Head count with visitor gate pass Software Ver-1.0

Installation Procedure

Software Application	**Head count with visitor gate pass Software**
Version	**1.0**
Licensed	**Indian Oil Corporation (PMC)**
Developed by	**ComSys-IT, New Delhi**

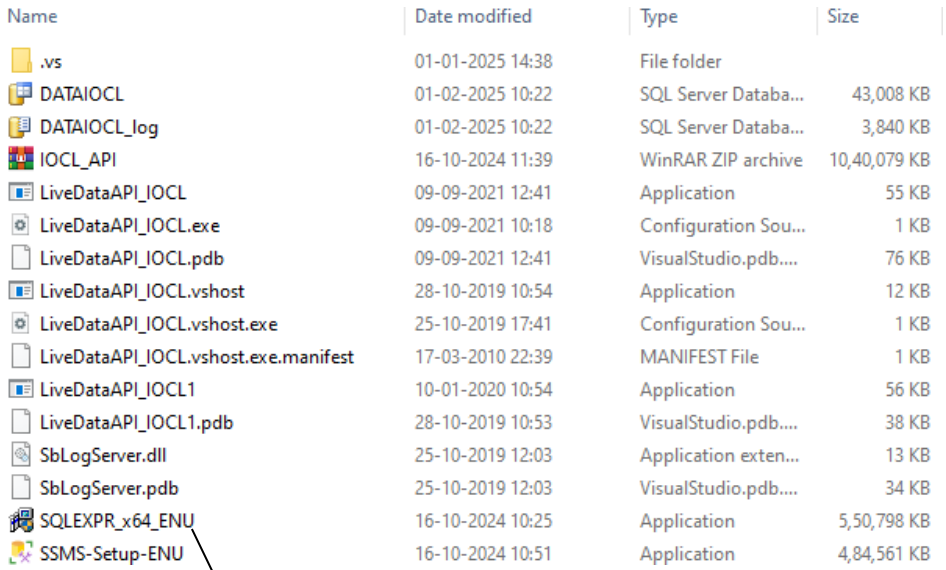
Follow the Following Steps to Install the LKMS Package

