

Introduction

Renting a property can be a very good option for people but only if they get a good landlord. According to a national survey of housing tenants, renters in Australia feel that they lack the power to demand standard property maintenance.¹ Tenants feel that even for basic property management or damage they have to make multiple calls to Landlords. Landlords act as middlemen between renters and tradesman, they manage when the renters would be home so that the traders can come and complete their repair jobs. This system makes the entire process full for hassle which results in delays, improper maintenance, and bad repair jobs. Looking at this from a landlord's point of view, one person can have multiple properties to look after, and it can get difficult to manage all property problems together. The system that I have developed here allows the tenants to file a complaint which the tradesmen can see and act upon. The system will show to the tradesmen - the complaint description and property ID. Traders will be able to edit or delete a complaint. The landlords will be able to see a combined result of property with their respective complaints filed. The current development of the idea is on a very small scale, but I believe this idea can be converted into a full-scale project and can be implemented in real life.

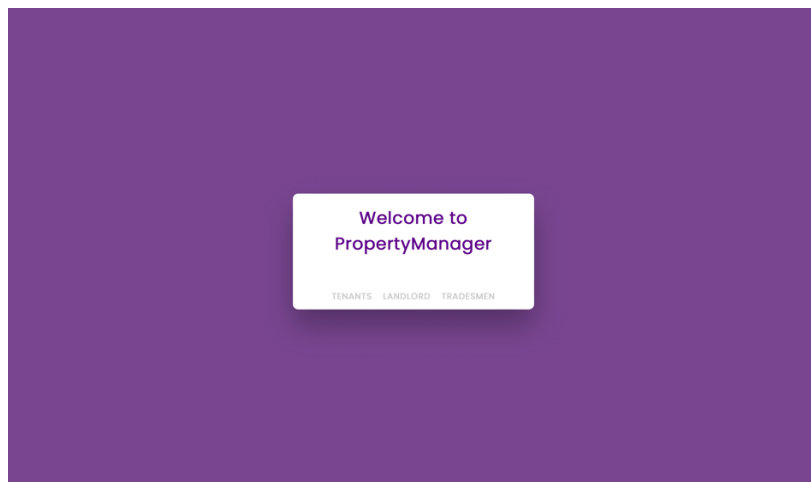


Fig 1: Welcome page of PropertyManager

Requirements:

Programming language:

Front end: HTML, CSS, Javascript, Bootstrap

Back end: MySQL, Python, Flask

Software: VS Code, XAMPP (For localhost and SQL database)

¹ <https://www.abc.net.au/news/2017-02-21/australians-share-their-rental-horror-stories/8277394>

Query Demonstration

1. Join query:

I have implemented join query between 'Complaints' table and 'Property' table to show results of all complaints filled corresponding to the property.

Both the tables had the attribute T_id (Tenant ID). I used this attribute to implement inner join.

This join query is useful for the landlord to keep an eye on all the complaints with the corresponding property details.

Complaints table:

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

```
SELECT * FROM `Complaint`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

		Complaint_id	Tenant_id	Job_id	Description	Td_id
<input type="checkbox"/>	Edit Copy Delete	1	4	1	Power points in living room not working	3
<input type="checkbox"/>	Edit Copy Delete	2	5	2	The oven is not working properly. Needs to be chan...	5
<input type="checkbox"/>	Edit Copy Delete	3	3	3	The door garage door is not opening	4
<input type="checkbox"/>	Edit Copy Delete	4	1	4	The dishwasher water is leaking	1
<input type="checkbox"/>	Edit Copy Delete	5	2	5	There is problem with the geyser, I don't have hot...	1

Property table:

Showing rows 0 - 6 (7 total, Query took 0.0004 seconds.)

