./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
|  |  | Yadala Venkata Sravan Kumar |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc55470819)

[Miniproject -1 [Team/Individual] 4](#_Toc55470820)

[Module/s 4](#_Toc55470821)

[Topic and Subtopics 4](#_Toc55470822)

[Objectives & Requirements 4](#_Toc55470823)

[Design 4](#_Toc55470824)

[Test Plan 4](#_Toc55470825)

[Implementation Summary 4](#_Toc55470826)

[Video Summary 4](#_Toc55470827)

[Git Link 4](#_Toc55470828)

[Git Dashboard 4](#_Toc55470829)

[Summary 4](#_Toc55470830)

[Individual Contribution & Highlights 5](#_Toc55470831)

[Summary 5](#_Toc55470832)

[Challenges faced and how were they overcome 5](#_Toc55470833)

[Future Scope (If applicable) 5](#_Toc55470834)

[Miniproject -2 [Team/Individual] 6](#_Toc55470835)

[Module/s 6](#_Toc55470836)

[Topic and Subtopics 6](#_Toc55470837)

[Objectives & Requirements 6](#_Toc55470838)

[Design 6](#_Toc55470839)

[Test Plan 6](#_Toc55470840)

[Implementation Summary 6](#_Toc55470841)

[Git Link 6](#_Toc55470842)

[Git Dashboard 6](#_Toc55470843)

[Summary 6](#_Toc55470844)

[Individual Contribution & Highlights 6](#_Toc55470845)

[Summary 6](#_Toc55470846)

[Challenges faced and how were they overcome 6](#_Toc55470847)

## **MINI PROJECT -1: TEAM – SDLC**

1. **Module-** modules used in this are SDLC and C-Programming

* **Topics and sub topics**
* Designing the Mini calculator in a way that it should be best while comparing with the presently available calculators in terms of cost, features and also in terms of efficiency and speed also it should be useful from students to engineers
* Designing the calculator in the method of SDLC by best possible model we have and implementing new features and testing it

1. **Objectives and requirements**
   1. **Pros and Cons**

## Pros

* + - 1. More operations possible.
      2. Efficient
      3. User friendly
      4. LCD display
    - Cons
      1. High cost
      2. Need to have some knowledge for operating calculators.

**SWOT ANALYSIS:**

**STRENGHTS:**

* It is very user friendly since it has limited functions.
* Dual power (battery and solar)
* High speed calculations
* High resolution LCD

**WEAKNESS:**

* High cost
* Not a water proof
* High maintenance cost

**OPPURTUNITIES:**

* The product will do very well in shop counters as the calculator is very user friendly and anybody can use it without prior knowledge of that calculator.
* Students up to class 10 will be attracted to this product because of its simplicity in operations and design.
* The product will also work in banking sectors and other government sectors where they want low price, minimum features, handy products

**THREATS:**

* Can be misused by students by over using from lower classes

**4W’S AND 1H ANALYSIS:**

WHAT: It is a simple electronic hardware device or software that are capable of performing the simple calculations such as addition, subtraction, multiplication, division, calculating power of number, exponential function, logarithmic function, permutation and combination, trigonometry, inverse-trigonometric functions, factorial of a number, binary to decimal conversion etc.

WHEN:

* Useful during exams, for getting complex calculation in very less time
* Calculation of bills in malls, shops, and Restaurants

WHERE:

* Exam hall
* Shop Counters
* Colleges and schools, Banking sectors.

HOW:

* Write the code for all the requirements.
* Write the code for all the requirements
* Saves human power
* We save our valuable time by using the

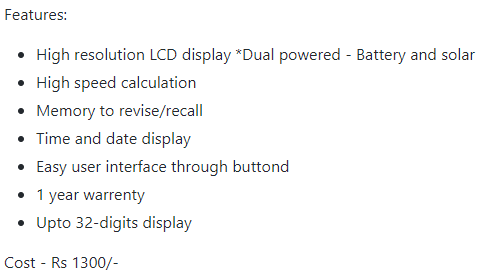
**HIGH LEVEL REQUIREMENT:**

|  |  |  |
| --- | --- | --- |
| **ID** | **DESCRIPTION** | **STATUS** |
| HLR\_01 | ARITHMETIC | IMPLIMENTED |
| HLR\_02 | REALTION | IMPLIMENTED |
| HLR\_03 | AREA CALCULATION | IMPLIMENTED |
| HLR\_04 | VOLUME CALCULATION | IMPLIMENTED |
| HLR\_05 | BODMAS CALCULATION | FUTURE |
| HLR\_06 | TRIGNOMETRIC CALCULATION | IMPLIMENTED |
| HLR\_07 | LOGICAL CALCULATION | IMPLIMENTED |
| HLR\_08 | EXPONENTIAL | IMPLIMENTED |
| HLR\_10 | FACTORIAL | IMPLIMENTED |
| HLR\_11 | DIFFERENTIATION | FUTURE |
| HLR\_12 | INTEGRATION | FUTURE |

## LOW LEVEL REQUIREMENTS

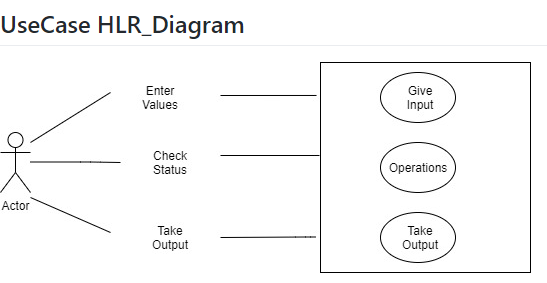
|  |  |  |
| --- | --- | --- |
| ID | DESCRIPTION | STATUS |
| LLR\_01 | IF CHOISE X=3; CHOISE Y =2  CLACULATES “X^Y” | IMPLIMENTED |
| LLR\_02 | IF INPUT =H  CALCULATES H FACTORIAL | IMPLIMENTED |
| LLR\_03 | IF INPUT ENTER VARIABLE  CALCULATES H DIFFERENTIATION | IMPLIMENTED |
| LLR\_04 | IF INPUT = ENTER VARIABLE  CALCULATES INTEGRATION | IMPLIMENTED |
| LLR\_05 | CALCULATES ADDITION, SUBRACTION, MULTIPLICATION, DIVISION, SQUARE ROOT | IMPLIMENTED |
| LLR\_06 | CALCULATES LESS THAN AND GREATER THAN OPERATIONS | IMPLIMENTED |
| LLR\_07 | CALCULATES AREAS FOR CIRCLE, RECTANGLE, SQUARE, TRIANGLE BY TAKING INPUTS LIKE RADIUS, LENGTH AND BREADTH | IMPLIMENTED |
| LLR\_08 | CALCULATES VOLUMES FOR CUBE, SYLINDER, CONE, SPHERE NY TAKING INPUTS LIKE HEIGHT, RADIUS, LENGTH | IMPLIMENTED |
| LLR\_09 | INPUT IS ENTIRE CALCULATED BASED ON BODMAS RULE | FUTURE |
| LLR\_10 | CALUCLATES THE SIN, COS, TAN, COSEC, COT, SEC VALUES BY TAKING INPUT AS ANGLES | IMPLIMENTED |

