🚀Detailed Implementation Prompts for Each Segment — setLedger

This document provides **detailed development prompts** for each segment of the setLedger system. These prompts are designed to be used phase-by-phase in Vibe Coding IDE (or any collaborative dev environment) to achieve full implementation with clarity, modularity, and efficiency.

🏗️Phase 0 — Discovery & Infrastructure Setup (2 Weeks)

🚢Goal: Establish the foundation — architecture, authentication, multi-tenant design, CI/CD setup.

Prompt 0.1 — System Architecture & Planning

> *"Create a modular architecture diagram for setLedger including: frontend (React), backend (Node.js + Express), AI microservice (Flask), databases (MongoDB + Firebase), and data flow between modules. Highlight authentication layer, API gateway, and error handling points."*

Prompt 0.2 — Database & Data Model Design

> *"Design MongoDB schemas for: Organization, User, Product, Invoice, Stock, Transaction, GST Report, and AI Analytics. Ensure relational linking via orgID_memberID and timestamp fields for versioning."*

Prompt 0.3 — Multi-Tenant Authentication Setup

> *"Implement multi-tenant authentication using JWT + bcrypt + Speakeasy. Generate userID = orgID_memberID dynamically. Add routes for registration, login, TOTP setup, and email OTP fallback via Firebase Auth."*

Prompt 0.4 — CI/CD Pipeline Initialization

> *"Set up GitHub Actions for automated build, linting, and deploy to GitHub Pages (frontend) and Render/Heroku (backend). Include environment variable setup via GitHub Secrets."*

Prompt 0.5 — Environment Configuration

> *"Create a .env.example file with all necessary keys: MONGO_URI, FIREBASE_KEY, JWT_SECRET, TOTP_SECRET, EMAIL_API_KEY, AI_API_KEY. Implement dotenv integration for backend and frontend."*

Phase 1 — Core MVP (8–12 Weeks)

🚢Goal: Build functional backend and modular frontend for primary services.

💈 Sprint 1–2: Authentication, Organization & Product Modules

Prompt 1.1 — Authentication & User Roles

> *"Implement login, register, and TOTP verification endpoints. Define roles: Admin, Accountant, Analyst, Staff. Protect routes with role-based middleware. Integrate Firebase for email OTP."*

Prompt 1.2 — Organization & Team Management

> *"Develop org registration page. Enable admin to invite members via email. Generate orgID and maintain org-user hierarchy. Store user metadata and permissions."*

Prompt 1.3 — Product Management & QR Code Generator

> *"Build Product CRUD API (Node.js + Express). Generate product QR codes using* `qrcode` *library. Add UI for product registration and QR preview/download. Include bulk QR export option."*

💈 Sprint 3–4: Billing & POS Basics

Prompt 1.4 — Billing Module

> *"Create an invoicing system: auto-calculate taxes (GST), discounts, and total. Generate PDF invoices using jsPDF. Include QR on invoices containing product and payment metadata."*

Prompt 1.5 — POS Interface & Offline Billing

> *"Develop a PWA-compatible Point of Sale (POS) interface using IndexedDB for offline billing. Enable QR scanning (React-Barcode-Scanner). Sync data with backend when online."*

💈 Sprint 5–6: Inventory & Accounting Basics

Prompt 1.6 — Inventory Management

> *"Implement stock tracking: add, remove, adjust quantities automatically upon billing. Include low-stock alert system with email triggers via Firebase Cloud Messaging."*

Prompt 1.7 — Accounting & Journal Entries

> *"Add journal entry creation for each sale or purchase. Maintain general ledger structure (debit, credit). Support CSV import for external transactions."*

💈 Sprint 7: Backup & Error Handling

Prompt 1.8 — Backup and Sync

> *"Implement dual backups — Firebase Firestore + LocalStorage. Use CRON job in Express to sync every 24 hours. Add restore feature for lost data."*

Prompt 1.9 — Error Handling Framework

> *"Integrate Winston logger and centralized error handler middleware. Implement frontend error toasts and fallback data from local backup if API fails."*

Phase 2 — AI & Pricing Engine (8 Weeks)

🚢Goal: Integrate predictive AI for stock, pricing, and financial forecasting.

