Program No: 10

Project Name: UniFest

Brief description about project:

UniFest is an open platform for all the colleges and universities to view and add fests and events in such a way that they don't clash with other college events or fests and are thus conducted in a smooth and successful manner.

React-Native Code:

App.js

```
import React, { Component} from 'react';
import {createAppContainer } from 'react-navigation';
import Home from './Home';
import { View,StyleSheet } from 'react-native';
import Splash from './Splash.js';
import SignUp from './SignUp'
import AppNavigator from './AppNavigator'



const AppContainer = createAppContainer(AppNavigator);
export default class App extends React.Component {
   constructor(props) {
      super(props);

      this.state = { isLoading: true }
     }
     componentDidMount()
   {
```

```
    setTimeout(() => {
      this.setState({isLoading: false});
    },5000)
  }
  render() {
    if(this.state.isLoading)
    return(
      <View style={styles.splash}>
        <Splash/>
      </View>
    )
    return <AppContainer />;
  }
}
const styles = StyleSheet.create({
  splash:{
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: '#448AFF',
  },
});
```

Splash.js

```
import React, { Component, } from 'react';
import { Text, View,StyleSheet } from 'react-native';

export default class Splash extends Component {
```

```
  constructor(props)
  {
    super(props);
  }
  render() {
    return (
      <View style={[styles.splash,this.props.style]}>
          <Text style={styles.splashtext}>
              uf
          </Text>
      </View>
    );
  }
}
const styles = StyleSheet.create({
  splash:{
    backgroundColor:'#1565C0',
    width:200,
    height:200,
    alignItems:"center",
    justifyContent:"center",
    borderRadius:200,
    shadowColor: '#000000',
    shadowOffset: { width: 5, height: 100 },
    shadowOpacity: 1,
    shadowRadius: 20,
    elevation: 20,
```

```
  },
  splashtext:
  {

     color:'white',
     fontSize:100,
    //  width:200,
    //  height:200,
  }

});
```

SignUp.js

```
import React, { Component, } from 'react';
import { Text, View,StyleSheet,TextInput,TouchableOpacity,KeyboardAvoidingView} from 'react-native';
import axios from 'axios';

export default class SignUp extends Component {
  constructor(props)
  {
    super(props);
    this.state={
      Username:"",
      Password:"",
      ConfirmPassword:"Password Does Not Match",
      color:"",
      response:"",
```

```
    error:"",
   }
   this.handlesubmit=this.handlesubmit.bind(this);
  }
  handlesubmit()
  {
   const {navigate}=this.props.navigation;
   console.log(this.state.Username,this.state.Password)
   if(this.state.ConfirmPassword==="Passwords Match"&&!(this.state.Username===""))
   {
    axios.post('http://192.168.43.158/api/auth/register/', {
    username: this.state.Username,
    password: this.state.Password,
   })
   .then((response) => {
    console.log(response);
    this.setState(
     {
      response:response,
      error:"",
     }
    );
    console.log(this.state.response)
    navigate('Home')
   })
   .catch((error)=> {
    console.log(error);
```

```
    this.setState(
      {
        error:""+error,
      }
    )
  });
  }
 }
 render() {
  return (
    <View style={styles.content}>
      <View style={styles.input}>
      <TextInput placeholder="Enter Username" style={styles.inputUsername}
onChangeText={(Username) => this.setState({Username})} />
      <TextInput secureTextEntry={true} placeholder="Enter Password"
style={styles.inputUsername} onChangeText={(Password) => this.setState({Password})} />
      <TextInput secureTextEntry={true} placeholder="Confirm Password"
style={styles.inputUsername} onChangeText={(CPassword)
=>{if(this.state.Password===CPassword){this.setState({ConfirmPassword:"Passwords
Match",color:"green"})}}}
     />
      <TouchableOpacity style={styles.buttonSubmit} onPress={this.handlesubmit}>
          <Text style={styles.buttonText}>Submit</Text>
        </TouchableOpacity>
        <Text
style={[{alignSelf:"center",color:this.state.color||'red'}]}>{this.state.ConfirmPassword}</Tex
t>
        <Text style={[{alignSelf:"center",color:'red'}]}>{this.state.error}</Text>
      </View>
    </View>
```

```
    );
  }
}
const styles = StyleSheet.create({
  inputUsername:
  {
  alignSelf:"flex-start",
  height:50,
  width:300,
  borderBottomColor:'#0D47A1',
  borderBottomWidth:1,
  fontSize:20,
  marginBottom: 30,
  },
  input:
{
    width:300,
},
content:{
  flex:1,
  flexDirection:"column",
  justifyContent:"space-around",
  alignItems:"center",
},
buttonSubmit:
{
  backgroundColor:'#1565C0',
```

```
  width:300,
  height:50,
  alignItems: 'center',
  justifyContent: 'center',
  borderRadius: 5,
},
buttonText:
{
  color:'white',
  alignSelf:'center',
  fontSize:20,
}
});
```

