**Q-3**

**LoginActivity:-**

package com.androidtutorialshub.loginregister.activities;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.design.widget.Snackbar;  
import android.support.design.widget.TextInputEditText;  
import android.support.design.widget.TextInputLayout;  
import android.support.v4.widget.NestedScrollView;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.AppCompatButton;  
import android.support.v7.widget.AppCompatTextView;  
import android.util.Log;  
import android.view.View;  
  
import com.androidtutorialshub.loginregister.R;  
import com.androidtutorialshub.loginregister.helpers.InputValidation;  
import com.androidtutorialshub.loginregister.sql.DatabaseHelper;  
  
public class LoginActivity extends AppCompatActivity implements View.OnClickListener {  
 private final AppCompatActivity activity = LoginActivity.this;  
  
 private NestedScrollView nestedScrollView;  
  
 private TextInputLayout textInputLayoutEmail;  
 private TextInputLayout textInputLayoutPassword;  
  
 private TextInputEditText textInputEditTextEmail;  
 private TextInputEditText textInputEditTextPassword;  
  
 private AppCompatButton appCompatButtonLogin;  
  
 private AppCompatTextView textViewLinkRegister;  
  
 private InputValidation inputValidation;  
 private DatabaseHelper databaseHelper;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
 getSupportActionBar().hide();  
  
 initViews();  
 initListeners();  
 initObjects();  
 }  
  
 private void initViews() {  
  
 nestedScrollView = (NestedScrollView) findViewById(R.id.*nestedScrollView*);  
  
 textInputLayoutEmail = (TextInputLayout) findViewById(R.id.*textInputLayoutEmail*);  
 textInputLayoutPassword = (TextInputLayout) findViewById(R.id.*textInputLayoutPassword*);  
  
 textInputEditTextEmail = (TextInputEditText) findViewById(R.id.*textInputEditTextEmail*);  
 textInputEditTextPassword = (TextInputEditText) findViewById(R.id.*textInputEditTextPassword*);  
  
 appCompatButtonLogin = (AppCompatButton) findViewById(R.id.*appCompatButtonLogin*);  
  
 textViewLinkRegister = (AppCompatTextView) findViewById(R.id.*textViewLinkRegister*);  
  
 }  
  
  
 private void initListeners() {  
 appCompatButtonLogin.setOnClickListener(this);  
 textViewLinkRegister.setOnClickListener(this);  
 }  
  
  
 private void initObjects() {  
 databaseHelper = new DatabaseHelper(activity);  
 inputValidation = new InputValidation(activity);  
  
 }  
  
  
 @Override  
 public void onClick(View v) {  
 switch (v.getId()) {  
 case R.id.*appCompatButtonLogin*:  
 verifyFromSQLite();  
 break;  
 case R.id.*textViewLinkRegister*:  
 // Navigate to RegisterActivity  
 Intent intentRegister = new Intent(getApplicationContext(), RegisterActivity.class);  
 startActivity(intentRegister);  
 break;  
 }  
 }  
  
  
 private void verifyFromSQLite() {  
 if (!inputValidation.isInputEditTextFilled(textInputEditTextEmail, textInputLayoutEmail, getString(R.string.*error\_message\_email*))) {  
 return;  
 }  
 if (!inputValidation.isInputEditTextEmail(textInputEditTextEmail, textInputLayoutEmail, getString(R.string.*error\_message\_email*))) {  
 return;  
 }  
 if (!inputValidation.isInputEditTextFilled(textInputEditTextPassword, textInputLayoutPassword, getString(R.string.*error\_message\_email*))) {  
 return;  
 }  
  
 if (databaseHelper.checkUser(textInputEditTextEmail.getText().toString().trim()  
 , textInputEditTextPassword.getText().toString().trim())) {  
  
  
 Intent accountsIntent = new Intent(activity, UsersListActivity.class);  
 accountsIntent.putExtra("EMAIL", textInputEditTextEmail.getText().toString().trim());  
 emptyInputEditText();  
 startActivity(accountsIntent);  
  
