

13-06-2023

Department of Applied
Informatics, University of
Macedonia

ADVISOR

Alexander
Chatzigeorgiou

STUDENT

George - David
Apostolidis
iis20001

Evaluation of Python code quality using multiple source code analyzers



Theoretical Background

I Theoretical Background

- **Python: Guido van Rossum, 1991**
- **Beginner-Friendly and Easy to Learn**
- **Rich Library Ecosystem**
- **Powerful and Versatile**



I Theoretical Background

- **Importance of Software Quality**

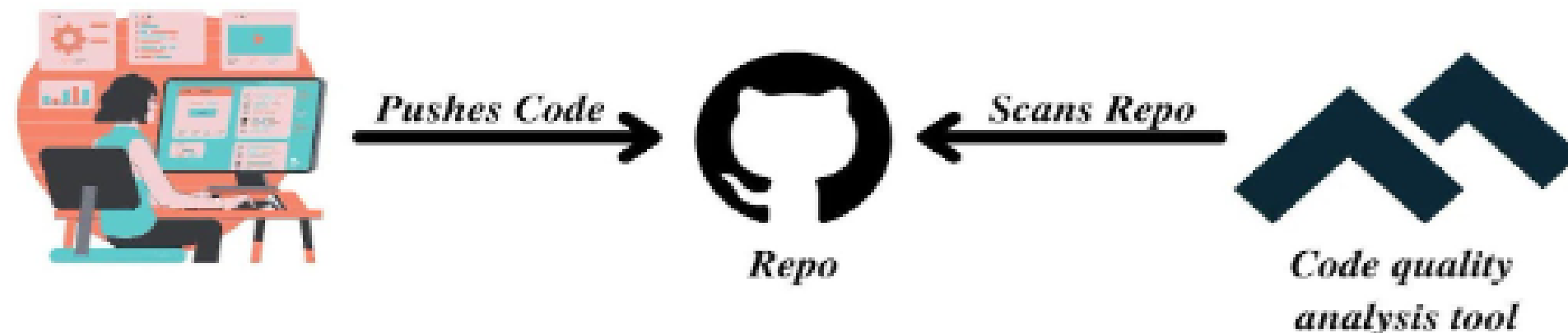
- Crucial for developers and customers
- Ensures a common understanding and effective measurement of quality.

- **Quality Model for Evaluation**

- ISO/IEC 25010 : functionality, performance, security, maintainability, and others.

- **Key Aspects of Software Quality**

- Readability, maintainability, modularity, performance, and security
 - **Static Code Analysis**
- Linters
- Quality gates in CI environments, where code analysis is performed on pull/merge requests.



I Theoretical Background

- **Code Quality Applications**

- SonarQube
- Crucible
- ESLint & JSHint

- **Linters**

- Flake8
- Pylama
- ESLint & JSHint

- **Test Coverage**

- unittest
- Pytest

Python Linters	Category
Pylint	Logical & Stylistic
PyFlakes	Logical
Pycodestyle	Stylistic
Pydocstyle	Stylistic
Bandit	Logical
MyPy	Logical

I Theoretical Background

• Related Work

- **code2graph**

- Automates the extraction of the structure of Python source code
- Constructs static call graphs
- Creates a similarity matrix for execution paths.

- **PyTA**

- Wrapper
- Designed to help novice developers find and fix common errors in their code.

- **Scalpel Framework**

- Pre-built tools for fundamental static analysis tasks
- Call graph construction
- Flow control graph construction
- Alias analysis.



