University Institute of Information Technology PMAS-Arid Agriculture University, Rawalpindi



Project Proposal For

Title (Campus Management System)

*(Valid Title reflecting scope and objectives)*

**Submitted By:**

## Sanan Ali

## 20-ARID-4197

## Syed Husnain Ahmad Bukhari 20-ARID-4201

## Raveem Sohail 20-ARID-4196

## Syed Raheel Ahmed 20-ARID-4202

**Supervised By:**

## Prof. Salman Bhutti

## Submission Date (14-12-2023)

## (Mention appropriate project proposal Document version)

## Original version/ revised version

Project ID 1

(ID will be assigned by coordinator)

* **Please select the appropriate category of your project** *(Select multiple if required)*

**A-**Desktop Application/Information System

**C-**Web Application/Web Application based Information System.

**E-**Smartphone Application

**G-**Image Processing

**I-** Other: Web Based Application











**B-**Problem Solving and AI **D-**Simulation & Modeling **F-**Smartphone Game

**H-**Networks









**Group Members**

## Student Name: Sanan Ali Registration No: 20-ARID-4197 Class: BSCS 8th Semester

## Shift: Morning

## Email: [sananali010@gmail.com](mailto:sananali010@gmail.com) WhatsApp No: 03116566223

## Student Name: Syed Husnain Ahmad Bukhari Registration No: 20-ARID-4201

## Class: BSCS 8th Semester Shift: Morning

## Email: [bukharisb55@gmail.com](mailto:bukharisb55@gmail.com) WhatsApp No: 03405617500

## Student Name: Raveem Sohail Registration No: 20-ARID-4196 Class: BSCS 8th Semester

## Shift: Morning

## Email: [raveemjutt@gmail.com](mailto:raveemjutt@gmail.com) WhatsApp No: 03096206707

## Student Name: Syed Raheel Ahmed Registration No: 20-ARID-4202 Class: BSCS 8th Semester

## Shift: Morning

## Email: [smraheel1300@gmail.com](mailto:smraheel1300@gmail.com) WhatsApp No: 03094508204

### TABLE OF CONTENTS

1. [Introduction **4**](#_bookmark0)
2. [Literature Review **4**](#_bookmark1)
3. [Problem Statement **5**](#_bookmark2)
4. [Problem Solution **5**](#_bookmark3)
5. [Advantages/Benefits of proposed system **5**](#_bookmark4)
6. [Scope **6**](#_bookmark5)
7. [Software Methodology **7**](#_bookmark6)
8. [Tools and Technologies **7**](#_bookmark7)
9. [Concepts **7**](#_bookmark8)
10. [Intended Users **8**](#_TOC_250000)
11. [Mockups **8**](#_bookmark9)
12. Conclusion **11**

# Introduction

The Campus Management System is like a helpful friend for universities, making all the paperwork and important tasks much easier. This project is here to fix the common issues universities face, like handling student registrations, organizing class schedules, managing teachers, and keeping important records. Imagine using the latest technology to make all these tasks smoother, with fewer mistakes and less manual work.

So, why are we doing this? The main goal is to create a simple and easy-to-use platform where everyone in the university – from the big bosses to the teachers and students – can talk to each other and work together effortlessly. By using this system, we want to make sure all the important academic info is well-managed, the data is accurate, and there's one central place to find everything you need. This project is like a superhero for our university, promising to change how we do things, making them way more organized and efficient.

# Literature Review

#### Existing Campus Management Systems (CMS) and Their Issues:

#### Banner by Ellucian:

Banner by Ellucian is a versatile system handling everything from admissions to payroll. However, it's like navigating a maze due to its complexity. Finding your class schedule can feel overwhelming, especially after a hectic week of finals.

#### Blackboard Learn:

Blackboard Learn acts like a virtual classroom with cool features. It's popular, but students sometimes find it clunky. Some even prefer smoother platforms like Zoom for a better online learning experience.

