UPPWISE INDIA

www.uppwise.com



**Quaterly Development Planning**

**Q1-2024**

**Feature: File Extension For Document Upload**

**Feature Title:**

File extension for document upload

**Priority:**

High

**Submitted By:**

SPM

**Target Users:**

AXA

**Feature Description:**

The objective of this implementation is to restrict users from uploading unnecessary files into the SPM Application.

**Acceptance Criteria:**

We have a requirement to restrict documents to get uploaded for AXA in SPM application, for which we provide analysis and some snippets of implementation.

**File Extension Restriction:**

The application must restrict the upload of documents based on their file extensions. Only the following file extensions should be allowed:

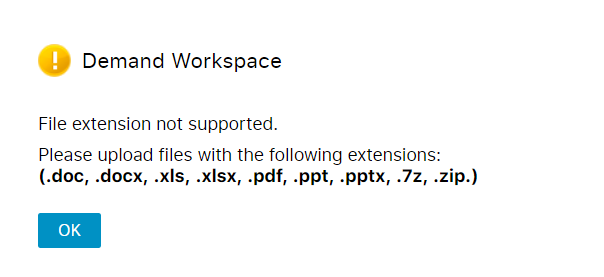
* .doc, .docx (Microsoft Word Document)
* .xls, .xlsx (Microsoft Excel Document)
* .pdf (Portable Document Format)
* .ppt, .pptx (Microsoft PowerPoint Document)
* .7z, .zip (Zip Document)
* All other file extensions must be restricted from upload.

**Alert Notification:**

When a user attempts to upload a file with a restricted extension, the application must display an alert with the following message:

Popup Define Content:

**Alert:** File extension not supported. Please upload files with the following extensions: .doc, .docx, .xls, .xlsx, .pdf, .ppt, .pptx, .7z, .zip.



**Database Integration:**

* The application must fetch the list of unrestricted file extensions from the "FileExtensions\_Accessibility" table in the AXA database.
* The file extensions marked as unrestricted in the database must be dynamically considered by the application for upload.

**Frontend Implementation:**

* The frontend logic for fetching the unrestricted file extensions should be implemented within the "documentAttach.js" file.
* The "fetchData()" function within "documentAttach.js" must integrate AJAX requests to retrieve the unrestricted file extensions.
* AJAX calls should be directed to the "GetFileAccessiableExtension" web method defined in the "ProjectHomePageWebMethod.aspx.vb" page.

**Backend Logic:**

* The backend logic for handling file extension restriction should be implemented securely and efficiently.
* Ensure that the logic accurately checks the file extension against the list of allowed extensions.
* Implement the necessary validations and error handling mechanisms to ensure the robustness of the file upload process.

Comprehensive testing must be performed to validate the behavior of the file upload feature under various scenarios:

* Uploading files with allowed extensions.
* Attempting to upload files with restricted extensions.
* Verifying the proper display of alert notifications.
* Testing the integration with the database for fetching unrestricted file extensions.
* Ensuring the frontend and backend components work seamlessly together.

By meeting these acceptance criteria, the new feature development for restricting document uploads based on file extensions in the application will be considered successful and ready for deployment.