**Updated Project Overview and Features**

**1. User Management:**

* **Sign Up/Login**: Users can register and log in using email/password or social media authentication.
* **User Profile**: Users create a profile with basic information such as name, age, interests, and a profile picture.
* **Interest Tags**: Users can select from a predefined set of interest tags that will help personalize event suggestions.

**2. Event Creation and Management:**

* **Create Event**: Users can create events with the following fields:
  + Name
  + Description
  + Privacy settings (Open/Private)
  + Age range (start and end)
  + Slots available
  + Event start and end time
  + Category (e.g., sports, games, networking)
  + Motivation (reason for organizing the event)
* **Event Listing**: Events are listed and searchable on the platform:
  + **Public Events**: Users who meet the criteria (e.g., age) can directly join.
  + **Private Events**: Events are visible to all users, but joining requires sending a request, which must be approved by the event organizer.

**3. Event Discovery and Interaction:**

* **Event Cards**: Events are displayed as cards in a dynamic discovery feed. Each card shows key information such as the event name, date, category, location, and a brief description. Users can interact with these cards using various actions:
  + **Like**: Users can like an event to show interest, which can notify the organizer and also appear in the user’s list of liked events.
  + **Bookmark**: Save events for later review or decision-making. These bookmarked events can be accessed from a dedicated section.
  + **Join or Request to Join**: Directly join public events if criteria are met. For private events, users can request to join, which sends a notification to the event organizer for approval.
* **Explore Feed**: A scrollable feed of events tailored to the user’s interests and past participation. This feed can include:
  + **Recommended Events**: Based on user interests, location, and behavior.
  + **Popular Events**: Events with high participation or interest.
  + **Nearby Events**: Events happening close to the user’s current location.

**4. Advanced Search and Filtering:**

* **Search Events**: Users can search for events using keywords.
* **Filter Events**: Filters can include categories, location, date, age range, and interest tags to help users find specific events.
* **Dynamic Sorting**: Users can sort events by relevance, popularity, date, or distance.

**5. Social Interaction and Engagement:**

* **Notifications**: Users receive notifications for various interactions, such as:
  + Event invitations.
  + When someone likes or shows interest in an event they created.
  + Approval or rejection of join requests for private events.
  + Reminders for upcoming events they are attending.
* **Comments and Discussions**: Allow users to comment on event pages, ask questions, or start discussions, fostering interaction around the event.
* **See Who’s Interested**: Show a list of users who have liked or shown interest in the event. This helps create a sense of community and encourages more participation.
* **Event Follow-Up**: After an event, suggest similar events based on the user’s participation, helping maintain ongoing engagement.

**6. Feedback and Rating:**

* **Rate Events**: Users can rate events after attending them. These ratings can be used to improve recommendations and highlight high-quality events.
* **Feedback System**: Allow users to provide feedback or reviews on events and organizers, which can help improve future events.

**System Architecture (Updated)**

* **Frontend (React.js)**:
  + **Components**: Build components for user profiles, event creation, event discovery feed, event detail views, notifications, and comments.
  + **State Management**: Use Redux or Context API to manage state for user profiles, event data, likes, bookmarks, and notifications.
  + **Responsive Design**: Ensure the UI is responsive, providing a seamless experience across web and mobile devices.
* **Backend (Node.js/Express)**:
  + **API Endpoints**: Develop RESTful API endpoints for handling user registration, event creation, event listing, likes, bookmarks, join requests, and notifications.
  + **Real-Time Updates**: Use WebSockets or libraries like Socket.IO for real-time updates, especially for notifications and chat functionalities.
  + **Matching and Recommendation Engine**: Implement algorithms to handle event recommendations based on user interests, behaviors, and interaction history.
* **Database (SQL Server)**:
  + **Tables**: Create tables for users, events, likes, bookmarks, join requests, notifications, comments, and interest tags.
  + **Relationships**: Define relationships between users and events (creator, participant, interested), users and likes/bookmarks, and event comments.
  + **Optimization**: Use indexing and query optimization to handle complex queries for event discovery and recommendation.

**Development Phases (Updated)**

1. **Phase 1: MVP (Minimum Viable Product)**
   * Implement basic user registration, login, and profile creation.
   * Develop core event creation, listing, and browsing functionalities.
   * Allow users to like, bookmark, join, and request to join events.
   * Set up notifications for basic interactions (join requests, approvals).
2. **Phase 2: Advanced Features**
   * Build the dynamic discovery feed with personalized event suggestions.
   * Implement advanced search and filtering options.
   * Introduce comment and discussion features on event pages.
   * Develop the feedback and rating system.
3. **Phase 3: Social Interaction and Optimization**
   * Add social interaction features like seeing who’s interested in an event.
   * Enhance notification and real-time update capabilities.
   * Optimize performance and scalability for handling a large number of users and events.
4. **Phase 4: Mobile Integration**
   * Develop a mobile app using React Native, leveraging shared code from the web version.
   * Implement mobile-specific features such as push notifications for real-time updates.
   * Ensure seamless synchronization between web and mobile platforms.