A Project Report on

METEORITE

Submitted by

P Jagan Mohan – R171149

Submitted to

IIIT RK VALLEY

Idupulapaya, Vempalli, YSR Kadapa

Andhra Pradesh, India PIN 516330

Under the guidance of

Mr. Satyanandaram N

Assistant Professor

HOD Of CSE

As a part of

Partial fulfilment of the degree of Bachelor of Technology in

Computer Science and Engineering

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

R.K. Valley, Kadapa (Dist), Andhra Pradesh, 516330

CERTIFICATE

This is certify that the project entitled "**METEORITE**" submitted by **P Jagan Mohan** (**R171149**) under our guidance and supervision for the partial fulfilment for degree Bachelor of Technology in Computer Science and Engineering – 4 during the academic semester – I and semester – II 2022-2023 at RGUKT, RK VALLEY. To the best of our knowledge, the report has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma

Project Internal Guide

N Satyanandaram,
Assistant Professor
Computer Science and Engineering
RK Valley, RGUKT.

Head of the Department

N Satyanandaram
Assistant Professor
Computer Science and Engineering
RK Valley, RGUKT.

Submitted for the practical examination held on …….

Internal Examiner **External Examiner**

Acknowledgement

We would like to express our sincere gratitude to **Mr. N Satyanandaram** Sir, our project internal guide for valuable suggestions and keen interest throughout the progress of my course of research.

Again, I am grateful to **Mr. N Satyanandaram** sir, HOD CSE, for providing excellent computing facilities and a congenial atmosphere for progressing with our project.

At the outset, we would like to thank **Rajiv Gandhi University of Knowledge Technologies** for providing all the necessary resources for the successful completion of our course work.

Index

S.no	Title	Page no
1	Abstract	5
2	Introduction	6 – 7
3	Technologies	7 - 8
4	Software Configurations	8
5	Design	9
6	Use Case Diagram	10
7	ER Diagram	11
8	Coding	12 – 21
9	Testing	22
10	Future Improvements	24
11	Snippets	25
12	References	28

Abstract

METEORITE is a full stack mobile application for both ANDROID and IOS. The Application focus on the student full pledged education system along with Entertainment. In this Major project module. We are focusing on the high standard quality education, Simplified university/board wise patterns. Our application helps the student to maintain their time in studies and have fun while learning.

The application provides good quality of simplified standard content according to their school/university syllabus and student should be focused on the studies with friendly environment with their friends. Not only for reference, this app will help the student recall and know how much they have learned by mock test. And so many features are awaiting to develop more accuracy for the student. The system is implemented in React Native, Amazon Web Services, Type Script, GraphQL API.

Introduction

The goal of this project is to provide high standard quality education with proficient content and lecture guidance through online. Our application will provide Academic content complete and simplified along with extra motivated and development lectures. Including that our application provides entertainment feed which is our algorithm focus on surrounding/campus friends, they can share their memories and ideas through their friends and much more.

1.1: Purpose

Meteorite app would allow users to study, improve external knowledge and entertainment through social media. They can communicate with their campus/school students. The main purpose to provide high standard quality education to every student through online and to share their ideas/memories with their friends.

1.2: Intended Audience:

The intended audience will be the users, System User, Super Admin. Where the users can watch all the posts and minimized study content. System user can post/upload and delete their posts along with he can give answer to student questions and progress will be monitored. Super admin protect all the users details and user privacy and security along with application terms and conditions protector.

Product Vision:

METEORITE creates a platform that provides users with an easy and stress-free study environment along with refreshment of social media. Every eligible student can access all the academic content both simplifies and raw materials along with extra knowledge which is mostly need to their standards.

Technologies:

➢ React Native

➢Amazon Web Services

➢Type script

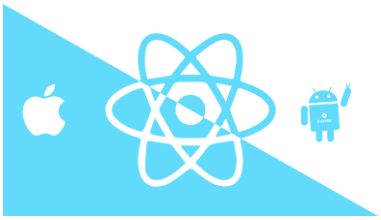
➢AWS Amplify

➢GraphQL API

➢ Expo Application Services

React Native :

React Native is an open source JavaScript/TypeScript Mobile framework from Facebook specially designed to build native mobile apps for iOS and Android. React Native is based on ReactJS JavaScript library that helps to build the user interface for mobile platforms.



React Native can be directly used inside an existing IOS or android app or you can build a native app right from scratch. At present React Native is used with some popular apps like Facebook mobile app, Instagram, Pinterest, Skype, etc.

Some important features of React Native that makes it a very popular mobile development app today are –

Cross Platform Support – To develop mobile apps you don't need a team expert in IOS and Android programming instead JavaScript developers who are enthusiastic to build apps can use React Native to build native apps without having to learn Kotlin or Java for Android and Swift or Objective-C for IOS apps. You can write one common code and React Native will take care of displaying it in IOS and Android.

React Native Components – React Native offers native components like View, Text, and Image that are changed to the IOS or Android native UI.

