------------------HACKATHON---------------------

VU21CSEN0101734

GIDUTURI SAI SRI ANJALI

**QUESTION 4:**

Child Vaccination Management System

Young children are at increased risk for infectious diseases because their immune systems have not yet built up the necessary defences to fight serious infections and diseases. Making sure that children have access to proper healthcare and immunization against diseases that can be prevented by vaccines, is a huge challenge that is being faced by developing countries like ours. This highlights the importance and need of having a better, smarter system in place, to improve the situations. This application provides a system to provide information, store records and help parents schedule vaccination appointments for their children.

It's user-friendly.

The system helps to book appointments for their child's vaccination easily.

Parents can view their updates and also their reminders for appointments.

**Functionality**

The system offers several functionalities:

* **Registering Children:** Parents can register their children by providing their name, birthdate, parent's name, and contact information.
* **Viewing Vaccine Schedule:** Based on the child's age, the system displays a recommended vaccination schedule with recommended vaccines and ages for administration.
* **Scheduling Vaccinations:** Parents can schedule vaccination appointments for specific vaccines. The system automatically schedules the appointment for 7 days from the current date.
* **Viewing All Registered Children:** This option allows viewing a list of all children registered in the system.
* **Viewing Vaccination Records:** Parents can view a list of all vaccinations received by a specific child, including the vaccine name and date of administration.
* **Sending Reminders (Not fully implemented):** This feature is partially implemented and displays upcoming vaccination reminders based on today's date and scheduled appointments (within 1 day).

**Code Explanation**

The code is organized into classes:

* **Child:** Stores child information like name, birthdate, parent's name, contact, and vaccination records.
* **Vaccine:** Represents a vaccine with details like name, diseases prevented, and recommended age.
* **VaccinationSystem:** Manages the overall system, including registering children, scheduling vaccinations, viewing records, etc.

The main function provides a menu-driven interface for users to interact with the system. Users can choose from options like registering a child, viewing vaccine schedules, scheduling vaccinations, and viewing records.

**Benefits**

This system offers several benefits for parents and healthcare providers:

* **Improved Vaccination Tracking:** Parents can easily track their children's vaccinations and ensure they are on schedule.
* **Timely Reminders:** Reminders help parents avoid missing scheduled appointments.
* **Better Communication:** Parents have access to vaccination information easily.

**Future Enhancements**

Several features can be added to improve the system:

* **Integration with Database:** Store data persistently in a database like MySQL for better data management.
* **SMS/Email Reminders:** Implement automated SMS or email reminders for upcoming appointments.
* **Detailed Vaccine Information:** Include a database of vaccine information with details about each vaccine.
* **Security:** Implement user authentication and authorization for secure access to data.
* **Vaccine Eligibility Checking:** Integrate logic to check if a child is eligible for a specific vaccine based on factors like age and past vaccinations.

This Python program provides a foundation for a Child Vaccination Management System. By incorporating the suggested enhancements, the system can become a valuable tool for parents and healthcare providers to ensure children receive timely vaccinations and maintain healthy development.

Code:

from datetime import datetime, timedelta

# Child class to store details of each child

class Child:

def \_init\_(self, name, birthdate, parent\_name, contact):

self.name = name

self.birthdate = birthdate

self.parent\_name = parent\_name

self.contact = contact

self.vaccination\_records = []

def \_repr\_(self):

return f"\nChild(Name: {self.name}, Birthdate: {self.birthdate.strftime('%Y-%m-%d')}, Parent: {self.parent\_name}, Contact: {self.contact})"

# Vaccine class to store details of each vaccine

class Vaccine:

def \_init\_(self, name, diseases\_prevented, recommended\_age):

self.name = name

self.diseases\_prevented = diseases\_prevented

self.recommended\_age = recommended\_age

def \_repr\_(self):

return f"\nVaccine(Name: {self.name}, Diseases: {', '.join(self.diseases\_prevented)}, Recommended Age: {self.recommended\_age} months)"

# VaccinationSystem class to manage the entire system

class VaccinationSystem:

def \_init\_(self):

self.children = []

self.vaccines = self.load\_vaccines()

# Load predefined vaccines

def load\_vaccines(self):

return [

Vaccine("BCG", ["Tuberculosis"], 0),

Vaccine("Hepatitis B", ["Hepatitis B"], 0),

Vaccine("DTP", ["Diphtheria", "Tetanus", "Pertussis"], 2),

Vaccine("Polio", ["Polio"], 2),

Vaccine("Measles", ["Measles"], 9),

]

# Register a child in the system

def register\_child(self, name, birthdate, parent\_name, contact):

new\_child = Child(name, birthdate, parent\_name, contact)

self.children.append(new\_child)

print(f"\n{new\_child.name} has been successfully registered.")

