A red and black logo

Description automatically generatedA red and black logo

Description automatically generated**Standard Operating Procedure for Annual Leave & Final Leave Payment Calculator**

|  |  |
| --- | --- |
| **Procedure Name:** | Annual Leave & Final Leave Payment Calculator |
| **Issued on:** | 07/04/2025 |
| **Issued by:** | Jayden Leung |

**Table of Contents:**

[Methodology & Procedure](#_Methodology_&_Procedure) … … … … … … … … … … … … … … … … … … … … … … … 1-16

[Troubleshooting & Tips](#_Troubleshooting_&_Tips) … … … … … … … … … … … … … … … … … … … … … … … … 16-18

[References & Glossary](#_References_&_Glossary) … … … … … … … … … … … … … … … … … … … … … … … … … 18

# **Methodology & Procedure**

1. GitHub Setup
   1. Creating a GitHub Account

* Navigate to <https://github.com/>
* Click “Sign Up”
* Create a GitHub account

A screenshot of a computer screen

AI-generated content may be incorrect.

* 1. Downloading Git
* Navigate to <https://git-scm.com/downloads/win>
* Under “Standalone Installer”, click on “Git for Windows/x64 Setup”

A screenshot of a computer

AI-generated content may be incorrect.

* Run the Git Setup Wizard, and click “Next” on each stage (use default Git recommended settings)
* After all steps are completed, click “Install”
* Click “Finish” to complete the Setup Wizard

[OPTIONAL]

* 1. Configuring Your Git Identity
* Navigate to <https://git-scm.com/book/en/v2>
* Click on “1.6 First-Time Git Setup” or click <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>
* Under the “Your Identity” section, copy the first line of code (excluding the $ sign) and paste it into Command Prompt

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Hit “Enter”
* Copy the second line of code and paste it into Command Prompt



A screenshot of a computer screen

AI-generated content may be incorrect.

* Hit “Enter”
* To ensure you’ve done it correctly, run “git config --list” and hit “Enter” (it should display the correct name and email)

A screen shot of a computer screen

AI-generated content may be incorrect.

1. GitHub Repository
   1. Creating a GitHub Repository

* On the Dashboard or Home page, click the “+” button and select “New repository”

A screenshot of a computer

AI-generated content may be incorrect.

* Choose a repository name (see here for repository naming conventions: <https://medium.com/@nur26691/repository-naming-conventions-1065467de776>)
  + i.e. annual-leave-final-leave-payment-calculator
* Select “public”
* Leave the “Add a README file” box unchecked
* A screenshot of a computer

  AI-generated content may be incorrect.Click “Create repository”

* 1. Pushing Code to Your GitHub Repository
* Ensure your three code files are in the proper folder on your computer
  + index.html
  + styles.css
  + script.js

A screenshot of a computer

AI-generated content may be incorrect.

* Open Command Prompt
* Ensure you are in the correct folder/directory (the folder that contains the code files) by typing in the command: “cd [folder name]”

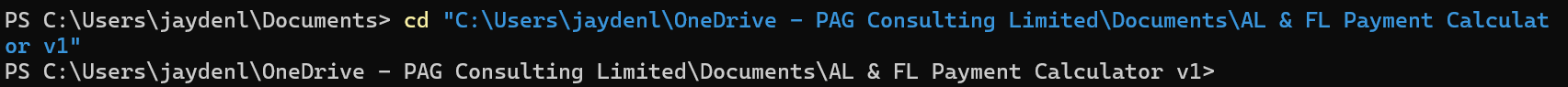
A screenshot of a computer

AI-generated content may be incorrect.

* OR simply copy the folder location address in File Explorer and type “cd [location address]

A screenshot of a computer

AI-generated content may be incorrect.



* Copy the first three lines of code under the “…or create a new repository on the command line” section

A white rectangular object with a black border

AI-generated content may be incorrect.

* Paste into the PowerShell and hit “Enter”
* Type *“git add .”* into the next line and hit “Enter”

A screenshot of a computer

AI-generated content may be incorrect.

* Copy the last four lines of code under the “…or create a new repository on the command line” section

A white rectangular object with a black border

AI-generated content may be incorrect.

* Run these four lines in command prompt

A screenshot of a computer

AI-generated content may be incorrect.

* Press “Enter” again

A computer screen with white text

AI-generated content may be incorrect.

* Navigate back to your GitHub repository and ensure the following files/folder exist in your project folder
  + .github/workflows
  + README.md
  + index.html
  + script.js
  + styles.css

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Hosting a Website Using GitHub Pages (usable link)
* Navigate to Settings -> Pages

A screenshot of a computer

AI-generated content may be incorrect.

* Under “Source”, select “GitHub Action”

A screenshot of a web page

AI-generated content may be incorrect.

* Under “Static HTML”, click the “Configure” button

A screenshot of a computer

AI-generated content may be incorrect.

* Click the green “Commit changes” button twice

A screenshot of a computer

AI-generated content may be incorrect.

* Under the “Actions” tab, wait till the “Create static.yml” action is completed (indicator: green checkmark)

A screenshot of a computer

AI-generated content may be incorrect.

* END RESULT: You should have a live website link in GitHub Pages

A close-up of a computer screen

AI-generated content may be incorrect.

