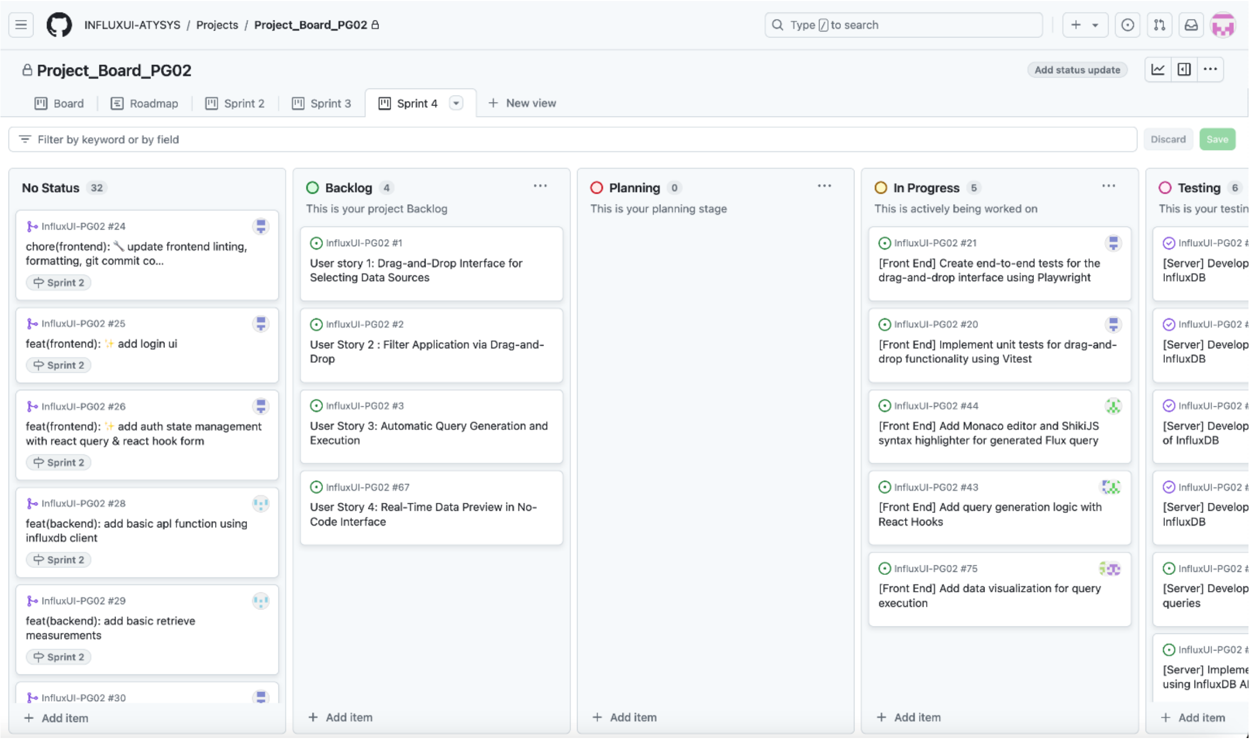
Snapshot Week 09 of Group PG02

Project of ATSYS

No-Code Solution for InfluxDB

LeStartUP
Zilin Song - a1833935
Jen-Hao Liu - a1893169
Dang Quy Duong - a1893592
Baojing Li - a1894836
Shih-Han Lin - a1900715
Feinan Guo - a1903270
Xiaoqing Zhao - a1904344
Hao Jiang - a1907177
Ziqi Zhang – a1909438

Product Backlog and Task Board



INFLUXUI-ATYSYS / Projects / Project_Board_PG02

Project_Board_PG02

No Status 32

- InfluxUI-PG02 #24 — chore(frontend): 🔧 update frontend linting, formatting, git commit co... — Sprint 2
- InfluxUI-PG02 #25 — feat(frontend): ✨ add login ui — Sprint 2
- InfluxUI-PG02 #26 — feat(frontend): ✨ add auth state management with react query & react hook form — Sprint 2
- InfluxUI-PG02 #28 — feat(backend): add basic apl function using influxdb client — Sprint 2
- InfluxUI-PG02 #29 — feat(backend): add basic retrieve measurements — Sprint 2
- InfluxUI-PG02 #30

