



React Native Insert Update Display Delete CRUD Operations MySQL DB

admin December 13, 2017 React Native

This tutorial is the fully advanced tutorial in react native because in this tutorial we would going to implement CRUD operations in react native android iOS applications. CRUD operations consist four type of major functions Add, Edit(Modify), Delete and View. So we would going to create a react native app with Insert Update Display Delete CRUD Operations From MySQL database using PHP.

What we are doing in this project :

We are creating a react native app with **react-navigation** library because without the react-navigation we cannot add multiple activities in our project. Now would insert student records in online MySQL database using PHP from our react native application. After that we would retrieve all those records into ListView in next activity. Now we would open the selected record into next activity and automatically fill each data into TextInput. We would also gives the option to delete the current selected data. So let's get started 😊 .

Project Description File Structure:

PHP Files in this project :

1. DBConfig.php
2. InsertStudentData.php
3. ShowAllStudentsList.php
4. UpdateStudentRecord.php
5. DeleteStudentRecord.php

Activity classes in this project created in App.js file

1. MainActivity
2. ShowStudentListActivity
3. EditStudentRecordActivity

Contents in this project Insert Update Display Delete CRUD Operations From MySQL DB in react native using PHP :

1. Create a fresh MySQL database in your online hosting server.
2. Create a Table named as **StudentDetailsTable** with 5 columns **student_id**, **student_name**, **student_class**, **student_phone_number** and **student_email**. Now we would make the **student_id** as **Primary key** with **Auto Increment**. See the below screenshot for more details. This would be our main table and we would perform all CRUD operations on this table from our MySQL database.

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	Auto Increment
student_id	INT	255	None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
student_name	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
student_class	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
student_phone_num	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>
student_email	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>

3. Create 5 PHP Script files to Receive and Insert, Display, Update and Delete data from MySQL database from React Native app:

We would make 5 PHP files named as **DBConfig.php** , **InsertStudentData.php** , **ShowAllStudentsList.php** , **UpdateStudentRecord.php** and **DeleteStudentRecord.php** . You need to upload all these files to your hosting server.

- DBConfig.php** : This file consist all the MySQL database connection configuration like MySQL database host name, MySQL database name, MySQL database username, MySQL database password.
- InsertStudentData.php** : This file would receive the sent data from react native app and insert that data into MySQL table.
- ShowAllStudentsList.php** : This file would convert all the existing present MySQL database data into JSON format.
- UpdateStudentRecord.php** : This file would update the existing opened record in app.
- DeleteStudentRecord.php** : This file would delete the existing open record in react native app.

1. Code for DBConfig.php file :

```

1 <?php
2
3 //Define your host here.
4 $HostName = "localhost";
5
6 //Define your database name here.
7 $DatabaseName = "id2070055_reactnatedb";
8
9 //Define your database username here.
10 $HostUser = "id2070055_reactnatedb_user";
11
12 //Define your database password here.
13 $HostPass = "1234567890";
14
15 ?>

```

2. Code for InsertStudentData.php file :

```

1 <?php
2
3 // Importing DBConfig.php file.
4 include 'DBConfig.php';
5
6 // Connecting to MySQL Database.
7 $con = mysqli_connect($HostName,$HostUser,$HostPass,$DatabaseName);
8
9 // Getting the received JSON into $json variable.
10 $json = file_get_contents('php://input');
11
12 // decoding the received JSON and store into $obj variable.
13 $obj = json_decode($json,true);
14
15 // Populate Student name from JSON $obj array and store into $$Name.
16 $$Name = $obj['student_name'];

```

```

17
18 // Populate Student Class from JSON $obj array and store into $$_Class.
19 $$_Class = $obj['student_class'];
20
21 // Populate Student Phone Number from JSON $obj array and store into $$_Phone_Number.
22 $$_Phone_Number = $obj['student_phone_number'];
23
24 // Populate Email from JSON $obj array and store into $$_Email.
25 $$_Email = $obj['student_email'];
26
27 // Creating SQL query and insert the record into MySQL database table.
28 $Sql_Query = "insert into StudentDetailsTable (student_name,student_class,student_phone_number,student_email) values ('$$_Name','$$_Class','$$_Phone_Number','$$_Email')";
29
30 if(mysqli_query($con,$Sql_Query)){
31
32 // If the record inserted successfully then show the message.
33 $MSG = 'Record Successfully Inserted Into MySQL Database.' ;
34
35 // Converting the message into JSON format.
36 $json = json_encode($MSG);
37
38 // Echo the message.
39 echo $json ;
40
41 }
42 else{
43
44 echo 'Try Again';
45
46 }
47
48 mysqli_close($con);
49 ?>

```

3. Code for ShowAllStudentsList.php file :

```

1 <?php
2 include 'DBConfig.php';
3
4 // Create connection
5 $conn = new mysqli($HostName, $HostUser, $HostPass, $DatabaseName);
6
7 if ($conn->connect_error) {
8
9 die("Connection failed: " . $conn->connect_error);
10 }
11
12 // Creating SQL command to fetch all records from Table.
13 $sql = "SELECT * FROM StudentDetailsTable";
14
15 $result = $conn->query($sql);
16
17 if ($result->num_rows > 0) {
18
19 while($row[] = $result->fetch_assoc()) {
20
21 $item = $row;
22
23 $json = json_encode($item);
24
25 }
26
27 } else {
28 echo "No Results Found.";
29 }
30
31 echo $json;
32 $conn->close();
33 ?>

```

4. Code for UpdateStudentRecord.php file :

```

1 <?php
2
3 // Importing DBConfig.php file.
4 include 'DBConfig.php';
5
6 // Connecting to MySQL Database.
7 $con = mysqli_connect($HostName,$HostUser,$HostPass,$DatabaseName);
8
9 // Getting the received JSON into $json variable.
10 $json = file_get_contents('php://input');
11
12 // decoding the received JSON and store into $obj variable.
13 $obj = json_decode($json,true);
14
15 // Populate Student ID from JSON $obj array and store into $$_Name.
16 $$_ID = $obj['student_id'];
17
18 // Populate Student name from JSON $obj array and store into $$_Name.
19 $$_Name = $obj['student_name'];
20
21 // Populate Student Class from JSON $obj array and store into $$_Class.
22 $$_Class = $obj['student_class'];
23
24 // Populate Student Phone Number from JSON $obj array and store into $$_Phone_Number.
25 $$_Phone_Number = $obj['student_phone_number'];
26

```

```

27 // Populate Email from JSON $obj array and store into $$Email.
28 $$Email = $obj['student_email'];
29
30 // Creating SQL query and insert the record into MySQL database table.
31 $Sql_Query = "UPDATE StudentDetailsTable SET student_name= '$S_Name', student_class = '$S_Class', student_phone_number = '$S_Phone_Numb
32
33 if(mysqli_query($con,$Sql_Query)){
34
35 // If the record inserted successfully then show the message.
36 $MSG = 'Record Successfully Inserted Into MySQL Database.' ;
37
38 // Converting the message into JSON format.
39 $json = json_encode($MSG);
40
41 // Echo the message.
42 echo $json ;
43
44 }
45 else{
46
47 echo 'Try Again';
48
49 }
50 mysqli_close($con);
51 ?>

```

5. Code for DeleteStudentRecord.php file :

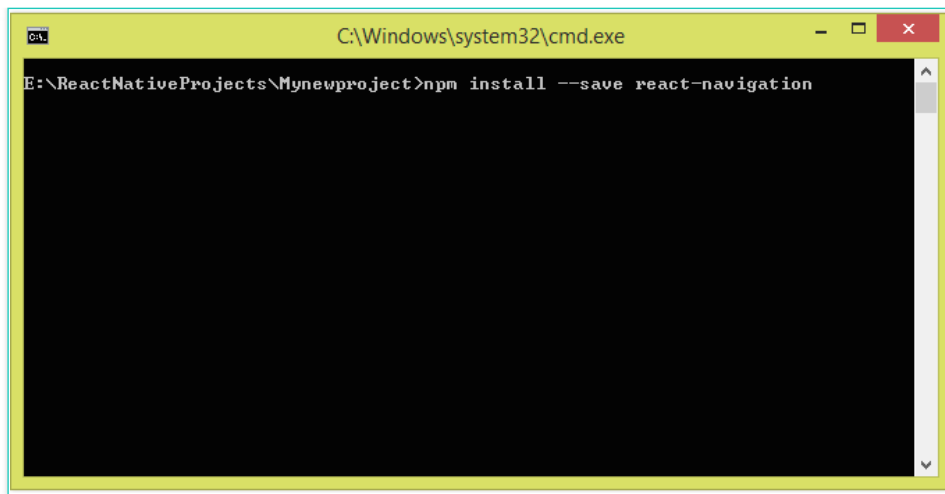
```

1 <?php
2
3 // Importing DBConfig.php file.
4 include 'DBConfig.php';
5
6 // Connecting to MySQL Database.
7 $con = mysqli_connect($HostName,$HostUser,$HostPass,$DatabaseName);
8
9 // Getting the received JSON into $json variable.
10 $json = file_get_contents('php://input');
11
12 // decoding the received JSON and store into $obj variable.
13 $obj = json_decode($json,true);
14
15 // Populate Student ID from JSON $obj array and store into $$ID.
16 $$ID = $obj['student_id'];
17
18 // Creating SQL query and Updating the current record into MySQL database table.
19 $Sql_Query = "DELETE FROM StudentDetailsTable WHERE student_id = '$S_ID' " ;
20
21
22 if(mysqli_query($con,$Sql_Query)){
23
24 // If the record inserted successfully then show the message.
25 $MSG = 'Record Deleted Successfully.' ;
26
27 // Converting the message into JSON format.
28 $json = json_encode($MSG);
29
30 // Echo the message.
31 echo $json ;
32
33 }
34 else{
35
36 echo 'Try Again';
37
38 }
39 mysqli_close($con);
40 ?>

```

4. Start a new react native project or open your existing project's **App.js** file.

5. Open your project's folder in Command Prompt / Terminal and install **react-navigation** library. by typing `npm install --save react-navigation` command. You can read my this [tutorial to understand how activity structure would works](#).



