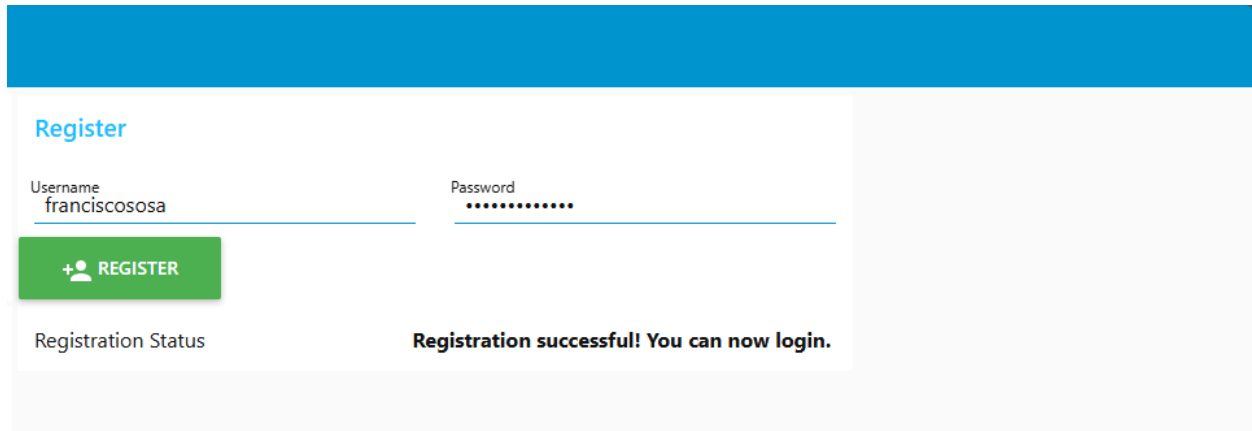


Screenshots 4th Industrial Revolution

Register:

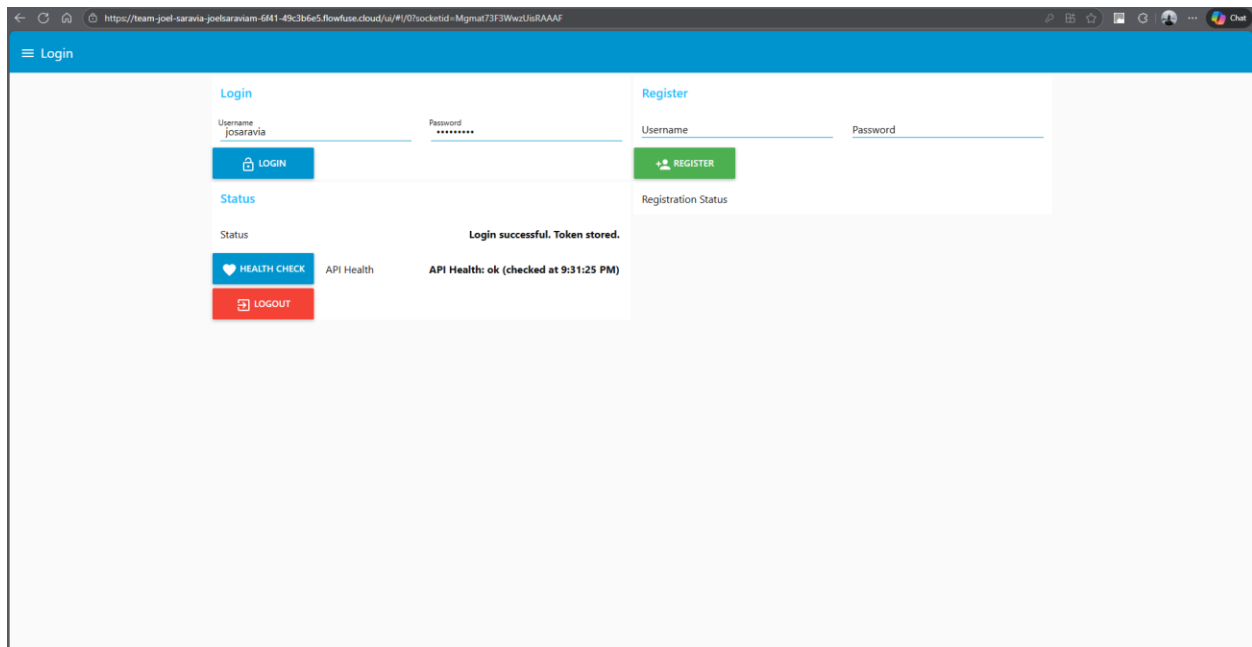
user: franciscososa

pwd: franciscososa



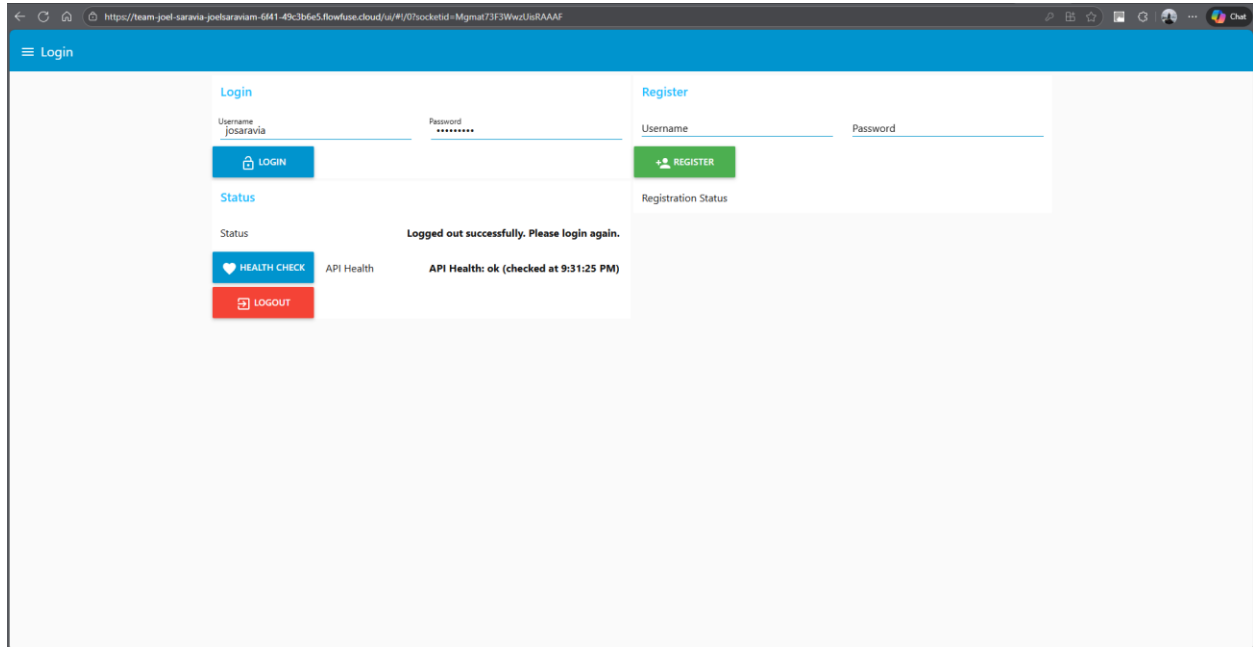
The screenshot shows a web interface with a blue header. Below the header, there is a 'Register' section. It contains two input fields: 'Username' with the value 'franciscososa' and 'Password' with masked characters. Below these fields is a green button with a plus icon and the text 'REGISTER'. Underneath the button, the text 'Registration Status' is followed by the message 'Registration successful! You can now login.'

Log in



The screenshot shows a web browser window with a URL starting with 'https://team-josel-saravia-josel-saravia-6641-49c3b5e5.flowfuse.cloud/'. The page has a blue header with a 'Login' button. Below the header, there are two main sections: 'Login' and 'Register'. The 'Login' section has input fields for 'Username' (value: 'josaravia') and 'Password' (masked), a blue 'LOGIN' button, and a 'Status' section showing 'Login successful. Token stored.' Below this is a 'HEALTH CHECK' button and 'API Health' status 'API Health: ok (checked at 9:31:25 PM)'. There is also a red 'LOGOUT' button. The 'Register' section has input fields for 'Username' and 'Password', a green 'REGISTER' button, and a 'Registration Status' section.

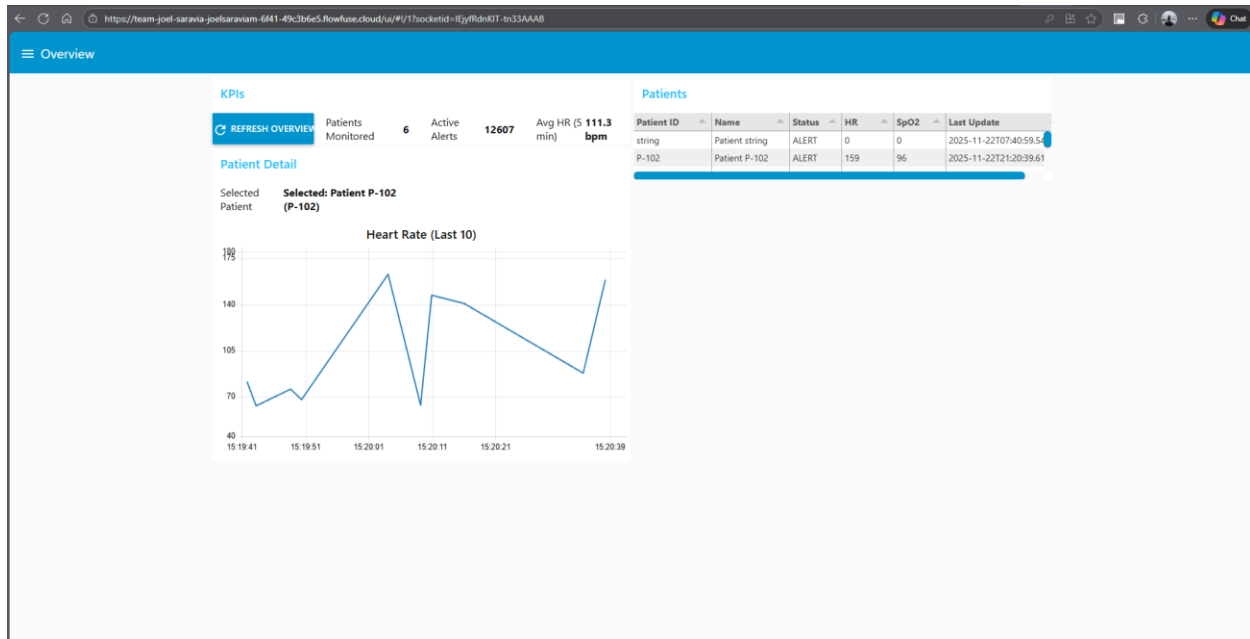
Log out



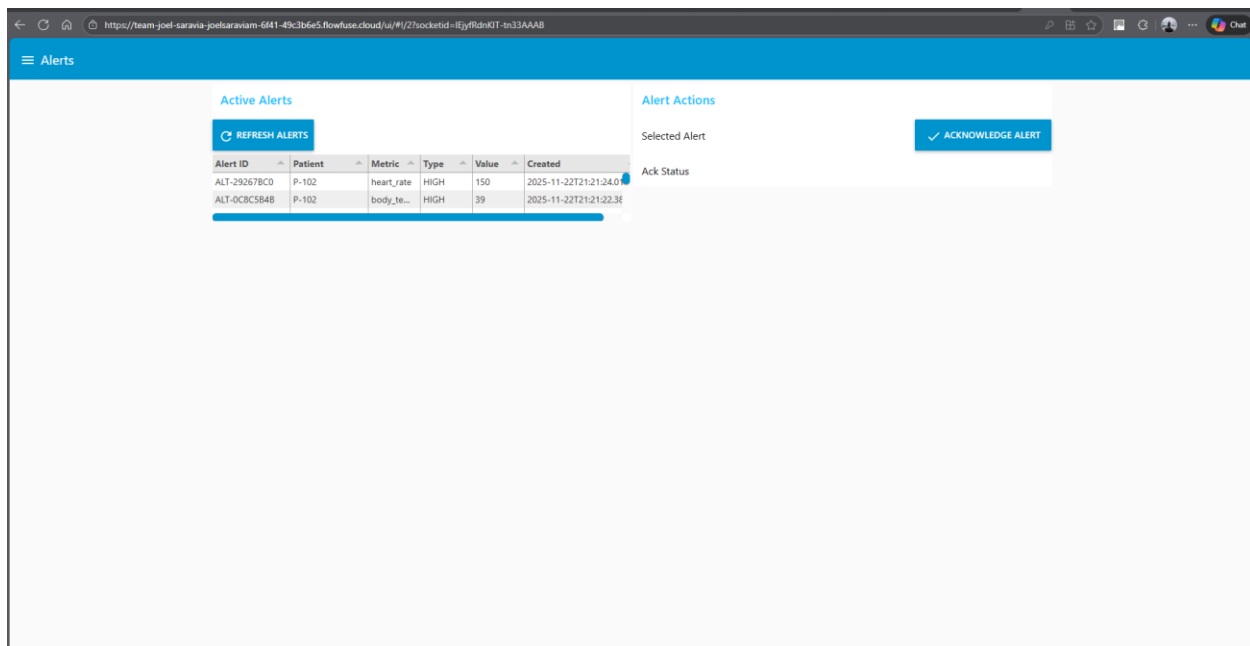
The screenshot shows a web application interface with a blue header bar containing a menu icon and the text "Login". The main content area is divided into three sections:

- Login:** Contains fields for "Username" (with the value "josaravia") and "Password" (masked with dots). Below these fields is a blue "LOGIN" button.
- Register:** Contains fields for "Username" and "Password". Below these fields is a green "REGISTER" button.
- Status:** Contains a "Status" label and a message: "Logged out successfully. Please login again." Below this message are two buttons: a blue "HEALTH CHECK" button and a red "LOGOUT" button. To the right of the "HEALTH CHECK" button, the text "API Health: ok (checked at 9:31:25 PM)" is displayed.

