BD Portal | Phase One | Development Outline

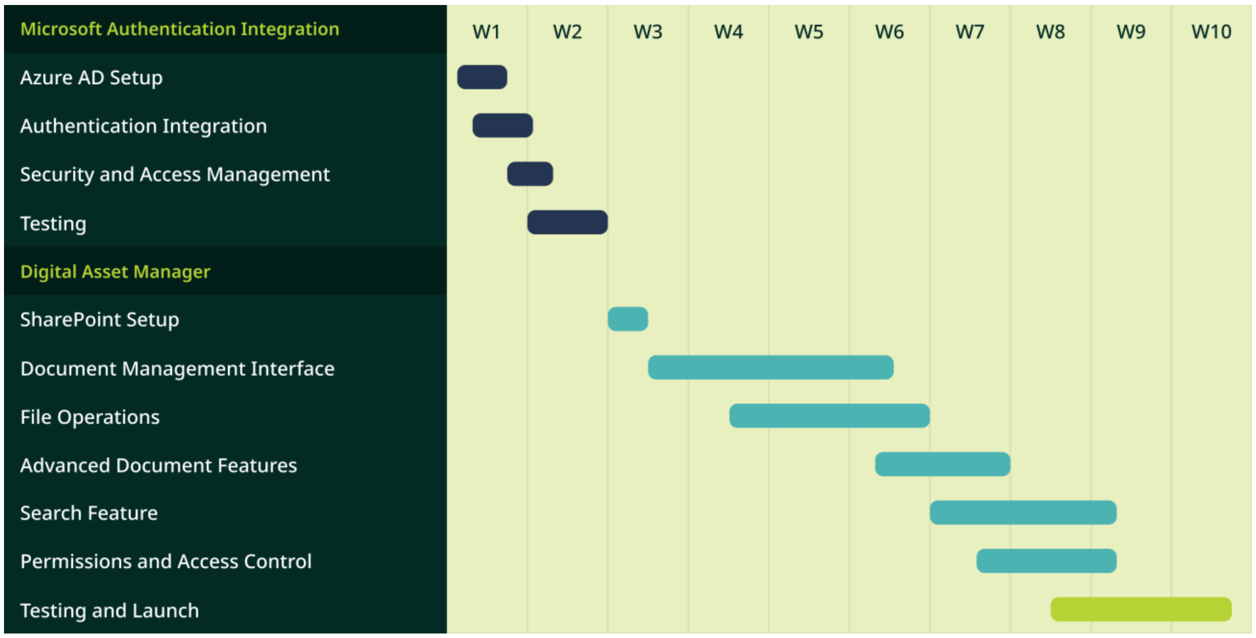

Figure 1: Phase One Timeline

Microsoft Authentication Integration

1.1 Configure Azure App

This step involves setting up the Azure application to establish the app's identity and enable Microsoft authentication. This configuration is crucial for both user and app authentication within the portal.

- Register the application in Azure
- Configure authentication settings
- Create client secret
- Assign API permissions

1.2 Set Up User Permissions in Azure

Configure the Azure app to handle user authentication by setting up delegated permissions. This allows the app to access user-specific data and perform actions on behalf of the user.

- Configure delegated permissions
- Set up user consent settings
- Validate user permission configuration

1.3 Configure Application Permissions in Azure

Set up application-level permissions in Azure to enable the app to perform autonomous tasks in the background, such as checking group memberships without user interaction.

- Assign application permissions
- Configure consent for application permissions
- Securely store client credentials
- Obtain Admin Consent

1.4 Integrate NextAuth.js with Microsoft

Integrate NextAuth.js into the project to manage user authentication with Microsoft accounts. This integration assists with sign-ins, sign-outs, and session management.

- Install and configure NextAuth.js
- Set up Microsoft provider in NextAuth.js
- Configure environment variables

1.5 Implement User Authentication Flows

Implement the core user authentication flows, including login and logout

- Implement login flow
- Implement logout flow

1.6 Implement Application-Level Authentication

Set up the authentication process for the application itself to perform background operations autonomously, using the application permissions configured in Azure.

