

# No-Code Solution for InfluxDB Snapshot 1.1

---

## Week 4

---

### Influx UI UG 9

Tyler Chapman (a1851834)  
Li Chen (a1809775)  
Yizhou Chen (a1761339)  
Siqi Gu (a1868071)  
Bradley Hill (a1704585)  
Xin Li (a1868279)  
Haotan Wang (a1809851)  
Zhihan Yang (a1791800)  
Yuzhe Zhang (a1809783)



# 1. Product Backlog and Task Board

## 1.1 Product Backlog

The screenshot shows a GitHub repository named 'INFLUXUI-ATSYS / InfluxUI-UG9'. The 'Product Backlog' tab is selected. The board has four columns: 'New / To Be Prioritized' (3 items), 'High Priority' (0 items), 'Medium Priority' (2 items), and 'Low Priority' (1 item). Each column contains cards with task details and a 'More' button.

Column	Count	Task Description	Added By
New / To Be Prioritized	3	Implement the user login feature Implement a visual display of the list of available buckets, measurements, and fields Implement a drag-and-drop interface that allows users to select buckets, measurements, and fields in InfluxDB	a1791800
High Priority	0		
Medium Priority	2	Prepare the necessary running components of the web application (such as express) Implement the application homepage	a1791800
Low Priority	1	Create user guides and help documents	a1791800

We have analysed some core requirements (especially those in the early stages of the project) and placed them in different columns corresponding to their priority level.

The Product Backlog is the core task list of our project, stored in the projects page of the GitHub repository. It brings together all project-related requirements, features, improvements, and fixes. It represents the overall scope of work for the project and is a dynamic, constantly updated document that records all requirements and work items to be completed during the project development process.

It is sorted by priority and covers not only the work in progress in the current Sprint, but also the work that needs to be completed in the future. Each item can be adjusted and re-evaluated through ongoing team discussions, client consultation, and project progress.

## 1.2 Task Board

The screenshot shows a GitHub Sprint 1 Backlog task board with three columns: To do, In progress, and Done.

- To do:** Contains 4 items:
  - 4. Discuss and sketch the design of the application #7 opened by a1791800
  - Welcome to GitHub projects: A welcome message with instructions and checkboxes for creating a new project, giving it a name, etc.
  - Checklist items: Create a new project (checked), Give your project a name (checked), Press the ? key to see available keyboard shortcuts (unchecked), Add a new column (unchecked), Drag and drop this card to the new column (unchecked), Search for and add issues or PRs to (unchecked).
- In progress:** Contains 3 items:
  - 1. Research: Understanding the Project #4 opened by a1791800
  - 2. Requirements Analysis: Extract and Organize Requirements from User Stories 1-3 #5 opened by a1791800
  - 3. Familiar with the repository and create a local clone #6 opened by a1791800
- Done:** Contains 0 items.

We use kanban in project to manage the backlog in current sprint. It is divided into “to do”, “in progress” and “done” columns.

## 2. Sprint Backlog and User Stories

### 2.1 Sprint Backlog

#### 1. Research: Understanding the Project #4



a1791800 opened this issue 20 hours ago · 0 comments



a1791800 commented 20 hours ago

...

In order to understand the project and master some necessary knowledge, we need to do some research. Here are some materials I recommend.

1. These are the materials provided by the course.

- [InfluxDB Website](#)
- [Grana website](#)
- [InfluxDB OSS v2 documentation](#)
- [Grafana v9.5 documentation](#)

2. Browse some no-code website examples, which can provide good examples and design inspiration for the project.

For example, Google Data Studio is a free and easy-to-use data visualization tool that allows users to create interactive reports and dashboards from a variety of data sources. Users can create various data charts by dragging and dropping components.  
Link: [Google Data Studio](#)

3. Learn or review web development technologies, such as html/css, vue.js, database, etc.

#### 2. Requirements Analysis: Extract and Organize Requirements from User Stories 1~3 #5

Edit

New issue



a1791800 opened this issue 20 hours ago · 0 comments



a1791800 commented 20 hours ago · edited

...

User stories 1~3 contain crucial information about the functionalities we need to implement. This task involves carefully reviewing each user story, identifying both explicit and implicit requirements, and organizing them into a comprehensive list. This process will help us refine our project scope, prioritize tasks, and ensure we don't overlook any critical features.

Assignees

No one—[assign yourself](#)



Labels

None yet



### 3. Familiar with the repository and create a local clone #6

Open

a1791800 opened this issue 19 hours ago · 0 comments



a1791800 commented 19 hours ago

...

The project needs to use GitHub to store code and manage versions.

1. Set up the Git environment
2. Each team member should clone the repository locally
3. Understand how to edit and submit code
4. Collaborative development: Understand how to manage branches, protect the main branch, resolve conflicts, etc.

### 4. Discuss and sketch the design of the application #7

Open

a1791800 opened this issue 38 minutes ago · 0 comments



a1791800 commented 38 minutes ago

...

Use a collaborative design online tool such as [Figma](#) to sketch out the design of our application.

We manage and track the task board through the issues page in our GitHub repository.

We currently have a list of things to accomplish in sprint 1.

## **2.2 User Stories**

As a user, I hope to use a drag and drop interface to filter the selected data, in order to optimize the query process without the need to write complex query code.

**Role:** User

**Pre-conditions:**

- The user has selected several data items using the drag and drop interface of the Influx DB UI.
- The data source is ready for filtering.

**Main flow:**

- Users access the filtering interface in a no code (UI) interface.
- The user selects specific filtering criteria.

- 
- Users set parameters for the filter, such as date range and data threshold.
  - The interface performs queries and jumps based on user input

**Post-condition:**

- The user's filter has been applied to the selected data, and the query has been optimized.
- The system is ready to execute queries with application filters.

**Acceptance criteria:**

- The interface must allow users to drag and drop filters onto the selected data fields.
- Filters should accurately respond to user input.
- The Influx DB UI interface should provide clear feedback on the impact of filters on data selection.

---

## Definition of Done

The standard for done a coding task is: The code has been written according to the standards in the report, reviewed, tested, modified by the team, passed peer review, and received unanimous recognition from all team members.