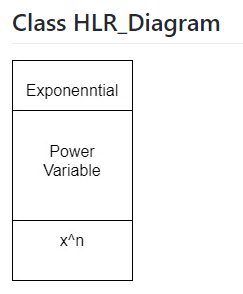
## **FEATURES AND COST**

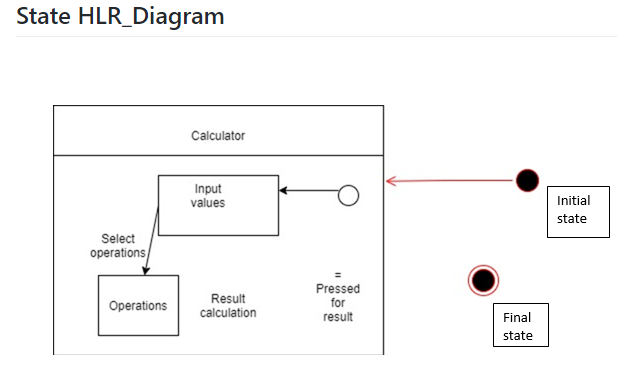


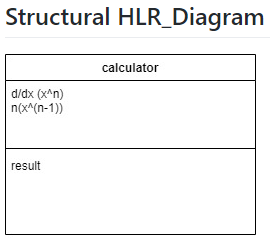
1. **Design**

**3.1 high Level Design**

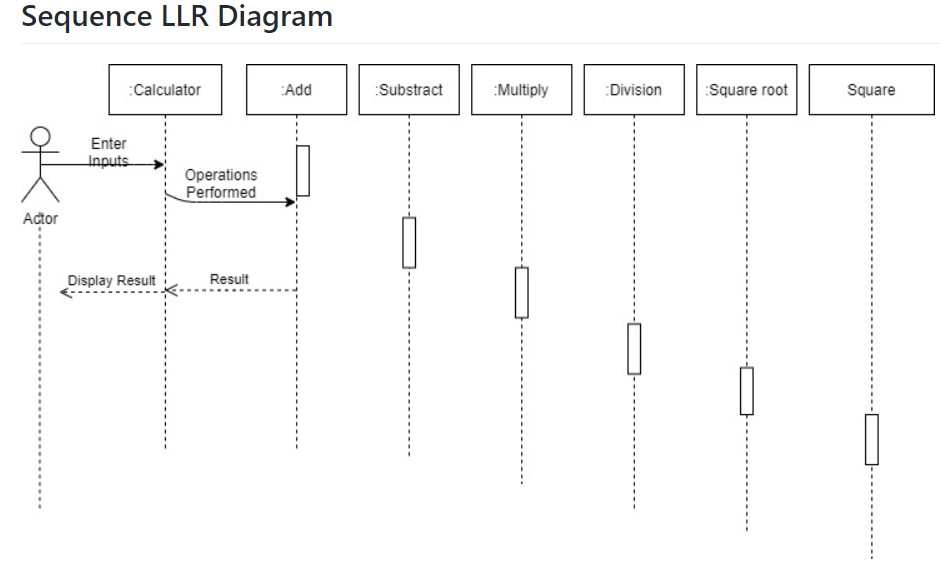
****

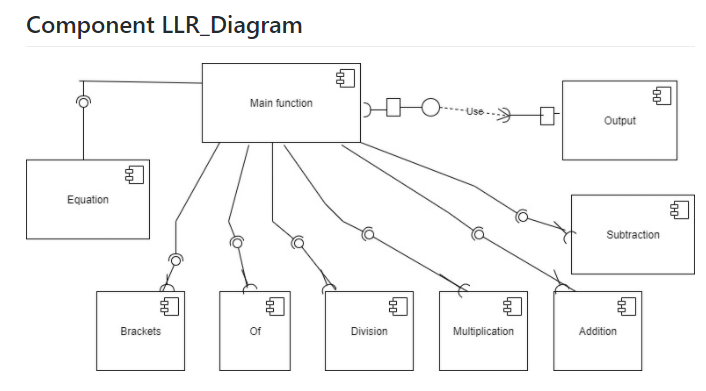
****

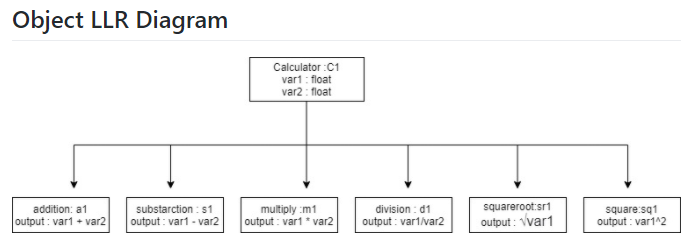
****

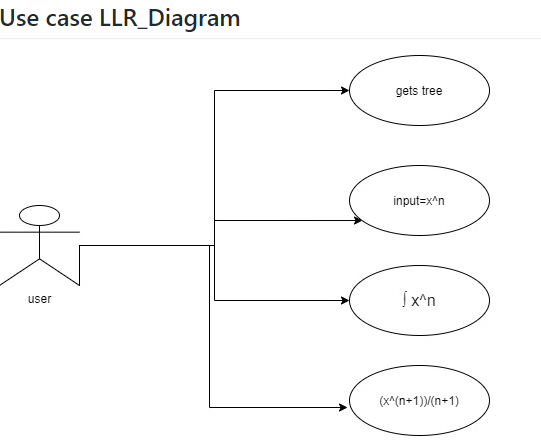
****

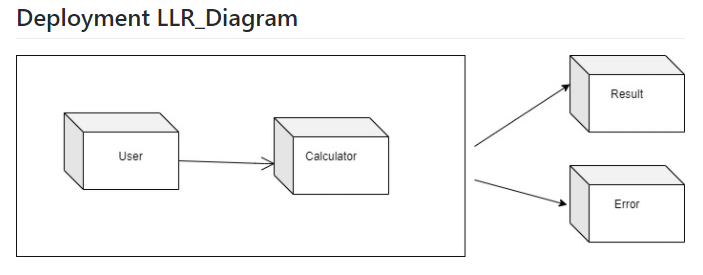
**Low Level Diagrams**

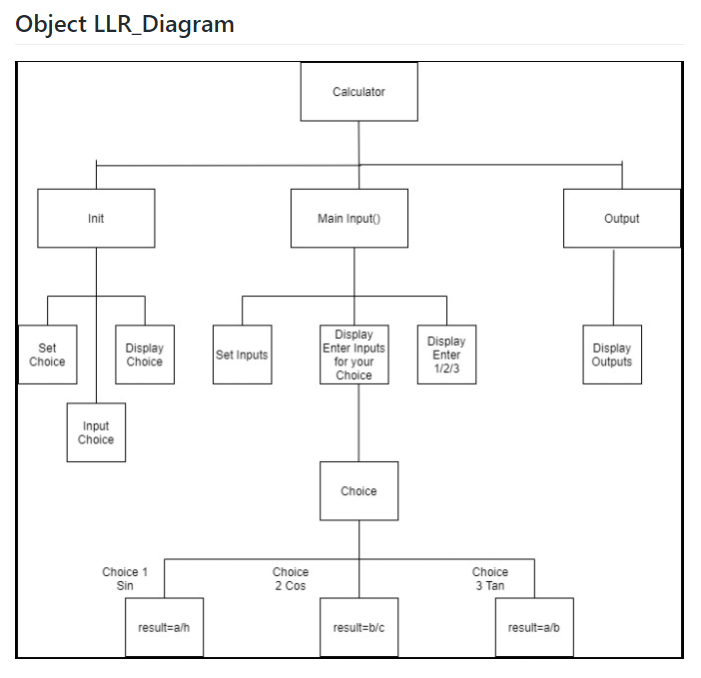
****



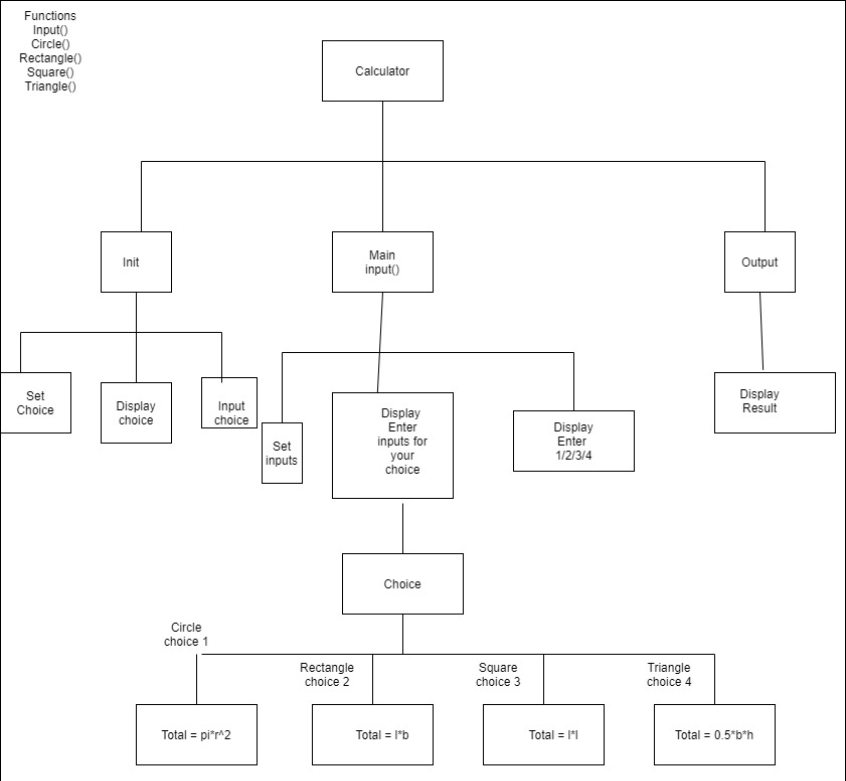








**ACTIVITY LLR\_DIAGRAM**



1. **Test Plan**

9.1 HIGH LEVEL TEST PLAN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TEST ID | DESCRIPTION | EXP IN | EXP OUT | ACTUAL OUT | TYPE OF TEST |
| H\_01 | Performs Arithmetic Operations | N1 = 6;  N2 = 4 | Performs  Arithmetic Operations | Calculates addition, subtraction, multiplication, division, square, square root | Scenario based |
| H\_02 | Performs Relational Operations | N1 =4; n2 = 5 | Performs Relational Operations | Calculates greater than, smaller than, equals to, not equals to | Scenario based |
| H\_03 | Performs Area Operations | 2 | Performs Area Operations | Calculates area of Triangle ,Rectangle, circle, square | Scenario based |
| H\_04 | Performs volume calculations | 3 | Performs volume calculations | Calculates area of cylinder, cube, cone and sphere | Scenario based |
| H\_05 | Performs trigonometric calculations | 4 | Performs trigonometric calculations | Calculates trigonometric calculations based on inputs | Scenario based |
| H\_06 | Performs exponential | 2,3 | Performs exponential | 8 | Scenario based |
| H\_07 | Performs factorial | 3 | Performs factorial | 6 | Scenario based |
| H\_08 | Performs differentiation | X | Performs differentiation | 1 | Scenario based |
| H\_09 | Performs integration | 1 | Performs integration | x | Scenario based |

