Software Design Specification

Faculty Management System

Revision 1.0

Prepared by Group 4

ADARSH M S - 5
RINU JAUHAR - 58
SREEKANTH CS - 65
DEVANARAYANAN - 21

Table of Contents

Revision History

Version	Name	Reason For Changes	Date
1.0		*Initial Revision*	

Approved By

Name	Signature	Department	Date

1. Introduction

This document serves as a software design guide for a web platform that allows users to manage faculty data in an educational institution. The purpose of this web-oriented service is to provide a user-friendly platform for faculties, staff, and college authorities to manage faculty data in an educational institution.

1.1 Project Scope

The scope of this project is to develop a web platform that enables users to manage faculty data in an educational institution. The web platform will provide a user-friendly interface that allows the faculties and other authorized members to access the data in the required format.

1.2 Project Overview

This document outlines the overall architecture, functionality, and design of the application. It provides a comprehensive guide for developers, designers, and stakeholders to understand the application's features and functionalities, including the system's design and user experience.

This application is developed using React, a free and open-source front-end JavaScript library for building user interfaces based on components. The application's primary goal is to streamline the administrative and academic activities of a particular department

2. Architecture

2.1 General Constraints

This app is made using React and Node js .The app is designed in such a way that it can handle large amount of traffic

- **Performance**

  The website will be created with the goal of offering quick response times to users, ensuring that there is minimal delay and website downtime. To achieve this, the website will be optimized for performance using efficient algorithms and data structures in its design

.

- **Software requirement**

  The application's front end is built using React js, while the back end uses Node js, which also offers real-time database functionality.

- **Hardware requirement**

  1. Processor

  A multi-core processor with a clock speed of at least 2GHz is recommended for running a web application using ReactJS and NodeJS.

  2. Random Access Memory

  At least 4GB of RAM is recommended, although the exact amount will depend on the size of the application and the expected traffic. For larger applications or high traffic websites, 8GB or more of RAM may be necessary.

  3. Storage

  React requires a minimum of 3GB of free disk space to install, but you'll also need additional space to store your project files.

  4. Graphics Card

  A dedicated graphics card is not necessary, but it can help with the performance of the application

  5. Emulator or Physical device

  You'll need a laptop device with chrome and an editor to test and run your web during development.

  6.Operating System

  You can use any modern operating system such as Windows, macOS, or Linux.

7.Network

You'll need a reliable internet connection with sufficient bandwidth for your application to run smoothly.

- **Data design**
  1. **Data Structures**
     - **A. Faculty**

       This entity represents a faculty member and includes attributes such as name, email, phone number, department, academic qualifications, and work experience. Each faculty member has a unique identifier (ID). This data structure should be designed to allow for efficient querying and searching of faculty details.

     - **B. Education**

       A data structure to store information about educational qualification information of faculty such as degree, branch, specification institution and other required details.

     - **C. Research**

       A data structure to store information about publications, funded projects, own lab, consultancy, guided projects, research guide.

