**SCHEME FOR DEVELOPMENT OF So4 Project Manager**

**1. System Architecture Overview**

┌────────────────────────────────────────────────────────────────┐

│ FRONTEND (UI Layer) │

│ - Dashboard (KPIs, active projects, profitability) │

│ - Project Management (CRUD, status tracking) │

│ - Task & Milestone Management │

│ - Time Entry Interface (daily/weekly logs) │

│ - Reports & Analytics View │

│ - Billing & Invoicing Interface │

│ - User Management (roles: Admin, Manager, Employee, Client) │

└────────────────────────────────────────────────────────────────┘

│

▼

┌────────────────────────────────────────────────────────────────┐

│ BACKEND (Logic Layer) │

│ - Project Controller: create, update, archive projects │

│ - Task Controller: task CRUD, progress, and completion logic │

│ - Time Tracking Controller: log hours, approvals, edits │

│ - Billing & Cost Engine: hourly rate \* hours = cost │

│ - Report Generator: cost summaries, performance insights │

│ - Authentication & Access Control │

│ - Notification & Alerts (deadlines, overspend warnings) │

└────────────────────────────────────────────────────────────────┘

│

▼

┌────────────────────────────────────────────────────────────────┐

│ DATABASE LAYER │

│ Tables: │

│ • Users (ID, name, role, email, hourly\_rate) │

│ • Clients (ID, name, contact, billing\_info) │

│ • Projects (ID, client\_id, name, desc, start, end, budget) │

│ • Tasks (ID, project\_id, name, status, assigned\_to) │

│ • TimeLogs (ID, user\_id, task\_id, date, hours\_worked) │

│ • Billing (project\_id, total\_hours, total\_cost, billed) │

│ • Reports (project\_id, metrics, profitability\_index) │

└────────────────────────────────────────────────────────────────┘

│

▼

┌────────────────────────────────────────────────────────────────┐

│ INTEGRATIONS & EXTENSIONS │

│ - Email Notifications / Slack integration │

│ - Export Reports (PDF, Excel) │

│ - Optional Cloud Sync / API for external tools │

│ - Future: Tokenized reward or smart contract billing module │

└────────────────────────────────────────────────────────────────┘

**2. Development Phases**

**Phase 1 – Core Database Design (MS Access / SQL / PostgreSQL)**

* Create relational tables for **Projects**, **Tasks**, **Users**, **TimeLogs**, **Rates**, and **Clients**.
* Establish relationships (1-to-many, many-to-many where needed).
* Add automatic **cost computation fields** via queries or triggers.

**Phase 2 – User Interface Development**

* Build intuitive **forms and dashboards**:
  + Project setup & tracker
  + Task assignment
  + Time entry (with date picker)
  + Cost overview panel
* Use conditional formatting for overdue tasks or over-budget projects.

**Phase 3 – Automation Logic**

* Implement:
  + Automatic hourly rate × time calculations.
  + Dynamic project summaries (cost, time spent, progress %).
  + Budget alerts (if cost > 90% of budget).

**Phase 4 – Reporting & Analytics**

* Create dashboards:
  + Total Hours vs. Budget
  + Employee utilization
  + Project profitability ranking
  + Billable vs. Non-billable time
* Export options (PDF, Excel, CSV).

**Phase 5 – Customization & Scalability**

* Add custom fields for industry-specific needs.
* Support cloud database sync or migration to a web app (React + Node.js backend).
* Optionally integrate a **tokenized payment/reward layer** (if extending toward blockchain billing).

**3. Tech Stack Options**

| **Layer** | **Local/Offline Option** | **Cloud/Web Option** |
| --- | --- | --- |
| Database | MS Access / SQLite | PostgreSQL / MySQL |
| Backend | VBA / Python Script | Node.js / Django REST |
| Frontend | Access Forms | React + Tailwind UI |
| Reporting | Access Reports / Excel | Power BI / Chart.js |
| Auth | Access Users Table | JWT Authentication |

**4. Workflow Diagram**

Project Created → Tasks Assigned → Time Logged →

Hourly Rate Applied → Cost Calculated → Report Generated → Client Invoiced

**5. Key Deliverables**

* ✅ Centralized database (projects, tasks, time, cost)
* ✅ Real-time cost tracking & profit analysis
* ✅ Dynamic reports for management and billing
* ✅ Scalable structure for future online expansion