

QUẢN LÝ DỰ ÁN PHẦN MỀM



PROOF OF CONCEPTS

Project name:**The Tool For Content Management**

21127065

Trần Bình Kha

21127333

Nguyễn Việt Kim

21127334

Lê Vũ Ngân Lam

21127337

Trần Tùng Lâm

21127466

Hoàng Anh Tú

21127505

Ngô Xuân Hiếu

21127545

Đặng Quốc Thái

21127597

Đỗ Dự Đức



Khoa Công nghệ thông tin
Đại học Khoa học tự nhiên TP HCM

MỤC LỤC

I. Mã nguồn	3
1. Mã nguồn hiển thị UI phần form xử lý lấy input của user để generate content	3
2. Mã nguồn xử lý input từ user để gửi request tới OpenAI và stream output về lại user	4
3. Mã nguồn hiển thị output được stream về từ OpenAI lên UI	4
4. Mã nguồn xử lý phân quyền page	5
5. Mã nguồn xử lý luồng Authentication (SignIn, SignUp, SignOut)	5
6. Mã nguồn hiển thị màn hình xử lý thông tin thanh toán mua thêm credits từ user	7
II. Kết quả demo	7
1. Input:	7
2. Output:	8

Link mã nguồn: [Louis2602/TFCM: Tool for Content Management \(github.com\)](https://github.com/Louis2602/TFCM: Tool for Content Management)

I. Mã nguồn

1. Mã nguồn hiển thị UI phần form xử lý lấy input của user để generate content

```
0 D:\WORKSPACE\FREELANCE\T 13 | const [price, setPrice] = useState(0);
1 > drizzle 12
2 > public 11 | const router = useRouter();
3 > scripts 10
4 > x src 9
5 README.md 8
6 {} components.json 7
7 ts drizzle.config.ts 6
8 JS next.config.mjs 5
9 x package-lock.json 4
10 package.json 3
11 JS postcss.config.mjs 2
12 tailwind.config.ts 1
13 tsconfig.json 67

1 |
2 |
3 |
4 |
5 |
6 |
7 |
8 |
9 |
10 |
11 |
12 |
13 |
14 |
15 |
16 |
17 |

const { complete, completion, isLoading } = useCompletion({
  api: '/api/generate/content',
  onResponse: (response: Response) => {
    if (response.status === 429) {
      toast({
        title: "You have reached your request limit for the day.",
      });
      return;
    }
  },
  onError: (error) => {
    toast({
      title: "Something went wrong",
      description: error.message,
    });
  },
  onFinish() {
    toast({
      title: "Congratulations 🎉",
      description: "Your content has been generated successfully",
    });
    router.refresh();
  },
});

const prevRef = useRef("");

: Initializing JS/TS language features... tsserver
NORMAL develop src\...\content-writer\content-writer.tsx 3 4 239 18 24% 67:3 15:54
```

```
0 D:\WORKSPACE\FREELANCE\T 14 | <h2 className="font-heading text-3xl">Content Writer</h2>
1 > drizzle 13
2 > node_modules 12
3 > public 11 |
4 > scripts 10
5 > src 9
6 README.md 8
7 {} components.json 7
8 ts drizzle.config.ts 6
9 ts next-env.d.ts 5
10 JS next.config.mjs 4
11 package-lock.json 3
12 package.json 2
13 JS postcss.config.mjs 1
14 tailwind.config.ts 160
15 tsconfig.json 1

1 |
2 |
3 |
4 |
5 |
6 |
7 |
8 |
9 |
10 |
11 |
12 |
13 |
14 |
15 |
16 |

<h2 className="font-heading text-3xl">Content Writer</h2>
<p className="text-muted-foreground">
  Generate cutting-edge content in less than a minute.
</p>
</div>
<Form {...form}>
  <Form
    onSubmit={form.handleSubmit(onSubmit)}
    className="flex flex-col gap-4"
  >
    <FormField
      control={form.control}
      name="keywords"
      render={({ field }) => (
        <FormItem>
          <FormLabel>SEO keywords</FormLabel>
          <FormControl>
            <Input
              placeholder="Example: nextjs tutorial"
              type="text"
              {...field}
            />
          </FormControl>
          <FormMessage />
        </FormItem>
      )}
    />
    <FormField
      control={form.control}
      name="title"
      render={({ field }) => (
        <FormItem>
          <FormLabel>Title</FormLabel>
          <FormControl>
            <Input
              placeholder="Example: nextjs tutorial"
              type="text"
              {...field}
            />
          </FormControl>
          <FormMessage />
        </FormItem>
      )}
    />
  </Form>
</Form>

NORMAL develop src\...\content-writer\content-writer.tsx 3 4 239 18 58% 160:3 15:57
```

