

NICHE CAMPUS MANAGEMENT SOFTWARE (NICHE ERP)

NICHE ERP Modules and Functionalities - Configuration and Operations



NICHE
KANYAKUMARI
DEEMED-TO-BE-UNIVERSITY

Table of Contents

ABSTRACT	5
II. PURPOSE OF THE SOFTWARE	6
III. INTRODUCTION	7
IV. NICHE CAMPUS MANAGEMENT SOFTWARE	8
4.1. Admission Module	8
4.2. Academics Module	8
4.3. Finance Module	9
4.4. Examination Module	10
4.5. Human Resource Module.....	11
4.6. Hostel Module	12
4.7. Administration Module	13
V. FEASIBILITY STUDY FOR NICHE CAMPUS MANAGEMENT SOFTWARE.....	14
5.1. Technical Feasibility.....	14
5.1.1. Infrastructure Requirements.....	14
5.1.2. Security Features	14
5.1.3. Technical Support and Maintenance.....	14
5.2. Operational Feasibility	14
5.2.1. Ease of Use.....	14
5.2.2. Scalability	14
5.2.3. Process Automation.....	14
5.3. Financial Feasibility	14
5.3.1. Initial Costs	14
5.3.2. Operational Costs	14
5.4. Legal Feasibility	14
5.4.1. Compliance with Regulations	14
5.4.2. Licensing and Copyrights	15
5.5. Time Feasibility	15
5.5.1. Implementation Timeline	15
5.5.2. Timeline for Benefits Realization.....	15
5.6. Risk Assessment	15
5.6.1. Potential Risks	15
5.6.2. Contingency Plans	15
VI. SYSTEM REQUIREMENTS	16
6.1. Software Requirements	16
6.1.1. Operating System.....	16
6.1.2. Database Management System (DBMS)	16
6.1.3. Web Server	16

6.1.4.	Web Browser Compatibility (For Client/Student Access)	16
6.1.5.	Middleware/Framework	16
6.1.6.	Additional Software Tools	16
6.2.	Hardware Requirements.....	17
6.2.1.	Server Requirements.....	17
6.2.2.	Client Workstation Requirements	17
6.2.3.	Mobile Device Requirements	18
6.2.4.	Backup and Disaster Recovery Hardware	18
6.3.	Additional Considerations.....	18
VII. INPUTS AND OUTPUTS FOR NICHE CAMPUS MANAGEMENT SOFTWARE.		19
7.1.	Admission Module	19
7.1.1.	Inputs:.....	19
7.1.2.	Outputs:	19
7.2.	Academics Module	19
7.2.1.	Inputs:	19
7.2.2.	Outputs:	20
7.3.	Finance Module	20
7.3.1.	Inputs:	20
7.3.2.	Outputs:	21
7.4.	Examination Module	21
7.4.1.	Inputs:	21
7.4.2.	Outputs:	21
7.5.	Human Resource Module.....	22
7.5.1.	Inputs:	22
7.5.2.	Outputs:	22
7.6.	Hostel Module	22
7.6.1.	Inputs:	22
7.6.2.	Outputs:	23
7.7.	Administration Module	23
7.7.1.	Inputs:	23
7.7.2.	Outputs:	23
VIII. PROCESS MODEL USED FOR NICHE CAMPUS MANAGEMENT SOFTWARE		24
.....		24
8.1.	Justification for Using the Spiral Model.....	24
8.2.	Key Features of the Spiral Model in the Context of NICHE Software:.....	24
8.2.1.	Iterative Development.....	24
8.2.2.	Risk Management.....	24
8.2.3.	User Feedback Integration	24
8.2.4.	Adaptability to Changing Requirements	25
8.2.5.	Focus on Prototyping.....	25
8.2.6.	Incremental Delivery	25
8.3.	Phases of the Spiral Model for NICHE Software Development	25
8.3.1.	Planning Phase.....	25
8.3.2.	Risk Analysis Phase.....	25
8.3.3.	Engineering Phase (Development & Prototyping)	25
8.3.4.	Testing & Evaluation Phase	26
8.3.5.	Feedback Phase	26
8.3.6.	Deployment & Maintenance Phase	26

8.4. Advantages of the Spiral Model for NICHE Campus Management Software	26
IX. FUNCTIONAL REQUIREMENTS	27
9.1. Admission Module	27
9.2. Academics Module	27
9.3. Finance Module	27
9.4. Examination Module	28
9.5. Human Resources Module	28
9.6. Hostel Management Module	28
9.7. Administration Module.....	28
X. SPECIFICATIONS OVERVIEW	29
10.1. PHP	29
10.1.1. Overview of PHP	29
10.1.2. Key Features of PHP	29
10.1.3. PHP in Campus Management Software	29
10.2. DBMS (Database Management System).....	30
10.2.1. Overview of DBMS	30
10.2.2. Types of DBMS.....	30
10.2.3. Key Features of DBMS	31
10.2.4. Role of DBMS in NICHE Campus Management Software	31
10.2.5. Database Choice for NICHE Campus Management Software.....	32
10.3. Benefits of Using PHP and DBMS	32
XI. DATABASE DESCRIPTION FOR NICHE CAMPUS MANAGEMENT SOFTWARE	33
11.1. Database Overview	33
11.2. Overview of Major Tables	33
11.2.1. Admissions and Student Management	33
11.2.2. Academics and Course Management	34
11.2.3. Attendance Management	34
11.2.4. Examinations Management	34
11.2.5. Human Resources (HR) Management	35
11.2.6. Hostel Management	35
11.2.7. Key Fields in Some Tables.....	35
11.3. Database Relationships	36
11.4. Security and Integrity	36
XII. SYSTEM TESTING	37
12.1. Unit Testing	37
12.2. Integration Testing.....	37
12.3. System Testing.....	37
12.4. User Acceptance Testing (UAT).....	37
12.5. Performance Testing	37
XIII. USER TRAINING	38

13.1. Admin and Faculty Training	38
13.2. Student Orientation	38
XIV. SECURITY IN SOFTWARE FOR NICHE CAMPUS MANAGEMENT SOFTWARE	39
14.1. User Authentication and Authorization.....	39
14.2. Data Encryption.....	39
14.3. Secure Database Management	40
14.4. Session Management.....	40
14.5. Audit Logs and Monitoring	41
14.6. Data Privacy and Compliance.....	41
14.7. Regular Security Updates and Patching	41
14.8. Backup and Disaster Recovery.....	42
XV. MAINTENANCE IN SOFTWARE FOR NICHE CAMPUS MANAGEMENT SOFTWARE	43
15.1. Types of Maintenance	43
15.2. Maintenance Activities	43
15.1.1. Regular Software Updates	44
15.1.2. Bug Fixing and Error Resolution	44
15.1.3 Performance Optimization	44
15.1.4. Security Maintenance	45
15.1.5. User Support and Feedback Management	45
15.1.6. Documentation Updates	45
15.7. Maintenance Schedule	46
15.8. Benefits of Ongoing Maintenance	46
CONCLUSION	47
XVII. LIMITATIONS	48
17.1. Dependency on Internet Connectivity.....	48
17.2. Training and Adaptability	48
17.3. Limited Mobile Functionality	48
XVIII. FUTURE IMPROVEMENTS	49
18.1. Mobile Application Development	49
18.2. Artificial Intelligence (AI) Integration	49
18.3. Advanced Data Analytics and Reporting	49
18.4. Seamless Integration with Other Educational Platforms	49
APPENDICES	50
APPENDIX I: SCREENSHOTS OF DATABASE TABLES.....	50
APPENDIX II: SCREENSHOTS OF DATABASE FIELDS FROM EACH TABLE	52

ABSTRACT

The NICHE Campus Management Software (ERP) is a comprehensive solution designed to streamline and automate the management of information related to students, faculty, courses, and administrative functions within the institution. This software provides an integrated platform to manage the entire academic lifecycle, from the initial registration of new students to the tracking of their attendance, academic performance, and overall progress. By automating administrative processes, NICHE ERP enhances efficiency, accuracy, and accessibility, ensuring that essential information is available to stakeholders across the campus. The ERP is an intranet-based campus-wide portal that consolidates data from all departments, enabling centralised access and retrieval of information. It facilitates the generation of various reports that support data-driven decisions and help measure both individual and collective student performance. Additionally, the software provides robust tools for handling financial operations, hostel and transport services, human resources, and examination management, thus covering the broad range of administrative and academic needs of the institution. The development process for the NICHE Campus Management Software begins with a thorough software analysis, where a formal model of the requirements and challenges is created. By providing a centralised, automated, and user-friendly software, NICHE ERP aims to support delivering high-quality education while optimising administrative tasks.

II. PURPOSE OF THE SOFTWARE

The NICHE Campus Management Software (ERP) is designed to efficiently manage and streamline information related to students, faculty, courses, and administrative functions across the institution. Unlike traditional systems, NICHE ERP provides an integrated and automated solution that is customizable for diverse college management requirements within the institution.

This system enables seamless retrieval and sharing of information through a secure, intranet-based campus-wide portal, allowing administrators, faculty, and students to access data in real time. NICHE ERP consolidates information from various departments, including admissions, academics, finance, and human resources, into a single platform. This centralization of data facilitates the generation of comprehensive reports that are critical for monitoring individual and overall student performance, assessing academic outcomes, and supporting data-driven decisions within the institution. NICHE ERP not only enhances operational efficiency but also ensures accuracy, transparency, and accessibility in managing campus activities.

III. INTRODUCTION

In today's rapidly evolving educational landscape, educational institutions face increasing demands for efficient, transparent, and scalable management solutions. From student admissions and academic scheduling to financial transactions and administrative tasks, the volume and complexity of data handled daily are substantial. Manual processes not only slow down operations but also increase the likelihood of errors and reduce overall efficiency. To address these challenges, the NICHE Campus Management Software (ERP) has been designed to provide a comprehensive, integrated solution.