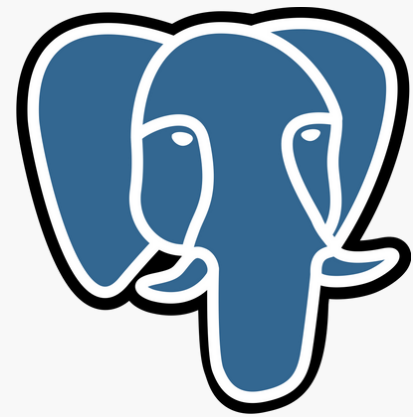
II

Technology Stack

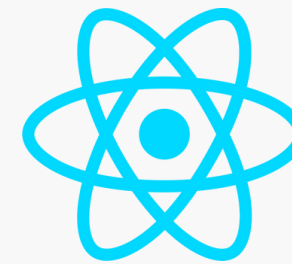
II Technology Stack

PyAssess

Backend



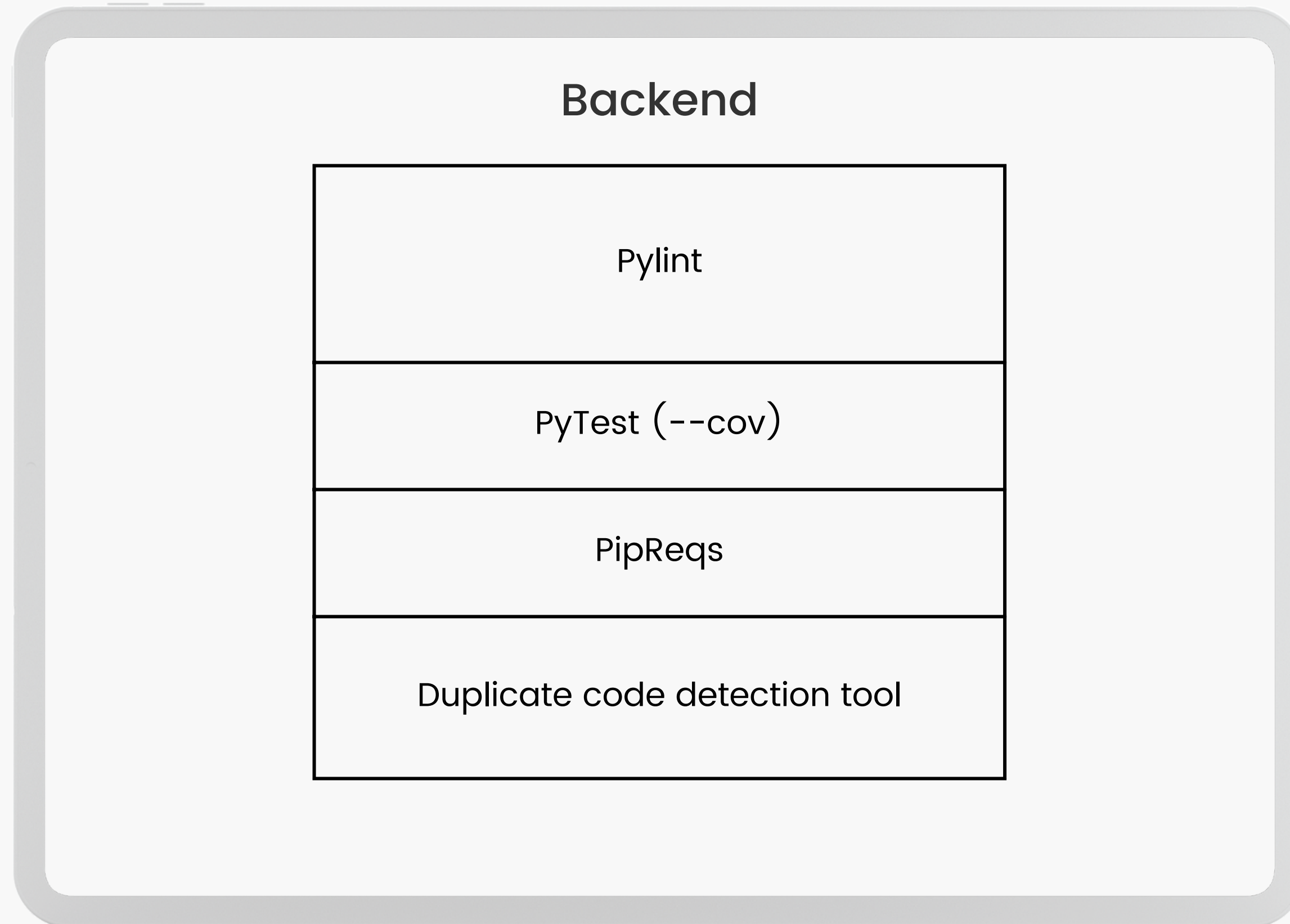
Frontend



React



II Technology Stack





App Functionality / Presentation

III App Functionality / Presentation

EOSC RAISE

Provide a Github URL

Enter a Github URL to start the analysis of a project.

Start Analysis **Gitlab CI** **Show data**

Project Data			
Git URL	Owner	Name	SHAs
https://github.com/jerempa/PDF-OCR-and-data-analysis.git	jerempa	PDF-OCR-and-data-analysis	<ul style="list-style-type: none"> - 6f3882fb2e44ac93910d63baaeda503d50f0b804 - 326c060dcb51db378d6f62b9b9404201331d44d7 - 76ed80b7eb5e0c50ad0ac78223b797cddb86563e - 455e2fc44b34066819860d1c0cce44d4de0528ed - e775c7b1c145ea352c7e172bc99395fb7c7abfe - aeceef942bb60bdcc80b4d920caa4d2a233136 - 7acc10c7ccb75ede252cc47edde11361b548a10e - 98decceb0a5111008e3e05071f90959334b6adaa - 84d0b7a1c07126077b16855b69bfab85647ace31 - 2127623594307c5ae9b42a34eb7eafbd04a22afe

Total Statements

totalStmts

Dependencies

dependenciesCounter

Total Miss Chart

totalMiss

Total Coverage Chart

totalCoverage

III App Functionality / Presentation

```
build-job: # This job runs in the build stage, which runs first.
  stage: build
  script:
    - echo "Build stage..."
    - export GIT_URL=$CI_PROJECT_URL
    - export GIT_BRANCH=$CI_COMMIT_BRANCH
    - 'curl --request POST "http://195.251.210.147:8181/project_analysis/?
gitUrl=$GIT_URL&branch=$GIT_BRANCH'
```

