Campus Maintenance and Management System

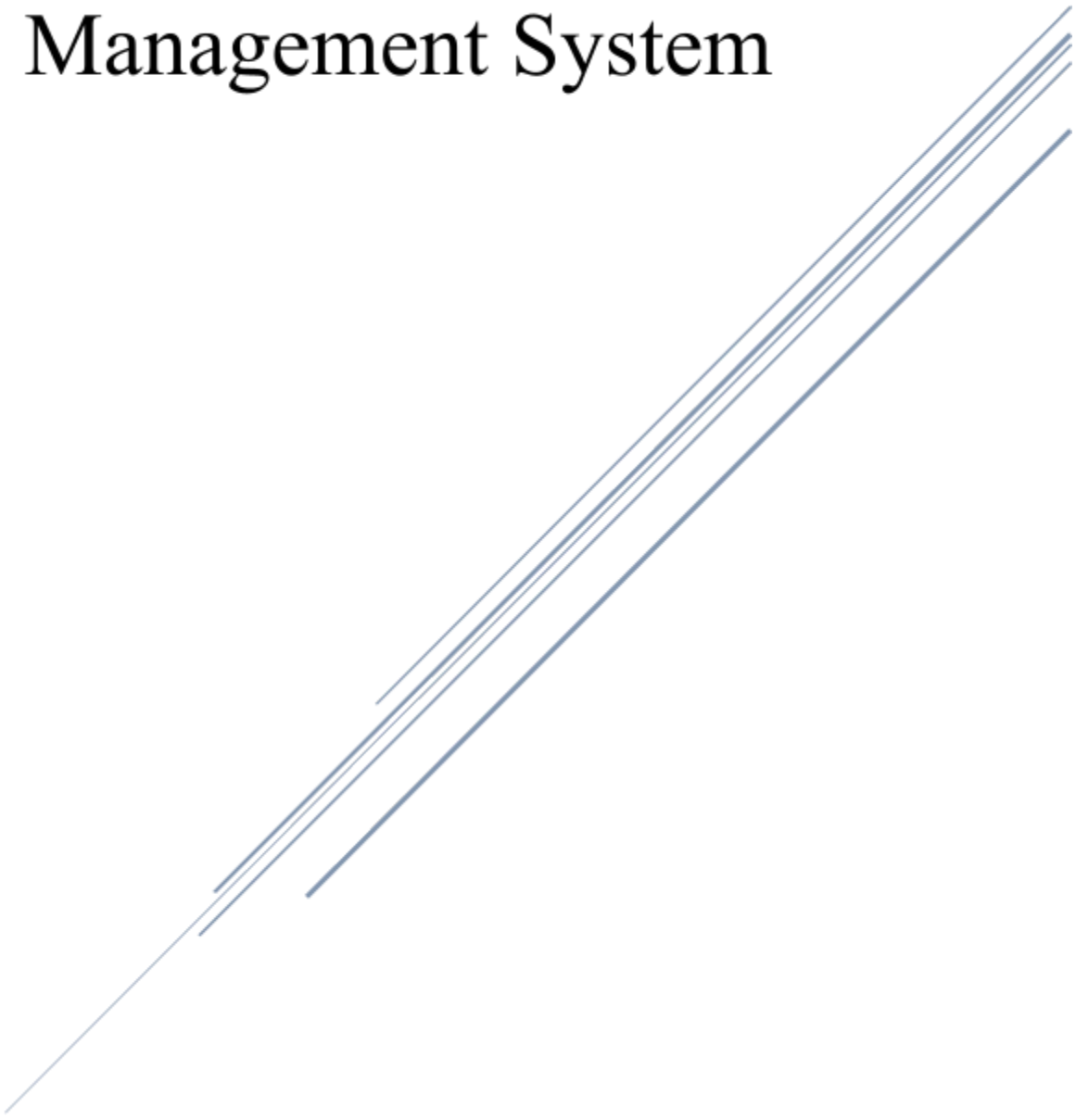


Table of Contents

1. Introduction..... 1
2.Requirement Analysis..... 3
  2.1 Software Requirements..... 3
  2.2 Hardware Requirements..... 3
  2.3 Functional Requirements..... 3
  2.4 Non-Functional Requirements..... 3
3. Programing languages and tools..... 4
  3.1 Frontend technologies..... 4
  3.2 backend technologies..... 4
  3.3 Framework..... 4
4. Database implementation..... 5
  4.1 Entity Relationship Diagram..... 5
  4.2 Database design..... 6
5. Testing..... 8
  5.1 Unit Testing..... 8
  5.2 System Testing..... 10

1. Introduction

1.1. Project Description: (Background, Objectives, Brief Introduction)

Background: The technicians check building systems; also, these systems include heating, cooling, plumbing, and electrical equipment. Paper records this information then spreadsheets receive it. It is hard for one to see problems or trends quickly, mistakes can readily occur, and this also takes extra time.

Objectives:

1) You can use a digital system for replacement of paper log sheets.
2) Allow technicians to record data directly on tablets when they do their rounds.
3) Give administrator tools for easy management of buildings, data categories, and users.
4) Creation of reports helps the admin track building performance. Spotting issues early helps too.
5) Reduce paper use to make the process environment friendly. Speed as well as accuracy should also improve by way of this change.

Brief Introduction: Technicians can collect data in real time using a digital maintenance tracking app developed by this project. Administrators can then analyze building systems with the app to manage them much more effectively. The system will save time, reduce errors, and provide a clear view into building health, and this leads to better decisions and smoother operations across campus as it moves from paper to a digital platform.

1.2. Project Scope: (Main functions included in the system)

The project CMMS, a digital tool that is going to replace the current paper-based process. The goal is to make the process simple, to raise a concern regarding a particular problem in the block that makes the admin address the concern and resolve it by sending it to the respective department for the maintenance, this helps in faster maintenance, more accurately. This would be a place for both admins and technicians to enter and use data to make faster and smart decisions to work on a problem. As the final stage to resolve an issue, the admin will be able to generate the reports, which makes the work more efficient.

Digital Data Collection: Instead of carrying, writing, and re-writing if there is a mistake in the entry sheet, this project makes it easier. Technicians will be able to record the data directly in the system while they are on the grounds. This majorly reduces mistakes that often happen with manual paperwork.

System and User Management: Administrators will have complete control of buildings, equipment, data that is entered into the system and user accounts. Different levels of access can be provided for the users depending on their job roles like technicians, supervisors, and guest access for the temporary staff. The level of access ensures the data safety

Performance Monitoring and Reporting: This function would be the key to perfect building maintenance. As part of this function that system would be able to create dashboards to show progress each day, reports of raised and resolved concerns, building and equipment performance. This makes it easier to track, spot, address, act, and resolve issues. This improves sustainability by improving speed and accuracy compared to the traditional approach.

2.Requirement Analysis

2.1 Software Requirements

- Web browser
- Stable network connection

2.2 Hardware Requirements

- Device with internet access (desktop, laptop, or tablet)
- Minimum 4 GB RAM
- Tablet or touchscreen support for technician field data entry.

2.3 Functional Requirements

2.3.1. User Authentication

- The system shall allow users to log in securely.
- The system shall restrict access based on user roles (Administrator, Technician).

2.3.2. Administrator Functionalities

- Manage user accounts (add, update, and deactivate users
- Manage building information (add, update, and deactivate buildings).
- Define data categories for each building (add, update, and deactivate data categories).
- Generate maintenance reports and graphs (bi-weekly).

2.3.3 Technician Functionalities

- Enter and edit building-specific data forms.
- Submit observations or concerns during regular rounds.

2.4 Non-Functional Requirements

2.4.1 Usability

- The interface shall be simple, clear, and easy to navigate for all users.

- The design shall support quick access for both administrators and technicians.

2.4.2 Performance

- The system shall support concurrent access by multiple users without delay.
- Reports and graphs shall be generated within seconds.

2.4.3 Security

- The system shall enforce role-based authorization and authentication.
- The system shall protect all users and build data according to university IT policies.

2.4.4 Operational

- The system shall be accessible on both desktop and tablet devices.
- The system shall support real-time data entry for technicians.

2.4.5 Legal and Policy Compliance

- The system shall comply with Northwest Missouri State University's IT and data protection policies.

3. Programing languages and tools

3.1 Frontend technologies

HTML, CSS, JavaScript, JavaScript validation, Visual Studio code.

3.2 backend technologies

Java, MS SQL, DB viewer, Spring boot, Eclipse

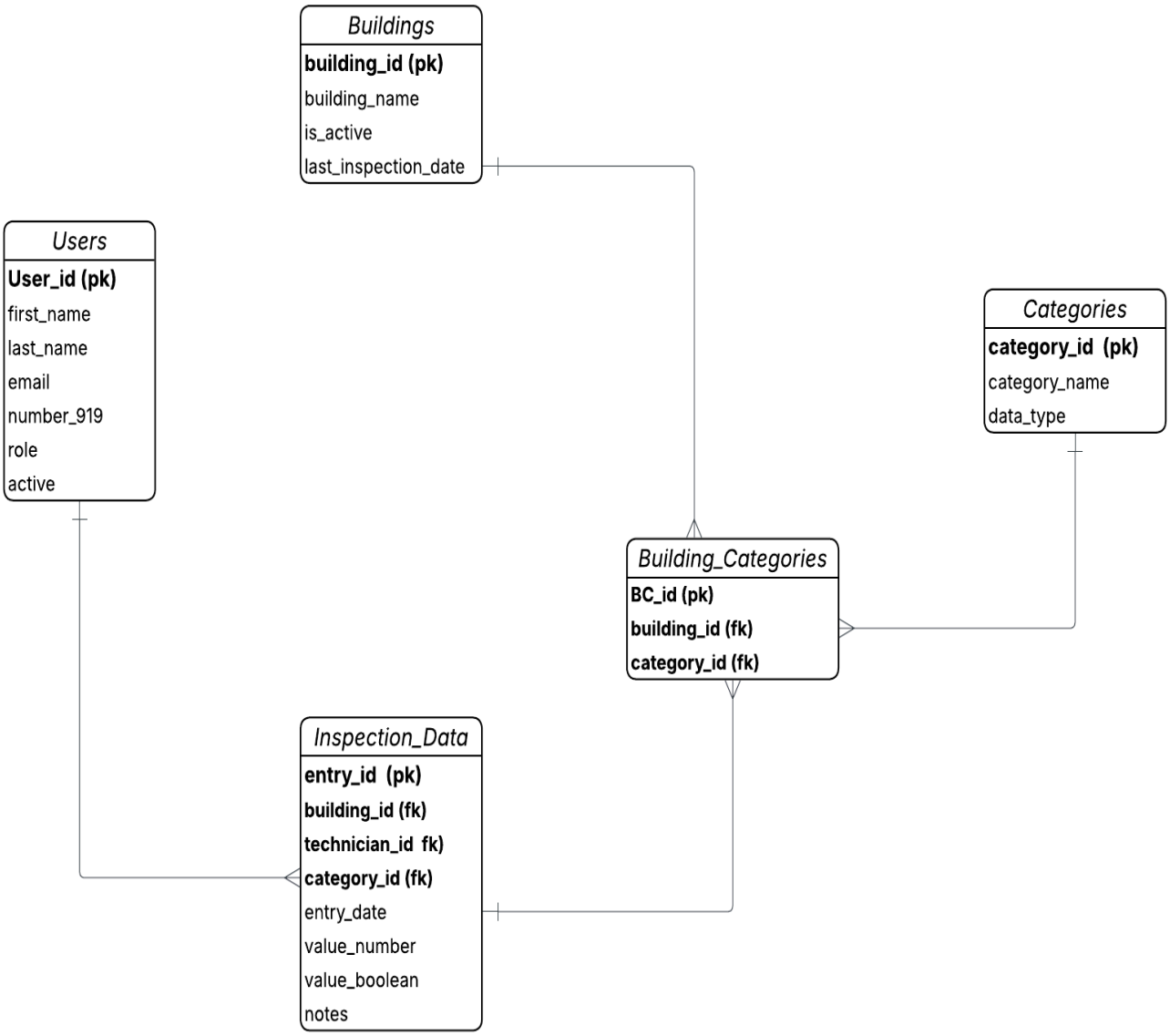
3.3 Framework

We are using Spring Boot for our backend and JavaScript validation for our frontend. Spring Boot helps us easily build and connect the backend parts of our project using Java. It also works smoothly with our MySQL database and saves us time by handling many setup tasks automatically.

On the frontend, JavaScript validation helps us check user inputs before sending them to the backend. This makes sure that the data entered is correct and keeps our system clean and error-free. Together, Spring Boot and JavaScript validation help our frontend and backend work well together and make the project run more smoothly.

4. Database implementation

4.1 Entity Relationship Diagram



4.2 Database design

1. **User Data**

	User	Data Type
PK	User_id	Int
	first_name	Text
	last_name	Text
	email	Text
	Number_919	Int
	Role	Text
	is_active	Boolean

2. **Inspection_Data**

	Inspection_Data	Data Type
PK	entry_id	Int
FK	building_id	Int
Fk	category_id	Int
FK	technician_id	Int
	entry_data	Datetime
	value	Text/Boolean/ Int
	is_active	Boolean

3. **Building_Categoires**

	Building_Categories	Data Type
PK	BC_id	Int
FK	building_id	Int
Fk	category_id	Int

4. Buildings

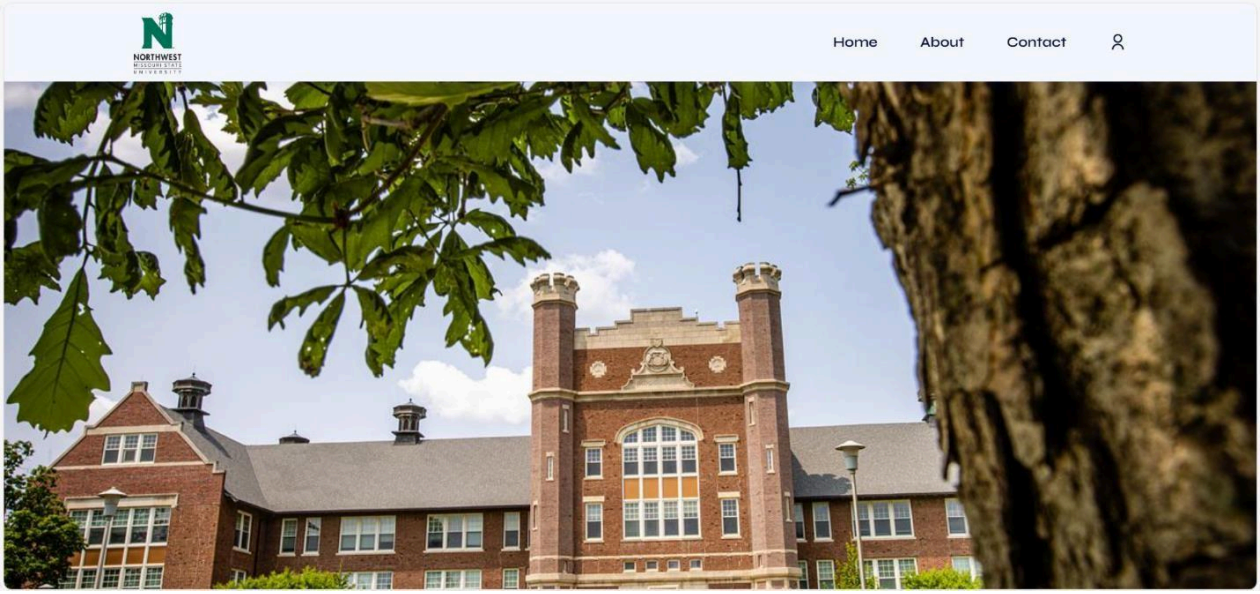
	Building	Data Type
PK	building_id	Int
	Building_name	Int
	Is_active	Boolean
	Last_inspection_date	datetime

5. Categories

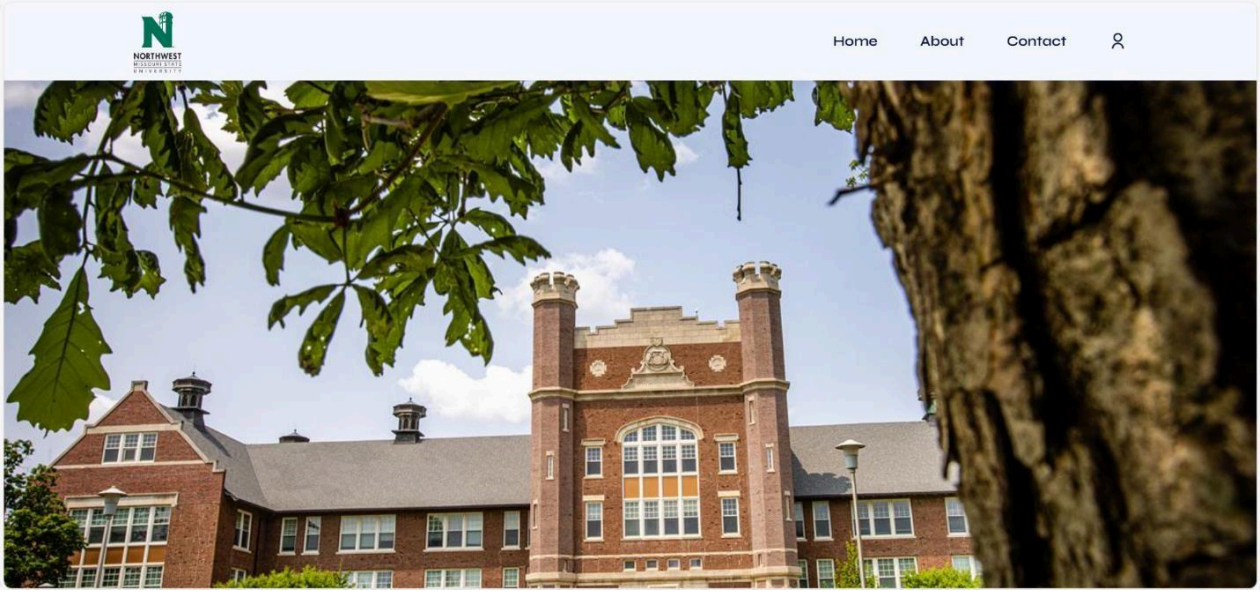
	Categories	Data Type
PK	category_id	Int
	Category_name	Text
	Data_type	Text/Boolean/Int

5. Testing
5.1 Unit Testing

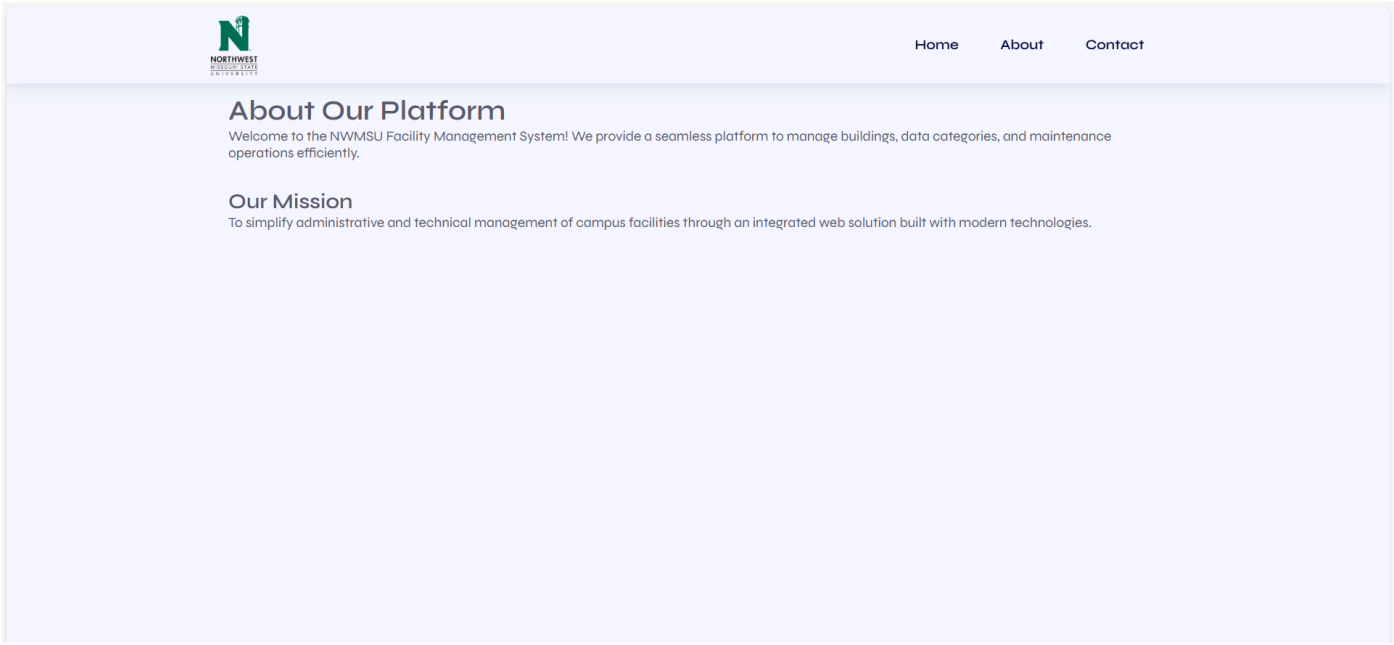
- This is our **Main Dashboard**. It has a Login icon, a Home button, and About and Contact options, making it easy for users to move around and get basic information.



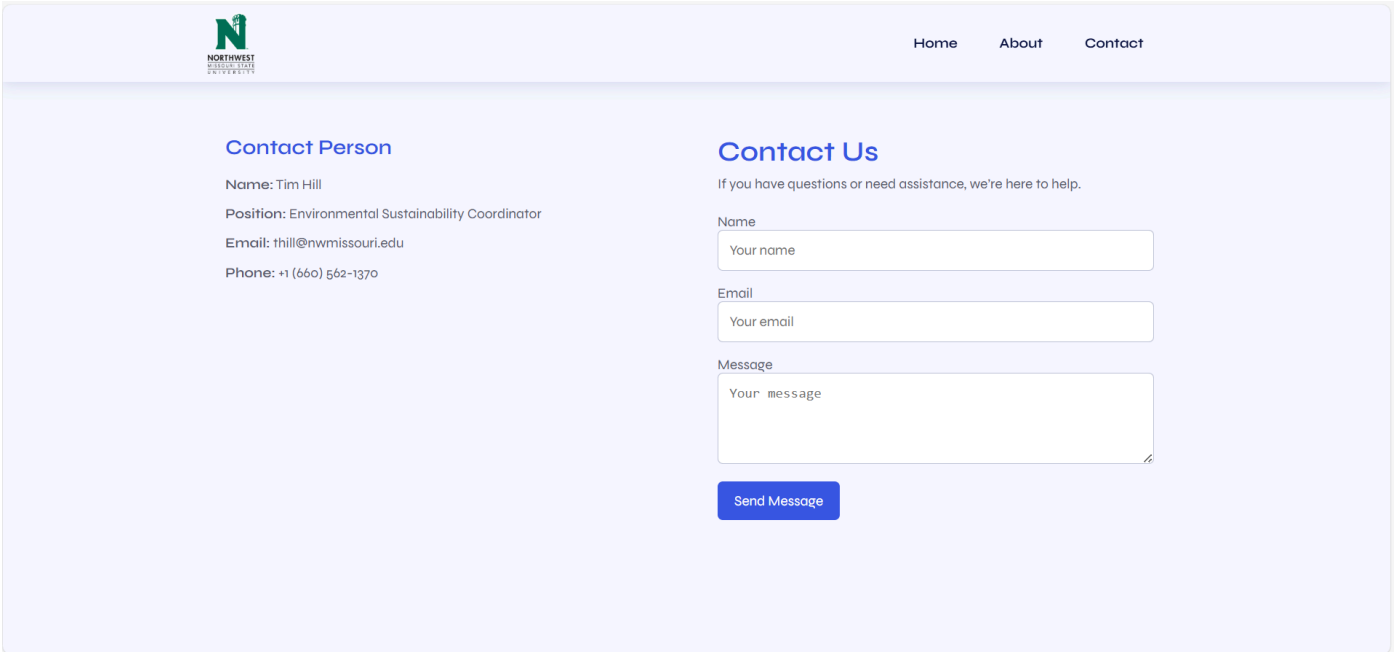
- The **Home** button reloads or updates the current page



- The **About** section gives information about the platform, it explains what the system does, why it was created, and the mission behind it.

