

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA” BELAGAVI- 590018, KARNATAKA



DBMS MINI PROJECT REPORT

On

“Hospital Management System”

Submitted in the partial fulfilment of the requirement for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

HARSHITH BABU K

(1RL19CS035)

THUKARAM RAO A.N.V

(1RL19CS001)

HARSHA VARDHAN S

(1RL19CS031)

Under the Guidance of,

Prof. VEENA K



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

R L JALAPPA INSTITUTE OF TECHNOLOGY

DODDABALLAPUR-561 203 (KARNATAKA) 2021-2022

(A Unit of Sri Devaraj Urs Educational Trust (Regd.))

R L JALAPPA INSTITUTE OF TECHNOLOGY

DODDABALLAPUR -561203 (KARNATAKA)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



This is to certify that the Project work entitled “**Hospital Management System**” is a bonafied work carried out by *Thukaram Rao A.N.V(1RL19CS001)*, *Harsha Vardhan S(1RL19CS031)*, *Harshith Babu K(1RL19CS035)* in partial fulfillment for the requirement of V semester, Bachelor of Engineering in Computer Science & Engineering of **VISVESVARAYATECHNOLOGICAL UNIVERSITY**, Belagavi, during the year **2021-22**. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report. This report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for said degree.

Signature of Guide

Mrs. Veena K

Assistant Professor

Dept. of CS&E, RLJIT

Signature of HOD

Prof. Basavaraj S Pol

Dept. of CS&E

RLJIT

Signature of Principal

Dr. M Sreenivasa Reddy

RLJIT

Evaluators

Name

Signature

1.

2.

Table of Contents

Chapter No : Chapter Name	Page No
CHAPTER 1:INTRODUCTION HOSPITAL MANAGEMENT SYSTEM	8
1.1 Introduction	
1.2 Problem Statements	
1.3 Objectives	
1.4 Scope	
CHAPTER 2: DESIGN AND ANALYSIS	10
2.1 Project planning and scheduling	
2.2 Methodology	
2.3 ER Diagram	
2.4 Schema Diagram	
CHAPTER 3: SYSTEM REQUIREMENTS	12
3.1 Hardware Requirements	
3.2 Software Requirements	
3.3 Other Requirements	
3.4 Tool used	
CHAPTER 4: IMPLEMENTATION	13
4.1 Implementation Accepts	
4.2 Front End:PyCharm	
4.3Back End: Sql Query	

Fig6.1 Homepage

Fig6.2 Login Patient Area

Fig6.3 Doctor Login Area

Fig6.4 Admin Login Area

Fig6.5 Appointment Details

Fig6.6 Registered Doctors

Fig6.7 Patient Appointment Details

Fig6.8 Patient Registration Page

Fig6.9 Book Appointment Page

Fig6.10 Hospital View

Fig6.11 The Available Doctors

Fig6.12 Developed By

Fig6.13 Validation Error box

Fig6.14 Forgot Password

Fig6.15 Welcome Message Box

Fig6.16 All Field Required Message Box

CHAPTER 7: CONCLUSION

33

7.1 Conclusions

7.2 Limitations of the system

REFERENCES

34

ACKNOWLEDGEMENT

With Great pride we would like to convey our gratitude and appreciation to our alma-mater **“R. L. Jalappa Institute of Technology”** for giving us the required platform for the fulfilment of the Mini project on ****Hospital Management System**** as per the V.T.U requirements, for the Fifth semester Lab on the DBMS subject.

We express my sincere thanks to **Dr. M Sreenivasa Reddy**, Principal of **R. L. Jalappa Insitiute of Technology**, Dodaballapura for providing me with excellent infrastructure to complete the Mini project.

We express wholehearted gratitude to **Prof. Basavaraj S Pol** who is the respectable **HOD of Computer Science Department**. We wish to acknowledge his help in making my task easy by providing me with his valuable help and encouragement.

It is our pleasure to thank our guide **Mrs. Veena K, Assistant Professor, Department of Computer Science and Engineering**, RLJIT for their guidance, encouragement and valuable suggestion from the beginning of the Mini project work till the completion without which this Mini project work would not have been accomplished. We are greatly indebted to them.

And last but not the least, We would be very pleased to express our heart full thanks to the Teaching and Non-teaching staff of the Department of Computer Science and Engineering, RLJIT for their motivation and support.

We also thank all those who extended their support and co-operation while bringing out this Mini project.

Thukaram Rao A.N.V(1RL19CS001)

Harsha Vardhan S (1RL19CS031)

Harshith Babu K (1RL19CS035)

ABSTRACT

Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support

The project 'Hospital Management System' is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses PyCharm as the front-end software which is an object oriented Programming and has connectivity with MY SQL. Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement.

This package has been widely accepted by the clients in India and overseas. Not stopping only to this but they are highly satisfied and appreciating. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable.

The package is highly customizable and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalities of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Doctor, Admin, Patient appointment, patient details, doctor details admin details etc.

CHAPTER1

INTRODUCTION TO HOSPITAL MANAGEMENT SYSTEM

1.1 Introduction

Human Body is a very complex and sophisticated structure and comprises of millions of functions. All these complicated functions have been understood by man him, part-by-part their research and experiments. As science and technology progressed, medicine became an integral part of the research. Gradually, medical science became an entirely new branch of science. As of today, the Health Sector comprises of Medical institutions i.e. Hospitals, HOSPITALs etc. research and development institutions and medical colleges. Thus the Health sector aims at providing the best medical facilities to the common man

1.2 Problem Statement

Since Hospital is associated with the lives of common people and their day-to-day routines so I decided to work on this project. The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate or make online, the process of day-to-day activities like doctor booking, Admission of New Patient, Details of appointed Patient, Assign a Doctor, and finally compute the over-view of details etc. I have tried my best to make the complicated process Hospital Management System as simple as possible using Structured & Modular technique & Menu oriented interface.

I have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is perform each Hospital's activity in computerized way rather than manually which is time consuming. I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error

1.3 Objective

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, and other staff personals that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process. Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

1.4 Scope

The proposed software product is the Hospital Management system (HMS). The system will be used in any hospital, clinic, dispensary or pathology labs. Clinic, dispensary or pathology to get the information from the patients and then storing that data for future usages. The current system in use is a paper based system. It is too slow and cannot provide updated lists of patients within reasonable time frame. The intention of the system is to reduce over-time pay and increase the number of patients that can be treated accurately. Requirement statements in these documents are both functional and non-functional.

CHAPTER 2

DESIGN AND ANALYSIS

Project Management

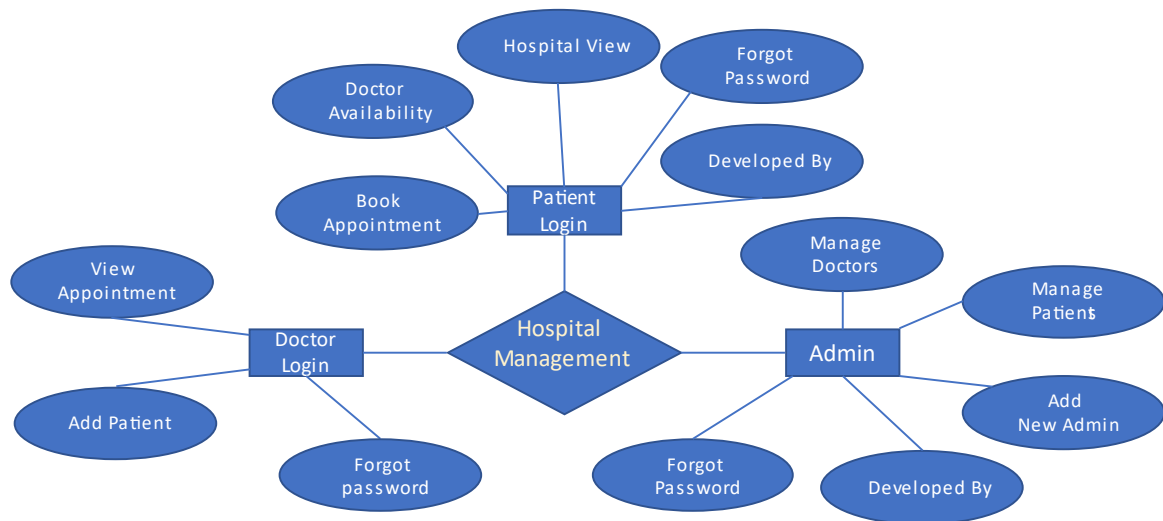
2.1 Project planning and scheduling

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path.

