A

Major Project

on

"Pocket College – A Mobile Application for Online Class Conduction"

Submitted to

CHHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY, BHILAI (C.G.), INDIA

In the partial fulfillment for the award of the degree

Bachelor of Engineering

in

COMPUTER SCIENCE & ENGINEERING

Submitted By:

Kanishka Tiwari (BI3717)
Sachin Karun (BG1125)
S Shashank (BI3718)
Hritwan Verma (BG1102)

Guided By:

Mrs. Anjula Shukla
Assistant Professor
Computer Science and Engineering
CEC, Bilaspur (CG)

Department of Computer Science and Engineering
CHOUKSEY ENGINEERING COLLEGE,
LALKHADAN MASTURI ROAD, BILASPUR (C.G)

Session: 2022

DECLARATION BY THE CANDIDATES

We the undersigned solemnly declare that the report of the project work entitled **"POCKET COLLEGE – A Mobile Application for Online Class Conduction"** is based on our own work carried out during the course of our study under the guidance of **Mrs. Anjula Shukla, Assistant Professor, Department of Computer Science & Engineering, Chouksey Engineering College, Bilaspur (CG).**

We further declare that the statements made, and conclusions drawn are an outcome of our project work.

(Signature Of Student)
Kanishka Tiwari
BI3717

(Signature Of Student)
Sachin Karun
BG1125

(Signature Of Student)
S Shashank
BI3718

(Signature Of Student)
Hritwan Verma
BG1102

CERTIFICATE BY GUIDE

This is to certify that the project entitled **"POCKET COLLEGE – A Mobile Application for Online Class Conduction"** is a record of work carried out by **Kanishka Tiwari (BI3717), Sachin Karun (BG1125), S Shashank (BI3718), Hritwan Verma (BG1102),** bearing Under my guidance and supervision for the award of Degree of **Bachelor of Engineering, Chhattisgarh Swami Vivekananda Technical University, Bhilai (C.G.), India.**

To the best of my knowledge and belief the Project:

- **i.** Embodies the work of the candidate him/herself.
- **ii.** Has not been submitted for the award of any degree.
- **iii.** Fulfils the requirement of the Ordinance relating to the B.E degree of the University, and,
- **iv.** Is up to the desired standard in respect of contents and is being referred to the examiners.

(Signature of the HOD with seal)
Dr (Mrs.) Shanu K Rakesh
Head of the Department
Computer Science and Engineering
CEC, Bilaspur (CG)

Recommendation

The Project work as mentioned above is here by being recommended and forwarded for examination and evaluation.

(Signature of the Project In-charge)
Dr (Mrs.) Shanu K Rakesh
Head of the Department
Computer Science & Engineering
CEC, Bilaspur (CG)

(Signature of the Guide)
Mrs. Anjula Shukla
Assistant Professor
Computer Science and Engineering
CEC, Bilaspur (CG)

CERTIFICATE BY THE EXAMINERS

This is to certify that the project work entitled "**POCKET COLLEGE – A Mobile Application for Online Class Conduction**" Submitted by **Kanishka Tiwari (BI3717), Sachin Karun (BG1125), S Shashank (BI3718), Hritwan Verma (BG1102),** have been completed under the guidance of **Mrs. Anjula Shukla, Assistant Professor, Department of Computer Science & Engineering, Chouksey Engineering College, Bilaspur(CG)** has been examined by the undersigned as a part of the examination for the award of Bachelor of Engineering degree in Computer Science & Engineering branch in **Chouksey Engineering College, Bilaspur of Chhattisgarh Swami Vivekanand Technical University, Bhilai(CG).**

"Project Examined and Approved"

Internal Examiner	**External Examiner**
Date:	**Date:**

(Signature of the Principal with seal)
Principal
Chouksey Engineering College, Bilaspur (C.G)

ACKNOWLEDGEMENT

At every outset we express our gratitude to almighty lord for showering his grace and blessing upon us to complete this project.

Although our name appears on the cover of this project, many people had contributed in some form or the other form to this project development. We could not done this project without the assistance or support of each of the following we thank you all.

We wish to place on our record our deep sense of gratitude to our project guide, **Mrs. Anjula Shukla (Asst. Prof.), Dept. of CSE, CEC Bilaspur** and our project in charge, **Dr (Mrs.) Shanu K Rakesh (HOD), Dept. of C.S.E, C.E.C Bilaspur** for their constant motivation and valuable help through the project work. We express our gratitude to **Dr (Mrs.) Shanu K Rakesh, Head of the Department of Computer Science and Engineering, Chouksey Engineering College, Bilaspur** for her valuable suggestions and advice throughout the course. We also extend our thanks to other faculties for their cooperation during our course.

Finally, we would like to thank our friends for their cooperation to complete this project.

(Signature Of Student)
Kanishka Tiwari
BI3717

(Signature Of Student)
Sachin Karun
BG1125

(Signature Of Student)
S Shashank
BI3718

(Signature Of Student)
Hritwan Verma
BG1102

ABSTRACT

The project **"POCKET COLLEGE – A Mobile Application for Online Class Conduction"** aims to provide an integrated platform for the conduction of online classes. Teachers can conduct online lectures seamlessly, take attendance, and post assignments through a single application. Students need not switch applications for different tasks. It is an easy-to-use and user-friendly mobile application with various features necessary for online lectures. The project aims to reduce the effort and hustle caused due to use of multiple applications for different tasks.

This project aims to reduce the hustle caused to students and teachers while switching applications during online classes. It also tackles the memory problem faced by installing multiple applications causing the mobile phone to lose its performance. Teachers can use this application to conduct online lectures through video conferencing, take daily attendance, and post assignments for students. Students can interact with their respective faculties inside the application through the chat feature.

TABLE OF CONTENT

Chapters **Page no**

LIST OF FIGURES

Description of figures **Page no**

LIST OF TABLES

Description of Tables **Page no**

LIST OF ABBREVIATIONS

GUI	Graphical User Interface
COVID-19	Coronavirus Disease 2019
I/O	Input/Output
SDLC	System Development Life Cycle
DFD	Data Flow Diagram
ER Diagram	Entity Relationship Diagram
UWP	Universal Windows Platform
XML	Extensible Markup Language
JS	JavaScript
JSON	JavaScript Object Notation

CHAPTER 1

INTRODUCTION

CHAPTER 01
INTRODUCTION

1.1 Introduction

With the onset of Covid-19 and the pandemic, the majority of educational institutions switched to online mode. Multiple applications, such as Google Meet for conducting online lectures, Google Classroom for posting assignments and tasks, and WhatsApp for conveying important information have been in use. This project aims to eliminate the need for multiple applications to increase accessibility by integrating all the different features into a single application.

The application provides the facilities of video conferencing to conduct seamless online lectures and a chat box feature for interaction among teachers and students. Teachers can share their lecture material through the chat box.

Tasks and assignments features are also available for faculty members to post assignments and an attendance manager to perform roll calls of the students.

Important announcements can be broadcasted by the teachers through the feed section that is accessible to all students. Students can also interact with the teachers through the individual chat section for any queries.

Multiple applications often cause the mobile phone to degrade its performance. This application eliminated the need of installing several applications and saves significant memory for the phone and doesn’t degrade its performance.

1.2 Problem Statement

The problem assigned to us was to design an online education and management application with the aim of improving accessibility for educational institutions. Analyzing the current system of online classes, we decided to reduce the contrast of applications into a single one and came up with this innovation and implemented “POCKET COLLEGE – A Mobile Application for Online Class Conduction”

1.3 Scope

Since the pandemic, in most colleges, a number of different applications are being used for different tasks. Students usually attend their lectures through Google Meet or Zoom. Teachers take their attendance either manually or through Google Classroom. This causes discomfort for both teachers and students to use a different application every time.

Keeping this in mind, we implemented this project "POCKET COLLEGE – A Mobile Application for Online Class Conduction".

This system will increase accessibility and reduce the hustle for the users. It is a user-friendly mobile application with an interactive GUI.

Apart from colleges, this application can also be used in other educational institutes such as schools or coaching centers, since the features remain the same.

1.4 Objective

The main objective of the project is:

- Reduce the complexity of the system
- Increase accessibility
- User satisfaction
- Cost-effective
- Reduce storage requirements

CHAPTER 2

EXISTING SYSTEM

CHAPTER 02
EXISTING SYSTEM

2.1 Google Meet and Classroom

Google Meet is a video-conferencing service developed by Google. It provides the facility of audio and video calling. It has an accompanying chat feature for basic text chatting among participants. It aims to provide high-quality video conferencing between multiple participants [1].

Google Classroom is a blended learning platform developed by Google. It aims to simplify creating and distributing assignments. Its primary purpose is to streamline the process of sharing files between teachers and students [2].

In the majority of colleges and schools, a combination of Google Meet and Google Classroom is popularly being used for the conduction of online classes [3]. The teacher usually sends the Google Meet code via Google Classroom and the lecture is conducted on the Google Meet platform. After the lecture, teachers share the lecture material and assignments/tasks on Google Classroom.

2.2 Microsoft Teams

Microsoft Teams is a proprietary communication platform developed by Microsoft. It provides various features such as video conferencing, document sharing, and file storage. It gained popularity after the COVID-19 pandemic as most meetings moved to a virtual environment [4].

Due to being proprietary software, it has gained popularity in business organizations and offices, and despite providing superior features, it is rarely being used in schools and colleges.

CHAPTER 3

PROBLEM DEFINITION

CHAPTER 03
PROBLEM DEFINITION

Since the pandemic, in most colleges, lectures are being conducted on Google Meet or Zoom, and the distribution of assignments and attendance is being taken through Google Classroom.

The objective of this project is to reduce the effort of switching between applications and increase accessibility for efficient online learning. The primary goal is to keep the interface interactive and user-friendly to ensure user satisfaction. Different features have been integrated into a single application to reduce the installation of multiple applications and save a considerable amount of storage.

CHAPTER 4

SOFTWARE REQUIREMENT ANALYSIS

CHAPTER 04
SOFTWARE REQUIREMENT ANALYSIS

4.1 Functional Requirements

Functional requirements define what a software system should do. It defines a function of a software system or its module. Functionality is measured as a set of inputs to the system under test to the output from the system.

Functional requirements talk about a particular system outcome when a task is performed on them by the user.

For this project, the functional requirements are:

- Multi-step registration and user authentication
- Security
- User-friendly and Interactive GUI

4.2 Non-Functional Requirements

The non-functional requirement says about “what a system should be” rather than “what a system should do” (functional requirement). They are mostly derived from functional requirements based on input from the customer and other stakeholders. Non-functional requirements explain the quality aspects of the system to be constructed viz. performance, portability, usability, etc. Non-functional requirements, unlike functional requirements, are implemented incrementally in any system.

The non-functional requirements considered are:

- Storage efficiency
- Reduction in complexity
- Improved performance

4.3 System Requirements

System requirements are the required specifications a device must have in order to use certain hardware or software. For example, a computer may require a specific I/O port to work with a peripheral device. A smartphone may need a specific operating system to run a particular application.

The minimum software requirements for running this application are:

- Hardware: Camera (Front and Back) enabled Smartphone
- Operating System: Android 6.0
- Memory: At least 50 MB of free space
- IDE: Visual Studio
- Development Framework: React Native
- Development Language: JavaScript
- Database: Firebase Realtime Database

CHAPTER 5

SYSTEM DESIGN

CHAPTER 05
SYSTEM DESIGN

5.1 Software Development Model

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.

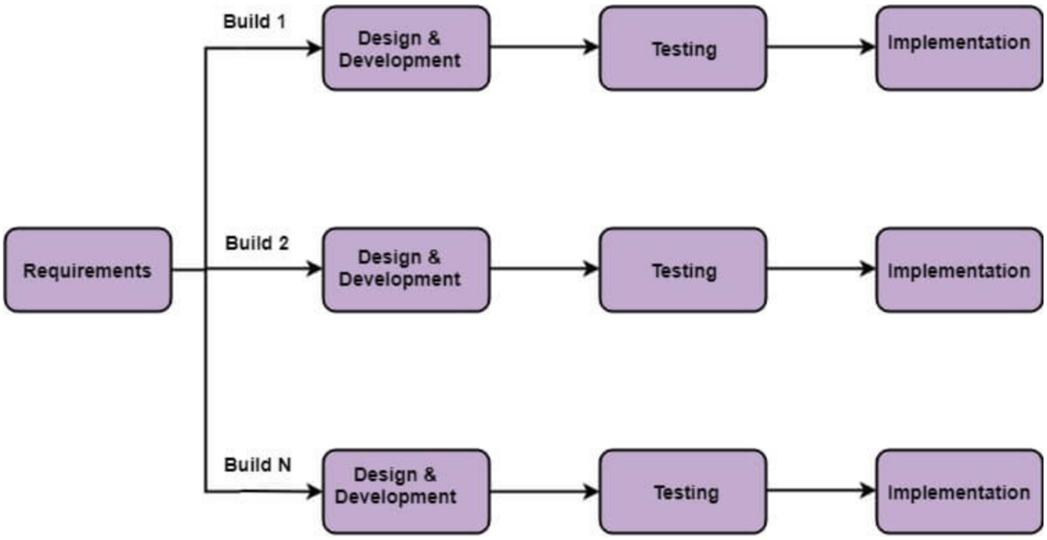


Fig. 5.1 Incremental Model

The various phases of incremental model are as follows:

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2. Design & Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

4. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product

5.2 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

5.2.1 LEVEL-0 DFD

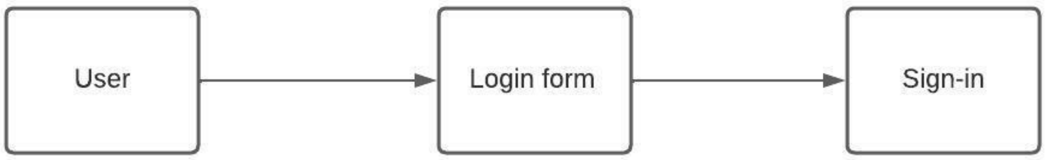


Fig. 5.2 Level 0 DFD

5.2.2 LEVEL-I DFD

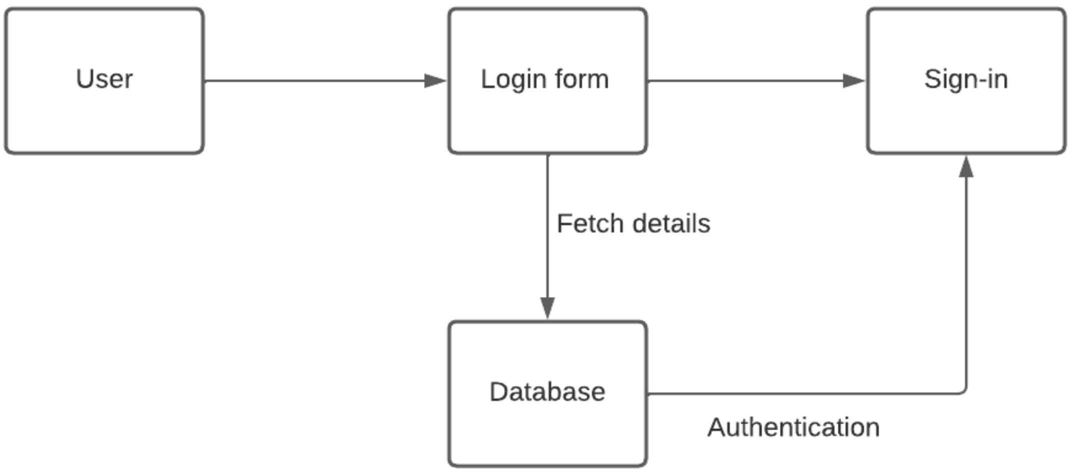


Fig. 5.3 Level 1 DFD

5.2.3 LEVEL-II DFD

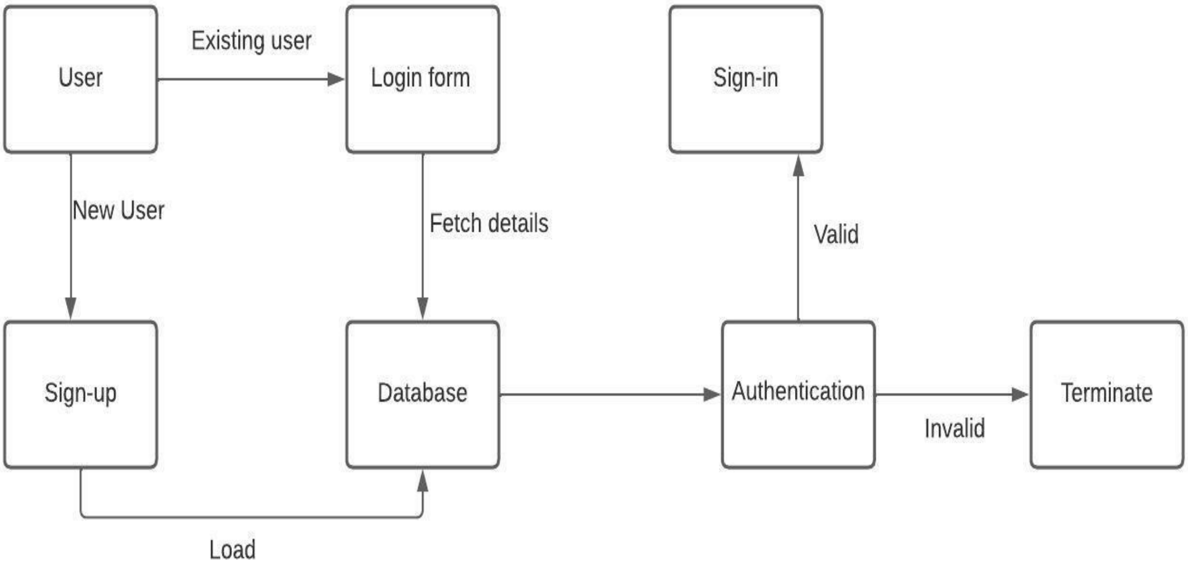


Fig. 5.4 Level 2 DFD

5.3 Component Diagram

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

5.3.1 Attendance Component Diagram

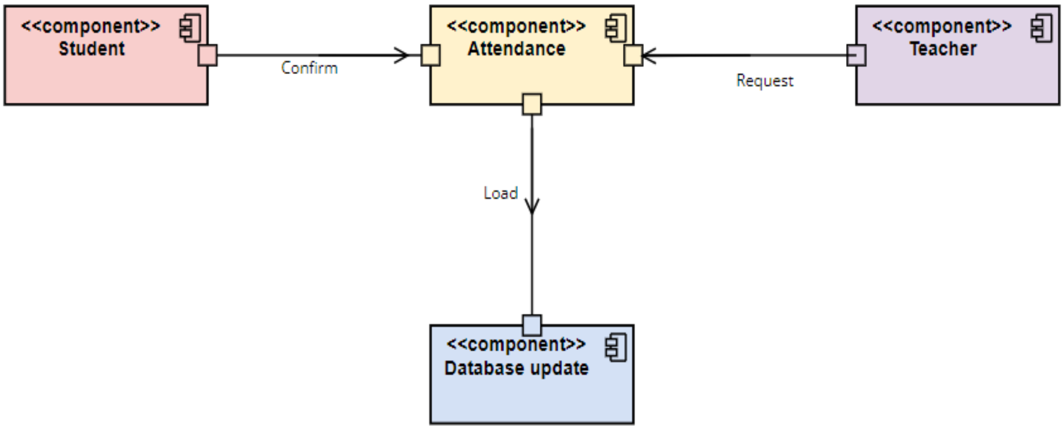


Fig. 5.5 Attendance Component Diagram

5.3.2 Assignment Component Diagram

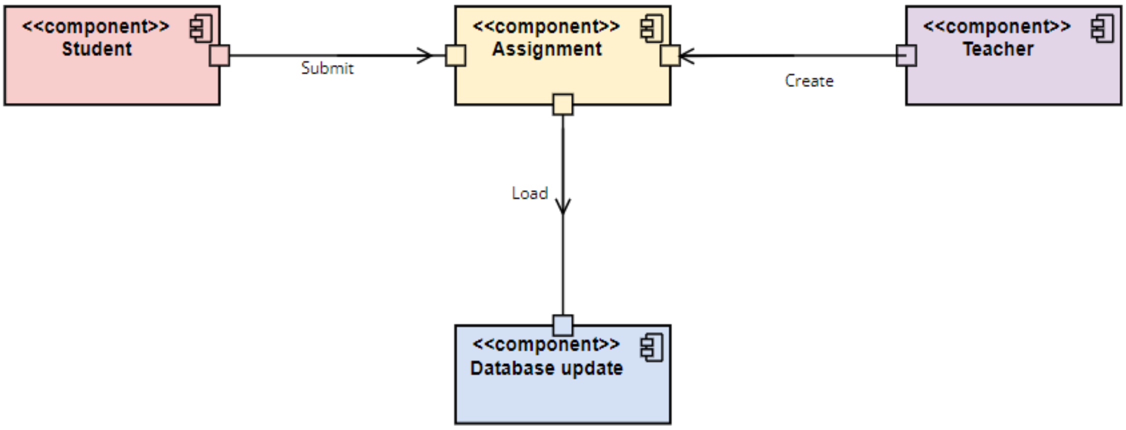


Fig. 5.6 Assignment Component Diagram

5.4 ER Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

5.4.1 Attendance ER Diagram

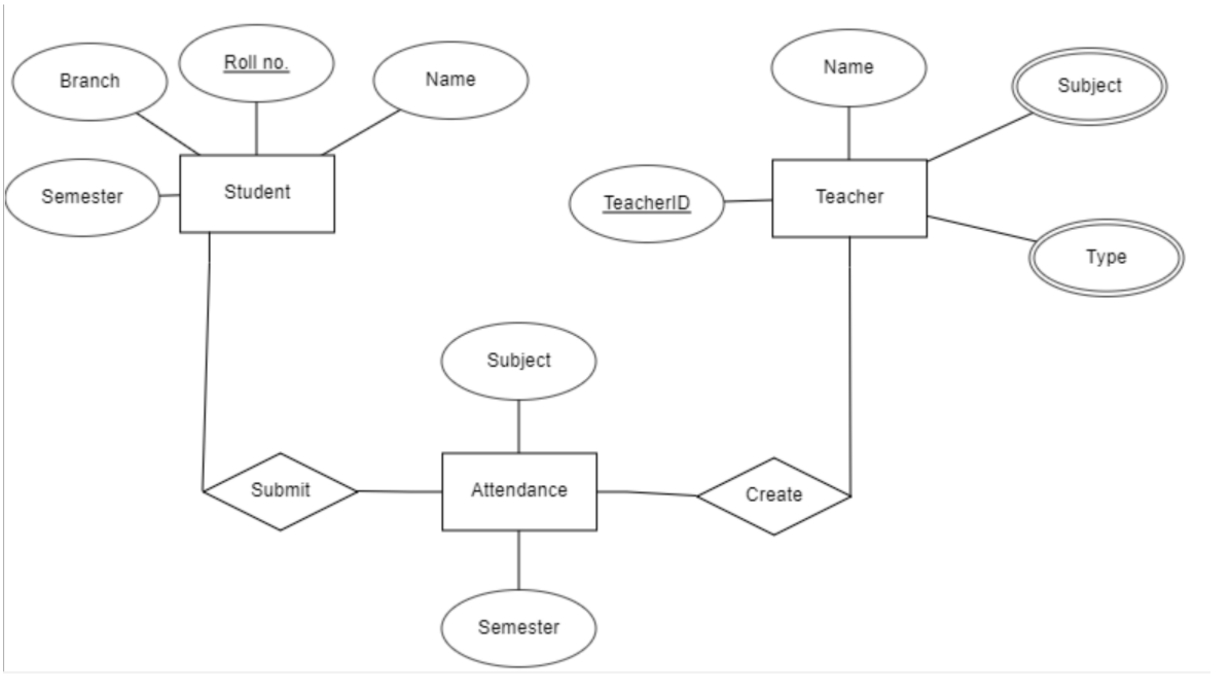


Fig. 5.7 Attendance ER Diagram

5.4.2 Assignment ER Diagram

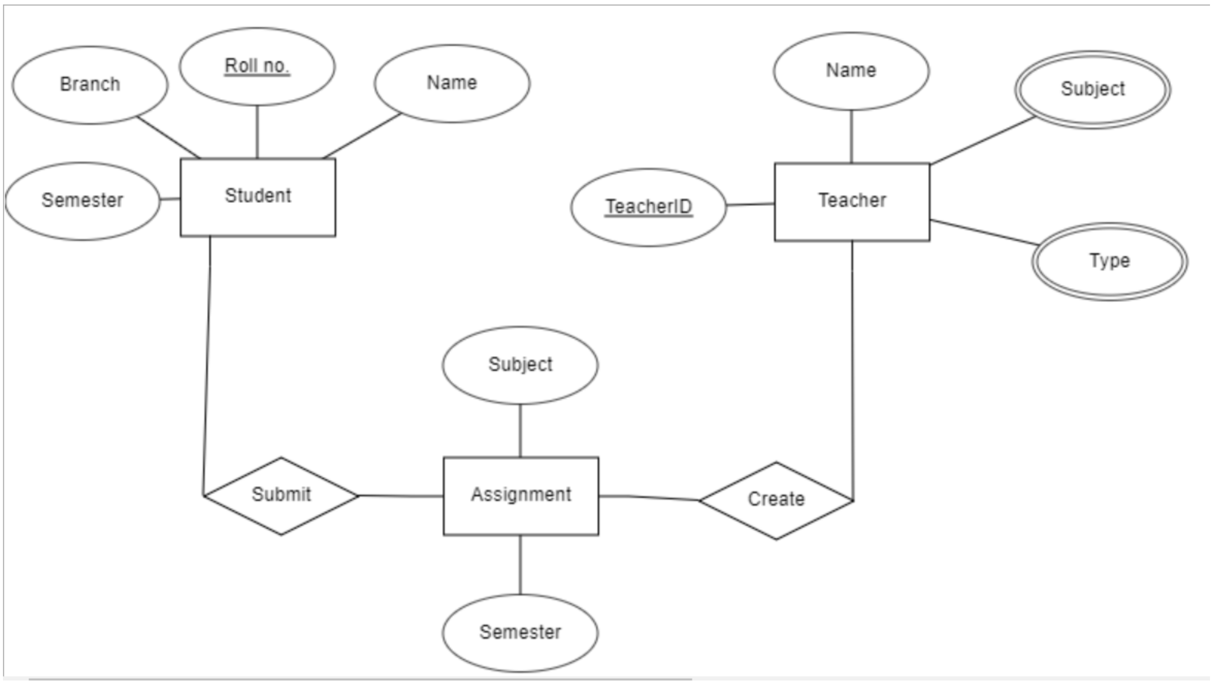


Fig. 5.8 Assignment ER Diagram

5.5 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

5.5.1 Attendance Class Diagram

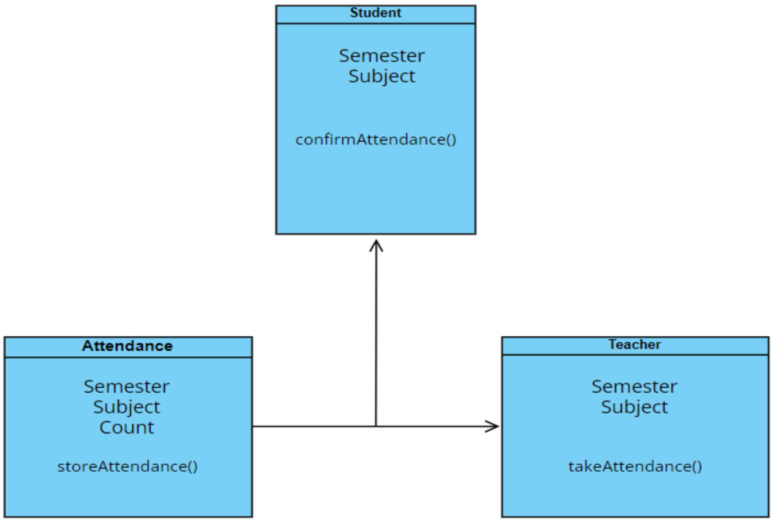


Fig. 5.9 Attendance Class Diagram

5.5.2 Assignment Class Diagram

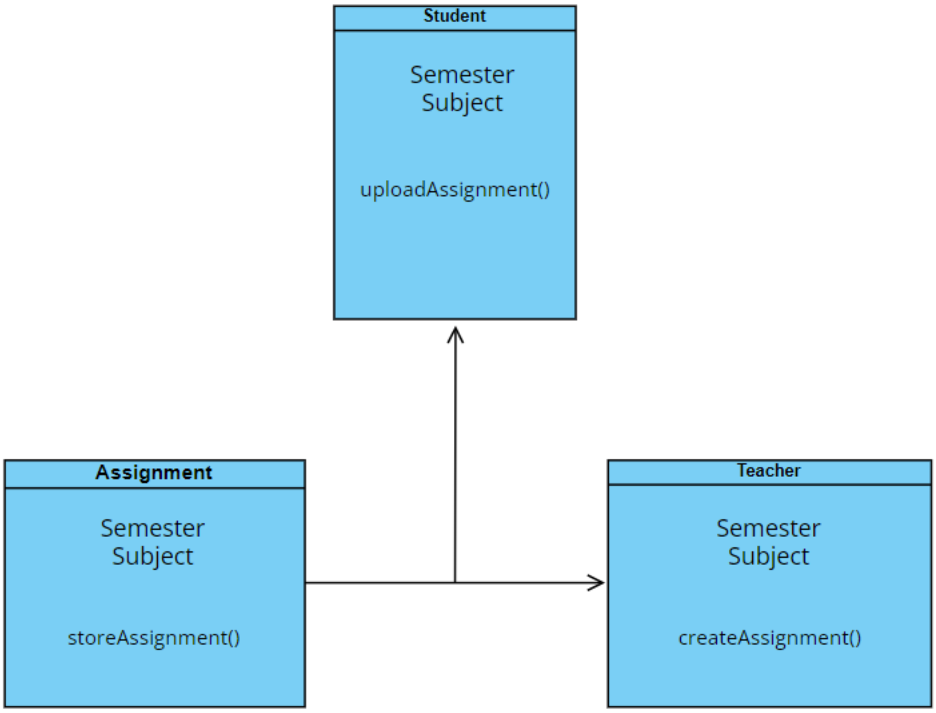


Fig. 5.10 Assignment Class Diagram

CHAPTER 6

PROPOSED SYSTEM

CHAPTER 06
PROPOSED SYSTEM

As online education has become much more prevalent since COVID-19, the problem that arose was all features required were not met by a single application. To reduce this effort of switching applications and making the process of conducting online learning for both students and teachers, multiple features have been integrated into a single application in this project "POCKET COLLEGE".

