**HR SYSTEM**

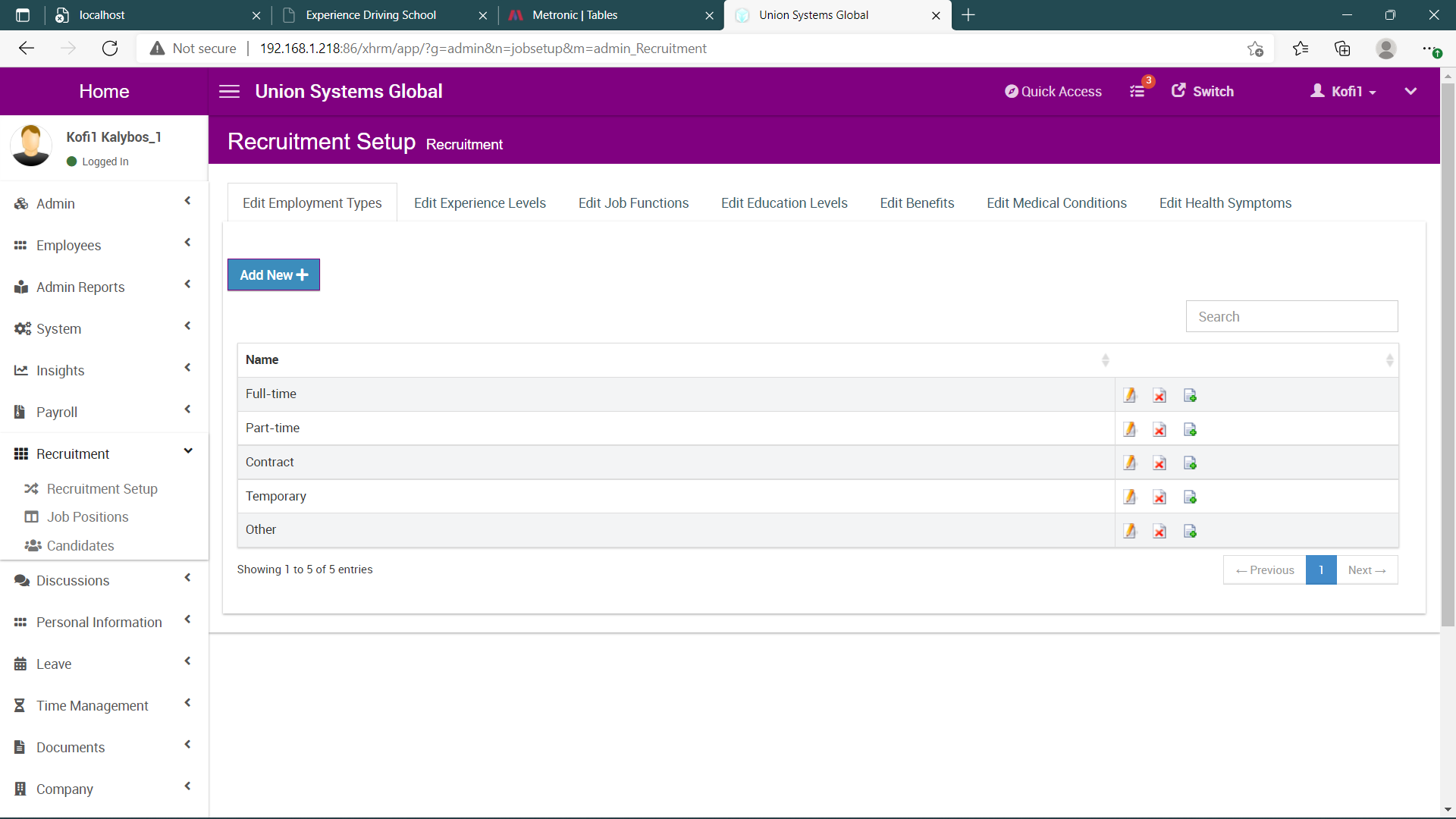
**USER ACCESS**

The hr system has two access levels

* ADMIN
* MANAGER
* EMPLOYEE
* OTHERS

**ADMIN STRUCTURE**

* **MODULES FOR ADMIN**
* Recruitment
* Employee
* Users
* Admin
* **RECRUITMENT MODULE**

****

The recruitment module has three different sessions

1. Recruitment Setup
2. Job Positions
3. Candidates

* **RECRUITMENT SETUP**

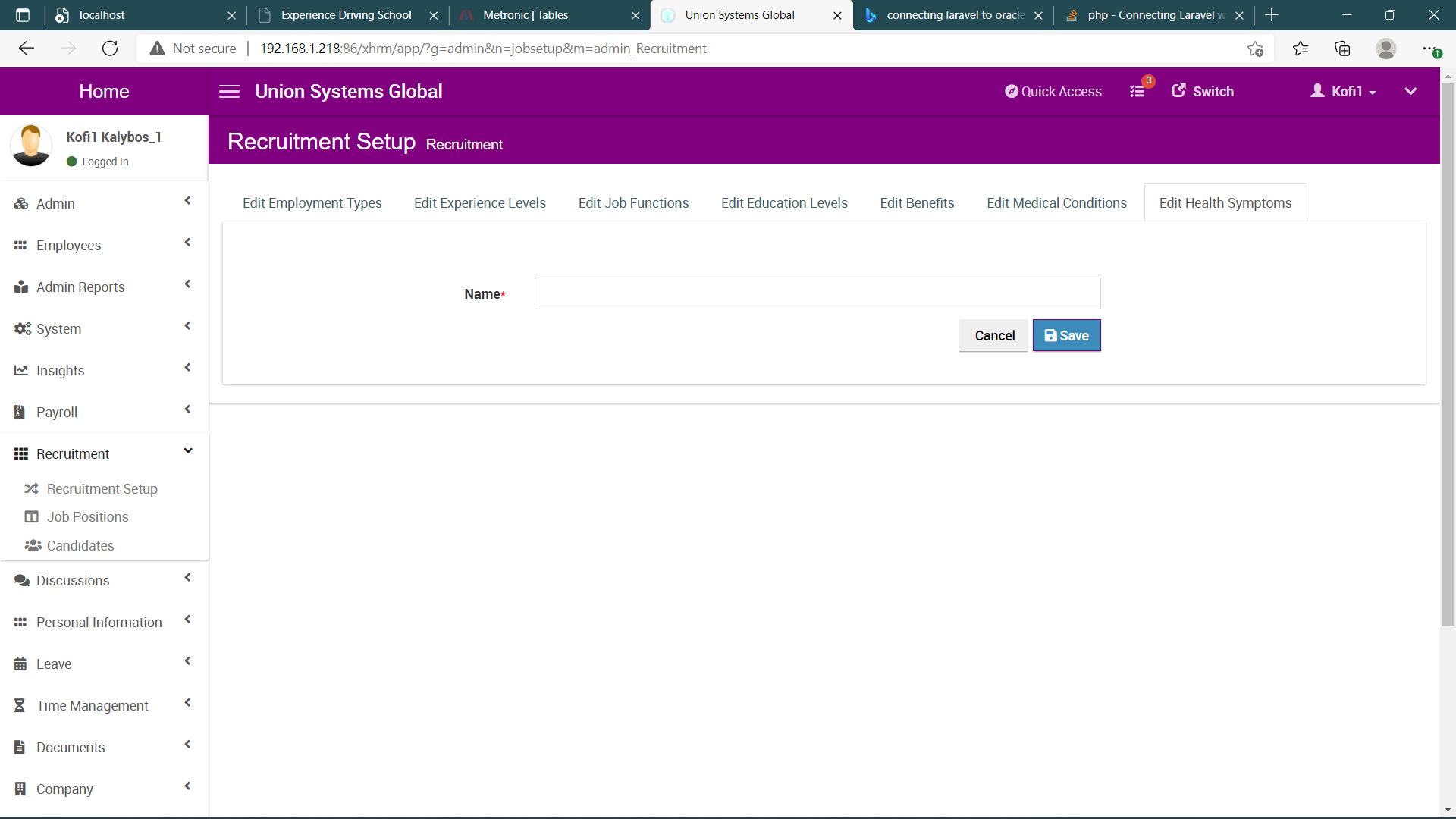
The recruitment setup is responsible for creating parameters for the recruitment form it has seven (7) different setup parameters

1. Edit Employment Types (Creation of employment types)
2. Edit Experience Level (Creation of Experience Levels)
3. Edit Job Functions (Creation of Job Functions)
4. Edit Education Levels (Creation of Educational Levels)
5. Edit Benefits (Creation of Benefits)
6. Edit Medical Conditions (Creation of Medical Conditions)
7. Edit Health Symptoms

Each tab handles the creation, editing and deleting of respective recruitment parameters

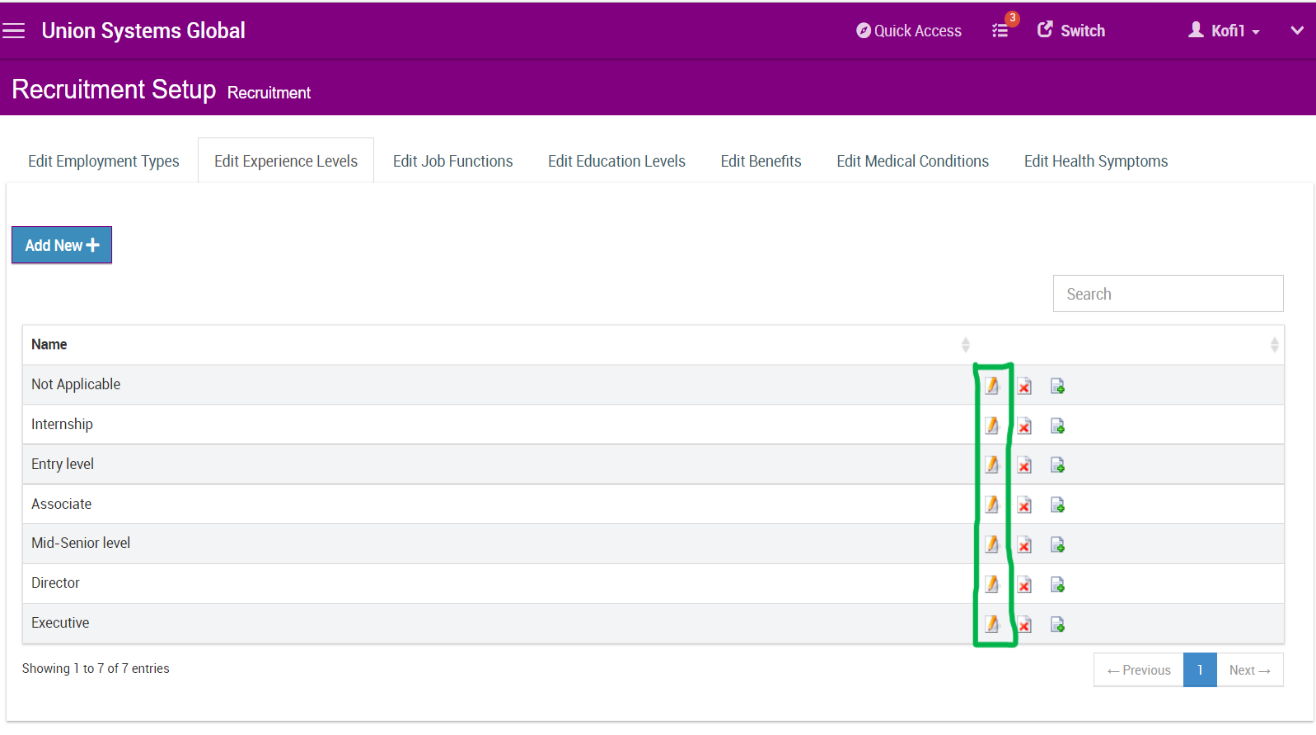
**CREATION OF RECRUITMENT PARAMETERS**

When a user clicks on the **ADD NEW BUTTON**, a form shows up for the user to add a parameter, once the save button is clicked a new parameter will be added, this applies to all the parameter tabs



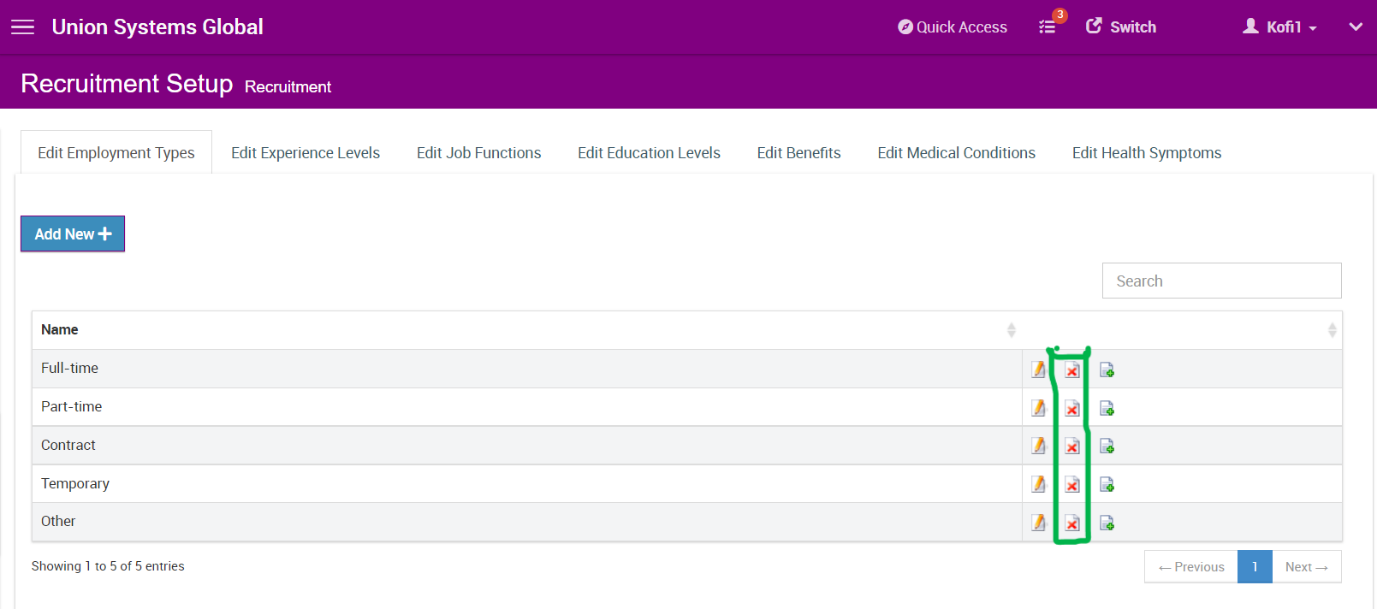
**EDIT RECRUITMENT PARAMETER**

A user can edit any recruitment parameter by clicking on the pencil icon relating to a particular item, this will bring an edit form, once the save button is clicked the parameter will be updated, this applies to all the tabs



