CODE ANALYZER Project Report

Professor: Tushar Sharma
TA: Ruchi Shinde
Course ID: CSCI 5308
Group ID: 4

Team Members:
Qurram Zaheer Syed
Bharatwaaj Shankaranarayanan
Arshdeep singh
Kavya Raval
Kishan Savaliya

Overview:

Code Analyzer is a platform where the analysis of the overall code is provided. It takes the insights from the existing version control systems and integrates with Github and JIRA, providing the analyses of metrics such as overall repositories count, total contributors in a repository, LOC refactored, and number of commits, and pull requests done by the user, and the file modifications performed.

The document below represents:

1. Dependencies used
2. Build documentation
3. User Scenarios
4. Smell Analysis
5. Testing
6. Project Dashboard Stats

DEPENDENCIES

Dependencies for backend:

Dependency name	Version	Usage of dependencies
octokit/core	^3.6.0	Extendable client for GitHub's REST & GraphQL APIs
octokit/plugin-paginate-rest	^2.17.0	Octokit plugin to paginate REST API endpoint responses
strapi/plugin-graphql	4.1.5	open-source headless CMS to build powerful APIs
types/react	^17.0.43	Used for type definitions for react
concurrently	^7.0.0	To run multiple commands concurrently
github-graphql-client	^1.0.0	Node.js client for the GitHub GraphQL API
mysql	^2.18.1	Database management system
mysql2	^2.3.3	MySQL client for Node.js with focus on performance. Supports prepared statements, non-utf8 encodings, binary log protocol, compression, SSL
pg	^8.7.3	Non-blocking PostgreSQL client for Node.js
pg-connection-string	^2.5.0	Functions for dealing with a PostgresSQL connection string
react	^18.0.0	Used for UI
sharp	^0.30.2	The typical use case for this high-speed Node.js module is to convert large images in common formats to smaller, web-friendly JPEG, PNG, WebP, GIF and AVIF images of varying dimensions.
sqlite3	5.0.2	Asynchronous, non-blocking SQLite3 bindings for Node.js.

Dependencies for frontend:

Dependency name	Version	Use
fortawesome/fontawesome-free	5.15.3	Used for icons
types/react	^17.0.43	Used for type definitions for react
axios	^0.26.1	Used to send asynchronous HTTP requests to REST endpoints and perform CRUD operations.
bootstrap	4.6.0	Used for designing webpages
Charts.js	2.9.4	Used to design charts
classnames	2.3.1	A simple JavaScript utility for conditionally joining classNames together.
moment	2.29.1	Used to display, format, parse, validate and manipulate for date and time
node-sass	6.0.1	Used to receive maintenance releases
node-sass-package-importer	5.3.2	Custom importer for node-sass to import packages from the node_modules directory.
nouislider	15.2.0	a lightweight JavaScript range slider.
react	^17.0.2	Used for UI
gulp	4.0.2	Toolkit to automate and enhance workflow

BUILD DOCUMENTATION

Building app from source

To build this app from source, follow these steps:

1. Clone the parent repository. This is a monorepo containing both the server and the client.

2. Navigate to `codeanalyzer-backend` and run `npm install`

3. Navigate to `codeanalyzer-frontend` and run `npm install`

4. Once the dependencies for both the front-end and the back-end are installed, you can

- Run the app locally. This can be done by navigating to `codeanalyzer-backend` and running `npm run full-dev-mode`
- Build the application. This can be done by navigating to `codeanalyzer-backend` and running `npm run full-prod-mode`

5. Optionally, you can build the `DesigniteJava` [1] execution script that automatically connects to the front-end and provides smell analysis of the repositories loaded in the application. This can be done by navigating to the `designite-shell` folder and running `npm run build`

Deployment

1. There is a `.gitlab-ci.yml` that uses Docker instances to do the building and deployment of server, client and designite-server on every push to `main` branch. This file can be configured according to requirements. The required deployment environment variables for this pipeline are:

<pre> HEROKU_APP: Name of the Heroku app for the front-end

HEROKU_APP_BACKEND: Name of the Heroku app for the back-end

SONAR_TOKEN: Token taken from SonarCloud setup

HEROKU_API_KEY: Heroku API Key generated by Heroku during account creation</pre>

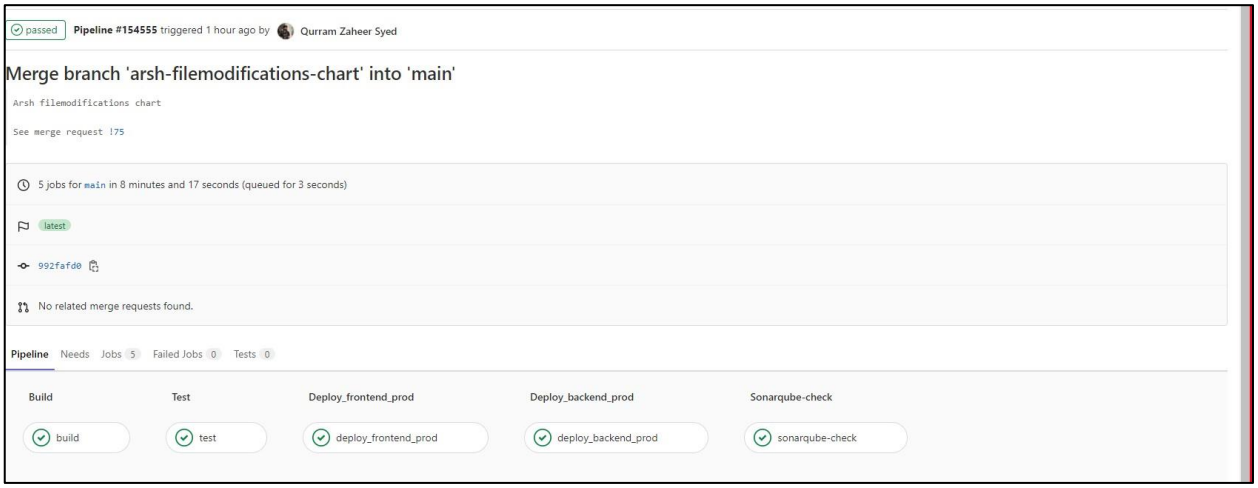
2. Additionally, a `.github/workflows/node.js.yml` file has also been provided with the necessary build and deploy scripts. For this pipeline, the same deployment variables as above are required.

> CI/CD pipelines were configured for both GitLab and GitHub, with **build**, **test**, and **Quality Checks** using SonarCloud. Two fully-functional pipelines were configured because of the GitLab runners' lack of reliable availability. Steps are mentioned below

CI/CD Pipeline setup

1. GitLab

In the GitLab CI/CD, a .gitlab-ci.yml file was created at the root of the monorepo. This file uses the Dalhousie docker runner to install Node.js v16.14.0, which is used throughout our project (for both the front-end and back-end). Within this file, there are stages for build, test, deploy_uat (for deploying to staging environment), deploy_frontend_prod (for deploying front-end to production), deploy_backend_prod (for deploying back-end to production), and sonarqube-check (For code quality checks. Initially, SonarQube was used, but since it was difficult to deploy due to lack of sufficient storage in free servers, SonarCloud was used as an alternative).



2. GitHub

A pipeline was also set up on GitHub by pushing the repo there as a backup, since the GitLab pipeline. This uses GitHub marketplace apps for Heroku and SonarCloud to build, test, deploy and do code qualitycheck for the front-end and the back-end. This pipeline was not configured with deploy_uat since it was supposed to be a backup, and the github repository is only a copy which doesn't have the same branches as the main GitLab repository.

