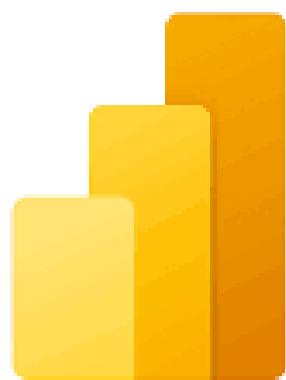


# **EXPERIMENT - 1**

**An Integrated Real-Time Analytics Model for GitHub  
Repository Insights via Power BI**



# **INTRODUCTION**

GitHub is one of the most widely used platforms for collaborative software development and version control, allowing developers to manage source code, track changes, and coordinate across teams. With the increasing volume of code contributions, issue tracking, and pull requests in modern software projects, there is a growing need to analyze this data effectively to improve productivity, team performance, and project planning.

Power BI, a powerful business analytics tool from Microsoft, enables the creation of interactive dashboards and real-time data visualizations. Integrating GitHub with Power BI offers a data-driven approach to monitoring repository activities, visualizing contributor performance, identifying workflow bottlenecks, and gaining valuable insights into development processes.

This case study demonstrates how to seamlessly integrate GitHub data with Power BI using a pre-built AppSource template, resulting in a dynamic dashboard that provides real-time analytics and decision-making support.

# **AIM**

To design and implement a comprehensive and interactive Power BI dashboard that integrates real-time GitHub repository data using a pre-built AppSource template. The objective is to automate the retrieval and visualization of key repository metrics such as commit history, pull requests, issue tracking, and contributor activity. This integration aims to provide stakeholders with actionable insights, enhance DevOps oversight, support agile project management, and facilitate data-driven decision-making through dynamic, user-friendly visual representations.

## KEY BENEFITS

- **Centralized Reporting:** Enables the consolidation of GitHub data into a unified platform for comprehensive and consistent reporting and analysis across all repositories.
- **Real-Time Monitoring:** Facilitates near real-time tracking of key repository activities including commits, pull requests, and issue resolution, allowing teams to stay up to date with the latest development changes.
- **Enhanced Team Performance Analysis:** Allows for detailed evaluations of individual and collective team contributions. This helps in assessing productivity levels, identifying high-performing contributors, and fostering collaborative practices.
- **Improved DevOps Oversight:** Supports efficient monitoring of software development pipelines, lifecycle of pull requests, merge frequencies, and code deployment metrics. This helps optimize continuous integration and delivery (CI/CD) workflows.
- **Effective Project Management:** Assists project managers in tracking sprint progress, managing backlog items, and analyzing issue closure rates. This leads to more informed planning and execution of development cycles.
- **Increased Transparency:** Promotes openness in software development by visualizing all key activities and metrics, allowing stakeholders to track progress and contributions without manual updates.
- **Time Efficiency:** Reduces the time spent on manual data compilation and analysis by automating the process, enabling teams to focus more on development and delivery.
- **Collaboration Enhancement:** Visual insights into team interactions and dependencies encourage better communication, responsibility sharing, and goal alignment among contributors.

# PROCEDURE:

## Using a GitHub Template in Power BI

### Step 1: Open Power BI Desktop

- Launch the Power BI Desktop application on your computer shown in fig 1.
- Make sure you are signed in with your Microsoft account to access AppSource services.
- Open Blank Report like shown in fig.2

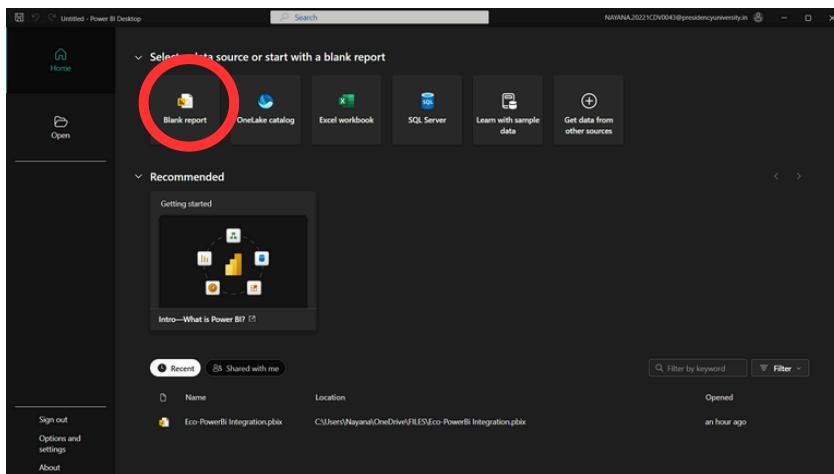


fig.1

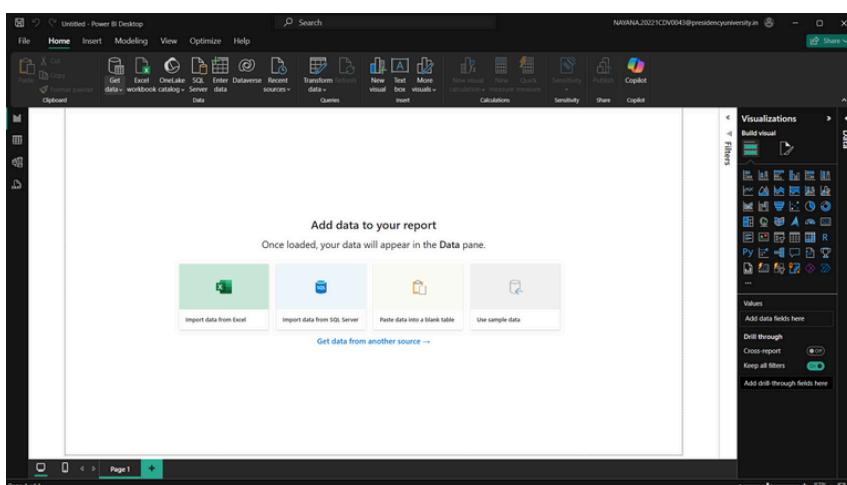


fig.2

## Step 2: Open the Data Connection Menu

- Click on the “File” menu in the top left corner of Power BI Desktop.
- Select “Get Data” from the dropdown menu(fig 3).
- In the Get Data window, go to the “Online Services” category(fig 4).
- Select GitHub(fig 5).

