KIET DATA CENTER

A PROJECT REPORT

Submitted by

NISHANT SHARMA
2100290140098

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Sangeeta Arora
Dr. Shashank Bharadwaj
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions,
Delhi-NCR, Ghaziabad Uttar Pradesh-201206
(JUNE 2023)

CERTIFICATE

Certified that **Nishant Sharma 2100290140098,** has/have carried out the project work having "**KIET DATA CENTER"** for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Nishant Sharma - 2100290140098

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Sangeeta Arora
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner   **Signature of External Examiner**

Dr. Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

This project consists of managing the college and department databases to effectively manage its various departments and functions. This project aims to create such a system by designing and implementing a centralized database management system.

The database system will include various modules or components that will manage the different aspects of the institution, including student information, event management, account department, human resources, and placement management. The student information module will store and manage information related to student certificates, academic details, and other personal information.

The event management module will help organize and manage different types of events held by the college, including participation details of the students. The account department module will keep track of the financial details of the institution, including managing fees, salaries, and other financial transactions.

The human resources module will manage the personnel details of the staff members, including their roles, responsibilities, and performance details. The placement management module will track the students who have been placed in various organizations after completing their studies.

The database system will allow the college or university to streamline its operations, improve communication and collaboration, and make data-driven decisions. By centralizing data and providing a single platform to manage various departments, the system will reduce costs and improve overall efficiency.

ACKNOWLEDGEMENTS

LIST OF CONTENTS

List of Chapters

LIST OF FIGURES

Chapter 1

INTRODUCTION

The project at hand focuses on the efficient management of the college and department databases. In today's world, information technology plays a vital role in the smooth functioning of any organization, and educational institutions are no exception. The management of data related to academic details, student certificates, and various events is a critical aspect of any college's operations. This project aims to provide a comprehensive and efficient database management system that will enable the college to manage its various departments and functions effectively

This project aims to provide a solution to the challenges faced by colleges in managing their databases. The project will focus on the efficient management of college and department databases, enabling the institution to manage its various functions effectively. The system will store and organize data related to student certificates, academic details, and various events held by the college with participating students.

The system will also keep track of the account and human resources departments, ensuring that all operations are integrated seamlessly. Additionally, it will track students who have been placed in various organizations, allowing the college to monitor their progress after they leave the institution.

For a given registrar / staff / student (Technical / Non-technical) the Administrator creates login-id & password, using these registrar/ staff / students (Technical / Non-technical) can access the system to either upload or download some information from the database.

The front-end will be HTML pages with Java Script for client-side validation whereas all business logics will be in PHP reside at middle layer. And these layers will interact with third layer of database, which will be MySQL database. The web server will be Apache. To start working on this project environment required is a server having Apache as web server, MySQL as database and LAMP (Linux, Apache, MySQL and PHP) as development environment.

Overall, the implementation of a comprehensive and efficient database management system will help the college to manage its various departments and

functions more efficiently. This will ultimately enhance the institution's reputation and provide better educational opportunities to its students.

1.1 Project Description

The project aims to streamline college data management and enhance operational efficiency. By centralizing and digitizing the data, the system will provide easy access to accurate and up-to-date information, eliminating manual paperwork and reducing errors risk.

The database management system will facilitate the storage, retrieval, and updating of student-related information, including certificates, academic records, and event participation details. It will also include modules for managing the accounts department, enabling efficient financial record-keeping and tracking expenses and income. The system will further cater to the human resources department by storing employee records, managing leave requests, and automating performance evaluations.

1.2 Project Scope

The scope of this project is to develop and implement a robust and comprehensive database management system for the college. The system will cover a wide range of functionalities, including the management of student certificates, academic details, event information, account department records, human resources department records, and student placements in various organizations.

The project will involve a detailed analysis of the college's existing data management practices and the development of a customized database management system to meet the specific needs of the institution. The system will prioritize data security, implementing appropriate access controls and encryption measures to safeguard sensitive information.

It is important to note that the scope of the project does not include infrastructure or hardware development. The system will be designed to be compatible with the college's existing IT infrastructure and will leverage appropriate technologies and software tools for effective data management.

In summary, the project scope encompasses the development and implementation of a comprehensive KIET Data Center that covers student data, account department management, human resources management, and student placement tracking. The system will enhance data management processes, improve operational efficiency, and provide a robust platform for effective decision-making within the college.

1.3 Brief over view of the technology:

Hardware Interfaces Requirement

Processor: i3 11th Gen

RAM: 8 GB

Operating System: Windows 11

System Type: 64-bit operating system

Hard disk: 256 GB

Software Interfaces Requirement

Front end: HTML, CSS, JavaScript.

HTML: HTML is used to create and save web document. E.g., Notepad/Notepad++
CSS: (Cascading Style Sheets) Create attractive Layout
Bootstrap: responsive design mobile friendly site
JavaScript: it is a programming language, commonly use with web browsers.

Back end: PHP, MySQL

PHP: Hypertext Pre-processor (PHP) is a technology that allows software developers to create dynamically generated web pages, in HTML, XML, or other document types, as per client request. PHP is open-source software. (Note: Use PHP Version 8.0.25 Server)

MySQL: MySQL is a database, widely used for accessing querying, updating, and managing data in databases.

Software Requirement (any one)

WAMP Server
XAMPP Server (PHP version 8.0.25)
MAMP Server
LAMP Server
BROWSER (You may use any browser that is compatible with your operating system)

Chapter 2

Feasibility Study

2.1 Operational Study

This project will be developed on computer, so first check whether the technology is technically available or not. Now a day's computer interaction with any job becomes common for any kind of job or work.

And because of increasing usage of Computer, Computer is also available with a variety of hardware. Vendors can fulfill any type of hardware requirement. The whole project is developed by some special tools or by using languages and databases, which are also available in a variety.

Preliminary investigation of a system examines the feasibility of a system that is useful to an organization. It is the first phase of system development.

The main objective of this phase is to identify the current deficiencies in the user's environment and to determine which existing problem are going to be solve in proposed system and also which new function needs to be added in proposed system.

An important outcome of such preliminary investigation is to determine whether the system that will meet all needed requirements.

Thus, three tests are carried out on the system namely operation, technical and economical.

Any project is beneficial if and only satisfies the organization requirement. For any new system setup, it only meets to be communicated and work the other supporting system.

The new system meets all existing operations since it provides right information at a right time to the right user. A Leigh man can easily operate with the system.

2.2 Technical Feasibility

Technical Feasibility examines whether the technology needed is available and if it is available then it feasible to carry out all project activities.

The technical needs of a system include:
- The facility to produce outputs in each time.
- Ability to process large number of transactions at a particular speed.

➢ Giving response to users under certain conditions.

The technology needed for our system is mainly:
➢ Latest version of browsers.
➢ Any operating system.

These technologies are available which helps to carry out the system efficiently.

2.3 Economical Feasibility

Economic feasibility of a system examines whether the finance is available for implementing the new system and whether the money spent is recoverable the satisfaction.

The cost involves is in designing and developing a good investment for the organization.

Thus, hardware requirements used for proposed system are very standard. Moreover, by making use of proposed system to carry out the work speedily will increase and saves the valuable time of an organization.

In the proposed system the finance is highly required for the installation of the software's which can also be recovered by implementing a better system.
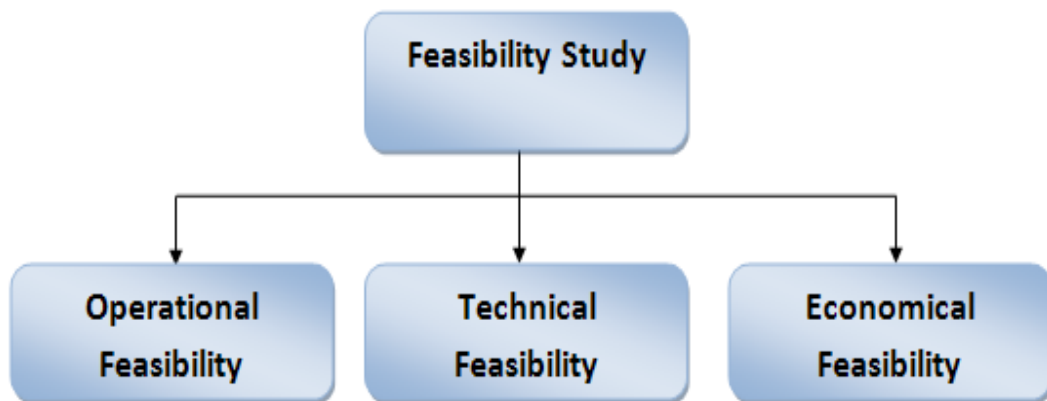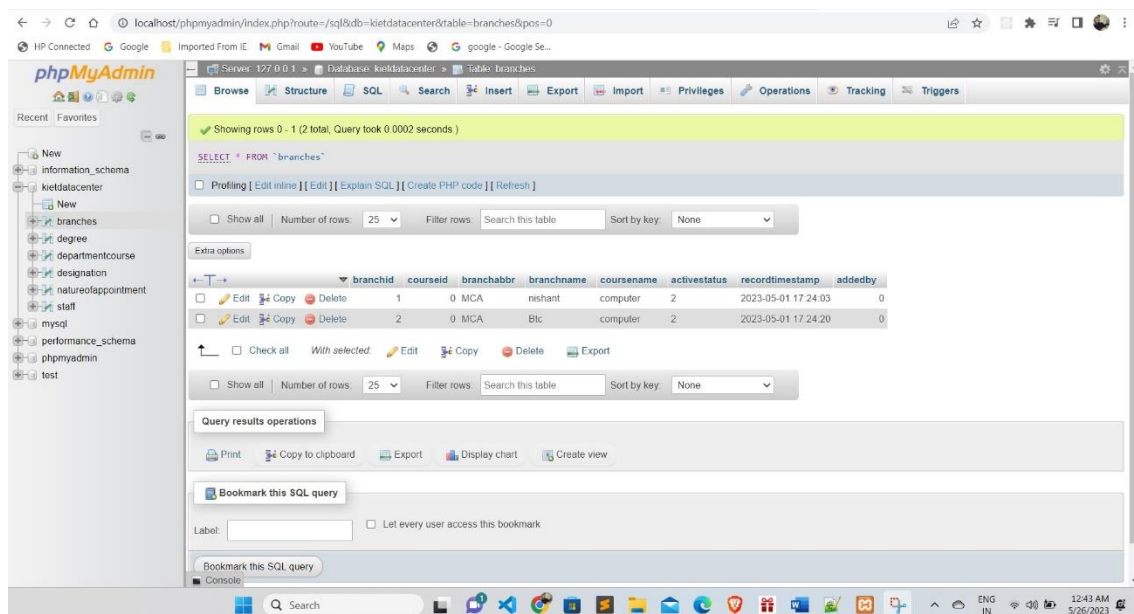
Fig: - 2.1(Economical Feasibility)

Chapter 3

Database Design

3.1 Database Table

Branches Info Table

Fig: - 3.1(Branches Info Table)

The Branches Info Table is a component of the college database management system that provides comprehensive information about the different branches or departments within the institution. It serves as a centralized repository of data related to various academic and administrative branches, facilitating easy access to branch-specific details. Here are some key elements typically included in the Branches Info Table:

Branch Table:

This column contains the names of all the branches or departments within the college. It includes disciplines such as Computer Science, Electrical Engineering, Business Administration, Mathematics, etc.

