

# **INTERNSHIP PROJECT REPORT**

**Internship Duration**

**5<sup>th</sup> JUNE 2023 – 31<sup>st</sup> JULY 2023**

**Done at**

**IT Division, Central Pollution Control Board, Delhi**

*Submitted by*

**MRIGA ARORA**  
(Registration Number: 200905197)

*Student of*

**BACHELOR OF TECHNOLOGY**  
*In*  
**Computer Science Engineering**

**MANIPAL INSTITUTE OF TECHNOLOGY, MANIPAL  
MANIPAL – 576104.**

# **INTERNSHIP PROJECT REPORT**

*Submitted in partial fulfillment of the requirements for the award of the degree*

*Of*

## **BACHELOR OF TECHNOLOGY**

*In*

**Computer Science Engineering**

**From**

**MANIPAL INSTITUTE OF TECHNOLOGY, MANIPAL**

**MANIPAL – 576104.**

**Project Title**

**Data Centre Servers Management System**

**At**

**IT Division**

---

**CENTRAL POLLUTION CONTROL BOARD**

---

**From**

**5<sup>th</sup> JUNE 2023 – 31<sup>st</sup> JULY 2023**

**Submitted by:**

Mriga Arora  
200905197  
B. Tech, CSE  
MIT, Manipal

**Submitted to:**

Divisional Head,  
IT Division



# MANIPAL INSTITUTE OF TECHNOLOGY

## MANIPAL

(A constituent unit of MAHE, Manipal)

## I. ACKNOWLEDGEMENT

I am deeply humbled by the opportunity to work as a summer intern at the Central Pollution Control Board. Throughout this enriching experience, I dedicated myself to continuously upskilling in programming and effectively applying development concepts to address real-world challenges. It was during this internship that I discovered my passion for full stack development, allowing me to combine my fields of interest seamlessly.

I would like to express my utmost gratitude to **Sh. B. Vinod Babu, Sc. F and IT Divisional Head** and to my esteemed guide, **Sh. Archit Uprit, Sc. D, IT Division**, along with **Sh. Abhishek Kumar Pandey, Senior Software Developer** and **Sh. Naman Srivastava, DPA**, for their invaluable mentorship and insightful explanations regarding the project requirements. Their guidance not only broadened my technical knowledge in my chosen field but also provided me with the opportunity to collaborate and thrive within a professional team in an academic setting. I am truly indebted to **Ms. Pooja Rani** for her kind recommendations, without which this internship would not have been possible. I would like to extend my heartfelt gratitude to the MIT, Manipal CSE department and the Head of Department for providing me with this incredible opportunity. Finally, to thank my parents and my peers for their constant support throughout my internship.

Mriga Arora  
200905197  
B. Tech CSE  
Manipal Institute of Technology, Manipal



# MANIPAL INSTITUTE OF TECHNOLOGY

## MANIPAL

(A constituent unit of MAHE, Manipal)

## II. DECLARATION

I, **Mriga Arora**, solemnly declare that the project report, DATA CENTRE SERVERS MANAGEMENT SYSTEM, is based on my own work carried out during the course of our study under the supervision of **Sh. Archit Uprit, Sc. D, IT department and CPCB**. I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that:

1. The work contained in the report is original and has been done by me under the supervision of my supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.
4. Whenever we have used materials (text, data, theoretical analysis/equations, codes/program, figures, tables, pictures, text etc.) from other sources, we have given due credit to them in the report and have also given their details in the references.

Mriga Arora  
200905197  
B. Tech CSE  
Manipal Institute of Technology, Manipal

### **III. ABSTRACT**

This project aims to create a web application that facilitates the effective management of server racks, companies, and servers. With the help of Django framework, the application provides a user-friendly interface for adding, editing, and deleting these entities, ensuring efficient organization and maintenance of IT infrastructure.

The application incorporates authentication mechanisms to ensure secure access and data integrity. Users can log in, granting them the ability to perform authorized operations. The intuitive interface allows users to seamlessly navigate through various functionalities.

The system allows users to manage server racks by adding, editing, or deleting them as needed. It also enables the association of companies with specific racks, providing a comprehensive overview of the infrastructure. Users can add servers to the system, specifying details such as make, model, serial number, ownership, warranty/AMC status, and other relevant information. Additionally, the application allows users to search for specific servers based on applications installed, portals running, IP addresses, and more.

Overall, this project simplifies server management and enhances the efficiency of IT operations. It provides a centralized platform for monitoring and maintaining server infrastructure, aiding organizations in effectively managing their resources and ensuring smooth functioning of their IT systems.

## IV. ORGANIZATION

The Central Pollution Control Board (CPCB) is a prominent organization responsible for maintaining and safeguarding environmental quality in India. As the apex regulatory body under the Ministry of Environment, Forest, and Climate Change, the CPCB plays a vital role in formulating policies, coordinating environmental programs, and implementing pollution control measures across the country.

Within the CPCB, the Information Technology (IT) division holds significant importance. The IT division is entrusted with the task of developing and managing technological solutions to enhance the efficiency and effectiveness of environmental monitoring, data analysis, and reporting systems. By leveraging advanced technologies, such as data analytics, networking, and web-based applications, the IT division empowers the CPCB in collecting, analyzing, and disseminating environmental data in a timely and accessible manner.

The objectives of the IT division revolve around enhancing environmental governance and decision-making processes. These objectives include:

1. Developing and maintaining robust information systems for comprehensive environmental data management.
2. Implementing advanced data analytics techniques to derive valuable insights for informed policy formulation.
3. Designing user-friendly web-based portals and mobile applications for easy access to environmental information by stakeholders and the public.
4. Strengthening the IT infrastructure to ensure secure and efficient data storage, transmission, and retrieval.
5. Collaborating with national and international agencies to adopt best practices in environmental IT systems.
6. Regular Monitoring and Maintenance: The CPCB IT division conducts regular monitoring of their servers to detect any issues or performance bottlenecks. They perform routine maintenance tasks such as software updates, security patches, and system optimizations to ensure the servers are running smoothly.

Through its IT division, the CPCB strives to integrate technology and environmental management, facilitating informed decision-making and fostering sustainable development practices across the nation.

## TABLE OF CONTENTS

CHAPTERS	PAGE NUMBER
ACKNOWLEDGEMENT	2
ABSTRACT	3
DECLARATION	4
ABOUT ORGANIZATION	5
PROBLEM STATEMENT & OBJECTIVES	7
TOOLS AND TECHNOLOGIES USED	8
DATA FLOW DIAGRAM	9
USER INTERFACE	12
MODELS	13
FEATURES AND FUNCTIONALITIES	15
CONCLUSION	28
FUTURE SCOPE	29
REFERENCES	30

## V. PROBLEM STATEMENT AND OBJECTIVES

### **Problem Statement:**

Managing server infrastructure efficiently is a crucial challenge for organizations. It often involves tracking server racks, associated companies, and individual servers, which can be time-consuming and prone to errors when done manually. There is a need for a reliable and user-friendly web application that streamlines server management processes, improves organization, and enhances overall productivity.

### **Objectives:**

1. Develop a web application using Django framework to facilitate the management of server racks, companies, and servers.
2. Implement user authentication mechanisms to ensure secure access and data integrity.
3. Enable users to add, edit, and delete server racks, companies, and servers through an intuitive interface.
4. Allow the association of companies with specific server racks, providing a comprehensive overview of the infrastructure.
5. Provide a search functionality to allow users to find specific servers based on various criteria, such as applications installed, portals running, IP addresses, etc.
6. Enhance the user experience by offering smooth navigation and intuitive interactions within the application.

