



Google Summer of Code

**Google Summer of Code 2022
Project Proposal for PostgreSQL**

**GUI representation of monitoring System Activity
with the system_stats Extension in pgAdmin 4**

GSOC Contributor: Kunal Garg

Email Id: gargkunal02@gmail.com

Country of Residence: India

Timezone: India/UTC+5:30, Indian Standard Time

University: [Thapar Institute of Engineering and Technology, Punjab, India](#)

Degree: [Bachelor of Technology in Computer Engineering](#), 2023

Primary Languages: English and Hindi

Social Handle: [GitHub](#), [LinkedIn](#)

Index of Contents

1. Overview	1
1.1. Project Abstract	1
1.2. Benefits to Community	1
2. Goals and Deliverables	1
2.1. Deliverables	1
2.2. Detailed Description	1
2.3. Outline of Approach	2
3. Schedule of deliverables or work plan	5
3.1. Application Review Period (April 19 - May 20, 2022)	5
3.2. Community Bonding Period (May 20 - June 12, 2022)	5
3.3. Coding (June 13 - Sept 12, 2022)	6
4. Past Experience	7
4.1. Academic Details	7
4.2. Personal/Open Source Projects	7
4.3. Motivation	8
4.4. Why me?	8
5. Availability Schedule and Post GSoC	9
5.1. Working hours	9
5.2. Regular Updates and Meetings with Mentor	9
5.3. Other commitments	9
5.4. Post GSoC	9

1. Overview

1.1 Project Synopsis

The expected outcome of the project is to have a GUI representation using charts and graphs of the system level statistics on the dashboard in pgAdmin 4 using the system_stats extension. This will allow users to interact with system metrics of the PostgreSQL database server of their projects. All the procedures provide profound details of important performance indicators related to OS, CPU, Memory, Disk, I/O, networking, etc. A comprehensive dashboard will be designed and developed to optimally represent all the system statistics that are relevant to the users, keeping the look minimal, simple, and consistent.

1.2 Benefits to Community

1. These stats are important in displaying the usage of resources of the OS.
2. The user will get an analysis of CPU performance and I/O completion with system interrupt status.
3. The user will get an in-depth understanding of memory usage and disk information.
4. The user will be able to track the network interface information.
5. The user will know how much memory is consumed by different processes running on the system.

2. Goals and Deliverables

2.1 Deliverables

1. Design and development of a dashboard consisting of graphs and charts in pgAdmin 4
2. Showcasing system statistics for monitoring Postgres activity by integrating system_stats in pgAdmin 4
3. Consistent and user-friendly GUI dashboard for monitoring System Metrics of the database server
4. Comprehensive analysis of system activities based on OS, CPU, Memory, I/O, Networking, etc

2.2 Detailed Description

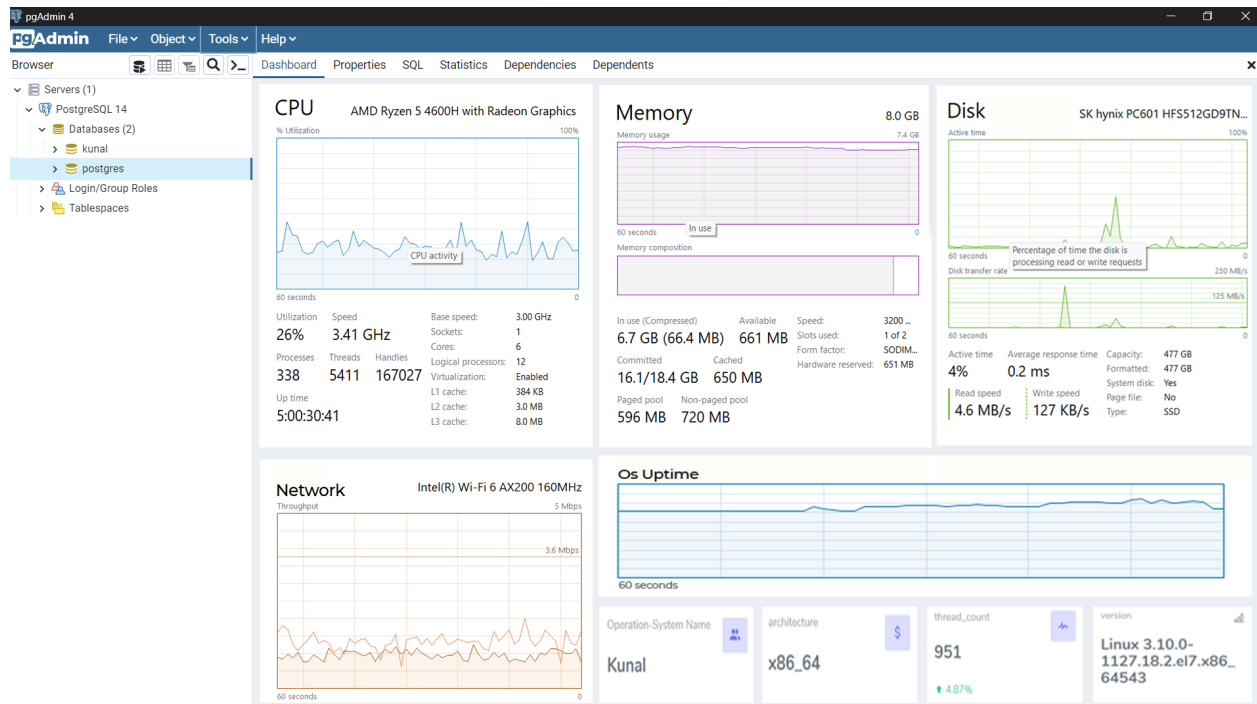
2.2.1 Dynamic system statistics to be displayed on the dashboard

