Project 4 Task 2 – Distributed Application and Dashboard
By Hojoon Lee (AndrewID: hojoonle)

Description: My mobile application allows users to search for a cryptocurrency by name or symbol, and view real-time information such as price, market cap, daily change percentage, and coin icon. It fetches data using the Coinranking API and displays the top 50 coins or details for a selected coin. All user search logs are stored in MongoDB Atlas for operational analytics, including top searched coins, error logs, total and daily successful searches and slowest responses.

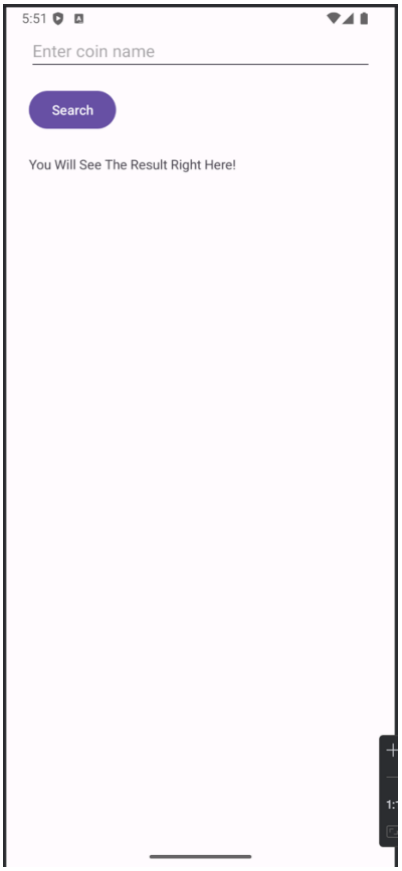
1. **Implement a native Android application**

The name of my native Android application project in Android Studio is:
Project4Task2_AndroidApp

a. Has at least three different kinds of Views in your Layout (TextView, EditText, ImageView, or anything that extends android.view.View)

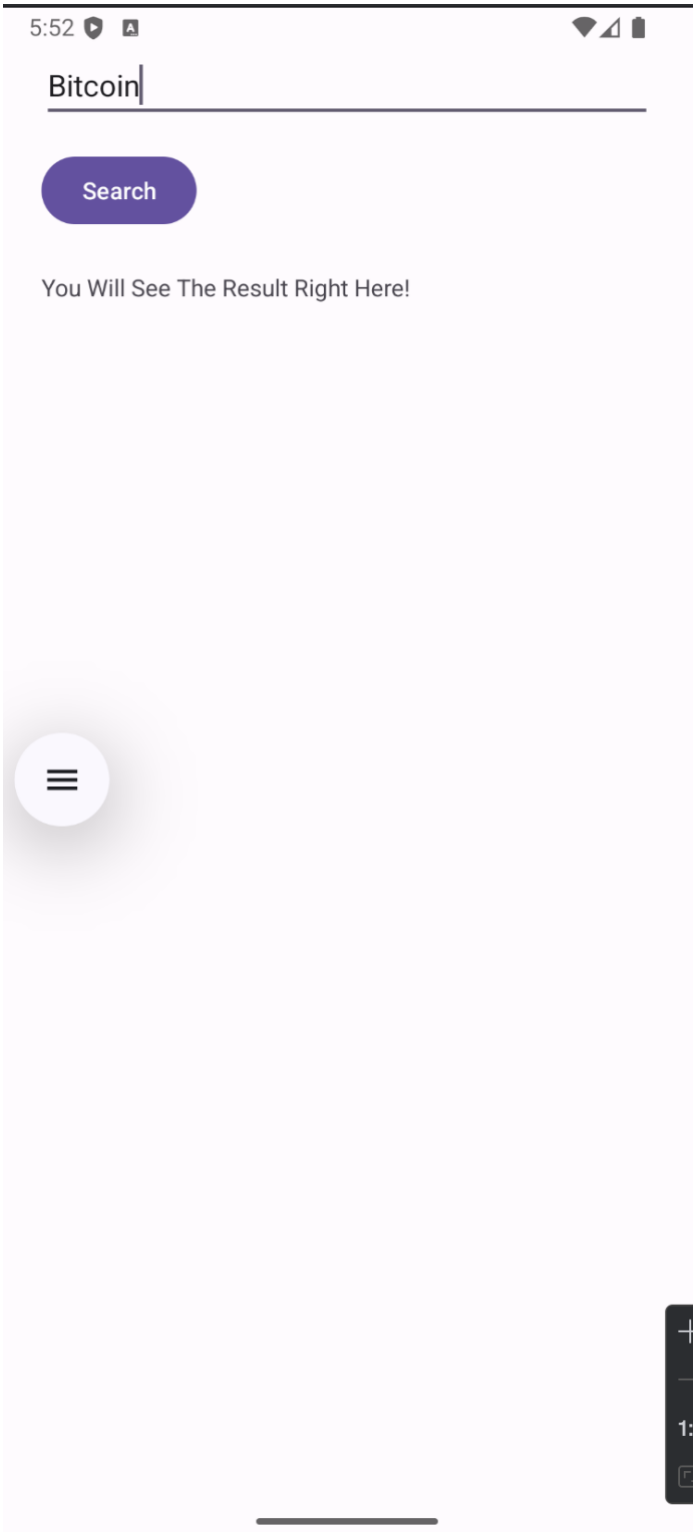
   My layout includes TextView, EditText, and Button. Please refer to activity_main.xml for complete UI definitions.

