# REFLECTION

# ON

# SOFTWARE ENGINEERING PROJECT MANAGEMENT MODULE

Emmanuel Eshun-Davies
Essex University, United Kingdom
MSc Computer Science

July 24, 2023

## Introduction

As a professional software engineer actively working in the software engineering industry, I have been exposed to project management concepts and practices in several instances. I understand the importance of defining clear project objectives, setting timelines, and allocating resources effectively to ensure the successful completion of projects. In my previous work experiences, I have been part of development teams where we followed agile methodologies, such as Scrum, to manage projects in an iterative and incremental manner.

During these projects, I actively participated in daily stand-up meetings, sprint planning sessions, and retrospectives, which allowed me to collaborate closely with team members and track progress effectively. Additionally, I utilized project management tools like Jira and Trello to manage tasks, track issues, and monitor project timelines.

While all that has given me a somewhat solid foundation in project management, I have always been eager to delve deeper into the subject and understand the underlying principles and best practices. I knew that by gaining a more comprehensive understanding of project management, I could contribute more effectively to the success of future projects.

When I started the software engineering project management module, my excitement was palpable. I looked forward to learning from the experienced instructor, coursebooks and collaborating with peers who shared the same passion for enhancing their project management skills. The course offered a structured approach to project management, covering topics like risk assessment, stakeholder management, and project scheduling, which I knew would further refine my abilities as a software engineer.

Though I was really excited about the course, like in previous modules, I also struggled with having a balance between family, work, ministry, and studies. This slowed down my participation in discussions. Haven already dealt with stress and time management issues; I would say I have gotten better by adopting new ways of getting on track with each unit. I however took advantage of every opportunity to ask questions and learn from everyone including my teammates. The course content has vastly expanded my knowledge base and introduced me to new tools and techniques that could be applied in real-world scenarios.

## Project Description and Critical Incidents

At the start of the course, we were divided into groups, and I was assigned to Group 1. Considering previous academic teams, I have been in, this has been my best and most productive team yet. Everyone in the team was ready to assist the other from the very onset. I honestly couldn't have asked for a better team. We were able to quickly identify the skill(s) of each teammate and analyzed the strengths and weaknesses of our team based on the requirements of the assignment. This already gave us an idea of how to proceed further. Due to my experience as a software Engineer, I was mainly in charge of the Gherkin Specifications, Identifying the software requirements of the project and a part of the application development, refinement and refactoring, Verification and Validation, entire unit tests, linting and linting report.

The whole module presented 2 assignments which were linked to each other. In its entirety, we were to present our analysis of the provided case study transcript and the subsequent development of an application for estimating project costs. Our team diligently extracted the initial requirements, justified the chosen development methodology, created a set of Gherkin specifications, and devised a fully costed project plan for the system. Our team comprised five members, each assigned specific roles and responsibilities.

At the beginning of the assignment, we had a team of five members, which allowed us to distribute responsibilities across different roles effectively. However, when one team member left the course, our team was reduced to four members. This sudden change disrupted the initial plan, leaving us with an uneven distribution of tasks and roles.

Another challenge we encountered was that some team members had expertise in specific roles, while others were not as well-versed in certain areas, particularly software development. This disparity in skills created a bottleneck in tasks related to software development, as the responsibility fell on a few individuals who were proficient in that domain. As a result, those members had to handle additional workload and responsibilities, causing a strain on their time and productivity.

Although we had a WhatsApp group where we communicated consistently, we all seemed to work and so could not communicate as effective as may be required during the day. Our strongest point of communication then was during our weekly meetings. For this, if anyone missed a meeting, it affected productivity to an extent. In the middle of the project, we had a few people not being able to attend meetings due to time differences and personal activities they did not have control over, leading to a temporary halt in progress.

In resolving the communication issues, teammates had to identify who they needed to get unblocked from their current task and set up a meeting with them to get updated or get the needed support to proceed. This solution brought us back on track as we now did not have to wait a whole week to get unblocked on a task.

There was also the instance of the unfamiliarity with writing Gherkin statements in the team. As the project progressed, it became apparent that we needed to create Gherkin specifications. However, none of us had prior experience in writing Gherkin statements, which posed a significant obstacle to the project's advancement.

