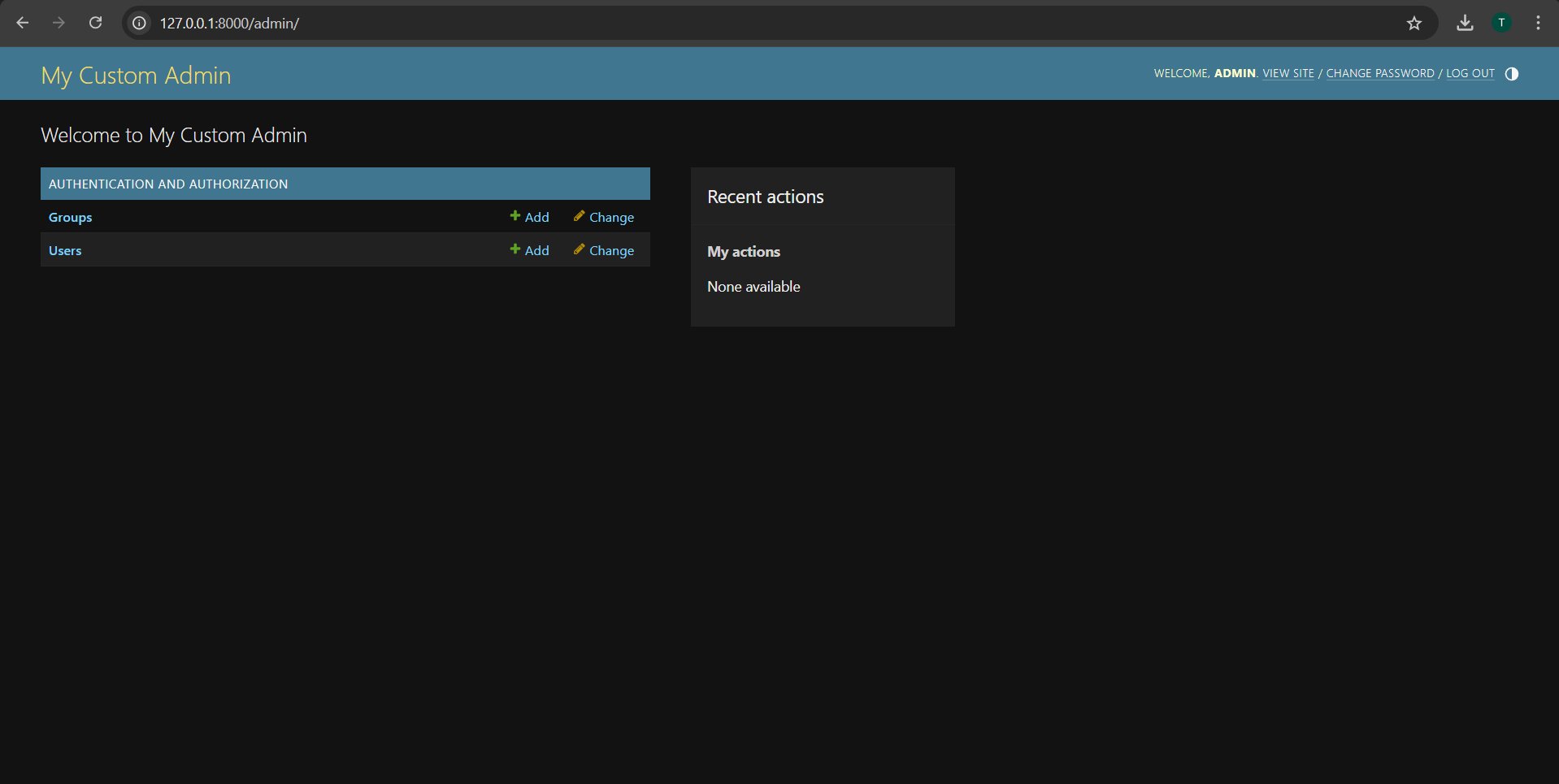
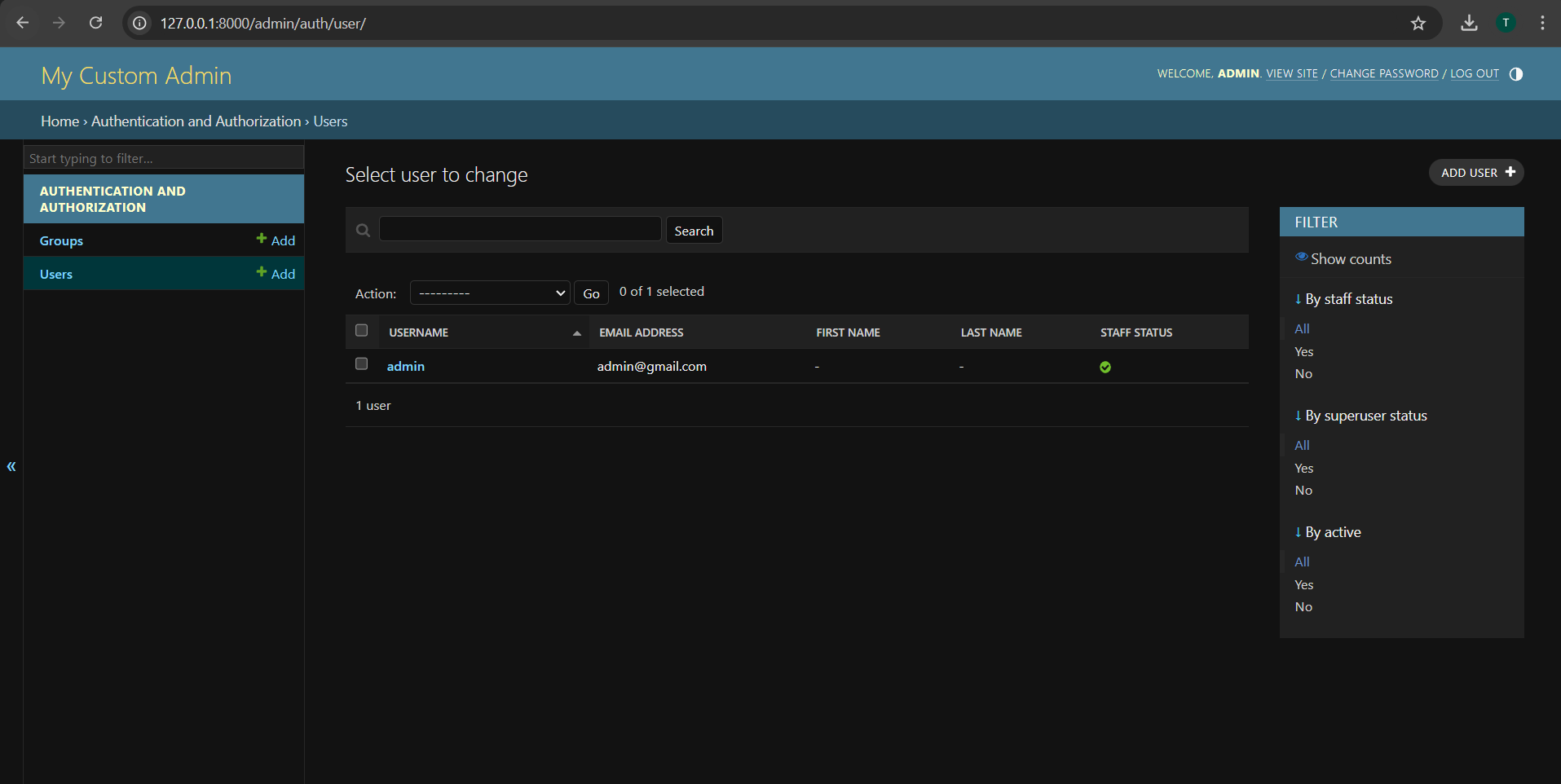
**Build a Model :**

