

Snapshot Week 08 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

While our frontend team focuses on resolving the presentation issues for measurements and fields on the interface, backend team aims to use cookies for authorization and authentication in API testing.

The screenshot shows a Jira Project Board for 'Project_Board_PG02' in a dark theme. The board is organized into columns representing different stages of the workflow: No Status, Backlog, Planning, In Progress, Testing, and Done. Each column contains a list of tasks, each represented by a card with a title, description, and progress indicator. The tasks are assigned to team members, indicated by avatars. The board also includes a search bar, a filter by keyword or by field, and a 'Discard' button. The 'No Status' column contains 12 tasks, including 'InfluxUI-PG02 #24' and 'InfluxUI-PG02 #25'. The 'Backlog' column contains 3 tasks, including 'InfluxUI-PG02 #1' and 'InfluxUI-PG02 #2'. The 'Planning' column contains 3 tasks, including 'InfluxUI-PG02 #43' and 'InfluxUI-PG02 #44'. The 'In Progress' column contains 6 tasks, including 'InfluxUI-PG02 #49' and 'InfluxUI-PG02 #19'. The 'Testing' column contains 6 tasks, including 'InfluxUI-PG02 #6' and 'InfluxUI-PG02 #7'. The 'Done' column contains 10 tasks, including 'InfluxUI-PG02 #4' and 'InfluxUI-PG02 #12'.

Project_Board_PG02

Board | Roadmap | Sprint 2 | **Sprint 3** | Sprint 4 | + New view

Filter by keyword or by field

No Status 12

- InfluxUI-PG02 #24 chore(frontend): ^, update frontend linting, formatting, git commit co...
- InfluxUI-PG02 #25 feat(frontend): ^ add login ui
- InfluxUI-PG02 #26 feat(frontend): ^ add auth state management with react query & react hook form
- InfluxUI-PG02 #28 feat(backend): add basic api function using influxdb client
- InfluxUI-PG02 #29 feat(backend): add basic retrieve measurements
- InfluxUI-PG02 #30 Fix/13 fix login state persistence
- InfluxUI-PG02 #27 Feature/6 api authentication
- InfluxUI-PG02 #42 docs(docs): ^ update server readme for api testing

+ Add Item

Backlog 3

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

+ Add Item

Planning 3

This is your planning stage

- InfluxUI-PG02 #43 [Front End] Add query generation logic using Zustand
- InfluxUI-PG02 #44 [Front End] Add Monaco editor and ShikJS syntax highlighter for generated Flux query
- InfluxUI-PG02 #50 [Front End] Add resizable for multi-pane editor layout

+ Add Item

In Progress 6

This is actively being worked on

- InfluxUI-PG02 #49 [Server] Implement Current User Endpoint using InfluxDB API
- InfluxUI-PG02 #19 [Front End] Add error handling and user feedback for drag-and-drop actions
- InfluxUI-PG02 #18 [Front End] Create state management for selected data sources
- 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

+ Add Item

Testing 6

This is your testing phase

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

+ Add Item

Done 10

This has been completed

- InfluxUI-PG02 #4 Project Initialization
- InfluxUI-PG02 #12 [Front End] Create login UI component
- InfluxUI-PG02 #11 [Infrastructure] Develop container environments
- InfluxUI-PG02 #13 [Front End] implement authentication flow using React Query
- InfluxUI-PG02 #15 [Front End] Create components for buckets, measurements, and fields display
- InfluxUI-PG02 #14 [Front End] Design and implement drag-and-drop interface using @xyflow/react
- InfluxUI-PG02 #17 [Front End] implement drag-and-drop between selection list and query builder using @dnd-kit/core
- InfluxUI-PG02 #40

+ Add Item

Sprint Backlog and User Stories

- Milestones of the end of sprint 3 and beginning of sprint 4:

The screenshot shows the 'Milestones' tab in the InfluxDB project repository. It displays three milestones: Sprint 2, Sprint 3, and Sprint 4. Each milestone includes a progress bar, completion percentage, and counts of open and closed issues.