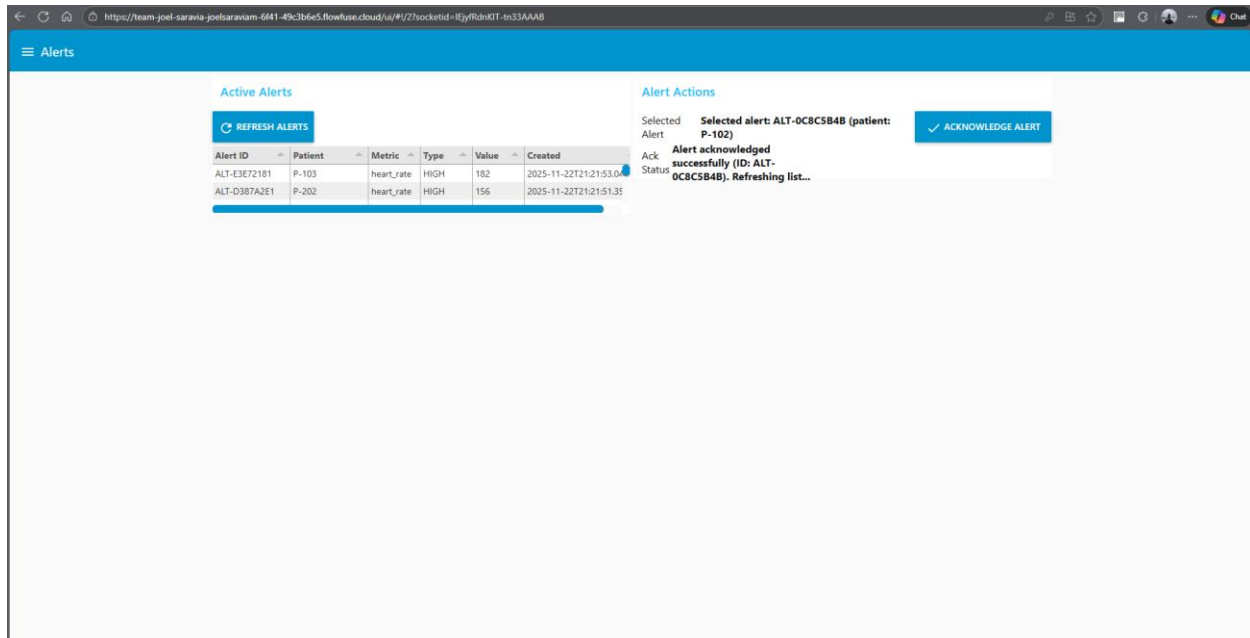
Visualizations - Overview



Visualizations – Alerts



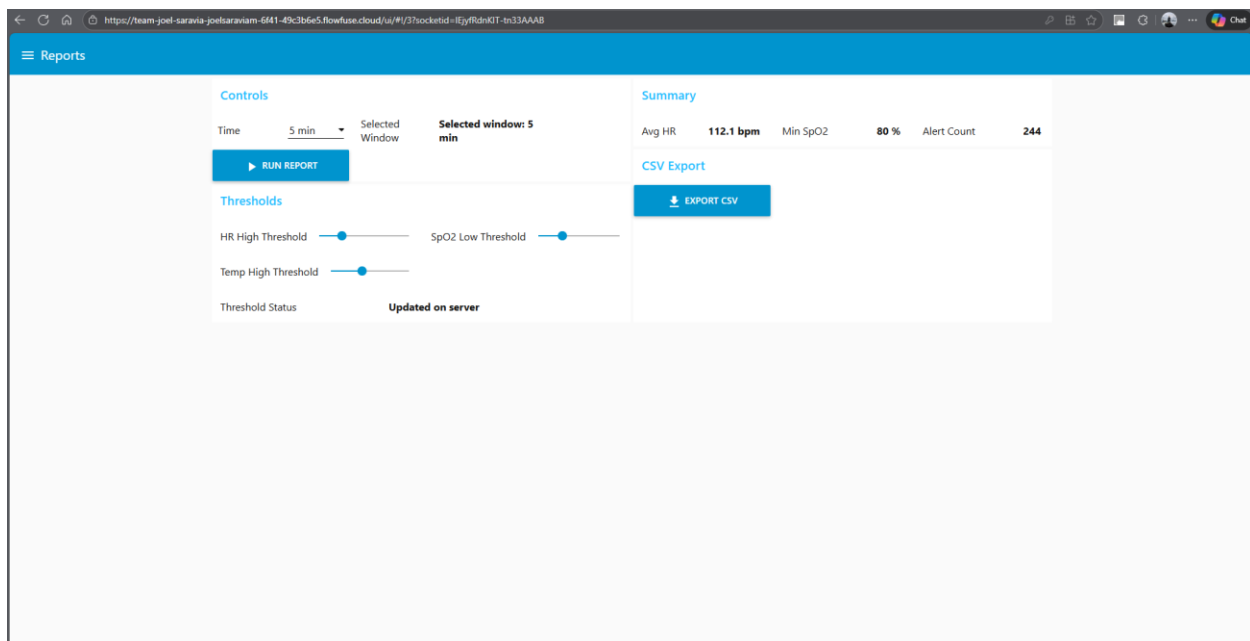
Success ack:



The screenshot shows the 'Alerts' dashboard. On the left, under 'Active Alerts', there is a table with columns: Alert ID, Patient, Metric, Type, Value, and Created. The table contains two rows of data. Above the table is a 'REFRESH ALERTS' button. On the right, under 'Alert Actions', there is a 'Selected alert: ALT-OC8C5B4B (patient: P-102)' and an 'ACKNOWLEDGE ALERT' button. Below this, the status is 'Alert acknowledged successfully (ID: ALT-OC8C5B4B). Refreshing list...'.

Alert ID	Patient	Metric	Type	Value	Created
ALT-E3E72181	P-103	heart_rate	HIGH	182	2025-11-22T21:21:53.0...
ALT-D387A2E1	P-202	heart_rate	HIGH	156	2025-11-22T21:21:51.3...

Visualizations - Reports



The screenshot shows the 'Reports' dashboard. On the left, under 'Controls', there is a 'Time' dropdown set to '5 min', a 'Selected Window' dropdown, and a 'Selected window: 5 min' label. Below these is a 'RUN REPORT' button. Under 'Thresholds', there are sliders for 'HR High Threshold', 'Temp High Threshold', and 'SpO2 Low Threshold'. A 'Threshold Status' label indicates 'Updated on server'. On the right, under 'Summary', there is a table with columns: Avg HR, Min SpO2, Alert Count, and a value of 244. Below this is a 'CSV Export' button with a download icon.

Avg HR	Min SpO2	Alert Count
112.1 bpm	80 %	244

← ↻ 🔍 https://team-joel-saravia-joelsaraviam-6641-49c3b6e5.flowfuse.cloud/ui/#/37/socketid=tpcv7h8CTgKvZhhIAAAH Chat

≡ Reports

Controls

Time 15 min Selected Window Selected window: 15 min

[▶ RUN REPORT](#)

Thresholds

HR High Threshold SpO2 Low Threshold

Temp High Threshold

Threshold Status Updated on server

Summary

Avg HR 108.8 bpm Min SpO2 80 % Alert Count 776

CSV Export

[EXPORT CSV](#)

← ↻ 🔍 https://team-joel-saravia-joelsaraviam-6641-49c3b6e5.flowfuse.cloud/ui/#/37/socketid=tpcv7h8CTgKvZhhIAAAH Chat

≡ Reports

Controls

Time 60 min Selected Window Selected window: 60 min

[▶ RUN REPORT](#)

Thresholds

HR High Threshold SpO2 Low Threshold

Temp High Threshold

Threshold Status Updated on server

Summary

Avg HR 108.1 bpm Min SpO2 80 % Alert Count 3316

CSV Export

[EXPORT CSV](#)

Controls

Time

5 min

15 min

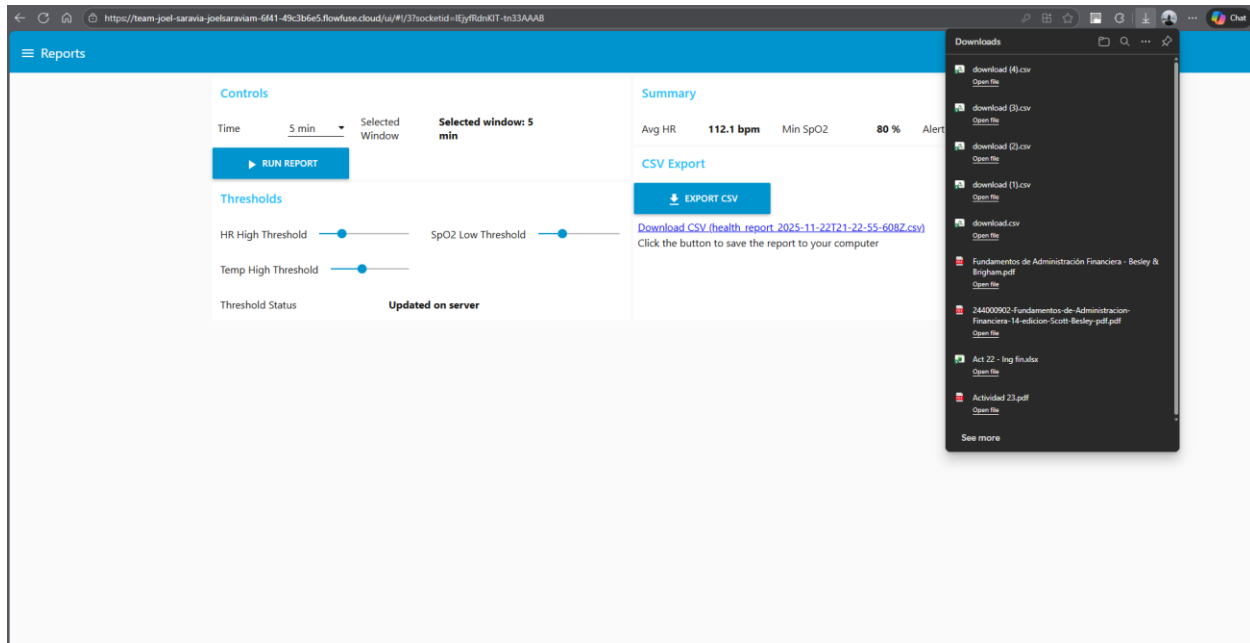
60 min

[▶ RUN REPORT](#)

Selected Window

Selected window: 60 min

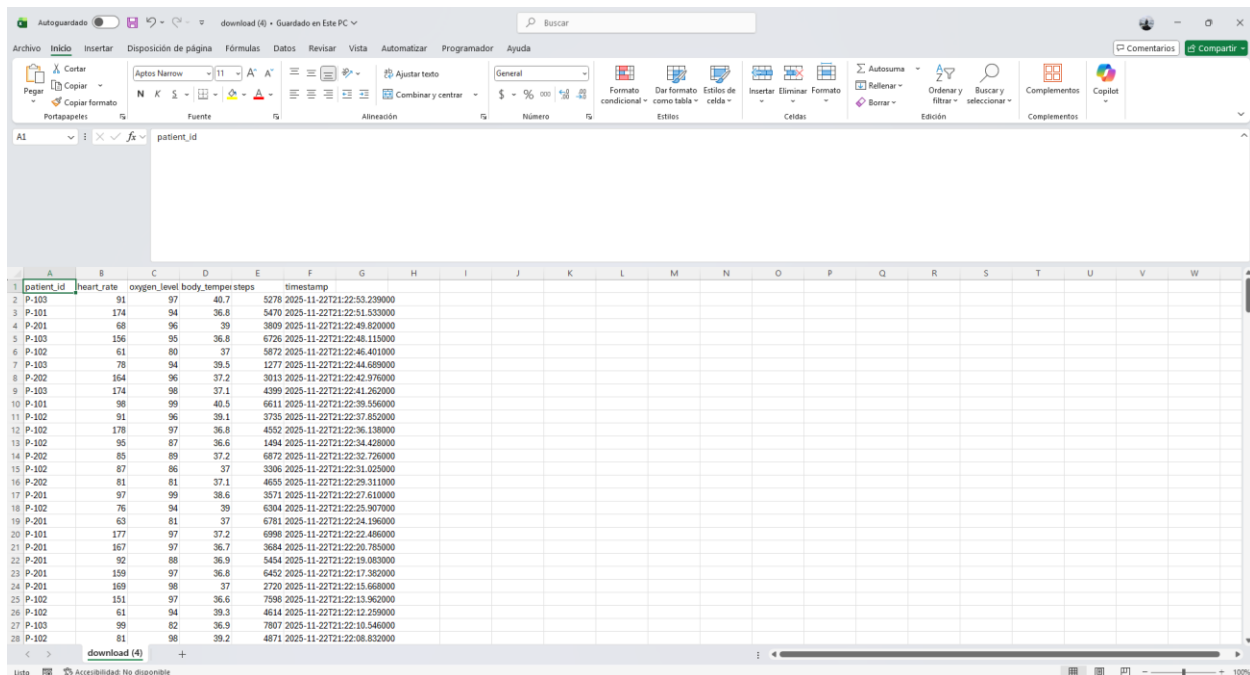
CSV Download:



The screenshot shows a web application interface for downloading a CSV report. The interface is divided into several sections:

- Reports**: A sidebar on the left with a 'Reports' menu item.
- Controls**: A section with a 'Time' dropdown set to '5 min', a 'Selected Window' dropdown set to '5 min', and a 'RUN REPORT' button.
- Thresholds**: A section with sliders for 'HR High Threshold', 'SpO2 Low Threshold', and 'Temp High Threshold'. Below the sliders is a 'Threshold Status' section indicating 'Updated on server'.
- Summary**: A section displaying 'Avg HR 112.1 bpm', 'Min SpO2 80 %', and an 'Alert' status.
- CSV Export**: A section with a 'Download CSV' button and a link to 'Download CSV (health_report_2025-11-22T21-22-55-608Z.csv)'. Below the link is a message: 'Click the button to save the report to your computer'.
- Downloads**: A sidebar on the right showing a list of downloaded files, including 'download (4).csv', 'download (3).csv', 'download (2).csv', 'download (1).csv', 'download.csv', and 'download CSV (health_report_2025-11-22T21-22-55-608Z.csv)'.

