

## TECHNOLOGY DETAILS

- Node
- Express
- Angular
- MongoDB
- Kotlin
- Swift
- Agora SDK
- DeepAR
- AWS

# CASE STUDY ON LIVE STREAMING AND GAMING APPLICATION

## OVERVIEW

The client wants to build a SAAS model-based project where people can interact with each other through live streaming. Users can go live and showcase their talent while others can join them as an audience, rate them, gift them and connect with them.

## CLIENT DETAILS

Name: Confidential  
Location: UAE  
Industry: Social media,  
Entertainment

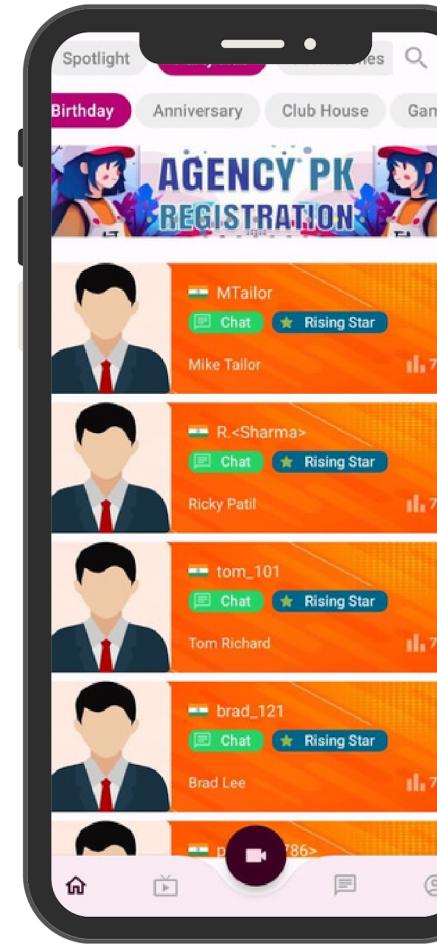
## PROJECT ARCHITECTURE



# SOLUTION

This application has mostly 4 user roles. Agent, Host, Viewer, and Admin.

- **Admins** onboard new agencies/organizations and hosts.
- **Agents** have control over the agency. She can help onboard new hosts. Can monitor the live sessions. Admin also makes the initial setup changes such as agency information, host approvals, etc. They validate hosts.
- **Hosts** can create new live sessions at their will, making it public or private, allow/disallow viewers, or members(viewers who want to go live with the host), create new matches, etc. Once the host creates the match, it sends the notification to the relevant opponents.
- **Viewers** can view live streaming of hosts, participate in pk matches, rooms, and parties, and can connect with people around the world. They can see the upcoming sessions, parties, and matches of their favorite hosts. They can play games and earn coins. And also gift coins to hosts and users during live sessions, and matches. Viewers can also request to become hosts and then can be approved/rejected by admins.



## PROJECT SCREENSHOTS

