

LIVE

จัดอบรมออนไลน์

- Vue 3 with TypeScript
- Vue 3 with Vite 3 in Deep
- Vue Composition API in Deep
- VueUse Library
- Pinia
- Vuetify
- Vue with Firebase API
- Ant Design
- Styled Components
- Deploying to production

🗓 11-12,18-19,25-26 November 2022
⌚ 20.00-23.30 PM



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน

Vue.js 3 ขั้นสูง



อาจารย์สาเมต
สถาบันไอทีจีเนียส

6 วัน
24 ชั่วโมงเต็ม
3



อ.สา米ตร โกยม

บริษัทโนเกียโซลูชันส์ เทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยี
พระจอมเกล้าพระนครเหนือ

[iamsamitdev](#)

[iamsamit](#)

[itgenius.co.th](#)

Frontend

Angular, React, NextJS, VueJS, NuxtJS, Bootstrap, TailwindCSS

Backend

PHP, Python, Java, NodeJS, ASP.net

Database

MySQL, PostgreSQL, MS SQL Server, MongoDB

Mobile

Java, Kotlin, Objective C, Swift, Cordova, ionic, Flutter

DevOps

Git, Github, Gitlab, Docker, Kubernetes



แบบทดสอบออนไลน์

quizizz.com/join



Course Outline

- Section 1: บทกวดการเขียน TypeScript
- Section 2: การทำงานกับ Vue 3 ร่วมกับ Vite 3 ขั้นสูง
- Section 3: การใช้งาน Vue Composition API ขั้นสูง
- Section 4: การประยุกต์ใช้งาน VueUse Library
- Section 5: จัดการ State management ด้วย Pinia
- Section 6: การใช้งาน Vuetify -Vuetify
- Section 7: การใช้งาน Vue ร่วมกับ Firebase API
- Section 8: การออกแบบ UI ด้วย Ant Design
- Section 9: การทำงานกับ Styled Components
- Section 10: การใช้งาน CKEditor 5 ร่วมกับ Vue.js 3
- Section 11: การ deployed project ขึ้นใช้งานจริงบน Production



ดาวน์โหลดเอกสารประกอบการอบรม

 Vue.JS3_Document

Day
1



อาจารย์สามิตร
สถาบันไอทีจีเนียส

Vue.js 3 ขั้นสูง

วัน
6 24
ชั่วโมงเต็ม

🗓 11-12,18-19,25-26 November 2022
⌚ 20.00-23.30 PM



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน

3



Frontend Web Developer Roadmap 2022



Start Here Teams We're Hiring

Search

Login

Sign Up

PARTNER Get the latest Frontend news from our sister site [TheNewStack.io](#)

>Create your own version of this roadmap

Frontend Developer

Step by step guide to becoming a modern frontend developer in 2024

← All Roadmaps

Share

Download

Suggest Changes

0% DONE

0 completed · 0 in progress · 0 skipped · 123 Total

Track Progress



What is Frontend Development?

- Personal Recommendation / Opinion
- Alternative Option - Pick this or purple
- Order in roadmap not strict (Learn anytime)
- I wouldn't recommend

Front-end

Internet

For resources and other roadmaps

<https://roadmap.sh>

Are you just getting started?

Visit the Beginner Version

How does the internet work?

What is HTTP?

What is Domain Name?

What is hosting?

Learn the basics

TIP Right-click on a topic to mark it as done. [Learn more.](#)

Writing Semantic HTML



Vue.JS Web Developer Roadmap 2022



Start Here Teams We're Hiring

Search

Login

Sign Up

Vue Developer

Everything that is there to learn about Vue and the ecosystem in 2024.

← All Roadmaps

Share

Download

Suggest Changes

0% DONE

0 completed · 0 in progress · 0 skipped · 67 Total

Track Progress

Related Roadmaps

- React Roadmap
- Angular Roadmap

JavaScript Roadmap

Find the detailed version of this roadmap along with resources and other roadmaps

<http://roadmap.sh>

Vue.js

Single File Components

Component Registration

Props Events

Attribute Inheritance

Vue CLI

Components

Slots

Directives

Fundamental Topics

App Configurations

Rendering Lists

Conditional Rendering

Lifecycle Hooks

Forms Handling

TIP Right-click on a topic to mark it as done. [Learn more.](#)



การเตรียมเครื่องมือ

Advanced Vue.JS 3

Tool and Editor

1. Visual Studio Code with Extension
2. Node.js 14+
3. Git 2.3+

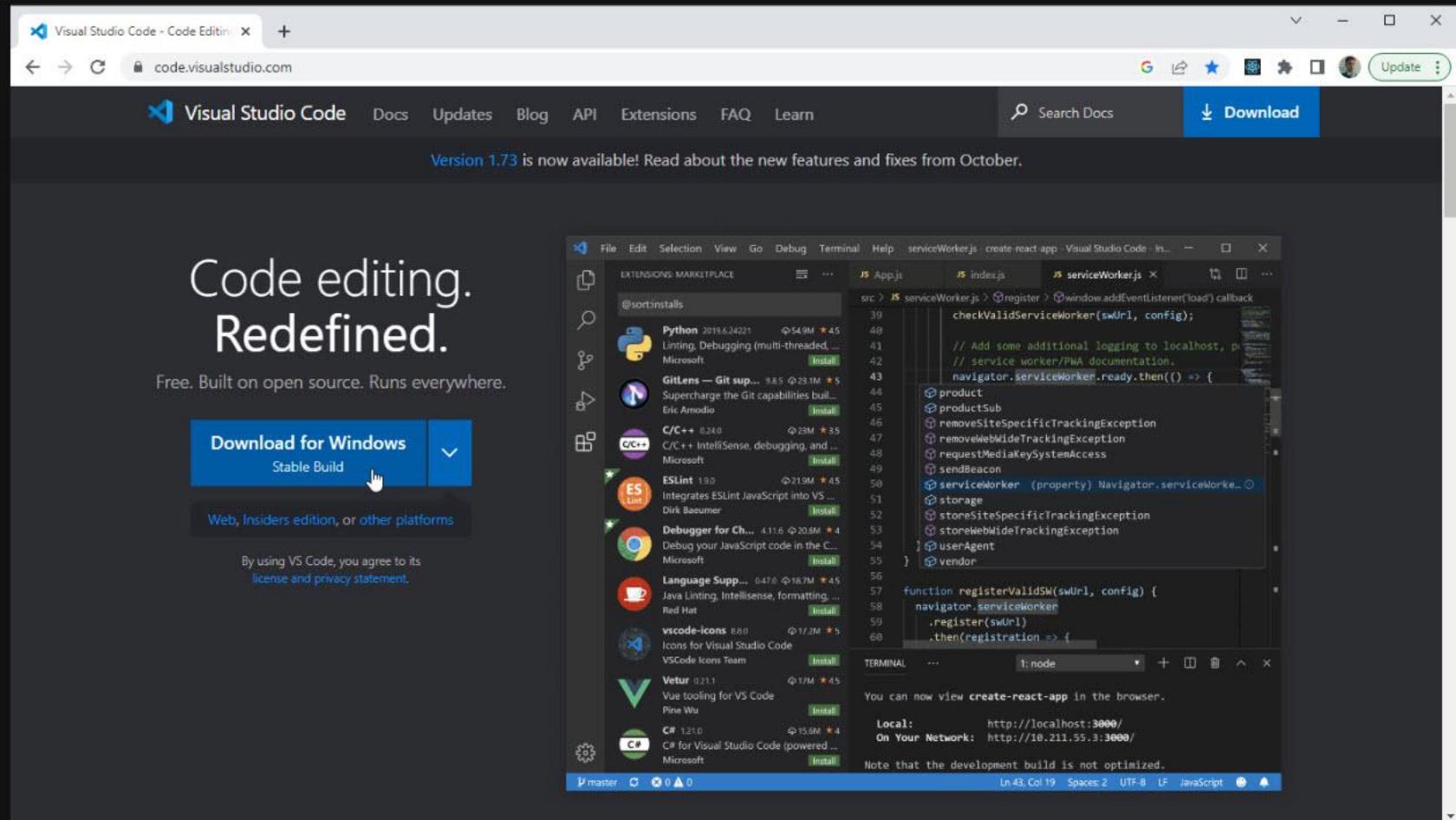




ຕິດຕັ້ງ Visual Studio Code

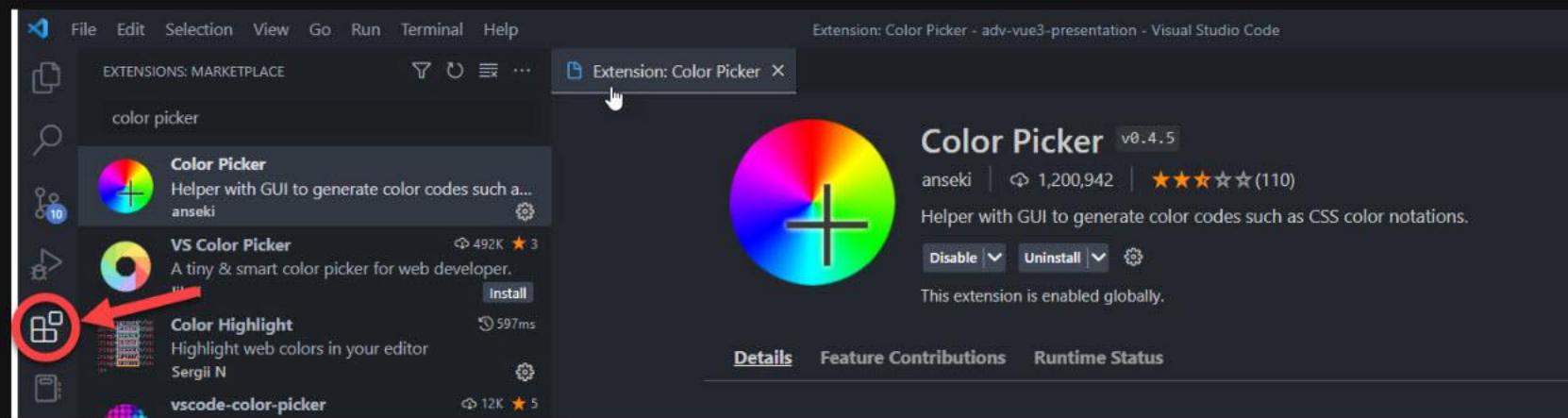


ดาวน์โหลด Visual Studio Code



เข้าไปดาวน์โหลด Visual Studio Code ได้ที่ <https://code.visualstudio.com>

การติดตั้งส่วนเสริม (Extension) ของ Visual Studio Code



คลิกที่รายการส่วนเสริม (Extensions) ด้านซ้ายมือแล้วทำการค้นหาดังภาพ



```

5   var dest
6   while(des
7     var
8       offs=
9       i=off
10    while i
11      dest.

```

Extension ที่แนะนำสำหรับ Vue.js

Extension พื้นฐาน

- 1 [Color Picker](#) by anseki
- 2 [Color Highlight](#) by sergiin
- 3 [AutoFileName](#) by jerryHong
- 4 [Material Icon Theme](#) by philipp kief
- 5 [JavaScript \(ES6\) Code Snippets](#) by charalampos karypidi
- 6 [Path Intellisense](#) by christian kohler
- 7 [Es6-String-HTML](#) by tobermory
- 8 [Sass](#) by syler
- 9 [Rainbow Brackets](#) by 2gua
- 10 [One Dark Pro](#) by binaryify

Extension for Vue.js

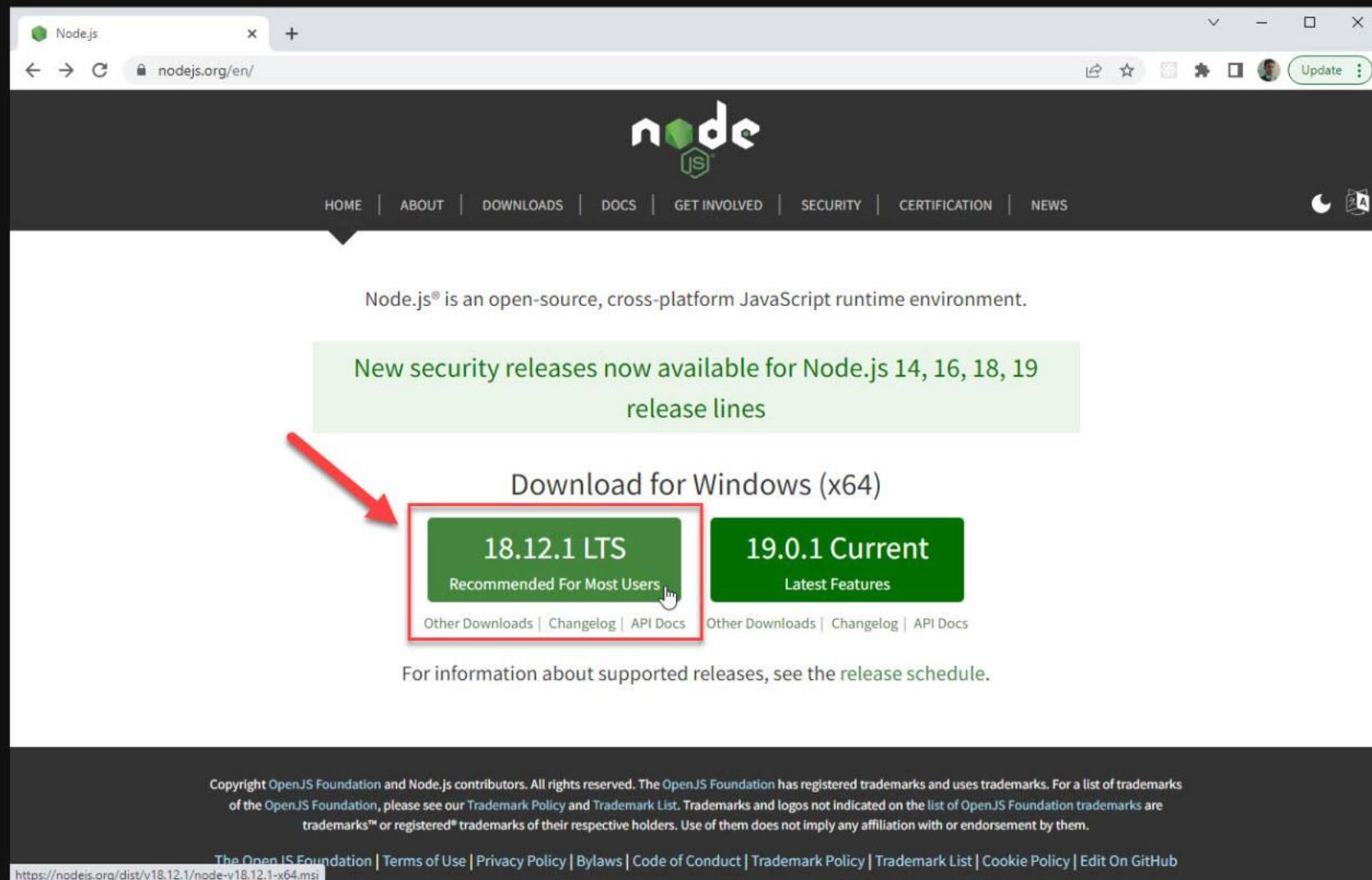
- 1 [Vue Language Features \(Volar\)](#) by vue
- 2 [TypeScript Vue Plugin \(Volar\)](#) by vue
- 3 [Vue 3 Snippets](#) by hollowtree
- 4 [Vue VSCode Snippets](#) by sarah.drasner