The standard for done a non-coding task is: The relevant tasks have been discussed, decided upon, and implemented during the group meeting, and all team members have reached a consensus, task will start shortly

## Summary of Changes

At the beginning of the project, we mainly focused on task planning and work deployment. Through close communication with clients on Teams, we gained a deep understanding of their needs and expectations, such as identifying audience targets, explicit the operating system environments, and the budget limits. In order to not deviate from the project objectives during the process, the team chose to convert specific requirements into user stories after in-depth discussions. At the same time, realizing the importance of programming languages, we have decided to make a professional technical training and guidance for all team members. In addition, we are already start to building development environment to ensure that project requirements can be effectively met in the upcoming sprint. Through these initial efforts, we believe the project has begun to develop in a positive direction

# No-Code Solution for InfluxDB Snapshot 2.1

---

## Week 5

---

### Influx UI UG 9

Tyler Chapman (a1851834)  
Li Chen (a1809775)  
Yizhou Chen (a1761339)  
Siqi Gu (a1868071)  
Bradley Hill (a1704585)  
Xin Li (a1868279)  
Haotan Wang (a1809851)  
Zhihan Yang (a1791800)  
Yuzhe Zhang (a1809783)



# Product Backlog and Task Board

## 1.1 Product Backlog

The screenshot shows a digital product backlog board with the following structure:

- New / To Be Prioritized (2 items):**
  - Implement the filtering feature:
    - A drag-and-drop interface enables filters to be applied to user-selected data.
    - The code-free interface should have filter options.
    - Users are able to drag and drop filter criteria onto selected data fields.
    - Users can set parameters for filters (e.g., date range, value threshold)
    - This interface prepares a filtered query based on user input.
    - The interface should provide clear feedback on how filters affect data selection.
  - Added by a1791800
  - High Priority (4 items):**
    - The project will be based on user stories ...  
Added by a1809851
    - Prepare the necessary running components \*\*\* of the web application (such as express) ...  
Added by a1791800
    - Implement the application homepage. ...  
Added by a1791800
    - Implement the user login feature. ...  
Added by a1791800
  - Medium Priority (3 items):**
    - Implement a visual display of the list of available buckets, measurements, and fields. ...  
Added by a1791800
    - Implement a drag-and-drop interface that \*\*\* allows users to select buckets, measurements, and fields in InfluxDB ...  
Added by a1791800
    - Implementing a feature that makes it possible to successfully prepare data sources without writing any code. ...  
Added by a1791800
  - Low Priority (1 item):**
    - Create user guides and help documents. ...  
Added by a1791800

We analyzed some core requirements (especially those in the early stages of the project) and placed them in different columns corresponding to their priorities.

The difference with the last week is, the priority of some tasks has changed and been raised to high priority, and the next work will gradually revolve around the level of priority.

In group discussions, project development will be based on user story, such as "As a developer, I will determine the first task to be completed based on task priority, such as prepare the necessary running components of the web application." Subsequent user stories will also be developed based on this. To ensure that the project is closely related and avoid breakage things happened.

## 1.2 Task Board

The image shows a GitHub project task board with three columns: "To do", "In progress", and "Done".

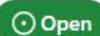
- To do:** Contains 9 items.
  - 8. Familiar with the repository and create a local clone
  - 9. Long term mind mapping plan
  - Welcome to GitHub projects! (instructions)
  - Group meeting
- In progress:** Contains 3 items.
  - 10. Code division of labour
  - 11. Discuss and sketch the design of the application
  - 12. Long-Term Objectives
- Done:** Contains 8 items.
  - 1. Research: Understanding the Project
  - 2. Requirements Analysis: Extract and Organize Requirements from User Stories 1~3
  - 3. Research and analysis: Deep research project, analysis code direction
  - 4. Project system analysis: Review requirements from provided user stories and create task cards
  - 5. Confirm the overall framework of the code
  - 6 Research Group Project
  - 7 Progress of Initial Report and snapshot 1.1
  - Requirement Analysis: User story 1

We used the board in the project to manage the backlog in current sprint. It is divided into “to do”, “in progress” and “done” columns. The changes of Sprint1 to Sprint2 are some of “to do” task was already finished and move to “done” column, the new “to do” item has been filled in.

## Sprint Backlog and User Stories

### 2.1 Sprint Backlog

## 8. Familiar with the repository and create a local clone #6

 Open a1791800 opened this issue 2 weeks ago · 0 comments



a1791800 (Zhihan Yang) commented 2 weeks ago

...

The project needs to use GitHub to store code and manage versions.

1. Set up the Git environment
2. Each team member should clone the repository locally
3. Understand how to edit and submit code
4. Collaborative development: Understand how to manage branches, protect the main branch, resolve conflicts, etc.

## 9. Long term mind mapping plan #18

 Open a1809851 opened this issue yesterday · 0 comments



a1809851 commented yesterday

...

As the project progresses, the mind map is modified and improved based on the situation of development of the project.

## 10. Code division of labour #17

 Open a1809851 opened this issue yesterday · 0 comments



a1809851 commented yesterday

...

Each team member is responsible for a portion, submit code from Github for other team members to view and analyze later.

1. Configuration GitHub environment
2. Make sure all team members joined GitHub group InfluxUI-UG9
3. Get to know how to edit and submit code on GitHub
4. Group work- Get to know group members code style and collaborative development are also necessary, the reason is in the later stages of the project, all team members are required to submit the code they are responsible for. If there is no collaborative development from start at all, the consequences can be catastrophic.

## 11. Discuss and sketch the design of the application #7

 Open a1791800 opened this issue 2 weeks ago · 0 comments



a1791800 (Zhihan Yang) commented 2 weeks ago

...

Use a collaborative design online tool such as [Figma](#) to sketch out the design of our application.

## 12. Long-Term Objectives #21

 Open

a1809775 opened this issue yesterday · 0 comments



a1809775 (Li Chen) commented yesterday

...

A long-term planning goal is required to complete the remaining work including reporting, searching data and code implementation

We manage and track the task board through the issues page in our GitHub repository.

We currently have three things to accomplish and two things in progress in sprint 2.

