BASAVARAJESWARI GROUP OF INSTITUTIONS

Ballari Institute of Technology & Management

AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA, BELAGAVI-590018

INTERNSHIP

Report On

HEALTHCARE APPOINTMENT BOOKING

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

TASNEEM BANU

3BR23CS170

Internship Carried OutBy

EZ TRAININGS & TECHNOLOGIES PVT.LTD
HYDERABAD

Internal Guide

S.STEFFI NIVEDITA

Asst.prof,CSE

VARADA ALEKHYA

Asst. prof,CSE

External Guide

Bhavna Vaishnav

Technical Trainer

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

2024-2025

BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

Autonomous institute under VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA,

BELAGAVI 590018

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"JnanaGangotri"Campus,No.873/2,Ballari-HospetRoad,Allipur, Ballar1-583 104 (Karnataka)(India)

Ph: 08392 – 237100 / 237190, Fax: 08392 –237197

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Internship entitled **"HEALTHCARE APPOINTMENT BOOKING"** hasbeen successfully completed by **TASNEEM BANU** bearing USN **3BR23CS170** a bonafide student of Ballari Institute of Technology and Management, Ballari. For the partial fulfillment of the requirements for the **Bachelor's Degree in Computer Science and Engineering** of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi during the academic year 2024-2025.

Signature of Internship Co-ordinator

S.STEFFI NIVEDITA
Asst.prof,CSE

VARADA ALEKHYA
Asst. prof,CSE

Signature of HOD

R N KULKARNI
Prof. and HOD(CSE)

DECLARATION

I, **TASNEEM BANU** second year student of Computer Science and Engineering, Ballari Institute of Technology, Ballari, declare that Internship entitled **HEALTHCARE APPOINMENT BOOKING** is a part of Internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT.LTD**

,Hyderabad at "**BITM,BALLARI".** This report is submitted in partial fulfillment of the requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi.

Date : **Signature of the Student**

Place : BALLARI

ACKNOWLEDGEMENT

The satisfactions that a company the successful completion of my internship on "HEALTHCARE APPOINTMENT BOOKING" would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to our respective coordinator **"S.Steffi Nivedita (Asst.prof,CSE) , Varada Alekya (Asst.prof,CSE)"** for his noble gesture, support co-ordination and valuable suggestions given to me in the completion of Internship.

I also thank **R N Kulkarni,** H.O.D. Department of **Computer science and engineering** for extending all his valuable support and encouragement.

Table of Contents

CHAPTER-2

COMPANY PROFILE

Company Name: EZ Trainings and Technologies Pvt. Ltd.

Introduction:

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

Mission:

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

Services:

College Trainings:

- Tailored training programs designed to enhance the employability of students.
- Industry-aligned curriculum covering technical and soft skills.
- Placement assistance and career guidance.

Development Projects:

- End-to-end development services, from ideation to execution.
- Expertise in diverse technologies and frameworks.
- Custom solutions to meet specific business needs.

Locations: Hyderabad | Delhi NCR

At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence

CHAPTER-3

Basavarajeshwari Group of Institutions
BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT
Autonomous Institute under VTU - Belagavi
"Jnana Gangotri" Campus, Bellary-Hospet Road, Near Allipura Village,
BALLARI - 583 104 (Karnataka)
Ph: 08392-237167/237153 Fax: 237197, e-mail: bitmbly@gmail.com
Website: www.bitm.edu.in

Internship Program on Python for BE-3rd Sem students
From 9th to 28th September 2024 (During 3rd semester vacations).

Student Name: TASNEEM BANU USN No: 3BR23CS170 Branch: CSE

INDEX PAGE

Day	Date	Content Covered		Signature of the faculty in-charge
1	09.09.24	Introduction to python (data types, variables)	Keywords, comments indentation, operators	
2	10.09.24	Conditional statements looping	Problem solving using loops, pattern program	
3	11.09.24	Functions, strings operat--ions problem solving	Recursion, class & object constructor	
4	12.09.24	List, tuple	Set,dictionary(problem using list and tuple)	
5	13.09.24	Oops(Inheritance, polymorphism)	Encapsulation, Abstraction	
6	14.09.24	Exceptional handling File handling	Problem solving	
7	15.09.24	Problem solving 15Q	Problem solving 10Q	
8	16.09.24	Revision and scenerios problem solving	Revision and scenerios problem solving	
9	18.09.24	Modules and packages	(Oop's revision)	
10	19.09.24	Introduction to DS, O (time complexity), Class & pointers	Linked list(get, set insert, remove)	
11	20.09.24	Linked list, stack (pop, isfull, isempty, Peck, display, push)	Queue(constructor, enqueue, dequeue) priority queue)	
12	21.09.24	Bubble sort, Selection sort, Insertion sort	Merge sort and task in sorting	
13	23.09.24	Recursion, quick sort 5Q	Linear search &proble -m solving(Assessment)	
14	24.09.24	Binary search, Introd--uction to trees, tree Traversal	Traversal, Insertion searching(BFS, DFS)	
15	25.09.24	Introduction to graph (BFS, DFS), Introduct -ion to graph algorithm	Project team separati -on &discussion of problem statement with each team	