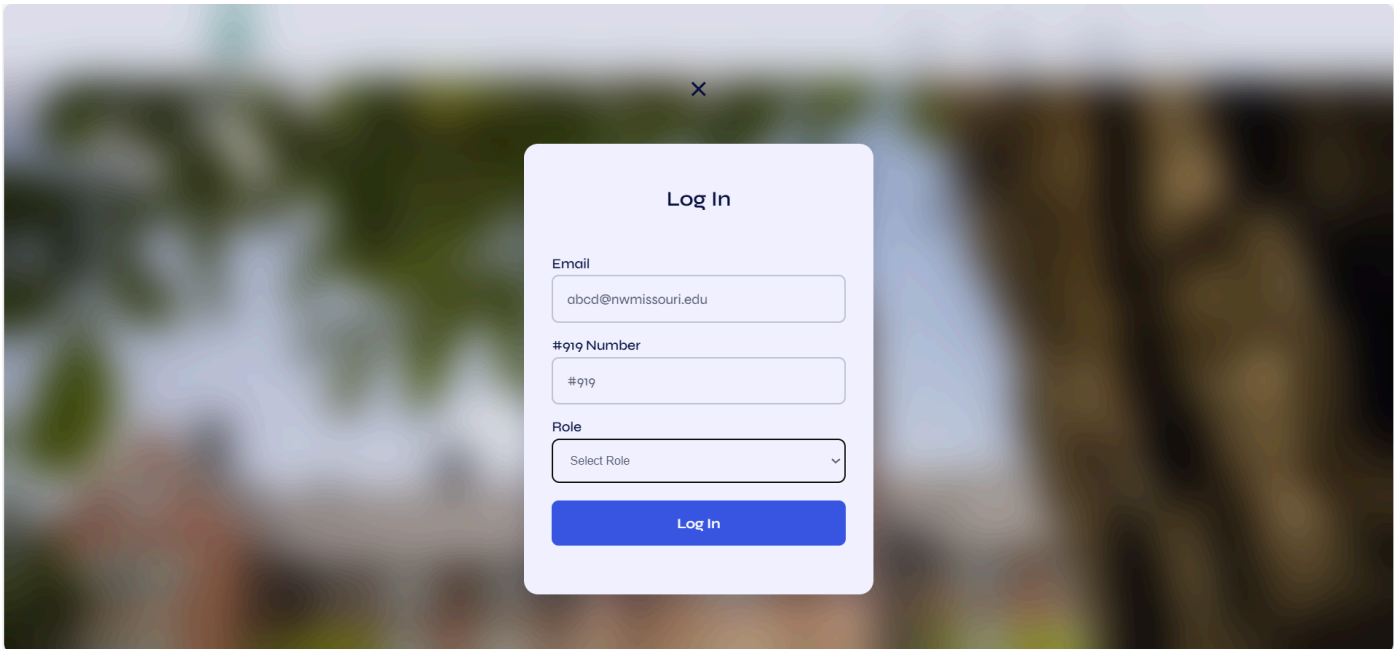


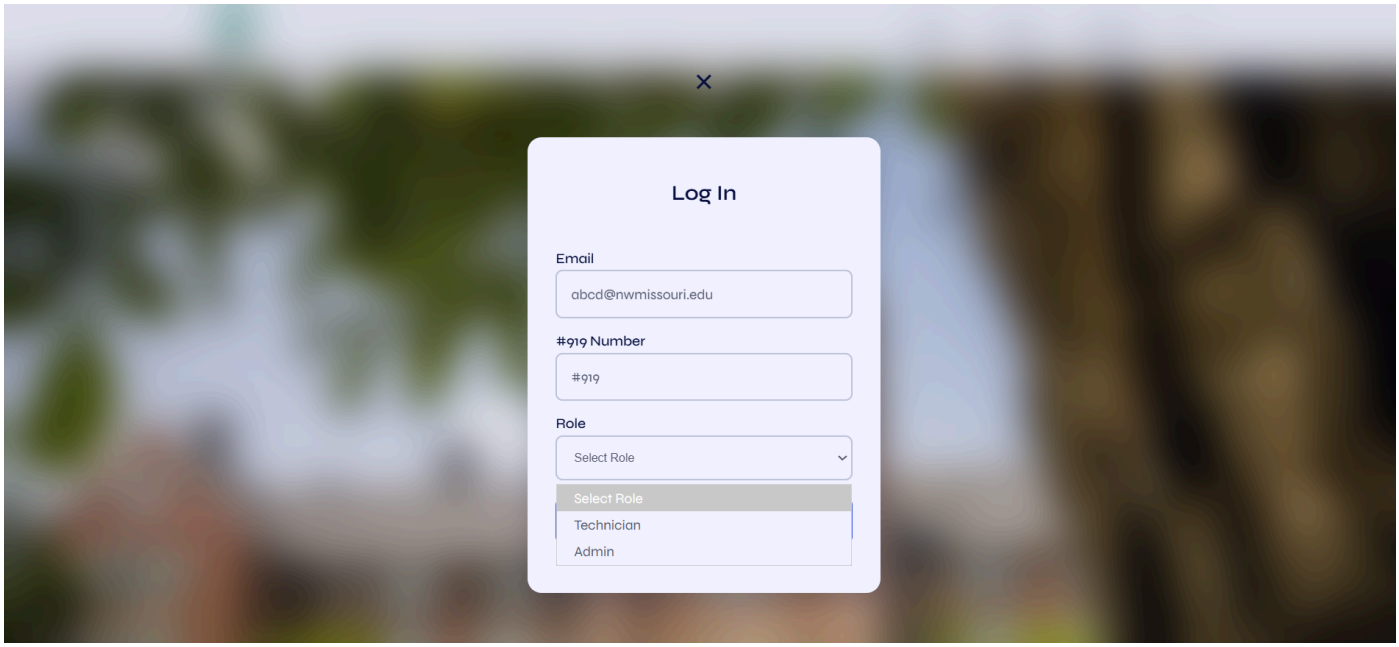
- The **Contact** section provides the client's details such as email or support information. Users can use this page to ask questions or send any queries if they need help.



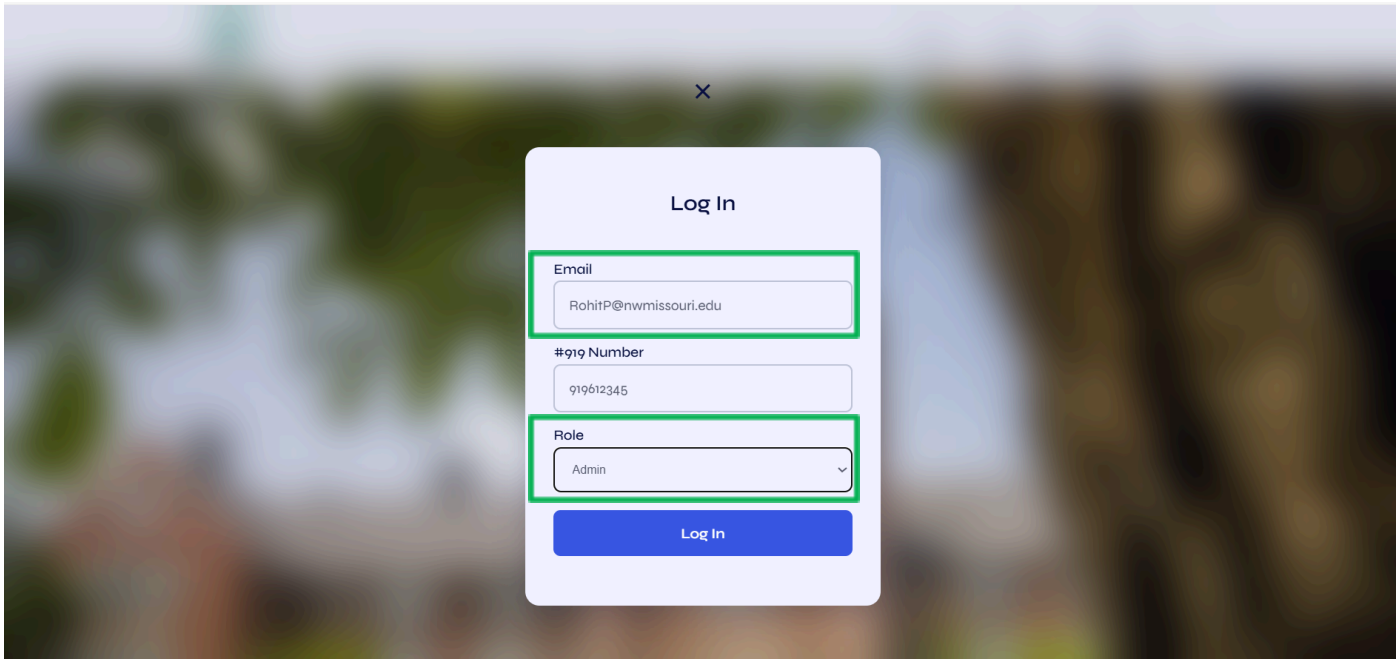
□ User Authentication:

- In the **Login** page only users with a **Bearcat email ID, a #919 number, and a designated role (Admin or Technician)** can access the system. This ensures access control specific to Northwest Missouri State University.

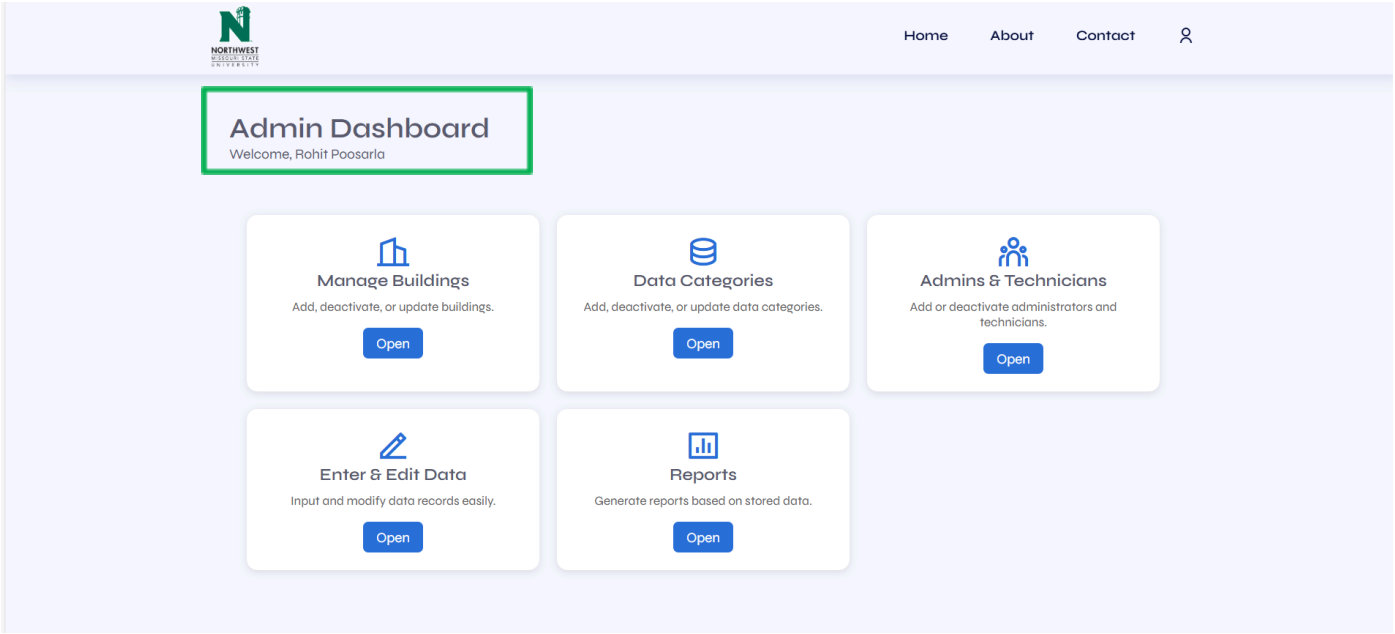




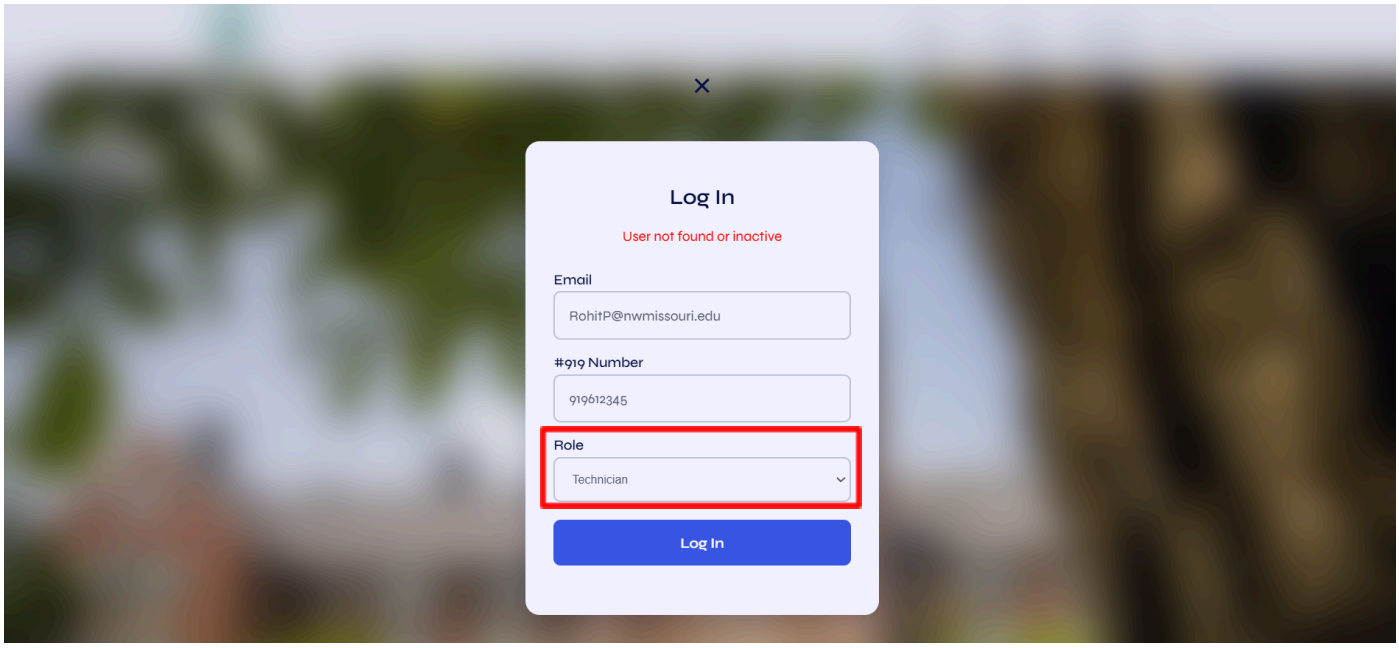
- **Verify login** with valid Bearcat email, #919 number, and for **ADMIN** Role



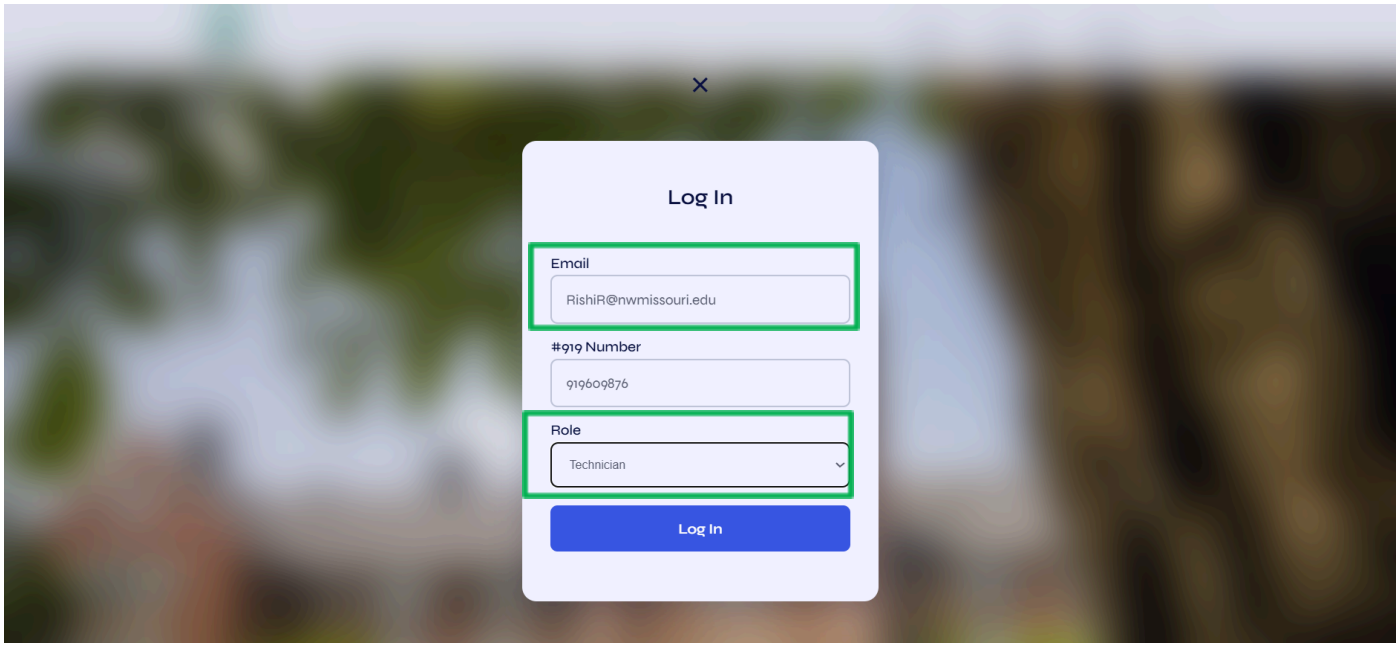
- **ADMIN** Dashboard **View**



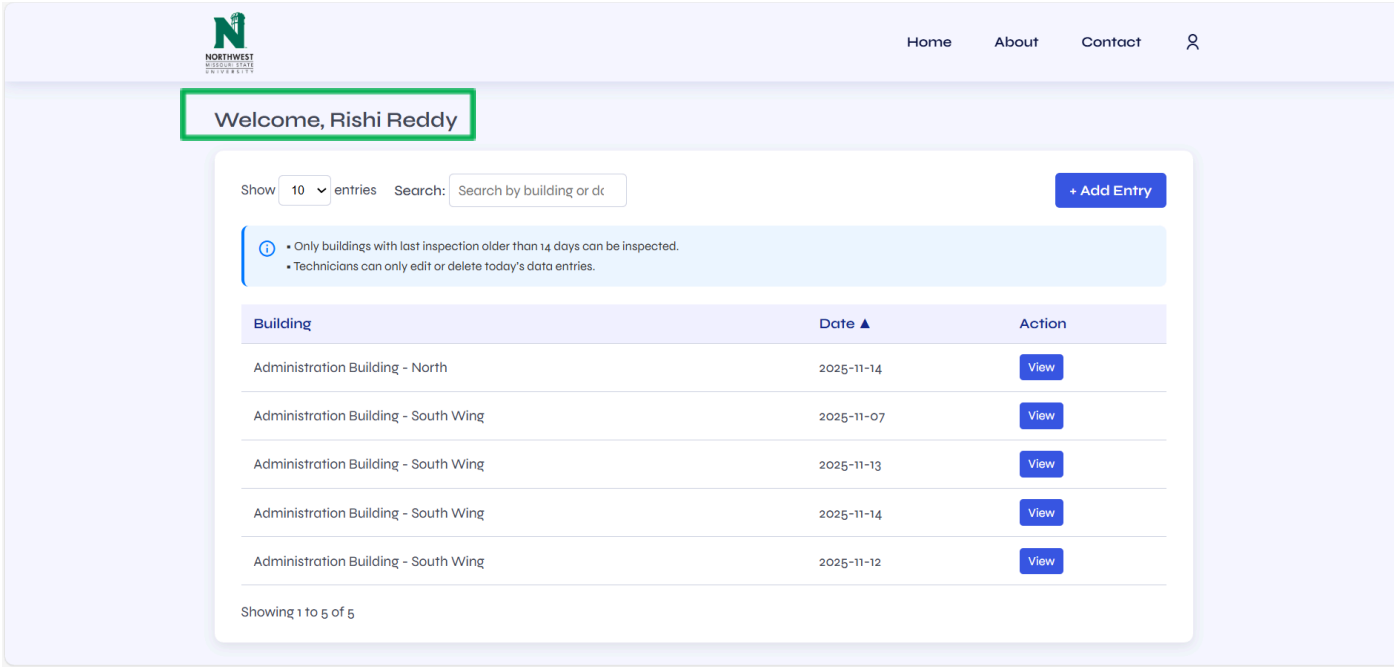
- When a user attempts to log in with a Bearcat email and #919 number but an **incorrect role**, the system displays the error message: '**User not found or inactive'. The same error message appears if the user enters an incorrect email or #919 number.**



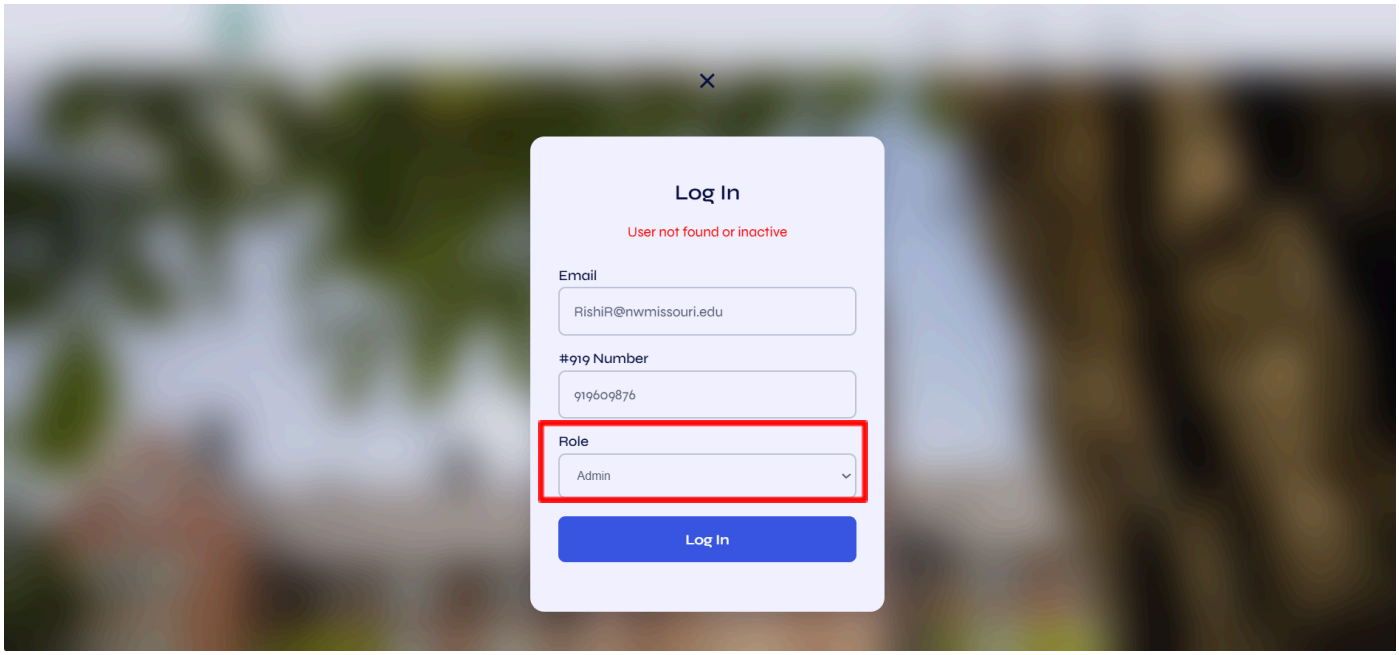
- **Verify login** with **valid** Bearcat email, #919 number, and for **TECHNICIAN** Role



- **TECHNICIAN** Dashboard **View**

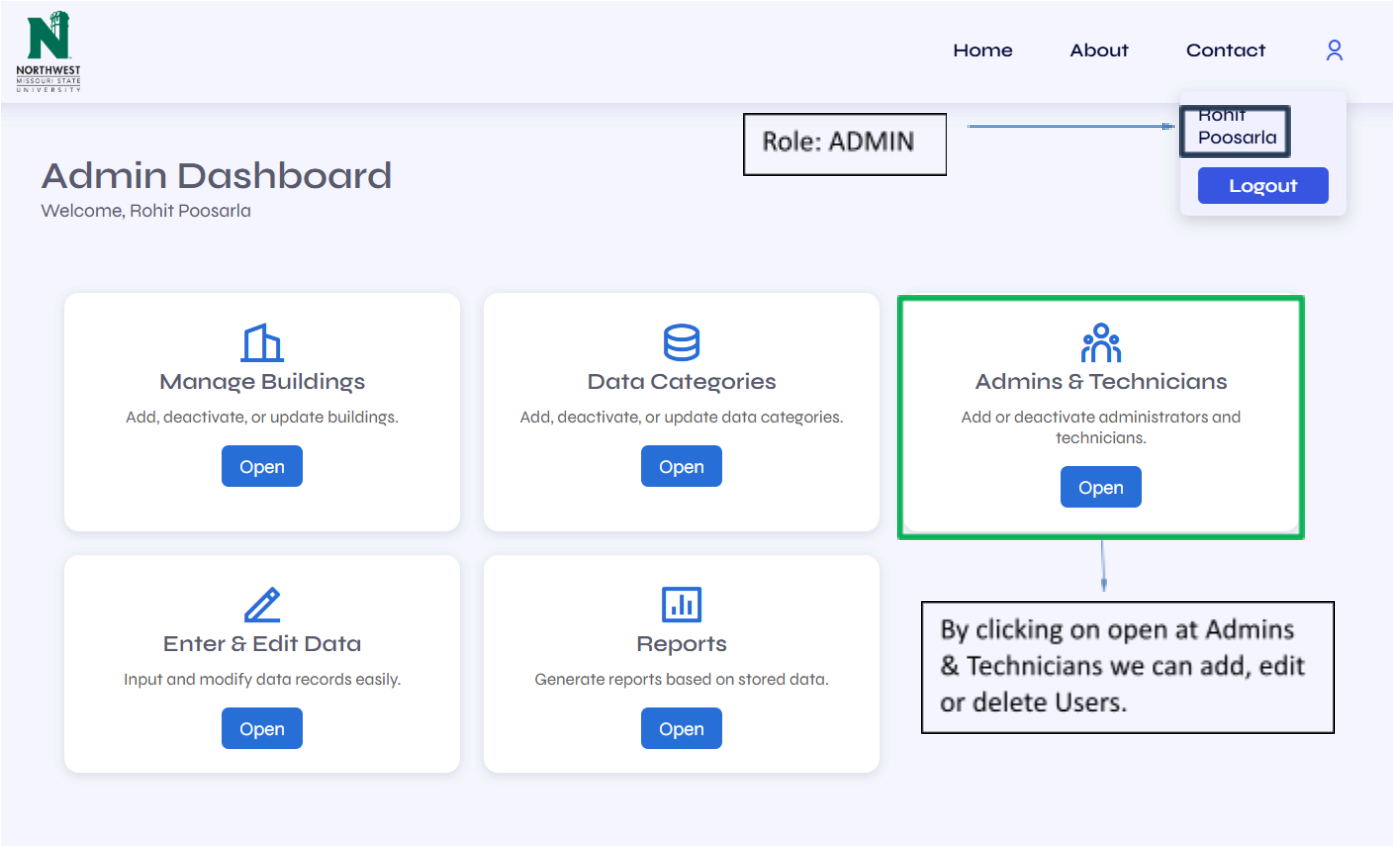


- When a user attempts to log in with a Bearcat email and #919 number but an incorrect role, the system displays the error message: **'User not found or inactive**'. **The same error message appears if the user enters an incorrect email or #919 number.**