Back

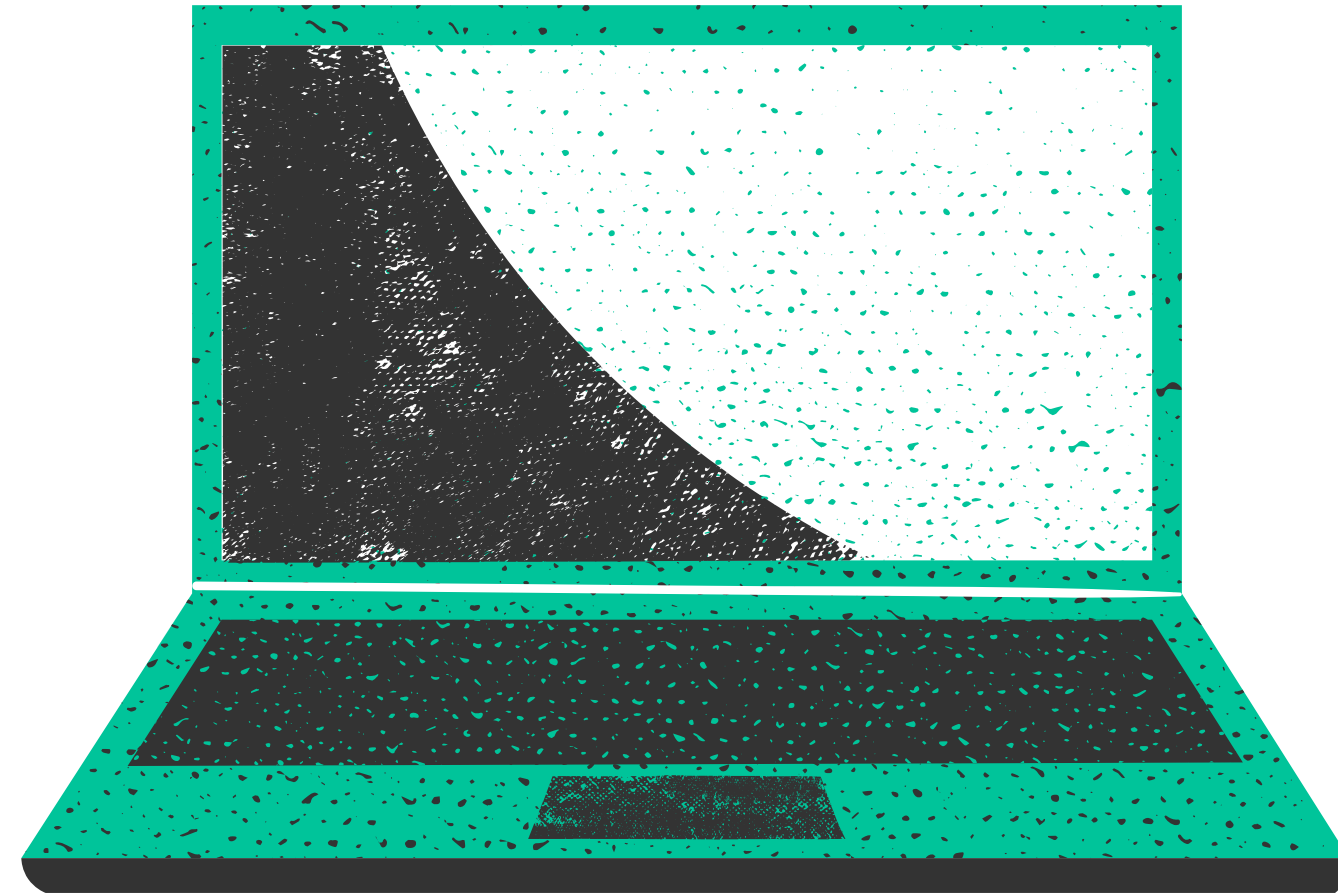
III App Functionality / Presentation

```
1  {
2    "id": 6,
3    "gitUrl": "https://github.com/jerempa/PDF-OCR-and-data-analysis.git",
4    "owner": "jerempa",
5    "name": "PDF-OCR-and-data-analysis",
6    "directory": "C:\\Files\\Univeristy\\Service-Based-Assessment-of-Python-",
7  > "projectAnalysis": [ ...
132695 ],
132696 "singleAnalyzedProjectList": [],
132697 "sha": [
132698   "6f3882fb2e44ac93910d63baaeda503d50f0b804",
132699   "326c060dcb51db378d6f62b9b9404201331d44d7",
132700   "76ed80b7eb5e0c50ad0ac78223b797cdbc86563e",
132701   "455e2fc44b34066819860d1c0cce44d4de0528ed",
132702   "e775c7b1c145ea352c7e172bc99395fbb7c7abfe",
132703   "aeceef942bb60bdcc80b4d920caa4df2a233136",
132704   "7acc10c7ccb75ede252cc47edde11361b548a10e",
132705   "98decceb0a5111008e3e05071f90959334b6adaa",
132706   "84d0b7a1c07126077b16855b69bfab85647ace31",
132707   "2127623594307c5ae9b42a34eb7eafbd04a22afe"
132708 ]
132709 }
```

```
"id": 27,
"gitUrl": "C:\\Files\\Univeristy\\Service-Based-Asses
"owner": "jerempa",
"name": "PDF-OCR-and-data-analysis",
"directory": "C:\\Files\\Univeristy\\Service-Based-As
"dependencies": [ ...
],
"dependenciesCounter": 11,
"files": [ ...
],
"totalCoverage": 0,
"totalMiss": 1606,
"totalStmts": 1606,
"sha": "6f3882fb2e44ac93910d63baaeda503d50f0b804"
```

III App Functionality / Presentation

```
28     "files": [  
29         {  
30             "id": 282,  
31             "name": "correct_seasons.py",  
32             "stmts": 20,  
33             "miss": 20,  
34             "coverage": 0,  
35             "comments": [...  
72         ],  
73         "rating": 4.0,  
74         "previousRating": null,  
75         "similarity": {  
76             "img_to_string.py": 14.0,  
77             "debt_visualization.py": 7.55,  
78             "extracting_data.py": 8.19,  
79             "values_for_analysis.py": 10.07,  
80             "transfermarkt_data_visualization.py": 6.92,  
81             "fetch_data_from_transfermarkt.py": 9.36,  
82             "download_pdfs.py": 9.12,  
83             "df_operations.py": 3.84,  
84             "data_visualization.py": 9.09,  
85             "img_conversion_and_processing.py": 4.76,  
86             "fetch_data_from_worldfootball.py": 11.8,  
87             "main.py": 9.09,  
88             "fetch_season_data_from_wiki.py": 7.92,  
89             "calculations.py": 6.99,  
90             "errors.py": 5.26,  
91             "__init__.py": 1.1,  
92             "financial_statement_data_visualization.py": 9.38,  
93             "file_handling.py": 10.38  
94         },  
95         "projectName": "PDF-OCR-and-data-analysis",
```