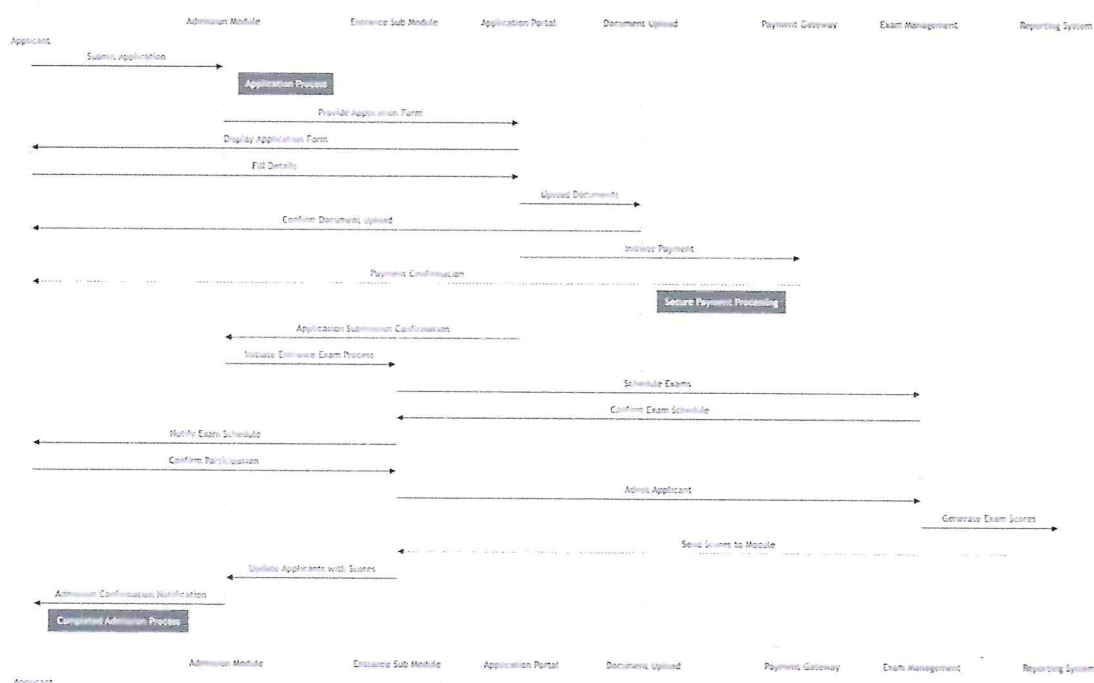
The NICHE ERP is a robust platform that automates and streamlines core institution functions across multiple departments, supporting administrative, academic, and operational activities. Through modular components covering admissions, academics, finance, examinations, human resources, hostel and fleet management, and overall administration, NICHE ERP delivers an intuitive and centralized approach to managing campus operations. Each module is crafted to enhance data accessibility, improve accuracy, and reduce the time spent on routine tasks, thereby allowing faculty and administrators to focus more on strategic initiatives and student support.

The following report provides an in-depth exploration of the proposed NICHE ERP, outlining each module and its specific functionalities. Implementing NICHE ERP enhances improved resource allocation, enhanced student engagement, and streamlined workflows. This report will serve as a comprehensive guide to understanding how NICHE ERP can contribute to transforming institution management, ultimately promoting a more efficient and effective educational environment.

IV. NICHE CAMPUS MANAGEMENT SOFTWARE

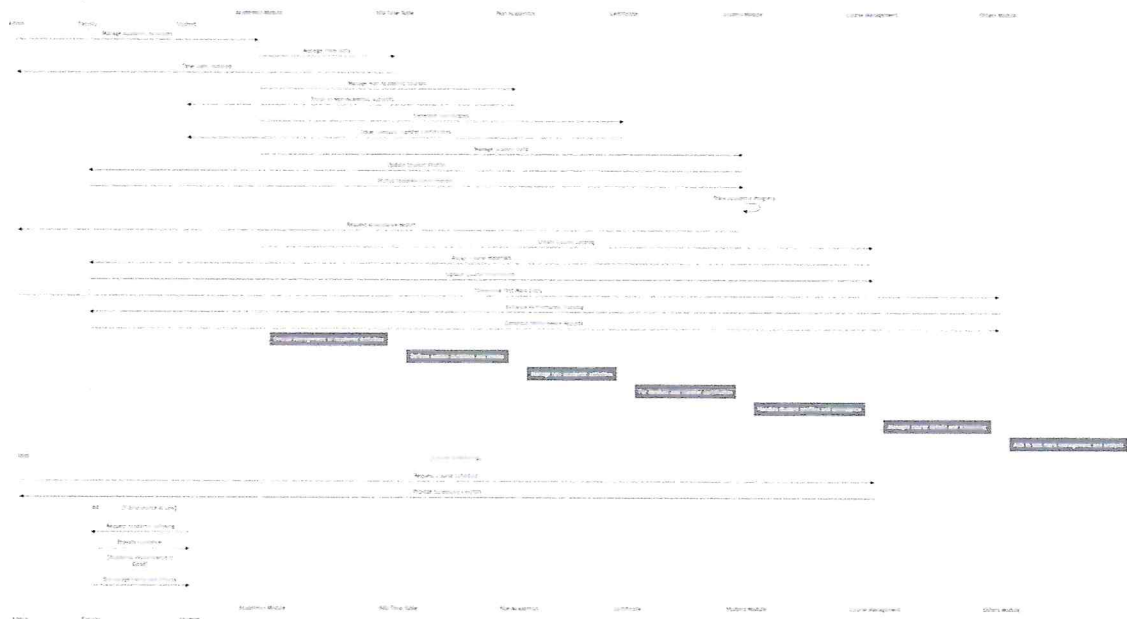
The NICHE Campus Management Software (ERP) is an integrated platform designed to streamline and automate diverse administrative and academic functions. It is modular, allowing for efficient management across admissions, academics, finance, examinations, human resources, hostel services, and overall administration. Each module enhances efficiency, accuracy, and real-time data access. The core features of the proposed system are as follows:

4.1. Admission Module: This module provides a user-friendly online platform for prospective students to apply, upload documents, and pay fees. The **Entrance Module** automates exam scheduling, scoring, and reporting, while the **Admission Module** manages applicant data, payment schedules, and daily admission reports. This reduces administrative load and improves the applicant experience.



4.2. Academics Module: The central component of NICHE ERP, this module manages academic activities like course scheduling, enrolment, and examination management. Faculty can update attendance, track student performance, and manage grades. Key submodules include **NICHE Time Table** (class scheduling), **Non-Academics** (extracurriculars), **Certificate** (conduct and transfer certificates), **Student** (attendance, profiles, and leave), **Course** (catalogue and scheduling), and **Others** (test marks and

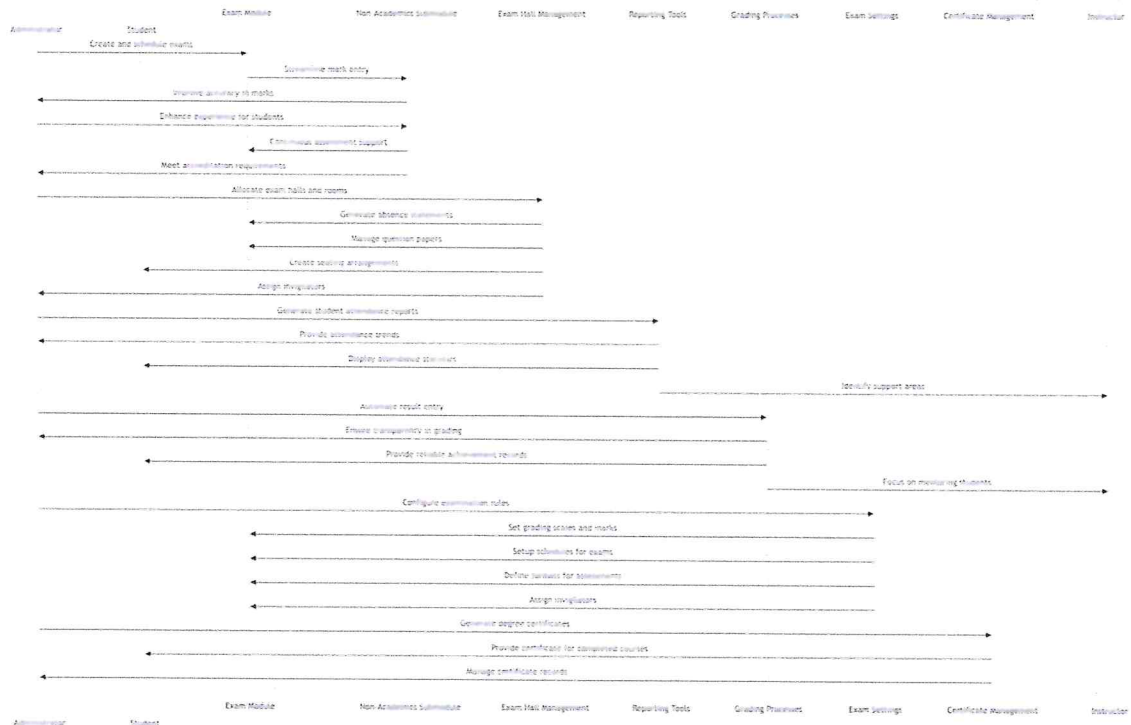
performance tracking). This module improves both academic organization and user experience.



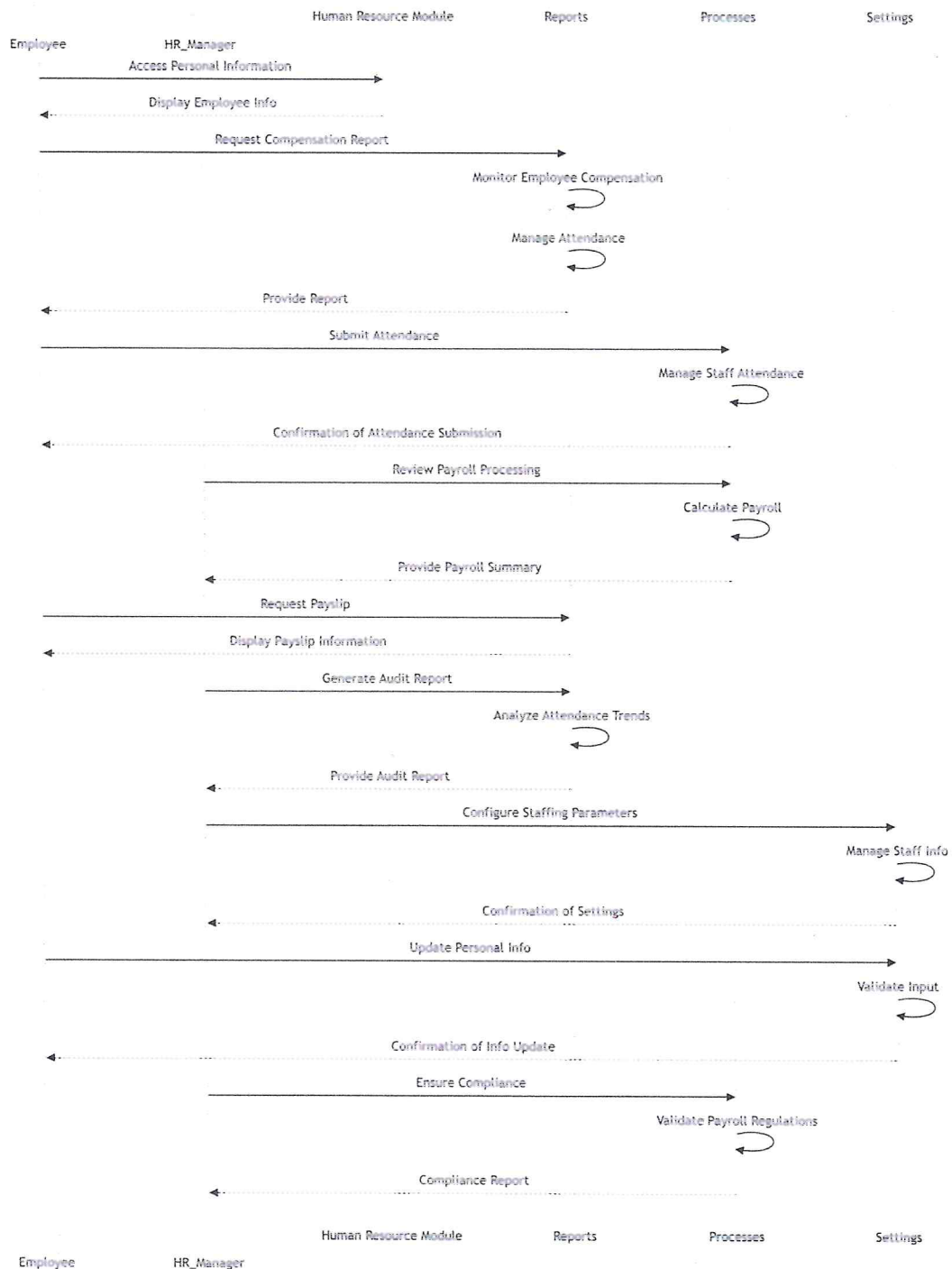
4.3. Finance Module: This module handles all financial operations, including tuition, exam, and transport fees. Submodules for **Reports** (statements on fees and payments), **Processes** (fees for certificates and compliance), and **Settings** (customizing workflows and integrations) streamline transactions, ensure regulatory compliance, and offer financial transparency. It reduces administrative burden and simplifies fee tracking.



