Financial Management & Simulation Tool

A comprehensive personal finance tracking and planning application with 2-year projection capabilities.

Features

Core Functionality

- **Dashboard**: Real-time financial overview with alerts and status indicators
- **Transaction Management**: Add, edit, import, and auto-categorize transactions
- **Budget Planning**: 2-year budget planning (Aug 2025 - Jul 2027) with templates
- **Category Analysis**: Deep spending insights with trends and patterns
- **What-If Simulator**: Scenario planning and impact analysis
- **Reports**: Multi-format export with professional reporting

Key Capabilities

- ✅ CSV/Excel import with intelligent auto-categorization
- ✅ Configurable alert thresholds (default 10%)
- ✅ 24-month planning horizon
- ✅ Multiple budget templates (Conservative, Moderate, Aggressive)
- ✅ Spending pattern analysis and anomaly detection
- ✅ Data backup and recovery
- ✅ Professional charts and visualizations (optional)

Installation

Prerequisites

- Python 3.8 or higher
- Tkinter (usually included with Python)

Setup

1. Clone the repository:

```bash
git clone https://github.com/yourusername/financial-management-tool.git
cd financial-management-tool
```

2. Install dependencies:

```bash
pip install -r requirements.txt
```

3. Run the application:

```bash
python main.py
```

Project Structure

```
financial_management_tool/
├── main.py          # Application entry point
├── config.py        # Configuration and constants
├── models/          # Data models
├── managers/        # Business logic
├── gui/             # User interface
│   ├── tabs/        # Main tab interfaces
│   └── dialogs/     # Dialog windows
├── utils/           # Utility functions
└── data/            # Data storage (created at runtime)
```

Usage

First Time Setup

1. Launch the application
2. Navigate to Budget Planning tab
3. Apply a budget template or set custom budgets
4. Start adding transactions manually or import from CSV

Importing Bank Statements

1. Go to Transactions tab
2. Click "Import CSV"
3. Select your bank statement file
4. Review auto-categorization
5. Confirm import

Creating What-If Scenarios

1. Navigate to What-If Simulator tab
2. Click "Create New Scenario"
3. Define scenario parameters
4. Run simulation to see impact
5. Save scenario for future reference

Category Structure

Loans & EMIs

- Credit Card EMI 1 & 2
- Personal Loan EMI 1 & 2
- Home Loan EMI

Investments

- Mutual Fund SIP
- PPF, RD
- Ponmagan Policy
- Gold & Bitcoin Investment
- Baby Health & Education Policy

Lifestyle & Essentials

- OTT Subscriptions
- Hospital (Medical)
- Swiggy/Food
- Petrol
- General Expenses
- Shopping

Configuration

Alert Thresholds

Default alert threshold is 10%. Configure in Settings or per category.

Budget Templates

- **Conservative**: Lower spending, higher savings
- **Moderate**: Balanced approach
- **Aggressive**: Higher spending allowances

Data Management

Backup

- Automatic daily backups (last 7 days retained)
- Manual backup available anytime
- Restore from any backup point

Export Formats

- JSON: Complete data backup
- Excel: Formatted spreadsheets with charts
- CSV: Simple data format
- PDF: Professional reports

Keyboard Shortcuts

- `Ctrl+N`: New transaction
- `Ctrl+I`: Import transactions
- `Ctrl+S`: Manual save
- `Ctrl+E`: Export current view
- `F5`: Refresh current tab
- `Ctrl+Tab`: Switch tabs

Troubleshooting

Common Issues

1. **Application won't start**: Check Python version (3.8+)
2. **Import errors**: Ensure CSV has Date, Amount, Description columns
3. **Charts not showing**: Install matplotlib (`pip install matplotlib`)
4. **Data not saving**: Check write permissions in application directory

Contributing

Pull requests are welcome. For major changes, please open an issue first.

License

Support

For issues or questions, please create an issue on GitHub.

Version History

- v2.0.0 - Complete rewrite with enhanced features
- v1.0.0 - Initial release

Author

Financial Management Team

Built with Python and Tkinter for reliable desktop financial management