## **Low level test plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TEST ID | DESCRIPTION | EXP IN | EXP OUT | ACTUAL OUT | TYPE OF TEST |
| L\_01 | Adding two numbers | 5+2 | 7 | 7 | Requirement based |
| L\_02 | Subtract two numbers | 5-3 | 2 | 2 | Requirement based |
| L\_03 | Divide two numbers | 10/2 | 5 | 5 | Requirement based |
| L\_04 | Multiplying two numbers | 2\*3 | 6 | 6 | Requirement based |
| L\_05 | Square of two numbers | 5\*5 | 25 | 25 | Requirement based |
| L\_06 | Greater than or less than | 3>5 | False | False | Requirement based |
| L\_07 | Equals to or not | 1=5 | false | false | Requirement based |
| L\_08 | Not equals to or not | 2!=3 | False | False | Requirement based |
| L\_09 | Square root | Square root (4) | 2 | 2 | Requirement based |
| L\_10 | Calculates area of a square | S1=2; s2=4 | 8 | 8 | Requirement based |
| L\_11 | Calculates volume of the square | A1=4; a2=6 | 24 | 24 | Requirement based |
| L\_12 | Calculates area of a triangle | H=3, b= 4 | 6 | 6 | Requirement based |
| L\_13 | Calculates volume of the square | H=3, b= 4, l=3 | 18 | 18 | Requirement based |
| L\_14 | Calculates area of a circle | R=4 | 50.265 | 50.265 | Requirement based |
| L\_15 | Calculates volume of the cone | B=4, h= 9 | 12 | 12 | Requirement based |
| L\_16 | Calculates area of a rectangle | L=6, w=4 | 24 | 24 | Requirement based |
| L\_17 | Calculates volume of the cube | A=2 | 8 | 8 | Requirement based |
| L\_18 | Calculates exponential | 2^3 | 8 | 8 | Requirement based |
| L\_19 | Calculates factorial | H=3 | 6 | 6 | Requirement based |
| L\_20 | Calculates trigonometric functions  (For sin) | Angle = 0 | 0 | 0 | Requirement based |
| L\_21 | Calculates trigonometric functions  (For cos) | Angle = 0 | 1 | 1 | Requirement based |

## **IMPLIMENTATION SUMMARY:**

* Implementation folder had all source files, header files, test files for different features of the calculator such as Basic Arithmetic, Square root, cube root, exponent, logarithm, etc.
* Here, **inc** folder holds all the header files with “.h” extension which contains prototype of all functions, structure definition, macro definition and definition of all the enumerators.
* The **src** folder holds all the source files with “.c” extension which has definitions of all the functions whose prototype is defined in header files.
* The **test** folder holds the ***test\_calculator\_operations.c*** file for cumulative testing of the source codes based on requirements, scenario and boundary.
* Other than these folders, there is also a **unity** folder which holds prototypes and definition of the standard unity test case functions.
* Also, there is a **Makefile** which builds, debugs using valgrind, check static and dynamic code quality, performs overall unit testing for all the codes together with the execution of single commands based on different defined targets.

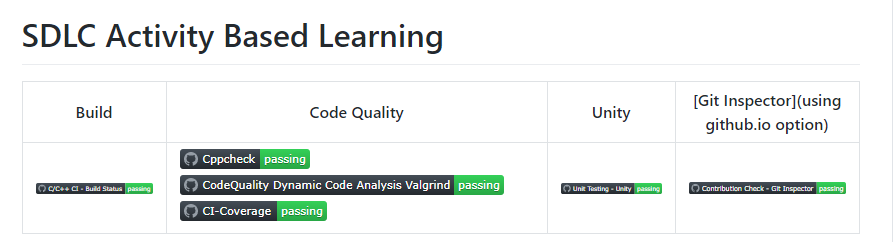
## **VIDEO SUMMARY:**

“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough”

### Git Link:

## https://github.com/99003756/NTEAM5\_SDLC\_CALCULATOR.git

### Git Dashboard:



### Summary

In this project, we mainly-focused on how to design a calculator which is bit different and cost-effective as compared to other calculators present in the market.

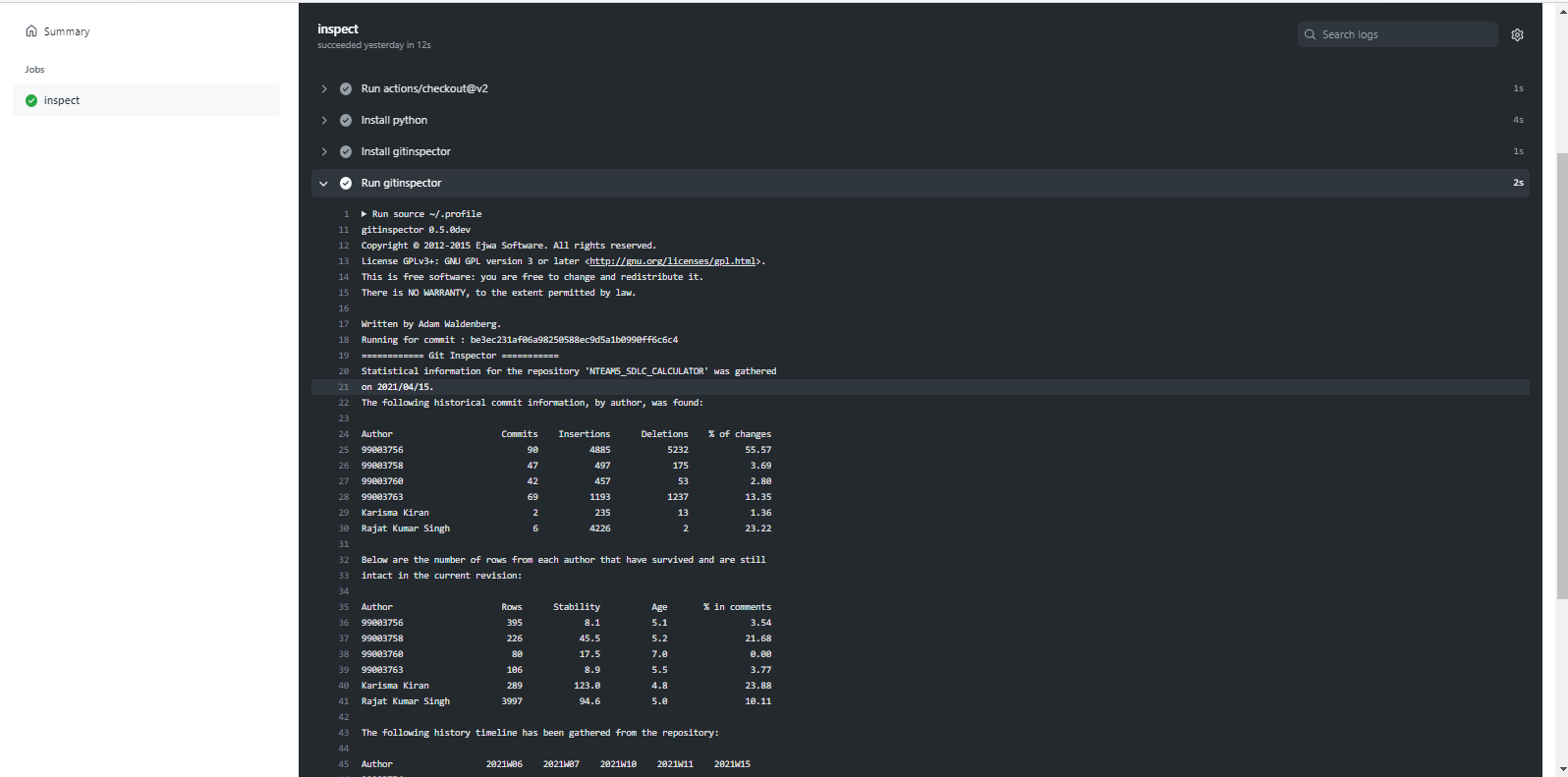
So, for this we first analyzed all the other calculators from low-end feature low cost calculator to high-end feature high-cost calculator and prepared a list of features to include in our modified cost-effective more featured calculator.

