

```
'use client';
```

```
import { User } from '@supabase/supabase-js';
```

```
import {
```

```
  createContext,
```

```
  useContext,
```

```
  useState,
```

```
  PropsWithChildren,
```

```
  useEffect,
```

```
  Dispatch,
```

```
  SetStateAction,
```

```
} from 'react';
```

```
interface AuthProviderProps extends PropsWithChildren {
```

```
  user: User | null;
```

```
}
```

```
interface AuthContextType extends AuthProviderProps {
```

```
  isAuthModalOpen: boolean;
```

```
  openModal: () => void;
```

```
  closeModal: () => void;
```

```
  setIsAuthModalOpen: Dispatch<SetStateAction<boolean>>;
```

```
}
```

```
const AuthContext = createContext<AuthContextType | undefined>(undefined);
```

```
export const AuthProvider = ({ children, user }: AuthProviderProps) => {
```

```
const [isAuthModalOpen, setIsAuthModalOpen] = useState(false);
```

```
const openModal = () => setIsAuthModalOpen(true);
```

```
const closeModal = () => setIsAuthModalOpen(false);
```

```
useEffect(() => {
```

```
  if (!user) {
```

```
    openModal();
```

```
  }
```

```
}, [user]);
```

```
return (
```

```
  <AuthContext.Provider
```

```
    value={{
```

```
      isAuthModalOpen,
```

```
      openModal,
```

```
      closeModal,
```

```
      setIsAuthModalOpen,
```

```
      user,
```

```
    }}  
  >
```

```
    {children}
```

```
  </AuthContext.Provider>
```

```
);
```

```
};
```

```
export const useAuthContext = () => {  
  
  const context = useContext(AuthContext);  
  
  if (!context) {  
  
    throw new Error('useModal must be used within a AuthProvider');  
  
  }  
  
  return context;  
  
};
```