Step 1

- Download the file and D://drive to unzip the folder link

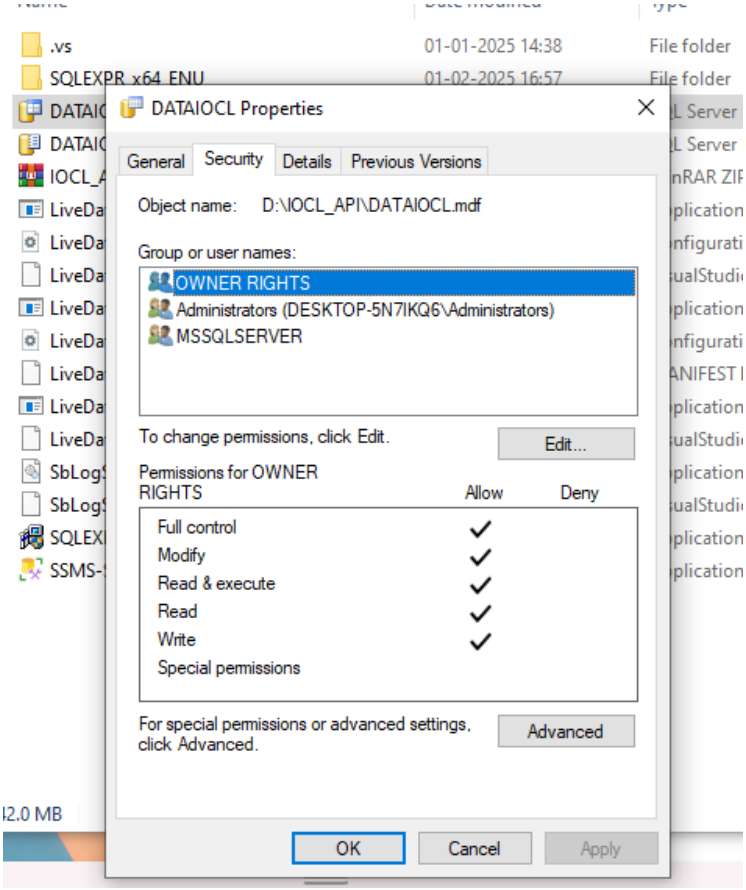
Step 2

- Open the **Package** Folder
- Double click on **Setup.exe** file to Run

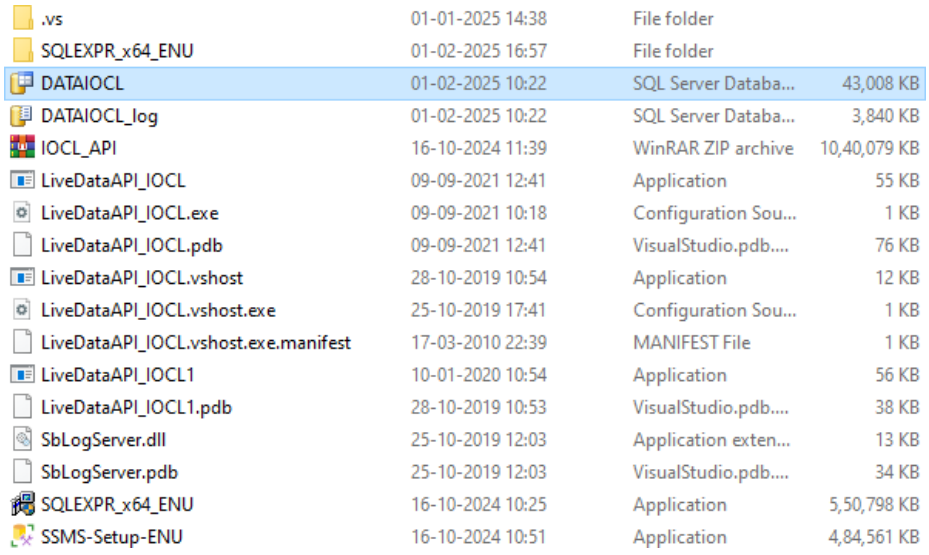


Name	Date modified	Type	Size
.vs	01-01-2025 14:38	File folder	
DATAIOCL	01-02-2025 10:22	SQL Server Databa...	43,008 KB
DATAIOCL_log	01-02-2025 10:22	SQL Server Databa...	3,840 KB
IOCL_API	16-10-2024 11:39	WinRAR ZIP archive	10,40,079 KB
LiveDataAPI_IOCL	09-09-2021 12:41	Application	55 KB
LiveDataAPI_IOCL.exe	09-09-2021 10:18	Configuration Sou...	1 KB
LiveDataAPI_IOCL.pdb	09-09-2021 12:41	VisualStudio.pdb....	76 KB
LiveDataAPI_IOCL.vshost	28-10-2019 10:54	Application	12 KB
LiveDataAPI_IOCL.vshost.exe	25-10-2019 17:41	Configuration Sou...	1 KB
LiveDataAPI_IOCL.vshost.exe.manifest	17-03-2010 22:39	MANIFEST File	1 KB
LiveDataAPI_IOCL1	10-01-2020 10:54	Application	56 KB
LiveDataAPI_IOCL1.pdb	28-10-2019 10:53	VisualStudio.pdb....	38 KB
SbLogServer.dll	25-10-2019 12:03	Application exten...	13 KB
SbLogServer.pdb	25-10-2019 12:03	VisualStudio.pdb....	34 KB
SQLEXPR_x64_ENU	16-10-2024 10:25	Application	5,50,798 KB
SSMS-Setup-ENU	16-10-2024 10:51	Application	4,84,561 KB

- Install the sql server
- When open the packages choos first column and next->page to choose username or password. By default username is “sa” and password=”comsys@123” then continue and every enable the automatic

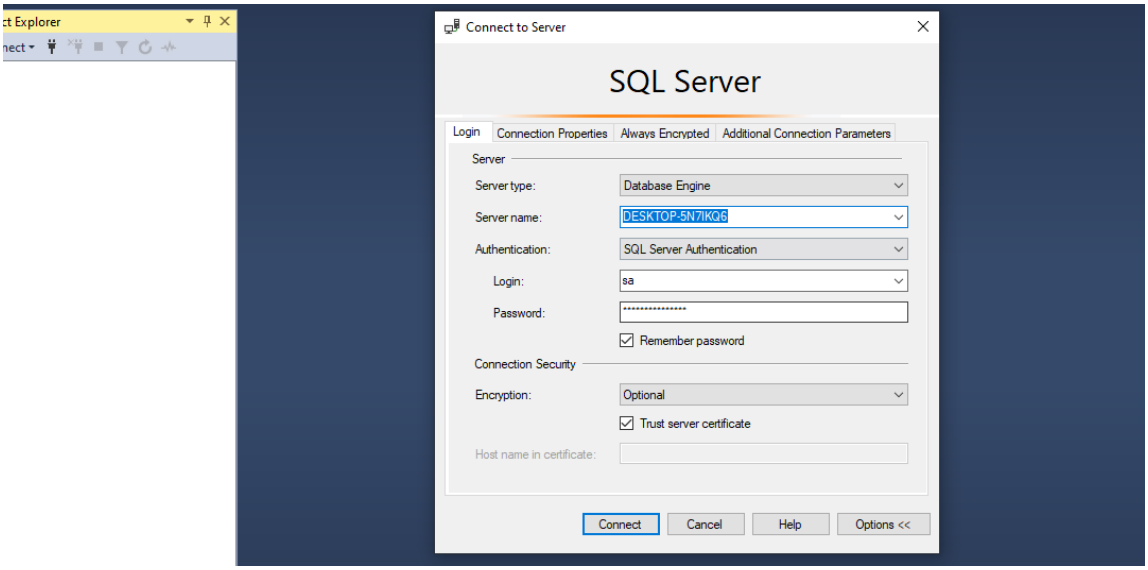


Full access the both the file . click right->security and edit ->full access

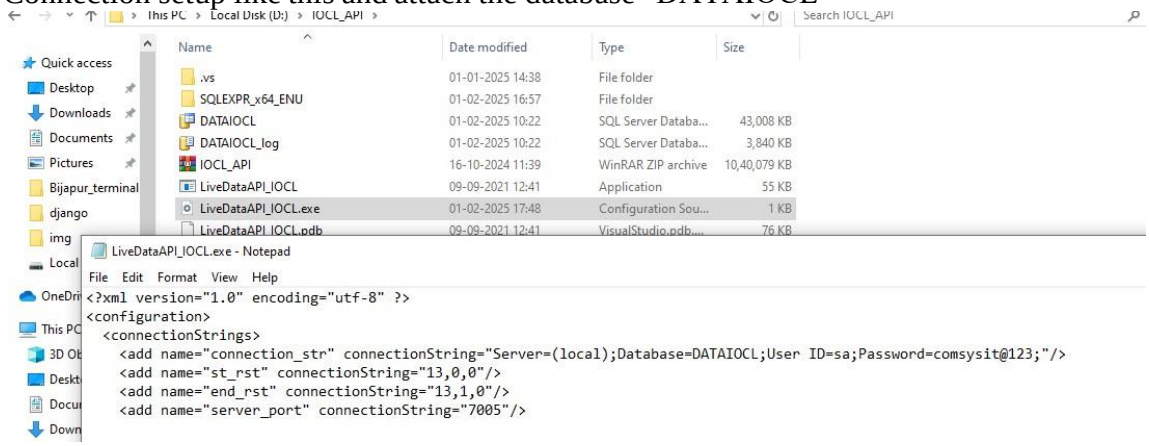


Click s

Click ssms-setup-ENU and complete install



Connection setup like this and attach the database “DATAIOCL”