By achieving these objectives, the web application will address the challenges associated with server management, streamline processes, and enhance the overall effectiveness of IT operations within organizations.

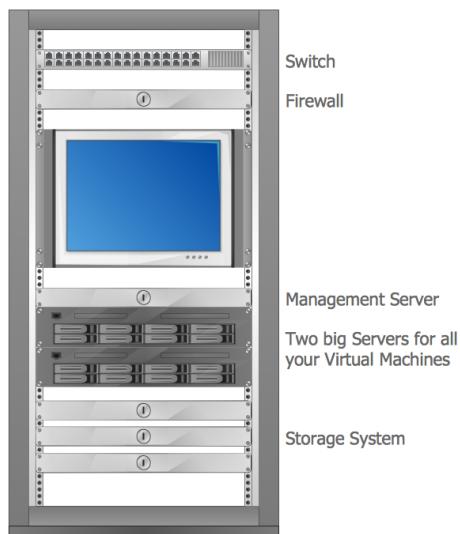


Figure 1

## VI. TOOLS AND TECHNOLOGUES USED

1. **StarUML:** StarUML, a powerful modeling tool, was employed to design the Data Flow Diagram (DFD) for the server rack management system. By using StarUML, the project team visualized the flow of data within the application, ensuring a clear understanding of how information moved between different components.
2. **Figma:** Figma was utilized for designing the user interface (UI) of the server rack management system. With its collaborative features, the design team could create visually appealing and interactive UI designs. Figma's web-based platform allowed for seamless collaboration and prototyping, resulting in an intuitive and user-friendly interface.
3. **Django:** (4.1.2) This powerful python (Python 3.11.2 version used) web framework was employed to develop the server rack management system. By leveraging Django's features and capabilities, the application was able to efficiently connect and manage racks, servers, and companies. Django's built-in functionalities for handling user authentication, database management, and routing were utilized to create a seamless and robust system.
4. **HTML:** HTML played a pivotal role in constructing the structure and content of the web pages. It allowed for the creation of well-organized and semantically structured pages, ensuring a clear representation of the server rack management system's components.
5. **CSS:** CSS was utilized to enhance the visual presentation of the web pages. By applying CSS styles, the application's interface was customized, enabling an aesthetically pleasing and consistent design across different pages. CSS helped in achieving a professional and user-friendly look and feel for the system.
6. **JavaScript:** JavaScript was incorporated to add interactivity and dynamic functionality to the server rack management system. It facilitated tasks such as form validation, handling asynchronous requests, and creating interactive elements that improved the user experience. JavaScript contributed to the overall responsiveness and usability of the application.
7. **SQLite:** The project utilized SQLite as the backend database management system. SQLite's lightweight nature and simplicity made it a suitable choice for storing and retrieving data related to racks, servers, and companies. It enabled efficient data management within the application, ensuring seamless access and manipulation of information.
8. **Source-Code Editor:** Visual Studio Code 2

## VII. DATA FLOW DIAGRAM

1. **Definition:** A DFD (Data Flow Diagram) is a visual or graphical representation of data functions and the interchange among those functions using a standardized set of symbols and notations for ease of understanding for the customer.
2. **Software Used:** StarUML is a software modelling tool mainly used to make use cases. Here it has been used to make a DFD. The process for making a DFD in StarUML is:
  - a) Select first an element where a new Data Flow Diagram to be contained as a child.
  - b) Select **Model | Add Diagram | Data Flow Diagram** in Menu Bar or select **Add Diagram | Data Flow Diagram** in Context Menu.

Further, the data flow and annotations that are easy to navigate are used to make the DFD. Under the Data Flow dropdown, the symbols available are: external entity, process, data store and data flow. Under the annotations: text, text box, free line, note, note link , hyperlink, rectangle, rounded rectangle, ellipse and image.

### 3. Diagrams:

Abbreviations used:

- a. Info: information
- b. CRUD: Create, Read, Update, Delete
- c. EmpID: Employee Identification

