

■ Hackathon Project - Frontend Code Export

■ Frontend JavaScript/JSX Files:

```
[JS] eslint.config.js
[JS] postcss.config.js
[JSX] src\App.jsx
[JSX] src\components\blog-generator\BlogGenerator.jsx
[JSX] src\components\blog-generator\BlogInput.jsx
[JSX] src\components\blog-generator\BlogOutput.jsx
[JSX] src\components\blog-generator\ExamplePrompts.jsx
[JSX] src\components\blog-generator\FeatureCards.jsx
[JSX] src\components\blog-generator\HeroSection.jsx
[JSX] src\components\ui\alert.jsx
[JSX] src\components\ui\badge.jsx
[JSX] src\components\ui\button.jsx
[JSX] src\components\ui\card.jsx
[JSX] src\components\ui\separator.jsx
[JSX] src\components\ui\tabs.jsx
[JSX] src\components\ui\textarea.jsx
[JS] src\hooks\useBlogGenerator.js
[JS] src\hooks\useLocalStorage.js
[JSX] src\main.jsx
[JS] src\services\blogApi.js
[JS] src\utils\blogMetrics.js
[JS] src\utils\constants.js
[JS] tailwind.config.js
[JS] vite.config.js
```

■ File: eslint.config.js

```
=====
1: import js from '@eslint/js'
2: import globals from 'globals'
3: import reactHooks from 'eslint-plugin-react-hooks'
4: import reactRefresh from 'eslint-plugin-react-refresh'
5: import { defineConfig, globalIgnores } from 'eslint/config'
6:
7: export default defineConfig([
8:   globalIgnores(['dist']),
9:   {
10:     files: ['**/*.{js,jsx}'],
11:     extends: [
12:       js.configs.recommended,
13:       reactHooks.configs.flat.recommended,
14:       reactRefresh.configs.vite,
15:     ],
16:     languageOptions: {
17:       ecmaVersion: 2020,
18:       globals: globals.browser,
19:       parserOptions: {
20:         ecmaVersion: 'latest',
21:         ecmaFeatures: { jsx: true },
22:         sourceType: 'module',
23:       },
24:     },
25:     rules: {
26:       'no-unused-vars': ['error', { varsIgnorePattern: '^[A-Z_]' }],
27:     },
28:   },
29: ])=====
```

■ File: postcss.config.js

```
=====
1: export default {
2:   plugins: {
3:     tailwindcss: {},
4:     autoprefixer: {},
5:   },
6: };=====
```

■ File: src\App.jsx

```
=====
1: // src/App.jsx
2: import { Toaster } from 'sonner';
3: import BlogGenerator from './components/blog-generator/BlogGenerator';
4: import './App.css';
5:
6: function App() {
7:   return (
8:     <>
9:       <BlogGenerator />
10:      <Toaster
11:        position="top-right"
12:        richColors
13:        closeButton
14:        duration={4000}
15:      />
16:     </>
17:   );
18: }
19:
20: export default App;
=====
```

■ File: src\components\blog-generator\BlogGenerator.jsx

```
=====
1: // src/components/blog-generator/BlogGenerator.jsx
2: import { useEffect } from "react";
3: import { toast } from "sonner";
4: import { useBlogGenerator } from "../../hooks/useBlogGenerator";
5: import { FEATURES } from "../../utils/constants";
6: import HeroSection from "./HeroSection";
7: import BlogInput from "./BlogInput";
8: import BlogOutput from "./BlogOutput";
9: import FeatureCards from "./FeatureCards";
10:
11: export default function BlogGenerator() {
12:   const {
13:     prompt,
14:     setPrompt,
15:     blog,
16:     isLoading,
17:     error,
18:     generate,
19:     reset,
20:     copyToClipboard,
21:   } = useBlogGenerator();
22:
23:   const handleGenerate = async () => {
24:     const result = await generate();
25:
26:     if (result.success) {
27:       toast.success("Blog generated successfully!", {
28:         description: `${result.blog.metrics.wordCount} words in ${result.blog.metrics.generationTime}s`,
29:       });
30:
31:       // Scroll to result
32:       setTimeout(() => {
33:         document.getElementById("blog-output")?.scrollIntoView({
34:           behavior: "smooth",
35:           block: "start",
36:         });
37:       }, 100);
38:     } else {
39:       toast.error("Generation failed", {
40:         description: result.error,
41:       });
42:     }
43:   };
44:
45:   const handleCopy = async () => {
46:     const result = await copyToClipboard();
47:     if (result.success) {
48:       toast.success("Copied to clipboard!");
49:     } else {
50:       toast.error("Failed to copy to clipboard");
51:     }
52:     return result;
53:   };
54:
55:   const handleReset = () => {
56:     reset();
57:     toast.info("Generator reset");
58:   };
59:
60:   const handleExampleSelect = (example) => {
61:     setPrompt(example);
62:     toast.info("Example prompt loaded");
63:   };
64:
65:   return (
66:     <div className="min-h-screen bg-gradient-to-br from-slate-50 via-blue-50 to-indigo-50 dark:from-slate-900 dark:to-indigo-900">
67:       <HeroSection />
68:
69:       <div className="container mx-auto px-4 py-8 md:py-12">
70:         <div className="max-w-7xl mx-auto space-y-8">
71:           <BlogInput
```

```

72:         prompt={prompt}
73:         setPrompt={setPrompt}
74:         isLoading={isLoading}
75:         error={error}
76:         onGenerate={handleGenerate}
77:         onReset={handleReset}
78:         onExampleSelect={handleExampleSelect}
79:     />
80:
81:         {blog && <BlogOutput blog={blog} onCopy={handleCopy} />}
82:     </div>
83:   </div>
84: </div>
85: );
86: }