### **2.2 User Stories**

As a user, I want to create and customize real-time data visualization dashboards so that I can monitor and analyze key metrics efficiently without requiring deep technical knowledge.

Role: User 2

Pre-conditions:

- Users can read and apply relevant information by clicking on the influx DB platform
- Users have a basic understanding of what they want to visualize.
- The data sources are configured correctly and transferred to the influx db.

Main Flow:

1. The user is able to access the dashboard creation interface in the Influx DB User Interface.
2. The user selects starts with a blank canvas.
3. Users can customize the appearance of the dashboard.
4. The dashboard displays real-time data and chart analysis based on the user's configuration.

---

Post-condition:

- The user has created a functional real-time dashboard that can accurately display the selected data.
- The system automatically updates real-time data from the dashboard.

Acceptance Criteria:

- When users customize the dashboard, Influx DB UI should provide a clear preview.
- Visualizations should be updated automatically and be able to accurately reflect the details of the selected data.
- The dashboard creation interface allows users to easily drag and drop widgets.

## Definition of Done

The standard for done a coding task is: The code has been written according to the standards in the report, reviewed, tested, modified by the team, passed peer review, and received unanimous recognition from all team members.

The standard for done a non-coding task is: The relevant tasks have been discussed, decided upon, and implemented during the group meeting, and all team members have reached a consensus, task will start shortly.

---

# Summary of Changes

This week, we advanced the project preparation. Through research, the team deeply understood the project requirements and related technologies. After communicating with the product owner, we identified 3 key user stories and completed the preliminary product backlog. This marks the completion of the current stage of feature.

The team has built a development environment and is working on designing the system architecture. The next sprint will focus on modelling, including UI prototyping and data flow diagrams, while deepening the understanding of InfluxDB's Flux query language and Grafana.

In addition, we plan to start implementing the front-end framework of the drag-and-drop query builder in sprint 2.

The main challenge currently faced is that the team is not familiar enough with InfluxDB's Flux query language, and we are solving this problem through intensive learning.

# No-Code Solution for InfluxDB Snapshot 3.1

---

## Week 7

---

### Influx UI UG 9

Tyler Chapman (a1851834)  
Li Chen (a1809775)  
Yizhou Chen (a1761339)  
Siqi Gu (a1868071)  
Bradley Hill (a1704585)  
Xin Li (a1868279)  
Haotan Wang (a1809851)  
Zhihan Yang (a1791800)  
Yuzhe Zhang (a1809783)



# Product Backlog and Task Board

## 1.1 Product Backlog

● Backlog 5 / 5 ...

This is your project Backlog

● InfluxUI-UG9 #2 ...

User Story 2 : Filter Application via Drag-and-Drop

● InfluxUI-UG9 #3

User Story 3: Automatic Query Generation and Execution

● InfluxUI-UG9 #35 

User Story 4: Secure User Login

● InfluxUI-UG9 #36

User Story 5: Intuitive Main Interface for Data Analysis

+ Add item

## 1.2 Task Board

Author ▾		Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
23 Open	10 Closed					
<input type="checkbox"/>	FRONTEND 7: UI design, testing and support (0/2) #34 opened 24 minutes ago by a1791800					
<input type="checkbox"/>	FRONTEND 6: Execute query button and result display area (0/1) #33 opened 32 minutes ago by a1791800 6 tasks					
<input type="checkbox"/>	FRONTEND 5: Query building area (0/1) #32 opened 43 minutes ago by a1791800 6 tasks					
<input type="checkbox"/>	FRONTEND 4: Drag and drop basics (1/2) #31 opened 49 minutes ago by a1791800 4 tasks					
<input type="checkbox"/>	FRONTEND 3: Data source page (0/1) #30 opened 1 hour ago by a1791800 6 tasks					
<input type="checkbox"/>	FRONTEND 2: Main interface layout (1/1) #29 opened 1 hour ago by a1791800 5 tasks					
<input type="checkbox"/>	FRONTEND 1: Login interface (0/1) #28 opened 1 hour ago by a1791800 5 tasks					
<input type="checkbox"/>	13. Project progress #27 opened 4 hours ago by a1809851					
<input type="checkbox"/>	Create a drag-and-drop interface. #26 opened 3 days ago by a1868071					
<input type="checkbox"/>	Create a drag-and-drop interface #25 opened 3 days ago by a1868071					
<input type="checkbox"/>	Create an express server #24 opened 3 days ago by a1868071					

## Sprint Backlog and User Stories

### 2.1 Sprint Backlog

<p><span style="color: red;">●</span> Planning 8 This is your planning stage</p> <ul style="list-style-type: none"><li>InfluxUI-UG9 #28 *** FRONTEND 1: Login interface (0/1)</li><li>InfluxUI-UG9 #30 FRONTEND 3: Data source page (0/1)</li><li>InfluxUI-UG9 #31 FRONTEND 4: Drag and drop basics (1/2)</li><li>InfluxUI-UG9 #32 FRONTEND 5: Query building area (0/1)</li><li>InfluxUI-UG9 #33 FRONTEND 6: Execute query button and result</li></ul> <p>+ Add item</p>	<p><span style="color: orange;">●</span> In Progress 6 This is actively being worked on</p> <ul style="list-style-type: none"><li>InfluxUI-UG9 #29 FRONTEND 2: Main interface layout (1/1)</li><li>InfluxUI-UG9 #26 Create a drag-and-drop interface.</li><li>InfluxUI-UG9 #6 8. Familiar with the repository and create a local clone</li><li>InfluxUI-UG9 #18 9. Long term mind mapping plan</li><li>InfluxUI-UG9 #21</li></ul> <p>+ Add item</p>	<p><span style="color: pink;">●</span> Testing 0 This is your testing phase</p> <ul style="list-style-type: none"></ul>	<p><span style="color: purple;">●</span> Done 8 This has been completed</p> <ul style="list-style-type: none"><li>InfluxUI-UG9 #16 5. Confirm the overall framework of the code</li><li>InfluxUI-UG9 #14 6 Research Group Project</li><li>InfluxUI-UG9 #19 7 Progress of Initial Report and snapshot 1.1</li><li>InfluxUI-UG9 #24 Create an express server</li></ul> <p>+ Add item</p>
---	---	---	--

## 2.2 User Stories

 InfluxUI-UG9 #1	
User story 1: Drag-and-Drop Interface for Selecting Data Sources	
 InfluxUI-UG9 #2	
User Story 2 : Filter Application via Drag-and-Drop	
 InfluxUI-UG9 #3 ...	
User Story 3: Automatic Query Generation and Execution	

## Definition of Done

The standard for done a coding task is: The code has been written according to the standards in the report, reviewed, tested, modified by the team, passed peer review, and received unanimous recognition from all team members.

