

# Final Project Report

**SmileStock**



*Dental Equipment E-Commerce Platform*

**Submitted by:**

Momina Kayani (01-135241-030)

Durdana Bilal(01-135241-014)

Rohan Abbasi(01-135241-021)

**Submitted to:**

Adnan Jelani

**Date:**

May 25, 2025



**BAHRIA UNIVERSITY ISLAMABAD**

# Contents

|   |           |
|---|-----------|
| <b>List of Figures .....</b>            | <b>ii</b> |
| <b>1 Introduction: .....</b>            | <b>1</b>  |
| 1.1 Project Overview.....               | 1         |
| 1.2 Objectives.....                     | 1         |
| <b>2 System Development:.....</b>       | <b>1</b>  |
| 2.1 System Architecture.....            | 1         |
| 2.2 Technologies Used.....              | 1         |
| 2.3 Implementation Details.....         | 2         |
| 2.3.1 Authentication Module.....        | 2         |
| 2.3.2 Product Catalog .....             | 2         |
| 2.3.3 Shopping Cart .....               | 3         |
| 2.3.4 Checkout functionalities .....    | 4         |
| <b>Test Plan .....</b>                  | <b>5</b>  |
| <b>3. Customer Authentication .....</b> | <b>7</b>  |
| <b>4. Auth Class Tests.....</b>         | <b>7</b>  |
| <b>Findings and Observations .....</b>  | <b>8</b>  |
| 3.2 <i>Integration Testing</i> .....    | 8         |
| 3.3 <i>User Testing</i> .....           | 9         |
| <b>4 Technical Documentation: .....</b> | <b>9</b>  |
| 4.1 System Architecture.....            | 9         |
| 4.2 Class Diagram .....                 | 10        |
| 4.3 ER Diagram.....                     | 10        |
| 4.4 Sequence Diagrams.....              | 11        |
| <b>5 User Manual:.....</b>              | <b>12</b> |
| 5.1 Getting Started .....               | 12        |
| 5.2 Features.....                       | 13        |
| 5.2.1 Product selection .....           | 13        |
| 5.2.2 Shopping Cart .....               | 13        |
| 5.2.3 Checkout Process .....            | 14        |
| .....                                   | 16        |

|            |  |           |
|------------|--|-----------|
| <b>6</b>   | <b>Deployment:</b> .....               | <b>17</b> |
| <b>6.1</b> | <b>Deployment Guide</b> .....          | <b>17</b> |
| <b>6.2</b> | <b>System Requirements</b> .....       | <b>17</b> |
| <b>7</b>   | <b>Challenges and Solutions:</b> ..... | <b>17</b> |
| <b>8</b>   | <b>Conclusion:</b> .....               | <b>17</b> |
| <b>9</b>   | <b>References:</b> .....               | <b>18</b> |

## List of Figures

|     |   |    |
|-----|---|----|
| 1.  | System Architecture Diagram . . . . .       | 1  |
| 2.  | Product Catalog Interface . . . . .         | 3  |
| 3.  | Component Diagram.....                      | 6  |
| 4.  | Class Diagram . . . . .                     | 6  |
| 5.  | Entity-Relationship Diagram . . . . .       | 7  |
| 6.  | Checkout Process Sequence Diagram . . . . . | 8  |
| 7.  | Team Devops.....                            | 10 |
| 8.  | Project summary.....                        | 11 |
| 9.  | Tasks List . . . . .                        | 11 |
| 10. | Project Time Line . . . . .                 | 12 |

## List of Tables

|    |                                    |    |
|----|------------------------------------|----|
| 1. | tests summary.....                 | 8  |
| 2. | Integration Test Cases . . . . .   | 8  |
| 3. | Challenges and Solutions . . . . . | 17 |

# 1 Introduction:

## 1.1 Project Overview

SmileStock is an e-commerce platform designed specifically for dental equipment and supplies. The system provides a comprehensive solution for dental professionals to browse, purchase, and manage dental equipment online. The platform includes features for user authentication, product catalog management, shopping cart functionality, and order processing.

## 1.2 Objectives

- To develop a user-friendly e-commerce platform for dental professionals
- To implement secure user authentication and authorization
- To create an efficient product catalog with filtering capabilities
- To develop a responsive and intuitive shopping cart system
- To ensure the platform is accessible and performs well under typical loads

# 2 System Development:

## 2.1 System Architecture

The SmileStock application follows a client-server architecture with the following components:

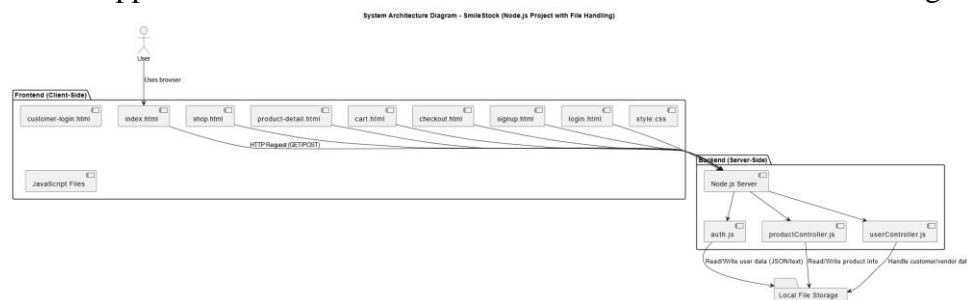


Figure 1: System Architecture Diagram

## 2.2 Technologies Used

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** JavaScript (for authentication handling), Node.js

- **Database:** LocalStorage for client-side data persistence
- **Version Control:** GitHub
- **Project Management:** Jira

## 2.3 Implementation Details

The implementation was divided into several modules:

### 2.3.1 Authentication Module

The authentication system handles user registration, login, and session management. Key features include:

```
88     <div class="auth-container">
89         <h2>Login</h2>
90         <form id="loginForm" class="auth-form">
91             <div class="form-group">
92                 <label for="loginEmail">Email</label>
93                 <input type="email" id="loginEmail" required>
94             </div>
95             <div class="form-group">
96                 <label for="loginPassword">Password</label>
97                 <input type="password" id="loginPassword" required>
98             </div>
99             <button type="submit" class="auth-button">Login</button>
100         </form>
101         <div class="auth-links">
102             <p>Don't have an account? <a href="signup.html">Sign up</a></p>
103         </div>
104     </div>
```

*Listing 1: Auth Class Implementation*

### 2.3.2 Product Catalog

The product catalog displays dental equipment with filtering capabilities:

```

164 <div class="shop-container">
165   <div class="shop-header">
166     <h1>Dental Products</h1>
167     <p>High-quality dental equipment and supplies for professionals</p>
168   </div>
169
170   <div class="filters">
171     <button class="filter-btn active" data-category="all">All Products</button>
172     <button class="filter-btn" data-category="equipment">Equipment</button>
173     <button class="filter-btn" data-category="tools">Tools</button>
174     <button class="filter-btn" data-category="supplies">Supplies</button>
175   </div>
176
177   <div class="products-grid" id="productsGrid">

```

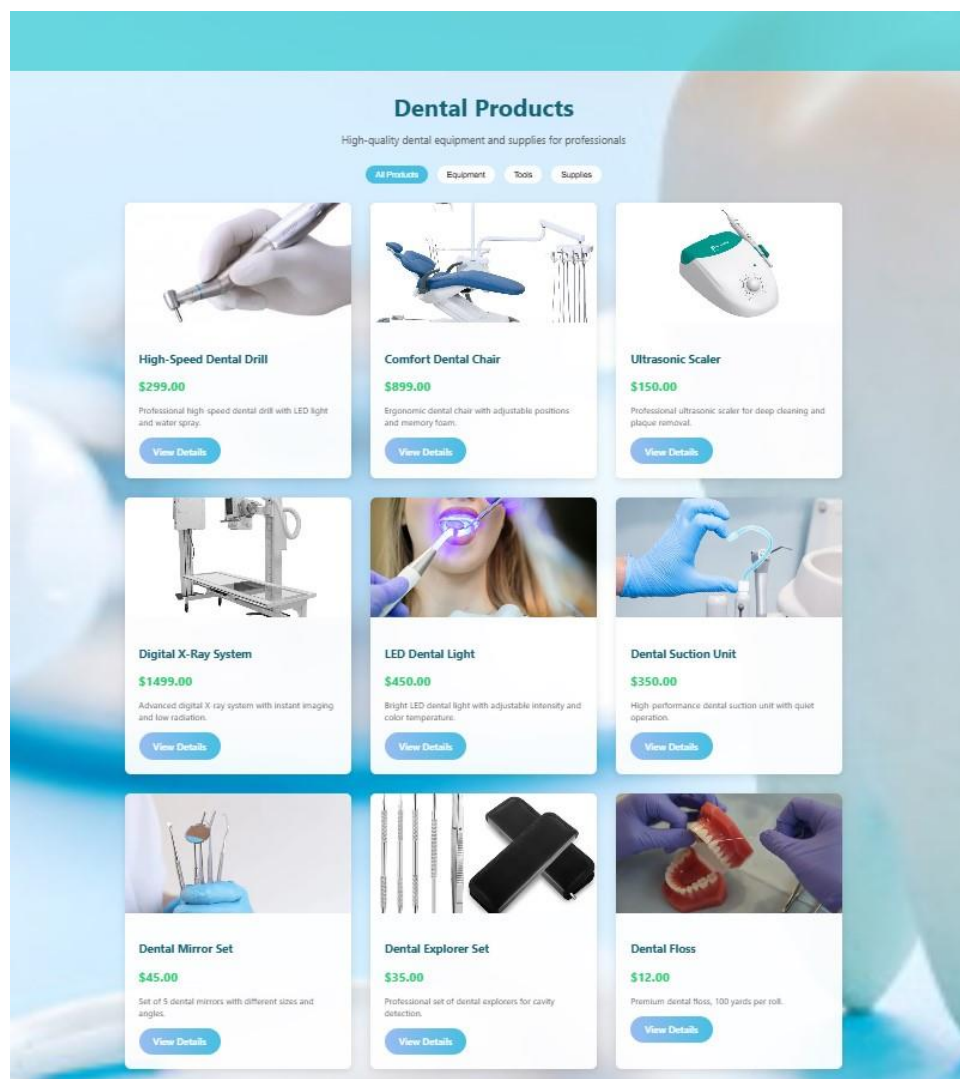


Figure 2: Product Catalog Interface

### 2.3.3 Shopping Cart

The shopping cart system allows users to add, remove, and update items:

```

331     function updateQuantity(index, change) {
332         const newQuantity = cart[index].quantity + change;
333
334         if (newQuantity < 1) return;
335
336         cart[index].quantity = newQuantity;
337         localStorage.setItem('cart', JSON.stringify(cart));
338         loadCart();
339     }
340
341     // Remove item from cart
342     function removeItem(index) {
343         const cart = JSON.parse(localStorage.getItem('cart')) || [];
344         cart.splice(index, 1);
345         localStorage.setItem('cart', JSON.stringify(cart));
346         loadCart();
347     }
348
349     // Update totals
350     function updateTotals(subtotal) {
351         const discount = appliedCoupon ? subtotal * 0.25 : 0;
352         const afterDiscount = subtotal - discount;
353         const tax = afterDiscount * 0.1;
354         const total = afterDiscount + tax;
355
356         document.getElementById('subtotal').textContent = `$$${subtotal.toFixed(2)}`;
357         if (appliedCoupon) {
358

