(Partial) Project Report On

User Git Analyzer

by

Neeraj Shilwant (BEAIDA67)

Jayesh Punjabi (BEAIDA75)

Vinit Ghadge (BEAIDA75)

Shubham Sonawane (BEAIDA75)

Under the guidance of

Dr. Bhagyashree Tingare

Department of Artificial Intelligence and Data Science

D. Y. Patil College of Engineering

Sector No. 29, Nigdi, Pradhikaran, Akurdi, Pune – 411044

SAVITRIBAI PHULE PUNE UNIVERSITY 2023 –2024

PROJECT APPROVAL SHEET A

Project

on

(User Git Analyzer)

Is successfully completed by

Neeraj Shilwant (BEAIDA67)

Jayesh Punjabi (BEAIDA75)

Vinit Ghadge (BEAIDA75)

Shubham Sonawane (BEAIDA75)

at

Department of Artificial Intelligence and Data Science

D. Y. Patil College of Engineering

Savitribai Phule Pune University

2023 -2024

Dr. Bhagyashree Tingare	Dr. V. G. Kottawar
Project Guide	Head of Department

Acknowledgement

We extend our heartfelt gratitude to Dr. V. G. Kottwar, Head of the Department of Artificial Intelligence and Data Science, for his unwavering support and guidance throughout this project. His leadership and mentorship have been instrumental in fostering an environment of innovation and learning within our academic institution.

We extend our sincere appreciation to Dr. Bhagyashree Tingare, Department of Artificial Intelligence and Data Science, for their ongoing guidance and unwavering support during the development of this project. Their mentorship and expertise continue to be crucial in our project's journey. We are deeply grateful for their continued encouragement and valuable insights, which are instrumental in shaping the project's direction and progress. We express our sincere gratitude for her simulating guidance, Continuous encouragement and supervision throughout the present work.

Abstract

The "GitHub Profile and Repository Analyzer" is a comprehensive web-based platform designed to address the need for enhanced security within GitHub repositories and provide in-depth insights into GitHub user profiles. This multifaceted project combines advanced technologies, including React for the frontend, Flask for backend API development, and Python for machine learning, to create a scalable and user-friendly solution. The platform employs static code analysis techniques to identify security vulnerabilities in repositories and utilizes machine learning algorithms to analyze user profiles based on key parameters.

The GitHub Profile and Repository Analyzer project is driven by the desire to foster code security, simplify collaboration, and aid in hiring decisions within the GitHub community. With data visualization features, user-centric interfaces, and a strong focus on privacy and data protection, this project offers a valuable tool for developers, project maintainers, and organizations looking to enhance the integrity of their codebases, streamline collaboration, and make informed hiring decisions.

Keywords

- **GitHub**: A web-based platform for version control and collaboration, primarily used for software development. It hosts code repositories and provides tools for tracking changes, managing projects, and collaborating with a global community of developers.

- **Repository Security**: The practice of identifying and addressing vulnerabilities, misconfigurations, and threats within code repositories to safeguard code integrity and prevent security breaches.

- **User Profiling**: The process of analyzing and characterizing users based on their behavior, preferences, and contributions. In this project, it involves assessing GitHub users' skills and contributions.

- **Static Code Analysis**: A code analysis technique that examines source code without executing it to identify potential vulnerabilities, security issues, and coding errors.

- **Machine Learning**: A subset of artificial intelligence that enables systems to learn and make predictions based on data. In this project, machine learning is used for user profile analysis.

- **React**: A popular JavaScript library for building user interfaces. It is used to create a user-friendly and responsive front-end interface in the project.

- **Flask**: A Python web framework used for creating web applications and APIs. In this project, Flask is employed for routing and handling API requests.

- **Data Visualization**: The representation of data through visual elements like charts and graphs, making complex information more accessible and understandable.

- **Collaboration Enhancement**: The improvement of teamwork and cooperation among users, developers, and organizations within the GitHub community, fostering a more efficient and productive environment for code development and collaboration.

Table of Content

1. Introduction

1.1 Overview:

In today's software development landscape, GitHub has become the epicentre of collaborative coding. It's the go-to platform for hosting open-source projects and collaborating with developers worldwide. However, this widespread adoption has also brought about significant security and quality challenges. Repositories on GitHub often contain vulnerabilities and misconfigurations that can be exploited, leading to data breaches, compromised applications, and security risks. Additionally, assessing the credibility and skills of GitHub users for project collaboration or hiring decisions can be a daunting task, as it often involves manual and time-consuming analysis.

To address these pressing issues, we introduce the GitHub Profile Analysis & Security Scanner. This web-based platform offers a comprehensive solution that combines the power of advanced security scanning and in-depth GitHub user profile analysis. It empowers developers, organizations, and hiring managers to enhance the security of their repositories, improve code quality, and make informed decisions about project contributors and potential job candidates.

The GitHub Profile Analysis & Security Scanner leverages a wide range of technologies, including data processing, security scanning tools, GitHub API integration, and a user-friendly interface. The platform's core features include automated security scans of GitHub repositories, which identify vulnerabilities, misconfigurations, and sensitive information exposure in code. These findings are presented in a clear, actionable manner, enabling users to rectify issues promptly.

On the profile analysis front, the platform provides deep insights into a GitHub user's contributions, project involvement, coding skills, and more. This information is invaluable for assessing the suitability of potential collaborators or evaluating candidates for job openings. By consolidating both security and user analysis within a single, user-friendly interface, the GitHub Profile Analysis & Security Scanner streamlines the processes of securing code repositories and making informed decisions about project contributors. It's an essential tool in the modern developer's arsenal for safeguarding code quality, enhancing security, and fostering efficient collaboration on GitHub.

