Oregon State University
College of Engineering

CS Capstone Technology Review

November 13, 2017

A Scalable Web Application Framework for Monitoring Energy Usage on Campus

Prepared for

Oregon State Office of Sustainability

Jack Woods _____ Signature _____ Date

Prepared by

Group 57

The Dream Team

Daniel Schroeder _____ Signature _____ Date

Aubrey Thenell _____ Signature _____ Date

Parker Bruni _____ Signature _____ Date

Abstract

This document provides an analysis of different technologies that could be used to satisfy different components of our web application. The purpose of this document is to compare and contrast different technologies in respect to our project's needs and goals and choose the best choice for implementation.

CONTENTS

1 Introduction

2 Visualization Frameworks

Our web application will provide near-real time data visualizations for energy consumption on campus buildings. This application will need to dynamically create charts and graphs based on energy data from the database. A key to choosing a visualization library will be to find one that can be dynmaically created and changed as new data is received from the data acquisition servers, and the ability to create chart templates that can be reused on multiple pages with different input parameters.

2.1 D3.js

Repository Commits: 4,104

Contributors: 120

Pros

- A lightwight, versatile javascript library that creates SVG elements within web pages and appends them to DOM elements.
- Makes use of javascript functions and DOM controlling functionality to dynamically change the content of the page.
- Provides a lot of variety and ability to customize graphics.
- Widely used and there is a lot of documentation and resources available to assist the learning and development processes.

Cons

- D3 is essentially an API to to manipulate SVG, it is not a charting library in of itself.
- You cannot easily pass a dataset into a specified chart type like other libraries.
- Considered to be “code-heavy” and difficult to jump right into as a novice user.
- Angular and D3 both attempt to control the DOM and so you have to find a way to make the two work together which is counterintuitive to both framework’s APIs.

2.2 vis.js

Repository Commits: 3,165

Contributors: 137

Pros

- Easy to use and less of a learning curve than D3.
- Allows for interaction and minpulation of data on the chart.
- Able to handle large amounts of dynamic data.
- Really clean and nice looking graphics.

Cons

- Limited amount of possible chart types.
- Does not have built in heat map.

2.3 Chart.js

Repository Commits: 2,465

Contributors: 236

Pros

- Uses HTML5 canvas element.
- Allows for easy creating based on chart type specification.
- Library provides Line Charts, Bar Charts, Radar Charts, Pie Charts, Polar Area Charts, and Doughnut Charts.
- Very responsive charts based on screen width.
- Simple API, easy to use.

Cons

- Limited amount of possible chart types.
- Does not have built in heat map.

2.4 Conclusion

In conclusion, despite the steep learning curve associated with D3.js we think it will be the best option for our web application. It has the widest range of available graphs to accomodate all the client's requirments and desired visualizations. There are also a number of wrapper libraries available for D3.js like DC.js and dimple.js to help create charts from D3. This is a great way to get around the clunkiness and downsides to D3.js and reap the benefits of all the other charting libraries. Another benefit to using D3 is the extensive amount of templates, examples, and documentation that exists to help guide the process and implmentation of our application.

3 Data-binding Technologies

4 Password Hashing Algorithms

4.1 PBKDF2

Pros

- Comes included with Node.js and is included with require()
- Uses salt hashing techniques.
- Contains other cryptographic primitives like symmetric and asymmetric encryption.

Cons

- Unsafe because PBKDF2 can be thoroughly optimized with GPU [1]

4.2 Bcrypt

Pros

- Very secure hash that can hash the same password multiple times.
- Widely used today and remains unbroken.
- Vetted by the entire crypto community as its now 15 years old [2].

Cons

- Slow and computationally expensive hashing
- Only used for password hashing, not a key-derivation function.

4.3 Scrypt

Pros

- New and
- Includes hashing capabilities.
- Easily incorporated with Mongoose schemas when storing user data.

Cons

-

REFERENCES

[1] T. Pornin, "Is bcrypt better than scrypt." https://security.stackexchange.com/questions/26245/is-bcrypt-better-than-scrypt, Dec 2012.

[2] M. Preziuso, "Password hashing: Pbkdf2, scrypt, bcrypt." https://medium.com/@mpreziuso/password-hashing-pbkdf2-scrypt-bcrypt-1ef4bb9c19b3.