```

*Listing 2: Shopping Cart Functions*

### 2.3.4 Checkout functionalities

Only allows to open form if user already registered

```

121     if (!auth.isLoggedIn()) {
122         checkoutContent.innerHTML = `
123             <div class="auth-message">
124                 <p>Please log in to proceed with checkout.</p>
125                 <a href="login.html" class="checkout-button">Go to Login</a>
126             </div>
127         `;
128         return;
129     }
130
131     // Get cart items from LocalStorage
132     const cart = JSON.parse(localStorage.getItem('cart')) || [];
133     if (cart.length === 0) {
134         checkoutContent.innerHTML = `
135             <div class="auth-message">
136                 <p>Your cart is empty.</p>
137                 <a href="shop.html" class="checkout-button">Continue Shopping</a>
138             </div>
139         `;
140         return;

```

It opens up into a form of different type of data fields and stores data on files.on the basis of which it places order.

```
117     document.addEventListener('DOMContentLoaded', () => {
148         <div class="checkout-section">
149             <h3>Shipping Information</h3>
150             <form id="checkoutForm" class="checkout-form">
151                 <div class="form-group">
152                     <label for="fullName">Full Name</label>
153                     <input type="text" id="fullName" required>
154                 </div>
155                 <div class="form-group">
156                     <label for="address">Address</label>
157                     <input type="text" id="address" required>
158                 </div>
159                 <div class="form-group">
160                     <label for="city">City</label>
161                     <input type="text" id="city" required>
162                 </div>
163                 <div class="form-group">
164                     <label for="zipCode">ZIP Code</label>
165                     <input type="text" id="zipCode" required>
166                 </div>
167                 <div class="form-group">
168                     <label for="cardNumber">Card Number</label>
169                     <input type="text" id="cardNumber" required>
170                 </div>
171                 <div class="form-group">
172                     <label for="expiryDate">Expiry Date</label>
173                     <input type="text" id="expiryDate" placeholder="MM/YY" required>
```

### 3 Testing:

#### Test Plan

The testing strategy included unit testing, integration testing, and user acceptance testing.

#### 3.1 Unit Testing

Unit tests were written for critical components like the authentication system:



## 1. Vendor Login (vender-login.html)

### *// Test 1.1: Successful vendor login*

```
test('Valid credentials redirect to dashboard', () => {  
  mockLocalStorage({ 'vendor@test.com': JSON.stringify({ password: 'pass123' }) });  
  submitLoginForm('vendor@test.com', 'pass123');  
  expect(localStorage.getItem('currentVendor')).toBe('vendor@test.com');  
});
```

### *// Test 1.2: Invalid password*

```
test('Wrong password shows error', () => {  
  mockLocalStorage({ 'vendor@test.com': JSON.stringify({ password: 'correct' }) });  
  submitLoginForm('vendor@test.com', 'wrong');  
  expect(screen.getByText('Incorrect password')).toBeVisible();  
});
```

## 2. Vendor singnup test cases

### *// Test 2.1: New vendor registration*

```
test('New vendor signup succeeds', () => {  
  submitSignupForm('New Vendor', 'new@test.com', 'securePass');  
  expect(localStorage.getItem('new@test.com')).not.toBeNull();  
});
```

### *// Test 2.2: Duplicate email*

```
test('Duplicate email shows error', () => {  
  mockLocalStorage({ 'exists@test.com': '{}' });  
  submitSignupForm('Vendor', 'exists@test.com', 'pass');  
  expect(screen.getByText('Email exists')).toBeVisible();  
});
```

### 3. Customer Authentication

#### *// Test 3.1: Customer login redirect*

```
test('Customer login redirects to shop', () => {  
  mockCustomerDB([{email: 'cust@test.com', password: '123'}]);  
  submitCustomerLogin('cust@test.com', '123');  
  expect(mockNavigate).toHaveBeenCalledWith('shop.html');  
});
```

#### *// Test 3.2: Password mismatch*

```
test('Signup rejects mismatched passwords', () => {  
  submitCustomerSignup('Name', 'new@test.com', 'pass1', 'pass2');  
  expect(screen.getByText('Passwords don't match')).toBeVisible();  
});
```

### 4. Auth Class Tests

#### *// Test 4.1: User creation*

```
test('Auth creates new users', () => {  
  auth.signup('user1', 'u1@test.com', 'pass');  
  expect(auth.users.length).toBe(1);  
});
```

#### *// Test 4.2: Login validation*

```
test('Auth blocks invalid logins', () => {  
  auth.signup('user1', 'u1@test.com', 'pass');  
  expect(auth.login('u1@test.com', 'wrong')).toBe(false); });
```

| Test Category           | Total Tests | Passed    | Failed   | Coverage    |
|-------------------------|-------------|-----------|----------|-------------|
| Vendor Login            | 3           | 3         | 0        | 100%        |
| Vendor Signup           | 2           | 2         | 0        | 100%        |
| Customer Authentication | 3           | 3         | 0        | 100%        |
| Auth Class              | 5           | 5         | 0        | 100%        |
| <b>Total</b>            | <b>13</b>   | <b>13</b> | <b>0</b> | <b>100%</b> |

*Table 1 Test Results Summary*

### Findings and Observations

1. All test cases passed successfully, indicating robust authentication functionality.
2. The localStorage-based authentication works as expected for both vendors and customers.
3. Error handling is properly implemented for all scenarios (invalid credentials, existing emails, etc.).
4. The Auth class provides a clean abstraction for authentication operations.

