

[**Conte**](#_q7jfcqtbzuui)**xt 1**

[**Problem statement**](#_9uvy81tsy073) **1**

[**Input Data**](#_jlxun6aejy1t) **2**

[**Scenarios:**](#_bbep4azyla2) **3**

[Login page](#_4yxqpj6k5vmr)  **3**

[GPS summary page](#_y9u3tue4pfw0) **3**

[Functional requirements](#_2m4iwd3ljacf) **3**

[GPS detail page](#_xf343f8wbgc3)  **3**

[**Technologies to be used**](#_c6hds0uvgkzp) **4**

[**Tips**](#_lq7u79wlq7qh) **4**

[**Evaluation (300 points)**](#_915bvqt59l8d) **4**

[**Deliverables**](#_cc6ojtw7em6b)**:**

[Solution doc](#_d1w1czbjktvr) **5**

[Manual Test case doc](#_mp24az1lqzox) **5**

[Project](#_7hjj8hrbsxqp) **5**

[Demo](#_xv54woktu9n4)  **5**

### Context

This coding test is intended to give a semblance of the work done at Pensieve. Besides the actual solution and coding skills, these competencies are evaluated implicitly -

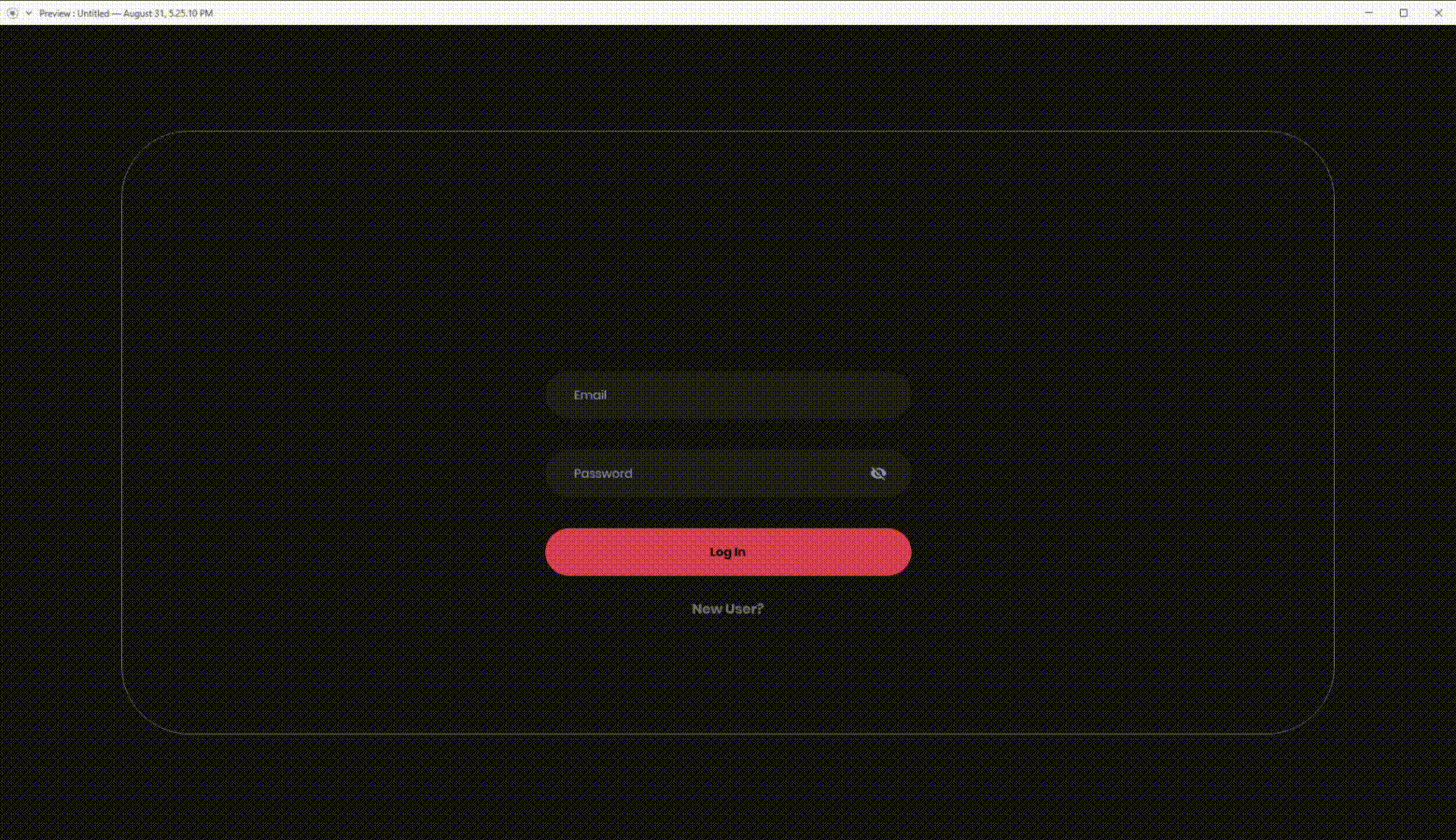
* Passion for building customer-centric products
* Innovation
* Self-drive and ability to ramp up
* Speed of execution
* Prioritisation
* Being a go-getter.

