

Visitorinout Manual

Requirment:

Vsitorinout is developed using Python language and tested with Koha ILMS Version 23.11 and Ubuntu 22.04, but it may work on another older versions too. This is basically developed to record the usage statistics of library patrons for taking various managerial decisions related to Libraries. visitorinout can be used as gate register to capture the IN/OUT management of people visiting the campuses or hostels entry/exit, etc.

Installation Steps:

A. Download and Installation of Necessary packages

On the server Koha should be running or you may install on testing machine for testing of visitorinout before implementing it on Production server. The software/Libraries required to install visitorinout is listed in requirement.txt file which will be installed together with a single command.

1. Download the visitorinout from GitHub repository which is available at : <https://github.com/mishravk79> and extract the file in a folder name “visitorinout”. Install python on Ubuntu if it is not installed on your machine with the following command:

```
apt install python3
```

2. Transfer your “visitorinout” folder/files to “opt” folder of Ubuntu with the following command:

```
cp -r /home/koha/visitorinout /opt
```

3. Create Virtual Environment Creating a virtual environment on the same machine where Apache is installed is straightforward, it helps in runnin the application smoothly. It can be created with below command. Execute the following command to Install the python3-pip if it is not installed on your Server.

```
apt install python3-pip
```

4. Navigate to the directory where you want to create the virtual environment. This should be the directory where your visitorinout project files are located. (example: /opt/visitorinout), to reach your project directory execute the following command:

```
cd /opt/visitorinout/
```

5. Install “virtualenv” (if not already installed): If “virtualenv” is not already installed on your system, you can install it using pip, Python's package manager:

```
sudo apt install python3-venv
```

6. Create the Virtual Environment: Use the “python3 -m venv” command to create a new virtual environment. You can specify the name of the directory where you want to create the virtual environment. Conventionally, it's named env:

```
python3 -m venv env
```

This command will create a new directory named “env” in your current directory (/opt/visitorinout/env), which will contain the virtual environment.

7. **Activate the Virtual Environment:** Once the virtual environment is created, you need to activate it. Activating the virtual environment ensures that any Python commands you run will use the Python interpreter and packages installed within the virtual environment:

```
source env/bin/activate
```

After activation, you should see (env) in your terminal prompt, indicating that the virtual environment is active. “(env) root@koha-OptiPlex-9010:/opt/visitorinout#”

8. **Install Dependencies** with the virtual environment activated, you can now install your project dependencies using pip. Typically, you would have a requirements.txt file containing a list of dependencies. You can install them using following command:

```
pip install -r requirements.txt
```

OR if you are facing any issue then you may open the requirements.txt file and install packages listed their one by one as:

```
pip install packagename (example: pip install flask)
```

Check all the packages listed in requirments.txt is installed with following command

```
pip list
```

9. **Deactivate the Virtual Environment:** Once you're done working on your project, you can deactivate the virtual environment with the command:

```
deactivate
```

By following the above steps, you can create a virtual environment on the same machine where Apache/Koha is installed. This virtual environment will isolate your project's dependencies from other Python projects and system-wide installations, ensuring that your project runs smoothly alongside Koha ILMS.

B. Creation of database and other related configurations

A database on the existing Koha ILMS MySQL/MariaDB will be created to record the Library Visitor details like time of Check-in/Out, Which staff performed the task, etc.