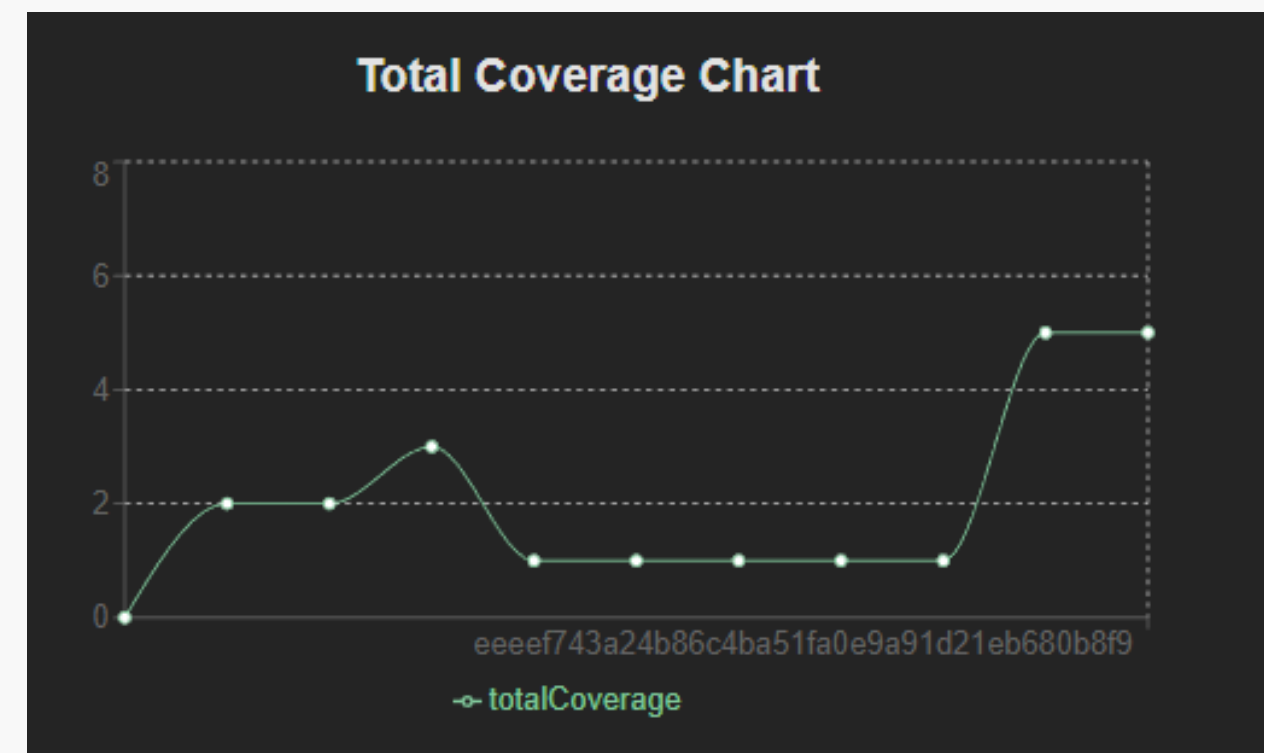
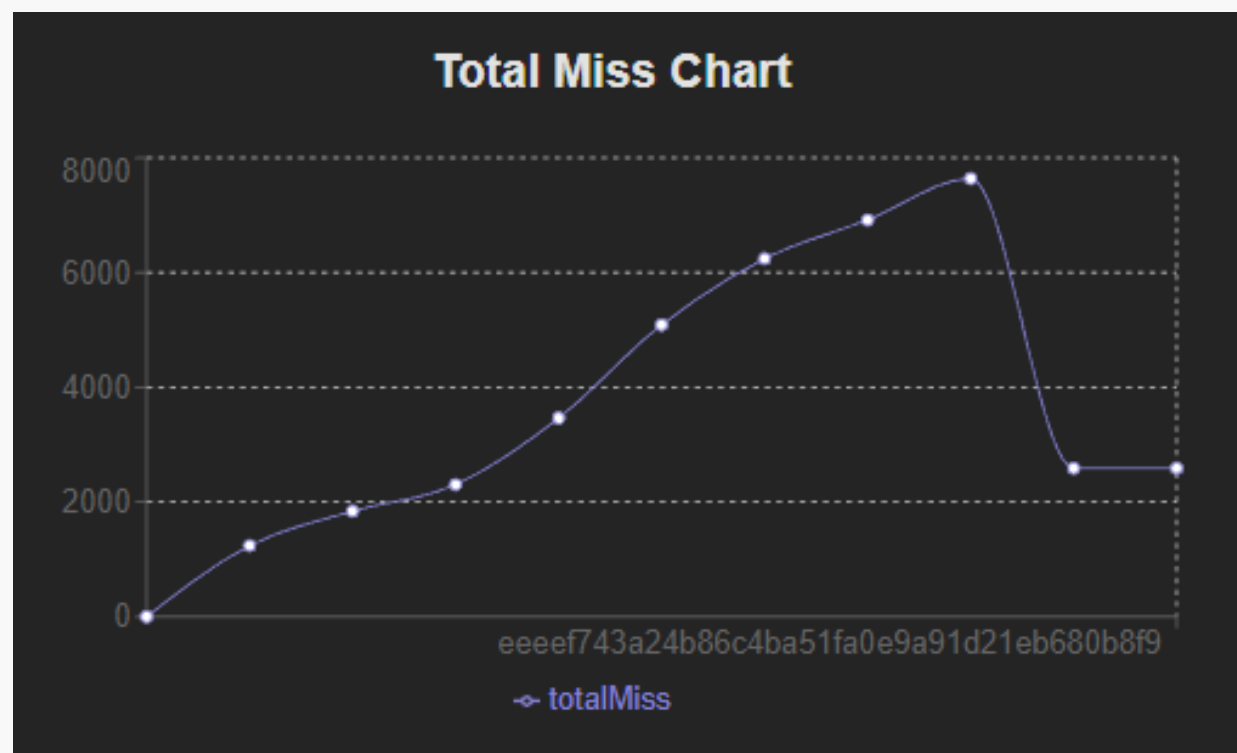
