Flow of performing Regression is like this:

- Identify the targets

- Check the targets

- Build the correct Target

- Run all targets

- Collect all results and log

- Analyse
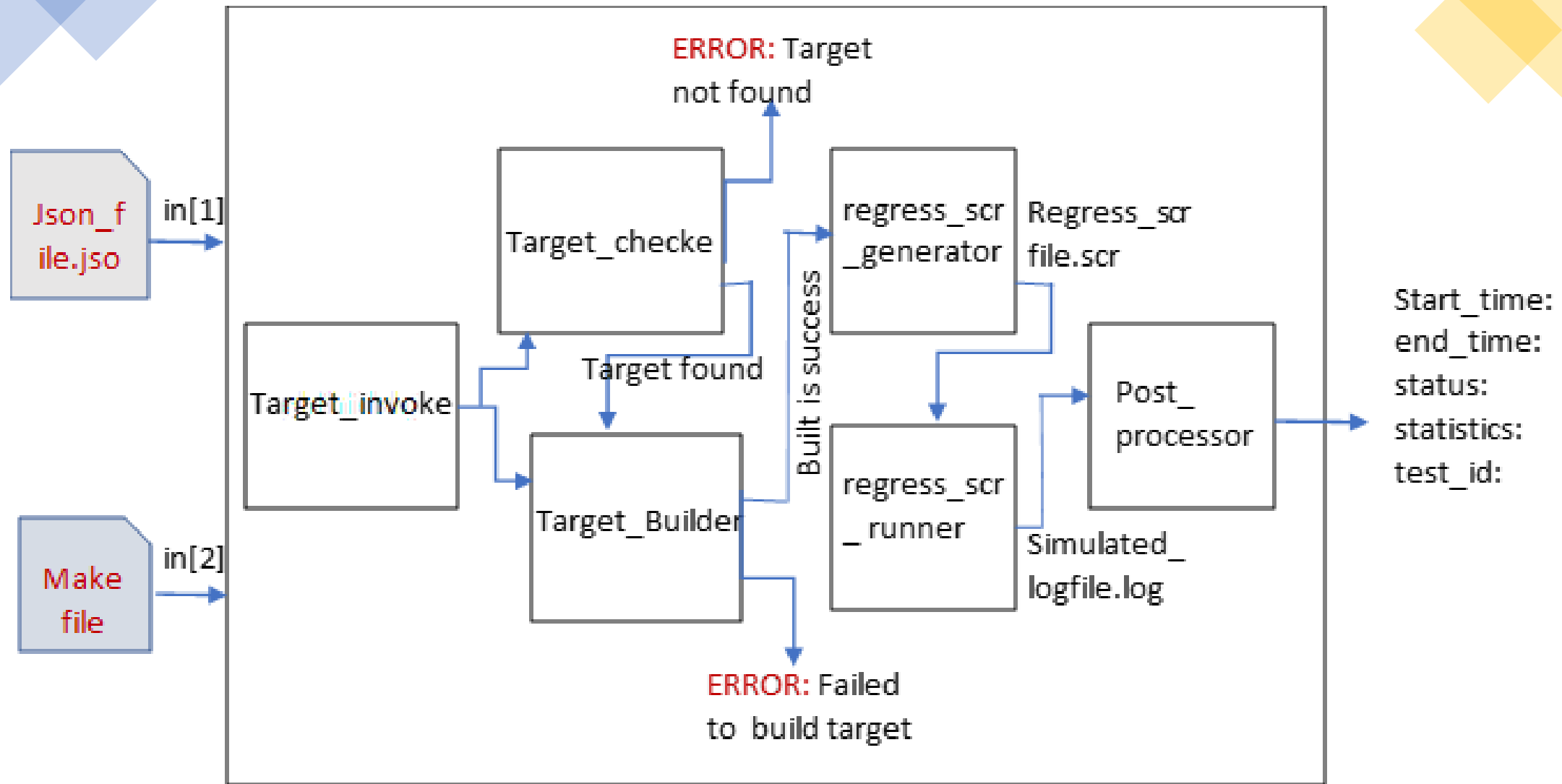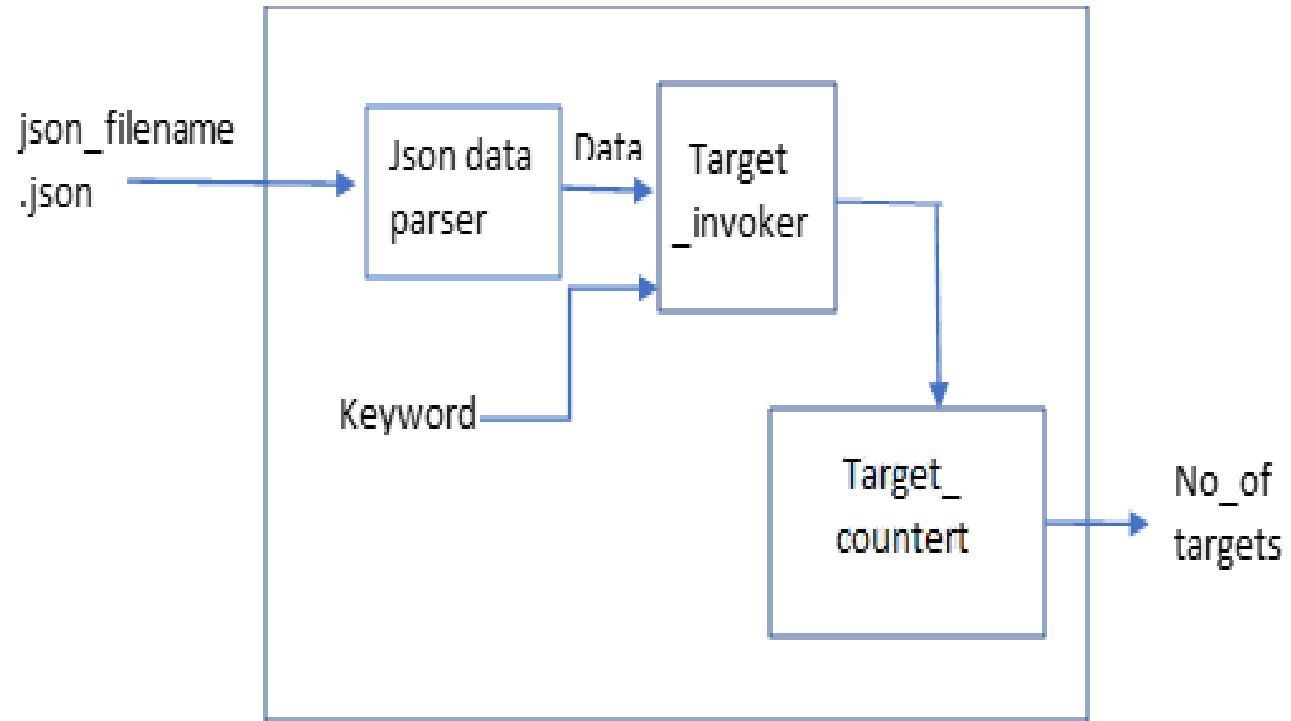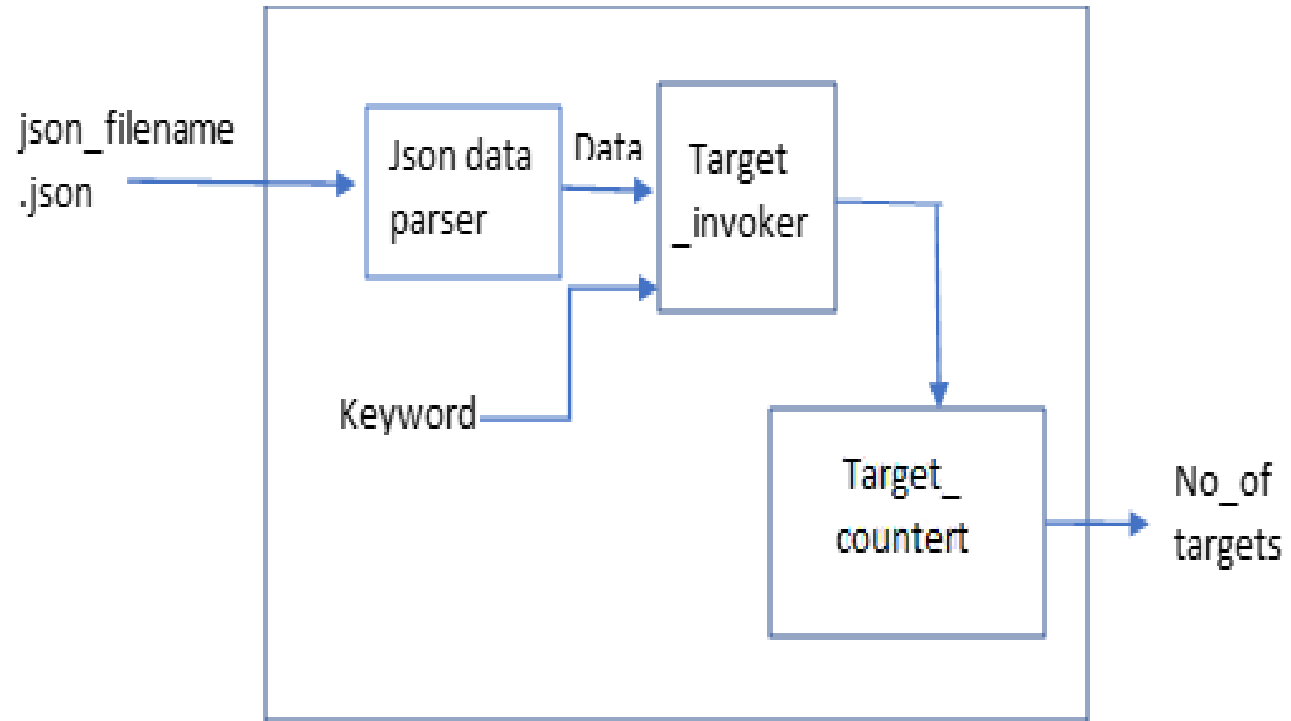
Fig1: System Block Diagram which performs regression

