Frontend Development Frameworks – React and Vu

Muhit Anik

Engineering Development Lead - Nasdaq

Before we start…why should you listen to me?

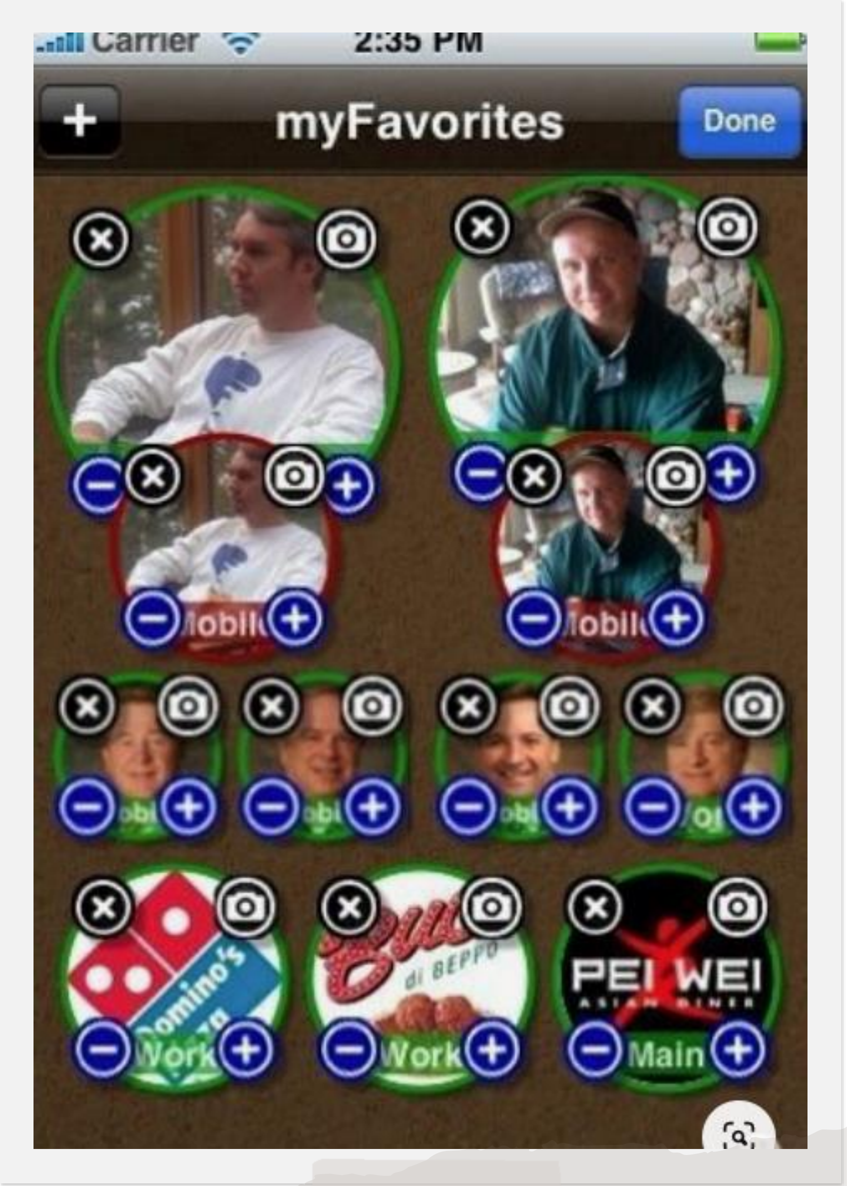
- I’ve been programming for almost 14 years now.
- I took various roles in my career - Full Stack Dev, Web dev, Front-end dev, Data Engineer, Data Scientist, Engineering Manager.
- I had a startup that helped people to learn programming.
- I am engineering lead for the second largest stock exchange in the world by market share (Nasdaq).
- I have worked for the Fortune 50 company (Facebook)
- But most importantly, I was in the same boat as you’re today - just a few years ago.

What is frontend development?

- It is what the users see and interact with.
- Think of HTML, CSS and Javascript when developing for web applications.
- When developing for mobile think of the UI/UX part of the codebase regardless of the language of choice (i.e. SwiftUI, Kotlin, React Native, Flutter etc)

But why is that important?