ติดตั้ง Node.JS

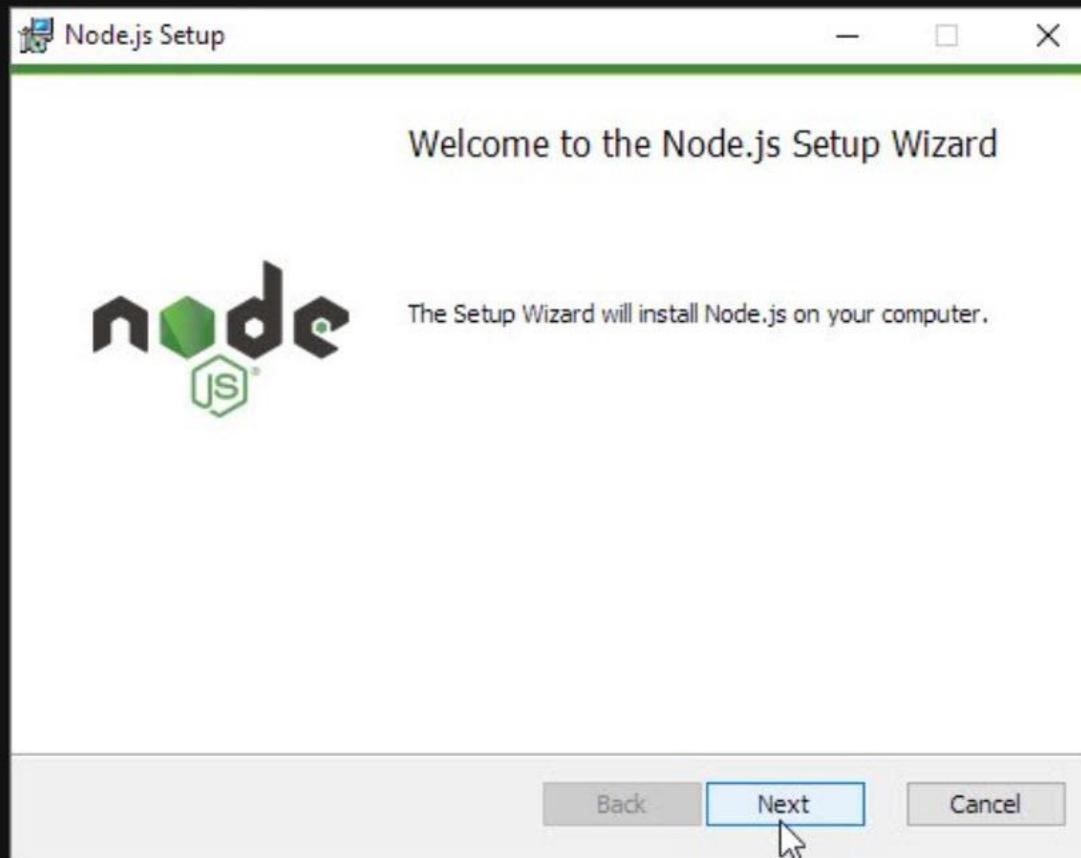


Download Node.js V.18.x

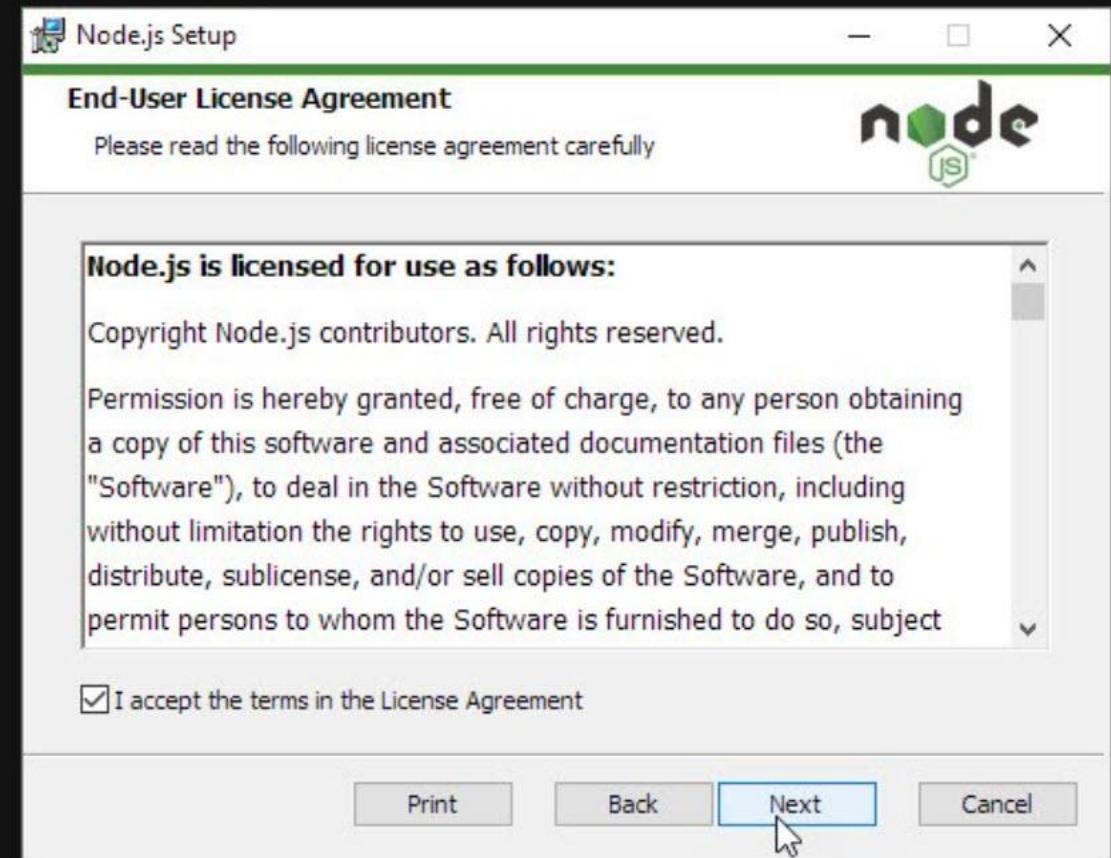


เข้าไปดาวน์โหลด Node.js ล่าสุดที่ <https://nodejs.org/en/>

Step 1



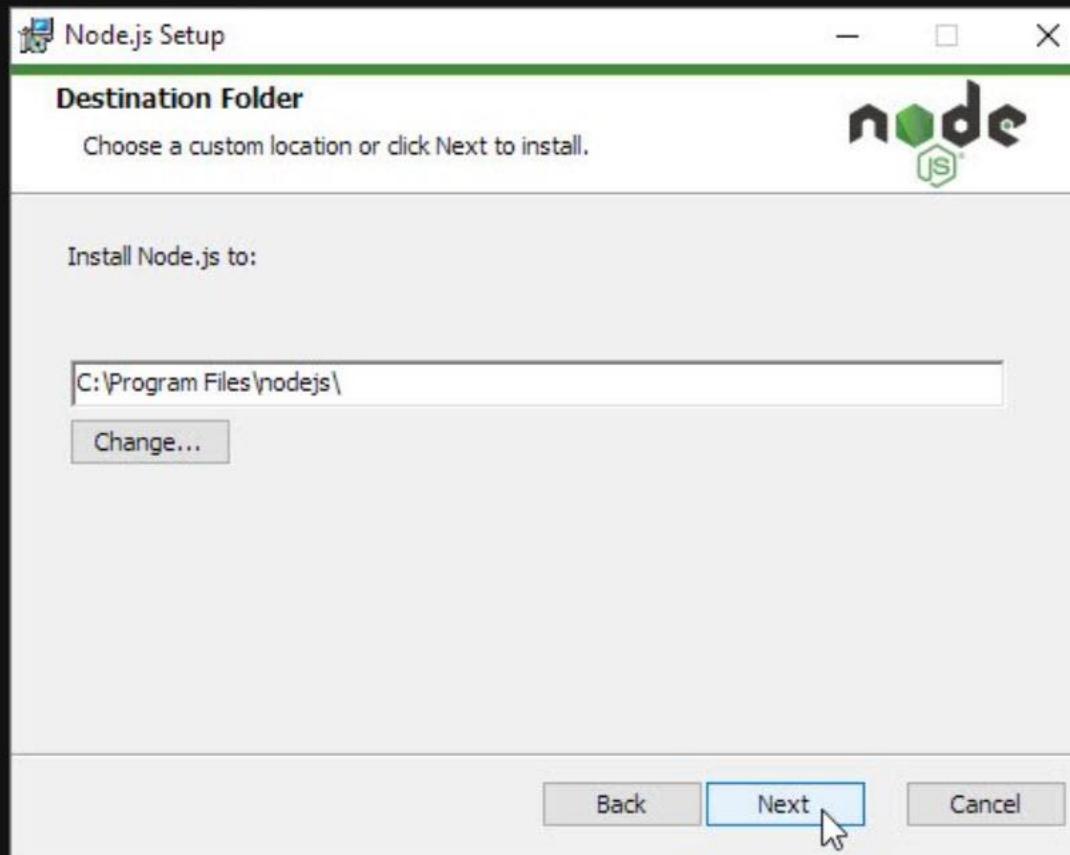
Step 2



คลิก Next

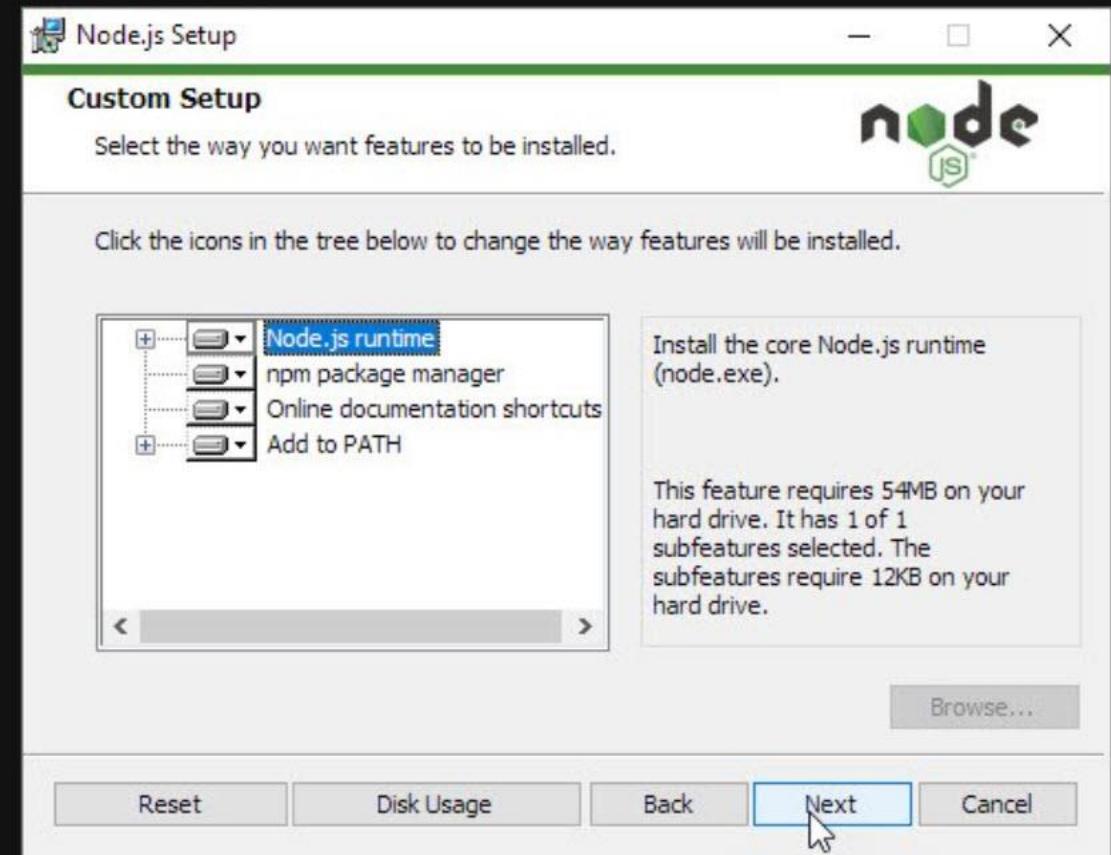
คลิก Next

Step 3



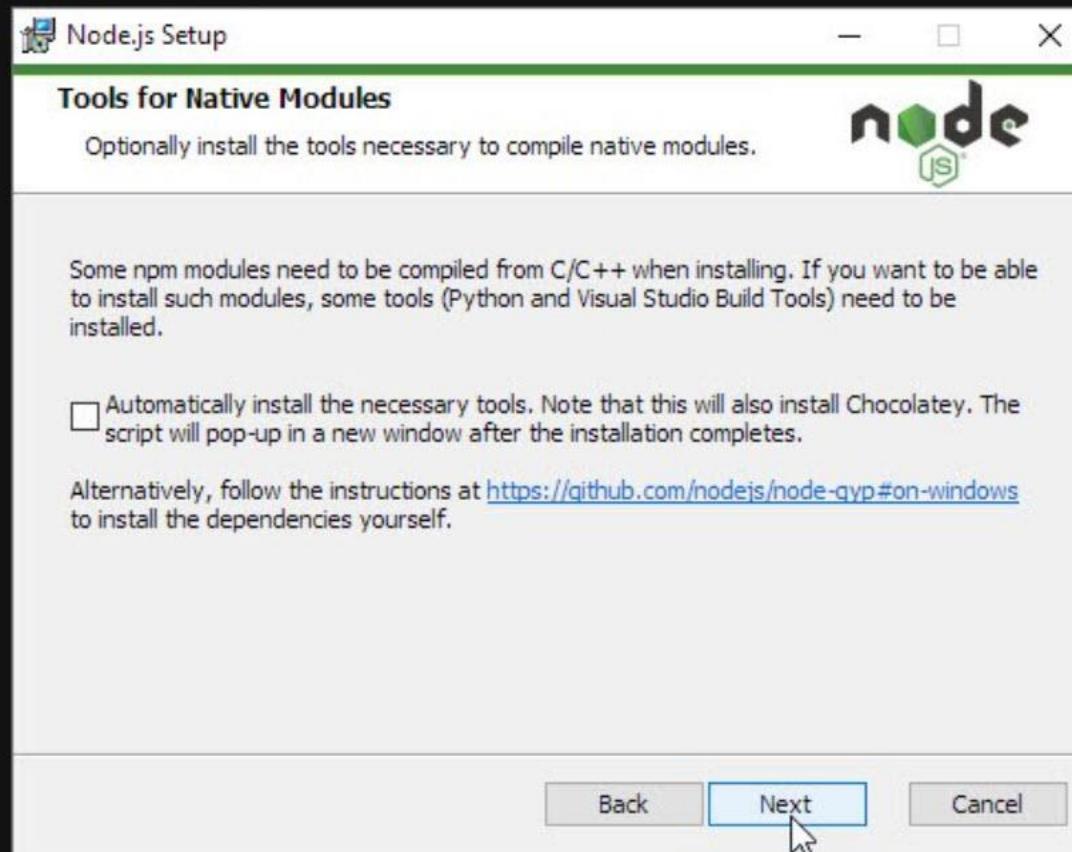
เลือก Path ที่ติดตั้ง (ไว้ตามค่าเริ่มต้น) คลิก Next

Step 4



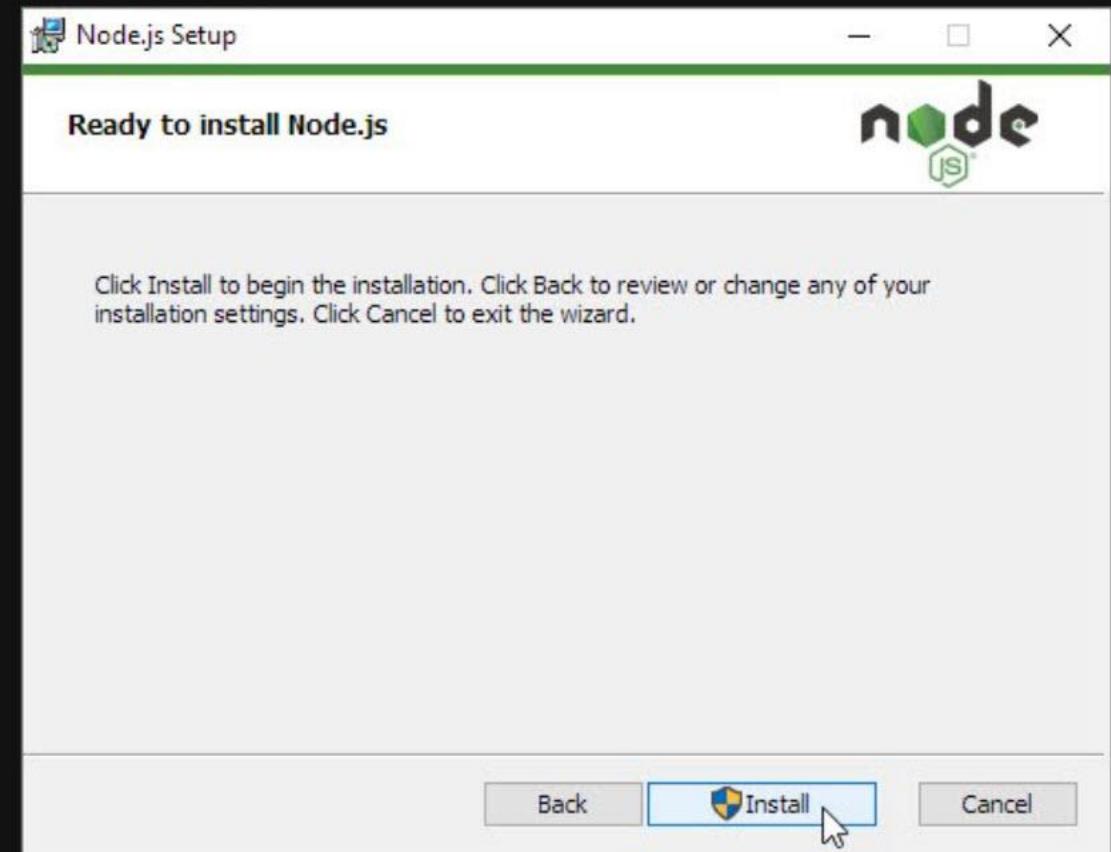
คลิก Next

Step 5



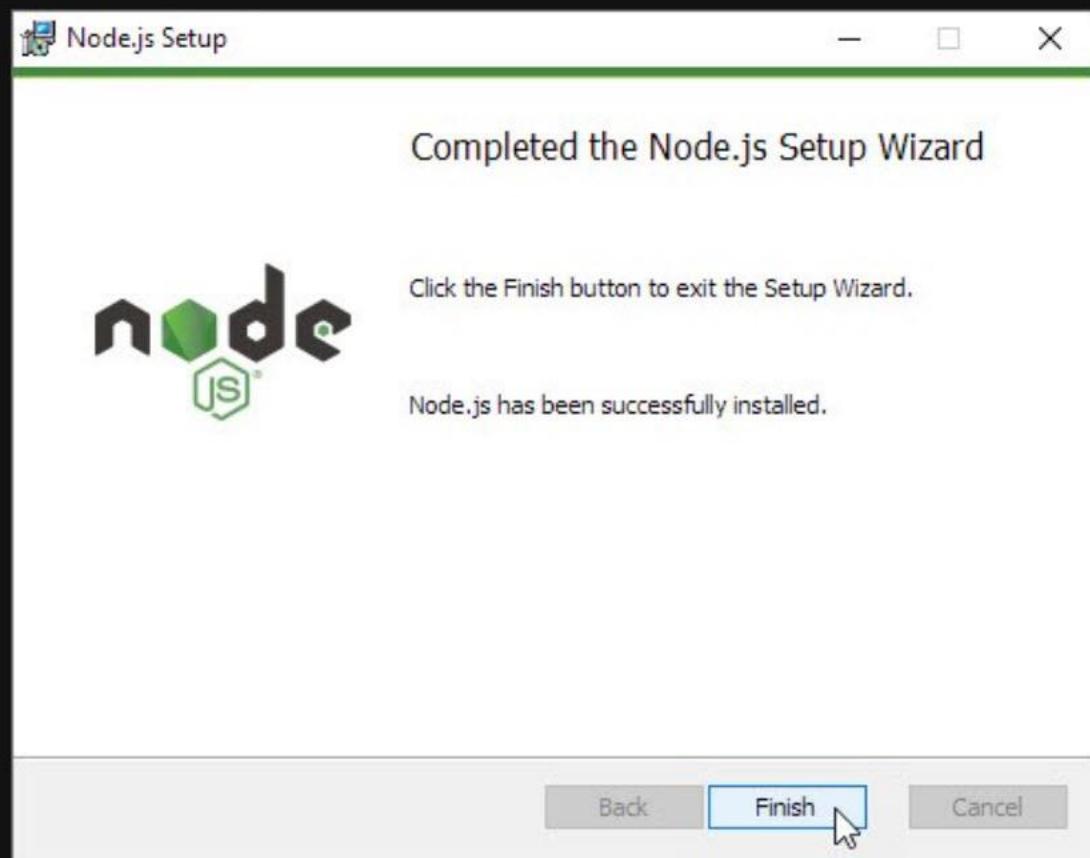
คลิก Next

Step 6



คลิก Install

Step 7



คลิก Finish

ทดสอบหลังติดตั้ง Node.js

```
1 node -v
```

```
1 npm -v
```

```
1 npx -v
```

```
1 where node
```