Features included in the calculator are basic arithmetic operations (addition, subtraction, multiplication, division), Combinatorics (Permutations and Combinations), Mensuration (Area and Volumes), trigonometric Calculations.

This calculator is implemented through C programming. This calculator will be mainly used by the school, college students, scientists, businessman, engineers for various purpose. Also, this calculator will be cost effective and if implemented over hardware it cost around 1300 INR.

#### Git inspector summary

Git Inspector



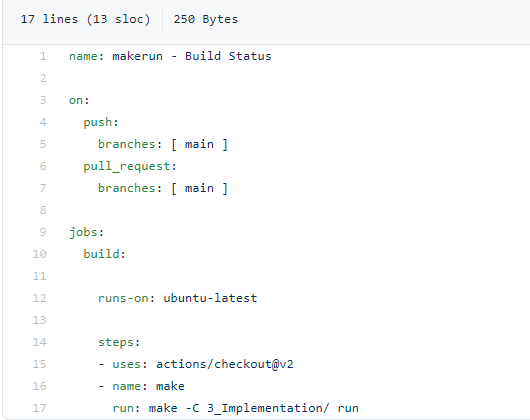
**Setup for Build**



#### Build

#### Build status:

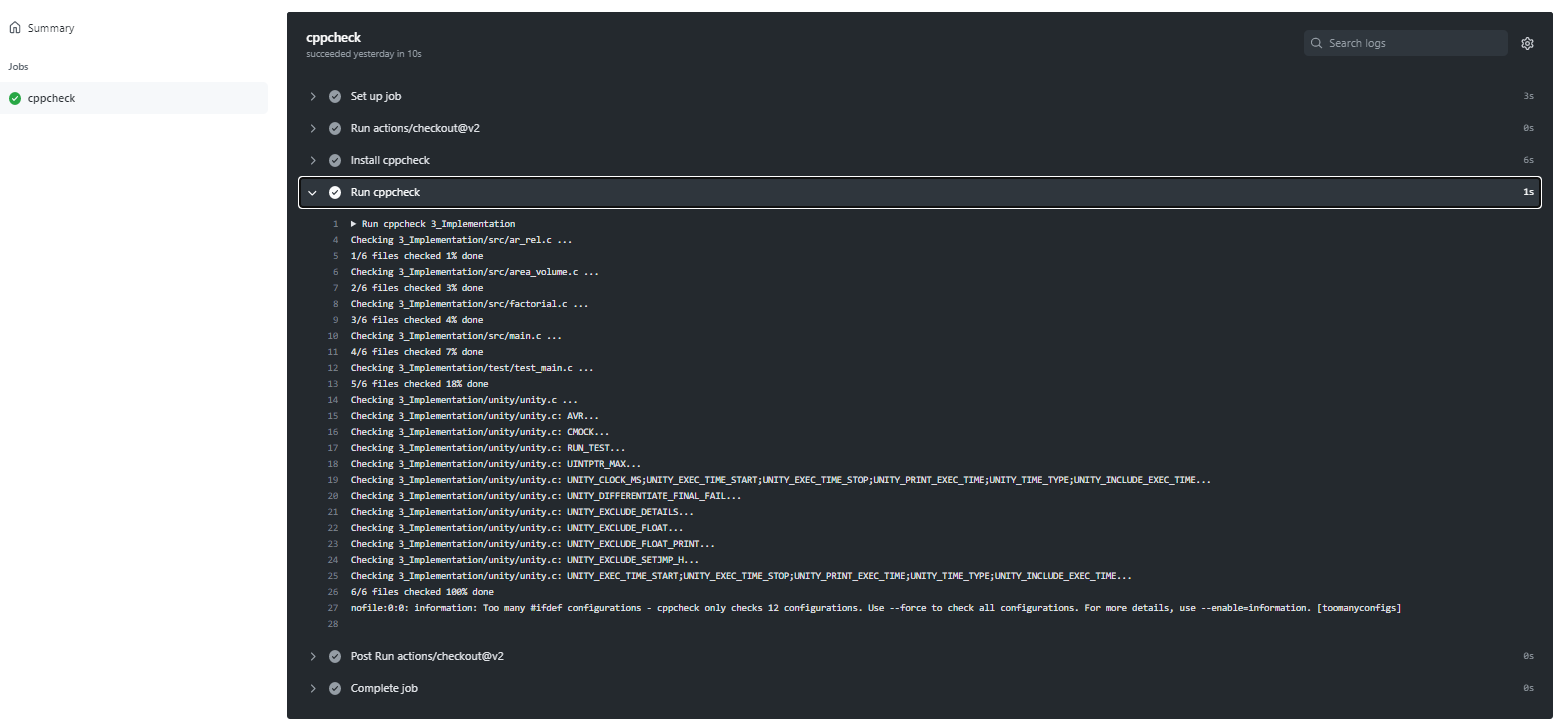
#### 

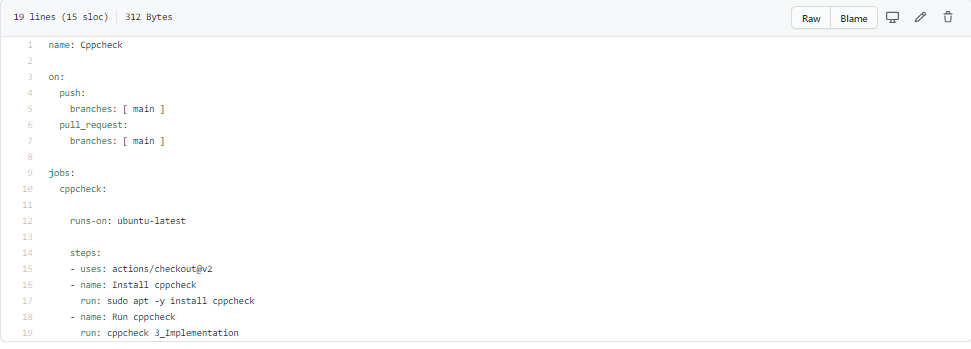


#### Code quality and Issues or Bug Tracking

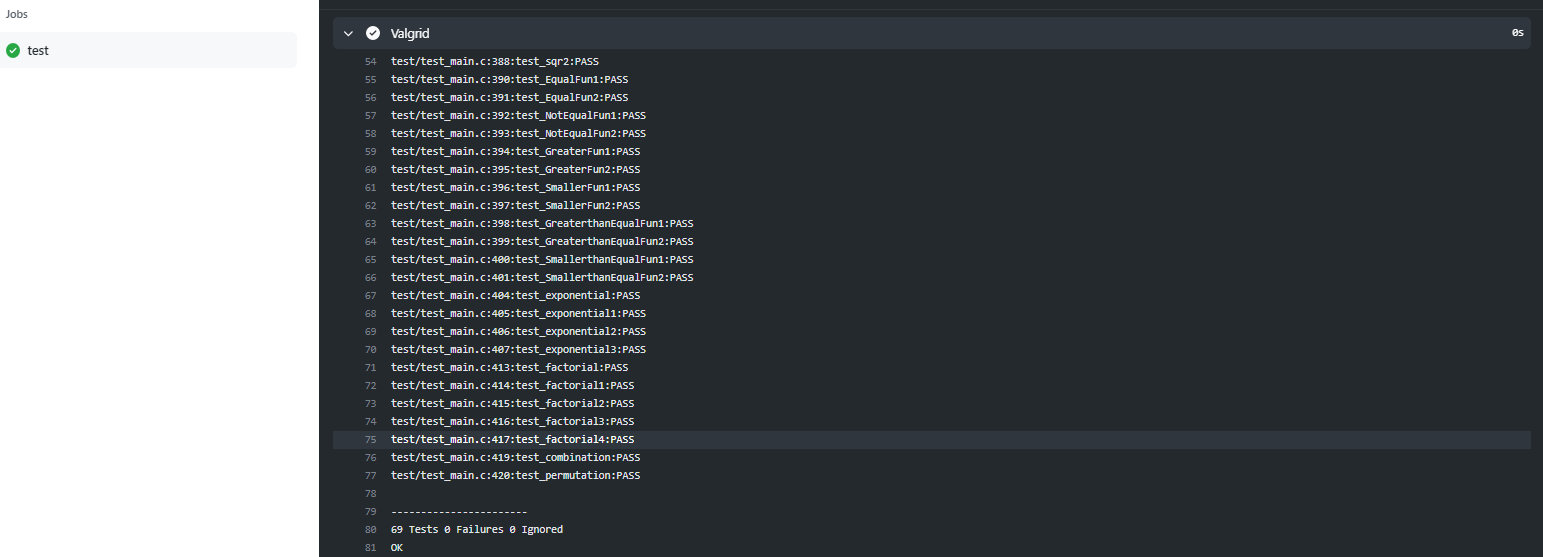
**CODE QUALITY:**

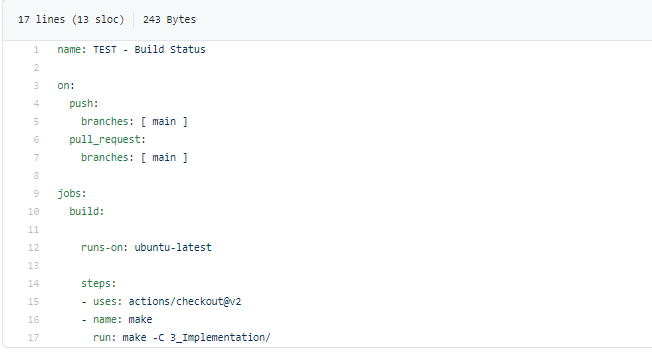
**STATIC CODE QUALITY:**



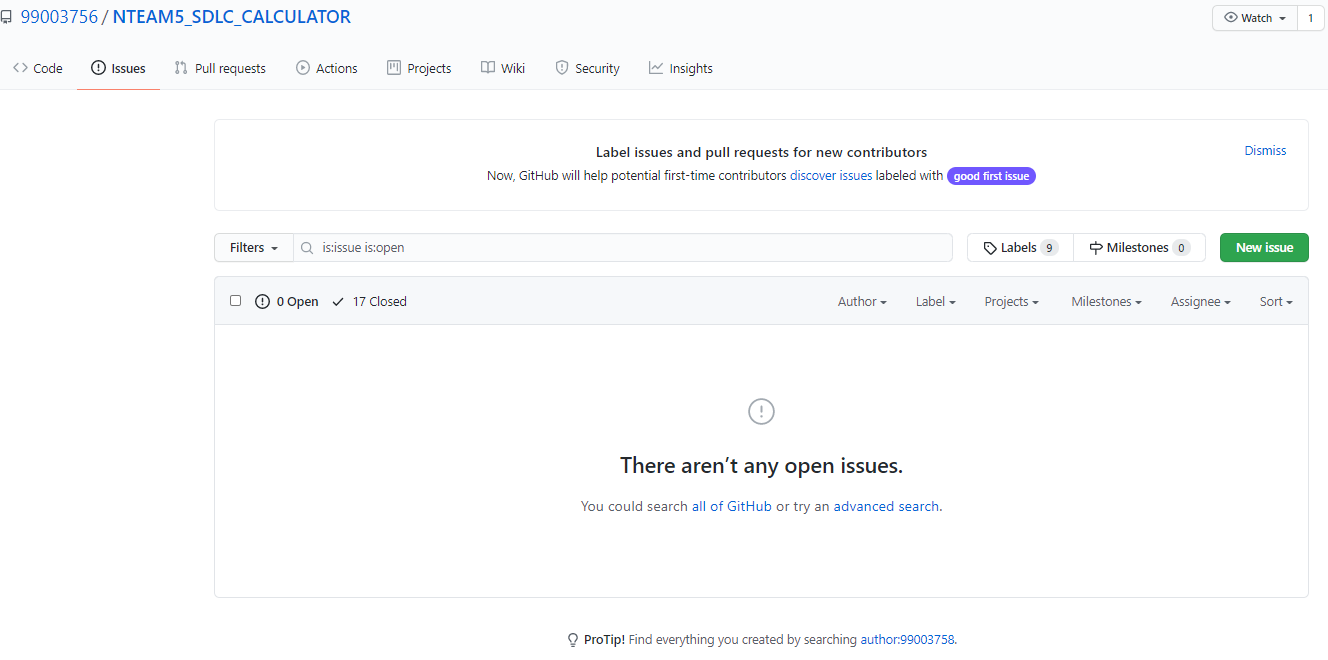


**DYNAMIC CODE QUALITY:**

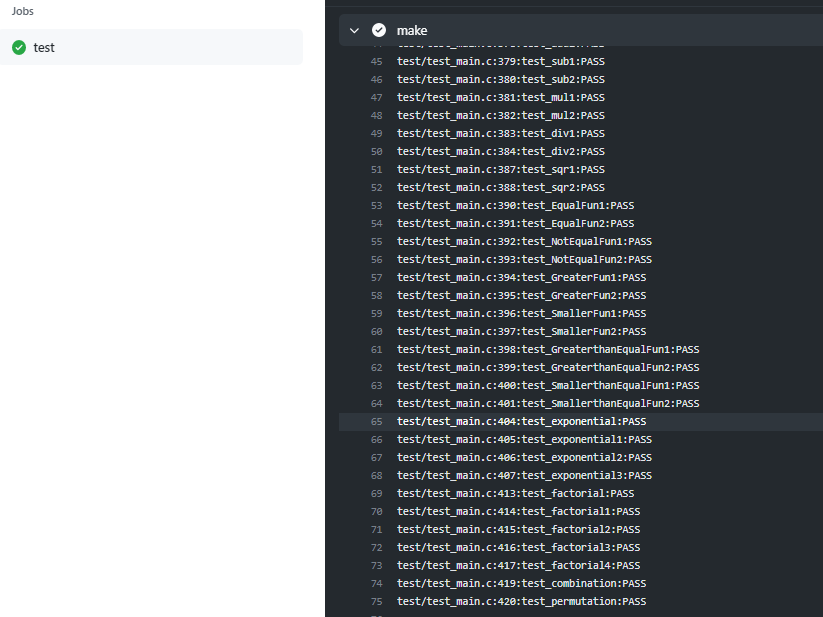
****

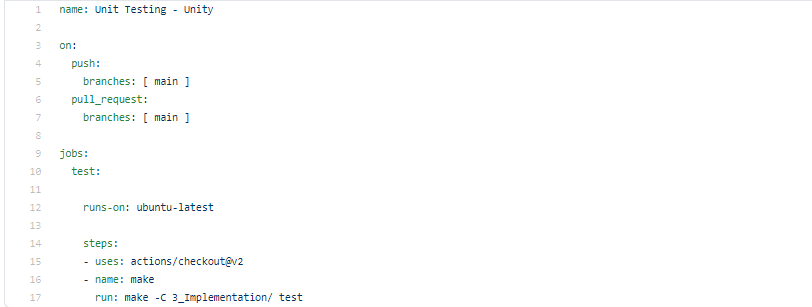