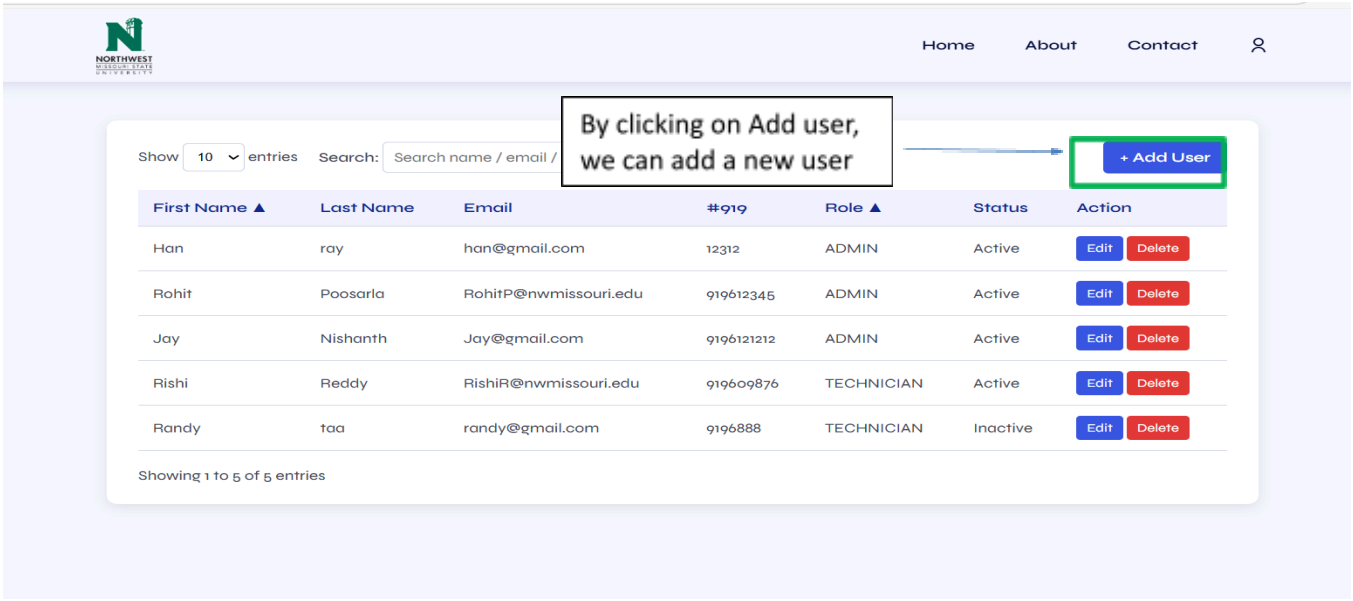


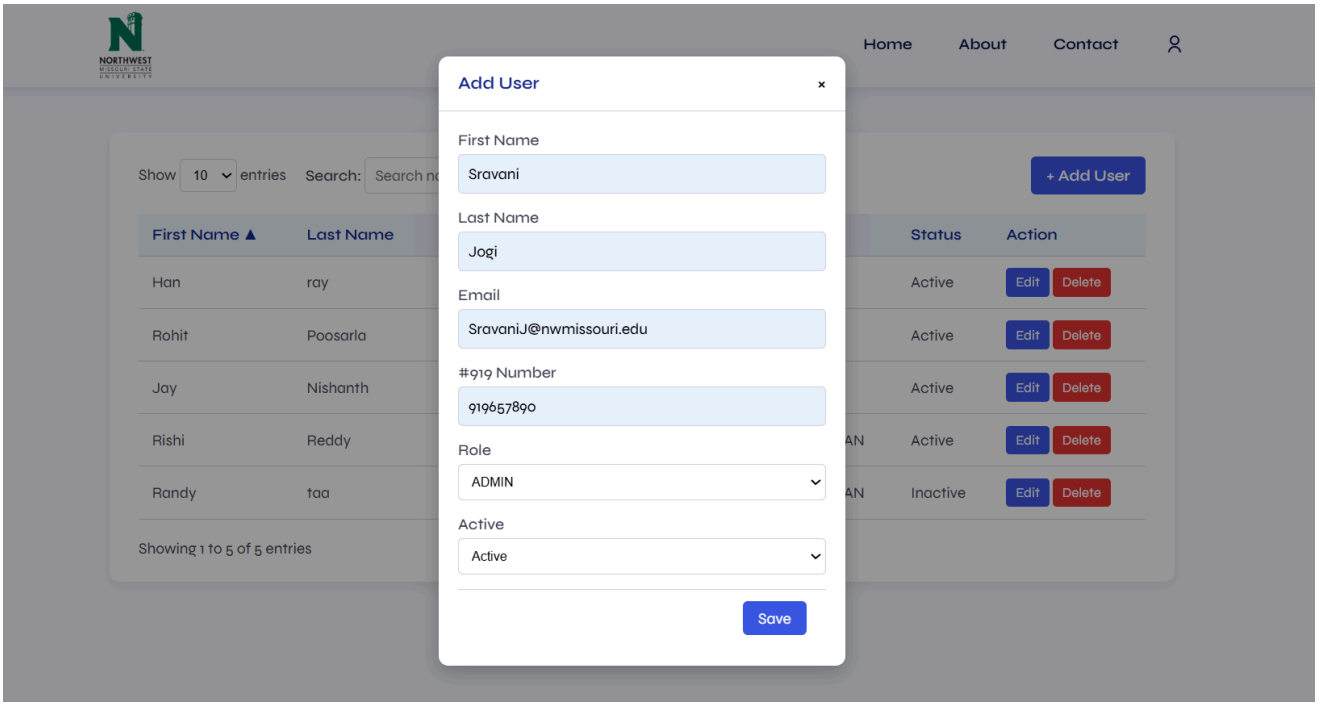
- **ADMIN Functionalities:**
- **User Management**

- Admin can add a New, edit or Delete **Users** in **Admins & Technicians**.

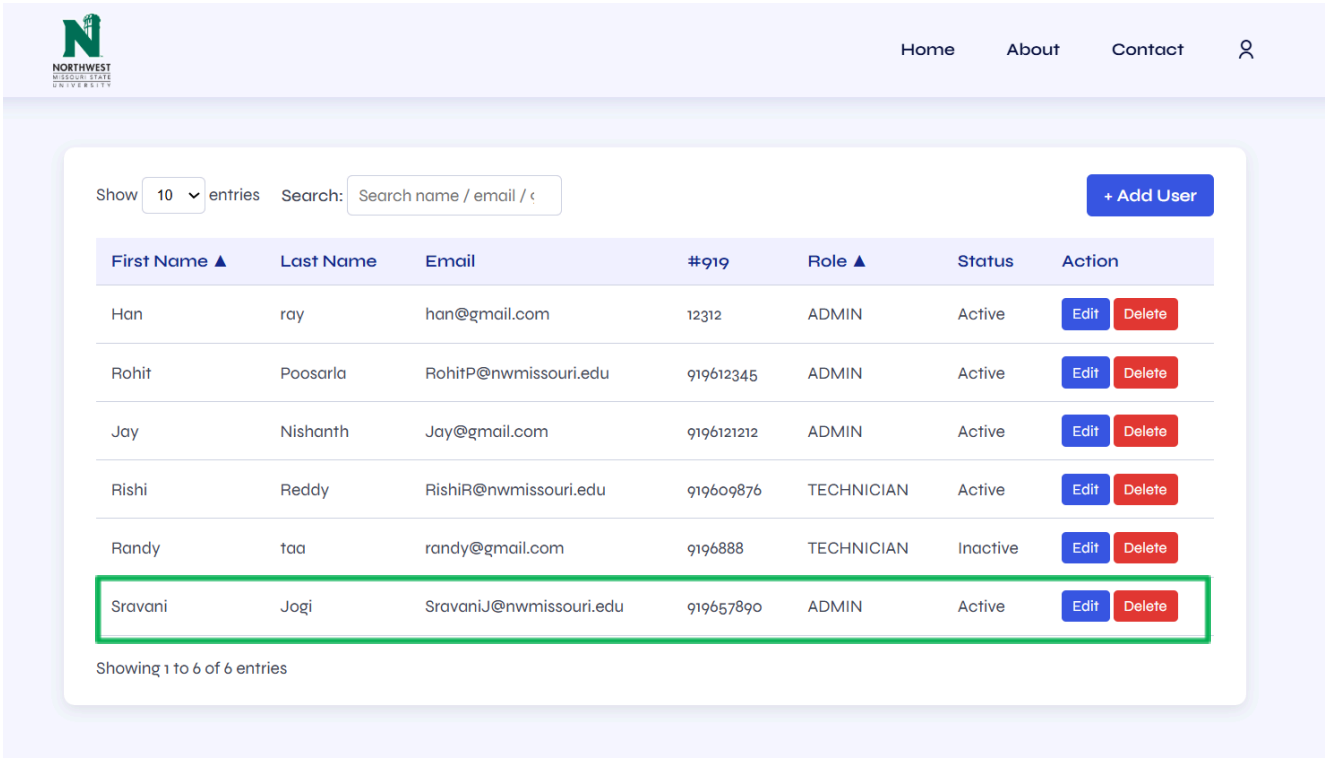


- Adding a **new User**.

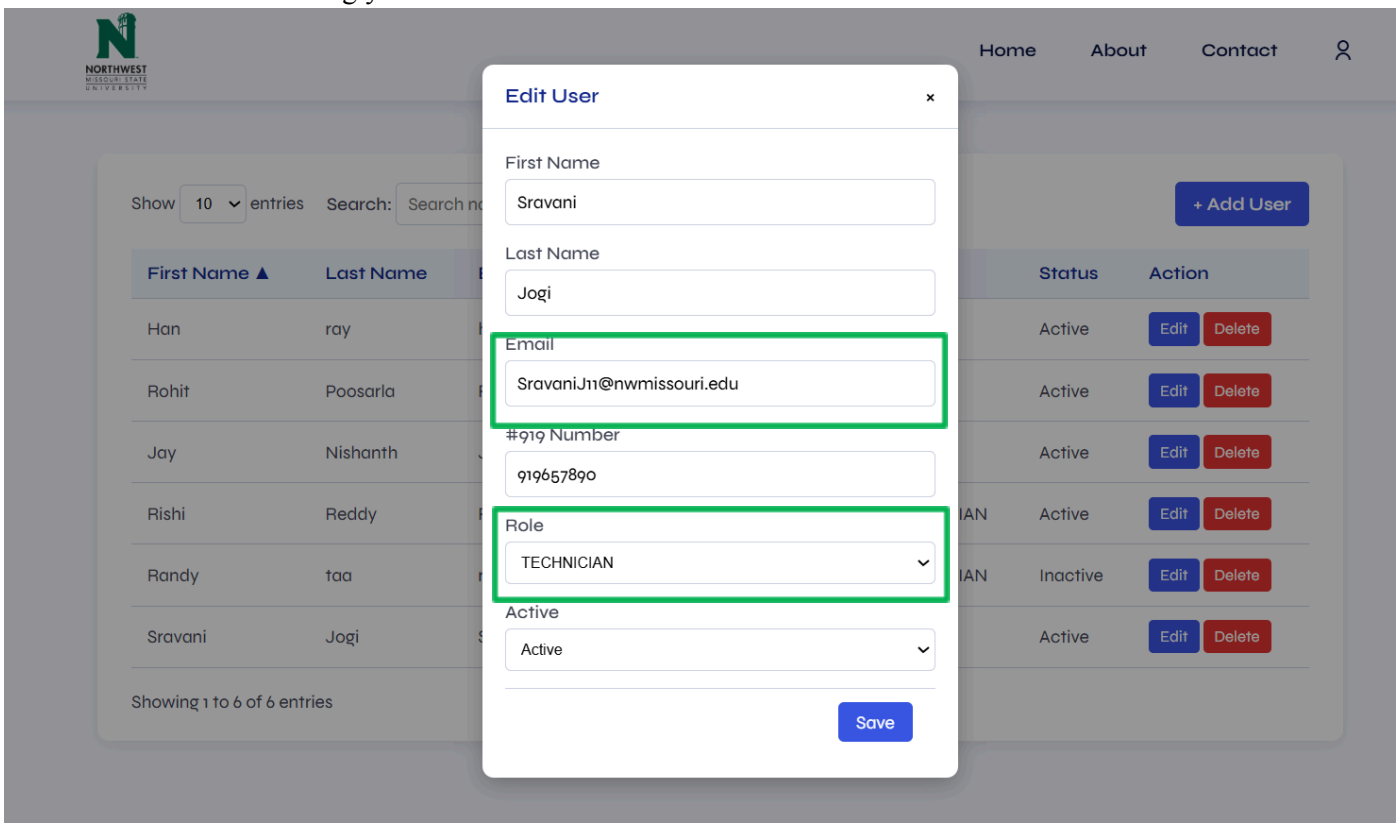


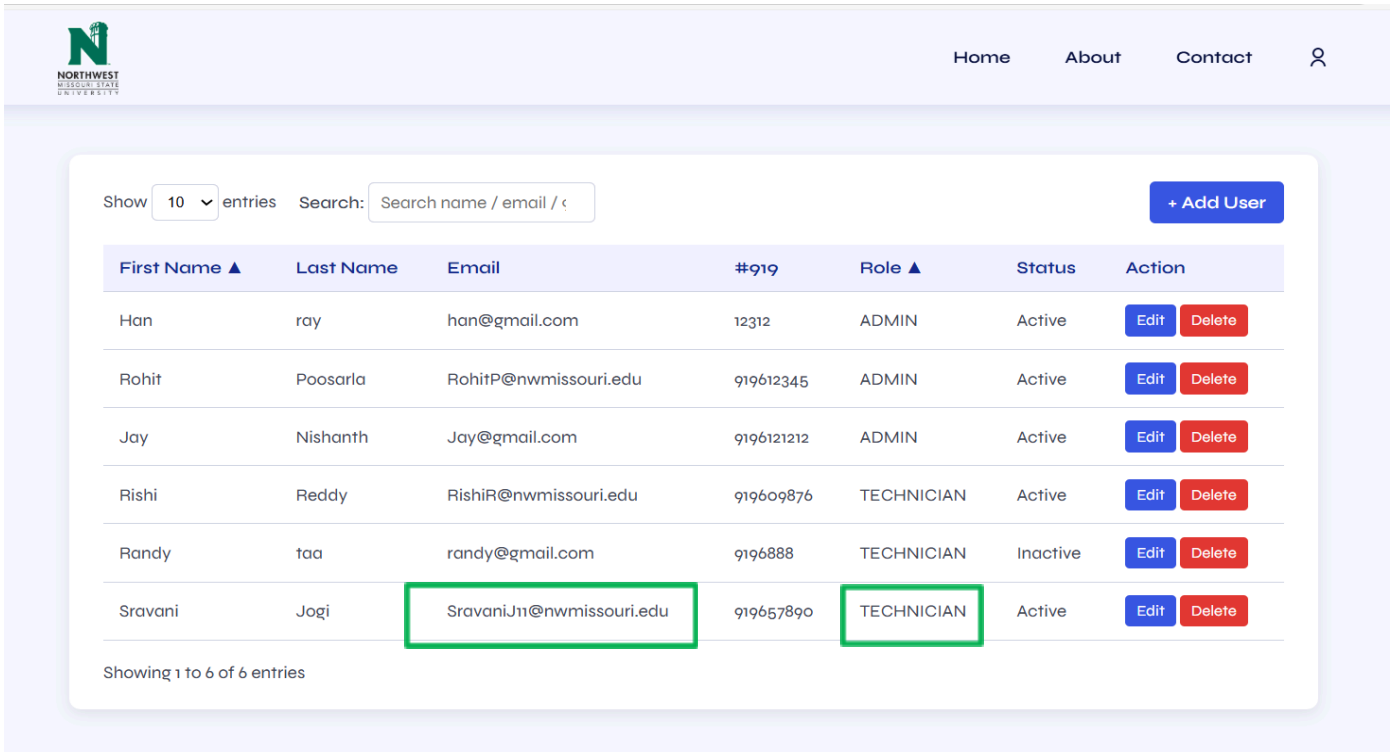


- Here we can see the **new Users** in the **Admins & Technicians**.

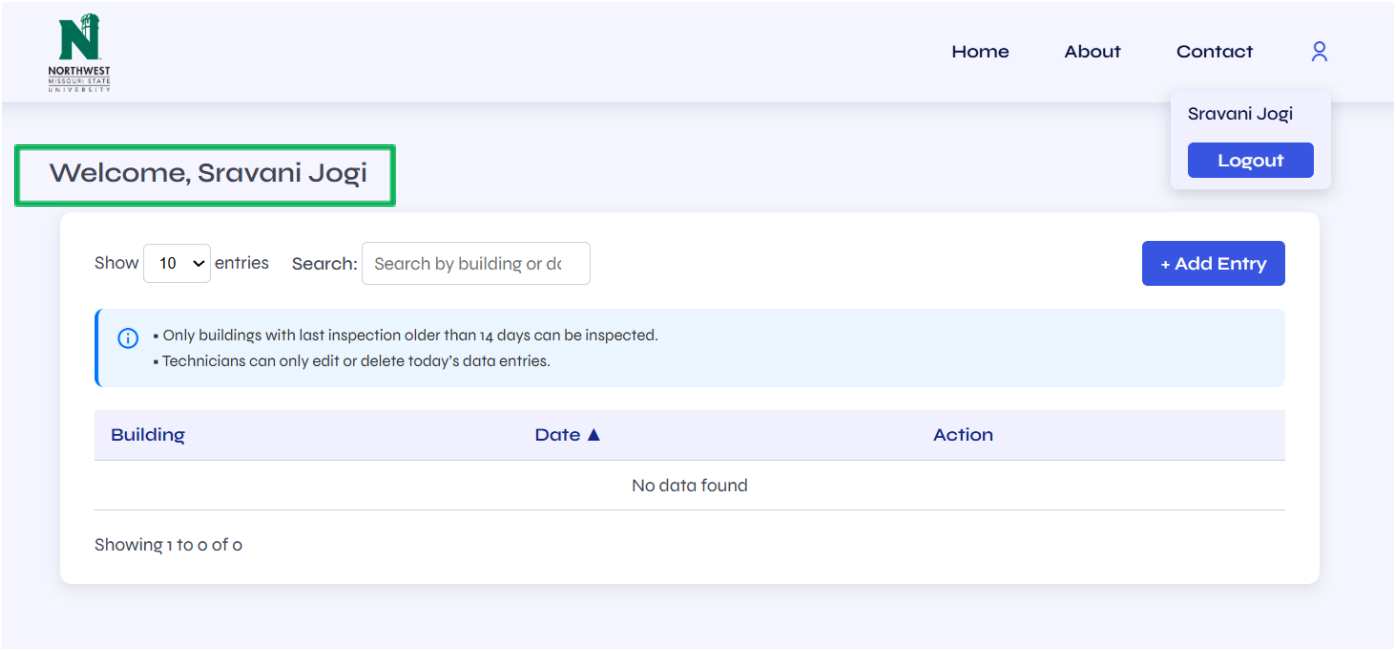


- The admin can **Edit** a user's name, email, or role by clicking the 'Edit' button and updating the user information accordingly.

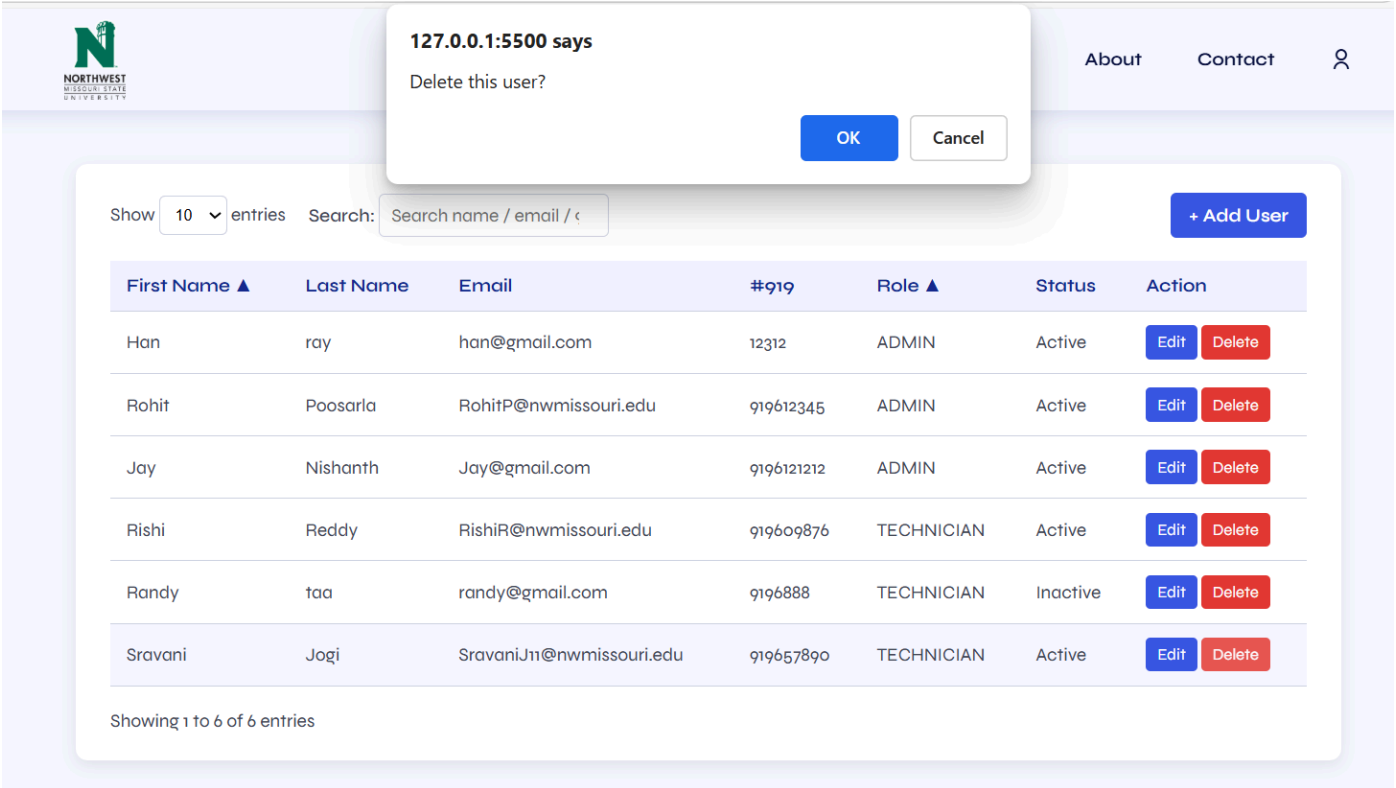




- The newly created user, **Sravani**, can **log in** to the system as **Technician,** so we can see the technician dashboard.

