GSOC 2022 Proposal

AboutCode

Lali Akhil Raj (lf32)

Scancode.io / Scancode Toolkit: Create Web Application to scan and review a single license text

April, 2022

About Contributor

Name: lali akhil raj

Github Username: lf32

Email: lali.akhilraj.cse19@itbhu.ac.in

Country Of Residence: India

Timezone: IST (UTC +5:30)

Personal Projects:

- **Kechup**

  Django Based WebApp to manage orders at restaurants

- **Gsocorg**

  Django Based WebApp to view organizations that participated in GSoC >>> View Deployment

- **Url96**

  Django Based WebApp to shorten large URLs

- **Eyelighter**

  NodeJS based WebApp to classify images of Animals

- **Bloodbase**

  Spring Boot based WebApp to administer blood transfusion data

Project Info

TITLE

Scancode.io / Scancode Toolkit: Create Web Application to scan and review a single license text

ABSTRACTION

The aim of this project is to create a django based web application to solve the issue of scanning and reviewing a single license text. The application will be designed around Django-REST-framework and it will be integrated with scancode-analyzer to find possible issues automatically, besides that it will allow the report of license detection integrated in the app based on the results and this is a **medium** size project idea.

Project Description

ABOUT Scancode-Analyzer

Scancode-Analyzer is a python module that helps in detection of various licenses, copyrights, package manifests and direct dependencies and more both in source code and binary files

- **Installation (PIP)**

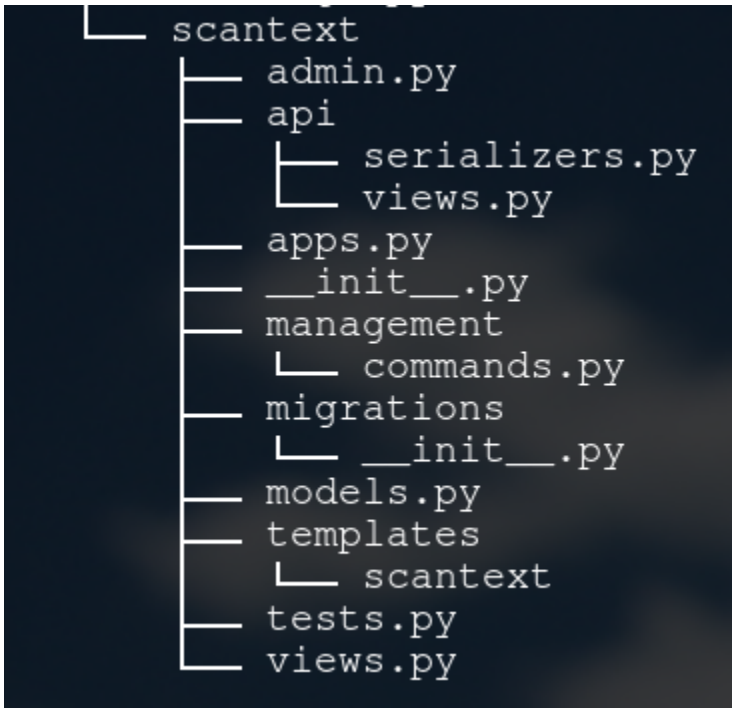
  pip install scancode-analyzer

- **Usage (CLI)**

```
scancode -l --json-pp output.json /path/to/scan_files/
--license-text --classify --analyze-license-results
```

- **Developing Scantext**

At first, we will create an app with django-admin called **scantext** then do as follows



scancode.io/

- urls.py

```
path("", include("scantext.urls"))
```

-scanpipe/

-scantext/

- urls.py & views.py

```
# view the result of the current scan
path("result/",
     views.scantext_result,
     name="scantext_result")
# download the scan results from history
path("history/<uuid>/download",
     views.scantext_history_download,
     name="scantext_single_history_download"),
# details about individual scan
path("history/<uuid>/",
     views.scantext_history_scan,
     name="scantext_single_history"),
# table view of all scans
path("history/",
     views.scantext_history,
     name="scantext_history"),
# home page to accept text to scan
path("",
     views.scantext,
     name="scantext_scan"),
```

