Incometry Implementation Progress Tracker

Project Overview

- **Platform**: Canadian Financial Planning (TFSA focus)
- **Tech Stack**: React + Vite + Tailwind + Recharts + Firebase
- **Hosting**: React frontend on Hostinger Business
- **Status**: Ready to implement Firebase backend

Completed ✅

- [ ] Basic React app structure
- [ ] DRIP calculator functionality
- [ ] Portfolio tracking (localStorage)
- [ ] Tooltips system for financial terms
- [ ] Responsive design
- [ ] Local development environment

Current Phase: Firebase Implementation

Day 1: Firebase Setup

- [ ] Create Firebase project
- [ ] Install Firebase dependencies
- [ ] Configure Firebase in React app
- [ ] Set up authentication rules

Day 2: Authentication System

- [ ] Create login/register components
- [ ] Replace localStorage auth with Firebase Auth
- [ ] Test user registration/login
- [ ] Add password reset functionality

Day 3: Database Migration

- [ ] Set up Firestore database
- [ ] Create portfolio data structure
- [ ] Migrate localStorage portfolios to Firestore
- [ ] Test cross-device sync

Day 4: Real-time Features

- [ ] Implement real-time portfolio updates
- [ ] Add loading states and error handling
- [ ] Test concurrent user sessions
- [ ] Verify data persistence

Day 5: Deployment & Testing

- [ ] Deploy updated React app to Hostinger
- [ ] Configure Firebase hosting rules
- [ ] Test production authentication
- [ ] Verify cross-device functionality

Code Snippets to Remember

Firebase Config Template

javascript

```javascript
// src/config/firebase.js
import { initializeApp } from 'firebase/app';
import { getAuth } from 'firebase/auth';
import { getFirestore } from 'firebase/firestore';

const firebaseConfig = {
  // Your config here
};

const app = initializeApp(firebaseConfig);
export const auth = getAuth(app);
export const db = getFirestore(app);
```

Portfolio Service Pattern

javascript

```javascript
// src/services/portfolioService.js
import { collection, addDoc, getDocs, onSnapshot } from 'firebase/firestore';
import { db } from '../config/firebase';

export class PortfolioService {
  static async createPortfolio(userId, portfolioData) {
    // Implementation
  }

  static subscribeToPortfolios(userId, callback) {
    // Real-time subscription
  }
}
```

Environment Variables Needed

bash

```bash
VITE_FIREBASE_API_KEY=your-api-key
VITE_FIREBASE_AUTH_DOMAIN=incometry-app.firebaseapp.com
VITE_FIREBASE_PROJECT_ID=incometry-app
VITE_FIREBASE_STORAGE_BUCKET=incometry-app.appspot.com
VITE_FIREBASE_MESSAGING_SENDER_ID=123456789
VITE_FIREBASE_APP_ID=your-app-id
```

Quick Context for New Chats

When starting a new chat, paste this:

"I'm building Incometry, a Canadian financial planning platform. Current status: React app working locally with localStorage. Need to implement Firebase for cross-device user accounts and portfolio sync. I'm a beginner but learning step-by-step. Ready to continue from [CURRENT PHASE] in the tracker."

Key Decisions Made

- ✅ Using Firebase for backend (chosen over Python for faster implementation)
- ✅ Keeping React frontend on Hostinger Business
- ✅ Focus on TFSA investment strategies for Canadian market
- ✅ Tooltips system for financial education
- ✅ DRIP calculator as core feature

File Structure

```
src/
├── components/
│   ├── auth/          # Login/Register components
│   ├── ui/            # Tooltips, reusable components
│   └── layout/        # Navigation, footer
├── contexts/
│   └── AuthContext.jsx # Firebase authentication context
├── services/
│   └── portfolioService.js # Firestore database operations
├── pages/
│   ├── Dashboard.jsx   # User portfolio management
│   ├── DRIPCalculator.jsx
│   └── TFSAGuide.jsx
├── config/
│   └── firebase.js     # Firebase configuration
└── utils/
    └── financialTerms.js # Tooltip dictionary
```

Next Steps Summary

1. Set up Firebase project and authentication
2. Replace localStorage with Firestore database
3. Test cross-device synchronization
4. Deploy to production on Hostinger

Update Log

- **Session 1**: Planned Firebase implementation approach
- **Session 2**: [Update with progress]
- **Session 3**: [Update with progress]

Last updated: [Current Date]