Aim:

To design and develop an Apartment management web application.

Modules:

Every management system needs the necessary functionality to maintain people's data in a secure manner. In an apartment a large number of people will be present and they might face some issues in the apartment. Every resident's complaint should be managed in such a way that if a resident calls and tells a complaint it must be added to the database or altered if the resident makes any changes to it or it should be deleted once the complaints have been resolved.
So basically, there should be 3 primary modules to store the complaints of the resident.
They are:
1)Create database module
2)Add-Complaint Module
3)Edit-Complaint Module
4)Update-Complaint module
5)Delete-Complaint Module

Use case:

In this project, the main problem statement is to store and manage the issues that people face in the apartment.
Whenever a resident faces an issue, given the management number they will call the apartment management and register their issue. So the management will collect the details like block number, floor number, door number, type of issue they are facing and a detailed description of the issue and will assign a complaint id to them. They make use of this data and add it to the management database and will assign a technician to resolve the problem. After the problem has been resolved, their complaint ticket will be closed and their complaint id will be deleted from their database.

Let's take a detailed use of the modules used:

1)Create database:

- First an empty database is created.
- Then the application must be connected to this database.
- A New table is created in this database with specific columns.

- Necessary information such as complaint id, registered name, resident's block and room number, description about the issue etc. are columns in the table.

- All the details are collected as inputs via add-complaints from which are then added to the database.
- Any insertion, deletion and update regarding the resident's complaint will be reflected in this database.

2)Add-complaint:

- Every new complaint is added to the database only using this add-complaint method.
- Resident's complaint details are collected and stored inside the table in this module.
- This is done by clicking a button called add complaints which will take to a new page that will consist of input boxes of the columns we created.
- Then the "submit" button will store the details to the database.
- This action is done by the backend function called add which is connected to this add complaints page and performs the SQL operation INSERT which inserts the values we filled into the database.
- Now the details of the complaint that was filled will be displayed in the application page.

3)Edit-Complaint:

- In this module, we modify the details of the existing complaints that are registered in the database.
- This module is dependent upon the previous add complaints module because we cannot edit details when there are no details added to the table.
- Resident's details can be modified and saved in the database without altering the other records in the table.
- Every row consists of a "Edit" button and the action of the button leads to a new page with the details of the complaint already filled in so we can add the necessary details we want.

4)Update-Complaint:

- This module is a sequence and dependent on the edit module.

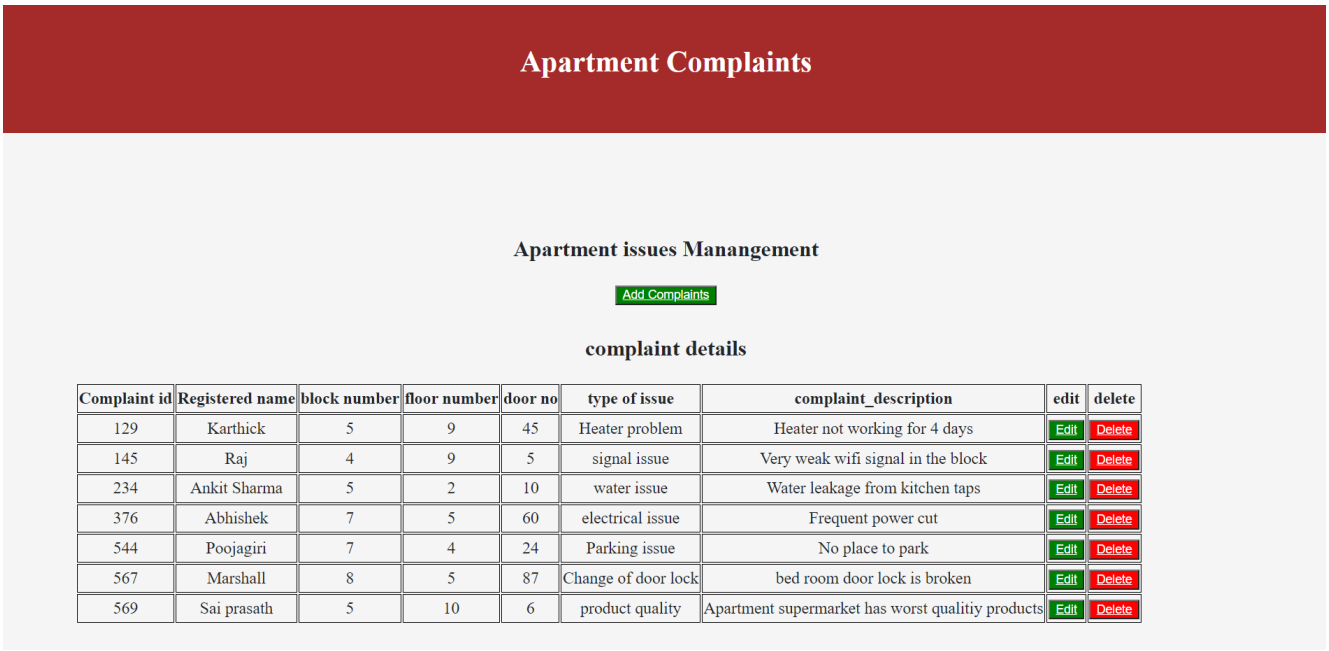
- After editing details, clicking the Update button will make the modifications to the database.
- This action is done by the backend function called edit which is connected to the update button by clicking triggers the SQL operation UPDATE that will modify the details of the complaint where the particular unique complaint id is present.
- Now the updated changes will be modified in the database and can be viewed in the application page.