1.1.1 GitHub Profile Analysis tool

The GitHub Profiler is a web-based platform that leverages a combination of advanced technologies to offer an in-depth analysis of GitHub user profiles and repositories. Developed with a user-centric approach, its frontend is built using React, ensuring a user-friendly and responsive interface. The backend integrates Python-based machine learning

and real-time API requests, granting the platform scalability and reliability. Notably, the GitHub Profiler does not store data locally, but rather fetches real-time data from the GitHub API, ensuring its users have access to the most up-to-date information. Machine learning algorithms are employed to cluster users based on key parameters such as Stars Count, Issue Count, Forks Count, Number of Contributions, and Pull Requests.

Features

- Visualization Capabilities: The platform offers visual insights into GitHub profiles through pie charts, allowing users to understand their data more intuitively.

- Language Breakdown: Users can view a visual breakdown of the programming languages used in their repositories, helping them gauge their coding preferences and expertise.

- Commit Distribution: The platform provides a distribution chart for commit activities across different repositories, enabling users to assess their contribution patterns.

- Star Highlight: Users can identify repositories that have received the most attention in terms of stars, assisting them in recognizing popular and well-received projects.

- Collaboration and Hiring: In addition to individual users, the GitHub Profiler serves as a valuable tool for project collaborators and employers, streamlining project assessment and informed hiring decisions.

1.1.2. GitHub Security Scanner Tool.

The GitHub Security Scanner is a repository scanning tool designed to bolster the security of GitHub repositories. It combines state-of-the-art technologies to provide a robust solution for identifying and addressing security vulnerabilities within code repositories. With an emphasis on usability and efficiency, this tool equips developers and organizations with the means to enhance the integrity of their codebases, ultimately fostering a more secure development ecosystem on GitHub.

Features:

- Static Code Analysis: The GitHub Security Scanner leverages advanced static code analysis, including SAST (Static Application Security Testing) and SCA (Software Composition Analysis) techniques. This comprehensive examination ensures that GitHub repositories are thoroughly scrutinized for security vulnerabilities, misconfigurations, and potential threats.

- User-Friendly Interface: Built with React, the tool offers an intuitive and responsive user interface, facilitating user interactions and making it effortless to access vital security insights. This user-centric design streamlines the process of identifying and resolving security issues, making it accessible to developers with varying levels of expertise.

1.2 Objectives:

- To develop a user-friendly web-based platform that provides a comprehensive report of a candidate's GitHub profile, including real-time analysis, intuitive visualizations, and continuous monitoring.

- To help technical leads identify efficient candidates for projects by providing them with information about the candidate's contributions to open-source projects, their coding style, and their familiarity with different programming languages and technologies.

- To develop a GitHub security scanner that identifies security vulnerabilities in GitHub repositories.
- To help developers improve the security of their code and protect their applications from attacks.

1.3. Motivation:

The motivation for this project is to make the HR recruitment process more efficient and to help developers improve the security of their code.

- HR recruitment: The GitHub profiler will help technical leads identify efficient candidates for projects by providing them with a comprehensive report of the candidate's GitHub profile. This report will include information such as the candidate's contributions to open-source projects, their coding style, and their familiarity with different programming languages and technologies.

- Security: The GitHub security scanner will help developers identify security vulnerabilities in their code. This will help them to improve the security of their code and protect their applications from attacks.

1.4 Future Scope:

The future scope of this project is promising, with opportunities for further enhancement and expansion. As the platform continues to evolve and as user's needs change, it can adapt and grow to provide even more comprehensive insights. Future developments could include deeper integration with evolving API features, enabling real-time monitoring and alerting for repository changes, as well as extending the range of parameters for machine learning analysis. Moreover, there's potential to offer advanced visualizations and analytics, facilitating decision-making in a more data-driven manner, not just for individual profiles but also for collaborative project assessments and hiring processes.

Additionally, the project could explore the incorporation of features that support code quality and security, such as vulnerability scanning and code analysis. These enhancements would position the platform as an invaluable tool for both developers and organizations, promoting secure and efficient open-source development. In the future, the platform could play a pivotal role in fostering collaboration and informed decision-making within the GitHub community.

2. Literature Survey

Research Paper	Conference	Author	Description
Fix that Fix Commit: A real-world remediation analysis of JavaScript projects	2020 IEEE 20th International Working Conference	V. Bandara, Rathnayake N.Werasekara C.Elvitigala, K.Thilakarathna.	The research investigates 118K commits from 53 JavaScript projects on GitHub to understand the dynamics of vulnerability remediation. It reveals that 82% of the projects fixed prior vulnerabilities but introduced new ones. Proper internal testing could have prevented 78% of these vulnerabilities, emphasizing the importance of robust testing practices in software development.
Anomalicious: Automated Detection of Anomalous and Potentially Malicious Commits on GitHub	2021 IEEE/ACM 43rd International Conference on Software Engineering	D.Gonzalez, T.Zimmermann, P.Godefroid M. Schaefer	Anomalicious is an automated tool that uses commit logs and repository metadata to identify potentially malicious contributions in open-source software (OSS) repositories. It showed promising results by detecting 53.33% of malicious commits while flagging less than 1% of commits in most repositories.
Data collection and analysis of GitHub repositories and users	2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)	Fragkiskos Chatziasimidis, Ioannis Stamelos	The Research explores GitHub data, analyzing behavior and project success factors. Using GitHub API, data from around 100K projects and 10K users/owners were collected. Statistical analysis, k-means discretization, and apriori algorithm in Weka (Data mining and ML tool) were applied to identify association rules. It focuses on rules with over 1000 downloads, revealing seven success factors in the GitHub ecosystem.
Vritthi - a theoretical framework for IT	2016 International Conference on Data	A. Giri, A.Ravikumar, S. Mote and R. Bharadwaj	The paper presents Vritthi, an innovative recruitment system leveraging social networks like Twitter and LinkedIn, code

recruitment based on machine learning techniques applied over Twitter, LinkedIn, SPOJ and GitHub profiles	Mining and Advanced Computing (SAPIENCE)		repositories like GitHub, and competitive coding platforms to automatically match jobseekers with job requirements using data mining and machine learning. It introduces the Vritthi Professional Quotient (VPQF) through K-means clustering for personalized improvement suggestions, streamlining candidate selection for recruiters.
DevFlair: A Framework to Automate the Pre-screening Process of Software Engineering Job Candidates	022 4th International Conference on Advancements in Computing (ICAC)	R.Jayasekara, K. A.N.D. KudaRachchi, K.G.S.S.K. Kariyawasam, D. Rajapaksha, S.L.Jayasinghe, S.Thelijjagoda	Devflair framework: DevFlair uses data from social media, GitHub, and open-ended questionnaires to predict the Big-Five personality traits, analyze technical skill expertise, and analyze the experience in using industry-related online platforms

