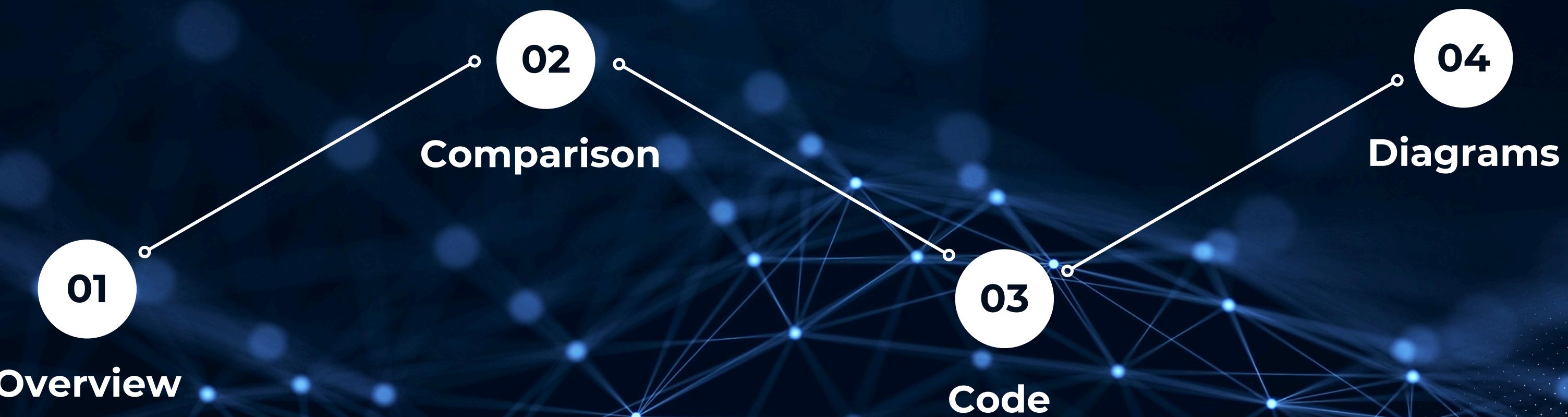


# PARALLEL PROCESSING PROJECT

SOCIAL MEDIA FETCHER  
IDEA

# EXECUTIVE SUMMARY



# PROJECT OVERVIEW

The Social Media Fetcher is a tool designed to collect posts and content from multiple social media platforms simultaneously. It aims to provide users with an efficient way to view updates from different social media sites like Instagram and Reddit in one consolidated feed

# COMPARISON

## PARALLEL VS SEQUENTIAL FETCHING

### PARALLEL

- **DEFINITION:** DATA IS FETCHED FROM MULTIPLE PLATFORMS AT THE SAME TIME USING DIFFERENT THREADS.
- **SPEED:** FASTER, AS TASKS ARE EXECUTED CONCURRENTLY.
- **EFFICIENCY:** MORE EFFICIENT FOR LARGE VOLUMES OF DATA SINCE MULTIPLE REQUESTS ARE HANDLED SIMULTANEOUSLY.
- **RESOURCE USAGE:** HIGHER RESOURCE CONSUMPTION, ESPECIALLY CPU AND MEMORY, DUE TO MULTIPLE THREADS RUNNING AT ONCE.
- **COMPLEXITY:** MORE COMPLEX TO IMPLEMENT BECAUSE OF POTENTIAL SYNCHRONIZATION ISSUES AND THREAD MANAGEMENT.
- **USE CASE:** BEST FOR REAL-TIME UPDATES AND WHEN YOU NEED TO FETCH DATA FROM MULTIPLE SOURCES SIMULTANEOUSLY.

### SEQUENTIAL

- **DEFINITION:** DATA IS FETCHED FROM ONE PLATFORM AT A TIME, WITH EACH REQUEST COMPLETING BEFORE THE NEXT BEGINS.
- **SPEED:** SLOWER, AS EACH TASK WAITS FOR THE PREVIOUS ONE TO FINISH.
- **EFFICIENCY:** LESS EFFICIENT FOR LARGE DATA FETCHES, AS ONLY ONE REQUEST IS HANDLED AT A TIME.
- **RESOURCE USAGE:** LOWER RESOURCE CONSUMPTION, AS ONLY ONE THREAD IS USED AT A TIME.
- **COMPLEXITY:** SIMPLER TO IMPLEMENT AS THERE IS NO NEED TO MANAGE MULTIPLE THREADS OR WORRY ABOUT SYNCHRONIZATION.
- **USE CASE:** SUITABLE FOR SMALLER TASKS OR WHEN SYSTEM RESOURCES ARE LIMITED, AND REAL-TIME UPDATES ARE NOT CRITICAL.