Here is a screenshot of the layout before the coin information has been fetched:



b. Requires input from the user The user must enter a coin name or symbol in the EditText field to search.

Here is a screenshot of the user searching for information of Bitcoin



c. Makes an HTTP request (using an appropriate HTTP method) My app makes a POST request to: https://fantastic-umbrella-gv9p7g96vpqcwr64-8080.app.github.dev/ Thepayload includes the coin query and device model in JSON format.

   Example POST body: { "coin": "bitcoin", "deviceModel": "Google Pixel 5" }

d. Receives and parses a JSON formatted reply from your web service

The response includes coin name, symbol, price, change, market cap, icon URL, and status. Parsing is done using Gson.

Example JSON response:

{"name": "Bitcoin",

"symbol": "BTC",

"price": 76435.77,

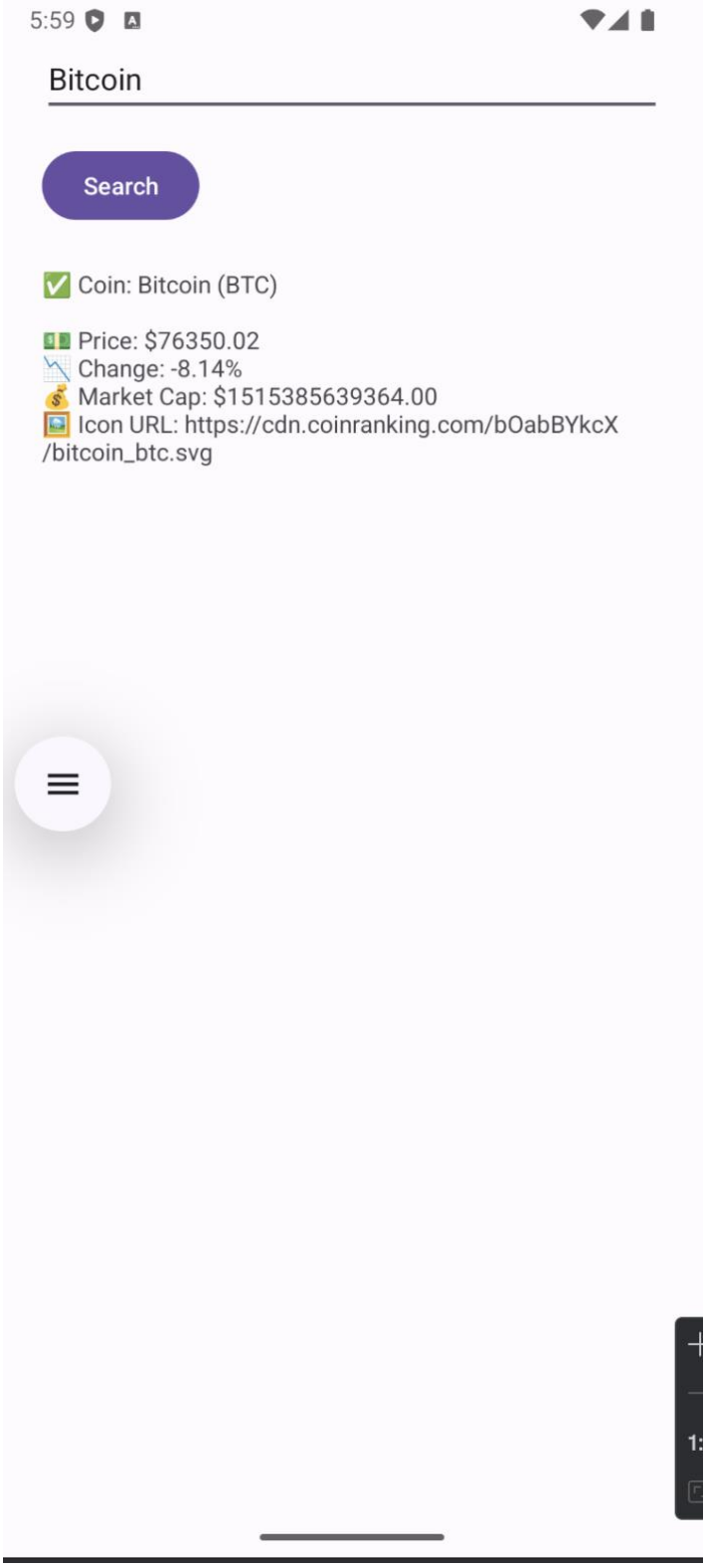
"change": -8.04,

"marketCap": 1517087584178.00,

"iconUrl": "https://cdn.coinranking.com/B1nb2Q5dY/bitcoin.svg"}

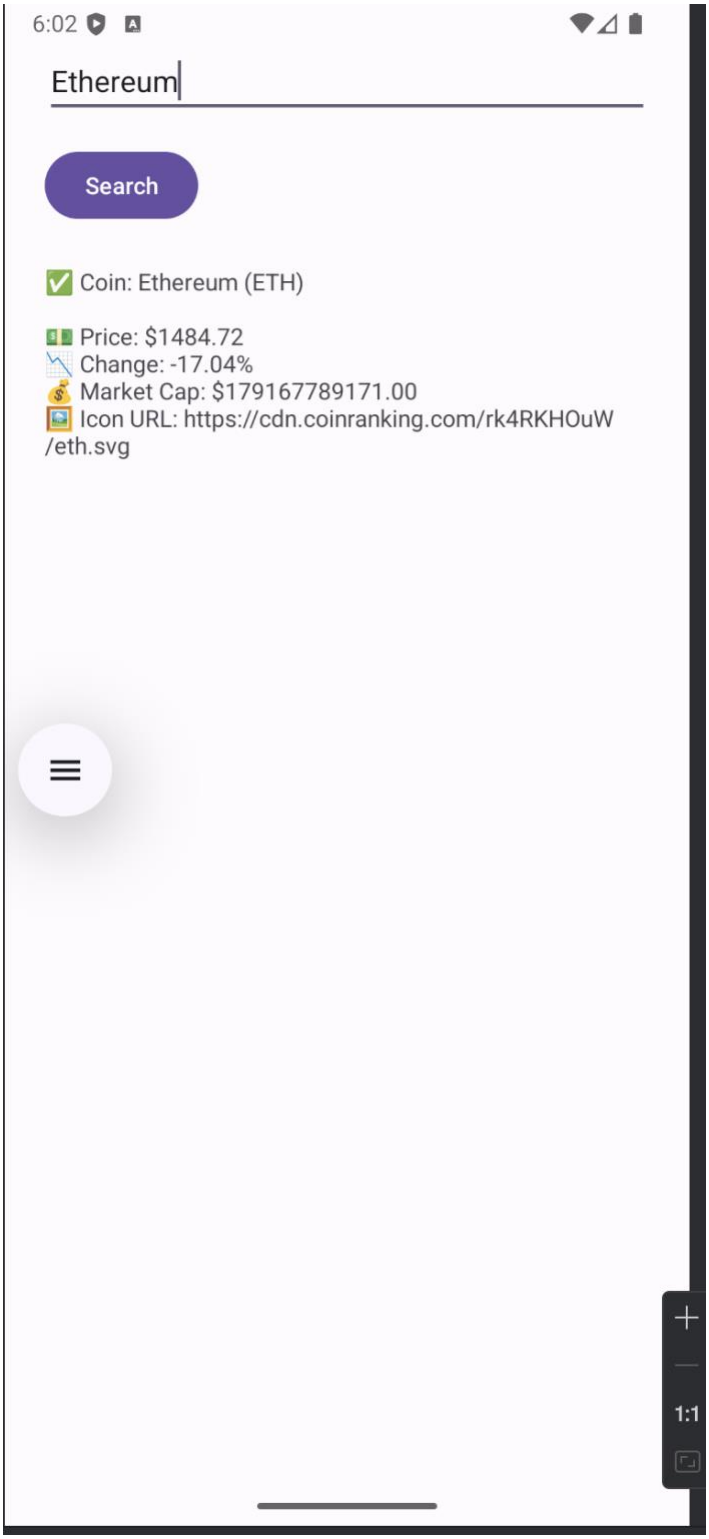
e. Displays new information to the user

Here is the screenshot after the information of Bitcoin has been returned.