  
 } else {  
 // Snack Bar to show success message that record is wrong  
 Snackbar.make(nestedScrollView, getString(R.string.*error\_valid\_email\_password*), Snackbar.LENGTH\_LONG).show();  
 }  
 }  
  
  
 private void emptyInputEditText() {  
 textInputEditTextEmail.setText(null);  
 textInputEditTextPassword.setText(null);  
 }  
}

**RegisterActivity:-**

package com.androidtutorialshub.loginregister.activities;  
  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.support.design.widget.Snackbar;  
import android.support.design.widget.TextInputEditText;  
import android.support.design.widget.TextInputLayout;  
import android.support.v4.widget.NestedScrollView;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.AppCompatButton;  
import android.support.v7.widget.AppCompatTextView;  
import android.view.View;  
  
import com.androidtutorialshub.loginregister.R;  
import com.androidtutorialshub.loginregister.helpers.InputValidation;  
import com.androidtutorialshub.loginregister.model.User;  
import com.androidtutorialshub.loginregister.sql.DatabaseHelper;  
  
public class RegisterActivity extends AppCompatActivity implements View.OnClickListener {  
  
 private final AppCompatActivity activity = RegisterActivity.this;  
  
 private NestedScrollView nestedScrollView;  
  
 private TextInputLayout textInputLayoutName;  
 private TextInputLayout textInputLayoutEmail;  
 private TextInputLayout textInputLayoutPassword;  
 private TextInputLayout textInputLayoutConfirmPassword;  
  
 private TextInputEditText textInputEditTextName;  
 private TextInputEditText textInputEditTextEmail;  
 private TextInputEditText textInputEditTextPassword;  
 private TextInputEditText textInputEditTextConfirmPassword;  
  
 private AppCompatButton appCompatButtonRegister;  
 private AppCompatTextView appCompatTextViewLoginLink;  
  
 private InputValidation inputValidation;  
 private DatabaseHelper databaseHelper;  
 private User user;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
 getSupportActionBar().hide();  
  
 initViews();  
 initListeners();  
 initObjects();  
 }  
  
 private void initViews() {  
 nestedScrollView = (NestedScrollView) findViewById(R.id.*nestedScrollView*);  
  
 textInputLayoutName = (TextInputLayout) findViewById(R.id.*textInputLayoutName*);  
 textInputLayoutEmail = (TextInputLayout) findViewById(R.id.*textInputLayoutEmail*);  
 textInputLayoutPassword = (TextInputLayout) findViewById(R.id.*textInputLayoutPassword*);  
 textInputLayoutConfirmPassword = (TextInputLayout) findViewById(R.id.*textInputLayoutConfirmPassword*);  
  
 textInputEditTextName = (TextInputEditText) findViewById(R.id.*textInputEditTextName*);  
 textInputEditTextEmail = (TextInputEditText) findViewById(R.id.*textInputEditTextEmail*);  
 textInputEditTextPassword = (TextInputEditText) findViewById(R.id.*textInputEditTextPassword*);  
 textInputEditTextConfirmPassword = (TextInputEditText) findViewById(R.id.*textInputEditTextConfirmPassword*);  
  
 appCompatButtonRegister = (AppCompatButton) findViewById(R.id.*appCompatButtonRegister*);  
  
 appCompatTextViewLoginLink = (AppCompatTextView) findViewById(R.id.*appCompatTextViewLoginLink*);  
  
 }  
  
 private void initListeners() {  
 appCompatButtonRegister.setOnClickListener(this);  
 appCompatTextViewLoginLink.setOnClickListener(this);  
  
 }  
  
 private void initObjects() {  
 inputValidation = new InputValidation(activity);  
 databaseHelper = new DatabaseHelper(activity);  
 user = new User();  
  