16	26.09.24	Project implementation Project implementatio -n &problem solving in Reinprep portal	
17	27.09.24	Project evaluation Phase-01	
18	28.09.24		

HEALTHCARE APPOINTMENT BOOKING

3.Abstract :

The Appointment Management System is a streamlined application designed to facilitate scheduling and managing appointments between healthcare providers and patients. Built using Python, the program allows users to input appointment details, check a doctor's availability, and manage patient records efficiently.

Key features include confirmation and deletion of appointments, as well as sending reminders to both doctors and patients, enhancing communication and reducing no-show rates.

The system employs a simple command-line interface that guides users through input collection and displays essential appointment details. By organizing appointments effectively, the application aims to improve time management for healthcare providers and enhance the overall patient experience.

The modular design of the codebase allows for future enhancements, such as database integration, mobile application development, and advanced notification systems, making it adaptable to evolving healthcare needs. This program serves as a foundational tool in the quest for more efficient healthcare delivery.

4.Introduction of the project :

This program serves as a simple appointment management system for doctors and patients, enabling efficient scheduling and reminders. Users can easily input appointment details, check doctor availability, and manage patient records for a seamless healthcare experience.

The program is a simple appointment management system designed for doctors and patients. It allows doctors to manage their schedules, check appointment availability, and send reminders to both doctors and patients.

Key points in the project :

1) Class Structure : The program utilizes object-oriented programming principles with classes representing Admin, Doctor, and Patient. The Doctor and Patient classes inherit properties from the Admin class.

2) Appointment Management :

- Doctors can check their availability for a given appointment time, confirm appointments, and delete them if necessary.
- Patients can be added to the system, and their appointment details are stored.

3) User Interaction : The program prompts the user to enter appointment details, patient information, and choose options for managing appointments through a text-based interface.

4) Reminders : The system can send reminders to both the doctor and the patients about their scheduled appointments.

Overall, this program provides a foundational structure for managing medical appointments

5.Description :

Code implementation :

```
from datetime import datetime
class Admin:
    def _init_(self, name, phone, email):
        self.name = name
        self.phone = phone
        self.email = email

class Doctor(Admin):
    def _init_(self, name, phone, email):
         self.name = name
         self.phone = phone
         self.email = email
         self.appointments = []
    def availability(self, appointment_time):
        for appointment in self.appointments:
            if appointment == appointment_time:
                return "Doctor unavailable"
        return "Doctor is Available”
    def send_reminder(self, appointment):
        print(f"Reminder sent to Dr. {self.name} for appointment        on
{appointment.strftime('%Y-%m-%d %H:%M')}.")

class Patient(Admin):
    def _init_(self, name, phone, email):
         self.name = name
```

```
        self.phone = phone
        self.email = email
    def send_reminder(self, appointment):
        print(f"Reminder sent to {self.name} for appointment on
{appointment.strftime('%Y-%m-%d %H:%M')}.")
appointment = input("Enter appointment date (YYYY-MM-DD HH:MM): ")
appointment_datetime = datetime.strptime(appointment, "%Y-%m-%d
%H:%M")
doctor = Doctor("Dr. XYZ", "123-456-7890", "dr.xyz@example.com")
patients = []

while True:
    name = input("Enter patient name (leave empty to finish): ")
    if not name:
        break
    phone = input("Enter patient phone number: ")
    email = input("Enter patient email: ")

    patient = Patient(name, phone, email)
    patients.append(patient)
def display(doctor, patients, appointment):
    print("APPOINTMENT DETAILS:\n")
    print("Doctor Details:")
    print(f"Doctor name: {doctor.name}\nPhone number:
{doctor.phone}\nEmail: {doctor.email}")
    print(f"\nSCHEDULED PATIENTS FOR - {appointment.strftime('%Y-%m-
%d %H:%M')}\n")
    for patient in patients:
```

```
        print(f"Patient name: {patient.name}\nPhone number:
{patient.phone}\nEmail: {patient.email}\n")
current_time = datetime.now()
print(f"CURRENT TIME: {current_time}")
print(f"\nSELECTED APPOINTMENT DATE: {appointment}\n")
print(doctor.availability(appointment_datetime), "\n")
display(doctor, patients, appointment_datetime)
while True:
    print("\nSelect an option:\n1. Confirm appointments\n2. Delete
appointment\n3. Send reminders\n")
    option = int(input())
    if option == 1:
        doctor.appointments.append(appointment_datetime)  # Adding
appointment to doctor's list
        print("\nAPPOINTMENTS CONFIRMED\n")
        for patient in patients:
            print(f"Appointment added for {patient.name} with {doctor.name} at
{appointment_datetime}\n")
    elif option == 2:
        if appointment_datetime in doctor.appointments:
            doctor.appointments.remove(appointment_datetime)  # Removing
appointment from doctor's list
            print("\nAPPOINTMENT DELETED\n")
    elif option == 3:
         for patient in patients:
            patient.send_reminder(appointment_datetime)
         doctor.send_reminder(appointment_datetime)
    else:
        break
```