The system can be divided into 6 different modules

Module-1: User Signup and Login

A new user is provided with multi-step interactive user registration. There can be two kinds of users: students and teachers. Depending on the type, the user shall provide relevant details.

Once the user is registered, the data is stored in the database for future reference. Existing users shall provide email and password and this data is authenticated from the database for successful login.

Module-2: Feed Section

This section is meant to be used by teachers to convey information about events taking place and other important announcements/notices and posters related to the college to everyone at once.

Module-3: Attendance Manager

This module provides a system for teachers to take daily attendance. The teacher has the option of creating a request of attendance for their respective class and subject. Once a request is created, students will be able to view the request on their side.

Students have the option of confirming the request for attendance. This confirmation is reflected back to the teacher's portal and the relevant information is stored in the database.

Module-4: Assignment, Task, and Notes Creation

Teachers are also provided with two important features, assignment, and task creation. Teachers can create assignments for their respective subjects. Once an assignment is posted by a teacher, the students of the respective class can view the assignment. Students have the option of uploading a PDF file and making submissions.

The teacher has access to view all submissions made by students and view the files uploaded for grading.

The task feature is to convey immediate instructions to the students. Teachers have the privilege to create a task and students have view access to this feature.

For both teachers and students, there is an additional notes section where a particular student or teacher can make their personal notes, reminders, or just use it as a to-do list. This is a private feature with the access limited only to that particular user.

Module-5: Chat Module

The chat module is a general interest forum that enables interaction between students and teachers without moving to a different application. The main objective of this feature is to limit student-faculty communication at a professional level for academic purposes.

Module-6: Video Conferencing

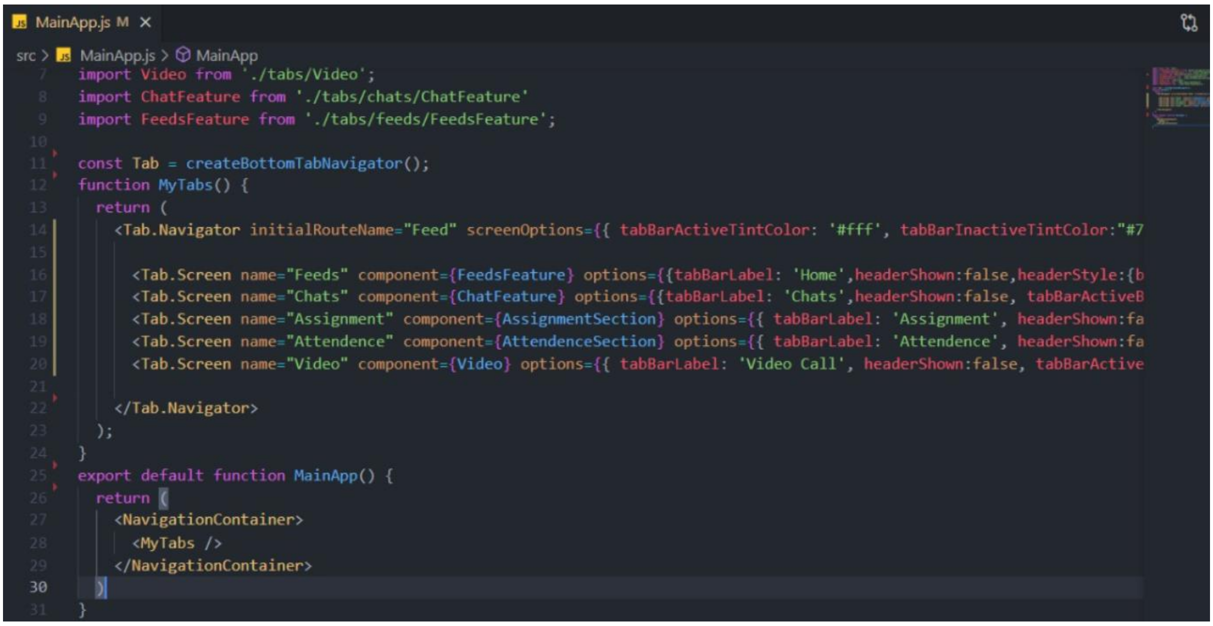
This module enables conduction of online lectures by creating a virtual classroom. Once a room is created, students can enter the room code and join the classroom. The classroom has both audio and video features enabling a two-way interaction between faculty and students.

CHAPTER 7
CODING

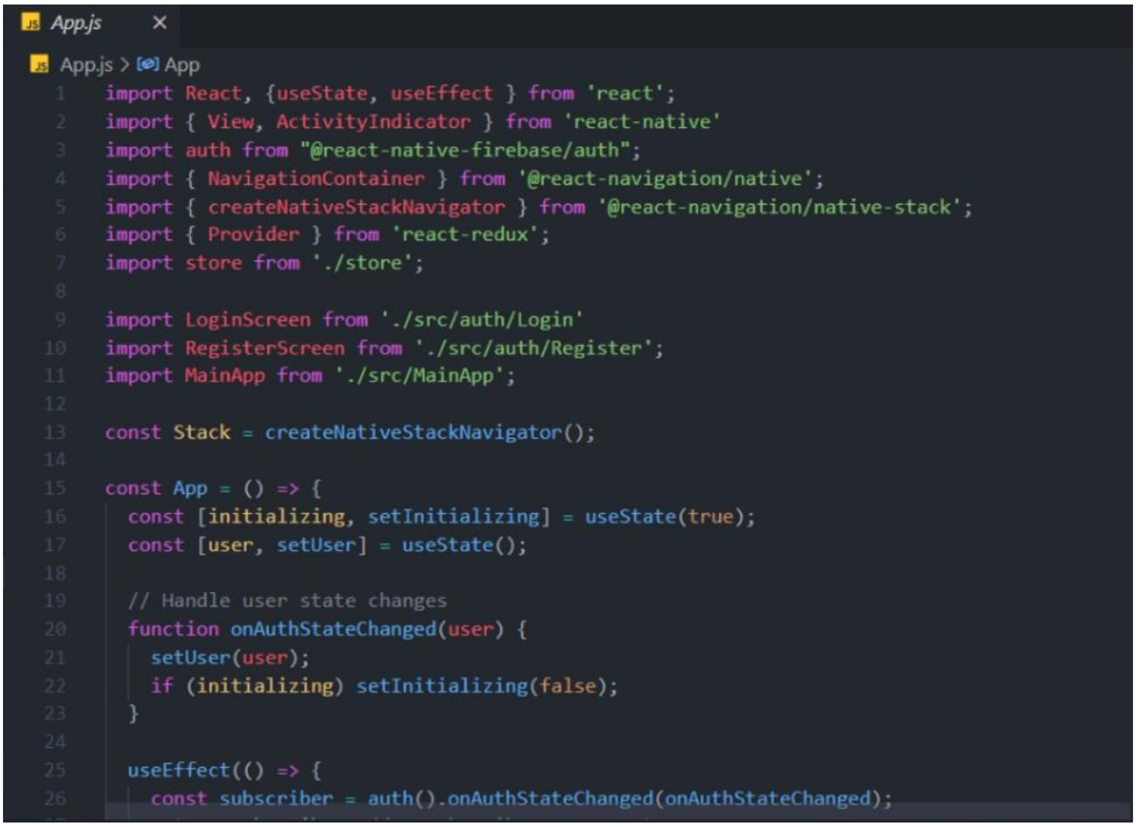
CHAPTER 07

CODING

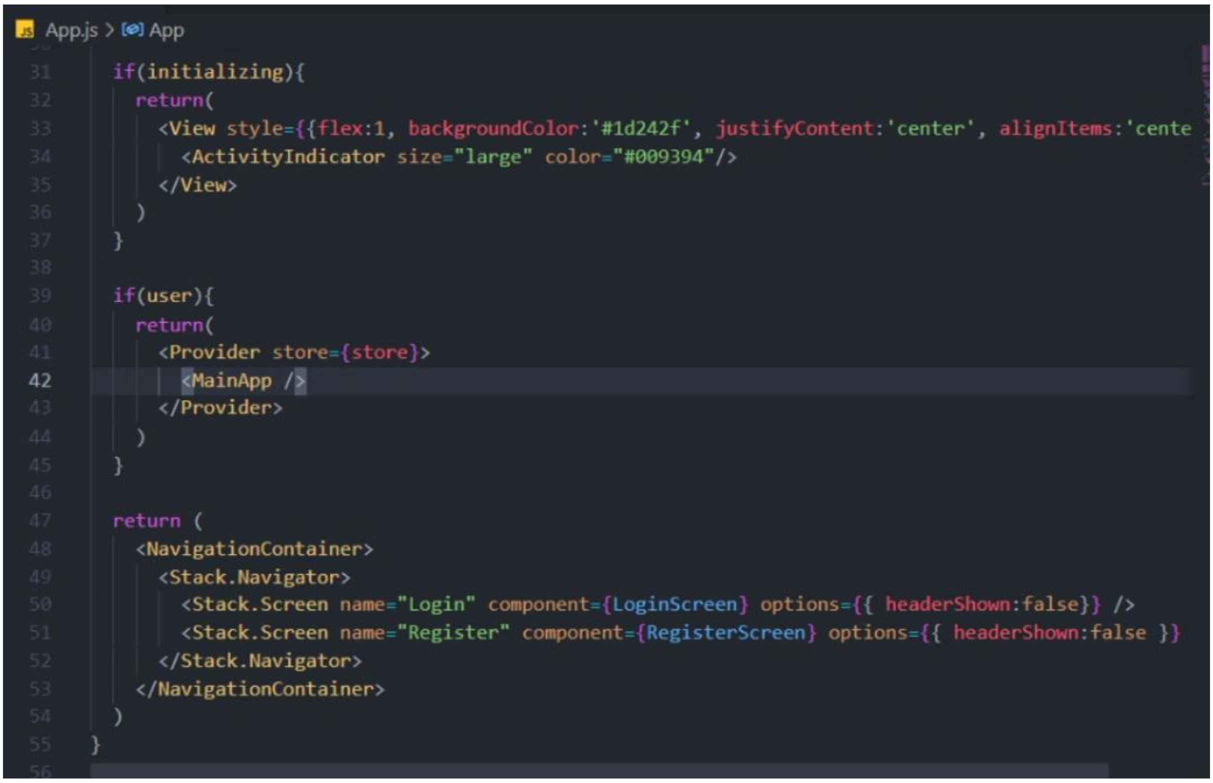
7.1 Main Code



```
import Video from './tabs/Video';
import ChatFeature from './tabs/chats/ChatFeature'
import FeedsFeature from './tabs/feeds/FeedsFeature';

const Tab = createBottomTabNavigator();
function MyTabs() {
  return (
    <Tab.Navigator initialRouteName="Feed" screenOptions={{ tabBarActiveTintColor: '#fff', tabBarInactiveTintColor:"#7

      <Tab.Screen name="Feeds" component={FeedsFeature} options={{tabBarLabel: 'Home',headerShown:false,headerStyle:{b
      <Tab.Screen name="Chats" component={ChatFeature} options={{tabBarLabel: 'Chats',headerShown:false, tabBarActiveB
      <Tab.Screen name="Assignment" component={AssignmentSection} options={{ tabBarLabel: 'Assignment', headerShown:fa
      <Tab.Screen name="Attendence" component={AttendenceSection} options={{ tabBarLabel: 'Attendence', headerShown:fa
      <Tab.Screen name="Video" component={Video} options={{ tabBarLabel: 'Video Call', headerShown:false, tabBarActive

    </Tab.Navigator>
  );
}
export default function MainApp() {
  return (
    <NavigationContainer>
      <MyTabs />
    </NavigationContainer>
  )
}
```



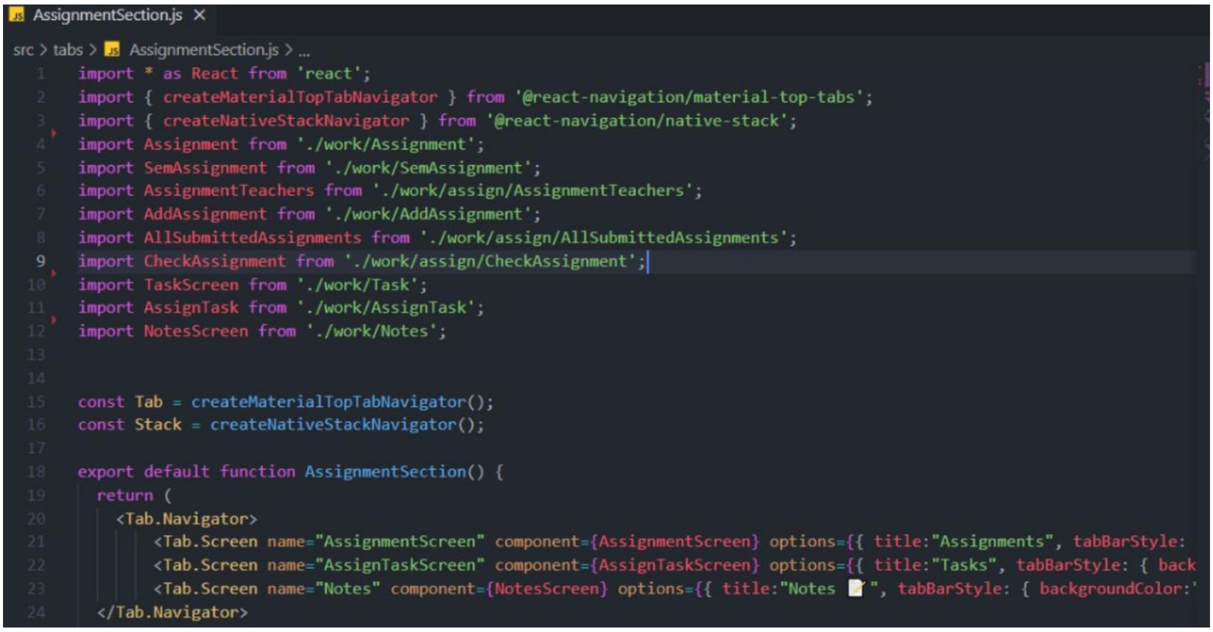
```
import React, {useState, useEffect } from 'react';
import { View, ActivityIndicator } from 'react-native'
import auth from "@react-native-firebase/auth";
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import { Provider } from 'react-redux';
import store from './store';

import LoginScreen from './src/auth/Login'
import RegisterScreen from './src/auth/Register';
import MainApp from './src/MainApp';

const Stack = createNativeStackNavigator();

const App = () => {
  const [initializing, setInitializing] = useState(true);
  const [user, setUser] = useState();

  // Handle user state changes
  function onAuthStateChanged(user) {
    setUser(user);
    if (initializing) setInitializing(false);
  }

  useEffect(() => {
    const subscriber = auth().onAuthStateChanged(onAuthStateChanged);
```