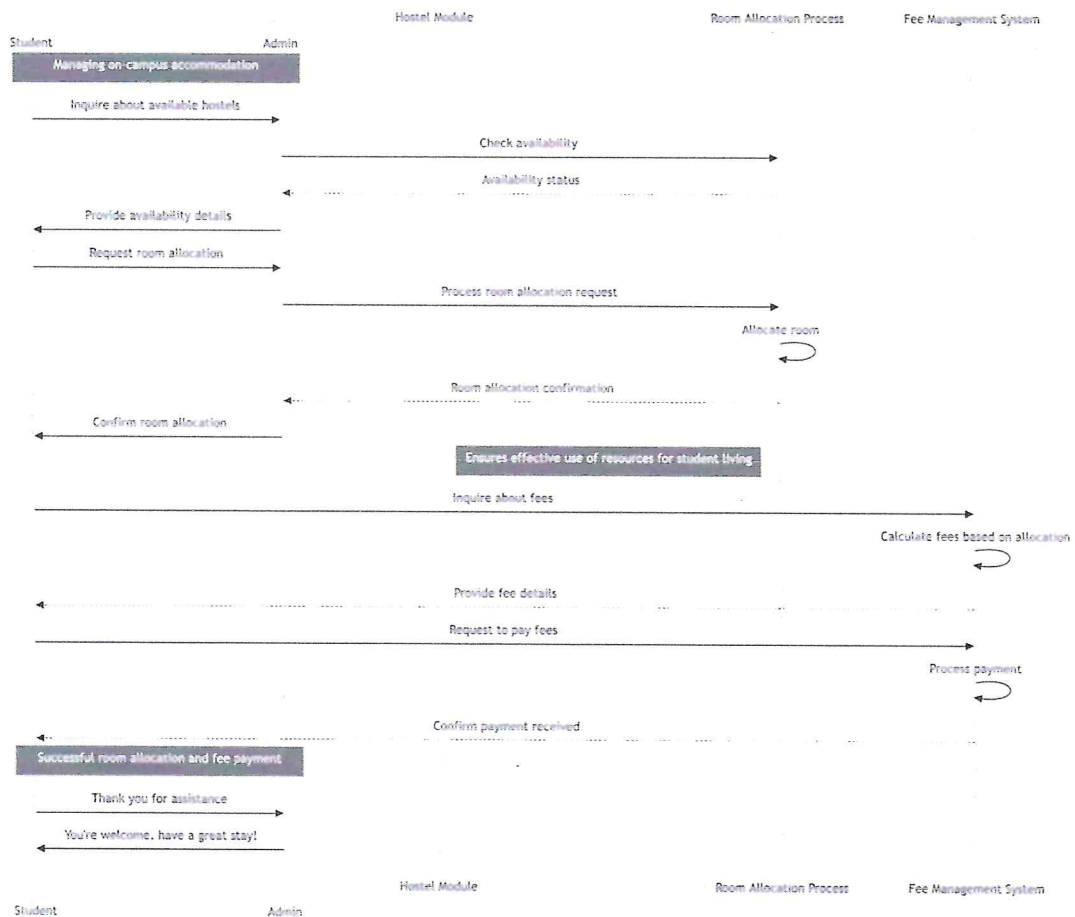
4.4. Examination Module: Supporting the end-to-end exam process, this module includes **Non-Academics** (non-academic course exams), **Exam Hall** (room and seating management), **Reports** (exam performance and attendance), **Processes** (automated result entry), **Settings** (exam rules and formats), and **Certificates** (degree issuance). It ensures secure exam administration and reliable grading, with tools for quality assurance.



4.5. Human Resource Module: This module centralizes employee data, payroll, and attendance. Submodules include **Reports** (payroll and attendance), **Processes** (payroll accuracy and compliance), and **Settings** (HR configurations). It supports transparent HR operations, reduces manual errors, and ensures payroll compliance.

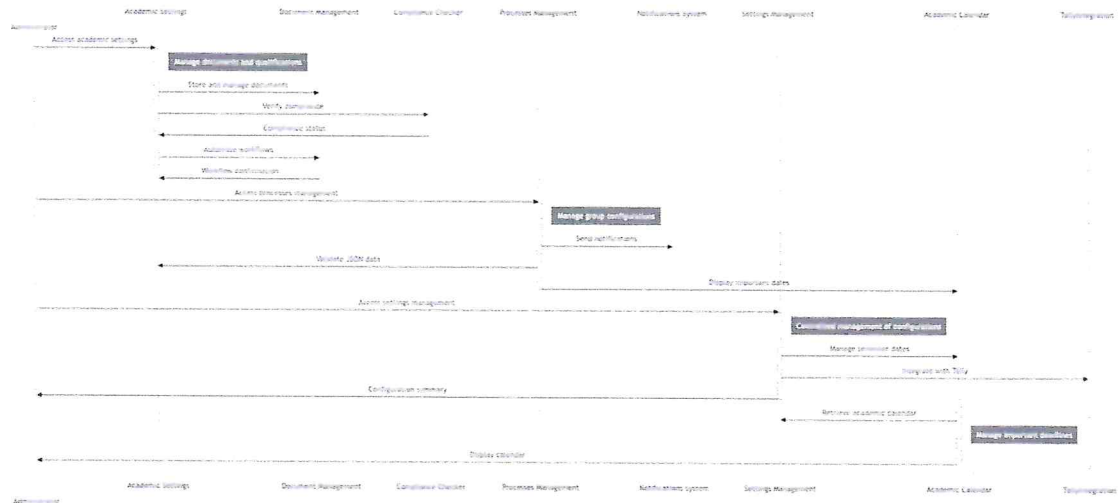


4.6. Hostel Module: Managing on-campus accommodation, this module includes **Hostel** (room allocation, fees). It ensures effective use of resources for student living.



NICHE ERP Modules and Functionalities - Configuration and Operations

4.7. Administration Module: This module enables centralized control and customization for institution operations. **Academic Settings** supports document management and compliance, **Processes** manages group settings and notifications, and **Settings** covers the academic calendar, Tally integration, and user preferences. It ensures smooth institution-wide operations and policy standardization.



Each module in NICHE ERP is designed to minimise manual processes, improve transparency, and enable data-driven decisions, enhancing the overall efficiency and experience for students, faculty, and administrators.

V. FEASIBILITY STUDY FOR NICHE CAMPUS MANAGEMENT SOFTWARE

5.1. Technical Feasibility

- 5.1.1. Infrastructure Requirements:** Evaluate the campus's current IT infrastructure (servers, hardware, and internet connectivity) to ensure compatibility with NICHE Campus Management Software.
- 5.1.2. Security Features:** Assess the software's ability to handle data security and comply with relevant regulations.
- 5.1.3. Technical Support and Maintenance:** Review the availability of support teams to maintain and troubleshoot the software post-implementation.

5.2. Operational Feasibility

- 5.2.1. Ease of Use:** Evaluate the user-friendliness of the software for different stakeholders (students, staff, administrators, etc.). This includes understanding the learning curve and training requirements.
- 5.2.2. Scalability:** Examine whether the software can scale according to the growing size of the institution or accommodate more users if required.
- 5.2.3. Process Automation:** Assess the degree to which the software automates routine tasks such as admissions, attendance, grading, and payroll to improve operational efficiency.

5.3. Financial Feasibility

- 5.3.1. Initial Costs:** Calculate the upfront costs for acquiring, installing, and configuring the software, including hardware.
- 5.3.2. Operational Costs:** Estimate ongoing costs for maintenance, support, upgrades, and any subscription fees.

5.4. Legal Feasibility

- 5.4.1. Compliance with Regulations:** Evaluate whether the software complies with local and national educational regulations, including those concerning student data, payments, and academic certifications.

5.4.2. Licensing and Copyrights: Verify that the software adheres to legal requirements for intellectual property and licenses, especially for proprietary features.

5.5. Time Feasibility

5.5.1. Implementation Timeline: Estimate how long it will take to deploy the software, including setup, configuration, training, and full-scale rollout.

5.5.2. Timeline for Benefits Realization: Define the time frame for realizing the benefits, such as time savings, improved accuracy, and operational efficiency.

5.6. Risk Assessment

5.6.1. Potential Risks: Identify risks such as system failures, data breaches, user resistance, and unforeseen costs. Propose mitigation strategies for these risks.

5.6.2. Contingency Plans: Develop contingency plans for technical failures, including backup systems, data recovery processes, and alternative manual operations.

VI. SYSTEM REQUIREMENTS

6.1. Software Requirements

6.1.1. Operating System

- **Server Side:**
 - Windows Server 2019
- **Client Side:**
 - Windows 10 or higher (for administrative staff and faculty use)
 - macOS Mojave or higher (for administrative staff and faculty use)
 - Linux (Ubuntu 18.04 or higher, CentOS 7 or higher) for students, staff, or faculty on non-Windows machines.

