**DLITHE PROJECT REPORT**

**PROJECT ID :** CP036

**PROJECT TITLE :** VACCINE REGISTRATION SYSTEM

**TEAM MEMBERS :** ABDULLA N A (4MT21CS004)

DHWANI SAGAR (4MT21CS048)

ASHWINI (4MT21CS033)

AJAY KUMAR YADAV (4MT21CS013)

DHANUSH S SHETTY (4MT21CS047)

**REPORT**

**Abstract :**

The Vaccine Registration System is a file handling-based application designed to track the vaccination records of individuals using their Aadhar numbers. This system allows users to register, manage vaccine records, and generate reports, ensuring a streamlined vaccination process.

**Introduction :**

**Background :**

The COVID-19 pandemic highlighted the need for efficient vaccine management systems. This project addresses the challenges of tracking vaccine distribution and administration.

**Objectives :**

* Create a user-friendly interface for registration.
* Store and manage vaccine records efficiently.
* Generate reports for tracking vaccine progress.

**Technologies Used**

* Programming Languages: C
* File Handling for data storage

**System Architecture**

Front-End :

The front-end is built using a c library for creating command line interface(CUI). It includes forms for user registration and vaccine data entry.

Back-End :

The back-end is written in c and handles user input validation, file handling for record management.

Database :

The database is implemented using text files where user information and vaccine records are stored in structured formats.

**Project Modules :**

The project consists of the following modules:

* Module 1: User Registration

- Users can register by providing their Aadhar numbers and basic information.

- Input validation ensures data accuracy.

* Module 2: Vaccine Record Management

- Allows authorized users to add and update vaccine records using Aadhar

- Ensures data integrity and security.

* Module 3: Report Generation

- Generates reports based on various criteria such as vaccination progress, date, etc

- Provides insights into vaccine distribution.

**Design and Implementation :**

Front-End Design :

* Design follows a simple and intuitive layout.
* Forms for registration and data entry are user-friendly.

Back-End Design :

* Code is organized into functions for readability and maintainability.
* Input validation prevents erroneous data entry.

Database Design :

* Data is stored in text files, each representing a user or vaccine record.
* Structured data format ensures easy retrieval and update.

**Features and Functionality :**

* Feature 1: User Registration

-Users can register with Aadhar numbers.

-Duplicate entries are prevented.

* Feature 2: Vaccine Record Entry

-Authorized personnel can add vaccine records linked to Aadhar numbers.

-Data accuracy is maintained.

* Feature 3: Search and Update Records

-Allows authorized users to search for records and update vaccine information.

* Feature 4: Report Generation

-Generates reports to track vaccine distribution and progress.

-Offers insights for decision-makers.

**Testing :**

Unit Testing :

* Individual functions are tested for correctness.
* Boundary cases are checked.

Integration Testing :

* Modules are tested for compatibility.
* Data flow is verified.

User Acceptance Testing :

* End-users validate system functionality.
* Feedback is incorporated for improvements.

**Challenges Faced :**

* Data security and privacy concerns.
* Handling concurrent access and file locking.
* Ensuring system scalability.

**Future Enhancements :**

* Integration with a centralized database.
* User roles and access control.

**Conclusion :**

The Vaccine Registration System is an essential tool for tracking and managing vaccination records. It offers a user-friendly interface and robust features to ensure data accuracy and accessibility.