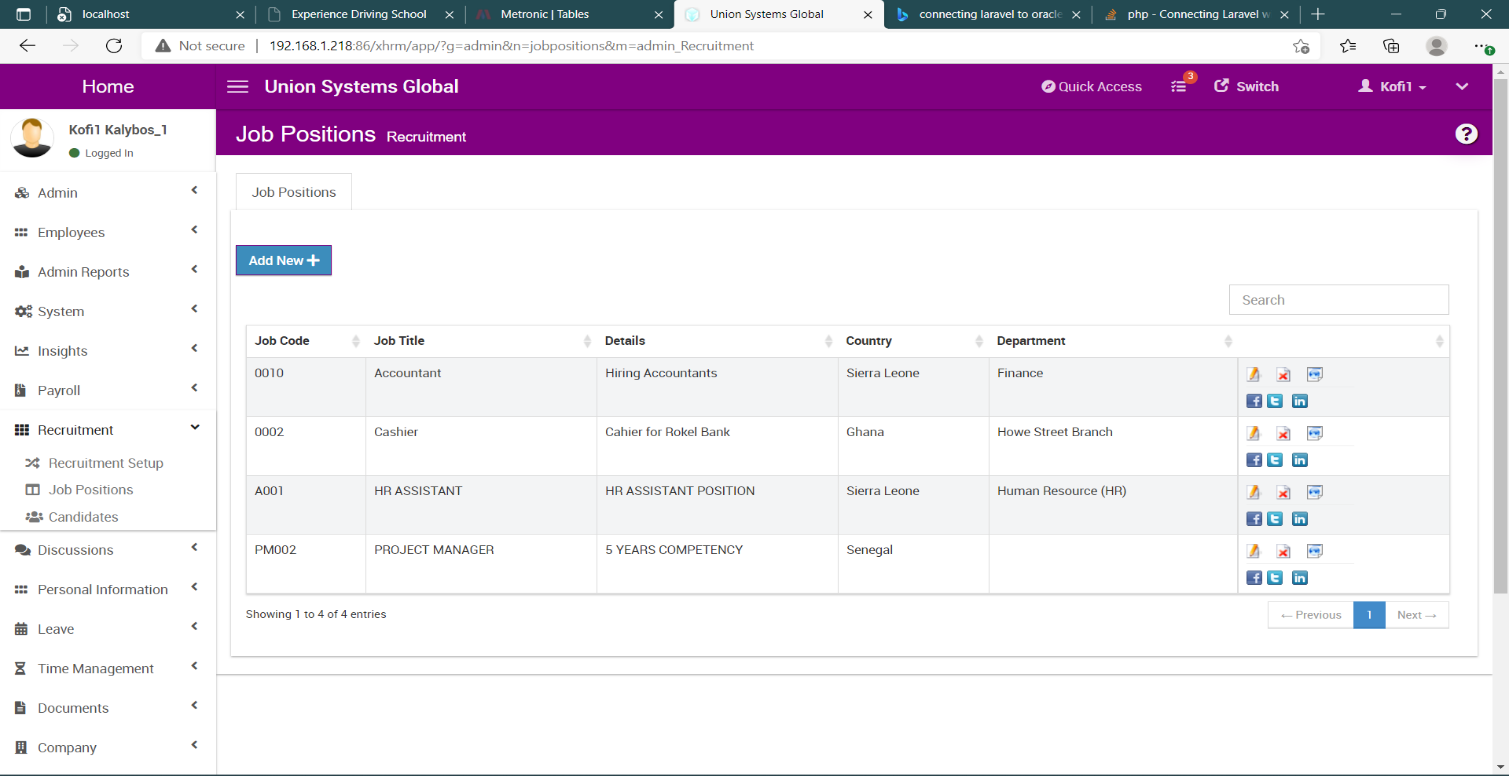
**DELETE RECRUITMENT PARAMETER**

A user can delete a parameter by clicking on the red crossed icon relating to a particular item to permanently remove a parameter, this applies to all the tabs



* **JOB POSITION**

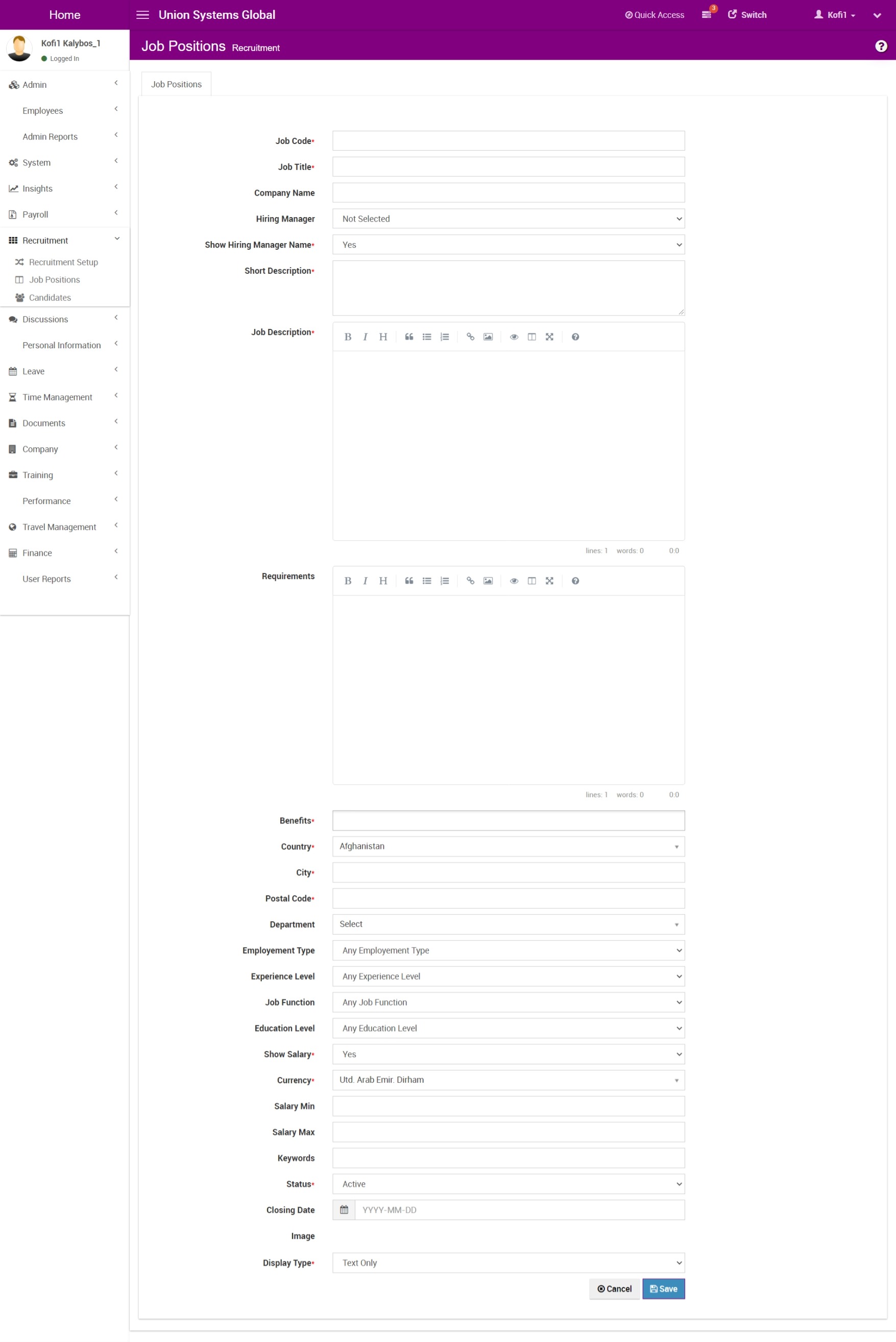
The job position setup, allows users to create, edit and delete new job positions.



**CREATION OF JOB POSITION**

A new job position can be created by clicking on the **ADD NEW BUTTON**, a job position form will be displayed.

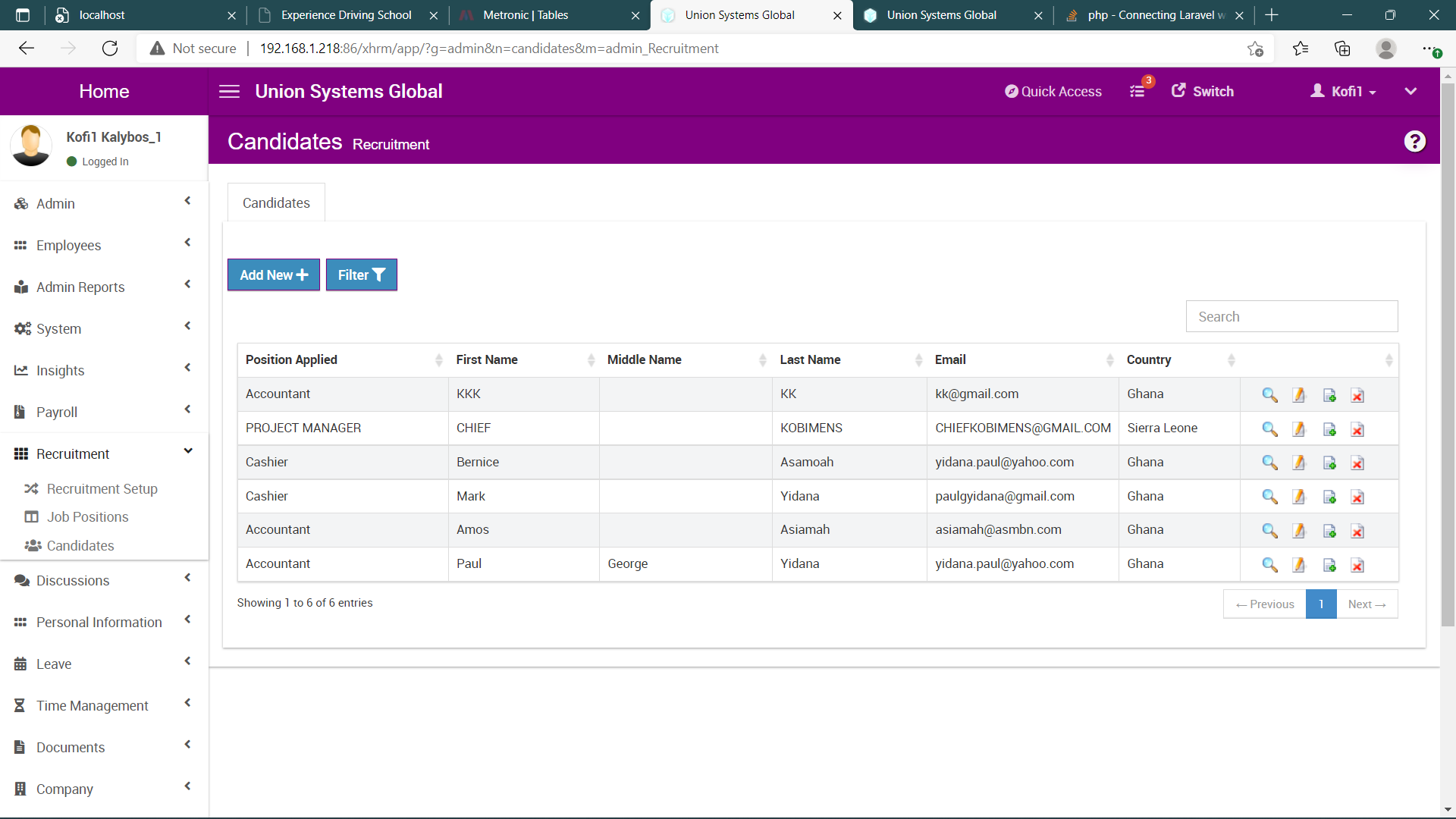
**JOB CREATION FORM**



When the necessary inputs are filled and the save button is clicked, a new job position will be added to the records.

