

Washing Machine Status Tracker

Database Management Systems - Project

Group Members:

CS22B1010-T.S.M THRIVEEDH

CS22B1099-B. SIDDHARTHA REDDY

CS22B2019-H. SAI SANDEEP

CS22B2023-T. SAI MADAN

CS22B2020 - G. SRI RAM



Contents:

1 Introduction

- 1.1 Purpose**
- 1.2 Intended Audience**
- 1.3 Project Scope**

2 Overall Description

- 2.1 Functional Requirements**
- 2.2 Non-functional Requirements**

3 System Features

- 3.1 Technology Stack**
- 3.2 System Requirements**

4 Frontend

5 Database

6 Queries

1 Introduction

Purpose:

This project allows students to efficiently manage their laundry without disturbing their schedules, reducing waiting time by being the one-stop solution for all laundry problems. Automation ensures that washing machines are utilized optimally, ensuring smooth operation for all users.

Intended Audience:

This project is particularly beneficial for hostel and PG students, catering to environments where the number of washing machines is limited compared to the high demand among the student population.

Project Scope:

This project is aimed at providing students with real-time updates on the status of individual washing machines, making it easier for them to see which ones are available whenever they need to do their laundry.

Additionally, there will be an administrative dashboard specifically designed for authorized personnel. This dashboard will allow them to update the status of the washing machines and handle any complaints that may arise.

It's worth mentioning that this project won't include any advanced features

like booking systems or remote-control capabilities for the

machines.

2 Overall Description

2.1 Functional Requirements:

- **User Authentication and Authorization:**

The system shall provide user authentication mechanisms to verify the identity of users accessing the system. Authorized users, such as staff and administration, shall have specific roles and permissions to perform certain actions within the system.

- **User Logging:**

The system shall log user interactions with the washing machines to maintain usage history. User interactions to be logged include the student's roll number and the machine ID they are using.

The logged information shall be stored securely and tamper-proof to ensure data integrity and traceability. Access to user logs shall be restricted to authorized personnel, such as administration and system operators.

- **Real-time Monitoring:**

The system shall continuously monitor the status of each washing machine in real-time, including availability, current usage, and maintenance status.

- **Admin Interface:**

The administrative dashboard shall provide authorized personnel with the ability to manage washing machine information and update statuses. Admins shall have the capability to view user log details.

2.2 Non-functional Requirements:

- **Performance:**

The system must efficiently handle concurrent requests from multiple users without experiencing significant performance degradation

- **Availability:**

The system should be available for use 24/7 with minimal planned down-time for maintenance or updates, ensuring uninterrupted service for users.

- **Security:**

Robust security measures must be implemented to protect user information. This includes the

implementation of authentication and authorization mechanisms to control access to system resources and safeguard sensitive data.

- **Usability:**

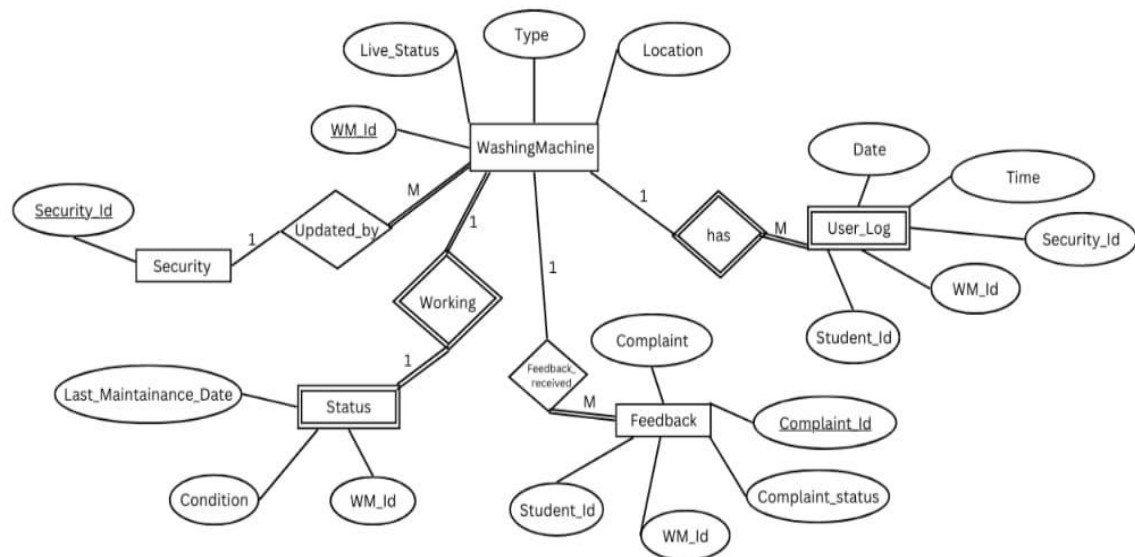
The user interface must be intuitive and user-friendly, featuring clear navigation and layout to facilitate ease of use for all users. This includes providing straightforward access to features and functionalities to enhance the overall user experience.

- **Maintainability:**

The system should be designed with modular and well documented code to facilitate ease of maintenance and accommodate future enhancements. This ensures that updates and modifications can be made efficiently without compromising system integrity or functionality.

- **Compatibility:**

The system is designed to support common web browsers and ensures seamless compatibility across a variety of platforms, ensuring accessibility and usability for all users, regardless of their preferred device or operating system.



ENTITY RELATIONSHIP DAIGRAM

3 System Features

3.1 Technology Stack:

Frontend Technologies:

- **HTML:** Used for creating the structure and content of web pages.
- **CSS:** Employed for styling and ensuring a responsive, visually appealing interface.

- **JavaScript:** Utilized for client-side scripting to add interactivity and dynamic behaviour.

