A PLB report on

Title : A Department Diary:

Department Activity Monitoring System

A Department Diary: Department Activity Monitoring System REPORT

By

Sarthak Rajesh Vispute (B25IT1130)

Sakshi Shrikrishna Shirole (B25IT1123)

Vedika Devendra Suryawanshi (B25IT1125)

Sharley Yogiraj More (B25IT1120)

Guide: **Prof.Priti Warungse**

Department of Engineering Science and Humanities

Marathwada Mitra Mandal's

College of Engineering,

Karvenagar, Pune

[Sem-I, A.Y. 2025-26]

Marathwada Mitra Mandal's

College of Engineering,

Karvenagar, Pune

C E R T I F I C A T E

This is to certify that **Sarthak Rajesh Vispute (B25IT1130), Sakshi Shrikrishna Shirole (B25IT1123), Vedika Devendra Suryawanshi (B25IT1125), Sharley Yogiraj More(B25IT1120),** have successfully completed the PBL Project entitled -"**A Department Diary: Department Activity Monitoring System"** under my supervision, in the partial fulfillment of First Year Engineering of Savitribai Phule Pune University.

Date: 05/12/2025

Place: Pune

Guide's Name : **Prof.Priti Warungse**

Dr. A.S. Sawaikar

Head Principal DESH MMCOE, Pune

MMCOE, Pune

ACKNOWLEDGEMENT

We take this opportunity here to thank all those who had helped us in making this PBL project a reality.

First of all, we express our deep gratitude to our project guide Prof.Priti Warungse for his valuable support, help & guidance from time to time during the project work. We are also
grateful to our Head of Department, Dr. A.S. Sawaikar for giving us this opportunity to present this project report.

Last but not the least; we would like to thank our entire teaching and Non-teaching staff who assisted us directly or indirectly throughout the duration of this project.

Name of students (PRN)

1) **Sarthak Rajesh Vispute** (B25IT1130)

2) **Sakshi Shrikrishna Shirole** (B25IT1123)

3) **Vedika Devendra Suryawanshi** (B25IT1125)

4) **Sharley Yogiraj More** (B25IT1120)

9.2	Project Demo Video Link (YouTube / Recorded Demo)	8
9.3	Blog or Tutorial Link (If any)	8
9.4	References (Books, Websites, Articles)	8

Symbol Meaning

Symbol	Meaning
`#include`	**Include library**
`#define`	**Define constant**
`FILE *`	**File pointer**
`struct`	**Custom data type**
`fopen()`	**Open file**
`fread()` / `fwrite()`	**Read/write file**
`fclose()`	**Close file**
`fgets()`	**Read input**

`printf()` / `scanf()`	**Output/input**
`getchar()`	**Clear buffer / read char**
`{ }` / `()` / `;`	**Code block, function args, statement end**
`*` / `&` / `->`	**Pointer / address / struct access**
`==` / `!=` / `>` / `<`	**Comparisons**

Greek Symbols

Symbol	Meaning
α (alpha)	**Priority / coefficient**
β (beta)	**Secondary value / follow-up**
γ (gamma)	**Weight / factor**

δ (delta)	**Change / time difference**
ε (epsilon)	**Small margin / tolerance**
θ (theta)	**Angle / time slot**
λ (lambda)	**Frequency / rate**
μ (mu)	**Average duration**
ρ (rho)	**Density / intensity**
σ (sigma)	**Sum / variation**
τ (tau)	**Total duration**
ω (omega)	**End time / final marker**

1. Introduction

1.1 Background of the Problem

- Departments maintain daily records manually.
- Manual records lead to errors, are time-consuming, and make data retrieval difficult.

1.2 About the Project

- The project aims to develop a **computerized diary management tool**.
- Stores information like notices, events, schedules, and updates in an organized manner.

1.3 Objectives

- Maintain an organized record of departmental activities.
- Reduce manual work and minimize errors.
- Provide quick access to daily records.
- Automate storage and retrieval using **C file handling**.

1.4 Scope of the Project

- Useful for academic departments.
- Can be extended for **multi-user access**.
- Future upgrades: cloud storage and mobile application.

---

2. Literature Review & Research

2.1 Review of Existing Programs

- Basic record programs in C.

- Attendance management systems.
- Simple digital diaries.

2.2 Research on Techniques / Methods

- File handling functions: `fopen`, `fprintf`, `fread`, etc.
- Structured data in C.
- Menu-driven interface for user interaction.