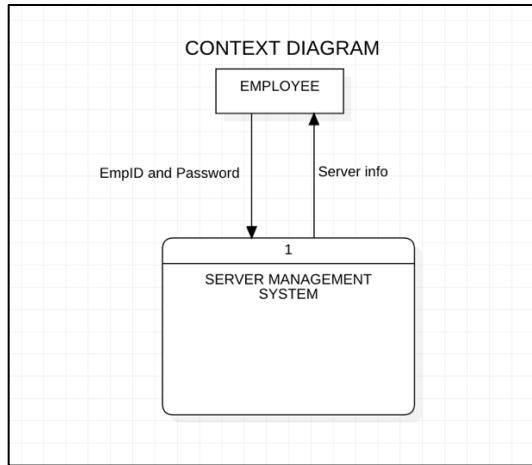


Figure 2

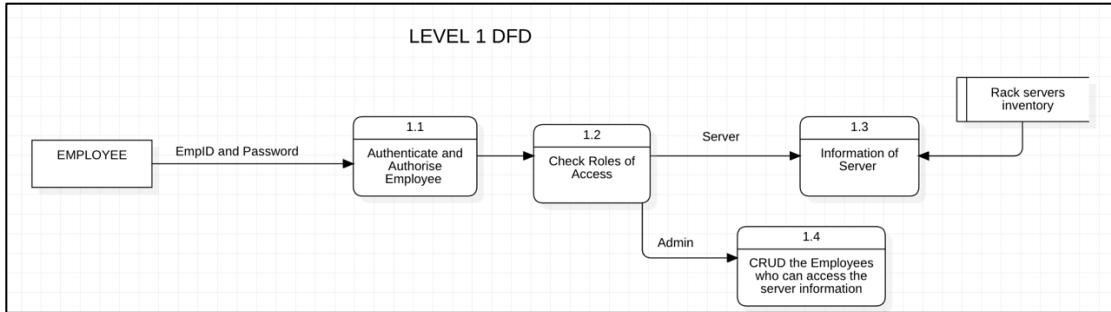


Figure 3

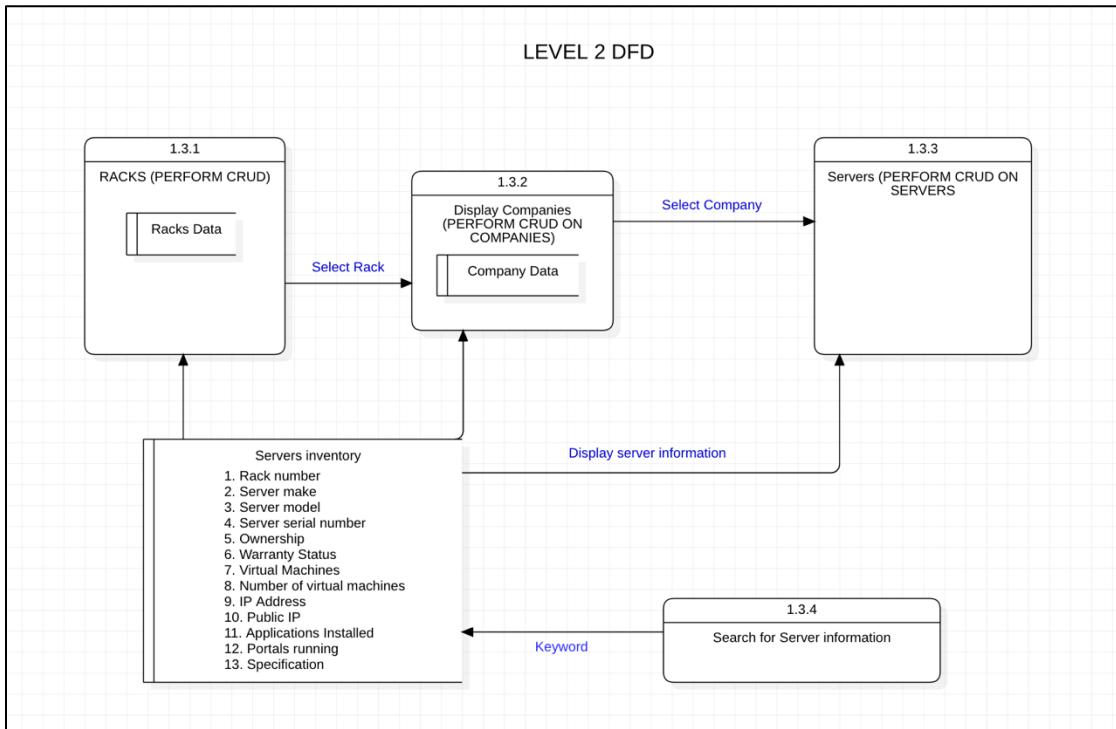


Figure 4

#### 4. Data Dictionary:

1. Employee: Represents an employee of Central Pollution Control Board that is allowed to create, read, update and delete server information of the company.
2. Admin: Represents an employee of CPCB that has further access to the management system.
3. Server Info: This Displays the server information that has been mentioned in the server inventory.
4. Authenticate and Authorise:
  - Input: User credentials (username and password)
  - Processes: Authenticates the user and creates a session.

- Output: Redirects the user to the Server Information if authenticated, or displays an error message.

5. Racks:

- Input: None
- Processes:
  - Retrieves data from the database(Racks, Server)
  - Renders the racks page with retrieved data and allows us to perform CRUD operations on the following data.
- Output: Display the rack page with retrieved data.

6. Companies:

- Input: Rack ID
- Processes:
  - Retrieves data from the database(Racks, Company, Server)
  - Renders the company page with matched rack id with company and retrieved data and allows us to perform CRUD operations on the following data.
- Output: Display the company page with retrieved data.

7. Servers:

- Input: Rack ID and Company ID
- Processes:
  - Retrieves data from the database(Racks, Company, Server)
  - Renders the company page with matched rack id with rack number, matched company id with server make and retrieved data and allows us to perform CRUD operations on the following data.
- Output: Display the server page with retrieved data.

8. Search for Server information:

- Inputs: None
- Processes:
  - Retrieves data from the database (racks, companies, servers).
  - Renders the search page with the retrieved data.
- Outputs: Displays the search page with the retrieved data.

## VIII. USER INTERFACE

The user interface of the Data Centre Servers Management System, designed using Figma, exhibits a clean and intuitive layout that enables efficient navigation and task execution. Users can easily manage servers, racks, and companies.

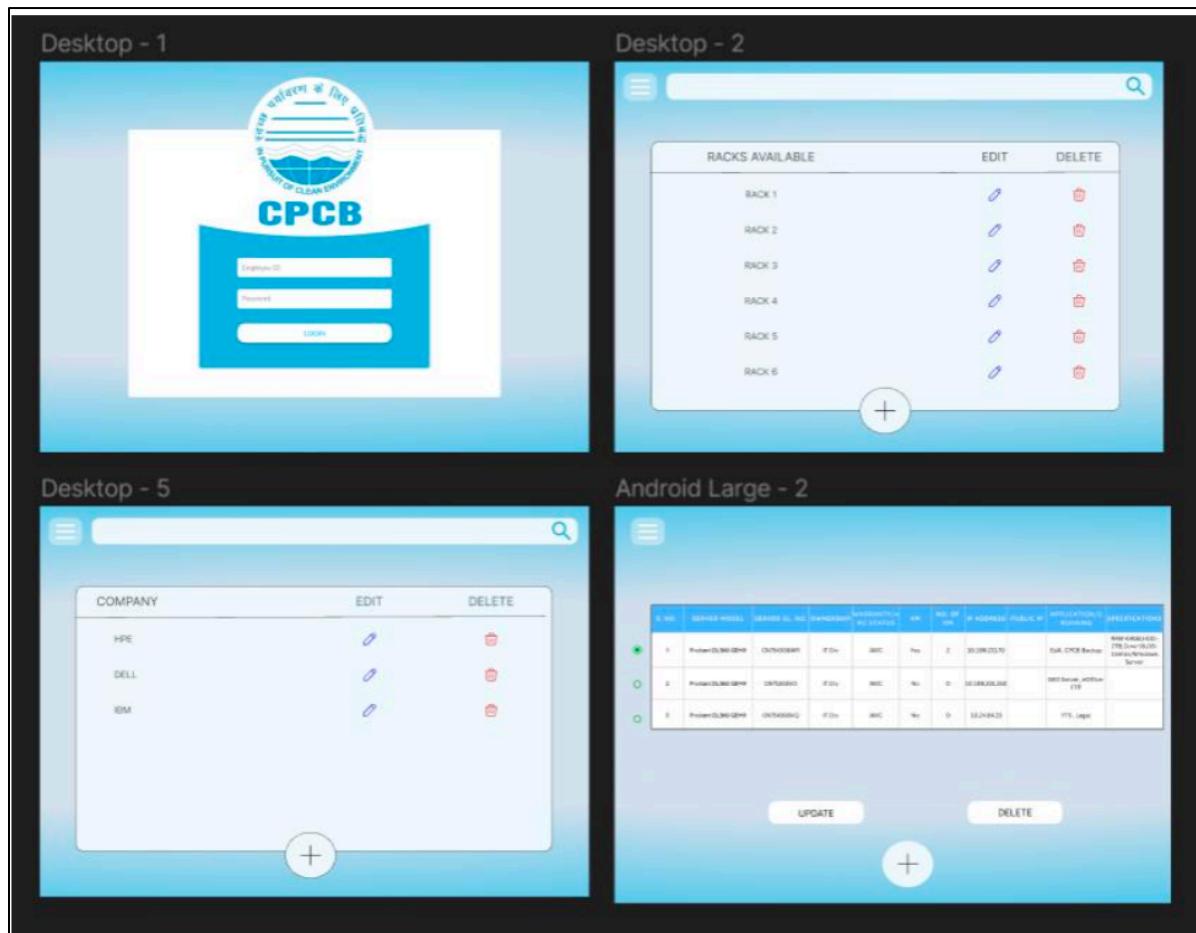


Figure 5

## IX. THE MODELS

1. **Rack Table:** This table is used to store information about different racks.

Fields:

- a. **Id:** This is a primary key to identify racks uniquely
- b. **Rack:** This is a CharField with a maximum length of 100 characters. It represents the name of the rack.

2. **Company Table:** This table is used to store information about different companies and their associations with racks.

The racks field establishes a many-to-many relationship between companies and racks, allowing a company to be associated with multiple racks, and vice versa.

Fields:

- a. **Id:** This is a primary key to identify companies uniquely
- b. **company:** This is a CharField with a maximum length of 100 characters. It represents the name of the company
- c. **racks:** This is a ManyToManyField related to the Rack table with the related\_name set to 'companies'. It represents the racks associated with the company.

3. **OwnershipChoice Table:** This table is used to store ownership choices that can be selected in the ownership field of the Server table.

Fields:

- a. **Id:** This is a primary key to identify ownership uniquely.
- b. **display\_name:** This is a CharField with a maximum length of 100 characters. It represents the display name for ownership choices.

