

FinPlanner - Complete Download & Setup Guide with Form Analysis

Download Instructions

Method 1: Git Clone (Recommended)

Prerequisites

- Git installed on your system
- Node.js 18.0.0 or higher
- PostgreSQL 13 or higher
- npm 8.0.0 or higher

Step-by-Step Instructions

```
# 1. Clone the repository
git clone https://github.com/venkatkrish78/finplanner.git
cd finplanner

# 2. Navigate to the app directory
cd app

# 3. Install dependencies
npm install

# 4. Set up environment variables
cp ../.env.local.example .env.local
# Edit .env.local with your database configuration

# 5. Set up PostgreSQL database
createdb finplanner_dev

# 6. Run database migrations
npm run db:migrate

# 7. Seed the database with sample data
npm run db:seed

# 8. Start the development server
npm run dev
```

Method 2: Download ZIP File

1. **Visit GitHub Repository:** Go to <https://github.com/venkatkrish78/finplanner.git>
2. **Download ZIP:** Click the green “Code” button → “Download ZIP”
3. **Extract Files:** Extract the downloaded ZIP file to your desired location
4. **Follow Setup:** Continue with steps 2-8 from Method 1 above

Method 3: GitHub Desktop

1. **Install GitHub Desktop:** Download from <https://desktop.github.com/>

2. Clone Repository:

- Open GitHub Desktop
- Click "Clone a repository from the Internet"
- Enter URL: `https://github.com/venkatkrish78/finplanner.git`
- Choose local path and click "Clone"

3. **Follow Setup:** Continue with steps 2-8 from Method 1 above

⚙️ Development Setup Requirements

Required Software

- **Node.js:** v18.0.0 or higher ([Download](https://nodejs.org/) (https://nodejs.org/))
- **npm:** v8.0.0 or higher (comes with Node.js)
- **PostgreSQL:** v13 or higher ([Download](https://www.postgresql.org/download/) (https://www.postgresql.org/download/))
- **Git:** Latest version ([Download](https://git-scm.com/) (https://git-scm.com/))

Environment Configuration

Required Environment Variables (.env.local)

```
# Database Configuration
DATABASE_URL="postgresql://postgres:password@localhost:5432/finplanner_dev"

# NextJS Configuration
NEXTAUTH_URL="http://localhost:3000"
NEXTAUTH_SECRET="local-development-secret-key-change-in-production"

# Application Configuration
NODE_ENV="development"
PORT=3000

# Development Settings
NEXT_TELEMETRY_DISABLED=1
```

Database Setup

Create PostgreSQL Database

```
# Start PostgreSQL service
sudo systemctl start postgresql # Linux
brew services start postgresql  # macOS

# Create database
createdb finplanner_dev

# Create user (optional)
sudo -u postgres psql
CREATE USER finplanner WITH PASSWORD 'your_password';
GRANT ALL PRIVILEGES ON DATABASE finplanner_dev TO finplanner;
\q
```

Build and Run Commands

```
# Development
npm run dev           # Start development server
npm run build         # Build for production
npm run start         # Start production server

# Database Management
npm run db:migrate    # Run database migrations
npm run db:seed       # Seed database with sample data
npm run db:studio     # Open Prisma Studio
npm run db:generate   # Generate Prisma client

# Code Quality
npm run lint          # Run ESLint
npm run lint:fix      # Fix ESLint issues
npm run type-check    # TypeScript type checking

# Docker (Optional)
npm run docker:up     # Start with Docker Compose
npm run docker:down   # Stop Docker services
```

Complete Form Inventory

Summary Statistics

- **Total Dialog Forms:** 18 components
- **Total Modal Forms:** 3 components
- **Total Calculator Forms:** 3 components
- **Total Add/Edit/Delete Forms:** 14 components
- **Total Interactive Components:** 38+ forms

