KISHKINDA UNIVERSITY

ADVANCING KNOWLEDGE TRANSFORMING LIVES

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

Department of Computer Science Engineering

Certificate

This is to certify that Mr./Ms. SHRUSHTI ASHOK BUDIHAL bearing USN: KUB23CSE135 has completed Internship entitled "**PYTHON**" during **9th** to **28th September 2024** for the partial fulfilment of requirements for the award of bachelor's degree in Kishkinda University, Ballari.

Signature of Internal Guide

Signature of External Guide

Signature of HOD

Signature of Dean

KISHKINDA UNIVERSITY

ADVANCING KNOWLEDGE TRANSFORMING LIVES

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

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

Student Name: **USN No:** **Branch: B.Tech-CSE**

INDEX PAGE

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

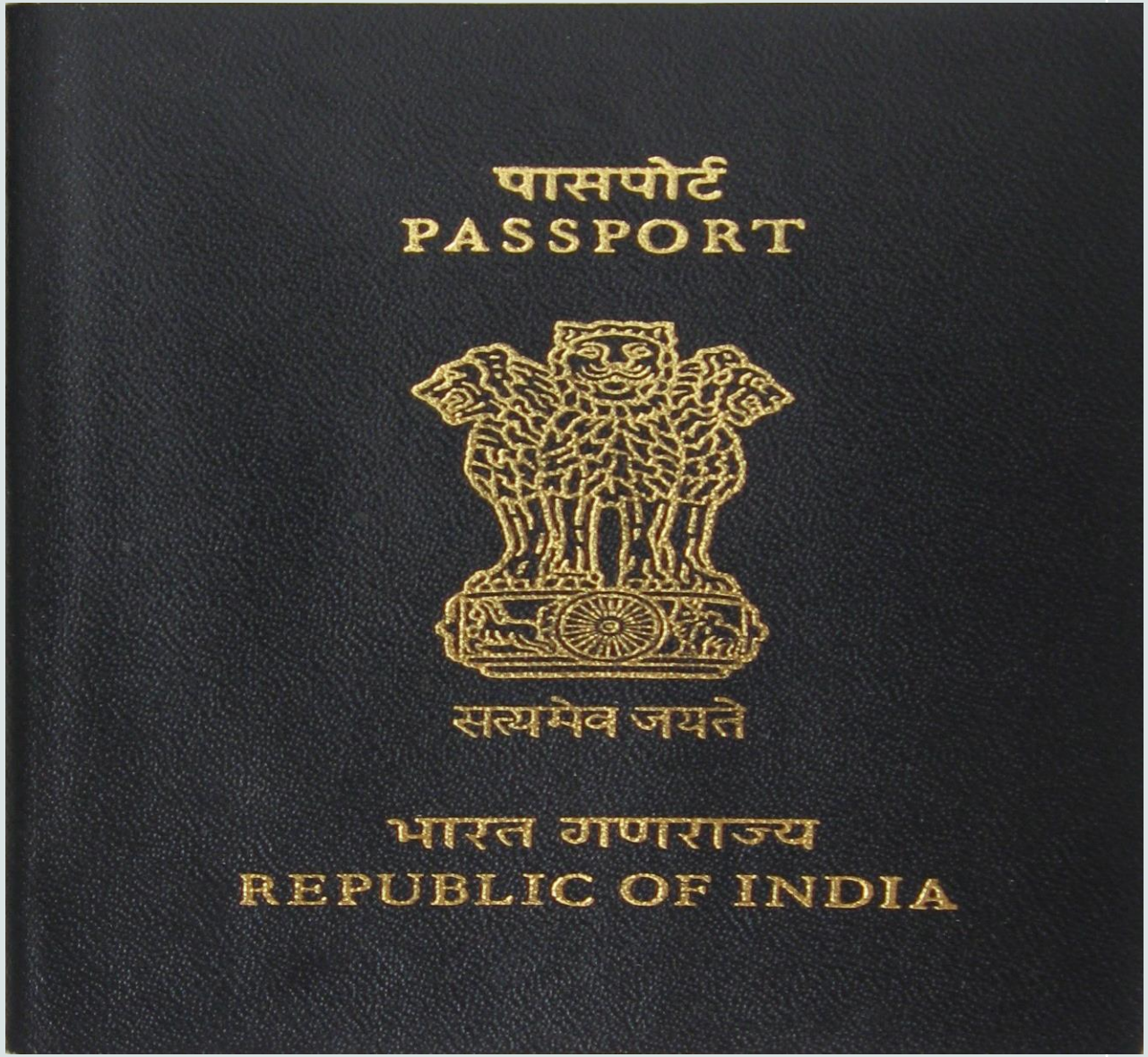
PASSPORT PRINTING QUEUE SYSTEM

KISHKINDA UNIVERSITY

Department of Computer Science and Engineering

TEAM MEMBERS

SAHANA.K.B - KUB23CSE120
SHRAVANI.V - KUB23CSE134
SHRUSHTI.A.B - KUB23CSE135
SINDHU.K.S - KUB23CSE136
SRUSHTI.K.H - KUB23CSE142



INTRODUCTION

The passport printing queue system POC aims to design and implement a
Efficient system for managing passport
Printing jobs. The system should allow for creating,reading,updating,and deleting printing jobs,managing the queue,and prioritizing urgent jobs.