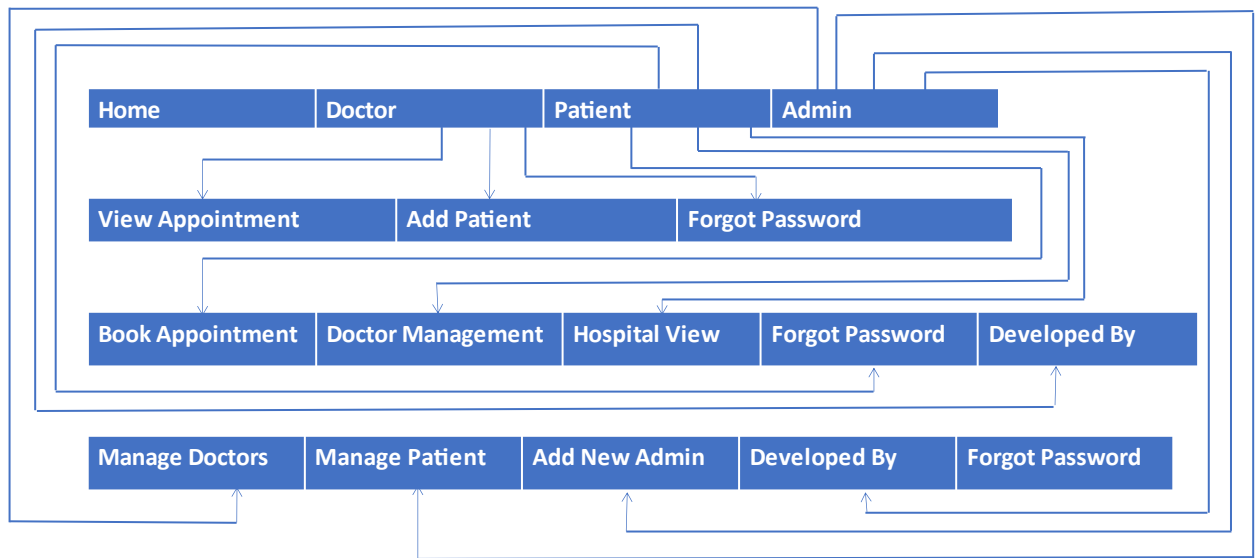
2.2 Methodology

We have used Iterative and Incremental Development model (IID) for our project development. This development approach is also referred to as Iterative Waterfall Development approach. Iterative and Incremental Development is a software development process developed in response to the more traditional waterfall model. This model is designed to take care of such big project. The large and complicate project chiefly demand better development and testing procedure. The waterfall model is well known for its repeated testing process. Hence I choose the waterfall model for developing my software.

2.3 E-R Diagram



2.4 Schema Diagram



CHAPTER 3

SYSTEM REQUIREMENT

3.1 Hardware Requirements:

- 64-bit versions of Microsoft Windows 10, 8
- 2 GB free RAM minimum, 8 GB of total system RAM recommended
- 2.5 GB hard disk space, SSD recommended
- 1024x768 minimum screen resolution
- Python 2.7, or Python 3.5 or newer

3.2 Software Requirements:

- XAMPP for **Windows** 7.4.28, 8.0.17 & 8.1.4
- New XAMPP release 7.4.28 , 8.0.17, 8.1.4 Mar 18
- phpMyAdmin 4.9.10 and 5.13

3.3 Toolused:

- Xampp
- PyCharm

CHAPTER 4

IMPLEMENTATION

4.1 Implementation Accepts:

Detailed Design of Implementation: This phase of the system development life cycle defines hardware and software specification, establishes programming plans, trains users and implements extensive testing procedures to evaluate design and operating specifications and/or provide the basis for further modification.

Technical Design: The activity builds upon the specifications produced during new system design adding detailed technical specifications and documentation.

Test Specifications and Planning: This activity prepares detailed test specifications for individual modules and program jobs streams subsystem for the system as a whole.

Programming and Testing: This activity encompasses actual development writing and testing of the program units or modules.

User Training: Thus activity encompasses writing user procedure manuals preparation of user training materials conducting training programs and testing procedures.

Installation Phase: In this phase new computerized system is installed the conversion to new procedure is fully implemented and potential of new system is explored.

System Installation: The process of starting the actual use of a system and training user personnel in its operation.

Review Phase: This phase evaluates the successes and failures during a system development project and to measure the results of new computerised trans system in terms of benefits and savings projected at the start of the project.

Development Recap: A review of a project immediately after completion to find successes and potential problems in future work.

Post-Implementation Review: A review conducted after a new system has been in operation for some time to evaluate actual system performance against original expectations and projections for
Cost-benefits improve the system.

4.2 Front End

```
import tkinter
import tkinter as Tk
from tkinter import *
from tkinter import ttk
from PIL import ImageTk
from tkinter import messagebox
import pymysql

def main():
    win=Tk()
    app=login(win)
    win.mainloop()

class login:
    def __init__(self,root):
        self.root=root
        self.root.title("Sign Up")
        self.root.geometry("1350x700+0+0")
        #self.root.resizable(False,False)

        #bg
        self.bg =
ImageTk.PhotoImage(file='C:\\Users\\thuku\\OneDrive\\Desktop\\Mini_Project\\mainpage.png')
        bg = Label(self.root, image=self.bg).place(x=0, y=0, relwidth=1, relheight=1)

        title = Label(self.root,text="HOSPITAL MANAGEMENT SYSTEM", font=("helvetica",
35, "bold"), fg="black", bg="white").place(x=50, y=85,width=1140)

        dct_btn = Button(self.root, text="Doctor", font=("helvetica", 25, "bold"), fg="black",
bg="white",
```

```

        cursor='hand2',activebackground='darkblue',activeforeground='white',
        command=self.home_dct).place(x=384, y=590,width=150,height=49)

    user_btn = Button( self.root,text="Patient", font=("helvetica", 25, "bold"), fg="black",
    bg="white",
        cursor='hand2',activebackground='darkblue',activeforeground='white',
        command=self.home_pat).place(x=573,y=590,width=150,height=49)

    adm_btn = Button( self.root,text="Admin", font=("helvetica", 25, "bold"), fg="black",
    cursor='hand2',bg="white",activebackground='darkblue',activeforeground='white',
        command=self.home_adm).place(x=760, y=590,width=150,height=49)

    def home_dct(self):
        self.new_window=Toplevel(self.root)
        self.app=doctor_login(self.new_window)

    def home_pat(self):
        self.new_window = Toplevel(self.root)
        self.app = user_login(self.new_window)

    def home_adm(self):
        self.new_window = Toplevel(self.root)
        self.app = admin_login(self.new_window)
class doctor_login:
    def __init__(self,root):
        self.root=root
        self.root.title("Sign Up")
        self.root.geometry("1350x700+0+0")
        #self.root.resizable(False,False)

```



```

self.bg =
ImageTk.PhotoImage(file='C:\\Users\\thuku\\OneDrive\\Desktop\\Mini_Project\\d2.png')
bg = Label(self.root, image=self.bg).place(x=0, y=0, relwidth=1, relheight=1)

title=Label(self.root,text="Doctor Login
",font=("Impact",33,"bold","underline"),fg="orange",bg="white")
title.place(x=160,y=125)
desc=Label(self.root,text="Doctor Login Area",font=("Goudy old
style",16,"bold","underline"),fg="orange",bg="white").place(x=210,y=195)

email=Label( self.root,text="Email", font=("times new roam", 20, "bold"), bg="white",
fg="black").place(x=80, y=260)
self.email= Entry(self.root, font=("times new roman", 15),bg="lightgray")
self.email.place(x=80, y=305, width=350,height=40)

password = Label(self.root,text="Password ", font=("times new roam", 20, "bold"),
bg="white", fg="black").place(x=80, y=360)
self.password= Entry(self.root,font=("times new roman", 15),bg="lightgray",show="*")
self.password.place(x=80, y=405, width=350,height=40)

forget_btn = Button(self.root,text="Forgot
Password?",bg="white",fg="red",bd=0,font=("times new roman",13),
command=self.email_frgt).place(x=340,y=455)

Login_btn = Button( self.root,text="Submit ", bg="white", fg="red",font=("times new
roman", 20),cursor='hand2',
command=self.usr_fun,).place(x=230, y=503)

