Modern Intranet Implementation Plan

1. Project Overview

Goal:
Deliver a modern, cloud-based intranet that supports staff communication, self-service, and collaboration using Microsoft 365 services, optimised for Windows 11 and Single Sign-On (SSO).

Objectives:

- Provide a single, trusted source of internal information
- Improve discoverability of policies, procedures, and resources
- Integrate seamlessly with Microsoft Teams and SharePoint
- Support future extensibility without re-platforming

Audience: All internal staff, segmented by role, department, and function
Ownership: IT (platform) + Internal Communications (content) + Business sponsors

---

2. Guiding Principles

- Cloud-first, Microsoft-first
- Security and compliance by default
- Simple information architecture
- Mobile and accessibility friendly
- Low-code over custom code where possible

---

3. Technology Stack

Hosting & Execution Model (Recommended)

The intranet will be **hosted natively in SharePoint Online**, using the modern SharePoint experience as the primary web hosting platform.

This means:

- Microsoft-hosted infrastructure (no on-prem or IaaS web servers)
- Built-in scalability, availability, and patching
- Native integration with Microsoft Entra ID for Single Sign-On
- No traditional server-side code execution within SharePoint

Design Position:
SharePoint Online is treated as an **experience and presentation layer**, not a general-purpose application server. Business logic and compute are handled via approved platform services (Power Platform and Azure) when required.

Development & UX Stack (Recommended)

To deliver a **clean, modern, dynamic UX** (sidebar, navbar, status bar, function cards, dashboards), the recommended code layer is:

- **SharePoint Framework (SPFx)** for custom intranet components
- **TypeScript + React** for UI development (VS Code)
- **Fluent UI** (Microsoft design system) for consistent, accessible components
- **PnPjs** for SharePoint/Graph data access ergonomics

Delivery Model

- SharePoint pages provide overall structure and navigation
- SPFx web parts provide interactive tools and rich experiences
- Power Platform used for rapid process apps (forms/approvals) where appropriate

Build & Release (Suggested)

- Source control: GitHub (aligned with DDRE Intranet workflow)
- CI/CD: GitHub Actions to package and deploy SPFx solutions
- Deployment: SharePoint App Catalog
- Environments: Dev / Test / Production tenants or environment separation via app catalog + site collections

---

Core Platform

- **Microsoft SharePoint Online (Modern Experience)**
  Primary intranet platform (Home site, hub sites, communication sites)

- **Microsoft Entra ID**
  Identity, authentication, and Single Sign-On
- **Microsoft Teams**
  Collaboration layer and intranet entry point
- **Viva Connections / Viva Engage**
  Employee engagement and intranet surfacing inside Teams

Automation & Apps

- **Power Automate** – workflows, approvals, notifications
- **Power Apps** – lightweight internal tools and forms

Reporting & Insights

- **Power BI** – usage and adoption reporting
- **Microsoft 365 Analytics** – platform-level insights

Security & Compliance

- **Microsoft Purview** – data governance, retention, DLP
- **Conditional Access & MFA** – enforced via Entra ID

---

4. Identity & Access Management

Identity Model

- Microsoft Entra ID is the authoritative identity provider
- All intranet access is authenticated via Entra ID (SSO)
- SharePoint, Teams, Viva, and Power Platform share a common identity context

Authentication

- Seamless SSO across SharePoint Online, Teams, and Power Platform
- No separate intranet credentials required

Access Control

- Role-based access to sites and content
- Use Microsoft 365 groups and security groups
- Permissions inherited wherever possible to reduce complexity

Security Controls

- Mandatory MFA
- Conditional access policies (location, device, risk)
- Guest access governed, time-bound, and audited

---

5. Information Architecture

Site Structure

- **Intranet Home Site** (global landing page)
- **Hub Sites** for departments and functions
- **Communication Sites** for publishing and reference content

Navigation

- Global navigation bar (home site)
- Consistent hub navigation across sites

Metadata & Search

- Defined taxonomy for content classification
- Managed metadata for documents and pages
- Promoted search results for key resources

---

6. Governance Model

Roles, Responsibilities & Access Control Model

The intranet uses **Microsoft Entra ID + SharePoint permissions** to enforce role-based access control (RBAC) across sites, content, and tools. Access is granted to **groups, not individuals**, wherever possible.

