

Sangeeth Sudheer

📧 hire@sangeeth.dev 📍 Thalassery, Kerala, 🇮🇳

Everyday geek, seasoned full-stack web developer with expert practical knowledge on JavaScript technologies, proficient in multiple programming languages and tech stacks, have an unquenchable thirst for learning new things and applying them.

Skills

Programming Languages	Expert: JavaScript, TypeScript; Advanced: Python, PHP; Intermediate: Java, C#, C++; Beginner: Rust, Go, Ruby
Web Dev	Expert: ES6+/ESNext, CSS, HTML, React, Redux Toolkit; Advanced: Web APIs, PWA, Next.js, Preact, Vue, jQuery, Mobx, RxJS; Past Experience: Angular, AngularJS, Backbone, Dojo
Web Tooling	Advanced: Vite, Rollup, Webpack, Parcel, Bun, pnpm, yarn; Legacy: Gulp, Grunt
Back-end	Advanced: Node.js, Next.js, GraphQL Intermediate: Flask, PHP
Tooling	Visual Studio Code, Neovim, zsh, tmux, Linux/macOS/Windows, Git, GitHub Actions, Jenkins, GoCD, GitLab, Gitea
Tech I play around with	Docker, Proxmox, Linux distros, Kubernetes, MongoDB, Vercel, Cloudflare, Supabase
Soft Skills	Fluent communication in English, Project management, Documentation, Mentoring, Workshopping

Work Experience

Amazon Prime Video

June 2022 – December 2023

Front-end Engineer II

Remote

PRIME VIDEO X-RAY WEB & TV

- Architected, built and launched a brand new X-Ray redesign experience on the web, seen by 2M+ Prime Video users within 2-3 days as part of our rolling release strategy. Helped the 5-person engineering team overcome challenges in implementation, oversaw code reviews and guided implementation strategies of key pieces while in parallel working on other initiatives. Unblocked major obstacles in delivering design team's vision using modern and native web technologies overcoming challenges due to dated tooling. Guided in improving and establishing good accessibility and performance practises during development. [Demo](#) / [Screenshots](#)
- Architected, built and launched a complete end-to-end testing solution integrated into CI/CD pipelines using Playwright (for Chromium and Firefox) to assist the X-Ray redesign QA. This was instrumental in spotting regressions as the team was iterating on bug fixes and final touches. Reduced QA effort that required days/weeks to under 30 minutes. Dove deep to unblock obstacles w.r.t DRM limitations on CI platforms. Wrote integrations to generate detailed test run reports with screenshots, recordings and notify via email/Slack.
- Led, architected the refactoring and splitting of X-Ray web and TV codebases to VOD and Live entities to help teams move faster. Wrote scripts and guidelines to migrate with commit history intact, upgraded build tooling to be able to use newer TypeScript versions and championed the implementation of modern unit/integration testing tools like Jest and [@testing-library/react](#) which let the team write tests with ease and confidence.
- Designed and developed a web app in collaboration with AI/ML team to aid operators in the extraction of metadata from end-credit scenes. With this new tool, operators were able to complete tasks within 20 minutes vs. 2-3 hours it took following manual chores. The app used React, Redux Toolkit as key technologies. Sped-up implementation by many months by creating API contracts and using Mirage.js to mock the backend APIs. Extended the app later to

overcome backend limitations by implementing metadata organization logic on client-side itself persisted to IndexedDB.

- Led the 2023 operational excellence charter for the team. Planned tasks to accomplish organization-wide goals, oversaw their execution, made regular reports to leadership. We hit 95% of the goals we set out to accomplish, notably enabling extensive unit testing on all our projects and going from 0-5% to >90% test coverage across all the projects.
- Mentored juniors in the team, oversaw and co-ordinated cross-team projects, organized and took part in workshops and learning sessions, participated and won internal hackathons.

Titan / Flock / Neo (formerly Directi)

December 2019 – June 2022

Software Development Engineer II

Bengaluru, Remote

TITAN EMAIL

Titan is a modern, fresh and fast email experience that's built to replace many of the legacy email experiences found today on the web, especially the ones you get when you buy a domain/hosting plan from a provider. Here are some of the highlights from my work on Titan and the surrounding ecosystem:

- Worked on the React.js codebase of Calendar, part of the Titan offering. Introduced extensive unit tests using `@testing-library/react` and `jest`, implemented preview builds and CI with custom GitHub actions which will report coverage diff on each PR.
- Co-ordinated with DevOps@Titan to monitor metrics like main branch test coverage, TypeScript adoption progress via GitHub Actions and Elastic stack.
- Built a developer-friendly, Rollup-based project with multiple entry-points (like a monorepo) to ship scripts that can be used by PM teams on no-code platforms such as Unbounce.
- Led the development of the cPanel plugin—which is a critical piece of our stack—that is deployed to our hosting partner's (e.g.name.com, Hostgator) machine. Refactored, modernized and introduced unit tests and CI checks into the pipeline to make the plugin more resilient and less error-prone (Python, Preact and JavaScript, Bash scripting, Perl, PHP).
- Landed multiple features and bug fixes to Titan projects (React, Preact, JS).