# View vaccination schedule for a specific child

def view\_vaccine\_schedule(self, child\_name):

child = self.find\_child\_by\_name(child\_name)

if child:

age\_in\_months = self.calculate\_age\_in\_months(child.birthdate)

print(f"\nVaccination Schedule for {child.name} (Age: {age\_in\_months} months):")

for vaccine in self.vaccines:

if age\_in\_months >= vaccine.recommended\_age:

print(f" - {vaccine}")

else:

print(f"\nChild with name {child\_name} not found.")

# Schedule a vaccination appointment for a child

def schedule\_vaccination(self, child\_name, vaccine\_name):

child = self.find\_child\_by\_name(child\_name)

vaccine = self.find\_vaccine\_by\_name(vaccine\_name)

if child and vaccine:

appointment\_date = datetime.now() + timedelta(days=7)

child.vaccination\_records.append((vaccine, appointment\_date))

print(f"\nVaccination for {vaccine.name} scheduled on {appointment\_date.strftime('%Y-%m-%d')} for {child.name}.")

else:

print(f"\nError scheduling vaccination: child or vaccine not found.")

# View all children registered in the system

def view\_all\_children(self):

if self.children:

print("\n--- Registered Children ---")

for child in self.children:

print(child)

else:

print("\nNo children registered yet.")

# View all vaccination records for a specific child

def view\_vaccination\_records(self, child\_name):

child = self.find\_child\_by\_name(child\_name)

if child:

print(f"\nVaccination Records for {child.name}:")

for record in child.vaccination\_records:

vaccine, date = record

print(f" - {vaccine.name} on {date.strftime('%Y-%m-%d')}")

else:

print(f"\nChild with name {child\_name} not found.")

# Send reminders to parents for upcoming appointments

def send\_reminders(self):

print("\n--- Vaccination Reminders ---")

today = datetime.now()

for child in self.children:

for record in child.vaccination\_records:

vaccine, date = record

if today <= date < today + timedelta(days=1):

print(f"Reminder: {child.parent\_name}, your child {child.name} has a {vaccine.name} vaccination appointment on {date.strftime('%Y-%m-%d')}.")

# Find a child by name

def find\_child\_by\_name(self, name):

for child in self.children:

if child.name.lower() == name.lower():

return child

return None

# Find a vaccine by name

def find\_vaccine\_by\_name(self, name):

for vaccine in self.vaccines:

if vaccine.name.lower() == name.lower():

return vaccine

return None

# Calculate age in months based on birthdate

def calculate\_age\_in\_months(self, birthdate):

today = datetime.now()

age\_in\_months = (today.year - birthdate.year) \* 12 + today.month - birthdate.month

return age\_in\_months

# Main application logic

def main():

system = VaccinationSystem()

print("Welcome to the Child Vaccination Management System")

while True:

print("\nOptions:")

print("1. Register a Child")

print("2. View Vaccine Schedule")

print("3. Schedule Vaccination")

print("4. View All Registered Children")

print("5. View Vaccination Records")

print("6. Send Reminders")

print("7. Exit")

choice = input("\nEnter your choice (1-7): ")

if choice == '1':

name = input("Enter child's name: ")

birthdate = datetime.strptime(input("Enter birthdate (YYYY-MM-DD): "), '%Y-%m-%d')

parent\_name = input("Enter parent's name: ")

contact = input("Enter contact number: ")

system.register\_child(name, birthdate, parent\_name, contact)

elif choice == '2':

child\_name = input("Enter child's name to view vaccine schedule: ")

system.view\_vaccine\_schedule(child\_name)

elif choice == '3':

child\_name = input("Enter child's name to schedule vaccination: ")

vaccine\_name = input("Enter vaccine name: ")

system.schedule\_vaccination(child\_name, vaccine\_name)

elif choice == '4':

system.view\_all\_children()

elif choice == '5':

child\_name = input("Enter child's name to view vaccination records: ")

system.view\_vaccination\_records(child\_name)

elif choice == '6':

print("\nExiting the system. Goodbye!")

break

else:

print("\nInvalid choice! Please select a valid option.")

# Run the main program

if \_name\_ == "\_main\_":

main()

**OUTPUT:**

Welcome to the Child Vaccination Management System Options:

1. Register a Child

2. View Vaccine Schedule

3. Schedule Vaccination

4. View All Registered Children

5. View Vaccination Records

6. Exit

Enter your choice (1-7): 1

Enter child's name: Anjali

Enter birthdate (YYYY-MM-DD): 2002-17-07

Enter parent's name: jana

Enter contact number: 8686289999

Anjali has been successfully registered.

Thank you …