

Qiong LIU

Databases related codes

0.1 Exercise 1: SharedPreferences

Create an Activity where the user inputs their information, saves this data to SharedPreferences, and can retrieve it later.

```
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center_horizontal">

        <!-- Input for Name -->
        <EditText
            android:id="@+id/inputName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter your name"
            android:inputType="text"
            android:minHeight="48dp"
            android:padding="12dp"
            android:layout_marginBottom="8dp" />

        ...
    </LinearLayout>
</ScrollView>
```

```
public class MainActivity2 extends AppCompatActivity {

    private static final String PREFS_NAME = "UserProfile";
    private static final String KEY_NAME = "name";
    private static final String KEY_AGE = "age";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        EditText inputName = findViewById(R.id.inputName);
        EditText inputAge = findViewById(R.id.inputAge);
        Button saveButton = findViewById(R.id.saveButton);
        Button loadButton = findViewById(R.id.loadButton);
        Button clearButton = findViewById(R.id.clearButton);
        TextView displayProfile = findViewById(R.id.displayProfile);

        // Save data to SharedPreferences
        saveButton.setOnClickListener(v -> {
            String name = inputName.getText().toString().trim();
            String age = inputAge.getText().toString().trim();

            if (name.isEmpty() || age.isEmpty()) {
                displayProfile.setText("Please fill in both name and age.");
                return;
            }

            SharedPreferences prefs = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
            SharedPreferences.Editor editor = prefs.edit();

            editor.putString(KEY_NAME, name);
            editor.putString(KEY_AGE, age);
            editor.apply();

            displayProfile.setText("Profile saved!");
        });
    }
}
```

```

        inputName.setText("");
        inputAge.setText("");
    });

    // Load data from SharedPreferences
    loadButton.setOnClickListener(v -> {
        SharedPreferences prefs = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
        String name = prefs.getString(KEY_NAME, "No name found");
        String age = prefs.getString(KEY_AGE, "No age found");

        displayProfile.setText(String.format("Name: %s\nAge: %s", name, age));
    });

    // Clear data from SharedPreferences
    clearButton.setOnClickListener(v -> {
        SharedPreferences prefs = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
        SharedPreferences.Editor editor = prefs.edit();

        editor.clear();
        editor.apply();

        displayProfile.setText("Profile cleared!");
    });
}
}
}

```

0.2 Exercise 2: SQLite

Create an fragment can add/ delete course items.

- Let students enter course info: name, teacher, time
- Save data into SQLite database
- Show all saved courses in a list (ListView)
- Long-click a course to delete it

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Input for Course Name -->
    <EditText
        android:id="@+id/edit_course_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Course Name" />

    <!-- Input for Teacher Name -->
    <EditText
        android:id="@+id/edit_teacher"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Teacher" />

    <!-- Input for Time Slot -->
    <EditText
        android:id="@+id/edit_time_slot"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Time Slot (e.g. Mon 10:00)" />

    <!-- Add Course Button -->

```

```

<Button
    android:id="@+id/button_add_course"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add Course"
    android:layout_marginTop="12dp" />

<!-- ListView to show saved courses -->
<ListView
    android:id="@+id/list_courses"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:dividerHeight="8dp" />
</LinearLayout>

```

Step 1: Creat a CourseDatabaseHelper

```

package com.example.cm2;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class CourseDatabaseHelper extends SQLiteOpenHelper {

    // version type
    private static final String DATABASE_NAME = "course.db"; // course.db is the name of the file where data will
        be stored
    private static final int DATABASE_VERSION = 1; // DATABASE_VERSION is used to track upgrades. keep it as 1

    // table name and its column names
    public static final String TABLE_NAME = "courses";
    public static final String COLUMN_ID = "id";
    public static final String COLUMN_COURSE_NAME = "course_name";
    public static final String COLUMN_TEACHER = "teacher";
    public static final String COLUMN_TIME_SLOT = "time_slot";

    // creat SQL
    private static final String TABLE_CREATE =
        "CREATE TABLE " + TABLE_NAME + " (" +
            COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_COURSE_NAME + " TEXT, " +
            COLUMN_TEACHER + " TEXT, " +
            COLUMN_TIME_SLOT + " TEXT" +
        ");";

    // Constructor
    public CourseDatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    // This method runs automatically the first time the database is created.
    //It executes the SQL string to create the courses table.
    @Override
    public void onCreate(SQLiteDatabase db) {
        // create
        db.execSQL(TABLE_CREATE);
    }

    // Handles schema updates (here, by dropping and recreating)
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        //
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}

```

Step 2: Fragment3

```

package com.example.cm2;

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;

```

```

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import java.util.ArrayList;

public class Fragment3 extends Fragment {

    // SQLiteOpenHelper subclass used to manage the database
    private CourseDatabaseHelper dbHelper;

    // UI elements for data input
    private EditText editCourseName;
    private EditText editTeacher;
    private EditText editTimeSlot;
    private Button buttonAdd;

    // UI elements for displaying course list
    private ListView listViewCourses; // ListView to show course data
    private ArrayAdapter<String> courseAdapter; // Adapter to bridge data and ListView
    private ArrayList<String> courseList; // List holding course strings

    public Fragment3() {
        // Required empty public constructor
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                             @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {

        View view = inflater.inflate(R.layout.fragment_3, container, false);

        // Create the database helper instance with current context
        dbHelper = new CourseDatabaseHelper(requireContext());

        // Bind views
        editCourseName = view.findViewById(R.id.edit_course_name);
        editTeacher = view.findViewById(R.id.edit_teacher);
        editTimeSlot = view.findViewById(R.id.edit_time_slot);
        buttonAdd = view.findViewById(R.id.button_add_course);

        // Set onClickListener for the Add button
        buttonAdd.setOnClickListener(v -> insertCourse());

        // Set up ListView and adapter to display courses
        listViewCourses = view.findViewById(R.id.list_courses);
        courseList = new ArrayList<>();
        courseAdapter = new ArrayAdapter<>(requireContext(), android.R.layout.simple_list_item_1, courseList);
        listViewCourses.setAdapter(courseAdapter);

        // Load existing courses from the database and show in list
        loadCourses();

        // Set a long-click listener to delete an item when long pressed
        listViewCourses.setOnItemClickListener((parent, view1, position, id) -> {
            // Get the selected item string
            String selectedItem = courseList.get(position);

            // Extract course name (you could also use ID if stored)
            String courseName = selectedItem.split(" \\(")[0]; // e.g. "Math (Prof. Smith)"

            // Confirm delete
            new android.app.AlertDialog.Builder(requireContext())
                .setTitle("Delete course")
                .setMessage("Are you sure you want to delete \"" + courseName + "\"?")
                .setPositiveButton("Yes", (dialog, which) -> {
                    deleteCourseByName(courseName); // Delete from DB
                })
                .setNegativeButton("No", null)
                .show();
        });
    }

    private void insertCourse() {
        // Insert logic
    }

    private void loadCourses() {
        // Load logic
    }

    private void deleteCourseByName(String courseName) {
        // Delete logic
    }
}

```

```

        loadCourses(); // Refresh the list
    })
    .setNegativeButton("No", null)
    .show();

    return true; // Long-click was handled
});

return view; // Return the inflated layout
}

// Method to insert course into the database
private void insertCourse() {
    // Get user input from EditTexts
    String courseName = editCourseName.getText().toString().trim();
    String teacher = editTeacher.getText().toString().trim();
    String timeSlot = editTimeSlot.getText().toString().trim();

    // Input validation
    if (courseName.isEmpty() || teacher.isEmpty() || timeSlot.isEmpty()) {
        Toast.makeText(getContext(), "Please fill all fields", Toast.LENGTH_SHORT).show();
        return;
    }

    // Execute SQL insert command with placeholders and arguments
    SQLiteDatabase db = dbHelper.getWritableDatabase();
    db.execSQL("INSERT INTO " + CourseDatabaseHelper.TABLE_NAME +
        " (" + CourseDatabaseHelper.COLUMN_COURSE_NAME + ", " +
        CourseDatabaseHelper.COLUMN_TEACHER + ", " +
        CourseDatabaseHelper.COLUMN_TIME_SLOT + ") VALUES (?, ?, ?)",
        new Object[]{courseName, teacher, timeSlot});

    Toast.makeText(getContext(), "Course added successfully", Toast.LENGTH_SHORT).show();

    // Clear input fields
    editCourseName.setText("");
    editTeacher.setText("");
    editTimeSlot.setText("");
    // Reload list
    loadCourses();
}

// Load courses from the database and display in ListView
private void loadCourses() {
    courseList.clear(); // Clear old data

    // Read access to the database
    SQLiteDatabase db = dbHelper.getReadableDatabase();
    // Query all data from the table
    Cursor cursor = db.rawQuery("SELECT * FROM " + CourseDatabaseHelper.TABLE_NAME, null);

    // Loop through all rows in the result
    if (cursor.moveToFirst()) {
        do {
            String courseName = cursor.getString(cursor.getColumnIndexOrThrow(CourseDatabaseHelper.
                COLUMN_COURSE_NAME));
            String teacher = cursor.getString(cursor.getColumnIndexOrThrow(CourseDatabaseHelper.COLUMN_TEACHER
                ));
            String timeSlot = cursor.getString(cursor.getColumnIndexOrThrow(CourseDatabaseHelper.
                COLUMN_TIME_SLOT));

            // Format course info string for display
            String courseInfo = courseName + " (" + teacher + ") - " + timeSlot;
            courseList.add(courseInfo);

        } while (cursor.moveToNext());
    }

    cursor.close();
    courseAdapter.notifyDataSetChanged(); // Refresh ListView
}

// Delete a course from the database using its name
private void deleteCourseByName(String name) {
    SQLiteDatabase db = dbHelper.getWritableDatabase();
    db.delete(CourseDatabaseHelper.TABLE_NAME,
        CourseDatabaseHelper.COLUMN_COURSE_NAME + " = ?",
        new String[]{name});
    Toast.makeText(getContext(), "Course \" " + name + "\" deleted", Toast.LENGTH_SHORT).show();
}
}

```