Backlog 4

This is your project Backlog

- InfluxUI-PG02 #1 — User story 1: Drag-and-Drop Interface for Selecting Data Sources
- InfluxUI-PG02 #2 — User Story 2 : Filter Application via Drag-and-Drop
- InfluxUI-PG02 #3 — User Story 3: Automatic Query Generation and Execution
- InfluxUI-PG02 #67 — User Story 4: Real-Time Data Preview in No-Code Interface

Planning 0

This is your planning stage

In Progress 5

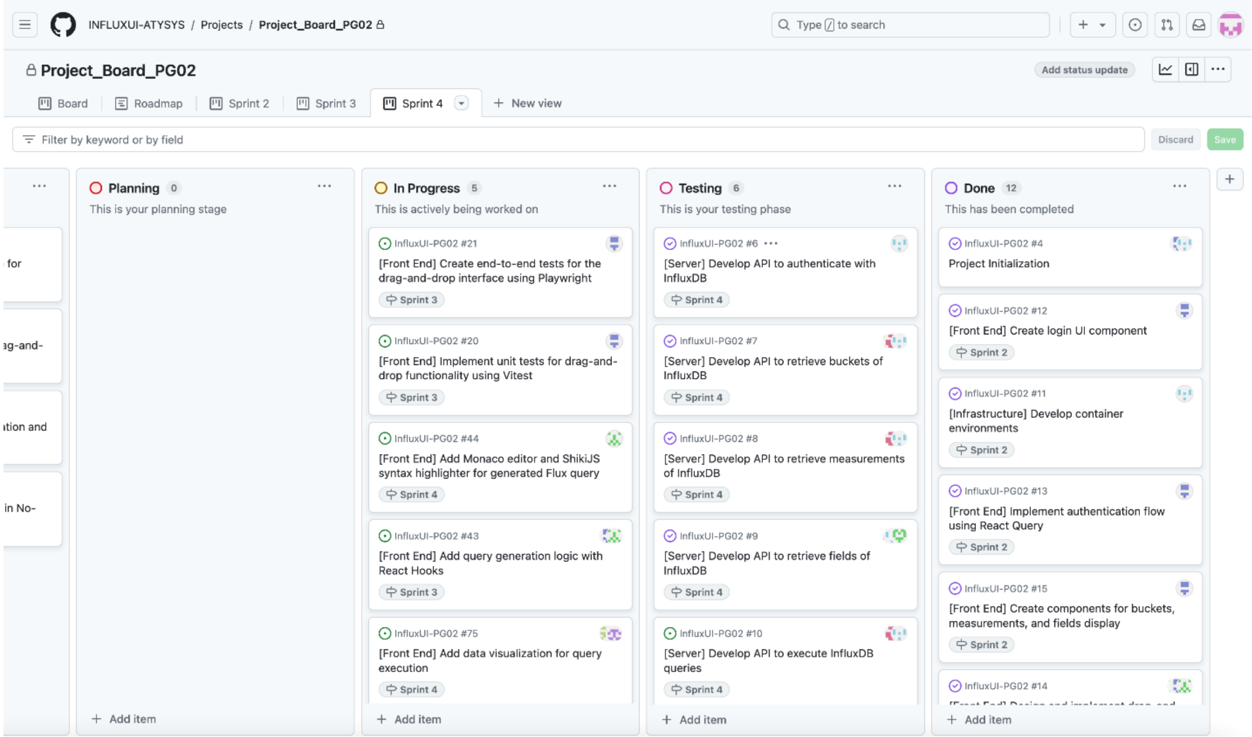
This is actively being worked on

- InfluxUI-PG02 #21 — [Front End] Create end-to-end tests for the drag-and-drop interface using Playwright
- InfluxUI-PG02 #20 — [Front End] Implement unit tests for drag-and-drop functionality using Vitest
- InfluxUI-PG02 #44 — [Front End] Add Monaco editor and ShikiJS syntax highlighter for generated Flux query
- InfluxUI-PG02 #43 — [Front End] Add query generation logic with React Hooks
- InfluxUI-PG02 #75 — [Front End] Add data visualization for query execution