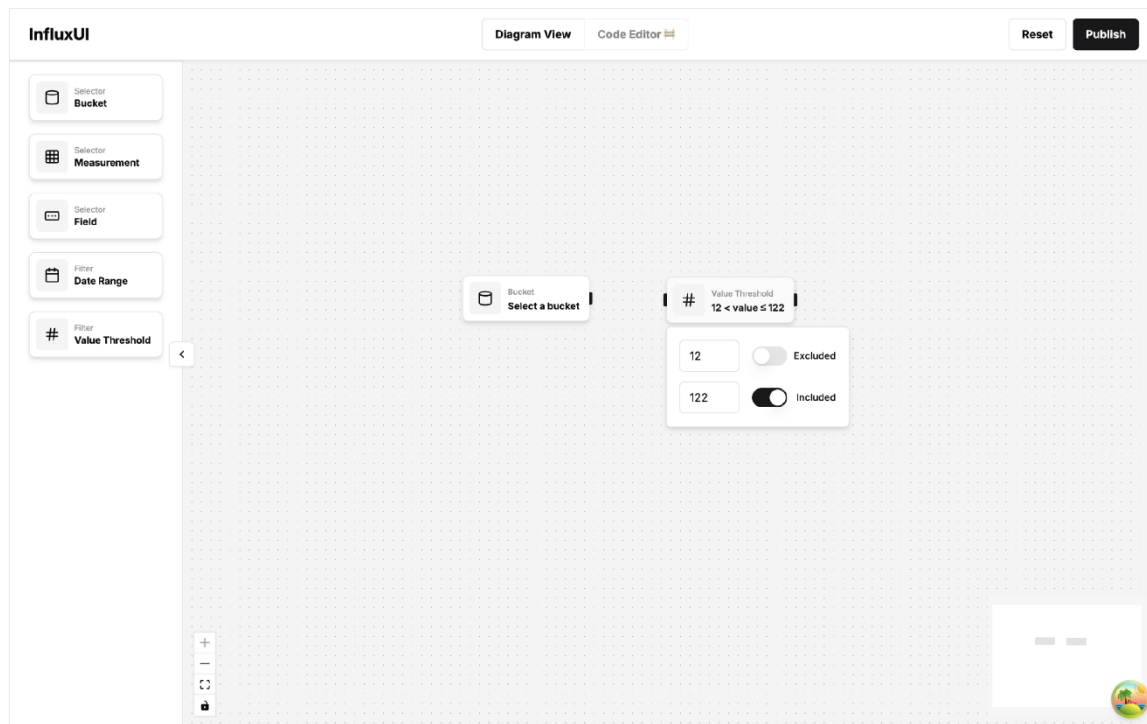
Sprint	Due Date	Last Updated	Completion	Open	Closed
Sprint 4	Due by October 11, 2024	Last updated about 1 hour ago	50% complete	4 open	4 closed
Sprint 3	Past due by 1 day	Last updated about 1 hour ago	58% complete	5 open	7 closed
Sprint 2	Past due by 12 days	Last updated 9 days ago	100% complete	0 open	14 closed

- Sprint issues we are working on:

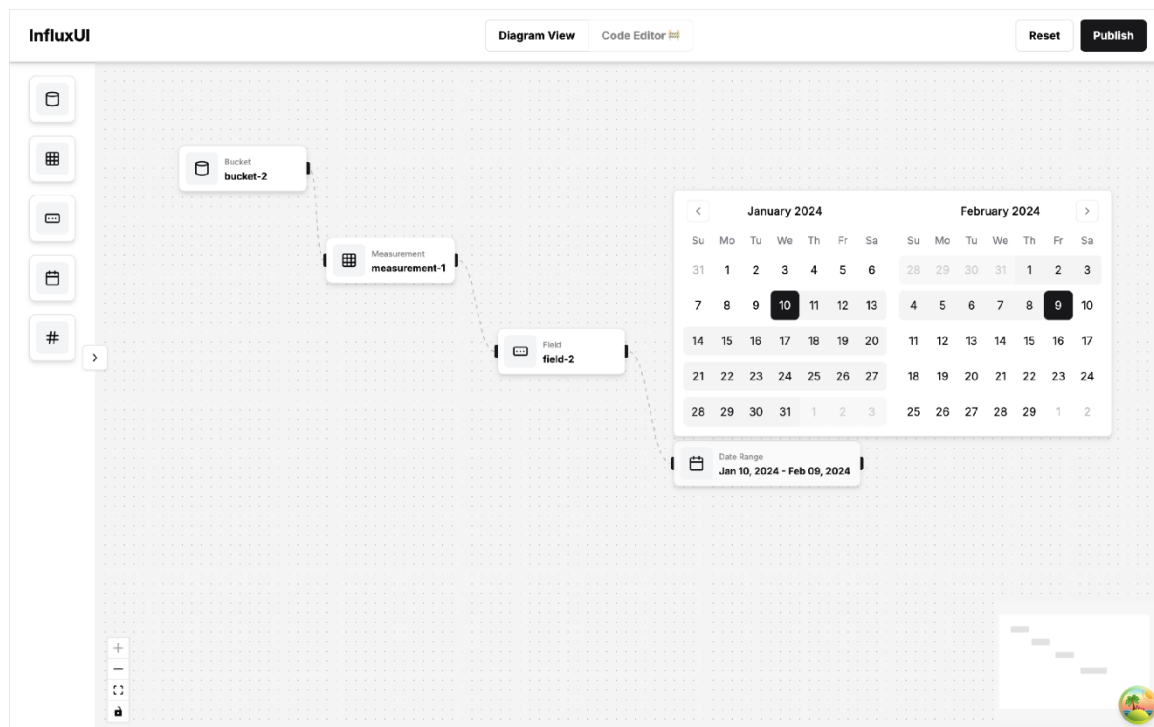
The screenshot shows the 'Issues' tab in the InfluxDB project repository. It displays a list of 13 open issues, each with a title, description, and status. The issues are categorized by sprint and include labels like 'enhancement'.

Issue ID	Title	Description	Status	Label
#49	[Server] Implement Current User Endpoint using InfluxDB API	opened yesterday by a1893592	6 tasks	enhancement
#44	[Front End] Add Monaco editor and ShikiJS syntax highlighter for generated Flux query	opened 3 days ago by a1907177	3 of 11 tasks	enhancement
#43	[Front End] Add query generation logic using Zustand	opened 3 days ago by a1907177	12 tasks	enhancement
#21	[Front End] Create end-to-end tests for the drag-and-drop interface using Playwright	opened 3 weeks ago by a1907177	2 tasks	enhancement
#20	[Front End] Implement unit tests for drag-and-drop functionality using Vitest	opened 3 weeks ago by a1907177	5 tasks	enhancement
#19	[Front End] Add error handling and user feedback for drag-and-drop actions	opened 3 weeks ago by a1907177	2 of 4 tasks	enhancement
#18	[Front End] Create state management for selected data sources	opened 3 weeks ago by a1907177	6 tasks	enhancement
#10	[Server] Develop API to execute InfluxDB queries	opened 3 weeks ago by a1893592	4 tasks	enhancement
#9	[Server] Develop API to retrieve fields of InfluxDB	opened 3 weeks ago by a1893592	4 tasks	enhancement
#5	[Server] Develop API to authorize with InfluxDB	opened 3 weeks ago by a1893592	6 tasks	enhancement
#3	User Story 3: Automatic Query Generation and Execution	opened on Aug 14 by a1672694		
	User Story 2 : Filter Application via Drag-and-Drop			

