Textarea Component

The Textarea component provides a multi-line text input control. It supports various states, sizes, and styling options while maintaining accessibility and form integration capabilities.

Import

```
import { Textarea } from "@/components/ui/textarea"
```

API Reference

Props

- `error?: boolean` - Error state
- `disabled?: boolean` - Disabled state
- `className?: string` - Additional CSS classes
- All HTML textarea props are supported

Styling

Base Styles

```
.textarea {
  min-height: var(--textarea-min-height);
  padding: var(--space-2) var(--space-3);
  background-color: var(--color-surface);
  border: 1px solid var(--color-border);
  border-radius: var(--radius-md);
  color: var(--color-text-primary);
  transition: var(--textarea-transition);
}

.textarea:focus {
  outline: none;
  ring: 2px solid var(--color-primary);
  ring-offset: 2px;
}
```

States

Basic States

```
// Default
<Textarea placeholder="Enter text here..." />

// Disabled
<Textarea disabled placeholder="Disabled textarea" />

// With error
<Textarea error placeholder="Error state" />

// Read-only
<Textarea readOnly value="Read-only content" />
```

Examples

Basic Textarea

```
<div className="flex flex-col space-y-2">
  <label htmlFor="message">Message</label>
  <Textarea
    id="message"
    placeholder="Type your message here"
  />
</div>
```

With Character Count

```
function TextareaWithCounter() {
  const [value, setValue] = useState("")
  const maxLength = 100

  return (
    <div className="textarea-wrapper">
      <Textarea
        value={value}
        onChange={(e) => setValue(e.target.value)}
        maxLength={maxLength}
      />
      <span className="textarea-counter">
        {value.length}/{maxLength}
      </span>
    </div>
  )
}
```

With Helper Text

```
<div className="space-y-2">
  <label>Bio</label>
  <Textarea placeholder="Tell us about yourself" />
  <p className="text-sm text-foreground-muted">
    Write a brief description about yourself.
  </p>
</div>
```

Animations

Focus Animation

```
.textarea {
  transition: all var(--duration-fast) var(--ease-out);
}

@keyframes textarea-focus {
  from {
    box-shadow: 0 0 0 0 var(--color-primary);
  }
  to {
    box-shadow: 0 0 0 4px var(--color-primary-alpha);
  }
}
```

Accessibility

Keyboard Navigation

- `Tab`: Focus the textarea
- `Shift + Tab`: Move focus in reverse
- `Ctrl + A`: Select all text
- `Enter`: New line

ARIA Support

```
<div role="group" aria-labelledby="message-label">
  <label id="message-label">Message</label>
  <Textarea
    aria-describedby="message-hint"
    aria-invalid={error}
    aria-required="true"
  />
  <span id="message-hint">
    Enter your message here
```

```
  </span>
</div>
```

Integration

With Forms

```
import { useForm, Controller } from "react-hook-form"

function TextareaForm() {
  const form = useForm({
    defaultValues: {
      message: "",
    },
  })

  return (
    <form onSubmit={form.handleSubmit(onSubmit)}>
      <Controller
        control={form.control}
        name="message"
        render={({ field, fieldState }) => (
          <div className="space-y-2">
            <Textarea
              {...field}
              error={!!fieldState.error}
              placeholder="Enter message"
            />
            {fieldState.error && (
              <span className="textarea-error-text">
                {fieldState.error.message}
              </span>
            )}
          </div>
        )}
      />
    </form>
  )
}
```

Design Tokens

```
:root {
  /* Textarea dimensions */
  --textarea-min-height: 5rem;
  --textarea-max-height: 20rem;

  /* Textarea spacing */
```

```
  --textarea-padding-x: var(--space-3);
  --textarea-padding-y: var(--space-2);

  /* Textarea colors */
  --textarea-background: var(--color-surface);
  --textarea-border: var(--color-border);
  --textarea-text: var(--color-text-primary);
  --textarea-placeholder: var(--color-text-tertiary);

  /* Textarea states */
  --textarea-hover-border: var(--color-border-hover);
  --textarea-focus-ring: var(--color-primary);
  --textarea-disabled-bg: var(--color-surface-hover);
  --textarea-error-border: var(--color-destructive);
}
```

Troubleshooting

Common Issues

1. **Textarea not resizing**

```
// Solution: Add resize class
<Textarea className="resize-y" />
```

2. **Value not updating**

```
// Solution: Use controlled component
const [value, setValue] = useState("")

<Textarea
  value={value}
  onChange={(e) => setValue(e.target.value)}
/>
```

3. **Height issues**

```
// Solution: Set explicit min-height
<Textarea className="min-h-[200px]" />
```

Best Practices

1. **Label Usage**
    - Always provide labels

- Use semantic HTML
- Include helper text when needed

2. **Sizing**
   - Set appropriate min-height
   - Consider max-height for long content
   - Use resize controls wisely
3. **Error Handling**
   - Display clear error messages
   - Use visual error indicators
   - Provide validation feedback
4. **Accessibility**
   - Include proper ARIA labels
   - Support keyboard interactions
   - Maintain focus management

Variants

Size Variants

```
// Small
<Textarea className="textarea-sm" />

// Large
<Textarea className="textarea-lg" />
```

Resize Controls

```
// No resize
<Textarea className="textarea-no-resize" />

// Vertical only
<Textarea className="textarea-resize-y" />

// Horizontal only
<Textarea className="textarea-resize-x" />
```

With Loading State

```
<Textarea
  className="textarea-loading"
```

```
    disabled
    placeholder="Loading..."
  />
```