3. Project Design

3.1. Requirement Analysis

3.1.1 Problem Definition:

The increasing reliance on GitHub for code hosting and collaboration has brought about a pressing need for comprehensive security assessment within repositories. Developers and organizations face the challenge of identifying and rectifying security vulnerabilities and misconfigurations within their codebases, which can lead to data breaches, compromised applications, and significant security risks. Additionally, without efficient tools, it becomes time-consuming and daunting to assess and address these security issues, hindering the development process and creating potential security loopholes. This problem emphasizes the necessity for a sophisticated and user-friendly repository scanning tool that can conduct systematic static code analysis, enabling users to proactively secure their GitHub repositories and maintain code integrity.

The project analyzes repositories of the user. The user is meant to insert the URL of the GitHub repositories and get the results as no of vulnerabilities in file which is there in repositories analyze the vulnerability type and level of criticalness. The other module is of the GitHub profile analyzer which the user profile on the GitHub is analyzed by inserting the user profile URL for GitHub.

3.1.2. Software Requirement:

- Visual Studio Code
- Jupyter Notebook
- Web browser (Chrome, Firefox, MS Edge)

3.1.3. Technical Requirements:

- React JS
- Html & CSS
- Python (for Machine Learning Model)
- Flask Web Framework

3.2. Software Requirement Specifications

3.2.1. Introduction.

This document lays out a project plan for the development of the “User Git Analyzer” open-source web application.

This document is intended for current and future developers working on “User Git Analyzer”, as well as supporters of the project. The plan will include but is not limited to the content of the physical activity, the work from the perspective of the " “User Git Analyzer" group (me and my mentor), time and delivery estimates, risks and how to reduce the risk. The process bt which we will develop the project and the precautions that will be recorded throughout the project.

3.2.2. **Overview**.

In response to the growing need for enhanced GitHub repository security and a better understanding of GitHub users, the GitHub Profile and Repository Analyzer project presents a comprehensive solution. The problem of unaddressed security vulnerabilities in code repositories and the challenge of identifying suitable collaborators or candidates are tackled head-on. The project entails the development of a multifaceted web platform that combines advanced static code analysis tools to identify security vulnerabilities in GitHub repositories with user profile analysis facilitated by machine learning. By providing in-depth insights into users and codebases, the project aims to foster secure coding practices, streamline collaboration, and facilitate informed hiring decisions within the GitHub community, all within a user-friendly and scalable environment.

3.2.2.1. **Potential Customers**

The project is designed to benefit a diverse range of users within the GitHub ecosystem. Developers and repository owners can utilize the security analysis tool to proactively identify and rectify vulnerabilities in their code repositories. Open-source project maintainers can use the platform to identify potential contributors with relevant skills, streamlining the collaboration process. Employers and hiring managers can make more informed hiring decisions by assessing the GitHub profiles of prospective candidates, while the broader GitHub community gains access to a valuable tool for improving code security, project collaboration, and hiring processes.

3.2.2.2. **Functionality**

- Users will be able to insert the URL of the GitHub repository.
- Users will be able to check the vulnerabilities in the files of the repositories and their types.
- Users will be able to check the criticalness of the vulnerability found so as to mark it important or ignore it.
- Users will be able to insert the URL of the GitHub profile.
- Users will get the analysis report of the specific GitHub profile with.
- Users can download the profile analysis report in the pdf format (unfixed functionality)

3.2.2.3. Platform.

The project will be developed and Launched as the Web- based Application and would be developed on Visual Studio code and Jupyter notebook for Machine Learning Model.

3.2.3. Goals and Scope.

- User should be able to insert the GitHub repository URL.
- User should be able to see the vulnerabilities found in a particular repository file.
- User should be able to insert the GitHub Profile URL.
- User should be able to get reports of analysis of GitHub profile.
- User should be able to download the GitHub profile analysis report.

3.2.4. **Deliverables**.

We'll be delivering the following using course of development:

- Feature specification
- Product design
- Test plan
- Development document
- Source code

3.2.5. **Risk Management**

3.2.5.1. Risk Identification

- Data Privacy and Security: Handling user and repository data raises privacy and security concerns. Ensuring the safe storage and transmission of sensitive information is crucial to prevent data breaches or misuse.

- API Reliability: The project relies on the GitHub API. Any changes or disruptions to the API can impact the project's functionality and data retrieval processes.

- False Positives/Negatives: The static code analysis component may generate false positives or negatives, potentially leading to the misidentification of security vulnerabilities or safe code.

- Machine Learning Accuracy: The accuracy of machine learning algorithms used for user profile analysis is dependent on the quality and diversity of the training data. Inaccuracies in clustering or user assessments can affect collaboration and hiring decisions.

- Scalability and Performance: As the project scales, ensuring the platform's performance and responsiveness becomes a challenge. Resource constraints or inefficiencies in data processing can impact the user experience.

3.2.5.2. Risk Mitigation.

- Data Privacy and Security: Implement robust data encryption and access controls. Regular security audits and data protection policy adherence.

- API Reliability: Stay informed about API changes and provide fallback options. Diversify data sources and ensure error handling.

- False Positives/Negatives: Fine-tune static code analysis algorithms. Allow user review and verification of findings.

- Machine Learning Accuracy: Regularly update and diversify training data. Implement user feedback mechanisms.

