

# **Manav Rachna International Institute** **of Research and Studies**



## **SCHOOL OF COMPUTER APPLICATION**

MCA - B 1st SEM

PROJECT REPORT

(Course Code: 6.0CA101C01H)

**SUBMITTED BY:**

Shreya 25/SCA/MCAN/074

Nitesh Kumar 25/SCA/MCAN/075

Akriti 25/SCA/MCAN/ 076

Nishtha Kalra 25/SCA/MCAN/ 077

**SUBMITTED TO:**

Dr. Sachin Sharma

- ♣ **Design a system that organises and preserves Vedic texts in digital form, ensuring easy access to verses, rituals, and philosophical content for future generations: -**

**CODE: -**

```
/* vedadb.c
   Simple Vedic Text Database (CLI)
   Features: add, list, search, save, load, export
   Compile: gcc -std=c11 -O2 -o vedadb vedadb.c
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
#include <ctype.h>

#define DB_FILENAME "vedadb.bin"
#define EXPORT_FILENAME "vedadb_export.txt"
#define MAX_LINE 4096

typedef struct {
    char *title;
    char *source;
    char *language;
    char *tags;    // comma-separated
    char *text;    // main verse/ritual/philosophy
} Entry;

typedef struct {
    Entry *items;
    size_t count;
    size_t capacity;
} DB;

static char* my_strdup(const char *s) {
```

```

    if(!s) return NULL;
    size_t n = strlen(s) + 1;
    char *d = malloc(n);
    if(!d) {
        perror("malloc");
        exit(1);
    }
    memcpy(d, s, n);
    return d;
}

void db_init(DB *db) {
    db->items = NULL;
    db->count = 0;
    db->capacity = 0;
}

void db_free_entry(Entry *e) {
    if(!e) return;
    free(e->title);
    free(e->source);
    free(e->language);
    free(e->tags);
    free(e->text);
}

void db_free(DB *db) {
    if(!db) return;
    for(size_t i=0; i<db->count; i++) db_free_entry(&db->items[i]);
    free(db->items);
    db->items = NULL;
    db->count = db->capacity = 0;
}

void db_ensure_capacity(DB *db) {
    if(db->count < db->capacity) return;
    size_t newcap = (db->capacity==0)?8:db->capacity*2;
    Entry *tmp = realloc(db->items, newcap*sizeof(Entry));
    if(!tmp) {
        perror("realloc");
        exit(1);
    }
    db->items = tmp;
    db->capacity = newcap;
}

```

```
}
```

```
void db_add(DB *db, Entry e) {  
    db_ensure_capacity(db);  
    db->items[db->count++] = e;  
}
```

```
static void trim_newline(char *s) {  
    if(!s) return;  
    size_t L = strlen(s);  
    while(L>0 && (s[L-1]=='\n' || s[L-1]=='\r')) {  
        s[L-1]=0;  
        L--;  
    }  
}
```

```
char *read_multiline(const char* prompt) {  
    printf("%s (end with a single line containing only a dot '.')\n", prompt);  
    char buffer[MAX_LINE];  
    size_t total = 0;  
    char *result = NULL;  
    while(1) {  
        if(!fgets(buffer, sizeof(buffer), stdin)) break;  
        if(strcmp(buffer, ".\n")==0 || strcmp(buffer, ".r\n")==0) break;  
        size_t len = strlen(buffer);  
        char *tmp = realloc(result, total + len + 1);  
        if(!tmp) {  
            perror("realloc");  
            free(result);  
            exit(1);  
        }  
        result = tmp;  
        memcpy(result + total, buffer, len);  
        total += len;  
        result[total] = '\0';  
    }  
    if(!result) result = my_strdup("");  
    return result;  
}
```

```
void cmd_add(DB *db) {  
    char line[MAX_LINE];  
    Entry e = {0};  
    printf("Title: ");
```

```

    fflush(stdout);
    if(!fgets(line, sizeof(line), stdin)) return;
    trim_newline(line);
    e.title = my_strdup(line);

    printf("Source (e.g., Rigveda 1.1): ");
    if(!fgets(line, sizeof(line), stdin)) return;
    trim_newline(line);
    e.source = my_strdup(line);

    printf("Language (e.g., Sanskrit, English translation): ");
    if(!fgets(line, sizeof(line), stdin)) return;
    trim_newline(line);
    e.language = my_strdup(line);

    printf("Tags (comma-separated, e.g., ritual,mantra,creation): ");
    if(!fgets(line, sizeof(line), stdin)) return;
    trim_newline(line);
    e.tags = my_strdup(line);

    e.text = read_multiline("Enter the verse/text (multiline)");

    db_add(db, e);
    printf("Entry added.\n");
}

void cmd_list(DB *db) {
    if(db->count==0) {
        printf("Database empty.\n");
        return;
    }
    for(size_t i=0; i<db->count; i++) {
        Entry *e = &db->items[i];
        printf("ID: %zu | Title: %s | Source: %s | Lang: %s | Tags: %s\n",
            i, e->title, e->source, e->language, e->tags);
    }
}

static void strtolower(char *s) {
    for(; *s; s++) *s = (char)tolower((unsigned char)*s);
}

void cmd_search(DB *db) {
    char q[MAX_LINE];