1. Create the database “libraryvisitor” with the following commands:

```
mysql -uroot -p
```

```
MariaDB [(none)]>
```

```
create database libraryvisitor;
```

```
MariaDB [(none)]>
```

```
quit;
```

2. Restore the database (schema.sql)given inside the folder visitorinout

```
mysql -uroot -p libraryvisitor < /opt/visitorinout/schema.sql
```

3. Create a Koha_library database Read-Only user to read the borrowers and book issue records from your Koha ILMS with strong password as per your wish and the following commands:

-- Log in to the MariaDB server as the root user

```
mysql -u root -p
```

-- Create a read-only user

```
CREATE USER 'koha_readonly'@'localhost' IDENTIFIED BY 'readonly_password';
```

-- Grant SELECT privileges to the read-only user on the koha_library database:

```
GRANT SELECT ON koha_library.* TO 'koha_readonly'@'localhost';
```

-- Grant all privileges on the libraryvisitor database to the koha_readonly user:

```
GRANT ALL PRIVILEGES ON libraryvisitor.* TO 'koha_readonly'@'localhost';
```

-- Apply the changes:

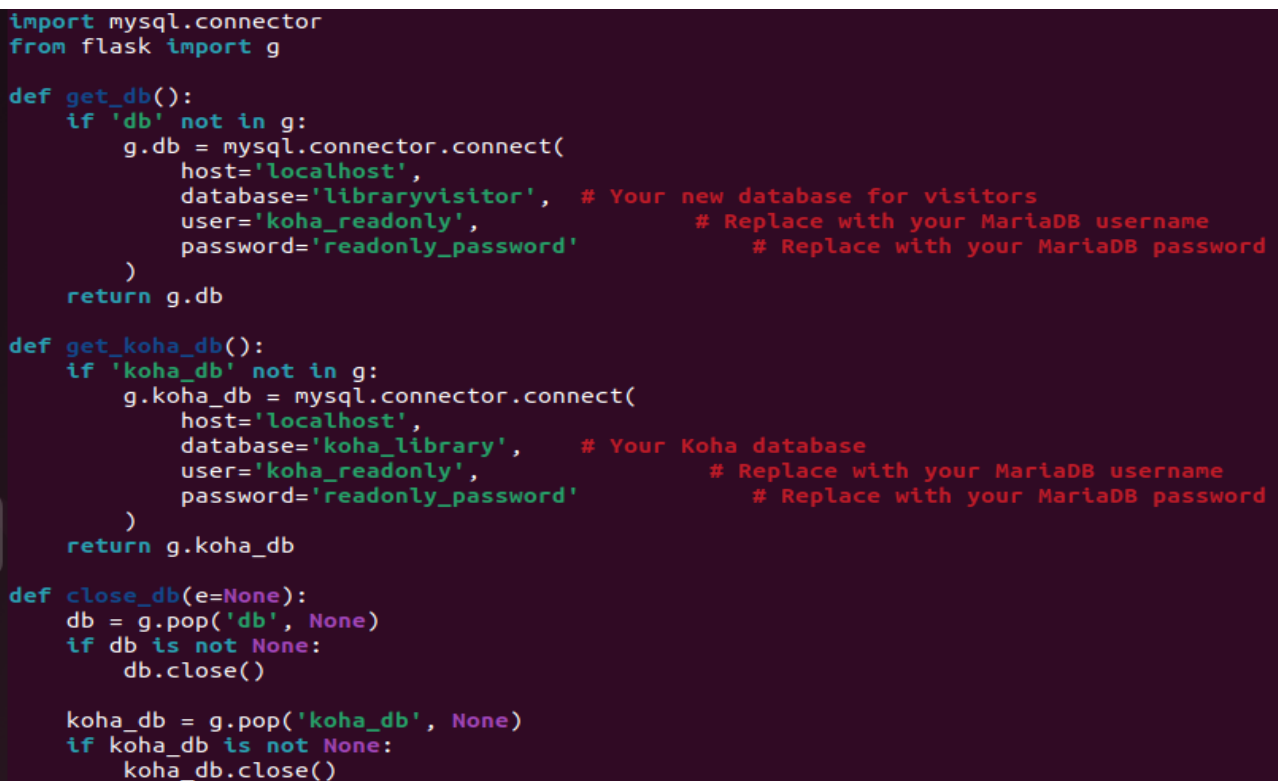
```
FLUSH PRIVILEGES;
```

```
Quit;
```