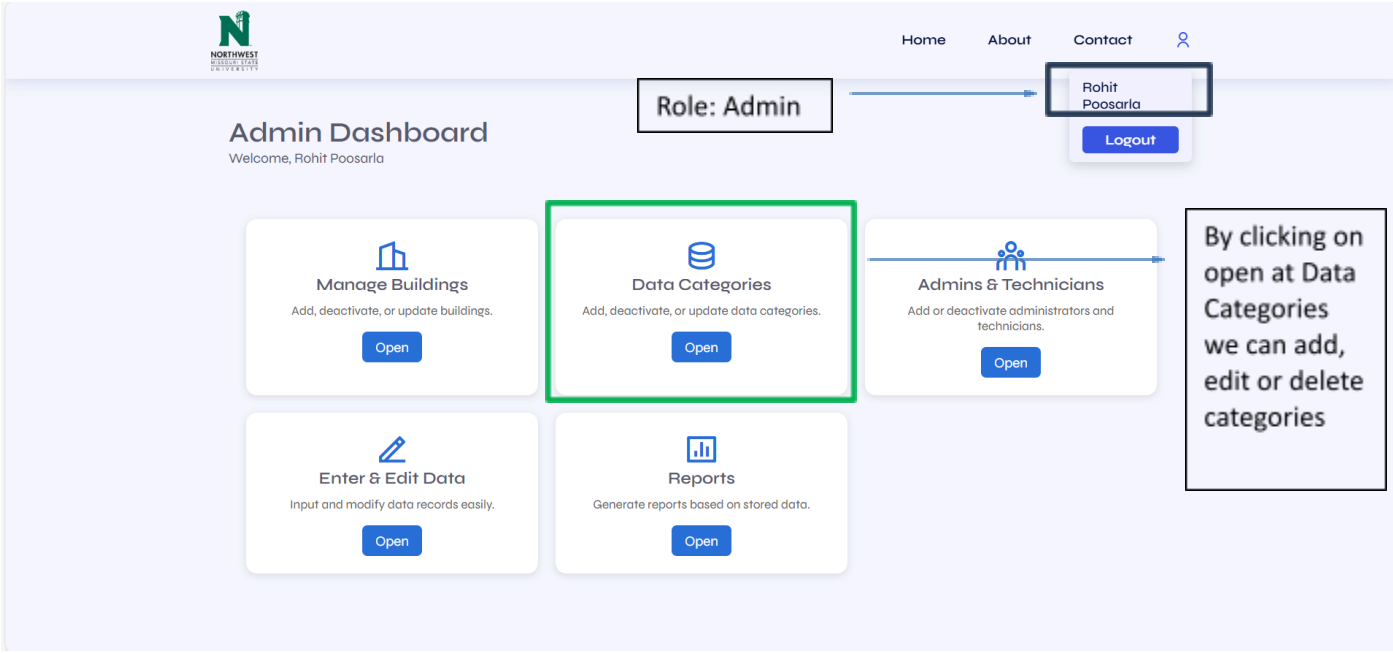


- Admin can also be able to **Delete the Users in Admins & Technicians.**

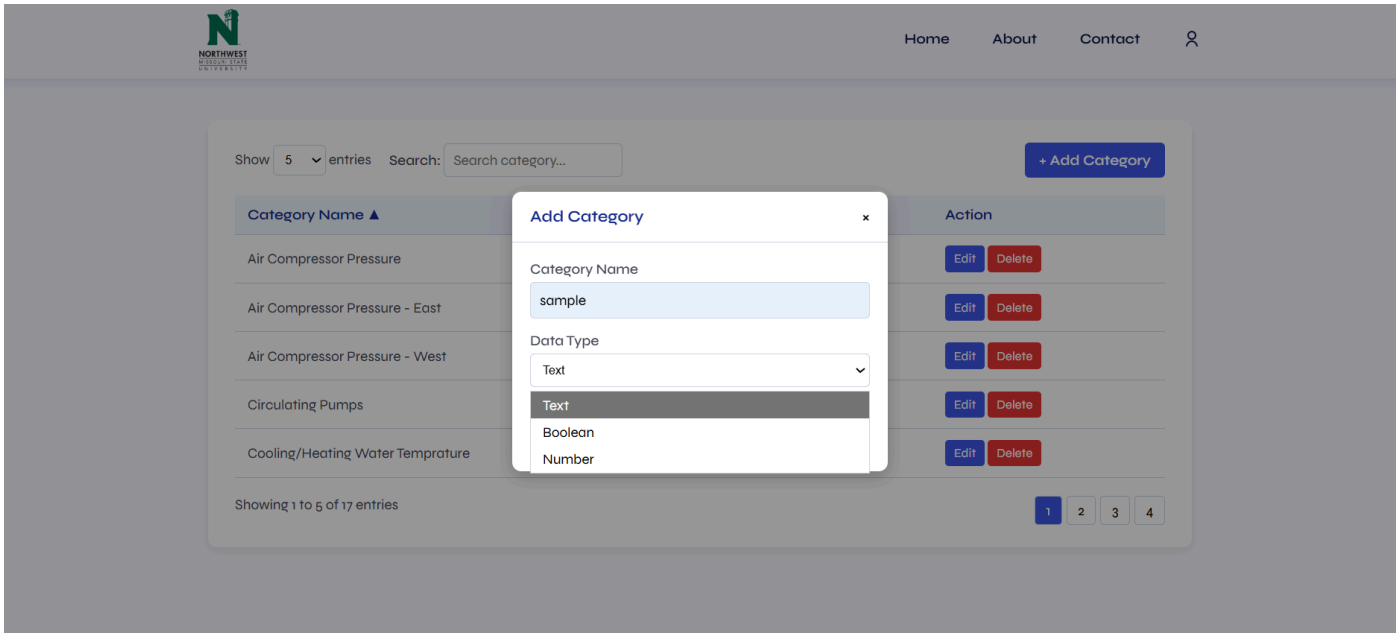


Data Categories

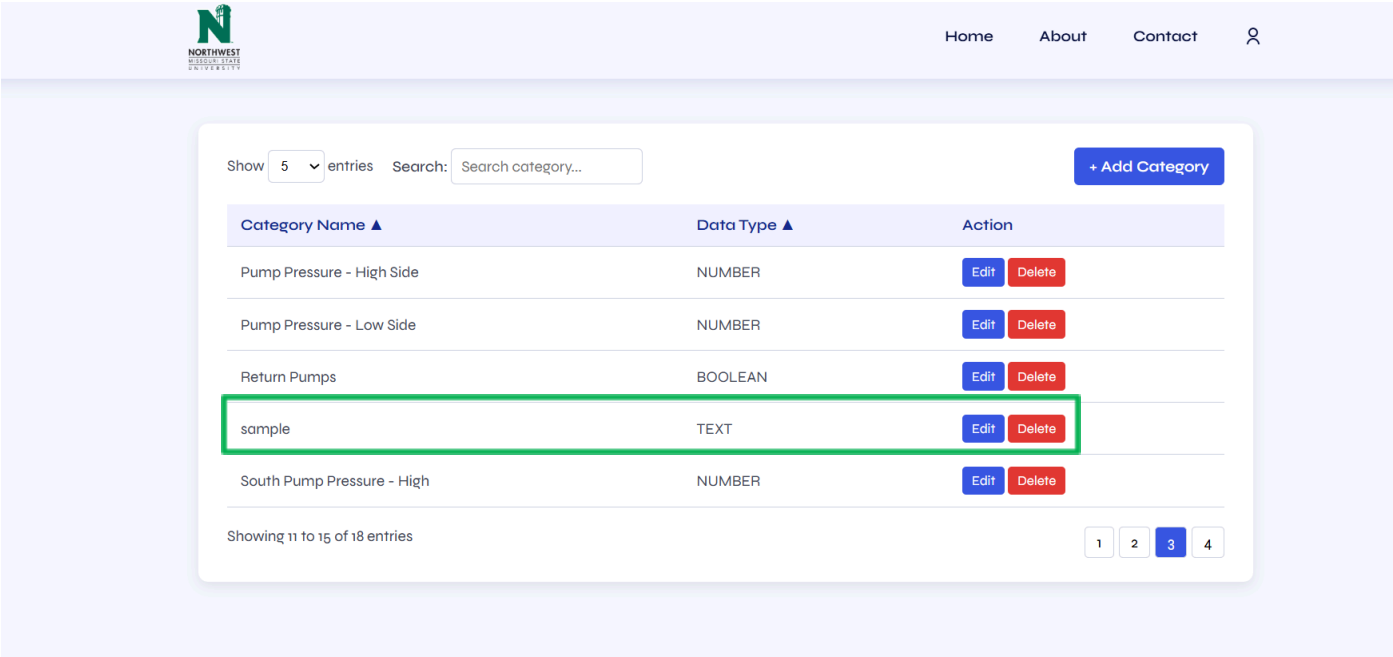
- Admin can Add, Edit or Delete **categories** in the **Data Categories**.



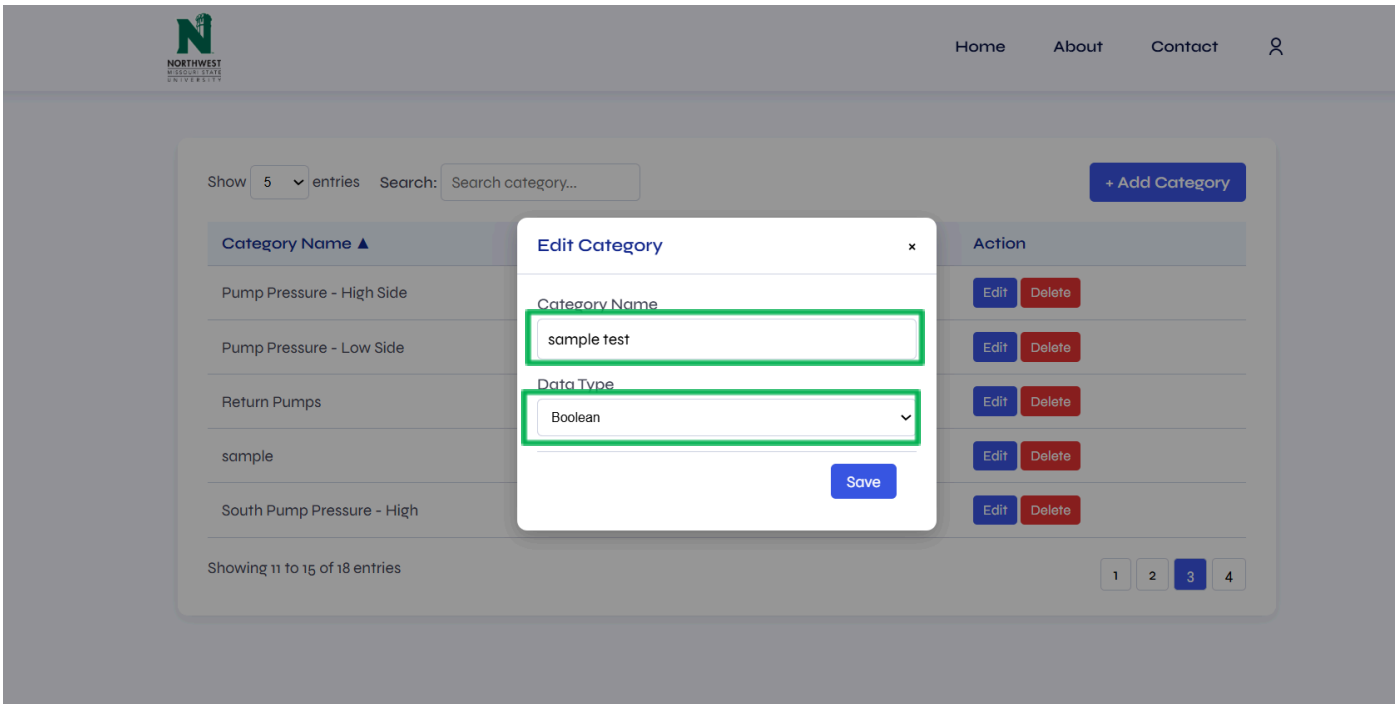
- Adding a **new Categories.**

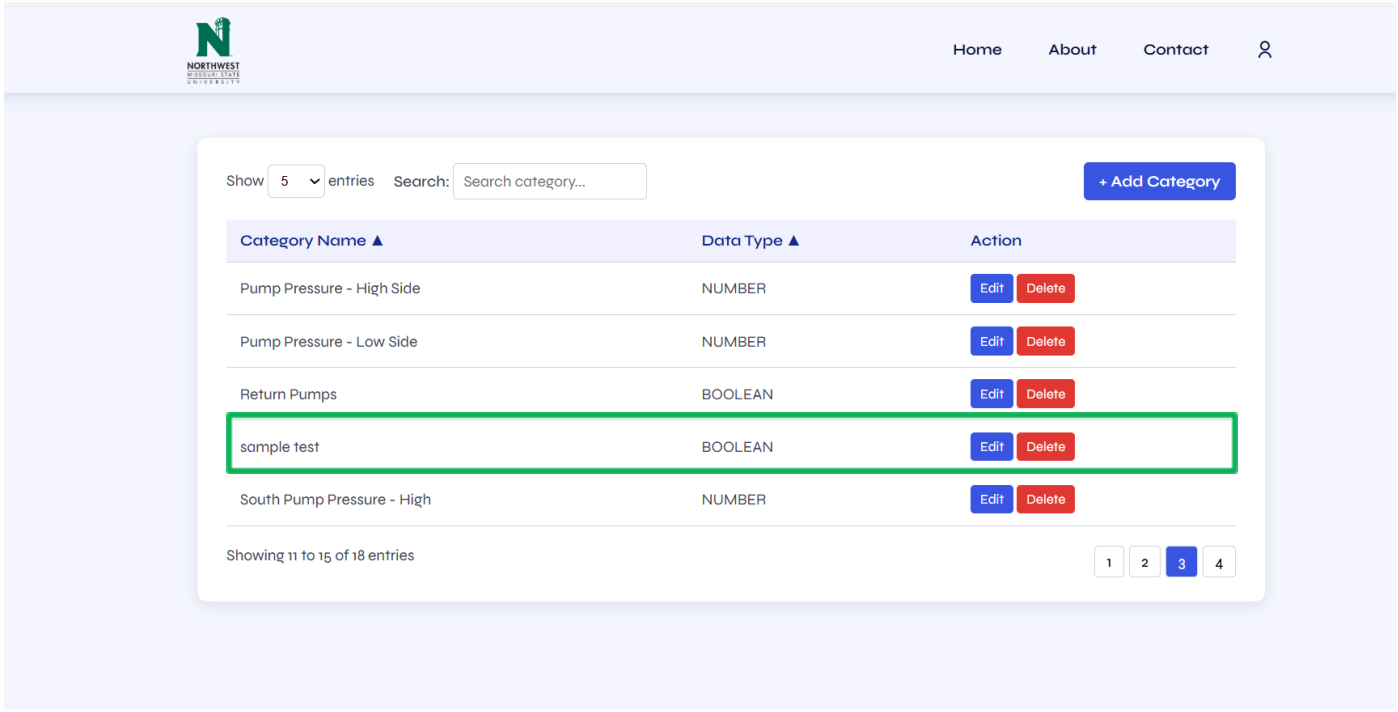


- Here we can see the **new category** in the data categories.

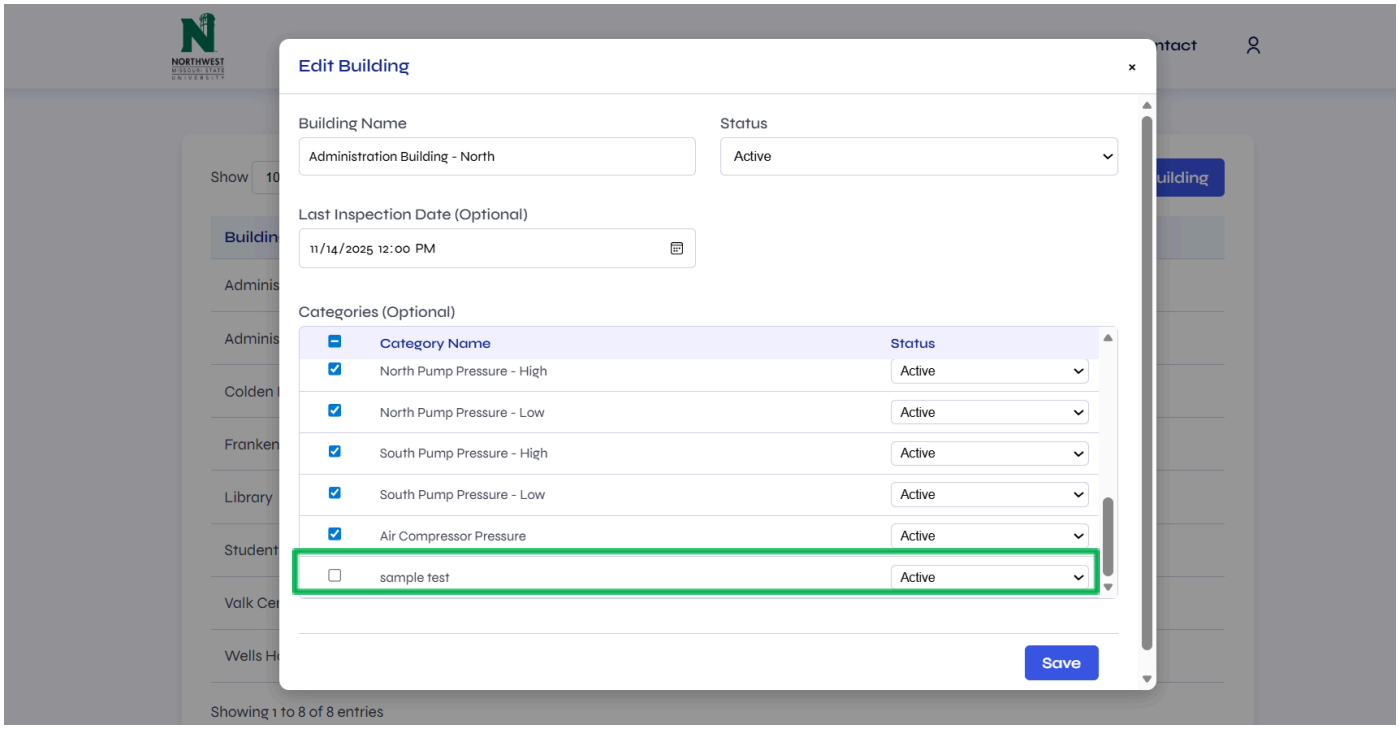


- Editing the category name and data type, here we can see the updated category.

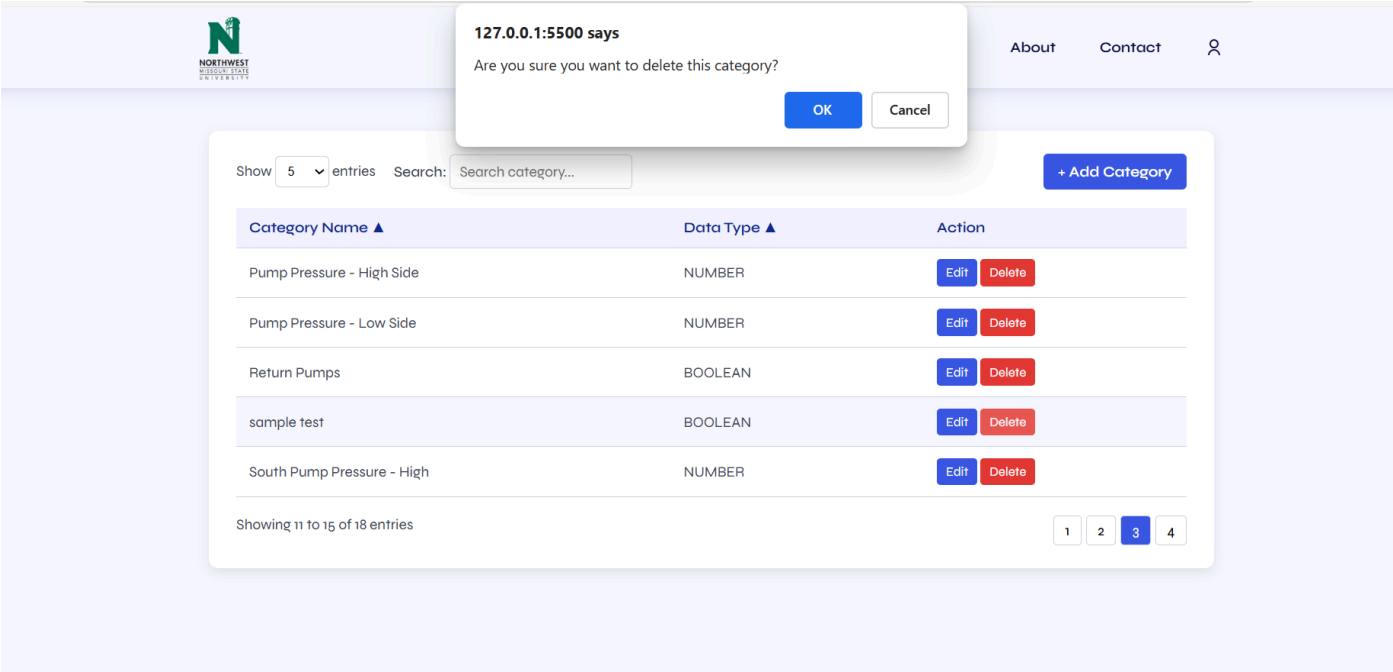




- Verifying that the **new category** should be display in **Building Categories**.

