AUTOMATION PROJECT

This project consist of the journey I went through from scratch to place of producing content that looks more professional and ability to set my machine work with github

Table of Contents

1ST POST

PROBLEM 1: OVER REPEATION OF WRITING FILE PATHS

I had a problem of writing file paths which some consisted of long filenames like repositories and this could make me take long time to access a file in a sub folder from other many sub folders

SOLUTION

I had to initiate auto completion of file names on my terminal on my Mac.

STEP 1: Install a shell and one them is ZSH SHELL

To install Zsh (Z Shell), you can follow these general steps. The process might vary slightly depending on
your operating system.

- **For Linux: Ubuntu/Debian-based systems:**
  Open a terminal.

Run the following command to install Zsh:

```
sudo apt-get update
sudo apt-get install zsh
```

After installation, you can start Zsh by typing zsh and pressing Enter.

- **For macOS:**
  Zsh is pre-installed on macOS. You can check the installed version by typing zsh --version in the terminal.

```
To start Zsh, simply type zsh and press Enter.
```

- **For Windows:**
  Using Windows Subsystem for Linux (WSL):
  If you don't have WSL installed, follow the instructions on the official Microsoft documentation to set up WSL.

Open the terminal in your WSL instance.

Run the following command to install Zsh:

```
bash
    Copy code
    sudo apt-get update
    sudo apt-get install zsh
```

After installation, you can start Zsh by typing zsh and pressing Enter.

During the installation process, make sure to select the zsh package from the package list..

After installation, you can start Zsh by running the Cygwin terminal and typing zsh and pressing Enter.

Customization (Optional):
Once Zsh is installed, you may want to customize it further using frameworks like Oh My Zsh or Prezto. These frameworks provide additional features, themes, and plugins. You can find installation instructions
on their respective GitHub repositories:

```
Oh My Zsh
Prezto
```

Keep in mind that the installation steps may change, so it's always a good idea to refer to the official documentation or the respective GitHub repositories for the most up-to-date information.

STEP 2. Install command: (this I did by running the command below on my terminal)

```
For Mac
brew install zsh-autosuggestions
```

STEP 3. Activating the auto suggestions

```
To activate the autosuggestions,l added the following at the end of your
.zshrc
to open .zshrc type vim .zshrc in your terminal

   source $(brew --prefix)/share/zsh-autosuggestions/zsh-
autosuggestions.zsh
```

STEP 4. Final step

At last I closed the terminal and opened it again then next was magic happening on my termminal where I got suggestipns of commands which had written before and this helped me to save time I would spend on typing

POST 2

PROBLEM: ADDITION OF TEXT IN BLOG CONSECUTIVELY WITH USE OF THE TERMINAL

I had to add text in my blog several times but without updating it daily my main blog at the github account

SOLUTION.

```
I used VIM as a tool to help me edit text in my blog and save it locally
on my machine several times using
    the terminal and then push it once on github
```

STEP 1

```
    I had to first have home brew on my mac and then install then vim
respectively following the commands below
Open Terminal (you can find it using Spotlight with Cmd + Space and
typing "Terminal").
```

If you don't have Homebrew installed, you can install it by running the following command:

```
  bash
 "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Once Homebrew is installed, you can install Vim with the following command:

STEP 2

```
bash
Copy code
brew install vim
```

After the installation is complete, you can run Vim by typing vim in the Terminal.

BASIC COMMANDS OF VIM THAT EASED MY CONTENT CREATION

Normal Mode:

Default mode for navigation and manipulation.
- Press Esc to enter Normal Mode.

Insert Mode:

Used for inserting or editing text.
- Press i in Normal Mode to enter Insert Mode.

Visual Mode:
Used for selecting and manipulating text.
Press v in Normal Mode to enter Visual Mode.

Navigation:
Move cursor:
h: Move left
j: Move down
k: Move up
l: Move right

Word-wise navigation:

```
w: Move to the beginning of the next word
b: Move to the beginning of the previous word
e: Move to the end of the current word
```

Line-wise navigation:

```
0: Move to the beginning of the line
^: Move to the first non-blank character of the line
$: Move to the end of the line
G: Move to the end of the file
gg: Move to the beginning of the file
:<line_number>: Move to a specific line number
Editing:
```

Inserting text:

```
Press i to enter Insert Mode before the cursor
Press I to enter Insert Mode at the beginning of the line
Press a to enter Insert Mode after the cursor
Press A to enter Insert Mode at the end of the line
Press o to open a new line below the current line
Press O to open a new line above the current line
```

Deleting text:

```
x: Delete the character under the cursor
dd: Delete the entire line
D: Delete from the cursor position to the end of the line
dw: Delete from the cursor position to the end of the word
db: Delete from the cursor position to the beginning of the word
Saving and Quitting:
Save:

:w: Save changes
```

Quit:

```
:q: Quit (if no changes were made)
:q!: Quit without saving changes
:wq or ZZ: Save and quit
Visual Mode:
Selecting text:
v: Enter Visual Mode (character-wise)
V: Enter Visual Mode (line-wise)
Ctrl + v: Enter Visual Mode (block-wise)
Copy and Paste:
```

Copy (yank):

```
yy: Copy the current line
y$: Copy from the cursor position to the end of the line
yw: Copy from the cursor position to the end of the word
```

Cut (delete):

```
dd: Cut (delete) the current line
dw: Cut (delete) from the cursor position to the end of the word
```

Paste:

```
p: Paste after the cursor
P: Paste before the cursor
```

POST 3

PROBLEM: UPDATING OF CONTENT IN THE LOCAL REPOSITORY TO THE REPOSITORY ON GITHUB

I encountered a challenge with laptop when trying to add/update text on github from my local repostory ussing the terminal.

SOLUTION

In summary, I discovered that I had to first create a public key locally on my machine and then add it to among the keys on github so that I can access to add, edit/update and push my content to github

STEP 1 (How I created a public key)

To create a public key using the terminal, you can use the ssh-keygen command. Here are the steps:

Open your terminal.

Use the following command to generate a new SSH key pair. You will be prompted to provide a location for the key and an optional passphrase.

