Thank you for your dedicated time to complete Aardvark Technologies frontend task.

THE TASK: You will need to create an application for roulette screen that would show upcoming and history roulette spins using Vue js. Unstoppable application workflow is a key when developing this screen as it could be loaded once on the screen and presented for clients who have no interaction with it.

Note: Task implementation should be placed in the git repository.

Evaluation: Your result will be evaluated in the following parts.

1. Repository usage (no unused packages/code should be left behind).
2. File structure.
3. Readability.
4. Code splitting where it makes sense.
5. Good practices.
6. Unstoppable application workflow(temporary internet connection problems should not stop this application).
7. Basic knowledge and use of UI/UX principles(it doesn't need to be super pretty, but please think about how to present the data to the user in a clear way).

The following UI elements should be implemented in the Roulette screen application:

1. **API URL input field:** There should be a reactive input field for that. (https://dev-games-backend.advbet.com/v1/ab-roulette/)
   This backend URL should be used to make all requests to API.
2. **Statistics block:** last 200 roulette spins (how many times roulette landed on particular numbers). Information needed can be retrieved using /{wheelID}/stats endpoint. These statistics should be used for informational purposes showing the number of occurrences for this particular result.
   - **HOT/COLD/NEUTRAL**(5 numbers with most/least/others hits) numbers from statistics should be marked differently so clients could identify them.
   - All roulette numbers should be painted accordingly using information from /{wheelID}/configuration endpoint for example 0 should be green, others black and red.Colors for numbers should be taken from /{wheelID}/configuration **colors** property.
3. **Upcoming spin timer:**
   Helpful notes:
   - Start from getting the next game /{wheelID}/nextGame
   - Use next game **fakeStartDelta** property value to show timer until next spin will take place
   - **startDelta** indicates When results should be ready.
   - Spinning should take place until result is present from /{wheelID}/game/{uuid}
   - Display next game countdown
4. **Spin history:**

- Display of last resulted spins that happened from application start.

5. **Game Board**: wheel or board that would display all available numbers in european wheel sequence order which can be found using /{wheelID}/configuration **positionToId** property. Also after the result for current spin is available via /{wheelID}/game/{uuid} endpoint it should somehow indicate on the board for a few seconds(for example blink).
6. **Actions Log:** show action log of all what happens with timeStamp. For example:
   - 2020-10-15T08:54:00.804Z Loading game board
   - 2020-10-15T08:54:00.805Z GET .../configuration
   - 2020-10-15T08:54:00.805Z Checking for new game
   - 2020-10-15T08:54:00.806Z GET .../nextGame
   - 2020-10-15T08:54:00.887Z GET .../stats?limit=200
   - 2020-10-15T08:54:00.887Z sleeping for fakeStartDelta 69 sec
   - 2020-10-15T08:55:09.888Z Spinning the wheel
   - 2020-10-15T08:55:09.890Z GET .../game/1552694
   - 2020-10-15T08:55:09.941Z Still no result continue spinning
   - 2020-10-15T08:55:10.943Z Spinning the wheel
   - 2020-10-15T08:55:10.943Z Wheel is already spinning ;\
   - 2020-10-15T08:55:10.943Z GET .../game/1552694
   - 2020-10-15T08:55:10.993Z Still no result continue spinning
   - 2020-10-15T08:55:11.995Z Spinning the wheel
   - 2020-10-15T08:55:11.995Z Wheel is already spinning ;\
   - 2020-10-15T08:55:11.995Z GET .../game/1552694
   - ...

Extra requirements for Mid-Senior:

1. **All above.**
2. **Use typescript.**
3. **Place content to different routes(API URL** should be seen all the time**):**
   - 3.1. **Game Board** and **Spin history.**
   - 3.2. **Statistics** and **Actions Log.**

   Important:

   TLDR: For both routes, data should be updated automatically, without the user even being in that particular route view.

   In both of these routes data on the screen should be updated after a new result was retrieved from the backend. E.g.: If you are in 'Statistics and **Actions Log' statistics should be updated automatically after new roulette spin data is available. If you return to the 'Game Board** and **Events' view, all events that happened while you were in 'Statistics** and **Actions Log' should still be visible in Events log. Also please spend a bit of time to think about how this could be achieved without code duplication.**

Backend API documentation:
http://petstore.swagger.io/?url=https://dev-games-backend.advbet.com/v1/ab-roulette/static/swagger.yaml

Backend Address (for requests use wheelID=1 value)**:**
https://dev-games-backend.advbet.com/v1/ab-roulette/

Already completed Task example:
https://origin.advbet.com/static.advbet.com/frontend-task/

Please share bitbucket/github repository access with:

1. **julius.kriukas@adv.bet**
2. **marijus.zelba@adv.bet**
3. **edvinas.bazevicius@adv.bet**
4. **zygimantas.drukteinis@adv.bet**
5. **marius.viliokaitis@adv.bet**