System stats is an extension that fetches and displays information related to CPU performance, OS information, Memory usage, IO completion, and Computer Networking. All of this information is accessed with the use of 10 functions that allow the user to get the following data about the system:

SN	Function	Data returned	Data types
1	pg_sys_os_info Operating system statistics	Name, Version, Host name, Domain name, Handle count, Process count, Thread count, Architecture, Last bootup time, Uptime in seconds	Textual Numeric Temporal
2	pg_sys_cpu_info CPU information	Vendor, Model name, Processor type, Logical processor, Physical processor, Number of cores, Architecture, Clock speed in hz, CPU type, CPU family, Byte order, cache size (L1d, L1i, L2, L3)	Textual Numeric Frequency Size
3	pg_sys_cpu_usage_info CPU usage information of time spent on all operations	usermode normal process, usermode niced process, kernel mode process, idle mode, io completion, servicing the interrupt, servicing software interrupt, user time spent, processor time spent, privileged time spent, interrupt time spent	Percent time spent processing
4	pg_sys_memory_info Memory usage information	Memory: Total, Used, Free, Total swap, Used swap, Free swap, Total cache, Total kernel, Kernel paged, Kernel non paged.	Memory in bytes
5	pg_sys_io_analysis_info I/O analysis of block devices	Total number of reads, Total number of writes, Read bytes, Written bytes, Time spent in ms for reading and writing	Numeric Bytes Temporal
6	pg_sys_disk_info Disk information	File system of the disk and type, Mount point for the file system, Drive letter, Drive type, Total space in bytes, Used space in bytes, Available space, number of total inodes, used inodes, and free inodes	Textual Numeric Bytes
7	pg_sys_load_avg_info Average load	Load Average of 1, 5, 10 and 15 minutes	Temporal
8	pg_sys_process_info Process information	Number of total, running, sleeping, stopped and zombie processes	Numeric
9	pg_sys_network_info Network interface information	Interface name, ipv4 address of the interface, Interface speed in mbps; Number of total bytes transmitted, total packets transmitted, transmit errors by this network device, packets dropped during transmission, total bytes received etc	Textual Numeric Speed
10	pg_sys_cpu_memory_by_process CPU and memory info for each process ID	PID of the process, Process name, CPU usage in bytes, Memory usage in bytes, Total memory used in bytes	Numeric Bytes

2.2.1 Proposed Layout

The data from the aforementioned functions can be displayed on the dashboard as shown in the diagram.



NOTE: This diagram is just to give an idea of the layout and is subject to change with further deliberations to further enhance the layout keeping the user's ease at utmost priority.

To maintain consistency and user-friendliness, pgadmin's current frontend will be extended. Here are a few alternatives that can be implemented:

1. A new tab for the system statistics dashboard, upon navigating there all the data from different functions will be displayed in depth.
2. The current dashboards tab can be extended further to include the system statistics as well by adding a show more button or a scroll down feature, by extending the pgAdmin frontend.

The essential and relevant information will be displayed to the user taking into account high levels of user experience and a competent user-friendly GUI. The fetched data will be represented in a visually appealing way with the assistance of charts and graphs in the views on the template. All data types will be displayed in the most suitable form of illustration, easily understood by the user.