5)Delete-Complaint:

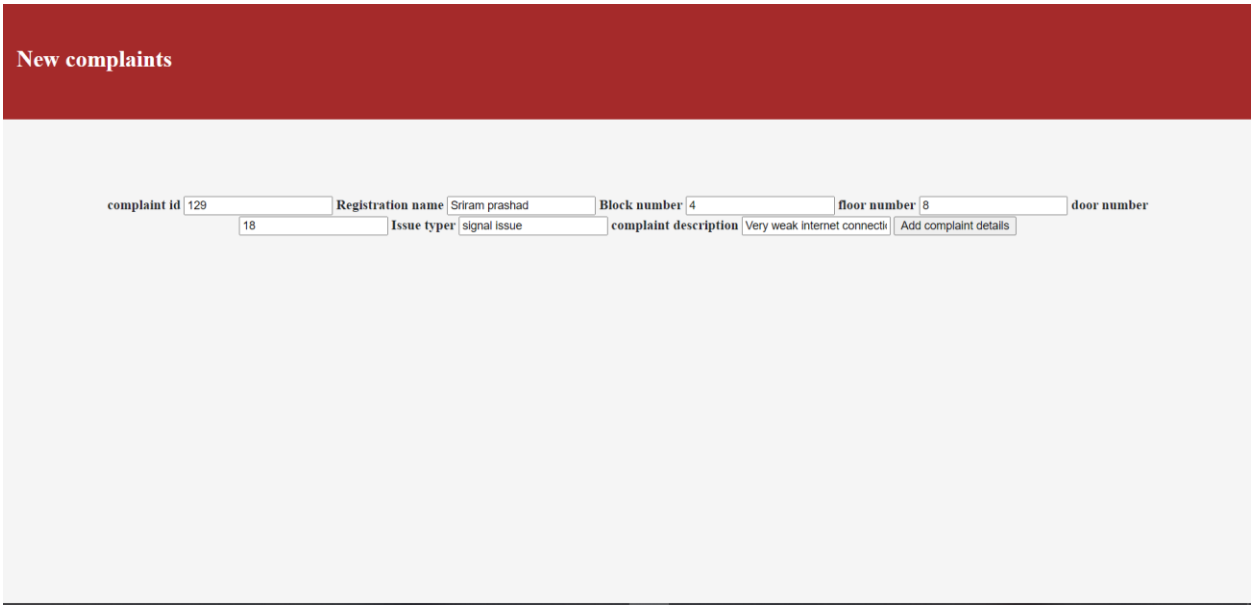
- Records in the table are deleted in this module
- Every row contains a delete button and clicking that will trigger a warning.
- By allowing it the following row is deleted.
- This action is performed by the delete function which performs the SQL operation DELETE that deletes all the details for that particular complaint id.
- The webpage now no longer consists of that record in the database.