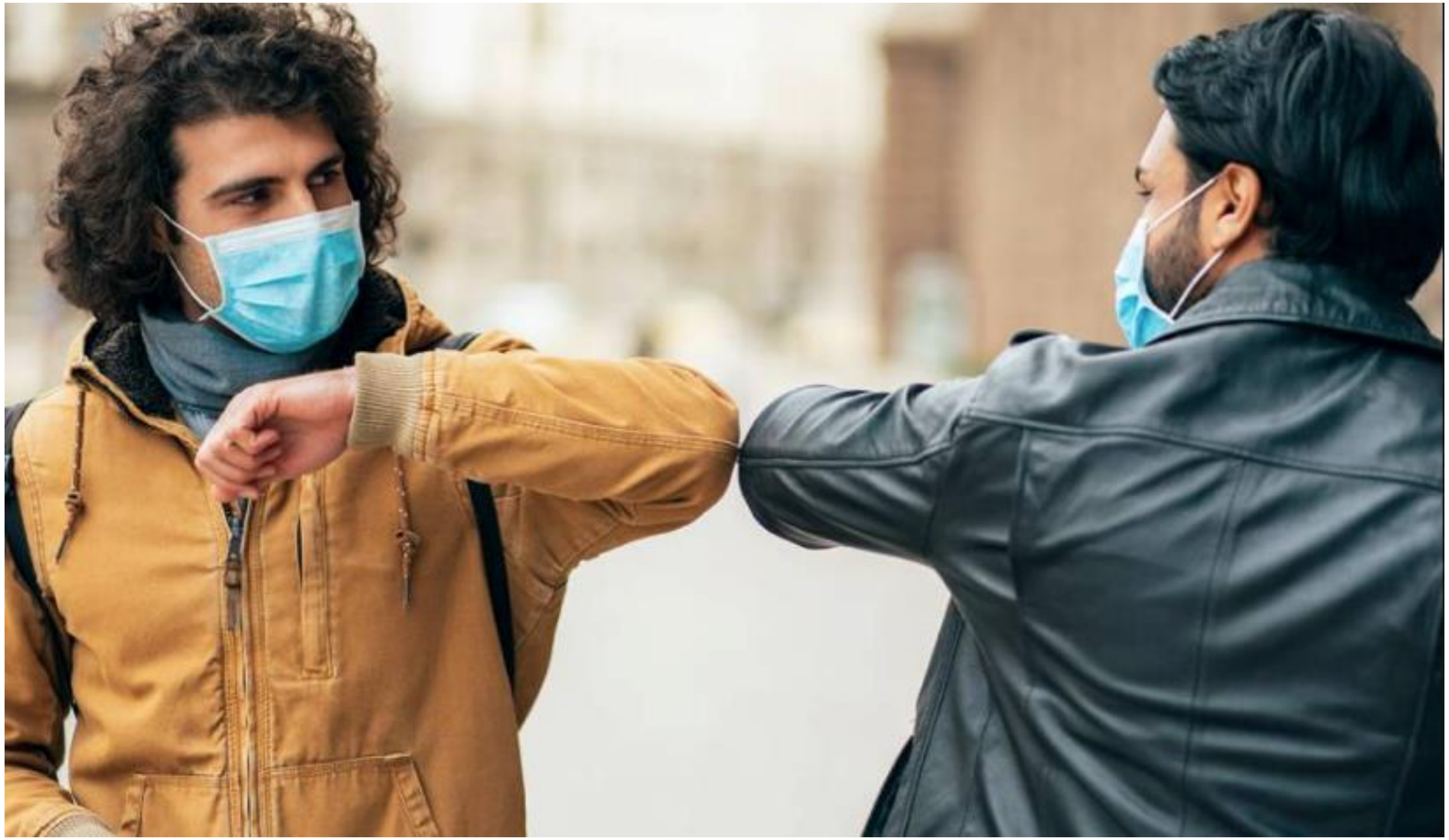
- Because that’s the part that users see!
- That’s what make users excited about using your application.
- Would you be using Instagram all day - if it looked like this?



Cool…but frontend is not engineering.

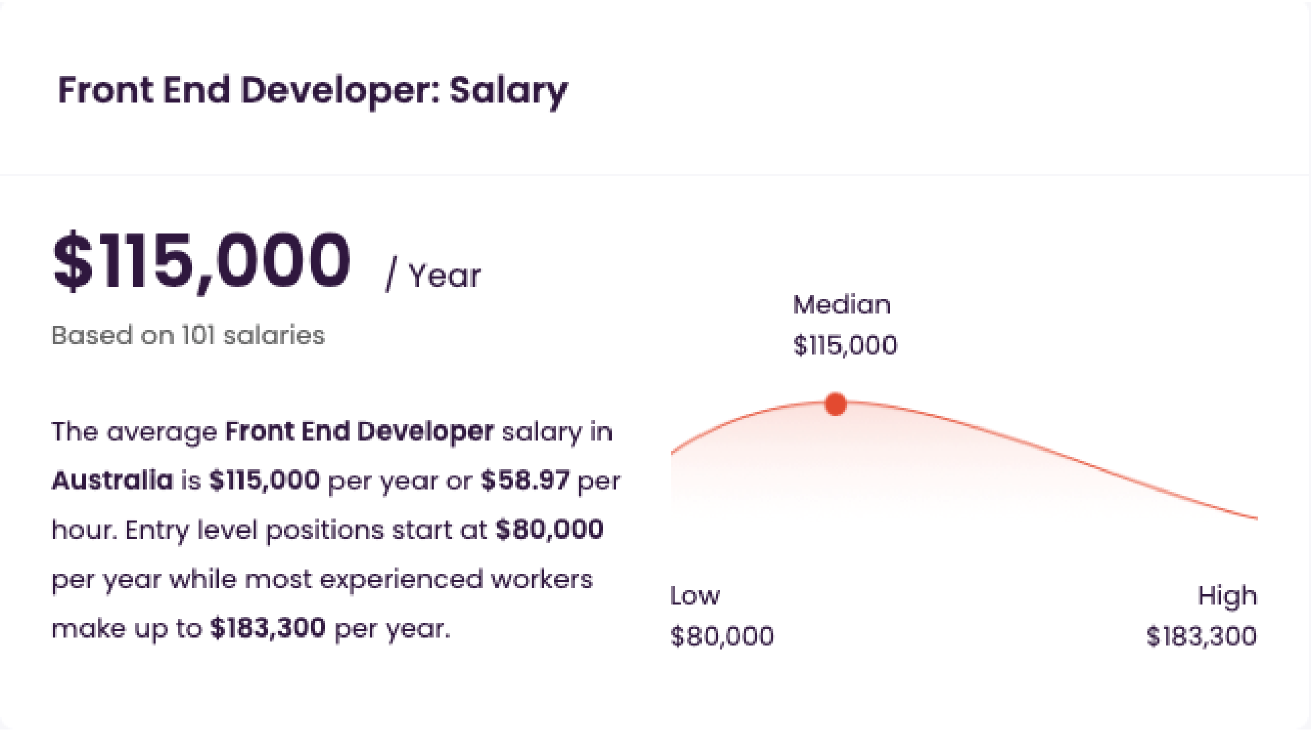
- WRONG!
- Frontend development can get as technical as the backend part.
- Despite having easier complexity to start with, it can get very challenging to master.
- Frontends change a lot faster than backend (i.e. tools, framework etc) – meaning frontend developers need to upskill every few years.
- Because how we interact in our daily lives change.

The new high five?



But does frontend developers make any money?