- Scalability and Performance: Optimize code and infrastructure for scalability. Employ load balancing for high usage.

3.2.6. Technical Process.

- Frontend Development (React JS): Create a user-friendly and responsive interface with React JS. Utilize HTML and CSS for web design and styling.

- Backend Development (Flask for ML Deployment): Set up Flask for handling routing and API requests, specifically for deploying the machine learning models. Develop API endpoints for data retrieval and processing, focusing on model deployment and data interaction.

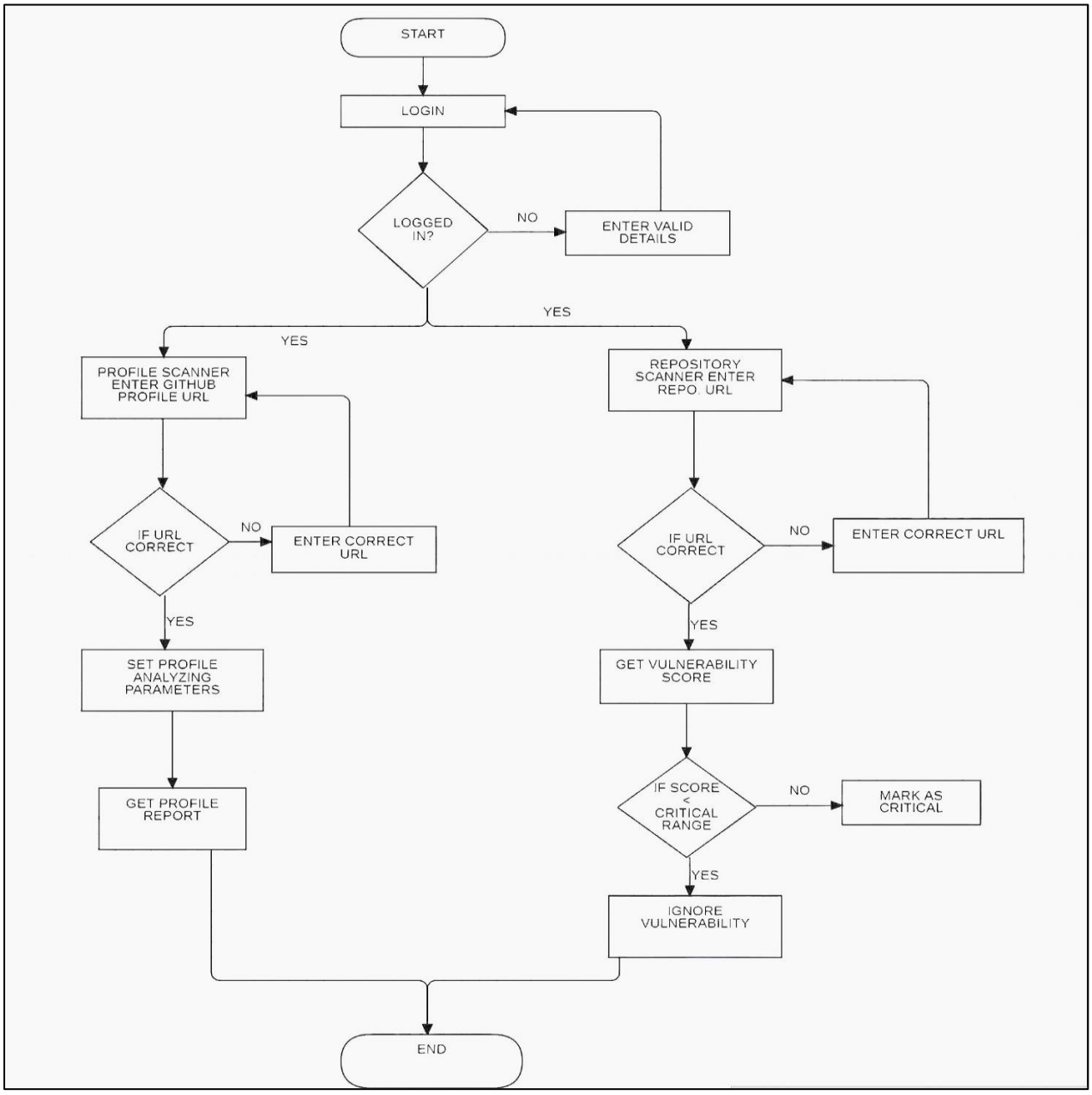
- Machine Learning for User Profiling (Python): Develop machine learning models for user profile analysis. Implement clustering algorithms based on specified parameters.

- User Interface and Visualization: Create data visualization components for user and repository insights. Develop an intuitive user interface for accessing analysis reports and insights.