App.js > App

```
  if(initializing){
    return(
      <View style={{flex:1, backgroundColor:'#1d242f', justifyContent:'center', alignItems:'cente
        <ActivityIndicator size="large" color="#009394"/>
      </View>
    )
  }

  if(user){
    return(
      <Provider store={store}>
        <MainApp />
      </Provider>
    )
  }

  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Login" component={LoginScreen} options={{ headerShown:false}} />
        <Stack.Screen name="Register" component={RegisterScreen} options={{ headerShown:false }}
      </Stack.Navigator>
    </NavigationContainer>
  )
}
```

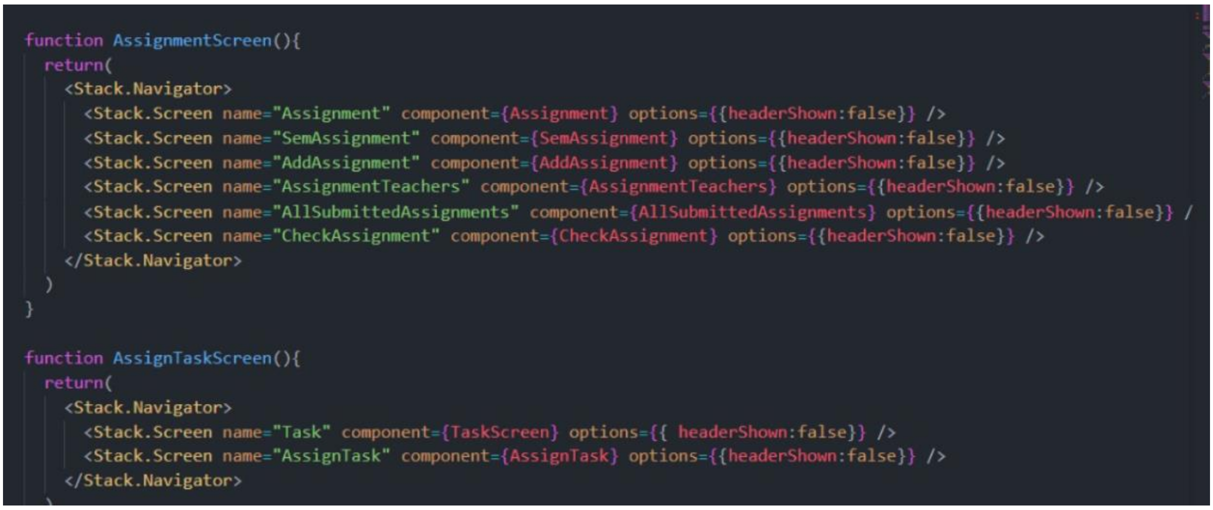
7.2 Assignment Section Code



AssignmentSection.js ×

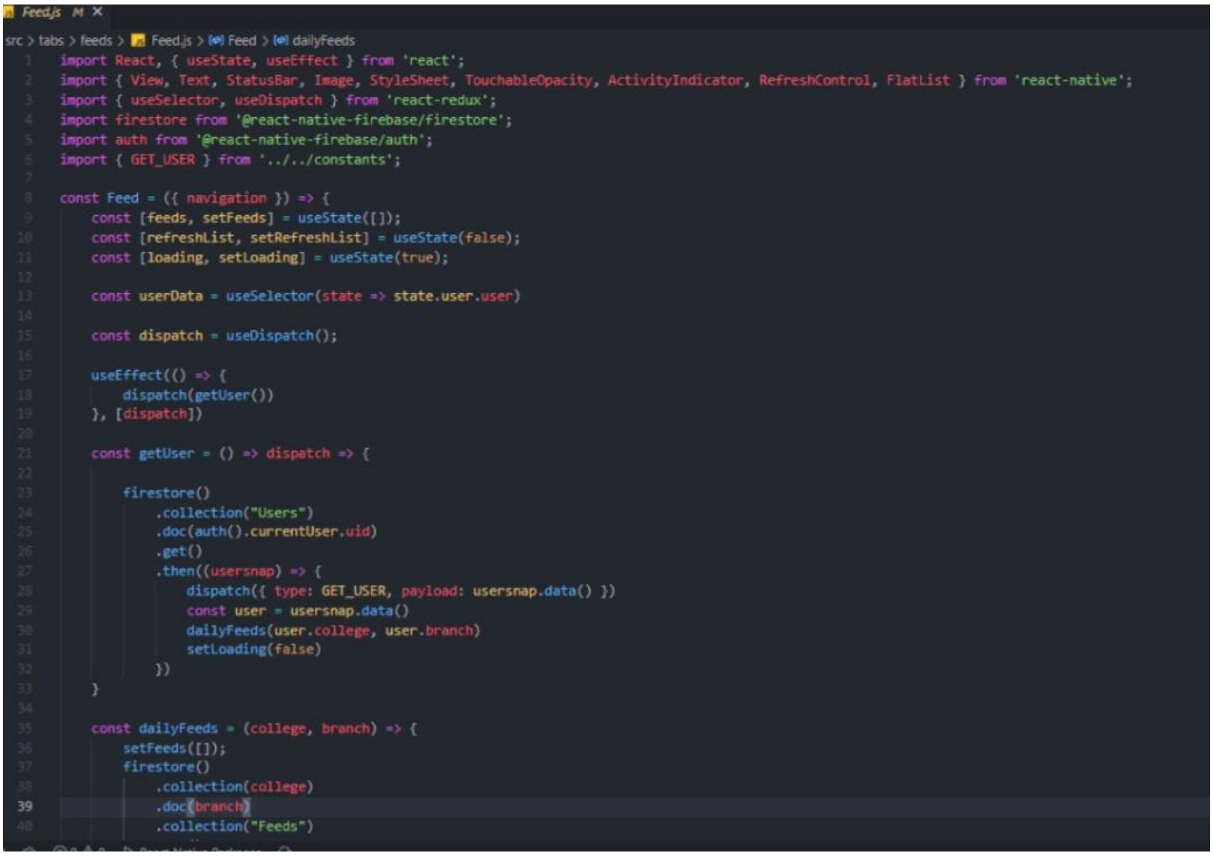
src > tabs > AssignmentSection.js > ...

```
import * as React from 'react';
import { createMaterialTopTabNavigator } from '@react-navigation/material-top-tabs';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import Assignment from './work/Assignment';
import SemAssignment from './work/SemAssignment';
import AssignmentTeachers from './work/assign/AssignmentTeachers';
import AddAssignment from './work/AddAssignment';
import AllSubmittedAssignments from './work/assign/AllSubmittedAssignments';
import CheckAssignment from './work/assign/CheckAssignment';
import TaskScreen from './work/Task';
import AssignTask from './work/AssignTask';
import NotesScreen from './work/Notes';


const Tab = createMaterialTopTabNavigator();
const Stack = createNativeStackNavigator();

export default function AssignmentSection() {
  return (
    <Tab.Navigator>
        <Tab.Screen name="AssignmentScreen" component={AssignmentScreen} options={{ title:"Assignments", tabBarStyle:
        <Tab.Screen name="AssignTaskScreen" component={AssignTaskScreen} options={{ title:"Tasks", tabBarStyle: { back
        <Tab.Screen name="Notes" component={NotesScreen} options={{ title:"Notes 📝", tabBarStyle: { backgroundColor:'
    </Tab.Navigator>
```

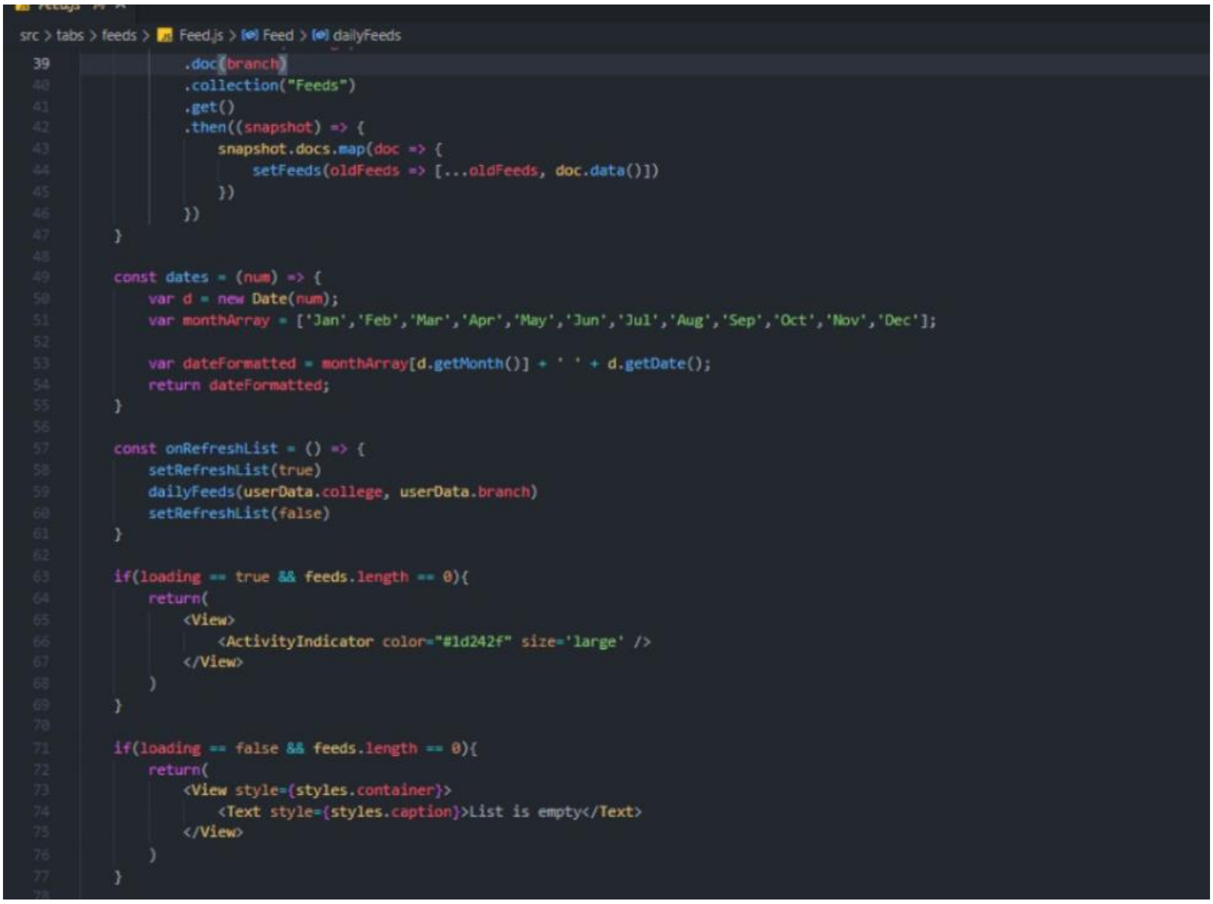


```
function AssignmentScreen(){
  return(
    <Stack.Navigator>
      <Stack.Screen name="Assignment" component={Assignment} options={{headerShown:false}} />
      <Stack.Screen name="SemAssignment" component={SemAssignment} options={{headerShown:false}} />
      <Stack.Screen name="AddAssignment" component={AddAssignment} options={{headerShown:false}} />
      <Stack.Screen name="AssignmentTeachers" component={AssignmentTeachers} options={{headerShown:false}} />
      <Stack.Screen name="AllSubmittedAssignments" component={AllSubmittedAssignments} options={{headerShown:false}} /
      <Stack.Screen name="CheckAssignment" component={CheckAssignment} options={{headerShown:false}} />
    </Stack.Navigator>
  )
}

function AssignTaskScreen(){
  return(
    <Stack.Navigator>
      <Stack.Screen name="Task" component={TaskScreen} options={{ headerShown:false}} />
      <Stack.Screen name="AssignTask" component={AssignTask} options={{headerShown:false}} />
    </Stack.Navigator>
```

7.3 Feed Section Code

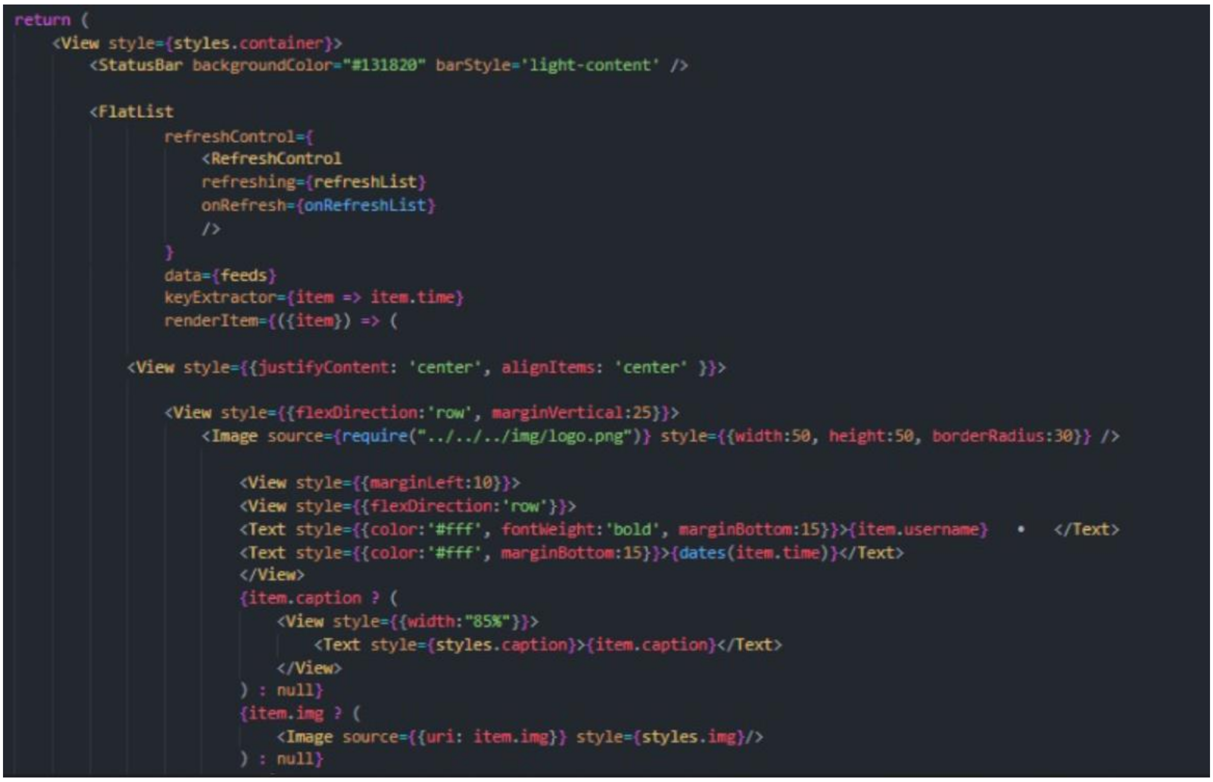


```
import React, { useState, useEffect } from 'react';
import { View, Text, StatusBar, Image, StyleSheet, TouchableOpacity, ActivityIndicator, RefreshControl, FlatList } from 'react-native';
import { useSelector, useDispatch } from 'react-redux';
import firestore from '@react-native-firebase/firestore';
import auth from '@react-native-firebase/auth';
import { GET_USER } from '../../constants';

const Feed = ({ navigation }) => {
    const [feeds, setFeeds] = useState([]);
    const [refreshList, setRefreshList] = useState(false);
    const [loading, setLoading] = useState(true);

    const userData = useSelector(state => state.user.user)

    const dispatch = useDispatch();

    useEffect(() => {
        dispatch(getUser())
    }, [dispatch])

    const getUser = () => dispatch => {

        firestore()
            .collection("Users")
            .doc(auth().currentUser.uid)
            .get()
            .then((usersnap) => {
                dispatch({ type: GET_USER, payload: usersnap.data() })
                const user = usersnap.data()
                dailyFeeds(user.college, user.branch)
                setLoading(false)
            })
    }

    const dailyFeeds = (college, branch) => {
        setFeeds([]);
        firestore()
            .collection(college)
            .doc(branch)
            .collection("Feeds")
```



