

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC AUTONOMOUS)

(Permanently Affiliated to AU Approved by AICTE and Accredited
by NBA & NAAC with 'A' Grade)

SANGIVALASA-531162, Bheemunipatnam Mandal, Visakhapatnam
District 2020-2024



“APP DEVELOPMENT”

An Industrial Training report submitted in partial fulfilment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

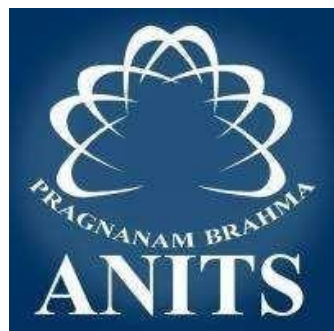
COMPUTER SCIENCE AND ENGINEERING

Submitted by

V Rama HareshKiran– 320126510185

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND
SCIENCES**

(Affiliated to Andhra University)



BONAFIDE CERTIFICATE

The Internship work entitled “APP DEVELOPMENT”, a bonafide work carried out by VALLEPU RAMA HARESHKIRAN as a part of B.TECH 4th year of Computer Science and Engineering in partial fulfilment of the requirements for the Internship.

Dr.M Ramakrishna Murthy

HEAD OF THE DEPARTMENT

Professor

Department of CSE

Ms.Venkata Ratnam

PROJECT REVIEWER

Assistant Professor

Department of CSE

Mrs.G.Gowri Pushpa

INTERNSHIP COORDINATOR

Assistant Professor

Department of CSE

COMPANY DETAILS

Bharat Intern, we believe that starting strong is the key to success. That's why we're dedicated to providing college students with the skills, experience, and knowledge they need to launch their careers and achieve their goals. Our internship programs are designed to help students start strong and finish even stronger, so they can stand out in the competitive job market and achieve their dreams.

We believe that a good start is half done. That's why we're committed to helping our interns start strong and develop the skills and experience they need to finish even stronger. Whether you're just starting your journey or looking to take the next step in your career, our internship programs can help you get there. Join us and discover the power of a strong start."

We are always open to help you to take the next step in your career!

Contact Info: bharatintern.info@gmail.com

DECLARATION

I, Bearing University Register No. 320126510185 do here by declare that this industrial training on entitled “WEB DEVELOPMENT”, is carried out from 10th SEPTEMBER 2023 to 10th OCTOBER 2023 at BHARAT INTERN.

I hereby declare that the original training and mini project is done by me as per requirements of training.

Name : Vallepu Rama HareshKiran

Reg No : 320126510185

ACKNOWLEDGEMENT

The successful completion of this internship report would not have been possible without the support and assistance of many individuals and organizations. I feel immensely blessed to have gotten this during the course of my internship program. I would like to take this opportunity to offer my earnest admiration to each and every one of them. The realization of this internship report's success is a testament to the invaluable support and assistance extended by numerous individuals and organizations.




I express my sentiment of gratitude to Bharat Intern who has been a continuous source of inspiration as my intern trainer. Without the constant guidance and suggestions, this report would have been nowhere near completion. My gratitude for his trust and generosity goes beyond words. I am indebted and thankful to my learned and revered Prof. Dr.M Ramakrishna Murthy, Head of Department of Information Technology, Anil Neerukonda Institute of Technology and Sciences, Sangivalasa, Visakhapatnam for his encouraging support and providing a meticulous platform to learn.

I extend my appreciation to my classmates whose direct and indirect contributions facilitated the accomplishment of my work. Their camaraderie and assistance were invaluable, and I am grateful for their support.

Finally, my sincere thanks go to all my teachers who equipped us for this endeavor. Your guidance and teachings have been the bedrock of my success, and I owe a debt of gratitude to each one of you.

In conclusion, the successful completion of this internship report stands as a collective achievement, made possible by the support and inspiration drawn from these exceptional individuals and institutions.

OFFER LETTER

 <https://bharatintern.live>
 bharatintern.info@gmail.com
 [Bharat Intern](#)



WELCOME LETTER

To:

VALLEPU RAMA HARESHKIRAN

Date:

11 September 2023

Congratulations on securing a virtual internship opportunity at Bharat Intern as an **"App Developer Intern"** commencing on September 10, 2023. We are excited to welcome you aboard!

At Bharat Intern, we view internships as more than just jobs, they are invaluable educational experiences for students to acquire and refine skills. Throughout this internship, you will engage in hands-on learning, applying theoretical knowledge to real-world scenarios.

Prepare to explore a plethora of opportunities to showcase and enhance your abilities, gaining profound insights into the industry's inner workings.

By accepting this offer, you commit to executing assigned tasks diligently and ensuring excellence in all aspects of your work. We firmly believe that your time with us will be enriching, equipping you with the expertise to excel in your future endeavours. We are thrilled to have you as part of our team, ready to embark on a rewarding journey together.

Warm regards,
Bharat Intern



TABLE OF CONTENTS

S.No	Topic	Page No
1	Introduction	8
2	Software	9
3	Hardware Requirements	10
4	Languages Used	11
5	Tasks Assigned:MovieRulz	15
6	Movierulz Project Completion Report	16
7	Project Code	17
8	Certificate	56

INTRODUCTION

An internship in App development offers an invaluable opportunity for individuals to gain hands-on experience and practical skills in the dynamic and rapidly evolving field of App development. App development is a crucial aspect of the technology industry, encompassing the design, creation, and maintenance of Applications.

During an app development internship, participants have the opportunity to closely collaborate with experienced professionals, gaining insights into various technologies, programming languages, and frameworks. This immersive learning experience occurs in a dynamic environment, enabling interns to contribute to real-world projects and build a robust portfolio.

Interns typically receive exposure to both front-end and back-end app development. They learn to craft visually appealing and user-friendly interfaces using tools like React Native for app development. Simultaneously, they delve into server-side logic using frameworks such as Flask for back-end development, ensuring the functionality of the movie streaming and download app named "Movierulz."

Languages like JavaScript, HTML, CSS, and Python become integral to their skill set, empowering them to navigate both client and server aspects of app creation. This holistic approach extends to the development of critical soft skills such as problem-solving, effective communication, and collaborative teamwork.

Throughout the internship, participants engage in cross-functional collaboration with designers, project managers, and quality assurance professionals. This interaction provides a comprehensive understanding of the app development process, mirroring the complexity of real-world projects.

Ultimately, the app development internship serves as a pivotal step for individuals aspiring to launch their careers in the tech industry. It not only offers hands-on experience but also facilitates networking with industry experts, instills knowledge of best practices, and keeps interns abreast of the latest trends and technologies in app development.

Software Requirements

VSCode (Visual Studio Code)

- **Description:** VSCode is a popular, free, and open-source code editor that provides features for debugging, syntax highlighting, intelligent code completion, and version control integration.
- **Purpose:** It is chosen as the preferred text editor for its ease of use, extensibility through plugins, and support for a wide range of programming languages commonly used in web development.

ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for Android app development. Based on the powerful code editor and developer tools from [IntelliJ IDEA](#), Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator

Hardware Requirements

RAM (Random Access Memory)

- **Minimum Requirement:** 8GB of RAM.
- **Explanation:** RAM is essential for the smooth operation of applications. With a minimum of 12GB RAM, the system should have enough memory to run the web development tools and processes efficiently. This is particularly important when working with modern development stacks that may include multiple services and tools running simultaneously.

Hard Disk Space

- **Minimum Requirement:** At least 100 MB of free space.
- **Explanation:** Hard disk space is necessary for storing the application code, dependencies, and other related files. The estimation of 100 MB is given based on the expected size of the project. This requirement ensures that there is sufficient space for the project files, any libraries or frameworks used, and potential build artifacts without causing storage constraints.

Languages Used

REACT NATIVE(Bare React Native):

- **Definition:** A bare React Native app is a project where developers make direct changes to their native android and ios project directories rather than continuously generating them on demand using app config and prebuild.
- **Purpose:** React Native (also known as RN) is a popular JavaScript-based mobile app framework that allows you to build natively-rendered mobile apps for iOS and Android. The framework lets you create an application for various platforms by using the same codebase.

