TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS

A MINOR PROJECT PROPOSAL ON PROJECT TITLE

BY

SANGYOG PURI (PUR078BCT078)

SANSKAR RIJAL(PUR078BCT079)

SUJAN NAINAWASTI (PUR078BCT091)

VISION PEER CHAUDHARY (PUR078BCT0)

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PURWANCHAL CAMPUS

DHARAN, NEPAL

MARCH, 2025

ACKNOWLEDGEMENT

We are very thankful to Department of Electronics and Computer Engineering for providing this opportunity, it has been a great endearing experience.

Firstly, sincierest appericiation to our superviser Asst. Prof. Prasanga Regmi for his invalueable insights.

STUDENT NAME
PUR076/BCT/000

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 Purwachal Campus (IOEPC) is an education institution in Nepal, offering a range of undergraduate and graduate programs in various engineering disciplines. Despite advancements in technology, many administrative and academic processes at campus remain manual and offline, leading to inefficiencies and delays. Technical processes are still being handled through paper-based methods, which are prone to errors and time-consuming. There is no direct connection between teachers and students, leading to delayed communication due to the dependency on class representatives. To address these issues, we propose the development of a mobile application that integrates all IOE services, providing a centralized and efficient platform for users. This application aims to streamline processes such as attendance management and assessment marks, making it easier for teachers to conduct daily activities and ensuring easy integration of existing offline systems and flexibility for future enhancements.

1.2 Gap Identification

Despite the availability of various educational management systems, there is a lack of tailored solutions that cater specifically to the needs of our college. Our applications address this gap by providing customized functionalities that align with the requirements of students, teachers, and administrators.

1.3 Motivation

The motivation behind this project is to simplify and automate the routine tasks of students, teachers, and administrators. By leveraging technology, we aim to reduce manual efforts, minimize errors, and improve the overall user experience. Our team, comprising students of IOEPC, saw an opportunity to contribute to the institution by developing a solution that benefits both students and the institute.

1.4 Objectives

- Develop a user-friendly application for teachers to manage attendance, internal marks, notes, and notices.
- Create a student application that allows students to view their attendance, internal marks, notes, and notices.
- Implement an admin interface to manage user accounts for students and teachers.
- Improve online access to the educational system by using a monolithic architecture.

1.5 Scope

The proposed applications can be adopted by educational institutions to streamline their academic and administrative processes. The teacher application allows teachers to manage attendance, internal marks, notes, and notices efficiently. The student application provides students with easy access to their academic information. The admin interface enables administrators to manage user accounts for students and teachers, ensuring a smooth and secure user experience. The application will serve as an external interface, facilitating activities such as result publication and online attendance, and integrating existing systems to better serve stakeholders.

1.5.1 Academic Study/Research

Academic research is a crucial scope of our project. By sharing our application implementation with other researchers, they can benefit from the study of educational management systems, optimization techniques, and user experience improvements. Our customizable applications with an easy-to-use interface have an important scope in academia.

1.5.2 Comparison of Scope with Existing Systems

There are various educational management systems available, but there are several key differences between these technologies and our applications. Existing systems may

not be tailored to the specific needs of our college, whereas our applications provide customized functionalities that align with the requirements of students, teachers, and administrators. Our applications support real-time data access and management.

CHAPTER 2

RELATED THEORY

2.1 Mobile Application Development

for ref

2.2 Monolithic Architecture

A monolithic architecture is a software design pattern where all components of an application are integrated into a single code base. This approach simplifies deployment and management, as all functionalities are contained within a single codebase. It can be faster in handling request when the number of requests are fixed and the load with less user.

CHAPTER 3

LITERATURE REVIEW

In 2010, JetBrains created Kotlin to enhance the Java Virtual Machine experience for the user. It supports both object-oriented and functional programming paradigms. With the support of non-nullable types, the applications are immune to null pointer exceptions. Furthermore, Kotlin appared as the most loved langague in the Stack Overflow developer survey for the year 2018 and 2019 [1].

The monolithic architecture has every sercvices of the application interconnected and have a very modular architecture and good decoupling between their internal components. In terms of Node.js, all the services are part of the same code base and run in a single process [2].

CHAPTER 4

METHODOLOGY

4.1 Overview

4.2 Other section goes here

CHAPTER 5

EXPECTED RESULTS

REFERENCES

[1] V. Oliveira, L. Teixeira, and F. Ebert, "On the adoption of kotlin on android development: A triangulation study," in *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, London, ON, Canada, 2020, pp. 206–216.

[2] M. Casciaro and L. Mammino, *Node.js Design Patterns*, 2nd ed. Packt Publishing, 2020, referenced pages: 418-421.

APPENDIX A

APPENDIX B