SOCIAL MEDIA INFORMATION RECONNAISSANCE PROGRAM

PROJECT I

REPORT

MEMBER LIST

Nguyen Thanh An	20225541	An.NT225541@sis.hust.edu.vn
Pham Doan Phuc Lam	20225546	Lam.PDP225546@sis.hust.edu.vn

Supervisor: Tong Van Van, PhD

Hà Nội — 2024

CONTENTS

LIST OF FIGURES

ABSTRACT

The "Social Media Information Reconnaissance Program and Synchronize AirTable" project aims to automate the process of gathering, synchronizing, and analyzing social media data from Facebook. The system is designed to extract various types of user data, including user IDs, feeds, liked pages, and managed pages,... This data is then seamlessly synchronized with AirTable via POST requests. Subsequently, the program retrieves the synchronized data from AirTable through GET requests and performs statistical analysis. The results of this analysis are used to generate comprehensive Excel files and visual charts (bar charts, pie charts, ...), providing valuable insights into social media interactions and trends. This project leverages the power of automated data handling and analytics to offer a robust solution for social media data management and visualization.

1 INTRODUCTION

The rapid proliferation of social media platforms has brought about a significant shift in how individuals and organizations interact, communicate, and share information. With the extensive use of platforms like Facebook, Twitter, Tiktok, .etc, there arises a critical need to address issues related to security and privacy. The "Social Media Information Reconnaissance Program and Synchronize AirTable" project is designed to address these concerns while providing robust tools for data collection, synchronization, and analysis.

The primary objective of this project is to develop a comprehensive system that automates the extraction and management of social media data. This involves a detailed exploration of security and privacy issues associated with social networks, ensuring that data collection processes adhere to ethical guidelines and legal requirements. Understanding these aspects is crucial to safeguarding user information and maintaining trust.

AirTable is utilized in this project as a versatile platform for organizing and managing the extracted data. By leveraging AirTable's API (1), the system can perform seamless data synchronization through POST requests, ensuring that all collected information is systematically stored and accessible. This step is critical for maintaining an organized database that can be easily queried and updated.

A significant component of this project involves the use of Open Source Intelligence (OSINT) techniques (2). OSINT enables the systematic collection of publicly available information from social media platforms, which can then be used for various analytical purposes. By employing these techniques, the program can gather extensive data on members, groups, hashtags, and other relevant entities on Facebook, Twitter, .etc. This data collection is performed according to predefined schedules, allowing for regular and automated updates.

Once the data is synchronized with AirTable, the program retrieves it through GET requests for further analysis. The analytical process includes statistical evaluation of the data to identify patterns, trends, and insights. The results are then visualized through various chart types, such as line and bar charts, making the information easily interpretable. Additionally, the system generates comprehensive reports in multiple formats, including Excel files, CSV files, and PDF reports. These reports are invaluable for stakeholders who require detailed and organized insights into social media activities.