```
SELECT * FROM `Property`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

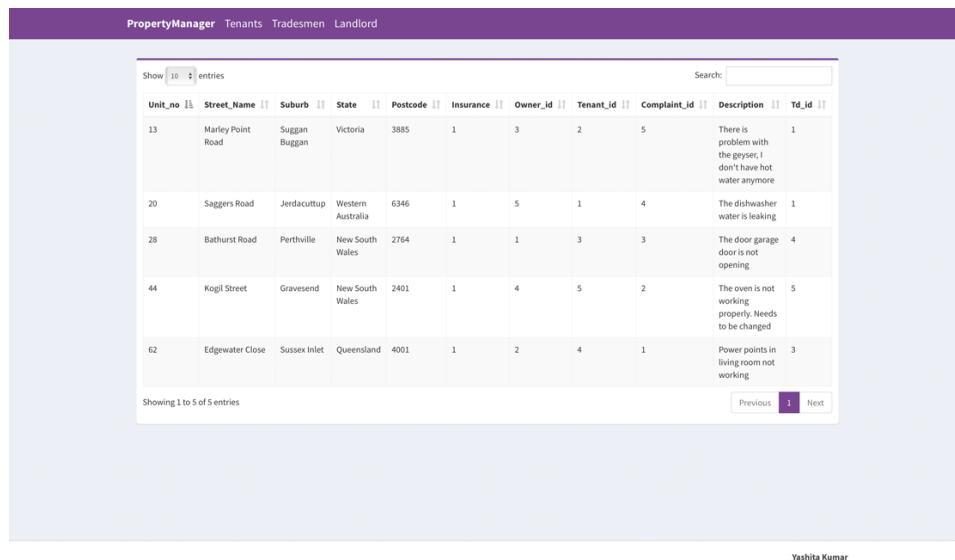
+ Options

		P_id	Type	Unit_no	Street_Name	Suburb	State	Postcode	Insurance	Owner_id	Tenant_id
<input type="checkbox"/>	Edit Copy Delete	1	House	28	Bathurst Road	Perthville	New South Wales	2764	1	1	3
<input type="checkbox"/>	Edit Copy Delete	2	Flat	13	Marley Point Road	Suggan Buggan	Victoria	3885	1	3	2
<input type="checkbox"/>	Edit Copy Delete	3	House	20	Saggers Road	Jerdacuttup	Western Australia	6346	1	5	1
<input type="checkbox"/>	Edit Copy Delete	4	House	44	Kogil Street	Gravesend	New South Wales	2401	1	4	5
<input type="checkbox"/>	Edit Copy Delete	5	House	62	Edgewater Close	Sussex Inlet	Queensland	4001	1	2	4
<input type="checkbox"/>	Edit Copy Delete	6	Bungalow	114	Bhagirath	Sodala	Rajasthan	3020	0	6	7
<input type="checkbox"/>	Edit Copy Delete	7	Flat	621	Hood street	St lucia	Queensland	4067	0	1	6

SQL CODE:

```
SELECT Property.Unit_no, Property.Street_Name, Property.Suburb, Property.State,
Property.Postcode, Property.Insurance, Property.Owner_id, Property.Tenant_id,
Complaint.Complaint_id, Complaint.Description, Complaint.Td_id FROM Property
INNER JOIN Complaint ON Property.Tenant_id=Complaint.Tenant_id ORDER BY
Complaint.Complaint_id ASC;
```

Pictorial UI:




The screenshot shows the 'PropertyManager' interface with tabs for 'Tenants', 'Tradesmen', and 'Landlord'. The 'Landlord' tab is active, displaying a table of complaints. The table has columns: Unit_no, Street_Name, Suburb, State, Postcode, Insurance, Owner_id, Tenant_id, Complaint_id, Description, and Td_id. Five rows of data are visible, showing details for various complaints such as 'There is problem with the geyser', 'The dishwasher water is leaking', and 'The door garage door is not opening'. A search bar and pagination controls are also present.