```

```

printf("Search query (case-insensitive substring): ");
if(!fgets(q, sizeof(q), stdin)) return;
trim_newline(q);
if(strlen(q)==0) {
    printf("Empty query.\n");
    return;
}
char ql[MAX_LINE];
strncpy(ql, q, sizeof(ql));
ql[sizeof(ql)-1]=0;
strtolower(ql);

int found = 0;
for(size_t i=0; i<db->count; i++) {
    Entry *e = &db->items[i];
    // prepare copies for case-insensitive test
    char *t = my_strdup(e->title?e->title:"");
    char *tx = my_strdup(e->text?e->text:"");
    char *tg = my_strdup(e->tags?e->tags:"");
    strtolower(t);
    strtolower(tx);
    strtolower(tg);

    if(strstr(t, ql) || strstr(tx, ql) || strstr(tg, ql) || strstr(e->source?strdup(e->source):"", ql))
    {
        printf("----\nID: %zu\nTitle: %s\nSource: %s\nLanguage: %s\nTags: %s\nText:\n%s\n",
i, e->title, e->source, e->language, e->tags, e->text);
        found++;
    }
    free(t);
    free(tx);
    free(tg);
}
if(!found) printf("No matches found.\n");
}

/* Persistence: store number of records then for each record store length-prefixed fields */
int save_db(DB *db, const char *filename) {
    FILE *f = fopen(filename, "wb");
    if(!f) {
        perror("fopen");
        return 0;
    }
    uint64_t count = db->count;

```

```

    if(fwrite(&count, sizeof(count), 1, f) != 1) {
        perror("fwrite");
        fclose(f);
        return 0;
    }
    for(size_t i=0; i<db->count; i++) {
        Entry *e = &db->items[i];
        // helper macro to write a string as uint32 length + bytes
#define WRITE_STR(s) do { \
            uint32_t len = (s)?(uint32_t)strlen(s):0; \
            if(fwrite(&len, sizeof(len),1,f)!=1){ perror("fwrite"); fclose(f); return 0; } \
            if(len>0 && fwrite(s,1,len,f)!=(size_t)len){ perror("fwrite"); fclose(f); return 0; } \
        } while(0)
        WRITE_STR(e->title);
        WRITE_STR(e->source);
        WRITE_STR(e->language);
        WRITE_STR(e->tags);
        WRITE_STR(e->text);
#undef WRITE_STR
    }
    fclose(f);
    return 1;
}

int load_db(DB *db, const char *filename) {
    FILE *f = fopen(filename, "rb");
    if(!f) return 0; // not fatal - no file yet
    uint64_t count = 0;
    if(fread(&count, sizeof(count), 1, f) != 1) {
        fclose(f);
        return 0;
    }
    for(uint64_t i=0; i<count; i++) {
#define READ_STR(to) do { \
            uint32_t len; \
            if(fread(&len, sizeof(len),1,f)!=1){ perror("fread"); fclose(f); return 0; } \
            if(len>0){ char *buf = malloc(len+1); if(!buf){ perror("malloc"); fclose(f); return 0; } \
                if(fread(buf,1,len,f)!=(size_t)len){ perror("fread"); free(buf); fclose(f); return 0; } \
            buf[len]=0; to = buf; } else { to = my_strdup(""); } \
        } while(0)
        Entry e;
        READ_STR(e.title);
        READ_STR(e.source);
        READ_STR(e.language);