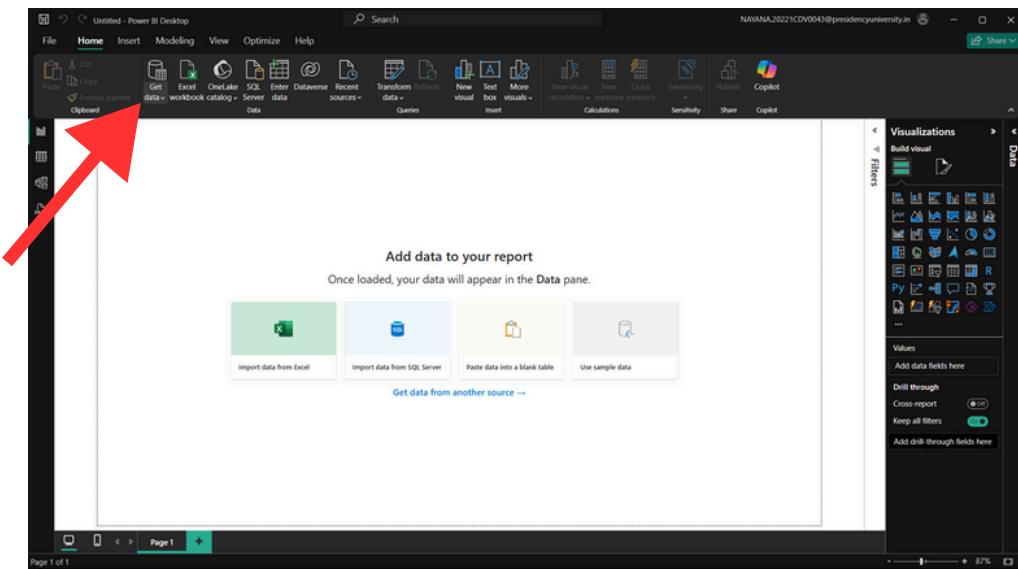


fig.3

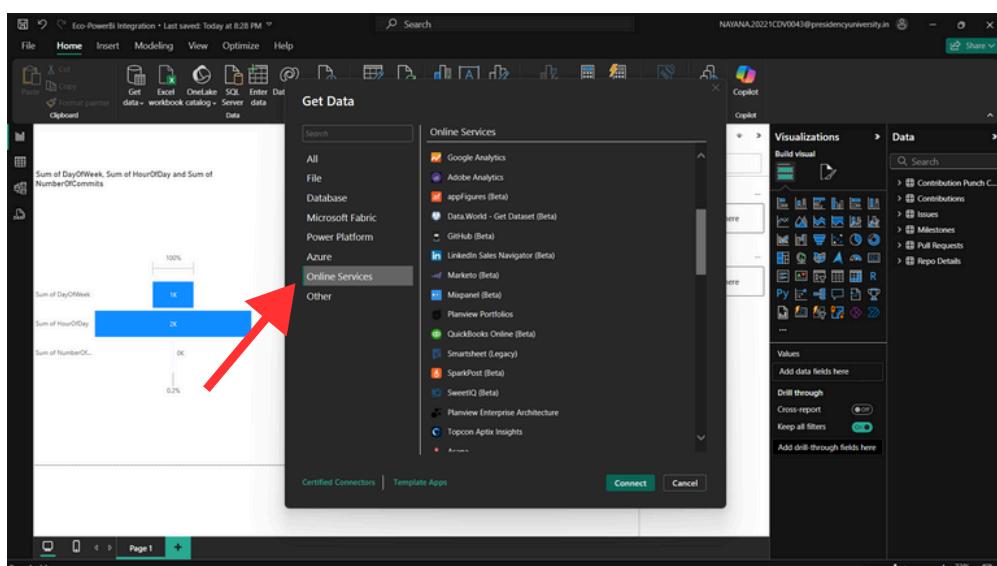


fig.4

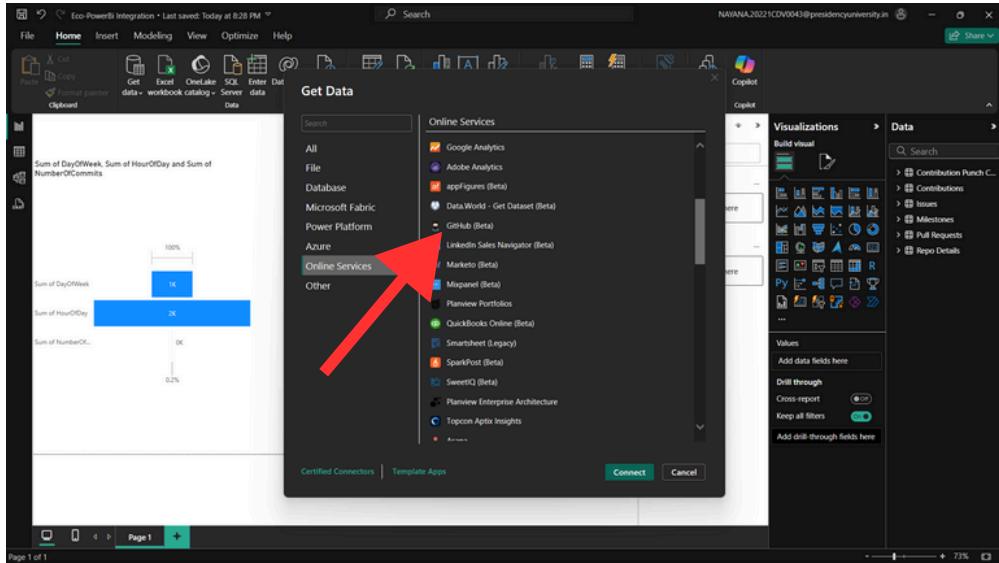


fig.5

If the GitHub AppSource template is already installed, Power BI will directly prompt you to sign in to your GitHub account. Otherwise, it will open the Microsoft AppSource window, where you can browse and install pre-built dashboards, templates, and analytics apps developed by Microsoft and third-party providers.