Native Api's

Heading

```
const Heading = () => (
  <Text style={{ fontSize: 24, fontWeight: 'bold' }}>This is a
  Heading</Text>
);
```

Paragraph

```
const Paragraph = () => (
  <Text>This is a paragraph of text.</Text>
);
```

List

```
const List = () => (
  <View>
    <Text>Item 1</Text>
    <Text>Item 2</Text>
    <Text>Item 3</Text>
  </View>
```

```
);
```

Link

```
const Link = () => (
  <TouchableOpacity onPress={() => alert('Link pressed')}>
    <Text style={{ color: 'blue' }}>Visit Example.com</Text>
  </TouchableOpacity>
);
```

Image

```
const ImageComponent = () => (
  <Image source={require('./image.jpg')} style={{ width: 100, height:
100 }} />
);
```

Form

```
const Form = () => (
  <View>
    <TextInput placeholder="Username" />
  </View>
);
```

Usage:

```
const App = () => (
  <View>
    <Heading />
    <Paragraph />
    <List />
```

```
<Link />  
<ImageComponent uri={{src:'https://example.com'}} />  
<Form />  
</View>  
);
```

CSS (Cascading Style Sheets):

- **Definition:** CSS is a style sheet language used for describing the presentation of a document written in HTML. It controls the layout, formatting, and appearance of the elements on a web page.
- **Purpose:** CSS allows developers to style HTML elements, specifying aspects like colors, fonts, spacing, and positioning.
It separates the content from its presentation, promoting a cleaner and more maintainable code structure.

CSS STYLES

Styling a Heading:

```
h1 {  
  color: blue;  
  font-size: 24px;  
}
```

Styling a Paragraph:

```
p {  
  font-family: Arial, sans-serif;  
  line-height: 1.5;  
}
```

Styling a List:

```
ul {  
  list-style-type: square;  
  margin-left: 20px;  
}
```

Styling a Link:

```
a {
    text-decoration: none;
    color: #0066cc;
}
```

Styling an Image:

```
img {
    width: 100%; }
```

JavaScript:

- **Definition:** JavaScript is a high-level, interpreted programming language that enables interactive and dynamic behavior on the client side of a web page. It is often used to enhance user experience and add interactivity.
- **Purpose:** JavaScript allows developers to manipulate the content of a web page in real-time, responding to user actions and events. It can be used for tasks such as form validation, AJAX requests, and creating interactive features like sliders and pop-ups.

Modules:

react-native-async-storage/async-storage:

```
_storeData = async () => {
    try {
        await AsyncStorage.setItem(
            '@MySuperStore:key',
            'I like to save it.',
        );
    } catch (error) {
        // Error saving data
    }
};
```

Axios:**GET request:**

```
axios({method:"get",url:
"https://jsonplaceholder.typicode.com/todos", params: { _limit: 5, }
});
```

POST request:

```
axios({ method: "post", url: "/login", data: { firstName: "Finn",
lastName: "Williams" } });
```

Tasks Assigned

Movierulz: A Free Movie Streaming Platform

Objective:

Develop a web and mobile application that allows users to watch movies for free. The application should also enable users to download movies, add movies to their watchlist, and view movie details such as cast and crew, available languages, and additional information.

Functional Requirements:

- Movie Streaming: Users should be able to stream movies for free without any subscription fees.
- Movie Download: Users should have the option to download movies for offline viewing.
- Watchlist: Users should be able to add movies to their watchlist for convenient access later.
- Movie Details: Provide detailed information about each movie, including cast and crew, available languages, plot summary, genre, release date, and other relevant details.
- Search Functionality: Implement a search function to enable users to easily find the movies they are looking for.
- Content Management: Develop a content management system (CMS) to allow authorized users to add, remove, and manage movie content.
- User Management: Implement a user management system to handle user registration, authentication, and profile management.

Non-Functional Requirements:

- Performance: The application should be responsive and efficient, ensuring smooth streaming and downloading experiences.
- Cross-Platform Compatibility: Develop the application to be compatible with various web browsers and mobile devices.

- Security: Implement robust security measures to protect user data and prevent unauthorized access.
- Scalability: Design the application to be scalable to accommodate a growing user base and movie library.
- User Interface: Create a user-friendly interface that is intuitive and easy to navigate.

Additional Considerations:

- Explore monetization options without compromising the free movie streaming experience.
- Integrate with social media platforms to enhance user engagement and sharing.
- Consider implementing a recommendation system to suggest movies based on user preferences.
- Provide parental controls to restrict access to inappropriate content.
- Stay up-to-date with the latest copyright laws and regulations.

Target Audience:

The Movierulz application targets a broad audience of movie enthusiasts who seek a free and convenient way to watch and enjoy their favorite movies. The application caters to individuals of all ages and backgrounds, with a particular focus on providing a seamless and engaging experience for users in India.

Movierulz Project Completion Report

Project Overview:

The Movierulz project was initiated with the goal of developing a web and mobile application that allows users to watch movies for free. The application was designed to be user-friendly and accessible to a wide audience, with a particular focus on providing a seamless and engaging experience for users in India.

Functional Requirements:

All functional requirements outlined in the task statement were successfully implemented. Users can stream movies for free, download movies for offline viewing, add movies to their watchlist, and view detailed information about each movie. The application also features a search function to help users find the movies they are looking for.

Non-Functional Requirements:

The application was developed with a focus on performance, cross-platform compatibility, security, scalability, and user interface. The application is responsive and efficient, ensuring smooth streaming and downloading experiences. It is compatible with various web browsers and mobile devices. Robust security measures are implemented to protect user data and prevent unauthorized access. The application is scalable to accommodate a growing user base and movie library. The user-friendly interface is intuitive and easy to navigate.

Additional Considerations:

- **Monetization:** The application currently relies on advertising revenue as its primary source of monetization. However, the team is exploring other monetization options without compromising the free movie streaming experience.
- **Social Media Integration:** The application is currently not integrated with any social media platforms. However, the team is considering adding social media integration features to enhance user engagement and sharing.
- **Recommendation System:** The application does not currently include a recommendation system. However, the team is considering implementing a recommendation system to suggest movies based on user preferences.
- **Parental Controls:** The application currently does not include parental controls. However, the team is considering adding parental controls to restrict access to inappropriate content.
- **Copyright Compliance:** The team is committed to staying up-to-date with the latest copyright laws and regulations. The application adheres to all applicable copyright laws and regulations.

Project Conclusion:

The Movierulz project was successfully completed on schedule and within budget. The application meets all project requirements and provides a valuable service to its users. The team is committed to ongoing development and improvement of the application.

Recommendations

- Continue to explore monetization options without compromising the free movie streaming experience.
- Implement social media integration features to enhance user engagement and sharing.
- Add a recommendation system to suggest movies based on user preferences.
- Implement parental controls to restrict access to inappropriate content.
- Continuously monitor and adhere to copyright laws and regulations.

Project Code:

App.js:

```
import 'react-native-gesture-handler';

import React, { useEffect } from 'react';
import { View, Text, StatusBar, Platform } from 'react-native';
import { NavigationContainer, useNavigation } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import { TextInput } from 'react-native-paper';
import TopBar from './MovieNavigation/TopBar';
import Search from './Screens/Search';
```

```

import TextInputBar from './componets/TextInputBar';
import Movie from './Screens/Movie';
import MovierulzScreen from './Screens/MovierulzScreen';
import Web from './Screens/Web';
import Test from './Screens/Test';
import { RefreshProvider } from './componets/MovieContext';
const Stack = createStackNavigator();

export default function App() {
  useEffect(()=>{
    console.log(Platform.isTV)
  },[])

  function Main(){
    return (
      <NavigationContainer>
        <StatusBar hidden={true} backgroundColor={"rgba(6, 2, 38,1)"} translucent={true}
      />
      /* <View style={{backgroundColor:'rgba(6, 2, 38,1)'}} ><Text
style={{textAlign:'center',width:200,color:"white",marginLeft:"25%",marginTop:3,fontSize:25,backgroundColor:"rgba(225,225,225,.5)",borderRadius:20}} >MovieRulz</Text></View>
*/
      <Stack.Navigator screenOptions={{ header: () => null }} initialRouteName='intro'
    >
      <Stack.Screen name='intro' component={MovierulzScreen} />
      <Stack.Screen name="topbar" component={TopBar} />
      <Stack.Screen name="movie" component={Movie} />
      <Stack.Screen name='web' component={Web} />
      <Stack.Screen name='test' component={Test} />

    </Stack.Navigator>
  </NavigationContainer>
  )
}
return (
  <RefreshProvider>
    <Main />
  </RefreshProvider>
);
}

```

MovierulzScreen.js:

```

import { View, Text, Image, Animated, ToastAndroid, Dimensions } from 'react-native';
import React, { useEffect } from 'react';