```

```
class viewap_window:

    def __init__(self, root):

        self.root = root

        self.root.title("Appointment Details")

        self.root.geometry("1000x320+100+150")

        self.root.focus_force()

        self.root.grab_set()

        self.root.config(bg="white")

        self.root.resizable(False, False)

        Frame_login = Frame(self.root, bg="#FFE4E1", bd=10, relief=RIDGE)

        Frame_login.place(x=12, y=20, height=280, width=970)

        connect= pymysql.connect(host="localhost", user="root", password="",
database="project",port=3306)

        con = connect.cursor()

        con.execute("select * from appointment")

        tree=ttk.Treeview(Frame_login)

        tree['show']='headings'

        s=ttk.Style(Frame_login)

        s.theme_use("clam")

        tree["columns"]=("pat_name", "gender", "age", "cno", "address", "symptoms", "date", "doctor")
```

```
tree.column( "pat_name",width=100,minwidth=50,anchor=tkinter.CENTER)

tree.column("gender", width=100, minwidth=50, anchor=tkinter.CENTER)

tree.column("age", width=100, minwidth=50, anchor=tkinter.CENTER)

tree.column("cno", width=100, minwidth=50, anchor=tkinter.CENTER)

tree.column("address", width=100, minwidth=50, anchor=tkinter.CENTER)

tree.column("symptoms", width=100, minwidth=50, anchor=tkinter.CENTER)

tree.column("date", width=130, minwidth=50, anchor=tkinter.CENTER)

tree.column( "doctor",width=130,minwidth=50,anchor=tkinter.CENTER)


tree.heading("pat_name",text="Patient name",anchor=tkinter.CENTER)

tree.heading("gender", text="Gender", anchor=tkinter.CENTER)

tree.heading("age", text="Age", anchor=tkinter.CENTER)

tree.heading("cno", text="Contact Number", anchor=tkinter.CENTER)

tree.heading("address", text="Address", anchor=tkinter.CENTER)

tree.heading("symptoms", text="Symptoms", anchor=tkinter.CENTER)

tree.heading("date", text="Appointment Date", anchor=tkinter.CENTER)

tree.heading("doctor", text="Appointed Doctor", anchor=tkinter.CENTER)

i=0
```

for rows in con:

```
tree.insert("",i,text="",values=(rows[0],rows[1],rows[2],rows[3],rows[4],rows[5],rows[6],rows[7]))
```

```
i=i+1
```

```
tree.pack()
```

```
class admin_login:
```

```
def __init__(self,root):
```

```
self.root=root
```

```
self.root.title("Admin Area")
```

```
self.root.geometry("1350x700+0+0")
```

```
#self.root.resizable(False,False)
```

```
self.bg = ImageTk.PhotoImage(file='C:\\Users\\thuku\\OneDrive\\Desktop\\Mini_Project\\p3.jpg')
```

```
bg = Label(self.root, image=self.bg).place(x=0, y=0, relwidth=1, relheight=1)
```

```
Frame_login=Frame (self.root,bg="white")
```

```
Frame_login.place(x=730,y=85,height=520,width=450)
```

```
title=Label(self.root,text="Admin Login",font=("Impact",35,"bold","underline"),fg="orange",bg="white")
```

```
title.place(x=833,y=125)
```

```
desc=Label(self.root,text="Admin Login Area",font=("Goudy old style",15,"bold","underline"),fg="orange",bg="white").place(x=875,y=195)
```

```
email=Label( self.root,text="Email ", font=("times new roam", 20, "bold"), bg="white", fg="black").place(x=750, y=260)
```

```
self.email= Entry(self.root, font=("times new roman", 15),bg="lightgray")
```

```
self.email.place(x=750, y=305, width=350,height=40)
```

```
self.pasw = Label(self.root,text="Password ", font=("times new roam", 20, "bold"), bg="white", fg="black").place(x=755, y=360)
```

```
self.pasw= Entry(self.root,font=("times new roman", 15),bg="lightgray",show="*")
```

```

self.pasw.place(x=750, y=405, width=350,height=40)

forget_btn = Button(self.root,text="Forgot Password?",bg="white",fg="red",bd=0,font=("times new
roman",13),

        command=self.email_frgt,).place(x=1020,y=455)

Login_btn = Button( self.root,text="Submit ", bg="white", fg="red",font=("times new roman",
20),cursor='hand2',

        command=self.log_fun).place(x=895, y=505)

home_btn = Button(self.root, text="Home", bg="white", fg="black", font=("times new roman", 17),

        cursor='hand2', bd=2,command=self.new013_window).place(x=0, y=0, width=215)

admin_btn = Button(self.root, text="Doctor", bg="white", fg="black", font=("times new roman", 17),

        cursor='hand2', bd=2,command=self.new014_window).place(x=215, y=0, width=215)

doctor_btn = Button(self.root, text="patient", bg="white", fg="black", font=("times new roman",
17),

        cursor='hand2', bd=2,command=self.new015_window).place(x=430, y=0, width=215)

patient_btn = Button(self.root, text="Admin", bg="white", fg="black", font=("times new roman",
17),

        cursor='hand2', bd=2).place(x=645, y=0, width=215)

abt_btn = Button(self.root, text="Developed By", bg="white", fg="black", font=("times new roman",
17),

        cursor='hand2', bd=2,command=self.pic_window).place(x=860, y=0, width=215)

cnt_btn = Button(self.root, text="Back", bg="white", fg="black", font=("times new roman", 17),

        cursor='hand2', bd=2,command=self.back_window).place(x=1075, y=0, width=215)

def new013_window(self):

    self.new_window = Toplevel(self.root)

    self.app = login(self.new_window)

def new014_window(self):

    self.new_window = Toplevel(self.root)

```

```
        self.app = doctor_login(self.new_window)

def new015_window(self):

    self.new_window = Toplevel(self.root)

    self.app = user_login(self.new_window)

def pic_window(self):

    self.new_window = Toplevel(self.root)

    self.app = pic_page(self.new_window)

def back_window(self):

    self.new_window = Toplevel(self.root)

    self.app = login(self.new_window)


if __name__ == "__main__":

    main()
```

4.3 SQLQUERIES

Table structure for table `Admin Registration`

```
CREATE TABLE `adm_reg` ( `fname` varchar(25) NOT NULL, `lname`  
varchar(15) NOT NULL, `contact` varchar(2) NOT NULL, `email` int(10)  
NOT NULL, `state` varchar(50) NOT NULL, `dob` varchar(12) NOT  
NULL, `cmb_quest` varchar(50) NOT NULL, `answer` varchar(20) NOT  
NULL, `jdate` varchar(50) NOT NULL, `qualif` varchar(50) NOT  
NULL, `password` varchar(30) NOT NULL ) ENGINE=InnoDB DEFAULT  
CHARSET=utf8mb4
```

Table structure for table `Appointment`

```
CREATE TABLE `appointment` ( `pat_name` varchar(25) NOT NULL, `gender` varchar(15) NOT  
NULL, `age` varchar(2) NOT NULL, `cno` varchar(10) NOT NULL, `address` varchar(50) NOT  
NULL, `symptoms` varchar(50) NOT NULL, `date` varchar(15) NOT NULL, `doctor` varchar(20) NOT  
NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

Table structure for table `Doctor Registration`

```
CREATE TABLE `dct_reg` ( `fname` varchar(25) NOT NULL, `lname` varchar(15) NOT NULL, `contact`  
int(10) NOT NULL, `email` varchar(50) NOT NULL, `state` varchar(50) NOT NULL, `dob` varchar(15)  
NOT NULL, `cmb_quest` varchar(50) NOT NULL, `answer` varchar(50) NOT NULL, `spcl` varchar(50)  
NOT NULL, `qualif` varchar(50) NOT NULL, `password` varchar(20) NOT NULL ) ENGINE=InnoDB  
DEFAULT CHARSET=utf8mb4
```

Table structure for table `Patient Registration`

```
CREATE TABLE `pat_reg` ( `fname` varchar(50) NOT NULL, `lname` varchar(50) NOT  
NULL, `contactno` varchar(10) NOT NULL, `email` varchar(50) NOT NULL, `state` varchar(20) NOT  
NULL, `dob` varchar(12) NOT NULL, `cmb_sque` varchar(50) NOT NULL, `answer` varchar(25) NOT  
NULL, `password` varchar(8) NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

CHAPTER 5

SYSTEM TESTING

Testing of the process in which the system is run on manually created input so that the system is correctly working on the desired or not.