Backend Technologies:

- **Node.js:** A JavaScript runtime environment for building the server-side application, known for handling multiple concurrent connections efficiently.
- **Express.js:** A web application framework for Node.js, used for routing, handling HTTP requests, and integrating middleware.
- **Pg:** A PostgreSQL client for Node.js that facilitates communication with the database.

Database Technologies:

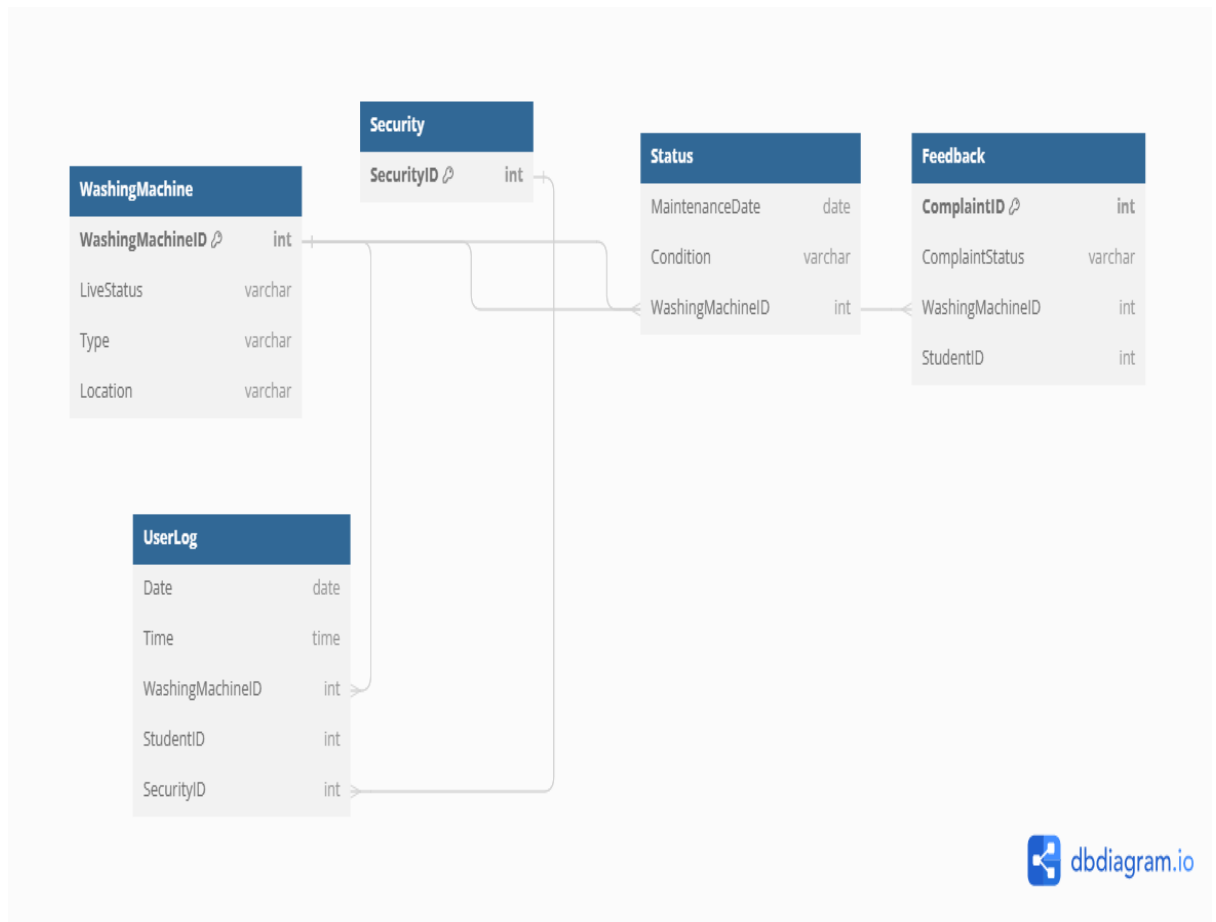
- **PostgreSQL:** An open-source relational database management system used to store and manage all application data, offering features like ACID compliance and support for complex queries.

3.2 System Requirements:

Works on any operating system.