These traits are essential to work in a fast-paced environment like Pensieve.

### Problem statement

Company X wants an application to store the GPS data of their devices and provide insights from this data. This is the expected user experience:

*Login page → GPS Summary page → GPS detail page*

**

### **Input Data**

Import the sample GPS data collected at intervals of 5 min to any RDBMS store (E.g: MySql / PostgreSQL).

All timestamps are in the same time zone.

| **DeviceId** | **Device Type** | **Timestamp** | **location** |
| --- | --- | --- | --- |
| D-1567 | Aircraft | 31-08-2022 10:05 | L1 |
| D-1567 | Aircraft | 31-08-2022 10:10 | L1 |
| D-1567 | Aircraft | 31-08-2022 10:15 | L1 |
| D-1567 | Aircraft | 31-08-2022 10:20 | L1 |
| D-1567 | Aircraft | 31-08-2022 10:25 | L2 |
| D-1568 | Personal | 31-08-2022 10:05 | L3 |
| D-1568 | Personal | 31-08-2022 10:10 | L3 |
| D-1568 | Personal | 31-08-2022 10:15 | L3 |
| D-1568 | Personal | 31-08-2022 10:20 | L3 |
| D-1568 | Personal | 31-08-2022 10:25 | L3 |
| D-1569 | Asset | 31-08-2022 10:15 | L4 |
| D-1569 | Asset | 31-08-2022 10:20 | L4 |
| D-1569 | Asset | 31-08-2022 10:25 | L1 |
| D-1569 | Asset | 31-08-2022 10:30 | L1 |
| D-1569 | Asset | 31-08-2022 10:35 | L2 |
| D-1570 | Personal | 31-08-2022 10:35 | L5 |
| D-1571 | Asset | 31-08-2022 10:35 | L6 |

**!! Important Note !!**

Please make sure that your database adheres to the following schema, table, column name and data types:

**Schema name**: pensieve-test

**Table 1**:

Name: gps

Columns:

* id: serial PRIMARY KEY
* device\_id: VARCHAR(255) NOT NULL
* device\_type: VARCHAR(255) NOT NULL
* timestamp: TIMESTAMP NOT NULL
* location: VARCHAR(255) NOT NULL
* created\_at: TIMESTAMP NOT NULL DEFAULT now()

