**Micro Front Architecture using web components**

**Web Component in Angular:**

Micro Front End using Angular: - Documents to create the web component in Angular and use that component at other micro front end application.

1. Create the workspace for Angular Project first using below command

**ng new workspace-name –createApplication=”false”**

This command will only create the work space and not application.

1. Open CMD and go to the project and generate application

**ng generate application AppName**

This will create the application in Project folder in workspace.

1. Add angular element to the project

**ng add @angular/element**

This will add all the required packages in project to build web component.

1. Do the below changes to make the project as web component.
   1. In index file change root app selector to our project name selector so we can use that selector at other place.
   2. In app module add the Injector from angular/core and createCustomElement from angular/element to create web component.
   3. Add entry component in app module and remove all component specific from bootstrap array.
   4. Define the custom element using createCustomElement and add the in ngDoBootstrap method.
   5. Change the selector name in in app-component.ts file and do necessary changes in html file.
2. Next is to create the bundle for project to use that outside the project
   1. We are using ngx-build-plus tool to create the bundle
   2. Use below link for reference

<https://www.npmjs.com/package/ngx-build-plus>

* 1. Run below commands that will change the polyfills files

ng add ngx-build-plus

ng g ngx-build-plus:wc-polyfill

ng g ngx-build-plus:externals

1. Once we run above command that will change package.json files and inside our project this will create a file webpack.externals.js
2. Run below command to build our project

npm run build:appName:externals

1. After running above command, that will create dist folder, which contains folder with appName. All our angular and polyfill file bundled in this folder.
2. Now we got the dist folder but the size of this files is large, next to create the single bundle file.
3. Create new file in project folder build-element.js, which contains code to concat all the code in single bundle.
4. Create new file in project folder “element.js” which contains all the code in single bundle file
5. Install npm packages for file and concat
   1. npm i fs-extra –save-dev
   2. npm i concat –save-dev
6. Run the updated build script that had been added in package.json file to build the project
7. This will create element.js folder in dist folder which contain all the angular code in single js file.
8. Now this generated bundle we can share across project/ outside project and use as web component.
9. Reference: <https://www.youtube.com/watch?v=oqle07Ifyoc>

<https://www.npmjs.com/package/ngx-build-plus>

<https://indepth.dev/posts/1116/angular-web-components-a-complete-guide>

<https://angular.io/guide/elements>

**Web Component using Vue JS:**

1. Web components are a set of web platform APIs that allow you to create new custom, reusable, encapsulated HTML tags to use in web pages and web apps.
2. Vue.js is a framework that can help developers in creating web applications.
3. Vue.js is a JavaScript framework that can help developers in creating web applications. On top of that, it also has a feature to code web component! It even says this on its website.
4. You only need to write one command line to make Vue.js project be a ready-made web component.
5. Before creating a web component with Vue.js, make sure you have: Node.js v8.9+ and NPM v5.51+
6. To create a Vue.js project, we need the Vue CLI first. Use CMD or terminal for the installation process. Type the following syntax - **npm i -g @vue/cli**
7. Next, we will create the Vue.js. Project. To do that, we type the following syntax: **vue create v-card**
8. Next, there will be some choices. Choose the default like the example below:

? Please pick a preset:

> default (babel, eslint)

1. Now, the Vue.js project has been created. You can enter the project directory. To run the Vue.js project, type the following syntax: npm run serve
2. Then, open your web browser and type the URL <http://localhost:8080>
3. We will start to create a web component with Vue.js. To convert our application to web component please refer below link:
4. <https://medium.com/tunaiku-tech/your-first-web-component-with-vue-js-3386cffc0b1f>
5. We will build the web component for production so that it can be used outside of Vue.js. Project. The build process is quite easy. We can use the Vue CLI by typing the following syntax:

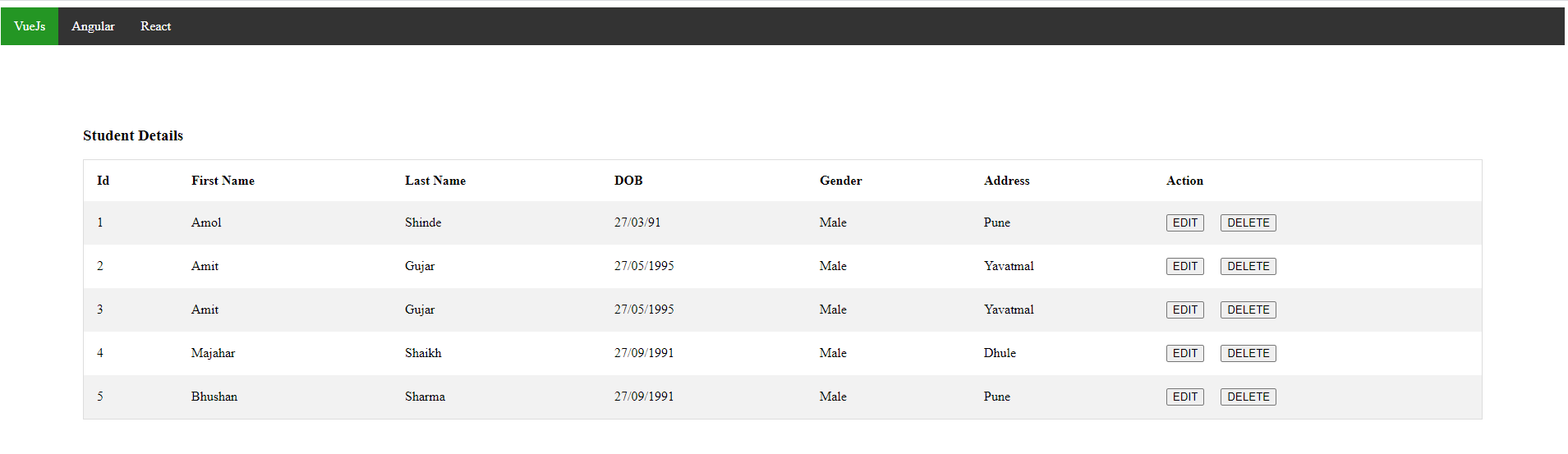
**npm run build -- --target wc --name v-card src\App.vue**

1. Copy the v-card.min.js file in dist/ folder, Paste the v-card.min.js file in our web application folder, Open the index.html file in our web application and load the vue.min.js and v-card.min.js scripts and card is ready to use.

**Integration of Vue and Angular Web components inside root application**

1. Created index.html which has routing for Angular and Vue Component and created src folder where we have placed web component js files.
2. We have three tabs – VueJs, Angular and React(not created any web component)
3. On click of VueJs, Student Registration form will be rendered.
4. After filling all information on form and click on submit.
5. On Submit, data will be passed to Angular and we can see in Grid/ListView
6. In this way, we can combine multiple web components in single application.
7. In nutshell, this is micro front architecture using web components.







**Data Sharing and Routing**

1. To create prototype we have used local storage for data sharing. For real word applications, we can use shared service or can middleware in node to share data between different web components.
2. Routing – We have used static routing in HTML using anchor tags. We can use Angular or react as base/root application for routing.

**Advantages of Micro front architecture using web components**

1. Reducing cross-team dependencies
2. Separate deploys for separate services
3. Ability to organize teams around business units or products
4. Independent development teams can collaborate on a front-end app more easily.
5. They can provide a means for migrating from an “old” app by having a “new” app running side by side with it.

**Disadvantages of Micro front architecture using web components**

1. Testing can be more difficult.
2. If integrating more external libraries, file size can be large for web components.
3. With each team making their own technology choices, browsers may end up downloading multiple frameworks & duplicate code