During system testing the system is used experimentally to ensure that the software does not fail in other words we can say that it will run accordingly to its specifications and in the way user expects special test data are input for processing and the results examined.

A limited number of users may be allowed to use the system so that analyst can see whether they try to use it in unforeseen ways. It is desirable to discover any surprises before the organisation implements the system and depends on it.

Testing of a system is generally done in two phase-one is **Unit Testing** which is done for each module independently on its completion and other one is **Software Testing** which is done at the end of a project.

- **Unit Testing:**

Unit testing focuses verification effort on the smallest unit of the software design module. The Unit testing we have is white box oriented and some modules the steps are conducted in parallel.

- **Software Testing:**

Software testing may also be viewed in the context of the spiral . Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in the source code. Testing progress by moving outward along the spiral to integrate testing where the focus is on the design and the construction of the software architecture.

Moving inward along the spiral we come to design and finally to coding to develop computer software spiral in along streamlines that decreases the level of abstraction on each turn.

CHAPTER6

SNAPSHOTS



Fig6.1 Homepage

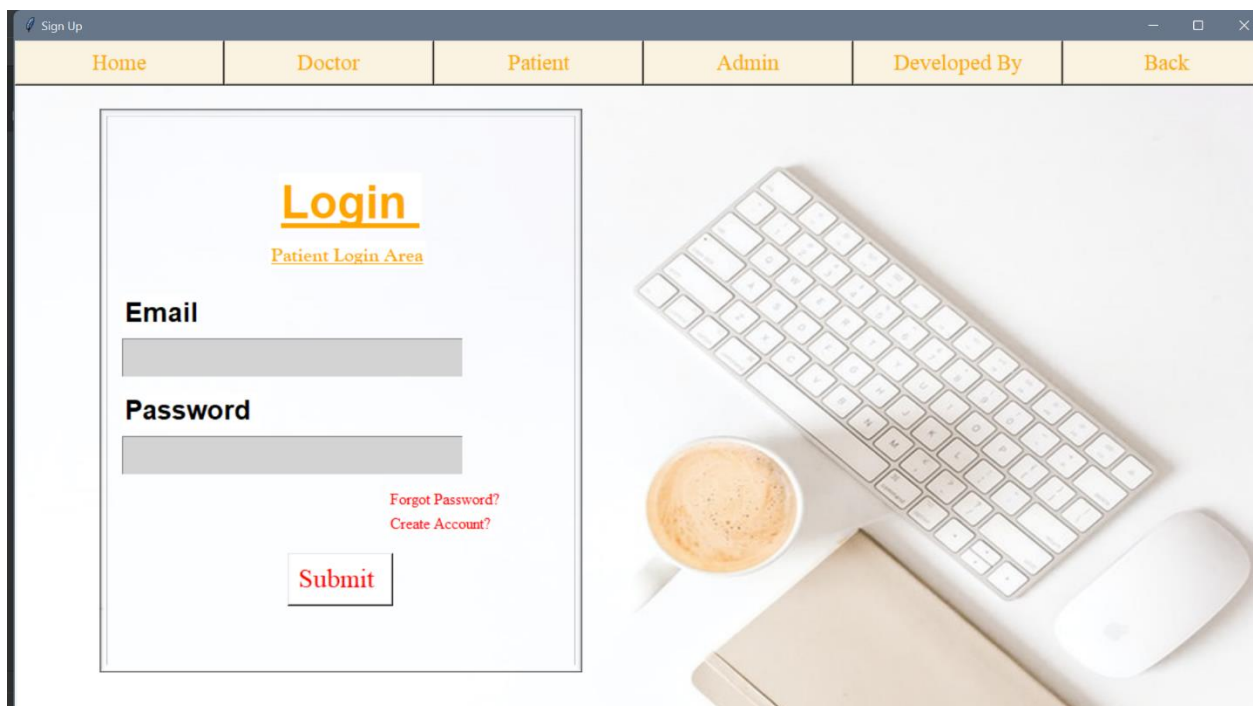


Fig6.2 Login Patient Area

Sign Up

Home	Doctor	Patient	Admin	Developed By	Back
------	--------	---------	-------	--------------	------

Doctor Login

[Doctor Login Area](#)

Email

Password

[Forgot Password?](#)

Submit

Fig6.3 Doctor Login Area

Sign Up

Home	Doctor	patient	Admin	Developed By	Back
------	--------	---------	-------	--------------	------

Admin Login

[Admin Login Area](#)

Email

Password

[Forgot Password?](#)

