**THAKUR COLLEGE OF SCIENCE AND COMMERCE**

KANDIVALI (EAST), MUMBAI

**A PROJECT REPORT ON**

**PATHOLAB   
(PATHOLOGICAL LAB MANAGEMENT SOFTWARE)**

**For**

**Thakur College of Science and Commerce**

**By**

**Mr. HARIKRISHNAN SATHYAN KONGIPARAMBIL**

**Roll no : 377**

**Submitted in Partial Fulfillment of Bachelors of Science (Computer Science)**

**[UNIVERSITY OF MUMBAI]**

**Thakur College of Science and Commerce**

**Kandivali (East), Mumbai**

**ACADEMIC YEAR 2021 - 2022**



**DECLARATION BY LEARNER**

I **MR. HARIKRISHNAN SATHYAN KONGIPARAMBIL** hereby, declare that the work embodied in this project work titled “**PATHOLAB**” forms my own contribution to the field of my interest under the guidance of **Asst** **Prof. MEGHNA SINGH** is a result of my own research work and has not been previously submitted to any other University for any other Degree/Diploma to this or any other University.

Wherever reference has been made to previous works of others, it has been clearly indicated as such and included in the bibliography.

I, hereby further declare that all information of this document has been obtained and presented in accordance with academic rules and ethical conduct.

Name and signature of the learner

HARIKRISHNAN SATHYAN KONGIPARAMBIL

Certified By

Name and Signature of Guiding Teacher

MS. MEGHNA SINGH

 **Certification from College**

**DATE:** 24/03/2022

**COMPUTER SCIENCE DEPARTMENT**

(**2021-2022) Certificate of Approval**

This is to certify that the project work entitled **“Patholab”** is prepared by **Mr. 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.

Ms. Meghna Singh Mr. Ashish Trivedi

**Project Guide 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 realize how 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’s been a long journey of failures, but consistency led me towards the path of success. After immense failures and hours of struggles the path towards success is found. It gives me immense pleasure to present this report towards the fulfilment of my project.

Though all my struggle has helped me achieved success, I would also like to thank and give the credit to the ones who had helped me and guided me for everything and helping me achieve the the first major project of my college academics.

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. This opportunity is more than a boon for me to display my skills and proficiency in the fields I’m interested in.

A special vote of thanks to **Ms. Meghna Singh**, who is our project guide 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 this project & to my family without whose support, motivation & encouragement this would not have been possible.

**-HARIKRISHNAN SATHYAN KONGIPARAMBIL**

**ABSTRACT**

Patholab is a patholgical laboratory management software which is developed in python and sqlite database. Local Pathological Laboratories are not keeping their daily test records and reports in a database and not sending reports to the patients via e-mail or through any other platforms, patient has visit the lab again to take the report.

Patholab solves this problem and make the report generation fully computerized. Storing data of the reports generated in the database and fetching it according to the requirement of the admin. Patholab helps the laboratory admin to get the daily report on the number of tests done in the laboratory by generating an excel sheet. Since the report generation is computerized the patient can get their report though email without visiting the laboratory . Patholab has a user friendly interface which makes the process hassle free  
searching for the availability of vaccine slots can be easily done using Patholab software.

The number of tests performed in the laboratories has increased which is leading to increase in data and this data has to be managed properly to keep record and for analyzing it later. During this pandemic the number of RTPCR test performed in the laboratory has reached to peak. The test reports are not maintained properly and also the patients don’t get the reports online or the through any of the digital platform. Considering this current situation it becomes very important for laboratories to manage the report with zero errors. Patholab solves all this problem by its take RTPCR test functionality and by generating V-CARD vaccination drives can be easily managed by saving all the records

**ORGANIZATIONAL REVIEW**

**Thakur College of Science and Commerce** (TCSC) is a college in Kandivali in Mumbai of Maharashtra, India running by **Thakur Educational Trust.**

Thakur College was started in **1992** to serve the needs of students passing

SSC examination from the schools around Kandivali area and Thakur Vidhya Mandir which has already established itself as one of the schools in the area. It offers courses at primarily the higher secondary and under-graduate levels. The courses at the undergraduate and post-graduate level are offered in affiliation with Mumbai University, Mumbai. An ISO 9001:2008 College with A grade as assessed by the National Assessment and Accreditation Council NAAC.

|  |  |
| --- | --- |
|  | Thakur College of Science and Commerce |
| **Founded:** | 1997 |
| **Address:** | Thakur College of Science and Commerce, Thakur Village Kandivali(E), Mumbai – 400 001 |
| **Motto:** | Journey towards Excellence |
| **Total Staff:** | 200 |
| **Number of Students:** | 12500 |
| **Email:** | Helpdesk@tcsc.org.in |

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR. NO** | **INDEX TOPIC** | **PAGE** |
| **1** | **INTRODUCTION** 1.1 INRODUCTION TO PROJECT 1.2 STATEMENT OF THE PROBLEM | **8** |
| **2** | **Objective, Scope and Methodology** 2.1Objective and Scope of the Project 2.2 Methodology of the Project | **10** |
| **3** | **System Analysis** 3.1 Technical Feasibility Study 3.2 Economic Feasibility Study | **11** |
| **4** | **Choice of platform for Hardware and Software** 4.1 Frontend 4.2 Backend 4.3 System Requirements | **12** |
| **5** | **Diagrams** 5.1 Architecture Diagram 5.2 Model flow Diagram 5.3 ER Diagram | **13** |
| **6** | **Implementation** 6.1 Sample Code 6.2 System Implementation and Output | **16** |
| **7** | **Conclusion** 7.1 Conclusion 7.2 Limitations 7.3 Future Enhancements 7.4 Bibliography | **44** |
| **8** | **Grantt chart** | **45** |
| **9** | **Miscellaneous** 8.1 Plagiarism Report | **46** |

**INTRODUCTION**

**1.1 Introduction to Project**

## [Digital transformation](https://perfectial.com/services/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.

## Digitalization refers to enabling or improving processes by leveraging digital technologies and digitized data. Therefore, digitalization presumes digitization. Examples of this could be as simple as PLC logic or PID control in a microprocessor-based system, sequenced logic for a batch process, automated shutdown logic, etc. It could also be something more complex, like an error in a transmitter generating a work order in the ERP maintenance system for a maintenance tech.

## Local Pathological Laboratories are not keeping their daily test records and reports in a database and not sending reports to the patients via e-mail or through any other platforms, patient has to go to lab again to take the report.

## In this current scenario of covid-19 RTPCR test is done in the lab on large scale, if a patient comes to be positive then proper report should be send to the patient and informed as soon as possible and also keep this data in the database to keep track of the test positivity rates and for many other report analysis.

## Digitalization increases productivity and efficiency while reducing costs. Digitalization improves an existing business process or processes but doesn’t change or transform them. That is to say, it takes a process from a human-driven event or series of events to software-driven.

* **What contribution would the project make?**
* Patholab will help user to manage report generation and processing the data of patients.
* Reduce the workload on the people who are working in the pathological laboratory.
* Easily find the vaccination slot available in the locality.
* Reduce spread of corona virus by sending immediate report to the patient so that they can isolate themselves.
* Keeping track of people who are vaccinated.

**1.2 STATEMENT OF THE PROBLEM**

* **Why is the particular topic chosen?**
* The number of tests performed in the laboratories has increased which is leading to increase in data and this data has to be managed properly to keep record and for analyzing it later
* Local Pathological Laboratories are not keeping their daily test records and reports in a database and not sending reports to the patients via e-mail or through any other platforms, patient has to go to lab again to take the report.
* During this pandemic the number of RTPCR test performed in the laboratory has reached to peak. The test reports are not maintained properly and also the patients don’t get the reports online or the through any of the digital platform
* Considering this current situation it becomes very important for laboratories to manage the report with zero errors.

**OBJECTIVE, SCOPE AND METHODOLOGY**

**2.1 Objective and Scope of the Project**

* + The main objective of this project is to make the report generation computerized.
  + Storing data of reports generated in the database and fetching it according to the requirement.
  + Get daily report on number of tests performed in the lab to the Admin.
  + Sending reports online to the patients.
  + Keeping track of Covid-19 test positivity rate.
  + Easy management of Covid-19 vaccination drives.
  + Reduce the errors occurred while report generation.
  + User friendly interface.

**2.2 Methodology of the Project**

* Project will have a home window where all the operation will be present such as create receipt, Take test, Edit report, Delete report, Covid-19 test, Send report, Vaccine booking, V-card.
* Creating receipt for the test to be performed, storing that data in the database and sending copy of receipt through e-mail to the patient as an acknowledgement.
* Admin or the lab assistant has to enter the test readings of the patient after performing the test into the take test field which will be stored in the database for report generation.
* Admin can also edit the entries in case of any mistake.
* Special section for Covid-19 tests.
* Generating the final report and options to print report and also send it to the patients email id, in case of any urgency like if Covid-19 test is positive then informing the patient through Whatsapp message.
* Covid-19 vaccination slot booking.
* Generating V-card for the people who are getting vaccinated from the lab   
  which will have the vaccination details.

**System Analysis**

Technical Feasibility assesses the current resources and technology, which are required to accomplish the goals within the allocated time and for this. The team ascertains whether the current resources and technology can be upgraded or added to accomplish specified requirements.

**Technical feasibility of the product has been studied under following heads:**

1. **Hardware availability** - This project requires any desktop or laptop with internet connection. (Here we will be using desktop for testing purpose)

2. **Platform Independence** - This project will be platform dependent since it’s an web application and can be accessed on Windows, Linux or MacOS operating system through any web browser which is available.

3. **User friendly** - The application interface will be built using Python therefore it has rich user-friendly interface.

As hardware availability and user-friendly nature of this project is possible to achieve, this system is technically feasible

**3.2 Economic Feasibility**

Study Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. It consists of market analysis, economic analysis, technical and strategic analysis.

1. **Hardware cost** – A computer and a good internet connection is required for both development and usage. Yet the hardware cost will manageable at company level.

2**. Software cost** - Software needed for development of this application are open-source software i.e., Python. Hence. there is no cost associated with it.

**Choice of platform for Hardware and Software**

**4.1 Frontend**

Python is a programming language known for its far-reaching applicability that goes beyond web development coding.

Python can be a good coding language for new or novice coders because of its readability and use of the English language.

Although it's an accessible program to learn and use, Python can be scaled up and implemented for vast, complex tasks, including compiling massive amounts of data and executing machine learning algorithms.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

**Softwares used**

* Python 3.9.6
* Sqlite3
* Pycharm
* Browser for Sqlite3

**4.2 Backend**

SQLite is used to develop embedded software for devices like televisions, cell phones, cameras, etc. It can manage low to medium-traffic HTTP requests. SQLite can change files into smaller size archives with lesser metadata. SQLite is used as a temporary dataset to get processed with some data within an application.