Amazon Web Services :

Amazon Web Services(AWS) is a cloud service from Amazon, which provides services in the form of building blocks, these building blocks can be used to create and deploy any type of application in the cloud.

These services or building blocks are designed to work with each other, and result in applications that are sophisticated and highly scalable.

The Amazon Web Services (AWS) platform provides more than 200 fully featured services from data centers located all over the world, and is the world's most comprehensive cloud platform. Amazon web service is an online platform that provides scalable and cost-effective cloud computing solutions.

AWS is a broadly adopted cloud platform that offers several on-demand operations like compute power, database storage, content delivery, etc., to help corporates scale and grow.

Each type of service in this “What is AWS” blog, is categorized under a domain, the few domains which are widely used are:

- Compute
- Storage
- Database
- Migration
- Network and Content Delivery
- Management Tools
- Security & Identity Compliance
- Messaging

Advantages of AWS

Easy to use

AWS is made to enable suppliers, ISVs, and application providers to swiftly and securely host your apps, whether they are SaaS-based or not. To access AWS’s application hosting platform, use the AWS Management Console or well-documented web services APIs.

Flexible

You can choose the web application platform, programming language, operating system, database, and other services you require with AWS. You get a virtual environment through AWS that you may fill with the programs and services your application needs. As a result, existing applications can be more easily migrated while still having alternatives for developing new solutions.

Cost-Effective

There are no long-term contracts or upfront payments; you simply pay for the computing power, storage, and other resources that you really utilize. Visit the AWS Economics Center for further details on comparing the expenses of other hosting options with those of AWS.

Reliable

You can benefit from a scalable, reliable, and secure global computing infrastructure with AWS, which serves as the virtual foundation for Amazon.com's multi-billion dollar online company and has been refined for more than ten years.

Scalable and high-performance

Your application can scale up or down depending on demand using AWS technologies like Auto Scaling and Elastic Load Balancing. You get immediate access to computation and storage resources thanks to Amazon's extensive infrastructure.

Secure

Physical, operational, and software safeguards are all used by AWS to secure and harden our infrastructure. The AWS Security Center has more details.

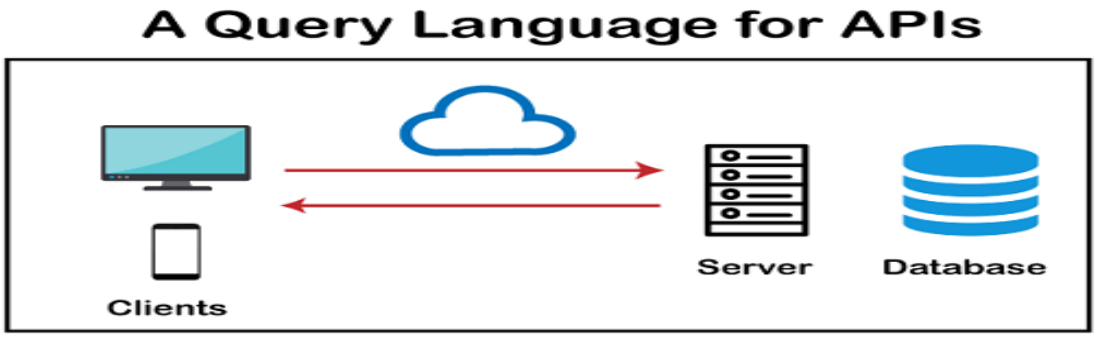
GraphQL API :

GraphQL is an open-source data query and manipulation language for APIs and a runtime for fulfilling queries with existing data. It's neither an architectural pattern nor a web service. GraphQL was developed internally by Facebook in 2012 before being publicly released in 2015.

GraphQL is a new **API standard invented and developed by Facebook**. It is **an open-source** server-side technology, now maintained by a large community of companies and individuals of all over the world. It is also an execution engine that works as a data query language and used to fetch declarative data.

It was developed to optimize RESTful API calls and provides a flexible, robust, and more efficient alternative to REST.

GraphQL is data query and manipulation language for your API and a server-side runtime for executing queries when you define a type system for your data. Unlike the REST APIs, a GraphQL server provides only a single endpoint and responds with the precise data that a client asked for.



Design

Modules

1.Authentication

2.Home/Social Media

3.Study

4.Profile

5.Notifications

Authentication:

Social login is a single sign-on (SSO) technology that allows users to authenticate themselves on various applications and sites by connecting through a social media site rather than typing a separate ID and password on each website. The sites most commonly associated with social login are Facebook, LinkedIn, Google and Twitter.

Login is the process of **accessing an existing account using a username and password**. It is used to grant access to the user's personal information, settings, and other resources. Sign up is the process of creating a new account. It involves entering personal information and setting up a username and password.

Home/Social Media:

Social networking refers to using internet-based social media sites to stay connected with friends, family, colleagues, or customers. Social networking can have a social purpose, a business purpose, or both through sites like Facebook, Twitter, Instagram, and Pinterest.

Social networking is also a significant opportunity for marketers seeking to engage customers. Facebook remains the largest and most popular social network, with 2 billion people using the platform daily, as of Feb 1, 2023. Other popular platforms in the U.S. are Instagram, Twitter, WhatsApp, TikTok, and Pinterest.