Target Identifier Block:

Target_identifier: Reading the data present in the json_file and counting the number of targets found in the json file by using the keyword "target_name".
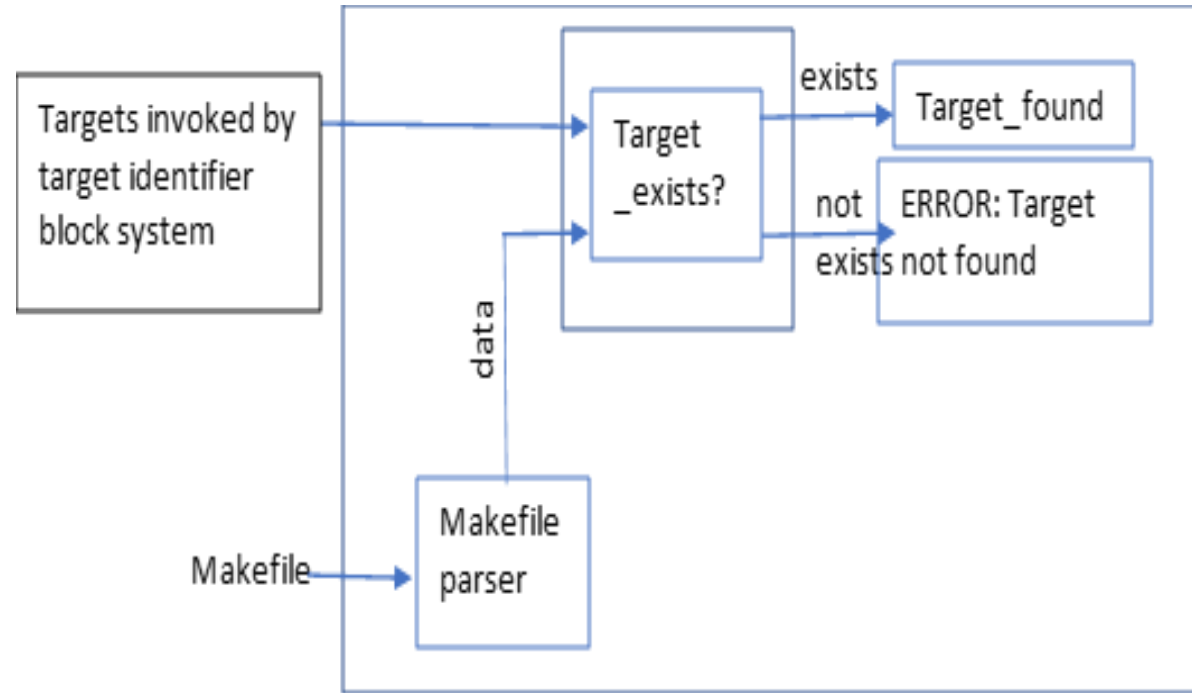
# Target Identifier Block:

Target_identifier: Reading the data present in the json_file and counting the number of targets found in the json file by using the keyword "target_name".

Target Checker Block:

Target_checker: The target identified in the json file must have the same targets in the makefile. Just need to check the targets in the makefile is same as the targets found inside json file.

Target Builder Block:
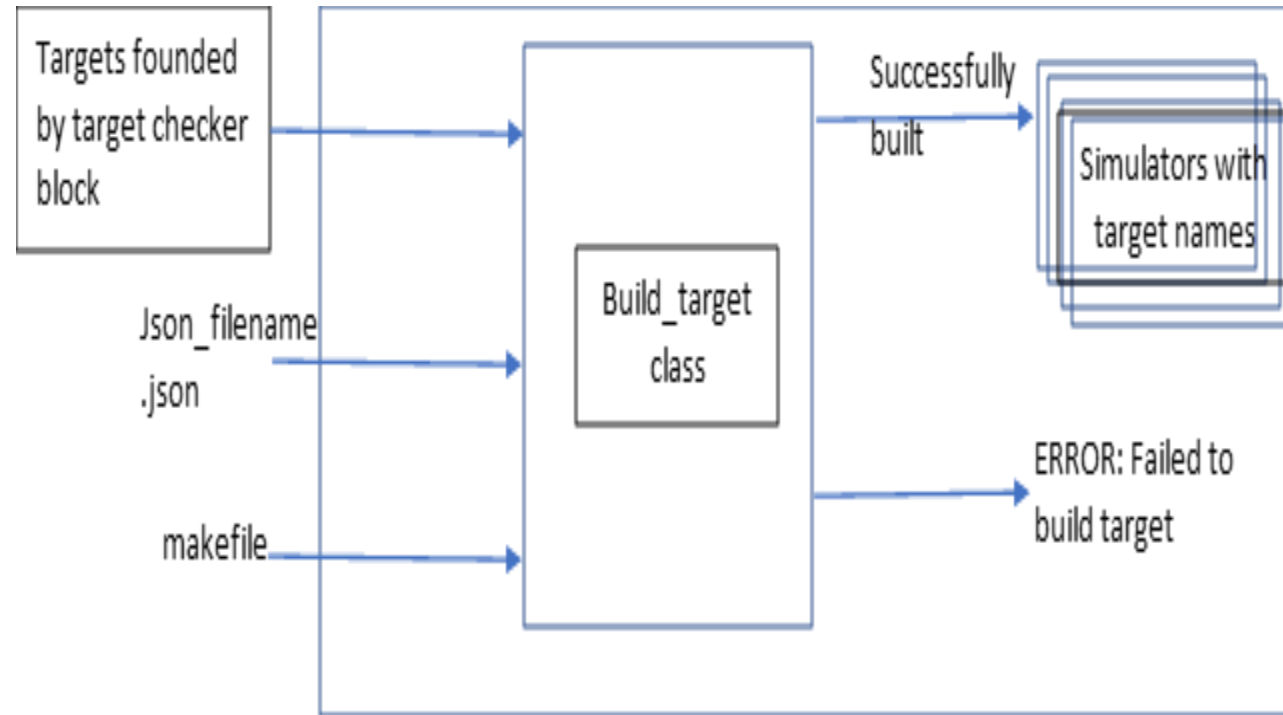
Target Builder: The build target block will obtain the simulated file or the executable file with the name of targets. The target build is successful when there will be a file exists with the same name.

