**1. Project Title:**

**Calendar Application Using Data Structures in C**

**2. Objective:**

The objective of this project is to design and implement a calendar application in C that allows users to view a monthly calendar, add, view, and delete events for specific dates. The application utilizes fundamental data structures such as linked lists to efficiently manage and store events.

**3. Tools and Technologies:**

* **Programming Language**: C
* **Data Structures**:
  + **Linked List**: For storing events dynamically.
  + **Structs**: To organize dates and associated event lists.
* **Standard C Libraries**:
  + stdio.h for input/output operations
  + stdlib.h for dynamic memory management
  + string.h for string manipulation

**4. Features:**

1. **View Calendar**:
   * Displays a calendar for any specified month and year.
   * Accounts for leap years and correct day alignment (starting from Sunday).
2. **Add Events**:
   * Users can add events to any specific date.
   * Each date can store multiple events, which are managed dynamically using a linked list.
3. **View Events**:
   * Users can view all events scheduled for a specific date.
   * The program lists events chronologically by traversing the linked list associated with the selected date.
4. **Delete Events (Optional Enhancement)**:
   * Future improvement can allow users to delete events for specific dates.

**5. Methodology:**

1. **Data Structures**:
   * **Date Structure**: Stores information about the day, month, year, and a pointer to a linked list of events.
   * **Event Structure**: Represents an event, including a description and a pointer to the next event, forming a linked list.
2. **Functional Modules**:
   * **Calendar Display**: Uses functions to calculate the first day of the month and display the calendar correctly, including the number of days in each month and leap year handling.
   * **Event Management**: Implements adding, viewing, and traversing events using linked lists to dynamically store event descriptions.
3. **Flow of Application**:
   * The application runs a menu-driven interface where users can choose to view the calendar, add events, or view existing events.
   * Events are stored using dynamic memory allocation, ensuring flexibility in the number of events per date.

**6. Flowchart:**

**7. Project Scope:**

* **Current Scope**:
  + Displaying a basic monthly calendar for any given month and year.
  + Adding and viewing events for specific dates using a linked list.
* **Future Enhancements**:
  + Deleting events for a given date.
  + File storage for saving and loading events across sessions.
  + Adding weekly views and reminders/notifications.
  + Adding a graphical user interface using a library such as ncurses.

**8. Conclusion:**

This project demonstrates how to build a simple yet efficient calendar application using fundamental data structures in C. The use of linked lists allows for dynamic and flexible event management, while the calendar logic handles leap years and varying month lengths accurately. With further enhancements, this application can evolve into a more comprehensive scheduling tool.

**9. References:**

* "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie
* Data structures and algorithms course materials.