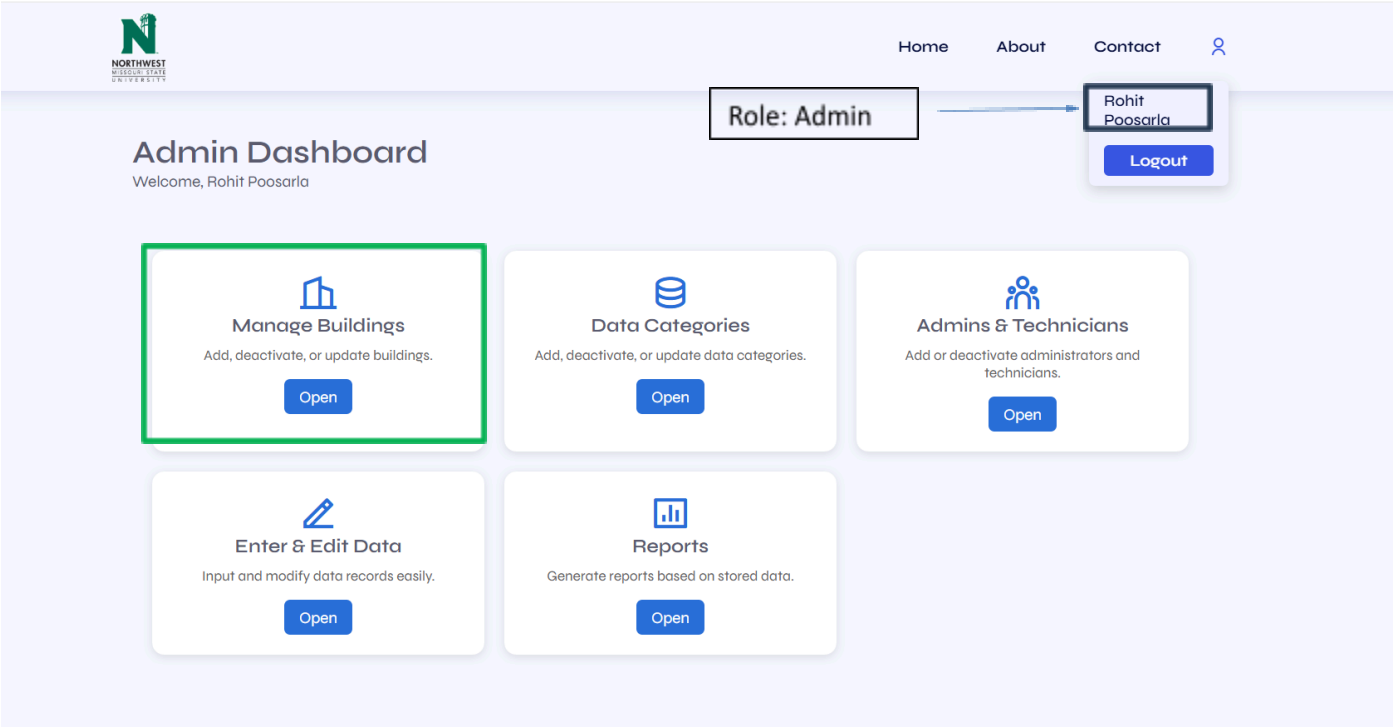


- Deleting a **Category.**

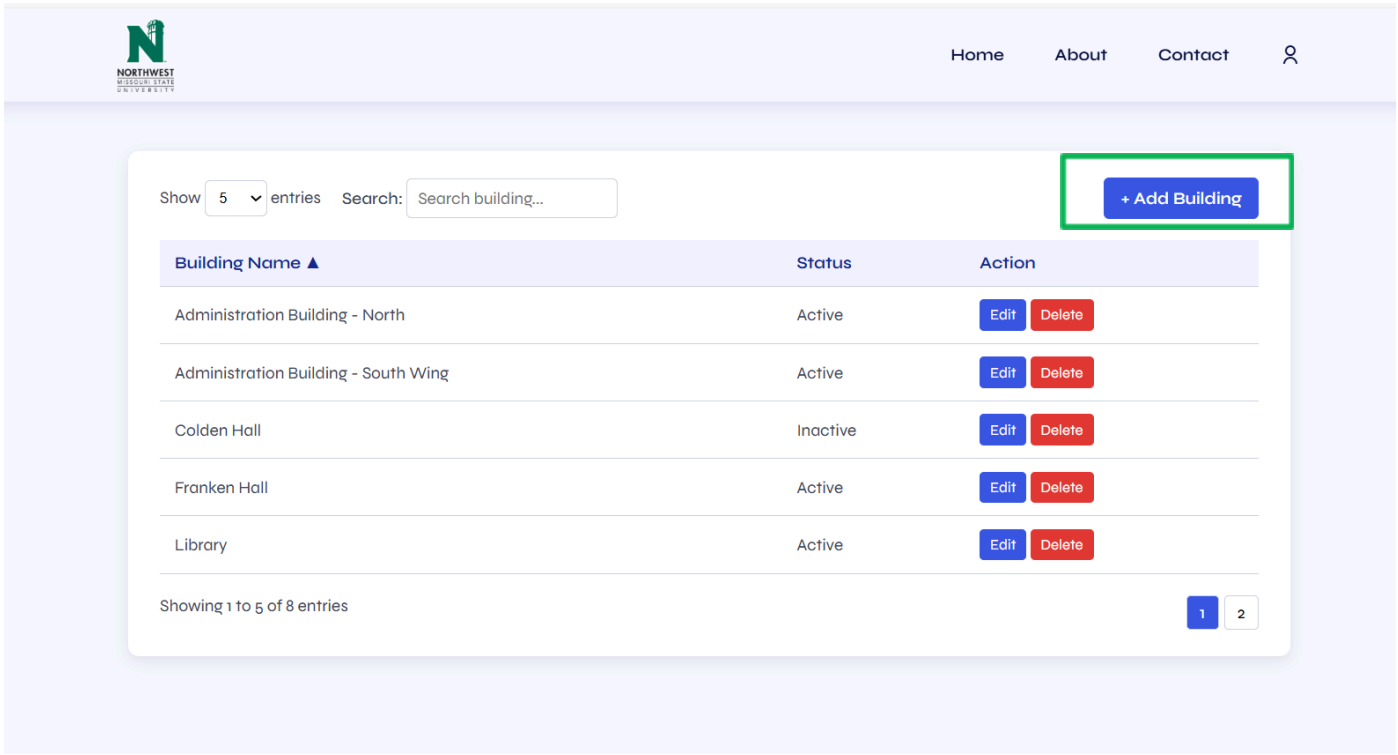


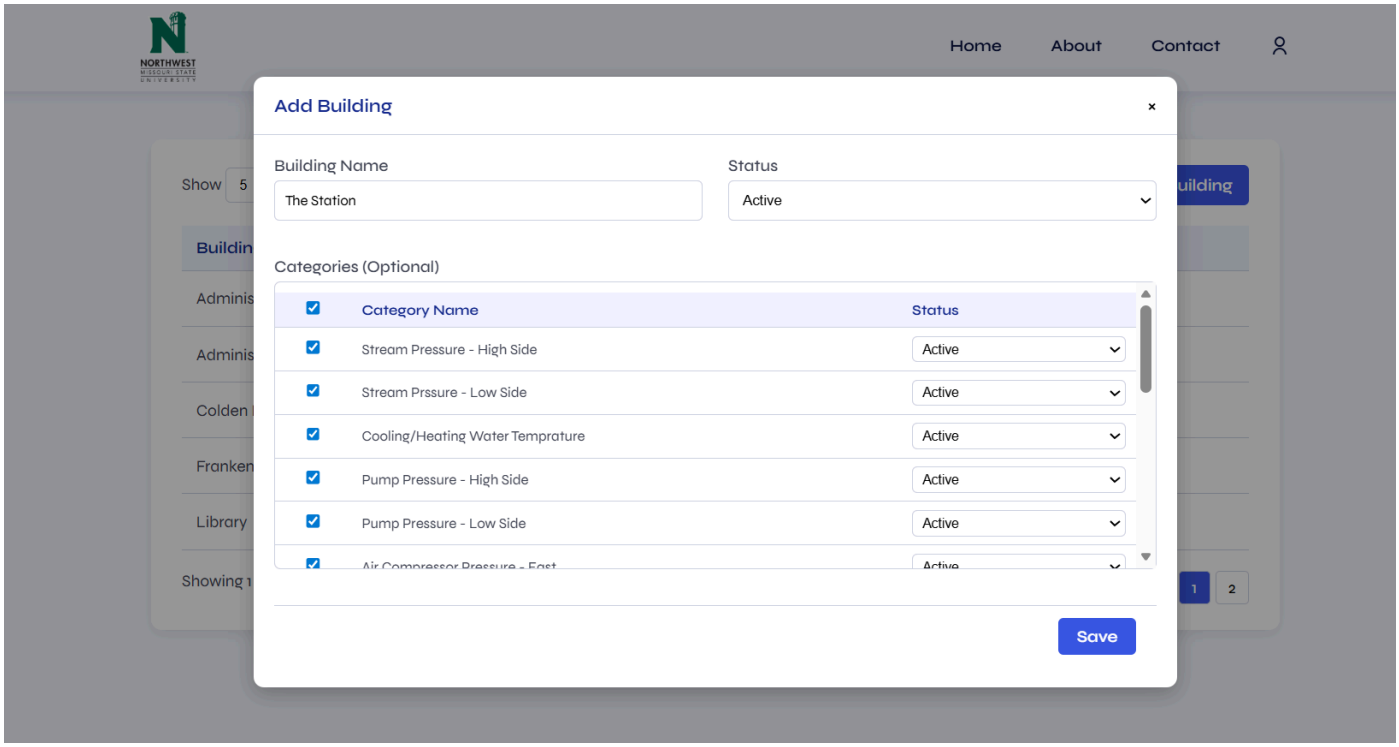
□ Building Management

- Admin can Add, Edit or Building in the Manage Buildings:

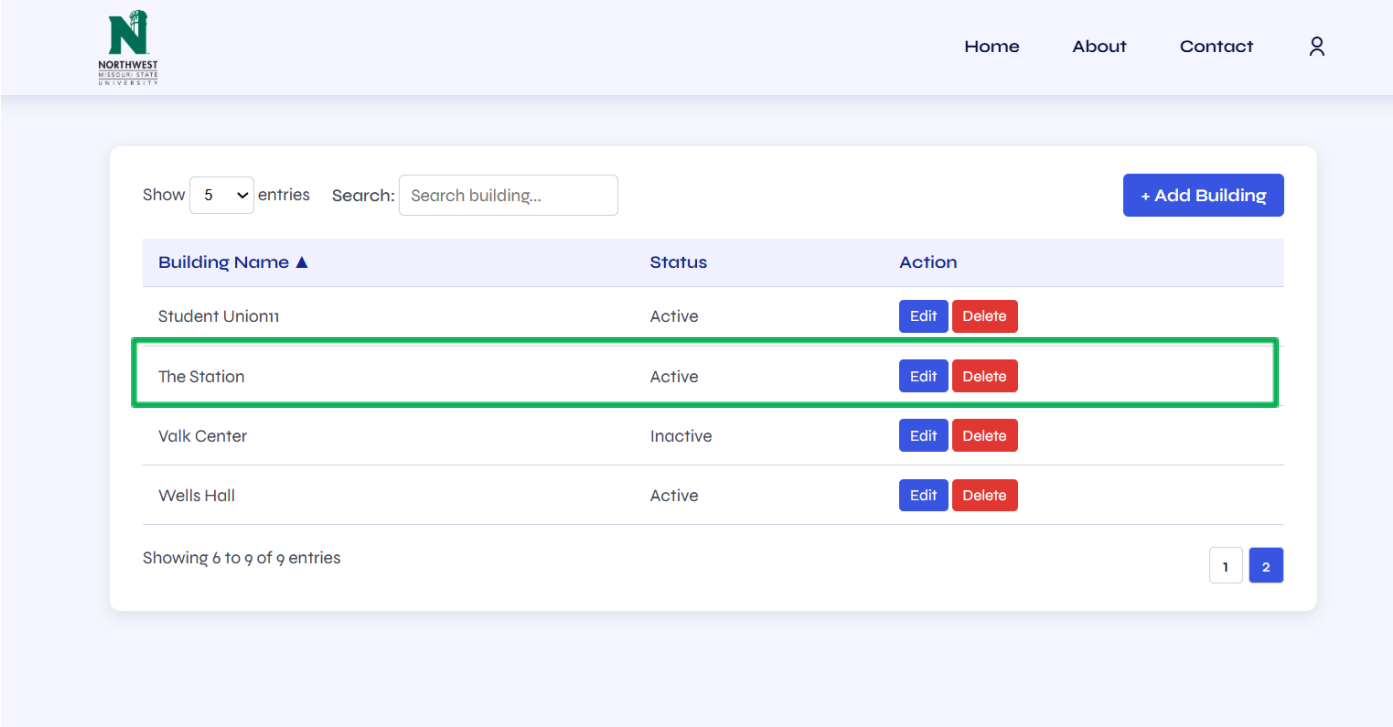


- Adding a **new Building.**
  While adding a new building, users can also select the categories that will be required in the data entry form during inspections. Since this is a new building, the latest inspection data will initially be empty.

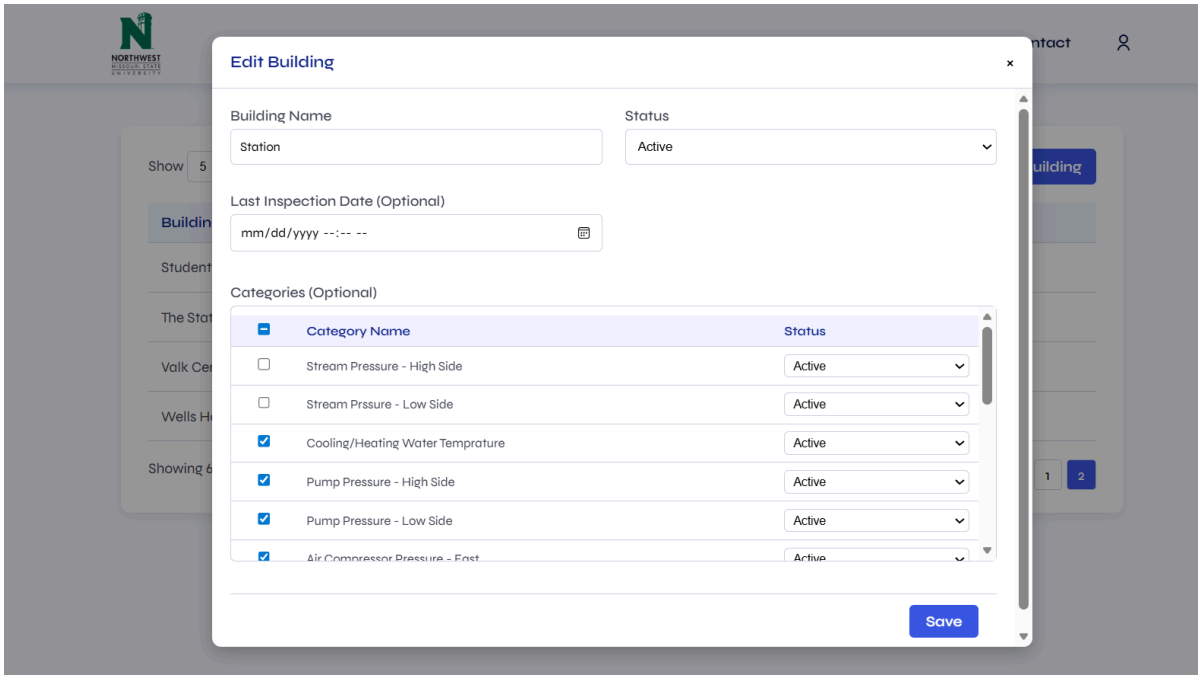


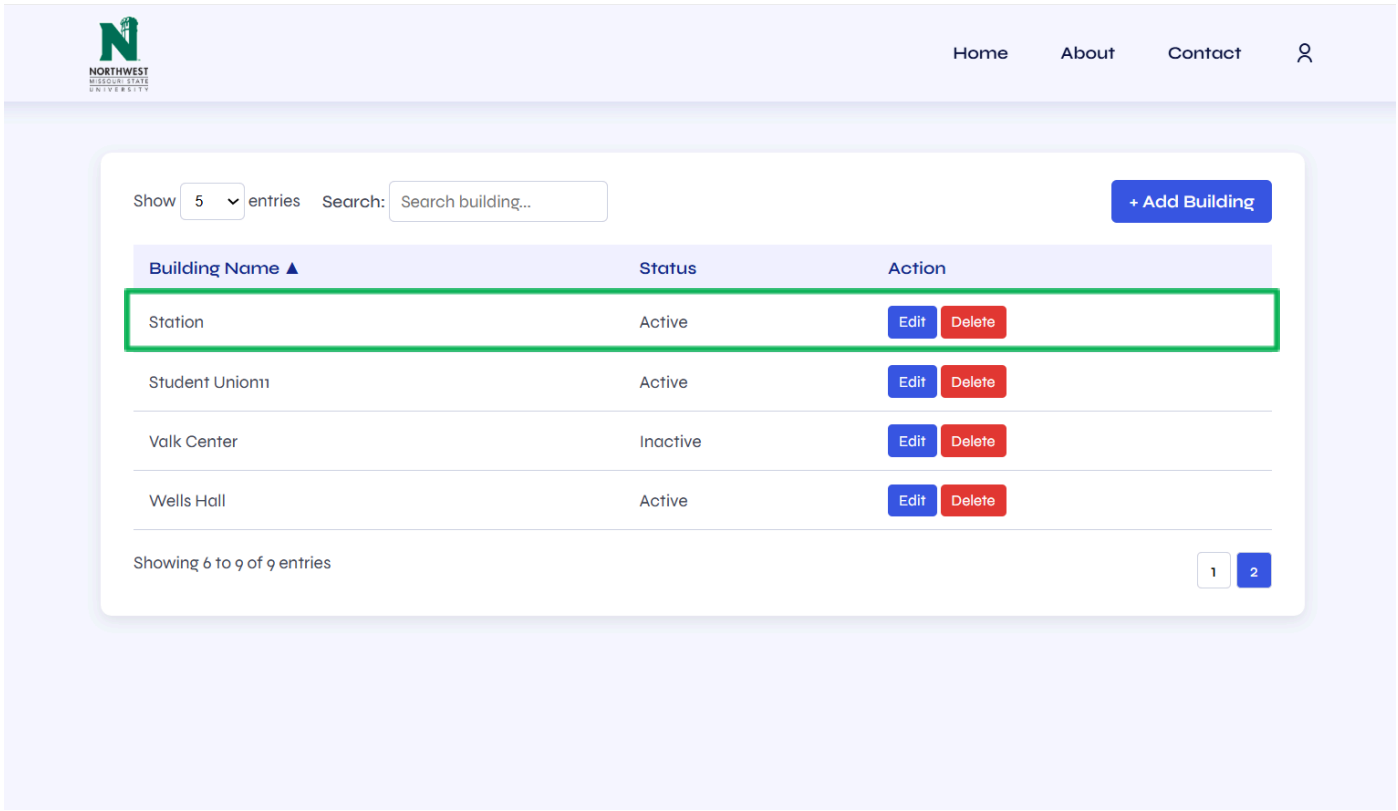


- Here we can see the **new Building** in the Manage Buildings.

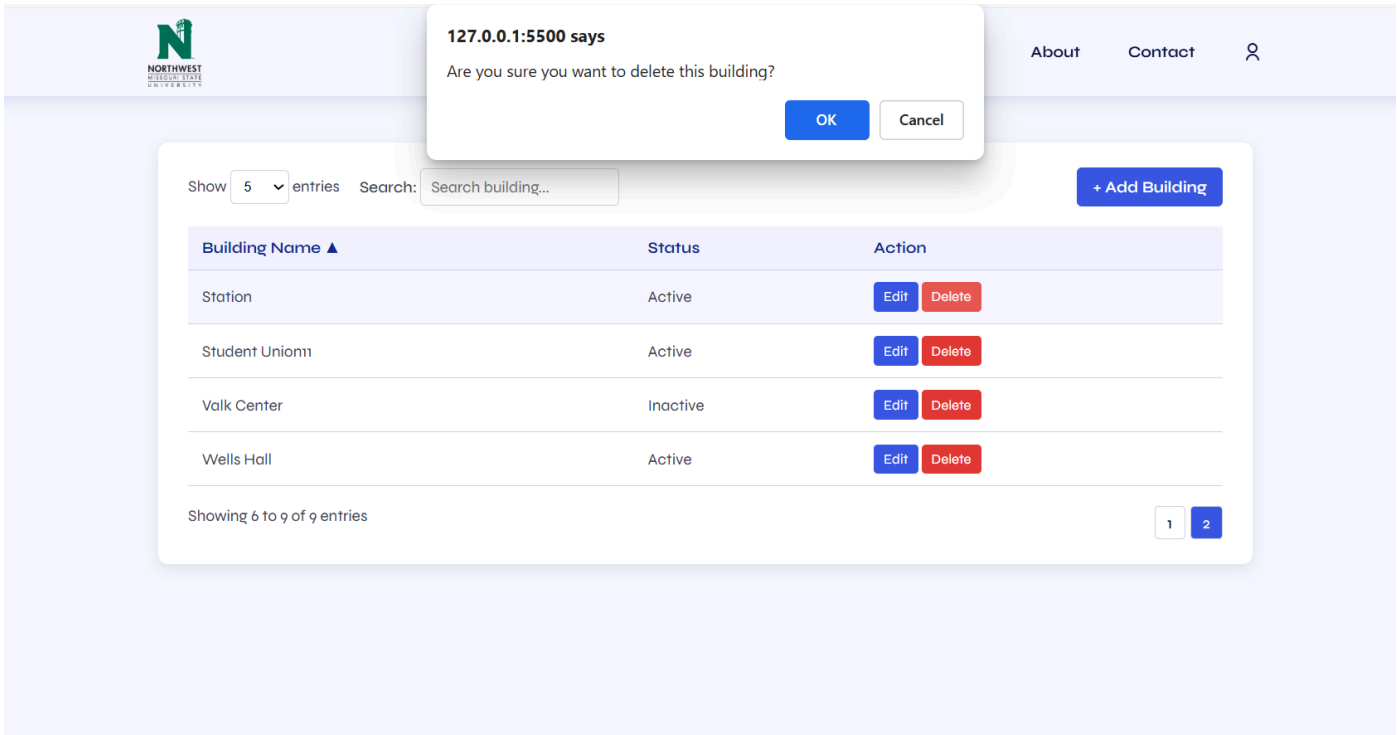


- Users can **edit** the building name and **add or remove categories for that building by selecting the appropriate checkboxes.**



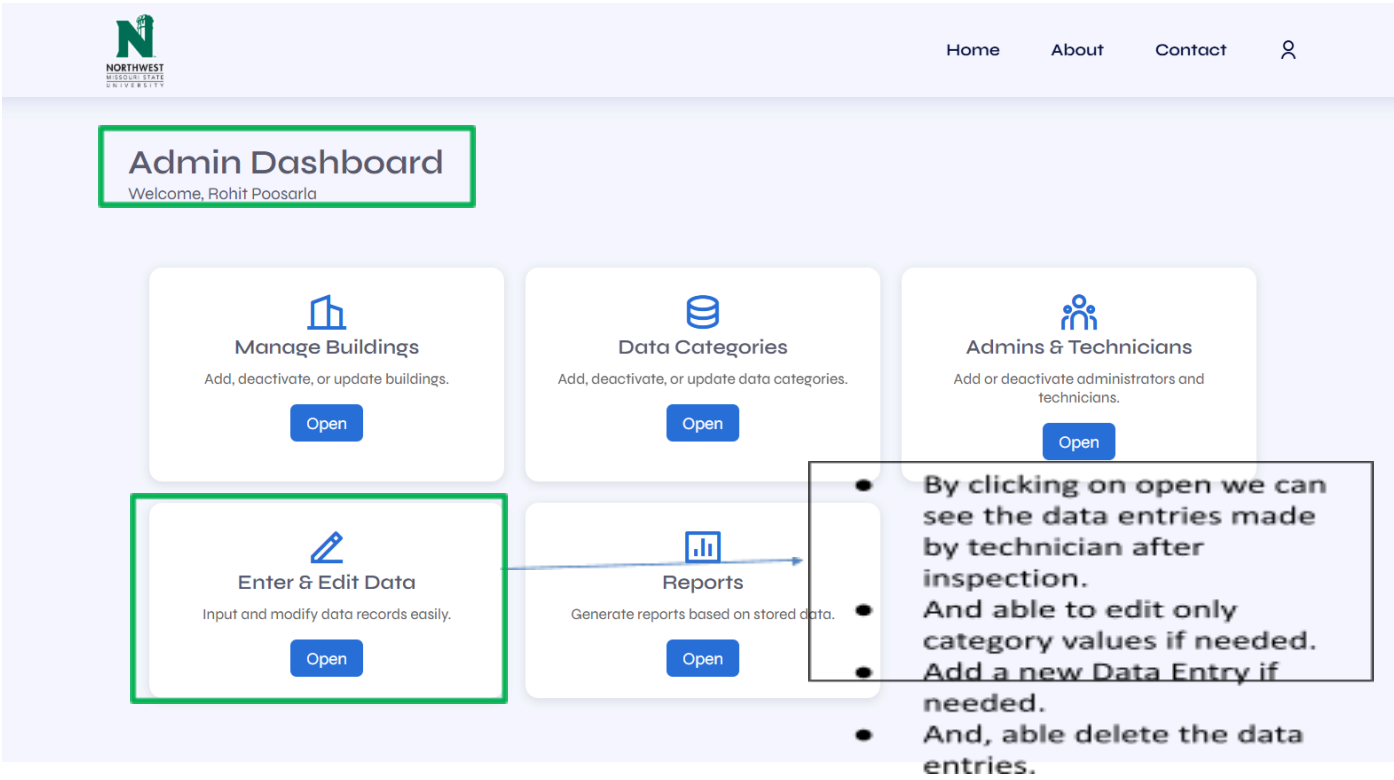


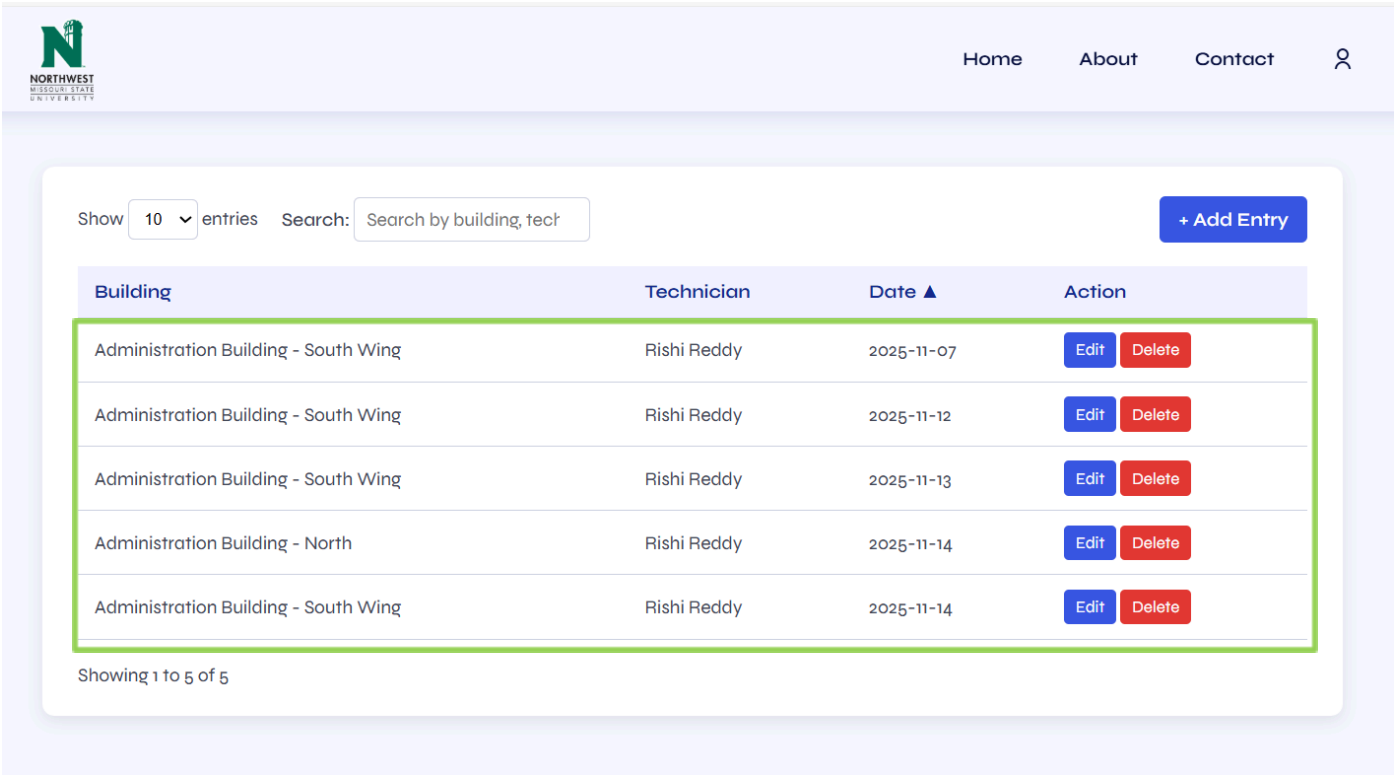
- Deleting a **Building** from Manage Buildings**.**