****

**GIT ISSUES:**

****

#### Unit Testing





## Individual Contribution

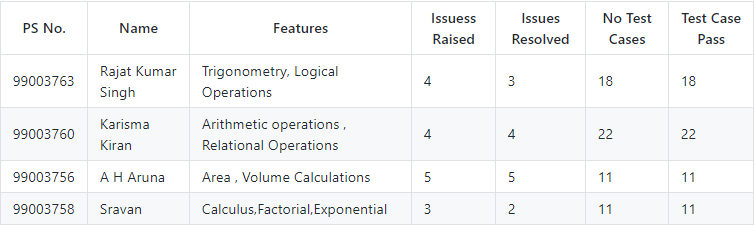
My Contribution to this project is that I have implemented the features of finding FACTORIAL and EXPONENTIAL for given inputs. I have implemented the feature of finding of INTEGRATIONS, DIFFERENTIATION for given inputs.

Also, I have performed the research analysis of different calculators present in market based on features and cost. I have done the SWOT analysis and 4W & 1H analysis in order to understand the pros and cons of our product.

I have written my code in main.c in src folder under implementation. Accordingly I have added main.h in inc folder under implementation. I have modified the test\_main.c where I have written my test cases.

I have modified Makefile so that program should be able to build as well as it is able to perform unity based unit testing.

**Contribution list and summary:**

****

### Summary

The main motto is to design a calculator with certain features according to the specific requirements. The target customers for the designed calculator are students, shopkeepers, banking executives and engineers.