Regress script generator:
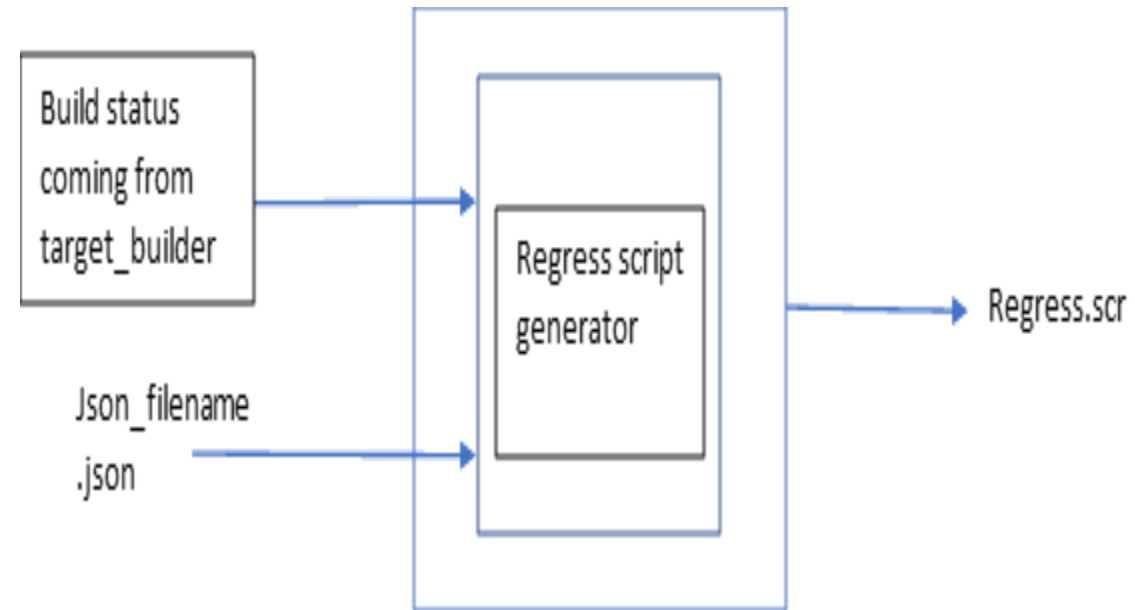
**Regress Script Generator:** This block will invoke each target in the json file by creating and writing in the form of commands to run that each line in the regress.scr file

Regress script runner:
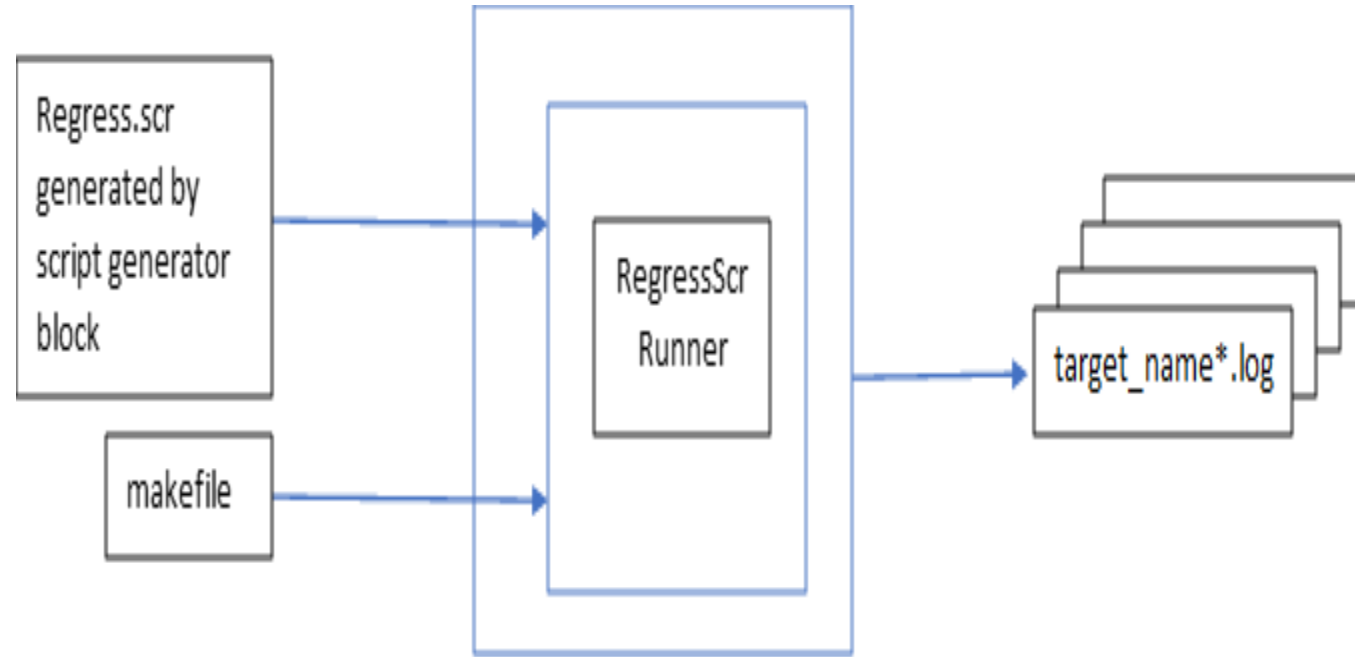
Regress Script Runner: This block will call the makefile with the target name: run_regress_scr, this will automatically run all the commands and each target simulation will outputs a log file with the <target_name.log>

