# PASSPORT\_PRINTING QUEUE SYSTEM

## KISHKINDA UNIVERSITY

Department of Computer Science and Engineering



#### **TEAM MEMBERS**

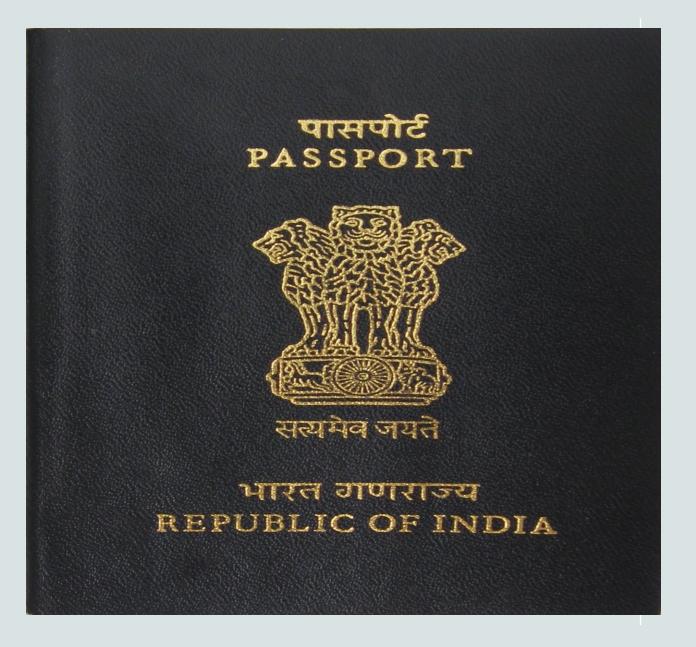
SAHANA.K.B - KUB23CSE120

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**

#### DELETE

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

- 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 PrintingOueue:

```
@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),
                                         SAMPLE FOOTER TEXT
                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()
```