- Social networking uses internet-based social media platforms to connect with friends, family, or peers.
- Some of the most popular social networking sites in the U.S. include Facebook, Instagram, TikTok, WhatsApp, and Twitter.
- Marketers use social networking to increase brand recognition and encourage brand loyalty.
- Social media can help connect people with businesses for various needs.

Study:

Virtual learning is usually associated with online courses or online environments, but it has much broader dimensions. In this article we will discuss its definition, characteristics, and benefits. We have also made for you a list of the most common forms of virtual learning, which describe the different aspects of learning and teaching process.

Virtual learning is a learning experience that is enhanced through utilizing computers and/or the internet both outside and inside the facilities of the educational organization. The instruction most commonly takes place in an online environment. The teaching activities are carried out online whereby the teacher and learners are physically separated (in terms of place, time, or both).

The study field examines how students behave while learning. It focuses on how learners respond to certain stimuli. When the teacher repeats the stimuli, they can observe, control, and modify the learner's individual behavior. Learners do what they are instructed to do and are only prepared to reproduce basic facts and automatically perform tasks. Behaviorism does not examine the mind or cognitive processes.

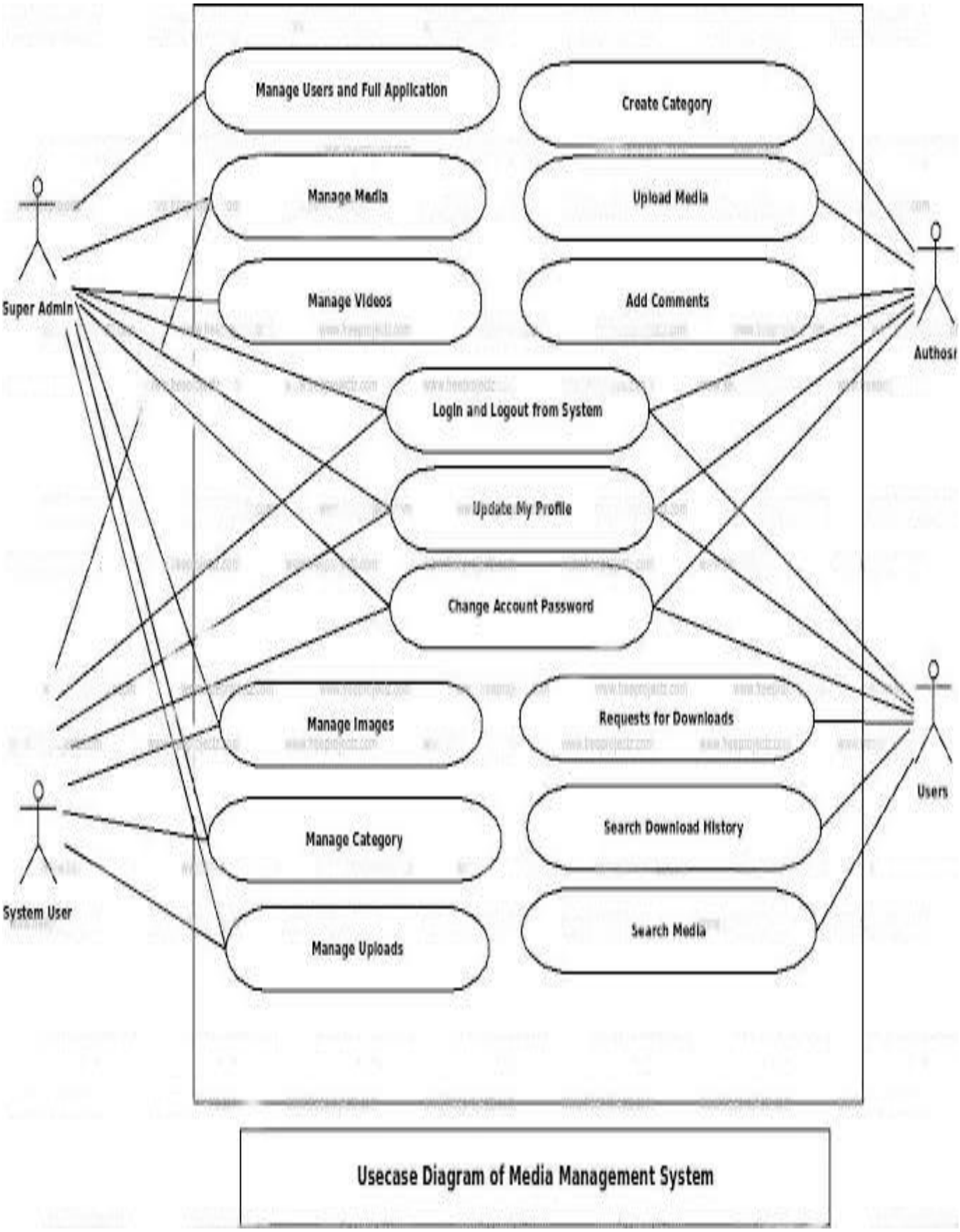
In virtual learning behaviorism can be applied through step-by-step video tutorials, game-based activities, regular and constructive feedback, quizzes, gamification, etc.