# CODE EXPLANATION OF THE SOCIAL MEDIA FETCHING

MAIN  
FUNCTIONS

Sequence of  
Running the  
Code

# MAIN FUNCTIONS

`safe_insert_text`

**FUN**

`fetch_instagram_posts`

**FUN**

`fetch_reddit_posts`

**FUN**

`auto_refresh`

**FUN**

`fetch_posts`

**FUN**

`run_sequential`

**FUN**

`run_parallel`

**FUN**

`create_windows`

**FUN**

## `safe_insert_text`

**Inserts text safely into the output widget from a different thread.**

## `fetch_instagram_posts`

**Fetches Instagram posts using the Graph API and displays them in the output widget.**

## `fetch_reddit_posts`

**Fetches posts from a specified Reddit subreddit using Reddit's API and displays them in the output widget.**

## `auto_refresh`

**Automatically refreshes the Instagram and Reddit posts every 10 seconds.**

# EXPLAINING

## `fetch_posts()`

**Initiates fetching posts from Instagram and Reddit based on the selected fetch method (sequential or parallel).**

## `run_sequential()`

**Fetches Instagram and Reddit posts sequentially, one after the other, for the specified number of threads.**

## `run_parallel()`

**Fetches Instagram and Reddit posts in parallel using threads for the specified number of threads.**

## `create_windows()`

**Creates separate windows for Instagram and Reddit posts for each thread and returns the window and output widget objects.**

# EXPLAINING

# SEQUENCE OF RUNNING THE CODE



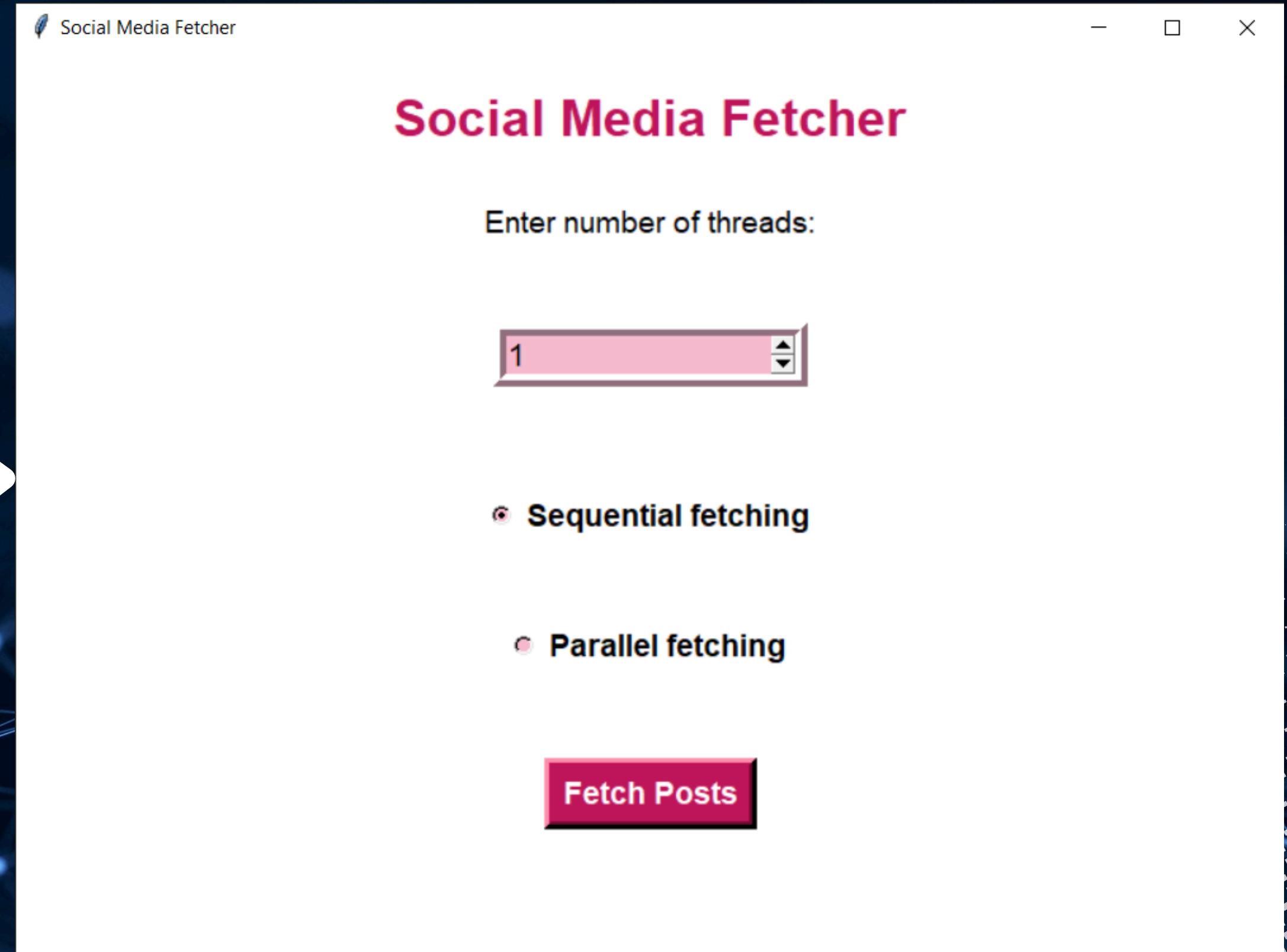
# SEQUENCE OF RUNNING THE CODE (2)



