FACULTY RECORD MANAGEMENT SYSTEM

MINOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

Information Technology

Submitted By:
Prabhkirat Singh (2004970)
Prerna Sharma (2004972)

Submitted To:
Prof. Harjot Kaur

Department of Information Technology,

Guru Nanak Dev Engineering College,

Ludhiana-141106

ABSTRACT

Faculty Record Management System (FRMS) is a comprehensive software solution designed to streamline and automate the management of faculty records within educational institutions. With the increasing complexity and volume of faculty-related data, it has become essential for institutions to adopt efficient and reliable systems to manage this information. The FRMS provides a user-friendly interface that allows administrators and staff members to effectively manage faculty records throughout their lifecycle. The system encompasses various modules, including faculty profile management,Student record, Their attendances , administrative access to the resources, Storing test records.

Through the faculty profile management module, the Faculty Record Management System enables administrators to maintain accurate and up-to-date information about faculty members, such as personal details, educational qualifications, areas of expertise, and contact information and publications record.

Performance evaluation is a crucial aspect of faculty management, and the Faculty Record Management System assists institutions in conducting fair and consistent evaluations. The system allows administrators to define evaluation criteria, schedule assessments, and generate comprehensive notices. It promotes transparency, aids in identifying areas of improvement, and recognizes outstanding performance.

Moreover, the Faculty Record Management System supports the tracking of faculty professional development activities, conferences, research publications, and grants. By recording and monitoring these activities, institutions can assess faculty engagement and allocate resources appropriately to promote ongoing growth and development.

ACKNOWLEDGEMENT

We are highly grateful to Dr. Sehajpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the minor project work at Guru Nanak Dev Engineering College , Ludhiana.

The constant guidance and encouragement received from DR. KULVINDER SINGH MANN, HEAD OF DEPARTMENT, IT, GNDEC Ludhiana have been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to Dr.Pankaj Bhambri , without his wise counsel and able guidance, it would have been impossible to complete the project in this manner.

We express gratitude to other faculty members of the Information Technology Department of GNDEC for their intellectual support throughout the course of this work.

Finally, We are indebted to all whosoever have contributed in this report work.

Prabkirat Singh

Prerna Sharma

Contents

1 Introduction

1.1 Introduction to Project

A Faculty Record Management System is a specialized software solution designed to streamline and automate the management of faculty records within educational institutions. It serves as a centralized platform that efficiently handles various aspects of faculty management, including faculty profiles,record of their journals and publications, It even manages the record of the student enrolled in the provided courses.

In today's digital age, educational institutions face the challenge of managing an increasing amount of faculty-related data and student related data. Manual record-keeping systems often lead to inefficiencies, data inaccuracies, and time-consuming administrative tasks. The Faculty Record Management System offers a comprehensive solution by providing a user-friendly interface and a range of features tailored to the specific needs of educational institutions.

The system typically includes modules that cater to different aspects of faculty management. These modules allow administrators to create and maintain comprehensive faculty profiles, including personal information, educational qualifications, student record management, and contact details. It becomes a centralized repository of faculty data, facilitating quick retrieval and reducing the chances of data duplication or loss.

Additionally, the system supports tracking faculty professional development activities such as conferences, research publications, and grants. By recording and monitoring these activities, institutions can assess faculty engagement and allocate resources appropriately to foster continuous growth and development.

1.2 Project Category

Application or System Development.

Our project is mainly based on Web Development Project.Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites. The word Web Development is made up of two words, that is: Web: It refers to websites, web pages or anything that works over the internet. Development: Building the application from scratch.

1.3 Objectives

1.To maintain all the record of faculty of concerned departments.

2. To provide administrative access to perform various task such as add as well as remove faculty as well as student, generate notices , add or remove courses.

3. To manage student data such as recording attendance and generating tests by the concerned faculties of the department.

4.To display the publication and journals details of concerned faculty members as per the dynamic requirements from the department

1.4 Problem Formulation

The problem addressed by the Faculty Record Management System is the inefficient and error-prone management of faculty records in educational institutions. Traditional manual record-keeping systems, involving paper-based documents or scattered digital files, often lead to challenges such as data duplication, inconsistency, lack of centralized access, and time-consuming administrative tasks. These issues can hinder effective faculty management, impede decision-making processes, and result in inefficiencies in recruitment, performance evaluation, leave management, and professional development tracking.

1.5 Identification/Reorganization of Need

Increasing Complexity of Faculty Data Educational institutions deal with a growing volume of faculty-related data, including personal information, qualifications, research activities, and performance evaluations. Manually managing and organizing this data becomes increasingly difficult and time-consuming, leading to the need for a more efficient and organized system.

Inefficiencies in Manual Processes Traditional methods of managing faculty records, student records such as paper-based documents or scattered digital files, often result in inefficiencies. Time-consuming administrative tasks, potential errors in data entry, difficulty in accessing information, and challenges in coordinating processes among various stakeholders highlight the need for a streamlined and automated solution.

Lack of Centralized Access and Data Integrity When faculty records are dispersed across different departments or stored in various formats, it becomes challenging to ensure centralized access and data integrity. The absence of a centralized system leads to duplication of efforts, difficulties in retrieving information, and potential discrepancies in faculty data.

Increasing Complexity of Student Data Educational institutions deal with a growing volume of student-related data, including personal information, attendance records, test records etc. Manually managing and organizing this data becomes increasingly difficult and time-consuming, leading to the need for a more efficient and organized system.

2 Requirement Analysis and System Specification

2.1 Feasibility Study

A feasibility study is done by analyzing technical, economic, legal, operational and time feasibility factors. This chapter describes all the feasibilities that come as questions to both the developers and other users during the development of software. The chapter contains technical feasibility, economic feasibility and operational feasibility. Our proposed system is legally feasible as our project meet all the legal requirements. Our proposed system is time feasible as time taken for execution and completion is reasonable. A quantitative approach is proposed to obtain measurable, comparable judgments of simulation correctness.The commonality between machine learning and simulation model validation is analyzed. We focus on the idea of applying cross validation in the area of simulation validation. Based on cross validation, a strategy is proposed to predict the fit of a simulation model to a validation set. Scaling factor is then introduced into the approach to improve its efficiency. The approach is applied in a simulation system to verify the usefulness of the approach proposed. The feasibility of our proposed system can be evaluated as: -

Technical Feasibility: This assesses whether the institution has the necessary technical infrastructure and resources to support the implementation and maintenance of the Faculty Record Management System. Considerations include hardware and software requirements, compatibility with existing systems, data security measures, and availability of technical expertise.

Code Editor: Visual Studio Code, Chrome (to display the output)
Languages: HTML5 , CSS3 , Js ,MERN stack
Model used: Software Development life cycle

Financial Feasibility: The financial feasibility study determines if the institution has the financial resources to invest in the development, implementation, and ongoing maintenance of the system. It involves assessing the costs associated with software acquisition or development, hardware upgrades, staff training, and system maintenance. Additionally, the study evaluates potential cost savings or benefits resulting from system implementation.

Operational Feasibility: The operational feasibility study focuses on evaluating whether the proposed Faculty Record Management System aligns with the institution's operational processes and workflows. It involves understanding the current faculty management practices, identifying potential process improvements, and assessing the system's impact on day-to-day operations. Considerations include the ease of system integration, user acceptance, and the level of disruption during implementation.

. The requirements of the traffic simulation are also very small therefore it is easy to operate in every environment. As all components needed to develop the proposed system are also available,the system will definitely work. Hence the project is operationally feasible.

2.2 Software Requirement Specification Document

A Software Requirements Specification (SRS) document for a Faculty Record Management System outlines the functional and non-functional requirements of the system. Here's an example of the sections that would typically be included in an SRS document:

1. Introduction
2. Purpose of the document
3. Scope of the system
4. Overview of the Faculty Record Management System
5. System Overview
6. Description of the system's purpose and functionality
7. Key stakeholders and users of the system
8. High-level architecture and components
9. Functional Requirements

Detailed description of the system's functional requirements, including:

1. Faculty profile management
2. Recruitment and hiring processes
3. Performance evaluation
4. Leave management
5. Professional development tracking
6. Reporting and analytics
7. User management and access control

Non-functional Requirements

1. Performance requirements (e.g., response time, scalability)
2. Security requirements (e.g., data protection, access control)
3. Usability and user interface requirements
4. Compatibility requirements (e.g., browser support, integration with existing systems)
5. Reliability and availability requirements
6. Data backup and recovery requirements

System Constraints

1. Hardware and software constraints
2. Technology stack and platform requirements
3. Regulatory and legal constraints
4. System Interfaces
5. External interfaces with other systems (e.g., HR systems, student information systems)
6. APIs and integration points

7. User interface specifications
8. Data Management
9. Data entities and attributes
10. Database schema and data relationships
11. Data validation and integrity requirements

Hardware Requirements:

1. RAM: A minimum of 8 GB is required as training any algorithm will require some heavy Lifting. Less than 8 GB can cause problems while Multitasking.
2. Processor: Intel i5 10th Gen or above or Ryzen 5 4th Gen or above
3. OS: This system can run on any latest windows System.