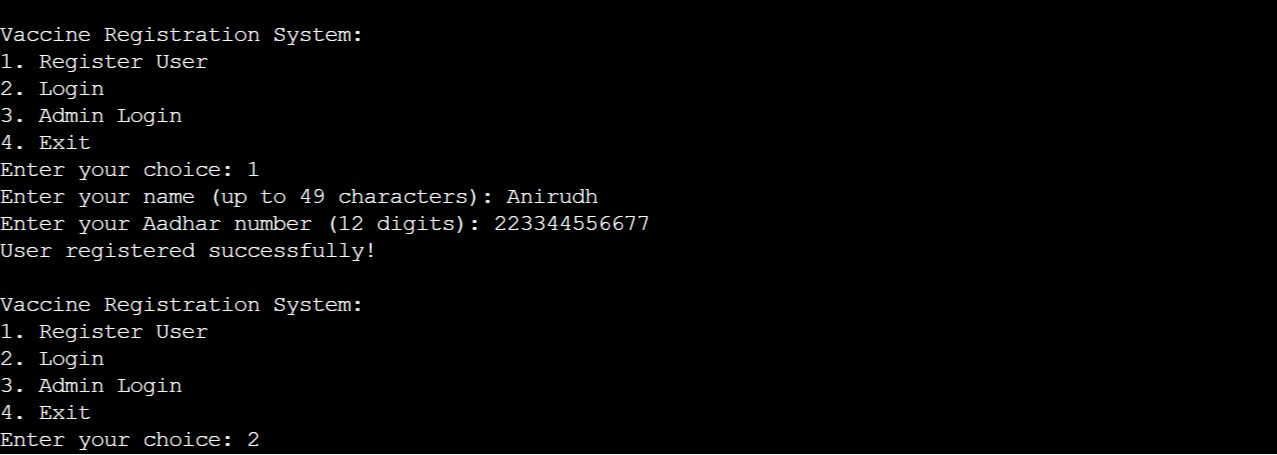
**References :**

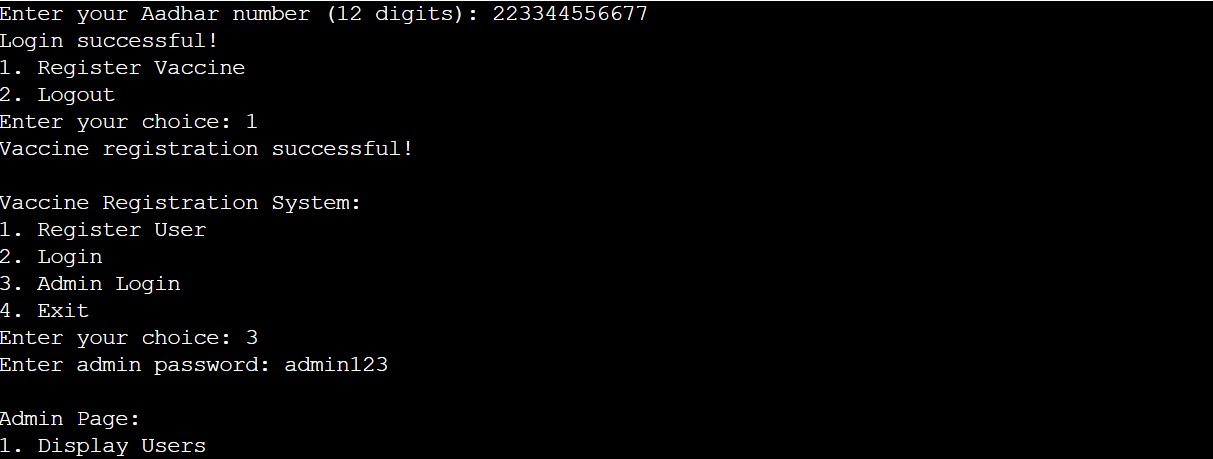
<https://www.geeksforgeeks.org/fseek-in-c-with-example/>