### Step 3: Search for “GitHub”

- In the AppSource search bar, type “GitHub” and press Enter (fig 6).
- You will see a list of available Power BI templates related to GitHub.
- Look for templates like:
  - GitHub Repository Insights (fig 7)
  - GitHub Analytics Dashboard
  - Open Source Contribution Dashboard
- Install and login to your GitHub account

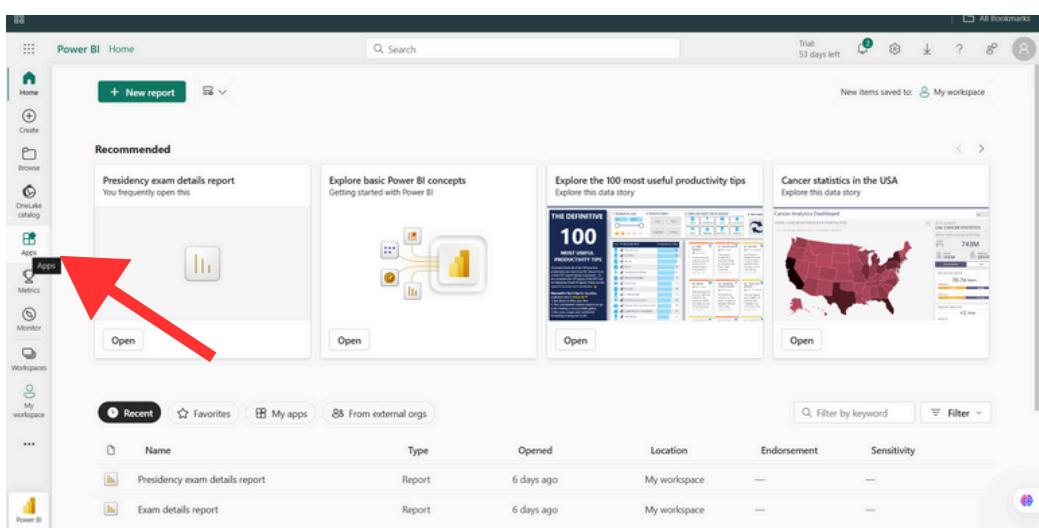


fig.6

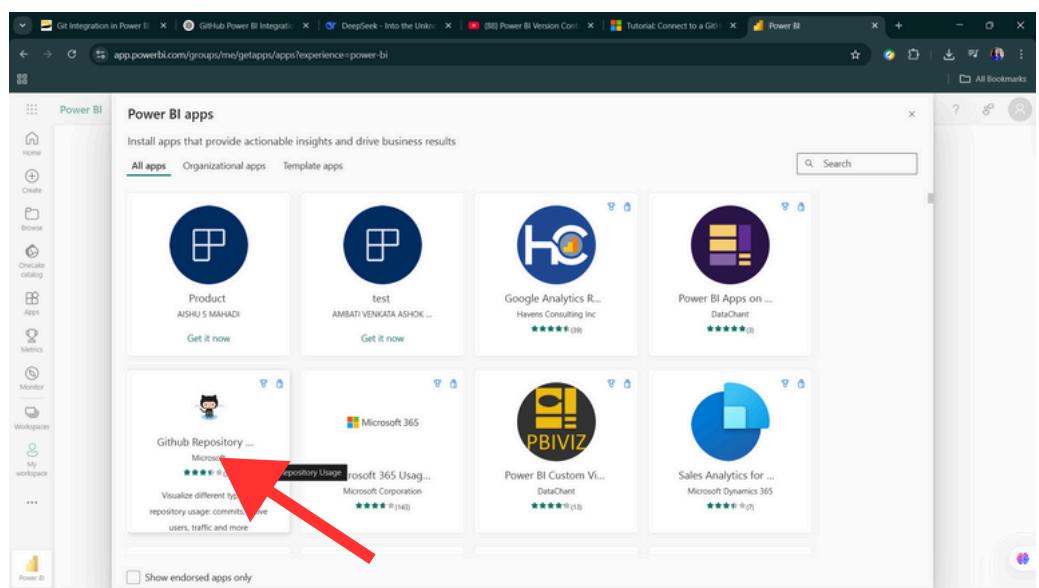


fig.7

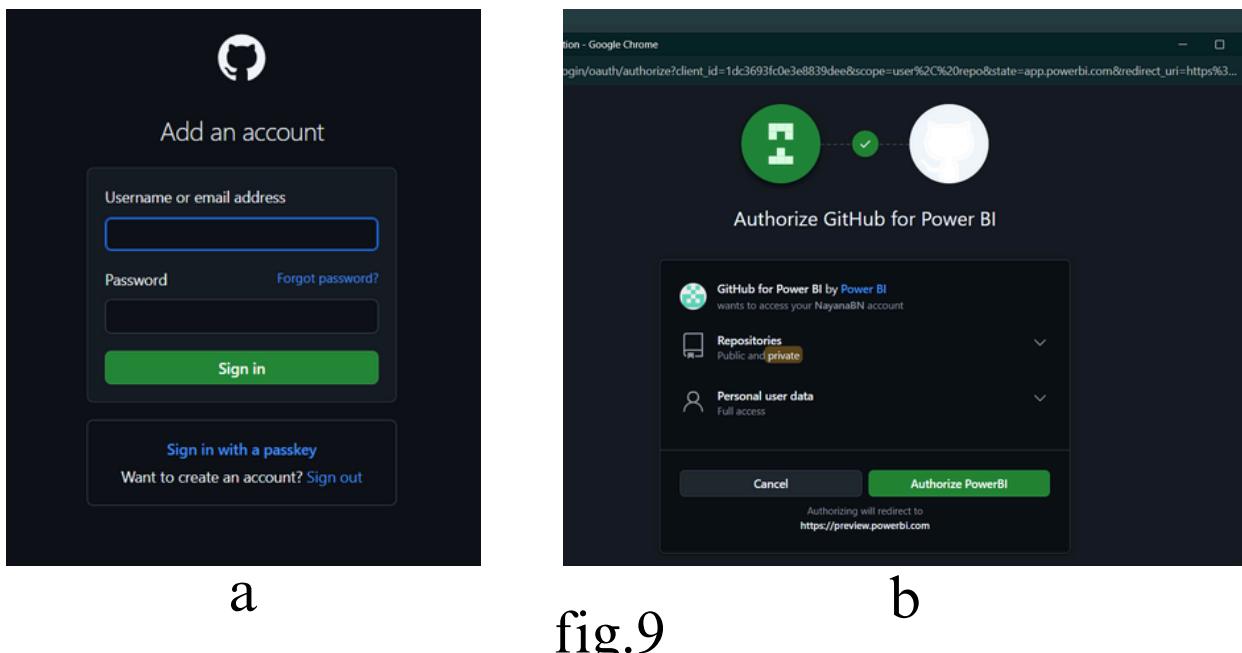


fig.8

## Step 5: Authenticate with GitHub (OAuth or PAT)

- When the template loads, it will prompt you to connect it to your GitHub account.(fig 9)
- You may be asked to:
  - Sign in using GitHub OAuth (i.e., logging in securely with your GitHub username/password)(fig 10)
  - OR enter a Personal Access Token (PAT) if required
- Follow the on-screen instructions to authorize the connection.

