GENESIS - Learning Outcome & Mini-project Summary Report

LTTS
GLOBAL
ENGINEERING
ACADEMY

L&T Technology Services

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	To be Approved	Remarks/Revision Details
1	9/10/2020	Shivani Khamitkar			
2	11/10/2020				
3	23/10/2020				

Details

Contents

Table of Figures

Miniproject -1 [Individual]

Module

C++ and LINUX

Topic and Subtopics

- STL containers
  Standard Containers. A container is a holder object that stores a collection of other objects (its elements). They are implemented as class templates.

- Iterators
  Iterators are used to traverse from one element to another element, a process is known as iterating through the container. The main advantage of an iterator is to provide a common interface for all the containers type. Iterators make the algorithm independent of the type of the container used.

- Classes
  A class is a user-defined data type that we can use in our program, and it works as an object constructor, and for creating objects.

- Virtual functions
  Virtual functions ensure that the correct function is called for an object, regardless of the type of reference (or pointer) used for function call.

- Inheritance
  Inheritance is a process in which one object acquires all the properties and behaviors of its parent object automatically.

- File handling
  In C++, files are mainly dealt by using three classes fstream, ifstream, ofstream for different sets of operations on file.

- Semaphore
  A semaphore is a variable or abstract data type used to control access to a common resource by multiple processes in a concurrent system such as a multitasking operating system

- Mutex
  The mutex class is a synchronization primitive that can be used to protect shared data from being simultaneously accessed by multiple threads. mutex offers exclusive, non-recursive ownership semantics

- Threads

  Thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system.

Objectives & Requirements

Requirement(High level and Low level)

High level requirements:

The HLR level is the most generalized breakdown of requirements of the system. This level corresponds to major system functions or business processes.

ID	**Description**
HL_01	Count the gold medal of the country in swimming
HL_02	Count the silver medal of the country in swimming
HL_03	Count the bronze medal of the country in swimming

Table 1: High Level Requirements

Low level requirements:

Low-level requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish to meet the high-level software requirements from which it is derived through software design analysis.

ID	Description
LL_01	Read .csv file
LL_02	Parse through country and medal column

Table 2: Low Level Requirements

Design

Flowchart:

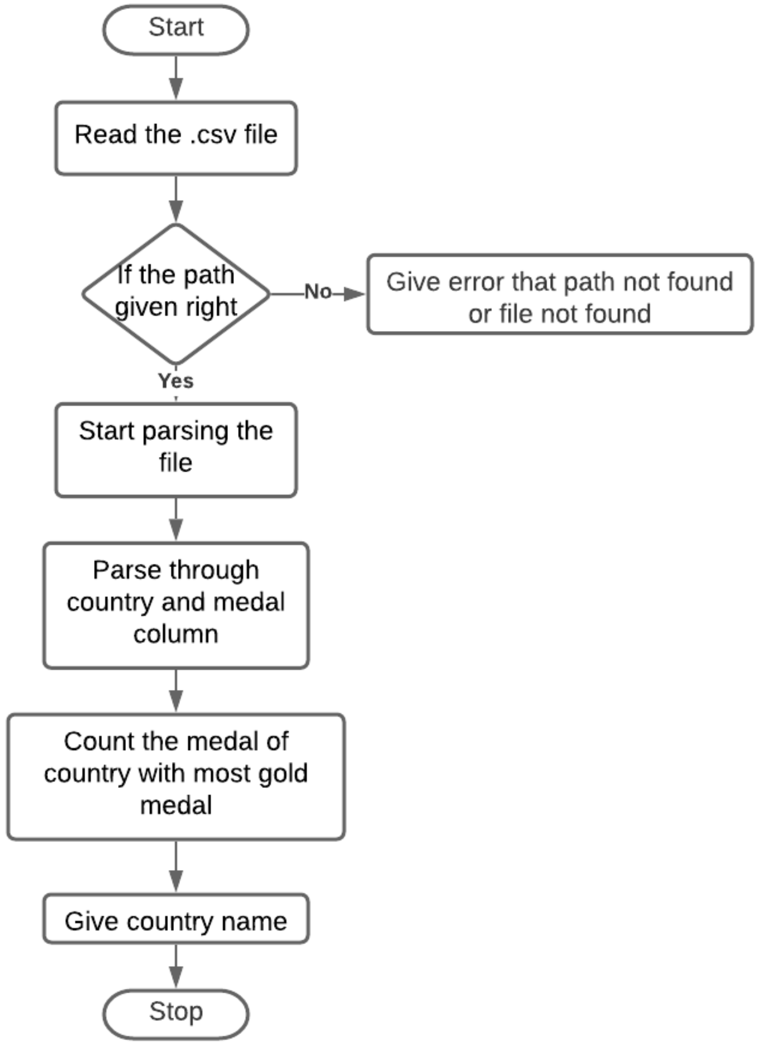

Start
Read the .csv file
If the path given right
No
Give error that path not found or file not found
Yes
Start parsing the file
Parse through country and medal column
Count the medal of country with most gold medal
Give country name
Stop

Figure 2: Flowchart

Class Diagram:

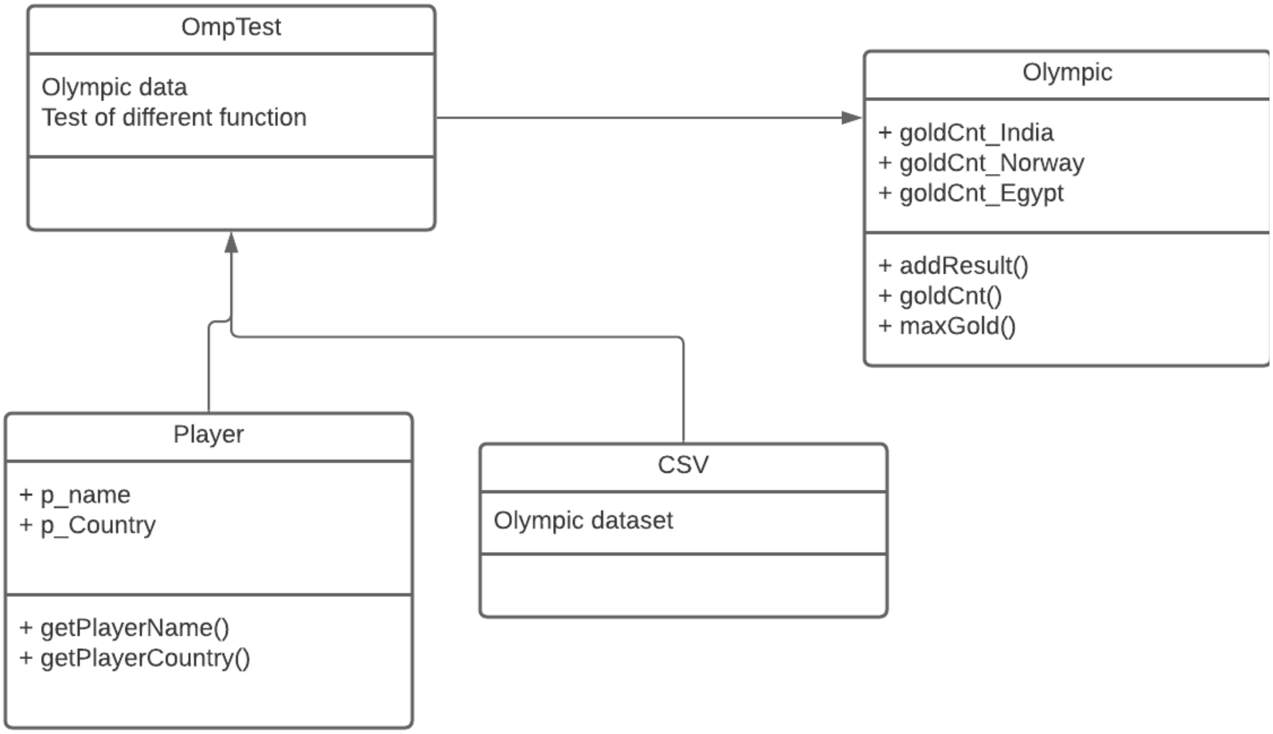


Figure 2: Class Diagram

Case Diagram:

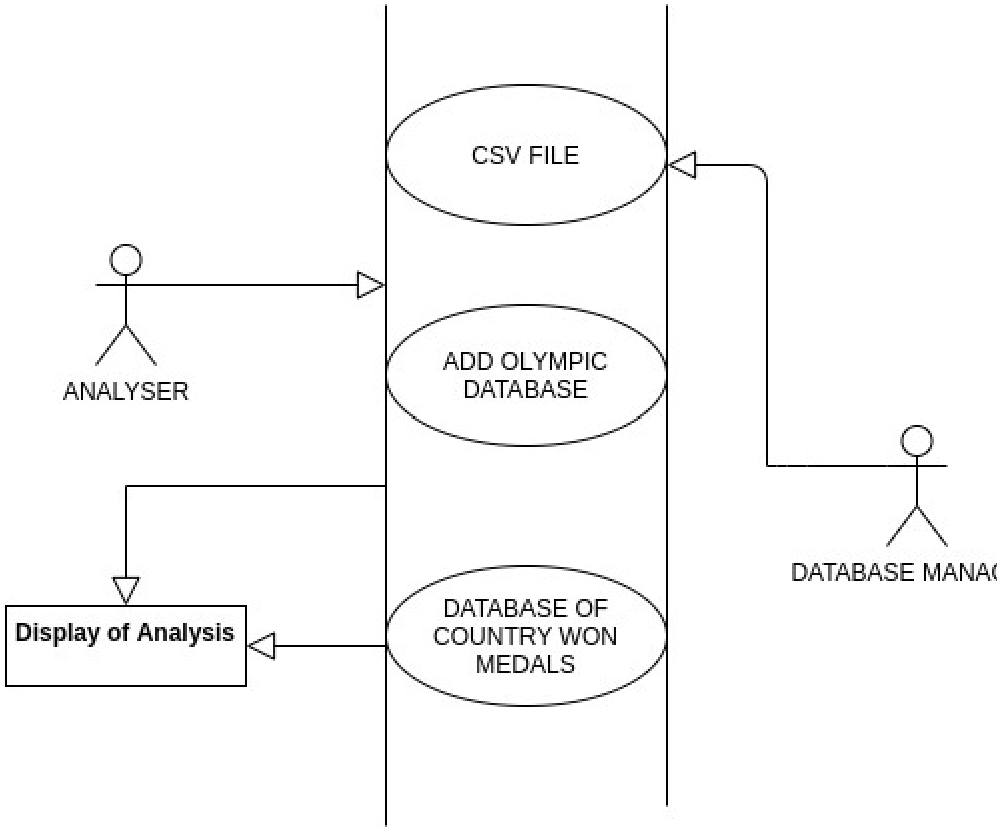


Figure 4: Case Diagram

Test Plan

Unit testing:

Test id	Description	Expected input	Expected output	Actual output
HH_01	Gold Medal count	List of country who won medals in swimming	India	India
HH_02	Silver Medal count	List of country who won medals in swimming	Egypt	Egypt
HH_03	Bronze Medal count	List of country who won medals in swimming	Norway	Norway

Table 3: Unit Testing

Integration testing:

Test id	Description	Expected input	Expected output	Actual output
LL_01	Program read .csv file	.csv file	.csv data	.csv data
LL_02	Read column	Country column	List of option available	List of option available
LL_03	Read column	Medal column	Display of option function	Display of option function
LL_04	CI/CD	GitHub Actions	Cppcheck, valgrind, unit testing , codacy	Passing all CI/CD

Table 4: Integration testing

Implementation Summary

The input file given was .csv file. The .csv file contains data set of the countries winning medals in swimming. Read the .csv file. If the path given is right,start parsing the file if path is not given correct give an error that path not found or file not found.Parse through country and medal column. Count the medals of the country. Extract the country with the most gold medal. Display the country name with most gold medals in the sport. Extract the country with the most silver medal. Display the country name with most gold medals in the sport. Extract the country with the most bronze medal. Display the country name with most gold medals in the sport.Various Linux concepts like Semaphore, Mutex and threads were used in implementing the project.

