

Appendix E: Full Source Code for Website

Admin-dashboard.css

```
body {
  margin: 0;
  padding: 0;
  background: var(--bg-main);
}

.content {
  padding-top: 80px;
  padding-left: 260px;
  display: flex;
  align-items: center;
  flex-direction: column;
}

h2 {
  font-size: 2rem;
  margin-bottom: 20px;
  text-align: center;
}

.inventory-container{
  border: 2px solid #000;
  width: 90%;
  box-shadow: 0 2px 8px rgba(0,0,0,0.15);
  background: #fff;
  border-radius: 6px;
  overflow: visible;
}

.inventory-list {
  display: flex;
  flex-direction: column;
  width: 100%;
  min-width: 600px;
}
```

```
.item-wrapper {
  position: relative;
  background-color: var(--bg-main);
}

.item-status {
  flex: 0.7 !important;
  min-width: 115px;
}

.item-location {
  flex: 0.5 !important;
  min-width: 100px;
}

@media (pointer:none), (pointer:coarse) {
  .delete-btn{
    opacity: 1 !important;
  }
}

.delete-btn {
  transition: background ease 0.2s, opacity ease 0.2s;
  position: absolute;
  left: -40px; /* adjust to change offset */
  top: 50%;
  z-index: 1000;
  transform: translateY(-50%);
  display: block;
  background: #fff;
  border: 1px solid #900;
  color: #900;
  border-radius: 4px;
  padding: 4px;
  cursor: pointer;
  opacity: 0;
}
```

```
.delete-btn:hover {
    background: #fdd;
}

/* Show delete buttons in edit mode */

/* Item Row */
.item {
    display: flex;
    border-bottom: 1px solid #000;
    align-items: stretch;
    min-height: 20px;
}

.item-wrapper:hover .delete-btn {
    opacity: 1;
}

/* Header Row */
.header {
    font-weight: bold;
    background: var(--bg-hover);
    border-bottom: 2px solid #000;
    color: #000;
    border-radius: 5px 5px 0 0;
}

.item-header > div > div , .item-wrapper > div > div{
    display: flex;
    align-items: center;
}

/* Each Column */
.item > div {
    flex: 1;
    padding: 10px;
    border-right: 1px solid #000;
}
```

```
    display: flex;
    align-items: center;
}

.item > div > p{
    margin:0;
}

.item > div:not(.material-symbols-outlined), .item > span:not(.tag){
    font-size: 15px;
}

.item-name {
    display: flex;
    justify-content: space-between;
    align-items: center;
    min-width: 125px;
    overflow-wrap: break-word;
}

.item-tags{
    min-width: none;
}

.sort-btn {
    padding: 0;
}

.sort-icon {
    font-size: 20px !important;
}

.add-button {
    flex: 1;
    display: flex;
    align-items: center;
    justify-content: center;
```

```

    gap: 6px;
    padding: 10px;
    cursor: pointer;
    position: relative;
    color: #3498DC;
    transition: background 0.2s ease;
    border-radius: 0 0 5px 5px;
}

.add-button:hover{
    background: var(--bg-hover);
}

.add-button .material-symbols-outlined {
    cursor: pointer;
    font-size: 24px;
}

.item > .view, .item > .add {
    flex: 0.3;
    text-align: center;
    position: relative; /* allow pseudo-element positioning */
    padding: 0 10px;
    display: flex;
    justify-content: center;
}

/* Hover circle effect */
.item > .view.material-symbols-outlined::after, .item >
.add.material-symbols-outlined::after {
    content: "";
    position: absolute;
    top: 50%;
    left: 50%;
    width: 36px;
    height: 36px;
    transform: translate(-50%, -50%);
    background: rgba(0, 0, 0, 0.15); /* half opacity */
}

```

```

border-radius: 50%;
opacity: 0; /* invisible by default */
transition: opacity 0.25s ease;
pointer-events: none; /* so the hover still works normally */
}

/* Show a small text to explain */
.item > .view.material-symbols-outlined::before, .item >
.add.material-symbols-outlined::before{
    content: "Open item page";
    font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI",
sans-serif !important;
    position: absolute;
    bottom: -20px;
    transform: translateY(+50%);
    left: 50%;
    transform: translateX(-50%);
    background-color: #333;
    color: #fff;
    padding: 6px 10px;
    border-radius: 4px;
    font-size: 12px;
    white-space: nowrap;
    opacity: 0;
    pointer-events: none;
    transition: opacity 0.25s ease;
    z-index: 10;
}

.item > .add.material-symbols-outlined::before{
    content: "Add item";
}

.item > .add{
    margin: auto;
}

.empty-search-warning {
    text-align: center;

```

```

}

/* Show the circle and text on hover */
.item > .view.material-symbols-outlined:hover::after,
.item > .view.material-symbols-outlined:hover::before,
.item > .add.material-symbols-outlined:hover::after,
.item > .add.material-symbols-outlined:hover::before{
    opacity: 1;
}

/* Remove borders from last column and rows */
.item > div:last-child {
    border-right: none;
}

.inventory-list > div:last-child > .item, .inventory-list > div:last-child{
    border-bottom: none;
    border-radius: 0 0 6px 6px;
}

/* Pagination Control Styles */
#pagination-controls{
    display: flex;
    justify-content: center;
    align-items: center;
}

#pagination-controls a{
    text-decoration: none;
    color: black;
}

/* Tags */
.item-tags {
    display: flex;
    flex-wrap: wrap;
    gap: 4px;

```

```
    align-items: center;
    flex: 1.5 !important;
}

/* Form */

.new-item > .item-tags{
    flex-direction: column;
    gap: 6px;
    padding: 10px;
}

.new-item > div{
    height: 223px;
    display: flex;
    justify-content: center;
    align-items: center;
}

.tag-list {
    display: flex;
    flex-wrap: wrap;
    gap: 4px;
    min-height: 19px;
}

.item input[type="text"],
.item select {
    appearance: none;
    width: 100%;
    padding: 6px 10px;
    border: 1px solid #ccc;
    border-radius: 6px;
    background: #fafafa;
    font-size: 0.95rem;
    transition: border 0.25s, background 0.25s;
    box-sizing: border-box;
}
```

```
.item select{
    background: url("https://cdn-icons-png.flaticon.com/256/5800/5800621.png")
white no-repeat 96% !important;
    background-size: 10px 8px !important;
}

.item input[type="text"]:focus,
.item select:focus {
    border-color: #4a90e2;
    background: #fff;
    outline: none;
}

/* Tag input field */
.tag-input {
    min-width: 80px;
    flex: 1;
    padding: 7px 10px;
    border-radius: 6px;
    font-size: 0.8em !important;
}

.custom-tag-row {
    display: flex;
    align-items: center;
    gap: 6px;
}

/* Add Custom Tag button */
.add-tag-btn {
    background: #4a90e2;
    border: none;
    color: white;
    padding: 4px 10px;
    margin-left: 6px;
    border-radius: 6px;
    cursor: pointer;
    font-weight: bold;
}
```

```
    transition: background 0.2s ease;
}

.add-tag-btn:hover {
    background: #3774b8;
}

/* Circle button next to tag input */
.tag-color-picker {
    width: 18px;
    height: 18px;
    border-radius: 50%;
    border: 1px solid #444;
    cursor: pointer;
    margin-left: 6px;
    position: relative;
    flex-shrink: 0;
}

.tag-color-picker.tag {
    width: 18px;
    height: 18px;
    border-radius: 50%;
    border: 1px solid #444;
    cursor: pointer;
    padding: 0;
}

/* Color palette popup */
.tag-color-popup {
    position: absolute;
    top: 24px;
    left: 0;
    background: #fff;
    border: 1px solid #444;
    border-radius: 6px;
    padding: 6px;
    display: flex;
    gap: 6px;
```

```
    z-index: 10;
    box-shadow: 0 2px 6px rgba(0,0,0,0.25);
}

/* Custom popup when hovered */

.tag-preview{
    position: relative;
    cursor: pointer;
}

.tag-preview::after{
    content: "Click to remove";
    font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI",
sans-serif !important;
    position: absolute;
    bottom: -30px;
    left: 50%;
    transform: translateX(-50%);
    background-color: #333;
    color: #fff;
    padding: 6px 10px;
    border-radius: 4px;
    font-size: 12px;
    white-space: nowrap;
    opacity: 0;
    pointer-events: none;
    transition: opacity 0.25s ease;
    z-index: 10;
}

.tag-preview:hover::after{
    opacity: 1;
}

/* Individual color choices */
.tag-color-option {
    width: 16px;
```

```
    height: 16px;
    border-radius: 50%;
    cursor: pointer;
    border: 1px solid rgba(0,0,0,0.4);
}

.small-label{
    border-top: 1px solid #aaa;
    margin-top: 8px;
    padding-top: 4px;
    font-size: 0.8rem;
    font-weight: 600;
    color: #444;
    width:100%;
}

.quick-tag-buttons {
    display: flex;
    gap: 6px;
    margin-top: 2px;
}

.quick-tag-btn {
    padding: 2px 6px;
    border-radius: 6px;
    font-size: 0.85rem;
    cursor: pointer;
    border: none;
    font-weight: 600;
    transition: transform 0.15s ease;
}

.quick-tag-btn:hover {
    transform: scale(1.06);
}

.quick-borrowed {
    background: #B9D8B4; /* same green tag style */
    color: #1F5A1A;
```

```
}

.quick-dispose {
  background: #F2B8B5; /* same red tag style */
  color: #7A1F20;
}

.tag {
  border-radius: 4px;
  padding: 2px 6px;
  font-size: 0.8rem;
}

.tag.tag-grey {
  background: #E5E5E5;
  color: #333;
}

.tag.tag-red {
  background: #F2B8B5;
  color: #7A1F20;
}

.tag.tag-orange {
  background: #F4C29A;
  color: #7A3D00;
}

.tag.tag-yellow {
  background: #F7E8A4;
  color: #6A5A00;
}

.tag.tag-green {
  background: #B9D8B4;
  color: #1F5A1A;
}
```

```
.tag.tag-blue {
  background: #B8D1EA;
  color: #1C3F6E;
}

.tag.tag-purple {
  background: #D3C2E8;
  color: #4A3066;
}

.material-symbols-outlined {
  cursor: pointer;
  vertical-align: middle;
  font-size: 24px;
  font-variation-settings:
    'FILL' 0,
    'wght' 400,
    'GRAD' 0,
    'opsz' 24
}

.item.new-item input[type="text"]:not(.tag-input),
.item.new-item select {
  margin: auto 0;
  width: 100%;
  box-sizing: border-box;
}

#loadingOverlay {
  position: fixed;
  inset: 0;
  background: rgba(0, 0, 0, 0.4);
  backdrop-filter: blur(100px);
  display: flex;
  justify-content: center;
  align-items: center;
  opacity: 1;
}
```

```
visibility: visible;
transition: opacity 0.4s ease, visibility 0.4s ease;
z-index: 99999;
}

#loadingOverlay.hidden {
  opacity: 0;
  visibility: hidden;
}

/* Spinner */
.spinner {
  width: 60px;
  height: 60px;
  border: 6px solid #ffffff50;
  border-top-color: white;
  border-radius: 50%;
  animation: spin 0.9s linear infinite;
}

@keyframes spin {
  to {
    transform: rotate(360deg);
  }
}

.delete-confirmation {
  position: fixed;
  inset: 0;
  background: rgba(0, 0, 0, 0.4);
  display: flex;
  justify-content: center;
  align-items: center;
  visibility: visible;
  opacity: 1;
  transition: opacity 0.2s ease, visibility 0.2s ease;
  z-index: 99998;
}
```

```
.delete-confirmation-box{
  background: white;
  width: 400px;
  height: 175px;
  border-radius: 10px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.15);
  border: 1px solid rgb(224, 224, 224);
}

.delete-confirmation-box > h3{
  font-size: 24px;
  margin: 20px 20px 0px 20px;
}

.delete-confirmation-box > p {
  margin: 10px 20px 10px 20px;
}

.confirmation-container{
  margin-top: 40px;
  display: flex;
  justify-content: center;
  gap: 10px;
}

.cancel, .confirm{
  width: 175px;
  height: 40px;
  font-size: 18px;
  display: flex;
  justify-content: center;
  align-items: center;
  border-radius: 6px;
  cursor: pointer;
  transition: background-color 0.2s ease;
}
```

```
.cancel {
  border: 1px solid rgb(224, 224, 224);
}

.cancel:hover{
  background-color: rgb(224, 224, 224);
}

.confirm {
  background-color: #f42c2c;
  color:white;
  font-weight: 400;
}

.confirm:hover{
  background-color: #d42727;
}

.delete-confirmation.hidden {
  opacity: 0;
  visibility: hidden;
}

.top-bar {
  position:fixed;
  display: flex;
  width: 100vw;
  height: 60px;
  background: var(--bg-surface);
  border-bottom: 1px solid var(--border-default);
  justify-content: left;
  align-items: center;
  margin-bottom: 20px;
  z-index: 100;
  padding-left:22px;
}

.top-bar .title {
```

```
margin: 0;
font-size: 24px;
margin-left: 20px;
}

.top-center {
  position: fixed;
  left: 40%;
}

.search-input {
  padding: 10px 14px 10px 40px;
  font-size: 14px;
  border: 1px solid #ccc;
  border-radius: 8px;
  transition: 0.2s;
  width: 400px;
}

.search-input:focus {
  outline: none;
  border-color: #888;
  background: #fff;
}

.search-symbol{
  color: #757575;
  position: relative;
  top: 50%;
  left: 40px;
}

@media (max-width: 1000px){
  .search-input{
    width: 300px;
  }
  .top-center{
    left: 50%;
  }
}
```

```
}  
  
}  
  
@media (max-width: 950px){  
  .content{  
    padding-left: 70px;  
  }  
  
  .top-center{  
    left: 40%;  
  }  
  
  .top-bar .title {  
    font-size: 20px;  
  }  
  .item {  
    min-width: 600px;  
  }  
}  
  
@media (max-width: 750px) {  
  .content{  
    padding-left:10px;  
  }  
  
  .inventory-container{  
    overflow-y: hidden;  
    overflow-x: auto;  
  }  
  
  .inventory-list{  
    position:relative;  
    padding-left:45px;  
  }  
  .item{  
    border-left: 1px solid #000;  
  }  
}
```

```

    }

    .item > .view.material-symbols-outlined::before, .item >
.add.material-symbols-outlined::before{
        content: "Open";
    }

}

@media (max-width: 660px) {

    .search-input{
        width:300px;
    }

    .search-symbol{
        transform:translateY(20%);
        left:35px;
    }

    .top-center{
        left:10px;
        width:100%;
        display:flex;
        justify-content: center;
    }

    .sidebar-button{
        z-index: 100;
    }

    .title{
        display:none;
    }

}

@media (max-width: 450px) {

    .search-symbol{
        display:none;
    }

    .search-input{

