LoadingButton Component

A customized button component that shows a loading state with a spinner. Built on top of Material-UI's Button component.

Features

- Loading state indicator with spinner
- Configurable spinner position (start/end/center)
- RTL support
- Retains all Material-UI Button props
- TypeScript support

Usage

```
import { LoadingButton } from '@/components/core';

// Basic usage
<LoadingButton loading={isLoading}>
  حفظ
</LoadingButton>

// With custom spinner position
<LoadingButton
  loading={isLoading}
  loadingPosition="center"
  variant="contained"
  color="primary"
>
  إرسال
</LoadingButton>

// With progress size and icons
<LoadingButton
  loading={isLoading}
  loadingPosition="start"
  progressSize={16}
  startIcon={<SaveIcon />}
  variant="outlined"
>
  حفظ التغييرات
</LoadingButton>
```

Props

```tsx
interface LoadingButtonProps extends Omit<ButtonProps, 'loading'> {
  // Whether the button is in loading state
  loading?: boolean;

  // Size of the loading spinner in pixels
  progressSize?: number;

  // Position of the loading spinner
  loadingPosition?: 'start' | 'end' | 'center';
}
```

Additional Props

All Material-UI Button props are supported, including:

- `variant`: 'text' | 'outlined' | 'contained'
- `color`: 'primary' | 'secondary' | 'error' | etc.
- `size`: 'small' | 'medium' | 'large'
- `startIcon`: ReactNode
- `endIcon`: ReactNode
- `disabled`: boolean
- `fullWidth`: boolean
- `sx`: SxProps

Examples

Form Submit Button

```tsx
function SubmitForm() {
  const [loading, setLoading] = useState(false);

  const handleSubmit = async () => {
    setLoading(true);
    try {
      await submitData();
    } finally {
      setLoading(false);
    }
  };

  return (
    <LoadingButton
      loading={loading}
      loadingPosition="center"
      variant="contained"
      color="primary"
      onClick={handleSubmit}
      fullWidth
```

```
    >
      حفظ البيانات
    </LoadingButton>
  );
}
```

Delete Button with Confirmation

```
function DeleteButton() {
  const [loading, setLoading] = useState(false);

  const handleDelete = async () => {
    setLoading(true);
    try {
      await deleteItem();
    } finally {
      setLoading(false);
    }
  };

  return (
    <LoadingButton
      loading={loading}
      loadingPosition="start"
      variant="outlined"
      color="error"
      startIcon={<DeleteIcon />}
      onClick={handleDelete}
    >
      حذف
    </LoadingButton>
  );
}
```