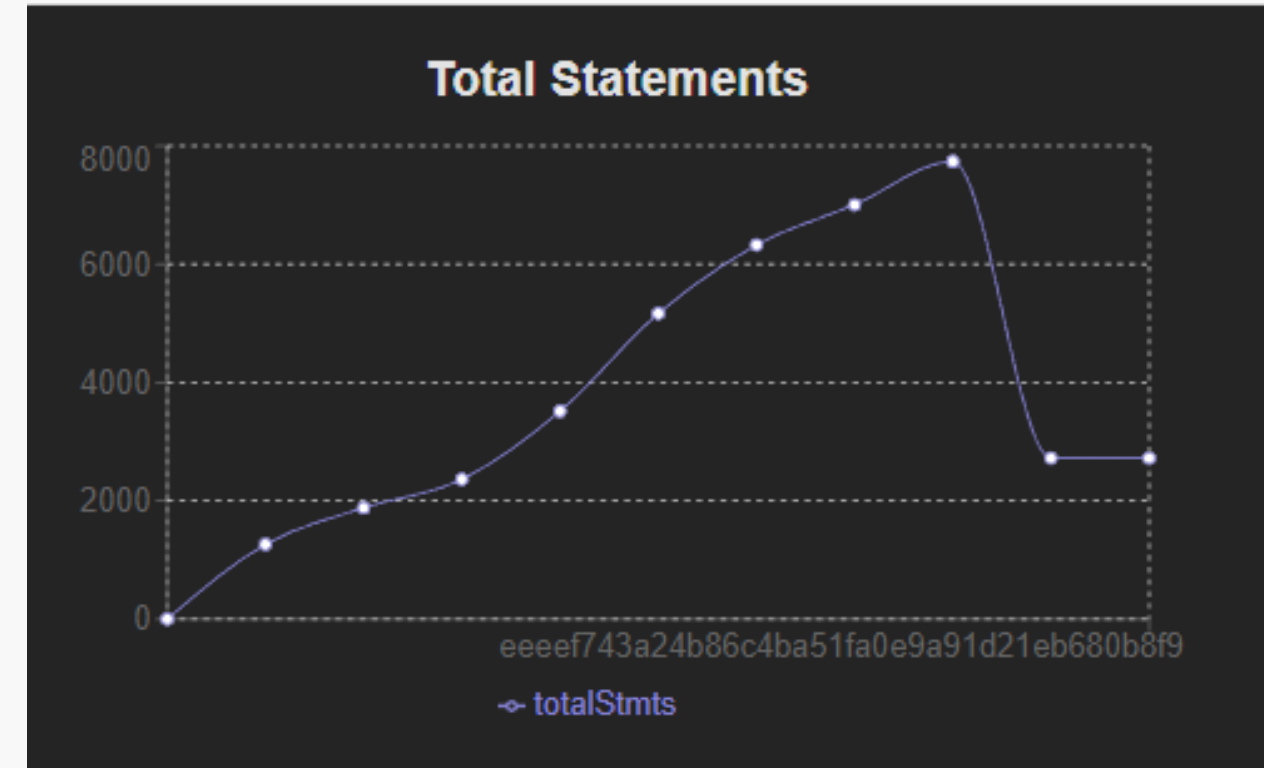
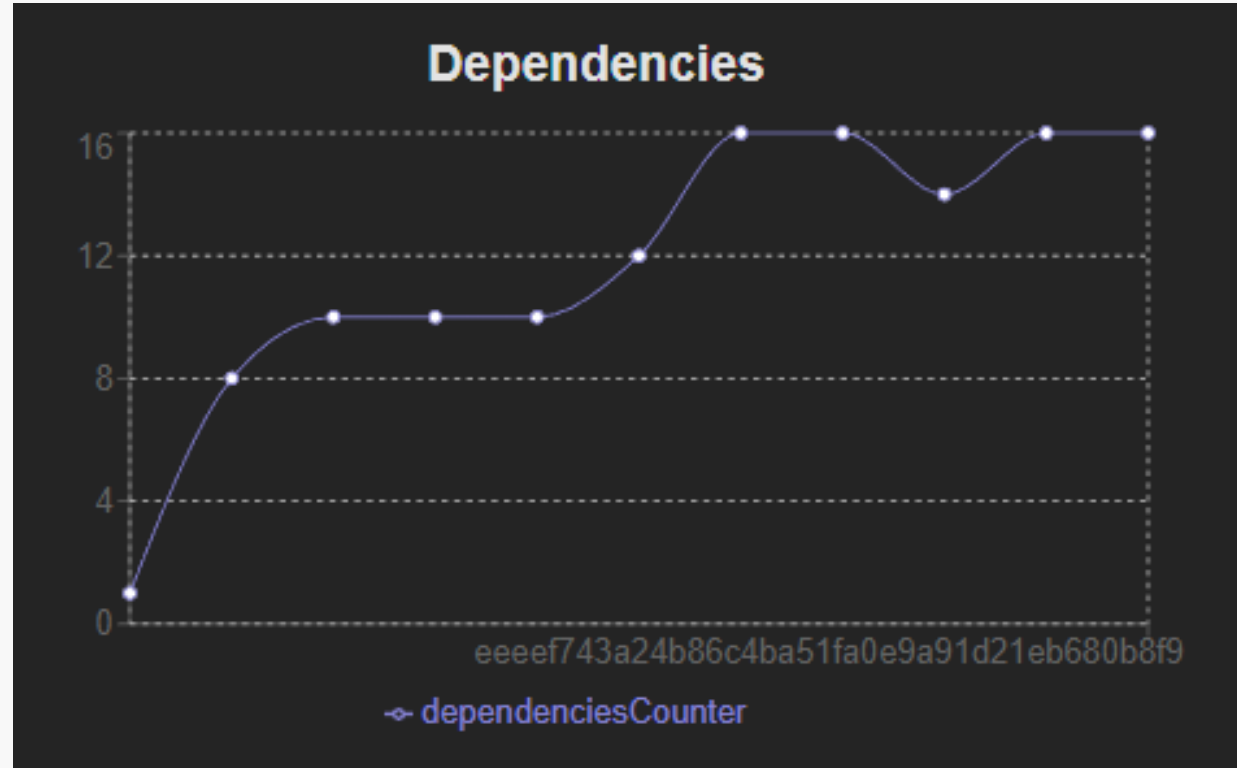
Application Demo