```

```
padding-left: 5px;
width:200px;
}
}
```

Catalogue.css

```
.inventory-list{
padding:0 !important;
}

.item{
border-left:0;
}
```

Catalogue.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="shortcut icon" href="images/cabinet.png" type="image/x-icon">
<link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,w
ght,FILL,GRAD@20..48,100..700,0..1,-50..200&icon_names=add,arrow_downward,arrow
_upward,chevron_left,chevron_right,delete,edit,home,inventory_2,library_books,l
ogin,logout,map,menu,open_in_new,remove,save,search,sort" />
<link rel="stylesheet" href="theme.css">
<link rel="stylesheet" href="admin-dashboard.css">
<link rel="stylesheet" href="catalogue.css">
<link rel="stylesheet" href="sidebar.css">
<title>St. Mary's International School Shed</title>
</head>
<body class="Catalogue">
<!-- Loading Screen -->
<div id="loadingOverlay" class="hidden">
<div class="spinner"></div>
</div>

<!-- Top Bar -->
```

```

<div class="top-bar">
  <span class="sidebar-button material-symbols-outlined">menu</span>
  <div class="sidebar-header">
    <h1>Shed Storage</h1>
  </div>
  <div class="bar-left">
    <h2 class="title">Catalogue</h2>
  </div>
  <div class="top-center">
    <span class="material-symbols-outlined search-symbol">search</span>
    <input type="text" id="search-bar" class="search-input"
placeholder="Search items, locations, and tags...">
  </div>
</div>
<!-- Sidebar Navigation -->
<div class="sidebar">
  <div class="sidebar-nav">
    <a class="nav-item" href="/index.html">
      <span class="material-symbols-outlined nav-icon">home</span>
      <span class="nav-text">Home</span>
    </a>
    <a class="nav-item active" href="/catalogue.html">
      <span class="material-symbols-outlined
nav-icon">inventory_2</span>
      <span class="nav-text">Catalogue</span>
    </a>
    <a class="nav-item" href="/admin-dashboard.html">
      <span class="material-symbols-outlined nav-icon">edit</span>
      <span class="nav-text">Edit</span>
    </a>
    <a class="nav-item" href="/map-view.html">
      <span class="material-symbols-outlined nav-icon">map</span>
      <span class="nav-text">Map Viewer</span>
    </a>
    <a class="nav-item" href="/logs.html">
      <span class="material-symbols-outlined
nav-icon">library_books</span>
      <span class="nav-text">Logs</span>

```

```

        </a>
    </div>
    <div class="sidebar-footer">
        <a class="login-btn" href="/login.html">
            <span class="material-symbols-outlined logoff-icon">login</span>
            <span class="nav-text">Login</span>
        </a>
    </div>
</div>
<!-- Website Content -->
<div class="content">

    <!-- Inventory Table -->
    <div class="inventory-container">
        <div class="inventory-list">
            <div class="item header">
                <div class="item-name">Item Name
                    <button id="sort-name-btn" class="sort-btn" title="Sort
alphabetically">
                        <span class="material-symbols-outlined
sort-icon">sort</span>
                    </button>
                </div>
                <div class="item-location">Location (ID)</div>
                <div class="item-tags">Tags</div>
                <div class="item-status">Status</div>
                <div class="view"> </div>
            </div>
        </div>
    </div>
</div>
<script defer src="bundle.js"></script>
</body>
</html>

```

Index.css

```

@property --myColor1 {
    syntax: '<color>';

```

```
    initial-value: #2a7b9b;
    inherits: false;
}

@property --myColor2 {
    syntax: '<color>';
    initial-value: #2a7b9b;
    inherits: false;
}

@property --myColor3 {
    syntax: '<color>';
    initial-value: #2a7b9b;
    inherits: false;
}

.content{
    padding: 120px 0px 0 0 !important;
    display: flex;
    flex-wrap:wrap;
    justify-content: center;
}

h3{
    font-size: 78px;
    margin:10px
}

h3.special{
    width: 600px;
    background: var(--brand-gradient);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
}

.lead{
    margin:20px 20px;
    color: var(--text-muted);
```

```
    font-size: 24px;
}

.title{
    width: 1000px;
}

/* Stats Display */
.stats-container {
    display: flex;
    gap: 20px;
    margin-top: 80px;
    flex-wrap: wrap;
}

.stat-card {
    background-color: var(--bg-surface);
    color: var(--text-primary);
    border-radius: 12px;
    border: var(--border-default) 1px solid;
    padding: 20px 25px;
    display: flex;
    align-items: center;
    min-width: 220px;
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.15);
}

.stat-icon {
    font-size: 40px;
    margin-right: 15px;
    color: var(--brand-primary);
}

.stat-card.checked-out .stat-icon {
    color: var(--brand-secondary);
}

.stat-number {
```

```
    font-size: 2rem;
    font-weight: bold;
    margin: 0;
}

.stat-label {
    font-size: 0.9rem;
    opacity: 0.8;
    margin: 0;
}

/* Landing Buttons */
.landing-buttons{
    display: flex;
    margin-top: 50px;
    margin-left: 35px;
    width: 1000px;
    gap: 20px;
}

.landing-button{
    align-items: center;
    padding: 12px 16px;
    cursor: pointer;
    border-radius: 6px;
    text-decoration: none;
    color: white;
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.15);
    transition:
        background-color 0.2s ease,
        border-color 0.2s ease,
        --myColor1 0.2s ease,
        --myColor2 0.2s ease,
        --myColor3 0.2s ease;
}

.catalogue-button{
```

```
    background: linear-gradient(90deg, var(--myColor1), var(--myColor2),
var(--myColor3));
    border: var(--accent-hover) 2px solid;
}

.catalogue-button:hover{
    --myColor1: var(--brand-gradient-1);
    --myColor2: var(--brand-gradient-2);
    --myColor3: var(--brand-gradient-3);
    background-color: var(--brand-primary);
    /* border-color: var(--brand-secondary); */
}

.map-button{
    color: black;
    background-color: var(--bg-surface);
    border: var(--brand-primary) 2px solid;
}

.map-button:hover{
    background-color: var(--accent-hover);
    color: white;
}

@media (max-width: 700px){
    h3 {
        font-size: 50px;
        width: 80vw;
    }

    .lead{
        font-size: 20px;
    }

    .special {
        width: 80vw !important;
    }

    .title {
```

```

        width: 80vw;
    }

    .landing-buttons{
        margin-left: 0;
        justify-content: center;
    }

    .stats-container{
        justify-content: center;
    }
}

```

Index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,w
ght,FILL,GRAD@20..48,100..700,0..1,-50..200&icon_names=add,delete,edit,home,inv
entory_2,login,logout,map,open_in_new,save,search"
    />
    <link rel="stylesheet" href="theme.css">
    <!-- <link rel="stylesheet" href="admin-dashboard.css"> -->
    <link rel="stylesheet" href="index.css">
    <link rel="stylesheet" href="sidebar.css">
    <title>Welcome!</title>
</head>
<body class="Home">
    <div class="content">
        <div class="title">
            <h3>Welcome to</h3>
            <h3 class="special">Shed Storage</h3>
            <p class="lead">All your sets and props, finally in one place. </p>
        </div>

```

```

        <div class="landing-buttons">
            <a href="/catalogue.html" class="catalogue-button
landing-button">See the Catalogue</a>
            <a href="/map-view.html" class="map-button landing-button">View the
Map </a>
        </div>
        <div class="stats-container">
            <div class="stat-card">
                <span class="material-symbols-outlined stat-icon">
                    inventory_2
                </span>
                <div class="stat-info">
                    <p class="stat-number" id="total-props">0</p>
                    <p class="stat-label">Total Props in Shed</p>
                </div>
            </div>

            <div class="stat-card checked-out">
                <span class="material-symbols-outlined stat-icon">
                    open_in_new
                </span>
                <div class="stat-info">
                    <p class="stat-number" id="checked-out-props">0</p>
                    <p class="stat-label">Checked Out</p>
                </div>
            </div>
        </div>
    </div>
</body>
<script defer src="bundle.js"></script>
</html>

