

# INTRODUCTION TO CLERK

Clerk is an authentication and user management solution for Next.js applications. It provides secure authentication, user session handling, and role-based access control. In this project, Clerk is used to enforce authentication via middleware, ensuring that only authorized users can access the admin dashboard.

## Admin Dashboard

### Overview

The Admin Dashboard in Next.js fetches and displays orders from Sanity. Admins can view user orders and delete them when necessary.

## Features

- Displays customer orders with user details.
- Lists ordered products with title, price, and quantity.
- Shows the total amount for each order.
- Allows order deletion.

## Key Functions

### FETCHING ORDERS

Orders are retrieved using `useEffect`:

```
useEffect(() => {  
  const fetchOrders = async () => {  
    try {  
      const data = await client.fetch(`  
        *[_type == "order"]{ _id, user { name, email }, products, totalAmount, date }  
      `)
```

```

    `);

    setOrders(data);

  } catch (error) {

    console.error('Error fetching orders:', error);

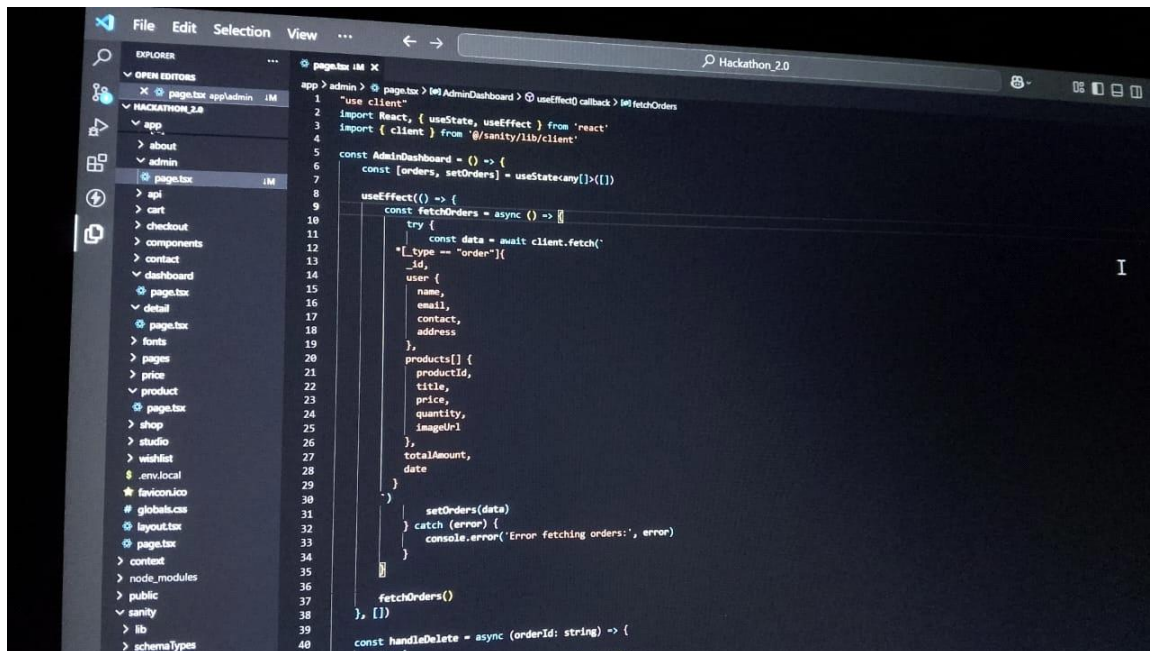
  }

};

fetchOrders();

}, []);

```



## Deleting Orders

Orders can be deleted with:

```

const handleDelete = async (orderId: string) => {

  try {

```

```

    await client.delete(orderId);

    setOrders(orders.filter(order => order._id !== orderId));

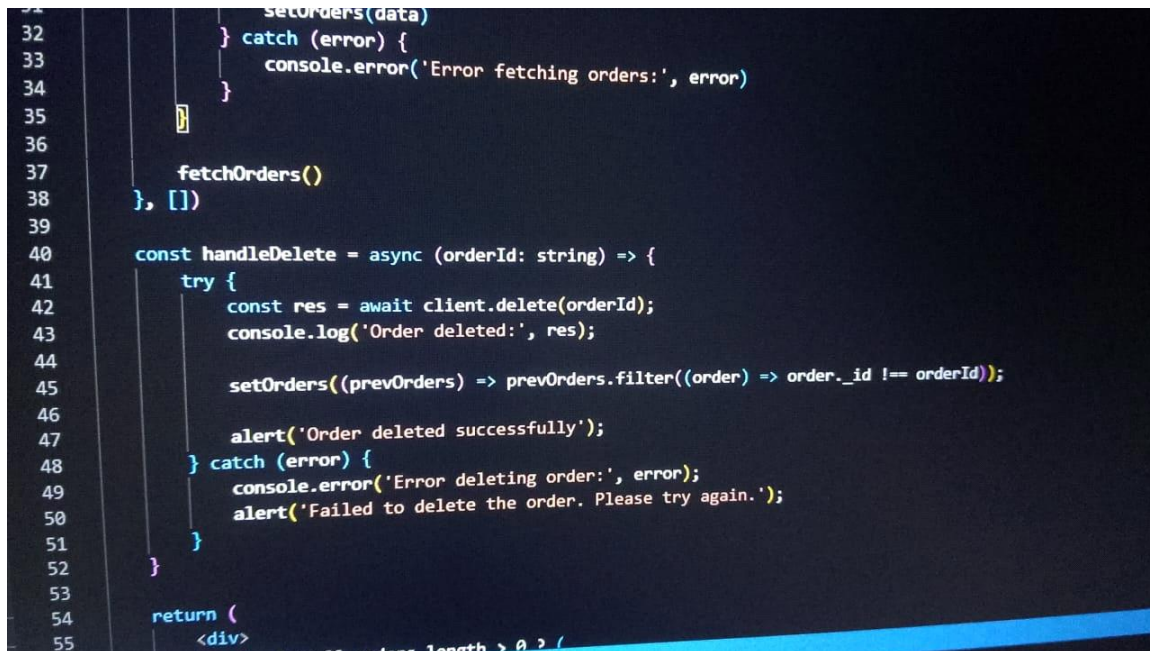
  } catch (error) {

    console.error('Error deleting order:', error);

  }

};