Input.js

```
import React, { Component, } from 'react';
import { Text, View,StyleSheet,TextInput,TouchableOpacity} from 'react-native';
import Splash from './Splash';
import axios from 'axios';


export default class Input extends Component {
  constructor(props)
  {
    super(props);
    this.state={
      Username:"",
      Password:"",
      Login:"",
```

```
    response:"",
    error:"",
  };
  this.handleLogin=this.handleLogin.bind(this);
 }
 handleLogin()
 {
  const {navigate}=this.props.navigation
  axios.post('http://192.168.43.158/api/auth/login/', {
    username: this.state.Username,
    password: this.state.Password,
  })
  .then((response) => {
    console.log(response);
    this.setState(
      {
        response:response,
        error:"",
      }
    );
    console.log(this.state.response)
    navigate('Fest',{token:this.state.response.data.token})
  })
  .catch((error)=> {
    console.log(error);
    this.setState(
      {
```

```
        error:""+error,
      }
    )
  });
 }
 render() {
  return (
    <View style={styles.input}>
      <Splash style={styles.resize}/>
      <TextInput placeholder="Enter Username" style={styles.inputUsername}
onChangeText={(Username) => this.setState({Username})} ></TextInput>
      <TextInput secureTextEntry={true} placeholder="Enter Password"
style={styles.inputUsername} onChangeText={(Password) =>
this.setState({Password})}></TextInput>
      <View>
      <TouchableOpacity style={styles.buttonLogin} onPress={this.handleLogin} >
        <Text style={styles.buttonText}>Login</Text>
      </TouchableOpacity>
      <TouchableOpacity style={[styles.buttonLogin,{top:60}]} onPress={() =>{const
{navigate}=this.props.navigation;navigate('SignUp')}}>
        <Text style={styles.buttonText}>Sign Up</Text>
      </TouchableOpacity>
      </View>
      <Text style={{color:'red',position:'absolute',alignSelf:'center',top:300}}>
       {this.state.error}
      </Text>
    </View>
  );
```

```
}
}
const styles = StyleSheet.create({
  input:
{
    width:300,
},
label:
{
    fontSize:20,
    alignSelf:"flex-start",
},
inputUsername:
{
  alignSelf:"flex-start",
  height:50,
  width:300,
  borderBottomColor:'#0D47A1',
  borderBottomWidth:1,
  fontSize:20,
  marginBottom: 30,
},
resize:
{
  width:150,
  height:150,
```

```
  alignSelf:"center",
  position:'absolute',
  bottom:200,
},
buttonLogin:
{
  backgroundColor:'#1565C0',
  width:300,
  height:50,
  alignItems: 'center',
  justifyContent: 'center',
  position:'absolute',
  borderRadius: 5,
},
buttonText:
{
  color:'white',
  alignSelf:'center',
  fontSize:20,
}
});
```

Home.js

```
import React, { Component, } from 'react';
import {View,StyleSheet} from 'react-native';
import Input from './Input.js'
```

```
export default class Home extends Component {



  render() {
    return (
      <View style={styles.content}>
        <Input navigation={this.props.navigation}/>
      </View>
    );
  }
}
const styles = StyleSheet.create({
  splash:{
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: '#448AFF',
  },
  content:{
    flex:1,
    flexDirection:"column",
    justifyContent:"space-around",
    alignItems:"center",
  },
});
```

