

Minimalist Sidebar Sync Summary

Changes Made

1. Vue/Vite Variant Sync (minimalist_sidebar_vue_vite)

New files created:

- `src/data/sidebar-data.ts` - Ported sidebar navigation data from React
- `src/services/vibe-kanban.ts` - Vibe Kanban API client (ported from React)
- `src/composables/useVibeKanban.ts` - Vue 3 composable for Vibe Kanban
- `src/composables/useSidebar.ts` - Vue 3 composable for sidebar state
- `src/components/Sidebar.vue` - Main sidebar component
- `src/App.vue` - Root Vue application component
- `src/main.ts` - Vue application entry point
- `tsconfig.json` - TypeScript configuration with path aliases
- `tsconfig.node.json` - Node-specific TypeScript config
- `env.d.ts` - TypeScript declarations for Vite env

Updated files:

- `vite.config.ts` - Added Vue plugin
- `package.json` - Added Vue dependencies
- `index.html` - Updated entry point

2. Shared Theme (frontend/shared)

New files:

- `theme/colors.ts` - Shared color constants for both React and Vue

3. Asset Sync

- Copied assets from `frontend/assets/` to Vue variant
- Copied brand images to Vue public folder

4. Vibe Kanban Integration

Both React and Vue variants now have:

- API client for Vibe Kanban backend (port 3001)
- WebSocket support for real-time updates
- Dashboard opening functionality
- Connection status indicators

Integration points:

- React: `hooks/use-vibe-kanban.ts` , `services/vibe-kanban.ts`
- Vue: `composables/useVibeKanban.ts` , `services/vibe-kanban.ts`

5. LazyVim/Neovim Configuration

Created at `config/nvim/` :

- `init.lua` - LazyVim bootstrap with project-specific extras
- `lua/config/keymaps.lua` - Project navigation keymaps
- `lua/config/options.lua` - Editor options

- `lua/plugins/ripple-env.lua` - Plugin configuration
- `README.md` - Installation and usage guide

Features:

- TypeScript/JavaScript tooling (typescript-tools.nvim)
- Vue support (Volar)
- Tailwind CSS IntelliSense
- Prettier/ESLint formatting
- LazyGit integration
- Project-specific keymaps for navigating frontend directories
- Pixi-integrated terminal commands

6. Pixi Package Management

Verified `pixi.toml` already has:

- `nodejs = ">=22.0,<23"` (LTS "Jod")
- `pnpm = "==10.27.0"`
- Frontend tasks: `frontend-install`, `frontend-dev`, `frontend-build`

Directory Structure

```

ripple-env/
├── frontend/
│   ├── assets/                # Brand assets (source of truth)
│   ├── shared/
│   │   ├── theme/colors.ts    # NEW: Shared theme
│   │   ├── minimalist_sidebar_react/ # React variant (source)
│   │   │   ├── hooks/vibe-kanban/ # Vibe Kanban hooks
│   │   │   ├── services/vibe-kanban.ts
│   │   │   └── ...
│   │   └── minimalist_sidebar_vue_vite/ # Vue variant (synced)
│   │       ├── src/
│   │       │   ├── composables/ # NEW: Vue composables
│   │       │   ├── data/        # NEW: Sidebar data
│   │       │   ├── services/    # NEW: Vibe Kanban service
│   │       │   ├── components/  # NEW: Sidebar.vue
│   │       │   └── App.vue      # NEW: Root component
│   │       └── ...
│   └── vibe-kanban/           # EXISTING (not modified)
├── config/
│   └── nvim/                  # NEW: LazyVim config
│       ├── init.lua
│       └── lua/
└── pixi.toml                  # Node.js via pixi (verified)

```

Usage

Running the Vue Sidebar

```

cd frontend/minimalist_sidebar_vue_vite
pixi run -- pnpm install
pixi run -- pnpm dev

```

Using LazyVim Config

```
# Option 1: Symlink
ln -sf /path/to/ripple-env/config/nvim ~/.config/nvim

# Option 2: Named config
NVIM_APPNAME=ripple-nvim nvim
```

Vibe Kanban Integration

The sidebar connects to Vibe Kanban at:

- Backend API: `http://localhost:3001`
- Frontend: `http://localhost:3000`

Configure via environment variables:

- `VITE_VIBE_KANBAN_API_URL`
- `VITE_VIBE_KANBAN_FRONTEND_URL`

Next Steps

1. Test Vue sidebar with `pixi run -- pnpm dev` in `minimalist_sidebar_vue_vite`
2. Start Vibe Kanban backend to test integration
3. Configure LazyVim for development workflow
4. Sync additional components as needed (KanbanBoard, TaskCard, etc.)