Testing 6

This is your testin

- InfluxUI-PG02 # — [Server] Develop InfluxDB
- InfluxUI-PG02 # — [Server] Develop InfluxDB
- InfluxUI-PG02 # — [Server] Develop of InfluxDB
- InfluxUI-PG02 # — [Server] Develop InfluxDB
- InfluxUI-PG02 # — [Server] Develop queries
- InfluxUI-PG02 # — [Server] Impleme using InfluxDB AI



INFLUXUI-ATYSYS / Projects / Project_Board_PG02

Project_Board_PG02

Planning 0

This is your planning stage

In Progress 5

This is actively being worked on

- InfluxUI-PG02 #21 — [Front End] Create end-to-end tests for the drag-and-drop interface using Playwright — Sprint 3
- InfluxUI-PG02 #20 — [Front End] Implement unit tests for drag-and-drop functionality using Vitest — Sprint 3
- InfluxUI-PG02 #44 — [Front End] Add Monaco editor and ShikiJS syntax highlighter for generated Flux query — Sprint 4
- InfluxUI-PG02 #43 — [Front End] Add query generation logic with React Hooks — Sprint 3
- InfluxUI-PG02 #75 — [Front End] Add data visualization for query execution — Sprint 4

Testing 6

This is your testing phase

- InfluxUI-PG02 #6 — [Server] Develop API to authenticate with InfluxDB — Sprint 4
- InfluxUI-PG02 #7 — [Server] Develop API to retrieve buckets of InfluxDB — Sprint 4
- InfluxUI-PG02 #8 — [Server] Develop API to retrieve measurements of InfluxDB — Sprint 4
- InfluxUI-PG02 #9 — [Server] Develop API to retrieve fields of InfluxDB — Sprint 4
- InfluxUI-PG02 #10 — [Server] Develop API to execute InfluxDB queries — Sprint 4