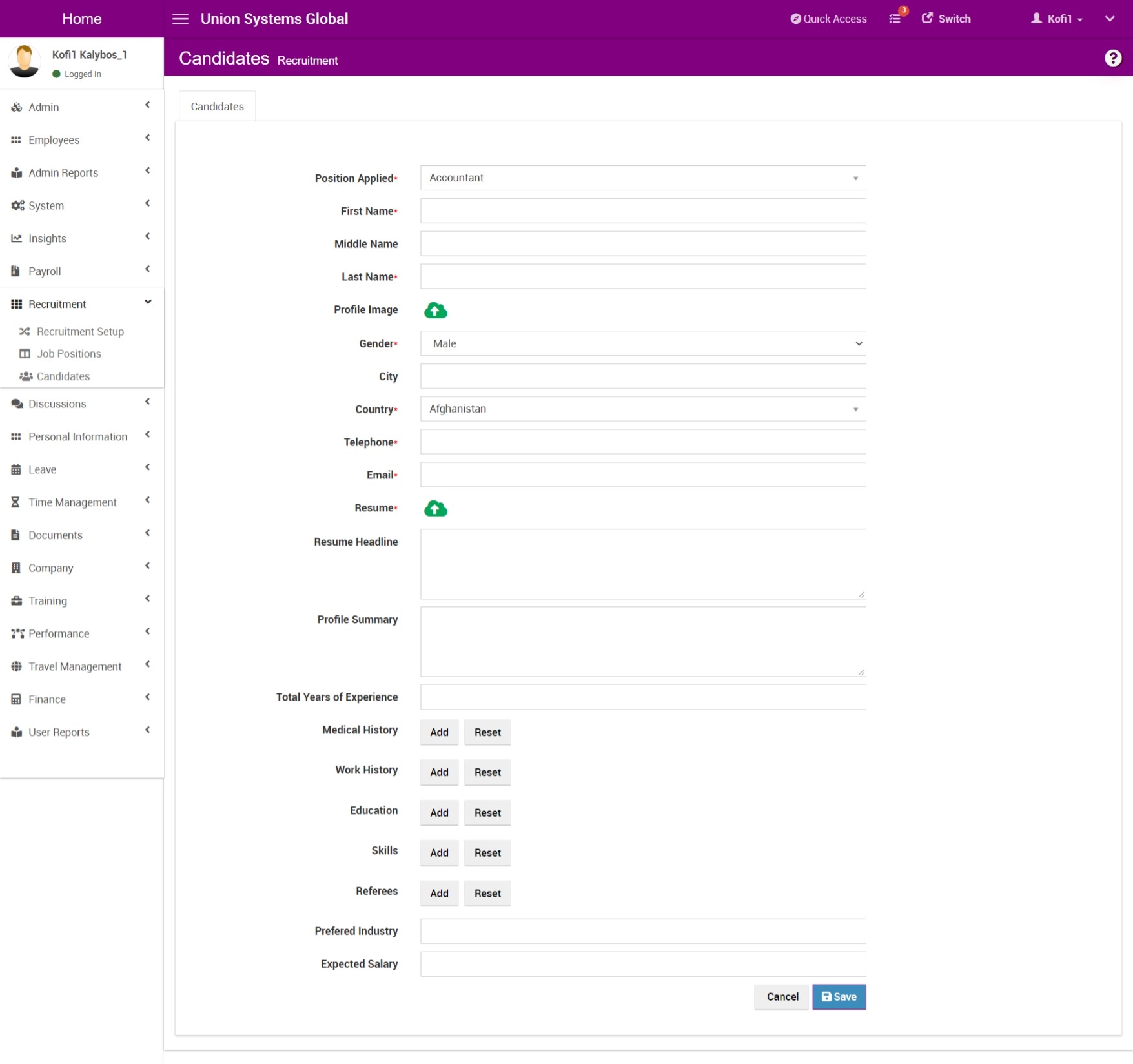
* **CANDIDATES**

The candidate module keeps records of all job applicants, scheduling an interview, recruiting a candidate as staff, candidates can also be edited and deleted



**CREATION OF CANDIDATE**

There are two ways of creating a candidate in the HR system, candidates can be created when a user fills a job application online or when the **ADD NEW BUTTON** is clicked.

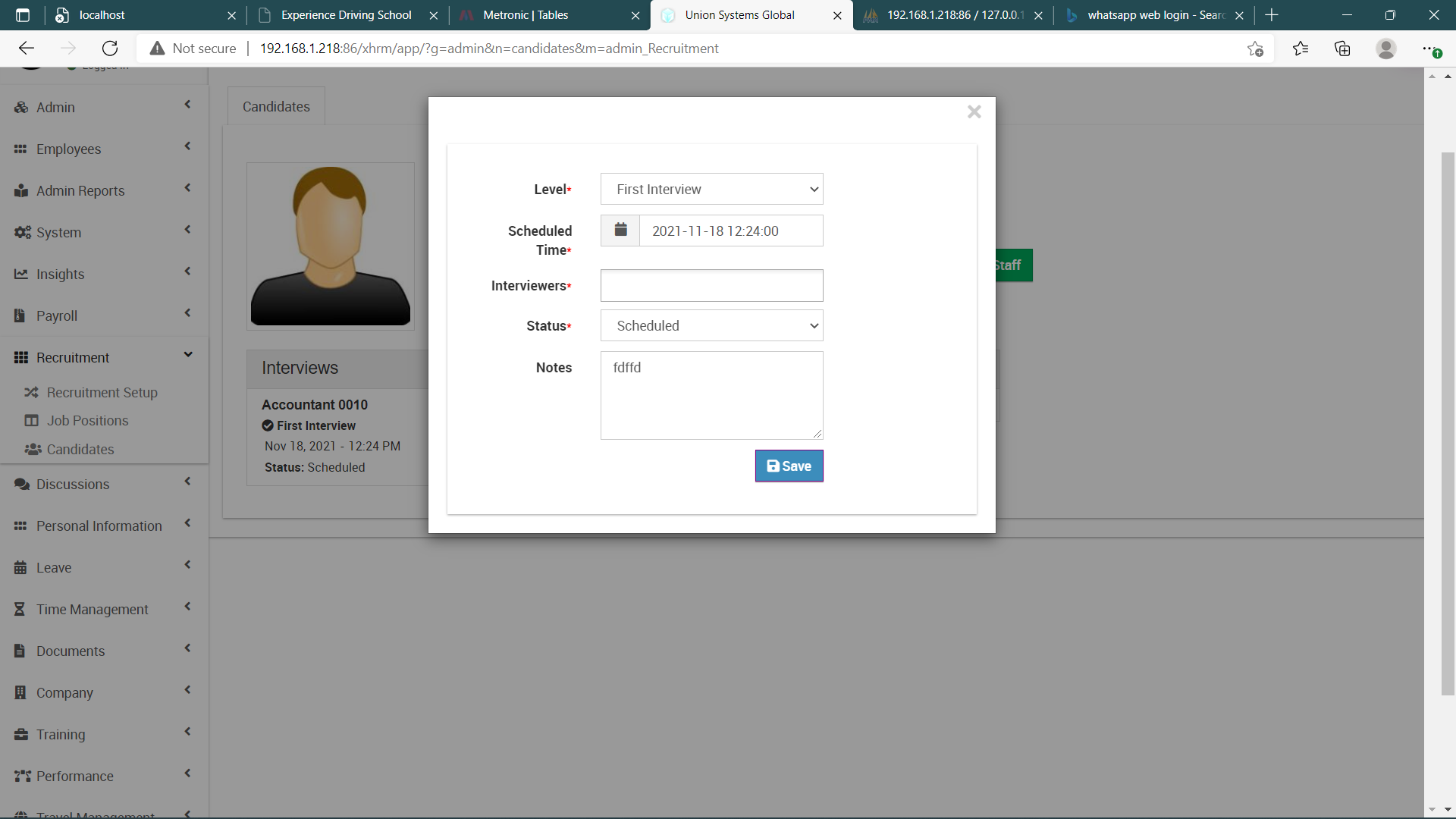


**RECRUIT AS STAFF**

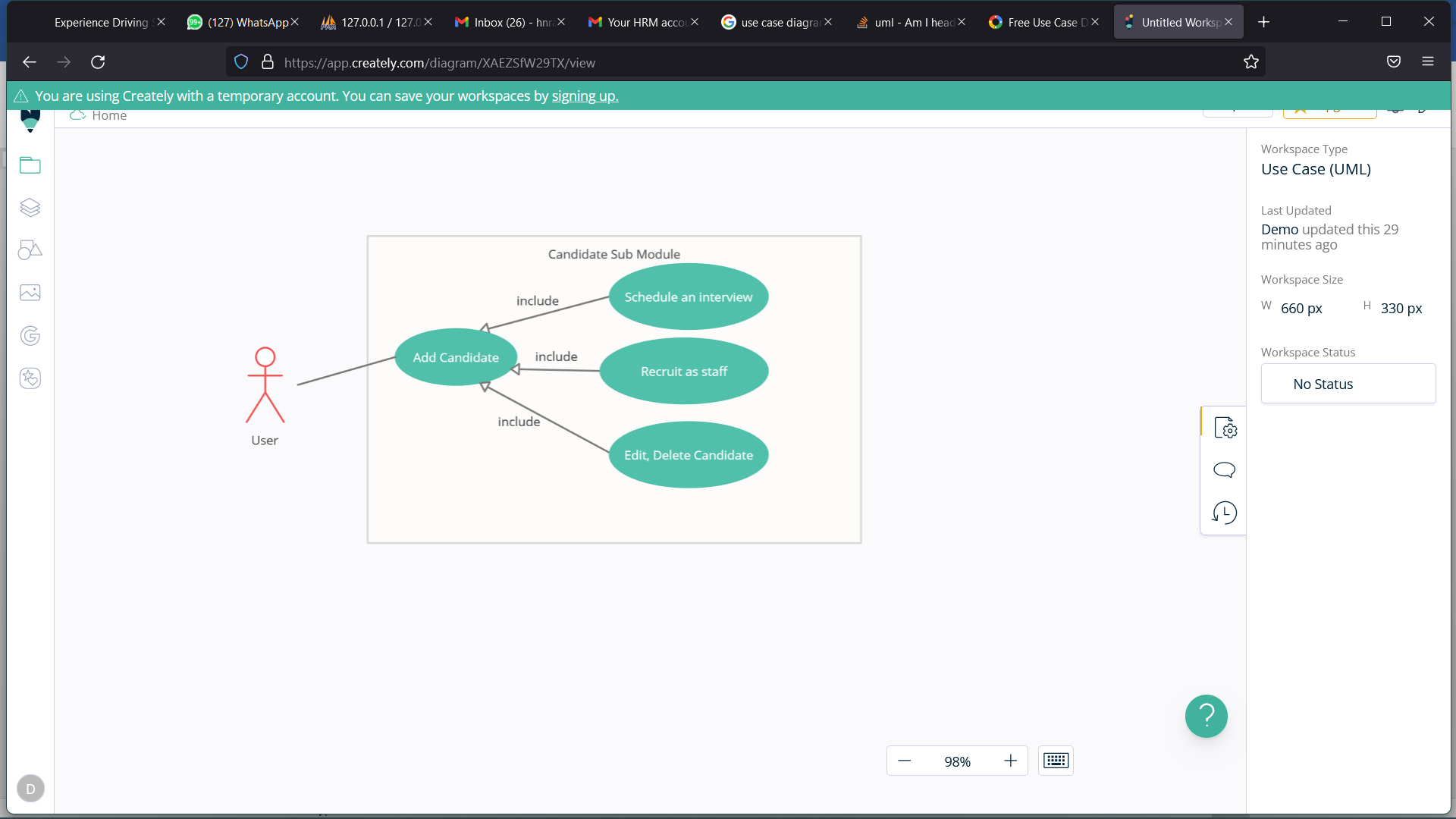
1. Click to view the profile of the candidate
2. Click on **RECRUIT AS STAFF** button to add the candidate as an employee

**SCHEDULING AN INTERVIEW**

1. Click to view the profile of the candidate
2. Click on **SCHEDULE INTERVIEW** button which will present a form for scheduling meeting
3. Once the user clicks on the save schedule button an email will be sent to the candidate with the necessary information about the interview

****

**A use case diagram of the candidate sub module**



**Candidate Table Structure**

|  |  |  |  |
| --- | --- | --- | --- |
| ***id*** | Bigint (20) | No |  |
| jobId | varchar(50) | Yes | NULL |
| first\_name | Varchar (100) | No |  |
| middle\_name | Varchar (50) | Yes | NULL |
| last\_name | Varchar (100) | No |  |
| nationality | Bigint (20) | Yes | NULL |
| birthday | datetime | Yes | NULL |
| gender | Enum ('Male', 'Female') | Yes | NULL |
| marital\_status | Enum ('Married', 'Single', 'Divorced', 'Widowed', 'Other') | Yes | NULL |
| address1 | Varchar (100) | Yes |  |
| address2 | Varchar (100) | Yes |  |
| city | Varchar (150) | Yes |  |
| country | Char (2) | Yes | NULL |
| province | Bigint (20) | Yes | NULL |
| postal\_code | Varchar (20) | Yes | NULL |
| email | Varchar (200) | Yes | NULL |
| home\_phone | Varchar (50) | Yes | NULL |
| mobile\_phone | Varchar (50) | Yes | NULL |
| cv\_title | Varchar (200) | No |  |
| cv | Varchar (150) | Yes | NULL |
| cvtext | text | Yes | NULL |
| industry | text | Yes | NULL |
| profileImage | Varchar (150) | Yes | NULL |
| head\_line | text | Yes | NULL |
| objective | text | Yes | NULL |
| work\_history | text | Yes | NULL |
| health\_history | text | Yes | NULL |
| education | text | Yes | NULL |
| skills | text | Yes | NULL |
| referees | text | Yes | NULL |
| linkedInUrl | Varchar (500) | Yes | NULL |
| linkedInData | text | Yes | NULL |
| totalYearsOfExperience | Int (11) | Yes | NULL |
| totalMonthsOfExperience | Int (11) | Yes | NULL |
| htmlCVData | longtext | Yes | NULL |
| generatedCVFile | Varchar (150) | Yes | NULL |
| created | datetime | Yes | NULL |
| updated | datetime | Yes | NULL |
| expectedSalary | Int (11) | Yes | NULL |
| preferedPositions | text | Yes | NULL |
| preferedJobtype | Varchar (60) | Yes | NULL |
| preferedCountries | text | Yes | NULL |
| tags | text | Yes | NULL |
| notes | text | Yes | NULL |
| calls | text | Yes | NULL |
| age | int(11) | Yes | NULL |
| hash | varchar(100) | Yes | NULL |
| linkedInProfileLink | varchar(250) | Yes | NULL |
| linkedInProfileId | varchar(50) | Yes | NULL |
| facebookProfileLink | varchar(250) | Yes | NULL |
| facebookProfileId | varchar(50) | Yes | NULL |
| twitterProfileLink | varchar(250) | Yes | NULL |
| twitterProfileId | varchar(50) | Yes | NULL |
| googleProfileLink | varchar(250) | Yes | NULL |
| googleProfileId | varchar(50) | Yes | NULL |
| hiringStage | bigint(20) | Yes | NULL |
| source | Enum ('Sourced', 'Applied') | Yes | Sourced |
| emailSent | int(11) | Yes | 0 |

* **EMPLOYEE MODULE**

The employee module has seven (7) different sections

1. Employee
2. Employee History
3. Document Management
4. Imprest Request
5. Monitor Attendance
6. HR Form Management
7. Performance Reviews

* **EMPLOYEE SECTION**

This section manages all employees of the company, it handles creating, approving, editing and deleting of employees

**CREATE EMPLOYEE**

On this screen, new employees can be added by clicking on the **ADD NEW** button, after providing the necessary details and clicking on the save button a new employee will be created, once an employee is created it has to go for approval before employee becomes an active employee

* **APPROVING AN EMPLOYEE**

To approve an employee, click to view the profile of the particular employee and then click the **APPROVE** button to make the employee active. Approval can be done by user with admin level privileges.