 }  
 @Override  
 public void onClick(View v) {  
 switch (v.getId()) {  
  
 case R.id.*appCompatButtonRegister*:  
 postDataToSQLite();  
 break;  
  
 case R.id.*appCompatTextViewLoginLink*:  
 finish();  
 break;  
 }  
 }  
  
 private void postDataToSQLite() {  
 if (!inputValidation.isInputEditTextFilled(textInputEditTextName, textInputLayoutName, getString(R.string.*error\_message\_name*))) {  
 return;  
 }  
 if (!inputValidation.isInputEditTextFilled(textInputEditTextEmail, textInputLayoutEmail, getString(R.string.*error\_message\_email*))) {  
 return;  
 }  
 if (!inputValidation.isInputEditTextEmail(textInputEditTextEmail, textInputLayoutEmail, getString(R.string.*error\_message\_email*))) {  
 return;  
 }  
 if (!inputValidation.isInputEditTextFilled(textInputEditTextPassword, textInputLayoutPassword, getString(R.string.*error\_message\_password*))) {  
 return;  
 }  
 if (!inputValidation.isInputEditTextMatches(textInputEditTextPassword, textInputEditTextConfirmPassword,  
 textInputLayoutConfirmPassword, getString(R.string.*error\_password\_match*))) {  
 return;  
 }  
  
 if (!databaseHelper.checkUser(textInputEditTextEmail.getText().toString().trim())) {  
  
 user.setName(textInputEditTextName.getText().toString().trim());  
 user.setEmail(textInputEditTextEmail.getText().toString().trim());  
 user.setPassword(textInputEditTextPassword.getText().toString().trim());  
  
 databaseHelper.addUser(user);  
  
 // Snack Bar to show success message that record saved successfully  
 Snackbar.make(nestedScrollView, getString(R.string.*success\_message*), Snackbar.LENGTH\_LONG).show();  
 emptyInputEditText();  
  
  
 } else {  
 // Snack Bar to show error message that record already exists  
 Snackbar.make(nestedScrollView, getString(R.string.*error\_email\_exists*), Snackbar.LENGTH\_LONG).show();  
 }  
 }  
 private void emptyInputEditText() {  
 textInputEditTextName.setText(null);  
 textInputEditTextEmail.setText(null);  
 textInputEditTextPassword.setText(null);  
 textInputEditTextConfirmPassword.setText(null);  
 }  
}

**UsersListActivity:-**

package com.androidtutorialshub.loginregister.activities;  
  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.AppCompatTextView;  
import android.support.v7.widget.DefaultItemAnimator;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
  
import com.androidtutorialshub.loginregister.R;  
import com.androidtutorialshub.loginregister.adapters.UsersRecyclerAdapter;  
import com.androidtutorialshub.loginregister.model.User;  
import com.androidtutorialshub.loginregister.sql.DatabaseHelper;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class UsersListActivity extends AppCompatActivity {  
  
 private AppCompatActivity activity = UsersListActivity.this;  
 private AppCompatTextView textViewName;  
 private RecyclerView recyclerViewUsers;  
 private List<User> listUsers;  
 private UsersRecyclerAdapter usersRecyclerAdapter;  
 private DatabaseHelper databaseHelper;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_users\_list*);  
 getSupportActionBar().setTitle("");  
 initViews();  
 initObjects();  
  
 }  
  
 private void initViews() {  
 textViewName = (AppCompatTextView) findViewById(R.id.*textViewName*);  
 recyclerViewUsers = (RecyclerView) findViewById(R.id.*recyclerViewUsers*);  
 }  
  
  
 private void initObjects() {  
 listUsers = new ArrayList<>();  
 usersRecyclerAdapter = new UsersRecyclerAdapter(listUsers);  
  
 RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(getApplicationContext());  
 recyclerViewUsers.setLayoutManager(mLayoutManager);  
 recyclerViewUsers.setItemAnimator(new DefaultItemAnimator());  
 recyclerViewUsers.setHasFixedSize(true);  
 recyclerViewUsers.setAdapter(usersRecyclerAdapter);  
 databaseHelper = new DatabaseHelper(activity);  
  