- models.py

```
class Text(UUID, ExtraDataFieldMixin, models.Model):
    """
    Create a text model with variables
    scan_id,
    is_file(bool),
    file_name,
    scan_text,
    scan_date,
    scan_results,
    scan_reported(bool)
    """
```

api/

- __init__.py
- serializers.py

```
class ScantextSerializer(serializers.ModelSerializer):
    """
    This class uses the Model `Text` as the scan serializer
    """
```

- views.py

```
class ScantextViewSet(viewsets.GenericViewSet):
    scans = Text.objects.All()
    serializer_class = ScantextSerializer

    def results():
        """
        Returns results compatilbe to scancode data format
        as a stream of text JSON format using JSONResultsGenerator
        """

    def results_download():
        """
        Returns result as attachment
        """
```

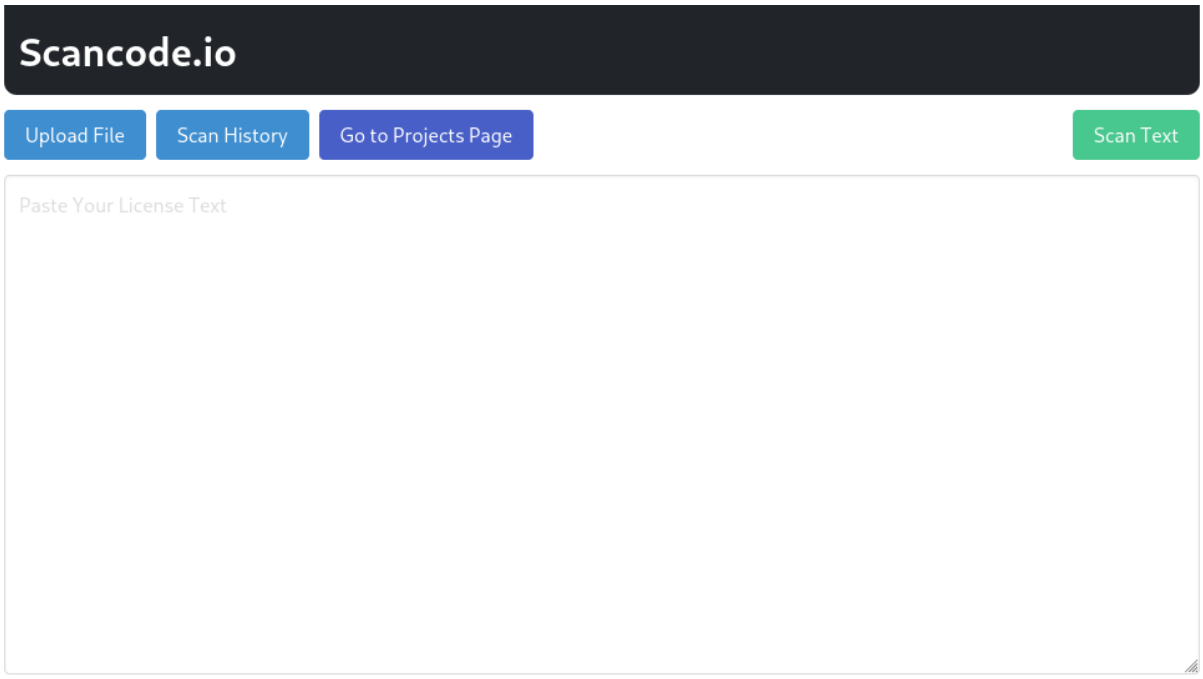
management/

- commands.py

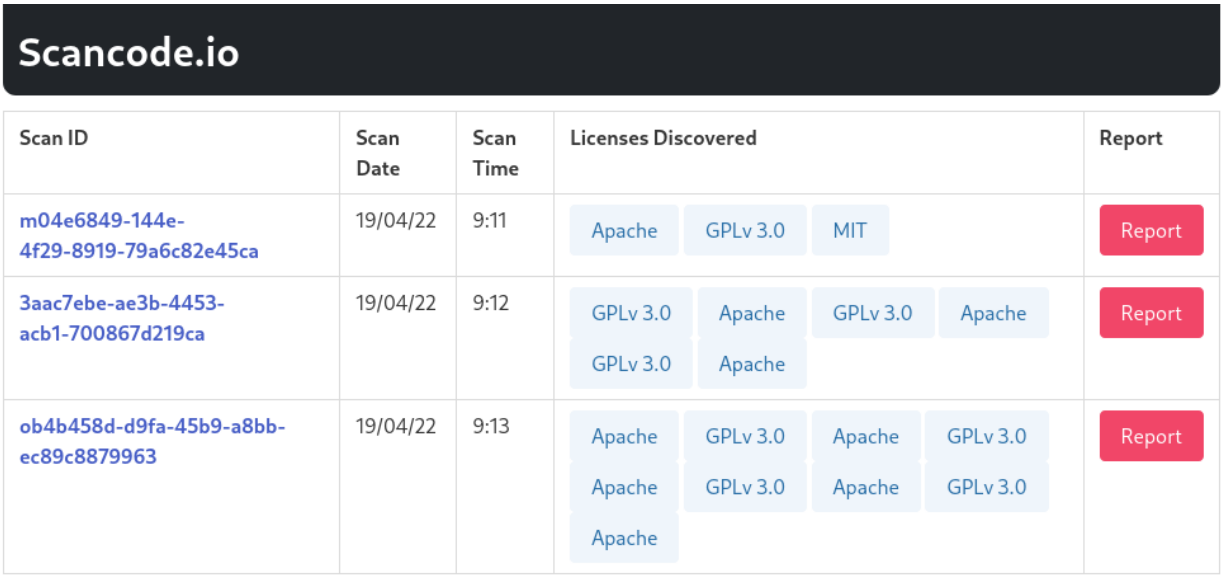
```
class ScanTextCommand(BaseCommand):
    """
    ScanText command takes in input text and executes the scancode-anyalyzer
    """

def validate_input_file(file_location):
    """
    Checks for valid input file path
    """
```

tests/{ test_views.py, test_model.py, test_commands.py, test_api.py, ...}

Example templates



Home page to scan text (html textarea/ace editor)



History page with table of all scans

for detailed view visiting the uuid path page will give more information

Project Timeline

- **Community Bonding Period (May 20th - June 12th):** Interact with the devs and discuss the required features and propose a well designed UI and API
- **Coding Period (June 13th - September 12th)**
    - **Week 1:** Implement Templates for the project
    - **Week 2:** Implement the Views and write unit tests
    - **Week 3:** Implement the Models and write unit tests
    - **Week 4 - 5:** Implement the API Views and Serializers