For detailed guidance and technical support, this project refers to resources such as the Facebook Developers documentation (3). This ensures that the implementation is aligned with current best practices and leverages the latest available tools and APIs.
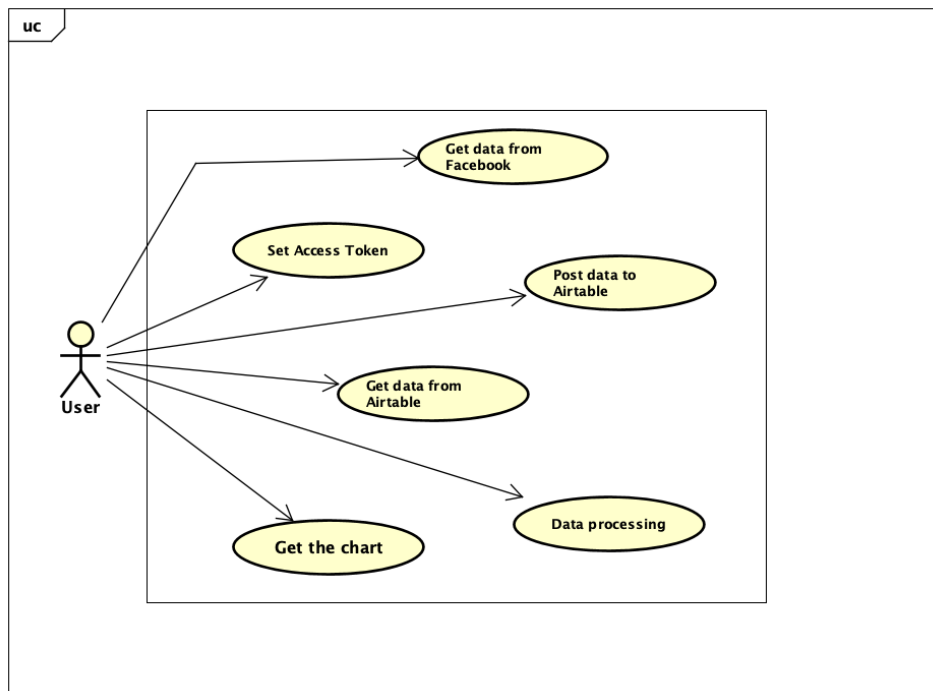
2 DESIGN

2.1 Use-case Diagram

Figure 1: Use-case Diagram

The use-case diagram illustrates the flow of operations within the "Social Media Information Reconnaissance Program and Synchronize AirTable" project, detailing how the user interacts with the system to:

- Set access tokens for authentication.
- Extract and synchronize social media data.
- Retrieve and process this data.
- Generate visualizations and reports based on the analyzed data.

This structured approach ensures that all critical tasks are covered, enabling effective data management and insightful analysis.

More detail:

- *Set Access Token:* This use case allows the user to set or update the access token required for authenticating and authorizing API requests to Facebook and AirTable.

- *Get Data from Facebook:* This use case enables the user to extract data from Facebook. The data collected can include user IDs, feeds, liked posts, and managed content. This operation is crucial for gathering the necessary social media information.

- *Post Data to AirTable:* After collecting data from Facebook, this use case allows the user to synchronize the data with AirTable by sending it via POST requests. This ensures that all the collected information is stored and organized systematically in AirTable.

- *Get Data from AirTable:* This use case involves retrieving the synchronized data from AirTable using GET requests. The user can access the stored data for further analysis or processing.

- *Data Processing:* This use case handles the analysis and statistical evaluation of the retrieved data. It involves processing the data to identify patterns and insights. The results of this analysis are crucial for generating visualizations.

