



Golden-Gate Int'l College

Affiliated to Tribhuvan University

Old Baneswor, Kathmandu

A

Lab Report on

Mobile Programming

(CACS 351)



Lab Title: CRUD Application

Lab Sheet No: 3

Submitted by:

Name: Surakshya Lama

Roll no:15

Registration no:6-2-453-15-2021

Submitted to:

Mr. Deepak Thakur

Submission date: 11th May, 2025

1. Write android program to create a small CRUD Application(P/H)

- a. Application must have menu (insert data, display data).
- b. Data should be store in MYSQL server.
- c. Display data in list view.
- d. When list item is clicked detail of item should be display in dialog box.
- e. When list item is long pressed it should display context menu for delete and update.
- f. When update is selected than the detail of list item should be display in another fragment for update.

PHP files:

db.php:

```
1  <?php
2  $host = "localhost";
3  $user = "root";
4  $password = "";
5  $db = "studentdb";
6
7  $conn = mysqli_connect($host, $user, $password, $db, 3307);
8  if (!$conn) {
9      die("Connection failed: " . mysqli_connect_error());
10 }
11 ?>
```

insert.php:

```
1  <?php
2  include "db.php";
3  $name = $_POST['name'];
4  $roll = $_POST['roll'];
5
6  $sql = "INSERT INTO students (name, roll) VALUES ('$name', '$roll')";
7  if (mysqli_query($conn, $sql)) {
8      echo "Success";
9  } else {
10     echo "Error";
11 }
12 ?>
```

Lab report 3

update.php:

```
1  <?php
2  header("Content-Type: application/json");
3  include "db.php";
4
5  // Get data from POST request
6  $data = json_decode(file_get_contents("php://input"), true);
7  $id = $data['id'] ?? null;
8  $name = $data['name'] ?? null;
9  $roll = $data['roll'] ?? null;
10
11 // Validate input
12 if (!$id || !$name || !$roll) {
13     echo json_encode(["status" => "error", "message" => "Missing parameters"]);
14     exit;
15 }
16
17 // Prevent SQL injection
18 $id = mysqli_real_escape_string($conn, $id);
19 $name = mysqli_real_escape_string($conn, $name);
20 $roll = mysqli_real_escape_string($conn, $roll);
21
22 // Update query
23 $sql = "UPDATE students SET name='$name', roll='$roll' WHERE id='$id'";
24
25 if (mysqli_query($conn, $sql)) {
26     echo json_encode(["status" => "success", "message" => "Student updated"]);
27 } else {
28     echo json_encode(["status" => "error", "message" => "Update failed: " . mysqli_error($conn)]);
29 }
30
31 mysqli_close($conn);
32 ?>
```

fetch.php:

```
1  <?php
2  include "db.php";
3  $result = mysqli_query($conn, "SELECT * FROM students");
4  $data = array();
5  while ($row = mysqli_fetch_assoc($result)) {
6      $data[] = $row;
7  }
8  echo json_encode($data);
9  ?>
```

Lab report 3

delete.php:

```
1  <?php
2  include "db.php";
3
4  $id = $_POST['id'] ?? null;
5
6  if ($id) {
7      $stmt = $conn->prepare("DELETE FROM students WHERE id = ?");
8      $stmt->bind_param("i", $id);
9
10     if ($stmt->execute()) {
11         echo "Deleted";
12     } else {
13         echo "Error";
14     }
15
16     $stmt->close();
17 } else {
18     echo "Invalid ID";
19 }
20 $conn->close();
21 ?>
```

Menu files:

main.xml:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android">
3  <item android:title="Insert"
4      android:id="@+id/insert" />
5      <item android:title="Display"
6          android:id="@+id/display" />
7  </menu>
```

main2.xml:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android">
3      <item
4          android:id="@+id/menu_update"
5          android:title="Update" />
6      <item
7          android:id="@+id/menu_delete"
8          android:title="Delete" />
9  </menu>
```