Profile:

A **user profile** is a collection of settings and information associated with a user. It contains critical information that is used to identify an individual, such as their name, age, portrait photograph and individual characteristics such as knowledge or expertise. User profiles are most commonly present on social media websites such as Facebook, Instagram,and LinkedIn; and serve as voluntary digital identity of an individual, highlighting their key features and traits. In personal computing and operating systems, user profiles serve to categorise files, settings, and documents by individual user environments, known as 'accounts', allowing the operating system to be more friendly and catered to the user. Physical user profiles serve as identity documents such as passports, driving licenses and legal documents that are used to identify an individual under the legal system.

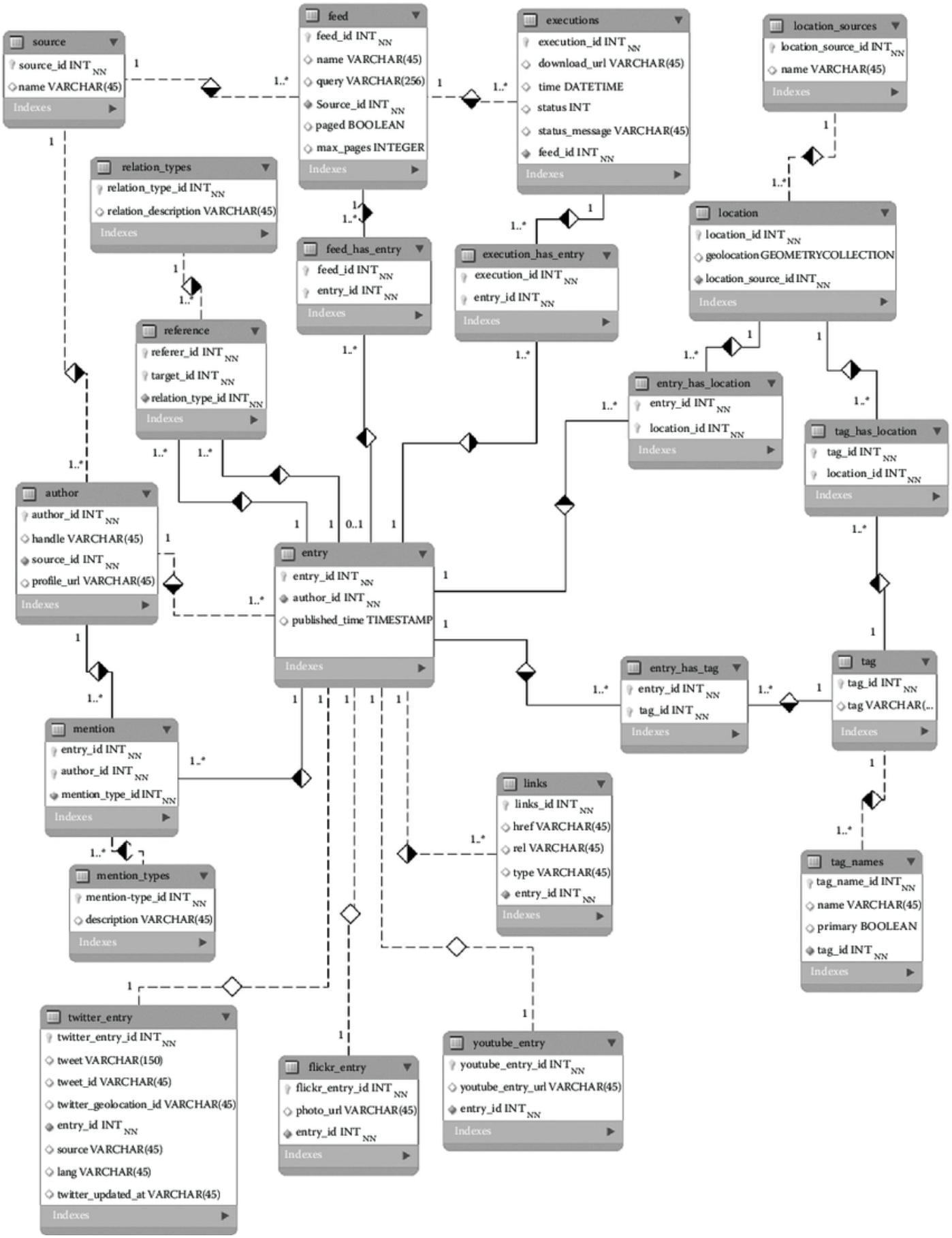
A user profile can also be considered as the computer representation of a user model. A user model is a (data) structure that is used to capture certain characteristics about an individual user, and the process of obtaining the user profile is called user modelling or profiling.