```

```

export default function MovierulzScreen({ navigation }) {
  const value = new Animated.Value(0);
  const scale = new Animated.Value(0);
  const textScale = new Animated.Value(0);
  const textOpacity = new Animated.Value(0);

  const rotation = () => {
    Animated.timing(value, {
      toValue: 1,
      duration: 1500,
      useNativeDriver: false,
    }).start();

    Animated.timing(textOpacity, {
      toValue: 1,
      delay: 1500,
      duration: 1500,
      useNativeDriver: false,
    }).start();

    Animated.sequence([
      Animated.timing(scale, {
        toValue: 1,
        duration: 1500,
        useNativeDriver: false,
      }),
      Animated.timing(textScale, {
        toValue: 1,
        duration: 1500,
        useNativeDriver: false,
      }),
    ]).start();
  };

  useEffect(() => {
    rotation();
    setTimeout(() => {
      navigation.replace('topbar');
    }, 3500);

    setTimeout(() => {
      ToastAndroid.show('Developed By Hari Kiran', 1500);
    }, 3000);

    console.log(Dimensions.get('screen').width, Dimensions.get('screen').height);
  }, []);

  const rotationValue = value.interpolate({
    inputRange: [0, 1],
    outputRange: ['0deg', '360deg'],
  });

```

```

});

const scaleValue = scale.interpolate({
  inputRange: [0, 1],
  outputRange: [3, 1],
});

const textScaleValue = textScale.interpolate({
  inputRange: [0, 1],
  outputRange: [100, 1],
});

const textOpacityValue = textOpacity.interpolate({
  inputRange: [0, 1],
  outputRange: [0, 1],
});

const deviceWidth = Dimensions.get('screen').width;
const imageWidth = deviceWidth * 0.8;
const imageHeight = imageWidth;

return (
  <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center',
backgroundColor: 'rgba(6, 2, 38,1)' }}>
    <Animated.View style={{ transform: [{ rotate: rotationValue }, { scale:
scaleValue }] }}>
      <Image source={require('./icon.jpg')} style={{ width: imageWidth, height:
imageHeight, borderRadius: imageWidth / 2 }} />
    </Animated.View>
    <Animated.Text style={{ color: 'white', fontSize: 40, fontWeight: "700",
letterSpacing: 5, transform: [{ scale: textScaleValue }], opacity: textOpacityValue
}}>MOVIE RULZ</Animated.Text>
  </View>
)
}

```

TopBar.js:

```

import { View, Text, StatusBar, Image } from 'react-native'
import React from 'react'
import { NavigationContainer } from '@react-navigation/native'
import { createMaterialTopTabNavigator } from '@react-navigation/material-top-tabs'
import Home from '../Screens/Home'
import Telugu from '../Screens/Telugu'
import Hindi from '../Screens/Hindi'
import Tamil from '../Screens/Tamil'
import Malayalam from '../Screens/Malayalam'

```

```

import English from '../Screens/English'
import AntDesign from 'react-native-vector-icons/AntDesign'
import Search from '../Screens/Search'
import Test from '../Screens/Test'
import WishList from '../Screens/WishList'

export default function TopBar() {
  const Tab = createMaterialTopTabNavigator()

  return (
    <Tab.Navigator
      initialRouteName='Home'
      tabBarPosition='bottom'
      screenOptions={({ route }) => ({
        tabBarIcon: ({ focused, size, color }) => {
          let icon;
          if (route.name === 'Home') {
            icon = 'home'
            color = focused ? 'blue' : 'white'
            size = focused ? 25 : 25
          }
          else if (route.name === 'Tel') {
            return (
              <Text style={{ color: focused ?
'blue': 'white', fontSize: 20, marginTop: 1, zIndex: 1, transform: [{ scale: focused ? 2 : 1 }] }}
>☎</Text>
            )
          }
          else if (route.name === 'Hin') {
            return (
              <Text style={{ color: focused ?
'blue': 'white', fontSize: 20, marginTop: 1, zIndex: 1, transform: [{ scale: focused ? 2 : 1 }] }}
>हि</Text>
            )
          }
          else if (route.name === 'Tam') {
            return (
              <Text style={{ color: focused ?
'blue': 'white', fontSize: 20, marginTop: 1, zIndex: 1, transform: [{ scale: focused ? 2 : 1 }] }}
>த</Text>
            )
          }
          else if (route.name === 'Mala') {
            return (
              <Text style={{ color: focused ?
'blue': 'white', fontSize: 20, marginTop: 1, zIndex: 1, transform: [{ scale: focused ? 2 : 1 }] }}
>ம</Text>
            )
          }
          else if (route.name === 'Eng') {
            return (

```

```

        <Text style={{color:focused ?
'blue':'white'},fontSize:20,marginTop:1,zIndex:1,transform:[{scale:focused ? 2 :1}]}
>E</Text>

    )
  }
  else if(route.name==='Search'){
    icon='search1'
    color= focused ? 'lightblue' : 'white'
    size=focused ? 25 : 25
  }
  else if(route.name==='Wish'){
    icon=focused?'clockcircle':'clockcircleo'
    color= focused ? 'red' : 'red'
    size=focused ? 22 : 22
  }
  return (
    <AntDesign
name={icon} style={{color:color,fontSize:size,transform:[{scale:focused ? 1.5
:1}]} />
    )
  }
  ,
  tabBarStyle: {
    backgroundColor: 'rgba(6, 2, 38,.9)',
    opacity: 1,
    // borderRadius: 10,
    borderTopLeftRadius:30,
    borderTopRightRadius:30,
    width: "100%",
    height: 60,
    position: 'absolute',
    bottom: "-0.5%",
    left: "0%"
  },
  tabBarShowLabel:false,
  tabBarActiveTintColor: 'green', // Change the active tab color here
)}}
keyboardDismissMode={'on-drag'}
>
  <Tab.Screen name='Home' component={Home}
options={{tabBarActiveTintColor:'rgba(225,0,0,.3)}} />
  <Tab.Screen name='Tel' component={Telugu} />
  <Tab.Screen name='Hin' component={Hindi} />
  <Tab.Screen name='Tam' component={Tamil} />
  <Tab.Screen name='Mala' component={Malayalam} />
  <Tab.Screen name='Eng' component={English} />
  <Tab.Screen name='Search' component={Search} />
  { /* <Tab.Screen name='Test' component={Test} /> */ }
  <Tab.Screen name="Wish" component={WishList} /></Tab.Navigator>
)}}

```

Movie.js:

```
import { View,
Text,FlatList,Image,StyleSheet,Pressable,Linking,TouchableOpacity,ToastAndroid,ImageBack
ground } from 'react-native'
import React,{useContext, useEffect,useState} from 'react'
import Video from 'react-native-video'
import axios from 'axios'
import WebView from 'react-native-webview'
import Player from './Player'
import AsyncStorage from '@react-native-async-storage/async-storage'
// import { useRefresh,RefreshContext } from '../componets/MovieContext'
import { useRefresh } from '../componets/MovieContext';
export default function Movie({route,navigation}) {
  const { forRefresh, setRefresh } = useRefresh();
  const data=route.params
  const [fetchData,setFecthdata]=useState(null)
  const [type,setType]=useState('Unknown')
  const [Language,setLanguage]=useState('Unknown')

  const [image,setImage]=useState('https://images.unsplash.com/photo-1495344517868-
8ebaf0a2044a?ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3J1LWZ1ZWR8MXx8fGVuZDB8fHx8fA%3D%3D&w=1000&q=80')
  useEffect(()=>{
    fetch()

  },[])

  function lan(title){
    try{
      if (title.includes('[')){
        setLanguage(title.slice(title.indexOf('[')+1,title.indexOf(']')))
      }
      else if(title.includes('Telugu')){
        setLanguage('Telugu')
      }
      else if(title.includes('Malayalam')){
        setLanguage('Malayalam')
      }
      else if(title.includes('Hindi')){
        setLanguage('Hindi')
      }
      else if(title.includes('Tamil')){
        setLanguage('Tamil')
      }
      else if(title.includes('Kannada')){
        setLanguage('Kannada')
      }
      else if(! title.includes('English')){
        setLanguage('English')
      }
    }
  }
}
```

```

    }
    catch{
        setLanguage('UnKnown')
    }
}
async function fetch(){
    try{
        await axios.get(`https://movierulzapi-
4njb.vercel.app/get?url=${data.title}`).then(res=>{
            console.log(res.data.title)
            setFecthdata(Array(res.data))
            setImage(res.data.image)
            try{
                if(res.data.title.includes('Season')){
                    setType(res.data.title.slice(res.data.title.indexOf('Season'),res.d
ata.title.indexOf('Season')+9))
                    setLanguage(res.data.title.slice(res.data.title.indexOf('[')+1,res.
data.title.indexOf(']')))
                }

                else if(res.data.title.includes("DVDScr")){
                    setType('DVDScr')
                    lan(res.data.title)
                }
                else if(res.data.title.includes("HDRip")){
                    setType('HDRip')
                    lan(res.data.title)
                }
                else if(res.data.title.includes("HQ Line")){
                    setType('HQ Line')
                    lan(res.data.title)
                }
                else if(res.data.title.includes("Song")){
                    setType('Song')
                    lan()
                }
            }
        }
    }
    catch{
        setType('unknown')
    }
    // try{
    //     if(res.data.title.includes('Telugu')){
    //         setLanguage('Telugu')
    //     }
    //     else if(res.data.title.includes('Malayalam')){
    //         setLanguage('Malayalam')
    //     }
    // }
    // catch{
    //     setLanguage('UnKnown')
    // }
}