 String emailFromIntent = getIntent().getStringExtra("EMAIL");  
 textViewName.setText(emailFromIntent);  
  
 getDataFromSQLite();  
 }  
  
 private void getDataFromSQLite() {  
 // AsyncTask is used that SQLite operation not blocks the UI Thread.  
 new AsyncTask<Void, Void, Void>() {  
 @Override  
 protected Void doInBackground(Void... params) {  
 listUsers.clear();  
 listUsers.addAll(databaseHelper.getAllUser());  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void aVoid) {  
 super.onPostExecute(aVoid);  
 usersRecyclerAdapter.notifyDataSetChanged();  
 }  
 }.execute();  
 }  
}

**UsersRecyclerAdapter:-**

package com.androidtutorialshub.loginregister.adapters;  
  
import android.support.v7.widget.AppCompatTextView;  
import android.support.v7.widget.RecyclerView;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
import com.androidtutorialshub.loginregister.R;  
import com.androidtutorialshub.loginregister.model.User;  
  
import java.util.List;  
  
public class UsersRecyclerAdapter extends RecyclerView.Adapter<UsersRecyclerAdapter.UserViewHolder> {  
  
 private List<User> listUsers;  
  
 public UsersRecyclerAdapter(List<User> listUsers) {  
 this.listUsers = listUsers;  
 }  
  
 @Override  
 public UserViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 // inflating recycler item view  
 View itemView = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*item\_user\_recycler*, parent, false);  
  
 return new UserViewHolder(itemView);  
 }  
  
 @Override  
 public void onBindViewHolder(UserViewHolder holder, int position) {  
 holder.textViewName.setText(listUsers.get(position).getName());  
 holder.textViewEmail.setText(listUsers.get(position).getEmail());  
 holder.textViewPassword.setText(listUsers.get(position).getPassword());  
 }  
  
 @Override  
 public int getItemCount() {  
 Log.*v*(UsersRecyclerAdapter.class.getSimpleName(),""+listUsers.size());  
 return listUsers.size();  
 }  
  
  
 public class UserViewHolder extends RecyclerView.ViewHolder {  
  
 public AppCompatTextView textViewName;  
 public AppCompatTextView textViewEmail;  
 public AppCompatTextView textViewPassword;  
  
 public UserViewHolder(View view) {  
 super(view);  
 textViewName = (AppCompatTextView) view.findViewById(R.id.*textViewName*);  
 textViewEmail = (AppCompatTextView) view.findViewById(R.id.*textViewEmail*);  
 textViewPassword = (AppCompatTextView) view.findViewById(R.id.*textViewPassword*);  
 }  
 }  
  
  
}

**InputValidation :-**

package com.androidtutorialshub.loginregister.helpers;  
  
import android.app.Activity;  
import android.content.Context;  
import android.support.design.widget.TextInputEditText;  
import android.support.design.widget.TextInputLayout;  
import android.view.View;  
import android.view.WindowManager;  
import android.view.inputmethod.InputMethodManager;  
  
*/\*\*  
 \* Created by lalit on 9/13/2016.  
 \*/*public class InputValidation {  
 private Context context;  
public InputValidation(Context context) {  
 this.context = context;  
 }  
  
public boolean isInputEditTextFilled(TextInputEditText textInputEditText, TextInputLayout textInputLayout, String message) {  
 String value = textInputEditText.getText().toString().trim();  
 if (value.isEmpty()) {  
 textInputLayout.setError(message);  
 hideKeyboardFrom(textInputEditText);  
 return false;  
 } else {  
 textInputLayout.setErrorEnabled(false);  
 }  
  
 return true;  
 }  
public boolean isInputEditTextEmail(TextInputEditText textInputEditText, TextInputLayout textInputLayout, String message) {  
 String value = textInputEditText.getText().toString().trim();  
 if (value.isEmpty() || !android.util.Patterns.*EMAIL\_ADDRESS*.matcher(value).matches()) {  
 textInputLayout.setError(message);  
 hideKeyboardFrom(textInputEditText);  
 return false;  
 } else {  
 textInputLayout.setErrorEnabled(false);  
 }  
 return true;  
 }  
  