#### Moodle:

Moodle is an open-source CMS, allowing schools to customize features. Yet, setting it up can be tricky, requiring tech-savvy folks. It's like a blank canvas, but the paint can get messy if you're not careful.

#### Common Issues Across CMS:

* + - **One-size-fits-all Trouble:** CMS struggles to adapt to different schools, resulting in awkward workarounds and frustrated users.
    - **Tech Tangles:** Some systems need tech expertise, making them less friendly for smaller schools with limited IT resources.
    - **Data Silos:** Information gets stuck in separate systems, making it hard to see the whole picture and improve things like graduation rates.
    - **Customization Confusion:** Pre-built features might not fit a school's specific workflows, feeling like they're wearing someone else's shoes.

#### The Takeaway:

Existing CMS systems have their challenges, highlighting the need for a more flexible, user-friendly, and tailored solution. Your project aims to create a CMS that's comfortable and well-equipped, avoiding the maze-like complexity of current systems.

# Problem Statement

Our software solves the problems faced by universities using current systems that are hard to use, struggle to handle growth, and lack good communication. We are creating this system to make university tasks easier and more organized. While similar systems exist, they often have issues, and by re-implementing, we want to learn and improve upon them. Our goal is to understand software design, create a user-friendly experience, and manage information more efficiently.

Developing this project will teach us valuable skills in software development, designing user interfaces, and managing databases. We aim to learn from both the challenges and successes of re- implementing a university management system, gaining practical knowledge in solving real-world problems in the education technology field.

# Problem Solution

Our project solves the university management issues by creating a simple and easy-to-use system. We're making things better by using smart technology that makes tasks quicker and reduces mistakes. Our approach focuses on improving how people interact with the system, making it friendly for everyone.

The solution includes advanced features that handle a growing number of users and tasks efficiently. We're using a smart algorithm to organize information and keep things running smoothly. Our project is like a helpful guide, providing a central hub for all important tasks and communication. By doing this, we're making university management not just easier, but also more enjoyable for everyone involved. The key is to simplify complex processes, and our system does exactly that.

# Advantages/Benefits of proposed system

* **Easy to Use:** Our system is super easy for everyone – from students to professors and administrators.
* **Less Mistakes:** With our system, there's less chance of making errors because it's smart and organized.
* **Faster Tasks:** Tasks like enrolling in courses or managing schedules become quicker, saving everyone's time.
* **Handles More People:** As more students and staff join, our system won't slow down. It can handle lots of users without any issues.
* **Better Communication:** Our system helps everyone talk and work together easily. Professors can chat with students, and administrators can share important info with everyone.
* **Everything in One Place:** All the important things like schedules, grades, and announcements are in one central spot, making it super convenient for everyone.

# Scope

Our project aims to enhance university management by focusing on enrollment, course schedules, and communication. We will develop a user-friendly system accessible to students, professors, and administrators, featuring easy course registration, dynamic scheduling, and a central hub for announcements. Security is a priority, with a robust authentication process to safeguard user data. The scope does not include financial transactions or replace existing financial systems, keeping our focus on usability, efficiency, and communication.

This carefully defined scope ensures each team member's involvement in meaningful tasks, providing a balanced workload for our three-member team. Emphasizing user interface design and efficient communication, our project strives to create a comprehensive and accessible university management solution, contributing to a more organized and user-friendly educational environment.

