APPENDIX 3 – CODES FOR MAIN TECHNIQUES

- **Sign-up Choice:**

FRONT-END

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


class SignUpChoice extends Component {
    teacherNavigate = () => {
        this.props.navigation.navigate('SignUpTeacher')
    }
    studentNavigate = () => {
        this.props.navigation.navigate('SignUpStudent')
    }
    loginNavigate = () => {
        this.props.navigation.goBack()
    }
    render(){
        return (
            <View style = {{flex:1}}>
                <TouchableOpacity onPress={() => this.teacherNavigate()} style={{
flex:3,backgroundColor:'red',justifyContent:'center',alignItems:'center'}}>
                    <Text style={{color:'white',fontSize:20}}> Teacher </Text>
                </TouchableOpacity>
                <TouchableOpacity onPress={() => this.studentNavigate()} style={{
flex:3,backgroundColor:'pink',justifyContent:'center',alignItems:'center'}}>
                    <Text style={{color:'black',fontSize:20}}> Student </Text>
                </TouchableOpacity>
                <TouchableOpacity onPress={() => this.loginNavigate()} style={{fl
ex:1,backgroundColor:'aqua',justifyContent:'center',alignItems:'center'}}>
                    <Text style={{color:'black'}}> Back to login </Text>
                </TouchableOpacity>
            </View>
        )
    }
}
export default SignUpChoice;
```

- **Generating OTP:**

FRONT-END:

```
import React, { Component } from 'react';
import { View,Text,TextInput,TouchableOpacity,Image } from 'react-native';
import AsyncStorage from '@react-native-community/async-storage';
import helpers from './helpers';

class Otp extends Component {

    constructor(props){
        super(props);

        this.state = {
          UserEmail: '',
          userOtp: ''
        }
    }
    componentDidMount(){
        this.getparamdata();
    }
    getparamdata = async () => {
        global.email = await this.props.navigation.state.params.UserEmail;

    }
    otp = () =>{
        // alert('please confirm your otp & verify yourself')
        // this.props.navigation.navigate('Otp')
        // return false;
        console.log(`otp is ${this.state.userOtp} and email is ${global.email}`);

        return fetch(helpers.baseUrl+'user_registration.php',{

            method: 'POST',
            headers: {
              'Accept': 'application/json',
              'Content-Type': 'application/json',
            },
            body: JSON.stringify({
              email: global.email,
              otp:this.state.userOtp
            })

          }).then((response) => response.json())
        .then((responseJson) => {

          this.props.navigation.navigate('SigninScreen')
        }).catch((error) => {

         this.props.navigation.navigate('SigninScreen')
        });

    }
  goToSignup = () => {
    this.props.navigation.navigate('SignUpChoice');
  }
  render(){
    return (
      <View style = {{flex:1, marginTop: 20, backgroundColor: 'white'}}>
        <View style={{flex:2 }}>
          <View style = {{flex:1.5, paddingTop: 20}}>

          <View style = {{flex: 1}}>
            <Image
            source = {require('./jpisLogo.jpeg')}
            style = {{height:'100%', width: '100%', resizeMode: 'cont
ain'}}

            />

          </View>

          </View>
          <View style = {{flex:1, justifyContent: 'space-
around', alignItems: 'center', backgroundColor: 'white'}}>
          <Text style = {{fontSize: 30}}>
            Event Manager App
          </Text>

          </View>



        </View>
        <View style={{flex:2, paddingRight: 20, paddingLeft:20}}>
          <View style = {{flex: 1}}>
          <TextInput
          style = {{height: 80, borderBottomColor: 'black', borderBotto
mWidth: 1}}
          onChangeText = {userOtp => this.setState({userOtp})}
          placeholder = 'ENTER OTP'
```

```
                    />
                    </View>
                    <TouchableOpacity onPress={() => this.otp()} style = {{flex:
0.4, justifyContent: 'center', alignItems: 'center', backgroundColor: 'skyblue',
borderRadius: 30}}>
                    <Text style = {{fontSize: 20,fontWeight:'bold'}}>
                        Submit
                    </Text>

                    </TouchableOpacity>

                </View>
            </View>
        )
    }
}
export default Otp;
```

BACK-END:

```
<?php

include 'dbconfig.php';
$con = mysqli_connect($HostName,$HostUser,$HostPass,$DatabaseName);
$json = file_get_contents('php://input');
$obj = json_decode($json,true);


$id = $obj['id'];

$Sql_Query = "select otp from user where id = '$id'";
$check = mysqli_fetch_array(mysqli_query($con,$Sql_Query));
$dbOtp = $check->otp;
if(isset($check)){