Lab report 3

Layout files:

layout.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6     <TextView
7         android:layout_width="match_parent"
8         android:layout_height="wrap_content"
9         android:textAlignment="center"
10        android:text="Student Details"
11        android:textSize="25sp"
12        android:layout_margin="10sp"/>
13     <FrameLayout
14        android:id="@+id/frag"
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content" />
17
18 </LinearLayout>
```

fragment_insert.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:orientation="vertical"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content">
6     <EditText
7         android:id="@+id/etName"
8         android:hint="Enter Name"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:layout_margin="15sp"/>
12     <EditText
13        android:id="@+id/etRoll"
14        android:hint="Enter Roll No: "
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content"
17        android:layout_margin="15sp"/>
18     <RelativeLayout
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:gravity="center">
22         <Button
23             android:layout_width="wrap_content"
24             android:layout_height="wrap_content"
25             android:layout_margin="15sp"
26             android:text="back"
27             android:id="@+id/back"
28             android:layout_toRightOf="@+id/save"/>
29         <Button
30             android:layout_width="wrap_content"
31             android:layout_height="wrap_content"
32             android:layout_margin="15sp"
33             android:text="insert"
34             android:id="@+id/save"/>
35     </RelativeLayout>
36 </LinearLayout>
```

Lab report 3

fragment_display.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <ListView
7         android:id="@+id/list"
8         android:layout_width="match_parent"
9         android:layout_height="wrap_content"
10        android:layout_margin="15dp" />
11 </FrameLayout>
```

fragment_update.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:orientation="vertical"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content">
6     <EditText
7         android:id="@+id/etName"
8         android:hint="Enter Name"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:layout_margin="15sp"/>
12     <EditText
13         android:id="@+id/etRoll"
14         android:hint="Enter Roll No: "
15         android:layout_width="match_parent"
16         android:layout_height="wrap_content"
17         android:layout_margin="15sp"/>
18     <RelativeLayout
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:gravity="center">
22         <Button
23             android:layout_width="wrap_content"
24             android:layout_height="wrap_content"
25             android:layout_margin="15sp"
26             android:text="back"
27             android:id="@+id/back"
28             android:layout_toRightOf="@+id/update"/>
29         <Button
30             android:layout_width="wrap_content"
31             android:layout_height="wrap_content"
32             android:layout_margin="15sp"
33             android:text="Update"
34             android:id="@+id/update"/>
35     </RelativeLayout>
36 </LinearLayout>
```

Lab report 3

display_list.xml:

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     android:orientation="horizontal"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content">
5     <TextView
6         android:layout_width="40dp"
7         android:layout_height="40dp"
8         android:id="@+id/stu_id"
9         android:textSize="20sp"
10        android:text="1"
11        android:gravity="center"
12        android:layout_margin="15sp"/>
13    <TextView android:id="@+id/Name"
14        android:text="Name"
15        android:textSize="20sp"
16        android:layout_width="150dp"
17        android:layout_height="40dp"
18        android:layout_margin="15sp"
19        android:gravity="center"/>
20    <TextView android:id="@+id/Roll"
21        android:text="Roll "
22        android:textSize="20sp"
23        android:layout_width="60dp"
24        android:layout_height="40dp"
25        android:layout_margin="15sp"
26        android:gravity="center"/>
27 </LinearLayout>
```

Java files:

Student.java:

```
1 package com.example.crudapp;
2 public class Student {
3     private String id, name, roll;
4
5     public Student(String id,String name, String roll) {
6         this.id= id;
7         this.name = name;
8         this.roll = roll;
9     }
10    public String getId() {
11        return id;
12    }
13    public void setId(String id) {
14        this.id = id;
15    }
16    public String getName() {
17        return name;
18    }
19    public void setName(String name) {
20        this.name = name;
21    }
22    public String getRoll() {
23        return roll;
24    }
25    public void setRoll(String roll) {
26        this.roll = roll;
27    }
28 }
```

Lab report 3

insert.java:

```
1 public class insert extends Fragment {
2     EditText etName, etRoll;
3     Button save, back;
4     @Override
5     public View onCreateView(LayoutInflater inflater, ViewGroup container,
6                             Bundle savedInstanceState) {
7         View v = inflater.inflate(R.layout.fragment_insert, container, false);
8         etName = v.findViewById(R.id.etName);
9         etRoll = v.findViewById(R.id.etRoll);
10        save = v.findViewById(R.id.save);
11        back = v.findViewById(R.id.back);
12        save.setOnClickListener(new View.OnClickListener() {
13            @Override
14            public void onClick(View v) {
15                insertData();
16            }
17        });
18        back.setOnClickListener(new View.OnClickListener() {
19            @Override
20            public void onClick(View v) {
21                getActivity().getSupportFragmentManager().popBackStack();
22            }
23        });
24        return v;
25    }
26    private void insertData() {
27        String name = etName.getText().toString().trim();
28        String roll = etRoll.getText().toString().trim();
29        if (name.isEmpty()) {
30            Toast.makeText(getContext(), "Enter name please", Toast.LENGTH_SHORT).show();
31            return;
32        } else if (roll.isEmpty()) {
33            Toast.makeText(getContext(), "Enter roll no please", Toast.LENGTH_SHORT).show();
34            return;
35        } else {
36            String url = "http://192.168.0.101/students/insert.php";
37            StringRequest request = new StringRequest(Request.Method.POST, url,
38                new Response.Listener<String>() {
39                    @Override
40                    public void onResponse(String response) {
41                        if (response.equalsIgnoreCase("Data Inserted")) {
42                            Toast.makeText(getContext(), "Data Inserted", Toast.LENGTH_SHORT).show();
43                        } else {
44                            Toast.makeText(getContext(), response, Toast.LENGTH_SHORT).show();
45                        }
46                    }
47                },
48                new Response.ErrorListener() {
49                    @Override
50                    public void onErrorResponse(VolleyError error) {
51                        if (error instanceof NoConnectionError) {
52                            Toast.makeText(getContext(), "No network connection", Toast.LENGTH_SHORT).show();
53                        } else {
54                            Toast.makeText(getContext(), error.toString(), Toast.LENGTH_SHORT).show();
55                        }
56                    }
57                }) {
58                @Nullable
59                @Override
60                protected Map<String, String> getParams() throws AuthFailureError {
61                    Map<String, String> params = new HashMap<String, String>();
62                    params.put("name", name);
63                    params.put("roll", roll);
64                    return params;
65                }
66            };
67            RequestQueue requestQueue = Volley.newRequestQueue(requireContext());
68            requestQueue.add(request);
69        }
70    }
71 }
```


Lab report 3

display.java:

```
1 public class display extends Fragment {
2     ListView listView;
3     MyAdapter adapter;
4     ArrayList<Student> studentArrayList = new ArrayList<>();
5     String fetchUrl = "http://192.168.0.101/students/fetch.php";
6     String deleteUrl = "http://192.168.0.101/students/delete.php";
7     @Override
8     public View onCreateView(LayoutInflater inflater, ViewGroup container,
9         Bundle savedInstanceState) {
10         View view = inflater.inflate(R.layout.fragment_display, container, false);
11         listView = view.findViewById(R.id.list);
12         adapter = new MyAdapter(requireContext(), studentArrayList);
13         listView.setAdapter(adapter);
14
15         listView.setOnItemClickListener((parent, view1, position, id) -> {
16             showStudentDetailsDialog(studentArrayList.get(position));
17         });
18         registerForContextMenu(listView);
19         fetchData();
20         return view;
21     }
22     private void fetchData() {
23         new FetchDataTask().execute();
24     }
25     private void showStudentDetailsDialog(Student student) {
26         new AlertDialog.Builder(getContext())
27             .setTitle("Student Details")
28             .setMessage("ID: " + student.getId() + "\nName: " + student.getName() + "\nRoll: " + student.getRoll())
29             .setPositiveButton("OK", null)
30             .show();
31     }
32     @Override
33     public void onCreateContextMenu(@NonNull ContextMenu menu, @NonNull View v,
34         ContextMenu.ContextMenuInfo menuInfo) {
35         super.onCreateContextMenu(menu, v, menuInfo);
36         if (v.getId() == R.id.list) {
37             requireActivity().getMenuInflater().inflate(R.menu.main2, menu);
38         }
39     }
40     @Override
41     public boolean onContextItemSelected(@NonNull MenuItem item) {
42         if (!isVisible()) return false;
43         AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
44         Student student = studentArrayList.get(info.position);
45         if (item.getItemId() == R.id.menu_update) {
46             Update updateFragment = new Update();
47             Bundle args = new Bundle();
48             args.putString("id", student.getId());
49             args.putString("name", student.getName());
50             args.putString("roll", student.getRoll());
51             updateFragment.setArguments(args);
52             FragmentManager manager = requireActivity().getSupportFragmentManager();
53             FragmentTransaction transaction = manager.beginTransaction();
54             transaction.replace(R.id.frag, updateFragment);
55             transaction.addToBackStack(null);
56             transaction.commit();
57             return true;
58         } else if (item.getItemId() == R.id.menu_delete) {
59             confirmDelete(student.getId());
60             return true;
61         } else {
62             return super.onContextItemSelected(item);
63         }
64     }
```

Lab report 3

```
65     private void confirmDelete(String studentId) {
66         new AlertDialog.Builder(getContext())
67             .setTitle("Delete Student")
68             .setMessage("Are you sure you want to delete this student?")
69             .setPositiveButton("Delete", (dialog, which) -> deleteStudent(studentId))
70             .setNegativeButton("Cancel", null)
71             .show();
72     }
73     private void deleteStudent(String studentId) {
74         new DeleteTask().execute(studentId);
75     }
76     private class FetchDataTask extends AsyncTask<Void, Void, String> {
77         @Override
78         protected String doInBackground(Void... voids) {
79             try {
80                 HttpURLConnection conn = (HttpURLConnection) new URL(fetchUrl).openConnection();
81                 InputStream is = conn.getInputStream();
82                 Scanner scanner = new Scanner(is).useDelimiter("\\A");
83                 return scanner.hasNext() ? scanner.next() : "";
84             } catch (Exception e) {
85                 return null;
86             }
87         }
88         @Override
89         protected void onPostExecute(String result) {
90             if (result == null) {
91                 Toast.makeText(getContext(), "Fetch failed", Toast.LENGTH_SHORT).show();
92                 return;
93             }
94             try {
95                 studentArrayList.clear();
96                 JSONArray array = new JSONArray(result);
97                 for (int i = 0; i < array.length(); i++) {
98                     JSONObject obj = array.getJSONObject(i);
99                     studentArrayList.add(new Student(
100                         obj.getString("id"),
101                         obj.getString("name"),
102                         obj.getString("roll")
103                     ));
104                 }
105                 adapter.notifyDataSetChanged();
106             } catch (Exception e) {
107                 Toast.makeText(getContext(), "Data parse error", Toast.LENGTH_SHORT).show();
108             }
109         }
110     }
111     private class DeleteTask extends AsyncTask<String, Void, Boolean> {
112         @Override
113         protected Boolean doInBackground(String... ids) {
114             try {
115                 URL url = new URL(deleteUrl);
116                 HttpURLConnection conn = (HttpURLConnection) url.openConnection();
117                 conn.setRequestMethod("POST");
118                 conn.setDoOutput(true);
119                 conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
120
121                 String postData = "id=" + ids[0];
122                 OutputStream os = conn.getOutputStream();
123                 os.write(postData.getBytes());
124                 os.flush();
125                 os.close();
126
127                 return conn.getResponseCode() == 200;
128             } catch (Exception e) {
129                 e.printStackTrace();
130                 return false;
131             }
132         }
133         @Override
134         protected void onPostExecute(Boolean success) {
135             if (success) {
136                 Toast.makeText(getContext(), "Deleted successfully", Toast.LENGTH_SHORT).show();
137                 fetchData();
138             } else {
139                 Toast.makeText(getContext(), "Delete failed", Toast.LENGTH_SHORT).show();
140             }
141         }
142     }
143 }
144 }
```

Lab report 3

MyAdapter.java:

```
1 public class MyAdapter extends ArrayAdapter<Student> {
2     private final Context context;
3     private final List<Student> arrayListStudent;
4     public MyAdapter(@NonNull Context context, List<Student> arrayListStudent) {
5         super(context, R.layout.display_list, arrayListStudent);
6         this.context = context;
7         this.arrayListStudent = arrayListStudent;
8     }
9     private static class ViewHolder {
10         TextView stu_id;
11         TextView Name;
12         TextView Roll;
13     }
14     @NonNull
15     @Override
16     public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
17         ViewHolder holder;
18         if (convertView == null) {
19             convertView = LayoutInflater.from(context).inflate(R.layout.display_list, parent, false);
20             holder = new ViewHolder();
21             holder.stu_id = convertView.findViewById(R.id.stu_id);
22             holder.Name = convertView.findViewById(R.id.Name);
23             holder.Roll = convertView.findViewById(R.id.Roll);
24             convertView.setTag(holder);
25         } else {
26             holder = (ViewHolder) convertView.getTag();
27         }
28         Student student = arrayListStudent.get(position);
29         holder.stu_id.setText(student.getId());
30         holder.Name.setText(student.getName());
31         holder.Roll.setText(student.getRoll());
32         return convertView;
33     }
34 }
```

MainActivity.java:

```
1 public class MainActivity extends AppCompatActivity {
2     @Override
3     protected void onCreate(Bundle savedInstanceState) {
4         super.onCreate(savedInstanceState);
5         setContentView(R.layout.layout);
6     }
7     @Override
8     public boolean onCreateOptionsMenu(Menu menu) {
9         super.onCreateOptionsMenu(menu);
10        MenuInflater inflater = new MenuInflater(this);
11        inflater.inflate(R.menu.main, menu);
12        return true;
13    }
14    @Override
15    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
16        if (item.getItemId() == R.id.insert) {
17            insert s = new insert();
18            FragmentManager manager = getSupportFragmentManager();
19            FragmentTransaction tran = manager.beginTransaction();
20            tran.replace(R.id.frag, s);
21            tran.addToBackStack(null);
22            tran.commit();
23        } else if (item.getItemId() == R.id.display) {
24            display d = new display();
25            FragmentManager manager = getSupportFragmentManager();
26            FragmentTransaction tran = manager.beginTransaction();
27            tran.replace(R.id.frag, d);
28            tran.addToBackStack(null);
29            tran.commit();
30        }
31        return super.onOptionsItemSelected(item);
32    }
33 }
```

Lab report 3

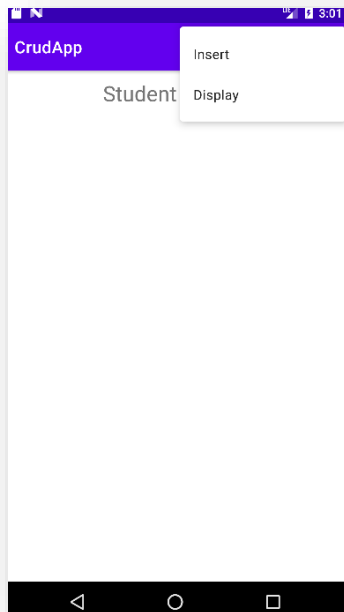
Update.java:

```
1 public class Update extends Fragment {
2     private EditText etName, etRoll;
3     private String studentId;
4     public static Update newInstance(Student student) {
5         Update fragment = new Update();
6         Bundle args = new Bundle();
7         args.putString("id", student.getId());
8         args.putString("name", student.getName());
9         args.putString("roll", student.getRoll());
10        fragment.setArguments(args);
11        return fragment;
12    }
13    @Override
14    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
15        Bundle savedInstanceState) {
16        inflater.inflate(R.layout.fragment_update, container, false);
17    }
18    @Override
19    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
20        super.onViewCreated(view, savedInstanceState);
21        etName = view.findViewById(R.id.etName);
22        etRoll = view.findViewById(R.id.etRoll);
23        Button btnUpdate = view.findViewById(R.id.update);
24        Button btnBack = view.findViewById(R.id.back);
25        if (getArguments() != null) {
26            studentId = getArguments().getString("id");
27            etName.setText(getArguments().getString("name"));
28            etRoll.setText(getArguments().getString("roll"));
29        }
30        btnUpdate.setOnClickListener(v -> updateStudent());
31        btnBack.setOnClickListener(v -> requireActivity().onBackPressed());
32    }
33    private void updateStudent() {
34        String name = etName.getText().toString().trim();
35        String roll = etRoll.getText().toString().trim();
36        if (name.isEmpty() || roll.isEmpty()) {
37            Toast.makeText(getContext(), "Please fill all fields", Toast.LENGTH_SHORT).show();
38            return;
39        }
40        new UpdateTask().execute(studentId, name, roll);
41    }
42    private class UpdateTask extends AsyncTask<String, Void, String> {
43        @Override
44        protected String doInBackground(String... params) {
45            try {
46                URL url = new URL("http://192.168.0.101/students/update.php");
47                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
48                conn.setRequestMethod("POST");
