

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Port of Entry | Command Center</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <link href="https://fonts.googleapis.com/css2?family=Space+Grotesk:wght@300;400;500;600;700&family=Inter:wght@300;400;500;600&display=swap" rel="stylesheet">
<style>
  /* Chosen Palette: Warm Neutrals & Calm Harmony */
  /* Background: #F9F8F6 (Warm Off-White) */
  /* Surface: #FFFFFF (White) */
  /* Primary Text: #2D2D2A (Soft Black) */
  /* Accent: #C4A484 (Light Brown/Sand) */
  /* Status Green: #8FBC8F (Dark Sea Green) */
  /* Status Blue: #778899 (Light Slate Gray) */

  body {
    font-family: 'Inter', sans-serif;
    background-color: #F9F8F6;
    color: #2D2D2A;
  }

  h1, h2, h3, h4, .font-display {
    font-family: 'Space Grotesk', sans-serif;
  }

  .chart-container {
    position: relative;
    width: 100%;
    max-width: 600px;
    margin-left: auto;
    margin-right: auto;
    height: 300px;
    max-height: 400px;
  }

  /* Custom Scrollbar for Logs */
  .log-scroll::-webkit-scrollbar {
    width: 8px;
  }
```

```

.log-scroll::-webkit-scrollbar-track {
  background: #f1f1f1;
}
.log-scroll::-webkit-scrollbar-thumb {
  background: #d1d5db;
  border-radius: 4px;
}
.log-scroll::-webkit-scrollbar-thumb:hover {
  background: #9ca3af;
}

/* Sphincter Animation Classes */
.valve-ring {
  transition: all 0.5s cubic-bezier(0.4, 0, 0.2, 1);
}
.valve-open {
  transform: scale(1.1);
  border-color: #8FBC8F;
  box-shadow: 0 0 20px rgba(143, 188, 143, 0.3);
}
.valve-closed {
  transform: scale(1.0);
  border-color: #D1D5DB;
}
.pulse-active {
  animation: pulse-border 2s infinite;
}

@keyframes pulse-border {
  0% { box-shadow: 0 0 0 rgba(196, 164, 132, 0.4); }
  70% { box-shadow: 0 0 10px rgba(196, 164, 132, 0); }
  100% { box-shadow: 0 0 0 rgba(196, 164, 132, 0); }
}

.stat-card {
  transition: transform 0.2s ease;
}
.stat-card:hover {
  transform: translateY(-2px);
}

```

</style>

<!--

Application Structure Plan:

1. Header: System Identity, Global Status, and "Commander" welcome.

2. Monitoring Station (Dashboard): Top-level metrics visualizing the flow from External -> Sphincter -> Internal.

3. Operation Deck:

- Left Column: The "Flow" Control. A vertical visualization of the pipeline allowing users to move mock docs between stages.

- Center Column: The Sphincter Valve. The core interaction point. A visual representation of the gate logic.

- Right Column: System Logs (Chain of Custody). Real-time feedback mimicking the bash script outputs.

4. Data Analysis: Charts showing passage history and badge distribution.

5. Codex (Knowledge Base): Tabbed access to the source documentation (Concept, Guide, Architecture).

Why this structure?

It mimics the "Control Panel" concept described in the files but translates the CLI experience into a visual flow.

The "Sphincter" is placed centrally as it is the defining metaphor of the system.

-->

<!--

Visualization & Content Choices:

1. System Traffic (Line Chart): Goal -> Show activity over time. JS/Chart.js used (No SVG).

2. Badge Distribution (Doughnut Chart): Goal -> Show quality metrics of processed docs. JS/Chart.js used.

3. Valve Visual (CSS/HTML): Goal -> Represent the "Sphincter" state (Open/Closed). CSS borders/transitions used (No SVG).

4. Log Console (Text Area): Goal -> Replicate the 'sphincter\_pass.sh' output for familiarity.

-->

<!-- CONFIRMATION: NO SVG graphics used. NO Mermaid JS used. -->

</head>

<body class="min-h-screen flex flex-col">

<!-- HEADER & STATUS -->

<header class="bg-white border-b border-gray-200 shadow-sm sticky top-0 z-50">

<div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 h-16 flex items-center justify-between">

<div class="flex items-center space-x-3">

<span class="text-3xl">☰</span>

<div>

<h1 class="text-xl font-bold tracking-tight text-gray-900">PORT OF ENTRY <span class="text-gray-400 font-light">| CONTROL PANEL</span></h1>

<div class="text-xs text-gray-500 font-mono">v2.0 MODULAR ARCHITECTURE • SYSTEM ONLINE</div>

</div>

