GyeongHo Kim

I Build
Web Experience

Gyeongho Kim
https://gyeongho.dev
https://github.com/gyeonghokim

Web Frontend engineer with expertise in web components, real-time video systems, and design systems

(+82)10-3812-5469
rlarudgh2017@gmail.com

Skills

Component Based Develop(Lit, Web Components, React)

Component Test, Visual Regression Test(Cypress, WTR, BackstopJS)

Everything for DX(Design System, WEB & Node.js video player libraries)

Experience

IDIS / Web Engineer
July 2023 - PRESENT, Pangyo

Developed Web UIs for the **Video Management System (VMS)** in the development department of a company manufacturing NVR/DVR systems utilizing their recorders and CCTV.

Introduced es module and a component-based development approach.
Created a **Design System** using **Web Component** and **StoryBook Server**.
Integrated **component testing, visual regression testing** into the Jenkins CI pipeline.
Developed several web solutions using the design system I built.

Built a **REAL TIME WEB VIDEO PLAYER** that parses RTP packets in the browser environment, decodes H264 NAL units, and plays real-time video. It also supports HLS and WebRTC.
Developed an **RTSP library** that operates in a Node.js environment.

Project

iNEX Web Components
November 2023 - PRESENT

Utilizing our recorder, CCTV, we created a library of web components for a design system that is being applied to all of our serviced web solutions. I had the ideal experience of the product growing with me. I paid attention to ShadowDOM and theme tokens to reflect OEM-specific themes.

Lit, Storybook, Design System(Typography, Theme Color, Grid System)

1. Introduced the concept of design systems and created a web component library.
2. Introduced concept of theme using shadowDOM and tokens for productivity for OEM-specific designs.
3. Build an internal NPM registry server and StoryBook server.
4. Implemented component testing with Cypress and visual

regression testing with BackstopJS, and integrated them into the CI pipeline
5. Introduced importMap, grunt minifying, and the rollup bundler for compatibility with existing pages, improving the initial load time of the legacy pages from over one minute to less than three seconds.

iNEX Live Player

February 2024 - June 2024

A library designed to display RTSP streams on the web.

Lit, Nest.js, Gin, FFMPEG, Live-streaming(RTP, HLS, WebRTC)

1. Developed a proxy server using Node.js or Golang for RTSP connection and extracting Video NALu.
2. Created a Web Component to decode NALu using WebCodecs API or a WebAssembly decoder and render the video.
3. Added Worker Thread support for video decoding to display multiple CCTV streams on a single screen.
4. Developed additional support for WebRTC, HLS, and LL-HLS protocols.

Clip Archive

May 2023 - PRESENT

When buses equipped with CCTV reach the depot, recorded videos are uploaded to a recording server. The system allows users to check the status of the recorder, search, manage, and create short clips under 30 minutes via the web interface. Users can view the clip creation status, check detailed information, add tags and notes, and view logs. The system also supports downloading clips and printing logs.

React, Tanstack Query, React Router Dom, FSD

1. Developed the WEB UI
2. Focused on domain separation and abstraction as I wanted to add GPS location information for the clips.
3. Abstracted the apiClient due to frequent changes in both our protocol and the HTTP protocol, and separated the Entity, DTO, and RDO for the frequently changing API.

Real-time Flood Monitor

February 2024 -July 2024

A project where users can select a region on a map UI to monitor the status of each CCTV and click markers to play the CCTV video. It also enables remote operation of road barriers via open/close buttons when the water level is high.

Lit, Redux, Gin

1. Developed the WEB UI.
2. Developed a Golang Gin-based media server that receives video via RTP and transmits it via various protocols (WebRTC, HLS,

ll-HLS, RTP).

Awards

Grand Prize in Hackerton
December 2023, Namuthon hosted by Google Developer Student Clubs & alpaco

Participated in a hackathon where teams developed services using Google technologies. Our team consisted of one designer, two machine learning engineers, and one full-stack developer. We built a service that generates virtual interview questions using OpenAI and extracts scripts from uploaded videos using Google's speech-to-text model. I was responsible for both backend and frontend development.

1. Set up GCR for Docker images, GKE for container orchestration, and GCS for video uploads.
2. Developed REST APIs & Auth APIs for User entities on the Main page, using Nest.js.
3. Assisted machine learning developers by building REST APIs using Flask for their models.