**Minimum Hardware Requirements**:

• Processor: Intel Pentium IV, 1GHZ or above

• RAM: 1024MB or above

• Video: 1024×768, 24-bit colours

• Keyboard: Standard 104 Keys

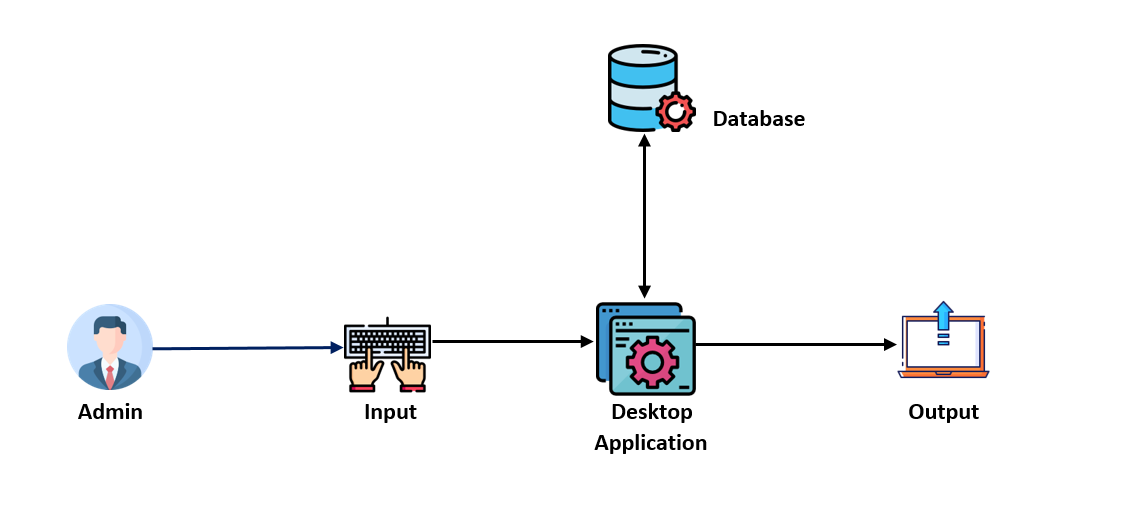
• Internet Speed: 1 Mbps or above

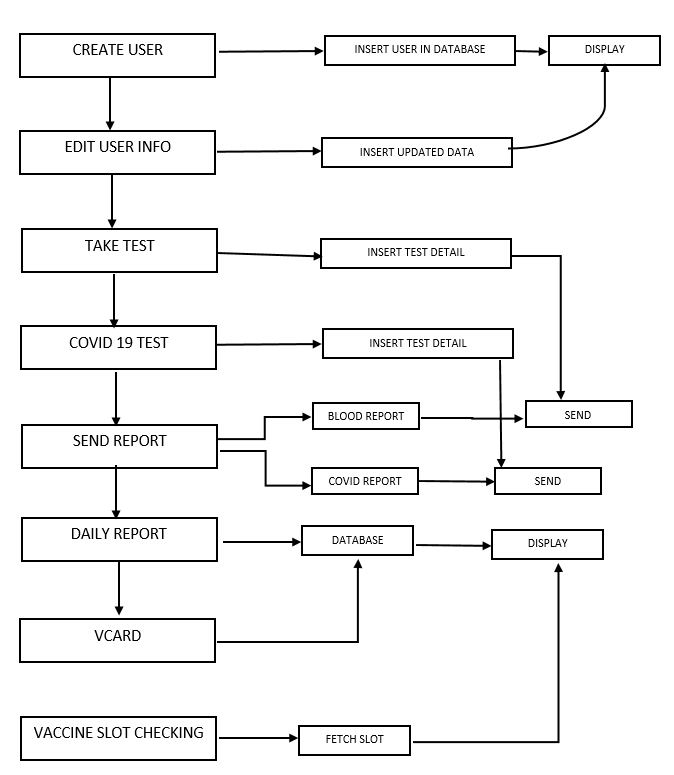
**Minimum Software Requirement**

* No specific software requirements.

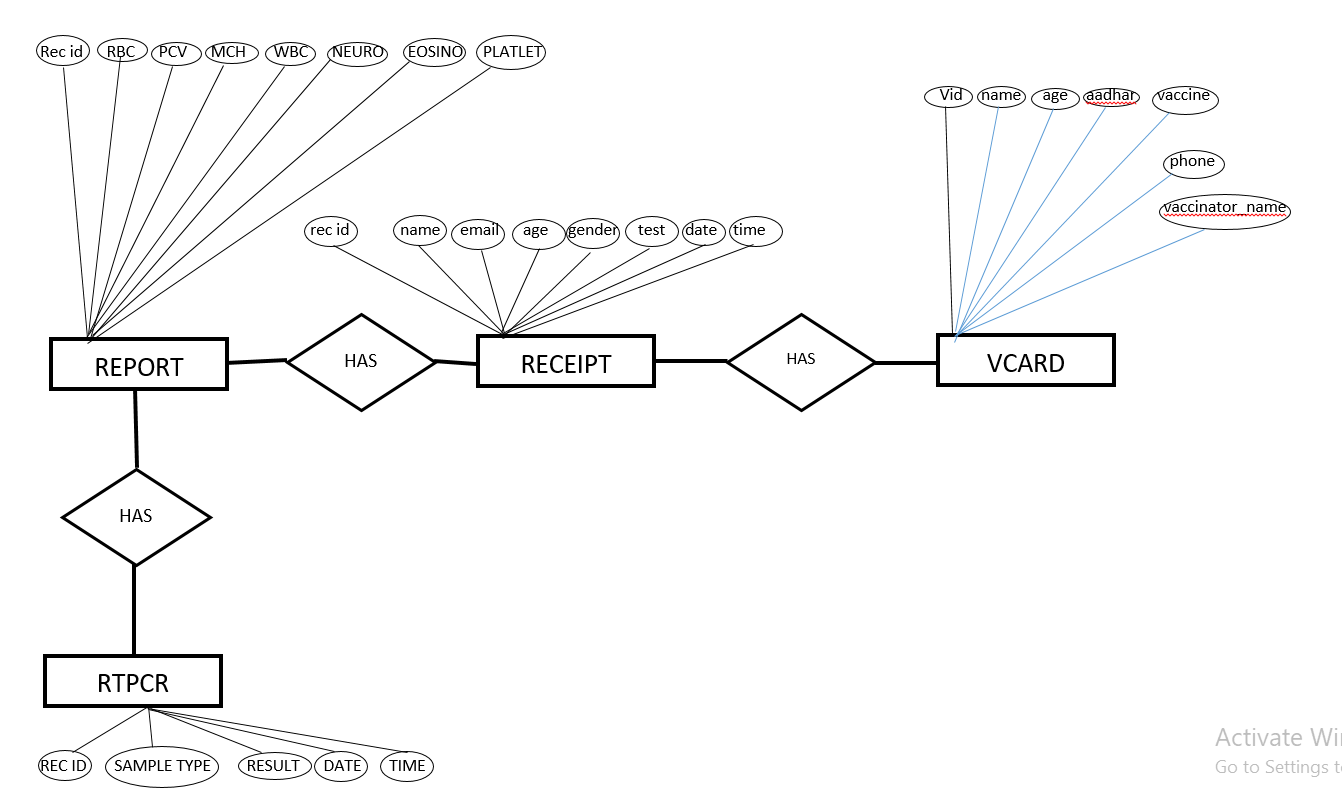
**Diagrams**

**5.1 Architecture Diagram**

****

**5.2 Model flow Diagram**

**5.3 ER DIAGRAM**

****

**Implementation**

**6.1 Sample Code**

**Function for home window**

def homewindow():  
 global home  
 home = Tk()  
 home.configure(bg="white")  
 home.title('PATHOLAB 1.1.0')  
 home.resizable(False**,**False)  
 home.iconbitmap("Images/icon4.ico")  
 window\_width**,** window\_height = **885, 650** 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\_template2.png")  
 temp\_resized = temp\_size.resize((**395,655**)**,** Image.ANTIALIAS)  
 template = ImageTk.PhotoImage(temp\_resized)  
 template\_image = Label(home**,**image=template**,**borderwidth="0")  
 template\_image.place(x="-1"**,**y="-3")  
  
 backtemplate\_size = Image.open("Images/back1.jpg")  
 backtemplate\_resized = backtemplate\_size.resize((**410,700**)**,** Image.ANTIALIAS)  
 backtemplate = ImageTk.PhotoImage(backtemplate\_resized)  
 backtemplate\_image = Label(home**,**image=backtemplate**,**borderwidth="0")  
 backtemplate\_image.place(x="430"**,**y="0")  
  
 developer = Label(home**,** text="Developed By HARKRISHNAN SATHYAN"**,** font="lucida 6 "**,** bg="navy"**,** fg="white")  
 developer.place(x="0"**,** y="635")  
  
 receipt\_size = Image.open("Images/receipt\_button.png")  
 receipt\_resized = receipt\_size.resize((**220,50**)**,** Image.ANTIALIAS)  
 receipt\_image = ImageTk.PhotoImage(receipt\_resized)  
 Label(image=receipt\_image)  
 button\_receipt = Button(home**,**image=receipt\_image**,**borderwidth="0"**,**activebackground='blue'**,**command=receiptEntry)  
 button\_receipt.place(x=**530,**y=**30**)  
  
 editreceipt\_size = Image.open("Images/editreceipt\_button.png")  
 editreceipt\_resized = editreceipt\_size.resize((**220,50**)**,** Image.ANTIALIAS)  
 editreceipt\_image = ImageTk.PhotoImage(editreceipt\_resized)  
 Label(image=editreceipt\_image)  
 button\_editreceipt = Button(home**,**image=editreceipt\_image**,**borderwidth="0"**,**activebackground='blue'**,**command=editReceipt)  
 button\_editreceipt.place(x=**530,**y=**100**)  
  
 taketest\_size = Image.open("Images/taketest\_button.png")  
 taketest\_resized = taketest\_size.resize((**220,50**)**,** Image.ANTIALIAS)  
 taketest\_image = ImageTk.PhotoImage(taketest\_resized)  
 Label(image=taketest\_image)  
 button\_taketest = Button(home**,**image=taketest\_image**,**borderwidth="0"**,**command=takeTest)  
 button\_taketest.place(x=**530,**y=**170**)  
  
  
  
 covtest\_size = Image.open("Images/covidtest\_button.png")  
 covtest\_resized = covtest\_size.resize((**220,50**)**,** Image.ANTIALIAS)  
 covtest\_image = ImageTk.PhotoImage(covtest\_resized)  
 Label(image=covtest\_image)  
 button\_covtest = Button(home**,**image=covtest\_image**,**borderwidth="0"**,**command=takecovidtest)  
 button\_covtest.place(x=**530,**y=**240**)  
  
 sendreport\_size = Image.open("Images/sendreport\_button.png")  
 sendreport\_resized = sendreport\_size.resize((**220,50**)**,** Image.ANTIALIAS)  
 sendreport\_image = ImageTk.PhotoImage(sendreport\_resized)  
 Label(image=sendreport\_image)  
 button\_sendreport = Button(home**,**image=sendreport\_image**,**borderwidth="0"**,**command=sendreport)  
 button\_sendreport.place(x=**530,**y=**310**)  
  
 deletereport\_size = Image.open("Images/deletereport\_button.png")  
 deletereport\_resized = deletereport\_size.resize((**220,50**)**,** Image.ANTIALIAS)  
 deletereport\_image = ImageTk.PhotoImage(deletereport\_resized)  
 Label(image=deletereport\_image)  
 button\_deletereport = Button(home**,**image=deletereport\_image**,**borderwidth="0"**,**command=dailyreport)  
 button\_deletereport.place(x=**530,**y=**380**)  
  
 vcard\_size = Image.open("Images/vcard\_button.png")  
 vcard\_resized = vcard\_size.resize((**220,50**)**,** Image.ANTIALIAS)  
 vcard\_image = ImageTk.PhotoImage(vcard\_resized)  
 Label(image=vcard\_image)  
 button\_vcard = Button(home**,**image=vcard\_image**,**borderwidth="0"**,**command=vcard\_entry)  
 button\_vcard.place(x=**530,**y=**450**)  
  
 vaccineslot\_size = Image.open("Images/vaccineslot\_button.png")  
 vaccineslot\_resized = vaccineslot\_size.resize((**220,50**)**,** Image.ANTIALIAS)  
 vaccineslot\_image = ImageTk.PhotoImage(vaccineslot\_resized)  
 Label(image=vaccineslot\_image)  
 button\_vaccineslot = Button(home**,**image=vaccineslot\_image**,**borderwidth="0"**,**command=age\_pin)  
 button\_vaccineslot.place(x=**530,**y=**520**)  
  
 exit\_size = Image.open("Images/Exit\_buttons.png")  
 exit\_resized = exit\_size.resize((**60,25**)**,** Image.ANTIALIAS)  
 exit\_image = ImageTk.PhotoImage(exit\_resized)  
 Label(image=exit\_image)  
 button\_exit = Button(home**,**image=exit\_image**,**borderwidth="0"**,**activebackground='red'**,**command=exitapp)  
 button\_exit.place(x=**818,**y=**620**)  
  
 home.mainloop()