4. The above credentials need to be setup in db.py file as shown in the picture below but do not use the same credentials in production server, to set the above credential run the following command and edit the file with any editor like nano.

```
nano /opt/visitorinout/db.py
```

Figure 1: Setting credentials in db.py file



```
import mysql.connector
from flask import g

def get_db():
    if 'db' not in g:
        g.db = mysql.connector.connect(
            host='localhost',
            database='libraryvisitor', # Your new database for visitors
            user='koha_readonly',      # Replace with your MariaDB username
            password='readonly_password' # Replace with your MariaDB password
        )
    return g.db

def get_koha_db():
    if 'koha_db' not in g:
        g.koha_db = mysql.connector.connect(
            host='localhost',
            database='koha_library', # Your Koha database
            user='koha_readonly',    # Replace with your MariaDB username
            password='readonly_password' # Replace with your MariaDB password
        )
    return g.koha_db

def close_db(e=None):
    db = g.pop('db', None)
    if db is not None:
        db.close()

    koha_db = g.pop('koha_db', None)
    if koha_db is not None:
        koha_db.close()
```

5. To generate sql report through Koha ILMS grant permission is needed on both the databases

```
mysql -u root -p
```

```
GRANT SELECT ON libraryvisitor.visitorsdetail TO 'koha_library'@'localhost';
```

```
SHOW GRANTS FOR 'koha_library'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
quit;
```

C. Configuration of VirtualHost and Apache

1. Configure the visitorinout on apache server (You may configure alongside Koha or other applications already running). Enable mod_wsgi: Apache uses modules to extend its functionality. One such module is mod_wsgi, which allows Apache to serve Python web applications. You'll need to enable mod_wsgi if it's not already enabled and install it if not available:

```
sudo apt-get install libapache2-mod-wsgi-py3
```

```
sudo a2enmod wsgi
```

2. Create a new Apache virtual host configuration file for your Python application (visitorinout). This file will specify how Apache should handle requests to your application. For example, create a file named “libraryvisitor.conf” in Apache's sites-available directory with following command:

```
sudo nano /etc/apache2/sites-available/visitorinout.conf
```

```
<VirtualHost *:8085>
    ServerName localhost

    # WSGI daemon process configuration
    WSGIDaemonProcess visitorinout_python_apps \
        python-home=/opt/visitorinout/env \
        python-path=/opt/visitorinout

    # WSGI process group for the daemon process
    WSGIProcessGroup visitorinout_python_apps

    # WSGI script alias for the Flask application
    WSGIScriptAlias / /opt/visitorinout/app.wsgi

    # Directory configuration for the WSGI script
    <Directory /opt/visitorinout>
        <Files app.wsgi>
            Require all granted
        </Files>
    </Directory>

    # Directory configuration for the templates
    Alias /templates /opt/visitorinout/templates
    <Directory /opt/visitorinout/templates>
        Require all granted
    </Directory>

    # Directory configuration for static files (if any)
    Alias /static /opt/visitorinout/static
    <Directory /opt/visitorinout/static>
        Require all granted
    </Directory>
</VirtualHost>
```

Replace yourdomain.com with your actual domain name or server IP address. Adjust /path/to/your/virtualenv and /path/to/your/project with the appropriate paths to your Python virtual environment and project directory (as you have created in previous steps and shown above).

3. Enable the Virtual Host and Port

--To Enable the virtual host you created and change the ports if changed above run the following command:

```
sudo a2ensite visitorinout.conf
```

--Add new port as you have added in the above virtualhost file and save it.

```
sudo nano /etc/apache2/ports.conf
```

After making any configuration changes, restart Apache to apply the changes with following command:

```
sudo systemctl restart apache2
```

Now you can open the Visitorinout with url **http://yourip:8085/login** (Your IP and the Port assigned in Virtual Host file above).

