**Library Management System Codes**

**M.N.F.Nifra**

**ICT/19/20/135**

**5069**

**MainActivity.java**

package com.example.common;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 }  
  
 public void navigateToAddBook(View view) {  
 Intent intent = new Intent(this, AddBookActivity.class);  
 startActivity(intent);  
 }  
  
 public void navigateToAddMember(View view) {  
 Intent intent = new Intent(this, AddMemberActivity.class);  
 startActivity(intent);  
 }  
  
  
 public void navigateToViewBook(View view) {  
 Intent intent = new Intent(this, BookManagementActivity.class);  
 startActivity(intent);  
 }  
  
 public void navigateToViewMember(View view) {  
 Intent intent = new Intent(this, MemberManagementActivity.class);  
 startActivity(intent);  
 }  
  
  
}

**AddBookActivity.java**

package com.example.common;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class AddBookActivity extends AppCompatActivity {  
  
 private EditText titleEditText, authorEditText, isbnEditText;  
 private Button saveButton;  
 private DBHelper dbHelper;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_add\_book);  
  
 dbHelper = new DBHelper(this);  
 titleEditText = findViewById(R.id.titleEditText);  
 authorEditText = findViewById(R.id.authorEditText);  
 isbnEditText = findViewById(R.id.isbnEditText);  
 saveButton = findViewById(R.id.saveButton);  
  
 saveButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 addBook();  
 }  
 });  
 }  
  
 private void addBook() {  
 String title = titleEditText.getText().toString().trim();  
 String author = authorEditText.getText().toString().trim();  
 String isbn = isbnEditText.getText().toString().trim();  
  
 if (title.isEmpty() || author.isEmpty() || isbn.isEmpty()) {  
 Toast.makeText(this, "Please fill all fields", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 long result = dbHelper.addBook(title, author, isbn);  
 if (result != -1) {  
 Toast.makeText(this, "Book added successfully", Toast.LENGTH\_SHORT).show();  
 finish(); // Close activity after adding book  
 } else {  
 Toast.makeText(this, "Failed to add book", Toast.LENGTH\_SHORT).show();  
 }  
 }  
}

**AddMemberActivity.java**

package com.example.common;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class AddMemberActivity extends AppCompatActivity {  
  
 private EditText nameEditText, contactEditText, addressEditText, unpaidDuesEditText, branchNameEditText, branchIDEditText, branchAddressEditText;  
 private Button saveButton;  
 private DBHelper dbHelper;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_add\_member);  
  
 dbHelper = new DBHelper(this);  
 nameEditText = findViewById(R.id.nameEditText);  
 contactEditText = findViewById(R.id.contactEditText);  
 addressEditText = findViewById(R.id.addressEditText);  
 unpaidDuesEditText = findViewById(R.id.unpaidduesEditText);  
 branchNameEditText = findViewById(R.id.branchEditText);  
 branchIDEditText = findViewById(R.id.branchIDEditText);  
 branchAddressEditText = findViewById(R.id.branchaddressEditText);  
 saveButton = findViewById(R.id.saveButton);  
  
 saveButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 addMember();  
 }  
 });  
 }  
  
 private void addMember() {  
 String name = nameEditText.getText().toString().trim();  
 String contact = contactEditText.getText().toString().trim();  
 String address = addressEditText.getText().toString().trim();  
 double unpaidDues = Double.parseDouble(unpaidDuesEditText.getText().toString().trim());  
 String branchName = branchNameEditText.getText().toString().trim();  
 String branchID = branchIDEditText.getText().toString().trim();  
 String branchAddress = branchAddressEditText.getText().toString().trim();  
  
 if (name.isEmpty() || contact.isEmpty() || address.isEmpty() || branchName.isEmpty() || branchID.isEmpty() || branchAddress.isEmpty()) {  
 Toast.makeText(this, "Please fill all fields", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 long result = dbHelper.addMember(name, contact, address, unpaidDues, branchName, branchID, branchAddress);  
 if (result != -1) {  
 Toast.makeText(this, "Member added successfully", Toast.LENGTH\_SHORT).show();  
 finish(); // Close activity after adding member  
 } else {  
 Toast.makeText(this, "Failed to add member", Toast.LENGTH\_SHORT).show();  
 }  
 }  
}

