



VIT
CHENNAI



NOSQL DATABASES

PROJECT REPORT

FACULTY NAME: SIVAGAMI M

COURSE CODE: CSE3086

SLOT: C1

**TOPIC: SMART INVENTORY
MANAGEMENT**

SUBMITTED BY

SAI ESWARI P	22MIA1118
TEJASH KUMAR GS	22MIA1146
MOHAMED IDRIS Y	22MIA1005

1. INTRODUCTION

In current retail and supply chain dynamics, inventory management maintains its pivotal position throughout the business process from operations to customer satisfaction and eventually up to profitability. Still, the problem is that the majority of traditional systems are labor-intensive handling, depending on disconnected spreadsheets, estimates by humans, and as a result, they suffer from the inefficiencies that frequent stockouts, overstock, waste, and loss of income.

Motivation

The process of manual tracking is gradual and full of mistakes made by people. It is a common practice that the logs get updated by the workers at the end of their shifts, thus making the information not current.

- Apart from the fact that demand forecasting is often inaccurate, stockouts lead to losing sales and making customers unhappy.
- Overstocking also ties up the money in the form of goods that are not selling, which is especially problematic for products that have a limited shelf life.
- The inventory that has exceeded its expiration date is wasted and generates compliance problems, particularly in the grocery, pharma, and food industries.
- Because of the shortage of visibility, the management cannot easily know which items are not selling well, are nearing expiration, or are due for reorder.
- Hence, modern companies require data-driven systems that incorporate real-time tracking, expiry alerts, analytics, and AI-powered decision support.

What FreshFlow Offers

FreshFlow is a smart, web-based inventory management system designed to assist organizations in keeping stock levels optimal, cutting down waste, and making data-backed operational decisions.

The system comes with a variety of features, such as:

1. Automated inventory tracking with expiry monitoring.
2. Smart discount recommendations for near-expiry or slow-moving products.
3. Analytics dashboards for insights on waste and stock health.
4. Chatbot assistant for quick queries and actions.
5. Bulk CSV import for seamless data updates.
6. Role-based user management for security and accountability.

2. PROBLEM STATEMENT:

1. Manually carrying out the inventory process results in updating stock records frequently with errors and delays.
2. Absence of instantaneous visibility hinders the managers' understanding of which products are low, expiring, or overstocked.
3. Inaccurate demand forecasting and poor tracking lead to stockouts and overstocking.
4. Wasted due to the expiration of perishable goods, resulting in a financial loss and an increase in waste.
5. No alert systems or notifications for items that are about to expire or have very little stock left.
6. Decision-making is based on guessing since there are no data analytics or insights for trends and wastage.
7. Poor discount management, in which almost-expired products are neither recognized nor discounted in time.
8. No tracking of donated products, resulting in waste of consumable goods that could have been donated.
9. Time-consuming bulk updates, since traditional systems do not support CSV imports or automation.

3. Project Modules:

- In light of the project format and the source code, the FreshFlow application has the following modules which are the most important ones:

1. Authentication & User Management:

- Login and registration are securely performed with passwords that are bcrypt-hashed.
- Access to system functions is restricted based on user roles: the Admin, Manager, and Warehouse Staff.
- Depending on the permissions assigned, every user has the right to manage or see the modules.
- Management of sessions grants safe reach to the data that is sensitive.

2. Inventory Management:

1. The main module is responsible for the inventory items' adding, viewing, editing, and deleting.
2. It encodes such details as name, category, supplier, quantity, cost, expiry date, and creation date.
3. Moreover, it offers searching and filtering possibilities by category, supplier, or expiry status.
4. It allows the uploading of a whole inventory via CSV file to bring in or refresh stock in large numbers.
5. By real-time status classification the following are distinguished:
 - Fresh
 - Expiring Soon
 - Expired

3. CSV Import Module:

- This is a facility through which the inventory data can be bulk-uploaded by administrators via CSV files.
- Headers and data validated in a flexible manner allow for a seamless integration of the data.
- It automatically detects common date formats as well as quantity mismatches.
- The import summary is (total added, skipped, errors) is shown.

4. Expiry Alert System:

- It automatically identifies items that are going to expire or have already expired.
- This is done through classification into:
 1. Expiring in 3 days
 2. Expiring in 7 days
 3. Already expired
- It sends alerts to the dashboard to initiate measures (discount, donation, disposal).

5. Discount & Mid-Sale Management:

- The managers can indicate items for which limited-time discounts will be applied.
- It works out the new prices considering either the pre-established or AI-recommended percentage.
- This method helps to speedily sell off the items near the expiry date and thus, minimize wastage.
- The system has a separate module for monitoring all discounted products.

6. Smart Discount Recommendation (AI-driven) :

- Applies a combination of heuristic and data-driven rules:
- Products that have a shelf life of less than 7 days
- Items that are overstocked (with an amount of 50 or more)
- Products that have low sales and thus are moving slowly
- Dynamically proposes discount rates of 5% to 50%.
- Stores all the suggestions in the discount_recommendations collection for review.

7. Analytics and Reporting :

- Utilizes Chart.js to show visual charts for:
- Fresh, Expiring, and Expired stock ratio
- Percentage of waste per category
- Low-stock alerts
- Condenses the data in a way that it is very easy for the management to get insights from it and also to do trend analysis.

8. Order & Replenishment Module:

- Eases the creation of purchase orders for items that are running low in stock.
- Maintains a record of suppliers, ordered items, their quantities, and total costs.
- Permits restocking to be done in an efficient manner and supplier performance to be monitored.

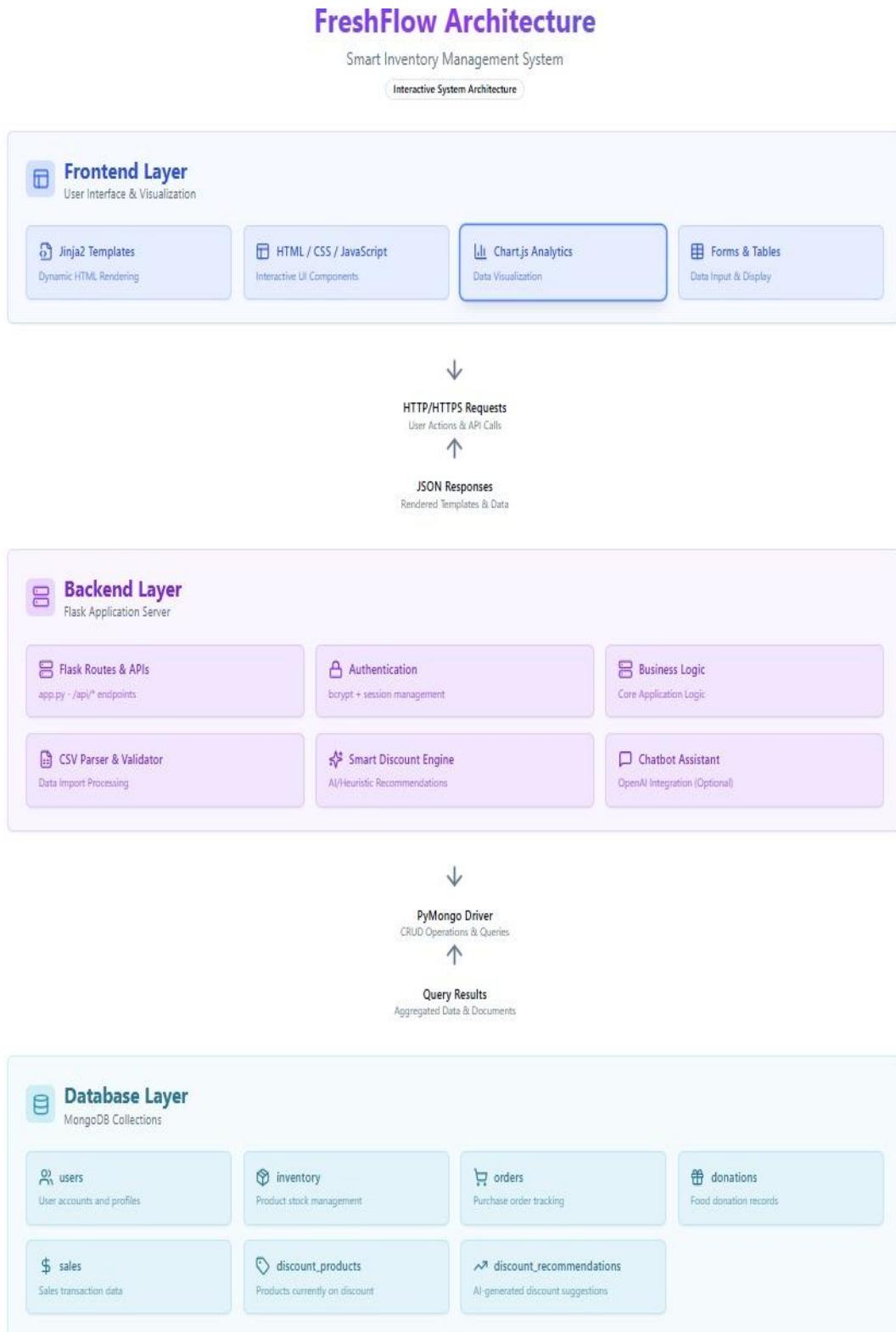
9. Donation Module:

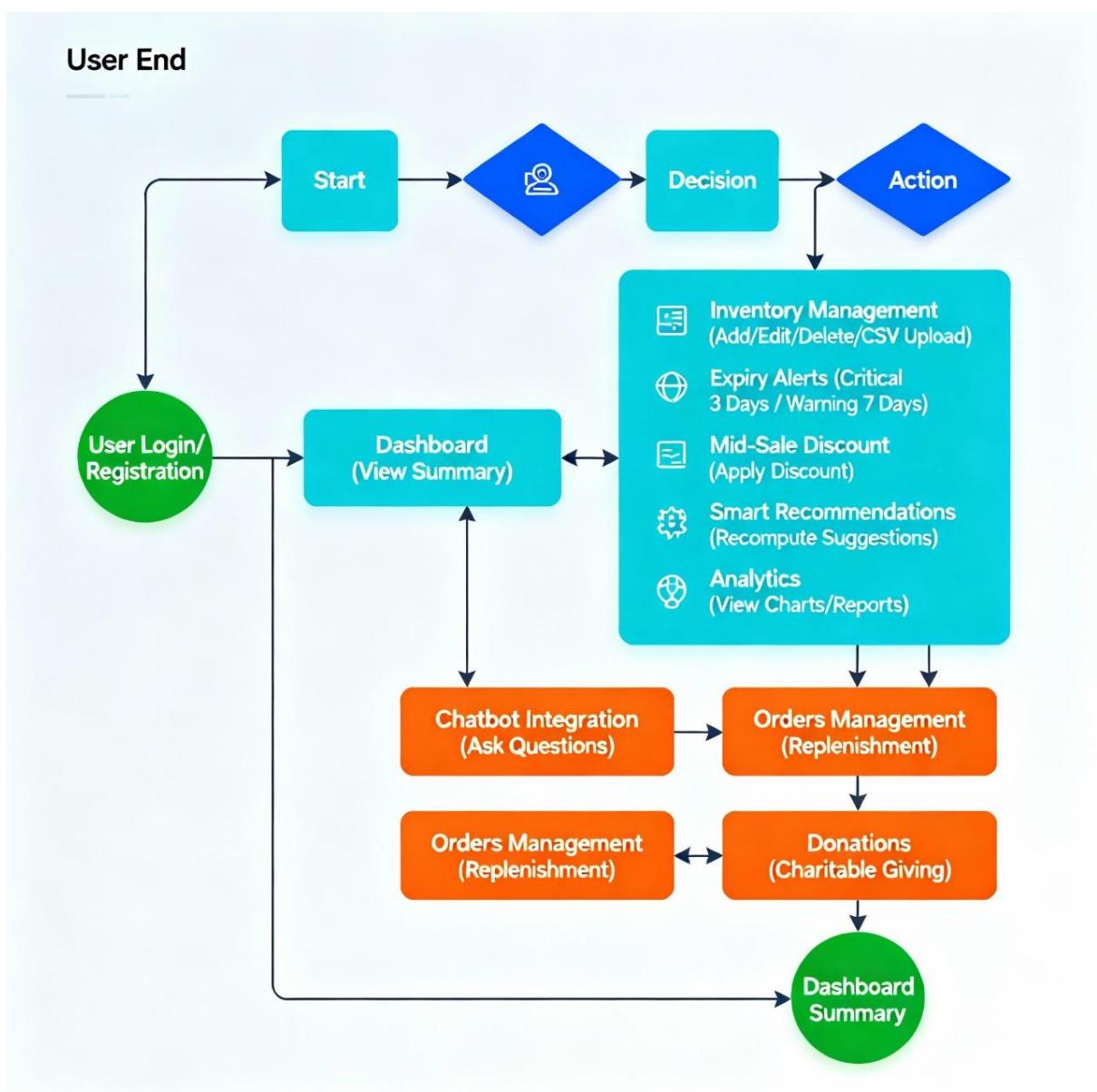
- Gives the choice to the organizations of donating items that are near their expiry date to charities rather than throwing them away.
- Keeps track of the amounts donated, the recipients, and the history of donations.
- Works towards waste reduction and supports the cause of promoting social responsibility.

10. Chatbot Assistant:

- Permits users to communicate through very basic text commands:
 - a. “Show expiring items”
 - b. “What is the quantity of milk?”
 - c. “Add 5 to rice stock”
 - d. “Show discount recommendations”
- Tracks and carries out fast updates and lookups without the necessity of UI navigation.
- Integrates OpenAI API optionally to clarify the responses by summarizing them.

4. ARCHITECTURE DIAGRAM:

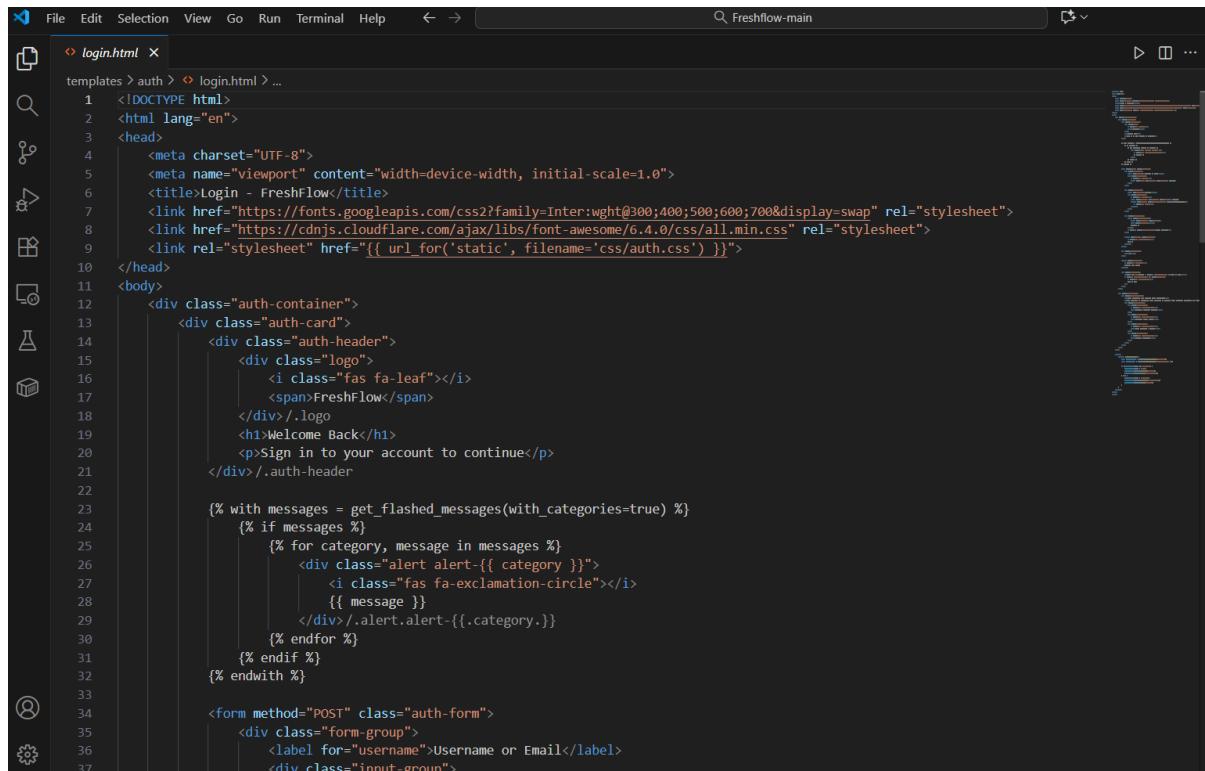




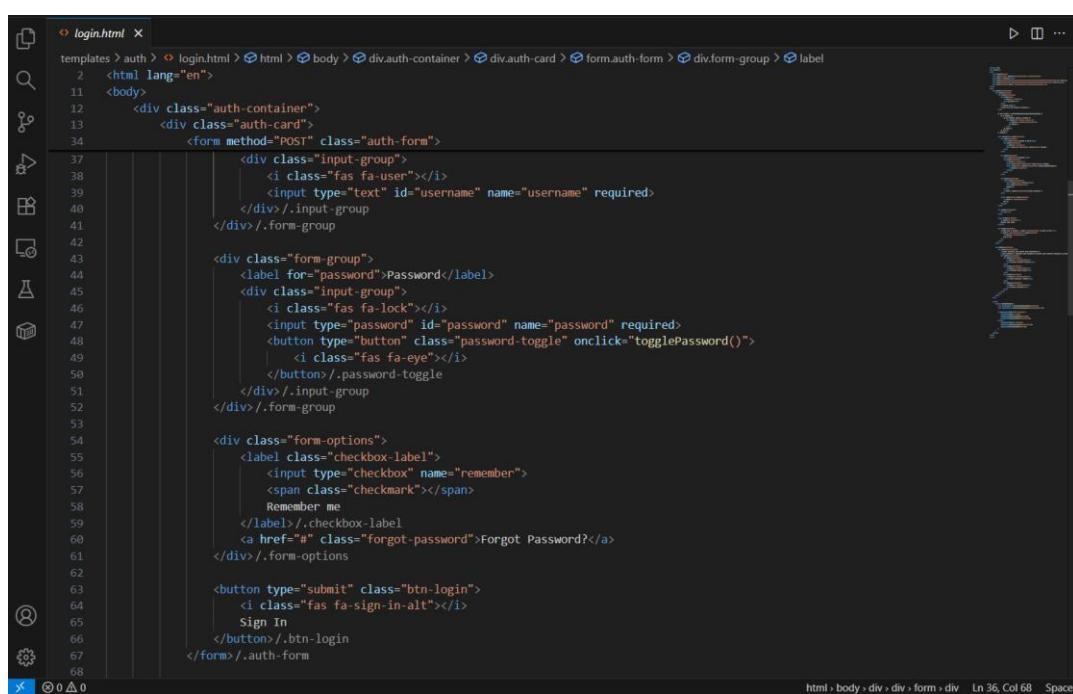
5.CODE & IMPLEMENTATION:

Authentication Page

Login.html



```
File Edit Selection View Go Run Terminal Help ← → 🔍 Freshflow-main
login.html ×
templates > auth > login.html ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Login - Freshflow</title>
7      <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">
8      <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet" rel="stylesheet">
9      <link rel="stylesheet" href="{{ url_for('static', filename='css/auth.css') }}>
10 </head>
11 <body>
12     <div class="auth-container">
13         <div class="auth-card">
14             <div class="auth-header">
15                 <div class="logo">
16                     <i class="fas fa-leaf"></i>
17                     <span>FreshFlow</span>
18                 </div>/.logo
19                 <h1>Welcome Back</h1>
20                 <p>Sign in to your account to continue</p>
21             </div>/.auth-header
22
23             {% with messages = get_flashed_messages(with_categories=true) %}
24             {% if messages %}
25                 {% for category, message in messages %}
26                     <div class="alert alert-{{ category }}">
27                         <i class="fas fa-exclamation-circle"></i>
28                         {{ message }}
29                     </div>/.alert.alert-{{ .category .}}
30                 {% endfor %}
31             {% endif %}
32             {% endwith %}
33
34             <form method="POST" class="auth-form">
35                 <div class="form-group">
36                     <label for="username">Username or Email</label>
37                     <div class="input-group">
```



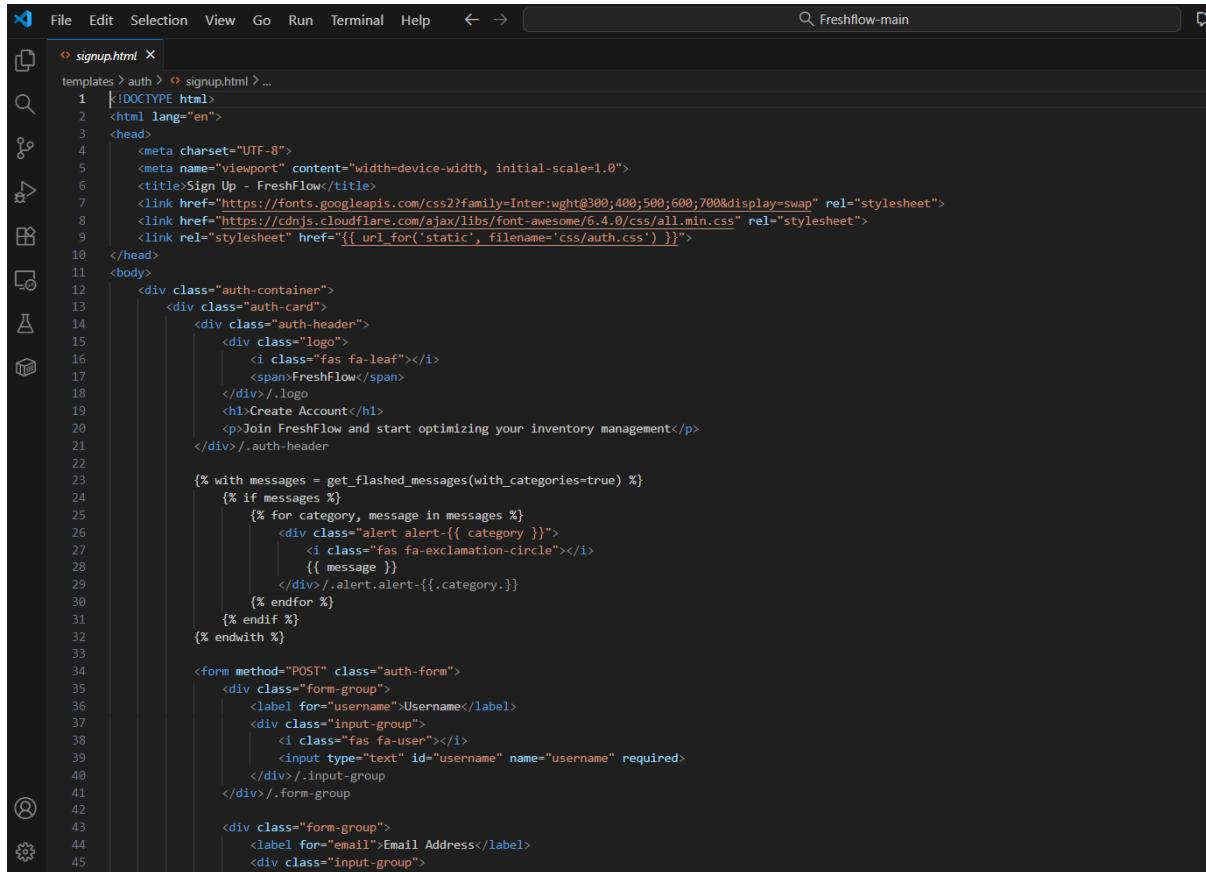
```
File Edit Selection View Go Run Terminal Help ← → 🔍 Freshflow-main
login.html ×
templates > auth > login.html > html > body > div.auth-container > div.auth-card > form.auth-form > div.form-group > label
2  <html lang="en">
11 <body>
12     <div class="auth-container">
13         <div class="auth-card">
14             <form method="POST" class="auth-form">
15                 <div class="input-group">
16                     <i class="fas fa-user"></i>
17                     <input type="text" id="username" name="username" required>
18                 </div>/.input-group
19
20                 <div class="form-group">
21                     <label for="password">Password</label>
22                     <div class="input-group">
23                         <i class="fas fa-lock"></i>
24                         <input type="password" id="password" name="password" required>
25                         <button type="button" class="password-toggle" onclick="togglePassword()">
26                             <i class="fas fa-eye"></i>
27                         </button>/.password-toggle
28                     </div>/.input-group
29
30                 <div class="form-options">
31                     <label class="checkbox-label">
32                         <input type="checkbox" name="remember">
33                         <span class="checkmark"></span>
34                         Remember me
35                     </label>/.checkbox-label
36                     <a href="#" class="forgot-password">Forgot Password?</a>
37                 </div>/.form-options
38
39                 <button type="submit" class="btn-login">
40                     <i class="fas fa-sign-in-alt"></i>
41                     Sign In
42                 </button>/.btn-login
43             </form>/.auth-form
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
```

html > body > div > div > form > div Ln 36, Col 68 Spaces:

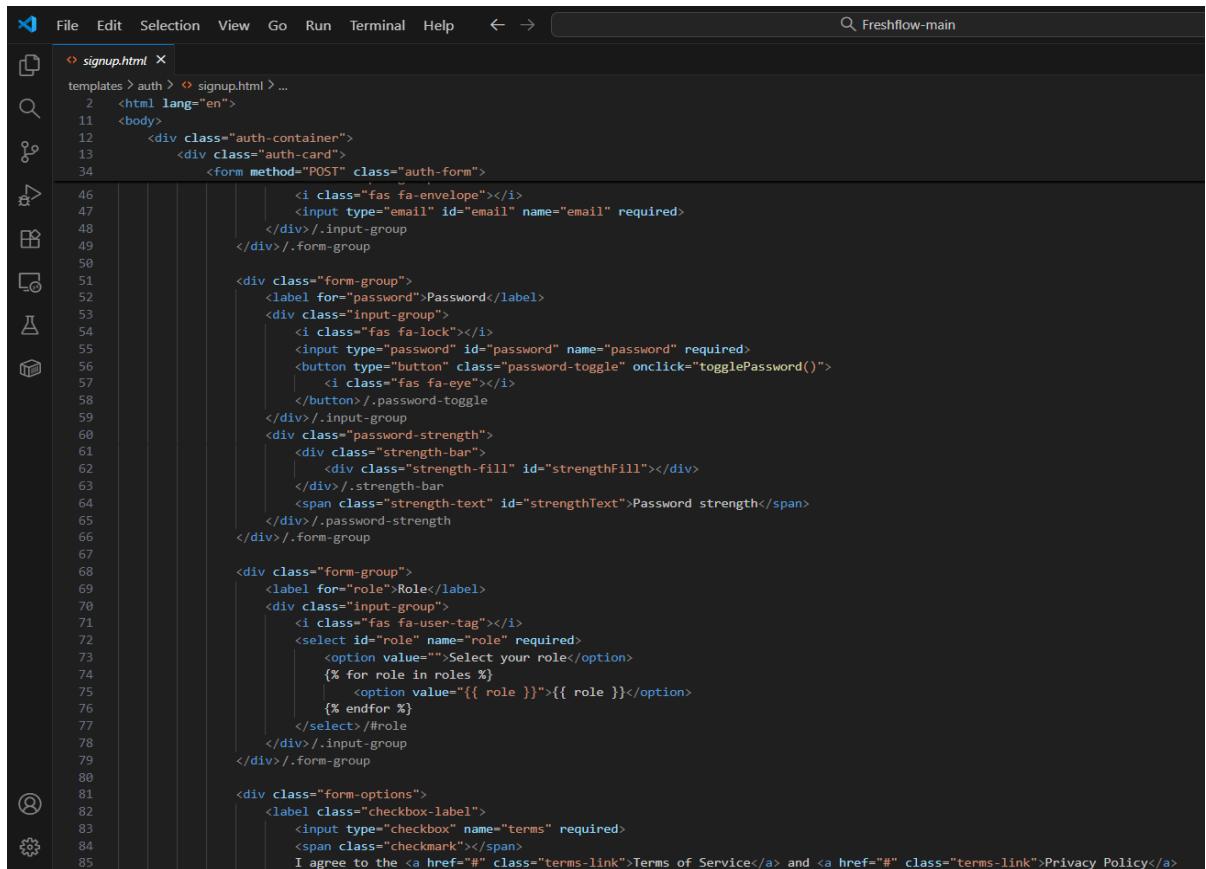
```
login.html x
templates > auth > login.html > html > body > div.auth-container > div.auth-card > form.auth-form > div.form-group > label
  2   <html lang="en">
  3   <body>
  4     <div class="auth-container">
  5       <div class="auth-card">
  6
  7         <div class="auth-divider">
  8           <span>or</span>
  9         </div>/.auth-divider
  10
  11         <button class="btn-google">
  12           <i class="fab fa-google"></i>
  13           Continue with Google
  14         </button>/.btn-google
  15
  16         <div class="auth-footer">
  17           <p>Don't have an account? <a href="{{ url_for('signup') }}>Sign up here</a></p>
  18           <a href="{{ url_for('landing') }}" class="back-home">
  19             <i class="fas fa-arrow-left"></i>
  20             Back to Home
  21           </a>/.back-home
  22         </div>/.auth-footer
  23       </div>/.auth-card
  24
  25       <div class="auth-sidebar">
  26         <div class="sidebar-content">
  27           <h2>Track Freshness. Save Wastage. Boost Efficiency.</h2>
  28           <p>Join thousands of businesses using FreshFlow to optimize their inventory management and reduce waste.</p>
  29           <div class="features-list">
  30             <div class="feature-item">
  31               <i class="fas fa-check-circle"></i>
  32               <span>Real-time inventory tracking</span>
  33             </div>/.feature-item
  34             <div class="feature-item">
  35               <i class="fas fa-check-circle"></i>
  36               <span>Automated expiry alerts</span>
  37             </div>/.feature-item
  38             <div class="feature-item">
  39
  40           </div>/.feature-item
  41         </div>/.features-list
  42       </div>/.auth-sidebar
  43     </div>/.auth-container
  44
  45   </body>
  46 </html>
```

```
File Edit Selection View Go Run Terminal Help ← → Q Freshflow-main
login.html x
templates > auth > login.html > html > body > div.auth-container > div.auth-card > form.auth-form > div.form-group > label
  2   <html lang="en">
  3   <body>
  4     <div class="auth-container">
  5       <div class="auth-sidebar">
  6         <div class="sidebar-content">
  7           <div class="feature-item">
  8             <i class="fas fa-check-circle"></i>
  9             <span>Smart analytics & reports</span>
 10           </div>/.feature-item
 11           <div class="feature-item">
 12             <i class="fas fa-check-circle"></i>
 13             <span>Donation management</span>
 14           </div>/.feature-item
 15         </div>/.features-list
 16       </div>/.auth-sidebar
 17     </div>/.auth-container
 18
 19   <script>
 20     function togglePassword() {
 21       const passwordInput = document.getElementById('password');
 22       const toggleButton = document.querySelector('.password-toggle i');
 23
 24       if (passwordInput.type === 'password') {
 25         passwordInput.type = 'text';
 26         toggleButton.classList.remove('fa-eye');
 27         toggleButton.classList.add('fa-eye-slash');
 28       } else {
 29         passwordInput.type = 'password';
 30         toggleButton.classList.remove('fa-eye-slash');
 31         toggleButton.classList.add('fa-eye');
 32       }
 33     }
 34   </script>
 35
 36   </body>
 37 </html>
```