Folder structure for Code Analyzer Frontend File system.

```
├── .github\workflow
├── Documentation
├── node_modules
├── public
└── src/
    ├── assets/
    │   ├── css
    │   ├── fonts
    │   ├── img
    │   ├── plugins
    │   └── scss
    ├── auth/
    │   ├── logout.js
    │   ├── PrivateRoute.js
    │   ├── useToken.js
    │   └── useUser.js
    ├── components/
    │   ├── Card
    │   ├── Charts
    │   ├── Footers
    │   ├── Headers
    │   ├── Navbars
    │   └── Sidebar
    ├── contexts/
```

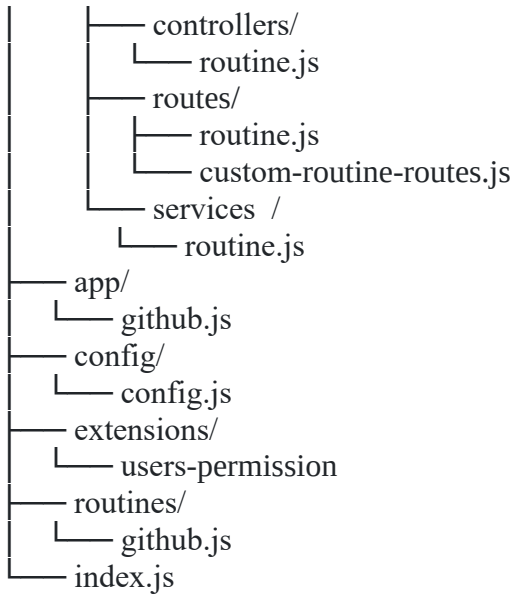
```
│   ├── GithubContext.js
│   ├── JiraContext.js
│   └── RepositoriesContext.js
├── hooks/
│   └── useQuery.js
├── layouts/
│   ├── Admin.js
│   └── Auth.js
├── lib\api/
│   ├── dashboardData.js
│   ├── index.js
│   └── jira.js
├── variables/
│   └── charts.js
├── views/
│   ├── subviews/
│   │   ├── Commits.js
│   │   ├── Icons.js
│   │   ├── Profile.js
│   │   ├── PullRequests.js
│   │   ├── PullRequestsRepo.js
│   │   └── Tables.js
│   ├── AddRepos.js
│   ├── Dashboard.js
│   ├── JiraAuth.js
│   ├── Login.js
│   ├── LoginForm.js
│   ├── Register.js
│   └── Repositories.js
├── App.js
├── COLORS.js
├── config.js
├── index.js
├── repoRoutes.js
└── routes.js
```

Folder structure for Code Analyzer Backend File system

```
├── .cache
├── build
├── config
├── database
├── node_modules
├── public
└── src/
```

```
├── admin/
│   ├── app.example.js
│   └── webpack.config.example.js
├── api/
│   ├── branch/
│   │   ├── content-types\branch/
│   │   │   └── schema.json
│   │   ├── controllers/
│   │   │   └── branch.js
│   │   ├── routes/
│   │   │   └── branch.js
│   │   └── services/
│   │       └── branch.js
│   ├── commit/
│   │   ├── content-types\commit/
│   │   │   └── schema.json
│   │   ├── controllers/
│   │   │   └── commit.js
│   │   ├── routes/
│   │   │   ├── commit.js
│   │   │   └── custom-commit-routes.js
│   │   └── services/
│   │       └── commit.js
│   ├── committedfile/
│   │   ├── content-types/
│   │   │   └── schema.json
│   │   ├── controllers/
│   │   │   └── committedfile.js
│   │   ├── routes/
│   │   │   └── committedfile.js
│   │   └── services/
│   │       └── committedfile.js
│   ├── contributor/
│   │   ├── content-types/
│   │   │   └── schema.json
│   │   ├── controllers/
│   │   │   └── contributor.js
│   │   ├── routes/
│   │   │   ├── contributor.js
│   │   │   └── custom-contributor-routes
│   │   └── services/
│   │       └── contributor.js
│   ├── github-auth/
│   │   ├── content-types/
│   │   │   └── schema.json
│   │   ├── controllers/
```

```
│   │   │   └── github-auth.js
│   │   ├── routes/
│   │   │   └── github-auth.js
│   │   └── services/
│   │       └── github-auth.js
│   ├── jira-app/
│   │   ├── content-types/
│   │   │   └── schema.js
│   │   ├── controllers/
│   │   │   └── jira-app.js
│   │   ├── routes/
│   │   │   └── jira-app.js
│   │   └── services/
│   │       └── jira-app.js
│   ├── jira-auth/
│   │   ├── content-types/
│   │   │   └── schema.json
│   │   ├── controllers/
│   │   │   └── jira-auth.js
│   │   ├── routes/
│   │   │   └── jira-auth.js
│   │   └── services/
│   │       └── jira-auth.js
│   ├── pull-request/
│   │   ├── content-types/
│   │   │   └── schema.json
│   │   ├── controllers/
│   │   │   └── pull-request.js
│   │   ├── routes/
│   │   │   ├── pull-request.js
│   │   │   └── custom-pull-request.rote.js
│   │   └── services/
│   │       └── pull-request.js
│   ├── repository/
│   │   ├── content-types/
│   │   │   └── schema.json
│   │   ├── controllers/
│   │   │   └── repository.js
│   │   ├── routes/
│   │   │   ├── custom-repository-routes.js
│   │   │   └── repository.js
│   │   └── services/
│   │       └── repository.js
│   └── routine/
│       ├── content-types/
│       │   └── schema.json
```

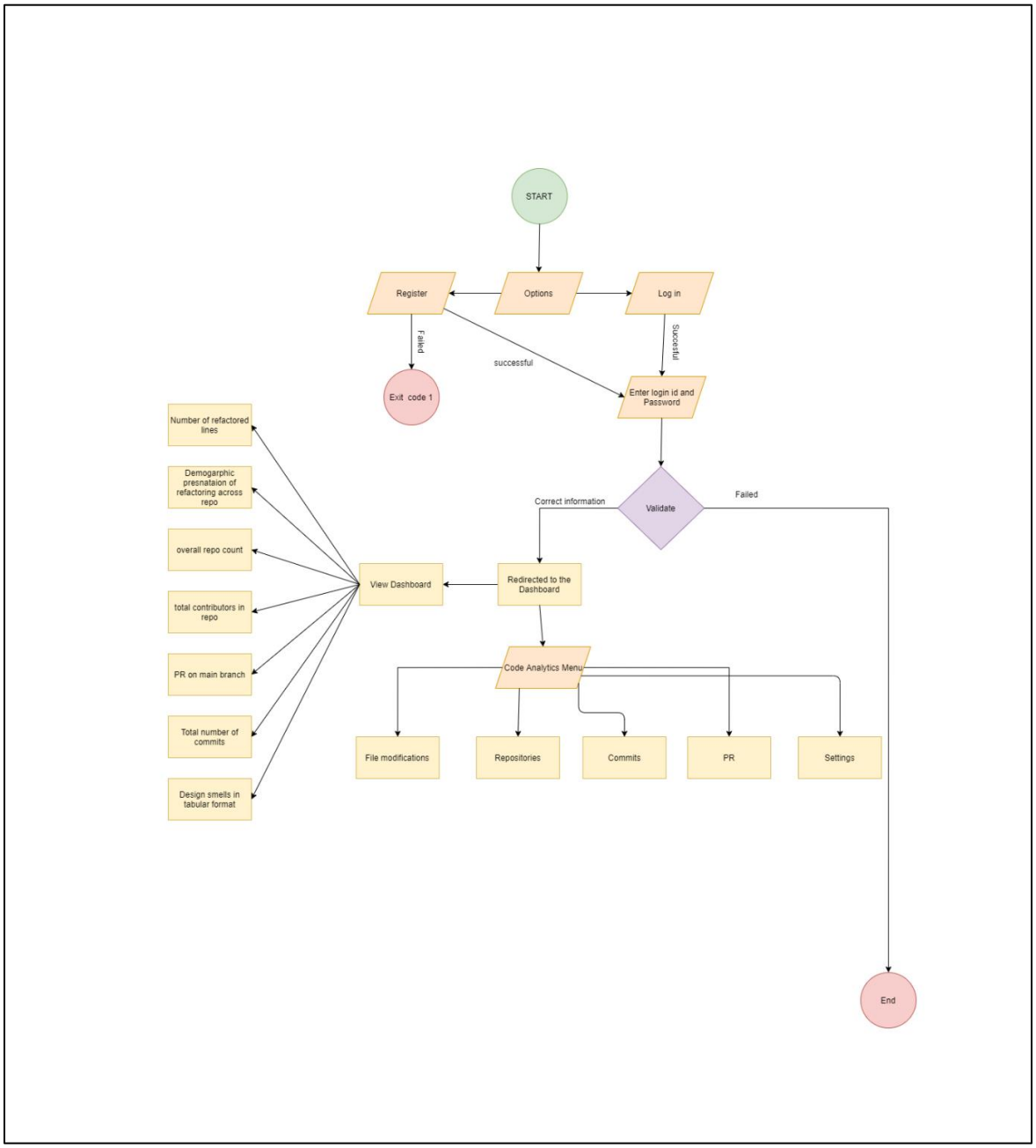


API Documentation:

Using postman API documentation is done.

https://documenter.getpostman.com/view/19892208/UVyswbFn#2ea14c92-48f2-47be-81bf-e2b13340a0b9

USER SCENARIOS



1. Login/Registration with Google or Github O Auth

Any new user can register either by creating an account with an email address and password or can sign-up using a google account or with Github OAuth.