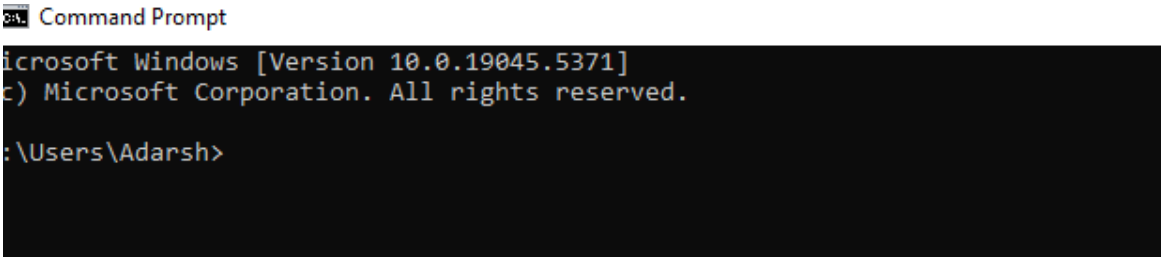


Setting file open and edit in notepad same as local change to servername,database=DATAIOCL and sa, password=comsysit@123 same as sql server password paste it

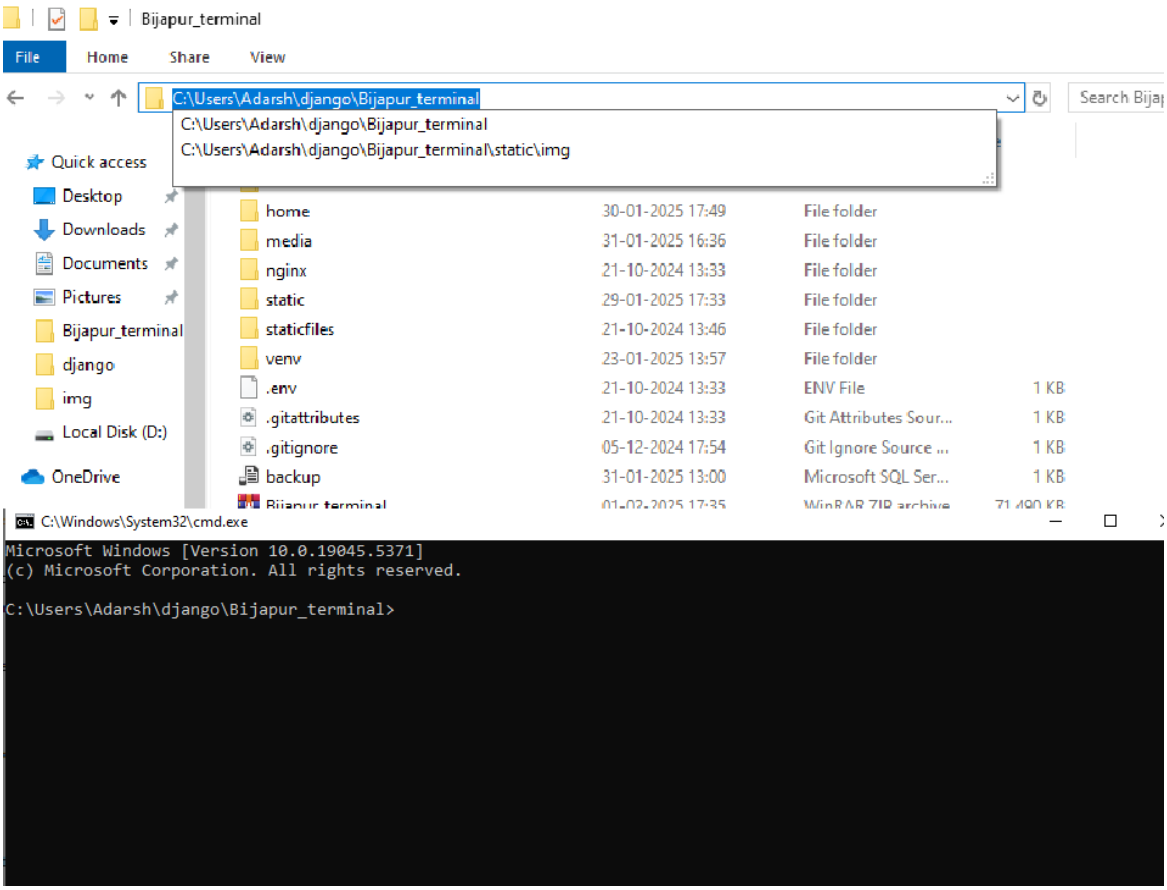
Then livedata iocl open and integrate ip addresss all gate->

Head Count Software With visitor Gatepass:

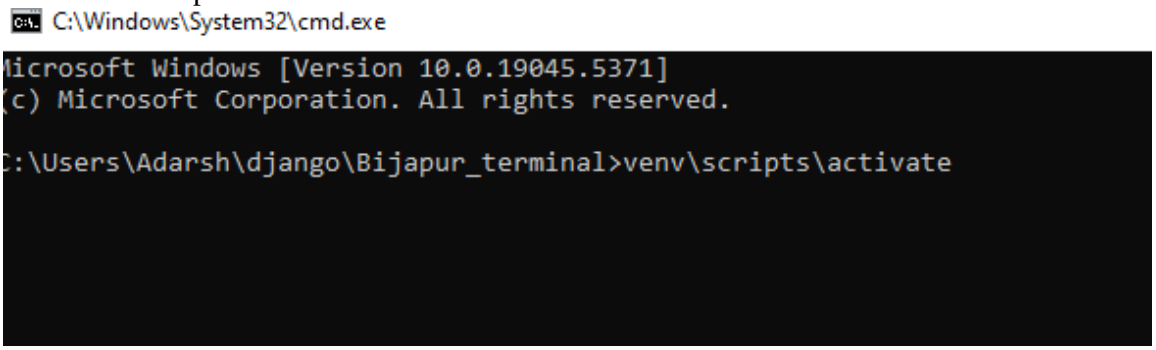
Headcount and visitor gatepass download zip file and then unzip file folder simple click cmd open commandprompt that particular folder paste project
python package downlod and install full all tick mark