     - **D. Roles**

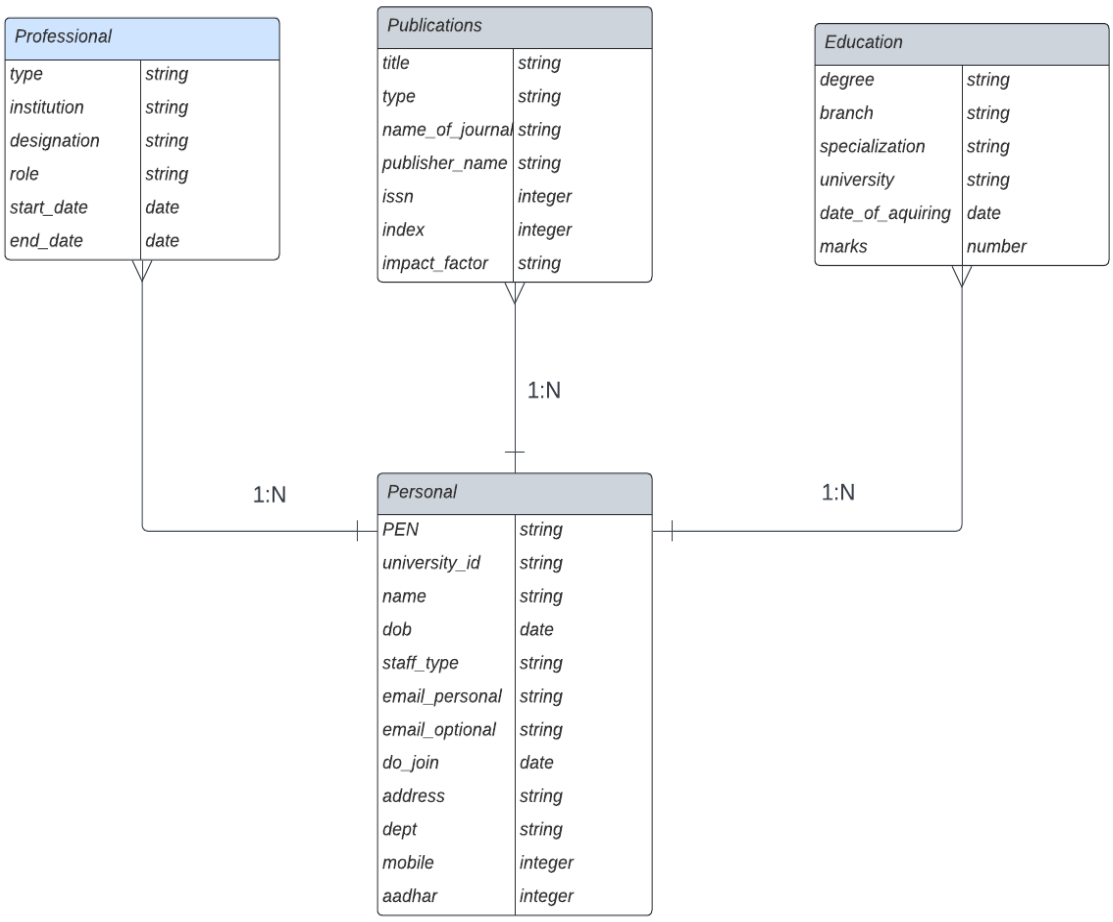
       A data structure to store information about the role of user such as administrative, co-curricular, extra-curricular and interaction with outside world.

     - **E. Teaching**

       A data structure to store the information about the history of teaching experience such as details of previous branch, semester, result and so on.

2. **Databases**

schema


Professional
type string
institution string
designation string
role string
start_date date
end_date date
Publications
title string
type string
name_of_journal string
publisher_name string
issn integer
index integer
impact_factor string
Education
degree string
branch string
specialization string
university string
date_of_aquiring date
marks number
1:N
1:N
1:N
Personal
PEN string
university_id string
name string
dob date
staff_type string
email_personal string
email_optional string
do_join date
address string
dept string
mobile integer
aadhar integer

3.Detailed design

Components

The following components will be used in this project.

A. User registration and login

Allow users to register and login to the website.

Input: User credentials (username,password,type)

Output: Successful login and access to user's account.

B. Personal details

Allow faculty to update their personal details.

Input: Faculty details like name, date of birth, email, address, Aadhar, department

Output: Display the personal details of the faculty

C. Education details

Allow faculty to update their education information.

Input: Details related to the degree like type, branch, university, marks, specialization

Output: Details of the educational information about the faculty

D. Professional details

Allow faculty to update their professional details.

Input: Details like their role, institution, designation, period

Output: Displays the professional details of the faculty

E. Publication details

Allow the faculty to update their details about various publications.

Input: Details like Title, publisher name, type, index, impact factor

Output: Displays the publication details of the faculty

4. User interface design