    $change_status = "UPDATE user SET verified = 1  where id = '$id'  ";
    $check_password = mysqli_fetch_array(mysqli_query($con,$password_checker));
    if(isset($check_password)){
        $SuccessLoginMsg = ['status'=>'200','message'=>'Data Matched'];
        $SuccessLoginJson = json_encode($SuccessLoginMsg);
        echo $SuccessLoginJson ;
    }else{
        $InvalidMSG = ['status'=>'500','message'=>'Invalid Password, Please Try A
gain'];
```

```
        $InvalidMSGJSon = json_encode($InvalidMSG);
        echo $InvalidMSGJSon ;
    }

}else{
    $InvalidMSG = ['status'=>'500','message'=>'Email id is not registered with us Please Try Again'];
    $InvalidMSGJSon = json_encode($InvalidMSG);
    echo $InvalidMSGJSon ;
}

mysqli_close($con);
?>
```

- **Forgot password:**

```
import React, {Component} from 'react';
import { Text, View, StyleSheet,TextInput, Image, Button, ImageBackground, TouchableOpacity } from 'react-native';
import helpers from './helpers';
export default class Forgotpassword extends Component {


  forgetPassword =() => {
    alert('A mail has been sent to you, please follow instruction.')
    this.props.navigation.navigate('SigninScreen');
    return false;
      return fetch(helpers.baseUrl+'user_registration.php',{
      method: 'POST',
      headers: {
          'Accept': 'application/json',
          'Content-Type': 'application/json',
          'x_rest_username':'admin@restuser',
          'x_rest_password':'admin@Access',
          'x_rest_version':'v1'
      },
      body: JSON.stringify({
          email: this.state.UserEmail

      })

      }).then((response) => response.json())
          .then((responseJson) => {
              if(responseJson){
                  alert(JSON.stringify(responseJson));
                  return false;
                  // AsyncStorage.setItem('user', JSON.stringify(responseJson.dat
a.id))
                  // AsyncStorage.setItem('type', JSON.stringify(responseJson.dat
a.type))
                  this.props.navigation.navigate('HomeAppScreen');
              }else{
                alert(JSON.stringify(responseJson))
              }
          }).catch((error) => {
              // this.showHideSpinner(false);
              alert(error.toString());
          });
  }

  render() {
    return (
      <View style = {{flex:1}}

      >

        <View style={{flex:0.7 }}>
            <View style = {{flex:1.5, paddingTop: 2}}>

              <View style = {{flex: 1}}>
                <Image
                  source = {require('./jpisLogo.jpeg')}
                  style = {{height:'100%', width: '100%', resizeMode: 'contain'}}
                />

              </View>

            </View>
            <View style = {{flex:1, justifyContent: 'space-
around', alignItems: 'center', backgroundColor: 'white'}}>
              <Text style = {{fontSize: 26, color: 'green'}}>
                Student Sign-Up
              </Text>

            </View>


          </View>
          <View style={{flex:2, paddingRight: 20, paddingLeft:20, flexDirection: 'column' }}>

            <View style = {{flex: 1, justifyContent: 'center'}}>

              <TextInput
              style = {{height:50, borderBottomColor: 'black', borderBottomWidth: 1}}
              onChangeText = {UserEmail => this.setState({UserEmail})}
              placeholder = 'E-MAIL'

              />

            </View>
            <View style = {{flex: 0.2, paddingLeft: 30, paddingRight: 30}}>
              <TouchableOpacity onPress={() => this.forgetPassword()} style = {{marginBottom: 5, flex: 1, justifyContent: 'center', alignItems: 'center', backgroundColor: 'black', borderRadius: 20}}>
                <Text style = {{fontSize: 28, fontWeight: 'bold', color: 'white'}}>
                  Forgot password
                </Text>

              </TouchableOpacity>
            </View>

          </View>
      </View>
    )
  }
}
```

- **Add Event function on Teacher’s home screen:**

```
import * as React from 'react';
import { Text, View, StyleSheet,TextInput, Image, Button, ImageBackground, TouchableOpacity} from 'react-native';
import Spinner from 'react-native-loading-spinner-overlay';
import helpers from './helpers';
class AddEvent extends React.Component {
```

```
constructor(props){
  super(props);

  this.state = {
    title: '',
    description:'',
    fee:'',
    classes:'',
    venue:'',
    startdate:'',
    enddate:'',
    spinner: false
  }
}

componentDidMount(){
  this.getparamdata();
}

getparamdata = async () => {
  global.userIdToken = await this.props.navigation.state.params.addedBy;
  global.type = await this.props.navigation.state.params.type;
    // alert(`${global.userIdToken} and ${global.type}`)
}

showHideSpinner(visible) {
      this.setState({spinner: visible});
}

addevent = () => {

        return fetch(helpers.baseUrl+'event_list.php',{
          method: 'POST',
          headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json',
          },
          body: JSON.stringify({
            title: this.state.title,
            description: this.state.description,
            fee:this.state.fee,
            agegroup:this.state.classes,
            venue:this.state.venue,
            start_date:this.state.startdate,
            end_date:this.state.enddate
          })

