Thakur College of Science and Commerce

A
PROJECT REPORT
ON

ELECTRICA (Eleclectricity Bill Management System)

By

HARIKRISHNAN SATHYAN KONGIPARAMBIL

Submitted in partial fulfilment of Bachelors of Science (Computer Science)

[UNIVERSITY OF MUMBAI]

Thakur College of Science and Commerce Kandivali (East), Mumbai.

ACADEMIC YEAR 2021 - 2022

Thakur Educational Trust's (Regd.)

Thakur College of Science & Commerce

UGC Recognised • Affiliated to University Of Mumbai
(NAAC Accredited with Grade "A" [3rd Cycle] & ISO 9001:2015 Certified)

COMPUTER SCIENCE DEPARTMENT

(2021-2022)

CERTIFICATE OF APPROVAL

This is to certify that the project work entitled **"ELECTRICA (Electricity Bill Management System)"** is prepared by **HARIKRISHNAN SATHYAN KONGIPARAMBIL** a student of **"Third Year Bachelor of Science (Computer Science)"** course of University of Mumbai, which is conducted by our college.

This is the original study work and important sources used have been duly acknowledged in the report. The report is submitted in partial fulfilment of B.Sc. (Computer Science) course as per rules of University of Mumbai.

Project Guide: Ms Drashti Shrimal **Head of Department**

ACKNOWLEDGEMENT

Achievement is finding out what you would be doing rather than what you have to do. It is not until you undertake such a project that you realizehow much effort and hard work it really is, what are your capabilities and how well you can present yourself or other things. It tells us how much we rely on the efforts and goodwill of others. It gives me immense pleasure to present thisreport towards the fulfilment of my project.

It has been rightly said that we are built on the shoulder of others. for everything I have achieved, the credit goes to all those who had helped me tocomplete this project successfully.

I take this opportunity to express my profound gratitude to management of Thakur Degree College of Science & Commerce for giving me this opportunity to accomplish this project work.

A special vote of thanks to our HOD **Mr. Ashish Trivedi** and toour project guide **Ms. Drashti Shrimal** for helping and guiding me throughout my project.

Finally, I would like to thank all my friends & entire Computer Science department who directly or indirectly helped me in completion of thisproject & to my family without whose support, motivation & encouragement this would not have been possible.

(Harikrishnan Sathyan)

INDEX

CONTENTS

CHAPTER 5 - SYSTEM DESIGNS

CHAPTER 6- CONCLUSION & BIBLIOGRAPHY

CHAPTER 1 - INTRODUCTION

1.1 Introduction to the Project

Electrica is an electricity bill management system, which mainly focuses on reducing the work load on the administrator, making the billgeneration process digital and with user friendly software interface it allows the user to manage the consumers as well as generating their bills, sending the bill to consumers email id also printing at administrators end and also completing the payment and updating the information in the company database.

Contribution

- Electricity Billing System will help to manage the details of Electricity, Bill, Connections, Store Record, Consumer.
- It manages all the information about Electricity, Electricity Board, Consumer.
- The project is totally built at administrative end and thus only administrator is guaranteed the access
- The purpose of the project is to build a software to reduce the manual work for managing the Electricity , Bill, Electricity Board, Connections.
- It tracks all the details about the Connections, Store Records, Consumers.
- Provide consumer a hassle free experience while receiving the bills and during the money payment

1.2 Statement about the Project

- Digital transformation is often viewed as an implementation of digital technologies into all areas of business in order to build more sustainable relationships and better understand the needs of customers.
- Electricity consumption is increasing day by day so proper management of the bills and the payment is required.
- If there is any unexpected hike in the electricity consumption the consumer is not notified.
- Consumer does not gets a copy of the electricity bill through e-mail.
- Consumer should get Information from the Electricity board about any power outage situation.
- The fraud and the defaulter consumers should be warned.
- Bill payment should be managed properly.

1.3 Proposed Software (What would s/w accomplish?)

The proposed software is a desktop based application that will manage the consumer information in the Oracle database. Add or retrieve the consumer information or the billing data into the Oracle database, Calculate the bill using the Oracle stored procedure which is in contrast with the python's Tkinter GUI .Which gives the user hassle free exprence while working on this software. New consumers can be added, their data can be edited, and deleted at any point of time. Consumers bill can be generated as well as send the bill to the consumers email id in pdf format. Warning messages and alert message can be sent to the cosumer through e-mail and even through whatsapp message which are preloaded in the software and can also edit those messages before sending.

1.4 Benefits & Limitations

- **Benefits**

  Electrica is a software the helps the admin the manage the consumers and their details in a database, communicate with the consumers though emails and whatsapp messages, generage their electricity bill effeciently with zero errors. Send the generated bill to the consumers email id in pdf format. Generate the bill payment and defaulter consumer reports and also make payment for the bill.

- **Limitations**

  Since Electrica runs offline and it is admin oriented software consumers cannot perform any operation in the software , only the authorized people can make use of all the benefits of the software. Consumer cannot pay their bill online it has to be through the admin of the software only

CHAPTER 2 - METHODOLOGY

2.1 Objective and scope of the project

- The main objective of this project is to make the bill generation computerized.
- Storing data of bill generated in the database and fetching it according to the requirement.
- Maintaining consumers (Adding, Deleting, Updating).
- Sending bill to the consumers through e-mail, and also printing at the admins end.
- Sending WhatsApp alerts if any power outage is detected.
- High power consumption alerts.
- Power consumption report generation.
- Generate defaulters list and send warning through e-mail.
- Fraud customer report generation.
- Direct bill payment for consumers.
- Sending payment acknowledgement through e-mail.

2.2 Methodology

Programming language : **Python**

IDLE : **PyCharm**

Database: **Oracle**

The main approach

The main modules of python used in the project are as follows :

smtplib (Sending email to the consumers email id)

FPDF (For creating the pdf of bill which is generated)

sendpdf (send pdf file of bill and receipt to the consumers email id)

cx_Oracle (To connect python with the oracle database, perform CRUD operations and also for charge calculation)

Tkinter and PIL (For GUI and Images)

Steps involved in exection of software are as follows

1) Electrica has a home window where all the operation are present like adding new consumer, Editing consumer details, Enter reading, Generate bill, Send alerts, Defaulters, Send bill, and payment

2) In add consumer section the admin can add a new user into the system and if any entry was wrong then it can be edited later in the edit consumer section

3) Admin has enter the meter readings is the enter reading section. After entering the meter readings the bill will be calculated in the generate bill seciton

4) Generating the bill by entering the consumer id in generate bill section and thereby sending bill to the consumer

5) Admin can see the defaulters list and generate report of the billing.

6) Bill payment section and sending receipt to the consumer through e-mail.

CHAPTER 3 – CHOICE OF PLATFORMS S/W AND H/W

3.1 Front End

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

3.2 Back End

Oracle Database (commonly referred to as Oracle DBMS or simply as Oracle) is a multi-model database management system produced and marketed by Oracle Corporation. It is a database commonly used for running online transaction processing (OLTP), data warehousing (DW) and mixed (OLTP & DW) database workloads. Oracle Database is available by several service providers on-prem, on-cloud, or as hybrid cloud installation.

3.3 Platform Used

The Software is targeted for any GUI based Windows, Linux or MacOS device.

3.4 System Specifications

MINIMUM HARDWARE REQUIREMENT

- **Memory Space:**
  Minimum – 1024 MB
  Recommended – 2084 MB
- **HDD** (To install the software at least 2 GB and the data storage is depending upon the organizational setup.)
- **Processor**: Intel Pentium IV, 1GHZ or above
- **Ram**: 1024MB or above
- **Video**: 1024×768, 24-bit colours
- **Keyboard:** Standard 104 Keys
- **Internet Speed**: 1 Mbps or above

MINIMUM SOFTWARE REQUIREMENT

- **Programming Language:** Python 3.9 or above
- **Database :** Oracle Sql developer, Microsoft Exel

CHAPTER 4 - CHART

4.1 Activity chart

THAKUR COLLEGE OF SCIENCE AND COMMERCE

Department of Computer Science

2021-2022

Student Name: HARIKRISHNAN SATHYAN KONGIPARAMBIL

Project Name: ELECTRICA (ELECTRICITY BILL MANAGEMENT SYSTEM)

PHASES	EXPECTED DATE OF COMPLETION	ACTUAL DATE OF COMPLETION
Preliminary Investigation	10/08/21	10/08/21
System Analysis	15/08/21	15/08/21
System Designing	16/09/21	16/09/21
System Coding	18/10/21	15/10/21
System Implementation	20/10/21	20/10/21
Report Submission	25/10/21	25/10/21

CHAPTER 5 – SYSTEM DESIGNS

5.1 System code

Find the complete code on Github

Imported python libraries and packages used in the project :

```
import pyautogui
from tkinter import *
from PIL import ImageTk,Image
from tkinter import messagebox
import cx_Oracle
import pygetwindow
import webbrowser
from plyer import notification
import time
from pdf_mail import sendpdf
import datetime
import smtplib
from email.message import EmailMessage
from fpdf import FPDF
```

Python function for sending email in the project

```
def sendmail():
    con = cx_Oracle.connect('system/12345@localhost:1521/xe')
    cursor = con.cursor()
    x = cursor.execute("SELECT * FROM ADD_CONSUMER WHERE CON_ID =
(SELECT MAX(CON_ID) FROM ADD_CONSUMER)")
    values = x.fetchall()
    for i in values:
        name = i[1]
        supply = i[10]
```

```
        requrement = i[14]
        email = i[7]
        # Create an object of sendpdf function
        k = sendpdf("electrica.org@gmail.com",
                    f"{email}",
                    "Electrica@1234",
                    "Electrica New Connection",
                    f"Dear {name} ,\nYour connection request for {supply} ({requrement})
current supply has been approved.\nConnection will be established within
24hrs.\n\nRegards,\nElectrica",
                    "Receipt",
                    "C:/Users/Vandana/Documents/Clg Doc/OneDrive/ProjectGit/Electrica")

    # sending an email
        k.email_send()
```

Function for the home window (GUI):

```
def homeWindow():
    global home
    home = Tk()

    home.configure(bg="white")
    home.title('Electrica 2.0.1')
    home.iconbitmap("Images/icon2.ico")
    home.resizable(False, False)
    window_width, window_height = 885, 700

    screen_width = home.winfo_screenwidth()
    screen_height = home.winfo_screenheight()
```

```
    position_top = int(screen_height / 2 - window_height / 2)
    position_right = int(screen_width / 2 - window_width / 2)


home.geometry(f"{window_width}x{window_height}+{position_right}+{position_top}
")

    temp_size = Image.open("Images/home_template3.png")
    temp_resized = temp_size.resize((395, 704), Image.ANTIALIAS)
    template = ImageTk.PhotoImage(temp_resized)
    template_image = Label(home, image=template, borderwidth="0")
    template_image.place(x="-3", y="-3")

    backtemplate_size = Image.open("Images/backtemp1.png")
    backtemplate_resized = backtemplate_size.resize((350, 690), Image.ANTIALIAS)
    backtemplate = ImageTk.PhotoImage(backtemplate_resized)
    backtemplate_image = Label(home, image=backtemplate, borderwidth="0")
    backtemplate_image.place(x="463", y="0")

    addcon_size = Image.open("Images/adconsumer_btn.png")
    addcon_resized = addcon_size.resize((220, 50), Image.ANTIALIAS)
    addcon_image = ImageTk.PhotoImage(addcon_resized)
    Label(image=addcon_image)
    button_receipt = Button(home, image=addcon_image, borderwidth="0",
activebackground='blue',command=consumerEntry)
```

```
    button_receipt.place(x=530, y=30)

    editdetails_size = Image.open("Images/edit_details_btn.png")
    editdetails_resized = editdetails_size.resize((220, 50), Image.ANTIALIAS)
    editdetails_image = ImageTk.PhotoImage(editdetails_resized)
    Label(image=editdetails_image)
    button_editreceipt = Button(home, image=editdetails_image, borderwidth="0",
activebackground='blue',command=editwindow)
    button_editreceipt.place(x=530, y=100)

    readings_size = Image.open("Images/readings_btn.png")
    readings_resized = readings_size.resize((220, 50), Image.ANTIALIAS)
    readings_image = ImageTk.PhotoImage(readings_resized)
    Label(image=readings_image)
    button_readings = Button(home, image=readings_image,
borderwidth="0",activebackground="blue",command=enterReadings)
    button_readings.place(x=530, y=170)

    generatebill_size = Image.open("Images/generatebill_btn.png")
    generatebill_resized = generatebill_size.resize((220, 50), Image.ANTIALIAS)
    generatebill_image = ImageTk.PhotoImage(generatebill_resized)
    Label(image=generatebill_image)
    button_generatebill = Button(home, image=generatebill_image,
borderwidth="0",activebackground="blue",command=generatebillWindow)
    button_generatebill.place(x=530, y=240)
```

```
    sendalert_size = Image.open("Images/sendalert_btn.png")
    sendalert_resized = sendalert_size.resize((220, 50), Image.ANTIALIAS)
    sendalert_image = ImageTk.PhotoImage(sendalert_resized)
    Label(image=sendalert_image)
    button_sendalert = Button(home, image=sendalert_image,
borderwidth="0",activebackground="blue",command=alerts)
    button_sendalert.place(x=530, y=310)

    defaulter_size = Image.open("Images/defaulters_btn.png")
    defaulter_resized = defaulter_size.resize((220, 50), Image.ANTIALIAS)
    defaulter_image = ImageTk.PhotoImage(defaulter_resized)
    Label(image=defaulter_image)
    button_defaulter = Button(home, image=defaulter_image,activebackground="blue",
borderwidth="0")
    button_defaulter.place(x=530, y=380)

    fraud_size = Image.open("Images/sendbill_btn.png")
    fraud_resized = fraud_size.resize((220, 50), Image.ANTIALIAS)
    fraud_image = ImageTk.PhotoImage(fraud_resized)
    Label(image=fraud_image)
    button_fraud = Button(home, image=fraud_image,
borderwidth="0",activebackground="blue",command=sendbill)
    button_fraud.place(x=530, y=450)

    payment_size = Image.open( "Images/payment_btn.png")
    payment_resized = payment_size.resize((220, 50), Image.ANTIALIAS)
```

```
    payment_image = ImageTk.PhotoImage(payment_resized)
    Label(image=payment_image)
    button_payment = Button(home, image=payment_image,
borderwidth="0",activebackground="blue",command=billPayment)
    button_payment.place(x=530, y=520)

    exit_size = Image.open("Images/Exit_buttons.png")
    exit_resized = exit_size.resize((65, 25), Image.ANTIALIAS)
    exit_image = ImageTk.PhotoImage(exit_resized)
    Label(image=exit_image)
    button_exit = Button(home, image=exit_image, borderwidth="0",
activebackground="blue",command=exitapp)
    button_exit.place(x=805, y=660)

    devinfo = Label(home,text="Developed by Harikrishnan Sathyan.",font="lucida 9
",bg="white",fg="grey60")
    devinfo.place(x=395,y=675)
    home.mainloop()
```