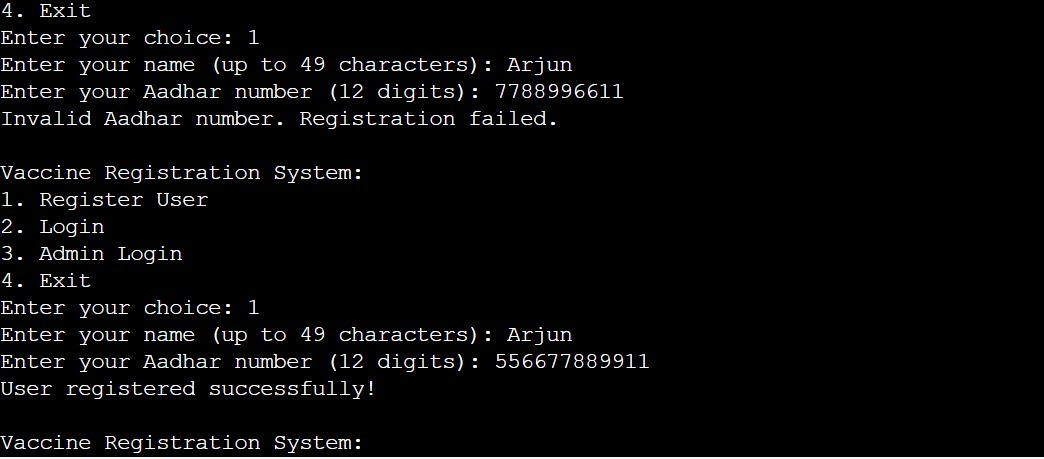
<https://www.cowin.gov.in>

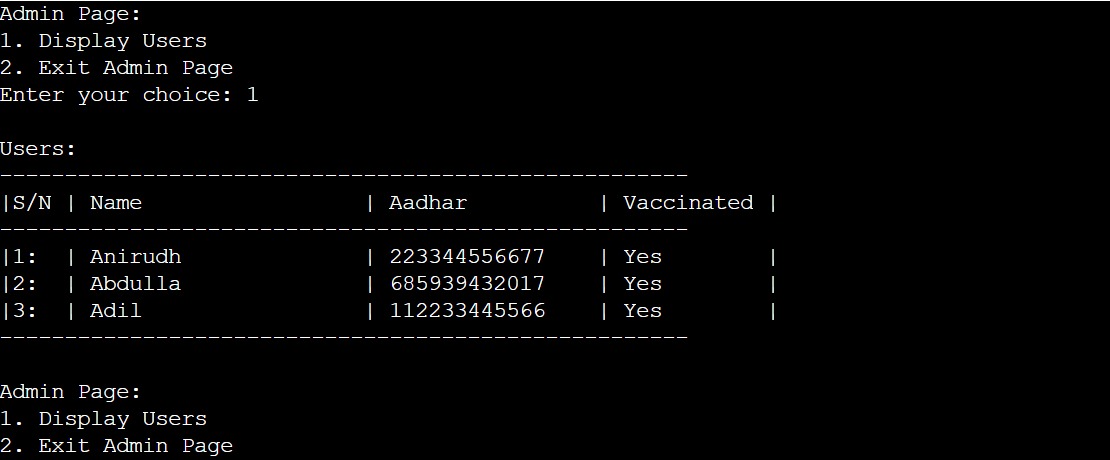
**Appendices :**

Screenshots









**Code Snippets:**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <stdbool.h>

#include <ctype.h>

#define MAX\_USERS 100 // Maximum number of users

#define MAX\_NAME\_LENGTH 50

#define MAX\_AADHAR\_LENGTH 15

struct User {

char name[MAX\_NAME\_LENGTH];

char aadhar[MAX\_AADHAR\_LENGTH];

int vaccinated;

};

void clearInputBuffer() {

int c;

while ((c = getchar()) != '\n' && c != EOF);

}

bool isAadharValid(const char \*aadhar) {

int length = strlen(aadhar);

if (length != 12) { // Aadhar numbers in India are typically 12 digits

return false;

}

for (int i = 0; i < length; i++) {

if (!isdigit(aadhar[i])) {

return false;

}

}

return true;

}

void registerUser() {

struct User newUser;

printf("Enter your name (up to %d characters): ", MAX\_NAME\_LENGTH - 1);

scanf("%49s", newUser.name);

clearInputBuffer();

printf("Enter your Aadhar number (12 digits): ");

scanf("%14s", newUser.aadhar);

clearInputBuffer();

if (!isAadharValid(newUser.aadhar)) {

printf("Invalid Aadhar number. Registration failed.\n");

return;

}

newUser.vaccinated = 0; // Set vaccinated status to 0 (not vaccinated)

FILE \*file = fopen("users.txt", "a");

if (file != NULL) {

fprintf(file, "%s %s %d\n", newUser.name, newUser.aadhar, newUser.vaccinated);

fclose(file);

printf("User registered successfully!\n");

} else {

printf("Error: Unable to register user.\n");