The standard for done a non-coding task is: The relevant tasks have been discussed, decided upon, and implemented during the group meeting, and all team members have reached a consensus, task will start shortly.

---

# Summary of Changes

This week, we will continue to push forward with the project. Through offline meetings, both code and non-code parts were properly allocated, and each team member was required to share their progress on their parts of the project in a weekly basis. The entire project will be developed in four stages: front-end, back-end, summation and reporting. The front-end has entered the in-process section in task board, which means this stage will be completed soon.

Group has completed the server setup, and the UI style are also confirmed as well. The next sprint will focus on development the front-end part of the project, including the login interface(login.html) and main interface(index.html) layout. While continuing to deepen the understanding of Influx DB's Flux query language and Grafana.

In addition, we plan to implement some front-end features in Sprint 3 thus to catch up the project schedule.

The main challenge currently faced is that the group was still not proficient enough in using InfluxDB's Flux query language. In future, daily meetings will paying more attention to strengthen training and learning this programming language, until all of team members can skilled use it.

# No-Code Solution for InfluxDB Snapshot 3.2

---

## Week 8

---

### Influx UI UG 9

Tyler Chapman (a1851834)  
Li Chen (a1809775)  
Yizhou Chen (a1761339)  
Siqi Gu (a1868071)  
Bradley Hill (a1704585)  
Xin Li (a1868279)  
Haotan Wang (a1809851)  
Zhihan Yang (a1791800)  
Yuzhe Zhang (a1809783)



# Product Backlog and Task Board

## 1.1 Product Backlog

Backlog 5 / 5 ...

This is your project Backlog

InfluxUI-UG9 #1 ... 

User story 1: Drag-and-Drop Interface for Selecting Data Sources

InfluxUI-UG9 #2

User Story 2 : Filter Application via Drag-and-Drop

InfluxUI-UG9 #3

User Story 3: Automatic Query Generation and Execution

InfluxUI-UG9 #35 

User Story 4: Secure User Login

+ Add item

## **1.2 Task Board**

Author ▾		Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	23 Open	<input checked="" type="checkbox"/> 13 Closed				
<input type="checkbox"/>	<b>Collaborative Documents: Design Sketches</b>	#37 opened 3 weeks ago by a1791800				
<input type="checkbox"/>	<b>User Story 5: Intuitive Main Interface for Data Analysis</b>	#36 opened 3 weeks ago by a1791800				
<input type="checkbox"/>	<b>User Story 4: Secure User Login</b>	#35 opened 3 weeks ago by a1791800				
<input type="checkbox"/>	<b>FRONTEND 7: UI design, testing and support (1/2)</b>	#34 opened 3 weeks ago by a1791800				
<input type="checkbox"/>	<b>FRONTEND 6: Execute query button and result display area (0/1)</b>	#33 opened 3 weeks ago by a1791800	6 tasks			
<input type="checkbox"/>	<b>FRONTEND 5: Query building area (1/1)</b>	#32 opened 3 weeks ago by a1791800	1 of 6 tasks			
<input type="checkbox"/>	<b>FRONTEND 4: Drag and drop basics (1/2)</b>	#31 opened 3 weeks ago by a1791800	4 tasks done			
<input type="checkbox"/>	<b>FRONTEND 3: Data source page (0/1)</b>	#30 opened 3 weeks ago by a1791800	6 tasks			
<input type="checkbox"/>	<b>FRONTEND 2: Main interface layout</b>	#29 opened 3 weeks ago by a1791800	5 tasks done			

# Sprint Backlog and User Stories

## 2.1 Sprint Backlog

● In Progress 7 ...

This is actively being worked on

● InfluxUI-UG9 #30 ...  
FRONTEND 3: Data source page (0/1)

● InfluxUI-UG9 #32  
FRONTEND 5: Query building area (1/1)

● InfluxUI-UG9 #33  
FRONTEND 6: Execute query button and result display area (0/1)

● InfluxUI-UG9 #37  
Collaborative Documents: Design Sketches

---

 InfluxUI-UG9 #18 ...