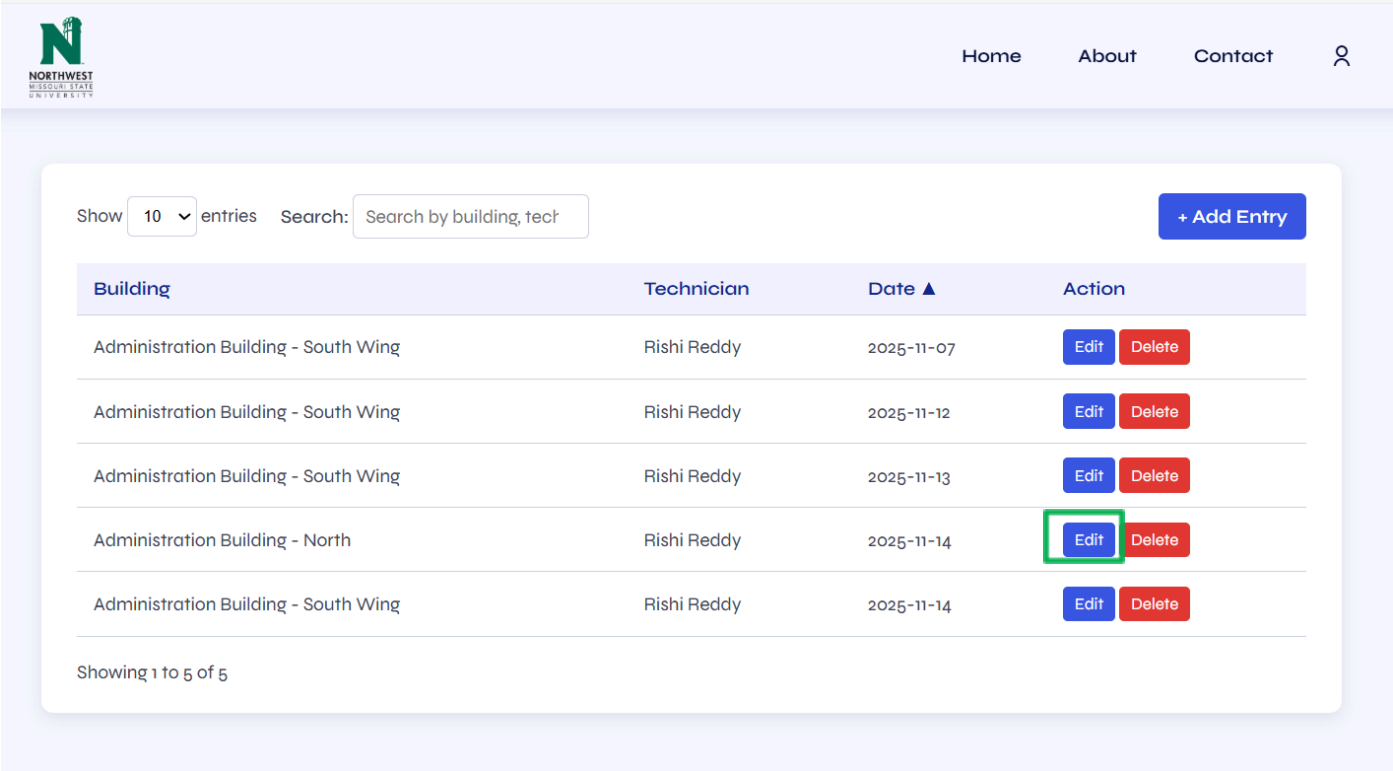
□ Enter and Edit Data

- Admin can **view all data entries** submitted by **Technicians** after **Building Inspections.**

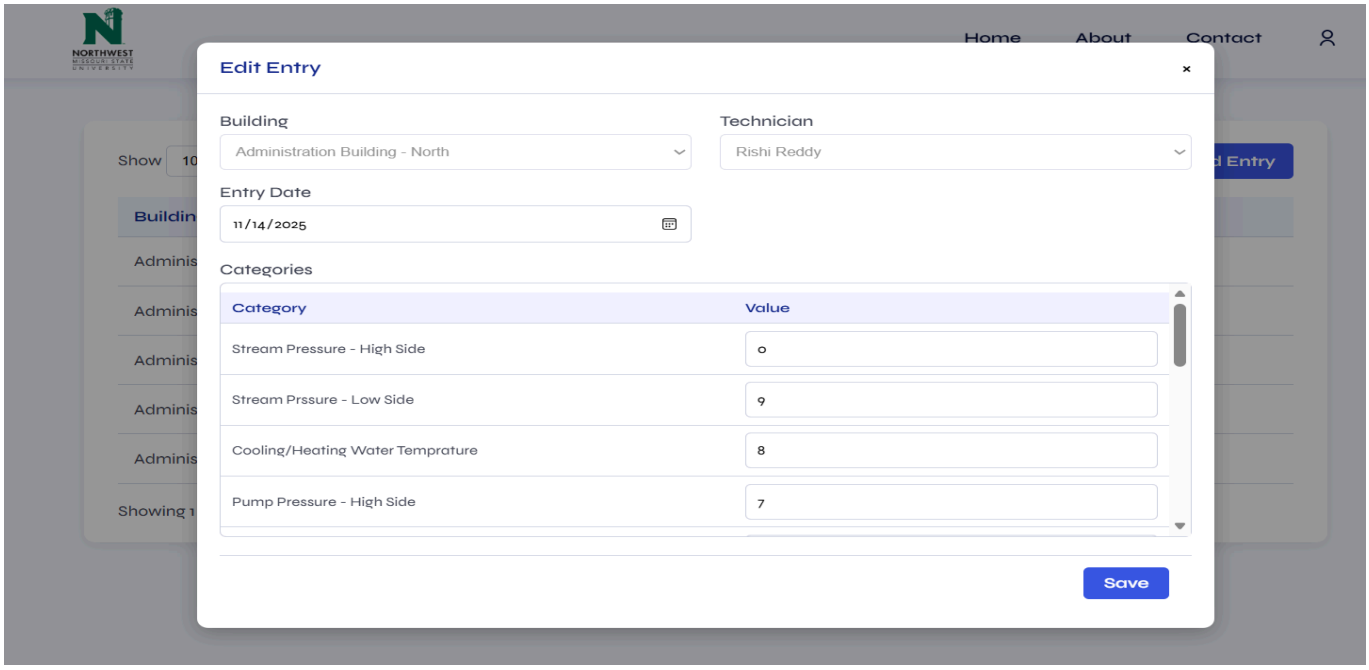




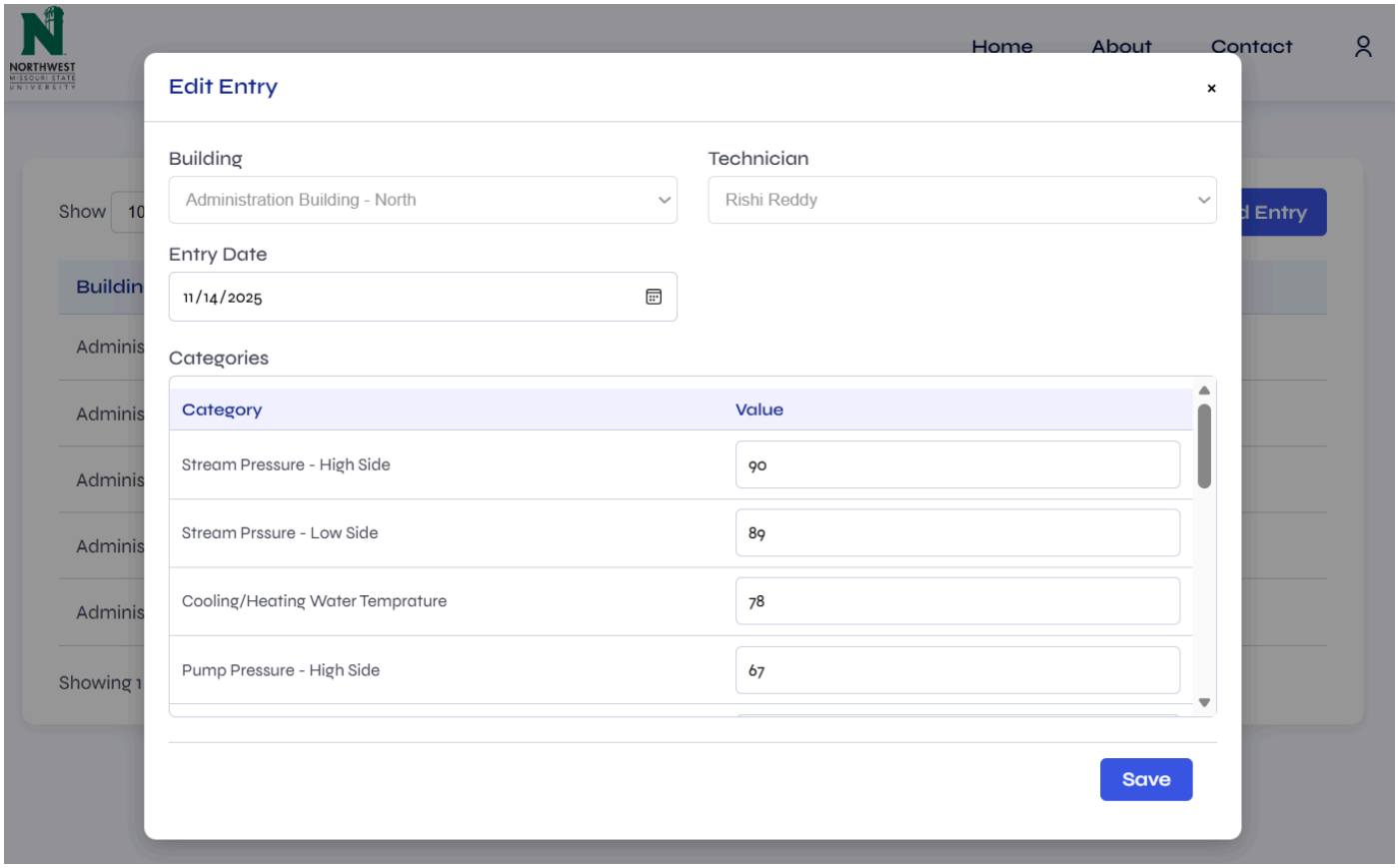
- Admin can **Edit** the values of the **category fields** only and **cannot** change the Technician's name, the building name are read only and if needed they can change the date of entry.



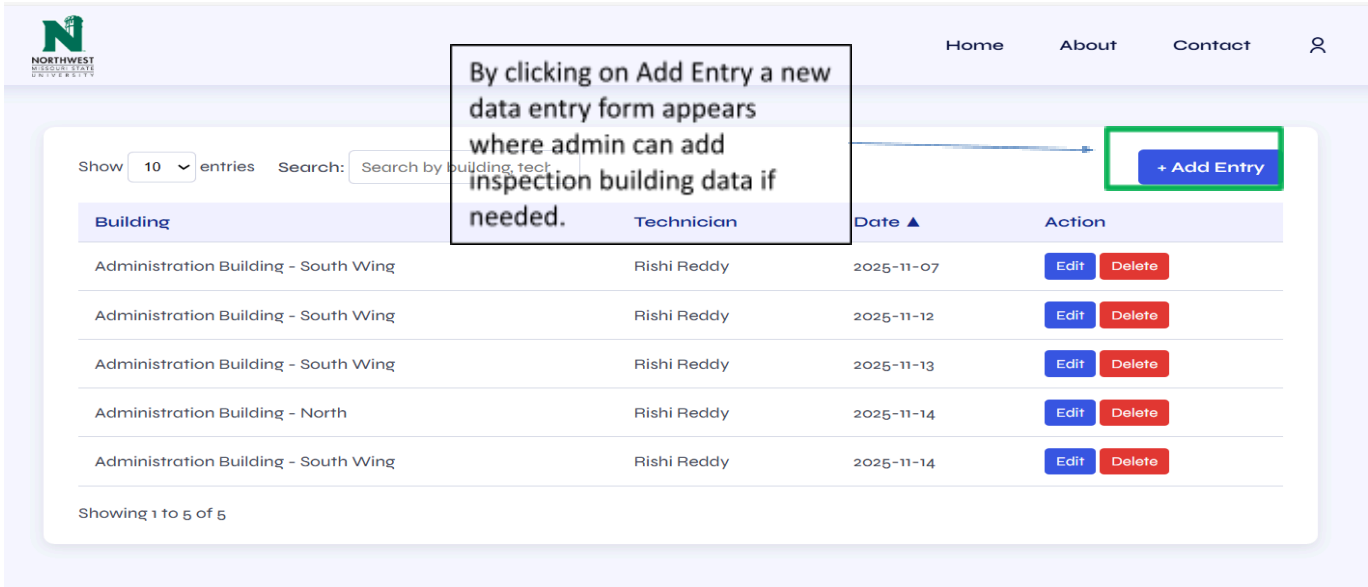
- Before edit the values are

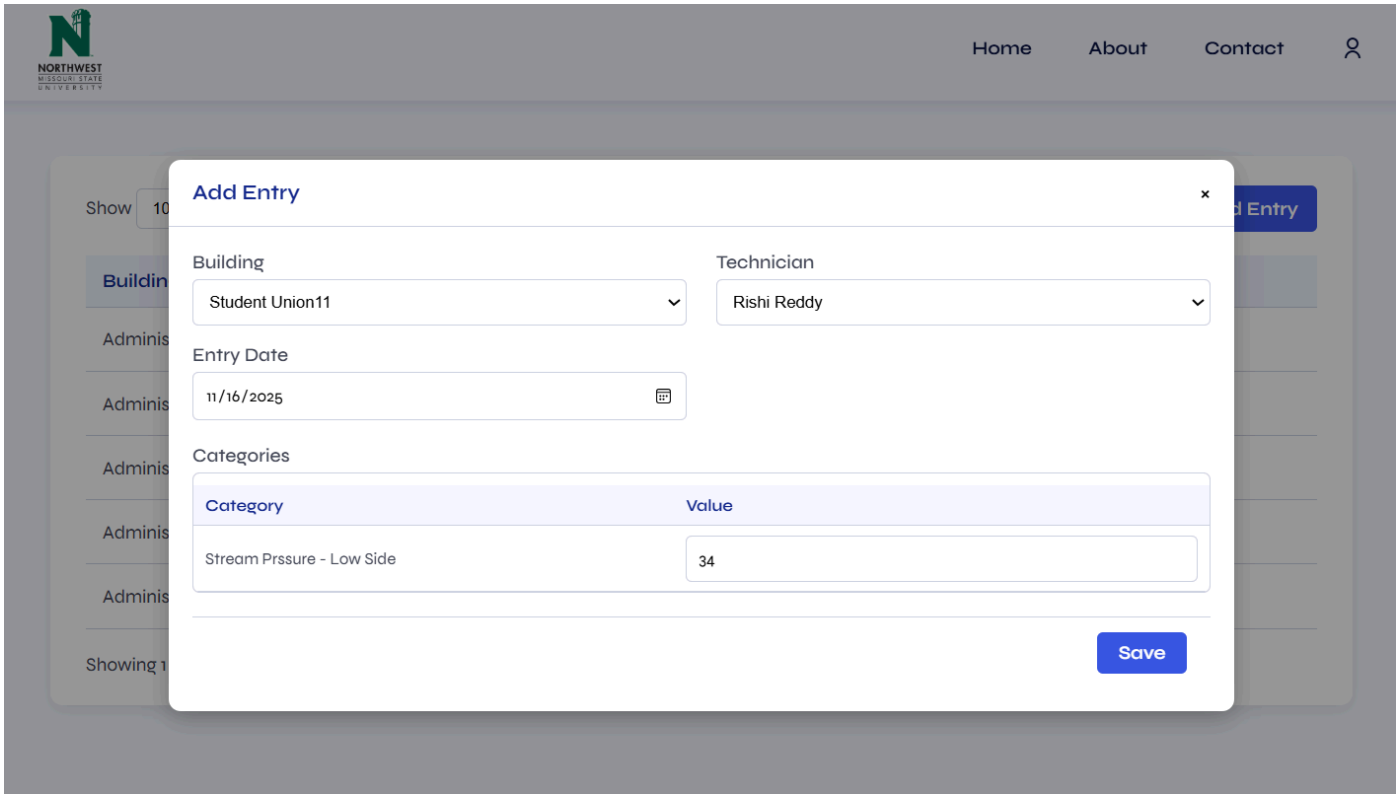


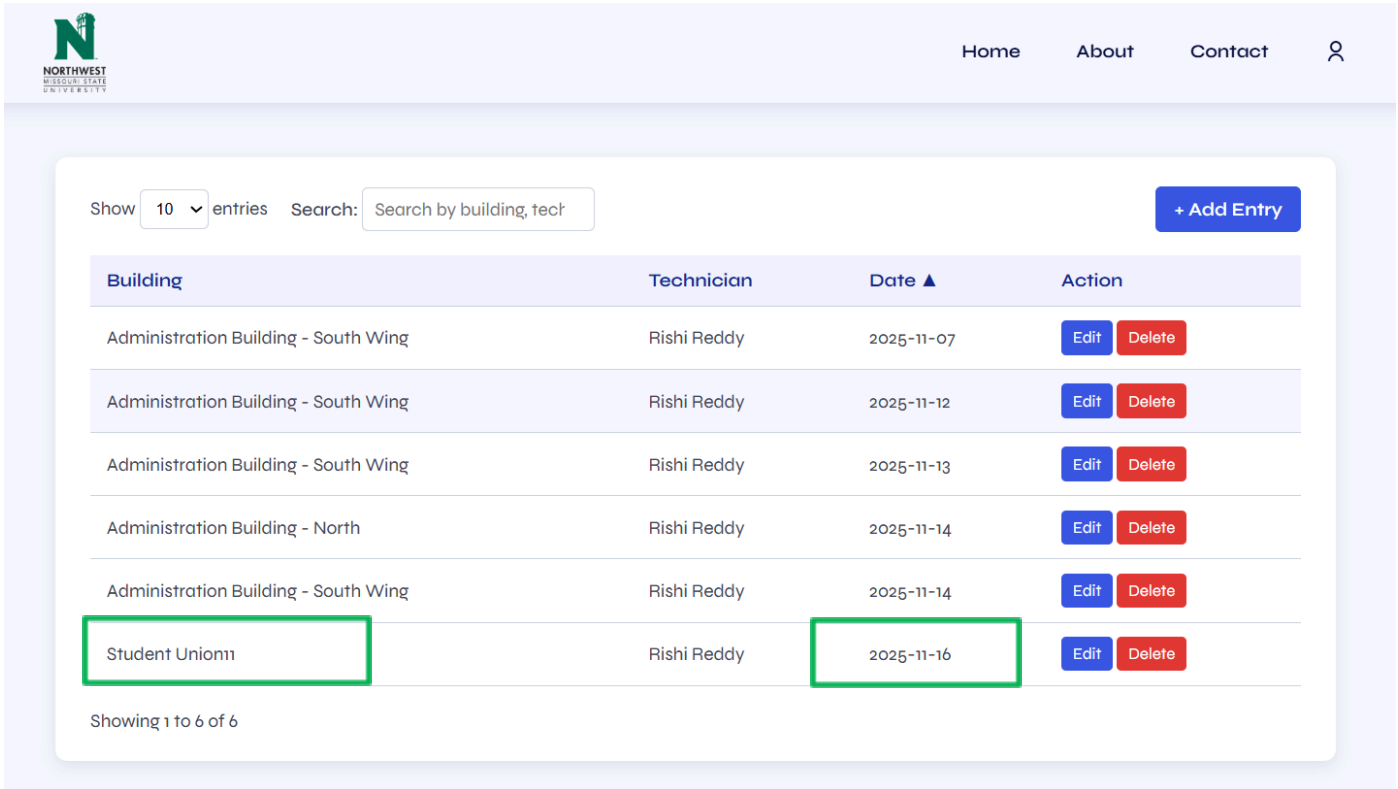
- After edit the values for category fields are changed and saved.



- Admin can **Add** a new data entry if required.

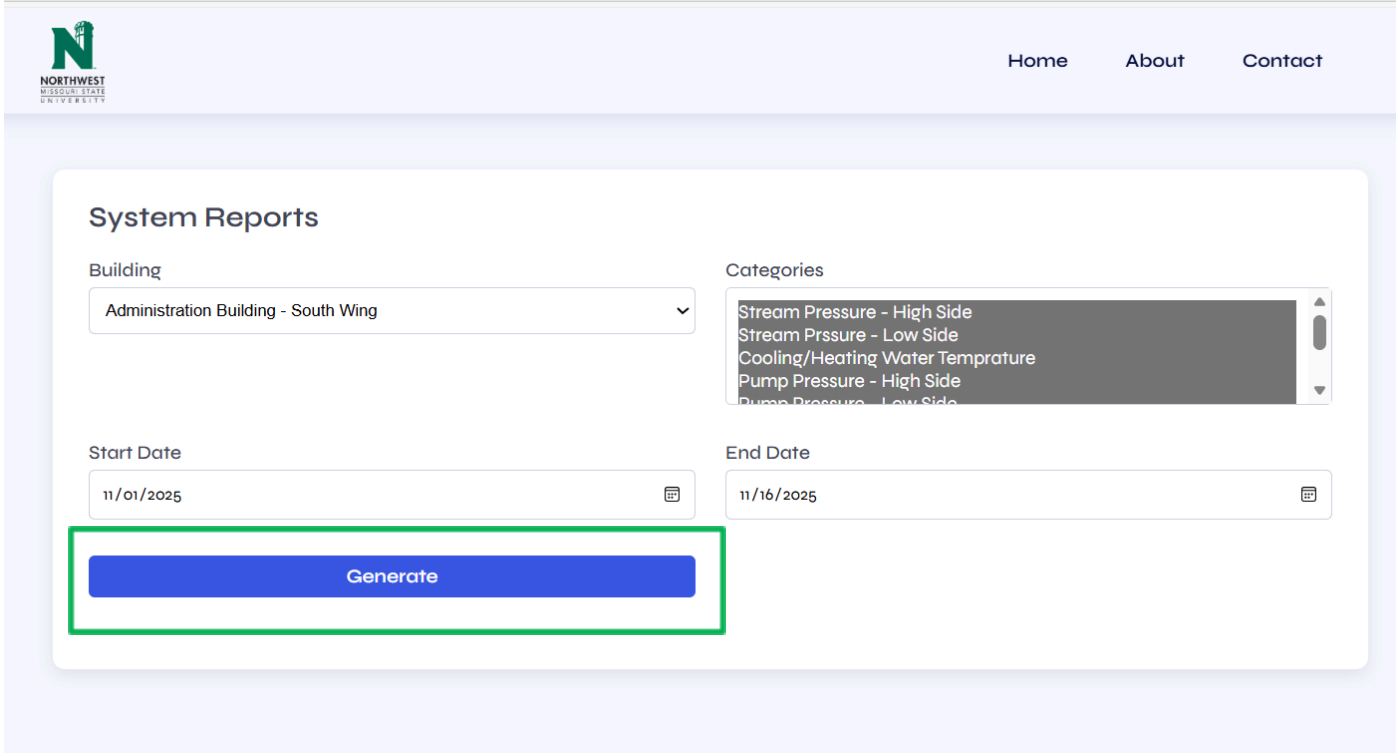


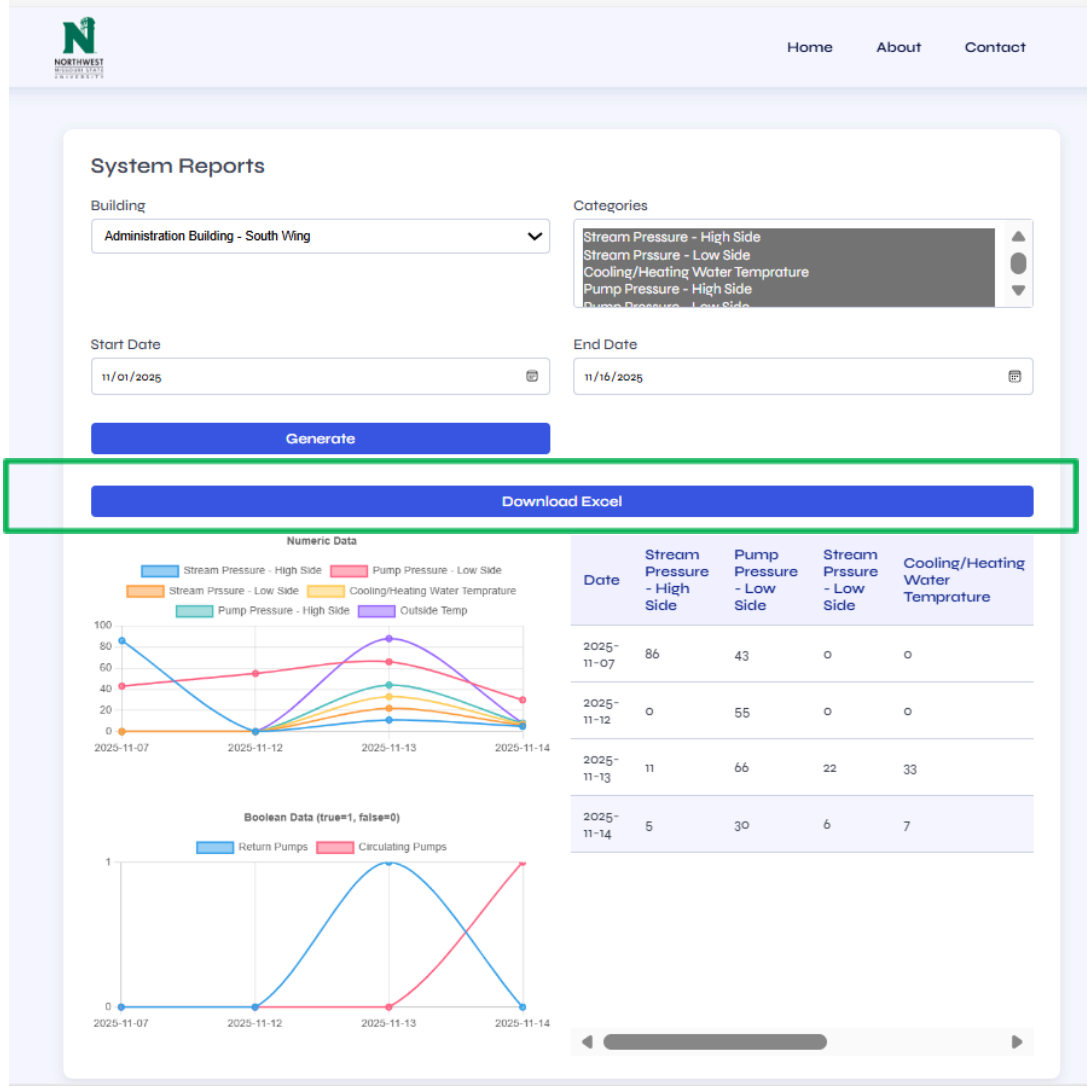

Home
About
Contact
Add Entry
Building
Student Union11
Technician
Rishi Reddy
Entry Date
11/16/2025
Categories
Category
Value
Stream Prssure - Low Side
34
Save