2.3 Expected Hurdles

While making this model we came across many hurdles, some of them are listed below: First, it was difficult to write a back-end code and also connect our editor with mongoDB database. On this basis, this was quite difficult to implement this, in our code.

2.4 Validation

Validation is the process of ensuring that a software system meets the specified requirements and fulfills the needs of the users and stakeholders. It involves evaluating the system to determine its correctness, completeness, and compliance with the intended functionality. In the context of a Faculty Record Management System, validation would typically involve the following steps:

1. Requirement Validation
2. Design Validation
3. Functional Validation
4. Non-functional Validation

5. User Acceptance Testing (UAT)

2.5 SDLC model to be used

The choice of SDLC model should be based on a careful analysis of the project's characteristics and the organization's needs. For a Faculty Record Management System, where requirements may evolve over time and stakeholder involvement is crucial, an Agile or Incremental model may be more suitable. These models allow for flexibility, frequent feedback, and the ability to deliver value in smaller increments. However, if the requirements are well-defined and stable, the Waterfall model can be considered. The Spiral model can be an option if risk management is a significant concern. Ultimately, the selection of the SDLC model should be made based on the specific requirements and constraints of the project. **Phases of Iterative Modes**

2.5.1 Requirements gathering and analysis

All the requirements were collected and then checked and analysed whether the particular requirement can be fulfilled or not. The analysis includes analysing whether the particular requirement is feasible or not and the budget allows it or not.

2.5.2 Design

After the iteration requirement is gathered then we need to implement the design phase. Effective design is decided to implement the requirement out of many alternatives. This is one of the critical phases as proper design can provide the most optimal outputs will low pressure on the funds from client. This design can be a new one or extension to the already build requirement or it can be flow chart or data-flow diagram.

2.5.3 Implementation

In this, the basic codes are written and then transformed into computer program's. Here the code is written and the database of sales and products is made. The sales database includes all the basic details of the items sold in season

2.5.4 Testing

We train the model using the training data-set which includes the sales and product details of the summer season.Once the code has been implemented then this testing phase is implemented to identify any defects that are present in the code and if present then they need to be reported back to the developers. The tester can write new test cases or use existing one which they have written in previous build but the through testing is a priority as any miss will impact the specification of software.

2.5.5 Deployment

The project is currently deployed on the local server and uses it as the working environment.

2.5.6 Review

In this phase, the developed requirement is reviewed to meet all the standards as per the currently decided requirement. Basing on this further plan requirement plan is drafted and implemented as part of the next iteration cycle.

2.5.7 Maintenence

After the deployment of the project, there might be some bugs left, so some updates will be required. This involves debugging and adding new additional features.

3 System Design

3.1 Design Approach

System Design for a Faculty Record Management System involves translating the requirements into a detailed technical design that outlines the system's architecture, components, databases, and interfaces. Here are the key aspects of the system design:

Architecture Design

1. Define the overall system architecture, including the high-level structure and components.
2. Identify the layers or tiers of the system, such as presentation layer, business logic layer, and data access layer.
3. Determine the communication protocols and technologies to be used between system components.

Database Design

1. Identify the required data entities, relationships, and attributes.
2. Design the database schema and tables to store faculty records, performance evaluations, leave management data, and other relevant information.
3. Define primary keys, foreign keys, indexes, and constraints to ensure data integrity and efficient data retrieval.

User Interface Design

1. Design an intuitive and user-friendly interface for different user roles, such as administrators, faculty members.
2. Create wireframes or mock-ups to visualize the user interface layout, navigation, and interaction flow.
3. Incorporate usability principles to ensure an efficient and enjoyable user experience.

Component Design

1. Identify the functional components/modules of the system based on the requirements.
2. Define the responsibilities and interfaces of each component.
3. Design the data models, classes, methods, and functions for each component.
4. Specify input and output formats for data exchange between components.

3.2 User Interface Design

The user interface of our project is very simple and elegant. The home page is as follows: There are two main components which includes various functions i.e there are main two home pages:

1. Admin The user interface of admin will contain various features. You have to first login as an admin then you will be provided with all the administrative access.

In UI design of admin The dashboard will look like this:

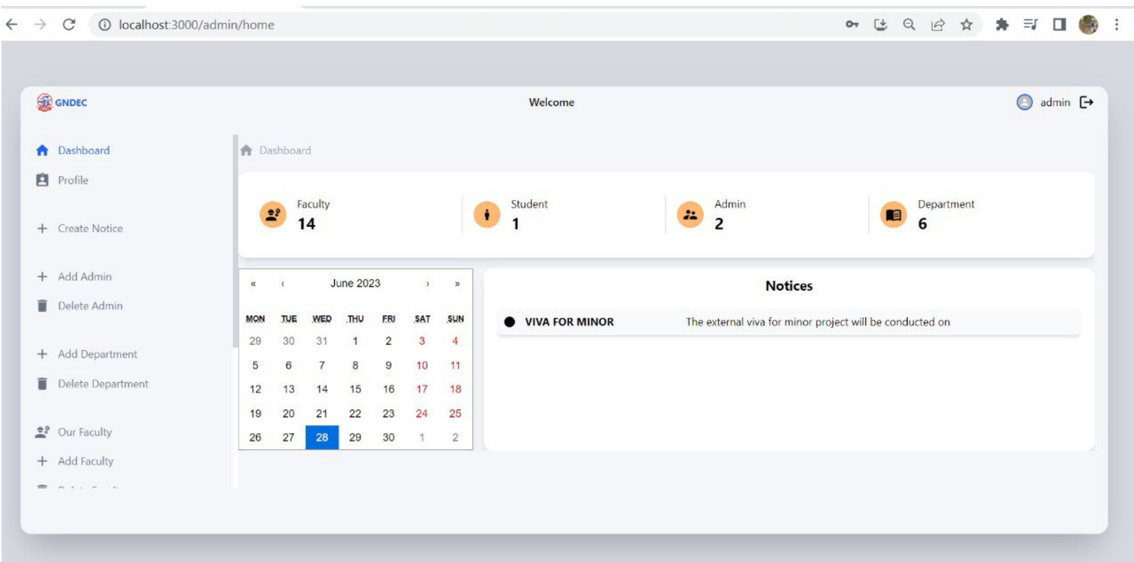

Figure 3.2.1: Admin home page

2. **Faculty**

The User interface is the page that directly interacts with the humna. Faculty home page will contain the information about the teacher, and a list of buttons on the left hand side that provide the basic features:

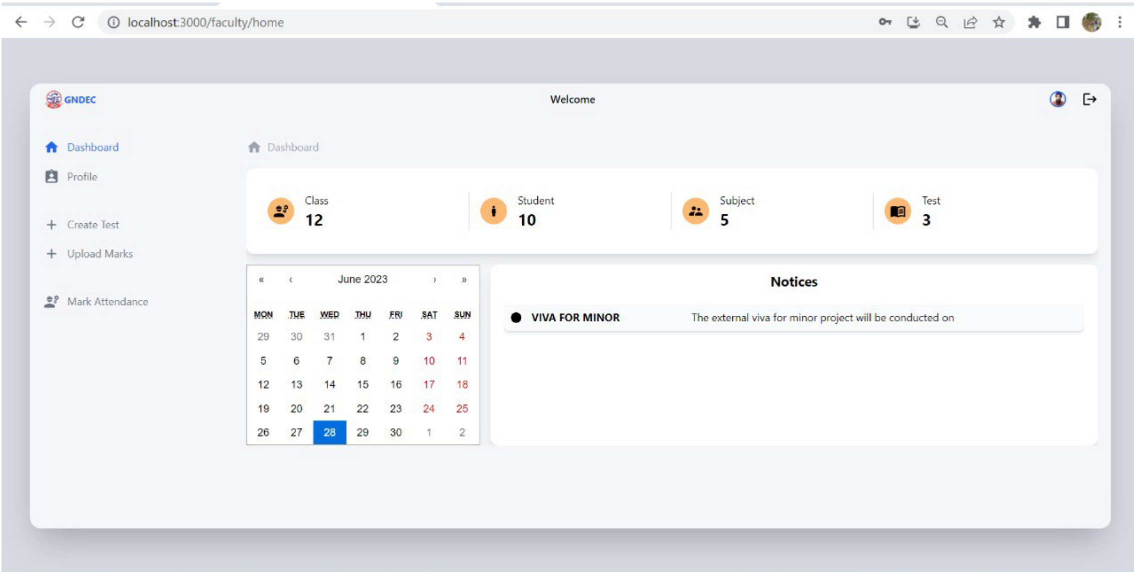

Figure 3.2.2: Admin home page

3.3 Methodology

3.3.1 Learning the required technologies

The development required proper hold over the required technologies both front end and back end such as MongoDB, Express, React js, Node js.It strictly follows the MERN stack.