Function to insert consumer details into the oracle database:

```
def addconsumerdb():
    name = conname_entry.get()
    phone = conphone_entry.get()
    address1 = address1_entry.get()
```

```
    address2 = address2_entry.get()
    address3 = address3_entry.get()
    pincode = pincode_entry.get()
    email = email_entry.get()
    aadhar = aadhar_entry.get()
    pan = pan_entry.get()
    supply = click.get()
    pos = var.get()
    meterno = meter_entry.get()
    requirement = click2.get()

    if (supply == 'SINGLE PHASE' and (requirement=='Up to 5 kW' or requirement=='5-
10 kW')):
        cc = 2050
    elif (supply == 'THREE PHASE' and requirement=='10-20 kW'):
        cc = 4575
    elif  (supply == 'THREE PHASE' and requirement=='20-50 kW'):
        cc = 6575
    elif (supply == 'THREE PHASE' and requirement=='50-150 kW'):
        cc = 12075
    elif (supply == 'THREE PHASE' and requirement=='Above 150 kW'):
        cc = 250075
    else:
        cc = 2050
    try:
        con = cx_Oracle.connect('system/12345@localhost:1521/xe')
```

```
        print(con.version)
        cursor = con.cursor()
        cursor.execute(f"INSERT INTO ADD_CONSUMER
VALUES(consumer_seq.nextval,'{name}',{phone},'{address1}','{address2}','{address3}'
,{pincode},'{email}',{aadhar},'{pan}','{supply}','{pos}',{meterno},sysdate,'{requirement
}',{cc})")
        messagebox.showinfo("Message", "Consumer Added Successfully")
        cursor.close()
        con.commit()
        print('Consumer added!')
        displayentry()
        con.close()
    except Exception as e:
        messagebox.showerror("Error","Some error occured.\n\n○ Enter valid details. \n○
          Fields should not be empty.")
```

Function for password validation

```
def passCheck():
    try:
        identered = id_entry.get()
        passentered = pass_entry.get()
        if (passentered=='1234'):
            con = cx_Oracle.connect('system/12345@localhost:1521/xe')
            cursor = con.cursor()
```

```
        x = cursor.execute(f"SELECT * FROM ADD_CONSUMER WHERE CON_ID =
{identered}")
        values = x.fetchall()
        for i in values:
            response = messagebox.askyesno("Ask Question", f"Are you sure you want to
delete details\n\nCONSUMER ID: {i[0]}\nNAME: {i[1]}\nMETER NO: {i[12]}")
            if response == True:
                cursor.execute(f"DELETE FROM ADD_CONSUMER WHERE
CON_ID={identered}")
                secure.destroy()
                messagebox.showinfo("MESSAGE",f"Consumer No {identered} deleted
from record.")
            elif response == False:
                pass
        cursor.close()
        con.commit()
        con.close()
        secure.destroy()
    else:
        messagebox.showinfo("MESSAGE","INVALID PASSWORD")
        secure.destroy()
  except Exception as e:
    messagebox.showerror("ERROR","Some Error Occured\n\n○ Enter Valid
Consumer_id\n○ Try again.")
    secure.destroy()
```

Function for inserting readings into the Oracle database:

```
def insertreadings():
    conid = conid_entry.get()
    meterreading = meterread_entry.get()
    entry_date = "01-"+str(month.get())
    try:
        con = cx_Oracle.connect('system/12345@localhost:1521/xe')
        cursor = con.cursor()
        x = cursor.execute(f"SELECT COUNT(*) FROM ADD_CONSUMER WHERE
CON_ID={conid}  ")
        list1  = x.fetchall()
        y = cursor.execute(f"SELECT COUNT(*) FROM METER_READING WHERE
CON_ID={conid} and reading_date='{entry_date}'")
        readings = y.fetchall()

        z = cursor.execute(f"select count(*) from charge_master_track  where
BILL_MONTH='{month.get()}'")
        cnt = z.fetchall()

        # l = cursor.execute(f"SELECT CURRENT_READING FROM
METER_READING WHERE CURRENT_READING=(SELECT
MAX(CURRENT_READING) FROM METER_READING WHERE
CON_ID={conid})")
        # val = l.fetchall()

        for i in list1:
```

```
        for value in readings:
            for predat in cnt:
                if (i[0]==0 ):
                    messagebox.showerror("Error",f"CON_ID {conid} Does not exist. ")
                    break
                if (predat[0]>0):
                    messagebox.showerror("Error","Billing already processed for this month")
                    break
                if (value[0] == 1):
                    response = messagebox.askyesno("Ask Question",f"CON_ID {conid}
Meter Reading already inserted\nfor current months billing.\n\nDo you want to edit and
update reading?")

                    if response == True:
                        readingUpdate()

                    elif response == False:
                        pass
                else:
                    cursor.execute(f"INSERT INTO METER_READING VALUES
({conid},{meterreading},'{entry_date}',sysdate)")
                    messagebox.showinfo("Message", "Readings Added Successfully!")
                    cursor.close()
                    con.commit()

                    con.close()
```

```
                meterread.destroy()

    except Exception as e:
        messagebox.showerror("Error", "Error occured\n\n ○ Entry field should not be Empty.\n ○ Enter Valid CON_ID\n ○ Meter reading should be greater than previous reading. ")
        print(Exception)
        print(e)
        print(entry_date)
```

Function for sending whatsapp alert message:

```
def submitalertmessage():
    try:
        alertconid = alertconid_entry.get()
        msg = alert_text.get("1.0",END)
        websitelink = ""
        paymentlink = ""

        print(msg)
        web = website.get()
        pay = payment.get()
        if web == 1:
            websitelink = "*Visit Website*:https://www.adanielectricity.com/"
        if pay == 1:
```

```
        paymentlink = "\n*Pay Bill*:https://www.adanielectricity.com/Payment/Online-Payments"

    con = cx_Oracle.connect('system/12345@localhost:1521/xe')
    cursor = con.cursor()
    checkcount = cursor.execute(f"SELECT COUNT(*) FROM ADD_CONSUMER WHERE CON_ID={alertconid} ")
    count = checkcount.fetchall()
    values = cursor.execute(f"SELECT * FROM ADD_CONSUMER WHERE CON_ID={alertconid}")
    phoneno = values.fetchall()
    for number in count:
        print(number[0])
        if (number[0]==0):
            sendsplash.destroy()
            messagebox.showerror("Error", f"CON_ID {alertconid} does not Exist.")
            break
        else:
            try:
                for i in phoneno:
                    now = datetime.datetime.now()
                    import pywhatkit as kit

kit.sendwhatmsg(f"+91{i[2]}",f"●●●●●●●\n*Alert*\n{msg}{websitelink}{paymentlink}\n⚠⚠⚠⚠⚠⚠⚠",now.hour, now.minute+1)
```

```
            except Exception as e:
                messagebox.showerror("ERROR","Network Error Occured\nPlease check
your Internet connection and Try Again")
                tryagainSplash()


        cursor.close()
        con.close()
    except Exception as e:
        messagebox.showerror("Error", "Some error occured.\n\n○ Enter valid details. \n○
          Fields should not be empty.")
```

Function for generating pdf

```
def pdfGeneration():
    consumer_id = sendbillconid_entry.get()
    pdf = FPDF('P', 'mm', (210, 297))


    pdf.add_page()

    pdf.set_font('helvetica', 'B', 8)
    pdf.set_text_color(0, 0, 0)

    pdf.image('Images/bill_toptemplate.png', 5, 5, 200, 47)
    pdf.image("Images/bill_nametemplate.png", 20, 55, 80, 80)
```

```
pdf.image("Images/bill_amounttemplate.png", 17, 135, 85, 65)
pdf.image("Images/bill_contacttemplate.png", 115, 60, 75, 37)
pdf.image("Images/bill_consuptiontemplate.png", 113, 103, 75, 50)
pdf.image("Images/bill_impmsgtemplate.png", 114, 155, 72, 35)
pdf.image("Images/bill_protocoltemplate.png", 114, 195, 72, 35)
pdf.image("Images/bill_cuttemplate.png", 5, 240, 200, 8)

name = "HARIKRISHNAN SATHYAN"
phone = 9820767941
pdf.text(30, 79, 'NAME          : ')
pdf.text(29.9, 84, 'PHONE NO  :')
pdf.text(30, 89, 'ADDRESS   :')
pdf.text(30, 104, 'PINCODE    :')
pdf.text(30, 109, 'EMAIL         :')
pdf.text(30, 114, 'AADHAR     :')
pdf.text(30, 119, 'CL in kW     :')

pdf.set_font('helvetica', 'B', 8)
pdf.set_text_color(0, 0, 0)
pdf.text(116, 103, "YOUR CURRENT CONSUMPTION")

pdf.set_font('helvetica', 'B', 7)
pdf.text(120, 114, "BILL NO                          :")
pdf.text(120, 120, "BILL DATE                       :")
pdf.text(120, 126, "TYPE OF SUPPLY              :")
pdf.text(120, 132, "PRESENT READING          :")
```

```
    pdf.text(120, 138, "PREVIOUS READING          :")
    pdf.text(120, 144, "CONSUPTION (UNIT kWh)   :")
    imp_msg = msg_text.get("1.0", END)
    pdf.set_xy(117, 166)
    pdf.multi_cell(200, 4,f"{imp_msg}")

    pdf.set_font('helvetica', 'B', 10)
    pdf.text(46, 209, "JOIN US ON")

    pdf.image("Images/bill_fbtemplate.png", 30, 213, 10, 10,
link="https://www.facebook.com/")
    pdf.image("Images/bill_instatemplate.png", 45, 213, 10, 10,
link="https://www.instagram.com/?hl=en")
    pdf.image("Images/bill_youtubetemplate.png", 60, 213, 10, 10,
link="https://www.youtube.com/")
    pdf.image("Images/bill_linkedintemplate.png", 75, 213, 10, 10,
link="https://in.linkedin.com/")

    # Check slip
    pdf.image("Images/bill_paysliptemplate.png", 7, 248, 8, 40)
    pdf.image("Images/bill_barcodetemplate.png", 20, 264, 100, 10)
    pdf.image("Images/bill_payslip2template.png", 20, 276, 145, 10)

    pdf.set_font('helvetica', '', 7)
    pdf.text(20, 250, "If paying by cheque, please remember:")
```

```
    pdf.text(20, 254, "- Cheque should be Account payee of local clearing and not post-dated")
    pdf.text(20, 258,
             "- Always attach payment slip. Do not staple          - Make cheque payable to Electrica Electricity Mumbai Ltd. A/C No.:152191709")
    pdf.text(20, 262,
             "- Mention A/c No. and respective amount on back of the cheque,when making multiple bill payments by single cheque")
    pdf.set_font('helvetica', 'B', 7)
    pdf.text(25, 280, "BILL DATE : ")
    pdf.text(96, 280, "BILL AMOUNT : ")
    pdf.text(25, 284.5, "DUE DATE : ")
    pdf.text(96, 284.5, "AMOUNT AFTER DUE DATE : ")

    # VALUES OF THE FIELDS



    con = cx_Oracle.connect('system/12345@localhost:1521/xe')
    cursor = con.cursor()
    consumer_details = cursor.execute(f"""
                            SELECT CON_NAME,PHONE_NO,ADDRESS1,ADDRESS2,ADDRESS3,PIN_CODE,EMAILID,AADHAR,
                            SUPPLY_TYPE,REQUIREMENT FROM ADD_CONSUMER WHERE CON_ID={consumer_id}
                             """)
```

```
    details_list = consumer_details.fetchall()
    for con_values in details_list:
        emailid = con_values[6]
        encpt_emailid = emailid[0:3] + "*******" + emailid[-10:]

        aadhar = str(con_values[7])
        encpt_aadhar = aadhar[0:3] + "******" + aadhar[-3:]

        pdf.set_font('helvetica', '', 7)
        pdf.text(50, 79, f"{con_values[0]}")
        pdf.text(50, 84, f"{con_values[1]}")
        pdf.text(50, 89, f"{con_values[2]}")
        pdf.text(50, 94, f"{con_values[3]}")
        pdf.text(50, 99, f"{con_values[4]}")
        pdf.text(50, 104, f"{con_values[5]}")
        pdf.text(50, 109, f"{encpt_emailid}")
        pdf.text(50, 114, f"{encpt_aadhar}")
        pdf.text(50, 119, f"{con_values[9]}")
        pdf.set_font('helvetica', 'B', 7)
        pdf.text(63, 62, f"{con_values[8]}")

    bill_details = cursor.execute(f"""
                            SELECT *  FROM CHARGE_MASTER_TRACK WHERE
CON_ID = {consumer_id} AND BILL_DATE=(SELECT MAX(BILL_DATE) FROM
CHARGE_MASTER_TRACK)
                             """)
```

```
bill_details_list = bill_details.fetchall()
for details in bill_details_list:
    pdf.set_font('helvetica', '', 7)
    pdf.text(155, 114, f"{details[14]}")
    pdf.text(155, 120, f"{str(details[2])[:11]}")
    pdf.text(155, 126, f"{details[1]}")
    pdf.text(155, 132, f"{details[4]}")
    pdf.text(155, 138, f"{details[3]}")
    pdf.text(155, 144, f"{details[5]}")
    pdf.text(40, 280, f"{str(details[2])[:11]}")
    bill_amt = str(details[7])
    o = ".00"
    pdf.text(115, 280, f"{bill_amt}{o}/- Rs")
    bill_amt_aftdue = str(details[7] + 50)
    o = ".00"
    pdf.text(131, 284.5, f"{bill_amt_aftdue}{o}/- Rs")
    pdf.set_font('helvetica', 'B', 14)
    pdf.text(51.5, 191.5, f"{bill_amt}{o}")

    pdf.set_font('helvetica', 'B', 8)

    pdf.text(65, 148, f"{details[0]}")
    pdf.text(65, 159, f"{details[15]}")

pdf.add_page()
```

```
    pdf.image("Images/bill_howtemplate.png", 7, 10, 110, 100)
    pdf.image("Images/bill_protocol2template.png", 135, 11, 60, 94)
    pdf.image("Images/bill_triff2template.png", 68, 120, 130, 35)
    pdf.image("Images/bill_trifftoptemplate.png", 66, 113, 135, 6)
    pdf.image("Images/bill_billsumtemplate.png", 6, 109, 60, 50)

    pdf.set_font('helvetica', 'B', 10)
    pdf.text(13, 20, "HOW BILL WAS CALCULATED ")
    pdf.set_font('helvetica', 'B', 7)

    pdf.text(13, 33, "FIXED CHARGE ")
    pdf.text(13, 38.5, "WHEELING CHARGE")
    pdf.text(13, 43.8, "REGULATORY ASSET CHARGE (RAC)")
    pdf.text(13, 49.2, "ENERGY CHARGE")
    pdf.text(13, 54.4, "FUEL ADJUSTMENT CHARGE (FAC)")
    pdf.text(13, 60, "GOVERNMENT ELECTRICITY DUTY
16%")
    pdf.text(13, 65.2, "MAHARASHTRA GOVT. TAX ON SALE OF ELECTRICITY
26.04 p/unit")
    pdf.text(13, 70.5, "CURRENT MONTHS BILL AMOUNT (A)")
    pdf.text(13, 75.8, "PREVIOUS MONTHS BILL AMOUNT")
    pdf.text(13, 81.3, "PROMPT PAYMENT DISCOUNT")
    pdf.text(13, 86.8, "NET PREVIOUS BALANCE (B)")
    pdf.text(13, 91.9, "TOTAL BILL AMOUNT (A+B)")
    pdf.text(69, 116, "KEEP A WATCH TO MANAGE YOUR ELECTRICITY
CONSUPTION")
```

```
    pdf.set_font('helvetica', '', 5)
    pdf.text(69, 118, "YOUR TRIFF STRUCTURE")

    pdf.set_font('helvetica', 'B', 9)
    pdf.text(11, 123, "ROUND SUM")
    pdf.text(11, 128, "PAYABLE")
    pdf.text(11, 133, "FOR THIS BILL")

    pdf.set_font('helvetica', 'B', 6)
    pdf.text(10, 143, "METER READING DATE")
    pdf.text(10, 149, "PREVIOUS METER")
    pdf.text(10, 151, "READING DATE")

    bill_details = cursor.execute(f"""
                            SELECT *  FROM CHARGE_MASTER_TRACK WHERE
CON_ID = {consumer_id} AND BILL_DATE=(SELECT MAX(BILL_DATE) FROM
CHARGE_MASTER_TRACK)
                             """)
    bill_details_list = bill_details.fetchall()
    for details in bill_details_list:
        pdf.set_font('helvetica', '', 8)
        pdf.text(103, 33, f"{details[8]}/-")
        pdf.text(103, 38.3, f"{details[10]}/-")
        pdf.text(103, 43.7, f"0.00/-")
        pdf.text(103, 49.3, f"{details[9]}/-")
```

```
        pdf.text(103, 54.4, f"0.00/-")
        pdf.text(103, 60, f"{details[11]}/-")
        pdf.text(103, 65.4, f"{details[4] * 0.26}/-")
        pdf.text(103, 70.5, f"{details[7]}/-")
        pdf.text(103, 75.5, f"{details[6]}/-")
        pdf.text(103, 81, f"0/-")
        pdf.text(103, 87.1, f"{details[6]}/-")
        pdf.text(103, 92, f"{details[7]}/-")

        pdf.set_font('helvetica', 'B', 9)
        pdf.text(43, 127, f"{details[7]}/- Rs")

    pdf.output('C:/Users/Vandana/Documents/Clg
Doc/OneDrive/ProjectGit/Electrica/Bill.pdf')
    billing.destroy()
    webbrowser.open_new(r'file://C:/Users/Vandana/Documents/Clg
Doc/OneDrive/ProjectGit/Electrica/Bill.pdf')
```