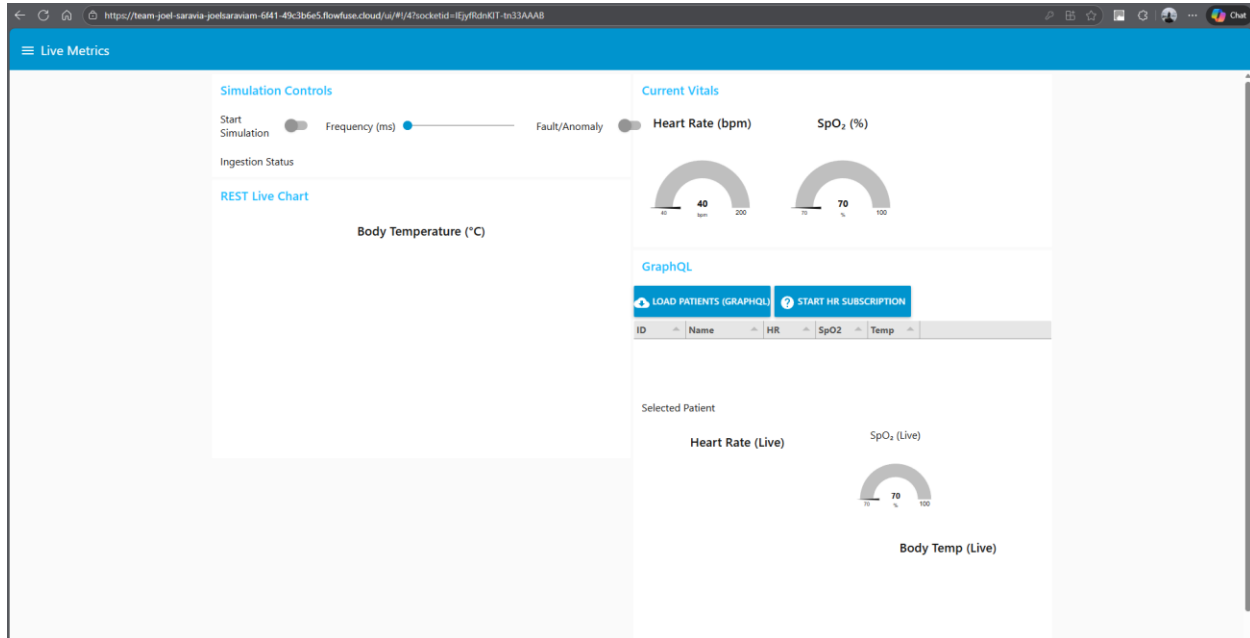
CSV (compatible with Excel)



patient_id	heart_rate	oxygen_level	body_temper	steps	timestamp
P-103	91	97	40.7	5278	2025-11-22T21:22:33.239000
P-101	174	94	36.8	5478	2025-11-22T21:22:31.533000
P-201	68	96	39	3809	2025-11-22T21:22:49.820000
P-103	156	95	36.8	6726	2025-11-22T21:22:48.115000
P-102	61	80	37	5872	2025-11-22T21:22:46.401000
P-103	78	94	39.5	1277	2025-11-22T21:22:44.689000
P-202	164	96	37.2	3013	2025-11-22T21:22:42.976000
P-103	174	96	37.1	4399	2025-11-22T21:22:41.262000
P-101	98	99	40.5	6611	2025-11-22T21:22:39.556000
P-102	91	96	39.1	3735	2025-11-22T21:22:37.852000
P-102	178	97	36.8	4552	2025-11-22T21:22:36.138000
P-102	95	87	36.6	1494	2025-11-22T21:22:34.428000
P-202	85	89	37.2	6872	2025-11-22T21:22:32.726000
P-102	87	86	37	3206	2025-11-22T21:22:31.025000
P-202	81	81	37.1	4655	2025-11-22T21:22:29.311000
P-201	97	99	38.6	3571	2025-11-22T21:22:27.610000
P-102	76	94	39	6304	2025-11-22T21:22:25.907000
P-201	63	81	37	6781	2025-11-22T21:22:24.196000
P-101	177	97	37.2	6990	2025-11-22T21:22:22.486000
P-201	167	97	36.7	3684	2025-11-22T21:22:20.785000
P-201	92	88	36.9	5454	2025-11-22T21:22:19.063000
P-201	159	97	36.8	6452	2025-11-22T21:22:17.382000
P-201	169	98	37	2720	2025-11-22T21:22:15.668000
P-102	151	97	36.6	7598	2025-11-22T21:22:13.962000
P-102	61	94	39.3	4614	2025-11-22T21:22:12.259000
P-103	99	82	36.9	7807	2025-11-22T21:22:10.546000
P-102	81	98	39.2	4871	2025-11-22T21:22:08.832000

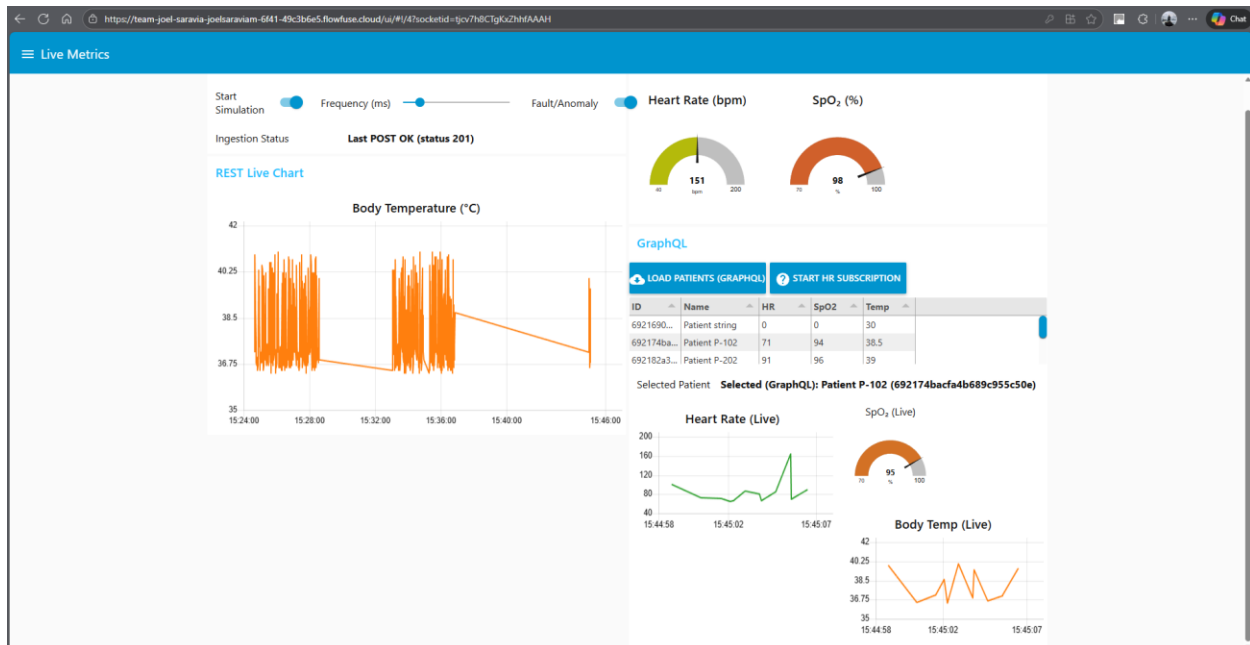
Visualizations – Live Metrics

Before clicking

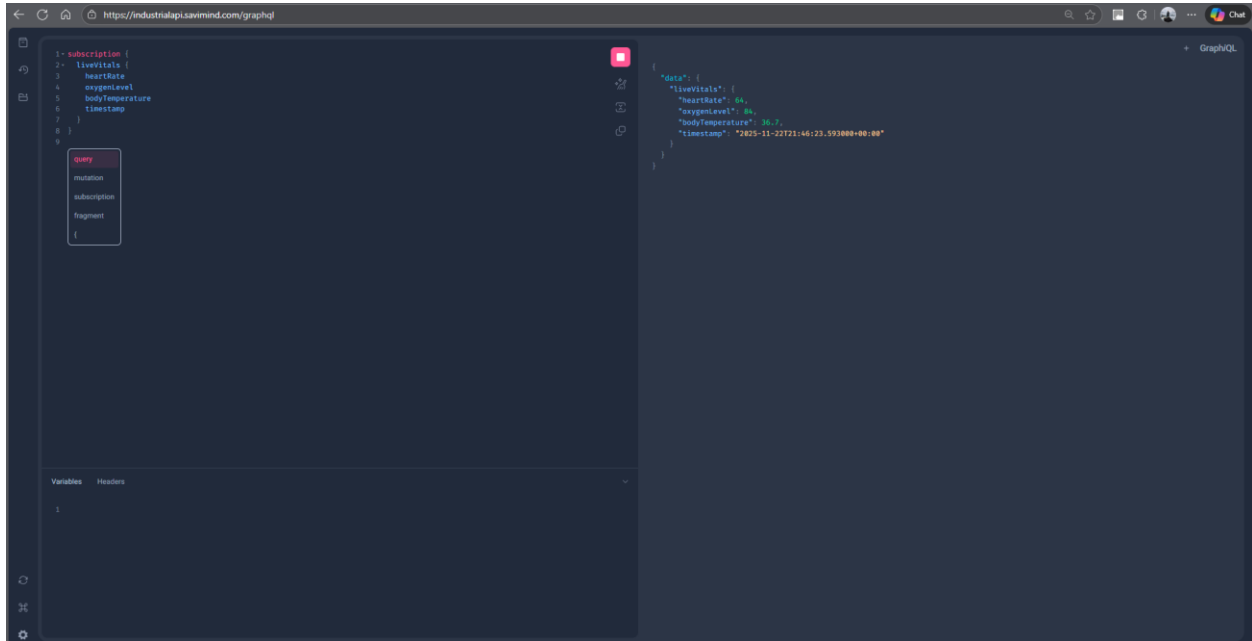


Visualizations – Live Metrics

After Click



GraphQL: wss

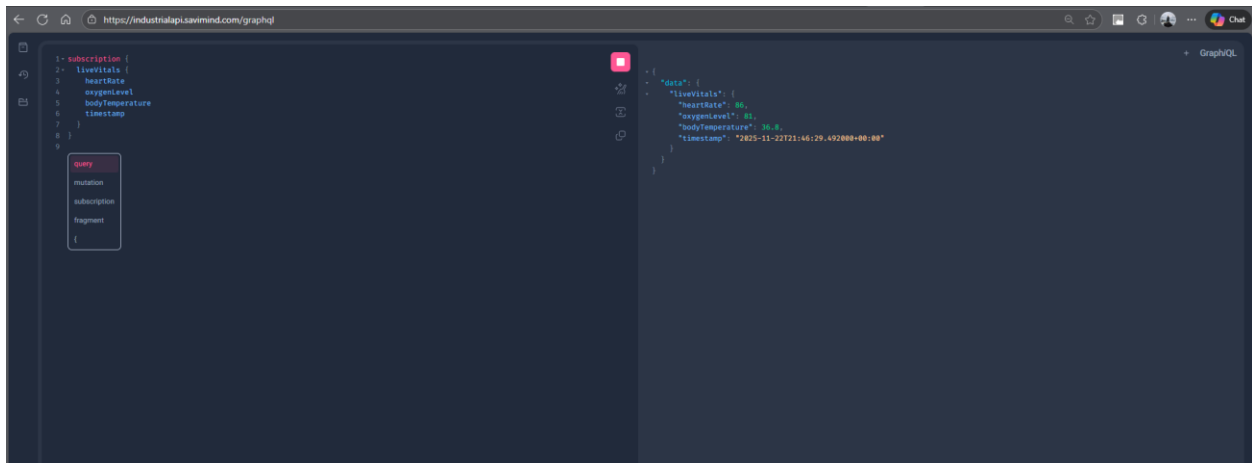


The screenshot shows the GraphQL Playground interface at <https://industrialapi.savimind.com/graphql>. On the left, a subscription query is defined:

```
1- subscription {
2-   liveVitals {
3-     heartRate
4-     oxygenLevel
5-     bodyTemperature
6-     timestamp
7-   }
8- }
```