2.3 Outline of Approach

2.3.1 Steps already Initiated

1. The Pgadmin4 Github repository was successfully cloned and the system_stats extension was installed.
2. They were individually tested and were working fine after certain installations and environment setup.

2.3.2 Steps to take during the standard coding period

system_stats extension is a library of stored procedures that allow users to access system-level statistics for monitoring Postgres activity. The extension has been installed in the local server, and the setup is configured following the given steps:

1. **The extension is built using the PGXS framework** by unpacking the file archive in a suitable directory. The directory containing the **pg_config binary** for PostgreSQL installation is included in the PATH environment variable. It is compiled and the code is installed.

2. **Installing the Extension**

The extension is installed in the database using the following SQL command.

```
CREATE EXTENSION system_stats ;
```

3. **Security**

Access is restricted to superusers and members of the **monitor_system_stats role** which are created on installation, due to the confidential nature of information returned. To allow users to access the functions without granting them superuser access, they are added to the monitor_system_stats role by using the following command.

```
GRANT monitor_system_stats to "dash" ;
```

After the setup is complete, information can be fetched by running the SQL queries in the database. The system has access to two servers namely client and server. To fetch the relevant information at runtime, add all the queries to **dashboard_stats.sql** present in the following directory in the repository: `D:\Kunal\Work\Gsoc\Pgadmin\pgadmin4\web\pgadmin\dashboard\templates\dashboard\sql\default`

The queries to be added are the ones mentioned in the [detailed description](#) section. For example, we can add the following command in the aforementioned file:

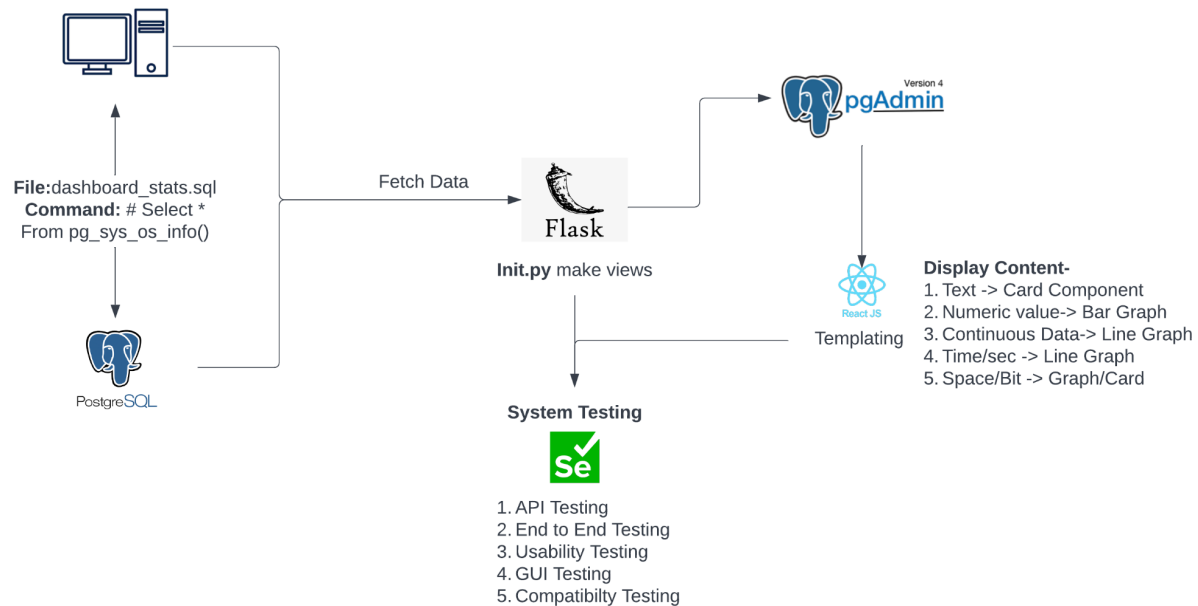
```
# SELECT * FROM pg_sys_os_info(); and get the following output
```

name	"Red Hat Enterprise Linux"	+
version	Linux 3.10.0-1127.18.2.el7.x86_64	
host_name	linux2	
domain_name	(none)	
handle_count	9280	
process_count	336	
thread_count	951	
architecture	x86_64	
last_bootup_time		
os_up_since_seconds	536186	

As shown above, data from all available functions in the system_stats extension will be called to fetch the data relevant to the user and displayed on the dashboard as shown in the proposed layout.