Fig 1 represents the registration or signup page of code analytics

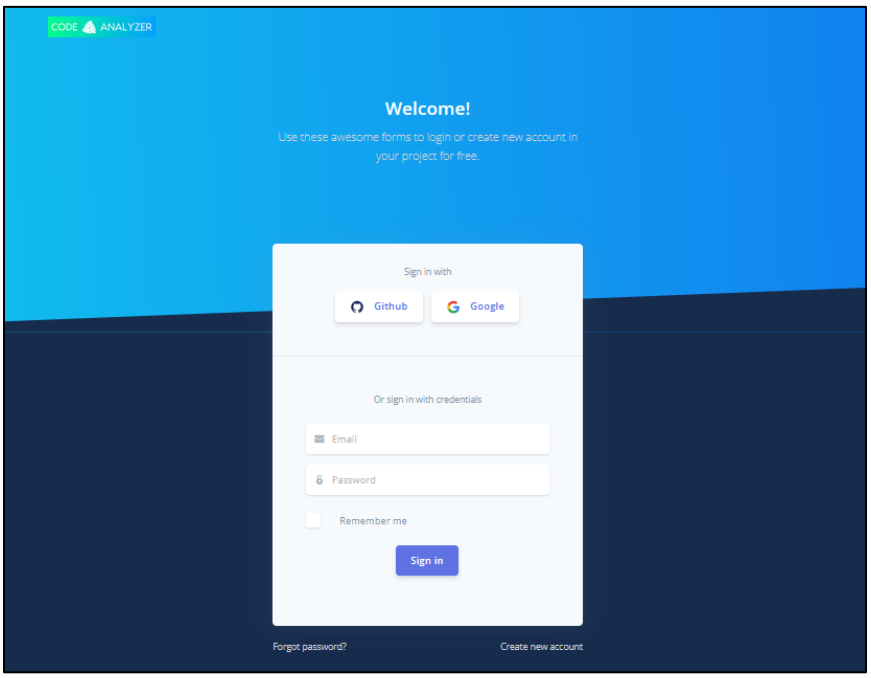
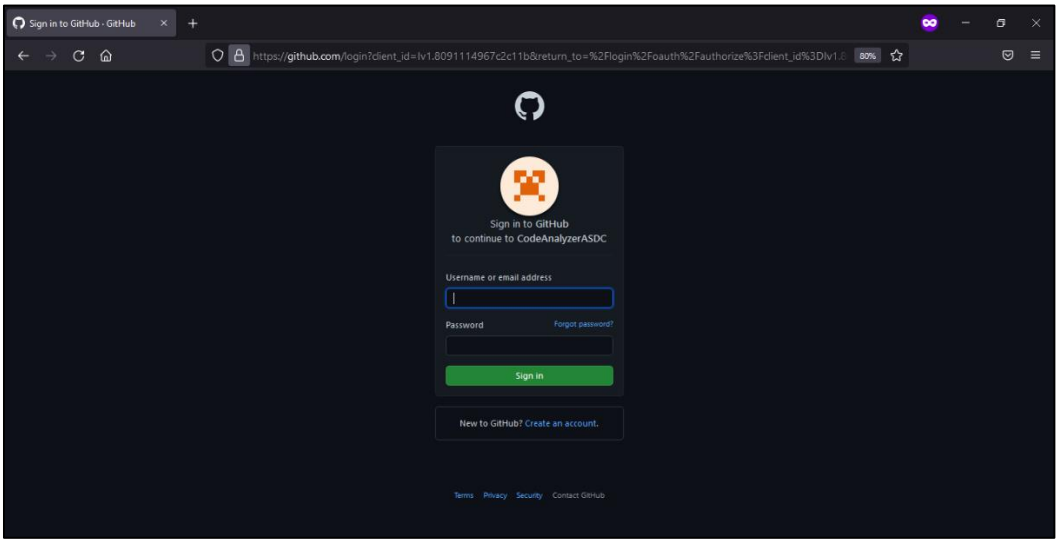
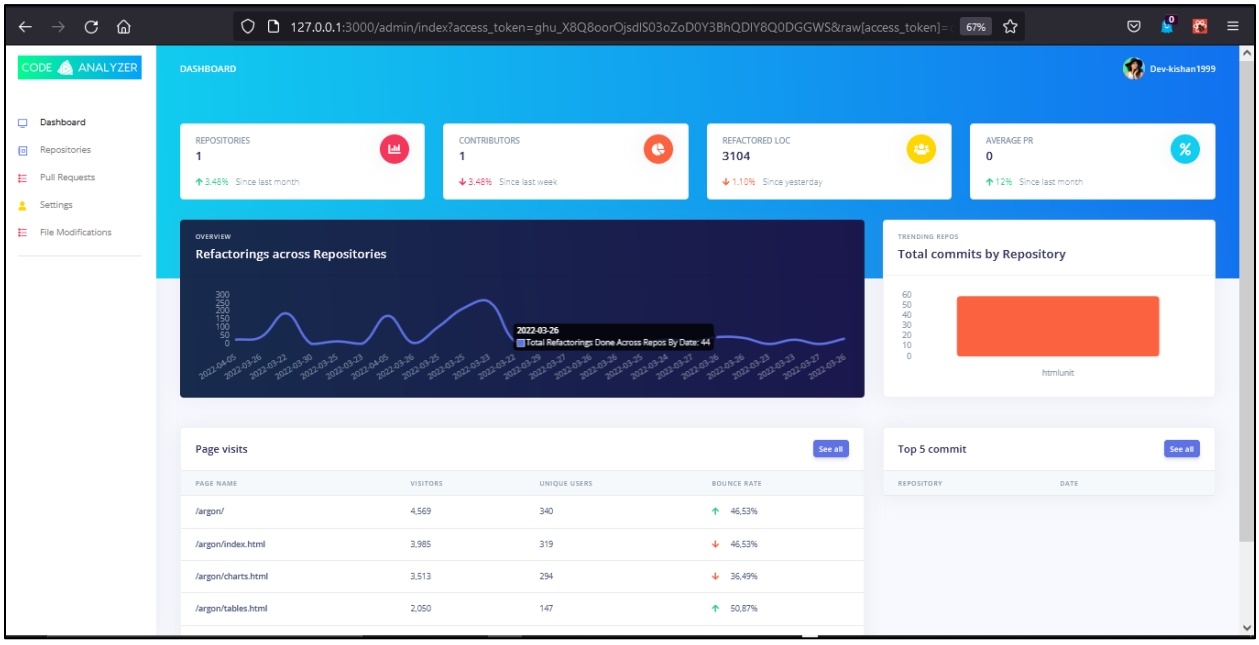


Fig 2 represents the signup page using Github OAuth



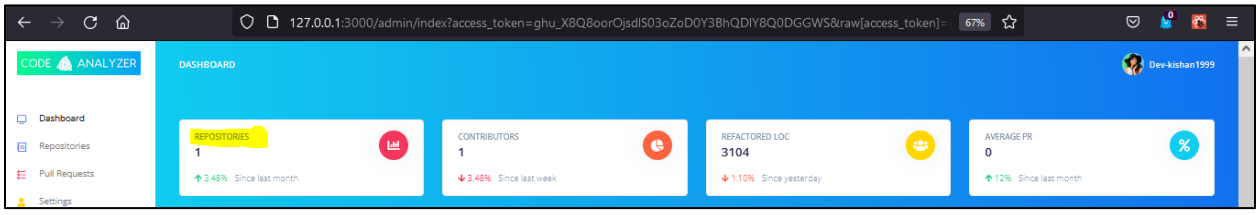
2. Dashboard

CodeAnalytics dashboard has multiple functionalities as described below:



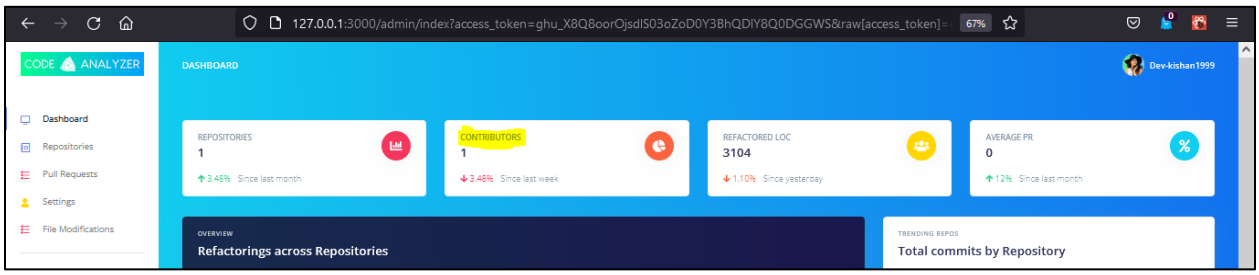
2.1 View Overall repositories count

This dashboard provides the number of repositories a user has



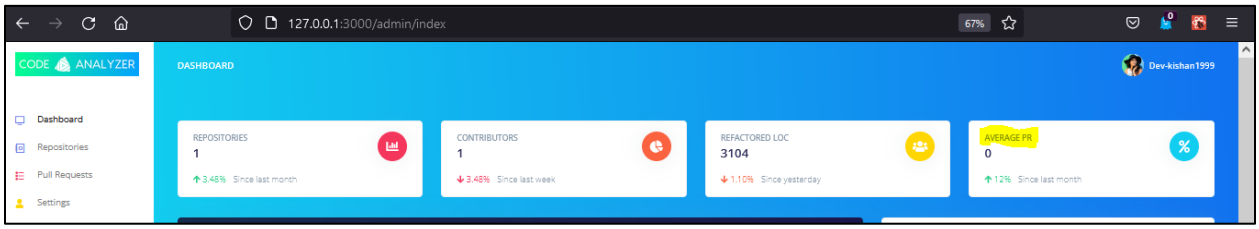
2.2 View total contributors in a repository

This dashboard provides the total number of contributors in the repository



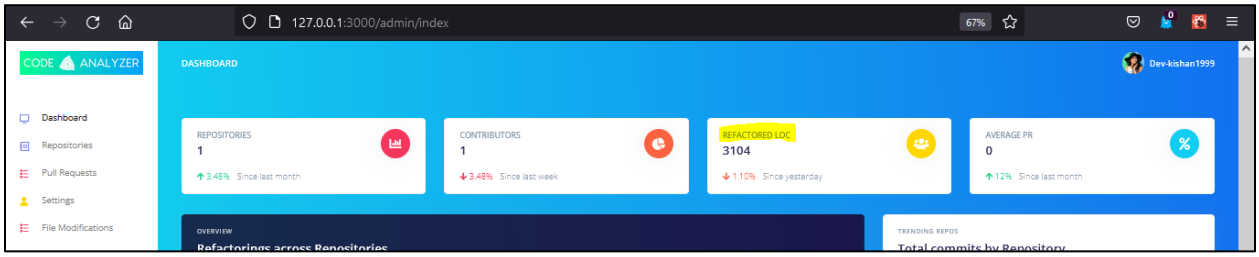
2.3 View the percentage of Pull Request on the main branch

This dashboard represents the percentage of pull requests performed on the main branch



2.4 View total line of code refactored

This dashboard represents the total number of lines of code refactored across all the repositories

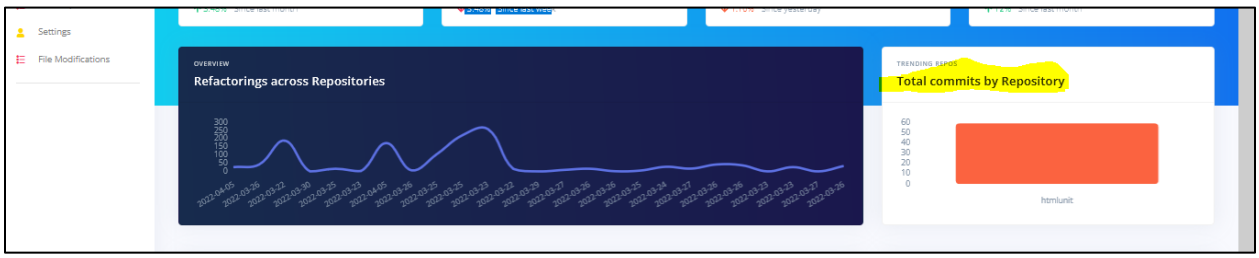


2.5 A chart representing refactoring across the repos

The line chart represents all the refactoring performed across the dates for all the repositories

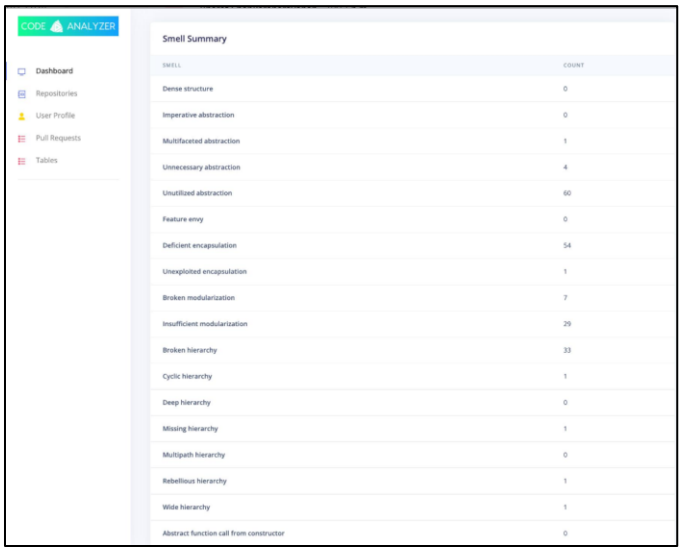
2.6 A chart representing total commits by repository

The bar chart represents the total number of commits each repository has



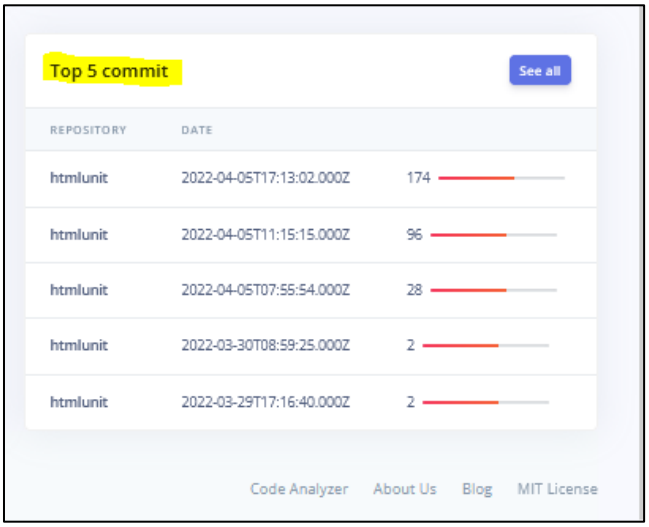
2.7 A table representing Designite Output

Smells summary report table displays the smells of the code checked by tool Designite



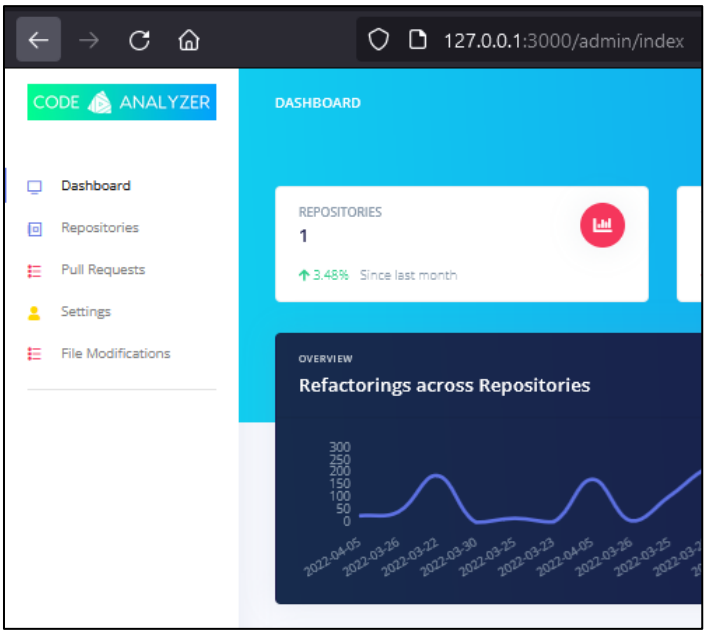
2.8 A table representing the latest commits performed on a repository