3.3.2 Creating a Database

The creation of faculty record database in MongoDB so that to perform various operations on the database via filters.
This database can be created or can be deleted as when required by the administrator

3.3.3 Creating User Interface

It is very important to create a simple yet unique UI that is easy to use. The user must be able to find and use all the options and should be very comfortable with it.

The user interface designed must be in such a way that is is easy to use for the end user.

3.3.4 Connecting the Front end with Database

The operations to be performed require a connection with the database. It is done by creating a back end server with the help of APIs.
These APIs are created at the backend and are connected to the frontend.

4 Implementing, Testing and Maintenence

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation

4.1.1 MongoDB

MongoDB is an open source NoSQL database management program. NoSQL (Not only SQL) is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.
MongoDB is used for high-volume data storage, helping organizations store large amounts of data while still performing rapidly. Organizations also use MongoDB for its ad-hoc queries, indexing, load balancing, aggregation, server-side JavaScript execution and other features.

4.1.2 Express.js

Express.js is a small framework that works on top of Node.js web server functionality to simplify its APIs and add helpful new features. It makes it easier to organize your application's functionality with middleware and routing. It adds helpful utilities to Node.js HTTP objects and facilitates the rendering of dynamic HTTP objects.
It's features are:
1. Develops Node.js web applications quickly and easily.
2. It's simple to set up and personalise.
3. Allows you to define application routes using HTTP methods and URLs.
4. Includes a number of middleware modules that can be used to execute additional requests and responses activities.
5. Simple to interface with a variety of template engines, including Jade, Vash, and EJS.
6. Allows you to specify a middleware for handling errors.

4.1.3 React.js

React is a JavaScript library for building user interface and is used to build single-page applications.It allows us to create reusable UI components.

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It is an open-source, component-based front-end library that is responsible only for the view layer of the application. ReactJS is not a framework, it is just a library developed by Facebook to solve some problems that we were facing earlier

4.1.4 Node.js

Node.js tutorial provides basic and advanced concepts of Node.js. Our Node.js tutorial is designed for beginners and professionals both.

Node.js is a cross-platform environment and library for running JavaScript applications which is used to create networking and server-side applications.

4.1.5 React router

React Router is a popular library for handling routing in React applications. It allows you to create single-page applications (SPAs) with multiple views or pages, enabling navigation between them without the need for full page reloads.

React Router provides a declarative way to define the routing configuration of your application. It offers several components and APIs that help you manage different routes, render the appropriate components based on the URL, and handle navigation events.

4.2 Testing

Testing is a crucial phase in the development of a Faculty Record Management System to ensure its functionality, performance, and reliability. Here are some key aspects of testing for such a system:

4.2.1 Test Planning

Define a comprehensive test plan that outlines the testing objectives, scope, approach, and schedule. Identify the testing types to be performed, such as unit testing, integration testing, system testing, and user acceptance testing. Determine the testing tools and technologies to be used.

4.2.2 Unit Testing

Test individual components or modules of the system in isolation to ensure their correctness and functionality. Write unit tests that cover different scenarios and validate the expected behavior of each component. Use testing frameworks and tools specific to the programming language or technology being used.

4.2.3 Integration Testing

Verify the interactions and communication between different system components. Test the integration points and data exchange between modules to ensure proper functionality and data integrity. Identify and resolve any compatibility or interface issues.

4.2.4 System Testing

Conduct end-to-end testing of the entire system to validate its functionality, performance, and usability. Test various scenarios and use cases to ensure all system requirements are met. Verify system behavior under different loads, including high user volumes and data processing.

4.2.5 User Acceptance Testing (UAT)

Involve end-users, such as administrators and faculty members, in testing the system's usability and suitability for their needs. Define UAT scenarios and use cases that reflect real-world usage scenarios. Collect feedback and incorporate necessary changes based on user input.

4.2.6 Documentation and Reporting

Maintain thorough documentation of test cases, test results, and any identified issues. Generate reports summarizing the testing process, results, and recommendations for improvement. Communicate test findings and recommendations to the development team for resolution.

5 Results and Discussions

The specific results of a Faculty Record Management System may vary based on the institution's requirements, implementation, and user adoption. However, the overall objective is to enhance the management of faculty records, optimize administrative processes, and support effective decision-making within the educational institution.
The result of a Faculty Record Management System can be measured in terms of the benefits and outcomes it delivers to the educational institution and its stakeholders.
The ultimate result of a Faculty Record Management System is the delivery of a user-friendly, efficient, and reliable solution that meets the specific needs of the institution. This system improves overall efficiency, data accuracy, and collaboration while saving time and reducing administrative burdens. The benefits include streamlined faculty record management, improved productivity, effective performance evaluation, accurate leave management, enhanced reporting and analytics, and increased compliance and security.

The main output of the web app provides two types of access rights :

5.1 Administrative access

In administrative access , The admin in given a number of features by which he or she can make entries of the data and store them as per the requirements.

Admin can perform various tasks which are listed below

5.1.1 Administrator login

Firstly user have to login by the admin username and the password to gain add the administrative rights.