D. Creation of Staff credentials to access visitorinout (Library Visitor) and Reports

1. It is a similar method as credentials is created through Patron module of Koha ILMS to access staff interface. You may create a user in the Koha as usual and give him/her minimum permission like to only access catalogue as shown in the below screenshot. However, all persons having any type of staff permission can access the Library Visitor (visitorinout), but through report it will be tracked who checked in/out patrons.

Figure 2: Setting minimum staff permission to access Visitorinout

The screenshot shows the Koha Patron module interface for setting permissions for a patron named Vinod Kumar Mishra (1). The interface includes a navigation bar with 'Circulation', 'Patrons', 'Search', and 'More' options. Below the navigation bar, there is a search bar and buttons for 'Search patrons', 'Check out', and 'Check in'. The main content area is titled 'Set permissions for Vinod Kumar Mishra (1)' and includes a 'Save' button, a 'Cancel' button, and buttons for 'Show all', 'Hide all', 'Select all', and 'Clear all'. A list of permissions is displayed, with the following permissions selected:

- ☒ Access to all librarian functions (superlibrarian) This permission grants access to all areas. If selected, specific sub-permissions ca
- ☒ Check out and check in items (circulate)
- ☒ Staff access, allows viewing of catalogue in staff interface (catalogue) Required for staff login.
- ☐ Manage Koha system settings (Administration panel) (parameters) Show de
- ☐ Add, modify and view patron information (borrowers) Show de
- ☐ Set user permissions (permissions)
- ☐ Place and modify holds for patrons (reserveforothers) Show de

A dropdown menu is open, showing the following options: Renew patron, Set permissions, Send welcome email, Delete, and Export today's checked in barcodes.

On the left side of the interface, there is a profile card for Vinod Kumar Mishra (1) with the following information:

- No address stored.
- No city stored.
- No phone stored.
- No email stored.
- No date of birth stored.
- Category: Library Staff (STAFF)
- Home library: My Central Library
- Borrowernumber: 1

2. All the reports related to Library Visitor can be created like other reports designed to generate Koha reports/statistics. Some of the reports format according to above configurations are given below:

a. Details of all visitors in between two date range and Staff who Checked-in/out

```
SELECT v.checkin_time 'Checked-in Time', v.checkout_time 'Checked-out Time',  
CONCAT(b.firstname, ' ', b.surname) 'Name', b.email 'Email', b.cardnumber 'ID Number', b.phone  
'Contact Number', b.sex 'Male/Female', v.staff_checkin 'Staff Checked-in', v.staff_checkout 'Staff  
Checked-out' FROM libraryvisitor.visitorsdetail v JOIN koha_library.borrowers b ON  
v.borrowernumber = b.borrowernumber WHERE DATE(v.checkin_time) BETWEEN <<Date  
Between (dd/mm/yyyy)|date>> AND <<and (dd/mm/yyyy)|date>> -- Replace with your desired  
date range ORDER BY v.checkin_time
```

b. Monthly Statistics of Visitors

```
SELECT DATE_FORMAT(v.checkin_time, '%m-%Y') AS 'Month and Year', COUNT(*) AS 'Total  
Visitors' FROM libraryvisitor.visitorsdetail v WHERE DATE(v.checkin_time) BETWEEN <<Start  
Date (dd/mm/yyyy)|date>> AND <<End Date (dd/mm/yyyy)|date>> -- Replace with your desired  
date range GROUP BY DATE_FORMAT(v.checkin_time, '%m-%Y') ORDER BY  
DATE_FORMAT(v.checkin_time, '%Y-%m')
```

c. Total Number of visitor per day in between two date range

```
SELECT DATE(v.checkin_time) AS 'Date of Visit', COUNT(*) AS 'Total Visitor' FROM  
libraryvisitor.visitorsdetail v WHERE DATE(v.checkin_time) BETWEEN <<Start Date  
(dd/mm/yyyy)|date>> AND <<End Date (dd/mm/yyyy)|date>> -- Replace with your desired date  
range GROUP BY DATE(v.checkin_time) ORDER BY DATE(v.checkin_time)
```