Recognizing the importance of Gherkin in defining test scenarios and requirements for behavior-driven development (BDD), I took the initiative to assist the team by conducting research on the topic and I was glad to have done this. Haven prior experience with Pseudo

Codes and User Stories, I thought this to be same however at the end of my research, I concluded that although they all seem to be used as abstractions to communicate complex ideas in a simplified manner, user stories are used to capture high-level requirements from the perspective of end-users or stakeholders. while Pseudo-code is a way to represent algorithms or logic in a human-readable format. And finally, Gherkin statements, which are part of Behavior-Driven Development (BDD) and are used to define behavior scenarios and acceptance criteria in a business-readable format.

**Emotional Response and Analysis**
Throughout this module, I have experienced a whirlwind of emotions that have shaped my learning journey and personal growth. Initially, as I enrolled in the course, I was filled with excitement and eagerness to dive into the world of software engineering project management. The prospect of contributing to the creation of a practical application filled me with a sense of accomplishment and anticipation. I was determined to make the best of every part of the module and actively engage in the team projects and course discussions.

However, as the weeks progressed and challenges arose, I found myself facing moments of frustration and self-doubt. Time management became a significant hurdle, and I struggled to strike a balance between my academic commitments, professional responsibilities, family life, and my role in ministry. The mounting pressure from all aspects of life weighed heavily on me, impacting my overall productivity and ability to fully immerse myself in the course.

In these moments of self-doubt, I reflected on the stress management module from the beginning of the MS Computer Science course. It served as a valuable reminder that stress can be both a motivating factor and a hindrance to progress. I realized that embracing stress as a positive force could drive me to excel, pushing me beyond my perceived limitations. Instead of succumbing to regrets about past time mismanagement, I made a conscious effort to focus on what remained and how I could maximize my contributions and participation moving forward.

One of the key lessons I learned from my prior experiences in team projects was the importance of effective communication and equitable task distribution. Recognizing that our team faced the challenge of one member leaving, I realized the need to reassess roles and responsibilities to ensure fairness and support for all team members. This open communication within the team allowed us to realign roles based on individual strengths and interests, thus optimizing our collective potential.

Although the emotional rollercoaster impacted my overall productivity, it also prompted self-reflection and growth. I learned to embrace challenges and view them as opportunities for development. It was essential for me to adopt a positive mindset and turn these moments of frustration into steppingstones towards personal and professional growth.

To manage the overwhelming demands from work, family, and ministry, I sought to enhance my time management skills and create a structured schedule that accommodated all aspects of my life. Prioritizing tasks and setting realistic goals helped me regain a sense of control and reduced the feeling of being overwhelmed.

Moreover, the support and understanding from my family, colleagues, and fellow team members were instrumental in easing the emotional burden. The sense of camaraderie within the team motivated me to persevere, knowing that we were all in this together and that our collective effort would lead to a successful outcome.

This module has been a journey of self-discovery, resilience, and growth. I have navigated a range of emotions, from excitement to frustration, and learned to embrace stress as a positive force that drives me to excel. Through open communication, equitable task distribution, and effective time management, I have been able to overcome challenges and make the most of this learning experience. I am grateful for the lessons learned and look forward to applying these newfound insights in my capstone project and future endeavors.

**Learning and Changed Actions**

This project has been an enlightening and transformative learning experience for me. At the outset, I must admit that I harbored certain misconceptions, especially regarding the usefulness of waterfall methodologies. However, as the module progressed, I came to realize that different engineering environments may benefit from implementing a waterfall approach to ensure a smooth flow of work. I now understand that selecting a development methodology involves careful consideration of project requirements, and personal preferences or trends should not dictate the decision-making process.

Moreover, this project has highlighted the paramount importance of open communication, transparency, and active participation within a team. I have come to recognize that these factors are pivotal in ensuring effective collaboration and successful project outcomes. The module has been instrumental in dispelling some of my prior assumptions about project management, and I have been eager to share this newfound knowledge with my colleagues. For instance, I initiated discussions on how we can improve our team's project management processes, advocating for shorter stand-up meetings focused on progress updates rather than problem-solving.

In parallel to this course, my workplace organized a SCRUM training, which further complemented my learning journey. Exposure to both project management methodologies has enabled me to discern the strengths and weaknesses of each and apply the most suitable practices to different projects. Moving forward, I am committed to expressing my opinions and concerns constructively, actively contributing to decision-making, and seeking feedback from others to foster a culture of continuous improvement.

In terms of technical skills, this project has deepened my understanding of development in a Python environment. My current work predominantly involves C# and C++, and this exposure to Python like from previous modules, has contributed to broadening my expertise, making me more versatile in software development. Additionally, I have gained valuable insights into project management principles, which I intend to apply not only in future real-world experiences but also in my academic pursuits.

To enhance my competence, I have realized the importance of proactive task management. As such, I am eager to seek challenging tasks and broaden my skillset in diverse areas of my career. I believe that taking initiative and venturing out of my comfort zone will lead to personal growth and ultimately contribute to the success of future projects.

Throughout the project, I made substantial contributions to various features, including error handling, logging, refactoring parts of the code that did not meet standards, unit tests, and verification and validation of the application we developed. Additionally, I played a significant role in the development of Gherkin specifications, ensuring precision and clarity in the project's execution. I have attached screenshots showcasing my individual contributions to the team project, demonstrating my commitment and dedication.

In team meetings, I actively participated, offering valuable suggestions and solutions to challenges faced. Collaboration was at the core of our team dynamics, and I eagerly worked alongside my teammates, engaging in code reviews to maintain code quality and foster a spirit of collective responsibility.

The team acknowledged and appreciated my contributions, and I strived to maintain a positive and collaborative attitude throughout the project. Recognizing the challenges arising from uneven task distribution, I took the initiative to address the issue promptly, leading to a fairer allocation of responsibilities, thus elevating team morale and productivity.

It has been an invaluable journey of growth and discovery. It has transformed my perspective on project management methodologies and emphasized the significance of effective communication and collaboration within a team. I am eager to apply this newfound knowledge in future endeavors, both professionally and academically, while continuously seeking opportunities to expand my skillset. This experience has equipped me with the tools to thrive in dynamic project environments and has bolstered my confidence in tackling new challenges with enthusiasm and determination.

**Conclusion**

In conclusion, this module has been a transformative and enriching experience, taking me on a rollercoaster ride of emotions and personal growth. although, I embarked on this course with excitement and enthusiasm, it was not enough to get me through. I encountered challenges that tested my ability to manage time effectively and strike a balance between various aspects of my life.

In retrospect, I drew upon the lessons learned from the stress management module earlier in the course and experiences from previous modules, which helped me view stress as a positive factor that propels growth and achievement. By embracing challenges as opportunities for development, I shifted my focus from regrets to maximizing my contributions and participation in the course.

Working in a collaborative and supportive team was a significant aspect of this project. Despite encountering hurdles when one team member left. Throughout the project, I have acquired valuable insights into project management methodologies, particularly the significance of tailored approaches for specific environments. The exposure to both waterfall and agile methodologies allowed me to discern the merits of each and apply the most suitable practices to different projects

Moving forward, I am looking at ways to maximize the impact of Project Management in Virtual environments or environments like my workplace where a team is distributed over several time zones and countries and the team is mostly left to work virtually.  Am wondering;

1. what strategies can we implement to foster open communication and collaboration within the team, especially in virtual or remote settings?
2. How can we effectively manage time and prioritize tasks to strike a balance between academic, professional, and personal commitments?

3. What steps can we take to encourage continuous learning and professional development?
4. How can we promote a culture of innovation and creativity within the team to drive unique and impactful project solutions?

It is my desire to conduct a few personal research to find practical answers to the above questions. Hopefully, am able to try some of my findings at work to see the outcomes.

This journey of self-discovery, resilience, and growth has been a valuable experience. I have emerged with a refined perspective on project management and a deeper understanding of my capabilities as a software engineer. I am grateful for the unwavering support from my tutor, family, colleagues, and teammates throughout this module. This module has been a pivotal milestone in my academic journey, and I look forward to leveraging these learnings to contribute more effectively to my career and make a meaningful impact in the software engineering industry.

**References**

Sutherland, J., & Schwaber, K. (2007). The Scrum Guide. Scrum.org. https://www.scrum.org/resources/scrum-guide

Essex University Online (n.d.), Induction Computing October 2021 A

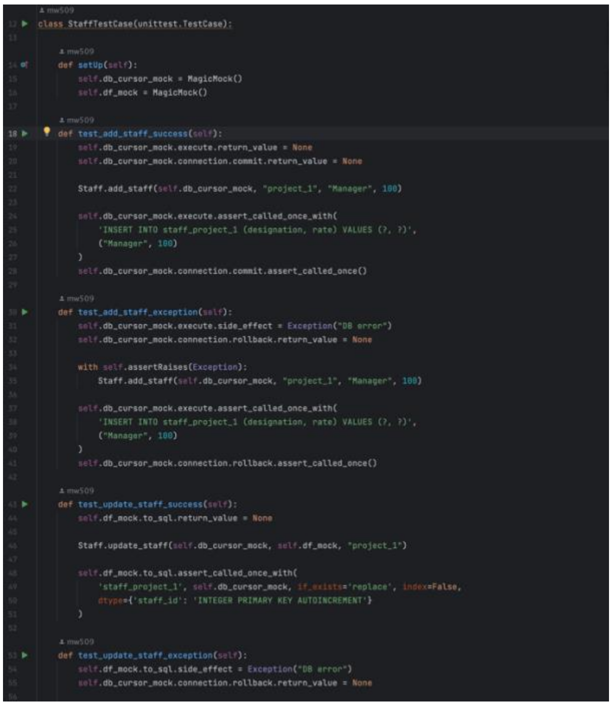
Lehtinen, A., Mäntylä, V., Vanhanen, J., Itkonen, J. & Lassenius, C. (2014) Perceived causes of software project failures – An analysis of their relationships. Information and Software Technology 56(6): 623–643.

Goatham, R. (2020) Why Projects Fail.

Behave Project. (n.d.). Gherkin Introduction. https://behave.readthedocs.io/en/stable/philosophy.html

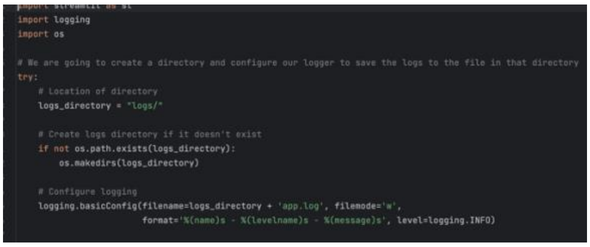
The Cucumber Team. (n.d.). The Gherkin Reference. https://cucumber.io/docs/gherkin/reference/

**Screenshots**



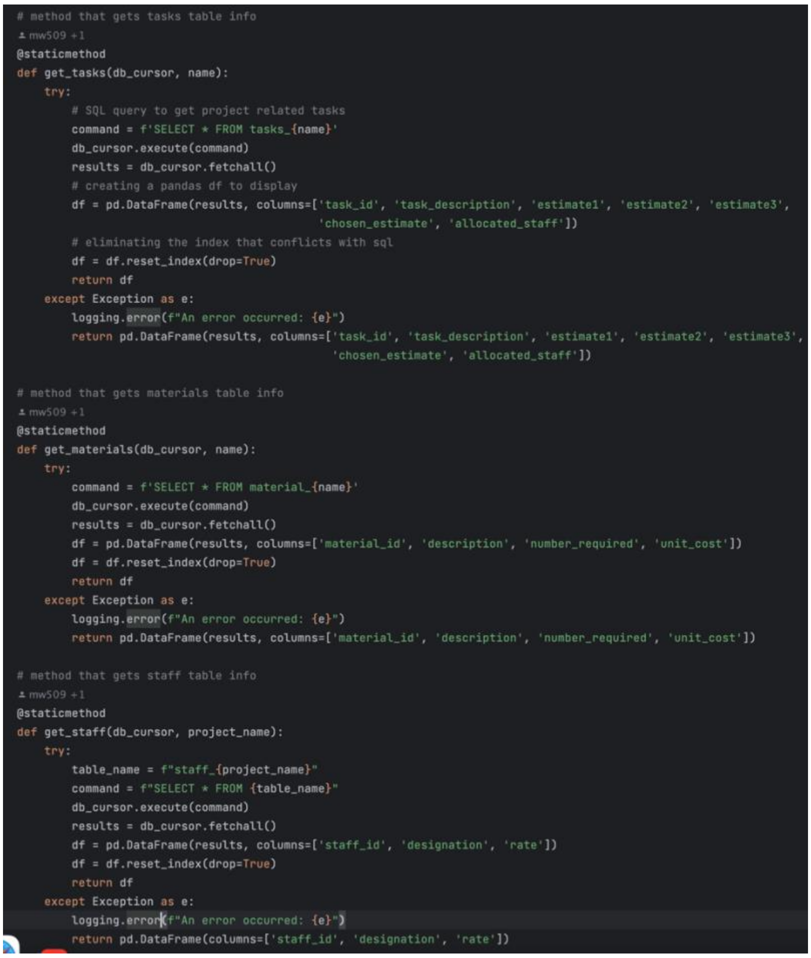
```
mw509
class StaffTestCase(unittest.TestCase):

    mw509
    def setUp(self):
        self.db_cursor_mock = MagicMock()
        self.df_mock = MagicMock()

    mw509
    def test_add_staff_success(self):
        self.db_cursor_mock.execute.return_value = None
        self.db_cursor_mock.connection.commit.return_value = None

        Staff.add_staff(self.db_cursor_mock, "project_1", "Manager", 100)

        self.db_cursor_mock.execute.assert_called_once_with(
            'INSERT INTO staff_project_1 (designation, rate) VALUES (?, ?)',
            ("Manager", 100)
        )
        self.db_cursor_mock.connection.commit.assert_called_once()

    mw509
    def test_add_staff_exception(self):
        self.db_cursor_mock.execute.side_effect = Exception("DB error")
        self.db_cursor_mock.connection.rollback.return_value = None

        with self.assertRaises(Exception):
            Staff.add_staff(self.db_cursor_mock, "project_1", "Manager", 100)

        self.db_cursor_mock.execute.assert_called_once_with(
            'INSERT INTO staff_project_1 (designation, rate) VALUES (?, ?)',
            ("Manager", 100)
        )
        self.db_cursor_mock.connection.rollback.assert_called_once()

    mw509
    def test_update_staff_success(self):
        self.df_mock.to_sql.return_value = None

        Staff.update_staff(self.db_cursor_mock, self.df_mock, "project_1")

        self.df_mock.to_sql.assert_called_once_with(
            'staff_project_1', self.db_cursor_mock, if_exists='replace', index=False,
            dtype={'staff_id': 'INTEGER PRIMARY KEY AUTOINCREMENT'}
        )

    mw509
    def test_update_staff_exception(self):
        self.df_mock.to_sql.side_effect = Exception("DB error")
        self.db_cursor_mock.connection.rollback.return_value = None
```

Screenshot of some unit tests I wrote.



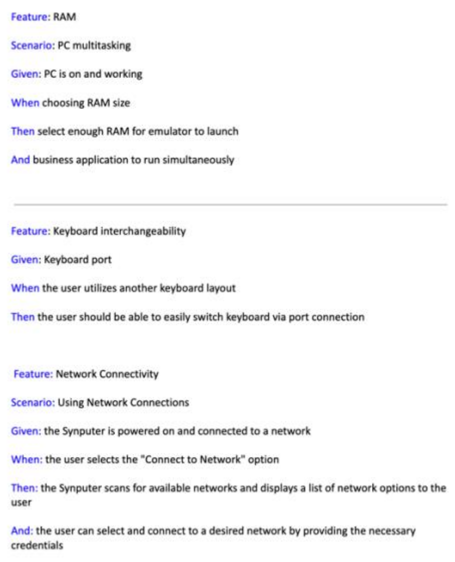
```
import streamlit as st
import logging
import os

# We are going to create a directory and configure our logger to save the logs to the file in that directory
try:
    # Location of directory
    logs_directory = "logs/"

    # Create logs directory if it doesn't exist
    if not os.path.exists(logs_directory):
        os.makedirs(logs_directory)

    # Configure logging
    logging.basicConfig(filename=logs_directory + 'app.log', filemode='w',
                        format='%(name)s - %(levelname)s - %(message)s', level=logging.INFO)
```

Screenshot of a refactoring done for the logging configuration.



```
# method that gets tasks table info
mw509 +1
@staticmethod
def get_tasks(db_cursor, name):
    try:
        # SQL query to get project related tasks
        command = f'SELECT * FROM tasks_{name}'
        db_cursor.execute(command)
        results = db_cursor.fetchall()
        # creating a pandas df to display
        df = pd.DataFrame(results, columns=['task_id', 'task_description', 'estimate1', 'estimate2', 'estimate3',
                                            'chosen_estimate', 'allocated_staff'])
        # eliminating the index that conflicts with sql
        df = df.reset_index(drop=True)
        return df
    except Exception as e:
        logging.error(f"An error occurred: {e}")
        return pd.DataFrame(results, columns=['task_id', 'task_description', 'estimate1', 'estimate2', 'estimate3',
                                              'chosen_estimate', 'allocated_staff'])

# method that gets materials table info
mw509 +1
@staticmethod
def get_materials(db_cursor, name):
    try:
        command = f'SELECT * FROM material_{name}'
        db_cursor.execute(command)
        results = db_cursor.fetchall()
        df = pd.DataFrame(results, columns=['material_id', 'description', 'number_required', 'unit_cost'])
        df = df.reset_index(drop=True)
        return df
    except Exception as e:
        logging.error(f"An error occurred: {e}")
        return pd.DataFrame(results, columns=['material_id', 'description', 'number_required', 'unit_cost'])

# method that gets staff table info
mw509 +1
@staticmethod
def get_staff(db_cursor, project_name):
    try:
        table_name = f"staff_{project_name}"
        command = f"SELECT * FROM {table_name}"
        db_cursor.execute(command)
        results = db_cursor.fetchall()
        df = pd.DataFrame(results, columns=['staff_id', 'designation', 'rate'])
        df = df.reset_index(drop=True)
        return df
    except Exception as e:
        logging.error(f"An error occurred: {e}")
        return pd.DataFrame(columns=['staff_id', 'designation', 'rate'])
```

Refactoring done for error handling and database handling.



Feature: RAM

Scenario: PC multitasking

Given: PC is on and working

When choosing RAM size

Then select enough RAM for emulator to launch

And business application to run simultaneously

Feature: Keyboard interchangeability

Given: Keyboard port

When the user utilizes another keyboard layout

Then the user should be able to easily switch keyboard via port connection

Feature: Network Connectivity

Scenario: Using Network Connections

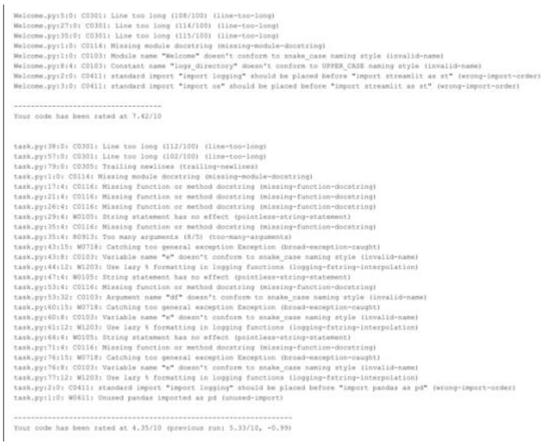
Given: the Synputer is powered on and connected to a network

When: the user selects the "Connect to Network" option

Then: the Synputer scans for available networks and displays a list of network options to the user

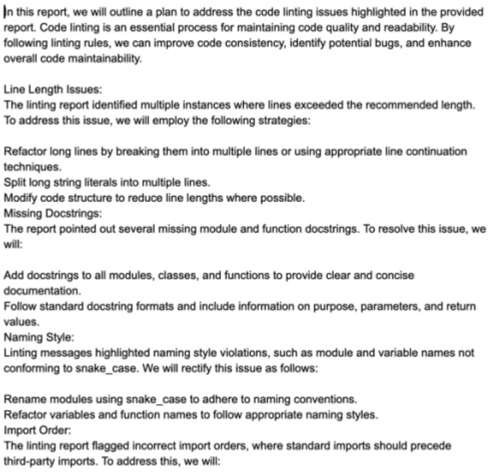
And: the user can select and connect to a desired network by providing the necessary credentials

Screenshot of some Gherkin Statements contributed to assignment 1.



```
Welcome.py:5:0: C0301: Line too long (108/100) (line-too-long)
Welcome.py:27:0: C0301: Line too long (114/100) (line-too-long)
Welcome.py:35:0: C0301: Line too long (115/100) (line-too-long)
Welcome.py:1:0: C0114: Missing module docstring (missing-module-docstring)
Welcome.py:1:0: C0103: Module name "Welcome" doesn't conform to snake_case naming style (invalid-name)
Welcome.py:8:4: C0103: Constant name "logs_directory" doesn't conform to UPPER_CASE naming style (invalid-name)
Welcome.py:2:0: C0411: standard import "import logging" should be placed before "import streamlit as st" (wrong-import-order)
Welcome.py:3:0: C0411: standard import "import os" should be placed before "import streamlit as st" (wrong-import-order)

-----------------------------------
Your code has been rated at 7.42/10


task.py:38:0: C0301: Line too long (112/100) (line-too-long)
task.py:57:0: C0301: Line too long (102/100) (line-too-long)
task.py:79:0: C0305: Trailing newlines (trailing-newlines)
task.py:1:0: C0114: Missing module docstring (missing-module-docstring)
task.py:17:4: C0116: Missing function or method docstring (missing-function-docstring)
task.py:21:4: C0116: Missing function or method docstring (missing-function-docstring)
task.py:26:4: C0116: Missing function or method docstring (missing-function-docstring)
task.py:29:4: W0105: String statement has no effect (pointless-string-statement)
task.py:35:4: C0116: Missing function or method docstring (missing-function-docstring)
task.py:35:4: R0913: Too many arguments (8/5) (too-many-arguments)
task.py:43:15: W0718: Catching too general exception Exception (broad-exception-caught)
task.py:43:8: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
task.py:44:12: W1203: Use lazy % formatting in logging functions (logging-fstring-interpolation)
task.py:47:4: W0105: String statement has no effect (pointless-string-statement)
task.py:53:4: C0116: Missing function or method docstring (missing-function-docstring)
task.py:53:32: C0103: Argument name "df" doesn't conform to snake_case naming style (invalid-name)
task.py:60:15: W0718: Catching too general exception Exception (broad-exception-caught)
task.py:60:8: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
task.py:61:12: W1203: Use lazy % formatting in logging functions (logging-fstring-interpolation)
task.py:64:4: W0105: String statement has no effect (pointless-string-statement)
task.py:71:4: C0116: Missing function or method docstring (missing-function-docstring)
task.py:76:15: W0718: Catching too general exception Exception (broad-exception-caught)
task.py:76:8: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
task.py:77:12: W1203: Use lazy % formatting in logging functions (logging-fstring-interpolation)
task.py:2:0: C0411: standard import "import logging" should be placed before "import pandas as pd" (wrong-import-order)
task.py:1:0: W0611: Unused pandas imported as pd (unused-import)

-----------------------------------------------------------------
Your code has been rated at 4.35/10 (previous run: 5.33/10, -0.99)
```

Screenshot of a lint report generated using PYLint.



In this report, we will outline a plan to address the code linting issues highlighted in the provided report. Code linting is an essential process for maintaining code quality and readability. By following linting rules, we can improve code consistency, identify potential bugs, and enhance overall code maintainability.

Line Length Issues:
The linting report identified multiple instances where lines exceeded the recommended length. To address this issue, we will employ the following strategies:

Refactor long lines by breaking them into multiple lines or using appropriate line continuation techniques.
Split long string literals into multiple lines.
Modify code structure to reduce line lengths where possible.
Missing Docstrings:
The report pointed out several missing module and function docstrings. To resolve this issue, we will:

Add docstrings to all modules, classes, and functions to provide clear and concise documentation.
Follow standard docstring formats and include information on purpose, parameters, and return values.
Naming Style:
Linting messages highlighted naming style violations, such as module and variable names not conforming to snake_case. We will rectify this issue as follows:

Rename modules using snake_case to adhere to naming conventions.
Refactor variables and function names to follow appropriate naming styles.
Import Order:
The linting report flagged incorrect import orders, where standard imports should precede third-party imports. To address this, we will:

Report on how to fix the linting error reports.