**Database Design for Service Management System**

**1. User Type Table**

* ID (Primary Key)
* Role (Admin, Service Head, Service Engineer)
* Description

**2. Users Table**

* + ID (Primary Key)
  + Full Name
  + Password (hashed)
  + Email
  + Phone Number
  + Type ID
  + Report to (User ID)
  + Created At (timestamp)
  + Updated At (timestamp)

**3. Customers Table**

* + Customer ID (Primary Key)
  + Full Name
  + Contact Information (Phone, Email)
  + Address
  + Company Name
  + Created At (timestamp)

**4. Tickets Table**

* + Ticket ID (Primary Key)
  + Customer ID (Foreign Key from Customers table)
  + Issue Description
  + Created At (timestamp)
  + Updated At (timestamp)
  + Is Taken (Boolean)
  + Status (Boolean)

**5. Ticket Assigned Table**

* + Assignment ID (Primary Key)
  + Ticket ID (Foreign Key from Tickets table)
  + Service Head ID (Foreign Key from Users table)
  + Service Engineer ID (Foreign Key from Users table)
  + Assigned Date
  + Status (Pending, In Progress, Completed)
  + Created At (timestamp)

**6. Ticket progress Table**

* ID (Primary Key)
* Ticket ID (Foreign Key from Tickets table)
* Service Engineer ID (Foreign Key from Users table)
* Priority (Low, Medium, High)
* Expected Completion Date
* Actual Completion Date
* Spare Parts Required (boolean)
* Cost
* Status (Pending, In Progress, Completed)
* Created At (timestamp)

**7. Spare Parts Table**

* + Part ID (Primary Key)
  + Ticket ID (Foreign Key from Tickets table)
  + Part Name
  + Quantity
  + Unit Price
  + Issued Date
  + Issued By (Service Head ID)

**8. Daily Updates Table**

* + Update ID (Primary Key)
  + Service Engineer ID (Foreign Key from Users table)
  + Progress Description
  + Update Date
  + Created At (timestamp)

**9. Travel Expense Reports Table**

* + Report ID (Primary Key)
  + Service Engineer ID (Foreign Key from Users table)
  + Ticket ID (Foreign Key from Tickets table)
  + Expense Details (description of costs: petrol, hotel, etc.)
  + Total Amount
  + Submission Date
  + Approval Status (Pending, Approved, Rejected)
  + Approved By (Service Head ID, Foreign Key from Users table)
  + Created At (timestamp)

**Module**

**1. User Management Module**

**Endpoints:**

* **User Registration & Authentication**
  + POST /api/register: Register a new user.
  + POST /api/login: User login.
  + POST /api/forgot-password: Request password reset.
  + POST /api/reset-password: Reset password using token.
* **User Profile Management**
  + GET /api/users/{user\_id}: Get user profile details.
  + PUT /api/users/{user\_id}: Update user profile.
  + DELETE /api/users/{user\_id}: Delete a user.
* **User Role Management**
  + POST /api/users/{user\_id}/assign-role: Assign or update user role.
  + GET /api/users/roles: List all user roles.

**2. Ticket Management Module**

**Endpoints:**

* **Ticket Creation & Management**
  + POST /api/tickets: Create a new service ticket.
  + GET /api/tickets/{ticket\_id}: Get details of a specific ticket.
  + PUT /api/tickets/{ticket\_id}: Update ticket details.
  + DELETE /api/tickets/{ticket\_id}: Delete a ticket.
* **Ticket Assignment & Tracking**
  + POST /api/tickets/{ticket\_id}/assign: Assign a ticket to a service engineer.
  + GET /api/tickets/{ticket\_id}/history: Get history and updates of a ticket.
  + PUT /api/tickets/{ticket\_id}/update-status: Update ticket status and priority.
* **Daily Progress Updates**
  + POST /api/tickets/{ticket\_id}/updates: Submit daily progress update.
  + GET /api/tickets/{ticket\_id}/updates: Get daily progress updates.

**3. Customer Management Module**

**Endpoints:**

* **Customer Profile Management**
  + POST /api/customers: Add a new customer.
  + GET /api/customers/{customer\_id}: Get customer details.
  + PUT /api/customers/{customer\_id}: Update customer information.
  + DELETE /api/customers/{customer\_id}: Delete a customer.
* **Customer Machinery Details**
  + GET /api/customers/{customer\_id}/machinery: Get machinery details for a customer.
  + POST /api/customers/{customer\_id}/machinery: Add or update machinery details.

**4. Spare Parts Management Module**

**Endpoints:**

* **Spare Parts Inventory**
  + POST /api/spare-parts: Add new spare parts to inventory.
  + GET /api/spare-parts/{part\_id}: Get details of a specific spare part.
  + PUT /api/spare-parts/{part\_id}: Update spare part information.
  + DELETE /api/spare-parts/{part\_id}: Delete a spare part.
* **Spare Parts Issuance**
  + POST /api/tickets/{ticket\_id}/spare-parts: Issue spare parts for a ticket.
  + GET /api/tickets/{ticket\_id}/spare-parts: Get spare parts issued for a ticket.

**5. Travel Expense Management Module**

**Endpoints:**

* **Expense Submission & Approval**
  + POST /api/expenses: Submit travel expense report.
  + GET /api/expenses/{expense\_id}: Get details of an expense report.
  + PUT /api/expenses/{expense\_id}/approve: Approve an expense report.
  + PUT /api/expenses/{expense\_id}/reject: Reject an expense report.
* **Expense History**
  + GET /api/users/{user\_id}/expenses: Get all expense reports submitted by a user.

**6. Reporting and Dashboard Module**

**Endpoints:**

* **Admin Dashboard**
  + GET /api/dashboard/admin: Get an overview of tickets, employee status, etc.
* **Reports**
  + GET /api/reports/employee: Get employee performance report.
  + GET /api/reports/service: Get service report for tickets.
  + GET /api/reports/service-head: Get report on service head’s team performance.
* **Custom Reports**
  + POST /api/reports/custom: Generate a custom report.

**7. Notification and Alert Module**

**Endpoints:**

* **Notifications**
  + GET /api/notifications: Get all notifications for the current user.
  + POST /api/notifications/mark-read: Mark notifications as read.
* **Alerts**
  + GET /api/alerts: Get system alerts related to ticket and expense reports.