Submit

Fig6.4 Admin Login Area

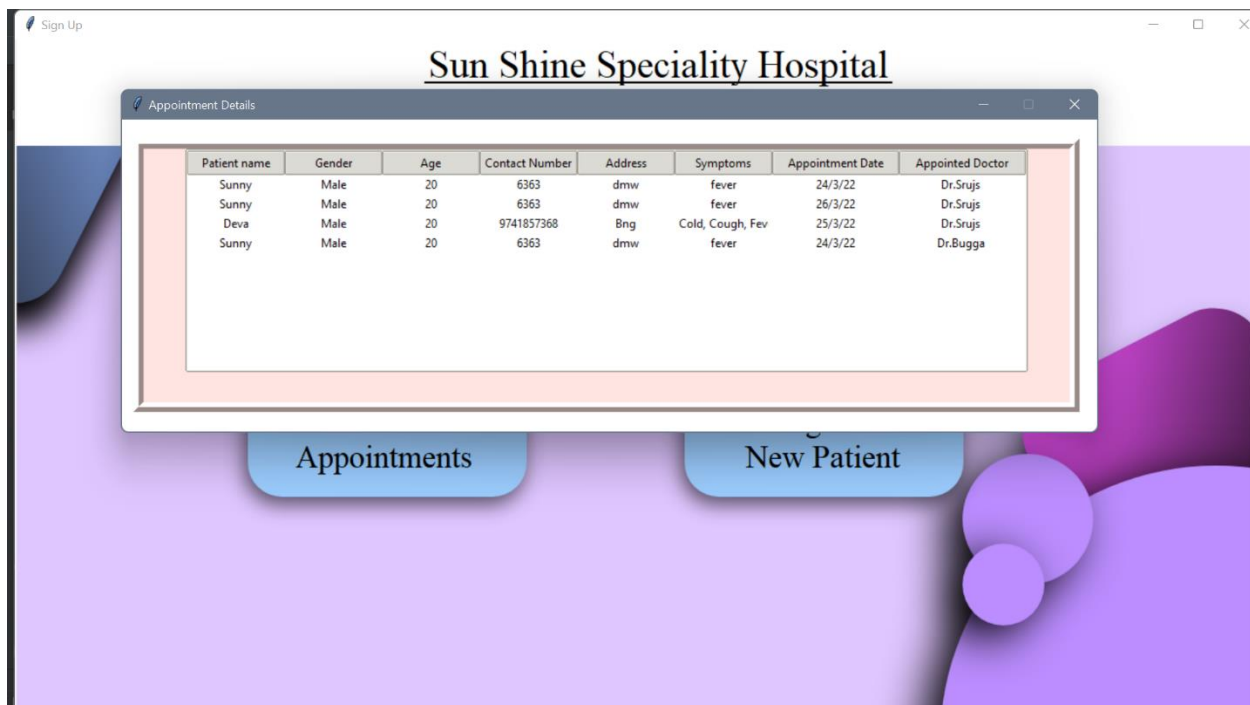


Fig6.5 Appointment Details

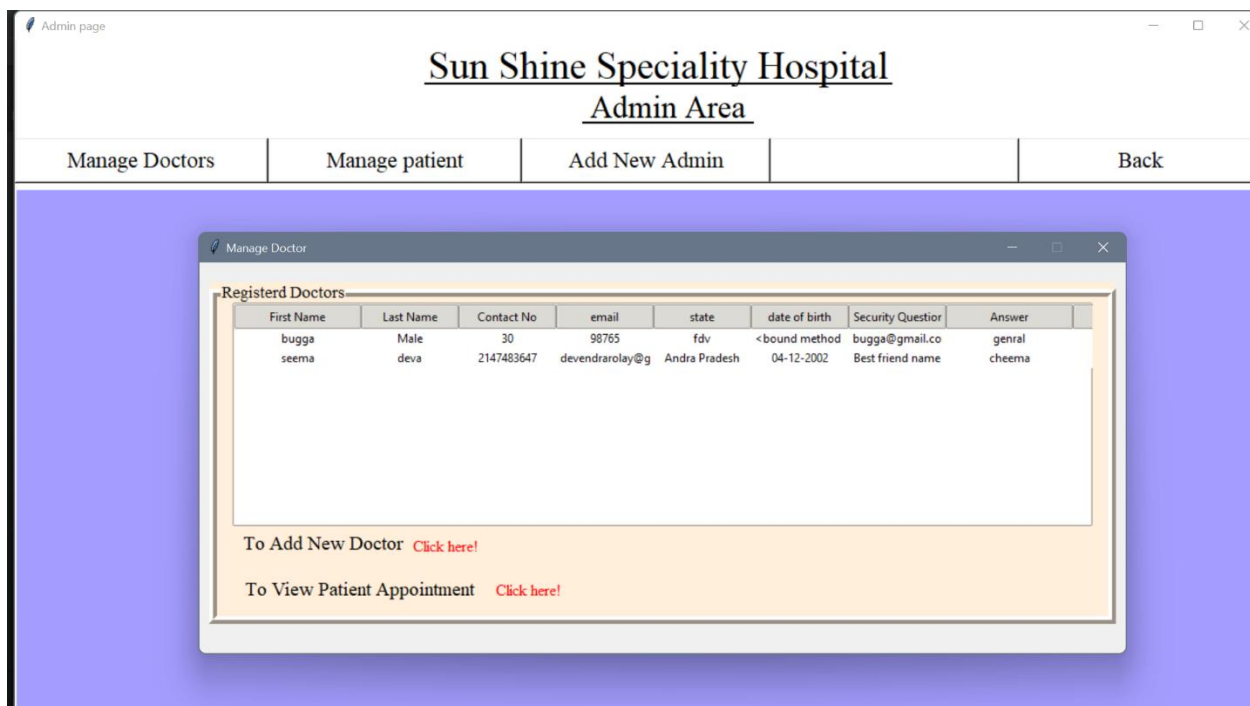


Fig6.6 Registered Doctors

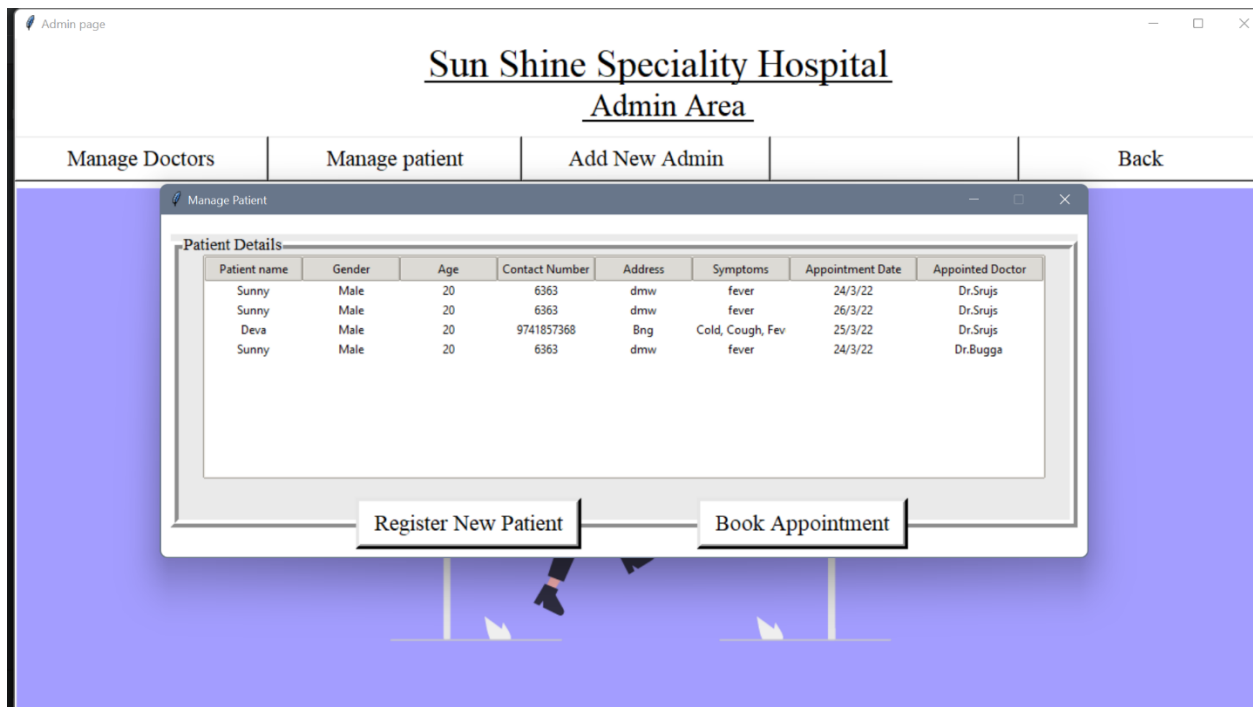


Fig6.7 Patient Appointment Details

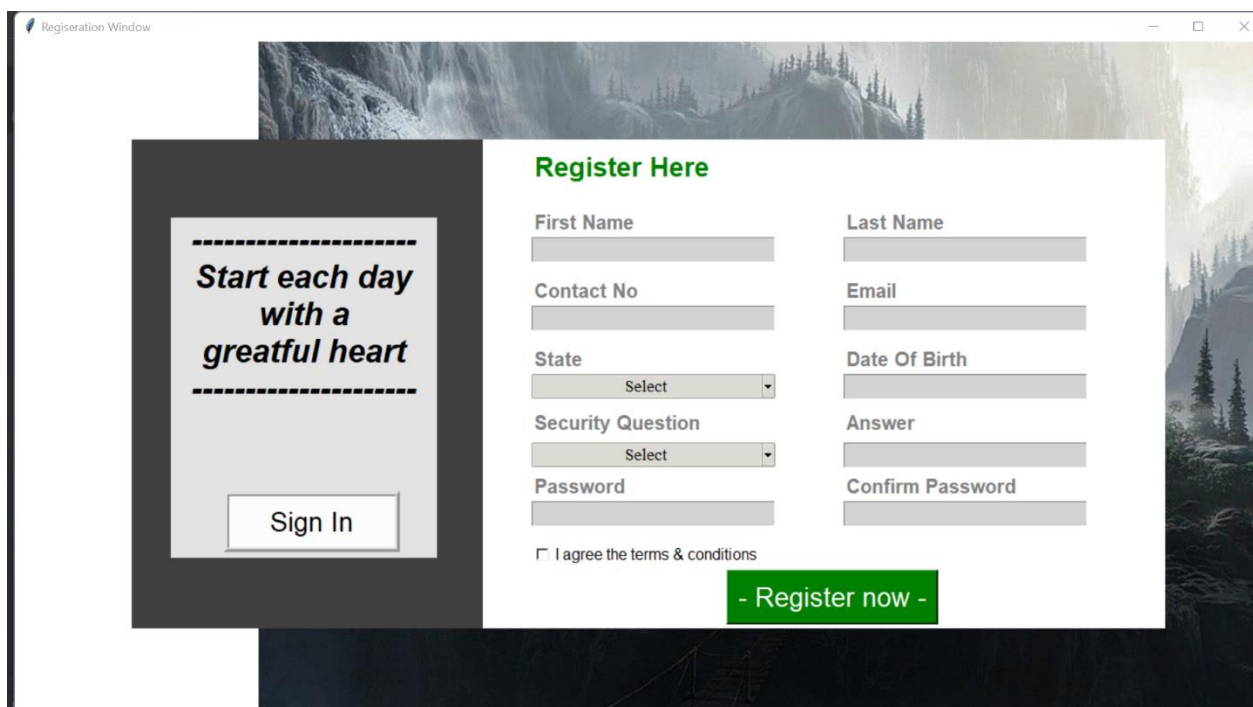


Fig6.8 Patient Registration Page

Appointment Area

Sun Shine Speciality Hospital

Appointment Area

[Book Appointment](#)
[Doctor availability](#)
[Hospital View](#)
[Developed By](#)
[Logut](#)
[Back](#)

