

# Kishkindha university

Ballari

## INTERNSHIP

### Report On PROJECT SCHEDULING ASSISTANT

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

### Bachelor of Engineering In

Computer Science and Engineering

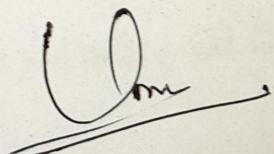
Submitted by

Gous

Internship Carried Out By  
EZ TRAININGS & TECHNOLOGIES PVT.LTD HYDERABAD

Internal Guide

External Guide





# Kishkindha university Ballari

DEPARTMENT OF Computer Science and  
ENGINEERING

## CERTIFICATE

This is to certify that the Internship entitled " PROJECT SCHEDULING ASSISTANT " has been successfully completed by Gous bearing USN  
bona fide students of Ballari Institute of Technology and Management, Ballari. For the partial fulfillment of the requirements for the Computer science and Engineering  
Engineering of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi during the academic year 2023-2024.

Signature of Internship

Co-ordinator

Signature of HOD

Ass.prof. Cse

Prof. and HOD of Of Cse

## DECLARATION

I, [REDACTED], second year student of Computer Science and Engineering Of Technology And Management, Ballari, declare that Internship entitled **PROJECT SCHEDULING ASSISTANT** is a part of internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT.LTD, HYDERABAD** at Kishkindha This report is submitted in partial fulfilment of requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering

Date : 28/09/2024  
Place : Ballari

  
Signature of the Student

## **ACKNOWLEDGEMENT**

The satisfactions that a company the successful completion of my internship on " PROJECT SCHEDULING ASSISTANT " 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 coordinators "Naga Pratyusha (Asst.prof,EEE)" for their noble gesture, support co-ordination and valuable suggestions given to us in the completion of Internship.

I also thank Sharan Reddy, H.O.D. Department of Electrical and Electronics Engineering for extending all her valuable support and encouragement.



# KISHKINDA UNIVERSITY

ADVANCING KNOWLEDGE TRANSFORMING LIVES

(Established under the Karnataka State Act No. 20 of 2023)

Internship Program on Python for B.Tech-3<sup>rd</sup> Sem students  
From 9<sup>th</sup> to 28<sup>th</sup> September 2024 (During 3<sup>rd</sup> semester vacations).

Student Name: Gous

USN No:

Branch: B.Tech-CSE 033

## INDEX PAGE

Day	Date	Content Covered	Signature of the faculty in-charge
1	09.09.24	Overview of Python-IO Statements	<u>J</u>
2	10.09.24	Operators basic Problem Solving	<u>J</u>
3	11.09.24	Conditional & Looping Statements	<u>J</u> <u>m</u>
4	12.09.24	List, Tuple with Problem Solving	<u>J</u>
5	13.09.24	Set & Dictionary with Problem Solving	
6	14.09.24	Overview of Strings	
7	15.09.24	Strings with Problem Solving	<u>J</u>
8	16.09.24	DSA Overview -Stack & Queue (List & Linked Model)	<u>J</u>
9	18.09.24	Linked List-Type Single & Circular	<u>J</u> <u>m</u>
10	19.09.24	Linked List-Types Double & Double Circular	<u>J</u>
11	20.09.24	Binary Tree with Traversal	<u>J</u>
12	21.09.24	Binary Search Tree with Traversal	
13	23.09.24	Graph -Build Matrix & Adj. List Model	
14	24.09.24	Graph -BFS, DFS	<u>J</u>
15	25.09.24	Project Overview -Submit Project Title	
16	26.09.24	Code Development	
17	27.09.24	Report & PPT Development	<u>J</u> <u>m</u>
18	28.09.24	Project review-PPT Presentation for Each team	



# KISHKINDA UNIVERSITY

5W38+WVG,Siruguppa Rd,Ashok Nagar,Rajeshwari Nagar, Bellari, Karnataka 5833275.

## Department of Computer Science & Engineering

### Project Presentation on “Consulate Booking System”

Presented by:-

Mohammad Azhar hussain  
Srirama D  
Gous  
Tameer Hussain  
Mohammed Tanveer P  
Mohammed Haseeb

# Consultate Booking System

APPOINTMENT MANAGEMENT WITH MYSQL INTEGRATION

# Introduction

## Overview of the System

- **Purpose:** Manage appointment bookings for consultate services.
- **Key Features:** Booking, cancelling, viewing appointments, and MySQL database integration.

# Class Structure

## Main Classes

- ▶ **Appointment:** Attributes: name, service, date, time
- ▶ **ConsulateBookingSystem:** Manages appointments (book, cancel, view)
- ▶ **AppointmentAdmin:** Manages MySQL database interactions.