f. Is repeatable

The user can type in another search term and hit Search button. Here is an example of having typed in “Ethereum”.



2. **Implement a web service**

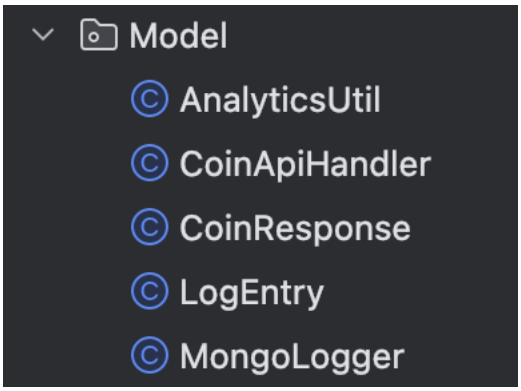
The URL of my web service deployed to Dashboard is: https://fantastic-umbrella-gv9p7g96vpqcwr64-8080.app.github.dev/

The project directory name is Project4Task2_WebService

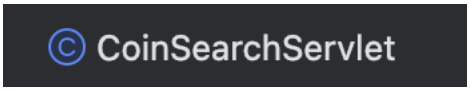
a. Using an HttpServlet to implement a simple API

In my web app project:

Model: AnalyticsUtil.java, CoinApiHandler.java, CoinResponse.java, LogEntry.java, MongoLogger.java



Controller: CoinSearchServlet.java



View: Index.jsp

b. Receives an HTTP request form the native Android application

The CoinSearchServlet receives POST requests from the Android app via the "/coin" endpoint. It reads the JSON request body containing the coin query and device model, parses it into Java objects using Gson, and then forwards the query to the Coinranking API for processing. The response is used to construct a result object and is logged into MongoDB Atlas.

c. Executes business logic appropriate to your application Upon receiving the coin query, the servlet calls the Coinranking API to fetch data and logs each search entry into MongoDB.

API Usage:

```
String apiUrl = "https://api.coinranking.com/v2/coins?limit=50";
HttpRequest request = HttpRequest.newBuilder()
```

Steps:

1. Parse JSON input (coin name + device model)
2. Query Coinranking API
3. Construct CoinResponse object
4. Log data to MongoDB
5. Send JSON reply to client

d. Replies to the Android application with a JSON formatted response The servlet constructs a JSON response using Gson and returns it to the Android app. The response includes fields such as coin name, symbol, price, change, market cap, and icon URL, structured in JSON format. The following snippet shows part of the code used in CoinSearchServlet.java:

Example from (CoinSearchServlet):
response.setContentType("application/json"); PrintWriter out = response.getWriter(); out.print(gson.toJson(coinResponse)); out.flush();

