Contributing to Fausse Nostalgie

Thank you for considering contributing to this project! This was my first Python project, and I welcome contributions from developers of all skill levels.

🎯 Areas Where Help is Needed

High Priority

1. **Windows compatibility** - The app currently uses Unix-specific modules (`termios`, `tty`). A Windows-compatible version would be brilliant.
2. **macOS testing** - I don't have access to macOS. Bug reports and fixes are highly appreciated.
3. **Performance optimisations** - The rendering could potentially be more efficient.

Medium Priority

4. **Additional colour schemes** - Creative retro colour palettes welcome!
5. **More timezone options** - Suggestions for better default timezones
6. **Documentation improvements** - Clearer explanations, better examples

Low Priority

7. **New features** - As long as they maintain the retro aesthetic
8. **Code refactoring** - Making the code more Pythonic whilst maintaining readability

🛠️ How to Contribute

Reporting Bugs

1. **Check existing issues** - Someone may have already reported it
2. **Provide details:**
    - Operating system and version
    - Python version (`python --version`)
    - Terminal emulator (Konsole, iTerm2, Windows Terminal, etc.)
    - Steps to reproduce
    - Expected vs actual behaviour
    - Screenshots if applicable

Suggesting Features

1. **Open an issue** with the "enhancement" label
2. **Describe the feature** - What it does and why it's useful
3. **Maintain the aesthetic** - New features should fit the retro IBM terminal theme
4. **Consider complexity** - This is meant to be a simple, focused application

Submitting Code

1. Fork and Clone

```bash
git clone https://github.com/yourusername/fausse-nostalgie.git
cd fausse-nostalgie
```

2. Create a Branch

```bash
git checkout -b feature/your-feature-name
```

Use descriptive branch names:

- `feature/add-custom-fonts`
- `fix/calendar-alignment-bug`
- `docs/improve-installation-guide`

3. Make Your Changes

Code Style Guidelines:

- Follow PEP 8 where sensible
- Maintain the existing code structure
- Add comments for complex logic
- Keep functions focused and readable
- Use descriptive variable names

Testing:

- Test on your target platform thoroughly
- Try different terminal sizes
- Test all features (clock, calendar, timezones, colours)
- Ensure no regressions (existing features still work)

4. Commit Your Changes

```bash
git add .
git commit -m "Add feature: brief description"
```

Commit Message Guidelines:

- Use present tense ("Add feature" not "Added feature")
- Be concise but descriptive
- Reference issues if applicable (#42)

Examples:

```
Add purple colour scheme to palette options
Fix calendar overflow bug for December 2026
Update README with Fedora installation instructions
```

5. Push and Create Pull Request

```bash
git push origin feature/your-feature-name
```

Then create a Pull Request on GitHub with:

- **Clear title** describing the change
- **Description** explaining what and why
- **Screenshots** if UI changes
- **Testing notes** - what you tested

📝 Code Review Process

1. I'll review PRs as soon as I can (usually within a week)
2. I might request changes or ask questions
3. Once approved, I'll merge it
4. Your contribution will be credited in the README

🤝 Code of Conduct

Be Respectful

- This is a learning project
- Everyone starts somewhere
- Constructive criticism only
- Assume good intentions

Be Patient

- I'm learning too
- Responses might take a few days
- Complex PRs need thorough review

Be Clear

- Explain your reasoning
- Provide context
- Ask questions if unclear

🎓 First Time Contributing?

That's fantastic! Here are some beginner-friendly tasks:

- **Documentation:** Fix typos, improve clarity, add examples
- **Testing:** Try the app on different systems and report issues
- **Colour schemes:** Add new palette options (just RGB values!)
- **Timezone suggestions:** Propose better default cities

Don't be intimidated by the code. Start small and ask questions!

📚 Resources

- Python PEP 8 Style Guide
- How to Write a Git Commit Message
- GitHub Flow Guide
- ANSI Escape Codes Reference

💬 Questions?

Feel free to:

- Open an issue with the "question" label
- Comment on existing issues
- Reach out directly

Thank you for contributing to Fausse Nostalgie! 🎉