ติดตั้ง Git





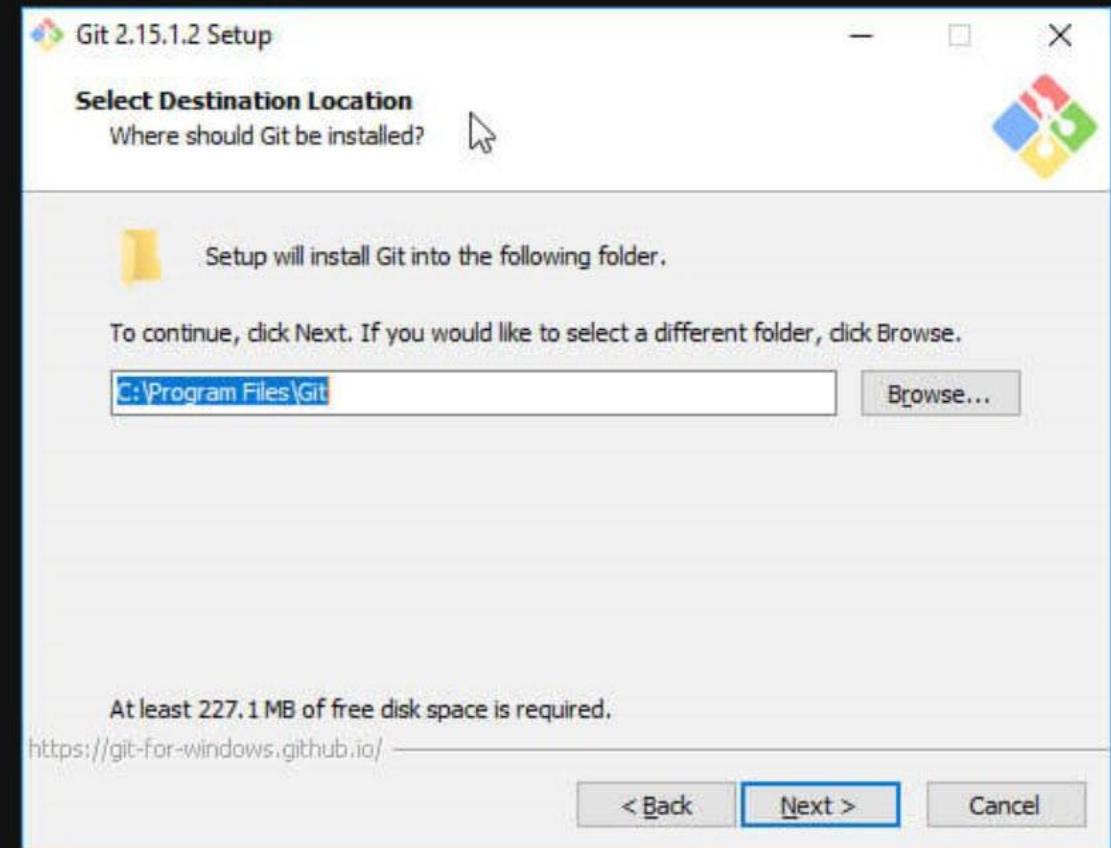
ติดตั้ง Git



Step 1



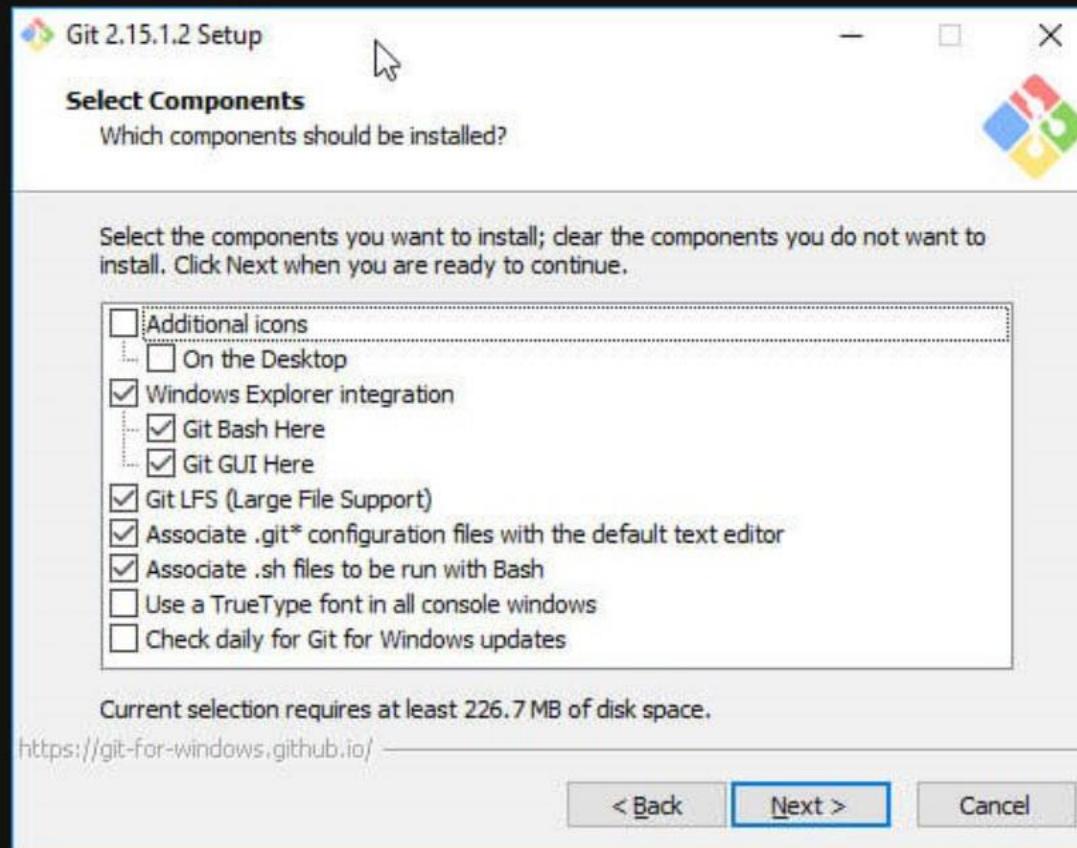
Step 2



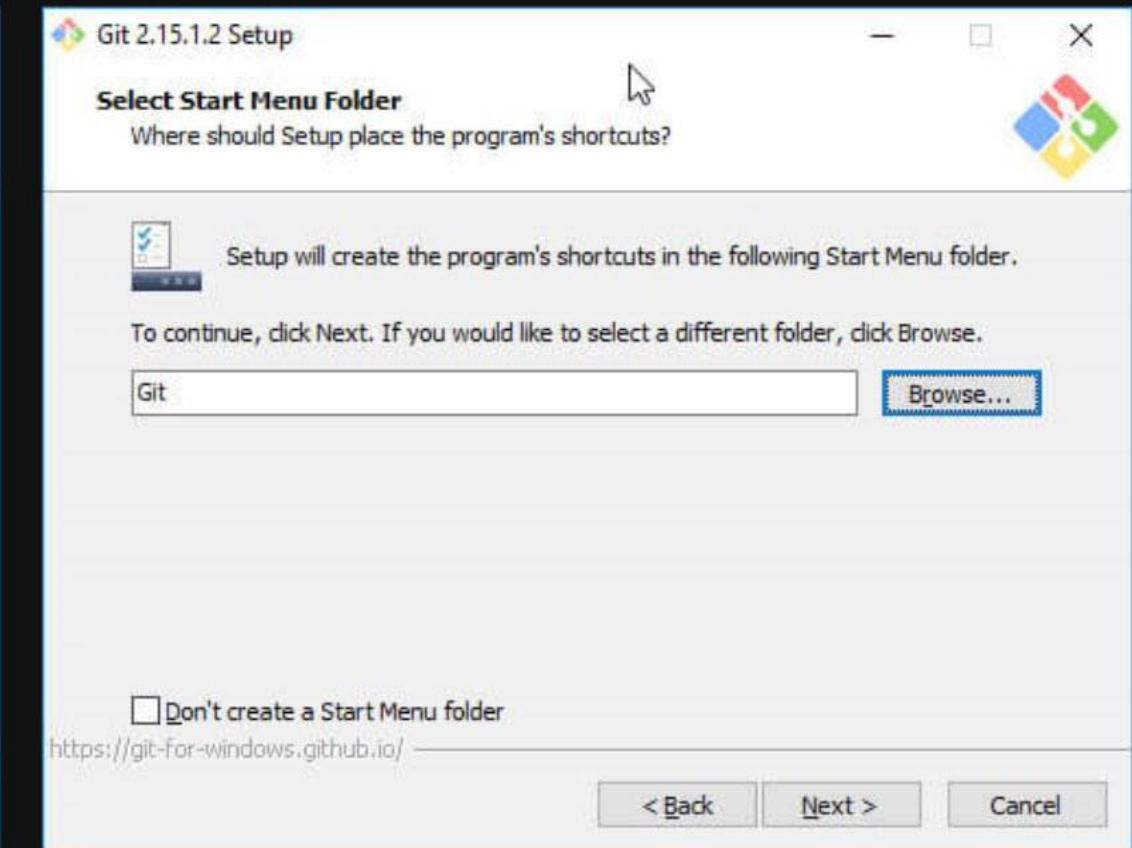
คลิก Next

เลือก Path (ไว้ตามค่าเริ่มต้น) คลิก Next

Step 3



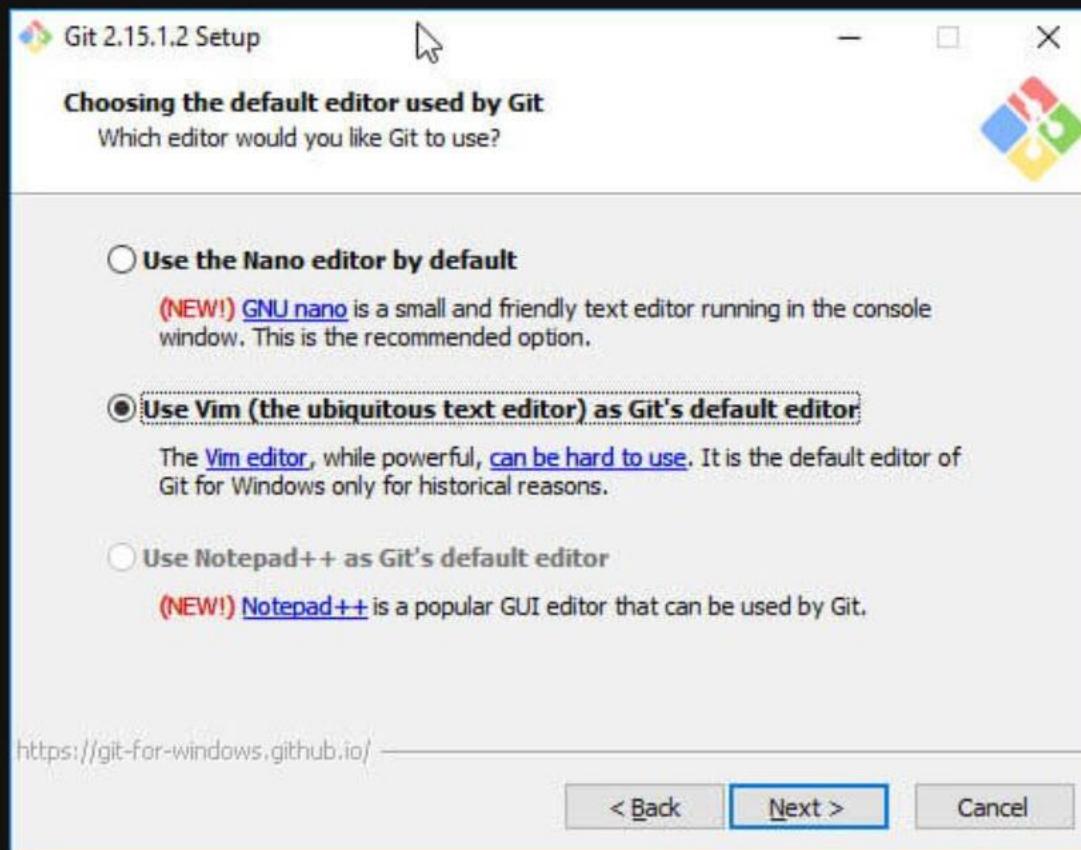
Step 4



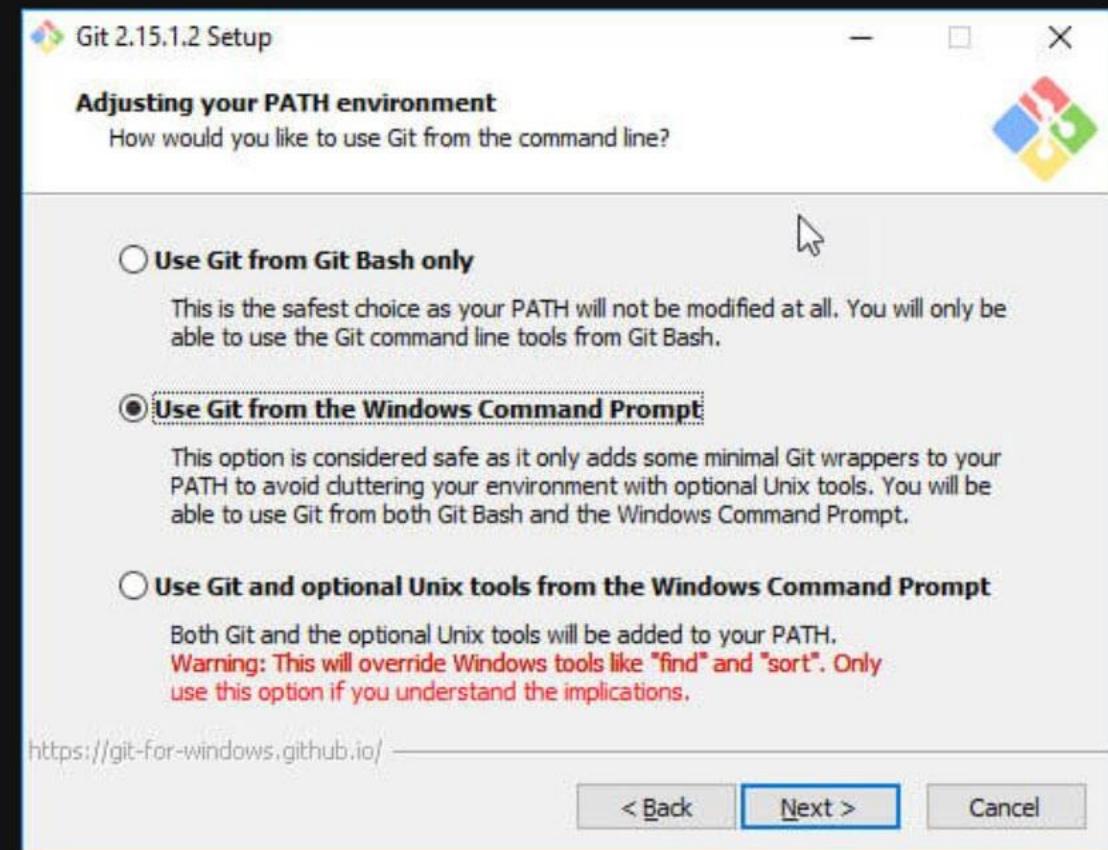
คลิก Next

คลิก Next

Step 5



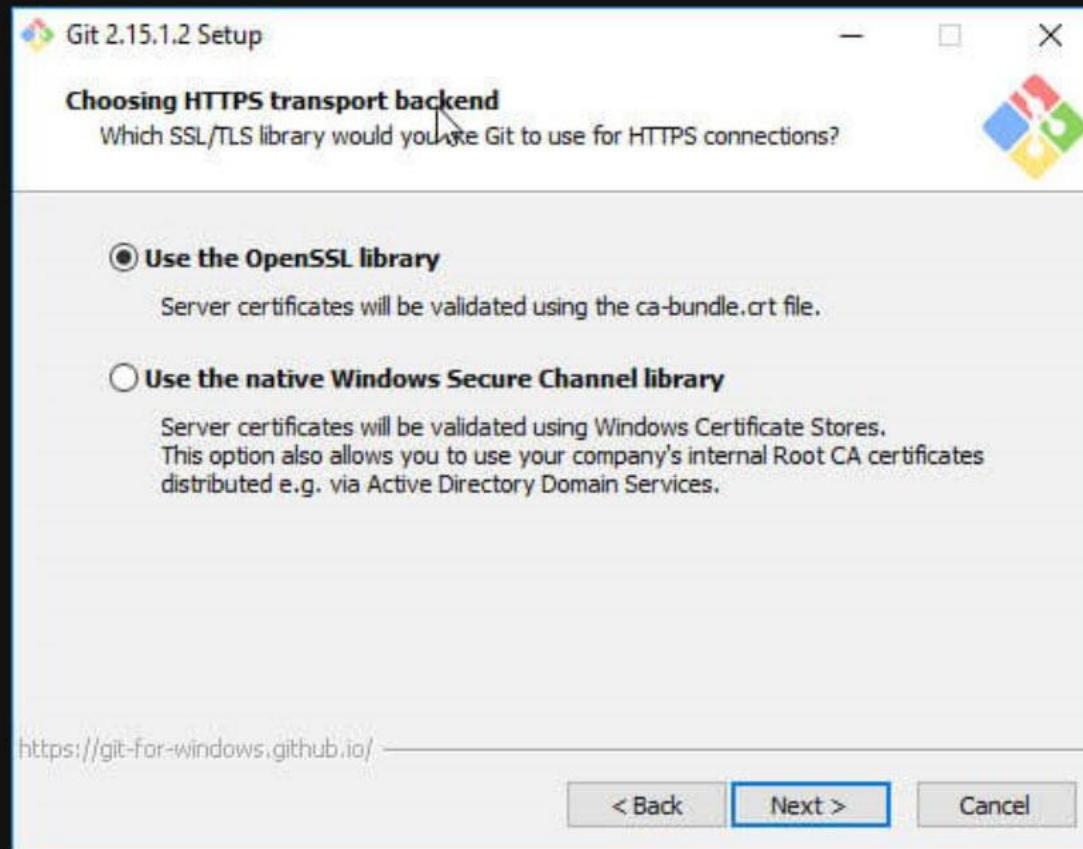
Step 6



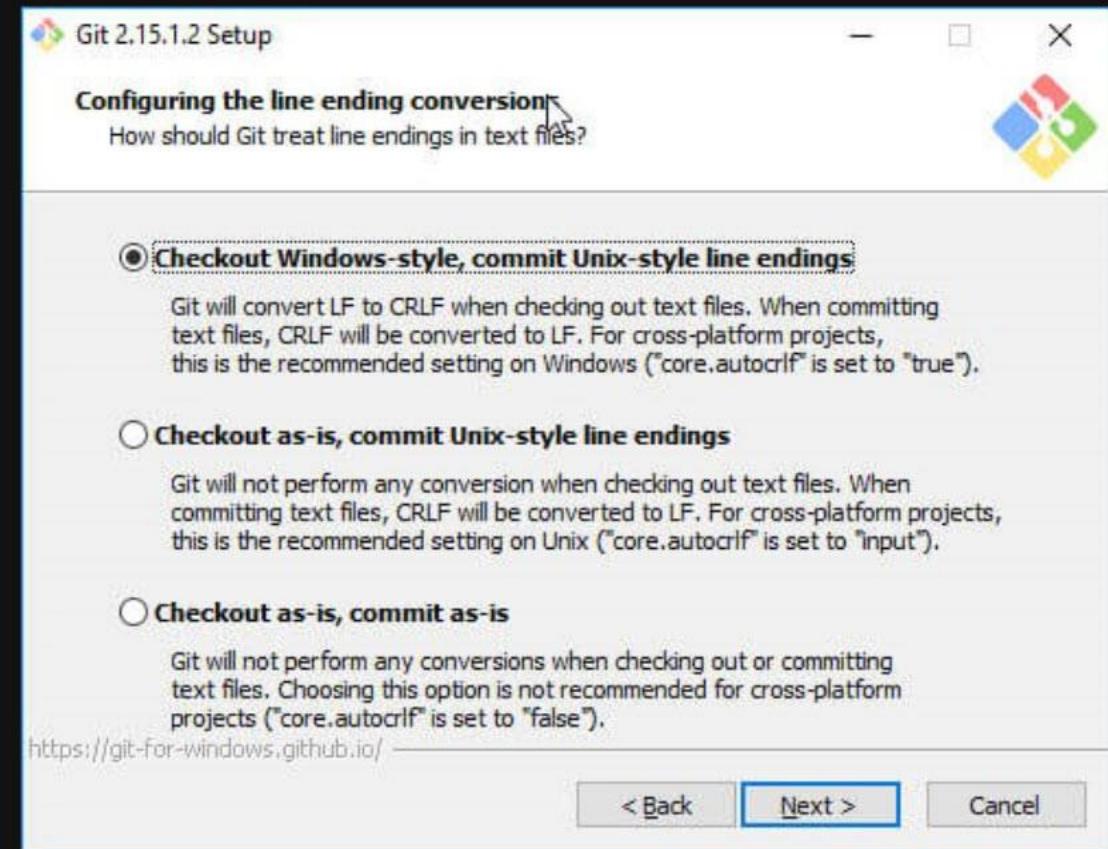
คลิก Next

คลิก Next

Step 7



Step 8



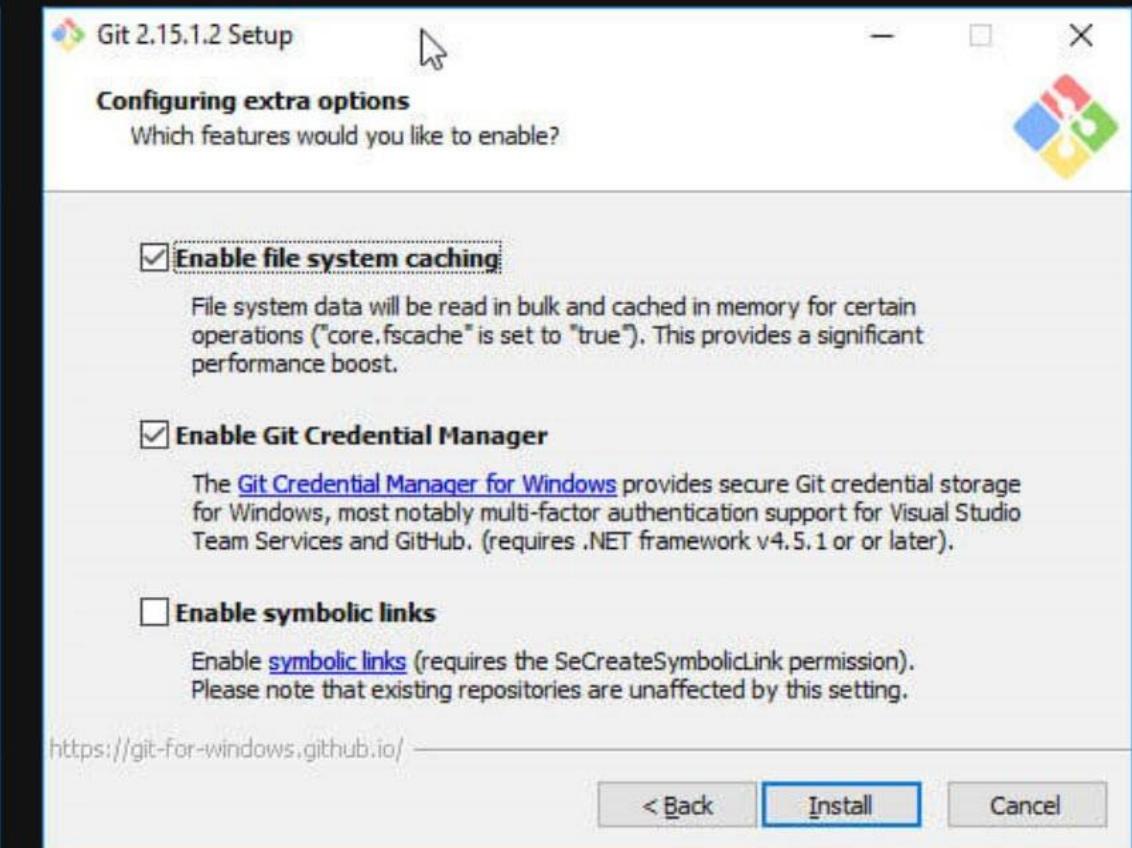
คลิก Next

คลิก Next

Step 9

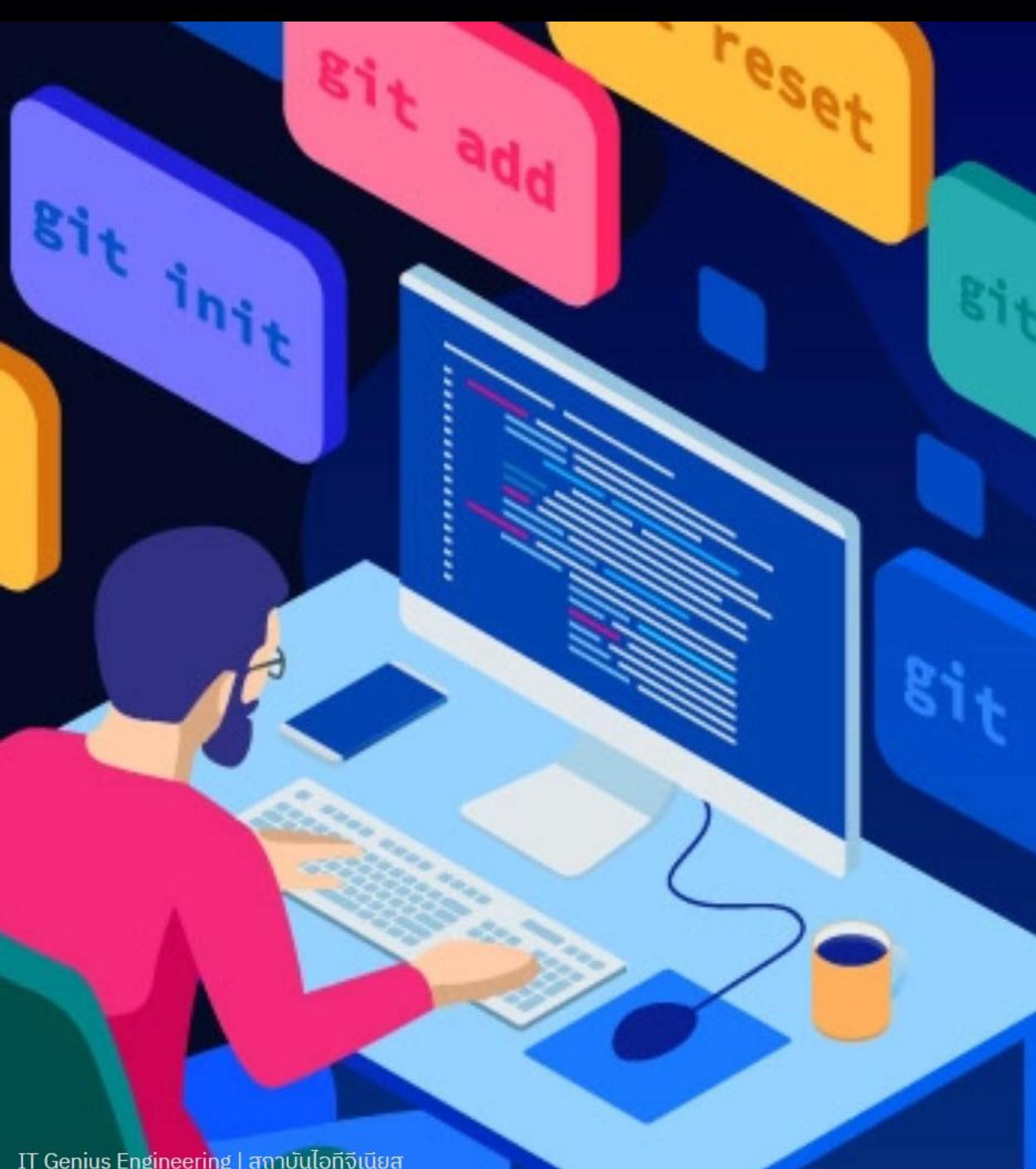


Step 10



คลิก Next

คลิก Install



ทดสอบหลังติดตั้ง Git

```
1 git version
```

Config Git First Time

```
1 git config --global user.name "John Doe"  
2 git config --global user.email johndoe@example.com  
3 git config --global init.defaultBranch main
```

ตรวจสอบค่าที่ Config ไว้

```
1 git config --list --global
```

Section 1

nunกับ TypeScript

- Types
- Functions
- Complex Types
- Union Types
- Type Narrowing
- Class and Object Type
- Interface



TypeScript Topic

1. Types
2. Functions
3. Complex Types
4. Union Types
5. Type Narrowing
6. Class and Object Types
7. Interface



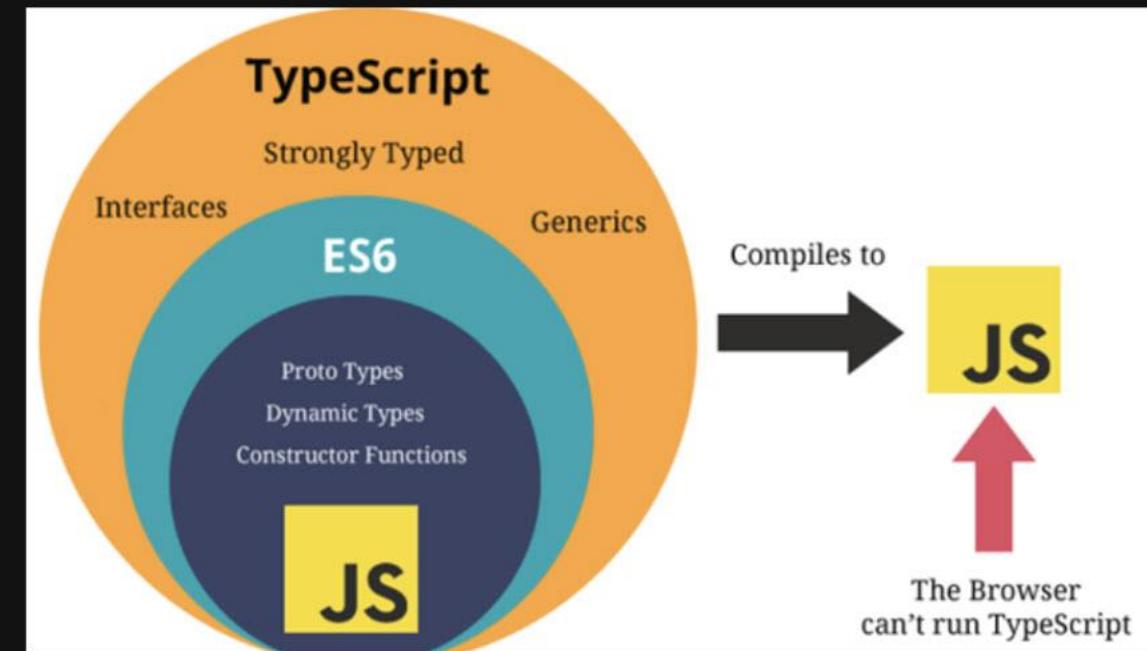
TypeScript คืออะไร ?

TypeScript เป็นภาษาที่ถูกออกแบบมาเพื่อช่วยให้การเขียนโปรแกรมในภาษา JavaScript มีความเข้มงวดในการใช้งานประเภทข้อมูล

TypeScript เป็นภาษาที่ในที่สุดแล้วจะถูกแปลงเป็นภาษา JavaScript ก่อนเริ่มทำงาน การใช้ภาษา TypeScript สามารถช่วยลดข้อผิดพลาดเดี่ยวๆ กับประเภทข้อมูลในโปรแกรมได้ มีประโยชน์มากในการพัฒนาระบบที่ซับซ้อนและมีขนาดใหญ่ เนื่องจากมันสามารถช่วยเพิ่มประสิทธิภาพของโปรแกรม ดูและรักษา และการพัฒนาโปรแกรมจะทำได้ง่าย

TypeScript นั้นเป็น superset ของ JavaScript ซึ่งหมายความว่า อะไรก็ JavaScript ทำได้ TypeScript ก็ทำได้ เช่นกัน

- สิ่งแรกที่ควรรู้จักรือ เรายังเขียนไฟล์ TypeScript ด้วยนามสกุล **.ts**
- จากนั้นจะทำการรันโค้ด TypeScript ผ่านตัว transpiler ตัว transpiler จะคอยกำหน้าที่เช็คໄວຍกรณ์ความถูกต้อง ตามมาตรฐานของ TypeScript ก่อนจะแสดงข้อผิดพลาดออกมา
- จากนั้นตัว transpiler จะทำการแปลงโค้ด TypeScript ให้เป็น JavaScript (**.js**) ธรรมดามาเพื่อให้ทำงานได้แบบราวนะกันไป



TypeScript เป็น superset ของ JavaScript

TypeScript คืออะไร ?

TypeScript เป็น superset ของ JavaScript ที่มีความสามารถเพิ่มเติม เช่น การระบุประเภทของตัวแปร ทำให้สามารถตรวจสอบความถูกต้องของโค้ดได้ รวมถึงยังเพิ่มความสามารถใหม่ๆ จากตัวนั้นเข้ามาอีกด้วย

ลองเขียน TypeScript โค้ดตัวอย่างนี้

```
1 let firstName = 'YourName'
```

ตัว transpiler ของ TypeScript จะทำการแปลงเป็น JavaScript ได้โค้ดออกมากดังนี้

```
1 "use strict";
2 let firstName = "YourName"
```

เบื้องต้นเราจะเห็นว่าบันได้ผลลัพธ์เดียวกัน เพราะโดยพื้นฐาน TypeScript รองรับคำสั่งของ JavaScript ก็จะหมดอยู่แล้ว

คราวนี้เรามาดูเรื่องต่อไปกันคือ Type Inferences

The screenshot shows the official TypeScript playground website. At the top, there's a blue header bar with the 'TypeScript' logo, a search bar labeled 'Search Docs', and navigation links for 'Download', 'Docs', 'Handbook', 'Community', and 'Tools'. Below the header, there's a dark-themed interface with a toolbar containing 'Playground', 'TS Config', 'Examples', 'Help', 'Version.....', 'Run', 'Export', 'Share', and a 'Settings' button. The main area is a code editor window where the following TypeScript code is written:

```
1 let firstName = 'YourName'
```

Below the code editor, the output section shows the generated JavaScript code:

```
1 "use strict";
2 let firstName = "YourName"
```

Type Inferences

โดยปกติภาษา JavaScript อนุญาตให้เรากำหนดค่าใน ๆ ที่ได้ให้กับตัวแปร มันดูยังดีนั่น แต่เมื่อเรามีตัวแปรจำนวนมากที่ไม่ได้กำหนดชนิดข้อมูลให้มันอย่างชัดเจน เมื่อถูกนำไปใช้งานจะมีข้อผิดพลาดบางอย่างตามมาบ่อยครั้ง

ทาง TypeScript จึงได้มีการกำหนด Type ให้กับตัวแปรก่อนนำไปใช้งาน โดยมี Type เเบ่งตัวมาจาก JavaScript ดังนี้

- boolean
- number
- null
- string
- undefined

ถ้าเราพยายามเปลี่ยนค่า value ให้กับตัวแปรต่างชนิดกัน ภาษา TypeScript จะตรวจ Error ให้ก่อนจะทำการ Run เช่น

```
1 let order = 'first'
2 order = 1
```

The screenshot shows the TypeScript Playground interface. At the top, there's a blue header bar with the 'TypeScript' logo, a search bar labeled 'Search Docs', and navigation links for 'Download', 'Docs', 'Handbook', 'Community', and 'Tools'. Below the header, the main menu includes 'Playground', 'TS Config ▾', 'Examples ▾', 'Help ▾', 'Version..... ▾', 'Run', 'Export ▾', 'Share', and 'Settings'. The main area is a dark-themed code editor containing the following TypeScript code:

```
1 let order = 'first'
2 order = 1
```

Type Shapes

TypeScript เหมือนอับดุล คาดเดาได้ว่าตัวแพร่หรือ
ออบเจกต์นั้น ๆ เป็น Type อะไร

เช่นใน JavaScript มีฟังก์ชัน `.length` สำหรับนับ
จำนวน String หรือฟังก์ชัน `.toLowerCase()`