**AuthenticationActivity.java**

package com.example.common;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class AuthenticationActivity extends AppCompatActivity {  
  
 private EditText usernameEditText, passwordEditText;  
 private TextView textView;  
 private DBHelper dbHelper;  
 private boolean isRegisterMode = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_authentication);  
  
 usernameEditText = findViewById(R.id.usernameEditText);  
 passwordEditText = findViewById(R.id.passwordEditText);  
 textView = findViewById(R.id.textView);  
 dbHelper = new DBHelper(this);  
  
 // Set initial text and action based on mode  
 setMode(isRegisterMode);  
 }  
  
 public void performAction(View view) {  
 String username = usernameEditText.getText().toString().trim();  
 String password = passwordEditText.getText().toString().trim();  
  
 if (username.isEmpty() || password.isEmpty()) {  
 Toast.makeText(this, "Please fill all fields", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 if (isRegisterMode) {  
 registerUser(username, password);  
 } else {  
 loginUser(username, password);  
 }  
 }  
  
 private void registerUser(String username, String password) {  
 long result = dbHelper.registerUser(username, password);  
 if (result != -1) {  
 Toast.makeText(this, "Registration successful", Toast.LENGTH\_SHORT).show();  
 setMode(!isRegisterMode);  
 isRegisterMode = !isRegisterMode;  
 } else {  
 Toast.makeText(this, "Registration failed", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 private void loginUser(String username, String password) {  
 boolean loginSuccessful = dbHelper.loginUser(username, password);  
 if (loginSuccessful) {  
 Toast.makeText(this, "Login successful", Toast.LENGTH\_SHORT).show();  
 Intent intent = new Intent(this, MainActivity.class);  
 startActivity(intent);  
 // Navigate to the next activity or perform desired action  
 } else {  
 Toast.makeText(this, "Invalid username or password", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 public void goToLogin(View view) {  
 setMode(!isRegisterMode);  
 isRegisterMode = !isRegisterMode;  
 System.out.println(isRegisterMode);  
 }  
  
 private void setMode(boolean isRegisterMode) {  
 Button actionButton = findViewById(R.id.actionButton); // Cast to Button  
 if (isRegisterMode) {  
 textView.setText("If you already have an account?");  
 actionButton.setText("Register");  
 } else {  
 textView.setText("If you don't have an account?");  
 actionButton.setText("Login");  
 }  
 }  
  
}

**BookManagementActivity.java**

package com.example.common;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.ListView;  
import android.widget.SimpleCursorAdapter;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class BookManagementActivity extends AppCompatActivity {  
  
 private DBHelper dbHelper;  
 private ListView listViewBooks;  
 private Button addBookButton;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_book\_management);  
  
 dbHelper = new DBHelper(this);  
 listViewBooks = findViewById(R.id.listViewBooks);  
 addBookButton = findViewById(R.id.addBookButton);  
  
 displayAllBooks();  
  
 listViewBooks.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 // Get the selected book from the cursor  
 Cursor cursor = (Cursor) parent.getItemAtPosition(position);  
 int bookId = cursor.getInt(cursor.getColumnIndexOrThrow("\_id")); // Correct column name  
  
 // Start UpdateBookActivity and pass bookId as extra  
 Intent intent = new Intent(BookManagementActivity.this, UpdateBookActivity.class);  
 intent.putExtra("bookId", bookId);  
 startActivity(intent);  
 }  
 });  
  
 addBookButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(BookManagementActivity.this, AddBookActivity.class);  
 startActivity(intent);  
 }  
 });  
  