# SEQUENCE OF RUNNING THE CODE

## Program Start:

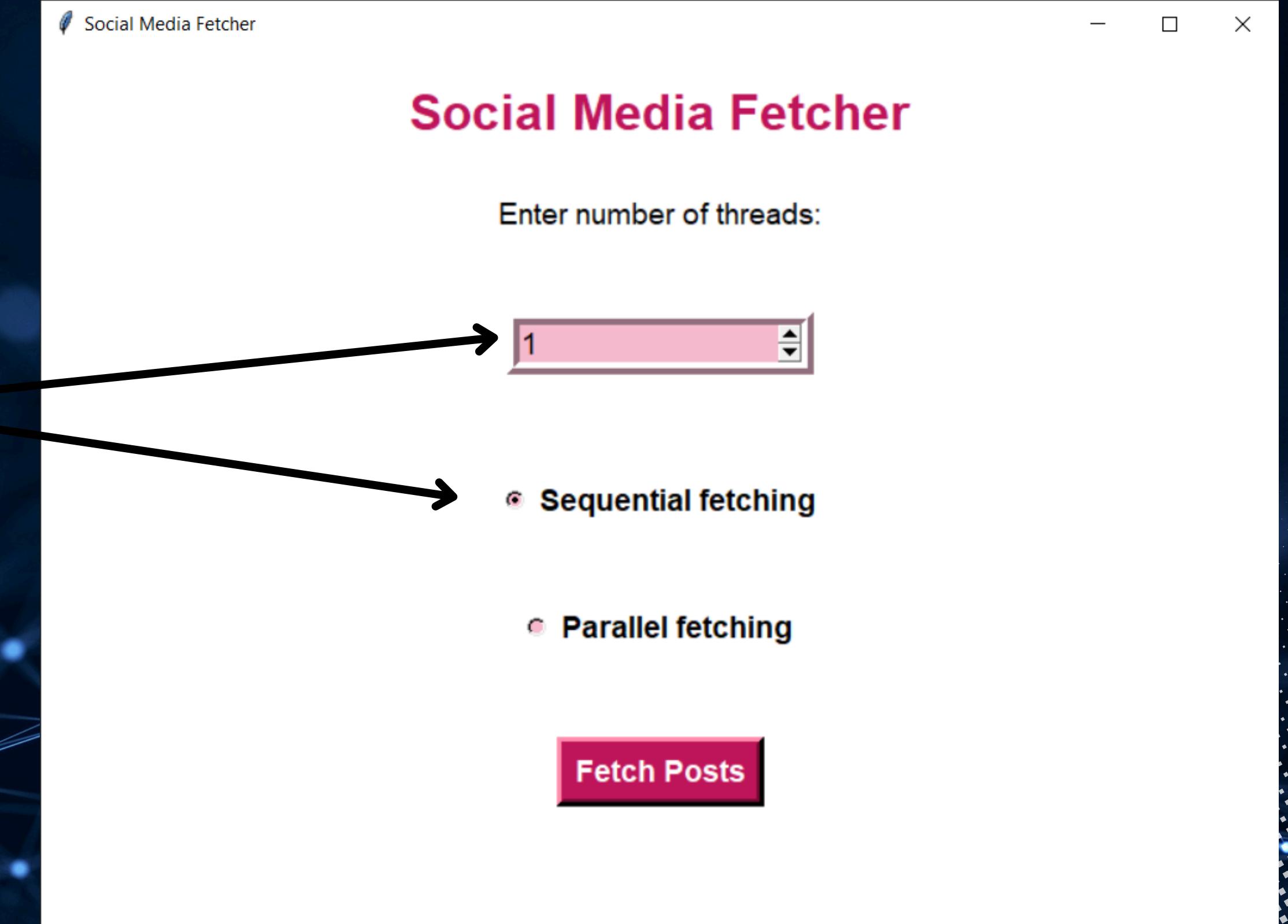
The program starts by creating the main GUI window using tkinter.