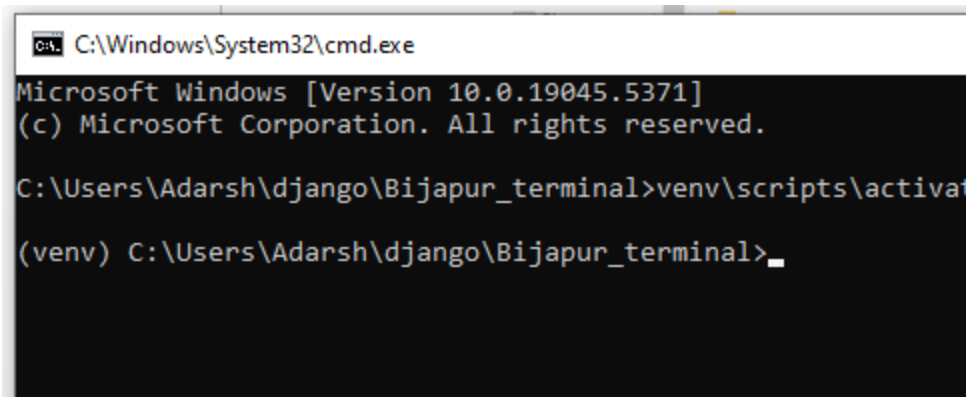
go to that folder and write cmd then enter direct open page promt



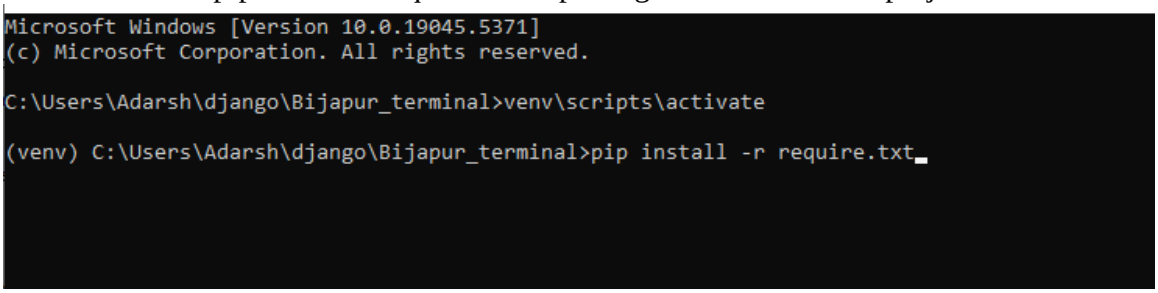
enter command python –m venv venv then press enter to create virtual enviroment
enter venv\script\activate then enter



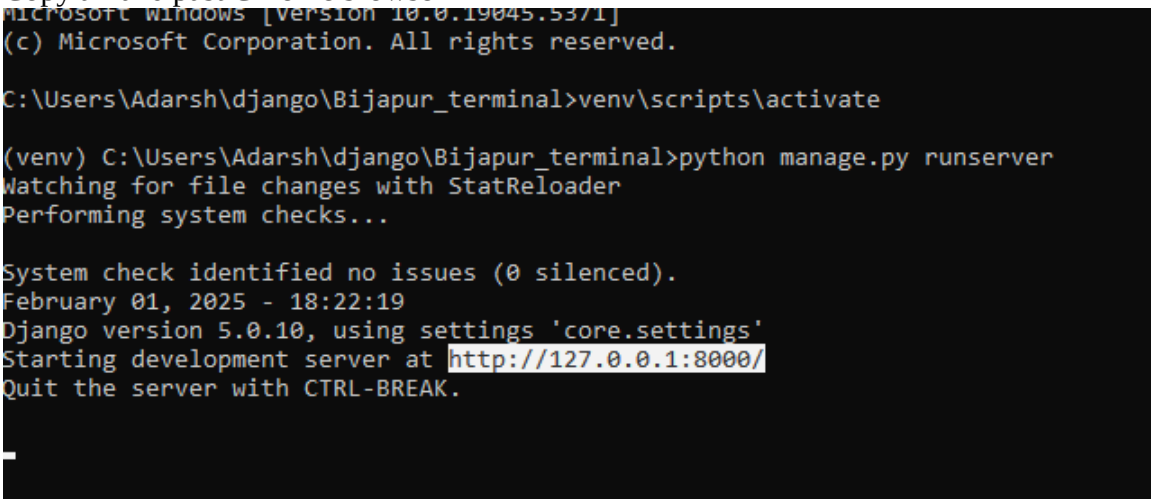
then activate the virtual enviroment



Enter command pip install –r require.txt all package download entire project



successfully download then Enter the command prompt:
"python manage.py runserver"
Copy url and past Chrome browser



With the Help of above mentioned process you can complete the Installation Process. project under table_reset.txt file to reset the table particular model

table_reset.txt

```
python manage.py shell
from django.db import connection
from home.models import EmpMast,EnrollMast,MonitorData,DepartMast,DesMast,gatePass
EmpMast.objects.all().delete()
EnrollMast.objects.all().delete()
MonitorData.objects.all().delete()
DepartMast.objects.all().delete()
DesMast.objects.all().delete()
with connection.cursor() as cursor:
    cursor.execute("DELETE FROM sqlite_sequence WHERE name='home_monitordata';")
    cursor.execute("DELETE FROM sqlite_sequence WHERE name='home_empmast';")
    cursor.execute("DELETE FROM sqlite_sequence WHERE name='home_enrollmast';")
    cursor.execute("DELETE FROM sqlite_sequence WHERE name='home_monitordata';")
    cursor.execute("DELETE FROM sqlite_sequence WHERE name='home_departmast';")
    cursor.execute("DELETE FROM sqlite_sequence WHERE name='home_desmast';")
```