Post Processor:
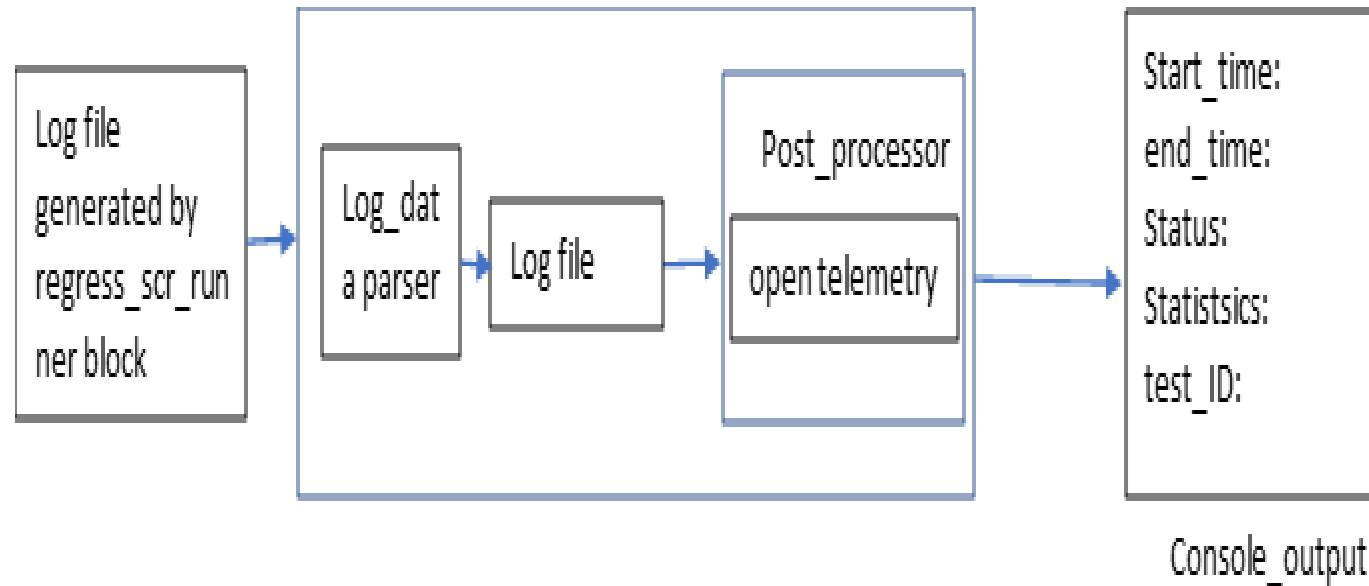
Post Processor: This block will read the log file data and extract the information like start_time, end_time, status, statistics, with respective test_id... using opentelemetry api.

Json file content

```json
{
    "target" : "gen_exe_apb_write_verify_test",
    "test_ID" : "apb_write_verify_test",
    "run_opts" : ["+UVM_TESTNAME="],
    "log_file" : "log/apb_write_verify_test.log "
},
{
    "target" : "gen_exe_apb_base_test",
    "test_ID" : "apb_base_test",
    "run_opts" : ["+UVM_TESTNAME="],
    "log_file" : "log/apb_base_test.log "
},
{
    "target" : "gen_exe_apb_write_verify_error_test",
    "test_ID" : "apb_write_verify_error_test",
    "run_opts" : ["+UVM_TESTNAME="],
    "log_file" : "log/apb_write_verify_error_test.log "
},
{
    "target" : "gen_exe_apb_random_read_write_override_test",
    "test_ID" : "apb_random_read_write_override_test",
    "run_opts" : ["+UVM_TESTNAME="],
    "log_file" : "log/apb_random_read_write_override_test.log "
},
{
    "target" : "module_dirwr_vfile",
    "test_name" : "check_direction",
    "log_sile" : ""
},
```