```

Item.css

```

body {
    margin: 0;
    background: var(--bg-surface)
}

.container {

```

```
max-width:1500px;
padding: 70px 0 0 250px;
display:flex;
justify-content: center;
}

.card {
padding: 20px;
width:1000px;
background: #fff;
border-radius: 10px;
box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}

.status-text-container{
display:flex;
flex-direction: row;
gap:7px;
}

p{
margin: 0;
}

.check-inout-button{
color: black;
border: 1.5px solid var(--text-primary);
background-color: var(--bg-surface);
border-radius: 5px;
box-shadow: 0 2px 8px rgba(0,0,0,0.1);
margin-left: 20px;
font-size: 14px;
padding: 5px 10px 5px 10px;
cursor:pointer;
}

.check-inout-button:hover{
background-color: var(--bg-hover);
```

```
}

.filled {
  font-variation-settings:
    'FILL' 1,
    'wght' 400,
    'GRAD' 0,
    'opsz' 24
}

.status-box {
  display: flex;
  align-items: center;
  margin-top: 10px;
  gap: 5px;
}

.location {
  margin-top: 25px;
}

.status-symbol{
  font-size: 19px !important;
}

.in-shed{
  color:green;
}

.checked-out{
  color: red;
}

.header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 20px;
}
```

```
.header h1 {
  margin: 0;
  font-size: 28px;
  max-width: 900px;
}

.info-grid {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 20px;
  margin-bottom: 20px;
}

.info-box {
  background: #fafafa;
  padding: 15px;
  border-radius: 8px;
  border: 1px solid #e0e0e0;
}

h3 {
  margin-top: 0;
  font-size: 18px;
}

.map-box {
  width: 100%;
  height: 420px;
  background: var(--bg-surface);
  border: 2px dashed #bbb;
  border-radius: 10px;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 18px;
  color: #777;
  overflow: hidden;
}

.map-box > iframe{
  transform: scale(0.70);
  position: absolute;
```

```
}

.tags-box {
  background: #fafafa;
  padding: 15px;
  border-radius: 8px;
  border: 1px solid #e0e0e0;
  margin-top: 20px;
}

.tag {
  border-radius: 4px;
  padding: 2px 6px;
  font-size: 0.8rem;
}

.tag.tag-grey {
  background: #E5E5E5;
  color: #333;
}

.tag.tag-red {
  background: #F2B8B5;
  color: #7A1F20;
}

.tag.tag-orange {
  background: #F4C29A;
  color: #7A3D00;
}

.tag.tag-yellow {
  background: #F7E8A4;
  color: #6A5A00;
}

.tag.tag-green {
  background: #B9D8B4;
```

```
    color: #1F5A1A;
}

.tag.tag-blue {
    background: #B8D1EA;
    color: #1C3F6E;
}

.tag.tag-purple {
    background: #D3C2E8;
    color: #4A3066;
}

.tags{
    display: flex;
    gap: 5px;
}

#loadingOverlay {
    position: fixed;
    inset: 0;
    background: rgba(0, 0, 0, 0.4);
    backdrop-filter: blur(100px);
    display: flex;
    justify-content: center;
    align-items: center;
    opacity: 1;
    visibility: visible;
    transition: opacity 0.4s ease, visibility 0.4s ease;
    z-index: 99999;
}

#loadingOverlay.hidden {
    opacity: 0;
    visibility: hidden;
}

/* Spinner */
```

```
.spinner {
  width: 60px;
  height: 60px;
  border: 6px solid #ffffff50;
  border-top-color: white;
  border-radius: 50%;
  animation: spin 0.9s linear infinite;
}

.top-bar {
  position: fixed;
  display: flex;
  width: 100vw;
  height: 60px;
  background: var(--bg-surface);
  border-bottom: 1px solid var(--border-default);
  justify-content: left;
  align-items: center;
  margin-bottom: 20px;
  z-index: 100;
  padding-left: 22px;
}

.top-bar .title {
  margin: 0;
  font-size: 24px;
  margin-left: 20px;
}

@keyframes spin {
  to {
    transform: rotate(360deg);
  }
}

@media (max-width: 1260px) {
  .map {
    transform: scale(0.6) !important;
  }
}
```

```
.map-box{
    height:375px;
}

}

@media (max-width: 1130px){
    .map{
        transform: scale(0.5) !important;
    }
    .map-box{
        height:300px;
    }
}

@media (max-width: 950px){
    .container{
        padding-left: 70px;
    }
    .top-bar .title {
        font-size: 20px;
    }
}

@media (max-width: 750px) {
    .container{
        padding-left:10px;
    }

    .status-box{
        flex-direction: column;
        align-items: flex-start;
        gap:15px;
    }

    .check-inout-button{
        margin-left: 0;
    }
}
```

```
@media (max-width: 700px) {  
    .map{  
        transform: scale(0.65) !important;  
    }  
    .map-box{  
        height:575px;  
    }  
}
```

```
@media (max-width: 600px) {  
    .map{  
        transform: scale(0.59) !important;  
    }  
    .map-box{  
        height:520px;  
    }  
}
```

```
@media (max-width: 550px) {  
    .map{  
        transform: scale(0.45) !important;  
    }  
    .map-box{  
        height:410px;  
    }  
}
```

```
@media (max-width: 450px) {  
    .map{  
        transform: scale(0.40) !important;  
    }  
    .map-box{  
        height:410px;  
    }  
}
```

```
@media (max-width: 450px) {
```

```

.map{
    transform: scale(0.33) !important;
}
.map-box{
    height:290px;
}
}

```

Item.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="images/label.png" type="image/x-icon">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,w
ght,FILL,GRAD@20..48,100..700,0..1,-50..200&icon_names=add,check_circle,delete,
do_not_disturb_on,edit,home,inventory_2,library_books,login,logout,map,menu,ope
n_in_new,save" />
    <link rel="stylesheet" href="theme.css">
    <link rel="stylesheet" href="sidebar.css">
    <link rel="stylesheet" href="item.css">
    <title>Item</title>
</head>
<body class="Item">
    <!-- Top Bar -->
    <div class="top-bar">
        <span class="sidebar-button material-symbols-outlined">menu</span>
        <div class="sidebar-header">
            <h1>Shed Storage</h1>
        </div>
        <div class="bar-left">
            <h2 class="title">Item</h2>
        </div>
    </div>

```

```
<div class="sidebar">
  <div class="sidebar-nav">
    <a class="nav-item" href="/index.html">
      <span class="material-symbols-outlined nav-icon">home</span>
      <span class="nav-text">Home</span>
    </a>
    <a class="nav-item active" href="/catalogue.html">
      <span class="material-symbols-outlined
nav-icon">inventory_2</span>
      <span class="nav-text">Catalogue</span>
    </a>
    <a class="nav-item" href="/admin-dashboard.html">
      <span class="material-symbols-outlined nav-icon">edit</span>
      <span class="nav-text">Edit</span>
    </a>
    <a class="nav-item" href="/map-view.html">
      <span class="material-symbols-outlined nav-icon">map</span>
      <span class="nav-text">Map Viewer</span>
    </a>
    <a class="nav-item" href="/logs.html">
      <span class="material-symbols-outlined
nav-icon">library_books</span>
      <span class="nav-text">Logs</span>
    </a>
  </div>
  <div class="sidebar-footer">
  </div>
</div>
<div id="loadingOverlay" class="hidden">
  <div class="spinner"></div>
</div>
<div class="container">
  <div class="card">
    <div class="header">
      <h1 class="name">Product Name</h1>
      <div class="id">ID: 123456</div>
    </div>
```

```

    <div class="info-grid">
      <div class="info-box">
        <h3>Location</h3>
        <p class="location">Location goes here</p>
      </div>
      <div class="info-box">
        <h3>Status</h3>
        <div class="status-box">
          <div class="status-text-container">
            <span class="status-symbol material-symbols-outlined
filled">check_circle</span>
            <p class="status">Status goes here</p>
          </div>
          <button class="check-inout-button">Check In</button>
        </div>
      </div>
    </div>

    <div class="map-box">
      <iframe src="map.html?location=A1" frameborder="0" width="1260"
height="600" class="map"></iframe>
    </div>

    <div class="tags-box">
      <h3>Tags</h3>
      <div class="tags">None</div>
    </div>
  </div>
</div>
</body>
<script defer src="bundle.js"></script>
</html>

```

Login.css

```

body{
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;

```

```
    align-items: center;
    margin: 0 auto;
    background: #fff;
    padding: 40px;
    flex-direction: column;
    height: 100%;
}

h2 {
    font-size: 2rem;
    margin-bottom: 40px;
    margin-top: 30px;
    text-align: center;
}

form {
    display: flex;
    flex-direction: column;
    width: 450px;
}

form input {
    height: 30px;
    border-radius: 6px;
    border: 1px solid #e0e0e0;
    margin-bottom: 10px;
    padding-left: 5px;
    font-size: 14px;
}

button {
    height: 30px;
    background-color: #E5E5E5;
    border-radius: 6px;
    border: 1px solid #6d6d6d;
    color: black;
}
```

```

label {
    font-size: 17px;
}

@media (max-width: 500px){
    .Login {
        width: 80vw;
    }
    form {
        width: 75vw;
    }

    h2{
        font-size: 28px;
    }

    label{
        font-size: 19px;
    }

    input {
        font-size:17px;
    }
}

```

Logs.css

```

body {
    margin: 0;
    padding: 0;
    background-color: var(--bg-main);
}

.content{
    padding: 60px 0 0 250px;
    max-width: 1220px;
    display: flex;
    justify-content: center;
}

```

```
.container{
  margin-top: 50px;
  width: 1000px;
  background: var(--bg-main);
  border-radius: 8px;
  border: var(--border-default) 1px solid;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  overflow: hidden;
}

.log{
  padding: 16px 20px;
  border-bottom: 1px solid #e0e0e0;
  display: grid;
  grid-template-columns: 140px 100px 1fr 150px;
  gap: 16px;
  align-items: center;
  transition: background 0.2s;
}

.log:hover{
  background: var(--bg-hover);
}

.log:last-child {
  border-bottom: none;
}

.timestamp {
  color: var(--text-muted);
  font-size: 14px;
}

.status {
  padding: 4px 12px;
  border-radius: 12px;
  font-size: 13px;
}
```

```
    font-weight: bold;
    display: flex;
    align-items: center;
    justify-content: center;
}

.status.in {
    background: #d4edda;
    color: #155724;
}

.status.out {
    background: #f8d7da;
    color: #721c24;
}

.item-name {
    font-weight: bold;
    color: var(--text-primary);
}

.author {
    color: var(--text-secondary);
    font-size: 14px;
}

.log p{
    margin:0;
    height: 20px;
    padding: 3px;
    /* border-right: black solid 1px; */
}

.log p:last-child{
    border-right: 0;
}

.log .header{
```

```
    font-weight: bold;
    padding-right: 0;
    border: none;
}

@media (max-width: 768px) {
    .log {
        grid-template-columns: 1fr;
        gap: 8px;
    }
}

/* Top Bar thing */
.top-bar {
    position: fixed;
    display: flex;
    width: 100vw;
    height: 60px;
    background: var(--bg-surface);
    border-bottom: 1px solid var(--border-default);
    justify-content: left;
    align-items: center;
    margin-bottom: 20px;
    z-index: 100;
    padding-left: 22px;
}

.top-bar .title {
    margin: 0;
    font-size: 24px;
    margin-left: 20px;
}

.top-center {
    position: fixed;
    left: 40%;
}
```

```
.search-input {  
  padding: 10px 14px 10px 40px;  
  font-size: 14px;  
  border: 1px solid #ccc;  
  border-radius: 8px;  
  transition: 0.2s;  
  width: 400px;  
}
```