        }).then((response) => response.json())
              .then((responseJson) => {
                 if(responseJson.status == 200){
                     this.showHideSpinner(false);
                     alert(`Event added sucessfully`);
                     this.props.navigation.push('Home',{
                         type:`${global.type}`,
                         userIdToken:`${global.userIdToken}`
                     })
                 }else{
                     this.showHideSpinner(false);
                     alert(responseJson.message);
                 }
              }).catch((error) => {
                alert(`server is down please try again`)
              });
}

render() {
  return (
    <View style = {{flex:1, marginTop: 20}}

    >
    <Spinner
          visible={this.state.spinner}
          overlayColor='rgba(0, 0, 0, 0.50)'
      />

      <View style={{flex:0.7 }}>
          <View style = {{flex:1.5, paddingTop: 2}}>

            <View style = {{flex: 1}}>
              <Image
                source = {require('./jpisLogo.jpeg')}
                style = {{height:'100%', width: '100%', resizeMode: 'contain'}}
              />

            </View>

          </View>
          <View style = {{flex:1, justifyContent: 'space-
around', alignItems: 'center', backgroundColor: 'white'}}>
```

```
              <Text style = {{fontSize: 26, color: 'black'}}>
                ADD EVENT
              </Text>

            </View>


          </View>
          <View style={{flex:2, paddingRight: 20, paddingLeft:20, flexDirection:
'column' }}>
            <View style = {{flex: 1, justifyContent: 'center'}}>
              <TextInput
              onChangeText = {title => this.setState({title})}
              style = {{height: 50, borderBottomColor: 'black', borderBottomWidth
: 1}}

              placeholder = 'EVENT TITLE'
              />
              <TextInput
              style = {{height:50, borderBottomColor: 'black', borderBottomWidth:
 1}}

              onChangeText = {description => this.setState({description})}
              placeholder = 'DESCRIPTION'

              />
              <TextInput
              style = {{height:50, borderBottomColor: 'black', borderBottomWidth:
 1}}

              onChangeText = {fee => this.setState({fee})}
              placeholder = 'FEE'

              />
              <TextInput
              style = {{height:50, borderBottomColor: 'black', borderBottomWidth:
 1}}

              onChangeText = {classes => this.setState({classes})}
              placeholder = 'CLASSES'

              />
              <TextInput
              style = {{height:50, borderBottomColor: 'black', borderBottomWidth:
 1}}

              onChangeText = {venue => this.setState({venue})}
              placeholder = 'VENUE'

              />
```

```
       <TextInput
       style = {{height:50, borderBottomColor: 'black', borderBottomWidth: 1}}
       onChangeText = {startdate => this.setState({startdate})}
       placeholder = 'START DATE'

       />
       <TextInput
       style = {{height:50, borderBottomColor: 'black', borderBottomWidth: 1}}
       onChangeText = {enddate => this.setState({enddate})}
       placeholder = 'END DATE'

       />

      </View>
      <View style = {{flex: 0.2, paddingLeft: 30, paddingRight: 30}}>
      <TouchableOpacity onPress={()=>this.addevent()} style = {{marginBottom: 5, flex: 1, justifyContent: 'center', alignItems: 'center', backgroundColor: 'skyblue', borderRadius: 20}}>
       <Text style = {{fontSize: 24}}>
        ADD
       </Text>

      </TouchableOpacity>
      </View>

     </View>

   </View>
  );
 }
}
export default AddEvent;
```

- **Add Teacher function on Teacher's home screen:**

```
import  * as React from 'react';
import { Text, View, StyleSheet,TextInput, Image, Button, ImageBackground, TouchableOpacity} from 'react-native';
import helpers from './helpers';
import Spinner from 'react-native-loading-spinner-overlay';

export default class AddTeacher extends React.Component {
```

```
constructor(props){
  super(props);

  this.state = {
    UserEmail: '',
    userFirstName:'',
    mobile:'',
    spinner: false
  }
}

componentDidMount(){
  this.getparamdata();
}

getparamdata = async () => {
  global.userIdToken = await this.props.navigation.state.params.addedBy;
  global.type = await this.props.navigation.state.params.type;
    // alert(`${global.userIdToken} and ${global.type}`)
}

showHideSpinner(visible) {
      this.setState({spinner: visible});
}

addTeach = () => {
  this.showHideSpinner(true);
  return fetch(helpers.baseUrl+'teacher_email.php',{
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      teacherEmail: this.state.UserEmail,
      addedBy: global.userIdToken