Find the complete code on [Github]

5.2 System Output

Home Page

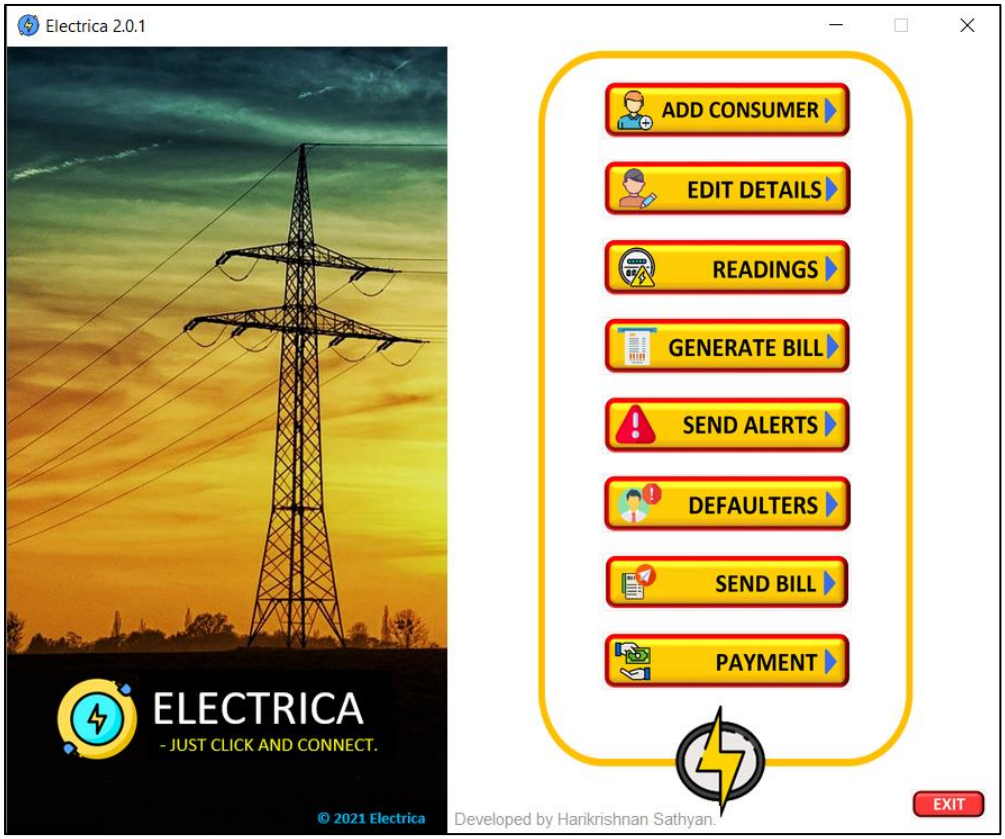


Fig 1.0

Add Consumer Window

Fig 1.1

Submit Without Entries Error (Exception handling)

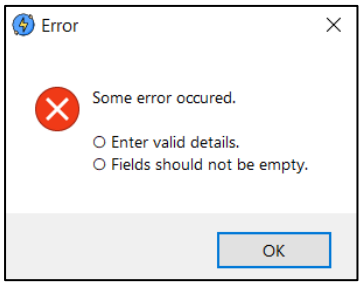


Fig 1.2

Submit without selection the declaration check box Error

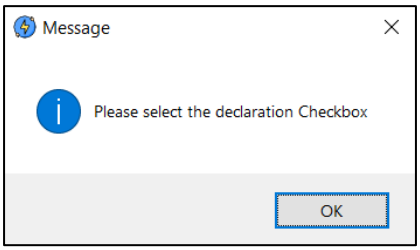


Fig 1.3

Filled Form view

Fig 1.4

Click submit after writing valid entries. (Message after submission)

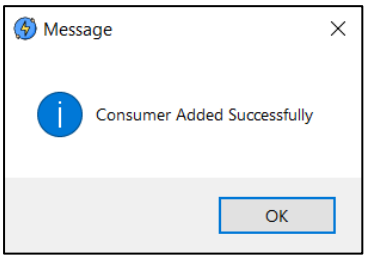


Fig 1.5

Displaying the details after clicking ok (Showing Receipt)

Fig 1.6

After clicking the print button

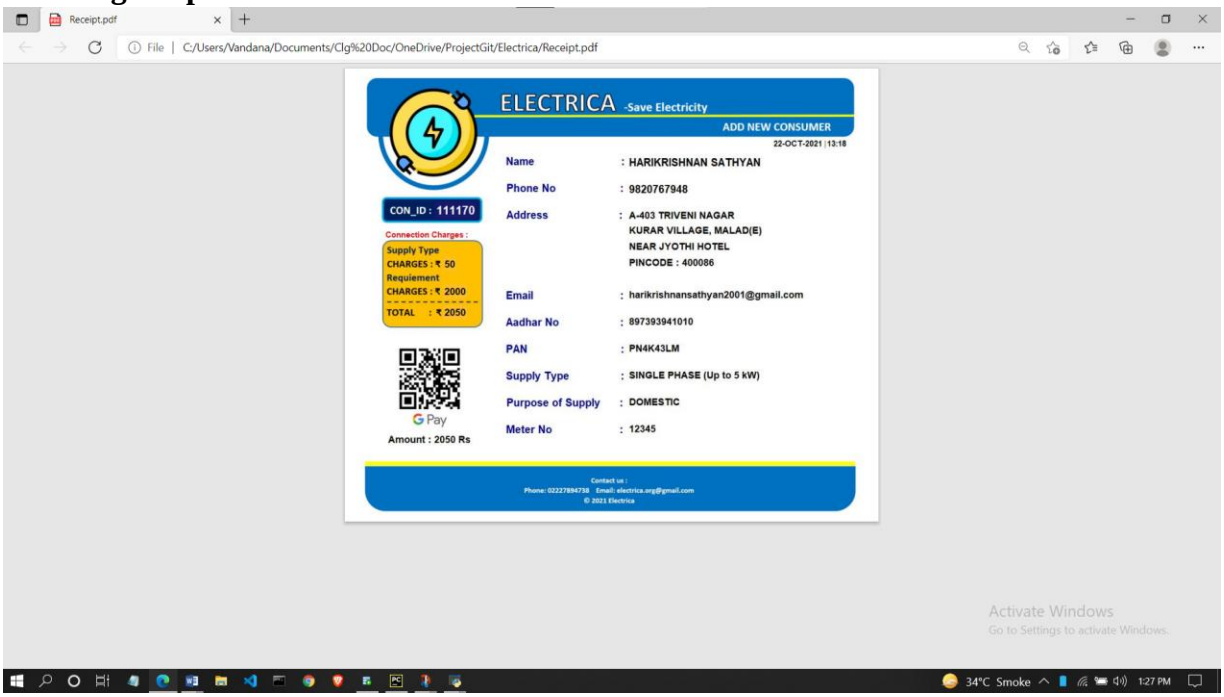


Fig 1.7

Printing the receipt

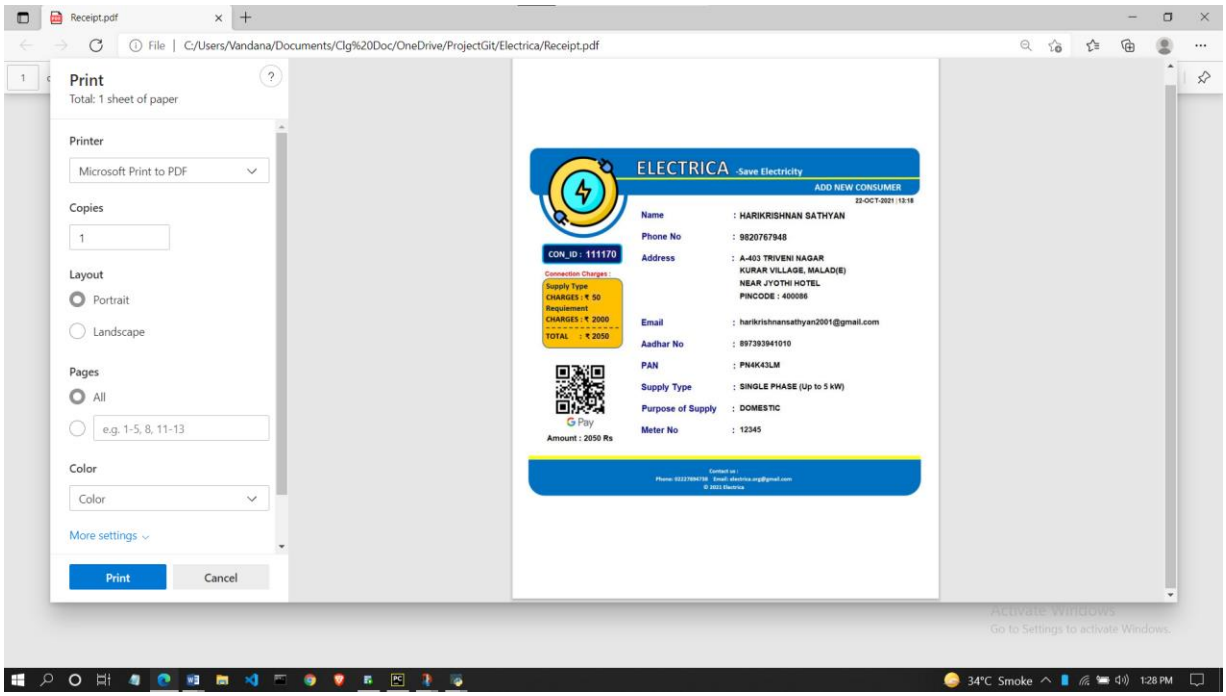


Fig 1.8

After clicking mail button (Splash window)

Fig 1.9

Mail sent messagebox

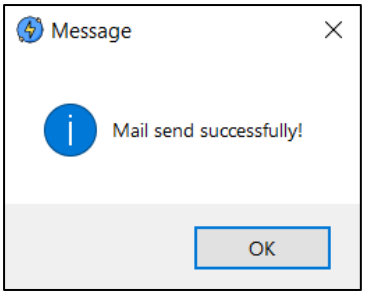


Fig 1.10

Received mail (Gmail)

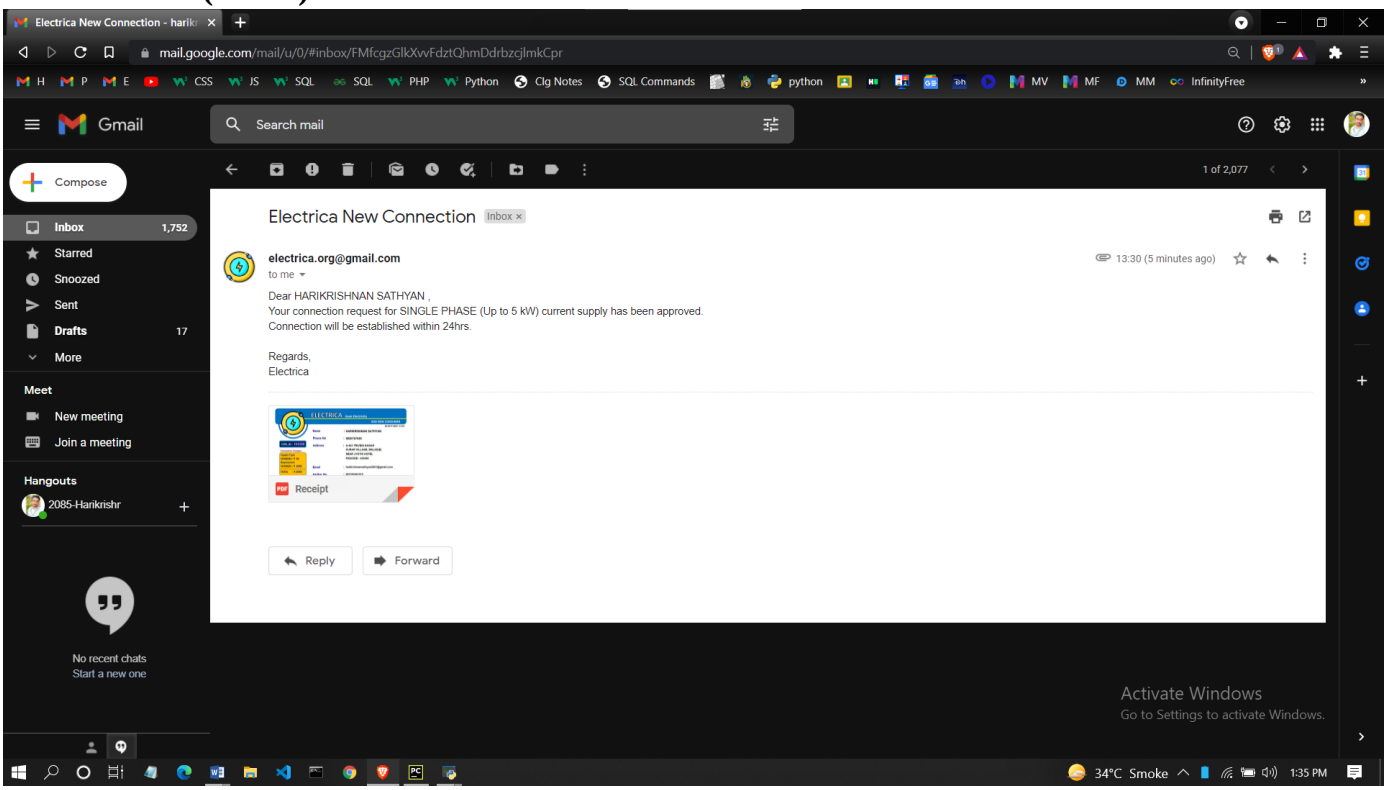


Fig 1.11

After clicking Edit Details button in home screen. As shown in Fig (1.0)

