



AI Python coding challenge

Our team is currently using AI coding agents to implement and test most of the system under development. We want to assess you on how effectively you use AI tools to implement high quality code at speed

Implement `merge_user_events(events)`.

Input

- events: a list of dictionaries, each like:

```
{"user_id": "u1", "ts": 1700000000, "type": "click", "meta": {"page": "/"}}
```

Output

Return a list of merged events, where events are grouped by `user_id` and then combined into "sessions".

Rules

1. A "session" is a consecutive run of events for the same `user_id` where adjacent events are at most 10 minutes apart (≤ 600 seconds).
2. Each output session must be a dict:

```
{  
  "user_id": "...",  
  "start_ts": ...,  
  "end_ts": ...,  
  "types": [...],  
  "meta": {...}  
}
```

3. types must include event types in chronological order with duplicates removed, but the count of each type must be preserved.
4. meta should be the deep merge of all meta objects in the session:

- If both have a key with dict values, merge recursively.
 - If there's a conflict on a non-dict value, keep the earliest value.
5. The output list must be sorted by start_ts ascending across all users.
 6. Input guarantee: events may be unsorted, and you must not modify events in-place.

Example

If two events for the same user occur at t=1000 and t=1500, they're in the same session (500s gap). If the next is at t=2200, it starts a new session (700s gap).

Once completed, please email back to schedule a call for the interview.