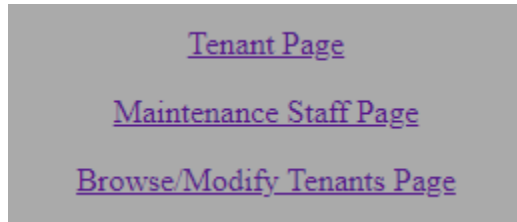


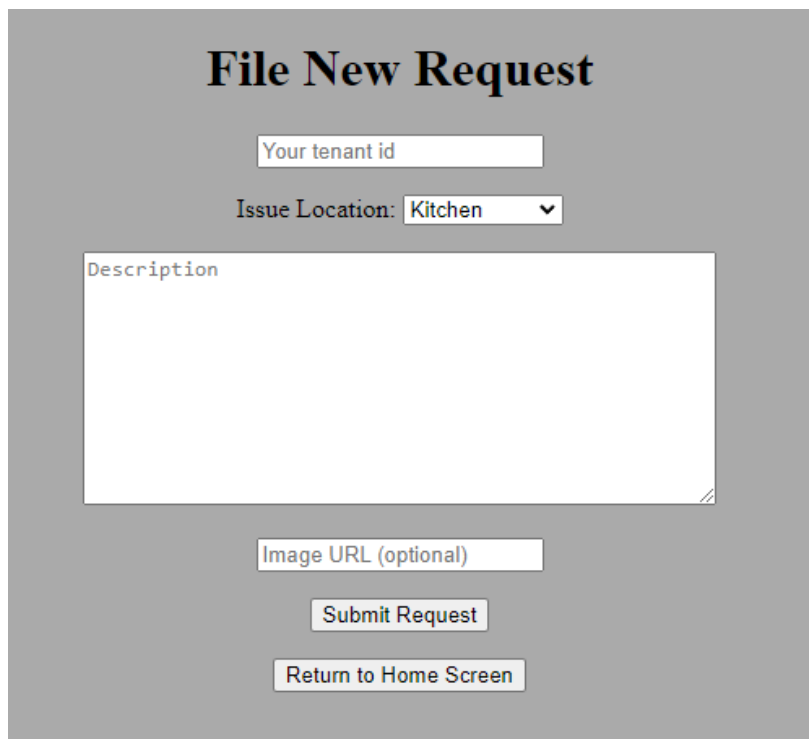
Casey Detwiler
cjd6058@psu.edu
CMPSC 487W
Project 3

Upon visiting localhost:8080, user is greeted by the Home Page:



In the full implementation, this page would be replaced by a login screen and automatically redirect the user to the correct page based on their login information.

Clicking "Tenant Page" directs the user to the Request Submission Page:

A screenshot of a web form titled "File New Request" in bold black text. The form has a grey background. It includes a text input field for "Your tenant id", a dropdown menu for "Issue Location" with "Kitchen" selected, a large text area for "Description", an optional text input field for "Image URL (optional)", a "Submit Request" button, and a "Return to Home Screen" button.

Here the user inputs the requested data, and clicks submit. The photo URL field is only for an external URL to a photo; the database does not store the photos themselves. If they hit submit with a non-numeric value in the ID field or input an ID that does not exist in the database, the request is not inserted into the database. If a valid ID is in the input field upon hitting submit, the request is inserted into the database and will be visible to the maintenance. In the full implementation the user would not be allowed to input their room number since it would already be known as soon as they log in.

If the user clicks the “Maintenance Staff Page” on the home screen, they are brought to the Browse Requests Page:

Browse Requests

Sort by...

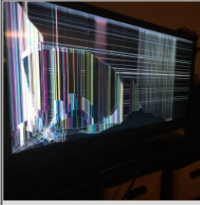
Apartment Number:

Issue Location:

☐ Status

Starting on: Ending on:

Search

Request ID	Image	Apartment Number	Location	Description	Date	Status	
1	(None)	100A	Kitchen	Sink is clogged.	2023-11-05	Pending	Close
5		245B	Living Room	TV is broken!	2023-11-16	Pending	Close
7	(None)	100A	Balcony	Broke a potted plant	2023-11-23	Pending	Close

Return to Home Screen

(If the user did not provide a URL for a photo, (None) is shown where the image would be.)