All the **SDLC** principles will be followed while development and the system built will have highly cohesive and lowly coupled units that will be integrated and ensured to work properly with the already existing code. In the **STLC** phase, it is essential to implement tests to verify and validate if the system meets the software requirements. Postgres tests its application for the given project in the **features_test directory**; a similar approach will be utilized for the following inevitable testing requirements. First and foremost, **API testing** will be executed to test the connection between pgadmin4 and system_stats. **End-to-End testing** of the software will be implemented in multiple iterations, to test the structure of the source code by **unit testing** every module, **integration testing** to ensure that separate modules work properly as a whole, and most importantly **regression testing**; a necessity to ensure seamless integration of my code with the existing systems developed in the open-source project. **Usability testing**, a part of operational readiness will be done to ensure user comfort as its primary focus, to further enhance user-friendliness **GUI testing** will also be implemented using the **automation suite Selenium and PyTest**. **Functional testing** will signify if the system is working as intended and thorough non-functional testing will be applied in the form of **compatibility testing** taking into account browser, hardware, network, OS, and database compatibility to ensure the software provides the same experience across different settings.

To Summarize



3. Schedule of deliverables or Work Plan

3.1 Application Review Period (April 19 - May 20, 2022)

This time will be utilized to polish skills relevant to the project which will help in making the solutions better and more accurate. Also, contact will be maintained with the mentor to enhance and work on the idea and its implementation.

3.2 Community Bonding Period (May 20 - June 12, 2022)

This time will be utilized to interact with the mentors and set up feedback loops; and continue the process of refining the plans and the workflow for the project in consultation with the mentors. The plan devised, consists of the path and procedure for the solutions in the proposal but they are flexible and subject to change as per the continuous dialogue with the mentor following agile principles. Also, extensive research on implementations and methodologies for the set goals to make the solutions easier will be done. Work will be initiated in this period to complete the tasks before the stipulated deadlines. Also, the opportunity of getting involved and network with other members of the community will be utilized.

3.3 Coding (June 13 - Sept 4, 2022)

Week 1 (June 13 - June 20, 2022)	Begin setup Add support for the system_stat extension in pgadmin4 API Testing
Week 2 (June 20 - June 27, 2022)	Fetch data from functions, implement views and templates in flask
Week 3 (June 27 - July 4, 2022)	Design and Develop the front-end of dashboard Create the user interface
Week 4 (July 4 - July 11, 2022)	Integrate views with template (combine frontend and backend) Publish blog 1
Week 5 (July 11 - July 18, 2022)	Buffer time to complete delayed tasks if any End to End Testing (unit, integration and regression testing) GUI Testing
Week 6 (July 18 - July 25, 2022)	Refactor the code Increase modularity of code Update documentation
Week 7 (July 25 - Aug 1, 2022)	Submit Phase 1 Evaluation Publish blog 2
Week 8 (Aug 1 - Aug 8, 2022)	Work as per the review from mentor post evaluation
Week 9 (Aug 8 - Aug 8, 2022)	Work on template components from user's perspective after conducting usability testing
Week 10 (Aug 15 - Aug 22, 2022)	Improve the user interface of the dashboard Generate test reports using TestNG
Week 11 (Aug 22 - Aug 29, 2022)	Final Touches Production-level integration and deployment
Week 12 (Aug 29 - Sept 5, 2022)	Functional (White Box) testing and implementation of corrections if any, Non functional compatibility testing Publish blog 3
Week 13 (Sept 5 - Sept 12, 2022)	Submit Code and Final Evaluations

NOTE: The Documentation updates (if any) will go on parallelly with feature work. Alongside, a 3-part blog will be maintained, writing about work, the entire workflow, and evaluations. There are some buffer days in the schedule, giving time to complete any delayed tasks. At all times Software Development and Testing Life Cycle will be followed to the T maintaining strong communication using SCRUM.

4. Past Experience

4.1 Academic Details

I am a third-year student pursuing computer engineering at Thapar Institute of Engineering and Technology, Patiala. I have a consistent CGPA of 9 above on a scale of 10 and am a recipient of a merit-based scholarship. I am a quick learner, capable of learning new things in a relatively short amount of time. I am a full stack developer with knowledge of and experience in frontend technologies like HTML, CSS, JS, jQuery, Bootstrap, React, etc, and backend technologies like Django, Flask, NodeJS, PHP, etc, and Database technologies like PostgreSQL, MongoDB, dbsqlite3, etc. I know how to deploy applications using AWS, Azure, GCP, Cpanel, etc. I am proficient in the tech stack relevant to the given project. Also as a computer science student, I have a strong grip on the concepts related to Operating Systems, Computer Networking, Computer Architecture and Organization, Database Management Systems, Microprocessors, Software Engineering, etc which gives me an edge in understanding the data to be displayed on the dashboard.