3.3. Project Design.

3.3.1. Flowchart.

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

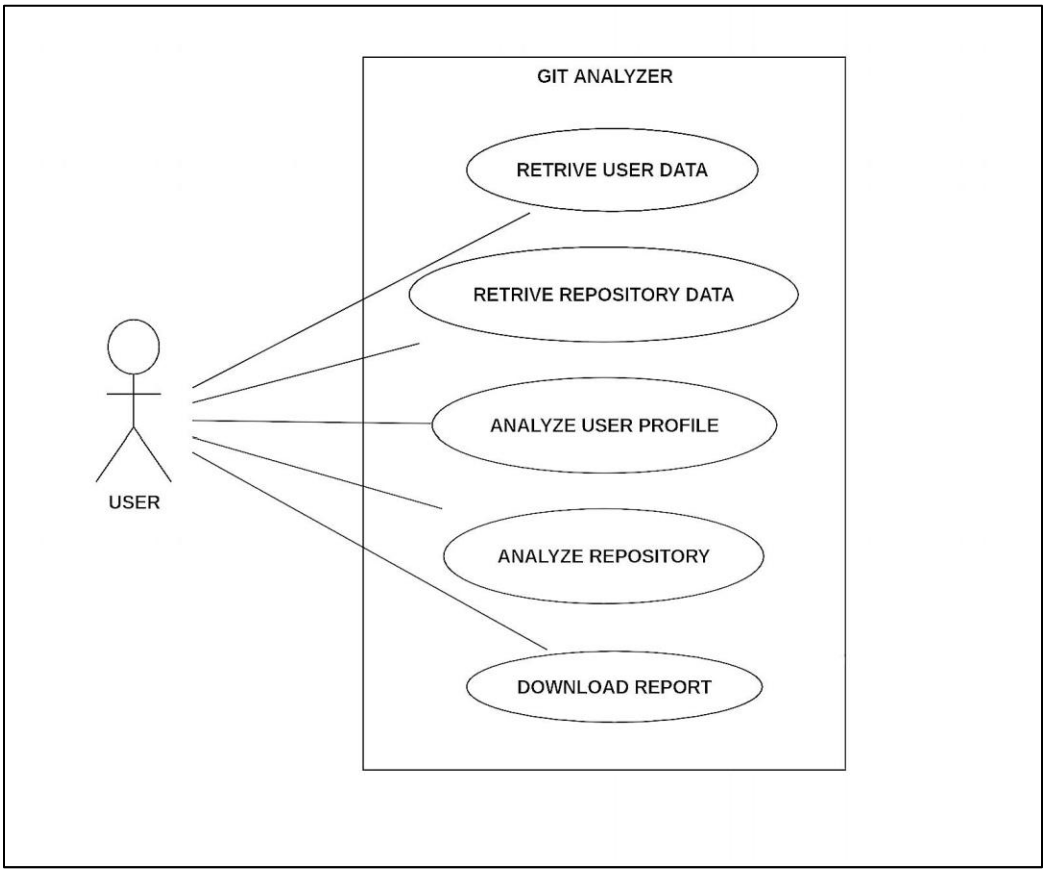


3.3.2. Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors.

The notation for a use case diagram is straightforward and doesn't involve as many types of symbols as other UML diagrams.

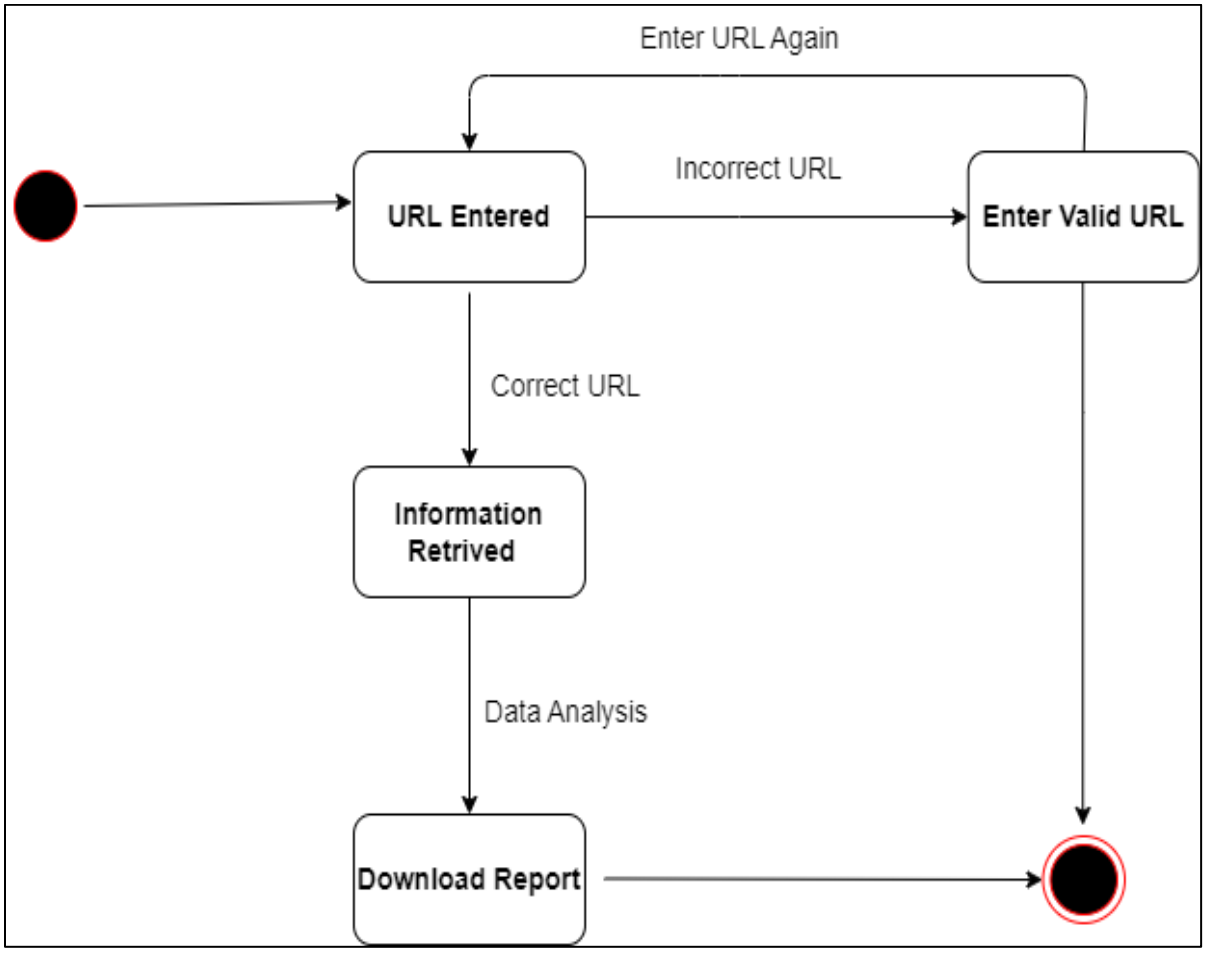
- Use cases: Horizontally shaped ovals that represent the different uses that a user might have.
- Actors: Stick figures that represent the people employing the use cases.
- Associations: A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.