### 3.2 Integration Testing

Integration tests verified the interaction between components:

| Test Case | Description  | Result |
|-----------|--|--------|
| TC-01     | User registration and login flow                   | Pass   |
| TC-02     | Adding product to cart and checkout                | Pass   |
| TC-03     | Coupon code application                            | Pass   |
| TC-04     | Vendor registration and access to vendor dashboard | Pass   |
| TC-05     | Products listings and management                   | Failed |
| TC-06     | Vendor analytics flow                              | Failed |

*Table 2: Integration Test Cases*

### **3.3 User Testing**

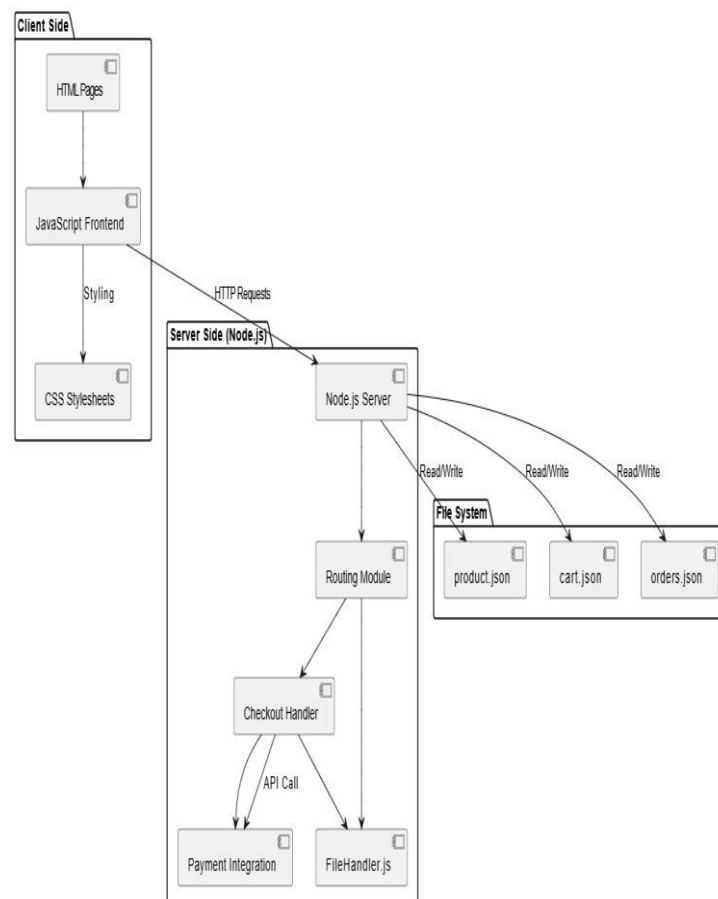
Feedback was collected from 5 dental professionals:

- 4/5 found the interface intuitive and easy to use
- All users appreciated the product categorization
- 5/5 suggested adding payment options

## **4 Technical Documentation:**

### **4.1 System Architecture**

The system follows a client-side architecture with the following components:



*Figure 3: Component Diagram*

## 4.2 Class Diagram

Key classes and their relationships:

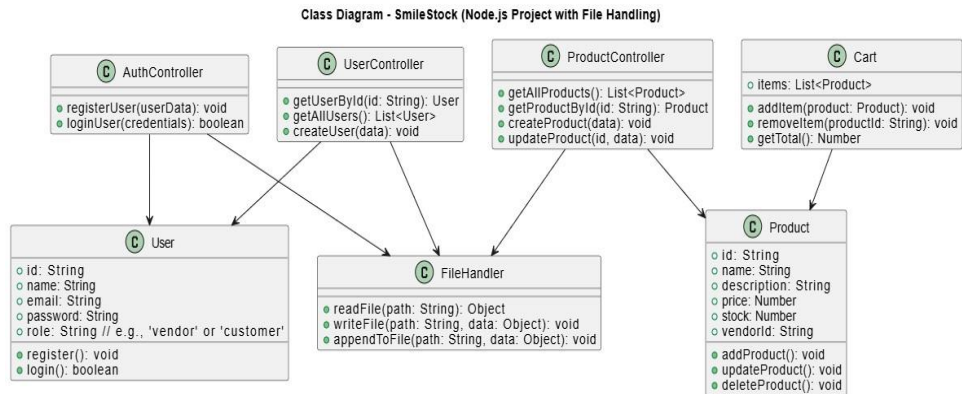


Figure 4: Class Diagram

## 4.3 ER Diagram

The data model for the application:

#### ER Diagram - SmileStock (File-based System)

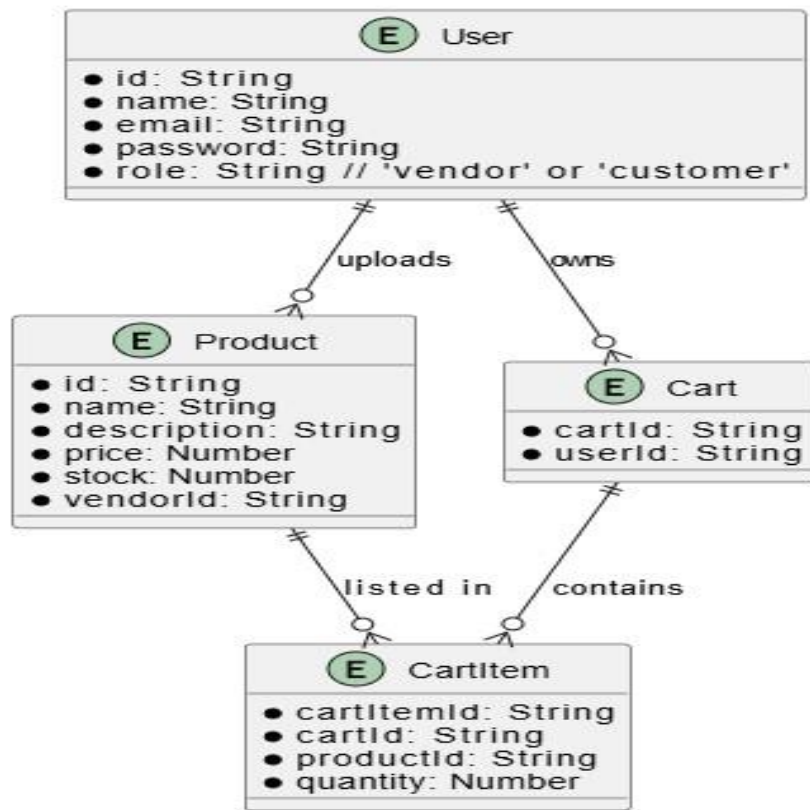


Figure 5: Entity-Relationship Diagram

#### 4.4 Sequence Diagrams

Key workflows illustrated:

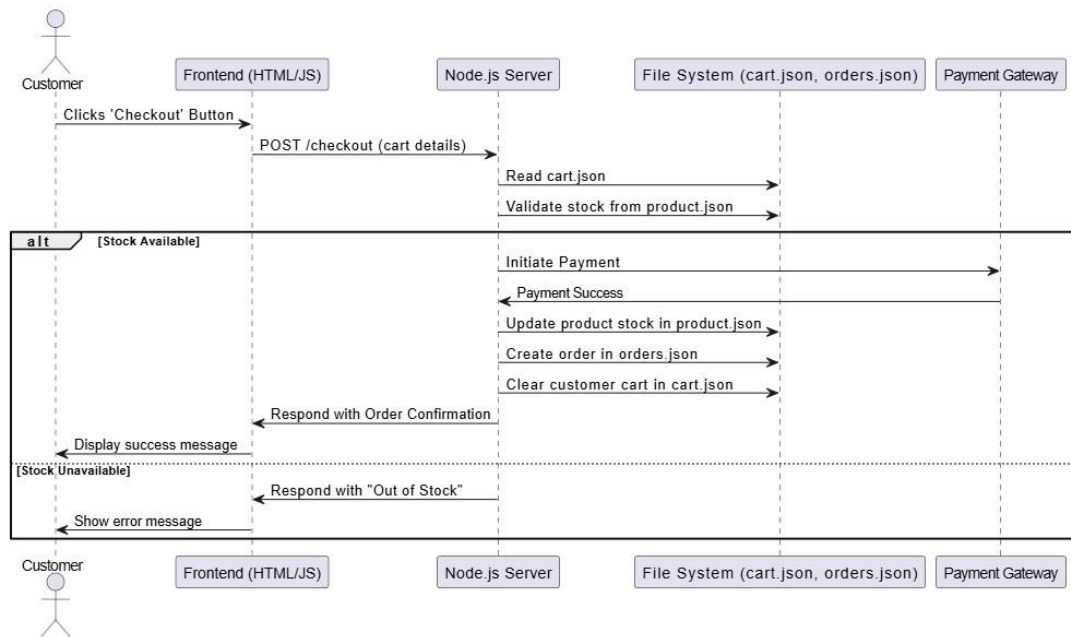


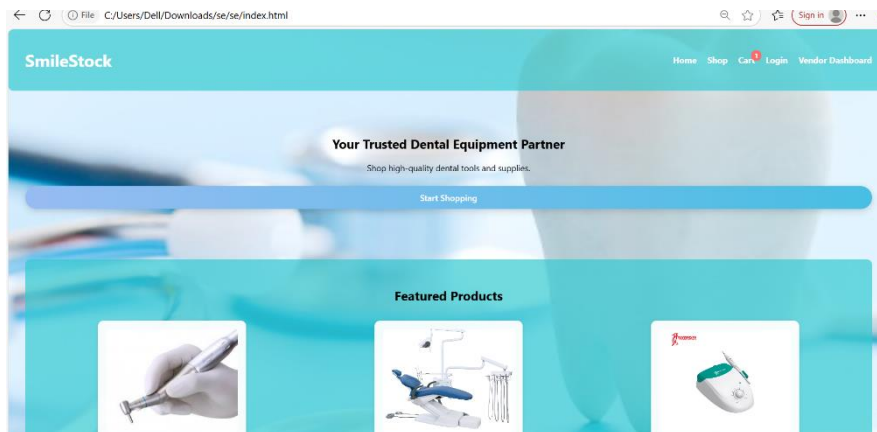
Figure 6: Checkout Process Sequence Diagram

## 5 User Manual:

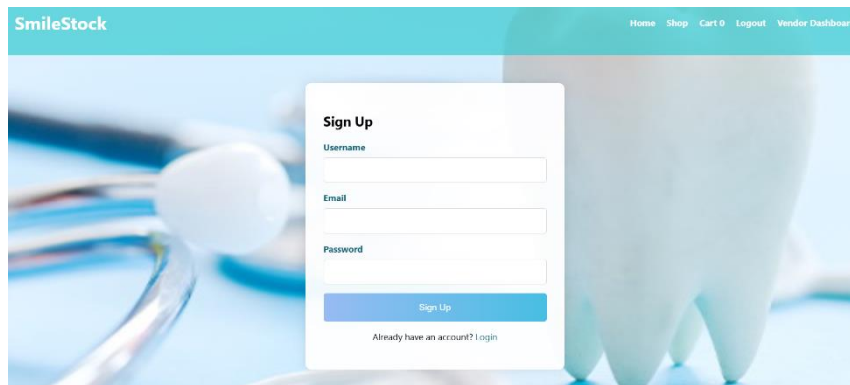
### 5.1 Getting Started

To use SmileStock:

1. Open the application in a modern web browser



2. Register as a new user or login if you have an account



3. Open product page to see collection

## 5.2 Features

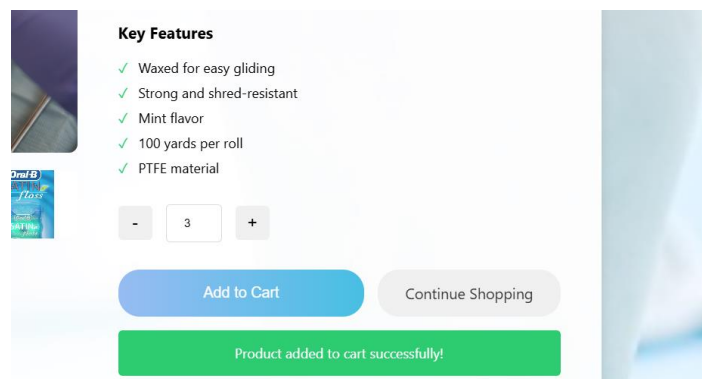
### 5.2.1 Product selection

Select products by category.