9. Long term mind mapping plan

 InfluxUI-UG9 #21

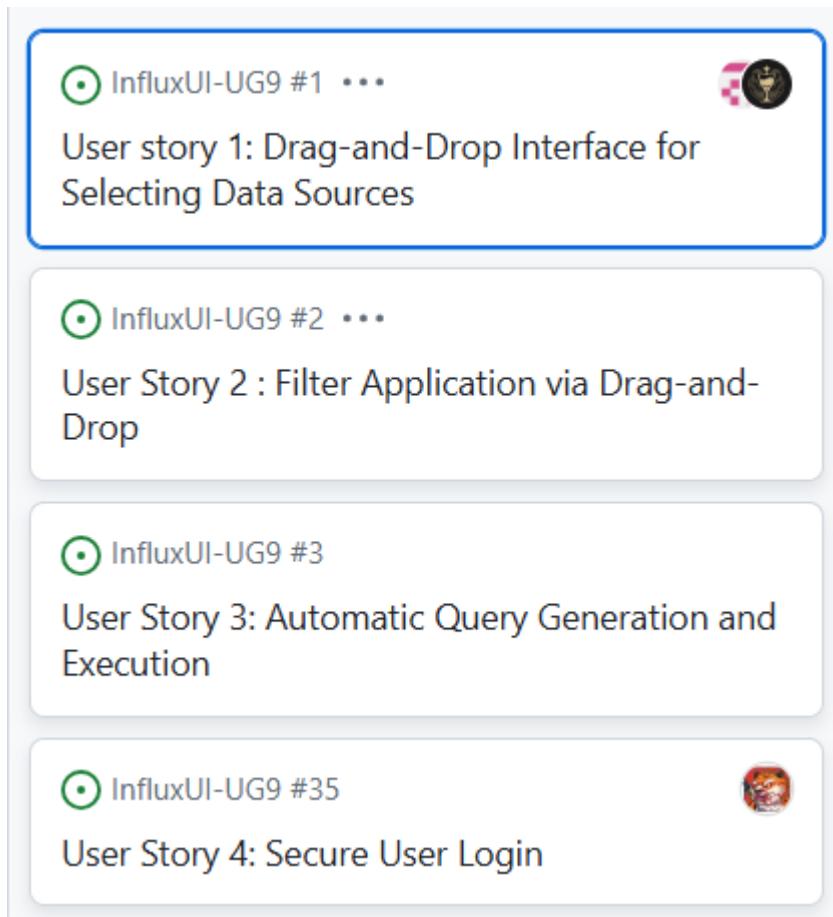
12. Long-Term Objectives

 InfluxUI-UG9 #27

13. Project progress

+ Add item

## 2.2 User Stories



## Definition of Done

The standard for done a coding task is: The code has been written according to the standards in the report, reviewed, tested, modified by the team, passed peer review, and received unanimous recognition from all team members.

The standard for done a non-coding task is: The relevant tasks have been discussed, decided upon, and implemented during the group meeting, and all team members have reached a consensus, task will start shortly.

---

# Summary of Changes

This time, we mainly followed up the front-end design project, and mainly completed the login interface (login.html) task: designed and implemented the basic login page, including the username and password input box, as well as the login button. The Skills required and achieved by Basic HTML and CSS.2. Main interface layout (index.html). Task: Create the basic layout of the main page, including the top navigation bar, sidebar (to display available data sources), and main content area. Skills required: HTML and CSS, basic page layout.3. Data source shows responsible person: Task: Display a list of available buckets, measurements, and fields in the sidebar. Skills required: HTML, CSS, basic JavaScript for dynamically loading data.

In the front-end design, we added NodeJS, express and bootstrap3 technologies. At the same time, faced with some disputes over details, the team finally decided to try to clean the appearance as much as possible. The button clicked on the page has feedback instead of directly clicking the display screen, which will make the customer experience more hierarchical.

# No-Code Solution for InfluxDB Snapshot 4.1

---

## Week 10

---

### Influx UI UG 9

Tyler Chapman (a1851834)  
Li Chen (a1809775)  
Yizhou Chen (a1761339)  
Siqi Gu (a1868071)  
Bradley Hill (a1704585)  
Xin Li (a1868279)  
Haotan Wang (a1809851)  
Zhihan Yang (a1791800)  
Yuzhe Zhang (a1809783)



# Product Backlog and Task Board

## 1.1 Product Backlog

Backlog 5 / 5 ...

This is your project Backlog

InfluxUI-UG9 #2  
User Story 2 : Filter Application via Drag-and-Drop

InfluxUI-UG9 #3  
User Story 3: Automatic Query Generation and Execution

InfluxUI-UG9 #35  
User Story 4: Secure User Login

InfluxUI-UG9 #36  
User Story 5: Intuitive Main Interface for Data Analysis

InfluxUI-UG9 #38  
User Story 4: Real-Time Data Preview in No-Code Interface

## 1.2 Task Board

Author		Label	Projects	Milestones	Assignee	Sort
22 Open	✓ 20 Closed					
<input type="checkbox"/>	<span>● User Story 7: Execute Query Button And Result Display Area</span> #43 opened yesterday by a1809851					
<input type="checkbox"/>	<span>● User Story 6: Data Source Display</span> #42 opened yesterday by a1809851					
<input type="checkbox"/>	<span>● Implementation of data source function</span> #41 opened yesterday by a1791800 2 tasks					
<input type="checkbox"/>	<span>● InfluxDB Login Integration</span> #40 opened 2 days ago by a1791800 2 tasks					
<input type="checkbox"/>	<span>● FRONTEND 8: SPA for core functions</span> #39 opened 2 days ago by a1791800					
<input type="checkbox"/>	<span>● User Story 4: Real-Time Data Preview in No-Code Interface</span> #38 opened 4 days ago by a1872694					

## Sprint Backlog and User Stories

### 2.1 Sprint Backlog

● In Progress 1 ...

This is actively being worked on

● InfluxUI-UG9 #40  
InfluxDB Login Integration

Done 19 ...  
This has been completed

- InfluxUI-UG9 #27  
13. Project progress
- InfluxUI-UG9 #21 ...  
[12. Long-Term Objectives](#)
- InfluxUI-UG9 #18  
9. Long term mind mapping plan
- InfluxUI-UG9 #17  
10. Code division of labour
- InfluxUI-UG9 #34   
FRONTEND 7: UI design, testing and support (1/2)
- InfluxUI-UG9 #29  
FRONTEND 2: Main interface layout
- InfluxUI-UG9 #28  
FRONTEND 1: Login interface (0/1)

Testing 5 ...  
This is your testing phase

- InfluxUI-UG9 #39  
FRONTEND 8: SPA for core functions
- InfluxUI-UG9 #32   
FRONTEND 5: Query building area (1/1)
- InfluxUI-UG9 #33  
FRONTEND 6: Execute query button and result display area (0/1)
- InfluxUI-UG9 #30  
FRONTEND 3: Data source page (0/1)
- InfluxUI-UG9 #37  
Collaborative Documents: Design Sketches

## 2.2 User Stories

- 
- User Story 7: Execute Query Button And Result Display Area**  
#43 opened yesterday by a1809851
  - User Story 6: Data Source Display**  
#42 opened yesterday by a1809851

- 
- User Story 5: Intuitive Main Interface for Data Analysis**  
#36 opened last month by a1791800

InfluxUI-UG9 #1...

User story 1: Drag-and-Drop Interface for Selecting Data Sources

InfluxUI-UG9 #2...