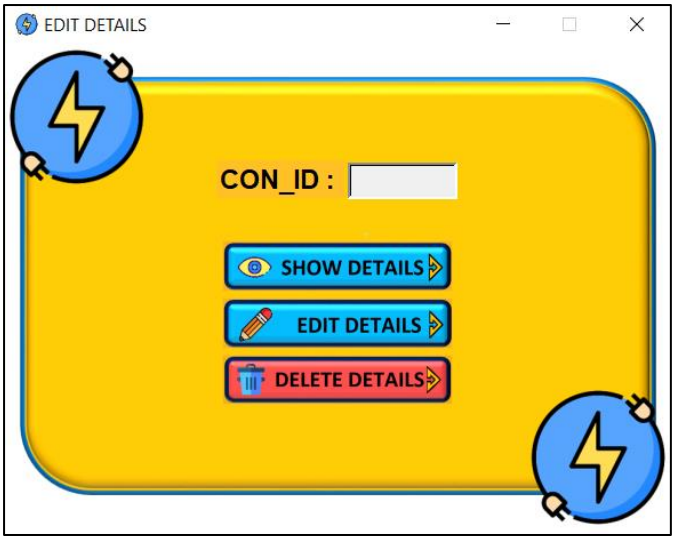


Fig 2.1

Clicking Show details/ Edit details/Delete details without entering the con_id [error] (Exception handing)

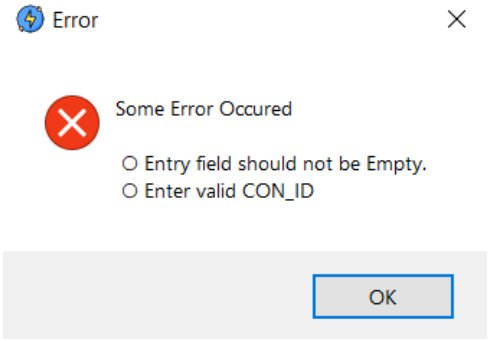


Fig 2.2

Error after entering invalid consumer id

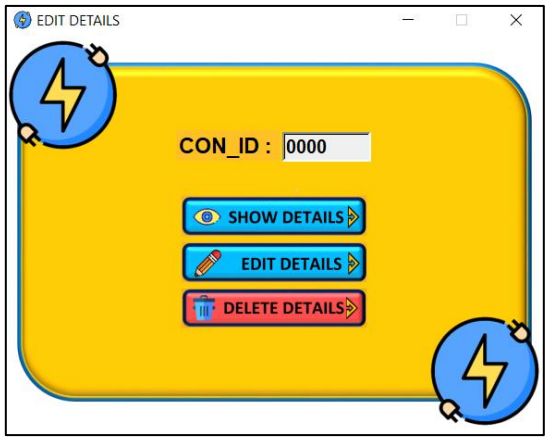


Fig 2.3

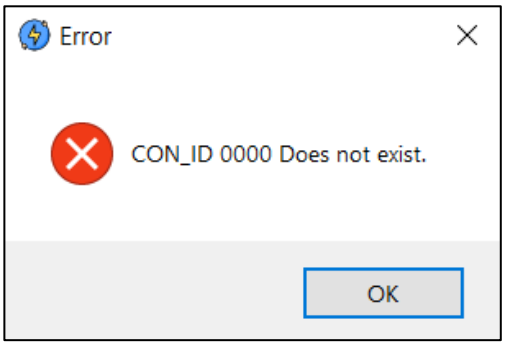


Fig 2.4

Clicking show details after entering consumer id as 111170

Fig 2.5

Clicking print and mail will print the receipt and mail the receipt respectively (As shown in Figs (1.7 to 1.11))

Clicking on the Edit details allows the editable window.

Fig 2.6

After editing the information click on Save changes. Then this Acknowledgement message box bill popup.

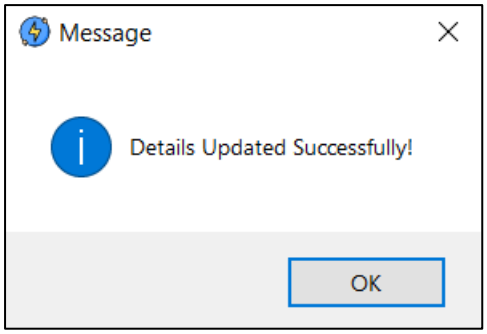


Fig 2.7

Clicking on Delete details button will ask for password

Fig 2.8

If the entered password in not valid then it will show error message (Invalid password error)

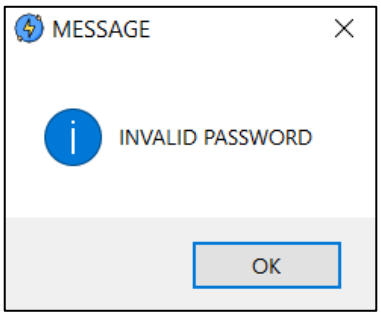


Fig 2.9

If the enterd password in correct then it will ask for conformation for deleting the details (Yes or no message box)

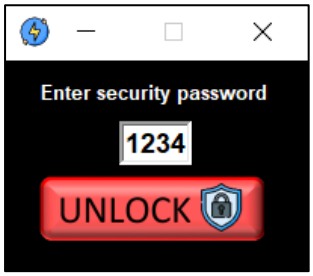


Fig 2.10

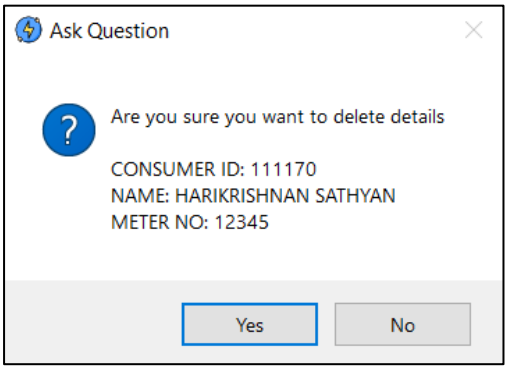


Fig 2.11

If no is clicked then entire process will stop and data will not be deleted.

If yes is clicked then that particular consumed data will be deleted and messagebox will popup.

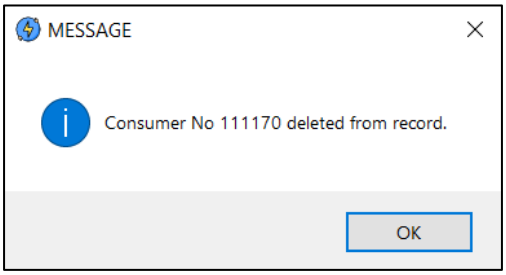


Fig 2.12

Click on Readings button in home page to insert meter readings of the consumer. Refer Fig 1.0 Meter reading window

Fig 3.0

If submit button is clicked without entering any value then Error error message will appear same as Fig 1.2. (Exception)

If invalid consumer id is entered then Consumer does not exist message will appear. (As shown in Fig 3.2)

Also the meter reading entering should be greater than previous reading if not then Error message will a popup.

(As shown in Fig 3.1)

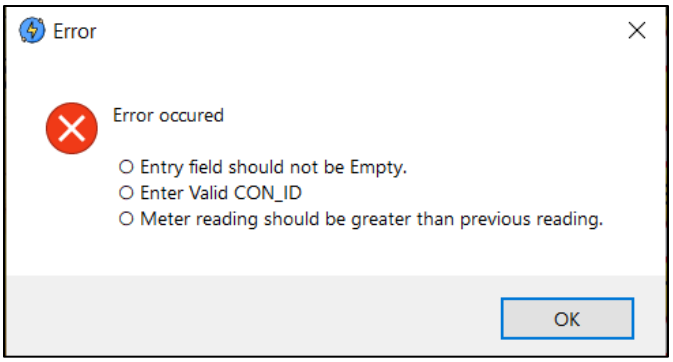


Fig 3.1

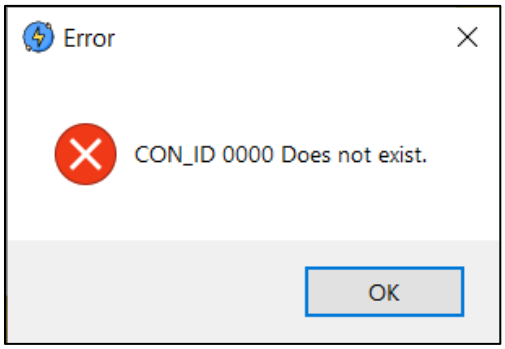


Fig 3.2

Click submit after entering the readings.

Fig 3.3

After clicking submit readings will be inserted and message will popup.

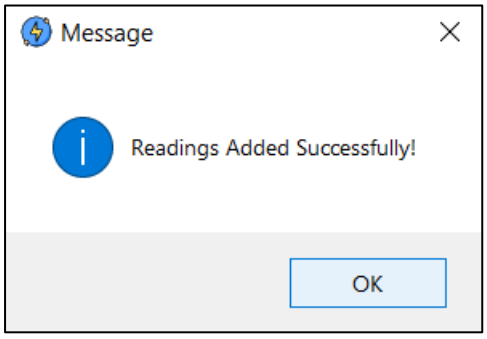


Fig 3.4

If bill generation is already done for a particular month then admin cannot edit readings and error message will popup.

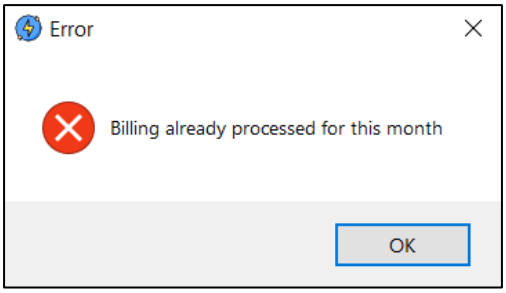


Fig 3.5

If reading of a consumer is added already for a particular month but the bill generation of that particular mothn is not done yet. if admin try to resubmit new reading than update reading window will popup

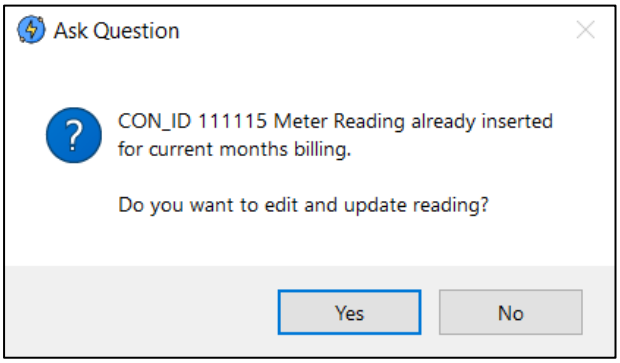


Fig 3.6

If admin clicks yes then edit reading window will appear.

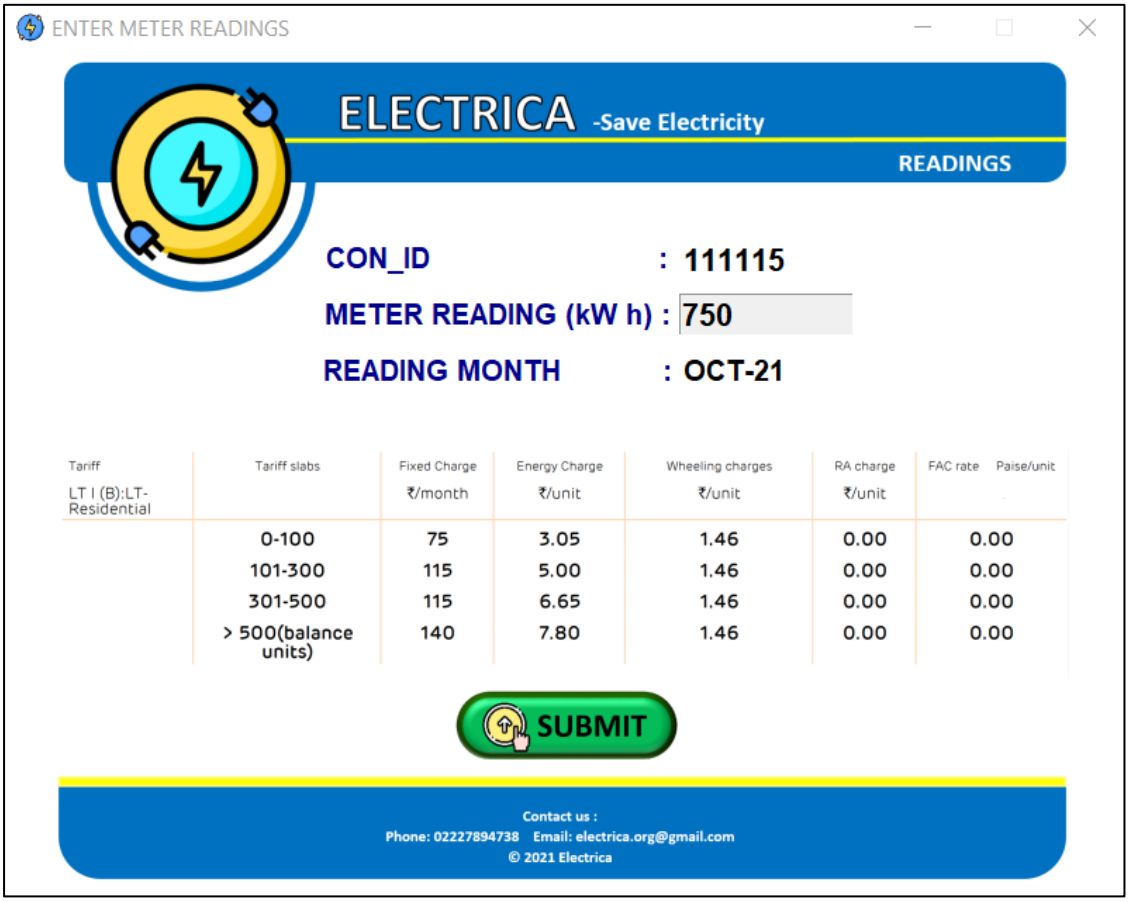


Fig 3.7

After entering the updated reading click on submit and message box will popup.