    })

  }).then((response) => response.json())
        .then((responseJson) => {
           if(responseJson.status == 200){
               this.showHideSpinner(false);
               alert(`Email added sucessfully`);
               this.props.navigation.push('Home',{
```

```
           type:`${global.type}`,
           userIdToken:`${global.userIdToken}`
         })
       }else{
         this.showHideSpinner(false);
         alert(responseJson.message);
       }
     }).catch((error) => {
      alert(`server is down please try again`)
     });
 }
 render() {
  return (
   <View style = {{flex:1}}

   >
   <Spinner
          visible={this.state.spinner}
          overlayColor='rgba(0, 0, 0, 0.50)'
        />

     <View style={{flex:1 }}>
      <View style = {{flex:1.5}}>

       <View style = {{flex: 1}}>
        <Image
         source = {require('./jpisLogo.jpeg')}
         style = {{height:'100%', width: '100%', resizeMode: 'contain'}}
        />

       </View>

      </View>
      <View style = {{flex:1, justifyContent: 'space-
around', alignItems: 'center', backgroundColor: 'white'}}>
       <Text style = {{fontSize: 30}}>
        Event Manager App
       </Text>

      </View>




     </View>
     <View style={{flex:2, paddingRight: 20, paddingLeft:20, flexDirection:
'column' }}>
     <Text style = {{fontSize: 22, color: 'orange'}}>
        ADD TEACHER
       </Text>
      <View style = {{flex: 1, justifyContent: 'center'}}>
       <TextInput
       style = {{height: 50, borderBottomColor: 'black', borderBottomWidth
: 1}}

       placeholder = 'Tecaher Name'
       />
       <TextInput
       style = {{height:50, borderBottomColor: 'black', borderBottomWidth:
 1}}

       onChangeText = {UserEmail => this.setState({UserEmail})}
       placeholder = 'E-mail'

       />
       <TextInput
       style = {{height:50, borderBottomColor: 'black', borderBottomWidth:
 1}}


       placeholder = 'PHONE NO.'

       />


      </View>
      <View style = {{flex: 0.2, paddingLeft: 30, paddingRight: 30}}>
       <TouchableOpacity onPress={()=>this.addTeach()} style = {{marginBot
tom: 5, flex: 1, justifyContent: 'center', alignItems: 'center', backgroundColor:
 'black', borderRadius: 20}}>
        <Text style = {{fontSize: 28, fontWeight: 'bold', color: 'white'}
}>

         ADD
        </Text>

       </TouchableOpacity>
      </View>

     </View>
```

```
      </View>
    );
  }
}
```

- **Checking if teacher's e-mail exists:**

```
<?php

// Importing DBConfig.php file.
include 'dbconfig.php';

// Creating connection.
$con = mysqli_connect($HostName,$HostUser,$HostPass,$DatabaseName);

// Getting the received JSON into $json variable.
$json = file_get_contents('php://input');

// decoding the received JSON and store into $obj variable.
$obj = json_decode($json,true);

 // Populate User name from JSON $obj array and store into $name.
 $teacherEmail = $obj['teacherEmail'];
 $addedBy = $obj['addedBy'];


//Checking Email is already exist or not using SQL query.
$CheckSQL = "SELECT * FROM teacher_email WHERE teacherEmail='$teacherEmail'";

// Executing SQL Query.
$check = mysqli_fetch_array(mysqli_query($con,$CheckSQL));


if(isset($check)){

 $EmailExistMSG = ['status'=>'500','message'=>'Teacher id already Exist, Please T
ry Again !!!'];

 // Converting the message into JSON format.
$EmailExistJson = json_encode($EmailExistMSG);

// Echo the message.
 echo $EmailExistJson ;

 }
 else{
    $CheckTeacherSql = "SELECT type FROM user WHERE id='$addedBy'";
    $CheckTeacher = mysqli_fetch_array(mysqli_query($con,$CheckTeacherSql));
    $is_teacher = $CheckTeacher['type'];
    if($is_teacher == 2){
        $Sql_Query = "insert into teacher_email (teacherEmail,addedBy) values ('$teacherEmail','$addedBy')";
        if(mysqli_query($con,$Sql_Query)){

            // If the record inserted successfully then show the message.
           $MSG = ['status'=>'200','message'=>'Email Added Successfully'];

           // Converting the message into JSON format.
           $json = json_encode($MSG);

           // Echo the message.
            echo $json ;

        }else{
            echo 'Try Again';
        }
    }else{
        $MSG = ['status'=>'500','message'=>'You are not allowed to make it as teacher'];