6
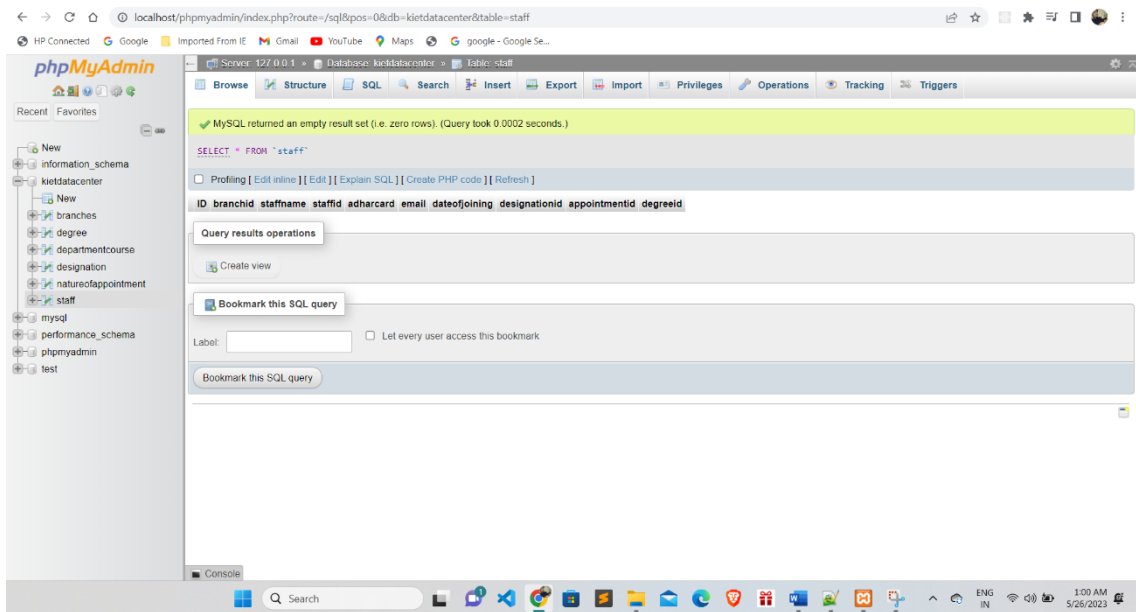
Staff Info Table

Fig: - 3.2(Staff Info Table)

The Staff Info Table is a component of the college database management system that stores comprehensive information about the staff members employed within the institution. It serves as a centralized repository of data related to the college's faculty, administrative staff, and other personnel. The Staff Info Table typically includes the following key elements:

Staff ID:

This column contains a unique identifier assigned to each staff member within the college. It serves as a primary key for record identification and retrieval purposes.

Staff Name:

This column stores the full name of each staff member. It includes their first name, last name, and any additional titles or designations.

Department:

This column indicates the department or branch in which the staff member is affiliated. It specifies the academic or administrative unit to which the staff member belongs.

Position:

This column specifies the job position or role held by the staff member. It includes positions such as professor, lecturer, administrative assistant, librarian, counselor, etc.

Qualifications:

This column provides information about the educational qualifications and professional certifications of the staff member. It includes details such as degrees, diplomas, specialized training, and areas of expertise.

The Staff Info Table serves as a comprehensive database of staff information, providing easy access to key details about each staff member within the college. It supports efficient staff management, facilitates communication, and enables effective decision-making regarding staffing requirements, assignments, and performance evaluation.

Degree Info Table
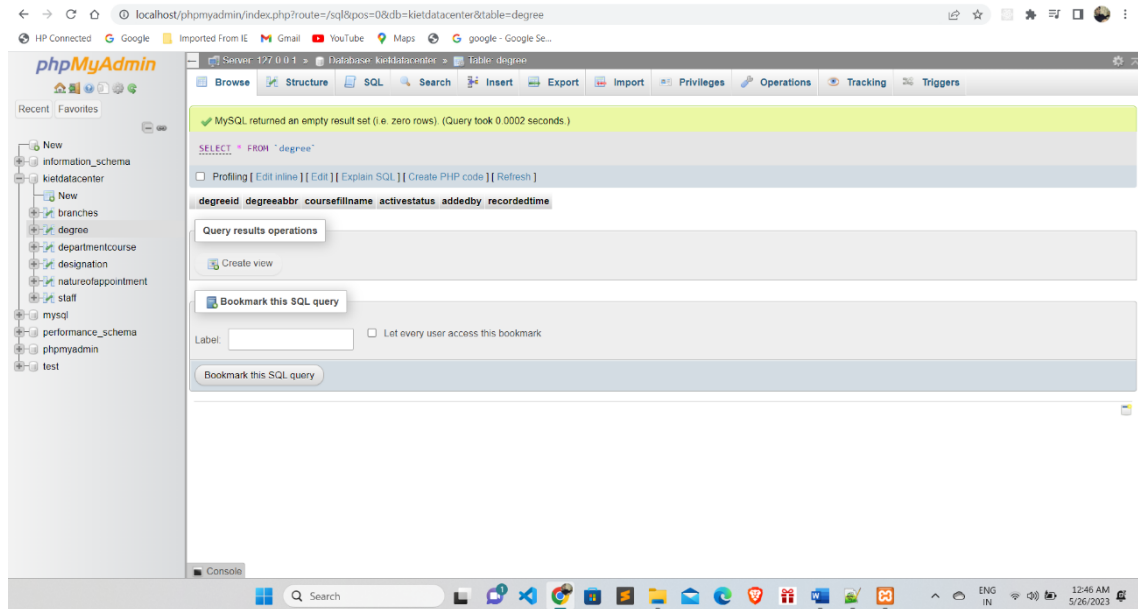
Fig: - 3.3(Degree Info Table)

Degree Info Table:

The Degree Info Table is a component of the college database management system that stores essential information about the different academic degrees offered by the institution. It serves as a centralized repository of data related to the various educational programs available to students. The Degree Info Table typically includes the following key elements:

Degree ID:

This column contains a unique identifier assigned to each degree program offered by the college. It serves as a primary key for record identification and retrieval purposes.

Degree Name:

This column stores the name of each degree program. It includes the official title of the degree, such as Bachelor of Science, Master of Arts, Doctor of Philosophy, etc.

The Degree Info Table serves as a valuable resource for students, faculty, and administrators to access comprehensive information about the various academic degrees offered by the college. It supports academic planning, assists in program evaluation and accreditation processes, and helps students make informed decisions regarding their educational pursuits and career paths.

Department course INFO Table
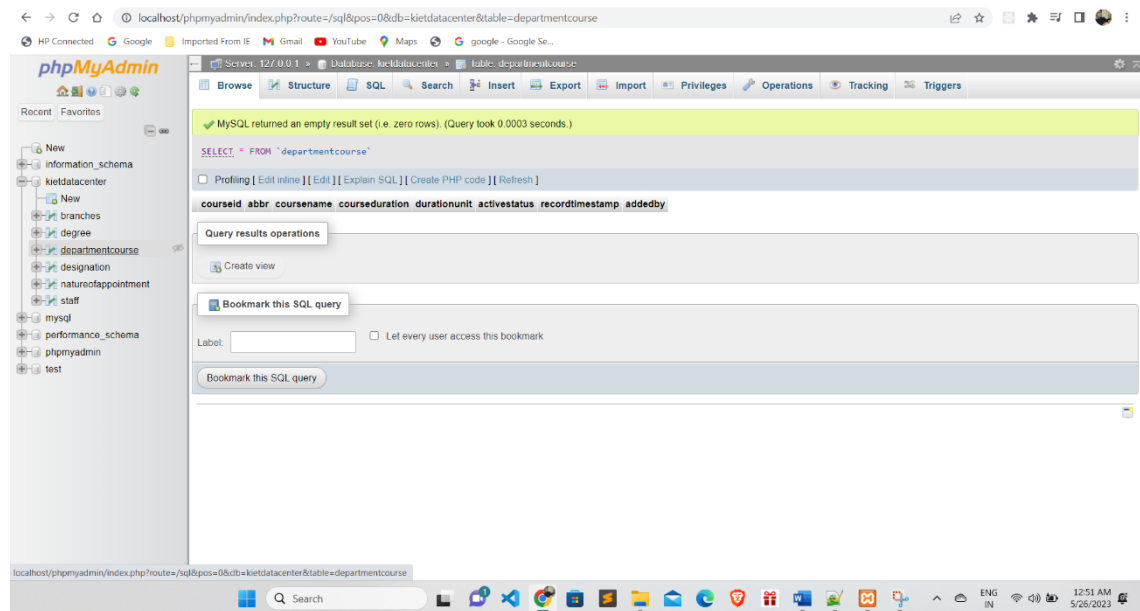
Fig: - 3.4(Department course INFO Table)

The Department Course Info Table is a component of the college database management system that stores comprehensive information about the courses offered by each department within the institution. It serves as a centralized repository of data related to the academic curriculum and course offerings in different disciplines. The Department Course Info Table typically includes the following key elements:

Course ID:

This column contains a unique identifier assigned to each course offered by the department. It serves as a primary key for record identification and retrieval purposes.