49                conn.setRequestProperty("Content-Type", "application/json");
50                conn.setDoOutput(true);
51                JSONObject jsonParam = new JSONObject();
52                jsonParam.put("id", params[0]);
53                jsonParam.put("name", params[1]);
54                jsonParam.put("roll", params[2]);
55                OutputStream os = conn.getOutputStream();
56                os.write(jsonParam.toString().getBytes(StandardCharsets.UTF_8));
57                os.close();
58                int responseCode = conn.getResponseCode();
59                if (responseCode == HttpURLConnection.HTTP_OK) {
60                    // Parse JSON response
61                    String response = new java.util.Scanner(conn.getInputStream()).useDelimiter("\\A").next();
62                    JSONObject jsonResponse = new JSONObject(response);
63                    return jsonResponse.getString("status");
64                }
65                return "error";
66            } catch (Exception e) {
67                e.printStackTrace();
68                return "error";
69            }
70        }
71        @Override
72        protected void onPostExecute(String result) {
73            if ("success".equals(result)) {
74                Toast.makeText(getContext(), "Update successful", Toast.LENGTH_SHORT).show();
75                requireActivity().onBackPressed();
76            } else {
77                Toast.makeText(getContext(), "Update failed", Toast.LENGTH_SHORT).show();
78            }
79        }
80    }
81 }
```

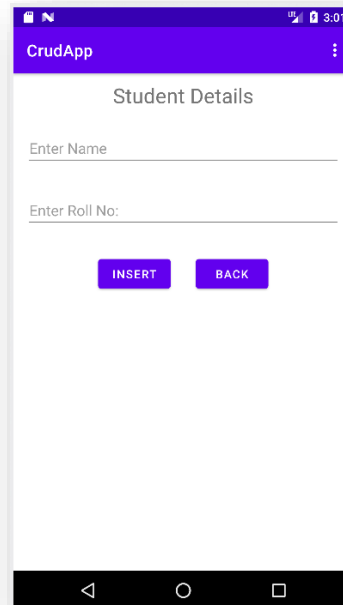
Lab report 3

Output:

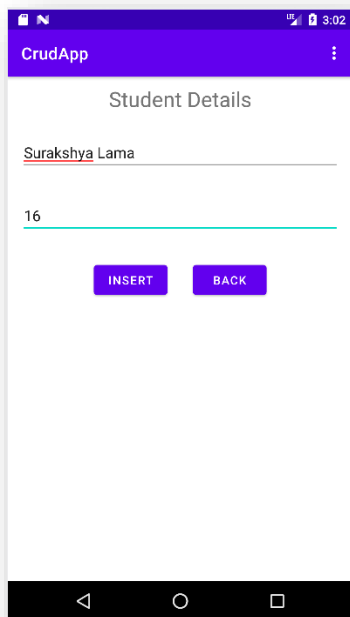
Entry Screen:



Display Screen with Insert Function:



Display Screen with Insert Function:

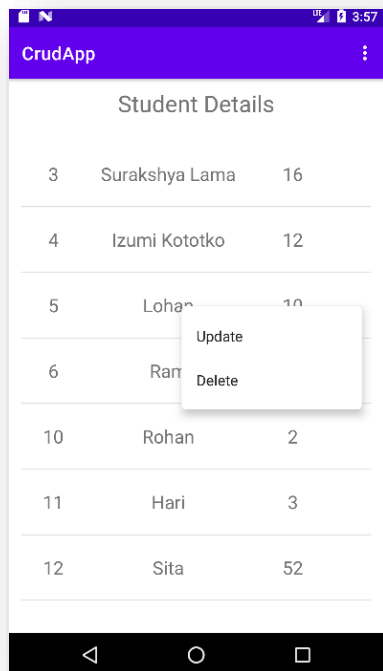


Display Screen with Display Function:

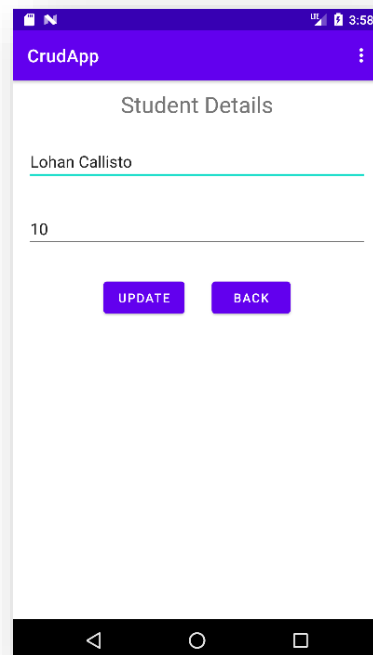


Lab report 3

Display Screen with Update & Delete Function:



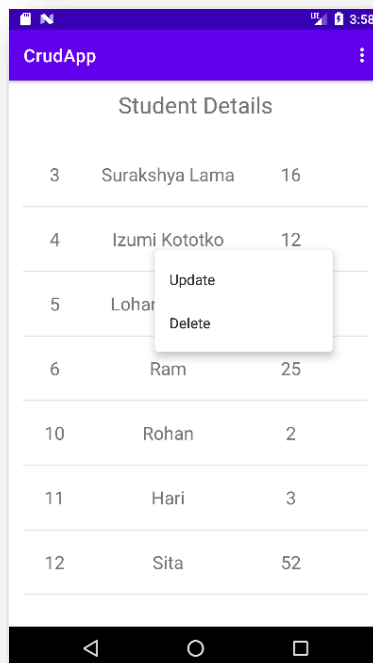
Display Screen with Update Function:



Display Screen After Updating:

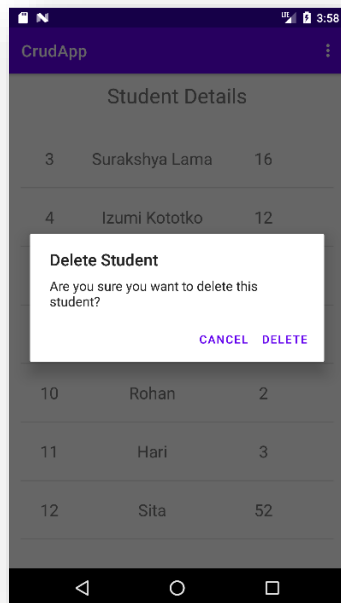


Display Screen with Update & Delete Function:



Lab report 3

Display Screen with Delete Dialog Box:



Display Screen After Deletion:



Display Screen with Detail Dialog Box:

