Team Contributions: Rev 0
Software Engineering

Team 4, EcoOptimizers

Nivetha Kuruparan
Sevhena Walker
Tanveer Brar
Mya Hussain
Ayushi Amin

This document summarizes the contributions of each team member for the Rev 0 Demo. The time period of interest is the time between the POC demo and the Rev 0 demo.

1 Demo Plans

For our **Rev 0 demonstration**, we will showcase the core functionality and usability of our **energy-efficient Python code refactoring tool**. This demonstration will highlight the following key aspects:

1. **Code Smell Detection:** We will demonstrate how our tool identifies inefficient coding patterns (*code smells*) in Python source code that contribute to higher energy consumption.
2. **Automated Refactoring:** Our tool will apply targeted refactorings to optimize the detected code smells, improving energy efficiency.
3. **Energy Consumption Measurement:** Using `CodeCarbon`, we will measure and compare the energy consumption of the original code versus the refactored version, providing clear insights into energy savings.
4. **Functionality Preservation:** To ensure that optimizations do not alter program behaviour, we will run the original test suite against both the unoptimized and refactored versions.
5. **VS Code Integration:** We will showcase our **VS Code plugin**, where users can analyze a Python file for code smells, choose specific optimizations, and preview the refactored code with energy measurements. Users will have the option to accept or reject the proposed changes.

2 Team Meeting Attendance

Our team stays in touch regularly through Discord and schedules a meeting when detailed discussion is needed. Below is the attendance of all format meetings that have taken place.

Student	**Meetings**
Total	3
Sevhena Walker	3
Mya Hussain	3
Ayushi Amin	3
Nivetha Kuruparan	3
Tanveer Brar	3

3 Supervisor/Stakeholder Meeting Attendance

We prioritize having in person meetings with our supervisor Dr David to be able to effectively communicate our progress and concerns. We have a weekly slot open with the supervisor, but in case there is little to discuss, the meeting is cancelled for the week and our progress is communicated via Discord.

Student	**Meetings**
Total	4
Sevhena Walker	4
Mya Hussain	4
Ayushi Amin	4
Nivetha Kuruparan	4
Tanveer Brar	4

4 Lecture Attendance

We aim to have at least one person attend the lecture so the team is upto date with all information. In the rare case that no one goes to the lecture, we update through the lecture slides.

Student	Lectures
Total	2
Sevhena Walker	1
Mya Hussain	0
Ayushi Amin	0
Nivetha Kuruparan	0
Tanveer Brar	0

5 TA Document Discussion Attendance

Student	Lectures
Total	2
Sevhena Walker	2
Mya Hussain	2
Ayushi Amin	2
Nivetha Kuruparan	2
Tanveer Brar	2

6 Commits

Student	Commits	Percent
Total	662	100%
Sevhena Walker	277	41.8429%
Mya Hussain	78	11.782485%
Ayushi Amin	129	19.4864%
Nivetha Kuruparan	102	15.407855%
Tanveer Brar	76	11.48036%

The number of commits does not necessarily reflect the amount of work done by each team member. Some members prefer to make frequent small commits while others make fewer but larger commits that include multiple changes.

7 Issue Tracker

Student	Authored (O+C)	Assigned (C only)
Tanveer Brar	1	47
Mya Hussain	8	51
Ayushi Amin	8	58
Sevhena Walker	81	72
Nivetha Kuruparan	17	53

Counts are considered by taking the total counts over the year and subtracting the counts at the time of the POC Team Contribution. This is the period of intrest. We did a lot of work over the winter break to exclude that period and only do "this term" would inaccurately represent the counts.

8 CICD

CI/CD was implemented in the project to automate essential development and documentation tasks, improving efficiency and maintaining code quality. The project utilises GitHub Actions workflows to enforce coding standards, validate changes, and ensure documentation integrity.

A dedicated workflow formats and compiles LaTeX documents whenever changes are made to `.tex` files or their dependencies. This ensures that only high quality documentation is pushed to the main branch.

Another workflow is responsible for maintaining code consistency and adhering to best practices. It uses `ruff` to lint and format `.py` files, identifying potential issues such as syntax errors, unused imports, and style inconsistencies. This workflow allows the python scripts to be written with a uniform coding style.

The final workflow ensures the correctness of the codebase. It runs `pytest` to execute unit and integration tests, validating the expected behaviour of the software. Additionally, it includes a code coverage check, ensuring that newly introduced or modified code is adequately tested. This helps prevent regressions and maintain the reliability of the project.