Patient Information

Patient Name
Sunny

Gender
Male

Age
20

Contact Number
9366634545

Address
Darmavaram

Symptoms
Fever

Enter Appointment Date
26/3/22

Select To Appoint Doctor
Dr.Srujs

Over view

Patient Name:
Gender:
Age:
Contact Number:
Address:
Symptoms:
Appointment Date:
Appointed Doctor:

Sunny
Male
20
9366634545
Darmavaram
Fever
26/3/22
Dr.Srujs

Register

Over-View

Reset

Patient Name	Gender	Age	Contact Number	Address	Symptoms	Appointment Date	Appointed Doctor
Gowtham	Male	20	6363	ATP	fever	26/3/22	Dr.Srujs
Sunny	Male	20	6363	dmw	fever	26/3/22	Dr.Srujs
Sunny	Male	20	9366634545	Darmavaram	Fever	26/3/22	Dr.Srujs

Fig6.9 Book Apooointment Page

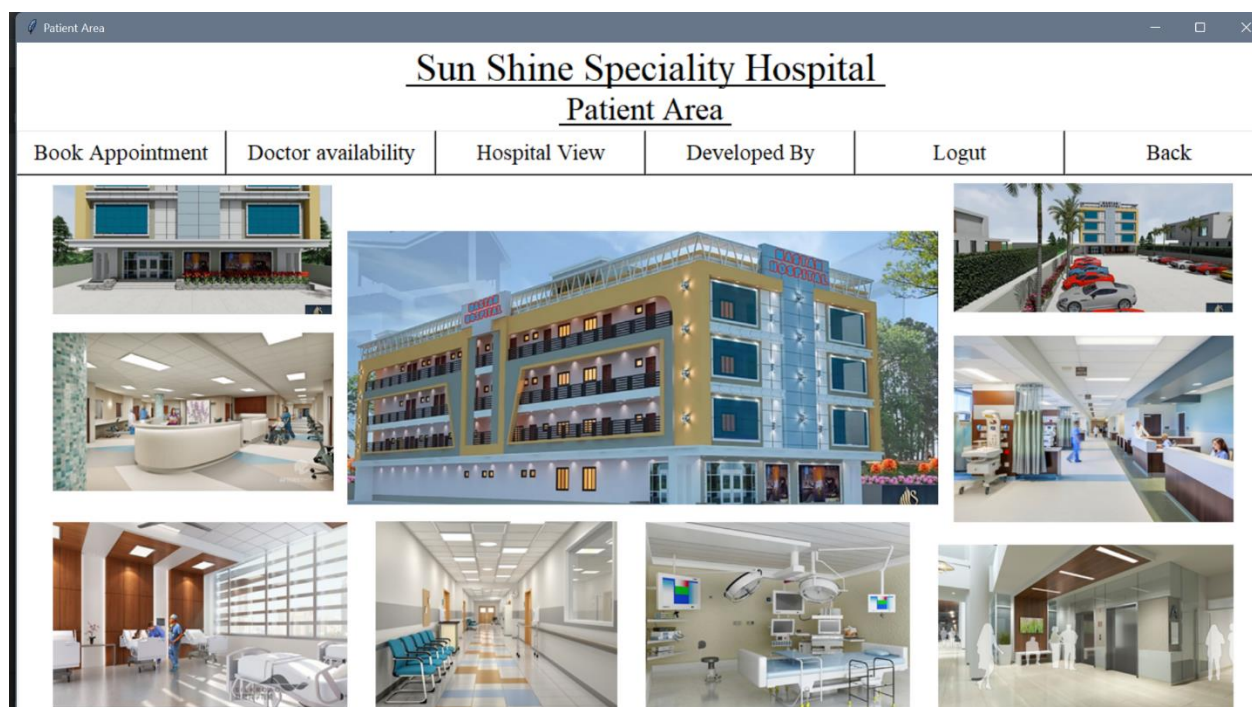


Fig6.10 Hospital View

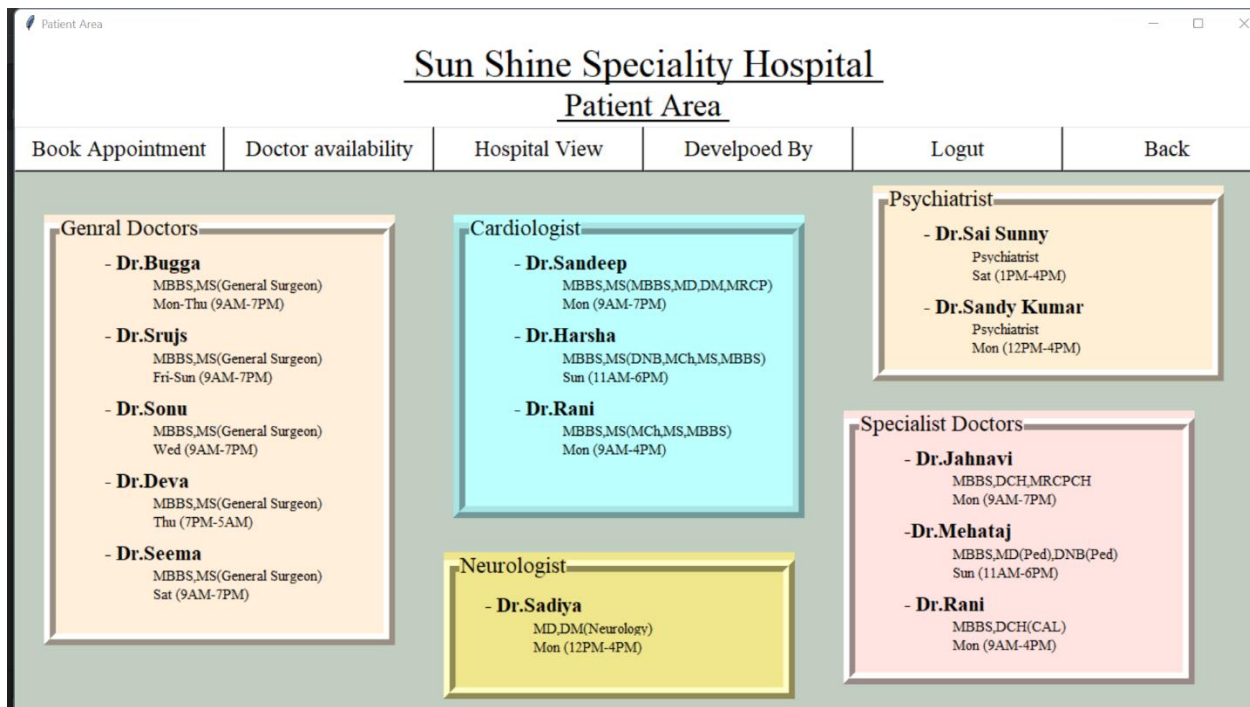


Fig6.11 The Available Doctors

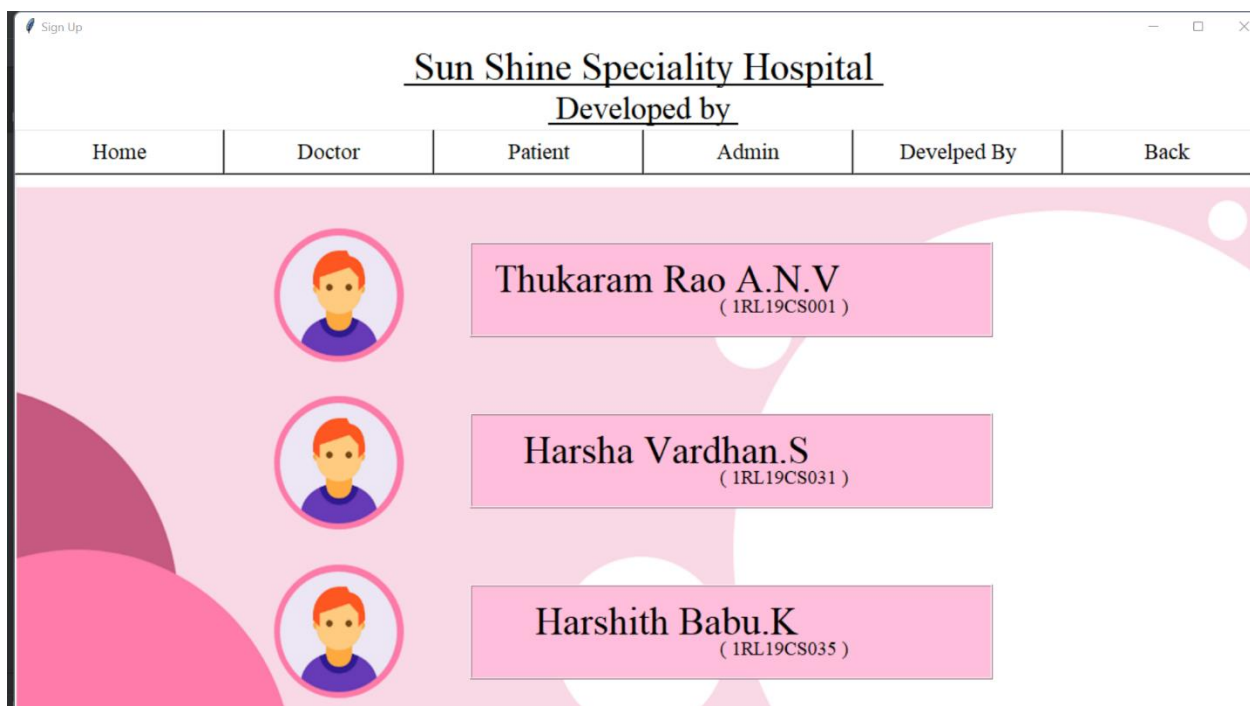


Fig6.12 Developed By

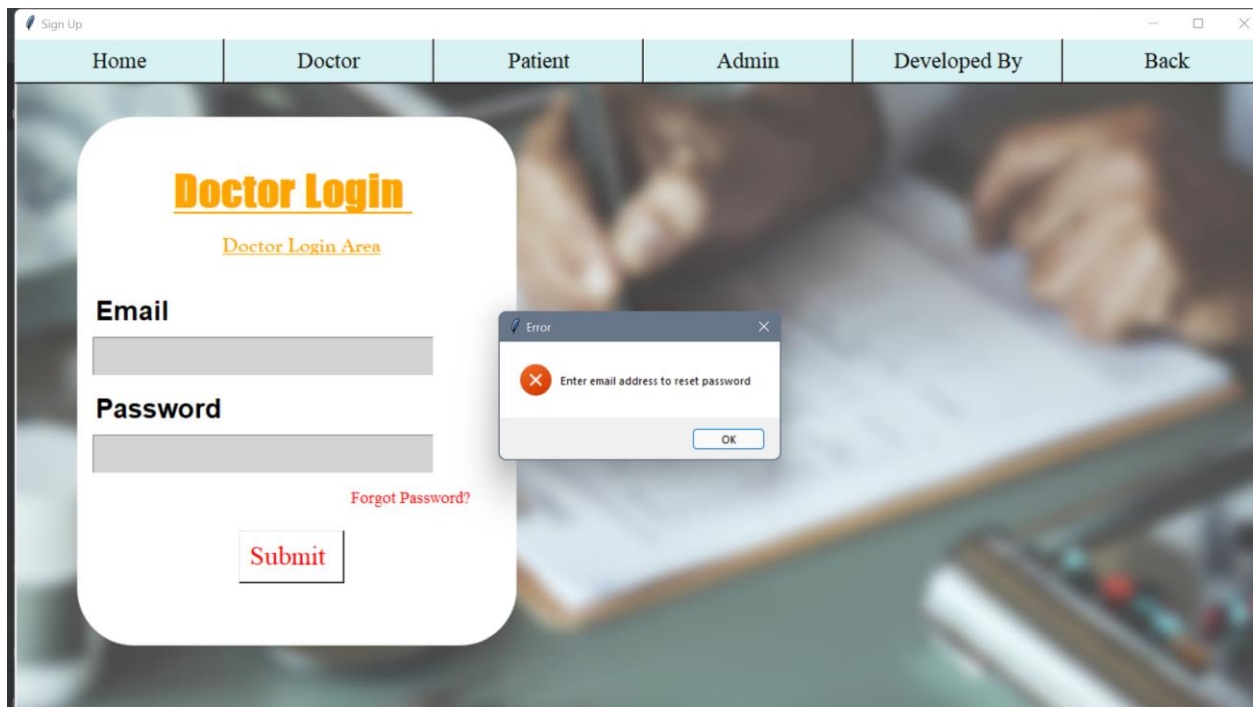


Fig6.13 Validation Error box