**Table 2**:

Name: users

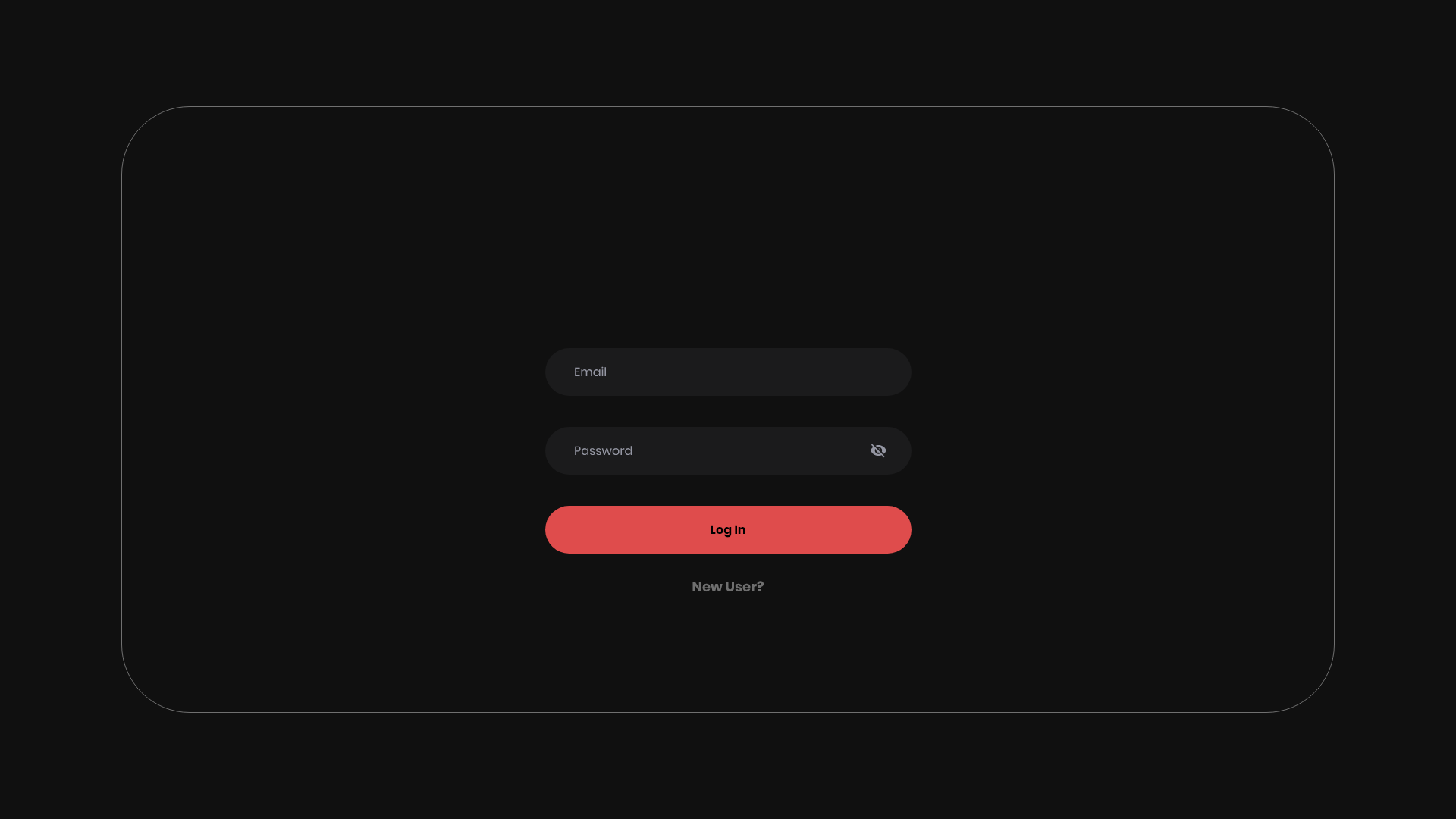
Columns:

* id: serial PRIMARY KEY
* username: VARCHAR(255) NOT NULL UNIQUE
* email: VARCHAR(255) NOT NULL UNIQUE
* password: VARCHAR(255) NOT NULL
* created\_at: TIMESTAMP NOT NULL DEFAULT now()