```

</div>
<div class="flex items-center space-x-6">
  <div class="flex items-center space-x-2">
    <div class="h-2 w-2 rounded-full bg-green-500 animate-pulse"></div>
    <span class="text-sm font-medium text-gray-600">Secure</span>
  </div>
  <div class="text-sm text-gray-500 font-mono" id="clock">00:00:00</div>
  <button onclick="toggleHelp()" class="text-gray-400 hover:text-gray-600 transition-colors font-medium text-sm">
    COMMANDER HELP [H]
  </button>
</div>
</div>
</header>

<main class="flex-grow max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-8 w-full space-y-8">

  <!-- INTRO & CONTEXT -->
  <section class="bg-white rounded-xl p-6 shadow-sm border border-gray-100">
    <h2 class="text-lg font-bold text-gray-800 mb-2 font-display">Mission Overview</h2>
    <p class="text-gray-600 leading-relaxed">
      Welcome to the command chair, Commander. This panel manages the boundary between the <strong>Universe (External)</strong> and the <strong>Project (Internal)</strong>.
    <p class="text-gray-600 leading-relaxed">
      Your mission is to maintain the <span class="font-mono text-sm bg-gray-100 px-1 rounded">Chain of Custody</span> using the Sphincter Protocol.
    <p class="text-gray-600 leading-relaxed">
      Documents must be screened, badged, and routed through the biological valve metaphor to ensure a clean workspace.
    <p class="text-gray-600 leading-relaxed">
      Use the controls below to simulate document passage operations.
    </p>
  </section>

  <!-- MONITORING STATION (KPIs) -->
  <section>
    <div class="flex items-center justify-between mb-4">
      <h3 class="text-lg font-semibold text-gray-700 flex items-center gap-2">
        <span> MONITORING STATION
      </h3>
      <span class="text-xs text-gray-400 font-mono">LIVE FEED</span>
    </div>
    <div class="grid grid-cols-1 md:grid-cols-4 gap-4">
      <!-- KPI 1 -->
      <div class="bg-white p-5 rounded-lg border border-gray-100 shadow-sm stat-card">
        <div class="text-xs font-bold text-gray-400 uppercase tracking-wider mb-1">Incoming Queue</div>
      </div>
    </div>
  </section>

```

```
<div class="flex items-end justify-between">
    <span class="text-3xl font-bold text-gray-800 font-display"
id="stat-incoming">12</span>
    <span class="text-xs text-yellow-600 bg-yellow-50 px-2 py-1
rounded-full">Pending</span>
</div>
</div>
<!-- KPI 2 -->
<div class="bg-white p-5 rounded-lg border border-gray-100 shadow-sm stat-card">
    <div class="text-xs font-bold text-gray-400 uppercase tracking-wider
mb-1">Review/Holding</div>
    <div class="flex items-end justify-between">
        <span class="text-3xl font-bold text-gray-800 font-display"
id="stat-review">5</span>
        <span class="text-xs text-blue-600 bg-blue-50 px-2 py-1
rounded-full">Screening</span>
    </div>
</div>
<!-- KPI 3 -->
<div class="bg-white p-5 rounded-lg border border-gray-100 shadow-sm stat-card">
    <div class="text-xs font-bold text-gray-400 uppercase tracking-wider
mb-1">Sphincter Passages</div>
    <div class="flex items-end justify-between">
        <span class="text-3xl font-bold text-gray-800 font-display"
id="stat-passages">128</span>
        <span class="text-xs text-green-600 bg-green-50 px-2 py-1
rounded-full">Total</span>
    </div>
</div>
<!-- KPI 4 -->
<div class="bg-white p-5 rounded-lg border border-gray-100 shadow-sm stat-card">
    <div class="text-xs font-bold text-gray-400 uppercase tracking-wider mb-1">Internal
Secure</div>
    <div class="flex items-end justify-between">
        <span class="text-3xl font-bold text-gray-800 font-display"
id="stat-internal">842</span>
        <span class="text-xs text-gray-600 bg-gray-100 px-2 py-1
rounded-full">Archived</span>
    </div>
</div>
</div>
</section>