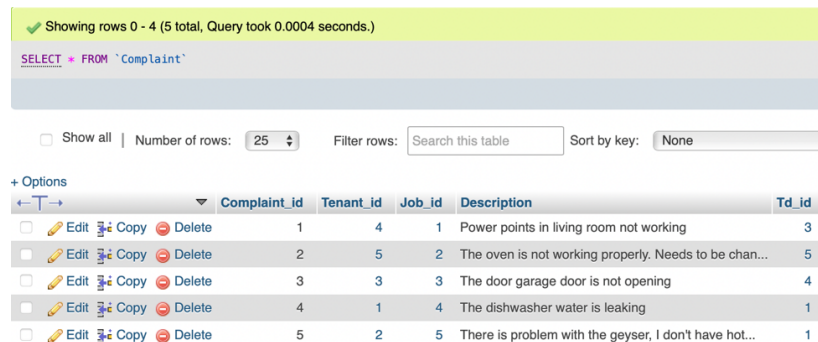
Unit_no	Street_Name	Suburb	State	Postcode	Insurance	Owner_id	Tenant_id	Complaint_id	Description	Td_id
13	Marley Point Road	Suggan Buggan	Victoria	3885	1	3	2	5	There is problem with the geyser, I don't have hot water anymore	1
20	Saggers Road	Jerdacuttup	Western Australia	6346	1	5	1	4	The dishwasher water is leaking	1
28	Bathurst Road	Perthville	New South Wales	2764	1	1	3	3	The door garage door is not opening	4
44	Kogil Street	Gravesend	New South Wales	2401	1	4	5	2	The oven is not working properly. Needs to be changed	5
62	Edgewater Close	Sussex Inlet	Queensland	4001	1	2	4	1	Power points in living room not working	3

Fig 2: Results from join query visible on Landlord's page

2. Update operation:

I implemented the update query on the tradesmen page the action button  allows the tradesmen to edit anything in the complaint. Changes when submitted will be updated in the database and on the tradesmen's main page. The update operation can be used by the tradesmen to update the description of the complaint to show the status, which is then visible to landlords and tenants.

Complaint table:



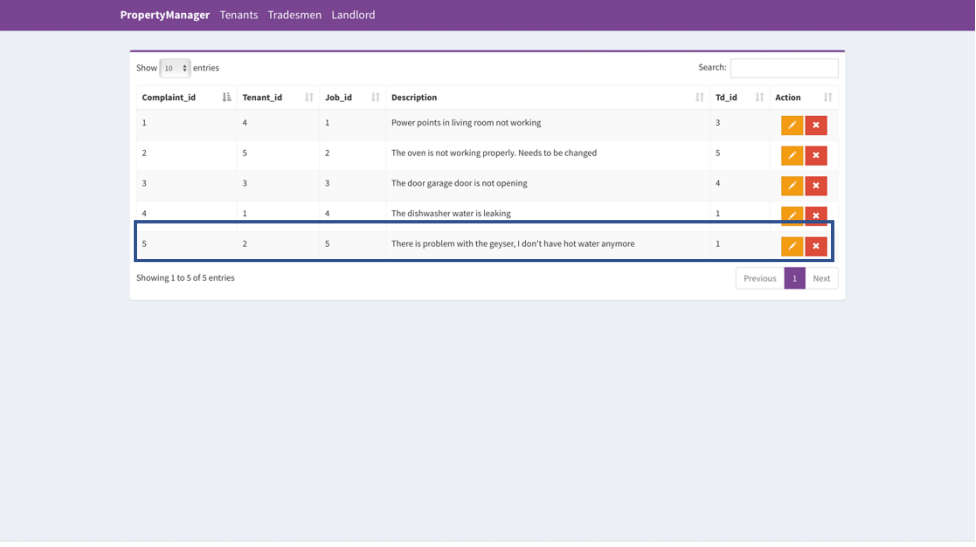
The screenshot shows a database interface with a green status bar indicating 'Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)'. Below this is a SQL query: 'SELECT * FROM `Complaint`'. The table has columns: Complaint_id, Tenant_id, Job_id, Description, and Td_id. Five rows of data are visible, showing details for various complaints such as 'Power points in living room not working', 'The oven is not working properly. Needs to be changed', and 'The door garage door is not opening'. Each row has action buttons for 'Edit', 'Copy', and 'Delete'.

