1. Project Title\*

\*EventPro: Event Management System\*

---

#### \*2. Project Overview\*

EventPro is a C-based application designed to streamline event management processes. It enables user authentication, event scheduling, participant registration, and real-time notifications for upcoming events. The system is intended to help organizations manage multiple events effectively while keeping track of participants and schedules.

---

#### \*3. Objectives\*

1. To provide a secure login and user authentication system.

2. To facilitate event creation and management.

3. To ensure effective notification alerts for upcoming events.

4. To maintain participant data for scheduled events.

---

#### \*4. Functional Requirements\*

- \*User Authentication\*:

- Users can sign up with a username and password.

- Users can log in to their accounts securely.

- \*Event Management\*:

- Register new events with details such as name, date, time, and description.

- Add and list participants for each event.

- Update event details as necessary.

- \*Notifications\*:

- Notify users of events scheduled within the next hour or happening on the current day.

- \*Data Storage\*:

- User accounts and event details are stored in plain text files for simplicity.

---

#### \*5. Non-Functional Requirements\*

- \*Performance\*:

- Handle up to 100 participants per event.

- Efficiently manage up to 50 events per user.

- \*Security\*:

- Basic encryption (future scope) for sensitive user data such as passwords.

- \*Scalability\*:

- Ability to increase the number of users by scaling file-based storage.

---

#### \*6. System Architecture Overview\*

The system operates in a modular manner, with the following key components:

1. \*Login Module\*: Handles user sign-up and authentication.

2. \*Event Management Module\*: Allows event creation, updates, and participant management.

3. \*Notification Module\*: Monitors event timings and generates notifications.

---

#### \*7. Technologies Used\*

- \*Programming Language\*: C

- \*File Handling\*: For storing user and event data.

- \*Time Functions\*: For handling event scheduling and notifications.

---

#### \*8. Data Requirements\*

- \*User Data\*:

- Stored in users.txt (username, password).

- \*Event Data\*:

- Stored in <username>.txt for each user, including event details and participant data.

---

#### \*9. Assumptions and Constraints\*

- Assumption: The user system runs in a local environment and does not require internet connectivity.

- Constraint: Plain text storage is used for simplicity instead of a database.

---

#### \*10. Timeline and Milestones\*

- \*Phase 1\*: Requirement gathering and design – (Completed).

- \*Phase 2\*: Development – (In progress).

- \*Phase 3\*: Testing – (Planned for [specific date]).

- \*Phase 4\*: Final presentation and submission – (Planned for [specific date]).

---

#### \*11. References\*

- Standard C programming textbooks and resources.

- Documentation for the time.h library.

SCREENSHOT OFINPUT ANDOUTPUT FOR REFRENCE

