-create in nodejs simple version of progressive jackpot server (typescript preferable, but javascript is ok too) and its database (postgres)

-provide:

architecture diagram with short explanation

database model

api explanation

source code with running scripts (can be in localhost, doesn't need to be deployed to remote server)

# GAME

-progressive jackpot is giving its win to specific game/player in any random moment, but prerequisite is that "seed" is filled

-there is actual amount and displayed amount, best is that player don't recognize that actually its starting from zero after providing of jackpot win

-jackpot levels current amounts can be seen in any moment on client side, not only as a response to spin

-there should be remote game server with game logics instances, but for purpose of this task, any kind of several clients can ontribute to progressive jackpot by sending wagers and neccessary info to it and maybe getting of win

2% of game bet should go to jackpot wager

2% of bet is distriburerd to 4 levels

seeds for levels are { 1k,10k,100k, 3000k € }

but all of this configs should be dynamic, flexibile and in database config

all jackpot transactions should be stored

and in case of restart of jackpot server, it should be possible to recover last situation

also, for example, jackpot server can handle several jackpot instances which is shared between several game logics instances on its client

there should be command for forcing of win for some specific player

Jackpot displayed amounts for all 4 levels are visible on the client (or should I have a separate client for each level and display only that level’s amount)? Client sends a certain bet (wager) and level to the server by spinning. 2% from the bet is added to the seed for the chosen level and 2% to the level’s jackpot(actual amount?). After the seed lower boundary is reached, the players that are playing at the moment become eligible for winning the jackpot.

Purpose of this test is not to provide commercial application,

so any runnable solution is valid