6. Import **StyleSheet, View, Alert, TextInput, Button, Text, Platform, TouchableOpacity, ListView** and **ActivityIndicator** component in your project.

```
1 import React, { Component } from 'react';
2
3 import { StyleSheet, View, Alert, TextInput, Button, Text, Platform, TouchableOpacity, ListView, ActivityIndicator } from 'react-native';
```

7. Import **StackNavigator** from **react-navigation** library.

```
1 import { StackNavigator } from 'react-navigation';
```

8. Create a class named as **MainActivity** in your project. This would be your first screen of app with User registration from.

navigationOptions : Used to set the Activity title which display inside the Action Title bar.

constructor() : Creating 4 States inside the constructor() named as TextInput_Student_Name, TextInput_Student_Class, TextInput_Student_PhoneNumber, TextInput_Student_Email . Each state is used to hold the entered value from TextInput.

InsertStudentRecordsToServer() : This function would run the react native's **fetch()** web api and send the all TextInput entered value to server and after successfully insertion it will show the response message coming from server into Alert dialog box.

GoTo_Show_StudentList_Activity_Function() : In this function we would open the **ShowStudentListActivity** represent as Second activity.

```
1 class MainActivity extends Component {
2
3   static navigationOptions =
4   {
5     title: 'MainActivity',
6   };
7
8   constructor(props) {
9
10    super(props)
11
12    this.state = {
13
14      TextInput_Student_Name: '',
15      TextInput_Student_Class: '',
16      TextInput_Student_PhoneNumber: '',
17      TextInput_Student_Email: '',
18    }
19  }
20
21 }
22
23 InsertStudentRecordsToServer = () =>{
24
25   fetch('https://reactnativecode.000webhostapp.com/Student/InsertStudentData.php', {
26     method: 'POST',
27     headers: {
28       'Accept': 'application/json',
29       'Content-Type': 'application/json',
30     },
31     body: JSON.stringify({
32
33       student_name : this.state.TextInput_Student_Name,
34       student_class : this.state.TextInput_Student_Class,
35       student_phone_number : this.state.TextInput_Student_PhoneNumber,
36       student_email: this.state.TextInput_Student_Email
37     })
38   })
39   .then((response) => response.json())
40   .then((responseJson) => {
```

```

45
46      // Showing response message coming from server after inserting records.
47      Alert.alert(responseJson);
48
49      }).catch((error) => {
50          console.error(error);
51      });
52  }
53 }
54
55 GoTo_Show_StudentList_Activity_Function = () =>
56 {
57     this.props.navigation.navigate('Second');
58 }
59
60
61 render() {
62     return (
63
64 <View style={styles.MainContainer}>
65
66
67     <Text style={{fontSize: 20, textAlign: 'center', marginBottom: 7}}> Student Registration Form </Text>
68
69     <TextInput
70
71         placeholder="Enter Student Name"
72
73         onChangeText={ TextInputValue => this.setState({ TextInput_Student_Name : TextInputValue }) }
74
75         underlineColorAndroid='transparent'
76
77         style={styles.TextInputStyleClass}
78     />
79
80     <TextInput
81
82         placeholder="Enter Student Class"
83
84         onChangeText={ TextInputValue => this.setState({ TextInput_Student_Class : TextInputValue }) }
85
86         underlineColorAndroid='transparent'
87
88         style={styles.TextInputStyleClass}
89     />
90
91     <TextInput
92
93         placeholder="Enter Student Phone Number"
94
95         onChangeText={ TextInputValue => this.setState({ TextInput_Student_PhoneNumber : TextInputValue }) }
96
97         underlineColorAndroid='transparent'
98
99         style={styles.TextInputStyleClass}
100    />
101
102    <TextInput
103
104        placeholder="Enter Student Email"
105
106        onChangeText={ TextInputValue => this.setState({ TextInput_Student_Email : TextInputValue }) }
107
108        underlineColorAndroid='transparent'
109
110        style={styles.TextInputStyleClass}
111    />
112
113    <TouchableOpacity activeOpacity = { .4 } style={styles.TouchableOpacityStyle} onPress={this.InsertStudentRecordsToServer} >
114
115        <Text style={styles.TextStyle}> INSERT STUDENT RECORD TO SERVER </Text>
116
117    </TouchableOpacity>
118
119    <TouchableOpacity activeOpacity = { .4 } style={styles.TouchableOpacityStyle} onPress={this.GoTo_Show_StudentList_Activity_Functi
120
121        <Text style={styles.TextStyle}> SHOW ALL INSERTED STUDENT RECORDS IN LISTVIEW </Text>
122
123    </TouchableOpacity>
124
125 </View>
126
127 );
128 }
129 }
130 }

```

Screenshot of MainActivity :

9. Create class named as **ShowStudentListActivity**. This class would display all the entered students records in **ListView**.

constructor() : We are creating a Boolean type State named as **isLoading**. This state is used to show and hide the ActivityIndicator after done loading JSON data into ListView. If the isLoading value is true then it will show the ActivityIndicator and if the isLoading value is false then it will hide the ActivityIndicator.

navigationOptions : Used to set the Activity Title shows inside the Title Action bar.

componentDidMount() : This is a inbuilt function of React Native. This function would automatically calls on activity start up time. We would write the **Fetch()** Web code inside componentDidMount() method so when the this activity opens then it will start a web call and fetch the JSON data from server and display inside ListView. We are only displaying Student Names inside this ListView.

GetStudentIDFunction() : This function is very important for this class because in this function we would send the student_id, student_name, student_class, student_phone_number and student_email address to next EditStudentRecordActivity on ListView item onPress event. When use clicks on any ListView item then it will automatically get all the values form current selected record and navigate to next activity where it will be automatically received and filled into TextInput.