6.1.2. Database Management System (DBMS)

- **Supported DBMS:**
 - MySQL 8.0

6.1.3. Web Server

- **Apache HTTP Server** (latest stable version)

6.1.4. Web Browser Compatibility (For Client/Student Access)

- **Google Chrome** (latest version)
- **Mozilla Firefox** (latest version)
- **Safari** (latest version for macOS)
- **Microsoft Edge** (latest version)

6.1.5. Middleware/Framework

- **PHP**

6.1.6. Additional Software Tools

- **Email Server** for communication purposes

6.2. Hardware Requirements

6.2.1. Server Requirements

- **Processor:**
 - Intel Xeon E5-2620 v4
- **Memory (RAM):**
 - 16 GB
- **Storage:**
 - 1 TB
- **Network Interface:**
 - 1 Gbps Ethernet
- **Backup Storage:**
 - Cloud daily backups
- **Redundancy and Failover Support:**
 - Load balancing and failover setup to ensure system availability
 - Backup power supply (UPS) for uninterrupted service

6.2.2. Client Workstation Requirements

- **Processor:**
 - Minimum: Intel Core i3 or equivalent
 - Recommended: Intel Core i5 or equivalent (for better multi-tasking performance)
- **Memory (RAM):**
 - Minimum: 4 GB
 - Recommended: 8 GB or higher
- **Storage:**
 - Minimum: 250 GB HDD or 120 GB SSD
 - Recommended: 500 GB SSD or higher for smooth performance
- **Display:**
 - Minimum: 1366 x 768 resolution
 - Recommended: Full HD (1920 x 1080) for better readability
- **Input Devices:**
 - Keyboard and mouse (or touchscreen for touch-enabled devices)

- **Network Connectivity:**
 - Stable internet connection (minimum 2 Mbps for optimal performance, higher for multiple users)

6.2.3. Mobile Device Requirements

- **iOS Devices:**
 - Minimum: iOS 12 or higher
 - Devices like iPhone or iPad with internet connectivity for accessing the system
- **Android Devices:**
 - Minimum: Android 8.0 (Oreo) or higher
 - Devices like smartphones or tablets with internet connectivity

6.2.4. Backup and Disaster Recovery Hardware

- **Backup Server:**
 - Additional storage (e.g., network-attached storage or cloud-based backups) for disaster recovery and long-term data retention.

6.3. Additional Considerations

- **Network Infrastructure:** High-speed internal network setup with Wi-Fi support in campus locations for seamless access to the system.
- **Security Devices:** Firewalls, intrusion detection systems, and other security hardware to protect the system from unauthorized access and cyber threats.

VII. INPUTS AND OUTPUTS FOR NICHE CAMPUS MANAGEMENT SOFTWARE

To ensure that the NICHE Campus Management Software operates effectively, it's important to define the inputs and outputs for each module. Below is a detailed list of inputs and outputs for the various modules of the system.

7.1. Admission Module

7.1.1. Inputs:

- **Prospective Student Data:**
 - Personal Information (Name, Address, Date of Birth, etc.)
 - Academic Records (Previous qualifications, grades, etc.)
 - Documents (Photographs, Mark Sheets, etc.)
 - Entrance Exam Scores (If applicable)
- **Application Fee Payments:**
 - Payment details (Amount, Payment Method, Transaction ID)
- **Admission Preferences:**
 - Preferred Courses
 - Hostel Preferences (if applicable)

7.1.2. Outputs:

- **Application Status:**
 - Confirmation of Application Submission
 - Admission Status (Accepted, Pending, Rejected)
- **Admission Confirmation:**
 - Admission Letter or Receipt
 - Payment Confirmation Receipt
- **Report:**
 - Daily Admission Status Report (Applicants, Accepted, Pending, Rejected)

7.2. Academics Module

7.2.1. Inputs:

- **Course Data:**

- Course Name, Description, Credit Hours, Pre-requisites
 - Faculty Assignment (Faculty Member Names)
 - Class Timings and Venue Information
- **Student Enrollment Data:**
 - Student's Course Registration
 - Student's Attendance Data
- **Examination Data:**
 - Exam Timetables, Dates, and Venue Information
 - Question Papers and Answer Sheets

7.2.2. Outputs:

- **Timetable:**
 - Individual Timetable for Students and Faculty
 - Academic Calendar (Semester-wise)
- **Student Progress Reports:**
 - Academic Performance (Grades, Attendance, etc.)
 - Mid-term and Final Exam Results
- **Certificate Generation:**
 - Conduct Certificates
 - Transfer Certificates
- **Reports:**
 - Course Enrollment Report
 - Class Attendance Report
 - Performance Reports (Exam Marks, Grades)

7.3. Finance Module

7.3.1. Inputs:

- **Fee Details:**
 - Tuition Fees, Hostel Fees, Exam Fees, Other Miscellaneous Fees
 - Payment Method (Online, Cash, Bank Transfer)
- **Student Payment Data:**
 - Payment Date, Payment Amount, Payment Method
 - Fee Waiver or Discount Information
- **Financial Processes Data:**

- Fee Schedules, Deadlines, Installment Plans

7.3.2. Outputs:

- **Payment Receipts:**
 - Detailed Receipts for Student Payments (Tuition, Hostel, Exam, etc.)
- **Fee Payment Reports:**
 - Fee Collection Status (Daily, Monthly, Yearly)
 - Outstanding Fee Balance Reports
- **Financial Statements:**
 - Tuition Fee Summary
 - Fees Paid and Balance Reports for Each Student
- **Receipts for Miscellaneous Fees:**
 - Provisional Certificate Fee
 - Duplicate Document Fee, Special Fees

7.4.Examination Module

7.4.1. Inputs:

- **Exam Data:**
 - Exam Schedules (Dates, Times, Venues)
 - Question Papers, Answer Sheets
 - List of Invigilators and Exam Supervisors
- **Student Data:**
 - Student Enrollment Data (for the exams)
 - Student Attendance Data for Exams
- **Grading Data:**
 - Marks for Each Student
 - Exam Performance Data

7.4.2. Outputs:

- **Exam Hall Allocation:**
 - Seating Arrangements for Students
 - Hall Occupancy Status
- **Grading Results:**
 - Individual Student Marks for Exams
 - Final Exam Results (Grades, Marks)

- **Reports:**
 - Exam Attendance Report
 - Exam Results Report
 - Performance Analysis Report
- **Certificates:**
 - Degree Certificates
 - Transcript Generation

7.5.Human Resource Module

7.5.1. Inputs:

- **Employee Data:**
 - Personal Information (Name, Address, Position, Department)
 - Attendance Data (Daily Attendance, Leaves, Holidays)
 - Payroll Data (Salary, Bonuses, Deductions)
- **Employee Requests:**
 - Leave Requests (Sick Leave, Casual Leave, etc.)
 - Recruitment Requests

7.5.2. Outputs:

- **Payslips:**
 - Monthly Salary Details, Deductions, Bonuses
- **Employee Reports:**
 - Attendance Report (Absenteeism, Latecomers, etc.)
 - Payroll Summary Report
- **Employee Leave Details:**
 - Leave Approval/Rejection Status
 - Leave Balance Report

7.6.Hostel Module

7.6.1. Inputs:

- **Hostel Data:**
 - Student Accommodation
 - Room Assignment Data
 - Hostel Fee Details

7.6.2. Outputs:

- **Hostel Allocation Report:**
 - Room Assignment Status
 - Student List in Each Hostel Room
- **Billing Reports:**
 - Hostel Fee Payments

7.7. Administration Module

7.7.1. Inputs:

- **Administrative Settings:**
 - Institutional Policies, Academic Calendar, and Holiday Information
 - Departmental and Faculty Assignments
- **Campus Data:**
 - Campus Facilities and Resource Allocation
 - Departmental Budgets and Financial Information

7.7.2. Outputs:

- **Institutional Reports:**
 - Academic and Administrative Reports
 - Accreditation and Compliance Reports
- **Campus Operations Reports:**
 - Resource Usage (Classrooms, Labs, etc.)
 - Budget and Financial Health Reports
- **Custom Reports:**
 - Tailored Reports for Specific Departments or Roles
 - Real-Time Analytics for Strategic Decision Making

VIII. PROCESS MODEL USED FOR NICHE CAMPUS MANAGEMENT SOFTWARE

The NICHE Campus Management Software is designed to streamline and automate the various administrative, academic, and operational processes. To ensure the effective functioning of this software, a well-defined process model is necessary. The chosen process model for NICHE is the **Spiral Model**, which is highly suitable for projects that require frequent iterations, constant feedback, and continuous improvement.

8.1. Justification for Using the Spiral Model

The Spiral Model is a risk-driven, iterative software development model that combines the iterative nature of prototyping with the systematic aspects of the waterfall model. It is particularly effective for large, complex systems like campus management software, which require frequent updates, testing, and feedback.

8.2. Key Features of the Spiral Model in the Context of NICHE Software:

- 8.2.1. Iterative Development:** The Spiral Model breaks down the project into smaller, manageable iterations, allowing for the continual refinement of features. Each iteration or "spiral" involves planning, risk analysis, development, testing, and evaluation. For NICHE software, this means that each module (Admission, Academics, Finance, etc.) can be developed and tested incrementally, with constant improvements based on user feedback.
- 8.2.2. Risk Management:** Each cycle of the Spiral Model includes risk analysis, which helps to identify potential issues early in the process. For NICHE software, this is crucial, as the system involves various stakeholders (students, faculty, administrators) and must meet a range of functional, security, and performance requirements. The Spiral Model allows for the identification and mitigation of risks such as security vulnerabilities, data integrity issues, and integration problems as they arise.
- 8.2.3. User Feedback Integration:** The Spiral Model allows for continuous user involvement and feedback. Since NICHE is meant to be used by multiple user groups (students, faculty, administrators), regular feedback from these groups during each iteration

ensures that the software meets their needs and addresses any concerns early in the development process.

8.2.4. Adaptability to Changing Requirements: One of the greatest advantages of the Spiral Model is its flexibility in responding to changing requirements. The NICHE system must constantly adapt to the evolving needs of the academic institution, including changes in policies, regulations, and technological advancements. By using the Spiral Model, the software can evolve to meet these needs without starting the development process from scratch.

8.2.5. Focus on Prototyping: The Spiral Model incorporates rapid prototyping, which allows for the early visualization of features, enabling stakeholders to interact with prototypes of the software and provide feedback. This is particularly important for NICHE software because administrators and faculty may need to review various modules (e.g., Finance, Examination) early on to ensure they meet institutional needs.

8.2.6. Incremental Delivery: Instead of delivering the entire system at once, the Spiral Model allows for incremental deliveries. This ensures that each module of the NICHE software is deployed, tested, and refined in smaller, manageable chunks. This reduces the time to market for each module and ensures that the software is continuously evolving to meet real-world needs.