  
 }  
 @Override  
 protected void onResume() {  
 super.onResume();  
 displayAllBooks(); // Refresh the ListView adapter  
 }  
  
 private void displayAllBooks() {  
 Cursor cursor = dbHelper.getAllBooks();  
 if (cursor.getCount() == 0) {  
 Toast.makeText(this, "No books available", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 String[] columns = new String[]{DBHelper.KEY\_TITLE, DBHelper.KEY\_AUTHOR, DBHelper.KEY\_ISBN};  
 int[] to = new int[]{R.id.textViewTitle, R.id.textViewAuthor, R.id.textViewISBN};  
  
 SimpleCursorAdapter adapter = new SimpleCursorAdapter(this, R.layout.book\_item, cursor, columns, to, 0);  
 listViewBooks.setAdapter(adapter);  
 }  
  
}

**DBHelper.java**

package com.example.common;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
public class DBHelper extends SQLiteOpenHelper {  
  
 private static final String DATABASE\_NAME = "library.db";  
 private static final int DATABASE\_VERSION = 1;  
  
 // Table names  
 private static final String TABLE\_USERS = "users";  
 private static final String TABLE\_BOOKS = "books";  
 private static final String TABLE\_MEMBERS = "members";  
  
 // Users table columns  
 private static final String KEY\_USER\_ID = "user\_id";  
 private static final String KEY\_USERNAME = "username";  
 private static final String KEY\_PASSWORD = "password";  
  
 // Books table columns  
 public static final String KEY\_BOOK\_ID = "book\_id";  
 public static final String KEY\_TITLE = "title";  
 public static final String KEY\_AUTHOR = "author";  
 public static final String KEY\_ISBN = "isbn";  
  
 // Members table columns  
 public static final String KEY\_MEMBER\_ID = "member\_id";  
 public static final String KEY\_MEMBER\_NAME = "name";  
 public static final String KEY\_MEMBER\_CONTACT = "contact";  
 public static final String KEY\_MEMBER\_ADDRESS = "address";  
 public static final String KEY\_MEMBER\_UNPAID\_DUES = "unpaid\_dues";  
 public static final String KEY\_MEMBER\_BRANCH\_NAME = "branch\_name";  
 public static final String KEY\_MEMBER\_BRANCH\_ID = "branch\_id";  
 public static final String KEY\_MEMBER\_BRANCH\_ADDRESS = "branch\_address";  
 // Create Users table query  
 private static final String CREATE\_USERS\_TABLE =  
 "CREATE TABLE " + TABLE\_USERS + "("  
 + KEY\_USER\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"  
 + KEY\_USERNAME + " TEXT,"  
 + KEY\_PASSWORD + " TEXT"  
 + ")";  
  
 // Create Books table query  
 private static final String CREATE\_BOOKS\_TABLE =  
 "CREATE TABLE " + TABLE\_BOOKS + "("  
 + KEY\_BOOK\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"  
 + KEY\_TITLE + " TEXT,"  
 + KEY\_AUTHOR + " TEXT,"  
 + KEY\_ISBN + " TEXT"  
 + ")";  
  
 // Create Members table query  
 private static final String CREATE\_MEMBERS\_TABLE =  
 "CREATE TABLE " + TABLE\_MEMBERS + "("  
 + KEY\_MEMBER\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"  
 + KEY\_MEMBER\_NAME + " TEXT,"  
 + KEY\_MEMBER\_CONTACT + " TEXT,"  
 + KEY\_MEMBER\_ADDRESS + " TEXT,"  
 + KEY\_MEMBER\_UNPAID\_DUES + " REAL,"  
 + KEY\_MEMBER\_BRANCH\_NAME + " TEXT,"  
 + KEY\_MEMBER\_BRANCH\_ID + " TEXT,"  
 + KEY\_MEMBER\_BRANCH\_ADDRESS + " TEXT"  
 + ")";  
  
 public DBHelper(Context context) {  
 super(context, DATABASE\_NAME, null, DATABASE\_VERSION);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 // Create required tables  
 db.execSQL(CREATE\_USERS\_TABLE);  
 db.execSQL(CREATE\_BOOKS\_TABLE);  
 db.execSQL(CREATE\_MEMBERS\_TABLE);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 // Drop older tables if they exist  
 db.execSQL("DROP TABLE IF EXISTS " + TABLE\_USERS);  
 db.execSQL("DROP TABLE IF EXISTS " + TABLE\_BOOKS);  
 db.execSQL("DROP TABLE IF EXISTS " + TABLE\_MEMBERS);  
 // Create tables again  
 onCreate(db);  
 }  
  