Source: au.talent.com



Front End Developer: Salary

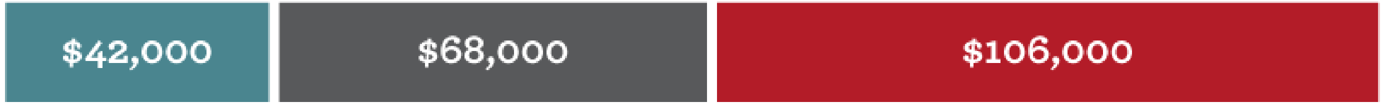
$115,000 / Year

Based on 101 salaries

The average **Front End Developer** salary in **Australia** is **$115,000** per year or **$58.97** per hour. Entry level positions start at **$80,000** per year while most experienced workers make up to **$183,300** per year.

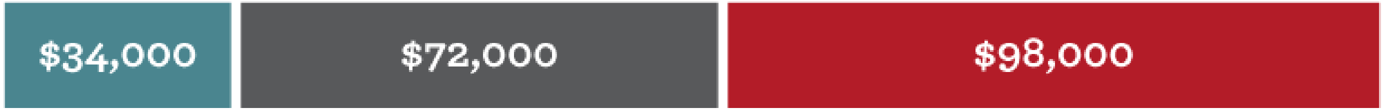
In comparison (US $)

FRONT-END DEVELOPERS | MEDIAN SALARY



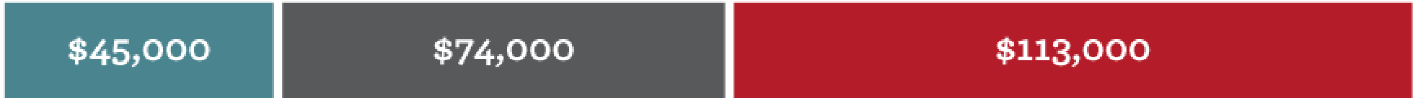
Source: Payscale.com

BACK-END DEVELOPERS | MEDIAN SALARY



Source: Payscale.com

FULL-STACK DEVELOPERS | MEDIAN SALARY



Source: Payscale.com

Alright, how do I get started?

Using a framework is a good place to start

- Because nobody writes code from scratch anymore
- Framework/Libraries provide support that's already available – rather than reinventing the wheel i.e. cuts development costs
- You will get paid on what works not the number of lines of codes you will be writing.
- Frameworks also limit various ways of doing one thing – which is a good thing. Less error!
- Most frameworks encourage code reusability.

Which one is the best framework?

- There's no best framework.
- Best framework is the one that works better for you, for your project, for your company - ***at this point in time***.
- Remember we said – things change because we change?

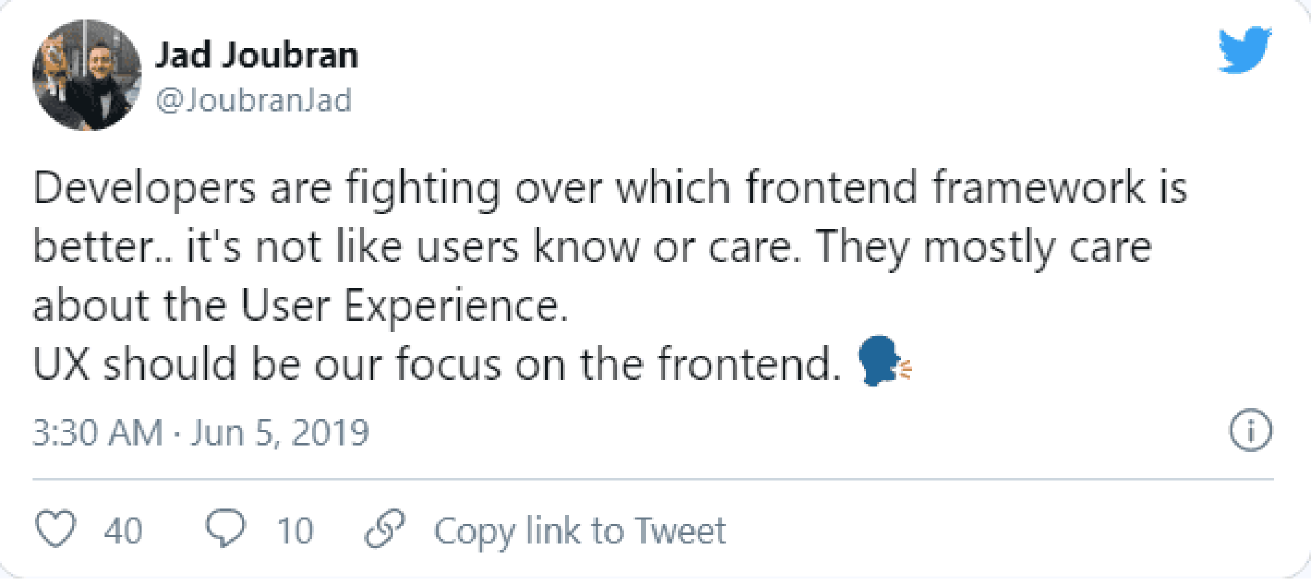
The frontend chronology

- 2005 : Dojo
- 2006: jQuery
- 2010: AngularJS, Backbone, Knockout
- 2011: CanJS, Ember, Enyo
- 2012: BaconJS, Meteor, Elm
- 2013: Mithril, React
- 2014: Done, Riot, VueJS
- 2015: Polymer, Preact
- 2016: Angular, Aurelia, InfernoJS, Svelte
- 2017: Stencil

How do I decide?

- Availability of learning – is enough resources already available for you to learn – if and when you get stuck? Latest is not always the best when it comes to development.
- Popularity – You might be learning something hot right now! But does it have enough support for the business? Is the framework stable enough? Are many companies using it?
- Core features – At the very basic level, is it solving enough business problems? Does it solve enough problems for your own project - out of the box?
- Usability – Is it easy to learn? Easy to get your head wrapped around? Are there other developers equally excited to learn about this?
- Ease of integration – Can you make it work with the backend, database of your choice, other parts of the codebase? Remember – Enterprises often have a mix of codebases with several languages.