8.3. Phases of the Spiral Model for NICHE Software Development

Each phase of the Spiral Model involves a set of activities that are crucial for the success of the project

8.3.1. Planning Phase:

- Define requirements for each module (Admission, Academics, Finance, etc.).
- Set project goals, milestones, and deadlines.
- Identify stakeholders and establish communication channels for regular feedback.

8.3.2. Risk Analysis Phase:

- Analyse risks such as system integration challenges, data migration, and security concerns.
- Develop strategies to mitigate these risks (e.g., using encryption for sensitive data, ensuring cross-module compatibility).

8.3.3. Engineering Phase (Development & Prototyping):

- Design and develop prototypes for different modules.
- Implement each module incrementally, starting with core functionalities such as the Admission module or Finance module.
- Integrate the modules into the overall system architecture.

8.3.4. Testing & Evaluation Phase:

- Test each module for functionality, usability, and security.
- Validate performance against requirements (load testing, scalability testing).
- Evaluate user feedback and incorporate changes.

8.3.5. Feedback Phase:

- Gather feedback from stakeholders (students, faculty, administrators).
- Identify issues, gather additional feature requests, and refine the design for future iterations.
- Update the project plan based on feedback and lessons learned.

8.3.6. Deployment & Maintenance Phase:

- Deploy modules in a phased manner across the institution.
- Monitor performance, security, and user experience.
- Provide ongoing maintenance and updates based on changing needs and feedback.

8.4. Advantages of the Spiral Model for NICHE Campus Management Software

- **Risk Reduction:** By addressing potential issues early in the development cycle and continuously refining the system, the Spiral Model reduces the likelihood of major project failures or costly delays.
- **Flexibility:** The ability to adjust and change course during each iteration allows NICHE software to adapt to new requirements, technological advancements, and user feedback.
- **Better Quality:** With continuous testing and refinement, each version of the system is of higher quality, ensuring fewer errors and better overall performance.
- **User-Centric Development:** By incorporating user feedback regularly, the software is more likely to meet the needs of students, faculty, and administrators, resulting in higher satisfaction.

IX. FUNCTIONAL REQUIREMENTS

The functional requirements describe the specific behaviour or functions of the system. The modules and their features are detailed below.

9.1. Admission Module

- **Entrance Sub-module:**
 - Allows students to apply for entrance exams online.
 - Provides scheduling and reporting features for entrance exams.
- **Admission Sub-module:**
 - Collects and processes student application data (personal information, academic records, documents).
 - Manages fee payments and provides status updates to applicants.
 - Generates admission reports and tracks progress.

9.2. Academics Module

- **Time Table Sub-module:**
 - Allows administrators to define and manage class schedules.
- **Course Management Sub-module:**
 - Allows administrators and faculty to create and manage courses.
- **Student Profile Sub-module:**
 - Manages student personal information, academic performance, and attendance records.
- **Attendance Management Sub-module:**
 - Tracks student attendance and generates attendance reports.

9.3. Finance Module

- **Reports Sub-module:**
 - Tracks fee payments (tuition, exam fees, etc.).
 - Generates financial reports, such as payment receipts, fee balance statements, and daily collection statements.
- **Payment Sub-module:**

- Handles payment processing for various fees, including online payments.
- Supports multiple payment methods (credit/debit cards, online bank transfers, etc.).

9.4. Examination Module

- **Exam Scheduling Sub-module:**
 - Allows administrators to schedule exams and assign invigilators.
 - Generates exam schedules and manages exam hall assignments.
- **Results Sub-module:**
 - Manages the entry of exam results.
 - Allows students to view their results online.

9.5. Human Resources Module

- **Employee Profile Management:**
 - Manages employee personal and professional details.
 - Tracks employee attendance, payroll, and performance.
- **Payroll Management:**
 - Calculates and processes employee salaries and benefits.
 - Generates payslips and tax reports.

9.6. Hostel Management Module

- **Hostel Management:**
 - Manages room allocation, occupancy, and hostel-related fees.
 - Allows students to request accommodations and tracks check-ins/check-outs.

9.7. Administration Module

- **Settings Management:**
 - Allows administrators to configure and manage academic calendars, policies, and financial settings.
 - Provides a centralized platform for managing institutional settings.

X. SPECIFICATIONS OVERVIEW

10.1.PHP

10.1.1. Overview of PHP

PHP is a widely used open-source server-side scripting language that is primarily used for web development. It is a powerful tool for creating dynamic and interactive web pages. PHP is highly popular due to its simplicity, flexibility, and ability to integrate seamlessly with various web technologies, making it a top choice for web developers. The language can be embedded directly into HTML, which makes it highly versatile for creating dynamic content.

10.1.2. Key Features of PHP

- **Server-Side Scripting:** PHP code runs on the server and generates dynamic HTML content for the client. This allows developers to create interactive web applications that respond to user actions or requests.
- **Ease of Use:** PHP is designed to be easy to learn and use. With simple syntax and built-in functions, PHP allows rapid development of web applications.
- **Database Integration:** PHP excels in database-driven applications. It supports many databases, especially MySQL, which is widely used in combination with PHP. It allows seamless integration with databases for tasks such as storing, retrieving, and updating data.
- **Cross-Platform Compatibility:** PHP is cross-platform, which means it can run on different operating systems such as Windows, Linux, macOS, and others. It can be deployed on multiple servers, including Apache and Nginx.
- **Open Source:** PHP is free to use and has a large community of developers contributing to its growth and improving its features. The open-source nature of PHP allows organizations to use it without licensing costs.
- **Security:** PHP provides robust security features such as data encryption, form validation, and secure connection handling for building secure applications.

10.1.3. PHP in Campus Management Software

PHP plays a central role in the development of the NICHE Campus Management Software. Here are the ways in which PHP is used in the system:

- **Web-based Interface:** PHP is used to build the web-based interface for various campus management functionalities like student admission, course registration, and fee payments.
- **Form Processing:** PHP handles form submissions, including student details, academic information, fee payment details, and more, ensuring smooth and secure data transfer between the client and the server.
- **Database Communication:** PHP communicates with the database to store and retrieve data. It processes requests such as retrieving student records, updating academic results, and generating reports.
- **Session Management:** PHP manages user sessions for administrators, faculty, and students, ensuring that user data is preserved across different pages and actions.

10.2.DBMS (Database Management System)

10.2.1. Overview of DBMS

A Database Management System (DBMS) is a software that manages and organizes data in a structured way. It provides an interface for users to interact with the data, allowing them to store, retrieve, update, and manage the information effectively. DBMS ensures the integrity, security, and consistency of data, which is essential for any application that deals with large volumes of data.

10.2.2. Types of DBMS

DBMS can be categorized into four types based on their data models:

- **Hierarchical DBMS:** Data is organized in a tree-like structure.
- **Network DBMS:** Data is organized in a graph structure, where each node can have multiple relationships.
- **Relational DBMS (RDBMS):** Data is organized into tables with rows and columns. Each table represents an entity, and relationships between entities are established through keys. Examples include MySQL, PostgreSQL, and Oracle.
- **Object-Oriented DBMS:** Data is stored as objects, similar to object-oriented programming principles.

For the NICHE Campus Management Software, Relational Database Management System (RDBMS) is the preferred choice due to its ability to manage structured data efficiently.

10.2.3. Key Features of DBMS

- **Data Integrity:** DBMS ensures that the data is accurate, consistent, and free from errors by enforcing constraints such as primary keys, foreign keys, and unique constraints.
- **Data Security:** DBMS offers advanced security features, including access control, encryption, and authentication, to protect sensitive data.
- **Data Retrieval:** DBMS provides powerful querying capabilities, allowing users to retrieve data using Structured Query Language (SQL). It can execute complex queries efficiently, which is essential for generating reports and records and managing student data.
- **Backup and Recovery:** DBMS systems support backup and recovery mechanisms to prevent data loss and ensure data availability even in case of system failures.
- **Concurrency Control:** DBMS ensures that multiple users can access and modify the database simultaneously without causing conflicts or data inconsistencies.
- **Data Redundancy Elimination:** DBMS reduces redundancy by organizing data into tables, thereby ensuring that the same information is not stored in multiple places, which improves storage efficiency.

10.2.4. Role of DBMS in NICHE Campus Management Software

In the NICHE Campus Management Software, the DBMS serves as the central repository for all campus-related data, including student records, academic information, fee payments, exam results, staff details, and more. Here are the ways the DBMS is utilized in the system:

- **Data Storage:** All records related to admissions, academic courses, finance transactions, and examinations are stored in the database.
- **Data Retrieval:** The system retrieves and displays data on user interfaces, such as student profiles, academic results, attendance, and financial statements.
- **Data Integrity and Security:** The DBMS ensures that only authorized users can access or modify the data, providing the necessary access control mechanisms for students, faculty, and administrators.
- **Report Generation:** The DBMS is used to generate various reports, such as fee payment reports, exam result reports, student attendance reports, and more. It allows administrators to query the data and generate custom reports for analysis and decision-making.

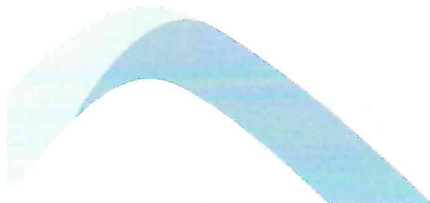
10.2.5. Database Choice for NICHE Campus Management Software

For the NICHE Campus Management Software, the database choice is typically MySQL (or any other suitable relational DBMS), which is a widely used open-source RDBMS known for its speed, reliability, and ease of use. It works seamlessly with PHP and is highly scalable, making it suitable for handling a large volume of data in a campus management system.

10.3. Benefits of Using PHP and DBMS

- **Seamless Integration:** PHP and MySQL (or other DBMS) integrate well together, allowing smooth interaction between the web application and the database.
- **Efficiency:** The combination of PHP for the backend and DBMS for data storage ensures that the system is both fast and efficient.
- **Scalability:** PHP and DBMS systems can scale to handle increased user loads and growing datasets, making it a viable solution for expanding campus management needs.
- **Cost-Effective:** As both PHP and MySQL are open-source, they offer a cost-effective solution for developing and maintaining the system.

By leveraging PHP and a relational DBMS like MySQL, the NICHE Campus Management Software ensures efficient, secure, and reliable management of campus operations.



XI. DATABASE DESCRIPTION FOR NICHE CAMPUS MANAGEMENT SOFTWARE

The **NICHE Campus Management Software** relies heavily on a well-structured and organized database to store, manage, and retrieve data efficiently. Below is a detailed description of the database structure, including the types of data stored, key tables, and relationships between them.

11.1. Database Overview

The database for NICHE Campus Management Software is designed to handle various aspects of institution management, including admissions, academics, finance, human resources, hostel management, and examination processes. It serves as the central repository for all data associated with these functions.