 // CRUD Operations for Users  
  
 public long registerUser(String username, String password) {  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(KEY\_USERNAME, username);  
 values.put(KEY\_PASSWORD, password);  
  
 long newRowId = db.insert(TABLE\_USERS, null, values);  
 db.close();  
  
 return newRowId;  
 }  
  
 public boolean loginUser(String username, String password) {  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = null;  
 boolean loginSuccessful = false;  
  
 try {  
 String[] projection = {KEY\_USER\_ID};  
 String selection = KEY\_USERNAME + " = ? AND " + KEY\_PASSWORD + " = ?";  
 String[] selectionArgs = {username, password};  
  
 cursor = db.query(TABLE\_USERS, projection, selection, selectionArgs, null, null, null);  
 loginSuccessful = cursor.moveToFirst();  
 } finally {  
 if (cursor != null) {  
 cursor.close();  
 }  
 db.close();  
 }  
  
 return loginSuccessful;  
 }  
  
 // CRUD Operations for Books  
  
 public long addBook(String title, String author, String isbn) {  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(KEY\_TITLE, title);  
 values.put(KEY\_AUTHOR, author);  
 values.put(KEY\_ISBN, isbn);  
  
 long newRowId = db.insert(TABLE\_BOOKS, null, values);  
 db.close();  
  
 return newRowId;  
 }  
  
 public Cursor getAllBooks() {  
 SQLiteDatabase db = this.getReadableDatabase();  
 String[] projection = {KEY\_BOOK\_ID + " AS \_id", KEY\_TITLE, KEY\_AUTHOR, KEY\_ISBN};  
 return db.query(TABLE\_BOOKS, projection, null, null, null, null, null);  
 }  
  
 public Cursor getBookById(int bookId) {  
 SQLiteDatabase db = this.getReadableDatabase();  
 String[] projection = {KEY\_TITLE, KEY\_AUTHOR, KEY\_ISBN};  
 String selection = KEY\_BOOK\_ID + " = ?";  
 String[] selectionArgs = {String.valueOf(bookId)};  
 return db.query(TABLE\_BOOKS, projection, selection, selectionArgs, null, null, null);  
 }  
  
 public int updateBook(int bookId, String title, String author, String isbn) {  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(KEY\_TITLE, title);  
 values.put(KEY\_AUTHOR, author);  
 values.put(KEY\_ISBN, isbn);  
  
 String selection = KEY\_BOOK\_ID + " = ?";  
 String[] selectionArgs = {String.valueOf(bookId)};  
  
 int rowsAffected = db.update(TABLE\_BOOKS, values, selection, selectionArgs);  
 db.close();  
  
 return rowsAffected;  
 }  
  
 // CRUD Operations for Members  
  
 public long addMember(String name, String contact, String address, double unpaidDues, String branchName, String branchId, String branchAddress) {  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(KEY\_MEMBER\_NAME, name);  
 values.put(KEY\_MEMBER\_CONTACT, contact);  
 values.put(KEY\_MEMBER\_ADDRESS, address);  
 values.put(KEY\_MEMBER\_UNPAID\_DUES, unpaidDues);  
 values.put(KEY\_MEMBER\_BRANCH\_NAME, branchName);  
 values.put(KEY\_MEMBER\_BRANCH\_ID, branchId);  
 values.put(KEY\_MEMBER\_BRANCH\_ADDRESS, branchAddress);  
  
 long newRowId = db.insert(TABLE\_MEMBERS, null, values);  
 db.close();  
  