But in the end…



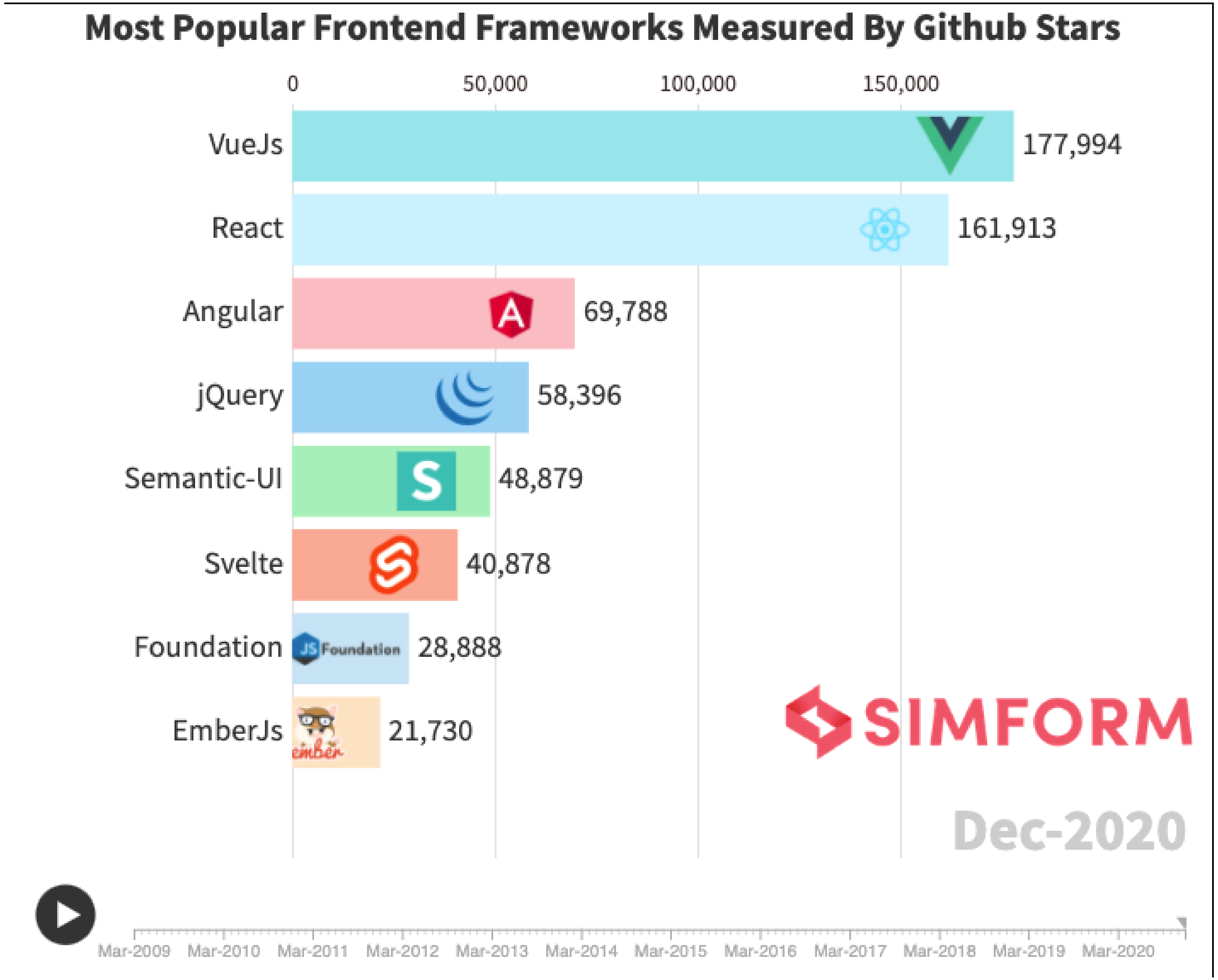
Jad Joubran
@JoubranJad

Developers are fighting over which frontend framework is better.. it's not like users know or care. They mostly care about the User Experience.
UX should be our focus on the frontend. 🗣

3:30 AM · Jun 5, 2019

But tell me what's hot right now?

Source: simform.com



Vuejs in 2 mins – “Hello, World!”