The database is relational, and the key components include:

- **Students:** Personal information, academic details, attendance, and performance.
- **Faculty:** Information on faculty members, their courses, schedules, and performance.
- **Courses:** Course details, schedules, registrations, and grades.
- **Finance:** Fee payments, billing, receipts, and financial reports.
- **Examinations:** Exam schedules, question papers, grades, and results.
- **HR:** Employee records, payroll, and attendance.
- **Hostel:** Student accommodation details, room allocation management.

The relational database structure uses **primary keys** to uniquely identify records in each table and **foreign keys** to establish relationships between tables.

11.2. Overview of Major Tables

11.2.1. Admissions and Student Management

- **applicant:** Contains information about applicants, including personal and educational background details.
 - **Fields:** app_no, name, email, dob, qualification, status
- **student:** Holds the primary details of enrolled students, covering both academic and personal information.

- **Fields:** roll_no, name, department_code, course_code, dob, batch, email, mobile_no
- **candidate_qualification:** Manages records of qualifications submitted by candidates during the admission process.
 - **Fields:** app_no, qualification_type, year_of_passing, institution

11.2.2. Academics and Course Management

- **academic_calendar:** Stores key dates and events for academic scheduling, including semester dates.
 - **Fields:** academic_year, semester, start_date, end_date
- **course:** Manages information on courses offered by the institution, such as capacity, prerequisites, and eligibility.
 - **Fields:** course_code, course_name, capacity, department_code
- **course_paper:** Lists the subjects associated with each course, with details on grading and evaluation criteria.
 - **Fields:** course_code, paper_code, semester, internal_pass_mark, external_pass_mark

11.2.3. Attendance Management

- **daily_attendance:** Logs attendance on a daily basis for students across subjects.
 - **Fields:** date, roll_no, subject_code, status
- **attendance_percentage:** Stores calculated attendance percentages for students.
 - **Fields:** roll_no, semester, subject_code, attendance_percentage

11.2.4. Examinations Management

- **exam_hall:** Lists exam halls, detailing their capacity and available seats.
 - **Fields:** hall_code, name, capacity, available_capacity
- **exam_hall_allocation_header:** Stores main details for exam hall allocations, linking dates, sessions, and halls.
 - **Fields:** code, exam_year_month, exam_date, hall_code, session
- **final_result:** Contains final exam results, covering both internal and external marks, and revaluation status.

- **Fields:** roll_no, semester, subject_code, internal_mark, external_mark, result_status

11.2.5. Human Resources (HR) Management

- **staff:** Holds detailed records of faculty and administrative staff, covering qualifications, personal details, and employment history.
 - **Fields:** staff_id, name, designation, department, qualification, email, contact
- **staff_attendance:** Tracks daily attendance of staff members.
 - **Fields:** staff_id, attendance_date, status
- **payroll_ledger:** Manages payroll entries, detailing earnings, deductions, and final payment amounts.
 - **Fields:** staff_id, pay_code, basic_pay, deductions, final_payment

11.2.6. Hostel Management

- **hostel:** Stores information about hostel blocks and their overall capacity.
 - **Fields:** hostel_id, block_name, total_capacity
- **hostel_room:** Manages individual room details, including availability and capacity.
 - **Fields:** room_no, hostel_id, capacity, available_capacity
- **hostel_allocation:** Tracks which students are assigned to specific rooms.
 - **Fields:** hostel_id, roll_no, room_number, start_date, end_date

11.2.7. Key Fields in Some Tables

- **Student Information (student):** roll_no, name, department_code, course_code, dob, batch, email, mobile_no
- **Course Details (course):** course_code, course_name, semester, internal_pass_mark, external_pass_mark
- **Exam Hall Allocation (exam_hall_allocation_header):** code, exam_date, session, hall_code
- **Final Exam Results (final_result):** roll_no, semester, subject_code, internal_mark, external_mark, result_status

11.3. Database Relationships

- **One-to-Many Relationships:**
 - A course (course) has multiple papers (course_paper).
 - A student (student) has multiple attendance records (daily_attendance).
 - One faculty member (staff) can teach multiple courses (course).
- **Many-to-Many Relationships:**
 - A student can enroll in multiple courses (course_paper), and each course can have many students.

11.4. Security and Integrity

- **Primary Keys:** Uniquely identify records across tables (e.g., roll_no in student, course_code in course).
- **Foreign Keys:** Establish relationships between tables to maintain referential integrity (e.g., department_code in staff references department).
- **Access Control:** Role-based access for restricted access to sensitive data (e.g., payroll details, grades).

XII. SYSTEM TESTING

System testing ensures the software works as expected, is free from bugs, and meets user requirements. Testing is performed in several stages

12.1. Unit Testing:

- Testing individual components or units of the software (e.g., database queries, form validation, report generation) for expected behavior.
- This is carried out by developers using PHPUnit or similar testing frameworks.

12.2. Integration Testing:

- Testing the interaction between modules, ensuring that data flows smoothly between different components like student registration, course management, and finance.
- This ensures that the integrated system functions as a whole.

12.3. System Testing:

- Testing the entire system to verify it meets the functional requirements. This includes testing the user interface, backend functionality, database queries, and overall performance.
- Testing is done by a QA (Quality Assurance) team using tools like Selenium or Postman for automated testing.

12.4. User Acceptance Testing (UAT):

- The system is tested by actual users (faculty, students, HR, etc.) to ensure that it meets their needs and is user-friendly.
- Feedback is collected and used to make any final adjustments to the system.

12.5. Performance Testing:

- Load testing is performed to ensure the system can handle a large number of concurrent users, especially during peak usage periods (e.g., during registration or exam periods).

XIII. USER TRAINING

Training is an essential part of system implementation. The following steps are taken to ensure users are comfortable using the system:

13.1. Admin and Faculty Training: Admins and faculty are trained on how to manage student data, generate reports, process attendance, manage finances, and oversee student course registration.

13.2. Student Orientation: Students are provided with instructions on how to register for courses, pay fees, and use the system's features relevant to them.

XIV. SECURITY IN SOFTWARE FOR NICHE CAMPUS MANAGEMENT SOFTWARE

The security of the NICHE Campus Management Software is crucial as it handles sensitive information such as student records, financial transactions, employee data, and academic results. The system is designed to incorporate multiple layers of security to safeguard data integrity, ensure privacy, and prevent unauthorized access. Below are the key security features and measures implemented in the software:

14.1. User Authentication and Authorization

- **Role-Based Access Control (RBAC):** The system utilizes role-based access control to ensure that users have access to only the information and features relevant to their roles. Users are categorized into different roles (e.g., Admin, Faculty, Students, HR, Finance) and granted specific permissions accordingly.
 - **Admins** have full access to the system's functionality, including user management and system configurations.
 - **Faculty and Staff** have access to student records, grades, attendance, and other relevant academic data.
 - **Students** can only access their personal information, course registration, and exam results.
 - **HR/Finance** have access to employee records, payroll, and student fee information.
- **Secure Authentication:** The system uses a secure login process to authenticate users before granting access. This includes:
 - **Password Protection:** Users must provide a valid username and password to access the system. Passwords are stored in an encrypted format to protect against unauthorized access.

14.2. Data Encryption

- **Data Encryption at Rest:** All sensitive data, including student records, employee details, financial transactions, and other personal information, is encrypted while stored in the database using industry-standard encryption algorithms. This ensures that even if data is accessed or stolen, it remains unreadable and protected.

- **Data Encryption in Transit:** Communication between the client (user's browser) and the server is encrypted using SSL/TLS protocols. This ensures that data transferred between the user's device and the server (e.g., login credentials, student records, payments) is encrypted and protected from interception during transmission.

14.3. Secure Database Management

- **Database Security:** The MySQL database is secured using best practices, including:
 - **Strong Authentication:** Only authorized personnel with a secure database login can access the database.
 - **Least Privilege Principle:** Database users are granted only the minimum necessary permissions, reducing the risk of unauthorized access.
 - **Regular Backups:** Regular backups are taken to prevent data loss and allow recovery in case of a security breach or system failure.
- **SQL Injection Prevention:** The system employs secure coding practices to prevent SQL injection attacks. This includes:
 - Using **Prepared Statements** with parameterized queries to ensure that user inputs are properly sanitized and validated before being passed to the database.
 - Avoiding dynamic SQL queries, which could allow malicious users to inject harmful SQL code into the system.

14.4. Session Management

- **Session Security:** Sessions are used to track authenticated users and maintain their logged-in status. To enhance session security:
 - **Session Timeout:** Users are automatically logged out after a period of inactivity (e.g., 15 minutes) to prevent unauthorized access if a user leaves their device unattended.
 - **Session Encryption:** Session data is encrypted and stored securely to prevent tampering or session hijacking.
- **Secure Cookies:** The system uses secure cookies to store session IDs, which are transmitted only over HTTPS and have the HttpOnly flag enabled. This prevents attackers from accessing session data via client-side scripts.

14.5. Audit Logs and Monitoring

- **Audit Trails:** The system maintains comprehensive logs of all user activities, including login attempts, data modifications, and system access. This provides a record of actions for audit purposes and helps track potential security breaches or unauthorized activity.
 - **Login History:** The system tracks and logs user login attempts, including failed login attempts, IP addresses, and timestamps, to detect any suspicious login behavior.
 - **Activity Logs:** Changes to sensitive data (e.g., student records, fee payments) are logged with details such as the user's identity, the change made, and the timestamp.
- **Real-Time Monitoring:** The system is equipped with monitoring tools to detect and respond to security threats in real-time. These tools can help identify abnormal patterns of access, unauthorized login attempts, and potential security breaches, allowing administrators to take quick action.

14.6. Data Privacy and Compliance

- **Data Minimization:** The system only collects and stores the minimum amount of personal data necessary to fulfill its functionality. This helps minimize the risk of exposure in case of a breach.
- **Data Anonymization:** In cases where sensitive personal information is not required for processing, the system uses data anonymization techniques to ensure that no personal identifiers are exposed or stored unnecessarily.

14.7. Regular Security Updates and Patching

- **Security Patches:** The software and underlying server infrastructure are regularly updated to patch known security vulnerabilities. This includes updates to the PHP framework, database server, web server, and any third-party libraries integrated into the system.
- **Vulnerability Scanning:** Periodic security scans and penetration testing are conducted to identify and fix potential vulnerabilities before they can be exploited by attackers.

14.8. Backup and Disaster Recovery

- **Data Backup:** Regular backups of critical data (including student records, financial data, and user settings) are taken and stored securely. These backups are encrypted to ensure data confidentiality.
- **Disaster Recovery Plan:** A disaster recovery plan is in place to ensure the availability of the system in the event of a security breach or system failure. This includes restoring the most recent secure backups and ensuring the system can be restored to normal operation in a timely manner.