This project was our first step to work in coordination as a team while working towards our development of our individual skills.

Also, we’re unorganized at the beginning but after planning through V-Model we’re able to analyze and design each scenario and perfectly implemented whole project in the given duration. So, we also learned a great skill called time-management.

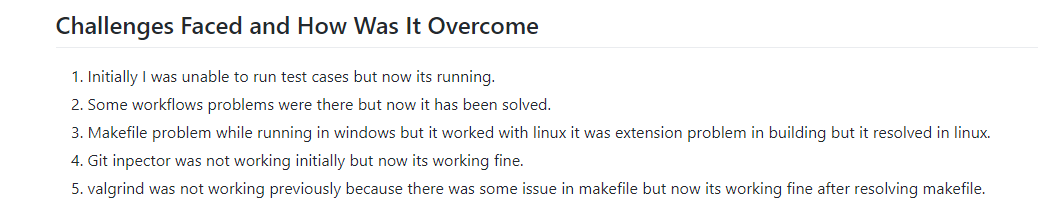
Technical skills developed: -

* Advanced C programming
* Make File
* Unit Testing through Unity
* Version Control using GitHub

Soft-Skills Developed: -

* Team-work
* Team-management
* Time Management
* Assertiveness

### Challenges faced and how were they overcome



### Future Scope (If applicable)

### 

1) Features like matrix operations can be added.

2) Features such as Physics operations (Acceleration, capacitive reactance, circular velocity, Coulomb’s Law, Gravitational force, projectile motion ) can be added.

3) Features like AP, GP sum (AP sum, GP sum) can be implemented in the future.

# Miniproject -2 [Individual] – Python

## **2.1 Module**

Modules used in this project are Core and Advanced Python.

