

Style Analysis Workflow - Automatic Fix Application

Updates Made

1. Fixed Shebang Issue in ai-style-fixer.js

Problem: The fixer script had a blank line before the shebang (`#!/usr/bin/env node`), causing a syntax error when executed.

Error:

```
SyntaxError: Invalid or unexpected token
    at Module._compile (node:internal/modules/cjs/loader:1328:27)
```

Solution: Removed the blank line so the shebang is on line 1, allowing the script to execute properly.

2. Enhanced Run Script with Automatic Fix Prompt

Updated: `scripts/run-style-analysis.sh`

New Feature: After running the analysis (option 1), the script now automatically:

1. Detects if analysis completed successfully
2. Finds the latest generated report
3. Prompts the user: "Would you like to apply fixes automatically? (y/n)"
4. If yes, immediately runs the fixer with the latest report
5. If no, provides instructions for running fixes later

3. Automatic GitHub Push (NEW!)

Updated: `scripts/ai-style-fixer.js`

New Feature: After applying fixes, the script now prompts:



```
Would you like to commit and push these changes to GitHub? (y/n):
```

If you choose 'y', it automatically:

1. Stages all changed files (`git add -A`)
2. Creates a descriptive commit message with:
 - Number of fixes applied
 - Number of files modified
 - Report filename used
 - Backup location
3. Commits the changes
4. Pushes to GitHub (`git push origin main`)

Benefits:

- ☒ No need to manually commit and push
- ☒ Changes are immediately saved to GitHub

-  Next time you pull, you get the fixed files
-  No more re-running the same fixes after updates

4. Improved User Experience

Before:

User runs analysis → Manual copy/paste of report path → Run fixer separately → Manual commit/push

After:

User runs analysis → Auto prompt to fix → Auto prompt to push → Done!




Usage

Quick Start (Recommended)

```
cd ~/Sports-Bar-TV-Controller
./scripts/run-style-analysis.sh
# Choose option 1 to analyze
# When prompted, press 'y' to automatically apply fixes
```

Analysis Report from Previous Run

The latest analysis found:

- **Total files analyzed:** 90
- **Files with issues:** 57
- **Total issues:** 596
-  High priority: 18
-  Medium priority: 13
-  Low priority: 26

Fixer Script Modes

When you run the fixer (either automatically or via option 2), you can choose:

1. **Interactive Mode** - Review and approve each file individually
 - Shows issues for each file
 - Ask y/n for each file
 - Press 'q' to quit at any time
2. **Auto-fix All** - Apply all fixes automatically
 - Processes all files without prompts
 - Creates backups before modifying
 - Fast for bulk fixes
3. **Review Only** - View issues without making changes
 - Shows what would be fixed
 - No files are modified
 - Good for understanding the scope

Safety Features

Automatic Backups: Before modifying any file, the fixer creates a backup in `ai-style-backups/` with timestamp:

```
ai-style-backups/
├── page.tsx.2025-10-01T04-55-00-000Z.bak
├── SportsGuide.tsx.2025-10-01T04-55-01-000Z.bak
└── ...
```

Restore from Backup: If something goes wrong:






```
# Find your backup
ls -lt ai-style-backups/

# Restore a file
cp ai-style-backups/ComponentName.tsx.2025-10-01T04-55-00-000Z.bak src/app/component/
ComponentName.tsx
```






What the Fixer Does

The fixer script automatically corrects:

Color Inconsistencies

-  `bg-blue-900` →  `bg-slate-900`
-  `text-blue-600` →  `text-slate-300`
-  `border-gray-700` →  `border-slate-700`






Background Colors

-  `bg-gray-800` →  `bg-slate-900`
-  `bg-black` →  `bg-slate-950`
-  `from-gray-900` →  `from-slate-900`

Text Colors

-  `text-white` →  `text-slate-100`
-  `text-gray-400` →  `text-slate-400`
-  `text-blue-400` →  `text-slate-300`

Borders & Accents

-  `border-gray-600` →  `border-slate-700`
-  `border-blue-500` →  `border-slate-600`
-  `ring-blue-500` →  `ring-slate-500`

Hover & Active States

-  `hover:bg-gray-700` →  `hover:bg-slate-800`
-  `hover:text-white` →  `hover:text-slate-100`
-  `active:bg-blue-900` →  `active:bg-slate-900`

Complete Workflow Example

```
# 1. Navigate to project
cd ~/Sports-Bar-TV-Controller

# 2. Run the style analysis tool
./scripts/run-style-analysis.sh

# 3. Choose option 1 (Analyze all components)
Enter your choice (1-5): 1

# Wait for analysis to complete...
# It will analyze 90 components and generate a report


# 4. When prompted, choose to apply fixes
Would you like to apply fixes automatically? (y/n): y

# 5. Choose a mode
Select mode:
  1. Interactive (review each file)
  2. Auto-fix all (apply all fixes automatically)
  3. Review only (show issues without fixing)

Enter choice (1-3): 2





# 6. Fixer runs and applies changes
# Backups are created automatically
# Summary shows how many fixes were applied

# 7. NEW! Automatic GitHub push prompt
Would you like to commit and push these changes to GitHub? (y/n): y

# If yes, it automatically:
# - Stages all changes (git add -A)
# - Creates a descriptive commit message
# - Commits the changes
# - Pushes to GitHub
# Done! 

# 8. Next time you run update_from_github.sh, you'll get these fixes!
```



Current Status

-  **Fixed:** Shebang syntax error in ai-style-fixer.js
-  **Enhanced:** Automatic fix prompt after analysis
-  **Improved:** User workflow streamlined
-  **Ready:** All scripts tested and working

Next Steps

1. **Run the analysis** to generate a fresh report
2. **Apply fixes** using auto-fix mode (recommended for bulk changes)
3. **Test the application** to ensure everything works
4. **Review the changes** in version control
5. **Commit** the standardized code to GitHub

Files Modified

-  `scripts/ai-style-fixer.js` - Fixed shebang position
 -  `scripts/run-style-analysis.sh` - Added automatic fix prompt
-

Date: October 1, 2025

Status:  Ready to use

Analysis Status: 596 issues found across 57 files