2. Mã nguồn xử lý input từ user để gửi request tới OpenAI và stream output về lại user

```
0 D:\WORKSPACE\FREELANCE\T 19
1 > drizzle 18
2 > node_modules 17
3 > public 16
4 > scripts 15
5 > src 14
6 README.md 13
7 {} components.json 12
8 TS drizzle.config.ts 11
9 TS next-env.d.ts 10
10 JS next.config.mjs 9
11 package-lock.json 8
12 package.json 7
13 postcss.config.mjs 6
14 tailwind.config.ts 5
15 tsconfig.json 4

33 if (!currentUser) {
17   return new Response("Unauthorized", { status: 403 });
16 }
15
14 let { prompt: content, price } = await req.json();
13
12 content = content.replace(/\\$/, "").slice(-5000) as string;
11
10 if (currentUser.credits < price) {
9   return new Response("Not enough credits to perform this action.", {
8     status: 403,
7   });
6 }
5
4 const response = await openai.createChatCompletion({
3   model: "gpt-3.5-turbo",
2   messages: [
1     {
33     role: "system",
1   content: `
2     You are a professtion content writer that responsible for writing articles based on
3     Your responses must be in Markdown format.
4     Here is an example for the content format:
5
6     =====
7
8     # Title of Your Document
9
10    ## Introduction
11  `
12    }
13  ]
14 }

NORMAL develop TS src\...\content\route.ts gj 31% 33:1 15:55
```

3. Mã nguồn hiển thị output được stream về từ OpenAI lên UI

```
0 D:\WORKSPACE\FREELANCE\T 20
1 > drizzle 19
2 > node_modules 18
3 > public 17
4 > scripts 16
5 > src 15
6 README.md 14
7 {} components.json 13
8 TS drizzle.config.ts 12
9 TS next-env.d.ts 11
10 JS next.config.mjs 10
11 package-lock.json 9
12 package.json 8
13 postcss.config.mjs 7
14 tailwind.config.ts 6
15 tsconfig.json 5

118 toast({
1   title: "Save to archive",
2   description: message,
3 });
4
5 return (
6   <div className="border border-border/50 rounded-lg w-full max-md:min-h-[30rem] max-h-[52rem]
7     <div className="h-full px-3 py-2 text-sm overflow-auto">
8       <MarkdownRenderer>{content.markdown}</MarkdownRenderer>
9     </div>
10
11     <div className="py-2 px-4 flex items-center justify-between bg-border/50">
12       <div className="text-sm">
13         <Hint label="Fullscreen">
14           <Icons.maximize
15             onClick={handleFullscreen}
16             size={18}
17             className="cursor-pointer"
18         />
19       </Hint>
20     </div>
21     <div className="flex gap-x-2">
22       <IconButton tooltip="Clear">
23         <Trash2 className="h-4 w-4" onClick={onClear} />
24       </IconButton>
25       <IconButton tooltip="Download" onClick={onDownload}>
26         <Download className="h-4 w-4" />
27       </IconButton>
28       <IconButton tooltip="Save">
29         <Save className="h-4 w-4" />
30       </IconButton>
31     </div>
32   </div>
33 )

NORMAL develop TS src\...\components\output-box.tsx ^U 82% 118:14 15:58
```