**EDITING EMPLOYEE**

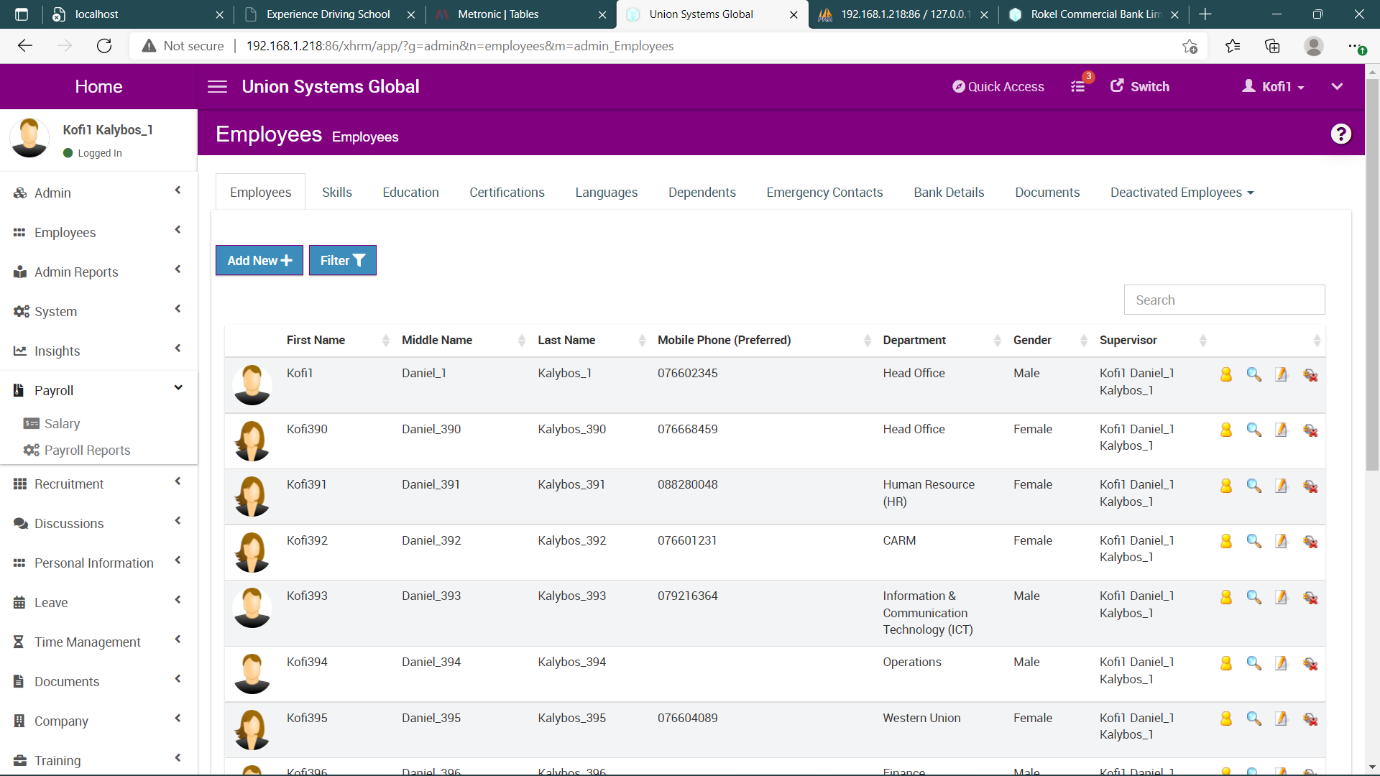
To make changes to an employee

1. Visit the profile of the employee and click on the edit button to make changes to the employee
2. On the employee grid, displaying all the employees, click on the edit icon relating to the specific employee you want to edit and make the necessary changes.

**DELETING EMPLOYEE**

To delete an employee, on the employee grid, displaying all the employees, click on the delete icon relating to the specific employee you want to delete.

**EMPLOYEE SECTION SCREEN**

****

The additional tabs (Skills, Education, Certifications, Languages, Dependents, Emergency Contacts, Bank Details, Documents, Deactivated Employees) on the employee screen allows the user to add information relating to the specific tab by clicking on the **ADD NEW** button, except the **Deactivated Employee** tab.

**DEACTIVATED EMPLOYEE TAB**

The is only for viewing Temporarily deactivated employees and Terminated employees

**EMPLOYEE DATABASE STRUCTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| employee\_id | varchar(50) | Yes | NULL |
| title | varchar (50) | Yes | NULL |
| first\_name | varchar(100) | No |  |
| middle\_name | varchar(100) | Yes | NULL |
| last\_name | varchar(100) | Yes | NULL |
| nationality | varchar(20) | Yes | NULL |
| religion | varchar(50) | Yes | NULL |
| birthday | date | No |  |
| place\_of\_birth | varchar(50) | No |  |
| spouse\_name | varchar(50) | Yes | NULL |
| father\_name | varchar(50) | Yes | NULL |
| mother\_name | varchar(50) | Yes | NULL |
| retirement\_date | date | Yes | NULL |
| gender | enum('Male', 'Female') | Yes | NULL |
| marital\_status | enum('Married', 'Single', 'Divorced', 'Widowed', 'Other') | Yes | NULL |
| nxt\_kin\_name | varchar(20) | Yes | NULL |
| nxt\_kin\_email | varchar(20) | Yes | NULL |
| nxt\_kin\_address | varchar(20) | Yes | NULL |
| nxt\_kin\_phone | varchar(100) | Yes | NULL |
| bank\_name | varchar(20) | Yes | NULL |
| bank\_acc\_no | varchar(100) | Yes | NULL |
| tin\_no | varchar(25) | Yes | NULL |
| staff\_level | varchar(20) | Yes | NULL |
| ssn\_num | varchar(100) | Yes | NULL |
| nassit\_num | varchar(100) | Yes | NULL |
| nic\_num | varchar(100) | Yes | NULL |
| other\_id | varchar(100) | Yes | NULL |
| driving\_license | varchar(100) | Yes | NULL |
| driving\_license\_exp\_date | date | Yes | NULL |
| employment\_status | bigint(20) | Yes | NULL |
| job\_title | bigint(20) | Yes | NULL |
| pay\_grade | bigint(20) | Yes | NULL |
| notches | varchar(50) | Yes | NULL |
| work\_station\_id | varchar(100) | Yes | NULL |
| address1 | varchar(100) | Yes | NULL |
| address2 | varchar(100) | Yes | NULL |
| city | varchar(150) | Yes | NULL |
| country | char(2) | Yes | NULL |
| province | bigint(20) | Yes | NULL |
| postal\_code | varchar(20) | Yes | NULL |
| home\_phone | varchar(50) | Yes | NULL |
| mobile\_phone | varchar(50) | Yes | NULL |
| work\_phone | varchar(50) | Yes | NULL |
| work\_email | varchar(100) | Yes | NULL |
| private\_email | varchar(100) | Yes | NULL |
| recruitment\_date | date | Yes | NULL |
| confirmation\_date | date | Yes | NULL |
| supervisor | bigint(20) | Yes | NULL |
| indirect\_supervisors | varchar(250) | Yes | NULL |
| department | bigint(20) | Yes | NULL |
| branch | varchar(50) | No |  |
| unit | varchar(50) | Yes | NULL |
| start\_date | date | Yes | NULL |
| probation\_period | bigint(250) | Yes | NULL |
| initials | varchar(20) | Yes | NULL |
| severance\_benefit | varchar(250) | Yes | NULL |
| custom5 | varchar(250) | Yes | NULL |
| custom6 | varchar(250) | Yes | NULL |
| custom7 | varchar(250) | Yes | NULL |
| custom8 | varchar(250) | Yes | NULL |
| custom9 | varchar(250) | Yes | NULL |
| custom10 | varchar(250) | Yes | NULL |
| termination\_date | date | Yes | NULL |
| notes | text | Yes | NULL |
| status | enum('Active', 'Terminated') | Yes | Active |
| ethnicity | bigint(20) | Yes | NULL |
| immigration\_status | bigint(20) | Yes | NULL |
| approver1 | bigint(20) | Yes | NULL |
| approver2 | bigint(20) | Yes | NULL |
| approver3 | bigint(20) | Yes | NULL |
| posting\_date | date | Yes | current\_timestamp() |

* **EMPLOYEE HISTORY**

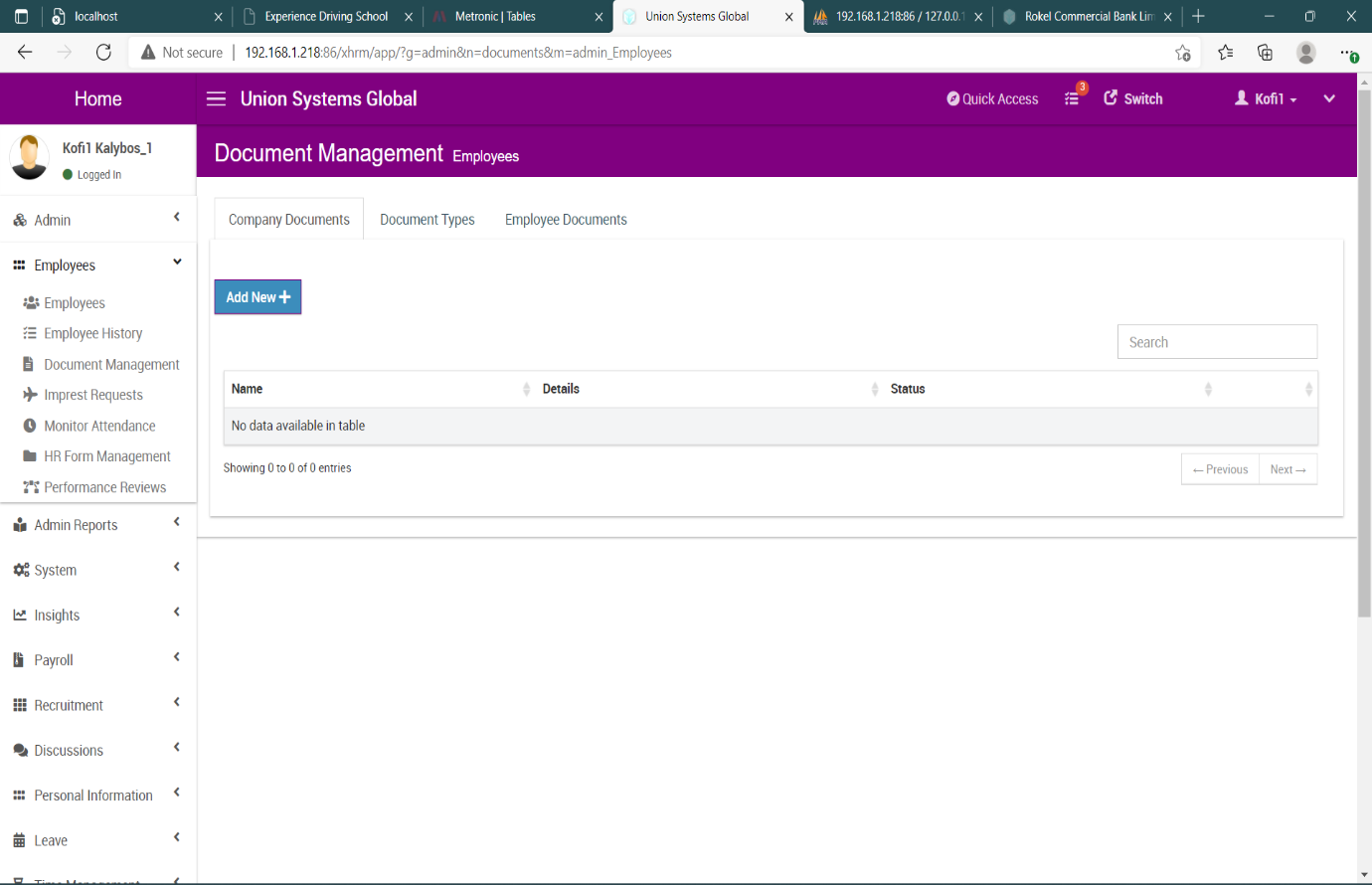
Employee history shows changes that has been made on employees