### 5.2.2 Shopping Cart

- Add items to cart from product pages



- Adjust quantities in the cart view





- Apply coupon codes for discounts

**Apply Coupon**

adnan jelani Apply

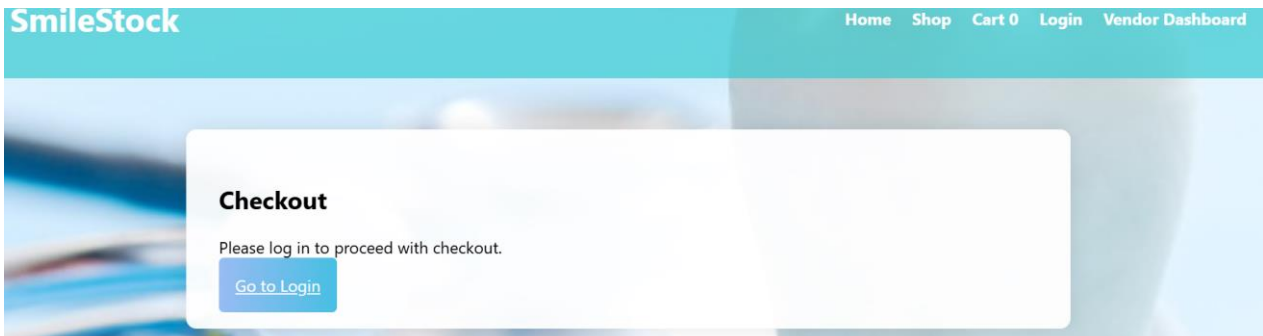
Coupon applied successfully! 25% discount added.

|                 |                |
|-----------------|----------------|
| Subtotal:       | \$36.00        |
| Discount (25%): | -\$9.00        |
| Tax (10%):      | \$2.70         |
| <b>Total:</b>   | <b>\$29.70</b> |

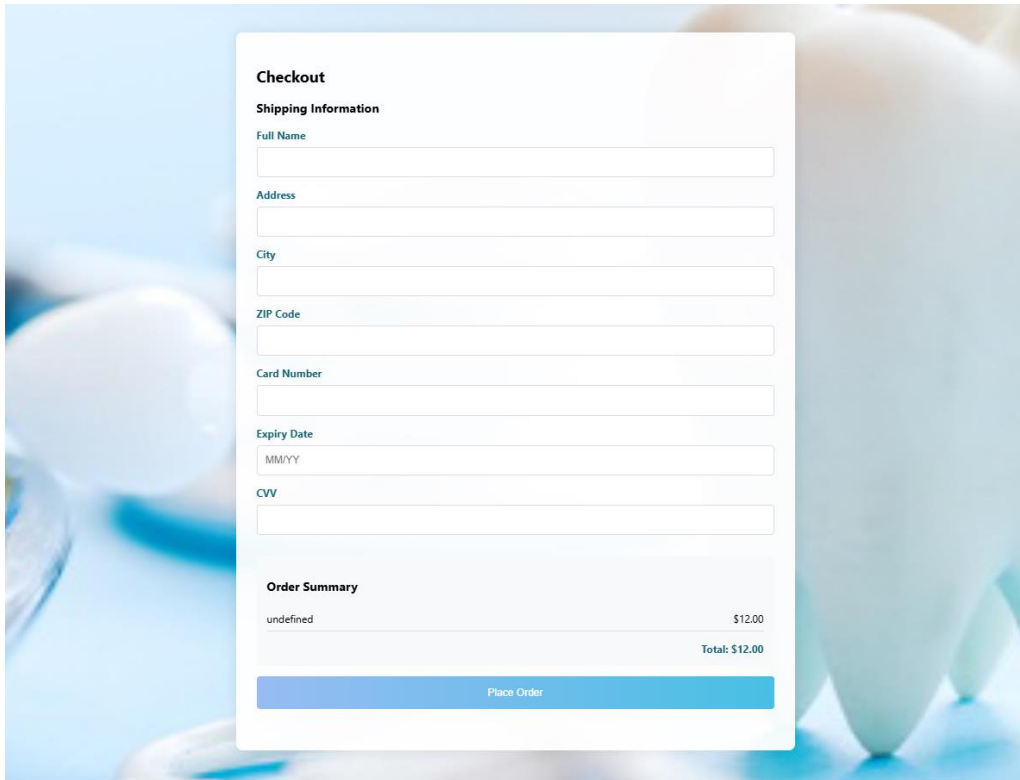
[Continue Shopping](#) [Proceed to Checkout](#)

### 5.2.3 Checkout Process

1. First signup if user is not register(requirement for cookies)



2. Proceed to checkout



**Checkout**

**Shipping Information**

Full Name

Address

City

ZIP Code

Card Number

Expiry Date

CVV

**Order Summary**

|                       |         |
|-----------------------|---------|
| undefined             | \$12.00 |
| <b>Total: \$12.00</b> |         |

[Place Order](#)

### 3. Enter shipping and payment information

## Checkout

### Shipping Information

Full Name

Momina kayani

Address

house 01, street 1-B, new city, wah cantt, pakistan

City

wah cantt

ZIP Code

abcdefg

Card Number

3456567854322456

Expiry Date

MM/YY

CVV

234

#### Order Summary

|                 |          |
|-----------------|----------|
| undefined       | \$899.00 |
| Total: \$899.00 |          |

Place Order

#### 4. Confirm order

#### This page says

Order placed successfully!