สำหรับเปลี่ยนเป็นตัวอักษรตัวเล็กทั้งหมด
TypeScript สามารถเรียกใช้งานได้กันที

ตัวอย่าง

```
1 "Awesome".length // 7
2 "MY Love Song".toLowerCase() // "my
```

หรือกรณีเราเรียกชื่อฟังก์ชันผิด ก็จะตรวจสอบได้
กันที

```
1 "MY Love Song".tolowercase()
2 // Property 'tolowercase' does not exist on type 'string'.
3 // Did you mean 'toLowerCase'?
```

ลองแก้ไขโปรแกรมด้านล่างนี้ให้ถูกต้อง

The screenshot shows the TypeScript Playground interface. At the top, there's a navigation bar with links for Download, Docs, Handbook, Community, Tools, and a search bar labeled "Search Docs". Below the navigation bar, there are tabs for "Playground", "TS Config", "Examples", and "Help". Under the "Playground" tab, there are dropdown menus for "Version", "Run", "Export", and "Share", along with a "Settings" link. The main area contains a code editor with the following content:

```
1 "Awesome".length // 7
2 "MY Love Song".toLowerCase() // "my"
```

Below the code editor, an error message is displayed:

```
1 "MY Love Song".tolowercase()
2 // Property 'tolowercase' does not exist on type 'string'.
3 // Did you mean 'toLowerCase'?
```

The playground interface has a dark theme with light-colored text and code blocks.

Any

หาก TypeScript ไม่สามารถคาดเดาได้ว่าชนิดข้อมูลที่กำหนดให้เป็นอะไร ค่าเริ่มต้นจะถือว่าเป็นชนิดข้อมูลแบบ any ซึ่งผู้ใช้งานสามารถกำหนด type แบบ any เข้าไปเองก็ได้ดังนี้

ตัวอย่างกรณีไม่กำหนดชนิดข้อมูลถือว่าเป็น any

```
1 let onOrOff
2 onOrOff = 1
3 onOrOff = false
```

หรือจะกำหนด Type ชนิด any เข้าไปตรงๆ ก็ได้

```
1 let onOrOff: any
2 onOrOff = 1
3 onOrOff = false
```

ลองแก้ไขโปรแกรมด้านล่างนี้ให้ถูกต้อง

```
1 let guess:string = "Prawit"
2 guess = 65
3 console.log(guess)
```

The screenshot shows the TypeScript playground interface. At the top, there's a navigation bar with links for Download, Docs, Handbook, Community, Tools, and a search bar labeled 'Search Docs'. Below the navigation bar, there are several dropdown menus: 'Playground', 'TS Config ▾', 'Examples ▾', 'Help ▾', 'Version..... ▾', 'Run', 'Export ▾', 'Share', and 'Settings'. The main area of the playground contains the code examples from the previous text blocks.



การใช้งาน TypeScript บน Visual Studio Code

ติดตั้ง TypeScript CLI

```
1 npm install -g typescript
```

เช็คเวอร์ชัน TypeScript

```
1 tsc -v
```

ติดตั้ง ts-node

```
1 npm install -g ts-node
```

เช็คเวอร์ชัน TypeScript

```
1 ts-node -v
```

โครงสร้างโปรเจกต์ TypeScript

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "LEARN-TYPESCRIPT" with files and folders: dist, node_modules, src, .eslintrc, index.html, package-lock.json, package.json, and tsconfig.json.
- Terminal:** Displays the contents of the tsconfig.json file, which includes compiler options like "incremental" and "composite", and language and environment settings like "target: es2016".

```

{
  "compilerOptions": {
    /* Visit https://aka.ms/tsconfig to read more about this file */
    /* Projects */
    // "incremental": true, /* allow for incremental compilation of projects. */
    // "composite": true, /* TypeScript project to be used with project references. */
    // "tsBuildInfoFile": "./tsbuildinfo", /* tsbuildinfo incremental compilation file. */
    // "disableSourceOfProjectReferenceRedirect": true, /* instead of declaration files when referencing composite */
    // "disableSolutionSearching": true, /* multi-project reference checking when editing. */
    // "disableReferencedProjectLoad": true, /* loaded automatically by TypeScript. */
    /* Language and Environment */
    "target": "es2016", /* version for emitted JavaScript and include compatible library declarations */
    // "lib": [], /* library declaration files that describe the target runtime */
    // "jsx": "preserve", /* generated. */
  }
}

```

TypeScript บน Visual Studio Code

1. สร้าง TypeScript Config File

```
1 tsc --init
```

2. สร้างไฟล์ index.html

3. Run typescript file

```
1 tsc
```

4. ลิงก์ไฟล์ .js ที่หน้า index.html และทดสอบดูผลการทำงาน

Functions

TypeScript Parameter Type Annotations

Function parameters may be given type annotations with the same syntax

```

1 function greet(noun: string) {
2   console.log(`Hello, ${noun}!`)
3 }
4 greet('World') // Prints: Hello, World
5 greet(2020) // Argument of type 'number' is not assignable to parameter of type 'string'

```

TypeScript Optional Parameter

Sometimes we would like to skip providing values to function calls. We can declare some parameters in a function to be optional if a value is not provided for all arguments in a function

```

1 function greet(name?: string) {
2   console.log(`Hello, ${name || 'stranger'}!`)
3 }
4 greet() // Prints: Hello, stranger!