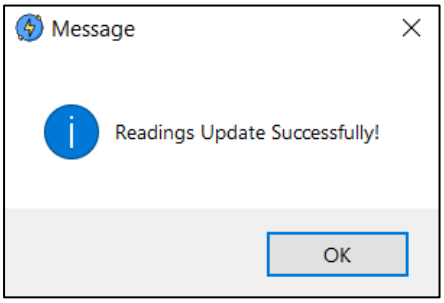


Fig 3.8

Generate bill button in the home screen will is used to generate the bill of consumers of a particular month selected.
After clicking on generate bill button bill month window will popup where the bill month has to be selected.

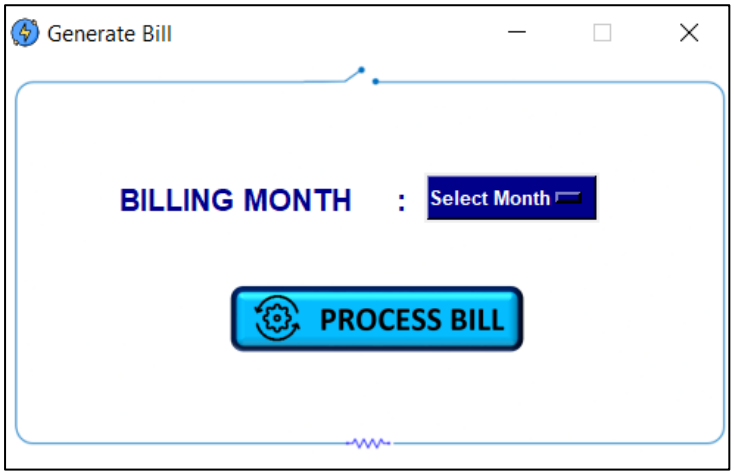


Fig 8.0

After selecting the month clicking on process bill button will start the bill generation process.

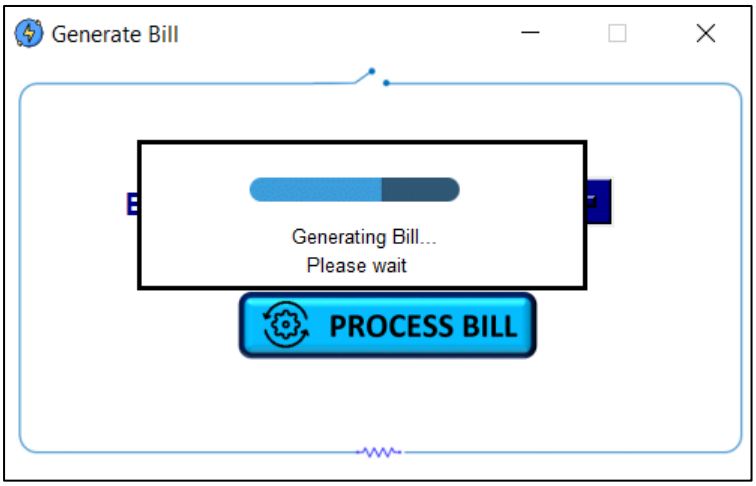


Fig 8.1

When the bill is generated message box will apper

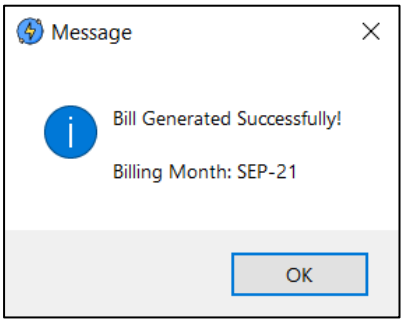


Fig 8.2

Send Alert button in the home windown (Fig 1.0) will send email and whatsapp alert to the consumer

Fig 5.0

If the fields are kept empty then error message will pop up.

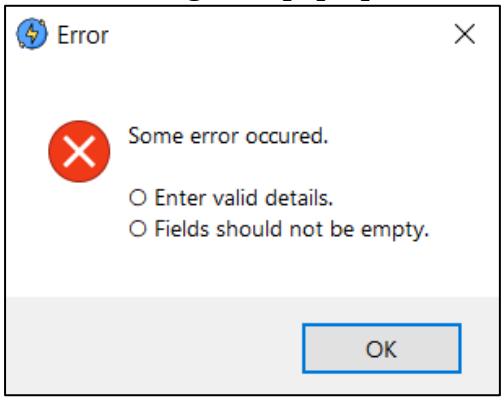


Fig 5.1

If invalid consumer id is entered then id does not exist error will pop up.

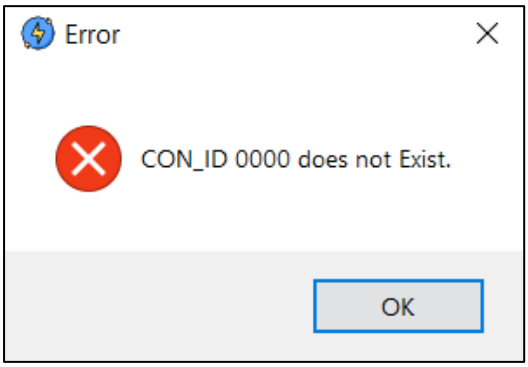


Fig 5.2

There are default prewritten messages option named as message 1, message2…. Buttons. And select the check box to include the website link and the the payment method link in the alert message.

Fig 5.3

After clicking on Whatsapp button the message will be sent to the consumers whatsapp number.

Fig 5.4

Message is sent to consumer in whatsapp.

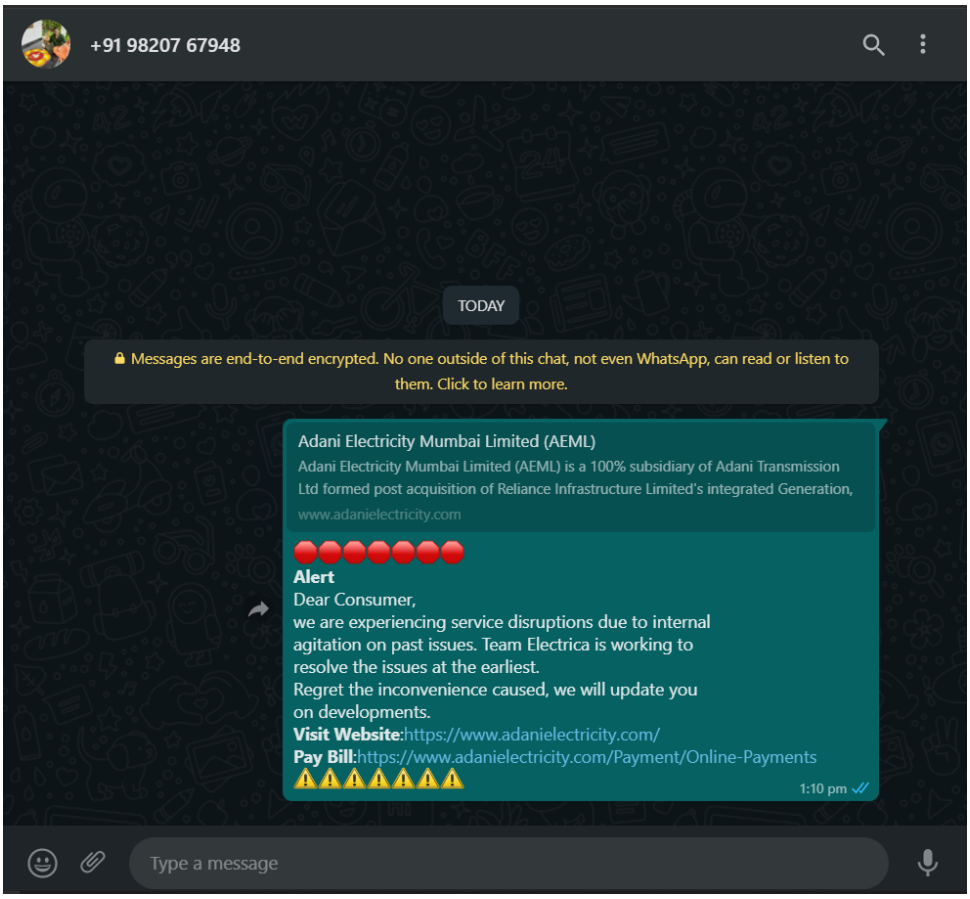


Fig 5.5

After clicking on Email button the alert message will the sent to the comsumers email id.

Fig 5.6

After sending message box will pop up.

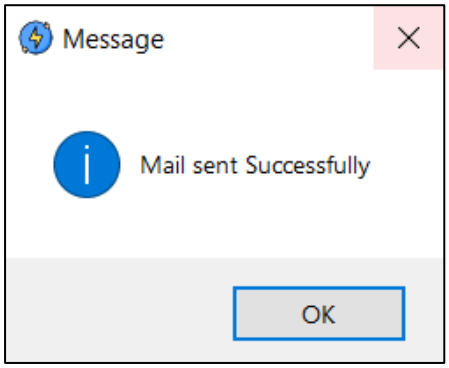


Fig 5.7

Email Received by the consumer.

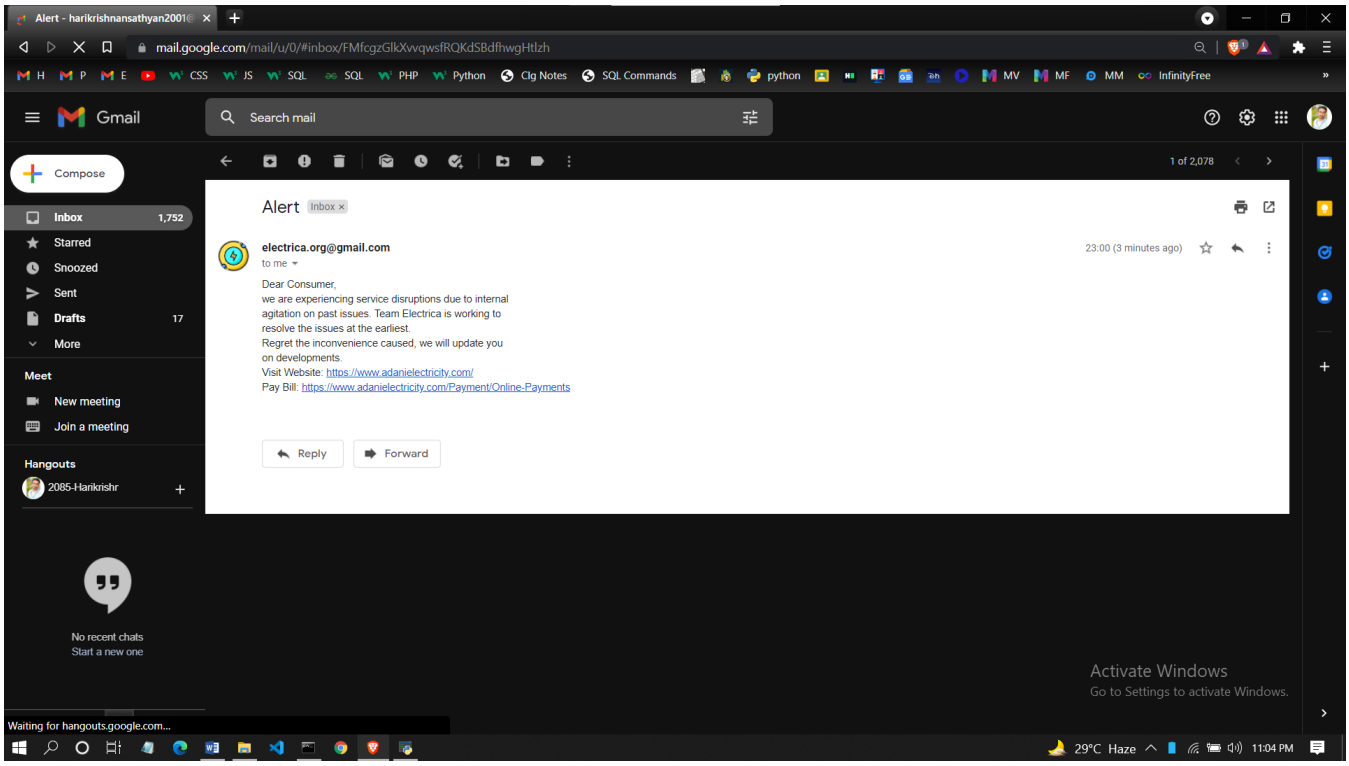


Fig 5.8

Clicking on send bill will open interface to send the Bill of the consumer by writing the consumer id or bill number.

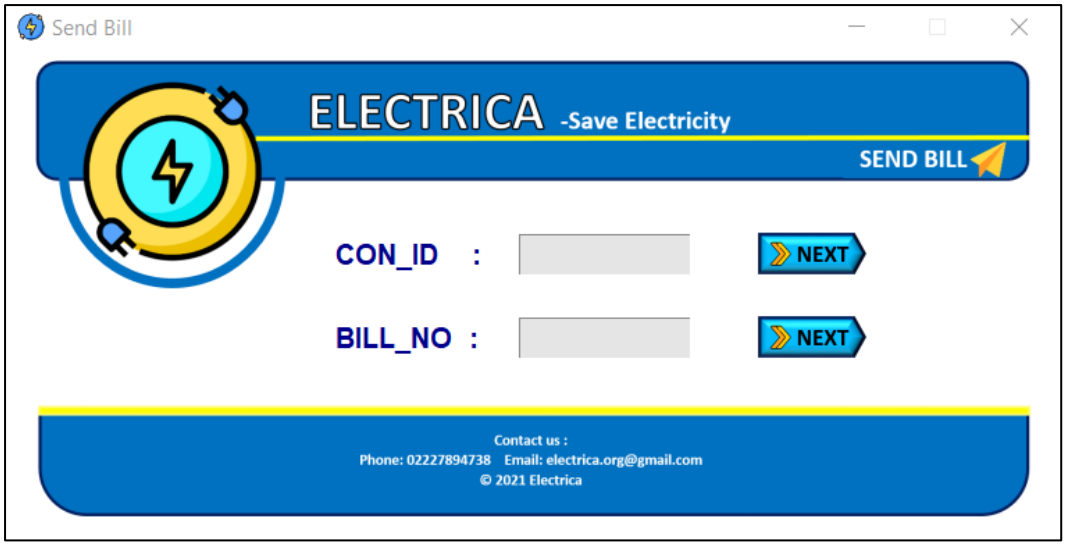


Fig 6.0

If the consumer id or bill number or fields are kept empty then error message will pop up. (As shown in Fig 2.2 & 2.4)

After writing the consumer id and clicking next option to write the important message will popup, This message will be displayed in the bill as well.

Fig 6.1

We can edit the message according to the requirement.