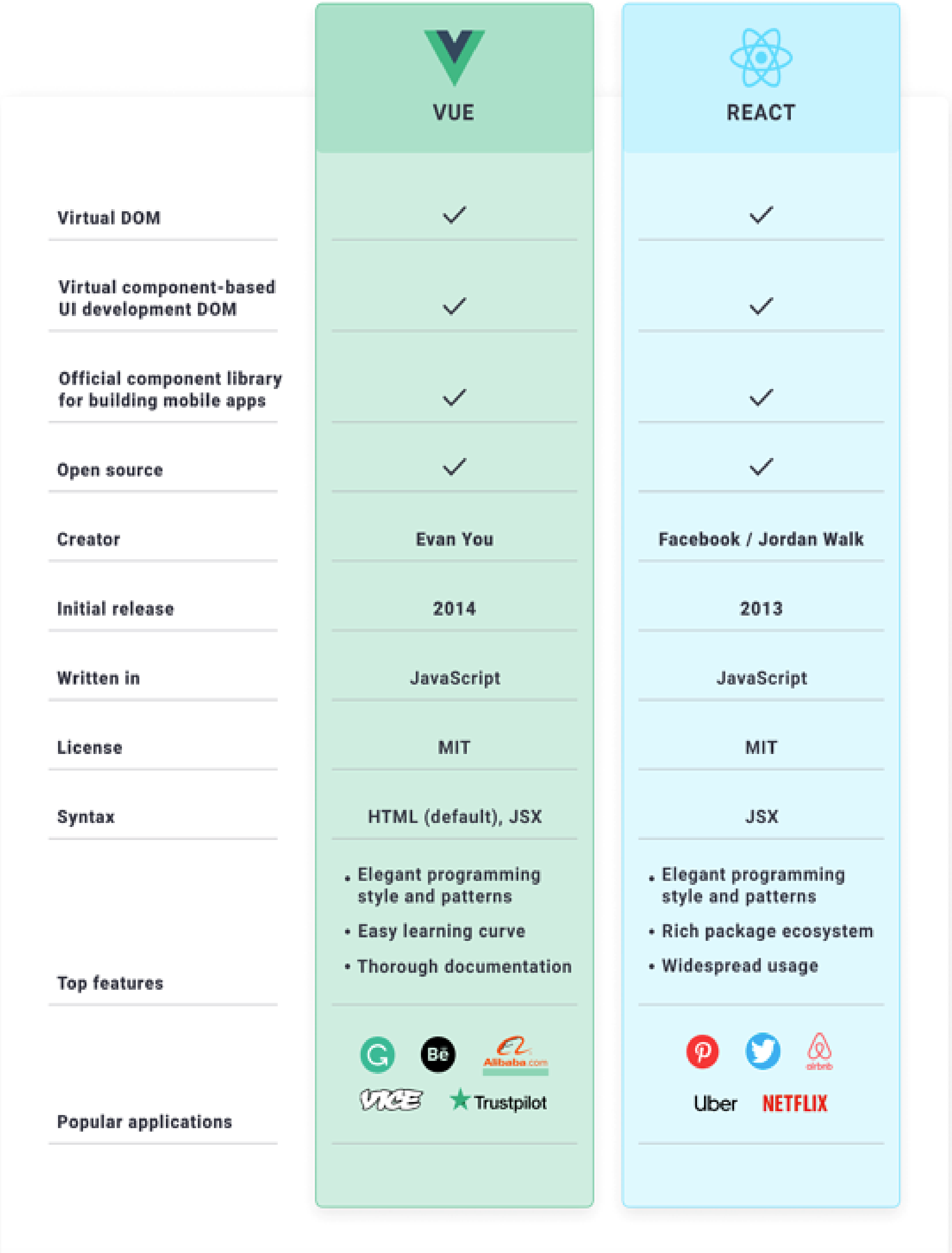
```
<!DOCTYPE html>
<html>
<head>
            <script src="https://cdn.jsdelivr.net/npm/vue"></script> <!-- Import Vue JS -->
</head>
<body>
            <div id="app">{{ message }}</div> <!-- Placeholder for your message -->
            <script>
                        const data = { message: "Hello, World!" }; // Actual message
                             new Vue({
                                    el: "#app", // Sent to the DOM element
                                    data: data
                                    });
            </script>
</body>
</html>
```

React in ~~2~~ 3 mins - “Hello, World!”

```
<html>
<head>
<script src="https://unpkg.com/react@16/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
</head>
<body>
<div id="root"></div>
        <script type="text/babel">
                class Hello extends React.Component {
                render() {
                                return <h1>Hello world!</h1>;
                }
        }
        ReactDOM.render(
                <Hello />,
                document.getElementById("root")
        );
</script>
</body>
</html>
```

A Quick Comparison between Vue and React

Source: monterail.com



	VUE	REACT
Virtual DOM	✓	✓
Virtual component-based UI development DOM	✓	✓
Official component library for building mobile apps	✓	✓
Open source	✓	✓
Creator	Evan You	Facebook / Jordan Walk
Initial release	2014	2013
Written in	JavaScript	JavaScript
License	MIT	MIT
Syntax	HTML (default), JSX	JSX
Top features	• Elegant programming style and patterns • Easy learning curve • Thorough documentation	• Elegant programming style and patterns • Rich package ecosystem • Widespread usage
Popular applications	G Be Alibaba.com VICE Trustpilot	airbnb Uber NETFLIX

What's common?

Both Vue and React uses virtual DOM - instead of updating the whole page, they update the DOM only when the objects change.

Both encourages code reusability - speeds up the development workflow

MVVM - MVVM is much more streamlined than MVC. It not only simplifies the dependence of business and interface, but also solves the problem of frequent data update, eliminating the need for selectors to manipulate DOM elements. Because in MVVM, View does not know the existence of Model, and Model and ViewModel cannot observe View. This low-coupling mode improves code reusability.

Let’s talk about the differences

Vue is good for smaller projects.

Vue gives a quick start in development and it is easy to understand due to HTML has better clarity over JSX.

React on the other hand has thousands of packages that can facilitate development.

React is good for projects with higher complexity and are typically large by nature.

Getting past the hello world!

Whichever framework you decide to choose for your project/work - always keep users in mind.

Build something personal - maybe your online portfolio to show your future employer?

Use it in other areas of work - integrate with a backend system

Development is not easy, it’s an iterative process of continuous improvement - always continue to improve your code and overall understanding.

Read on design patterns, design thinking, UI/UX etc

A Simple Table - Vue (5 mins)

- We will be adding some simple data and try to visualize them in a nice table.
- For the purpose we will use materialize which is an open source web boilerplate created by Google to make our table look nice and polished.
- We have created some sample data in the form of JS Array. Which could essentially be JSON.