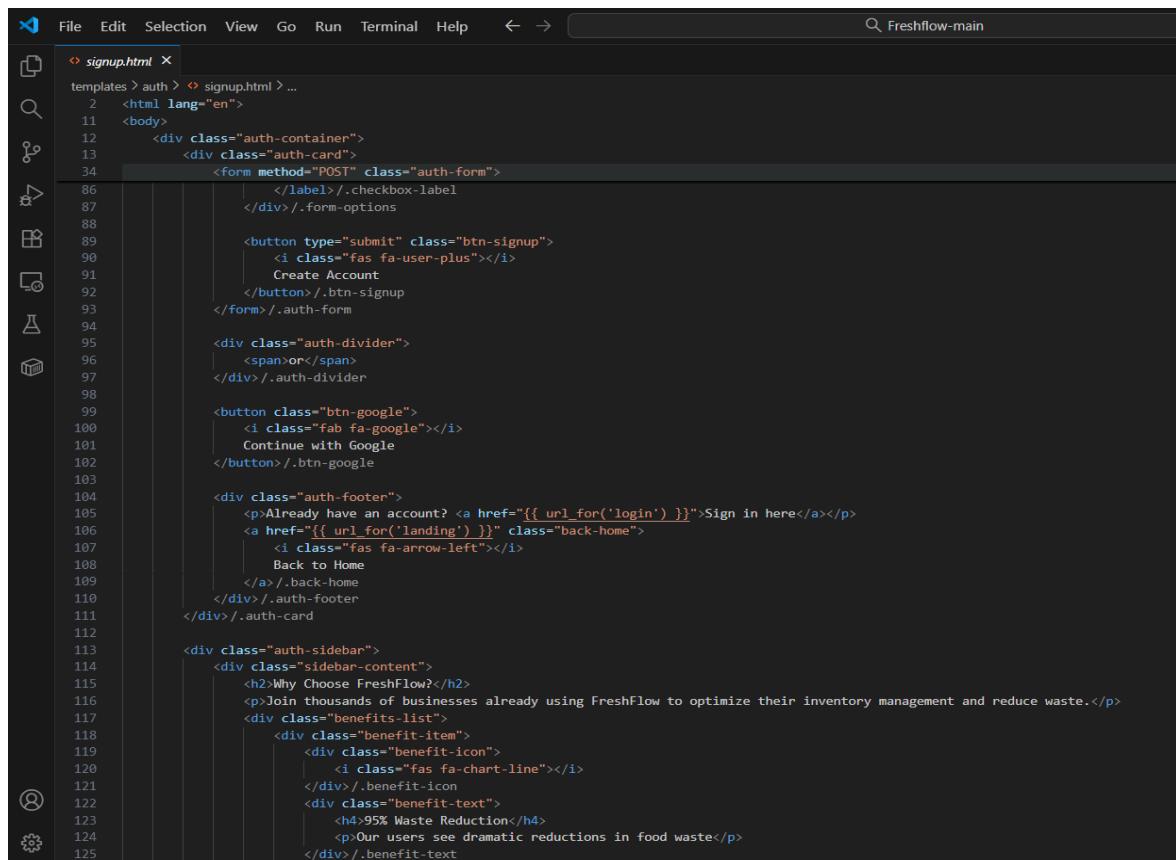
signup.html



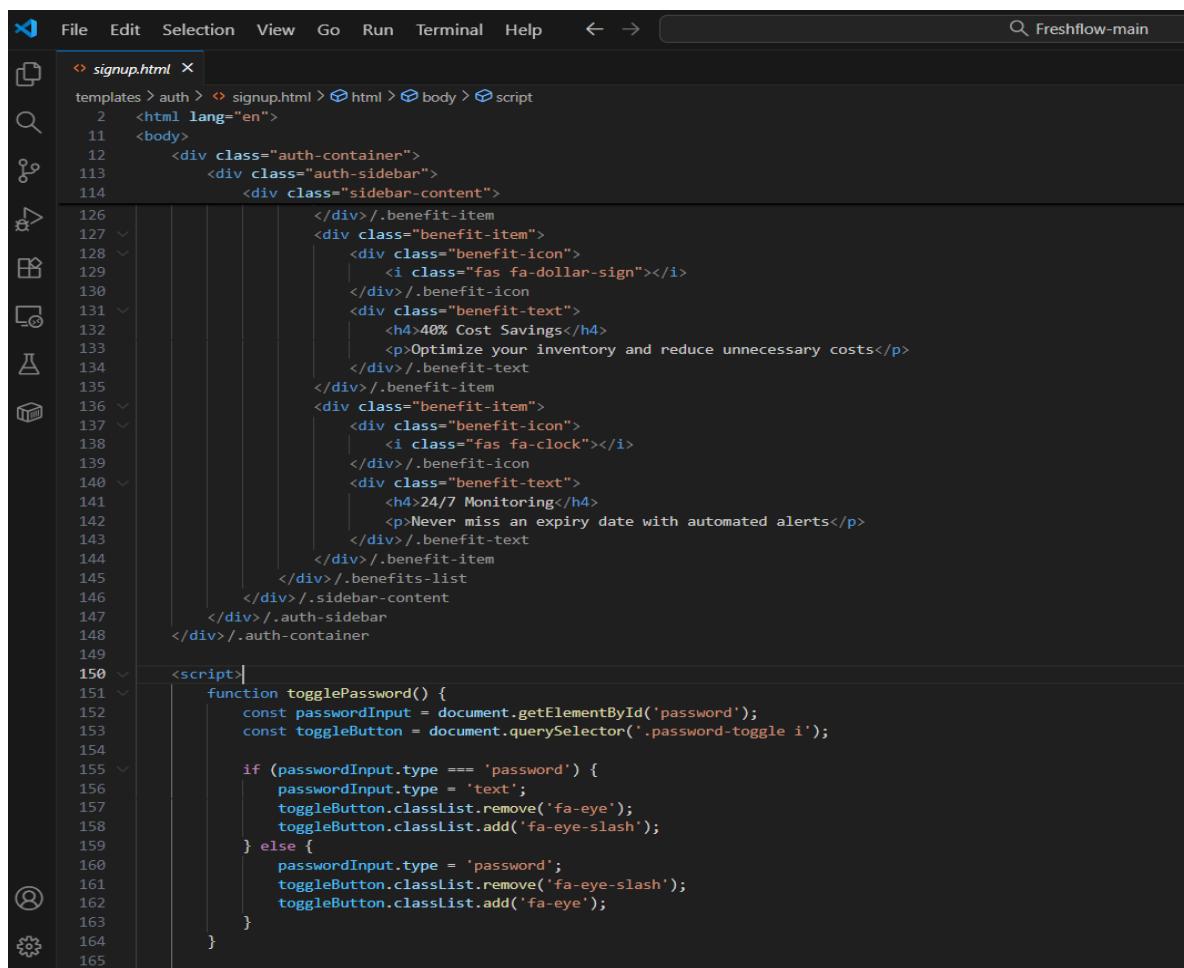
```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ Freshflow-main
templates > auth > signup.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Sign Up - FreshFlow</title>
7      <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">
8      <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
9      <link rel="stylesheet" href="{{ url_for('static', filename='css/auth.css') }}>
10 </head>
11 <body>
12     <div class="auth-container">
13         <div class="auth-card">
14             <div class="auth-header">
15                 <div class="logo">
16                     <i class="fas fa-leaf"></i>
17                     <span>FreshFlow</span>
18                 </div>/.logo
19                 <h1>Create Account</h1>
20                 <p>Join Freshflow and start optimizing your inventory management</p>
21             </div>/.auth-header
22
23             {% with messages = get_flashed_messages(with_categories=true) %}
24             {% if messages %}
25                 {% for category, message in messages %}
26                     <div class="alert alert-{% category %}">
27                         <i class="fas fa-exclamation-circle"></i>
28                         {{ message }}
29                     </div>/.alert.alert-{%.category.%}
30                 {% endfor %}
31             {% endif %}
32         {% endwith %}
33
34         <form method="POST" class="auth-form">
35             <div class="form-group">
36                 <label for="username">Username</label>
37                 <div class="input-group">
38                     <i class="fas fa-user"></i>
39                     <input type="text" id="username" name="username" required>
40                 </div>/.input-group
41             </div>/.form-group
42
43             <div class="form-group">
44                 <label for="email">Email Address</label>
45                 <div class="input-group">
```



```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ Freshflow-main
templates > auth > signup.html > ...
2  <html lang="en">
11 <body>
12     <div class="auth-container">
13         <div class="auth-card">
14             <form method="POST" class="auth-form">
46             <i class="fas fa-envelope"></i>
47             <input type="email" id="email" name="email" required>
48             </div>/.input-group
49         </div>/.form-group
50
51         <div class="form-group">
52             <label for="password">Password</label>
53             <div class="input-group">
54                 <i class="fas fa-lock"></i>
55                 <input type="password" id="password" name="password" required>
56                 <button type="button" class="password-toggle" onclick="togglePassword()">
57                     <i class="fas fa-eye"></i>
58                 </button>/.password-toggle
59             </div>/.input-group
60             <div class="password-strength">
61                 <div class="strength-bar">
62                     <div class="strength-fill" id="strengthFill"></div>
63                 </div>/.strength-bar
64                 <span class="strength-text" id="strengthText">Password strength</span>
65             </div>/.password-strength
66         </div>/.form-group
67
68         <div class="form-group">
69             <label for="role">Role</label>
70             <div class="input-group">
71                 <i class="fas fa-user-tag"></i>
72                 <select id="role" name="role" required>
73                     <option value="">Select your role</option>
74                     {% for role in roles %}
75                         <option value="{{ role }}>{{ role }}</option>
76                     {% endfor %}
77                 </select>/.#role
78             </div>/.input-group
79         </div>/.form-group
80
81         <div class="form-options">
82             <label class="checkbox-label">
83                 <input type="checkbox" name="terms" required>
84                 <span class="checkboxmark"></span>
85                 I agree to the <a href="#" class="terms-link">Terms of Service</a> and <a href="#" class="privacy-link">Privacy Policy</a>
86         </div>/.form-options
87     </div>/.auth-card
88 </body>
```



```
File Edit Selection View Go Run Terminal Help < > Freshflow-main
signup.html x
templates > auth > signup.html > ...
2 <html lang="en">
11 <body>
12   <div class="auth-container">
13     <div class="auth-card">
14       <form method="POST" class="auth-form">
15         <label>Email</label>
16         <input type="text" class="auth-input" />
17         <label>Password</label>
18         <input type="password" class="auth-input" />
19         <div class="checkbox-label">
20           <input checked="" type="checkbox" /> I agree to the
21             <div>Terms of Service</div> and <div>Privacy Policy</div>
22         </div>
23         <button type="submit" class="btn-signup">
24           <i class="fas fa-user-plus"></i>
25           Create Account
26         </button>
27       </form>
28     </div>
29   </div>
30
31   <div class="auth-divider">
32     <span>OR</span>
33   </div>
34
35   <button class="btn-google">
36     <i class="fab fa-google"></i>
37     Continue with Google
38   </button>
39
40   <div class="auth-footer">
41     <p>Already have an account? <a href="{{ url_for('login') }}>Sign in here</a></p>
42     <a href="{{ url_for('landing') }}" class="back-home">
43       <i class="fas fa-arrow-left"></i>
44       Back to Home
45     </a>
46   </div>
47
48 </div>
49
50 <div class="auth-sidebar">
51   <div class="sidebar-content">
52     <h2>Why Choose Freshflow?</h2>
53     <p>Join thousands of businesses already using FreshFlow to optimize their inventory management and reduce waste.</p>
54     <div class="benefits-list">
55       <div class="benefit-item">
56         <div class="benefit-icon">
57           <i class="fas fa-chart-line"></i>
58         </div>
59         <div class="benefit-text">
60           <h4>95% Waste Reduction</h4>
61           <p>Our users see dramatic reductions in food waste</p>
62         </div>
63       </div>
64       <div class="benefit-item">
65         <div class="benefit-icon">
66           <i class="fas fa-dollar-sign"></i>
67         </div>
68         <div class="benefit-text">
69           <h4>40% Cost Savings</h4>
70           <p>Optimize your inventory and reduce unnecessary costs</p>
71         </div>
72       </div>
73       <div class="benefit-item">
74         <div class="benefit-icon">
75           <i class="fas fa-clock"></i>
76         </div>
77         <div class="benefit-text">
78           <h4>24/7 Monitoring</h4>
79           <p>Never miss an expiry date with automated alerts</p>
80         </div>
81       </div>
82     </div>
83   </div>
84 </div>
85
86 <script>
87   function togglePassword() {
88     const passwordInput = document.getElementById('password');
89     const toggleButton = document.querySelector('.password-toggle i');
90
91     if (passwordInput.type === 'password') {
92       passwordInput.type = 'text';
93       toggleButton.classList.remove('fa-eye');
94       toggleButton.classList.add('fa-eye-slash');
95     } else {
96       passwordInput.type = 'password';
97       toggleButton.classList.remove('fa-eye-slash');
98       toggleButton.classList.add('fa-eye');
99     }
100   }
101 </script>
```



```
File Edit Selection View Go Run Terminal Help < > Freshflow-main
signup.html x
templates > auth > signup.html > ...
2 <html lang="en">
11 <body>
12   <div class="auth-container">
13     <div class="auth-sidebar">
14       <div class="sidebar-content">
15         <div class="benefit-item">
16           <div class="benefit-icon">
17             <i class="fas fa-dollar-sign"></i>
18           </div>
19           <div class="benefit-text">
20             <h4>40% Cost Savings</h4>
21             <p>Optimize your inventory and reduce unnecessary costs</p>
22           </div>
23         </div>
24         <div class="benefit-item">
25           <div class="benefit-icon">
26             <i class="fas fa-clock"></i>
27           </div>
28           <div class="benefit-text">
29             <h4>24/7 Monitoring</h4>
30             <p>Never miss an expiry date with automated alerts</p>
31           </div>
32         </div>
33       </div>
34     </div>
35   </div>
36
37 <script>
38   function togglePassword() {
39     const passwordInput = document.getElementById('password');
40     const toggleButton = document.querySelector('.password-toggle i');
41
42     if (passwordInput.type === 'password') {
43       passwordInput.type = 'text';
44       toggleButton.classList.remove('fa-eye');
45       toggleButton.classList.add('fa-eye-slash');
46     } else {
47       passwordInput.type = 'password';
48       toggleButton.classList.remove('fa-eye-slash');
49       toggleButton.classList.add('fa-eye');
50     }
51   }
52 </script>
```

The screenshot shows a code editor interface with a dark theme. The top bar includes standard menu items: File, Edit, Selection, View, Go, Run, Terminal, Help, and navigation arrows. A search bar on the right contains the text "Freshflow-main". The left sidebar features various icons for file operations like copy, paste, find, and refresh. The main pane displays the content of the file "signup.html".

```
File Edit Selection View Go Run Terminal Help ← → 🔍 Freshflow-main
signup.html ✎
templates > auth > signup.html > html > body > script
  2   <html lang="en">
  11  <body>
  150 <script>
  167  document.getElementById('password').addEventListener('input', function() {
  174
  175      if (password.length >= 8) strength++;
  176      if (/^[a-z]+$/i.test(password)) strength++;
  177      if (/^[A-Z]+$/i.test(password)) strength++;
  178      if (/^[0-9]+$/i.test(password)) strength++;
  179      if (/^[\w\W]+$/i.test(password)) strength++;
  180
  181      switch (strength) {
  182          case 0:
  183          case 1:
  184              strengthLabel = 'Very Weak';
  185              strengthFill.style.width = '20%';
  186              strengthFill.style.backgroundColor = '#e74c3c';
  187              break;
  188          case 2:
  189              strengthLabel = 'Weak';
  190              strengthFill.style.width = '40%';
  191              strengthFill.style.backgroundColor = '#f39c12';
  192              break;
  193          case 3:
  194              strengthLabel = 'Fair';
  195              strengthFill.style.width = '60%';
  196              strengthFill.style.backgroundColor = '#f1c40f';
  197              break;
  198          case 4:
  199              strengthLabel = 'Good';
  200              strengthFill.style.width = '80%';
  201              strengthFill.style.backgroundColor = '#27ae60';
  202              break;
  203          case 5:
  204              strengthLabel = 'Strong';
  205              strengthFill.style.width = '100%';
  206              strengthFill.style.backgroundColor = '#27ae60';
  207              break;
  208      }
  209
  210      strengthText.textContent = strengthLabel;
  211  });
  212 </script>
  213 </body>
  214 </html>
```

Dashboard Page

Add_item.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Item - Freshflow</title>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
</head>
<body>
    <div class="dashboard-container">
        <aside class="sidebar">
            <div class="sidebar-header">
                <div class="logo">
                    <i class="fas fa-leaf"></i>
                    <span>FreshFlow</span>
                </div>/.logo
            </div>/.sidebar-header
            <nav class="sidebar-nav">
                <ul class="nav-list">
                    <li class="nav-item"><a href="{{ url_for('dashboard') }}" class="nav-link"><i class="fas fa-home"></i><span>Dashboard</span></a></li>
                    <li class="nav-item active"><a href="{{ url_for('inventory') }}" class="nav-link"><i class="fas fa-boxes"></i><span>Inventory</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('expiry_alerts') }}" class="nav-link"><i class="fas fa-bell"></i><span>Expiry Alerts</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('analytics') }}" class="nav-link"><i class="fas fa-chart-line"></i><span>Analytics</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('orders') }}" class="nav-link"><i class="fas fa-shopping-cart"></i><span>Orders</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('donations') }}" class="nav-link"><i class="fas fa-heart"></i><span>Donations</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('settings') }}" class="nav-link"><i class="fas fa-cog"></i><span>Settings</span></a></li>
                </ul>/.nav-list
            </nav>/.sidebar-nav
            <div class="sidebar-footer">
                <div class="user-info">
                    <div class="user-avatar"><i class="fas fa-user"></i></div>
                    <div class="user-details"><span class="user-name">{{ session.username }}</span><span class="user-role">Admin</span></div>
                </div>/.user-info
                <a href="{{ url_for('logout') }}" class="logout-btn"><i class="fas fa-sign-out-alt"></i></a>
            </div>/.sidebar-footer
        </aside>/.sidebar
        <main class="main-content">
            <header class="top-nav">
                <div class="nav-left">
                    <button class="sidebar-toggle"><i class="fas fa-bars"></i></button>
                    <h1 class="page-title">Add Inventory Item</h1>
                </div>/.nav-left
            </header>/.top-nav
            <div class="nav-right">
                <a href="{{ url_for('inventory') }}" class="btn-secondary"><i class="fas fa-arrow-left"></i> Back</a>
            </div>/.nav-right
        </header>/.top-nav
        <div class="dashboard-content">
            <div class="settings-card">
                <div class="card-header">
                    <h3>New Item Details</h3>
                </div>/.card-header
                <div class="card-content">
                    <form method="POST" action="{{ url_for('add_item') }}" class="form-grid">
                        <div class="form-group">
                            <label for="name">Item Name</label>
                            <input type="text" id="name" name="name" required>
                        </div>/.form-group
                        <div class="form-group">
                            <label for="category">Category</label>
                            <select id="category" name="category" required>
                                <option value="">Select category</option>
                                {% for c in categories %}
                                    <option value="{{ c }}">{{ c }}</option>
                                {% endfor %}
                            </select>/.category
                        </div>/.form-group
                        <div class="form-group">
                            <label for="quantity">Quantity</label>
                            <input type="number" id="quantity" name="quantity" min="0" required>
                        </div>/.form-group
                        <div class="form-group">
                            <label for="expiry_date">Expiry Date</label>
                            <input type="date" id="expiry_date" name="expiry_date" required>
                        </div>/.form-group
                        <div class="form-group">
                            <label for="supplier">Supplier</label>
                            <input type="text" id="supplier" name="supplier" placeholder="Optional">
                        </div>/.form-group
                        <div class="form-group">
                            <label for="cost">Cost</label>
                            <input type="number" id="cost" name="cost" step="0.01" min="0" placeholder="0.00">
                        </div>/.form-group
                    </form>/.card-content
                </div>/.settings-card
            </div>/.dashboard-content
        </main>/.main-content
    </div>/.dashboard-container
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<body>
    <div class="dashboard-container">
        <main class="main-content">
            <header class="top-nav">
                <div class="nav-right">
                    <a href="{{ url_for('inventory') }}" class="btn-secondary"><i class="fas fa-arrow-left"></i> Back</a>
                </div>/.nav-right
            </header>/.top-nav
            <div class="dashboard-content">
                <div class="settings-card">
                    <div class="card-header">
                        <h3>New Item Details</h3>
                    </div>/.card-header
                    <div class="card-content">
                        <form method="POST" action="{{ url_for('add_item') }}" class="form-grid">
                            <div class="form-group">
                                <label for="name">Item Name</label>
                                <input type="text" id="name" name="name" required>
                            </div>/.form-group
                            <div class="form-group">
                                <label for="category">Category</label>
                                <select id="category" name="category" required>
                                    <option value="">Select category</option>
                                    {% for c in categories %}
                                        <option value="{{ c }}">{{ c }}</option>
                                    {% endfor %}
                                </select>/.category
                            </div>/.form-group
                            <div class="form-group">
                                <label for="quantity">Quantity</label>
                                <input type="number" id="quantity" name="quantity" min="0" required>
                            </div>/.form-group
                            <div class="form-group">
                                <label for="expiry_date">Expiry Date</label>
                                <input type="date" id="expiry_date" name="expiry_date" required>
                            </div>/.form-group
                            <div class="form-group">
                                <label for="supplier">Supplier</label>
                                <input type="text" id="supplier" name="supplier" placeholder="Optional">
                            </div>/.form-group
                            <div class="form-group">
                                <label for="cost">Cost</label>
                                <input type="number" id="cost" name="cost" step="0.01" min="0" placeholder="0.00">
                            </div>/.form-group
                        </form>/.card-content
                    </div>/.settings-card
                </div>/.dashboard-content
            </main>/.main-content
        </div>/.dashboard-container
    </body>
</html>
```

```

<!-- add_item.html -->
templates > dashboard > <-- add_item.html > ...
2   <html lang="en">
3     <body>
4       <div class="dashboard-container">
5         <main class="main-content">
6           <div class="dashboard-content">
7             <div class="form-group">
8               <label for="category">Category</label>
9               <select id="category" name="category" required>
10                 <option value="">Select category</option>
11                 {% for c in categories %}<option value="{{ c }}">{{ c }}</option>
12                 {% endfor %}
13             </select> //category
14           </div>/.form-group
15           <div class="form-group">
16             <label for="quantity">Quantity</label>
17             <input type="number" id="quantity" name="quantity" min="0" required>
18           </div>/.form-group
19           <div class="form-group">
20             <label for="expiry_date">Expiry Date</label>
21             <input type="date" id="expiry_date" name="expiry_date" required>
22           </div>/.form-group
23           <div class="form-group">
24             <label for="supplier">Supplier</label>
25             <input type="text" id="supplier" name="supplier" placeholder="Optional">
26           </div>/.form-group
27           <div class="form-group">
28             <label for="cost">Cost</label>
29             <input type="number" id="cost" name="cost" step="0.01" min="0" placeholder="0.00">
30           </div>/.form-group
31           <div class="form-actions">
32             <button type="submit" class="btn-primary"><i class="fas fa-save"></i> Save Item</button>
33           </div>/.form-actions
34         </div>/.form-grid
35       </div>/.card-content
36     </div>/.settings-card
37   </div>/.dashboard-content
38 </main>/.main-content
39 </div>/.dashboard-container
40
41   <script src="{{ url_for('static', filename='js/dashboard.js') }}"/></script>
42 </body>
43 </html>
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

Analytics page

Analytics.html

```

<!-- analytics.html -->
templates > dashboard > <-- analytics.html > ...
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Analytics - FreshFlow</title>
7       <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">
8       <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
9       <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}"/>
10      <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
11    </head>
12    <body>
13      <div class="dashboard-container">
14        <!-- Sidebar -->
15        <aside class="sidebar">
16          <div class="sidebar-header">
17            <div class="logo">
18              <i class="fas fa-leaf"></i>
19              <span>FreshFlow</span>
20            </div>/logo
21          </div>/.sidebar-header
22
23          <nav class="sidebar-nav">
24            <ul class="nav-list">
25              <li class="nav-item">
26                <a href="{{ url_for('dashboard') }}" class="nav-link">
27                  <i class="fas fa-home"></i>
28                  <span>Dashboard</span>
29                </a>/nav-link
30              </li>/.nav-item
31              <li class="nav-item">
32                <a href="{{ url_for('inventory') }}" class="nav-link">
33                  <i class="fas fa-boxes"></i>
34                  <span>Inventory</span>
35                </a>/nav-link
36              </li>/.nav-item
37              <li class="nav-item">
38                <a href="{{ url_for('expiry_alerts') }}" class="nav-link">
39                  <i class="fas fa-bell"></i>
40                  <span>Expiry Alerts</span>
41                </a>/nav-link
42              </li>/.nav-item
43              <li class="nav-item active">
44                <a href="{{ url_for('analytics') }}" class="nav-link">
45                  <i class="fas fa-chart-line"></i>
46                </a>/nav-link
47              </li>/.nav-item
48            </ul>/.nav-list
49          </nav>/.sidebar
50        </aside>/.sidebar
51      </div>/.dashboard-container
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
99
100
101

```

The screenshot shows a code editor window with the file 'analytics.html' open. The code is a template for a dashboard sidebar. It includes a navigation bar with links to 'orders', 'donations', and 'settings'. Below the navigation is a user info section displaying the username and role ('Admin'). A logout button is also present. The code uses Bootstrap classes like 'fa-chart-line', 'fa-shopping-cart', 'fa-heart', 'fa-cog', and 'fa-user'.

```
<html lang="en">
<body>
    <div class="dashboard-container">
        <aside class="sidebar">
            <nav class="sidebar-nav">
                <i class="fas fa-chart-line"></i>
                <span>Analytics</span>
            </a>/.nav-link
            <li class="nav-item active">
                <a href="{{ url_for('orders') }}" class="nav-link">
                    <i class="fas fa-shopping-cart"></i>
                    <span>Orders</span>
                </a>/.nav-link
            </li>/.nav-item
            <li class="nav-item">
                <a href="{{ url_for('donations') }}" class="nav-link">
                    <i class="fas fa-heart"></i>
                    <span>Donations</span>
                </a>/.nav-link
            </li>/.nav-item
            <li class="nav-item">
                <a href="{{ url_for('settings') }}" class="nav-link">
                    <i class="fas fa-cog"></i>
                    <span>Settings</span>
                </a>/.nav-link
            </li>/.nav-item
            </ul>/.nav-list
        </nav>/.sidebar-nav

        <div class="sidebar-footer">
            <div class="user-info">
                <div class="user-avatar">
                    <i class="fas fa-user"></i>
                </div>/.user-avatar
                <div class="user-details">
                    <span class="user-name">{{ session.username }}</span>
                    <span class="user-role">Admin</span>
                </div>/.user-details
            </div>/.user-info
            <a href="{{ url_for('logout') }}" class="logout-btn">
                <i class="fas fa-sign-out-alt"></i>
            </a>/.logout-btn
        </div>/.sidebar-footer
    </aside>/.sidebar
</body>
</html>
```

The screenshot shows a code editor window with the file 'analytics.html' open. The code includes a script block containing functions for updating charts based on a date range and exporting a report. It also includes a style block with CSS rules for a '.kpi-grid' grid and '.kpi-card' cards, including gradients and shadows.

```
<script>
    // Update charts based on date range
    function updateCharts() {
        const dateRange = document.getElementById('dateRange').value;
        // In a real app, this would fetch new data based on the date range
        console.log('Updating charts for date range:', dateRange);
    }

    // Export report
    function exportReport() {
        // In a real app, this would generate and download a PDF/Excel report
        alert('Export functionality would generate a comprehensive analytics report');
    }
</script>

<style>
    /* Additional styles for analytics page */
    .kpi-grid {
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
        gap: 1.5rem;
        margin-bottom: 2rem;
    }

    .kpi-card {
        background: var(--white);
        border-radius: 12px;
        padding: 1.5rem;
        box-shadow: var(--shadow);
        border: 1px solid var(--border-color);
        display: flex;
        align-items: center;
        gap: 1rem;
    }

    .kpi-icon {
        width: 60px;
        height: 60px;
        background: linear-gradient(135deg, var(--calm-blue), #3498db);
        border-radius: 12px;
        display: flex;
        justify-content: center;
        align-items: center;
        font-size: 24px;
        color: white;
    }
</style>
```

The screenshot shows a code editor interface with a dark theme. The title bar reads "File Edit Selection View Go Run Terminal Help" and "Freshflow-main". The left sidebar contains icons for file operations like Open, Save, Find, and others. The main area displays the following CSS code:

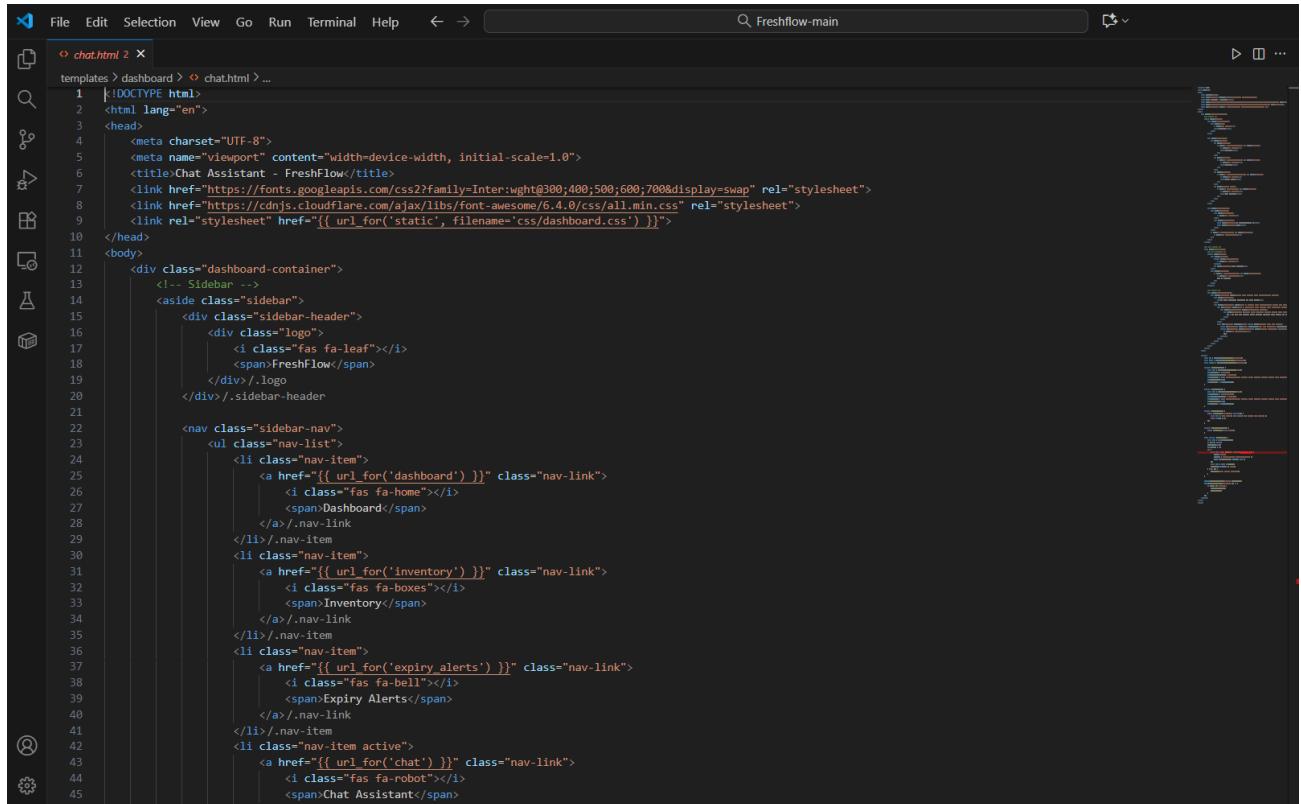
```
templates > dashboard > analytics.html > ...
2   <html lang="en">
12  <body>
513    <style>
543      }
544
545      .kpi-content {
546        flex: 1;
547      }
548
549      .kpi-value {
550        font-size: 2rem;
551        font-weight: 700;
552        color: var(--text-dark);
553        margin-bottom: 0.25rem;
554      }
555
556      .kpi-label {
557        font-size: 0.9rem;
558        color: var(--text-light);
559        margin-bottom: 0.5rem;
560      }
561
562      .kpi-change {
563        font-size: 0.8rem;
564        display: flex;
565        align-items: center;
566        gap: 0.25rem;
567      }
568
569      .kpi-change.positive {
570        color: var(--fresh-green);
571      }
572
573      .kpi-change.negative {
574        color: #e74c3c;
575      }
576
577      .charts-grid {
578        display: grid;
579        grid-template-columns: 2fr 1fr;
580        gap: 1.5rem;
581        margin-bottom: 2rem;
582      }
583
584      .chart-card.large {
```

The screenshot shows a code editor interface with a dark theme. The title bar reads "File Edit Selection View Go Run Terminal Help" and "Freshflow-main". The left sidebar contains icons for file operations like Open, Save, Find, and others. The main area displays the following CSS code:

```
templates > dashboard > analytics.html > ...
2   <html lang="en">
12  <body>
513    <style>
649      }
650
651      .trend.neutral {
652        color: var(--text-light);
653      }
654
655      .trend.negative {
656        color: #e74c3c;
657      }
658
659      .recommendations-card {
660        background: var(--white);
661        border-radius: 12px;
662        box-shadow: var(--shadow);
663        border: 1px solid var(--border-color);
664        overflow: hidden;
665      }
666
667      .recommendation-badge {
668        background: linear-gradient(135deg, var(--fresh-green), #229954);
669        color: var(--white);
670        padding: 0.25rem 0.75rem;
671        border-radius: 12px;
672        font-size: 0.8rem;
673        font-weight: 500;
674      }
675
676      .recommendations-list {
677        display: flex;
678        flex-direction: column;
679        gap: 1rem;
680      }
681
682      .recommendation-item {
683        display: flex;
684        align-items: flex-start;
685        gap: 1rem;
686        padding: 1rem;
687        background: var(--light-gray);
688        border-radius: 8px;
689      }
```