Use Case Diagram :



Usecase Diagram of Media Management System

ER Diagram :


source
source_id INT NN
name VARCHAR(45)
Indexes
feed
feed_id INT NN
name VARCHAR(45)
query VARCHAR(256)
Source_id INT NN
paged BOOLEAN
max_pages INTEGER
Indexes
executions
execution_id INT NN
download_url VARCHAR(45)
time DATETIME
status INT
status_message VARCHAR(45)
feed_id INT NN
Indexes
location_sources
location_source_id INT NN
name VARCHAR(45)
Indexes
relation_types
relation_type_id INT NN
relation_description VARCHAR(45)
Indexes
feed_has_entry
feed_id INT NN
entry_id INT NN
Indexes
execution_has_entry
execution_id INT NN
entry_id INT NN
Indexes
location
location_id INT NN
geolocation GEOMETRYCOLLECTION
location_source_id INT NN
Indexes
reference
referer_id INT NN
target_id INT NN
relation_type_id INT NN
Indexes
entry_has_location
entry_id INT NN
location_id INT NN
Indexes
tag_has_location
tag_id INT NN
location_id INT NN
Indexes
author
author_id INT NN
handle VARCHAR(45)
source_id INT NN
profile_url VARCHAR(45)
Indexes
entry
entry_id INT NN
author_id INT NN
published_time TIMESTAMP
Indexes
entry_has_tag
entry_id INT NN
tag_id INT NN
Indexes
tag
tag_id INT NN
tag VARCHAR(..
Indexes
mention
entry_id INT NN
author_id INT NN
mention_type_id INT NN
Indexes
links
links_id INT NN
href VARCHAR(45)
rel VARCHAR(45)
type VARCHAR(45)
entry_id INT NN
Indexes
mention_types
mention-type_id INT NN
description VARCHAR(45)
Indexes
tag_names
tag_name_id INT NN
name VARCHAR(45)
primary BOOLEAN
tag_id INT NN
Indexes
twitter_entry
twitter_entry_id INT NN
tweet VARCHAR(150)
tweet_id VARCHAR(45)
twitter_geolocation_id VARCHAR(45)
entry_id INT NN
source VARCHAR(45)
lang VARCHAR(45)
twitter_updated_at VARCHAR(45)
Indexes
flickr_entry
flickr_entry_id INT NN
photo_url VARCHAR(45)
entry_id INT NN
Indexes
youtube_entry
youtube_entry_id INT NN
youtube_entry_url VARCHAR(45)
entry_id INT NN
Indexes

Coding

HomeScreen:

```
import {FlatList ,ViewabilityConfig,ViewToken,View} from 'react-native';
import FeedPost from '../../components/FeedPost';
import posts from '../../assets/data/posts.json'
import { useRef, useState } from 'react';

const HomeScreen = () => {
  const [activePostId, setActivePostId] = useState<string | null>(null);

  const viewabilityConfig: ViewabilityConfig = {
    itemVisiblePercentThreshold: 51,
  };

  const onViewableItemsChanged = useRef(
    ({viewableItems}: {viewableItems: Array<ViewToken>}) => {
      if (viewableItems.length > 0) {
        setActivePostId(viewableItems[0].item.id);
      }
    },
  );

  return (
    <View>
      <FlatList
        data={posts}
        renderItem={({item}) => <FeedPost post={item} isVisible={activePostId === item.id}/>}
        showsVerticalScrollIndicator={false}
        viewabilityConfig={viewabilityConfig}
        onViewableItemsChanged={onViewableItemsChanged.current}
      />
    </View>
  )
```

```
}

export default HomeScreen;
```