CORE ACCESS ROLES

Role	Typical Permissions	Notes
Visitor / Reader	Read-only	Default for most staff; can view pages, documents, dashboards
Contributor / Author	Create & edit pages/items	Departmental content authors
Tool User	Read/write to specific lists/apps	Scoped to specific tools (e.g. PM Dashboard)
Tool Admin	Full control of tool data	Usually senior staff or system owners
Site Owner	Full control of site	Limited and tightly governed
Platform Admin	Tenant-level admin	IT only

GROUP STRATEGY

- Use **Microsoft 365 Groups** for:
  - Departmental access
  - Tool-specific access (e.g. `PM-Dashboard-Users`, `PM-Dashboard-Admins`)
- Use **Security Groups** where mailboxes/Teams are not required
- Avoid direct user permissions on sites, lists, or libraries

PERMISSION SCOPING

- **Site-level permissions** control broad access (read vs edit)
- **Hub-level permissions** align to departments
- **List/library-level permissions** used for tools and sensitive data
- **Item-level permissions** used sparingly (exception cases only)

Department-Based Access Model (Initial)

The intranet will use a **department-first access model**, layered on top of the core access roles. This model is intentionally simple and can be refined over time.

DEPARTMENTS IN SCOPE

- **Administration**
- **Property Management**
- **Sales**
- **Office (General Staff)**

STANDARD DEPARTMENT GROUPS (INITIAL PROPOSAL)

Each department has two primary groups:

- `<Dept>-Readers` – read-only access
- `<Dept>-Contributors` – content/tool editing where appropriate

Examples:

- `Administration-Readers`, `Administration-Contributors`
- `PropertyManagement-Readers`, `PropertyManagement-Contributors`
- `Sales-Readers`, `Sales-Contributors`
- `Office-Readers`

HIGH-LEVEL ACCESS MATRIX (V1)

Site / Area	Administration	Property Management	Sales	Office
Intranet Home	Read	Read	Read	Read
Company Policies	Edit	Read	Read	Read
Admin Hub	Edit	Read	Read	None
PM Hub	Read	Edit	Read	None
Sales Hub	Read	Read	Edit	None
PM Dashboard Tool	Read	Edit	Read	None
Marketing Budget Tool	Read	Read	Edit	None

> *Note: "None" indicates the area is hidden via permissions and/or audience targeting.*

DESIGN NOTES

- Everyone can see the **Intranet Home** (baseline transparency)
- Department hubs are writable only by their owning department
- Cross-department visibility is read-only by default
- Office group receives access only to general content unless explicitly granted

EVOLUTION STRATEGY

- Add tool-specific groups as complexity grows (e.g. `PM-Dashboard-Admins`)
- Introduce sub-department or role-based groups later if required
- Review and adjust the matrix quarterly based on usage and feedback

---

Access Requests & Group Management (Administrator Experience)

The intranet will provide an **Admin-facing access management page** to support day-to-day changes (new starters, role changes, cross-department access) without ad-hoc permission changes on individual sites.

IMPORTANT PLATFORM NOTE

- SharePoint Online does **not** provide a fully-featured, built-in “manage all groups” UI within the intranet itself.
- Source-of-truth management for membership remains:
  - **Microsoft Entra ID / Microsoft 365 Admin Centre** for groups
  - **SharePoint site permissions** for sites/libraries/lists

The intranet Admin page is therefore treated as a **controlled front-end** for *requesting* and *approving* access, and (where permitted) automating group membership updates.

RECOMMENDED PATTERN (SELF-SERVICE, GOVERNED)

1. **Access Management Page (SharePoint + SPFx)**
   - Displays:
     - Department groups and tool groups (read-only list)
     - Who the group owners are
     - User’s current access (where feasible)
   - Provides actions:
     - “Request Access” (select group/tool, justification)
     - “Request Removal”
2. **Approval Workflow (Power Automate)**

- Requests route to the correct approver:
  - Group Owner (department/tool owner)
  - Optional IT approval for high-privilege groups
- Approved requests:
  - Automatically add/remove user from the relevant group (preferred)
  - Or generate an IT service ticket if policy requires manual changes

3. **Audit & Reporting**
   - All requests stored in a SharePoint List (audit trail)
   - Quarterly review of:
     - High-privilege groups
     - Cross-department exceptions

OWNERSHIP & SAFEGUARDS

- Each group must have:
  - At least **two owners** (to avoid access bottlenecks)
  - A documented purpose and scope
- “Admin/Owner” roles are limited and periodically reviewed
- Avoid direct user permissions; group membership is the only supported exception path

---

Intranet Administration & Permissions Management

To support day-to-day access changes and controlled crossover between departments, the intranet will include a **dedicated Administration area** for authorised administrators.

