Vue.js Workshop Reflection Paper

On September 17th, Dr. Dijana Kosmajac talked about how to build a single page application by Vue.js She introduced that Vue.js as a progressive framework for building front-end interfaces has become one of the three most popular front-end frameworks. Compare with its two competitors: Angular.js and React.js, Vue.js is a younger framework that is not backed by a giant corporation, like Google for Angular.js and Facebook for React.js. It is based on a flourishing open source community. Vue.js is created by Evan You, an ex-engineer of Google, with the inspiration from Auglar.js. He took his favorite part of Angular.js to make a lightweight framework that is using MVVM design pattern with the focus on View-model, connecting view and model with two-way reactive data binding.

In terms of the technical aspect, the core value of Vue.js is componentization and reactivity. Reactivity means that Vue.js can automatically react to the change of data on the page and respond to the change dynamically. The reactivity based on the two-way reactive data binding through MVVM helps developers save the time of manipulating with DOM object. Componentization represents that Vue.js breaks the modules of the single page application into components. A component is a Vue instance with predefined options. The componentization makes the code reusable and easy to access. It also increases maintainability since the project is broken into small parts and makes a significant positive impact on developing efficiency, especially when it comes to collaborative development.

Dr. Kosmajac also taught us how to build a Todo List App step by step. I do not

have any previous experience with front-end frameworks. I did work with a web development team that uses Angular.js but I was only required to code some very simple HTML part. The whole project was like another dimension to me. However, in Dr. Kosmajac's session, I found the framework is not that hard to understand. It was basically HTML pages that connected and interacted thought some pre-defined JavaScript syntax. Actually, it is much simpler and more convenient to add an event and implement data interaction using Vue.js than using native JavaScript.

This workshop aroused my interest in Vue.js. After the live session, I visited its official documentation and did some little practice on my own. I found the CLI provided by Vue makes it easy to start a Project that combines Webpack with Vue. A single command creates a project that supports Babel, Linting, testing, and reasonable directory structures. It also has very good expandability: with one command "vue add element", I can import the Element UI into my project without searching and pasting a long link in my project. With the help of npm, it handles all dependencies properly, so I do not need to worry about the dependency issue. Another feature I am surprised with is its component-scoped CSS. Simply add a "scoped" attribute in <style> tag can automatically scope this CSS to this component which frees me from coming up with new classes for a minor difference.

Vue.js could be a powerful tool in my future technology innovation work. It can make a significant reduction in developing time and make my work more professional. However, there are also challenges to implement Vue.js in my future development. For example, I usually develop a website with an integrated front and back end using pure

PHP. If I want to switch to Vue.js, I need to learn more about how to use Node.js or Laravel, etc. to build a back end.