```

```

        READ_STR(e.tags);
        READ_STR(e.text);
        db_add(db, e);
#undef READ_STR
    }
    fclose(f);
    return 1;
}

void cmd_save(DB *db) {
    if(save_db(db, DB_FILENAME)) printf("Saved to %s\n", DB_FILENAME);
    else printf("Failed to save DB.\n");
}

void cmd_load(DB *db) {
    if(load_db(db, DB_FILENAME)) printf("Loaded DB from %s\n", DB_FILENAME);
    else printf("No DB found or load failed.\n");
}

void cmd_export(DB *db) {
    FILE *f = fopen(EXPORT_FILENAME, "w");
    if(!f) {
        perror("fopen");
        return;
    }
    for(size_t i=0; i<db->count; i++) {
        Entry *e = &db->items[i];
        fprintf(f, "ID: %zu\nTitle: %s\nSource: %s\nLanguage: %s\nTags: %s\nText:\n%s\n", i, e->title, e->source, e->language, e->tags, e->text);
        fprintf(f, "-----\n");
    }
    fclose(f);
    printf("Exported to %s\n", EXPORT_FILENAME);
}

void print_help() {
    printf("Commands:\n");
    printf(" add  - add new entry\n");
    printf(" list - list entries (ID, title, source, tags)\n");
    printf(" search - search substring in title/text/tags\n");
    printf(" save  - save DB to disk\n");
    printf(" load  - load DB from disk\n");
    printf(" export - export all entries to text file\n");
    printf(" help  - show this help\n");
}

```



```

    printf(" quit - exit\n");
}

int main(void) {
    DB db;
    db_init(&db);
    cmd_load(&db);
    printf("Vedic Text DB CLI\nType 'help' for commands.\n");
    char cmd[MAX_LINE];
    while(1) {
        printf("> ");
        fflush(stdout);
        if(!fgets(cmd, sizeof(cmd), stdin)) break;
        trim_newline(cmd);
        if(strcmp(cmd, "quit")==0 || strcmp(cmd, "exit")==0) break;
        else if(strcmp(cmd, "help")==0) print_help();
        else if(strcmp(cmd, "add")==0) cmd_add(&db);
        else if(strcmp(cmd, "list")==0) cmd_list(&db);
        else if(strcmp(cmd, "search")==0) cmd_search(&db);
        else if(strcmp(cmd, "save")==0) cmd_save(&db);
        else if(strcmp(cmd, "load")==0) {
            db_free(&db);
            db_init(&db);
            cmd_load(&db);
        }
        else if(strcmp(cmd, "export")==0) cmd_export(&db);
        else if(strlen(cmd)==0) continue;
        else printf("Unknown command. Type 'help'.\n");
    }
    printf("Saving DB before exit...\n");
    save_db(&db, DB_FILENAME);
    db_free(&db);
    return 0;
}

```

## OUTPUT: -

```
6
7 #include <stdio.h>
8 #include <stdlib.h>
9 #include <string.h>
10 #include <stdint.h>
11 #include <ctype.h>
12
13 #define DB_FILENAME "vedadb.bin"
14 #define EXPORT_FILENAME "vedadb_export.txt"
15 #define MAX_LINE 4096
16
17 #include <stdint.h>
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Type 'help' for commands.  
>

PS C:\Users\Witesh> & 'c:\Users\Witesh\.vscode\extensions\ms-vscode.cpptools-1.28.3-win32-x64\debugAdapters\bin\WindowsDeb  
id-sc3ah8r3.1z5' '--dbgExe=C:\Windows\bin\gdb.exe' '--interpreter=mi'

No DB found or load failed.  
Vedic Text DB CLI  
Type 'help' for commands.  
> help

Commands:

- add - add new entry
- list - list entries (ID, title, source, tags)
- search - search substring in title/text/tags
- save - save DB to disk
- load - load DB from disk
- export - export all entries to text file
- help - show this help
- quit - exit

> add

Title: Nasadiya Sukta  
Source (e.g., Rigveda 1.1): Rigveda 10.129  
Language (e.g., Sanskrit, English translation): Sanskrit (with English translation)  
Tags (comma-separated, e.g., ritual,mantra,creation): creation,cosmology,mantra  
Enter the verse/text (multiline) (end with a single line containing only a dot '.')

In the beginning there was neither existence nor non-existence.  
There was neither the realm of space nor the sky beyond.  
What stirred? Whence came this creation? The gods came afterward, with the creation of this universe.

.

Entry added.  
> add

Title: Gayatri Mantra  
Source (e.g., Rigveda 1.1): Rigveda 3.62.10  
Language (e.g., Sanskrit, English translation): Sanskrit (transliteration + eng.)  
Tags (comma-separated, e.g., ritual,mantra,creation): mantra,devotional  
Enter the verse/text (multiline) (end with a single line containing only a dot '.')

Om bhur bhuvah svah  
Tat savitur varenyam  
Bhargo devasya dhimahi  
Dhiyo yo nah prachodayat

.

Entry added.  
> save

Saved to vedadb.bin  
> load


Loaded DB from vedadb.bin  
> quit


Saving DB before exit...