**This step allows Power BI to securely access your GitHub data in read-only mode.**



a b  
fig.9

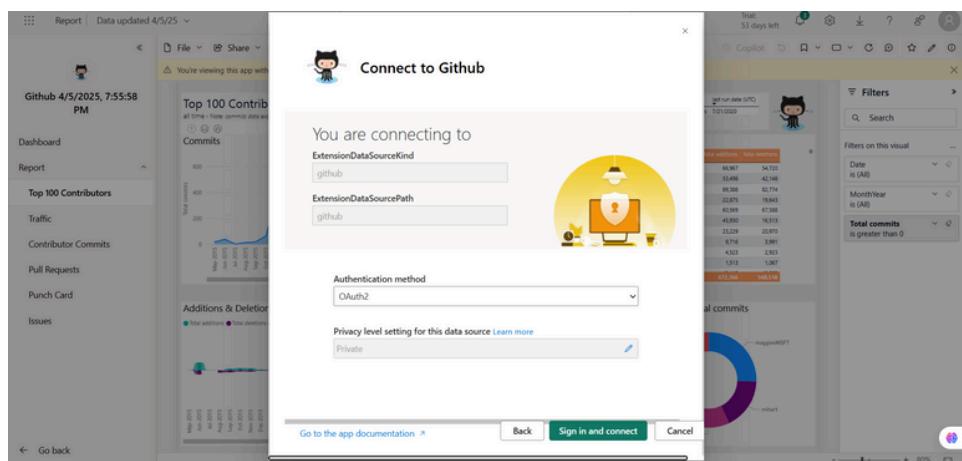


fig.10

**Close all the windows and restart Power Bi Desktop.**

**Repeat Step -2**

## Step 6: Configure GitHub Connection Settings

- After selecting GitHub as the data source, a dialog box will appear.
- Click Continue to proceed with the connection(fig 11).
- Enter the following details:(fig 12)

Owner Name – the GitHub username or organization that owns the repository.

Repository Name – the specific repository you want to visualize

Sign in to your GitHub account again if prompted, to authorize access.

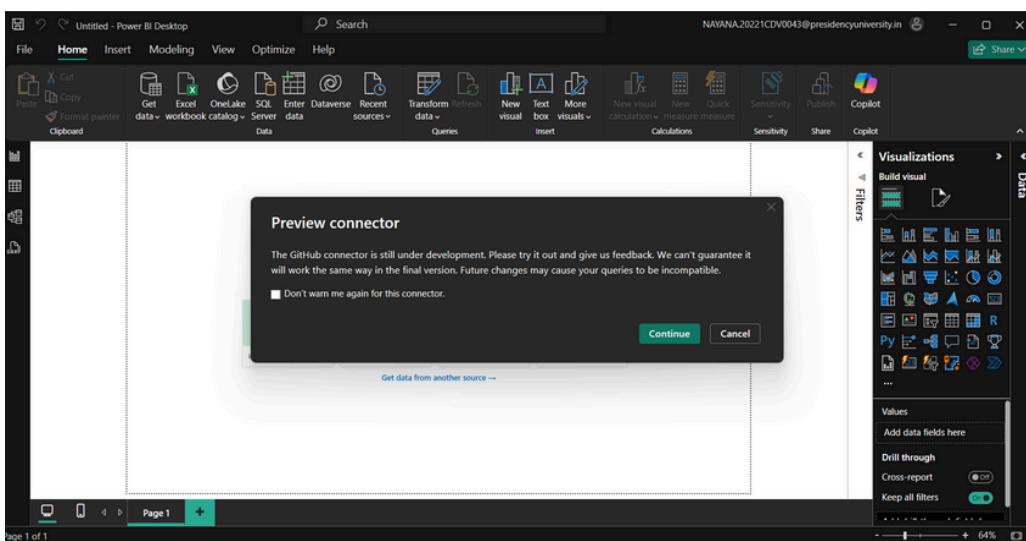


fig.11

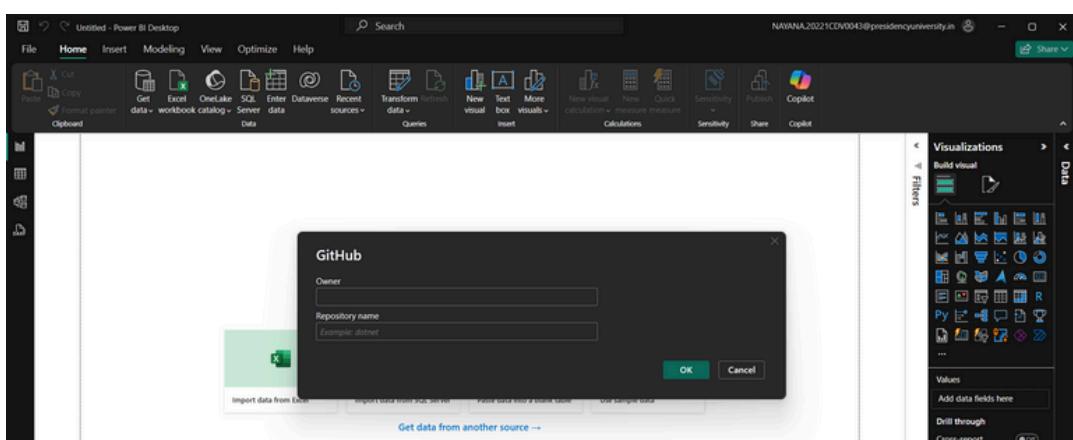


fig.12

## Step 7: Load and View the Visualizations

- Once connected, the template will automatically as shown in fig 13:
- Pull data from your GitHub repositories
- Populate pre-built dashboards and charts with your actual data
- click on load data after selecting the attributes