The table represents the latest 5 commits which show the day of commit, repository name, and number of changes done on that repository

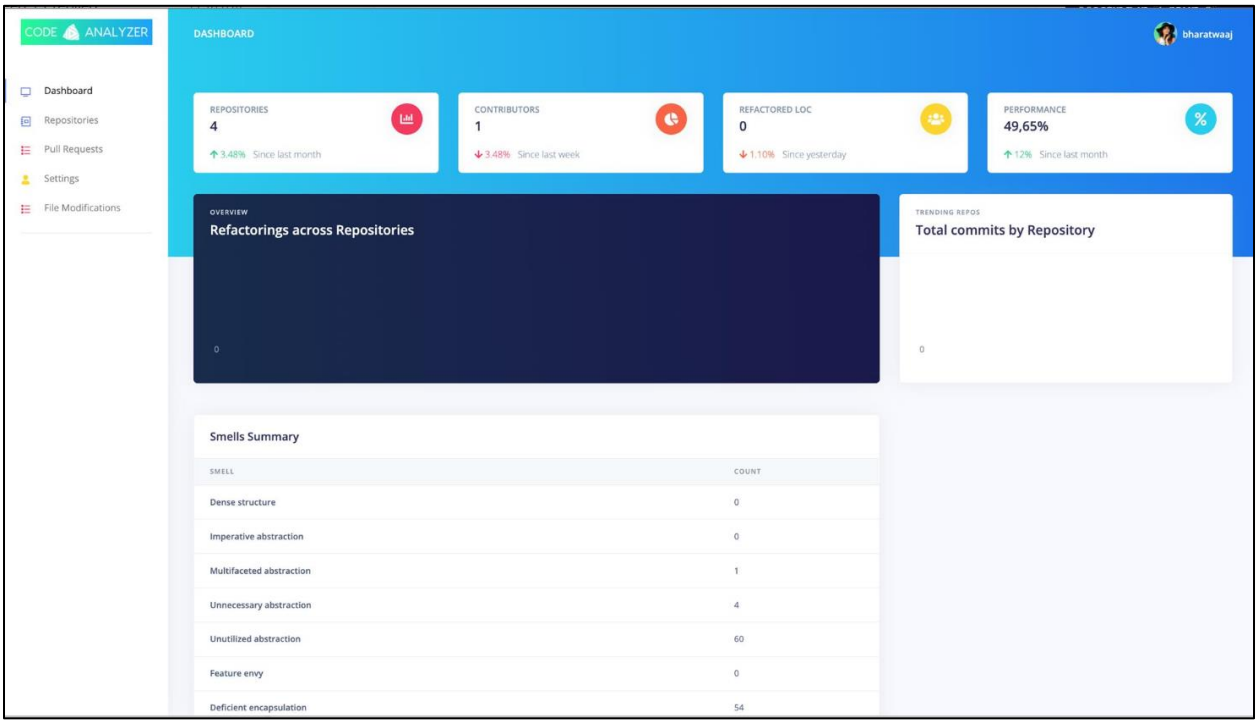


2.9 A menu bar that consists

The menu bar consists of multiple options mentioned in the below snapshot

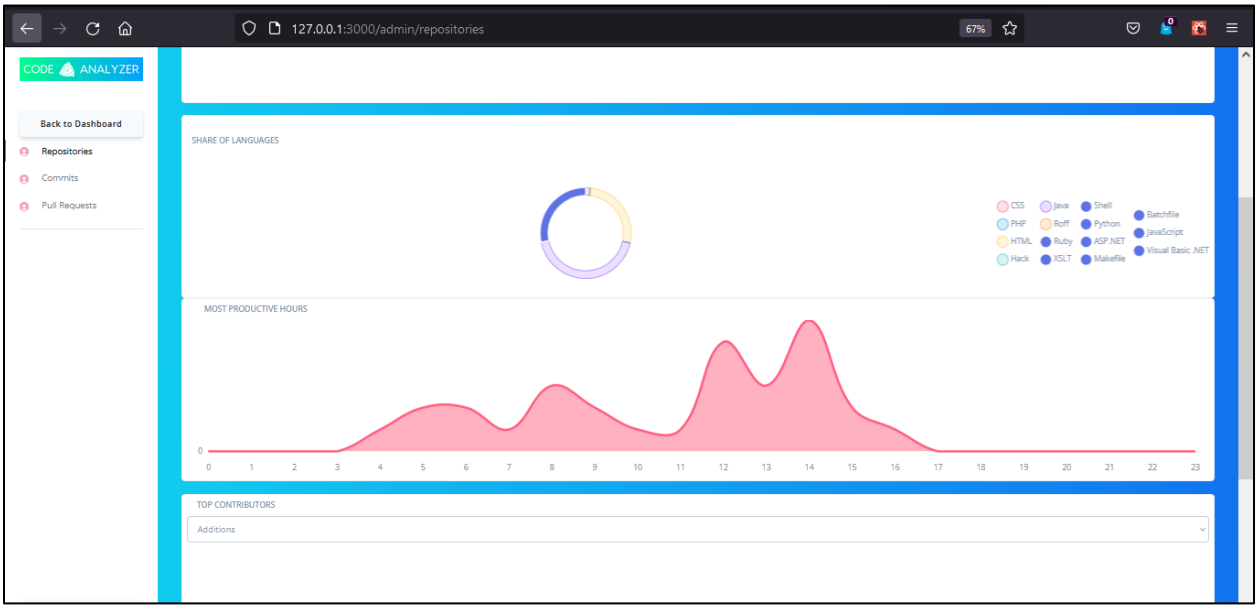


2.9.1 Dashboard



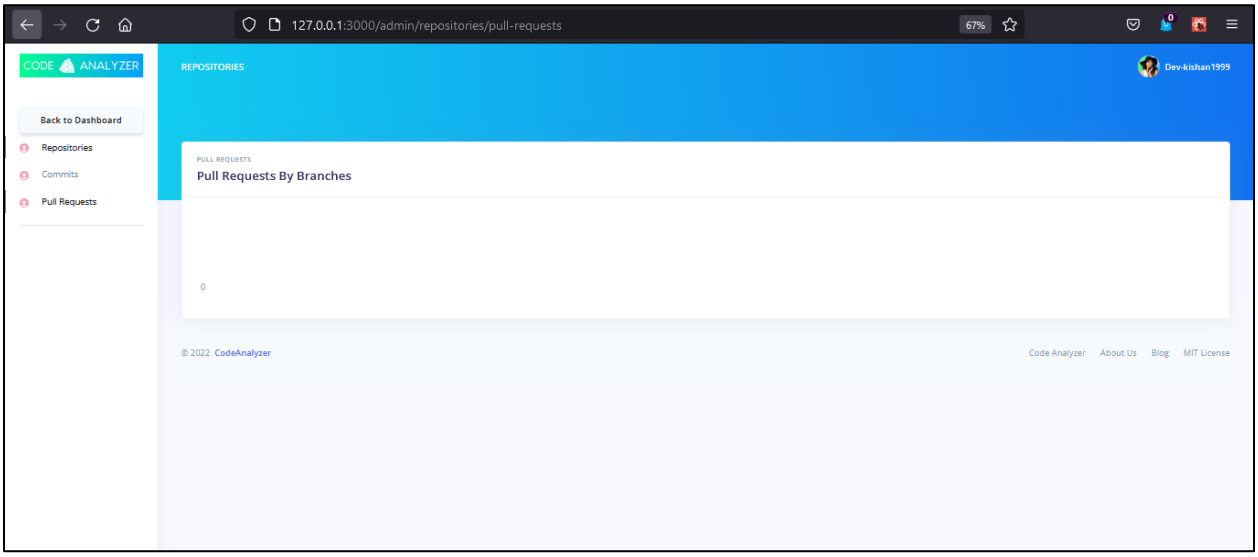
2.9.2 Repositories

This section represents the data for a single repository. In this section, different features such as language-wise contributors, the share of languages, most productive hours of the commits, and top contributors are shown