Feed Post:

```
import {useState} from 'react';
import { StyleSheet, Text, View,Image,Pressable } from 'react-native';
import {useNavigation} from '@react-navigation/native'
import colors from '../../theme/colors';
import fonts from '../../theme/fonts';
import Entypo from 'react-native-vector-icons/Entypo';
import AntDesign from 'react-native-vector-icons/AntDesign';
import Feather from 'react-native-vector-icons/Feather';
import Ionicons from 'react-native-vector-icons/Ionicons';
import MaterialCommunityIcons from 'react-native-vector-icons/MaterialCommunityIcons';
import Comment from '../Comment';
import { Ipost } from '../../types/models';
import styles from './styles';
import DoublePressable from '../DoublePressable';
import Carousel from '../Carousel';
import VideoPlayer from '../VideoPlayer';
import { FeedNavigationProp } from '../../types/navigation';

interface IFeedPost {
  post: Ipost;
  isVisible: boolean;
}

const FeedPost = (props : IFeedPost) => {
  const {post,isVisible = false} = props
  const [isDescriptionExpanded, setIsDescriptionExpanded] = useState(false);
  const [isLiked,setIsLiked] = useState(false)
  const navigation = useNavigation<FeedNavigationProp>();
```

```
const navigateToUser=()=> {
  navigation.navigate('UserProfile', {user: post.user})
};
const navigateToComments=()=> {
  navigation.navigate('Comments', {postId: post.id})
}

const toggleDescriptionExpanded =() => {
  setIsDescriptionExpanded(v => !v)
};

const toggleLike =()=> {
  setIsLiked(v=>!v)
};

let content = null;
if (post.image) {
  content =(
    <DoublePressable onDoublePress={toggleLike}>
    <Image source= {{
      uri: post.image,
    }}
    style={styles.image}
    />
    </DoublePressable>
  );
}else if (post.images) {
  content =<Carousel images= {post.images} onDoublePress={toggleLike} />;
} else if (post.video) {
  content = (
    <DoublePressable onDoublePress={toggleLike}>
    <VideoPlayer uri={post.video} paused={!isVisible}/>
```

```
    </DoublePressable>
  );
}
return (
  <View style={styles.post}>
    {/* header */}
    <View style={styles.header}>
      <Image
        source={{
          uri: post.user.image,
        }}
        style={styles.avatar}
      />
      <View style={{flexDirection: 'column'}}>
      <Text style={styles.username}>{post.user.username}</Text>
      <View style={{flexDirection:'row'}}>
      <Ionicons name="location-outline" size={16} color={colors.gray} />
      <Text>{' '}{post.location}</Text>
      </View>
      </View>
      <Entypo name="dots-three-vertical" size={18} style={styles.threedots}/>
    </View>

    {/* content */}
    {content}

    {/* footer */}
    <View style={styles.footer}>
    <View style={styles.iconcontainer}>
    <Pressable onPress={toggleLike} style={styles.ficons}>
    <AntDesign
      name= {isLiked ? 'heart' : 'hearto'}
      size={20}
```

```
          style={styles.icon}
          color={isLiked ? colors.accent : colors.gray}
        />
        <Text>{post.nofLikes +1}</Text>
        </Pressable>
        <View style={styles.ficons}>
        <Ionicons name="chatbubble-outline" size={20} style={styles.icon} color={colors.gray} />
        <Text>{post.nofComments}</Text>
        </View>
        <View style={styles.ficons}>
        <Ionicons name="stats-chart" size={20} style={styles.icon} color={colors.gray}/>
        <Text>{post.poll}</Text>
        </View>
        <View style={styles.ficons}>
        <MaterialCommunityIcons name="share" size={20} style={styles.icon} color={colors.gray}
/>
        <Text>{post.shares}</Text>
        </View>
        <Feather name="bookmark" size={20} style={{marginLeft:'auto'}} color={colors.gray} />
        </View>
        {/* likes */}
        <Text style={styles.text}>Liked by{' '}<Text style={styles.bold}>mahendra</Text> and{' '}
        <Text style={styles.bold}>{post.nofLikes} others</Text></Text>

        {/* discription */}
        <Text style={styles.text} numberOfLines={isDescriptionExpanded ? 0: 2} >
        <Text style={styles.bold}>{post.user.username}</Text>{' '}{post.description}</Text>
        <Text onPress={toggleDescriptionExpanded}> {isDescriptionExpanded ? ' ' : 'see
more'}</Text>
          {/* comments */}
         <Text onPress={navigateToComments}>View all {post.nofComments} comments</Text>
         <Text>{post.createdAt}</Text>
         {post.comments.map(comment => (
           <Comment key={comment.id} comment={comment} />
```

```
        ))}
      </View>

    </View>
  )
}

export default FeedPost
```

CourseScreen:

```
import { Dimensions, FlatList, StyleSheet, Text, View } from 'react-native'
import React from 'react'
import tution from '../../assets/data/tution.json'
import CourseHelper from './CourseHelper';

const {height, width} =Dimensions.get('window')

const Cource = () => {
  return (
    <View>
        <FlatList
        horizontal
        data={tution}
        renderItem={({item}) => <CourseHelper add={item} />}
        showsVerticalScrollIndicator={false}
      />
    </View>
  )
}

export default Cource
```

```
const styles = StyleSheet.create({
  c1:{
    width: width*0.7,
    height: height/8,
    //justifyContent:'center',
    alignItems:'center',
    marginTop:10,
    marginHorizontal:3,
    borderWidth:1,
    borderColor:'green',
    borderRadius:5,
    color:'#FF9F9F',
},
slide: {
    width: '97%',
    height:'100%',
    backgroundColor: '#FF9F9F',

    borderRadius: 5,
    resizeMode: 'contain',
    margin: 3,
    marginBottom:3,
},
})
```