<!-- OPERATIONS DECK (Main Interaction) -->
```

```
<div class="grid grid-cols-1 lg:grid-cols-12 gap-6">

    <!-- LEFT: PIPELINE CONTROL -->
    <div class="lg:col-span-3 space-y-4">
        <div class="bg-white rounded-xl shadow-sm border border-gray-200 overflow-hidden h-full flex flex-col">
            <div class="p-4 border-b border-gray-100 bg-gray-50">
                <h4 class="font-bold text-gray-700 text-sm uppercase">Pipeline Control</h4>
            </div>
            <div class="p-4 flex-grow space-y-6 flex flex-col justify-center">
                <!-- Step 1 -->
                <div class="relative pl-6 border-l-2 border-gray-200 hover:border-yellow-400 transition-colors">
                    <div class="absolute -left-[9px] top-0 w-4 h-4 rounded-full bg-yellow-100 border-2 border-yellow-400"></div>
                    <h5 class="font-bold text-sm text-gray-800">1. External Port</h5>
                    <p class="text-xs text-gray-500 mb-2">Google Drive / Downloads</p>
                    <button onclick="addIncoming()" class="w-full text-left px-3 py-2 text-xs font-medium text-gray-700 bg-gray-50 hover:bg-yellow-50 rounded border border-gray-200 transition-colors">
                        + Simulate Arrival (GDrive)
                    </button>
                </div>

                <!-- Step 2 -->
                <div class="relative pl-6 border-l-2 border-gray-200 hover:border-blue-400 transition-colors">
                    <div class="absolute -left-[9px] top-0 w-4 h-4 rounded-full bg-blue-100 border-2 border-blue-400"></div>
                    <h5 class="font-bold text-sm text-gray-800">2. Review Queue</h5>
                    <p class="text-xs text-gray-500 mb-2">Quarantine & Organize</p>
                    <button onclick="moveToApproved()" class="w-full text-left px-3 py-2 text-xs font-medium text-gray-700 bg-gray-50 hover:bg-blue-50 rounded border border-gray-200 transition-colors">
                        ✓ Approve Next Doc
                    </button>
                </div>

                <!-- Step 3 -->
                <div class="relative pl-6 border-l-2 border-gray-200 hover:border-green-400 transition-colors">
                    <div class="absolute -left-[9px] top-0 w-4 h-4 rounded-full bg-green-100 border-2 border-green-400"></div>
                    <h5 class="font-bold text-sm text-gray-800">3. Sphincter Gate</h5>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

<p class="text-xs text-gray-500 mb-2">The Valve Mechanism</p>
<div class="text-xs text-gray-400 italic">Use center console to activate</div>
</div>

<!-- Step 4 -->
<div class="relative pl-6 border-l-2 border-gray-200 hover:border-gray-400 transition-colors">
    <div class="absolute -left-[9px] top-0 w-4 h-4 rounded-full bg-gray-100 border-2 border-gray-400"></div>
        <h5 class="font-bold text-sm text-gray-800">4. Internal Routing</h5>
        <p class="text-xs text-gray-500 mb-2">Final Destination</p>
        <button onclick="routeDocuments()" class="w-full text-left px-3 py-2 text-xs font-medium text-gray-700 bg-gray-50 hover:bg-gray-100 rounded border border-gray-200 transition-colors">
            → Route to Codex
        </button>
    </div>
</div>
</div>
</div>