Generated Regress script content

```
apb_write_verify_test +UVM_TESTNAME=apb_write_verify_test -l log/apb_write_verify_test.log

apb_base_test +UVM_TESTNAME=apb_base_test -l log/apb_base_test.log

apb_write_verify_error_test +UVM_TESTNAME=apb_write_verify_error_test -l log/apb_write_verify_error_test.log

apb_random_read_write_override_test +UVM_TESTNAME=apb_random_read_write_override_test -l log/apb_random_read_write_override_test.log
```

Dashboard

Log Information

| Test ID	Start_time	End_time	Status	Statistics	Error Information
apb_write_verify_error_test	None	None	Fail	Error:16	Functional Failure
apb_write_verify_error_test	Feb 22 12:13 2024	Feb 22 12:13:40 2024	Fail	Error:16	Functional Failure
apb_random_read_write_override_test	Feb 15 17:40 2024	Feb 15 17:40:08 2024	Fail	Error:40	Functional Failure
apb_write_verify_test	Feb 22 12:13 2024	Feb 22 12:13:39 2024	Pass	Error:0	not applicable
apb_base_test	Feb 22 12:13 2024	Feb 22 12:13:40 2024	Fail	Error:0	UVM_error
apb_write_verify_test	None	None	Pass	Error:0	not applicable
apb_base_test	Mar 5 15:17 2024	Mar 5 15:17:01 2024	Fail	Error:0	UVM_error
apb_random_read_write_override_test	None	None	Fail	Error:40	Functional Failure

Sending mail to the given mail-id with excel file attachment

```
connecting to server...
connected to server
Sending email to:lakshmi.m@proxelera.com
Email successfully sent to: lakshmi.m@proxelera.com
```