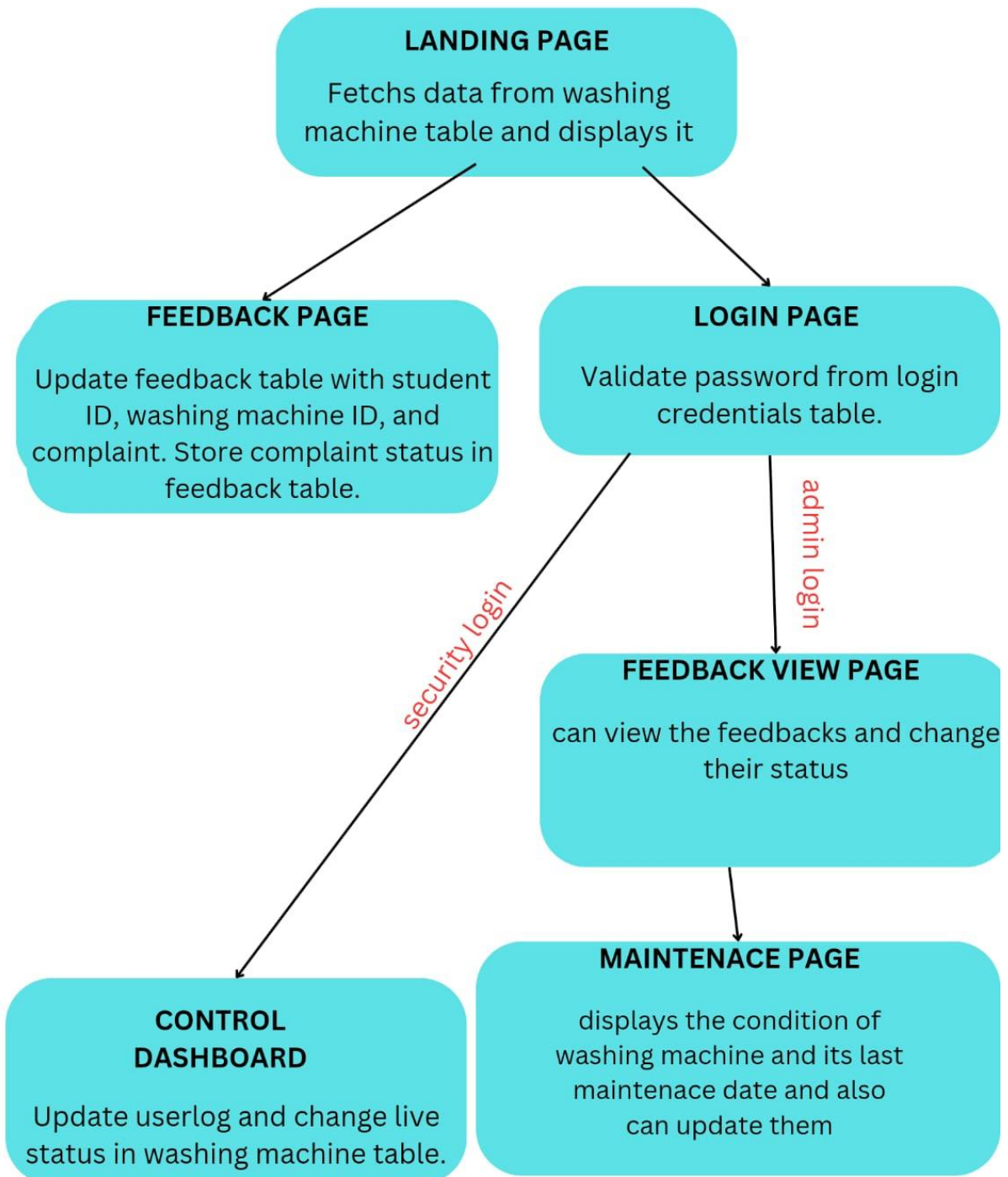
Internet connection is required.

Works on any browser.



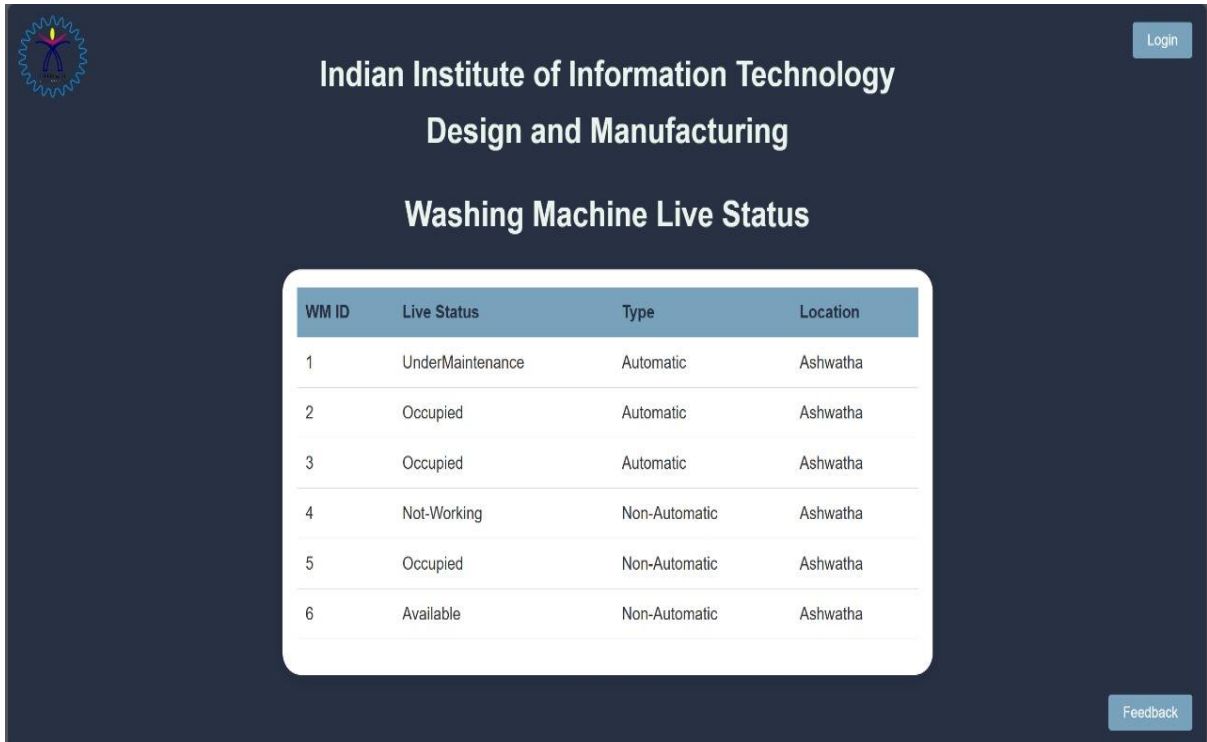
SCHEMA

PROJECT FLOW CHART



4 Frontend

Landing Page:

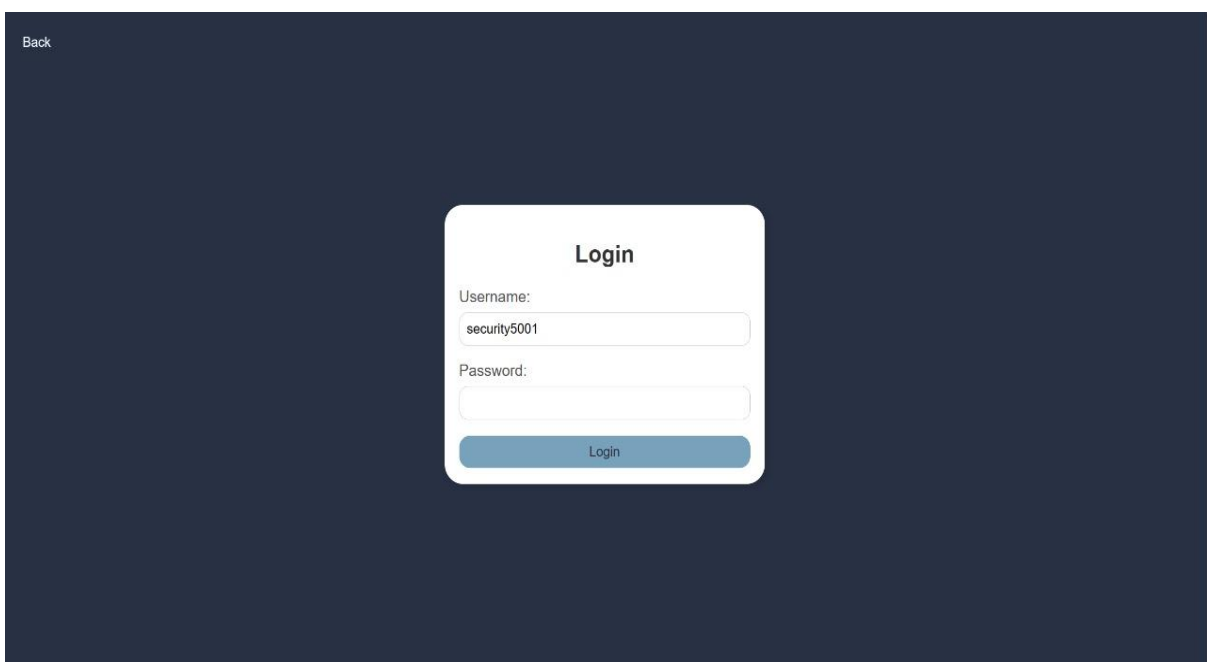


The landing page features a dark blue background. In the top left corner is the Indian Institute of Information Technology logo. The header text reads "Indian Institute of Information Technology" and "Design and Manufacturing". A "Login" button is in the top right. The main title is "Washing Machine Live Status". Below it is a table with 6 rows and 4 columns: WM ID, Live Status, Type, and Location. The table data is as follows:

WM ID	Live Status	Type	Location
1	UnderMaintenance	Automatic	Ashwatha
2	Occupied	Automatic	Ashwatha
3	Occupied	Automatic	Ashwatha
4	Not-Working	Non-Automatic	Ashwatha
5	Occupied	Non-Automatic	Ashwatha
6	Available	Non-Automatic	Ashwatha

A "Feedback" button is located in the bottom right corner.

Login Page:



The login page has a dark blue background. A "Back" link is in the top left. The login form is centered and contains the following fields:

- Login** (title)
- Username:**
- Password:**
- Login** (button)

Feedback Page:

[Back](#)

Help Us Improve: Your Feedback Counts!

Feedback Form

Student ID:

Complaint:

Washing Machine ID:

Submit Feedback

Live Status Updating Page:

[Back](#)

Washing Machine Live Status

Control Dashboard

WM ID	Live Status	Type	Location	Roll Number	Security ID	Toggle
1	Available	Automatic	Ashwatha	<div>Roll Number</div>	<div>Security ID</div>	<div>Toggle</div>
2	Occupied	Automatic	Ashwatha	<div>Roll Number</div>	<div>Security ID</div>	<div>Toggle</div>
3	Occupied	Automatic	Ashwatha	<div>Roll Number</div>	<div>Security ID</div>	<div>Toggle</div>
4	Not-Working	Non-Automatic	Ashwatha	<div>Roll Number</div>	<div>Security ID</div>	<div>Toggle</div>
5	Occupied	Non-Automatic	Ashwatha	<div>Roll Number</div>	<div>Security ID</div>	<div>Toggle</div>
6	Available	Non-Automatic	Ashwatha	<div>Roll Number</div>	<div>Security ID</div>	<div>Toggle</div>

User Log

Security ID	Date	Time	WM ID	Student ID
5002	2024-08-06T18:30:00.000Z	11:00:00	4	CS22B2024
5003	2024-08-05T18:30:00.000Z	09:00:00	1	CS22B2022
5003	2024-08-05T18:30:00.000Z	10:00:00	3	CS22B2023

continue..User Log:

Back

WM ID	Condition	Automatic	Washing Machine	Roll Number	Security ID	Pass
4	Not-Working	Non-Automatic	Ashwatha	<input type="text"/>	<input type="text"/>	<input type="button" value="Toggle"/>
5	Occupied	Non-Automatic	Ashwatha	<input type="text"/>	<input type="text"/>	<input type="button" value="Toggle"/>
6	Available	Non-Automatic	Ashwatha	<input type="text"/>	<input type="text"/>	<input type="button" value="Toggle"/>

User Log

Security ID	Date	Time	WM ID	Student ID
5002	2024-08-06T18:30:00.000Z	11:00:00	4	CS22B2024
5003	2024-08-05T18:30:00.000Z	09:00:00	1	CS22B2022
5003	2024-08-05T18:30:00.000Z	10:00:00	3	CS22B2023
5001	2024-08-04T18:30:00.000Z	17:00:00	1	CS22B2020
5002	2024-08-04T18:30:00.000Z	18:00:00	2	CS22B2021
5001	2024-05-16T18:30:00.000Z	09:52:40	1	CS22B2019
5003	2024-05-16T18:30:00.000Z	12:00:09	2	CS24B2020
5003	2024-05-16T18:30:00.000Z	12:32:17	6	CS24B2021
5003	2024-05-16T18:30:00.000Z	14:33:05	6	CS24B2021
5002	2024-05-16T18:30:00.000Z	15:15:47	2	CS22B2058