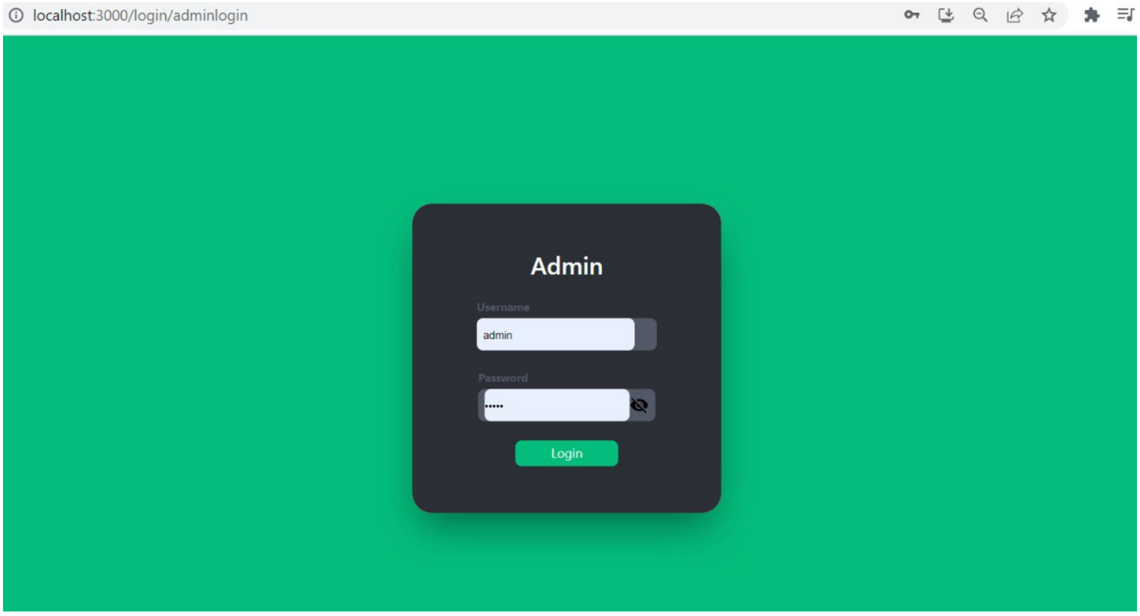

Figure 5.1.1: Admin login page

5.1.2 Dashboard

1. The dashboard includes the basic details of the number of faculties, no of students and the admins.

2. This also provides a calendar from where you can access the latest and the previous notices.

3. This dashboard provides an access to all the route pages linked to the dashboard

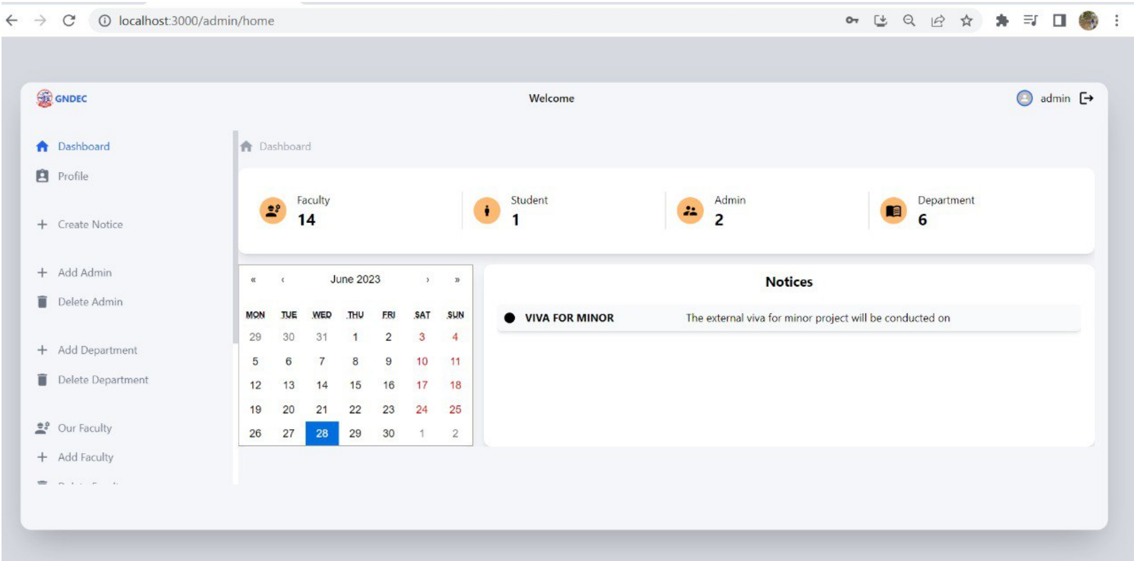

Figure 5.1.2: Admin home page

5.1.3 Admin profile

This profile contains all the basic details of the admin.

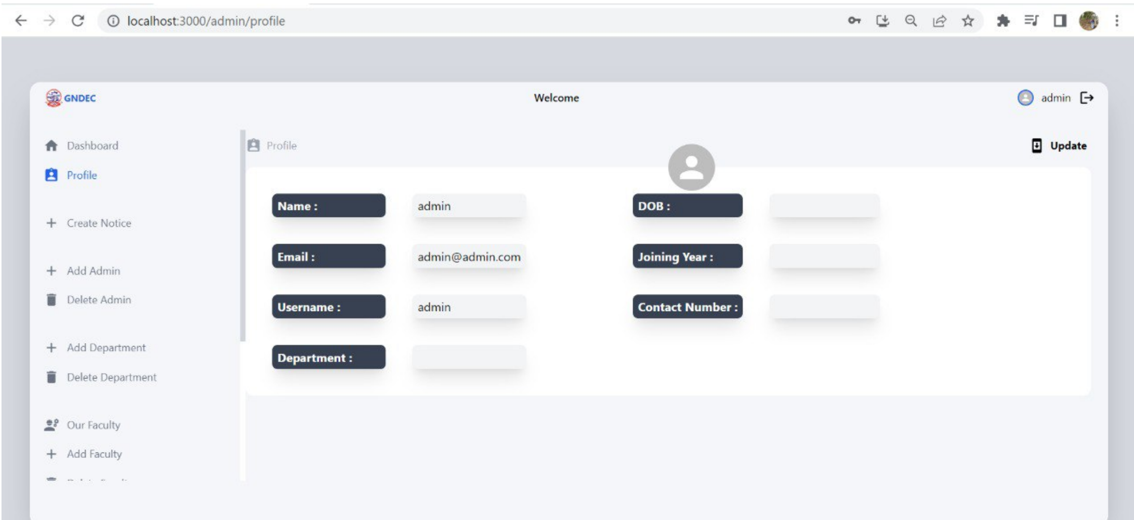

Figure 5.1.3: Admin Profile

5.1.4 Create/ Delete Admin

By this page you can create or delete admin

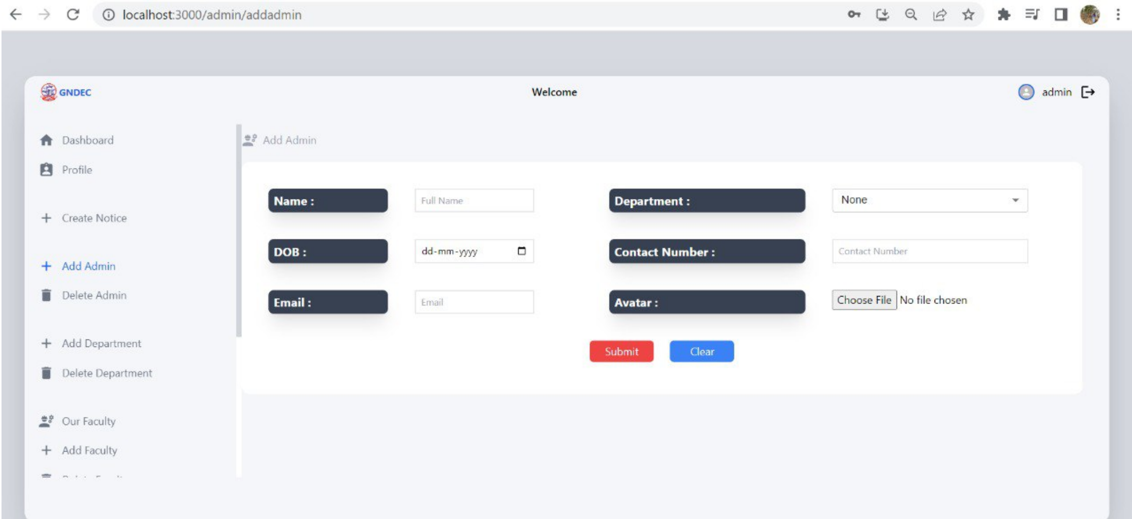

Figure 5.1.4: Adding new admin

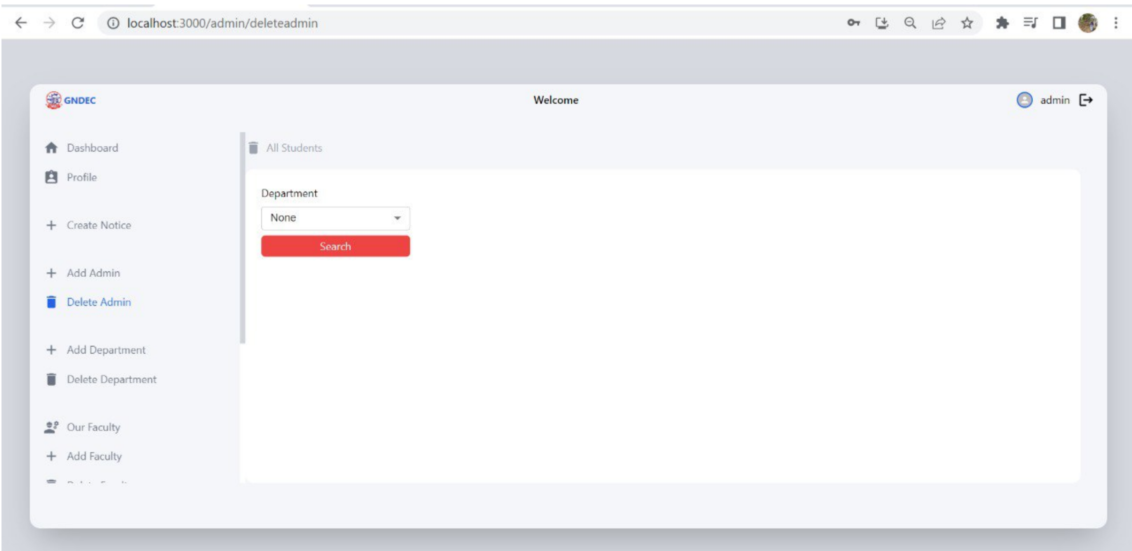

Figure 5.1.2: Deleting admin

5.1.5 Add Notice

Admin has given a right to publish a notice which will be visible to the faculty.