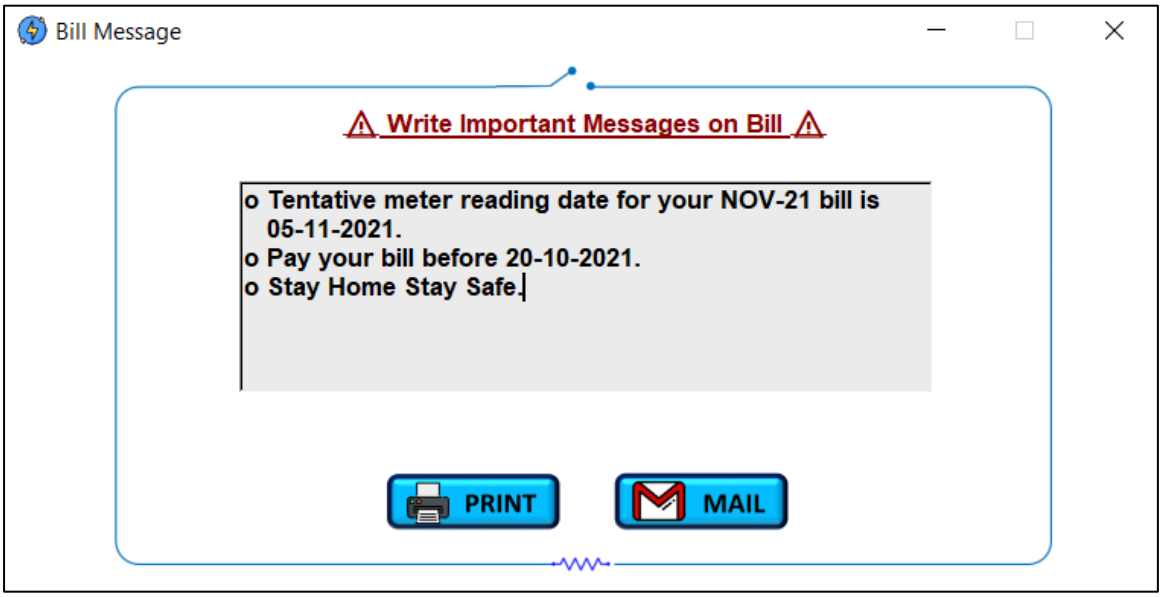


Fig 6.2

If print button is clicked then the consumers bill will be generated and printed.

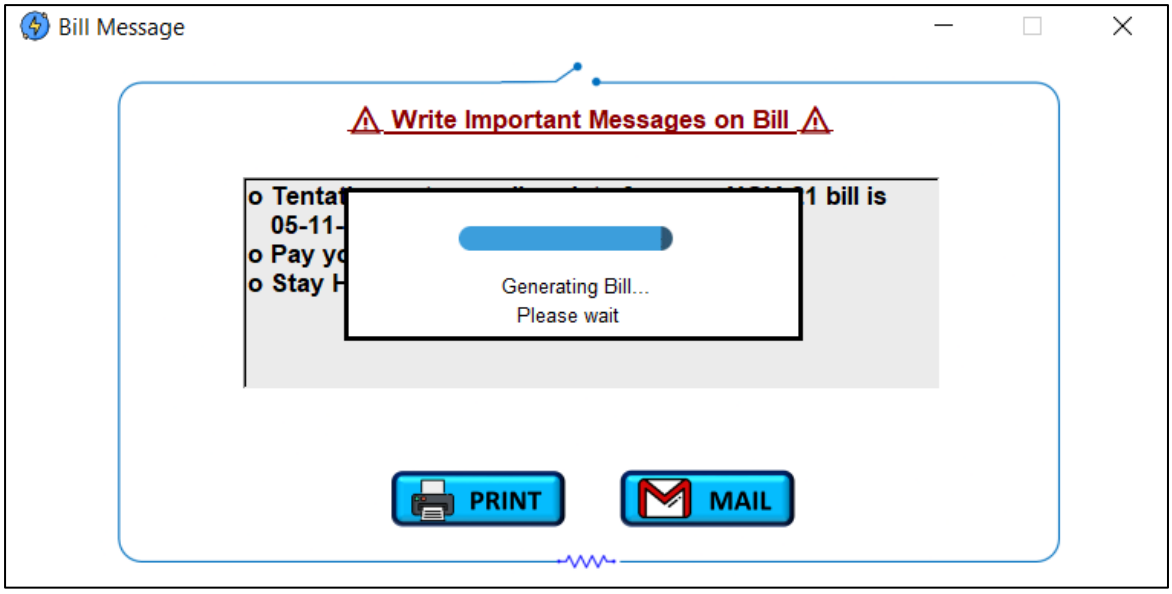


Fig 6.3

Then admin can take printout at his end. (As shown in Fig 1.8)

Fig 6.4

Bill pdf Image

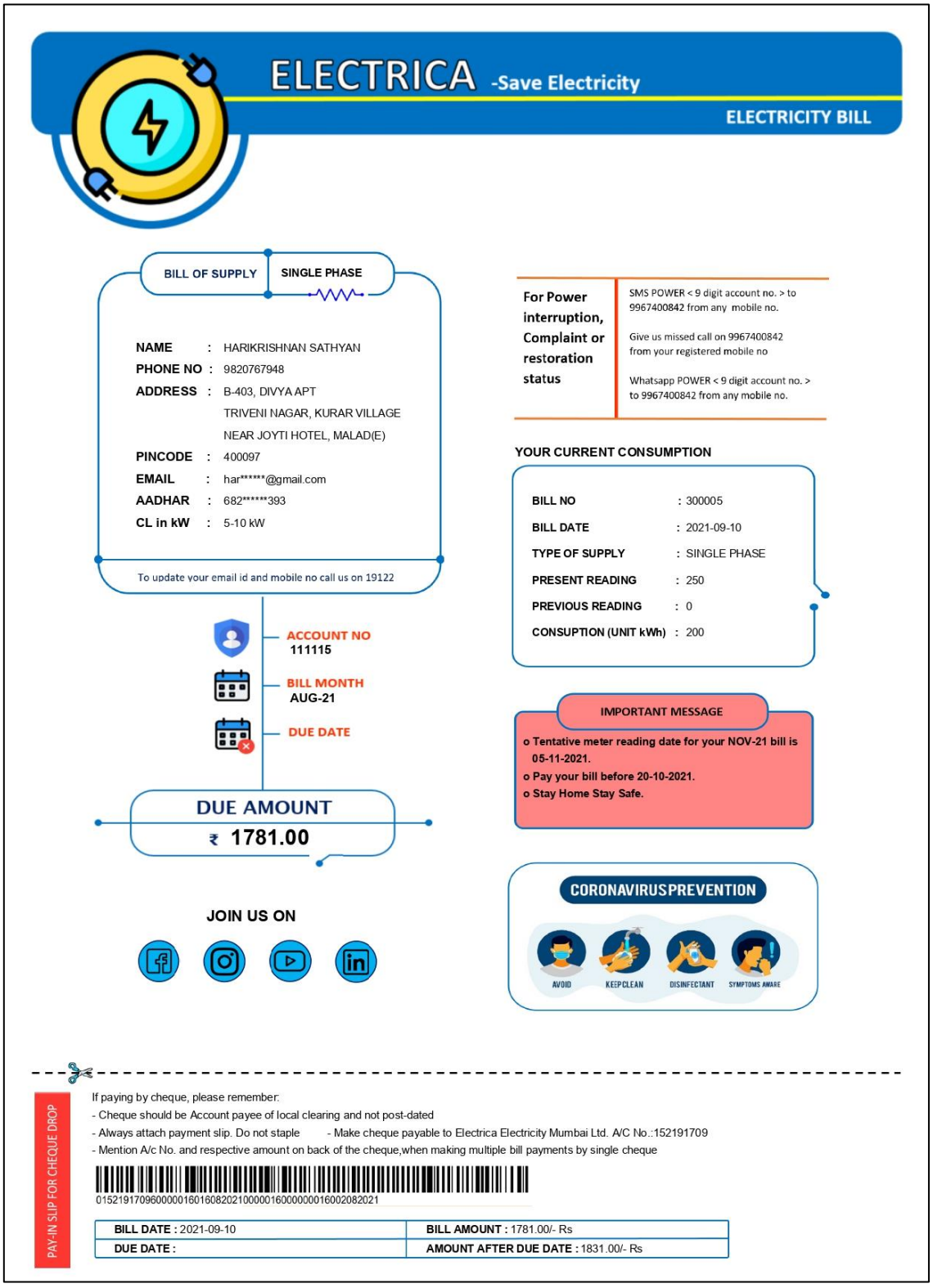
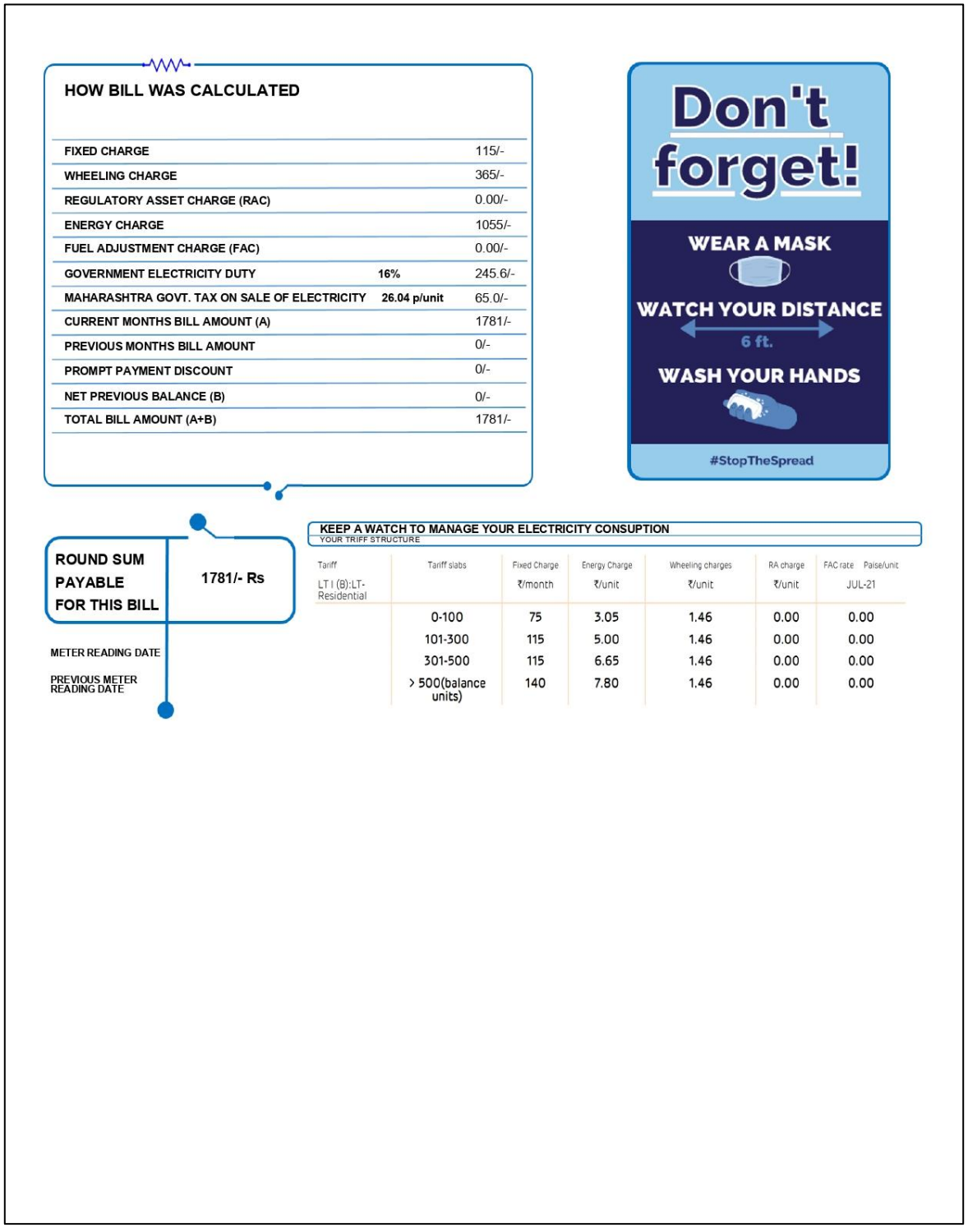


Fig 6.5
Pdf page 1



HOW BILL WAS CALCULATED

FIXED CHARGE		115/-
WHEELING CHARGE		365/-
REGULATORY ASSET CHARGE (RAC)		0.00/-
ENERGY CHARGE		1055/-
FUEL ADJUSTMENT CHARGE (FAC)		0.00/-
GOVERNMENT ELECTRICITY DUTY	16%	245.6/-
MAHARASHTRA GOVT. TAX ON SALE OF ELECTRICITY	26.04 p/unit	65.0/-
CURRENT MONTHS BILL AMOUNT (A)		1781/-
PREVIOUS MONTHS BILL AMOUNT		0/-
PROMPT PAYMENT DISCOUNT		0/-
NET PREVIOUS BALANCE (B)		0/-
TOTAL BILL AMOUNT (A+B)		1781/-

ROUND SUM PAYABLE FOR THIS BILL 1781/- Rs

METER READING DATE

PREVIOUS METER READING DATE

KEEP A WATCH TO MANAGE YOUR ELECTRICITY CONSUPTION

YOUR TRIFF STRUCTURE

Tariff	Tariff slabs	Fixed Charge ₹/month	Energy Charge ₹/unit	Wheeling charges ₹/unit	RA charge ₹/unit	FAC rate Paise/unit JUL-21
LT I (B):LT-Residential	0-100	75	3.05	1.46	0.00	0.00
	101-300	115	5.00	1.46	0.00	0.00
	301-500	115	6.65	1.46	0.00	0.00
	> 500(balance units)	140	7.80	1.46	0.00	0.00