 return newRowId;  
 }  
  
 public Cursor getAllMembers() {  
 SQLiteDatabase db = this.getReadableDatabase();  
 String[] projection = {  
 KEY\_MEMBER\_ID + " AS \_id",  
 KEY\_MEMBER\_NAME,  
 KEY\_MEMBER\_CONTACT,  
 KEY\_MEMBER\_ADDRESS, // Include address  
 KEY\_MEMBER\_UNPAID\_DUES, // Include unpaid dues  
 KEY\_MEMBER\_BRANCH\_NAME, // Include branch name  
 KEY\_MEMBER\_BRANCH\_ID, // Include branch id  
 KEY\_MEMBER\_BRANCH\_ADDRESS // Include branch address  
 };  
 return db.query(TABLE\_MEMBERS, projection, null, null, null, null, null);  
 }  
  
 public Cursor getMemberById(int memberId) {  
 SQLiteDatabase db = this.getReadableDatabase();  
 String[] projection = {  
 KEY\_MEMBER\_NAME,  
 KEY\_MEMBER\_CONTACT,  
 KEY\_MEMBER\_ADDRESS, // Include address  
 KEY\_MEMBER\_UNPAID\_DUES, // Include unpaid dues  
 KEY\_MEMBER\_BRANCH\_NAME, // Include branch name  
 KEY\_MEMBER\_BRANCH\_ID, // Include branch id  
 KEY\_MEMBER\_BRANCH\_ADDRESS // Include branch address  
 };  
 String selection = KEY\_MEMBER\_ID + " = ?";  
 String[] selectionArgs = {String.valueOf(memberId)};  
 return db.query(TABLE\_MEMBERS, projection, selection, selectionArgs, null, null, null);  
 }  
  
  
  
 public int updateMember(int memberId, String name, String contact, String address, double unpaidDues, String branchName, String branchId, String branchAddress) {  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(KEY\_MEMBER\_NAME, name);  
 values.put(KEY\_MEMBER\_CONTACT, contact);  
 values.put(KEY\_MEMBER\_ADDRESS, address);  
 values.put(KEY\_MEMBER\_UNPAID\_DUES, unpaidDues);  
 values.put(KEY\_MEMBER\_BRANCH\_NAME, branchName);  
 values.put(KEY\_MEMBER\_BRANCH\_ID, branchId);  
 values.put(KEY\_MEMBER\_BRANCH\_ADDRESS, branchAddress);  
  
 String selection = KEY\_MEMBER\_ID + " = ?";  
 String[] selectionArgs = {String.valueOf(memberId)};  
  
 int rowsAffected = db.update(TABLE\_MEMBERS, values, selection, selectionArgs);  
 db.close();  
  
 return rowsAffected;  
 }  
  
 public int deleteBook(int bookId) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 String selection = KEY\_BOOK\_ID + " = ?";  
 String[] selectionArgs = {String.valueOf(bookId)};  
 int rowsDeleted = db.delete(TABLE\_BOOKS, selection, selectionArgs);  
 db.close();  
 return rowsDeleted;  
 }  
  
}

**MemberManagementActivity.java**

package com.example.common;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.ListView;  
import android.widget.SimpleCursorAdapter;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MemberManagementActivity extends AppCompatActivity {  
  
 private DBHelper dbHelper;  
 private ListView listViewMembers;  
 private Button addMemberButton;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_member\_management);  
  
 dbHelper = new DBHelper(this);  
 listViewMembers = findViewById(R.id.listViewMembers);  
 addMemberButton = findViewById(R.id.addMemberButton);  
  
 displayAllMembers();  
  
 listViewMembers.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 // Get the selected member from the cursor  
 Cursor cursor = (Cursor) parent.getItemAtPosition(position);  
 int memberId = cursor.getInt(cursor.getColumnIndexOrThrow("\_id"));  
  