3. **Handle error conditions - Does not need to be documented.**

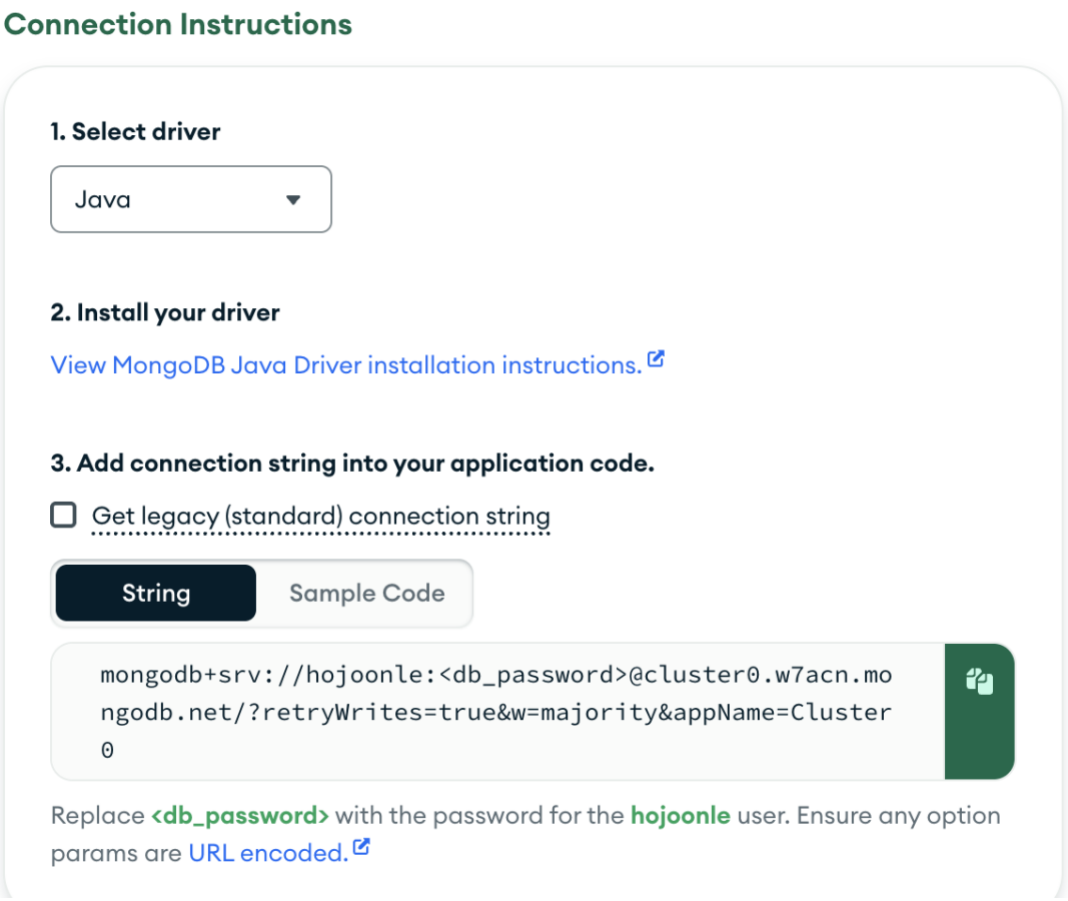
4. **Log useful information - Itemize what information you log and why you chose it. Each request logs the following fields in MongoDB Atlas:**

- timestamp: to track when the request was made.
- clientIP: to identify the request origin.
- coinQuery: to understand which coins are most frequently searched.
- deviceModel: to analyze access trends by device.
- status: to determine the success or failure of the search.
- responseTime: to monitor system performance and latency.
- searchSuccess: to track effectiveness of search matching logic.

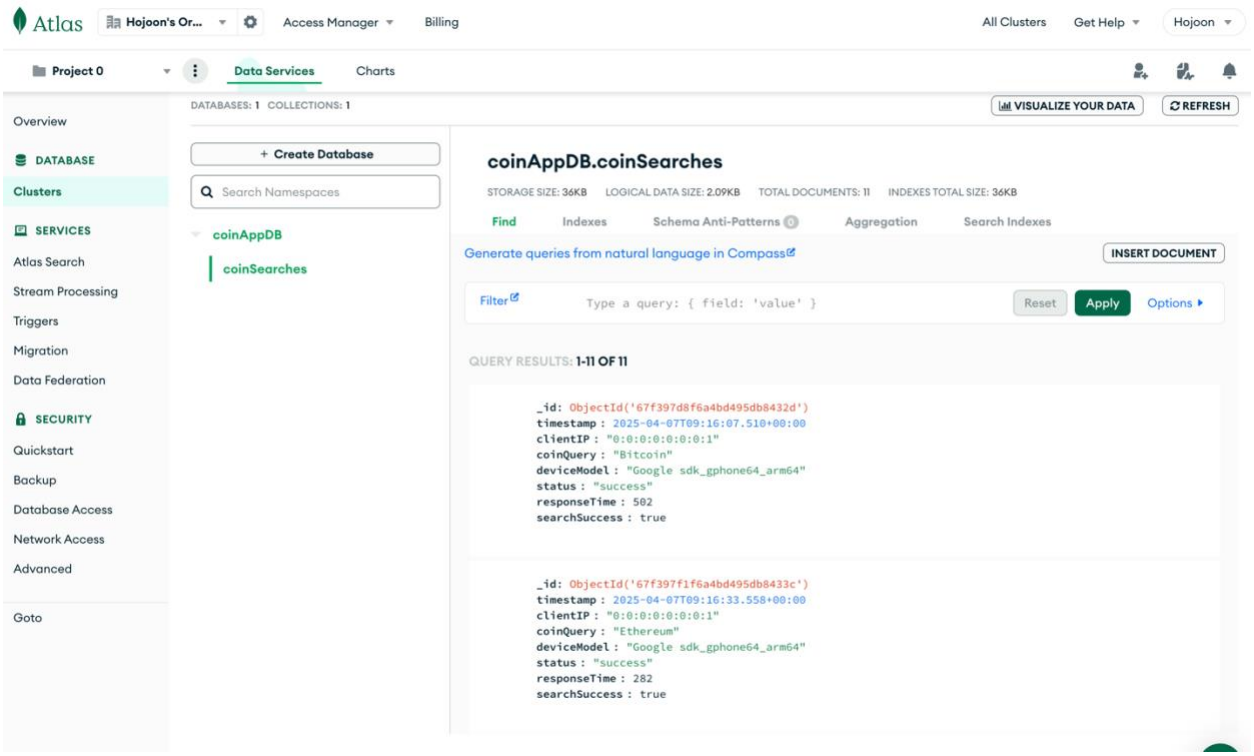
5. **Store the log information in a database.**