```
32 memcpy(result + total, buffer, len);
33 total += len;
34 return result;
}


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Title: Gayatri Mantra
Source (e.g., Rigveda 1.1): Rigveda 3.62.10
Language (e.g., Sanskrit, English translation): Sanskrit (transliteration + eng.)
Tags (comma-separated, e.g., ritual, mantra, creation): mantra, devotional
Enter the verse/text (multiline) (end with a single line containing only a dot '.')
Om bhur bhuvah svah
tat savitur varenyam
Bhargo devasya dhīmahi
Dhiyo yo nah prachodayat
.
Entry added.
> save
Saved to vedadh.bin
> load
Loaded DB from vedadh.bin
> quit
Saving DB before exit...
PS C:\Users\Nitesh> & 'c:\Users\Nitesh\vscode\extensions\ms-vscode.cpptools-1.20.3-win32-x64\debugadapters\bin\windowsdbg_launcher.exe' --stdin=Microsoft.VisualStudio.Debugger.Host --stdout=Microsoft.VisualStudio.Debugger.Host --stderr=Microsoft.VisualStudio.Debugger.Host --pid=Microsoft.VisualStudio.Debugger.Host --log=C:\Users\Nitesh\vscode\logs\cpptools-1.20.3-win32-x64\log.txt --interpreter=ai
Loaded DB from vedadh.bin
Vedic Text DB CLI
Type 'help' for commands.
> list
ID: 0 | Title: Nasadiya Sukta | Source: Rigveda 10.129 | Lang: Sanskrit (with English translation) | Tags: creation, cosmology, mantra
ID: 1 | Title: Gayatri Mantra | Source: Rigveda 3.62.10 | Lang: Sanskrit (transliteration + eng.) | Tags: mantra, devotional
> search
Search query (case-insensitive substring): Nasadiya Sukta
----
ID: 0
Title: Nasadiya Sukta
Source: Rigveda 10.129
Language: Sanskrit (with English translation)
Tags: creation, cosmology, mantra
Text:
In the beginning there was neither existence nor non-existence.
There was neither the realm of space nor the sky beyond.
What stirred? Whence came this creation? The gods came afterward, with the creation of this universe.
> export
Exported to vedadh_export.txt
> |
```

 Received Files

 vedadh\_export.txt  
C:\Users\Nitesh\Downloads

 Open

 Show hidden icons

## Exported data file: -

```
File Edit View H1 v H1 v B I
ID: 0
Title: Nasadiya Sukta
Source: Rigveda 10.129
Language: Sanskrit (with English translation)
Tags: creation,cosmology,mantra
Text:
In the beginning there was neither existence nor non-existence.
There was neither the realm of space nor the sky beyond.
What stirred? Whence came this creation? The gods came afterward, with the creation of this universe.

-----
ID: 1
Title: Gayatri Mantra
Source: Rigveda 3.62.10
Language: Sanskrit (transliteration + eng.)
Tags: mantra,devotional
Text:
Om bhur bhuvah svah
Tat savitur varenyam
Bhargo devasya dhimahi
Dhiyo yo nah prachodayat

-----
```