* **DOCUMENT MANAGEMENT**

The document management module takes care of storing and sharing documents in the hr system, there three (3) sections in this module

1. Company Document
2. Document Types
3. Employee Documents

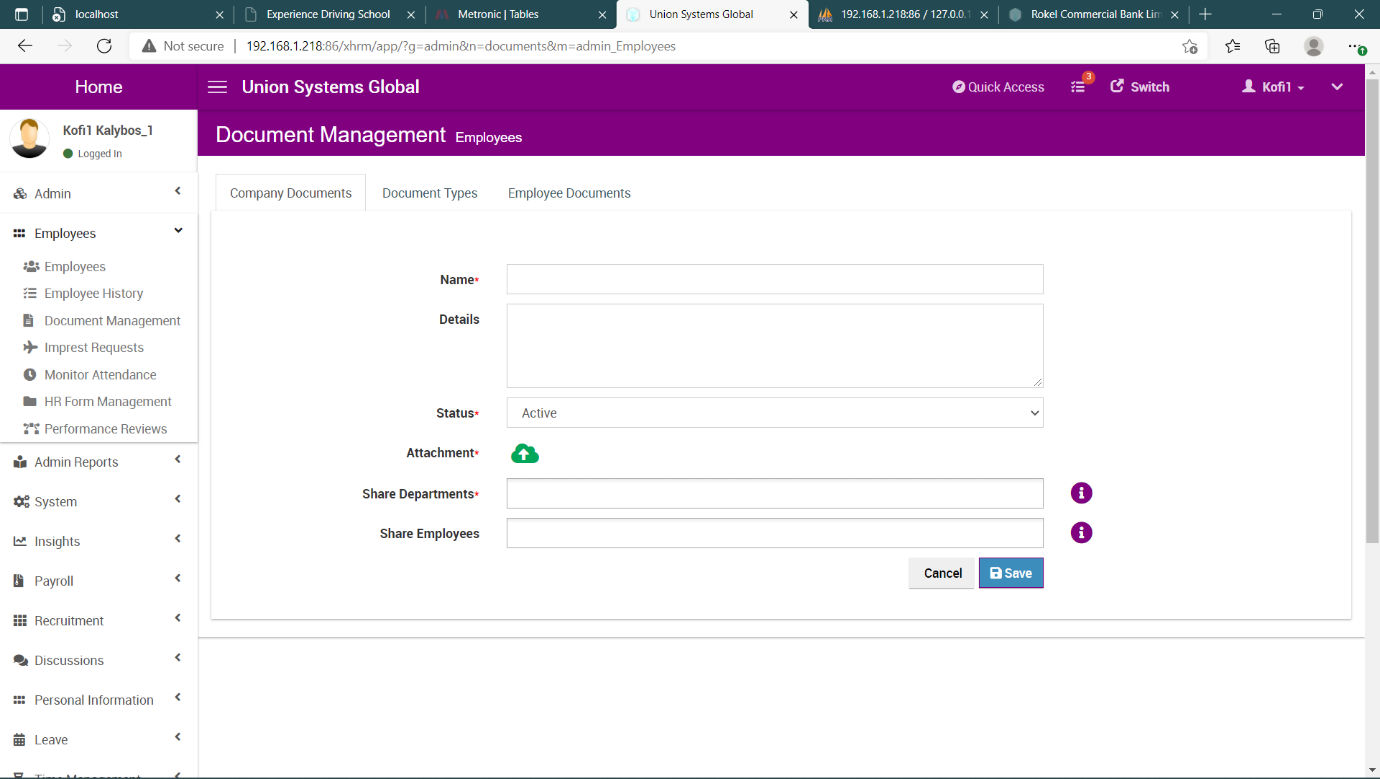
**DOCUMENT MANAGEMENT MODULE**



* **Company Document**

The company document section keeps records all documents for the company, to add a document

1. Move to the company document session
2. Click on add new, to display the add company document form



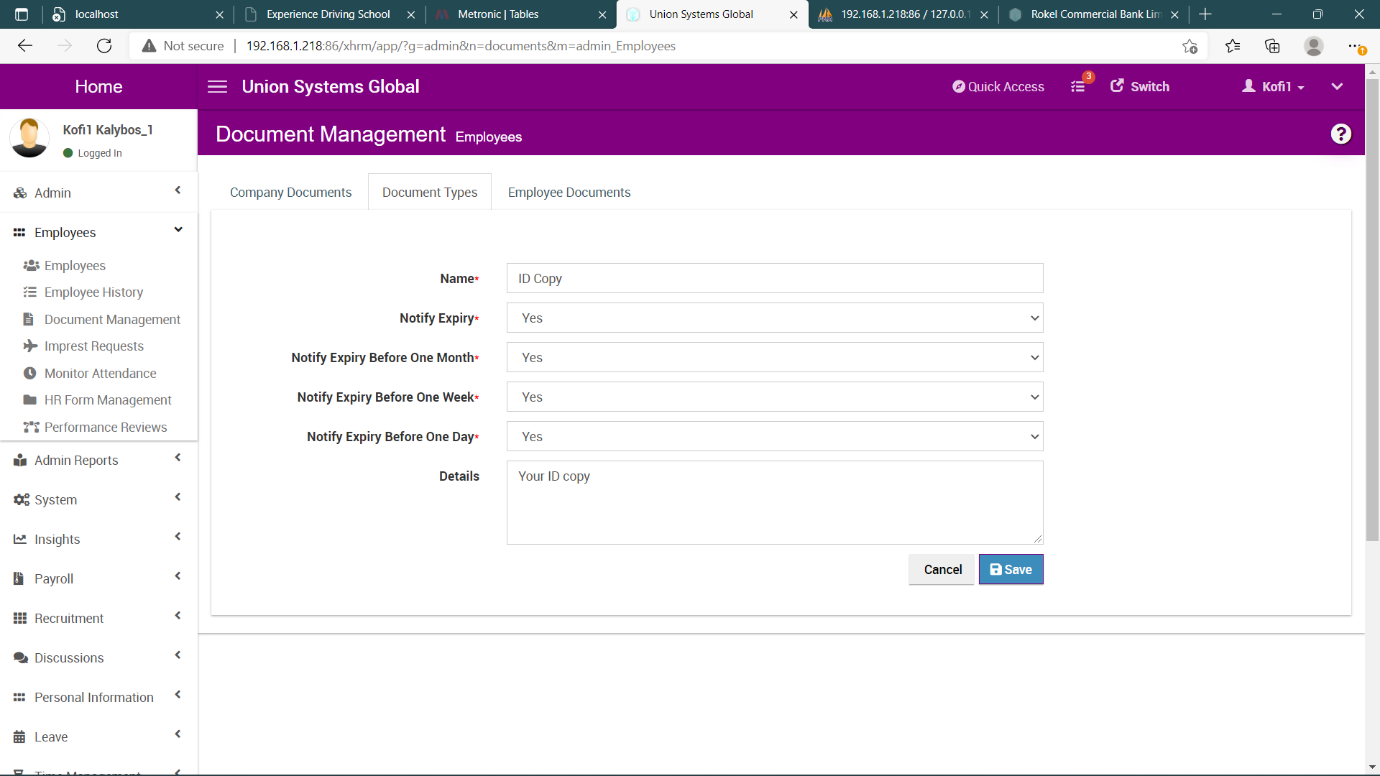
1. Provide the necessary details and click on the **SAVE** button to add document to the database

Documents added can be shared among departments and employees by selecting the departments and employees you want the document to be shared with

* **Document Types**

The document type section allows users to create categories for documents, the user can also set notifications for document expiry, to add a document

1. Move to the document type section
2. Click on the **ADD NEW** button, to display the document types form



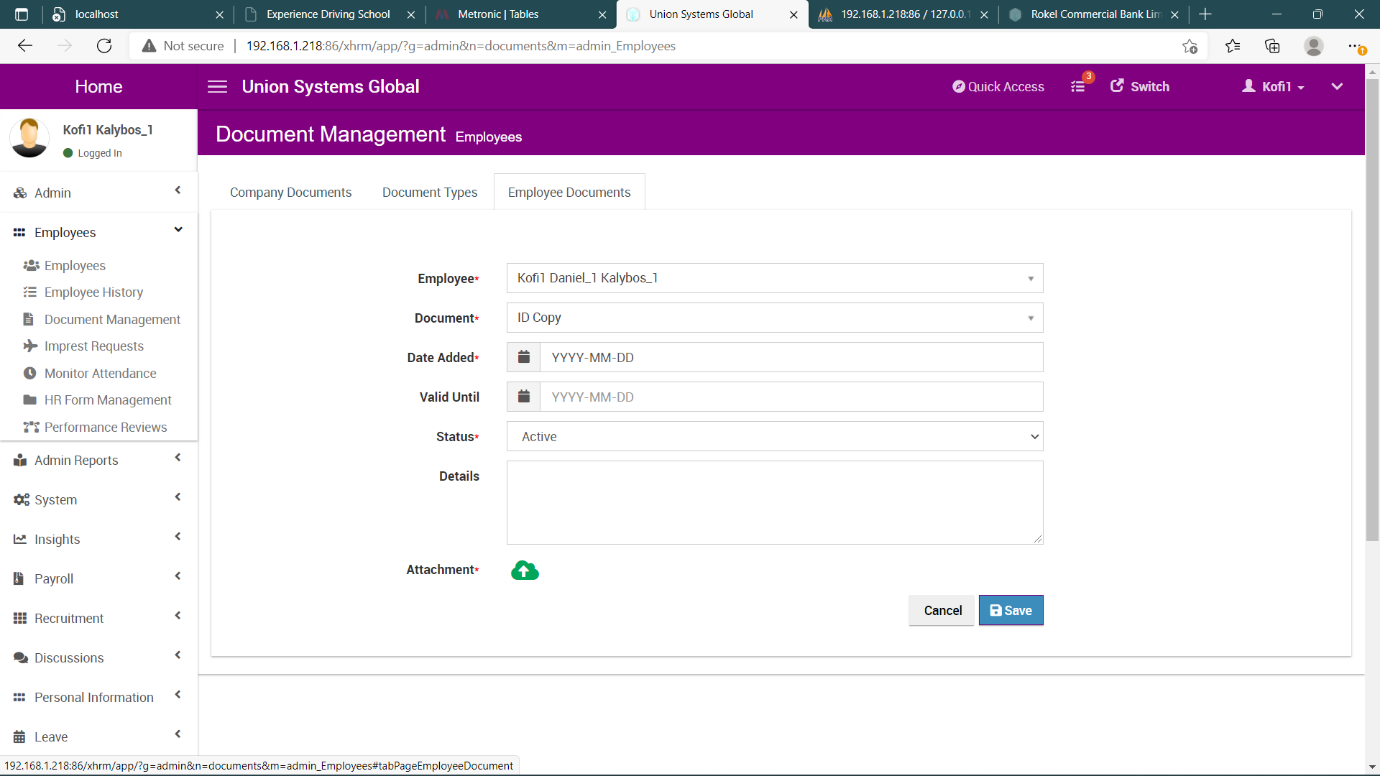
1. Provide the necessary details and click on the **SAVE** button to add a new document type

Document types can be edited and deleted.

* **Employee Documents**

The employee documents section allows users with admin level privileges to add documents for employees, the documents added for an employee will be shared with the employee, to add an employee document