Complaint_id	Tenant_id	Job_id	Description	Td_id
1	4	1	Power points in living room not working	3
2	5	2	The oven is not working properly. Needs to be changed	5
3	3	3	The door garage door is not opening	4
4	1	4	The dishwasher water is leaking	1
5	2	5	There is problem with the geyser, I don't have hot...	1











SQL CODE:

```
UPDATE Complaint set Complaint_id = %s, Tenant_id = %s, Job_id = %s, Description = %s, Td_id = %s where Complaint_id = %s",  
(data['Complaint_id'],data['Tenant_id'],data['Job_id'],data['Description'],data['Td_id'],id,)
```

Pictorial UI:

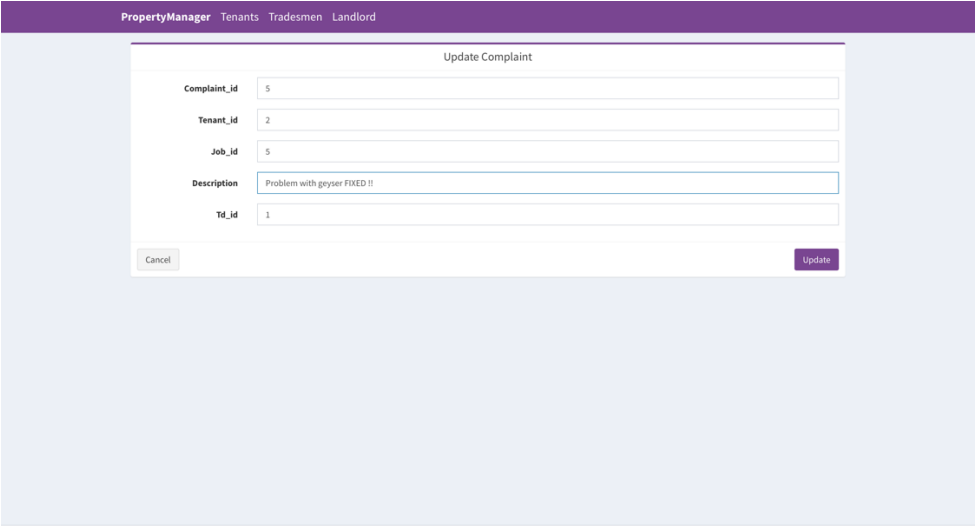


The screenshot shows the 'PropertyManager' interface with a navigation bar containing 'PropertyManager', 'Tenants', 'Tradesmen', and 'Landlord'. Below the navigation bar, there is a table titled 'Show 5 entries' with a search bar. The table has columns: 'Complaint_id', 'Tenant_id', 'Job_id', 'Description', 'Td_id', and 'Action'. The table contains 5 rows of data. The 5th row is highlighted with a blue border. Below the table, it says 'Showing 1 to 5 of 5 entries' and has 'Previous', '1', and 'Next' buttons.

Complaint_id	Tenant_id	Job_id	Description	Td_id	Action
1	4	1	Power points in living room not working	3	 
2	5	2	The oven is not working properly. Needs to be changed	5	 
3	3	3	The door garage door is not opening	4	 
4	1	4	The dishwasher water is leaking	1	 
5	2	5	There is problem with the geyser, I don't have hot water anymore	1	 

Yashita Kumar

Fig 3: Tradesmen's home page shows the result of Complaint table



The screenshot shows the 'Update Complaint' form. It has input fields for 'Complaint_id', 'Tenant_id', 'Job_id', 'Description', and 'Td_id'. The 'Description' field is highlighted with a blue border. Below the fields are 'Cancel' and 'Update' buttons.