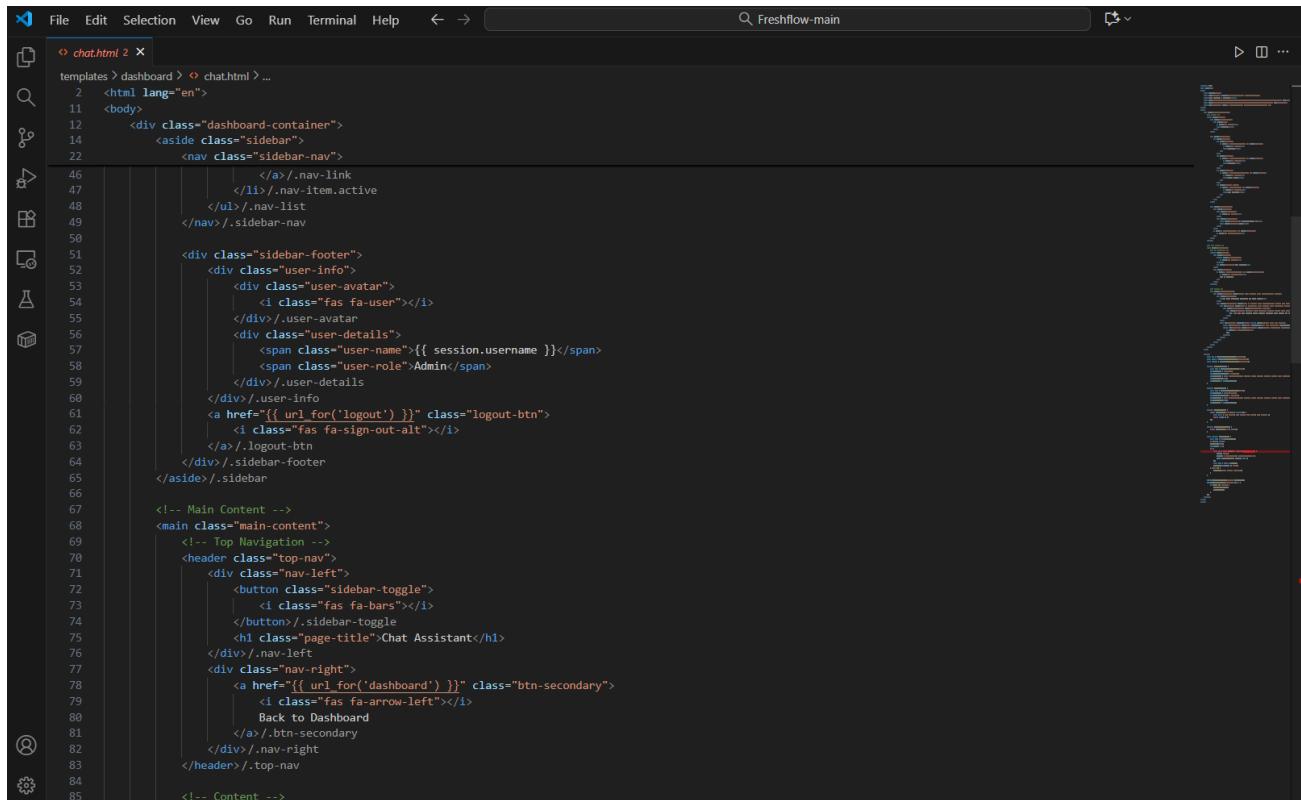
Chat Page

Chat.html



```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ 🔍 Freshflow-main

chat.html 2 x
templates > dashboard > chat.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Chat Assistant - FreshFlow</title>
7      <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;700&display=swap" rel="stylesheet">
8      <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
9      <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
10 </head>
11 <body>
12     <div class="dashboard-container">
13         <!-- Sidebar -->
14         <aside class="sidebar">
15             <div class="sidebar-header">
16                 <div class="logo">
17                     <i class="fas fa-leaf"></i>
18                     <span>FreshFlow</span>
19                 </div>/.
20             </div>/.
21             <nav class="sidebar-nav">
22                 <ul class="nav-list">
23                     <li class="nav-item">
24                         <a href="{{ url_for('dashboard') }}" class="nav-link">
25                             <i class="fas fa-home"></i>
26                             <span>Dashboard</span>
27                         </a>/.
28                     </li>/.
29                     <li class="nav-item">
30                         <a href="{{ url_for('inventory') }}" class="nav-link">
31                             <i class="fas fa-boxes"></i>
32                             <span>Inventory</span>
33                         </a>/.
34                     </li>/.
35                     <li class="nav-item">
36                         <a href="{{ url_for('expiry_alerts') }}" class="nav-link">
37                             <i class="fas fa-bell"></i>
38                             <span>Expiry Alerts</span>
39                         </a>/.
40                     </li>/.
41                     <li class="nav-item active">
42                         <a href="{{ url_for('chat') }}" class="nav-link">
43                             <i class="fas fa-robot"></i>
44                             <span>Chat Assistant</span>
45                         </a>/.
46                 </li>/.
47             </ul>/.
48         </nav>/.
49     </aside>/.
50 
```



```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ 🔍 Freshflow-main

chat.html 2 x
templates > dashboard > chat.html > ...
2  <html lang="en">
11 <body>
12     <div class="dashboard-container">
13         <aside class="sidebar">
14             <nav class="sidebar-nav">
15                 <ul class="nav-list">
16                     <li class="nav-item active">
17                         <a href="{{ url_for('chat') }}" class="nav-link">
18                             <i class="fas fa-robot"></i>
19                             <span>Chat Assistant</span>
20                         </a>/.
21                     </li>/.
22                 </ul>/.
23             </nav>/.
24             <div class="sidebar-footer">
25                 <div class="user-info">
26                     <div class="user-avatar">
27                         <i class="fas fa-user"></i>
28                     </div>/.
29                     <div class="user-details">
30                         <span class="user-name">{{ session.username }}</span>
31                         <span class="user-role">Admin</span>
32                     </div>/.
33                     <div class="user-info">
34                         <a href="{{ url_for('logout') }}" class="logout-btn">
35                             <i class="fas fa-sign-out-alt"></i>
36                         </a>/.
37                     </div>/.
38                 </div>/.
39             </div>/.
40         </aside>/.
41 
42         <!-- Main Content -->
43         <main class="main-content">
44             <!-- Top Navigation -->
45             <header class="top-nav">
46                 <div class="nav-left">
47                     <button class="sidebar-toggle">
48                         <i class="fas fa-bars"></i>
49                     </button>/.
50                     <h1 class="page-title">Chat Assistant</h1>
51                 </div>/.
52                 <div class="nav-right">
53                     <a href="{{ url_for('dashboard') }}" class="btn-secondary">
54                         <i class="fas fa-arrow-left"></i>
55                         Back to Dashboard
56                     </a>/.
57                 </div>/.
58             </header>/.
59         <!-- Content -->
60 
```

```

File Edit Selection View Go Run Terminal Help ← → ⌘ Freshflow-main
chat.html 2
templates > dashboard > chat.html ...
1 <html lang="en">
2   <body>
3     <script>
4       function appendBot(text) {
5         ...
6       }
7
8       function escapeHTML(str) {
9         return str.replace(/[\&<\>]/g, function(s) {
10           const map = { '&': '&amp;', '<': '&lt;', '>': '&gt;', '"'': '&quot;' };
11           return map[s] || s;
12         });
13     }
14
15     function formatMultiline(str) {
16       return str.replace(/\n/g, '<br>');
17     }
18
19     async function sendMessage() {
20       const text = input.value.trim();
21       if (!text) return;
22       appendUser(text);
23       input.value = '';
24       try {
25         const res = await fetch('{{ url_for('api_chat') }}', {
26           method: 'POST',
27           headers: { 'Content-Type': 'application/json' },
28           body: JSON.stringify({ message: text })
29         });
30         const data = await res.json();
31         appendBot(data.reply || '...');
32       } catch (e) {
33         appendBot(`Error sending message.`);
34       }
35     }
36
37     sendBtn.addEventListener('click', sendMessage);
38     input.addEventListener('keydown', (e) => {
39       if (e.key === 'Enter') {
40         e.preventDefault();
41         sendMessage();
42       }
43     });
44   </body>
45 </html>

```

Chatbot Page

Chatbot.html

```

File Edit Selection View Go Run Terminal Help ← → ⌘ Freshflow-main
chatbot.html 2
templates > dashboard > chatbot.html ...
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Chatbot - Freshflow</title>
7     <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">
8     <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
9     <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
10  </head>
11  <body>
12    <div class="dashboard-container">
13      <aside class="sidebar">
14        <div class="sidebar-header">
15          <div class="logo">
16            <i class="fas fa-leaf"></i>
17            <span>Freshflow</span>
18          </div>/.logo
19        </div>/.sidebar-header
20        <nav class="sidebar-nav">
21          <ul class="nav-list">
22            <li class="nav-item"><a href="{{ url_for('dashboard') }}" class="nav-link"><i class="fas fa-home"></i><span>Dashboard</span></a></li>
23            <li class="nav-item"><a href="{{ url_for('inventory') }}" class="nav-link"><i class="fas fa-boxes"></i><span>Inventory</span></a></li>
24            <li class="nav-item"><a href="{{ url_for('expiry_alerts') }}" class="nav-link"><i class="fas fa-bell"></i><span>Expiry Alerts</span></a></li>
25            <li class="nav-item active"><a href="{{ url_for('chatbot_page') }}" class="nav-link"><i class="fas fa-robot"></i><span>Chatbot</span></a></li>
26          </ul>/.nav-list
27        </nav>/.sidebar-nav
28        <div class="sidebar-footer">
29          <div class="user-info">
30            <div class="user-avatar"><i class="fas fa-user"></i></div>
31            <div class="user-details">
32              <span class="user-name">{{ session.username }}</span>
33              <span class="user-role">Admin</span>
34            </div>/.user-details
35            <a href="{{ url_for('logout') }}" class="logout-btn"><i class="fas fa-sign-out-alt"></i></a>
36          </div>/.sidebar-footer
37        </aside>/.sidebar
38
39        <main class="main-content">
40          <header class="top-nav">
41            <div class="nav-left">
42              <button class="sidebar-toggle"><i class="fas fa-bars"></i></button>
43              <h1 class="page-title">Chatbot Assistant</h1>
44            </div>/.nav-left
45        </main>/.main-content

```

```
File Edit Selection View Go Run Terminal Help ↶ ↷ Freshflow-main

chatbot.html 2 x
templates > dashboard > chatbot.html > ...
2   <html lang="en">
11  <body>
12    <div class="dashboard-container">
40      <main class="main-content">
41        <header class="top-nav">
46          <div class="nav-right">
47            <a href="{{ url_for('dashboard') }}><i class="fas fa-arrow-left"></i> Back to Dashboard</a>
48          </div>/.nav-right
49        </header>/.top-nav
50
51        <div class="dashboard-content">
52          <div class="table-card" style="height: 75vh; display: flex; flex-direction: column;">
53            <div class="card-header">
54              <h3>Ask about inventory, discounts, stock, or update quantities.</h3>
55            </div>/.card-header
56            <div class="card-content" style="flex: 1; display: flex; flex-direction: column; gap: 0.75rem;">
57              <div id="chat-log" style="flex: 1; overflow-y: auto; padding: 1rem; background: var(--light-gray); border-radius: 8px;">
58                <div class="assistant-msg" style="margin-bottom: 0.75rem;">
59                  <div style="background: #eef9ff; color: #f1f8dc; padding: 0.75rem 1rem; border-radius: 8px; display: inline-block; max-width: 80%;">
60                    Hi! I can help with expiring items, discount products, stock levels, and adjusting quantities. Try asking: "Which products are expiring soon?"</div>
61                </div>/.assistant-msg
62              </div>/.#chat-log
63            <form id="chat-form" onsubmit="return false;" style="display: flex; gap: 0.5rem;">
64              <input id="chat-input" type="text" placeholder="Type your message..." autocomplete="off" style="flex: 1; padding: 0.75rem; border: 1px solid #f1f8dc; border-radius: 8px;">
65              <button id="chat-send" class="btn-primary" style="display: inline-flex; align-items: center; gap: 0.5rem;">
66                Send
67                <i class="fas fa-paper-plane"></i>
68              </button>/.#chat-send.btn-primary
69            </form>/.#chat-form
70          </div>/.table-card
71        </div>/.dashboard-content
72      </main>/.main-content
73    </div>/.dashboard-container
74
75    <script>
76      const log = document.getElementById('chat-log');
77      const input = document.getElementById('chat-input');
78      const sendBtn = document.getElementById('chat-send');
79
80      function appendUser(text) {
81        const wrap = document.createElement('div');
82        wrap.className = 'user-msg';
83        wrap.style.marginBottom = '0.75rem';
84        wrap.innerHTML = text;
85        log.appendChild(wrap);
86      }
87
88      function escapeHtml(str) {
89        return (str || '').replace(/(&lt;|&gt;|&quot;|&#39;|&amp;)/g, function(s) {
90          const map = { '&lt;': '<', '&gt;': '>', '&quot;': '"', '&#39;': '' };
91          return map[s] || s;
92        });
93
94      }
95
96      function formatMultiline(str) {
97        return (str || '').replace(/\n/g, '<br>');
98      }
99
100     function sendMessage() {
101       const text = (input.value || '').trim();
102       if (!text) return;
103       appendUser(text);
104       input.value = '';
105       try {
106         const res = await fetch('{{ url_for('chatbot_message') }}', {
107           method: 'POST',
108           headers: { 'Content-Type': 'application/json' },
109           body: JSON.stringify({ message: text })
110         });
111         const data = await res.json();
112         appendBot(data.reply || '...');

... (remaining code from line 112 to 139)

```

```
File Edit Selection View Go Run Terminal Help ↶ ↷ Freshflow-main

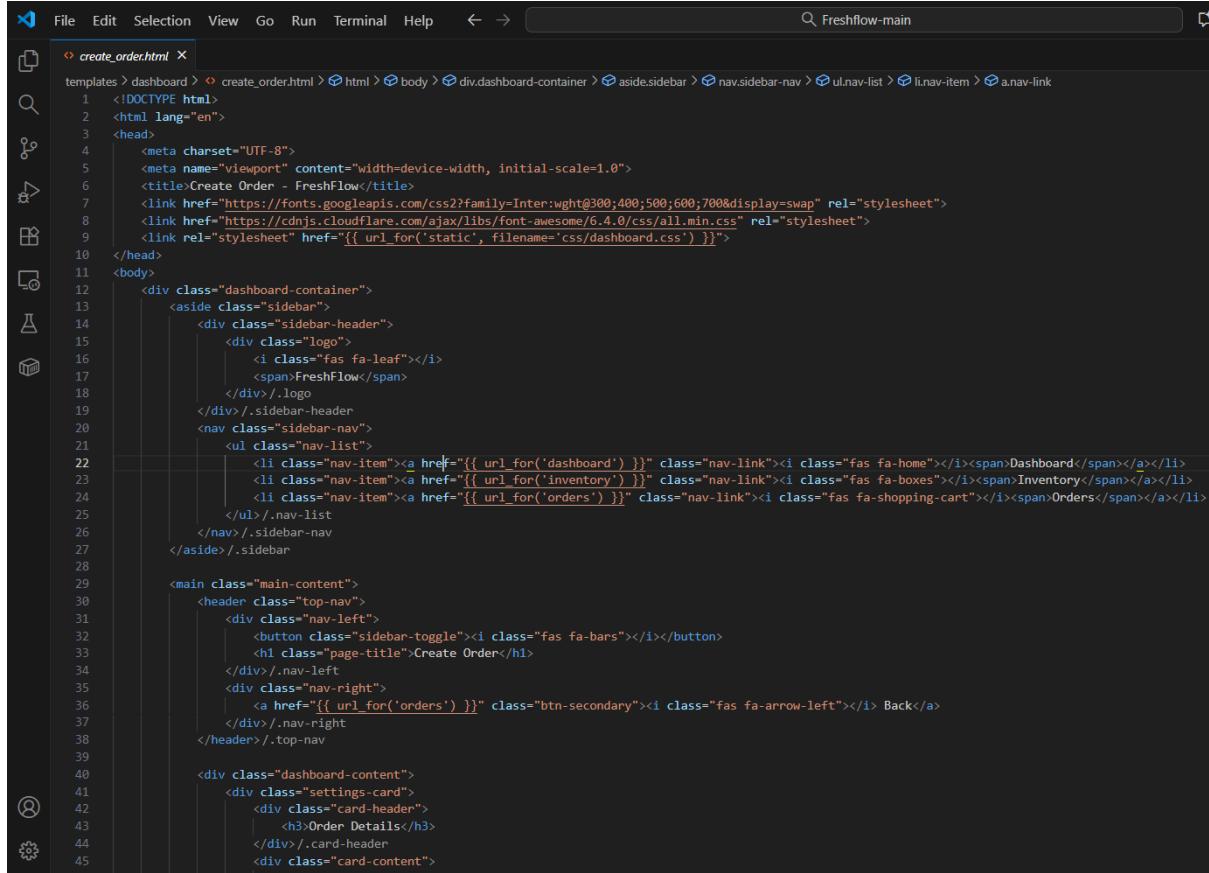
chatbot.html 2 x
templates > dashboard > chatbot.html > ...
2   <html lang="en">
11  <body>
77    <script>
78      ...
79
80      function escapeHtml(str) {
81        return (str || '').replace(/(&lt;|&gt;|&quot;|&#39;|&amp;)/g, function(s) {
82          const map = { '&lt;': '<', '&gt;': '>', '&quot;': '"', '&#39;': '' };
83          return map[s] || s;
84        });
85
86      }
87
88      function formatMultiline(str) {
89        return (str || '').replace(/\n/g, '<br>');
90      }
91
92      async function sendMessage() {
93        const text = (input.value || '').trim();
94        if (!text) return;
95        appendUser(text);
96        input.value = '';
97        try {
98          const res = await fetch('{{ url_for('chatbot_message') }}', {
99            method: 'POST',
100            headers: { 'Content-Type': 'application/json' },
101            body: JSON.stringify({ message: text })
102          });
103          const data = await res.json();
104          appendBot(data.reply || '...');

... (remaining code from line 104 to 139)

```

Order Creation Page

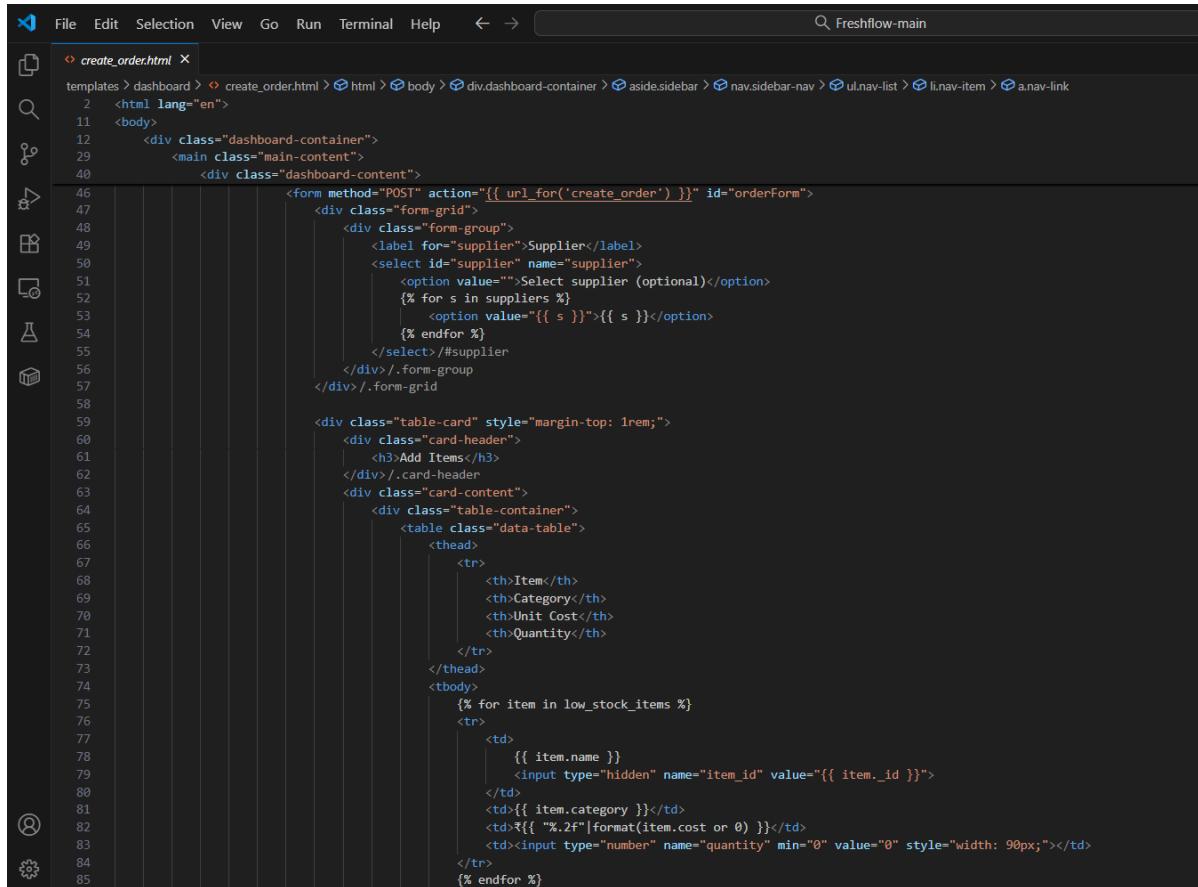
Order.html



A screenshot of a code editor showing the `Order.html` template. The code is a Jinja2 template for a dashboard. It includes a header with a logo and navigation links for Dashboard, Inventory, and Orders. The main content area displays a card titled "Order Details" with a table showing low-stock items.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Create Order - FreshFlow</title>
        <link href="https://fonts.googleapis.com/css?family=Inter:wght@300;400;500;700&display=swap" rel="stylesheet">
        <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
        <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
    </head>
    <body>
        <div class="dashboard-container">
            <aside class="sidebar">
                <div class="sidebar-header">
                    <div class="Logo">
                        <i class="fas fa-leaf"></i>
                        <span>FreshFlow</span>
                    </div>/.Logo
                </div>/.sidebar-header
                <nav class="sidebar-nav">
                    <ul class="nav-list">
                        <li class="nav-item"><a href="{{ url_for('dashboard') }}" class="nav-link"><i class="fas fa-home"></i><span>Dashboard</span></a></li>
                        <li class="nav-item"><a href="{{ url_for('inventory') }}" class="nav-link"><i class="fas fa-boxes"></i><span>Inventory</span></a></li>
                        <li class="nav-item"><a href="{{ url_for('orders') }}" class="nav-link"><i class="fas fa-shopping-cart"></i><span>Orders</span></a></li>
                    </ul>/.nav-list
                </nav>/.sidebar-nav
            </aside>/.sidebar
            <main class="main-content">
                <header class="top-nav">
                    <div class="nav-left">
                        <button class="sidebar-toggle"><i class="fas fa-bars"></i></button>
                        <h1 class="page-title">Create Order</h1>
                    </div>/.nav-left
                    <div class="nav-right">
                        <a href="{{ url_for('orders') }}" class="btn-secondary"><i class="fas fa-arrow-left"></i> Back</a>
                    </div>/.nav-right
                </header>/.top-nav
                <div class="dashboard-content">
                    <div class="settings-card">
                        <div class="card-header">
                            <h3>Order Details</h3>
                        </div>/.card-header
                        <div class="card-content">

```



A screenshot of a code editor showing the `create_order.html` template. This template is a form for creating a new order. It includes a dropdown for selecting a supplier and a table for adding items. The table has columns for Item, Category, Unit Cost, and Quantity.

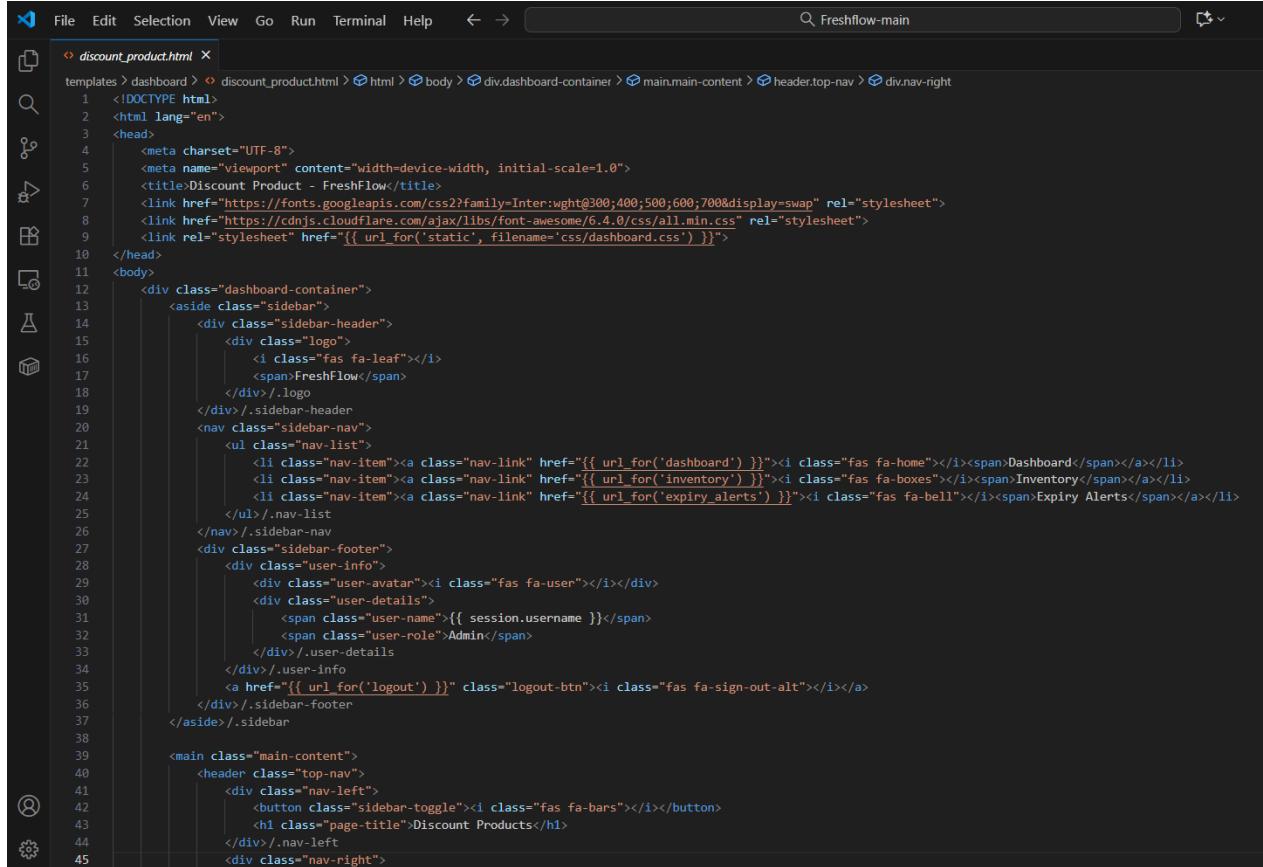
```
<form method="POST" action="{{ url_for('create_order') }}" id="orderForm">
    <div class="form-grid">
        <div class="form-group">
            <label for="supplier">Supplier</label>
            <select id="supplier" name="supplier">
                <option value="">Select supplier (optional)</option>
                {% for s in suppliers %}
                    <option value="{{ s }}>{{ s }}</option>
                {% endfor %}
            </select>#supplier
        </div>/.form-group
    </div>/.form-grid

    <div class="table-card" style="margin-top: 1rem;">
        <div class="card-header">
            <h3>Add Items</h3>
        </div>/.card-header
        <div class="card-content">
            <div class="table-container">
                <table class="data-table">
                    <thead>
                        <tr>
                            <th>Item</th>
                            <th>Category</th>
                            <th>Unit Cost</th>
                            <th>Quantity</th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for item in low_stock_items %}
                            <tr>
                                <td>
                                    {{ item.name }}
                                    <input type="hidden" name="item_id" value="{{ item._id }}>
                                </td>
                                <td>{{ item.category }}</td>
                                <td>{{ "%2f" | format(item.cost or 0) }}</td>
                                <td><input type="number" name="quantity" min="0" value="0" style="width: 90px;"></td>
                            </tr>
                        {% endfor %}
                    </tbody>
                </table>
            </div>
        </div>
    </div>