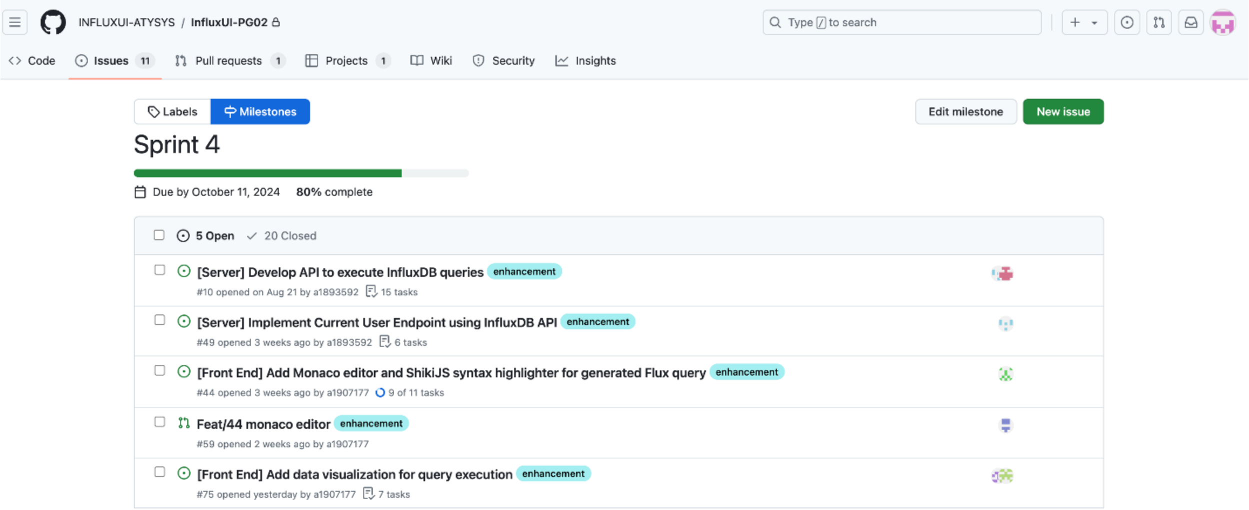
Done 12

This has been completed

- InfluxUI-PG02 #4 — Project Initialization
- InfluxUI-PG02 #12 — [Front End] Create login UI component — Sprint 2
- InfluxUI-PG02 #11 — [Infrastructure] Develop container environments — Sprint 2
- InfluxUI-PG02 #13 — [Front End] Implement authentication flow using React Query — Sprint 2
- InfluxUI-PG02 #15 — [Front End] Create components for buckets, measurements, and fields display — Sprint 2
- InfluxUI-PG02 #14 — [Front End] Design and implement drag-and

Sprint Backlog and User Stories

- Milestones of the end of sprint 4 and beginning of sprint 5:



Sprint 4

Due by October 11, 2024 **80% complete**

5 Open ✓ 20 Closed

□ [Server] Develop API to execute InfluxDB queries `enhancement`
#10 opened on Aug 21 by a1893592 15 tasks

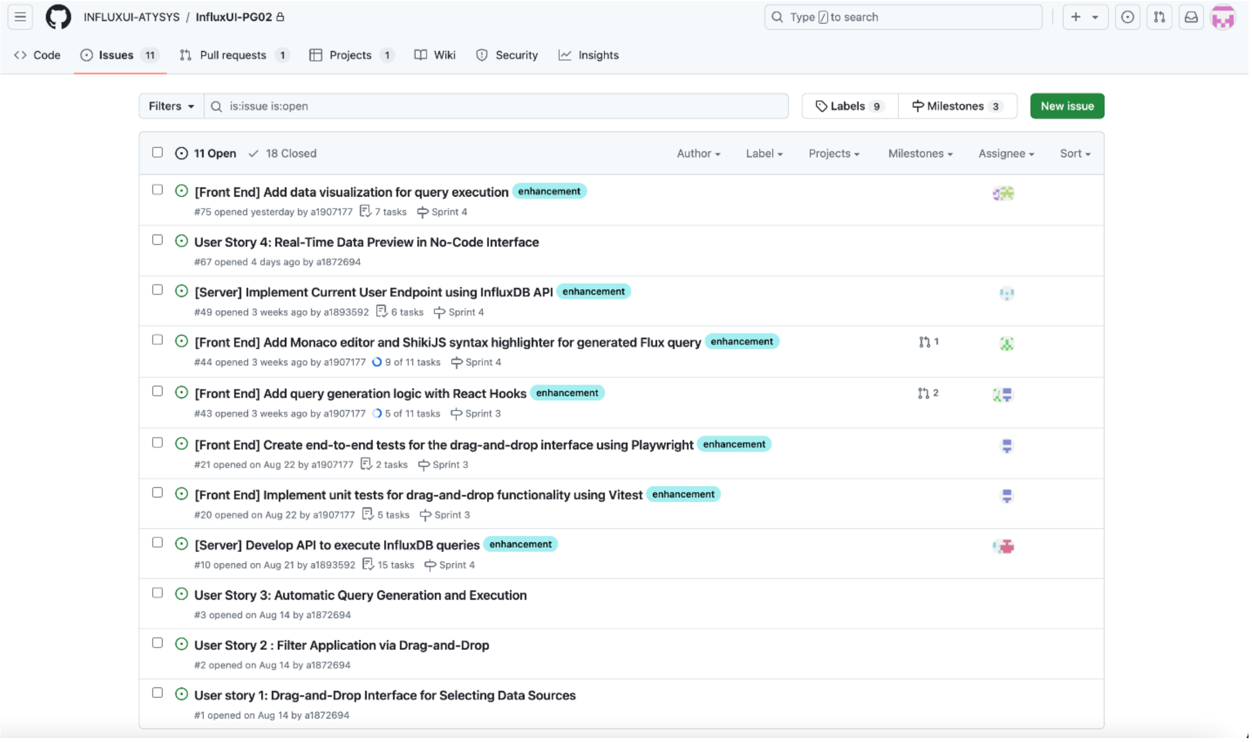
□ [Server] Implement Current User Endpoint using InfluxDB API `enhancement`
#49 opened 3 weeks ago by a1893592 6 tasks

