Monaco Editor Integration - Verification Report

Status: ✅ COMPLETE

Monaco Editor has been successfully integrated into the Heaven UI with all requested features and custom theming.

Implementation Summary

1. Package Installation ✅

- **Package**: `@monaco-editor/react` (installed)
- **Location**: `node_modules/@monaco-editor/react`

2. CodeEditor Component ✅

Location: `/src/components/web/CodeEditor.tsx`

Features Implemented:
- ✅ Full TypeScript support with type definitions
- ✅ Multi-language support (typescript, javascript, react, html, css, json, markdown, python, rust, go, yaml)
- ✅ File path detection for automatic language selection
- ✅ Forward ref API for programmatic control
- ✅ Auto-save functionality with visual indicator
- ✅ Keyboard shortcuts (Cmd/Ctrl+S to save)
- ✅ Loading states with custom skeleton
- ✅ Read-only mode support
- ✅ File path and status indicators

TypeScript Interface:

```
interface CodeEditorProps {
  filePath?: string;
  content?: string;
  language?: string;
  onChange?: (value: string | undefined) => void;
  onSave?: (content: string) => void;
  readOnly?: boolean;
  loading?: boolean;
  height?: string | number;
  className?: string;
  showAutoSave?: boolean;
  autoSaveDelay?: number;
}
```

Exposed API Methods:

```
interface CodeEditorHandle {
  getValue: () => string | undefined;
  setValue: (value: string) => void;
  format: () => void;
  getEditor: () => editor.IStandaloneCodeEditor | null;
}
```

3. Custom Heaven Theme ✅

Location: `/src/config/monaco-theme.ts`

Theme Name: `heaven` (registered with Monaco)

Color Scheme (matches design specifications):
- Background: `#0A0E1A` ✅
- Keywords: `#C792EA` ✅
- Functions: `#82AAFF` ✅
- Strings: `#C3E88D` ✅
- Numbers: `#F78C6C` ✅
- Comments: `#546E7A` ✅
- Variables: `#EEFFFF` ✅
- Tags: `#F07178` ✅
- Types: `#FFCB6B` ✅
- Properties: `#89DDFF` ✅

UI Elements Themed:
- ✅ Editor background and foreground
- ✅ Line highlighting
- ✅ Selection colors
- ✅ Line numbers (normal and active)
- ✅ Cursor color
- ✅ Gutter (with git diff colors)
- ✅ Indentation guides
- ✅ Bracket matching
- ✅ Scrollbar styling
- ✅ Minimap colors
- ✅ Widget backgrounds (suggestions, hover, find)
- ✅ Peek view
- ✅ Diff editor

4. Editor Features ✅

Visual Features:
- ✅ Line numbers enabled
- ✅ Minimap enabled (right side, shows on mouseover)
- ✅ Syntax highlighting for all supported languages
- ✅ Code folding with indentation strategy
- ✅ Bracket matching
- ✅ Active line highlighting
- ✅ Indentation guides with active highlight

Code Features:
- ✅ Auto-closing brackets and quotes

- ✅ Auto-indent
- ✅ Format on paste and type
- ✅ IntelliSense/Suggestions
- ✅ Snippets support
- ✅ Multi-cursor support
- ✅ Find and replace
- ✅ Code folding

Editor Configuration:
- ✅ Tab size: 2 spaces
- ✅ Insert spaces (not tabs)
- ✅ Word wrap: off (configurable)
- ✅ Font: JetBrains Mono with ligatures
- ✅ Font size: 14px
- ✅ Line height: 22px
- ✅ Smooth scrolling
- ✅ Smooth cursor animation

Keyboard Shortcuts:
- ✅ Cmd/Ctrl+S: Save file
- ✅ Cmd/Ctrl+P: Command palette (via MainLayout)
- ✅ All standard Monaco shortcuts (find, replace, etc.)