**Function for sending report**

def sendbloodreport():  
 try:  
 recid = bld\_recidentry.get()  
 conn = sqlite3.connect('Labdb.db')  
 cur = conn.cursor()  
 details = cur.execute(f"SELECT \* FROM RECEIPT WHERE REC\_ID = {recid}")  
 values = details.fetchall()  
 for i in values:  
 name = i[**1**]  
 test = i[**6**]  
 email = i[**5**]  
 # Create an object of sendpdf function  
 k = sendpdf("patholabsreport@gmail.com"**,** f"{email}"**,** "Patho@175"**,** "Patholab Report"**,** f"Dear {name} ,\nThis is your Report for {test} lab test. Please download the report. \nFor any complaints or queries visit our website. \nWebsite:www.patholab.com \n\nRegards,\nPatholab."**,** "Patient Report"**,** f"{os.getcwd()}")  
  
 # sending an email  
 k.email\_send()  
 notifyrecmail()  
  
 except Exception as e:  
 messagebox.showerror("Error ⚠"**,** f"Poor network connection, Please try again later.\n\n{e} ")

**Function for inserting patient data into the data base**

def receiptInsertdb():  
 name = str(recname\_entry.get())  
 age = recage\_entry.get()  
 gender = var.get()  
 phone = recphone\_entry.get()  
 doc = recrefdr\_entry.get()  
 test = str(click.get())  
 email = str(recemail\_entry.get())  
 timenow =(datetime.today().strftime("%I:%M %p"))  
  
 # testamt = 750  
 if (test == "Complete Blood Count"):  
 testamt=**700** elif(test=="RT-PCR"):  
 testamt=**500** elif(test == "Prothrobin Time"):  
 testamt = **750** elif(test=="Basic Metabolic Panel"):  
 testamt=**800** elif(test=="Comprehensive Metabolic Panel"):  
 testamt=**900** elif(test=="Lipid Panel"):  
 testamt=**500** elif(test=="Liver Panel"):  
 testamt=**950** elif(test=="Thyroid Stimulating Hormone"):  
 testamt=**750** elif(test=="Hemoglobin A1C"):  
 testamt=**1200** elif(test=="Urinalysis"):  
 testamt=**600** elif(test=="Cultures"):  
 testamt=**1100** else:  
 testamt=**500** try:  
 conn = sqlite3.connect('Labdb.db')  
 cur = conn.cursor()  
 cur.execute(f"INSERT INTO Receipt values((SELECT max (REC\_ID)+1 from Receipt),'{name}',{age},'{gender}',{phone},'{email}','{test}','{doc}',{testamt},STRFTIME('%d/%m/%Y'),'{timenow}')") # get method gets values from the variable  
 cur.close()  
 conn.commit()  
 conn.close()  
 showmessage()  
 except Exception as e:  
 showError()

**Function for generating VCARD :**

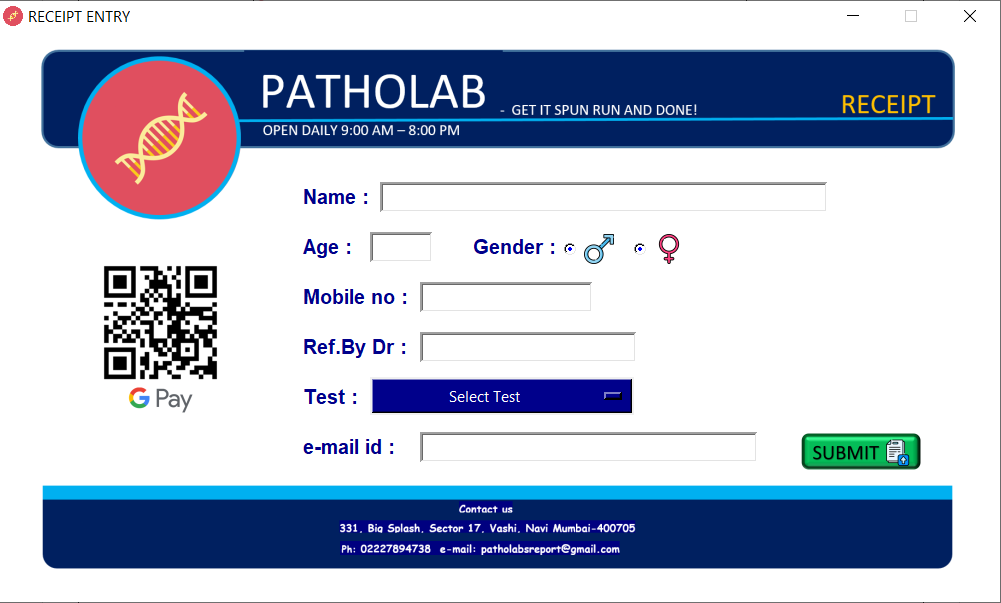
def vcard\_entry():  
 global ventry  
 ventry = Toplevel()  
 ventry.iconbitmap("Images/icon4.ico")  
 ventry.config(bg="white")  
 window\_width**,** window\_height = **700, 400** screen\_width = ventry.winfo\_screenwidth()  
 screen\_height = ventry.winfo\_screenheight()  
  
 position\_top = int(screen\_height / **2** - window\_height / **2**)  
 position\_right = int(screen\_width / **2** - window\_width / **2**)  
  
 ventry.geometry(f"{window\_width}x{window\_height}+{position\_right}+{position\_top}")  
  
 ventry.title("V-CARD")  
  
  
 global v\_name\_entry**,**v\_age\_entry**,**v\_aadhar\_entry**,**v\_phone\_entry**,**click**,**v\_vaccinator\_entry**,**image\_name  
 rentry\_top = Image.open("Images/vcard\_bg.png")  
 rephoto = ImageTk.PhotoImage(rentry\_top)  
 ventry.photo = rephoto # solution for bug in `PhotoImage`  
 receipt\_toplogo = Label(ventry**,** image=rephoto**,** borderwidth="0")  
 receipt\_toplogo.place(x="25"**,** y="10")  
  
  
 choose\_size = Image.open("Images/Choose\_file\_btn.png")  
 choose\_resized = choose\_size.resize((**100, 30**)**,** Image.ANTIALIAS)  
 choose\_image = ImageTk.PhotoImage(choose\_resized)  
 Label(image=choose\_image)  
 button\_choose = Button(ventry**,** image=choose\_image**,** borderwidth="0"**,** activebackground='blue'**,** command=browseImage)  
 button\_choose.place(x=**87,** y=**195**)  
  
 generate\_size = Image.open("Images/generateid.png")  
 generate\_resized = generate\_size.resize((**130, 40**)**,** Image.ANTIALIAS)  
 generate\_image = ImageTk.PhotoImage(generate\_resized)  
 Label(image=generate\_image)  
 button\_generate = Button(ventry**,** image=generate\_image**,** borderwidth="0"**,** activebackground='blue'**,** command=vcard\_db)  
 button\_generate.place(x=**280,** y=**310**)  
  
 v\_name = Label(ventry**,** text="NAME : "**,** font="lucida 9 bold"**,** bg= 'white'**,** fg="blue4")  
 v\_name.place(x="260"**,** y="100")  
 v\_name\_entry = Entry(ventry**,** width="35"**,** font="lucida 8 bold"**,** bd="3")  
 v\_name\_entry.place(x="380"**,** y="100")  
  
 v\_age = Label(ventry**,** text="AGE : "**,** font="lucida 9 bold"**,** bg='white'**,** fg="blue4")  
 v\_age.place(x="260"**,** y="130")  
 v\_age\_entry = Entry(ventry**,** width="7"**,** font="lucida 8 bold"**,** bd="3")  
 v\_age\_entry.place(x="380"**,** y="130")  
  
 v\_aadhar = Label(ventry**,** text="AADHAAR NO : 682981917161 "**,** font="lucida 9 bold"**,** bg='white'**,** fg="blue4")  
 v\_aadhar.place(x="260"**,** y="160")  
 v\_aadhar\_entry = Entry(ventry**,** width="15"**,** font="lucida 8 bold"**,** bd="3")  
 v\_aadhar\_entry.place(x="380"**,** y="160")  
  
 v\_vaccine = Label(ventry**,** text="VACCINE : COVISHIELD "**,** font="lucida 9 bold"**,** bg='white'**,** fg="blue4")  
 v\_vaccine.place(x="260"**,** y="190")  
  
 vaccine\_names = ['COVISHIELD'**,**'COVAXIN'**,**'PFIZER'**,**'SPUTNIK V']  
 click = StringVar()  
 click.set("Select Test")  
 test\_dropdown = OptionMenu(ventry**,** click**,** \*vaccine\_names)  
 test\_dropdown.config(bg="blue4"**,** fg="white"**,**height="0"**,** width="15"**,**font='lucida 5 bold'**,** activebackground="dodger blue"**,** activeforeground="black")  
 test\_dropdown.place(x="379"**,** y="187")  
  
 v\_phone = Label(ventry**,** text="PHONE NO :"**,** font="lucida 9 bold"**,** bg='white'**,** fg="blue4")  
 v\_phone.place(x="260"**,** y="220")  
 v\_phone\_entry = Entry(ventry**,** width="15"**,** font="lucida 8 bold"**,** bd="3")  
 v\_phone\_entry.place(x="380"**,** y="220")  
  
 v\_vaccinator = Label(ventry**,** text="VACCINATOR'S NAME :"**,** font="lucida 9 bold"**,** bg='white'**,** fg="blue4")  
 v\_vaccinator.place(x="260"**,** y="250")  
 v\_vaccinator\_entry = Entry(ventry**,** width="25"**,** font="lucida 8 bold"**,** bd="3")  
 v\_vaccinator\_entry.place(x="440"**,** y="250")  
  
 global label\_file\_explorer  
 label\_file\_explorer = Label(ventry**,** text=""**,**font="lucida 5 bold"**,** width=**23,** height=**3,** fg="black"**,**bg="white"**,**)  
 label\_file\_explorer.place(x="77"**,**y="230")  
  
 ventry.mainloop()  
  
  
def browseImage():  
 global filename**,**image\_name  
 filename = filedialog.askopenfilename(initialdir="/"**,** title="Select a File"**,** filetypes=(("Image files"**,** "\*.png\*")**,** ("all files"**,** "\*.\*")))  
  
