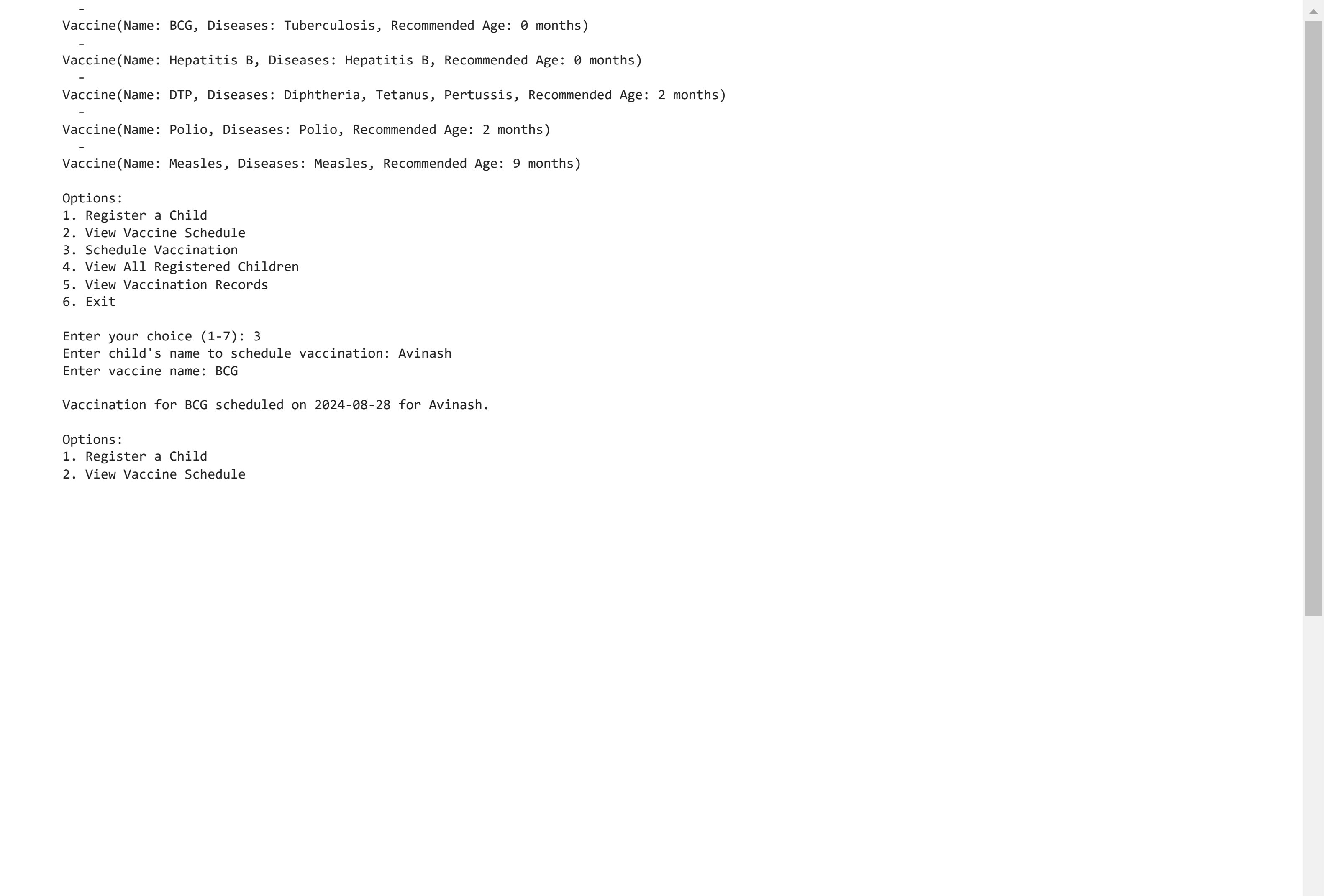
```python
from datetime import datetime, timedelta

# 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. 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()
```

```
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: Avinash
Enter birthdate (YYYY-MM-DD): 2002-08-03
Enter parent's name: Ravi
Enter contact number: 8008191065

Avinash has been successfully registered.

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): 2
Enter child's name to view vaccine schedule: Avinash

Vaccination Schedule for Avinash (Age: 264 months):
```



```
  -
Vaccine(Name: BCG, Diseases: Tuberculosis, Recommended Age: 0 months)
  -
Vaccine(Name: Hepatitis B, Diseases: Hepatitis B, Recommended Age: 0 months)
  -
Vaccine(Name: DTP, Diseases: Diphtheria, Tetanus, Pertussis, Recommended Age: 2 months)
  -
Vaccine(Name: Polio, Diseases: Polio, Recommended Age: 2 months)
  -
Vaccine(Name: Measles, Diseases: Measles, Recommended Age: 9 months)

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): 3
Enter child's name to schedule vaccination: Avinash
Enter vaccine name: BCG

Vaccination for BCG scheduled on 2024-08-28 for Avinash.

Options:
1. Register a Child
2. View Vaccine Schedule
```