d. Details of all visitors who have not yet checked out from Library

```
SELECT v.checkin_time AS 'Checked-in Time', v.checkout_time AS 'Checked-out Time',  
CONCAT(b.firstname, ' ', b.surname) AS 'Name', b.email AS 'Email', b.cardnumber AS 'ID  
Number', b.phone AS 'Contact Number', b.sex AS 'Male/Female' FROM libraryvisitor.visitorsdetail  
v JOIN koha_library.borrowers b ON v.borrowernumber = b.borrowernumber WHERE  
v.checkout_time IS NULL ORDER BY v.checkin_time
```

Figure 3: Sample reports prepared with Report Group (Visitor Report)

All

Accounts

Acquisitions

Catalog

Circulation

Patrons

Serials

Visitor Report

Subgroup:

All

Showing 1 to 4 of 4 entries (filtered from 5 total entries)

Show

20

entries

< Previous

Next >

Search:

✕ Clear filter

⚙ Columns

▼

📄 Export

▼

	ID	Report name	Type	Subgroup	Notes	Author	Creation date	Last edit	Last run	Public	JSON URL	Cache expiry (seconds)
<input type="checkbox"/>	1	Details of all visitors in between two date range				Mishra, Vinod Kumar (1)	07/07/2024	02/08/2024 21:59	25/07/2024 00:28	No	/cgi-bin/koha/svc/report?id=1	300
<input type="checkbox"/>	2	Total visitor per day in between two date range				Mishra, Vinod Kumar (1)	23/07/2024	02/08/2024 21:58	30/07/2024 12:56	No	/cgi-bin/koha/svc/report?id=2	300
<input type="checkbox"/>	3	Monthly Statistics of Visitors				Mishra, Vinod Kumar (1)	23/07/2024	02/08/2024 21:58	23/07/2024 19:50	No	/cgi-bin/koha/svc/report?id=3	300
<input type="checkbox"/>	4	Details of all visitors who have not yet checked out from Library				Mishra, Vinod Kumar (1)	30/07/2024	02/08/2024 21:59	30/07/2024 17:36	No	/cgi-bin/koha/svc/report?id=4	300

Figure 4: Sample Bar Chart created from the daily visitor statistics in Koha ILMs