```

TypeScript Default Parameters

If we assign a function parameter to have a default value, TypeScript will infer the parameter type to be the same as the default value's type

```

1 function exponentiation(power = 1) {
2   console.log(4 ** power)
3 }
4
5 exponentiation() // Prints: 4
6 exponentiation(4) // Prints: 256
7 exponentiation(true); // Error: Argument of type 'true' is not assignable to type 'number'

```

TypeScript Inferring Return Types

By looking at the types of the values in a function's return statement, TypeScript can infer the return type of a function.

```

1 function factOrFiction() {
2   return Math.random() >= .5 ? 'true' : 'false'
3 }
4
5 const myAnswer : boolean = factOrFiction(); // Type 'string' is not assignable to type 'boolean'

```

Functions

TypeScript Void Return Type

If a function does not return any value, then you can specify void as a return type using type annotation

```
1  function sayHello(): void {  
2      console.log('Hello!')  
3  }
```

TypeScript Explicit Return Types

We can be explicit about what type a function returns by adding type annotation (: followed by the type) after a function's closing parenthesis

```
1  function trueOrFalse(value: boolean): boolean {  
2      if (value) {  
3          return true  
4      }  
5      return 'false' // Typescript Error: Type 'string' is not assignable to type  
6  }
```

Complex Types

TypeScript Type for One-dimensional Array

The type annotation for a one-dimensional array in TypeScript is similar to a primitive data type, except we add a `[]` after the type

```

1 // zipcodes is an array of strings
2 let zipcodes: string[] = ['03255', '02134', '08002', '03063']
3 // Pushing a number to zipcodes will generate an error
4 // Error: Argument of type 'number' is not assignable to parameter of type 'string'
5 zipcodes.push(90210)

```

Generic Type for One-Dimensional Array

Array, where T stands for the type

```

1 // zipcodes is an array of strings
2 let zipcodes: Array<string> = ['03255', '02134', '08002', '03063']
3
4 // Error: Argument of type 'number' is not assignable to parameter of type 'string'
5 zipcodes.push(90210)

```

TypeScript Type for Multi-dimensional Array

The type for a multi-dimensional array can be annotated by adding an extra `[]` for each extra dimension of the array

```

1 // one-dimensional arrays
2 let zipcodesNH: string[] = ['03255', '03050', '03087', '03063']
3 let zipcodesMA: string[] = ['02334', '01801']
4
5 // two-dimensional array
6 let zipcodes: string[][] = [zipcodesNH]
7
8 // Pushing a one-dimensional array to a two-dimensional array
9 zipcodes.push(zipcodesMA)
10 console.log(zipcodes) // prints [[{"03255", "03050", "03087", "03063"}, [{"02334", "01801"}]]

```

TypeScript Tuple Type

```

1 // This is an array
2 let header: string[] = ['Name', 'Age', 'Smoking', 'Salary']
3 // This is a tuple
4 let profile: [string, number, boolean, number] = ['Kobe', 39, true, 150000]

```

Union Types

TypeScript Union Type

TypeScript allows a flexible type called `any` that can be assigned to a variable whose type is not specific. On the other hand, TypeScript allows you to combine specific types together as a union type

```
1 let answer: any    // any type
2 let typedAnswer: string | number // union type
```

TypeScript Union Type Syntax

TypeScript lets you create a union type that is a composite of selected types separated by a vertical bar, `|`.

```
1 let myBoolean: string | boolean
2
3 myBoolean = 'TRUE' // string type
4 myBoolean = false // boolean type
```

TypeScript Union Type Narrowing

Since a variable of a union type can assume one of several different types

```
1 const choices: [string, string] = ['NO', 'YES']
2 const processAnswer = (answer: number | boolean) => {
3   if (typeof answer === 'number') {
4     console.log(choices[answer]);
5   } else if (typeof answer === 'boolean') {
6     if (answer) {
7       console.log(choices[1])
8     } else {
9       console.log(choices[0])
10    }
11  }
12}
13 processAnswer(true) // Prints "YES"
14 processAnswer(0) // Prints "NO"
```

ออบเจ็ค ในภาษา TypeScript

การประกาศออบเจ็ค

ออบเจ็ค เป็นประเภทข้อมูลแบบโครงสร้างที่เก็บข้อมูลในรูปแบบของ Key/Value โดยที่ Key ของออบเจ็คจะเป็น String ส่วน Value สามารถเป็นข้อมูลประเภทใดๆ ก็ได้ในภาษา TypeScript นี้ เป็นรูปแบบของการประกาศออบเจ็คในภาษา TypeScript

```

1 let objectName: objectType = {
2   key1: value1,
3   key2: value2,
4   ...
5 }
```

```

1 let user: object = {
2   id: 1,
3   name: "Samit",
4   dateOfBirth: new Date("1982-11-10")
5 };
6
7 console.log(user)
```

การเข้าถึงและอ่านค่าในเจ็ค

การดำเนินการพื้นฐานเดียวกับออบเจ็ค นั่นคือการกำหนดค่าและอ่านค่าในออบเจ็คผ่านทาง Key ของมัน

```

1 let song: object = {
2   name: "Brooklyn Boy",
3   artist: "Jeremy Zucker",
4   year: 2021
5 }
6
7 console.log(`Song: ${song.name}`)
8 console.log(`Artist: ${song.artist}`)
9 console.log(`Year: ${song.year}`)
10
11 song.year = 2025
12 console.log(`Changed year: ${song.year}`)
```

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

```

1 Song: Brooklyn Boy
2 Artist: Jeremy Zucker
3 Year: 2021
4 Changed year: 2025
```

ออบเจ็ค ในภาษา TypeScript

การสร้างประเภทข้อมูลของออบเจ็ค

ออบเจ็ค Literal มีประเภทข้อมูลเป็น object ที่เก็บออบเจ็คที่มีโครงสร้างเป็นแบบໄหนเด็ด อย่างไรก็ตาม เราสามารถใช้คำสั่ง type เพื่อกำหนดโครงสร้างให้กับออบเจ็คได้ นี่จะทำให้เราสามารถกำหนด Property ที่ออบเจ็คสามารถมี และประเภทข้อมูลที่แต่ละ Property สามารถมีได้

```

1  type userType = {
2      id: number,
3      name: string,
4      dateOfBirth: Date
5  }
6
7  let user: userType = {
8      id: 1,
9      name: "Samit",
10     dateOfBirth: new Date("1982-11-10")
11 }
```

เมетодจากคลาส Object

คลาส Object มีเมethodที่เราสามารถใช้สำหรับจัดการกับออบเจ็ค Literal ได้ แม้ว่าบันจะมีเมethodเป็นจำนวนมาก แต่เราจะแนะนำเมethodพื้นฐานที่ใช้งานบ่อยในภาษา TypeScript มาดูตัวอย่างการใช้งานเมethodสำหรับวิธีการค่าในออบเจ็ค

```

1  let country: object = {
2      name: "Thailand",
3      code: "TH",
4      population: 69950850
5  }
6
7  console.log(Object.keys(country))
8  console.log(Object.values(country))
9  console.log(Object.entries(country))
```

นี่เป็นผลลัพธ์การทำงานของโปรแกรม

```

1  [ 'name', 'code', 'population' ]
2  [ 'Thailand', 'TH', 69950850 ]
3  [
4      [ 'name', 'Thailand' ],
5      [ 'code', 'TH' ],
6      [ 'population', 69950850 ]
7  ]
```

คลาสและออบเจ็คใน TypeScript

การสร้างคลาส

คลาสสามารถสร้างได้อย่างง่ายดายด้วยการใช้งานคำสั่ง class นี่เป็นรูปแบบการสร้างคลาสในภาษา TypeScript

```
1 class NameOfClass {
2     // member declarations
3 }
```

ในการสร้างคลาสเราจะใช้คำสั่ง class ตามด้วยชื่อของคลาส NameOfClass โดยการตั้งชื่อของคลาสจะเหมือนกับตัวแปรและเมธอด แต่ตัวอักษรตัวแรกของคลาสมักจะเริ่มต้นด้วยตัวพิมพ์ใหญ่แทนเพื่อทำให้มันแยกแยะจากออบเจ็คอื่นๆ ได้ง่ายขึ้น

```
1 class Rectangle {
2     width: number
3     height: number
4
5     area(): number {
6         return this.width * this.height
7     }
8 }
```

การสร้างออบเจ็ค

เมื่อคลาสถูกประกาศแล้วมันสามารถนำไปสร้างออบเจ็คได้ และอย่างที่กล่าวไปในตอนต้น หนังคลาஸสามารถนำไปสร้างออบเจ็คเป็นจำนวนมากได้ มาดูการใช้งานในตัวอย่างนี้ ที่เป็นการสร้างคลาสเพื่อไปสร้างหลายออบเจ็คของบุคคล

```
1 class Person {
2     firstName: string
3     lastName: string
4     occupation: string
5
6     getName(): string {
7         return this.firstName + " " + this.lastName
8     }
9
10    getInfo(): string {
11        return `${this.getName()} is a ${this.occupation}.`
12    }
13 }
14
15 let p1: Person = new Person();
16 p1.firstName = "Samit"
17 p1.lastName = "Koyom"
18 p1.occupation = "Programmer"
```

គុណសារការងារ

ការប្រកាសគុណសារការងារ

គុណសារការងារ (Constructor) ឱ្យបានរៀនដឹងពីការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។ គុណសារការងារនេះមានសំណង់ចាប់ពីការប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។ គុណសារការងារនេះមានសំណង់ចាប់ពីការប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។

```

1  class Rectangle {
2      width: number
3      height: number
4
5      constructor(width: number, height: number) {
6          console.log("constructor has been called.")
7          this.width = width
8          this.height = height
9      }
10
11     area(): number {
12         return this.width * this.height
13     }
14 }
15
16 let rect1: Rectangle = new Rectangle(3, 4)

```

នៃព័ត៌មាននេះ យើងបានប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។ គុណសារការងារនេះមានសំណង់ចាប់ពីការប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។

```

1  constructor(width: number, height: number) {
2      console.log("constructor has been called.")
3      this.width = width
4      this.height = height
5  }

```

នៃព័ត៌មាននេះ យើងបានប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។ គុណសារការងារនេះមានសំណង់ចាប់ពីការប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។

```

1  let rect1: Rectangle = new Rectangle(3, 4);
2  let rect2: Rectangle = new Rectangle(6, 5);

```

នៃព័ត៌មាននេះ យើងបានប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។ គុណសារការងារនេះមានសំណង់ចាប់ពីការប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។

នៃព័ត៌មាននេះ យើងបានប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។ គុណសារការងារនេះមានសំណង់ចាប់ពីការប្រកាសគុណសារការងារដែលត្រូវបានធ្វើឡើងនៅក្នុងការប្រកាសគុណសារការងារ។

Interface ในภาษา TypeScript

การใช้งาน Interface

โดยปกติแล้วการกำหนดประเภทข้อมูลให้กับออบเจ็คในภาษา TypeScript จะสามารถใช้ object เป็นประเภทข้อมูลในตัวแบบประกาศตัวแปรได้ แต่ประเภทข้อมูลนี้จะทำให้ออบเจ็คสามารถมีรูปทรงเป็นแบบใดๆ ก็ได้ หรืออ่านว่าอีกนัยหนึ่ง มันสามารถมีจำนวนของ Property และประเภทข้อมูลเป็นแบบอะไรก็ได้

ในภาษา TypeScript นั้นเราสามารถใช้คำสั่ง interface สำหรับสร้างประเภทข้อมูลใหม่เพื่อกำหนดโครงสร้างของออบเจ็คให้มีความเฉพาะเจาะจงมากขึ้น ต่อไปเรามาสร้าง Interface สำหรับออบเจ็คที่ใช้เก็บข้อมูลของผู้ใช้งาน นี่เป็นตัวอย่าง

```
1 interface User {
2   id: number
3   name: string
4 }
```

ในตัวอย่างนี้ เป็นการสร้าง Interface ซึ่ว่า User ก็มีสอง Property คือ id มีประเภทข้อมูลเป็นตัวเลข และ name มีประเภทข้อมูลเป็น String เมื่อ Interface ถูกสร้างแล้ว เราสามารถนำมันไปใช้เป็นประเภทข้อมูลของออบเจ็คได้ และนี่เป็นตัวอย่างการใช้งาน

```
1 interface User {
2   id: number;
3   name: string;
4 }
```

Interface vs. Type

ทั้งคำสั่ง type และ Interface นั้นสามารถกำหนดประเภทข้อมูลให้กับออบเจ็คได้ นี่ทำให้เราสามารถควบคุม Property และประเภทข้อมูลที่ Property เหล่านั้นจะมี การทำงานของมันไม่แตกต่างกันสามารถใช้แบบไหนก็ได้

```
1 type Type1 = {
2   id: number,
3   name: string
4 };
5
6 interface Type2 {
7   id: number
8   name: string
9 };
10
11 let user1: Type1 = {
12   id: 1,
13   name: "John"
14 };
15
16 let user2: Type2 = {
17   id: 2,
18   name: "Julia"
```

Section 2

การทำงานกับ Vue 3 ร่วมกับ Vite 3 ขั้นสูง

- Creating a Project with Vite and TypeScript
- The project structure in deep
- Networking requests from Vite
- Production Builds
- Using Vite in Existing Vue Projects
- Migrating vue.config.js



Create Project with Vite

ติดตั้ง Vite แบบ Global

```
1 npm install -g vite
```

เช็คเวอร์ชันของ Vite

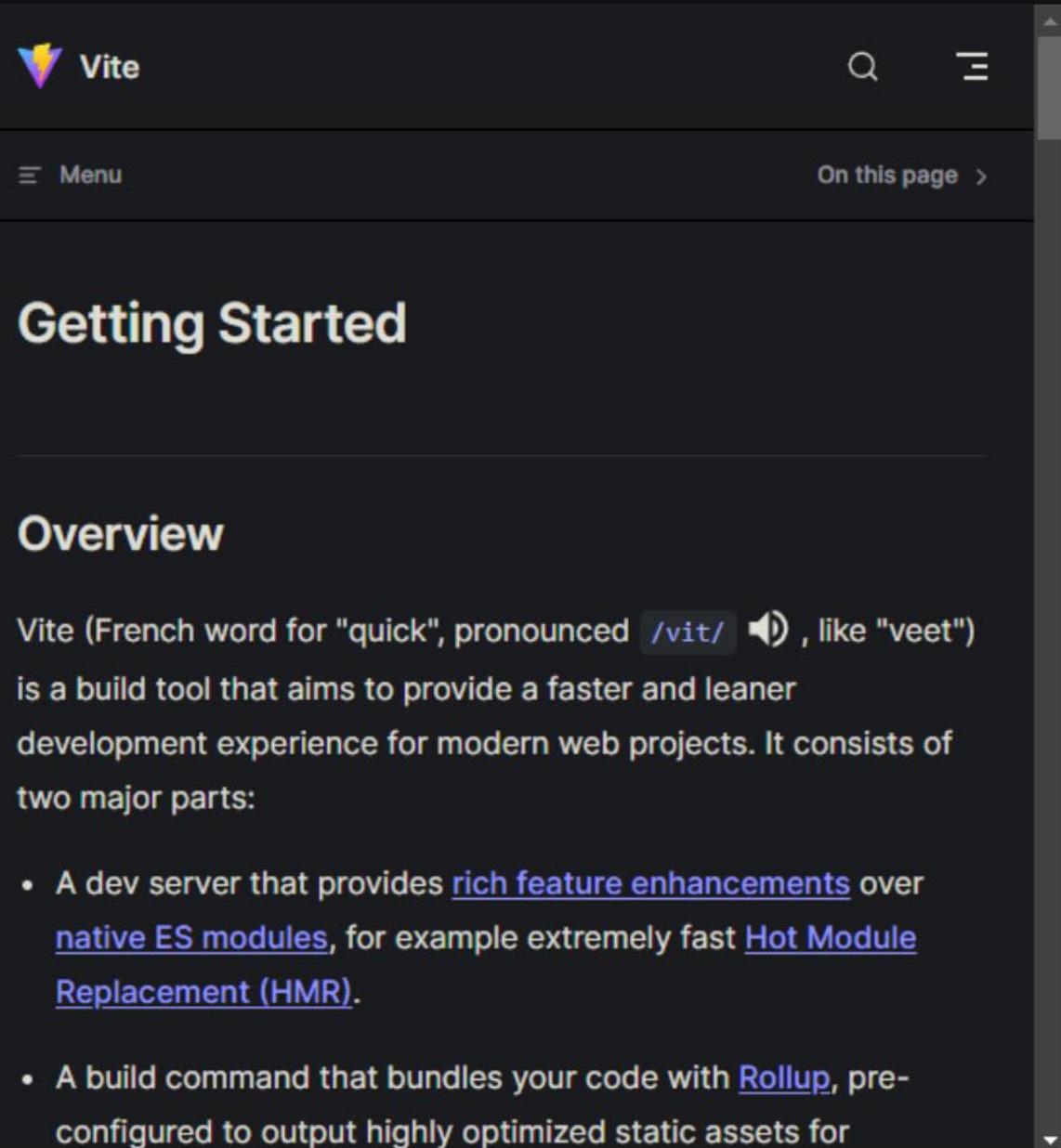
```
1 vite -v
```

สร้างโปรเจกต์ Vue.js ด้วย Vite

```
1 npm create vite@latest
```

Step 1: ตั้งชื่อโปรเจกต์เป็น "vue-vite-demo" และเลือก framework เป็น Vue

```
C:\Windows\system32\cmd.exe + 
G:\TrainingWorkshop\VueJS3AdvancedClass>npm create vite@latest
? Project name: ... vue-vite-demo
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
>  Vue
  React
  Preact
  Lit
  Svelte
  Others
```



The screenshot shows the official Vite documentation website. At the top, there's a navigation bar with the Vite logo, a search icon, and a menu icon. Below the header, a breadcrumb navigation "On this page >" is visible. The main content area features a large heading "Getting Started" followed by a horizontal line. Below this, there's a section titled "Overview". The text in "Overview" describes Vite as a French word for "quick", pronounced /vit/ (with a speaker icon), and states that it is a build tool aiming for a faster and leaner development experience for modern web projects. It highlights two major parts: a dev server providing rich feature enhancements over native ES modules, including extremely fast Hot Module Replacement (HMR), and a build command that bundles code with Rollup, pre-configured to output highly optimized static assets.

