

Chakra UI

Chakra UI is a popular React component library that provides accessible, modular, and customizable building blocks to help you build frontend applications faster. Its primary advantage is excellent developer experience (DX) and style props that allow you to add CSS directly to components.

Installation and Setup

Create a React Project: If you don't have one, you can use Vite or Create React App.

bash

```
npm create vite@latest my-app -- --template react-ts
cd my-app
npm install
```

Install Chakra UI: Install the core package and its required dependencies.

bash

```
npm i @chakra-ui/react @emotion/react @emotion/styled framer-motion
```

Set up `ChakraProvider`: Wrap your application's root component (usually `index.tsx` or `main.tsx`) with the `ChakraProvider` to enable the theme and styling system. `src/main.tsx` (or `index.tsx`):

tsx

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { ChakraProvider } from '@chakra-ui/react';
import { App } from './App';

ReactDOM.createRoot(document.getElementById('root')).render(
```

```
<React.StrictMode>
  <ChakraProvider>
    <App />
  </ChakraProvider>
</React.StrictMode>,
);
```

Example: A Simple Login Form

You can use Chakra UI components as building blocks, applying styles using props like `mt` (margin top), `maxW` (max width), `p` (padding), and `colorScheme`.

`src/App.tsx`:

tsx

```
import {
  Box,
  Button,
  Center,
  FormControl,
  FormLabel,
  Input,
  Stack,
  Heading,
  Text,
  useToast,
} from '@chakra-ui/react';
import React, { useState } from 'react';

function App() {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const toast = useToast();
```

```

const handleSubmit = (e: React.FormEvent) => {
  e.preventDefault();
  // In a real app, you would handle authentication here
  console.log('Login attempt with:', { email, password });
  toast({
    title: 'Login successful!',
    description: "You've successfully logged in with Chakra UI.",
    status: 'success',
    duration: 3000,
    isClosable: true,
  });
};

return (
  <Center h="100vh" bg="gray.50">
    <Box p={8} maxW="md" borderWidth={1} borderRadius="lg" boxShadow="lg" bg="white">
      <Stack spacing={4}>
        <Heading fontSize="2xl" textAlign="center">Sign in to your account</Heading>
        <Text fontSize="md" color="gray.600" textAlign="center">
          Enter your details below to log in.
        </Text>
        <form onSubmit={handleSubmit}>
          <Stack spacing={4}>
            <FormControl id="email" isRequired>
              <FormLabel>Email address</FormLabel>
              <Input
                type="email"
                value={email}
                onChange={(e) => setEmail(e.target.value)}
                placeholder="you@example.com"
              />
            </FormControl>
            <FormControl id="password" isRequired>
              <FormLabel>Password</FormLabel>
              <Input
                type="password"
                value={password}
                onChange={(e) => setPassword(e.target.value)}
                placeholder="*****"
              />
          </Stack>
        </form>
      </Stack>
    </Box>
  </Center>
);

```

```
        </FormControl>
        <Button type="submit" colorScheme="blue" size="lg" fontSize="md">
          Sign In
        </Button>
      </Stack>
    </form>
  </Stack>
</Box>
</Center>
);
}

export default App;
```

For more examples, component documentation, and styling options, refer to the official [Chakra UI documentation](#).

Revision #1

Created 11 December 2025 17:43:03 by AI Channel

Updated 11 December 2025 17:43:42 by AI Channel