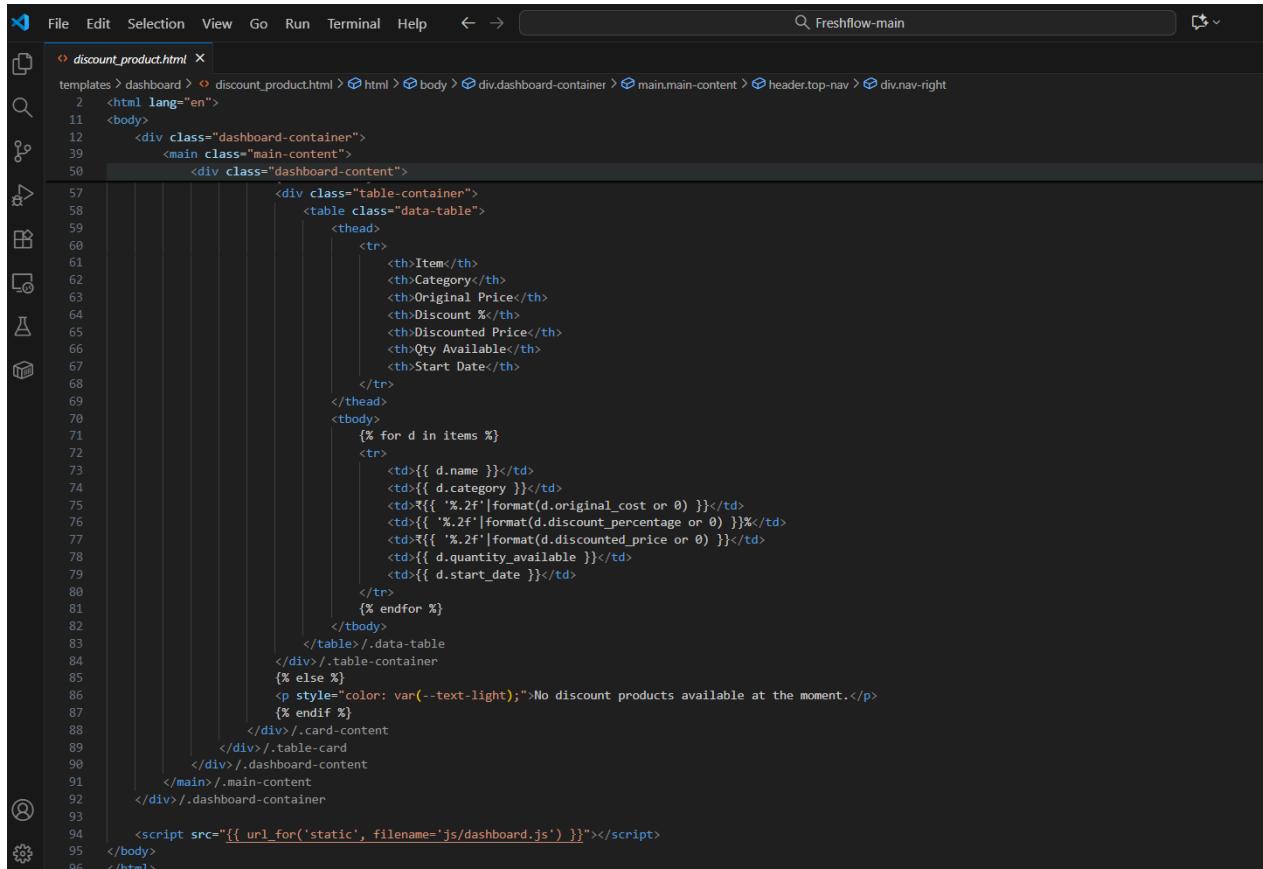
```

Product Discount Page

Discount_product.html



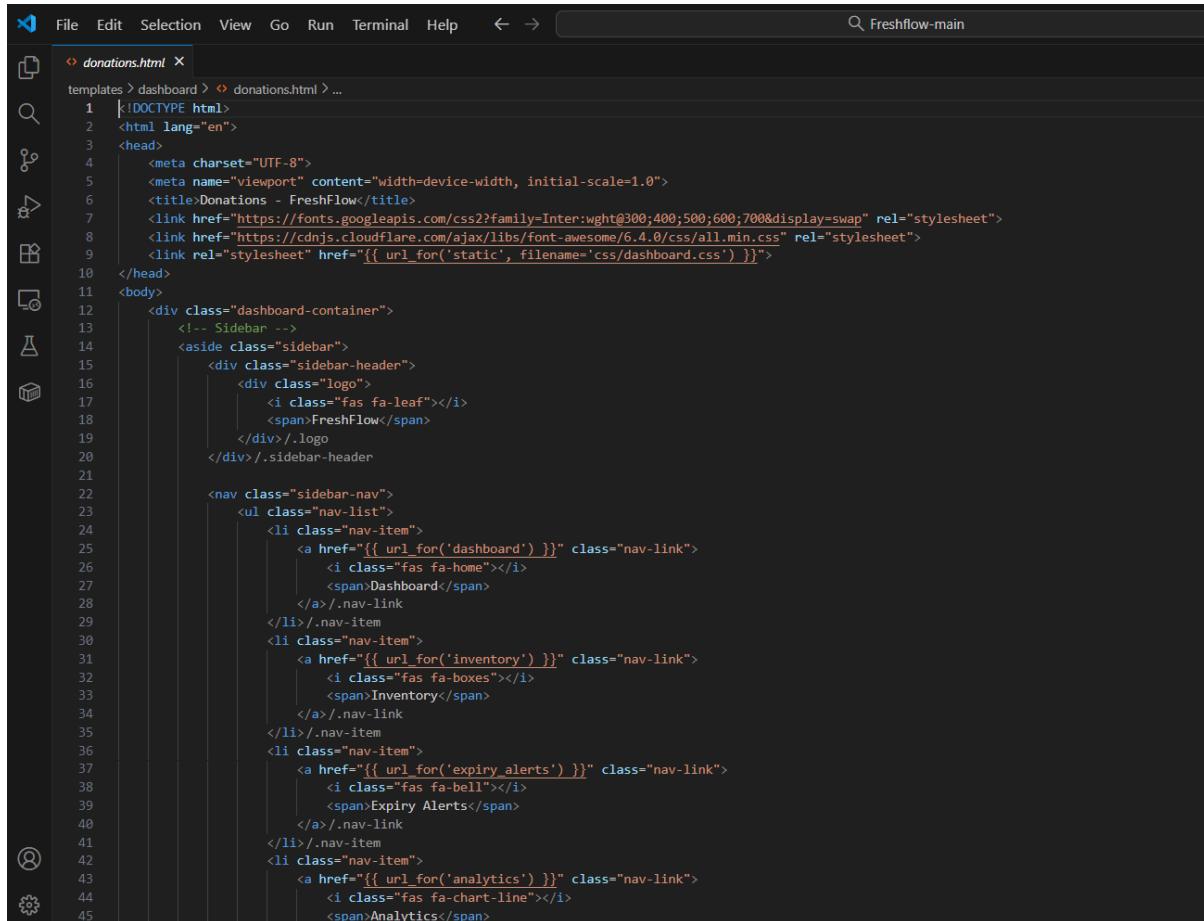
```
<div> discount_product.html </div>
templates > dashboard > discount_product.html > html > body > div.dashboard-container > main.main-content > header.top-nav > div.nav-right
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Discount Product - FreshFlow</title>
7       <link href="https://fonts.googleapis.com/css?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">
8       <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.4.0/css/all.min.css" rel="stylesheet">
9       <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
10  </head>
11  <body>
12    <div class="dashboard-container">
13      <aside class="sidebar">
14        <div class="sidebar-header">
15          <div class="logo">
16            <i class="fas fa-leaf"></i>
17            <span>FreshFlow</span>
18          </div>/.logo
19        </div>/.sidebar-header
20        <nav class="sidebar-nav">
21          <ul class="nav-list">
22            <li class="nav-item"><a class="nav-link" href="{{ url_for('dashboard') }}><i class="fas fa-home"></i><span>Dashboard</span></a></li>
23            <li class="nav-item"><a class="nav-link" href="{{ url_for('inventory') }}><i class="fas fa-boxes"></i><span>Inventory</span></a></li>
24            <li class="nav-item"><a class="nav-link" href="{{ url_for('expiry_alerts') }}><i class="fas fa-bell"></i><span>Expiry Alerts</span></a></li>
25          </ul>/.nav-list
26        </nav>/.sidebar-nav
27        <div class="sidebar-footer">
28          <div class="user-info">
29            <div class="user-avatar"><i class="fas fa-user"></i></div>
30            <div class="user-details">
31              <span class="user-name">{{ session.username }}</span>
32              <span class="user-role">Admin</span>
33            </div>/.user-details
34          </div>/.user-info
35          <a href="{{ url_for('logout') }}><span>Logout</span></a>
36        </div>/.sidebar-footer
37      </aside>/.sidebar
38
39      <main class="main-content">
40        <header class="top-nav">
41          <div class="nav-left">
42            <button class="sidebar-toggle"><i class="fas fa-bars"></i></button>
43            <h1 class="page-title">Discount Products</h1>
44          </div>/.nav-left
45          <div class="nav-right">
```



```
<div> discount_product.html </div>
templates > dashboard > discount_product.html > html > body > div.dashboard-container > main.main-content > header.top-nav > div.nav-right
2   <html lang="en">
3   <body>
4     <div class="dashboard-container">
5       <main class="main-content">
6         <div class="dashboard-content">
7           <div class="table-container">
8             <table class="data-table">
9               <thead>
10                 <tr>
11                   <th>Item</th>
12                   <th>Category</th>
13                   <th>Original Price</th>
14                   <th>Discount %</th>
15                   <th>Discounted Price</th>
16                   <th>Qty Available</th>
17                   <th>Start Date</th>
18                 </tr>
19               </thead>
20               <tbody>
21                 <% for d in items %>
22                   <tr>
23                     <td>{{ d.name }}</td>
24                     <td>{{ d.category }}</td>
25                     <td>{{ '%.2f' | format(d.original_cost or 0) }}</td>
26                     <td>{{ '%.2f' | format(d.discount_percentage or 0) }}%</td>
27                     <td>{{ '%.2f' | format(d.discounted_price or 0) }}</td>
28                     <td>{{ d.quantity_available }}</td>
29                     <td>{{ d.start_date }}</td>
30                   </tr>
31                 <% endfor %>
32               </tbody>
33             </table>/.data-table
34           <div>/.table-container
35           (% else %)
36           <p style="color: var(--text-light);">No discount products available at the moment.</p>
37           (% endif %)
38         </div>/.card-content
39       </div>/.table-card
40     </div>/.dashboard-content
41   </main>/.main-content
42 </div>/.dashboard-container
43
44   <script src="{{ url_for('static', filename='js/dashboard.js') }}></script>
45 </body>
46 </html>
```

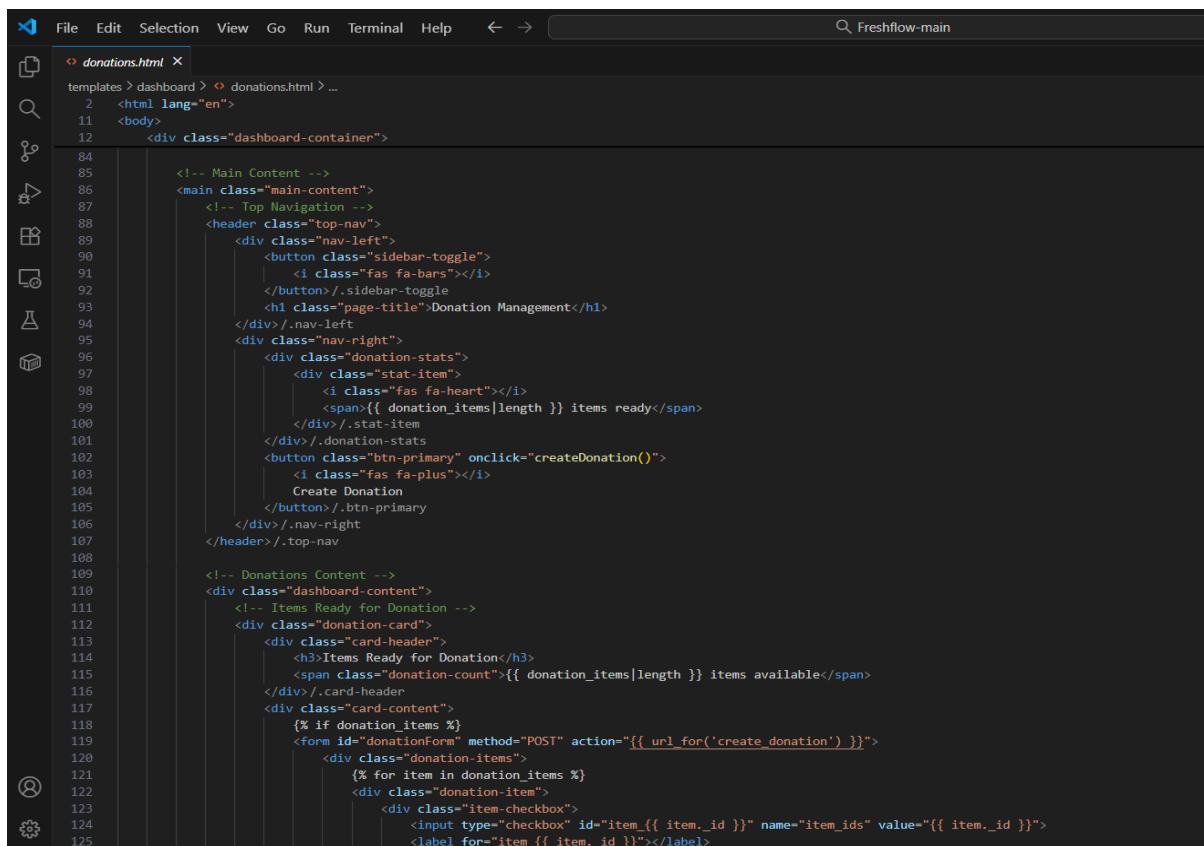
Donation Page

Donation.html



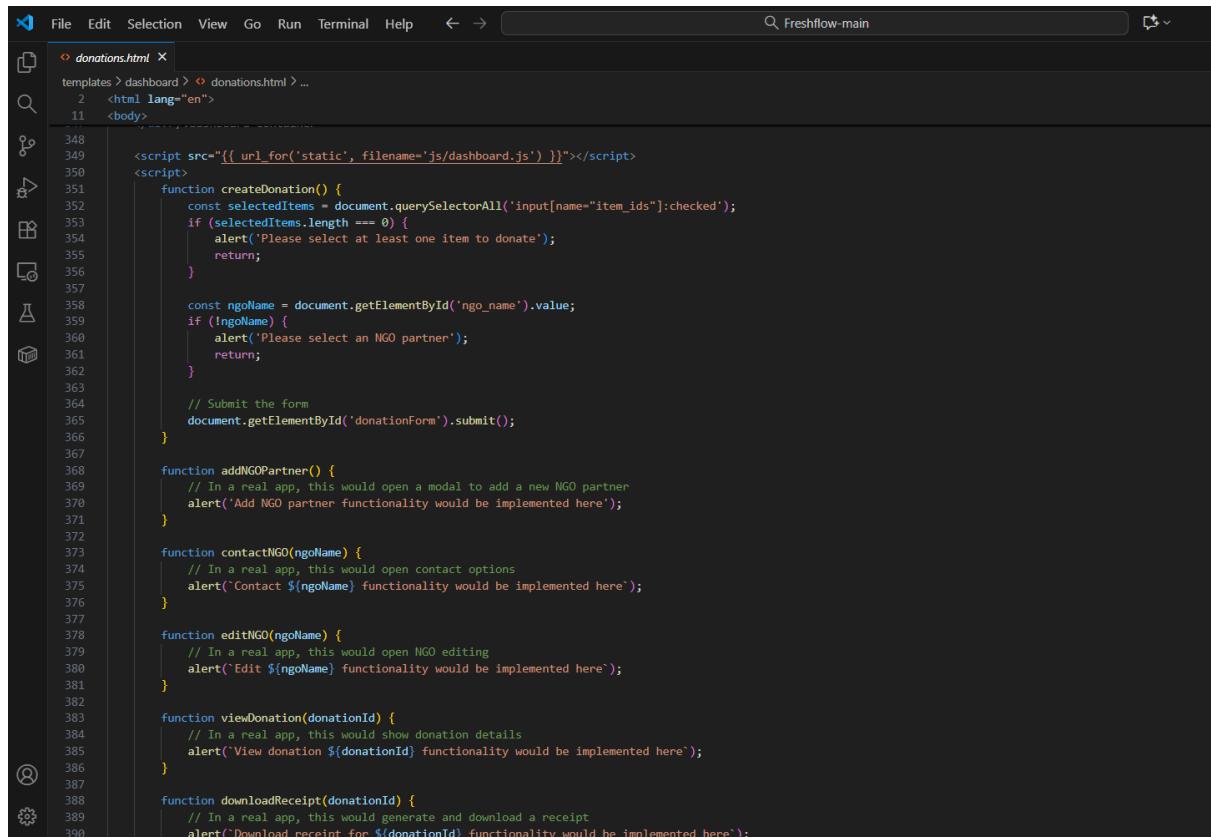
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Donations - FreshFlow</title>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300,400,500,600,700&display=sans" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}">
  </head>
  <body>
    <div class="dashboard-container">
      <!-- Sidebar -->
      <aside class="sidebar">
        <div class="sidebar-header">
          <div class="logo">
            <i class="fas fa-leaf"></i>
            <span>FreshFlow</span>
          </div>/.logo
        </div>/.sidebar-header

        <nav class="sidebar-nav">
          <ul class="nav-list">
            <li class="nav-item">
              <a href="{{ url_for('dashboard') }}" class="nav-link">
                <i class="fas fa-home"></i>
                <span>Dashboard</span>
              </a>/.nav-link
            </li>/.nav-item
            <li class="nav-item">
              <a href="{{ url_for('inventory') }}" class="nav-link">
                <i class="fas fa-boxes"></i>
                <span>Inventory</span>
              </a>/.nav-link
            </li>/.nav-item
            <li class="nav-item">
              <a href="{{ url_for('expiry_alerts') }}" class="nav-link">
                <i class="fas fa-bell"></i>
                <span>Expiry Alerts</span>
              </a>/.nav-link
            </li>/.nav-item
            <li class="nav-item">
              <a href="{{ url_for('analytics') }}" class="nav-link">
                <i class="fas fa-chart-line"></i>
                <span>Analytics</span>
              </a>/.nav-link
            </li>/.nav-item
          </ul>/.nav-list
        </nav>/.sidebar-nav
      </aside>/.dashboard-container
    </body>
  </html>
```



```
<div class="dashboard-container">
  <!-- Main Content -->
  <main class="main-content">
    <!-- Top Navigation -->
    <header class="top-nav">
      <div class="nav-left">
        <button class="sidebar-toggle">
          <i class="fas fa-bars"></i>
        </button>/.sidebar-toggle
      </div>/.nav-left
      <div class="nav-right">
        <div class="donation-stats">
          <div class="stat-item">
            <i class="fas fa-heart"></i>
            <span>{{ donation_items|length }} items ready</span>
          </div>/.stat-item
        </div>/.donation-stats
        <button class="btn-primary" onclick="createDonation()">
          <i class="fas fa-plus"></i>
          Create Donation
        </button>/.btn-primary
      </div>/.nav-right
    </header>/.top-nav

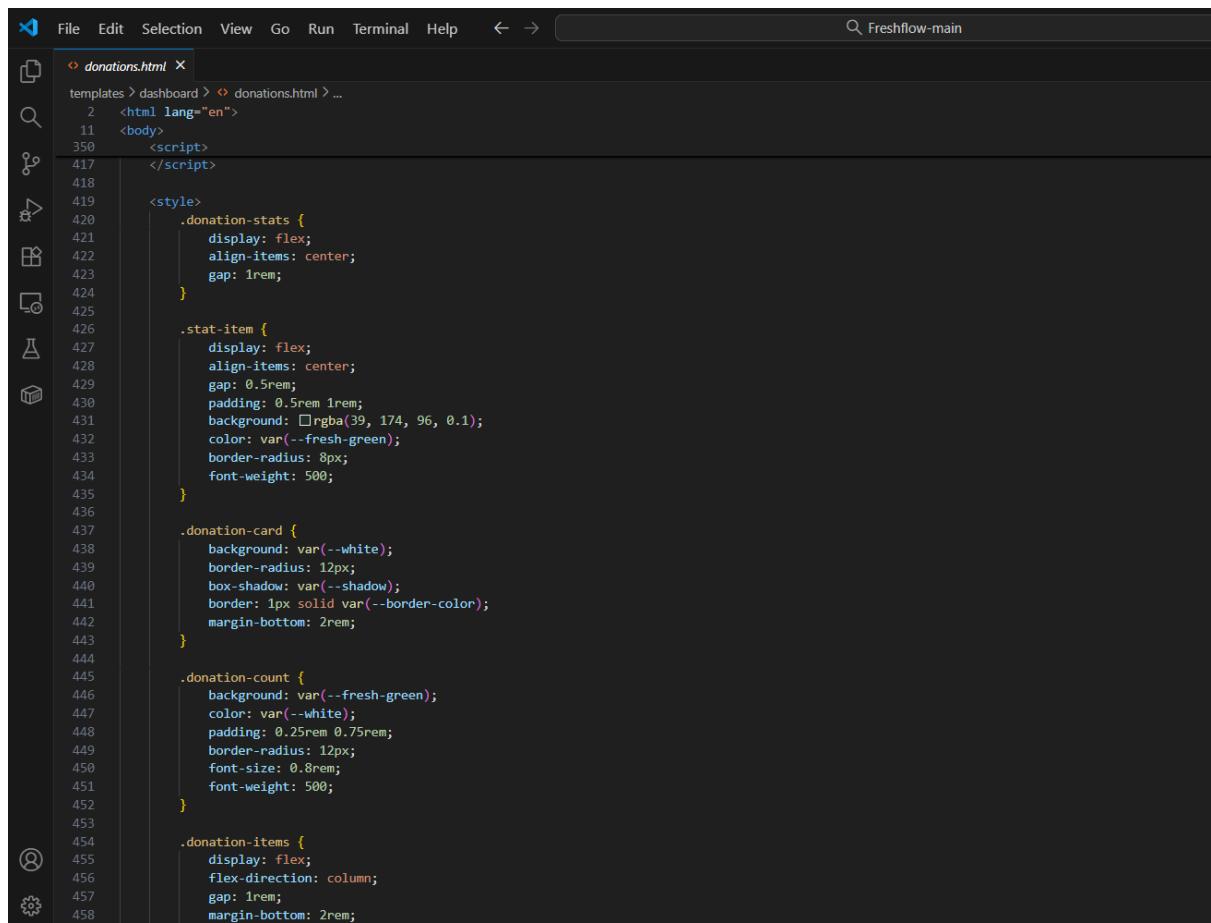
    <!-- Donations Content -->
    <div class="dashboard-content">
      <!-- Items Ready for Donation -->
      <div class="donation-card">
        <div class="card-header">
          <h3>Items Ready for Donation</h3>
          <span class="donation-count">{{ donation_items|length }} items available</span>
        </div>/.card-header
        <div class="card-content">
          {% if donation_items %}
            <form id="donationForm" method="POST" action="{{ url_for('create_donation') }}">
              <div class="donation-items">
                {% for item in donation_items %}
                  <div class="donation-item">
                    <div class="item-checkbox">
                      <input type="checkbox" id="item_{{ item._id }}" name="item_ids" value="{{ item._id }}"/>
                      <label for="item_{{ item._id }}"></label>
                    </div>/.item-checkbox
                  </div>/.donation-item
                {% endfor %}
              </div>/.donation-items
            </form>/.donation-form
          {% endif %}
        </div>/.card-content
      </div>/.donation-card
    </div>/.dashboard-content
  </main>/.main-content
</div>/.dashboard-container
```



A screenshot of a code editor showing the file `donations.html`. The code contains several JavaScript functions:

- `createDonation()`: Checks if at least one item is selected. If none, it alerts "Please select at least one item to donate".
- `ngoName`: Gets the value of the `ngo_name` input field. If empty, it alerts "Please select an NGO partner".
- `document.getElementById('donationForm').submit()`: Submits the form.
- `addNGOPartner()`: Alerts "Add NGO partner functionality would be implemented here".
- `contactNGO(ngoName)`: Alerts "Contact \${ngoName} functionality would be implemented here".
- `editNGO(ngoName)`: Alerts "Edit \${ngoName} functionality would be implemented here".
- `viewDonation(donationId)`: Alerts "View donation \${donationId} functionality would be implemented here".
- `downloadReceipt(donationId)`: Alerts "Download receipt for \${donationId} functionality would be implemented here".

The code uses ES6 syntax like arrow functions and template literals.

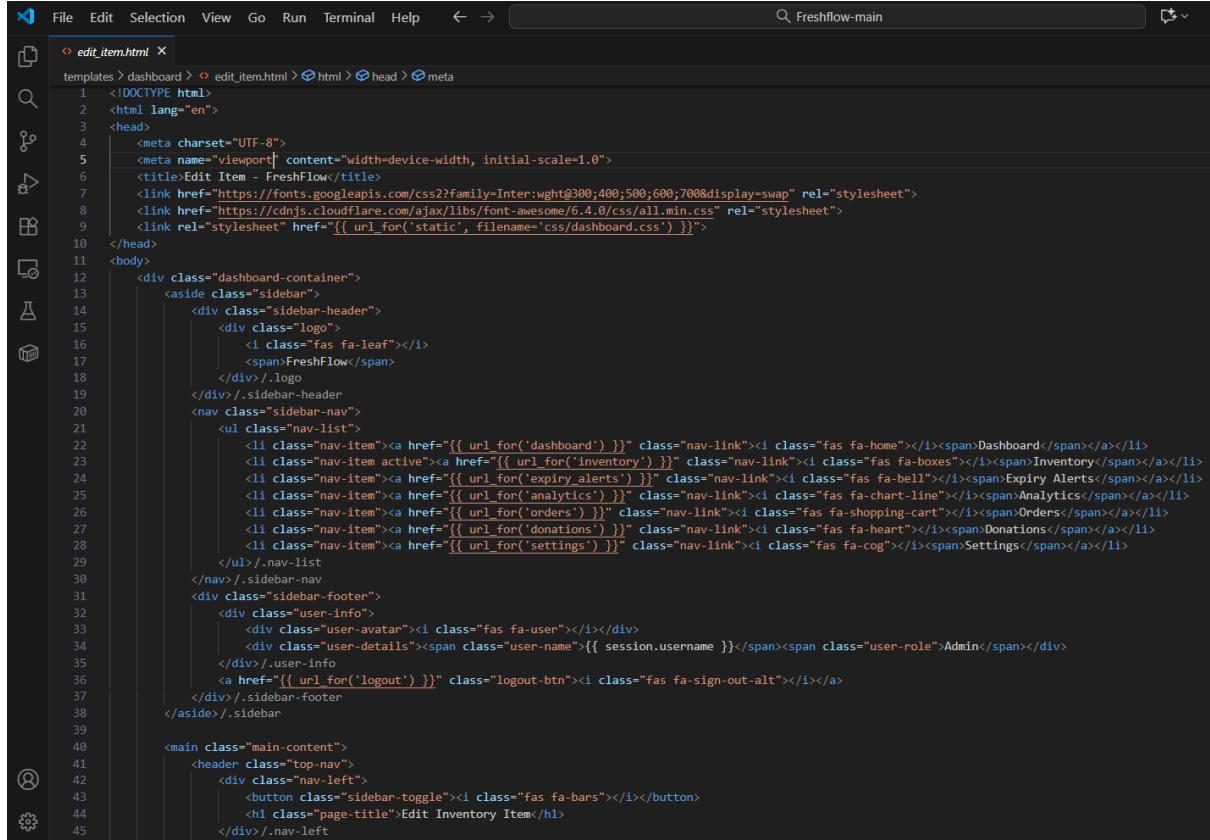


A screenshot of a code editor showing the file `donations.html`. The code contains CSS styles for various components:

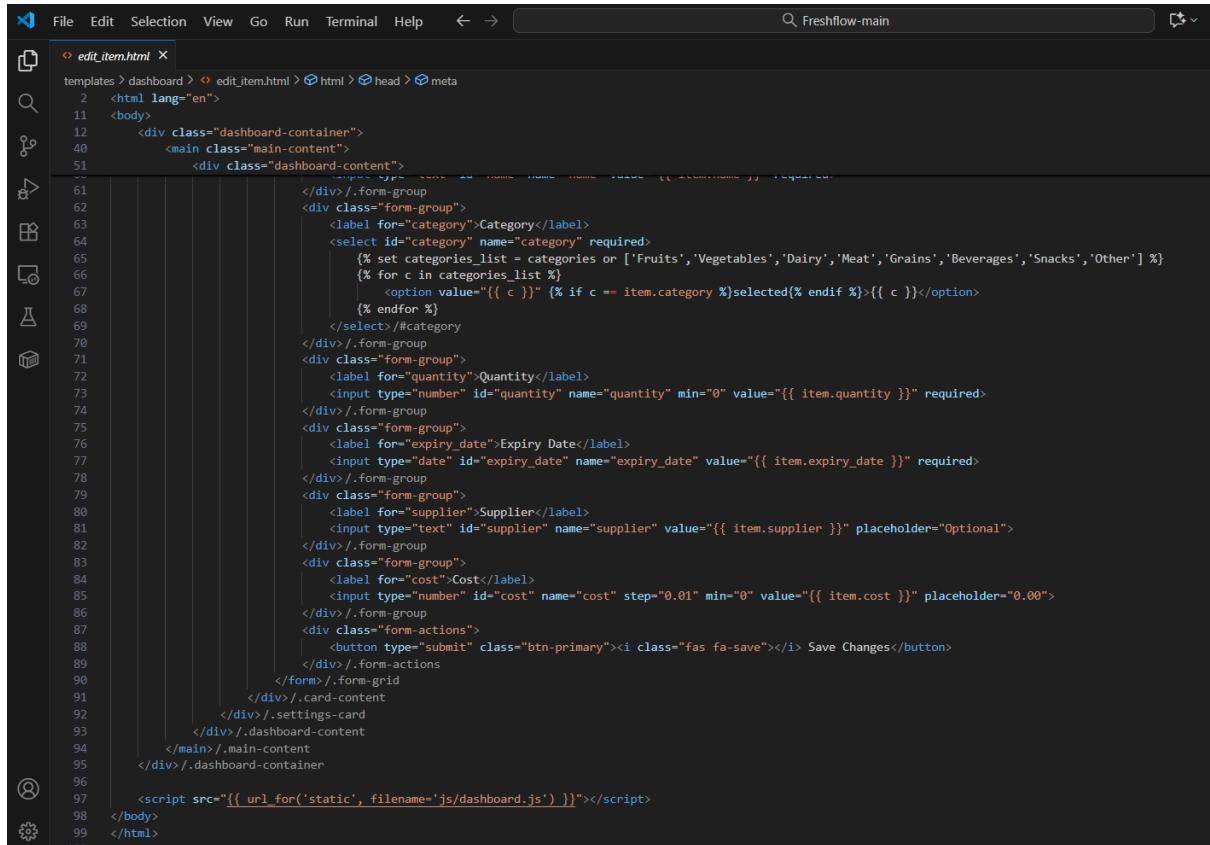
- `.donation-stats`: A flex container with items centered, gap set to 1rem.
- `.stat-item`: A flex container with items centered, gap set to 0.5rem, padding of 0.5rem 1rem, background color `rgba(39, 174, 96, 0.1)`, border-radius of 8px, and font-weight of 500.
- `.donation-card`: A card with white background, rounded corners, shadow, border, and margin-bottom of 2rem.
- `.donation-count`: A card with green background, white color, padding of 0.25rem 0.75rem, rounded corners, font-size of 0.8rem, and font-weight of 500.
- `.donation-items`: A flex container with items in column direction, gap set to 1rem, and margin-bottom of 2rem.

Add or Delete Products Page

Edit_item.html



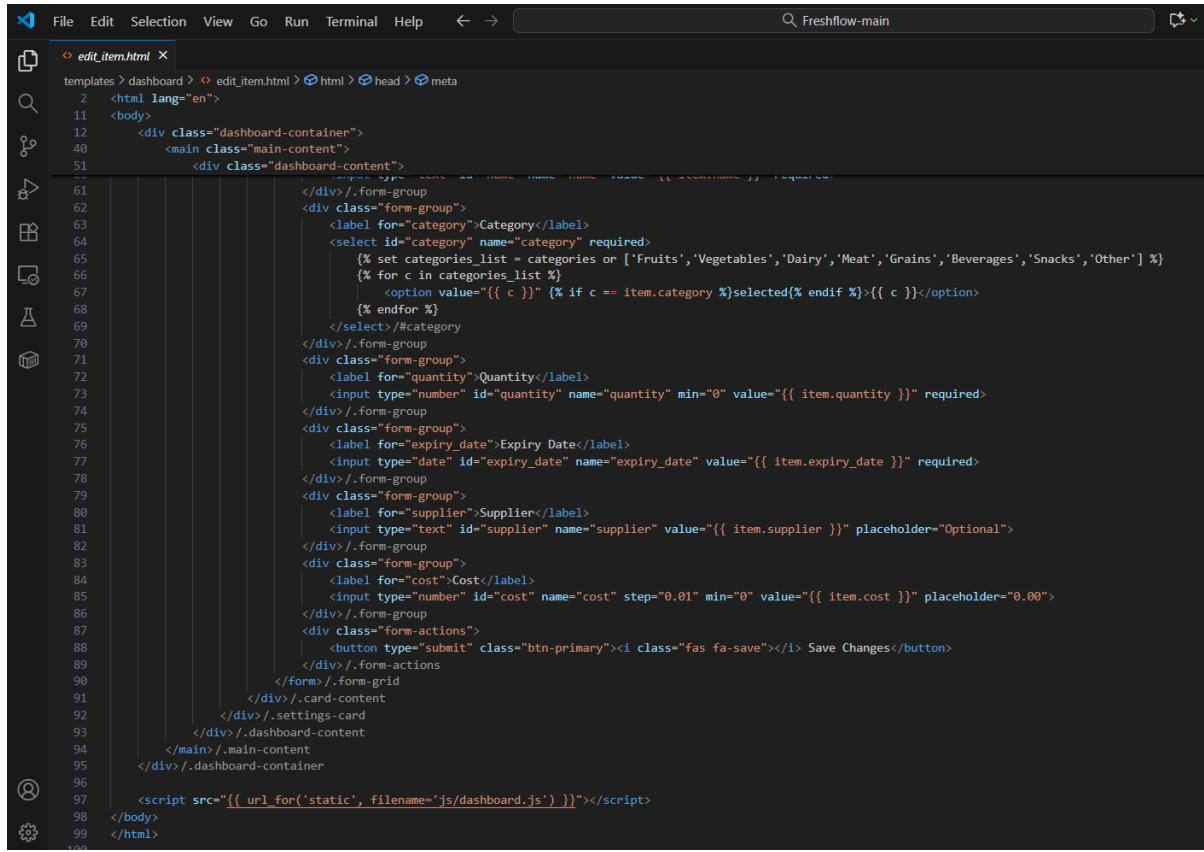
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Item - Freshflow</title>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
</head>
<body>
    <div class="dashboard-container">
        <aside class="sidebar">
            <div class="sidebar-header">
                <div class="logo">
                    <i class="fas fa-leaf"></i>
                    <span>FreshFlow</span>
                </div>/.sidebar-header
            <nav class="sidebar-nav">
                <ul class="nav-list">
                    <li class="nav-item"><a href="{{ url_for('dashboard') }}" class="nav-link"><i class="fas fa-home"></i><span>Dashboard</span></a></li>
                    <li class="nav-item active"><a href="{{ url_for('inventory') }}" class="nav-link"><i class="fas fa-boxes"></i><span>Inventory</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('expiry_alerts') }}" class="nav-link"><i class="fas fa-bell"></i><span>Expiry Alerts</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('analytics') }}" class="nav-link"><i class="fas fa-chart-line"></i><span>Analytics</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('orders') }}" class="nav-link"><i class="fas fa-shopping-cart"></i><span>Orders</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('donations') }}" class="nav-link"><i class="fas fa-heart"></i><span>Donations</span></a></li>
                    <li class="nav-item"><a href="{{ url_for('settings') }}" class="nav-link"><i class="fas fa-cog"></i><span>Settings</span></a></li>
                </ul>/.nav-list
            </nav>/.sidebar-nav
            <div class="sidebar-footer">
                <div class="user-info">
                    <div class="user-avatar"><i class="fas fa-user"></i></div>
                    <div class="user-details"><span class="user-name">{{ session.username }}</span><span class="user-role">Admin</span></div>
                </div>/.user-info
                <a href="{{ url_for('logout') }}" class="logout-btn"><i class="fas fa-sign-out-alt"></i></a>
            </div>/.sidebar-footer
        </aside>/.sidebar
        <main class="main-content">
            <header class="top-nav">
                <div class="nav-left">
                    <button class="sidebar-toggle"><i class="fas fa-bars"></i></button>
                    <h1 class="page-title">Edit Inventory Item</h1>
                </div>/.nav-left
            </header>
            <div class="card">
                <div class="card-content">
                    <form>
                        <div class="form-group">
                            <label for="category">Category:</label>
                            <select id="category" name="category" required>
                                <% set categories_list = categories or [ 'Fruits', 'Vegetables', 'Dairy', 'Meat', 'Grains', 'Beverages', 'Snacks', 'Other' ] %>
                                <% for c in categories_list %>
                                    <option value="{{ c }}><% if c == item.category %>selected(% endif %)><{{ c }}</option>
                                <% endfor %>
                            </select>/.category
                        </div>/.form-group
                        <div class="form-group">
                            <label for="quantity">Quantity:</label>
                            <input type="number" id="quantity" name="quantity" min="0" value="{{ item.quantity }}" required>
                        </div>/.form-group
                        <div class="form-group">
                            <label for="expiry_date">Expiry Date:</label>
                            <input type="date" id="expiry_date" name="expiry_date" value="{{ item.expiry_date }}" required>
                        </div>/.form-group
                        <div class="form-group">
                            <label for="supplier">Supplier:</label>
                            <input type="text" id="supplier" name="supplier" value="{{ item.supplier }}" placeholder="Optional">
                        </div>/.form-group
                        <div class="form-group">
                            <label for="cost">Cost:</label>
                            <input type="number" id="cost" name="cost" step="0.01" min="0" value="{{ item.cost }}" placeholder="0.00">
                        </div>/.form-group
                        <div class="form-actions">
                            <button type="submit" class="btn-primary"><i class="fas fa-save"></i> Save Changes</button>
                        </div>/.form-actions
                    </form>/.form-grid
                </div>/.card-content
            </div>/.settings-card
        </main>/.main-content
    </div>/.dashboard-container
    <script src="{{ url_for('static', filename='js/dashboard.js') }}></script>
</body>
</html>
```