 # Change label contentsx  
 imagelocation\_count = filename.rfind("/")  
 global image\_name**,**label\_file\_explorer  
 image\_name = filename[(imagelocation\_count + **1**):]  
 label\_file\_explorer.configure(text=image\_name)  
  
  
  
def vcard\_db():  
 name = v\_name\_entry.get()  
 age = v\_age\_entry.get()  
 aadhar = v\_aadhar\_entry.get()  
 phone = v\_phone\_entry.get()  
 vaccine = click.get()  
 vaccinator = v\_vaccinator\_entry.get()  
  
  
  
 try:  
  
 conn = sqlite3.connect('Labdb.db')  
 cur = conn.cursor()  
 cur.execute(f"INSERT INTO VCARD VALUES((SELECT max (VID)+1 from VCARD),'{name}',{age},{aadhar},'{vaccine}',{phone},'{vaccinator}','{image\_name}',STRFTIME('%d/%m/%Y'))") # get method gets values from the variable  
 cur.close()  
 conn.commit()  
 conn.close()  
 messagebox.showinfo("Message"**,** "Card created successfully!")  
 generateid()  
  
 except Exception as e:  
 messagebox.showerror("Error"**,** "Some Error Occured \n\n ⭕ Entry field should not be Empty.\n ⭕ Please select the profile image.")  
 print(e)

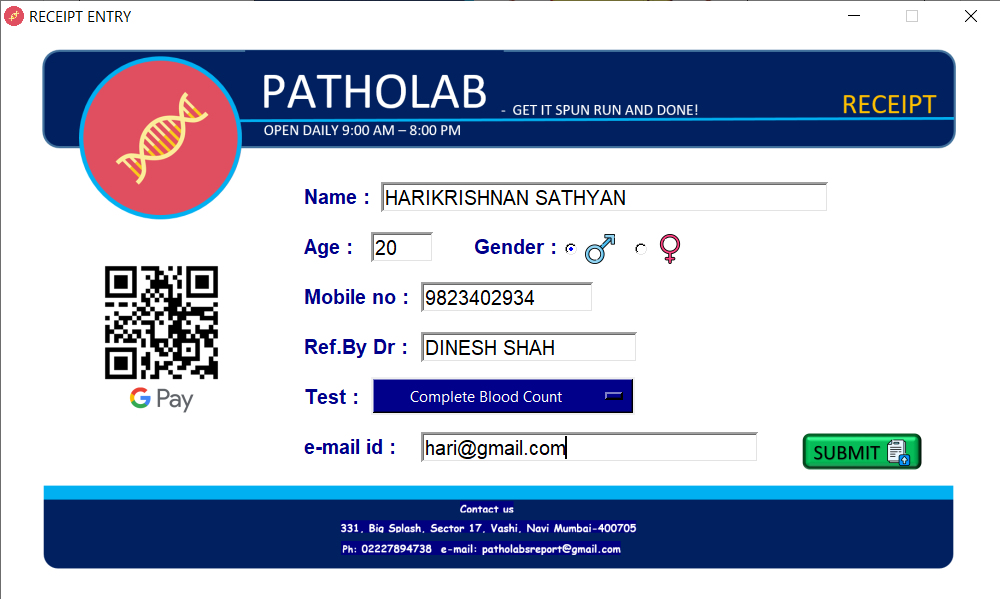
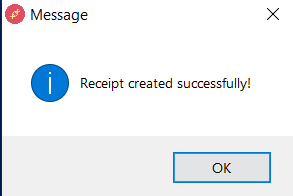
**Function to generate Report pdf**

def generatereportpdf():  
  
 recid = bld\_recidentry.get()  
 pdf = FPDF('P'**,** 'mm'**,** (**210, 297**))  
 pdf.add\_page()  
  
 pdf.set\_font('helvetica'**,** 'B'**, 8**)  
 pdf.set\_text\_color(**0, 0, 0**)  
  
 # Images  
 pdf.image('Images/Report\_toptemp.png'**, 5, 15, 200, 40**)  
 pdf.image('Images/pdf\_report\_down.png'**, 5, 260, 200, 23**)  
 pdf.image('Images/pdf\_backtemp2.png'**, 54, 130, 100, 70**)  
 pdf.image('Images/signature2.jpg'**, 155, 241, 25, 12**)  
  
 pdf.set\_font('helvetica'**,** 'B'**, 6**)  
 pdf.text(**162, 253,** 'Consultant')  
 pdf.text(**153, 257,** '(Microbiologist/Pathologist)')  
  
 conn = sqlite3.connect('Labdb.db')  
 cur = conn.cursor()  
 details = cur.execute(f"SELECT \* FROM RECEIPT WHERE REC\_ID = {recid}")  
 pdetails = details.fetchall()  
 for i in pdetails:  
 pdf.set\_font('helvetica'**,** 'B'**, 10**)  
 pdf.set\_text\_color(**0, 0, 96**)  
 pdf.text(**50, 50,** f'NAME : {i[**1**]}')  
 pdf.text(**150, 50,** f'AGE : {i[**2**]}')  
 pdf.text(**50, 58,** f'GENDER : {i[**3**]}')  
 pdf.text(**50, 66,** f'TEST : {i[**6**]}')  
 pdf.text(**50, 74,** f'DOCTOR : {i[**7**]}')  
 pdf.text(**150, 74,** f'REC ID : {i[**0**]}')  
 pdf.text(**150, 58,** f'DATE : {i[**9**]}')  
 pdf.text(**150, 66,** f'TIME : {i[**10**]}')  
 pdf.set\_font('helvetica'**,** 'BU'**, 12**)  
 pdf.set\_text\_color(**0, 0, 0**)  
 pdf.text(**76, 85,** f'{i[**6**]}')  
  
  
 # TESTS  
 pdf.set\_text\_color(**0, 0, 0**)  
 pdf.text(**20, 93,** 'TEST NAME')  
 pdf.text(**95, 93,** 'RESULT')  
 pdf.text(**155, 93,** 'NORMAL RANGE')  
  
 pdf.set\_font('helvetica'**,** 'B'**, 10**)  
  
 pdf.text(**16, 79,** '\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ ')  
 pdf.text(**16, 233,** '\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ ')  
  
 pdf.set\_font('helvetica'**,** 'BU'**, 12**)  
 pdf.text(**80, 170,** 'DIFFERENTIAL COUNT')  
  
 conn = sqlite3.connect('Labdb.db')  
 cur = conn.cursor()  
 details = cur.execute(f"SELECT \* FROM Report WHERE REC\_ID = {recid}")  
 pdetails = details.fetchall()  
 for i in pdetails:  
 # RESULT  
 pdf.set\_font('helvetica'**,** ''**, 8**)  
 pdf.text(**95, 110,** f'{i[**1**]}')  
 pdf.text(**95, 120,** f'{i[**2**]}')  
 pdf.text(**95, 130,** f'{i[**3**]}')  
 pdf.text(**95, 140,** f'{i[**4**]}')  
 pdf.text(**95, 150,** f'{i[**5**]}')  
 pdf.text(**95, 160,** f'{i[**6**]}')  
 pdf.text(**95, 181,** f'{i[**7**]}')  
 pdf.text(**95, 191,** f'{i[**8**]}')  
 pdf.text(**95, 201,** f'{i[**9**]}')  
 pdf.text(**95, 211,** f'{i[**10**]}')  
 pdf.text(**95, 221,** f'{i[**11**]}')  
 pdf.text(**95, 231,** f'{i[**12**]}')  
  
 # NORMAL VALUES  
  
 pdf.set\_font('helvetica'**,** ''**, 8**)  
 pdf.text(**155, 110,** '11.00 - 16.00')  
 pdf.text(**155, 120,** '3.5 - 5.50')  
 pdf.text(**155, 130,** '42 - 50')  
 pdf.text(**155, 140,** '82 - 95')  
 pdf.text(**155, 150,** '27 - 31')  
 pdf.text(**155, 160,** '4.5 - 11')  
 pdf.text(**155, 181,** '40 - 70')  
 pdf.text(**155, 191,** '1 - 6')  
 pdf.text(**155, 201,** '0 - 2')  
 pdf.text(**155, 211,** '20 - 45')  
 pdf.text(**155, 221,** '2 - 10')  
 pdf.text(**155, 231,** '150 - 450')  
  
 # TEST NAME  
 pdf.set\_font('helvetica'**,** 'B'**, 8**)  
 pdf.text(**20, 110,** 'HEMOGLOBIN')  
 pdf.text(**20, 120,** 'RBC')  
 pdf.text(**20, 130,** 'PCV %')  
 pdf.text(**20, 140,** 'MCV FL')  
 pdf.text(**20, 150,** 'MCH')  
 pdf.text(**20, 160,** 'TOTAL WBC')  
 pdf.text(**20, 181,** 'NEUROPHILES %')  
 pdf.text(**20, 191,** 'EOSINOPHILES %')  
 pdf.text(**20, 201,** 'BASOPHILES %')  
 pdf.text(**20, 211,** 'LYMPHOCYTES %')  
 pdf.text(**20, 221,** 'MONOCYTES %')  
 pdf.text(**20, 231,** 'PLATELET')  
  
 pdf.output('Patient Report.pdf')  
 billing.destroy()  
 webbrowser.open\_new(r'Patient Report.pdf')