# SEQUENCE OF RUNNING THE CODE

## User Input:

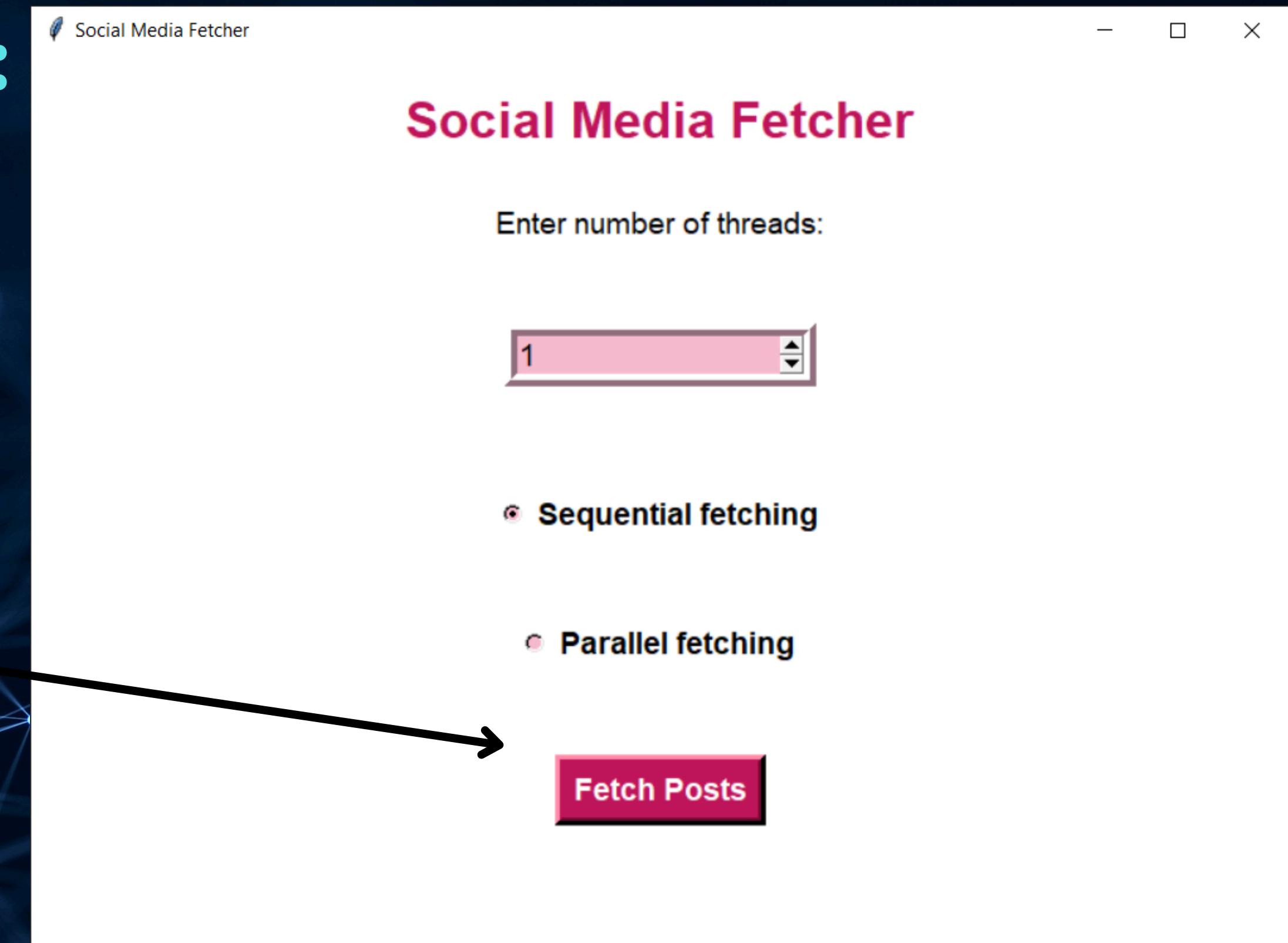
- The user enters the number of threads using the Spinbox widget.
- The user selects whether they want to fetch posts sequentially or in parallel using the RadioButton options.