    - **Week 6:** Implement the Tests for the API
    - **Week 7 - 8:** Integrate Scancode-analyzer with scancode.io
    - **Week 9 - 10:** Implement results download option and feedback functionality
    - **Week 11:** Cleanup bugs and implement features
    - **Week 12:** Write more tests for Implemented features
    - **Week 13:** Meet all the requirements for the Project and submit final product
- **Post GSoC:** Cleanup as many bugs as possible and improve the documentation.

Project FAQ

- **Which Project am I applying for?**

  Scancode.io: Create web application to scan and review a single license text

- **How is this project going to help the Community ?**

  A new web application supporting JSON Rest API in Django to Scan the text uploaded by the user and figuring out what kind of license does the text represent would help the end-user to get their work done as quickly as possible

  Adding Support for **Scancode-Analyzer** to detect licenses, copyrights, package manifests and direct dependencies and more both in source code and binary files.

- **Am I the right person to work on this project?**

  Yes, I think I'm because I am good with python and javascript and I especially have keen interest in Django based application development and frontend UI development. I am also familiar with GIT. So It can be a cakewalk for me in developing software for scancode and most importantly I can learn more and dive deep into MVC based architecture.

- **Any contributions made to nexB/ScanCode.io?**

  Pull Requests

  #422 Merged: Close btn works as expected without archiving the project (fixes **#421**)

  #425 Merged: license/ web page renders without 500 error

  Issues Raised

  - **#421 Closed :** close button doesn't work as expected during the archive process
  - **#426 Open:** don't show the editor for archives and directories