User Story 2 : Filter Application via Drag-and-Drop

InfluxUI-UG9 #3

User Story 3: Automatic Query Generation and Execution

InfluxUI-UG9 #35

User Story 4: Secure User Login

## Definition of Done

The completion criteria for a coding task are as follows: The code must be developed in strict adherence to the established standards outlined in the report. It should undergo thorough review, testing, and iteration by the team, successfully pass peer review, and gain unanimous approval from all team members.

---

For non-coding tasks, completion is defined by the task being thoroughly discussed and agreed upon in a group meeting. The decisions must reach full consensus among all team members, and implementation is to commence promptly thereafter.

## Summary of Changes

In this phase of the project, we continued to follow up the front-end design work and further improve each part. First of all, in the login.html interface, I designed and implemented a simple login page, including user name and password input boxes and a responsive login button, mainly using the basic skills of HTML and CSS.

Second, we completed the main interface layout (index.html), which includes a top navigation bar, a sidebar for displaying data sources, and the main content area. This part of the implementation is also based on HTML and CSS layout design.

At the same time, we have optimized the data source presentation feature, which is responsible for dynamically displaying the list of buckets, measurement, and fields in the sidebar, and this part combines HTML, CSS, and basic JavaScript to dynamically load data.

Throughout the project, we introduced technologies such as Node.js, Express, and Bootstrap 3 to optimize the front-end design. In the face of disagreements over some design details, the team finally decided on a more concise visual style and added feedback effects for button clicks on the page to make the user experience more

---

hierarchical. We further improved the integration of these functions to ensure a smoother overall design.



---

# No-Code Solution for InfluxDB Snapshot 4.2

---

## Week 10

---

### Influx UI UG 9

Tyler Chapman (a1851834)

Li Chen (a1809775)

Yizhou Chen (a1761339)

Siqi Gu (a1868071)

Bradley Hill (a1704585)

Xin Li (a1868279)

Haotan Wang (a1809851)

Zhihan Yang (a1791800)

Yuzhe Zhang (a1809783)

### Influx UI UG 9

Tyler Chapman (a1851834)

Li Chen (a1809775)

Yizhou Chen (a1761339)

Siqi Gu (a1868071)

Bradley Hill (a1704585)

Xin Li (a1868279)

Haotan Wang (a1809851)

Zhihan Yang (a1791800)

---

Yuzhe Zhang (a1809783)

# Product Backlog and Task Board

## 1.1 Product Backlog

Backlog 3 / 5 ...

This is your project Backlog

● InfluxUI-UG9 #42 ...

User Story 6: Data Source Display

● InfluxUI-UG9 #43

User Story 7: Execute Query Button And Result Display Area

● InfluxUI-UG9 #38

User Story 4: Real-Time Data Preview in No-Code Interface

+ Add item

## 1.2 Task Board

		Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	<span>24 Open</span>	<span>✓ 20 Closed</span>					
<input type="checkbox"/>	<span> CORE 1: Data-source page</span>						
	#45	opened 15 hours ago by a1791800	 3 tasks				
<input type="checkbox"/>	<span> TESTING: InfluxDB Setup and Testing Guide for New Login Functionality</span>						
	#44	opened last week by a1791800					
<input type="checkbox"/>	<span> User Story 7: Execute Query Button And Result Display Area</span>						
	#43	opened last week by a1809851					
<input type="checkbox"/>	<span> User Story 6: Data Source Display</span>						
	#42	opened last week by a1809851					
<input type="checkbox"/>	<span> Implementation of data source function</span>						
	#41	opened last week by a1791800	 2 tasks				
<input type="checkbox"/>	<span> InfluxDB Login Integration</span>						
	#40	opened last week by a1791800	 2 tasks				
<input type="checkbox"/>	<span> FRONTEND 8: SPA for core functions</span>						
	#39	opened last week by a1791800					
<input type="checkbox"/>	<span> User Story 4: Real-Time Data Preview in No-Code Interface</span>	<span>good first issue</span>					
	#38	opened 2 weeks ago by a1872694					

## Sprint Backlog and User Stories

### 2.1 Sprint Backlog

## ● In Progress 2

...

This is actively being worked on

● InfluxUI-UG9 #41

Implementation of data source function

● InfluxUI-UG9 #45 ...



CORE 1: Data-source page

+ Add item

**Testing** 1

This is your testing phase

InfluxUI-UG9 #37  
Collaborative Documents: Design Sketches

**Done** 29

This has been completed

InfluxUI-UG9 #36  
User Story 5: Intuitive Main Interface for Data Analysis

InfluxUI-UG9 #35  
User Story 4: Secure User Login

InfluxUI-UG9 #3  
User Story 3: Automatic Query Generation and Execution

InfluxUI-UG9 #40  
InfluxDB Login Integration

InfluxUI-UG9 #44  
TESTING: InfluxDB Setup and Testina Guide for

+ Add item

## 2.2 User Stories

## User Story 8: Backend setup #46

 Open

a1809851 opened this issue now · 0 comments



a1809851 commented now

...

As a user, I want to create a page that displays a list of buckets, allowing users to click on the bucket they want to query

Role: Developer

Pre-condition

Users can click on the page to find the bucket they want to query.

Users after clicking on the bucket, the measurements and fields within the bucket will be displayed, and users can drag them into the selection area.

Main Flow:

The users can find the search area on the main page and enter relevant information.

The users interact with the search system and the system records relevant information.

The users search histories in the search bar will be retained and recorded by the data center for the next search.

## Definition of Done

The standard for done a coding task is: The code has been written according to the standards in the report, reviewed, tested, modified by the team, passed peer review, and received unanimous recognition from all team members.

The standard for done a non-coding task is: The relevant tasks have been discussed, decided upon, and implemented during the group meeting, and all team members have reached a consensus, task will start shortly.

## Summary of Changes

We have completed the construction of the back-end code, the front and back ends have completed the link, and can work smoothly. On a technical level, we made appropriate changes to the code this week, integrating the back-end code into the

---

drag-and-drop page. In detail, the page should generate a list of buckets and let the user click to search for the bucket they need to query. After you click the bucket, the measured values and fields in the bucket are displayed. You can drag them to the selected area. After dragging, click the next button to go to the query builder page to follow the process. Users pass the data to the next step by selecting a bucket and dragging the desired measurements and fields on the data source page, and then applying filters on the query builder page to generate and execute the full influxDB query in the background.

