**Goal**: Build a **web-based KPI Management System** for an IT service-based organization using **Python** (Flask or Django) for the backend and frontend logic, and **Power BI** for dashboards and reporting. The system will help teams define, track, govern, and report on KPIs effectively.

**Tech Stack**

| **Layer / Area** | **Technology / Tool** | **Purpose / Notes** |
| --- | --- | --- |
| Frontend (UI) | HTML, CSS, Bootstrap | Responsive UI, fast and simple styling |
|  | Jinja2 (Flask templating) | Embeds data into HTML pages from backend |
| Charts & Graphs | Chart.js | Visualize KPIs, trends, progress bars, etc. |
| Tables | DataTables.js | Interactive tables with search, pagination, export options |
| Backend (Server) | Flask | Lightweight, easy-to-learn Python framework |
|  | Flask-Login | User login/logout session management |
|  | Flask-WTF | Secure form handling with CSRF protection |
|  | Werkzeug Security | Password hashing and verification |
|  | Flask-Mail *(optional)* | For password reset and notifications |
|  | Flask Principal | Role based access |
| Database | SQLite (for MVP) | SQLite for early dev, PostgreSQL for production |
|  | SQLAlchemy | ORM (Object Relational Mapper) to interact with DB using Python |
| APIs (optional) | Flask-RESTful or Flask APIView | To expose data via APIs if needed (future-proofing) |
| Authentication | Flask-Login + password hashing | Simple and secure session-based login |
| Deployment | Render / Heroku / Railway | Free or low-cost, easy deployment options for Flask apps |
| Version Control | GitHub | Code versioning and team collaboration |
| Security | HTTPS (via deployment platform), CSRF, password hashing | Basic security best practices |

**System Workflow**

1. **User Roles & Access**:
   * Admin: Configure system and users, Logs, Configuration, Project Addition, and resource allocation
   * Project Manager: Define & assign KPIs, benchmark KPI, Add notes highlight or low lights
   * Team Member: can read and view KPI for assigned projects
   * Management: Can view and manage KPI, dashboards for decision making
2. **KPI Lifecycle**:
   * Create Project
   * Assign Project manager and other team members
   * Create KPI definitions (custom or predefined KPI by Admin)
   * Create a benchmark for KPI. Could be a manual entry or based on average of last 6 month data. Project manager or admin can select how to measure the benchmark.
   * Rule to be created on how KPI should reflect (Red/Green/Amber) based on achieved KPI versue the benchmark
   * Assign KPIs to projects. Can also be assigned to individuals
   * Enable periodic updates with status and notes
   * Managers validate submissions
   * Auto-flag anomalies or delays
   * Display real-time Power BI dashboards
   * Archive KPI history for audit
   * KPI can be set Daily/Weekly/Monthly timeline. Depending on nature of project the recurrence can be selected by Project manager.
   * Mandatory to update the KPI for each recurrence

**Key Features**

* Dashboard overview of all KPIs with filters
* Status tracking with color coding (green/yellow/red)
* Notifications for overdue or abnormal KPIs which not meeting the benchmark standard.
* Feedback & comments per KPI
* Export options (CSV, PDF)
* Role-based data visibility
* Secure audit logs