XV. MAINTENANCE IN SOFTWARE FOR NICHE CAMPUS MANAGEMENT SOFTWARE

Maintenance of the NICHE Campus Management Software is a critical aspect of ensuring that the system continues to function effectively, remains secure, and adapts to changing institutional needs. The maintenance process includes activities that help keep the software up-to-date, improve its performance, resolve issues, and enhance its functionality. Below are the key maintenance activities involved in the NICHE Campus Management Software.

15.1. Types of Maintenance

The maintenance of the system is categorized into the following types:

- **Corrective Maintenance:** Corrective maintenance focuses on fixing issues or defects that may arise in the system, including bugs, errors, and malfunctions. This type of maintenance is necessary to ensure that the software continues to function as expected and resolves problems reported by users.
- **Adaptive Maintenance:** Adaptive maintenance involves modifying the software to accommodate changes in the system environment or user requirements. This includes adapting the software to changes in the hardware, operating systems, or third-party software libraries. It also includes updates required due to changes in laws, regulations, or business practices.
- **Perfective Maintenance:** Perfective maintenance aims at improving the software's performance and functionality. It involves enhancements to existing features, optimizing code, improving user interface design, and adding new features or modules that improve the overall user experience.
- **Preventive Maintenance:** Preventive maintenance is performed to identify and resolve potential issues before they can cause major problems. This includes tasks such as conducting security audits, updating software components, optimizing the database, and ensuring system backups are taken regularly.

15.2. Maintenance Activities

The specific activities involved in the maintenance of NICHE Campus Management Software include:

15.1.1. Regular Software Updates

- **Patch Management:** The system will be periodically updated with security patches, bug fixes, and minor enhancements. These updates ensure that the software remains secure and functions without errors. Updates may include patching the application, database management systems, and any third-party libraries or frameworks used in the system.
- **Version Upgrades:** As new versions of the software's framework, programming languages (e.g., PHP), and database management system (e.g., MySQL) are released, the system will be upgraded to ensure compatibility with the latest features, security fixes, and performance improvements.

15.1.2. Bug Fixing and Error Resolution

- **Bug Tracking:** Any bugs or errors reported by users or identified by the system administrators are logged in a bug-tracking system. These bugs are then prioritized and fixed in the appropriate maintenance cycle.
- **Error Logs Review:** Regular reviews of error logs and system logs will be conducted to detect anomalies or failures in the system. Any errors related to user authentication, database queries, payment processing, or other critical areas will be promptly addressed to prevent service disruption.

15.1.3 Performance Optimization

- **Database Optimization:** Over time, as the amount of data increases, the database may experience performance issues. Regular database optimization activities, such as indexing, query optimization, and clean-up of obsolete records, will be conducted to ensure efficient data retrieval and better system performance.
- **Code Optimization:** The software's codebase will be periodically reviewed and optimized for performance. This includes refactoring inefficient code, removing redundant processes, and ensuring that the software is running as efficiently as possible.
- **Load Testing:** Periodic load testing will be conducted to simulate high user traffic and ensure that the system can handle peak loads without compromising performance. Any identified performance bottlenecks will be addressed.

15.1.4. Security Maintenance

- **Security Patches:** The software will receive regular security updates to fix any vulnerabilities identified by the development team or through external security audits. Security patches for the underlying server, PHP, and other related software will be applied promptly.
- **Security Audits:** Regular security audits will be conducted to assess the system for vulnerabilities, such as potential SQL injection points, weak encryption, or other security risks. A vulnerability scanning tool will be used to proactively detect any weaknesses.
- **Data Backups:** Regular backups of the system's database and critical data will be scheduled and stored securely. Backup schedules will be defined to ensure that data can be recovered in case of any system failure or data corruption.

15.1.5. User Support and Feedback Management

- **User Support:** A dedicated support team will be available to address issues reported by users, whether they are related to functionality, user experience, or bugs. Support may include answering user queries, troubleshooting problems, and providing guidance on using the system.
- **User Feedback Analysis:** Feedback from end users (administrators, faculty, students, and staff) will be continuously collected and analyzed to identify areas for improvement or new features to be added. This helps the software adapt to changing user needs and improve overall user satisfaction.

15.1.6. Documentation Updates

- **System Documentation:** The system's technical and user documentation will be regularly updated to reflect any changes made during the maintenance process, such as new features, updated workflows, or changes in security procedures. This ensures that both developers and users have access to accurate and up-to-date information.
- **User Training and Manuals:** As new features or changes are implemented, the training materials and user manuals will be updated. This includes creating new tutorials, guides, and training sessions for administrators, faculty, and students to ensure they can effectively use the system.

15.7. Maintenance Schedule

A formal maintenance schedule will be established, detailing the frequency of different types of maintenance activities:

- **Daily Maintenance:** Monitoring system health, database backups, and error log reviews.
- **Weekly Maintenance:** Performance optimization tasks and bug fixing.
- **Monthly Maintenance:** Security audits, patch application, and user feedback review.
- **Quarterly Maintenance:** Major version upgrades, load testing, and system health checks.

15.8. Benefits of Ongoing Maintenance

Ongoing maintenance provides several key benefits:

- **Enhanced Security:** Regular updates and security patches ensure that the system is protected against evolving cyber threats.
- **Improved Performance:** Optimizations ensure that the system runs efficiently and can handle increasing user loads.
- **User Satisfaction:** Regular updates based on user feedback ensure that the software evolves to meet user needs.
- **Reduced Downtime:** Proactive maintenance helps prevent system failures and ensures that the system remains operational at all times.

CONCLUSION

The NICHE Campus Management Software is a comprehensive solution designed to streamline and automate the administrative, academic, financial, and support. The software's modular architecture, encompassing areas such as Admissions, Academics, Finance, Examinations, Human Resources, Hostels, and Fleet management, ensures that all aspects of campus operations are managed efficiently and effectively.

The system leverages modern technologies such as PHP and MySQL, ensuring high performance, scalability, and flexibility. The integration of advanced features like real-time updates, secure payment gateways, attendance tracking, and performance analysis tools contributes to a seamless user experience for both staff and students. The inclusion of detailed reports and dashboards enhances decision-making capabilities and operational transparency.

Throughout the development and implementation phases, emphasis has been placed on security, reliability, and user-friendliness. Rigorous testing procedures, including bug fixes, system optimization, and regular security updates, are in place to ensure that the system continues to meet the evolving needs of the institution.

Furthermore, the software's ongoing maintenance and support mechanisms are designed to address emerging requirements, adapt to changes in technology, and resolve any issues promptly. Regular updates, user feedback integration, and proactive security audits guarantee that the software remains secure, efficient, and aligned with best practices.

In essence, the NICHE Campus Management Software provides a powerful tool to enhance operational efficiency, improve student and staff experiences, and ensure smooth functioning across all areas of campus management. As technology continues to evolve, this software will serve as a robust platform for supporting future innovations and adapting to the dynamic needs of the institution.

XVII. LIMITATIONS

While the NICHE Campus Management Software offers a comprehensive and integrated solution for managing various aspects of campus operations, there are certain limitations that should be considered:

- 17.1. Dependency on Internet Connectivity:** As the software is web-based, a stable internet connection is essential for its optimal functioning. In areas with unreliable internet infrastructure, users may face connectivity issues, which could disrupt access to critical functionalities.
- 17.2. Training and Adaptability:** Despite the system's user-friendly interface, there may be a learning curve for staff and faculty, especially those who are not tech-savvy. Comprehensive training and ongoing support are required to ensure smooth adoption across the institution.
- 17.3. Limited Mobile Functionality:** The current version may not fully support mobile-based functionalities, such as mobile apps or responsive design, which could limit access for users who prefer using smartphones or tablets over desktop systems.

XVIII. FUTURE IMPROVEMENTS

While the NICHE Campus Management Software is designed to meet the current needs of educational institutions, there are several areas where improvements can be made to enhance its functionality, scalability, and user experience. These potential improvements can ensure that the software remains relevant and adaptable to future technological advancements and institutional requirements:

- 18.1. Mobile Application Development:** Developing a dedicated mobile application for students, faculty, and administrators would enhance accessibility and enable users to access the system from anywhere at any time. Mobile-based features like notifications, course updates, attendance tracking, and fee management would improve the user experience.
- 18.2. Artificial Intelligence (AI) Integration:** AI can be integrated to enhance the functionality of the system. AI-driven analytics could help in predicting student performance, tracking behavioral trends, and even suggesting personalized academic or career paths for students. Automated tutoring or chatbots for students and staff queries could be another useful AI-based addition.
- 18.3. Advanced Data Analytics and Reporting:** Introducing more advanced data analytics capabilities could enable institutions to generate deeper insights into student performance, financial trends, and academic outcomes. Interactive dashboards and AI-powered reporting tools could be integrated to help administrators make data-driven decisions.
- 18.4. Seamless Integration with Other Educational Platforms:** To enhance the software's utility, integration with other widely-used educational tools, such as Learning Management Systems (LMS), Student Information Systems (SIS), and Communication platforms (e.g., Zoom, Google Classroom), could improve functionality. This would provide a more holistic system for managing all educational and administrative needs in one place.

APPENDICES

APPENDIX I: SCREENSHOTS OF DATABASE TABLES

Schemata	Bookmarks	History	Schemata	Bookmarks	History	Schemata	Bookmarks	History
 - niu_ver1 - academic_calendar - academic_calendar_list - academic_year - alarm_setting - alumni - applicant - applicant_list - applicant_qualification - application_form - attendance_log - attendance_mark - attendance_mark_intern - attendance_percentage - attendance_register - au_course - au_course_break - au_course_paper - au_course_time - au_paper - au_staff_paper_list - backup_daily_attendance - backup_fee_ledger - book - book_journal - book_ledger - bus - campaign_header - campaign_lines - campaign_template - candidate - candidate_fee_ledger - candidate_qualification - candidate_student_docs			 - candidate_student_eligibility - caste - chat - company_settings - conference_publication - consumables - consumables_list - course - course_break - course_capacity - course_docs - course_eligibility - course_fee - course_fee_heads - course_paper - course_time - daily_attendance - daily_attendance_mba - daily_attendance_new - dashlets - denomination - department - designation - dictionary - document - eligibility - enquiry - entrance_exam - entrance_exam_settings - equipments - exam_attendance - exam_center - exam_dummysno - exam_hall - exam_hall_allocation_details - exam_hall_allocation_header - exam hall allocation line			 - examination_settings - examiner - examtimetable - fee_balance - fee_ledger - fee_ledger_temp - filters - final_result - final_result_other - final_result_temp - graduated_status - graduation - graduation1 - group_access - group_dashlet - group_header - hall - hall_calender - help - hostel - hostel_attendance_log - hostel_block - hostel_daily_attendance - hostel_room - hostel_student - invoice_header - invoice_lines - item - item_group - item_journal - item_ledger - journal_publication - leave_type - maintainance - menu - menu_old		