# SEQUENCE OF RUNNING THE CODE

## Fetch Button Clicked:

**When the user clicks the "Fetch Posts" button, the `fetch_posts()` function is triggered.**



# SEQUENCE OF RUNNING THE CODE

## Fetching Logic (Sequential or Parallel):

Inside `fetch_posts()`, it checks whether the user chose "Sequential" or "Parallel" fetching:

### If Sequential:

- Calls the `run_sequential()` function.

### If Parallel:

- Calls the `run_parallel()` function

# SEQUENCE OF RUNNING THE CODE

## Thread Creation (Sequential or Parallel):

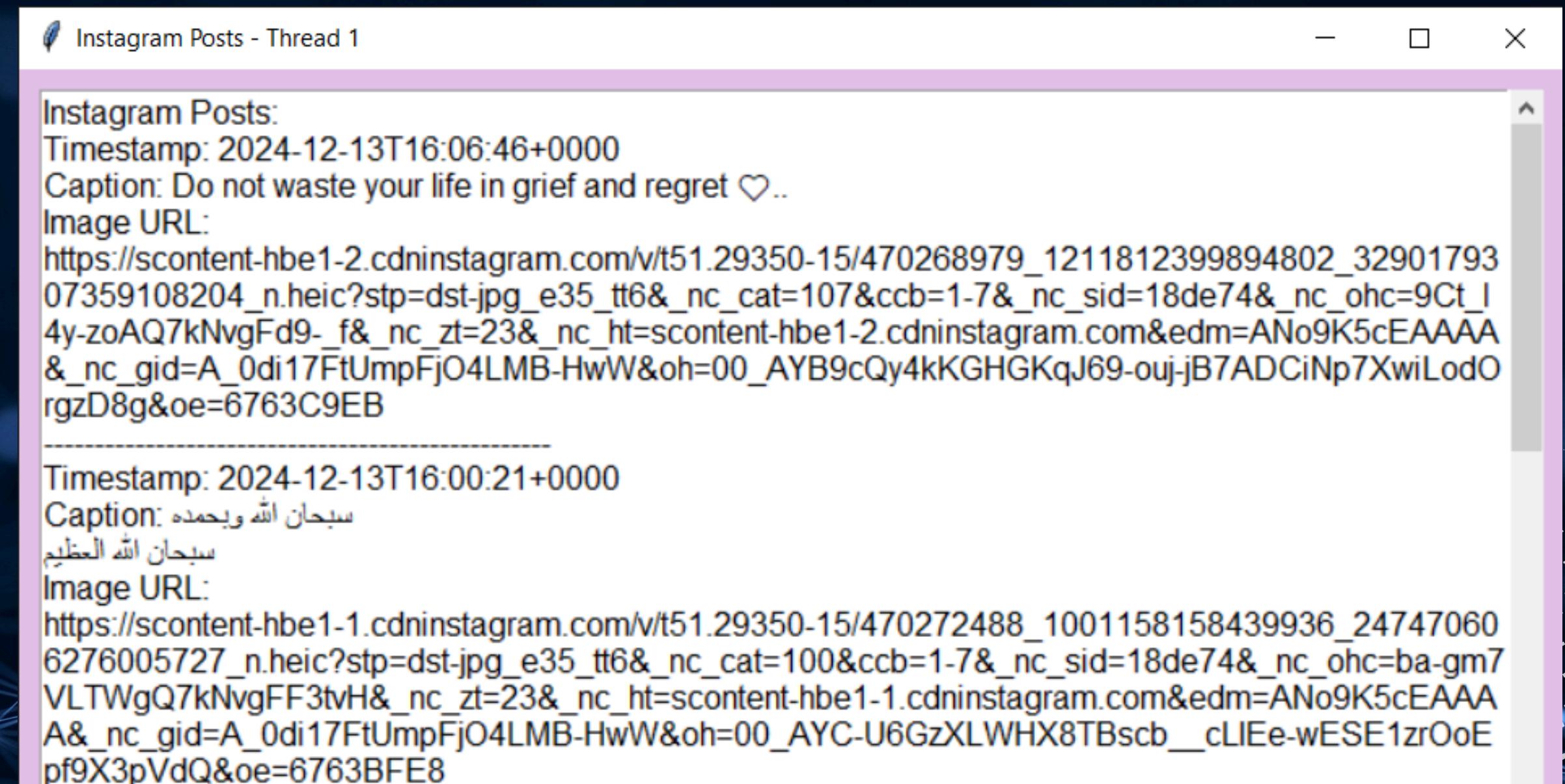
**For each thread (based on the number entered by the user):**