        $json = json_encode($MSG);
        echo $json;
    }
}
 mysqli_close($con);
?>
```

Selection screen (Approve, Disapprove function):

- `import * as React from 'react';`
- `import { Text, View, StyleSheet,TextInput, Image, Button, ImageBackground, TouchableOpacity, ScrollView} from 'react-native';`
- `import helpers from './helpers';`

```
-
- export default class selection extends React.Component {
-   constructor(props){
-     super(props);
-
-     this.state = {
-       UserEmail: '',
-       userFirstName:'',
-       mobile:'',
-       stud_class:'',
-       section:'',
-       reason_of_intrest:'',
-       prev_achi:'',
-       appr_status:'0',
-       eventName:'',
-       eventDesc:'',
-       eventStartdt:'',
-       eventEnddt:'',
-
-     }
-   }
-
-   componentDidMount(){
-
-     this.getparamData();
-
-   }
-
-   getparamData =  async () => {
-     const { navigation } = this.props;
-     global.name = await navigation.getParam('name');
-     global.description = await navigation.getParam('description');
-     global.startdate = await navigation.getParam('startdate');
-     global.enddate = await navigation.getParam('enddate');
-     global.username = await navigation.getParam('username');
-     global.email = await navigation.getParam('email');
-     global.class = await navigation.getParam('class');
-     global.phone_no = await navigation.getParam('phone_no');
-     global.intrest_reason = await navigation.getParam('intrest_reason');
-     global.previous_achievement = await navigation.getParam('previous_achi
  evement');
-
-     // alert(`${global.name} ${global.description} ${global.startdate} ${g
  lobal.enddate}`);
```

```
    alert(`${global.username} ${global.email} ${global.class} ${global.phone_no} ${global.intrest_reason}`)

    this.setState({
      eventName:global.name,
      eventDesc:global.description,
      eventStartdt:global.startdate,
      eventEnddt:global.enddate,
    })

  }

  changeAppr = (status) => {
    this.props.navigation.navigate('Home')
    return false;
    return fetch(helpers.baseUrl+'user_registration.php',{
      method: 'POST',
      headers: {
          'Accept': 'application/json',
          'Content-Type': 'application/json',
      },
      body: JSON.stringify({
          appr_status: this.state.appr_status

      })

      }).then((response) => response.json())
          .then((responseJson) => {
              if(responseJson){
                  alert(JSON.stringify(responseJson));
                  return false;
                  // AsyncStorage.setItem('user', JSON.stringify(responseJson.data.id))
                  // AsyncStorage.setItem('type', JSON.stringify(responseJson.data.type))
                  this.props.navigation.navigate('HomeAppScreen');
              }else{
                alert(JSON.stringify(responseJson))
              }
          }).catch((error) => {
              // this.showHideSpinner(false);
              alert(error.toString());
          });

  }
```

```
-
-     render() {
-
-       let addteacher = (
-         <View style = {{marginBottom: 10, flex: 0.115, paddingLeft: 5, paddi
   ngRight: 5, flexDirection: 'row'}}>
-           <TouchableOpacity onPress={()=>this.changeAppr('1')} style = {{mar
   ginBottom: 5, flex: 1, justifyContent: 'center', alignItems: 'center', bac
   kgroundColor: 'green', borderRadius: 20, paddingRight: 5}}>
-             <Text style = {{fontSize: 26}}>
-               APPROVE
-             </Text>
-
-           </TouchableOpacity>
-           <TouchableOpacity onPress={()=>this.changeAppr('2')} style = {{mar
   ginBottom: 5, flex: 1, justifyContent: 'center', alignItems: 'center', bac
   kgroundColor: 'red', borderRadius: 20, paddingLeft: 10 }}>
-             <Text style = {{fontSize: 26}}>
-               DISAPPROVE
-             </Text>
-
-           </TouchableOpacity>
-         </View>
-       );
-
-       return (
-
-         <View style = {{}}>
-           <ScrollView>
-             <View style = {{ height: 220}}>
-               <View style = {{backgroundColor: 'black', flex: 1}}>
-                 <View style = {{flex:1, backgroundColor: 'black', justifyCon
   tent: 'center', alignItems: 'center'}}>
-                   <Text style = {{fontSize: 40, fontWeight: 'bold', color: '
   white'}}>
-
-                   </Text>
-                 </View>
-
-
-
-               </View>
-               <View style = {{ backgroundColor: 'grey', flex:1.5 }}>
-                 <Text style = {{fontSize: 18, paddingTop : 10, paddingLeft:
   5, color: 'black'}}>
```