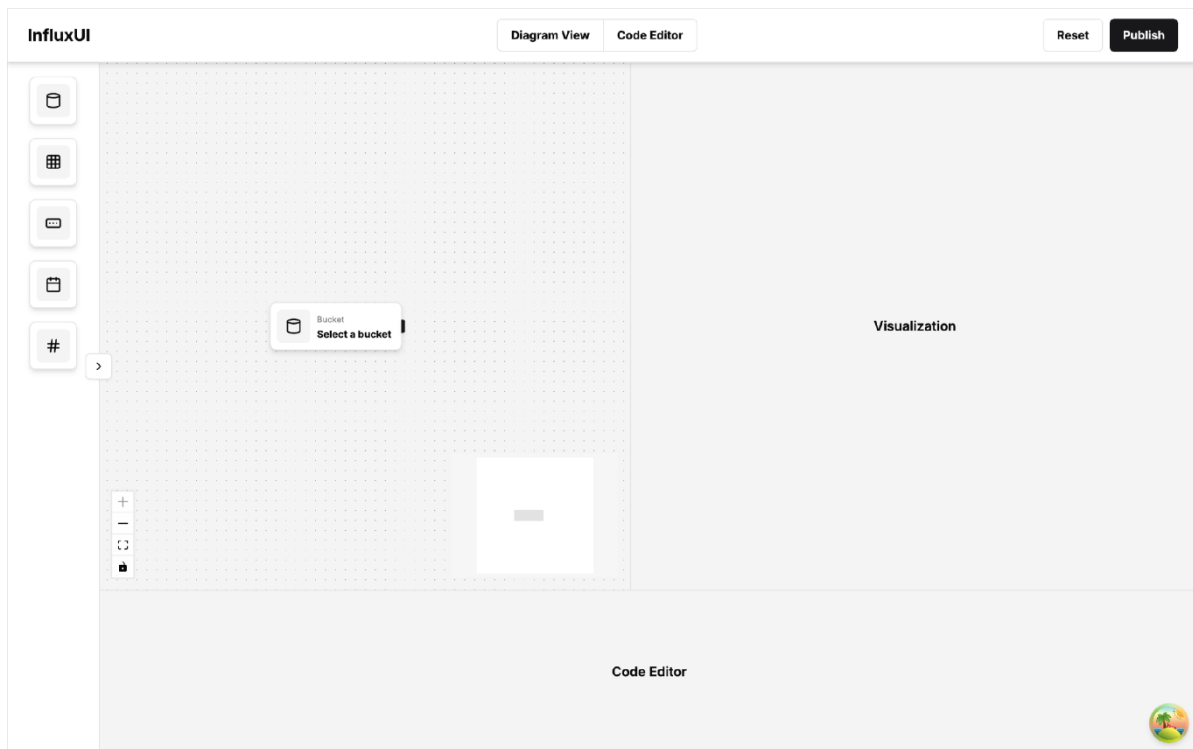
- Frontend: Implement Value Threshold Filter Component



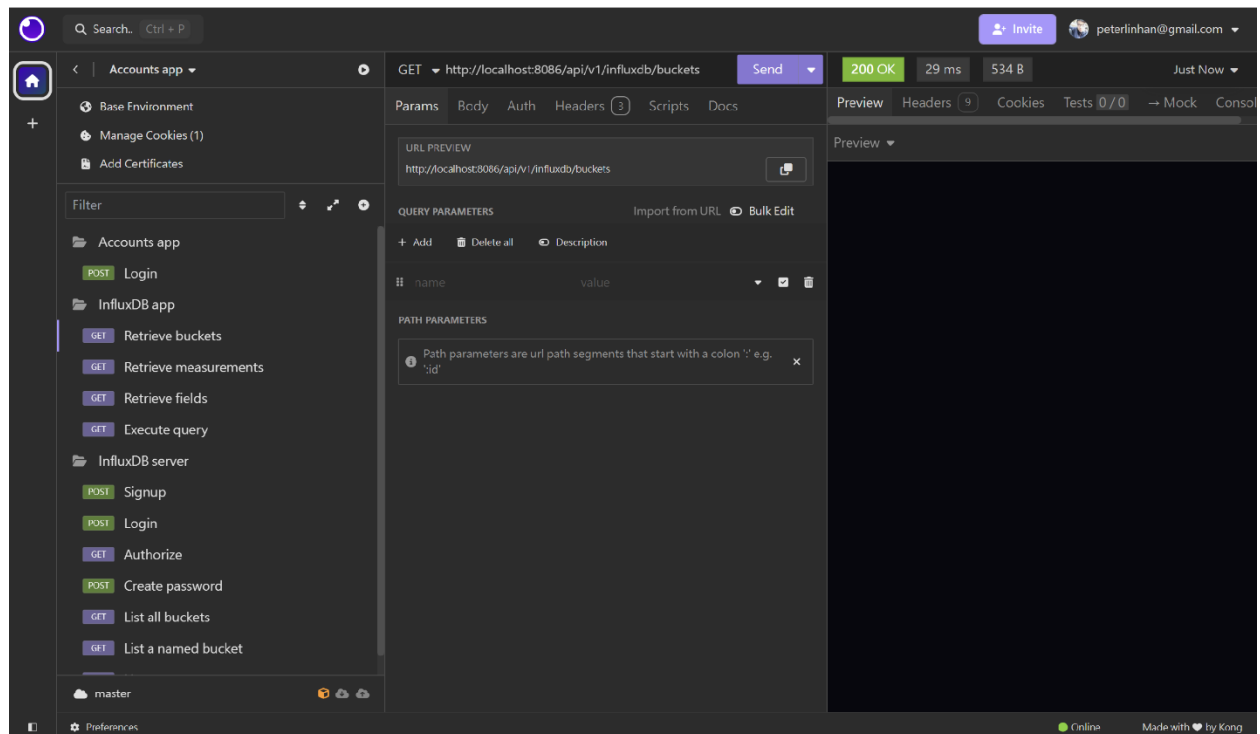
- Frontend: Implement Date Range Filter Component