- *Get the Chart:* Based on the processed data, this use case allows the user to generate visual charts such as pie and bar charts. These charts help in visualizing the insights and trends identified during data processing, making the information more interpretable and actionable.
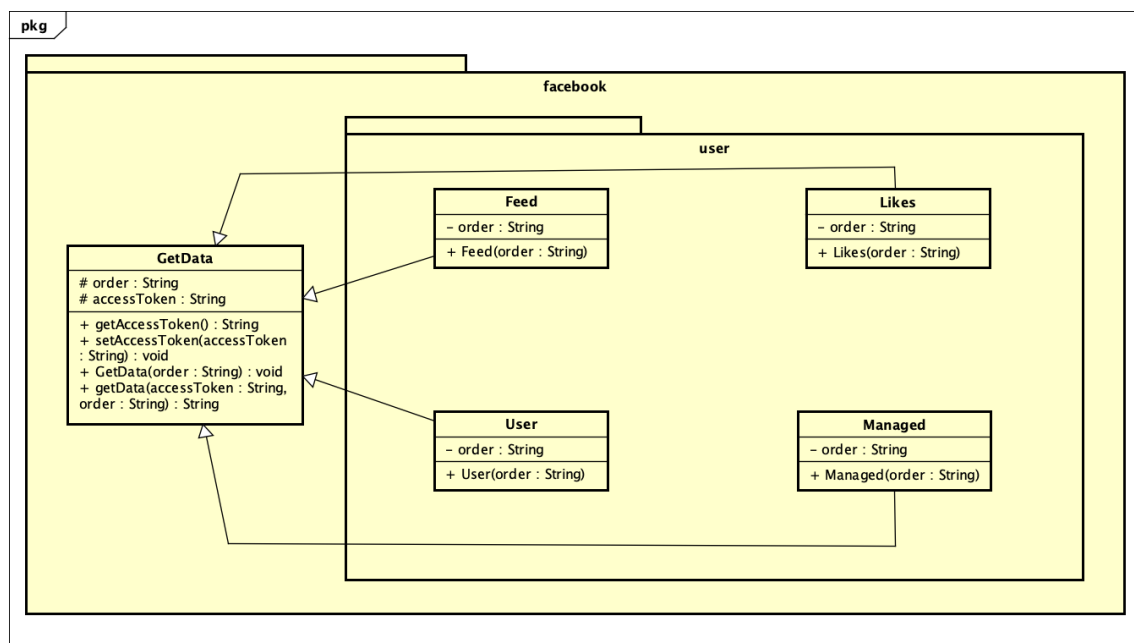
2.2 Class Diagram

Facebook package:

Figure 2: Facebook Package Class Diagram

The Facebook package consists of several classes related to fetching and managing data from Facebook.

GetData class:

- **Attributes:**
    - *order:* A String attribute that holds the order information.
    - *accessToken:* A static String attribute that holds the access token for making API requests.

- **Methods:**
    - *getAccessToken():* Retrieves the current access token.
    - *setAccessToken(accessToken : String):* Sets the access token.

- *GetData(order : String):* Initiates the process to get data based on the specified order.
- *getData(accessToken : String, order : String):*
    * Takes an accessToken and an order as parameters.
    * Constructs a URL using the Facebook Graph API endpoint and the given parameters.
    * Opens an HTTP connection to the constructed URL.
    * Sends a GET request to the URL.
    * Reads the response from the server.
    * Returns the response as a String.

GetData has associations with **Feed**, **User**, **Likes**, and **Managed** classes, indicating it handles or interacts with these types of data.
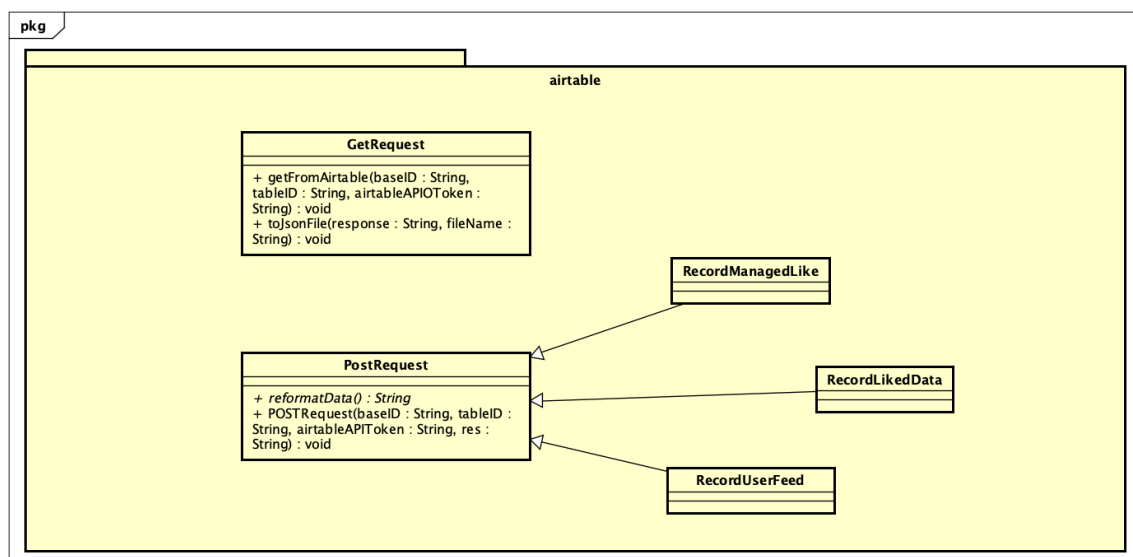
Airtable package:

Figure 3: AirTable Package Class Diagram

The Airtable package consists of classes responsible for handling the interaction with AirTable, including posting and retrieving data.

GetRequest class:

- **Methods:**
    - *getFromAirtable(baseID : String, tableID : String, airtableAPIToken : String):*
        * Parameters:
            · *baseID:* The ID of the Airtable base.
            · *tableID:* The ID of the Airtable table.
            · *airtableAPIToken:* The API token for authentication with Airtable.
        * Functionality:
            · Constructs a URL to the Airtable API endpoint for the specified base and table.

· Opens an HTTPS connection to the constructed URL.
· Sets the request method to **GET**.
· Sets the **Authorization** header to include the Airtable API token.
· Reads the response from the server.
· Returns the response as a String.

– *toJsonFile(response : String, fileName : String):*
  * Parameters:
    · *response:* The response string to be written to a file.
    · *fileName:* The name of the file where the JSON object will be written.
  * Functionality:
    · Attempts to parse the response string into a JSONObject.
    · Writes the JSON object to a file with the specified file name.
    · Handles exceptions related to JSON parsing, I/O operations, and other unexpected errors.
    · Ensures the **FileWriter** is closed properly in the finally block.

PostRequest class:

• **Methods:**

– *reformatData():*
  * This is an abstract method that must be implemented by any subclass of PostRequest.
  * It is intended to be used for reformatting data before making a POST request.
  * Throws an IOException.
– *POSTRequest(baseID : String, tableID : String, airtableAPIToken : String, res : String):*
  * Parameters:
    · *baseID:* The ID of the Airtable base.
    · *tableID:* The ID of the Airtable table.
    · *airtableAPIToken:* The API token for authentication with Airtable.
    · *res:* The data to be sent in the POST request body.
  * Functionality:
    · Constructs a URI for the Airtable API endpoint for the specified base and table.
    · Converts the URI to a URL.
    · Opens an HTTP connection to the constructed URL.
    · Sets the request method to **POST**.
    · Enables output for the connection.
    · Sets the **Authorization** header to include the Airtable API token.
    · Sets the **Content-Type** header to .
    · Writes the data to the output stream in UTF-8 encoding.
    · Prints the response code and message.
    · Disconnects the HTTP connection.

GetRequest and **PostRequest** are associated with **RecordManagedLike**, **RecordLikedData**, and **RecordUserFeed**, indicating these operations involve handling these types of records.

3   DEMONSTRATION

This demo section showcases the practical implementation and functionality of the "Social Media Information Reconnaissance Program and Synchronize AirTable" project. Below is a step-by-step guide demonstrating how to get data from Facebook, synchronize it with AirTable using an access token, and then retrieve and analyze the data using various options, with the output visualized in Excel files and charts.