Let’s import the libaries

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.12/dist/vue.js"></script> <!-- import vue -->
```

```
<!-- Compiled and minified CSS Materialize -->
```

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
```

```
<!-- Compiled and minified JavaScript Materialize -->
```

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
```

Our dataset

```
const dataItems = [
{ Year: 2004, femalewage: 21.04, malewage: 24.27 },
{ Year: 2005, femalewage: 21.89, malewage: 24.73 },
{ Year: 2006, femalewage: 22.37, malewage: 25.46 },
{ Year: 2007, femalewage: 24.26, malewage: 27.78 },
{ Year: 2008, femalewage: 24.65, malewage: 28.74 },
{ Year: 2009, femalewage: 25.42, malewage: 29.59 },
{ Year: 2010, femalewage: 26.75, malewage: 30.99 },
{ Year: 2011, femalewage: 27.73, malewage: 32.71 },
{ Year: 2012, femalewage: 28.84, malewage: 33.79 },
{ Year: 2013, femalewage: 29.5, malewage: 34.03 },
{ Year: 2014, femalewage: 32.63, malewage: 36.7 },
{ Year: 2015, femalewage: 33.46, malewage: 37 },
{ Year: 2016, femalewage: 33.96, malewage: 36.95 },
{ Year: 2017, femalewage: 35.65, malewage: 39.05 }
];
```

Vue code

```
var example1 = new Vue({
   el: '#example-1',
   data: {
   items: dataItems
   }
})
```

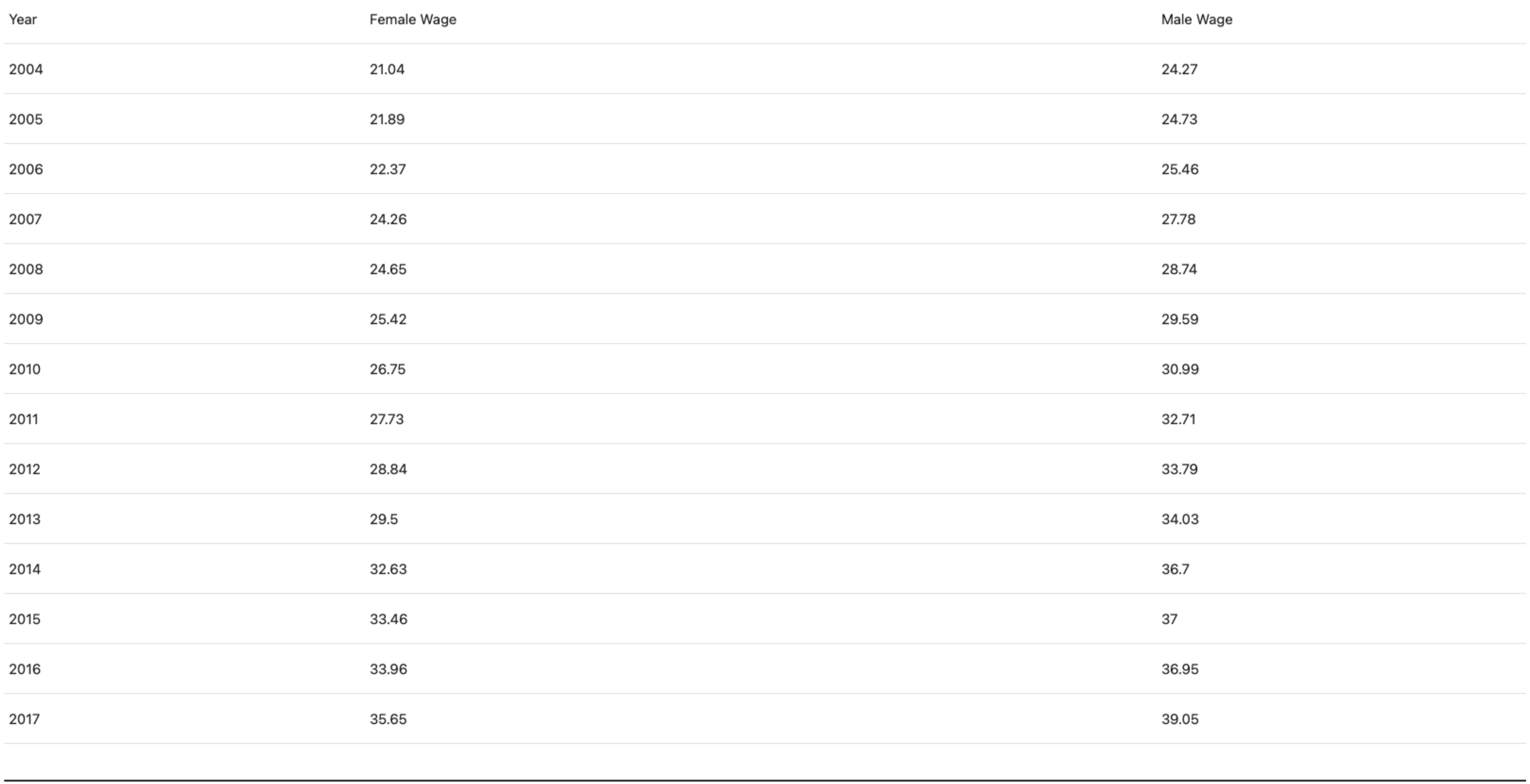
HTML