Course Name:

This column stores the name of each course. It includes the official title of the course, such as Introduction to Biology, Financial Accounting, Algorithms and Data Structures, etc.

Department:

This column indicates the department or branch offering the course. It specifies the academic unit responsible for teaching and managing the course.

Course Description:

This column provides a brief description of the course, outlining its objectives, content, and learning outcomes. It offers an overview of the topics covered and the skills and knowledge students can expect to gain from the course.

Course Syllabus:

This column may include a link or reference to the detailed course syllabus, which provides comprehensive information about the course objectives, learning outcomes, assessment methods, and a week-by-week breakdown of topics covered.

The Department Course Info Table serves as a valuable resource for students, faculty, and administrators to access comprehensive information about the courses offered by each department. It supports academic planning, assists in course registration and scheduling processes, and helps individuals make informed decisions regarding their course selections and academic pathways within the college.

Designation Info Table
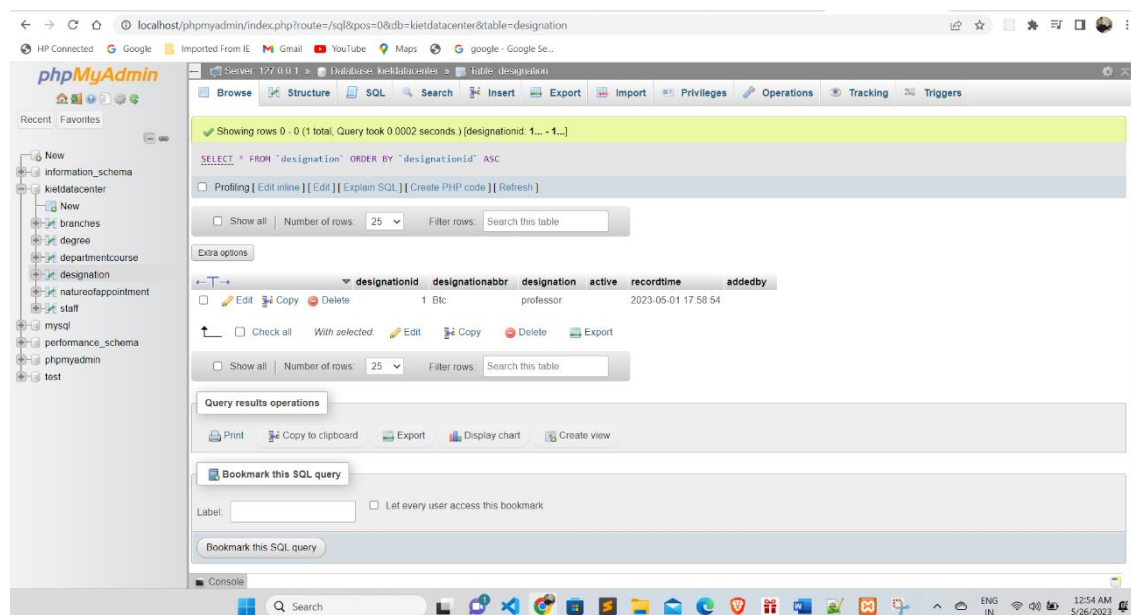
Fig: - 3.5(Designation Info Table)

The Designation Info Table is a component of the college database management system that stores information about different job designations or positions within the institution. It serves as a centralized repository of data related to the various roles and responsibilities assigned to staff members. The Designation Info Table typically includes the following key elements:

Designation ID:

This column contains a unique identifier assigned to each job designation within the college. It serves as a primary key for record identification and retrieval purposes.

Designation Name:

This column stores the name or title of each job designation. It includes positions such as Professor, Assistant Professor, Administrator, Librarian, Clerk, etc.

Department:

This column indicates the department or branch to which the designation belongs. It specifies the academic or administrative unit within the college where the position is assigned.

Qualifications:

This column specifies the educational qualifications and professional requirements for the job designation. It includes details such as degrees, certifications, specialized skills, or years of experience needed for the position.

The Designation Info Table serves as a valuable resource for the college administration, human resources department, and employees to access information about different job designations within the institution. It supports workforce planning, recruitment, and employee management processes, and helps individuals understand the roles and responsibilities associated with various positions.

Nature of appointment Info Table
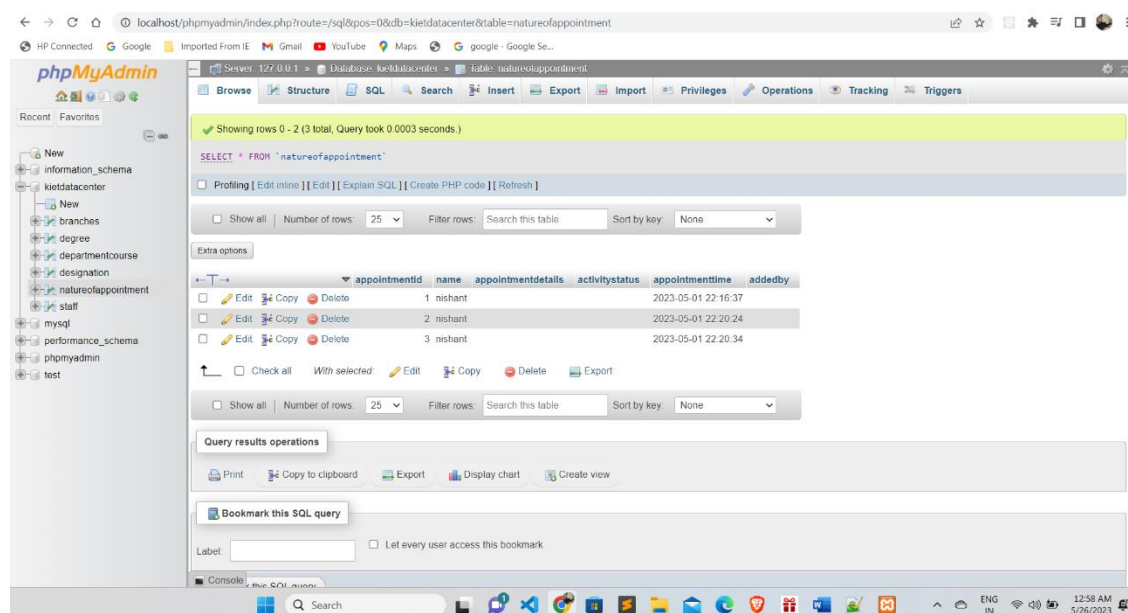
Fig: - 3.6(Nature of appointment Info Table)

The Nature of Appointment Info Table is a component of the college database management system that stores information about the different types of appointments or employment statuses for staff members within the institution. It serves as a centralized

repository of data related to the terms and conditions of employment. The Nature of Appointment Info Table typically includes the following key elements:

Appointment ID:

This column contains a unique identifier assigned to each nature of appointment within the college. It serves as a primary key for record identification and retrieval purposes.

Employment Status:

This column specifies the employment status associated with the appointment type. It indicates whether the staff member is considered full-time, part-time, or casual.

Terms and Conditions:

This column highlights any specific terms and conditions associated with the appointment type. It may include contractual obligations, confidentiality agreements, code of conduct, or other contractual provisions.

Applicable Policies:

This column references the relevant policies, rules, or regulations governing the nature of appointment. It provides information about the guidelines or frameworks that shape the employment terms for staff members.

The Nature of Appointment Info Table serves as a valuable resource for the human resources department, staff members, and administration to access information about the different types of appointments within the institution. It supports the management of staff contracts, benefits administration, and compliance with employment regulations.

3.2 Use Case Diagram

- Use Case Diagram for User and Authorization
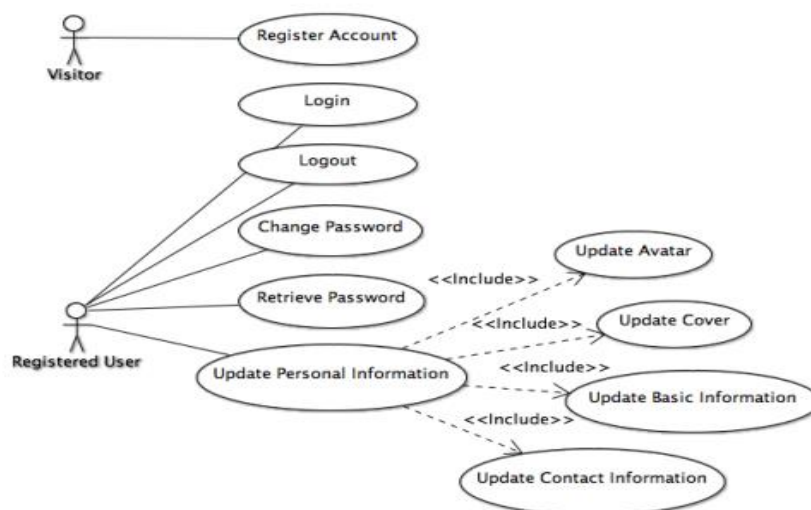
Fig: - 3.1(Use Case Diagram)

3.3 Sequence Diagram

- For Login
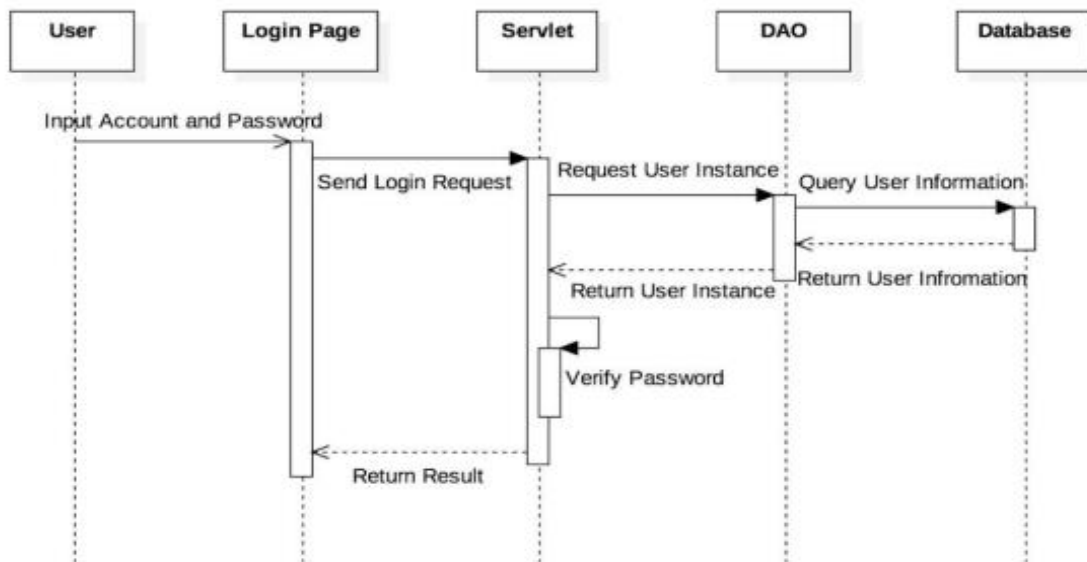
Fig: - 3.2(Sequence Diagram Login)

The basic flow of user login is as follows:

1. The user accesses the login page and input his account and password.
2. The login page sends the login request together with the user's input information to the Login Servlet
3. Login Servlet calls the user DAO (data access object) to perform query operation on the database, and returns the instance object of user
. 4. Login by User Email method Compare and verify the input password with the correct password (saved in the instance object).
5. If the password is wrong, login fails and then returns to the login page,   reporting an error. Otherwise, login succeeds and then jumps to the user's personal homepage.