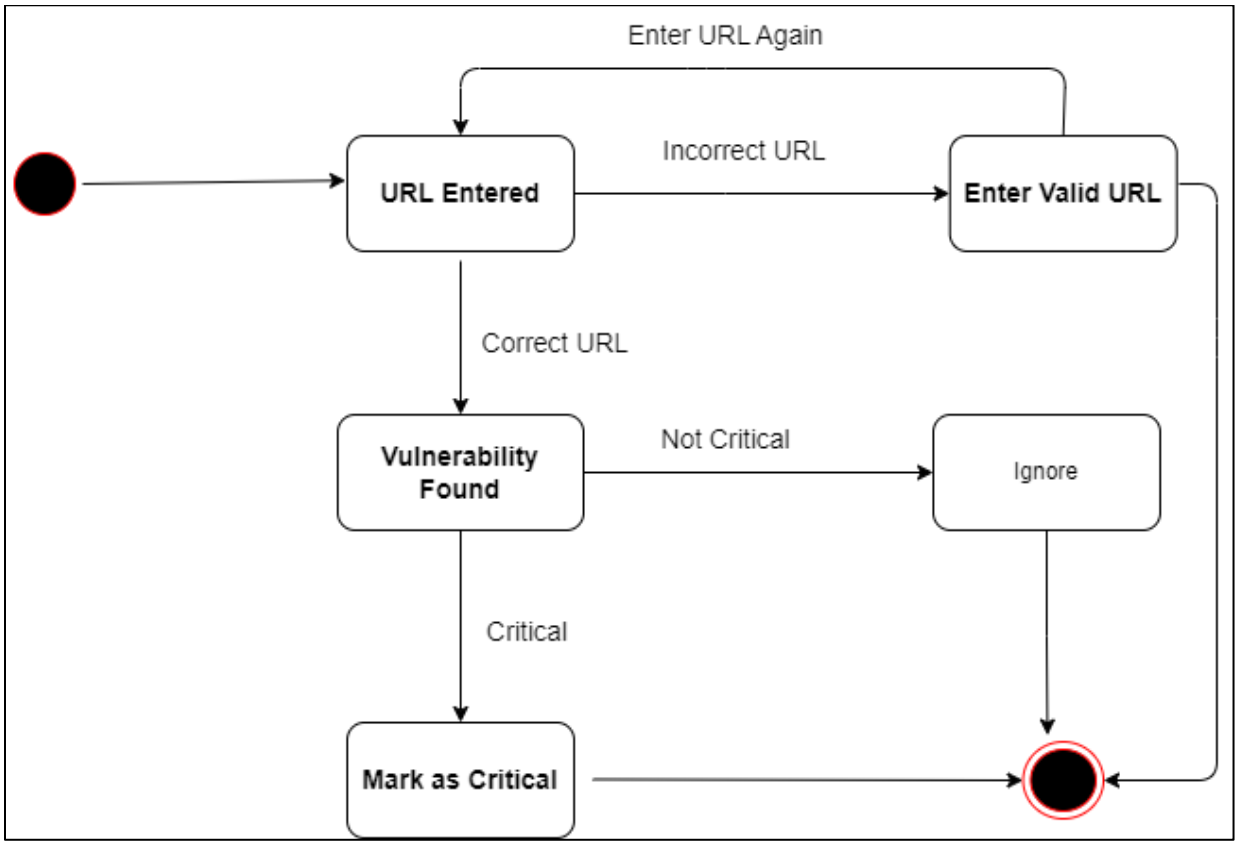
3.3.3. State Machine Diagram

A State Machine Diagram is a UML diagram that depicts the various states an object or system component can exist in and the transitions between those states. It's a visual representation of how an object responds to events and changes its behaviour based on its internal state, helping to model complex systems' dynamic behaviour and decision logic.