```



```

    }).catch(err=>{
        console.log("movie doesnot exists")
        ToastAndroid.show('There is error with Movie ',1500)
        navigation.pop(1)
    })
}
catch{
    console.log("fetching failed")
}
}
const [wishTitle,setWishTitle]=useState(false)
const [wishurl,setWishurl]=useState(null)
async function wishR(){
    try{
        await AsyncStorage.getItem("movierulzWhish",async(err,result)=>{
            if(result!=null){
                let s=0
                let retriiveData=JSON.parse(result)
                retriiveData.data.forEach(element => {
                    if(data.title==element.link){
                        setWishTitle(false)
                        s=1
                    }
                });
                if(s==0){
                    setWishTitle(true)
                }
            }
            else{
                await
AsyncStorage.setItem("movierulzWhish",JSON.stringify({data:[],status:200}))
            }
        })
    }catch{
        console.log("e")
    }
}

async function unWish(url){
    await AsyncStorage.getItem('movierulzWhish',async(err,res)=>{
        if(res!=null){
            let retriiveData=JSON.parse(res)
            let newData={data:[],status:200}
            retriiveData.data.forEach(async element => {
                if(element.link!=url){
                    console.log(element)
                    newData.data.push(element)
                }
            })
            await
AsyncStorage.setItem('movierulzWhish',JSON.stringify(newData),(err)=>{
                if(!err){

```

```

        setRefresh(forRefresh+1)
    }
    })
  });
}
})
}
return (
  <View style={styles.container} >
    <ImageBackground source={{uri: image}}
style={[{backgroundColor: 'rgba(0,0,0,.4)'}]} blurRadius={.8} >
      /* {console.log(fetchData[0].image)} */
    <View style={[styles.frame,{opacity:1,backgroundColor: 'rgba(0,0,0,.4)'}]} >

      <FlatList data={fetchData} renderItem={({item})=>{
        // setWishurl(item.url)
        wishR()
        return (
          <View>
            <Text style={[styles.text,{marginTop:50}]} >{
              item.title.includes("DVDScr") || item.title.includes("HDRip") ?
              item.title.slice(0,item.title.indexOf('(')) :
              item.title.includes("Season") ?
item.title.slice(0,item.title.indexOf('(')) +
item.title.slice(item.title.indexOf('[')+1,item.title.indexOf(']'))
              : item.title
            }
          </Text>
          <Image source={{uri:item.image}}
style={[styles.image,{width:200,height:300,borderRadius:10},]} />
          {wishTitle?<TouchableOpacity
style={[styles.button,{width:150,marginLeft:120}]}
            onPress={async ()=>{
              let
wishData={link:item.url,image:item.image,title:item.title}
              let retrieveData
              try{
                await
AsyncStorage.getItem("movierulzWhish",async(err,result)=>{
                  if(result!=null){
                    retrieveData=JSON.parse(result)
                  }
                  else{
                    await
AsyncStorage.setItem("movierulzWhish",JSON.stringify({data:[],status:200}))
                  }
                })
              }catch{
                console.log("e")
              }
              retrieveData.data.push(wishData)
            }
          }
        )
      }
    }
  )
)

```

```

        await
AsyncStorage.setItem("movierulzWhish",JSON.stringify(retrieveData),(err)=>{if(!err){
        setWishTitle(false)
        setRefresh(forRefresh+1)
        }})
    })
    <Text style={{color:'white',fontWeight:'bold'}} >Add to
Watch</Text></TouchableOpacity>:
    <TouchableOpacity
style=[[styles.button,{width:150,marginLeft:120}]] onPress={()=>{unWish(item.url)}} >
    <Text style={{color:'white',fontWeight:'bold'}} >Un-
Watch</Text>
    </TouchableOpacity>
    }
    {/* <Video source={{uri:'blob:https://streamwish.to/70d2737e-5d41-
4417-b970-a2bac7c40e5c'}} style={{height:200,width:400}} /> */}

    {/* <Text>Link:{item.torrent.length}</Text> */}
    <Text
style={{color:'white',fontStyle:'italic',padding:5,textAlign:'center'}} ><Text
style={{fontWeight:"bold"}} >Type:</Text>{type}</Text>
    {/* <Text
style={{color:'white',fontStyle:'italic',textAlign:'center'}} ><Text
style={{fontWeight:"bold"}} >Starring
by:</Text>{item.cast.slice(item.cast.indexOf('Starring
by:')+12,item.cast.indexOf('Genres'))}</Text>
    <Text style={{color:'white',fontStyle:'italic',textAlign:'center'}}
><Text style={{fontWeight:"bold"}} >Directed
by:</Text>{item.cast.slice(item.cast.indexOf(':')+1,item.cast.indexOf('Written
by'))}</Text>
    <Text style={{color:'white',fontStyle:'italic',textAlign:'center'}}
><Text style={{fontWeight:"bold"}} >Written
by:</Text>{item.cast.slice(item.cast.indexOf('Written
by:')+11,item.cast.indexOf('Starring by'))}</Text> */}
    <Text
style={{color:'white',fontStyle:'italic',padding:5,textAlign:'center',fontWeight:"bold"
}} ><Text style={{fontWeight:"bold"}}
></Text>{item.cast.slice(0,item.cast.indexOf("Country"))}</Text>

    <Text style={{color:'white',fontStyle:'italic',textAlign:'center'}}
><Text style={{fontWeight:"bold"}} >Language:</Text>{Language}</Text>

    <Text
style={{color:'white',fontStyle:'italic',padding:20,textAlign:'center'}} ><Text
style={{fontWeight:"bold"}} >Description:</Text>{item.description}</Text>

    <Text style={styles.text} >Torrent Files</Text>
    {item.torrent?
    <FlatList data={item.torrent} renderItem={({item:innerItem})=>{
    return(
    <View>

```

```

        <TouchableOpacity style={styles.button} onPress={()=>{
            try{
                Linking.canOpenURL(innerItem.magnet)?Linking.openURL(in
nerItem.magnet): ToastAndroid.show('cannot open url',1500)}
            catch{
                console.log('error')
            }
        }}>
        <Text style={{color:'white'}} >Magnet-
{innerItem.quality}{innerItem.size}</Text>
        </TouchableOpacity>
    </View>
)
}}
keyExtractor={({innerItem})=>innerItem.magnet}
/>:<Text>No Torrent Files</Text>
}

<Text style={[styles.text,{transform:[{scale:.9}]]]} >Links of Hosted
files</Text>

<View style={styles.linkview} >
    { item.other_links ?
    < FlatList
        data={item.other_links}
        renderItem={({ item: innerItem }) => (
            <View >
                { /* <Text>Type: {innerItem.type}</Text> */}
                { /* <Text>URL: {innerItem.url}</Text> */}

                {
                    innerItem.type=== " Streamtape"?
                    <Pressable onPress={() => { navigation.navigate('web',
{ url: innerItem.url }) }} style={[styles.linkButton, { backgroundColor:
'rgba(0,225,0,.4)' }]]>
                        <Text style={{ color: 'rgba(225,225,225,1)' }}>
{innerItem.type}<Text style={{ fontSize: 11, color: 'rgba(100,200,200,1)'
}}>(Recomended to watch)</Text></Text>
                        </Pressable>
                        :
                        <Pressable
onPress={()=>{navigation.navigate('web',{url:innerItem.url})}}
style={styles.linkButton} ><Text style={{color:'white'}} >
{innerItem.type}</Text></Pressable>
                    }
                </View>
            )}
            keyExtractor={({innerItem}) => innerItem.url}
        />
        : <Text>No Hosted Links</Text>
    }
</View>
</View>
)

```

```

    }}
    keyExtractor={({item})=> item.description}
  />
</View>
</ImageBackground>
</View>
)
}

const styles=StyleSheet.create({
  container:{
    flex:1,
    justifyContent:"center",
    alignItems:"center",
    backgroundColor:'rgba(6, 2, 38,1)'
  },
  frame:{
    marginTop:0,
    height:"100%",
    width:"100%"
  },
  text:{
    color:'white',
    fontSize:35,
    padding:2,
    textAlign:"center"
  },
  image:{
    margin:110,
    marginTop:50,
    marginBottom:20
  },
  button:{
    width:300,
    height:30,
    backgroundColor:'rgba(225,225,225,.1)',
    color:'white',
    alignItems:"center",
    justifyContent:"center",
    flex:1,
    margin:2,
    marginLeft:50,
    borderRadius:10,
  },
  linkButton:{
    width:300,
    height:30,
    backgroundColor:'rgba(225,225,225,.1)',
    color:'white',
    alignItems:"center",
    justifyContent:"center",