```
<!DOCTYPE html>
<html lang="en">
<body>
    <div class="dashboard-container">
        <main class="main-content">
            <div class="dashboard-content">
                <form>
                    <div class="form-group">
                        <label for="category">Category:</label>
                        <select id="category" name="category" required>
                            <% set categories_list = categories or [ 'Fruits', 'Vegetables', 'Dairy', 'Meat', 'Grains', 'Beverages', 'Snacks', 'Other' ] %>
                            <% for c in categories_list %>
                                <option value="{{ c }}><% if c == item.category %>selected(% endif %)><{{ c }}</option>
                            <% endfor %>
                        </select>/.category
                    </div>/.form-group
                    <div class="form-group">
                        <label for="quantity">Quantity:</label>
                        <input type="number" id="quantity" name="quantity" min="0" value="{{ item.quantity }}" required>
                    </div>/.form-group
                    <div class="form-group">
                        <label for="expiry_date">Expiry Date:</label>
                        <input type="date" id="expiry_date" name="expiry_date" value="{{ item.expiry_date }}" required>
                    </div>/.form-group
                    <div class="form-group">
                        <label for="supplier">Supplier:</label>
                        <input type="text" id="supplier" name="supplier" value="{{ item.supplier }}" placeholder="Optional">
                    </div>/.form-group
                    <div class="form-group">
                        <label for="cost">Cost:</label>
                        <input type="number" id="cost" name="cost" step="0.01" min="0" value="{{ item.cost }}" placeholder="0.00">
                    </div>/.form-group
                    <div class="form-actions">
                        <button type="submit" class="btn-primary"><i class="fas fa-save"></i> Save Changes</button>
                    </div>/.form-actions
                </form>/.form-grid
            </div>/.card-content
        </div>/.settings-card
    </main>/.main-content
</div>/.dashboard-container
<script src="{{ url_for('static', filename='js/dashboard.js') }}></script>
</body>
</html>
```

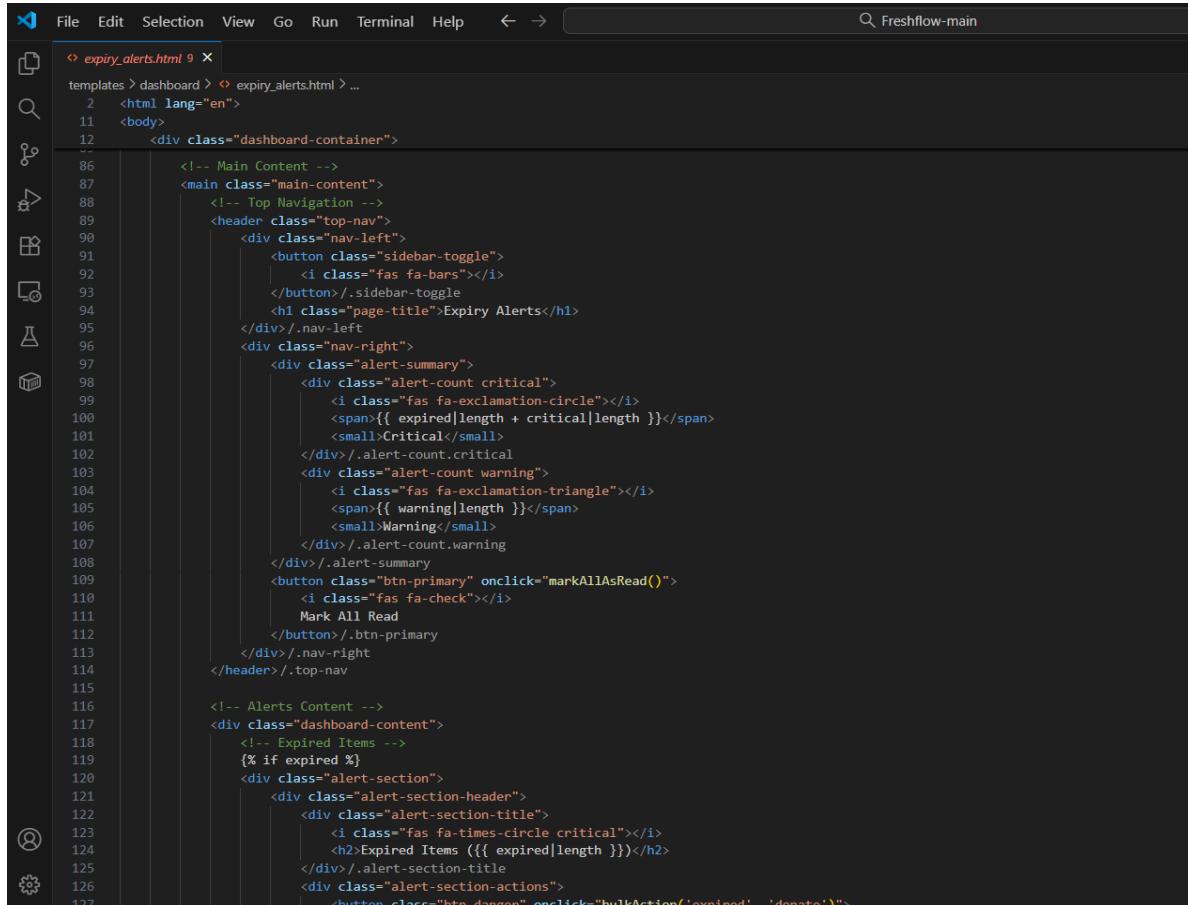
Expiry Alert page

Expiry_alerts.html



```
File Edit Selection View Go Run Terminal Help ← → 🔍 Freshflow-main

edit_item.html X
templates > dashboard > edit_item.html > html > head > meta
2   <html lang="en">
11    <body>
12      <div class="dashboard-container">
40        <main class="main-content">
51          <div class="dashboard-content">
52            <form class="card">
53              <div class="card-content">
54                <div class="form-grid">
55                  <div class="form-group">
56                    <label for="category">Category</label>
57                    <select id="category" name="category" required>
58                      <% set categories_list = categories or ['Fruits', 'Vegetables', 'Dairy', 'Meat', 'Grains', 'Beverages', 'Snacks', 'Other'] %>
59                      <% for c in categories_list %>
60                        <option value="{{ c }}>{{ c }}</option>
61                      <% endfor %>
62                    </select> //#category
63                  </div>/.form-group
64                  <div class="form-group">
65                    <label for="quantity">Quantity</label>
66                    <input type="number" id="quantity" name="quantity" min="0" value="{{ item.quantity }}" required>
67                  </div>/.form-group
68                  <div class="form-group">
69                    <label for="expiry_date">Expiry Date</label>
70                    <input type="date" id="expiry_date" name="expiry_date" value="{{ item.expiry_date }}" required>
71                  </div>/.form-group
72                  <div class="form-group">
73                    <label for="supplier">Supplier</label>
74                    <input type="text" id="supplier" name="supplier" value="{{ item.supplier }}" placeholder="Optional">
75                  </div>/.form-group
76                  <div class="form-group">
77                    <label for="cost">Cost</label>
78                    <input type="number" id="cost" name="cost" step="0.01" min="0" value="{{ item.cost }}" placeholder="0.00">
79                  </div>/.form-group
80                  <div class="form-actions">
81                    <button type="submit" class="btn-primary">i class="fas fa-save"></i> Save Changes</button>
82                  </div>/.form-actions
83                </div>/.form-grid
84              </div>/.card-content
85            </div>/.settings-card
86          </div>/.dashboard-content
87        </main>/.main-content
88      </div>/.dashboard-container
89
90      <script src="{{ url_for('static', filename='js/dashboard.js') }}></script>
91
92    </body>
93
94  </html>
```



```
File Edit Selection View Go Run Terminal Help ← → 🔍 Freshflow-main

expiry_alerts.html 9 X
templates > dashboard > expiry_alerts.html > ...
2   <html lang="en">
11    <body>
12      <div class="dashboard-container">
13        <!-- Main Content -->
14        <main class="main-content">
15          <!-- Top Navigation -->
16          <header class="top-nav">
17            <div class="nav-left">
18              <button class="sidebar-toggle">
19                <i class="fas fa-bars"></i>
20              </button>/.sidebar-toggle
21            <div class="page-title">Expiry Alerts</h1>
22          </div>/.nav-left
23          <div class="nav-right">
24            <div class="alert-summary">
25              <div class="alert-count critical">
26                <i class="fas fa-exclamation-circle"></i>
27                <span>{{ expired|length + critical|length }}</span>
28                <small>Critical</small>
29              </div>/.alert-count.critical
30              <div class="alert-count warning">
31                <i class="fas fa-exclamation-triangle"></i>
32                <span>{{ warning|length }}</span>
33                <small>Warning</small>
34              </div>/.alert-count.warning
35            </div>/.alert-summary
36            <button class="btn-primary" onclick="markAllAsRead()">
37              <i class="fas fa-check"></i>
38              Mark All Read
39            </button>/.btn-primary
40          </div>/.nav-right
41        </header>/.top-nav
42
43        <!-- Alerts Content -->
44        <div class="dashboard-content">
45          <!-- Expired Items -->
46          <% if expired %>
47            <div class="alert-section">
48              <div class="alert-section-header">
49                <div class="alert-section-title">
50                  <i class="fas fa-times-circle critical"></i>
51                  <h2>Expired Items {{ expired|length }}</h2>
52                </div>/.alert-section-title
53                <div class="alert-section-actions">
54                  <button class="btn-danger" onclick="bulkAction('expired', 'donate')">
55
```

The screenshot shows a code editor interface with a dark theme. The title bar reads "Freshflow-main". The left sidebar contains icons for file operations like Open, Save, Find, and Run. The main editor area displays a script block within a file named "expiry_alerts.html". The script handles various actions such as closing modals, confirming actions, and auto-refreshing alerts every 5 minutes.

```
File Edit Selection View Go Run Terminal Help ← → Freshflow-main



<script>
    // Close modal
    function closeModal(modalId) {
        document.getElementById(modalId).style.display = 'none';
        currentAction = null;
    }

    // Confirm action
    function confirmAction() {
        if (!currentAction) return;

        const { type, action, itemId, items } = currentAction;

        if (type === 'single') {
            if (action === 'donate') {
                window.location.href = `/donations?item=${itemId}`;
            } else if (action === 'remove') {
                window.location.href = `/inventory/delete/${itemId}`;
            } else if (action === 'reorder') {
                window.location.href = `/orders/reorder=${itemId}`;
            }
        } else if (type === 'bulk') {
            if (action === 'donate') {
                const itemIds = items.join(',');
                window.location.href = `/donations?items=${itemIds}`;
            } else if (action === 'reorder') {
                const itemIds = items.join(',');
                window.location.href = `/orders/reorder=${itemIds}`;
            } else if (action === 'delete') {
                // In a real app, this would make an API call to delete multiple items
                alert(`Bulk delete ${items.length} items functionality would be implemented here`);
            }
        }
        closeModal('actionModal');
    }

    // Auto-refresh alerts every 5 minutes
    setInterval(() => {
        window.location.reload();
    }, 300000);
</script>


```

The screenshot shows a code editor interface with a dark theme. The title bar reads "Freshflow-main". The left sidebar contains icons for file operations like Open, Save, Find, and Run. The main editor area displays a style block within a file named "expiry_alerts.html". The styles define various CSS rules for elements like h3, p, and different alert sections, including media queries for responsive design.

```
File Edit Selection View Go Run Terminal Help ← → Freshflow-main



<style>
    .empty-state h3 {
        margin-bottom: 0.5rem;
    }

    .empty-state p {
        color: var(--text-light);
        margin-bottom: 2rem;
    }

    @media (max-width: 768px) {
        .alert-section-header {
            flex-direction: column;
            gap: 1rem;
            align-items: flex-start;
        }

        .alert-section-actions {
            width: 100%;
            justify-content: flex-start;
        }

        .alert-item-header {
            flex-direction: column;
            gap: 1rem;
            align-items: flex-start;
        }

        .alert-item-meta {
            align-items: flex-start;
        }

        .alert-item-details {
            grid-template-columns: 1fr;
        }

        .alert-item-actions {
            flex-direction: column;
        }
    }
</style>
</body>
</html>


```

Inventory Stocks Page

Inventory.html

```
File Edit Selection View Go Run Terminal Help ⏎ ⏎ Freshflow-main  
elixir_alerts.html 9 inventory.html  
templates > dashboard > inventory.html ...  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4     <meta charset="UTF-8">  
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6     <title>Inventory - Freshflow</title>  
7     <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
8     <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">  
9     <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}">  
10    </head>  
11    <body>  
12        <div class="dashboard-container">  
13            <!-- Sidebar -->  
14            <aside class="sidebar">  
15                <div class="sidebar-header">  
16                    <div class="logo">  
17                        <i class="fas fa-leaf"></i>  
18                        <span>FreshFlow</span>  
19                    </div>/ .logo  
20                </div>/ .sidebar-header  
21  
22                <nav class="sidebar-nav">  
23                    <ul class="nav-list">  
24                        <li class="nav-item">  
25                            <a href="{{ url_for('dashboard') }}" class="nav-link">  
26                                <i class="fas fa-home"></i>  
27                                <span>Dashboard</span>  
28                            </a>/ .nav-link  
29                        </li>.nav-item  
30                        <li class="nav-item active">  
31                            <a href="{{ url_for('inventory') }}" class="nav-link">  
32                                <i class="fas fa-boxes"></i>  
33                                <span>Inventory</span>  
34                            </a>/ .nav-link  
35                        </li>.nav-item.active  
36                        <li class="nav-item">  
37                            <a href="{{ url_for('expiry_alerts') }}" class="nav-link">  
38                                <i class="fas fa-bell"></i>  
39                                <span>Expiry Alerts</span>  
40                            </a>/ .nav-link  
41                        </li>.nav-item  
42                        <li class="nav-item">  
43                            <a href="{{ url_for('analytics') }}" class="nav-link">  
44                                <i class="fas fa-chart-line"></i>  
45                                <span>Analytics</span>  
46                    </ul>.nav-list  
47                </nav>/ .sidebar-nav  
48            </aside>/ .dashboard-container  
49        </div>/ .body  
50    </body>  
51 </html>
```

```
File Edit Selection View Go Run Terminal Help ⏎ ⏎ Freshflow-main  
elixir_alerts.html 9 inventory.html  
templates > dashboard > inventory.html ...  
1 <html lang="en">  
2 <body>  
3     <div class="dashboard-container">  
4         <!-- Main Content -->  
5         <main class="main-content">  
6             <!-- Top Navigation -->  
7             <header class="top-nav">  
8                 <div class="nav-left">  
9                     <button class="sidebar-toggle">  
10                         <i class="fas fa-bars"></i>  
11                     </button>/ .sidebar-toggle  
12                     <h1 class="page-title">Inventory Management</h1>  
13                 </div>/ .nav-left  
14                 <div class="nav-right">  
15                     <div class="search-box">  
16                         <i class="fas fa-search"></i>  
17                         <input type="text" placeholder="Search inventory..." id="searchInput">  
18                     </div>/ .search-box  
19                     <a href="{{ url_for('add_item') }}" class="btn-primary">  
20                         <i class="fas fa-plus"></i>  
21                         Add Item  
22                     </a>/ .btn-primary  
23                 </div>/ .nav-right  
24             </header>/ .top-nav  
25  
26             <!-- Inventory Content -->  
27             <div class="dashboard-content">  
28                 <!-- Filters -->  
29                 <div class="filters-card">  
30                     <div class="filters-header">  
31                         <h3>Filters</h3>  
32                         <button class="btn-secondary" onclick="clearFilters()">  
33                             <i class="fas fa-times"></i>  
34                             Clear All  
35                         </button>/ .btn-secondary  
36                     </div>/ .filters-header  
37                     <div class="filters-content">  
38                         <div class="filter-group">  
39                             <label for="categoryFilter">Category</label>  
40                             <select id="categoryFilter" onchange="applyFilters()">  
41                                 <option value="">All Categories</option>  
42                                 {% for category in categories %}<option value="{{ category }}">{{ category }}</option>  
43                                 {% endfor %}  
44                             </select>/ #categoryFilter  
45                         </div>/ .filter-group  
46                     </div>/ .filters-content  
47                 </div>/ .filters-card  
48             </div>/ .dashboard-content  
49         </main>/ .body  
50     </div>/ .dashboard-container  
51 </body>  
52 </html>
```

A screenshot of a code editor showing the `inventory.html` template file. The code is a complex HTML structure with embedded JavaScript and CSS. It includes sections for bulk actions, a modal for delete confirmation, and various button and link definitions. The editor has a dark theme with syntax highlighting.

```
<!-- Bulk Actions -->
<div class="bulk-actions" id="bulkActions" style="display: none;">
    <div class="bulk-actions-content">
        <span class="selected-count">0 items selected</span>
        <div class="bulk-buttons">
            <button class="btn-secondary" onclick="bulkEdit()">
                <i class="fas fa-edit"></i>
                Edit Selected
            </button> / .btn-secondary
            <button class="btn-danger" onclick="bulkDelete()">
                <i class="fas fa-trash"></i>
                Delete Selected
            </button> / .btn-danger
            <button class="btn-warning" onclick="bulkDonate()">
                <i class="fas fa-heart"></i>
                Donate Selected
            </button> / .btn-warning
        </div> / .bulk-buttons
    </div> / .bulk-actions-content
</div> #bulkActions.bulk-actions
</div> / .dashboard-content
</main> / .main-content
</div> / .dashboard-container
<!-- Delete Confirmation Modal -->
<div class="modal" id="deleteModal">
    <div class="modal-content">
        <div class="modal-header">
            <h3>Confirm Delete</h3>
            <button class="modal-close" onclick="closeModal('deleteModal')">
                <i class="fas fa-times"></i>
            </button> / .modal-close
        </div> / .modal-header
        <div class="modal-body">
            <p>Are you sure you want to delete this item? This action cannot be undone.</p>
        </div> / .modal-body
        <div class="modal-footer">
            <button class="btn-secondary" onclick="closeModal('deleteModal')">Cancel</button>
            <button class="btn-danger" onclick="confirmDelete()">Delete</button>
        </div> / .modal-footer
    </div> / .modal
```

A screenshot of a code editor showing the `inventory.html` template file, specifically the script section. The code contains JavaScript functions for search functionality, applying filters, clearing filters, and sorting a table. It uses document.querySelectorAll and other DOM manipulation techniques.

```
// Search functionality
document.getElementById('searchInput').addEventListener('input', function() {
    const searchTerm = this.value.toLowerCase();
    const rows = document.querySelectorAll('#inventoryTable tbody tr');

    rows.forEach(row => {
        const itemName = row.cells[1].textContent.toLowerCase();
        const category = row.cells[2].textContent.toLowerCase();
        const supplier = row.cells[5].textContent.toLowerCase();

        if (itemName.includes(searchTerm) || category.includes(searchTerm) || supplier.includes(searchTerm)) {
            row.style.display = '';
        } else {
            row.style.display = 'none';
        }
    });
});

// Apply filters
function applyFilters() {
    const category = document.getElementById('categoryFilter').value;
    const status = document.getElementById('statusFilter').value;
    const quantity = document.getElementById('quantityFilter').value;

    const url = new URL(window.location);
    url.searchParams.set('category', category);
    url.searchParams.set('status', status);
    url.searchParams.set('quantity', quantity);

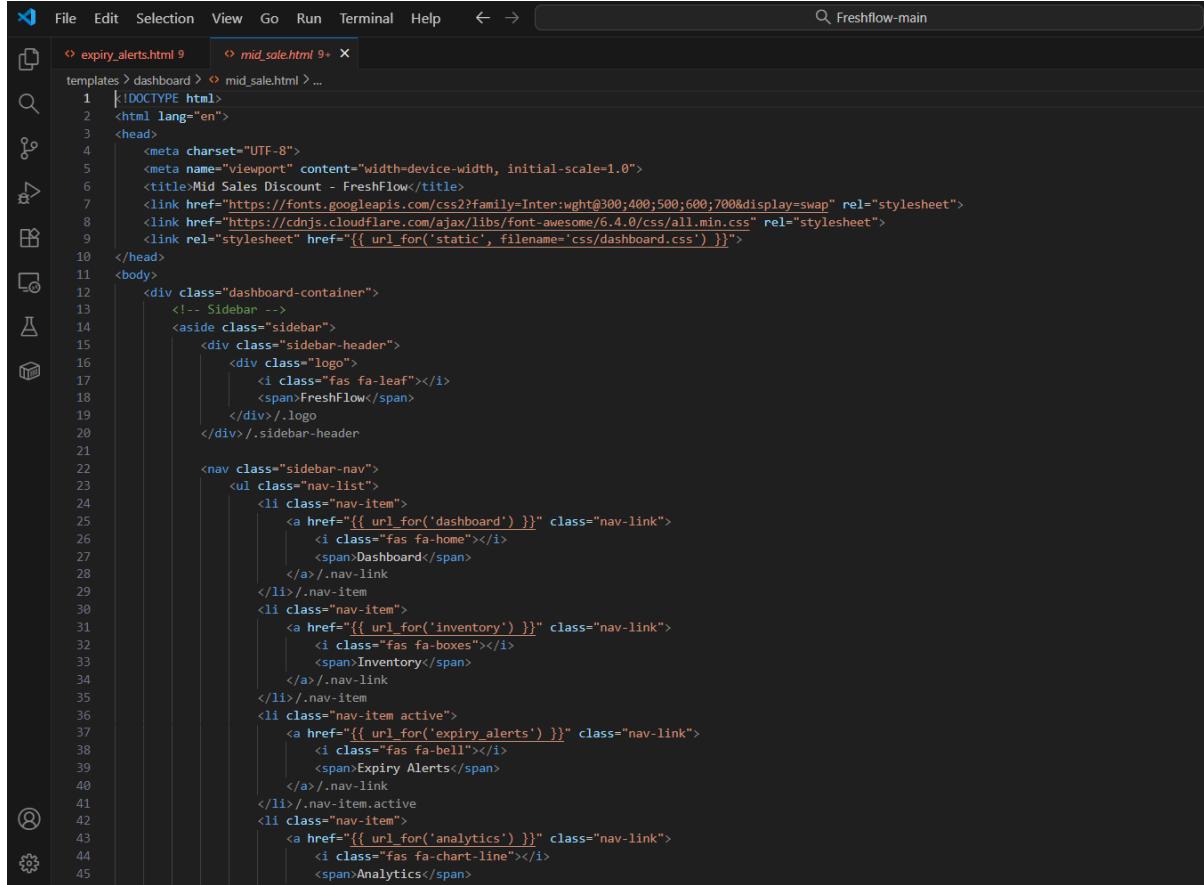
    window.location.href = url.toString();
}

// Clear filters
function clearFilters() {
    document.getElementById('categoryFilter').value = '';
    document.getElementById('statusFilter').value = '';
    document.getElementById('quantityFilter').value = '';
    applyFilters();
}

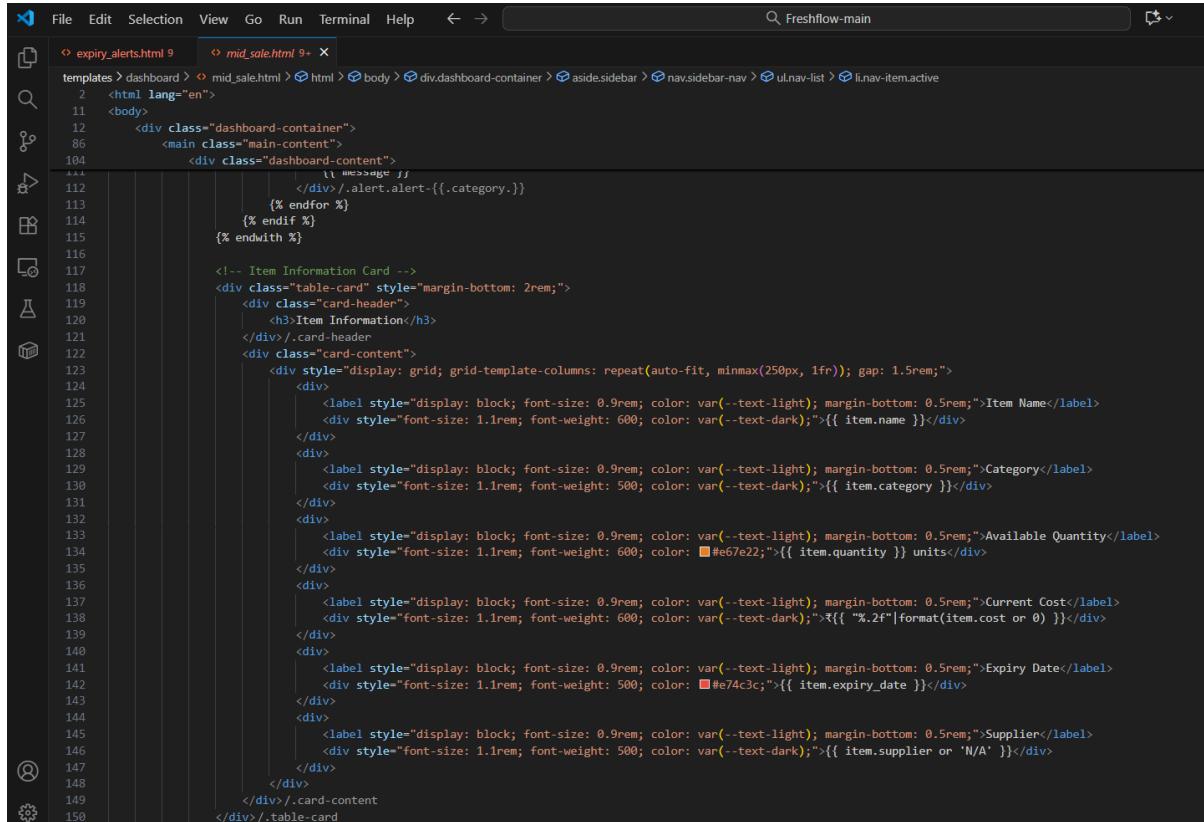
// Sort table
function sortTable(column) {
```

Mid-sale Discount Page

Mid_sale.html



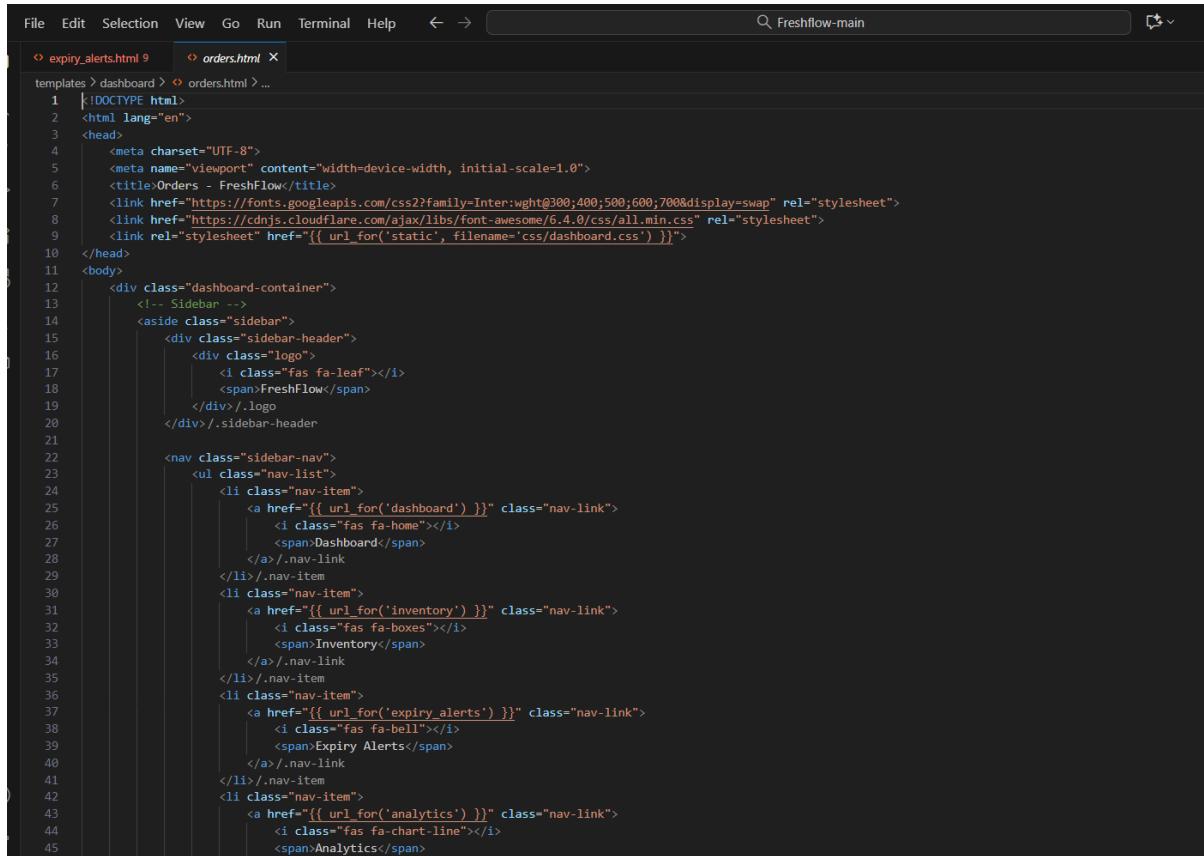
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mid Sales Discount - FreshFlow</title>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;700&display=swap" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
</head>
<body>
    <div class="dashboard-container">
        <!-- Sidebar -->
        <aside class="sidebar">
            <div class="sidebar-header">
                <div class="logo">
                    <i class="fas fa-leaf"></i>
                    <span>FreshFlow</span>
                </div>/.logo
            </div>/.sidebar-header
            <nav class="sidebar-nav">
                <ul class="nav-list">
                    <li class="nav-item">
                        <a href="{{ url_for('dashboard') }}" class="nav-link">
                            <i class="fas fa-home"></i>
                            <span>Dashboard</span>
                        </a>/.nav-link
                    </li>/.nav-item
                    <li class="nav-item">
                        <a href="{{ url_for('inventory') }}" class="nav-link">
                            <i class="fas fa-boxes"></i>
                            <span>Inventory</span>
                        </a>/.nav-link
                    </li>/.nav-item
                    <li class="nav-item active">
                        <a href="{{ url_for('expiry_alerts') }}" class="nav-link">
                            <i class="fas fa-bell"></i>
                            <span>Expiry Alerts</span>
                        </a>/.nav-link
                    </li>/.nav-item.active
                    <li class="nav-item">
                        <a href="{{ url_for('analytics') }}" class="nav-link">
                            <i class="fas fa-chart-line"></i>
                            <span>Analytics</span>
                        </a>/.nav-link
                    </li>/.nav-item
                </ul>/.nav-list
            </nav>/.sidebar-nav
        </aside>/.dashboard-container
    </body>
</html>
```