**6.2 System Implementation and Output**

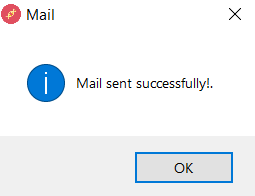
** HOME PAGE   
 FIG 1.1**

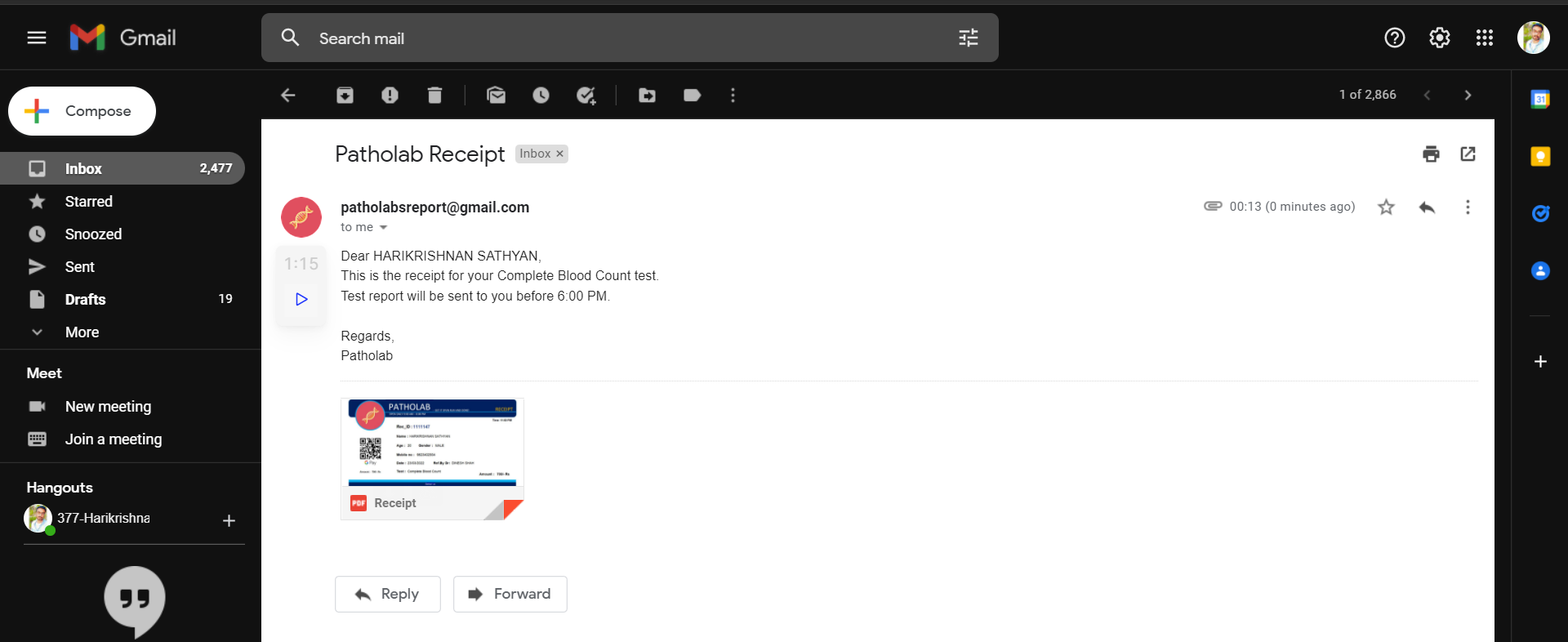
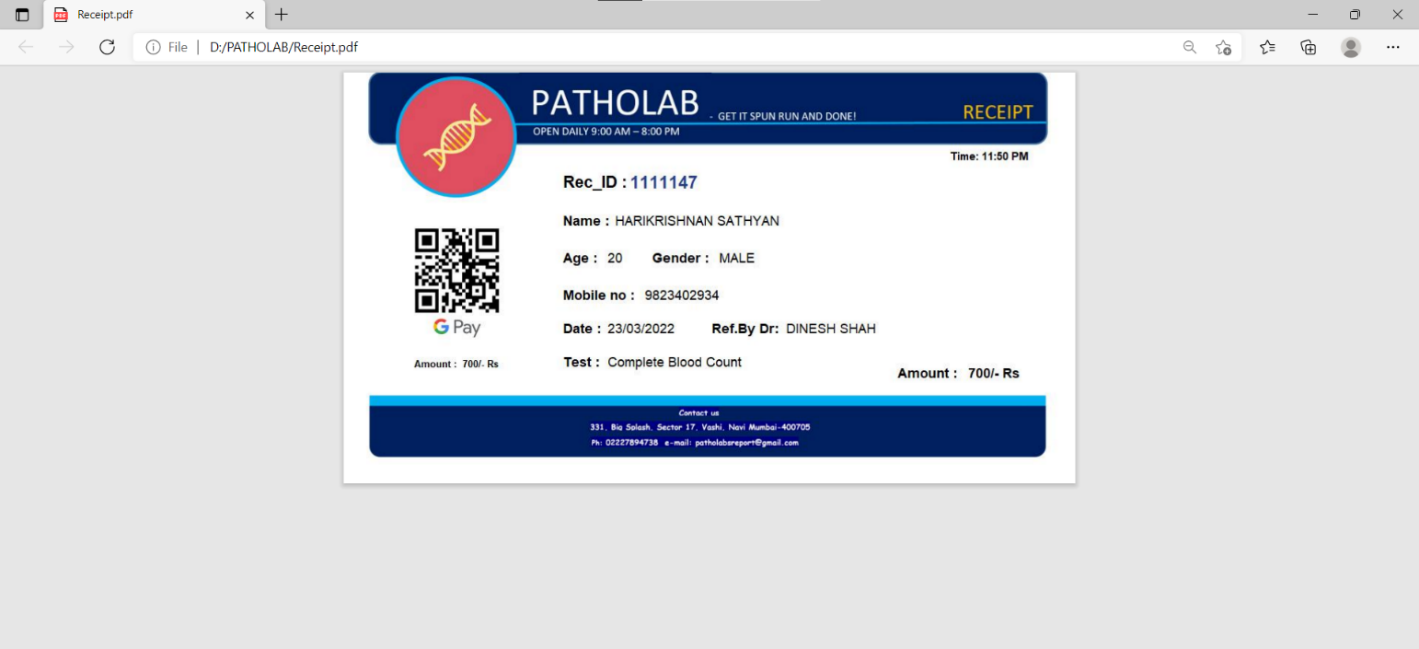
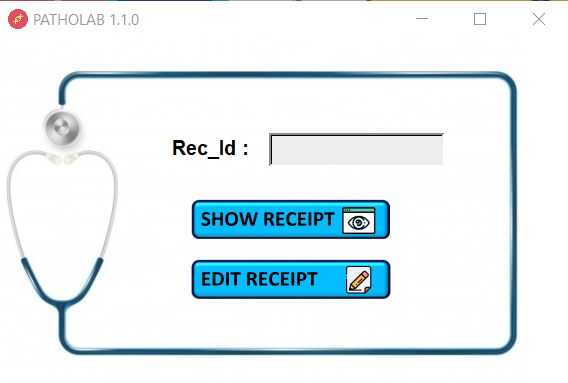
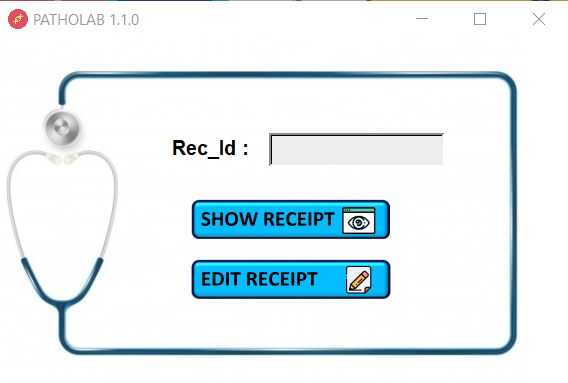
**RECEIPT WINDOW  
 FIG 1.2**

**After inserting the values, click submit.**  **FIG 1.3 FIG 1.4**

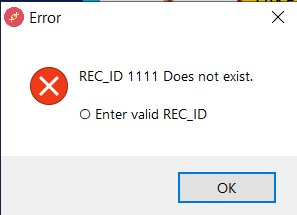
**Display receipt**

**  
 FIG 1.5**

**After clicking on mail  
 FIG 1.6**

**Received mail  
 FIG 1.7  
After print  
 FIG 1.8  
Edit Receipt window**

**FIG 2.1**

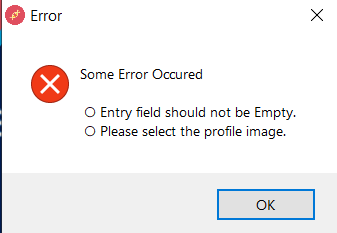
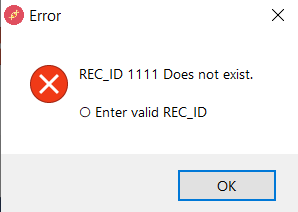
**After writing invalid id   
 FIG 2.2**

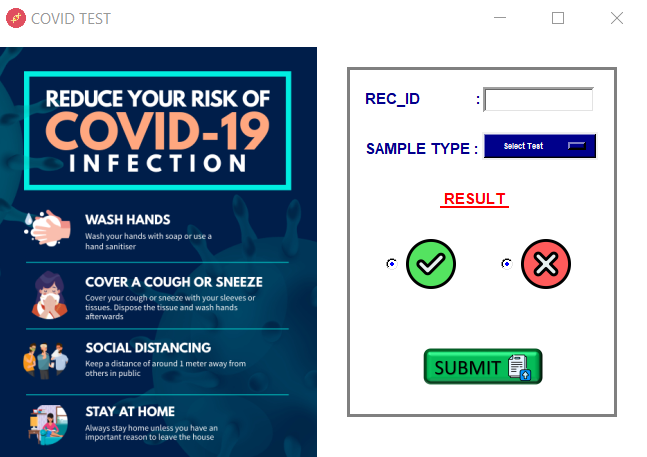
**Show Receipt will display the receipt as shown in FIG 1.5. & can perform functions from FIG 1.5 to FIG 1.8**

**Clicking on Edit Receipt  
 FIG 2.3**

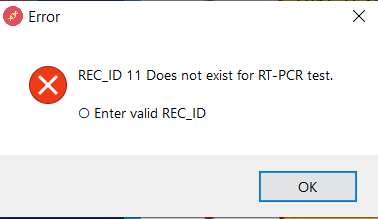
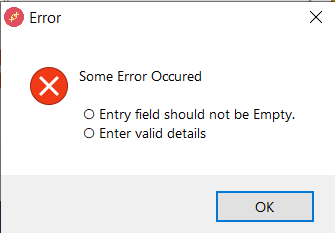
**After editing click on save changes**

**Receipt will be display the receipt as shown in FIG 1.5. & can perform functions from FIG 1.5 to FIG 1.8.**

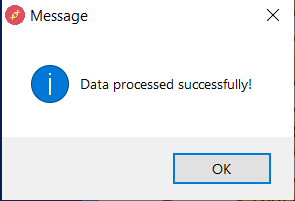
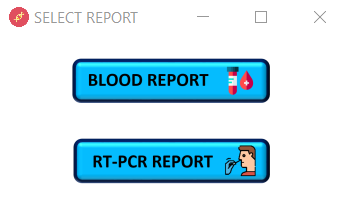
**Take test window  
 FIG 3.1  
  
Entering invaild details will display the following message**   
****  
** FIG 3.2 FIG 3.3**

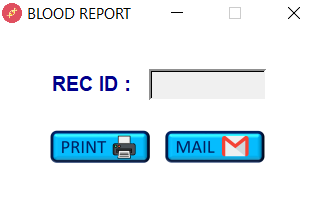
**After inserting details  
 FIG 3.4  
Covid-19 test window  
 FIG 4.1**

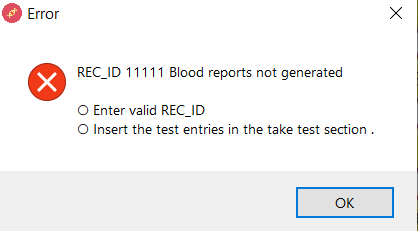
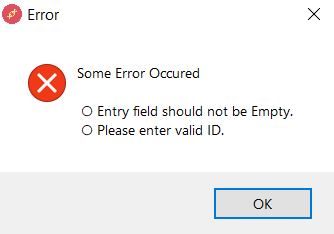
**If invalid details or fields are kept empty**