EditProfile Screen:

```
import { View, Text, StyleSheet,Image,TextInput } from 'react-native'
import {useState} from 'react';
import {useForm, Controller, Control} from 'react-hook-form';
import { launchImageLibrary } from 'react-native-image-picker';
import user from '../../assets/data/user.json';
import colors from '../../theme/colors';
```

```
import fonts from '../../theme/fonts';
import { Iuser } from '../../types/models';

type IEditableUserField = 'name' | 'username' | 'website' | 'bio';
type IEditableUser =Pick <Iuser, IEditableUserField >;

interface ICustomInput {
   lable: string;
   multiline?: boolean;
   control: Control<IEditableUser,object>;
   name:IEditableUserField;
   rules?: object;
}

const CustomInput =({name, control,lable,multiline=false,rules={}}: ICustomInput) =>(
   <Controller
   control={control}
   name={name}
   rules={rules}
   render={({field: {onChange,value,onBlur},fieldState:{error}})=> {
      return(
      <View style={styles.inputContainer}>
      <Text style={styles.label}>{lable}</Text>
      <View style={{flex:1}}>
      <TextInput
         value={value}
         onChangeText={onChange}
         onBlur={onBlur}
         placeholder={lable}
         style={[styles.input,{borderColor: error? colors.error : colors.border,}]}
         multiline={multiline}
      />
      {error && <Text style={{color: colors.error}}>{error.message || 'Error'}</Text>}
```

```
        </View>

    </View>
    )}}
    />



)

const EditProfileScreen = () => {
    const [selectedPhoto, setSelectedPhoto] = useState<null | Asset>(null)
    const {control, handleSubmit,formState:{errors},} = useForm<IEditableUser>({
        defaultValues : {
            name: user.name,
            username: user.username,
            website: user.website,
            bio: user.bio
        }
    });

    const onSubmit = (data:IEditableUser) => {
        console.log('submit',data)
    }

    const onChangePhoto =()=> {
        launchImageLibrary (
            {mediaType:'photo'},
            ({didCancel, errorCode, errorMessage,assets}) => {
                if (!didCancel && !errorCode && assets &&assets.length > 0) {
                    setSelectedPhoto(assets[0]);
                }
            },
```

```
      );
    };
  return (
    <View style={styles.page}>
      <Image source={{uri: selectedPhoto?.uri || user.image}} style={styles.avatar} />
      <Text onPress={onChangePhoto} style={styles.textButton}>Change profile photo</Text>

      <CustomInput name="name" control={control} rules={{required: 'Name is required'}} lable="Name"/>
      <CustomInput name="username" control={control} rules={{required:"Username is required", minLength:{value:3, message: "Username should be atleast 3 characters"}}} lable="Username"/>
      <CustomInput name="website" control={control} rules={{required:false}} lable="Website"/>
      <CustomInput name="bio" control={control} rules={{required:"Bio is required", maxLength:{value:200, message: "Bio should not exceed 200 characters"}}} lable="Bio" multiline/>

      <Text onPress={handleSubmit(onSubmit)} style={styles.textButton}>Submit</Text>
    </View>
  );
};

const styles = StyleSheet.create ({
  page: {
    alignItems:'center',
    padding:10,
  },
  avatar:{
    width: '30%',
    aspectRatio:1,
    borderRadius:20,
  },
  textButton:{
    color: colors.primary,
```

```
        fontSize: fonts.size.md,
        fontWeight: fonts.weight.semi,
        margin: 10,
    },
    inputContainer: {
        flexDirection: 'row',
        alignItems: 'center',
        alignSelf: 'stretch',
    },
    label: {
        width: 75,
    },
    input: {

        borderBottomWidth:1
    },
})

export default EditProfileScreen
```

Testing

Here we performed two types of testing to the software for finding bugs

1.Functional Testing:

we tested main features like testing each and every module like login ,signup, Profile, social media, study & courses.

Integration Testing:

Here, the data flow is tested .For example ,if we take login module by entering valid credentials it redirects to the respected user's Dashboard .

System Testing:

Here, the end to end Testing is done on application from entering credentials, navigating to the all modules such as Profile of the User etc. and at last to the logout page.

2.Non-Functional Testing:

Here we tested the Non-functional features like Compatibility, Performance

Compatibility Testing:

Here We tested this software on Various Operating System Such as Linux, windows etc…