IV

Results

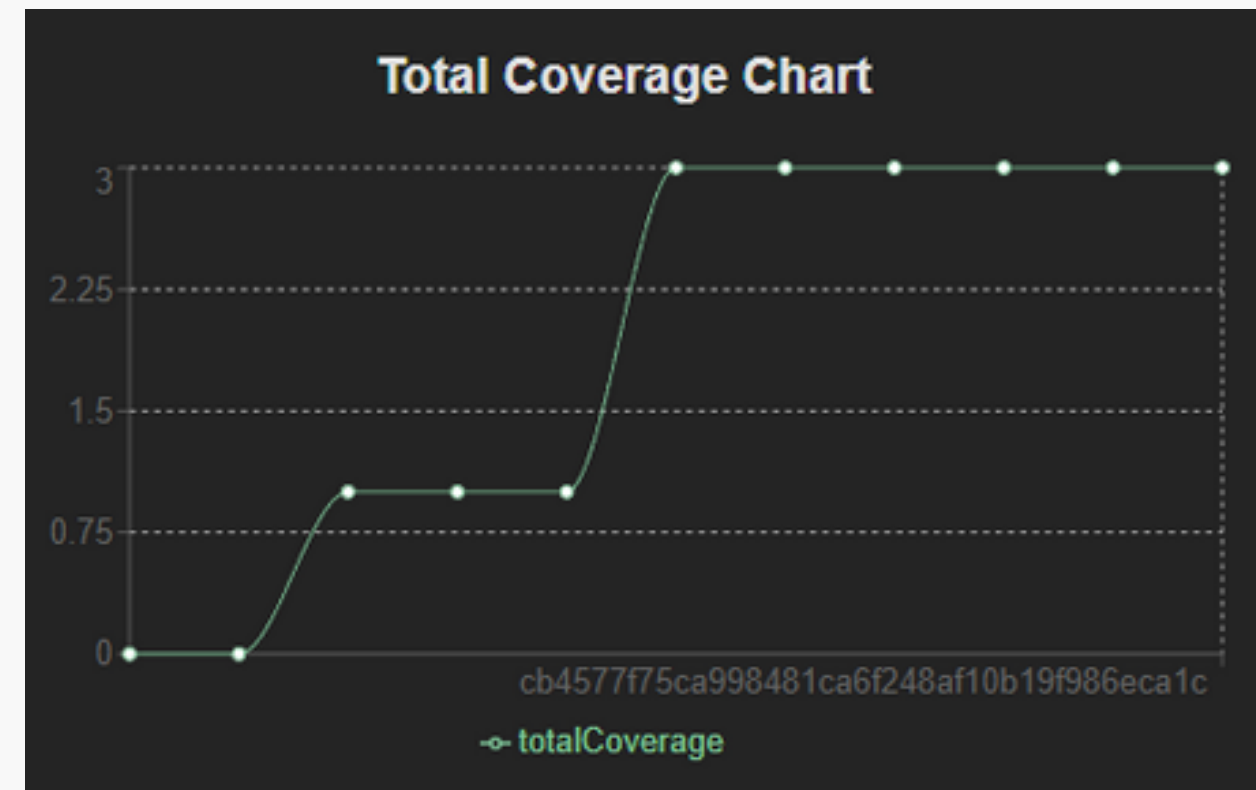
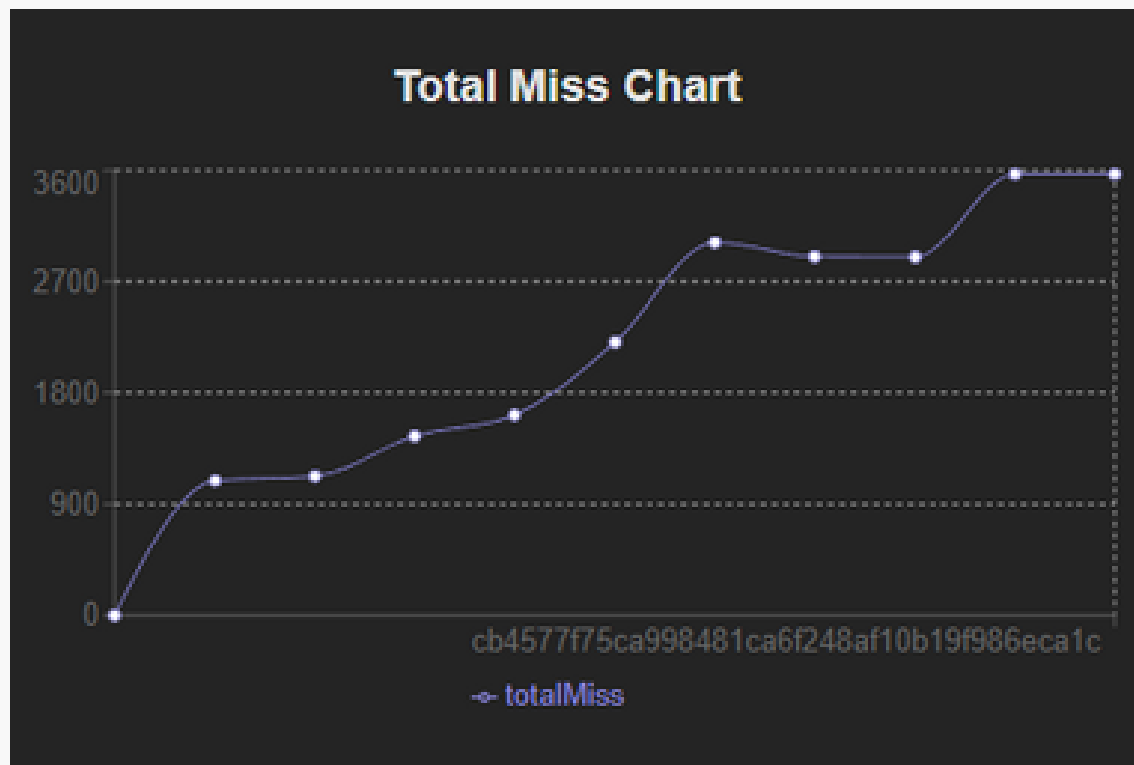