□ [Front End] Add Monaco editor and ShikiJS syntax highlighter for generated Flux query `enhancement`
#44 opened 3 weeks ago by a1907177 9 of 11 tasks

□ Feat/44 monaco editor `enhancement`
#59 opened 2 weeks ago by a1907177

□ [Front End] Add data visualization for query execution `enhancement`
#75 opened yesterday by a1907177 7 tasks

- Sprint issues we are working on:



11 Open ✓ 18 Closed

□ [Front End] Add data visualization for query execution `enhancement`
#75 opened yesterday by a1907177 7 tasks Sprint 4

□ User Story 4: Real-Time Data Preview in No-Code Interface
#67 opened 4 days ago by a1872694

□ [Server] Implement Current User Endpoint using InfluxDB API `enhancement`
#49 opened 3 weeks ago by a1893592 6 tasks Sprint 4

□ [Front End] Add Monaco editor and ShikiJS syntax highlighter for generated Flux query `enhancement` 1
#44 opened 3 weeks ago by a1907177 9 of 11 tasks Sprint 4

□ [Front End] Add query generation logic with React Hooks `enhancement` 2
#43 opened 3 weeks ago by a1907177 5 of 11 tasks Sprint 3

□ [Front End] Create end-to-end tests for the drag-and-drop interface using Playwright `enhancement`
#21 opened on Aug 22 by a1907177 2 tasks Sprint 3

□ [Front End] Implement unit tests for drag-and-drop functionality using Vitest `enhancement`
#20 opened on Aug 22 by a1907177 5 tasks Sprint 3

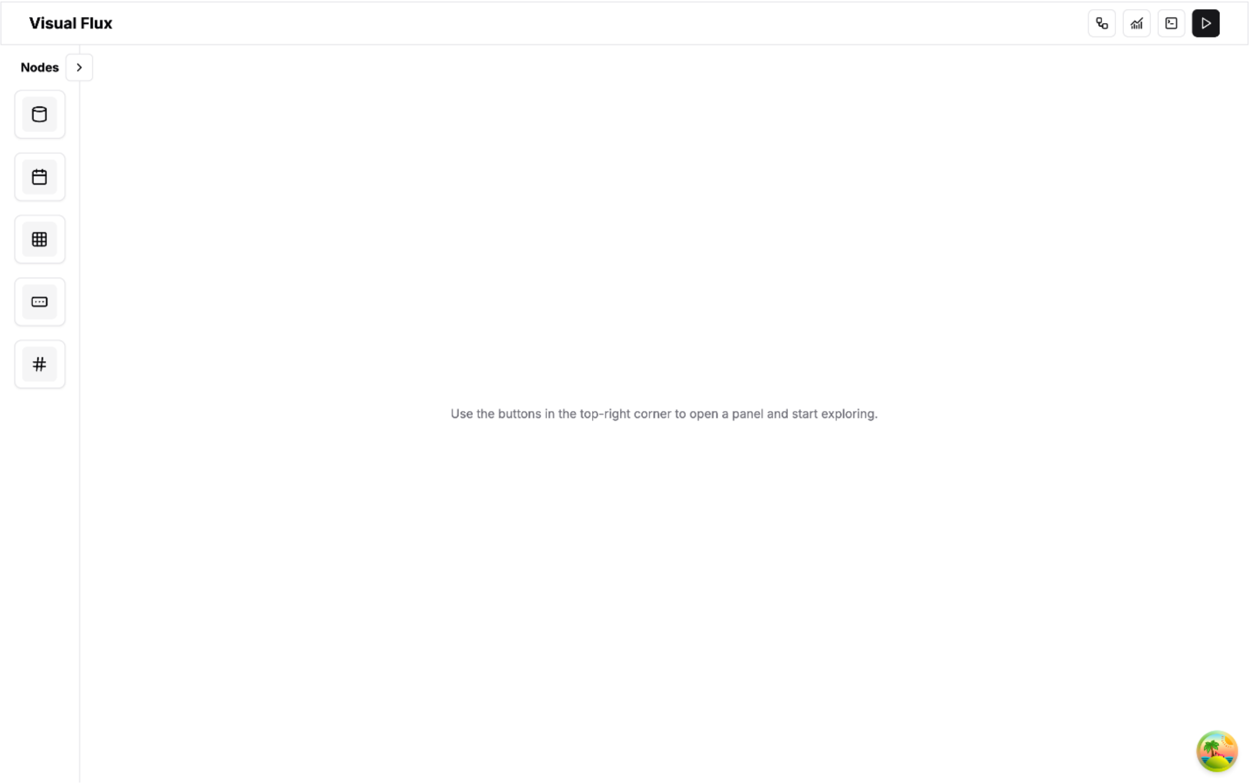
□ [Server] Develop API to execute InfluxDB queries `enhancement`
#10 opened on Aug 21 by a1893592 15 tasks Sprint 4

□ User Story 3: Automatic Query Generation and Execution
#3 opened on Aug 14 by a1872694

□ User Story 2 : Filter Application via Drag-and-Drop
#2 opened on Aug 14 by a1872694

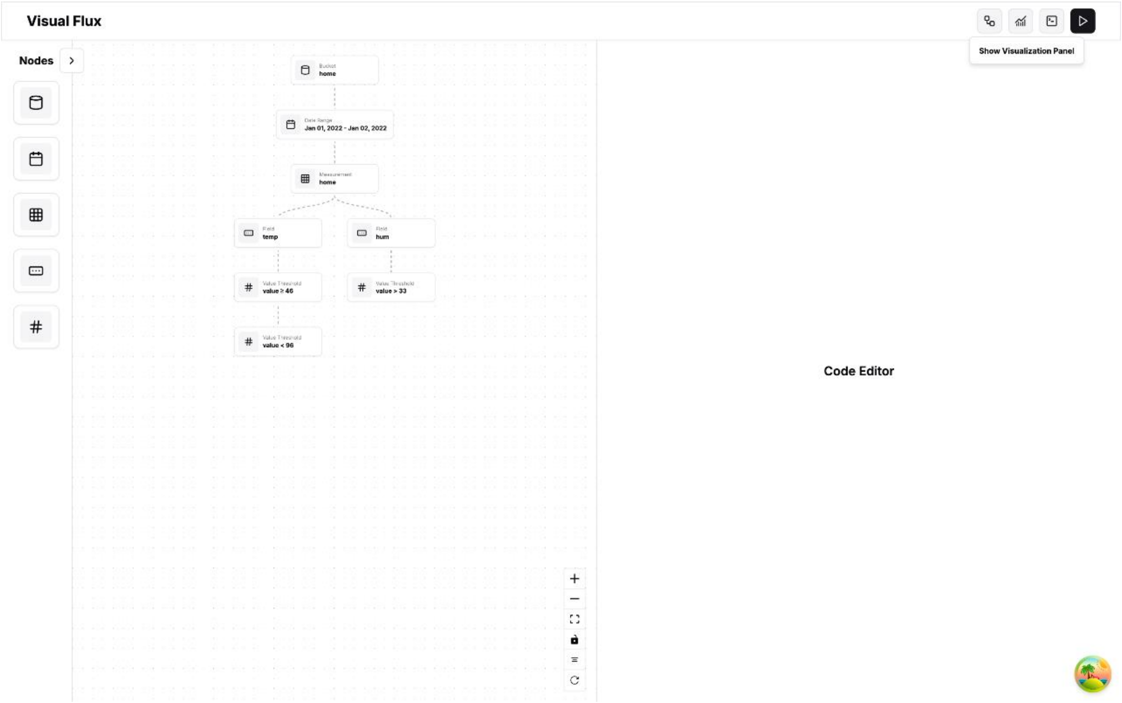
□ User story 1: Drag-and-Drop Interface for Selecting Data Sources
#1 opened on Aug 14 by a1872694

