Perfect 🚀 Now we’ll start building the **Real-Time Financial Insights & Analytics Platform** (our Scotiabank-style dashboard project) step by step, using **TDD from day one**.

Here’s the **kickoff plan** so we don’t get lost:

**🔹 Phase 1 — Setup & Foundations**

1. **Create Django project + app**
   * Project name: realtime\_dashboard
   * App name: dashboard
2. **Setup DRF & pytest**
   * Install Django REST Framework, pytest, pytest-django
   * Configure testing environment
3. **Setup React frontend**
   * Project name: frontend
   * Install React + Axios + Tailwind (for UI)

**🔹 Phase 2 — Core Features (Backend-first with TDD)**

✅ We’ll build in this order:

1. **Authentication (JWT login/signup)**
   * TDD: test login API before writing code.
2. **Portfolio Management (Add/View stocks)**
   * TDD: write test for user portfolio API.
3. **Real-time Stock Prices (TSX API integration)**
   * Mock API responses in tests.
4. **Transactions (Buy/Sell simulation)**
   * Ensure ACID with DB transactions.
5. **Audit Logs (Triggers-like behavior)**
   * Keep history of every action.

**🔹 Phase 3 — Frontend (React)**

1. **Authentication Pages** (Login/Register)
2. **Dashboard with Real-time Data (WebSocket or Polling)**
3. **Portfolio Page** (List, Add, Edit, Delete)
4. **Charts & Analytics** (Recharts for visualizations)

**🔹 Phase 4 — Integration & Testing**

1. **Integration Tests** → frontend ↔ backend
2. **Performance Tests** → simulate load
3. **Security Tests** → permissions, rate-limiting

⚡ First Step: Initialize the **Django project** + **pytest setup** (so we can start with TDD for Authentication).

👉 Do you want me to:

1. **Generate the commands + folder structure** for you to run locally,  
   or
2. **Write the initial tests/test\_auth.py file** so we immediately jump into TDD?