4. Server Table:

Fields:

- a. **rack\_no:** This is a ForeignKey related to the Rack table with the on\_delete set to CASCADE. It represents the rack to which the server belongs.
- b. **server\_make:** This is a ForeignKey related to the Company table with the on\_delete set to CASCADE. It represents the company that manufactures the server.
- c. **server\_model:** This is a CharField with a maximum length of 100 characters and is marked as unique. It represents the model name of the server.
- d. **server\_serial\_no:** This is a CharField with a maximum length of 100 characters and is marked as unique. It represents the serial number of the server.
- e. **ownership:** This is a ForeignKey related to the OwnershipChoice table with the on\_delete set to CASCADE. It represents the ownership status of the server.
- f. **warranty\_amc\_status:** This is a CharField with a maximum length of 100 characters. It represents the warranty or AMC status of the server.
- g. **vms:** This is a BooleanField with a default value of False. It indicates whether the server is used for virtual machines (True) or not (False).

- h. **num\_of\_vms**: This is an IntegerField that can be blank and nullable. It represents the number of virtual machines if vms is True.
- i. **ip\_address**: This is a GenericIPAddressField marked as unique. It represents the IP address of the server.
- j. **public\_ip**: This is a GenericIPAddressField that can be blank and nullable, marked as unique. It represents the public IP address of the server if applicable.
- k. **applications\_installed**: This is a TextField. It represents the list of applications installed on the server.
- l. **portals\_running**: This is a TextField that can be blank and nullable. It represents the list of portals running on the server if applicable.
- m. **specification**: This is a TextField that can be blank and nullable. It represents the specifications or additional details of the server.
- n. **vm\_information**: This is a File Field which allows users to upload a file. It is meant to store a file containing information related to virtual machines on the server.

#### **Features:**

- The server\_model and server\_serial\_no fields are marked as unique, ensuring that no two servers have the same model or serial number.
- The rack\_no, server\_make, and ownership fields are ForeignKeys, establishing relationships with the Rack, Company, and OwnershipChoice tables, respectively.
- The vms field is a BooleanField that allows indicating whether the server is configured for virtual machines or not.
- The num\_of\_vms and vm\_information field is only relevant if vms is True, and it can be blank and nullable.
- The ip\_address field is marked as unique, ensuring that each server has a unique IP address.
- The public\_ip field is optional and can be used to store the public IP address of the server if applicable.
- The applications\_installed, portals\_running, and specification fields are TextFields, allowing for flexible storage of application and server details.

The models have been designed to represent the relationships and attributes of the entities involved in the system, facilitating efficient data storage and retrieval. The ForeignKey and ManyToManyField relationships enable linking data between the tables. Additionally, unique constraints on relevant fields prevent duplicate or conflicting information.

## X. FEATURES AND FUNCTIONALITIES

### 1. Login Page:

- **User Authentication:** Allows users to log in by entering their Employee ID and Password. provided to them via organization. These details will be provided only to the authorized users who can see the data as well as add, delete, or edit the data when required.
- **Form Validation:** Validates the form to ensure required fields are filled and provide feedback on incorrect entries.
- **Error Messages:** Displays warning messages if login credentials are incorrect or unauthorized.
- **Admin Access:** Provides a link to the admin login page for administrators to access the system.
- **Responsive Design:** The page is designed to adapt and provide an optimal user experience on various devices, including desktop and mobile.



Fig. 8 Login page

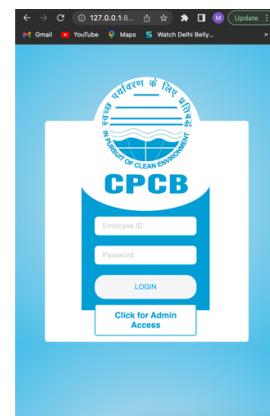


Fig. 7 Responsive page

### 2. Side Bar:

- **Sidebar Toggle:** The sidebar has an icon toggle functionality. When the icon (with ID "sidebar-icon") is clicked, the sidebar toggles between being visible and hidden. The icon changes between a "hamburger" icon (three horizontal lines) and a close (cross) icon when the sidebar is active.

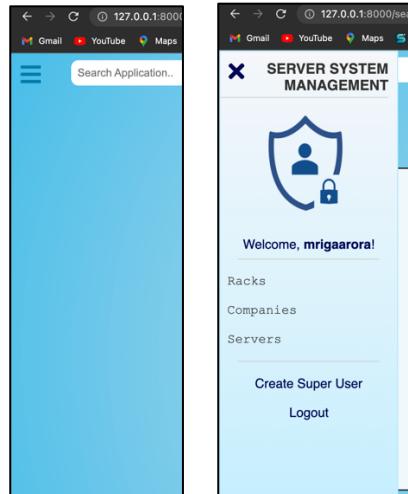


Fig. 8 Sidebar Toggle

- **User Profile and Welcome Message:** The sidebar shows a user profile picture (with the source "/media/1.png") and a welcome message. If the user is authenticated, it greets the user by their first name if available; otherwise, it uses the username.
- **Racks Submenu:** The sidebar contains a submenu for managing racks. When the "Racks" submenu link is clicked, it displays a list of options:
  1. View All Racks: Redirects to the "/search" URL.
  2. Add Rack: Opens a popup form for adding a new rack.
  3. Edit Rack: Opens a popup form for editing an existing rack.
  4. Delete Rack: Opens a popup form for deleting an existing rack.



Fig. 9 Racks Submenu

- **Companies Submenu:** The "Companies" submenu contains options for managing companies which have a similar UI to the Rack Submenu in Figure 9.
  1. View All Companies: Redirects to the "/com" URL.
  2. Add Company: Opens a popup form for adding a new company.
  3. Edit Company: Opens a popup form for editing an existing company.
  4. Delete Company: Opens a popup form for deleting an existing company.

- **Servers Submenu:** The "Servers" submenu includes options for managing server which have a similar UI to the Rack Submenu in Figure 9:
  1. View All Servers: Redirects to the "/apps" URL.
  2. Add Server: Opens a popup form for adding a new server.
  3. Edit Server: Opens a popup form for editing an existing server.
  4. Delete Server: Opens a popup form for deleting an existing server.
- **Create Super User:** Clicking the "Create Super User" link triggers a prompt to enter a password. If the entered password is valid, it redirects to the "/create\_superuser/" URL.
- **Logout:** Clicking the "Logout" link triggers a confirmation prompt. If the user confirms, it redirects to the "/search" URL with a "logout" query parameter set to "True".

**Popup Forms for Adding, Editing, and Deleting:** There are several popup forms (HTML div elements) for adding, editing, and deleting racks, companies, and servers. The popup forms contain appropriate form fields and buttons to perform the specified actions.

The views.py file contains Django view functions for handling different operations related to racks, companies, and servers. These functions handle adding, editing, and deleting records in the respective database models based on the user's input from the frontend.

