Yash Baheti

hello@yash.bio

Experience & Projects

- **2020 - Present:** ClayStack, Full stack developer - Creating the front-ends for the main product, static website and other centralized services. Also, taking care of back-end needs with node.js for centralized services.
- **2017 - 2019:** Zolo Poker **-** It is a gaming platform leveraging blockchain technology, focusing on bringing poker on the decentralized world. For the prototype, we opted for EOSIO as our choice of blockchain, and a Vue.js based PWA.
- **2019:** Reddit Trends **-** Get an overview of an activity on a subreddit. Uses the open Reddit JSON API to query data, and generate charts with charts.js. The interface is made using Vue.js.
    - Demo: https://reddit-trends.firebaseapp.com
    - GitHub: https://github.com/butttons/reddit-trends
- **2018:** Cleos plus - Tool to speed up smart contract development for the eos.io blockchain.
    - GitHub: https://github.com/butttons/cleos-plus
- **2019:** Netflix Remote - Control your Netflix video player on Chrome, from your mobile device. Uses WebRTC to establish a connection, a brokering server, using socket.io, to facilitate this, a chrome extension and an interface written with Vue.js to control the player.
    - GitHub: https://github.com/butttons/netflix-remote
- **2019 - 2020**: Ecommerce Marketplace - A marketplace focusing on clothing, using Vue.js for frontend, Express + Postgraphile for the backend.
- **2010 - 2012**: Facebook Apps - Developed apps for Facebook using PHP and the image functions to generate images tailored to the user.
- **2016**: ClashCaller bot - A bot for GroupMe messaging app, facilitating using a famous tool ClashCaller for the mobile game Clash of Clans, written in PHP.

Awards

- **2011: Application Development - 16 and Under**
  ThinkQuest 2011 by Oracle Education Foundation
  1st Place: The PHP Guide - by "TheTutorialMakers"

Technical Skills

Languages used:

- **2008 - present**: JavaScript - Proficient in ES6+
- **2018 - present**: TypeScript **-** It helps a lot by providing type safety, avoiding silly bugs.
- **2008 - 2016**: PHP - Started in 2008, to script some automatic tasks on the web. Moved on ahead with it, to learn it's role on the web.

- **2017 - present**: Python - Using Python for general purpose scripting, or web scraping before node.js had more mature options like puppeteer.
- **2010 - present**: SQL - Started with picking up SQL in 2010, to use with PHP. Since then, I have transitioned from using MySQL to PostgreSQL.
- **2008 - present**: HTML & CSS - Picked up HTML & CSS along with JavaScript.

Frameworks & Libraries used:

- **2016 - present**: Node.js - Started with my node experience in 2016, transitioning from PHP to node.js for my back end needs. My knowledge of JavaScript translated well, and didn't have many issues getting the hang of it.
- **2016 - present**: Express.js - Express is my go to framework for writing servers in node.js. Being the most famous one, it has a lot of mature plugins and middlewares that can help build resilient apps.
- **2018 - present**: Vue.js - For my front end work, I choose Vue.js. Coming from writing JavaScript, HTML and CSS without using bundlers like webpack and parcel, to a framework like Vue.js was jarring, but the single file components along with the fantastic documentation helped me learn Vue.js by making the mental model of writing HTML-JS-CSS in conjunction with each other, very relatable.
- **2019 - present**: Postgraphile - My default choice of generating a GraphQL API, with minimal configuration needed, and allows for rapid prototyping. Used correctly, with the right security mechanisms in place, this is a very handy tool.
- **2019 - present**: GraphQL - GraphQL does a very good job of exposing relational data, and along with the developer tools it ships with, it provides us with good developer experience.
- **2019 - present:** Tailwind CSS - Tailwind is my default choice for CSS frameworks. It provides us with countless customization options, to make your website look how you want it to, without locking you in, like many other CSS frameworks.
- **2018 - present**: PostgreSQL - Though not a framework or library, I started using PostgreSQL soon after picking up node.js. Postgres provides some features that MySQL lacks, like JSON data types, and row-level security. Also has the benefit of being available on all major cloud providers as a database option.
- **2017 - present**: Knex.js - A SQL query builder that I frequently use for small projects, helps with rapid prototyping of ideas.
- **2019 - present**: TypeORM - My choice for an ORM, it provides features that make schema designing a breeze, and also has full TypeScript support which many others don't have natively, like Sequelize. As soon as I discovered TypeORM, I migrated my then current project from Sequelize to TypeORM, and that was enough for me to continue using this.
- **2016 - present**: Pug - A very good templating engine for node.js to write HTML. Using pug in conjecture with single-file components of Vue.js results in a much cleaner and readable template, in my opinion.

Some more frameworks and libraries I've worked with - jQuery, Medoo, moment.js (using date-fns now because of the modularity provided), lowdb, puppeteer, socket.io, anime.js, Animate.css.

Education

2012

Ryan International School, Faridabad – *XII*th *class, CBSE*

2010

Ryan International School, Faridabad – *X*th *class, CBSE*