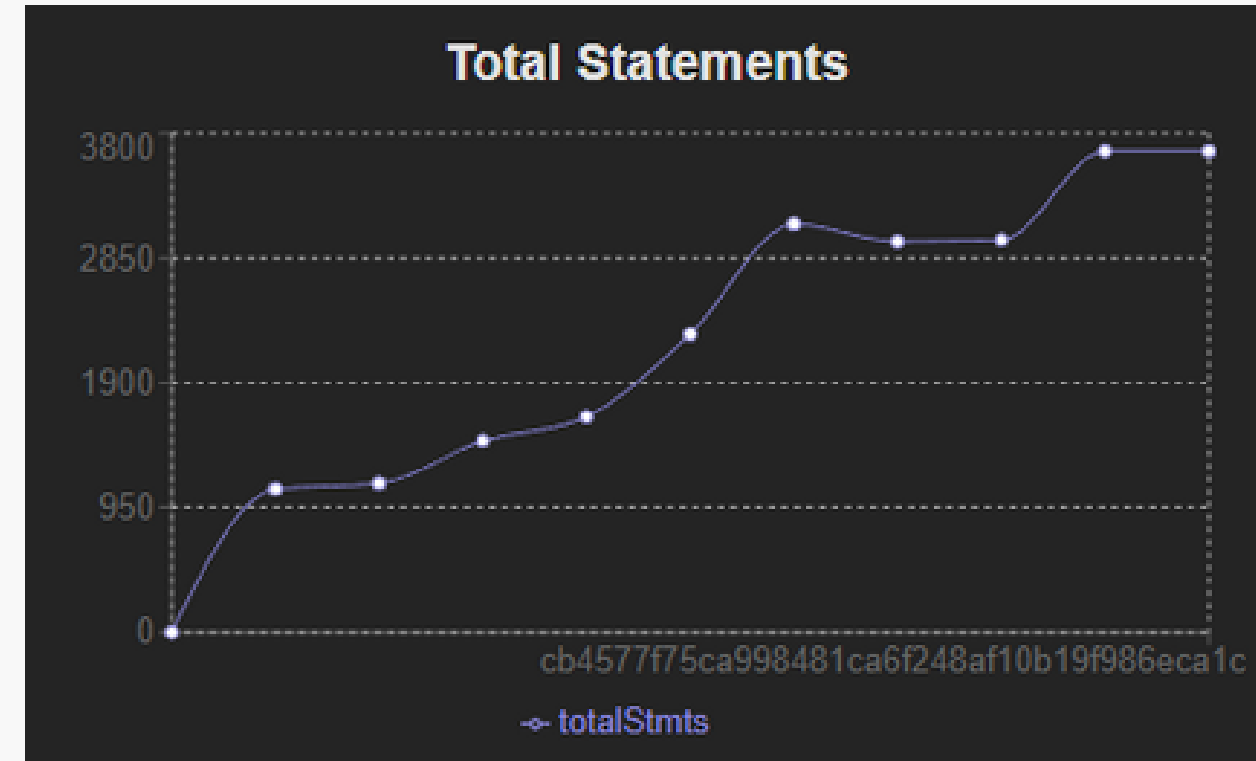
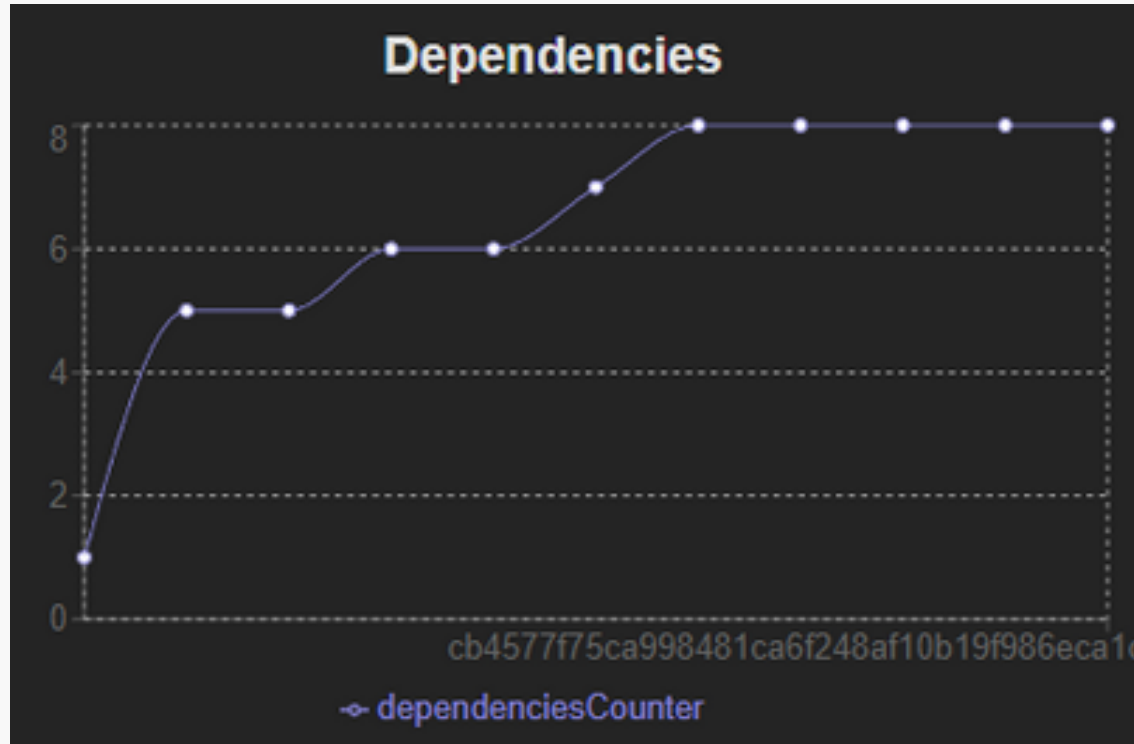
IV Results