```

■ File: src\components\blog-generator\BlogInput.jsx

```

1: // src/components/blog-generator/BlogInput.jsx
2: import { Loader2, Sparkles, Zap, ArrowRight, RefreshCw, Brain, AlertCircle } from 'lucide-react';
3: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from '../ui/card';
4: import { Button } from '../ui/button';
5: import { Textarea } from '../ui/textarea';
6: import { Alert, AlertDescription } from '../ui/alert';
7: import ExamplePrompts from './ExamplePrompts';
8: import { PROMPT_CONFIG } from '../../utils/constants';
9:
10: export default function BlogInput({
11:   prompt,
12:   setPrompt,
13:   isLoading,
14:   error,
15:   onGenerate,
16:   onReset
17: }) {
18:   const handleExampleSelect = (example) => {
19:     setPrompt(example);
20:   };
21:
22:   return (
23:     <Card className="border-2 shadow-lg">
24:       <CardHeader>
25:         <CardTitle className="flex items-center gap-2 text-2xl">
26:           <Sparkles className="h-6 w-6 text-blue-600" />
27:           Create Your Blog
28:         </CardTitle>
29:         <CardDescription className="text-base">
30:           Describe your topic in detail. The more specific you are, the better the results.
31:         </CardDescription>
32:       </CardHeader>
33:       <CardContent className="space-y-6">
34:         <div className="space-y-3">
35:           <Textarea
36:             placeholder={PROMPT_CONFIG.PLACEHOLDER}
37:             value={prompt}
38:             onChange={(e) => setPrompt(e.target.value)}
39:             rows={6}
40:             className="resize-none text-base"
41:             disabled={isLoading}
42:             maxLength={PROMPT_CONFIG.MAX_LENGTH}
43:           />
44:           <div className="flex items-center justify-between text-sm text-muted-foreground">
45:             <span>{prompt.length}/{PROMPT_CONFIG.MAX_LENGTH} characters</span>
46:             {prompt && (
47:               <Button
48:                 variant="ghost"
49:                 size="sm"
50:                 onClick={onReset}
51:                 className="h-8"
52:                 disabled={isLoading}
53:               >

```

```

54:             <RefreshCw className="h-3 w-3 mr-1" />
55:             Clear
56:         </Button>
57:     )
58:   </div>
59: </div>
60:
61:   <ExamplePrompts
62:     onSelectPrompt={handleExampleSelect}
63:     disabled={isLoading}
64:   />
65:
66:   {error && (
67:     <Alert variant="destructive">
68:       <AlertCircle className="h-4 w-4" />
69:       <AlertDescription>{error}</AlertDescription>
70:     </Alert>
71:   )}
72:
73:   <Button
74:     onClick={onGenerate}
75:     disabled={isLoading || !prompt.trim()}
76:     className="w-full h-12 text-base"
77:     size="lg"
78:   >
79:     {isLoading ? (
80:       <>
81:         <Loader2 className="mr-2 h-5 w-5 animate-spin" />
82:         Generating Your Blog...
83:       </>
84:     ) : (
85:       <>
86:         <Zap className="mr-2 h-5 w-5" />
87:         Generate Professional Blog
88:         <ArrowRight className="ml-2 h-4 w-4" />
89:       </>
90:     )}
91:   </Button>
92:
93:   <div className="flex items-center justify-center gap-2 text-xs text-muted-foreground">
94:     <span className="inline-flex items-center gap-1">
95:       <Brain className="h-3 w-3" />
96:       Powered by Llama 3.3 70B
97:     </span>
98:     <span>?</span>
99:     <span>Average generation: 5-10 seconds</span>
100:   </div>
101: </CardContent>
102: </Card>
103: );
104: }