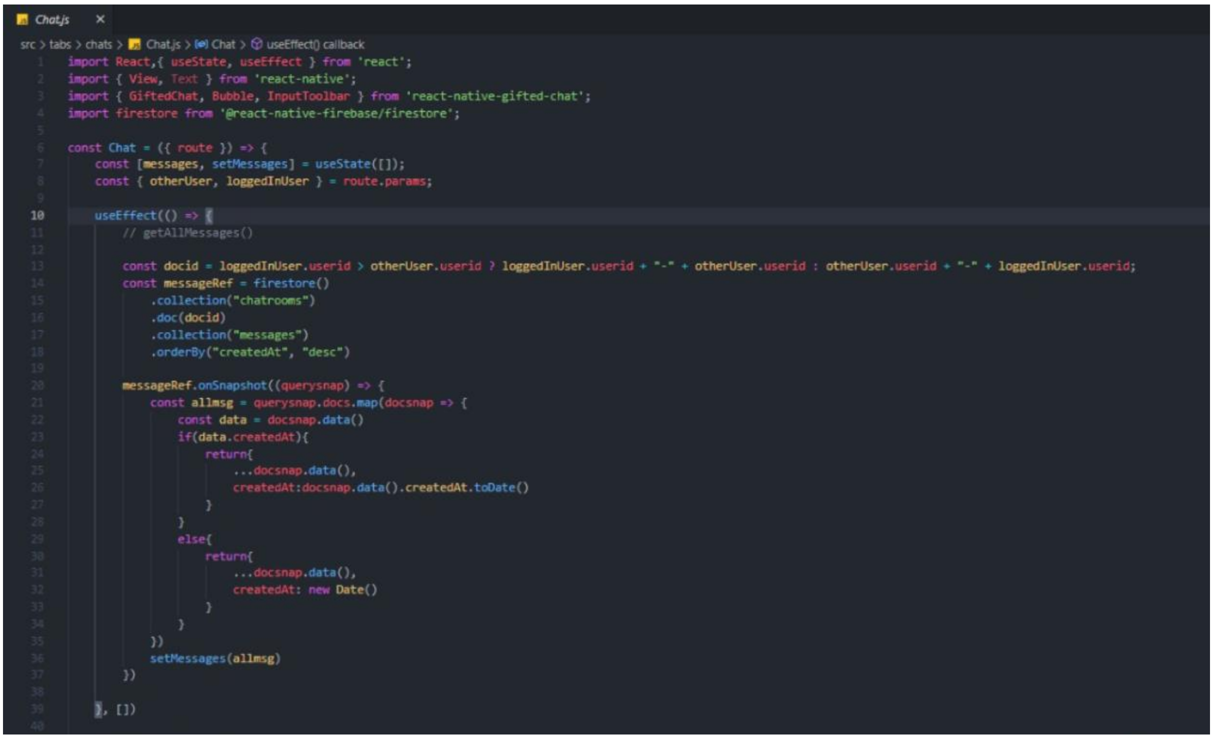
src > tabs > feeds > Feed.js > Feed > dailyFeeds

```
                .doc(branch)
                .collection("Feeds")
                .get()
                .then((snapshot) => {
                    snapshot.docs.map(doc => {
                        setFeeds(oldFeeds => [...oldFeeds, doc.data()])
                    })
                })
    }

    const dates = (num) => {
        var d = new Date(num);
        var monthArray = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'];

        var dateFormatted = monthArray[d.getMonth()] + ' ' + d.getDate();
        return dateFormatted;
    }

    const onRefreshList = () => {
        setRefreshList(true)
        dailyFeeds(userData.college, userData.branch)
        setRefreshList(false)
    }

    if(loading == true && feeds.length == 0){
        return(
            <View>
                <ActivityIndicator color="#1d242f" size='large' />
            </View>
        )
    }

    if(loading == false && feeds.length == 0){
        return(
            <View style={styles.container}>
                <Text style={styles.caption}>List is empty</Text>
            </View>
        )
    }
```



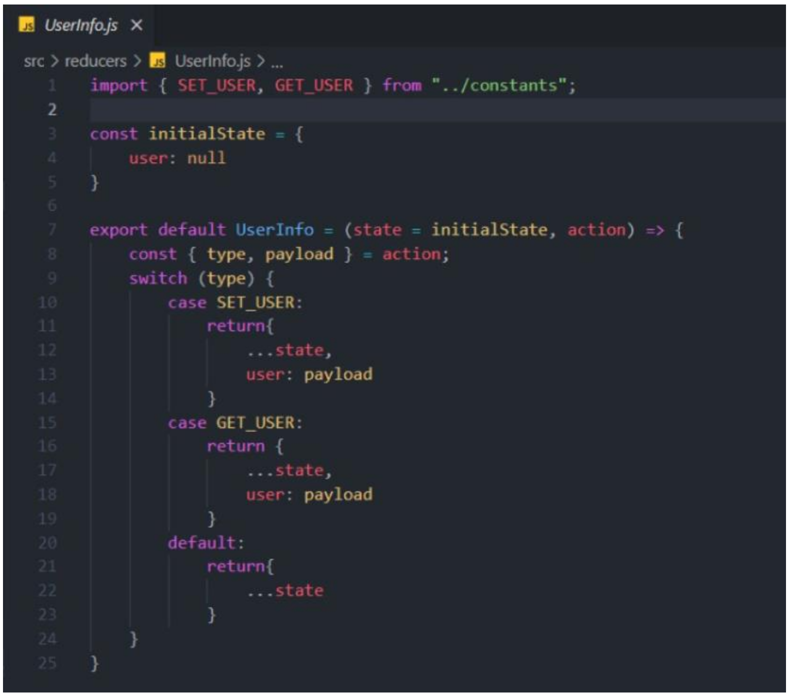
```
return (
    <View style={styles.container}>
        <StatusBar backgroundColor="#131820" barStyle='light-content' />

        <FlatList
                refreshControl={
                    <RefreshControl
                    refreshing={refreshList}
                    onRefresh={onRefreshList}
                    />
                }
                data={feeds}
                keyExtractor={item => item.time}
                renderItem={({item}) => (

            <View style={{justifyContent: 'center', alignItems: 'center' }}>

                <View style={{flexDirection:'row', marginVertical:25}}>
                    <Image source={require("../../../img/logo.png")} style={{width:50, height:50, borderRadius:30}} />

                        <View style={{marginLeft:10}}>
                        <View style={{flexDirection:'row'}}>
                        <Text style={{color:'#fff', fontWeight:'bold', marginBottom:15}}>{item.username}   •   </Text>
                        <Text style={{color:'#fff', marginBottom:15}}>{dates(item.time)}</Text>
                        </View>
                        {item.caption ? (
                            <View style={{width:"85%"}}>
                                <Text style={styles.caption}>{item.caption}</Text>
                            </View>
                        ) : null}
                        {item.img ? (
                            <Image source={{uri: item.img}} style={styles.img}/>
                        ) : null}
```

7.4 Chat Section Code

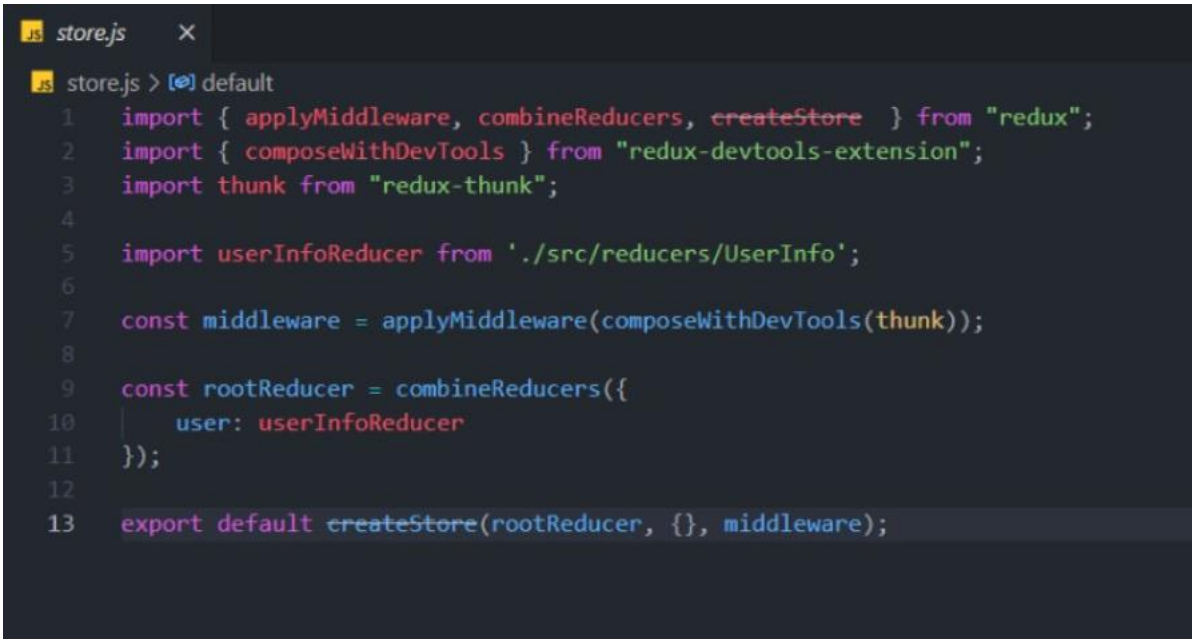


```
import React,{ useState, useEffect } from 'react';
import { View, Text } from 'react-native';
import { GiftedChat, Bubble, InputToolbar } from 'react-native-gifted-chat';
import firestore from '@react-native-firebase/firestore';

const Chat = ({ route }) => {
    const [messages, setMessages] = useState([]);
    const { otherUser, loggedInUser } = route.params;

    useEffect(() => {
        // getAllMessages()

        const docid = loggedInUser.userid > otherUser.userid ? loggedInUser.userid + "-" + otherUser.userid : otherUser.userid + "-" + loggedInUser.userid;
        const messageRef = firestore()
            .collection("chatrooms")
            .doc(docid)
            .collection("messages")
            .orderBy("createdAt", "desc")

        messageRef.onSnapshot((querysnap) => {
            const allmsg = querysnap.docs.map(docsnap => {
                const data = docsnap.data()
                if(data.createdAt){
                    return{
                        ...docsnap.data(),
                        createdAt:docsnap.data().createdAt.toDate()
                    }
                }
                else{
                    return{
                        ...docsnap.data(),
                        createdAt: new Date()
                    }
                }
            })
            setMessages(allmsg)
        })

    }, [])

```

7.5 User Information Code



```
import { SET_USER, GET_USER } from "../constants";

const initialState = {
    user: null
}

export default UserInfo = (state = initialState, action) => {
    const { type, payload } = action;
    switch (type) {
        case SET_USER:
            return{
                ...state,
                user: payload
            }
        case GET_USER:
            return {
                ...state,
                user: payload
            }
        default:
            return{
                ...state
            }
    }
}
```



```javascript
import { applyMiddleware, combineReducers, createStore  } from "redux";
import { composeWithDevTools } from "redux-devtools-extension";
import thunk from "redux-thunk";

import userInfoReducer from './src/reducers/UserInfo';

const middleware = applyMiddleware(composeWithDevTools(thunk));

const rootReducer = combineReducers({
    user: userInfoReducer
});

export default createStore(rootReducer, {}, middleware);
```

CHAPTER 8

TESTING

CHAPTER 08
TESTING

8.1 Testing

Testing is the process of exercising software with the intent of finding errors and ultimately correcting them. The following testing techniques have been used to make this project free of errors.

Content Review

The whole content of the project has been reviewed thoroughly to uncover typographical errors, grammatical errors, and ambiguous sentences.

Navigation Errors

Different users were allowed to navigate through the project to uncover the navigation errors. The views of the user regarding navigation flexibility and user-friendliness were taken into account and implemented in the project.

Methods:

8.1.1 Unit Testing

- Focuses on individual software units, and groups of related units.
- Unit – the smallest testable piece of software.
- A unit can be compiled /assembled/linked/loaded, and put under a test harness.
- Unit testing is done to show that the unit does not satisfy the application and/or its implemented software does not match the intended designed structure.

8.1.2 Integration Testing

- Focuses on combining units to evaluate the interaction among them
- Integration is the process of aggregating components to create a larger
- components.
- Integration testing is done to show that even though components were
- individually satisfactory, the combination is incorrect and inconsistent.

8.1.3 System testing

- Focuses on a complete integrated system to evaluate compliance with specified requirements (test characteristics that are only present when the entire system is run)
- A system is a big component.
- System testing is aimed at revealing bugs that cannot be attributed to a component as such, to inconsistencies between components or planned interactions between components.
- Concern: issues, behaviors that can only be exposed by testing the entire integrated system (e.g., performance, security, recovery) each form encapsulates (labels, texts, grid, etc.). Hence in the case of projects in V.B. form are the basic units. Each form is tested thoroughly in terms of calculation, display, etc.

8.1.4 White-Box testing

- White-Box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) tests the internal structures or workings of a program, as opposed to the functionality exposed to the end-user.
- In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.
- The tester chooses inputs to exercise paths through the code and determines the appropriate outputs.
- This is analogous to testing nodes in a circuit, e.g., in-circuit testing (ICT).
- While white-box testing can be applied at the unit, integration, and system levels of the software testing process, it is usually done at the unit level.

- It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test.
- Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

8.1.5 Black-box testing

- Black-box testing treats the software as a "black-box", examining functionality without any knowledge of the internal implementation.
- The tester is only aware of what the software is supposed to do, not how it does it.
- Black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case or leaves some parts of the program untested.
- This method of test can be applied to all levels of software testing: unit, integration, system, and acceptance.

