



Himanshu Verma &lt;himanshuverma9170@gmail.com&gt;

## GoQuant Front-End Application

1 message

**GoQuant** <careers@goquant.io>

Reply-To: careers@goquant.io

To: himanshuverma9170@gmail.com

Mon, Nov 3, 2025 at 11:00 PM

Hello,

Thank you for applying to join our team at GoQuant. After a thorough review of your application, we are pleased to inform you that you have been selected to move forward in our recruitment process.

As the next step, we ask you to complete the following assignment. This will provide us with a deeper understanding of your skills and how well they align with the role you have applied for. Please ensure that you submit your completed work within 7 days from when you received this email.

### **Assignment Title:** Latency Topology Visualizer

#### **Objective**

Create a Next.js application that displays a 3D world map visualizing exchange server locations and real-time/historical latency data across AWS, GCP, and Azure co-location regions for cryptocurrency trading infrastructure.

#### **Where to Start**

1. Create a Next.js project if you don't already have one set up.
2. Use a 3D mapping library (e.g., Three.js, Mapbox GL JS, or similar) for creating the 3D world map visualization.
3. Implement real-time latency monitoring using any open API that provides network latency data (e.g., Cloudflare Radar, Pingdom, or similar). Ensure the service is unpaid/free for demo purposes.

#### **Functional Requirements**

1. 3D World Map Display
  - Render an interactive 3D world map.
  - Display exchange server locations as 3D markers on the map.
  - Allow users to rotate, zoom, and pan the 3D map for exploration.
  - Implement smooth camera controls and transitions.

2. Exchange Server Locations

- Plot major cryptocurrency exchange server locations (e.g., OKX, Deribit, Bybit, Binance, etc.).
- Display server information on hover/click (exchange name, location, cloud provider).
- Use different visual markers or colors to distinguish between AWS, GCP, and Azure hosted servers.
- Show a legend explaining the marker types and colors.

### 3. Real-time Latency Visualization

- Create animated latency connections between exchange servers and cloud regions.
- Display real-time latency values as animated data streams or pulse effects.
- Update latency data at regular intervals (e.g., every 5-10 seconds).
- Use color-coded connections to represent latency ranges (green for low, yellow for medium, red for high).

### 4. Historical Latency Trends

- Implement a time-series chart showing historical latency data for selected server pairs.
- Allow users to select specific exchange pairs or regions to view historical trends.
- Display latency statistics (min, max, average) for the selected time period.
- Include time range selectors (1 hour, 24 hours, 7 days, 30 days).

### 5. Cloud Provider Regions

- Visualize AWS, GCP, and Azure co-location regions on the 3D map.
- Show region boundaries or clusters with distinct visual styling.
- Display region information including provider name, region code, and server count.
- Implement filtering options to show/hide specific cloud providers.

### 6. Interactive Controls

- Add a control panel for filtering by exchange, cloud provider, or latency range.
- Implement search functionality to quickly locate specific exchanges or regions.
- Add toggle switches for different visualization layers (real-time, historical, regions).
- Include performance metrics dashboard showing current system status.

### 7. Responsive Design

- Ensure the application is responsive and user-friendly across various screen sizes.
- Optimize 3D rendering performance for mobile devices.
- Implement touch controls for mobile interaction with the 3D map.

## Bonus Features

- Implement latency heatmap overlay on the 3D map surface.
- Add network topology visualization showing connection paths between exchanges.
- Create animated data flow visualization showing trading volume or order flow.
- Implement dark/light theme toggle for better user experience.
- Add export functionality for latency reports and visualizations.

## Technical Requirements

- Use TypeScript for type safety and better code organization.
- Implement proper error handling and loading states.
- Optimize 3D rendering performance.
- Use modern React patterns (hooks, context, etc.) for state management.
- Implement proper data caching and state management for real-time updates.

## Submission Guidelines

- Provide a video recording demonstrating the functionality and reviewing your code.
- Include instructions on how to run the project locally.
- Share a link to the GitHub repository with your submission.
- Document any assumptions made and libraries used in your README file.

## Evaluation Criteria

- Code quality and organization
- 3D visualization implementation and performance
- Real-time data integration and updates
- User interface design and interactivity
- Responsive design and mobile optimization
- Bonus features implementation (if any)
- Performance optimization and best practices

Any questions you are uncertain of must be researched by you, with any resources at your disposal for use.

## INSTRUCTIONS FOR SUBMISSION - MANDATORY FOR ACCEPTANCE

- **SUBMIT THE ASSIGNMENT VIA EMAIL TO: [careers@goquant.io](mailto:careers@goquant.io) AND CC THE FOLLOWING [jennifer.carreno@goquant.io](mailto:jennifer.carreno@goquant.io).**
- **ATTACH YOUR RESUME TO THE EMAIL**
- **PROVIDE VIDEO DEMONSTRATION EXPLAINING YOUR CODE**
- **PROVIDE DOCUMENTATION OF YOUR CODE**
- **PLEASE SET THE SUBJECT OF THE EMAIL AS THE ASSIGNMENT TITLE**

Upon receiving your submission, our team will carefully review your work. Should your assignment meet our expectations and demonstrate the skills we are looking for, we will move forward with the final steps, which could include extending a formal offer to join our team.

We look forward to seeing your work and wish you the best of luck in this important stage of the process.

Best regards,  
GoQuant Team

**Confidentiality Notice (PLEASE READ)**

The contents of this assignment and any work produced in response to it are strictly confidential. This document and the developed solution are intended solely for the GoQuant recruitment process and should not be posted publicly or shared with anyone outside the recruitment team. For example: do not publicly post your assignment on GitHub or Youtube. Everything must remain private and only accessible to GoQuant's team.

GoQuant

[150 SE 2nd Avenue, Suite 401, Miami, FL 33131](https://150 SE 2nd Avenue, Suite 401, Miami, FL 33131)

[Unsubscribe](#)