Getting Started

Overview

Vite (French word for "quick", pronounced /vit/ 🔊, like "veet") is a build tool that aims to provide a faster and leaner development experience for modern web projects. It consists of two major parts:

- A dev server that provides [rich feature enhancements](#) over [native ES modules](#), for example extremely fast [Hot Module Replacement \(HMR\)](#).
- A build command that bundles your code with [Rollup](#), pre-configured to output highly optimized static assets for

Create Project with Vite

Step 2: เลือก TypeScript

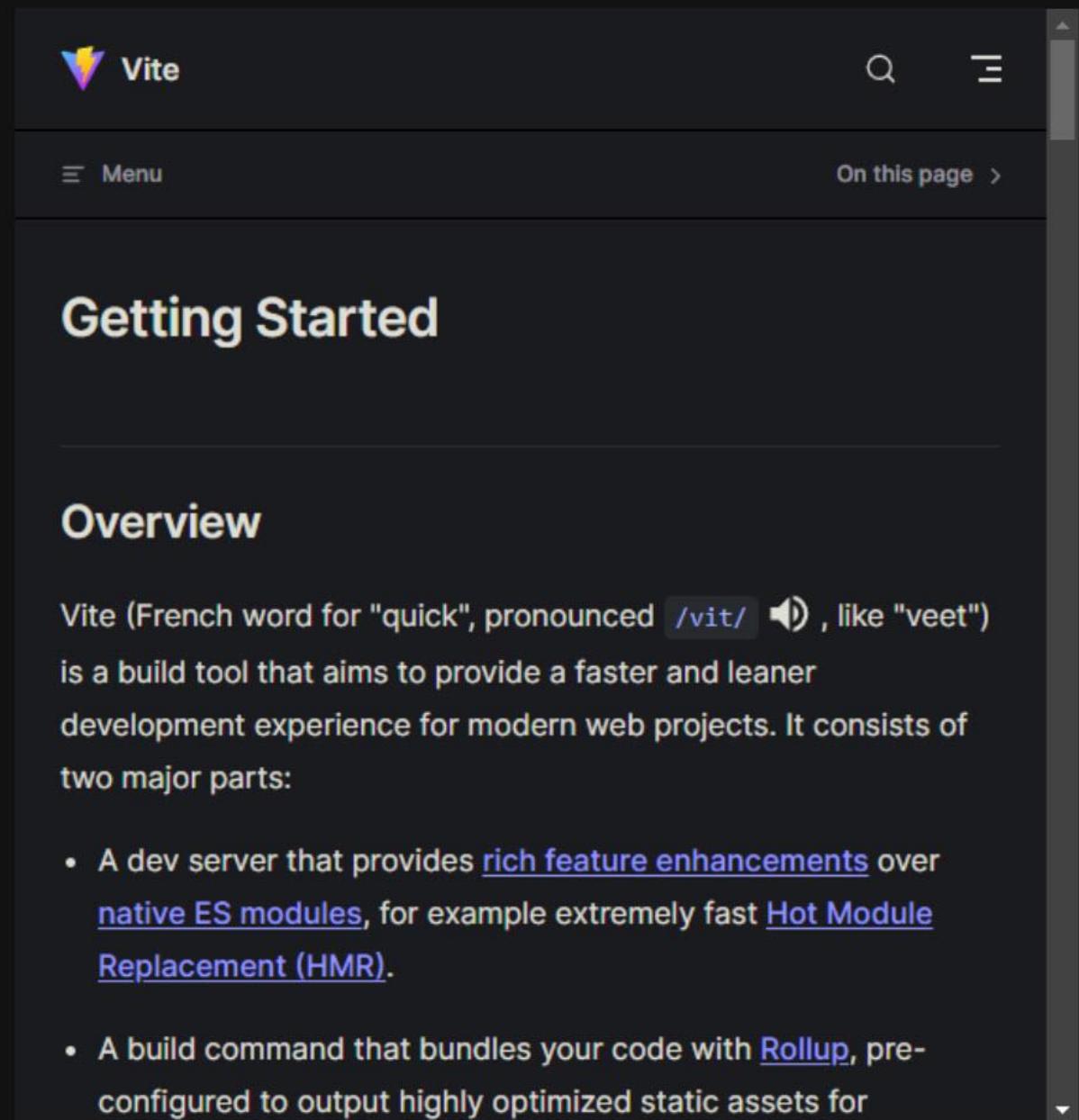
```
C:\Windows\system32\cmd.exe + 
G:\TrainingWorkshop\VueJS3AdvancedClass>npm create vite@latest
✓ Project name: ... vue-vite-demo
✓ Select a framework: » Vue
? Select a variant: » - Use arrow-keys. Return to submit.
    JavaScript
>  TypeScript
    Customize with create-vue ✘
    Nuxt ✘
```

Step 3: หลังจากสร้างโปรเจ็คต์เสร็จ

```
C:\Windows\system32\cmd.exe + 
G:\TrainingWorkshop\VueJS3AdvancedClass>npm create vite@latest
✓ Project name: ... vue-vite-demo
✓ Select a framework: » Vue
✓ Select a variant: » TypeScript

Scaffolding project in G:\TrainingWorkshop\VueJS3AdvancedClass\vue-vite-demo...
Done. Now run:

cd vue-vite-demo
npm install
npm run dev
```



The screenshot shows the official Vite documentation website. The header features the Vite logo and navigation links for 'Menu' and 'On this page'. The main content area has a dark background with white text. A large heading 'Getting Started' is centered at the top. Below it, a section titled 'Overview' contains text about Vite's purpose and structure. To the right of the text, there is a bulleted list of features.

Getting Started

Overview

Vite (French word for "quick", pronounced */vit/*  , like "veet") is a build tool that aims to provide a faster and leaner development experience for modern web projects. It consists of two major parts:

- A dev server that provides rich feature enhancements over native ES modules, for example extremely fast Hot Module Replacement (HMR).
- A build command that bundles your code with Rollup, pre-configured to output highly optimized static assets for

Create Project with Vite

Step 4: เปิดเข้า VS Code

```
C:\Windows\system32\cmd.exe X + 
G:\TrainingWorkshop\VueJS3AdvancedClass>code vue-vite-demo
G:\TrainingWorkshop\VueJS3AdvancedClass>
1 code vue-vite-demo
```

Step 5: ຕິດຕັ້ງ Dependency

ຕົວຢ່າງໂປຣເຈີກຕົ້ນ Vue 3 with Vite

Create Project with Vite

Step 6: Run Project

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
+ 1: node

6:\TrainingWorkshop\VueJS3AdvancedClass\vue-vite-demo>npm run dev

> vue-vite-demo@0.0.0 dev
> vite

VITE v3.2.3 ready in 319 ms

→ Local: http://127.0.0.1:5173/
→ Network: use --host to expose

1 npm run dev
```

Step 7: Build Project

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
+ 1: cmd

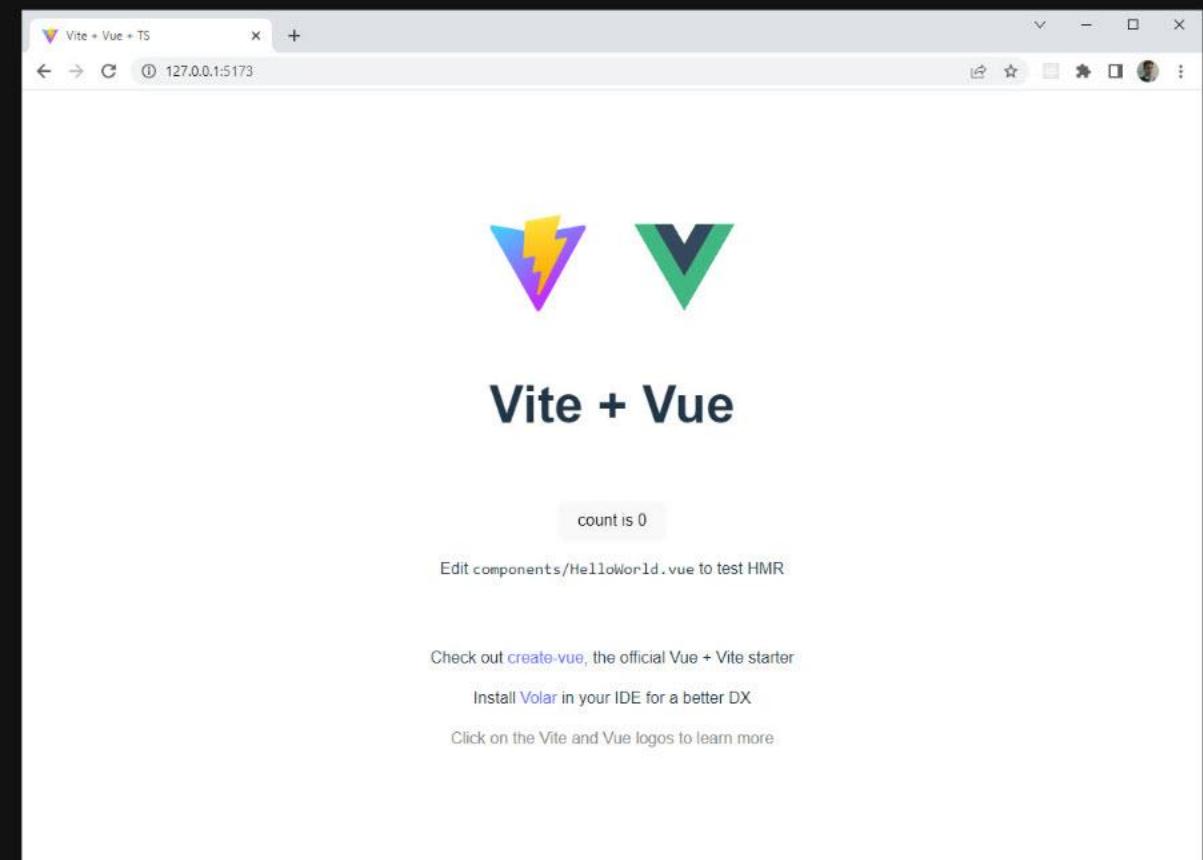
6:\TrainingWorkshop\VueJS3AdvancedClass\vue-vite-demo>npm run build

> vue-vite-demo@0.0.0 build
> vue-tsc && vite build

vite v3.2.3 building for production...
✓ 18 modules transformed.
dist/assets/vue.5532db34.svg 0.48 KiB
dist/index.html 0.44 KiB
dist/assets/index.35e49caf.css 1.26 KiB / gzip: 0.65 KiB
dist/assets/index.547b54de.js 54.71 KiB / gzip: 22.13 KiB

6:\TrainingWorkshop\VueJS3AdvancedClass\vue-vite-demo>
```

ตัวอย่างโปรเจกต์ที่รันสำเร็จ



Step 8: Run Project Production

```
1 serve -s dist
```

The project structure in deep

โครงสร้างโปรเจกต์ Vue.js 3 ด้วย Vite มีความคล้ายกับแบบ Vue CLI ดังภาพ

สิ่งแรกที่มีความแตกต่างจาก Vue CLI คือไม่มีไฟล์ **index.html** ในโฟลเดอร์ **public** แต่โครงสร้างแบบ Vite สร้างไฟล์ **index.html** วางไว้ภายนอก

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <link rel="icon" type="image/svg+xml" href="/vite.svg" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Vite + Vue + TS</title>
8  </head>
9  <body>
10     <div id="app"></div>
11     <script type="module" src="/src/main.ts"></script>
12 </body>
13 </html>
```

ตัวอย่างโครงสร้างโปรเจกต์ด้วย Vite

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under the folder "VUE-VITE-DEMO". It includes:
 - dist (empty)
 - node_modules (empty)
 - public (empty)
 - src (contains assets, components, style.css, vite-env.d.ts, .gitignore, index.html, package-lock.json, package.json, README.md, tsconfig.json, tsconfig.node.json, vite.config.ts)
- Code Editor:** The file "main.ts" is open in the editor tab. The code content is:


```

1 import { createApp } from 'vue'
2 import './style.css'
3 import App from './App.vue'
4
5 createApp(App).mount('#app')
6 
```
- Terminal:** Not visible in the screenshot.

The project structure in deep

โครงสร้างโปรเจกต์ Vue.js 3 ด้วย Vite มีความคล้ายกับแบบ Vue CLI ดังภาพ

สิ่งแรกที่มีความแตกต่างจาก Vue CLI คือไม่มีไฟล์ **index.html** ในโฟลเดอร์ **public** แต่โครงสร้างแบบ Vite สร้างไฟล์ **index.html** วางไว้ภายนอก

```

1  <!DOCTYPE html>
2
3  <html lang="en">
4
5      <head>
6          <meta charset="UTF-8" />
7          <link rel="icon" type="image/svg+xml" href="/vite.svg" />
8          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
9          <title>Vite + Vue + TS</title>
10
11     <body>
12         <div id="app"></div>
13         <script type="module" src="/src/main.ts"></script>
14
15     </body>
16
17 </html>
```

ตัวอย่างโครงสร้างโปรเจกต์ด้วย Vite

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "VUE-VITE-DEMO". It includes a ".vscode" folder, a "dist" folder containing "index.html" and "vite.svg", a "node_modules" folder, a "public" folder containing "vite.svg", and a "src" folder containing "assets" (with "vue.svg"), "components" (with "HelloWorld.vue" and "App.vue"), and "main.ts". Other files in "src" include "style.css", "vite-env.d.ts", ".gitignore", "index.html", "package-lock.json", "package.json", "README.md", "tsconfig.json", "tsconfig.node.json", and "vite.config.ts".
- Code Editor:** Displays the content of the "main.ts" file.
- Terminal:** Not visible in the screenshot.
- Status Bar:** Shows "main.ts - vue-vite-demo - Visual Studio Code".

```

1  import { createApp } from 'vue'
2  import './style.css'
3  import App from './App.vue'
4
5  createApp(App).mount('#app')
6
```

The project structure in deep

โครงสร้างโปรเจกต์ Vue.js 3 ด้วย Vite มีความคล้ายกับแบบ Vue CLI ดังภาพ

สิ่งแรกที่มีความแตกต่างจาก Vue CLI คือไม่มีไฟล์ **index.html** ในโฟลเดอร์ **public** แต่โครงสร้างแบบ Vite สร้างไฟล์ **index.html** วางไว้ภายนอก

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <link rel="icon" type="image/svg+xml" href="/vite.svg" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Vite + Vue + TS</title>
8  </head>
9  <body>
10     <div id="app"></div>
11     <script type="module" src="/src/main.ts"></script>
12 </body>
13 </html>

```

จุดที่น่าสนใจคือ Vite มีการเรียก JS Script แบบ “module” เข้ามาในโปรเจกต์ การเรียกลักษณะนี้ส่งผลดีอย่างไร ?

Networking requests from Vite

ตัวอย่างโครงสร้างโปรเจกต์ด้วย Vite

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "VUE-VITE-DEMO". It includes a ".vscode" folder, a "dist" folder containing "assets/index.html" and "vite.svg", a "public" folder containing "vite.svg", and a "src" folder containing "assets/vue.svg", "components/HelloWorld.vue", "components/App.vue", "main.ts", "style.css", "vite-env.d.ts", ".gitignore", "index.html", "package-lock.json", "package.json", "README.md", "tsconfig.json", "tsconfig.node.json", and "vite.config.ts".
- Code Editor:** The "main.ts" file is open, showing the following code:

```

1 import { createApp } from 'vue'
2 import './style.css'
3 import App from './App.vue'
4
5 createApp(App).mount('#app')
6

```

ภาพด้านล่างนี้คือตัวอย่างที่รันด้วย Vue CLI และ Webpack

The screenshot shows the Network tab in a browser developer tools window, displaying the following requests:

Name	Status	Type	Initiator	Size	Time	Waterfall
localhost	200	document	Other	900 B	5 ms	
chunk-vendors.js	200	script	(index)	827 kB	109 ms	
app.js	200	script	(index)	20.7 kB	13 ms	
ws	101	websocket	WebSocketClient.js?558...	0 B	Pending	
favicon.ico	200	vnd.microsoft	Other	1.2 kB	3 ms	

A red arrow points to the "app.js" row, highlighting it.

Using Vite in Existing Vue Projects

กรณีต้องการใช้ Vite กับโปรเจกต์ Vue CLI เก่าๆ ของเรา

Step 1: เริ่มต้นให้ทำการติดตั้ง Vite

```
1 "devDependencies": {
2   "@vue/compiler-sfc": "^3.0.5",
3   "vite": "^3.2.41"
4 }
```

Step 2: ถอนติดตั้ง plugin-vue

```
1 npm i vite @vitejs/plugin-vue --save-dev
```

Step 3: แก้ไข scripts ใน package.json

```
1 "scripts": {
2   "dev": "vite",
3   "build": "vite build",
4   "preview": "vite preview"
5 },
```

Step 4: ลบไฟล์ package.json.lock และทำการติดตั้งเข้ามาใหม่ด้วย

```
1 npm install
```

Step 5: สร้างไฟล์ vite.config.js และกำหนดค่าดังนี้

```
1 // vite.config.js
2 import { defineConfig } from 'vite'
3 import vue from '@vitejs/plugin-vue'
4
5 export default defineConfig({
6   plugins: [vue()]
7 })
```

Step 6: กำหนด script ไฟล์ /src/main.js หรือ /src/main.ts ดังนี้

```
1 <script type="module" src="/src/main.js"></script>
```

Step 7: ทดสอบรันโปรเจกต์ด้วยคำสั่ง npm run dev (แทน npm run serve)

```
1 npm run dev
```

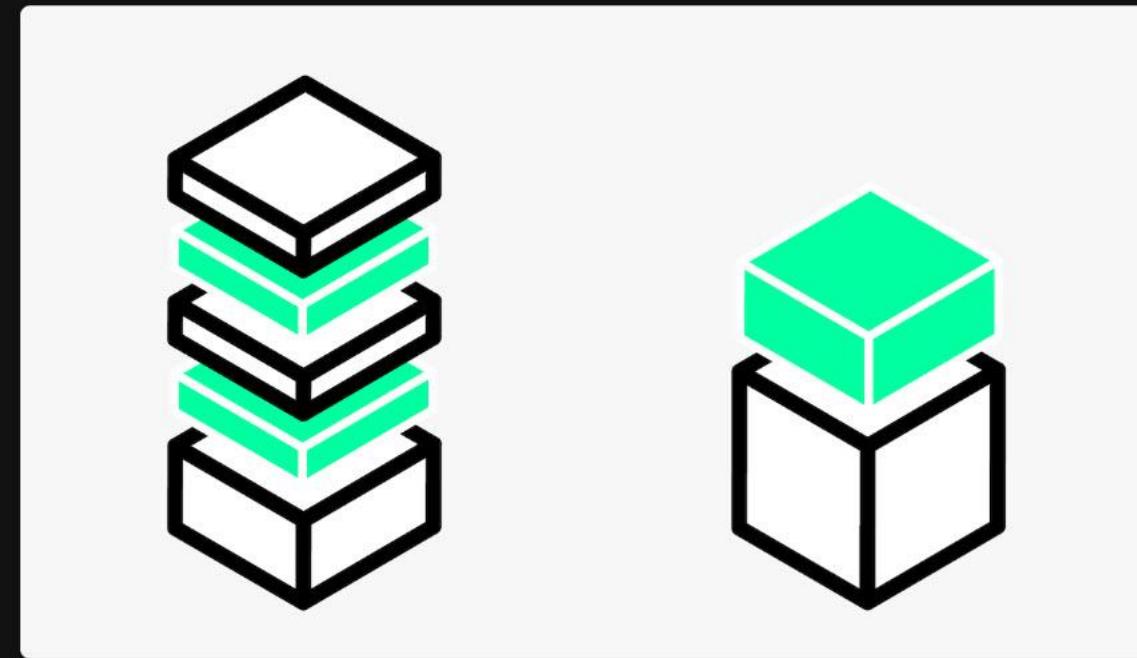
Section 3

การใช้งาน Vue Composition API ขั้นสูง

- Introduction to the Vue.js 3 Composition API
- Set up a Local Environment with Vite
- Vue Composition API Setup Method
- The Setup Method's Parameters: props and context
- Reactive Primitives with Refs
- Reactive Data with the Composition API
- Refs vs Reactive With the Vue 3 Composition API
- Computed Properties in the Composition API
- Using Watch with the Composition API
- The Composition API watchEffect Function
- Using provide / inject with the Composition API
- Lifecycle hooks and the Composition API
- Script Setup
- Creating a usePost Composable
- Using the usePost Composable in Multiple Components
- Using Vue Router with Composition API
- Refactoring for a More Versatile Composable
- Asynchronous Data and the Composition API
- The Suspense Component
- Use Shared State Between Uses of a Composable



Introduction to the Vue.js 3 Composition API



ทำไมต้องใช้ composition API ?

