Project #3 – Redis / Key-Value Stores
Due Date: May 14th, 2023

Assume a Teams/Zoom-like environment. Admins set up meetings (one-off or recurring).

A database of:

- users (userID, name, age, gender, email)
- meetings (meetingID, title, description, isPublic, audience)
- meeting_instances (meetingID, orderID, fromdatetime, todatetime)

A database of eventsLog (event_id, userID, event_type, timestamp)
- event_type can be 1 (join_meeting), 2 (leave_meeting), 3 (timeout)

A meeting either has audience (a list of emails) or it is public.

Scheduler: a meeting instance becomes active when it is due (scan database every 1 minute to "activate"/"deactivate" meeting instances).

Use Python/Java that connects to Redis to implement the following functions:

* Function: a user joins an active meeting instance – if it is public, always, otherwise only if s/he is allowed, i.e. his email is in audience (➔ eventsLog is updated)
* Function: a user leaves a meeting that has joined (➔ eventsLog is updated)
* Function: show meeting's current participants
* Function: show active meetings
* Function: when a meeting ends, all participants must leave (➔ eventsLog is updated)
* Function: a user posts a chat message
* Function: show meeting's chat messages in chronological order
* Function: show for each active meeting when (timestamp) current participants joined
* Function: show for an active meeting and a user his/her chat messages