**Feature Development Plan Template**

**Feature Name**

*Import as Document*

# Requirement Analysis

**User Story Reference:** *[Link or reference to the User Story] (optinal)*

**Summary:** *The Import as Document feature aims to provide users with the ability to import project-related information as a document into the system. This feature caters to the need for seamless data integration from external sources..*

# Impact Analysis (Optional)

**Affected APIs:** *No direct impact on existing APIs.*

**Analysis:** *The feature introduces a new functionality without directly affecting existing APIs.*

# Database Schema Changes (Optional)

**Current Schema Overview:** *No changes required in the existing schema*

**Proposed Changes:** *No modifications needed for the current database schema.*

# API Endpoints Design

**Endpoint Summary: *api/Import-as-document****.*

**Design Details:** *For each endpoint, provide the following:*

* *Endpoint URL :* ***/api/Import-as-document***
* *HTTP Method :* *POST*
* *Request Parameters: None*
* *Request Body Schema :*
* *Response Body Schema:* *Success/Failure message*
* *Error Codes and Messages :*
* *200: Document successfully imported*
* *400: Invalid document format*
* *500: Internal server error*

# Pseudo Code for Key Functionalities

**Function Overview:** *Briefly describe the key functions to be implemented.*

**Pseudo Code:** *For each key function, provide pseudo code to illustrate the core logic and flow.*

Pseudo code is a way to describe the functionality of a program or algorithm in a format that’s more readable than actual code, but not as detailed as plain language. It helps in planning and understanding the logic before diving into the actual coding.

Let’s say the feature in question involves adding a new functionality to a user profile system, where the backend needs to handle a request to update a user’s email address. Here’s an example of how the pseudo code for this functionality might look:

**Pseudo Code for Update Email Function Functionality: Import as Document**

FUNCTION ImportDocument(documentContent)

IF documentContent is not valid THEN

RETURN "Error: Invalid document format"

END IF

BEGIN Database Transaction

TRY

Parse and validate documentContent

Store document in the database

SAVE database changes

RETURN "Document imported successfully"

CATCH any parsing or database errors

ROLLBACK Database Transaction

RETURN "Error: Import failed due to [Error Detail]"

END TRY

END Database Transaction

END FUNCTION

This pseudo code outlines the basic logic for updating a user’s email. It includes input validation, checking if the user exists, and handling the database transaction, which are common steps in backend operations. Note that pseudo code is not meant to be executed, and the syntax is not language-specific. It serves as a blueprint for the actual code that will be written.

# Deployment and Configuration Changes (Optional)

**New App Settings:** *None.*

**Deployment Changes:** *No additional deployment steps required.*

* *Testing and Verification Conduct unit tests for document parsing and database integration.*
* *Perform integration tests for the Import as Document feature.*
* *Manual testing for various document formats and error scenarios*

*.*