- ข้อจำกัดใน Vue 2 เมื่อ component มีขนาดใหญ่จะยากต่อการอ่าน
- โค้ดไม่ค่อยได้เกิดการ reuse
- ใน Vue 2 มีข้อจำกัดกับการทำงานด้วย TypeScript

ตัวอย่างเราต้องการทำช่องค้นหา



ถ้าเราเขียนโค้ดด้วย Vue 2

```
search.vue
```

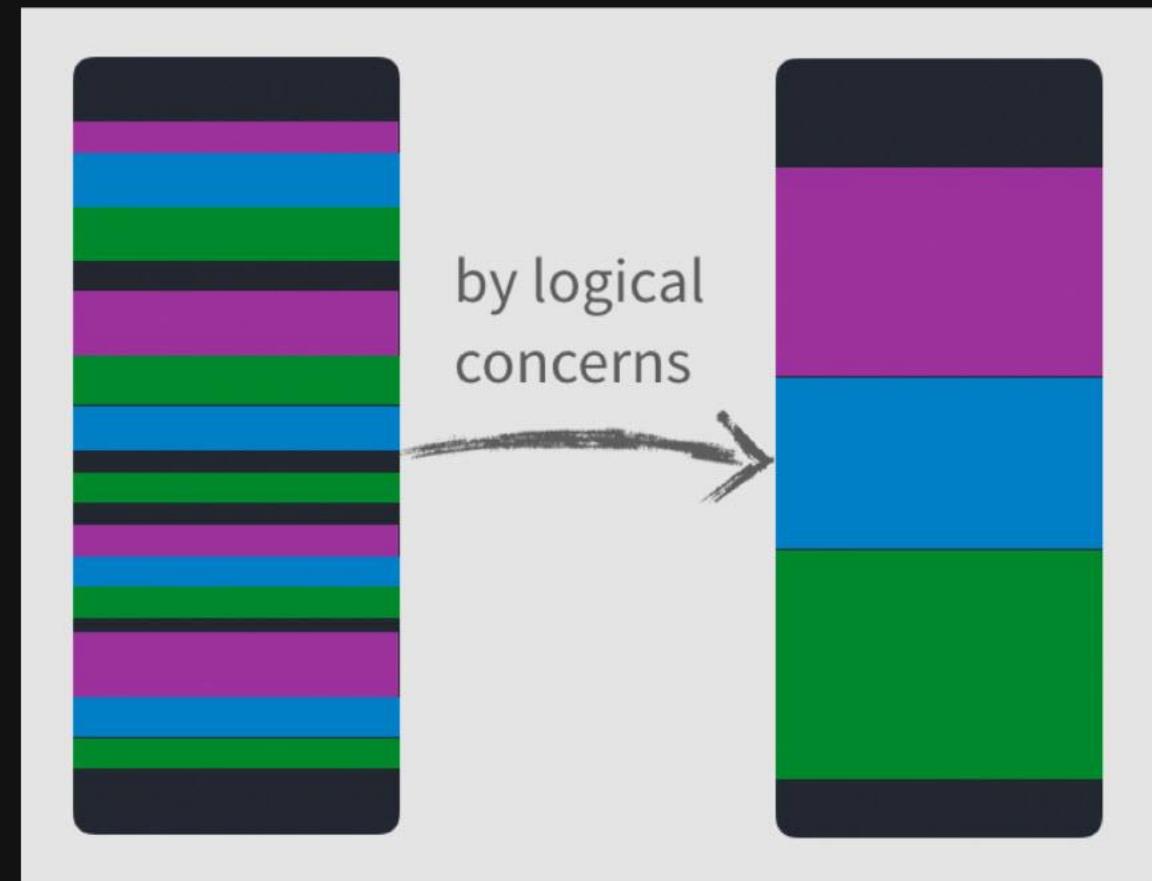
```
<script>
export default {
  data() {
    return {
      }
    },
  methods: {
    }
  }
</script>
```



จะเกิดอะไรขึ้นหากเราไม่การเพิ่มฟังก์ชันอื่น ๆ เข้าไป เช่น การ Sorting

และเมื่อเราเพิ่มฟังก์ชันอื่น ๆ วิถีเซ่น search filter, pagination คราวนี้ component ของเรามีดูง่ายขึ้น

search.vue

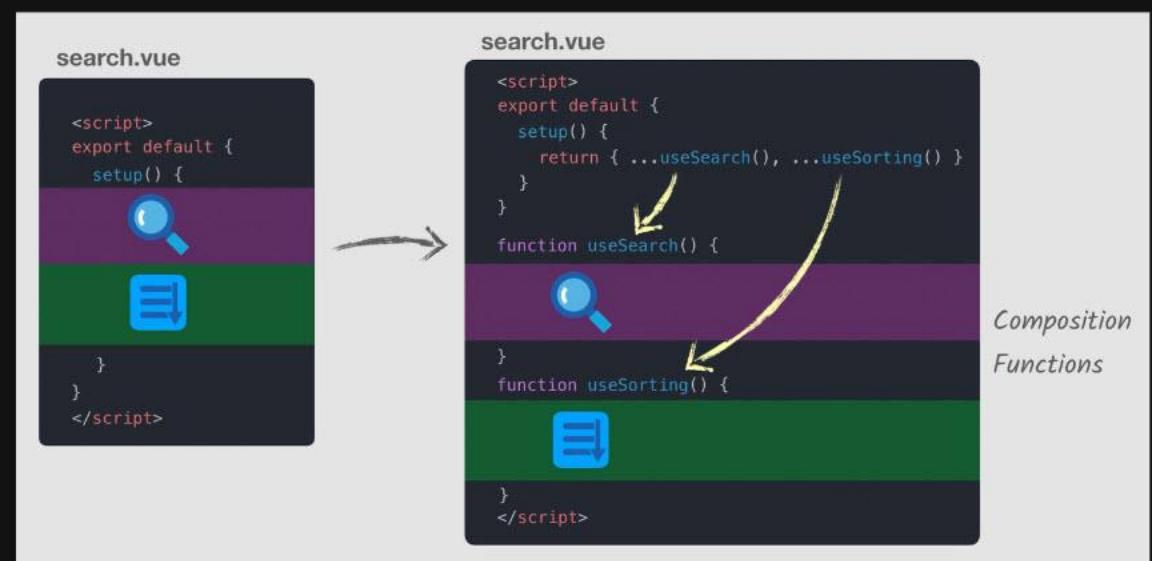


จะดีกว่าเมื่ย หากเราสามารถจัดกลุ่มโค้ด feature แบบเดียวกันไว้ด้วยกัน น่าจะช่วยให้อ่านง่ายขึ้น นี่จึงเป็นที่มาของ method **setup()** ที่เพิ่มเข้ามาใน Vue 3 Composition API

search.vue

```
<script>
export default {
  setup() {
    <img alt="Search icon" style="width: 20px; height: 20px; margin-right: 10px;">
    <img alt="List icon" style="width: 20px; height: 20px; margin-right: 10px;">
  }
}</script>
```

แต่หลายคนอาจจะคิดว่า แบบนี้เราจะต้องเขียน logic ขนาดใหญ่ยักษ์ไว้ใน method **setup()** ทั้งหมดใช่มั้ย ? คำตอบคือ "ไม่" เราสามารถแยกเป็นฟังก์ชันย่อยได้ที่เรารอเรียกว่า **Composition Functions**



ไม่มีแนวทางที่สมบูรณ์แบบในการ Reuse logic ระหว่าง Component

ใน Vue 2 เราจะนิยม Reuse logic ด้วยการใช้ Mixins

```

search.vue
<script>
export default {
  data() {
    return {
      ...
    }
  },
  methods: {
    ...
  }
}
</script>

```

```

<script>
const productSearchMixin = {
  data() {
    return {
      ...
    }
  },
  methods: {
    ...
  }
}
const resultSortMixin = {
  data() {
    return {
      ...
    }
  },
  methods: {
    ...
  }
}
export default {
  mixins: [productSearchMixin, resultSortingMixin]
}

```

ปัญหาคือว่า 만약 ความสัมพันธ์ของแต่ละตัวไม่ค่อยซัดเจน และ reuse ได้ไม่ดีพอ

ใน Vue 3 Composition API เราจะใช้หลักการแยกโค้ดส่วนที่ต้อง Reuse ออกจากภายนอก และใช้ composition function ในการเรียกเข้ามาใช้งาน

```

use/search.js
export default function useSearch(getResults) {
  ...
}

use/sorting.js
export default function useSorting({ input, options }) {
  ...
}

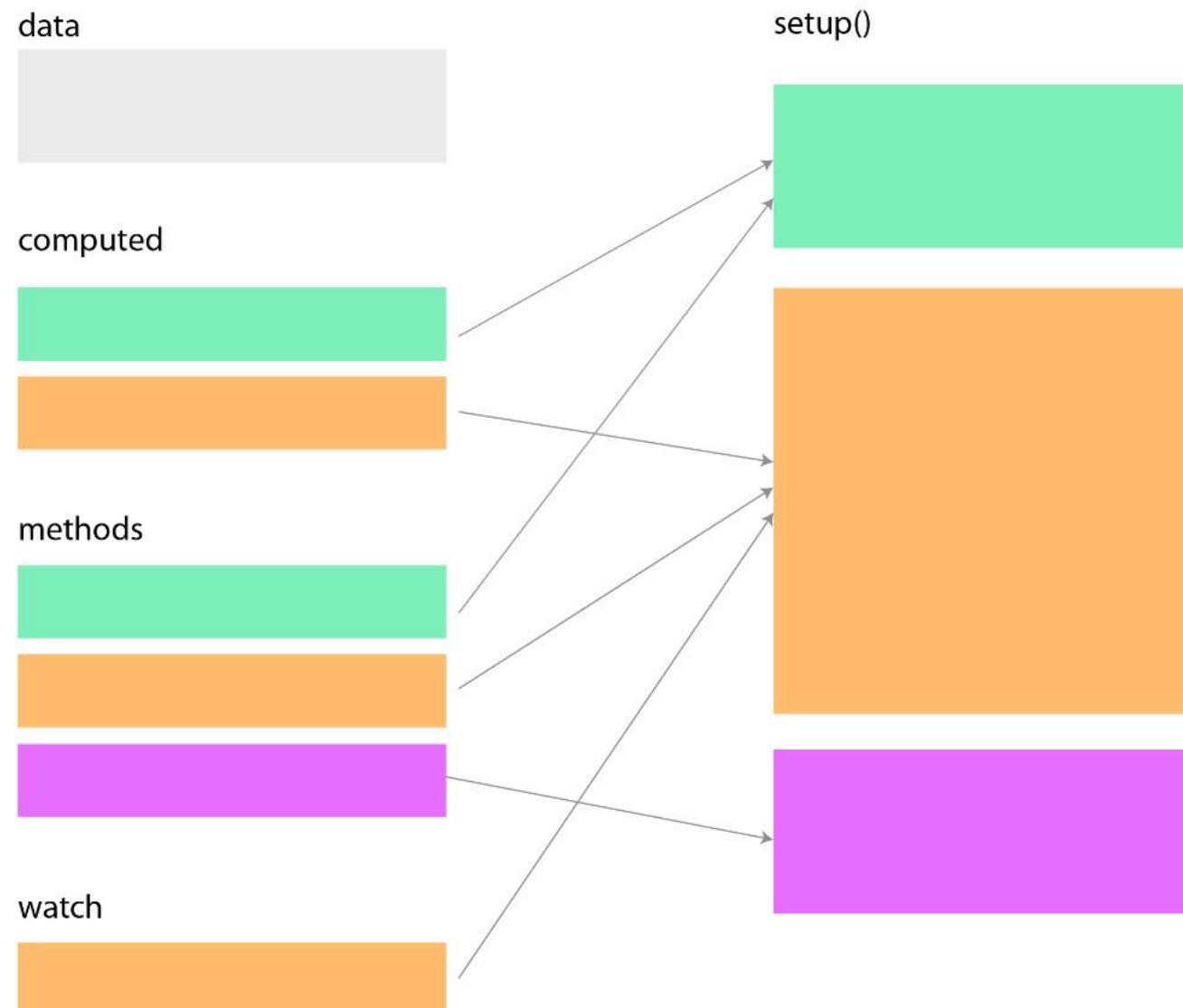
search.vue
import useSearch from '@use/search'
import useSorting from '@use/sorting'

export default {
  setup() {
    const productSearch = useSearch( )
    const resultSorting = useSorting({ })
    ...
  }
}

```

Options API

Composition API



Composition API Lifecycle Hooks

Options API (Vue 2)	Options API (Vue 3)	Composition API (Vue 3)
beforeCreate	beforeCreate	setup().
created	created	setup().
beforeMount	beforeMount	onBeforeMount
mounted	mounted	onMounted
beforeUpdate	beforeUpdate	onBeforeUpdate
updated	updated	onUpdated
beforeDestroy	beforeUnmount <small>VUE3</small>	onBeforeUnmount
destroyed	unmounted <small>VUE3</small>	onUnmounted
activated	activated	onActivated
deactivated	deactivated	onDeactivated



**REF
vs
REACTIVE**

REF

for primitive data type a reactive property, ref()

- String
- Number
- BigInt
- Boolean
- Symbol
- Null
- Undefined

```

1 <script>
2 import { ref } from "vue"
3 export default {
4   setup() {
5     const title = ref("Welcome")
6     const one = ref(1)
7     const isValid = ref(true)
8     const foo = ref(null)
9   }
10 }
11 </script>
```

REACTIVE

that reactive() will not work with primitive values.
reactive() takes an object and returns a reactive proxy of the original

```

1 <script>
2 import { reactive } from "vue"
3
4 export default {
5   setup() {
6     const data = reactive({
7       title: "Hello, Vue 3"
8     })
9
10    return { data }
11  }
12 }
13 </script>
```

Section 4

การประยุกต์ใช้งาน VueUse Library

- Intro to VueUse
- Browser - Reactive Browser Wrappers in VueUse - useTitle
- Browser - Reactive Browser Wrappers in Vueuse - useClipboard
- Browser - Theming with VueUse useDark and useColorMode
- Sensors - The Keyboard and VueUse
- Sensors - Device Sensors and VueUse
- Sensors - Mouse & Touch with VueUse
- Sensors - Scroll and VueUse
- Sensors - Media and VueUse
- Animation - useInterval and useIntervalFn in VueUse
- Animation - Create an Animated Sprite with useIntevalFn and useRafFn
- Animation - Timeouts and VueUse
- Animation - Animating Time with VueUse
- Animation - Transitions and VueUse
- State - Ref History with VueUse
- State - Persistent Storage with VueUse
- VueUse Utilities: useFetch and Reactify
- VueUse Utilities: refAutoReset and useBase64
- VueUse Utilities: computedAsync and logic utils
- Create an Image Carousel with VueUse
- How to Extend a VueUse Composable
- DOM Elements and the Window with VueUse
- Vue Components and VueUse





@vueuse/core

Core functions of VueUse



@vueuse/shared

Internal functions of VueUse



@vueuse/components

Renderless components



@vueuse/math

Reactive Math Utilities



@vueuse/head

Document head manager

by @egoist



@vueuse/sound

Composable for playing sound

by @Tahul



@vueuse/motion

Vue components in motion

by @Tahul



@vueuse/gesture

Composables for interactive

by @Tahul



@vueuse/schema-org

Schema.org graphs for Vue

by @harlan-zw



@vueuse/integrations

Integrations for popular

packages



@vueuse/firebase

Firebase integrations



@vueuse/rxjs

RxJS integrations

Day
3

วัน
624
ชั่วโมงเต็ม



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน



อาจารย์สามิตร
สถาบันไอทีจีเนียส

Vue.js 3 ขั้นสูง

🗓 11-12,18-19,25-26 November 2022
⌚ 20.00-23.30 PM

3

Section 5

จัดการ State management ด้วย Pinia

- Introduction to Pinia
- Install and Setup Up Pinia
- Define Your First Pinia Store
- Access State from a Pinia Store
- Synchronous and Asynchronous Actions in Pinia
- Create a Cart Store and the Pinia Devtools Timeline
- An Example of Local Vs Global State
- Getters in Pinia
- Dynamic Getters and Displaying the Cart Items
- Reset the State in Pinia
- Access Stores from Other Stores
- Access Pinia State in the Options API
- Access Pinia Getters in the Options API
- Access Pinia Actions in the Options API
- Preserve State with Hot Module Replacement
- Subscribing to Actions
- Subscribe to the State
- Pinia Plugins
- Use Composables in the Pinia State



Pinia Shopping Cart



Home x +

localhost:5173

ร้านไอทีเนย์ส จำนวนร้านค้า (4) ☀️

หน้าหลัก เดียวกับเรา

รายการสินค้า



Iphone 12
Iphone 12 Deatil Some quick example text to build on the card title and make up the bulk of the content
\$700 เพิ่มลงรถเข็น



Samsung S10
Samsung S10 Deatil Some quick example text to build on the card title and make up the bulk of the content
\$400 เพิ่มลงรถเข็น



Samsung TV
Samsung TV Deatil Some quick example text to build on the card title and make up the bulk of the content
\$1200 เพิ่มลงรถเข็น



Huawei Mate Pro
Huawei Mate Pro Deatil Some quick example text to build on the card title and make up the bulk of the content



Google Home
ลำโพงวิจัยเร้แมพลงตามคำสั่งเสียงควบคุมโดยอิเล็กทรอนิกส์



ลำโพง JBL FLIP6 สีดำ
ลำโพงพกพา จากแบรนด์ JBL สุดยอดคุณภาพเสียงเด็ด พลังยิ่งกว่าเดิม ลุยไปทั่วโลกทุกที่

IT Genius Engineering | สถาบันไอทีเนย์ส

Cart x +

localhost:5173/cart

ร้านไอเก็จเนยส์ ชั้น ๔ ณ กรุงรัตนค่า (4) ☀️

หน้าหลัก เที่ยวทั่วเรา

ตะกร้าสินค้า

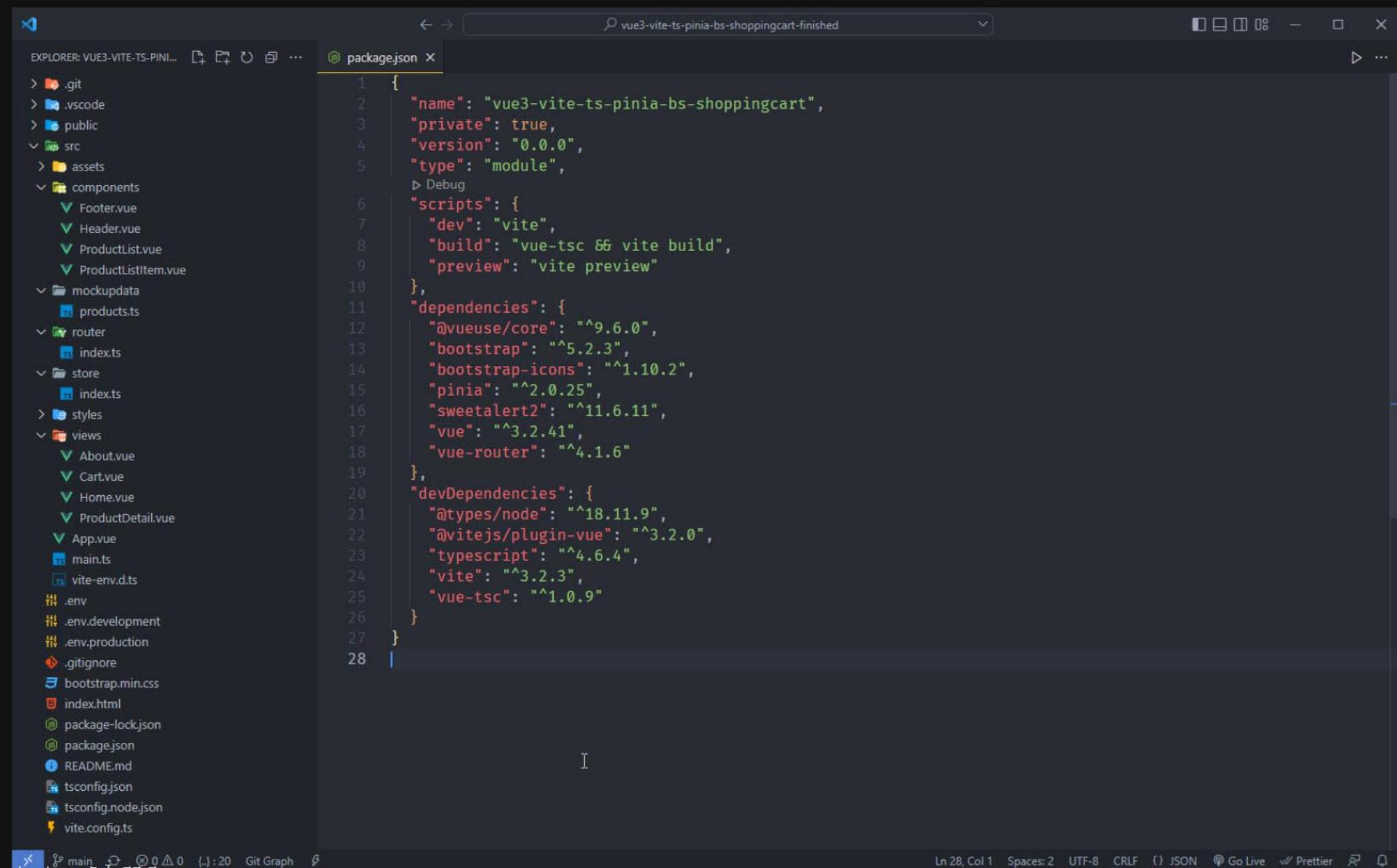
#	ภาพสินค้า	ชื่อ	จำนวน	ราคาร่อซื้อ	ราคารวม
2		Samsung S10	△ 1 ▽	\$400	\$400 ☒
3		Samsung TV	△ 2 ▽	\$1200	\$2400 ☒
6		ลำโพง JBL FLIP6 สีดำ	△ 3 ▽	\$159	\$477 ☒
5		Google Home	△ 1 ▽	\$289	\$289 ☒
				ทั้งหมด	\$3566

◀ กลับหน้าหลัก ▶ ไปชำระเงิน

Section[Home](#)[Features](#)[Pricing](#)[FAQs](#)**Section**[Home](#)[Features](#)[Pricing](#)[FAQs](#)**Section**[Home](#)[Features](#)[Pricing](#)[FAQs](#)**Subscribe to our newsletter**

Monthly digest of what's new and exciting from us.