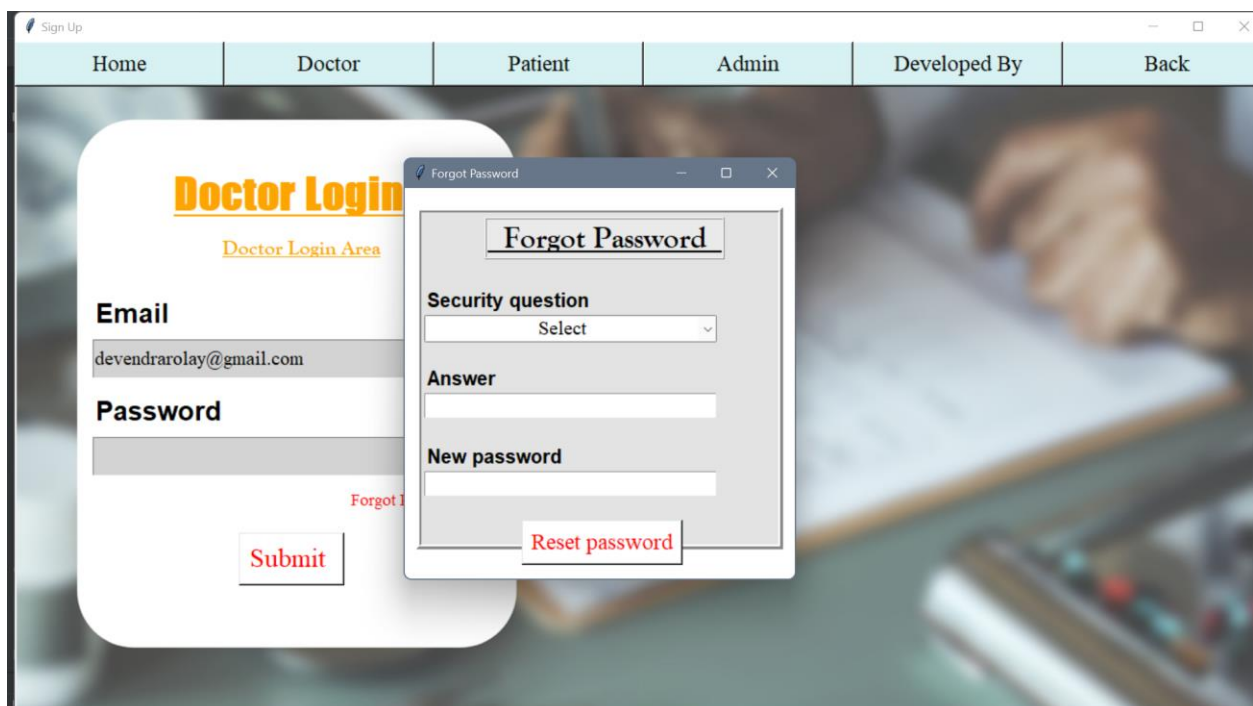


Fig6.14 Forgot Password

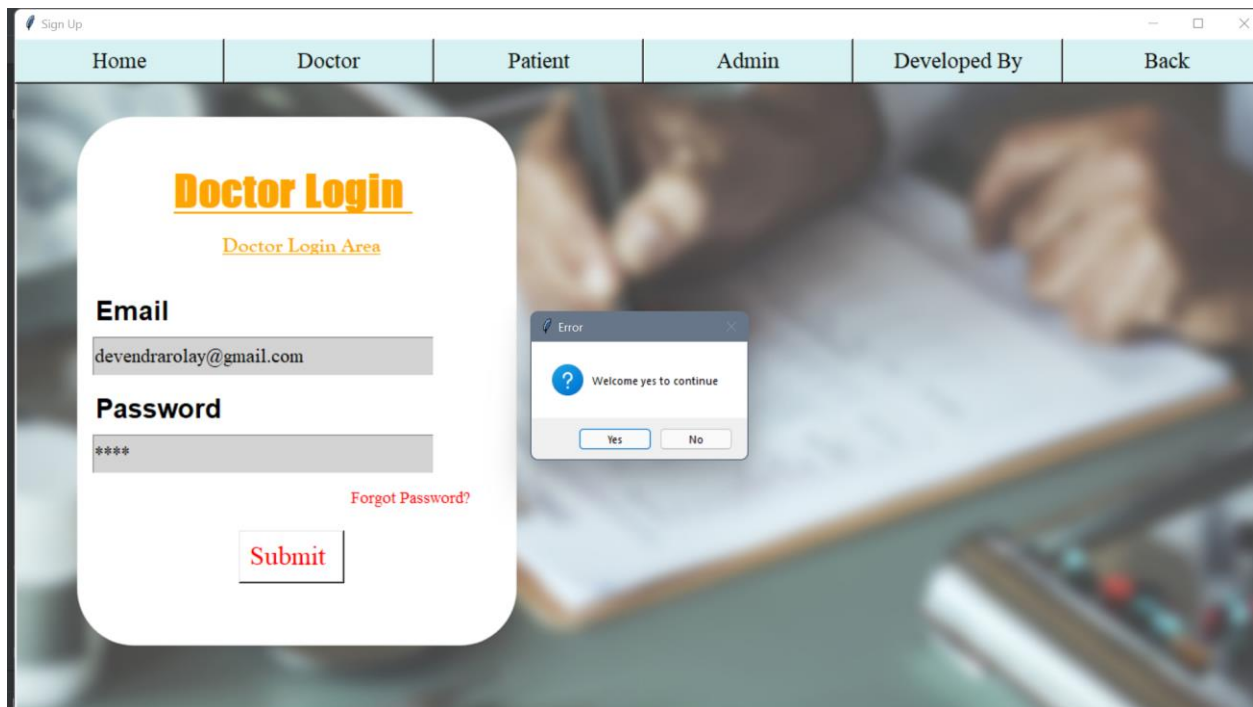


Fig6.15 Welcome Message Box

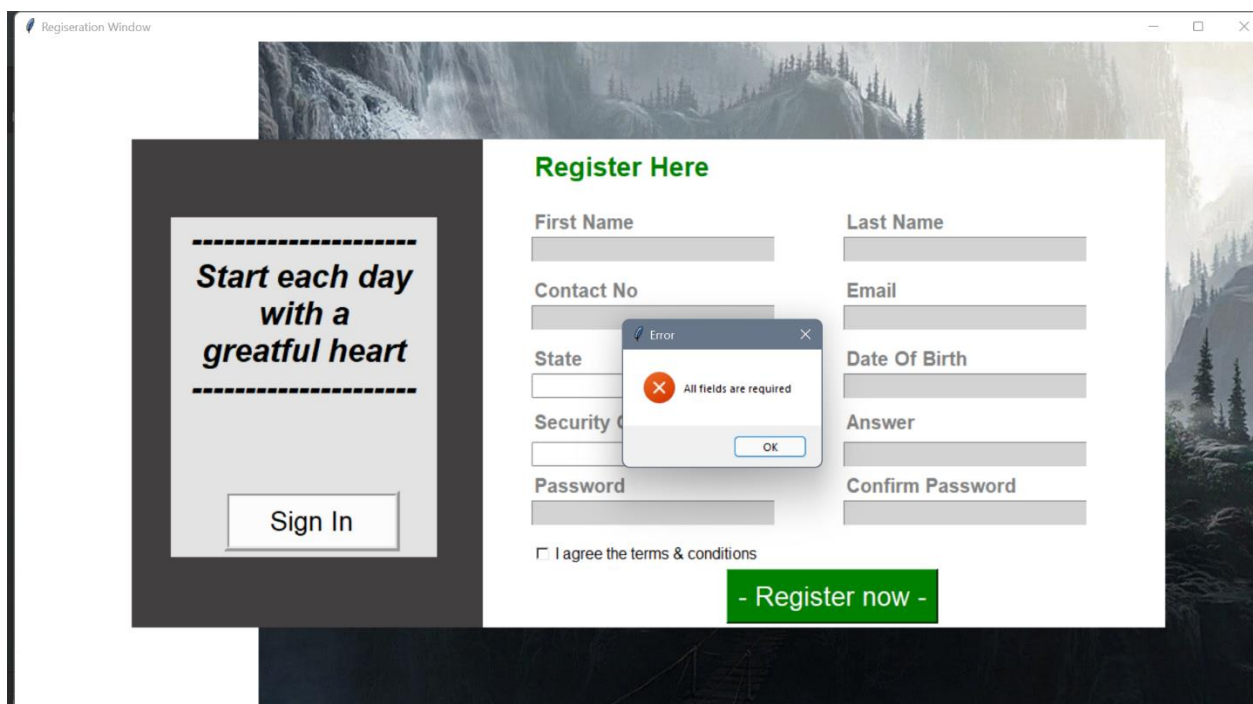


Fig6.16 All Field Required Message Box

CHAPTER 7

CONCLUSION

7.1 Conclusions

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas.

- a) We have gained an insight into the working of the HOSPITAL. This represents a typical real world situation.
- b) Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed.
- c) Scheduling a project and adhering to that schedule creates a strong sense of time management.
- d) Sense of teamwork has developed and confidence of handling real life project has increased to a great extent.
- e) Initially, there were problems with the validation but with discussions, we were able to implement validations.

7.2 Limitations of the system

- Online payment is not available at this version.
- Data delete & edit system is not available for all sections.