```
-                  DETAILS:-
-                </Text>
-                <Text style = {{fontSize: 18, paddingTop : 10, paddingLeft: 5, color: 'black'}}>
-                  {global.description}
-                </Text>
-                <Text style = {{fontSize: 18, paddingTop : 10, paddingLeft: 5, color: 'black'}}>
-                  {global.startdate}
-                </Text>
-                <Text style = {{fontSize: 18, paddingTop : 10, paddingLeft: 5, color: 'black'}}>
-                  {global.enddate}
-                </Text>
-              </View>
-            </View>
-            <View style = {{ height: 800}}>
-              <View style = {{flex:0.13, backgroundColor: 'white', justifyContent: 'center', alignItems : 'center'}}>
-                <Text style = {{fontSize: 38, color: 'black'}}>
-                  FORM Values
-                </Text>
-              </View>
-              <View style = {{flex: 1, justifyContent: 'space-evenly', paddingLeft: 10, paddingRight: 10, backgroundColor: 'white'}}>
-                <Text
-                style = {{height: 40, borderBottomColor: 'black', borderBottomWidth: 1}}
-                >
-                 Name :- Viraj
-                </Text>
-
-                <Text
-                style = {{height: 40, borderBottomColor: 'black', borderBottomWidth: 1}}
-                >
-                 Email :-  virajvijaywargiya@yahoo.com
-                </Text>
-
-                <Text
-                style = {{height: 40, borderBottomColor: 'black', borderBottomWidth: 1}}
-                >
-                  Phone :- 7568507629
-                </Text>
```

```
-
-        <Text
-        style = {{height: 40, borderBottomColor: 'black', borderBott
  omWidth: 1}}
-        >
-         Class :-  12
-        </Text>
-
-        <Text
-        style = {{height: 40, borderBottomColor: 'black', borderBott
  omWidth: 1}}
-        >
-         Section :-  A
-        </Text>
-
-        <Text
-        style = {{height: 40, borderBottomColor: 'black', borderBott
  omWidth: 1}}
-        >
-         Reason :-  I like to participate in MUNs
-        </Text>
-
-        <Text
-        style = {{height: 40, borderBottomColor: 'black', borderBott
  omWidth: 1}}
-        >
-         Previous Aachievements :-  none
-        </Text>
-
-       </View>
-       {
-        (this.state.appr_status == '0') ? addteacher : null
-       }
-      </View>
-     </ScrollView>
-    </View>
-
-   );
-  }
- }
```

- **Navigation: (Sign-in screen)**

```
import React, { Component } from 'react';
import { View,Text,TextInput,TouchableOpacity,Image,Alert } from 'react-native';
import AsyncStorage from '@react-native-community/async-storage';
import helpers from './to_be_sent/screens/helpers';
import Spinner from 'react-native-loading-spinner-overlay';


class SigninScreen extends Component {

    constructor(props){
        super(props);

        this.state = {
          UserEmail: '',
          userPassword: '',
          modalVisible: false,
          spinner: false
        }
    }

    showHideSpinner(visible) {
        this.setState({spinner: visible});
    }
    navigatetohome = () =>{

        const { UserEmail }  = this.state ;
        const { userPassword }  = this.state ;

        this.showHideSpinner(true);
        return fetch(helpers.baseUrl+'user_Login.php',{
         method: 'POST',
         headers: {
           'Accept': 'application/json',
           'Content-Type': 'application/json',
         },
         body: JSON.stringify({

           email: UserEmail,

           password: userPassword

         })

       }).then((response) => response.json())
             .then((responseJson) => {
                if(responseJson.status == 200){
                    AsyncStorage.setItem('user',responseJson.data.id )
                    AsyncStorage.setItem('user_type', responseJson.data.type )
                    this.showHideSpinner(false);
                    this.props.navigation.navigate('Home',{
                        type:`${responseJson.data.type}`,
                        userIdToken:`${responseJson.data.id}`
                    })
                }else{
                    this.showHideSpinner(false);
                    alert(responseJson.message);
                }
             }).catch((error) => {
               alert(`server is down please try again`)
             });

         }


    goToSignup = () => {
        this.props.navigation.navigate('SignUpChoice');
    }
    render(){
        return (
            <View style = {{flex:1, marginTop: 20, backgroundColor: 'white'}}>
            <Spinner
                    visible={this.state.spinner}
                    overlayColor='rgba(0, 0, 0, 0.50)'
                />
                <View style={{flex:2 }}>
                    <View style = {{flex:1.5, paddingTop: 20}}>

                    <View style = {{flex: 1}}>
                        <Image
                        source = {require('./jpisLogo.jpeg')}
                        style = {{height:'100%', width: '100%', resizeMode: 'cont
ain'}}

                        />

                    </View>

                    </View>
                    <View style = {{flex:1, justifyContent: 'space-
around', alignItems: 'center', backgroundColor: 'white'}}>
                    <Text style = {{fontSize: 30}}>
```