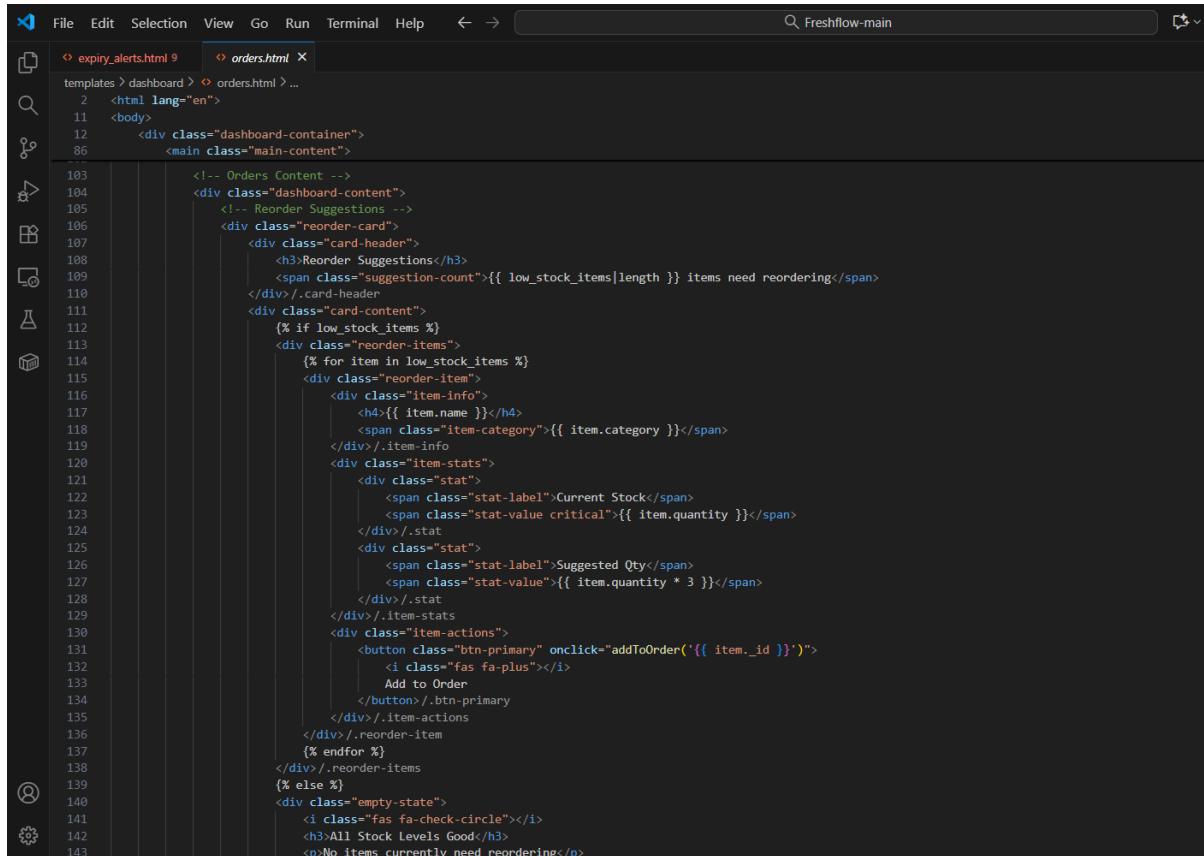
```
<html lang="en">
<body>
    <div class="dashboard-container">
        <main class="main-content">
            <div class="dashboard-content">
                {{ message }}
                <div>/.alert.alert-{{ .category }}</div>
                {% endif %}
                {% endwith %}
                <!-- Item Information Card -->
                <div class="table-card" style="margin-bottom: 2rem;">
                    <div class="card-header">
                        <h3>Item Information</h3>
                    </div>/.card-header
                    <div class="card-content">
                        <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 1.5rem;">
                            <div>
                                <label style="display: block; font-size: 0.9rem; color: var(--text-light); margin-bottom: 0.5rem;">Item Name</label>
                                <div style="font-size: 1.1rem; font-weight: 600; color: var(--text-dark);>{{ item.name }}</div>
                            </div>
                            <div>
                                <label style="display: block; font-size: 0.9rem; color: var(--text-light); margin-bottom: 0.5rem;">Category</label>
                                <div style="font-size: 1.1rem; font-weight: 500; color: var(--text-dark);>{{ item.category }}</div>
                            </div>
                            <div>
                                <label style="display: block; font-size: 0.9rem; color: var(--text-light); margin-bottom: 0.5rem;">Available Quantity</label>
                                <div style="font-size: 1.1rem; font-weight: 600; color: #e67e22;>{{ item.quantity }} units</div>
                            </div>
                            <div>
                                <label style="display: block; font-size: 0.9rem; color: var(--text-light); margin-bottom: 0.5rem;">Current Cost</label>
                                <div style="font-size: 1.1rem; font-weight: 600; color: var(--text-dark);>{{ "%2f" |format(item.cost or 0) }}</div>
                            </div>
                            <div>
                                <label style="display: block; font-size: 0.9rem; color: var(--text-light); margin-bottom: 0.5rem;">Expiry Date</label>
                                <div style="font-size: 1.1rem; font-weight: 500; color: #e74c3c;>{{ item.expiry_date }}</div>
                            </div>
                            <div>
                                <label style="display: block; font-size: 0.9rem; color: var(--text-light); margin-bottom: 0.5rem;">Supplier</label>
                                <div style="font-size: 1.1rem; font-weight: 500; color: var(--text-dark);>{{ item.supplier or 'N/A' }}</div>
                            </div>
                        </div>/.card-content
                    </div>/.table-card
                {{ items |length }} items found
            </div>/.dashboard-content
        </main>/.main-content
    </div>/.dashboard-container
</body>
</html>
```

Orders Page

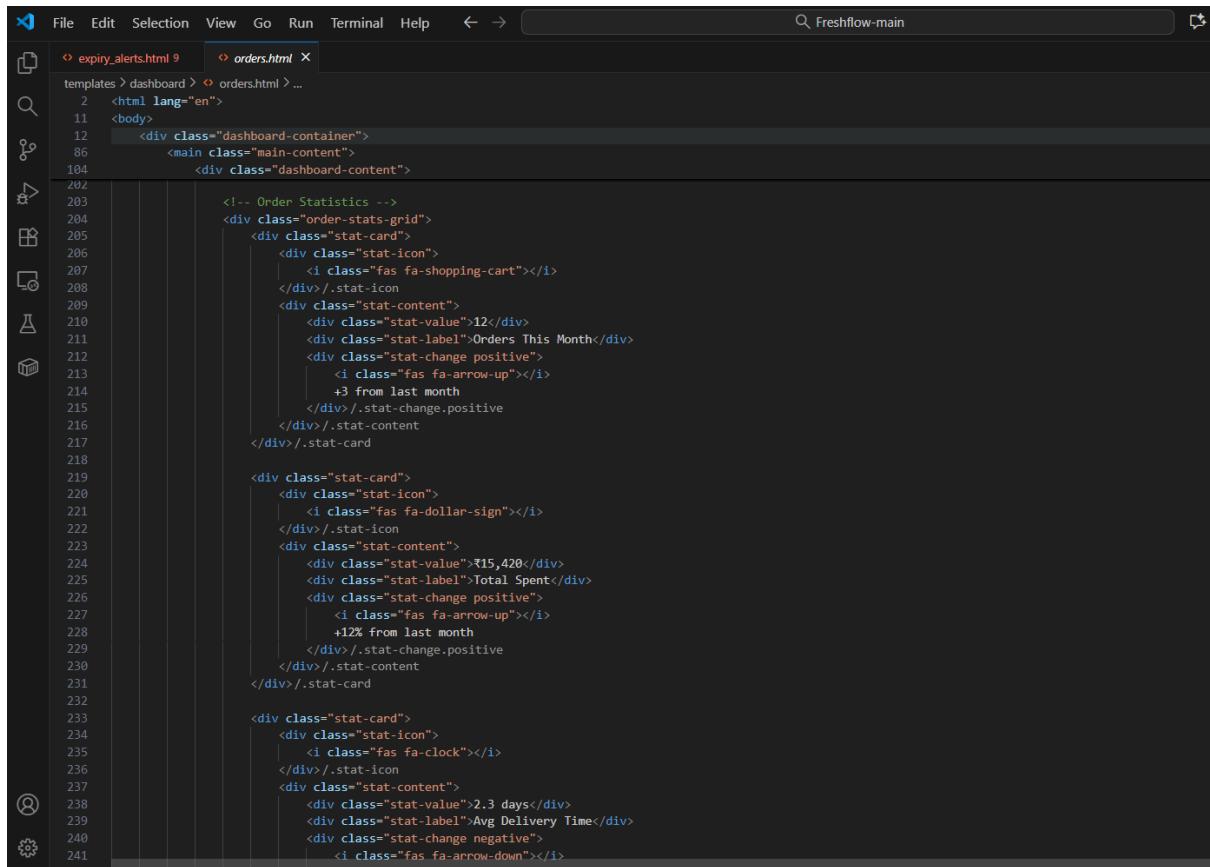
Order.html



```
<!-- expiry_alerts.html 9 --> <!-- orders.html x -->
templates > dashboard > <!-- orders.html ... -->
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Orders - FreshFlow</title>
7       <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">
8       <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
9       <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
10  </head>
11  <body>
12    <div class="dashboard-container">
13      <!-- Sidebar -->
14      <aside class="sidebar">
15        <div class="sidebar-header">
16          <div class="logo">
17            <i class="fas fa-leaf"></i>
18            <span>FreshFlow</span>
19          </div>/.logo
20        </div>/.sidebar-header
21
22        <nav class="sidebar-nav">
23          <ul class="nav-list">
24            <li class="nav-item">
25              <a href="{{ url_for('dashboard') }}" class="nav-link">
26                <i class="fas fa-home"></i>
27                <span>Dashboard</span>
28              </a>/.nav-link
29            </li>/.nav-item
30            <li class="nav-item">
31              <a href="{{ url_for('inventory') }}" class="nav-link">
32                <i class="fas fa-boxes"></i>
33                <span>Inventory</span>
34              </a>/.nav-link
35            </li>/.nav-item
36            <li class="nav-item">
37              <a href="{{ url_for('expiry_alerts') }}" class="nav-link">
38                <i class="fas fa-bell"></i>
39                <span>Expiry Alerts</span>
40              </a>/.nav-link
41            </li>/.nav-item
42            <li class="nav-item">
43              <a href="{{ url_for('analytics') }}" class="nav-link">
44                <i class="fas fa-chart-line"></i>
45                <span>Analytics</span>
46            </li>/.nav-item
47          </ul>/.nav-list
48        </nav>/.sidebar-nav
49      </aside>/.dashboard-container
50
51      <main class="main-content">
52        <!-- Orders Content -->
53        <div class="dashboard-content">
54          <!-- Reorder Suggestions -->
55          <div class="reorder-card">
56            <div class="card-header">
57              <h3>Reorder Suggestions</h3>
58              <span class="suggestion-count">{{ low_stock_items.length }} items need reordering</span>
59            </div>/.card-header
60            <div class="card-content">
61              {% if low_stock_items %}
62                <div class="reorder-items">
63                  {% for item in low_stock_items %}
64                    <div class="reorder-item">
65                      <div class="item-info">
66                        <h4>{{ item.name }}</h4>
67                        <span class="item-category">{{ item.category }}</span>
68                      </div>/.item-info
69                      <div class="item-stats">
70                        <div class="stat">
71                          <span class="stat-label">Current Stock</span>
72                          <span class="stat-value critical">{{ item.quantity }}</span>
73                        </div>/.stat
74                        <div class="stat">
75                          <span class="stat-label">Suggested Qty</span>
76                          <span class="stat-value">{{ item.quantity * 3 }}</span>
77                        </div>/.stat
78                      </div>/.item-stats
79                      <div class="item-actions">
80                        <button class="btn-primary" onclick="addToOrder('{{ item._id }}')">
81                          <i class="fas fa-plus"></i>
82                        </button>/.btn-primary
83                      </div>/.item-actions
84                    </div>/.reorder-item
85                  {% endfor %}
86                </div>/.reorder-items
87              {% else %}
88                <div class="empty-state">
89                  <i class="fas fa-check-circle"></i>
90                  <h3>All Stock Levels Good</h3>
91                  <p>No items currently need reordering</p>
92                </div>/.empty-state
93              {% endif %}
94            </div>/.card-content
95          </div>/.reorder-card
96        </div>/.dashboard-content
97
98        <!-- Footer -->
99        <div class="footer">
100          <span>FreshFlow - Your One-Stop Shop for All Your Supply Chain Needs</span>
101          <span>© 2024 FreshFlow. All Rights Reserved.</span>
102        </div>/.footer
103      </main>/.main-content
104    </div>/.dashboard-container
105
106    <div class="fixed-footer">
107      <span>FreshFlow - Your One-Stop Shop for All Your Supply Chain Needs</span>
108      <span>© 2024 FreshFlow. All Rights Reserved.</span>
109    </div>/.fixed-footer
110
111  </body>
112</html>
```



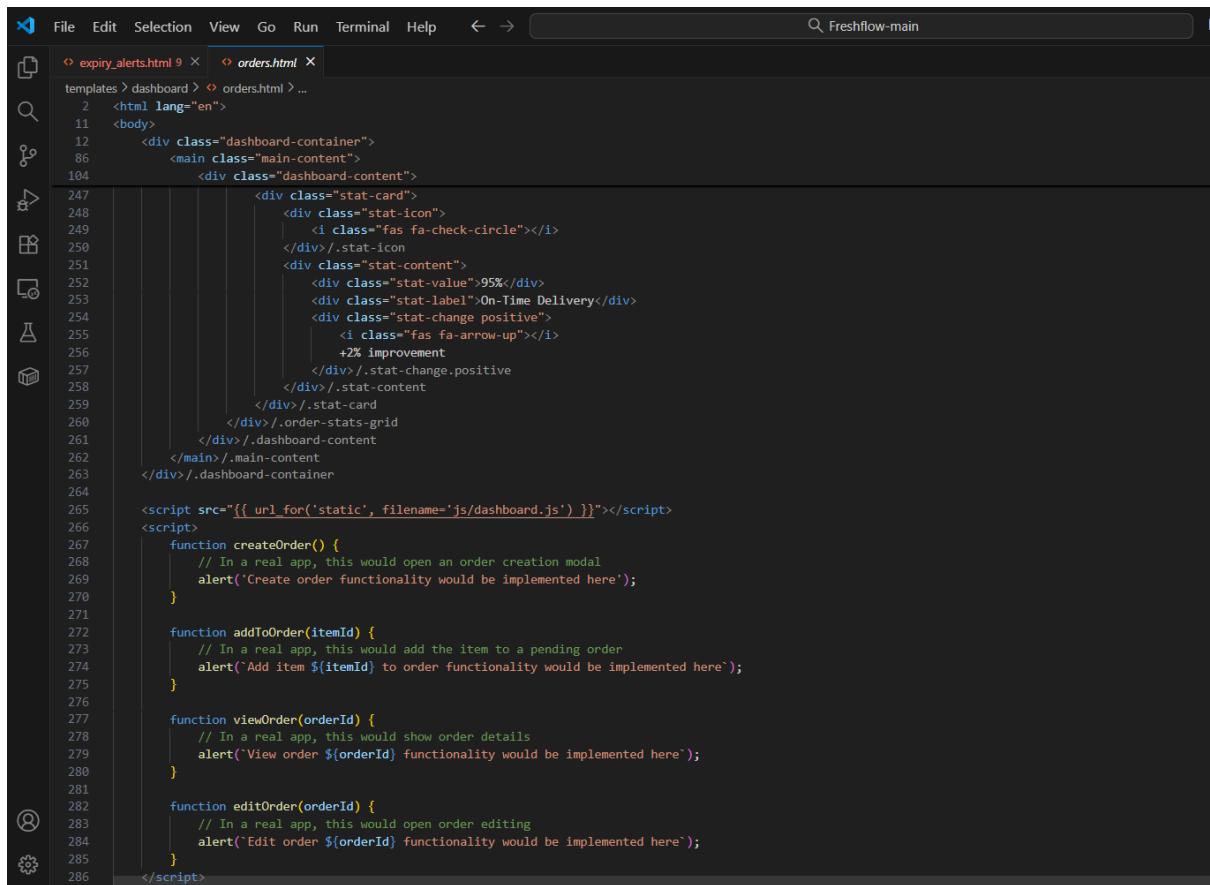
```
<!-- expiry_alerts.html 9 --> <!-- orders.html x -->
templates > dashboard > <!-- orders.html ... -->
1   <html lang="en">
2     <body>
3       <div class="dashboard-container">
4         <main class="main-content">
5           <!-- Orders Content -->
6           <div class="dashboard-content">
7             <!-- Reorder Suggestions -->
8             <div class="reorder-card">
9               <div class="card-header">
10                 <h3>Reorder Suggestions</h3>
11                 <span class="suggestion-count">{{ low_stock_items.length }} items need reordering</span>
12               </div>/.card-header
13               <div class="card-content">
14                 {% if low_stock_items %}
15                   <div class="reorder-items">
16                     {% for item in low_stock_items %}
17                       <div class="reorder-item">
18                         <div class="item-info">
19                           <h4>{{ item.name }}</h4>
20                           <span class="item-category">{{ item.category }}</span>
21                         </div>/.item-info
22                         <div class="item-stats">
23                           <div class="stat">
24                             <span class="stat-label">Current Stock</span>
25                             <span class="stat-value critical">{{ item.quantity }}</span>
26                           </div>/.stat
27                           <div class="stat">
28                             <span class="stat-label">Suggested Qty</span>
29                             <span class="stat-value">{{ item.quantity * 3 }}</span>
30                           </div>/.stat
31                         </div>/.item-stats
32                         <div class="item-actions">
33                           <button class="btn-primary" onclick="addToOrder('{{ item._id }}')">
34                             <i class="fas fa-plus"></i>
35                           </button>/.btn-primary
36                         </div>/.item-actions
37                       </div>/.reorder-item
38                     {% endfor %}
39                   </div>/.reorder-items
40                 {% else %}
41                   <div class="empty-state">
42                     <i class="fas fa-check-circle"></i>
43                     <h3>All Stock Levels Good</h3>
44                     <p>No items currently need reordering</p>
45                   </div>/.empty-state
46                 {% endif %}
47               </div>/.card-content
48             </div>/.reorder-card
49           </div>/.dashboard-content
50
51           <!-- Footer -->
52           <div class="fixed-footer">
53             <span>FreshFlow - Your One-Stop Shop for All Your Supply Chain Needs</span>
54             <span>© 2024 FreshFlow. All Rights Reserved.</span>
55           </div>/.fixed-footer
56         </main>/.main-content
57       </div>/.dashboard-container
58
59       <div class="fixed-footer">
60         <span>FreshFlow - Your One-Stop Shop for All Your Supply Chain Needs</span>
61         <span>© 2024 FreshFlow. All Rights Reserved.</span>
62       </div>/.fixed-footer
63
64     </body>
65   </html>
```



```
<!-- Order Statistics -->
<div class="order-stats-grid">
    <div class="stat-card">
        <div class="stat-icon">
            <i class="fas fa-shopping-cart"></i>
        </div>/.stat-icon
        <div class="stat-content">
            <div class="stat-value">12</div>
            <div class="stat-label">Orders This Month</div>
            <div class="stat-change positive">
                <i class="fas fa-arrow-up"></i>
                +3 from last month
            </div>/.stat-change.positive
        </div>/.stat-content
    </div>/.stat-card

    <div class="stat-card">
        <div class="stat-icon">
            <i class="fas fa-dollar-sign"></i>
        </div>/.stat-icon
        <div class="stat-content">
            <div class="stat-value">₹15,420</div>
            <div class="stat-label">Total Spent</div>
            <div class="stat-change positive">
                <i class="fas fa-arrow-up"></i>
                +12% from last month
            </div>/.stat-change.positive
        </div>/.stat-content
    </div>/.stat-card

    <div class="stat-card">
        <div class="stat-icon">
            <i class="fas fa-clock"></i>
        </div>/.stat-icon
        <div class="stat-content">
            <div class="stat-value">2.3 days</div>
            <div class="stat-label">Avg Delivery Time</div>
            <div class="stat-change negative">
                <i class="fas fa-arrow-down"></i>
                -2.3 days
            </div>/.stat-change.negative
        </div>/.stat-content
    </div>/.stat-card
</div>/.order-stats-grid
</div>/.dashboard-content
</main>/.main-content
</div>/.dashboard-container
```



```
<div class="stat-card">
    <div class="stat-icon">
        <i class="fas fa-check-circle"></i>
    </div>/.stat-icon
    <div class="stat-content">
        <div class="stat-value">95%</div>
        <div class="stat-label">On-Time Delivery</div>
        <div class="stat-change positive">
            <i class="fas fa-arrow-up"></i>
            +2% improvement
        </div>/.stat-change.positive
    </div>/.stat-content
</div>/.stat-card
</div>/.order-stats-grid
</div>/.dashboard-content
</main>/.main-content
</div>/.dashboard-container

<script src="{{ url_for('static', filename='js/dashboard.js') }}"></script>
<script>
    function createOrder() {
        // In a real app, this would open an order creation modal
        alert('Create order functionality would be implemented here');
    }

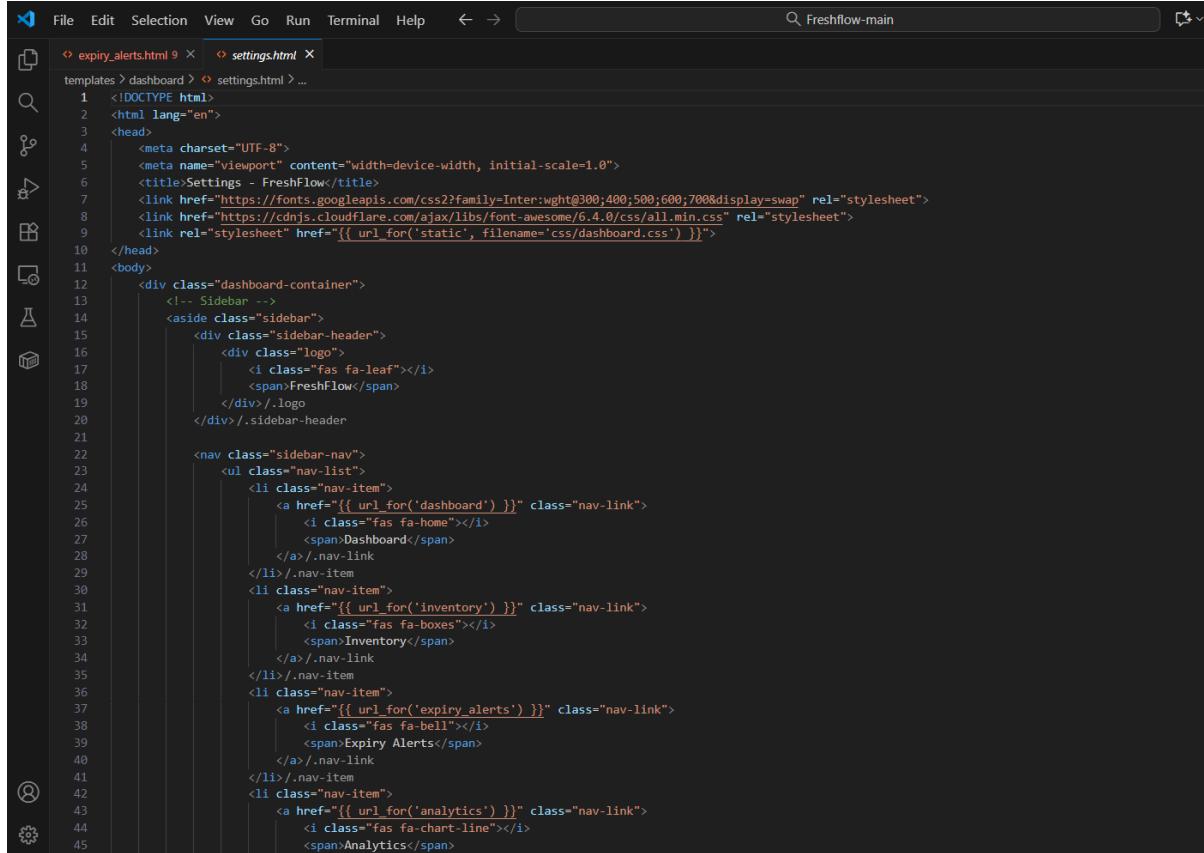
    function addToOrder(itemId) {
        // In a real app, this would add the item to a pending order
        alert(`Add item ${itemId} to order functionality would be implemented here`);
    }

    function viewOrder(orderId) {
        // In a real app, this would show order details
        alert(`View order ${orderId} functionality would be implemented here`);
    }

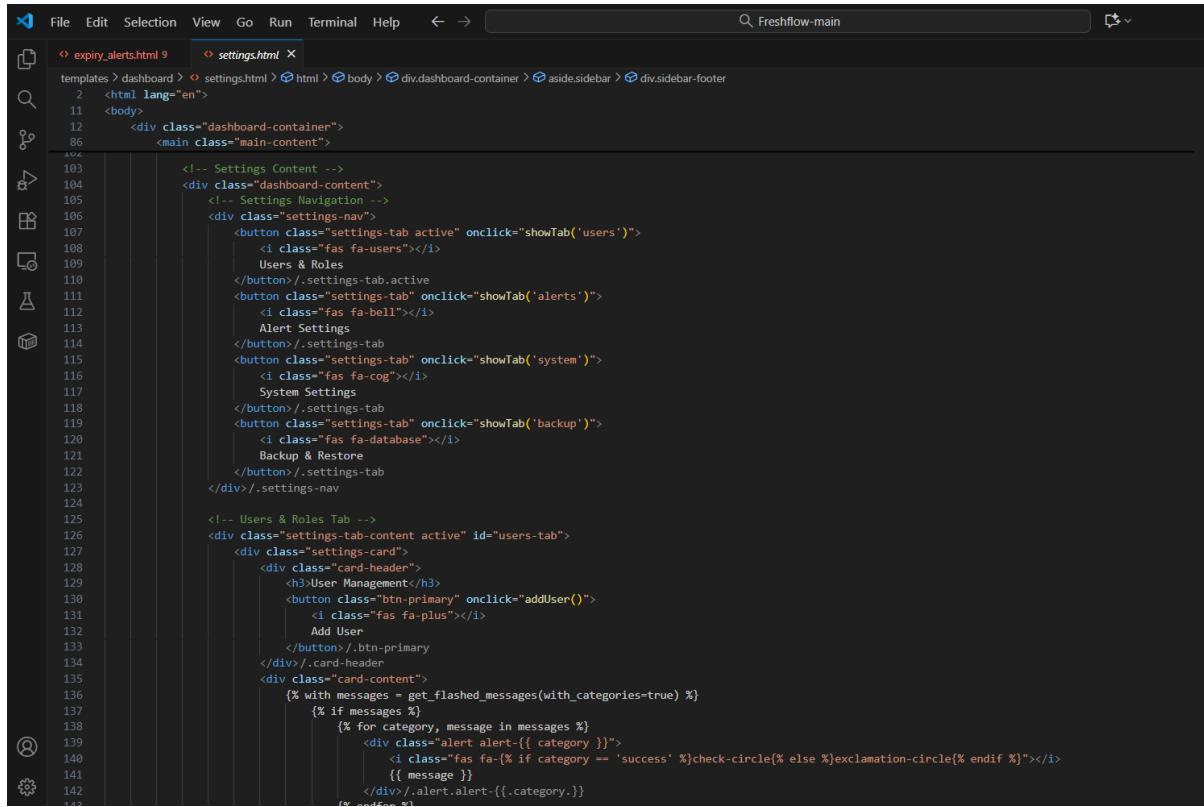
    function editOrder(orderId) {
        // In a real app, this would open order editing
        alert(`Edit order ${orderId} functionality would be implemented here`);
    }
</script>
```

Settings Page

Setting.html



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Settings - FreshFlow</title>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300,400,500,700&display=swap" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" rel="stylesheet">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.css') }}>
</head>
<body>
    <div class="dashboard-container">
        <!-- Sidebar -->
        <aside class="sidebar">
            <div class="sidebar-header">
                <div class="logo">
                    <i class="fas fa-leaf"></i>
                    <span>FreshFlow</span>
                </div>/.logo
            </div>/.sidebar-header
            <nav class="sidebar-nav">
                <ul class="nav-list">
                    <li class="nav-item">
                        <a href="{{ url_for('dashboard') }}" class="nav-link">
                            <i class="fas fa-home"></i>
                            <span>Dashboard</span>
                        </a>/.nav-link
                    </li>/.nav-item
                    <li class="nav-item">
                        <a href="{{ url_for('inventory') }}" class="nav-link">
                            <i class="fas fa-boxes"></i>
                            <span>Inventory</span>
                        </a>/.nav-link
                    </li>/.nav-item
                    <li class="nav-item">
                        <a href="{{ url_for('expiry_alerts') }}" class="nav-link">
                            <i class="fas fa-bell"></i>
                            <span>Expiry Alerts</span>
                        </a>/.nav-link
                    </li>/.nav-item
                    <li class="nav-item">
                        <a href="{{ url_for('analytics') }}" class="nav-link">
                            <i class="fas fa-chart-line"></i>
                            <span>Analytics</span>
                        </a>/.nav-link
                    </li>/.nav-item
                </ul>/.nav-list
            </nav>/.sidebar-nav
        </aside>/.dashboard-container
    </div>/.dashboard
</body>
```

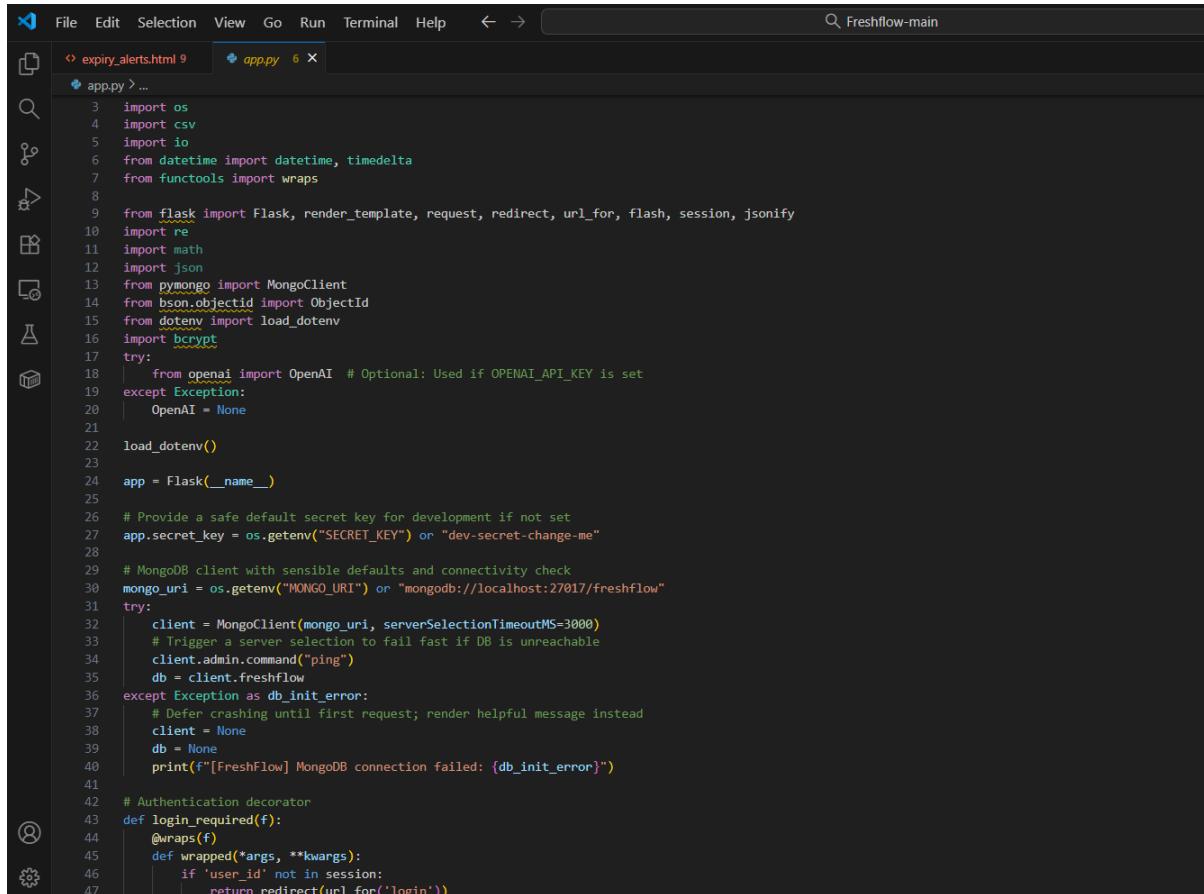