- Windows for Instagram and Reddit posts are created using `create_windows(thread_index)`

# SEQUENCE OF RUNNING THE CODE

## Thread Creation (Sequential or Parallel):

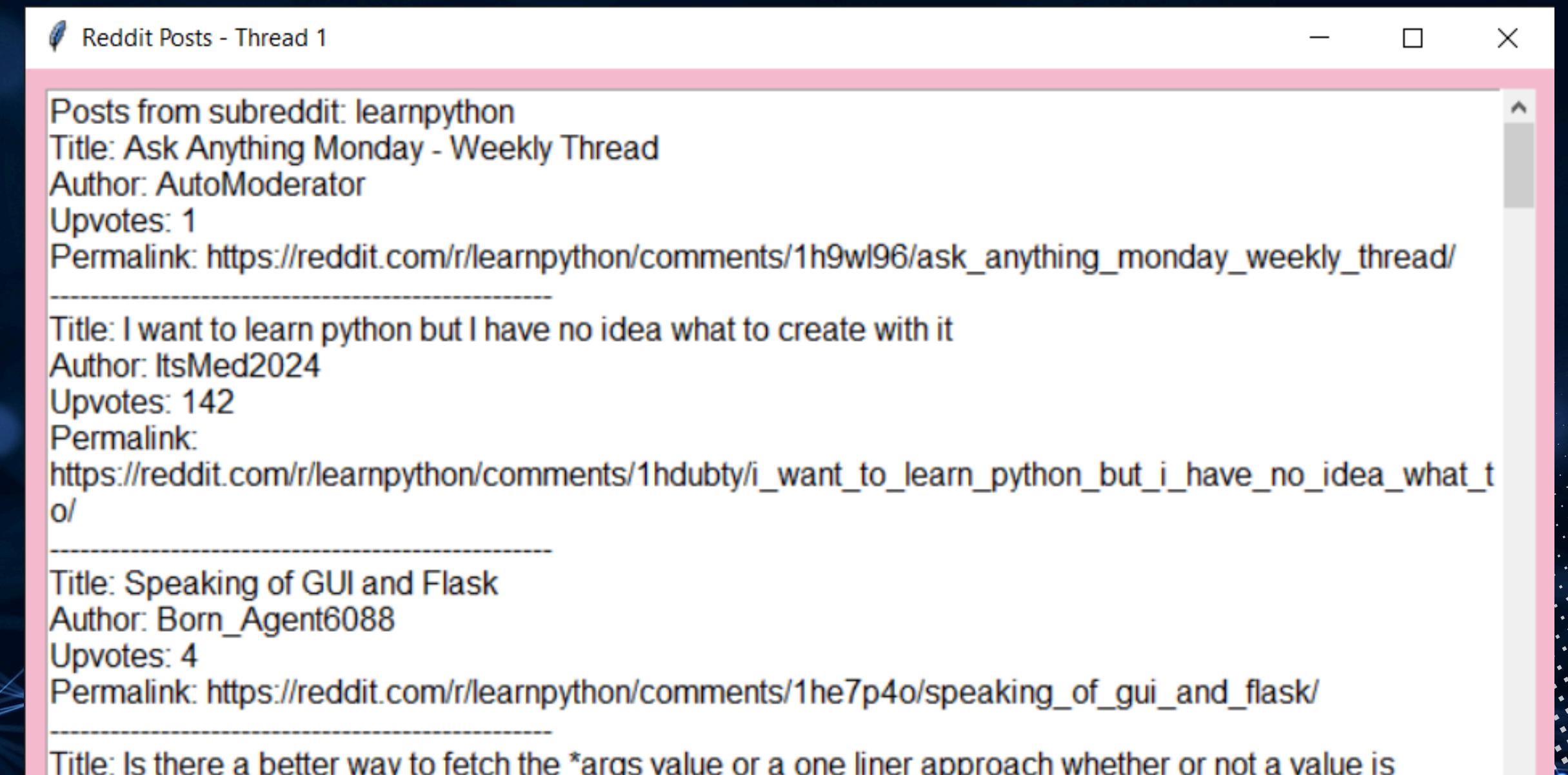
**fetch\_instagram\_posts()**  
**is called to fetch Instagram posts and display them in the respective window**



