CS 421 Programming Assignment 1 – PseudoGit

Görkem Kadir Solun 22003214

I have done all the functionalities, and you will be able to run the tool successfully.

# Core Features

## Clone

## The get\_file\_from\_github function retrieves a file from a GitHub repository, accepting file\_name, directory, and parallel\_count as parameters (with defaults for the last two). It opens a secure socket, sends an HTTP GET request via send\_request, and closes it. The response body is parsed as JSON, and the file content is Base64-decoded if present. If non-empty, the content is written to a file in the specified directory. If empty, the function calculates the file size and download URL, splits the file into chunks, and creates threads to download each chunk in parallel. Once all threads are complete, the chunks are combined, and temporary files are removed.

## The download\_file\_chunk function downloads a file chunk from a URL. It accepts URL, start, end, file\_name, and directory parameters. It creates a secure socket, constructs an HTTP GET request with a Range header, sends the request, and writes the received chunk to a file in the specified directory before closing the socket.

## The get\_repository\_contents function fetches the contents of a GitHub repository with an optional path parameter for subdirectories. It creates a secure socket, sends an HTTP GET request, parses the JSON response, and returns a list of files and their types (e.g., "file" or "dir").

## The download\_files function downloads multiple files from a repository, accepting files, directory, and parallel\_count parameters (with defaults for the latter two). It iterates through the files, creating directories as needed for subdirectories, and recursively calls get\_repository\_contents and download\_files. For regular files, it creates threads to download each file using get\_file\_from\_github, ensuring no more than MAX\_THREAD\_COUNT threads run concurrently. After starting all threads, it waits for them to finish.

## Upload File

## The push\_changes function pushes changes to a specified branch in a GitHub repository. It takes file\_name, branch\_name, and an optional message (defaulting to "Pushed changes"). First, it retrieves the file's SHA using get\_file\_sha, then reads and encodes the file content in Base64. Using this information, it constructs a JSON payload with the commit message, encoded content, branch name, and file SHA if available. The function creates a secure socket, sends an HTTP PUT request with headers and the payload via send\_request, and closes the socket. Based on the response status code, it prints a success message for updates (200) or creation (201); otherwise, it indicates failure.

## The get\_file\_sha function retrieves the SHA of a specified file. It takes file\_name as a parameter, creates a secure socket, and sends an HTTP GET request with the necessary headers. After sending the request via send\_request, it closes the socket, parses the JSON response to extract the SHA, and returns it if available; otherwise, it returns None.

## Create Branch

## The create\_branch function creates a new branch in a GitHub repository. It takes branch\_name as a parameter and retrieves the latest commit SHA using get\_latest\_commit\_sha. The function then creates a secure socket, sends an HTTP POST request with headers and a JSON body containing the branch name, and commits SHA via send\_request, closing the socket afterward. If the response status code is 201, it confirms branch creation; otherwise, it indicates failure.

## The get\_latest\_commit\_sha function retrieves the latest commit SHA for a specified branch. It creates a secure socket, sends an HTTP GET request with necessary headers via send\_request, and closes the socket. The JSON response is parsed to extract and return the latest commit SHA.

## List Pull Requests

## The list\_open\_pull\_requests function retrieves open pull requests for a specified GitHub repository. It creates a secure socket, sends an HTTP GET request with necessary headers via send\_request, and closes the socket. The JSON response is parsed, and a pull request number list is returned.

## Create Pull Request

## The create\_pull\_request function creates a new pull request in a GitHub repository. It takes title, body, head, and base as parameters for the pull request title, body, head branch, and base branch. The function creates a secure socket, sends an HTTP POST request with headers and a JSON payload via send\_request, and closes the socket. If the response status code is 201, it confirms successful creation; otherwise, it indicates failure.

## Merge Pull Request

## The merge\_pull\_request function merges a specified pull request in a GitHub repository. It takes pull\_request\_number as a parameter, creates a secure socket, and sends an HTTP PUT request with headers via send\_request. After closing the socket, it checks the response status code. If 200, it confirms the merge was successful; otherwise, it indicates failure.

# Additional Features

## Close Pull Request

## The close\_pull\_request function closes a specified pull request in a GitHub repository. It takes pull\_request\_number as a parameter, creates a secure socket, and sends an HTTP PATCH request with headers and a JSON body to change the pull request state to "closed" via send\_request, then closes the socket. If the response status code is 200, it confirms successful closure; otherwise, it indicates failure.

## Delete Branch

# The delete\_branch function deletes a specified branch from a GitHub repository. It takes branch\_name as a parameter, creates a secure socket, sends an HTTP DELETE request with necessary headers via send\_request, and then closes the socket. If the response status code is 204, it confirms successful deletion; otherwise, it indicates failure.

# Other Functions

The create\_secure\_socket function creates a secure SSL/TLS socket. It takes an optional server\_hostname parameter, defaulting to GITHUB\_API. SSL.create\_default\_context() wraps a standard socket in SSL for secure communication and returns it.

The send\_request function sends an HTTP request to a server and retrieves the response. It takes secure\_socket, host, port, and request as parameters. The function connects to the server, sends the encoded request, and receives the response in chunks combined into a single-byte string. It splits the response into headers and body, extracts the status code, and decodes it. It returns a dictionary with the status code, response body, and headers.