## ****Endpoints Of Leave Management System****

## ****1. Authentication****

| **Endpoint** | **Method** | **Roles** | **Body Example** | **Description** |
| --- | --- | --- | --- | --- |
| /auth/login | POST | All | json { "username": "admin", "password": "admin123" } | Login to get JWT token |
| /auth/register | POST | Admin | json { "username": "john", "password": "password123", "role": "STAFF", "name": "John Doe", "department": "IT" } | Create new user |

## ****2. User Endpoints****

| **Endpoint** | **Method** | **Roles** | **Body Example** | **Description** |
| --- | --- | --- | --- | --- |
| /api/users | POST | Admin | json { "username": "jane", "password": "pass123", "role": "MANAGER" } | Create user |
| /api/users | GET | Admin: any, Manager: non-admin | - | Get all users |
| api/users/paged | GET | Admin: any, Manager: non-admin | ?page=0&size=10&sort=username,asc | Get paginated users |
| /api/users/{id} | GET | Admin: any, Manager: non-admin, Staff: self | - | Get single user |
| /api/users/{id} | PUT | Admin: any, Manager & Staff: self | json { "username": "jane\_updated", "role": "STAFF" } | Update user |
| /api/users/{id} | PATCH | Admin: any, Manager & Staff: self | json { "role": "MANAGER" } | Partial update |
| /api/users/{id} | DELETE | Admin | - | Delete user |

## ****3. EmployeeProfile Endpoints****

| **Endpoint** | **Method** | **Roles** | **Body Example** | **Description** |
| --- | --- | --- | --- | --- |
| /api/user-profiles | GET | Admin, Manager | - | Get all users with employee profile and leave requests |
| /api/user-profiles/paged | GET | Admin, Manager | ?page=0&size=10&sort=username,asc | Get paginated users with employee profile and leave requests |
| /api/user-profiles/{userId} | GET | Admin: any, Manager: non-admin, Staff: self | - | Get single user profile + leave requests |
| /api/user-profiles/me | GET | Admin, Manager, Staff | - | Get own profile + leave requests |

## ****4. LeaveRequest Endpoints (Initiator/Employee)****

| **Endpoint** | **Method** | **Roles** | **Body Example** | **Description** |
| --- | --- | --- | --- | --- |
| /api/leaves | GET | Employee | - | Get all leave requests of current user |
| /api/leaves/pending | GET | Employee | - | Get all pending leaves of current user |
| /api/leaves/approved | GET | Employee | - | Get all approved leaves of current user |
| /api/leaves/rejected | GET | Employee | - | Get all rejected leaves of current user |
| /api/leaves/{id} | GET | Employee | - | Get leave request by ID |
| /api/leaves | POST | Employee | json { "startDate": "2025-09-15", "endDate": "2025-09-20", "days": 5, "requestComment": "Vacation" } | Create new leave request |
| /api/leaves/{id} | PUT | Employee | json { "startDate": "2025-09-16", "endDate": "2025-09-21", "days": 6, "requestComment": "Changed plan" } | Update leave request (only pending) |
| /api/leaves/{id} | PATCH | Employee | json { "requestComment": "Updated reason" } | Partial update (only pending) |
| /api/leaves/{id} | DELETE | Employee | - | Delete leave request (only pending) |

## ****5. LeaveRequest Endpoints (Action/Admin/Manager)****

| **Endpoint** | **Method** | **Roles** | **Body Example** | **Description** |
| --- | --- | --- | --- | --- |
| /api/action-leaves | GET | Admin, Manager | - | Get all leave requests |
| /api/action-leaves/pending | GET | Admin, Manager | - | Get all pending leave requests |
| /api/action-leaves/approved | GET | Admin, Manager | - | Get all approved leave requests |
| /api/action-leaves/rejected | GET | Admin, Manager | - | Get all rejected leave requests |
| /api/action-leaves/{id} | GET | Admin, Manager | - | Get leave request by ID |
| /api/action-leaves/{id}/action | PATCH | Admin, Manager | json { "status": "APPROVED", "actionComment": "Enjoy your leave" } | Approve/Reject leave request with comment |
| /api/action-leaves/{id} | DELETE | Admin, Manager | - | Delete leave request by ID |

### **Notes on Roles & Authorization**

* **Employee**: Can manage only their own leave requests (create, update, delete pending).
* **Manager**: Can view and action leave requests of all employees except Admins; cannot action on their own leave.
* **Admin**: Full access for users, leave requests, and actions.

### ****Notes for Testing****

1. Always include **JWT token** in Authorization: Bearer <token> header.
2. Staff can only perform operations on their own leave requests.
3. Manager cannot approve/reject their own leave requests.
4. PENDING leave requests are the only ones that can be updated/deleted by the employee.
5. Leave requests include both:

* requestComment → submitted by employee
* actionComment → filled by manager/admin when approved/rejected