# Coding Standards

## TypeScript Guidelines

1. **Strict mode enabled** - "strict": true in tsconfig.json
2. **No explicit any** - Use unknown if type is truly unknown
3. **Prefer interfaces over types** for object shapes
4. **Use type inference** when obvious
5. **Export types with implementations**

## Component Guidelines

1. **One component per file**
2. **Server Components by default** - Only add 'use client' when necessary
3. **Props interface named <ComponentName>Props**
4. **Destructure props in function signature**
5. **Use composition over inheritance**

**Example**:

// components/InitiativeCard.tsx  
interface InitiativeCardProps {  
 initiative: Initiative  
 onEdit?: (id: string) => void  
}  
  
export function InitiativeCard({ initiative, onEdit }: InitiativeCardProps) {  
 return (  
 <Card>  
 <CardHeader>  
 <CardTitle>{initiative.name}</CardTitle>  
 </CardHeader>  
 <CardContent>  
 <p>{initiative.description}</p>  
 </CardContent>  
 {onEdit && (  
 <CardFooter>  
 <Button onClick={() => onEdit(initiative.id)}>Edit</Button>  
 </CardFooter>  
 )}  
 </Card>  
 )  
}

## File Naming Conventions

* **Components**: PascalCase (InitiativeCard.tsx)
* **Utilities**: camelCase (formatCurrency.ts)
* **Types**: PascalCase (Initiative, User)
* **Constants**: SCREAMING\_SNAKE\_CASE (MAX\_INITIATIVES)
* **Routes**: kebab-case folders (/strategic-plans/)