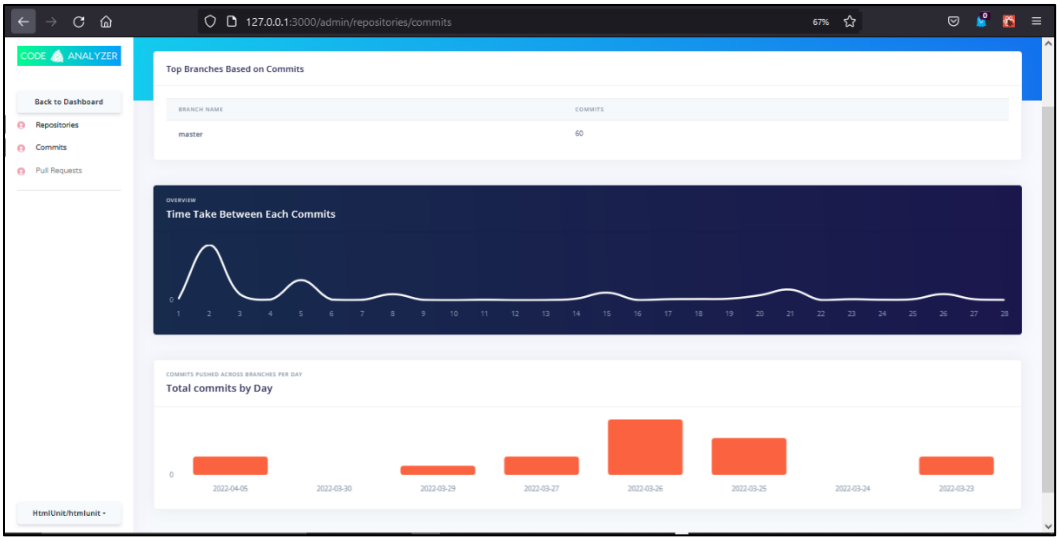
2.9.3 Pull Requests

This section represents the number of pull requests a branch consist of



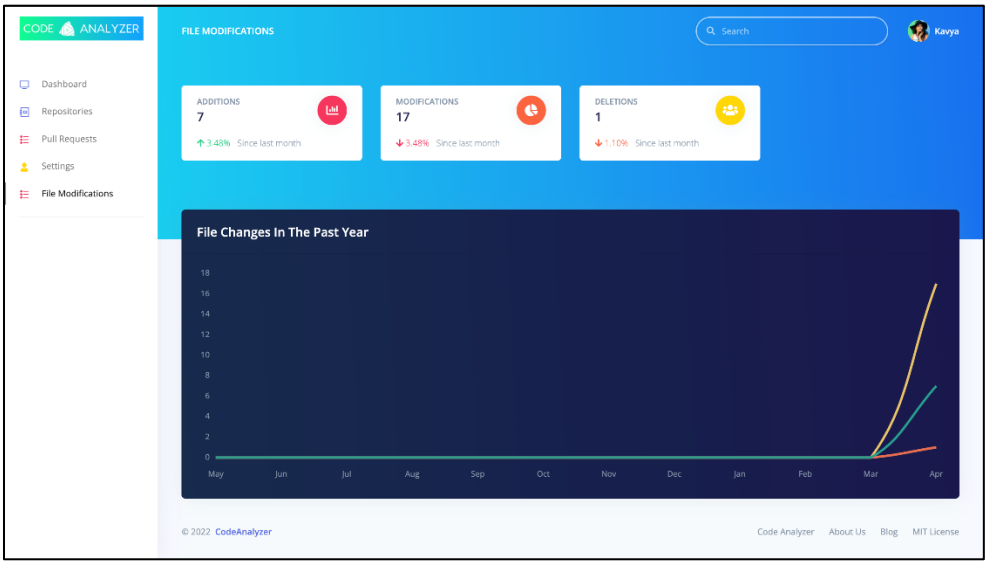
2.9.4 Commits

This section represents the top commits by the branch. The user needs to select the branch name. The overview of that branch name is shown in terms of charts. The line chart represents the Commits frequency, it represents the difference in the number of days between the commits. Another, chart represents the total number of commits done on a particular day for the selected branch



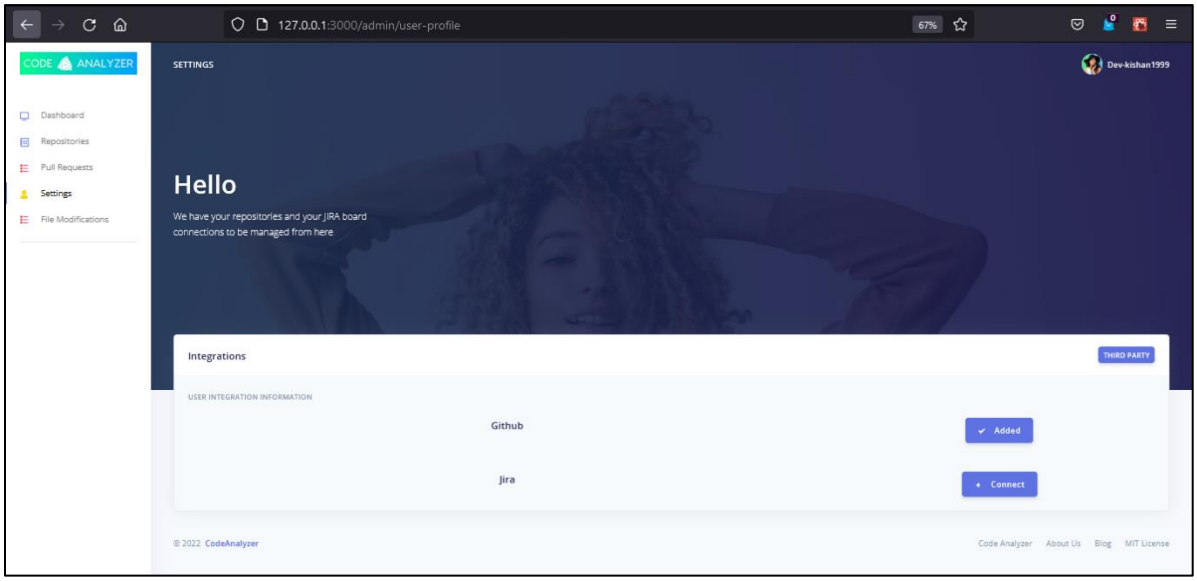
2.9.5 File Modifications

This section represents the 3 dashboards displaying the number of additions, modifications, or deletions performed in a file. The graph represents the file changes in the past year