```

```

        flex:1,
        margin:2,
        marginLeft:50,
        borderRadius:10
    },
    linkview:{
        width:400,
        height:'max-content',
        marginBottom:100
    }
})

```

Home.js:

```

import { View, Text, Image, StyleSheet, ScrollView, FlatList, Pressable, RefreshControl
} from 'react-native';

import React, { useState, useEffect } from 'react';
import axios from 'axios';

export default function Home({navigation}) {
    const [data, setData] = useState([]);
    const [refreshing, setRefreshing] = React.useState(false);

    async function fetchData() {
        setRefreshing(true);
        try {
            const response = await axios.get('https://movierulz.vercel.app/');
            if (response.status === 200) {
                setData(response.data.data);
                setRefreshing(false);
            } else {
                console.log('Failed to fetch data');
                setRefreshing(false);
            }
        } catch (error) {
            console.error('Error fetching data:', error);
            setRefreshing(false);
        }
    }

    const onRefresh = React.useCallback(() => {
        setRefreshing(true);
        setTimeout(() => {
            setRefreshing(false);
        }, 2000);
    }, []);

```

```

useEffect(() => {
  fetchData();
}, []);

const renderRow = ({ item, index }) => {
  if (index % 2 === 0) {
    return (
      <View style={styles.row}>
        <View style={[styles.movie,{marginRight:"2%",marginLeft:"1%"}]}>
          <Pressable onPress={()=>{navigation.navigate('movie',{title:item.link})}} >
            <Image source={{ uri: item.image }} style={styles.image} />
            <Text style={styles.text} >{
              item.title.slice(0,item.title.indexOf('('))
              // item.title
            }
          </Text>
        </Pressable>
      </View>
      {index + 1 < data.length && (
        <View style={styles.movie]}>
          <Pressable onPress={()=>{navigation.navigate('movie',{title:data[index +
1].link})}} >
            <Image source={{ uri: data[index + 1].image }} style={styles.image} />
            <Text style={styles.text}>{data[index + 1].title.slice(0,data[index +
1].title.indexOf("("))}</Text>
          </Pressable>
        </View>
      )}
    </View>
  );
}
};

return (
  <View style={styles.view}>
    <FlatList
      data={data}
      renderItem={renderRow}
      keyExtractor={(item, index) => index.toString()}
      keyboardDismissMode='on-drag'
      // refreshing={true}
      // onRefresh={fetchData}
      refreshControl={
        <RefreshControl refreshing={refreshing} onRefresh={fetchData} />
      }
    />
  </View>
);
}

const styles = StyleSheet.create({

```

```

image: {
  width: 170,
  height: 260,
  borderRadius: 10,
},
text: {
  width: 170,
  fontSize: 20,
  // marginRight:5,
  color: 'black',
  textAlign:"center",
  color:"white",
  // borderWidth:1,
  // borderColor:"red"
},
movie: {
  flex: 1,
  height: 350,
  flexDirection: 'column',
  marginTop:10
},
row: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  paddingHorizontal:5,
  marginBottom: -30,
},
view: {
  flex: 1,
  width: '100%',
  backgroundColor: 'rgba(6, 2, 38,1)',
},
});

```

Telugu.js:

```

import { View, Text, Image, StyleSheet, ScrollView, FlatList, Pressable, RefreshControl
} from 'react-native';
import React, { useState, useEffect } from 'react';
import axios from 'axios';

export default function Home({navigation}) {
  const [data, setData] = useState([]);
  const [refreshing, setRefreshing] = React.useState(false);

  async function fetchData() {
    setRefreshing(true);
    try {
      const response = await axios.get('https://movierulz.vercel.app/');

```



```

    if (response.status === 200) {
      setData(response.data.data);
      setRefreshing(false);
    } else {
      console.log('Failed to fetch data');
      setRefreshing(false);
    }
  } catch (error) {
    console.error('Error fetching data:', error);
    setRefreshing(false);
  }
}

const onRefresh = React.useCallback(() => {
  setRefreshing(true);
  setTimeout(() => {
    setRefreshing(false);
  }, 2000);
}, []);

useEffect(() => {
  fetchData();
}, []);

const renderRow = ({ item, index }) => {
  if (index % 2 === 0) {
    return (
      <View style={styles.row}>
        <View style={[styles.movie, {marginRight: "2%", marginLeft: "1%"}]}>
          <Pressable onPress={() => {navigation.navigate('movie', {title: item.link})}} >
            <Image source={{ uri: item.image }} style={styles.image} />
            <Text style={styles.text} >{
              item.title.slice(0, item.title.indexOf('('))
              // item.title
            }
          </Text>
        </Pressable>
      </View>
      {index + 1 < data.length && (
        <View style={styles.movie}>
          <Pressable onPress={() => {navigation.navigate('movie', {title: data[index +
1].link})}} >
            <Image source={{ uri: data[index + 1].image }} style={styles.image} />
            <Text style={styles.text}>{data[index + 1].title.slice(0, data[index +
1].title.indexOf("("))}</Text>
          </Pressable>
        </View>
      )}
    </View>
  );
}
};

```

```

return (
  <View style={styles.view}>
    <FlatList
      data={data}
      renderItem={renderRow}
      keyExtractor={(item, index) => index.toString()}
      keyboardDismissMode='on-drag'
      // refreshing={true}
      // onRefresh={fetchData}
      refreshControl={
        <RefreshControl refreshing={refreshing} onRefresh={fetchData} />
      }
    />
  </View>
);
}

const styles = StyleSheet.create({
  image: {
    width: 170,
    height: 260,
    borderRadius: 10,
  },
  text: {
    width: 170,
    fontSize: 20,
    // marginRight: 5,
    color: 'black',
    textAlign: "center",
    color: "white",
    // borderWidth: 1,
    // borderColor: "red"
  },
  movie: {
    flex: 1,
    height: 350,
    flexDirection: 'column',
    marginTop: 10
  },
  row: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 5,
    marginBottom: -30,
  },
  view: {
    flex: 1,
    width: '100%',
    backgroundColor: 'rgba(6, 2, 38, 1)',
  },
});

```

```
});
```

Hindi.js:

```
import { View, Text, Image, StyleSheet, FlatList,
TouchableOpacity,Pressable,RefreshControl } from 'react-native';
import React, { useState, useEffect, useRef } from 'react';
import axios from 'axios';
import Icon from 'react-native-vector-icons/MaterialIcons'
export default function Hindi({navigation}) {
  const [data,setData]=useState([])
  const [pageno, setPageno] = useState(1);
  const [refreshing, setRefreshing] = React.useState(false);
  const flatlist=useRef(null);
  async function fetchData() {
    setRefreshing(true);
    try {
      const response = await axios.get(`https://movierulz.vercel.app/hindi/${pageno}`);
      if (response.status === 200) {
        setData(response.data.data)
        setRefreshing(false);
      } else {
        console.log('Failed to fetch data');
        setRefreshing(false);
      }
    } catch (error) {
      console.error('Error fetching data:', error);
      setRefreshing(false);
    }
  }
  function h(index) {
    if (index === data.length - 2) {
      if (pageno === 1) {
        return (
          <View style={{ position: 'absolute', bottom: 80,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={[styles.togglebutton,]} >
              /* <Text
style={[{color:'white',fontSize:20,textAlign:'center'}}>back</Text> */
              <Icon name='arrow-back-ios-new'
style={[{color:'white',fontSize:20,textAlign:'center'}} />
            </TouchableOpacity>
          </View>
        );
      } else {
        return (
          <View style={{ position: 'absolute', bottom: 70,width:'100%',left:5 }}>
```

```

        <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style=[{styles.tooglebutton,{backgroundColor:'rgba(225, 225, 225,.3)'}}]] >
        {/* <Text
style=[[{color:'white',fontSize:20,textAlign:'center'}]]>prev</Text> */}
        <Icon name='arrow-back-ios-new'
style=[[{color:'white',fontSize:20,textAlign:'center'}]] />
        </TouchableOpacity>
        <TouchableOpacity onPress={() => {setPageno(pageno -
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style=[{styles.tooglebutton,{position:'absolute',bottom:0,right:15}}]] >
        {/* <Text
style=[[{color:'white',fontSize:20,textAlign:'center'}]]>New</Text> */}
        <Icon name='navigate-next'
style=[[{color:'white',fontSize:20,textAlign:'center'}]] />
        </TouchableOpacity>
    </View>
    );
}
} else {
    return null;
}
}

useEffect(() => {
    fetchData();
}, [pageno]);

const renderRow = ({ item, index }) => {
    if (index % 2 === 0) {
        return (
            <View>
            <View style={styles.row}>
                <View style=[{styles.movie, { marginRight:"2%",marginLeft:"1%" }}]>
                    <Pressable onPress={()=>{navigation.navigate('movie',{title:item.link})}} >
                        <Image source={{ uri: item.image }} style={styles.image} />
                        <Text style={styles.text}>{item.title.slice(0,
item.title.indexOf("("))}</Text>
                    </Pressable>
                </View>
                {index + 1 < data.length && (
                    <View style={styles.movie}>
                        <Pressable onPress={()=>{navigation.navigate('movie',{title:data[index +
1].link})}} >
                            <Image source={{ uri: data[index + 1].image }} style={styles.image} />
                            <Text style={styles.text}>{data[index + 1].title.slice(0, data[index +
1].title.indexOf("("))}</Text>
                        </Pressable>
                    </View>
                )}
            )}

