Systems Design Engineering | Full Stack Developer

chamod.gamage@uwaterloo.ca

% https://www.chamod.ga

in /in/chamod-gamage

Chamod-gamage

Feel free to check out some of my work:

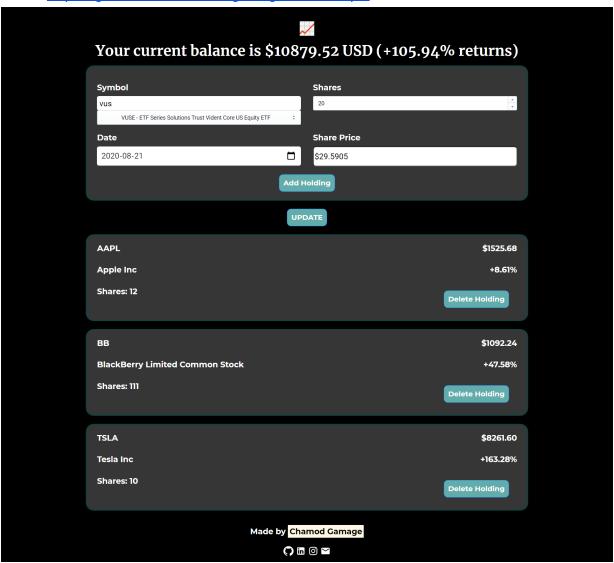
Stocksimple

Here's a simple stock portfolio tracker built using **React, Node.js, Express.js, and MongoDB** that allows you to stream live and historical data from the **Tradier Developer API** and track the value of your portfolio from a single page. Authentication was custom built with password hashing using **bcrypt** and user sessions with **JWT** cookies.

Visit the page any time and log into your account from anywhere - your portfolio's value will have been automatically updated!

Check it out at https://stocksimple.netlify.app/

GitHub: https://github.com/chamod-gamage/stocksimple



Systems Design Engineering | Full Stack Developer

chamod.gamage@uwaterloo.ca

% https://www.chamod.ga

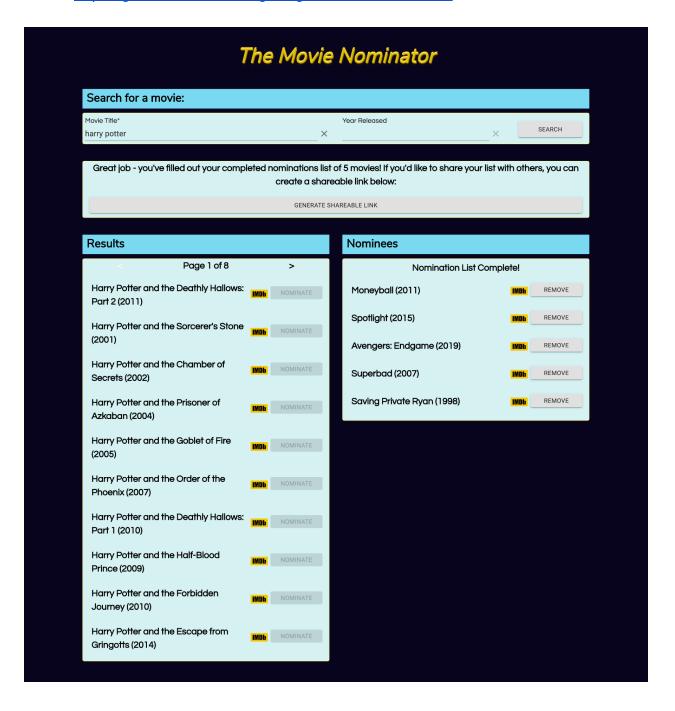
in /in/chamod-gamage

The Movie Nominator

Here's a movie nominator inspired by a Shopify challenge! This **React** app uses the **OMDB API** to allow you to search for and choose up to 5 movies to nominate for an award.

Check it out at https://the-movie-nominator.netlify.app!

GitHub: https://github.com/chamod-gamage/the-movie-nominator



Systems Design Engineering | Full Stack Developer

chamod.gamage@uwaterloo.ca

% https://www.chamod.ga

in /in/chamod-gamage

Honeybee - Full Stack Web Developer - Scheduled Messaging Feature

During my first co-op term at Honeybee (a company fostering a digital research study marketplace) I was a full stack web developer. One of the key features I conceived of, designed, pitched, and developed was a scheduled messaging feature to address the pain point of researchers having to message individual participants in real-time.

This feature allows researchers to schedule messages to be sent to a group of participants. Messages can be scheduled for any time, and reviewed/edited for content, delivery time, or recipients any time prior to their delivery. The feature also allows researchers to view their historical scheduled messages.

I developed the feature end-to-end, using **React** and the **React Context API** on the frontend, **Node.js** for the backend, and created additional collections (equivalent of schemas) in **MongoDB**.

Here are some related images on this page and the next:



Systems Design Engineering | Full Stack Developer

% https://www.chamod.ga

in /in/chamod-gamage