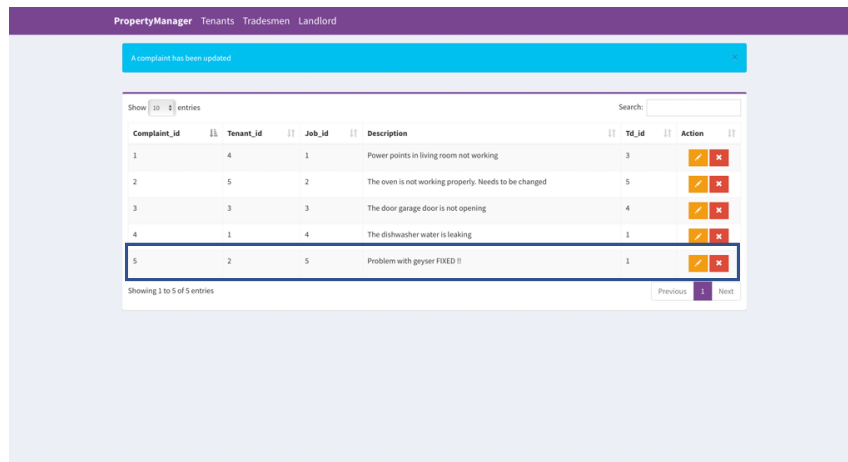
Field	Value
Complaint_id	5
Tenant_id	2
Job_id	5
Description	Problem with geyser FIXED !!
Td_id	1

Yashita Kumar

Fig 4: Update complaint page on Tradesmen portal

In the figure I changed the description of Complaint_id = 5 from “There is problem with geyser, I don’t get hot water anymore” to “Problem with geyser FIXED !!”

The GUI send a pop-up message to show that the complaint has been updated.



Yashita Kumar

Fig 5: UI's response to update query

When the user clicks on update, they get a message informing them that the query has been performed. The update on the complaint description is visible in the tradesmen and landlord portals.

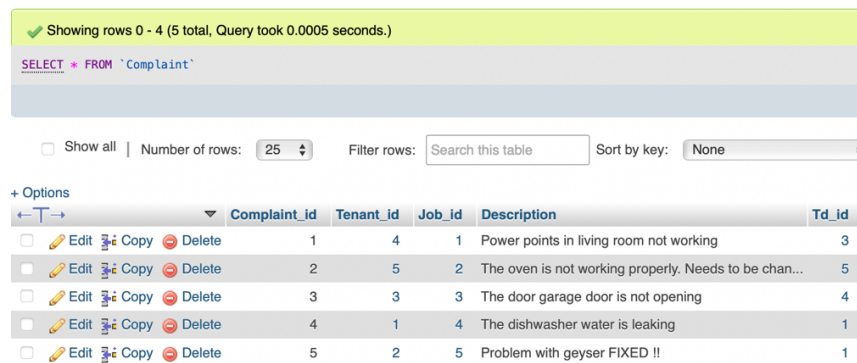



Fig 6: 'Complaint' table after update operation

3. Delete operation with cascade:

I implemented the delete query on the tradesmen page where the action button  allows the tradesmen to delete any complaint. When deleted the complaint gets deleted from the 'Complaints' Table and therefore will no longer be visible on the landlord's page.

Tradesmen can use this functionality to delete a complaint which has been fixed and completed or to delete a complaint which the tenant might have filed by mistake. The cascade constraints have been set in PhpMyAdmin and therefore when a 'complaint' is deleted the corresponding 'Job' details are also deleted.

