JJCapitalManager Documentation

General Documentation

How to Use the App

1. Launch the JJCapitalManager app:
    - I sync the application then run it to ensure everything it working properly.
2. Use the following buttons for core functionality:
    - **Clear Button**: Deletes all records from the database.
    - **Populate Button**: Populates the database with sample investment data.
    - **Add Button**: Opens a dialog to add a new investment record.
    - **Print Button**: Logs all investment details to Logcat.
3. Tap on a list item in the RecyclerView to view details or modify an investment.
4. Edit or delete investments directly from the detail dialog.

Classes

- **DBHelper**: Manages the SQLite database, including table creation, data manipulation, and query execution.
- **MainActivity**: Main activity that handles user interactions and RecyclerView updates.
- **Investment**: Data class representing an investment object.
- **MyRecyclerAdapter**: Adapter for displaying investment data in the RecyclerView.

Methods and Properties

- **DBHelper**:
    - onCreate: Creates the investments and investment_details tables.
    - populateDatabase: Inserts 12 sample investment records into the database.
    - clearDatabase: Deletes all records from the database.
    - addInvestment: Adds a new investment record to the database.
    - editInvestment: Updates an existing investment record.
    - deleteInvestment: Deletes an investment record.
    - getAllInvestmentsWithDetails: Retrieves all investment data with details for display.
    - getInvestmentById: Retrieves a specific investment by its ID.
- **MainActivity**:
    - updateRecyclerView: Updates the RecyclerView with current database data.
    - showAddEditDialog: Opens a dialog for adding or editing investments.

    - showDetailView: Displays detailed information for a specific investment.
- **MyRecyclerAdapter**:
    - onBindViewHolder: Binds investment data to RecyclerView items.

Problems Encountered and Solutions

1. Database Initialization

- **Problem**: The app would crash or fail to initialize the database when first launched due to incorrect table creation or schema mismatch.
- **Solution**: Ensured that the `onCreate` method in `DBHelper` correctly creates the `investments` and `investment_details` tables. Added reinstall testing (uninstalling and reinstalling the app) to verify database initialization works as expected.

2. RecyclerView Updates

- **Problem**: The RecyclerView was not updating dynamically after database operations, such as adding, editing, or deleting records.
- **Solution**: Used `lifecycleScope` with coroutines to perform database operations in the `IO` thread and update the RecyclerView data on the main thread. This ensured a responsive UI without blocking the main thread.

Requirements Documentation

SQLite Database and RecyclerView

- **Database Structure**:
    - **investments**
        - id (INTEGER, PRIMARY KEY)
        - investment_type (TEXT, NOT NULL)
        - amount (REAL, NOT NULL)
        - date (TEXT, NOT NULL)
    - **investment_details**
        - id (INTEGER, PRIMARY KEY)
        - investment_id (INTEGER, FOREIGN KEY REFERENCES investments(id))
        - growth_rate (REAL, NOT NULL)
        - duration_years (INTEGER, NOT NULL)
- **RecyclerView**:
    - Displays combined data from investments and investment_details using a JOIN query.

Add, Delete, and Edit List Items

- **Adding Items**:
    - Tap the **Add Button** to open the dialog.
    - Enter details and press "Add".
- **Editing Items**:
    - Tap an item in the RecyclerView.
    - Modify the details in the dialog and press "Update".
- **Deleting Items**:
    - Tap an item in the RecyclerView.
    - Select "Delete" in the dialog.

Clear and Populate Database

- **Clearing Database**:
    - Press the **Clear Button**.
    - Removes all records from both tables.
- **Populating Database**:
    - Press the **Populate Button**.
    - Adds 12 sample investment records into the database without clearing existing data.

Non-Trivial SQL Queries

Queries:

1. **Join Query**:

```
SELECT investments.id, investments.investment_type,
investments.amount, investments.date,
       investment_details.growth_rate,
investment_details.duration_years
FROM investments
JOIN investment_details ON investments.id =
investment_details.investment_id;
```

    - **Purpose:** Displays all investment details in the RecyclerView.

2. **Filtered Query:**

```
SELECT investments.id, investments.investment_type,
investments.amount, investments.date,
```

```
       investment_details.growth_rate,
investment_details.duration_years
FROM investments
JOIN investment_details ON investments.id =
investment_details.investment_id
WHERE investments.id = ?;
```

- **Purpose:** Retrieves details for a specific investment using the ID.

How to View Results:

- Join Query: Results displayed in the RecyclerView.
- Filtered Query: Results shown in the detail dialog.