Log entries are stored in the MongoDB Atlas collection "coinSearches" in the "coinAppDB" database. Connection handled via MongoClient.

MongoDB Connection



My Mongo DB



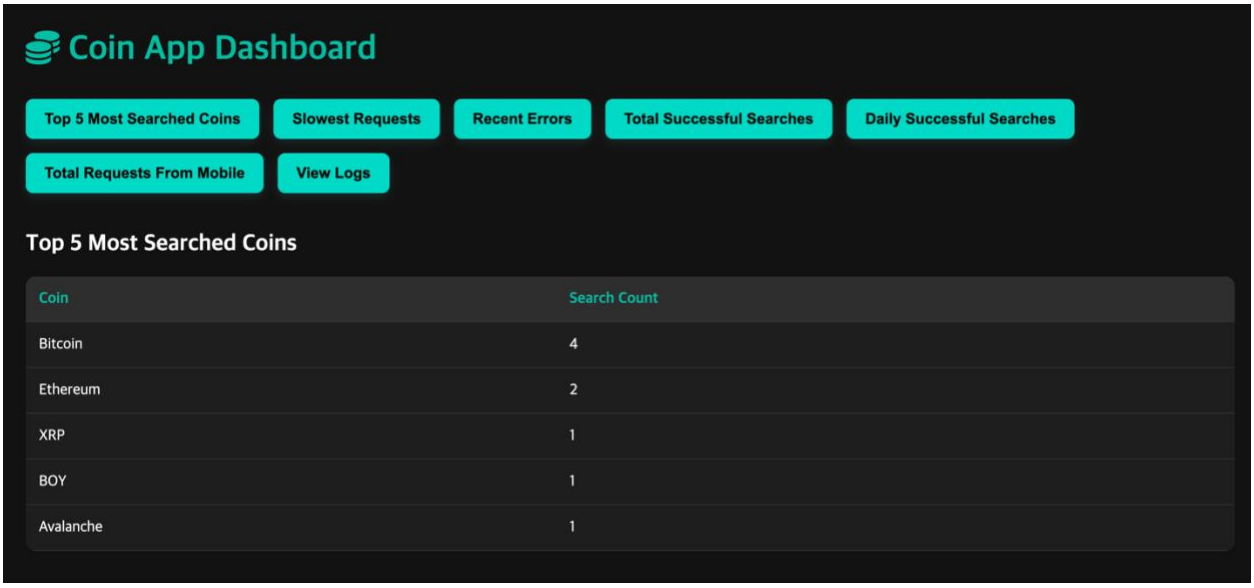
Atlas Connection String:

```
private static final String MONGO_URI = "mongodb+srv://hojoonle:Jacob842455@cluster0.w7acn.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0";
private static final String DB_NAME = "coinAppDB";  2 usages
private static final String COLLECTION_NAME = "coinSearches";  2 usages
```

