

How to Create a New Page with Cards - Team Guide

Quick Overview

You'll need to:

1. Create an HTML file for your page layout
2. Create a JavaScript file for functionality
3. Add navigation in the dashboard

Step 1: Create Your HTML Page

Create a new file: pages/your-page-name.html

```
<!-- pages/your-page-name.html -->
<link rel="stylesheet" href="css/simStyle.css">

<div class="page-container">
  <!-- Page Header -->
  <div class="page-header">
    <h1 class="page-title">Your Page Title</h1>
    <p class="page-subtitle">Brief description of what this page does</p>
  </div>

  <!-- Cards Grid -->
  <div class="sim-cards-grid">

    <!-- Card 1 -->
    <div class="sim-card" id="card1">
      <div class="sim-icon">
        <!-- Icon goes here - get from heroicons.com -->
        <svg width="24" height="24" viewBox="0 0 24 24" fill="none"
xmlns="http://www.w3.org/2000/svg">
          <path d="M9 5H7C5.89543 5 5.89543 5 7V19C5 20.1046 5.89543 21 7
21H17C18.1046 21 19 20.1046 19 19V7C19 5.89543 18.1046 5 17 5H15M9 5C9 6.10457 9.89543 7
11 7H13C14.1046 7 15 6.10457 15 5M9 5C9 3.89543 9.89543 3 11 3H13C14.1046 3 15 3.89543
15 5" stroke="#323544" stroke-width="2"/>
        </svg>
      </div>
      <div class="sim-content">
        <div class="sim-body">
          <div class="sim-info">
            <h3 class="sim-name">Card Title</h3>
            <p class="sim-description">Brief description of what this card does</p>
          </div>
          <button class="sim-action-btn" id="openCard1Btn" aria-label="Open Card 1">
            <!-- Keep this arrow icon as is -->
            <svg width="24" height="24" viewBox="0 0 24 24" fill="none"
xmlns="http://www.w3.org/2000/svg">
              <path d="M19.4357 13.9174C20.8659 13.0392 20.8659 10.9608 19.4357
10.0826L9.55234 4.01389C8.05317 3.09335 6.125 4.17205 6.125 5.93128L6.125 18.0688C6.125
19.828 8.05317 20.9067 9.55234 19.9861L19.4357 13.9174Z" fill="white"/>
            </svg>
          </button>
        </div>
      </div>
    </div>
  </div>
```

```

    </div>

    <!-- Add more cards by copying the above structure -->

</div>

<!-- Modal for popups -->
<div id="yourPageModal" class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    <h2 id="modalTitle">Modal Title</h2>
    <iframe id="modalIframe" src="" style="width: 100%; height: 500px; border:
none;"></iframe>
  </div>
</div>
</div>

```

Step 2: Create Your JavaScript File

Create a new file: `pages/your-page-name.js`

```

javascript
// pages/your-page-name.js
let auth0Client = null;
let currentBusiness = null;

// Get Auth0 client
async function getAuth0Client() {
  if (window.auth0Client) {
    return window.auth0Client;
  }

  try {
    const response = await fetch("/auth_config.json");
    const config = await response.json();

    auth0Client = await auth0.createAuth0Client({
      domain: config.domain,
      clientId: config.clientId,
      cacheLocation: 'localStorage',
      useRefreshTokens: true
    });

    window.auth0Client = auth0Client;
    return auth0Client;
  } catch (error) {
    console.error("Error configuring Auth0:", error);
    return null;
  }
}

// Main function - runs when page loads
export async function initYourPageName() {
  const auth0Client = await getAuth0Client();

  if (!auth0Client) {
    console.error("Failed to get Auth0 client");
  }
}

