You will start with the android program you developed to display RSS server data and modify the cardview to display like and display buttons and show the total number of likes and dislikes for each news.

See Historical Records Documenting the Pope's Creole Roots in New Orleans

⊕ 1   ✕ 1

Earthquake Rocks Parts of Tennessee and Georgia

⊕ 1   ✕ 0

Pope Leo the White Sox Fan: Something to Celebrate for Chicago's South Side

⊕ 0   ✕ 4

Pope's Childhood in a Changing Chicago Tells a Story of Catholic America

⊕ 1   ✕ 0

Attacking Trump's Tariffs, Democrats Focus on Small Business Struggles

⊕ 1   ✕ 3

Create a table in XAMPP

Write a flask server to write to pos column, neg column
And to read from it to display the values of pos and neg in android app

The following server code is for my design, your design may be different so use it as an example

```
from flask import Flask, request, jsonify
import pymysql

app = Flask(__name__)

# ✓ MySQL connection setup (no SSL)
def get_db_connection():
    return pymysql.connect(
        host='localhost',
        user='root',
        password='',
        db='raif',
        charset='utf8mb4',
        cursorclass=pymysql.cursors.DictCursor
    )

@app.route('/register_news/<title>', methods=['POST'])
def register_news(title):
    conn = get_db_connection()
    with conn.cursor() as cur:
        cur.execute("SELECT COUNT(*) AS count FROM newastat WHERE news = %s", (title,))
        exists = cur.fetchone()['count']
        if exists == 0:
            cur.execute("INSERT INTO newastat (news, pos, neg) VALUES (%s, %s, %s)", (title, 0, 0))
            conn.commit()
    conn.close()
    return jsonify({'message': 'News registered or already exists'})

# ✓ Get statistics for a news item by title
@app.route('/get_stats/<title>', methods=['GET'])
def get_stats(title):
    conn = get_db_connection()
    with conn.cursor() as cur:
        cur.execute("SELECT news, pos, neg FROM newastat WHERE news = %s", (title,))
```

```python
        row = cur.fetchone()
    conn.close()
    if row:
        return jsonify(row)
    else:
        return jsonify({'error': 'News item not found'}), 404

@app.route('/thumb_up/<title>', methods=['POST'])
def thumb_up(title):
    conn = get_db_connection()
    with conn.cursor() as cur:
        # Ensure news is in the table
        cur.execute("SELECT COUNT(*) AS count FROM newastat WHERE news = %s", (title,))
        exists = cur.fetchone()['count']
        if exists == 0:
            cur.execute("INSERT INTO newastat (news, pos, neg) VALUES (%s, %s, %s)", (title, 0, 0))

        # Increment thumbs up
        cur.execute("UPDATE newastat SET pos = pos + 1 WHERE news = %s", (title,))
        conn.commit()

        # Fetch updated stats
        cur.execute("SELECT news, pos, neg FROM newastat WHERE news = %s", (title,))
        row = cur.fetchone()
    conn.close()

    return jsonify(row)

@app.route('/thumb_down/<title>', methods=['POST'])
def thumb_down(title):
    conn = get_db_connection()
    with conn.cursor() as cur:
        cur.execute("SELECT COUNT(*) AS count FROM newastat WHERE news = %s", (title,))
        exists = cur.fetchone()['count']
        if exists == 0:
            cur.execute("INSERT INTO newastat (news, pos, neg) VALUES (%s, %s, %s)", (title, 0, 0))

        cur.execute("UPDATE newastat SET neg = neg + 1 WHERE news = %s", (title,))
        conn.commit()

        cur.execute("SELECT news, pos, neg FROM newastat WHERE news = %s", (title,))
        row = cur.fetchone()
    conn.close()

    return jsonify(row)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

I used curl command for the ease of use.
Registering news:

```
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp % conda activate rssfeed && python3.12 database.py
Table created successfully
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp % curl -X POST "http://localhost:5005/register_news/Test%20News%20Article
"
{"message":"News registered successfully"}
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp %
```

Getting the news information;:

```
"
{"message":"News registered successfully"}
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp % curl "http://localhost:5005/get_stats/Test%20News%20Article"
{"neg":0,"pos":0}
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp %
```

Testing the thumbs up endpoint:

```
{"neg":0,"pos":0}
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp % curl -X POST "http://localhost:5005/thumb_up/Test%20News%20Article"
{"neg":0,"pos":1}
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp %
```

Testing the thumbs down endpoint:
```
{"neg":0,"pos":1}
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp % curl -X POST "http://localhost:5005/thumb_down/Test%20News%20Article"
{"neg":1,"pos":1}
(rssfeed) eren@eren-MacBook-Pro RSSReaderApp %
```
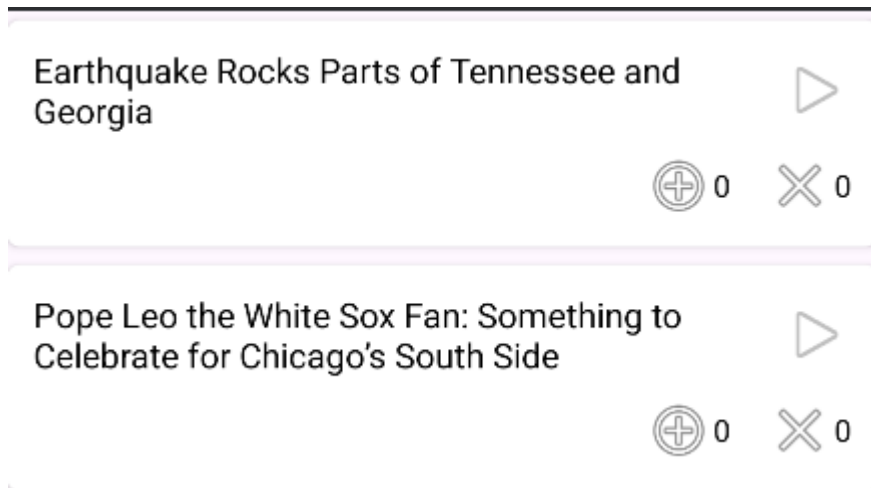
POST /register_news/<title> - Registers a new article
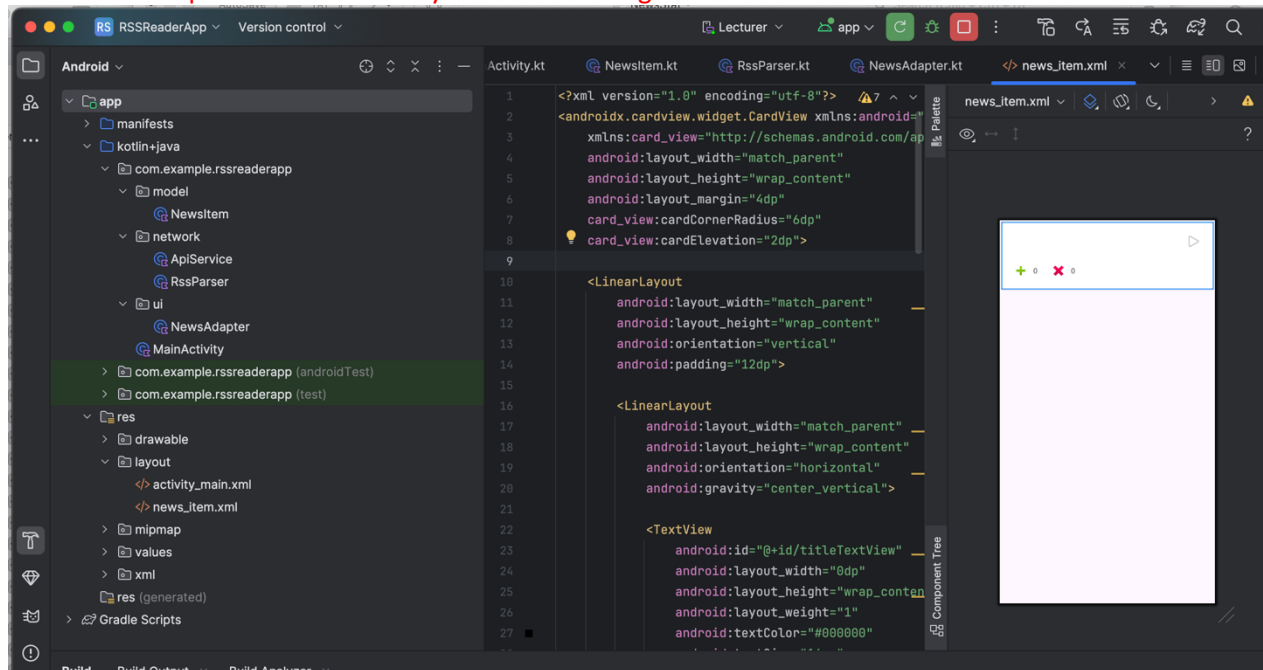GET /get_stats/<title> - Gets current like/dislike counts
POST /thumb_up/<title> - Adds a like
POST /thumb_down/<title> - Adds a dislike

In your Android app, modify your layout file defined for news item (card view) to include thumbs up and thumbs down UI elements You can use any image you like

Earthquake Rocks Parts of Tennessee and Georgia

▷

⊕ 0   ✕ 0

Pope Leo the White Sox Fan: Something to Celebrate for Chicago's South Side

▷

⊕ 0   ✕ 0

Paste here the printscreen of the layout view showing the two additional buttons

News_itemxml code:

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="4dp"
    card_view:cardCornerRadius="6dp"