- User account not verified by Mobile SMS not available in this system.

REFERENCES

REFERENCES:

[1] DEEPAK THOMAS "BEGINNING PHP 4 DATABASES", WROX PRESS LTD. PAPERBACK- 17, OCTOBER, 2002. 70-130 pp.

[2] MATT DOYLE, "BEGINNING PHP 5.3, 2ND EDITION", OCTOBER 2009. 150-270 pp. [3] LUKE WELLING, LAURA THOMSON. SAMS

[3] PHP AND MYSQL WEB DEVELOPMENT, 2ND EDITION, PAPERBACK- 20 FEBRUARY, 2003. 105-209 pp.

[4] W. JASON GILMORE "BEGINNING PHP 5 AND MYSQL 5 FROM NOVICE TO PROFESSIONAL SECOND EDITION", JUL 9, 2008. 100-150 pp.

[5] ABRAHAM SILBER SCHATZ, HENRY F. KORTH AND S. SUDARSHAN "SIXTH EDITION DATABASE SYSTEM CONCEPTS RELEASED", JANUARY 28, 2010. 206-253 pp.

[6] SERVER-SIDE SCRIPTING [HTTP://PHP.NET/MANUAL/EN/INDEX.PHP](http://php.net/manual/en/index.php),

[7] [HTTPS://WWW.W3SCHOOLS.COM/](https://www.w3schools.com/),

[8] BOOTSTRAP [HTTP://GETBOOTSTRAP.COM/](http://getbootstrap.com/),

[9] [HTTPS://STACKOVERFLOW.COM/](https://stackoverflow.com/),