The Popups:

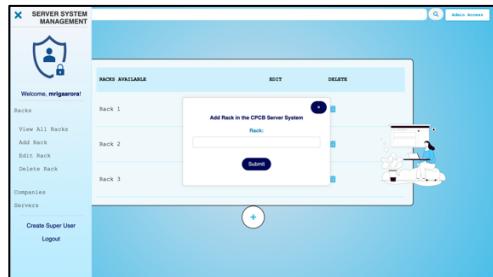


Fig. 10 Add Rack Popup

Fig. 11 Edit Rack Popup

Fig. 12 Delete Rack Popup

Fig. 13 Add Company Popup

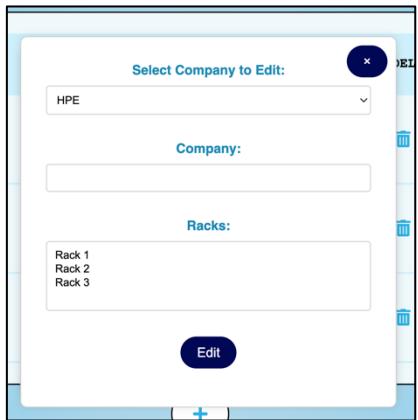


Fig. 14 Edit Company Popup

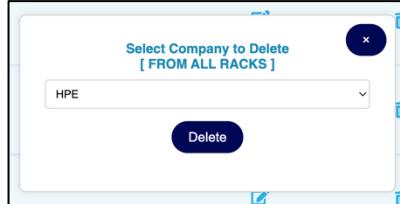


Fig. 15 Delete Company Popup

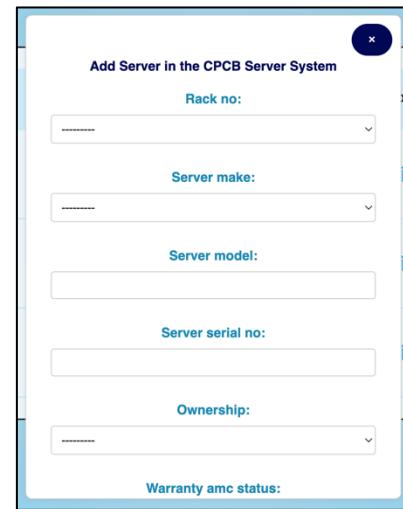


Fig. 16 Add Server Popup

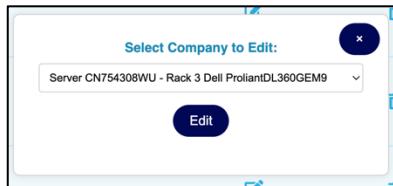


Fig. 17 Edit Server Popup

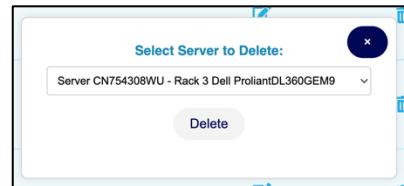


Fig. 18 Delete Server Popup

### 3. Create Superuser Page:

- **Authorization Control:** Only staff members who have the password to access the create superuser page are allowed to access the "Create Superuser" functionality. Unauthorized users will be denied access.
- **Superuser Creation Form:** The "Create Superuser" page displays a form with the following fields:
  1. **Username:** An input field where the staff member can enter the desired username for the new superuser.
  2. **Password:** An input field where the staff member can enter the password for the new superuser.
  3. **Confirm Password:** An input field to re-enter the password for confirmation.
- **Form Validation:** The form enforces validation rules for the username and password fields to ensure they meet specific criteria, such as character limits and allowed characters. Any validation errors are displayed to guide the user in providing valid input.

- **Superuser Creation:** When the staff member submits the form, the view function processes the data and creates a new superuser based on the provided information. The `is_staff` and `is_superuser` attributes of the user are set to True to grant superuser privileges.
- **Success Message:** After successful creation, a confirmation message is displayed to the staff member at the login page, indicating that the superuser has been created successfully.
- **Password Security:** The password field is designed to protect the security of the superuser account. It ensures that the password is not too similar to other personal information and is not easily guessable.

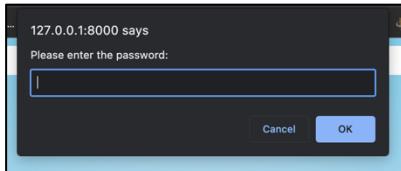


Fig. 19 Password Authentication

Fig. 20 Create Super User Form



Fig. 21 Success Message

4. **Admin Page:** The Admin Page allows staff members to perform various administrative tasks, such as adding, editing, and deleting records for the registered models (Rack, Company, Server, and Ownership Choice). The use of import and export functionality, custom form rendering, and theme integration enhances the efficiency and usability of the admin interface. Authorized members can easily manage data, apply filters, and view associated data, making it a powerful tool for managing the website's backend:

- **Model Registration:** The admin page registers the Django models (Rack, Company, Server, and Ownership Choice) to be managed via the Django Admin interface.
- **Import and Export Data:** The `ImportExportModelAdmin` from the `import_export` library allows for easy import and export of data in various formats (e.g., CSV, JSON) for the Rack, Company, and Server models.
- **Customizing Display:** The `list_display` attribute is used to customize the columns displayed in the list view for each model. For example:
  - a) The RacksAdmin model displays the id and Rack columns in the list view.

- b) The CompanyAdmin model displays the id, company, and display\_racks columns, where display\_racks is a custom function displaying the associated racks for each company.
  - c) The ServerAdmin model displays various server-related fields such as rack\_no, server\_make, server\_model, etc.
- **Customizing List Filter:** The list\_filter attribute is used to provide filtering options in the list view for the Server model. Staff members can filter servers based on server make and rack number.
  - **Displaying Associated Racks:** In the CompanyAdmin model, a custom function display\_racks is defined to show a comma-separated list of associated racks for each company. This enhances the readability of the displayed data.
  - **Customizing Ownership Choices:** The OwnershipChoiceAdmin model displays the id and display\_name columns, allowing staff members to manage ownership choices effectively.
  - **Themes from Django-admin-interface:** The admin page's appearance is enhanced with themes provided by the Django-admin-interface package. The themes improve the visual aesthetics of the admin interface and provide a user-friendly experience for managing the models.

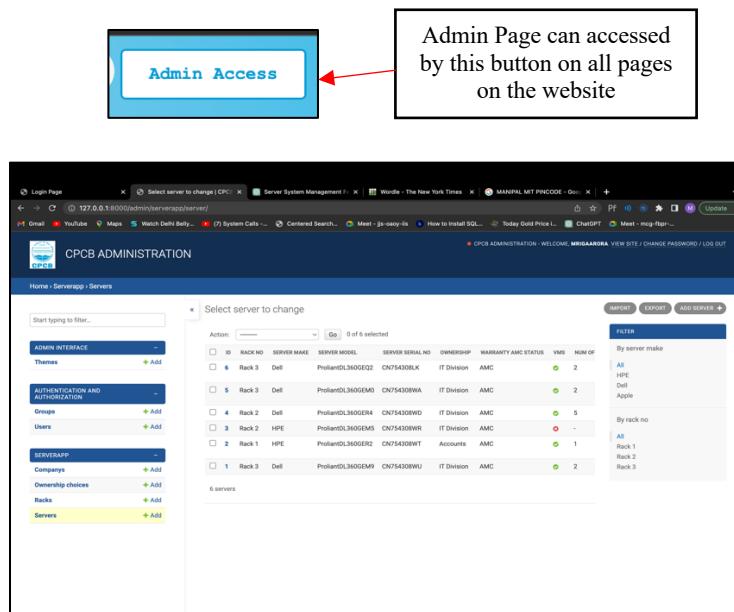


Fig. 22 Admin Page

5. **Rack Page:** After logging in, the user (staff member) will be directed to the rack page where multiple racks present in the system can be accessed and viewed. Within these racks, different companies will be present.

