

```

import LoadingSpinner from '@shared/components/loading-spinner';

import { useAuthContext } from '@shared/components/ui/auth.provider';

import { trpc } from '@shared/utils/trpc/trpc';

import React from 'react';

const FREE_CREDITS = 20;

export default function CreditsUsage() {

  const { user } = useAuthContext();

  const userFreeCredits = user

    ? trpc.panel.getUserFreeCredits.useQuery()

    : null;

  const MAX_FREE_CREDITS =

    (Number(userFreeCredits?.data?.grant) ?? 0) + FREE_CREDITS;

  const usedCredits =

    MAX_FREE_CREDITS - Number(userFreeCredits?.data?.freeCredit);

  const usedPercentage = (usedCredits / MAX_FREE_CREDITS) * 100;

  const remainingPercentage =

    (Number(userFreeCredits?.data?.freeCredit) / MAX_FREE_CREDITS) * 100;

  return (

    <section className="mt-10 xl:mt-16">

      <p className="mb-3">

        Credit Rewards <span className="text-gray-400 ml-2">${</span>

      </p>

```

```
<div className="flex items-center gap-4">

  <div className="flex items-center gap-2">

    <p className="w-4 h-4 bg-primary rounded-sm" />

    <p className="text-sm">Used</p>

  </div>
```

```
<div className="flex items-center gap-2">

  <p className="w-4 h-4 bg-[#4BC0C0] rounded-sm" />

  <p className="text-sm">Available</p>

</div>

</div>
```

```
{user && (

  <div className="mt-4">

    {userFreeCredits?.isLoading ? (

      <LoadingSpinner />

    ) : (

      <>

        <div className="relative w-full h-6 rounded-sm">

          <div

            className="absolute h-6 bg-primary w-full"

            style={{ width: `${usedPercentage}%` }}

          ></div>

          <div

            className="absolute h-6 bg-[#4BC0C0] w-full"

            style={{
```

width: `\${remainingPercentage}%`,

left: `\${usedPercentage}%`,

}}

></div>

</div>

<p className="mt-2 text-right">

$\frac{\text{\${usedCredits.toFixed(2)}}}{\text{\${MAX_FREE_CREDITS.toFixed(2)}}$

</p>

</>

)}

</div>

)}

</section>

);

}