Maintenance Page:

Back

Washing Machine Maintenance

Status Table

Machine ID	Condition	Last Maintenance Date
1	Working	2024-05-14
2	Working	2024-05-14
3	Working	2023-04-07
4	Not-Working	2023-04-07
5	Working	2023-07-07
6	Working	2024-04-16

Update Section

Select Washing Machine ID:

1

Select Condition:

Under Maintenance

Select Maintenance Date:

mm/dd/yyyy

Update

5 Database

Feedback table : stores feedbacks and complaints given by user

Attributes : Complaint, Complaint_ID(Primary Key), Complaint_Status, WM_ID(Foreign key referencing WashingMachine.WM_ID)

,

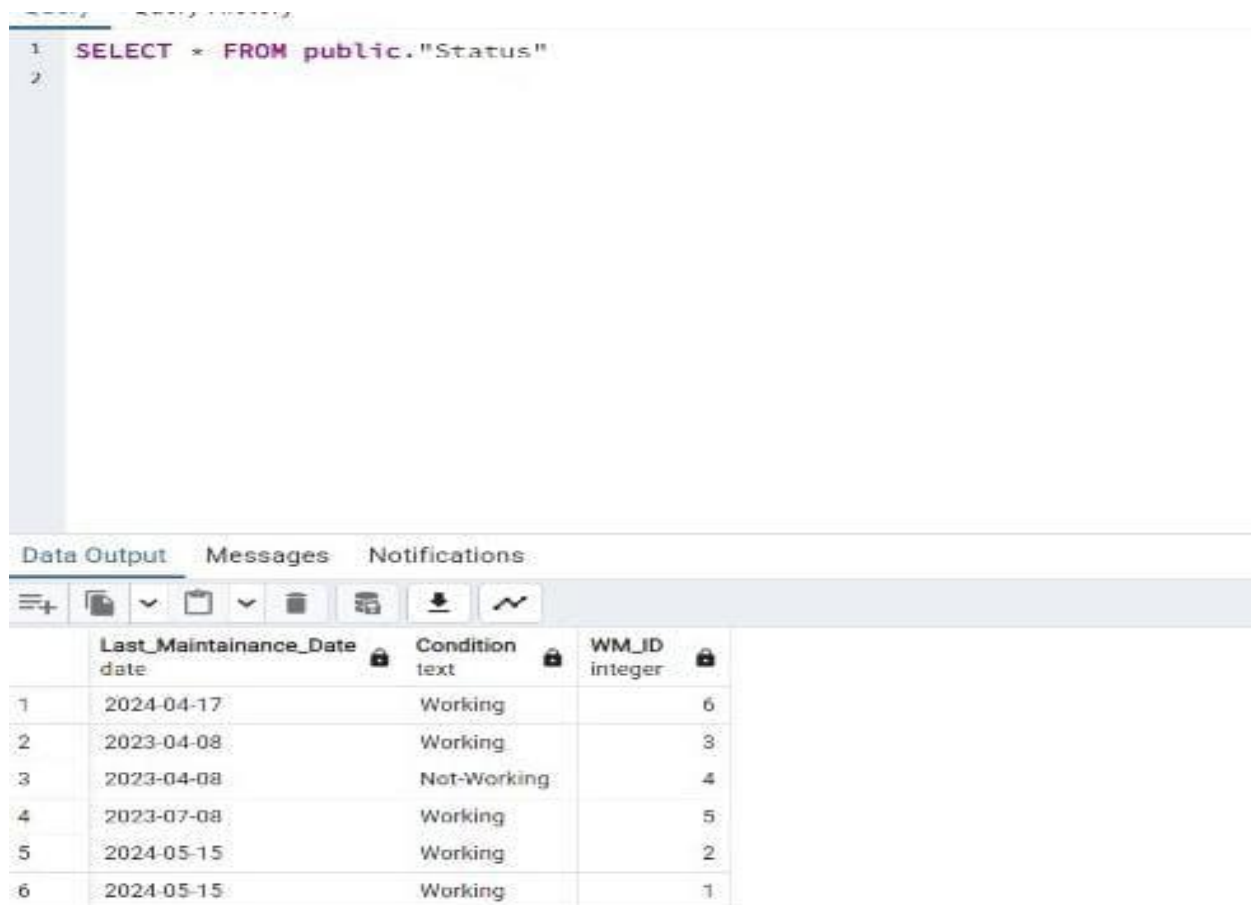
Student_ID

Query		Query History	
1	SELECT * FROM public."Feedback"		
2	ORDER BY "Complaint_ID" ASC		

Data Output		Messages		Notifications	
	Complaint text	Complaint_ID [PK] integer	Complaint_Status text	WM_ID integer	Student_ID text
1	WashingMachine 1 not working	1	Solved	1	2020
2	WashingMachine 5 spinnner not working	2	Solved	5	2023
3	Getting shock from washing machine 3	3	Working on it	3	1100
4	Rinse cycle issue	4	Working on it	4	2025
5	No hot water	5	Working on it	5	2014
6	Slow spin cycle	6	Working on it	6	2016

Status table : Stores status of each washing machine

Attributes: Last_Maintainance_Date,Condition,WM_ID(Foreign key referencing WashingMachine.WM_ID)



The screenshot shows a database query interface. At the top, a SQL query is entered in a text area:

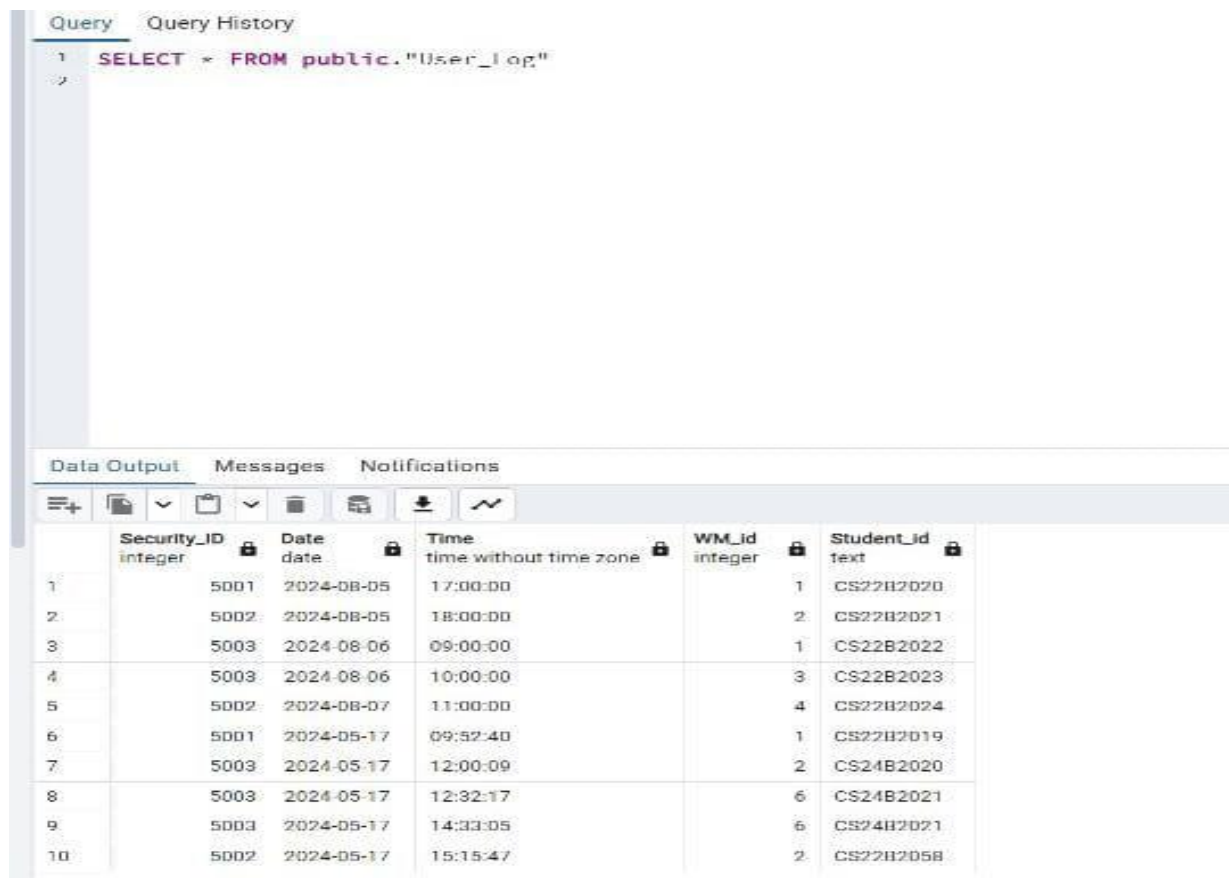
```
1 SELECT * FROM public."Status"
2
```

Below the query area, there are three tabs: "Data Output", "Messages", and "Notifications". The "Data Output" tab is selected, displaying a table of results. The table has four columns: "Last_Maintainance_Date" (date), "Condition" (text), "WM_ID" (integer), and an unlabeled column with a lock icon. The data is as follows:

	Last_Maintainance_Date date	Condition text	WM_ID integer	
1	2024-04-17	Working		6
2	2023-04-08	Working		3
3	2023-04-08	Not-Working		4
4	2023-07-08	Working		5
5	2024-05-15	Working		2
6	2024-05-15	Working		1

User_Log table : Stores complete Logs of users who used washingMachine

Attributes: Security_ID(Foreign Key referencing security.security_id),Date,Time,WM_id(Foreign key referencing WashingMachine.WM_ID),Student_id