**Modules of Project:**

1. **User Authentication Module:**
   * This module ensures that only authorized users can access the system.
   * Users, including students, professors, and administrators, will have secure login credentials.
   * It adds an extra layer of protection to keep sensitive information safe.
   * Special Feature: Two-factor authentication for enhanced security.
2. **Enrollment Module:**
   * Streamlines the course registration process for students.
   * Allows students to easily view available courses, select preferences, and enroll.
   * Provides real-time updates on course availability and changes.
   * Special Feature: Intelligent suggestions based on student preferences for efficient course selection.
3. **Dynamic Scheduling Module:**
   * Manages and optimizes the scheduling of classes and events.
   * Allows administrators to easily create, modify, and share dynamic schedules.
   * Ensures efficient use of resources and avoids scheduling conflicts.
   * Special Feature: Automated conflict resolution and real-time updates for all stakeholders.
4. **Communication Hub Module:**
   * Facilitates seamless communication between administrators, professors, and students.
   * Centralizes announcements, updates, and messages for easy access.
   * Supports group messaging and notifications for important events.
   * Special Feature: Integration with email and push notifications for timely communication.
5. **Record Keeping Module:**
   * Manages and organizes student records, grades, and academic history.
   * Allows administrators to easily retrieve and update student information.
   * Provides a secure and centralized database for academic records.