8.2 Test Cases

TEST CASE DESCRIPTION	EXPECTED OUTPUT	ACTUAL OUTPUT
Login with incorrect credentials	Login unsuccessful	Login unsuccessful (Invalid user)
Multiple attendance requests	Attendance counted only once	Attendance counted only once
Assignment submission	Only PDF files accepted	PDF and JPG Files Accepted
Number of users in video calling	Upper limit = 100	Upper limit = 100

Table. 8.1 Test Cases

CHAPTER 9

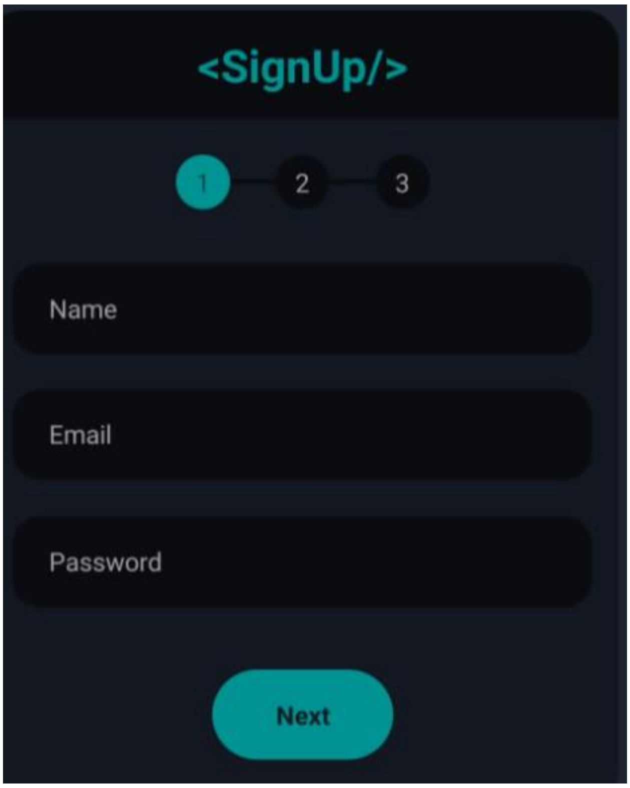
OUTPUT SCREENS/ RESULTS SNAPSHOTS

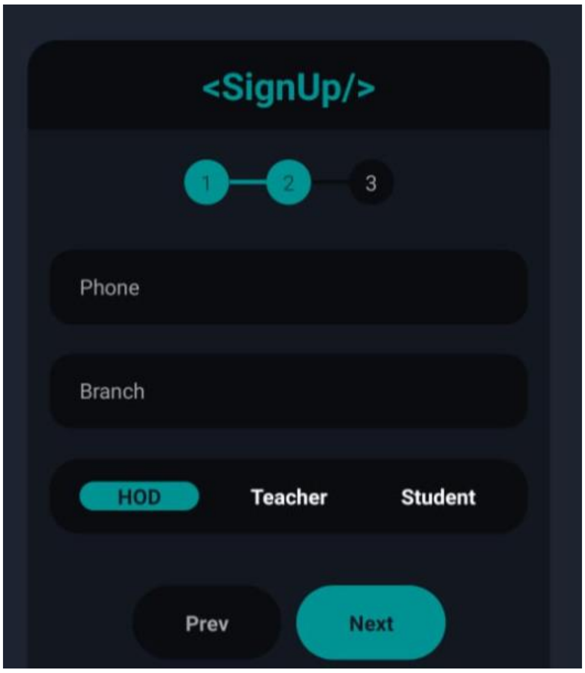
CHAPTER 09

OUTPUT SCREENS/ RESULTS SNAPSHOTS

9.1 Sign-up and Login

9.1.1 Sign Up





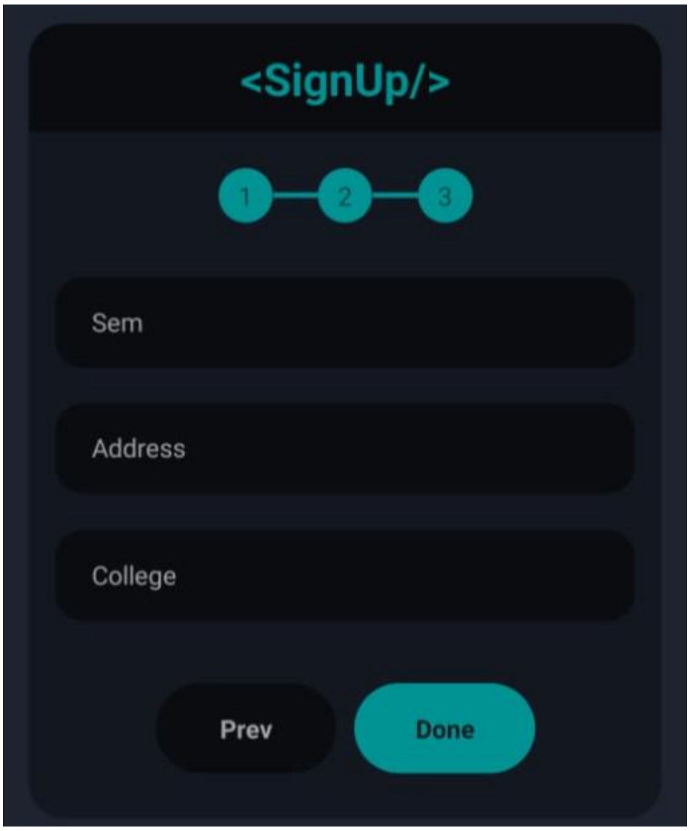


Fig. 9.1 Sign-up Page

9.1.2 User Sign-in

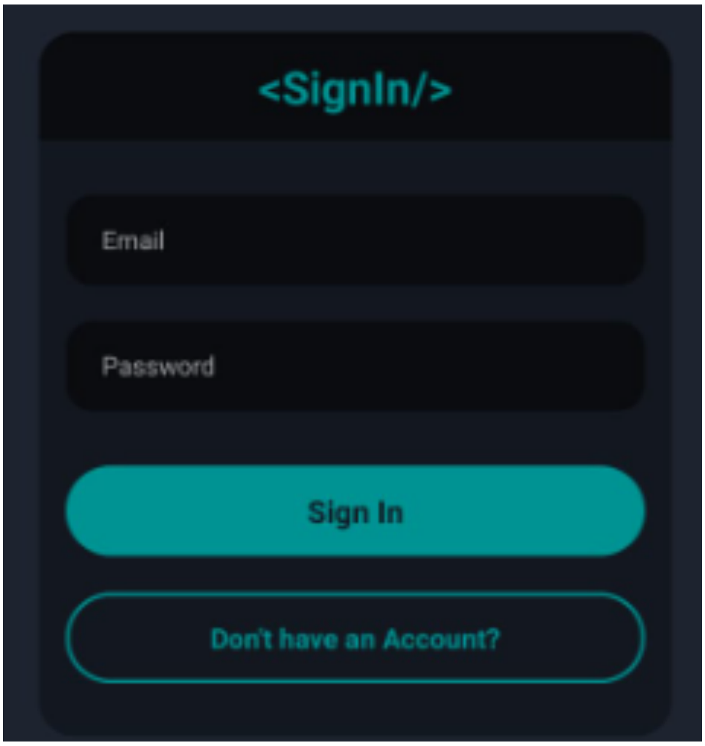
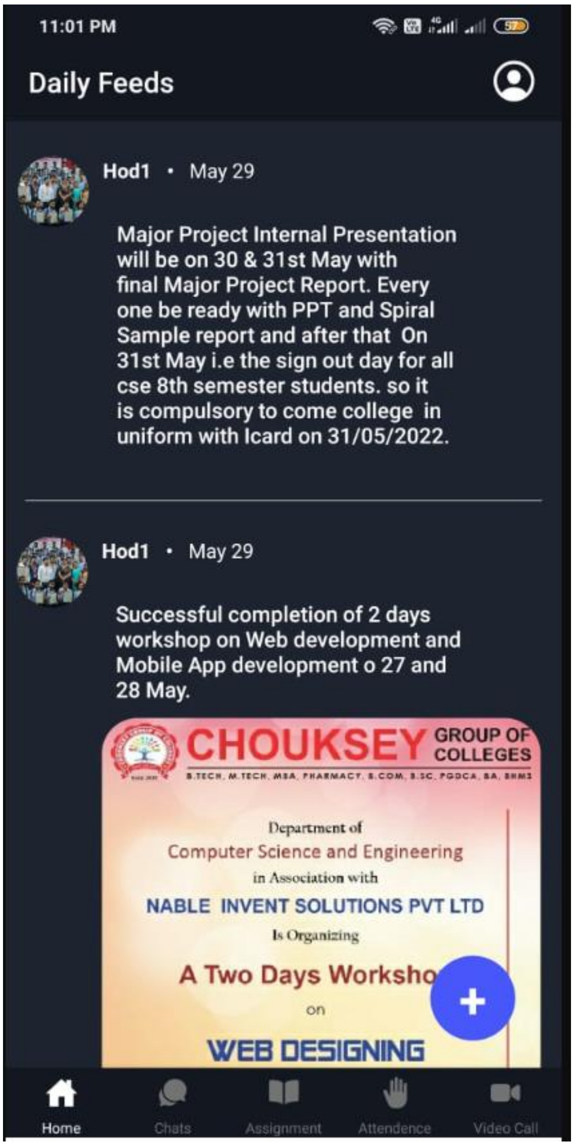


Fig. 9.2 User Sign-in

9.2 Feed Section

9.2.1 Feed View



11:01 PM

Daily Feeds

Hod1 • May 29

Major Project Internal Presentation will be on 30 & 31st May with final Major Project Report. Every one be ready with PPT and Spiral Sample report and after that On 31st May i.e the sign out day for all cse 8th semester students. so it is compulsory to come college in uniform with Icard on 31/05/2022.

Hod1 • May 29

Successful completion of 2 days workshop on Web development and Mobile App development o 27 and 28 May.

CHOUKSEY GROUP OF COLLEGES
B.TECH, M.TECH, MBA, PHARMACY, B.COM, B.SC, PGDCA, BA, BHMS
Department of
Computer Science and Engineering
in Association with
NABLE INVENT SOLUTIONS PVT LTD
Is Organizing
A Two Days Worksho
on
WEB DESIGNING
+
Home
Chats
Assignment
Attendence
Video Call

Fig. 9.3 Feed Section View

9.2.2 Feed Creation

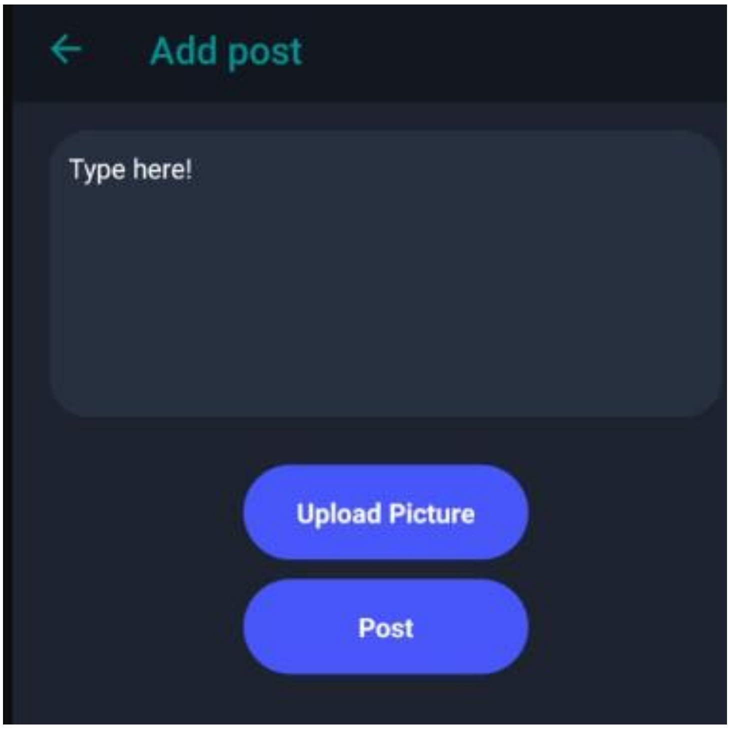
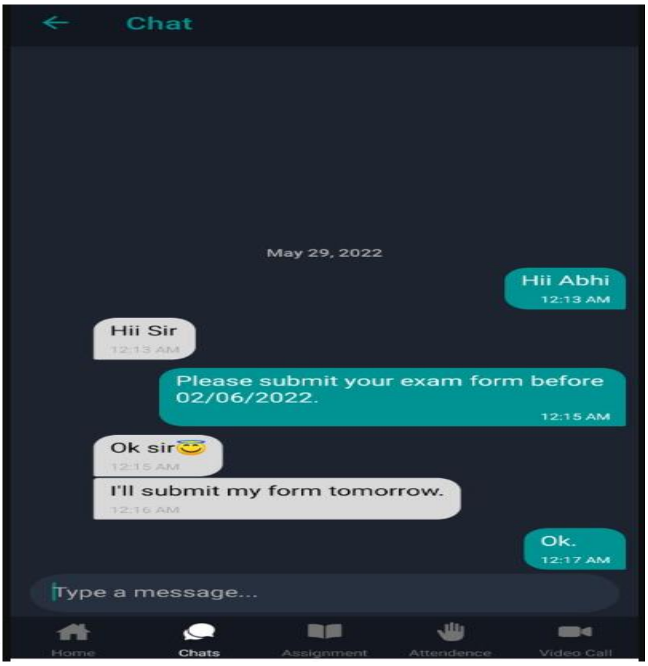