TorchIO



IV Results

DPPy



V

Conclusion & Discussions



- **Suitable for automating the evaluation process**

Metrics and visualizations can aid in identifying areas of improvement for software quality.

- **Code quality can vary significantly**

Projects require more attention to improve quality

- **Significant contribution to the field of software development**

Reliable and efficient method for evaluating code quality.



V Conclusion & Discussions



Additional Programming Languages

Broader range of projects to be analyzed

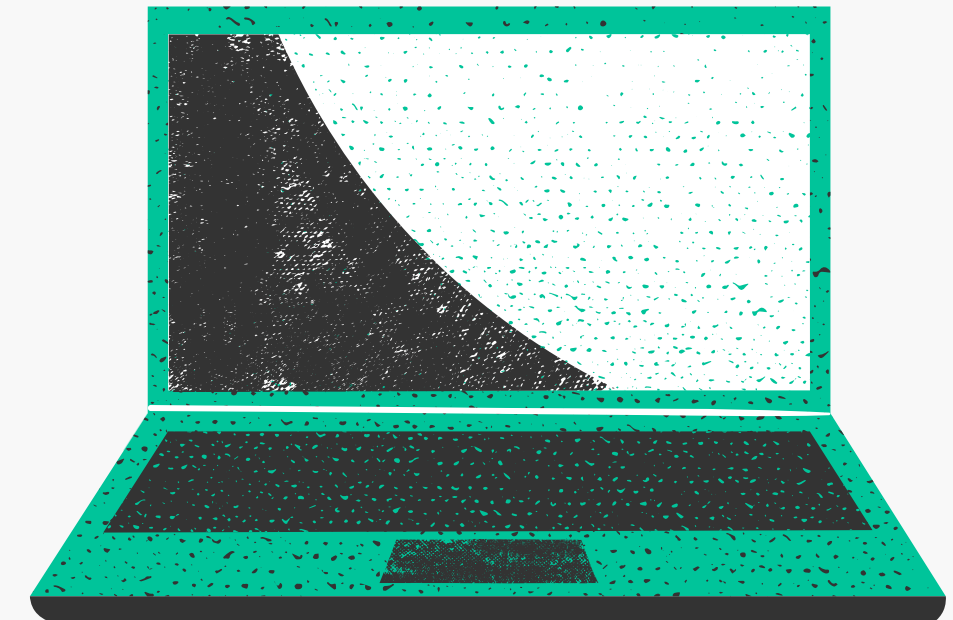
Provide more comprehensive understanding



Machine Learning Techniques

Use of Natural Language Processing

Security vulnerabilities, performance issues



Algorithms suggesting fixes

Automation of the identification and resolution of quality issues in real-time

13-06-2023

Department of Applied
Informatics, University of
Macedonia

ADVISOR

Alexander
Chatzigeorgiou

STUDENT

George - David
Apostolidis
iis20001

Thank you!