4.2 Personal/Open Source Projects

To name a few:

1. **Uwal:** <http://www.uwal.in/>, <https://github.com/Gargkunal02/Uwal>

Overview: A production-level website catering to the business of a pharmaceutical company.

Tech Stack: Django, HTML, CSS, JS, Twillio, PostgreSQL

2. **Saic:** <https://www.saictiet.com/>,

Overview: An official website for student interaction with Thapar University Alumni.

Tech Stack: Django, HTML, CSS, JS, dbsqlite3, Azure

3. **Smart India Hackathon 2022:** <https://github.com/Gargkunal02/SIH-2022>

Overview: A system using speech recognition and NLP to get downloadable summaries and main keywords of recorded audios. The important keywords are color-coded in the summary according to their score level to access them easily.

Tech Stack: Django, PostgreSQL, HTML, CSS, JavaScript, APIs, Neural network, and Deep Learning

4. **FaceUnFace:** <https://github.com/Gargkunal02/FaceUnface-2.0>

Overview: It is a support group for teens faced with mental health challenges

Tech Stack: HTML, JS, CSS

5. **Humara Ansh:** <https://github.com/Gargkunal02/Humara-ansh-2.0>

Overview: A portal that connects expecting parents with doctors, patients can access all their reports sorted in time and prescribed medicines can be bought directly, parents also get a customized feed of educational content.

Tech Stack: Django, HTML, CSS, JS, dbsqlite3

6. **Dr. Flight Mode:** <https://github.com/Gargkunal02/Dr.flight>

Overview: A portal that connects patients, laboratories, and doctors. A track of conducted tests is maintained here. (ongoing)

Tech Stack: Django, HTML, CSS, JS, React

7. **Doctor Mukesh Nigam** (To be deployed)

Overview: A portal for connecting with Dr. Mukesh Nigam and scheduling appointments.

Tech Stack: Django, HTML, CSS, JS, Twillio, PostgreSQL

4.3 Motivation

I have worked on various projects that use PostgreSQL as a production database. The idea of contributing to an organization whose products and services are used globally is a motivation in itself. I have been working with Django, Flask, Python, and PostgreSQL for a considerable amount of time, having prior experience makes this project interesting and relatable. The chance to be involved professionally in a project at a level as colossal as PostgreSQL is more than I can ask for as it is a great learning opportunity and advancement in my career. This project will help me to equip essential real-world software skills as well as soft skills which I believe will help me a lot in the longer run.

4.4 Why me?

I highly value this opportunity and getting it means a lot to me. I guarantee that I will never be careless and lethargic throughout three months and put in 100% effort towards its completion. This project is apt for me as it involves constant learning of new technology. I am consistent with my efforts and never leave any task in between, no matter how hard or new it is for me. I make sure that I keep researching the issue at hand and then work to make it happen. I think this skill of mine proves to be very helpful to achieve feature addition and bug solving for this particular project.

5. Availability Schedule and Other Commitments

5.1 Working hours

I can commit 40 or more hours weekly to achieve my goals and deliverables. If need be I can spend more time getting things done.

- **Work Timings for weekdays (5 - 6 hours daily)**
7 pm IST to 1 am IST (1:30 pm UTC to 7:30 pm UTC)
- **Work Timings for weekends (7 - 8 hours daily)**
10 am IST to 6 pm IST (4:30 am UTC to 12.30 pm UTC)

5.2 Regular Updates and Meetings with Mentor

1. Daily progress reports will be mailed by the end of every working day and code will be regularly committed on Github following the guidelines.
2. During working hours I will be available on slack or any other platform suitable for communication.
3. Regular SCRUM meeting to report progress, finalize next steps and acquire guidance on impediments. I will discuss doubts if any, or alternative methods for solving a subtask better.

5.3 Other commitments

I have my university end semester examinations from 20th May to 6th June 2022 during which I will be able to dedicate less time to the project. I will make sure that I am on or ahead of my stipulated schedule. There are no other work commitments or vacations planned in the given duration and I will be able to devote my entire time to the completion of the project.

5.4 Post GSOC

I believe that open-source contribution is not just restricted to Google Summer of Code. I would love to continue working on the developments and enhancements of the project post-GSoC. Eventually, I wish to contribute to the core PostgreSQL server repository as well.

Here are some of the ideas that I would love to implement after GSoC

- Notification from the system if the system resources are being over-utilized.
- Add more customization for the user to interact with the application.
- Add dark mode to the theme.