Experiment 3

AIM:- CRUD Operations for Product Database Using Mongoose

CODE:-

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Product CRUD App</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <h1>📦 Product Manager</h1>

    <!-- Product Form -->
    <form id="productForm" class="card">
      <input type="text" id="name" placeholder="Product Name" required>
      <input type="number" id="price" placeholder="Price (₹)" required>
      <input type="text" id="category" placeholder="Category">
      <button type="submit" class="btn">Add Product</button>
    </form>

    <!-- Product List -->
    <div class="card">
```

```
      <h2>All Products</h2>
      <ul id="productList"></ul>
    </div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS

```
/* General */
body {
  font-family: "Poppins", sans-serif;
  background: linear-gradient(120deg, #667eea, #764ba2);
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: flex-start;
  min-height: 100vh;
}

.container {
  max-width: 600px;
  width: 100%;
  margin: 30px;
  color: #333;
}
```

```css
h1 {
  text-align: center;
  color: white;
  margin-bottom: 20px;
}

.card {
  background: white;
  border-radius: 12px;
  padding: 20px;
  box-shadow: 0px 5px 15px rgba(0,0,0,0.15);
  margin-bottom: 20px;
}

h2 {
  margin-top: 0;
}

/* Form */
form input {
  width: calc(33% - 10px);
  padding: 10px;
  margin: 5px 5px 10px 0;
  border: 1px solid #ccc;
  border-radius: 8px;
}

.btn {
  background: #667eea;
  color: white;
  padding: 10px 15px;
  border: none;
  border-radius: 8px;
  cursor: pointer;
  transition: 0.3s;
}

.btn:hover {
  background: #5a67d8;
}

/* Product List */
ul {
  list-style: none;
  padding: 0;
  margin: 0;
}

li {
  background: #f8f9fa;
  margin-bottom: 10px;
  padding: 12px;
  border-radius: 8px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

```
li span {
  font-weight: 500;
}

.actions button {
  margin-left: 5px;
  padding: 6px 10px;
  border: none;
  border-radius: 6px;
  cursor: pointer;
  font-size: 14px;
}

.edit {
  background: #38a169;
  color: white;
}

.delete {
  background: #e53e3e;
  color: white;
}
```

JavaScript

```
const form = document.getElementById("productForm");
const productList = document.getElementById("productList");

let products = []; // dummy local array (replace later with API calls)
```

```
// Render Products
function renderProducts() {
  productList.innerHTML = "";
  products.forEach((p, i) => {
    const li = document.createElement("li");
    li.innerHTML = `
      <span>${p.name} - ₹${p.price} (${p.category || "Uncategorized"})</span>
      <div class="actions">
        <button class="edit" onclick="editProduct(${i})">Edit</button>
        <button class="delete" onclick="deleteProduct(${i})">Delete</button>
      </div>
    `;
    productList.appendChild(li);
  });
}

// Add Product
form.addEventListener("submit", (e) => {
  e.preventDefault();
  const product = {
    name: document.getElementById("name").value,
    price: document.getElementById("price").value,
    category: document.getElementById("category").value
  };
  products.push(product);
  renderProducts();
  form.reset();
});
```

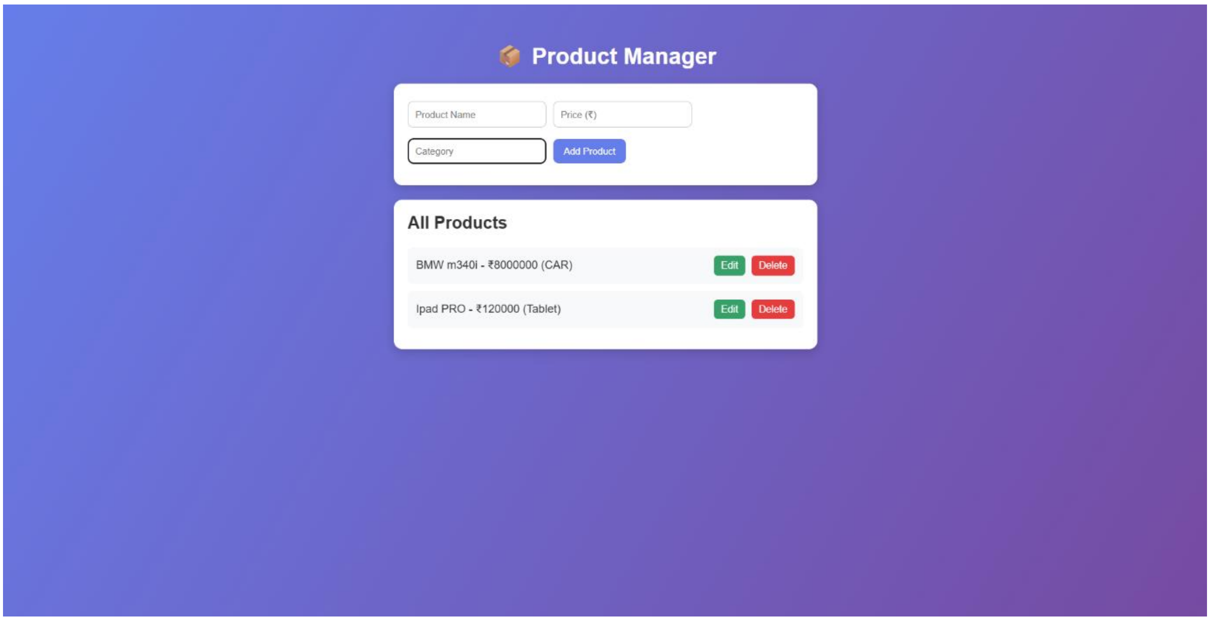
```
// Delete Product
function deleteProduct(index) {
  products.splice(index, 1);
  renderProducts();
}

// Edit Product
function editProduct(index) {
  const p = products[index];
  document.getElementById("name").value = p.name;
  document.getElementById("price").value = p.price;
  document.getElementById("category").value = p.category;

  products.splice(index, 1); // remove old entry so updated one can be re-added
  renderProducts();
}

// Initial render
renderProducts();
```

OUTPUT:



All Products

BMW m340i - ₹8000000 (CAR)

Ipad PRO - ₹120000 (Tablet)

Wanna try it out then go on to the following link
http://127.0.0.1:3000/index.html