DATABASES

For

The Parliamentary voting system

Simulate/operate a simplified election workflow: registration, polling, counting (including PR allocation), preference counting, and results

Election Admin chapter 01

districts.txt

Election Admin → District Registration
Election Admin → view Districts

District_code | District | seat

D002|Gampaha|18
D003|Kandy|16

Data saving example

```c
#include <stdlib.h>
#define DISTRICT_FILE "districts.txt"

char district_code[10];
char district_name[50];
int seats;
FILE *fp = fopen(PARTY_FILE, "a");
    if (fp == NULL) {
        printf("Error opening file for parties!\n");
        exit(1);
    }
fprintf(fp, "%s|%s|%s\n", party_code, party_name, party_symbol);
fclose(fp);
```

Data reding in file example

```c
char line[200];
FILE *fp = fopen(DISTRICT_FILE, "r");

if (fp == NULL) {
        printf("No districts registered yet.\n");
        return;
}


while (fgets(line, sizeof(line), fp)) {
        printf("%s", line);
}
fclose(fp);
```

parties.txt

Election Admin → party Registration
Election Admin → view partys

```
----------------------------------------------------------------

Party_code | party | symbol

P001|Unity National Party|Hand
P002|People's Freedom Front|Lotus
P003|National People's Alliance|Lamp
```

Party Admin chapter 02

RO_nomination_<party_code>.txt

```
Candidate Name|Candidate Code|District Code|NIC|Date of Birth
Mr. A. Perera|C001|D012|199052315678|1990-05-23
Ms. B. Silva|C002|D012|199184512345|1985-11-14
```

After Submission to RO (system renames file):
RO_nomination_P001.txt

```c
#include <stdlib.h>
#include <string.h>

char party_code[10], party_name[50];
    char filename[60];
    char candidate_name[50];
    char candidate_code[10];
    char district_code[10];
    char nic[20];
    char dob[15];
    char choice;

printf("Enter Party Code   : ");
    scanf("%s", party_code);

    printf("Enter Party Name   : ");
    scanf(" %[^\n]", party_name);

    // Output file
    sprintf(filename, "RO_nomination_%s.txt", party_code);

    FILE *fp = fopen(filename, "w");
    if (fp == NULL) {
        printf("Error: cannot create file!\n");
        exit(1);
    }

    // Write party header at top
    fprintf(fp, "Party Code: %s | Party Name: %s\n", party_code, party_name);
    fprintf(fp, "---------------------------------------------------\n");
    fprintf(fp, "Candidate Name|Candidate Code|District Code|NIC|Date of
Birth\n");

    do {
        printf("---------------------------------------------------\n");
        printf("Enter Candidate Name : ");
```

```c
        scanf(" %[^\n]", candidate_name);

        printf("Enter Candidate Code : ");
        scanf("%s", candidate_code);

        printf("Enter District Code  : ");
        scanf("%s", district_code);

        printf("Enter NIC Number     : ");
        scanf("%s", nic);

        printf("Enter Date of Birth  : ");
        scanf("%s", dob);

        // Save to file
        fprintf(fp, "%s|%s|%s|%s|%s\n",
                candidate_name, candidate_code, district_code, nic, dob);

        printf("--------------------------------------------------\n");
        printf("Add another candidate? (Y/N): ");
        scanf(" %c", &choice);

    } while (choice == 'Y' || choice == 'y');

    fclose(fp);
    printf("\nNomination list successfully created: %s\n", filename);
}
```

RO **chapter 03**

AC_nomination_<party_code>.txt

```
0001|Mr. A. Perera|C001|D012|199052315678|1990-05-23
0002|Ms. B. Silva|C002|D012|199184512345|1985-11-14

printf("Enter Party Code to submit nomination: ");
scanf("%s", party_code);

// Original filename prepared by Party Agent
sprintf(old_filename, "RO_nomination_%s.txt", party_code);

// New filename for RO
sprintf(new_filename, "AC_nomination_%s.txt", party_code);

// Try renaming the file
    if (rename(old_filename, new_filename) == 0) {
        printf("Nomination file submitted successfully!\n");
        printf("File renamed to: %s\n", new_filename);
    } else {
        perror("Error: Unable to submit nomination file");
    }
```