}

}

int loginUser(const char \*aadhar) {

FILE \*file = fopen("users.txt", "r");

if (file != NULL) {

struct User currentUser;

int userIndex = 0;

while (fscanf(file, "%49s %14s %d", currentUser.name, currentUser.aadhar, &currentUser.vaccinated) != EOF) {

if (strcmp(currentUser.aadhar, aadhar) == 0) {

fclose(file);

return userIndex; // Return user index if Aadhar is found

}

userIndex++;

}

fclose(file);

}

return -1; // Return -1 if Aadhar is not found

}

void registerVaccine(int userIndex) {

FILE \*file = fopen("users.txt", "a+");

if (file != NULL) {

struct User currentUser;

int currentIndex = 0;

while (fscanf(file, "%49s %14s %d", currentUser.name, currentUser.aadhar, &currentUser.vaccinated) != EOF) {

if (currentIndex == userIndex) {

currentUser.vaccinated = 1;

fseek(file, -strlen(currentUser.aadhar) - 2, SEEK\_CUR); // Move back to update line

fprintf(file, "%s %s %d\n", currentUser.name, currentUser.aadhar, currentUser.vaccinated);

printf("Vaccine registration successful!\n");

break;

}

currentIndex++;

}

fclose(file);

}

}

void displayUsers() {

FILE \*file = fopen("users.txt", "r");

int cnt=0;

if (file != NULL) {

struct User currentUser;

printf("\nUsers:\n");

printf("-----------------------------------------------------\n");

printf("|S/N | %-20s | %-15s | Vaccinated |\n", "Name", "Aadhar");

printf("-----------------------------------------------------\n");

while (fscanf(file, "%49s %14s %d", currentUser.name, currentUser.aadhar, &currentUser.vaccinated) != EOF) {

//printf("| %-20s | %-15s | %-10s |\n", currentUser.name, currentUser.aadhar, currentUser.vaccinated ? "Yes" : "No");

if(file,currentUser.vaccinated == 1){

cnt++;

printf("|%d: | %-20s | %-15s | %-10s |\n",cnt, currentUser.name, currentUser.aadhar, currentUser.vaccinated ? "Yes" : "No");

}

}

printf("-----------------------------------------------------\n");

fclose(file);

} else {

printf("Error: Unable to display users.\n");

}

}

void adminLogin() {

char adminPassword[15];

printf("Enter admin password: ");

scanf("%14s", adminPassword);

clearInputBuffer();

// Change the password to match your admin password

if (strcmp(adminPassword, "admin123") == 0) {

int choice;

while (1) {

printf("\nAdmin Page:\n");

printf("1. Display Users\n2. Exit Admin Page\n");

printf("Enter your choice: ");

scanf("%d", &choice);

clearInputBuffer();

switch (choice) {

case 1:

displayUsers(); // Display users from here

break;

case 2:

printf("Exiting Admin Page...\n");

return;

default:

printf("Invalid choice!\n");

break;

}

}

} else {

printf("Admin login failed.\n");

}

}

int main() {

int choice;

char aadhar[15];

int userIndex = -1; // Default value indicating no user is logged in

while (1) {

printf("\nVaccine Registration System:\n");

printf("1. Register User\n2. Login\n3. Admin Login\n4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

clearInputBuffer();

switch (choice) {

case 1:

registerUser();

break;

case 2: {

printf("Enter your Aadhar number (12 digits): ");

scanf("%14s", aadhar);

clearInputBuffer();

userIndex = loginUser(aadhar);

if (userIndex != -1) {

printf("Login successful!\n");

printf("1. Register Vaccine\n2. Logout\nEnter your choice: ");

int userChoice;

scanf("%d", &userChoice);

clearInputBuffer();

switch (userChoice) {

case 1:

registerVaccine(userIndex);

break;

case 2:

userIndex = -1;

printf("Logged out.\n");

break;

default:

printf("Invalid choice!\n");

break;

}

} else {

printf("Login failed. User not found.\n");

}

break;

}

case 3:

adminLogin();

break;

case 4:

printf("Exiting...\n");

return 0;

default:

printf("Invalid choice!\n");

break;

}

}

return 0;

}