Sequence Diagram for Personal Information

The basic flow of displaying a user's representative photos is as follows:
1. The user visits the personal homepage for browsing.
2.Personal homepage requests the Homepage Servlet for relevant information about the current user's representative photos.
3. Homepage Servlet calls the user data access object for query operation and returns a collection of picture objects containing information of the user's representative photos.
 4. The Personal homepage requests to the server, accesses picture resources and displays representative photos using the URLs.
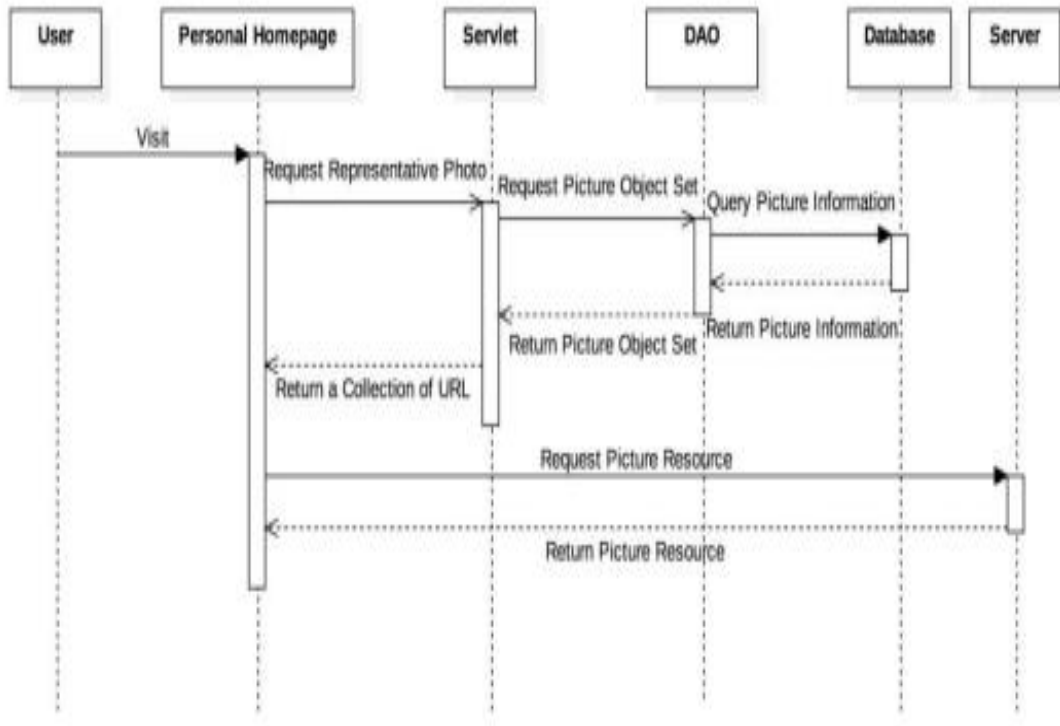
Fig: - 3.3(Sequence Diagram)

3.4 Data Flow Diagram (DFD)

Level 0 DFD ADMIN
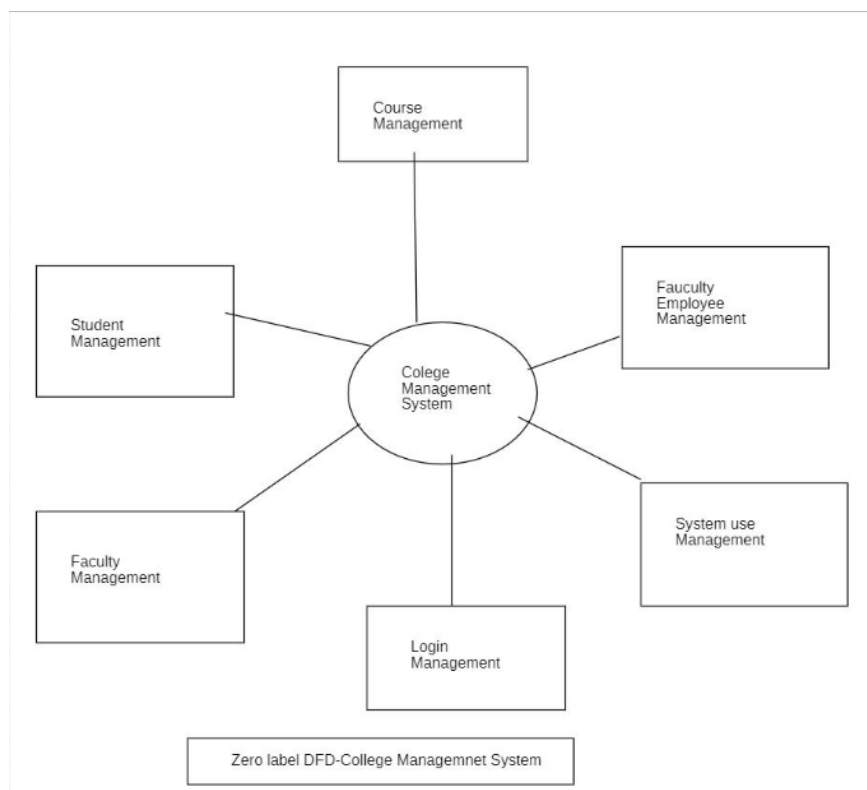
Fig: - 3.4(Level 0 DFD ADMIN)

The DFD-ADMIN (Data Flow Diagram - ADMIN) is a critical component of our project, designed to illustrate the flow of data and processes within the administrative functions of our system. It provides a visual representation of how information is captured, processed, and stored within the administrative module of our project.

The DFD-ADMIN helps to provide a high-level understanding of how data flows and processes are interconnected within the administrative module of our project. It aids in identifying potential bottlenecks, data dependencies, and areas for improvement within the administrative functions.

By visually representing the data flow and processes, the DFD-ADMIN assists in communicating the overall architecture and functionality of the administrative module to stakeholders, project team members, and developers. It serves as a blueprint for designing and implementing the administrative functionalities effectively and ensures a clear understanding of how the system operates.

Level 1-DFD USER
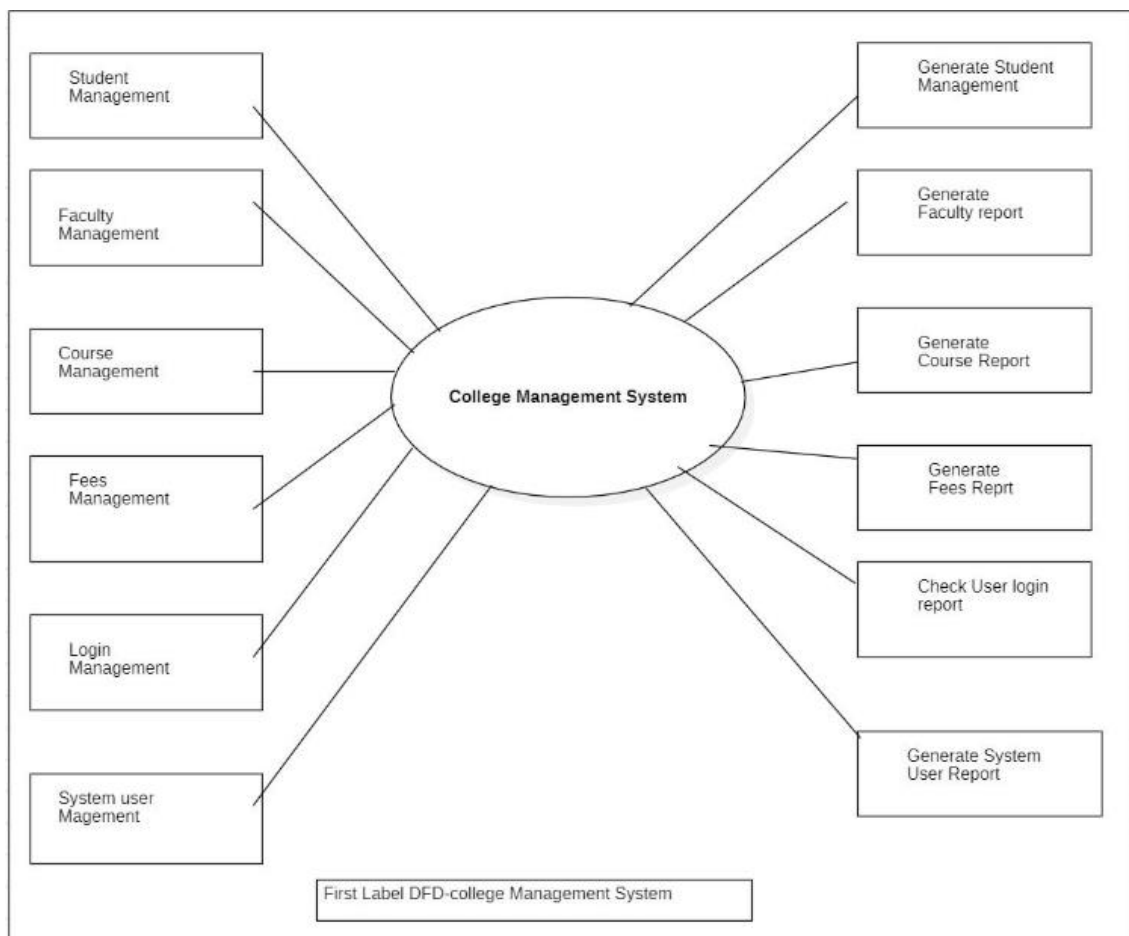
Fig: - 3.5(Level 1-DFD USER)

The DFD-USER (Data Flow Diagram - USER) is an important component of our project, designed to depict the flow of data and processes related to user interactions

within the system. It provides a visual representation of how data is exchanged between users and the various components of our project.