2.3 Summary of Findings

- Existing programs lack **multi-category departmental record management**.

2.4 How This Project is Different / Innovative

- Custom-built for **departmental operations**.
- Supports multiple record categories.
- Provides an **easy text-based interface**.

---

3. Problem Definition & Requirements

3.1 Problem Statement

- Manual diary management is slow, disorganized, and prone to data loss.
- A digital system is required.

3.2 Motivation

- To create an **efficient, accurate, and accessible record-management system**.

3.3 Project Features

- Add, view, update, and delete records.
- Search functionality.
- Categorized entries.

3.4 Requirements

3.4.1 Hardware Requirements

- PC/Laptop with **1 GHz Processor** and **2 GB RAM**.
- Minimum 200 MB free storage.
- Standard VGA/HD display.
- Keyboard as input device.

3.4.2 Software Requirements

- Operating System: Windows/Linux.
- C Compiler: GCC, MinGW, Turbo C, or CodeBlocks.
- Text/data file creation supported.

3.5 Expected Output

- A **fully functioning digital diary system** with menu-driven navigation.

4. Methodology

4.1 System Architecture

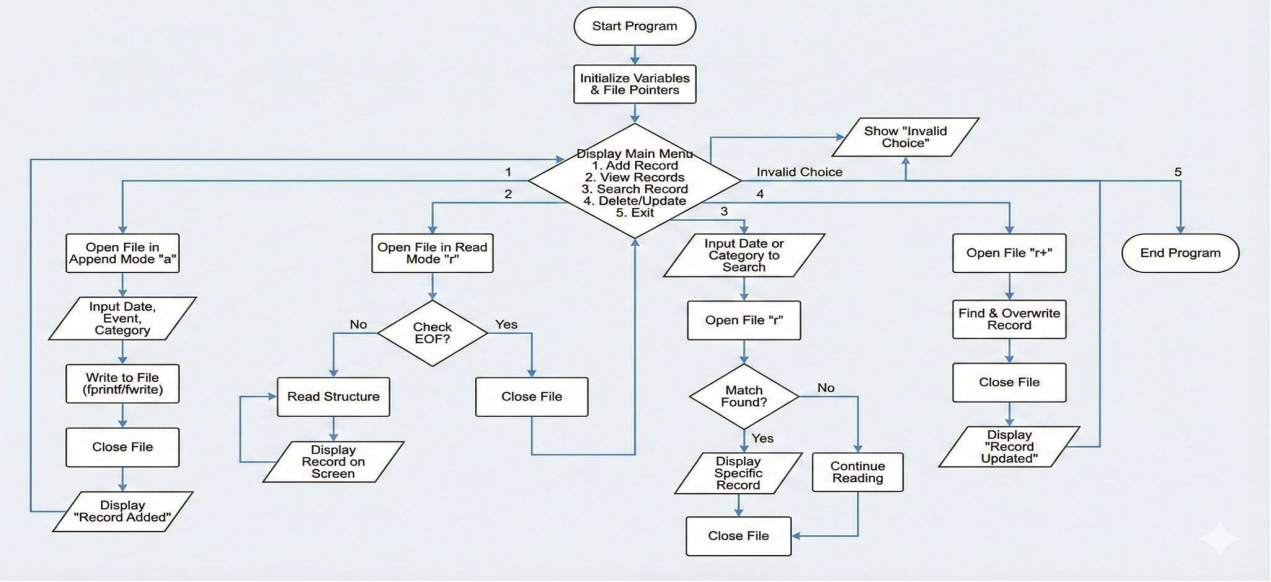
```
User → Menu → File Operations → Output
```

4.2 Algorithm

1. Start program.
2. Display menu options.
3. Choose operation.
4. Perform read/write operations on file.
5. Display results.
6. Exit program.

4.3 Flowchart

- (Flowchart showing menu → operations → output → exit)



4.4 Explanation of Logic Used

- **Switch-case** for menu navigation.
- **Structs** to store data.
- **File handling** for data persistence.

---

5. Implementation in C Language

5.1 Sample Input and Output

Menu:

```
1. Add Activity
2. View All
3. Edit
4. Delete
5. Follow-up
6. Exit
```

Example Input/Output:

- Choice: 1
    - Activity Name: Yoga
    - Date (DD-MM-YYYY): 2-12-25
    - Time: 9:00 AM
    - Duration: 1 Hour
    - Address: Yoga Hall
- Activity (ID 1) added.
- Choice: 2 (View All)

```
ID | Name | Date      | Time      | Duration | Address
-----------------------------------------------------------------------------------------------------
1  | Yoga | 2-12-25   | 9:00 AM   | 1 Hour   | Yoga Hall
```

5.2 C Concepts Used

- Arrays
- Structures
- File Handling
- Functions

5.3 Program Flow / Stepwise Execution

- Stepwise process from user input → file writing → output display.

