

Props are custom attributes that you can register any components. You define your data on the parent component and give it a value, then you go to the child component that needs the data and pass the value to a props attributes. Therefore the data becomes a property in the child component.

rendering template from another file.

```

Name.vue
<template>
  <div>
    <p> dhinesh </p>
    <p> 22 </p>
  </div>
</template>

App.vue
<script setup>
  import Name from './components/Name.vue'
</script>
<template>
  <div>
    <Name />
  </div>
</template>

```

output → dhinesh
22

```
[ {id: 1, "name": "dhinesh", "age": 22 }, {id: 2, "name": "velam", "age": 43 }
```

Name.vue.

<script setup>

import { defineProps } from 'vue';

const { names } = defineProps(['names'])

</script>

<template>

<div>

<p> {{ names.name }} </p>

<p> {{ names.age }} </p> </div> </template>

App.vue.

<script setup>

import Name from ". / components / Name.vue;

import NameList from ". / data / Name.json"

< templates >

< div >

< Name v-for = "names in NameList" : key = "names.id" : names = "names" / >

props we can use whatever name

App.vue.

: n = "names"

Name.vue.

const { n } = defineProps(['n']) ← script

{ n.names }
{ n.age }

← Template.

example if we want to use : n of

we can also base the writable base from the last example

Routing.

npm install vue-router.

→ it will add the needed node modules and dependencies.

HomeView.vue

< templates

< div> < h1> home page < /h1> < /div>

< templates

AboutView.vue.

< templates

< div> < h1> About page < /h1> < /div>

< templates

index.js.

import { createRouter, createWebHistory } from "vue-router";

import HomeView from "../views/HomeView.vue"

import AboutView from "../views/AboutView.vue"

routing.

→ IF we add the usage more we can add

```
const router = createRouter({ history: createWebHistory(import.meta.env.BASE_URL),  
  routes: [  
    { path: '/',  
      name: 'home',  
      component: HomeView },  
    { path: '/about',  
      name: 'about',  
      component: AboutView }  
  ]  
})  
  
export default router;
```

we declared the
the router path
using the
components file.

And exporting it
to use.

main.js.

```
import { createApp } from 'vue';  
import App from './App.vue'  
import router from './router'  
import './assets/main.css'
```

← importing the router file.

const app = createApp(App)

app.use(router)

app.mount('#app').

using the router.

(and now we need to tell where to use the router)

App.vue.

<script setup>

import { RouterView } from "vue-router"

</script>


```
import { createApp } from 'vue';  
import App from './App.vue'  
import router from './router'  
import './assets/main.css'
```

← importing the router file.

```
const app = createApp(App)  
app.use(router)
```

using the router.

```
app.mount('#app').
```

(and now we need to tell where to use the router

App.vue.

```
<script setup>
```

```
import { RouterView } from "vue-router"
```

```
</script>
```

```
<template>
```

```
<div> <RouterView /> </div> </template>
```

in the last example we create routes but that needs manually typing, instead we use clickable to render the page,

Nav.vue,

```
<template>
```

```
<div>
```

```
<a href="/">Home </a>
```

```
<a href="/about">About </a> </div>
```

```
</template>
```

App.vue.

```
<script setup>
```

```
import Nav from './components/Nav.vue'
```

```
</script>
```

```
<template>
```

```
<div>
```

```
<Nav />
```

```
<RouterView />
```

(This code will work perfectly fine but when we clicking the the a tag it will refresh the full page and re-render the about (or) home in vue we don't want that we need to render the pages without website reloading, so we can use RouterLink instead of a tag.

</script>
<script setup>

import { RouterLink } from "vue-router"

</script>

<template>

<div>

< RouterLink to = "/" > Home </RouterLink>

< RouterLink to = "/about" > About </RouterLink>

Active class -

in this example we can add active-class.

< RouterLink to = "/" active-class = "active" > Home </RouterLink>

<style scoped>

.active { color: gold; }

</style>.

(when the button is active it will trigger the style.)

~~npm~~ npm install element-plus --save.

import ElementPlus from 'element-plus';

dropdown

checkbox

button.

datepicker

Vue elements.

npm install element-plus --save. → installation

to use the elements, in main.js.

main.js

```
import { createApp } from 'vue';
```

```
import App from './App.vue';
```

```
import './assets/main.css'
```

```
import ElementPlus from 'element-plus';
```

```
const app = createApp(App)
```

```
app.use(ElementPlus)
```

```
app.mount('#app');
```


Amazon EC2 → Secure and resizable compute capacity for virtually any workload.