```

■ File: src\components\blog-generator\BlogOutput.jsx

```

1: // src/components/blog-generator/BlogOutput.jsx
2: import { useState } from "react";
3: import { Copy, Download, CheckCircle } from "lucide-react";
4: import ReactMarkdown from "react-markdown";
5: import {
6:   Card,
7:   CardContent,
8:   CardDescription,
9:   CardHeader,
10:  CardTitle,
11: } from "../ui/card";
12: import { Button } from "../ui/button";
13: import { Separator } from "../ui/sePARATOR";
14: import { Tabs, TabsContent, TabsList, TabsTrigger } from "../ui/tabs";
15: import { downloadBlog } from "../../utils/blogMetrics";
16:
17: export default function BlogOutput({ blog, onCopy }) {

```

```
18: const [viewMode, setViewMode] = useState("preview");
19: const [copied, setCopied] = useState(false);
20:
21: const handleCopy = async () => {
22:   const result = await onCopy();
23:   if (result.success) {
24:     setCopied(true);
25:     setTimeout(() => setCopied(false), 2000);
26:   }
27: };
28:
29: const handleDownload = () => {
30:   downloadBlog(blog.content);
31: };
32:
33: return (
34:   <Card className="border-2 shadow-lg" id="blog-output">
35:     <CardHeader>
36:       <div className="flex flex-col sm:flex-row sm:items-center justify-between gap-4">
37:         <div className="space-y-1">
38:           <CardTitle className="text-2xl">Generated Blog</CardTitle>
39:           <CardDescription>
40:             Created on {new Date(blog.timestamp).toLocaleString()}
41:           </CardDescription>
42:         </div>
43:         <div className="flex gap-2">
44:           <Button
45:             variant="outline"
46:             size="sm"
47:             onClick={handleCopy}
48:             className="gap-2"
49:           >
50:             {copied ? (
51:               <>
52:                 <CheckCircle className="h-4 w-4 text-green-600" />
53:                 Copied!
54:               </>
55:             ) : (
56:               <>
57:                 <Copy className="h-4 w-4" />
58:                 Copy
59:               </>
60:             )}
61:           </Button>
62:           <Button
63:             variant="outline"
64:             size="sm"
65:             onClick={handleDownload}
66:             className="gap-2"
67:           >
68:             <Download className="h-4 w-4" />
69:             Download
70:           </Button>
71:         </div>
72:       </div>
73:
74:       <Separator className="mt-4" />
75:
76: /* Metrics */
77: <div className="grid grid-cols-2 sm:grid-cols-4 gap-4 pt-4">
78:   <div className="space-y-1">
79:     <p className="text-sm text-muted-foreground">Words</p>
80:     <p className="text-2xl font-bold">
81:       {blog.metrics.wordCount.toLocaleString()}
82:     </p>
83:   </div>
84:   <div className="space-y-1">
85:     <p className="text-sm text-muted-foreground">Characters</p>
86:     <p className="text-2xl font-bold">
87:       {blog.metrics.characterCount.toLocaleString()}
88:     </p>
89:   </div>
90:   <div className="space-y-1">
```

```

91:         <p className="text-sm text-muted-foreground">Read Time</p>
92:         <p className="text-2xl font-bold">{blog.metrics.readingTime} min</p>
93:     </div>
94:     <div className="space-y-1">
95:         <p className="text-sm text-muted-foreground">Generated</p>
96:         <p className="text-2xl font-bold">{blog.metrics.generationTime}s</p>
97:     </div>
98:   </div>
99: </CardHeader>
100:
101: <CardContent>
102:   <Tabs value={viewMode} onValueChange={setViewMode}>
103:     <TabsList className="grid w-full max-w-md grid-cols-2">
104:       <TabsTrigger value="preview">Preview</TabsTrigger>
105:       <TabsTrigger value="markdown">Markdown</TabsTrigger>
106:     </TabsList>
107:
108:     <TabsContent value="preview" className="mt-6">
109:       <div className="prose prose-slate dark:prose-invert max-w-none prose-headings:font-bold prose-h">
110:         <ReactMarkdown>{blog.content}</ReactMarkdown>
111:       </div>
112:     </TabsContent>
113:
114:     <TabsContent value="markdown" className="mt-6">
115:       <pre className="p-6 bg-slate-50 dark:bg-slate-900 rounded-lg overflow-x-auto text-sm">
116:         <code>{blog.content}</code>
117:       </pre>
118:     </TabsContent>
119:   </Tabs>
120: </CardContent>
121: </Card>
122: );
123: }