CRUD Operation

CREATE

- Insert new data into tables.
- Assign unique identifiers (primary keys).
- Establish relationships between tables(foreign keys).

READ

- Query the database.
- Retrieve specific records or fields.
- Display data.

UPDATE

- Locate the record to update .
- Modify field values.
- Save changes.

DELETE

- Locate the record to delete
- Remove relationships
- Delete the record

Conclusion

The Passport printing queue system efficiently manages passport printing applications, ensuring streamlined processing security and accuracy its automated queue management priority scheduling and real-time status updates improve priority reduce processing time and enhance customer satisfaction

```
import mysql.connector
from mysql.connector import Error

# Establish a connection to MySQL
mydb = mysql.connector.connect(
    host='localhost',
    user='root',
    password='user'
)

mycursor = mydb.cursor()

class PassportDetails:
    def __init__(self, person_name, pid, urgent, idate, edate):
        self.person_name = person_name
        self.pid = pid
        self.urgent = urgent
        self.idate = idate
        self.edate = edate
        self.price = 200 if urgent else 100

    def __str__(self):
        return f'{self.person_name}, {self.pid}, {self.price}'

class PrintingQueue:
```

```
@staticmethod
    def createDB():
        try:
            mycursor.execute("CREATE DATABASE IF NOT EXISTS Passportprinting;")
            print("Database 'Passportprinting' created successfully or already exists.")
        except mysql.connector.Error as err:
            print(f'Error: {err}')

    @staticmethod
    def useDB():
        try:
            mycursor.execute("USE Passportprinting;")
            print("Using database 'Passportprinting'.")
        except mysql.connector.Error as err:
            print(f'Error: {err}')

    @staticmethod
    def createPassportTable():
        try:
            mycursor.execute("""
            CREATE TABLE IF NOT EXISTS PassportPrinting(
                name VARCHAR(20),
                pid INT(10) PRIMARY KEY,
                price INT(20),
                idate DATE,
```

```
edate DATE
            );
            """)
            print('Table "PassportPrinting" created successfully or already exists.')
        except mysql.connector.Error as err:
            print(f'Error: {err}')

    @staticmethod
    def insertPassport(details: PassportDetails):
        try:
            sql = "INSERT INTO PassportPrinting (name, pid, price, idate, edate)
VALUES (%s, %s, %s, %s, %s)"
            values = (details.person_name, details.pid, details.price, details.idate,
details.edate)
            mycursor.execute(sql, values)
            mydb.commit()
            print("Passport details inserted successfully.")
        except mysql.connector.Error as err:
            print(f'Error: {err}')

    @staticmethod
    def updatePassport(pid, new_details: PassportDetails):
        try:
```

```
sql = "UPDATE PassportPrinting SET name = %s, price = %s, idate = %s, edate = %s
WHERE pid = %s"
            values = (new_details.person_name, new_details.price, new_details.idate,
new_details.edate, pid)
            mycursor.execute(sql, values)
            mydb.commit()
            print("Passport details updated successfully.")
        except mysql.connector.Error as err:
            print(f'Error: {err}')

    @staticmethod
    def deletePassport(pid):
        try:
            sql = "DELETE FROM PassportPrinting WHERE name = %s"
            mycursor.execute(sql, (pid,))
            mydb.commit()
            print("Passport details deleted successfully.")
        except mysql.connector.Error as err:
            print(f'Error: {err}')

pq = PrintingQueue()
pq.createDB()
pq.useDB()
pq.createPassportTable()
```

```
passport = PassportDetails("John doe",
123456,False,"2020-01-01", "2025-01-10")
pq.insertPassport(passport)
p1= PassportDetails("sirisha",647884,True , "2020-
01-01", "2025-02-16")
pq.insertPassport(p1)
updated_passport = PassportDetails("John doe",
123456, False, "2020-02-01", "2025-02-10")
pq.updatePassport(123456, passport)
pq.deletePassport("shrushti")
mycursor.close()
mydb.close()
```

File Edit Selection View Go Run Terminal Help

Search

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python

```
PS C:\Users\User> & "C:/Program Files/Python312/python.exe" c:/Users/User/Downloads/KUB23CSE135/imp2.py
Database 'Passportprinting' created successfully or already exists.
Using database 'Passportprinting'.
Table "PassportPrinting" created successfully or already exists.
Error: 1062 (23000): Duplicate entry '123456' for key 'passportprinting.PRIMARY'
Error: 1062 (23000): Duplicate entry '647884' for key 'passportprinting.PRIMARY'
Passport details updated successfully.
Passport details deleted successfully.
PS C:\Users\User> mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 60
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| company            |
| information_schema |
| mysql              |
| passportprinting   |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

mysql> use passportprinting
Database changed
mysql> show tables;
+----------------------------+
| Tables_in_passportprinting |
+----------------------------+
| passportprinting           |
+----------------------------+
1 row in set (0.00 sec)

mysql> select *from passportprinting;
+----------+--------+-------+------------+------------+
| name     | pid    | price | idate      | edate      |
+----------+--------+-------+------------+------------+
| John doe | 123456 |   100 | 2020-01-01 | 2025-01-10 |
| sirisha  | 647884 |   200 | 2020-01-01 | 2025-02-16 |
+----------+--------+-------+------------+------------+
2 rows in set (0.00 sec)

mysql>
```

Ln 3, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit

33°C Partly sunny

Search

ENG IN 16:02 28-09-2024


THANK
you
THE FISCHERS