Excel Editor Pro - Professional Data Management System

A modern, professional Excel file editor with smart suggestions, beautiful Japanese-inspired UI, and powerful data management features.

✨ Features

- 📊 **Smart Suggestions**: Backend + algorithm-based suggestions for faster data entry
- 🎨 **Beautiful UI**: Japanese-inspired design with glass effects and smooth animations
- 🔄 **Dual View Modes**: Table view and Card view for different editing needs
- 📝 **Multi-Datatype Support**: Automatic detection for numbers, dates, and text
- 🔍 **Advanced Filtering**: Multi-column filters and global search
- ↩️ **Undo/Redo**: Full history tracking with keyboard shortcuts
- 💾 **Auto-save**: Automatic saving to localStorage
- 📱 **Responsive**: Works on desktop, tablet, and mobile
- ⌨️ **Keyboard Navigation**: Full keyboard support for suggestions and navigation

🚀 Quick Start

Development

```
bash

# Install dependencies
npm install

# Run development server
npm run dev

# In another terminal, run the API server (optional for suggestions)
npm start
```

Visit `http://localhost:5173`

Production Build

```
bash
```

```
# Build for production
npm run build

# Start production server
npm start
```

🚂 Railway Deployment

Step 1: Push to GitHub

```
bash

git init
git add .
git commit -m "Initial commit"
git remote add origin YOUR_GITHUB_REPO_URL
git push -u origin main
```

Step 2: Deploy on Railway

1. Go to Railway.app
2. Click "New Project" → "Deploy from GitHub repo"
3. Select your repository
4. Railway will auto-detect the Node.js app
5. Set the following:
   - **Build Command**: `npm install && npm run build`
   - **Start Command**: `npm start`
   - **Port**: Railway auto-assigns (no change needed)

Step 3: Verify Deployment

- Railway will provide a URL like `your-app.railway.app`
- Visit the URL to see your app live!

🔧 Configuration

Environment Variables (Optional)

Create a `.env` file for custom configuration:

```
env
```

```
PORT=3000
NODE_ENV=production
```

Customizing Suggestions

Edit `data/suggestions.json` to add backend suggestions for specific columns.

📋 Checklist After Deployment

- ☐ Verify the app loads without blank page
- ☐ Test file upload functionality
- ☐ Check that suggestions work (both backend and common values)
- ☐ Test undo/redo
- ☐ Verify auto-save
- ☐ Test on mobile device
- ☐ Check browser console for errors

🐛 Troubleshooting

Blank Page Issues

1. **Check Console**: Open browser DevTools → Console tab
   - Look for 404 errors on chunk files
   - Check for JavaScript errors
2. **Check Network Tab**:
   - Verify all assets load from correct paths
   - Ensure server is serving `index.html` for all routes
3. **Verify Build**:

```bash
npm run build
ls dist/  # Should show index.html and assets/
```

4. **Check Server Logs**:
   - In Railway dashboard, check application logs
   - Look for startup messages

Common Issues

Problem: 404 on chunk files **Solution**: Ensure `vite.config.js` has `base: '/'` and server has SPA fallback

Problem: Suggestions not working **Solution**: Check if `data/suggestions.json` exists and server can read it

Problem: Auto-save not persisting **Solution**: localStorage may be disabled; data persists per-session only

Problem: Service worker caching old version **Solution**: Already handled - service workers are unregistered on load

🎨 Color Palette

The app uses a neutral, warm color scheme:

- **Sage**: `#6B705C` - Primary text and accents
- **Stone**: `#B7B7A4` - Secondary elements
- **Terracotta**: `#C5866D` - Primary buttons and highlights
- **Sand**: `#DDBEA9` - Gradients and backgrounds
- **Cream**: `#F1E3D3` - Borders and subtle backgrounds
- **Ivory**: `#FFF6EB` - Main background

📦 Project Structure

```
excel-editor-pro/
├── src/
│   ├── components/
│   │   ├── SuggestionInput.jsx    # Smart input with suggestions
│   │   ├── Toast.jsx              # Toast notifications
│   │   └── JapaneseOrnament.jsx   # Decorative background
│   ├── App.jsx                    # Main application
│   ├── main.jsx                   # Entry point
│   └── index.css                  # Global styles
├── data/
│   └── suggestions.json           # Backend suggestions
├── server.js                      # Express server
├── vite.config.js                 # Vite configuration
├── tailwind.config.js             # Tailwind configuration
└── package.json                   # Dependencies
```

🎯 Keyboard Shortcuts

- Ctrl/Cmd + Z: Undo
- Ctrl/Cmd + Y or Ctrl/Cmd + Shift + Z: Redo
- Ctrl/Cmd + S: Manual save
- Arrow Keys: Navigate suggestions dropdown
- Enter: Select suggestion
- Escape: Close suggestion dropdown
- Left/Right Arrow (Card view): Navigate between cards

📱 Mobile Support

The app is fully responsive and includes:

- Touch-friendly buttons and inputs
- Optimized card view for mobile
- Responsive table with horizontal scroll
- Mobile-optimized suggestion dropdown

🔒 Data Privacy

- All data is stored locally in browser localStorage
- No data is sent to external servers
- File processing happens client-side
- Suggestions API only processes column metadata

📄 License

MIT License - Feel free to use for personal or commercial projects

🤝 Contributing

Contributions welcome! Please open an issue or PR.

Built with ❤️ using React, Vite, Tailwind CSS, and Framer Motion