Video Summary

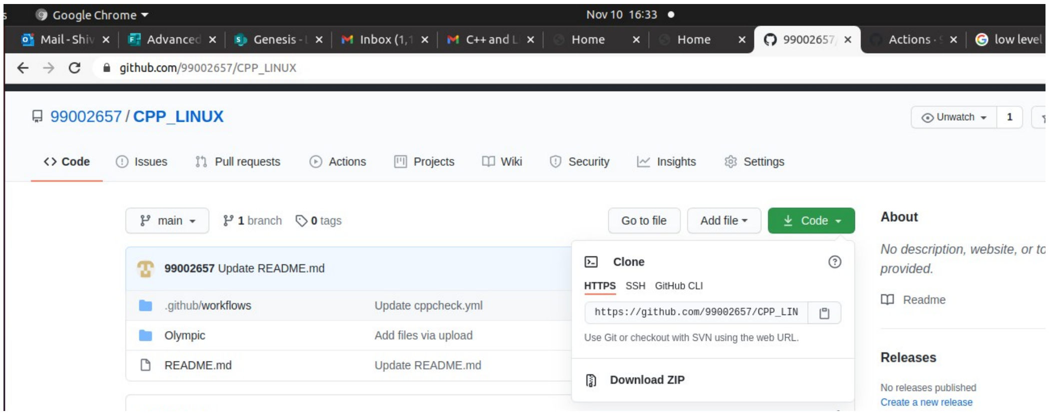
CPP_LINUX_VIDEO

Git Link

CPP and LINUX PROJECT

Git Dashboard

-Code quality
-Cpp-Check
-Valgrind check
-C++ CI
-Unit Test



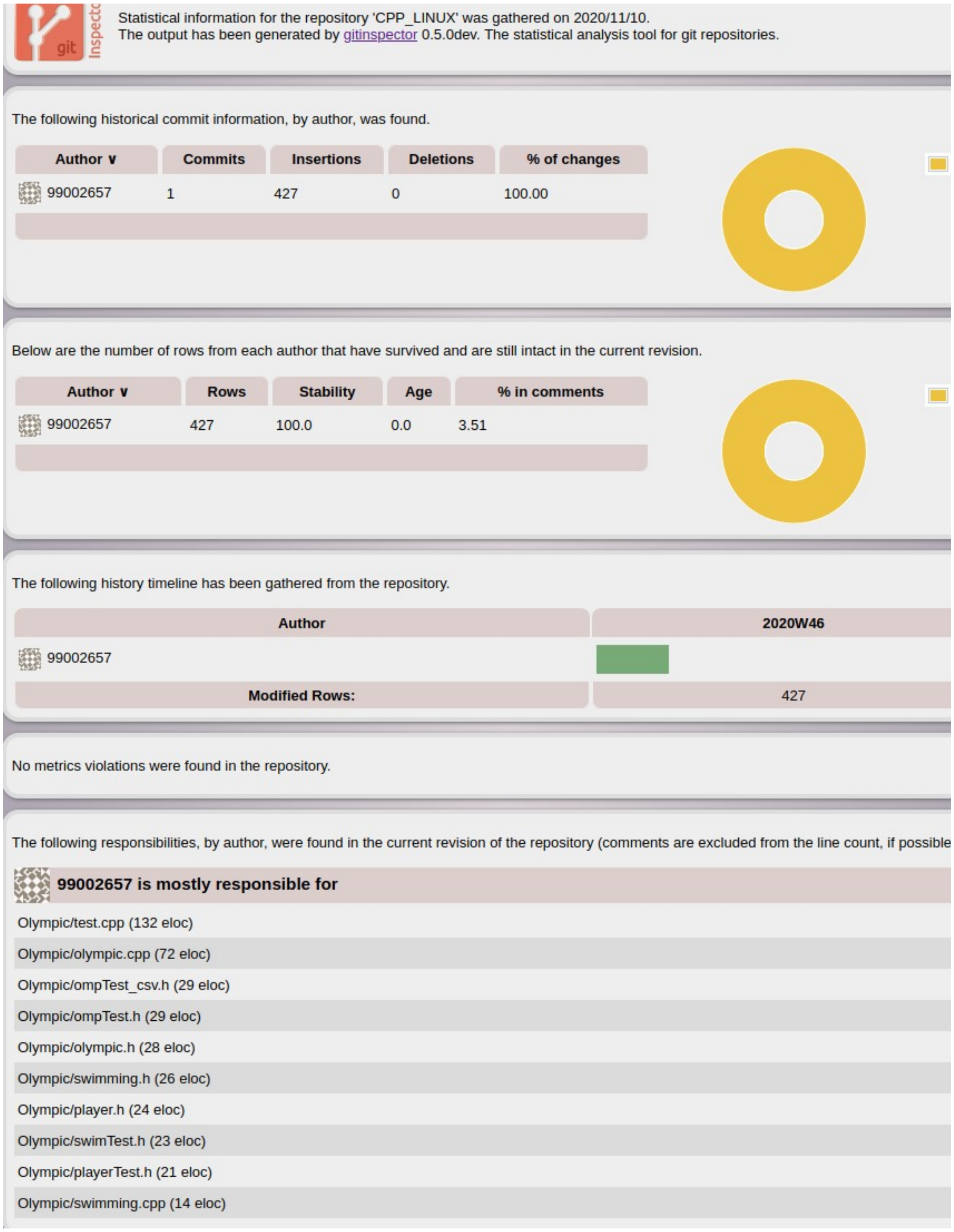
99002657 Update README.md

.github/workflows	Update cppcheck.yml
Olympic	Add files via upload
README.md	Update README.md

Summary

Swimming Olympic data is handled in this project.The extraction of the country name with highest gold ,silver and bronze medal is done with the help of this cpp script..csv file handling is done.STL concepts are used in in this script which made an ease in handling the data set.

Git Inspector Summary



Statistical information for the repository 'CPP_LINUX' was gathered on 2020/11/10.
The output has been generated by gitinspector 0.5.0dev. The statistical analysis tool for git repositories.

The following historical commit information, by author, was found.

Author ∨	Commits	Insertions	Deletions	% of changes
99002657	1	427	0	100.00

Below are the number of rows from each author that have survived and are still intact in the current revision.

Author ∨	Rows	Stability	Age	% in comments
99002657	427	100.0	0.0	3.51

The following history timeline has been gathered from the repository.

Author	2020W46
99002657	🟩
Modified Rows:	427

No metrics violations were found in the repository.

The following responsibilities, by author, were found in the current revision of the repository (comments are excluded from the line count, if possible

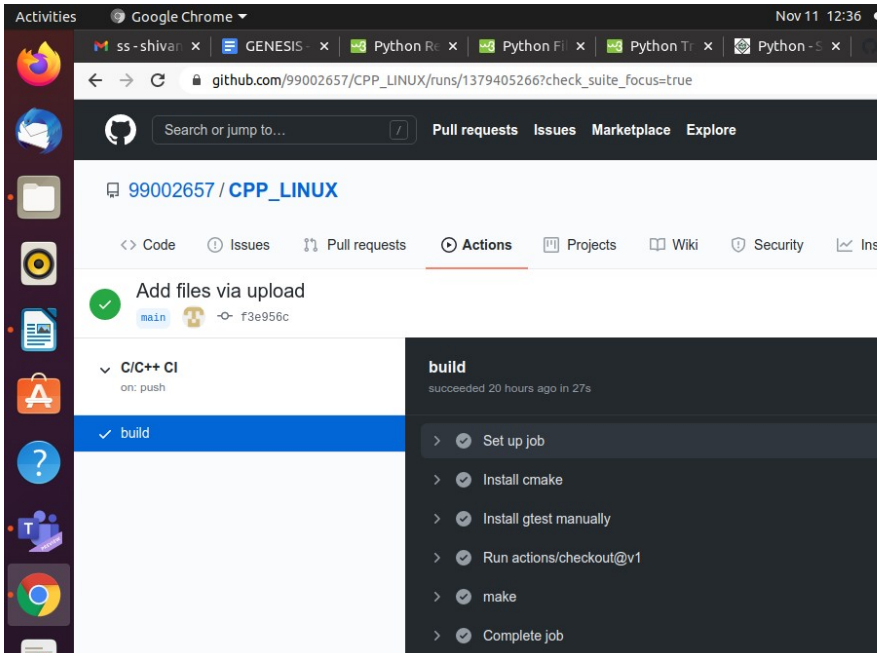
99002657 is mostly responsible for

- Olympic/test.cpp (132 eloc)
- Olympic/olympic.cpp (72 eloc)
- Olympic/ompTest_csv.h (29 eloc)
- Olympic/ompTest.h (29 eloc)
- Olympic/olympic.h (28 eloc)
- Olympic/swimming.h (26 eloc)
- Olympic/player.h (24 eloc)
- Olympic/swimTest.h (23 eloc)
- Olympic/playerTest.h (21 eloc)
- Olympic/swimming.cpp (14 eloc)

Figure 6: Git Inspector Summary

Build

It is the process of creating automated compile environment for multiple file execution.



Add files via upload

main f3e956c

C/C++ CI
on: push

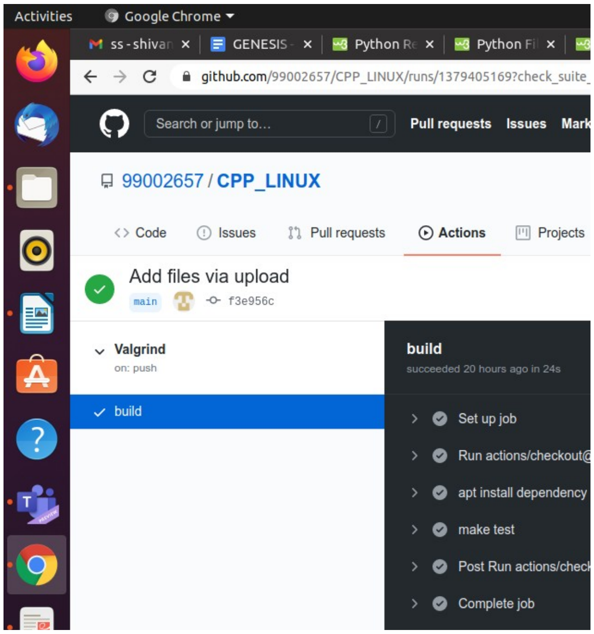
build

build
succeeded 20 hours ago in 27s

- Set up job
- Install cmake
- Install gtest manually
- Run actions/checkout@v1
- make
- Complete job

Figure 8: C/C++ CI

Valgrind is a programming tool for memory debugging, memory leak deduction and profiling.



99002657 / CPP_LINUX

Add files via upload
main f3e956c

Valgrind
on: push

✓ build

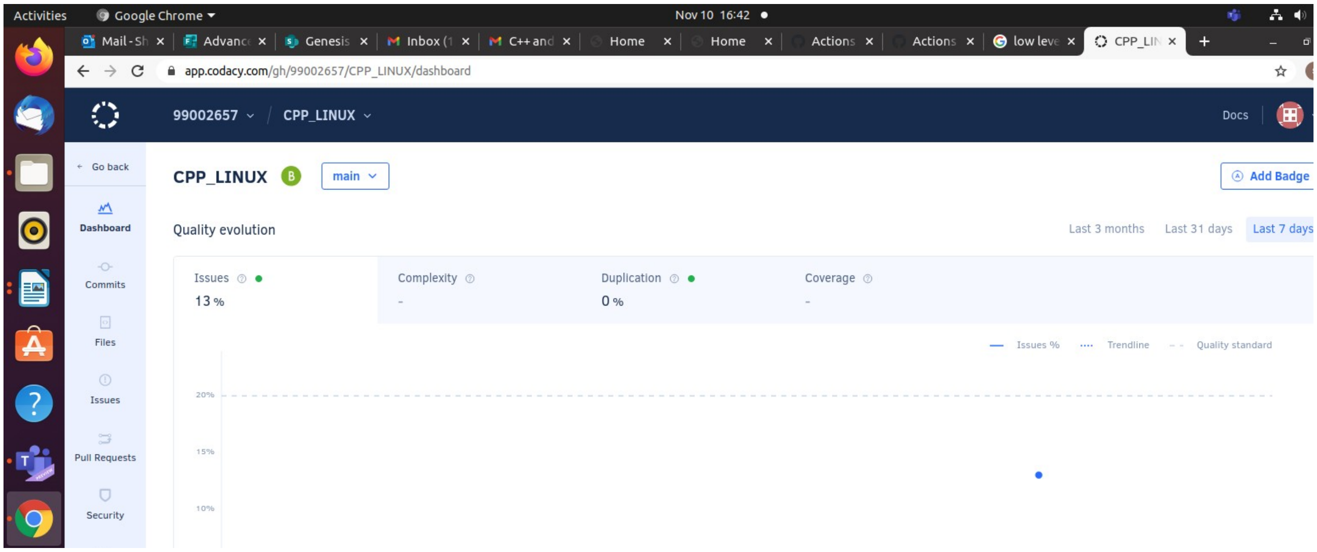
build
succeeded 20 hours ago in 24s

- Set up job
- Run actions/checkout@
- apt install dependency
- make test
- Post Run actions/check
- Complete job

Figure 9: Valgrind

Code quality

Code quality exhibits the clean standard and proper quality of code.



CPP_LINUX B main

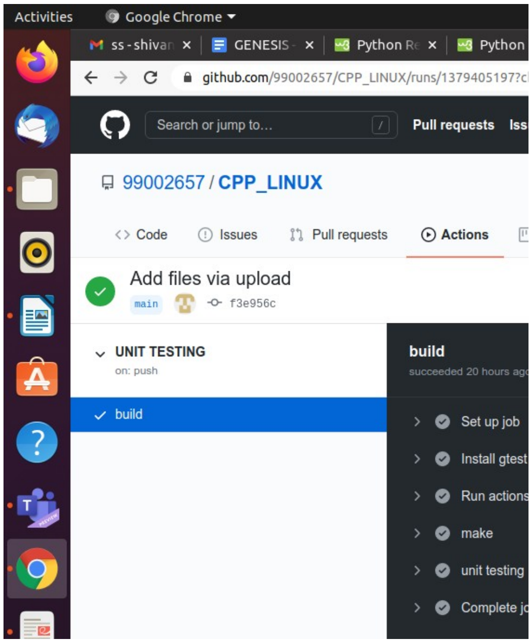
Quality evolution

Issues	Complexity	Duplication	Coverage
13 %	-	0 %	-

Figure 10: Code Quality

Unit Testing

Unit testing plan is being setup to get the desired output and the test plan being cross checked after the implementation.

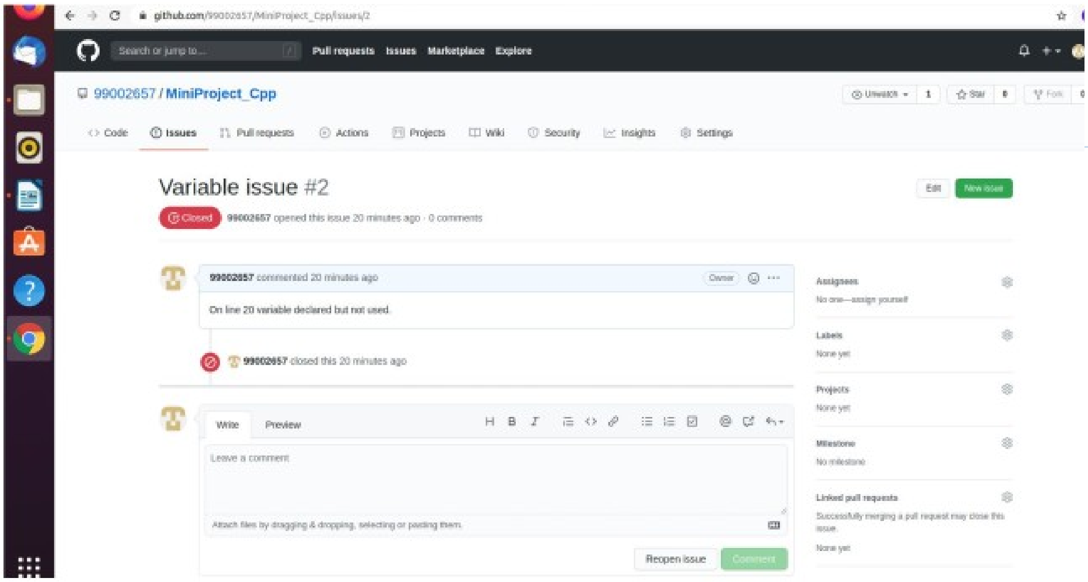


Add files via upload

- Set up job
- Install gtest
- Run actions
- make
- unit testing
- Complete j

Figure 11: Unit Testing

Issues



Variable issue #2

99002657 opened this issue 20 minutes ago · 0 comments

99002657 commented 20 minutes ago

On line 20 variable declared but not used.

99002657 closed this 20 minutes ago

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Linked pull requests: Successfully merging a pull request may close this issue.

None yet

Figure 12: Issues raised and solved

Challenges faced and how were they overcome

Challenges faced during C++ mini project was file handling and parsing the .csv file.Extracting the data from the cells was challenging.

Future Scope

Expansion of this project can be done by including different form of sports as this scripts analysis only one sport that is swimming hence by adding different sports we can analysis different sports using same script

Miniproject -2 [Team]

Module-Python

Topic and Subtopics

Objectives & Requirements

Concepts used in micro project includes:

- Classes

  A class is a user-defined data type that we can use in our program, and it works as an object constructor, and for creating objects.
- Inheritance

  Inheritance is a process in which one object acquires all the properties and behaviors of its parent object automatically.
- Regular expression

  A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern.RegEx can be used to check if a string contains the specified search pattern.

- File handling

  File handling is an important part of any web application.

  Python has several functions for creating, reading, updating, and deleting files.

- Exception handling

  When an error occurs, or exception as we call it, Python will normally stop and generate an error message.The `try` block lets you test a block of code for errors. The except block lets you handle the error. The finally block lets you execute code, regardless of the result of the try-and except blocks.

- SMTP Protocols

  Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.

  Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

The libraries utilized in the micro project includes:

- re
  The Python module re provides full support for Perl-like regular expressions in Python. The re module raises the exception if an error occurs while compiling or using a regular expression

- os
  The OS module in python provides functions for interacting with the operating system. OS, comes under Python’s standard utility modules.

- smtplib
  The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP

- textract
  This package is organized to make it as easy as possible to add new extensions and support the continued growth and coverage of textract

- pandas as pd -
  pandas is a software library written for the Python programming language for data manipulation and analysis

Design

Flowchart

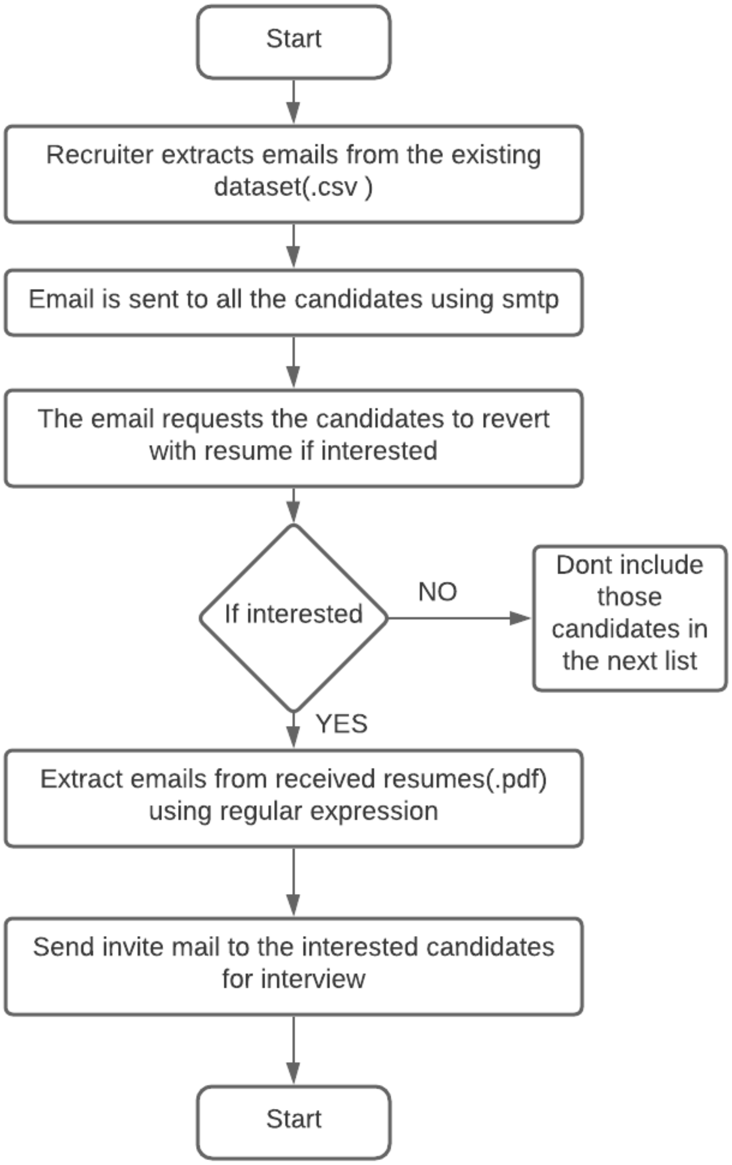

Start
Recruiter extracts emails from the existing dataset(.csv)
Email is sent to all the candidates using smtp
The email requests the candidates to revert with resume if interested
If interested
NO
Dont include those candidates in the next list
YES
Extract emails from received resumes(.pdf) using regular expression
Send invite mail to the interested candidates for interview
Start

Figure 14: Flowchart

Test Plan

Sr No.	Test Description	Expected Output	Actual Output
1	Send emails to all candidates	Successfully sent	Successfully sent
2	Send email to interested candidates	Successfully sent	Failure
3	Send email to interested candidates	Successfully sent	Successfully sent

Implementation Summary

- In this project, the first step was creating a class diagram based on the requirements.
- Extraction of the required data from .csv file was done and the recruiter extracts emails from the data set(.csv).
- Email is sent to all the candidates using SMTP protocol.
- The email requests the candidates to revert with resume if interested.
- The resume contains the email address of the candidates. Using regular expression, we extract the email IDs from the resumes.
- After this, an email is sent to the interested candidates stating the date and time of the interview.

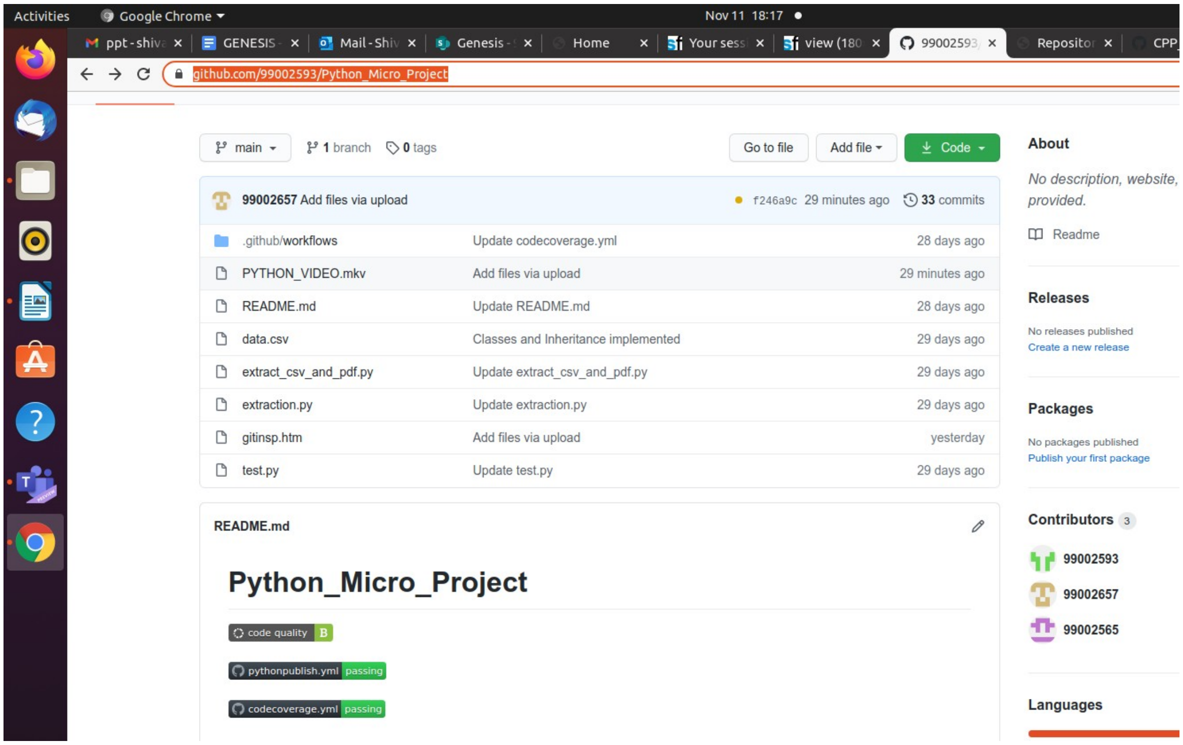
Video Summary

PYTHON_VIDEO

Git Link

PYTHON PROJECT

Git Dashboard



99002657 Add files via upload · f246a9c 29 minutes ago · 33 commits

.github/workflows	Update codecoverage.yml	28 days ago
PYTHON_VIDEO.mkv	Add files via upload	29 minutes ago
README.md	Update README.md	28 days ago
data.csv	Classes and Inheritance implemented	29 days ago
extract_csv_and_pdf.py	Update extract_csv_and_pdf.py	29 days ago
extraction.py	Update extraction.py	29 days ago
gitinsp.htm	Add files via upload	yesterday
test.py	Update test.py	29 days ago

README.md

Python_Micro_Project

code quality B

pythonpublish.yml passing

codecoverage.yml passing

About

No description, website, provided.

