1. برای شروع، یک فایل HTML ایجاد می کنیم
2. با زدن میانبر ! چارچوب اولیه فایل HTML را ایجاد می کنیم HTML Boilerplate in VS Code with Emmet
3. و در قسمت head فایل html لینک سی دی ان به فایل VUE را اضافه میکنیم، این روش سریعترین راه افزودن Vue به پروژه است
4. قسمت Head را جمع می کنیم , بر روی قسمت Body تمرکز میکنیم
5. با افزودن یک تگ دیو ، یک کانتینر برای اپلیکشن ایجاد می کنیم
6. با اضافه کردن یک برچسب اسکریپت ، یک نمونه از Vue ایجاد می کنیم و مقدار خاصیت el را بر روی شناسه برچسب div قرار می دهیم ، که در مرحله قبل به عنوان ظرف برای برنامه ایجاد کردیم
7. و یک دیتا پراپرتی نام اضافه میکنیم و مقدار اون رو برابر با john قرار می دهیم
8. به کمک اینترپولیشن مقدار این پراپرتی را در یک پاراگراف درج می کنیم
9. اجازه بدین یک input control هم به فرم اضافه کنیم و مقدار اون رو به پراپرتی نام بایند کنیم
10. با توجه به بایند دو طرفه ، مقدار این پراپرتی و عبارت پاراگراف با هم تغییر می کنند
11. یک آرایه بنام Tasks ، برای نگهداری لیست to-do ایجاد می کنیم
12. و به کمک قابلیت Template Vue، و افزودن یک تک ul/li لیست Task ها را نمایش میدهیم
13. ظاهر لیست رو با افزودن استایل به تگ UL بهتر می کنیم
14. پراپرتی نام را به newTask تغییر می دهیم و یک label و یک دکمه هم به فرم اضافه میکنیم
15. یک متد با نام Add ایجاد میکنیم که یک عنوان رو بعنوان ورودی دریافت می کنید و یک تسک را به آرایه تسک ها اضافه می کند
16. بیایید متد رو با افزودن یک تسک جدید به لیست تسک ها، آزمایش کنیم
17. خوب، حالا اجازه بدین با استفاده از قابلیت بایند کردن استایل در Vue ، تسک های انجام شده را مشخص کنیم
18. با افزودن یک عبارت شرطی ، استایل text-decoration و color را بر اساس پراپرتی isDone بایند می کنیم
19. حالا با افزودن یک چک باکس و بایند کردن مقدار آن به پراپرتی isDone فرم را تکمیل می کنیم
20. بیایید با افزودن یک مقدار محاسباتی ، تعداد تسک های باقی مانده را نشان می دهیم
21. ابتدا یک خط افقی و یک پاراگراف اضافه می کنیم
22. و برای این مقدار محاسباتی یک متد اضافه می کنیم
23. همانطور که می بینید، مقدار فیلد محاسباتی با تغییر وضعیت تسک ها دوباره تغییر می کند. Vue یک چارچوب reactive است، و همه مقادیر متناسب با هر تغییر در Dom منعکس می شوند
24. و در پایان از قابلیت شرطی Vue استفاده می کنیم، تا وقتی تعداد Task های باقی مانده صفر شد این پاراگراف از Dom حذف شود
25. همانطور که مشاهده کردین، ما همه این قابلیت ها را با افزودن مقدار بسیار کمی کد پیاده سازی کردیم و این یکی از اصلی ترین دلایل محبوبیت فریمورک Vue است، قدرت در عین سادگی
26. In the following, Let's look at some of the Vue’s Pros
27. To begin, we create an empty HTML file
28. With the Emmet shortcut, Insert an HTML Boilerplate
29. In the head section, we add the CDN link of VUE file, this is the fastest way to add Vue to the project
30. Collapse the Head section
31. By Adding a div tag, we create a container section for the app
32. By adding a script tag, we create an Instance of Vue, and set the el property value, to the id of the div tag, which we created in the previous step, as the container for the app
33. Add a name data property and set its value to john
34. Now, by using interpolation, we insert the name value into a paragraph
35. Let's to add an input control to the form and bind its value, to the name data property, by using v-model attribute of Vue
36. As you can see, the paragraph expression changes, by changing the input value, because of two-way data binding of Vue
37. Next, we add an array of Tasks, to maintain the to-do list
38. By using the Templating feature of Vue, we render the list of tasks, by adding a ul/li tag
39. We Improve the look of the list, by changing the list-style-type of the UL tag
40. Now, we change the name of property to newTask and add a Label and a Button to the form
41. To add a task to the to-do list, we create a method called Add
42. Let's try the method by adding a new task to list
43. Okay, now let's highlight the tasks being done, by using the style binding feature of Vue
44. By adding a conditional expression, we bind the text-decoration and color style to isDone property
45. Next, by adding a checkbox and bind its value to isDone property, we can change the value of this property
46. Now, let's show the number of remaining tasks, by adding a computed property
47. First, we add a hr. tag and a paragraph to the form
48. And for this computed value, we add a method
49. As you can see, the value of the computed property, is recalculated by changing the task state. Vue is a reactive framework, and any changes to the properties are reflected in the DOM, Immediately
50. Finally, by using the Vue conditional expressions, we remove this paragraph from DOM, when the value of this computed property was zero
51. As you can see, we implemented all these functionalities, only by adding a few lines of code. This is one of the main reasons for the popularity of the Vue framework
52. In the following, let's look at some of the advantages and pros of Vue