```
<!-- Settings Content -->
<div class="dashboard-content">
    <!-- Settings Navigation -->
    <div class="settings-nav">
        <button class="settings-tab active" onclick="showTab('users')">
            <i class="fas fa-users"></i>
            Users & Roles
        </button>/.settings-tab.active
        <button class="settings-tab" onclick="showTab('alerts')">
            <i class="fas fa-bell"></i>
            Alert Settings
        </button>/.settings-tab
        <button class="settings-tab" onclick="showTab('system')">
            <i class="fas fa-cog"></i>
            System Settings
        </button>/.settings-tab
        <button class="settings-tab" onclick="showTab('backup')">
            <i class="fas fa-database"></i>
            Backup & Restore
        </button>/.settings-tab
    </div>/.settings-nav

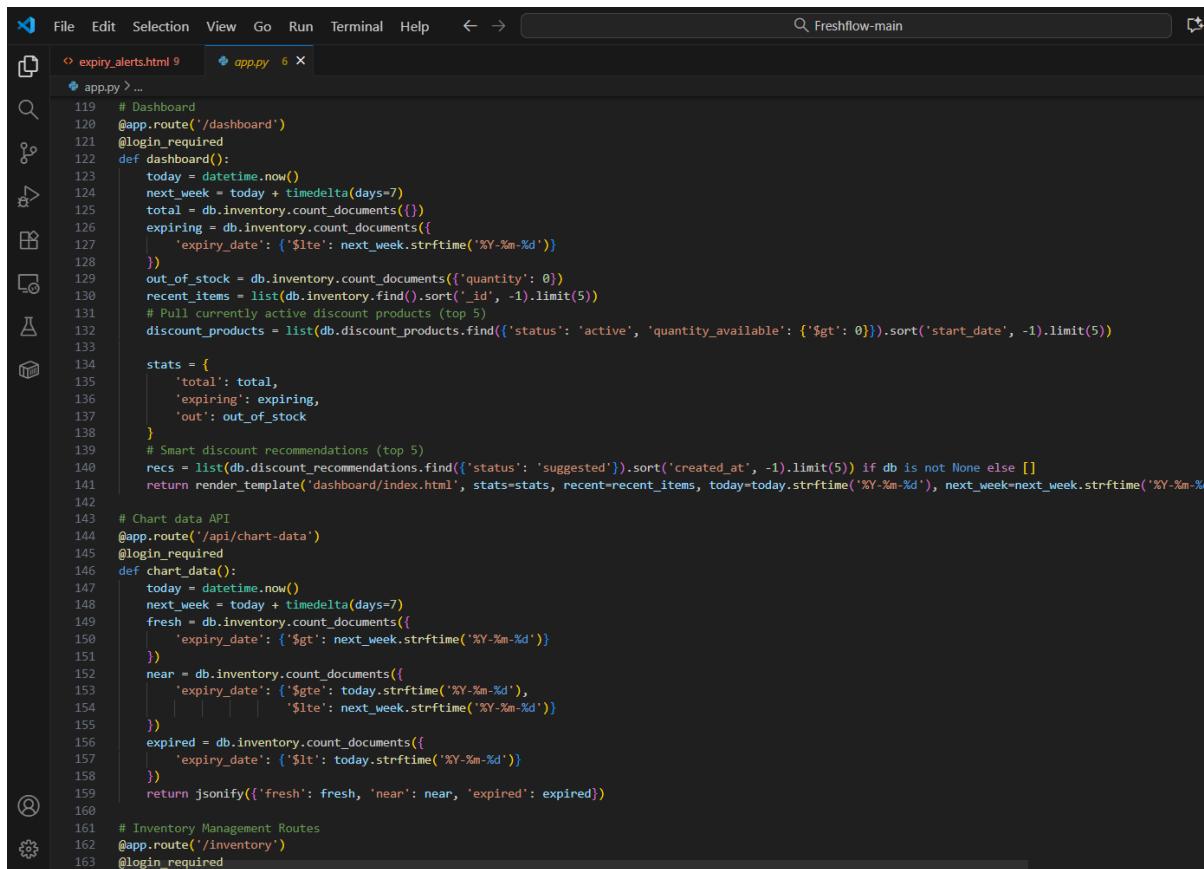
    <!-- Users & Roles Tab -->
    <div class="settings-tab-content active" id="users-tab">
        <div class="settings-card">
            <div class="card-header">
                <h3>User Management</h3>
                <button class="btn-primary" onclick="addUser()">
                    <i class="fas fa-plus"></i>
                    Add User
                </button>/.btn-primary
            </div>/.card-header
            <div class="card-content">
                {% with messages = get_flashed_messages(with_categories=true) %}
                    {% if messages %}
                        {% for category, message in messages %}
                            <div class="alert alert-{{ category }}>
                                <i class="fas fa-% if category == 'success' %check-circle(% else %)exclamation-circle(% endif %)"></i>
                                {{ message }}
                            </div>/.alert.alert-{{ category }}
                        {% endfor %}
                    {% endif %}
                {% endwith %}
            </div>/.card-content
        </div>/.settings-card
    </div>/.settings-tab-content
</div>/.dashboard-content
</div>/.dashboard
```

Backend

app.py



```
 3 import os
 4 import csv
 5 import io
 6 from datetime import datetime, timedelta
 7 from functools import wraps
 8
 9 from flask import Flask, render_template, request, redirect, url_for, flash, session, jsonify
10 import re
11 import math
12 import json
13 from pymongo import MongoClient
14 from bson.objectid import ObjectId
15 from dotenv import load_dotenv
16 import bcrypt
17 try:
18     from openai import OpenAI # Optional: Used if OPENAI_API_KEY is set
19 except Exception:
20     OpenAI = None
21
22 load_dotenv()
23
24 app = Flask(__name__)
25
26 # Provide a safe default secret key for development if not set
27 app.secret_key = os.getenv("SECRET_KEY") or "dev-secret-change-me"
28
29 # MongoDB client with sensible defaults and connectivity check
30 mongo_uri = os.getenv("MONGO_URI") or "mongodb://localhost:27017/freshflow"
31 try:
32     client = MongoClient(mongo_uri, serverSelectionTimeoutMS=3000)
33     # Trigger a server selection to fail fast if DB is unreachable
34     client.admin.command("ping")
35     db = client.freshflow
36 except Exception as db_init_error:
37     # Defer crashing until first request; render helpful message instead
38     client = None
39     db = None
40     print(f"[FreshFlow] MongoDB connection failed: {db_init_error}")
41
42 # Authentication decorator
43 def login_required(f):
44     @wraps(f)
45     def wrapped(*args, **kwargs):
46         if "user_id" not in session:
47             return redirect(url_for('login'))
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
```



```
119 # Dashboard
120 @app.route('/dashboard')
121 @login_required
122 def dashboard():
123     today = datetime.now()
124     next_week = today + timedelta(days=7)
125     total = db.inventory.count_documents({})
126     expiring = db.inventory.count_documents({
127         'expiry_date': {'$lte': next_week.strftime('%Y-%m-%d')}
128     })
129     out_of_stock = db.inventory.count_documents({'quantity': 0})
130     recent_items = list(db.inventory.find().sort('_id', -1).limit(5))
131     # Pull currently active discount products (top 5)
132     discount_products = list(db.discount_products.find({'status': 'active', 'quantity_available': {'$gt': 0}}).sort('start_date', -1).limit(5))
133
134     stats = {
135         'total': total,
136         'expiring': expiring,
137         'out': out_of_stock
138     }
139     # Smart discount recommendations (top 5)
140     recs = list(db.discount_recommendations.find({'status': 'suggested'}).sort('created_at', -1).limit(5)) if db is not None else []
141     return render_template('dashboard/index.html', stats=stats, recent_items=recent_items, today=today.strftime('%Y-%m-%d'), next_week=next_week.strftime('%Y-%m-%d'))
142
143 # Chart data API
144 @app.route('/api/chart-data')
145 @login_required
146 def chart_data():
147     today = datetime.now()
148     next_week = today + timedelta(days=7)
149     fresh = db.inventory.count_documents({
150         'expiry_date': {'$gt': next_week.strftime('%Y-%m-%d')}
151     })
152     near = db.inventory.count_documents({
153         'expiry_date': {'$gte': today.strftime('%Y-%m-%d'),
154                         '$lte': next_week.strftime('%Y-%m-%d')}
155     })
156     expired = db.inventory.count_documents({
157         'expiry_date': {'$lt': today.strftime('%Y-%m-%d')}
158     })
159     return jsonify({'fresh': fresh, 'near': near, 'expired': expired})
160
161 # Inventory Management Routes
162 @app.route('/inventory')
163 @login_required
```

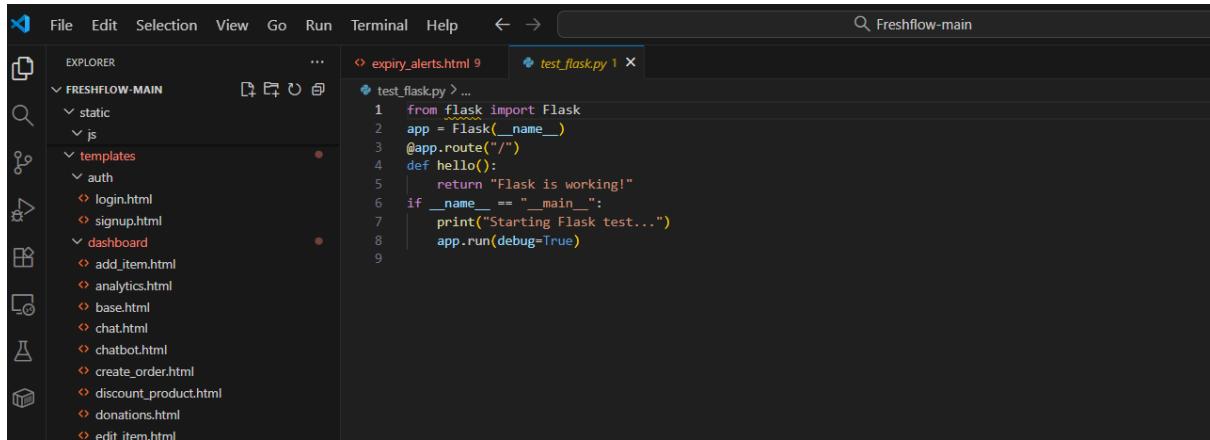
```
File Edit Selection View Go Run Terminal Help ← → Freshflow-main  
app.py 6 x  
449  
450 # Expiry Alerts Routes  
451 @app.route('/expiry-alerts')  
452 @login_required  
453 def expiry_alerts():  
454     today = datetime.now()  
455     next_3_days = today + timedelta(days=3)  
456     next_week = today + timedelta(days=7)  
457  
458     expired_items = list(db.inventory.find({  
459         'expiry_date': {'$lt': today.strftime('%Y-%m-%d')}  
460     }).sort('expiry_date', 1))  
461  
462     critical_items = list(db.inventory.find({  
463         'expiry_date': {'$gte': today.strftime('%Y-%m-%d'), '$lte': next_3_days.strftime('%Y-%m-%d')}  
464     }).sort('expiry_date', 1))  
465  
466     warning_items = list(db.inventory.find({  
467         'expiry_date': {'$gt': next_3_days.strftime('%Y-%m-%d'), '$lte': next_week.strftime('%Y-%m-%d')}  
468     }).sort('expiry_date', 1))  
469  
470     return render_template('dashboard/expiry_alerts.html',  
471         expired=expired_items,  
472         critical=critical_items,  
473         warning=warning_items)  
474  
475 # Mid Sales Discount Route  
476 @app.route('/mid-sale', methods=['GET', 'POST'])  
477 @login_required  
478 def mid_sale():  
479     if db is None:  
480         flash('Database not connected. Please ensure MongoDB is running.', 'error')  
481     return redirect(url_for('expiry_alerts'))  
482  
483     item_id = request.args.get('item_id') or (request.form.get('item_id') if request.method == 'POST' else None)  
484  
485     if not item_id:  
486         flash('No item selected for discount sale.', 'error')  
487         return redirect(url_for('expiry_alerts'))  
488  
489     try:  
490         item = db.inventory.find_one({'_id': ObjectId(item_id)})  
491         if not item:  
492             flash('Item not found!', 'error')  
493             return redirect(url_for('expiry_alerts'))  
494
```

```
File Edit Selection View Go Run Terminal Help ← → Freshflow-main  
app.py 6 x  
637 # ----- Smart Discount Recommendations -----  
638 def compute_discount_recommendations():  
639     """Compute heuristic discount recommendations and store them.  
640     This is a lightweight placeholder for ML. It focuses on near-expiry and slow-moving items.  
641     """  
642     if db is None:  
643         return 0  
644  
645     today = datetime.now()  
646     next_week = today + timedelta(days=7)  
647  
648     items = list(db.inventory.find({}))  
649  
650     # Build sales rate map: sales in last 14 days by item_id  
651     fourteen_days_ago = (today - timedelta(days=14)).strftime('%Y-%m-%d')  
652     sales = list(db.sales.find({'sale_date': {'$gte': fourteen_days_ago}}))  
653     item_id_to_sales_qty = {}  
654     for s in sales:  
655         id = str(s.get('item_id'))  
656         qty = int(s.get('sale_quantity', 0))  
657         item_id_to_sales_qty[id] = item_id_to_sales_qty.get(id, 0) + qty  
658  
659     # Clear previous suggestions (optional: keep history)  
660     db.discount_recommendations.delete_many({'status': 'suggested'})  
661  
662     created = 0  
663     for it in items:  
664         try:  
665             item_id = str(it.get('_id'))  
666             name = it.get('name', '')  
667             category = it.get('category', '')  
668             qty = int(it.get('quantity', 0))  
669             cost = float(it.get('cost', 0.0))  
670             expiry = it.get('expiry_date', '')  
671  
672             # Parse expiry and compute days to expiry  
673             days_to_expiry = 999  
674             try:  
675                 dt = datetime.strptime(str(expiry), '%Y-%m-%d')  
676                 days_to_expiry = (dt - today).days  
677             except Exception:  
678                 pass  
679  
680             # Recent sales quantity  
681             recent_sales = item_id_to_sales_qty.get(item_id, 0)
```

```
File Edit Selection View Go Run Terminal Help ← → Q Freshflow-main  
app.py > ... app.py 6 x  
732     def api_recommend_discounts():  
733         try:  
734             return jsonify({'success': False, 'error': str(e)}), 500  
740 # ----- Chatbot Routes -----  
741 @app.route('/chat')  
742 @login_required  
743 def chat():  
744     # Initialize minimal context store  
745     if 'chat_history' not in session:  
746         session['chat_history'] = []  
747     return render_template('dashboard/chat.html')  
748  
749  
750     def _format_items_table(items, fields):  
751         lines = []  
752         header = " | ".join(fields)  
753         lines.append(header)  
754         lines.append("-" * len(header))  
755         for it in items[:20]:  
756             row = []  
757             for f in fields:  
758                 row.append(str(it.get(f, '')))  
759             lines.append(" | ".join(row))  
760         return "\n".join(lines)  
761  
762  
763     def _openai_summarize(system_prompt, user_prompt):  
764         api_key = os.getenv('OPENAI_API_KEY')  
765         if not api_key or OpenAI is None:  
766             return None  
767         try:  
768             client = OpenAI(api_key=api_key)  
769             completion = client.chat.completions.create(  
770                 model=os.getenv('OPENAI_MODEL', 'gpt-4o-mini'),  
771                 messages=[  
772                     {"role": "system", "content": system_prompt},  
773                     {"role": "user", "content": user_prompt},  
774                 ],  
775                 temperature=0.2,  
776             )  
777             return completion.choices[0].message.content  
778         except Exception:  
779             return None  
780  
781  
782     def _extract_quantity_and_name(user_msg):
```

```
File Edit Selection View Go Run Terminal Help ← → Q Freshflow-main  
app.py > ... app.py 6 x  
805     def api_chat():  
806         quantity_available : (gte : 0)  
807         ).sort('start_date', -1).limit(50)  
808     if not items:  
809         reply = "No active discount products right now."  
810     else:  
811         table = _format_items_table(items, ['name', 'category', 'discount_percentage', 'discounted_price', 'quantity_available'])  
812         summary = _openai_summarize(  
813             "You summarize discount products briefly.",  
814             f"User asked: {user_msg}\nDiscounts (name|category|discount_%|price|qty):\n{table}"  
815         )  
816         reply = summary or f"Current discount products:\n{table}"  
817         session['chat_history'].append({"role": "assistant", "content": reply})  
818     return jsonify({"reply": reply})  
819  
820     # Intent: stock quantity query  
821     if 'stock' in lower or 'quantity' in lower:  
822         name = _extract_quantity_and_name(user_msg)  
823         if not name:  
824             # Try simpler extraction: words after 'of'  
825             name = user_msg  
826             item = db.inventory.find_one({'name': {'$regex': re.escape(name), '$options': 'i'}})  
827             if not item:  
828                 reply = f"I couldn't find an item matching '{name}'."  
829             else:  
830                 reply = f"{item.get('name', 'Item')}: quantity {item.get('quantity', 0)}, expiry {item.get('expiry_date', 'N/A')}, category {item.get('category', 'None')}"  
831             session['chat_history'].append({"role": "assistant", "content": reply})  
832             return jsonify({"reply": reply})  
833  
834     # Intent: add/remove quantity  
835     if ('add' in lower or 'increase' in lower or 'remove' in lower or 'decrease' in lower) and ('quantity' in lower or 'stock' in lower):  
836         qty, name = _extract_quantity_and_name(user_msg)  
837         if not name:  
838             return jsonify({"reply": "Please specify the product name."}), 400  
839         item = db.inventory.find_one({'name': {'$regex': re.escape(name), '$options': 'i'}})  
840         if not item:  
841             return jsonify({"reply": f"I couldn't find an item matching '{name}'."}), 404  
842         if qty is None:  
843             return jsonify({"reply": "Please specify a valid quantity (e.g., add 5)."}), 400  
844         delta = qty if ('add' in lower or 'increase' in lower) else -abs(qty)  
845         new_q = max(0, int(item.get('quantity', 0)) + delta)  
846         db.inventory.update_one({'_id': item['_id']}, {"$set": {"quantity": new_q}})  
847         reply = f"Updated '{item.get('name')}': quantity to {new_q}."  
848         session['chat_history'].append({"role": "assistant", "content": reply})  
849         return jsonify({"reply": reply})
```

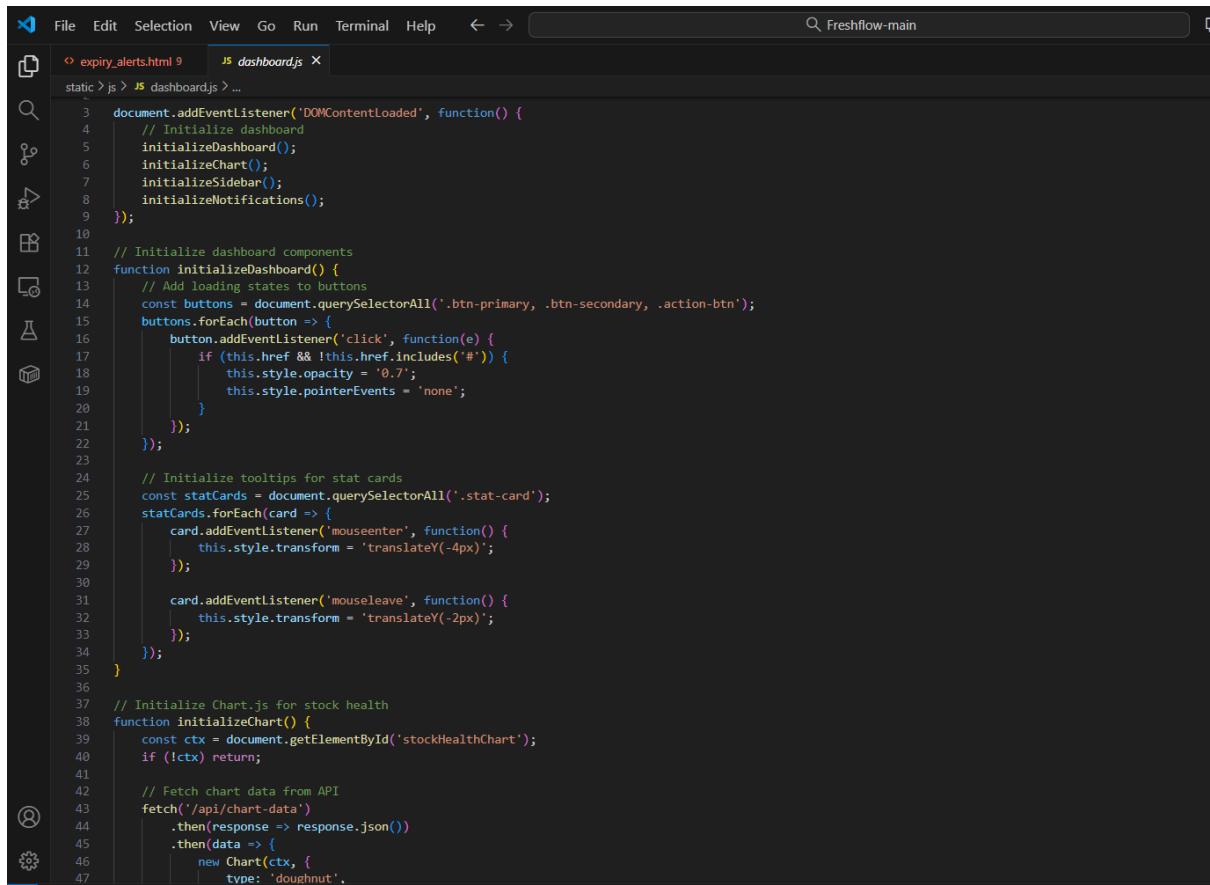
Test_flask.py



```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello():
    return "Flask is working!"
if __name__ == "__main__":
    print("Starting Flask test...")
    app.run(debug=True)
```

JavaScript codes

Dashboard.js



```
document.addEventListener('DOMContentLoaded', function() {
    // Initialize dashboard
    initializeDashboard();
    initializeChart();
    initializeSidebar();
    initializeNotifications();
});

// Initialize dashboard components
function initializeDashboard() {
    // Add loading states to buttons
    const buttons = document.querySelectorAll('.btn-primary, .btn-secondary, .action-btn');
    buttons.forEach(button => {
        button.addEventListener('click', function(e) {
            if (this.href && !this.href.includes('#')) {
                this.style.opacity = '0.7';
                this.style.pointerEvents = 'none';
            }
        });
    });
}

// Initialize tooltips for stat cards
const statCards = document.querySelectorAll('.stat-card');
statCards.forEach(card => {
    card.addEventListener('mouseenter', function() {
        this.style.transform = 'translateY(-4px)';
    });
    card.addEventListener('mouseleave', function() {
        this.style.transform = 'translateY(-2px)';
    });
});

// Initialize Chart.js for stock health
function initializeChart() {
    const ctx = document.getElementById('stockHealthChart');
    if (!ctx) return;

    // Fetch chart data from API
    fetch('/api/chart-data')
        .then(response => response.json())
        .then(data => {
            new Chart(ctx, {
                type: 'doughnut',
                data: data
            });
        })
}
```

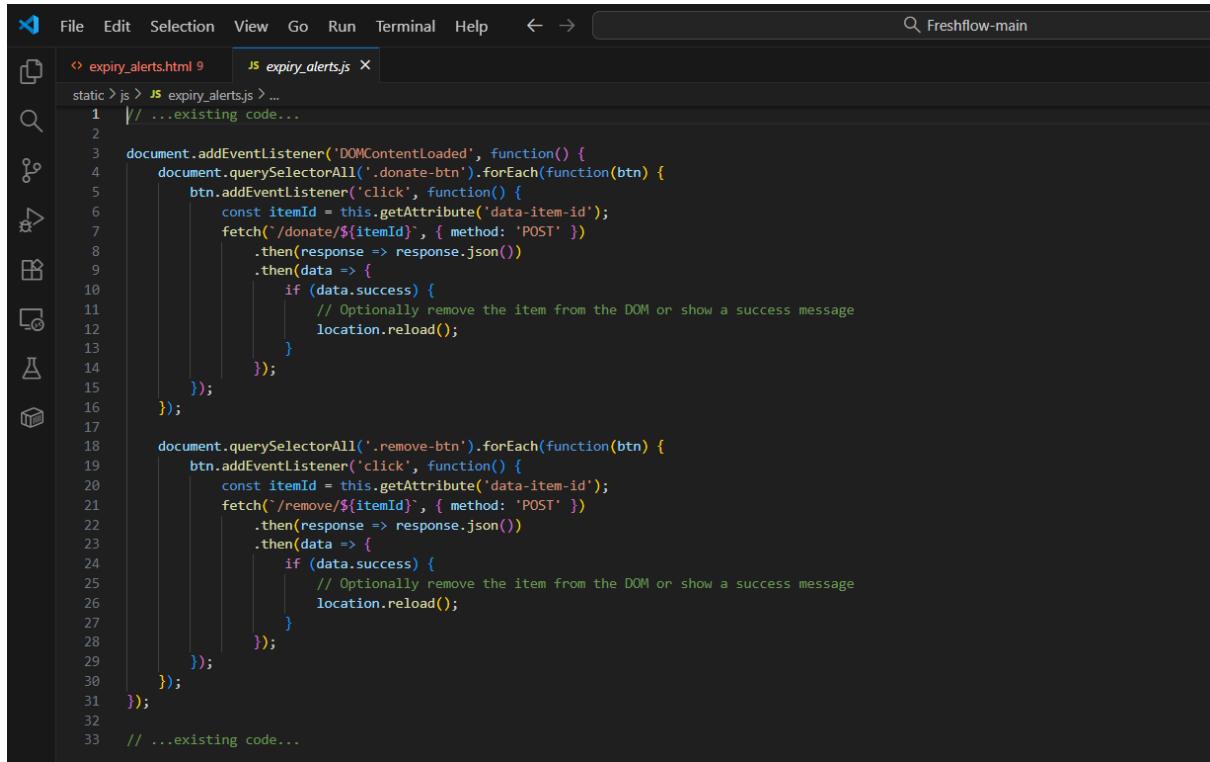
A screenshot of a code editor window titled "Freshflow-main". The active tab is "dashboard.js". The code is a JavaScript file containing a chart configuration object. It includes properties for chart type ("doughnut"), data labels, datasets, background colors, border width, and hover offset. It also defines responsive options, a legend position at the bottom, and a tooltip with specific styling like background color and title color.

```
static > js > JS dashboard.js > ...
107
108
109 // Create fallback chart with sample data
110 function createFallbackChart(ctx) {
111     new Chart(ctx, {
112         type: 'doughnut',
113         data: {
114             labels: ['Fresh', 'Near Expiry', 'Expired'],
115             datasets: [
116                 {
117                     data: [1200, 132, 12],
118                     backgroundColor: [
119                         '#27ae60',
120                         '#e67e22',
121                         '#e74c3c'
122                     ],
123                     borderWidth: 0,
124                     hoverOffset: 4
125                 ]
126             },
127             options: {
128                 responsive: true,
129                 maintainAspectRatio: false,
130                 plugins: {
131                     legend: {
132                         position: 'bottom',
133                         labels: {
134                             padding: 20,
135                             usePointStyle: true,
136                             font: {
137                                 family: 'Inter',
138                                 size: 12
139                             }
140                         }
141                     },
142                     tooltip: {
143                         backgroundColor: 'rgba(0, 0, 0, 0.8)',
144                         titleColor: '#fff',
145                         bodyColor: '#fff',
146                         borderColor: '#27ae60',
147                         borderWidth: 1,
148                         cornerRadius: 8,
149                         displayColors: false,
150                         callbacks: {
151                             label: function(context) {
152                                 const total = context.dataset.data.reduce((a, b) => a + b, 0);
153                                 return `Total: ${total} days`;
154                             }
155                         }
156                     }
157                 }
158             }
159         }
160     });
161 }
162 
```

A screenshot of a code editor window titled "Freshflow-main". The active tab is "dashboard.js". The code is a JavaScript file containing logic for initializing notifications and simulating real-time notifications. It includes functions to initialize notifications by clicking a button and to simulate notifications by updating a badge's text content every 30 seconds. It also includes search functionality with a debounce mechanism for the search input.