The DFD-USER provides a visual representation of how users interact with the system, illustrating the flow of data and processes involved in their interactions. It helps in understanding the overall architecture and functionality from a user's perspective.

By using the DFD-USER, stakeholders, project team members, and developers can gain a clear understanding of how data flows and processes are interconnected within the user interactions. It assists in identifying user requirements, potential bottlenecks, and areas for improvement in the user experience.

3.5 E-R Diagram

Fig: - 3.6(E-R Diagram)

Chapter 4

Modules Description

4.1 Login Module
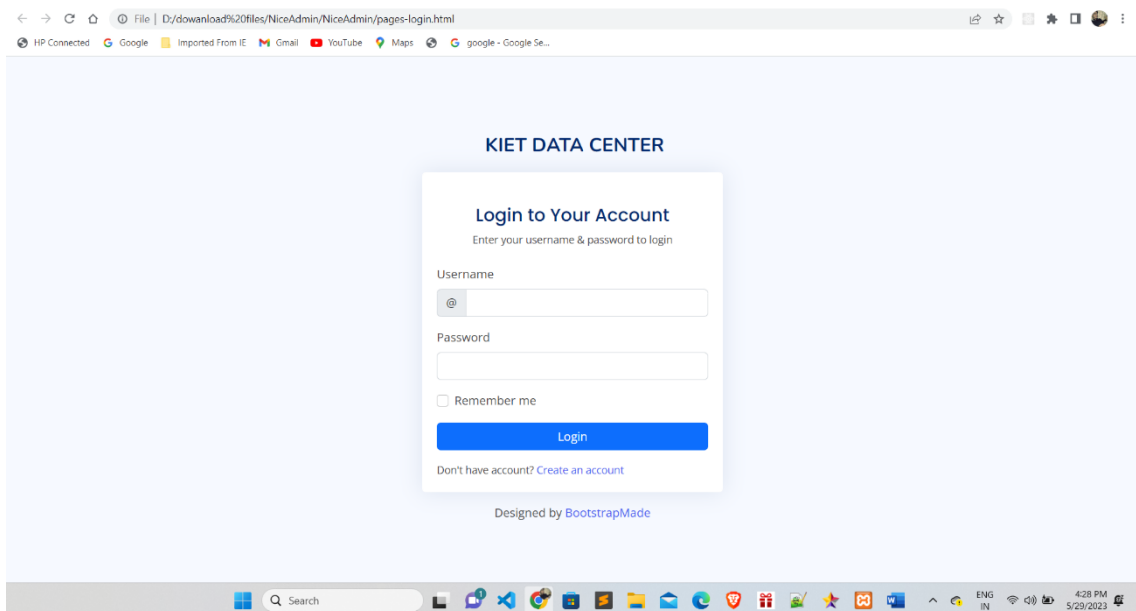
Admin can login/logout of the system.

Fig: - 4.1(Login Module)

Our Login/Signup Page Module is an essential component of our project, designed to provide a seamless and secure user authentication experience. This module allows users to create new accounts by registering with their personal information and credentials or log in using their existing account details.

The module features a user-friendly interface with clear and intuitive design elements, ensuring that users can easily navigate through the login/signup process. It includes fields for entering email addresses, passwords, and other necessary information, with appropriate validation to ensure data accuracy and security.

4.2 Student Management Module.

The Student Management Module is a key component of the comprehensive database management system designed for colleges. It focuses on managing student-related information and provides a range of features and functionalities to streamline student administration processes. This module aims to enhance the efficiency, accuracy, and accessibility of student data, facilitating effective student management throughout their academic journey. Here are some key features of the Student Management Module:

Student Profile Management:
The module allows colleges to create and maintain comprehensive student profiles that include personal information, contact details, academic records, attendance, and disciplinary history. It provides a centralized repository for storing and updating student data, ensuring easy access and retrieval whenever required.

Enrollment and Registration:
The module facilitates the enrollment and registration process for new and existing students. It allows colleges to collect and store essential information such as course preferences, program selection, and enrollment documentation. Automated workflows and notifications streamline the enrollment process, reducing paperwork and manual errors.

Academic Progress Monitoring:
With the Student Management Module, colleges can track and monitor students' academic progress effectively. It enables the recording of grades, transcripts, and course completion data. The module also supports generating progress reports and academic transcripts, providing students with a comprehensive overview of their academic performance.

The Student Management Module plays a pivotal role in optimizing student administration processes within colleges. By leveraging the features and functionalities of this module, colleges can improve student data management, enhance communication and collaboration, monitor academic progress, and provide comprehensive support services. Ultimately, this module contributes to a positive and enriching student experience, empowering them to succeed academically and personally.

4.3 Faculty and Staff Management Module

The Faculty and Staff Management Module is a critical component of the comprehensive database management system designed for colleges. It is designed to streamline and optimize the management of faculty and staff information, ensuring efficient administration of the college's workforce. This module offers a range of features and functionalities to simplify the processes related to faculty and staff management. Here are some key features of the Faculty and Staff Management Module:

Personnel Information Management:
The module allows colleges to maintain a centralized repository of faculty and staff information. It includes personal details, contact information, employment history, qualifications, certifications, and performance evaluations. It provides easy access to accurate and up-to-date information about faculty and staff members.

Recruitment and Hiring:
The module supports the recruitment and hiring processes within the college. It facilitates the creation and management of job postings, applicant profiles, and interview schedules. It enables the tracking of candidate evaluations, document verification, and the generation of appointment letters. The module streamlines the entire recruitment cycle, saving time and effort for the HR department.

Leave and Attendance Management:
Efficient leave and attendance management is crucial for colleges. This module allows faculty and staff to apply for leave online, while administrators can approve or reject leave requests within the system. It tracks leave balances, calculates attendance records, and generates reports. This ensures accurate and automated leave management, reducing manual paperwork and errors.

The Faculty and Staff Management Module streamlines administrative processes related to faculty and staff, improves data accuracy, enhances communication and collaboration, and promotes compliance with employment regulations. By leveraging the features and functionalities of this module, colleges can effectively manage their workforce, promote employee engagement, and create a positive work environment for faculty and staff members.

4.4 Human resources:

The Human Resources (HR) Management Module is a critical component of the comprehensive database management system designed for colleges. It focuses on managing the various aspects of human resource functions within the institution. This module offers a range of features and functionalities to streamline HR processes, automate administrative tasks, and ensure efficient management of the college's workforce.

Employee Information Management:

The module provides a centralized database to store and manage employee information. It includes features to record and update employee profiles, personal details, contact information, employment history, and qualifications. This ensures accurate and up-to-date employee records, simplifying HR administration.

Recruitment and Hiring:

The module supports the recruitment and hiring process within the college. It includes features for posting job vacancies, receiving and managing applications, conducting interviews, and making job offers. It streamlines the hiring process, helps in identifying qualified candidates, and facilitates efficient onboarding.

Attendance and Leave Management:

The module enables the tracking and management of employee attendance and leave. It allows employees to record their attendance, request leave, and view their leave balances. It also provides administrators with tools to monitor attendance, manage leave approvals, and generate attendance reports. This helps in ensuring workforce attendance and managing leave policies effectively.

Employee Self-Service:

The module offers employee self-service capabilities, allowing employees to access and manage their personal information, view their attendance records, request leave, and access training resources. This empowers employees to take ownership of their HR-related activities, reducing administrative workload for HR staff.

The HR Management Module enhances the efficiency and effectiveness of human resource management within the college. By leveraging its features and functionalities, colleges can streamline HR processes, improve employee engagement, and ensure compliance with HR policies and regulations. Ultimately, this module contributes to the development and well-being of the college's workforce.

4.5 Financial Management Module

The Financial Management Module is a vital component of the comprehensive database management system designed for colleges. It focuses on managing the financial aspects of the institution, including budgeting, accounting, and financial reporting. This module offers a range of features and functionalities to streamline financial processes, ensure financial accuracy, and support informed decision-making. Here are some key features of the Financial Management Module:

Budget Planning and Management:

The module allows colleges to create, manage, and monitor budgets for various departments, programs, and projects. It facilitates the allocation of funds, tracking of

expenses, and comparison of actual spending against budgeted amounts. This helps colleges maintain financial control and make informed decisions based on budget performance.

Financial Reporting and Analysis:
The module provides robust reporting capabilities to generate financial statements, including income statements, balance sheets, and cash flow statements. It also offers customizable reports to analyze financial data, track key performance indicators, and monitor financial health. These reports assist in decision-making and help colleges meet financial reporting requirements.

Audit and Compliance:
The module assists colleges in maintaining compliance with financial regulations and auditing requirements. It provides features for documenting and tracking financial transactions, facilitating internal and external audits, and ensuring adherence to financial policies and procedures. This promotes transparency and accountability in financial management.

The Financial Management Module empowers colleges to effectively manage their financial resources, maintain financial control, and make informed decisions based on accurate and up-to-date financial information. By leveraging the features and functionalities of this module, colleges can streamline financial processes, enhance financial transparency, and ensure compliance with financial regulations. Ultimately, this module contributes to the financial stability and success of the institution.

4.6 Placement Management Module.

The Placement Management Module is an essential component of the comprehensive database management system designed for colleges. It aims to streamline and enhance the management of student placements in various organizations. This module offers a range of features and functionalities to facilitate the placement process, automate administrative tasks, and ensure successful placements. Here are some key features of the Placement Management Module:

Placement Registration:
The module enables students to register for placement opportunities offered by different organizations. It provides a user-friendly interface where students can submit their resumes, academic transcripts, and other required documents. This simplifies the registration process and ensures that students' information is easily accessible for placement activities.