### Scenarios:

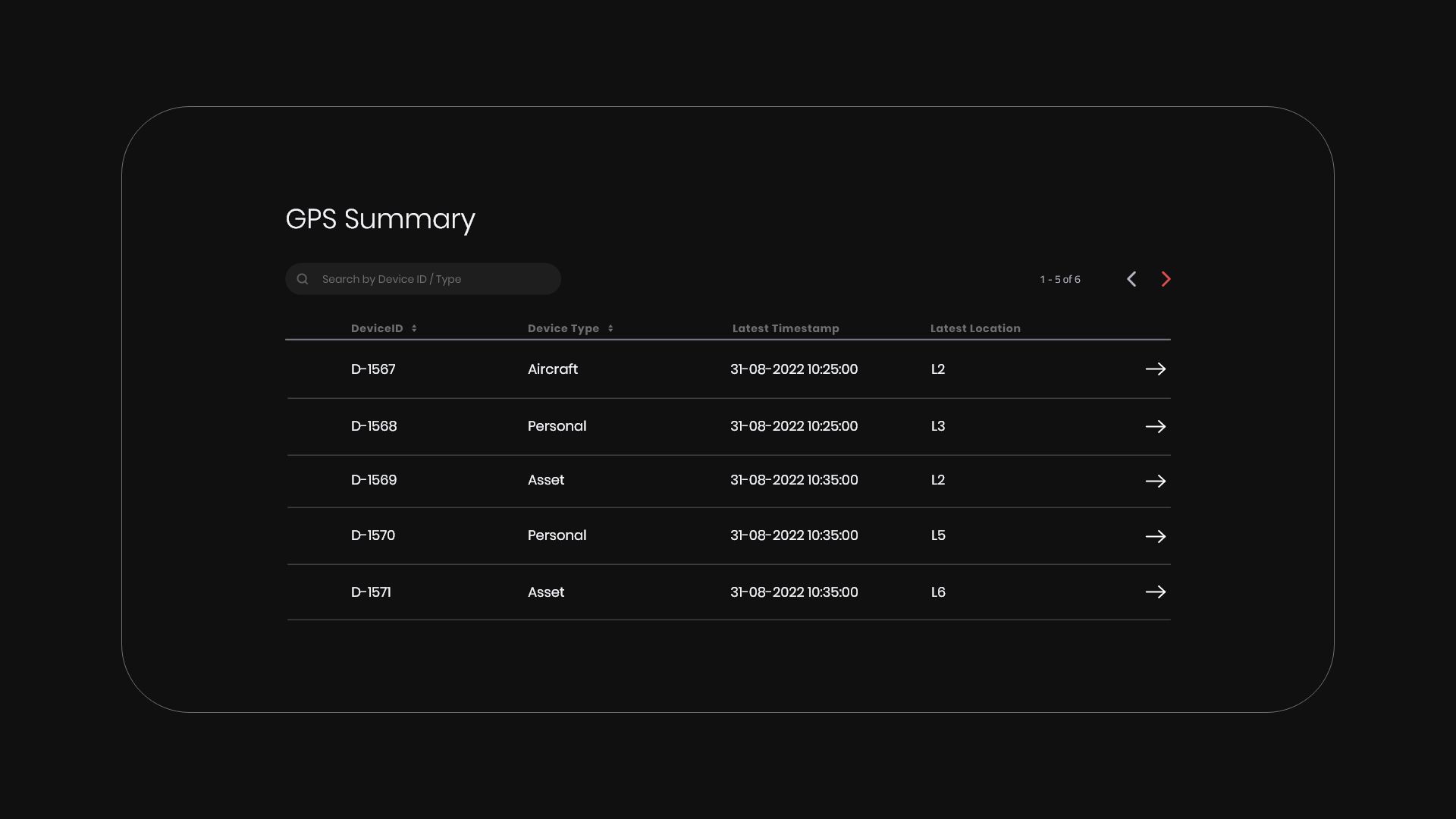
#### Login page

* User creation
  + Take username and password from user and store in database.
  + Mocks are not given. Choose the best user experience.
* User login
  + Validate user credentials and allow login.



#### GPS summary page

Display the unique GPS devices with the latest GPS entry in the format shown below.