NICHE ERP Modules and Functionalities - Configuration and Operations

Schemata | Bookmarks | History

- ▶ misc_paper
- ▶ modules
- ▶ nac_final_result
- ▶ nac_paper
- ▶ nac_pre_final_result
- ▶ nac_serial_test
- ▶ navision
- ▶ no_series
- ▶ on duty_attendance
- ▶ online_exam_headers
- ▶ online_exam_lines
- ▶ online_final_result
- ▶ online_question_options
- ▶ online_questions
- ▶ online_user_answers
- ▶ other_final_result
- ▶ paper
- ▶ paper_book_list
- ▶ paper_other
- ▶ paper_unit
- ▶ payroll_elements
- ▶ payroll_ledger
- ▶ pre_final_result
- ▶ programme
- ▶ programme_branch
- ▶ qualification_points
- ▶ qualifications
- ▶ question_bank
- ▶ question_type
- ▶ range_header
- ▶ range_list
- ▶ resource
- ▶ resource_calender
- ▶ revaluation
- ▶ revaluation_history
- ▶ route
- ▶ route map

Schemata | Bookmarks | History

- ▶ serial_test
- ▶ sgpa
- ▶ sms_queue
- ▶ staff
- ▶ staff_attendance
- ▶ staff_declaration
- ▶ staff_designation
- ▶ staff_header
- ▶ staff_paper_list
- ▶ staff_payroll
- ▶ state
- ▶ student
- ▶ student_arrear
- ▶ student_certificate_history
- ▶ student_document
- ▶ student_exam_details
- ▶ student_exam_details_other
- ▶ student_nac_exam_details
- ▶ student_nac_paper
- ▶ student_online_exam_paper
- ▶ student_paper
- ▶ student_paper_history
- ▶ student_password
- ▶ student_qualification
- ▶ student_remarks
- ▶ student_status
- ▶ subject_calender
- ▶ supplementary_exam
- ▶ tally_server_settings
- ▶ tbl_exam
- ▶ tbl1_exam
- ▶ temp
- ▶ temp_course
- ▶ temp_course_fee
- ▶ temp2
- ▶ tempmobile

Schemata | Bookmarks | History

- ▶ temp_course
- ▶ temp_course_fee
- ▶ temp2
- ▶ tempmobile
- ▶ test
- ▶ test1
- ▶ time_table
- ▶ time_table_history
- ▶ time_table_staff_availability
- ▶ units
- ▶ user_dashlet
- ▶ user_filters
- ▶ user_list
- ▶ user_settings
- ▶ vacancy
- ▶ vender
- ▶ vender_group
- ▶ active_arrears
- ▶ first_student
- ▶ paper_suggest
- ▶ staff_master
- ▶ student_allocation_list
- ▶ student_paper_lists
- ▶ student_paper_lists_phd
- ▶ student_paper_new
- ▶ first_sem_new_store_spga
- ▶ first_sem_store_spga
- ▶ generate_permanent_rollno
- ▶ generate_phd_permanent_roll
- ▶ nac_store_spga
- ▶ set_tcnno
- ▶ store_spga
- ▶ store_spga_1stsem
- ▶ store_spga_regular
- ▶ store_spga_final

APPENDIX II: SCREENSHOTS OF DATABASE FIELDS FROM EACH TABLE

Schemata	Bookmarks	History
student		
slno		
department_code		
course_code		
course_name		
discipline		
app_no_		
roll_no_		
alt_roll_no_		
temp_roll_no		
first_year_no_		
section		
batch		
name		
fathers_name		
guardian_name		
present_address		
present_address2		
present_address3		
present_city_taluk		
present_district		
present_pin_code		
present_state		
permanent_address		
permanent_address2		
permanent_address3		
permanent_city_taluk		
permanent_district		
permanent_pin_code		
permanent_state		
dob		
doa		
doj		
dol		
year_of_passing		
nationality		
sex		

Schemata	Bookmarks	History
tel_no_		
mobile_no_		
email_id		
photo		
aadharbarcode		
qccode		
current_semester		
caste		
religion		
community		
aadharano		
institution_last_studied		
tc_issued_date		
is_tc_issued		
provisional_issued_no		
provisional_issued_date		
course_completion_issued		
course_completion_issued		
conduct_cer_issued_no		
conduct_cer_issued_date		
suspended_start_date		
suspended_end_date		
tc_no		
blood_group		
height		
weight		
allergy		
eyes_vision		
others		
is_hostel		
is_fleet		
is_bus		
HallTicket		
is_parttime		
hostel_block		
hostel_room_no_		

Schemata	Bookmarks	History
fleet_route_no_		
fleet_destination		
status		
student_status		
course_complete_month		
course_complete_year		
is_lateral		
block_attendance		
block_exam		
is_annauniv		
block_hostel		
student_password		
parents_password		
is_calendar_year		
comments		
data_log		
copy_to_staff		
academic_year		
contact		
medium		
is_other_univ		
ts		

Schemata	Bookmarks	History	Schemata	Bookmarks	History	Schemata	Bookmarks	History
pre_final_result sno month_year course_code roll_no_ semester cleared_semester subject_code internal_mark internal_mark_posting_date external_mark external_mark_posting_date grace_mark grace_mark_posting_date mark_100 reevaluation_status result_processed result_publish paper_clared external_insert arrear_attempt paper_cleared data_log star_rating sheetno dummyno dummymonthyear ts			course sno graduation department_code code gra_code name discipline faculty_stream name_desc duration roll_no_key roll_no_lateral_key full_discipline pmark tpmark ts			department sno code name dept_code caption is_teaching ts		

Schemata	Bookmarks	History	Schemata	Bookmarks	History	Schemata	Bookmarks	History
- course_paper - sino - course_code - line_no_ - semester - paper_code - priority - internal_pass_mark - internal_total_mark - external_pass_mark - external_total_mark - ts - course_time - sino - course_code - line_no_ - semester - hour - time - ts - daily_attendance - sino - date - roll_no_ - subject_code - course_code - semester - hour - day - status - ts			- course_paper - sino - course_code - line_no_ - semester - paper_code - priority - internal_pass_mark - internal_total_mark - external_pass_mark - external_total_mark - ts - course_time - sino - course_code - line_no_ - semester - hour - time - ts - daily_attendance - sino - date - roll_no_ - subject_code - course_code - semester - hour - day - status - ts			- exam_hall - sino - code - name - capacity - available_capacity - start_seat_no - end_seat_no - ts - exam_hall_allocation_details - sino - code - date - session - capacity - is_allocate - ts - exam_hall_allocation_header - sino - code - exam_year_month - exam_date - hall_code - hall_name - capacity - available_capacity - session - ts - exam_hall_allocation_line - sino - code - line_no_ - exam_year_month - roll_no_ - paper_code - date - session		

Schemata	Bookmarks	History	Schemata	Bookmarks	History	Schemata	Bookmarks	History
- fee_ledger - sno - type - entry_no_ - roll_no_ - semester - ex_yr_month - tally - pay_code - description - posting_date - final_payment_date - final_payment_date_withfine - fine_charges - fee_amount - invoice_no_ - temp_no - status - ts			- examination_settings - sno - code - exam_year_code - exam_yearmonth - exam_display_yearmonth - examination_id - revaluation_amount - retotal_amount - photocopy_amount - revaluation_last_date - flag - star_rating - ts - supplement_amount_pg - supplement_amount - examiner - examtimetable - sno - exam_yearmonth - paper_code - graduation - semester - session - date - time - ts			- filters - sno - table_name - filter_options - filter_captions - filter_by - ts - final_result - sno - month_year - course_code - roll_no_ - semester - cleared_semester - subject_code - internal_mark - internal_mark_posting_date - external_mark - external_mark_posting_date - grace_mark - grace_mark_posting_date - revaluation_status - result_processed - external_insert - arrear_attempt - data_log - star_rating - ts		

NICHE ERP Modules and Functionalities
- Configuration and Operations

Schemata	Bookmarks	History	Schemata	Bookmarks	History	Schemata	Bookmarks	History
- staff - slno - staff_id - name - middle_name - surname - short_code - father_name - mother_name - DOB - sex - marital_status - department - designation - ug_degree - ug_discipline - ug_univname - ug_regular - pg_degree - pg_discipline - pg_univname - pg_regular - doctoral_degree - phd_discipline - phd_univname - doctoral_regular - other_qualification - other_discipline - other_univname - other_regular - appointment_type - doj - dor - leaving_reason - religion - caste - address			- address_2 - address_3 - city - state - country - tel_no_ - mobile_no_ - email_id - bank_name - is_chairman_list - bank_branch - bank_account_no - pan_number - aadhar_no - pf_no_ - isfc_code - basic_pay - actual_rent_paid - age - experience - research_experience - industry_experience - is_head - register_type - register_priority - vacancy_allocate - avail_cl - is_resource - blocked - student_staff - ts			- course_break - slno - course_code - line_no_ - semester - after_hour - type - break_time - ts - course_capacity - slno - course_code - line_no_ - start_date - capacity - ts - course_docs - slno - course_code - line_no_ - document_code - doc_type - name - ts - course_eligibility - slno - course_code - line_no_ - eligibility_code - description - ts - course_fee - slno - year_of_admission - course_code - line_no_ - fee type		

NICHE ERP Modules and Functionalities
- Configuration and Operations

Schemata	Bookmarks	History	Schemata	Bookmarks	History	Schemata	Bookmarks	History
- attendance_log - slno - attendance_date - semester - hour - course_code - subject_code - flag - ts - attendance_mark - slno - roll_no_ - subject_code - semester - attendance_mark - attendance_percentage - attendance_posting_date - assignment_mark - assignment_posting_date - ts			- group_access - slno - group_id - type - name - access_level - write_access - read_only - hide_tabs - hide_actions - hide_fields - add_new - delete - list_properties - ts - group_dashlet - slno - group_id - line_no_ - id - caption - order - blocked - ts - group_header - slno - group_name - group_id - is_department - department_code - status - ts			- ts - hall - slno - hall_type - block_name - hall_no_ - hall_breath - hall_length - capacity - occupancy - available_capacity - star_rating - ts - hall_calender - slno - date - hall_no_ - session - status - ts		



P. Thirumalvalavan

Dr. P. THIRUMALVALAVAN,
 Registrar
 Noorul Islam Centre for Higher Education,
 Kumaracoil - 629 180, Thuckalay,
 Kanyakumari District.

A. Shajin Nargunam
Prof. Dr. A. Shajin Nargunam, M.E., Ph.D.,
 Director-Academic Affairs
 Noorul Islam Centre for Higher Education
 Kumaracoil, Thuckalay.