TypeScript/JavaScript Configuration:
- ✅ Semantic validation enabled
- ✅ Syntax validation enabled
- ✅ JSX/React support configured
- ✅ ESModule interop enabled
- ✅ Modern target (ESNext)
- ✅ Node.js module resolution

5. Demo Page ✅

Location: `/src/pages/CodeEditorDemo.tsx`

Features Demonstrated:
- ✅ File list sidebar (toggle with Cmd+B)
- ✅ Multiple file switching
- ✅ Command palette integration
- ✅ Auto-save indicator
- ✅ Status bar with file info
- ✅ Read-only mode
- ✅ Format document command

Sample Files Included:
1. `Button.tsx` - TypeScript React component
2. `helpers.ts` - TypeScript utility functions
3. `README.md` - Markdown documentation

6. Integration with Heaven UI ✅

Used In:
- ✅ `MainLayout.tsx` - Center panel with full editor
- ✅ `CodeEditorDemo.tsx` - Standalone demo page

Theme Consistency:
- ✅ Matches Heaven UI color palette
- ✅ Consistent with design system
- ✅ Dark theme optimized for deep space aesthetic

Verification Tests

TypeScript Type Checking ✅

```
npm run type-check
```

Result: ✅ PASSED - No TypeScript errors

File Structure ✅

```
src/
├── components/
│   └── web/
│       └── CodeEditor.tsx          ✅ Main component
├── config/
│   └── monaco-theme.ts             ✅ Theme configuration
└── pages/
    └── CodeEditorDemo.tsx          ✅ Demo page
```

Language Support ✅

Tested and verified support for:
- ✅ TypeScript (`.ts` , `.tsx`)
- ✅ JavaScript (`.js` , `.jsx`)
- ✅ JSON (`.json`)
- ✅ HTML (`.html`)
- ✅ CSS (`.css` , `.scss`)
- ✅ Markdown (`.md`)
- ✅ Python (`.py`)
- ✅ Rust (`.rs`)
- ✅ Go (`.go`)
- ✅ YAML (`.yml` , `.yaml`)
- ✅ And more...

Usage Example

```
import { CodeEditor, CodeEditorHandle } from '@/components/web/CodeEditor';
import { useRef, useState } from 'react';

function MyComponent() {
  const editorRef = useRef<CodeEditorHandle>(null);
  const [content, setContent] = useState('// Write your code here');

  const handleSave = (value: string) => {
    console.log('Saving:', value);
    // Save logic here
  };

  const formatCode = () => {
    editorRef.current?.format();
  };

  return (
    <div className="h-screen">
      <CodeEditor
        ref={editorRef}
        filePath="/src/example.ts"
        content={content}
        onChange={(value) => setContent(value || '')}
        onSave={handleSave}
        showAutoSave={true}
        autoSaveDelay={2000}
      />
      <button onClick={formatCode}>Format</button>
    </div>
  );
}
```

Next Steps (Optional Enhancements)

While Monaco Editor is fully integrated and functional, here are some optional enhancements that could be added in the future:

1. **Custom Language Definitions**: Add custom language support for domain-specific languages
2. **AI-Powered Completions**: Integrate AI suggestions into IntelliSense
3. **Collaborative Editing**: Add real-time collaboration features
4. **Git Integration**: Show inline git blame and diff markers
5. **Advanced Debugging**: Add breakpoint and debugging UI
6. **Custom Themes**: Allow users to customize editor themes
7. **Extensions**: Add Monaco extension support
8. **Performance Monitoring**: Add editor performance metrics

Conclusion

✅ **Monaco Editor integration is COMPLETE and PRODUCTION-READY**

All requested features have been implemented:
- ✅ Monaco Editor package installed

- ✅ CodeEditor component created with full TypeScript support
- ✅ Custom Heaven-dark theme configured with exact color specifications
- ✅ All editor features implemented (line numbers, minimap, shortcuts, auto-save)
- ✅ Multi-language support
- ✅ TypeScript type checking passes
- ✅ Demo page available

The editor is ready for use in the Heaven UI and can be accessed via:
- Component import: `import { CodeEditor } from '@/components/web/CodeEditor';`
- Demo page: Navigate to `/demo/code-editor` route

Generated: October 20, 2025
Status: ✅ COMPLETE
Version: v1.0.0