 // Start UpdateMemberActivity and pass memberId as extra  
 Intent intent = new Intent(MemberManagementActivity.this, UpdateMemberActivity.class);  
 intent.putExtra("memberId", memberId);  
 startActivity(intent);  
 }  
 });  
  
 addMemberButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(MemberManagementActivity.this, AddMemberActivity.class);  
 startActivity(intent);  
 }  
 });  
 }  
 @Override  
 protected void onResume() {  
 super.onResume();  
 displayAllMembers();  
 }  
 private void displayAllMembers() {  
 Cursor cursor = dbHelper.getAllMembers();  
 if (cursor.getCount() == 0) {  
 Toast.makeText(this, "No members available", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 // Inside displayAllMembers() method in MemberManagementActivity.java  
  
 String[] columns = new String[]{  
 DBHelper.KEY\_MEMBER\_NAME,  
 DBHelper.KEY\_MEMBER\_CONTACT,  
 DBHelper.KEY\_MEMBER\_ADDRESS, // Add additional columns here  
 DBHelper.KEY\_MEMBER\_UNPAID\_DUES,  
 DBHelper.KEY\_MEMBER\_BRANCH\_NAME,  
 DBHelper.KEY\_MEMBER\_BRANCH\_ID,  
 DBHelper.KEY\_MEMBER\_BRANCH\_ADDRESS  
 };  
  
 int[] to = new int[]{  
 R.id.textViewMemberName,  
 R.id.textViewMemberContact,  
 R.id.textViewMemberAddress, // Add corresponding TextViews for additional details  
 R.id.textViewMemberUnpaidDues,  
 R.id.textViewMemberBranchName,  
 R.id.textViewMemberBranchID,  
 R.id.textViewMemberBranchAddress  
 };  
  
  
 SimpleCursorAdapter adapter = new SimpleCursorAdapter(this, R.layout.member\_item, cursor, columns, to, 0);  
 listViewMembers.setAdapter(adapter);  
 }  
}

**UpdateBookActivity.java**

package com.example.common;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class UpdateBookActivity extends AppCompatActivity {  
  
 private EditText titleEditText, authorEditText, isbnEditText;  
 private Button updateButton;  
 private DBHelper dbHelper;  
 private int bookId;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_update\_book);  
  
 dbHelper = new DBHelper(this);  
 titleEditText = findViewById(R.id.titleEditText);  
 authorEditText = findViewById(R.id.authorEditText);  
 isbnEditText = findViewById(R.id.isbnEditText);  
 updateButton = findViewById(R.id.updateButton);  
  
 // Get the bookId from the intent extras  
 Intent intent = getIntent();  
 bookId = intent.getIntExtra("bookId", -1);  
  
 if (bookId == -1) {  
 Toast.makeText(this, "Invalid book", Toast.LENGTH\_SHORT).show();  
 finish(); // Close activity if bookId is not provided  
 }  
  
 // Fetch book details from database and populate EditText fields  
 populateBookDetails();  
  
 updateButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 updateBook();  
 }  
 });  
 }  
  
 private void populateBookDetails() {  
 // Fetch book details from database using bookId  
 Cursor cursor = dbHelper.getBookById(bookId);  
 if (cursor != null && cursor.moveToFirst()) {  
 String title = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_TITLE));  
 String author = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_AUTHOR));  
 String isbn = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_ISBN));  
  
 titleEditText.setText(title);  
 authorEditText.setText(author);  
 isbnEditText.setText(isbn);  
 cursor.close(); // Close cursor when done  
 }  
 }  
  
 private void updateBook() {  
 String title = titleEditText.getText().toString().trim();  
 String author = authorEditText.getText().toString().trim();  
 String isbn = isbnEditText.getText().toString().trim();  
  
 if (title.isEmpty() || author.isEmpty() || isbn.isEmpty()) {  
 Toast.makeText(this, "Please fill all fields", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 int rowsAffected = dbHelper.updateBook(bookId, title, author, isbn);  
 if (rowsAffected > 0) {  
 Toast.makeText(this, "Book updated successfully", Toast.LENGTH\_SHORT).show();  
 finish(); // Close activity after updating book  
 } else {  
 Toast.makeText(this, "Failed to update book", Toast.LENGTH\_SHORT).show();  
 }  
 }  
}