Purpose

- Provide visibility of intranet groups and their intent
- Enable controlled management of group membership
- Reduce reliance on IT for routine access changes
- Maintain auditability and governance

Implementation Model (Recommended)

- An **Administration Hub** accessible only to authorised Admin roles
- An **SPFx-based Admin Page** that:
  - Displays department groups and tool-specific groups
  - Shows group purpose and access scope (read/write/admin)
  - Links directly to Microsoft Entra ID / M365 group management
  - Optionally surfaces membership lists (read-only or delegated edit)

> *Note: The Admin page does not replace Entra ID or SharePoint security. It acts as a governed front-end to approved group operations.*

Who Can Use This Area

- Platform Admins (IT)
- Delegated Intranet Administrators (trusted business users)

What Can Be Managed

- Add/remove users to existing approved groups
- Assign users to multiple department or tool groups (crossover support)
- View effective access for a given user (where supported)

What Is Explicitly Out of Scope

- Creating new security groups without governance approval
- Changing permission inheritance on sites/lists
- Granting direct user permissions

Governance Standards

- Least-privilege by default
- Groups before individuals
- Clear ownership for every site, list, and tool
- Permissions reviewed quarterly
- No broken inheritance without documented justification
- Naming conventions
- Page templates and layouts
- Branding and accessibility guidelines

Lifecycle Management

- Site provisioning process
- Quarterly content review
- Archival and retention policies

---

7. Content & User Experience Strategy

Content Types

- News and announcements
- Policies and procedures
- Forms and self-service tools
- Onboarding and training materials

UX Principles

- Audience-targeted content
- Minimal clicks to key resources
- Consistent layouts and visual cues

Prototyping

- Early wireframes for home and hub sites
- Pilot content with selected departments

---

8. Integrations

Departmental Tools & Utilities (Dashboards, Calculators, Apps)

The intranet will include departmental tools such as a **PM Dashboard** (currently spreadsheet-based) and a **Sales Marketing Budget Calculator** (currently HTML/CSS/JS and using SQLite), plus additional tools as the intranet evolves.

Recommended Implementation Pattern

- Build each tool as an **SPFx (React) web part** for a consistent UX and governance-friendly deployment.
- Reuse existing logic where possible:
  - Existing HTML/CSS/JS tools are migrated into SPFx by:
    - Converting the UI into React components, and/or

- Reusing existing JS calculation logic inside the SPFx web part

Data Approach (by tool type)

- **Spreadsheet-like trackers (e.g., PM Dashboard):**
  - Move data into a governed store such as **SharePoint Lists** (simple, fast, permissions-friendly) or **Dataverse** (if relational complexity/workflows/reporting justify it).
  - Present the UI using Fluent UI tables (sorting/filtering), with optional Power BI reporting.
- **Standalone calculators (e.g., Marketing Budget):**
  - Host as an SPFx tool to avoid unsupported script embedding.
  - If the calculator is truly stateless: no database required.
  - If the calculator needs saving scenarios/presets/history: store those in a SharePoint List.
- **Existing SQLite-backed tools (e.g., Marketing Budget today):**
  - **SharePoint Online cannot run SQLite as a server-side database.** SQLite is file-based and typically embedded in an app runtime.
  - Preferred migration paths:
    1. **Replace SQLite with SharePoint Lists** (common for simple tables/config/presets)
    2. **Replace SQLite with Dataverse** (if relationships, validations, workflows, and auditing are important)
    3. **Keep SQLite only as a client-side cache (advanced/limited):** using a browser-compatible SQLite implementation (e.g., WebAssembly) for offline/local use, with an authoritative store still in SharePoint/Dataverse
    4. **Move the DB to Azure SQL (only if justified):** access via an API layer (Azure Functions/App Service) – SPFx does not connect directly to SQL

UX Standards

- Common layout shell: sidebar + navbar + page header + status bar
- Function cards for tool entry points
- Consistent styling via Fluent UI (and DDRE branding)

Obsidian Vault (Markdown Knowledge Base)

The organisation maintains an existing Obsidian Vault containing Markdown-formatted policies, processes, and guidelines. **This vault already exists as a SharePoint Online document library**, and Obsidian is used as the primary authoring tool.