```
.search-input:focus {  
  outline: none;  
  border-color: #888;  
  background: #fff;  
}
```

```
.search-symbol{  
  color: #757575;  
  position: relative;  
  top: 8px;  
  left: 40px;  
}
```

```
@media (max-width: 900px){  
  .content{  
    padding-left: 70px;  
  }  
  .top-bar{  
    width: calc(100vw);  
  }  
  
  .top-center{  
    left: 40%;  
  }  
  
  .top-bar .title {  
    font-size: 20px;  
  }  
  .item {
```

```

        min-width: 600px;
    }
}

@media (max-width: 550px) {
    .content{
        padding-left:0;
    }
}

```

Map-View.css

```

body{
    margin:0;
    background:var(--bg-main);
}

.content {
    width: 100vw;
    height: 100vh;
    background-image: repeating-linear-gradient(90deg, rgba(173, 173, 173, 0.09)
0px, rgba(173, 173, 173, 0.09) 1px, rgba(0, 0, 0, 0) 1px, rgba(0, 0, 0, 0)
96px), repeating-linear-gradient(0deg, rgba(173, 173, 173, 0.09) 0px, rgba(173,
173, 173, 0.09) 1px, rgba(0, 0, 0, 0) 1px, rgba(0, 0, 0, 0) 96px),
repeating-linear-gradient(0deg, rgba(173, 173, 173, 0.09) 0px, rgba(173, 173,
173, 0.09) 1px, rgba(0, 0, 0, 0) 1px, rgba(0, 0, 0, 0) 12px),
repeating-linear-gradient(90deg, rgba(173, 173, 173, 0.09) 0px, rgba(173, 173,
173, 0.09) 1px, rgba(0, 0, 0, 0) 1px, rgba(0, 0, 0, 0) 12px),
linear-gradient(90deg, rgb(255, 255, 255), rgb(255, 255, 255))
}

.container {
    display: flex;
    justify-content: center;
    gap: 50px;
    align-items: center;
    position:absolute;

```

```
    left: 230px;
    top: 100px;
    transform: scale(0.9);
}

.container h1, .container h2{
    text-align:center;
    margin:10px 0
}

h1 {
    font-size: 35px;
}

.top-view{
    display: flex;
    flex-direction: column;
    margin-bottom: 30px;
    margin-left: auto;
    margin-right: auto;
    width: 820px;
    padding: 12px;
    background-color: var(--bg-surface);

    border: 2px solid var(--border-default);
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);
}

.low-shelves {
    display: flex;
    justify-content: center;
    align-items: flex-end;
}

.alignment1{
    width: 252px;
```

```
}

.alignment2{
  width: 300px;
}

.main-shelves {
  display: flex;
  justify-content: center;
  gap: 12px;
  margin-bottom: 12px;
  align-items: center;
}

.walkway-top{
  width: 40px;
  height: 200px;
  background: #fafafa;
  border: 2px dashed #bbb;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 12px;
  color: #666;
  border-radius: 6px;
}

.top-rect{
  width: 60px;
  height: 200px; /* increased height */
  background: #6D9EEB;
  border: 2px solid #555;
  display: flex;
  justify-content: center;
  align-items: center;
  font-weight: 700;
  border-radius: 6px;
  transition: background-color 0.2s ease, color 0.2s ease;
```

```
    cursor: pointer;
}

.big-bookshelf {
    width: 200px;
    height: 40px;
}

.small-bookshelf {
    height: 180px;
    width: 60px;
}

.shelf-view{
    width:300px;
    margin:0 auto;
    display:grid;
    grid-template-columns:repeat(3,1fr);
    /* grid-template-rows:repeat(3,80px); */
    column-gap: 0;
    row-gap: 4px;
    background: var(--bg-hover);
    box-shadow: 0 5px 8px rgba(0,0,0,0.1);
    padding: 4px;
    opacity:1;
    transition:opacity 0.2s ease;
}

.hidden {
    opacity:0;
}

.large-shelf {
    grid-template-rows:repeat(3,60px);
}

.medium-shelf{
```

```
    grid-template-columns: repeat(2, 1fr);
    grid-template-rows: repeat(3, 80px);
}

.small-shelf {
    grid-template-rows: repeat(2, 80px);
}

.slot{
    background: var(--bg-surface);
    border: 1px dashed #aaa;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: 600;
    transition: background-color 0.2s ease, color 0.2s ease;
    cursor: pointer;
}

.slot:hover, .top-rect:hover, .selected{
    background: #ff3f3f;
    color: #fff
}

.shelf-tooltip{
    font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI",
sans-serif !important;
    position: fixed;
    transform: translateX(-50%);
    background-color: #333;
    color: #fff;
    padding: 6px 10px;
    border-radius: 4px;
    font-size: 12px;
    white-space: nowrap;
    pointer-events: none;
    transition: opacity 0.25s ease;
    z-index: 10;
}
```

```
    opacity: 0;
    height: 14px;
}

.shelf-title{
    transition:opacity 0.2s ease;
}

.shelf-tooltip.active{
    opacity: 1;
}

/* Top Bar Styles */

.top-bar {
    position:fixed;
    display: flex;
    width: 100vw;
    height: 60px;
    background: var(--bg-surface);
    border-bottom: 1px solid var(--border-default);
    justify-content: left;
    align-items: center;
    margin-bottom: 20px;
    z-index: 100;
    padding-left:22px;
}

.top-bar .title {
    margin: 0;
    font-size: 24px;
    margin-left:20px;
}

.material-symbols-outlined{
    font-variation-settings: 'FILL' 0, 'wght' 400, 'GRAD' 0, 'opsz' 24;
}
```

```

@media (max-width: 900px){
    .content{
        padding-left: 70px;
    }
    .top-bar{
        width: calc(100vw);
    }

    .top-center{
        left: 40%;
    }

    .top-bar .title {
        font-size: 20px;
    }
    .item {
        min-width: 600px;
    }
}

@media (max-width: 550px) {
    .content{
        padding-left:0;
    }
}

```

Map-view.html

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <title>Storage Area Map</title>
    <link rel="shortcut icon" href="images/map.png" type="image/x-icon">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,w

```

```
ght,FILL,GRAD@20..48,100..700,0..1,-50..200&icon_names=add,delete,edit,home,inventory_2,library_books,login,logout,map,menu,open_in_new,remove,save,search" />
<link rel="stylesheet" href="theme.css">
<link rel="stylesheet" href="map-view.css">
<link rel="stylesheet" href="sidebar.css">

</head>
<body class="MapView">
  <!-- Sidebar Navigation -->
  <div class="sidebar">
    <div class="sidebar-nav">
      <a class="nav-item" href="/index.html">
        <span class="material-symbols-outlined nav-icon">home</span>
        <span class="nav-text">Home</span>
      </a>
      <a class="nav-item" href="/catalogue.html">
        <span class="material-symbols-outlined nav-icon">inventory_2</span>
        <span class="nav-text">Catalogue</span>
      </a>
      <a class="nav-item" href="/admin-dashboard.html">
        <span class="material-symbols-outlined nav-icon">edit</span>
        <span class="nav-text">Edit</span>
      </a>
      <a class="nav-item active" href="/map-view.html">
        <span class="material-symbols-outlined nav-icon">map</span>
        <span class="nav-text">Map Viewer</span>
      </a>
      <a class="nav-item" href="/logs.html">
        <span class="material-symbols-outlined nav-icon">library_books</span>
        <span class="nav-text">Logs</span>
      </a>
    </div>
    <div class="sidebar-footer">
    </div>
  </div>
  <!-- Top Bar -->
  <div class="top-bar">
    <span class="sidebar-button material-symbols-outlined">menu</span>
```

```
<div class="sidebar-header">
  <h1>Shed Storage</h1>
</div>
<div class="bar-left">
  <h2 class="title">Shed Map</h2>
</div>
</div>
<!-- Content -->
<div class="shelf-tooltip">Click to View Shelf</div>
<div class="content">
  <div class="container">
    <div class="section">
      <h2>Top View</h2>
      <div class="top-view">
        <div class="main-shelves">
          <div class="top-rect selected" id="A">A</div>
          <div class="walkway-top"></div>
          <div class="top-rect" id="B">B</div>
          <div class="top-rect" id="C">C</div>
          <div class="walkway-top"></div>
          <div class="top-rect" id="D">D</div>
          <div class="top-rect" id="E">E</div>
          <div class="walkway-top"></div>
          <div class="top-rect" id="F">F</div>
          <div class="top-rect" id="G">G</div>
          <div class="walkway-top"></div>
          <div class="top-rect" id="H">H</div>
        </div>
        <div class="low-shelves">
          <div class="top-rect small-bookshelf" id="I">I</div>
          <div class="alignment1"></div>
          <div class="top-rect big-bookshelf" id="J">J</div>
          <div class="alignment2"></div>
        </div>
      </div>
    </div>
    <div class="section">
      <h2 class="shelf-title">Shelf A</h2>
```

```
        <div class="shelf-view"></div>
      </div>
    </div>
  </div>
  <script defer src="bundle.js"></script>
</body>
</html>
```

Map.css

```
body{
  margin:20px;
  background:var(--bg-surface);
}

.container {
  display: flex;
  justify-content: center;
  gap: 100px;
  align-items: center;
}

.vertical{
  flex-direction: column;
  gap:0;
}

.vertical .top-rect {
  font-size: 30px;
}

.vertical .slot {
  font-size:25px;
}

.vertical h2 {
  font-size: 35px;
}
```

```
.big-title{
  font-size:50px;
}

.vertical h1 {
  font-size:100px;
}

h1, h2{
  text-align:center;
  margin:10px 0
}

h1 {
  font-size: 35px;
}

.top-view{
  display: flex;
  flex-direction: column;
  margin-bottom: 30px;
  margin-left: auto;
  margin-right: auto;
  width: 820px;
  padding: 12px;

  border: 2px solid #555;
}

.low-shelves {
  display: flex;
  justify-content: center;
  align-items: flex-end;
}
```

```
.alignment1{
  width: 252px;
}

.alignment2{
  width: 300px;
}

.main-shelves {
  display:flex;
  justify-content:center;
  gap:12px;
  margin-bottom: 12px;
  align-items:center;
}

.walkway-top{
  width:40px;
  height:200px;
  background:#fafafa;
  border:2px dashed #bbb;
  display:flex;
  align-items:center;
  justify-content:center;
  font-size:12px;
  color:#666;
  border-radius:6px;
}

.top-rect{
  width:60px;
  height:200px; /* increased height */
  background: #6D9EEB;
  border:2px solid #555;
  display:flex;
  justify-content:center;
  align-items:center;
  font-weight:700;
}
```

```
    border-radius: 6px;
}

.big-bookshelf {
    width: 200px;
    height: 40px;
}

.small-bookshelf {
    height: 180px;
    width: 60px;
}

.shelf-view{
    width: 300px;
    margin: 0 auto;
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    /* grid-template-rows: repeat(3, 80px); */
    column-gap: 0;
    row-gap: 4px;
    background: #555;
    padding: 4px;
}

.large-shelf {
    grid-template-rows: repeat(3, 60px);
}

.medium-shelf{
    grid-template-columns: repeat(2, 1fr);
    grid-template-rows: repeat(3, 80px);
}

.small-shelf {
    grid-template-rows: repeat(2, 80px);
}
```

```

.slot{
  background:#fff;
  border:1px dashed #aaa;
  display:flex;
  align-items:center;
  justify-content:center;
  font-weight:600;
}

.selected{
  background: #ff3f3f;
  color: #fff
}