```

```

        </View>
        {index}>data.length-2? <View style={{width:"100%",height:100}} ></View> : null}
        {h(index)}
      </View>
    );
  }
};

return (
  <View style={styles.view}>
    <FlatList
      data={data}
      renderItem={renderRow}
      keyExtractor={(item, index) => item.image}

      ref={flatlist}
      refreshControl={
        <RefreshControl refreshing={refreshing} onRefresh={fetchData} />
      }

    />
  </View>
);
}

const styles = StyleSheet.create({
  image: {
    width: 170,
    height: 260,
    borderRadius: 10,
  },
  text: {
    width: 170,
    fontSize: 20,
    color: 'black',
    textAlign: "center",
    color: "white",
  },
  movie: {
    flex: 1,
    height: 350,
    flexDirection: 'column',
    marginTop: 10,
  },
  row: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 5,
    marginBottom: -30,
  },
});

```

```

view: {
  flex: 1,
  width: '100%',
  backgroundColor: 'rgba(6, 2, 38,1)',
},
togglebutton: {
  backgroundColor:'rgba(225, 225, 225,.3)',
  padding: 7,
  borderRadius:40,
  width:40,
  height:40
},
pagination: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  paddingHorizontal: 10,
},
});

```

Tamil.js:

```

import { View, Text, Image, StyleSheet, FlatList,
TouchableOpacity,Pressable,RefreshControl } from 'react-native';
import React, { useState, useEffect, useRef } from 'react';
import axios from 'axios';
import Icon from 'react-native-vector-icons/MaterialIcons'
export default function Tamil({navigation}) {
  const [data,setData]=useState([])
  const [pageno, setPageno] = useState(1);
  const [refreshing, setRefreshing] = React.useState(false);
  const flatlist=useRef(null);
  async function fetchData() {
    setRefreshing(true);
    try {
      const response = await axios.get(`https://movierulz.vercel.app/tamil/${pageno}`);
      if (response.status === 200) {
        setData(response.data.data)
        setRefreshing(false);
      } else {
        console.log('Failed to fetch data');
        setRefreshing(false);
      }
    } catch (error) {
      console.error('Error fetching data:', error);
      setRefreshing(false);
    }
  }
  function h(index) {

```

```

    if (index === data.length - 2) {
      if (pageno === 1) {
        return (
          <View style={{ position: 'absolute', bottom: 80,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={styles.tooglebutton,}] >
              /* <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>back</Text> */
              <Icon name='arrow-back-ios-new'
style={[[{color:'white',fontSize:20,textAlign:'center'}]]} />
            </TouchableOpacity>
          </View>
        );
      } else {
        return (
          <View style={{ position: 'absolute', bottom: 70,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={styles.tooglebutton,{backgroundColor:'rgba(225, 225, 225,.3)'}]} >
              /* <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>prev</Text> */
              <Icon name='arrow-back-ios-new'
style={[[{color:'white',fontSize:20,textAlign:'center'}]]} />
            </TouchableOpacity>
            <TouchableOpacity onPress={() => {setPageno(pageno -
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={styles.tooglebutton,{position:'absolute',bottom:0,right:15}}] >
              /* <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>New</Text> */
              <Icon name='navigate-next'
style={[[{color:'white',fontSize:20,textAlign:'center'}]]} />
            </TouchableOpacity>
          </View>
        );
      }
    } else {
      return null;
    }
  }
}

useEffect(() => {
  fetchData();
}, [pageno]);

const renderRow = ({ item, index }) => {
  if (index % 2 === 0) {
    return (
      <View>
        <View style={styles.row}>

```

```

        <View style={[styles.movie, { marginRight:"2%",marginLeft:"1%" }]}>
        <Pressable onPress={()=>{navigation.navigate('movie',{title:item.link})}} >
            <Image source={{ uri: item.image }} style={styles.image} />
            <Text style={styles.text}>{item.title.slice(0,
item.title.indexOf("("))}</Text>
        </Pressable>
    </View>
    {index + 1 < data.length && (
        <View style={styles.movie}>
            <Pressable onPress={()=>{navigation.navigate('movie',{title:data[index +
1].link})}} >
                <Image source={{ uri: data[index + 1].image }} style={styles.image} />
                <Text style={styles.text}>{data[index + 1].title.slice(0, data[index +
1].title.indexOf("("))}</Text>
            </Pressable>
        </View>
    )}

    </View>
    {index>=data.length-2? <View style={{width:"100%",height:100}} ></View> : null}
    {h(index)}
    </View>
);
}
};

return (
    <View style={styles.view}>
        <FlatList
            data={data}
            renderItem={renderRow}
            keyExtractor={(item, index) => item.image}

            ref={flatlist}
            refreshControl={
                <RefreshControl refreshing={refreshing} onRefresh={fetchData} />
            }

        />
    </View>
);
}

const styles = StyleSheet.create({
    image: {
        width: 170,
        height: 260,
        borderRadius: 10,
    },
    text: {
        width: 170,

```



```

    fontSize: 20,
    color: 'black',
    textAlign: "center",
    color: "white",
  },
  movie: {
    flex: 1,
    height: 350,
    flexDirection: 'column',
    marginTop: 10,
  },
  row: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 5,
    marginBottom: -30,
  },
  view: {
    flex: 1,
    width: '100%',
    backgroundColor: 'rgba(6, 2, 38,1)',
  },
  tooglebutton: {
    backgroundColor:'rgba(225, 225, 225,.3)',
    padding: 7,
    borderRadius:40,
    width:40,
    height:40
  },
  pagination: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 10,
  },
});

```

Malayalam.js:

```

import { View, Text, Image, StyleSheet, FlatList,
TouchableOpacity,Pressable,RefreshControl } from 'react-native';
import React, { useState, useEffect, useRef } from 'react';
import axios from 'axios';
import Icon from 'react-native-vector-icons/MaterialIcons'
export default function Malayalam({navigation}) {
  const [data,setData]=useState([])
  const [pageno, setPageno] = useState(1);
  const [refreshing, setRefreshing] = React.useState(false);
  const flatlist=useRef(null);
  async function fetchData() {
    setRefreshing(true);

```

```

    try {
      const response = await
axios.get(`https://movierulz.vercel.app/malayalam/${pageno}`);
      if (response.status === 200) {
        setData(response.data.data)
        setRefreshing(false);
      } else {
        console.log('Failed to fetch data');
        setRefreshing(false);
      }
    } catch (error) {
      console.error('Error fetching data:', error);
      setRefreshing(false);
    }
  }
  function h(index) {
    if (index === data.length - 2) {
      if (pageno === 1) {
        return (
          <View style={{ position: 'absolute', bottom: 80,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={[[styles.tooglebutton,]]} >
              /* <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>back</Text> */
              <Icon name='arrow-back-ios-new'
style={[[{color:'white',fontSize:20,textAlign:'center'}]]} />
            </TouchableOpacity>
          </View>
        );
      } else {
        return (
          <View style={{ position: 'absolute', bottom: 70,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={[[styles.tooglebutton,{backgroundColor:'rgba(225, 225, 225,.3)'}]]} >
              /* <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>prev</Text> */
              <Icon name='arrow-back-ios-new'
style={[[{color:'white',fontSize:20,textAlign:'center'}]]} />
            </TouchableOpacity>
            <TouchableOpacity onPress={() => {setPageno(pageno -
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={[[styles.tooglebutton,{position:'absolute',bottom:0,right:15}]]} >
              /* <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>New</Text> */
              <Icon name='navigate-next'
style={[[{color:'white',fontSize:20,textAlign:'center'}]]} />
            </TouchableOpacity>
          </View>
        );
      }
    }
  }

