Instructions

1. You have 5 days to implement the solution.
2. Your solution must build+run on Linux.
3. You are free to choose any tech stack but language must be Java
4. Your code will be evaluated for both functional and non-functional requirements such as code quality, ease of setting up dev and production environment, load testing scripts etc.
5. Automated tests are mandatory, so please include tests/specs. Additionally, it's a huge plus if you test drive your code.
6. We are really, really interested in a clean and maintainable codebase, so please solve the problem keeping this in mind.
7. Please ensure that the coding conventions, directory structure and build approach of your project follow the conventions set by popular open source projects in the language that you're using.
8. When implementing this solution, please use Git for version control. We expect you to send us a zip/tarball of your source code when you're done that includes Git metadata (the .git folder. in the tarball so we can look at your commit logs and understand how your solution evolved.
9. You should provide the architecture diagram of your solution

README.md

README for your project should include

1. A brief explanation of why a particular tech stack was chosen Infrastructure requirements for running your solution
2. Setup instructions, automated deployment of the program and dependencies to development and test environment is a plus

The Bidding System

1. It allows administrators to start the auction for a defined duration.
2. It allows users to participate in placing the bid for running auctions.
3. Auction automatically over once the auction duration is complete and after that no user can place the bid.
4. After completion (auction over) of the bidding, this system announces the winner of the auction.

An Auction have following attributes:

1. **Item Code -** for which auction is running
2. **Minimum Base Price -** This is starting bidding amount, no user can place the bid lesser than this defined price
3. **Step Rate -** minimum amount difference b/w two consecutive bids. For example, if a user placed the bid of 1000 /- INR then the next acceptable bid will be a minimum of

1000 + Step Rate. If the step rate is 250 /- INR then the next acceptable bid should be >= 1250.

4. **Start Time -** Start time of Auction
5. **Duration -** For how long the auction is supposed to be in RUNNING state.
6. **Status -**
    a. **RUNNING:** Only running auctions are the candidates of placing the bid
    b. **OVER:** Once auction is over then no user can place the bid on the corresponding item
7. **User Bids -** All user bids should be recorded whether it was accepted or rejected.

Scope of the Assignment

In Scope

1. Start the Auction
2. Stop Auction once duration is over
3. Fetch running auctions through API
4. Place the bid for running auctions through API
5. Only logged in users should be allowed to place the bid

Out of scope

1. User Onboarding
2. Login and Logout the users

Problem Statements

Start the Auction

Administrators should be able to start the auction for an item through API.

Input Parameters

1. Item Code
2. Base Price
3. Step Rate
4. Duration

Criteria for valid auction

1. Item should be available in system
2. There shouldn't be any running auction on the same item i.e. only one auction can be in running state for an item.

Stop Auction

Auction should go in OVER state once the duration is complete.

Fetch Running auctions

GET /auction?status=RUNNING

Response should contain the following attributes

1. Item Code
2. Highest bid amount placed
3. Step Rate

API should be paginated in nature. And you are free to change API structure based on your design.

Place Bid

1. You are free to change API structure based on your design.
2. API must be able to handle the concurrency/race condition of bids for an auction.

POST /auction/{itemCode}/bid
Request Payload: bidAmount

Where:

1. itemCode: Item Code of the item
2. bidAmount: amount placed by user

Response

1. 201 - Bid is accepted
2. 404 - Auction not found
3. 406 - Bid is rejected

Good to Have
Generally users will be placing the bid through a website or app and hence the system should be able to notify all users about the last highest placed bid.

Announce Winner

As soon as the auction is over, system should be able to notify the winner over the email.