PROGRAM:-

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#define PORT 5050

#define BUFFER\_SIZE 1024

// Simulated DNS records (static database)

struct DNSRecord {

char domain[50];

char ip[20];

};

struct DNSRecord dnsTable[] = {

{"example.com", "93.184.216.34"},

{"google.com", "142.250.190.14"},

{"yahoo.com", "98.137.11.163"},

{"facebook.com", "157.240.22.35"}

};

char\* resolveDomain(char\* domain) {

int records = sizeof(dnsTable) / sizeof(dnsTable[0]);

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

if (strcmp(domain, dnsTable[i].domain) == 0) {

return dnsTable[i].ip;

}

}

return "Not Found"; // If domain is not in our database

}

int main() {

int sockfd;

struct sockaddr\_in serverAddr, clientAddr;

char buffer[BUFFER\_SIZE];

socklen\_t addrLen = sizeof(clientAddr);

// Create UDP socket

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) < 0) {

perror("Socket creation failed");

exit(EXIT\_FAILURE);

}

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_addr.s\_addr = INADDR\_ANY;

serverAddr.sin\_port = htons(PORT);

// Bind the socket

if (bind(sockfd, (struct sockaddr\*)&serverAddr, sizeof(serverAddr)) < 0) {

perror("Bind failed");

exit(EXIT\_FAILURE);

}

printf("DNS Server is running on port %d...\n", PORT);

while (1) {

// Receive request from client

recvfrom(sockfd, buffer, BUFFER\_SIZE, 0, (struct sockaddr\*)&clientAddr, &addrLen);

buffer[strcspn(buffer, "\n")] = '\0'; // Remove newline if any

printf("Received request for domain: %s\n", buffer);

// Get the corresponding IP

char\* ipAddress = resolveDomain(buffer);

// Send response to client

sendto(sockfd, ipAddress, strlen(ipAddress), 0, (struct sockaddr\*)&clientAddr, addrLen);

printf("Responded with IP: %s\n", ipAddress);

}

close(sockfd);

return 0;

}

OUTPUT:--

DNS Server is running on port 5050...

Received request for domain: google.com

Responded with IP: 142.250.190.14