NITK Management System

A CS814 Course Project
Report

Submitted by

Rishi Verma (202CS022)
Harshil Jain (202CS011)
Mrinmoy Guria (202CS017)

Department of Computer Science and Engineering

National Institute of Technology Karnataka, Surathkal

P.O. Srinivasnagar, Surathkal, Mangalore-575025 Karnataka, India
January 2021

Contents

1 Introduction

The NITK management system is a software that use to maintain the record of the teachers and students of NITK. It helps in managing the details of students like personal information and marks and information related to faculties like their department and roles, and also enforce the authorization . NITK Management Systems is software that helps to maintain a database that is useful to enter new details of teacher and student with the main authority as the Admin. Moreover, it also reduces the manual record burden of the Admin.

NITK management system allows the Admin to maintain the records and update them in a more operative manner that saves the time. It is also convenient for the Admin to manage the process of records like updating, adding and deleting records of everyone. Also gives the role and authority to the teacher to update marks of their respective subject for the enrolled students.

1.1 Importance of NITK management system

- A NITK management system is a very proficient and easy to use system for managing all the student-faculty related activities in a very effective way.
- This system will reduce all the manual work and the whole process can be managed just through single clicks and edits.
- There will be no headache and doubtfulness of storing the data securely and searching the records of any individual afterward.
- The Admin can facilitate stakeholders with some extra authorizations and privileges.
- Only, one person is required to take care of the whole system, without any chances of mistakes.

1.2 Advantage of the NITK management system

- Admin can update the information of Student and faculty Faculty can update the details of its own students.
- It saves human efforts and time.
- Student can view their marks and details.

This system is nowadays essential for very organizations where there is role based access is required in this case it is an institute. This software helps in maintaining the roles and permissions given to a particular role like faculty can efficiently view and update ts students data and admin is the overall head of the whole activity

2 System Requirements

2.1 Non Functional Requirements

- Efficiency requirement: This system efficiently helps in giving managing the records of student and faculty. The admin is the owner but can also assign role to faculty which will keep work distributed in an efficient way.

- Reliability requirement: The system should accurately performs data management like addition and updation of student and faculty data by admin and manipulation of student marks by their respective faculties.
- Usability requirement: The system is designed for a user friendly environment so that admin, faculty and student of NITK can perform the various tasks easily and in an effective way.
- Implementation requirements: In implementing whole system is uses angular fro front end, Python for back end and for server 000webhost and for database PhpMyadmin uses MYSQL to store table.

2.2 Functional requirements

- **Role Based User login:** Single login is created for the login of Admin, Student and faculty, the login process is role based and all of them are directed to their respective dashboard on the basis of their role. Here the username is also associated with the role-id

  -username is entered by user during registration.

  -The system must only allow user with valid id and password to enter the system.

  -The system perform authorization process which decides what user level can access to.

  -The user must be able to logout after they finished using system.

2.3 Software and Hardware requirements

Software requirements:

- Operating system: any operating system
- Database: MYSQL is used as database as it easy to maintain and retrieve records by simple queries which are in English language which is easy to understand and easy to write.

- Programming language: Angular, HTML, CSS for front end
- Back end: PHP, PhpMyadmin.

Hardware requirements

- intel core
- RAM 1GB

2.4 Software Tools used

The whole project is divided in two parts the front end and the back end.

Front end

- HTML: Html or hyper text markup language is the main markuplan- guage for creating web pages and other information that can be dis- played in a web browser. HTML is written in the form of HTML ele- ments consisting of tags enclosed in angle brackets, within the web page content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.
- CSS: is used for styling web pages.
- Angular: It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly.
- MYSQL is also used in many high-profile, large-scale websites.

3 Authorization

RBAC: Role-based access control (RBAC) systems assign access and actions according to a person's role within the system. Everyone who holds that role has the same set of rights. Those who hold different roles have different rights.

3.1 Need of RBAC based authorization

Every company has sensitive documents, programs, and records. Protect them too strictly, and company's work grinds to a halt. Leave them open, and catastrophic security issues can arise.
Using RBAC we can grant access to those who need it while blocking those who don't need access. Make changes based on a person's role rather than individual attributes. We can make these changes quickly by altering access by role.

3.1.1 What is RBAC

All role-based control systems share core elements, such as:

- **Administrators**: They identify roles, grant permissions, and other- wise maintain security systems.
- **Roles**: Workers are grouped together based on the tasks they perform.
- **Permissions**: Access and actions attach to each role, and they outline what people can and cannot do.

RBAC systems do not require:

- **Differentiation of individual freedoms**: Access is defined by a person's role, not that person's preferences or wishes. This makes it easy to manage permissions.
- **Intensive maintenance**: Permissions follow roles. A new job function becomes a new role applied to dozens (or hundreds or thousands) of employees with only a small amount of work for the administrator. Promotions involve changing roles, not editing permissions as line items.

3.1.2 Roles withing RBAC

Roles dictate authorization within an RBAC system. It's critical to define them properly. Otherwise, large groups of people withing company can't do their work.
Roles can be defined by:

- **Authority**: Senior management needs access to flies interns should never see.
- **Responsibility**: A board member and a CEO might hold similar authority within a company, but they are each responsible for different core functions.
- **Competence**: A skilled worker can be trusted to work within sensitive documents without errors, while a novice could make catastrophic mistakes. It's important to tailor access accordingly.