```

```

    }
  } else {
    return null;
  }
}

useEffect(() => {
  fetchData();
}, [pageno]);

const renderRow = ({ item, index }) => {
  if (index % 2 === 0) {
    return (
      <View>
        <View style={styles.row}>
          <View style={[styles.movie, { marginRight:"2%",marginLeft:"1%" }]}>
            <Pressable onPress={()=>{navigation.navigate('movie',{title:item.link})}} >
              <Image source={{ uri: item.image }} style={styles.image} />
              <Text style={styles.text}>{item.title.slice(0,
item.title.indexOf("("))}</Text>
            </Pressable>
          </View>
          {index + 1 < data.length && (
            <View style={[styles.movie]}>
              <Pressable onPress={()=>{navigation.navigate('movie',{title:data[index +
1].link})}} >
                <Image source={{ uri: data[index + 1].image }} style={styles.image} />
                <Text style={styles.text}>{data[index + 1].title.slice(0, data[index +
1].title.indexOf("("))}</Text>
              </Pressable>
            </View>
          )}
        </View>
        {index>=data.length-2? <View style={{width:"100%",height:100}} ></View> : null}
        {h(index)}
      </View>
    );
  }
};

return (
  <View style={styles.view}>
    <FlatList
      data={data}
      renderItem={renderRow}
      keyExtractor={(item, index) => item.image}

      ref={flatlist}
      refreshControl={

```

```

        <RefreshControl refreshing={refreshing} onRefresh={fetchData} />
    }

    />
</View>
);
}

const styles = StyleSheet.create({
  image: {
    width: 170,
    height: 260,
    borderRadius: 10,
  },
  text: {
    width: 170,
    fontSize: 20,
    color: 'black',
    textAlign: "center",
    color: "white",
  },
  movie: {
    flex: 1,
    height: 350,
    flexDirection: 'column',
    marginTop: 10,
  },
  row: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 5,
    marginBottom: -30,
  },
  view: {
    flex: 1,
    width: '100%',
    backgroundColor: 'rgba(6, 2, 38,1)',
  },
  tooglebutton: {
    backgroundColor:'rgba(225, 225, 225,.3)',
    padding: 7,
    borderRadius:40,
    width:40,
    height:40
  },
  pagination: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 10,
  },
});

```

English.js:

```
import { View, Text, Image, StyleSheet, FlatList,
TouchableOpacity,Pressable,RefreshControl } from 'react-native';
import React, { useState, useEffect, useRef } from 'react';
import axios from 'axios';
import Icon from 'react-native-vector-icons/MaterialIcons'
export default function Malayalam({navigation}) {
  const [data,setData]=useState([])
  const [pageno, setPageno] = useState(1);
  const [refreshing, setRefreshing] = React.useState(false);
  const flatlist=useRef(null);
  async function fetchData() {
    setRefreshing(true);
    try {
      const response = await
axios.get(`https://movierulz.vercel.app/malayalam/${pageno}`);
      if (response.status === 200) {
        setData(response.data.data)
        setRefreshing(false);
      } else {
        console.log('Failed to fetch data');
        setRefreshing(false);
      }
    } catch (error) {
      console.error('Error fetching data:', error);
      setRefreshing(false);
    }
  }
  function h(index) {
    if (index === data.length - 2) {
      if (pageno === 1) {
        return (
          <View style={{ position: 'absolute', bottom: 80,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style=[styles.tooglebutton,]] >
              {/* <Text
style=[[{color:'white',fontSize:20,textAlign:'center'}]]>back</Text> */}
              <Icon name='arrow-back-ios-new'
style=[[{color:'white',fontSize:20,textAlign:'center'}]] />
            </TouchableOpacity>
          </View>
        );
      } else {
        return (
          <View style={{ position: 'absolute', bottom: 70,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style=[styles.tooglebutton,{backgroundColor:'rgba(225, 225, 225,.3)'}]] >
```

```

        { /* <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>prev</Text> */}
        <Icon name='arrow-back-ios-new'
style={[[{color:'white',fontSize:20,textAlign:'center'}]]} />
        </TouchableOpacity>
        <TouchableOpacity onPress={() => {setPageno(pageno -
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={[[styles.tooglebutton,{position:'absolute',bottom:0,right:15}]]} >
        { /* <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>New</Text> */}
        <Icon name='navigate-next'
style={[[{color:'white',fontSize:20,textAlign:'center'}]]} />
        </TouchableOpacity>
    </View>
    );
  }
} else {
  return null;
}
}

useEffect(() => {
  fetchData();
}, [pageno]);

const renderRow = ({ item, index }) => {
  if (index % 2 === 0) {
    return (
      <View>
        <View style={styles.row}>
          <View style={[[styles.movie, { marginRight:"2%",marginLeft:"1%" }]]}>
            <Pressable onPress={()=>{navigation.navigate('movie',{title:item.link})}} >
              <Image source={{ uri: item.image }} style={styles.image} />
              <Text style={styles.text}>{item.title.slice(0,
item.title.indexOf("("))}</Text>
            </Pressable>
          </View>
          {index + 1 < data.length && (
            <View style={[[styles.movie]]}>
              <Pressable onPress={()=>{navigation.navigate('movie',{title:data[index +
1].link})}} >
                <Image source={{ uri: data[index + 1].image }} style={styles.image} />
                <Text style={styles.text}>{data[index + 1].title.slice(0, data[index +
1].title.indexOf("("))}</Text>
              </Pressable>
            </View>
          )}
        </View>
        {index>=data.length-2? <View style={{width:"100%",height:100}} ></View> : null}

```

```

        {h(index)}
      </View>
    );
  }
};

return (
  <View style={styles.view}>
    <FlatList
      data={data}
      renderItem={renderRow}
      keyExtractor={(item, index) => item.image}

      ref={flatlist}
      refreshControl={
        <RefreshControl refreshing={refreshing} onRefresh={fetchData} />
      }

    />
  </View>
);
}

const styles = StyleSheet.create({
  image: {
    width: 170,
    height: 260,
    borderRadius: 10,
  },
  text: {
    width: 170,
    fontSize: 20,
    color: 'black',
    textAlign: "center",
    color: "white",
  },
  movie: {
    flex: 1,
    height: 350,
    flexDirection: 'column',
    marginTop: 10,
  },
  row: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 5,
    marginBottom: -30,
  },
  view: {
    flex: 1,
    width: '100%',

```

```

    backgroundColor: 'rgba(6, 2, 38,1)',
  },
  togglebutton: {
    backgroundColor:'rgba(225, 225, 225,.3)',
    padding: 7,
    borderRadius:40,
    width:40,
    height:40
  },
  pagination: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 10,
  },
});

```

Search.js:

```

import { View, Text, Image, StyleSheet, FlatList, TouchableOpacity, Pressable } from
'react-native';
import React, { useState, useEffect, useRef } from 'react';
import axios from 'axios';
import { TextInput } from 'react-native-gesture-handler';

export default function Search({navigation}) {
  const [data, setData] = useState([])
  const flatlist = useRef(null);
  const [input, setInput] = useState(null)
  const inputRef = useRef()
  async function fetchData() {
    try {
      const response = await
axios.get(`https://movierulz.vercel.app/search?query=${input}`);
      if (response.status === 200) {
        setData(response.data.data)
      } else {
        console.log('Failed to fetch data');
      }
    } catch (error) {
      console.error('Error fetching data:', error);
    }
  }
  useEffect(() => {
    if (input !== null) {
      fetchData()
    }
  }, [input])
  function h(index) {

```



```

    if (index === data.length - 2) {
      if (pageno === 1) {
        return (
          <View style={{ position: 'absolute', bottom: 80,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={styles.tooglebutton,}] >
              <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>back</Text>
            </TouchableOpacity>
          </View>
        );
      } else {
        return (
          <View style={{ position: 'absolute', bottom: 70,width:'100%',left:5 }}>
            <TouchableOpacity onPress={() => {setPageno(pageno +
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={styles.tooglebutton} >
              <Text
style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>prev</Text>
            </TouchableOpacity>
            <TouchableOpacity onPress={() => {setPageno(pageno -
1);flatlist.current?.scrollToOffset({ offset: 0, animated: true });}}
style={[[styles.tooglebutton,{position:'absolute',bottom:0,right:15}]]} >
              <Text style={[[{color:'white',fontSize:20,textAlign:'center'}]]}>New</Text>
            </TouchableOpacity>
          </View>
        );
      }
    } else {
      return null;
    }
  }
}

const renderRow = ({ item, index }) => {
  if (index % 2 === 0) {
    return (
      <View>
        <View style={styles.row}>
          <View style={[[styles.movie, { marginRight:"2%",marginLeft:"1%" }]]}>
            <Pressable onPress={()=>{navigation.navigate('movie',{title:item.link})}} >
              <Image source={{ uri: item.image }} style={styles.image} />
              /* <Text style={styles.text}>{item.title.slice(0,
item.title.indexOf("("))}</Text> */
              <Text style={[[styles.text]]} >{
                item.title.includes("DVDScr") || item.title.includes("HDRip") ?
                item.title.slice(0,item.title.indexOf('(')) :
                item.title.includes("Season") ?
item.title.slice(0,item.title.indexOf('(')) +
item.title.slice(item.title.indexOf('[')+1,item.title.indexOf(''))