Form Categories by Functionality

1. Transaction Management Forms (8 forms)

Form Name	Type	Purpose	Location
Add Transaction Dialog	Create	Add new income/ex-pense transactions	add-transaction-dialog.tsx
Edit Transaction Dialog	Edit	Modify existing transactions	edit-transaction-dialog.tsx
Transaction Detail Modal	View	Display detailed transaction information	transaction-detail-modal.tsx
Transaction Filters	Filter	Filter and search transactions	transaction-filters.tsx
SMS Parser Dialog	Import	Parse transactions from SMS messages	sms-parser-dialog.tsx
Enhanced Transaction List	List/Bulk	Manage multiple transactions	enhanced-transaction-list.tsx
Transaction View Selector	View	Switch between different transaction views	transaction-view-selector.tsx
Transaction Summary Cards	Dashboard	Display transaction summaries	transaction-summary-cards.tsx

2. Bill Management Forms (4 forms)

Form Name	Type	Purpose	Location
Add Bill Dialog	Create	Create new recurring bills	add-bill-dialog.tsx
Bill Detail Modal	View	View detailed bill information	bill-detail-modal.tsx
Bill Calendar	Schedule	Manage bill due dates	bill-calendar.tsx
Bill Reminders	Notification	Set up bill payment reminders	bill-reminders.tsx

3. Financial Goals Forms (5 forms)

Form Name	Type	Purpose	Location
Add Goal Dialog	Create	Create new financial goals	add-goal-dialog.tsx
Edit Goal Dialog	Edit	Modify existing goals	edit-goal-dialog.tsx
Add Goal Contribution Dialog	Create	Add contributions to goals	add-goal-contribution-dialog.tsx
Goal Calculator	Calculator	Calculate goal requirements	goal-calculator.tsx
Goal List	List	Manage multiple goals	goal-list.tsx

4. Investment Management Forms (8 forms)

Form Name	Type	Purpose	Location
Add Investment Dialog	Create	Add new investments	add-investment-dialog.tsx
Edit Investment Dialog	Edit	Modify investment details	edit-investment-dialog.tsx
Delete Investment Dialog	Delete	Remove investments	delete-investment-dialog.tsx
Add SIP Dialog	Create	Set up Systematic Investment Plans	add-sip-dialog.tsx
Link Investment Dialog	Link	Link investments to goals	link-investment-dialog.tsx
Investment Calculators	Calculator	Various investment calculations	investment-calculators.tsx
SIP Management	Manage	Manage SIP investments	sip-management.tsx
Investment List	List	View and manage investments	investment-list.tsx

5. Loan Management Forms (6 forms)

Form Name	Type	Purpose	Location
Add Loan Dialog	Create	Add new loans	add-loan-dialog.tsx
Edit Loan Dialog	Edit	Modify loan details	edit-loan-dialog.tsx
Delete Loan Dialog	Delete	Remove loans	delete-loan-dialog.tsx
Add Loan Payment Dialog	Create	Record loan payments	add-loan-payment-dialog.tsx
Loan Detail Modal	View	View detailed loan information	loan-detail-modal.tsx
EMI Calculator	Calculator	Calculate loan EMIs	emi-calculator.tsx

6. UI Framework Forms (2 forms)

Form Name	Type	Purpose	Location
Form Component	Framework	Base form component with validation	ui/form.tsx
Alert Dialog	Framework	Confirmation dialogs	ui/alert-dialog.tsx

Form Architecture & Patterns

Validation Framework

- **React Hook Form:** Used for form state management and validation
- **Zod Schema Validation:** Type-safe form validation
- **Custom Validation:** Business logic validation for financial data

Form Submission Patterns

```
// Standard form submission pattern
const handleSubmit = async (e: React.FormEvent) => {
  e.preventDefault();
  setLoading(true);

  try {
    const response = await fetch('/api/endpoint', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(formData)
    });

    if (response.ok) {
      onSuccess();
      setOpenChange(false);
    }
  } catch (error) {
    toast.error('Operation failed');
  } finally {
    setLoading(false);
  }
};
```