```



```

32     setOrders(data)
33   } catch (error) {
34     console.error('Error fetching orders:', error)
35   }
36
37   fetchOrders()
38 }, [])
39
40 const handleDelete = async (orderId: string) => {
41   try {
42     const res = await client.delete(orderId);
43     console.log('Order deleted:', res);
44
45     setOrders((prevOrders) => prevOrders.filter((order) => order._id !== orderId));
46
47     alert('Order deleted successfully');
48   } catch (error) {
49     console.error('Error deleting order:', error);
50     alert('Failed to delete the order. Please try again.');

```

## Middleware Setup

**Clerk authentication is enforced via middleware:**

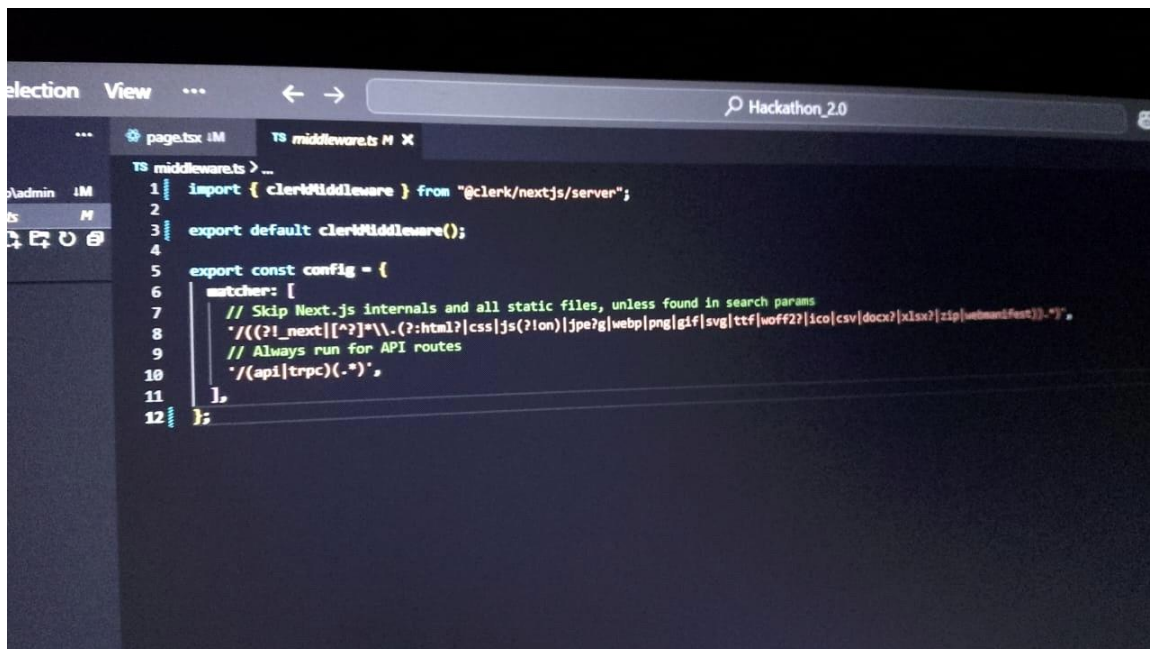
```
import { clerkMiddleware } from "@clerk/nextjs/server";
```

```
export default clerkMiddleware();
```

```
export const config = {
```

```
  matcher: ['/(?!_next|.*(?:css|js|png|jpg|svg)).*', '/(api|trpc)(.*)'],
```

```
};
```



## Future Enhancements

- Confirmation prompt before deletion.
- Pagination for better order management
- Sorting and filtering options.

**This document provides a brief overview of the Clerk Admin ,Dashboard ,and its authentication middleware.**