Software Requirements Specification (SRS)
Faculty Activity Reporting and Management System (FARM)

1. Abstract

In many academic departments, managing faculty-related data such as training programs, research activities, and departmental events is a time-consuming and inefficient process. At ANITS, the CSM department currently relies on manual reporting via email to designated coordinators. These coordinators are responsible for consolidating reports sent by faculty members, often facing challenges such as email overload, delayed submissions, repeated data collection requests, and potential data loss. Moreover, institutional audits and reviews like AICTE, NAAC, and UGC demand quick access to well-organized and accurate data, which is hard to achieve through the current system.

To address these issues, we propose the development of a centralized web-based application. This application will enable faculty members to submit their training, research, and event reports directly through a structured digital interface. Submissions will be verified and approved by the HOD, after which the data will be securely stored and easily retrievable. The application will also support dynamic report generation by filtering and exporting relevant data, thereby streamlining the documentation process for institutional reviews.

This system will not only reduce dependency on individual coordinators but also ensure data continuity and accuracy over time. It lays the foundation for a scalable digital repository that transforms departmental data management from a fragmented, manual process to a smart, automated system.

2. Introduction

2.1 Purpose

The purpose of this document is to describe the requirements for a centralized Data Management and Reporting Application for the CSE with AI & ML (CSM) department at ANITS. The application is intended to streamline the process of collecting, verifying, and managing faculty data related to training programs, research activities, and departmental events.

2.2 Scope

This document outlines the functional and non-functional requirements for the proposed application and serves as the reference for the development team. The system will enable faculty to submit information directly through the application interface, allow the HOD to approve or reject entries, and facilitate automated generation of structured reports for academic reviews. All changes to this document shall follow a formal change approval process. The development team is responsible for clarifying any ambiguities and should not modify requirements without proper authorization.

2.3 Definitions, Acronyms, Abbreviations

CSM – CSE with AI & ML

HOD – Head of Department

AICTE – All India Council for Technical Education

NAAC – National Assessment and Accreditation Council

UGC – University Grants Commission

2.4 References

Not applicable.

2.5 Developer's Responsibilities

The developer is responsible for:

(a) designing and developing the application,

(b) deploying the application on the client's infrastructure,

(c) training faculty and administrators to use the system effectively

3. General Description

3.1 Product Functions Overview

In the CSE with AI & ML (CSM) department at ANITS, faculty-related data is currently managed manually through emails, which is inefficient and error-prone. The department has three coordinators responsible for tracking:

(a) Faculty Training Details

(b) Research Activities and Publications

(c) Departmental Event Information for Newsletters

The proposed application aims to automate and centralize these processes by allowing faculty members to directly submit their data (training, research, events) through an online platform. Submissions will be verified and approved by the Head of the Department (HOD), after which they become a part of the verified dataset.

The application will also support:

(a) Status tracking of each submission (Pending/Approved)

(b) Dynamic filtering and selection of data fields

(c) Generating custom reports based on faculty/category/column filters

(d) Building and maintaining a structured ML-ready dataset to support institutional reviews and research output

(e) Allows filtering of faculty based on relevant attributes to easily find suitable profiles for specific academic or administrative needs.

This platform will significantly reduce redundancy, prevent data loss, and ensure smooth transitions between coordinators.

3.2 User Characteristics

The users of this system include:

(a) Faculty members: moderately tech-savvy, able to use online forms for submission.

(b) HOD: responsible for verifying data submissions, familiar with online verification workflows.

(c) Review Members: Faculty Coordinators along with selected faculty members who will review, test, and validate the submitted data and overall functionality of the application.

3.3 General Constraints

(a) The application must be accessible via standard web browsers on desktops or laptops.

(b) User authentication and role-based access control must be implemented.

(c) Data should be stored securely in a structured format (e.g., using MongoDB or any suitable database).

(d) Reports must be exportable in standard formats (PDF/Excel).

(e) Application must follow university data protection and privacy norms.

3.4 General Assumptions and Dependencies

(a) All users have a stable internet connection.

(b) Coordinators and HOD will participate in the validation and verification process.

(c) Faculty will use the application regularly to update their data.

(d) Hosting infrastructure will be provided by the department or college IT support.

4. Specific Requirements

4.1 Inputs and Outputs

Inputs:

(a) Faculty members submit reports (training programs attended, research work details, and event information) via the application interface for further analysis.

(b) HOD verifies and approves or rejects submissions.

(c) Users select columns for generating reports dynamically from the stored dataset.

Outputs:

(a) Status of faculty submissions: Pending, Approved, or Rejected.

(b) Dynamic reports generated based on selected data columns for department reviews or other purposes.

(c) Alerts or notifications for pending verifications.

(d) Historical dataset for analysis and record-keeping.

4.2 Functional Requirements

(i) Data Submission Module:

Faculty members can upload details related to:

(a) Training programs attended (mode, type, duration).

(b) Research works (domains, publications, timelines).

(c) Departmental events (dates, venue, participants, prizes).

(d) Initial submission status is set to "Pending."

(ii) Verification Module:

(a) HOD reviews submitted data for accuracy and completeness.

(b) HOD approves or rejects submissions.

(c) On approval, submission status changes to "Approved."

(iii) Data Storage and Management:

(a) The system maintains persistent records of all submissions.

(b) Data is stored as a structured dataset suitable for machine learning and report generation.

(c) Records include faculty details and timestamps.

(iv) Report Generation:

(a)Authorized users can dynamically select dataset columns via the interface.

(b) On submission, the system generates reports filtered by faculty, category, or date.

(c) Reports are exportable and suitable for departmental review and audits.

(v) Notification and Reminder System:

(a)Automated reminders sent to faculty who have not submitted required reports.

(b) Notifications to HOD for pending verifications.

(vi) User Roles and Access:

(a) Faculty: Submit and view own submissions.

(b) HOD: Verify and approve/reject submissions, view reports.

(c) Coordinators: View reports relevant to their coordination area.

4.3 External Interface Requirements

(i) User Interface:

(a) Web-based interface accessible via desktop and mobile browsers.

(b) Submission forms for different types of data entry.

(c) Approval dashboard for HOD.

(d) Dynamic report generation UI with selectable filters.

(ii) System Interface:

(a) Secure authentication for faculty and HOD.

(b) Database backend to store and retrieve faculty reports and statuses.

4.4 Performance Constraints

(a) The system must handle simultaneous submissions from all faculty members without significant delay.

(b) Report generation should complete within 30 seconds for typical departmental datasets.

(c) The application should support at least 50 concurrent users during peak submission times.

4.5 Design Constraints

(a) The system will be deployed within the college intranet or cloud environment with secure access.

(b) Use of standard web technologies for frontend and backend (e.g., React/Angular for frontend, Node.js/Django for backend).

(c) Database management system must support relational or NoSQL structures suitable for dynamic querying.

(d) The system must comply with college IT policies and data privacy regulations.

4.6 User Characteristics

(a) Primary users: Faculty members with basic computer skills, familiar with email and web forms.

(b) Secondary users: HOD and departmental coordinators with moderate IT literacy.

(c) Users require minimal training to use the submission and report interface.

4.7 General Assumptions and Dependencies

(a) Faculty members have internet access and devices capable of accessing the web application.

(b) The HOD and coordinators will actively verify and approve submissions.

(c) Data provided by faculty is truthful and complete.

(d) The college IT infrastructure supports deployment and maintenance of the system.