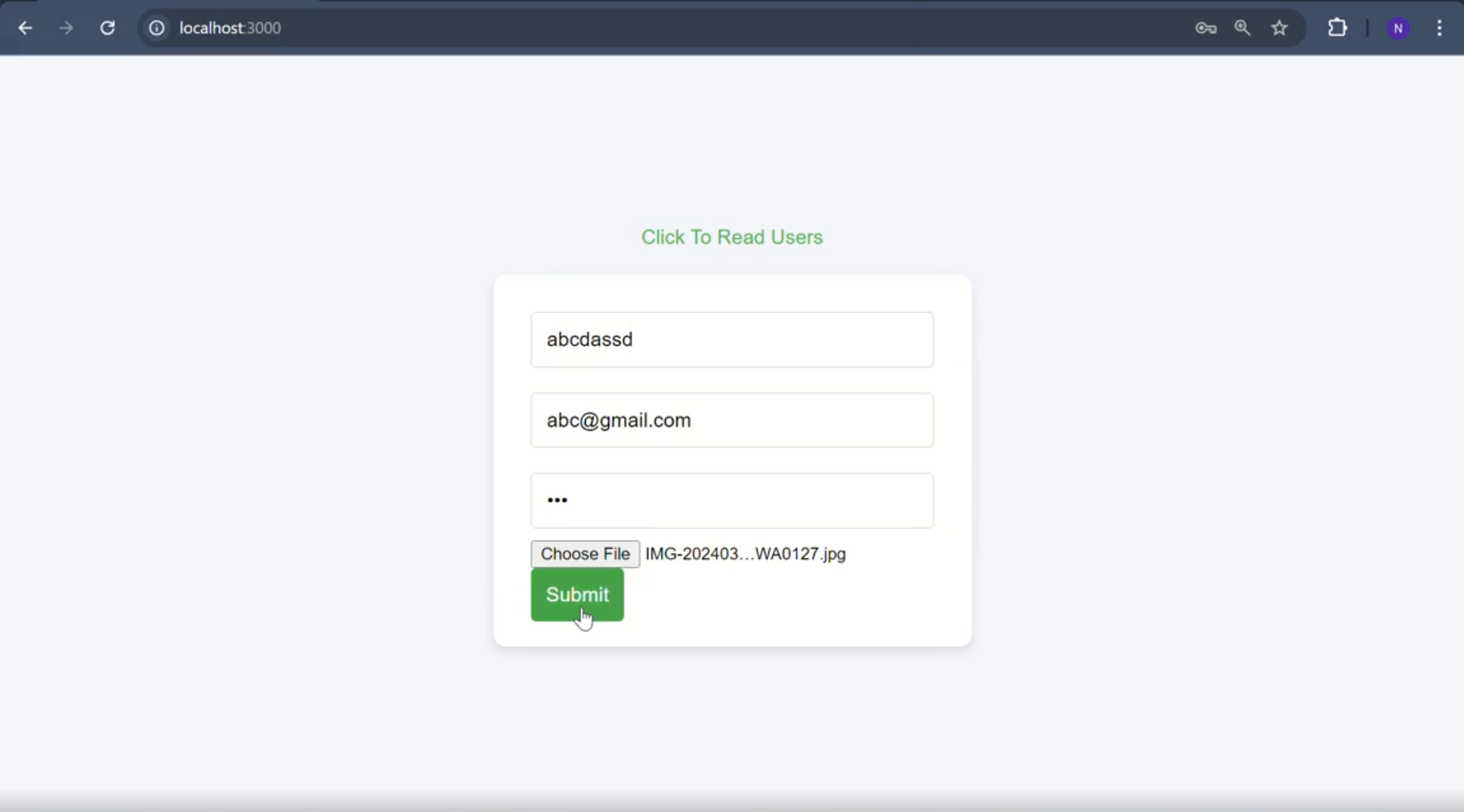
// Replace with actual \_id after creating

// await updateProduct("64f9d0b0e1a9c3bcd1234567");

// await deleteProduct("64f9d0b0e1a9c3bcd1234567");

})();

**Output**



**Learning Outcomes**

* Understood CRUD operations in MongoDB using Mongoose.
* Learned how to define schemas and models in Mongoose.
* Implemented asynchronous functions with async/await.
* Practiced connecting Node.js with MongoDB.
* Gained skills to build a basic product management system.