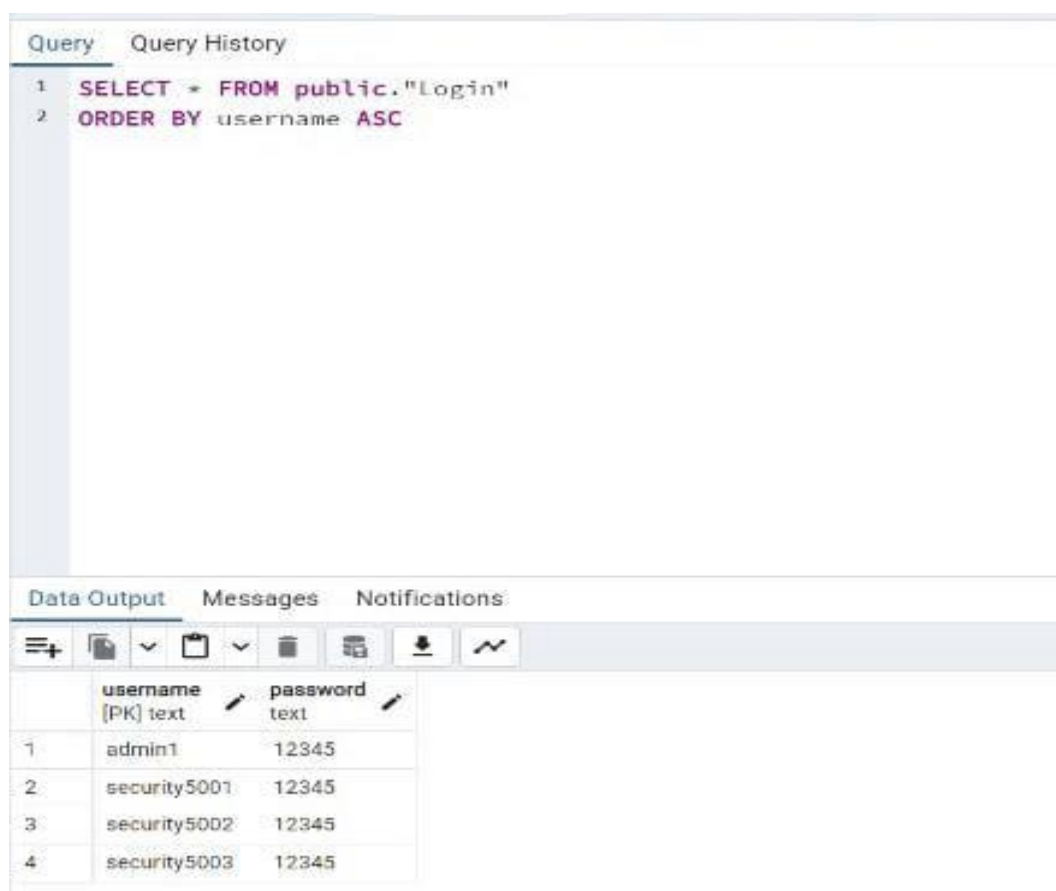


The screenshot displays a database query interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, showing a SQL query: `SELECT * FROM public."User_Log"`. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with 10 rows of data. The table has columns for 'Security_ID' (integer), 'Date' (date), 'Time' (time without time zone), 'WM_id' (integer), and 'Student_id' (text). Each column header has a lock icon. The data rows are numbered 1 through 10 in the first column.

	Security_ID integer	Date date	Time time without time zone	WM_id integer	Student_id text
1	5001	2024-08-05	17:00:00	1	CS22B2020
2	5002	2024-08-05	18:00:00	2	CS22B2021
3	5003	2024-08-06	09:00:00	1	CS22B2022
4	5003	2024-08-06	10:00:00	3	CS22B2023
5	5002	2024-08-07	11:00:00	4	CS22B2024
6	5001	2024-05-17	09:52:40	1	CS22B2019
7	5003	2024-05-17	12:00:09	2	CS24B2020
8	5003	2024-05-17	12:32:17	6	CS24B2021
9	5003	2024-05-17	14:33:05	6	CS24B2021
10	5002	2024-05-17	15:15:47	2	CS22B2058

Login table : Stores Login details

Attributes: UserName(Primary Key), Password



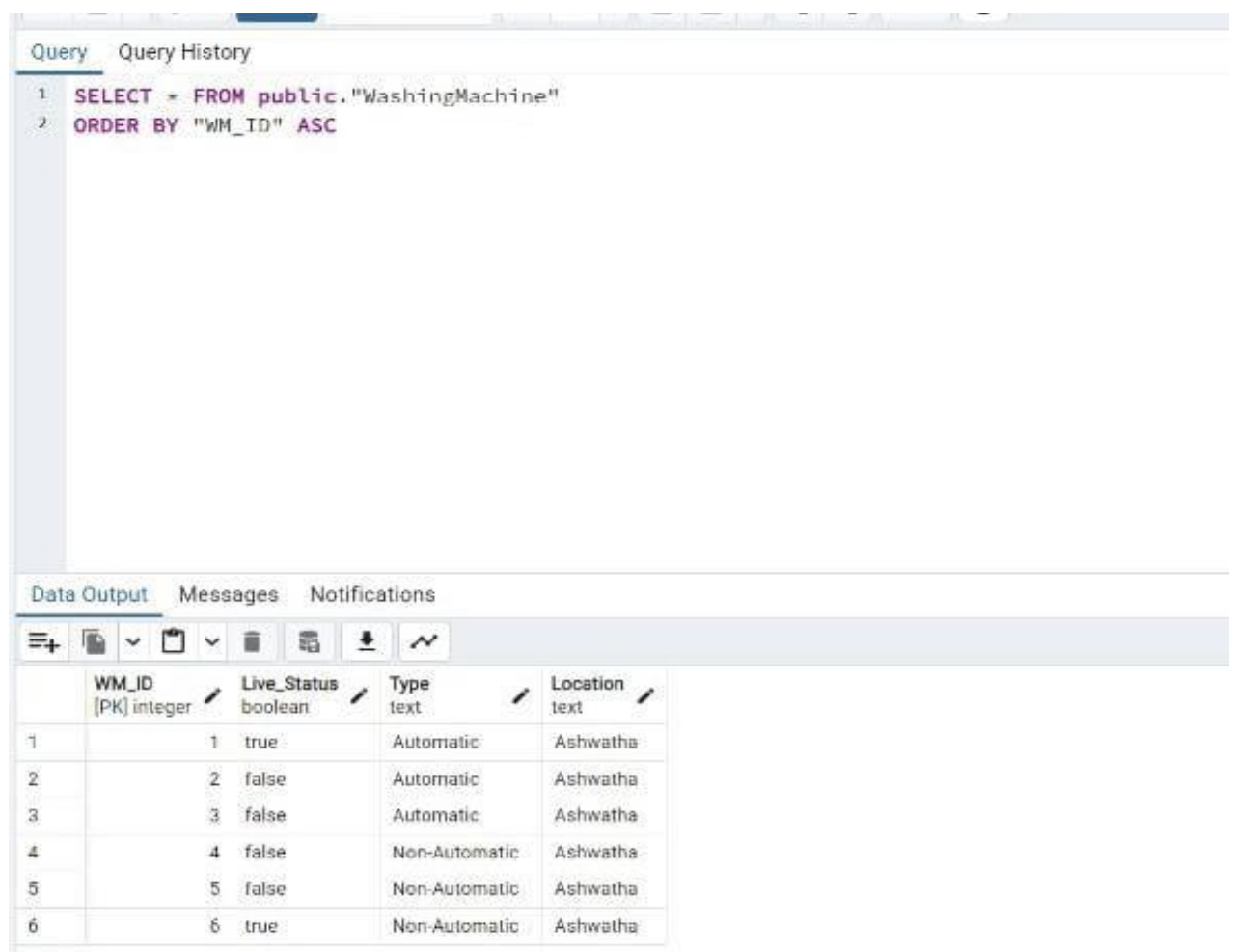
The screenshot shows a database query tool interface. The top section has tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query:

```
1 SELECT * FROM public."Login"  
2 ORDER BY username ASC
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table of results. The table has two columns: 'username' (marked as a primary key [PK] and text type) and 'password' (text type). The results are as follows:

	username [PK] text	password text
1	admin1	12345
2	security5001	12345
3	security5002	12345
4	security5003	12345

WashingMachine table : Stores details and live status(available/occupied) of each washing machine
Attributes: WM_ID(Primary Key),Live_Status,Type,Location



The screenshot shows a database query tool interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query:

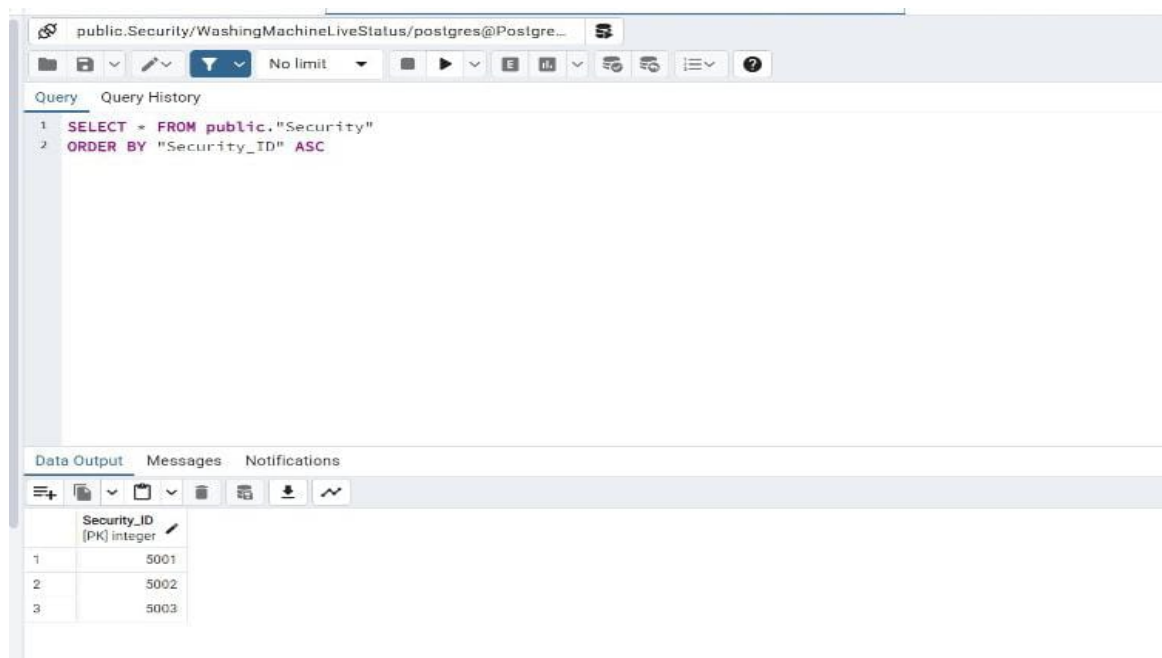
```
1 SELECT * FROM public."WashingMachine"  
2 ORDER BY "WM_ID" ASC
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table of results. The table has five columns: 'WM_ID' (integer, primary key), 'Live_Status' (boolean), 'Type' (text), and 'Location' (text). There are six rows of data.