ListViewItemSeparator() : This function would render a separator line between each item of ListView.

```

1  class ShowStudentListActivity extends Component {
2
3      constructor(props) {
4
5          super(props);
6
7          this.state = {
8
9              isLoading: true
10
11          }
12      }

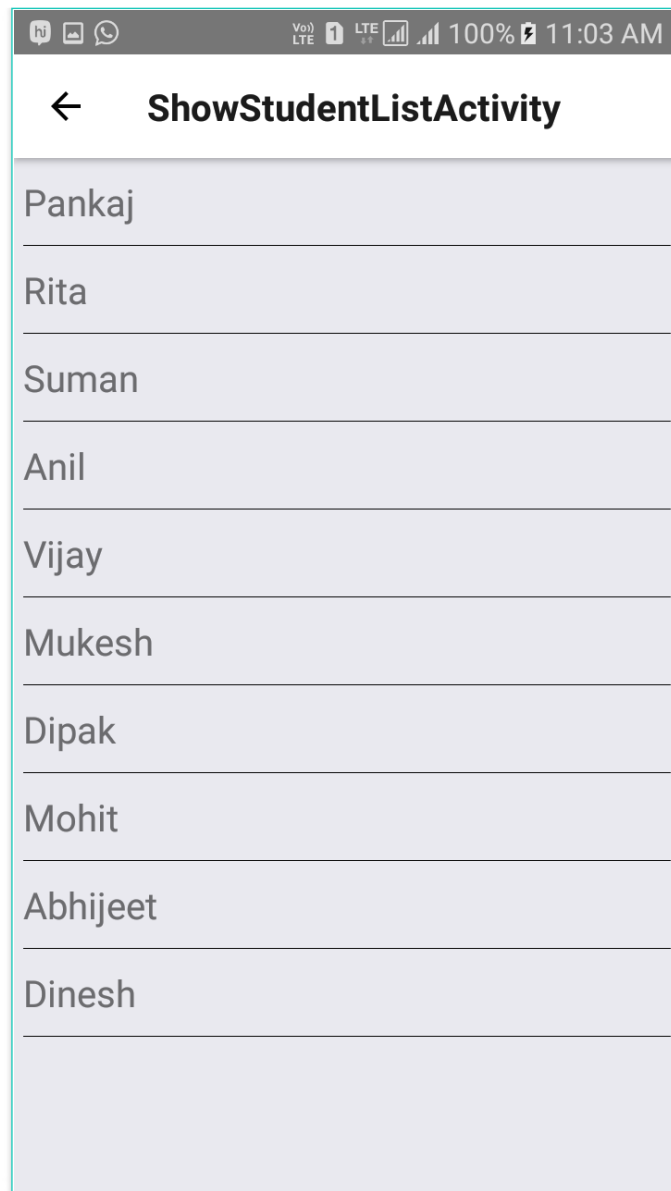
```

```

13
14 static navigationOptions =
15 {
16   title: 'ShowStudentListActivity',
17 };
18
19 componentDidMount() {
20
21   return fetch('https://reactnativecode.000webhostapp.com/Student/ShowAllStudentsList.php')
22     .then((response) => response.json())
23     .then((responseJson) => {
24       let ds = new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2});
25       this.setState({
26         isLoading: false,
27         dataSource: ds.cloneWithRows(responseJson),
28       }, function() {
29         // In this block you can do something with new state.
30       });
31     })
32     .catch((error) => {
33       console.error(error);
34     });
35 }
36
37 GetStudentIDFunction=(student_id,student_name, student_class, student_phone_number, student_email)=>{
38
39   this.props.navigation.navigate('Third', {
40
41     ID : student_id,
42     NAME : student_name,
43     CLASS : student_class,
44     PHONE_NUMBER : student_phone_number,
45     EMAIL : student_email
46
47   });
48
49 }
50
51 ListViewItemSeparator = () => {
52   return (
53     <View
54       style={{
55         height: .5,
56         width: "100%",
57         backgroundColor: "#000",
58       }}
59     />
60   );
61 }
62
63 render() {
64   if (this.state.isLoading) {
65     return (
66       <View style={{flex: 1, paddingTop: 20}}>
67         <ActivityIndicator />
68       </View>
69     );
70   }
71
72   return (
73     <View style={styles.MainContainer_For_Show_StudentList_Activity}>
74
75       <ListView
76
77         dataSource={this.state.dataSource}
78
79         renderSeparator= {this.ListViewItemSeparator}
80
81         renderRow={ (rowData) => <Text style={styles.rowViewContainer}
82
83           onPress={this.GetStudentIDFunction.bind(
84             this, rowData.student_id,
85             rowData.student_name,
86             rowData.student_class,
87             rowData.student_phone_number,
88             rowData.student_email
89           )} >
90
91           {rowData.student_name}
92
93           </Text> }
94
95       </View>
96     />
97   );
98 }
99
100 }
101
102 }

```

Screenshot of ShowStudentListActivity :



10. Create a class named as **EditStudentRecordActivity**. This class is used to Edit the current open record to MySQL database. This class is also used to delete the current selected opened record.

constructor : We would create 5 state named as TextInput_Student_ID, TextInput_Student_Name, TextInput_Student_Class, TextInput_Student_PhoneNumber, TextInput_Student_Email in constructor(). Each state is used to store and hold the already sent data from previous activity.

componentDidMount() : Storing all the sent value from previous activity into State using this.props.navigation.state.params .

navigationOptions : Setting up the activity title shows inside the Title Action bar.

UpdateStudentRecord() : This function would get the all TextInput entered value and send them to server using fetch() API where they will update the existing record using ID.

DeleteStudentRecord() : This function would get the only Student ID from TextInput_Student_ID and send the only TextInput_Student_ID to server . On server we would delete the current record according to ID.

```

1  class EditStudentRecordActivity extends Component {
2
3      constructor(props) {
4
5          super(props)
6
7          this.state = {
8
9              TextInput_Student_ID: '',
10             TextInput_Student_Name: '',
11             TextInput_Student_Class: '',
12             TextInput_Student_PhoneNumber: '',
13             TextInput_Student_Email: '',
14
15         }
16     }
17 }

```

```

18
19 componentDidMount(){
20
21   // Received Student Details Sent From Previous Activity and Set Into State.
22   this.setState({
23     TextInput_Student_ID : this.props.navigation.state.params.ID,
24     TextInput_Student_Name : this.props.navigation.state.params.NAME,
25     TextInput_Student_Class : this.props.navigation.state.params.CLASS,
26     TextInput_Student_PhoneNumber : this.props.navigation.state.params.PHONE_NUMBER,
27     TextInput_Student_Email : this.props.navigation.state.params.EMAIL,
28   })
29
30   }
31
32   static navigationOptions =
33   {
34     title: 'EditStudentRecordActivity',
35   };
36
37   UpdateStudentRecord = () =>{
38
39     fetch('https://reactnativecode.000webhostapp.com/Student/UpdateStudentRecord.php', {
40       method: 'POST',
41       headers: {
42         'Accept': 'application/json',
43         'Content-Type': 'application/json',
44       },
45       body: JSON.stringify({
46
47         student_id : this.state.TextInput_Student_ID,
48
49         student_name : this.state.TextInput_Student_Name,
50
51         student_class : this.state.TextInput_Student_Class,
52
53         student_phone_number : this.state.TextInput_Student_PhoneNumber,
54
55         student_email: this.state.TextInput_Student_Email
56
57       })
58     }).then((response) => response.json())
59       .then((responseJson) => {
60
61         // Showing response message coming from server updating records.
62         Alert.alert(responseJson);
63
64         }).catch((error) => {
65           console.error(error);
66         });
67
68   }
69
70
71   DeleteStudentRecord = () =>{
72
73     fetch('https://reactnativecode.000webhostapp.com/Student/DeleteStudentRecord.php', {
74       method: 'POST',
75       headers: {
76         'Accept': 'application/json',
77         'Content-Type': 'application/json',
78       },
79       body: JSON.stringify({
80
81         student_id : this.state.TextInput_Student_ID
82
83       })
84     }).then((response) => response.json())
85       .then((responseJson) => {
86
87         // Showing response message coming from server after inserting records.
88         Alert.alert(responseJson);
89
90         }).catch((error) => {
91           console.error(error);
92         });
93
94         this.props.navigation.navigate('First');
95
96   }
97
98   }
99
100  render() {
101
102    return (
103
104    <View style={styles.MainContainer}>
105
106      <Text style={{fontSize: 20, textAlign: 'center', marginBottom: 7}}> Edit Student Record Form </Text>
107
108      <TextInput
109
110        placeholder="Student Name Shows Here"
111
112        value={this.state.TextInput_Student_Name}
113
114        onChangeText={ TextInputValue => this.setState({ TextInput_Student_Name : TextInputValue }) }
115
116        underlineColorAndroid='transparent'

```

```

117         style={styles.TextInputStyleClass}
118     />
119
120     <TextInput
121
122         placeholder="Student Class Shows Here"
123         value={this.state.TextInput_Student_Class}
124         onChangeText={ TextInputValue => this.setState({ TextInput_Student_Class : TextInputValue }) }
125         underlineColorAndroid='transparent'
126         style={styles.TextInputStyleClass}
127     />
128
129     <TextInput
130
131         placeholder="Student Phone Number Shows Here"
132         value={this.state.TextInput_Student_PhoneNumber}
133         onChangeText={ TextInputValue => this.setState({ TextInput_Student_PhoneNumber : TextInputValue }) }
134         underlineColorAndroid='transparent'
135         style={styles.TextInputStyleClass}
136     />
137
138     <TextInput
139
140         placeholder="Student Email Shows Here"
141         value={this.state.TextInput_Student_Email}
142         onChangeText={ TextInputValue => this.setState({ TextInput_Student_Email : TextInputValue }) }
143         underlineColorAndroid='transparent'
144         style={styles.TextInputStyleClass}
145     />
146
147     <TouchableOpacity activeOpacity = { .4 } style={styles.TouchableOpacityStyle} onPress={this.UpdateStudentRecord} >
148         <Text style={styles.TextStyle}> UPDATE STUDENT RECORD </Text>
149     </TouchableOpacity>
150
151     <TouchableOpacity activeOpacity = { .4 } style={styles.TouchableOpacityStyle} onPress={this.DeleteStudentRecord} >
152         <Text style={styles.TextStyle}> DELETE CURRENT RECORD </Text>
153     </TouchableOpacity>
154
155 </View>
156
157 );
158 }
159 }

```

11. Create **StackNavigator** and define all your activity classes inside it.

```

1 export default MyNewProject = StackNavigator(
2
3   {
4
5     First: { screen: MainActivity },
6     Second: { screen: ShowStudentListActivity },
7     Third: { screen: EditStudentRecordActivity }
8   }
9 );

```

12. Create CSS Style classes for All activities.

```

1 const styles = StyleSheet.create({
2
3   MainContainer :{
4
5     alignItems: 'center',
6     flex:1,
7     paddingTop: 30,
8     backgroundColor: '#fff'
9   },
10
11   MainContainer_For_Show_StudentList_Activity :{
12
13     flex:1,
14     paddingTop: (Platform.OS == 'ios') ? 20 : 0,
15     marginLeft: 5,
16     marginRight: 5
17   },
18
19