```

Map.html

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Storage Area Map</title>
  <link rel="stylesheet" href="theme.css">
  <link rel="stylesheet" href="map.css">
</head>
<body class="Map">
  <h1>Storage Area Map</h1>
  <div class="container">
    <div class="section">
      <h2>Top View</h2>
      <div class="top-view">
        <div class="main-shelves">
          <div class="top-rect" id="A">A</div>
          <div class="walkway-top"></div>
          <div class="top-rect" id="B">B</div>

```

```

    <div class="top-rect" id="C">C</div>
    <div class="walkway-top"></div>
    <div class="top-rect" id="D">D</div>
    <div class="top-rect" id="E">E</div>
    <div class="walkway-top"></div>
    <div class="top-rect" id="F">F</div>
    <div class="top-rect" id="G">G</div>
    <div class="walkway-top"></div>
    <div class="top-rect" id="H">H</div>
  </div>
  <div class="low-shelves">
    <div class="top-rect small-bookshelf" id="I">I</div>
    <div class="alignment1"></div>
    <div class="top-rect big-bookshelf" id="J">J</div>
    <div class="alignment2"></div>
  </div>
</div>
<div class="section">
  <h2 class="shelf-title">Shelf A</h2>
  <div class="shelf-view">
  </div>
</div>
</div>
<script defer src="bundle.js"></script>
</body>
</html>

```

Sidebar.css

```

/* ===== Sidebar ===== */
.sidebar {
  width: 250px;
  background-color: var(--bg-surface);
  color: var(--text-primary);
  display: flex;
  flex-direction: column;
  position: fixed;

```

```
height: 100%;
border-right: 1px solid var(--border-default);
transition: all 0.3s ease;
z-index: 60;
}

/* Header */
.sidebar-header {
  text-align: center;
  background: var(--brand-gradient);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  height: 60px;
  display: flex;
  justify-content: center;
  align-items: center;
  margin-left: 15px;
  z-index: 101;
}

.sidebar-header h1 {
  font-size: 1.8rem;
  font-weight: 600;
  line-height: 0;
}

/* Navigation */
.sidebar-nav {
  flex: 1;
  padding: 12px 0;
  padding-top: 60px;
}

.nav-item {
  display: flex;
  align-items: center;
  padding: 12px 20px;
  cursor: pointer;
```

```
text-decoration: none;
color: var(--text-primary);
border-left: 4px solid transparent;
transition: background-color 0.2s ease;
}

.nav-item:hover {
  background-color: var(--bg-hover);
}

.nav-item.active {
  background-color: rgba(42, 123, 155, 0.12); /* brand-primary @ low opacity */
  border-left: 4px solid var(--brand-primary);
  font-weight: 500;
}

/* Icons & text */
.nav-icon {
  margin-right: 14px;
  font-size: 24px !important;
  color: var(--text-secondary);
}

.nav-item.active .nav-icon {
  color: var(--brand-primary);
}

.nav-text {
  font-size: 0.95rem;
}

/* Footer */
.sidebar-footer {
  padding: 16px;
  border-top: 1px solid var(--border-default);
}
```

```
/* Buttons */
.logoff-btn,
.login-btn {
  display: flex;
  align-items: center;
  padding: 12px 16px;
  cursor: pointer;
  border-radius: 6px;
  font-size: 0.95rem;
  border: none;
  text-decoration: none;
}

.logoff-btn {
  background-color: var(--danger);
  color: #ffffff;
}

.logoff-btn:hover {
  background-color: var(--danger);
  opacity: 0.9;
}

.login-btn {
  background-color: var(--accent);
  color: #ffffff;
}

.login-btn:hover {
  background-color: var(--accent-hover);
}

/* Icons spacing */
.logoff-icon,
.login-icon {
  margin-right: 12px;
}
```

```
/* Forceopen Styles */
.sidebar.force-open{
  display: flex;
  width: 250px;
}

.force-open .top-center{
  display:none;
}

.force-open .title{
  display:none;
}

.force-open .sidebar-header{
  display:flex;
  z-index: 101;
}

.top-bar.force-open .nav-text{
  display:block;
}

.force-open .nav-item {
  display: flex;
  align-items: center;
  justify-content: flex-start;
  padding: 12px 20px;
  width:initial;
}

.force-open .nav-text{
  display: block;
}

.force-open .nav-item .nav-icon{
  margin-right:14px;
```

```

}

.force-open .login-icon, .force-open .logoff-icon{
    margin-right:12px;
}

.force-open .sidebar-footer{
    display:flex;
    flex-direction: column;
}

.force-open .login-btn, .force-open .logoff-btn{
    padding: 12px 16px;
    font-size: 0.95rem;
    /* width: 250px; */
}

/* Hide delete in catalogue */
.Catalogue .delete-btn {
    display: none;
}

.sidebar .material-symbols-outlined{
    font-variation-settings: "FILL" 0, "wght" 400, "GRAD" 0, "opsz" 24;
}

.sidebar-button{
    cursor: pointer;
}

@media (max-width: 950px) {

    .sidebar-header{
        display:none;
    }

    .sidebar {

```

```
        width: 60px;
    }

    .nav-text{
        display:none;
    }

    .nav-item{
        width:47px;
        padding: 12px 5px;
        display: flex;
        justify-content: center;
    }

    .nav-icon{
        margin-right: 0;
    }

    .logoff-btn,
    .login-btn {
        display: flex;
        align-items: center;
        padding: 5px;
        cursor: pointer;
        border-radius: 6px;
        font-size: 10px !important;
        border: none;
        text-decoration: none;
    }

    .logoff-icon,
    .login-icon {
        margin-right: 0;
    }

    .sidebar-footer {
        padding: 10px;
        display: flex;
        justify-content: center;
```

```

    }

}

@media (max-width: 750px) {
    .sidebar{
        display: none;
    }
}

```

Theme.css

```

:root {
    /* ===== Brand ===== */
    --brand-gradient: linear-gradient(90deg, rgba(42, 123, 155, 1) 0%, rgba(49,
145, 145, 1) 50%, rgba(87, 199, 150, 1) 100%);
    --brand-gradient-1: rgba(42, 123, 155, 1);
    --brand-gradient-2: rgba(49, 145, 145, 1);
    --brand-gradient-3: rgba(87, 199, 150, 1);

    --brand-primary: #2a7b9b;
    --brand-secondary: #57c796;
    --brand-tertiary: rgb(75, 171, 130);

    /* ===== Backgrounds ===== */
    --bg-main: #ffffff;
    --bg-surface: #f7f8fa;
    --bg-hover: #d9dcde;

    /* ===== Text ===== */
    --text-primary: #000000;
    --text-secondary: #555555;
    --text-muted: #7F899E;

    /* ===== Borders ===== */
    --border-default: #e5e7eb;

```

```

/* ===== UI Accents ===== */
--accent: #2a7b9b;
--accent-hover: #245f78;

/* ===== Status ===== */
--success: #2e7d32;
--warning: #f9a825;
--danger: #c62828;
--info: #1565c0;
}

body{
    font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI",
sans-serif;
}

.material-symbols-outlined{
    font-display: block;
}

```

Catalogue.js

```

import { clearTable, getQuery, renderItem } from "../db";

if (document.body.classList.contains("Catalogue")){
    loadSorting();
}

export let itemSortMode = 0;
function loadSorting(){
    const itemSortButton = document.getElementById("sort-name-btn");
    const searchInput = document.getElementById("search-bar");

    itemSortButton.addEventListener('click', async () => {
        let sorted = [];
        if (itemSortMode == 0 || itemSortMode == 2){
            itemSortMode = 1;

```

```

        itemSortButton.querySelector(".sort-icon").textContent =
"arrow_upward";

        //Ascending order

        const queryText = searchInput.value.trim().toLowerCase();

        const querySnapshot = await getQuery(queryText);

        sorted = sortItems(querySnapshot, "asc");

    } else if (itemSortMode == 1) {
        itemSortMode = 2;
        itemSortButton.querySelector(".sort-icon").textContent =
"arrow_downward";

        //Descending order

        const queryText = searchInput.value.trim().toLowerCase();

        const querySnapshot = await getQuery(queryText);

        sorted = sortItems(querySnapshot, "dsc");

    } else {
        console.log("An error has occurred!");
    }

    clearTable();
    renderItem(sorted);

    })
}

//Direction parameter can be "asc" or "dsc"
export function sortItems(snapshot, direction){
    let sortDocs = [...snapshot]; //Copy the snapshot into another variable

```

```

let sorted = false
while (sorted == false){
    sorted = true;
    for (let i = 1; i < sortDocs.length; i++) {
        if (sortDocs[i-1].get("item-name") > sortDocs[i].get("item-name")){
            sorted = false;

            let temp = sortDocs[i-1];
            sortDocs[i-1] = sortDocs[i];
            sortDocs[i] = temp;
        }
    }
}

if (direction == "dsc") {
    let reversedDocs = [];
    for (let i = 0; i < sortDocs.length; i++){
        reversedDocs[i] = sortDocs[sortDocs.length-i-1];
    }
    sortDocs = sortDocs.reverse();
    console.log(reversedDocs);
    console.log(sortDocs);
    console.log(sortDocs.join(',') === reversedDocs.join(','));
}

return sortDocs;
}

```

Dashboard.js

```
import { getCheckedCount, getItemCount } from "../db.js"
```

```

if (document.body.classList.contains("Home")) {
    await LoadStatistics();
}

async function LoadStatistics() {
    const propsCount = document.getElementById("total-props")
    const checkedCount = document.getElementById("checked-out-props")
    propsCount.textContent = await getItemCount();
    checkedCount.textContent = await getCheckedCount();
}

```

Db.js

```

// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { doc, getDoc, getDocs, getFirestore, collection, query, where, setDoc,
updateDoc, deleteDoc, serverTimestamp, orderBy } from "firebase/firestore";
import { showLoading, hideLoading, isLoggedIn } from "./script.js";
import { itemSortMode, sortItems } from "./catalogue.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
    apiKey: "AIzaSyBzDsZwqblxL0rPNAhwPOiO4uCfED6KQtA",
    authDomain: "stmarys-theatre-inventory.firebaseio.com",
    projectId: "stmarys-theatre-inventory",
    storageBucket: "stmarys-theatre-inventory.firebaseio.com",
    messagingSenderId: "711202423549",
    appId: "1:711202423549:web:b44c44d405525c9c63f494",
    measurementId: "G-ZGKL3NF0NM"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);

const db = getFirestore(app);
const colRef = collection(db, "props-shed");

```

```

const docSnap = await getDocs(colRef);

const inventoryList = document.querySelector(".inventory-list");
const addItemRow = document.getElementById('addItemRow');

const ITEMS_PER_PAGE = 10;

console.log("Start Loading!");
if (document.body.classList.contains("MainPage") ||
document.body.classList.contains("Catalogue")){

    const searchParams = new URLSearchParams(document.location.search);
    const searchParamBar = searchParams.get("search") || "";
    const currentPage = parseInt(searchParams.get("page")) || 1;

    document.getElementById("search-bar").value = searchParamBar;

    let snapshot = await getQuery(searchParamBar);
    renderItem(snapshot, currentPage);
    hideLoading();
} else if (document.body.classList.contains("Item")){
    const searchParams = new URLSearchParams(document.location.search);
    const paramId = searchParams.get("id"); //Get "id" from the URL
    await displayItem(paramId); //Pass the ID into the display item function
    console.log("Finished Loading!");
    hideLoading();
}

function createTag(name, color){
    let tag = document.createElement('span');
    tag.classList.add('tag', color);
    tag.textContent = name;
    return tag;
}

function isMapSmall(){
    if (document.body.clientWidth < 700){
        return true;
    }
}

```

