Project Proposal

Developing Testing Harness For PGWEB

SYNOPSIS

PGWEB is the official repository containing the code of the website at https://www.postgresql.org/ The website code is written in Python using the Django framework. However, this infrastructure lacks a testing suite to check for the correct functioning of the website. Testing is an essential part of the development process and will help immensely while developing the website further in the future and check for breaking changes.

This project aims to integrate a complete testing suite into the current CI/CD pipeline, including unit tests to test for the view, models, forms, and links, integration tests, functional tests for cross-browser compatibility testing, performance testing, and accessibility testing.

DELIVERABLES

- Create a separate tests module in the root of the repository.
- Write unit tests for all the apps' views, models, and for other necessary logic.
- Integration tests to further check for working of forms, cookies, and interacting pages.
- Functional tests to simulate a real browser environment and to test the website's compatibility and working across all major browsers.

- Performance tests to ensure the website's speed and performance.
- Accessibility tests to make sure the website is accessible to all users and is user-friendly.
- Write the CI/CD workflow for GitHub actions to run the tests as required.
- Update the documentation about the changes made and how to test after contributing.

APPROACH

Unit Tests, Integration Tests

For unit tests and integration tests that will test for the correct functionalities and working of each app's URLs, views, models, and/or forms. Other logic can be checked as required. For this, Django provides its own testing tools which can be used. "**Pytest"** is a library that I propose can be used for these purposes as it provides some additional features over the standard Django testing. "**pytest-django**" is a plugin for pytest which is used for testing Django applications. Similarly, the integration tests can be run.

Functional Testing

These tests will simulate a browser instance and check the proper working of forms, navigation, HTML and CSS. Also, test to check whether cookies are working as expected or not. We can use "**Selenium**" for these tasks. It is open source and is ideal for automated testing of web applications across different browsers like Firefox, Chrome, Internet Explorer, and Microsoft Edge. Thus this can be used to check for cross-browser compatibility too.

Performance Testing

In this, different approaches can be used to test as required. One way is to use a middleware named "**django-performance-testing**"; this is used to set limits on different factors such as query count, the time it ran for, reads and writes, etc. The other approach is about measuring these parameters like execution time, the number of functions called, memory usage, etc. instead of setting a limit on them. Various packages can be used for monitoring this, one of them being “**SILK**”. It is a middleware for Django applications and can be used in monitoring the app’s performance.

Accessibility Testing

Web accessibility is all about making sites and applications that everyone can use, especially people with disabilities. There are multiple websites and tools that can be used to check for compliant percentages. https://www.w3.org/WAI/ER/tools/ contains a list of tools.

REFERENCES

- https://djangostars.com/blog/django-pytest-testing/
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Testing
- https://www.sitepoint.com/how-to-use-selenium-webdriver-for-cross-browser-testing/
- https://www.toptal.com/python/performance-optimization-testing-django
- https://www.w3.org/WAI/ER/tools/

KEY TECHNOLOGIES

- pytest-django
- pytest
- Selenium
- django-performance-testing
- django-silk
- accessibility

TIMELINE

Pre GSoC Period and Community Bonding Period

- Get myself accustomed to the codebase and discuss the design of the module.
- Interact more with the community by regularly discussing the ideas and tools/libraries to be used that I put forward and discussing which ones to use.
- Get accustomed to submitting a patch and general workflow.
- Learn more about testing, especially with Python/Django.
- Try and resolve existing issues or bugs to get familiar with the workflow.
- Finalize the modifications (if any) that need to be done by discussing with the community through the mailing list.

Coding Period

- **Week 1 (June 7 - June 13)**
    - Create the test module and download the required libraries and tools.
    - Update requirements.txt
    - Start writing unit tests for the apps' URLs, views, and models.

- **Week 2, 3 (June 14 - June 27)**
    - Continue with writing the unit tests.
    - Try completing the unit tests.

- **Week 4 (June 28 - July 4)**
    - Complete unit tests.
    - Start writing integration tests.
    - Update documentation.

- **Week 5 (July 5 - July 11)**
    - Complete integration tests.
    - Start with functional testing and Selenium.
- **Week 6 (July 12 - July 18)**
    - Mentor Evaluations.
    - Work for cross-browser testing.
    - Check forms and cookies working.
- **Week 7, 8 (July 19 - August 1)**
    - Complete any tests remaining above
    - Complete performance tests.
- **Week 9, 10 (August 2 - August 16)**
    - Complete automated accessibility tests
    - Complete any remaining work
    - Submit code for final evaluations
    - Update documentation.

Post GSoC

I plan to contribute further to other projects as well. I also want to review submissions on commitfests and help contribute. Other projects which I want to work on include pgarchives and pgfarm. Eventually, I wish to contribute to the core PostgreSQL SQL server repository as well. I would like to contribute to PostgreSQL's projects as much as I can.

ABOUT ME

General Information	
Name	Chirag Kasat
Institute	Sophomore at Birla Institute of Technology and Science, Pilani - K.K. Birla Goa Campus
Course	Bachelorette of Engineering, Computer Science
Contact Information	
Email Address	chiragkasat0@gmail.com
Website Link (Portfolio)	chiragkasat.com
Github, LinkedIn Links	ChiragKasat · GitHub , Chirag Kasat - LinkedIn
Timezone	Asia/Kolkata (UTC+5:30)

WHY ME?

I am familiar with web development and have worked previously on multiple projects and have experience with test-driven-development style. I have good experiences in technologies like JavaScript, Python using frameworks like Node and Django. I have started contributing to open source projects since last year and I especially like to contribute to projects that I like using in my projects. PostgreSQL is my go-to database for my projects and I am really happy to contribute to it. I have been diving into the codebase and getting myself accustomed to the community since March.

I have experience in writing tests and Django, also I am willing to put in the efforts to learn the required technologies and tools used. I have worked in Python programming and have used it in some web projects as well as machine learning projects, and have learned various concepts such as OOP, algorithms, etc. This experience has made my skills stronger and I think I will prove to be a valid candidate for completing this project if selected. I plan to contribute to this and also other projects in the future on regular basis.

COMMITMENTS

I have my university end semester examinations from May 01 2021 to May 13 2021 during which I will not be able to dedicate much time to this project. Apart from that during the official coding period, I have a family function due to which I will be busy on the 1st and 2nd of July but I will make sure that I am on or ahead of my defined schedule. There are no other commitments during this period and will be able to devote my entire time towards the completion of the project.

WHY DID I CHOOSE THIS PROJECT? (BENEFITS TO THE COMMUNITY)

This project aims to add a testing suite to the project thus improving productivity and efficiency. Reviewing times for future contributions will be reduced and bugs caused by changes if any can be easily rectified then and there. Having prior experience in developing websites I found this project interesting. Browser compatibility tests will ensure the website provides the same experience and look across all browsers which users can use. The performance tests will help to keep the website's stability speed and productivity in check and identify issues. Accessibility tests will make sure the website is user-friendly and accessible to everybody. The addition of further tests can be done while developing the website further in the future.