The intranet must present a **dynamic, read-only view** of this content to staff, while preserving SharePoint as the storage layer and Obsidian as the authoring experience.

Key Requirements

- Single source of truth remains the SharePoint-hosted Obsidian Vault
- No duplication of content between systems
- Markdown rendered dynamically as modern web pages
- Folder structure, tags, and frontmatter leveraged for navigation and search
- Access controlled via SharePoint permissions and Microsoft Entra ID

Recommended Approaches (to validate with IT)

1. **SharePoint-Native Markdown Rendering (Preferred)**
   - Obsidian Vault stored in a dedicated SharePoint document library
   - Custom SharePoint Framework (SPFx) web parts render Markdown directly
   - Pages generated dynamically without content copying
2. **Automated Markdown-to-Page Publishing**
   - Power Automate or Azure-based process converts Markdown to modern SharePoint pages
   - Updates triggered on file change or scheduled sync
   - Read-only published pages linked back to source files
3. **Azure Rendering Service (Fallback / Advanced)**
   - Azure Static Web App or App Service reads Markdown directly from SharePoint
   - Content embedded into the intranet with SSO
   - Considered only if SharePoint-native rendering proves insufficient

Governance Considerations

- Editorial workflow remains entirely within Obsidian
- Draft vs published state managed via folders, metadata, or frontmatter
- Clear ownership for both source content and rendering components

---

Microsoft Teams

- Intranet accessible via Viva Connections
- SharePoint libraries surfaced as Teams tabs

Power Platform

- Automated approvals and notifications
- Forms replacing email-based processes

Future Integrations (Optional)

- HR or CRM systems via APIs or connectors
- Knowledge base or AI-assisted search

---

9. Execution Capabilities & Constraints (SharePoint Online)

Supported Capabilities

- Interactive client-side components using SharePoint Framework (SPFx)
- Rich UI built with JavaScript / TypeScript (e.g. React)
- Secure API calls using the user's Entra ID context
- Dynamic content rendering and personalization
- Integration with Power Apps and Power Automate

Platform Constraints

- No server-side code execution within SharePoint
- No background services or long-running processes
- No direct filesystem or OS-level access
- Browser sandbox and performance constraints apply

Architectural Implications

- SharePoint is used for **navigation, presentation, and security**
- Business logic and automation handled via:
  - Power Platform (preferred)
  - Azure Functions / APIs (when justified)
- Heavy processing and custom backends are explicitly out of scope for SharePoint

This model ensures a secure, supportable intranet without introducing unmanaged infrastructure.

---

10. Security & Compliance

- Least-privilege access model
- Data classification and sensitivity labels
- Audit logging and monitoring
- Compliance with internal policies

---

11. Deployment Approach

Phase 1 – Discovery & Planning

- Stakeholder workshops
- Content audit
- Architecture and governance definition

Phase 2 – Prototype

- Home site and one hub site
- Initial navigation and templates

Phase 3 – Pilot

- Limited user group
- Feedback and refinement

Phase 4 – Full Rollout

- Company-wide launch
- Training and communications

Phase 5 – Continuous Improvement

- Analytics-driven iteration
- Regular roadmap reviews

12. Training & Adoption

- Role-based training sessions
- Quick reference guides
- Champions network
- Ongoing support model

13. Success Measures

Quantitative

- Active users
- Page views and search usage
- Content freshness metrics

Qualitative

- Staff feedback
- Survey results
- Support ticket trends

14. Risks & Mitigations

Risk	Mitigation
Low adoption	Strong launch comms, Teams integration
Content sprawl	Governance and regular reviews
Security gaps	Enforced policies and audits

15. Open Questions for IT Workshop (Jan 2026)

- Identity model and tenant constraints
- Licensing assumptions and gaps
- Network or compliance considerations
- Integration boundaries and API access
- Long-term ownership and support model