Amazon S3 → Object Storage build to retrieve any amount of data from anywhere.

Amazon RDS → Setup operate and scale a relational database in the cloud with just a few clicks.

IAM → Identity & Access Management

• IAM has a global view

• Your whole AWS Security is there,

1) Users (usually a physical person)

2) Groups (Containing users)

3) Roles. (internal usage within AWS resources).

IAM
EC2

• VPC
• Elastic IP

Key points :

- * One IAM user per physical person ,
- * One IAM Role per Application ,
- * IAM credentials should never be shared,
- * Should not write IAM credentials in code,
- * Never use the root account ^{obscure} except for initial setup,
- * Never use Root IAM credentials .

FOR SUBPOF



EC2 instance = Amazon machine image (AMI)

Amazon EC2 launch EC2 instance → EC2 instance

94450

physical parameters

→ CPU + RAM + Disk + NIC + ^{observed} network

9445073215

traditional server → CPU + RAM + Disk + NIC + operating system.

Physical hardware

ditto

Amazon EC2 launch EC2 instance → EC2 instance.

EC2 instance = Amazon machine image (AMI)



EBS snapshot

Operating system

Operating system

AMI

AMI

ECS (Elastic Compute Cloud)

- + ECS instances - Virtual server in AWS (billed by seconds).
- + ECS Service - Provision ECS instances on Virtual server.
- + You can create new instance based on AMI.
- + AMIs are preconfigured template that include base operating system and any additional software in a single package.
- + It provides various combinations of CPU, memory, storage, networking capacity for provisioning instances. These combinations are called instance types.

Factor influencing you billing,

- + Types of instance
- + How long do you run for it.
- + Right size your resources.
- + Choose the right IOPS for your storage.

AWS VPC.

region (geographical location). (e.g) us west 1 (or) eu west 11
Availability zone \rightarrow (basically data centers within that region), (2 or more per region).

CIDR (Classless Inter-Domain Routing)

Notation for describing block of IP addresses.

Network ACLs and Security Groups,

Network ACLs

Firewall that controls

traffic in/out of a subnet

Rules for Allow and deny.

Rules include IP

Address (only)

Security groups.

Firewall that controls traffic in/out of an EC2 instance.

Rules for Allow (only).

Rules include IP addresses and other

security groups.

Amazon S3 is an object storage service from Amazon that provides scalability, data availability, security, and performance.

Benefits: 1) Performance, scalability, availability, and durability that are unrivalled in the industry.

2) Most cost effective storage classes are available.

3) Security compliance, auditing capabilities that are unrivalled.

4) Manage data and access permission with ease

5) Query in place and on demand processing.

6) The most widely used cloud storage service.

Creating bucket, (aws, amazon/consol)

Login with Account ID, Email, name, password.

23 006 5006 - IV

23 14 11 11 11 11 - 11 11 11

23 2 10 10 10 - IV

23 2 10 10 10 - IV

(for bucket access)

(for bucket access)

Working of Amazon S3.

Amazon S3 → S3 Standard, (For frequent data access) (working hours)

S3 Standard-IA, (Infrequent data access).

S3 intelligent-tiering,

S3 one zone-IA,

working with AWS Glue, AWS Redshift, AWS EMR, AWS Athena.

classified properly (S3 Glacier Deep Archive)

(Performance report
(old records)

Amazon S3 Standard Reduced Redundancy Storage

(external stored data)

Redundancy 5 storage any on growing business.

It moves data and access permission with ease

Example: external data stored which are not important and can be

3. Infallible under this category. classes are available

The important:

4 stage all the changes. run build.

installing vue.

`npm init vue@latest -`

`npm install`

`npm run dev`

} inside the project folder.

after this it will ask for project
after.

Ⓜ️ click (or) A-on : click → if miss page is op? (or) from previous page

Composition API

<template>

<script setup>

import {ref} from 'vue';

const count = ref(0)

const add = () => { count.value++ }

const sub = () => { count.value-- }

</script>

<template>

<main>

<h1> {{ count }} </h1>

<button @click = 'add' > + </button>

<button @click = 'sub' > - </button>

</main>

State.

handler.

Option API

<script>

export default ({

~~data~~ data() {

return { count: 0 }

};

methods: {

add() { this.count++ },

sub() { this.count-- }

})

Same template.

let message = ref('hello')

→ tweaking the message using .value.

If you don't want to use like this

Go to vite.config.js inside this add vue({ reactivityTransform: true, })
inside the plugin.

now you can,

let message = \$ref('hello').