3.1.3 Role-Based Access Control Permissions

Permissions specify what people can access and what they can do in the system. Think of permissions as the rules people follow per the roles have outlined. Permissions should involve:

- **Access**: Who can open a specific drive, program, file, or record? Who shouldn't even know these things exist? Access will limit what people can see.
- **Reading**: Who can scan through these documents, even if they can't change anything inside of them? Some roles may have the ability to reference materials but not make changes to them.
- **Writing**: Who can change documents? Does someone else have to approve the changes, or are they permanent? This will define permissions.
- **Sharing**: Who can download a document or send it as an email attachment? As with the other permissions, some users will not be able to share materials even if they can reference them.
- **Finances**: Who can charge money? Who can offer refunds? Permissions could involve the ability to deal with charges and refunds, set up credit accounts, or cancel payments.

3.1.4 Role-Based Access Control Benefits

Security options abound, and it's not always easy to make the right choice for company. RBAC comes with plenty of tried-and-true benefits that set it apart from the competition. An RBAC system can:

- **Reduce complexity**: New employees gain access based on their roles, not on long lists of server and document requirements. This simplifies creating, maintaining, and auditing policies.
- **Allow global administration**: Change access for many employees all at once by altering permissions associated with a role.
- **Ease on boarding**: As people join, move withing, or are promoted within organizations, and we don't have to worry about the individual's permissions, just that they're in the right place. The roles take care of the rest.
- **Reduce Blunders**: Traditional security administration is error-prone. Adding permissions for individuals gives us plenty of options to make a mistake. Change a role's access, and you're less likely to give someone too much (or too little) power.
- **Lower overall costs**: When admin duties shrink, comanies save on security administration. This saves our organization time and money.

3.2 Components of RBAC in our application

NITK management system has following components:

Admin

- Can view information of any Student.
- Can Edit information of any Student
- Can delete information of any Student.
- Can view information of any Student

- Can delete information of any Student.
- Can edit information of any Student.
- It is the main authority of the system.
- Can add/remove a faculty/student.

Faculty

- Can view marks of student of all subject.
- Can add marks of students of his subject.
- Can edit marks of student of his subject.
- Can delete marks of student of his subject.
- Can update its own personal information.

Student

- Can view marks of all students.
- Can only view his marks and cannot edit it.
- Can update its own personal information.

Permissions

- Create:
  - Admin:
    - Can create a new student or faculty.
  - Faculty:
    - Can add his subject's marks
- Update:
  - Admin:
    - Can update information of any Student or faculty.

- Delete:
  - Admin:
    - Can delete any data regarding to a particular student or faculty.
    - Can also delete a particular student or faculty.
  - Faculty:
    - Can delete marks of student of his own subject.
- Partial Edit:
  - Faculty:
    - Can only edit marks of the student of his own subject.
    - Can only edit his own personal information.
  - Student:
    - Can update his own personal information.

3.3 Components of administrative model

To ensure better functioning and smooth running of an organization, these are following components in our project:

- **Planning**: In this project the admin have to create a database such it will store all the information about student and faculties in the organization, here we have unique User-id assigned on the basis of role for all the users namely admin, faculty and student which allow us to uniquely identify the user.
- **Organization**: In the NITK management system the decision is taken as to what kind of authority is to be provided for an area of service.
- **Directing**: The admin plays the main directing role in the system as he is the one which assign a particular role to the faculty and also add information regarding the other users, by implementing

access control model it distributes the work among various faculties in the hierarchy which reduces the load over a single person and also keeps the working efficient. Admin has to be very smart and keen about the responsibility he have.

- **Coordinating**: It is essential to interrelate various part of an organization in a harmonious way. The coordination can be achieved if the head knows about all the jobs and effect such as administrative machinery that he feels practically on necessary to interfere. It depends upon the kind of organization has been brought in to being, as to whether is line type or line and staff type of structure or functional type of organization.

4 Implementation

The NITK Management System starts from the common login page.

- **Login Page:**
  - login module is kept as role based.
  - User-id entered by the user is checked in the **database**.
  - Role assigned to the corresponding user-id is checked and that screen is opened.
- **Screen:**
  - There are three types of screen namely:
    - People
      - Admin: Can Add, Update, Delete roles and information for student and faculty
      - Teacher: Can Add marks for his subject student and edit his information
      - Student: Can view his marks and can update his own personal information

- Marks
    - Admin: Can add, update, delete marks of any student.
    - Faculty: Can only add, update, delete marks of his subject's student.
    - Student: Can only view his marks and other students marks

- **Roles:** There are three types of users, namely:
    - Admin
    - Faculty
    - Student
- **Permissions:**
    - There are four types of permissions namely:
        - Create:
            - Admin can create data for student and faculty
        - Update:
            - Admin and faculty are given the update permissions
        - Delete:
            - Admin can delete anyone's data
        - Partial Edit:
            - Faculty can only add, update or delete information of

student of his subject only.

5 Conclusion

The project NITK management system is for computerizing the working in a NITK. The software takes care of all the requirements of a NITK management for student and faculty related information and is capable to provide easy and effective storage of information. NITK management system follows RBAC policy for authorizaion which makes sure that roles as assigned properly with the permission and no permissions should overlap.

References

[1] https://www.okta.com/identity-101/what-is-role-based-access-control- rbac/

[2] https://pypi.org/project/django-rbac/

[3] https://django-role-permissions.readthedocs.io/en/stable/roles.html

[4] https://en.wikipedia.org/wiki/Role-based$_a ccess_c ontrol$