

SE575 – Final Project Deliverable – Blockchain Demonstration Simulator

(come up with a short name for your project mine is Blockchain Demonstration Simulator)

Project Team Members

1. Brian Mitchell (submitter)
2. Student 2
3. Student 3
4. Student 4

(note that 1 student must submit an entire document, the other team members only submit up to this point – the project name and the project team members including the name of the person who is submitting all of the details. So if I was Student 2 I would stop with the information provided above. The remainder of this document would apply to the submitting student, me in this case)

Demonstration Video Link: <link to a place where I can view the demo video>

(most students post to youtube, this is best – do not attach large .mov or mp4 files to the submission, blackboard has issues with them – make sure its cloud accessible – also make sure you cap your video length at around 5 min – you should be able to give a good overview and demo of the project in that time. I watch all of these so please no really long videos, and don't make them super short – this is a pretty important aspect of the project that I use for grading. Don't sweat over exact time – if you go to 6 or 7 min that's fine, but any longer than that is not necessary)

Description

This project demonstrates a stand-alone blockchain simulator. It is not intended to be a comprehensive “real” blockchain, instead it introduces many novel features to demonstrate good architecture and design practices for building real world applications that run at scale and consider resiliency as a first class concern. Specific features included:

1. Standalone modern front end application built using the Angular framework and showcasing design attributes such as web components and reactive behavior
2. Integrated Proof-of-Work solver built into the front end to allow for easy testing of the solution, and simple demonstration.
3. A wide collection of API based solvers that demonstrate features of different programming languages and servers as they relate to solving brute force hashing solutions given the computational complexity.
4. Deployment options that embrace cloud-based serverless, AWS Lambda specifically, and high resiliency behavior using Kubernetes.

(as you can see this is the “proposal” section from the directions. A brief description and 3-5 bullet points outlining the key/novel attributes used in your project – this doesn't need to be very long)

Project Code Location: <link to a publicly hosted repository, e.g., GitHub or somewhere else>

(make sure that this is publicly accessible, test it please, I always get a few project with access issues)

Project Build Information: See readme section from the github repo:

<https://github.com/ArchitectingSoftware/bc-se575-aws-lambda-demo>

(Although you can writeup your build and deploy information inline, I would prefer you write build information and deploy with your code, see the example I provided for my lambda repo shown above)

High Level Design Overview

(The below is the level of picture that I would expect for your final project. High level, but shows the overall architecture of your solution)