Under the virtual environment the paste query step by step:

```
(venv) C:\Users\Adarsh\django\Vasco_Terminal>python manage.py shell
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>>
```

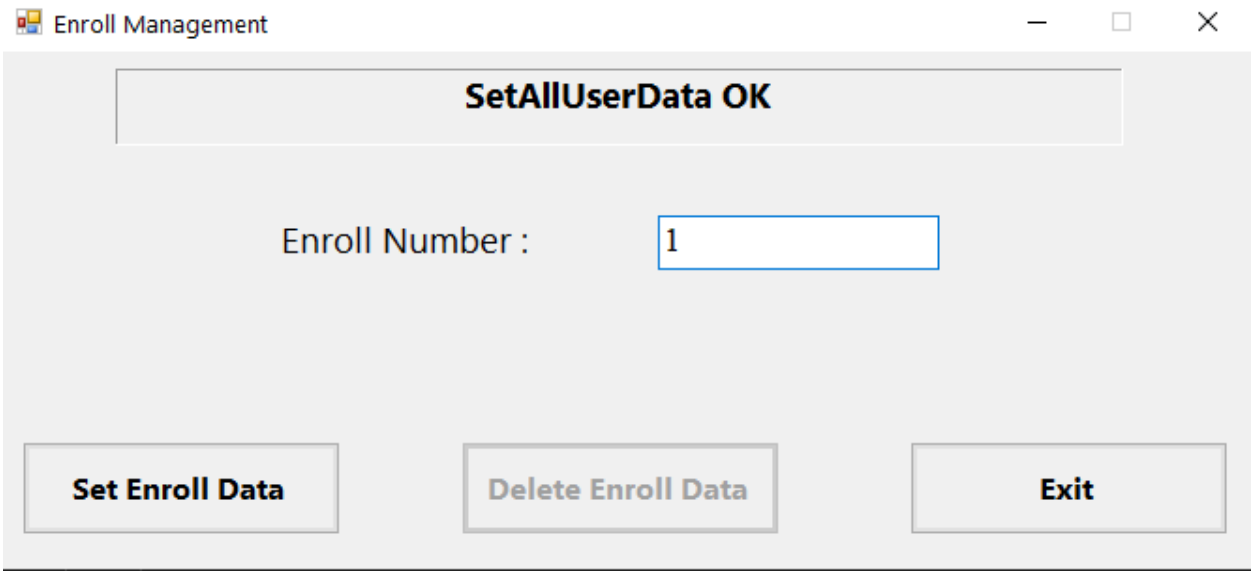
Exit() the shell

```
(venv) C:\Users\Adarsh\django\Vasco_Terminal>python manage.py shell
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> exit()

(venv) C:\Users\Adarsh\django\Vasco_Terminal>
```

Sending Data SDK Manual (Automatic)

Overview

The Sending Data SDK facilitates the transmission of enrollment data to biometric devices. This manual provides a concise guide to key methods and their functionality.



1. frmEnroll_Load Method

Purpose: Initializes the form, prepares data structures, and triggers enrollment data setup.

Key Operations:

- Initializes byte arrays and integer arrays for enrollment data.
- Loads enrollment data from the database using `GetEnrollDatas()`.
- Deletes old visitor data using `DeleteOldVisitData()`.
- Gets the latest enrollment number using `GetLatestEnrollNumber()`.
- Initiates data sending using `cmdSetEnrollData_Click()`.

2. DeleteOldVisitData Method

Purpose: Deletes outdated visitor data from both the device and the database.

Key Steps:

- Fetches old visitor data from the database using a SQL query.
- Connects to the device using `sbxpc.SBXPCDLL.ConnectTcpip()` and deletes the enrollment data using `sbxpc.SBXPCDLL.DeleteEnrollData()`.
- Deletes the outdated visitor records from the database using a SQL `DELETE` command.

3. GetLatestEnrollNumber Method

Purpose: Retrieves the most recent enrollment number from the database.

Key Steps:

- Executes a SQL query to fetch the latest enrollment number.
- Updates the `txtEnrollNumber` text box with the retrieved enrollment number.

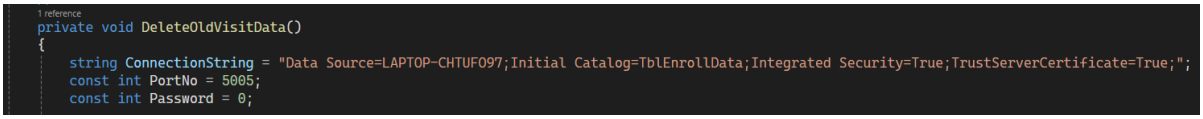
4. cmdSetEnrollData_Click Method

Purpose: Sends enrollment data to the biometric device based on the enrollment number.