```
<table id="example-1">
<thead>
        <tr>
                <td>Year</td>
                <td>Female Wage</td>
                <td>Male Wage</td>
        </tr>
</thead>
<tbody>
        <tr v-for="item in items" :key="item.Year">
                <td>{{ item.Year }}</td>
                <td>{{ item.femalewage }}</td>
                <td>{{ item.malewage }}</td>
        </tr>
</tbody>
</table>
```

Entire codebase

https://github.com/muhit04/comp5347_guest_lecture

Voila!



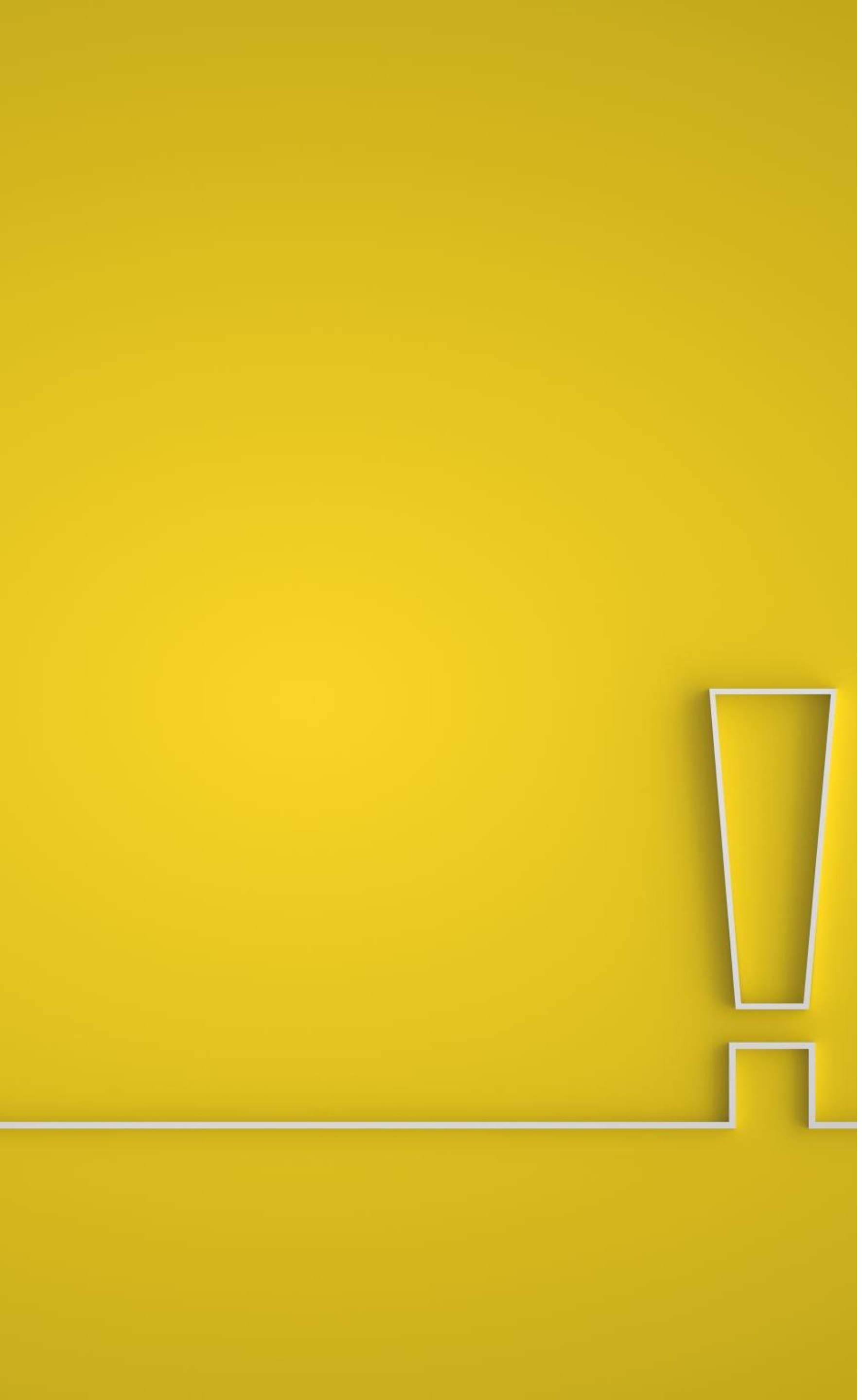
Year	Female Wage	Male Wage
2004	21.04	24.27
2005	21.89	24.73
2006	22.37	25.46
2007	24.26	27.78
2008	24.65	28.74
2009	25.42	29.59
2010	26.75	30.99
2011	27.73	32.71
2012	28.84	33.79
2013	29.5	34.03
2014	32.63	36.7
2015	33.46	37
2016	33.96	36.95
2017	35.65	39.05

Alright show me something cool!

- Let’s visualize the same dataset in line charts - because everyone loves some charts!
- We will be using chart.js for this and vue-chartjs wrapper

Same process - import the libraries first

```
<script src="https://unpkg.com/vue"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.1/Chart.min.js"></script>

<script src="https://unpkg.com/vue-chartjs/dist/vue-chartjs.min.js"></script>
```

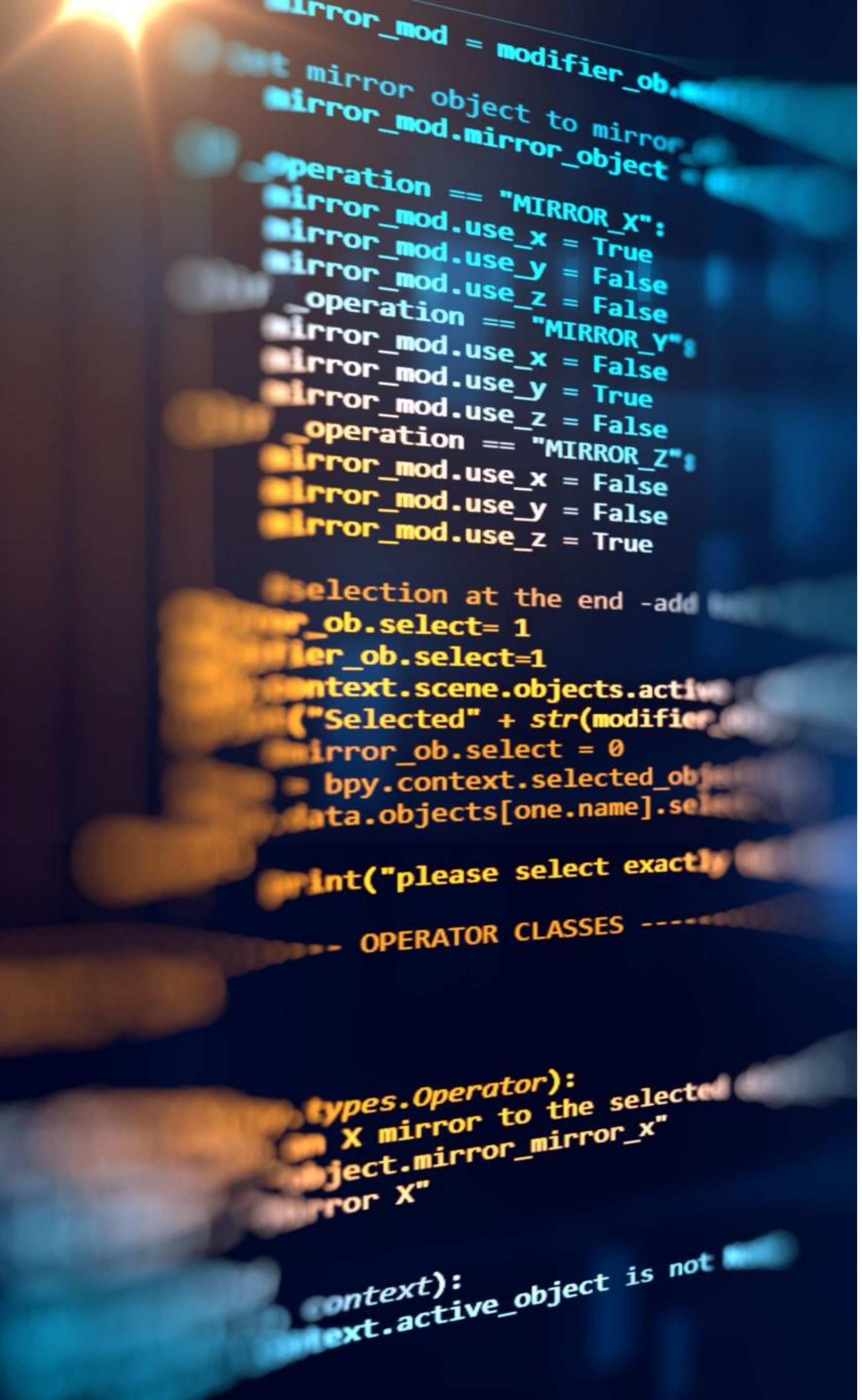


HTML placeholder