```

```

20
21   TextInputStyleClass: {
22
23     textAlign: 'center',
24     width: '90%',
25     marginBottom: 7,
26     height: 40,
27     borderWidth: 1,
28     borderColor: '#FF5722',
29     borderRadius: 5 ,
30
31   },
32
33   TouchableOpacityStyle: {
34
35     paddingTop:10,
36     paddingBottom:10,
37     borderRadius:5,
38     marginBottom:7,
39     width: '90%',
40     backgroundColor: '#00BCD4'
41
42   },
43
44   TextStyle:{
45     color:'#fff',
46     textAlign:'center',
47   },
48
49   containerView: {
50     fontSize: 20,
51     paddingRight: 10,
52     paddingTop: 10,
53     paddingBottom: 10,
54   }
55
56 });

```

13. Complete source code for **App.js** File :

```

1  import React, { Component } from 'react';
2
3  import { StyleSheet, View, Alert, TextInput, Button, Text, Platform, TouchableOpacity, ListView, ActivityIndicator } from 'react-native'
4
5  import { StackNavigator } from 'react-navigation';
6
7  class MainActivity extends Component {
8
9     static navigationOptions =
10     {
11       title: 'MainActivity',
12     };
13
14   constructor(props) {
15
16     super(props)
17
18     this.state = {
19
20       TextInput_Student_Name: '',
21       TextInput_Student_Class: '',
22       TextInput_Student_PhoneNumber: '',
23       TextInput_Student_Email: '',
24
25     }
26
27   }
28
29   InsertStudentRecordsToServer = () =>{
30
31     fetch('https://reactnativecode.000webhostapp.com/Student/InsertStudentData.php', {
32       method: 'POST',
33       headers: {
34         'Accept': 'application/json',
35         'Content-Type': 'application/json',
36       },
37       body: JSON.stringify({
38
39         student_name : this.state.TextInput_Student_Name,
40
41         student_class : this.state.TextInput_Student_Class,
42
43         student_phone_number : this.state.TextInput_Student_PhoneNumber,
44
45         student_email: this.state.TextInput_Student_Email
46
47       })
48     }).then((response) => response.json())
49       .then((responseJson) => {
50
51         // Showing response message coming from server after inserting records.
52         Alert.alert(responseJson);
53
54       }).catch((error) => {
55         console.error(error);
56       });
57
58   }

```

```

59 }
60
61 GoTo_Show_StudentList_Activity_Function = () =>
62 {
63   this.props.navigation.navigate('Second');
64 }
65
66
67 render() {
68   return (
69     <View style={styles.MainContainer}>
70
71
72       <Text style={{fontSize: 20, textAlign: 'center', marginBottom: 7}}> Student Registration Form </Text>
73
74       <TextInput
75
76         placeholder="Enter Student Name"
77
78         onChangeText={ TextInputValue => this.setState({ TextInput_Student_Name : TextInputValue }) }
79
80         underlineColorAndroid='transparent'
81
82         style={styles.TextInputStyleClass}
83       />
84
85       <TextInput
86
87         placeholder="Enter Student Class"
88
89         onChangeText={ TextInputValue => this.setState({ TextInput_Student_Class : TextInputValue }) }
90
91         underlineColorAndroid='transparent'
92
93         style={styles.TextInputStyleClass}
94       />
95
96       <TextInput
97
98         placeholder="Enter Student Phone Number"
99
100        onChangeText={ TextInputValue => this.setState({ TextInput_Student_PhoneNumber : TextInputValue }) }
101
102        underlineColorAndroid='transparent'
103
104        style={styles.TextInputStyleClass}
105      />
106
107      <TextInput
108
109        placeholder="Enter Student Email"
110
111        onChangeText={ TextInputValue => this.setState({ TextInput_Student_Email : TextInputValue }) }
112
113        underlineColorAndroid='transparent'
114
115        style={styles.TextInputStyleClass}
116      />
117
118      <TouchableOpacity activeOpacity = { .4 } style={styles.TouchableOpacityStyle} onPress={this.InsertStudentRecordsToServer} >
119
120        <Text style={styles.TextStyle}> INSERT STUDENT RECORD TO SERVER </Text>
121
122      </TouchableOpacity>
123
124      <TouchableOpacity activeOpacity = { .4 } style={styles.TouchableOpacityStyle} onPress={this.GoTo_Show_StudentList_Activity_Functi
125
126        <Text style={styles.TextStyle}> SHOW ALL INSERTED STUDENT RECORDS IN LISTVIEW </Text>
127
128      </TouchableOpacity>
129
130    </View>
131  );
132 }
133
134 }
135 }
136 }
137
138 class ShowStudentListActivity extends Component {
139   constructor(props) {
140     super(props);
141
142     this.state = {
143
144       isLoading: true
145     }
146   }
147
148   static navigationOptions =
149   {
150     title: 'ShowStudentListActivity',
151   };
152
153   componentDidMount() {
154
155
156
157

```

```

158     return fetch('https://reactnativecode.000webhostapp.com/Student/ShowAllStudentsList.php')
159     .then((response) => response.json())
160     .then((responseJson) => {
161         let ds = new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2});
162         this.setState({
163             isLoading: false,
164             dataSource: ds.cloneWithRows(responseJson),
165         }, function() {
166             // In this block you can do something with new state.
167         });
168     })
169     .catch((error) => {
170         console.error(error);
171     });
172 }
173
174 GetStudentIDFunction=(student_id,student_name, student_class, student_phone_number, student_email)=>{
175
176     this.props.navigation.navigate('Third', {
177
178         ID : student_id,
179         NAME : student_name,
180         CLASS : student_class,
181         PHONE_NUMBER : student_phone_number,
182         EMAIL : student_email
183
184     });
185
186 }
187
188 ListViewItemSeparator = () => {
189     return (
190         <View
191             style={{
192                 height: .5,
193                 width: "100%",
194                 backgroundColor: "#000",
195             }}
196         />
197     );
198 }
199
200 render() {
201     if (this.state.isLoading) {
202         return (
203             <View style={{flex: 1, paddingTop: 20}}>
204                 <ActivityIndicator />
205             </View>
206         );
207     }
208
209     return (
210         <View style={styles.MainContainer_For_Show_StudentList_Activity}>
211
212             <ListView
213
214                 dataSource={this.state.dataSource}
215
216                 renderSeparator= {this.ListViewItemSeparator}
217
218                 renderRow={ (rowData) => <Text style={styles.rowViewContainer}
219
220                     onPress={this.GetStudentIDFunction.bind(
221                         this, rowData.student_id,
222                         rowData.student_name,
223                         rowData.student_class,
224                         rowData.student_phone_number,
225                         rowData.student_email
226                     )} >
227
228                     {rowData.student_name}
229
230                     </Text> }
231
232             />
233
234         </View>
235     );
236 }
237
238 }
239
240
241 class EditStudentRecordActivity extends Component {
242
243     constructor(props) {
244
245         super(props)
246
247         this.state = {
248
249             TextInput_Student_ID: '',
250             TextInput_Student_Name: '',
251             TextInput_Student_Class: '',
252             TextInput_Student_PhoneNumber: '',
253             TextInput_Student_Email: '',
254
255         }
256

```

```

257     }
258
259     componentDidMount(){
260
261         // Received Student Details Sent From Previous Activity and Set Into State.
262         this.setState({
263             TextInput_Student_ID : this.props.navigation.state.params.ID,
264             TextInput_Student_Name: this.props.navigation.state.params.NAME,
265             TextInput_Student_Class: this.props.navigation.state.params.CLASS,
266             TextInput_Student_PhoneNumber: this.props.navigation.state.params.PHONE_NUMBER,
267             TextInput_Student_Email: this.props.navigation.state.params.EMAIL,
268         })
269     }
270 }
271
272 static navigationOptions =
273 {
274     title: 'EditStudentRecordActivity',
275 };
276
277 UpdateStudentRecord = () =>{
278
279     fetch('https://reactnativecode.000webhostapp.com/Student/UpdateStudentRecord.php', {
280         method: 'POST',
281         headers: {
282             'Accept': 'application/json',
283             'Content-Type': 'application/json',
284         },
285         body: JSON.stringify({
286
287             student_id : this.state.TextInput_Student_ID,
288
289             student_name : this.state.TextInput_Student_Name,
290
291             student_class : this.state.TextInput_Student_Class,
292
293             student_phone_number : this.state.TextInput_Student_PhoneNumber,
294
295             student_email: this.state.TextInput_Student_Email
296
297         })
298     }).then((response) => response.json())
299     .then((responseJson) => {
300
301         // Showing response message coming from server updating records.
302         Alert.alert(responseJson);
303
304     }).catch((error) => {
305         console.error(error);
306     });
307 }
308
309
310
311
312 DeleteStudentRecord = () =>{
313
314     fetch('https://reactnativecode.000webhostapp.com/Student/DeleteStudentRecord.php', {
315         method: 'POST',
316         headers: {
317             'Accept': 'application/json',
318             'Content-Type': 'application/json',
319         },
320         body: JSON.stringify({
321
322             student_id : this.state.TextInput_Student_ID
323
324         })
325     }).then((response) => response.json())
326     .then((responseJson) => {
327
328         // Showing response message coming from server after inserting records.
329         Alert.alert(responseJson);
330
331     }).catch((error) => {
332         console.error(error);
333     });
334
335     this.props.navigation.navigate('First');
336 }
337
338
339
340 render() {
341
342     return (
343
344     <View style={styles.MainContainer}>
345
346         <Text style={{fontSize: 20, textAlign: 'center', marginBottom: 7}}> Edit Student Record Form </Text>
347
348         <TextInput
349
350             placeholder="Student Name Shows Here"
351
352             value={this.state.TextInput_Student_Name}
353
354             onChangeText={ TextInputValue => this.setState({ TextInput_Student_Name : TextInputValue }) }
355