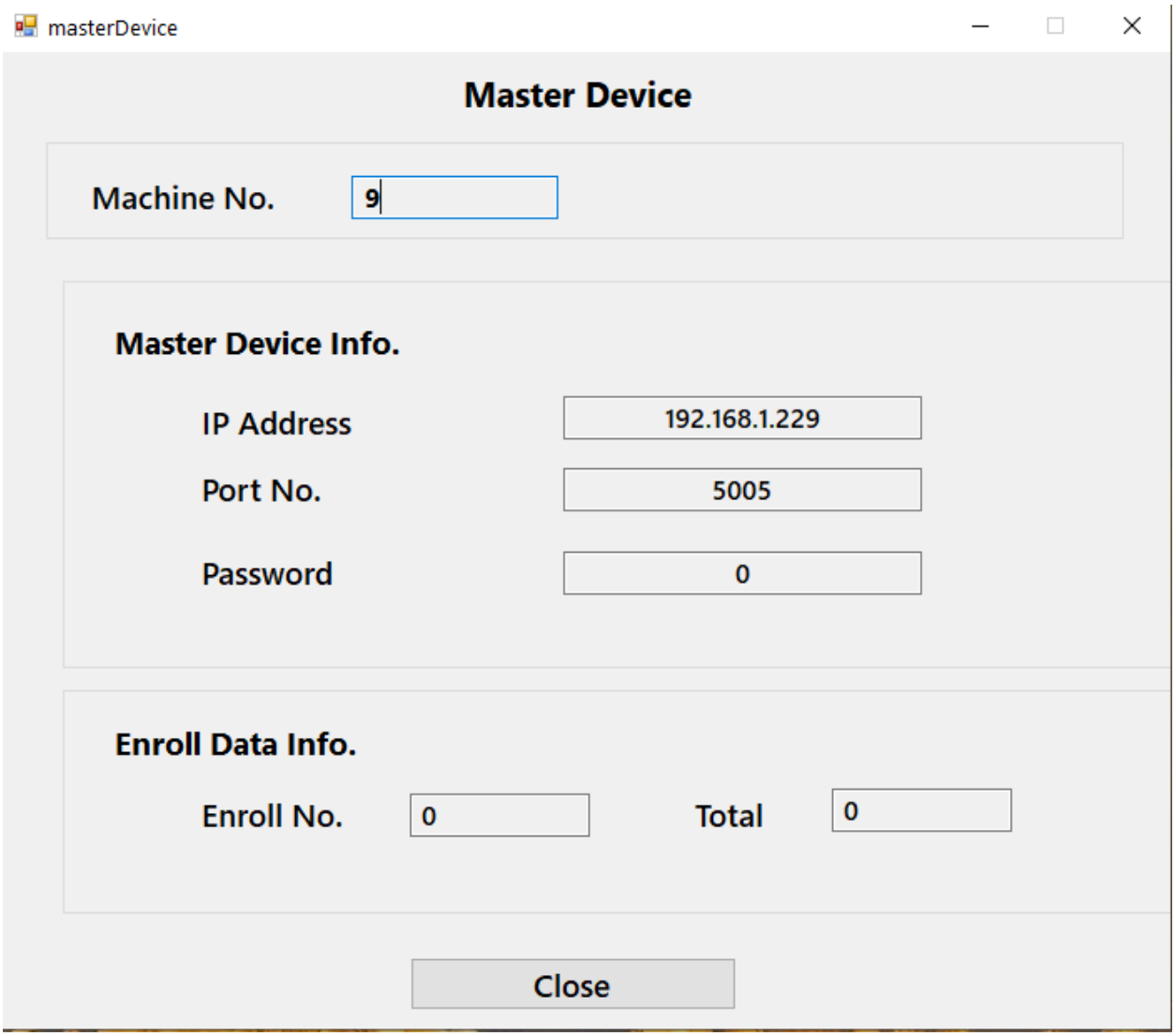
Key Steps:

- Fetches visitor and enrollment data using SQL queries.
- Connects to the biometric device using `sbxpc.SBXPCDLL.ConnectTcpip()`.
- Converts fingerprint data into a format suitable for the device.
- Sends the data using `sbxpc.SBXPCDLL.SetEnrollData1()`.
- Displays success or error messages based on the operation result.

Modifications



Update the connection string in `DeleteOldVisitData() , GetLatestEnrollNumber(),cmdSetEnrollData_Click(object sender, EventArgs e) cmdDeleteEnrollData_Click(object sender, EventArgs e)` and in `EnrollData.cs` based on location site .



Master Device SDK

Overview

This manual provides documentation for the SDK that retrieves enrollment data from a master device and inserts new data into the database. The SDK interacts with the master device to extract user enrollment data and saves it in the SQL Server database if the data is not already present.

1. masterDevice_Load Method

Purpose: Initializes the form, establishes a connection to the master device, retrieves enrollment data, and initiates the exit timer.

Key Steps:

- Disables the maximize box.
- Sets multiple textboxes as readonly to prevent user input.
- Displays master device information (Machine No, IP Address, Port No, Password).

- Initializes `gTemplngEnrollData` and `gbytEnrollData` arrays.
- Connects to the master device using `sbxpc.SBXPCDLL.ConnectTcpip()`.
- Calls `GetAllEnrollData()` to fetch all enrollment data.
- Starts the exit timer to automatically close the form after 2 minutes.

2. GetAllEnrollData Method

Purpose: Fetches all enrollment data from the master device and saves it to the database if it does not already exist.

Key Steps:

- Enables the device using `sbxpc.SBXPCDLL.EnableDevice()`.
- Reads all user IDs using `sbxpc.SBXPCDLL.ReadAllUserID()`.
- Loops through each user to get enrollment data using `sbxpc.SBXPCDLL.GetAllUserID()`.
- Checks for existing data using `EnrollDataExist()`.
- Calls `SaveEnrollDB()` to insert new data into the database.
- Displays error messages for failed operations.
- Disables the device after processing using `sbxpc.SBXPCDLL.EnableDevice()`.

3. EnrollDataExist Method

Purpose: Checks if the enrollment data for a given enroll number and finger number already exists in the database.

4. SaveEnrollDB Method

Purpose: Inserts new enrollment data into the database if it does not already exist.

Key Steps:

- Creates a new DataRow for the enrollment data.
- Checks for duplicates to prevent redundant data.
- Converts fingerprint data to a byte array and saves it to the database.
- Handles special cases for passwords and facial data (finger numbers 10, 11, 15, 17).

5. ExitTimer_Tick Method

Purpose: Automatically closes the form after a 2-minute interval.

