

## Take Home Assignment

### Bidding Application Backend

#### Expected Features

- Create, View, and Close Listings - Users should be able to create listings for items, view active listings, and close their own listings when bidding is complete.
- Real-time updates - Implement real-time updates for the current bid amount and the count of bidders using WebSockets.
- Room Management - Each bid should have its own dedicated room. When a user places a bid, they join the corresponding room to receive updates specific to that bid.
- Authentication - Only authenticated users should be able to create a bid.
- Authorization - Only the creator of a listing should be able to close the bid and have access to the list of bidders.

#### Evaluation Criteria

- Adherence to requirements
- Project Architecture and Design
- Schema and Table Design
- Input Validation and Error Handling
- Following Security Best Practices
- Real-time communication
- Configuration and Environment Variable management
- Project Documentation
- Version Control
- Typescript Usage

#### Suggestions

- Database - use PostgreSQL and any ORM of your choice
- Real-time communication - use [Socket.IO](#)
- Follow [Conventional Commits](#)
- Documentation should cover environment setup, project setup, architecture and tools used

#### Submission Requirements

- The project should be submitted within 2 days
- The source code should be on a private repository with access provided to reviewer(s)
- After 2 days have passed, any changes to the source code is disregarded(i.e. any commit past the two day mark is not reviewed or accepted)
- The candidate should privately share credentials and/or environment variables required to setup and run the project by the reviewer(s)
- Attach a link to a recording (up-to 5 minute long) that
  - Outlines Project Architecture
  - Demos the final outcome

\* We suggest you use [loom](#) for the video recording