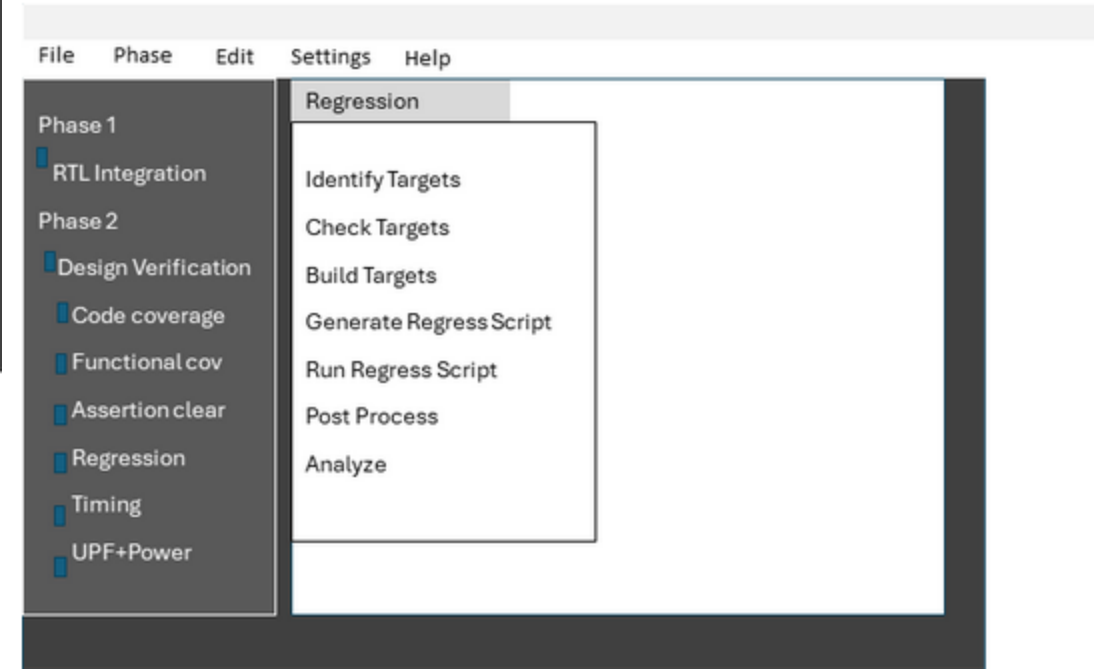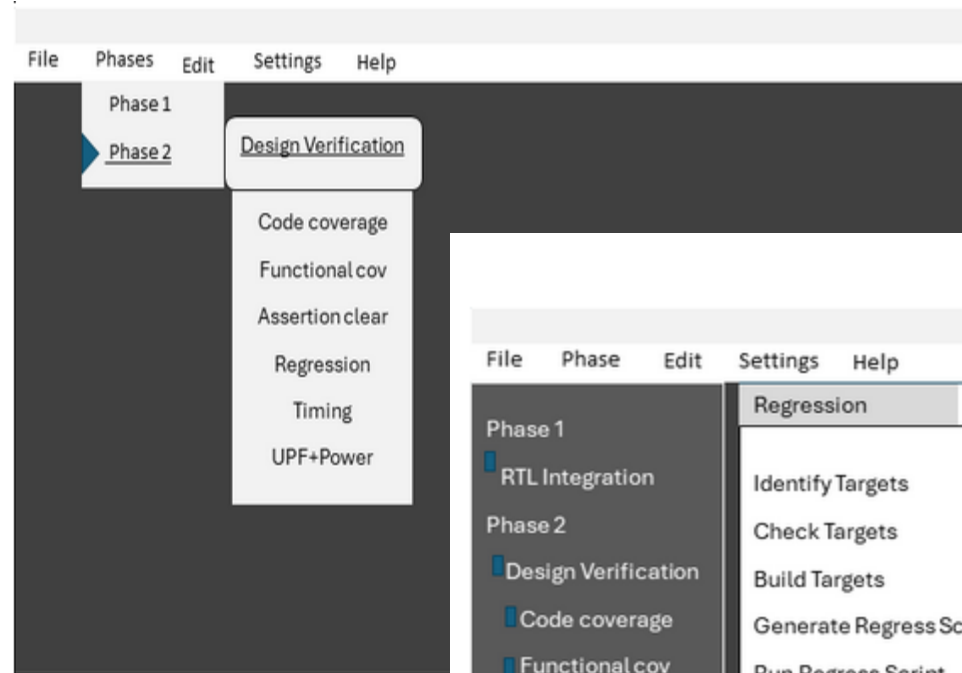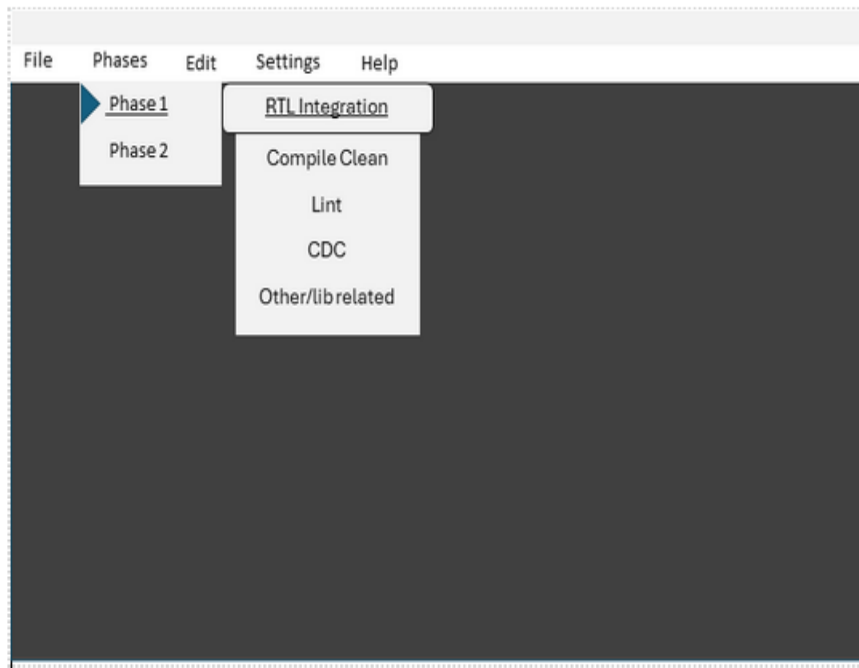# Graph showing total number of test cases passed and failed
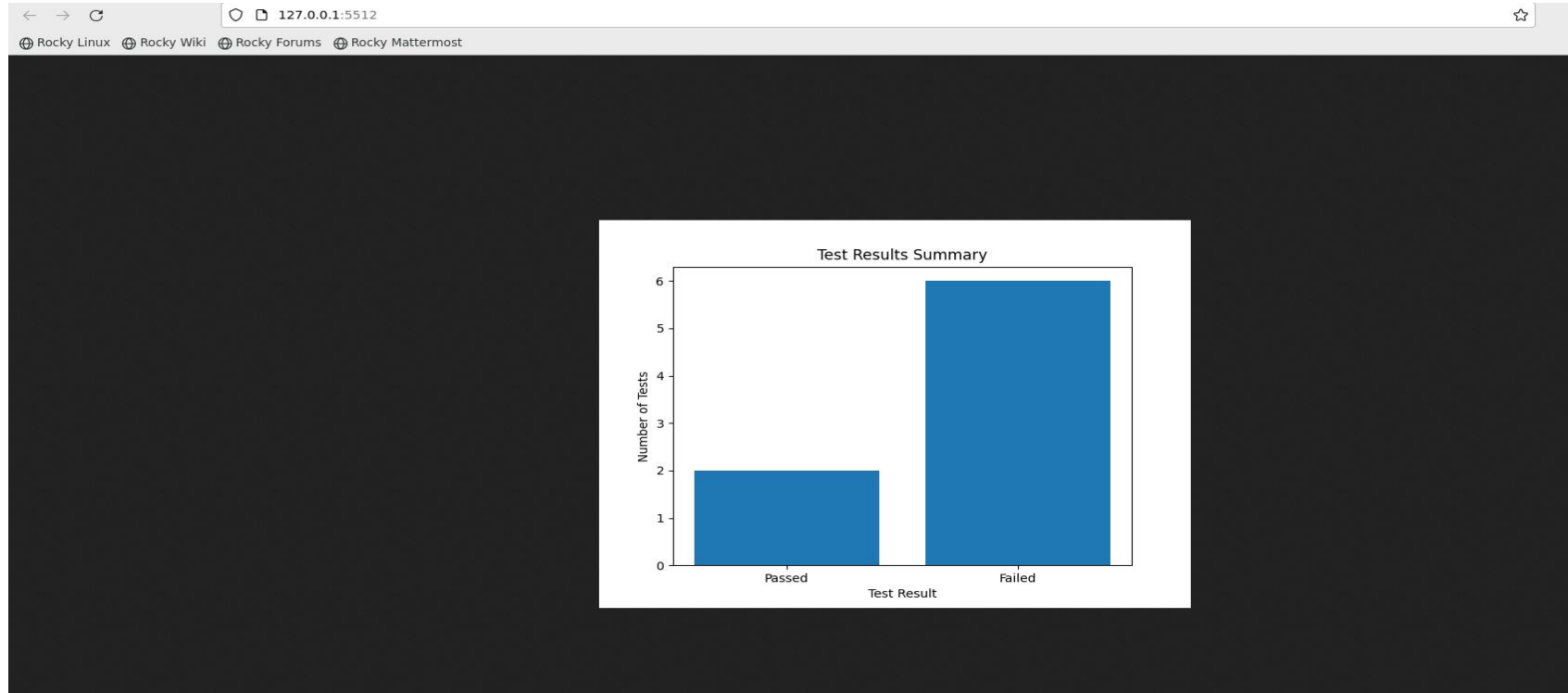
Graph showing test-cases pass and fail in day wise