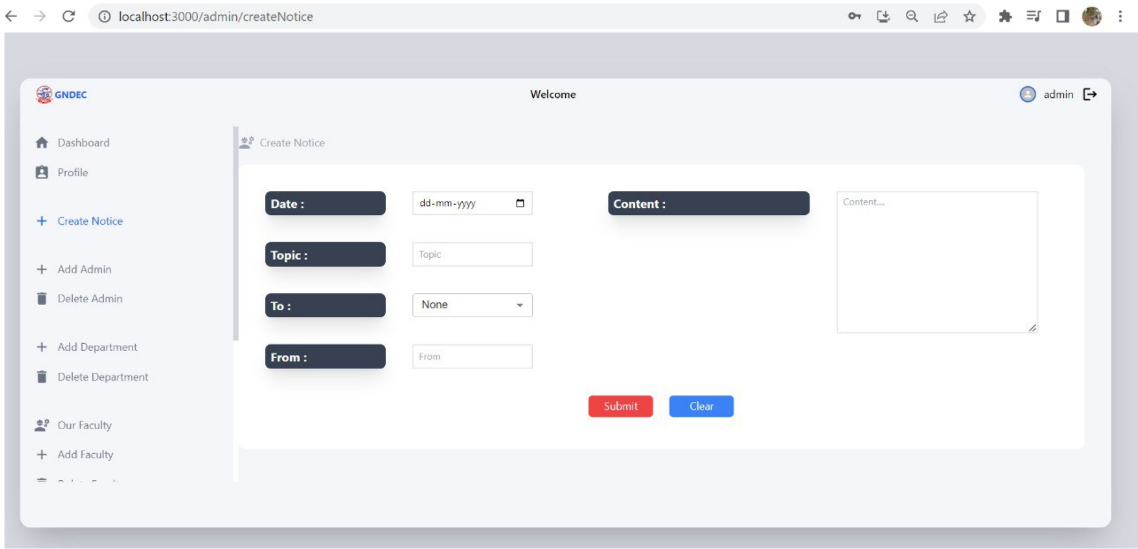

Figure 5.1.4: creating a note

5.1.6 Add/ Delete Departments

Admin also have an access to add or delete the departments, which are offered in an organisation.

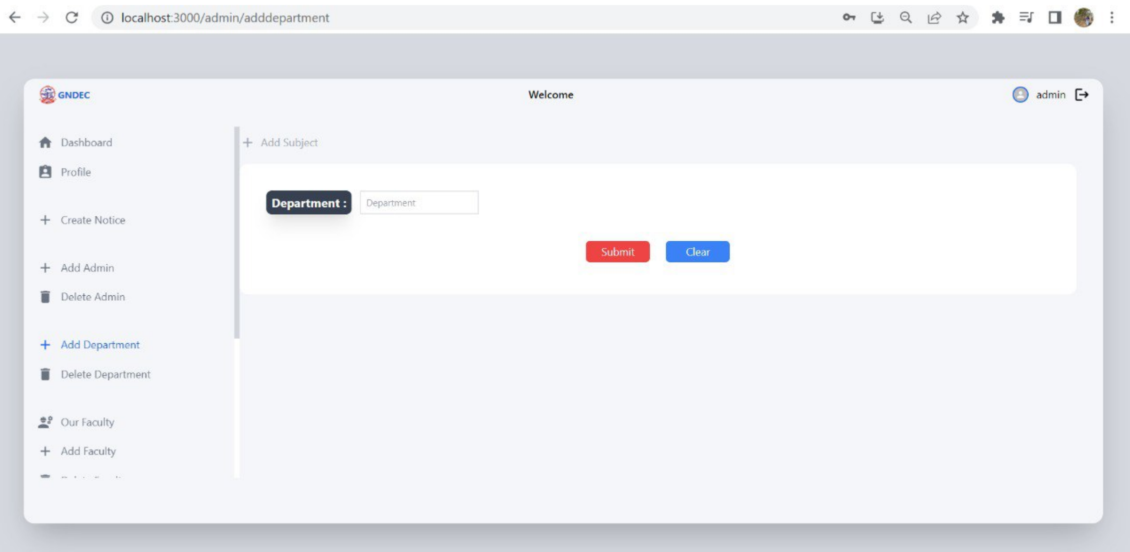

Figure 5.1.6: Adding department

5.1.7 Add/ Delete Students

Admin also have an access to add or delete the students which are studying or an enrolled in the institutions, admin also have a list of all for the same.

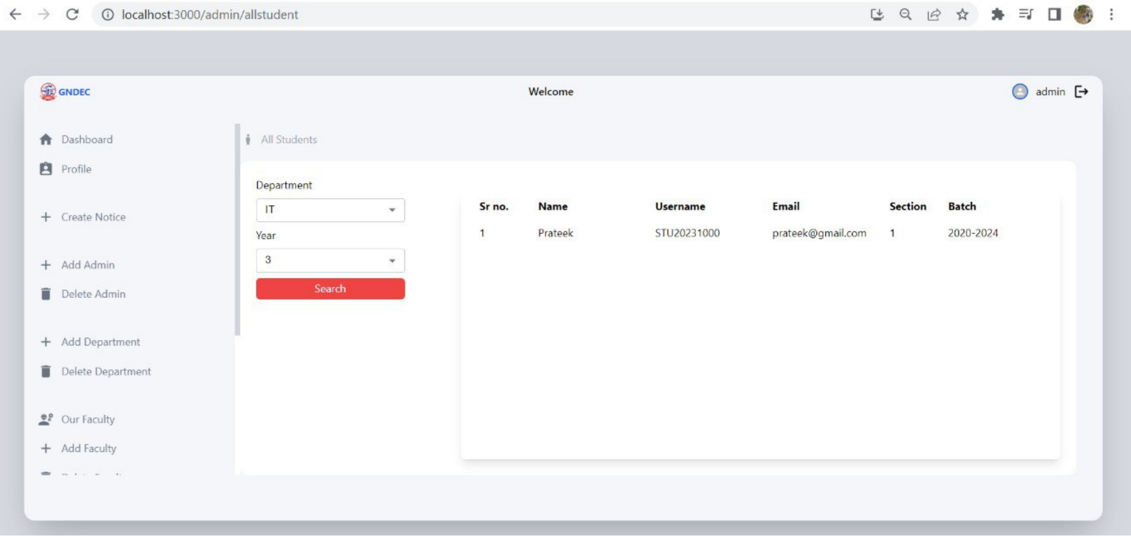

Figure 5.1.7: Student list

5.1.8 Add/ Delete Faculty

Admin also have an access to add or delete the Faculty which are teaching in the institutions, admin also have a list of all for the same.

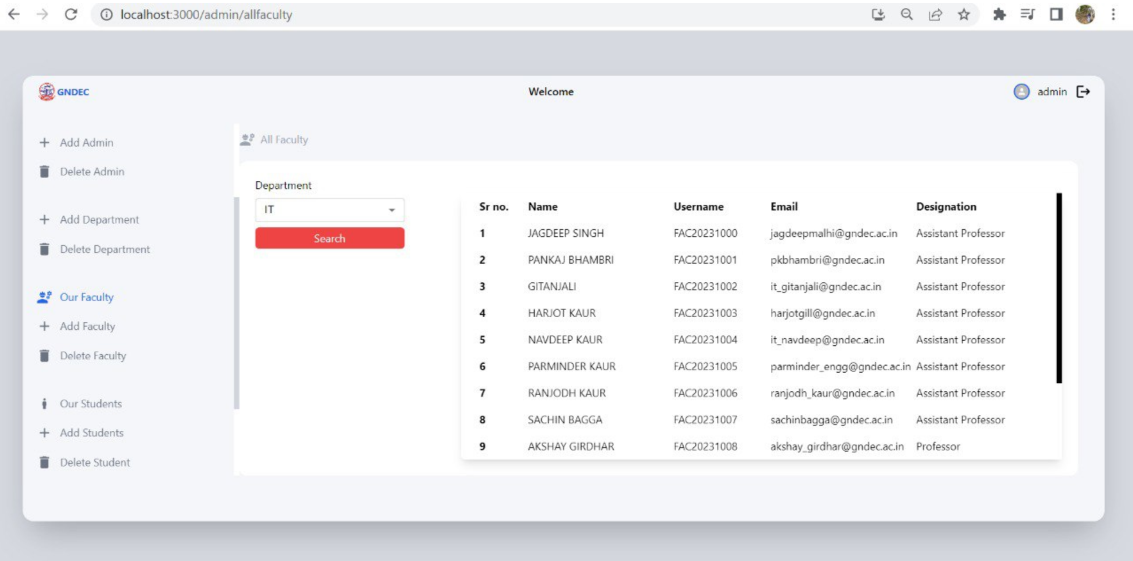

Figure 5.1.8: Faculty list

5.1.9 Add/ Delete Subjects

Admin also have an access to add or delete the list of subjects which are to be taught in the institution.