<!-- CENTER: THE SPHINCTER (Valve) -->
<div class="lg:col-span-5">
    <div class="bg-white rounded-xl shadow-md border border-gray-200 overflow-hidden h-full flex flex-col relative">
        <div class="p-4 border-b border-gray-100 flex justify-between items-center">
            <h4 class="font-bold text-gray-700 text-sm uppercase">Sphincter Mechanism</h4>
            <div class="flex space-x-1">
                <span class="w-2 h-2 rounded-full bg-gray-300" id="valve-light-1"></span>
                <span class="w-2 h-2 rounded-full bg-gray-300" id="valve-light-2"></span>
                <span class="w-2 h-2 rounded-full bg-gray-300" id="valve-light-3"></span>
            </div>
        </div>
        <!-- Visualization Area -->
        <div class="flex-grow flex flex-col items-center justify-center p-8 bg-gray-50 relative">
            <!-- The Valve Visual -->
            <div id="valve-outer" class="w-64 h-64 rounded-full border-[16px] border-gray-200 flex items-center justify-center bg-white shadow-inner valve-ring relative">
                <!-- Inner Ring -->

```

```
<div id="valve-inner" class="w-48 h-48 rounded-full border-[8px]
border-dashed border-gray-300 flex items-center justify-center transition-all duration-700 relative
overflow-hidden">
    <div class="absolute inset-0 bg-gray-50 opacity-50"></div>
    <!-- Core Status -->
    <div class="z-10 text-center">
        <div class="text-4xl mb-1 transition-transform duration-300"
id="valve-icon">🔒 </div>
        <div class="text-xs font-mono text-gray-400"
id="valve-status">CLOSED</div>
    </div>
    <!-- Passing particles (hidden by default) -->
    <div id="particle-stream" class="absolute inset-0 hidden">
        <div class="absolute top-1/2 left-1/2 w-2 h-2 bg-green-500 rounded-full
transform -translate-x-1/2 -translate-y-1/2 animate-ping"></div>
    </div>
    </div>
</div>

<!-- Action Button -->
<div class="mt-8 w-full max-w-xs">
    <button id="activate-btn" onclick="activateSphincter()" disabled class="w-full
py-3 px-6 bg-gray-200 text-gray-400 font-bold rounded-lg shadow-sm transition-all transform
active:scale-95 flex items-center justify-center gap-2 cursor-not-allowed">
        <span>⚡ </span> INITIATE PASSAGE
    </button>
    <p class="text-center text-xs text-gray-400 mt-2 font-mono"
id="action-hint">Waiting for approved document...</p>
    </div>
</div>
</div>
</div>

<!-- RIGHT: SYSTEM LOGS -->
<div class="lg:col-span-4">
    <div class="bg-[#1e1e1e] rounded-xl shadow-lg border border-gray-700
overflow-hidden h-[500px] flex flex-col font-mono text-xs">
        <div class="p-3 bg-[#252526] border-b border-[#333] flex justify-between
items-center">
            <span class="text-gray-300">TERMINAL OUTPUT</span>
            <div class="flex space-x-2">
                <div class="w-2 h-2 rounded-full bg-red-500"></div>
                <div class="w-2 h-2 rounded-full bg-yellow-500"></div>
                <div class="w-2 h-2 rounded-full bg-green-500"></div>
            </div>
        </div>
    </div>
</div>
```

```

        </div>
    </div>
    <div id="log-console" class="flex-grow p-4 overflow-y-auto text-gray-300 space-y-1
log-scroll">
        <div class="text-green-500">System initialized...</div>
        <div class="text-gray-500">Loading module: monitoring_station.sh</div>
        <div class="text-gray-500">Loading module: sphincter_pass.sh</div>
        <div class="text-blue-400">Ready for command input.</div>
        <br>
        <div>Waiting for events...</div>
    </div>
</div>
</div>