1. Move to the employee documents section
2. Click on the **ADD NEW** button, to display the employee document form

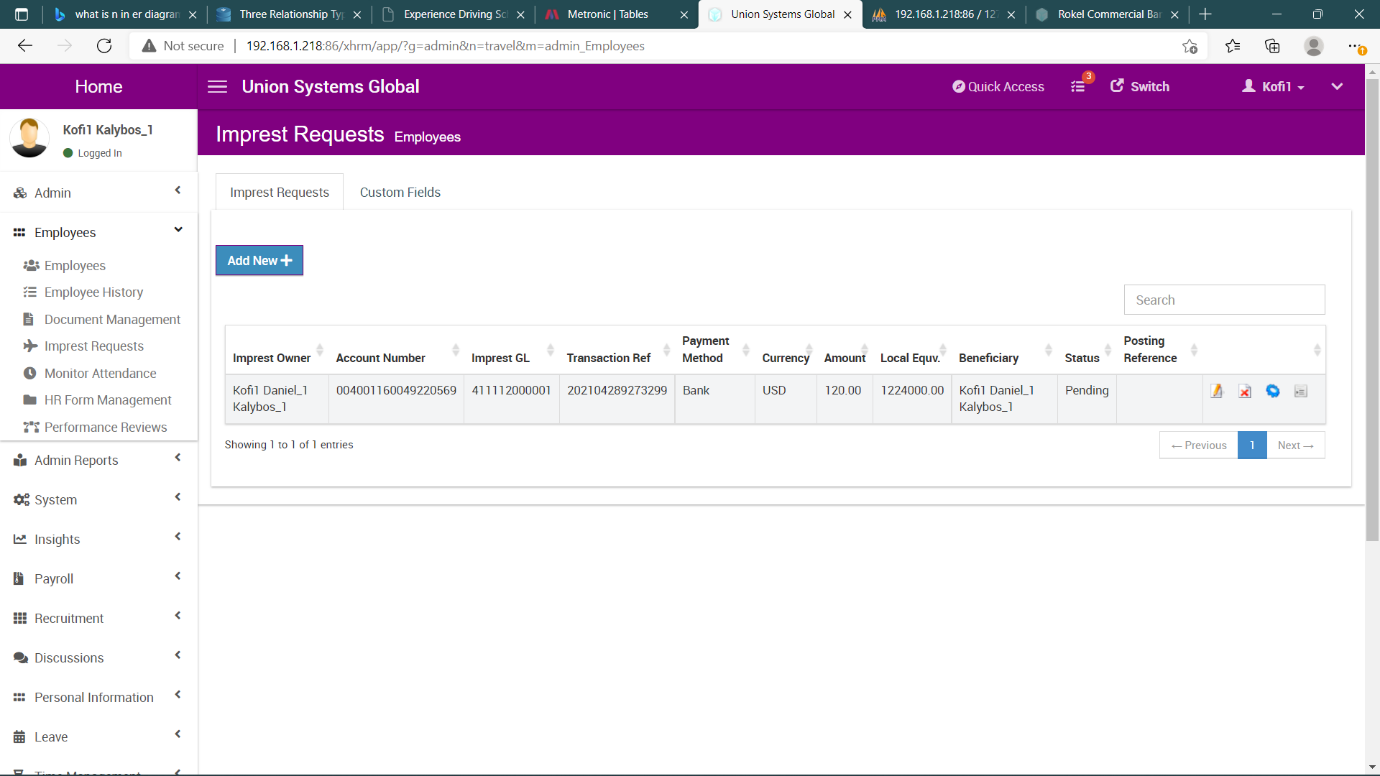


1. Provide the necessary details and click on the **SAVE** button to add an employee document

* **IMPREST REQUEST**

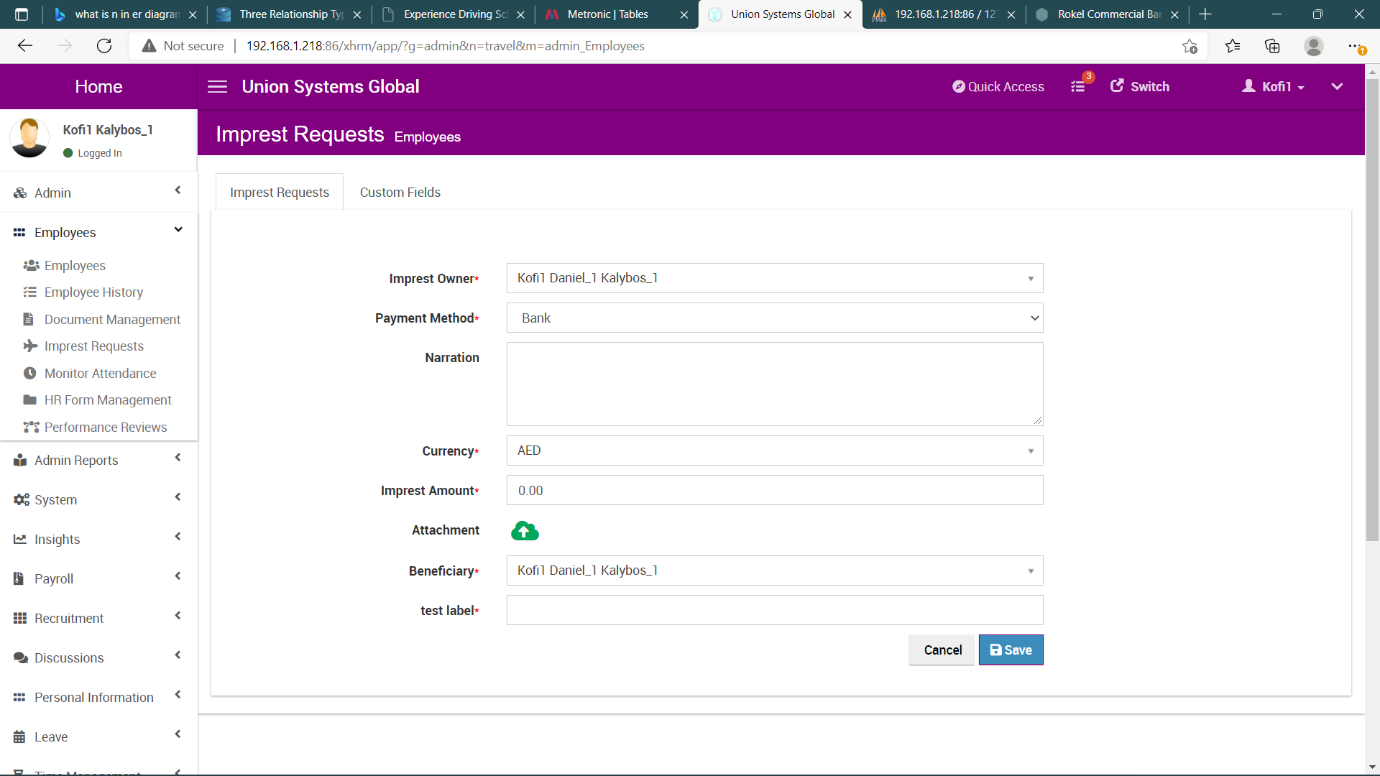
The imprest request module allows users to make advance request for travel expenses.

**IMPREST REQUEST SCREEN**



**TO REQUEST FOR AN IMPREST:**

1. Click on **ADD NEW** button, to display the imprest form



1. Provide the necessary details and click the **SAVE** button to request the imprest.

* **IMPREST APPROVAL**

Imprest approval are done by only managers of employees, to approve an imprest go the imprest screen and click on approve icon relating to the imprest you want to approve

**IMPREST DATABASE STRUCTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| employee | bigint(20) | No |  |
| imprest\_gl | varchar(50) | Yes | NULL |
| type | varchar(200) | Yes |  |
| purpose | varchar(200) | Yes | NULL |
| ref\_no | varchar(20) | Yes | NULL |
| emp\_acc\_no | varchar(50) | No |  |
| cost\_unit | varchar(50) | Yes | NULL |
| cost\_dept | varchar(50) | Yes | NULL |
| branch | varchar(10) | Yes | NULL |
| xcg\_rate | decimal(20,2) | Yes | 0.00 |
| imprest\_contra | varchar(50) | Yes | NULL |
| travel\_from | varchar(200) | Yes | NULL |
| travel\_to | varchar(200) | Yes | NULL |
| travel\_date | datetime | Yes | NULL |
| return\_date | datetime | Yes | NULL |
| details | varchar(500) | Yes | NULL |
| payment\_method | varchar(50) | Yes | NULL |
| funding | decimal(10,2) | Yes | NULL |
| local\_eqv | decimal(30,2) | Yes | NULL |
| currency | varchar(20) | Yes | GHS |
| attachment1 | varchar(100) | Yes | NULL |
| attachment2 | varchar(100) | Yes | NULL |
| attachment3 | varchar(100) | Yes | NULL |
| created | datetime | Yes | NULL |
| updated | datetime | Yes | NULL |
| beneficiary | bigint(20) | Yes | NULL |
| beneficiary\_acc | varchar(50) | Yes | NULL |
| app\_type | varchar(10) | No | IMP |
| posting\_date | datetime | Yes | current\_timestamp() |
| status | enum('Approved', 'Pending', 'Rejected', 'Cancellation Requested', 'Cancelled', 'Processing') | Yes | Pending |

* **MONITOR ATTENDANCE**

The monitor attendance section helps to manage the attendance of employees, the monitor attendance section is made up of

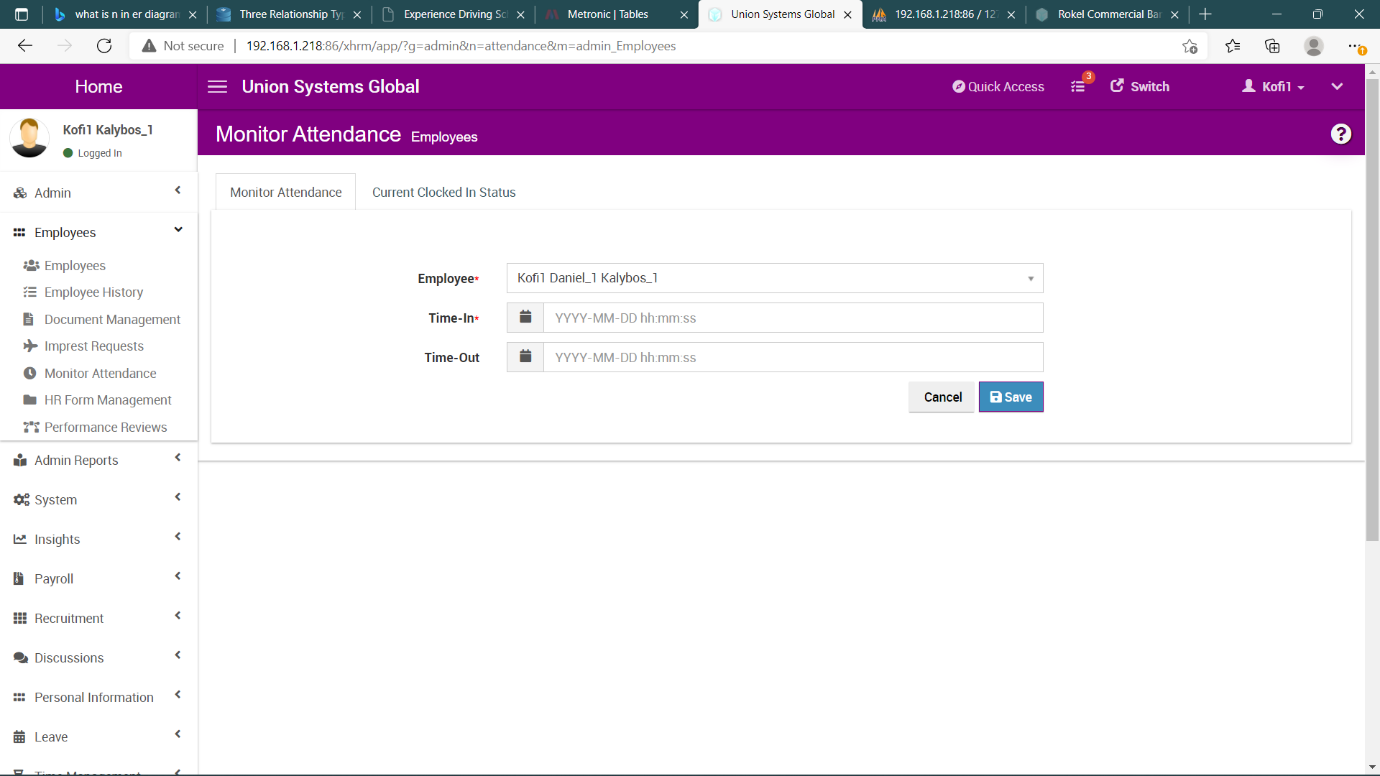
1. Monitor attendance
2. Current clocked in status

* **MONITOR ATTENDANCE**

The monitor attendance allows users to clock in, this can be done using the biometric time attendance machine or through manual entry

To manually clock in employees,

1. Click on **ADD NEW** button, to display the clock in form



1. Select the employee and add the time in (arrival time) for the user, this can be updated later to add the time out (departure time) for the user.
2. Click on the save button, to add the attendance data for the user.

* **CURRENT CLOCKED IN STATUS**

This section displays all employees who have been clocked in and those who have not clocked in for today

**ATTENDANCE TABLE STRUCTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| employee | bigint(20) | No |  |
| department | varchar(100) | No |  |
| isInvalid | int(20) | Yes | NULL |
| iAttState | int(20) | Yes | NULL |
| iVerifyMethod | int(20) | Yes | NULL |
| date | date | Yes | NULL |
| time | time | Yes | NULL |
| time\_stamp | timestamp | No | current\_timestamp() |
| in\_time | datetime | Yes | NULL |
| out\_time | datetime | Yes | NULL |
| note | varchar(500) | Yes | NULL |
| image\_in | longtext | Yes | NULL |
| image\_out | longtext | Yes | NULL |
| map\_lat | decimal(10,8) | Yes | NULL |
| map\_lng | decimal(10,8) | Yes | NULL |
| map\_snapshot | longtext | Yes | NULL |
| map\_out\_lat | decimal(10,8) | Yes | NULL |
| map\_out\_lng | decimal(10,8) | Yes | NULL |
| map\_out\_snapshot | longtext | Yes | NULL |
| in\_ip | varchar(25) | Yes | NULL |
| out\_ip | varchar(25) | Yes | NULL |