   * Special Feature: Advanced search and reporting capabilities for efficient record retrieval.

# Software Methodology

We're using the Agile method to develop our project because it lets us be flexible and adapt easily to changes. This way, we can work closely with everyone involved and get feedback throughout the process. Agile allows us to deliver the project in smaller parts, making it easier to test and improve. This method is like building with LEGO bricks – we can adjust things as we go and make sure the final project fits everyone's needs. It's a great fit for creating our user-friendly university management system.

# Tools and Technologies

|  |  |  |
| --- | --- | --- |
| **Tools And**  **Technologies** | **Tools** | **Version** |
| MS Visual Studio | 1.85 |
| Git | 2.4 |
| GitHub | 3.11 |
| **Technology** | **Version** |
| React JS | 18.0 |
| JavaScript | 2015 |
| HTML | 5 |
| CSS | 3 |
| Node JS | 21 |
| Mongo DB | 7.0 |

# Concepts

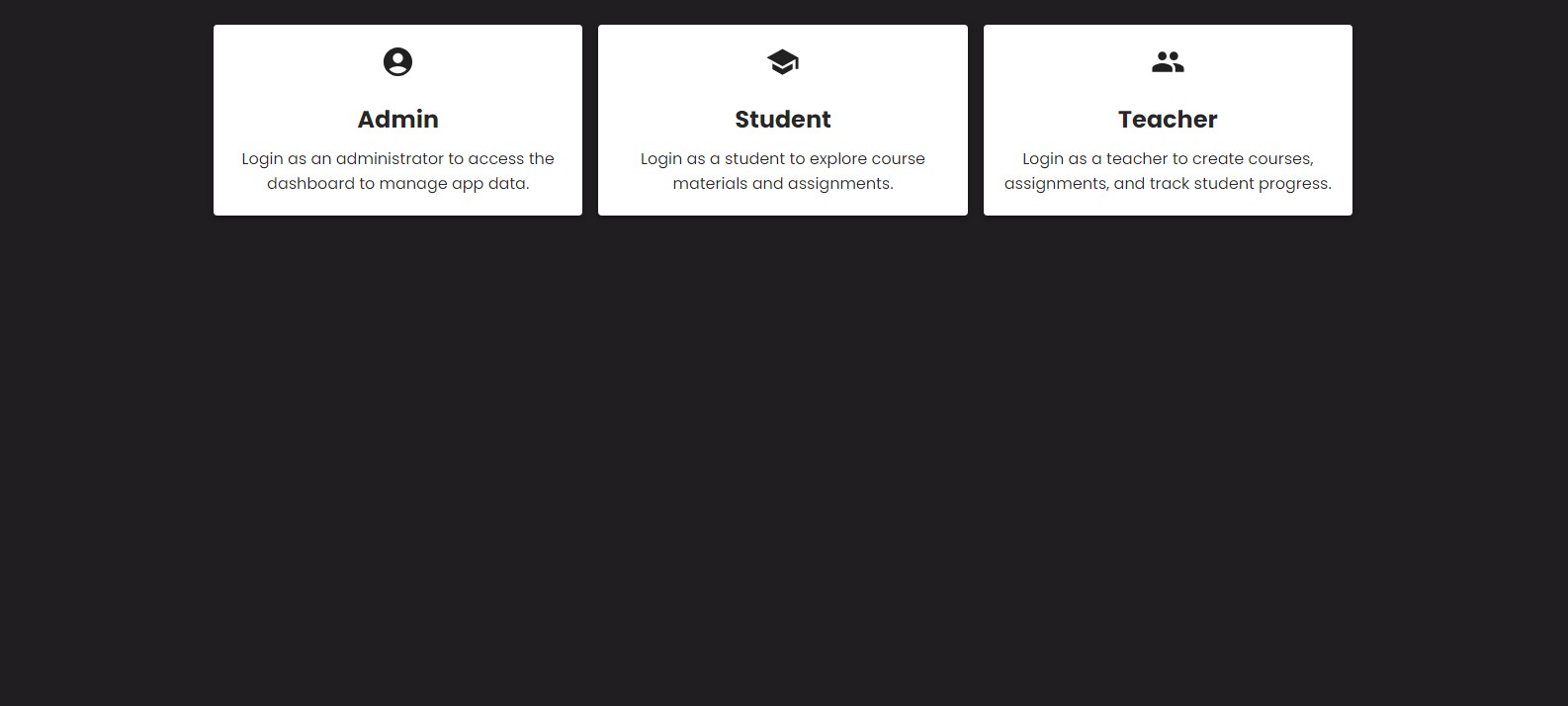
1. **User Authentication**: Understanding how to securely manage and authenticate users, ensuring the system remains safe and only authorized individuals have access.
2. **Database Management:** Learning how to organize and interact with databases, specifically MongoDB or PostgreSQL, to efficiently store and retrieve information.
3. **Responsive Web Design:** Grasping the concept of creating web interfaces that work well on various devices, making it easy for users to interact with the system from their computers or smartphones.
4. **Agile Development:** Embracing the Agile methodology to work in iterative cycles, allowing for continuous improvement, flexibility, and collaboration throughout the project's development.