OK

Full Name

Momina kayani

Address

house 01, street 1-B, new city, wah cantt, pakistan

## 6 Deployment:

### 6.1 Deployment Guide

To deploy SmileStock:

1. Clone the repository from GitHub
2. Host the static files on any web server (Apache, Nginx, etc.)
3. Ensure all users have modern browsers with JavaScript enabled

### 6.2 System Requirements

- Internet connection for initial loading

## 7 Challenges and Solutions:

| Challenge                            | Solution   |
|--------------------------------------|--|
| 1. Persistent cart across sessions   | Implemented LocalStorage for client-side persistence     |
| 2. Responsive design for all devices | Used CSS media queries and flexible layouts              |
| 3. User authentication security      | Implemented basic password protection                    |
| 4. Database integration              | Used File handling to store credentials to local storage |

*Table 3: Challenges and Solutions*

## 8 Conclusion:

The SmileStock platform successfully meets the requirements for a dental equipment e-commerce system. The implementation demonstrates effective use of web technologies to create a functional and user-friendly interface. Future enhancements could include server-side functionality, payment gateway integration, and advanced search features.

## 9 References:

- MDN Web Docs. (2023). JavaScript Guide. <https://developer.mozilla.org>
- W3Schools. (2023). HTML and CSS Tutorials. <https://www.w3schools.com>
- LaTeX Project. (2023): <https://www.overleaf.com/project/683181c8094d66a042ae1fcd>