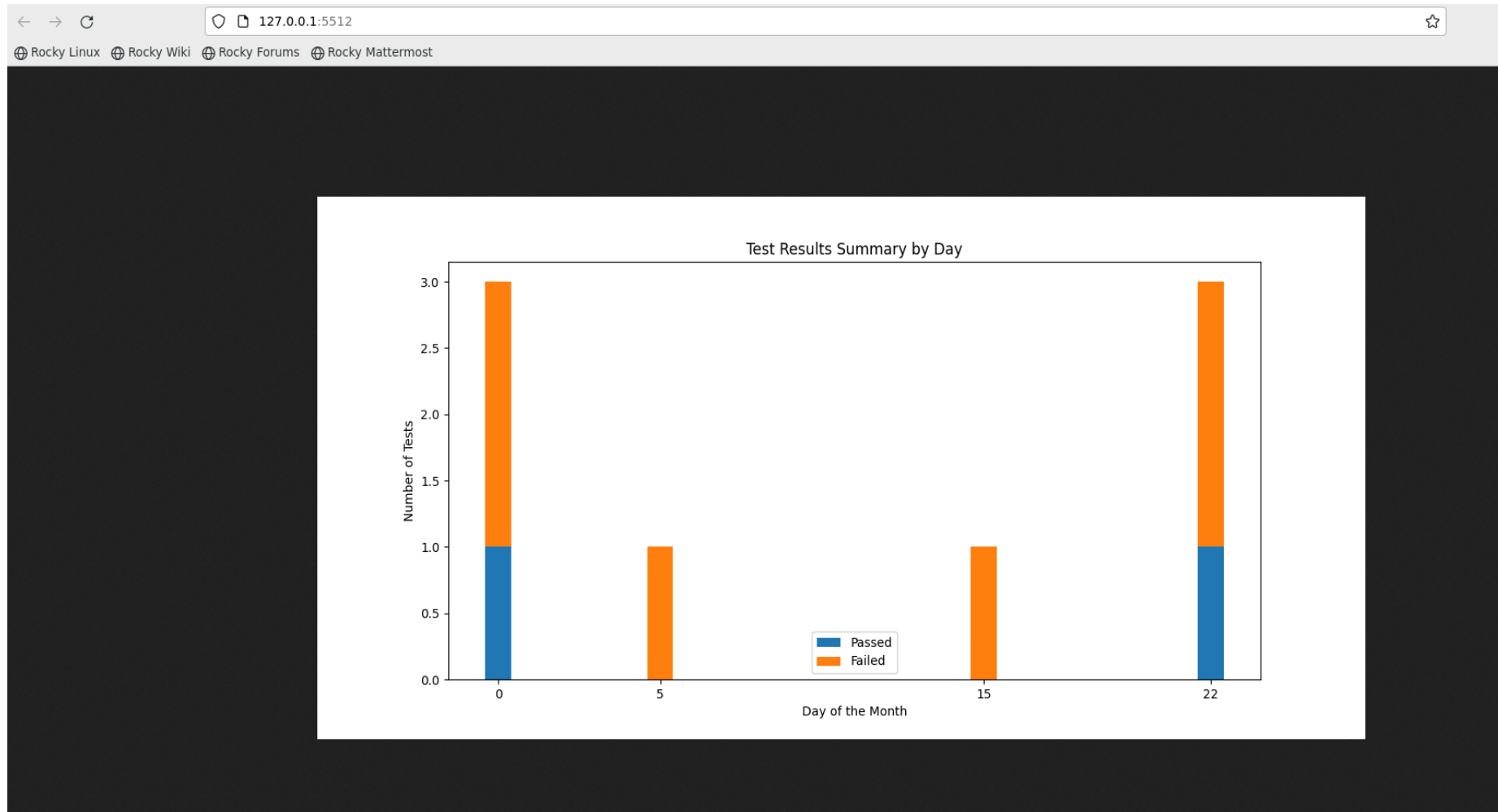
Future Work: Building automation application

Pictorial view of GUI:

Timeline:

S N	Task	ETA
1	Application and Tab Creation	28/03/2024
2	Function Invoking for DV regression	29/03/2024