```

```

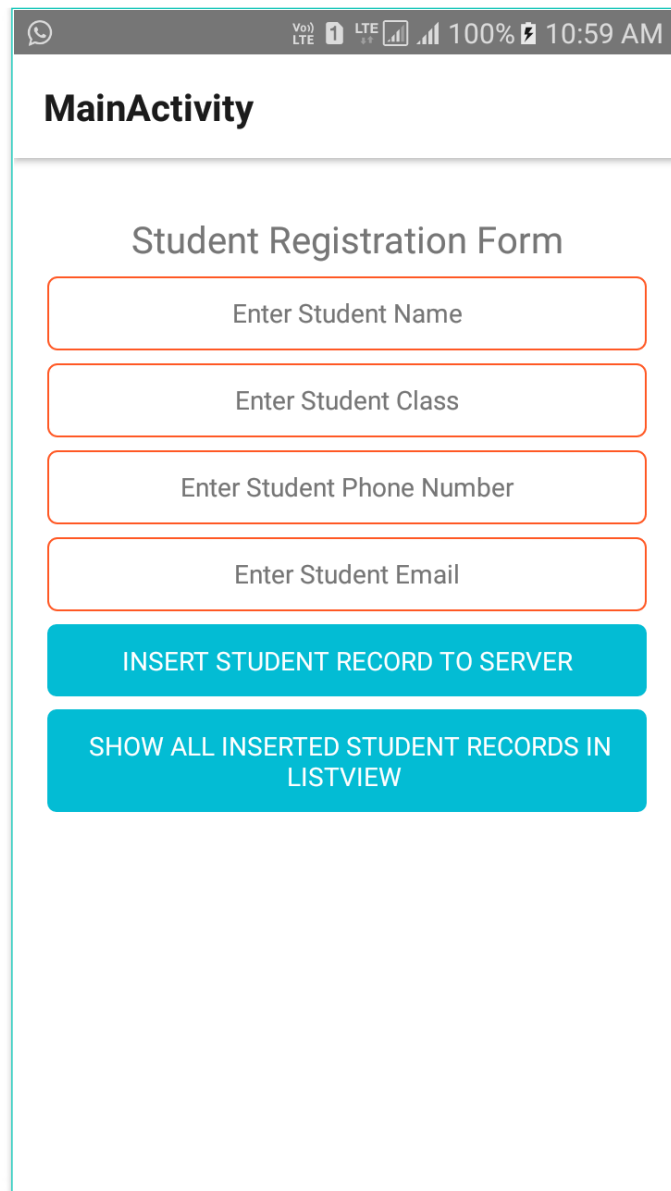
356         underlineColorAndroid='transparent'
357
358         style={styles.TextInputStyleClass}
359     />
360
361     <TextInput
362
363         placeholder="Student Class Shows Here"
364
365         value={this.state.TextInput_Student_Class}
366
367         onChangeText={ TextInputValue => this.setState({ TextInput_Student_Class : TextInputValue }) }
368
369         underlineColorAndroid='transparent'
370
371         style={styles.TextInputStyleClass}
372     />
373
374     <TextInput
375
376         placeholder="Student Phone Number Shows Here"
377
378         value={this.state.TextInput_Student_PhoneNumber}
379
380         onChangeText={ TextInputValue => this.setState({ TextInput_Student_PhoneNumber : TextInputValue }) }
381
382         underlineColorAndroid='transparent'
383
384         style={styles.TextInputStyleClass}
385     />
386
387     <TextInput
388
389         placeholder="Student Email Shows Here"
390
391         value={this.state.TextInput_Student_Email}
392
393         onChangeText={ TextInputValue => this.setState({ TextInput_Student_Email : TextInputValue }) }
394
395         underlineColorAndroid='transparent'
396
397         style={styles.TextInputStyleClass}
398     />
399
400     <TouchableOpacity activeOpacity = { .4 } style={styles.TouchableOpacityStyle} onPress={this.UpdateStudentRecord} >
401
402         <Text style={styles.TextStyle}> UPDATE STUDENT RECORD </Text>
403
404     </TouchableOpacity>
405
406     <TouchableOpacity activeOpacity = { .4 } style={styles.TouchableOpacityStyle} onPress={this.DeleteStudentRecord} >
407
408         <Text style={styles.TextStyle}> DELETE CURRENT RECORD </Text>
409
410     </TouchableOpacity>
411
412 </View>
413
414     );
415 }
416
417 }
418
419
420 export default MyNewProject = StackNavigator(
421
422     {
423
424         First: { screen: MainActivity },
425
426         Second: { screen: ShowStudentListActivity },
427
428         Third: { screen: EditStudentRecordActivity }
429
430     });
431
432 const styles = StyleSheet.create({
433
434     MainContainer :{
435
436         alignItems: 'center',
437         flex:1,
438         paddingTop: 30,
439         backgroundColor: '#fff'
440
441     },
442
443     MainContainer_For_Show_StudentList_Activity :{
444
445         flex:1,
446         paddingTop: (Platform.OS === 'ios') ? 20 : 0,
447         marginLeft: 5,
448         marginRight: 5
449
450     },
451
452     TextInputStyleClass: {
453
454         textAlign: 'center',

```



```
455 width: '90%',
456 marginBottom: 7,
457 height: 40,
458 borderWidth: 1,
459 borderColor: '#FF5722',
460 borderRadius: 5 ,
461 },
462 },
463
464 TouchableOpacityStyle: {
465   paddingTop:10,
466   paddingBottom:10,
467   borderRadius:5,
468   marginBottom:7,
469   width: '90%',
470   backgroundColor: '#00BCD4'
471 },
472 },
473 },
474
475 TextStyle:{
476   color:'#fff',
477   textAlign:'center',
478 },
479
480 rowViewContainer: {
481   fontSize: 20,
482   paddingRight: 10,
483   paddingTop: 10,
484   paddingBottom: 10,
485 }
486
487 };
```

Screenshot in Android device :



The screenshot shows a mobile application interface for a "Student Registration Form". At the top, there is a status bar with various icons and the time "10:59 AM". Below the status bar, the title "MainActivity" is displayed. The form itself consists of four text input fields, each with a red border and a light gray background. The labels for these fields are "Enter Student Name", "Enter Student Class", "Enter Student Phone Number", and "Enter Student Email". Below the input fields, there are two blue buttons with white text. The first button is labeled "INSERT STUDENT RECORD TO SERVER" and the second button is labeled "SHOW ALL INSERTED STUDENT RECORDS IN LISTVIEW".