Common Form Components

- **Input Fields:** Text, number, email, password inputs
- **Select Dropdowns:** Category selection, type selection
- **Date Pickers:** Transaction dates, bill due dates
- **Textareas:** Descriptions and notes
- **Checkboxes:** Boolean options
- **Radio Buttons:** Single choice selections
- **File Uploads:** Document attachments

Form State Management

- **useState:** Local form state management
- **useEffect:** Form initialization and cleanup
- **Custom Hooks:** Reusable form logic
- **Context API:** Shared form state across components

Form Types by Operation

Create Forms (8 forms)

- Add Transaction Dialog
- Add Bill Dialog
- Add Goal Dialog
- Add Goal Contribution Dialog
- Add Investment Dialog
- Add SIP Dialog
- Add Loan Dialog
- Add Loan Payment Dialog

Edit Forms (4 forms)

- Edit Transaction Dialog
- Edit Goal Dialog
- Edit Investment Dialog
- Edit Loan Dialog

Delete Forms (2 forms)

- Delete Investment Dialog
- Delete Loan Dialog

View/Detail Forms (3 forms)

- Transaction Detail Modal
- Bill Detail Modal
- Loan Detail Modal

Filter/Search Forms (1 form)

- Transaction Filters

Calculator Forms (3 forms)

- Goal Calculator
- Investment Calculators
- EMI Calculator

Import/Parser Forms (1 form)

- SMS Parser Dialog

Database Integration

Form-to-Database Mapping

Each form corresponds to specific database tables:

- **Transaction Forms** → `transactions` table
- **Bill Forms** → `bills` and `bill_instances` tables
- **Goal Forms** → `financial_goals` and `goal_contributions` tables
- **Investment Forms** → `investments` and `investment_transactions` tables
- **Loan Forms** → `loans` and `loan_payments` tables

API Endpoints

Forms interact with RESTful API endpoints:

- `POST /api/transactions` - Create transactions
- `PUT /api/transactions/[id]` - Update transactions
- `DELETE /api/transactions/[id]` - Delete transactions
- Similar patterns for bills, goals, investments, and loans

Getting Started Checklist

Pre-Development Setup

- [] Install Node.js 18+
- [] Install PostgreSQL 13+
- [] Install Git
- [] Clone repository

- ☐ Install dependencies (`npm install`)

Environment Configuration

- ☐ Copy `.env.local.example` to `.env.local`
- ☐ Configure database URL
- ☐ Set NextAuth secret
- ☐ Configure other environment variables

Database Setup

- ☐ Create PostgreSQL database
- ☐ Run migrations (`npm run db:migrate`)
- ☐ Seed database (`npm run db:seed`)
- ☐ Verify database connection

Development Server

- ☐ Start development server (`npm run dev`)
- ☐ Access application at `http://localhost:3000`
- ☐ Verify all forms are working
- ☐ Test database operations

Optional Docker Setup

- ☐ Install Docker and Docker Compose
- ☐ Run `docker-compose up -d` for containerized setup
- ☐ Verify services are running

Troubleshooting

Common Issues

Database Connection Issues

```
# Check PostgreSQL service
sudo systemctl status postgresql

# Verify database exists
psql -l | grep finplanner

# Test connection
psql "postgresql://postgres:password@localhost:5432/finplanner_dev"
```

Node.js Version Issues

```
# Check Node.js version
node --version

# Use Node Version Manager (nvm) if needed
nvm install 18
nvm use 18
```

Port Already in Use

```
# Find process using port 3000
lsof -i :3000

# Kill process if needed
kill -9 <PID>

# Or use different port
PORT=3001 npm run dev
```

Permission Issues

```
# Fix npm permissions
sudo chown -R $(whoami) ~/.npm

# Fix PostgreSQL permissions
sudo -u postgres createuser --superuser $USER
```

This comprehensive guide provides everything needed to download, set up, and understand the form architecture of the FinPlanner application. The application contains 38+ interactive forms covering all aspects of personal finance management, built with modern React patterns and robust validation frameworks.