- **Accessing the Menu:**
  After setting up the access token and ensuring the system is ready to retrieve data from Facebook, the user is presented with a menu of options. Each option corresponds to a different type of data retrieval and analysis requirement.
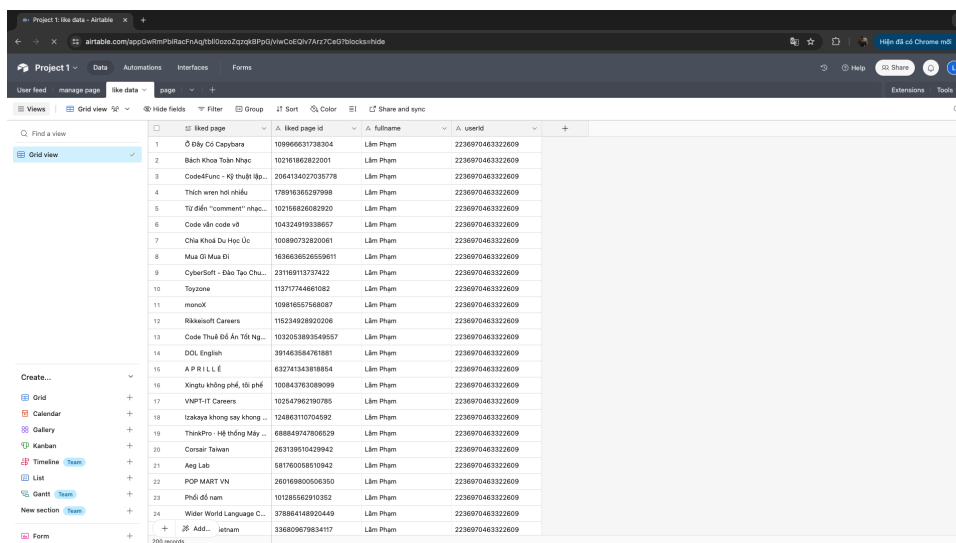
Figure 4: Menu

- **Synchronizing Data with AirTable:**
  Once the user selects an option from the menu, the system retrieves the relevant data from Facebook and syncs it to AirTable using the provided access token. The synchronized data is organized in AirTable for easy access and management.

Figure 5: Synchronizing Data with AirTable

- **Retrieving and Analyzing Data, Visualizing Data with Charts:**
  The user can then choose to retrieve data from AirTable based on specific requirements.
  The system processes the data, performing statistical analysis and generating reports.