```

■ File: src\components\blog-generator\ExamplePrompts.jsx

```

1: // src/components/blog-generator/ExamplePrompts.jsx
2: import { FileText } from "lucide-react";
3: import { EXAMPLE_PROMPTS } from "../../utils/constants";
4:
5: export default function ExamplePrompts({ onSelectPrompt, disabled }) {
6:   return (
7:     <div className="space-y-3">
8:       <p className="text-sm font-medium">
9:         Try these examples
10:      </p>
11:      <div className="grid gap-2 sm:grid-cols-2">
12:        {EXAMPLE_PROMPTS.map((example, index) => (
13:          <button
14:            key={index}
15:            onClick={() => onSelectPrompt(example)}
16:            disabled={disabled}
17:            className="text-left p-3 rounded-lg border border-dashed hover:border-solid hover:bg-accent tra
18:          >
19:            <span className="flex items-center gap-2">
20:              <FileText className="h-4 w-4 shrink-0 text-muted-foreground" />
21:              <span className="truncate">{example}</span>
22:            </span>
23:          </button>
24:        ))}
25:      </div>
26:    </div>
27:  );
28: }

```

■ File: src\components\blog-generator\FeatureCards.jsx

```

1: // src/components/blog-generator/FeatureCards.jsx
2: import { FileText, Sparkles, Zap, CheckCircle } from "lucide-react";

```

```

3: import { Card,CardContent } from "../ui/card";
4:
5: const iconMap = {
6:   FileText,
7:   Sparkles,
8:   Zap,
9:   CheckCircle,
10: };
11:
12: export default function FeatureCards({ features }) {
13:   return (
14:     <div className="grid gap-6 sm:grid-cols-2 lg:grid-cols-4">
15:       {features.map((feature, index) => {
16:         const Icon = iconMap[feature.icon];
17:         return (
18:           <Card key={index} className="border-2">
19:             <CardContent className="pt-6">
20:               <div className="space-y-3">
21:                 <div className="flex h-12 w-12 items-center justify-center rounded-lg bg-blue-100 dark:bg-blue-900">
22:                   <Icon className="h-6 w-6 text-blue-600 dark:text-blue-400" />
23:                 </div>
24:                 <h3 className="font-semibold text-lg">{feature.title}</h3>
25:                 <p className="text-sm text-muted-foreground">
26:                   {feature.description}
27:                 </p>
28:               </div>
29:             </CardContent>
30:           </Card>
31:         );
32:       ))}
33:     </div>
34:   );
35: }

```

■ File: src\components\blog-generator\HeroSection.jsx

```

1: // src/components/blog-generator/HeroSection.jsx
2: import { Brain, CheckCircle } from "lucide-react";
3:
4: export default function HeroSection() {
5:   return (
6:     <div className="border-b bg-white/50 dark:bg-slate-900/50 backdrop-blur-sm"></div>
7:   );
8: }

```

■ File: src\components\ui\alert.jsx

```

1: // src/components/ui/alert.jsx
2: import * as React from "react"
3:
4: const alertVariants = {
5:   default: "bg-background text-foreground",
6:   destructive: "border-destructive/50 text-destructive dark:border-destructive [&>svg]:text-destructive",
7: }
8:
9: const Alert = React.forwardRef(({ className = "", variant = "default", ...props }, ref) => {
10:   const variantClass = alertVariants[variant] || alertVariants.default
11:
12:   return (
13:     <div
14:       ref={ref}
15:       role="alert"
16:       className={`relative w-full rounded-lg border p-4 [&>svg~*]:pl-7 [&>svg+div]:translate-y-[-3px] [&>sv`]
17:       {...props}
18:     />
19:   )
20: })
21: Alert.displayName = "Alert"
22:

```

```

23: const AlertTitle = React.forwardRef(({ className = "", ...props }, ref) => (
24:   <h5
25:     ref={ref}
26:     className={`mb-1 font-medium leading-none tracking-tight ${className}`}
27:     {...props}
28:   />
29: ))
30: AlertTitle.displayName = "AlertTitle"
31:
32: const AlertDescription = React.forwardRef(({ className = "", ...props }, ref) => (
33:   <div
34:     ref={ref}
35:     className={`text-sm [&_p]:leading-relaxed ${className}`}
36:     {...props}
37:   />
38: ))
39: AlertDescription.displayName = "AlertDescription"
40:
41: export { Alert, AlertTitle, AlertDescription }

```

■ File: src\components\ui\badge.jsx

```

1: // src/components/ui/badge.jsx
2: import * as React from "react";
3:
4: const badgeVariants = {
5:   default:
6:     "border-transparent bg-primary text-primary-foreground hover:bg-primary/80",
7:   secondary:
8:     "border-transparent bg-secondary text-secondary-foreground hover:bg-secondary/80",
9:   destructive:
10:    "border-transparent bg-destructive text-destructive-foreground hover:bg-destructive/80",
11:   outline: "text-foreground",
12: };
13:
14: function Badge({ className = "", variant = "default", ...props }) {
15:   const variantClass = badgeVariants[variant] || badgeVariants.default;
16:
17:   return (
18:     <div
19:       className={`inline-flex items-center rounded-full border px-2.5 py-0.5 text-xs font-semibold transition`}
20:       {...props}
21:     />
22:   );
23: }
24:
25: export { Badge };