5.4 Data Structures Used

```
struct Record

{

    char date[20];

    char event[100];

};
```

5.5 Challenges / Issues Faced

- File pointer mismanagement.

5.6 Testing

- White-box testing
- Boundary tests

5.7 Test Cases Used

Test Case	Input	Expected Output
Add New Record	Date: 12/01/2025 Event: Staff Meeting	Record successfully saved to the file
View All Records	Select “View Records”	All previously saved records displayed
Search Record	Date: 12/01/2025	Display matching record(s)
Search Record (Not Found)	Date: 20/01/2025	“Record not found” message
Delete Record	Date: 12/01/2025	Record removed; file updated
Update Record	Date: 10/01/2025 → modify event	Updated record displayed
Invalid Input	Special characters/empty date	Error: “Invalid input. Please try again”
Exit Program	Choose Exit	Program terminates safely

---

6. Innovative Techniques / Enhancements

6.1 New Features Added

- Search by date
- Categorized listings

6.2 Efficiency Improvements

- Optimized file reads
- Structured data storage

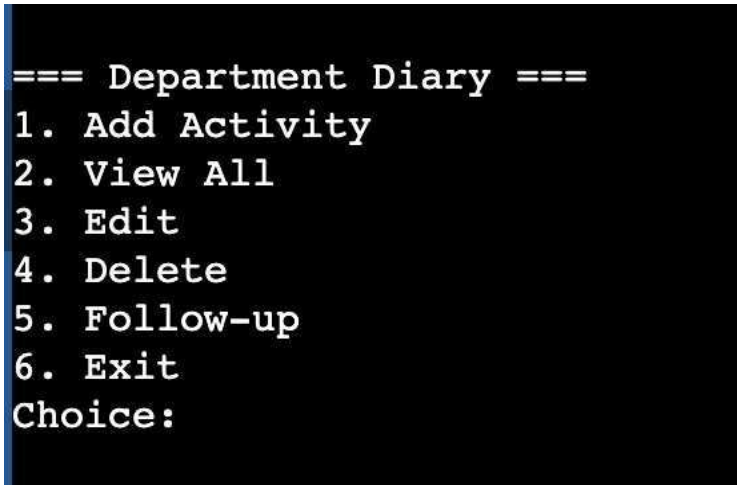
6.3 UI / Ease-of-Use Enhancements

- Clean, menu-driven interface
- Error-handling messages

---

7. Results and Discussion

7.1 Output Screens



7.2 Result Discussion

- System works efficiently and handles large records.

---

8. Conclusion

8.1 Conclusion

- Successfully digitalizes departmental records.

8.2 Limitations

- No graphical user interface.
- Single-user mode only.

8.3 Future Improvements

- GUI version.
- Cloud storage.
- Multi-user login.

---

9. Project Links / References

9.1 GitHub Repository

- git@github.com:sarthakvispute2025it/Department-Dairy-.git

9.2 Project Demo Video

- https://youtu.be/iHsae3nwKxU

9.3 Blog / Tutorial

- GeeksforGeeks Diary Management in C

9.4 References

- C Programming by Dennis Ritchie
- TutorialsPoint
- GeeksforGeeks

Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define FILE_NAME "department_diary.dat"

typedef struct {

int id;

char name[50], date[15], time[10], duration[20], address[100];

} Diary;

int next_id();

void add_entry(), view_entries(), edit_entry(), delete_entry(), follow_up();

int main() {

int ch;

do {

printf("\n=== Department Diary ===\n"