Fest.js

```
import React, { Component, } from 'react';
```

```
import { Text, View,StyleSheet,FlatList,TouchableOpacity} from 'react-native';
import axios from 'axios';
export default class Fest extends Component {
  constructor(props)
  {
    super(props);
    this.state={
      isLoading:true,
      token:'',
      fests:'',
    };
    this.navigateToDetail=this.navigateToDetail.bind(this);
  }
  componentDidMount()
  {
    const { navigation } = this.props;
    const token = navigation.getParam('token', '');
    this.setState(
      {
        token:token
      }
    )
    console.log(token)
    const authpayload='Token '+token
    console.log(authpayload)
    axios.get('http://192.168.43.158/api/Fests/', { headers: { Authorization:authpayload } })
 .then(response => {
```

```
    // If request is good...
    console.log(response.data);
    this.setState(
      {
        fests:response.data,
      }
    );
    console.log(this.state.fests)
  })
 .catch((error) => {
    console.log('error ' + error);
  });


  }
  navigateToDetail()
  {


  }


  render() {
    return (
      <View>
            <FlatList
               data={this.state.fests}
               renderItem={({item}) =>
               <View>
                 <TouchableOpacity style={styles.card} onPress={()=>
```

```
                {
                  const {navigate}=this.props.navigation
                  console.log("Navigated")
                  console.log(item.name)
                  navigate('Detail',{token:this.state.token,name:item.name})

                }}>
                  <Text style={styles.title}>{item.name}</Text>
                  <Text>{item.description}</Text>
                </TouchableOpacity>
              </View>
                }
            />
          </View>
    );
  }
}
const styles = StyleSheet.create({
  card:{
    backgroundColor:"#E3F2FD",
    padding:20,
    borderBottomColor:'#2196F3',
    borderBottomWidth:1,
  },
  title:
  {
    fontSize:25,
```

```
  },


});
```

Detail.js

```
import React, { Component, } from 'react';
import { Text, View,StyleSheet,FlatList} from 'react-native';
import axios from 'axios'
export default class Detail extends Component {
  constructor(props)
  {
    super(props);
    this.state={
        events:'',
    }
  }
  componentDidMount()
  {
    const {navigation}=this.props
    const token = navigation.getParam('token', '');
    console.log(token)
    const name = navigation.getParam('name', '');
    console.log(name)
    console.log(token)
    const authpayload='Token '+token
    console.log(authpayload)
    axios.get('http://192.168.43.158/api/FestEvent/'+name+'/', { headers: {
Authorization:authpayload } })
```

```
.then(response => {
    // If request is good...
    console.log(response.data);
    this.setState(
        {
            events:response.data
        }
    )
})
.catch((error) => {
    console.log('error ' + error);
});
}
render() {
  return (
    <View >
        <FlatList
              data={this.state.events}
              renderItem={({item}) =>
              <View style={styles.cards}>
                 <Text style={styles.text}>
                    {item.name}
                 </Text>
              </View>
                 }
            />
    </View>
```

```
    );
  }
}
const styles = StyleSheet.create({
  cards:
  {
    backgroundColor:"#E3F2FD",
    padding:20,
    borderBottomColor:'#2196F3',
    borderBottomWidth:1,
  },
  text:
  {
    fontSize:20,
  }

});
```

AppNavigator.js

```
import {createStackNavigator} from 'react-navigation-stack'
import Home from './Home';
import SignUp from './SignUp'
import Fest from './Fest'
import Detail from './Detail'
const AppNavigator = createStackNavigator(
    {
        Home: {screen:Home,
            navigationOptions:{
```

```
            header:null,
          }
      },
      SignUp:{ screen: SignUp,
        navigationOptions:{
          title:'Sign Up',
        }
      },
      Fest:{
        screen:Fest,
        navigationOptions:{
          title:'Fests',
          headerLeft: null,
        }
      },
      Detail:{
        screen:Detail,
        navigationOptions:{
          title:'Events',
        }
      }

  },
  {
    initialRouteName: "Home"
  }
);
```

```
export default AppNavigator;
```

Web Service Code:

UniFestAPI/urls.py

```
"""UniFestAPI URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.urls import path, include
from django.conf.urls import url
from django.contrib import admin



urlpatterns = [
    path('admin/', admin.site.urls),
    url(r'api/', include('UniFestAPP.urls')),
]
```