```

■ File: src\components\ui\button.jsx

```

1: // src/components/ui/button.jsx
2: import * as React from "react";
3:
4: const buttonVariants = {
5:   default: "bg-primary text-primary-foreground hover:bg-primary/90",
6:   destructive:
7:     "bg-destructive text-destructive-foreground hover:bg-destructive/90",
8:   outline:
9:     "border border-input bg-background hover:bg-accent hover:text-accent-foreground",
10:  secondary: "bg-secondary text-secondary-foreground hover:bg-secondary/80",
11:  ghost: "hover:bg-accent hover:text-accent-foreground",
12:  link: "text-primary underline-offset-4 hover:underline",
13: };
14:
15: const buttonSizes = {
16:   default: "h-10 px-4 py-2",
17:   sm: "h-9 rounded-md px-3",
18:   lg: "h-11 rounded-md px-8",
19:   icon: "h-10 w-10",

```

```

20: };
21:
22: const Button = React.forwardRef(
23:   (
24:     { className = "", variant = "default", size = "default", ...props },
25:     ref
26:   ) => {
27:     const variantClass = buttonVariants[variant] || buttonVariants.default;
28:     const sizeClass = buttonSizes[size] || buttonSizes.default;
29:
30:     return (
31:       <button
32:         className={`inline-flex items-center justify-center whitespace nowrap rounded-md text-sm font-medium ${variantClass} ${sizeClass}`}
33:         ref={ref}
34:         {...props}
35:       />
36:     );
37:   }
38: );
39: Button.displayName = "Button";
40:
41: export { Button };

```

■ File: src\components\ui\card.jsx

```

1: // src/components/ui/card.jsx
2: import * as React from "react";
3:
4: const Card = React.forwardRef(({ className = "", ...props }, ref) => (
5:   <div
6:     ref={ref}
7:     className={`${rounded-lg border bg-card text-card-foreground shadow-sm ${className}`}`}
8:     {...props}
9:   />
10: ));
11: Card.displayName = "Card";
12:
13: const CardHeader = React.forwardRef(({ className = "", ...props }, ref) => (
14:   <div
15:     ref={ref}
16:     className={`${flex flex-col space-y-1.5 p-6 ${className}`}`}
17:     {...props}
18:   />
19: ));
20: CardHeader.displayName = "CardHeader";
21:
22: const CardTitle = React.forwardRef(({ className = "", ...props }, ref) => (
23:   <h3
24:     ref={ref}
25:     className={`${text-2xl font-semibold leading-none tracking-tight ${className}`}`}
26:     {...props}
27:   />
28: ));
29: CardTitle.displayName = "CardTitle";
30:
31: const CardDescription = React.forwardRef(
32:   ({ className = "", ...props }, ref) => (
33:     <p
34:       ref={ref}
35:       className={`${text-sm text-muted-foreground ${className}`}`}
36:       {...props}
37:     />
38:   )
39: );
40: CardDescription.displayName = "CardDescription";
41:
42: constCardContent = React.forwardRef(({ className = "", ...props }, ref) => (
43:   <div ref={ref} className={`${p-6 pt-0 ${className}`}} {...props} />
44: ));
45:CardContent.displayName = "CardContent";
46:

```

```
47: const CardFooter = React.forwardRef(({ className = "", ...props }, ref) => (
48:   <div
49:     ref={ref}
50:     className={`flex items-center p-6 pt-0 ${className}`}
51:     {...props}
52:   />
53: ));
54: CardFooter.displayName = "CardFooter";
55:
56: export {
57:   Card,
58:   CardHeader,
59:   CardFooter,
60:   CardTitle,
61:   CardDescription,
62:   CardContent,
63: };
```

■ File: src\components\ui\separator.jsx

```
1: // src/components/ui/separator.jsx
2: import * as React from "react";
3:
4: const Separator = React.forwardRef(
5:   (
6:     { className = "", orientation = "horizontal", decorative = true, ...props },
7:     ref
8:   ) => (
9:     <div
10:       ref={ref}
11:       role={decorative ? "none" : "separator"}
12:       aria-orientation={orientation}
13:       className={`shrink-0 bg-border ${
14:         orientation === "horizontal" ? "h-[1px] w-full" : "h-full w-[1px]"
15:       } ${className}`}
16:       {...props}
17:     />
18:   )
19: );
20: Separator.displayName = "Separator";
21:
22: export { Separator };
```