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

99002593
99002657
99002565

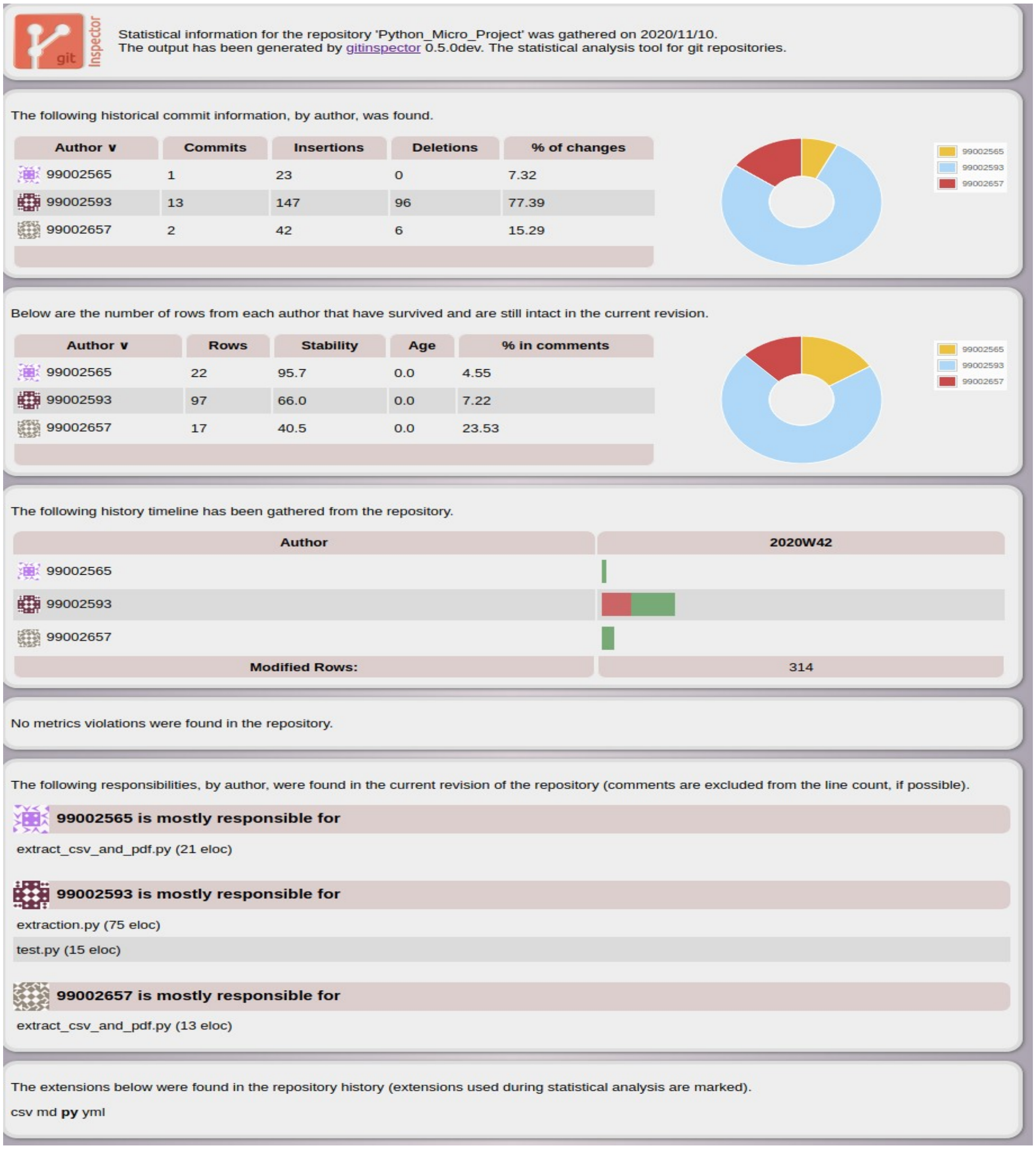
Languages

Figure 14: Git Dashboard

Summary

Traditionally the process in recruiting the candidates involves a lot of manual work which consumes a lot of time. In this project, this scenario is highlighted and a solution is provided in the project with the help of certain python concepts and libraries like SMTP protocol, regular expressions, textract and various others. The solution includes automated recruiting process which extracts the email from the data set and communicates with the candidates.

Git inspector summary



Statistical information for the repository 'Python_Micro_Project' was gathered on 2020/11/10.
The output has been generated by gitinspector 0.5.0dev. The statistical analysis tool for git repositories.

The following historical commit information, by author, was found.

Author ∨	Commits	Insertions	Deletions	% of changes
99002565	1	23	0	7.32
99002593	13	147	96	77.39
99002657	2	42	6	15.29

Below are the number of rows from each author that have survived and are still intact in the current revision.

Author ∨	Rows	Stability	Age	% in comments
99002565	22	95.7	0.0	4.55
99002593	97	66.0	0.0	7.22
99002657	17	40.5	0.0	23.53

The following history timeline has been gathered from the repository.

Author	2020W42
99002565	
99002593	
99002657	
Modified Rows:	314

No metrics violations were found in the repository.

The following responsibilities, by author, were found in the current revision of the repository (comments are excluded from the line count, if possible).

99002565 is mostly responsible for

extract_csv_and_pdf.py (21 eloc)

99002593 is mostly responsible for

extraction.py (75 eloc)

test.py (15 eloc)