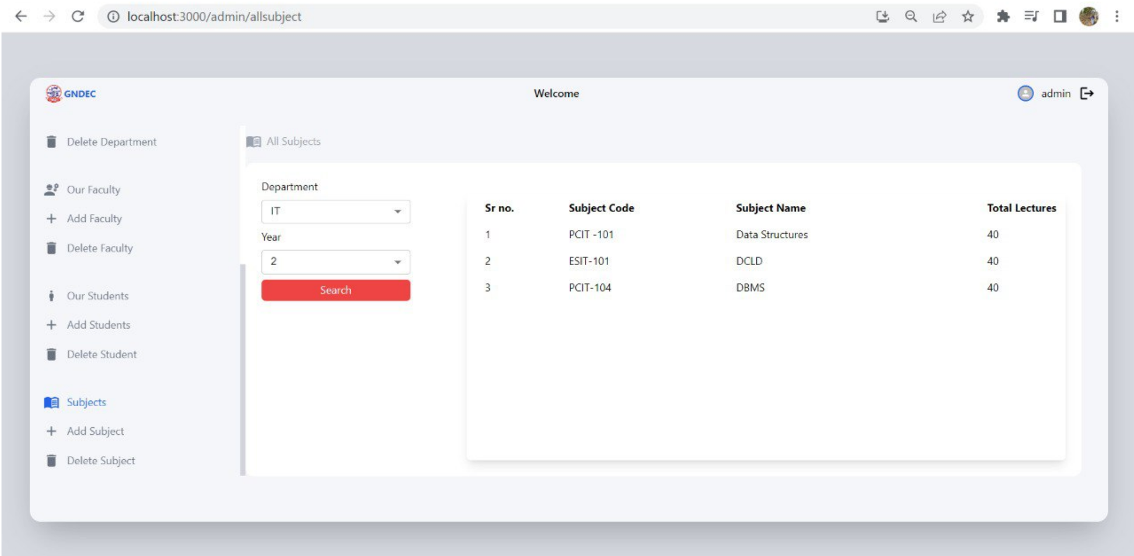

Figure 5.1.9: subject list

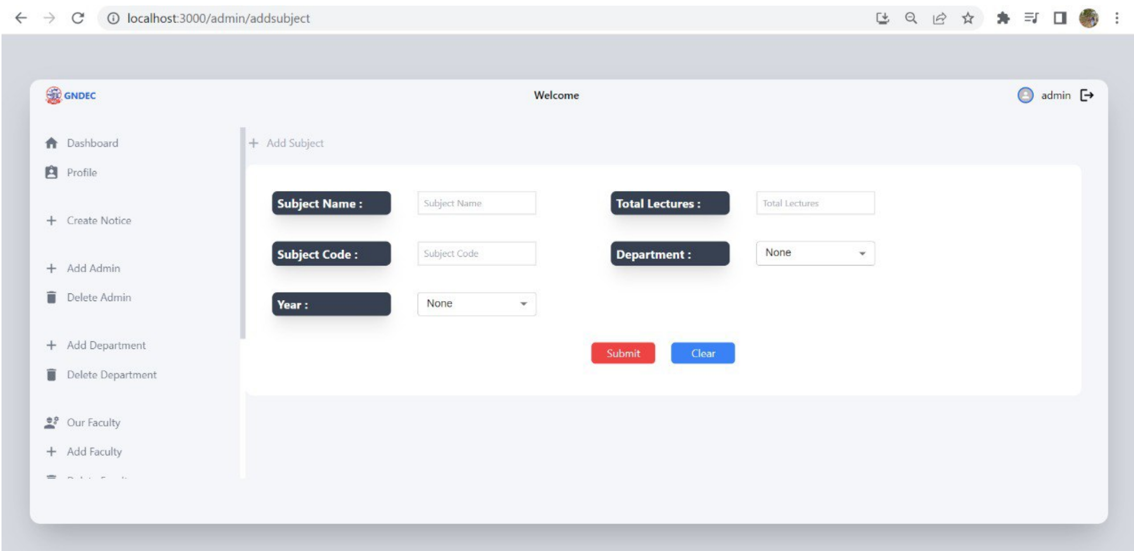

Figure 5.1.9: Adding subject list

5.2 Faculty Access

The faculty access provides various features to enhance the workflow oh the organizations.

Faculty can perform various tasks which can be done as listed below

5.2.1 Faculty login

Firstly user have to login by the admin username and the password to gain add the Faculty rights.

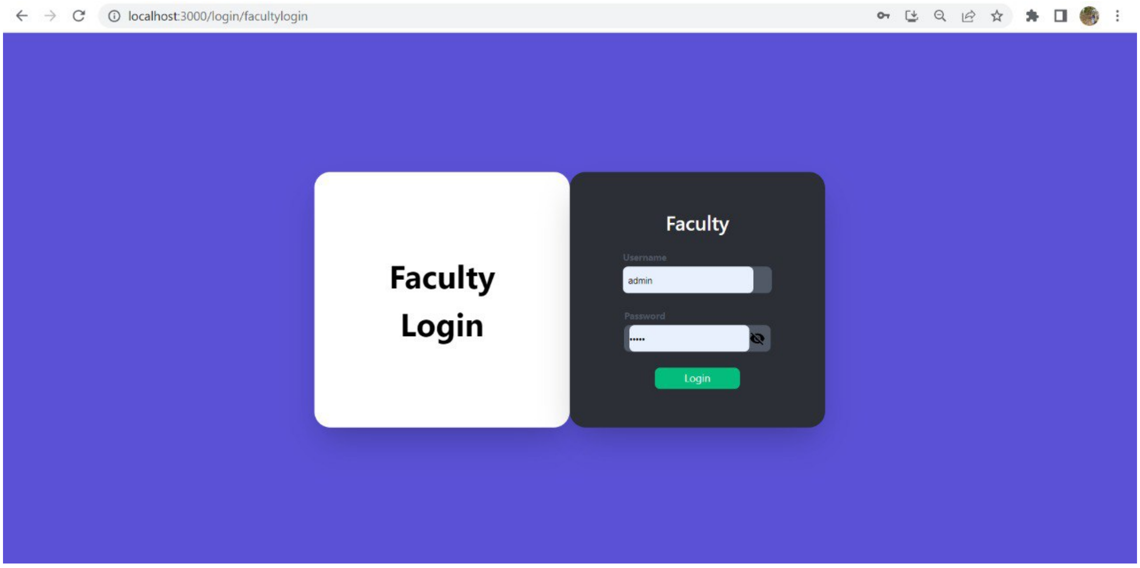

Figure 5.2.1: Faculty login page

5.2.2 Dashboard

1. The dashboard includes the basic details of the number of classes, no of students and the number of tests.

2. This also provides a calendar from where you can access the latest and the previous notices.

3. This dashboard provides an access to all the route pages linked to the dashboard

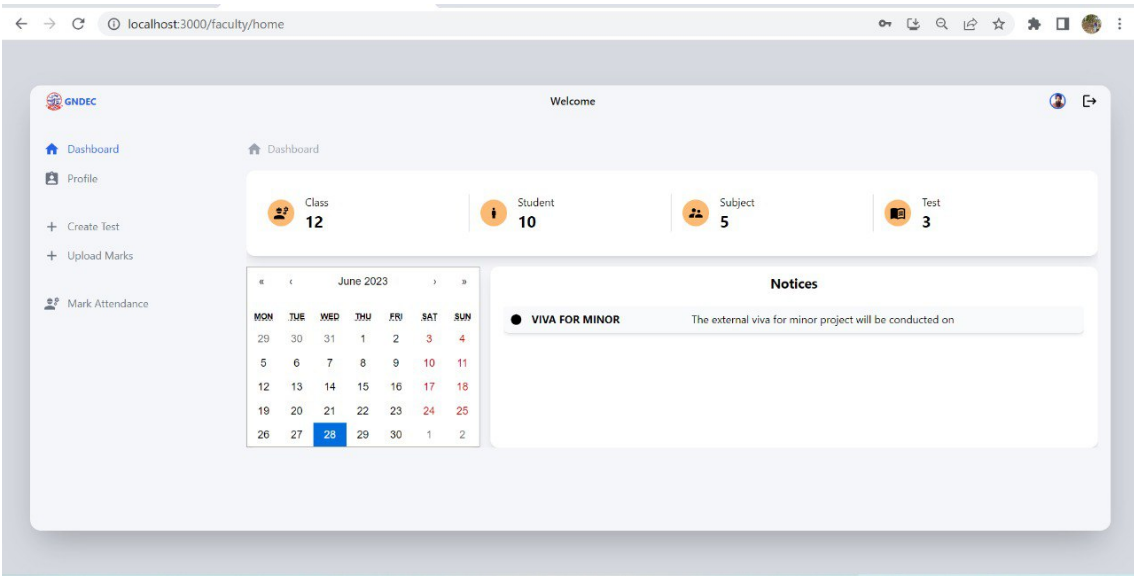

Figure 5.2.2: Faculty dashboard

5.2.3 Faculty profile

This page provides and access to the personal information of the faculty which includes name, phone number, profile picture, department details, username, joining year.