```
                        Event Manager App
                    </Text>

                    </View>


                </View>
                <View style={{flex:2, paddingRight: 20, paddingLeft:20}}>
                    <View style = {{flex: 1}}>
                    <TextInput
                    style = {{height: 80, borderBottomColor: 'black', borderBotto
mWidth: 1}}
                    onChangeText = {UserEmail => this.setState({UserEmail})}
                    placeholder = 'ENTER E-MAIL'
                    />
                    <TextInput
                    style = {{height:80, borderBottomColor: 'black', borderBottom
Width: 1}}
                    onChangeText = {userPassword => this.setState({userPassword})
}
                    placeholder = 'ENTER PASSWORD'
                    secureTextEntry = {true}
                    />
                    </View>
                    <TouchableOpacity onPress={() => this.navigatetohome()} style
 = {{flex: 0.4, justifyContent: 'center', alignItems: 'center', backgroundColor:
'skyblue', borderRadius: 30}}>
                    <Text style = {{fontSize: 20,fontWeight:'bold'}}>
                        LOG IN
                    </Text>
                    <View style={{height:20}} />
                    </TouchableOpacity>


                </View>
                <View style={{flex:1}}>
                <TouchableOpacity onPress = {() => this.props.navigation.push('Fo
rgotpassword')} style={{flex:1, alignItems: 'flex-
end', justifyContent: 'center', paddingRight: 26}}>
                    <Text style = {{fontSize: 16, color: 'red'}}>
                    Forgot Password?
                    </Text>
                </TouchableOpacity>
                <View style={{flexDirection: 'row',flex:2}}>
```

```
                    <View style = {{flex: 2, justifyContent: 'center', alignItems
: 'flex-end'}}>
                    <Text style = {{fontSize: 20}}>
                        Don't have an account?
                    </Text>
                    </View>
                    <TouchableOpacity onPress = {() => this.goToSignup()} style =
 {{flex: 1, justifyContent: 'center', alignItems: 'flex-start',paddingLeft:5}}>
                    <Text style = {{fontSize: 20,fontWeight:'bold'}}>
                        Sign Up
                    </Text>
                    </TouchableOpacity>
                    </View>
                </View>
            </View>
        )
    }
}
export default SigninScreen;
```

- **Uploading image on screen:**

```
render() {
    return (
      <View style = {{flex:1, marginTop: 20, backgroundColor: 'white'}}>


          <View style={{flex:2 }}>
            <View style = {{flex:1.5, paddingTop: 20}}>

              <View style = {{flex: 1}}>
                <Image
                  source = {require('./jpisLogo.jpeg')}
                  style = {{height:'100%', width: '100%', resizeMode: 'contain'}}
                />

              </View>

            </View>
```

- **Mail (SMTP) function:**

```
<?php

// Import PHPMailer classes into the global namespace
// These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require './src/Exception.php';
require './src/PHPMailer.php';
require './src/SMTP.php';

// Load Composer's autoloader
// require 'vendor/autoload.php';

// Instantiation and passing `true` enables exceptions
$mail = new PHPMailer(true);

try {
    //Server settings
    // $mail-
>SMTPDebug = 2;                                       // Enable verbose debug out
put
    $mail-
>isSMTP();                                            // Set mailer to use SMTP
    $mail-
>Host       = 'smtp.gmail.com';  // Specify main and backup SMTP servers
    $mail-
>SMTPAuth   = true;                                   // Enable SMTP authenticati
on
    $mail-
>Username   = 'virox0220@gmail.com';                     // SMTP username
    $mail-
>Password   = 'virajrahil02';                               // SMTP password
    $mail-
>SMTPSecure = 'ssl';                                  // Enable TLS encryption, `
ssl` also accepted
    $mail-
>Port       = 465;                                    // TCP port to connect to

    //Recipients
    $mail->setFrom('virox0220@gmail.com', 'Mailer');
    $mail-
>addAddress('rahiljain.intelligence@gmail.com');               // Name is optiona
l
    $mail->addReplyTo('info@example.com', 'Information');
    // Content
    $mail-
>isHTML(true);                                  // Set email format to HTML
    $mail->Subject = 'Here is the subject';
    $mail->Body    = 'This is the HTML message body <b>in bold!</b>';
    $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

    $mail->send();
    echo 'Message has been sent';
} catch (Exception $e) {
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
```

- **Showing events:**

```
<?php

include 'dbconfig.php';
$con = mysqli_connect($HostName,$HostUser,$HostPass,$DatabaseName);
$json = file_get_contents('php://input');
$obj = json_decode($json,true);