Job Posting and Management:

The module allows organizations to post job vacancies and internship opportunities for students. It provides a centralized platform where organizations can upload job descriptions, qualifications, and application deadlines. This facilitates easy access to placement opportunities for students and helps organizations attract qualified candidates.

Placement Analytics and Reporting:

The module offers analytics and reporting capabilities to track placement trends, measure success rates, and gather valuable insights. It generates reports on the number of placements, average salary packages, placement ratios by department or program, and feedback from organizations and students. This data helps colleges assess the effectiveness of their placement efforts and make data-driven decisions for continuous improvement.

The Placement Management Module simplifies and automates the entire placement process, from registration and matching to selection and acceptance. By leveraging the features and functionalities of this module, colleges can efficiently manage student placements, enhance collaboration with organizations, and ensure successful career outcomes for their students.

4.7 Resource Management Module

The Resource Management Module is a vital component of the comprehensive database management system designed for colleges. It focuses on efficient management and allocation of resources within the institution. This module offers a range of features and functionalities to streamline resource utilization, optimize resource allocation, and ensure effective resource management.

Resource Inventory:

The module provides a centralized database to manage and track various resources available in the college. It includes features to categorize and record different types of resources such as equipment, facilities, materials, and supplies. This ensures a comprehensive inventory of resources, making it easier to track availability and utilization.

Resource Usage Tracking:

The module enables tracking and monitoring of resource usage. It records details such as the duration of resource usage, the purpose of usage, and the users involved. This helps in analyzing resource utilization patterns, identifying areas of high demand, and optimizing resource allocation based on usage data.

Resource Maintenance and Service Requests:

The module supports the management of resource maintenance and service requests. It allows users to report maintenance issues, request repairs or replacements, and track the status of service requests. It helps in ensuring that resources are well-maintained, minimizing downtime and disruptions caused by faulty equipment or facilities.

Resource Allocation and Planning:

The module assists in resource allocation and planning for various activities within the college. It allows administrators to allocate resources based on specific requirements, such as events, classes, research projects, or departmental needs. It provides tools to forecast resource needs, allocate resources efficiently, and avoid resource shortages or excesses.

The Resource Management Module enhances the efficient utilization and allocation of resources within the college. By leveraging its features and functionalities, colleges can optimize resource utilization, minimize wastage, and ensure that resources are allocated effectively to support various activities and functions. Ultimately, this module contributes to cost savings, improved productivity, and better resource management within the institution.

CHAPTER-5

Coding

5.1 Login Module
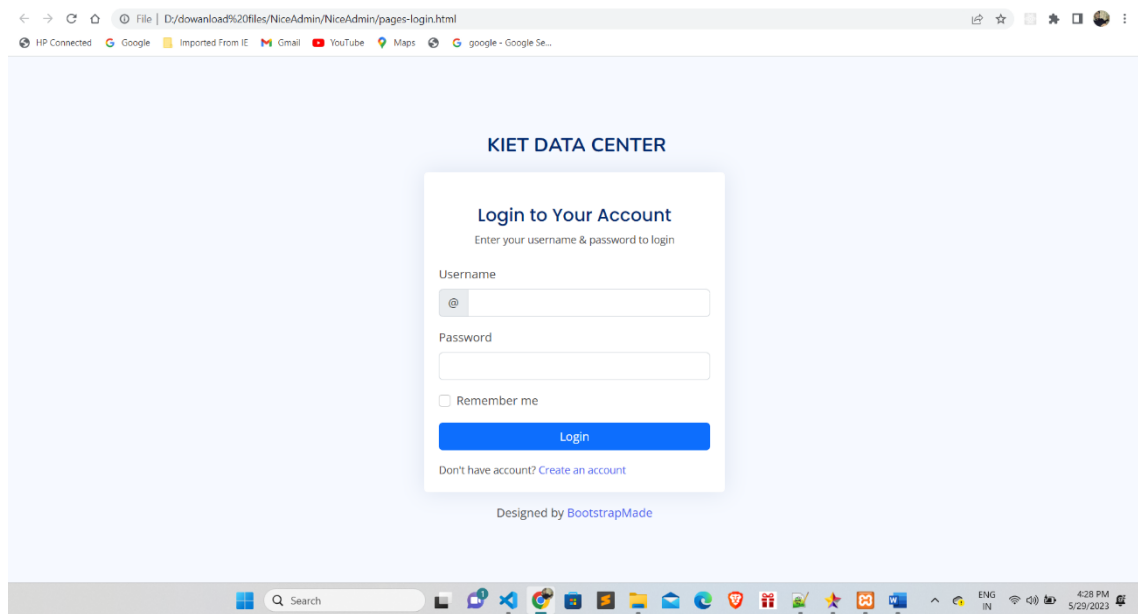
Fig :- 5.1(Login Module)

```
<!DOCTYPE html>
<html lang="en">

<head>
 <meta charset="utf-8">
 <meta content="width=device-width, initial-scale=1.0" name="viewport">

 <title>Pages / Login - NiceAdmin Bootstrap Template</title>
 <meta content="" name="description">
 <meta content="" name="keywords">

 <!-- Favicons -->
 <link href="assets/img/favicon.png" rel="icon">
 <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

 <!-- Google Fonts -->
 <link href="https://fonts.gstatic.com" rel="preconnect">
```

```
    <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700
i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,70
0,700i" rel="stylesheet">

    <!-- Vendor CSS Files -->
    <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
    <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
    <link href="assets/vendor/quill/quill.snow.css" rel="stylesheet">
    <link href="assets/vendor/quill/quill.bubble.css" rel="stylesheet">
    <link href="assets/vendor/remixicon/remixicon.css" rel="stylesheet">
    <link href="assets/vendor/simple-datatables/style.css" rel="stylesheet">

    <!-- Template Main CSS File -->
    <link href="assets/css/style.css" rel="stylesheet">

    <!-- =======================================================
    * Template Name: NiceAdmin - v2.4.1
    * Template URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/
    * Author: BootstrapMade.com
    * License: https://bootstrapmade.com/license/
    ======================================================== -->
</head>

<body>

  <main>
    <div class="container">

      <section class="section register min-vh-100 d-flex flex-column align-items-center justify-
content-center py-4">
        <div class="container">
          <div class="row justify-content-center">
            <div class="col-lg-4 col-md-6 d-flex flex-column align-items-center justify-content-
center">

              <div class="d-flex justify-content-center py-4">
                <a href="index.html" class="logo d-flex align-items-center w-auto">
                  <img src="assets/img/logo." alt="">
                  <span class="d-none d-lg-block">KIET DATA CENTER</span>
                </a>
              </div><!-- End Logo -->

              <div class="card mb-3">

                <div class="card-body">
```

```html
    <div class="pt-4 pb-2">
      <h5 class="card-title text-center pb-0 fs-4">Login to Your Account</h5>
      <p class="text-center small">Enter your username & password to login</p>
    </div>

    <form class="row g-3 needs-validation" novalidate>

      <div class="col-12">
        <label for="yourUsername" class="form-label">Username</label>
        <div class="input-group has-validation">
          <span class="input-group-text" id="inputGroupPrepend">@</span>
          <input type="text" name="username" class="form-control" id="yourUsername" required>
          <div class="invalid-feedback">Please enter your username.</div>
        </div>
      </div>

      <div class="col-12">
        <label for="yourPassword" class="form-label">Password</label>
        <input type="password" name="password" class="form-control" id="yourPassword" required>
        <div class="invalid-feedback">Please enter your password!</div>
      </div>

      <div class="col-12">
        <div class="form-check">
          <input class="form-check-input" type="checkbox" name="remember" value="true" id="rememberMe">
          <label class="form-check-label" for="rememberMe">Remember me</label>
        </div>
      </div>
      <div class="col-12">
        <button class="btn btn-primary w-100" type="submit">Login</button>
      </div>
      <div class="col-12">
        <p class="small mb-0">Don't have account? <a href="pages-register.html">Create an account</a></p>
      </div>
    </form>

  </div>
</div>

<div class="credits">
  <!-- All the links in the footer should remain intact. -->
  <!-- You can delete the links only if you purchased the pro version. -->
  <!-- Licensing information: https://bootstrapmade.com/license/ -->
```

```
        <!-- Purchase the pro version with working PHP/AJAX contact form:
https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/ -->
        Designed by <a href="https://bootstrapmade.com/">BootstrapMade</a>
      </div>

    </div>
   </div>
  </div>

  </section>

 </div>
</main><!-- End #main -->

<a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi
bi-arrow-up-short"></i></a>

<!-- Vendor JS Files -->
<script src="assets/vendor/apexcharts/apexcharts.min.js"></script>
<script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="assets/vendor/chart.js/chart.min.js"></script>
<script src="assets/vendor/echarts/echarts.min.js"></script>
<script src="assets/vendor/quill/quill.min.js"></script>
<script src="assets/vendor/simple-datatables/simple-datatables.js"></script>
<script src="assets/vendor/tinymce/tinymce.min.js"></script>
<script src="assets/vendor/php-email-form/validate.js"></script>

<!-- Template Main JS File -->
<script src="assets/js/main.js"></script>

</body>

</html>
```