VoLTE LTE 100% 10:59 AM

MainActivity

Student Registration Form

Enter Student Name

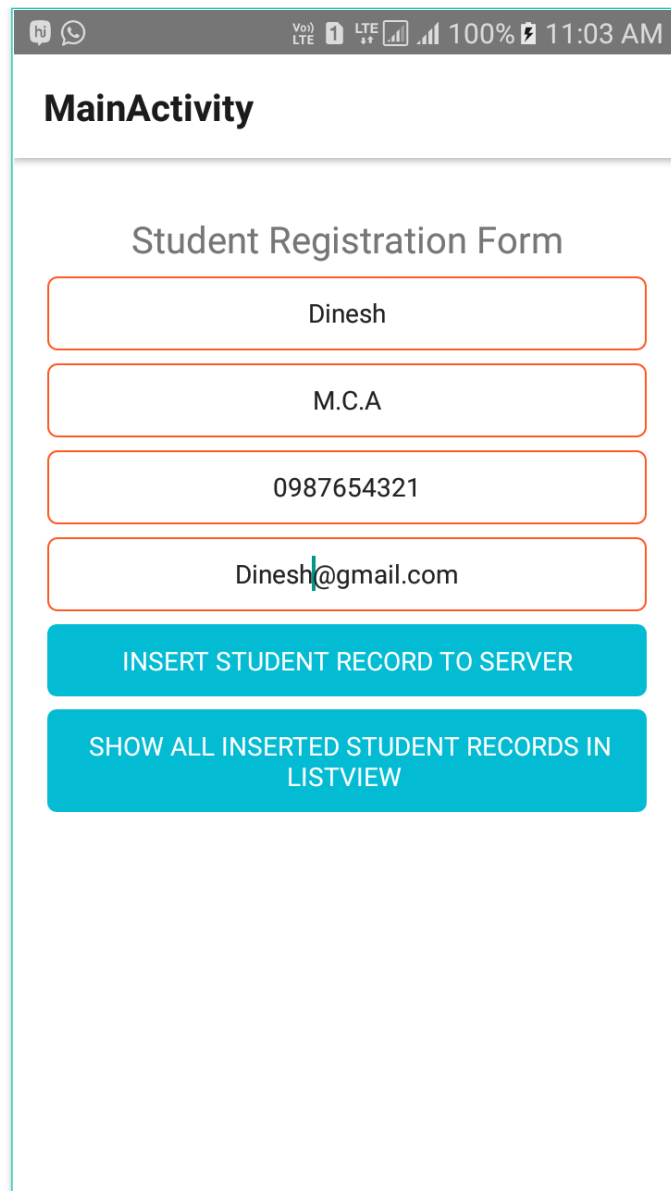
Enter Student Class

Enter Student Phone Number

Enter Student Email

INSERT STUDENT RECORD TO SERVER

SHOW ALL INSERTED STUDENT RECORDS IN LISTVIEW



The screenshot shows a mobile application interface titled "MainActivity". Below the title is a "Student Registration Form". The form consists of four text input fields, each with a red border, containing the following text: "Dinesh", "M.C.A", "0987654321", and "Dinesh@gmail.com". Below the input fields are two blue buttons with white text. The first button says "INSERT STUDENT RECORD TO SERVER" and the second button says "SHOW ALL INSERTED STUDENT RECORDS IN LISTVIEW". The top of the screen shows a status bar with various icons and the time "11:03 AM".

MainActivity

Student Registration Form

Dinesh

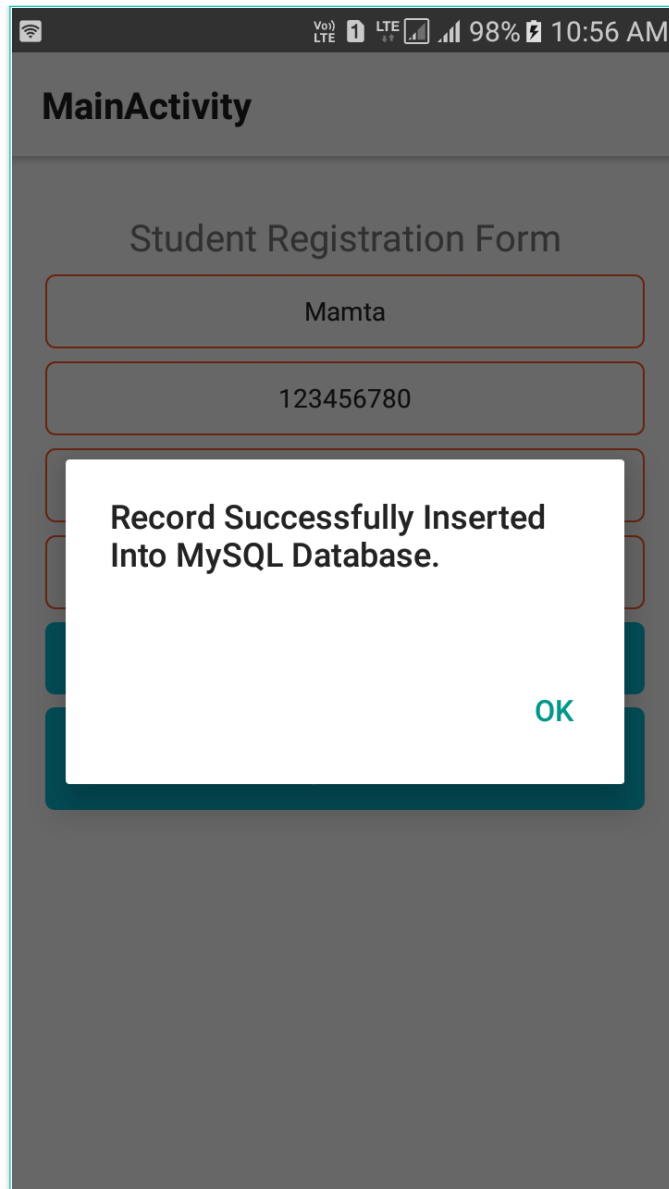
M.C.A

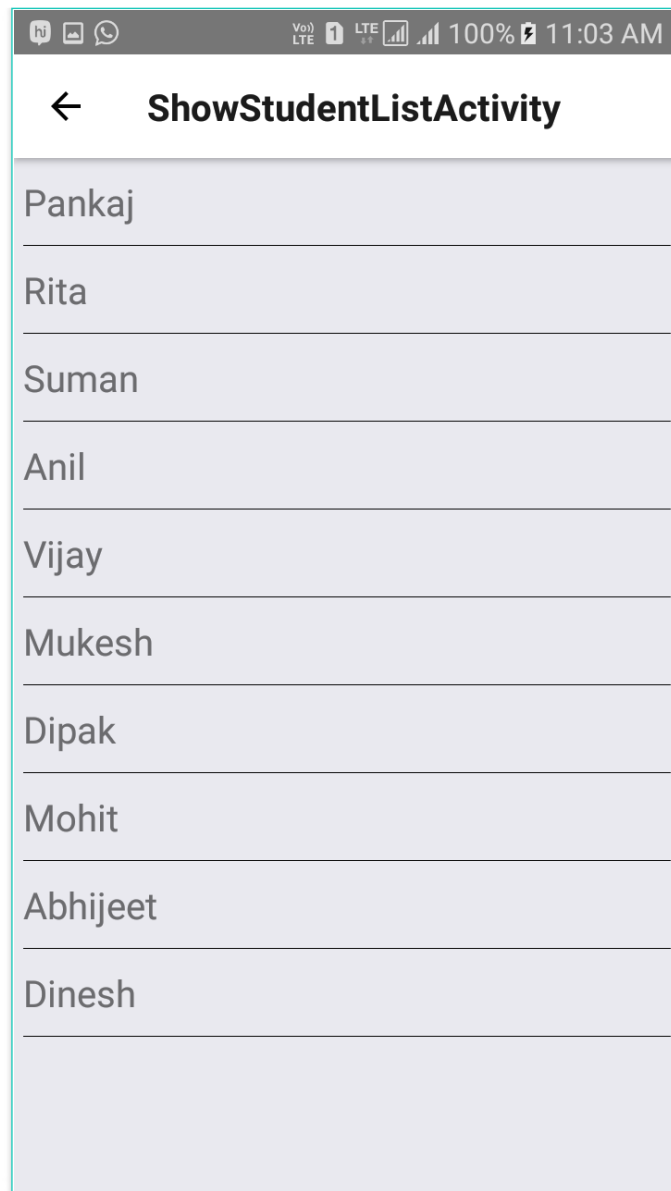
0987654321

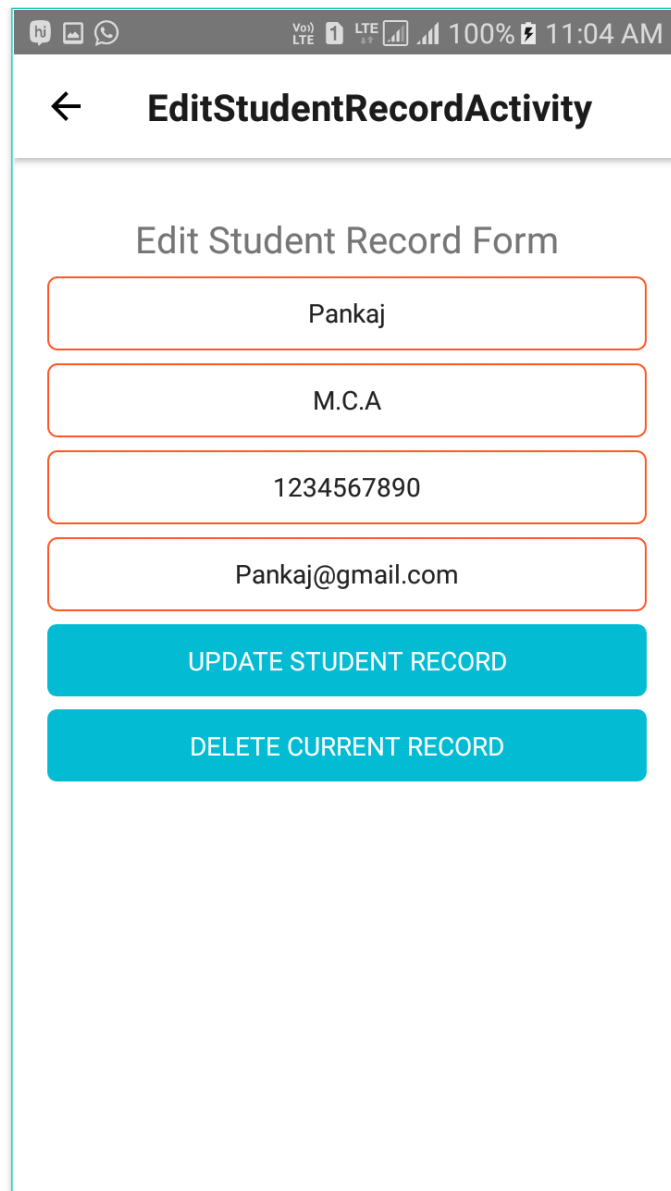
Dinesh@gmail.com

INSERT STUDENT RECORD TO SERVER

SHOW ALL INSERTED STUDENT RECORDS IN LISTVIEW







The screenshot shows a mobile application interface for editing a student record. At the top, there is a status bar with icons for notifications, LTE signal, and battery level at 100%. Below the status bar is a header bar with a back arrow icon and the title "EditStudentRecordActivity". The main content area is titled "Edit Student Record Form" and contains four text input fields with red borders. The first field contains "Pankaj", the second "M.C.A", the third "1234567890", and the fourth "Pankaj@gmail.com". Below the input fields are two blue buttons with white text: "UPDATE STUDENT RECORD" and "DELETE CURRENT RECORD".