```
bash
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Explanation of options:

```
-t rsa: Specifies the type of key to create (RSA).
-b 4096: Specifies the number of bits in the key (4096 bits is
recommended for stronger security).
-C "your_email@example.com": Adds a label/comment to the key, typically
your email address.
```

Press Enter to accept the default file location (usually ~/.ssh/id_rsa) or provide a custom path.

If you want to add a passphrase for extra security, enter a passphrase when prompted. You can also leave it blank if you don't want a passphrase.

Note: Even if you leave it blank, it's generally a good idea to use a passphrase for added security.

Once the key pair is generated, you'll see a message indicating the location of the public and private keys.

```
vbnet

Your public key has been saved in /path/to/your/home/.ssh/id_rsa.pub.
Your private key has been saved in /path/to/your/home/.ssh/id_rsa.
```

Now you have successfully created an SSH key pair. The public key is stored in the file with the .pub extension (e.g., id_rsa.pub). You can share this public key with services like GitHub, GitLab, or others where you need to authenticate using SSH keys. The private key (id_rsa) should be kept secure and not shared.

To display the content of your public key, you can use the following command:

```
bash

cat ~/.ssh/id_rsa.pub
```

Copy the output and use it wherever you need to add your public key.

STEP 2 (In this step, I got a copy of the repostory onto my computer by carrying ou git clone)

To clone a repository from GitHub (or any Git repository) using the terminal, follow these steps:

Open your terminal.

Navigate to the directory where you want to clone the repository. For example:

```
bash

cd /path/to/destination/directory
```

Get the clone URL from the GitHub repository, but it is better to use SSH URL

If you are using SSH, you can use the SSH URL:

```
bash

git clone git@github.com:username/repository.git
```

STEP 3 (This is a step where we now send/update our content to the github account)

To send content from a cloned repository to GitHub, you'll typically follow these steps:

Navigate to the Local Repository:
Open your terminal and navigate to the local directory of the cloned Git repository.

```
bash

cd /path/to/cloned/repository
Check the Status:
```

It's a good practice to check the status of your local repository to see which files have been modified, added, or deleted.

```
bash

git status
Add and Commit Changes:
```

If you've made changes to your files, add them to the staging area and commit the changes:

```
bash

git add .
git commit -m "Your commit message here"
Push Changes to GitHub:
```

Finally, push your changes to the GitHub repository. If you cloned the repository using HTTPS, you might be prompted to enter your GitHub credentials.

```
bash

git push origin master
```

Then you are able to find your content on git hub

POST 4

PROBLEM: ENGAGING OTHERS DEVELOPERS TO MAKE SUGGESTION AND ADVICE OVER MY CONTENT THROUGH PULL REQUESTS AND FORKS

```
This is my last post, it is not by a mistake because indeed it has not
been that simple for me to establish pull requests so that other
developers can engage ange comment about my content,\
l indeed though its just a simple thing but the best part of the story
is that i was finnaly able to fix it.
```

SOLUTION

```
In my research, l found this information helpful as it helped me come up
with all the process of make pull requests to contribute to my fellow
developers' content
Creating a pull request on GitHub is a collaborative way to propose
changes to a codebase. It allows other developers to review your changes
and provide feedback before merging them into the main branch of the
repository. Here's a step-by-step guide on how to create a pull request
on GitHub:
```

STEP 1: FORK THE REPOSITORY:

```
If you're contributing to an existing project, you'll need to fork the
repository first. Forking creates a copy of the repository under your
own account, allowing you to work on your changes without affecting the
original repository.

Make your changes: Clone your forked repository to your local machine
and make the changes you want to contribute. Once you're satisfied with
your changes, commit them and push them to your forked repository.
```

STEP 2: CREATING THE PULL REQUEST:

```
Navigate to the repository: Open GitHub and go to the main page of the
repository you want to contribute to.
```

STEP 3: SELECT THE BRANCH:

```
On the repository page, in the "Branches" section, choose the branch
containing your commits. This is typically the branch you worked on
(e.g., "feature-branch").

Initiate the pull request: Above the file list, in the yellow banner,
click the "Compare & pull request" button.
```

STEP 4:SELECT BASE AND COMPARE BRANCHES:

```
In the pull request creation dialog, choose the "base branch" as the
main branch of the original repository where you want to merge your
changes (e.g., "master"). Then, select the "compare branch" as the
branch containing your commits.

Provide a title and description: Enter a clear and concise title for
your pull request, summarizing the purpose of your changes. In the
description, provide a more detailed explanation of your changes, why
they were made, and any relevant context.
```

STEP 5: CREATING THE PULL REQUEST:

Once you're satisfied with the title, description, and diff review, click the "Create pull request" button

I will summarize the POTENTIAL ADVANTAGES AND DISADVANTAGES, COST- BENEFIT AND ANALYSIS in the paragraph below

First, the advantages is already elaborated in the solution of each problem mentioned and the disadvantages could rise from applying a wrong command to a wrong operating system forexample applying a linux command to a mac machine.
For cost benefit and analysis is openly shown in the efficiency of solutions where a business personel isable to create his/her own content on his own without requiring a third party hence being cost effective.