You can explore:

- Commits per contributor: See which team members are most active
- Open vs. closed issues: Analyze issue management trends
- Pull request trends: Monitor PR volume and activity over time
- Repository health: General overview of project development

Samples of how the dashboard can look in shown in fig 14

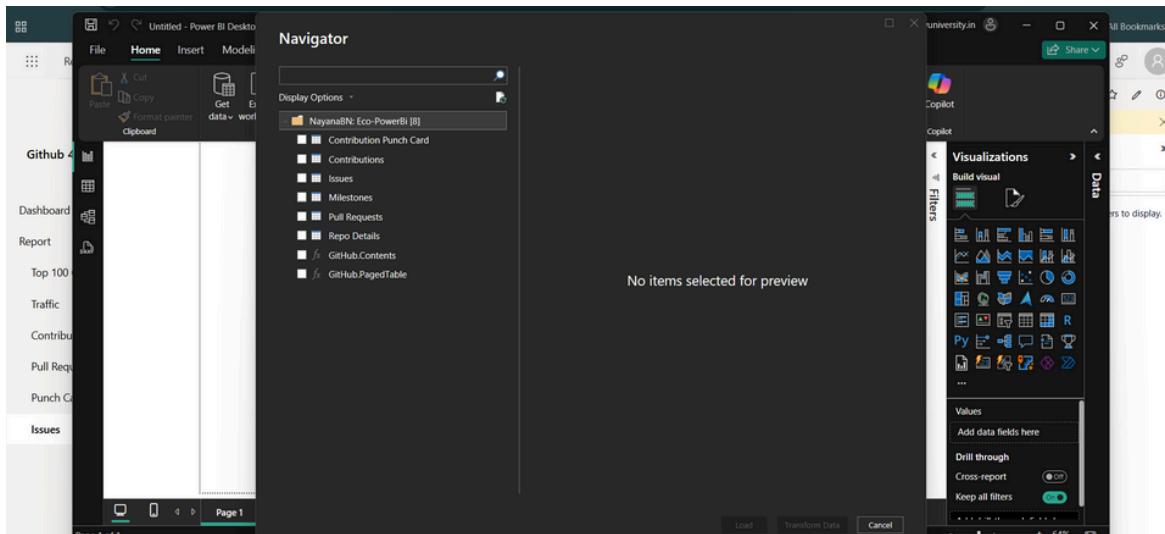


fig.13

# Visualizations

## FUNNEL CHART

STEP 1 Select the Funnel Chart from the Visualizations pane.

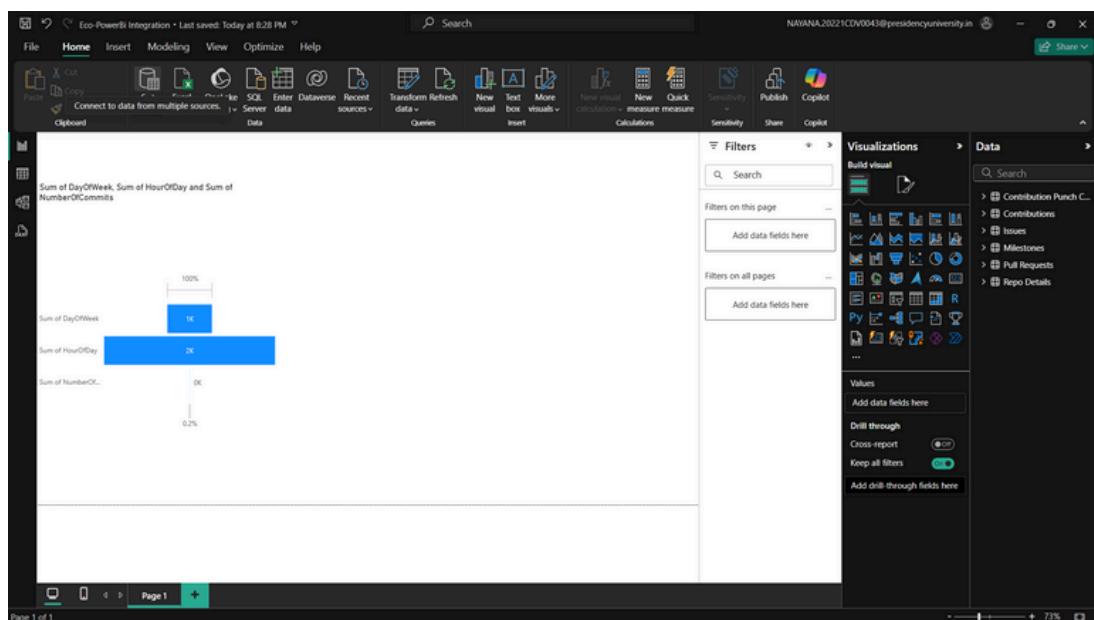
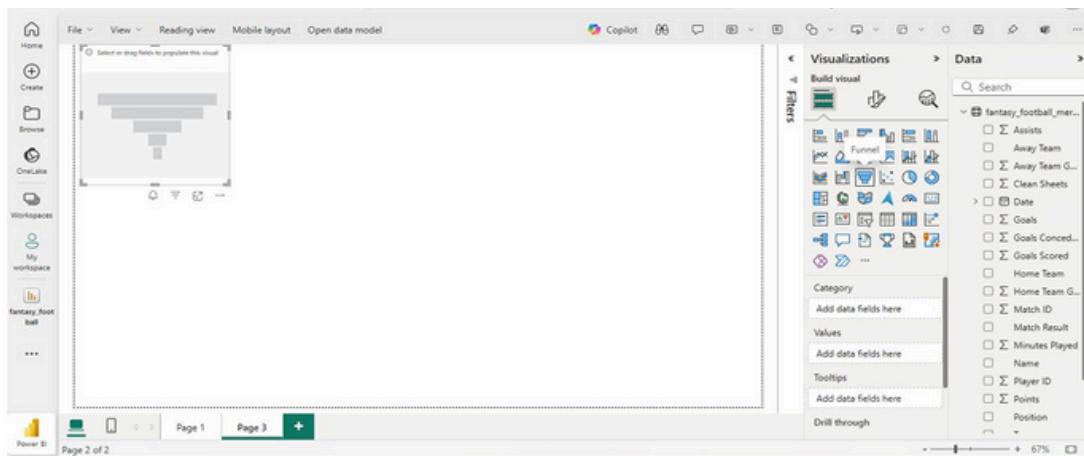


fig.14.a

# DONUT CHART

STEP 2: Select the **Donut Chart** from the **Visualizations** pane.