VoLTE LTE 100% 11:04 AM

← **EditStudentRecordActivity**

Edit Student Record Form

Pankaj

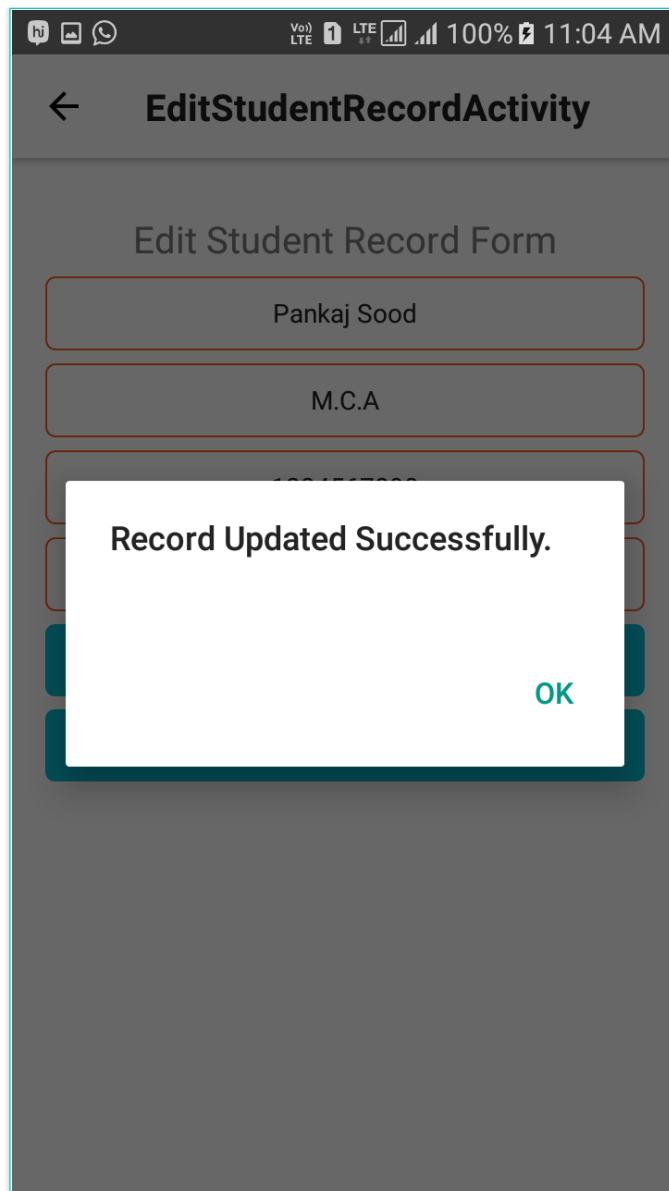
M.C.A

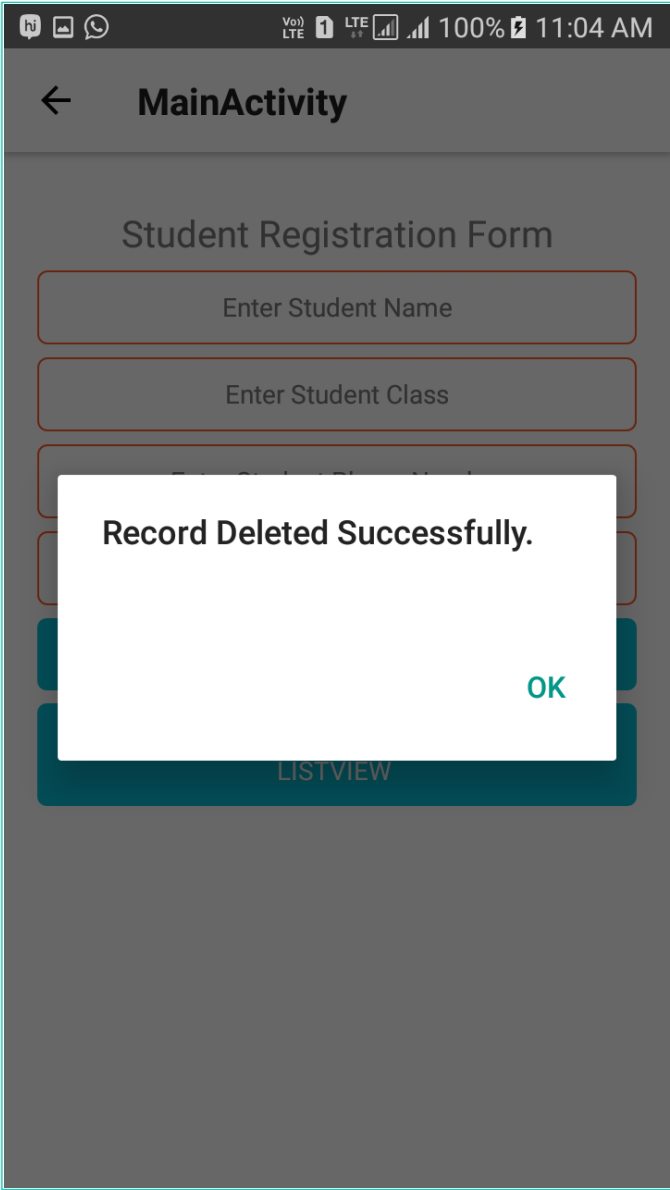
1234567890

Pankaj@gmail.com

UPDATE STUDENT RECORD

DELETE CURRENT RECORD





Screenshot in iOS device :

Carrier 11:37 AM

MainActivity

Student Registration Form

Enter Student Name

Enter Student Class

Enter Student Phone Number

Enter Student Email

INSERT STUDENT RECORD TO SERVER

SHOW ALL INSERTED STUDENT RECORDS IN LISTVIEW

Carrier 11:38 AM

MainActivity

Student Registration Form

Vikas

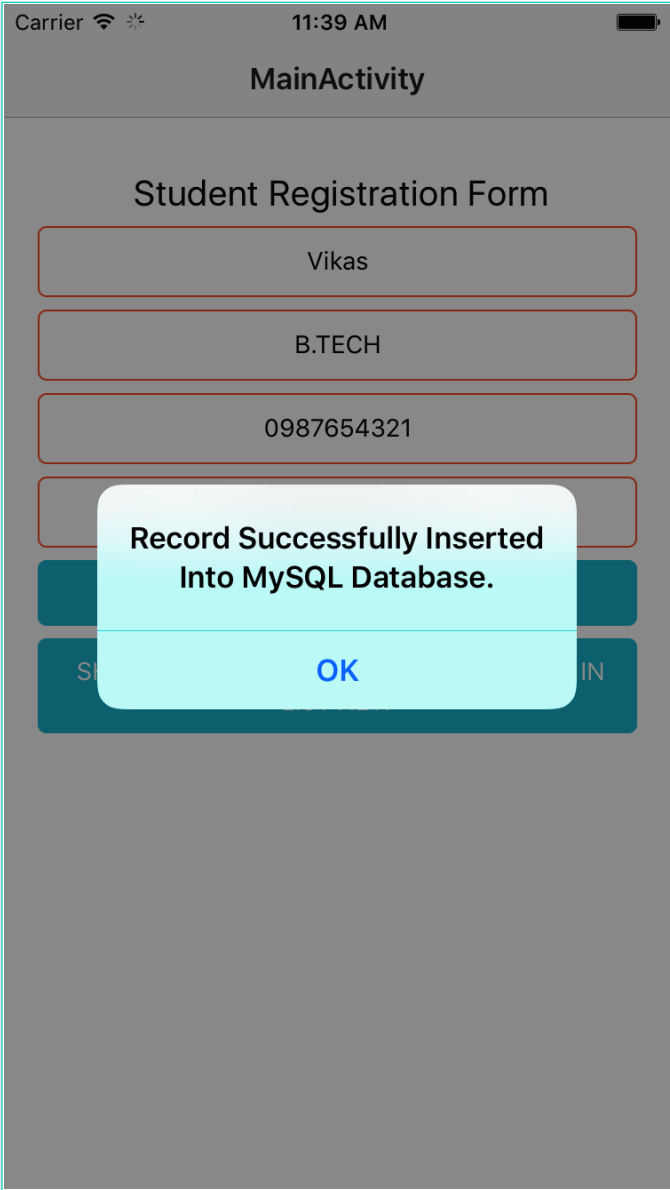
B.TECH

0987654321



Vikas@gmail.com


INSERT STUDENT RECORD TO SERVER

SHOW ALL INSERTED STUDENT RECORDS IN LISTVIEW





Carrier  

11:39 AM 

[< Back](#) EditStudentRecordActi...