```
<div class="app">
<h1>{{message}}</h1>
<line-chart></line-chart>
</div>
```

Vue Code

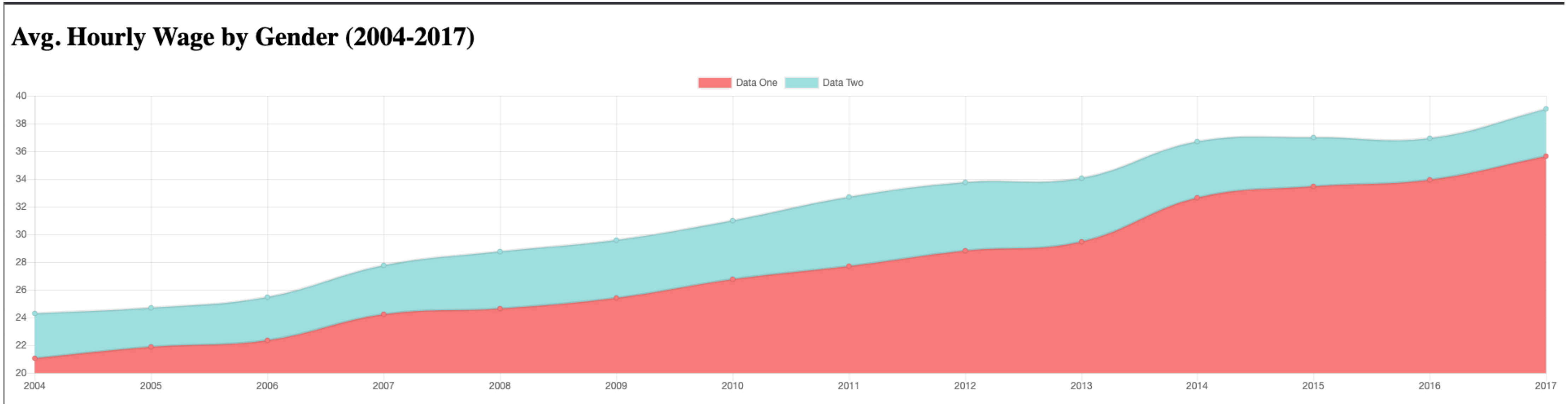
```
<script type="module">
Vue.component('line-chart', {
           extends: VueChartJs.Line,
           mounted () {
           this.renderChart({
           labels: ['2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016',
'2017’],
           datasets: [{
                      label: 'Data One’,
                      backgroundColor: '#f87979’,
                      data: [21.04, 21.89, 22.37, 24.26, 24.65, 25.42, 26.75, 27.73, 28.84, 29.5, 32.63, 33.46, 33.96, 35.65]
           },{
                      label: 'Data Two’,
                      backgroundColor: '#9DDEDE’,
                      data: [24.27, 24.73, 25.46, 27.78, 28.74, 29.59, 30.99, 32.71, 33.79, 34.03, 36.7, 37, 36.95, 39.05]
           }]
           }, {responsive: true, maintainAspectRatio: false})
           } })

           var vm = new Vue({
                      el: '.app’,
                      data: { message: 'Avg. Hourly Wage by Gender (2004-2017)’
           } })
</script>
```



Complete code

- https://github.com/muhit04/comp5347_guest_lecture

Coolness!



Avg. Hourly Wage by Gender (2004-2017)

Are you saying Vue is better than React?

No

All of these can also be done using React

It’s a matter of preference which one you need for your project and how do you find the style of coding.

You can also setup your project using command line https://cli.vuejs.org/ which sets up the project boilerplate for you - handy if you use nodejs.

How to learn more

- Vue JS official documentation - https://vuejs.org/v2/guide/
- React official documentation - https://reactjs.org/docs/getting-started.html

Final thoughts

- Don’t limit yourself learning just these two - But either of the two will give you a good start.
- There are hundreds of front-end libraries to choose from.
- Most importantly - have fun and make learning an enjoyable process.

How to connect

LinkedIn - https://www.linkedin.com/in/muhit04

Questions and Feedback