1.0 **INTRODUCTION**

This document details the project plan for the development of Appraisal System Application.

It is intended for developers, designers, and testers working on the application. The plan will include summary of:

- How the system will function
- The scope of the project from the development point of view
- The technology used to develop the app
- Overall description

The Appraisal System Application would be built to evaluate how well an employee is doing in their current position in relation to results achieved within a period of time. This is to enable an organization to achieve its goals and objectives. It is also aimed at ensuring career development. The system is web-based that allows supervisors to assess their staff remotely. All staff in the system receive email which contains a link to access the forms at the various phases of the appraisal process. Once the form is completed and submitted, supervisors get notified in their emails and then review to either approve or disapprove with comments. The staff are again notified in their email letting them know whether their forms have been approved or disapproved by their supervisors.

1.1 **Intended Audience and Reading Suggestions**

This document is intended for system implementers and product owners; AITI-KACE, to provide a framework for both implementers and stakeholders.

1.2 **Platform**

The application will be developed in Angular for the front-end. This will then connect to a REST API with Fast API and retrieve data from a postgres database.

1.3 **Definition of Terms**

Term	Definition
Appraisal Form	A form used to assess how well staff are doing in their current position.
Appraisee	A person who would be assessed in their position.
Appraiser / Supervisor	A person who would be accessing someone else.

Deadline	The time range to fill an appraisal form
Appraisal Phase	A stage in the appraisal process.
Authorized User	A person who is giving the permission to perform specific task in the system

2.0 **Functional Requirement**

2.1 **Authentication Module**

- The system shall allow authorized users to sign in with their credentials
- The system shall allow authorized users to sign out of the system.
- The system shall allow authorized users to reset or create password
- The system shall use unique long characters of hash that would be different in each phase of the appraisal process and difficult to guess which would help users access their forms.
- The hash should expire once the form has been approved.
- The hash for a phase should work when it's within the deadline.

2.2 **Appraisal Module**

- The system should allow authorized users to have access to the appraisal form.
- The system should allow users to submit an appraisal form upon completion for approval.
- The system should allow users to save an appraisal form and continue later.
- The system should allow only authorized users to approve or disapprove a form.

2.3 **Staff Module**

- The system should allow authorized users to add new staff.
- The system should allow authorized users to edit staff details.
- The system shall allow authorized users to deactivate staff.
- The system shall allow authorized user to view staff details
- The system should allow authorized users to send staff email which contains the link to access an appraisal form.

2.4 **Report Module**

- The system should allow authorized users to view staff report
- The system should allow only authorized users to print staff reports.

2.5 **Notification Module**

- The system should notify appraisers to view form’s detail
- The system should notify appraisees that a form has been approved.
- The system should notify appraisees that a form has been disapproved
- The system should notify appraisees about deadlines to fill forms.

2.6 **Permission Module**

- The system shall allow authorized users to create staff.
- The system shall allow authorized users to create deadlines for various phases.
- The system should allow authorized users to send emails to staff.
- The system should allow authorized users to view appraisal form details.
- The system should allow authorized users to approve or disapprove appraisal forms.

3.0 **Non-Functional Requirement**

3.1 **Performance Requirement**

- The application should update the interface on interaction within a few seconds.
- The application should load and be usable within a few seconds.
- The database should be normalized to prevent redundant data and improve performance.
- The database should be distributed to prevent outages.

3.2 **Safety Requirement**

- The database should be secured to prevent data loss.

3.3 **Capacity Requirement**

- The application should be able to handle at least 200GB of data.

3.4 **Security Requirement**

- The application should include access control to restrict authorized usage of the software.
- Any keys used for the REST API should be stored securely.

3.5 **Software Quality Requirement**

- The application should be available to authorized users only.
- The application should never allow anyone to read an appraisal form not intended for that person.
- The application should use continuous integration so that features and bugs fixes can be deployed quickly without downtime.
- The interface should be easy to learn without tutorials and allows them to accomplish their goals without errors.