<!-- ANALYTICS SECTION -->
<section class="grid grid-cols-1 md:grid-cols-2 gap-6">
    <div class="bg-white p-6 rounded-xl border border-gray-100 shadow-sm">
        <div class="mb-4">
            <h3 class="font-bold text-gray-700">Passage Velocity</h3>
            <p class="text-xs text-gray-400">Daily processing volume</p>
        </div>
        <!-- Chart Container Rule: Wrapped in styled div -->
        <div class="chart-container">
            <canvas id="velocityChart"></canvas>
        </div>
    </div>
    <div class="bg-white p-6 rounded-xl border border-gray-100 shadow-sm">
        <div class="mb-4">
            <h3 class="font-bold text-gray-700">Badge Distribution</h3>
            <p class="text-xs text-gray-400">Quality markers assigned</p>
        </div>
        <!-- Chart Container Rule: Wrapped in styled div -->
        <div class="chart-container">
            <canvas id="badgeChart"></canvas>
        </div>
    </div>
</section>

<!-- KNOWLEDGE BASE -->
<section class="bg-white rounded-xl shadow-sm border border-gray-200 overflow-hidden">
    <div class="border-b border-gray-200 bg-gray-50 px-6 py-4">
        <h3 class="font-bold text-gray-800"> SYSTEM ARCHITECTURE &
DOCUMENTATION</h3>

```

```
</div>
<div class="p-6">
  <div class="grid grid-cols-1 md:grid-cols-3 gap-8">
    <!-- Column 1: Concept -->
    <div>
      <h4 class="font-bold text-gray-900 mb-3 flex items-center">
        <span class="bg-blue-100 text-blue-800 text-xs px-2 py-1 rounded
mr-2">CONCEPT</span>
          The Sphincter Metaphor
        </h4>
      <p class="text-sm text-gray-600 mb-4">
        Based on the <code class="bg-gray-100 px-1
rounded">SPHINCTER_CONCEPT.md</code>, the Port of Entry functions as a biological valve.
        It embodies <strong>intentional choice</strong>. Each passage is a decision
point preventing random, unconscious content flow.
      </p>
      <div class="text-xs font-mono bg-gray-50 p-3 rounded border border-gray-200
text-gray-500">
        EXTERNAL CHAOS<br>
        &nbsp;&nbsp;&nbsp;&nbsp;| <br>
        &nbsp;&nbsp;&nbsp;&nbsp;▼<br>
        SPHINCTER (Valve)<br>
        &nbsp;&nbsp;&nbsp;&nbsp;| <br>
        &nbsp;&nbsp;&nbsp;&nbsp;▼<br>
        CLEAN PROJECT
      </div>
    </div>
  </div>

  <!-- Column 2: Architecture -->
  <div>
    <h4 class="font-bold text-gray-900 mb-3 flex items-center">
      <span class="bg-purple-100 text-purple-800 text-xs px-2 py-1 rounded
mr-2">TECH</span>
        Modular Architecture
      </h4>
    <p class="text-sm text-gray-600 mb-4">
      The system is built on isolated modules (v2.0), similar to airport security
stations.
    </p>
    <ul class="text-sm text-gray-600 space-y-2 list-disc list-inside">
      <li><strong>Monitoring Station:</strong> Real-time dashboard.</li>
      <li><strong>Inspection Station:</strong> Quality gates.</li>
      <li><strong>Routing Center:</strong> Destination logic.</li>
      <li><strong>Registry:</strong> Permanent logs.</li>
    </ul>
  </div>
</div>
```



```

internal: 842,
queueForPassage: [], // Docs ready for sphincter
lastActionTime: new Date()
};

const mockDocs = [
  "research_notes_2025.pdf",
  "entity_alignment_matrix.docx",
  "sector_7_blueprints.json",
  "field_report_alpha.txt",
  "observation_log_v2.md"
];

// Chart Instances
let velocityChartInst = null;
let badgeChartInst = null;

// --- DOM ELEMENTS ---
const elIncoming = document.getElementById('stat-incoming');
const elReview = document.getElementById('stat-review');
const elPassages = document.getElementById('stat-passages');
const elInternal = document.getElementById('stat-internal');
const elValveOuter = document.getElementById('valve-outer');
const elValveInner = document.getElementById('valve-inner');
const elValveIcon = document.getElementById('valve-icon');
const elValveStatus = document.getElementById('valve-status');
const elActivateBtn = document.getElementById('activate-btn');
const elActionHint = document.getElementById('action-hint');
const elLog = document.getElementById('log-console');
const elValveLights = [
  document.getElementById('valve-light-1'),
  document.getElementById('valve-light-2'),
  document.getElementById('valve-light-3')
];
}