```

```

    return;
}

// Get modal elements
const modal = document.getElementById("yourPageModal");
const modalTitle = document.getElementById("modalTitle");
const iframe = document.getElementById("modalIframe");
const closeBtn = document.getElementsByClassName("close")[0];

try {
    // Check if user is logged in
    const isAuthenticated = await auth0Client.isAuthenticated();

    if (!isAuthenticated) {
        console.error("User not authenticated");
        window.location.href = '/';
        return;
    }

    // Get user info
    const user = await auth0Client.getUser();
    const userEmail = user.email || 'unknown';
    const userName = user.name || 'User';

    // Get the selected business (NEW!)
    currentBusiness = window.dataManager.getSelectedBusinessOrFirst();
    const businessName = currentBusiness?.store_info?.name || 'No Business';

    console.log("Page loaded for:", businessName);

    // Setup your buttons
    const buttons = [
        {
            btn: document.getElementById("openCard1Btn"),
            title: "Tool Name 1",
            url: `https://your-tool-
url.com?email=${encodeURIComponent(userEmail)}&business=${encodeURIComponent(businessName)}`,
        },
        {
            btn: document.getElementById("openCard2Btn"),
            title: "Tool Name 2",
            url: `https://your-tool-url-2.com?email=${encodeURIComponent(userEmail)}`,
        }
    ];

    // Add more buttons as needed
};

// Add click handlers
buttons.forEach(({btn, title, url}) => {
    if (btn) {
        btn.onclick = function(e) {
            e.stopPropagation();
            modalTitle.textContent = title;
            iframe.src = url;
            modal.style.display = "block";
            document.body.style.overflow = 'hidden';
        }
    }
});

```

```

// Setup close button
if (closeBtn) {
  closeBtn.onclick = function() {
    modal.style.display = "none";
    iframe.src = "";
    document.body.style.overflow = 'auto';
  }
}

// Close modal when clicking outside
window.addEventListener('click', function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
    iframe.src = "";
    document.body.style.overflow = 'auto';
  }
});

} catch (error) {
  console.error("Error in initYourPageName:", error);
}
}
}

```

Step 3: Add to Dashboard

3.1 Add Navigation (in dashboard.html)

Find the nav menu and add your page:

```

<li class="nav-item">
  <a href="#" class="nav-link" data-page="your-page-name">
    <svg width="24" height="24" viewBox="0 0 24 24" fill="none"
xmlns="http://www.w3.org/2000/svg">
      <!-- Your icon here -->
      <path d="M3 7H21L19 2H5L3 7Z" stroke="#323544" stroke-width="2"/>
      <path d="M3 12H21" stroke="#323544" stroke-width="2"/>
      <path d="M8 12V20" stroke="#323544" stroke-width="2"/>
      <path d="M16 12V20" stroke="#323544" stroke-width="2"/>
    </svg>
    <span class="nav-text">Your Page Name</span>
  </a>
</li>

```

3.2 Update dashboard.js

Add these three sections to dashboard.js:

1. In the loadPage function, find the switch statement and add:

```

case "your-page-name":
  content = await loadYourPageName();
  break;

```

2. Add this function near the other load functions:

```
async function loadYourPageName() {  
  if (!loadYourPageName.cache) {  
    loadYourPageName.cache = await fetch('pages/your-page-name.html').then(r =>  
r.text());  
  }  
  return loadYourPageName.cache;  
}
```

3. In *initializePageFunctionality*, add:

```
case "your-page-name":  
  if (!window.__yourPageNameLoaded) {  
    try {  
      const mod = await import("../pages/your-page-name.js");  
      mod.initYourPageName();  
      window.__yourPageNameLoaded = true;  
    } catch (error) {  
      console.error("Error loading your page:", error);  
    }  
  } else {  
    import("../pages/your-page-name.js").then(mod => mod.initYourPageName());  
  }  
  break;
```