```
static > js > JS dashboard.js > ...
206
207 // Initialize notifications
208 function initializeNotifications() {
209     const notificationBtn = document.querySelector('.notification-btn');
210
211     if (notificationBtn) {
212         notificationBtn.addEventListener('click', function() {
213             // Toggle notification dropdown (placeholder)
214             console.log('Notifications clicked');
215             // In a real app, this would show a dropdown with notifications
216         });
217     }
218
219     // Simulate real-time notifications
220     simulateNotifications();
221 }
222
223 // Simulate real-time notifications
224 function simulateNotifications() {
225     const notificationBadge = document.querySelector('.notification-badge');
226     if (!notificationBadge) return;
227
228     // Update notification count every 30 seconds
229     setInterval(() => {
230         const currentCount = parseInt(notificationBadge.textContent);
231         const newCount = Math.max(0, currentCount + Math.floor(Math.random() * 3) - 1);
232         notificationBadge.textContent = newCount;
233
234         // Add animation for new notifications
235         if (newCount > currentCount) {
236             notificationBadge.style.animation = 'pulse 0.5s ease-in-out';
237             setTimeout(() => {
238                 notificationBadge.style.animation = '';
239             }, 500);
240         }
241     }, 30000);
242
243 // Search functionality
244 function initializeSearch() {
245     const searchInput = document.querySelector('.search-box input');
246     if (!searchInput) return;
247
248     let searchTimeout;
249     searchInput.addEventListener('input', function() {
250         clearTimeout(searchTimeout);
251     });
252 }
```

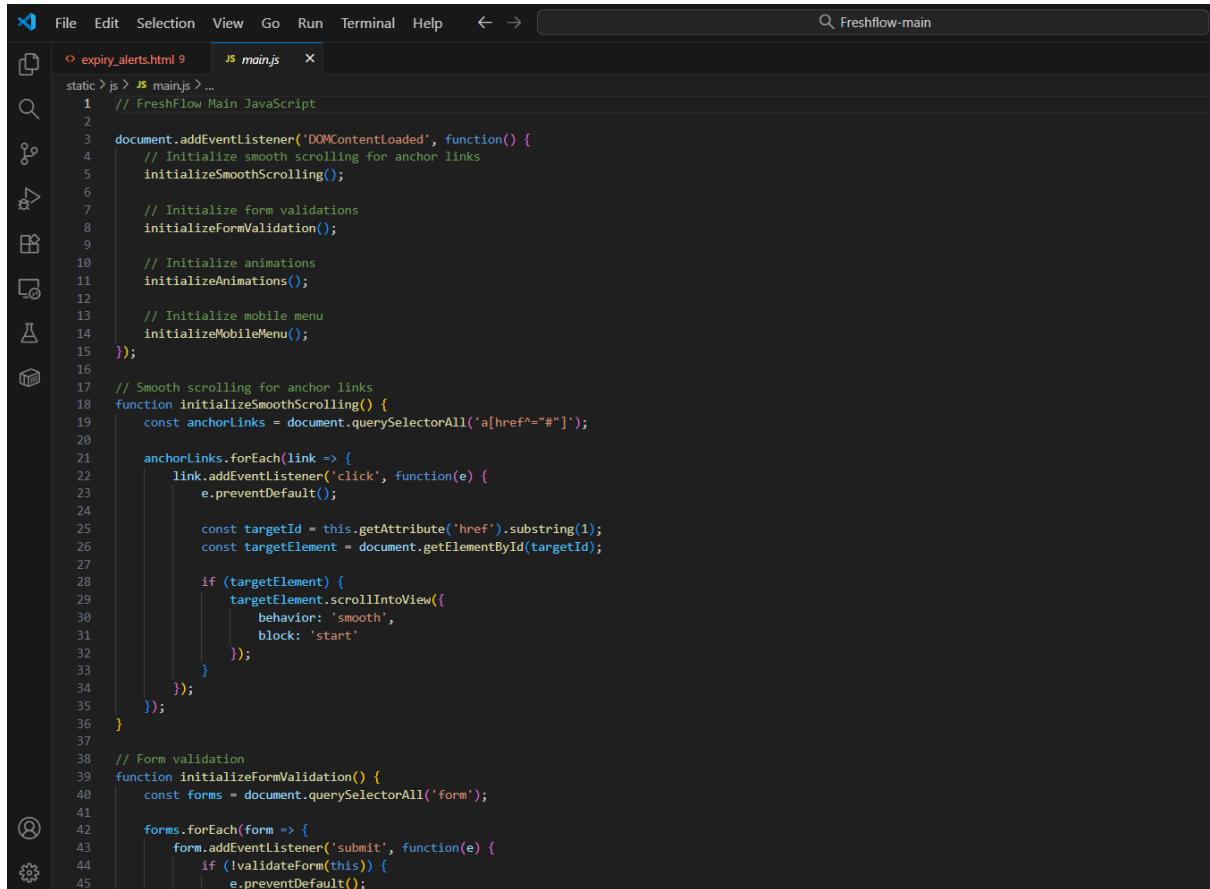
Expiry-alert.js



```
File Edit Selection View Go Run Terminal Help ⌘ ⌘ Freshflow-main

static > js > expiry_alerts.js > ...
1 // ...existing code...
2
3 document.addEventListener('DOMContentLoaded', function() {
4     document.querySelectorAll('.donate-btn').forEach(function(btn) {
5         btn.addEventListener('click', function() {
6             const itemId = this.getAttribute('data-item-id');
7             fetch('/donate/${itemId}', { method: 'POST' })
8                 .then(response => response.json())
9                 .then(data => {
10                     if (data.success) {
11                         // Optionally remove the item from the DOM or show a success message
12                         location.reload();
13                     }
14                 });
15             });
16         });
17
18 document.querySelectorAll('.remove-btn').forEach(function(btn) {
19     btn.addEventListener('click', function() {
20         const itemId = this.getAttribute('data-item-id');
21         fetch('/remove/${itemId}', { method: 'POST' })
22             .then(response => response.json())
23             .then(data => {
24                 if (data.success) {
25                     // Optionally remove the item from the DOM or show a success message
26                     location.reload();
27                 }
28             });
29         });
30     });
31 });
32
33 // ...existing code...
```

Main.js



```
File Edit Selection View Go Run Terminal Help ⌘ ⌘ Freshflow-main

static > js > main.js > ...
1 // FreshFlow Main JavaScript
2
3 document.addEventListener('DOMContentLoaded', function() {
4     // Initialize smooth scrolling for anchor links
5     initializeSmoothScrolling();
6
7     // Initialize form validations
8     initializeFormValidation();
9
10    // Initialize animations
11    initializeAnimations();
12
13    // Initialize mobile menu
14    initializeMobileMenu();
15 });
16
17 // Smooth scrolling for anchor links
18 function initializeSmoothScrolling() {
19     const anchorLinks = document.querySelectorAll('a[href^="#"]');
20
21     anchorLinks.forEach(link => {
22         link.addEventListener('click', function(e) {
23             e.preventDefault();
24
25             const targetId = this.getAttribute('href').substring(1);
26             const targetElement = document.getElementById(targetId);
27
28             if (targetElement) {
29                 targetElement.scrollIntoView({
30                     behavior: 'smooth',
31                     block: 'start'
32                 });
33             }
34         });
35     });
36 }
37
38 // Form validation
39 function initializeFormValidation() {
40     const forms = document.querySelectorAll('form');
41
42     forms.forEach(form => {
43         form.addEventListener('submit', function(e) {
44             if (!validateForm(this)) {
45                 e.preventDefault();
46             }
47         });
48     });
49 }
50
51 // Utility functions
52 function validateForm(form) {
53     const inputs = form.querySelectorAll('input');
54
55     let isValid = true;
56
57     for (let input of inputs) {
58         if (input.type === 'text' && input.value === '') {
59             isValid = false;
60             break;
61         }
62     }
63
64     return isValid;
65 }
66
67 // Initialize everything
68 initializeSmoothScrolling();
69 initializeFormValidation();
```

```
File Edit Selection View Go Run Terminal Help ← → Q, Freshflow-main

main.js

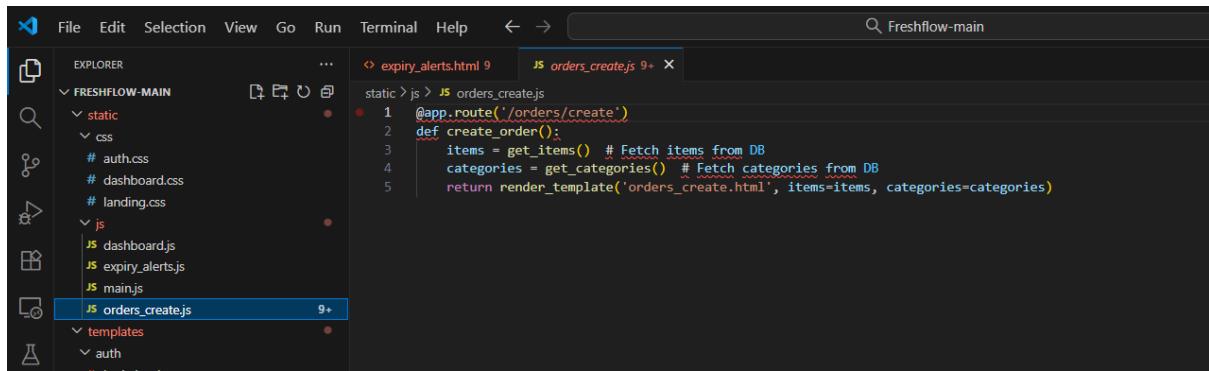
static > js > main.js > ...
63 // Validate entire form
64 function validateForm(form) {
65     let isValid = true;
66     const inputs = form.querySelectorAll('input[required], select[required], textarea[required]');
67
68     inputs.forEach(input => {
69         if (!validateField(input)) {
70             isValid = false;
71         }
72     });
73
74     return isValid;
75 }
76
77 // Validate individual field
78 function validateField(field) {
79     const value = field.value.trim();
80     const fieldName = field.name || field.id;
81     let isValid = true;
82     let errorMessage = '';
83
84     // Required field validation
85     if (field.getAttribute('required') && !value) {
86         isValid = false;
87         errorMessage = `${getFieldLabel(field)} is required`;
88     }
89
90     // Email validation
91     if (field.type === 'email' && value) {
92         const emailRegex = /^[^@\s]+@[^\s]+\.\[^@\s]+$/;
93         if (!emailRegex.test(value)) {
94             isValid = false;
95             errorMessage = 'Please enter a valid email address';
96         }
97     }
98
99     // Password validation
100    if (field.type === 'password' && value) {
101        if (value.length < 8) {
102            isValid = false;
103            errorMessage = 'Password must be at least 8 characters long';
104        }
105    }
106 }
```

```
File Edit Selection View Go Run Terminal Help ← → Q, Freshflow-main

main.js

static > js > ...
175 // Initialize mobile menu
176 function initializeMobileMenu() {
177     const mobileMenuToggle = document.querySelector('.mobile-menu-toggle');
178     const mobileMenu = document.querySelector('.mobile-menu');
179
180     if (mobileMenuToggle && mobileMenu) {
181         mobileMenuToggle.addEventListener('click', function() {
182             mobileMenu.classList.toggle('open');
183             this.classList.toggle('active');
184         });
185
186         // Close menu when clicking outside
187         document.addEventListener('click', function(e) {
188             if (!mobileMenu.contains(e.target) && !mobileMenuToggle.contains(e.target)) {
189                 mobileMenu.classList.remove('open');
190                 mobileMenuToggle.classList.remove('active');
191             }
192         });
193     }
194 }
195
196 // Utility functions
197 function debounce(func, wait) {
198     let timeout;
199     return function executedFunction(...args) {
200         const later = () => {
201             clearTimeout(timeout);
202             func(...args);
203         };
204         clearTimeout(timeout);
205         timeout = setTimeout(later, wait);
206     };
207 }
208
209 function throttle(func, limit) {
210     let inThrottle;
211     return function() {
212         const args = arguments;
213         const context = this;
214         if (!inThrottle) {
215             func.apply(context, args);
216             inThrottle = true;
217             setTimeout(() => inThrottle = false, limit);
218         }
219     };
220 }
```

Orders_create.js



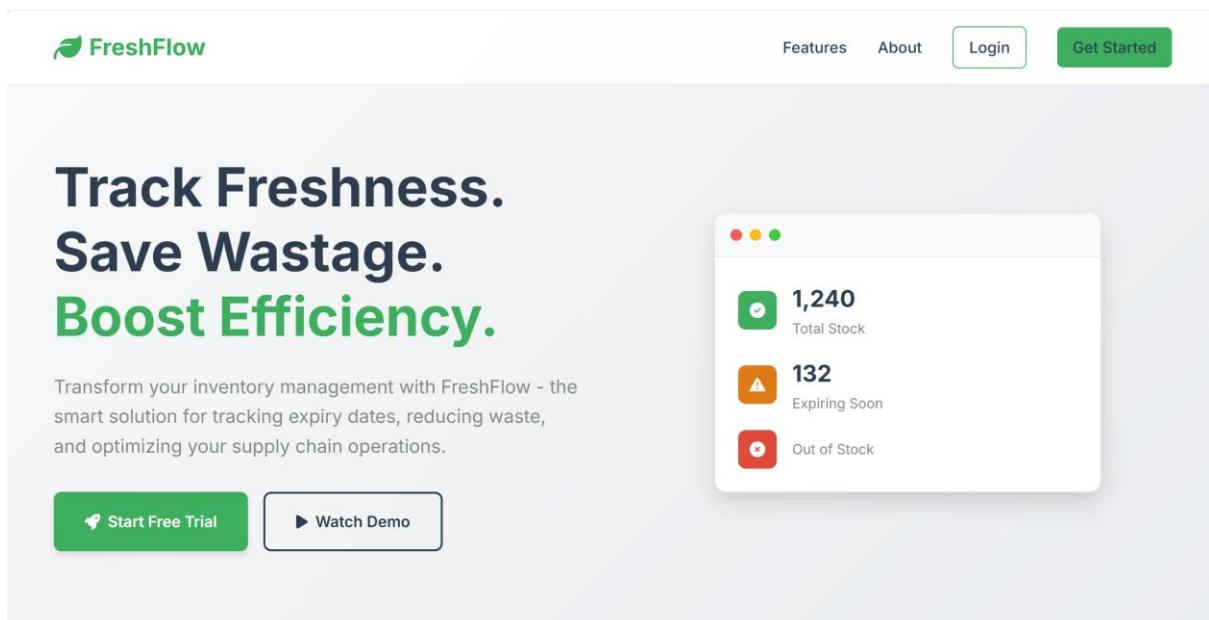
```
File Edit Selection View Go Run Terminal Help < > Freshflow-main
EXPLORER FRESHFLOW-MAIN ...
static
  static
    # auth.css
    # dashboard.css
    # landing.css
  js
    JS dashboard.js
    JS expiry_alerts.js
    JS main.js
  JS orders_create.js 9+
  templates
    auth
JS orders_create.js 9+ ...
```

```
static > js > JS orders_create.js
1 @app.route('/orders/create')
2 def create_order():
3     items = get_items() # Fetch items from DB
4     categories = get_categories() # Fetch categories from DB
5     return render_template('orders_create.html', items=items, categories=categories)
```

OUTPUT:

```
PS C:\Users\P SAI ESWARI\OneDrive\Documents\Freshflow-main (2)\Freshflow-main\Freshflow-main> python app.py
Starting FreshFlow...
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a
production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
Starting FreshFlow...
* Debugger is active!
* Debugger PIN: 559-594-947
127.0.0.1 - - [05/Nov/2025 20:34:49] "GET / HTTP/1.1" 200 -
```

HOME PAGE:



FreshFlow

Features About Login Get Started

Track Freshness. Save Wastage. Boost Efficiency.

Transform your inventory management with FreshFlow - the smart solution for tracking expiry dates, reducing waste, and optimizing your supply chain operations.

1,240 Total Stock
132 Expiring Soon
Out of Stock

Start Free Trial Watch Demo

Powerful Features for Modern Inventory Management

Everything you need to manage your inventory efficiently and reduce waste



Smart Inventory Tracking

Real-time monitoring of stock levels with automated expiry date tracking and alerts.



Expiry Alerts

Never miss an expiry date with intelligent notifications via email, SMS, or dashboard alerts.



Analytics & Reports

Comprehensive insights into waste patterns, consumption trends, and optimization opportunities.



Donation Management

Connect with NGOs and charities to donate expiring items, reducing waste and helping communities.



Auto Reorder

Intelligent reorder suggestions based on consumption patterns and minimum stock levels.



Role-Based Access

Secure multi-user access with different permission levels for admins, managers, and staff.



Features

About

Login

Get Started

Ready to Transform Your Inventory Management?

Join hundreds of businesses already using FreshFlow to reduce waste and boost efficiency.

Get Started Today



Track Freshness. Save Wastage. Boost Efficiency.

Product

Features
Pricing
Demo

Company

About
Contact
Careers

Support

Help Center
Documentation
Status

CREATE ACCOUNT:

 FreshFlow

Create Account

Join FreshFlow and start optimizing your inventory management

Username

Email Address

Password

Password strength

Role

 Select your role

I agree to the [Terms of Service](#) and [Privacy Policy](#)

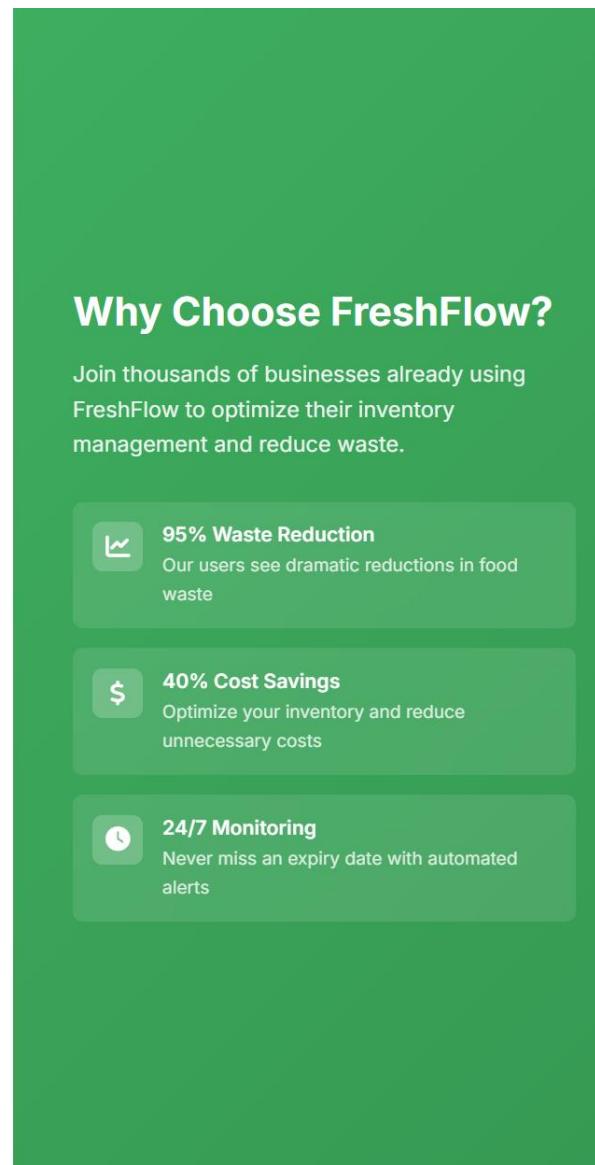
 **Create Account**

or

 **Continue with Google**

Already have an account? [Sign in here](#)

 [Back to Home](#)



Role

 Select your role

Select your role

- Admin
- Manager
- Warehouse Staff

 **Continue with Google**

Already have an account? [Sign in here](#)

 [Back to Home](#)

SIGN IN

 FreshFlow

Welcome Back

Sign in to your account to continue

Username or Email

Password

Remember me [Forgot Password?](#)

→ Sign In

or

G Continue with Google

Don't have an account? [Sign up here](#)

[← Back to Home](#)

Track Freshness. Save Wastage. Boost Efficiency.

Join thousands of businesses using FreshFlow to optimize their inventory management and reduce waste.

- ✓ Real-time inventory tracking
- ✓ Automated expiry alerts
- ✓ Smart analytics & reports
- ✓ Donation management

DASHBOARD:

 FreshFlow

Dashboard

Search inventory... 2046 admin

Stock Health Overview



50016 Total Stock +12% from last month

2846 Expiring Soon -8% from last week

1 Out of Stock - No change

50 Donation Suggested +5 new items

Recent Inventory [View All](#)

Item Name	Category	Quantity	Expiry Date	Status
Pear	Fruit	968	2026-02-12	Fresh
Bun	Bakery	992	2025-12-24	Fresh
Pork	Meat	565	2026-05-01	Fresh
Pear	Fruit	693	2026-03-08	Fresh
Pear	Fruit	589	2025-12-25	Fresh

Discount Products [View All](#)

Item	Category	Original Price	Discount %	Discounted Price	Qty Avail
Yogurt	Dairy	₹388.98	10.00%	₹350.08	10
Cake	Bakery	₹223.20	20.00%	₹178.56	2

admin Admin

Stock Health Overview

Fresh: 47170 items (94.3%)

Legend: Fresh (Green), Near Expiry (Orange), Expired (Red)

Recent Inventory

Item Name	Category	Quantity	Expiry Date	Status
Pear	Fruit	968	2026-02-12	Fresh
Bun	Bakery	992	2025-12-24	Fresh
Pork	Meat	565	2026-03-01	Fresh
Pear	Fruit	693	2026-03-08	Fresh
Pear	Fruit	589	2025-12-25	Fresh

Discount Products

Item	Category	Original Price	Discount %	Discounted Price	Qty Avail
Yogurt	Dairy	₹388.98	10.00%	₹350.08	10
Cake	Bakery	₹223.20	20.00%	₹178.56	2

Smart Discount Recommendations

Item	Category	Qty	Current Price	Suggested %	Suggested Price	Reason	Action
Pork	Meat	592	₹492.03	20.0%	₹393.62	slow sales: 0.00/day, overstock: 592 units	Apply
Donut	Bakery	312	₹457.32	20.0%	₹365.86	slow sales: 0.00/day, overstock: 312 units	Apply
Potato	Vegetable	71	₹149.82	20.0%	₹119.86	slow sales: 0.00/day, overstock: 71 units	Apply
Yogurt	Dairy	563	₹458.26	20.0%	₹366.61	slow sales: 0.00/day, overstock: 563 units	Apply
Yogurt	Dairy	297	₹214.12	20.0%	₹171.30	slow sales: 0.00/day, overstock: 297 units	Apply

Quick Actions

[Add Item](#)

[Bulk Upload](#)

[View Alerts](#)

[Donate Items](#)

[Chatbot](#)

Recent Alerts

			View All
	Milk Expired	2 hours ago	
	Bread Expiring Soon	1 day ago	
	Low Stock: Apples	2 days ago	

SMART DISCOUNT RECOMMENDATIONS:

Smart Discount Recommendations

Item	Category	Qty	Current Price	Suggested %	Suggested Price	Reason	Action
Pork	Meat	592	₹492.03	20.0%	₹393.62	slow sales: 0.00/day, overstock: 592 units	Apply
Donut	Bakery	312	₹457.32	20.0%	₹365.86	slow sales: 0.00/day, overstock: 312 units	Apply
Potato	Vegetable	71	₹149.82	20.0%	₹119.86	slow sales: 0.00/day, overstock: 71 units	Apply
Yogurt	Dairy	563	₹458.26	20.0%	₹366.61	slow sales: 0.00/day, overstock: 563 units	Apply
Yogurt	Dairy	297	₹214.12	20.0%	₹171.30	slow sales: 0.00/day, overstock: 297 units	Apply

MID SALES DISCOUNT:

FreshFlow

Mid Sales Discount

Back to Alerts

Item Information

Item Name	Category	Available Quantity	Current Cost	Expiry Date
Pork	Meat	592 units	₹492.03	2026-04-30
Supplier	N/A			

Apply Mid Sales Discount

Discount Percentage (%)

20

Enter discount percentage (0-100%)

Discounted Price (₹)

393.62

Final price after discount

Sale Quantity

30

Maximum: 592 units

Sale Date

05-11-2025

Notes (Optional)

Sale Quantity

30

Maximum: 592 units

Sale Date

05-11-2025

Notes (Optional)

Add any additional notes about this discount sale...

Sale Summary

Original Price:	₹492.03
Discount:	20%
Discounted Price:	₹393.62
Quantity:	30 units
Total Revenue:	₹11808.72

Cancel

Apply Discount Sale

DISCOUNT PRODUCTS LIST

FreshFlow

Discount Products

Back to Dashboard

Currently Available Discounts

Item	Category	Original Price	Discount %	Discounted Price	Qty Available	Start Date
Pork	Meat	₹492.03	20.00%	₹393.62	30	2025-11-05
Yogurt	Dairy	₹388.98	10.00%	₹350.08	10	2025-11-03
Cake	Bakery	₹223.20	20.00%	₹178.56	2	2025-11-03

INVENTORY MANAGEMENT

The screenshot shows the FreshFlow Inventory Management interface. On the left is a sidebar with navigation links: Dashboard, Inventory (selected), Expiry Alerts, Analytics, Orders, Donations, and Settings. The main area is titled "Inventory Management" and contains a "Filters" section with dropdowns for Category (All Categories), Status (All Status), and Quantity (All Quantities). Below the filters is a message: "Mid sales discount applied successfully! 30 units listed at 20.0% discount." A table titled "Inventory Items (50016 items)" lists items like 7up, Apple, and various types of Apples with columns for Item Name, Category, Quantity, Expiry Date, Supplier, Cost, Status, and Actions. A green "Upload CSV" button is in the top right.

CSV FILE UPLOAD:

This screenshot shows a modal dialog titled "Upload CSV File" overlaid on the inventory management page. The dialog includes a "CSV Format Requirements" section with instructions and a "Select CSV File:" input field with a "Choose File" button. A "Cancel" button and a "Upload & Import" button are at the bottom. The background shows the same inventory list as the previous screenshot.

EXPIRY ALERTS:

FreshFlow

Expiry Alerts

● 1761 Critical ▲ 1085 Warning ✓ Mark All Read

Expired Items (661)

Item	Quantity	Supplier	Cost	Action
Bread	15 units	Bakery Co	₹ 3.00	Mid Sales Discount Donate Remove View Details
Bread	15 units	Bakery Co	₹ 3.00	Mid Sales Discount Donate Remove View Details
Bananas	50 units	Fresh Foods	₹ 1.20	Mid Sales Discount Donate Remove View Details
Bananas	50 units	Fresh Foods	₹ 1.20	Mid Sales Discount Donate Remove View Details

Expiry Alerts

● 1761 Critical ▲ 1085 Warning ✓ Mark All Read

Critical Alert - Expiring in 3 Days (1100)

Item	Quantity	Supplier	Cost	Action
Onion	809 units	N/A	₹ 465.62	Mid Sales Discount Donate Reorder View Details
Cucumber	1 units	N/A	₹ 105.66	Mid Sales Discount Donate Reorder View Details
Chicken Breast	378 units	N/A	₹ 494.72	Mid Sales Discount Donate Reorder View Details

Expiry Alerts

● 1761 Critical ▲ 1085 Warning ✓ Mark All Read

Warning - Expiring in 7 Days (1085)

Item	Quantity	Supplier	Cost	Action
Cucumber	904 units	N/A	₹ 124.02	Donate Reorder View Details
Pear	249 units	N/A	₹ 139.20	Donate Reorder View Details

ANALYTICS & REPORTS

FreshFlow

Analytics & Reports

Last 30 Days Export Report

Dashboard Inventory Expiry Alerts Analytics Orders Donations Settings

admin Admin

1.3% Waste Percentage (↓ -5.2% from last month)

50016 Total Items (↑ +12% from last month)

661 Expired Items (↑ +3 from last week)

₹2,450 Cost Savings (↑ +18% from donations)

Waste Trend Over Time

Category Distribution

Waste by Category

Category Performance

Category	Total Items	Expired	Waste %	Trend
Bakery	9997	120	1%	↓ Good
Beverages	1	0	0%	↓ Good
Dairy	10098	151	1%	↓ Good
Fruit	10071	134	1%	↓ Good
Fruits	4	4	100.0%	↑ Poor
Grains	2	2	100.0%	↑ Poor
Meat	9986	130	1%	↓ Good
Snacks	2	1	50%	↑ Poor
Vegetable	9855	119	1%	↓ Good

Top Waste Contributors

Category	Waste Percentage (%)
Dairy Products	25.3%
Fresh Fruits	18.7%
Vegetables	15.2%

Recommendations

AI Generated

- Optimize Dairy Ordering (High Impact)
Reduce dairy orders by 15% to minimize waste. Current waste rate is 25.3%.
- Implement FIFO System (Medium Impact)
Use First-In-First-Out rotation to reduce expiry issues by 20%.
- Increase Donation Program (High Impact)
Expand donation partnerships to reduce waste and improve community impact.

ORDERS & REORDERS:

FreshFlow

- Dashboard
- Inventory
- Expiry Alerts
- Analytics
- Orders**
- Donations
- Settings

admin Admin

Orders & Reorders

Reorder Suggestions

512 Items need reordering

Item	Category	Current Stock	Suggested Qty	Action
Fish	Meat	0	0	+ Add to Order
Mutton	Meat	1	3	+ Add to Order
Panner	Dairy	1	3	+ Add to Order
Spinach	Vegetable	1	3	+ Add to Order
Cheese	Dairy	1	3	+ Add to Order
Broccoli	Vegetable	1	3	+ Add to Order
Cake	Bakery	1	3	+ Add to Order

Recent Orders

[View All Orders](#)

Order ID	Supplier	Items	Total	Status	Date	Actions
68ca82ab58981e2aed7e692c	Dairy Direct	<built-in method items of dict object at 0x000001EACC379D80>	₹0.00	Pending	2025-09-17	🕒 ⏺
68cac9db58981e2aed7e692d	Meat Market	<built-in method items of dict object at 0x000001EACC379E40>	₹0.00	Pending	2025-09-17	🕒 ⏺
68c9757aad52ea53ab57e04c	Bakery Co	<built-in method items of dict object at 0x000001EACF4C1EC0>	₹0.00	Pending	2025-09-16	🕒 ⏺
68c97591ad52ea53ab57e04d	Bakery Co	<built-in method items of dict object at 0x000001EACFC00B40>	₹0.00	Pending	2025-09-16	🕒 ⏺
68c9925451c1d5153920cf7a	Dairy Direct	<built-in method items of dict object at 0x000001EACA54EFC0>	₹0.00	Pending	2025-09-16	🕒 ⏺
68c9936751c1d5153920cf7d	Bakery Co	<built-in method items of dict object at 0x000001EACC393E40>	₹0.00	Pending	2025-09-16	🕒 ⏺

12
Orders This Month
↑ +3 from last month

₹15,420
Total Spent
↑ +12% from last month

2.3 days
Avg Delivery Time
↓ -0.5 days improvement

95%
On-Time Delivery
↑ +2% improvement

DONATIONS:

FreshFlow

- Dashboard
- Inventory
- Expiry Alerts
- Analytics
- Orders
- Donations**
- Settings

admin Admin

Donation Management

2185 items ready

2185 items available

Items Ready for Donation

2185 items available

Item	Category	Units	Expires	Value	Action
Carrot	Vegetable	331 units	Expires 2025-11-05	₹ 159.70	Expiring Soon
Mango	Fruit	452 units	Expires 2025-11-05	₹ 352.36	Expiring Soon
Spinach	Vegetable	419 units	Expires 2025-11-05	₹ 289.55	Expiring Soon
Pork	Meat	945 units	Expires 2025-11-05	₹ 15.37	Expiring Soon
Onion	Vegetable	759 units	Expires 2025-11-05	₹ 221.34	Expiring Soon
Eggs	Dairy	540 units	Expires 2025-11-05	₹ 313.80	Expiring Soon
Carrot	Vegetable	657 units	Expires 2025-11-05	₹ 476.64	Expiring Soon
Mutton	Meat	495 units	Expires 2025-11-05	₹ 152.13	Expiring Soon

Donation Management

2185 items ready | + Create Donation

Select NGO Partner: Choose an NGO partner

NGO Partners

Partner	Donations	Items
Food Bank Foundation contact@foodbank.org +1-555-0123	24	156
Community Kitchen info@communitykitchen.org +1-555-0456	24	156
Hope Shelter donations@hopeshelter.org +1-555-0789	24	156

Recent Donations

[View All](#)

Donation ID	NGO Partner	Items	Value	Date	Status	Actions
DON001	Food Bank Foundation	12	₹245.50	2024-01-15	Completed	Edit Delete
DON002	Community Kitchen	8	₹189.20	2024-01-12	Pending	Edit Delete

156
 Items Donated
↑ +24 this month

₹2,450
 Value Donated
↑ ₹450 this month

3
 NGO Partners
— No change

95%
 Waste Reduction
↑ +5% improvement

SETTINGS:

Settings

[Save Changes](#)

[Users & Roles](#) [Alert Settings](#) [System Settings](#) [Backup & Restore](#)

User Management

Username	Email	Role	Last Login	Status	Actions
admin	admin@freshflow.com	Admin	2024-01-15	Active	Edit Delete
Sai Eswari P	sai2004eswari@gmail.com	Manager	2024-01-15	Active	Edit Delete
raki	nprakash68@gmail.com	Warehouse Staff	2024-01-15	Active	Edit Delete
priya	saleswarri1118@gmail.com	Warehouse Staff	2024-01-15	Active	Edit Delete

FreshFlow Settings - Alert Configuration

Save Changes

- Dashboard
- Inventory
- Expiry Alerts
- Analytics
- Orders
- Donations
- Settings

Alert Configuration

Email Notifications
Receive expiry alerts via email

SMS Notifications
Receive critical alerts via SMS

WhatsApp Notifications
Receive alerts via WhatsApp

Alert Frequency
How often to send expiry alerts

Alert Threshold
Days before expiry to send alerts

FreshFlow Settings - System Configuration

Save Changes

- Dashboard
- Inventory
- Expiry Alerts
- Analytics
- Orders
- Donations
- Settings

System Configuration

Database Connection
MongoDB connection settings

Auto Backup
Automatically backup data daily

Maintenance Mode
Enable maintenance mode

Log Level
System logging level

FreshFlow Settings - Backup & Restore

Save Changes

- Dashboard
- Inventory
- Expiry Alerts
- Analytics
- Orders
- Donations
- Settings

Backup & Restore

Create Backup
Create a full backup of all data

Restore Backup
Restore data from a backup file

Recent Backups
View and manage recent backups

CONCLUSION:

Fresh Flow is a complete smart inventory management solution for organizations, especially those dealing with perishable goods, that had their stock handling methods modernized. It provides a single system that incorporates data-driven decision-making, automation, and simplicity, which in turn reduces waste, increases profit margins, and raises operational efficiency.

FreshFlow, through advanced analytics, expiry alerts, discount recommendations, and conversational support, shows the impact of AI in inventory management systems to challenge existing practices.

REFERENCES :

1. IEEE Xplore - AI-Powered Smart Inventory Management
<https://ieeexplore.ieee.org/document/11070285/>
Research paper presenting AI-driven inventory system using machine learning and predictive analytics.
2. Rapid Innovation - AI-Powered Inventory Management Ultimate Guide 2024
<https://www.rapidinnovation.io/post/ai-in-inventory-management>
Detailed guide explaining AI features like predictive analytics, automation, and real-time tracking.
3. MindInventory - AI In Inventory Management: A Detailed Overview
<https://www.mindinventory.com/blog/ai-inventory-management/>
Explains how AI and ML transform inventory control for improved efficiency and accuracy.
4. IRJMETS - AI-BASED INVENTORY SYSTEM (PDF)
https://www.irjmets.com/uploadedfiles/paper/issue 1 january 2025/66064/final/fin_irjmets1736415256.pdf
Research exploring AI models to improve forecasting and anomaly detection in inventory.
5. Persistence Market Research — AI-Powered Inventory Management
<https://www.persistencemarketresearch.com/blog/ai-powered-inventory-management-for-smarter-operations.asp>
Insights on AI improving forecasting, automating stock control, and reducing costs.

6. DevOpsSchool — Top 10 AI Inventory Management Tools in 2025
<https://www.devopsschool.com/blog/top-10-ai-inventory-management-tools-in-2025-features-pros-cons-comparison/>
A comparison of leading AI-based inventory software solutions.
7. LinkedIn Article — AI in Inventory Management 2025: A Detailed Guide
<https://www.linkedin.com/pulse/ai-inventory-management-2025-detailed-guide-the-algorithm-loxaf/>
Detailed breakdown of AI's transformative impact on inventory processes.
8. SuperAGI — Top 10 AI Inventory Systems for 2025
<https://superagi.com/top-10-ai-inventory-management-systems-for-2025-a-comprehensive-guide-to-forecasting-and-optimization/>
Comprehensive report on leading AI inventory platforms and success stories.
9. OrangeMantra — Revolutionizing Inventory Management with AI
<https://www.orangemantra.com/case-studies/revolutionizing-inventory-management-with-ai-orangemantras-automated-solution/>
Case study showcasing real-world application and benefits of AI in inventory.
10. ScienceDirect — AI-driven Supply Chain Solution
<https://www.sciencedirect.com/science/article/pii/S2949697725000086>
Research on AI techniques for intelligent replenishment and waste reduction.