Technical Test For Backend Engineer

Let's design an API for a taxi service web app that includes a bidding feature. In this scenario, clients can make ride requests with their proposed prices, and fleets can view these requests and place bids on them.

Context:

You are developing an advanced API for a taxi service with a unique bidding system. Clients can submit ride requests with their proposed prices, and fleets can view these requests and make bids to provide the service.

Feature List:

1. **Request a Ride:**
    - Allow clients to request a ride by providing details such as pickup location, drop-off location, and their proposed price.
    - Verify that the response confirms the successful ride request.
2. **View Ride Requests:**
    - Allow fleets to view a list of ride requests from clients, including details like pickup/drop-off locations, date-time and proposed prices.
    - Verify that the response contains details of available ride requests.
3. **Make Bid on Ride:**
    - Allow fleets to place bids on a specific ride request
    - Verify that the response confirms the successful bid placement.
4. **View Bids on Ride:**
    - Allow clients to view bids received for their requested ride.
    - Verify that the response contains details of bids received on the ride.
5. **Accept Bid:**

- Allow clients to accept a bid for a specific ride request, closing the bidding process.
- Verify that the response confirms the successful bid acceptance.

Database:

- Our preference is MongoDB, yet feel free to select any database that aligns with your preferences. (Utilize a Docker image for the database.)

Running Locally:

- Dockerize the application and Database for seamless local execution.

IAC:

- Implement a simple Infrastructure as Code (IAC) solution for this service.

Expectations:

- Utilize TypeScript and Node.js in the implementation.
- Ensure the API responses align with the described functionality.
- Implement at least three features from the provided feature list.
- Dockerize the application.
- Ensure to include necessary documentation for your code in the README file.

Example Data:

```
{
  "clients": [
    {
      "id": "client1",
      "name": "John Doe",
      "email": "john.doe@example.com",
      "phone": "+1234567890"
```

```
    },
    {
      "id": "client2",
      "name": "Jane Smith",
      "email": "jane.smith@example.com",
      "phone": "+9876543210"
    },
    {
      "id": "client3",
      "name": "Alex Johnson",
      "email": "alex.johnson@example.com",
      "phone": "+1122334455"
    },
    {
      "id": "client4",
      "name": "Emily White",
      "email": "emily.white@example.com",
      "phone": "+9988776655"
    },
    {
      "id": "client5",
      "name": "David Brown",
      "email": "david.brown@example.com",
      "phone": "+5544332211"
    }
  ]
}


// Fleets

{
  "fleets": [
    {
      "id": "fleet1",
      "name": "City Cabs",
      "email": "citycabs@example.com",
      "phone": "+1112223333"
```

```
    },
    {
      "id": "fleet2",
      "name": "Quick Rides",
      "email": "quickrides@example.com",
      "phone": "+4445556666"
    },
    {
      "id": "fleet3",
      "name": "Metro Taxis",
      "email": "metrotaxis@example.com",
      "phone": "+7778889999"
    },
    {
      "id": "fleet4",
      "name": "Swift Transits",
      "email": "swifttransits@example.com",
      "phone": "+1234567890"
    },
    {
      "id": "fleet5",
      "name": "Express Cars",
      "email": "expresscars@example.com",
      "phone": "+9876543210"
    }
  ]
}

// Rides requests with bids and without bids
{
  "rides": [
    {
      "id": "ride1",
      "clientId": "client1",
      "pickupLocation": "123 Main St",
      "dropoffLocation": "456 Elm St",
      "proposedPrice": 20,
      "bids": [
```

```
        {
          "id": "bid1",
          "fleetId": "fleet1",
          "bidAmount": 18
        },
        {
          "id": "bid2",
          "fleetId": "fleet2",
          "bidAmount": 22
        }
      ]
    },
    {
      "id": "ride2",
      "clientId": "client2",
      "pickupLocation": "789 Oak St",
      "dropoffLocation": "101 Pine St",
      "proposedPrice": 25,
      "bids": [
        {
          "id": "bid1",
          "fleetId": "fleet1",
          "bidAmount": 23
        },
        {
          "id": "bid2",
          "fleetId": "fleet2",
          "bidAmount": 26
        },
                {
          "id": "bid3",
          "fleetId": "fleet4",
          "bidAmount": 25
        }
      ]
    },
    {
      "id": "ride3",
```

```
      "clientId": "client3",
      "pickupLocation": "456 Elm St",
      "dropoffLocation": "789 Oak St",
      "proposedPrice": 18,
      "bids": []
    },
    {
      "id": "ride4",
      "clientId": "client4",
      "pickupLocation": "101 Pine St",
      "dropoffLocation": "123 Main St",
      "proposedPrice": 30,
      "bids": []
    }
  ]
}
```