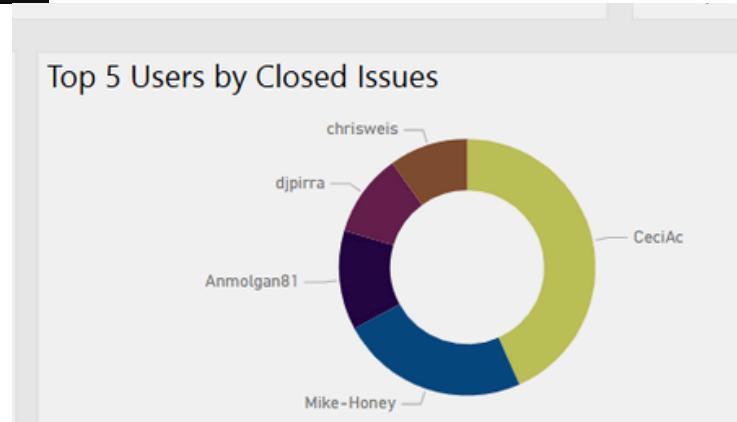
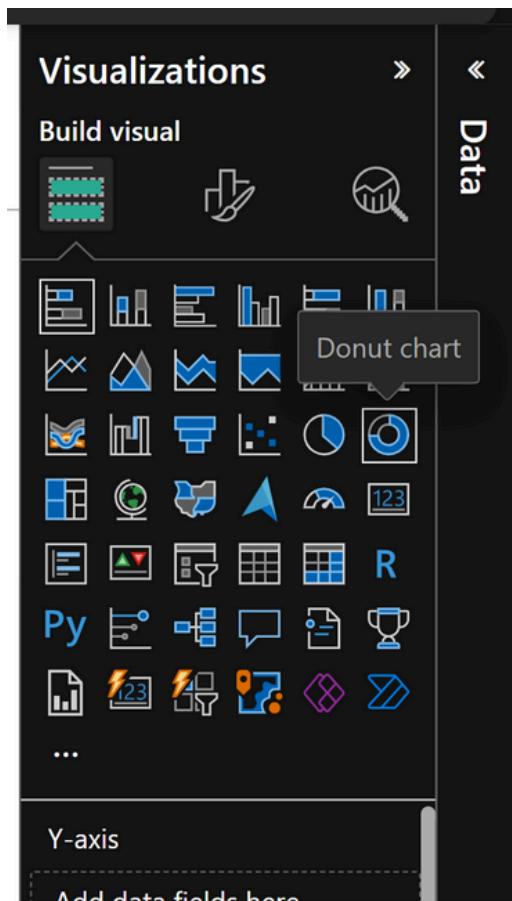


fig.14.b

# 100% STACKED COLUMN CHART

STEP 3: Select the **100% Stacked column chart** from the **Visualizations pane**.

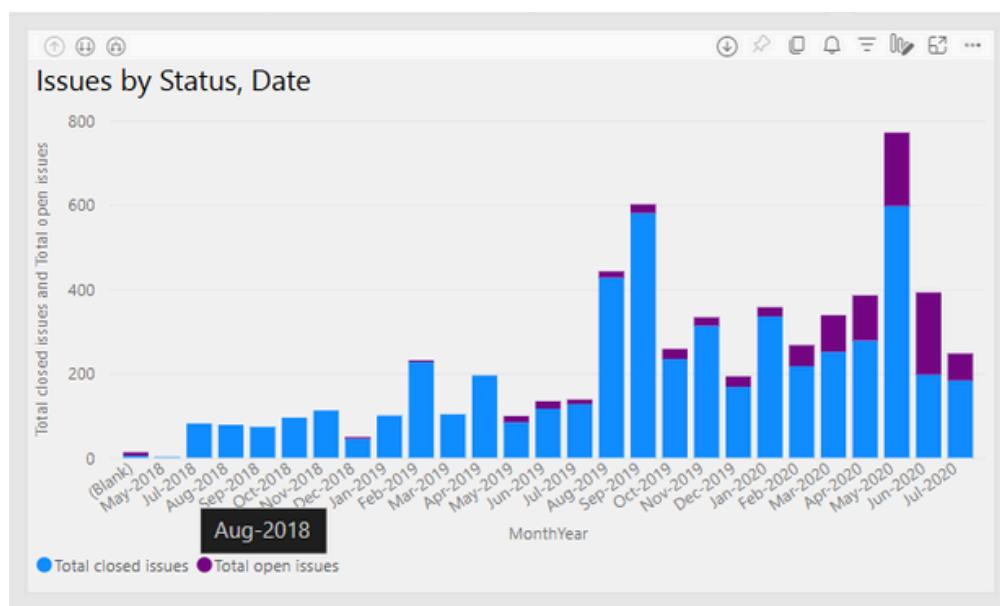
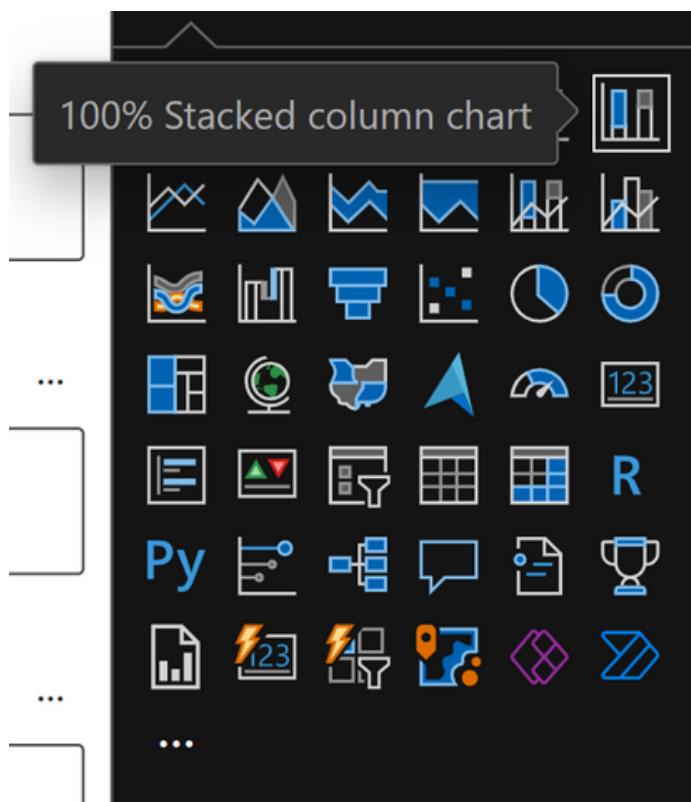


fig.14.c

# STACKED BAR CHART

STEP 4: Select the **Stacked bar chart** from the **Visualizations** pane.