UniFestAPP/urls.py

```
from django.conf.urls import url
from rest_framework.authtoken.views import obtain_auth_token
from .views import CreateUserAPIView, LogoutUserAPIView,Fests,Event,EventFest



urlpatterns = [
    url(r'^auth/login/$',
        obtain_auth_token,
        name='auth_user_login'),
    url(r'^auth/register/$',
        CreateUserAPIView.as_view(),
        name='auth_user_create'),
    url(r'^auth/logout/$',
        LogoutUserAPIView.as_view(),
        name='auth_user_logout'),


url(r'^Fests/$',
        Fests.as_view(),
        name='Fests'),
url(r'^Events/$',
        Event.as_view(),
        name='Events'),
url(r'^FestEvent/(?P<Fest1>[\w\-]+)/$',EventFest.as_view(),name='EventFest'),
]
```

views.py

```
from django.shortcuts import render
```

```
# Create your views here.
from django.contrib.auth import get_user_model
from rest_framework.generics import CreateAPIView
from rest_framework.permissions import AllowAny
from rest_framework.response import Response
from rest_framework.authtoken.models import Token
from rest_framework import status
from rest_framework.views import APIView
from UniFestAPP.serializers import CreateUserSerializer,FestSerializer,EventSerializer
from rest_framework.permissions import IsAuthenticated
from .models import Fest,Events

class CreateUserAPIView(CreateAPIView):
    serializer_class = CreateUserSerializer
    permission_classes = [AllowAny]

    def create(self, request, *args, **kwargs):
        serializer = self.get_serializer(data=request.data)
        serializer.is_valid(raise_exception=True)
        self.perform_create(serializer)
        headers = self.get_success_headers(serializer.data)
        # We create a token than will be used for future auth
        token = Token.objects.create(user=serializer.instance)
        token_data = {"token": token.key,}
        return Response(
            {**serializer.data, **token_data},
```

```
            status=status.HTTP_201_CREATED,
            headers=headers
        )


class LogoutUserAPIView(APIView):
    queryset = get_user_model().objects.all()

    def get(self, request, format=None):
        # simply delete the token to force a login
        request.user.auth_token.delete()
        return Response(status=status.HTTP_200_OK)

class Fests(APIView):
    def get(self,request):
        data=Fest.objects.all()
        serializer=FestSerializer(data,many=True)
        return Response(serializer.data)
    def post(self,request):
        serializer=FestSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data,status=status.HTTP_201_CREATED)
        return Response(serializer.errors,status=status.HTTP_400_BAD_REQUEST)

class Event(APIView):
    def get(self,request):
```

```
        data=Events.objects.all()
        serializer=EventSerializer(data,many=True)
        return Response(serializer.data)
    def post(self,request):
        serializer=EventSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data,status=status.HTTP_201_CREATED)
        return Response(serializer.errors,status=status.HTTP_400_BAD_REQUEST)
class EventFest(APIView):
    def get(self,request,Fest1):
        print(Fest1)
        fest=Fest.objects.get(name=Fest1)
        print(fest.id)
        data=Events.objects.filter(fest=fest.id)
        serializer=EventSerializer(data,many=True)
        return Response(serializer.data)
```

models.py

```
from django.db import models


# Create your models here.
class Fest(models.Model):
    name=models.CharField(max_length=24)
    description=models.CharField(max_length=200,default="")


    def __str__(self):
        return self.name
```

```
class Events(models.Model):
    name=models.CharField(max_length=25)
    fest=models.ForeignKey(Fest,on_delete=models.CASCADE,related_name='events')


    def __str__(self):
        return self.name
```

admin.py

```
from django.contrib import admin

# Register your models here.
from .models import Fest,Events

@admin.register(Fest)
class FestAdmin(admin.ModelAdmin):
    search_fields =['name','description']
    list_display =['name','description']
    list_filter = ['name']



@admin.register(Events)
class EventAdmin(admin.ModelAdmin):
    search_fields =['name','fest']
    list_display =['name','fest']
    list_filter = ['name','fest']
```

serializer.py

```
from django.contrib.auth import get_user_model
from rest_framework import serializers
```

```
from .models import Fest,Events
class CreateUserSerializer(serializers.ModelSerializer):
    username = serializers.CharField()
    password = serializers.CharField(write_only=True,
                                     style={'input_type': 'password'})

    class Meta:
        model = get_user_model()
        fields = ('username', 'password', 'first_name', 'last_name')
        write_only_fields = ('password')
        read_only_fields = ('is_staff', 'is_superuser', 'is_active',)

    def create(self, validated_data):
        user = super(CreateUserSerializer, self).create(validated_data)
        user.set_password(validated_data['password'])
        user.save()
        return user

class FestSerializer(serializers.ModelSerializer):
    class Meta:
        model = Fest
        fields='__all__'

class EventSerializer(serializers.ModelSerializer):
    class Meta:
        model=Events
        fields='__all__'
```

Sample Output: