TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS

A MINOR PROJECT PROPOSAL ON PROJECT TITLE

BY

SANGYOG PURI (PUR078BCT078)

SANSKAR RIJAL (PUR078BCT079)

SUJAN NAINAWASTI (PUR078BCT090)

VISION PEER CHAUDHARY (PUR078BCT094)

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PURWANCHAL CAMPUS

DHARAN, NEPAL

DECEMBER, 2024

ACKNOWLEDGEMENT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of abbreviations

API   : Application Programming Interface

Colab   : Colaboratory

CHAPTER 1

Introduction

1.1 Background

The Institute of Engineering Purwanchal Campus (IOEPC) is the leading technical education institution in Nepal and offers undergraduate and graduate programs in various fields of engineering, such as civil engineering, mechanical engineering, electrical engineering, electronics and communication engineering, computer engineering, and architecture. It currently carries out administrative and academic tasks both offline and online. While most academic notices are published through social media, including daily routines, this causes important notices to be missed and reliance on third-party software. Moreover, there is a lack of a centralized website or application to access all college-related work. To address this, we propose the development of a mobile application that can integrate all college services and is flexible for future enhancements. Our aim is to provide a convenient and efficient platform for college students while ensuring easy integration of existing offline processes. We anticipate that this application will be highly advantageous to the Institute of Engineering.

1.2 Gap Identification

- The current services of IOEPC are mostly manual and offline, leading to delays and inefficiencies.

- The current system lacks real-time updates and fails to provide timely and relevant information to students; the class schedule is heavily reliant on the class representative.

- Dependency on social media causes some important notices, like late fee notices, to be missed.

- The library management is not integrated and is run separately.

- A modernized system, such as a mobile app, is needed to provide a centralized platform for communication, access to course materials, and other relevant re-

sources.

1.3 Motivation

The Campus Connect app is inspired by the difficulties we faced during our time in college. The inconsistencies between the services provided by the campus and their accessibility have been the driving force. A student isn't aware when fee notices are published if they don't come across a specific Facebook post. Teachers canceling classes at the last minute, while failing to convey it to the class representative, causes commuting students hassle.

1.4 Objectives

- To create an easily scalable and integrable college application.

- To improve online access to the educational system and mitigate current problems by using microservice architecture.

CHAPTER 2

Related Theory

2.1 Related Theory goes here

CHAPTER 3

Literature Review

Literature review goes here

CHAPTER 4

Methodology

4.1 Overview

This section outlines the systematic approach that will be employed to develop the proposed mobile application for the Institute of Engineering Purwanchal Campus (IOEPC). The methodology includes phases of development, tools and technologies used, and strategies for testing and deployment. The application will be developed using a monolithic architecture, with the frontend aspects such as the admin panel implemented in React/React Native, and the student and teacher portals designed in Kotlin.

4.2 Requirements Gathering

- **Interviews and Surveys**: Conduct interviews and surveys with students, faculty, and administrative staff to gather requirements and understand the current challenges.

- **Document Analysis**: Review existing documents, such as academic schedules, notice formats, and library management processes.

4.3 System Design

- **Architecture Design**: Design a monolithic architecture to ensure simplicity and ease of deployment.

- **Database Design**: Develop a relational database schema to store user data, academic schedules, notices, and other relevant information.

- **UI/UX Design**: Create wireframes and prototypes for the mobile application to ensure a user-friendly interface.

4.4 Development

- **Technology Stack**: Use React/React Native for the admin panel and Kotlin for the student and teacher portals. The backend will be developed using Node.js and Express.js.

- **Module Development**: Develop individual modules for different functionalities, such as notice management, academic schedules, library management, and user authentication.

- **Integration**: Integrate all modules to ensure seamless communication between different components of the application.

4.5 Testing

- **Unit Testing**: Perform unit testing for individual modules to ensure they function correctly.

- **Integration Testing**: Conduct integration testing to verify that different modules work together as expected.

- **User Acceptance Testing (UAT)**: Involve a group of students and faculty in testing the application to gather feedback and make necessary improvements.

4.6 Deployment

- **Beta Release**: Deploy a beta version of the application to a limited group of users for initial feedback.

- **Full Deployment**: Roll out the final version of the application to all students and faculty after incorporating feedback from the beta release.

4.7 Maintenance and Updates

- **Monitoring**: Continuously monitor the application for any issues or bugs.

- **Regular Updates**: Provide regular updates to add new features and improve existing functionalities based on user feedback.

4.8 Documentation

- **User Manual**: Create a comprehensive user manual to guide users on how to use the application.

- **Technical Documentation**: Provide detailed technical documentation for future developers to understand the system architecture and codebase.

CHAPTER 5

Expected Results

References

APPENDIX A

APPENDIX B