Complaint table:

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

```
SELECT * FROM `Complaint`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

		Complaint_id	Tenant_id	Job_id	Description	Td_id
<input type="checkbox"/>	Edit Copy Delete	1	4	1	Power points in living room not working	3
<input type="checkbox"/>	Edit Copy Delete	2	5	2	The oven is not working properly. Needs to be chan...	5
<input type="checkbox"/>	Edit Copy Delete	3	3	3	The door garage door is not opening	4
<input type="checkbox"/>	Edit Copy Delete	4	1	4	The dishwasher water is leaking	1
<input type="checkbox"/>	Edit Copy Delete	5	2	5	There is problem with the geyser, I don't have hot...	1

SQL CODE:

DELETE FROM Complaint where Complaint_id = 5;

Pictorial UI:

PropertyManager Tenants Tradesmen Landlord

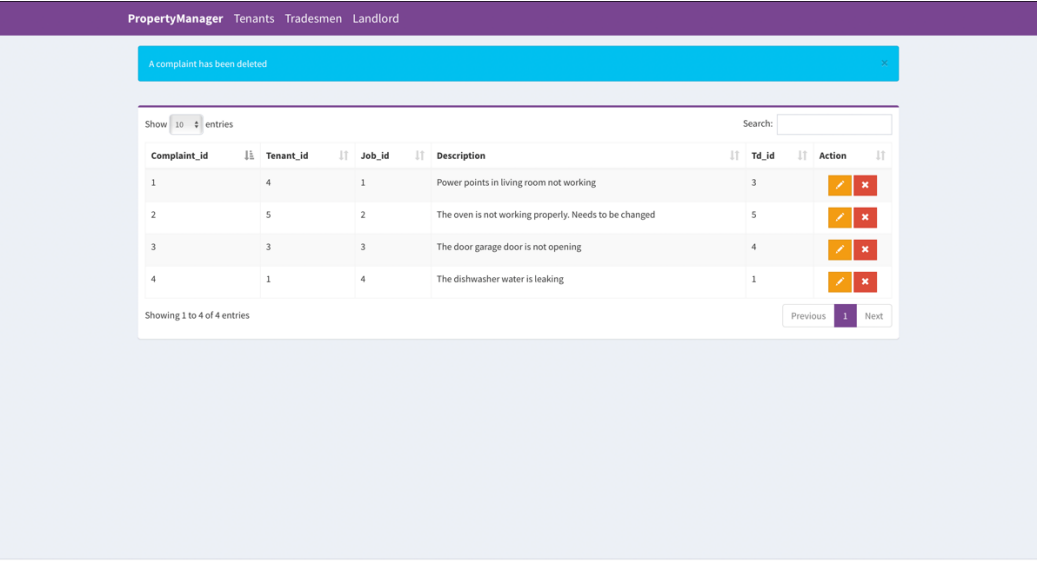
Delete Complaint

Complaint_id	5
Tenant_id	2
Job_id	5
Description	Problem with geyser FIXED !!
Td_id	1

Cancel Delete

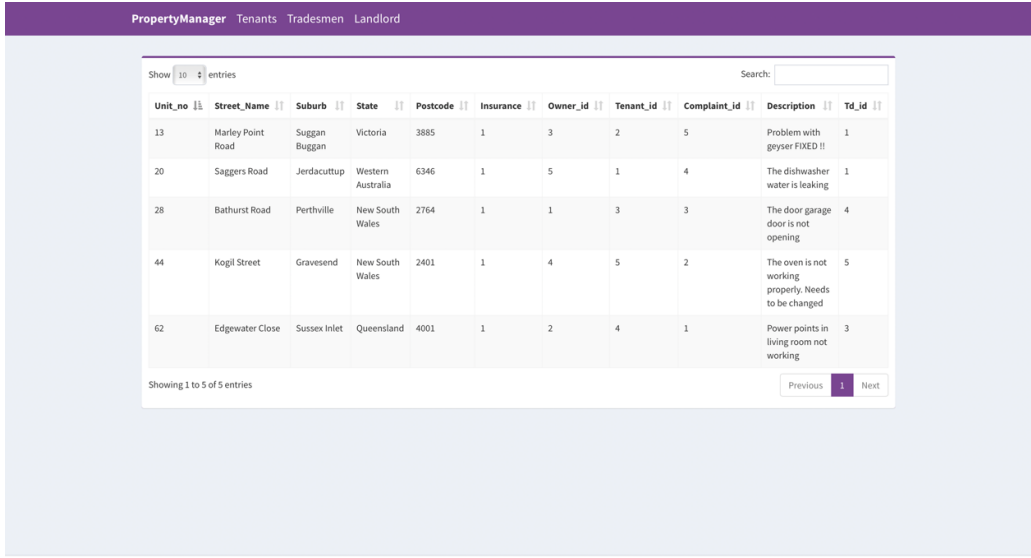
Yashita Kumar

Fig 7: Delete complaint page on the tradesmen portal



Yashita Kumar

Fig 8: UI's response to delete operation



Yashita Kumar

Fig 9: Landlord's page before the delete operation

PropertyManager Tenants Tradesmen Landlord										
Show 10 entries		Search:								
Unit_no	Street_Name	Suburb	State	Postcode	Insurance	Owner_Id	Tenant_Id	Complaint_Id	Description	Td_Id
20	Saggers Road	Jerdacuttup	Western Australia	6346	1	5	1	4	The dishwasher water is leaking	1
28	Bathurst Road	Perthville	New South Wales	2764	1	1	3	3	The door garage door is not opening	4
44	Kogil Street	Gravesend	New South Wales	2401	1	4	5	2	The oven is not working properly. Needs to be changed	5
62	Edgewater Close	Sussex Inlet	Queensland	4001	1	2	4	1	Power points in living room not working	3
Showing 1 to 4 of 4 entries								Previous	1	Next

Yashita Kumar

Fig 10: Landlord's page after the delete operation

4. Aggregation query (functions such as min, max, average or count)

I have implemented the aggregation query to find the average cost of service from the 'Service' table. This will be helpful for the tenants as well as landlords to have an idea about what the average cost of service would be.

Service table:

✓ Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)								
SELECT * FROM `Service`								
<input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table Sort by key: None								
+ Options								
			Job_id	Labour_cost	Total_cost	Start_time	Finish_time	Service_date
			P_id					
