**Project Overview**

We are developing a complete ERP system from scratch to streamline business operations, improve data accessibility, and enhance decision-making. The system will include core modules such as finance, HR, inventory, supply chain management, and CRM.

The project will be executed in phases over an estimated **6-month period**, with a single developer and 2 tester and review working on a project basis. This timeline can be adjusted based on complexity, additional resources, and evolving business needs.

**Required Skills for the Developer**

To ensure successful execution, the developer must possess the following technical and functional skills:

**Technical Skills**

* **Programming Languages:** Proficiency in Java, Python (Django), or Node.js for backend development.
* **Frontend Development:** Expertise in React.js, Vue.js, or Angular.
* **Database Management:** Strong knowledge of PostgreSQL, MySQL, or MongoDB.
* **API Development & Integration:** Experience in RESTful APIs, GraphQL, and third-party software integrations.
* **Cloud & Deployment:** Familiarity with AWS, Azure, or self-hosted server deployment, including Docker and Kubernetes.
* **Security Protocols:** Proficiency in authentication (OAuth, JWT), encryption, and role-based access control.
* **ERP Experience:** Prior experience developing or customizing ERP solutions.
* **Version Control & Project Management:** GitHub, GitLab, Jira, or equivalent.

**Soft Skills & Business Acumen**

* Strong understanding of business processes (finance, HR, supply chain).
* Excellent problem-solving and critical-thinking skills.
* Effective communication and documentation abilities.
* Commitment to deadlines and agile development methodology.

**Project Goals (Desktop Version)**

1. Centralize Operations: Unify all key business processes into a desktop platform.
2. Offline Accessibility: Enable full functionality without internet connectivity.
3. Efficiency Gains: Automate tasks across departments.
4. Real-Time Insights: Provide comprehensive analytics within the desktop app.
5. Scalability: Build a foundation for future enhancements and web integration.
6. User-Friendly Design: Deliver an intuitive desktop interface.
7. Data Security: Protect sensitive data with offline-capable measures.

**Required Modules (Desktop Version)**

The desktop ERP will include all six core modules at launch:

1. Finance and Accounting
   * Features: General ledger, accounts payable/receivable, budgeting, financial reporting, tax management.
2. Human Resources (HR)
   * Features: Employee records, payroll, attendance tracking, recruitment, performance reviews.
3. Inventory Management
   * Features: Stock tracking, reorder alerts, warehouse management, inventory valuation.
4. Supply Chain Management
   * Features: Supplier database, purchase orders, logistics tracking, demand forecasting.
5. Customer Relationship Management (CRM)
   * Features: Customer database, sales tracking, lead management, support ticketing.
6. Reporting and Analytics
   * Features: Customizable dashboards, KPI tracking, exportable reports (PDF, Excel).

**Technical Requirements (Desktop Version)**

* Platform: Standalone desktop app for Windows (macOS/Linux optional later).
* Development Approach: Single developer using a framework like Electron, .NET, or Qt.
* Database: SQLite or local PostgreSQL for offline use, with sync capability for future web integration.
* Integration: Support for key tools (e.g., QuickBooks, Excel export) via local APIs/file handling.
* Security: Role-based access, local data encryption.
* Scalability: Modular design to align with future web version.
* UI/UX: Native-like desktop interface.

**Team Structure (Desktop Version)**

* 1 Developer: Full-time, responsible for end-to-end development.
  + Tasks: Backend, frontend, integrations, security.
* 2 Testers/Reviewers: Part-time or as-needed roles.
  + Tester/Reviewer 1: Monitors progress, reviews designs, conducts initial testing.
  + Tester/Reviewer 2: Performs detailed testing (unit, integration, UAT), validates functionality.

**1. Define the System Architecture**

The ERP system should follow a **3-Tier Architecture** for maintainability and scalability:

1. **Presentation Layer (Frontend)**
   * **Technology:** Electron (for desktop), or .NET/Qt
   * **Responsibilities:** UI components, user interaction, validation
   * **Tools:** React.js/Vue.js/Angular (if using Electron), WPF (for .NET), Qt (for C++)
2. **Business Logic Layer (Backend)**
   * **Technology:** Python (Django/FastAPI), Java (Spring Boot), or Node.js
   * **Responsibilities:** Processing requests, enforcing business rules, API endpoints
   * **Tools:** Django ORM, Express.js, Hibernate (Java)
3. **Data Layer (Database & Storage)**
   * **Technology:** PostgreSQL, SQLite (for offline mode), or MySQL
   * **Responsibilities:** Storing and managing structured data
   * **Tools:** SQLAlchemy (Python), TypeORM (Node.js), Hibernate (Java)