- **Table Display:** The racks page displays a table that lists all available racks. Each row in the table represents a rack and shows its name. (Figure 23)

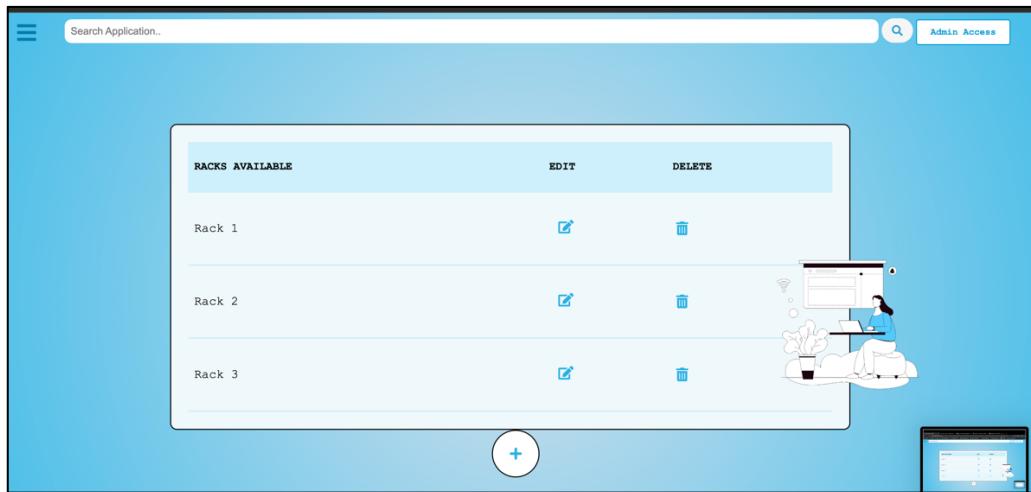


Fig.23 Rack Page

- **Toggle Form Display:** The "Add" button acts as a toggle to show or hide the form for adding a new rack. Clicking the "Add" button displays the form, and clicking it again hides the form.
- **Add Rack:** There is an "Add" button represented by a plus icon. Clicking on this button displays a form below the table, allowing staff members to add a new rack by providing its details.(Figure 24).

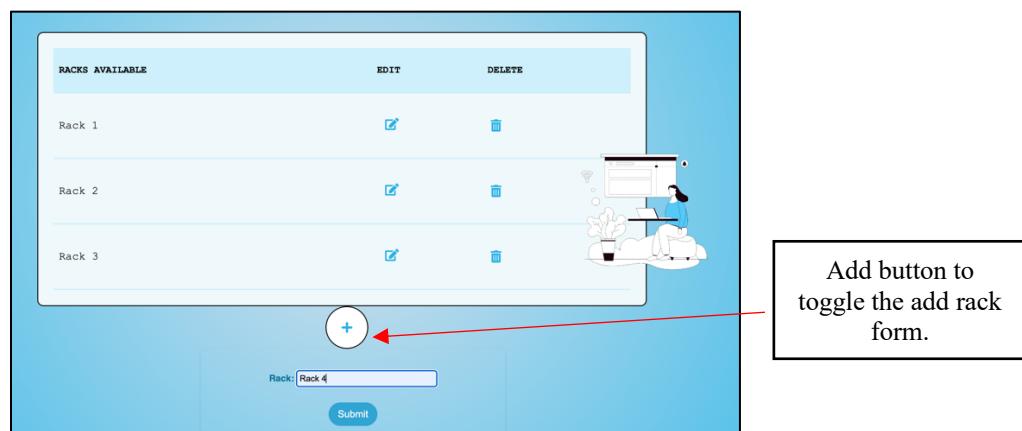
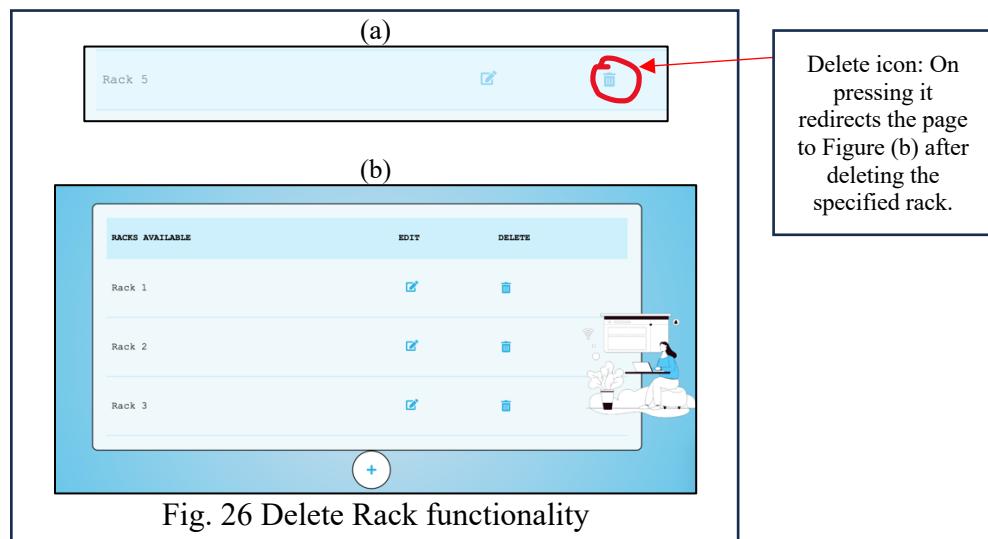
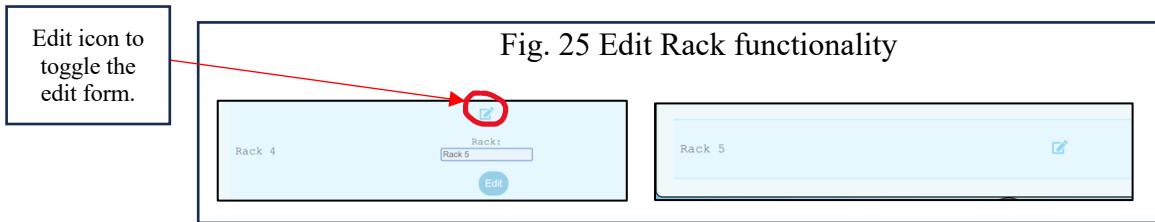


Fig. 24 Add button functionality

- **Form Validation:** The form for adding or editing racks is validated before saving the data. If the entered data is valid, the rack details are saved, and the table is updated with the new data.

- **Edit and Delete Actions:** For each rack in the table, there are "Edit" and "Delete" icons. Clicking on the "Edit" icon allows staff members to modify the rack's details, and clicking on the "Delete" icon removes the rack from the database. (Figure 25 and Figure 26)



- **Styling and Animations:** The rack page includes CSS styling and animations for visual appeal and improved user experience. For example, the table, forms, buttons, and icons are styled using CSS.

## 6. Company Page:

- **Filter Companies by Associated Racks:** On the Company page, the website filters and lets the staff view only those companies that are associated with a specific rack. This allows staff members to focus on companies that have a connection to a particular rack.
- **Table Display:** The company page displays a table that lists all companies associated with a specific rack (if provided). Each row in the table represents a company and shows its name.