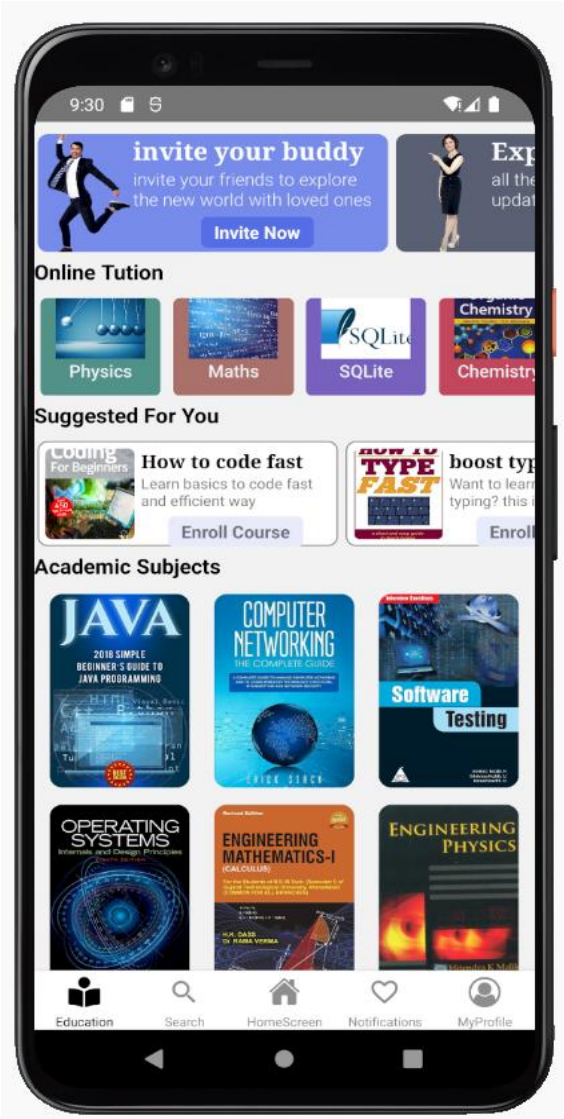
Performance Testing

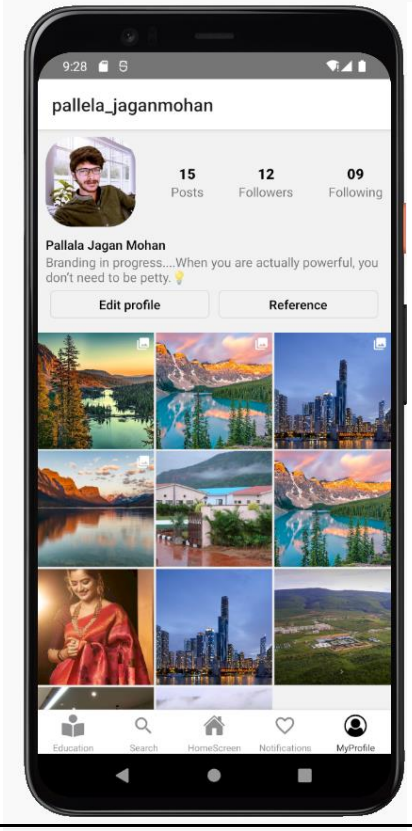
Here we tested the speed, efficiency. The software is given accurate results when the user enters the data.

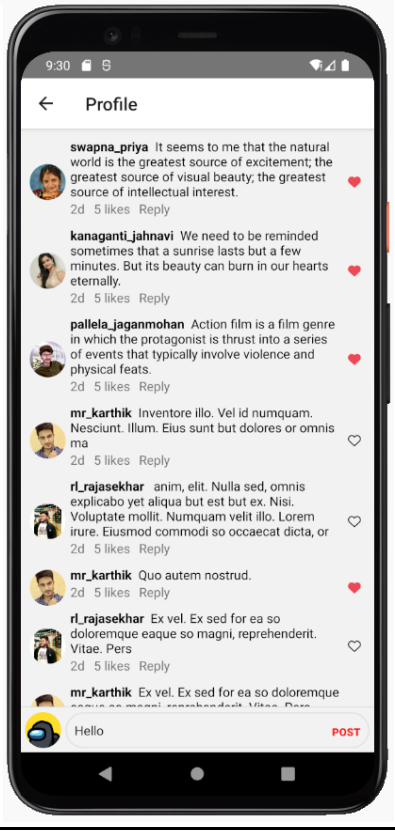
Future Improvements

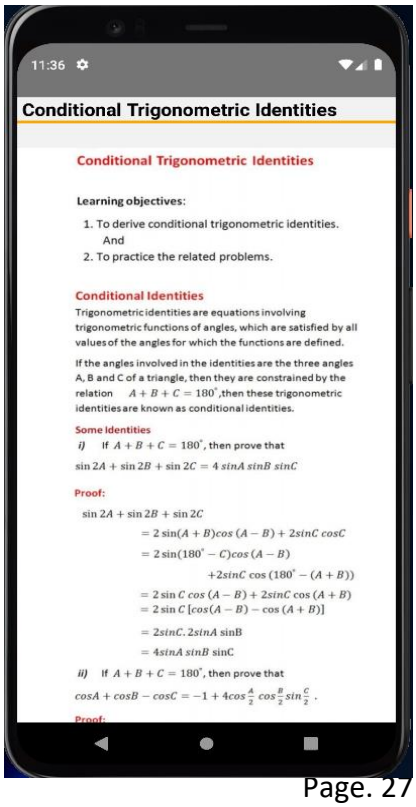
- Implementing artificial intelligence (AI) and machine learning (ML) algorithms to provide users with personalized course and feed recommendations based on their search history and preferences. This would improve the user experience by offering more relevant and customized results.
- Integration with virtual reality (VR) technology to provide users with immersive 3D tours of game based study, entertainment, and gaming. This would enhance the user experience by giving users a more realistic sense of what the meteorite is like their favorite one.
- The upcoming version comes with Social communication between the same field of study like UPSC, JEE ,NEET etc... Along with that gaming environment will be introduce soon with multiplayer through their friends to increase the more refreshment and increase boost of a user. All the needs of a user will be added one by one in upcoming updates.
- Implementation of a loyalty program to reward users for their continued use of the platform. This would encourage users to study through the app more frequently and build a sense of brand loyalty. Always user privacy is our policy of Meteorite.

Module Snippets









References

- https://www.w3schools.com/
- https://awsconsole.dev/
- https://reactnative.dev/