**UpdateMemberActivity.java**

package com.example.common;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class UpdateMemberActivity extends AppCompatActivity {  
  
 private EditText nameEditText, contactEditText, addressEditText, unpaidduesEditText, branchEditText, branchIDEditText, branchAddressEditText;  
 private Button updateButton;  
 private DBHelper dbHelper;  
 private int memberId;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_update\_member);  
  
 dbHelper = new DBHelper(this);  
 nameEditText = findViewById(R.id.nameEditText);  
 contactEditText = findViewById(R.id.contactEditText);  
 addressEditText = findViewById(R.id.addressEditText);  
 unpaidduesEditText = findViewById(R.id.unpaidduesEditText);  
 branchEditText = findViewById(R.id.branchEditText);  
 branchIDEditText = findViewById(R.id.branchIDEditText);  
 branchAddressEditText = findViewById(R.id.branchaddressEditText);  
 updateButton = findViewById(R.id.updateButton);  
  
 // Get the memberId from the intent extras  
 Intent intent = getIntent();  
 memberId = intent.getIntExtra("memberId", -1);  
  
 if (memberId == -1) {  
 Toast.makeText(this, "Invalid member", Toast.LENGTH\_SHORT).show();  
 finish(); // Close activity if memberId is not provided  
 }  
  
 // Fetch member details from database and populate EditText fields  
 populateMemberDetails();  
  
 updateButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 updateMember();  
 }  
 });  
 }  
  
 private void populateMemberDetails() {  
 // Fetch member details from database using memberId  
 Cursor cursor = dbHelper.getMemberById(memberId);  
 if (cursor.moveToFirst()) {  
 String name = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_MEMBER\_NAME));  
 String contact = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_MEMBER\_CONTACT));  
 String address = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_MEMBER\_ADDRESS));  
 double unpaiddues = cursor.getDouble(cursor.getColumnIndexOrThrow(DBHelper.KEY\_MEMBER\_UNPAID\_DUES));  
 String branchName = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_MEMBER\_BRANCH\_NAME));  
 String branchID = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_MEMBER\_BRANCH\_ID));  
 String branchAddress = cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.KEY\_MEMBER\_BRANCH\_ADDRESS));  
  
 nameEditText.setText(name);  
 contactEditText.setText(contact);  
 addressEditText.setText(address);  
 unpaidduesEditText.setText(String.valueOf(unpaiddues));  
 branchEditText.setText(branchName);  
 branchIDEditText.setText(String.valueOf(branchID));  
 branchAddressEditText.setText(branchAddress);  
 }  
 cursor.close(); // Close the cursor after use  
 }  
  
 private void updateMember() {  
 String name = nameEditText.getText().toString().trim();  
 String contact = contactEditText.getText().toString().trim();  
 String address = addressEditText.getText().toString().trim();  
 double unpaiddues = Double.parseDouble(unpaidduesEditText.getText().toString().trim());  
 String branchName = branchEditText.getText().toString().trim();  
 String branchID = branchIDEditText.getText().toString().trim();  
 String branchAddress = branchAddressEditText.getText().toString().trim();  
  
 if (name.isEmpty() || contact.isEmpty() || address.isEmpty() || branchName.isEmpty() || branchAddress.isEmpty()) {  
 Toast.makeText(this, "Please fill all fields", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
  
 int rowsAffected = dbHelper.updateMember(memberId, name, contact, address, unpaiddues, branchName, branchID, branchAddress);  
 if (rowsAffected > 0) {  
 Toast.makeText(this, "Member updated successfully", Toast.LENGTH\_SHORT).show();  
 finish(); // Close activity after updating member  
 } else {  
 Toast.makeText(this, "Failed to update member", Toast.LENGTH\_SHORT).show();  
 }  
 }  
}