**** ****  
 **FIG 4.2 FIG 4.3**

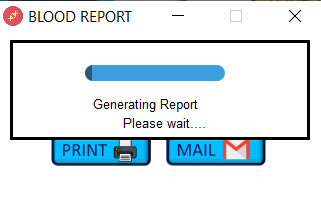
**After entering valid details**

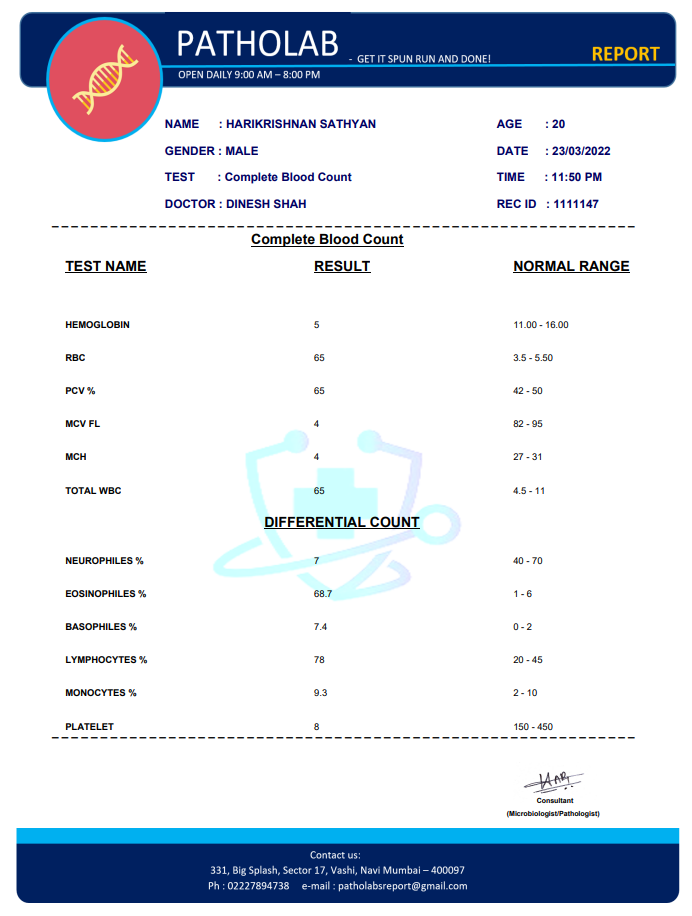
**  
 FIG 4.4  
  
Send Report window  
 FIG 5.1**

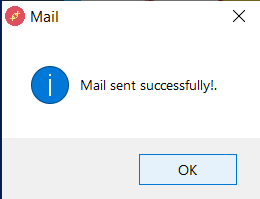
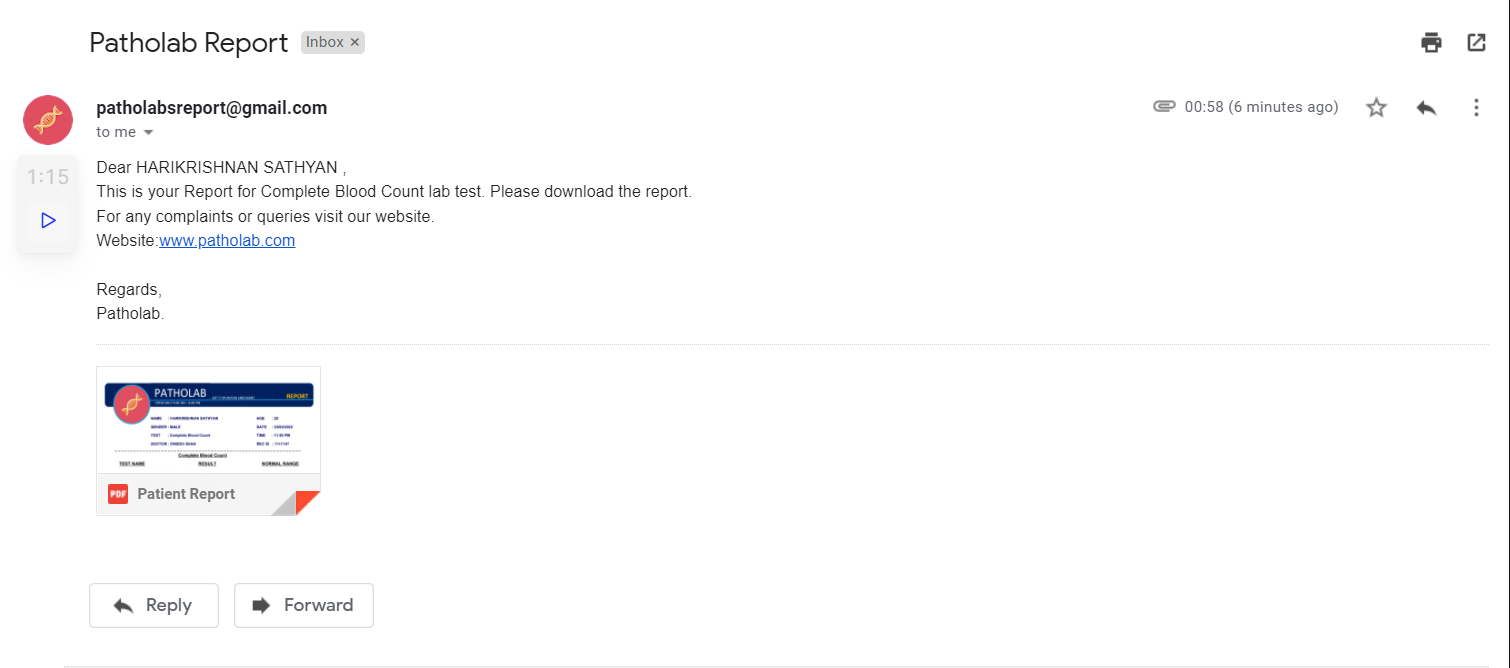
**After clicking on blood report   
 FIG 5.2**

**If invalid details are entered**

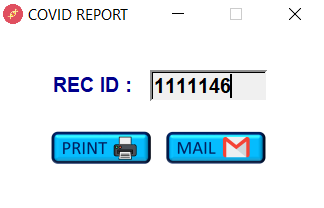
**FIG 5.3 FIG 5.4**

**After clicking on print button  
 FIG 5.5**

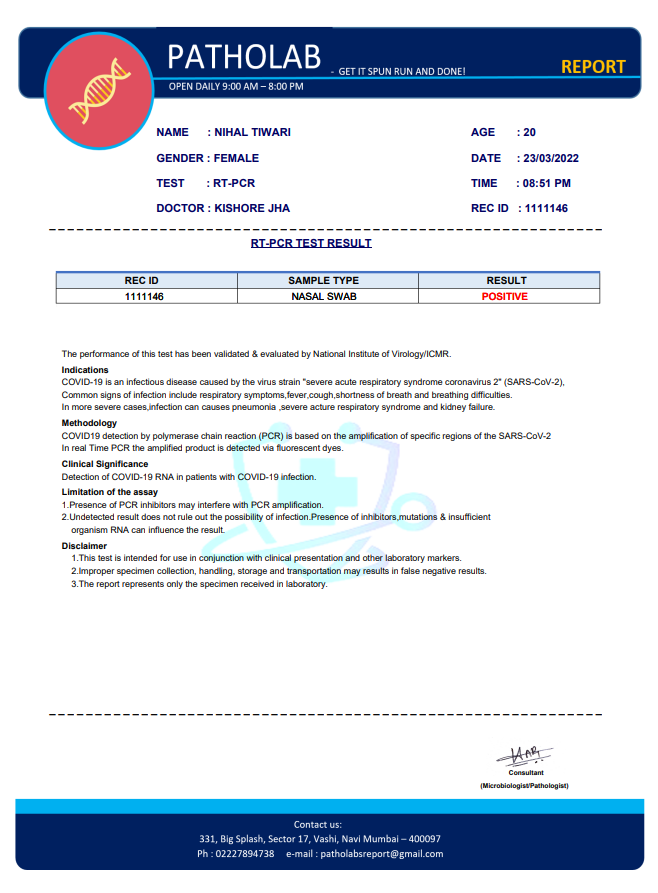
**Report will be generated   
 FIG 5.6**

**After clicking on mail button  
 FIG 5.7**

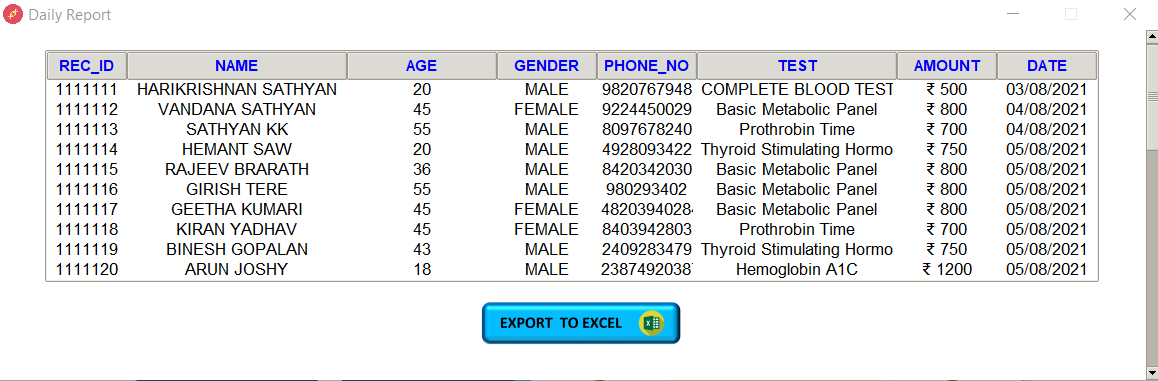
**FIG 5.8**

**Clicking on RTPCR report will open the below window  
 FIG 5.9**

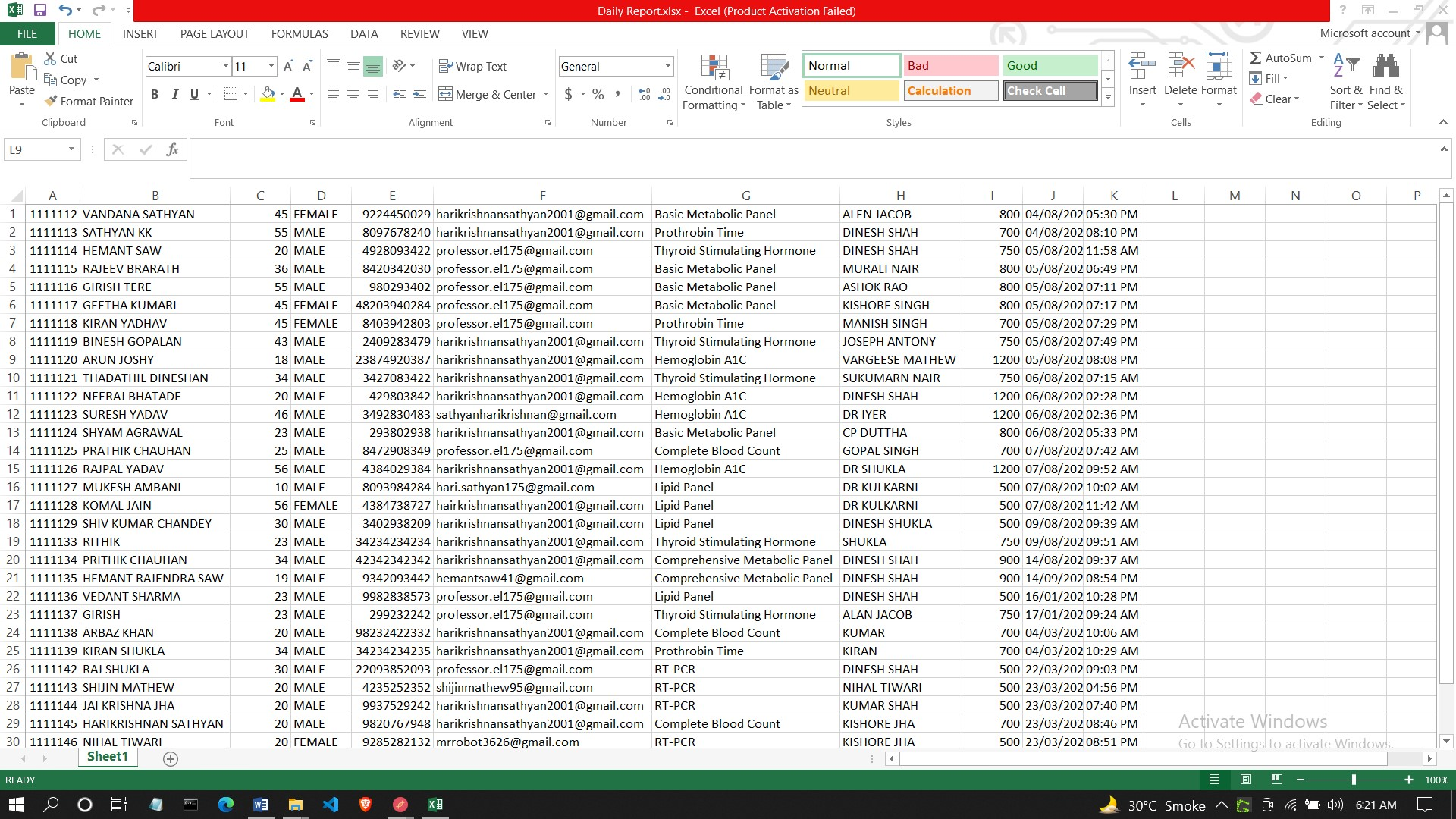
**Entering invalid will pop up messages as shown in fig 5.3 and 5.4.**