Fig. 9.4 Feed Creation

9.3 Chat Section

9.3.1 Chat Window



May 29, 2022

Hii Abhi
12:13 AM

Hii Sir
12:13 AM

Please submit your exam form before 02/06/2022.
12:15 AM

Ok sir😇
12:15 AM

I'll submit my form tomorrow.
12:16 AM

Ok.
12:17 AM

‘

Fig. 9.5 Chat Window

9.3.2 Recent Chats

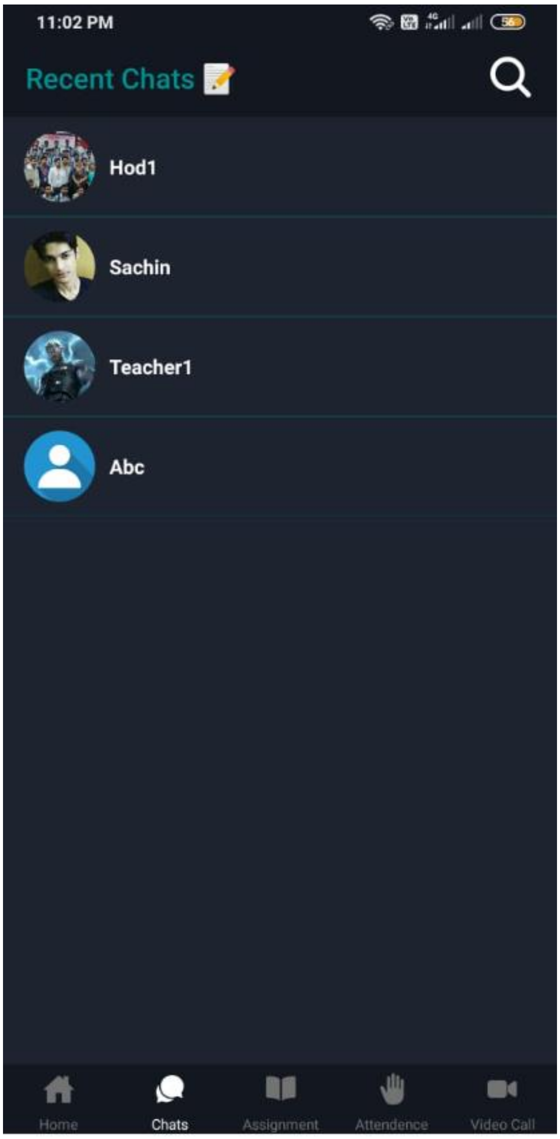


Fig. 9.6 Recent Chats

9.4 Assignment and Task Creation

9.4.1 Assignment Creation

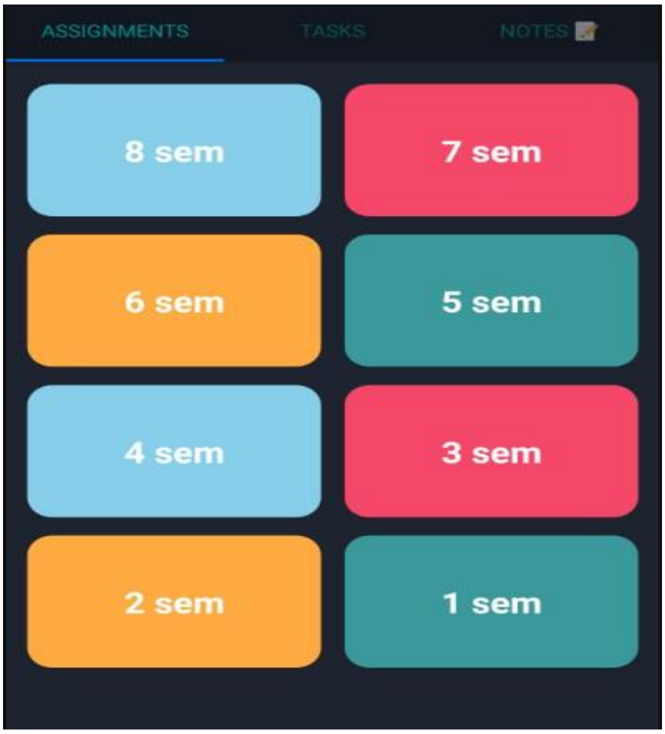


Fig. 9.7 Assignment Semester Selection

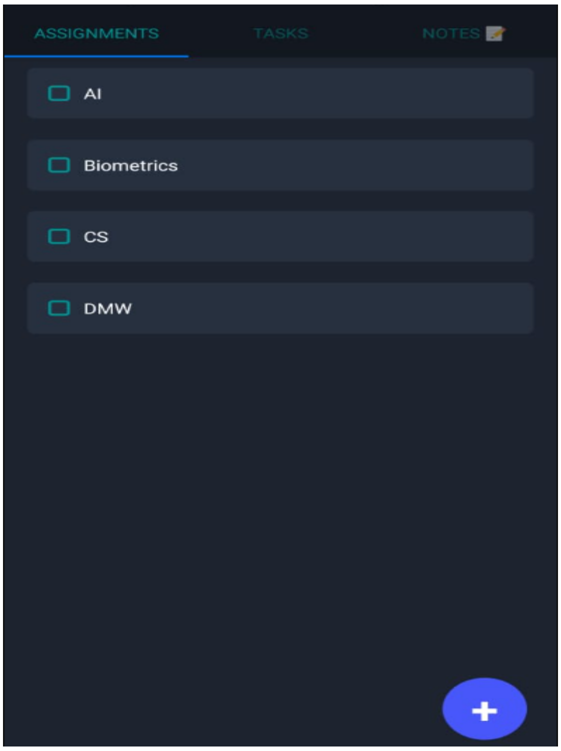


Fig. 9.8 Assignment Subject Selection

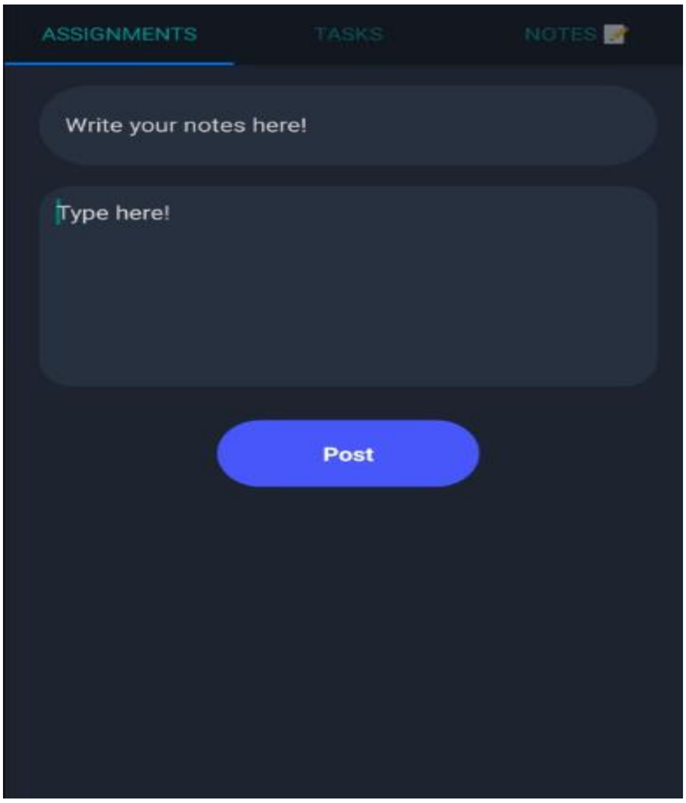
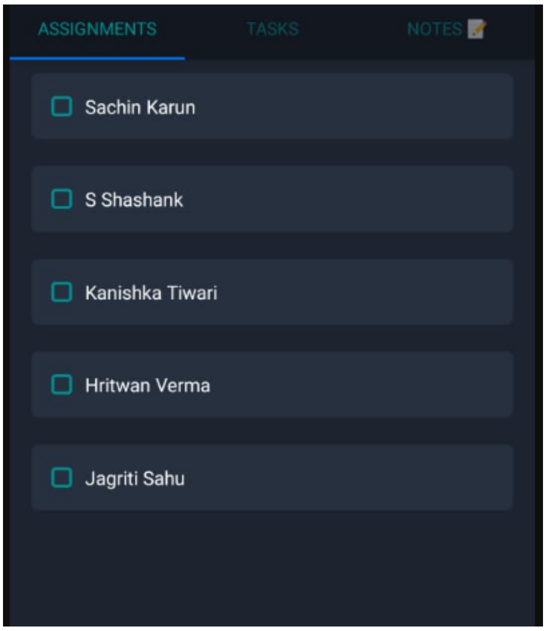


Fig. 9.9 Assignment Creation

9.4.2 Assignment Submission



□ Sachin Karun

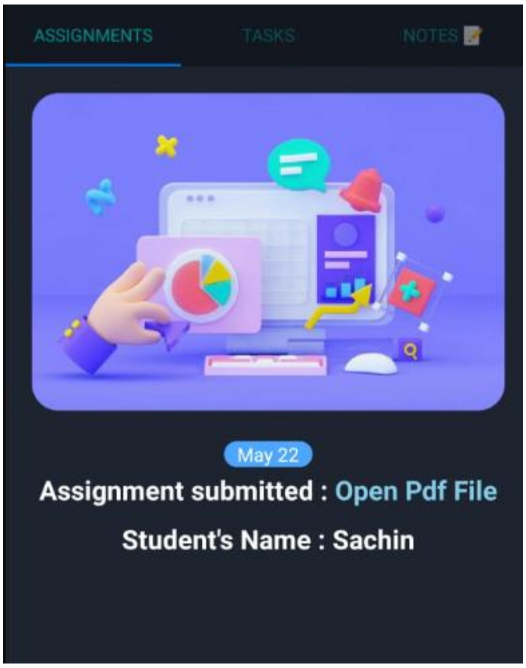
□ S Shashank

□ Kanishka Tiwari

□ Hritwan Verma

□ Jagriti Sahu

Fig. 9.10 Assignment Submission



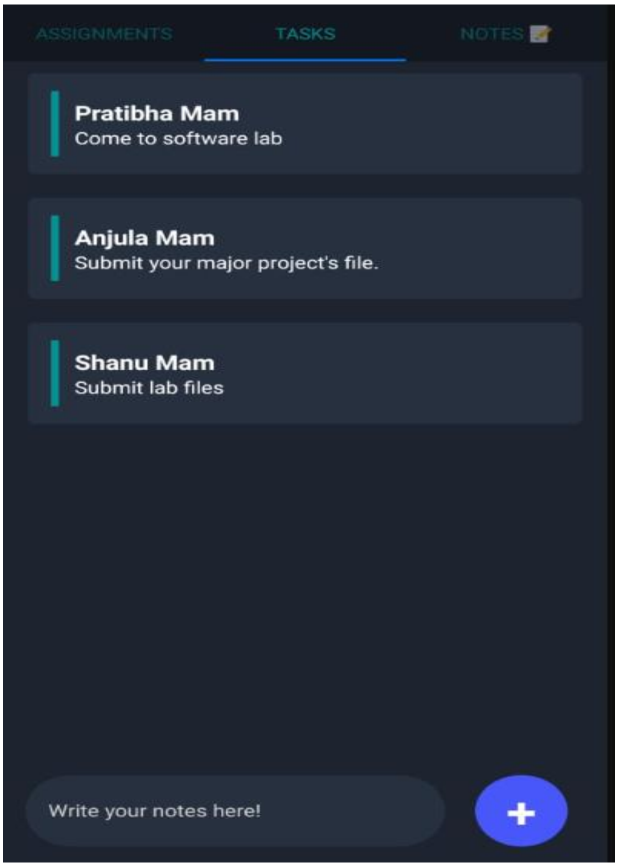
May 22

Assignment submitted : Open Pdf File

Student's Name : Sachin

Fig. 9.11 Assignment View

9.4.3 Task Creation



Pratibha Mam
Come to software lab

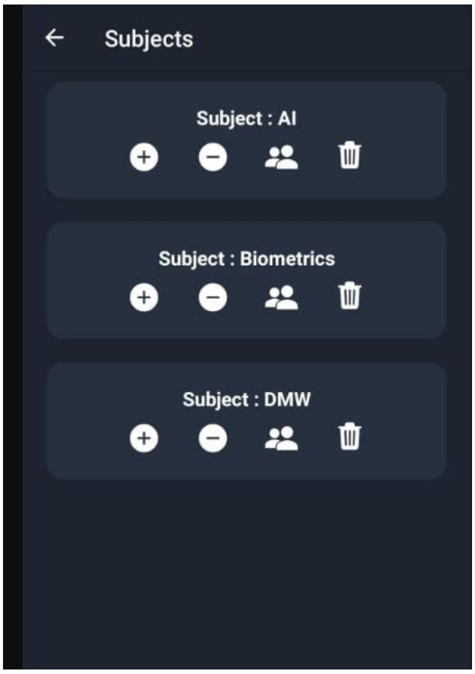
Anjula Mam
Submit your major project's file.

Shanu Mam
Submit lab files

Write your notes here!