Here, the user can browse the requests that have been submitted by the tenants. At the top of the page there are several filters, and they can be combined for more complex queries. Clicking the “Status” Check Box reveals two radio buttons to select to only include pending or closed requests, like so:

Browse Requests

Sort by...

Apartment Number:

Issue Location:

☒ Status ☐ Pending ☐ Closed

Starting on: Ending on:

Search

Pushing enter with the Apartment Number Box selected or pushing the Search Button will execute a query with the information provided. Here are the results with “11/01/2023” for the start date, “Living Room” selected from the dropdown menu, and status set to “Pending.”:

Browse Requests

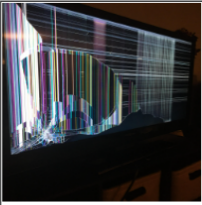
Sort by...

Apartment Number:

Issue Location:

☐ Status

Starting on: Ending on:

Request ID	Image	Apartment Number	Location	Description	Date	Status	
5		245B	Living Room	TV is broken!	2023-11-16	Pending	<input type="button" value="Close"/>

Clicking the Close Button for any request will close it, and refresh the page with the default order (by Request ID in ascending order). The following screenshot shows what would happen if the Close Button was pressed in the previous screen shot:

Browse Requests

Sort by...

Apartment Number:

Issue Location:

☐ Status

Starting on: Ending on:

Request ID	Image	Apartment Number	Location	Description	Date	Status	
1	(None)	100A	Kitchen	Sink is clogged.	2023-11-05	Pending	<input type="button" value="Close"/>
5		245B	Living Room	TV is broken!	2023-11-16	Closed	
7	(None)	100A	Balcony	Broke a potted plant	2023-11-23	Pending	<input type="button" value="Close"/>

If the query the user inputs has no results, the table is replaced with an error message:

Browse Requests

Sort by...

Apartment Number:

Issue Location:

All

☐ Status

Starting on: Ending on:

Search

No requests found!

Return to Home Screen

If the user clicks on “Browse/Modify Tenants Page” on the Home Page, they are brought to the Browse Tenants Page:

Browse Tenants

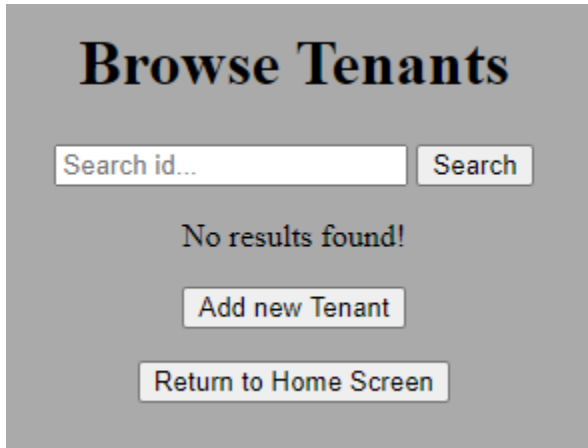
Tenant ID	Name	Apartment Number	Email	Phone Number	Check In Date	Check Out Date	
1	Andy Anderson	100A	aa@gmail.com	123-456-7890	2022-05-21	2024-08-25	<input type="button" value="Modify"/>
3	Bob Barker	245B	bb@yahoo.com	352-633-2355	2023-12-27	2024-08-01	<input type="button" value="Modify"/>

An ID field is provided if the user wishes to find a single tenant by their ID. The following screenshot was taken after “3” was input in the ID Field:

Browse Tenants

Tenant ID	Name	Apartment Number	Email	Phone Number	Check In Date	Check Out Date	
3	Bob Barker	245B	bb@yahoo.com	352-633-2355	2023-12-27	2024-08-01	<input type="button" value="Modify"/>

If the user inputs an invalid ID, the table is replaced with an error message:

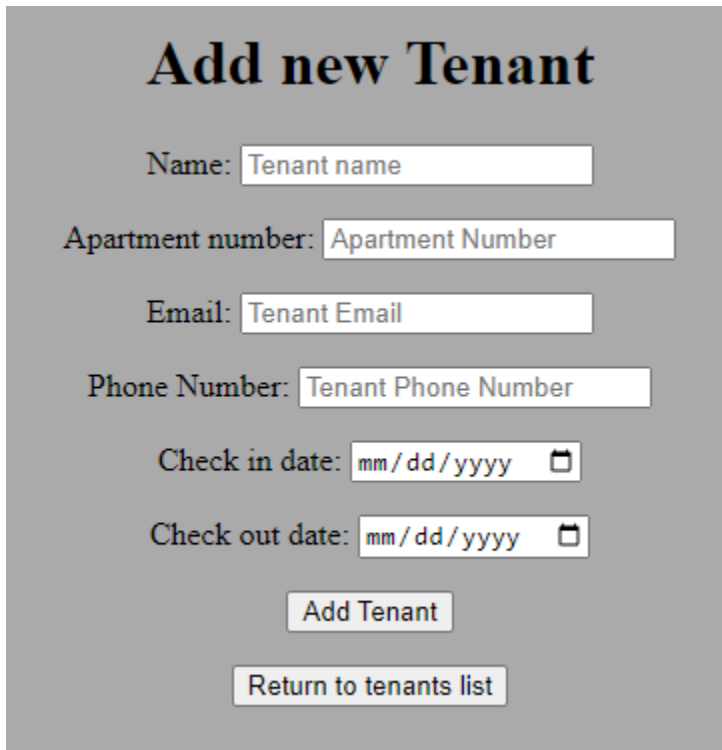


Browse Tenants

Search id...

No results found!

If the user clicks the Add New Tenant Button, they are brought to the Add New Tenant Page:



Add new Tenant

Name:

Apartment number:

Email:

Phone Number:

Check in date:

Check out date:

Here, the user inputs the respective information of the tenant and clicks the Add Tenant Button. The tenant's ID is automatically generated. There are no checks to make sure the check in date is before the check out date, or to make sure that both fields are in the future.


Add new Tenant


Name:

Apartment number:

Email:

Phone Number:

Check in date: 

Check out date: 

After hitting the Add Tenant Button, the tenant is inserted into the database and the user is brought back to the Browse Tenants Page:

Browse Tenants

Tenant ID	Name	Apartment Number	Email	Phone Number	Check In Date	Check Out Date	
1	Andy Anderson	100A	aa@gmail.com	123-456-7890	2022-05-21	2024-08-25	<input type="button" value="Modify"/>
3	Bob Barker	245B	bb@yahoo.com	352-633-2355	2023-12-27	2024-08-01	<input type="button" value="Modify"/>
4	Dave Dolchester	346D	dd@aol.com	4242352353	2023-11-30	2023-12-07	<input type="button" value="Modify"/>

If the user clicks any of the Modify Buttons on the right side of the table, they are brought to the Modify Tenant Page, which is the same as the Add Tenant Page, but with the fields already filled out and a couple additional buttons:


Modify Tenant


Name:

Apartment number:

Email:

Phone Number:

Check in date: 

Check out date: 

If the user clicks the Delete Tenant Button, the tenant is deleted and the user is returned to the Browse Tenants Page. Requests from the deleted tenant ARE NOT deleted from the Requests Table, as there may still be issues with the room that have not been resolved.

If the user clicks the Return to Tenants List Button, they are returned to the Browse Tenants Page, and their changes to the tenant are not saved.

If the user clicks the commit changes, the Tenants Table is updated with the new information, and the user is returned to the Browse Tenants Page:

Modify Tenant


Delete Tenant


Name:

Apartment number:

Email:

Phone Number:

Check in date: 

Check out date: 

Commit changes

Return to tenants list

After clicking “Commit Changes”:

Browse Tenants

Tenant ID	Name	Apartment Number	Email	Phone Number	Check In Date	Check Out Date	
1	Andy Anderson the Third	159A	aaaa@gmail.com	123-456-9999	2022-05-21	2024-08-25	<input type="button" value="Modify"/>
3	Bob Barker	245B	bb@yahoo.com	352-633-2355	2023-12-27	2024-08-01	<input type="button" value="Modify"/>
4	Dave Dolchester	346D	dd@aol.com	4242352353	2023-11-30	2023-12-07	<input type="button" value="Modify"/>

Add new Tenant

Return to Home Screen