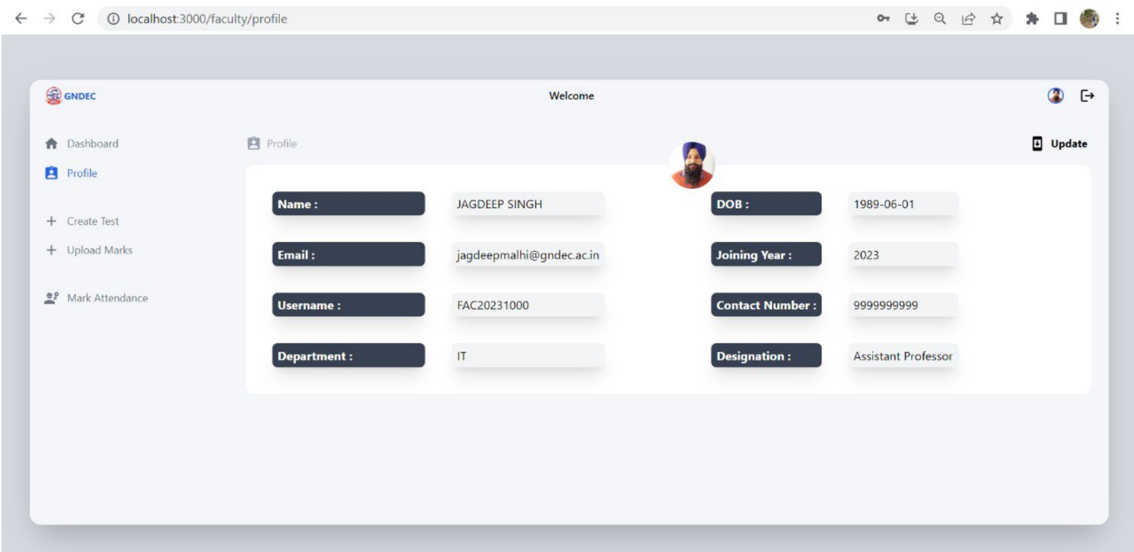

Figure 5.2.3: Faculty Profile

5.2.4 Create test

This page provides a platform to create a test and check for the students who are enrolled in it.

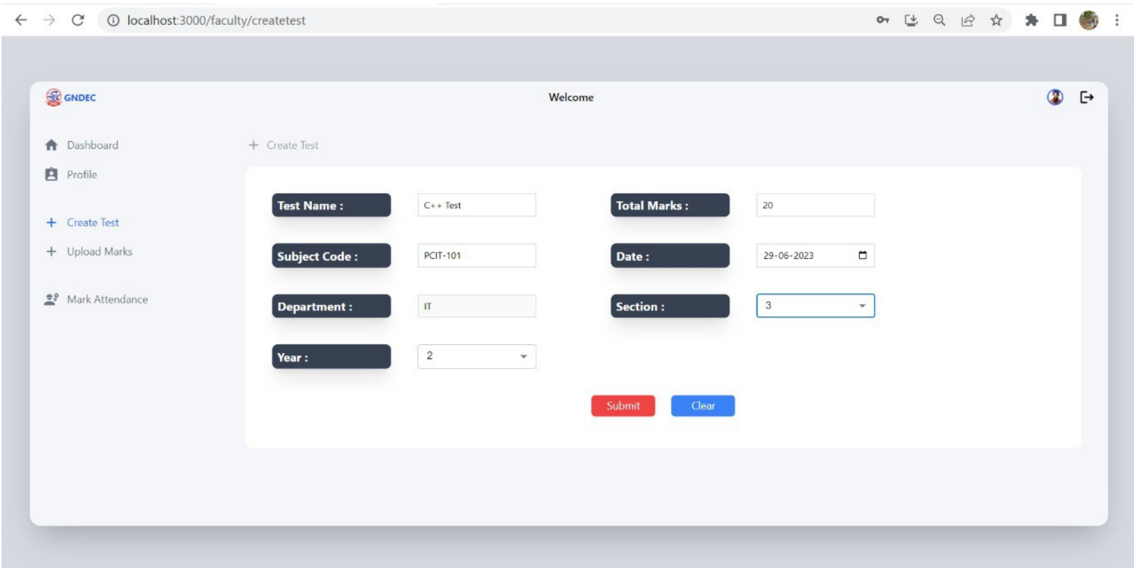

Figure 5.2.4: Create Test

5.2.5 Upload Marks

Faculty can update the marks of the given students which are enrolled in the given test generated by the faculty itself.

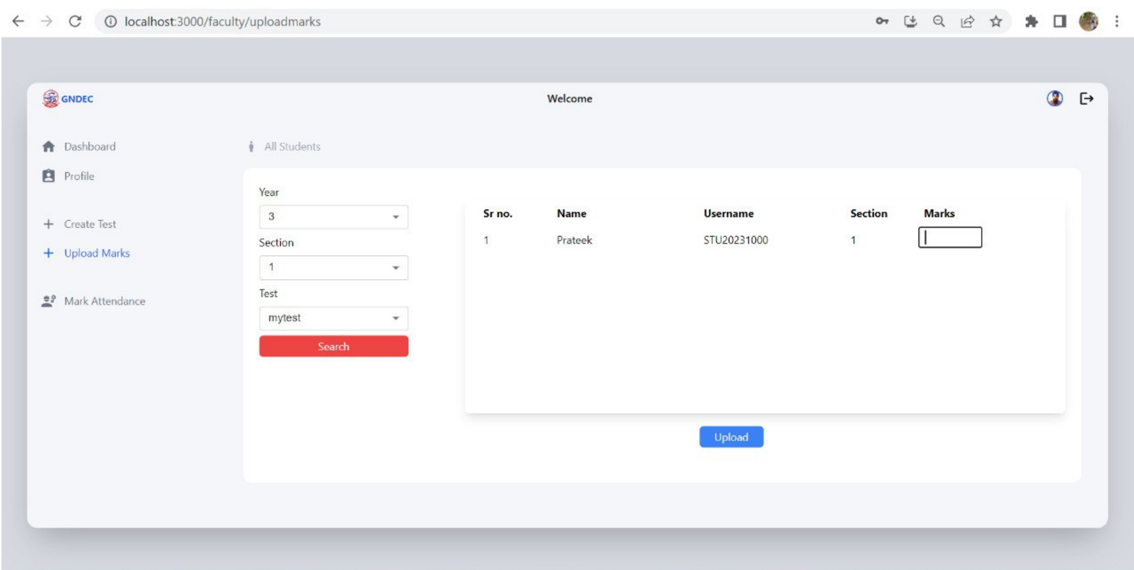

Figure 5.2.5: Uploading Marks

5.3 MongoDb

Mongo is a database server which is used to stre all the information provided by the faculty as well as the administrator.