1. **Task Description**

This project focuses on customizing the Django Superadmin interface by changing its color theme using internal CSS in admin.py. Additionally, it involves updating the admin panel’s title and using different icons to enhance the user experience and align the interface with branding guidelines.

1. **Task Output Screenshot**

****



## Project Setup

### Starting the Project-

Create a Django Project:

* Set up a new Django project or navigate to an existing project.
* Ensure that the Django admin is properly configured and accessible.

### Creating the Model-

Define Models:

* Create necessary models (if not already defined) for the application.
* Register these models in the admin interface.

## User Interface Development

### Data Insertion -

### Customizing the appearance of forms in the admin panel is not directly part of the CSS changes, but ensuring that form styles match the new theme is essential.

## Functionality Improvements

**Change Admin Titles:**

* Modify the site\_header, site\_title, and index\_title properties to personalize the admin interface.

### Admin Interface-

### The admin panel is enhanced to display device names instead of generic object names, facilitating easier management. The Django admin interface allows authorized users to add, modify, and delete devices efficiently.

## Conclusion

Customizing the Django Superadmin interface by altering the color theme, updating titles, and using different icons significantly enhances usability and aesthetics. Implementing these changes in admin.py through internal CSS not only improves the visual appeal but also reinforces brand identity, making the admin experience more engaging for users.