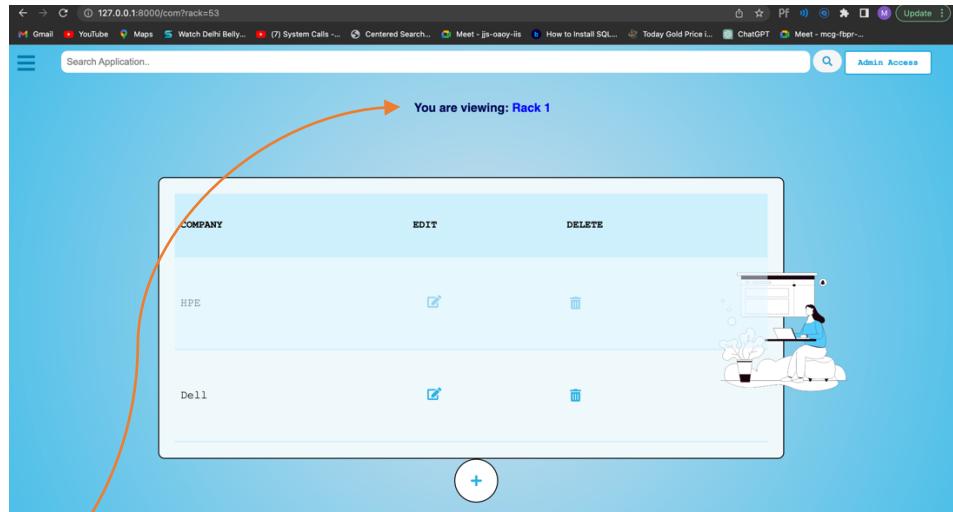


Fig. 27 Company Page

- **Current Rack Information:** If a specific rack is selected, the page displays a header indicating that the user is viewing the companies associated with that rack.
- **Add Company:** There is an "Add" button represented by a plus icon. Clicking on this button displays a form below the table, allowing staff members to add a new company by providing its details.(Figure 28)
- **Toggle Form Display:** The "Add" button also acts as a toggle to show or hide the form for adding a new company. Clicking the "Add" button displays the form, and clicking it again hides the form.

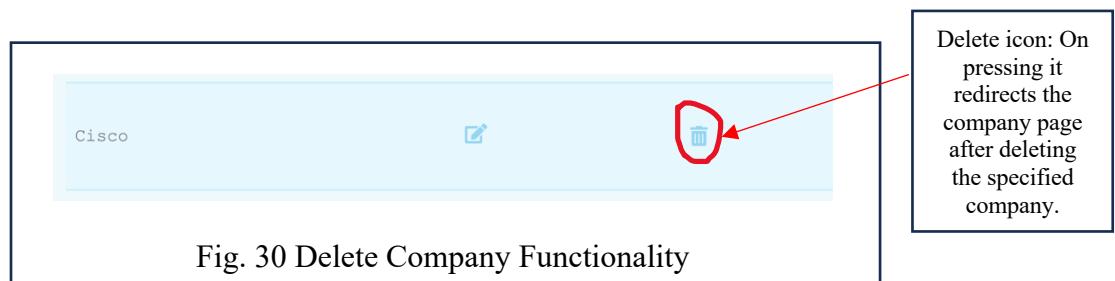
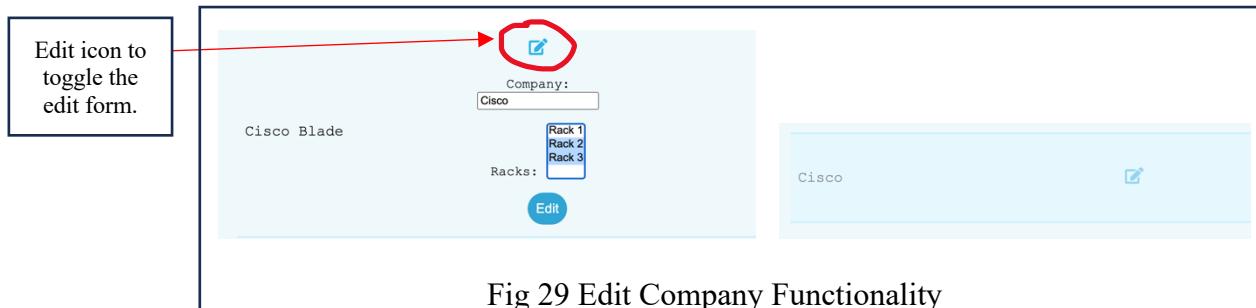
(a) shows the 'Add' form expanded, displaying fields for 'Company' (Cisco Blade), 'Racks' (Rack 1, Rack 2, Rack 3), and a 'Submit' button.

(b) shows the 'Add' form collapsed, appearing as a plus sign icon at the bottom of the table.

A callout box with a red arrow points to the plus sign icon in (b), containing the text: "Add button to toggle the add company form."

Fig. 28 Add Company functionality

- **Edit and Delete Actions:** For each company in the table, there are "Edit" and "Delete" icons. Clicking on the "Edit" icon allows staff members to modify the company's details, and clicking on the "Delete" icon removes the company from the database.



- **Submit Form:** The form for adding a new company or editing an existing company includes a "Submit" button. Clicking this button saves the entered data and updates the table accordingly.
- **Form Validation:** The form for adding or editing companies is validated before saving the data. If the entered data is valid, the company details are saved, and the table is updated with the new data.

7. **Server Page:** The Apps (Server) page provides a user-friendly interface for managing server information and offers flexibility in filtering and viewing servers based on specific criteria. Staff members can easily add, edit, and delete servers using the intuitive interface provided by the page.

- **Filter Servers:** The Apps (Server) page allows staff members to view a list of servers. The servers can be filtered based on the following criteria:
  - a) By Rack: If a rack\_id is provided in the URL parameters, only servers belonging to that specific rack will be displayed.
  - b) By Company: If a company\_name is provided in the URL parameters, only servers of the selected company will be shown.

This helps staff members to narrow down the list of servers based on company and rack selections.

- Display Server Information:** The page displays server information in a tabular format. Each row represents a server and shows details such as Rack No., Server Make, Server Model, Server Serial No., Ownership, Warranty/AMC Status, VMs (Yes/No), No. of VMs, IP Address, Public IP, Applications Installed, Portals Running, and Server Specification.

S.	Rack No.	Server Model	Server Serial No.	OwnerShip	Warranty AMC Status	VMs	No. of VMs	IP Address	Public IP	Applications Installed	Portals Running	Specification	Edit	Delete
1	1	Dell	Precision T3500	IT Department	Yes	None	None	192.168.231.76	192.168.231.42	Apache, MySQL	None	RAM: 16GB		
2	2	Dell	Precision T3500	IT Department	Yes	None	None	192.168.231.76	192.168.231.42	Apache, MySQL	None	RAM: 16GB		
3	3	Dell	Precision T3500	IT Department	Yes	None	None	192.168.231.76	192.168.231.42	Apache, MySQL	None	RAM: 16GB		

Fig. 31 Server Page

Add button to toggle the add server form.

- Add New Server:** Users can add a new server by clicking on the "+" button. This action reveals a form where staff members can input details for a new server and submit the form to add it to the database.
- Toggle Form Display:** The "Add" button also acts as a toggle to show or hide the form for adding a new company. Clicking the "Add" button displays the form, and clicking it again hides the form.

Fig. 32 Add Server Functionality

- Edit Server Details:** Staff members can edit server information by clicking on the edit icon next to the server. This action redirects to an edit form on the edit page for the selected server, allowing users to modify server details. After making changes, users can submit the form to update the server information. (details in section 8)
- Delete Servers:** Staff members can delete servers by clicking on the delete icon next to the server. This action removes the server entry from the database.

BLIC IP	APPLICATIONS INSTALLED	PORTALS RUNNING	SPECIFICATION	EDIT	DELETE
192.168.231.76	CPCB Backup, MySQL	Battery Wastage Portal	RAM: 16GB		

Delete icon: On pressing it redirects the server page after deleting the specified server.

Fig. 33 Delete Server Functionality

- **Validation Error Handling:** When adding or editing a server, the page handles validation errors. If the form submission contains errors, a pop-up box appears, displaying the error message. Users can click the pop-up box to dismiss it.