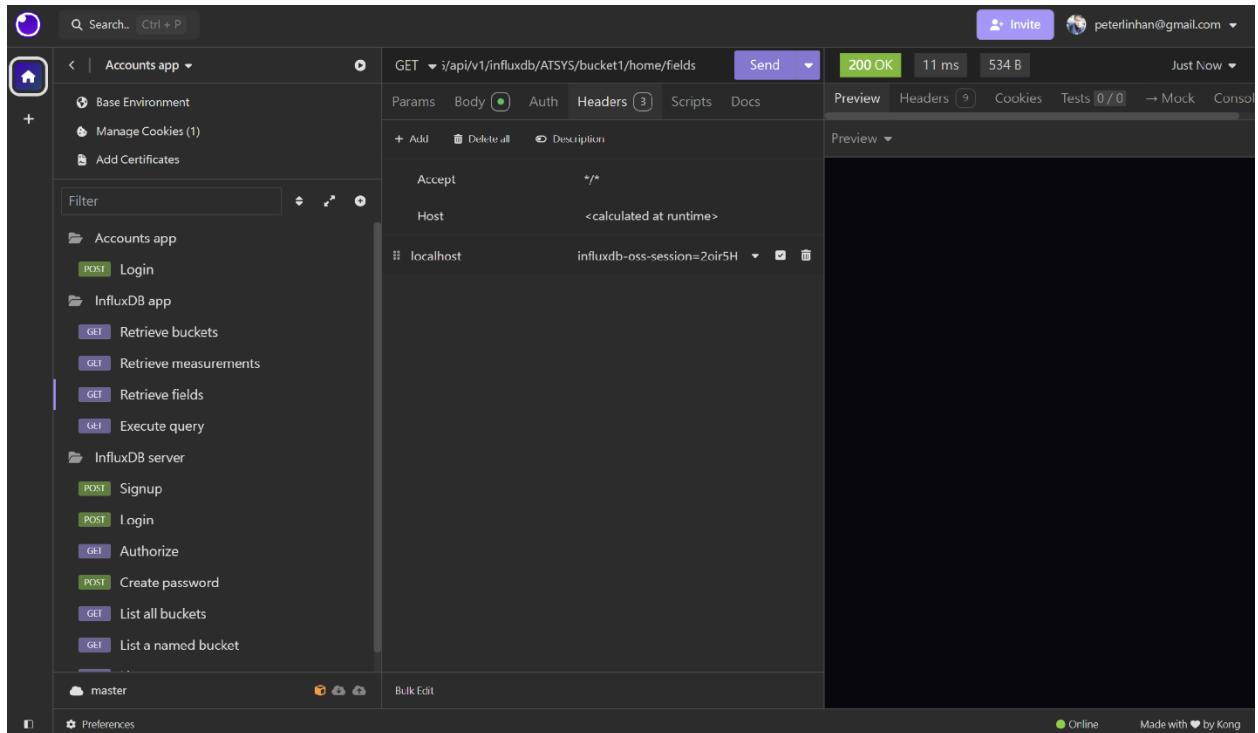
- Frontend: Implement Resizable Multi-Panel Layout