# SEQUENCE OF RUNNING THE CODE

## Thread Creation (Sequential or Parallel):

**fetch\_reddit\_posts()**  
is called to fetch  
Reddit posts and  
display them in  
the respective  
window



# SEQUENCE OF RUNNING THE CODE

## Fetching Logic (Sequential or Parallel):

### Auto-refresh Mechanism:

After the initial fetch, the function `auto_refresh()` is called to refresh Instagram and Reddit posts every 10 seconds

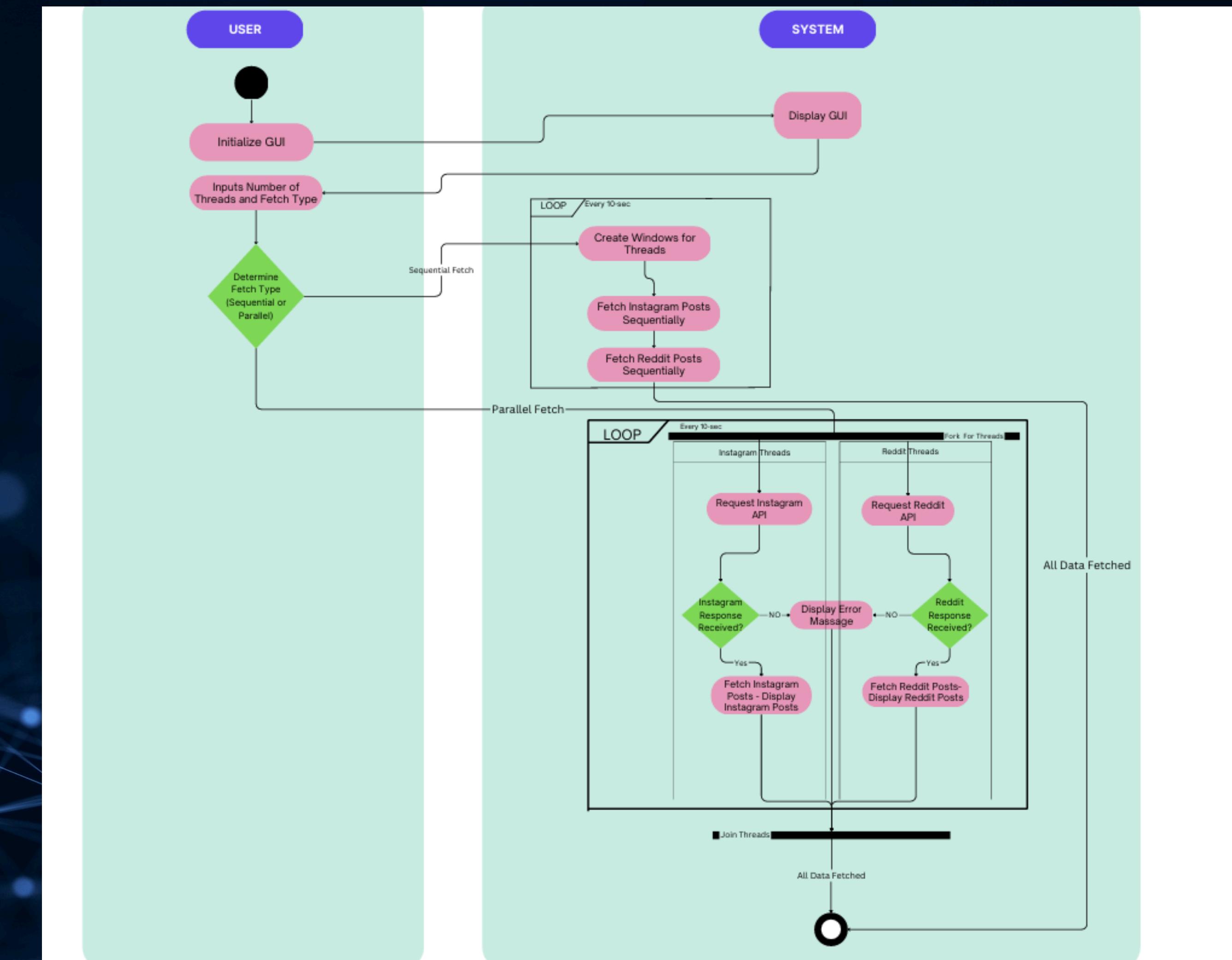
### Multithreading (for Parallel Fetching):