	WM_ID [PK] integer	Live_Status boolean	Type text	Location text
1	1	true	Automatic	Ashwatha
2	2	false	Automatic	Ashwatha
3	3	false	Automatic	Ashwatha
4	4	false	Non-Automatic	Ashwatha
5	5	false	Non-Automatic	Ashwatha
6	6	true	Non-Automatic	Ashwatha

Security table : Stores Security_IDs

Attributes: Security_ID(Primary Key)



6 Queries

For Login:

To check the login credentials, if rows are returned its successful else user will be asked to try again.

```
query = `SELECT * FROM public.\"Login\" WHERE username = $1 AND
password = $2`;
```

To show the live status in landing page:

```
query = `
    SELECT wm."WM_ID", wm."Live_Status", wm."Type",
    wm."Location", s."Condition"
    FROM public."WashingMachine" wm
    JOIN public."Status" s ON wm."WM_ID" = s."WM_ID"
`;
```

To insert into User Log table:

```
query = `
    INSERT INTO public."User_Log" ("Security_ID",
    "Date", "Time", "WM_id", "Student_id")
    VALUES ($1, $2, $3, $4, $5)
`;
```

To update the live status of washing machine:

```
query('UPDATE public."WashingMachine" SET "Live_Status" = NOT
"Live_Status" WHERE "WM_ID" = $1 RETURNING *', [wmId])
```

To register a complaint or feedback:

```
query = `
    INSERT INTO public."Feedback" ("Complaint", "Complaint_ID",
    "Complaint_Status", "WM_ID", "Student_ID")
    VALUES ($1, $2, $3, $4, $5)
`;
```

To update the status of the complaint (solved/working on it):

```
query('UPDATE public."Feedback" SET "Complaint_Status" = $1 WHERE  
"Complaint_ID" = $2')
```

To fetch and display whole data from status table:

```
query('SELECT * FROM "Status"')
```

To update the status of washing machine in
Status table (Under maintenance/Not-
Working/Working):

```
query = 'UPDATE "Status" SET "Condition" = $1,  
"Last_Maintainance_Date" = $2 WHERE "WM_ID" = $3';
```

To fetch and show all the logs in user log:

```
query = `
    SELECT "Security_ID", "Date", "Time", "WM_id", "Student_id"
    FROM public."User_Log"
`;
```