 public boolean isInputEditTextMatches(TextInputEditText textInputEditText1, TextInputEditText textInputEditText2, TextInputLayout textInputLayout, String message) {  
 String value1 = textInputEditText1.getText().toString().trim();  
 String value2 = textInputEditText2.getText().toString().trim();  
 if (!value1.contentEquals(value2)) {  
 textInputLayout.setError(message);  
 hideKeyboardFrom(textInputEditText2);  
 return false;  
 } else {  
 textInputLayout.setErrorEnabled(false);  
 }  
 return true;  
 }  
private void hideKeyboardFrom(View view) {  
 InputMethodManager imm = (InputMethodManager) context.getSystemService(Activity.*INPUT\_METHOD\_SERVICE*);  
 imm.hideSoftInputFromWindow(view.getWindowToken(), WindowManager.LayoutParams.*SOFT\_INPUT\_STATE\_ALWAYS\_HIDDEN*);  
 }  
}

**User:-**

package com.androidtutorialshub.loginregister.model;  
public class User {  
  
 private int id;  
 private String name;  
 private String email;  
 private String password;  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
}

**DatabaseHelper:-**

package com.androidtutorialshub.loginregister.sql;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import com.androidtutorialshub.loginregister.model.User;  
  
import java.util.ArrayList;  
import java.util.List;public class DatabaseHelper extends SQLiteOpenHelper {  
 private static final int *DATABASE\_VERSION* = 1;  
  
 // Database Name  
 private static final String *DATABASE\_NAME* = "UserManager.db";  
  
 // User table name  
 private static final String *TABLE\_USER* = "user";  
  
 // User Table Columns names  
 private static final String *COLUMN\_USER\_ID* = "user\_id";  
 private static final String *COLUMN\_USER\_NAME* = "user\_name";  
 private static final String *COLUMN\_USER\_EMAIL* = "user\_email";  
 private static final String *COLUMN\_USER\_PASSWORD* = "user\_password";  
  
 // create table sql query  
 private String CREATE\_USER\_TABLE = "CREATE TABLE " + *TABLE\_USER* + "("  
 + *COLUMN\_USER\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT," + *COLUMN\_USER\_NAME* + " TEXT,"  
 + *COLUMN\_USER\_EMAIL* + " TEXT," + *COLUMN\_USER\_PASSWORD* + " TEXT" + ")";  
  
 // drop table sql query  
 private String DROP\_USER\_TABLE = "DROP TABLE IF EXISTS " + *TABLE\_USER*;  
public DatabaseHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL(CREATE\_USER\_TABLE);  
 }  
  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
 //Drop User Table if exist  
 db.execSQL(DROP\_USER\_TABLE);  
  
 // Create tables again  
 onCreate(db);  
  
 }  
public void addUser(User user) {  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(*COLUMN\_USER\_NAME*, user.getName());  
 values.put(*COLUMN\_USER\_EMAIL*, user.getEmail());  
 values.put(*COLUMN\_USER\_PASSWORD*, user.getPassword());  
  
 // Inserting Row  
 db.insert(*TABLE\_USER*, null, values);  
 db.close();  
 }  
public List<User> getAllUser() {  
 // array of columns to fetch  
 String[] columns = {  
 *COLUMN\_USER\_ID*,  
 *COLUMN\_USER\_EMAIL*,  
 *COLUMN\_USER\_NAME*,  
 *COLUMN\_USER\_PASSWORD* };  
 // sorting orders  
 String sortOrder =  
 *COLUMN\_USER\_NAME* + " ASC";  
 List<User> userList = new ArrayList<User>();  
  
 SQLiteDatabase db = this.getReadableDatabase();  
  
 // query the user table  
 */\*\*  
 \* Here query function is used to fetch records from user table this function works like we use sql query.  
 \* SQL query equivalent to this query function is  
 \* SELECT user\_id,user\_name,user\_email,user\_password FROM user ORDER BY user\_name;  
 \*/* Cursor cursor = db.query(*TABLE\_USER*, //Table to query  
 columns, //columns to return  
 null, //columns for the WHERE clause  
 null, //The values for the WHERE clause  
 null, //group the rows  
 null, //filter by row groups  
 sortOrder); //The sort order  
  