Fig 6.6
Pdf page 2

Clicking mail button will send the bill pdf to the consumers email id

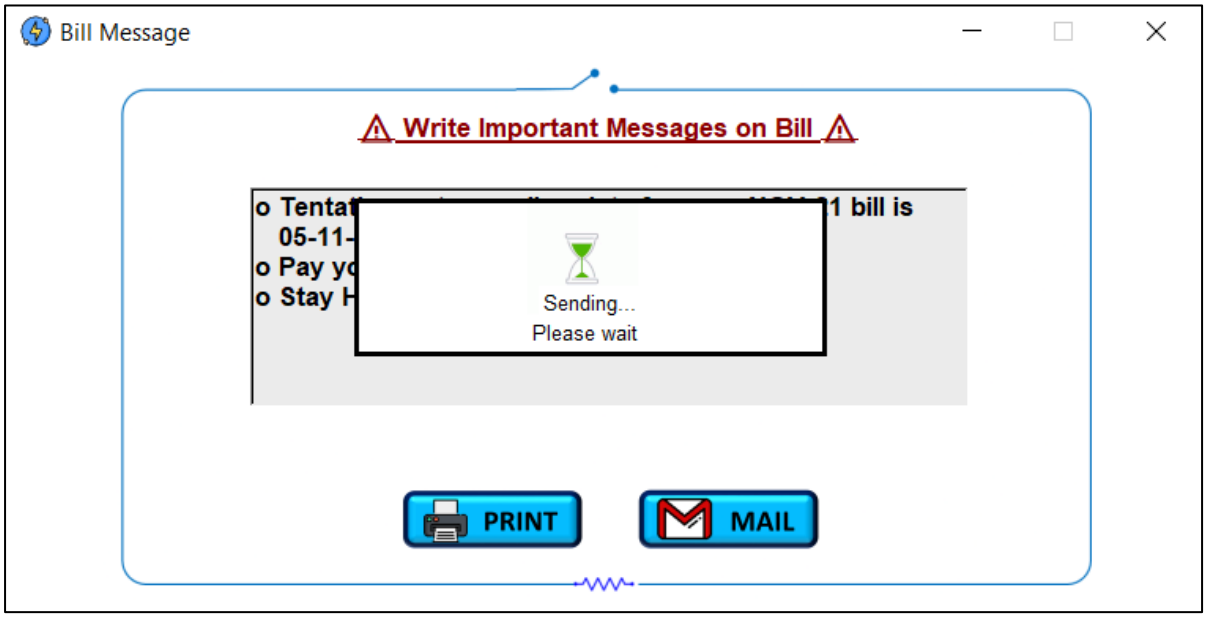

Bill Message
Write Important Messages on Bill
Sending...
Please wait
PRINT
MAIL

Fig 6.7

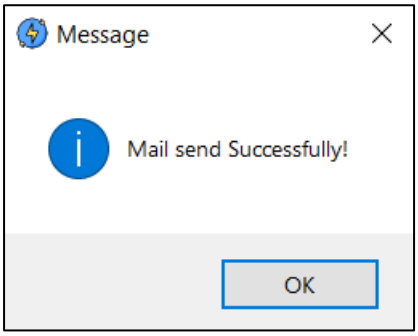

Message
Mail send Successfully!
OK

Fig 6.8

Bill received by the consumer through email

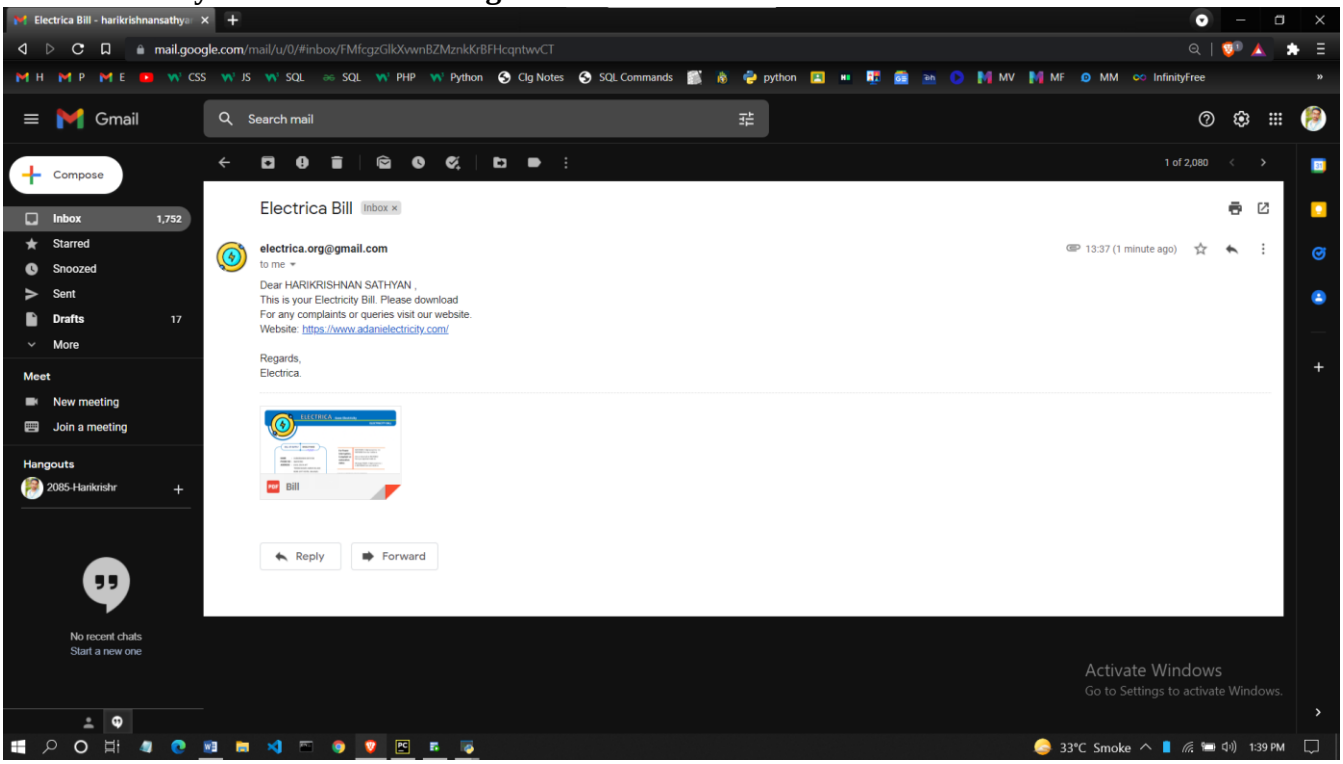


Fig 6.9

Admin can print the bill by the writing the bill number and clicking next, this will generate the bill (As shown in Fig (6.4))

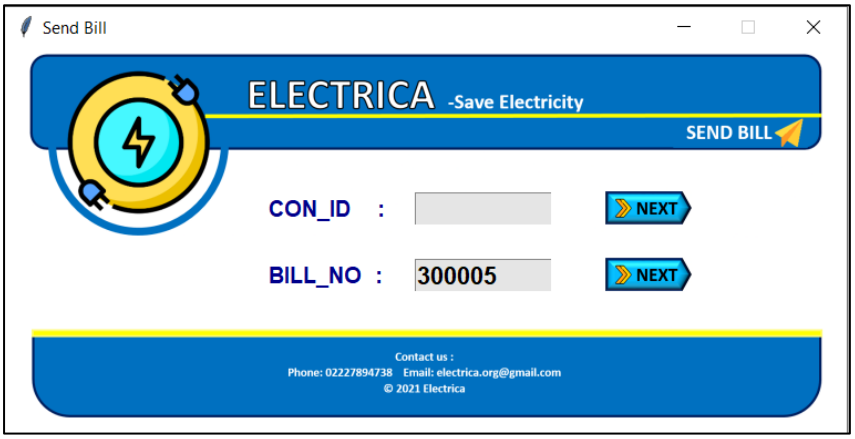


Fig 6.10

To pay the bill click on Payment button (Shown in Fig 1.0) , Payment interface will popup.

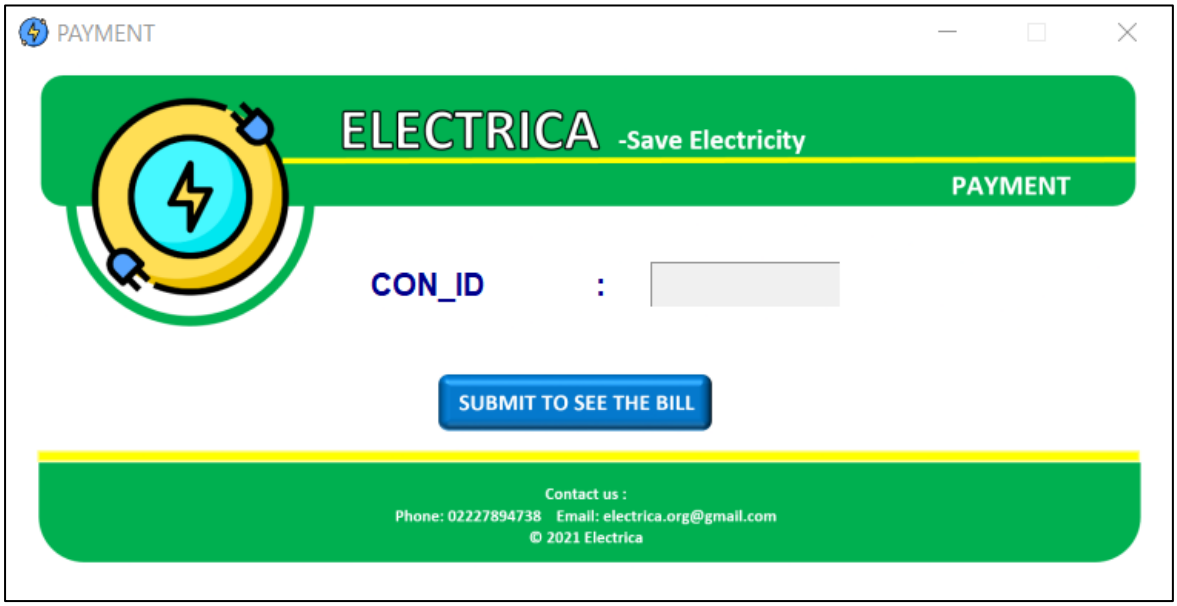


Fig 7.0

If invalid consumer id is entered or entry field is kept empty then error message will popup. (As shown in Fig 5.1 & 5.2)

After clicking submit consumers billing details will be displayed.

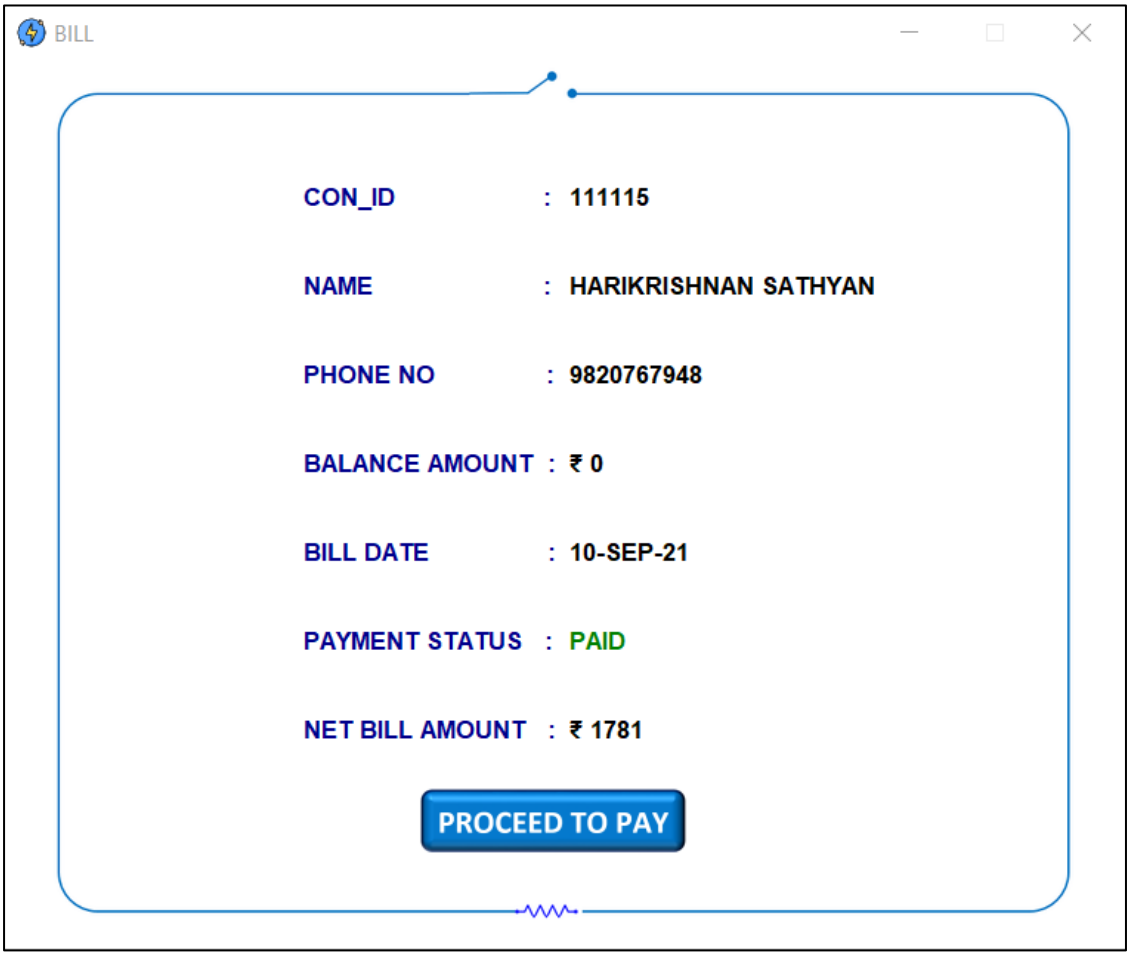


Fig 7.1

If the bill is payed then the payement status will be PAID in green colour or else the status bill be UNPAID in red colour (As shown in Fig 7.1 and 7.2)

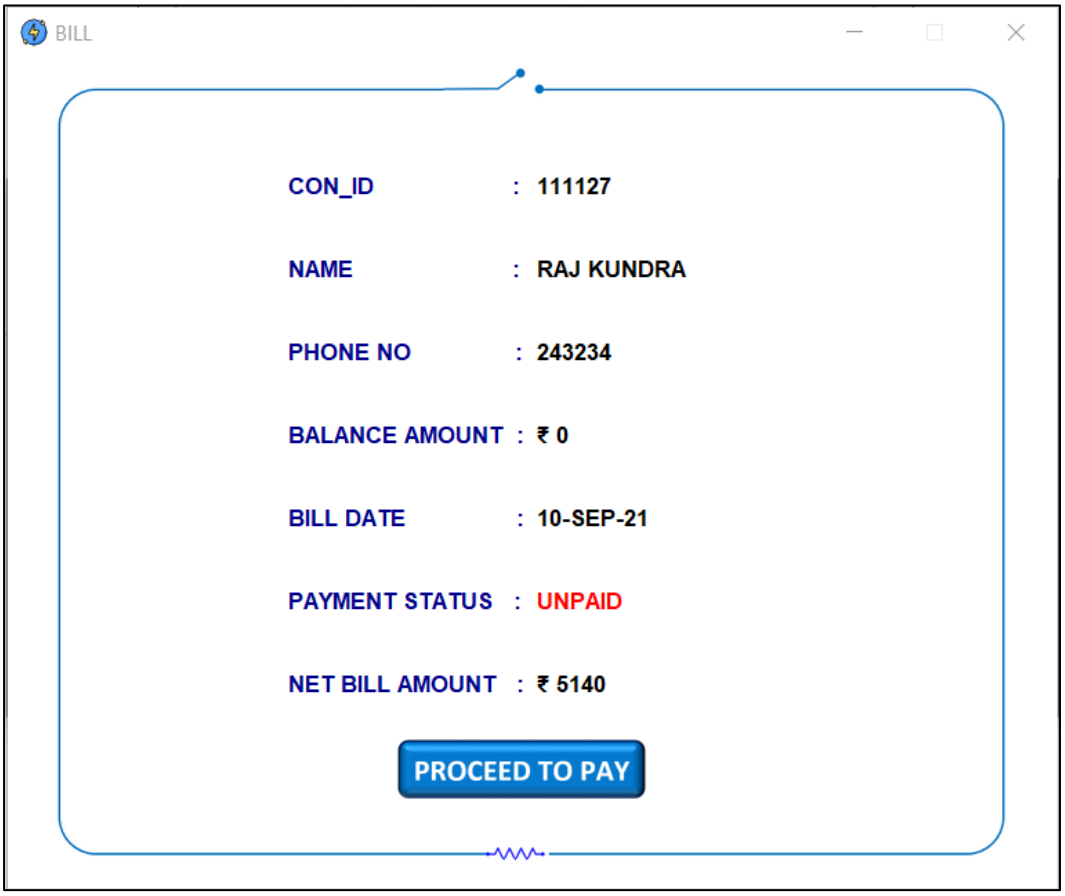


Fig 7.2

If the bill is paid then after clicking proceed to pay warning message will popup.