6.Algorithm :

Algorithm for Appointment Management System

- Define Classes:

Create a base class Admin with attributes name, phone, and email.

Define a class Doctor that inherits from Admin, including methods for checking availability and sending reminders.

Define a class Patient that inherits from Admin, with a method to send reminders.

- Get Appointment Input:

Prompt the user to enter an appointment date and time in the format "YYYY-MM-DD HH".

Validate the input format and convert it to a datetime object.

- Collect Patient Information:

Initialize an empty list for patients.

Continuously prompt the user to enter patient details (name, phone, email) until the user leaves the name empty.

Create a Patient object for each entry and append it to the patient list.

- Display Appointment Details:

Print the current time and the selected appointment time.

Check and display the doctor's availability for the appointment time.

Print the details of the doctor and the scheduled patients.

Manage Appointments:

Present a menu of options:

Confirm Appointments:

Add the appointment to the doctor's list and confirm to the user.

Delete Appointment:

Check if the appointment exists in the doctor's list. If so, remove it and notify the user.

- Send Reminders:

Loop through each patient and send a reminder for the appointment.

Send a reminder to the doctor as well.

Exit:

Exit the program.

- Handle Invalid Inputs:

For each menu option, handle invalid selections by prompting the user to try again.

7.Output：

Enter appointment date (YYYY-MM-DD HH:MM):

2024-09-26 09:10

Enter patient name (leave empty to finish): ANKUSH

Enter patient phone number: 123-456-7890

Enter patient email: ankush@gmail.com

Enter patient name (leave empty to finish): DIA

Enter patient phone number: 987-654-3210

Enter patient email: dia@gmail.con

Enter patient name (leave empty to finish):

CURRENT TIME: 2024-09-27 12:04:43.341582

SELECTED APPOINTMENT DATE: 2024-09-26 09:10

Doctor is Available

APPOINTMENT DETAILS:

Doctor Details:

Doctor name: Dr. XYZ

Phone number: 123-456-7890

Email: dr.xyz@example.com

SCHEDULED PATIENTS FOR - 2024-09-26 09:10

Patient name: ANKUSH

Phone number: 123-456-7890

Email: ankush@gmail.com

Patient name: DIA

Phone number: 987-654-3210

Email: dia@gmail.con

Select an option:

1. Confirm appointments

2. Delete appointment

3. Send reminders

4.EXIT

1

APPOINTMENTS CONFIRMED

Appointment added for ANKUSH with Dr. XYZ at 2024-09-26 09:10:00

Appointment added for DIA with Dr. XYZ at 2024-09-26 09:10:00

Select an option:

1. Confirm appointments

2. Delete appointment

3. Send reminders

4.EXIT

3

Reminder sent to ANKUSH for appointment on 2024-09-26 09:10.

Reminder sent to DIA for appointment on 2024-09-26 09:10.

Reminder sent to Dr. Dr. XYZ for appointment on 2024-09-26 09:10.

Select an option:

1. Confirm appointments

2. Delete appointment

3. Send reminders

4.EXIT

2

APPOINTMENT DELETED

Select an option:

1. Confirm appointments

2. Delete appointment

3. Send reminders

4.EXIT

4

EXIT

8.Conclusion :

Conclusion

The appointment management system implemented in this code provides a straightforward solution for doctors and patients to manage their appointments efficiently. Key features include:

- User-Friendly Interface: Users can easily input appointment details and patient information through a simple command-line interface.
- Doctor Availability Check: The system checks if the doctor is available for the requested appointment time, helping to avoid scheduling conflicts.
- Appointment Management: Users can confirm or delete appointments and send reminders to both doctors and patients, enhancing communication and organization.
- Scalability: The code is structured to allow for future enhancements, such as integrating a database for persistent storage, adding user authentication, or developing a graphical user interface.

Overall, this system streamlines the appointment scheduling process, making it more efficient and effective for healthcare providers and their patients. Future improvements could further enhance its functionality and user experience, adapting to the evolving needs of the healthcare sector.

9.Reference :

Healthcare appointment booking program’s code is constructed from our group .

Information of the report has been referred from internet resources.

Hence the project’s success .