Output:

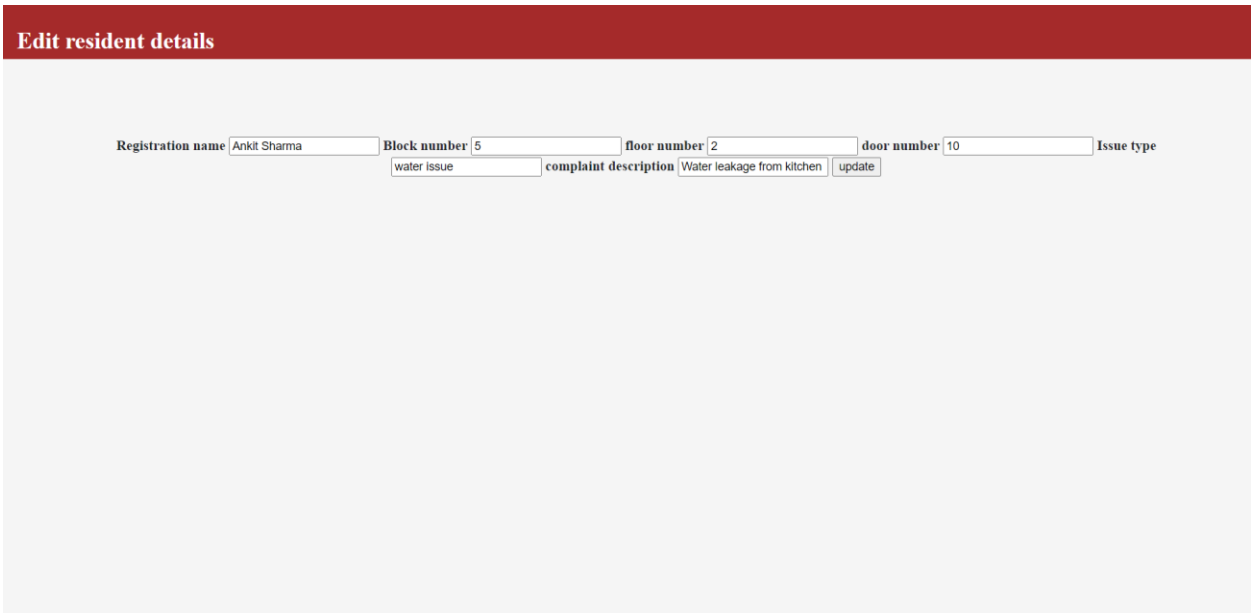
1)User Interface



2)Add complaints:

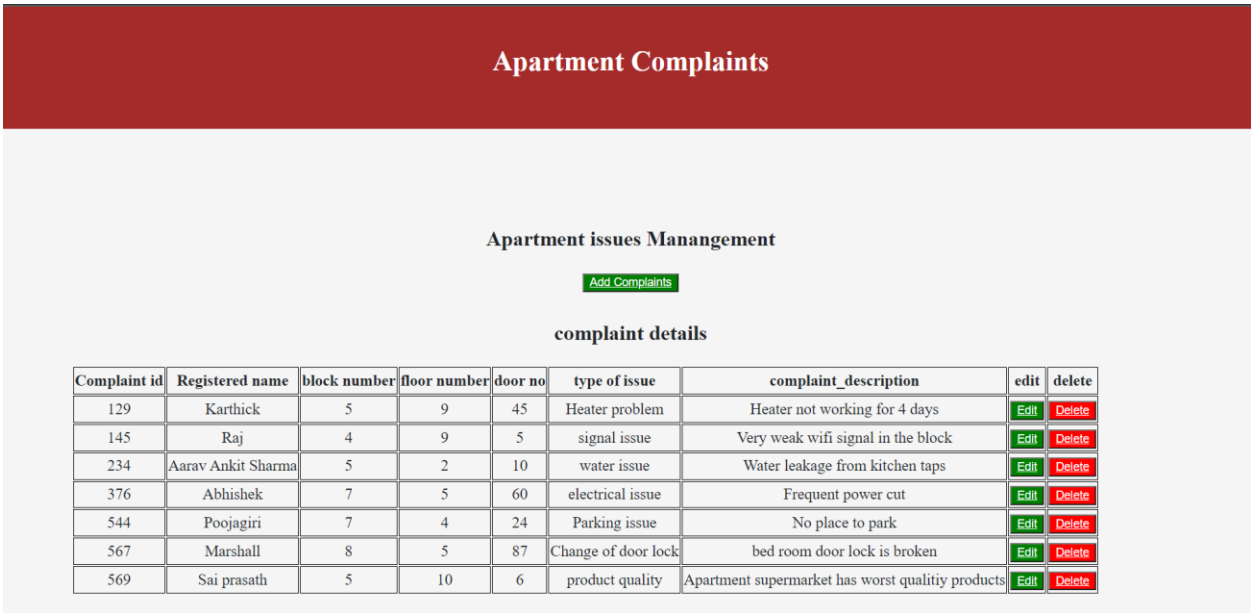


3)Editing complaint details



4)After editing:

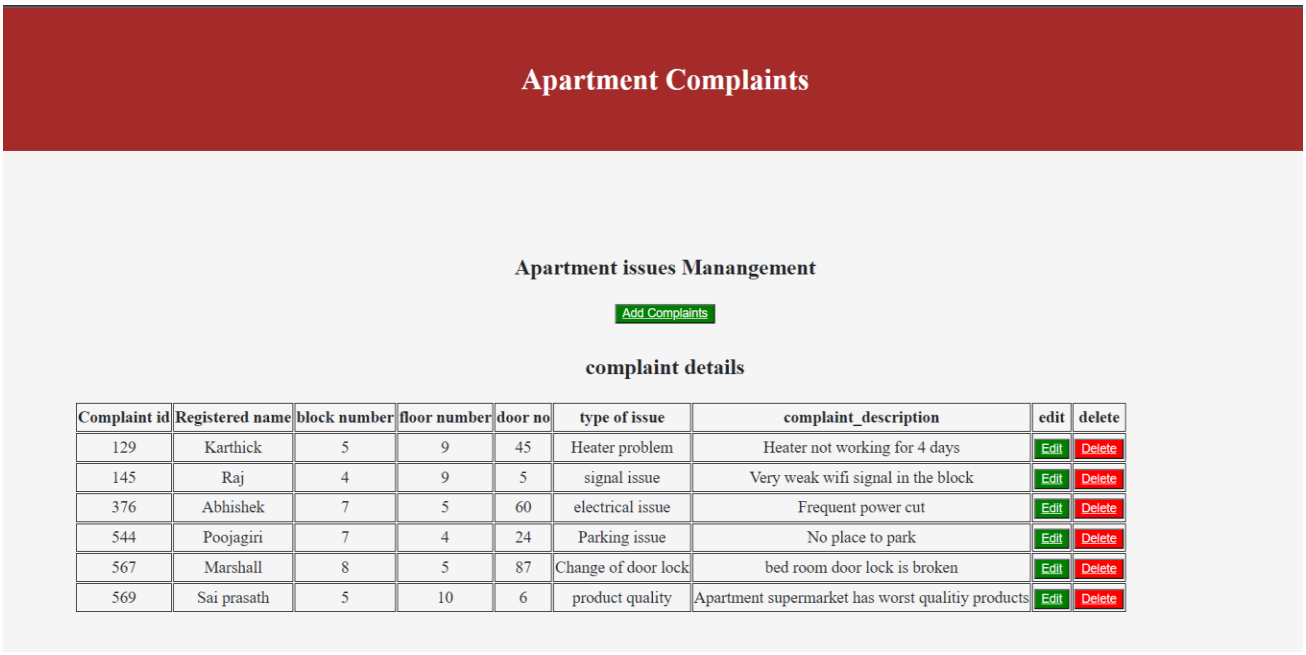
The name of Ankit Sharma has been modified to Aaarav Ankit Sharma



Complaint id	Registered name	block number	floor number	door no	type of issue	complaint_description	edit	delete
129	Karthick	5	9	45	Heater problem	Heater not working for 4 days	Edit	Delete
145	Raj	4	9	5	signal issue	Very weak wifi signal in the block	Edit	Delete
234	Aarav Ankit Sharma	5	2	10	water issue	Water leakage from kitchen taps	Edit	Delete
376	Abhishek	7	5	60	electrical issue	Frequent power cut	Edit	Delete
544	Poojagiri	7	4	24	Parking issue	No place to park	Edit	Delete
567	Marshall	8	5	87	Change of door lock	bed room door lock is broken	Edit	Delete
569	Sai prasath	5	10	6	product quality	Apartment supermarket has worst qualitiy products	Edit	Delete

5)Deleting complaint id 234

Here Aaarav Ankit Sharma row has been deleted using delete button



Apartment Complaints

Apartment issues Manangement

Add Complaints

complaint details

Complaint id	Registered name	block number	floor number	door no	type of issue	complaint_description	edit	delete
129	Karthick	5	9	45	Heater problem	Heater not working for 4 days	Edit	Delete
145	Raj	4	9	5	signal issue	Very weak wifi signal in the block	Edit	Delete
376	Abhishek	7	5	60	electrical issue	Frequent power cut	Edit	Delete
544	Poojagiri	7	4	24	Parking issue	No place to park	Edit	Delete
567	Marshall	8	5	87	Change of door lock	bed room door lock is broken	Edit	Delete
569	Sai prasath	5	10	6	product quality	Apartment supermarket has worst qualitiy products	Edit	Delete

Result:

An apartment management application has been successfully designed and implemented.