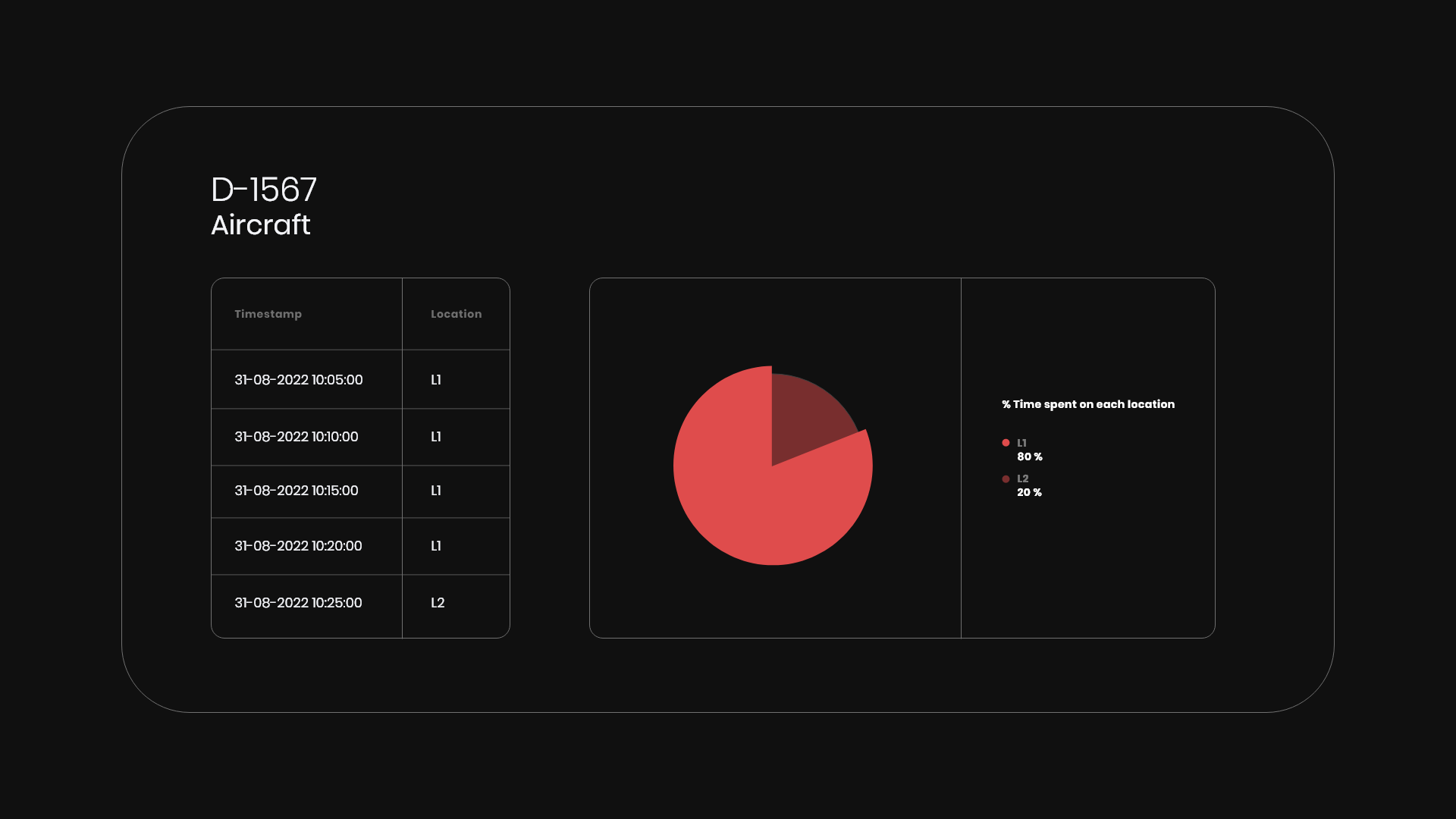
##### Functional requirements

* Sort data by any field
* Search by device id and device type.
* Provide pagination support i.e. display only 5 entries per page.
* Clicking on any specific entry (--->) should navigate to the GPS details page.

#### GPS detail page

* Display the details of the selected GPS device.
* Display in a pie chart with the % of time spent at each location.
  + Each entry is of 5 min duration.
    - For D-1567, time spent at
      * L1 is 20 min as it has 4 entries - 80%
      * L2 is 5 min as it has 1 entry - 20%
    - For D-1571, time spent at L6 (one entry) is 100%.

| **DeviceId** | **Device Type** | **Timestamp** | **location** |
| --- | --- | --- | --- |
| D-1567 | Aircraft | 31-08-2022 10:05 | L1 |
| D-1567 | Aircraft | 31-08-2022 10:10 | L1 |
| D-1567 | Aircraft | 31-08-2022 10:15 | L1 |
| D-1567 | Aircraft | 31-08-2022 10:20 | L1 |
| D-1567 | Aircraft | 31-08-2022 10:25 | L2 |
| D-1571 | Asset | 31-08-2022 10:35 | L6 |



### Technologies to be used

* **Frontend (FE):** React.JS
* **Backend (BE):** Node.JS
* FE & BE communicate through REST APIs.
* **Database:** RDBMS (E.g.: MySql)

### Tips

* Feel free to choose the theme, font and style that is more appealing to the user.
* Apply your own validations/error messages/enhancements wherever required.

Mocks are given only for reference. All details are not given in mocks.These are left open-ended deliberately.

* You are given only the sample data. Ensure that this solution works with data of similar format.
* Scalability is out of scope.
* Prioritise your execution to get the maximum points in the given time.

### Evaluation (300 points)

| **Deliverable** | **Max. Points** |
| --- | --- |
| Login Page (50 points) | User creation:20  User login:20  Product validations: 10 |
| Navigation to GPS summary page  (80 points) | Display results: 30  Pagination: 20  Searching: 20  Sorting:10 |
| Navigation to GPS details page (70 points) | Pie chart: 40  Table: 30 |
| Manual Testing document | 20 |
| Unit tests - React.JS | 25 |
| Unit tests - Node.JS | 25 |
| API documentation | 10 |
| Possible enhancements/Scalability | 20 |
| Functionality bugs | High priority (P0): **-15**  Medium priority (P1): **-10**  Low priority (P2): **-5** |

### Deliverables

Share these deliverables in a zip file/G-drive link providing proper access. Please check your mail for more details.

##### Solution doc

* Assumptions/constraints in your execution
* Technology concepts used
* Possible enhancements to improve user experience.
* Thoughts on scalability of the solution.

##### Manual Test case doc

* Mention all possible use cases (E.g.: Invalid password) you want to test to certify

your application. Think through the scenarios in detail.

##### Project

* Code
* Database Schema
* Database file
* ReadMe file: Steps to execute the project locally.

#### Demo

* Recording/demo of the project.