Key Steps:

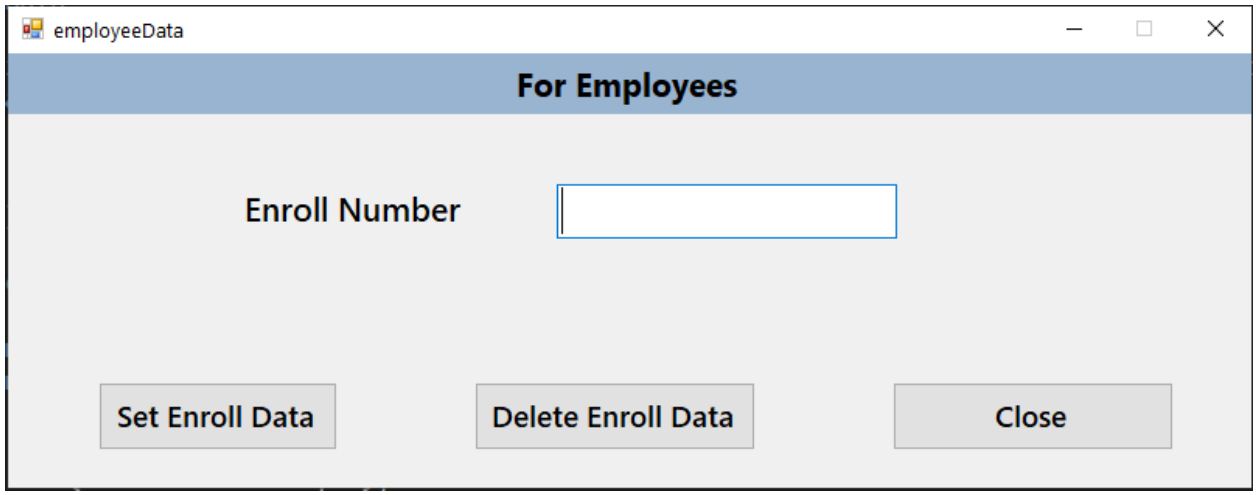
- Stops and disposes of the timer.
- Closes the form using `Close()`.

Modifications

```
3 references
public partial class masterDevice : Form
{
    1 reference
    public masterDevice()
    {
        InitializeComponent();
    }

    Dictionary<string, string> masterDeviceData = new Dictionary<string, string>
    {
        { "MachineID", "9"} ,
        { "IpAddress", "192.168.1.229" },
        { "PortNo", "5005" },
        { "Password", "0" }
    };

    int DATASIZE_FACE = (27668) / 4;
    int DATASIZE = (1404 + 12) / 4;
    int[] gTemplngEnrollData;
    int glngEnrollPData;
    string connectionString = "Data Source=LAPTOP-CHTUFO97;Initial Catalog=TblEnrollData;Integrated Security
    Byte[] gbytEnrollData;
    DataSet dsEnrolls;
```

Set masterDeviceData(Master Device) configuration in Dictionary as well as connectioString based on location site. As well as change connection string in EnrollData.cs

(Note : In Enroll Data Info. Section it shows the last fetched Enroll No from the device in Enroll No and how many records/data have been updated into the database in Total)



Employee Data SDK

Overview

The Employee Data Management application is designed to manage enrollment data for employees using biometric devices. The application connects to biometric devices to fetch, save, and delete employee enrollment data and stores it in a database.

Modules:

1. Send Enrollment Data to Devices
2. Delete Enrollment Data

1. Send Enrollment Data to Devices

- Method: `setButton_Click()`
- Purpose: Sends saved enrollment data to connected devices.
- Flow:
    - Connects to each device using `ConnectTcpip()`.
    - Iterates over each employee record and sends data using `SetEnrollData1()`.
    - Data is sent in the form of a byte array for fingerprints or face templates.

Error Handling:

- If the data transmission fails, the user is prompted to retry.

2. Delete Enrollment Data

- Method: `deleteEnrollButton_Click()`
- Purpose: Deletes specific enrollment data from connected devices.
- Flow:
    - Connects to each device using `ConnectTcpip()`.
    - Deletes the data using `DeleteEnrollData()` based on the specified `EnrollNumber` and `FingerNumber`.

Error Handling:

- If the deletion fails, an error message is displayed.

Database Structure

The database table used in this project is `TblEnroll` with the following structure:

Column Name	Data Type	Allow Nulls
EMachineNumber	int	☐
EnrollNumber	int	☐
FingerNumber	int	☐
Privilige	int	☑
Password1	int	☑
FPData	varbinary(MAX)	☑
		☐

Explanation of Columns:

- **EMachineNumber**: Represents the machine number associated with the enrollment data.
- **EnrollNumber**: The unique identifier for the enrolled employee.
- **FingerNumber**: Specifies the finger number associated with the enrollment data.
- **Privilige**: The privilege level assigned to the enrolled employee (nullable).
- **Password1**: The password associated with the employee (nullable).
- **FPData**: The fingerprint data stored as binary data (nullable).

The `FPData` column is used to store fingerprint data in binary format, enabling the application to handle large data sizes effectively.