  
 // Traversing through all rows and adding to list  
 if (cursor.moveToFirst()) {  
 do {  
 User user = new User();  
 user.setId(Integer.*parseInt*(cursor.getString(cursor.getColumnIndex(*COLUMN\_USER\_ID*))));  
 user.setName(cursor.getString(cursor.getColumnIndex(*COLUMN\_USER\_NAME*)));  
 user.setEmail(cursor.getString(cursor.getColumnIndex(*COLUMN\_USER\_EMAIL*)));  
 user.setPassword(cursor.getString(cursor.getColumnIndex(*COLUMN\_USER\_PASSWORD*)));  
 // Adding user record to list  
 userList.add(user);  
 } while (cursor.moveToNext());  
 }  
 cursor.close();  
 db.close();  
  
 // return user list  
 return userList;  
 }  
  
 */\*\*  
 \* This method to update user record  
 \*  
 \** ***@param*** *user  
 \*/* public void updateUser(User user) {  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(*COLUMN\_USER\_NAME*, user.getName());  
 values.put(*COLUMN\_USER\_EMAIL*, user.getEmail());  
 values.put(*COLUMN\_USER\_PASSWORD*, user.getPassword());  
  
 // updating row  
 db.update(*TABLE\_USER*, values, *COLUMN\_USER\_ID* + " = ?",  
 new String[]{String.*valueOf*(user.getId())});  
 db.close();  
 }  
public void deleteUser(User user) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 // delete user record by id  
 db.delete(*TABLE\_USER*, *COLUMN\_USER\_ID* + " = ?",  
 new String[]{String.*valueOf*(user.getId())});  
 db.close();  
 }  
public boolean checkUser(String email) {  
  
 // array of columns to fetch  
 String[] columns = {  
 *COLUMN\_USER\_ID* };  
 SQLiteDatabase db = this.getReadableDatabase();  
  
 // selection criteria  
 String selection = *COLUMN\_USER\_EMAIL* + " = ?";  
  
 // selection argument  
 String[] selectionArgs = {email};  
  
 // query user table with condition  
 */\*\*  
 \* Here query function is used to fetch records from user table this function works like we use sql query.  
 \* SQL query equivalent to this query function is  
 \* SELECT user\_id FROM user WHERE user\_email = 'jack@androidtutorialshub.com';  
 \*/* Cursor cursor = db.query(*TABLE\_USER*, //Table to query  
 columns, //columns to return  
 selection, //columns for the WHERE clause  
 selectionArgs, //The values for the WHERE clause  
 null, //group the rows  
 null, //filter by row groups  
 null); //The sort order  
 int cursorCount = cursor.getCount();  
 cursor.close();  
 db.close();  
  
 if (cursorCount > 0) {  
 return true;  
 }  
  
 return false;  
 }  
public boolean checkUser(String email, String password) {  
  
 // array of columns to fetch  
 String[] columns = {  
 *COLUMN\_USER\_ID* };  
 SQLiteDatabase db = this.getReadableDatabase();  
 // selection criteria  
 String selection = *COLUMN\_USER\_EMAIL* + " = ?" + " AND " + *COLUMN\_USER\_PASSWORD* + " = ?";  
  
 // selection arguments  
 String[] selectionArgs = {email, password};  
  
 // query user table with conditions  
 */\*\*  
 \* Here query function is used to fetch records from user table this function works like we use sql query.  
 \* SQL query equivalent to this query function is  
 \* SELECT user\_id FROM user WHERE user\_email = 'jack@androidtutorialshub.com' AND user\_password = 'qwerty';  
 \*/* Cursor cursor = db.query(*TABLE\_USER*, //Table to query  
 columns, //columns to return  
 selection, //columns for the WHERE clause  
 selectionArgs, //The values for the WHERE clause  
 null, //group the rows  
 null, //filter by row groups  
 null); //The sort order  
  
 int cursorCount = cursor.getCount();  
  
 cursor.close();  
 db.close();  
 if (cursorCount > 0) {  
 return true;  
 }  
  
 return false;  
 }  
}

**Output:**