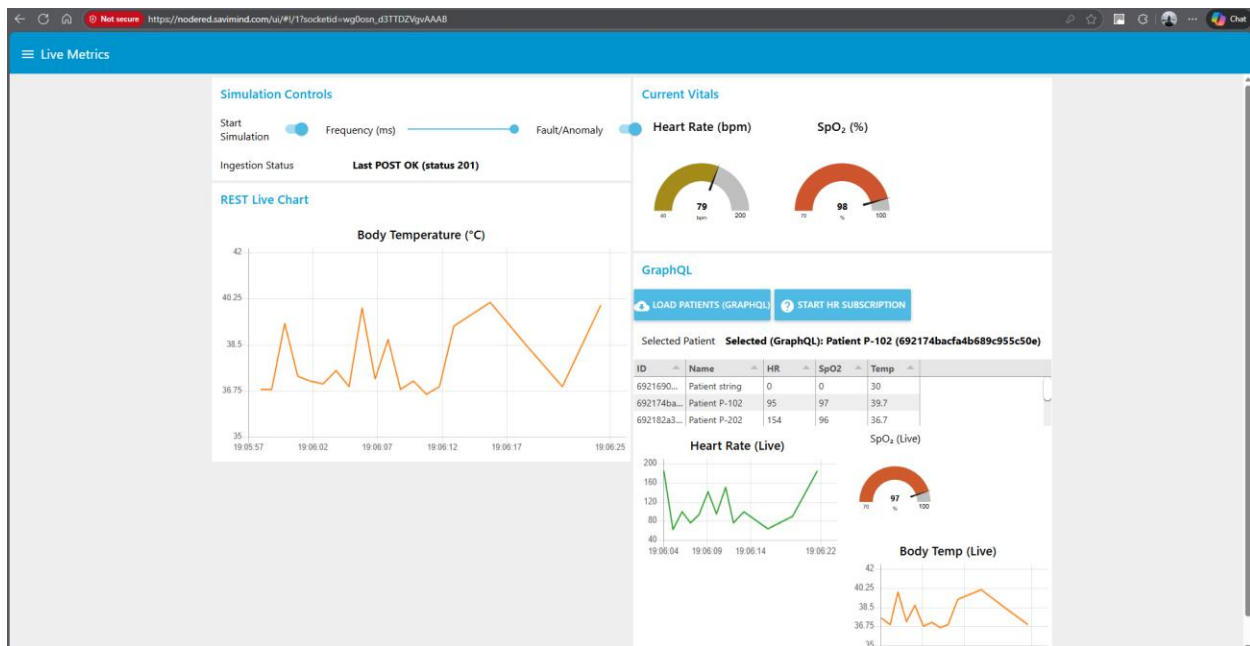
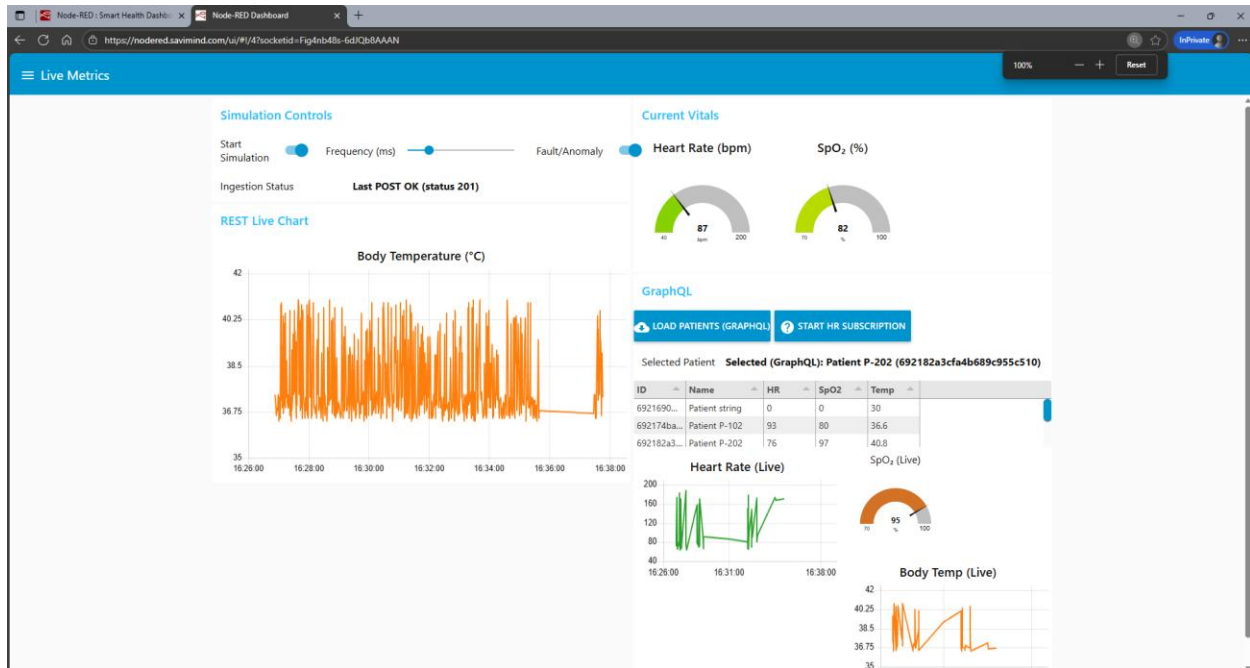
A dropdown menu on the left shows options: query, mutation, subscription, and fragment. The 'subscription' option is selected. Below the query editor, there are tabs for 'Variables' and 'Headers'. On the right, the JSON response of the subscription is displayed:

```
{
  "data": {
    "liveVitals": {
      "heartRate": 64,
      "oxygenLevel": 88,
      "bodyTemperature": 36.7,
      "timestamp": "2025-11-22T21:46:23.593000+00:00"
    }
  }
}
```



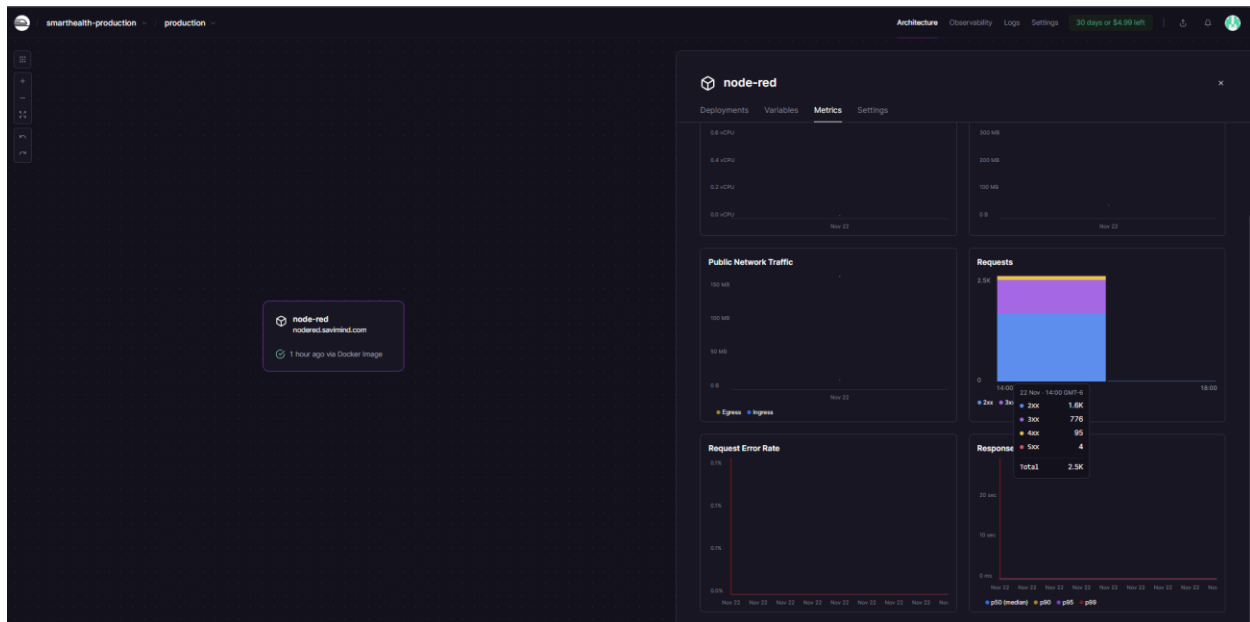
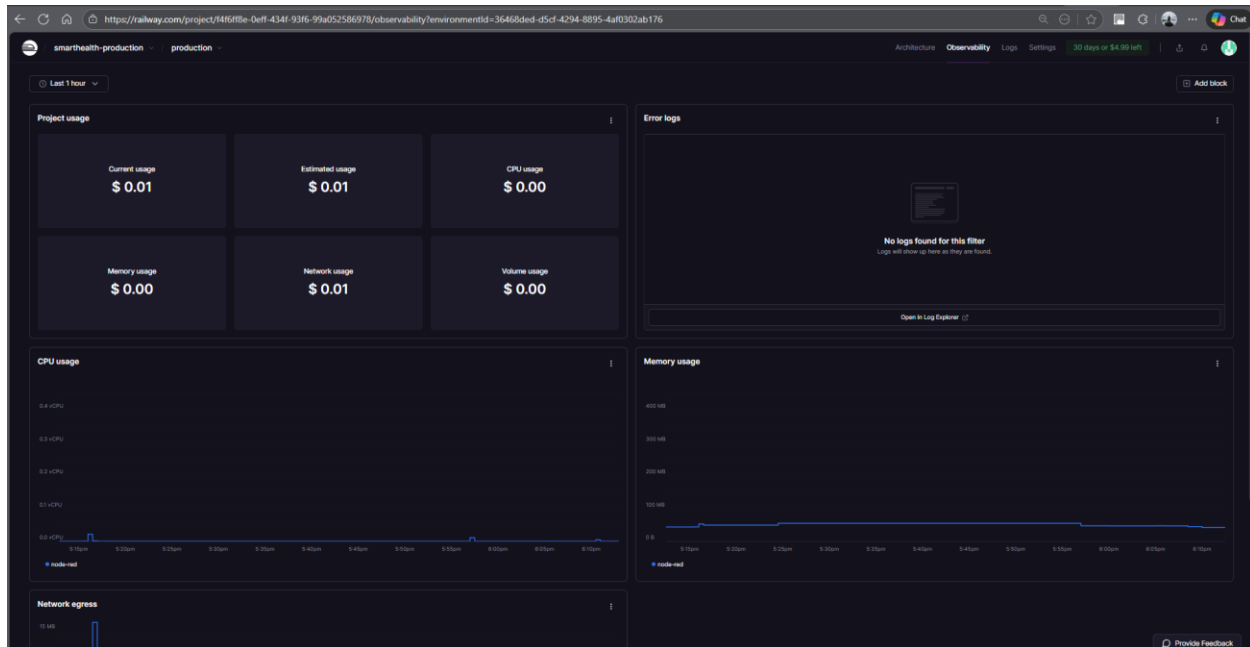
This screenshot is identical to the one above, showing the same GraphQL Playground interface with the subscription query and its JSON response.

Deployed with custom domain:

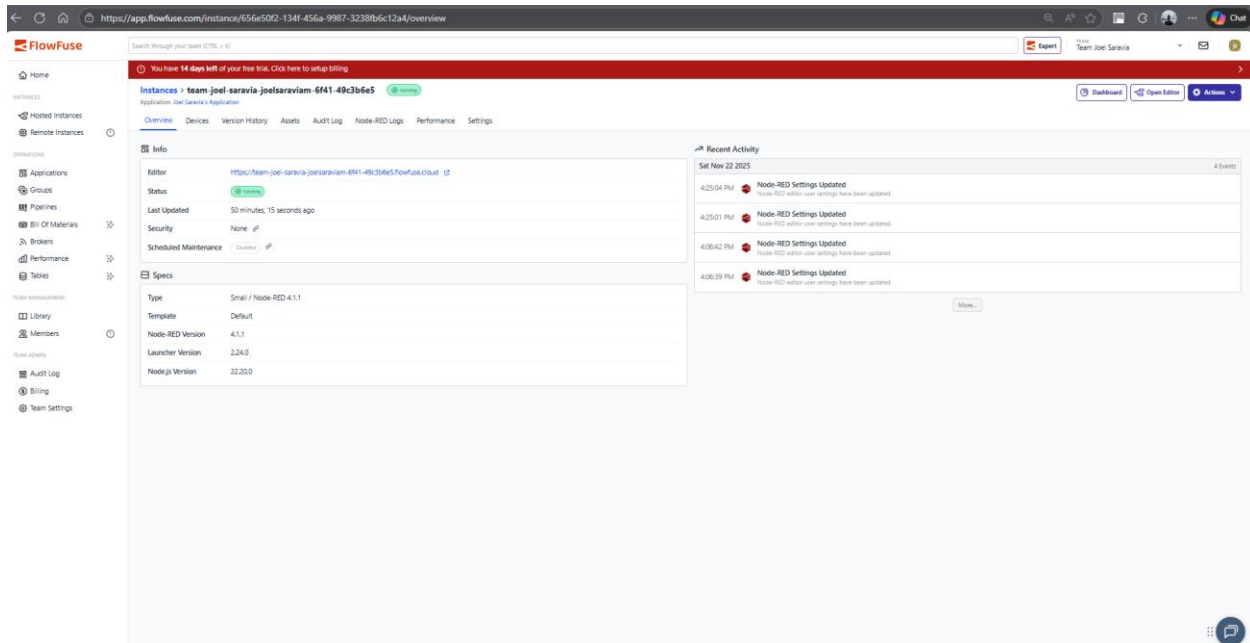


Deployment:

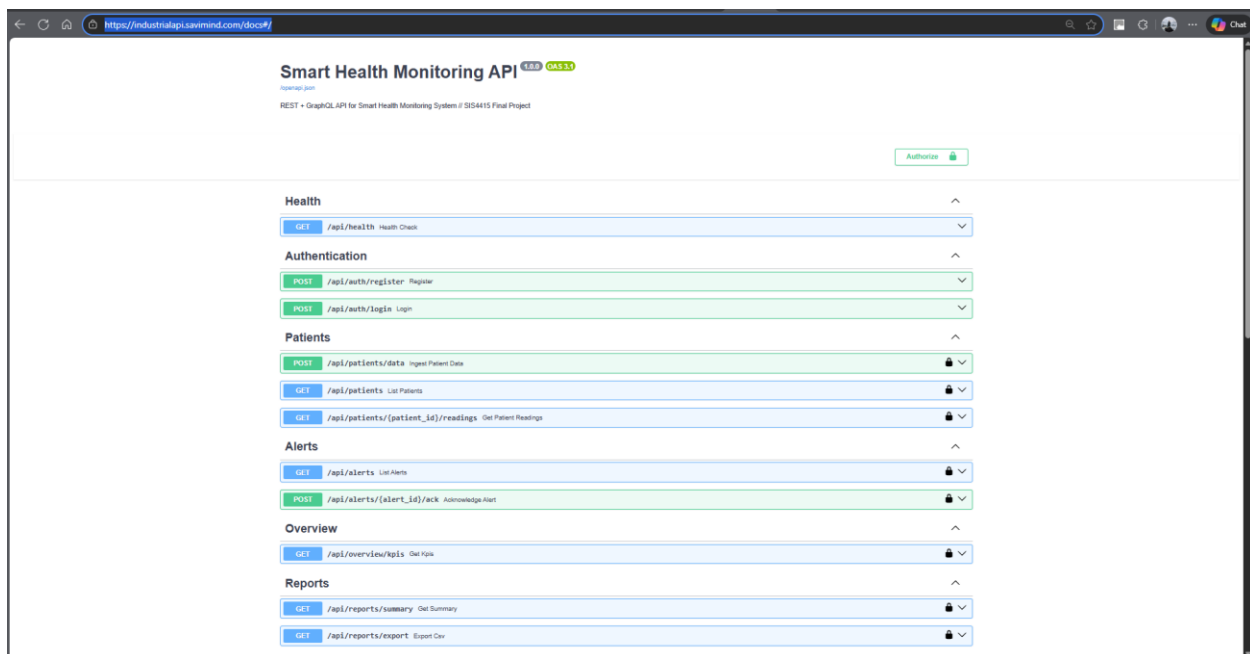
Railway for node-red Flow (docker container)



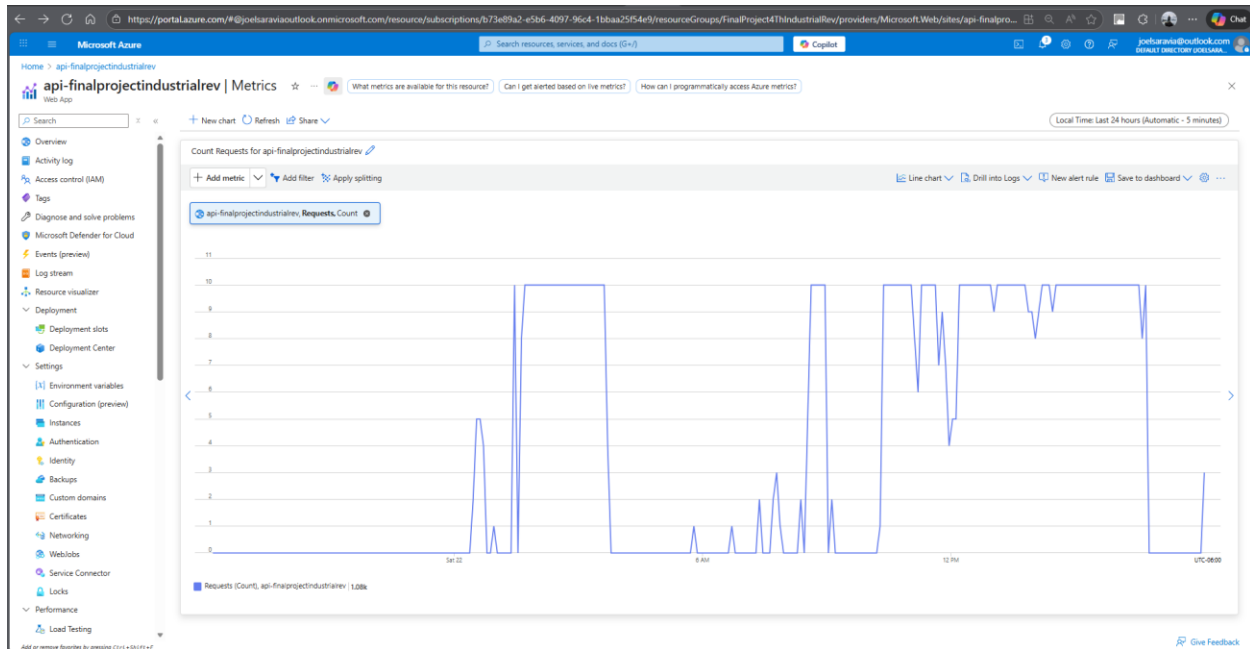
FlowFuse automatically deploys, just copied my JSON flow there:



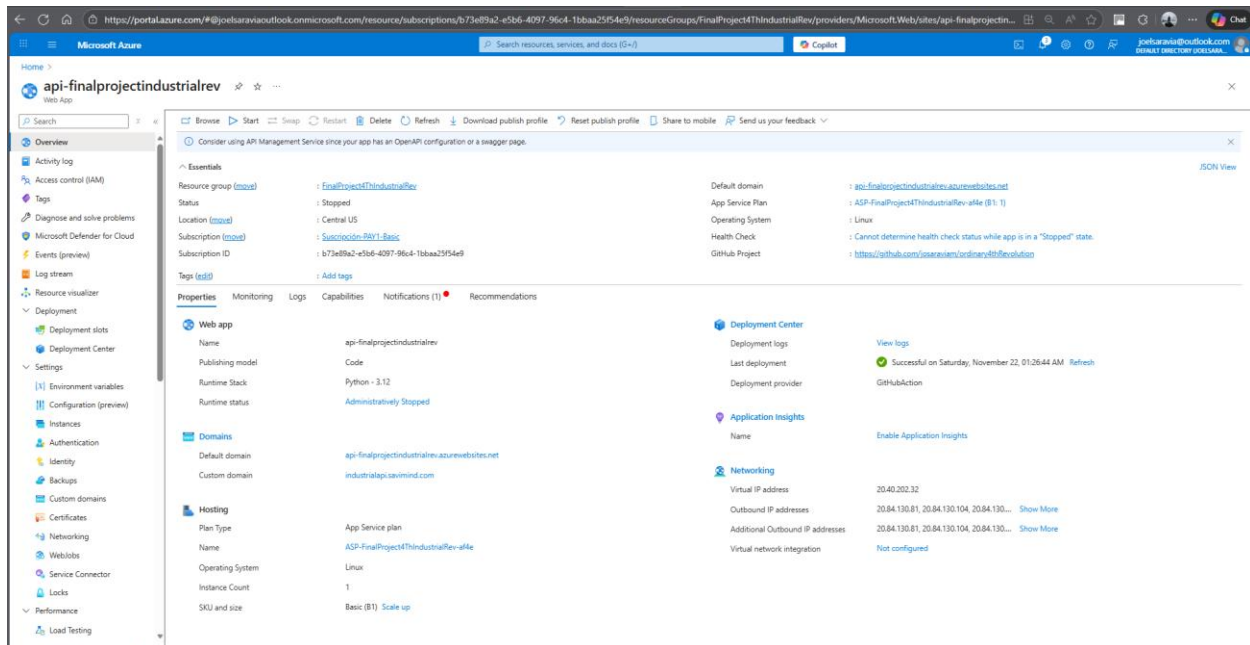
API Deployed using Azure:



Monitoring:

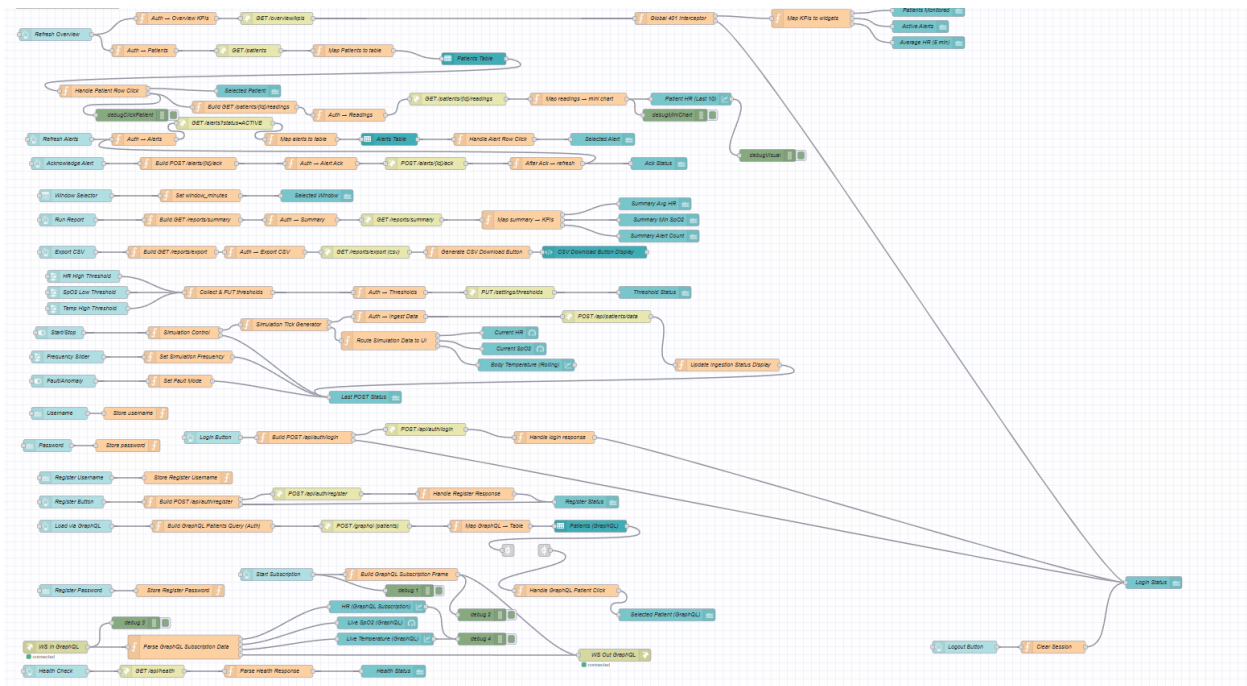


Setup - currently stopped-

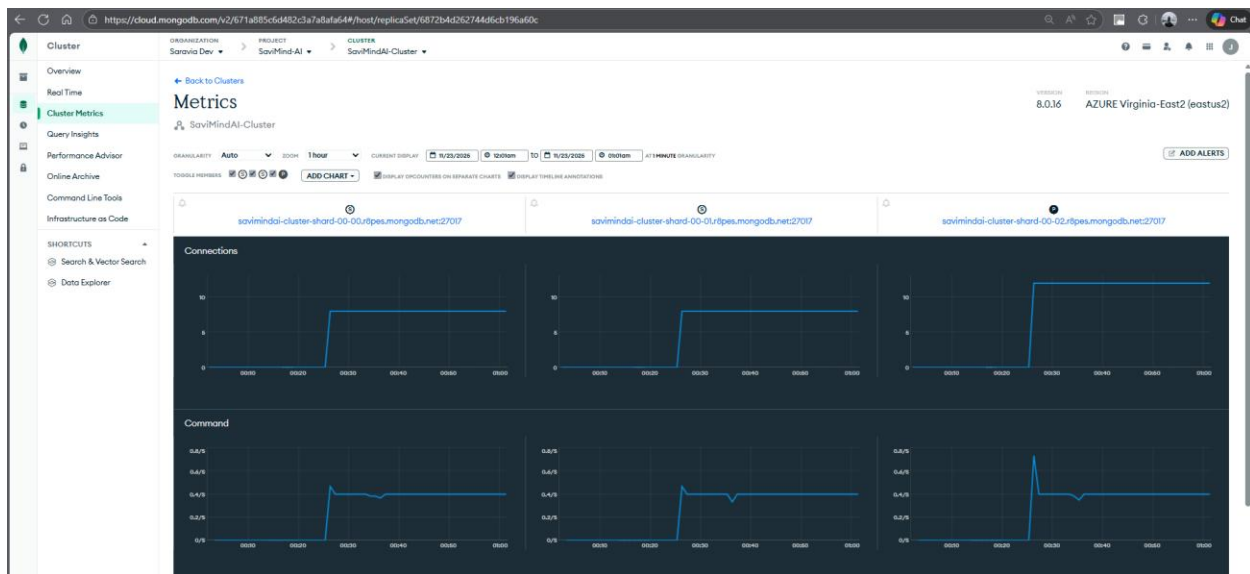


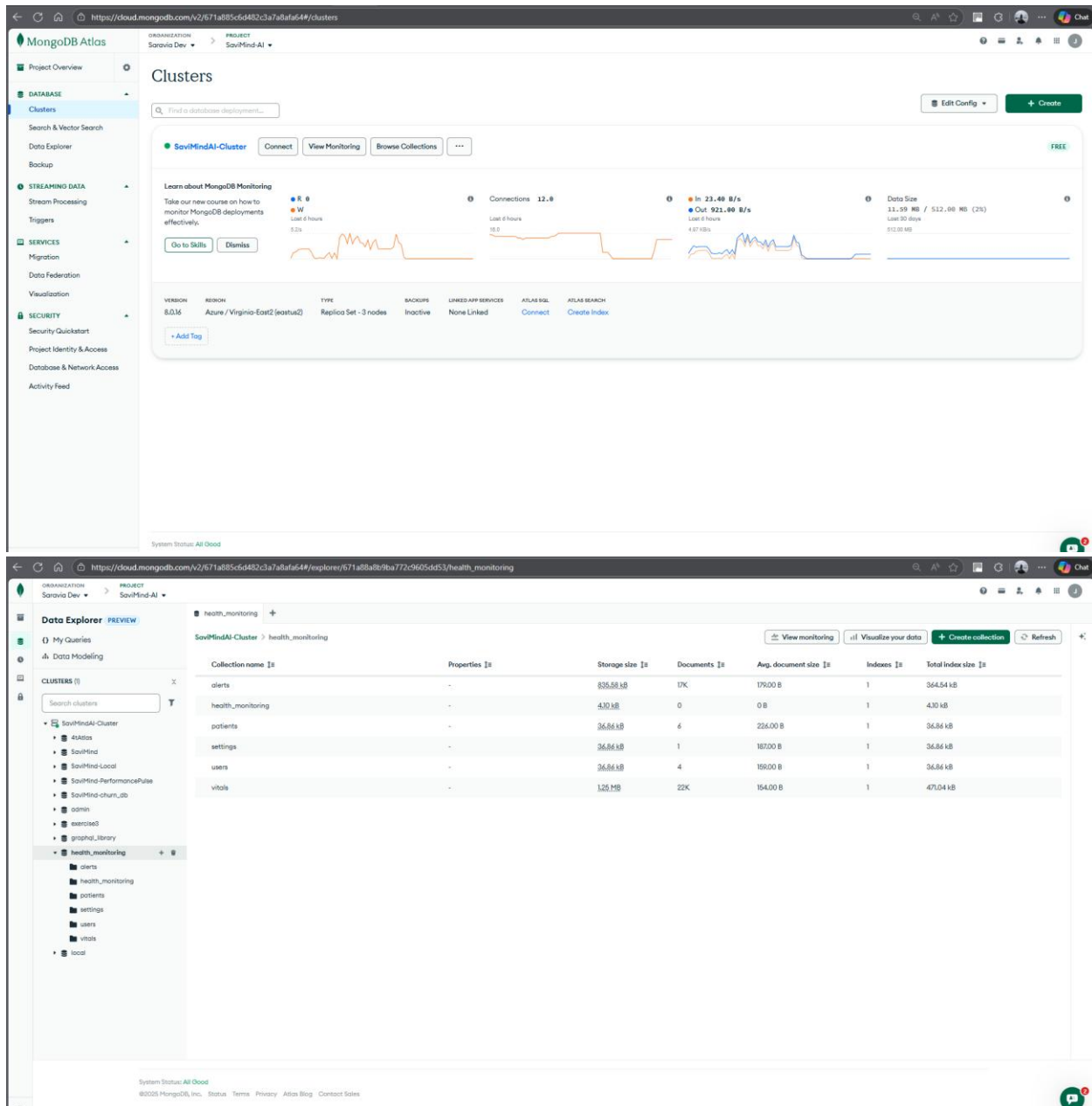
The screenshot shows the Azure portal interface for the 'api-finalprojectindustrialrev' Web App. The 'Overview' tab is selected, displaying various properties and settings. The 'Essentials' section shows the Resource group as 'FinalProject4ThIndustrialRev', Status as 'Stopped', Location as 'Central US', Subscription as 'Subscription: BAY1-Basic', and Subscription ID as 'b73e89a2-e5b6-4097-96c4-1bba2554e9'. The 'Properties' section shows the Web app name as 'api-finalprojectindustrialrev', Publishing model as 'Code', Runtime Stack as 'Python - 3.12', and Runtime status as 'Administratively Stopped'. The 'Deployment Center' section shows the Deployment logs as 'Successful on Saturday, November 22, 01:26:44 AM' and the Deployment provider as 'GitHubAction'. The 'Application Insights' section shows the Name as 'api-finalprojectindustrialrev' and the status as 'Enable Application Insights'. The 'Networking' section shows the Virtual IP address as '20.40.202.32', Outbound IP addresses as '20.84.130.81, 20.84.130.104, 20.84.130...', and Additional Outbound IP addresses as '20.84.130.81, 20.84.130.104, 20.84.130...'. The 'Hosting' section shows the Plan Type as 'App Service plan', Name as 'ASP-FinalProject4ThIndustrialRev-af4e', Operating System as 'Linux', Instance Count as '1', and SKU and size as 'Basic (B1) Scale up'.

Node-red Diagram:



Mongo Cluster:



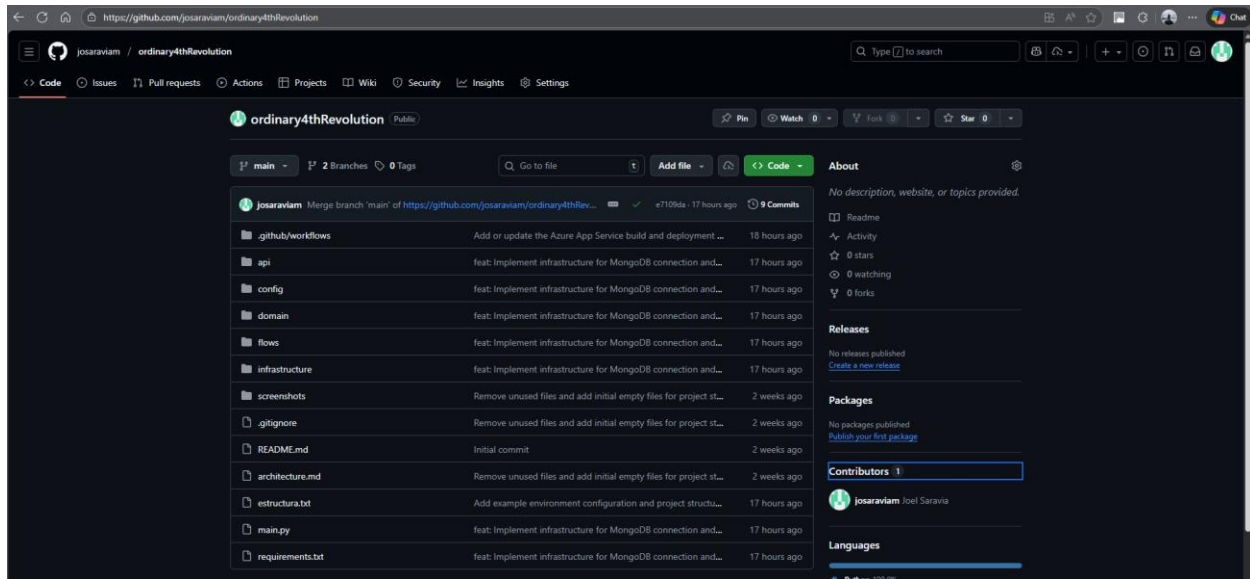


The image displays two screenshots of the MongoDB Atlas web interface. The top screenshot shows the 'Clusters' page for a project named 'SaviMind-AI'. It features a sidebar with navigation options like 'Project Overview', 'DATABASE', 'STREAMING DATA', 'SERVICES', and 'SECURITY'. The main content area shows the 'SaviMind-AI-Cluster' with various monitoring metrics and a table of cluster details.

The bottom screenshot shows the 'Data Explorer' page for the same project. It displays a table of collections within the 'health_monitoring' database. The table includes columns for Collection name, Properties, Storage size, Documents, Avg. document size, Indexes, and Total index size.

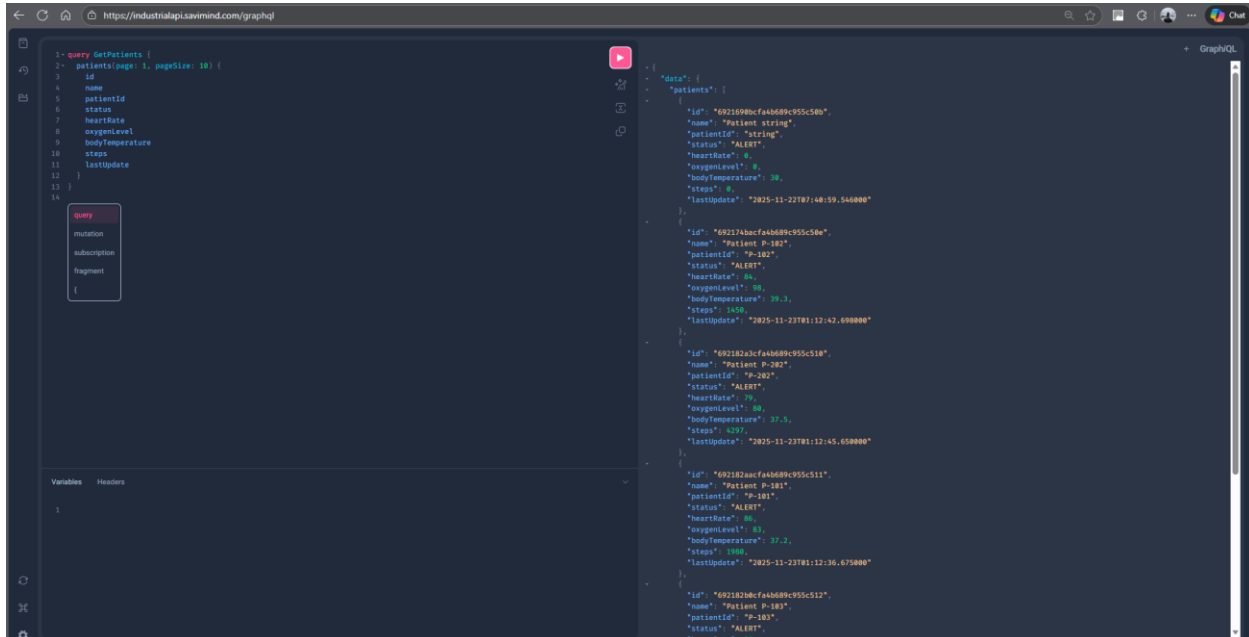
Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
alerts	-	835.58 kB	17K	179.00 B	1	364.54 kB
health_monitoring	-	4.30 kB	0	0 B	1	4.30 kB
patients	-	36.84 kB	6	226.00 B	1	36.84 kB
settings	-	36.84 kB	1	187.00 B	1	36.84 kB
users	-	36.84 kB	4	159.00 B	1	36.84 kB
vitals	-	125 MB	22K	154.00 B	1	47.04 kB

GitHub Repo:



GraphQL Queries

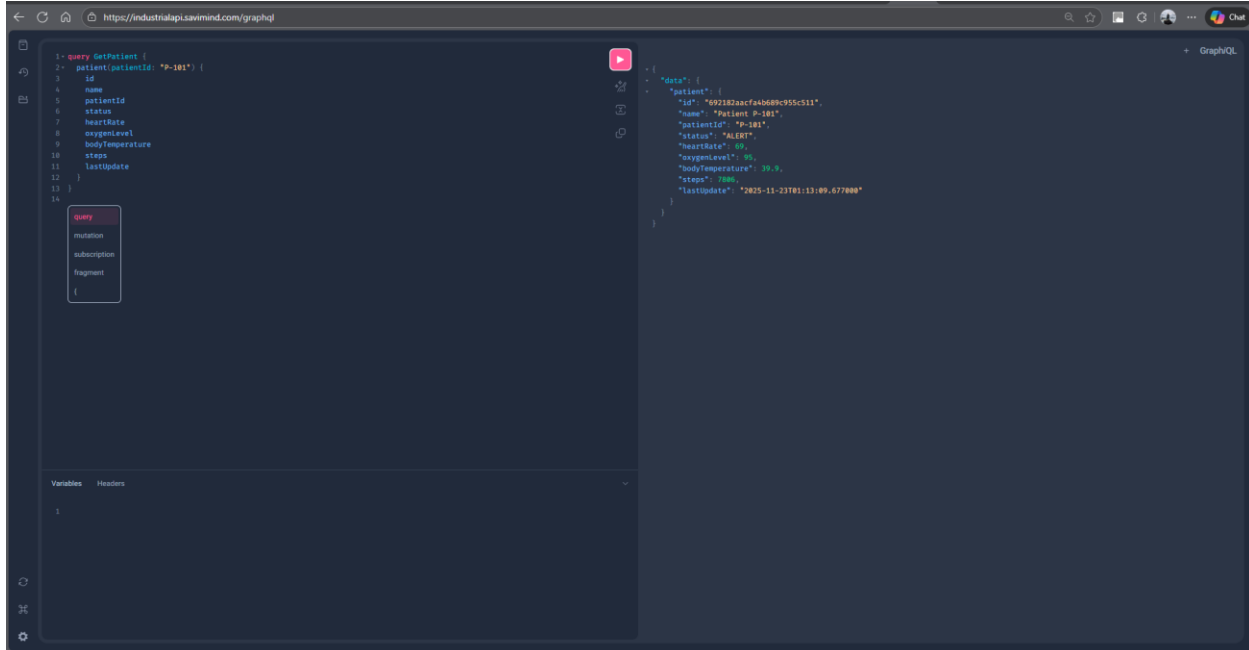
Get all patients



```
1 query GetPatients {
2   patients(page: 1, pageSize: 10) {
3     id
4     name
5     patientId
6     status
7     heartRate
8     oxygenLevel
9     bodyTemperature
10    steps
11    lastUpdate
12  }
13 }
14
```

```
{
  "data": {
    "patients": [
      {
        "id": "682168b6fa6b689c955c58a",
        "name": "Patient string",
        "patientId": "string",
        "status": "ALERT",
        "heartRate": 0,
        "oxygenLevel": 0,
        "bodyTemperature": 30,
        "steps": 0,
        "lastUpdate": "2025-11-22T07:48:59.546000"
      },
      {
        "id": "68217abacf4b6b689c955c58e",
        "name": "Patient P-182",
        "patientId": "P-182",
        "status": "ALERT",
        "heartRate": 88,
        "oxygenLevel": 88,
        "bodyTemperature": 39.3,
        "steps": 1458,
        "lastUpdate": "2025-11-23T01:12:43.698000"
      },
      {
        "id": "682182abcf4b6b689c955c510",
        "name": "Patient P-202",
        "patientId": "P-202",
        "status": "ALERT",
        "heartRate": 79,
        "oxygenLevel": 88,
        "bodyTemperature": 37.3,
        "steps": 4091,
        "lastUpdate": "2025-11-23T01:12:45.658000"
      },
      {
        "id": "682182aacf4b6b689c955c511",
        "name": "Patient P-181",
        "patientId": "P-181",
        "status": "ALERT",
        "heartRate": 88,
        "oxygenLevel": 88,
        "bodyTemperature": 37.2,
        "steps": 1068,
        "lastUpdate": "2025-11-23T01:12:36.675000"
      },
      {
        "id": "6821821b6cf4b6b689c955c512",
        "name": "Patient P-183",
        "patientId": "P-183",
        "status": "ALERT",
        "heartRate": 88,
        "oxygenLevel": 88,
        "bodyTemperature": 37.2,
        "steps": 1068,
        "lastUpdate": "2025-11-23T01:12:36.675000"
      }
    ]
  }
}
```