Home
About
Contact
Show 10 entries
Search: Search by building, tech
+ Add Entry
Building
Technician
Date ▲
Action
Administration Building - South Wing
Rishi Reddy
2025-11-07
Edit
Delete
Administration Building - South Wing
Rishi Reddy
2025-11-12
Edit
Delete
Administration Building - South Wing
Rishi Reddy
2025-11-13
Edit
Delete
Administration Building - North
Rishi Reddy
2025-11-14
Edit
Delete
Administration Building - South Wing
Rishi Reddy
2025-11-14
Edit
Delete
Student Union11
Rishi Reddy
2025-11-16
Edit
Delete
Showing 1 to 6 of 6

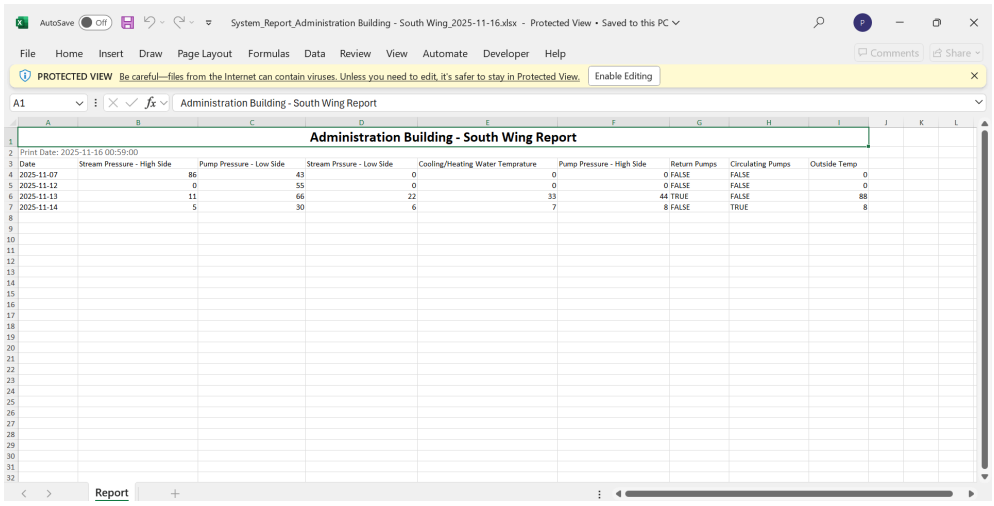
- **Reports**
  Generate reports based on stored data

- **Graphs** - Select the building, data field(s), and the start and end dates to generate visual graphs.
- **Reports/ Excel Download** - Select the building, data field(s), and the start and end dates to download the data in Excel format.



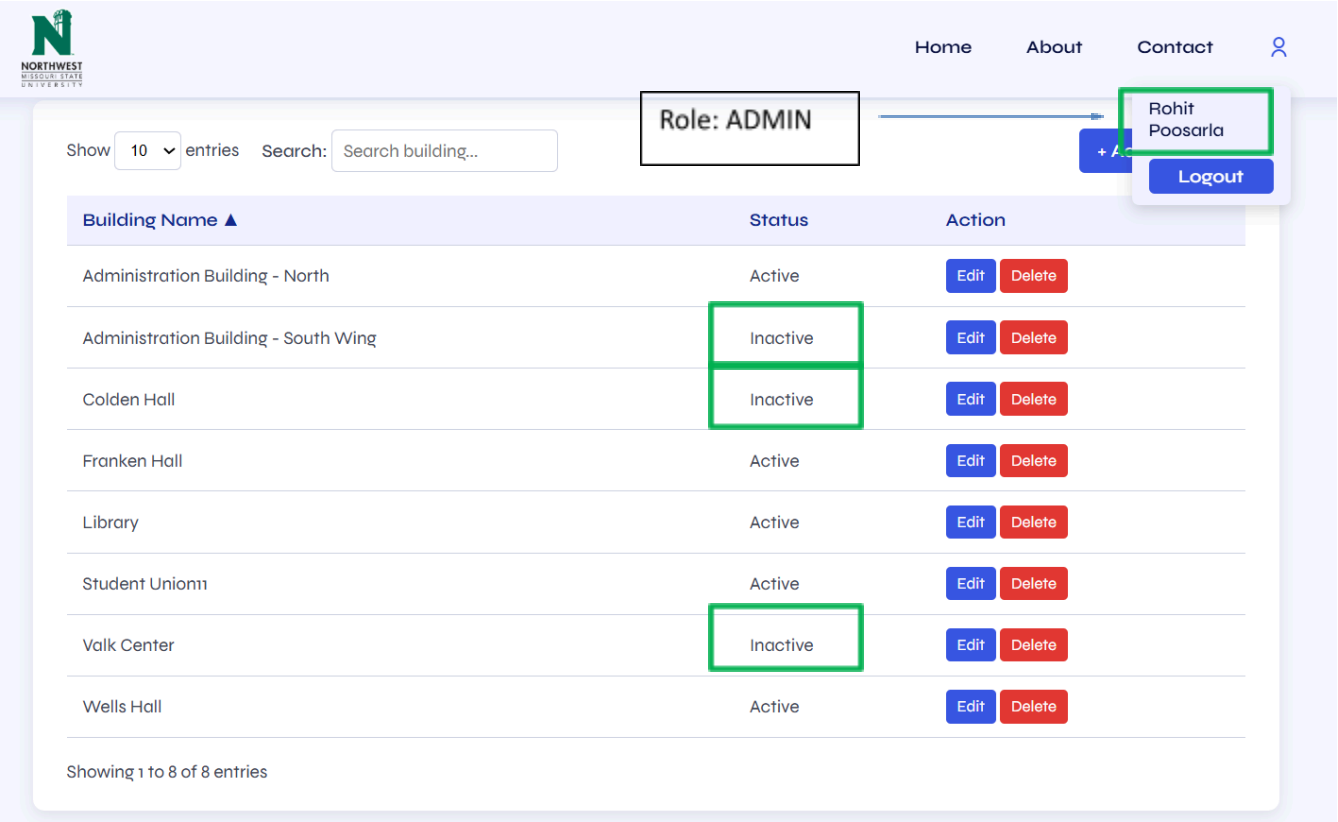


By clicking on **Download Excel,** we can get data in Excel format.



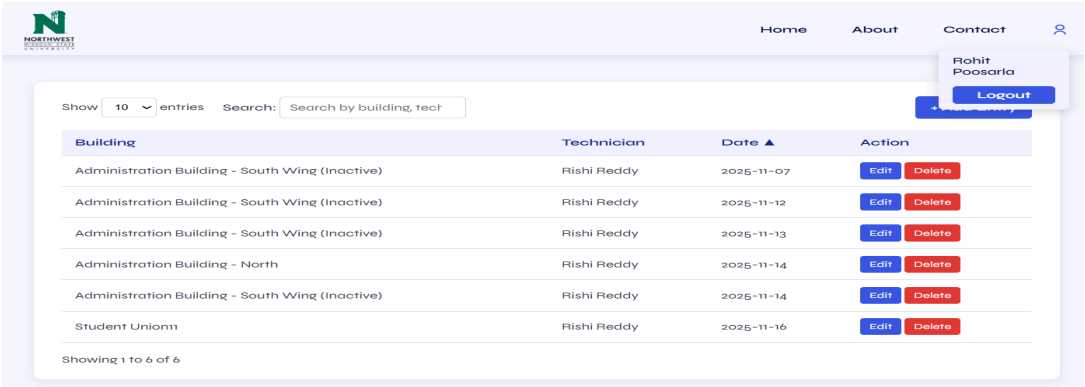
- **Building Status (Active / Inactive)**

1. when a building is marked as **Inactive**, it will **not appear** in the dropdown list for Technicians while performing inspections.
2. Only **Active** buildings are available for selecting and filling data entry forms.



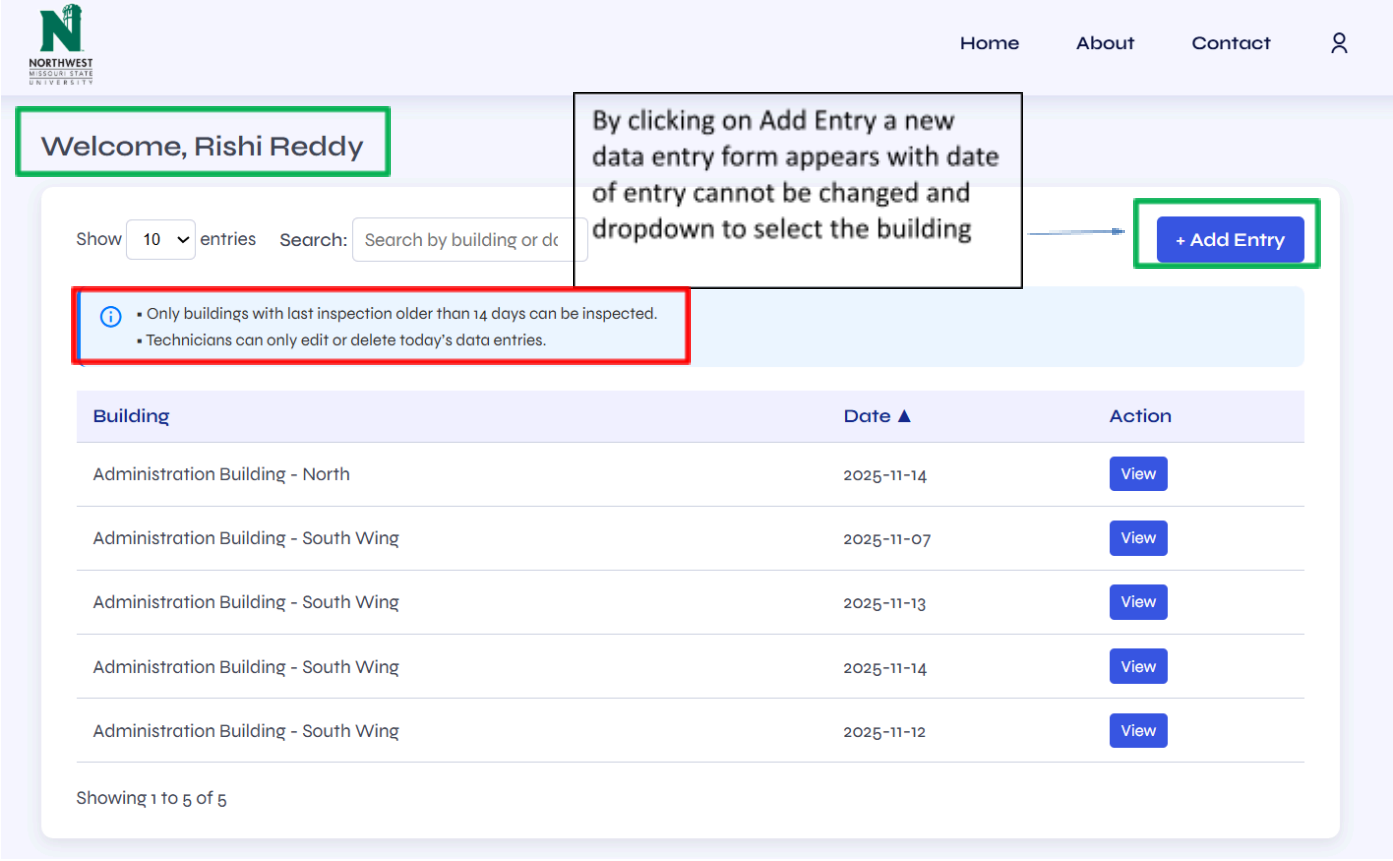
2. Display for Admin

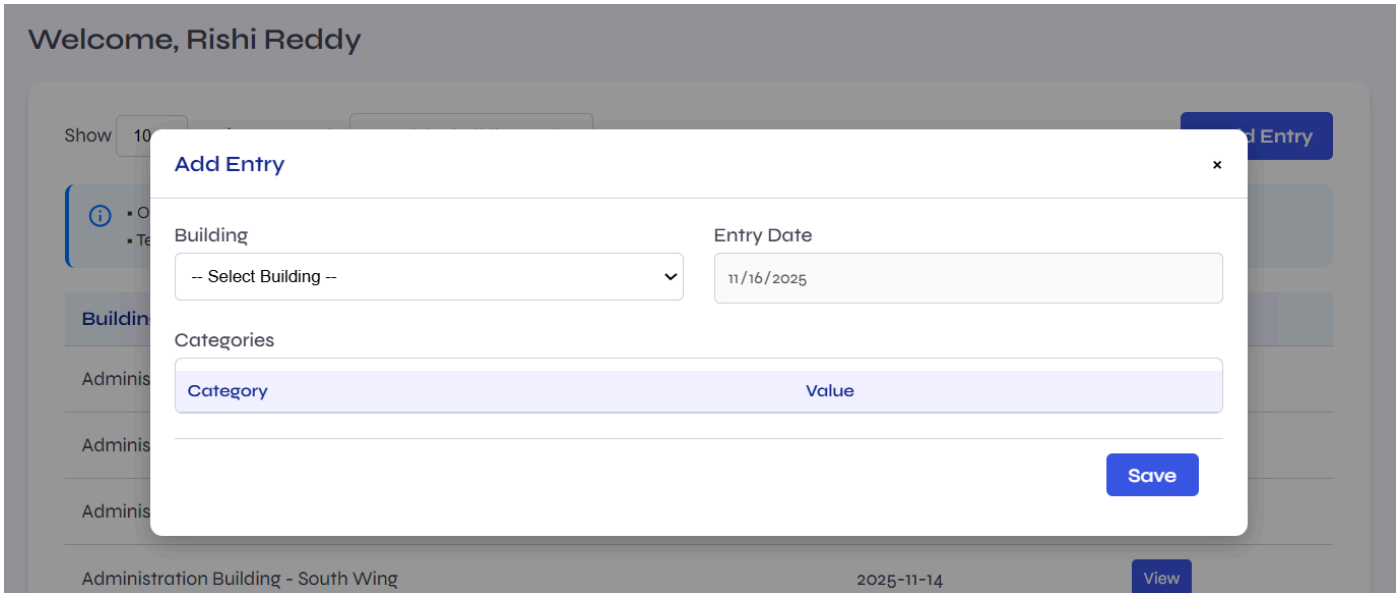
- If a building has previous inspection data, the admin can still view those entries.
- For such buildings, the admin will see an **"Inactive" status label** beside the building name, indicating the building is currently inactive

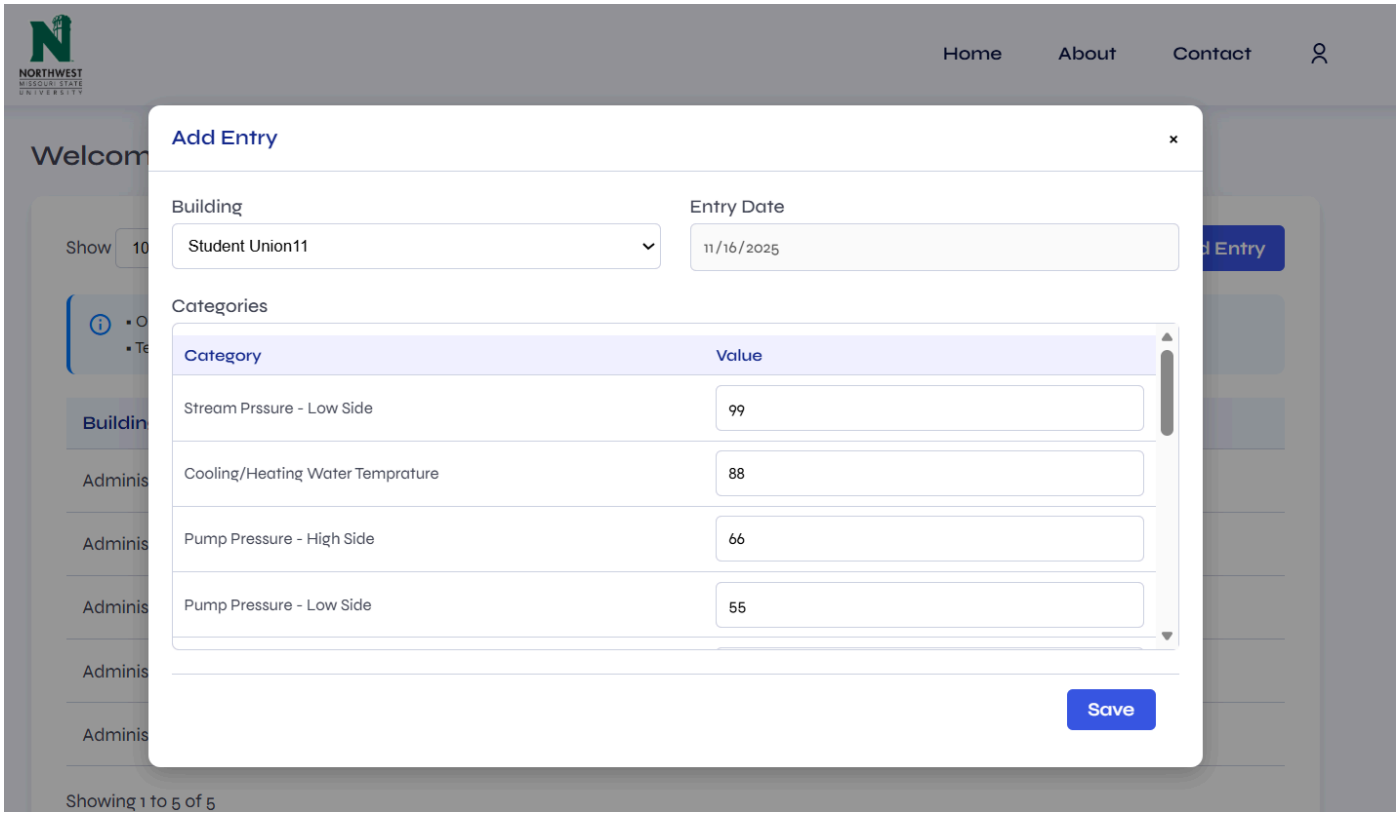


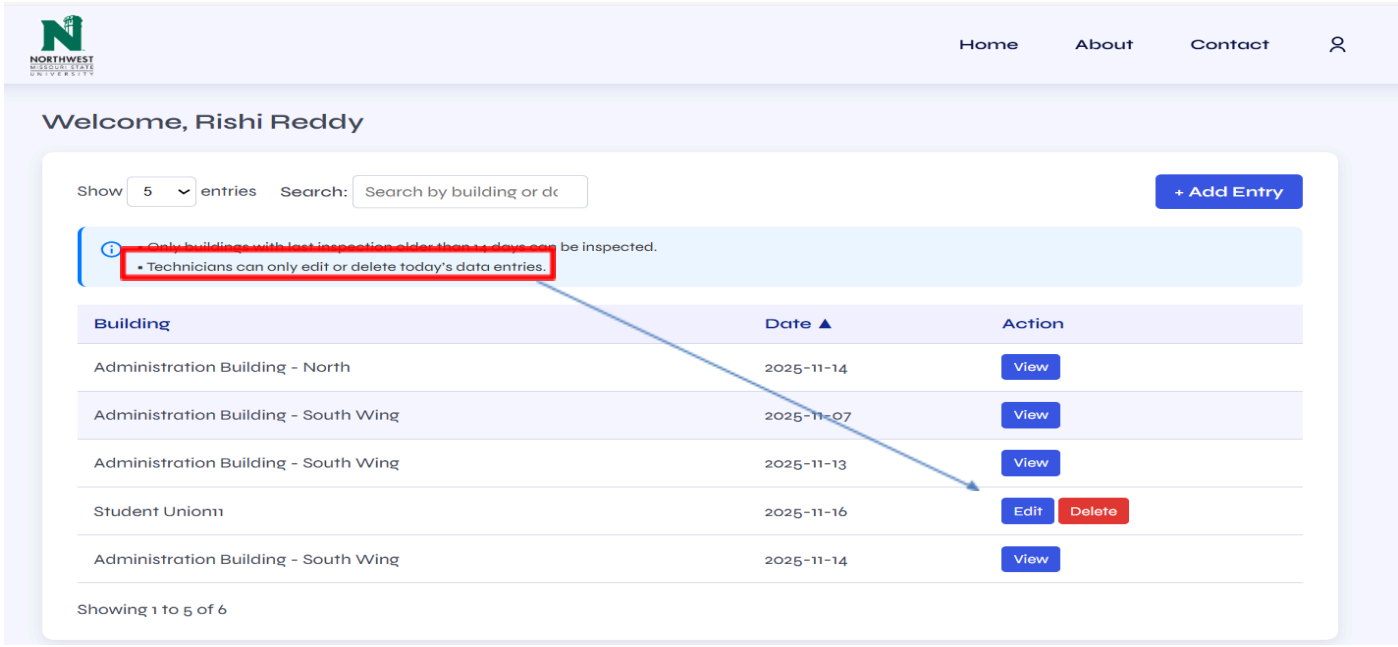
Technician Functionalities:

- Enter new data entry form while he was inspecting the building he needs to enter the values in particular category field for that building.