Prompt 2.1 — AI Stock Prediction

> *"Create a Flask-based AI microservice that predicts stock depletion dates using time-series forecasting (ARIMA/Prophet). Return API endpoint for frontend dashboard display."*

Prompt 2.2 — Smart Pricing Algorithm

> *"Implement AI-driven dynamic pricing using regression models trained on historical sales, competitor prices (scraped), and demand elasticity. Include manual override in UI."*

Prompt 2.3 — Financial Forecasting Dashboard

> *"Build chart components (Recharts/Chart.js) for revenue, expenses, profit trends. Include AI-based forecast overlay lines and interactive data filters."*

Prompt 2.4 — AI Chat Assistant

> *"Create an interactive AI assistant using Gemini API or OpenAI free-tier model. Train with organization data (sales, expenses) for context-aware insights. Display chat UI with quick action buttons."*

🏫Phase 3 — GST, Compliance, and Reporting (6 Weeks)

🚢Goal: Enable full compliance and reporting functionality.

Prompt 3.1 — GST & Tax Module

> *"Integrate free GST validation API. Auto-generate GSTR-1 and GSTR-3B reports. Provide download and print options. Store reports per organization securely."*

Prompt 3.2 — Tax Alerts & Compliance

> *"Set up automated reminders for upcoming tax deadlines via Firebase Notifications. Include dashboard widget for pending filings."*

Prompt 3.3 — Financial Report Generator

> *"Generate financial reports (P&L, balance sheet, cash flow). Add export to PDF, Excel, CSV. Display with data visualizations."*

🏪Phase 4 — Cloud Integration & Data Reliability (4 Weeks)

🚢Goal: Strengthen cloud sync, data persistence, and redundancy.

Prompt 4.1 — Firebase Cloud Integration

> *"Integrate Firestore for data sync and backups. Encrypt sensitive fields before upload. Provide manual sync and restore buttons in settings."*

Prompt 4.2 — Offline Sync Engine

> *"Implement real-time sync using Service Workers. When offline, queue all POST/PUT requests and replay them on reconnect."*

Prompt 4.3 — Monitoring & Logs

> *"Integrate admin dashboard for system logs. Use Winston and Firebase Crashlytics for error reports. Provide UI for log filtering by severity."*

🚲Phase 5 — UI/UX Design & Frontend Experience (4 Weeks)

🚢Goal: Design for user psychology and accessibility.

Prompt 5.1 — UI Layout

> *"Create a modular dashboard with sidebar navigation. Each service (Billing, Inventory, GST, AI, etc.) should be a button-click away. Use Tailwind + Framer Motion for modern transitions."*

Prompt 5.2 — Accessibility & Design Psychology

> *"Implement high-contrast mode, dark/light themes, and keyboard shortcuts. Follow 'one-click-to-action' principle for usability."*

Prompt 5.3 — Personalization

> *"Allow theme customization per organization. Save preferences in user profile. Include welcome dashboard showing revenue, alerts, and recommendations."*

🚛Phase 6 — Security, Deployment & Future Enhancements (6 Weeks)

🚢Goal: Finalize security layers, deployment, and extendibility.

Prompt 6.1 — Security Audit

> *"Run full app audit with OWASP checklist. Test authentication flow, rate limiting, and environment key isolation."*

Prompt 6.2 — Hosting & Deployment

> *"Host frontend via GitHub Pages (free). Deploy backend and Flask AI microservice via Render/Heroku. Use Firebase for real-time sync and notifications."*

Prompt 6.3 — Future-Ready Features

> *"Plan integration with blockchain for transaction integrity. Add multi-language support and WhatsApp/email alerts for invoices."*

End Result

By following these prompts phase-by-phase, **setLedger** will become a complete, modular, AI-enhanced financial platform that unifies accounting, inventory, billing, and analytics — with robust security, modularity, and scalability built for every business scale.