- Implement credentials flow
- Obtain and manage application tokens
- Test application authentication

1.7 Implement Session Management

Set up session management to store and manage user session data, ensuring that user states are maintained throughout their interaction with the application.

- Implement secure session storage
- Manage session expiry and renewal
- Test session persistence

1.8 Establish Role-Based Access Control

Implement role-based access control by leveraging Azure AD groups to manage permissions and access levels based on user roles. This enhances security and simplifies user management by allowing admins to control access through group memberships.

- Define user roles and map to Azure AD groups
- Implement RBAC logic using Azure AD group memberships
- Test role-based access via group memberships

1.9 Apply Middleware Security

Add middleware to the project to protect pages, ensuring that only authenticated users can access the application.

- Implement middleware for route protection
- Configure middleware for role-based access
- Test route security

1.10 Connect User Profile to UI

Integrate the authenticated user's account details into the application's UI, allowing users to view their profile information.

- Display user profile information in UI
- Test user account integration

1.11 Conduct Authentication Testing

Perform thorough testing of all authentication-related components to ensure secure and reliable user and application authentication flows.

- Test user authentication scenarios
- Test application authentication scenarios
- Validate session and role-based access control

Digital Asset Manager Development

2.1 Configure SharePoint Document Library

Set up a SharePoint document library to store and manage digital assets

- Create document library in SharePoint
- Configure library settings and permissions
- Set up custom metadata fields

2.2 Set Up API Access to Document Library

Implement API access to the SharePoint document library, enabling the application to interact with stored documents programmatically.

- Register API permissions in Azure
- Implement API authentication
- Test API access to document library

2.3 Develop Document Management Interface

Build the user interface for managing documents, providing features for browsing, searching, and interacting with digital assets.

- Develop Collections page
- Create custom folder component
- Develop list and tile views
- Implement sharing and link generation features

2.4 Create File Uploader Component

Develop a file uploader component to allow users to upload documents to the SharePoint library directly from the application.

- Implement file upload API
- Build file uploader UI
- Test file upload functionality

2.5 Develop Folder Page

Create a dedicated page for managing and viewing documents within a specific folder.

- Build folder page layout
- Integrate folder specific features
- Test folder navigation

2.6 Build Files Overview Page

Create a comprehensive overview page that displays all documents within the library, with sorting and filtering options.

- Develop overview page layout
- Dynamic file type icon

- Implement sorting and filtering logic
- Test overview page functionality

2.7 Integrate File Viewer

Enable users to view documents directly within the application without needing to download them, enhancing convenience.

- Develop file viewer page
- Test file viewing functionality

2.8 Implement Document Download Functionality

Enable users to download documents through the application interface.

- Integrate download API
- Build download UI component
- Test download functionality

2.9 Add Document Favoriting Feature

Allow users to favorite documents for easy access, with favorited items stored and managed through custom metadata.

- Implement favoriting logic
- Update UI to support favoriting
- Test favoriting and retrieval
- Develop Favorites page

2.10 Set Up Document Search Capabilities

Integrate search functionality to allow users to find documents quickly within the digital asset manager.

- Implement search integration
- Develop search UI component
- Test search accuracy and performance

2.11 Implement File Tagging System

Allow users to tag documents improving organization and searchability.

- Develop tagging logic
- Update UI for tag management

- Test tagging and filtering

2.12 Integrate Notification System

Add a notification system to inform users of actions like successful uploads, downloads, or errors.

- Set up notification component
- Develop notification UI
- Implement and test notifications

2.13 Configure Document Permissions and Access Control

Set up detailed permissions and access controls for documents, ensuring that only authorized users can view or modify them.

- Configure AD permissions
- Implement permission logic in app
- Test access control scenarios

2.14 Perform testing and validation

Conduct comprehensive testing to ensure the Digital Asset Manager functions correctly and meets all requirements.

- Test all UI components
- Validate API interactions
- Perform end-to-end testing