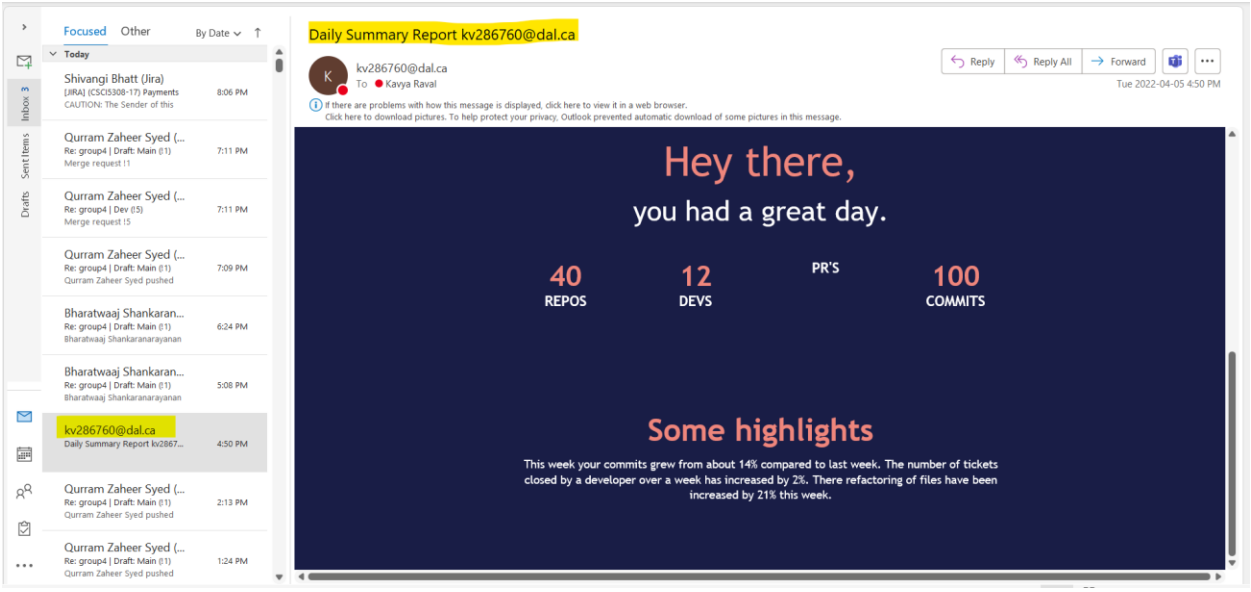
2.9.6 Settings

Settings dashboard provides an option for user to connect with JIRA or Github OAuth



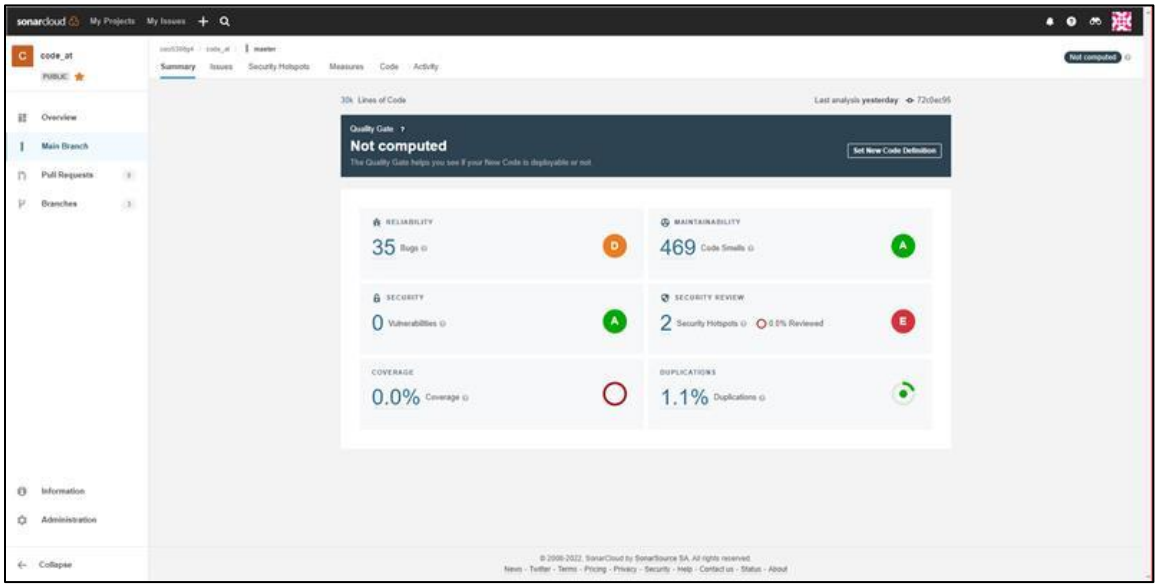
3. Cron Tasks

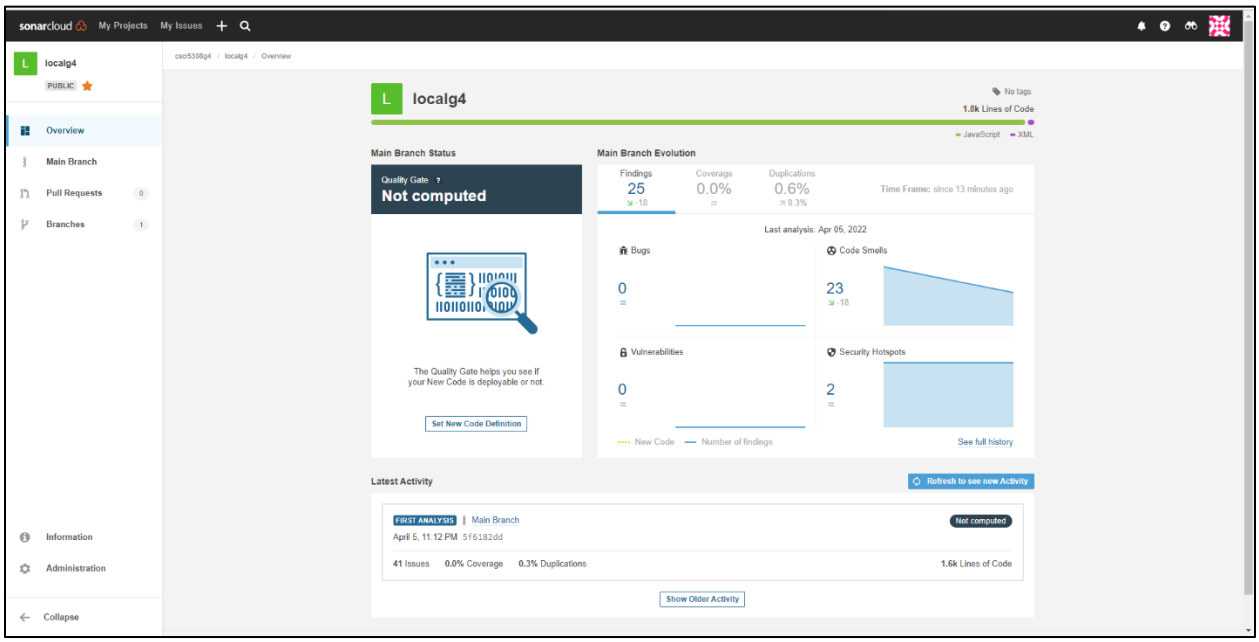
Code Analyzer has a feature that sends a summary report email daily at a prescribed time. That email consists of the number of repos, developments, Push Requests, and commits. Along with it provides the highlights which consist of an analytical comparison with the previous week's data. For example percentage increase in commits, increase or decrease in the number of tickets closed, and the number of files refactored in comparison to the previous week.



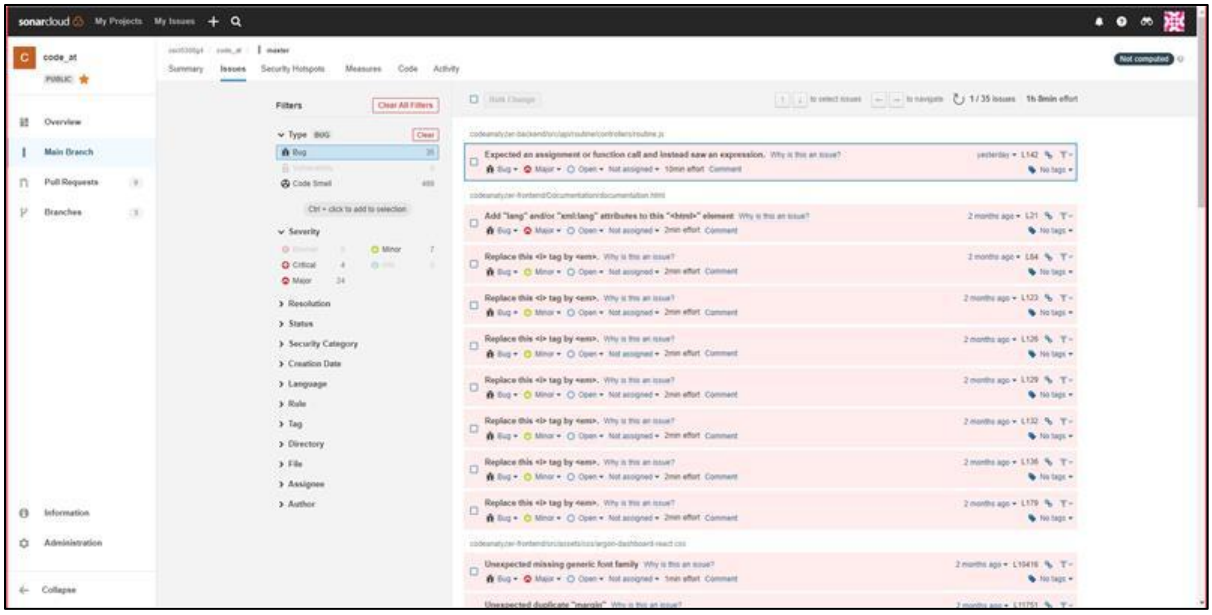
Smells Analysis

For smell analysis, we have used SonarQube which is an open-source tool, that analyzes source code and provides reports on the code quality of your project.





