Approach:

To train an open-source LLM (Large Language Model) for event recommendation, we follow a structured methodology:

1. **Dataset Collection**:

   - Scrape and collect event-related data from platforms like Eventbrite, Meetup, and local event websites.

   - Gather user preferences through surveys, past event interactions, and historical data.

   - Use APIs (e.g., Ticketmaster, Google Events) for real-time event listings.

2. **Data Preprocessing**:

   - Clean and normalize event descriptions, dates, locations, and categories.

   - Use NLP techniques (tokenization, stemming, stopword removal) to enhance event text representation.

   - Create embeddings using models like Word2Vec, TF-IDF, or BERT to capture semantic meaning.

3. **Model Selection & Training**:

   - Fine-tune an open-source LLM (e.g., LLaMA, Mistral, Falcon) using instruction tuning.

   - Implement Reinforcement Learning with Human Feedback (RLHF) to improve user preference understanding.

   - Train the model to generate personalized recommendations based on input prompts.

4. **Integration with Chat Platforms**:

   - Deploy the trained model as an API using FastAPI or Flask.

   - Connect the model to WhatsApp/Telegram via Twilio or Telegram Bot API.

   - Enable real-time notifications and conversational event discovery.

5. **Continuous Improvement**:

   - Collect user feedback to refine recommendations.

   - Update datasets regularly for improved accuracy.

   - Implement A/B testing to evaluate different recommendation strategies.

Challenges Faced:

1. **Data Availability & Quality:**

   - Many event platforms restrict API access or require paid plans.

   - User preferences may be diverse and require extensive labeling.

2. **LLM Fine-tuning Complexity:**

   - Requires significant computational resources.

   - Balancing generalization and personalization is challenging.

3. **Latency & Real-time Processing:**

   - Ensuring fast response times for chatbot interactions.

   - Handling high query loads efficiently.

4. **User Privacy & Data Security:**

   - Storing user preferences securely.

   - Preventing unauthorized access to recommendation data.

Improvements & Future Work:

1. **Hybrid Recommendation System:**

   - Combine LLM-based recommendations with collaborative filtering techniques.

   - Use graph-based approaches to find event relationships.

2. **Better Context Understanding:**

   - Implement memory-based techniques so the bot remembers user preferences.

   - Use real-time feedback loops for refinement.

3. **Multi-Modal Event Discovery:**

- Integrate images, videos, and social media insights.
- Provide richer event descriptions for users.

4. **Scalability & Optimization:**
   - Use model distillation to deploy lightweight LLM variants.
   - Optimize database queries for better event search performance.

This approach ensures that users receive highly personalized and timely event recommendations, enhancing their overall experience.