Complete Example: Creating a "Reports" Page

1. Create pages/reports.html:

```
<link rel="stylesheet" href="css/simStyle.css">

<div class="page-container">
  <div class="page-header">
    <h1 class="page-title">Business Reports</h1>
    <p class="page-subtitle">Generate and view business reports</p>
  </div>

  <div class="sim-cards-grid">
    <!-- Sales Report Card -->
    <div class="sim-card" id="salesReportCard">
      <div class="sim-icon">
        <svg width="24" height="24" viewBox="0 0 24 24" fill="none"
xmlns="http://www.w3.org/2000/svg">
          <path d="M9 19V13H15V19M3 9L12 2L21 9V20C21 20.5304 20.7893 21.0391 20.4142
21.4142C20.0391 21.7893 19.5304 22 19 22H5C4.46957 22 3.96086 21.7893 3.58579
21.4142C3.21071 21.0391 3 20.5304 3 20V9Z" stroke="#323544" stroke-width="2"/>
        </svg>
      </div>
      <div class="sim-content">
        <div class="sim-body">
          <div class="sim-info">
            <h3 class="sim-name">Sales Report</h3>
            <p class="sim-description">View detailed sales analytics and trends</p>
          </div>
          <button class="sim-action-btn" id="openSalesReportBtn" aria-label="Open Sales Report">
            <svg width="24" height="24" viewBox="0 0 24 24" fill="none"
xmlns="http://www.w3.org/2000/svg">
              <path d="M19.4357 13.9174C20.8659 13.0392 20.8659 10.9608 19.4357
10.0826L9.55234 4.01389C8.05317 3.09335 6.125 4.17205 6.125 5.93128L6.125 18.0688C6.125
19.828 8.05317 20.9067 9.55234 19.9861L19.4357 13.9174Z" fill="white"/>
            </svg>
          </button>
        </div>
      </div>
    </div>

    <!-- Inventory Report Card -->
    <div class="sim-card" id="inventoryReportCard">
      <div class="sim-icon">
        <svg width="24" height="24" viewBox="0 0 24 24" fill="none"
xmlns="http://www.w3.org/2000/svg">
          <path d="M12 2L2 7V12C2 16.5 4.23 20.68 7.62 23.15L12 21L16.38 23.15C19.77
20.68 22 16.5 22 12V7L12 2Z" stroke="#323544" stroke-width="2"/>
        </svg>
      </div>
      <div class="sim-content">
        <div class="sim-body">
          <div class="sim-info">
            <h3 class="sim-name">Inventory Report</h3>
            <p class="sim-description">Track stock levels and product movement</p>
          </div>
          <button class="sim-action-btn" id="openInventoryReportBtn" aria-label="Open
Inventory Report">
```

```

        <svg width="24" height="24" viewBox="0 0 24 24" fill="none"
xmlns="http://www.w3.org/2000/svg">
        <path d="M19.4357 13.9174C20.8659 13.0392 20.8659 10.9608 19.4357
10.0826L9.55234 4.01389C8.05317 3.09335 6.125 4.17205 6.125 5.93128L6.125 18.0688C6.125
19.828 8.05317 20.9067 9.55234 19.9861L19.4357 13.9174Z" fill="white"/>
        </svg>
        </button>
        </div>
        </div>
        </div>
        </div>
        </div>

        <!-- Modal -->
        <div id="reportsModal" class="modal">
        <div class="modal-content">
        <span class="close">&times;</span>
        <h2 id="modalTitle">Report Viewer</h2>
        <iframe id="modalIframe" src="" style="width: 100%; height: 500px; border:
none;"></iframe>
        </div>
        </div>
        </div>

```

2. Create pages/reports.js:

```

// pages/reports.js
let auth0Client = null;
let currentBusiness = null;

async function getAuth0Client() {
  if (window.auth0Client) {
    return window.auth0Client;
  }

  try {
    const response = await fetch("/auth_config.json");
    const config = await response.json();

    auth0Client = await auth0.createAuth0Client({
      domain: config.domain,
      clientId: config.clientId,
      cacheLocation: 'localStorage',
      useRefreshTokens: true
    });

    window.auth0Client = auth0Client;
    return auth0Client;
  } catch (error) {
    console.error("Error configuring Auth0:", error);
    return null;
  }
}

export async function initReportsPage() {
  const auth0Client = await getAuth0Client();

  if (!auth0Client) {
    console.error("Failed to get Auth0 client");
  }
}