```

    }
}

async function displayItem(id) {
  const itemDoc = doc(db, "props-shed", id);
  const itemSnap = await getDoc(itemDoc);

  //If the item does not exist, display a message and redirect the user
  if (!itemSnap.exists()) {
    alert("This prop does not exist!");
    window.location.href = "catalogue.html";
  }
  let name = itemSnap.get("item-name");
  let location = itemSnap.get("item-location");
  let tagNames = [];
  let tagColors = [];
  for (let i = 0; i < itemSnap.get("tag-colors").length; i++) {
    tagNames.push(itemSnap.get("tag-names")[i]);
    tagColors.push(itemSnap.get("tag-colors")[i]);
  }
  let status = itemSnap.get("item-status");

  //Create the taglist
  const tagList = document.createElement('div');
  tagList.classList.add("tag-list");
  for (let i = 0; i < tagNames.length; i++) {
    //Create new tag objects for each tag required for the item
    if (tagNames[i] != null) {
      tagList.appendChild(createTag(tagNames[i], tagColors[i]));
    }
  }
  console.log("Found!");

  const nameText = document.querySelector(".name");
  const locationText = document.querySelector(".location");
  const statusText = document.querySelector(".status");
  const idText = document.querySelector(".id");
  const statusSymbol = document.querySelector(".status-symbol")

```

```

const checkButton = document.querySelector('.check-inout-button');
const tagsBox = document.querySelector(".tags")
//Convert the tag objects into HTML
const tags = [...tagList.children].map(t => t.outerHTML).join('');
nameText.textContent = name;
locationText.textContent = location;
if (status === "In Shed"){
    statusSymbol.textContent = "check_circle";
    statusSymbol.classList.add("in-shed");
    checkButton.textContent = "Check Out"
} else {
    statusSymbol.textContent = "do_not_disturb_on";
    statusSymbol.classList.add("checked-out");
    checkButton.textContent = "Check In"
}
statusText.textContent = status;
idText.textContent = "ID: " + id;
tagsBox.innerHTML = tags;

const mapFrame = document.querySelector(".map"); //Fetch the map element from
HTML

if (isMapSmall()){
    mapFrame.src = `map.html?location=${location}&size=small`; //Update the
map's seach parameter to the item's location
    mapFrame.height = 870;
    mapFrame.width = 850;
} else {
    mapFrame.src = `map.html?location=${location}`;
    mapFrame.height = 600;
    mapFrame.width = 1260;
}
document.title = "Item: " + name;

window.addEventListener('resize', () => {
    if (isMapSmall() && mapFrame.src == (window.location.origin +
"/map.html?location=" + location)){
        mapFrame.height = 870;

```

```

        mapFrame.width = 850;
        mapFrame.src = `map.html?location=${location}&size=small`; //Update the
map's seach parameter to the item's location
    } else if (!isMapSmall() && mapFrame.src == (window.location.origin +
"/map.html?location=" + location + "&size=small")){
        mapFrame.height = 600;
        mapFrame.width = 1260;
        mapFrame.src = `map.html?location=${location}`;
    }
    document.title = "Item: " + name;
})

checkButton.addEventListener('click', async () => {
    if (!(await isLoggedIn())){
        let loginConfirm = confirm("You are not logged in! Press OK to login");
        if (loginConfirm){
            window.location.href = "login.html?redirect=" +
encodeURIComponent(window.location.pathname + window.location.search);
            return;
        } else {
            return;
        }
    }

    let logAuthor = "Unknown";
    let logStatus = "Unknown";
    let itemStatus = "Unknown";
    const logItem = itemSnap.ref;

    //Uses if statements to toggle status
    if (status === "In Shed"){
        logAuthor = prompt("Who is checking out this item?");
        logStatus = "Checked Out";
        itemStatus = "Checked Out";
    } else {
        logAuthor = prompt("Who is checking in this item?");
        logStatus = "Checked In";
        itemStatus = "In Shed";
    }

```

```

}

//Only continue if user entered a value in the prompt
if (!(logAuthor === null)){
  //Creates a new log
  const newLog = {
    "log-author": logAuthor,
    "log-item": logItem,
    "log-status": logStatus,
    "log-timestamp": serverTimestamp(),
  };

  const docRef = doc(collection(db, "logs")); //Fetches the logs collection
  await setDoc(docRef, newLog); //Creates a new log

  //Update the item itself

  //Gets all the logs of the item
  let itemLogs = [];
  for (let i = 0; i < itemSnap.get("item-logs").length; i++){
    itemLogs.push(itemSnap.get("item-logs")[i]);
  }

  //Adds new log to the item's logs
  itemLogs.push(docRef);

  //Updates the item with the new log and toggled status
  await updateDoc(itemDoc, {
    "item-logs": itemLogs,
    "item-status": itemStatus
  });

  window.location.reload(); //Refreshes the page to see changes
}

});
}

```

```

export function clearTable(){
  const items = inventoryList.querySelectorAll(".item-wrapper");
  const newItemRow = inventoryList.querySelector(".new-item");

  const addItemRow = document.getElementById("addItemRow");
  if (addItemRow){
    const addItemText = addItemRow.querySelector(".add-text");
    const addItemSymbol = addItemRow.querySelector(".add-symbol");
    addItemText.textContent = "Add Item";
    addItemSymbol.textContent = 'add';
  }

  if (newItemRow) newItemRow.remove();

  items.forEach(item => {
    item.remove();
  })
}

const searchInput = document.getElementById("search-bar");
let searchTimeout = null; //Initialize the timeout variable

if (document.body.classList.contains("Catalogue") ||
document.body.classList.contains("MainPage")){
  searchInput.addEventListener("keydown", async (event) => { //Runs function
every keystroke

    clearTimeout(searchTimeout); //Clears the timer every keystroke

    searchTimeout = setTimeout(async () => {
      const queryText = searchInput.value.trim().toLowerCase(); //Trims the text
query
      let currentSnapshot = await getQuery(queryText); //Fetches the items from
the database

      //Verifies if the list it to be sorted in ascending or descending order

```

```

    if (itemSortMode == 1) {
        currentSnapshot = sortItems(currentSnapshot, "asc");
    } else if (itemSortMode == 2) {
        currentSnapshot = sortItems(currentSnapshot, "dsc");
    }

    // Update search parameter without refreshing page
    const params = new URLSearchParams();
    if (queryText) params.set("search", queryText);
    params.set("page", "1");
    history.replaceState(null, "", `?${params.toString()}`);

    clearTable();
    renderItem(currentSnapshot, 1);

    }, 250);
});
}

export async function getQuery(queryText) {
    if (!queryText) {
        console.log("Search cleared");
        const snapshot = await getDocs(collection(db, "props-shed"));

        return snapshot.docs; //If there is no query, return the full collection
        without filtering.
    }

    try {
        // Get all documents
        const colRef = collection(db, "props-shed");
        const snapshot = await getDocs(colRef);

        //Trim the text if it has any trailing spaces
        const searchTerm = queryText.toLowerCase().trim();

        const filteredDocs = snapshot.docs.filter(doc => { //Filters through all
items in the database

```

```

const itemName = doc.data()['item-name']?.toLowerCase() || '';
const location = doc.data()['item-location']?.toLowerCase() || '';

// Splits both item name and location into words
const itemWords = itemName.split(/\s+/);
const locationWords = location.split(/\s+/);

//Collects all tag names into one array
const tags = doc.data()['tag-names'].map(tag => tag?.toLowerCase() || '');

// Check if any word starts with the query OR if the text includes the
query
// for item name, location, and tag name
return itemWords.some(word => word.startsWith(searchTerm)) ||
    itemName.includes(searchTerm) ||
    locationWords.some(word => word.startsWith(searchTerm)) ||
    location.includes(searchTerm) ||
    tags.some(word => word.startsWith(searchTerm));
});

return filteredDocs;// <--- Datatype is snapshot

} catch (error) { //If an error occurs, the website won't crash and simply
return an empty snapshot
    console.error("Search error:", error);
    return [];
}

}

export function renderItem(snapshot, currentPage = 1) { //Page 1 by default
    const totalPages = Math.max(1, Math.ceil(snapshot.length / ITEMS_PER_PAGE));
    //Find the total number of pages
    const start = (currentPage - 1) * ITEMS_PER_PAGE; //The index of the first
item to be added
    const pageItems = snapshot.slice(start, start + ITEMS_PER_PAGE) //Slices the
snapshot to only have 50 items in the page.

```

```

pageItems.forEach((doc) => { //Loops through every document in snapshot

    //Set temporary variables for each attribute of an item
    let id = doc.get("item-id");
    let name = doc.get("item-name");
    let location = doc.get("item-location");

    let tagNames = [];
    let tagColors = [];

    //Populating tag data onto parallel arrays
    for (let i = 0; i < doc.get("tag-colors").length; i++) {
        tagNames.push(doc.get("tag-names") [i]);
        tagColors.push(doc.get("tag-colors") [i]);
    }
    let status = doc.get("item-status");

    const tagList = document.createElement('div');
    tagList.classList.add("tag-list");
    for (let i = 0; i < tagNames.length; i++) {
        if (tagNames[i] != null) tagList.appendChild(createTag(tagNames[i],
tagColors[i]));
    }

    const newItem = document.createElement("div");
    inventoryList.insertBefore(newItem, addItemRow);
    const tags = [...tagList.children].map(t => t.innerHTML).join('');

    newItem.innerHTML = `
        <div class="item-wrapper" id="${id}">
            <button class="delete-btn material-symbols-outlined">delete</button>
            <div class="item">
                <div class="item-name"><p>${name}</p></div>
                <div class="item-location"><p>${location}</p></div>
                <div class="item-tags">${tags}</div>
                <div class="item-status"><p>${status}</p></div>
                <div class="view material-symbols-outlined">open_in_new</div>
            </div>
    `

```

```

        </div>`;
    });

    if (snapshot.length === 0) { //If there are no items, render an appropriate
alert
        const newItem = document.createElement("div");
        inventoryList.insertBefore(newItem, addItemRow);
        newItem.innerHTML = `<div class="item-wrapper empty-search-warning">No items
matched your search. Try searching something else...</div>`;
    }

    // Render pagination controls
    renderPaginationControls(currentPage, totalPages);
}

function renderPaginationControls(currentPage, totalPages) {
    //Removes the controls if they exist already to reset them
    const existing = document.getElementById("pagination-controls");
    if (existing) existing.remove();

    if (totalPages <= 1) return; //Do not add the controls if there is only 1 page

    //Gets the current search query
    const search = new URLSearchParams(document.location.search).get("search") ||
    "";

    //Gets the URL to the previous and next pages
    function pageUrl(page) {
        const params = new URLSearchParams();
        if (search) params.set("search", search);
        params.set("page", page);
        return `?${params.toString()}`;
    }

    //Creates the controls
    const controls = document.createElement("div");
    controls.id = "pagination-controls";
    controls.innerHTML = `

```

```

    ${currentPage > 1
      ? ``
      : ``}
    <span id="pageIndicator">Page \${currentPage} of \${totalPages}</span>
    \${currentPage < totalPages
      ? ``
      : ``}
    `;

    const contentContainer = document.querySelector\\(".content"\\);