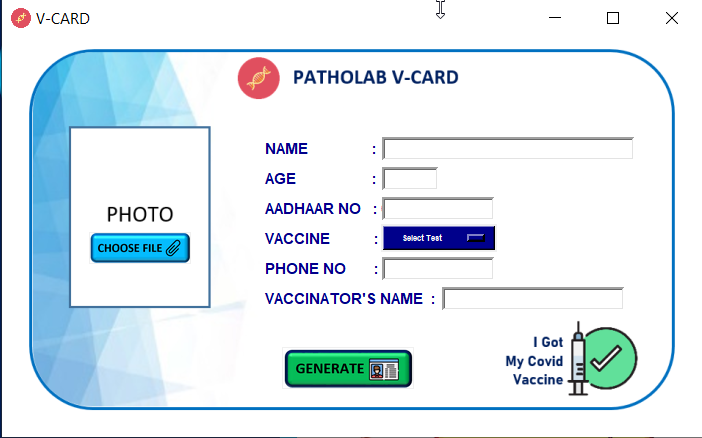
**After clicking on print, Covid report will be generated  
 FIG 5.10**

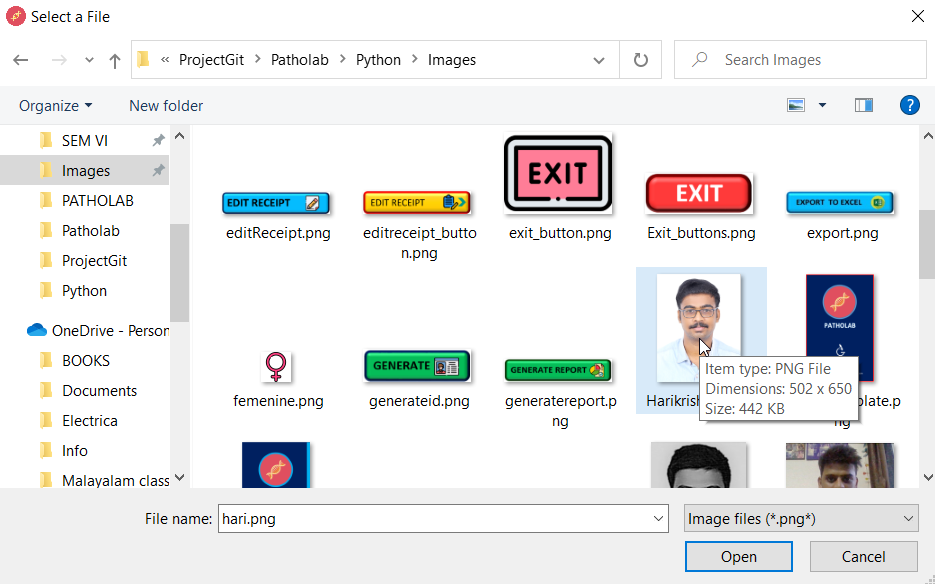
**Clicking on mail will send report as show in fig 5.7 and 5.8**

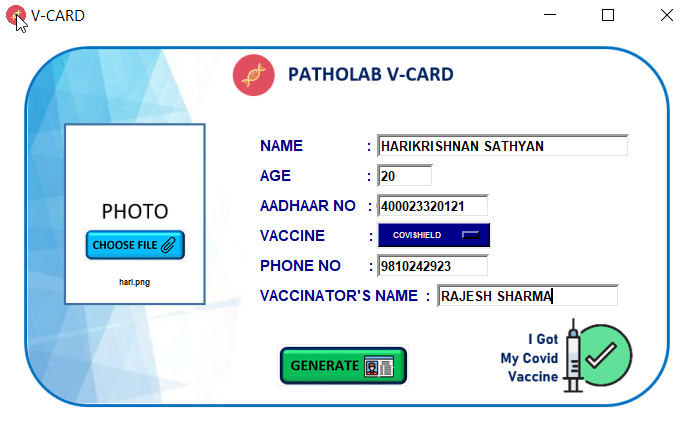
**Daily Report window  
  
 FIG 6.1**

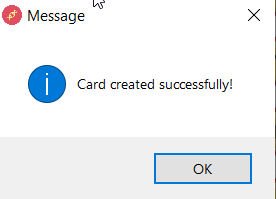
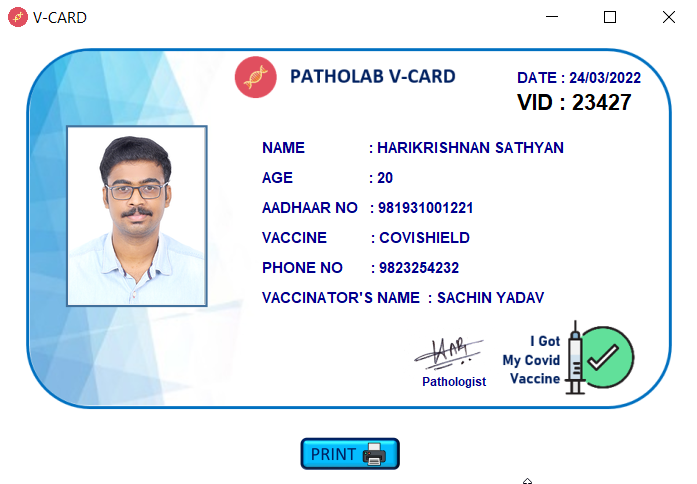
**After clicking on Export to Excel**

**  
 FIG 6.2**

**V-CARD Window  
 FIG 7.1**

**After inserting values select choose files and browse the your image  
 FIG 7.2**

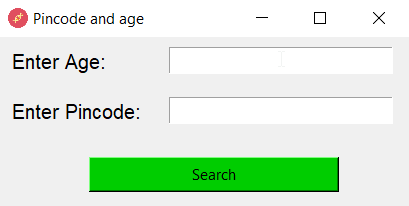
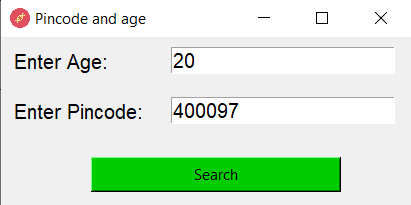
**After choosing the image click generate.  
 FIG 7.3**

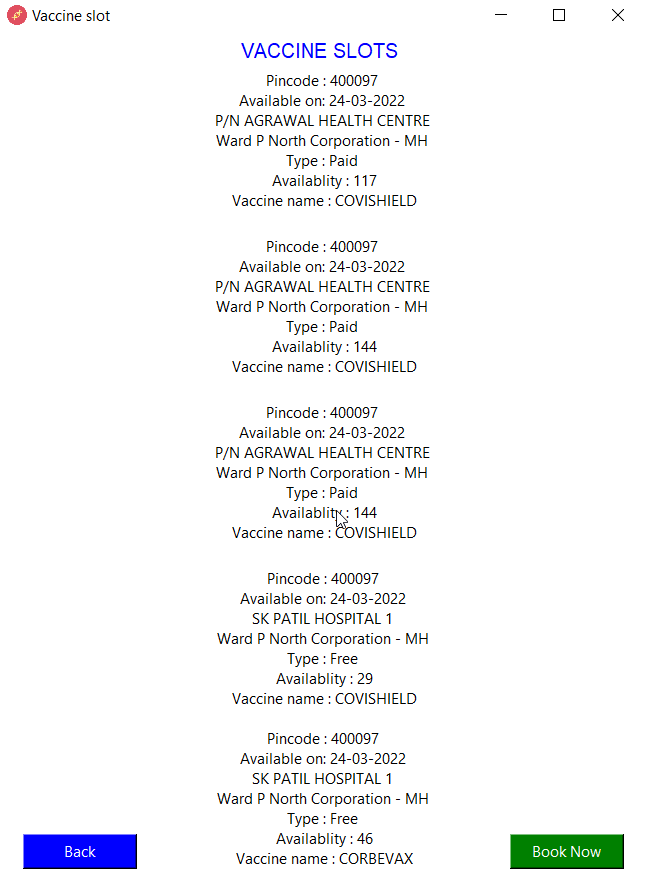
**Card generated successfully message will popup.  
 FIG 7.4  
After clicking ok V-CARD will be displayed.   
 FIG 7.5**