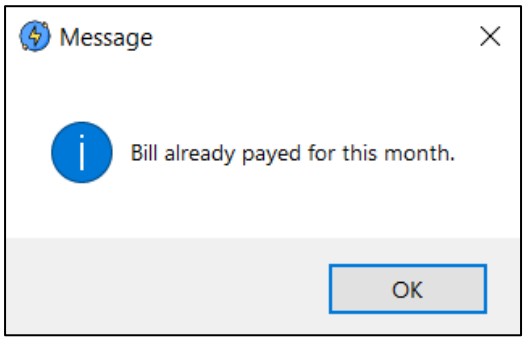


Fig 7.3

If no payed and clicked proceed to pay button then it will ask for the amount received by the consumer.

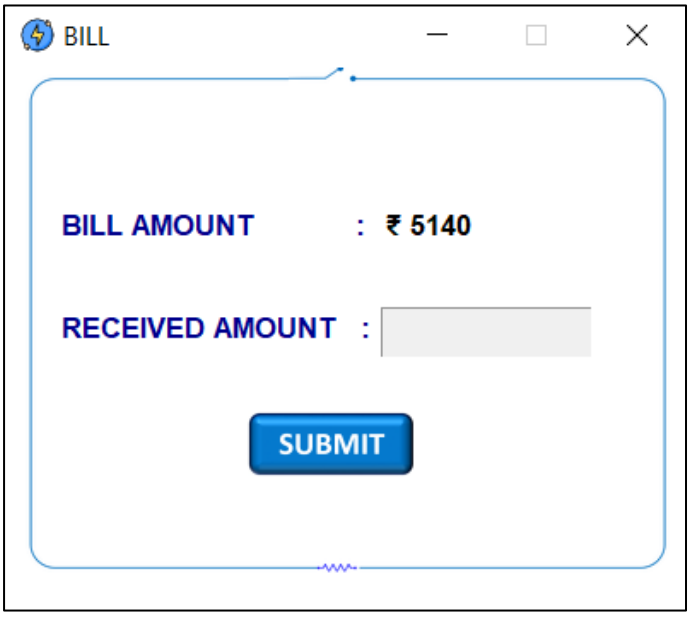


Fig 7.4

After entering the amount received by the consumer it will ask for security to make payment

Fig 7.5

After writing the correct password payment processing splash will popup

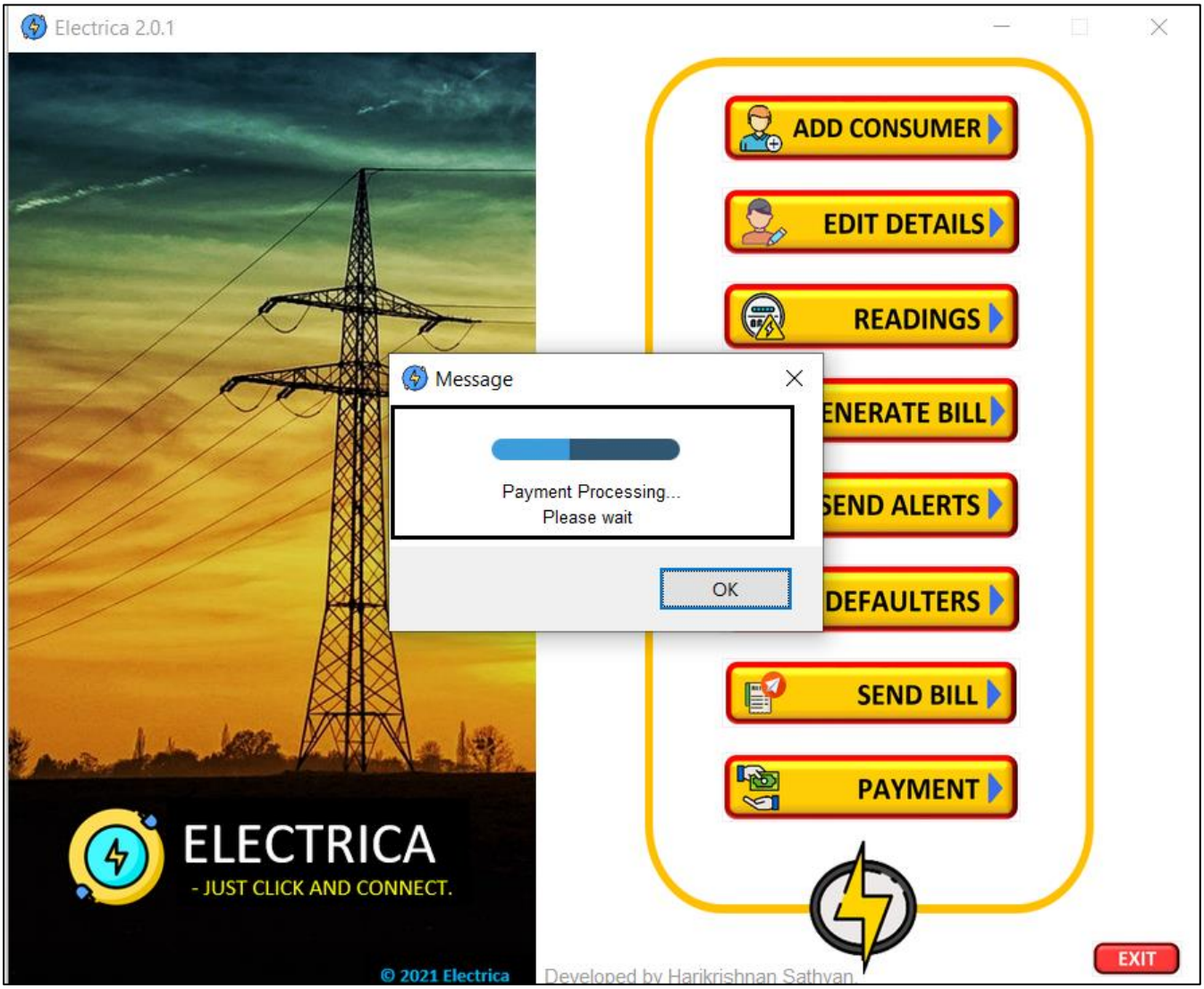


Fig 7.6

After successful payment bill payed message box bill popup.

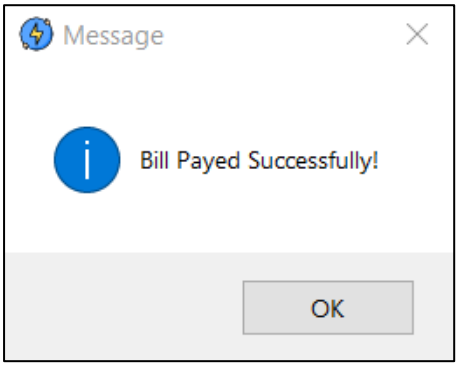


Fig 7.7

Clicking the exit button window will ask confirmation to exit or not.

If yes then the software will be closed.

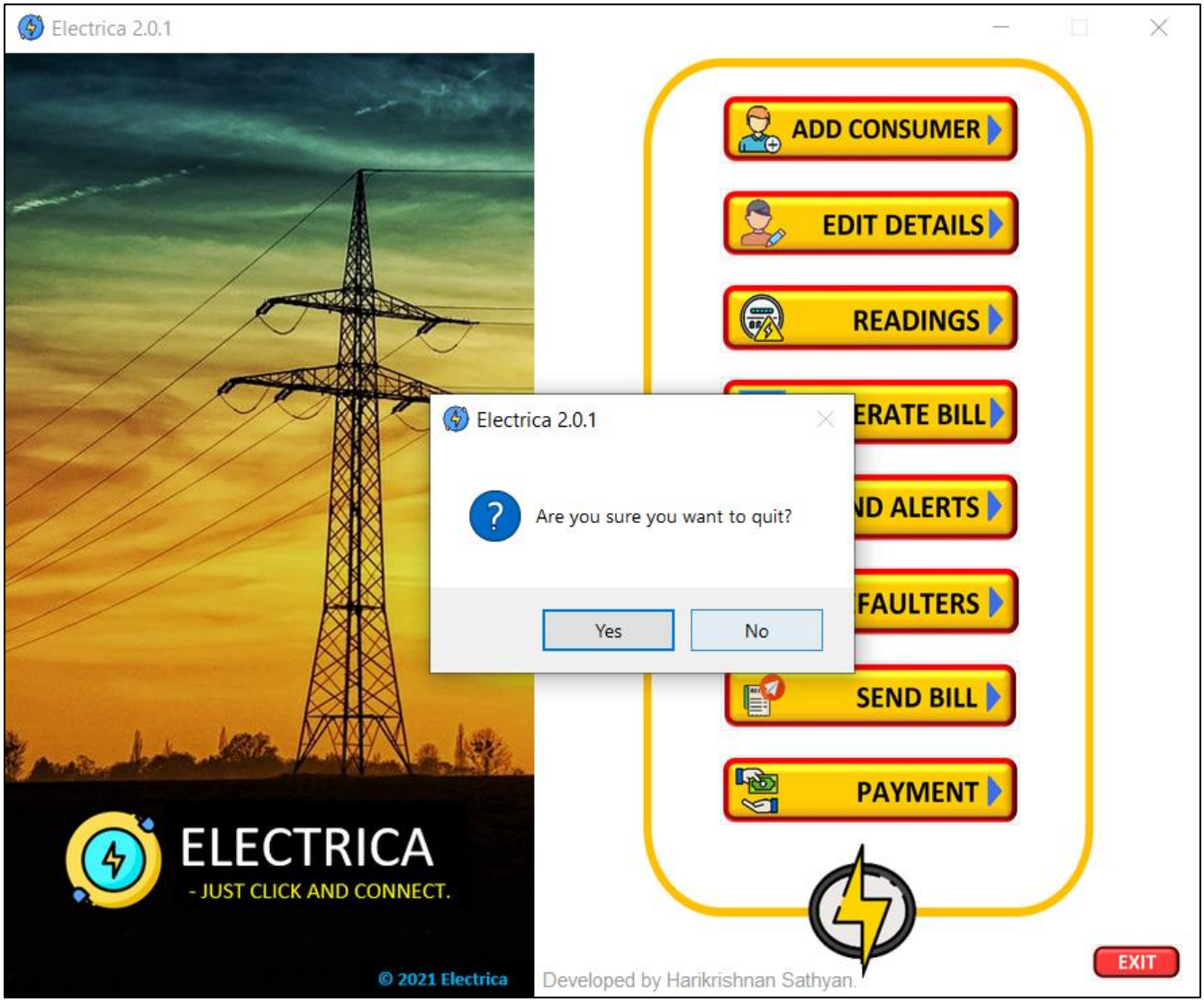
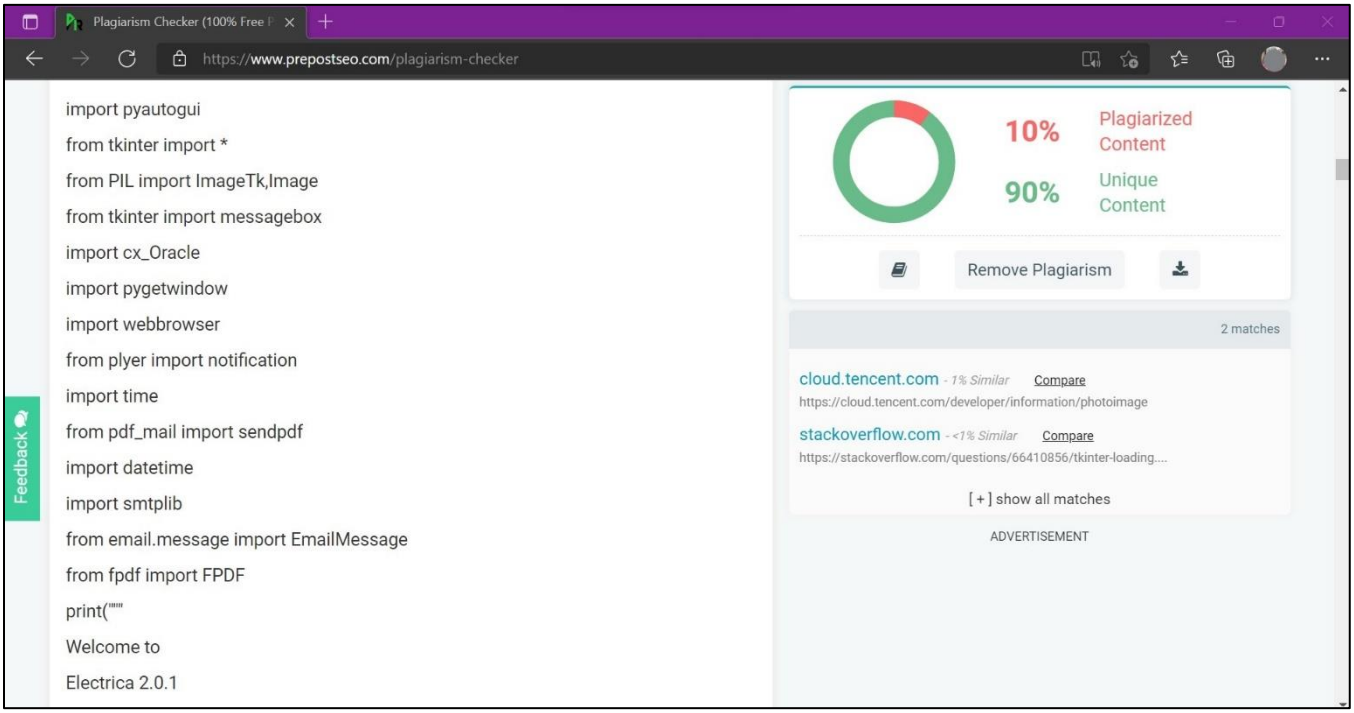


Fig 7.8

Plagiarism Report



CHAPTER 6 - CONCLUSION & BIBLIOGRAPHY

6.1 Conculsion and Future Enhancement

Electrica is a software tool which is capable of doing all the functionality for maintaining the Electricity bill using the latest technologies.

This project works perfectly with 99.9% of accuracy. But still there are various things to be added in the project which can be considered as future enhancement.

I have got successful results and all the fuctionality which where mentioned are implemented and are working without any errors.

- More functionalities can be added in future by applying the machine learning algorithms on the comsumers bill like predicting the consumers bill, and generating the bill accordingly.
- Consumers involvement can be added in the upcoming verisions.
- Sending bills and messages to multiple consumers at is also an important feature which can be implemented in future
- Consumers image in their profile details can be a new feature.
- Many process which are done in the software can be done with automation in the upcoming versions of Electrica.

6.2 Bibliography

- https://www.codeproject.com/Questions/1107878/Photoimage-not-displying-in-new-window-using-Pytho
- https://stackoverflow.com/questions
- https://www.adanielectricity.com/
- https://www.adanielectricity.com/Tariff
- https://www.prepostseo.com/plagiarism-checker
- https://www.youtube.com/c/Codemycom