- If "Parallel" fetching is selected:  
**New threads are created for each API call to fetch Instagram and Reddit posts in parallel using threading.Thread**

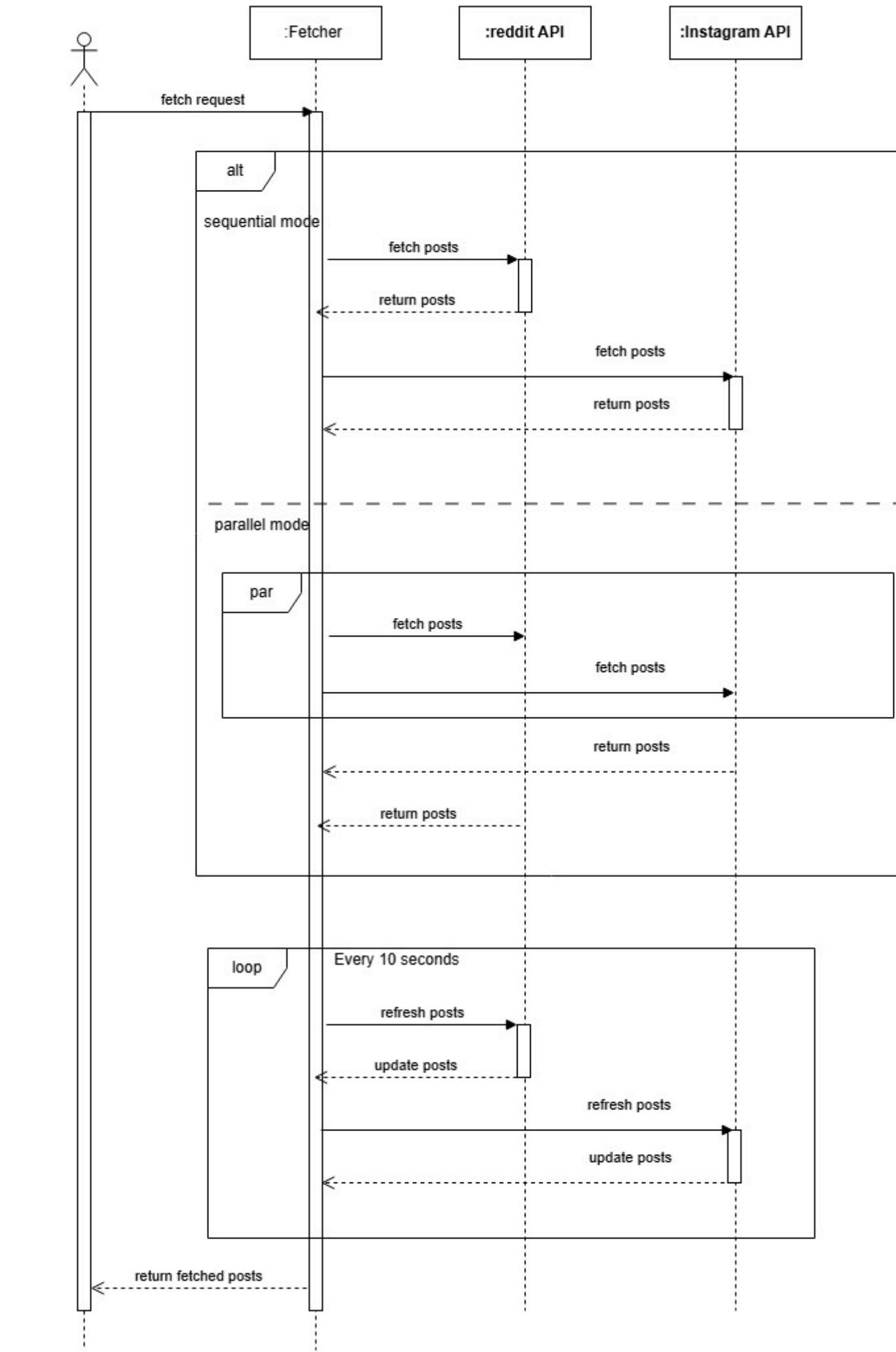
### Data Display and Refresh:

- For each fetched post (from Instagram or Reddit), the posts are displayed in their respective scrolled text output widgets.
- The posts are auto-refreshed periodically.

# ACTIVITY DIAGRAM



# SEQUENCE DIAGRAM



# THANK YOU

BY: SHROUK MOSTAFA