5.3.1 Student Database

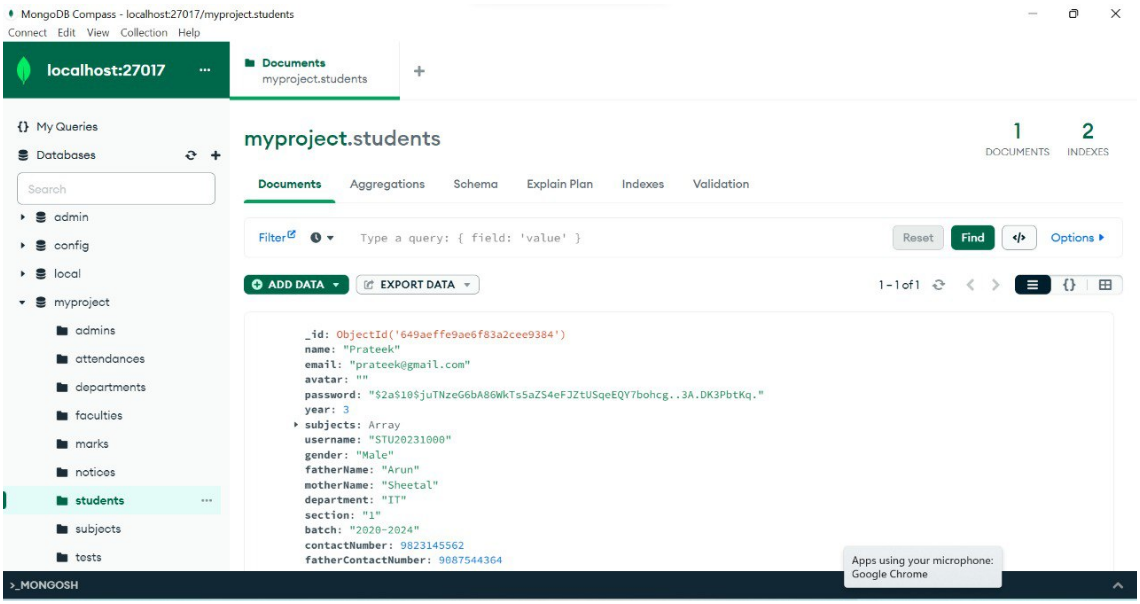

Figure 5.3.1: Student Database

5.3.2 Subject Database

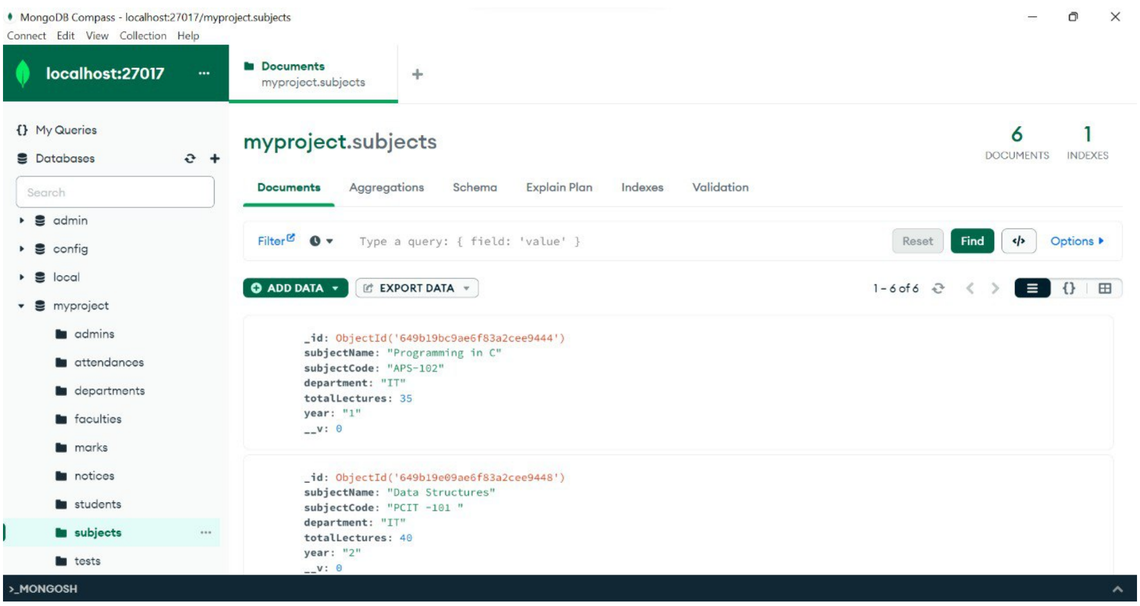

Figure 5.3.2: Subject Database

5.3.3 Admin Database

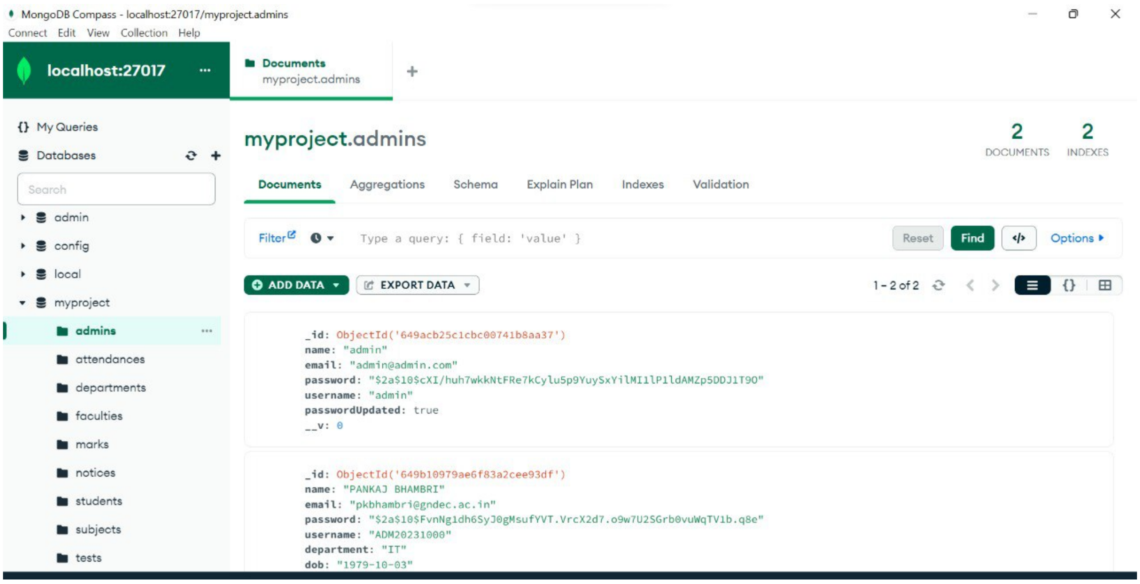

Figure 5.3.3: Admin Database

6 Conclusion and Future Scope

6.1 Conclusion

In conclusion, a Faculty Record Management System is a valuable tool for educational institutions to efficiently manage and organize faculty records. By implementing such a system, institutions can streamline administrative processes, improve data accuracy.

The objectives of a Faculty Record Management System revolve around centralizing and automating tasks related to faculty information, performance evaluation. By achieving these objectives, institutions can experience benefits such as increased productivity, time savings, improved data integrity, and enhanced decision-making.

The development of a Faculty Record Management System follows a structured methodology, which can include steps like requirements gathering, system design, iterative development, testing, and user feedback. Selecting an appropriate software development life cycle (SDLC) model, such as Agile or Waterfall, ensures an organized and efficient development process.

The ultimate result of a Faculty Record Management System is the delivery of a user-friendly, efficient, and reliable solution that meets the specific needs of the institution. This system improves overall efficiency, data accuracy, and collaboration while saving time and reducing administrative burdens. The benefits include streamlined faculty record management, improved productivity, effective performance evaluation, accurate leave management, enhanced reporting and analytics, and increased compliance and security.

6.2 Future Scope

The future scope of a Faculty Record Management System is promising, with several potential areas for expansion and improvement. Here are some key areas of future development and enhancement:

Mobile Accessibility: As mobile devices become increasingly prevalent, providing mobile accessibility to the Faculty Record Management System can offer convenience and flexibility to users. Developing mobile applications or responsive web interfaces can allow faculty members and administrators to access and manage records on-the-go.

Integration with Learning Management Systems (LMS): Integrating the Faculty Record Management System with existing Learning Management Systems can enable seamless data exchange and integration. This integration can enhance the overall learning experience by connecting faculty records with course management, grading, and student information systems.

Advanced Analytics and Reporting: Expanding the system's analytical capabilities can provide valuable insights into faculty performance, productivity, and engagement. Implementing advanced reporting features, data visualization, and predictive analytics can help institutions make data-driven decisions and identify trends and patterns.

Automation of Workflows: Further automating workflows within the system can improve process efficiency and reduce manual effort. Automating tasks such as performance evaluations, leave approvals, and document generation can save time and enhance accuracy.

7 References

[1] Sommerville, Software Engineering, 10th ed. Pearson Education, 2016.

[2] Borislav Hadzhiev, "BobbyHadz," [Online]. Available: https://bobbyhadz.com/blog/react-onclick-redirect [Accessed: June 3, 2023].

[3] M. Sheetal Singh, Prakash Choudhary: https://www.scribbr.com/ieee/ieee-reference-page/ [Accessed:June 1 , 2023].

[4] React Router [Online] Available at: https://reactrouter.com/en/main
[Accessed: June 3 , 2023].