→ tweaking message no need .value.

```
import { watch } from 'vue';  
const search = ref('')
```

```
watch(search, () => { console.log("hello") })
```

watch function watches the value. in the case its search, whenever the search value change it will execute the following function.

open Xampp start Apache and MySQL.

open the MySQL admin page and we can create the database.

in vs code open the folder Xampp → htdocs → create folder → project.
in terminal,

composer create-project laravel/laravel sample-app

open the project folder in vs code.
Project name.

in .env file change the DB name to → what we create in Xampp.

Go to folder → 'sample-app' → 'config' → 'database' → 'DB_CONNECTION = mysql' → 'DB_DATABASE = sample-app'.

Composer require laravel/breeze --dev → it will create all authentication

include registration, password, reset pass, email verify, pass

PHP artisan breeze:install vue
PHP artisan migrate.

combobox address | process -- qsa

→ if null, create all at
major redistribution

Blaravel → backend frame work.

features → routing, validating, caching, DB - DATABASE = 20mb, FILES storage and

in our tips spends the DB names for → major ms create in Xambb
other the brotst folder in as code

brotst names

combobox create-project folders | folders | combob | 5 - abb
in forming

in as code other the folder Xambb → process → create folder → brotst
other the mbd | again bade and ms can create the qofopass
Xambb effort Xbocps and mbd

controllers

information as the note

controllers, migrations, routes, resources
(MVC)
model → allow to interact with database.
view → user interface.

Controller → it helps to connect model & view
and contains all the business logic.

↳ JS
↳ Pages

→ Frontend page.

github.

create repository.



sent invitation.

Settings → collaborators → Add people (enter the user name who want to add
git clone,

copy the https code in github, and paste this into VS code new window
opening the clone git repository.

↑ opening the code in VS code.

(ctrl + shift + P)

laravel breeze → implementation of all laravel authentication features
include login, registration, password reset, email verification, password confirmation.
inertia → is a small library that allows you to render single file vue components

namespaces →

namespaces →

lay file storing.

```
public function store (Request $request)
```

```
{ Valida
```

```
$document = $request → file('document'); →
```

```
$document->file_name = time().$document->getClientOriginal() →
```

```
$document → storeAs('employee_details', $document->file_name)
```

```
Employee::create([
```

```
'document' => $document ])
```

```
→ redirect() → route('employees.index');
```

```
return redirect() → route('employees.index');
```

POPIS KONTAKT

POPIS KONTAKT →

450

Route.

routes → web.php.

```
Route::resource('documents', DocumentController::class);
```

```
Route::get('districts', [App\Http\Controllers\DocumentController::class, 'getDistricts']);
```

→ name()

creating app,

into Xampp/htdocs → in terminal → composer create-project laravel/laravel office

cd office-app → code . (it will open the folder in VS Code)

php artisan serve →

project

creating app,

into xampp/htdocs → in terminal → composer create-project laravel/laravel office-app
cd office-app → code . (it will open the folder in vs code) project name.
php artisan serve →

File storing,

PPC → pay per click, cost (USD, CAD, EUR, pound)

PPC campaign → showing their Ads in top.

target audience → (age wise, gender wise, householding, location, ↓

include and exclude opt

ad schedule (schedule the ads per day and hours).

device (computer, mobile, tablet, tv screen).

filter audience using audience segment.

Types of PPC ads, grouping

(search ads, display ads, shopping ads, youtube ads, App ads

+ E-commerce (business page) (conversion page).

+ Local promotion (service page) (conversion)

MOFG, GMA, BMA

Note,

+ lead generation (service based) (conversion)
form, phone call.

+ Ecommerce (product based) (conversion value).

+ google ads ranking

→ Quality Score. (CTR, ad relevance, landing page relevance, budget)

view and impression (clicking and watch).

include an

basic PPC metrics.

1. cost 2. impression

3. clicks

4. CPC (cost per click)

5. CTR (click through rate)

6. clicks / impression

(Ecommerce).

8) Conversion value.

5. conversions \rightarrow CPC (cost per conversions).

(low)

CPA (cost per Acquisition).

(4) $(\%) \text{ Conversion rate} = \text{conversion} / \text{click}$

9) ROAS (Revenue on Ad Spend) in (%).

Conversion value / cost $\times 100$.

11. Search lost 15 (Rank)

12. Search lost 15 (Budget).

13. Quality score (QS) (out of 10).

Campaign Optimizing goals.

১০

plants

Search form.
(Negative Search Term list) → filtering the correct set of
1) Campaign → Ad group → key words & ads and assets.
Campaign overview PPC. the search

with and without negative
+ pop up page