■ File: src\components\ui\tabs.jsx

```
1: // src/components/ui/tabs.jsx
2: import * as React from "react";
3:
4: const TabsContext = React.createContext({});
5:
6: const Tabs = ({
7:   value,
8:   onValueChange,
9:   defaultValue,
10:  children,
11:  className = "",
12: }) => {
13:   const [selectedValue, setSelectedValue] = React.useState(
14:     defaultValue || value
15:   );
16:
17:   React.useEffect(() => {
18:     if (value !== undefined) {
19:       setSelectedValue(value);
20:     }
21:   }, [value]);
22:
23:   const handleValueChange = (newValue) => {
24:     setSelectedValue(newValue);
```

```

25:     if (onValueChange) {
26:         onValueChange(newValue);
27:     }
28:   };
29:
30:   return (
31:     <TabsContext.Provider
32:       value={{ value: selectedValue, onValueChange: handleValueChange }}
33:     >
34:       <div className={className}>{children}</div>
35:     </TabsContext.Provider>
36:   );
37: };
38:
39: const TabsList = React.forwardRef(({ className = "", ...props }, ref) => (
40:   <div
41:     ref={ref}
42:     className={`inline-flex h-10 items-center justify-center rounded-md bg-muted p-1 text-muted-foreground ${props.className}`}
43:     {...props}
44:   />
45: ));
46: TabsList.displayName = "TabsList";
47:
48: const TabsTrigger = React.forwardRef(
49:   ({ className = "", value, ...props }, ref) => {
50:     const { value: selectedValue, onValueChange } =
51:       React.useContext(TabsContext);
52:     const isSelected = selectedValue === value;
53:
54:     return (
55:       <button
56:         ref={ref}
57:         className={`inline-flex items-center justify-center whitespace nowrap rounded-sm px-3 py-1.5 text-sm ${props.className}`}
58:         isSelected
59:           ? "bg-background text-foreground shadow-sm"
60:           : "hover:bg-background/50"
61:         } ${props.className}`}
62:         onClick={() => onValueChange(value)}
63:         {...props}
64:       />
65:     );
66:   }
67: );
68: TabsTrigger.displayName = "TabsTrigger";
69:
70: const TabsContent = React.forwardRef(
71:   ({ className = "", value, ...props }, ref) => {
72:     const { value: selectedValue } = React.useContext(TabsContext);
73:
74:     if (selectedValue !== value) {
75:       return null;
76:     }
77:
78:     return (
79:       <div
80:         ref={ref}
81:         className={`mt-2 ring-offset-background focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-inset ${props.className}`}
82:         {...props}
83:       />
84:     );
85:   }
86: );
87: TabsContent.displayName = "TabsContent";
88:
89: export { Tabs, TabsList, TabsTrigger, TabsContent };

```

■ File: src\components\ui\textarea.jsx

```

1: // src/components/ui/textarea.jsx
2: import * as React from "react"
3:

```

```

4: const Textarea = React.forwardRef(({ className = "", ...props }, ref) => {
5:   return (
6:     <textarea
7:       className={`flex min-h-[80px] w-full rounded-md border border-input bg-background px-3 py-2 text-sm r
8:       ref={ref}
9:       {...props}
10:      />
11:    )
12:  })
13: Textarea.displayName = "Textarea"
14:
15: export { Textarea }

```

■ File: src\hooks\useBlogGenerator.js

```

1: // src/hooks/useBlogGenerator.js
2: import { useState, useCallback } from "react";
3: import { generateBlog } from "../services/blogApi";
4: import { calculateBlogMetrics } from "../utils/blogMetrics";
5: import { useLocalStorage } from "./useLocalStorage";
6: import { PROMPT_CONFIG } from "../utils/constants";
7:
8: export function useBlogGenerator() {
9:   const [prompt, setPrompt] = useLocalStorage("blog-prompt", "");
10:  const [blog, setBlog] = useState(null);
11:  const [isLoading, setIsLoading] = useState(false);
12:  const [error, setError] = useState(null);
13:
14:  const validatePrompt = (text) => {
15:    if (!text.trim()) {
16:      return "Please enter a prompt";
17:    }
18:    if (text.trim().length < PROMPT_CONFIG.MIN_LENGTH) {
19:      return `Please provide a more detailed prompt (at least ${PROMPT_CONFIG.MIN_LENGTH} characters)`;
20:    }
21:    return null;
22:  };
23:
24:  const generate = useCallback(async () => {
25:    const validationError = validatePrompt(prompt);
26:    if (validationError) {
27:      setError(validationError);
28:      return { success: false, error: validationError };
29:    }
30:
31:    setIsLoading(true);
32:    setError(null);
33:    setBlog(null);
34:
35:    const startTime = Date.now();
36:
37:    try {
38:      const data = await generateBlog(prompt);
39:      const generationTime = (Date.now() - startTime) / 1000;
40:      const metrics = calculateBlogMetrics(data.blog, generationTime);
41:
42:      const blogData = {
43:        content: data.blog,
44:        metrics,
45:        timestamp: new Date().toISOString(),
46:      };
47:
48:      setBlog(blogData);
49:      return { success: true, blog: blogData };
50:    } catch (err) {
51:      const errorMessage =
52:        err instanceof Error ? err.message : "Failed to generate blog";
53:      setError(errorMessage);
54:      return { success: false, error: errorMessage };
55:    } finally {
56:      setIsLoading(false);