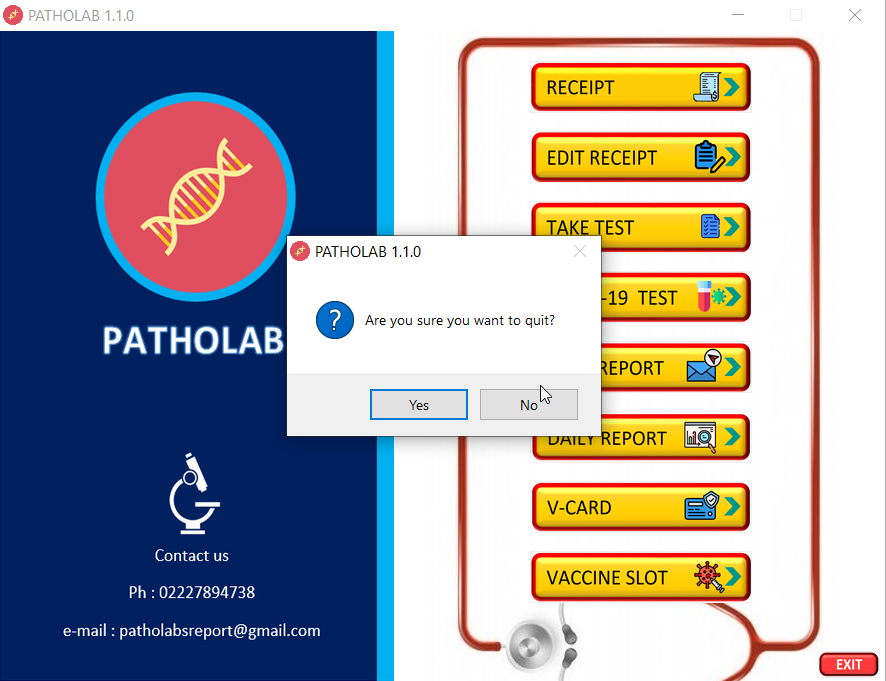
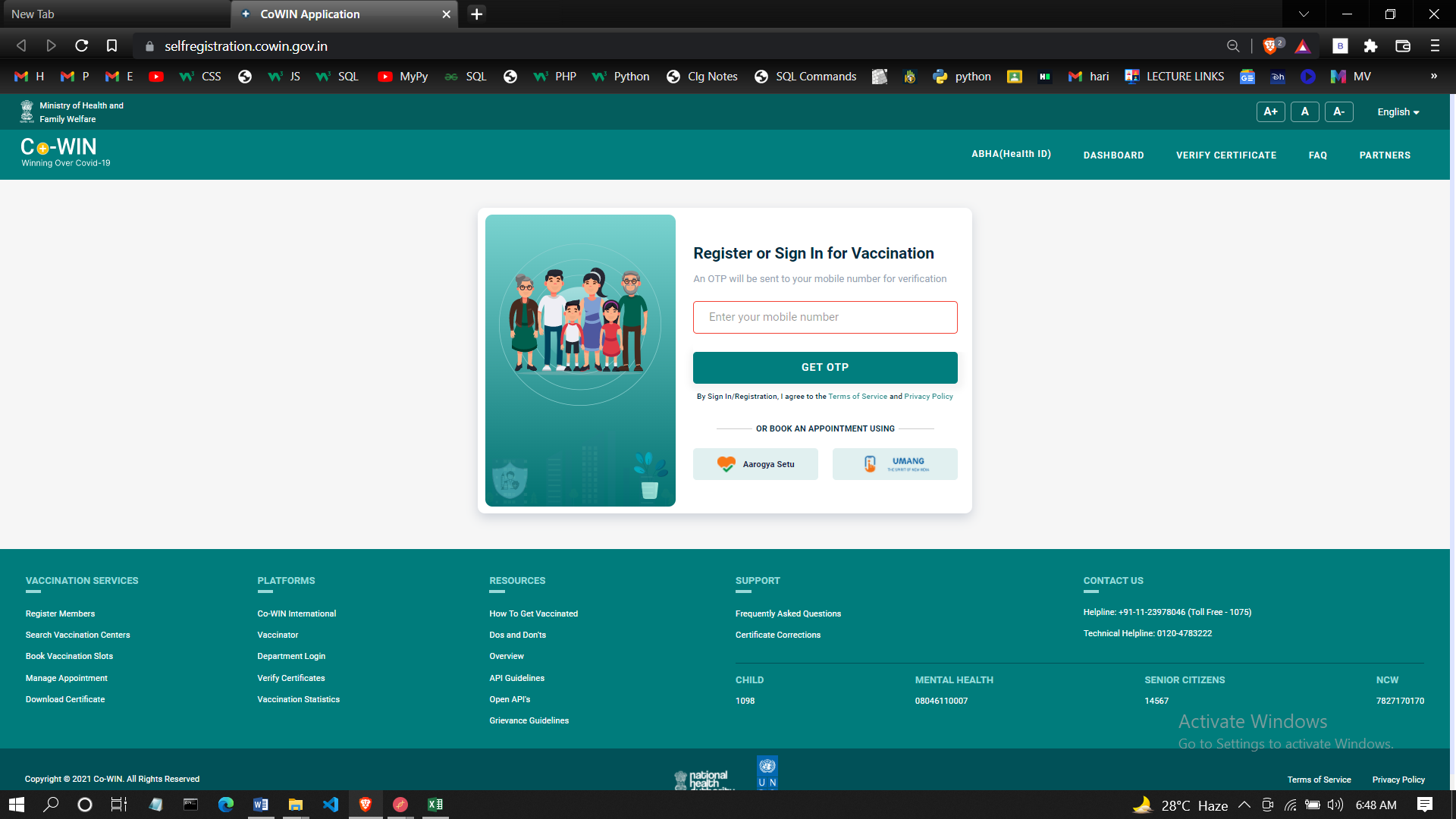
**Clicking on print button will print the V-CARD**

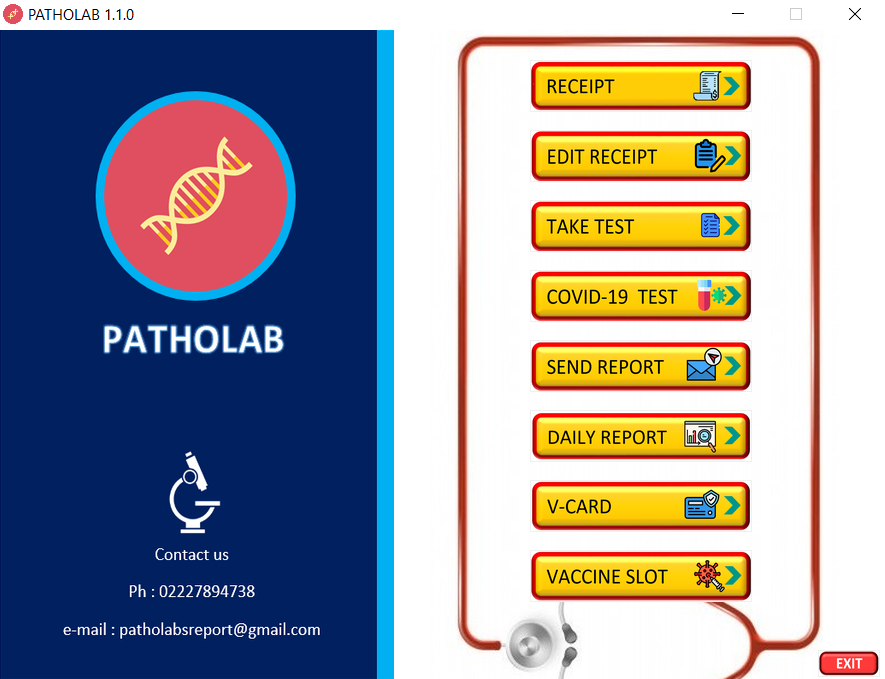
**FIG 7.6**

**  
 FIG 7.7**

**Vaccine slot window  
 FIG 8.1 FIG 8.2**

**After clicking on search button.  
 FIG 8.3**

******Clicking on booknow will redirect to slot booking site  
 FIG 8.4  
  
  
After Clicking Exit button in HOME PAGE  
 FIG 8.5**

**If clicked yes then sofware will be closed else the sofware will remain as it is.  
 FIG 8.6**

**Conclusion**

**7.1 Conclusion**

Patholab is a pathological lab management software which is capable of performing all the operation that are need to run a pathological laboratory. Patholab will help user to manage report generation and processing the data of patients.Reduce the workload on the people who are working in the pathological laboratory.Easily find the vaccination slot available in the locality.Reduce spread of corona virus by sending immediate report to the patient so that they can isolate themselves.Keeping track of people who are vaccinated.  
  
**7.2 Limitations**  
 Since this software is an admin oriented therefore the admin has the full access to the software there is no involvement or interaction with the patients.

**7.3 Future Enhancements** Machine learning can be used in this software for predicting the diseases from the generated report and giving warning to the patients**.**

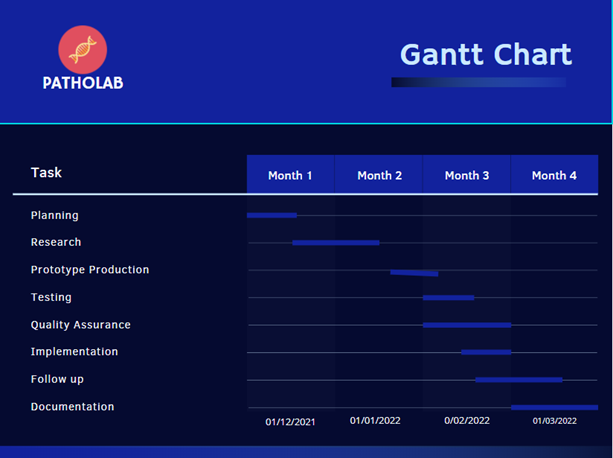
**7.4 Bibliography**

<https://www.python.org/>

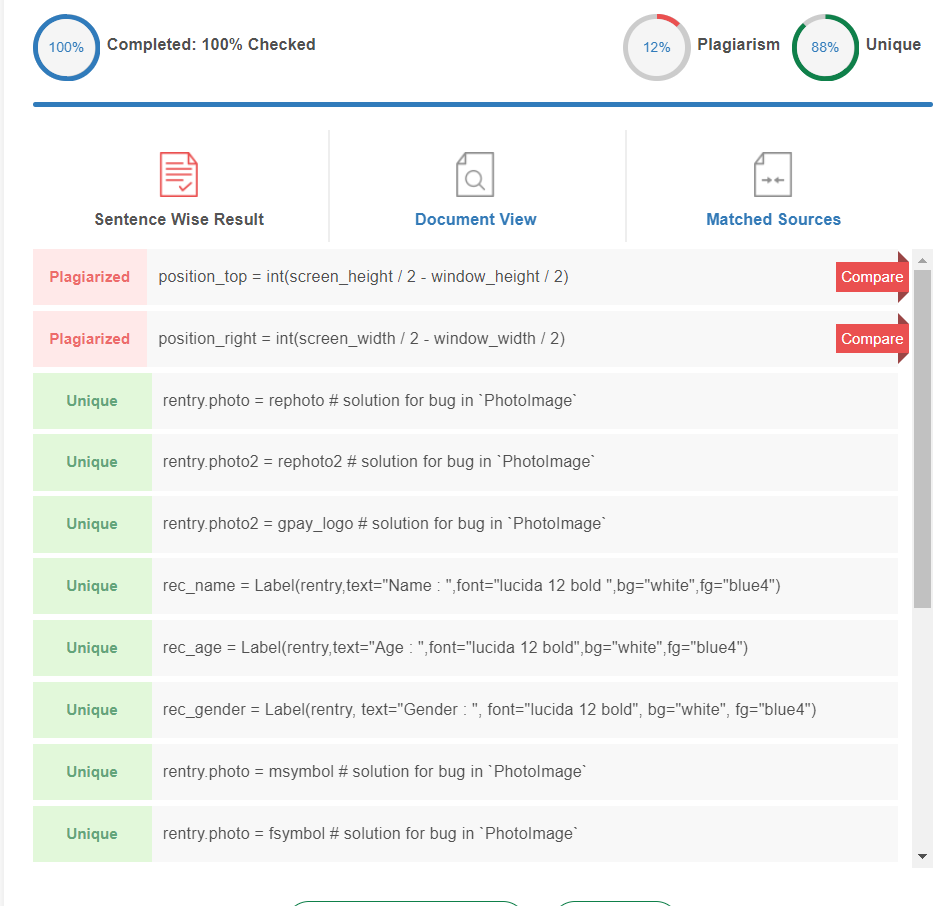
<https://www.youtube.com/codemewo2>

<https://www.tutorialpoint.com/>

<https://www.github.com/>

**Grantt chart**

**Plagiarism Report**

****