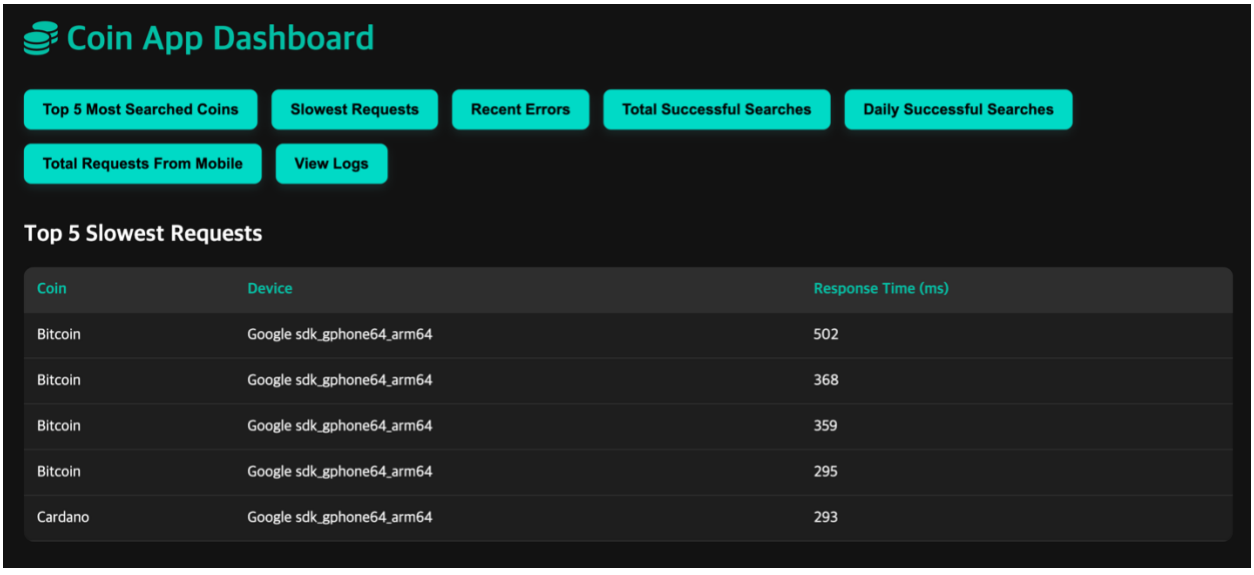
6. **Display operations analytics and full logs on a web-based dashboard - Provide a screenshot.**