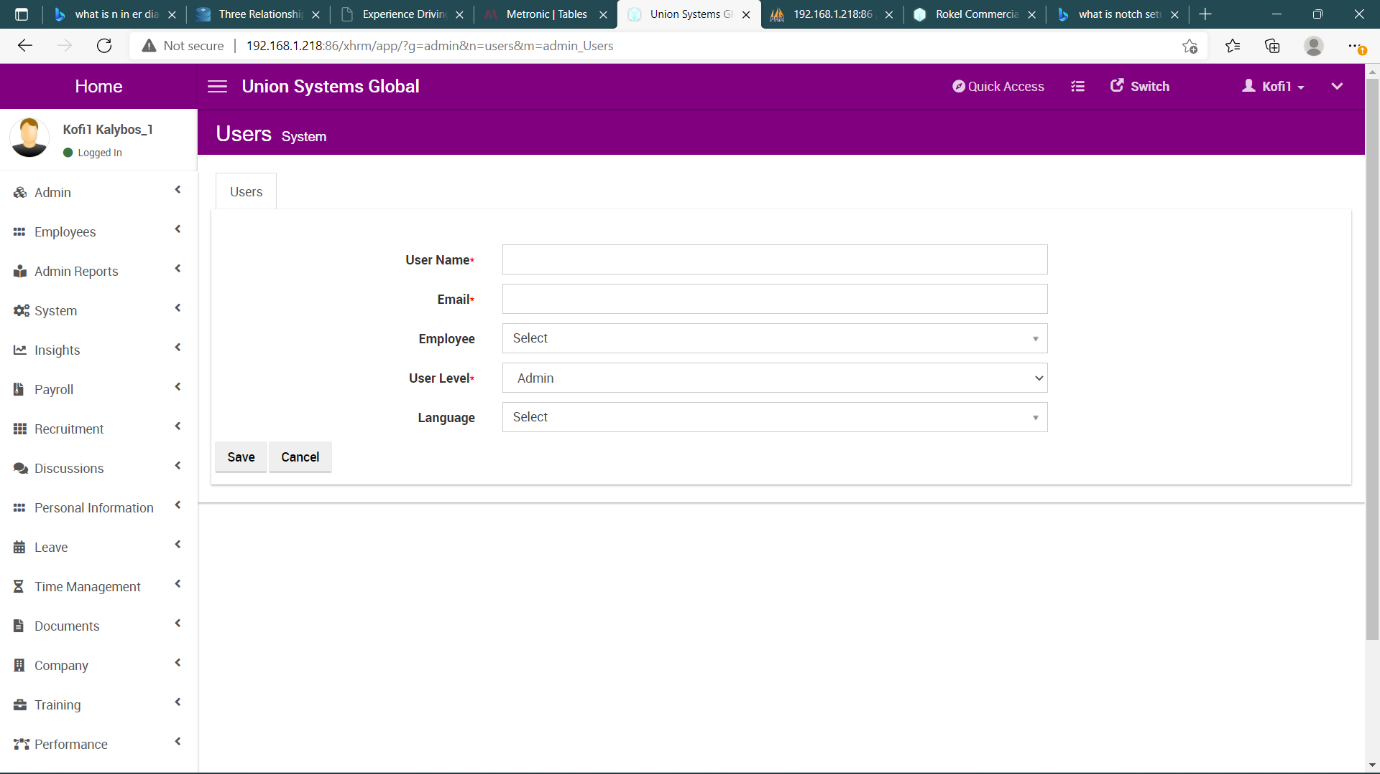
* **USERS**

The user module handles the creation of users of the system.

* **USER CREATION**

To add a user to the system

1. In the user module click on **ADD NEW** button, to display the user creation form



1. Provide the necessary details and click the save button to add a new user.
2. When a user is created a password will be sent to the users’ email to login into the system

* **ADMIN**

The admin module has 13 different sections

1. Dashboard
2. Company Structure
3. Job Details Setup
4. Qualification setup
5. Training Setup
6. Project/Client Setup
7. Leave Settings
8. Expense Administration
9. Overtime Administration
10. Company Loans
11. Company Assets
12. Audit Log

* **DASHBOARD**

The dashboard sends the user to the main page of the hr system

* **COMPANY STRUCTURE**

The company structure section displays the structure company, the head office and the branches

* **JOB SETUP DETAILS**

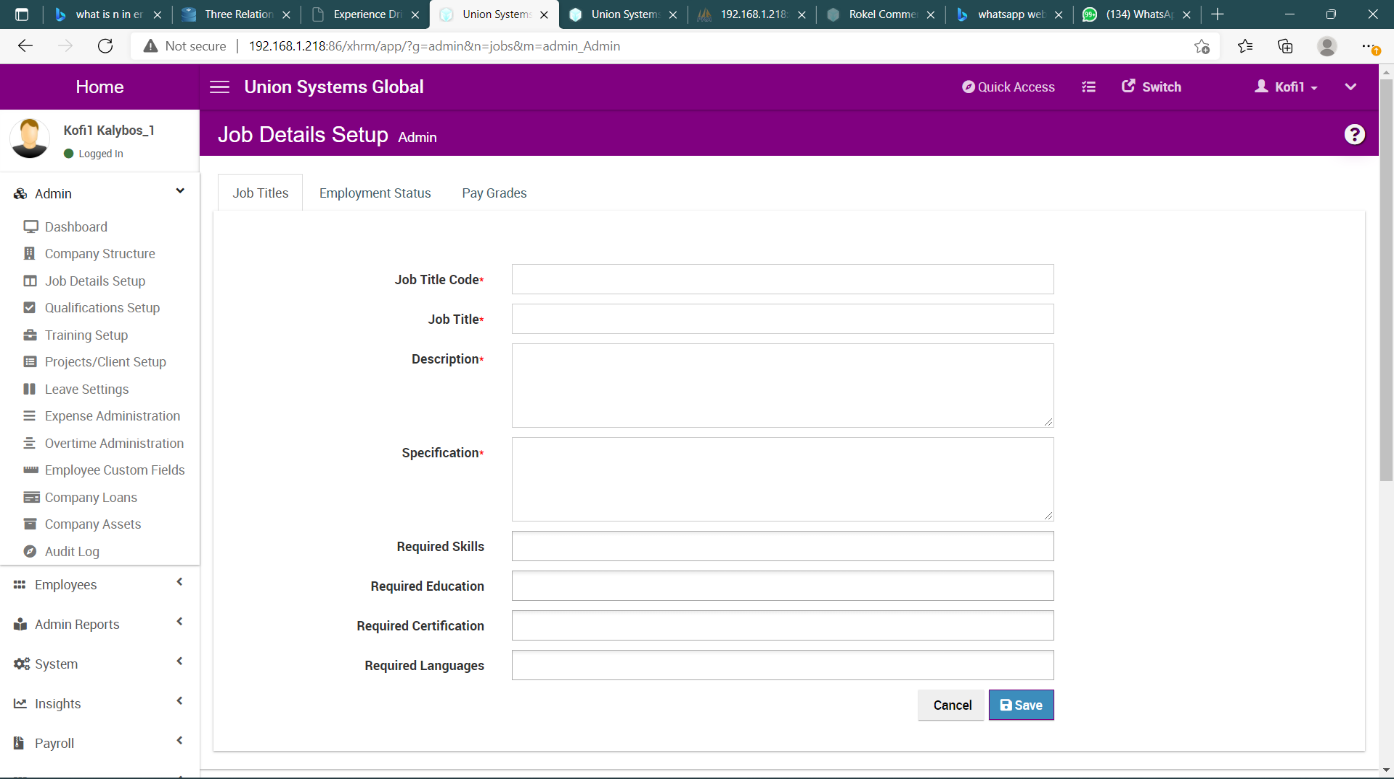
The job setup details section allows the admin user to setup parameters for the job module, it has three (3) setup parameters

1. Job Titles
2. Employment Status
3. Pay Grades

**JOB TITLES**

To add job titles,

1. Move to the job titles section
2. Click on the **ADD NEW** button, to display the job title form



1. Provide the necessary details and click the **SAVE** button to add the job title.

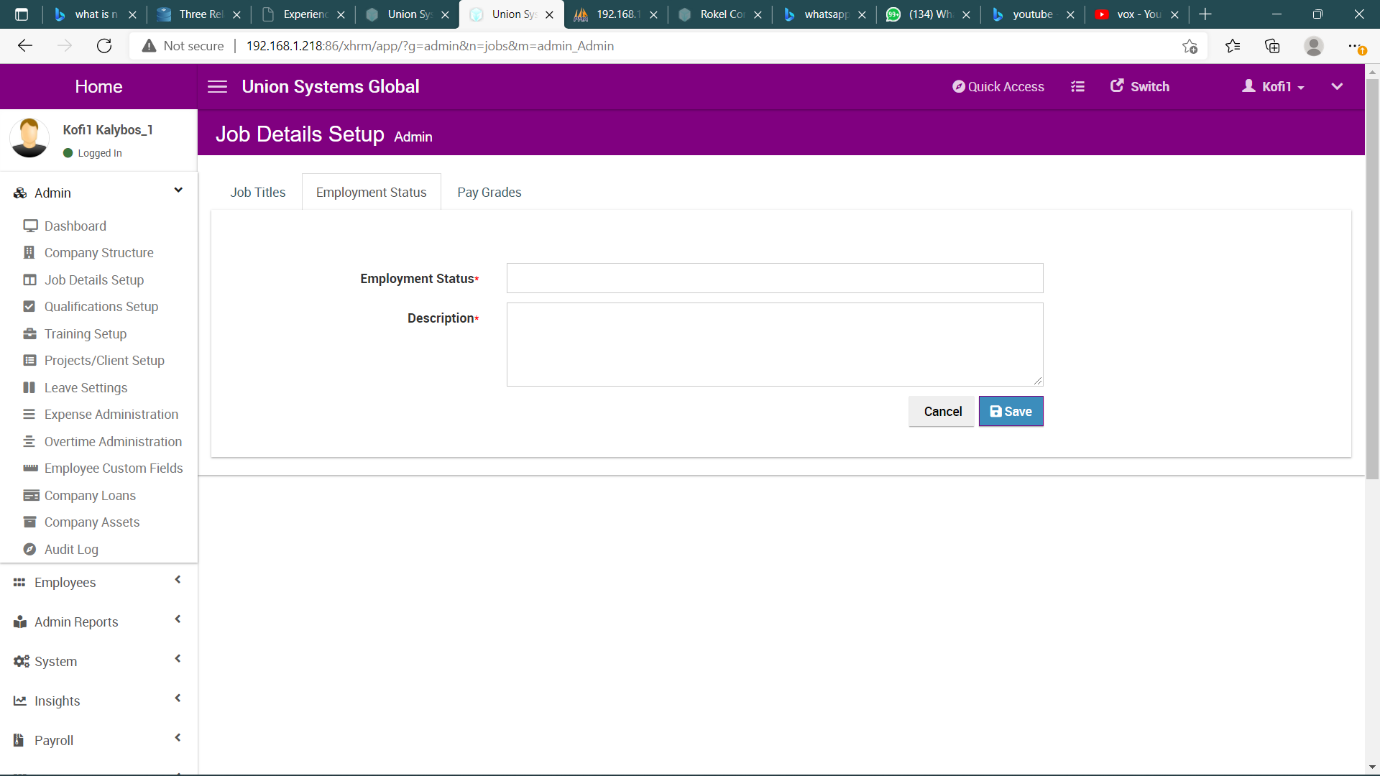
**JOB TITLES TABLE STRUCTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| **code** | varchar(10) | Yes | NULL |
| name | varchar(100) | Yes | NULL |
| category | varchar(30) | Yes | NULL |
| description | varchar(200) | Yes | NULL |
| specification | varchar(400) | Yes | NULL |

**EMPLOYMENT STATUS**

To add employment status,

1. Move to the employment status section
2. Click on the **ADD NEW** button, to display the employment form



1. Click on the **SAVE** button to add employment status

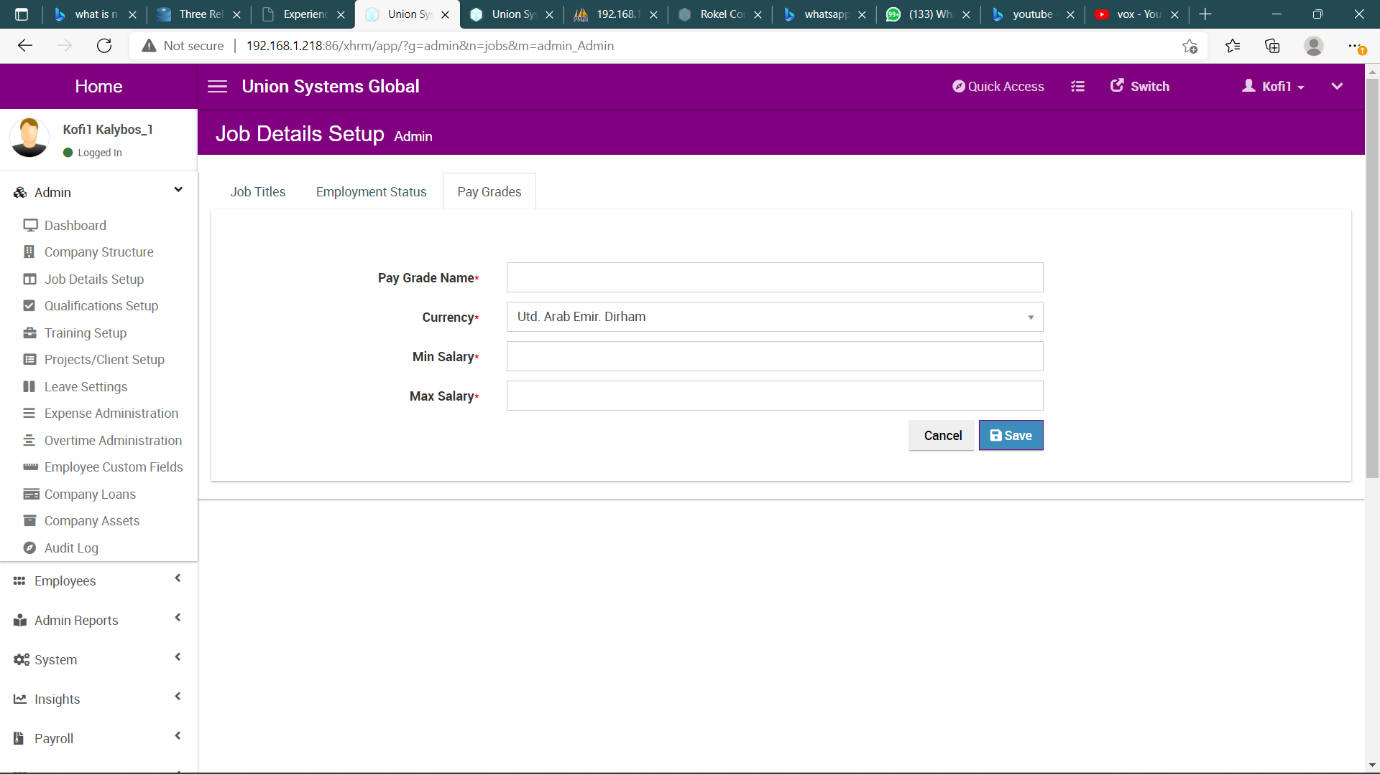
**EMPLOYMENT STATUS TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| name | varchar(100) | Yes | NULL |
| description | varchar(400) | Yes | NULL |