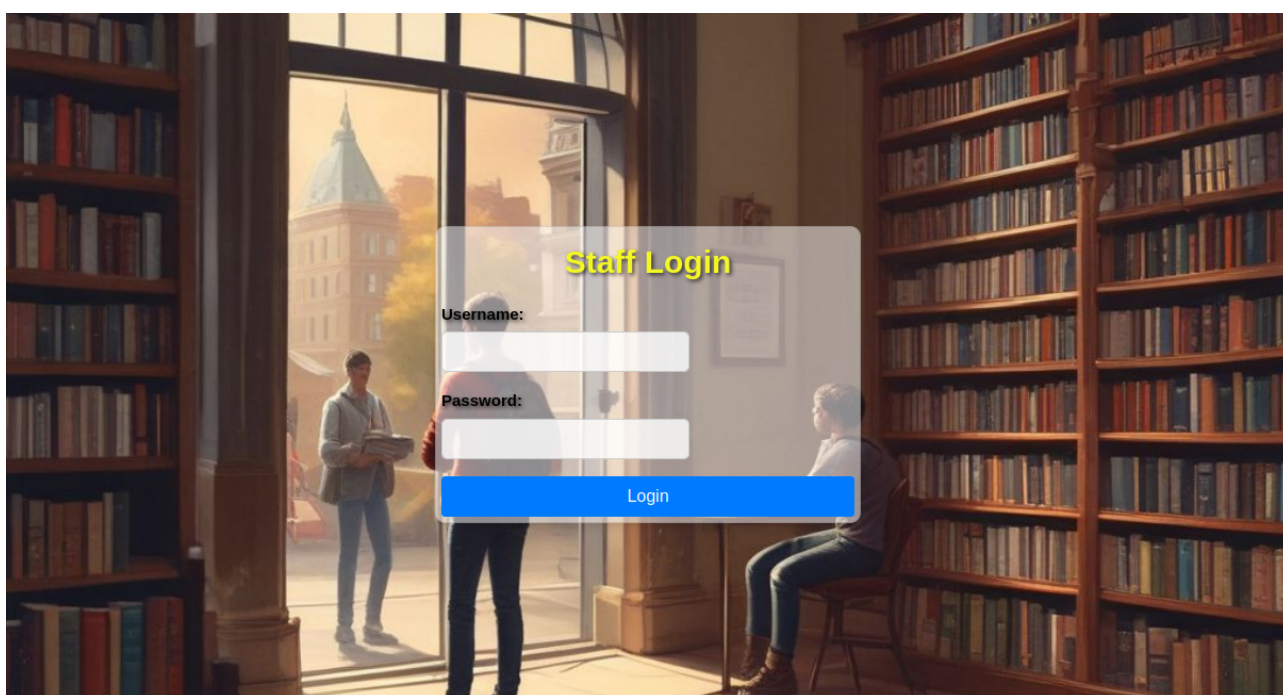


Handling of Visitorinout and other Options

The visitorinout have common features like any other In/Out management system apart from that at the time of Check-out it display the list of book/s issued to the member so that the staff at exit gate can verify the books taken by him/her without any additional printing of Check-out slip.

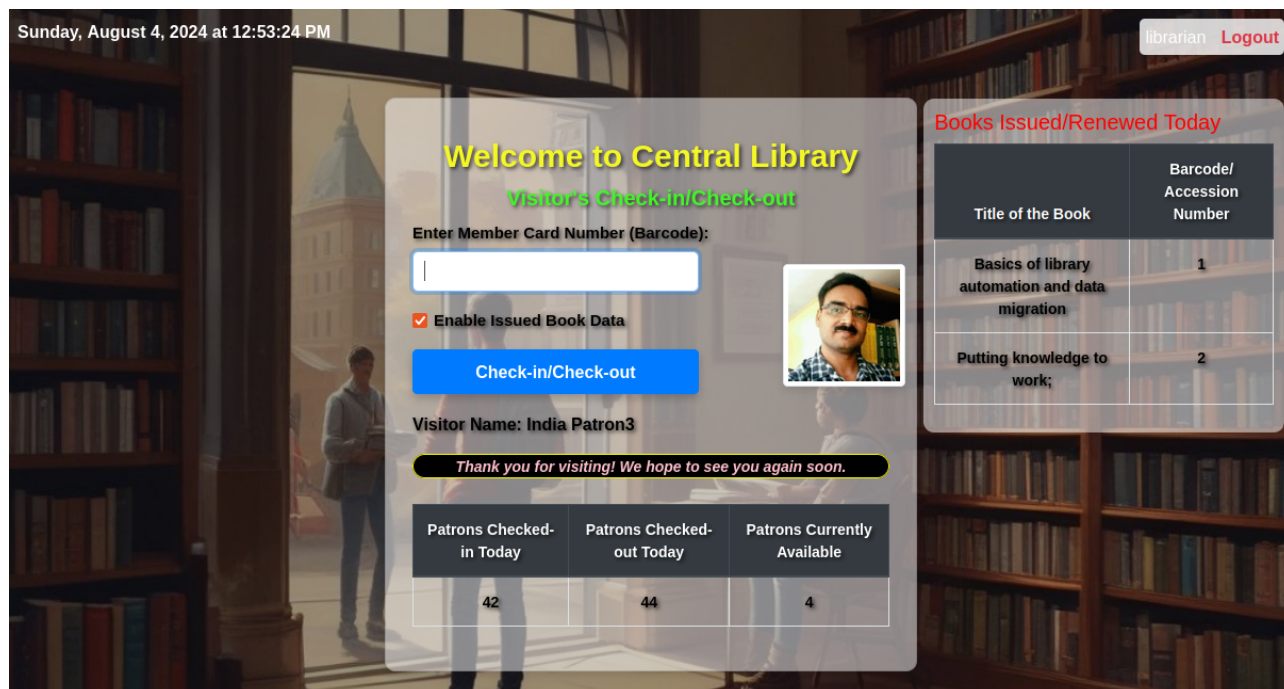
1. Staff Login to the portal is very simple and anyone having any type of staff permission in Koha ILMS can loged into the system and perform the Visitor In/Out task. The User ID of the Staff responsible to execute the check in/out task will be saved in the libraryvisitor database which can be tracked with reports.