5.2 Staff Module

```
<!DOCTYPE html>
<html lang="en">

<head>
 <meta charset="utf-8">
 <meta content="width=device-width, initial-scale=1.0" name="viewport">

 <title>Dashboard - NiceAdmin Bootstrap Template</title>
 <meta content="" name="description">
 <meta content="" name="keywords">
```

```html
<!--Font awesome-->
<link         rel="stylesheet"        href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

<!-- Favicons -->
<link href="assets/img/favicon.png" rel="icon">
<link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

<!-- Google Fonts -->
<link href="https://fonts.gstatic.com" rel="preconnect">
<link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700
i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,70
0,700i" rel="stylesheet">

<!-- Vendor CSS Files -->
<link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
<link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
<link href="assets/vendor/quill/quill.snow.css" rel="stylesheet">
<link href="assets/vendor/quill/quill.bubble.css" rel="stylesheet">
<link href="assets/vendor/remixicon/remixicon.css" rel="stylesheet">
<link href="assets/vendor/simple-datatables/style.css" rel="stylesheet">

<!-- Template Main CSS File -->
<link href="assets/css/style.css" rel="stylesheet">
<link href="assets/css/custom_css.css" rel="stylesheet">

<!-- ========================================================
* Template Name: NiceAdmin
* Updated: Mar 09 2023 with Bootstrap v5.2.3
* Template URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/
* Author: BootstrapMade.com
* License: https://bootstrapmade.com/license/
======================================================== -->
</head>

<body>

<!-- ======= Header ======= -->
<?php include 'header.php'?>
<!-- End Header -->

<!--side bar-->
  <?php include 'sidenavbar.php'?>
<!--side bar-->

<main id="main" class="main">
```

```html
<div class="pagetitle">
 <h1>Dashboard</h1>
 <nav>
  <ol class="breadcrumb">
   <li class="breadcrumb-item"><a href="index.php">Home</a></li>
   <li class="breadcrumb-item active">Staff Dashboard</li>
  </ol>
 </nav>
</div><!-- End Page Title -->

<section class="section dashboard">
  <div class="row">
    <div class="col-md-12 col-lg-12 col-xl-12">
      <div class="card">
        <div class="card-body">
          <!-- Custom Styled Validation -->
          <form class="row py-3 needs-validation" action="" method="POST" novalidate>
            <div class="col">
              <div class="row">
                <div class="col-md-6 col-lg-6 col-xl-6">
                  <label             for="validationCustom04"             class="form-label">Department</label>
                  <select            class="form-select"           name="department" id="validationCustom04" required>
                    <option selected disabled value="">Choose...</option>
                    <option>MBA</option>
                    <option>B.Tech</option>
                    <option>MCA</option>
                  </select>
                  <div class="invalid-feedback">Please select a valid state.</div>
                </div>
                <div class="col-md-6 col-lg-6 col-xl-6">
                  <label        for="validationCustom01"        class="form-label">Staff name</label>
                  <input      type="text"     class="form-control"      name="staffname" id="validationCustom01" value="" required>
                  <div class="valid-feedback">Looks good!</div>
                </div>
              </div>

              <div class="row">
                <div class="col-md-6 col-lg-6 col-xl-6">
                  <label for="validationCustom01" class="form-label">ID</label>
                  <input      type="text"      name="staffid"       class="form-control" id="validationCustom01" value="" required>
                  <div class="valid-feedback">Looks good!</div>
                </div>
```

```html
                    <div class="col-md-6 col-lg-6 col-xl-6">
                        <label      for="validationCustom01"      class="form-label">Aadhar
Card</label>
                        <input     type="text"     class="form-control"     name="adharcard"
id="validationCustom01" value="" required>
                        <div class="valid-feedback">Looks good!</div>
                    </div>
                </div>

                <div class="row">
                    <div class="col-md-6 col-lg-6 col-xl-6">
                        <label for="validationCustom01" class="form-label">E-mail</label>
                        <input     type="text"     class="form-control"     name="email"
id="validationCustom01" value="" required>
                        <div class="valid-feedback">Looks good!</div>
                    </div>

                    <div class="col-md-6 col-lg-6 col-xl-6">
                        <label      for="validationCustom04"      class="form-label">Date      of
Joining</label>
                        <input      type="date"      class="form-control"      name="doj"
id="validationCustom01" value="" required>
                        <div class="invalid-feedback">Please select a valid state.</div>
                    </div>

                </div>

                <div class="row">
                    <div class="col-md-6 col-lg-6 col-xl-6">
                        <label              for="validationCustom04"              class="form-
label">Designation</label>
                        <select           class="form-select"           name="designation"
id="validationCustom04" required>
                            <option selected disabled value="">Choose...</option>
                            <option>Designation 1</option>
                        </select>
                        <div class="invalid-feedback">Please select a valid Designation.</div>
                    </div>

                    <div class="col-md-6 col-lg-6 col-xl-6">
                        <label    for="validationCustom04"    class="form-label">Nature    of
Appointment</label>
                        <select           class="form-select"           name="appointment"
id="validationCustom04" required>
                            <option selected disabled value="">Choose...</option>
                            <option>Designation 1</option>
                        </select>
                        <div class="invalid-feedback">Please select a valid Designation.</div>
```

```html
            </div>
          </div>

          <div class="row">
            <div class="col-md-6 col-lg-6 col-xl-6">
              <label     for="validationCustom04"     class="form-label">Highest
Degree</label>
                <select class="form-select" name="degree" id="validationCustom04"
required>
                  <option selected disabled value="">Choose...</option>
                  <option>Designation 1</option>
                </select>
                <div class="invalid-feedback">Please select a valid Designation.</div>
            </div>
          </div>

          <div class="row">
            <div class="col-12">
              <div class="form-check">
                <input    class="form-check-input"    type="checkbox"    value=""
id="invalidCheck" required>
                  <label class="form-check-label" for="invalidCheck">Agree to terms
and conditions</label>
                  <div    class="invalid-feedback">You    must    agree    before
submitting.</div>
              </div>
            </div>
          </div>

          <div class="row">
            <div class="col-12">
              <button       class="btn       btn-primary"       name="submit"
type="submit">Submit form</button>
            </div>
          </div>
        </div>
      </form>
      <!-- End Custom Styled Validation -->
    </div>
  </div>
  </div>
</section>

<section class="section dashboard">
  <div class="row">
    <div class="col-md-12 col-lg-12 col-xl-12">
      <div class="card">
```

```html
            <div class="card-body">
                <?php include 'stafftable.php'?>
            </div>
          </div>
        </div>
     </section>

  </main><!-- End #main -->

  <!-- ======= Footer ======= -->
   <?php include 'footer.php'?>
  <!-- End Footer -->

  <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi
bi-arrow-up-short"></i></a>

  <!-- Vendor JS Files -->
  <script src="assets/vendor/apexcharts/apexcharts.min.js"></script>
  <script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script src="assets/vendor/chart.js/chart.umd.js"></script>
  <script src="assets/vendor/echarts/echarts.min.js"></script>
  <script src="assets/vendor/quill/quill.min.js"></script>
  <script src="assets/vendor/simple-datatables/simple-datatables.js"></script>
  <script src="assets/vendor/tinymce/tinymce.min.js"></script>
  <script src="assets/vendor/php-email-form/validate.js"></script>

  <!-- Template Main JS File -->
  <script src="assets/js/main.js"></script>
  <script>
   if ( window.history.replaceState ) {
   window.history.replaceState( null, null, window.location.href );
   }
  </script>
  <?php include "connection.php"?>
</body>


<?php
    include 'connection.php';
    if(isset($_POST['submit'])){
        $department = $_POST['department'];
        $staffname = $_POST['staffname'];
        $staffid =$_POST['staffid'];
        $adharcard =$_POST['adharcard'];
        $email =$_POST['email'];
        $doj =$_POST['doj'];
        $designation =$_POST['designation'];
```

```php
$appointment =$_POST['appointment'];
$degree =$_POST['degree'];

$insertquery                                        ="insert                                        into
staff(branchid,staffname,staffid,adharcard,email,dateofjoining,designationid,appointmentid,degr
eeid)

values('$department','$staffname','$staffid','$adharcard','$email','$doj','$designation','$appointm
ent','$degree')";

$res = mysqli_query($con,$insertquery);

if($res){
    ?>
    <script>
    alert("data inserted");
    </script>

    <?php
}else{
?>
<script>
    alert ('not inserted');
</script>
<?php
    }

    }
?>
</html>
```
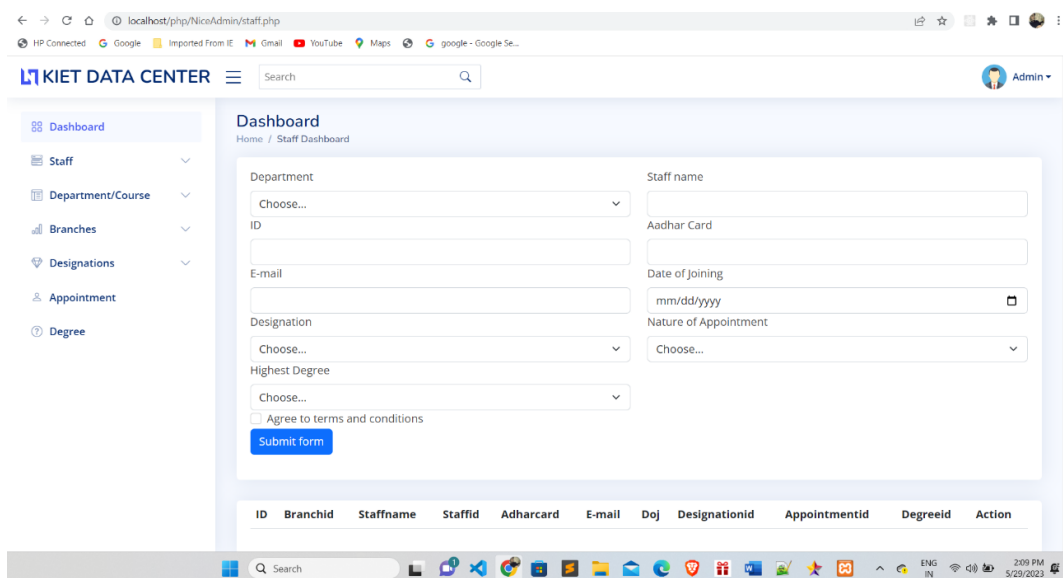

Fig:- 5.2(Staff Module)