Modifications

```
Dictionary<string, string> masterDeviceData = new Dictionary<string, string>
{
    { "MachineID", "9"} ,
    { "IpAddress", "192.168.1.229" },
    { "PortNo", "5005" },
    { "Password", "0" }
};

int DATASIZE_FACE = (27668) / 4;
int DATASIZE = (1404 + 12) / 4;
int[] gTemplngEnrollData;
int glngEnrollPData;
string connectionString = "Data Source=LAPTOP-CHTUFO97;Initial Catalog=TblEnrollData;Integrated Security
Byte[] gbytEnrollData;
DataSet dsEnrolls;
```

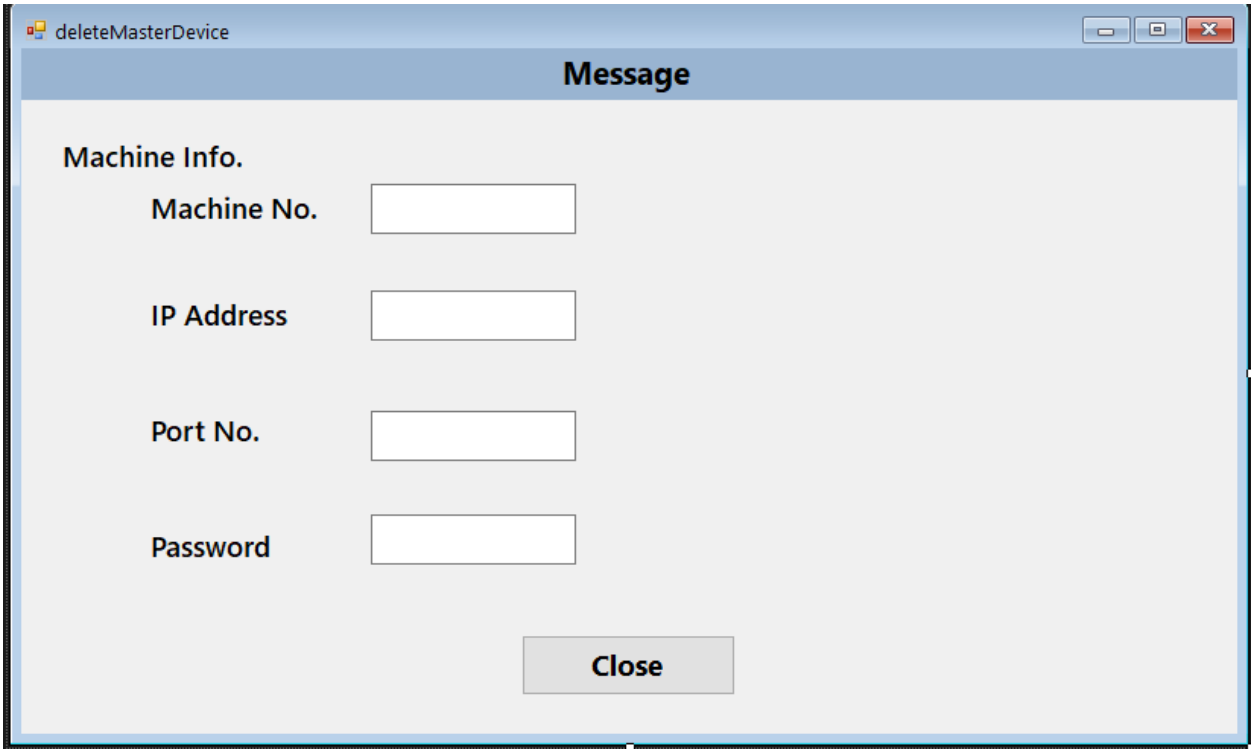
Change masterDeviceData (master machine) data and connection string according to the location site.

```
public employeeData()
{
    InitializeComponent();

    machineInfo.Columns.Add("MachineIP", typeof(string));
    machineInfo.Columns.Add("MachineNumber", typeof(int));

    machineInfo.Rows.Add("192.168.1.225", 1);
    machineInfo.Rows.Add("192.168.1.226", 2);
    machineInfo.Rows.Add("192.168.1.227", 3);
    machineInfo.Rows.Add("192.168.1.228", 4);
}
```

Set other devices data like Machine IP and Machine no based on requirements or configuration and you can add another machine data by entering machineInfo.Rows.Add("x",Y).

As well as change connection string in EnrollData,cs .

(**Note : When an employee register in the master machine and use Master Device SDK to save those data to the database . That Employee SDK will send data of that employee to the all available devices in the plant. Same goes for delete button/method.)**



Master Device Delete SDK

Overview:

The Master SDK Delete module is responsible for deleting enrollment data only from the master biometric device. The module interacts with the SBXPC SDK to connect to the master device, enable/disable it, and delete the enrollment data. This document provides an overview of the functionality, implementation, and key functions of the Master SDK Delete module.

Functionalities:

1. **Data Deletion from Master Device Only:** The module connects to the master device using TCP/IP and deletes enrollment data based on the records retrieved from the `TblEnroll` table in the database.
2. **User Interface:** The module displays the machine details such as Machine ID, IP Address, Port No, and Password in read-only text boxes. It also shows the status of the deletion process.

Implementation Details:

- The `deleteMasterDevice_Load` method is executed when the form is loaded. It initializes the UI, connects to the master device, and deletes the data.
- The `deleteData()` method retrieves all records from `TblEnroll` and deletes each record from the master device using the `DeleteEnrollData` method of the SBXPC SDK.

Key SDK Functions Used:

- `ConnectTcpip(int machineID, string ipAddress, int port, int password)` - Connects to the master device.
- `EnableDevice(int machineID, int enable)` - Enables or disables the master device.
- `DeleteEnrollData(int machineID, int enrollNumber, int backupNumber, int fingerNumber)` - Deletes the enrollment data from the master device.
- `Disconnect(int machineID)` - Disconnects the master device.

Error Handling:

- If the connection to the master device fails, an error message "No device Found" is displayed.
- If enabling the master device fails, an error message "Device Communication Fail" is displayed.

Modifications

```
Dictionary<string, string> masterDeviceData = new Dictionary<string, string>
{
    { "MachineID", "9"} ,
    { "IpAddress", "192.168.1.229" },
    { "PortNo", "5005" },
    { "Password", "0" }
};

string connectionString = "Data Source=LAPTOP-CHTUFO97;Initial Catalog=TblEnrollData;Integrated Security
1 reference
```

Change masterDeviceData(Master machine) and connection string as per the requirement.
Also change connection string in EnrollData,cs .

Help Desk/ Assistance

For further any Assistance Pls. Contact;

Aadarsh
Ph:-7999669691
E-mail:-aadarsh@comsysit.com