```

```

    return;
}

const modal = document.getElementById("reportsModal");
const modalTitle = document.getElementById("modalTitle");
const iframe = document.getElementById("modalIframe");
const closeBtn = document.getElementsByClassName("close")[0];

try {
    const isAuthenticated = await auth0Client.isAuthenticated();

    if (!isAuthenticated) {
        console.error("User not authenticated");
        window.location.href = '/';
        return;
    }

    const user = await auth0Client.getUser();
    const userEmail = user.email || 'unknown';
    const userName = user.name || 'User';

    // Get selected business
    currentBusiness = window.dataManager.getSelectedBusinessOrFirst();

    if (!currentBusiness) {
        document.querySelector('.page-container').innerHTML = `
            <div class="empty-state" style="text-align: center; padding: 60px 20px;">
                <h2>No Business Selected</h2>
                <p>Please select a business from the header to view reports.</p>
            </div>
        `;
        return;
    }

    const businessName = currentBusiness.store_info?.name || 'Business';
    const businessId = currentBusiness._id;

    // Setup buttons
    const buttons = [
        {
            btn: document.getElementById("openSalesReportBtn"),
            title: "Sales Report - " + businessName,
            url: `https://your-reports-
url.com/sales?businessId=${businessId}&businessName=${encodeURIComponent(businessName)}&
email=${encodeURIComponent(userEmail)}~`
        },
        {
            btn: document.getElementById("openInventoryReportBtn"),
            title: "Inventory Report - " + businessName,
            url: `https://your-reports-
url.com/inventory?businessId=${businessId}&businessName=${encodeURIComponent(businessName)}~`
        }
    ];

    // Add click handlers
    buttons.forEach(({btn, title, url}) => {
        if (btn) {
            btn.onclick = function(e) {
                e.stopPropagation();
            }
        }
    });
}

```



```

        modalTitle.textContent = title;
        iframe.src = url;
        modal.style.display = "block";
        document.body.style.overflow = 'hidden';
    }
}
});

// Close handlers
if (closeBtn) {
    closeBtn.onclick = function() {
        modal.style.display = "none";
        iframe.src = "";
        document.body.style.overflow = 'auto';
    }
}

window.addEventListener('click', function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
        iframe.src = "";
        document.body.style.overflow = 'auto';
    }
});

} catch (error) {
    console.error("Error in initReportsPage:", error);
}
}
}

```

3. Update dashboard.js:

```

// Add in loadPage switch statement:
case "reports":
    content = await loadReportsPage();
    break;

// Add function:
async function loadReportsPage() {
    if (!loadReportsPage.cache) {
        loadReportsPage.cache = await fetch('pages/reports.html').then(r => r.text());
    }
    return loadReportsPage.cache;
}

// Add in initializePageFunctionality:
case "reports":
    if (!window.__reportsLoaded) {
        try {
            const mod = await import("../pages/reports.js");
            mod.initReportsPage();
            window.__reportsLoaded = true;
        } catch (error) {
            console.error("Error loading reports page:", error);
        }
    } else {
        import("../pages/reports.js").then(mod => mod.initReportsPage());
    }
}

```

```
break;
```

Key Points for Team:

1. File Naming

- HTML file: pages/your-page-name.html
- JS file: pages/your-page-name.js
- Keep names lowercase with hyphens

2. Required Changes

- Replace your-page-name everywhere with your actual page name
- Replace YourPageName in JavaScript with camelCase version
- Update all IDs to be unique

3. Business Data Access

javascript

// Always get the selected business:

```
currentBusiness = window.dataManager.getSelectedBusinessOrFirst();
```

// Use business data:

```
const businessName = currentBusiness?.store_info?.name || 'Business';
```

```
const businessId = currentBusiness?._id;
```

4. Common Mistakes to Avoid

- Don't forget to export your init function: export async function initYourPageName()
- Make sure button IDs in HTML match JavaScript
- Always check if user is authenticated
- Handle the case when no business is selected

5. Testing Checklist

- Page loads without errors
- Clicking nav item shows your page
- Cards display correctly
- Buttons open modals
- Modal closes properly
- Business data is available

6. Getting Icons

Visit heroicons.com and:

1. Search for an icon
2. Click "Copy SVG"
3. Paste in your HTML
4. Add stroke="#323544" stroke-width="2" to paths

Quick Reference

To add a new page:

1. Copy an existing page's HTML/JS files
2. Rename them
3. Update all references to the page name

4. Add to navigation in dashboard.html
5. Add three snippets to dashboard.js
6. Test!

This approach ensures your new page works with the global business selector and follows the established patterns in your codebase.