4. Mã nguồn xử lý phân quyền page

```
0 D:\WORKSPACE\FREELANCE\T
1 > drizzle
2 > node_modules
3 > public
4 > scripts
5 > src
6 README.md
7 {} components.json
8 TS drizzle.config.ts
9 TS next-env.d.ts
10 JS next.config.mjs
11 package-lock.json
12 package.json
13 JS postcss.config.mjs
14 tailwind.config.ts
15 tsconfig.json

20 import { clerkMiddleware, createRouteMatcher } from '@clerk/nextjs/server';
19 import { NextResponse } from 'next/server';
18
17 const isDashboardRoute = createRouteMatcher(['/dashboard(.*)']);
16 const isAdminRoute = createRouteMatcher(['/admin(.*)']);
15 const isAuthRoute = createRouteMatcher(['sign-in', 'sign-up']);
14 const isPublicRoute = createRouteMatcher([
13   '/',
12   '/sign-in(.*)',
11   '/sign-up(.*)',
10   '/api/webhooks(.*)',
9   '/api/uploadthing(.*)',
8 ]);
7
6 export default clerkMiddleware((auth, req) => {
5   // Restrict admin route to users with specific role
4   if (isAdminRoute(req)) auth().protect({ role: 'org:admin' });
3
2   // Restrict dashboard routes to signed in users
1   if (isDashboardRoute(req)) auth().protect();
21
1   if (!auth().userId && !isPublicRoute(req)) {
2     const signInUrl = new URL('/sign-in', req.url);
3     signInUrl.searchParams.set('redirect_url', req.url);
4     return NextResponse.redirect(signInUrl);
5   }
6
7   if (auth().userId && isAuthRoute(req)) {
8     const indexUrl = new URL('/', req.url);
9     return NextResponse.redirect(indexUrl);
10  }
11 }

NORMAL develop TS src\middleware.ts 58% 21:1 15:59
```

5. Mã nguồn xử lý luồng Authentication (SignIn, SignUp, SignOut)

```
0 D:\WORKSPACE\FREELANCE\T 1 import { authFeatureItems } from '@config/auth-features'; page.tsx
1 > drizzle 1 import { AuthFeature } from '@components/auth/auth-feature';
2 > node_modules 2 import type { Metadata } from 'next';
3 > public 3 import { SignIn } from '@clerk/nextjs';
4 > scripts 4
5 > src 5 export const metadata: Metadata = {
6   README.md 6   title: 'Sign in',
7   {} components.json 7   description: 'Sign in to your account',
8   ts drizzle.config.ts 8 };
9   ts next-env.d.ts 9
10 JS next.config.mjs 10 export default function SignInPage() {
11   package-lock.json 11   return (
12   package.json 12     <div className='relative container py-16 md:py-32 flex flex-col items-center'>
13   JS postcss.config.mjs 13       <div className='max-w-[51.25rem] w-full p-6 border border-border/50 rounded-lg bg-border/2
14   tailwind.config.ts 14         <div className='hidden px-6 w-full md:flex flex-col gap-12 pt-12'>
15   tsconfig.json 15           {authFeatureItems.map((feature) => (
16             <AuthFeature key={feature.label} {...feature} />
17           ))}
18         </div>
19         <div className='px-6 w-full flex flex-col'>
20           <SignIn />
21         </div>
22       </div>
23     </div>
24   );
25 }
```