// --- CORE FUNCTIONS ---

function init() {
  updateUI();
  initCharts();
  setInterval(updateClock, 1000);
  log("System Ready. Waiting for commands...", "text-green-500");
}

```

```

function updateClock() {
    const now = new Date();
    document.getElementById('clock').innerText = now.toISOString().split('T')[1].split('.')[0] + "
UTC";
}

function updateUI() {
    // Update stats
    elIncoming.innerText = state.incoming;
    elReview.innerText = state.review;
    elPassages.innerText = state.passages;
    elInternal.innerText = state.internal;

    // Update Valve Button State
    if (state.approved > 0) {
        elActivateBtn.disabled = false;
        elActivateBtn.classList.remove('bg-gray-200', 'text-gray-400', 'cursor-not-allowed');
        elActivateBtn.classList.add('bg-gray-800', 'text-white', 'hover:bg-gray-900',
'cursor-pointer');
        elActionHint.innerText = `${state.approved} document(s) approved and ready for
passage.`;
        elActionHint.classList.add('text-green-600');
        elValveOuter.classList.add('pulse-active'); // Visual cue
    } else {
        elActivateBtn.disabled = true;
        elActivateBtn.classList.add('bg-gray-200', 'text-gray-400', 'cursor-not-allowed');
        elActivateBtn.classList.remove('bg-gray-800', 'text-white', 'hover:bg-gray-900',
'cursor-pointer');
        elActionHint.innerText = "No approved documents in queue.";
        elActionHint.classList.remove('text-green-600');
        elValveOuter.classList.remove('pulse-active');
    }
}

function log(message, colorClass = "text-gray-300") {
    const timestamp = new Date().toISOString().split('T')[1].split('.')[0];
    const div = document.createElement('div');
    div.className = `${colorClass} font-mono text-xs border-l-2 border-transparent pl-2
hover:border-gray-600 transition-colors`;
    div.innerHTML = `[${timestamp}] ${message}`;
    elLog.appendChild(div);
    elLog.scrollTop = elLog.scrollHeight;
}

```

```

// --- ACTIONS ---

function addIncoming() {
    state.incoming++;
    const docName = mockDocs[Math.floor(Math.random() * mockDocs.length)];
    log(`EXTERNAL PORT: New signal detected. File: ${docName}`, "text-yellow-400");
    updateUI();
}

function moveToApproved() {
    if (state.incoming > 0) {
        state.incoming--;
        // Skip review manually for speed in simulation, assume moved to review then
        approved
        state.review++;
        setTimeout(() => {
            state.review--;
            state.approved++;
            log(`INSPECTION: Document cleared for passage. Quality: High. Badges
assigned: ★`, "text-blue-400");
            updateUI();
        }, 500); // Small delay to simulate review
        updateUI();
    } else {
        log("ERROR: No incoming documents to process.", "text-red-500");
    }
}

function activateSphincter() {
    if (state.approved <= 0) return;

    // 1. Lock Interface
    elActivateBtn.disabled = true;
    elActivateBtn.innerHTML = "<span>⚙</span> PROCESSING...";

    // 2. Generate ID
    const docID = "LOCAL-" + Date.now().toString().slice(-6);
    log(`SPHINCTER: Protocol Initiated. ID: ${docID}`, "text-green-400");

    // 3. Animation Sequence
    // Stage 1: Pressurization
    elValveStatus.innerText = "PRESSURIZING";
    elValveIcon.innerText = "⏳";
    elValveLights[0].classList.replace('bg-gray-300', 'bg-yellow-400');
}

```