Operations Analytcis

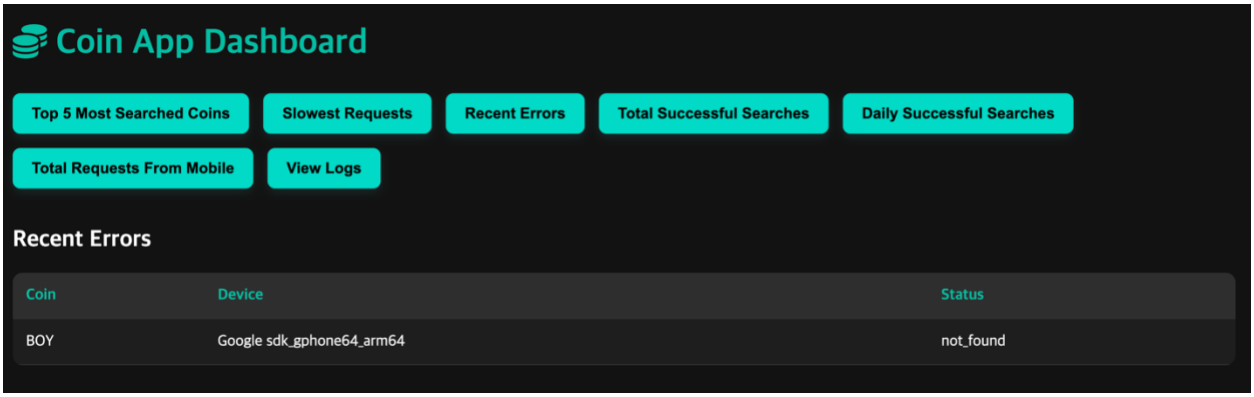
Top 5 Most Searched Coins



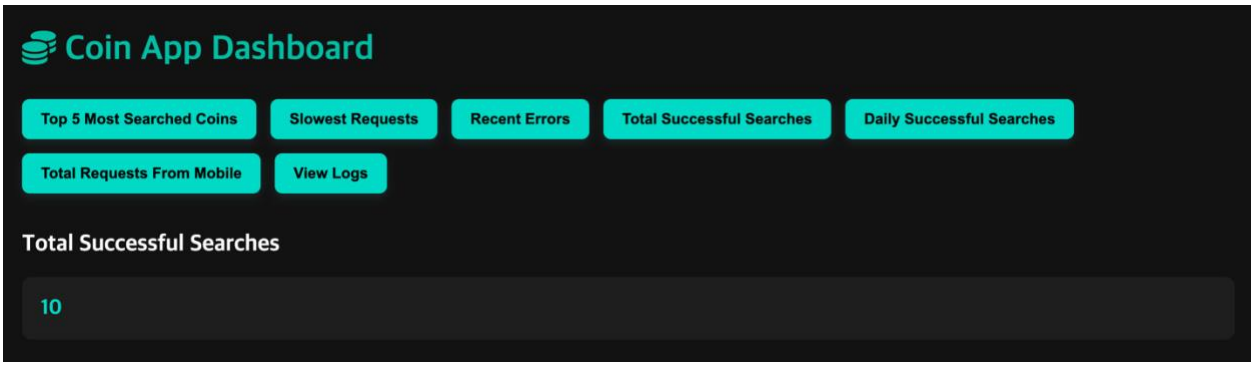
Slowest Requests



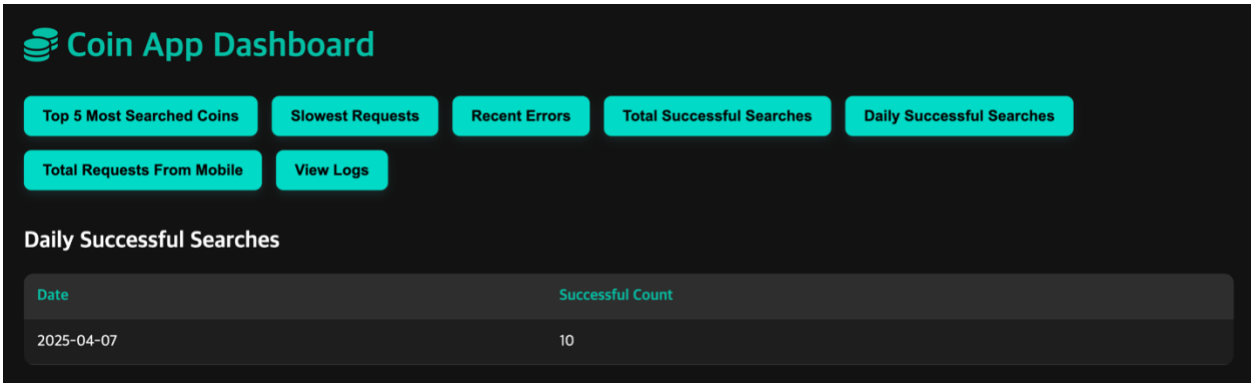
Recent Errors



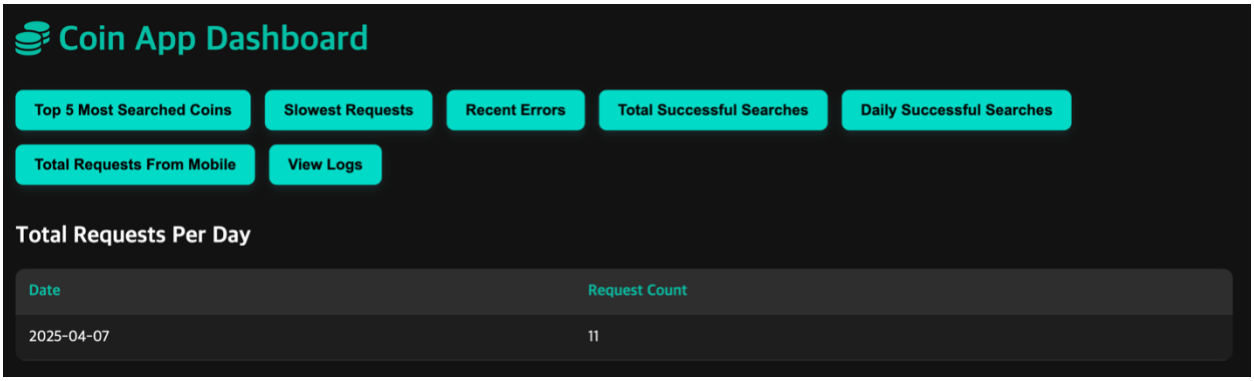
Total Successful Searches



Daily Successful Searches



Total Requests From Mobile



View Logs