## **2.2 Project title:** Retrieve data from multiple Excel sheet

## **2.3 Topic and Subtopics**

* Openpyxl
  + Read excel file
  + Write excel file
* Barchart
* functions

## **2.4 Objectives:**

## Creating one Excel file with five Sheets, one sheet is the master sheet. Here in 4 Excel sheets some data should be same. By searching on the particular data we can get the total data in the mastersheet.

## **2.5 Requirements**:

### 2.5.1 High Level requirement analysis

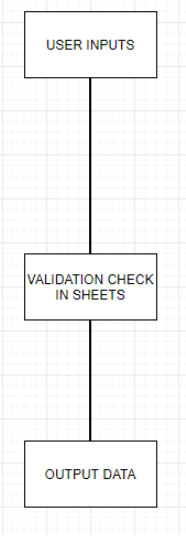
|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Requirements | Description | Status |
| HL1 | Creating | In 5 sheets data of a person is present, one is the master sheet | Implemented |
| HL2 | Combining | All the datasheets are combined in one data sheet | Implemented |
| HL3 | Searching | By using the Ps number of that person, we can get all the information of that person in master sheet | Implemented |

### 2.5.2 Low Level requirement analysis

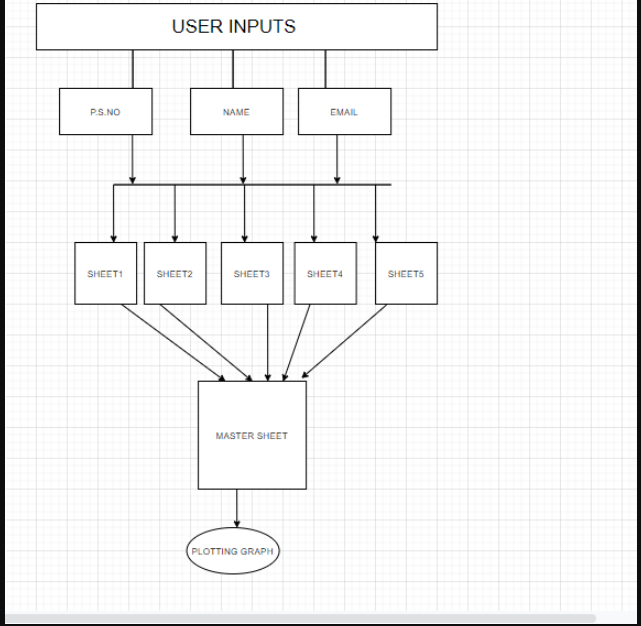
|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Requirements** | **Description** | **Status** |
| LL1 | Creating rows and columns | In each sheet, we have to create 10 columns and 40 rows | Implemented |
| LL2 | Searching | Searching the data, we need in all sheets | Implemented |
| LL3 | Output | As per the requirement we have to display the output and store it | Implemented |

**2.6 Design**

HIGH LEVEL DIAGRAM



LOW LEVEL DIAGRAM



## **2.7 Implementation Summary**

The aim of the project is to extract the data present in different spreadsheets in one excel file as required by the user. The excel sheet consists of 5 spreadsheets. The user defines the data that needs to be searched on the basis of Name, PS Number and Email ID. The python program then reads the data corresponding to the particular data from different spreadsheets of excel. It then creates a master sheet and adds the data from all the sheets to it. In the end, it will plot the bar graph from the data present in the master sheet.

## **2.8** **Video Summary**

“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough

## **2.9 Git Link**

<https://github.com/99003758/mini-project1.git>

## **2.10** **Summary**

### 2.10.1 Outcomes:

Technical:

* Improved implementation of Python concepts.
* Practical implementation of SDLC lifecycle.
* Source code management. (GitHub)

Soft skills:

* Project management
* Conflict management.

## **2.11** Challenges faced and how were they overcome

* System issues(crashing and Interfacing).
* Differentiation of high level and low level.
* Committing to GitHub, pull and push in GitHub.
* Converting pictures & tables into readme

# Miniproject -4 [Individual] – Kernel Programming and Device Drivers

## **4.1 Module/s:**

The modules used in this are Linux and Kernel Device drivers.

## **4.2 Topic and Subtopics:**

* Basic Linux commands.
* Qemu Based Emulation.
* Creation of SD card.
* Building custom Kernel.
* Cross Compilation.
* Static and dynamic libraries.
* System calls.
* Adding system calls in kernel space.
* Invoking system calls from user space.
* Kernel modules.
  + In-Tree modules: Dynamic.
  + In-tree modules: static.
* Basics of Kernel Device Drivers.
* Registering Char Driver.
* Kernel Data Structure.
  + Kfifo API.
  + List API.
* IPC Kernel
  + Concurrency.
    - Kernel Threads.
  + Locking and Synchronization.
    - Mutex.
    - Semaphore.
    - Spinlocks.
    - Wait queues.
* IOCTL.
* Driver model.

## **4.3 Objectives & Requirements:**

The main objective of this module is to apply the concepts of Linux kernel, kernel device drivers to develop:

* Custom kernel.
* Create char drivers.
* Developing cross compiled code for target qemu.
* Creating own system calls.

### 4.3.1 Requirements:

* Basic Linux commands.
* Programming in Linux Environment.
* Custom kernel.
  + zImage
  + vexpress-v2p-ca9.dtb
  + rootfs.img
* Operating system Basics.
* IPC concepts.
* Concurrency.
* File handling using system calls.
* Virtual Memory concept.

## **4.4 Implementation Summary:**

### 4.4.1 Hands-on Activity that are implemented are as follow:

* Register char driver
* Register file operations
* Device Create, Class Create
* Read, write operations using global buffer
* Read, write operations using kfifo.
* ioctl operations, returning length/remaining space, reset operation
* ioctl operations - filling length/remaining space in structure
* synchronization in char driver - using wait queue

### 4.4.2 User space code:

* simple read, write
* multiple read, multiple write
* User space code for IOCTL operations

### 4.4.3 kthread examples:

* simple two threads
* Race condition scenarios
* Mutual exclusion using semaphore, mutex, spinlock
* Synchronization using semaphores, wait queues
* Device Tree based platform driver code -- dummy UART
* Activity that are implemented are as follow:
* System calls -- echo back the given string.
* System calls—traverse process list print pid and ppid.
* System calls—length of string.
* System calls—taking simple parameter.
* IOCTL operation traverse the list.

## **4.5 Git Link:**

<https://github.com/99003758/EMBEDDED-LINUX-KERNEL-PROGRAMMING-.git>

## **4.6 Summary:**

## **4.7 Challenges faced and how were they overcome:**