Figure 6: Data in Excel file

The analyzed data can be visualized using various types of charts to provide insights and facilitate decision-making. The system supports generating bar charts and pie charts, among other visualization formats.
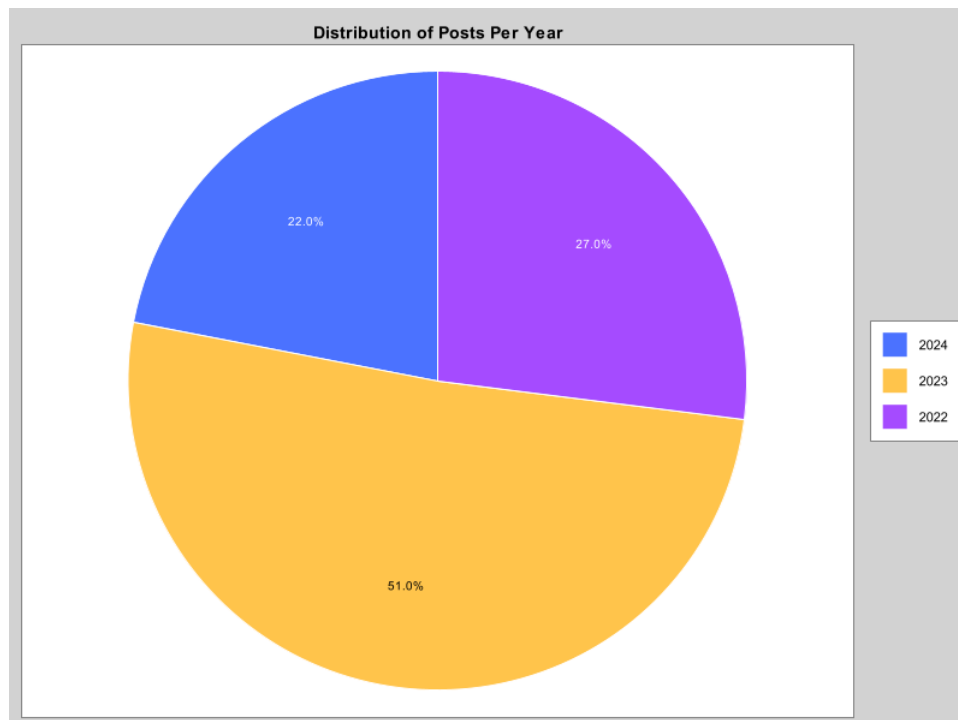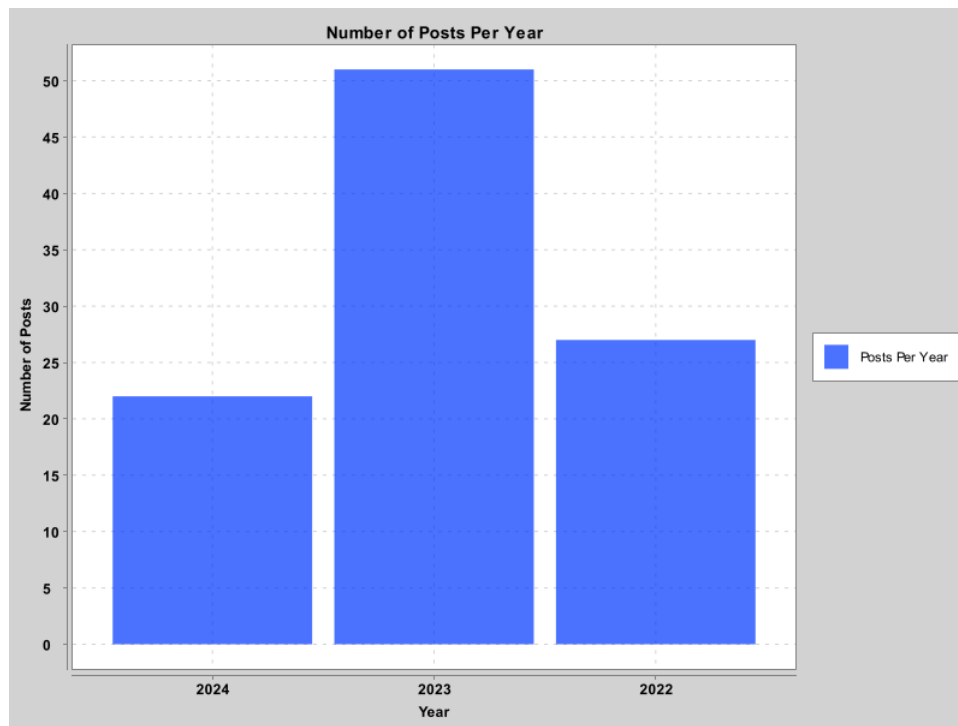

Figure 7: User Pie Chart

Figure 8: User Bar Chart

4 VERSION CONTROL AND CODE QUALITY ANALYSIS

4.1 Version Control

To manage our code effectively, we used Github as our control system. We created a repository name **Project_1** (4) where all project-related and documentation were stored. In repository, we organized it as follow:

- **Master Branch**: This branch served as the complete version of our code.
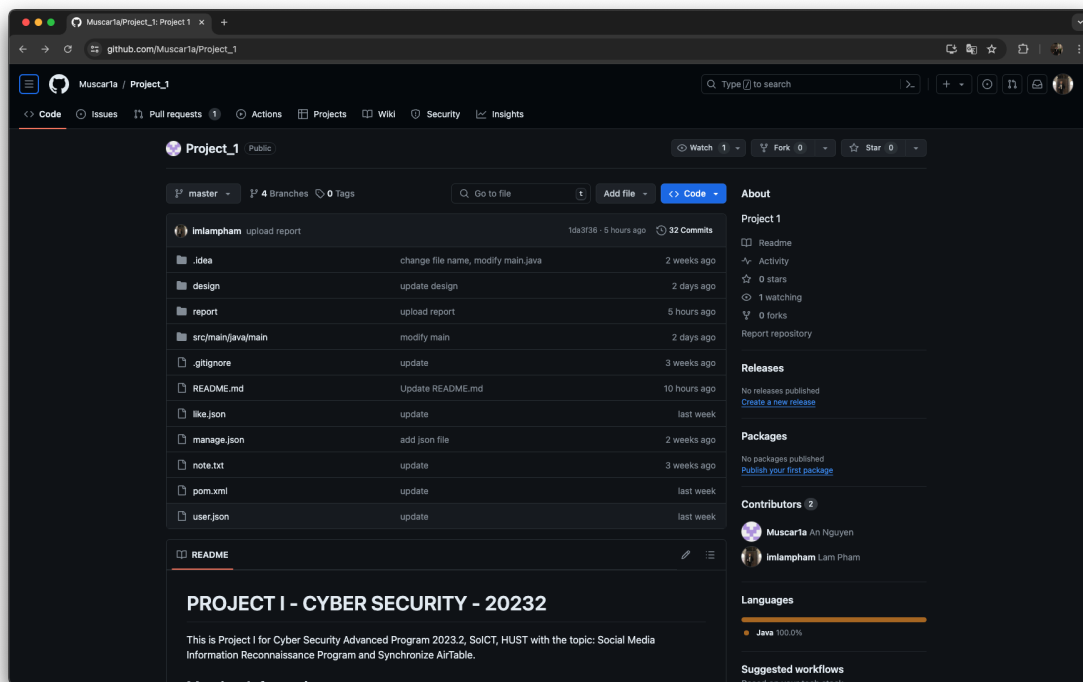- **Feature Branch**: We have 3 other separated branches to code different functions.

Figure 9: Control System

4.2 Code Quality Analysis

We implemented Jenkins (5) for our Continuous Integration (CI) needs. Jenkins pipeline was set up to automates the build, test and deployment:

- **Build**: Jenkins automatically built the project upon new code commits, ensuring the latest code was complied and ready for testing
- **Test**: A suit of automated tests ran after each build to verify the code's functionality.
- **Deploy**: Jenkins deployed the tested code to a staging environment for further validation.

To maintain high code quality, we integrated SonarQube with Jenkins. SonarQube analyzed our code for potential bugs or vulnerabilities and provided detailed reports on code quality. Using **SonarQube Scanner** tool, SonarQube was integrated into Jenkins pipeline, which executed code analysis during Jenkins build process. The analysis included metrics such as bugs, vulnerabilities, code smells. SonarQube also displayed a real-time insights into code quality, highlighting areas that needed to be improved or to be fixed.
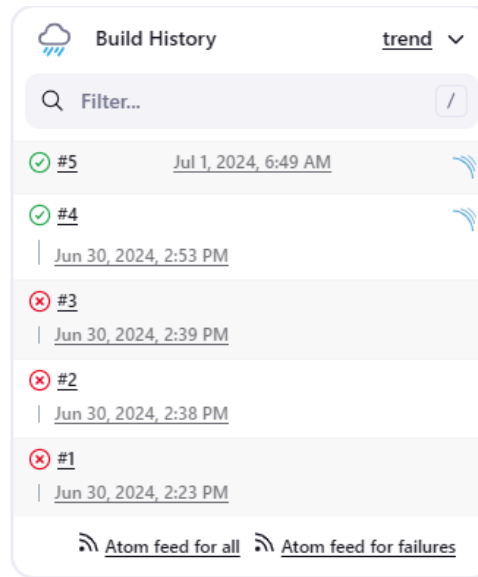
This is our build history on Jenkins:

Figure 10: Build History on Jenkins

At first, we Jenkins build failed, and returned the error like this:

```
Also:    org.jenkinsci.plugins.workflow.actions.ErrorAction$ErrorId: 6a8901fa-2ba9-4f40-8a28-d0c237b09051
java.lang.NoSuchMethodError: No such DSL method 'git' found among steps [archive, bat, build, catchError,
checkout, compareVersions, deleteDir, dir, echo, error, fileExists, findBuildScans, findFiles, getContext,
input, isUnix, junit, library, libraryResource, load, mail, milestone, node, nodesByLabel, parallel,
```

Figure 11: Error

After some researches, we found that update all installed plugins solved the problem.
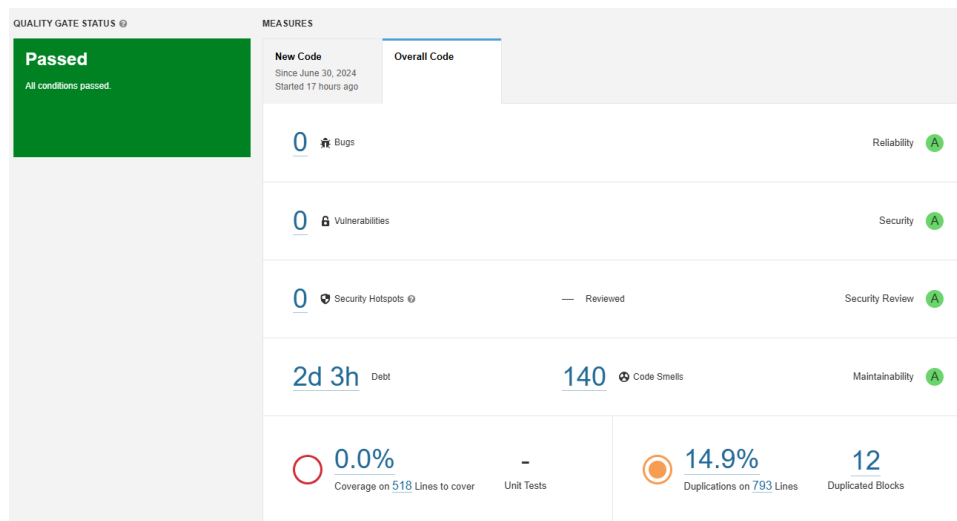

Figure 12: SonarQube Score

Above is the response from SonarQube when we scan our code through Jenkins, which shows **0** bugs, **0** vunelrabilities, **0** security hotspots, but there are some code smells.

After several researches, we concluded that these smells do not alter the code or affect the security of our program.

5   CONCLUSION

The "Social Media Information Reconnaissance Program and Synchronize AirTable" project successfully addresses the need for an integrated tool that can efficiently collect, synchronize, and analyze social media data from Facebook. By leveraging Facebook's Graph API and AirTable's API, the project provides a seamless solution for data management and analysis, enabling users to gain valuable insights and make informed decisions based on social media activities.

- *Comprehensive Data Collection:* The project facilitates the retrieval of various types of data from Facebook, ensuring a broad and detailed dataset for analysis.
- *Efficient Data Synchronization:* The synchronization process with AirTable ensures that data is organized and up-to-date, providing a reliable database for further analysis.
- *Robust Data Analysis:* The statistical analysis and visualization tools included in the project allow users to derive meaningful insights from the data, presented in accessible formats such as Excel reports and charts.
- *Enhanced Security and Privacy:* The use of access tokens and adherence to privacy practices ensure that user data is handled securely throughout the process.

The integration of Facebook and AirTable APIs has proven to be an effective approach, offering a scalable and versatile solution for social media information reconnaissance.

REFERENCES

[1] Airtable's api, `https://airtable.com/developers/web/api/introduction`.

[2] Osint technique, `https://www.imperva.com/learn/application-security/open-source-intelligence-osint/`.

[3] Facebook developers, `https://developers.facebook.com/docs/`.

[4] Github of project, `https://github.com/Muscar1a/Project_1`.

[5] Jenkins documentation, `https://www.jenkins.io/doc/`.