# Intended Users

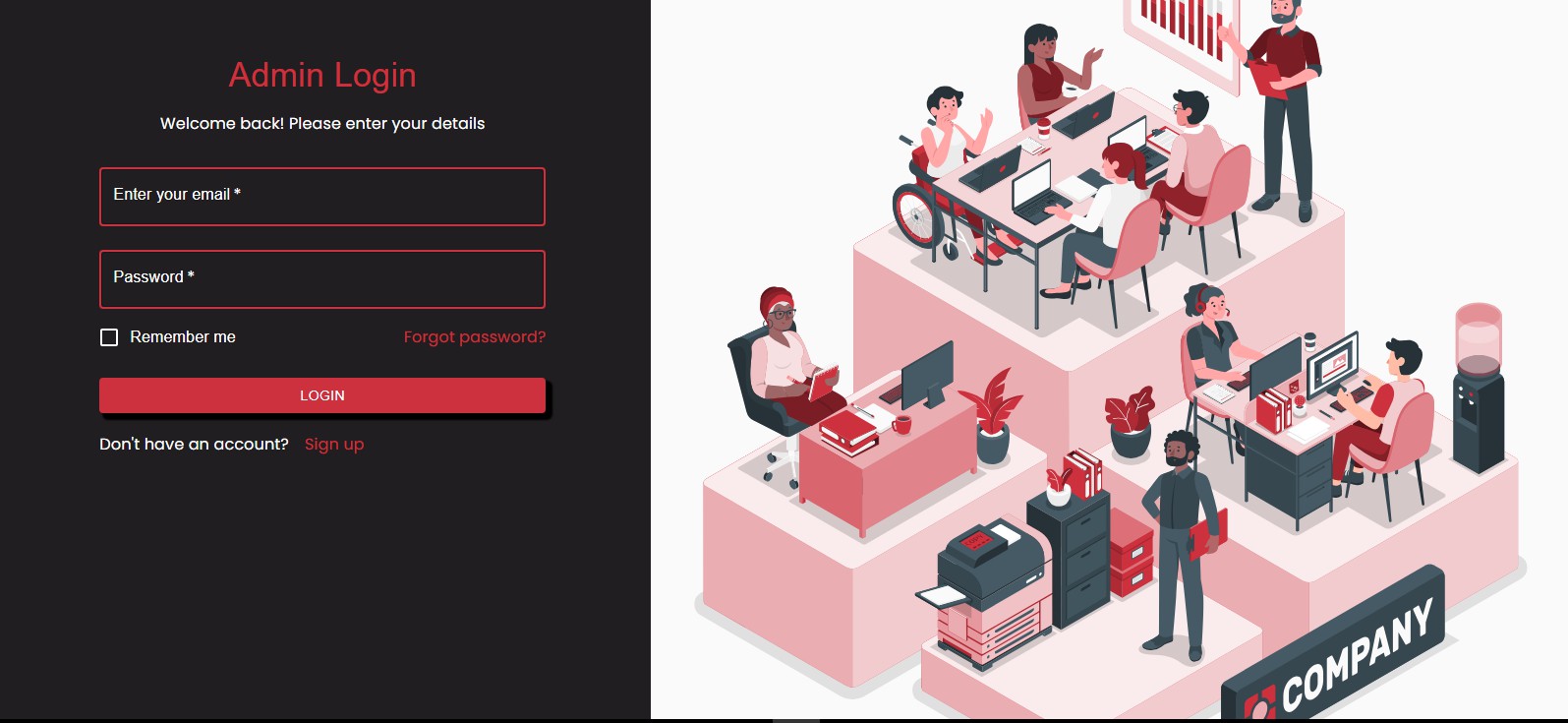
* **Students:** The primary users, who will use the system to enroll in courses, access schedules, and view academic records.
* **Professors:** Faculty members will utilize the system for managing course details, updating grades, and communicating with students.
* **Administrators:** University staff responsible for overseeing the entire system, managing user accounts, and handling system configurations.
* **System Administrators:** Individuals responsible for maintaining and managing the technical aspects of the University Management System, ensuring its smooth operation.