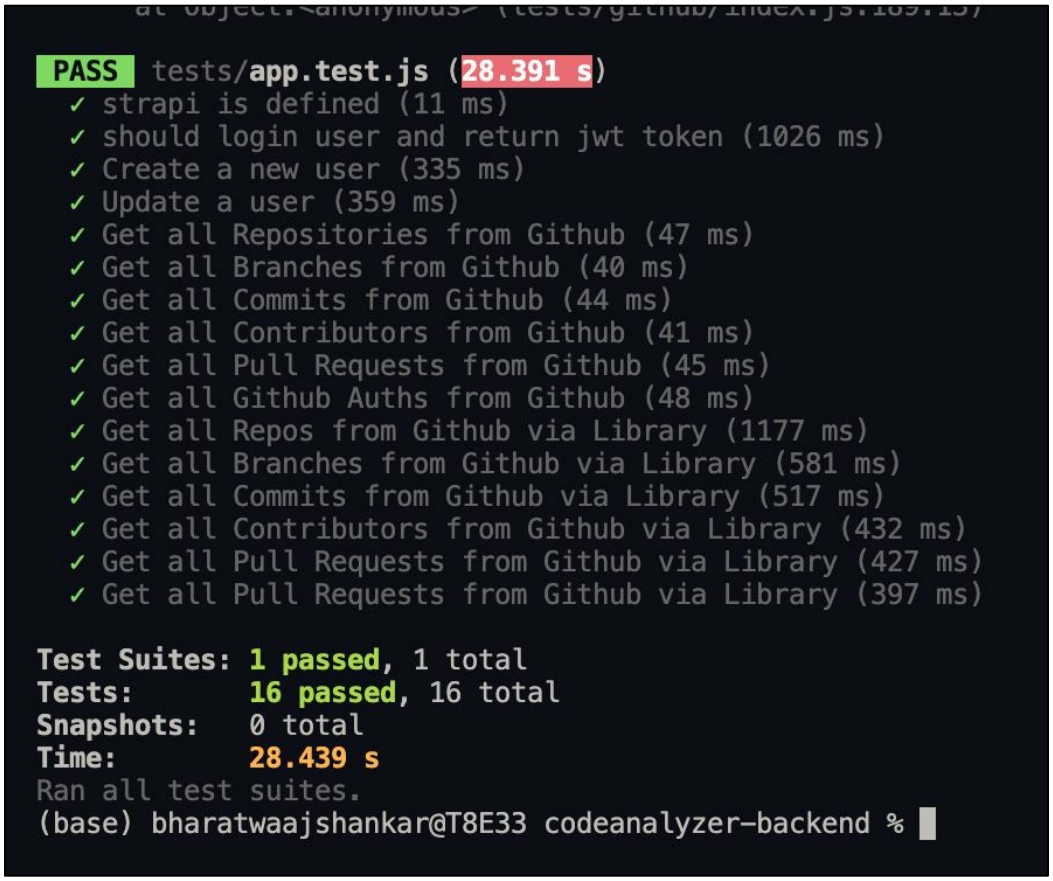
Encountered some smells in the backend and front-end. Resolved some backend issues, whereas major issues for frontend are because csv, thus, it cannot be resolved further


sonarcloud
My Projects
My Issues
code_at
PUBLIC
Overview
Main Branch
Pull Requests
Branches
Information
Administration
Collapse
Summary
Issues
Security Hotspots
Measures
Code
Activity
Not computed
Filters
Clear All Filters
Type
Bug
Code Smell
Severity
Minor
Critical
Major
Resolution
Status
Security Category
Creation Date
Language
Rule
Tag
Directory
File
Assignee
Author
1 / 35 issues
1h 0min effort
Expected an assignment or function call and instead saw an expression.
Add "lang" and/or "xml:lang" attributes to this "<html>" element
Replace this <i> tag by .
Unexpected missing generic font family


codeanalyzer-backend/config/database.js
Remove the unused function parameter "env" or rename it to "_env" to make intention explicit. Why is this an issue?
2 months ago L4
Code Smell Major Open Not assigned 5min effort Comment
No tags
codeanalyzer-backend/src/admin/webpack.config.example.js
Remove the unused function parameter "webpack" or rename it to "_webpack" to make intention explicit. Why is this an issue?
2 months ago L4
Code Smell Major Open Not assigned 5min effort Comment
No tags
codeanalyzer-backend/src/api/commit/controllers/commit.js
Remove the unused function parameter "next" or rename it to "_next" to make intention explicit. Why is this an issue?
2 days ago L11
Code Smell Major Open Not assigned 5min effort Comment
No tags
Remove the unused function parameter "next" or rename it to "_next" to make intention explicit. Why is this an issue?
2 days ago L23
Code Smell Major Open Not assigned 5min effort Comment
No tags
Expected a `for-of` loop instead of a `for` loop with this simple iteration. Why is this an issue?
2 days ago L46
Code Smell Minor Open Not assigned 5min effort Comment
No tags
Remove the unused function parameter "next" or rename it to "_next" to make intention explicit. Why is this an issue?
2 days ago L55
Code Smell Major Open Not assigned 5min effort Comment
No tags
Remove this useless assignment to variable "accessToken". Why is this an issue?
2 days ago L56
Code Smell Major Open Not assigned 15min effort Comment
No tags
Remove the declaration of the unused 'accessToken' variable. Why is this an issue?
2 days ago L56
Code Smell Minor Open Not assigned 5min effort Comment
No tags
codeanalyzer-backend/.../api/contributor/controllers/contributor.js
Remove the unused function parameter "ctx" or rename it to "_ctx" to make intention explicit. Why is this an issue?
2 days ago L11
Code Smell Major Open Not assigned 5min effort Comment
No tags
codeanalyzer-backend/.../api/github-auth/controllers/github-auth.js

Testing

Followed TDD approach, and performed about 50 test cases



Project Dashboard Stats

Sno.	Module	Features	Author	Status
1	Login	Github Oauth	Bharat	Completed
		Registration	Arshdeep	Completed
		Login by email	Arshdeep	Completed
2	Dashboard	Count of Repositories	Kishan	Completed
		Count of Contributors	Qurram	Completed
		Count of Refactored LOC	Bharat	Completed
		Average PR to main	Bharat	Completed
		Refactoring across Repository	Kishan	Completed
		Total commits by Repository	Kavya	Completed
		Designite Smells summary	Arshdeep	Completed
		Top 5 commits	Kishan	Completed
3	Repositories	Language wise contribution	Qurram/Kishan	Completed
		Share of languages	Qurram/Kavya	Completed
		Most productive hours	Qurram	Completed
		Top contributors by total changes	Qurram/Kavya	Completed
4	Commits	Top branches based on commits	Bharat/Kishan	Completed
		Time taken between each commit	Bharat	Completed
		Total commits by day	Bharat/Kavya	Completed
5	Pull request	Pull request by branches	Bharat/Kishan	Completed
6	File modification	File additions count	Arshdeep/Kavya	Completed
		File modifications count	Arshdeep	Completed
		File changes in past year	Arshdeep/Qurram	Completed
7	Miscellaneous	Jira board management	Kavya	Completed
		Documentation	Kavya	Completed
		Data modelling	Kavya/Kishan	Completed
		Scrum Master	Entire team	Completed
		MOM taker	Kavya	Completed
8	DevOps	Ci/CD pipeline	Qurram	Completed
		SonarQube Integration	Qurram	Completed
9	Additional Acheivements	Designite Integration	Arshdeep	Completed
		Dashboard Integration across whole repo	Entire team	Completed
10	TDD	Test Cases	Entire team	Completed