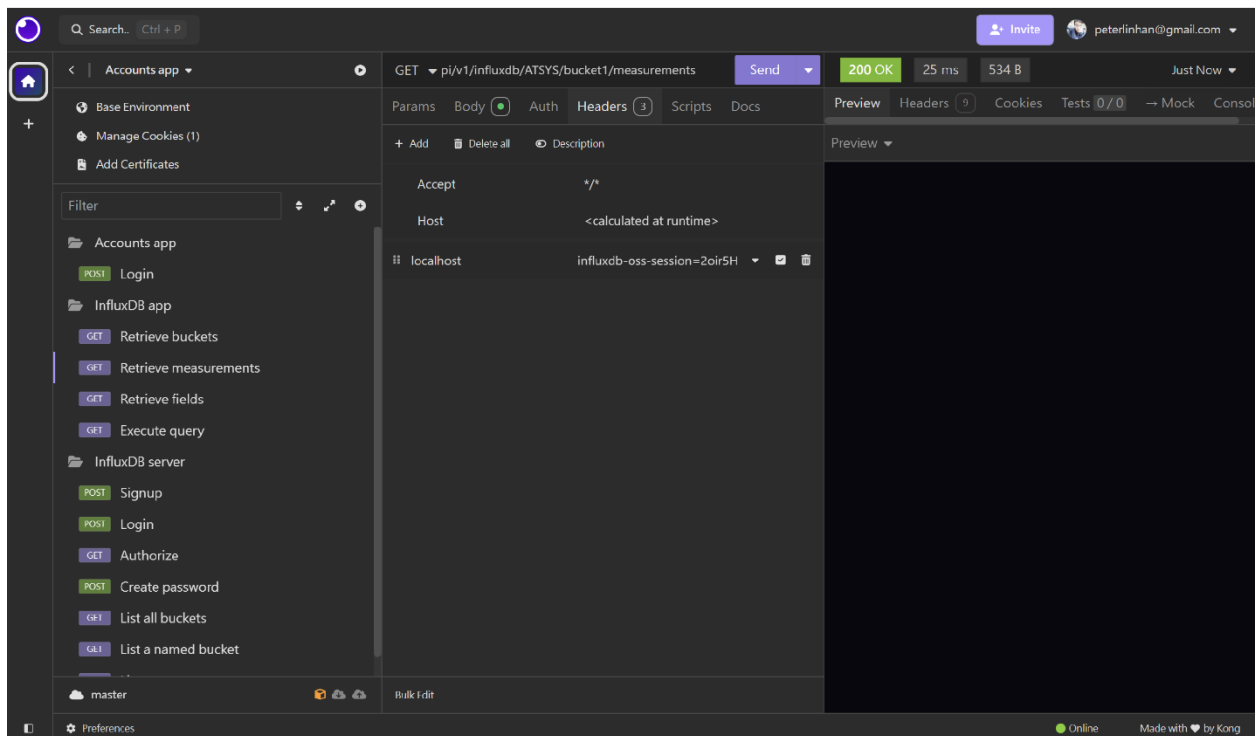
- Backend: retrieve buckets successfully via Insomnia



- Backend: retrieve field



- Backend: retrieve measurements



On the frontend, our team successfully completed two key backlog items: implementing the presentation of buckets and developing date range filtering functionality. And we also finished adding a resizable multi-panel layout for the editor page. Meanwhile, the backend team made strides in updating the authentication mechanism, shifting from token-based to cookie-based authentication for API testing and the 'retrieve buckets' functionality.

However, several items remain in progress, the frontend team still needs to address the presentation of measurements and fields, fix value threshold bugs, and resolve browser compatibility issues. On the backend, work continues retrieving measurements and fields from InfluxDB, as well as executing InfluxDB queries. As we move into sprint planning for the next iteration, our team is focusing on refining user stories, aligning frontend and backend API testing, and conducting user flow and edge case testing.

Definition of Done

- Date range and value thresholds filter components are implemented
- Multi-panel layout is added
- Frontend and backend API testing is aligned
- Buckets, measurements, and fields retrieval using cookies instead of tokens.

Summary of Changes

- Refine user stories 1 and 2 and collaborate with the backend team to fetch data (95% in progress)
- More user flow testing to be conducted once the API of execution of InfluxDB queries is completed.