```markdown
# Modern Project Creation

Create cutting-edge projects with Scale's intelligent project generation system.

## Overview

Scale's project creation feature allows you to instantly generate modern,
production-ready projects with best practices built-in. Choose from 6 different
project types, each optimized for specific use cases and automatically configured with
the latest tools and frameworks.

## Supported Project Types

### ⚛️ React Application
- **TypeScript** pre-configured
- **Vite** for lightning-fast development
- **ESLint** and **Prettier** for code quality
- **Tailwind CSS** for styling
- **React Router** for navigation
- **Testing** setup with Vitest

### 🚀 Next.js Application
- **Full-stack** React framework
- **TypeScript** support
- **App Router** (latest Next.js 14+)
- **Tailwind CSS** styling
- **API routes** ready
- **Deployment** optimized

### ⚡ Vite Application
- **Ultra-fast** development server
- **Hot Module Replacement** (HMR)
- **TypeScript** support
- **Multiple framework** options (React, Vue, Vanilla)
- **Build optimization**
- **Plugin ecosystem**

### 🟢 Node.js API
- **Express.js** server framework
- **TypeScript** configuration
- **RESTful API** structure
- **Middleware** setup
- **Environment** configuration
- **Testing** with Jest

### 🎨 Vue.js Application
- **Vue 3** with Composition API
- **TypeScript** support
- **Vite** build tool
- **Vue Router** for navigation
- **Pinia** for state management
- **Component** library ready

### 📱 React Native Application
- **Cross-platform** mobile development
- **TypeScript** pre-configured
- **Navigation** setup
- **State management** ready
- **Testing** environment
- **Build** scripts configured

## How to Create a Project

### Method 1: Quick Access (Recommended)
1. **Open Quick Access**: `Ctrl+Shift+P` → "Scale: Quick Access"
2. **Select "Create Project"**
3. **Choose your project type** from the list
4. **Enter project details**:
  - Project name
  - Directory location
  - Additional options
5. **Wait for generation** (usually 30-60 seconds)
6. **Open the new project** in VS Code

### Method 2: Command Palette
1. **Open Command Palette**: `Ctrl+Shift+P`
2. **Type**: "Scale: Create Project"
3. **Follow the same steps** as above

### Method 3: Sidebar
1. **Click the Scale icon** in the sidebar
2. **Click "Create Project"** button
3. **Follow the project creation wizard**
```

~~~markdown
## Project Creation Process

### Step 1: Project Type Selection
```
Choose from:
⚛️ React App - Modern React with TypeScript
🚀 Next.js App - Full-stack React framework
⚡ Vite App - Lightning-fast development
🟢 Node.js API - Express server with TypeScript
🎨 Vue.js App - Vue 3 with Composition API
📱 React Native App - Cross-platform mobile
```

### Step 2: Configuration
- **Project Name**: Enter a descriptive name
- **Location**: Choose where to create the project
- **Template Options**: Select additional features
- **Package Manager**: Choose npm, yarn, or pnpm

### Step 3: Generation
Scale will:
1. **Create** the project directory
2. **Generate** all necessary files
3. **Install** dependencies automatically
4. **Configure** development tools
5. **Set up** best practices

### Step 4: Ready to Code
- **Open** the project in VS Code
- **Start** the development server
- **Begin** coding immediately

## Project Templates

### React Application Structure
```
my-react-app/
├── public/
│   └── index.html
├── src/
│   ├── components/
│   ├── pages/
~~~

~~~
│   ├── hooks/
│   ├── utils/
│   ├── App.tsx
│   └── main.tsx
├── package.json
├── vite.config.ts
├── tsconfig.json
└── tailwind.config.js
```

### Next.js Application Structure
```
my-nextjs-app/
├── app/
│   ├── api/
│   ├── components/
│   ├── globals.css
│   ├── layout.tsx
│   └── page.tsx
├── public/
├── package.json
├── next.config.js
├── tsconfig.json
└── tailwind.config.js
```

### Node.js API Structure
```
my-api/
├── src/
│   ├── controllers/
│   ├── middleware/
│   ├── models/
│   ├── routes/
│   ├── utils/
│   └── server.ts
├── tests/
├── package.json
├── tsconfig.json
└── .env.example
```
~~~

```markdown
## Advanced Configuration

### Custom Templates
Scale allows you to:
- **Modify** existing templates
- **Add** custom configurations
- **Include** additional dependencies
- **Set up** specific tools

### Environment Setup
Each project includes:
- **Development** environment configuration
- **Production** build optimization
- **Testing** setup
- **Linting** and formatting rules
- **Git** initialization

## Best Practices

### Project Naming
- Use **kebab-case** for project names
- Keep names **descriptive** but concise
- Avoid **special characters** and spaces
- Consider **future scalability**

### Directory Organization
- Follow **framework conventions**
- Use **clear folder structures**
- Separate **concerns** appropriately
- Include **documentation** files

### Development Workflow
1. **Create** project with Scale
2. **Review** generated structure
3. **Customize** as needed
4. **Set up** version control
5. **Start** development

### Code Quality
- Use **TypeScript** for type safety
- Follow **ESLint** rules
- Format with **Prettier**
- Write **tests** from the start
- Document **complex logic**

## Troubleshooting

### Common Issues

#### Project Creation Fails
**Problem**: Error during project generation
**Solutions**:
- Check **internet connection**
- Verify **disk space** availability
- Ensure **write permissions** in target directory
- Try a **different project name**

#### Dependencies Not Installing
**Problem**: npm/yarn install fails
**Solutions**:
- Check **package manager** installation
- Clear **cache**: `npm cache clean --force`
- Try **different package manager**
- Check **network connectivity**

#### TypeScript Errors
**Problem**: TypeScript configuration issues
**Solutions**:
- Verify **tsconfig.json** is correct
- Check **TypeScript version** compatibility
- Restart **TypeScript server** in VS Code
- Update **@types** packages

#### Build Failures
**Problem**: Project won't build or start
**Solutions**:
- Check **Node.js version** compatibility
- Verify **all dependencies** are installed
- Review **configuration files**
- Check **console errors** for details

### Getting Help
- Use **Scale AI Chat** for instant help
- Check **project documentation**
```

~~~~markdown
- Review **framework-specific** guides
- Ask in **community forums**

## Examples

### Creating a React E-commerce App
```bash
# Using Scale Quick Access
1. Ctrl+Shift+P → "Scale: Quick Access"
2. Select "Create Project"
3. Choose "React App"
4. Name: "my-ecommerce-store"
5. Add: React Router, Tailwind CSS, Axios
6. Generate and start coding!
```

### Building a Next.js Blog
```bash
# Complete setup in minutes
1. Scale → Create Project → Next.js App
2. Name: "my-blog"
3. Features: App Router, Tailwind, MDX
4. Ready with: SEO, routing, styling
```

### Node.js API for Mobile App
```bash
# Backend API creation
1. Scale → Create Project → Node.js API
2. Name: "mobile-api"
3. Features: Express, TypeScript, JWT auth
4. Includes: CORS, validation, testing
```

## Related Documentation

- [Database Integration](./database-integration.md) - Connect your project to databases
- [AI Chat & Agent](./ai-chat-agent.md) - Get help during development
- [Terminal Integration](../workflow/terminal-integration.md) - Manage your project
- [Testing & Debugging](../workflow/testing-debugging.md) - Quality assurance
~~~~