5.3 Department Course
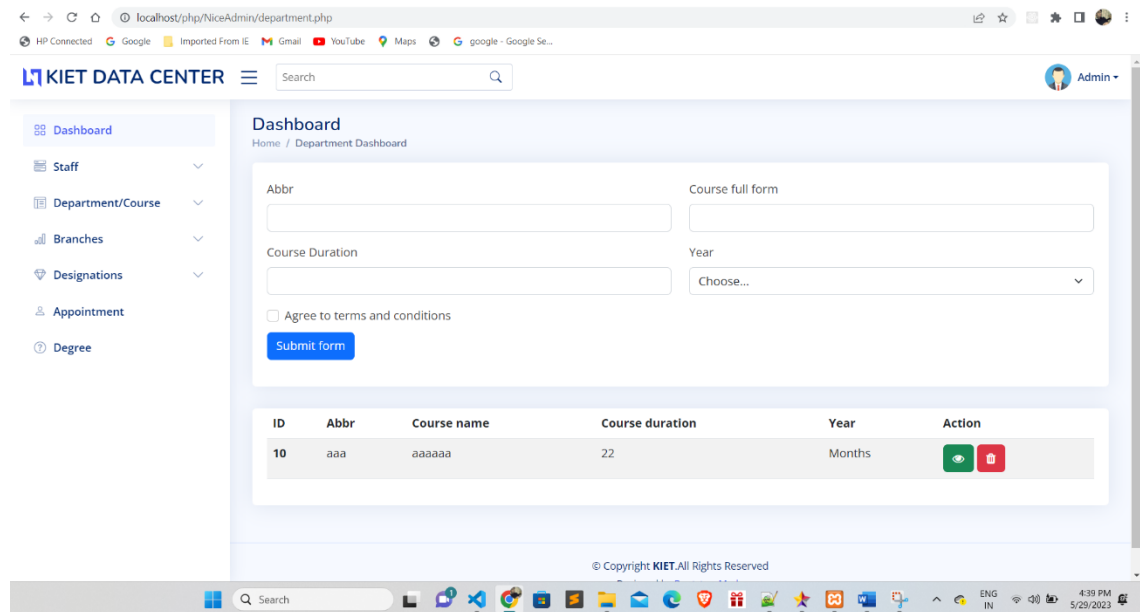
Fig :- 5.3(Department Course)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Dashboard - NiceAdmin Bootstrap Template</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!--Fontawesome-->
  <link          rel="stylesheet"          href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

  <!-- Favicons -->
  <link href="assets/img/favicon.png" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link href="https://fonts.gstatic.com" rel="preconnect">
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Nunito:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">
```

```html
<!-- Vendor CSS Files -->
<link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
<link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
<link href="assets/vendor/quill/quill.snow.css" rel="stylesheet">
<link href="assets/vendor/quill/quill.bubble.css" rel="stylesheet">
<link href="assets/vendor/remixicon/remixicon.css" rel="stylesheet">
<link href="assets/vendor/simple-datatables/style.css" rel="stylesheet">

<!-- Template Main CSS File -->
<link href="assets/css/style.css" rel="stylesheet">
<link href="assets/css/custom_css.css" rel="stylesheet">

<!-- =======================================================
* Template Name: NiceAdmin
* Updated: Mar 09 2023 with Bootstrap v5.2.3
* Template URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/
* Author: BootstrapMade.com
* License: https://bootstrapmade.com/license/
======================================================== -->
</head>

<body>

<!-- ======= Header ======= -->
<?php include 'header.php'?>
<!-- End Header -->

<!--side bar-->
  <?php include 'sidenavbar.php'?>
<!--side bar-->

<main id="main" class="main">

  <div class="pagetitle">
   <h1>Dashboard</h1>
   <nav>
    <ol class="breadcrumb">
     <li class="breadcrumb-item"><a href="index.php">Home</a></li>
     <li class="breadcrumb-item active">Department Dashboard</li>
    </ol>
   </nav>
  </div><!-- End Page Title -->

  <section class="section dashboard">
    <div class="row">
      <div class="col-md-12 col-lg-12 col-xl-12">
        <div class="card">
```

35

```html
<div class="card-body">
    <!-- Custom Styled Validation -->
    <form class="row py-3 needs-validation" action="" method="POST" novalidate>
        <div class="col">
            <div class="row py-2">
                <div class="col-md-6 col-lg-6 col-xl-6">
                    <label for="validationCustom01" class="form-label">Abbr</label>
                    <input type="text" class="form-control" name="abbr" id="validationCustom01" value="" required>
                    <div class="valid-feedback">Looks good!</div>
                </div>

                <div class="col-md-6 col-lg-6 col-xl-6">
                    <label for="validationCustom01" class="form-label">Course full form</label>
                    <input type="text" class="form-control" name="coursefullname" id="validationCustom01" value="" required>
                    <div class="valid-feedback">Looks good!</div>
                </div>
            </div>

            <div class="row py-2">
                <div class="col-md-6 col-lg-6 col-xl-6">
                    <label for="validationCustom01" class="form-label">Course Duration</label>
                    <input type="text" class="form-control" name="courseduration" id="validationCustom01" value="" required>
                    <div class="valid-feedback">Looks good!</div>
                </div>

                <div class="col-md-6 col-lg-6 col-xl-6">
                    <label for="validationCustom04" class="form-label">Year</label>
                    <select class="form-select" name="year" id="validationCustom04" required>
                        <option selected disabled value="">Choose...</option>
                        <option>Months</option>
                        <option>Years</option>
                        <option>Weeks</option>
                    </select>
                    <div class="invalid-feedback">Please select a valid Designation.</div>
                </div>
            </div>

            <div class="row py-2">
                <div class="col-12">
                    <div class="form-check">
                        <input class="form-check-input" type="checkbox" value="" id="invalidCheck" required>
```

```html
                              <label class="form-check-label" for="invalidCheck">Agree to terms
and conditions</label>
                              <div      class="invalid-feedback">You      must      agree      before
submitting.</div>
                           </div>
                        </div>
                     </div>

                     <div class="row">
                        <div class="col-12">
                           <button         class="btn         btn-primary"         name="submit"
type="submit">Submit form</button>
                        </div>
                     </div>
                  </div>
               </form>
               <!-- End Custom Styled Validation -->
            </div>
         </div>
       </div>
   </section>

   <section class="section dashboard">
      <div class="row">
         <div class="col-md-12 col-lg-12 col-xl-12">
            <div class="card">
               <div class="card-body">
                  <?php include 'departmenttable.php'?>
               </div>
            </div>
         </div>
      </div>
   </section>

 </main><!-- End #main -->

 <!-- ======= Footer ======= -->
  <?php include 'footer.php'?>
 <!-- End Footer -->

 <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi
bi-arrow-up-short"></i></a>

 <!-- Vendor JS Files -->
 <script src="assets/vendor/apexcharts/apexcharts.min.js"></script>
 <script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
 <script src="assets/vendor/chart.js/chart.umd.js"></script>
```

```
<script src="assets/vendor/echarts/echarts.min.js"></script>
<script src="assets/vendor/quill/quill.min.js"></script>
<script src="assets/vendor/simple-datatables/simple-datatables.js"></script>
<script src="assets/vendor/tinymce/tinymce.min.js"></script>
<script src="assets/vendor/php-email-form/validate.js"></script>

<!-- Template Main JS File -->
<script src="assets/js/main.js"></script>
<script>
  if ( window.history.replaceState ) {
  window.history.replaceState( null, null, window.location.href );
  }
</script>
<?php include "connection.php"?>
</body>


<?php
    include 'connection.php';
    if(isset($_POST['submit'])){
       $abbr = $_POST['abbr'];
       $coursefullname = $_POST['coursefullname'];
       $courseduration =$_POST['courseduration'];
       $year =$_POST['year'];



       $insertquery                          ="insert                          into
departmentcourse(abbr,coursename,courseduration,durationunit)
       values('$abbr','$coursefullname','$courseduration','$year')";
       $res = mysqli_query($con,$insertquery);
       if($res){
          ?>
          <script>
          alert("data inserted");
          </script>
          <?php
       }else{
   ?>
   <script>
     alert ('not inserted');
   </script>
   <?php
       }
     }
?>
</html>
```
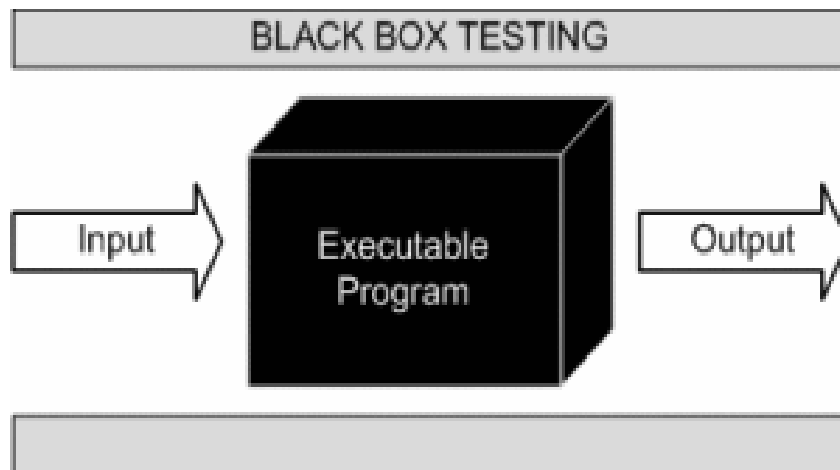
CHAPTER 6

Testing

6.1 Test Case – 1

Black-Box Testing

Black Box Testing, also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This can be following way:
- ➢ Input interfacing
- ➢ Processing
- ➢ Output interfacing

Fig: - 6.1(Black-Box Testing)

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:
- ➢ Incorrect or missing functions
- ➢ Interface errors

> ➢ Errors in data structures or external database access
> ➢ Behaviour or performance errors
> ➢ Initialization and termination errors.

6.2 Test Case – 2

White-Box Testing

White Box Testing, also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.

The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential.

White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like a white/ transparent box; inside which one clearly sees.

Conclusion

In conclusion, the management of college and department databases plays a crucial role in the efficient operation of educational institutions. This project aims to develop and implement a comprehensive database management system that covers various aspects, including student certificates, academic details, event information, account department records, human resources department records, and student placements in organizations.

By implementing this system, the college will benefit from improved data management processes, streamlined workflows, and enhanced decision-making capabilities. The system will enable easy access to information, facilitate efficient updating and editing of records, and ensure proper resource management of college data. This, in turn, will lead to improved services for students, faculty, and staff.

Furthermore, the system will minimize paperwork, reduce dependency on manual processes, and promote a more organized and efficient working environment. By automating tasks and eliminating redundant procedures, valuable time will be saved, allowing staff and students to focus on more productive activities.

Additionally, the proposed database management system will improve the overall visibility and tracking of student certificates, academic details, and event participation. It will also provide better management of the account department and human resources department, enabling effective financial record-keeping and streamline HR processes.

The system's ability to track student placements in various organizations will enhance the college's ability to monitor and support students' career progress. It will also strengthen relationships with partnering organizations, leading to increased placement opportunities for students.

Overall, this project will provide a comprehensive and efficient database management system for the college, leading to improved data management, streamlined processes, and better services for all stakeholders. It will enhance the institution's reputation as a leading educational establishment and contribute to its overall success and growth.