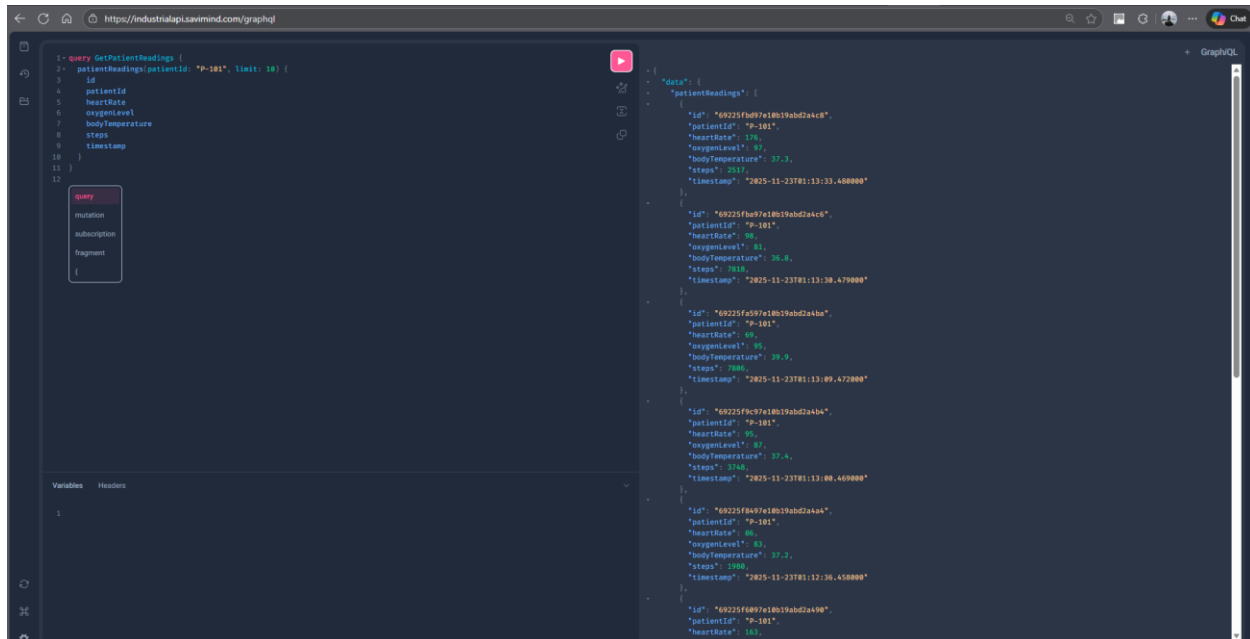
Get a Patient



```
1 query GetPatient {
2   patient(patientId: "P-181") {
3     id
4     name
5     patientId
6     status
7     heartRate
8     oxygenLevel
9     bodyTemperature
10    steps
11    lastUpdate
12  }
13 }
14
```

```
{
  "data": {
    "patient": {
      "id": "682182aacf4b6b689c955c511",
      "name": "Patient P-181",
      "patientId": "P-181",
      "status": "ALERT",
      "heartRate": 88,
      "oxygenLevel": 88,
      "bodyTemperature": 39.9,
      "steps": 7886,
      "lastUpdate": "2025-11-23T01:13:09.677000"
    }
  }
}
```


Patient Readings

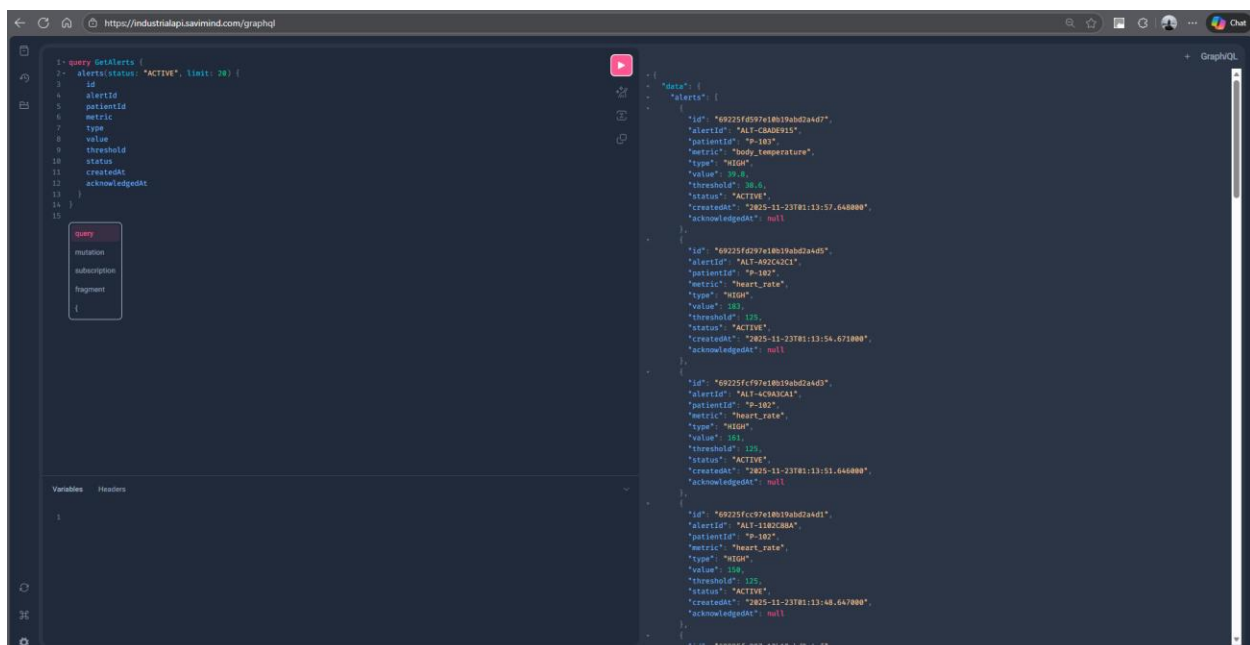


```
1 query GetPatientReadings {
2   patientReadings(patientId: "P-101", limit: 10) {
3     id
4     patientId
5     heartRate
6     oxygenLevel
7     bodyTemperature
8     steps
9     timestamp
10  }
11 }
12
```

```
{
  "data": {
    "patientReadings": [
      {
        "id": "68025f6b7e10b19abd2a4c6",
        "patientId": "P-101",
        "heartRate": 176,
        "oxygenLevel": 95,
        "bodyTemperature": 37.3,
        "steps": 2033,
        "timestamp": "2025-11-23T01:13:33.400000"
      },
      {
        "id": "68025f6b7e10b19abd2a4c6",
        "patientId": "P-101",
        "heartRate": 96,
        "oxygenLevel": 91,
        "bodyTemperature": 36.8,
        "steps": 1038,
        "timestamp": "2025-11-23T01:13:30.470000"
      },
      {
        "id": "68025f6b7e10b19abd2a4c6",
        "patientId": "P-101",
        "heartRate": 65,
        "oxygenLevel": 95,
        "bodyTemperature": 36.9,
        "steps": 1066,
        "timestamp": "2025-11-23T01:13:09.470000"
      },
      {
        "id": "68025f6b7e10b19abd2a4c6",
        "patientId": "P-101",
        "heartRate": 95,
        "oxygenLevel": 87,
        "bodyTemperature": 37.4,
        "steps": 2145,
        "timestamp": "2025-11-23T01:13:00.450000"
      },
      {
        "id": "68025f6b7e10b19abd2a4c6",
        "patientId": "P-101",
        "heartRate": 86,
        "oxygenLevel": 83,
        "bodyTemperature": 37.2,
        "steps": 1066,
        "timestamp": "2025-11-23T01:12:36.450000"
      },
      {
        "id": "68025f6b7e10b19abd2a490",
        "patientId": "P-101",
        "heartRate": 163,

```

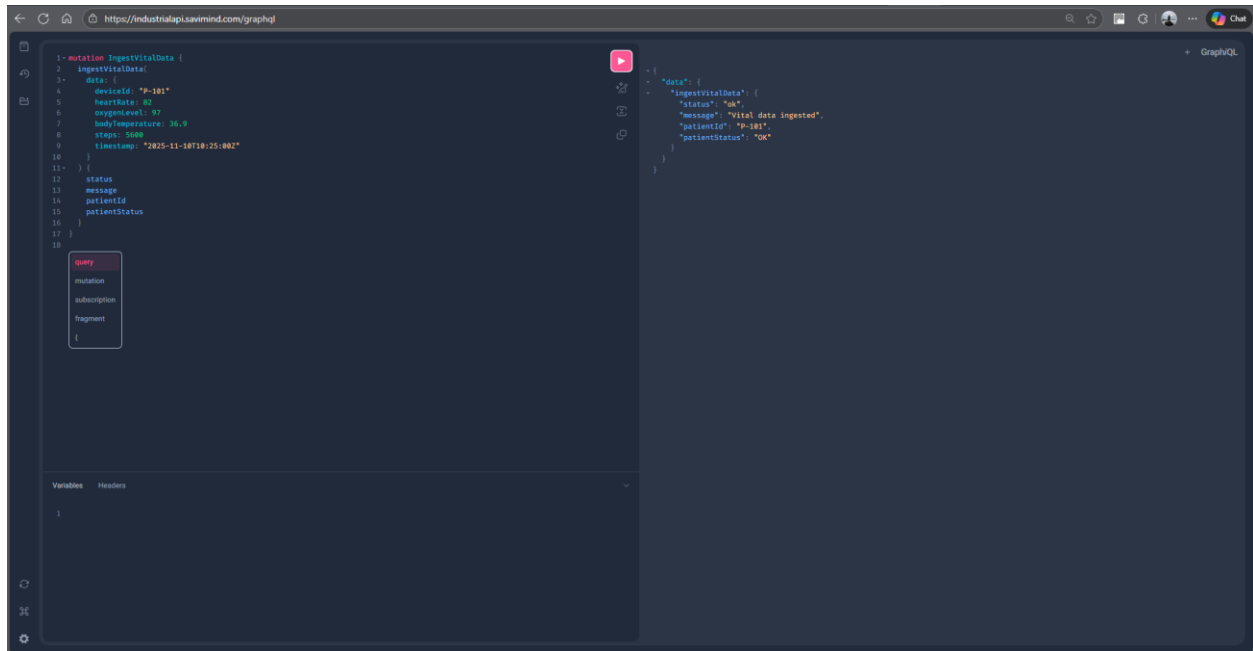
GetAlerts



```
1 query GetAlerts {
2   alerts(status: "ACTIVE", limit: 20) {
3     id
4     alertId
5     patientId
6     metric
7     type
8     value
9     threshold
10    status
11    createdAt
12    acknowledged
13  }
14 }
15
```

```
{
  "data": {
    "alerts": [
      {
        "id": "68025fd97e10b19abd2a4d7",
        "alertId": "ALT-CBADE915",
        "patientId": "P-101",
        "metric": "bodyTemperature",
        "type": "HIGH",
        "value": 38.8,
        "threshold": 38.5,
        "status": "ACTIVE",
        "createdAt": "2025-11-23T01:13:57.640000",
        "acknowledged": null
      },
      {
        "id": "68025fd97e10b19abd2a4d5",
        "alertId": "ALT-892C42C1",
        "patientId": "P-102",
        "metric": "heart_rate",
        "type": "HIGH",
        "value": 183,
        "threshold": 175,
        "status": "ACTIVE",
        "createdAt": "2025-11-23T01:13:54.671000",
        "acknowledged": null
      },
      {
        "id": "68025fd97e10b19abd2a4d3",
        "alertId": "ALT-4C9A3CA1",
        "patientId": "P-102",
        "metric": "heart_rate",
        "type": "HIGH",
        "value": 161,
        "threshold": 155,
        "status": "ACTIVE",
        "createdAt": "2025-11-23T01:13:51.640000",
        "acknowledged": null
      },
      {
        "id": "68025fcc7e10b19abd2a4d1",
        "alertId": "ALT-1182CBA",
        "patientId": "P-102",
        "metric": "heart_rate",
        "type": "HIGH",
        "value": 159,
        "threshold": 155,
        "status": "ACTIVE",
        "createdAt": "2025-11-23T01:13:48.647000",
        "acknowledged": null
      }
    ]
  }
}
```