# No-Code Solution for InfluxDB Snapshot 5.1

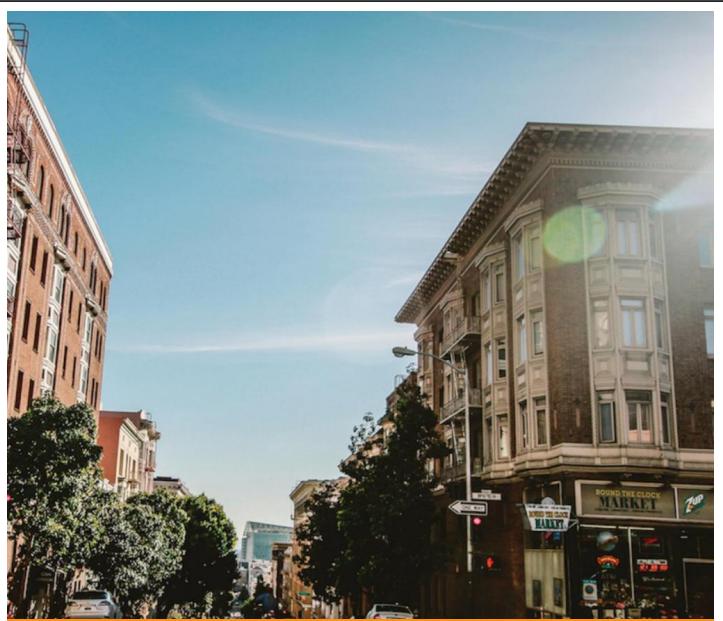
---

Week **11**

---

## Influx UI UG 9

Tyler Chapman (a1851834)  
Li Chen (a1809775)  
Yizhou Chen (a1761339)  
Siqi Gu (a1868071)  
Bradley Hill (a1704585)  
Xin Li (a1868279)  
Haotan Wang (a1809851)  
Zhihan Yang (a1791800)  
Yuzhe Zhang (a1809783)



# Product Backlog and Task Board

## 1.1 Product Backlog

Backlog 3 / 5 ...

This is your project Backlog

• InfluxUI-UG9 #42 ...

User Story 6: Data Source Display

• InfluxUI-UG9 #43

User Story 7: Execute Query Button And Result Display Area

• InfluxUI-UG9 #38

User Story 4: Real-Time Data Preview in No-Code Interface

+ Add item

## 1.2 Task Board

Filters ▾		Q is:issue is:open	Labels 9	Milestones 0	New issue		
		Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	16 Open	<input checked="" type="checkbox"/> 33 Closed					
<input type="checkbox"/>	TEST: data source page	#50 opened 2 days ago by a1791800					
<input type="checkbox"/>	User story 9: Technique connect	#48 opened 2 days ago by a1809851					
<input type="checkbox"/>	About the login function <span>help wanted</span>	#47 opened 5 days ago by a1791800					
<input type="checkbox"/>	User Story 8: Backend setup	#46 opened last week by a1809851					
<input type="checkbox"/>	CORE 1: Data-source page	#45 opened last week by a1791800	1 of 3 tasks				
<input type="checkbox"/>	User Story 7: Execute Query Button And Result Display Area	#43 opened 2 weeks ago by a1809851					
<input type="checkbox"/>	User Story 6: Data Source Display	#42 opened 2 weeks ago by a1809851					
<input type="checkbox"/>	Implementation of data source function	#41 opened 2 weeks ago by a1791800	2 tasks				
<input type="checkbox"/>	User Story 4: Real-Time Data Preview in No-Code Interface <span>good first issue</span>	#38 opened 3 weeks ago by a1872694					

# Sprint Backlog and User Stories

## 2.1 Sprint Backlog

The image shows a digital sprint backlog board with three columns: In Progress, Testing, and Done. Each column has a header card with a status icon, a count of items, and a '...' button. Below each header are several item cards, each with a circular icon, a unique ID, and a brief description. At the bottom of each column is a '+ Add item' button.

In Progress	Testing	Done
<p>This is actively being worked on</p> <ul style="list-style-type: none"><li>InfluxUI-UG9 #41 Implementation of data source function</li><li>InfluxUI-UG9 #45 CORE 1: Data-source page</li><li>InfluxUI-UG9 #50 TEST: data source page</li></ul>	<p>This is your testing phase</p> <ul style="list-style-type: none"><li>InfluxUI-UG9 #49 Complete test the submitted code by team members and ensure the code quality and can be stable work</li></ul>	<p>This has been completed</p> <ul style="list-style-type: none"><li>InfluxUI-UG9 #1 User story 1: Drag-and-Drop Interface for Selecting Data Sources</li><li>InfluxUI-UG9 #36 User Story 5: Intuitive Main Interface for Data Analysis</li><li>InfluxUI-UG9 #35 User Story 4: Secure User Login</li><li>InfluxUI-UG9 #3 User Story 3: Automatic Query Generation and Execution</li></ul>
<p>+ Add item</p>		

## 2.2 User Stories

### User Story 8: Backend setup #46

 Open a1809851 opened this issue last week · 0 comments



a1809851 (Haotan Wang) commented last week

...

As a user, I want to create a page that displays a list of buckets, allowing users to click on the bucket they want to query

Role: Developer

Pre-condition

Users can click on the page to find the bucket they want to query.

Users after clicking on the bucket, the measurements and fields within the bucket will be displayed, and users can drag them into the selection area.

Main Flow:

The users can find the search area on the main page and enter relevant information.

The users interact with the search system and the system records relevant information.

The users search histories in the search bar will be retained and recorded by the data center for the next search.

---

## Definition of Done

The standard for done a coding task is: The code has been written according to the standards in the report, reviewed, tested, modified by the team, passed peer review, and received unanimous recognition from all team members.

The standard for done a non-coding task is: The relevant tasks have been discussed, decided upon, and implemented during the group meeting, and all team members have reached a consensus, task will start shortly.