FLOCK MESSENGER

Flock is messenger and productivity marketplace targeted at small-medium teams. The core of the Flock experience is a modern, React.js-based application which is shipped to web and desktop as an Electron application. Besides the core experience, Flock also has a marketplace of homebrew apps and hosts apps from third-parties. Some of the key highlights from my work:

- Helped accelerate the migration of Flock from Dojo 1.7 to React.js and shipped the brand-new, performant Flock to all our customers.
- Co-ordinated with PMs and Design team to ideate, design and ship a new Preferences modal for Flock.
- Led the development and modern tooling/framework upgrade for multiple Flock apps—Todo, Search, Reactions and more.
- Led and refactored the Electron codebase to use the latest version and use secure primitives like `contextIsolation`, sandboxing and custom handler for the `file://` protocol which resulted in a faster, safer desktop experience.
- Shipped a brand-new, reusable, React.js based Emoji Picker using MobX. Pulled high-resolution emoji files from emoji-pedia using a custom scraper built in Node.js.
- Championed and kick-started the adoption of TypeScript in our codebases. Held multiple sessions for developers in the team.
- Automated and streamlined the code review process via GitHub CI and GitHub Actions to lint, test and run custom sanity checks which resulted in less build-time errors and faster merges.

NetApp

October 2018 – December 2019

Full Stack Engineer

Bengaluru

DATA FABRIC

Data Fabric is NetApp's grand vision for the cloud of all things. It unifies all of NetApp's offerings which span on-prem and hybrid-cloud—accessible in a single pane of glass. As a part of the engineering effort for the Data Fabric, I worked on a multitude of technologies:

- Vue.js (Vuex + Vue-Router) and Meteor.js for the front-end experience
- Developing a design system/component library for Data Fabric and beyond
- Host of Node.js microservices for various functionalities
- Managing a Kubernetes cluster and working with k8s companion tech such as Helm, Istio, Træfik, Envoy etc.
- Version Control and CI/CD with GitLab Enterprise

CLOUDSTATS

CloudStats is an internal NetApp application that aggregates information from various cloud vendors such as AWS and neatly organizes it textually and visually on a dashboard. Modern MEAN stack was used to build this product from scratch. Plotly.js/D3 was used to render the visualizations.


CLOUD ADVISOR

We built a product based on MEAN stack, ASP.NET to analyze large volumes of log data from on-prem ONTAP boxes to predict costs, potential issues and use this to estimate the savings if the workload was migrated to our cloud offering.

Education

Integrated M.Tech in Software Engineering


2014 – 2019

 Vellore Institute of Technology

CGPA: 9.13

Middle and High School

2006 - 2014

 Indira Gandhi Public School


CBSE 12th CGPA: 8.8

CBSE 10th GPA: 10

Side Projects

My Homelab

2019 – Present

 status.sangeeth.dev

I have a humble homelab setup at my home, which consists of 3 Raspberry Pis. They all run the latest Raspbian and I use Docker and docker-compose to run containerized applications. I do plan to shift to k8s (or maybe k3s) but I'm waiting to get a beefier machine for more intensive tasks. Here are some of the few things I do with these little marvels of engineering:

- AdGuard Home is my DHCP and DNS server and helps secure the entire household. It's pretty good at blocking a lot of ads and spam and this is especially huge for me since I don't need to micromanage my parents' phones.
- I use nginx-proxy-manager and its ridiculously easy Let's Encrypt + Cloudflare integration to assign human readable domain names under *.lan.sangeeth.dev so that various services can easily be accessed from my local network. I'm a little scared at present to expose things on the public web but that's in the works.
- I am experimenting with Portainer as an alternative to maintaining my own compose files. It's pretty easy and convenient to run quick, one-off containers or stacks.
- A custom Next.js based dashboard and Node.js based cron job that runs fast.com tests thrice a day to ensure my ISP isn't throttling my speeds. It'll notify me if the last 3 runs are going below the expected threshold.
- A custom project to track orders across Indian courier services and notify on a Telegram channel. I started this because I was frustrated with Bluedart but gradually added a few other providers.

Client-side video compression on modern browsers using WebAssembly

December 2018 – April 2019

For my capstone project, I explored the possibilities of compressing videos on the browser without having to communicate with a dedicated server. My goal was to use Web Workers along with WebAssembly to compile FFmpeg and use it on the browser. I tried several approaches to get to this goal:

- To gain familiarity with WebAssembly, I tried to encode the libwebp library similar to how Squoosh did it and managed to achieve similar results.
- Modified Whammy.js, a pure JavaScript implementation for generating WebM videos from canvas animations to work with any video input. This worked but had serious limitations which included heavy performance hits as well as being limited to WebM output.
- Stitching together video frames after intraframe compression using libwebp compiled to WASM to achieve a result similar to (2). Had the same limitations.
- Compiling FFmpeg encoding functions to WASM using Emscripten and processing input video to result in acceptable compression. Web Worker threads were used to run the FFmpeg encoding functions so as not to impact the main thread which would otherwise result in poor UI performance. OffscreenCanvas rendering was also used to speed up processing of individual frames. A React web app was created to deliver the final experience.

FFCS On The Go — College Course Registration Planner

2016 – 2019

 GitHub

- Co-developed this open-source, highly successful and popular pure front-end project during my time as a Microsoft Student Partner and as a part of Microsoft Innovation Center to aid students in planning for course registration.
- A fast, familiar UI was conceived and has excellent mobile support.
- Utilized HTML5 features such as localStorage, IndexedDB, AppCache to help visualize slot-clashes, store multiple timetable offline for easy viewing at anytime.
- Course data fed to the front-end via AJAX provides an autocomplete functionality to quickly fill-in course details for a smooth experience.

ASME SLDC 2016 Website

2016

Built website for ASME VIT Vellore Chapter's Student-Led Design Conference 2016. The front-end consist of custom CSS and JavaScript for the design and animations. PHP and MySQL were used as the back-end and around which, a payment system was built for students to register for the listed events.