```

setTimeout(() => {
    // Stage 2: Open
    elValveOuter.classList.add('valve-open');
    elValveInner.classList.add('border-green-400');
    elValveInner.classList.remove('border-gray-300');
    elValveStatus.innerText = "PASSAGE ACTIVE";
    elValveIcon.innerText = "🔒";
    elValveLights[1].classList.replace('bg-gray-300', 'bg-yellow-400');

    log(`Passage open. Transferring payload...`, "text-gray-300");

    setTimeout(() => {
        // Stage 3: Close & Complete
        elValveOuter.classList.remove('valve-open');
        elValveInner.classList.remove('border-green-400');
        elValveInner.classList.add('border-gray-300');
        elValveStatus.innerText = "CLOSED";
        elValveIcon.innerText = "🔓";
        elValveLights[2].classList.replace('bg-gray-300', 'bg-green-500');

        // Reset Lights
        setTimeout(() => {
            elValveLights.forEach(l => {
                l.classList.remove('bg-yellow-400', 'bg-green-500');
                l.classList.add('bg-gray-300');
            });
        }, 1000);

        // Update Data
        state.approved--;
        state.passages++;
        state.internal++; // Temporarily in "ready for routing" but counted as inside

        // Log Success (Format from files)
        log(`SUCCESS: Document ${docID} entered internal hub.`, "text-green-500");
        log(`CUSTODY: Chain of Custody record created.`, "text-gray-500");

        // Reset Button
        elActivateBtn.innerHTML = "<span>⚡</span> INITIATE PASSAGE";
        updateUI();
        updateCharts(); // Refresh charts with new data

    }, 1500); // Open duration
}

```

```

        }, 800); // Pressurize duration
    }

function routeDocuments() {
    // Mock routing function
    log(`ROUTER: Scanning internal hub...`, "text-blue-300");
    setTimeout(() => {
        log(`ROUTER: 1 file routed to /ENTITY/PRIME/docs/codex`, "text-blue-300");
    }, 800);
}

function toggleHelp() {
    alert("COMMANDER HELP:\n\n1. Use 'Simulate Arrival' to populate Incoming Queue.\n2. Use 'Approve' to move docs to the Sphincter Queue.\n3. Click 'Initiate Passage' to run the Valve Protocol.\n\nSee System Logs for Chain of Custody details.");
}

// --- CHARTS ---

function initCharts() {
    // Chart 1: Velocity (Line)
    const ctxVel = document.getElementById('velocityChart').getContext('2d');
    velocityChartInst = new Chart(ctxVel, {
        type: 'line',
        data: {
            labels: ['08:00', '10:00', '12:00', '14:00', '16:00', 'Now'],
            datasets: [{

                label: 'Documents Processed',
                data: [12, 19, 15, 25, 22, state.passages], // Start with mock history
                borderColor: '#8FBC8F',
                backgroundColor: 'rgba(143, 188, 143, 0.2)',
                tension: 0.4,
                fill: true
            }]
        },
        options: {
            responsive: true,
            maintainAspectRatio: false,
            plugins: {
                legend: { display: false }
            },
            scales: {
                y: { beginAtZero: true, grid: { display: false } },

```

```

        x: { grid: { display: false } }
    }
}
});

// Chart 2: Badges (Doughnut)
const ctxBadge = document.getElementById('badgeChart').getContext('2d');
badgeChartInst = new Chart(ctxBadge, {
    type: 'doughnut',
    data: {
        labels: ['Gold Star', 'Secure', 'Priority', 'Standard'],
        datasets: [{{
            data: [15, 45, 20, 20],
            backgroundColor: [
                '#FFD700', // Gold
                '#8FBC8F', // Green
                '#CD5C5C', // Red/Terra
                '#D3D3D3' // Grey
            ],
            borderWidth: 0
        }}]
    },
    options: {
        responsive: true,
        maintainAspectRatio: false,
        plugins: {
            legend: { position: 'right' }
        }
    }
});
});

function updateCharts() {
    if(velocityChartInst) {
        // Update last point of velocity chart
        const data = velocityChartInst.data.datasets[0].data;
        data[data.length - 1] = state.passages;
        velocityChartInst.update();
    }
}

// Start
window.addEventListener('load', init);

```

```
</script>
</body>
</html>
```