Ingestion

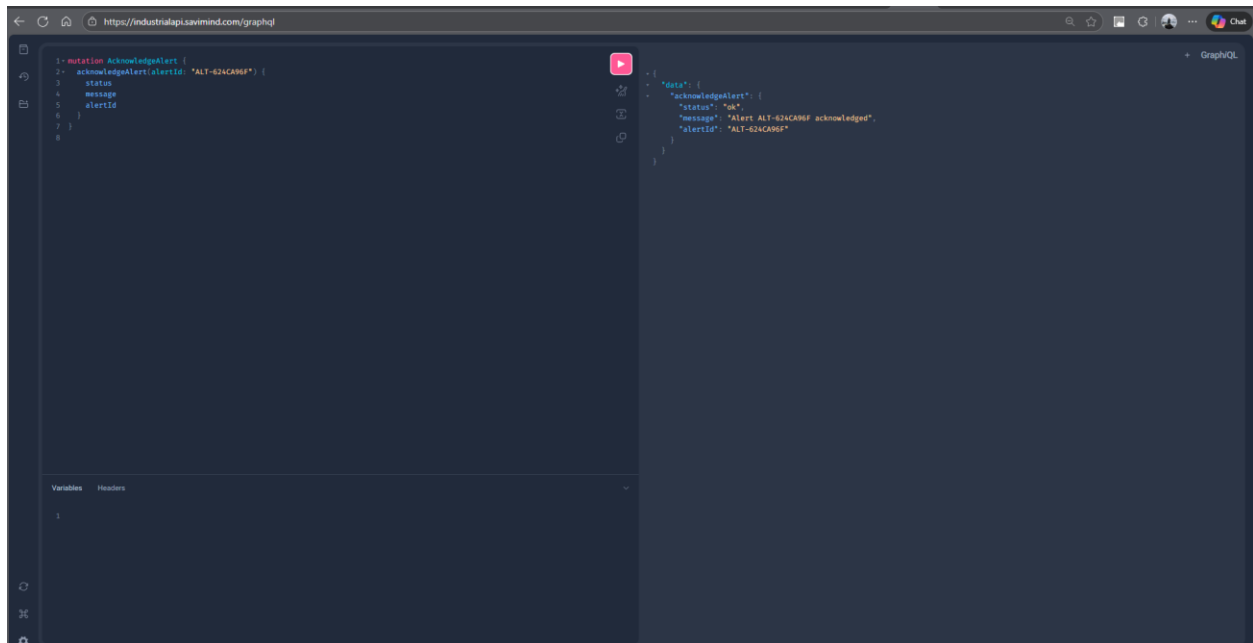


The screenshot shows the GraphQL Playground interface with the following query and response:

```
mutation IngestVitalData {
  ingestVitalData: {
    data: {
      deviceId: "P-181"
      heartRate: 82
      oxygenSat: 97
      bodyTemperature: 36.9
      steps: 5000
      timestamp: "2025-11-18T10:25:00Z"
    }
  }
}
```

```
{
  "data": {
    "ingestVitalData": {
      "status": "ok",
      "message": "Vital data ingested",
      "patientId": "P-181",
      "patientStatus": "OK"
    }
  }
}
```

Alert Acknowledge:

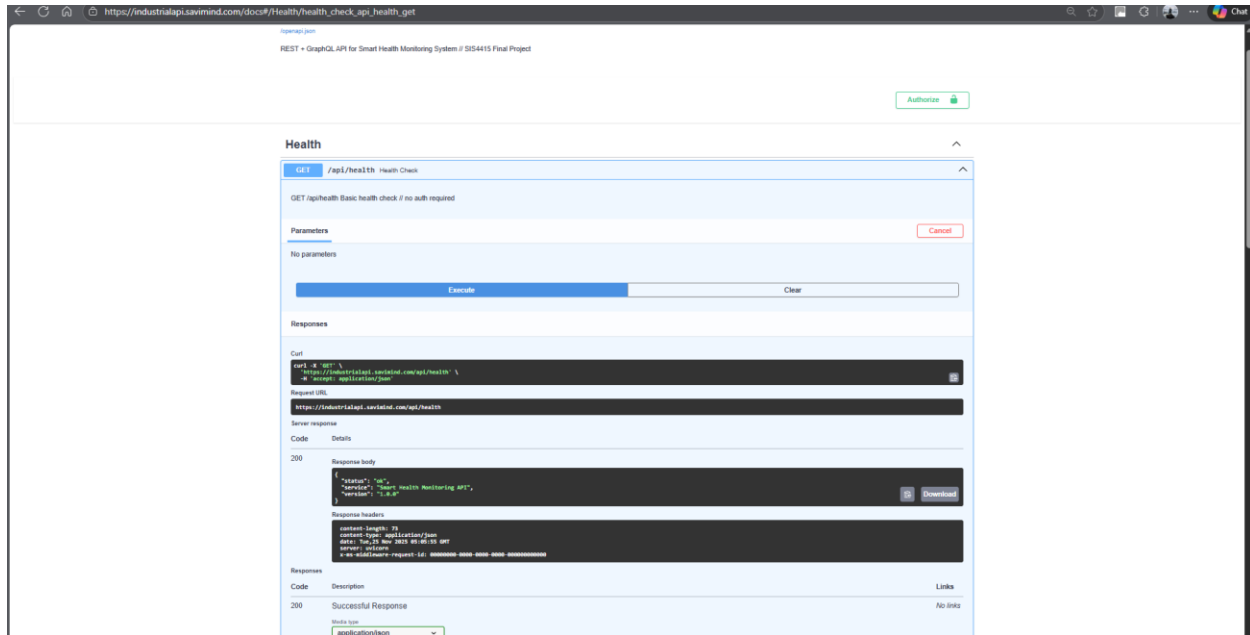


The screenshot shows the GraphQL Playground interface with the following query and response:

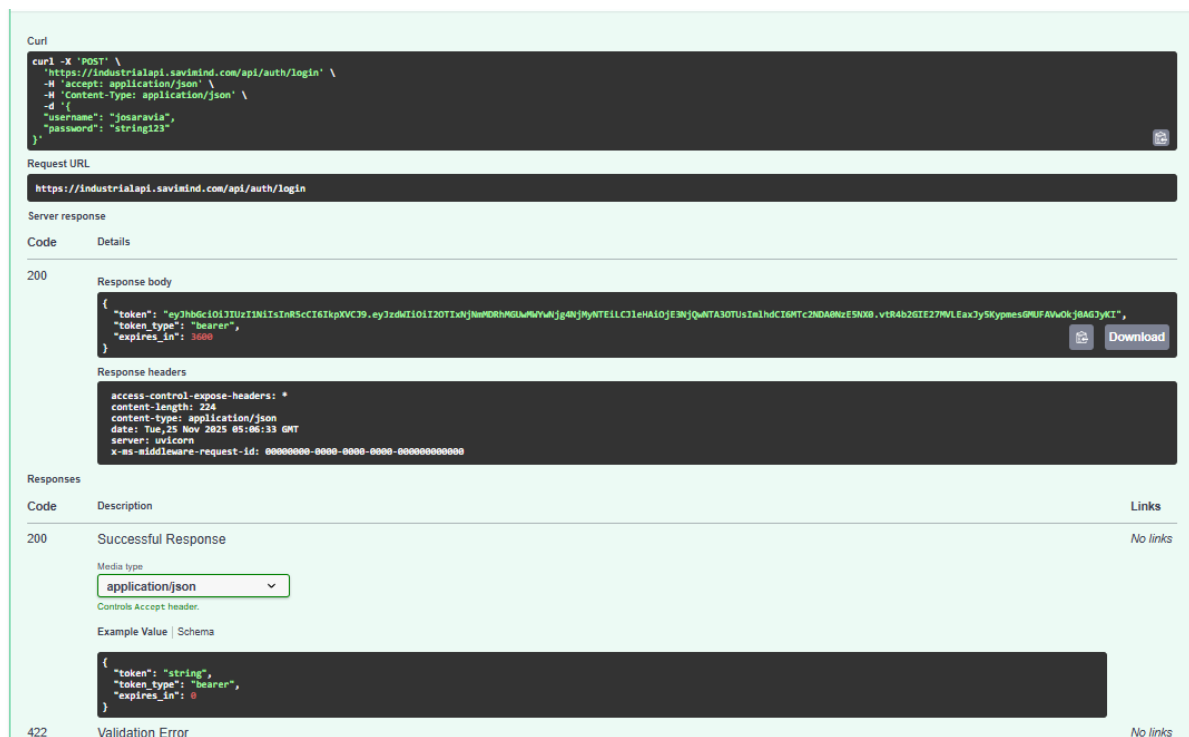
```
mutation AcknowledgeAlert {
  acknowledgeAlert(alertId: "ALT-624CA96F") {
    status
    message
    alertId
  }
}
```

```
{
  "data": {
    "acknowledgeAlert": {
      "status": "ok",
      "message": "Alert ALT-624CA96F acknowledged",
      "alertId": "ALT-624CA96F"
    }
  }
}
```


REST API Swagger



Login





Acknowledgement

POST

/api/alerts/{alert_id}/ack Acknowledge Alert

POST /api/alerts/{alert_id}/ack Acknowledge an active alert Protected: requires JWT

Parameters

Cancel

Name	Description
alert_id * required	
string (path)	ALT-C6672E49

ExecuteClear

Responses

Curl

```
curl -X 'POST' \
  https://industrialapi.savind.com/api/alerts/ALT-C6672E49/ack' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer ey3hb6c1013TuzI3N1tsIn85cC161kKVC39.eyJzdktl01T20TdxJWpDRm9GdWwWWhJg4k3hWTEl1C31eHA1OjE3NjQwMTA3OTUsInRhdC16MTc2NDQ4NzE5N08.v1R4b26TE27WVLEax3ySkypmesGwUFAVwOkj8AG3yKI' \
  -d ''
```

Request URL

https://industrialapi.savind.com/api/alerts/ALT-C6672E49/ack

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "status": "ok", "message": "Alert ALT-C6672E49 acknowledged", "alert_id": "ALT-C6672E49" }</pre></div><div>Download</div></div> <div><div>Response headers</div><div><pre>access-control-expose-headers: * content-length: 85 content-type: application/json date: Tue, 25 Nov 2025 05:08:55 GMT server: uvicorn x-ms-middleware-request-id: 00000000-0000-0000-0000-000000000000</pre></div></div>

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

Saravia Monreal Joel Antonio

Patient Readings (by id)

GET

/api/patients/{patient_id}/readings

Get Patient Readings

🔒

GET /api/patients/{patient_id}/readings Get recent readings for a patient Protected: requires JWT

Parameters

Cancel

Name	Description
patient_id * required	
string (path)	P-103
limit	
integer (query) maximum: 100 minimum: 1	10

Execute

Clear

Responses

Curl

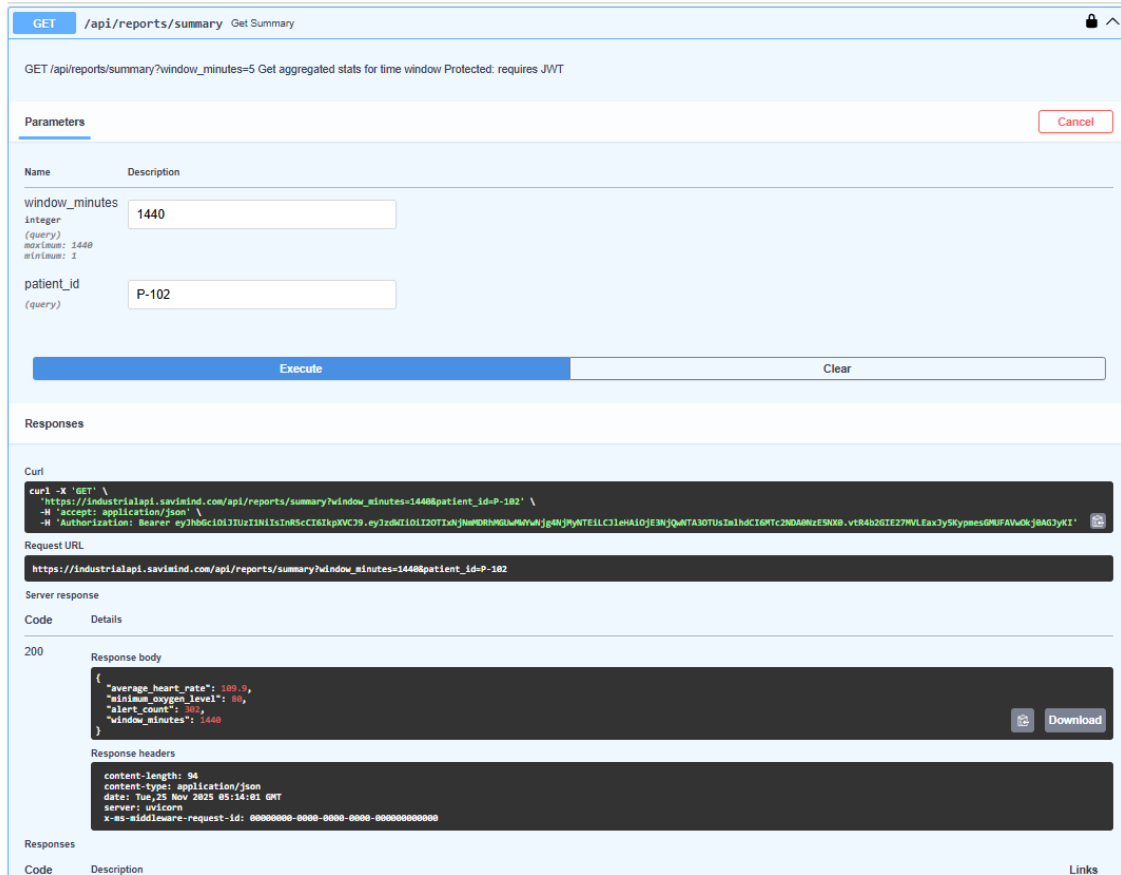
```
curl -X 'GET' \
  'https://industrialapi.savimind.com/api/patients/P-103/readings?limit=10' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ2ZmTiO1I2OTI0djh5aWQ0RmRlZmVhYjg4NjMyNTU1LjI0eA10jE3KjQwNTA3OTUsIm1hdCI6MTY2NDh0NzE5MDk0LmVtR4b2GE27MVEax3y5Kypnes0WUFAVw0Kj8AGZyKT'
```

Request URL

https://industrialapi.savimind.com/api/patients/P-103/readings?limit=10

Server response

Code	Details
200	<div>Response body</div> <pre>{ "items": [{ "id": "69252a2e136ad67b9fa3841", "patient_id": "P-103", "heart_rate": 73, "oxygen_level": 96, "body_temperature": 37.1, "steps": 2166, "timestamp": "2025-11-25T04:03:46.372000" }, { "id": "69252a90a136ad67b9fa3835", "patient_id": "P-103", "heart_rate": 166, "oxygen_level": 96, "body_temperature": 36.7, "steps": 6690, "timestamp": "2025-11-25T04:03:28.366000" }, { "id": "69252a7ea136ad67b9fa3829", "patient_id": "P-103" }] }</pre>



Saravia Monreal Joel Antonio