99002657 is mostly responsible for

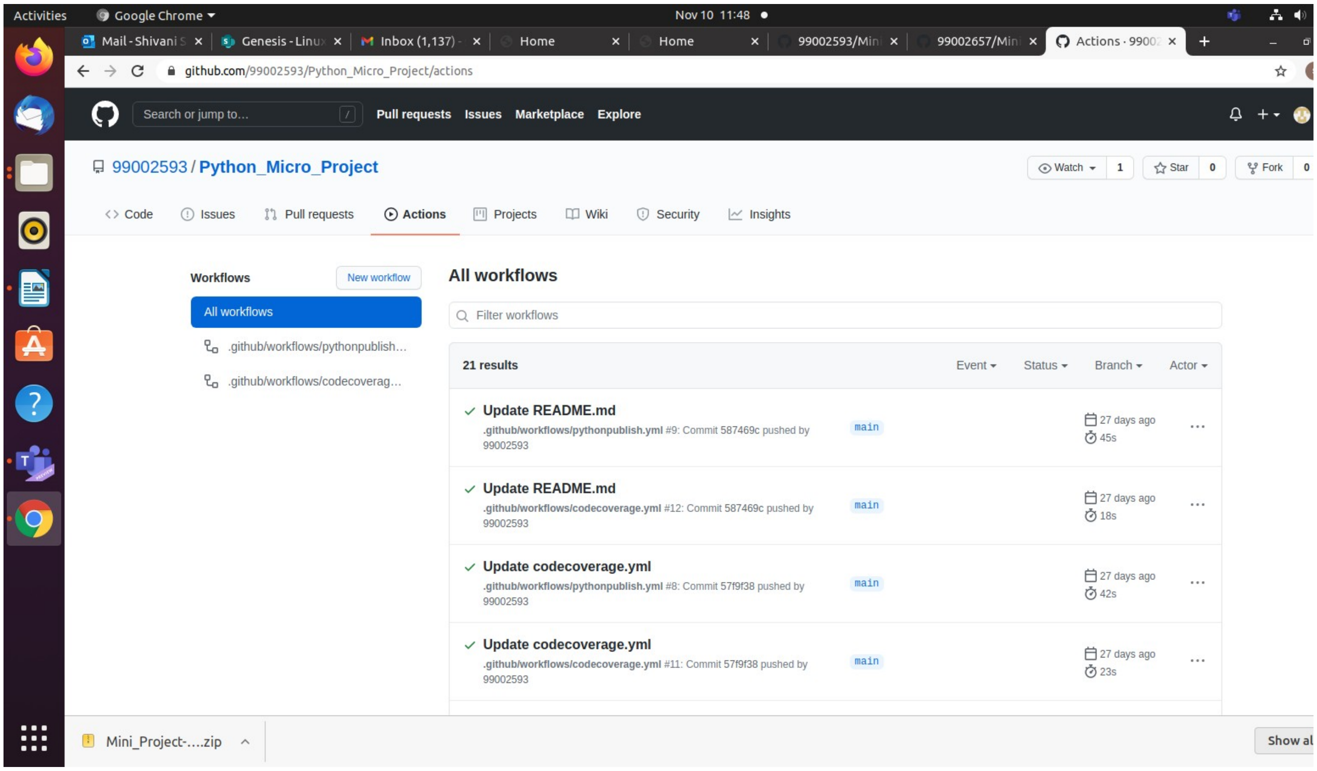
extract_csv_and_pdf.py (13 eloc)

The extensions below were found in the repository history (extensions used during statistical analysis are marked).

csv md **py** yml

Figure 16: Git Inspector Summary

Build



All workflows

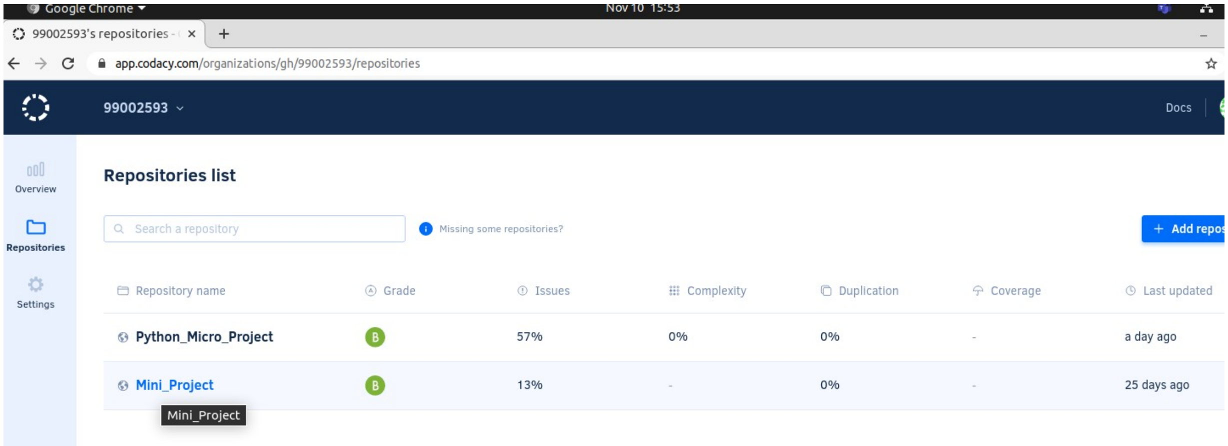
Filter workflows

21 results — Event, Status, Branch, Actor

Workflow run	Branch	When	Duration
✓ Update README.md .github/workflows/pythonpublish.yml #9: Commit 587469c pushed by 99002593	main	27 days ago	45s
✓ Update README.md .github/workflows/codecoverage.yml #12: Commit 587469c pushed by 99002593	main	27 days ago	18s
✓ Update codecoverage.yml .github/workflows/pythonpublish.yml #8: Commit 57f9f38 pushed by 99002593	main	27 days ago	42s
✓ Update codecoverage.yml .github/workflows/codecoverage.yml #11: Commit 57f9f38 pushed by 99002593	main	27 days ago	23s

Figure 17: Build

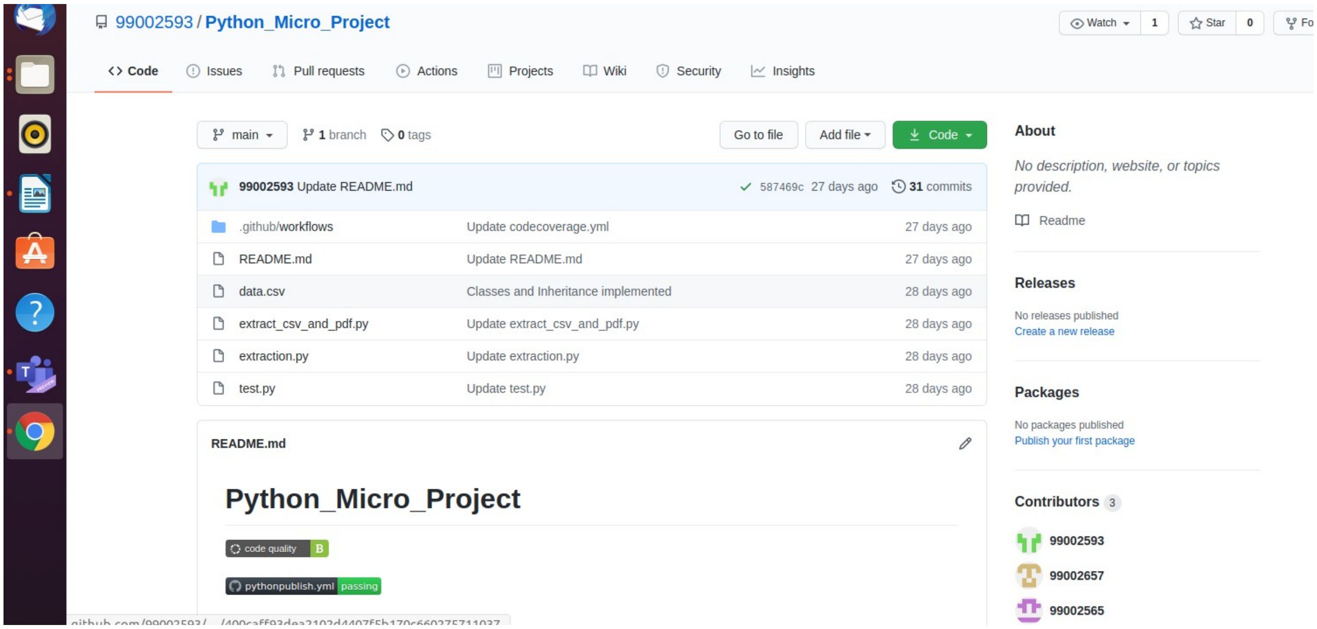
Code quality



Repositories list

Repository name	Grade	Issues	Complexity	Duplication	Coverage	Last updated
Python_Micro_Project	B	57%	0%	0%	-	a day ago
Mini_Project	B	13%	-	0%	-	25 days ago