Figure 5: Staff Login



2. You may “Enable Issued Book Data” fetching while member is checking-out from the exit gate to ensure proper transaction of books. If it is not applicable to your library you may Untick the option “Enable Issued Book Data” to show only meber detail checking-out from the library.

Figure 6: The list of checked out books issued to member to be verified by staff at exit.



- Customization of Various nomenclature seen in the above screenshot like Library Name, Other headings, fonts, etc is possible by editing the concern files like css, html, etc. with the help of editor.

Libraryvisitor Database Schema

Only 1 table (visitorsdetail) is created under the libraryvisitor database with Six columns and their headings as shown in the below schreenshot. These columns heading may be used for generating various reports and other data analysis. The borrowernumber is linked with Koha database table brrowers.borrowernumber.

Figure 7: Libraryvisitor database table structure/schema

+ Options							
		id	borrowernumber	checkin_time	checkout_time	staff_checkin	staff_checkout
<input type="checkbox"/>	Edit	1080	10000016	2024-08-05 00:24:54	2024-08-05 00:30:05	librarian	librarian
<input type="checkbox"/>	Edit	1072	10000016	2024-08-04 01:45:19	2024-08-04 01:45:59	librarian	librarian

IMPORTANT NOTES

- Do not use the above User ID and Paaswords in your production server and set your own strong credentials.**

- The Above process assumes your Koha database name is “Koha_library” and if there is any changes of database name you need to make modifications in /opt/visitorinout/app.py and /opt/visitorinout/db.py file with appropriate editor like nano.
- If you are attempting to upgrade Koha and there is any changes in borrowers table (borrowernumber) system may hardly allow neither it will allow you to delete the Koha database as the borrowers table is linked with libraryvisitor database. In this case you may take backup of libraryvisitor database and delete it, after upgradation of Koha or other works you may restore the libraryvisitor database as it is.

- To know apache problem run the following command:

```
sudo apachectl configtest
```

- Check apache error with the following command:

```
sudo tail -f /var/log/apache2/error.log
```

- Create index of columns in libraryvisitor database table visitorsdetail for fast searching if any NULL value is left. It will help in fast processing of member check-out and fetch of book issued records if activated. Access your MySQL/MariaDB database by typing the following command and entering your password when prompted:

```
mysql -uroot -p
```

Switch to your database:

```
USE libraryvisitor;
```

Create the indexes by running the following SQL commands:

```
CREATE INDEX idx_checkout_time_null ON visitorsdetail(checkout_time);
```

```
quit;
```

- There may be other methods of installations and other customizations which may be explored and shared to the community.

Third-Party Libraries

This project uses the following third-party libraries:

- **Bootstrap:** A popular front-end framework. License information can be found in the licenses/bootstrap-license.txt file.
- **jQuery:** A fast, small, and feature-rich JavaScript library. License information can be found in the licenses/jquery-license.txt file.

The license files are included in the licenses directory of this project.

License

This project is licensed under the MIT License - see the (LICENSE) file for details.