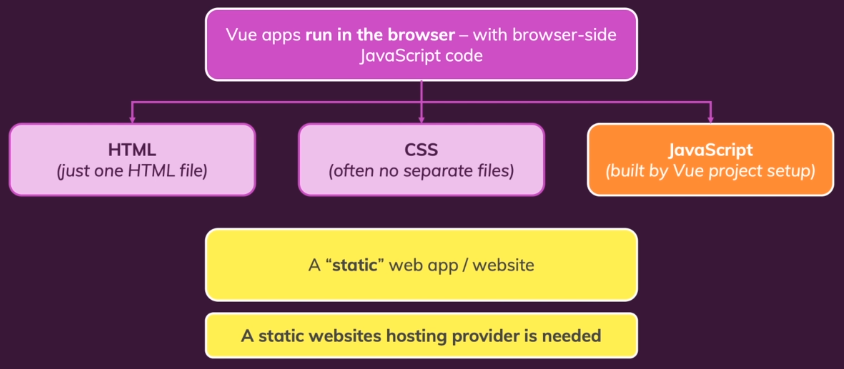
**Section 18 – Optimizing And Deploying Vue**

Chapter 273 – What To Deploy ?



Chapter 274 – Optimization: Using Asynchronous Components

1. Optimize Your Code
2. Test for errors
3. Refactor and “Don’t repeat yourself”
4. Consider using “Asynchronous Components”
5. In Vuex until this course, we always download all the code, even we didn’t need the code, we still download it, and it will make your browser become slower. Therefore, we need to optimize it, for example BaseDialog.vue, we are not always need this file. All the important file must be imported in main.js. We may input this code below in main.js. It will download BaseDialog.vue when only needed.

import { createApp, defineAsyncComponent } from 'vue';

// import BaseDialog from './components/ui/BaseDialog.vue';

const BaseDialog = defineAsyncComponent(() => import('./components/ui/BaseDialog.vue'));

app.component('base-dialog', BaseDialog);

1. In router.js, we may add this code.

import { defineAsyncComponent } from 'vue';

1. We can change the router syntax like below.

// import CoachDetail from './pages/coaches/CoachDetail.vue';

import CoachesList from './pages/coaches/CoachesList.vue';

// import CoachRegistation from './pages/coaches/CoachRegistration.vue';

// import ContactCoach from './pages/requests/ContactCoach.vue';

// import RequestsReceived from './pages/requests/RequestsReceived.vue';

import NotFound from './pages/NotFound.vue';

// import UserAuth from './pages/auth/UserAuth.vue';

import store from './store/index.js';

const CoachDetail = defineAsyncComponent(() =>

  import('./pages/coaches/CoachDetail.vue')

);

const CoachRegistation = defineAsyncComponent(() =>

  import('./pages/coaches/CoachRegistration.vue')

);

const ContactCoach = defineAsyncComponent(() =>

  import('./pages/requests/ContactCoach.vue')

);

const RequestsReceived = defineAsyncComponent(() =>

  import('./pages/requests/RequestsReceived.vue')

);

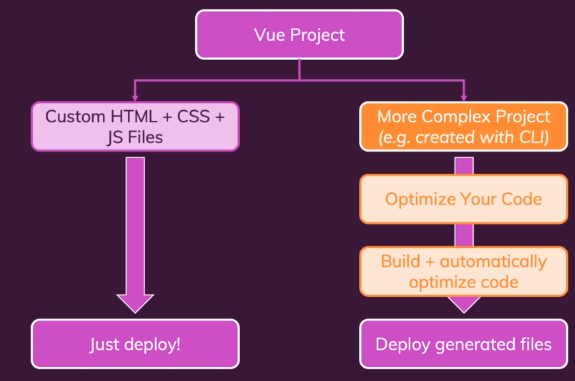
const UserAuth = defineAsyncComponent(() =>

  import('./pages/auth/UserAuth.vue')

);

Chapter 275 – Building The Project For Production

1. Our Optimization Steps.



1. To build our App vue, go to terminal and write the code below. You will see dist folder contains a single page index.html, javascript folder, and css folder.

npm run build

Chapter 276 – Deploying A Vue App

1. Setting our FireBase, go to Hosting, click Get Started, run the command below in our terminal.

npm install -g firebase-tools

1. Run this code to log into Firebase. You need to verify it.

firebase login

firebase init

1. Chose Hosting …, enter, use existing one, enter, choose vue-http-demo-… (our project name).

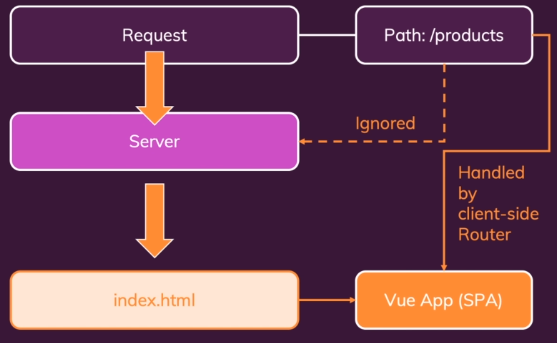
What do you want to use as your public directory ? dist

Configure as a single page app ? Y

File dist/index.html already exists. Overwrite ? N

Set up automatic builds and deploys with Github ? N

1. Process to access our website using a single page app.



1. Don’t touch any code, file, or folder which is generated in dist folder. We can change the code in index.html before build the project and rebuild it. For example, we want to change the title in index.html into Find a Coach and add meta description. Run npm run build again to rebuild it.
2. If you don’t want to change anything and just deploy our Vue App in Firebase, run this code below.

firebase deploy

1. It will give you an URL, you can access your Vue App through that URL. Then, we can run our app on the real web server.