<input type="checkbox"/>	Edit	Copy	Delete	1	183	200	18:30:30	19:45:30
				2022-04-08				
<input type="checkbox"/>	Edit	Copy	Delete	2	200	200	10:40:29	10:54:33
				2022-03-17				
<input type="checkbox"/>	Edit	Copy	Delete	3	1000	1300	09:10:00	09:50:00
				2022-05-18				
<input type="checkbox"/>	Edit	Copy	Delete	4	100	126	12:20:00	14:50:00
				2022-03-14				
<input type="checkbox"/>	Edit	Copy	Delete	5	150	160	15:30:03	18:40:03
				2022-04-02				

SQL CODE:

SELECT AVG(Total_cost) as average_cost FROM `Service`

Pictorial UI:



Fig 11: Result from the aggregation query

After performing the above query to find the average cost of service, it can be seen that the average_cost = \$669.4286

5. Aggregation with group-by (aggregated value for each group)

The query I have implemented here would allow the Landlord to see the number of complaints filled by each tenant. I have also applied order by to sort the number of complaints from highest to lowest (descending).

Complaint table:

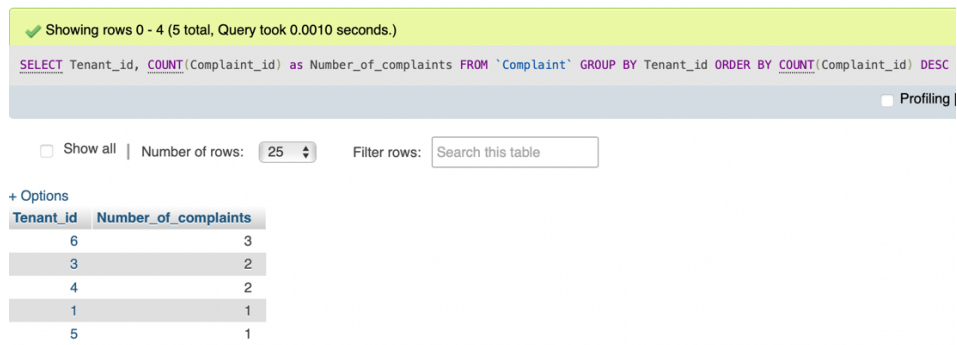
The screenshot shows a database interface with a query editor at the top containing the SQL query: `SELECT * FROM `Complaint``. Below the query editor, there is a status bar indicating 'Showing rows 0 - 8 (9 total, Query took 0.0003 seconds.)'. Below this, there are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (a search box), and 'Sort by key' (set to None). A '+ Options' link is visible. The result table has columns: 'Complaint_id', 'Tenant_id', 'Job_id', 'Description', and 'Td_id'. The table contains 9 rows of data, with the 6th row (Complaint_id 19) highlighted.

	Complaint_id	Tenant_id	Job_id	Description	Td_id
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	1	4	1	Power points in living room not working	3
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	2	5	2	The oven is not working properly. Needs to be chan...	5
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	3	3	3	The door garage door is not opening	4
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	4	1	4	The dishwasher water is leaking	1
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	17	6	5	Table top is broken	6
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	19	6	6	Table top repair	4
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	20	6	7	Kitchen sink is blocked	1
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	21	4	1	Swimming pool filter not working	1
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	22	3	2	Study room's power outlets have stopped working	3

SQL CODE:

```
SELECT Tenant_id, COUNT(Complaint_id) as Number_of_complaints FROM  
`Complaint` GROUP BY Tenant_id ORDER BY COUNT(Complaint_id) DESC
```

Pictorial UI:



The screenshot shows a database query interface. At the top, a green status bar indicates "Showing rows 0 - 4 (5 total, Query took 0.0010 seconds.)". Below this, the SQL query is displayed: `SELECT Tenant_id, COUNT(Complaint_id) as Number_of_complaints FROM `Complaint` GROUP BY Tenant_id ORDER BY COUNT(Complaint_id) DESC`. A "Profiling" button is visible on the right. Below the query, there are controls for "Show all", "Number of rows" (set to 25), and a "Filter rows" search box. The results are shown in a table with two columns: "Tenant_id" and "Number_of_complaints". The data is as follows:

Tenant_id	Number_of_complaints
6	3
3	2
4	2
1	1
5	1

Fig 12: The result from the aggregation with group by query

After performing the above query, it can be seen that `tenant_id = 6` has filed the maximum number of complaints which is equal to 3.

CONCLUSION

The project is still in its initial stages and has a lot of scope to improve, I plan to keep working on it and implement more functionalities for the system. I have a background in programming and have developed various websites therefore I would say that implementing the SQL queries was the easiest part for me, but I have never worked with python and flask to build projects so getting familiar with a new language was fun! I would say I had spent the maximum amount of time coming up with an idea which would be useful in real life and then designing the database.

Time spent on parts of project:

1. Idea: Almost a month, because I had a lot of ideas, and it was difficult to decide which idea I should work on.
2. Database design: Almost 1.5 months because after deciding what I wanted to do I had to thinking about how to implement my idea using an actual system.
3. Implementation: Around 2 weeks because I'm familiar with building websites using SQL database. For the GUI, I used the skeleton structure provided by our tutor.²
4. Report: Personally, it was the least time-consuming task for me. It took me 2 days to complete it.

For me the most interesting part was to learn how to use a new language (flask) to create webpages. The practicals helped me a lot, but I also used various YouTube videos to build my project. There is nothing in particular that I didn't like about this project, it was a fun assignment that we had to work on the entire duration of our semester.

The best way to learn something new is practising it, just studying theory would not help you practice your skills for industry. If someone really wants to learn how to design a database, I would suggest them to come up with ideas and try to implement them and work on live projects.

² https://github.com/xurong-liang/crud_flask_mysql