

[사용자] 회원탈퇴 기능 구현하기

≡ 구현 어려움 정도	☆☆
☑ 구현 완료	☑
📅 기간	@December 14, 2024
☑ 정리 완료	☑

- 회원 탈퇴 버튼을 클릭하면 서버에서 사용자 정보를 삭제하고 세션을 종료한 후 리다이렉트
- 클라이언트 측에서 탈퇴 요청을 보내면 성공 여부에 따라 화면을 리다이렉트하거나 오류메세지를 보내도록 설계

1. deleteAccount

"use server";

- 로그인 상태 확인 - `verifySession`
 - `verifySession`를 호출해 현재 로그인한 사용자의 세션을 확인
 - 만약 세션이 없다면 로그인 상태가 아니므로 오류 메시지를 반환
- 사용자 존재 여부 확인 - `db.select()`
 - 세션에서 얻은 `session.id`를 사용해 사용자가 데이터베이스에 존재하는지 확인
 - 해당 사용자가 데이터베이스에 없다면 사용자 정보가 존재하지 않으므로 오류메세지를 반환
- 사용자 삭제 - `db.delete`
 - 데이터베이스에서 사용자의 정보를 삭제
 - 사용자는 `user.id`가 세션에 있는 `id`와 일치하는 경우에만 삭제
- 세션종료 - `deleteSession`
 - 사용자가 탈퇴한 후에 세션을 종료, 로그인된 상태 해제
- 리다이렉트
 - 탈퇴가 성공적으로 완료되면 `/login` 페이지로 리다이렉트

```
"use server";
```

```
import { db } from "@/db";
import { user } from "@/db/schema";
import { verifySession, deleteSession } from "@/actions/sessions";
import { redirect } from "next/navigation";
import { eq } from "drizzle-orm";
```

```
export const deleteAccount = async () => {
  try {
    // 1. 로그인 상태 확인 (세션 체크)
    const session = await verifySession();
    if (!session || !session.id) {
      return { errorMessage: "로그인 상태가 아닙니다. 다시 로그인해주세요." };
    }

    // 2. 해당 사용자가 존재하는지 확인
    const existingUser = await db.select().from(user).where(eq(user.id, session.id)).limit(1);
    if (!existingUser.length) {
      return { errorMessage: "존재하지 않는 사용자입니다." };
    }

    // 3. 사용자 탈퇴: 사용자 데이터 삭제
    await db.delete(user).where(eq(user.id, session.id));
  }
}
```

```

// 4. 세션 종료 (로그아웃)
await deleteSession();

} catch (error) {
  console.error(error);
}

// 5. 리다이렉트
redirect("/login");
}

```

2. DeleteAccountButton

"use client";

- isDeleting
 - 사용자가 탈퇴 요청을 보냈을 때, 탈퇴가 진행 중인 상태인지의 여부를 관리
 - 버튼 클릭 시, isDeleting을 true로 설정하고 탈퇴가 완료되면 false로 설정해 버튼 텍스트를 변경하고 로딩 상태를 처리
- handleDelete
 - 탈퇴 버튼 클릭 시 발생하는 함수
 - 서버 측 deleteAccount() 함수를 호출해 탈퇴 요청을 서버로 보냄
 - 서버의 응답에 따라 성공, 실패 처리를 함
- updateUser(null);
 - header 상태 변경(로그아웃 → 로그인)을 위해 유저 정보 null로 변경하여 상태 값 변경

```

"use client";

import React, { useState } from "react";
import { deleteAccount } from "@/actions/deleteAccount";
import { useUserStore } from "@/stores/useUserStore";

const DeleteAccountButton = () => {
  const [isDeleting, setIsDeleting] = useState(false);
  const updateUser = useUserStore((state) => state.updateUser);

  const handleDelete = async () => {
    setIsDeleting(true);

    try {
      updateUser(null);
      await deleteAccount();
    } catch (error) {
      window.location.href = "/login";
    }

    setIsDeleting(false);
  };

  return (
    <div>
      <button
        onClick={handleDelete}
        disabled={isDeleting}
      >
        {isDeleting ? "탈퇴 중..." : "회원 탈퇴"}
      </button>
    </div>
  );
};

```

```
        </div>
    );
};

export default DeleteAccountButton;
```