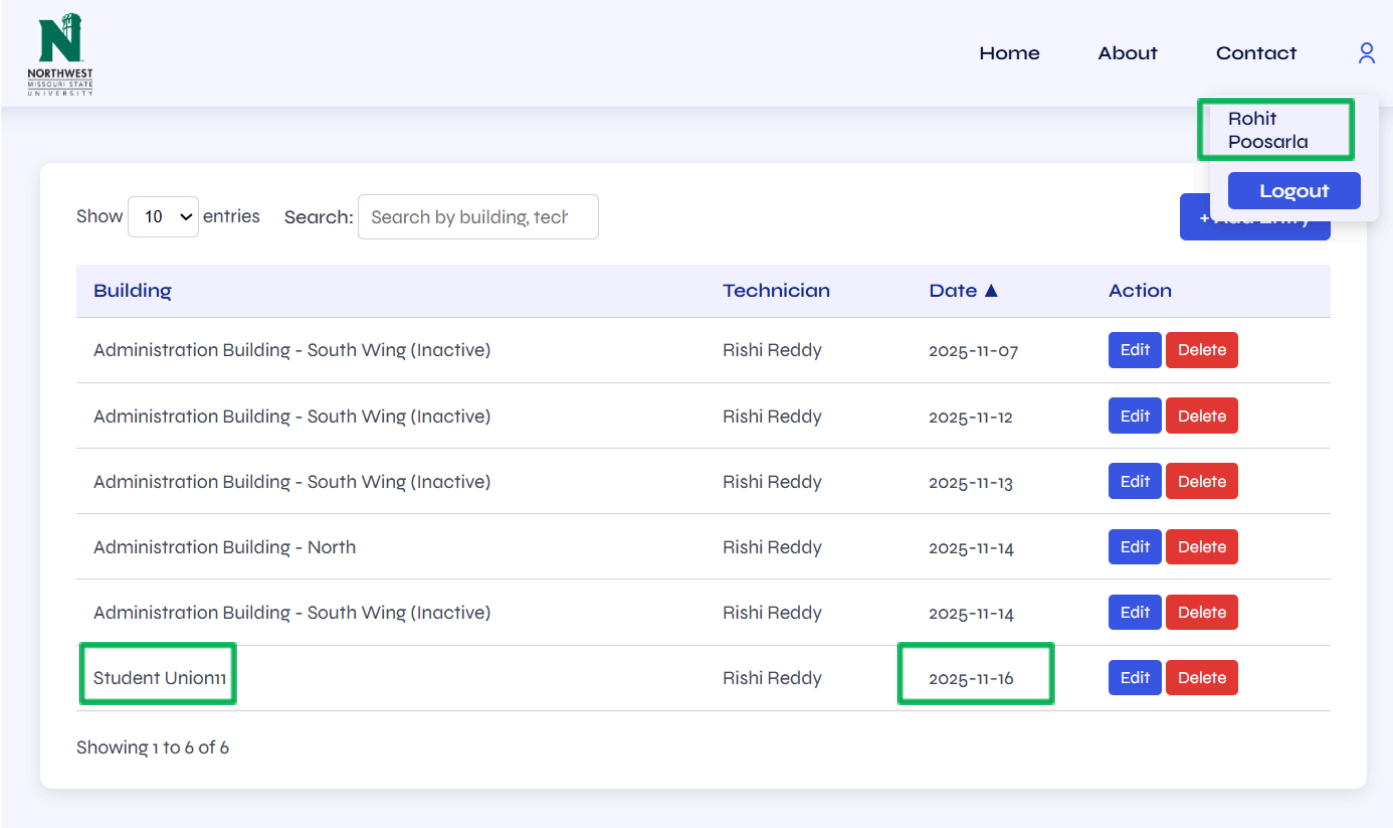





Home
About
Contact
Welcom
Add Entry
Building
Student Union11
Entry Date
11/16/2025
Categories
Category
Value
Stream Prssure - Low Side
99
Cooling/Heating Water Temprature
88
Pump Pressure - High Side
66
Pump Pressure - Low Side
55
Save
Showing 1 to 5 of 5

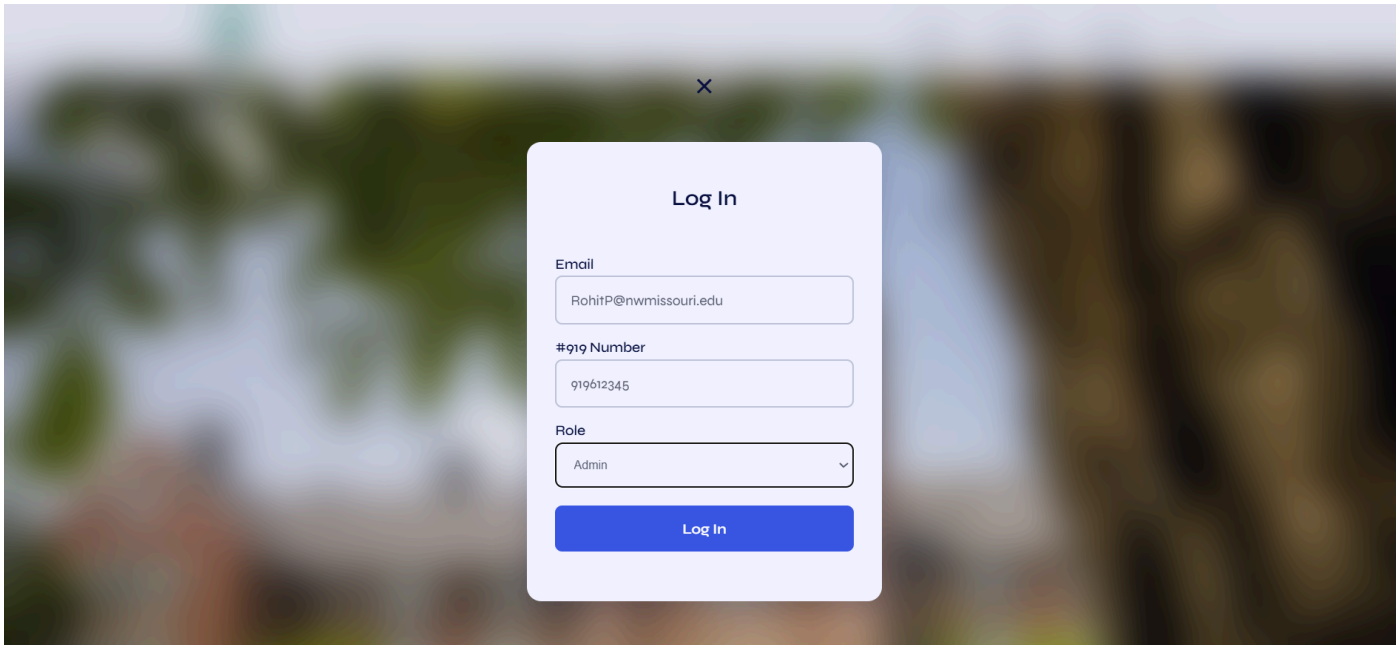

Home
About
Contact
Welcome, Rishi Reddy
Show 5 entries
Search:
Search by building or dc
+ Add Entry
Technicians can only edit or delete today's data entries.
Building
Date ▲
Action
Administration Building - North
2025-11-14
View
Administration Building - South Wing
2025-11-07
View
Administration Building - South Wing
2025-11-13
View
Student Union11
2025-11-16
Edit
Delete
Administration Building - South Wing
2025-11-14
View
Showing 1 to 5 of 6

- **Admin** can be able to see the latest data entry and can also generate reports accordingly.

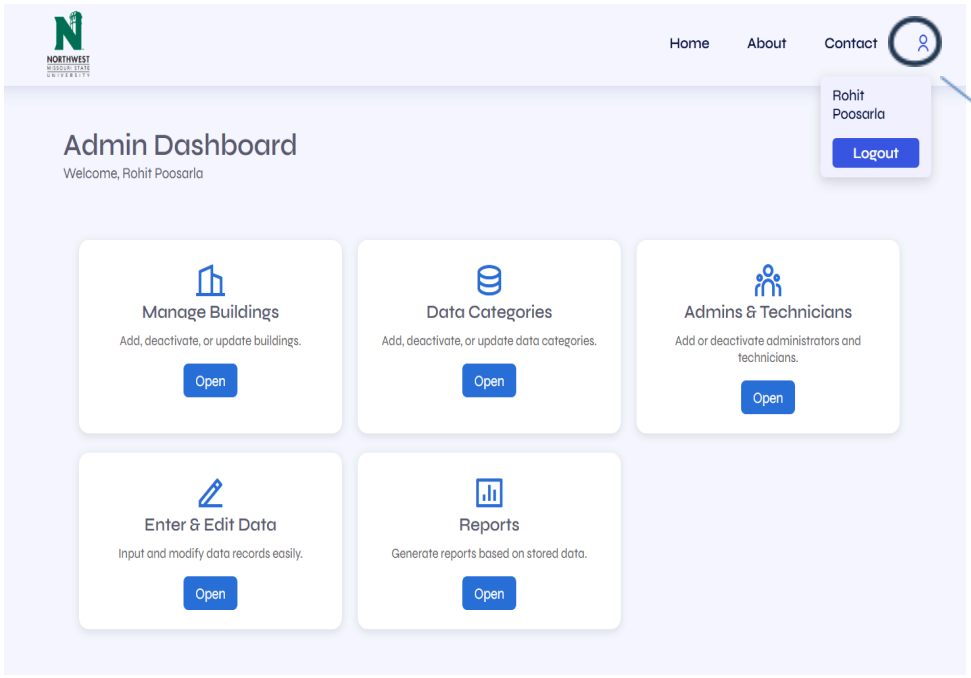


5.2 System Testing

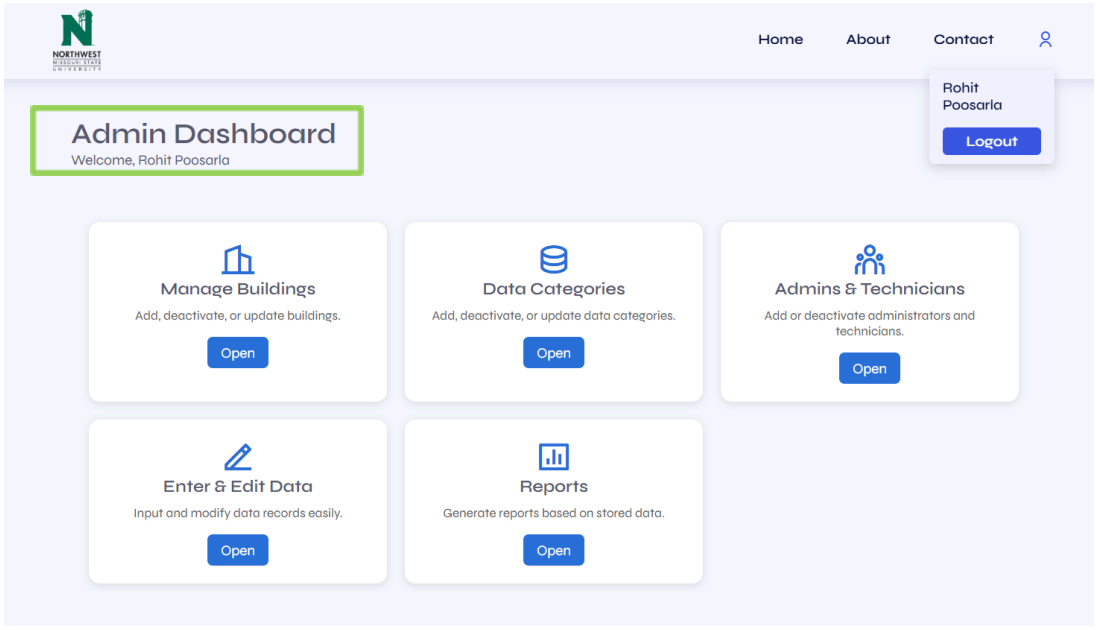
Below are screenshots taken after the manual testing of the app.



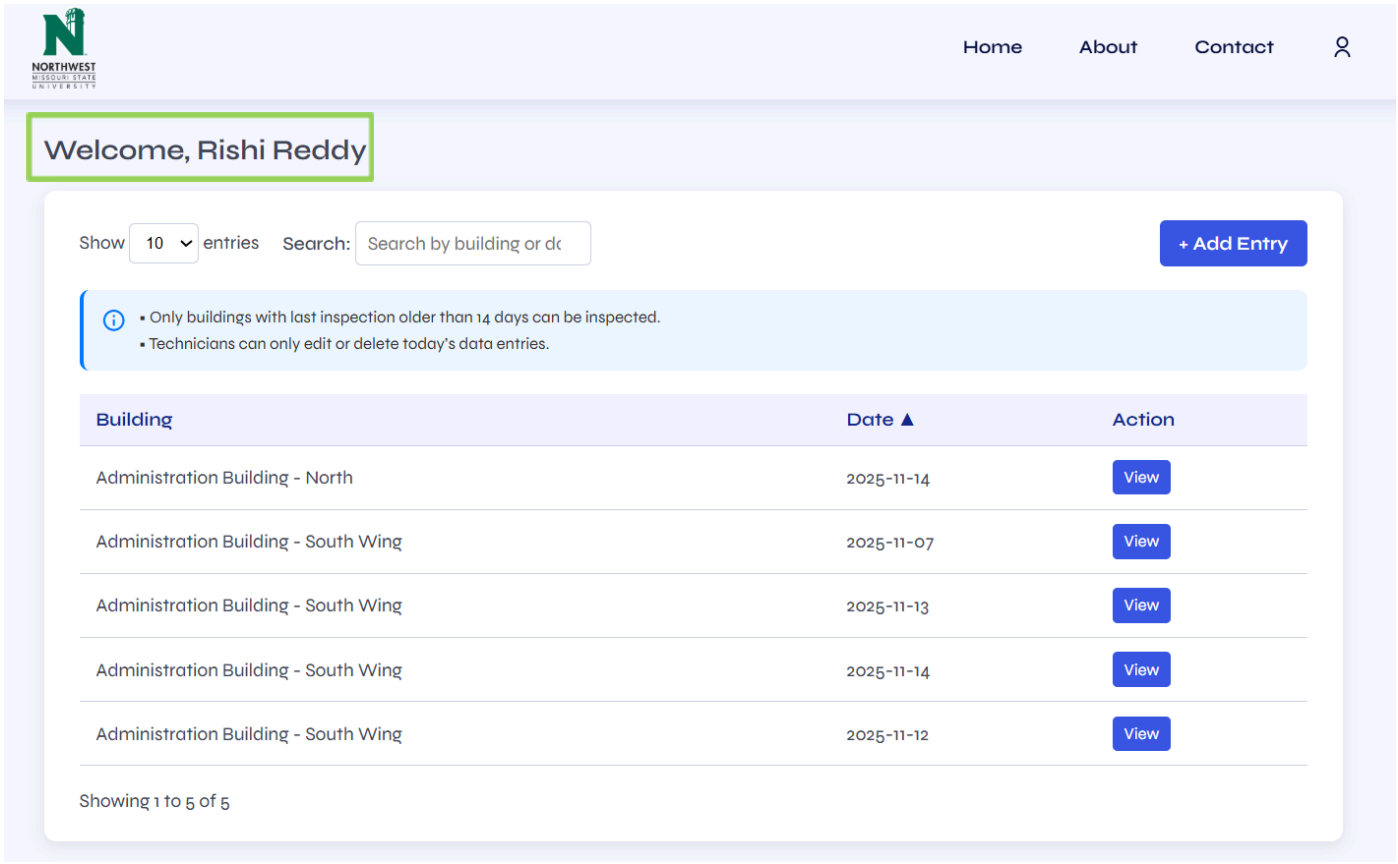
This indicates that the login system for the Northwest Missouri State University's Maintenance Technician Building Rounds system is restricted to users with a specific "Bearcat" email address and #919 number and Role requires authentication through a Microsoft account.



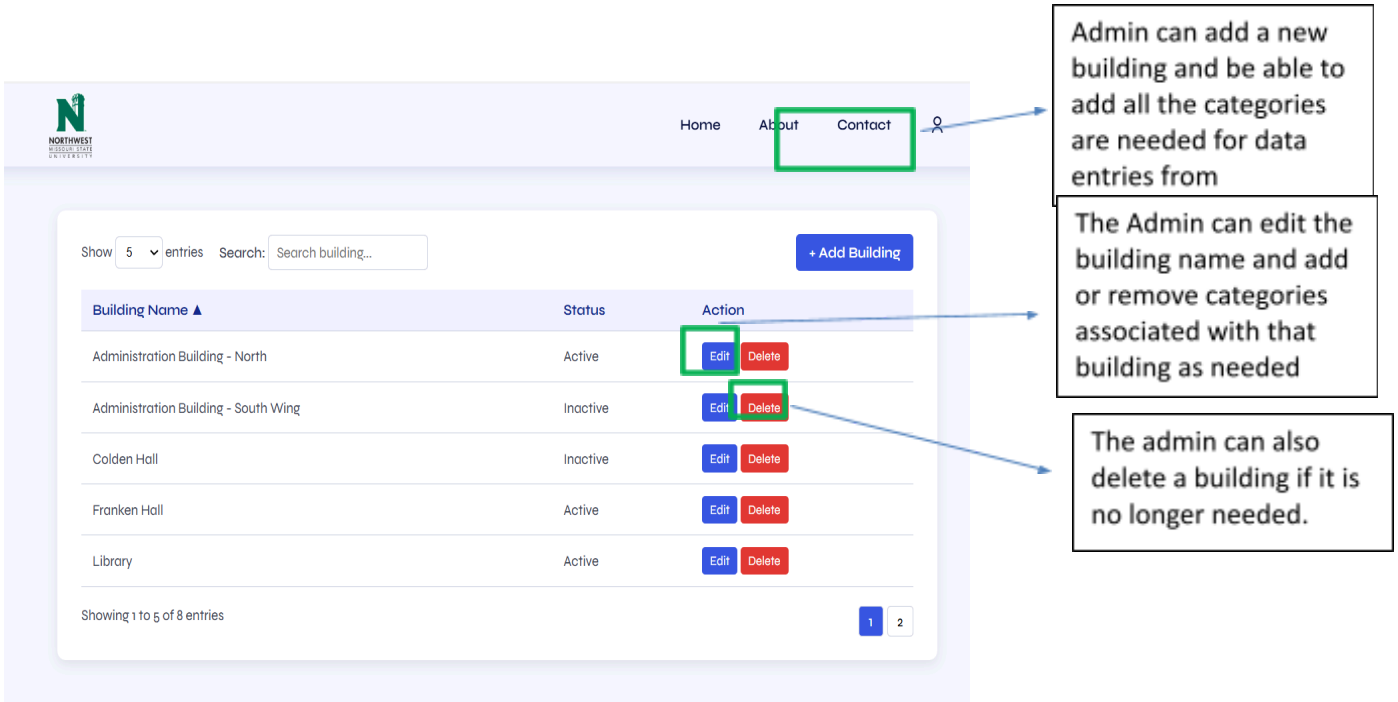
The image confirms that the user profile is visible and that there is a logout option. In the top right corner of the image, you can see "Your profile name" and "Log out" options under the user's name

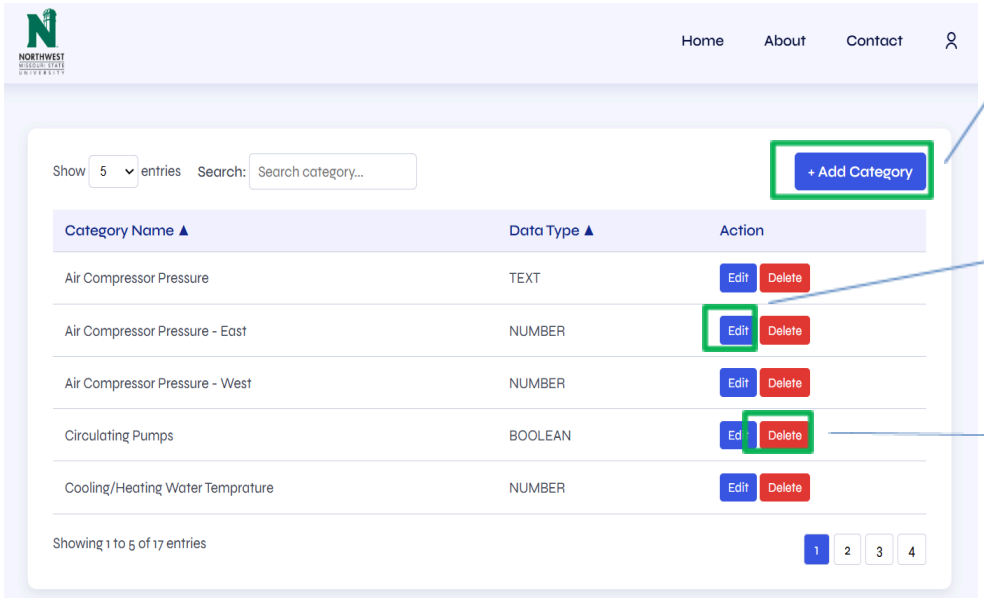


The image shows the ADMIN dashboard of the Campus Maintenance and Management System. It provides access to key functions like Manage Buildings, Data Categories, Admins & Technicians, Enter & Edit Data, Reports.



The image displays the TECHNICIAN dashboard of the Campus Maintenance and Management System, where technicians can access essential features, including entering data for buildings during inspections.

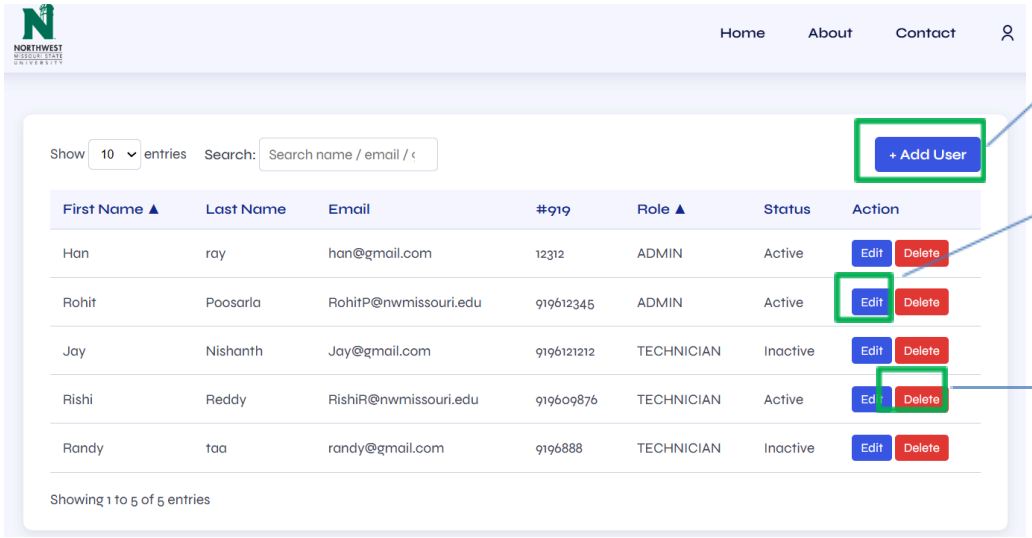




The admin can add new categories with name and data type.

The admin can edit the category name, and its data type as needed.

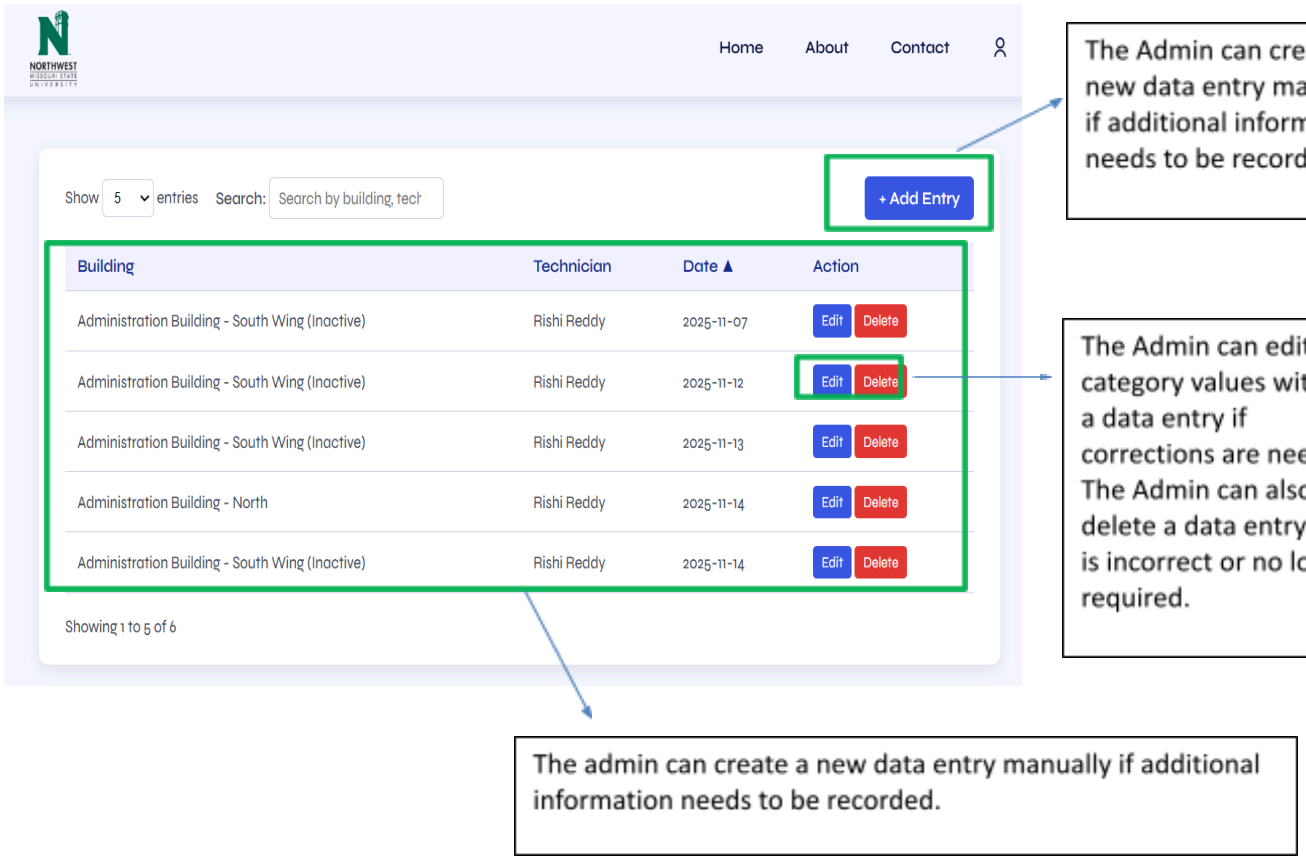
The admin can also delete a category if it is no longer required.



The Admin can create a new user by assigning a name, email, #919 number, and role (Admin or Technician)

The Admin can update user information such as name, email, role, or status (Active/Inactive).

The Admin can delete a user if they are no longer needed, or mark them as **Inactive** to prevent them from logging into the system.



6. Result of the Project

+. Technicians can now record inspection data in real time using a web-based system, reducing errors and saving time. Administrators can manage buildings, categories, and users with ease, and generate detailed reports and graphs for performance tracking.

The system improved accuracy, streamlined data entry, and provided better visibility into building conditions across campus. Real-time data availability, role-based access, and automated reporting helped enhance decision-making and overall operational efficiency. By replacing paper logs with a centralized digital solution, the project achieved faster maintenance workflows, improved sustainability, and a more organized maintenance management process for the university.