Figure 18: Code Quality



99002593 Update README.md ✓ 587469c 27 days ago 31 commits

.github/workflows	Update codecoverage.yml	27 days ago
README.md	Update README.md	27 days ago
data.csv	Classes and Inheritance implemented	28 days ago
extract_csv_and_pdf.py	Update extract_csv_and_pdf.py	28 days ago
extraction.py	Update extraction.py	28 days ago
test.py	Update test.py	28 days ago

README.md

Python_Micro_Project

About

No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

Packages

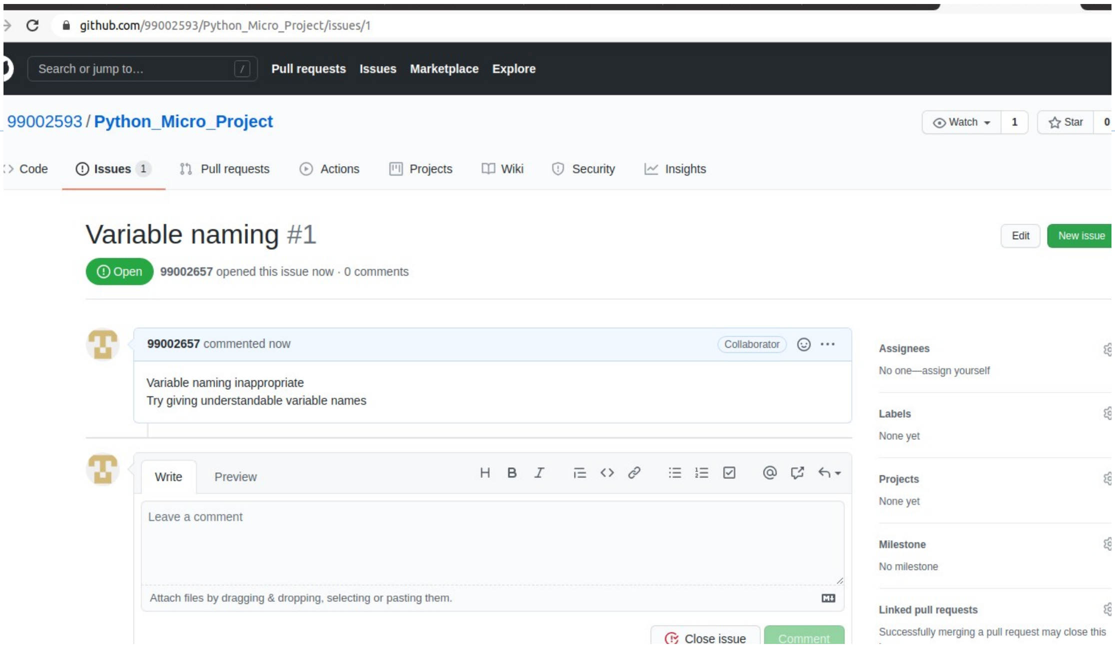
No packages published
Publish your first package

Contributors 3

99002593
99002657
99002565

Figure 19: Code Quality

Issues



Variable naming #1

Open 99002657 opened this issue now · 0 comments

99002657 commented now

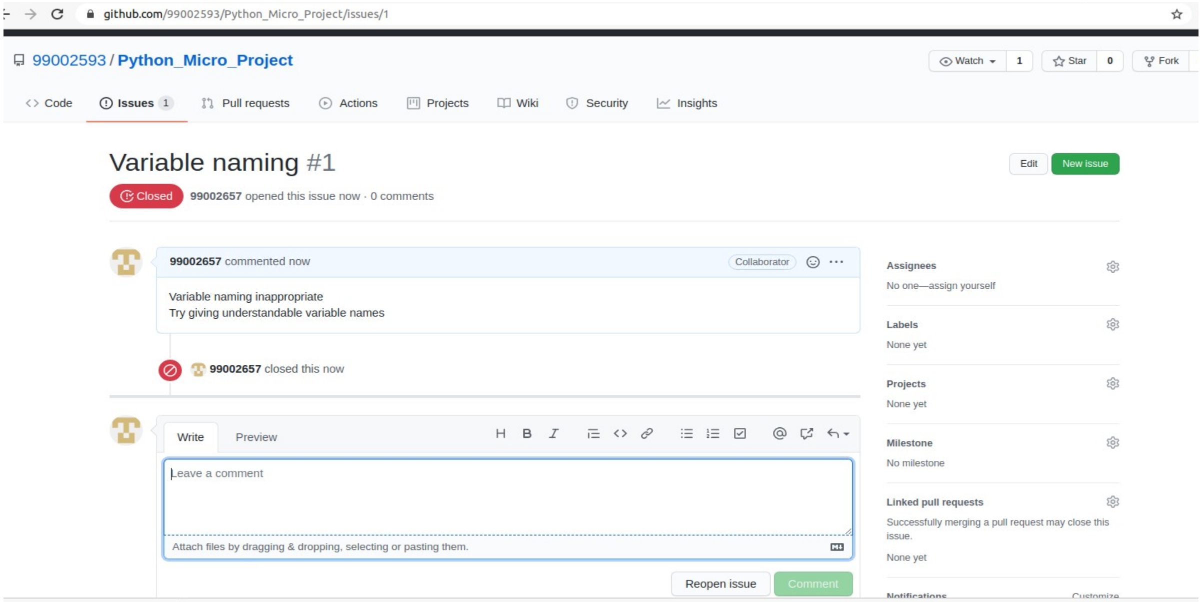
Variable naming inappropriate
Try giving understandable variable names

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Assignees: No one—assign yourself
Labels: None yet
Projects: None yet
Milestone: No milestone
Linked pull requests: Successfully merging a pull request may close this

Figure 20: Issues Raised


github.com/99002593/Python_Micro_Project/issues/1

99002593 / Python_Micro_Project

Variable naming #1

Closed 99002657 opened this issue now · 0 comments

99002657 commented now Collaborator

Variable naming inappropriate
Try giving understandable variable names

99002657 closed this now

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

Notifications Customize

Figure 21: Issues

Individual Contribution & Highlights

The project was developed and evolve by me (Shekar Jk) PS. No 99002539, Shivani Suryakant Khamitkar PS. No 99002657 and Nipun Bhaskar Tank PS. No 99002565 in the due course of genesis program as my python micro project where we implement all the major concepts taught. Also, I acknowledge with immense pleasure the sustained interest, encouraging attitude and constant inspiration rendered by Mr. Sarvepalli Venkata Naga Badrinath for the continued drive for better quality in everything.

NAME	PS NO	MAIL ID	CONTRIBUTION
Shivani Khamitkar	99002657	shivani.suryakantkhamitkar@ltts.com	Implemented regular expressions and files handling.
Nipun Bhaskar Tank	99002565	nipun.bhaskartank@ltts.com	Implemented SMTP Protocols
Shekar Jk	99002593	shekar.jk@ltts.com	Implemented OOP concepts.

Challenges faced and how were they overcome

Challenges faced during this project were implementation of SMTP protocol as we had to access g mail account through python script this was challenging as g mail doesn't entertain scripts to handle it due to security reasons. We had to put a lot of effort to find a way to access it by changing various settings of g mail. Implementation of including OOP concepts into the program. Setting up git workflows for Python.

Future Scope

Traditionally the process in recruiting the candidates involves a lot of manual work which consumes a lot of time. In this project, this scenario is highlighted and a. The solution includes automated recruiting process which extracts the email from the data set and communicates with the candidates which reduces a lot of manual work hence can be implemented in most of the colleges and offices in near future.This project can be further updated to make it available for the nautomated acknowledgement as a single line message received to the sender once the mail is sent to all.
This project can also be updated to make it more safer by using highly-secured protocol to transfer the mails.