NORMAL develop src\...\[...sign-in]\page.tsx 16:00

```
1 import { authFeatureItems } from '@config/auth-features'; page.tsx
1 import { AuthFeature } from '@components/auth/auth-feature';
2 import type { Metadata } from 'next';
3 import { SignUp } from '@clerk/nextjs';
4
5 export const metadata: Metadata = {
6   title: 'Sign up',
7   description: 'Create new account',
8 };
9
10 export default function SignUpPage() {
11   return (
12     <div className='relative container py-16 md:py-24 flex flex-col items-center'>
13       <div className='max-w-[51.25rem] w-full p-6 border border-border/50 rounded-lg bg-border/20 flex flex-col justify-between
14         <div className='hidden px-6 w-full md:flex flex-col gap-12 pt-12'>
15           {authFeatureItems.map((feature) => (
16             <AuthFeature key={feature.label} {...feature} />
17           ))}
18         </div>
19         <div className='px-6 w-full flex flex-col'>
20           <SignUp />
21         </div>
22       </div>
23     </div>
24   );
25 }
```

NORMAL develop src\...\[...sign-up]\page.tsx 16:01

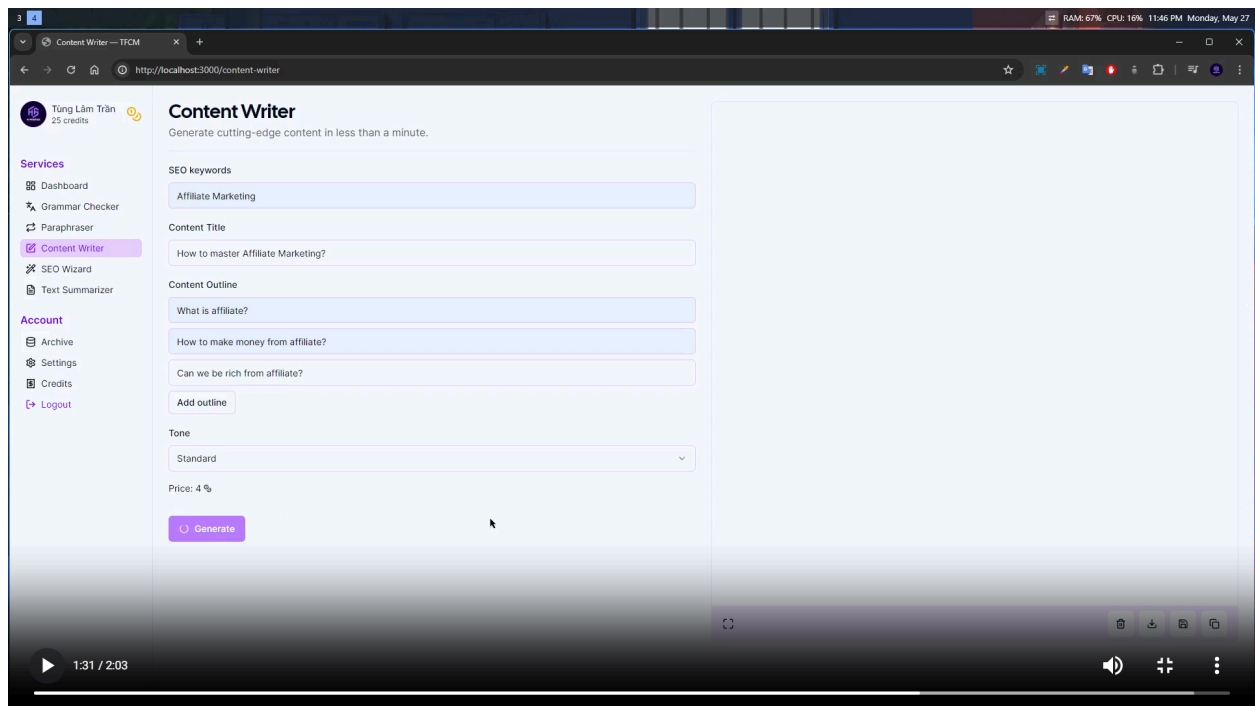
A < 8 < 3 Top 1:1 16:01

A < 8 < 3 75% 48:2 16:02

II. Kết quả demo

1. Input:

- SEO keywords
- Content Title
- Content Outlines
- Tone



2. Output:

- Nội dung content được generate ở output-box
- Credits được trừ trong tài khoản