- Frontend: current side bar nodes

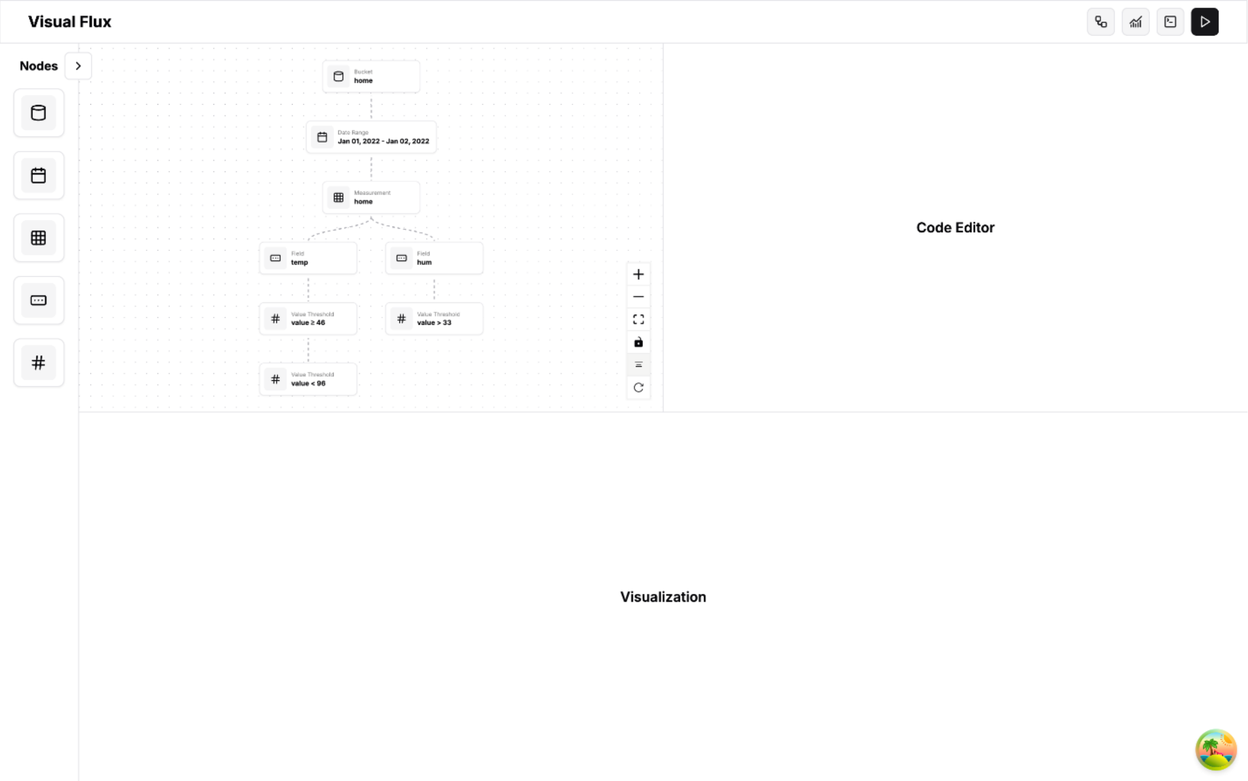


Use the buttons in the top-right corner to open a panel and start exploring.