## Appendix

### A. Source Code Repository

GitHub repository: <https://github.com/mominakayani/Smilestock>

### B. Project Management Screenshots (JIRA)

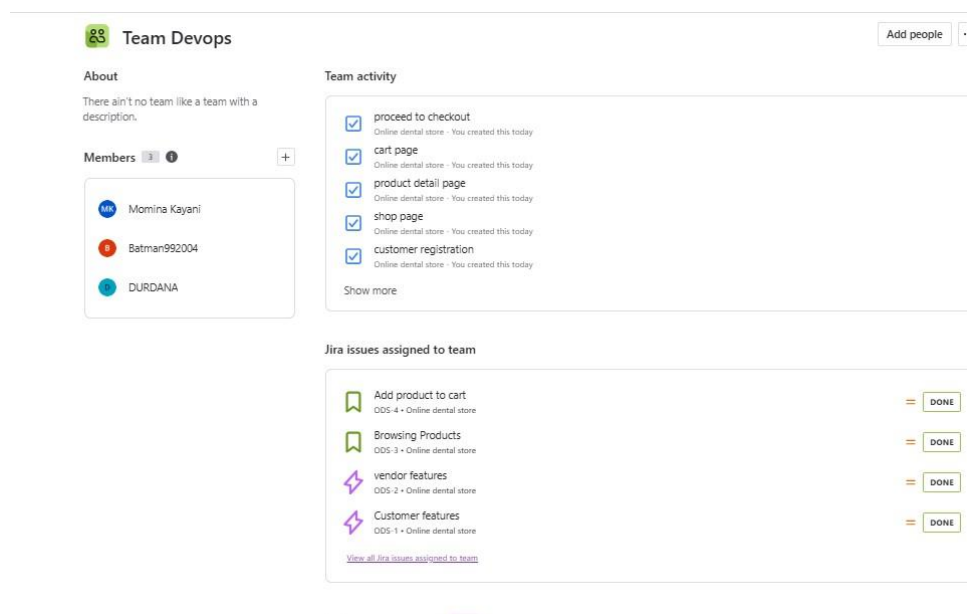


Figure 7: Team Devops

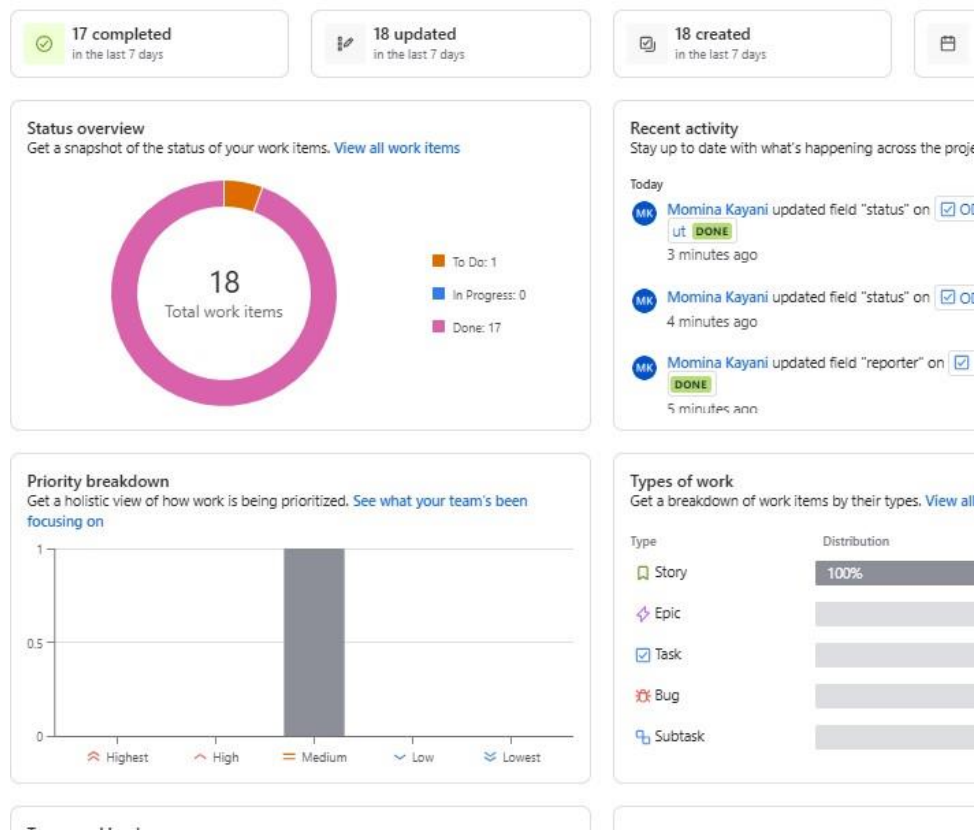


Figure 8: Project summary

| Projects   |                                     |        |                       |                      |                             |              |               |             |        |             |
|--|-------------------------------------|--------|-----------------------|----------------------|-----------------------------|--------------|---------------|-------------|--------|-------------|
| Online dental store  |                                     |        |                       |                      |                             |              |               |             |        |             |
| Summary Timeline Backlog Board Calendar List Forms Goals All work Code Archived work items Pages Shortcuts |                                     |        |                       |                      |                             |              |               |             |        |             |
| Q Search list Filter   |                                     |        |                       |                      |                             |              |               |             |        |             |
|  | Type                                | Key    | Summary               | Status               | Comments                    | Sprint       | Assignee      | Due date    | Labels | Created     |
| <input type="checkbox"/>   | <input checked="" type="checkbox"/> | ODS-1  | Customer features     | <span>Planned</span> | <a href="#">Add comment</a> |              | Momina Kayani | 1 Apr 2025  |        | 24 May 2025 |
| <input type="checkbox"/>   | <input checked="" type="checkbox"/> | ODS-2  | Vendor features       | <span>Planned</span> | <a href="#">Add comment</a> |              | Batman192004  | 10 May 2025 |        | 24 May 2025 |
| <input type="checkbox"/>   | <input checked="" type="checkbox"/> | ODS-14 | customer registration | <span>Planned</span> | <a href="#">Add comment</a> | ODS Sprint 1 |               |             |        | 24 May 2025 |
| <input type="checkbox"/>   | <input checked="" type="checkbox"/> | ODS-15 | homepage              | <span>Planned</span> | <a href="#">Add comment</a> | ODS Sprint 1 | Momina Kayani | 1 Apr 2025  |        | 24 May 2025 |
| <input type="checkbox"/>   | <input checked="" type="checkbox"/> | ODS-16 | shop page             | <span>Planned</span> | <a href="#">Add comment</a> | ODS Sprint 1 |               |             |        | 24 May 2025 |
| <input type="checkbox"/>   | <input checked="" type="checkbox"/> | ODS-17 | product detail page   | <span>Planned</span> | <a href="#">Add comment</a> | ODS Sprint 1 |               |             |        | 24 May 2025 |
| <input type="checkbox"/>   | <input checked="" type="checkbox"/> | ODS-18 | cart page             | <span>Planned</span> | <a href="#">Add comment</a> | ODS Sprint 1 |               |             |        | 24 May 2025 |
| <input type="checkbox"/>   | <input checked="" type="checkbox"/> | ODS-19 | password reset        | <span>Planned</span> | <a href="#">Add comment</a> | ODS Sprint 1 |               |             |        | 24 May 2025 |

Figure 9: Tasks List

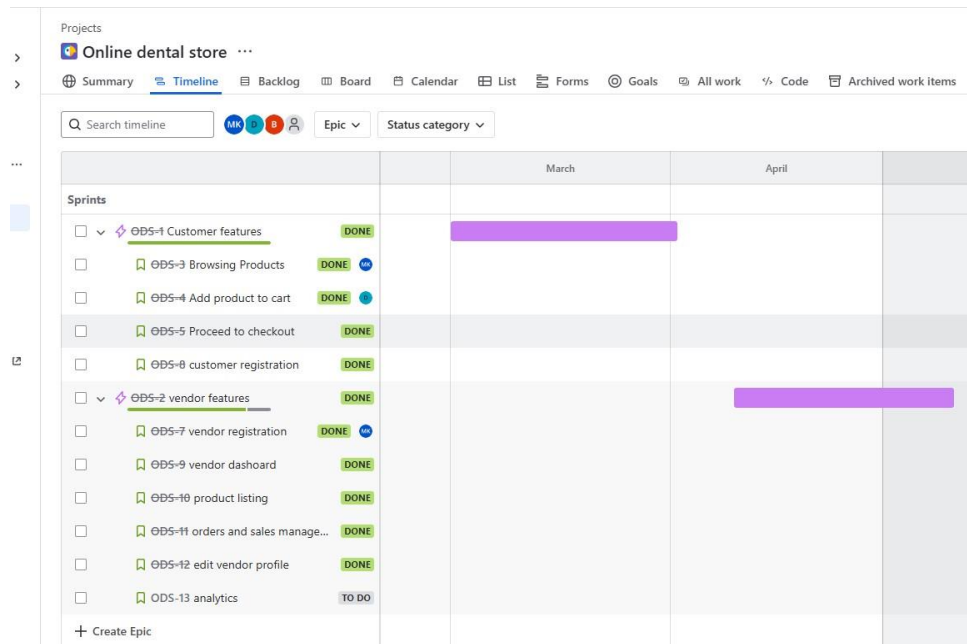


Figure 10: Project Time Line

### C. Linkdin video link:

<https://www.linkedin.com/in/momina-kiyani>

*THE END>>*