* 1. Pushing New Code to GitHub Repo
* If you made any changes to the code files in Cursor AI, then you need to apply the changes to the code files in your GitHub repository
* Go to Code -> click on one of the code files you’d like to change
* Click the pencil icon to edit this code file

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.

* Copy the changed code from Cursor AI into the code editor in GitHub and press “Commit changes” to make the change

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# **Troubleshooting & Tips**

**Troubleshooting & Tips**

**GitHub Setup and Account Creation**

* **Issue: Unable to Sign Up**
  + Ensure a stable internet connection. Clear browser cache or try a different browser if the "Sign Up" button doesn’t work.
* **Tip: Password Security**
  + Use a strong, unique password and enable two-factor authentication (2FA) for added security on your GitHub account.

**Downloading and Installing Git**

* **Issue: Setup Wizard Fails**
  + Verify you have administrative privileges on your computer. If the installation stalls, download the latest version from the Git website and run it again.
* **Tip: Check Installation**
  + After installation, open Command Prompt and type git --version to confirm Git is installed correctly.

**Configuring Git Identity**

* **Issue: git config --list Shows No Output**
  + Ensure you entered the commands (git config --global user.name "Your Name" and git config --global user.email "your.email@example.com") correctly without extra spaces. Re-run them if needed.
* **Tip: Verify Identity**
  + Use git config --global --get user.name and git config --global --get user.email to check specific settings individually.

**Creating and Pushing to GitHub Repository**

* **Issue: src refspec main does not match any Error**
  + This occurs if the main branch doesn’t exist. Initialize the repository with git init and create the main branch with git branch -M main before pushing.
* **Tip: Verify Folder Contents**
  + Before running git add ., ensure index.html, styles.css, and script.js are in the correct directory using dir in Command Prompt.

**Hosting with GitHub Pages**

* **Issue: No Live Website Link Appears**
  + Check that the "GitHub Action" is set correctly under Settings > Pages. Ensure the static.yml action completed successfully (green checkmark).
* **Tip: Test Locally First**
  + Use a local server (e.g., with Visual Studio Code’s Live Server extension) to test your HTML, CSS, and JS files before deploying.

**Pushing New Code**

* **Issue: Changes Not Reflecting on GitHub**
  + After editing in GitHub, ensure you commit the changes by clicking "Commit changes" and verify the files update in the repository.
* **Tip: Use Pull Before Push**
  + Run git pull origin main before git push to avoid conflicts if the repository was updated elsewhere.

**Tips for Using Cursor AI to Make Changes to Code**

* **Clear Instructions**: Provide specific, detailed prompts to Cursor AI (e.g., "Add a function to calculate annual leave balance" or "Fix the CSS for button alignment"). Vague requests may lead to irrelevant changes.
* **Review Generated Code**: Always inspect the AI-generated code for accuracy and security, as it may contain errors or inefficiencies. Test it in a local environment before committing.
* **Iterate Gradually**: Make small, incremental changes and test after each modification. Use version control (e.g., git commit) to revert if needed.
* **Leverage Context**: Share the relevant file or code snippet with Cursor AI to ensure it understands the existing structure. For example, paste the index.html content first.
* **Debug with AI**: Ask Cursor AI to explain errors or suggest fixes if your code doesn’t work as expected (e.g., "Why is my JavaScript function returning undefined?").
* **Save Changes Locally**: Before pushing to GitHub, save AI-modified files locally and compare them with the original using a diff tool (e.g., VS Code’s built-in diff).

**Overview of Each File**

* **index.html**:
  + **Purpose**: The main structure of the webpage, written in HTML (HyperText Markup Language). It defines the layout, including headings, paragraphs, buttons, and forms for the payment calculator.
  + **Typical Content**: Contains the <html>, <head> (for metadata and links to CSS/JS), and <body> (for visible content like the calculator interface).
  + **Example**: Might include a form with input fields for leave days and salary, and a button to trigger calculations.
* **styles.css**:
  + **Purpose**: Controls the visual appearance using CSS (Cascading Style Sheets). It styles the HTML elements, such as colors, fonts, spacing, and layout.
  + **Typical Content**: Defines classes or IDs (e.g., .calculator, #result) to style the calculator’s design, ensuring it’s user-friendly and responsive.
  + **Example**: Could set a blue background, align inputs centrally, or adjust button sizes.
* **script.js**:
  + **Purpose**: Handles the interactive logic using JavaScript. It processes user inputs (e.g., leave days, salary) and performs calculations for annual and final leave payments.
  + **Typical Content**: Includes functions to validate input, compute payments (e.g., based on a formula), and update the HTML with results.
  + **Example**: Might have a function like calculatePayment(days, salary) that returns the payment amount and displays it on the page.

# **References & Glossary**

**Push Code to your GitHub Account**

<https://www.youtube.com/watch?v=vpRkAoCqX3o>

**How to Host a Website on GitHub Pages**

<https://www.youtube.com/watch?v=e5AwNU3Y2es&t=16s>

**Repository Naming Conventions**

<https://medium.com/@nur26691/repository-naming-conventions-1065467de776>

**Download Cursor AI**

<https://cursor.com/cn>

**Git Download**

<https://git-scm.com/downloads/win>

**Git Documentation**

<https://git-scm.com/doc>