# Booking, Cancelling, viewing Appointments methods

- **Booking Appointment**  
**Method:** book\_appointment(name, service, date, time)
- **Cancelling Appointments.**  
**Method:** cancel\_appointment(appointment\_id )by using appointment ID.
- **Viewing Appointments**  
**Method:** view\_appointments() Displays all scheduled appointments.
- **Adding Appointment Holders**  
**Method:** add\_appointment\_holder(appointment) Inserts appointment details into the database.

- **Details()**  
**Method:** details() used to displays all records from the database
- Database Connection Connect Details Parameters:  
host, user, password, database. Error handling for connection issues.
- Unit Testing Test Appointment Service System  
**Class** Uses unittest framework. Tests for adding appointments and fetching details.

“Thank you kindly”

```

import mysql.connector           import mysql.connector
from mysql.connector import Error
import unittest

class Appointment:
    def __init__(self, name, service, date, time):
        self.name = name
        self.service = service
        self.date = date
        self.time = time

    def __str__(self):
        return f"Appointment for {self.name}: {self.service} on {self.date} at {self.time}"

class ConsulateBookingSystem:
    def __init__(self):
        self.appointments = {}

    def book_appointment(self, name, service, date, time):
        appointment_id = len(self.appointments) + 1
        appointment = Appointment(name, service, date, time)
        self.appointments[appointment_id] = appointment
        print(f"Appointment booked successfully! ID: {appointment_id}")

    def cancel_appointment(self, appointment_id):
        if appointment_id in self.appointments:
            del self.appointments[appointment_id]
            print("Appointment canceled successfully!")
        else:
            print("Invalid appointment ID!")

    def view_appointments(self):
        if not self.appointments:
            print("No appointments scheduled.")
            return
        for appointment_id, appointment in self.appointments.items():
            print(f"ID: {appointment_id}, {appointment}")

class AppointmentAdmin:
    def __init__(self, host, user, password, database):
        try:
            self.conn = mysql.connector.connect(
                host=host,
                user=user,
                password=password,
                database=database
            )
            if self.conn.is_connected():
                print("Connected to MySQL Database")
                self.cursor = self.conn.cursor()
            else:
                print("Connection to MySQL failed.")
                self.cursor = None
        except Error as e:
            print(f"Error connecting to MySQL: {e}")
            self.conn = None
            self.cursor = None

    def add_appointment_holder(self, appointment):
        if self.cursor is None:
            print("Cannot add Appointment Holder Details. No database connection.")
            return

```

```

import mysql.connector
try:
    query = "INSERT INTO service (name, service, date, time) VALUES (%s, %s, %s, %s)"
    values = (appointment.name, appointment.service, appointment.date, appointment.time)
    self.cursor.execute(query, values)
    self.conn.commit()
    print(f"Account record for {appointment.name} added.")
except Error as e:
    print(f"Failed to insert record: {e}")

def details(self):
    if self.cursor is None:
        print("Cannot fetch Appointment Holder Details. No database connection.")
        return
    try:
        query = "SELECT * FROM service"
        self.cursor.execute(query)
        results = self.cursor.fetchall()
        print('----**Details**----')
        for row in results:
            print(row)
    except Error as e:
        print(f'Failed to fetch details: {e}')

def main():
    system = ConsulateBookingSystem()
    admin = AppointmentAdmin("localhost", "root", "user", "passport_renewal")

    while True:
        print("\nConsulate Services Booking System")
        print("1. Book Appointment")
        print("2. Cancel Appointment")
        print("3. View Appointments")
        print("4. Details")
        print("5. Exit")

        choice = input("Choose an option: ")

        if choice == '1':
            name = input("Enter your name: ")
            service = input("Enter the service required (e.g., Visa, Passport): ")
            date = input("Enter the date (YYYY-MM-DD): ")
            time = input("Enter the time (HH:MM): ")
            system.book_appointment(name, service, date, time)
            appointment = Appointment(name, service, date, time)
            admin.add_appointment_holder(appointment)

        elif choice == '2':
            appointment_id = int(input("Enter appointment ID to cancel: "))
            system.cancel_appointment(appointment_id)

        elif choice == '3':
            system.view_appointments()

        elif choice == '4':
            admin.details() # call details from admin instance

        elif choice == '5':
            print("Exiting the system.")
            break

```

```
        import mysql.connector
else:
    print("Invalid choice, please try again.")

class TestAppointmentServicesystem(unittest.TestCase):
    def setUp(self):
        self.admin = AppointmentAdmin("localhost", "root", "user",
"passport_renewal")
        self.appointment = Appointment('John Doe', 'Passport', '12-12-2022',
'12:30')

    def test_add_appointment(self):
        self.admin.add_appointment_holder(self.appointment)

    def test_view_details(self):
        self.admin.details()

if __name__ == "__main__":
    main()
```