    card_view:cardElevation="2dp">
```

```xml
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="12dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical">

        <TextView
            android:id="@+id/titleTextView"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:textColor="#000000"
            android:textSize="16sp"
            android:maxLines="2"
            android:ellipsize="end" />

        <ImageButton
            android:id="@+id/playPauseButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@android:color/transparent"
            android:src="@android:drawable/ic_media_play"
            android:contentDescription="Play/Pause"
            android:padding="8dp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="8dp">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical"
            android:layout_marginEnd="16dp">

            <ImageButton
                android:id="@+id/likeButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@android:color/transparent"
                android:src="@android:drawable/ic_input_add"
                android:contentDescription="Like"
                android:padding="8dp" />

            <TextView
                android:id="@+id/likesCount"
                android:layout_width="wrap_content"
```
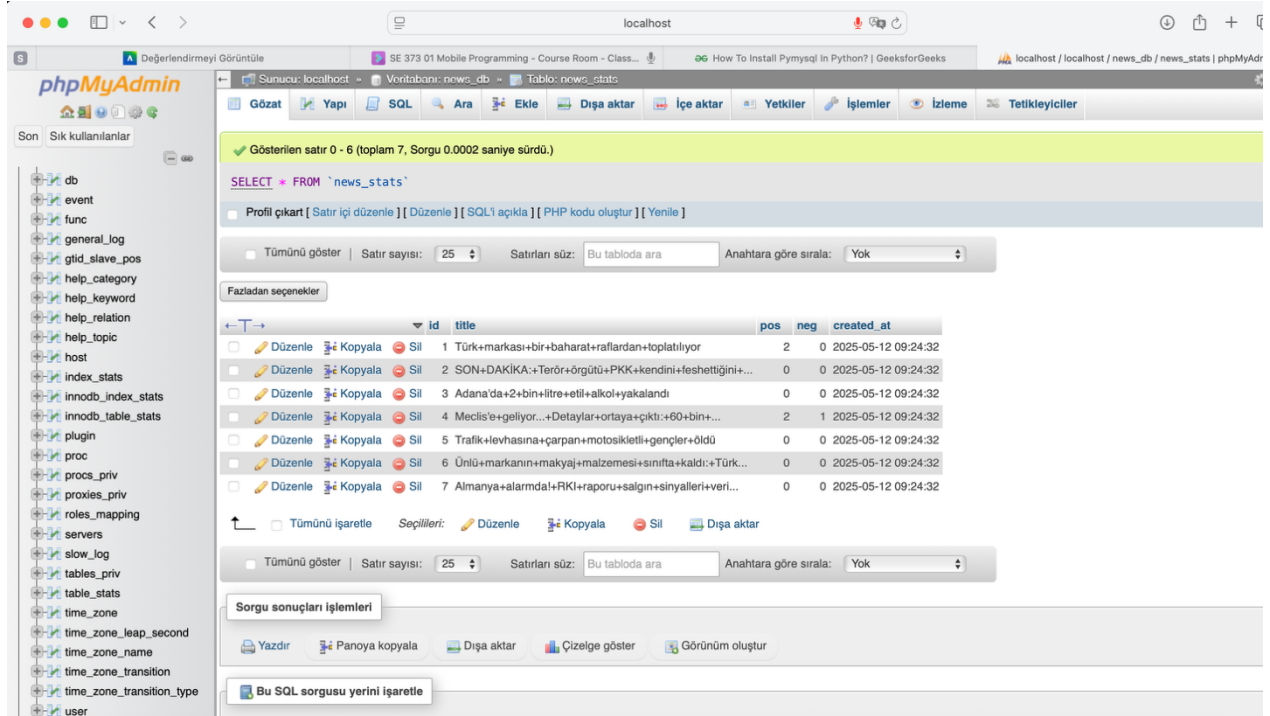
```xml
                android:layout_height="wrap_content"
                android:textSize="14sp"
                android:text="0" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center_vertical">

            <ImageButton
                android:id="@+id/dislikeButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@android:color/transparent"
                android:src="@android:drawable/ic_delete"
                android:contentDescription="Dislike"
                android:padding="8dp" />

            <TextView
                android:id="@+id/dislikesCount"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="14sp"
                android:text="0" />
        </LinearLayout>
    </LinearLayout>
    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Include network_security_config.cml under xml folder and define it in android manifest file. You should already have internet permission in android manifest.

You need to modify news adapter kotlin class and create a new class to access the database.
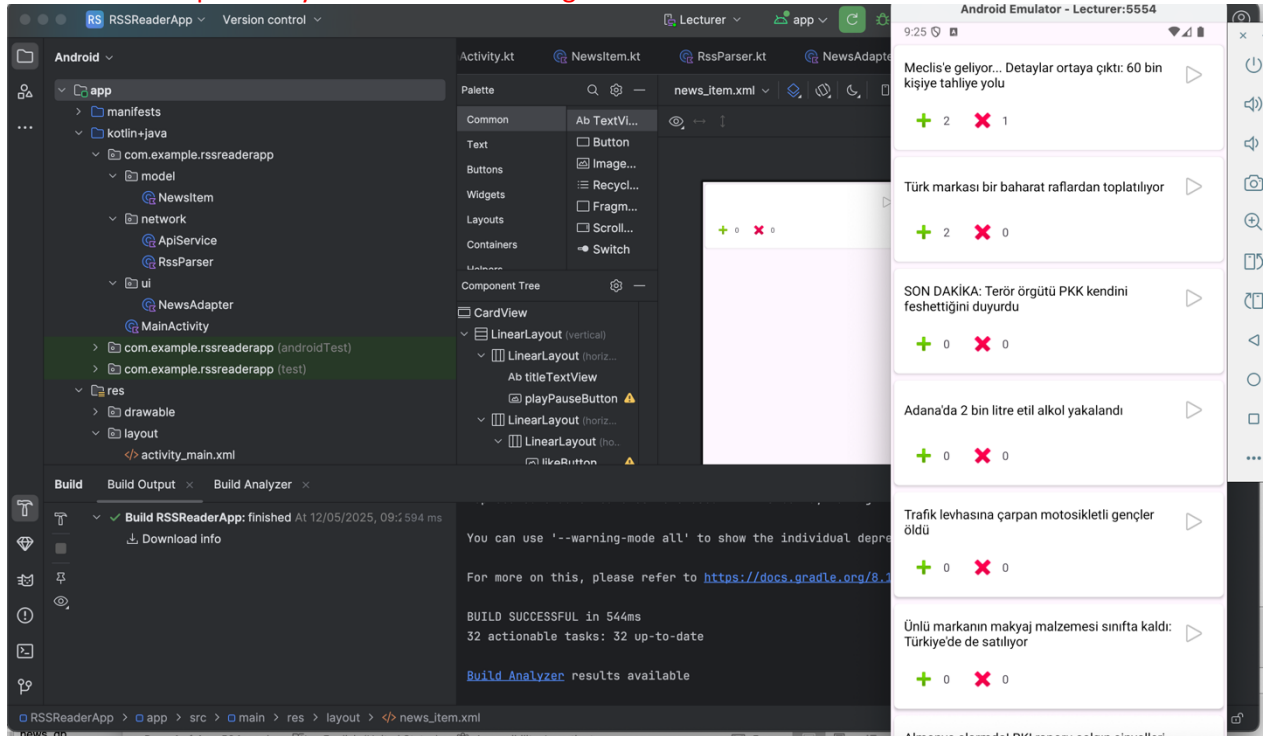
Take a screencapture of your table after marking a few ups and downs for different news

Take a screencapture of your emulator showing the same numbers for news