1. Profile Analyser:

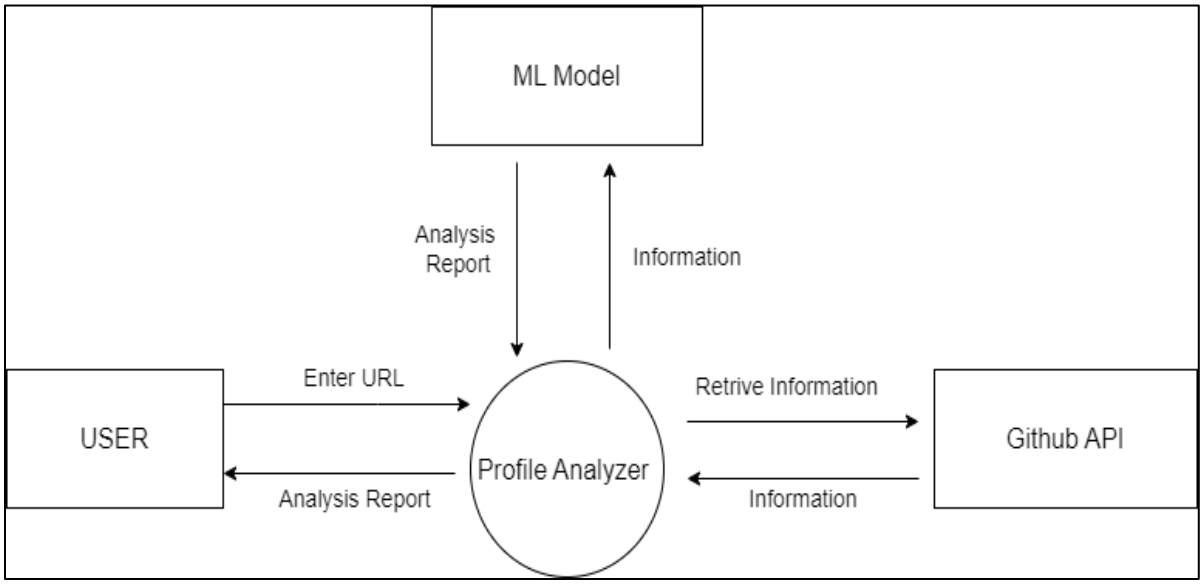


2. Repository Analysis



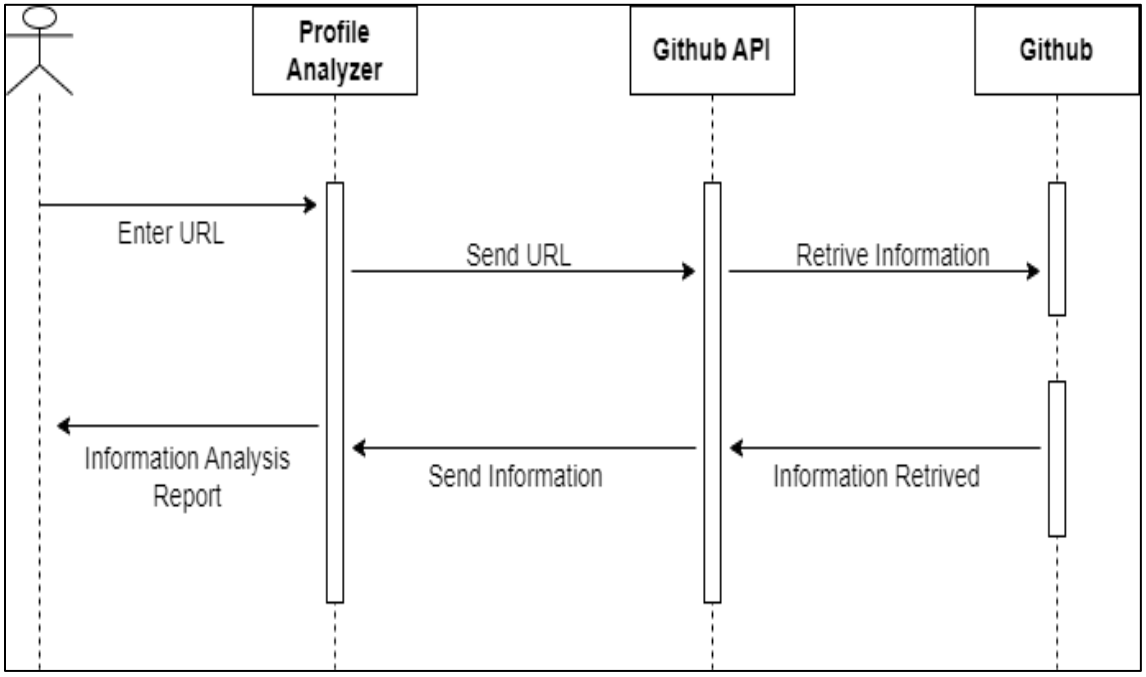
3.3.4. Data flow Diagram

A Data Flow Diagram (DFD) is a visual representation of how data flows within a system, showing the processes that transform and store data, data sources, data destinations, and the data flow paths. DFDs are used to understand, analyze, and model the flow of data in a structured way, making them valuable for clarifying system requirements and interactions. Below Data Flow Diagram is for Profile Analyser.



3.3.5 Sequence Diagram

A Sequence Diagram is a type of UML diagram that illustrates the interactions and order of messages between objects or components in a system. It offers a time-ordered view of how these entities collaborate to accomplish specific tasks or scenarios.



3.4. Planning

3.4.1. Technologies To Be Used.

- **Frontend Technology (React JS):** React is a popular JavaScript library for building user interfaces. It allows developers to create interactive, responsive, and user-friendly web applications. In the context of your project, React will be used to design the frontend of the GitHub Profile and Repository Analyzer, providing a visually appealing and interactive user interface.

- **Backend Framework (Flask):** Flask is a micro web framework for Python. It is used to build web applications and APIs. In your project, Flask will serve as the backend technology, handling routing, API requests, and the interaction between the frontend and the database.

- **HTML/CSS:** HTML (Hypertext Markup Language) is the standard markup language used to create web pages. It provides the structure and content of web pages. CSS (Cascading Style Sheets) is used for styling and layout. In your project, HTML and CSS will be used to structure and style the frontend for a visually appealing and user-friendly interface.

- **Python (for Machine Learning Models):** Python is a versatile and widely used programming language. In your project, Python will be employed for developing and deploying machine learning models used for user profile analysis. These models will assist in identifying proficient candidates for collaboration or hiring on GitHub.

- **GitHub API:** The GitHub API (Application Programming Interface) allows your project to interact with GitHub's data and resources in real-time. This API provides access to user profiles and repository data, enabling your application to fetch information directly from GitHub.

3.4.2. Project Plan.

Phase 1: Data Gathering and Preparation

Collect data from GitHub, including user profiles, code repositories, and issue reports. Clean and pre-process the data to make it suitable for analysis. Create a data flow diagram to show how the data will be collected, processed, and used.

Phase 2: System Design

Partition the project into two phases: User Profile Analyzer and Security Scanner. Create a flowchart for each phase to show the sequence of steps that will be performed.

Phase 3: User Profile Analyzer

Gather requirements and features for the User Profile Analyzer phase. Create a generalized logistics of data flow for the User Profile Analyzer. Analyze which parameters are important for analysing a user profile on GitHub. Gather a dataset of related parameters for making a machine learning model from various data sources.

Phase 4: Security Scanner

Gather requirements and features for the Security Scanner phase. Create a logistics to make the tool and the dataflow for the analysis of the codebases. Analyze the method which will be used for making the tool.

Phase 5: Technology Usage

Identify the technologies that will be used for the project.

Phase 6: Development of Phase 1 (Frontend)

Create the UI and system architecture for the User Profile Analyzer phase.

Phase 7: Development of Phase 1 (Backend)

Create the backend for the User Profile Analyzer phase using JavaScript. Create and test the machine learning model. Implement the model in the frontend and test it with real-time data.

Phase 8: Development of Phase 2 (Frontend)

Create the UI and system architecture for the Security Scanner phase.