$type = $obj['type'];
$todayDate = '2019-09-10';
if($type == 'student'){
    $Sql_Query = "select * from events where start_date > '$todayDate'";
}else{
    $Sql_Query = "select * from events";
}
$check = mysqli_fetch_array(mysqli_query($con,$Sql_Query));
if(isset($check)){
    $SuccessLoginMsg = ['status'=>'200','message'=>'Results found','data'=>$check
];
    $SuccessLoginJson = json_encode($SuccessLoginMsg);
    echo $SuccessLoginJson ;
}else{
  $InvalidMSG = ['status'=>'500','message'=>'No result found'];
  $InvalidMSGJSon = json_encode($InvalidMSG);
  echo $InvalidMSGJSon ;
}

mysqli_close($con);
?>
```

- **Home screen (displaying events):**

```
import * as React from 'react';
import { Text, View, StyleSheet,TextInput, Image, Button, ImageBackground, TouchableOpacity,FlatList} from 'react-native';

export class Home extends React.Component {
 constructor(props){
  super(props);
  this.state = {
   dummyText:'EVENTS'
  }
 }
 static navigationOptions = {
  header:null
 }
 navigationtoform = (item) => {
  this.setState({
   dummyText:'hello'
  })
  return false;
  this.props.navigation.push('form',{
    name:item.eventname,
    description:item.description,
    startdate:item.startdate,
    enddate:item.enddate
   })
 }
 render() {
  return (
    <View style = {{flex: 1, backgroundColor: 'black', marginTop: 22}}>
    <Text style={{paddingTop: 5, color:'#FFF', fontWeight: 'bold', fontSize: 20, alignItems: 'center'}}>{this.state.dummyText}</Text>
  <View style={{paddingTop:5}}>
```

```
   <FlatList
 data={
    [
     {eventname: 'MUN',fees:'1500',startdate:'June 15,2019',enddate:'June 21,2019', description: 'this is an mun event', venue: 'jpis', agegroup: '9th-12th'},
     {eventname: 'DANCE',fees:'1200',startdate:'June 25,2019',enddate:'July 02,2019', description: 'this is a dance event', venue: 'jpis', agegroup: '9th-12th'},
     {eventname: 'FLAIRFEST', fees: '0', startdate: 'july 6, 2019', enddate: 'july 20, 2019', description: 'this is a falirfest event', venue: 'jpis', agegroup: '9th-12th'}

    ]
    }
 renderItem={
   ({item}) => <TouchableOpacity onPress={() => this.navigationtoform(item)} style = {{flex: 1, backgroundColor: 'white', marginTop: 10, marginLeft: 10, marginRight: 10, height: 250}}>

     <ImageBackground style = {{flex: 1, flexDirection: 'column'}}
     source = {item.image}

    >
      <View style = {{flex: 1,  justifyContent: 'center', alignItems: 'center'}}>
       <Text style = {{fontSize: 36, fontWeight: 'bold', color: 'black'}}>
        {item.eventname}
       </Text>
      </View>
      <View style = {{flex: 3,  flexDirection: 'row'}}>
       <View style = {{flex:1, borderRightWidth: 1, borderRightColor: 'black'}}>
        <Text style = {{fontSize: 16, fontWeight: 'bold', color: 'black'}}>
         DESCRIPTION:-
         {item.description}
        </Text>
       </View>
       <View style = {{flex: 1, justifyContent: 'space-around'}}>
        <Text style = {{fontSize: 18, paddingLeft: 10,  fontWeight: 'bold', color: 'black'}}>
         FEE - {item.fees}
        </Text>
```

```
                <Text style = {{fontSize: 18, paddingLeft: 10,  fontWeight: 'bold
', color: 'black'}}>
                  VENUE - {item.venue}
                </Text>
                <Text style = {{fontSize: 18, paddingLeft: 10,  fontWeight: 'bold
', color: 'black'}}>
                  AGE GROUP - {item.agegroup}
                </Text>
              </View>
            </View>
            <View style = {{flex: 1, flexDirection: 'row'}}>
              <View style = {{flex: 1, justifyContent: 'center', alignItems: 'fle
x-start'}}>
                <Text style = {{fontSize: 14,  fontWeight: 'bold', color: 'black'
}}>
                  start date- {item.startdate}
                </Text>
              </View>

              <View style = {{flex: 1, justifyContent: 'center', alignItems: 'cen
ter'}}>
                <Text style = {{fontSize: 14,  fontWeight: 'bold', color: 'black'
}}>
                   end date - {item.enddate}
                </Text>
              </View>


            </View>
          </ImageBackground>
        </TouchableOpacity>
      }
      ItemSeparatorComponent={this.renderSeparator}
      keyExtractor={(item,index) => index.toString()}
/>
</View>
</View>
    );
  }
}
```