**PAY GRADES**

To add a pay grade parameter,

1. Move to the pay grade section,
2. Click on the **ADD NEW** button, to display the forms



1. Provide the necessary details and save the parameter

**PAY GRADES TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| name | varchar(100) | Yes | NULL |
| currency | varchar(3) | No |  |
| min\_salary | decimal(12,2) | Yes | 0.00 |
| max\_salary | decimal(12,2) | Yes | 0.00 |

* **QUALIFICATION SETUP**

The qualification setup section allows the admin user to setup qualification parameters for the recruitment module, it has four (4) setup parameters

1. Skills
2. Education
3. Certifications
4. Languages

To add a parameter for any of the setups

1. Move to that section
2. Click on the **ADD NEW** button, to display a form for the data entry
3. Click on the **SAVE** button save the data

**SKILLS TABLE STRUCTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| name | varchar(100) | Yes | NULL |
| description | varchar(400) | Yes | NULL |

**EDUCATION, CERTIFICATIONS, LANGUAGES TABLE STRUCTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| name | varchar(100) | Yes | NULL |
| description | varchar(400) | Yes | NULL |

* **TRAINING SETUP**

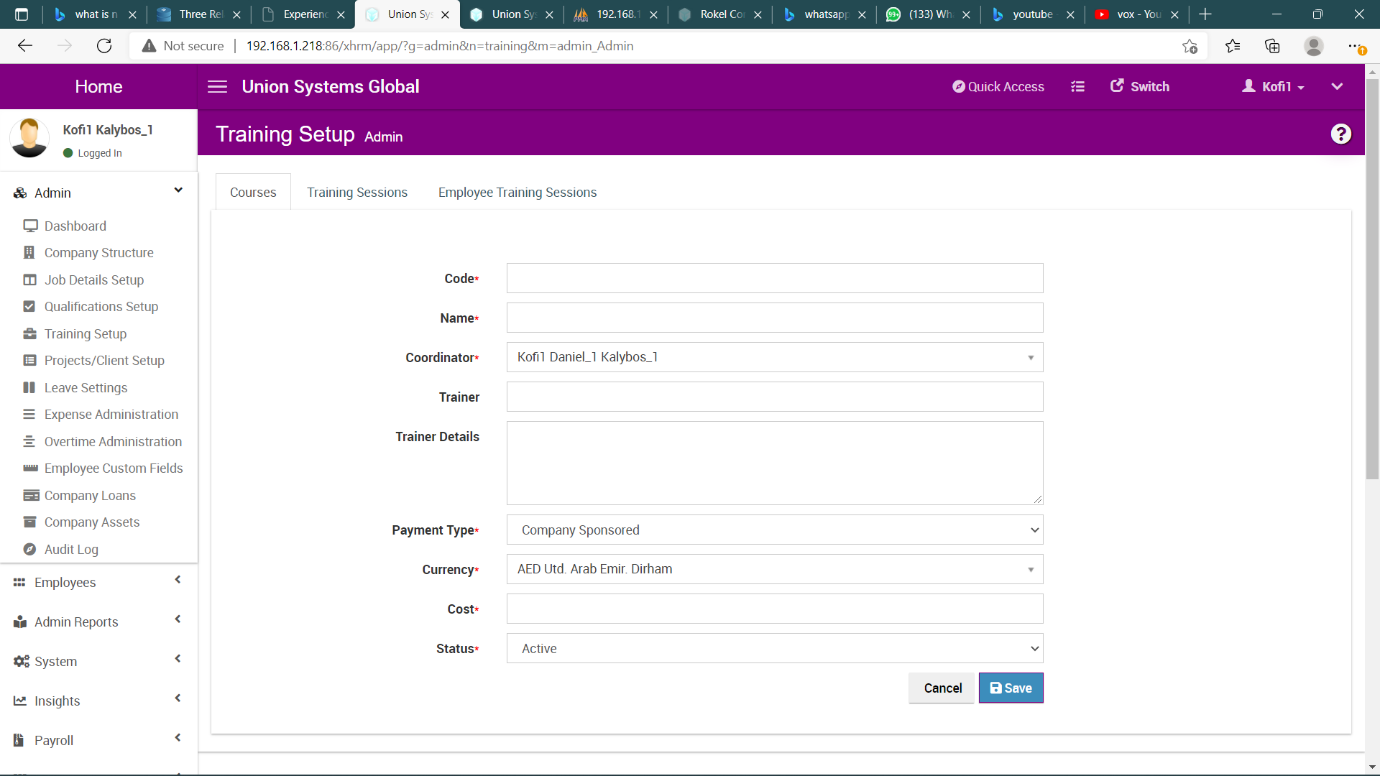
The training setup section allows the admin user to setup training parameters for the training module, it has three (3) setup parameters

1. Courses
2. Training Section
3. Employee Training Section

**COURSES**

The course section allows the admin to create courses, to create a course

1. Move to the course section
2. Click on the **ADD NEW** button to display the course form



1. Enter the details and click the **SAVE** button to add a course

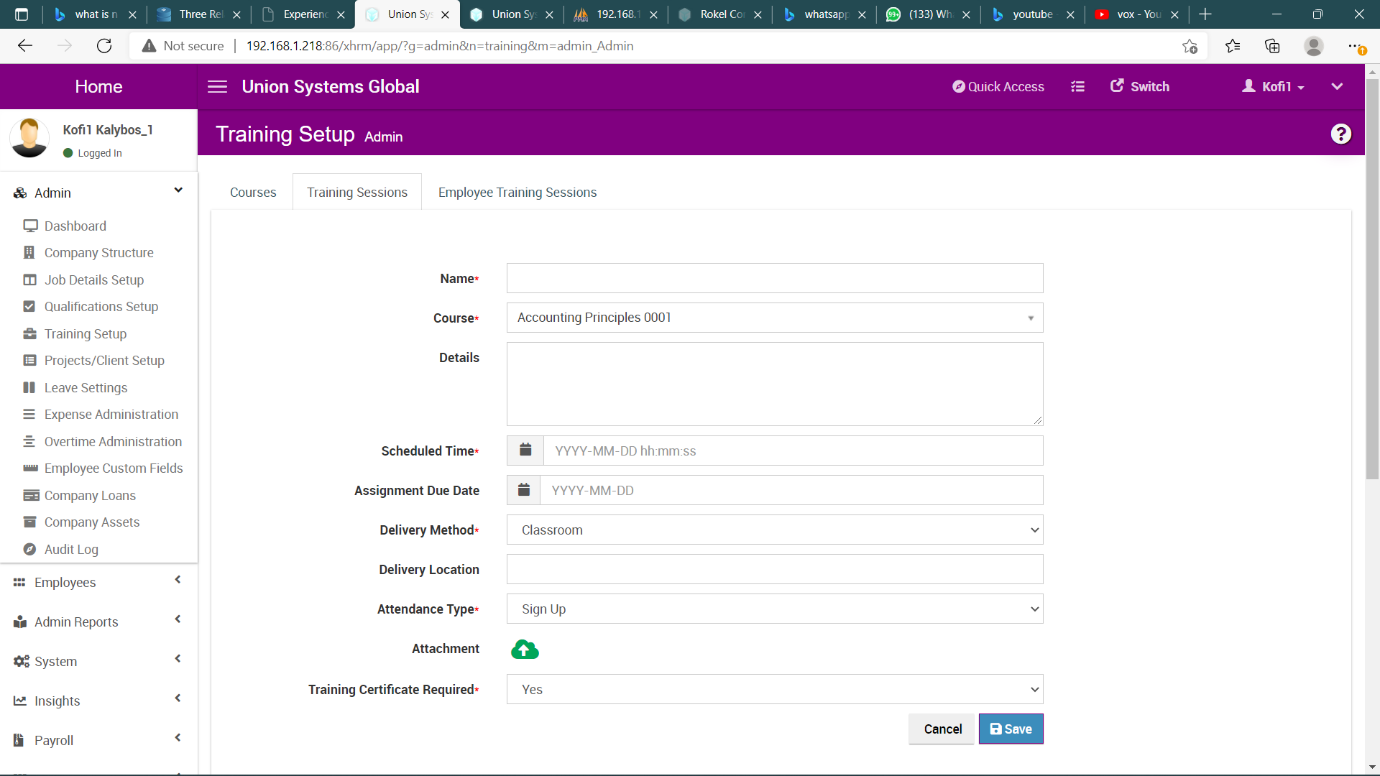
**COURSES TABLE STRUCTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| code | varchar(300) | No |  |
| name | varchar(300) | No |  |
| category | varchar(50) | Yes | NULL |
| description | text | Yes | NULL |
| coordinator | bigint(20) | Yes | NULL |
| trainer | varchar(300) | Yes | NULL |
| trainer\_info | text | Yes | NULL |
| paymentType | enum('Company Sponsored', 'Paid by Employee') | Yes | Company Sponsored |
| currency | varchar(3) | Yes | NULL |
| cost | decimal(12,2) | Yes | 0.00 |
| status | enum('Active', 'Inactive') | Yes | Active |
| created | datetime | Yes | NULL |
| updated | datetime | Yes | NULL |

**TRAINING SESSION**

The training session allows the admin to create training sessions for courses, to add a training session

1. Move to the training section
2. Click on the **ADD NEW** button, to display the training session form



1. Click the **SAVE** button to add the training session

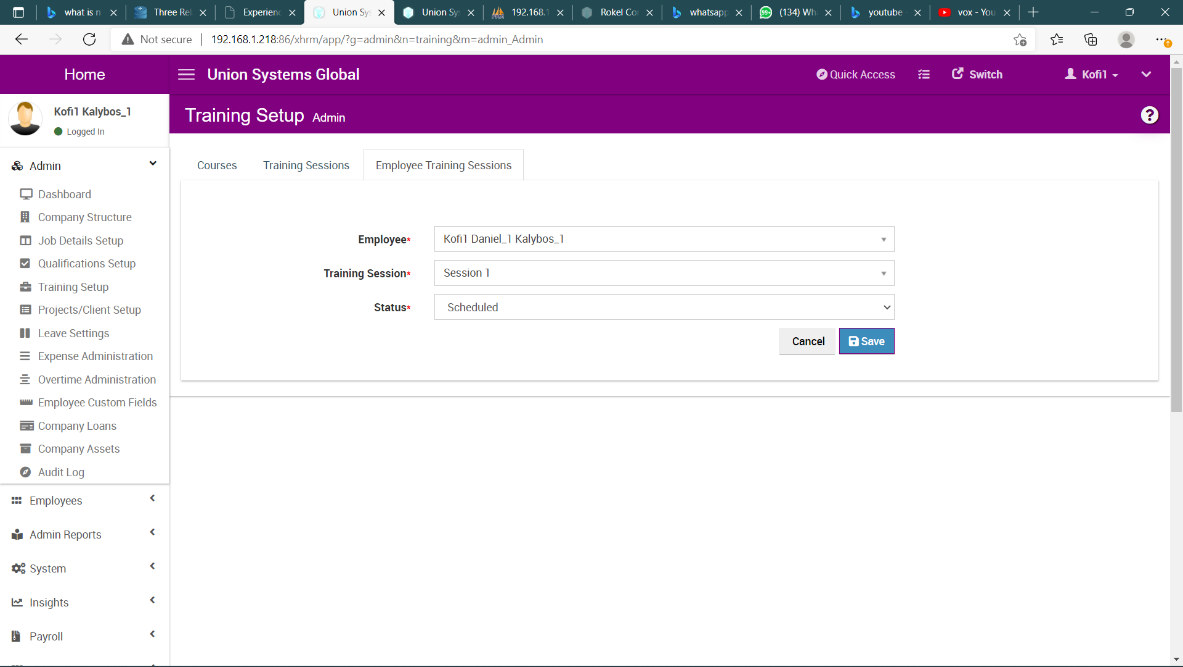
**TRAINING SESSION TABLE STRUCTURE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | bigint(20) | No |  |
| name | varchar(300) | No |  |
| course | bigint(20) | No |  |
| description | text | Yes | NULL |
| scheduled | datetime | Yes | NULL |
| dueDate | datetime | Yes | NULL |
| deliveryMethod | enum('Classroom', 'Self Study', 'Online') | Yes | Classroom |
| deliveryLocation | varchar(500) | Yes | NULL |
| status | enum('Pending', 'Approved', 'Completed', 'Cancelled') | Yes | Pending |
| attendanceType | enum('Sign Up', 'Assign') | Yes | Sign Up |
| attachment | varchar(300) | Yes | NULL |
| created | datetime | Yes | NULL |
| updated | datetime | Yes | NULL |
| requireProof | enum('Yes', 'No') | Yes | Yes |

**EMPLOYEE TRAINING SESSION**

The employee training section allows the admin to assign an employee to a particular training section, to assign an employee to a training section

1. Move to the employee training section
2. Click on the **ADD NEW** button, to display the employee training session form



1. Click the **SAVE** button to assign training session to employee