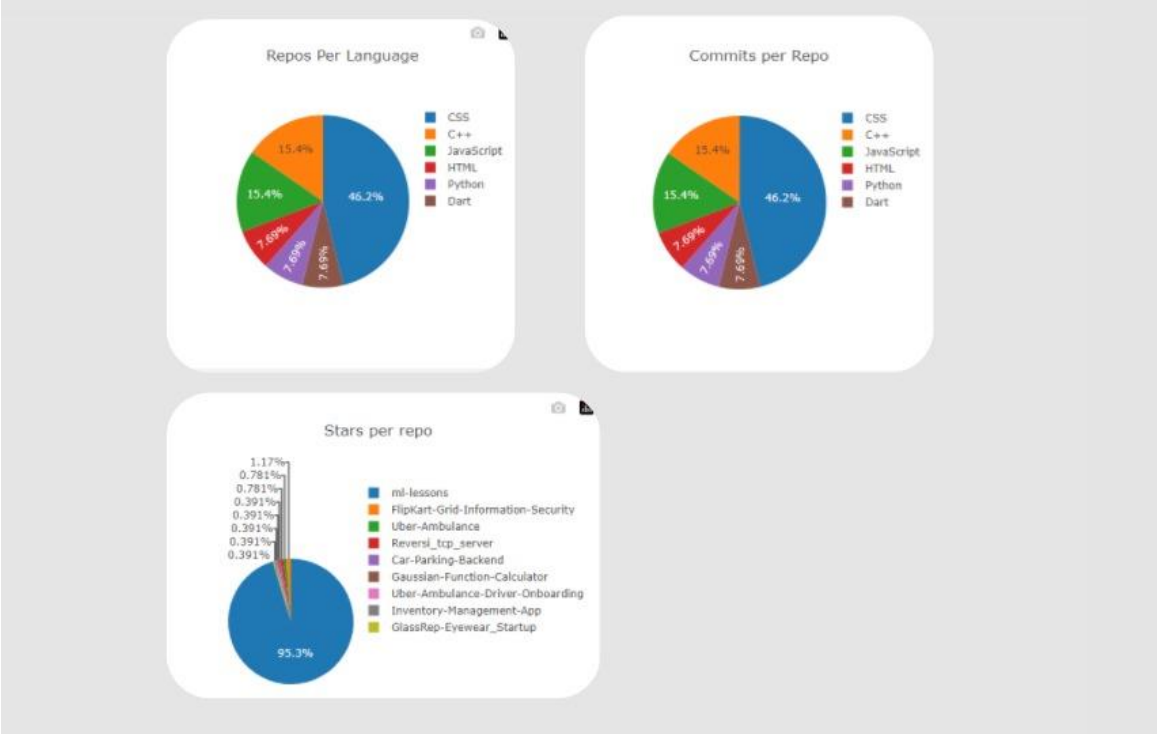
Phase 9: Development of Phase 2 (Backend)

Create the backend for the Security Scanner phase using JavaScript. Test the project on a real-time database.

Phase 10: Deployment

Deploy the project to a production environment.

3.5. Project Prototype


Repos Per Language
CSS
C++
JavaScript
HTML
Python
Dart
46.2%
15.4%
15.4%
7.69%
7.69%
7.69%
Commits per Repo
CSS
C++
JavaScript
HTML
Python
Dart
46.2%
15.4%
15.4%
7.69%
7.69%
7.69%
Stars per repo
ml-lessons
FlipKart-Grid-Information-Security
Uber-Ambulance
Reversi_tcp_server
Car-Parking-Backend
Gaussian-Function-Calculator
Uber-Ambulance-Driver-Onboarding
Inventory-Management-App
GlassRep-Eyewear_Startup
95.3%
1.17%
0.781%
0.781%
0.391%
0.391%
0.391%
0.391%
0.391%

4. Conclusion

The "GitHub Profile and Repository Analyzer" project represents a pivotal step forward in addressing the evolving needs of the software development and open-source community. This meticulously designed web-based platform integrates advanced technologies like React, Flask, and Python for machine learning to deliver a scalable, secure, and user-centric solution. By incorporating static code analysis and machine learning, it empowers users to enhance code security, streamline collaboration, and make informed decisions about project contributors.

With real-time data retrieval from the GitHub API and data visualization features, the project offers a comprehensive toolset for developers, project maintainers, and organizations. It not only advances the field of collaborative coding but also prioritizes data privacy and ethical technology use. As the project nears completion, it signifies the potential of technology to drive innovation and provides a tangible example of how dedication and ingenuity can transform ideas into impactful solutions.

5. References

- V. Bandara et al., "Fix that Fix Commit: A real-world remediation analysis of JavaScript projects," 2020 IEEE 20th International Working Conference on Source Code Analysis and Manipulation (SCAM), Adelaide, SA, Australia, 2020.

- D. Gonzalez, T. Zimmermann, P. Godefroid and M. Schaefer, "Anomalicious: Automated Detection of Anomalous and Potentially Malicious Commits on GitHub," 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Madrid, ES, 2021.

- R. T. R. Jayasekara, K. A. N. D. Kudarachchi, K. G. S. S. K. Kariyawasam, D. Rajapaksha, S. L. Jayasinghe and S. Thelijjagoda, "DevFlair: A Framework to Automate the Pre-screening Process of Software Engineering Job Candidates," 2022 4th International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, 2022.

- A. Giri, A. Ravikumar, S. Mote and R. Bharadwaj, "Vritthi - a theoretical framework for IT recruitment based on machine learning techniques applied over Twitter, LinkedIn, SPOJ and GitHub profiles," 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, India, 2016.

- F. Chatziasimidis and I. Stamelos, "Data collection and analysis of GitHub repositories and users," 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), Corfu, Greece, 2015.

- Let’s analyze your code repository in GitHub via CodeQL engine

- Git 2.39.1 Fixes Two Critical Remote Code Execution Vulnerabilities

- CodeQL Github Repository

- Take GitHub to the command line