# Summary of Changes

---

This week, we made two significant changes to the website. The first change is that users can now log in to their InfluxDB account through the website. Users must first create an account and an organization on the InfluxDB web app. Then, using their accounts API token, organization name and the URL of the InfluxDB web app, they can connect to their account using our website.

The second change made is that users can access buckets created on their InfluxDB account from our website. Once logged in, users can drag-and-drop buckets along with their associated measurements and fields into a box.

Currently, we are designing functionality for the selected data to be filtered and finally for a Flux query to be generated using the data and filters selected.



---

# No-Code Solution for InfluxDB Snapshot 5.2

---

## Week 10

---

### Influx UI UG 9

Tyler Chapman (a1851834)

Li Chen (a1809775)

Yizhou Chen (a1761339)

Siqi Gu (a1868071)

Bradley Hill (a1704585)

Xin Li (a1868279)

Haotan Wang (a1809851)

Zhihan Yang (a1791800)

Yuzhe Zhang (a1809783)

### Influx UI UG 9

Tyler Chapman (a1851834)

Li Chen (a1809775)

Yizhou Chen (a1761339)

Siqi Gu (a1868071)

Bradley Hill (a1704585)

Xin Li (a1868279)

Haotan Wang (a1809851)

Zhihan Yang (a1791800)

---

Yuzhe Zhang (a1809783)

# Product Backlog and Task Board

## 1.1 Product Backlog

This screenshot shows a product backlog interface with the following details:

- Backlog** 5 / 5
- User Story 6: Data Source Display**  
InfluxUI-UG9 #42
- User Story 10: Dynamic Threshold Alerts in No-Code Interface Goal:**  
InfluxUI-UG9 #57
- User Story 5: Automated Query Suggestions Based on Selection History**  
InfluxUI-UG9 #56

## 1.2 Task Board

<input type="checkbox"/>	<b>User Story 10: Dynamic Threshold Alerts in No-Code Interface Goal:</b> #57 opened 4 minutes ago by a1809775	
<input type="checkbox"/>	<b>User Story 5: Automated Query Suggestions Based on Selection History</b> #56 opened 8 minutes ago by a1809775	
<input type="checkbox"/>	<b>2.4 Constructing a Query</b> #55 opened 3 days ago by a1791800  1 of 3 tasks	
<input type="checkbox"/>	<b>2.3 Allow Attachments</b> #54 opened 3 days ago by a1791800  3 tasks done	
<input type="checkbox"/>	<b>2.2 Filter Modules</b> #53 opened 4 days ago by a1791800  9 tasks done	
<input type="checkbox"/>	<b>2.1 Select Region</b> #52 opened 5 days ago by a1791800  4 tasks done	
<input type="checkbox"/>	<b>CORE 2: Query Builder</b> #51 opened 5 days ago by a1791800  6 of 9 tasks	
<input type="checkbox"/>	<b>TEST: data source page</b> #50 opened last week by a1791800	
<input type="checkbox"/>	<b>User story 9: Technique connect</b> #48 opened last week by a1809851	
<input type="checkbox"/>	<b>About the login function</b>	
	#47 opened 2 weeks ago by a1791800	

# Sprint Backlog and User Stories

## 2.1 Sprint Backlog

The image shows a digital sprint backlog board with three columns: In Progress, Testing, and Done.

- In Progress:** 1 item. Description: This is actively being worked on. Item: InfluxUI-UG9 #55, 2.4 Constructing a Query.
- Testing:** 4 items. Description: This is your testing phase. Items:
  - InfluxUI-UG9 #49, Complete test the submitted code by team members and ensure the code quality and can be stable work.
  - InfluxUI-UG9 #52, 2.1 Select Region.
  - InfluxUI-UG9 #53, 2.2 Filter Modules.
  - InfluxUI-UG9 #54, 2.3 Allow Attachments.
- Done:** 34 items. Description: This has been completed. Items:
  - InfluxUI-UG9 #1, User story 1: Drag-and-Drop Interface for Selecting Data Sources.
  - InfluxUI-UG9 #36, User Story 5: Intuitive Main Interface for Data Analysis.
  - InfluxUI-UG9 #35, User Story 4: Secure User Login.
  - InfluxUI-UG9 #3, User Story 3: Automatic Query Generation and Execution.

Each card includes a plus icon to add more items.

## 2.2 User Stories

## User Story 11: Custom Time Range Selection for Real-Time Data Analysis #58

[Open](#)

a1809775 opened this issue now · 0 comments



a1809775 commented now

...

Goal:

As a user, I want to customize the time range of data displayed in real-time within the no-code interface, so I can focus on relevant periods and perform more precise analysis.

Actors:

User

Pre-conditions:

The user is logged into the no-code interface.

The user has access to time-series data stored in InfluxDB.

Main Flow:

The user selects the data source, bucket, and fields using drag-and-drop functionality.

The user accesses the time range selector in the interface and specifies a custom time range (e.g., last 24 hours, previous week, or a specific date range).

The interface updates and displays real-time data preview using Grafana based on the selected time range.

The user observes and analyzes the data in the specified time window, making any necessary adjustments to the time range or the fields being analyzed.

The user confirms the selected time range to proceed with further actions, such as visualization or exporting the data.

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Create a branch for this issue or link to an external issue

Notifications

[Unsubscribe](#)

You're receiving notifications because you're assigned to this issue

## Definition of Done

The standard for done a coding task is: The code has been written according to the standards in the report, reviewed, tested, modified by the team, passed peer review, and received unanimous recognition from all team members.

The standard for done a non-coding task is: The relevant tasks have been discussed, decided upon, and implemented during the group meeting, and all team members have reached a consensus, task will start shortly.

---

## Summary of Changes

This week, as this is the final snapshot of the project, the overall task of the team is to refine and improve the project. Through testing, the website has basically no minor issues such as jumping bugs or font errors.

At the same time, we have also optimized the login function so that users can link to their InfluxDB account more quickly after entering relevant information (such as API tokens).

Secondly, we have also improved the bucket access function, allowing users to more intuitively drag and drop buckets and their related measurement values and fields to designated areas, making it easier to manage and use their data.