Edit Student Record Form

Dipak

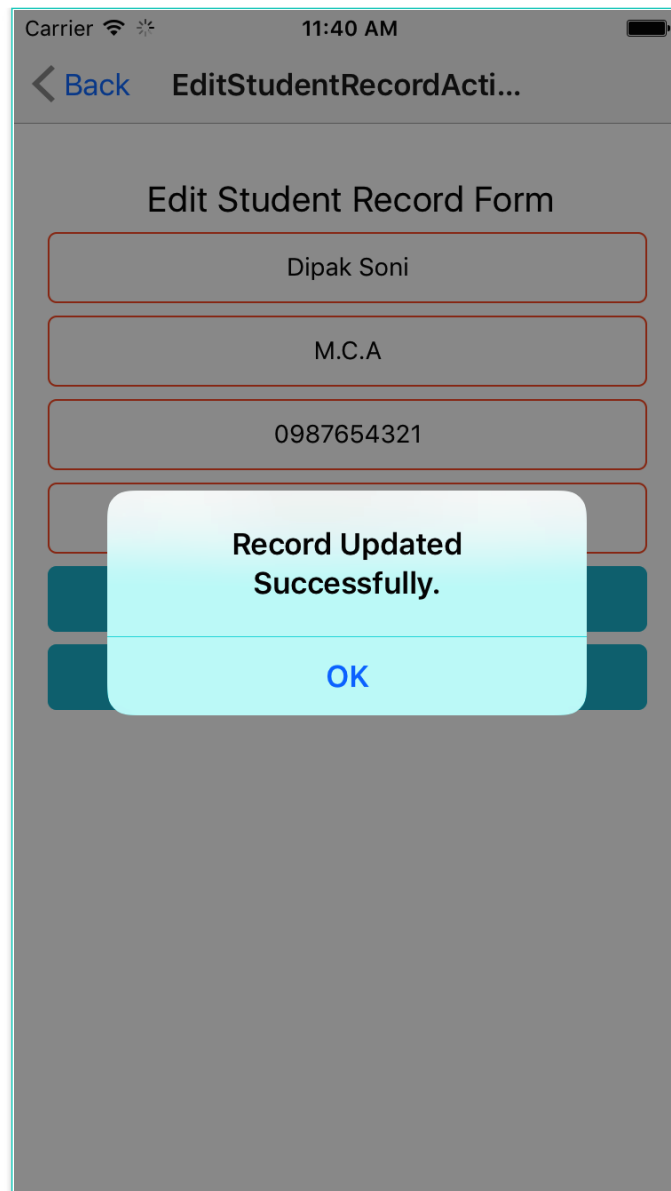
M.C.A

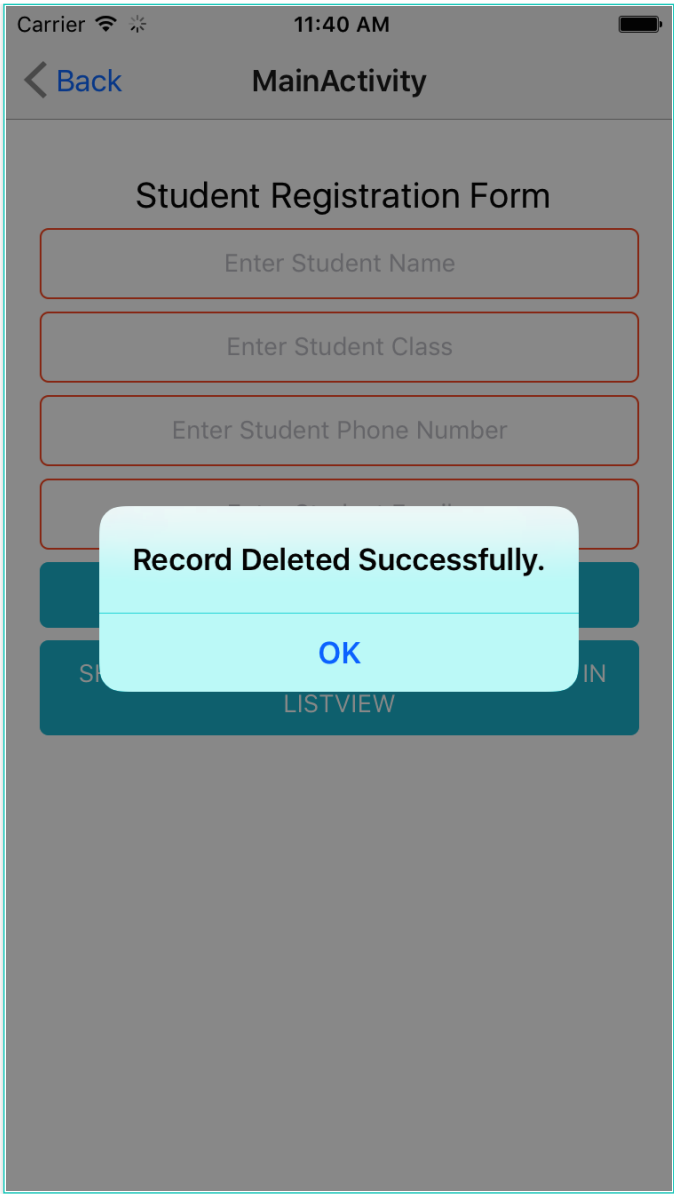
0987654321

Dipak@gmail.com

UPDATE STUDENT RECORD

DELETE CURRENT RECORD

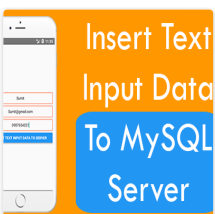




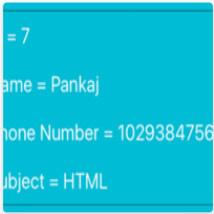
Also Read :



React Native Apply Filter on JSON ListView Data Using PHP MySQL



React Native Insert Text Input Data To MySQL Server Android iOS Tutorial



React Native Create Custom JSON ListView with Multiple Items Tutorial



React Native Fetch API Tutorial to Insert Into MySQL Database Using PHP iOS Android Example



React Native Create Picker Spinner Using Dynamic JSON MySQL Data

[Bookmark.](#)[◀ React Native Start Stop Device Vibration on Button Click
Android iOS](#)[Create Gradient Shade Effect Button in React Native Android
iOS Example ▶](#)

4 Comments

**BlackBox**

December 14, 2017 at 1:36 pm

Aweome tutorial, i was waiting for it for a very long time 😊

[↩ Reply](#)**admin**

December 14, 2017 at 3:47 pm

Welcome Bro 😊 .

[↩ Reply](#)**Winson**

February 9, 2018 at 1:16 am

Is there any tutorial that can do the same thing like this for Firebase?

[↩ Reply](#)**admin**

February 9, 2018 at 3:27 am

No Winson i have not created tutorial on Firebase but soon i will publish new tutorial on Firebase 😊 .

[↩ Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name*

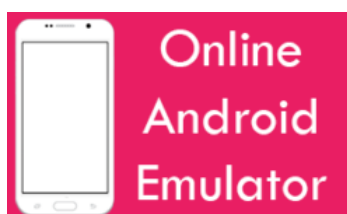
Email*

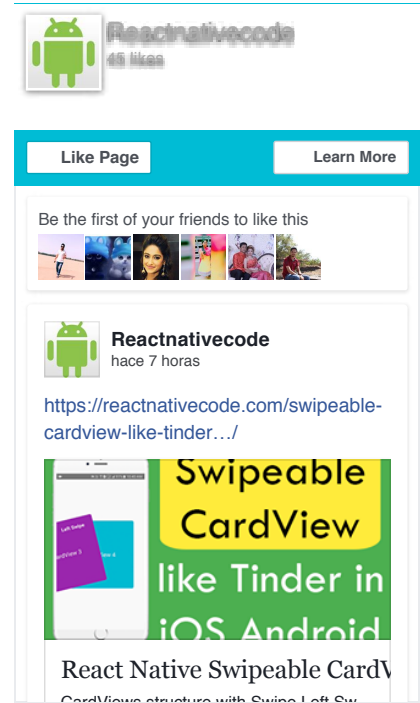
Website

[Post Comment](#)

React Native 0.54-RC

Search Here





© ReactNativeCode.com . All Rights reserved.

Powered by Nirvana & WordPress.