    // Insert after inventory list
    contentContainer.append\\(controls\\);
  }

export async function renderLogs\\(\\) {
  const q = query\\(collection\\(db, "logs"\\), orderBy\\("log-timestamp", "desc"\\)\\);
  //Collect all logs, ordered based on the timestamp
  const snapshot = await getDocs\\(q\\);
  const container = document.querySelector\\(".container"\\);
  snapshot.docs.forEach\\(async \\(doc\\) => {

    //Fetching data
    let logAuthor = doc.get\\("log-author"\\);
    let logTimestamp = doc.get\\("log-timestamp"\\);
    let formattedTimestamp = logTimestamp.toDate\\(\\).toLocaleString\\("sv-SE", {
    //Convert the UTC timecode into a readable format
      year: 'numeric',
      month: '2-digit',
      day: '2-digit',
      hour: '2-digit',
      minute: '2-digit'
    }\\);

    let logStatus = doc.get\\("log-status"\\);

```

```

let logClass;

if (logStatus == "Checked In"){
  logClass = "in";
} else {
  logClass = "out";
}

let logItemRef = doc.get("log-item");
let logItemName = (await getDoc(logItemRef)).get("item-name");

//Ceating the HTML element for a log
const logHTML = `
<div class="log">
  <p class="timestamp">${formattedTimestamp}</p>
  <p class="status ${logClass}">${logStatus}</p>
  <p class="item-name">${logItemName}</p>
  <p class="author">by ${logAuthor}</p>
</div>
`;

const logElement = document.createElement("div");
container.appendChild(logElement); //Appending the log onto the GUI element
logElement.outerHTML = logHTML;
});
}

export async function generateId(){
  //Default value if there are no items. Fixed length of 5
  let id = "00000";

  //Access the ID of the mostly recently added document
  const snap = await getDocs(colRef);
  const lastId = snap.docs[snap.size-1].id;

  //Converts the string into a number and increments
  let number = Number(lastId) + 1;

```

```

    // Convert the new number back into a string and pad it with leading zeros
    // to maintain a consistent 5-character identifier format
    id = ("0".repeat((5-number.toString().length)) + number.toString());

    return id;
}

export async function addItem(docData, newId) {
    await setDoc(doc(db, "props-shed", newId), docData);
}

export async function removeItem(targetId) {
    await deleteDoc(doc(db, "props-shed", targetId));
}

export async function getItemCount() {
    const snap = await getDocs(colRef);
    return snap.size;
}

export async function getCheckedCount() {
    const checkedQuery = query(colRef, where("item-status", "==", "Checked Out"));
    const snap = await getDocs(checkedQuery);
    return snap.size;
}

```

Entry.js

```

import './db.js'
import './map.js'
import './script.js'
import './login.js'
import './catalogue.js'
import './dashboard.js'
import './logs.js'

```

Login.js

```

if (document.body.classList.contains("Login")) {

```

```

    loginLoad();
}

function loginLoad(){
    const searchParams = new URLSearchParams(document.location.search); //Gets
all search parameters from the URL
    const paramId = searchParams.get("err"); //Finds the parameter named "err"
    if (paramId == "error"){
        alert("Incorrect username or password"); //Alerts user of incorrect
credentials if the server returned an error.
    }
    const redirect = searchParams.get('redirect') || '/admin-dashboard.html';
    document.getElementById('redirect-field').value = redirect;
}

```

Logs.js

```

import { renderLogs } from "../db";

if (document.body.classList.contains("Logs")){
    renderLogs();
}

```

Map.js

```

if (document.body.classList.contains("Map")){
    loadMap();
}

if (document.body.classList.contains("MapView")){
    loadMapView();
}

function loadMap() {
    const searchParams = new URLSearchParams(document.location.search);
    const paramLocation = searchParams.get("location");
    const paramSize = searchParams.get("size");
    const shelfLetter = paramLocation.charAt(0);

```

```

const shelfNumber = paramLocation.charAt(1);
const selectedShelf = document.getElementById(shelfLetter);
const container = document.querySelector(".container");
const pageTitle = document.querySelector("h1");

if (paramSize == "small"){
    container.classList.add("vertical");
    pageTitle.classList.add("big-title")
}

selectedShelf.classList.add("selected");
console.log(paramLocation);

const shelfView = document.querySelector(".shelf-view");
const shelfTitle = document.querySelector(".shelf-title");

shelfTitle.textContent = "Shelf " + shelfLetter;

if (shelfLetter == "J"){
    shelfView.classList.add("large-shelf");
    generateShelf(shelfLetter, shelfNumber, 9); //Generate a large shelf
with 9 slots
} else if (shelfLetter == "I"){
    shelfView.classList.add("medium-shelf");
    generateShelf(shelfLetter, shelfNumber, 6); //Generate a medium shelf
with 6 slots
}
else {
    shelfView.classList.add("small-shelf");
    generateShelf(shelfLetter, shelfNumber, 6) //Generate a small shelf with
6 slots
}
}

function generateShelf(letter, selectedSlot, slots){
    const shelfView = document.querySelector(".shelf-view"); //Fetch the shelf
element from the front-end

```

```

shelfView.innerHTML = ""; //Clear the shelf if there was any existing data
inside it

for (let i = 0; i < slots; i++){
    let slot = document.createElement('div'); //Create a new DIV element for
each slot in the grid
    slot.classList.add("slot");
    if ((i+1) == selectedSlot){
        slot.classList.add("selected"); //Adds a red background on the
specified slot (ex. A2)
    }
    slot.textContent = letter + (i+1).toString(); //Adds text to each slot
specifying their ID
    slot.id = letter + (i+1).toString();
    shelfView.appendChild(slot); //Add the slot to the original shelf
element
}
}

function loadMapView(){
    selectShelf();

    document.addEventListener('click', e => {
        if (e.target.classList.contains("top-rect")){
            console.log("clicked");
            const selectedShelf = document.querySelector(".selected");
            selectedShelf.classList.remove("selected");
            e.target.classList.add("selected");
            selectShelf();
        } else if (e.target.classList.contains("slot")){
            console.log(e.target.id);

            window.location.href =
`/catalogue.html?search=${encodeURIComponent(e.target.id)}`;

        }
    })
}

```

```

    loadToolTips();
}

export function verifyShelf(location){
    const shelfLetter = location.charAt(0);
    const shelfNumber = location.charAt(1);

    const validLetters = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J"];

    if(location.length !== 2){
        return false;
    }

    if (validLetters.includes(shelfLetter)){
        if (shelfLetter === "J"){
            if (parseInt(shelfNumber) <= 9 && parseInt(shelfNumber) > 0){
                return true;
            } else {
                return false;
            }
        } else {
            if (parseInt(shelfNumber) <= 6 && parseInt(shelfNumber) > 0){
                return true;
            } else {
                return false;
            }
        }
    } else {
        return false;
    }
}

function selectShelf(){
    const selectedShelf = document.querySelector(".selected");
    const shelfLetter = selectedShelf.id;
    console.log(shelfLetter);
}

```

```

const shelfTitle = document.querySelector(".shelf-title");
const shelfView = document.querySelector(".shelf-view");
shelfView.classList.add("hidden");
shelfTitle.classList.add("hidden");
setTimeout(() => {
    if (shelfLetter == "J"){
        shelfView.classList.remove("small-shelf");
        shelfView.classList.remove("medium-shelf");
        shelfView.classList.add("shelf-view");
        shelfView.classList.add("large-shelf");
        generateShelf(shelfLetter, 0, 9); //Generate a large shelf with 9
slots
    } else if (shelfLetter == "I"){
        shelfView.classList.remove("small-shelf");
        shelfView.classList.remove("large-shelf");
        shelfView.classList.add("shelf-view");
        shelfView.classList.add("medium-shelf");
        generateShelf(shelfLetter, 0, 6); //Generate a medium shelf with 6
slots
    }
    else {
        shelfView.classList.remove("large-shelf");
        shelfView.classList.remove("medium-shelf");
        shelfView.classList.add("shelf-view");
        shelfView.classList.add("small-shelf");
        generateShelf(shelfLetter, 0, 6) //Generate a small shelf with 6
slots
    }

    shelfTitle.textContent = "Shelf " + shelfLetter;
}, 200);

setTimeout(() => {
    shelfView.classList.remove("hidden");
    shelfTitle.classList.remove("hidden");
}, 300);
}

```

```

function loadToolTips(){
    window.onmousemove = function(e) {

        if (e.target.classList.contains("top-rect") ||
e.target.classList.contains("slot")){

            let text = this.document.querySelector(".shelf-tooltip")
            if (e.target.classList.contains("top-rect")){
                text.textContent = "Click to View Shelf"
            } else {
                text.textContent = "View Items in Location"
            }
            if (!text.classList.contains('active')) {
                text.classList.add('active');
            } else {
                var tipDist = 18;
                // Set new tooltip location
                text.style.top = (e.clientY + tipDist) + 'px';
                text.style.left = (e.clientX) + 'px';
            }

        } else {
            let text = this.document.querySelector(".shelf-tooltip")
            if (text.classList.contains('active')) {
                text.classList.remove('active');
            }
        }

    }
}

```

Script.js

```

import { addItem, generateId, removeItem } from "../db";
import { verifyShelf } from "../map.js";

export function showLoading() {
    const overlay = document.getElementById("loadingOverlay");
    if (overlay){
        overlay.classList.remove("hidden");
    }
}

```

```

    }
}

export function hideLoading() {
    const overlay = document.getElementById("loadingOverlay");
    if (overlay) {
        overlay.classList.add("hidden");
    }
}

function loadMenuButton() {

    try {
        const sideBar = document.querySelector(".sidebar");
        const topBar = document.querySelector(".top-bar");
        const sideBarButton = document.querySelector(".sidebar-button");

        sideBarButton.addEventListener("click", () => {
            console.log("clicked");
            if (document.body.clientWidth <= 950) {
                sideBar.classList.toggle("force-open");
                topBar.classList.toggle("force-open");
            }
        })

        document.addEventListener("click", (e) => {

            if (!sideBar.contains(e.target) &&
                !sideBarButton.contains(e.target)) {

                sideBar.classList.remove("force-open");
                topBar.classList.remove("force-open");
            }
        })

        window.addEventListener("resize", () => {

```

```

        if (document.body.clientWidth > 950 &&
(sideBar.classList.contains("force-open") ||
topBar.classList.contains("force-open"))){
            sideBar.classList.remove("force-open");
            topBar.classList.remove("force-open");
        }
    })
} catch (error){
    console.log(error);
}
}

showLoading();
loadMenuButton();

const logoutButtonHTML = `
<a class="logoff-btn"
href="/logout?redirect=${encodeURIComponent(window.location.pathname +
window.location.search)}">
    <span class="material-symbols-outlined logoff-icon">logout</span>
    <span class="nav-text">Log Out</span>
</a>
`;

const loginButtonHTML = `
<a class="login-btn"
href="/login.html?redirect=${encodeURIComponent(window.location.pathname +
window.location.search)}">
    <span class="material-symbols-outlined logoff-icon">login</span>
    <span class="nav-text">Log In</span>
</a>
`;