"1. Add Activity\n2. View All\n3. Edit\n4. Delete\n5. Follow-up\n6. Exit\nChoice: ");

if (scanf("%d", &ch) != 1) { printf("Invalid input!\n"); while (getchar()!='\n'); continue; }

getchar();

switch (ch) {

case 1: add_entry(); break;

case 2: view_entries(); break;

case 3: edit_entry(); break;

case 4: delete_entry(); break;

case 5: follow_up(); break;

case 6: puts("Exiting..."); break;

default: puts("Invalid choice!");
```

```
}
} while (ch != 6);
return 0;
}
int next_id() {
FILE *f = fopen(FILE_NAME, "rb");
Diary d; int id = 0;
if (!f) return 1;
while (fread(&d, sizeof(d), 1, f)) if (d.id > id) id = d.id;
fclose(f); return id + 1;
}
void add_entry() {
FILE *f = fopen(FILE_NAME, "ab"); if (!f) { printf("File error!\n"); return; }
Diary d; d.id = next_id();
printf("Activity Name: "); fgets(d.name,50,stdin);
printf("Date (DD-MM-YYYY): "); fgets(d.date,15,stdin);
printf("Time (HH:MM AM/PM): "); fgets(d.time,10,stdin);
printf("Duration: "); fgets(d.duration,20,stdin);
printf("Address: "); fgets(d.address,100,stdin);
d.name[strcspn(d.name,"\n")] = 0;
d.date[strcspn(d.date,"\n")] = 0;
d.time[strcspn(d.time,"\n")] = 0;
d.duration[strcspn(d.duration,"\n")] = 0;
d.address[strcspn(d.address,"\n")] = 0;
fwrite(&d, sizeof(d), 1, f);
fclose(f);
```

```
printf(" Activity (ID %d) added.\n", d.id);
}
void view_entries() {
FILE *f = fopen(FILE_NAME, "rb"); if (!f) { printf("No records found!\n"); return; }
Diary d;
printf("\nID | Name | Date | Time | Duration | Address\n"
"----------------------------------------------------------\n");
while (fread(&d, sizeof(d), 1, f))
printf("%-2d | %-15s | %-10s | %-8s | %-8s | %s\n",
d.id, d.name, d.date, d.time, d.duration, d.address);
fclose(f);
}
void edit_entry() {
int id, found = 0; Diary d;
printf("Enter ID to edit: ");
if (scanf("%d", &id) != 1) { printf("Invalid input!\n"); while (getchar()!='\n'); return; }
getchar();
FILE *in = fopen(FILE_NAME, "rb");
FILE *out = fopen("temp.dat", "wb");
if (!in || !out) { printf("File error!\n"); if (in) fclose(in); if (out) fclose(out); return; }
while (fread(&d, sizeof(d), 1, in)) {
if (d.id == id) {
found = 1;
printf("New Name: "); fgets(d.name,50,stdin);
printf("New Date: "); fgets(d.date,15,stdin);
printf("New Time: "); fgets(d.time,10,stdin);
```

```
printf("New Duration: "); fgets(d.duration,20,stdin);
printf("New Address: "); fgets(d.address,100,stdin);
d.name[strcspn(d.name,"\n")] = 0;
d.date[strcspn(d.date,"\n")] = 0;
d.time[strcspn(d.time,"\n")] = 0;
d.duration[strcspn(d.duration,"\n")] = 0;
d.address[strcspn(d.address,"\n")] = 0;
printf(" Updated successfully!\n");
}
fwrite(&d, sizeof(d), 1, out);
}
fclose(in); fclose(out);
remove(FILE_NAME); rename("temp.dat", FILE_NAME);
if (!found) printf("ID %d not found!\n", id);
}
void delete_entry() {
int id, found = 0; Diary d;
printf("Enter ID to delete: ");
if (scanf("%d", &id) != 1) { printf("Invalid input!\n"); while (getchar()!='\n'); return; }
getchar();
FILE *in = fopen(FILE_NAME, "rb");
FILE *out = fopen("temp.dat", "wb");
if (!in || !out) { printf("File error!\n"); if (in) fclose(in); if (out) fclose(out); return; }
while (fread(&d, sizeof(d), 1, in))
if (d.id == id) found = 1;
else fwrite(&d, sizeof(d), 1, out);
```

```
fclose(in); fclose(out);
remove(FILE_NAME); rename("temp.dat", FILE_NAME);
if (found) printf(" ID %d deleted.\n", id);
else printf("ID %d not found!\n", id);
}
void follow_up() {
puts("\n--- Follow-up Activities ---");
view_entries();
puts("\nReview complete.");
    }
```

OUTPUT:

- Choice: 1
  - Activity Name: Yoga
  - Date (DD-MM-YYYY): 2-12-25
  - Time: 9:00 AM
  - Duration: 1 Hour
  - Address: Yoga Hall
- ■ □ Activity (ID 1) added.
- Choice: 2 (View All)

ID | Name | Date | Time | Duration | Address

1 | Yoga | 2-12-25 | 9:00 AM | 1 Hour | Yoga Hall