```

```

        : item.title
    }
    </Text>
  </Pressable>
</View>
{index + 1 < data.length && (
  <View style={styles.movie}>
    <Pressable onPress={()=>{navigation.navigate('movie',{title:data[index +
1].link)}}} >
      <Image source={{ uri: data[index + 1].image }} style={styles.image} />
    {/*
      <Text style={styles.text}>{data[index + 1].title.slice(0, data[index
+ 1].title.indexOf("("))}</Text>
    */}

    <Text style={styles.text}>{
      data[index + 1].title.includes("DVDScr")
|| data[index + 1].title.includes("HDRip") ?
      data[index + 1].title.slice(0,data[index +
1].title.indexOf('(')) :
      data[index + 1].title.includes("Season") ? data[index +
1].title.slice(0,data[index + 1].title.indexOf('(')) + data[index +
1].title.slice(data[index + 1].title.indexOf('[')+1,data[index + 1].title.indexOf(']'))
      : data[index + 1].title
    }
  </Text>
  </Pressable>
</View>
)}

</View>
{index>=data.length-2?
  <View style={{width:"100%",height:50,backgroundColor:'rgba(0,0,0,0)'}}
></View>:null
}

  {/* {h(index)} */}
  {/* {index>=data.length-2? <View style={{width:"100%",height:100}} ></View> :
null} */}
  </View>
);
}
};

return (
  <View style={styles.view}>
    <TextInput placeholder='Search' placeholderTextColor={'rgba(6, 2, 38,.7)'}
style={{color:'rgba(6, 2,
38,.7)',fontWeight:'bold',fontSize:30,backgroundColor:'rgba(225, 225,
225,.7)',width:300,height:60,borderRadius:40,textAlign:'center',position:'relative',top
:100,left:55}}
    onChangeText={value=>{
      setInput(value)
      inputRef.current.setNativeProps({

```

```

        top:10,
        zIndex:1
    })
  }}
  ref={inputRef}
/>
  {input === null ? <Text
style={{color:'white',fontSize:40,textAlign:'center',marginTop:300}} >No search
Founded</Text>:<FlatList
  data={data}
  renderItem={renderRow}
  keyExtractor={(item, index) => item.title}

  ref={flatlist}

/>}

</View>
);
}

const styles = StyleSheet.create({
  image: {
    width: 170,
    height: 260,
    borderRadius: 10,
  },
  text: {
    width: 170,
    fontSize: 20,
    color: 'black',
    textAlign: "center",
    color: "white",
  },
  movie: {
    flex: 1,
    height: 350,
    flexDirection: 'column',
    marginTop: 10,
  },
  row: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 5,
    marginBottom: -30,
  },
  view: {
    flex: 1,
    width: '100%',
    backgroundColor: 'rgba(6, 2, 38,1)',
    // justifyContent:'center',

```

```

    // alignItems:'center'
  },
  togglebutton: {
    backgroundColor: 'rgba(6, 2, 38,.7)',
    padding: 10,
    borderRadius:100,
    width:100
  },
  pagination: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 10,
  },
});

```

WhishList.js:

```

import { View, Text, Image, StyleSheet, ScrollView, FlatList, Pressable,
RefreshControl, Touchable } from 'react-native';

import React, { useState, useEffect, useContext } from 'react';
import axios from 'axios';
import AsyncStorage from '@react-native-async-storage/async-storage'
import { useRefresh } from '../componets/MovieContext';
import { TouchableOpacity } from 'react-native-gesture-handler';
export default function WishList({navigation}) {
  const [data, setData] = useState([]);
  const [refreshing, setRefreshing] = React.useState(false);

  async function fetchData() {
    setRefreshing(true);
    try {
      let response
      await AsyncStorage.getItem('movierulzWhish',(err,res)=>{
        if(res){
          response=JSON.parse(res)
          console.log(response)
        }
      });
      if (response.status === 200) {
        console.log(response.status)
        setData(response.data);
        setRefreshing(false);
      } else {
        console.log('Failed to fetch data');
        setRefreshing(false);
      }
    } catch (error) {
      console.error('Error fetching data:', error);
      setRefreshing(false);
    }
  }

```

```

    }
  }

  const onRefresh = React.useCallback(() => {
    setRefreshing(true);
    setTimeout(() => {
      setRefreshing(false);
    }, 2000);
  }, []);
  const { forRefresh, setRefresh } = useRefresh();

  useEffect(() => {
    fetchData();
  }, [forRefresh]);

  async function unWish(url){
    await AsyncStorage.getItem('movierulzWhish', async(err, res)=>{
      if(res!=null){
        let retrieveData=JSON.parse(res)
        let newData={data:[],status:200}
        retrieveData.data.forEach(async element => {
          if(element.link!=url){
            console.log(element)
            newData.data.push(element)
          }
        })
        await
        AsyncStorage.setItem('movierulzWhish', JSON.stringify(newData), (err)=>{
          if(!err){
            setRefresh(forRefresh+1)
          }
        })
      }
    });
  }
})

const renderRow = ({ item, index }) => {
  if (index % 2 === 0) {
    return (
      <View style={styles.row}>
        <View style={[styles.movie, {marginRight:"2%",marginLeft:"1%"}]}>
          <Pressable onPress={()=>{navigation.navigate('movie',{title:item.link})}}
onLongPress={()=>{unWish(item.link)}} >
            <Image source={{ uri: item.image }} style={styles.image} />
            <Text style={styles.text} >{
              item.title.slice(0,item.title.indexOf('('))
              // item.title
            }
          </Text>
        </Pressable>
      </View>
    )
  }
}

```

```

    </View>
    {index + 1 < data.length && (
      <View style={styles.movie}>
        <Pressable onPress={()=>{navigation.navigate('movie',{title:data[index +
1].link})}} onLongPress={()=>{unWish(data[index + 1].link)}} >
          <Image source={{ uri: data[index + 1].image }} style={styles.image} />
          <Text style={styles.text}>{data[index + 1].title.slice(0,data[index +
1].title.indexOf("("))}</Text>
        </Pressable>
      </View>
    )}
  </View>
);
}
};

return (
  <View style={styles.view}>
    {data.length?<FlatList
      data={data}
      renderItem={renderRow}
      keyExtractor={(item, index) => index.toString()}
      keyboardDismissMode='on-drag'
      // refreshing={true}
      // onRefresh={fetchData}
      refreshControl={
        <RefreshControl refreshing={refreshing} onRefresh={fetchData} />
      }
      />:<TouchableOpacity onPress={()=>{navigation.navigate('Home')}}
style={{backgroundColor:'rgba(225,225,225,.2)',marginTop:'100%',width:300,marginLeft:"1
0%",borderRadius:10,padding:20}} ><Text
style={{color:'white',fontSize:20,fontWeight:'bold',textAlign:'center'}} >Add Movies to
Watch later</Text></TouchableOpacity>
    </View>
  );
}

const styles = StyleSheet.create({
  image: {
    width: 170,
    height: 260,
    borderRadius: 10,
  },
  text: {
    width: 170,
    fontSize: 20,
    // marginRight:5,
    color: 'black',
    textAlign:"center",
    color:"white",
    // borderWidth:1,

```

```
// borderColor:"red"
},
movie: {
  flex: 1,
  height: 350,
  flexDirection: 'column',
  marginTop:10
},
row: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  paddingHorizontal:5,
  marginBottom: -30,
},
view: {
  flex: 1,
  width: '100%',
  backgroundColor: 'rgba(6, 2, 38,1)',
},
});
```

INTERSHIP CERTIFICATE



Completion Certificate

Proudly Presented to

VALLEPU RAMA HARESHKIRAN

successfully completed the Virtual Internship Program at
BHARAT INTERN in App Development as an active participant
from September 10, 2023 to October 10, 2023.



Verified by,
BHARAT INTERN