//Check whether user is logged in

export async function isLoggedIn(){

```

```

    try{
        const response = await fetch("/api/auth/status");
        const data = await response.json();

        return data.loggedIn;
    } catch(error) {
        console.error("Error checking login status: ", error);
        return false;
    }
}

async function updateLogin(){
    try{
        let logged = await isLoggedIn();

        const footer = document.querySelector(".sidebar-footer");

        if (logged == true) {
            footer.innerHTML = logoutButtonHTML;
        } else {
            footer.innerHTML = loginButtonHTML;
        }
    } catch(error) {
        console.error("Error checking login status: ", error);
    }
}

if (document.querySelector(".sidebar")){
    await updateLogin();
}

if (document.body.classList.contains("MainPage")){
    MainLoad();
} else if (document.body.classList.contains("Catalogue")) {
    loadOpenButtons();
}

function MainLoad(){

```

```

const inventoryList = document.querySelector('.inventory-list');
const addItemRow = document.getElementById('addItemRow');
const addItemText = addItemRow.querySelector('.add-text');
const addItemSymbol = addItemRow.querySelector('.add-symbol');

addItemRow.addEventListener('click', () => {
  // Prevent multiple form rows
  if (document.querySelector('.item.new-item')){
    document.querySelector('.item.new-item').remove();

    addItemText.textContent = "Add Item";
    addItemSymbol.textContent = 'add';
    return;
  }
  const newRow = document.createElement('div');
  // Generates the form to create a new item
  newRow.classList.add('item', 'new-item');
  newRow.innerHTML = `
    <div class="item-name"><input type="text" placeholder="New Item
Name" maxlength="25"></div>
    <div class="item-location"><input type="text" placeholder="Location
ID" maxlength="3"></div>
    <div class="item-tags">

      <span class="small-label">Preview</span>
      <div class="tag-list"></div>

      <span class="small-label">Quick Add</span>
      <div class="quick-tag-buttons">
        <button type="button" class="quick-tag-btn
quick-borrowed">Borrowed</button>
        <button type="button" class="quick-tag-btn
quick-dispose">Dispose</button>
      </div>

      <span class="small-label">Custom Tags</span>
      <div class="custom-tag-row">

```

```

        <input type="text" class="tag-input" placeholder="Tag"
maxlength="20">

        <div class="tag-color-picker tag tag-grey"></div>
        <button type="button" class="add-tag-btn">+</button>
    </div>
</div>

<div class="item-status">
    <select>
        <option>In Shed</option>
        <option>Checked Out</option>
    </select>
</div>
<div class="add submit-item material-symbols-outlined">add</div>
`;

addItemText.textContent = "Close Dialogue";
addItemSymbol.textContent = 'remove';

inventoryList.insertBefore(newRow, addItemRow);

// Tag adding behavior
const tagInput = newRow.querySelector('.tag-input');
const tagList = newRow.querySelector('.tag-list');
const addTagButton = newRow.querySelector('.add-tag-btn');
const colorPicker = newRow.querySelector('.tag-color-picker');

let selectedColorClass = "tag-grey"; // default tag color

// Show color picker when the color picker is pressed
colorPicker.addEventListener('click', () => {

    let popup = newRow.querySelector('.tag-color-popup');
    if (popup) {
        popup.remove();
        return;
    }

```

```

    setTimeout(() => {
        document.addEventListener('click', closePopup);
    }, 0);

    function closePopup() {
        popup?.remove();
        document.removeEventListener('click', closePopup);
    }

    popup = document.createElement('div');
    popup.classList.add('tag-color-popup');

    const colors = ["tag-grey", "tag-red", "tag-orange", "tag-yellow",
"tag-green", "tag-blue", "tag-purple"];

    colors.forEach(c => {
        const opt = document.createElement('div');
        opt.classList.add('tag-color-option', 'tag', c);
        opt.addEventListener('click', () => {
            selectedColorClass = c;
            colorPicker.className = "tag-color-picker tag " + c;
        });
        popup.appendChild(opt);
    });

    colorPicker.appendChild(popup);
});

//When a tag is added using the custom part
addTagButton.addEventListener('click', () => {
    const value = tagInput.value.trim();
    if (!value) return;

    for (let tag of tagList.children) {
        if (tagList.childElementCount > 4) return;
        if (tag.textContent === value) return;
    }
}

```

```

    const tag = document.createElement('span');
    tag.classList.add('tag', selectedColorClass, 'tag-preview');
    tag.textContent = value;
    tagList.appendChild(tag);

    tagInput.value = '';
  });

  //When a tag is added using quick tags
  const quickBorrowed = newRow.querySelector('.quick-borrowed');
  const quickDispose = newRow.querySelector('.quick-dispose');

  function addQuickTag(text, colorClass) {
    const tag = document.createElement('span');

    for (let tag of tagList.children) {
      if (tag.textContent === text) return;
      if (tag.textContent === value) return;
    }

    tag.classList.add('tag', colorClass, 'tag-preview');
    tag.textContent = text;
    tagList.appendChild(tag);
  }

  // Standard tags
  quickBorrowed.addEventListener('click', () => addQuickTag("Borrowed",
"tag-green"));
  quickDispose.addEventListener('click', () => addQuickTag("Dispose",
"tag-red"));

  document.addEventListener('click', (tag) => {
    if (tag.target.classList.contains('tag-preview')){
      tag.target.remove();
    }
  })

  // If the submit button is pressed

```

```

    newRow.querySelector('.submit-item').addEventListener('click', async ()
=> {

    const id = await generateId();

    //Get the text input from the website front end
    const name = (newRow.querySelector('.item-name
input').value).toLowerCase();
    const location = newRow.querySelector('.item-location input').value;

    if (!name || !location){
        alert("Please enter a name and location");
        return;
    }; //Reject input if either the name or location is empty
    console.log(verifyShelf(location));
    if (!verifyShelf(location)){
        alert("Please enter a valid location format between A1 to J9");
        return;
    }

    const status = newRow.querySelector('.item-status select').value;
    for (let tag of tagList.children){
        //Remove the "preview" tag so they become permanent tags
        tag.classList.remove("tag-preview");
    }
    const tags = [...tagList.children].map(t => t.outerHTML).join('');

    //Convert tag list into 2 one dimensional arrays
    const tagNames = [...tagList.children].map(t =>
t.textContent.trim());
    const tagColors = [...tagList.children].map(t => t.classList[1]);

    const addItemRow = document.getElementById("addItemRow");
    const addItemText = addItemRow.querySelector(".add-text");
    const addItemSymbol = addItemRow.querySelector(".add-symbol");

    addItemText.textContent = "Add Item";
    addItemSymbol.textContent = 'add';
    newRow.outerHTML = `

```

```

        <div class="item-wrapper" id="${id}">
            <button class="delete-btn
material-symbols-outlined">delete</button>
            <div class="item">
                <div class="item-name"><p>${name}</p></div>
                <div class="item-location"><p>${location}</p></div>
                <div class="item-tags">${tags}</div>
                <div class="item-status"><p>${status}</p></div>
                <div class="view
material-symbols-outlined">open_in_new</div>
            </div>
        </div>
    `;

    const newItemData = {
        "item-id": id,
        "item-location": location,
        "item-name": name,
        "item-status": status,
        "tag-colors": tagColors,
        "tag-names": tagNames,
        "item-logs": []
    };

    addItem(newItemData, id);

    });
});

const inventory = document.querySelector('.inventory-list');

// Delete item when clicking delete button
document.addEventListener('click', (button) => {
    if (button.target.classList.contains('delete-btn')) {
        const deleteConfirmation = document.createElement('div');
        document.body.append(deleteConfirmation);

        deleteConfirmation.outerHTML = `

```

```

    <div class="delete-confirmation hidden">
      <div class="delete-confirmation-box">
        <h3>Are you sure?</h3>
        <p>This action cannot be undone!</p>
        <div class="confirmation-container">
          <div class="cancel">Cancel</div>
          <div class="confirm">Confirm</div>
        </div>
      </div>
    </div>
    `

    const confirmationBox =
document.querySelector(".delete-confirmation")

    setTimeout(() => {
      // This code will execute after a given amount of seconds
      confirmationBox.classList.remove('hidden');
    }, 100);

    const cancel = document.querySelector(".cancel");
    cancel.addEventListener('click', (e) =>{
      confirmationBox.classList.add('hidden');
      setTimeout(() => {
        confirmationBox.remove();
      }, 200);
    });

    const confirm = document.querySelector(".confirm");
    confirm.addEventListener('click', (e) => {
      const toBeRemoved = button.target.parentNode
      console.log(toBeRemoved.id + ": is removed!");
      toBeRemoved.remove();
      confirmationBox.classList.add('hidden');
      setTimeout(() => {
        confirmationBox.remove();
      }, 200);

      //DELETE THE ITEM

```

```

        removeItem(toBeRemoved.id);
    })
}

});

loadOpenButtons();
}

function loadOpenButtons() {
    document.addEventListener("click", (e) => {
        const openBtn = e.target.closest(".view");
        if (!openBtn) return;
        if (openBtn.parentNode.classList.contains("new-item")) return;

        const itemWrapper = openBtn.closest(".item-wrapper");
        const itemID = itemWrapper.id;

        // Navigate with ID in URL
        window.open(`item.html?id=${itemID}`, "_blank");
    });
}

```

Server.js

```

const express = require("express");
const session = require("express-session");
const path = require("path");

const app = express();

app.use(express.urlencoded({ extended: true }));
app.use(express.json());

// Serve public files to the user
app.use("/", express.static(path.join(__dirname, "dist/public")));

// Sessions
app.use(
    session({

```

```

    secret: "super-secret-key",
    resave: false, //Only save session when session is changed or updated
    saveUninitialized: false, //Remove failed login session attempts
    cookie: {
      maxAge: 15 * 60 * 1000 //Require re-authentication after 15 minutes
    }
  })
);

// Middleware to protect pages
function requireLogin(req, res, next) {
  if (!req.session.user) {
    req.session.redirectTo = req.originalUrl; //If user logs in from a public page,
    redirect back to public page
    return res.redirect("/login.html");
  }
  next();
}

// Protected admin page
app.get("/admin-dashboard.html", requireLogin, (req, res) => {
  res.sendFile(path.join(__dirname, "dist/admin-dashboard.html"));
});

//Protected logs page
app.get("/logs.html", requireLogin, (req, res) => {
  res.sendFile(path.join(__dirname, "dist/logs.html"));
});

// Login page
app.get("/login.html", (req, res) => {
  res.sendFile(path.join(__dirname, "dist/login.html"));
});

// Example user
const USER = {
  email: "admin@shedstorage.net",
  password: "titans",
};

// Login handler
app.post("/login", (req, res) => {

```

```

const { email, password, redirect } = req.body; //Gets data from login page when user
submits

if (email === USER.email && password === USER.password) { //Checks whether both
credential match
  req.session.user = email;
  const redirectTo = req.session.redirectTo || redirect;
  delete req.session.redirectTo; //Clears the "redirectTo" variable for later user
  return res.redirect(redirectTo);
}
res.redirect('/login.html?err=error') //If they do not match, refresh the page with
an error
});

// Logout
app.get("/logout", (req, res) => {
  const redirectPath = req.query.redirect;
  req.session.destroy(() => {
    //If user logs out from a protected page, redirect to the main page
    if (redirectPath === "/logs.html" || redirectPath === "/admin-dashboard.html"){
      res.redirect("/");
    } else {
      //If user logs out from a public page, simply redirect back to the same page
      res.redirect(redirectPath);
    }
  });
});

//Custom logged-in API chheck for later use throughout website
app.get("/api/auth/status", (req, res) => {
  if (req.session && req.session.user) {
    return res.json({ loggedIn: true, user: req.session.user });
  }
  res.json({ loggedIn: false });
});

app.listen(3000, () =>
  console.log("Server running on http://localhost:3000")
);

```