# Mockups Main Page:

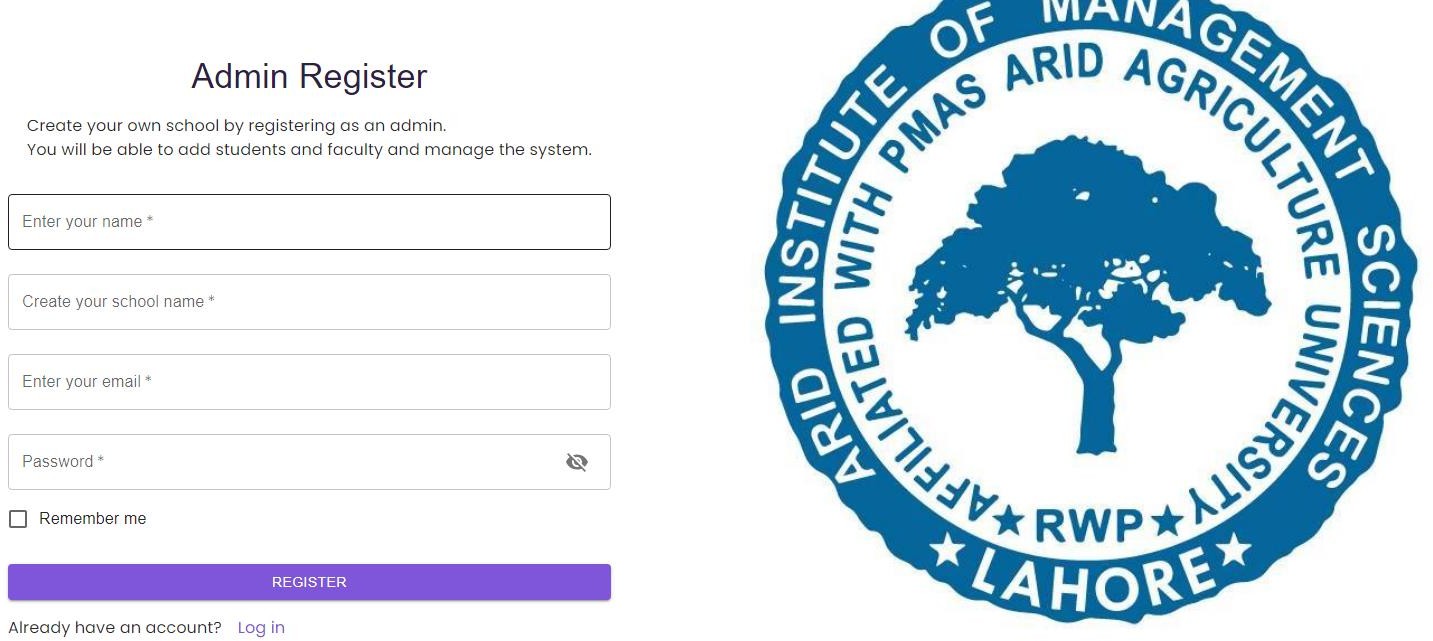
**User Selector:**



**Login Page:**



**Registration Page:**



1. **Timeline**

.

**Project Initiation and Planning**

#### Week 1-2: Project Kickoff

Define project scope, objectives, and key features.

Set up collaboration tools (GitHub, Jira) and communication channels.

#### Week 3-4: Requirement Analysis

Conduct stakeholder meetings to gather detailed requirements. Identify and prioritize project modules and features.

**December 2023: Design and Framework Selection 3.**

#### Week 1-2: System Architecture Design

Plan the overall system architecture.

Decide on the technology stack, considering MERN components.

#### Week 3-4: Design

Create wireframes and mockups for major modules.

Gather feedback from potential users for design improvement.

**January 2024: Development Kickoff**

#### Week 1-2: Front-end Development

Start building the front-end using React.js. Implement basic navigation and layout.

#### Week 3-4: Back-end Development

Initiate the development of the back-end using Node.js and Express.js.

Set up the initial database structure using MongoDB.

**February 2024: Core Functionality Implementation**

#### Week 1-2: Enrollment Module Implementation

Develop the enrollment module with features like course selection and confirmation.

#### Week 3-4: Dynamic Scheduling Module Implementation

Build the dynamic scheduling module, allowing administrators to create and modify schedules.

**March 2024: Advanced Features and Testing**

#### Week 1-2: Communication Hub Module Implementation

Implement the communication hub with features like announcements and group messaging.

#### Week 3-4: Record Keeping Module Implementation

Develop the record-keeping module for managing student records and grades.

**April 2024: Testing, Refinement, and Conclusion 11.**

**Week 1-2: Testing Phase** - Conduct thorough testing of each module for functionality and user experience. - Address any identified bugs or issues.

**Week 3-4: Refinement, Optimization, and Conclusion** Implement user feedback and make necessary refinements. Optimize code, enhance performance, and prepare for deployment. Conduct final testing to ensure everything works smoothly.

Submit the completed project for evaluation and gather feedback.

1. **Conclusion**

In conclusion, our University Management System project aims to revolutionize the way academic institutions handle administrative tasks. With a user-friendly approach and key modules like Enrollment, Dynamic Scheduling, and Communication Hub, the system seeks to streamline processes for students, professors, and administrators. Through the adoption of the Agile methodology, we ensure flexibility and adaptability during development, promoting continuous improvement. The project focuses on creating a more efficient, organized, and enjoyable educational ecosystem.

1. **References**

* Online tutorials and documentation for React.js, Node.js, and MongoDB.
* Insights and inspiration drawn from similar projects on collaborative platforms like GitHub.
* Various coding forums and communities providing solutions to specific programming challenges.
* Stack Overflow discussions contributing to problem-solving and code optimization.

**Evaluation Page**

#### Evaluator 1 Comments:

Signatures

#### Evaluator 2 Comments:

Signatures

#### Project Coordinator Comments

Signatures