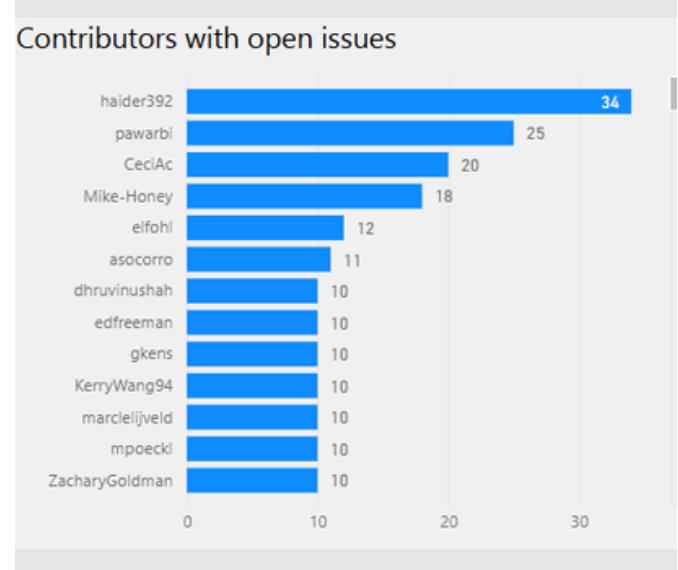
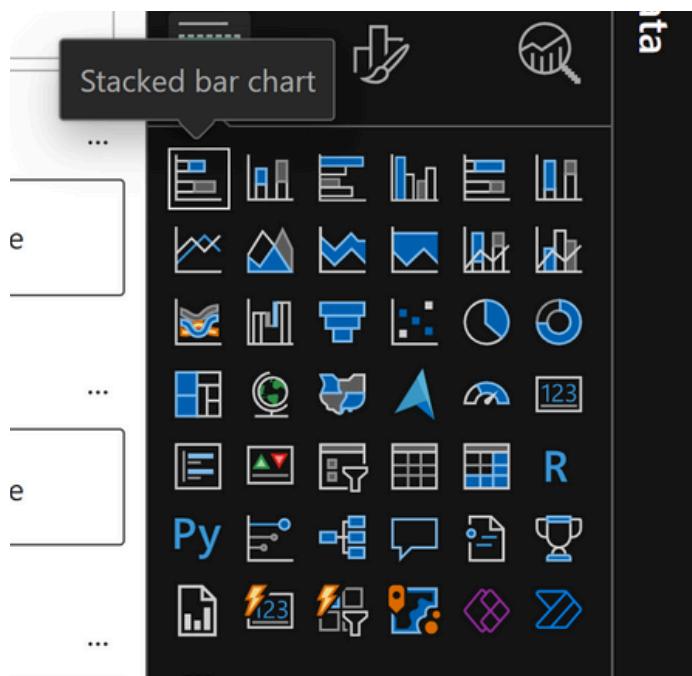


fig.14.d

# LINE CHART

STEP 4: Select the **Line chart** from the **Visualizations** pane.

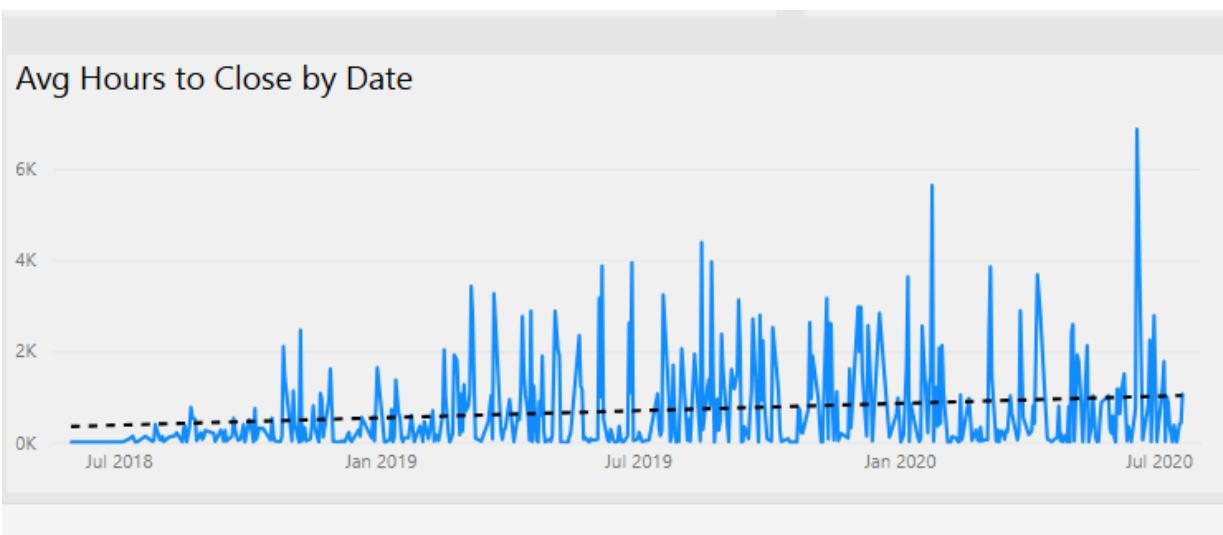
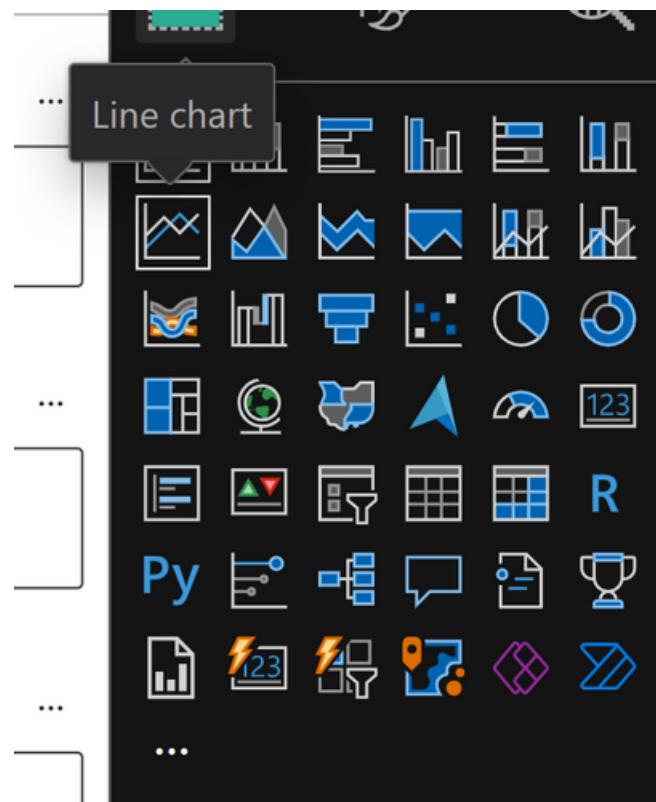