For example:

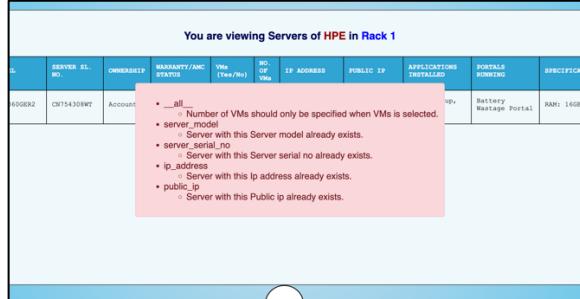


Fig. 34 Errors

## 8. Edit Server Page:

The screenshot shows a table with columns: PUBLIC IP, APPLICATIONS INSTALLED, PORTALS RUNNING, SPECIFICATION, EDIT, and DELETE. The PUBLIC IP column shows 123.4.231.21. The APPLICATIONS INSTALLED column shows CPCB Backup, MySQL. The PORTALS RUNNING column shows Battery Wastage Portal. The SPECIFICATION column shows RAM: 16GB. The EDIT and DELETE buttons are in the last two columns respectively. A red arrow points to the EDIT button.

Edit icon on the server page: On pressing it redirects the edit page.

Fig. 35 Edit icon to redirect to Edit Server Page

- The Edit Server page allows staff members to modify the details of an existing server. The page displays a form **pre-populated** with the **current server information**, allowing users to make changes. After making the necessary modifications, staff members can click the "Save" button to save the edited server details. It then redirects back to the display all servers page.
- The page also handles validation errors that may occur during the form submission. If there are any errors, a pop-up box appears, displaying the error message to the user. The user can click the pop-up box to dismiss it

The screenshot shows the "Edit Server Page" with fields for Rack no, Server make, Server model, Server serial no, Ownership, Warranty amc status, Vms, Num of vms, Ip address, and Public ip. To the right, a pop-up box displays an error message: "Number of VMs should only be specified when VMs is selected." Below the pop-up are sections for Applications installed (MySQL, MySQL), Portals running (RAM: 16GB), and Specification, with a Save button at the bottom.

Fig. 36 Edit Server Page

9. **Search Servers:** The Search Bar feature enhances the user experience by providing an easy and efficient way to find specific servers based on their applications and other relevant information.

- **Search Query Handling:** When the user enters a search query and submits the form, the page processes the search query using the GET method.
- **Server Filtering:** The search bar filters the servers based on the search query. It looks for the query in multiple fields, including:
  1. Applications Installed
  2. Portals Running
  3. Specification
  4. IP Address
  5. Server Model
  6. Server Serial Number
  7. Warranty/AMC Status
  8. Ownership Display Name
  9. Server Make (Company)
  10. Rack Number
  11. Number of VMs

The search query is case-insensitive, allowing users to enter partial or full strings to find relevant servers.

- **Dynamic Search Results:** After submitting the search query, the page displays a list of servers that match the search criteria. The search results dynamically update the server list without reloading the entire page.
- **Clearing Search:** If users want to remove the search query and view all servers again they can leave the search bar empty and submit the form.

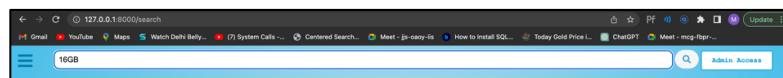


Fig. 37 Search bar

S. NO.	RACK NO.	SERVER MAKE	SERVER MODEL	SERVER ID.	CHASSIS/ROLE	WARRANTY/AMC STATUS	VMs (Type/No.)	NO. OF VMs	IP ADDRESS	PUBLIC IP	APPLICATIONS INSTALLED	PORTALS RUNNING	SPECIFICATION	EDIT	DELETE
1	Rack 3	Dell	ProliantDL360B8M	CNT1430BMO	IT Division	AMC	True	2	10.199.231.79	12.34.231.42	Kerpp, MySQL	BMS Alert	RAM: 16GB	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Rack 1	HPE	ProliantDL360B8M	CNT1430BMT	Accounts	AMC	True	1	10.199.231.21	12.34.231.21	CNCB Backup, MySQL	Battery Management Portal	RAM: 16GB	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Rack 2	Dell	ProliantDL360B8M	CNT1430BMD	IT DIVISION	AMC	True	5	10.199.231.22	12.34.231.32	Kerpp	EMP Portal	RAM: 16GB	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Rack 3	Dell	ProliantDL360B8QZ	CNT1430BLK	IT Division	AMC	True	2	10.199.231.43	12.34.20.90	Geo	Battery Management Portal	RAM: 16GB	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Fig. 38 Filtered Search result

## XI. CONCLUSION

In conclusion, in this Django project, we have implemented various views and functionalities to manage servers, racks, companies, and users. The application allows users to perform CRUD (Create, Read, Update, Delete) operations on servers, racks, and companies. Additionally, it supports user authentication and authorization, allowing only staff members to access certain views.

The application's main functionality includes:

- User Authentication: Users can log in with valid credentials and are redirected to the search page upon successful login. Unauthorized users are shown a warning message.
- Superuser Creation: A staff member can create a new superuser by using the "create\_superuser" view, which grants administrative access to the system.
- Search Functionality: Users can search for servers based on various criteria, including server specifications, IP addresses, companies, rack numbers, and more.
- Server Management: Users can view a list of servers, add new servers, edit existing servers, and delete servers from the database.
- Rack Management: Staff members can manage racks by adding, editing, and deleting rack information.
- Company Management: Staff members can manage company information, including adding, editing, and deleting company details.
- Side Bar functionality: Staff members can easily access and perform CRUD operations on racks, companies and servers through the side bar as well.

The website is well-structured and functional, providing users with the ability to manage server-related information effectively.

## XII. FUTURE SCOPE

The future scope of the data centre server management system project includes potential enhancements and additions to further improve its functionality and address evolving needs. Some possibilities for future development and expansion are:

1. Enhanced Monitoring and Alerting:

- Implement real-time monitoring capabilities to track server performance, and resource utilization.
- Integrate with monitoring tools or APIs to capture server metrics and generate alerts for critical events or thresholds.
- Provide notifications and automated actions for proactive server management.

2. Resource Allocation and Optimization:

- Develop features for optimizing server resource allocation and capacity planning.
- Implement algorithms or intelligent systems to recommend server consolidation or virtualization strategies based on workload patterns and resource utilization data.
- Provide insights and analytics on server performance and efficiency to support decision-making for resource optimization.

3. Integration with Cloud and Virtualization Technologies:

- Extend the system's capabilities to manage servers in virtualized environments, such as VMware or Hyper-V.
- Integrate with cloud platforms, such as Amazon Web Services (AWS) or Microsoft Azure, to manage cloud-based servers and resources.
- Provide features for seamless migration of servers between physical and virtual environments.

4. Compliance and Security Enhancements:

- Incorporate additional security measures, such as two-factor authentication and encryption, to further protect sensitive server data.
- Enhance access control mechanisms to ensure granular permissions and auditing capabilities.

5. Mobile Application:

- Develop a mobile application to provide on-the-go access to server management functionalities.
- Enable remote server monitoring, alerts, and basic management tasks through the mobile application.

### XIII. REFERENCES

- [1] <https://docs.staruml.io/>
- [2] <https://cpcb.nic.in/>
- [3] <https://docs.djangoproject.com/en/4.2/>
- [4] <https://stackoverflow.com/>
- [5] <https://www.w3schools.com/django/>
- [6] <https://django.readthedocs.io/en/stable/contents.html>
- [7] <https://chat.openai.com/>
- [8] <https://www.figma.com/>