Fig. 9.12 Task Creation

9.5 Attendance Section

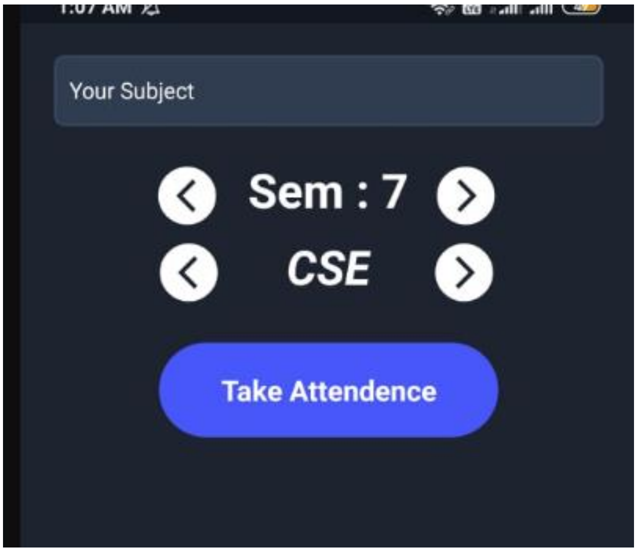


Subject : AI

Subject : Biometrics

Subject : DMW

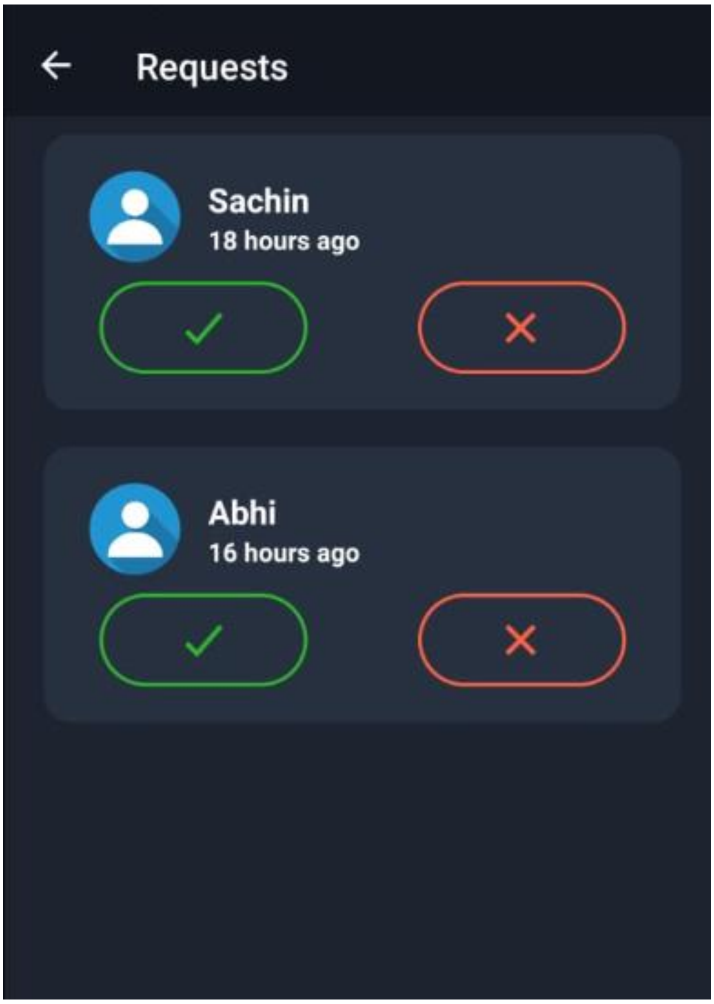
Fig. 9.13 Attendance Subject Creation



Sem : 7

CSE

Fig. 9.14 Attendance Creation



Requests

Sachin

18 hours ago

Abhi

16 hours ago

Fig. 9.15 Attendance Confirmation

9.6 Video Conferencing

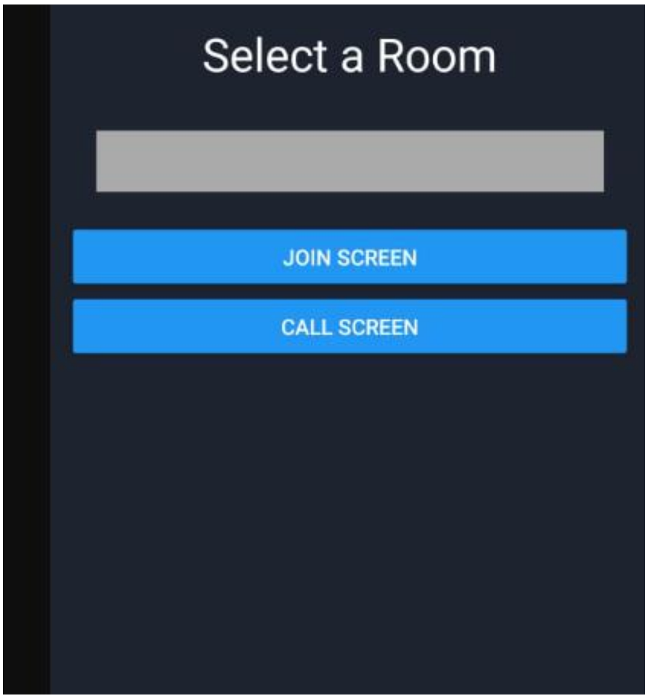


Fig. 9.16 Video Conferencing Room Creation

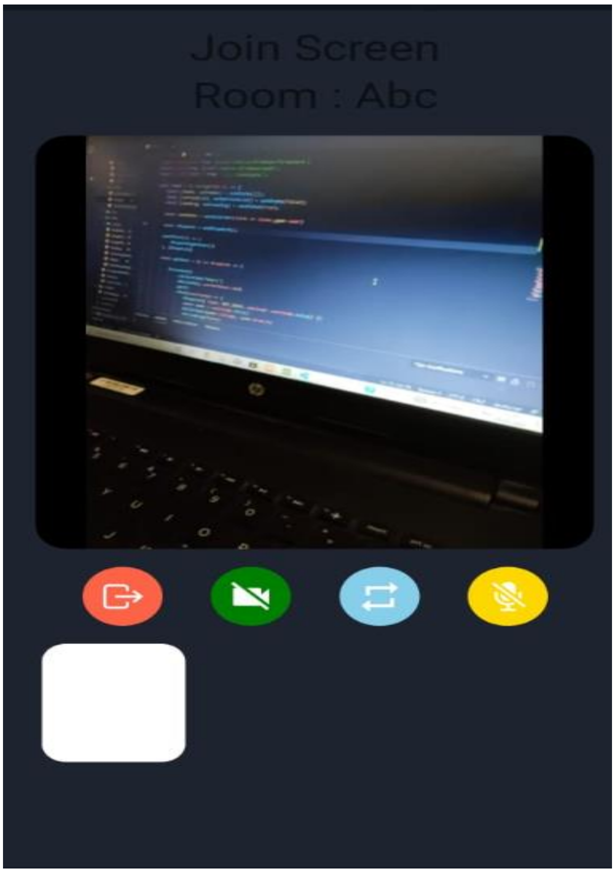
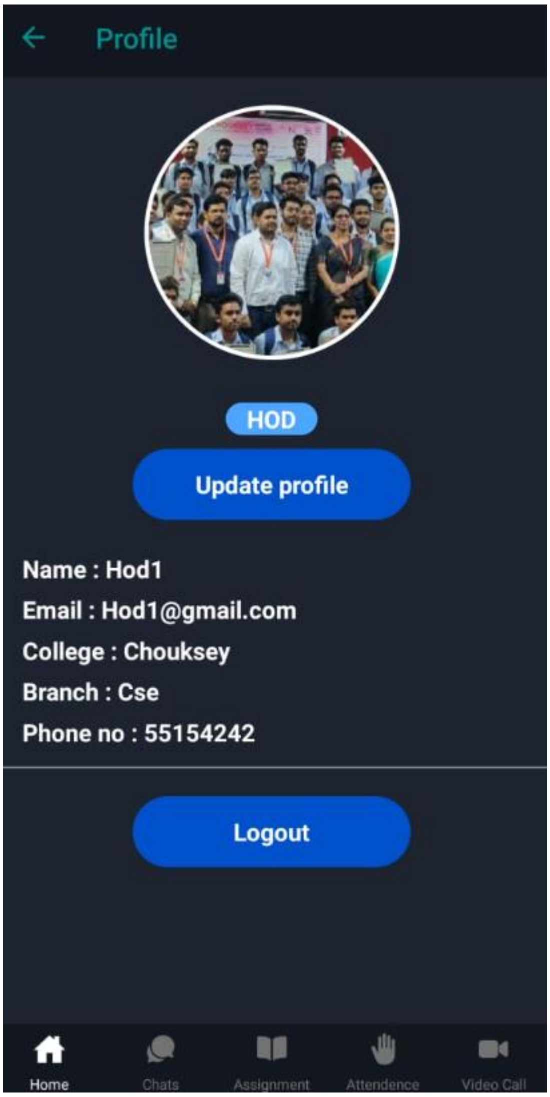


Fig. 9.17 Video Call Screen

9.7 User Profile



Name : Hod1

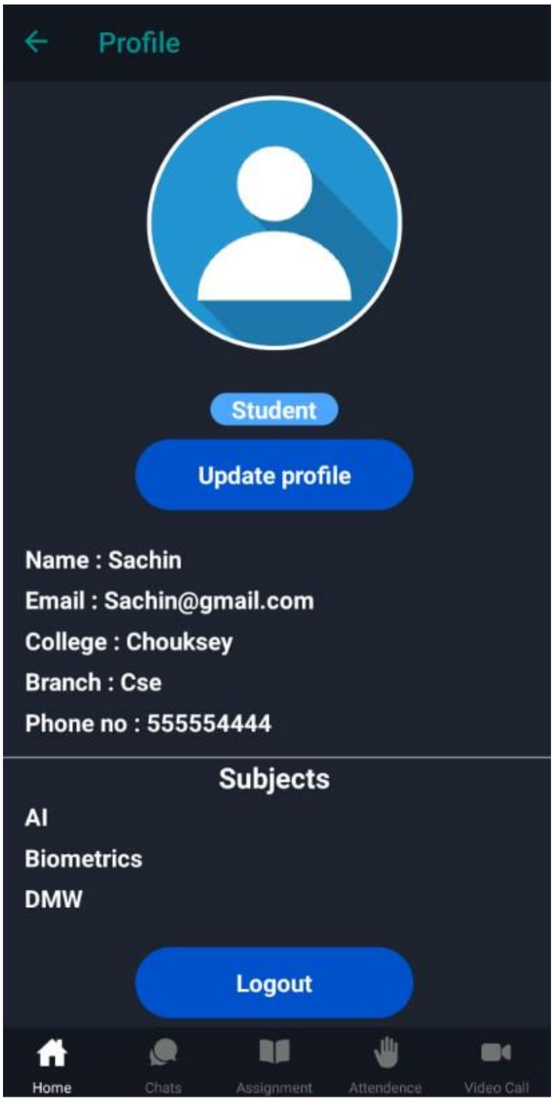
Email : Hod1@gmail.com

College : Chouksey

Branch : Cse

Phone no : 55154242

Fig. 9.18 Teacher Profile



Profile

Student

Update profile

Name : Sachin

Email : Sachin@gmail.com

College : Chouksey

Branch : Cse

Phone no : 555554444

Subjects

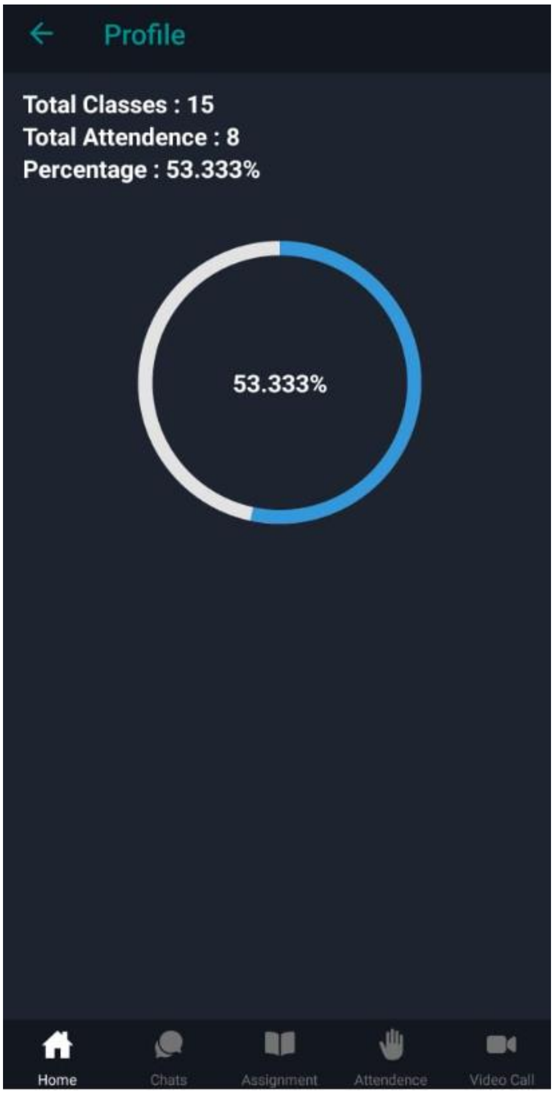
AI

Biometrics

DMW

Logout

Fig. 9.19 Student Profile



Profile

Total Classes : 15
Total Attendence : 8
Percentage : 53.333%

Fig. 9.20 Student Statistics

CHAPTER 10

CONCLUSION & FUTURE WORK

CHAPTER - 10
CONCLUSION & FUTURE WORK

10.1 Conclusion

- We can conclude that the hustle required by the students and teachers is reduced as of the current system.
- A user-friendly and interactive interface of the application makes it easy for the user to perform their tasks.
- The application is cost-effective and optimizes the memory usage and smartphone performance.

10.2 Future Scope

- Timed/proctored assessments for student evaluation.
- Grading System to record student performance.
- Common repository of resources such as text books, question papers etc.
- Integration with AI based libraries such as TensorIO and Tensorflowjs to implement intelligent notes creation.

Ultimately, the aim of this project is to create a one stop user-friendly application for online class conduction in colleges.

CHAPTER 11

REFERENCES

REFERENCES

[1] Pratama, Hendri & M.N.A, Azman & Kassymova, G. & Duisenbayeva, Shakizat. (2020). The Trend in Using Online Meeting Applications for Learning During the Period of Pandemic COVID-19: A Literature Review. Journal of Innovation in Educational and Cultural Research. 1. 58-68. 10.46843/jiecr.v1i2.15.

[2] Azhar, Kaukab & Iqbal, Nayab. (2018). Effectiveness of Google Classroom: Teachers' Perceptions. 2. 52-66.

[3] Buchal, Ralph & Songsore, Emmanuel. (2019). USING MICROSOFT TEAMS TO SUPPORT COLLABORATIVE KNOWLEDGE BUILDING IN THE CONTEXT OF SUSTAINABILITY ASSESSMENT. Proceedings of the Canadian Engineering Education Association (CEEA). 10.24908/pceea.vi0.13806.

[4] Thakker, S.V., Parab, J. and Kaisare, S. (2021), "Systematic research of e-learning platforms for solving challenges faced by Indian engineering students", *Asian Association of Open Universities Journal*, Vol. 16 No. 1, pp. 1-19.

[5] Guzacheva, N.. (2020). Zoom Technology as an Effective Tool for Distance Learning in Teaching English to Medical Students. Bulletin of Science and Practice. 6. 457-460. 10.33619/2414-2948/54/61.

[6] Bonnie Eisenman. 2016. Learning React Native: Building Native Mobile Apps with JavaScript (1st. ed.). O'Reilly Media, Inc.

[7] Akshat Paul and Abhishek Nalwaya. 2019. React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Applications (2nd. ed.). Apress, USA.

CHAPTER 12

APPENDICES

APPENDIX – I

TECHNOLOGIES USED

React Native

React Native is an open-source UI software framework created by Meta Platforms, Inc. It is used to develop applications for Android, Android TV, iOS, macOS, tvOS, Web, Windows and UWP by enabling developers to use the React framework along with native platform capabilities. It is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It's based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. React Native applications are written using a mixture of JavaScript and XML like markup, known as JSX.

Firebase

Firebase is a mobile and web application platform which provides tools and technology to build applications. The most popular services are Firebase Analytics, Firebase Cloud Messaging, Firebase Auth, Realtime Database, Firebase Storage and Firebase Hosting. The Firebase Realtime Database is a NoSQL Database which has a lot of optimizations and features compared with most of relational databases. It includes a flexible rule to define how the data should be structured to provide security and flexibility. Firebase is a Database stored as JSON objects, which is easier to use than some SQL databases for the way to handle the data like a tree. When you start adding data to your database, it automatically creates a node in the existing JSON structure with an associated key.

APPENDIX – II

INTRODUCTION TO JAVA SCRIPT

JavaScript (JS) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

Features of JavaScript

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

History of JavaScript

In 1993, Mosaic, the first popular web browser, came into existence. In the year 1994**,** Netscape was founded by Marc Andreessen. He realized that the web needed to become more dynamic. Thus, a 'glue language' was believed to be provided to HTML to make web designing easy for designers and part-time programmers. Consequently, in 1995, the company recruited Brendan Eich intending to implement and embed Scheme programming language to the browser. But, before Brendan could start, the company merged with Sun Microsystems for adding Java into its Navigator so that it could compete with Microsoft over the web technologies and platforms. Now, two languages were there: Java and the scripting language. Further, Netscape decided to give a similar name to the scripting language as Java's. It led to 'Javascript'. Finally, in May 1995, Marc Andreessen coined the first code of Javascript named 'Mocha'. Later, the marketing team replaced the name with 'LiveScript'. But, due to trademark reasons and certain other reasons, in December 1995, the language was finally renamed to 'JavaScript'. From then, JavaScript came into existence.

CHOUKSEY ENGINEERING COLLEGE, BILASPUR (C.G.)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

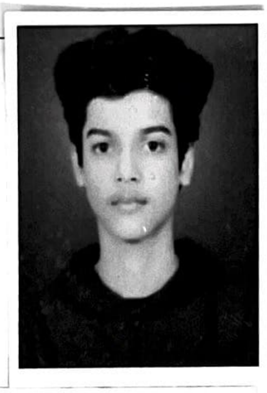
B.TECH VIII SEMESTER
MAJOR PROJECT MONITORING REPORT

GROUP No.:

PROJECT TITLE: Pocket College - Mobile Application for online class conduction

GUIDE: Prof. Anjula Shukla Mam

Weekly (Date)	Progress/Status	Guide Signature	Project In-charge
12/04/22	TOPIC APPROVAL		
18/04/22	Idea Presentation		
19/04/22	UI design		
20/04/22	Videocall & Feed session		
21/04/22	chatting, Registration & Login		
22/04/22	Attendence UI design		
26/04/22	Assignment Section UI		
30/04/22	Task and Notes Section UI		
07/05/22	Project Completion		
14/05/22	Documentation Work		
17/05/22	Presentation Completion		



Sachin

(Signature of the candidate)

Name: Sachin Karun

Enrollment Number : BG1125

Roll Number: 300202218025

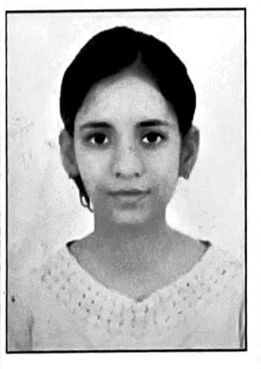
S.Shashank

(Signature of the candidate)

Name: S.Shashank

Enrollment Number: BI3718

Roll Number: 300202220542



(Signature of the candidate)

Name: KarishKa Tiwari

Enrollment Number : BI3717

Roll Number: 300202220541

(Signature of the candidate)

Name: Hritwan Verma

Enrollment Number: BG1102

Roll Number: 300202218050

Signature of Project In-Charge
Dr. (Mrs.) Shanu K. Rakesh
Assistant Professor, CSE
CEC, Bilaspur(C.G.)

Signature of H.O.D
Dr. (Mrs.) Shanu K. Rakesh
Assistant Professor, CSE
CEC, Bilaspur(C.G