```

```

57:     }
58:   }, [prompt]);
59:
60: const reset = useCallback(() => {
61:   setPrompt("");
62:   setBlog(null);
63:   setError(null);
64: }, [setPrompt]);
65:
66: const copyToClipboard = useCallback(async () => {
67:   if (!blog) return { success: false };
68:
69:   try {
70:     await navigator.clipboard.writeText(blog.content);
71:     return { success: true };
72:   } catch (err) {
73:     return { success: false, error: "Failed to copy to clipboard" };
74:   }
75: }, [blog]);
76:
77: return {
78:   prompt,
79:   setPrompt,
80:   blog,
81:   isLoading,
82:   error,
83:   generate,
84:   reset,
85:   copyToClipboard,
86: };
87: }

```

■ File: src\hooks\useLocalStorage.js

```

1: // src/hooks/useLocalStorage.js
2: import { useState, useEffect } from "react";
3:
4: /**
5:  * Hook to sync state with localStorage
6:  * @param {string} key - localStorage key
7:  * @param {*} initialValue - Initial value if key doesn't exist
8:  * @returns {[any, Function]} [value, setValue]
9: */
10: export function useLocalStorage(key, initialValue) {
11:   const [storedValue, setStoredValue] = useState(() => {
12:     try {
13:       const item = window.localStorage.getItem(key);
14:       return item ? JSON.parse(item) : initialValue;
15:     } catch (error) {
16:       console.error("Error reading from localStorage:", error);
17:       return initialValue;
18:     }
19:   });
20:
21:   const setValue = (value) => {
22:     try {
23:       const valueToString =
24:         value instanceof Function ? value(storedValue) : value;
25:       setStoredValue(valueToString);
26:
27:       if (
28:         valueToString === null ||
29:         valueToString === undefined ||
30:         valueToString === ""
31:     ) {
32:       window.localStorage.removeItem(key);
33:     } else {
34:       window.localStorage.setItem(key, JSON.stringify(valueToString));
35:     }
36:   } catch (error) {
37:     console.error("Error writing to localStorage:", error);

```

```
38:     }
39:   };
40:
41:   return [storedValue, setValue];
42: }
```

■ File: src\main.jsx

```
=====
1: import { StrictMode } from 'react'
2: import { createRoot } from 'react-dom/client'
3: import './index.css'
4: import App from './App.jsx'
5:
6: createRoot(document.getElementById('root')).render(
7:   <StrictMode>
8:     <App />
9:   </StrictMode>,
10: )
```

■ File: src\services\blogApi.js

```
=====
1: // src/services/blogApi.js
2: import { API_CONFIG } from '../utils/constants';
3:
4: /**
5:  * Generate blog using API
6:  * @param {string} prompt - User prompt
7:  * @returns {Promise<Object>} API response
8: */
9: export async function generateBlog(prompt) {
10:   const response = await fetch(API_CONFIG.ENDPOINT, {
11:     method: 'POST',
12:     headers: {
13:       'Content-Type': 'application/json',
14:       'Accept': 'application/json',
15:     },
16:     body: JSON.stringify({ prompt: prompt.trim() }),
17:   });
18:
19:   const data = await response.json();
20:
21:   if (!response.ok) {
22:     throw new Error(data.error || `Failed to generate blog (${response.status})`);
23:   }
24:
25:   if (!data.blog || data.blog.trim().length === 0) {
26:     throw new Error('Generated blog is empty. Please try again.');
27:   }
28:
29:   return data;
30: }
31:
32: /**
33:  * Check API health
34:  * @returns {Promise<Object>} Health status
35: */
36: export async function checkApiHealth() {
37:   const response = await fetch(API_CONFIG.ENDPOINT, {
38:     method: 'GET',
39:   });
40:
41:   return response.json();
42: }
```

■ File: src\utils\blogMetrics.js

```
1: // src/utils/blogMetrics.js
2:
3: /**
4:  * Calculate blog content metrics
5:  * @param {string} content - The blog content
6:  * @param {number} generationTime - Time taken to generate (in seconds)
7:  * @returns {Object} Metrics object
8: */
9: export function calculateBlogMetrics(content, generationTime) {
10:   const words = content
11:     .trim()
12:     .split(/\s+/)
13:     .filter((word) => word.length > 0);
14:   const wordCount = words.length;
15:   const characterCount = content.length;
16:   const readingTime = Math.ceil(wordCount / 200); // 200 words per minute average
17:
18:   return {
19:     wordCount,
20:     characterCount,
21:     readingTime,
22:     generationTime: generationTime.toFixed(1),
23:   };
24: }
25:
26: /**
27:  * Download blog as markdown file
28:  * @param {string} content - Blog content
29: */
30: export function downloadBlog(content) {
31:   const element = document.createElement("a");
32:   const file = new Blob([content], { type: "text/markdown;charset=utf-8" });
33:   element.href = URL.createObjectURL(file);
34:   element.download = `blog-${new Date().toISOString().split("T")[0]}.md`;
35:   document.body.appendChild(element);
36:   element.click();
37:   document.body.removeChild(element);
38:   URL.revokeObjectURL(element.href);
39: }
```

■ File: src\utils\constants.js

```
1: // src/utils/constants.js
2:
3: export const EXAMPLE_PROMPTS = [
4:   "The impact of artificial intelligence on modern healthcare",
5:   "Sustainable living practices for urban environments",
6:   "The future of remote work and digital nomadism",
7:   "Blockchain technology and its real-world applications"
8: ];
9:
10: export const FEATURES = [
11:   {
12:     title: '1000+ Words',
13:     description: 'Comprehensive blog posts with rich content',
14:     icon: 'FileText'
15:   },
16:   {
17:     title: 'AI-Powered',
18:     description: 'Advanced language model for quality output',
19:     icon: 'Sparkles'
20:   },
21:   {
22:     title: 'Lightning Fast',
23:     description: 'Generate blogs in 5-10 seconds',
24:     icon: 'Zap'
25:   },
26:   {
27:     title: 'Pro Formatting',
```

```
28:     description: 'Perfect structure with markdown support',
29:     icon: 'CheckCircle'
30:   }
31: ];
32:
33: export const PROMPT_CONFIG = {
34:   MIN_LENGTH: 10,
35:   MAX_LENGTH: 1000,
36:   PLACEHOLDER: "Example: Write a comprehensive guide about the benefits of meditation, including scientific",
37: };
38:
39: export const API_CONFIG = {
40:   ENDPOINT: import.meta.env.VITE_API_ENDPOINT || '/api/generate-blog',
41:   TIMEOUT: 30000
42: };
```

■ File: tailwind.config.js

```
1: /** @type {import('tailwindcss').Config} */
2: export default {
3:   content: [
4:     "./index.html",
5:     "./src/**/*.{js,ts,jsx,tsx}",
6:   ],
7:   theme: {
8:     extend: {
9:       colors: {
10:         border: "hsl(214.3 31.8% 91.4%)",
11:         input: "hsl(214.3 31.8% 91.4%)",
12:         ring: "hsl(221.2 83.2% 53.3%)",
13:         background: "hsl(0 0% 100%)",
14:         foreground: "hsl(222.2 84% 4.9%)",
15:         primary: {
16:           DEFAULT: "hsl(221.2 83.2% 53.3%)",
17:           foreground: "hsl(210 40% 98%)",
18:         },
19:         secondary: {
20:           DEFAULT: "hsl(210 40% 96.1%)",
21:           foreground: "hsl(222.2 47.4% 11.2%)",
22:         },
23:         destructive: {
24:           DEFAULT: "hsl(0 84.2% 60.2%)",
25:           foreground: "hsl(210 40% 98%)",
26:         },
27:         muted: {
28:           DEFAULT: "hsl(210 40% 96.1%)",
29:           foreground: "hsl(215.4 16.3% 46.9%)",
30:         },
31:         accent: {
32:           DEFAULT: "hsl(210 40% 96.1%)",
33:           foreground: "hsl(222.2 47.4% 11.2%)",
34:         },
35:         card: {
36:           DEFAULT: "hsl(0 0% 100%)",
37:           foreground: "hsl(222.2 84% 4.9%)",
38:         },
39:       },
40:       borderRadius: {
41:         lg: "0.5rem",
42:         md: "calc(0.5rem - 2px)",
43:         sm: "calc(0.5rem - 4px)",
44:       },
45:     },
46:   },
47:   plugins: [],
48: }
```

■ File: vite.config.js

```
=====
1: import { defineConfig } from 'vite'
2: import react from '@vitejs/plugin-react'
3:
4: // https://vitejs.dev/config/
5: export default defineConfig({
6:   plugins: [react()],
7:   server: {
8:     port: 3000,
9:   },
10: })
```