- Frontend: auto align and reset for flow state node



- Frontend: visualisation feature



Bucket
home

Date Range
Jan 01, 2022 - Jan 02, 2022

Measurement
home

Field
temp

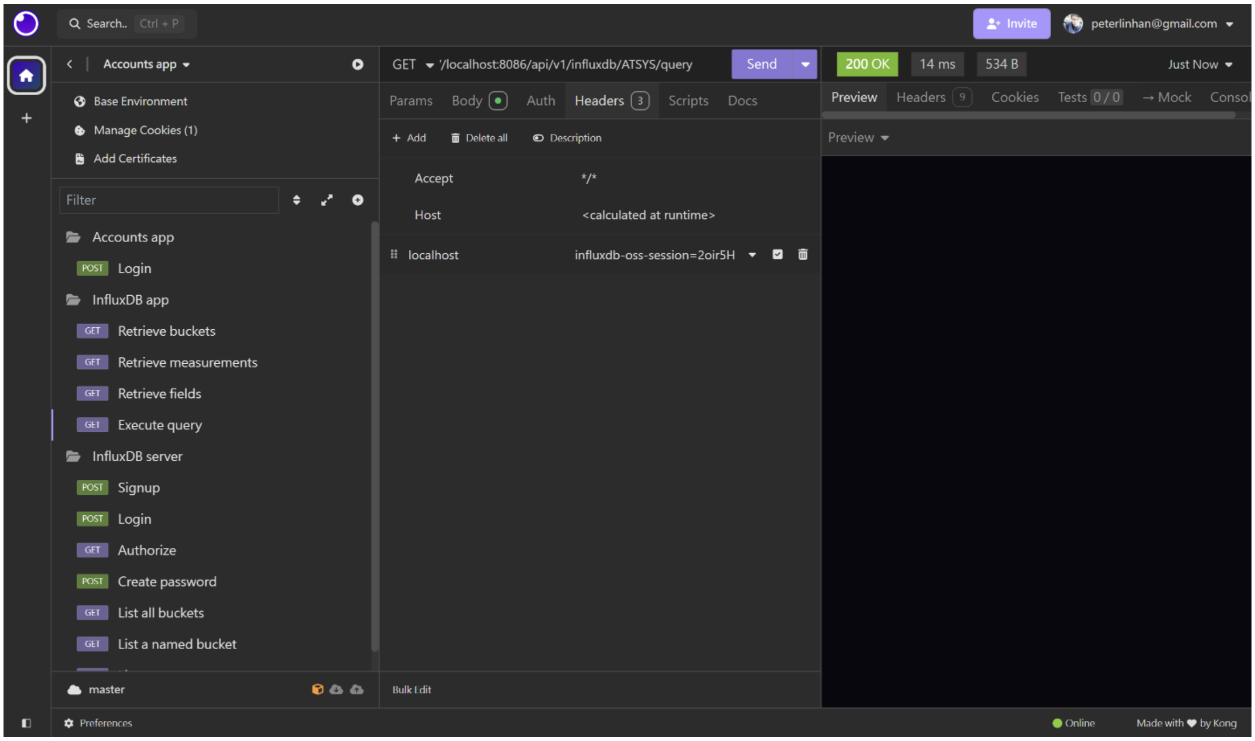
Field
hum

Value Threshold
value ≥ 46

Value Threshold
value > 33

Value Threshold
value < 96

- Backend: execute query



GET '/localhost:8086/api/v1/influxdb/ATSYS/query

200 OK 14 ms 534 B

Accept	*/*
Host	<calculated at runtime>
localhost	influxdb-oss-session=2oir5H

On the frontend, our team successfully added a toggle reset and execute button to enhance user control within the code editor with improved the workspace layout, making it easier to manage multiple tasks/views. Meanwhile, we also ensured that workflow progress is saved even after page refreshes and fixed several bugs that were unsolved in previous sprint.

As for the backend, our team developed and tested four key APIs for retrieving data from InfluxDB. By adopting a more RESTful design, we continued to improve flexibility by using query parameters for key values. We also implemented ISO-8601 format for time series data handling.

Definition of Done

- Frontend:
    - features for the current sprint are fully implemented, tested, and confirmed to function without critical bugs.
    - Usability enhancements (e.g., flow state persistence) are ensured.
- Backend:
    - The API is marked complete after it has been thoroughly tested, refactored to a RESTful structure, and confirmed to handle edge cases consistently.

Summary of Changes

- Improve user experience with a more intuitive and flexible UI, including multi-panel design, enhanced control nodes
- Backend APIs are now more scalable and robust, with improvements in flexibility, query parameter usage, and timestamp consistency
- Bug fixed and usability refinements enhance the product’s overall reliability and user experience