fig.14.e

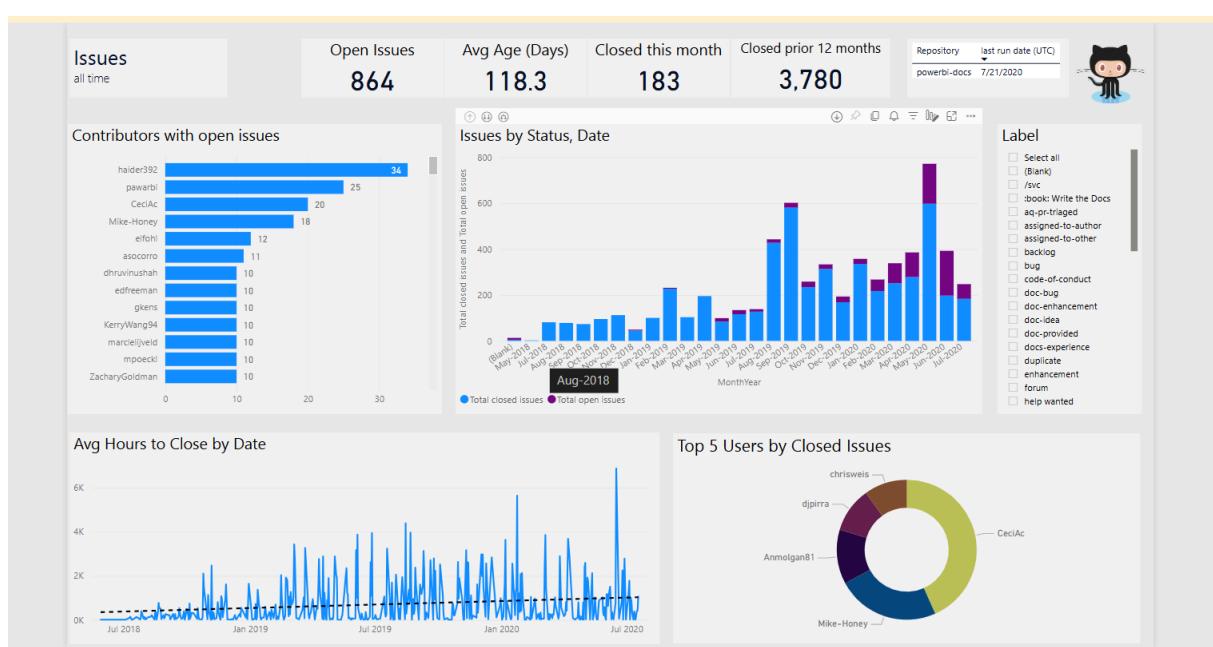
# CARD

STEP 5: Select the **card** from the **Visualizations** pane.

The screenshot shows the 'Visualizations' pane in Power BI, which contains a grid of icons representing different types of visualizations. A callout bubble labeled 'Card' points to the icon in the top-right corner of the grid. Below the grid, there are four summary cards:

Open Issues	Avg Age (Days)	Closed this month	Closed prior 12 months
864	118.3	183	3,780

fig.14.f



Complete Dashboard

## **FUNCTIONS USED:**

### **Power BI Functions & Features:**

- Data Import: Power Query for connecting to GitHub via Online Services
- Authentication: OAuth and PAT for secure access
- Visualizations: Pie charts, bar charts, line graphs, KPIs, tables, and slicers
- Filtering: Slicers and filters for dynamic interactivity
- DAX: Used to calculate contributor metrics, time to resolution, and PR activity
- Formatting: Custom backgrounds, fonts, and themes for improved presentation

## **ADVANTAGES:**

- Low-Code Setup: Requires minimal coding or scripting due to pre-configured templates
- Rapid Insights: Templates offer quick setup with meaningful visualizations
- Secure Authentication: OAuth and PAT methods ensure secure, read-only data access
- Scheduled Refresh: Ensures continuous and updated insights without manual data pulls
- Enhanced DevOps Monitoring: Tracks code health, team contributions, and agile metrics

## **ADVANTAGES OF USING THIS METHOD:**

Feature	Benefit
Low-Code Setup	Minimizes the need for writing custom API queries or handling raw JSON data manually.
Faster Time-to-Insights	Offers pre-configured data models and tables, allowing quick access to key metrics.
Enhanced Security	Utilizes secure, token-based authentication methods for safe data access.
Comprehensive Visualization	Leverages the full suite of Power BI features for in-depth data analysis and reporting.
Scheduled Data Refresh	Allows for automated refresh of data, ensuring reports remain current and relevant.

## **CONCLUSION:**

Integrating GitHub with Power BI through third-party connectors is an efficient and scalable solution for development analytics. It simplifies data extraction, reduces the need for manual scripting, and provides rich visual insights into your repositories. This approach is ideal for teams and organizations that want to monitor software engineering performance, track progress, and make informed, data-driven decisions.