Subscribe



```

EXPLORER: VUE3-VITE-TS-PINIA... ⌂ ⌃ ⌁ ⌂ ... package.json x
1  {
2    "name": "vue3-vite-ts-pinia-bs-shoppingcart",
3    "private": true,
4    "version": "0.0.0",
5    "type": "module",
6    "scripts": {
7      "dev": "vite",
8      "build": "vue-tsc && vite build",
9      "preview": "vite preview"
10   },
11   "dependencies": {
12     "@vueuse/core": "^9.6.0",
13     "bootstrap": "^5.2.3",
14     "bootstrap-icons": "^1.10.2",
15     "pinia": "^2.0.25",
16     "sweetalert2": "^11.6.11",
17     "vue": "^3.2.41",
18     "vue-router": "^4.1.6"
19   },
20   "devDependencies": {
21     "@types/node": "^18.11.9",
22     "@vitejs/plugin-vue": "^3.2.0",
23     "typescript": "^4.6.4",
24     "vite": "^3.2.3",
25     "vue-tsc": "^1.0.9"
26   }
27 }
28

```

The screenshot shows the VS Code interface with the title bar "vue3-vite-ts-pinia-bs-shoppingcart-finished". The left sidebar displays the project structure under "EXPLORER: VUE3-VITE-TS-PINIA...". The main editor area shows the content of the "package.json" file. The code is color-coded: blue for objects, green for strings, red for numbers, and purple for booleans. Brackets and punctuation are also highlighted. The status bar at the bottom shows "Ln 28, Col 1" and other standard status indicators.

ขั้นตอนการติดตั้ง

```

1  ++++++
2  | การสร้างโปรเจกต์ใหม่
3  | Vue 3+ TypeScript + Vite 3 + Pinia + Bootstrap 5
4  ++++++
5  Step 1: Install Vite in Global
6  ---
7  npm install -g vite
8
9  Step 2: Create new Vite Project with Vue
10 ---
11 npm create vite@latest
12 or
13 yarn create vite
14
15 กำหนดชื่อโปรเจกต์เป็น "vue3-vite-ts-pinia-bs-shoppingcart"
16
17 Step 3: เปิดโปรเจกต์เข้า VS Code
18 ---
19 code vue3-vite-ts-pinia-bs-shoppingcart -r
20
21 Step 4: Install node module in Vite Project
22 ---
23 npm install
24

```

โครงสร้างโปรเจกต์

```

1  vue3-vite-ts-pinia-bs-shoppingcart
2  |
3  +---.vscode
4  |     extensions.json
5  +---public
6  |     vite.svg
7  \---src
8   |     App.vue
9   |     main.ts
10  |     vite-env.d.ts
11  +---assets
12  |     vue.svg
13  +---components
14  |     Footer.vue
15  |     Header.vue
16  |     ProductList.vue
17  |     ProductListItem.vue
18  +---mockupdata
19  |     products.ts
20  +---router
21  |     index.ts
22  +---store
23  |     index.ts
24  +---styles

```

vite.config.ts

```

1 import { defineConfig } from 'vite'
2 import vue from '@vitejs/plugin-vue'
3 import { fileURLToPath, URL } from 'url'
4
5 // https://vitejs.dev/config/
6 export default defineConfig({
7   resolve: {
8     alias: {
9       '@': fileURLToPath(new URL('./src', import.meta.url))
10    }
11  },
12  plugins: [vue()]
13})

```

.env

```
1 VITE_APP_TITLE=Vue Shopping Cart
```

.env.development

```

1 VITE_APP_URL=/
2 VITE_APP_URL_API=https://laravel-backend-class.herokuapp.com/api/

```

.env.production

tsconfig.json

```

1  {
2    "compilerOptions": {
3      "paths": {
4        "@/*": [
5          "./src/*" // set path `@/*` as alias of `src/*`
6        ]
7      },
8      "target": "ESNext",
9      "useDefineForClassFields": true,
10     "module": "ESNext",
11     "moduleResolution": "Node",
12     "strict": true,
13     "jsx": "preserve",
14     "resolveJsonModule": true,
15     "isolatedModules": true,
16     "esModuleInterop": true,
17     "lib": ["ESNext", "DOM"],
18     "skipLibCheck": true,
19     "noEmit": true
20   },
21   "include": ["src/**/*.ts", "src/**/*.d.ts", "src/**/*.tsx", "src/**/*.vue", "src/**/*.{js,jsx,css,less,scss}"],
22   "references": [{ "path": "./tsconfig.node.json" }]
23 }

```

Pinia Shopping Cart



Vue Shopping Cart

localhost:5173

ຮ້ານອອນໄລນ໌ ມັນເກົ່າ ເຊິ່ງວັດທະນາ

ບັນທຶກຕົ້ນທາງ...



Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops
\$109.95

ເພີ້ມອຳນວຍ



Mens Casual Premium Slim Fit T-Shirts
\$22.30

ເພີ້ມອຳນວຍ



Mens Cotton Jacket
\$55.99

ເພີ້ມອຳນວຍ



Mens Casual Slim Fit
\$15.99

ເພີ້ມອຳນວຍ



John Hardy Women's Legends Naga Gold & Silver Dragon Station Chain Bracelet
\$695.00

ເພີ້ມອຳນວຍ



Solid Gold Petite Micropave
\$168.00

ເພີ້ມອຳນວຍ



White Gold Plated Princess
\$9.99

ເພີ້ມອຳນວຍ



Pierced Owl Rose Gold Plated Stainless Steel Double
\$10.99

ເພີ້ມອຳນວຍ

IT Genius Engineering | ສາມານີໂຄງເນັດ

Vue Shopping Cart X +

localhost:5173/product/1

🔍 🌐 ⭐

ร้านออนไลน์ หน้าหลัก เพื่อนๆ กับเรา

บันทึกการซื้อ...



Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops

Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday.

\$109.95

 เพิ่มลงตะกร้า

SERVICES

- Branding
- Design
- Marketing

COMPANY

- About us
- Contact
- Jobs

LEGAL

- Terms of use
- Privacy policy
- Cookie policy

IT Genius Engineering | สถาบันไอทีจีเอช

Vue Shopping Cart x +

localhost:5173/cart

ร้านออนไลน์ หน้าหลัก เกี่ยวกับเรา

🛒 (6) บัญชีค้าค้นหา...

 Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops
\$199.00

- 1 + Delete

 Mens Cotton Jacket
\$199.00

- 2 + Delete

 Mens Casual Slim Fit
\$199.00

- 3 + Delete

ยอดรวม: \$269.90

SERVICES COMPANY LEGAL

Vue Shopping Cart x +

localhost:5173

ร้านออนไลน์ หน้าหลัก เกี่ยวกับเรา

🛒 6 ac

Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops
 Mens Cotton Jacket
 John Hardy Women's Legends Naga Gold & Silver Dragon Station Chain Bracelet
 Silicon Power 256GB SSD 3D NAND A55 SLC Cache Performance Boost SATA
 Mens Casual III 2.5

			
Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops \$109.95	Mens Casual Premium Slim Fit T-Shirts \$22.30	Mens Cotton Jacket \$55.99	Mens Casual III 2.5
<button>เพิ่มลงรถเข็น</button>	<button>เพิ่มลงรถเข็น</button>	<button>เพิ่มลงรถเข็น</button>	<button>เพิ่มลงรถเข็น</button>
			
IT Genius Engineering สถาบันอุปกรณ์	Solid Gold Petite Micropave	White Gold Plated Princess	Pierced Owl Rose Gold Plated

```

{
  "name": "vue3-vite-ts-tailwind-daisyui-cart",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vue-tsc && vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "daisyui": "^2.41.0",
    "pinia": "^2.0.26",
    "vue": "^3.2.41",
    "vue-router": "^4.1.6"
  },
  "devDependencies": {
    "@types/node": "^18.11.9",
    "@vitejs/plugin-vue": "^3.2.0",
    "autoprefixer": "^10.4.13",
    "postcss": "^8.4.19",
    "tailwindcss": "^3.2.4",
    "typescript": "^4.6.4",
    "vite": "^3.2.3",
    "vue-tsc": "^1.0.9"
  }
}

```

ขั้นตอนการติดตั้ง

```

1  ++++++
2  | การสร้างโปรเจกต์ใหม่
3  | Vue + TS + Vite + Pinia + Tailwind + daisyUI
4  ++++++
5  Step 1: Install Vite in Global
6  ---
7  npm install -g vite
8
9  Step 2: Create new Vite Project with Vue
10 ---
11 npm create vite@latest
12 or
13 yarn create vite
14
15 กำหนดชื่อโปรเจกต์เป็น "vue3-vite-ts-tailwind-daisyui-cart"
16
17 Step 3: เปิดโปรเจกต์เข้า VS Code
18 ---
19 code vue3-vite-ts-tailwind-daisyui-cart
20
21 Step 4: Install node module in Vite Project
22 ---
23 npm install
24

```

โครงสร้างโปรเจกต์

```

1  vue3-vite-ts-tailwind-daisyui-cart
2  |
3  +---.vscode
4  |     extensions.json
5  +---public
6  |     vite.svg
7  \---src
8  |     App.vue
9  |     main.ts
10 |     vite-env.d.ts
11 +---assets
12 |     vue.svg
13 +---components
14 |     CartCard.vue
15 |     CartCardSkeleton.vue
16 |     Footer.vue
17 |     Nav.vue
18 |     ProductCard.vue
19 |     ProductCardSkeleton.vue
20 |     Search.vue
21 +---composables
22 |     usePersistCart.ts
23 +---mockupdata
24 |     product.json

```

vite.config.ts

```

1 import { defineConfig } from 'vite'
2 import vue from '@vitejs/plugin-vue'
3 import { fileURLToPath, URL } from 'url'
4
5 // https://vitejs.dev/config/
6 export default defineConfig({
7   resolve: {
8     alias: {
9       '@': fileURLToPath(new URL('./src', import.meta.url))
10    }
11  },
12  plugins: [vue()]
13})

```

.env

```
1 VITE_APP_TITLE=Vue Shopping
```

.env.development

```

1 VITE_APP_URL=/
2 VITE_APP_URL_API=https://laravel-backend-class.herokuapp.com/api/

```

.env.production

tsconfig.json

```

1  {
2    "compilerOptions": {
3      "paths": {
4        "@/*": [
5          "./src/*" // set path `@/*` as alias of `src/*`
6        ]
7      },
8      "target": "ESNext",
9      "useDefineForClassFields": true,
10     "module": "ESNext",
11     "moduleResolution": "Node",
12     "strict": true,
13     "jsx": "preserve",
14     "resolveJsonModule": true,
15     "isolatedModules": true,
16     "esModuleInterop": true,
17     "lib": ["ESNext", "DOM"],
18     "skipLibCheck": true,
19     "noEmit": true
20   },
21   "include": ["src/**/*.ts", "src/**/*.d.ts", "src/**/*.tsx", "src/**/*.vue"],
22   "references": [{ "path": "./tsconfig.node.json" }]
23 }

```

Section 6

การใช้งาน Vuetify

- Introduction Vuetify 3 and Vue 3
- Creating a Vuetify Project
- Using Javascript HTML - CSS
- Layout
- Vuetify Components
- Grid System
- CSS Classes
- Adding Functionality



Day
4



อาจารย์สามิตร
สถาบันไอทีจีเนียส

วัน
6 24
ชั่วโมงเต็ม

🗓 11-12,18-19,25-26 November 2022

⌚ 20.00-23.30 PM



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน

Vue.js 3 ขั้นสูง

3

Section 7

การใช้งาน Vue ร่วมกับ Firebase API

- Firebase 9 Introduction & Setup
- Introduction to Firebase
- Create a Firebase Project
- Create App & Install Firebase
- Setup Firestore Database
- Connect to Database
- Display Notes from Firestore
- Get Notes in Real Time
- Add Note
- Delete Note
- Update Note
- Order Notes by Date (ID)
- Firebase 9 Authentication
- Login & Register Page - Tabs
- Login & Register Page - Form
- Firestore Authentication & Auth Store
- Register User
- Logout User
- Login User
- Listen for Auth Changes & Store User Data
- Redirect User on Auth Change



Section 8

การออกแบบ UI ด้วย Ant Design

- Getting started
- Create a new Vue 3 app
- Install ant-design-vue
- Ant Design components
- Form handling
- Ant Design icons



Vue.js 3 ขั้นสูง

Day 5



อาจารย์สามิตร
สถาบันไอทีจีเนียส

วัน 6 24 ชั่วโมงเต็ม

🗓 11-12,18-19,25-26 November 2022
⌚ 20.00-23.30 PM



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน

3

Section 9

การทำงานกับ Styled Components

- Installing styled-components in Vue.JS
- Passing the props in styled-components
- Dynamic styling with props in Vue.JS
- Applying the same style to multiple components
- Dynamically changing the type of rendered component
- Extending styles



Section 10

การใช้งาน CKeditor 5 ร่วมกับ Vue.JS 3

- Install the CKEditor 5 for Vue.JS
- Using ES6 modules
- Using the component locally
- Integrating a build from the online builder
- Using CKEditor from source
- Using the Document editor build
- Localization
- Component directives
- Component events



Day
6

วัน
6 24
ชั่วโมงเต็ม



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน



อาจารย์สามิตร
สถาบันไอทีจีเนียส

Vue.js 3 ขั้นสูง

🗓 11-12,18-19,25-26 November 2022
⌚ 20.00-23.30 PM

3

Section 11

การ deployed project ขึ้นใช้งานจริงบน Production

- Vue 3 build and config project for production
- Deploying A Vue.JS App With Netlify
- Deploying A Vue.JS Site with Vercel
- Deploying A Vue.JS App To Firebase Hosting



ขอบคุณมากครับ
ติดตามผลงานในช่องทาง Youtube ที่

[youtube.com/itgeniuseng](https://www.youtube.com/itgeniuseng)

เก็บตก



6 วัน
24 ชั่วโมงเต็ม



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน



อาจารย์สาเมตร
สถาบันไอโอเก็จเนียส

🗓 11-12,18-19,25-26 November 2022
⌚ 20.00-23.30 PM

Vue.js 3 ขั้นสูง

3

Vue with Vuetify and Firebase



สินค้า

John Doe
john@email.co

ไทย

สินค้า

+ เพิ่มสินค้า

ชื่อสินค้า ราคา จัดการ

24	เสื้อยืด	72.00 บาท	
23	กระเบ้า	504.00 บาท	
22	รองเท้า	108.00 บาท	
21	น้ำดื่ม	72.00 บาท	
20	หนังสือ	180.00 บาท	
19	ase	900.00 บาท	
18	tyri	1,404.00 บาท	
17	pooooo	504.00 บาท	
16	กระดาษ	900.00 บาท	
15	eeee	3,348.00 บาท	
14	hoho	396.00 บาท	
13	demo 2	756.00 บาท	
12	demo	792.00 บาท	

สินค้า

1 ไทย

เพิ่มสินค้า

#	ชื่อสินค้า	ราคา	จัดการ	
24	เสื้อยืด	72.00 บาท		
23	กระเบ้า	504.00 บาท		
22	รองเท้า	108.00 บาท		
21	น้ำดื่ม	72.00 บาท		
20	หนังสือ	180.00 บาท		
19	ase	900.00 บาท		
18	tyri	1,404.00 บาท		
17	poooooo	504.00 บาท		
16	กระดาษ	900.00 บาท		
15	eeee	3,348.00 บาท		
14	hoho	396.00 บาท		
13	demo 2	756.00 บาท		
12	demo	792.00 บาท		

เพิ่มสินค้า

ชื่อสินค้า
demo

ราคา (USD)
10

ปิด บันทึก

สินค้า

John Doe
john@email.co

แดชบอร์ด

สินค้า

การสั่งซื้อ

รายงาน

ตั้งค่าระบบ

ออกจากระบบ

1 ไทย เพิ่มสินค้า

สินค้า

#	ชื่อสินค้า	ราคา	จัดการ	
24	เสื้อยืด	72.00 บาท		
23	กระเพา	504.00 บาท		
22	รองเท้า	108.00 บาท		
21	น้ำดื่ม	72.00 บาท		
20	หนังสือ	180.00 บาท		
19	ase	900.00 บาท		
18	tyri	1,404.00 บาท		
17	oooooo	504.00 บาท		
16	กระดาษ	900.00 บาท		
15	eeee	3,348.00 บาท		
14	hoho	396.00 บาท		
13	demo 2	756.00 บาท		
12	demo	792.00 บาท		

แก้ไขข้อมูลสินค้า

ชื่อสินค้า
เสื้อยืด

ราคา (USD)
2

ปิด อัพเดต

สินค้า

John Doe
john@email.co

ไทย

เพิ่มสินค้า

แดชบอร์ด

สินค้า

การสั่งซื้อ

รายงาน

ตั้งค่าระบบ

ออกจากระบบ

#	ชื่อสินค้า	ราคา	จัดการ
24	เสื้อยืด	72.00 บาท	 
23	กระเบื้อง	504.00 บาท	 
22	รองเท้า	108.00 บาท	 
21	ปั๊ดเมม	72.00 บาท	 
20	หนังสือ	180.00 บาท	 
19	ase	900.00 บาท	 
18	tyri	1,404.00 บาท	 
17	poooooo	504.00 บาท	 
16	กระดาษ	900.00 บาท	 
15	eeee	3,348.00 บาท	 
14	hoho	396.00 บาท	 
13	demo 2	756.00 บาท	 
12	demo	792.00 บาท	 

ยืนยันการลบรายการนี้ ?

รายการนี้จะถูกลบจากวาระเมื่อคุณยืนยัน

ไป  ใช่ ลบเลย

เก็บตก



6 วัน
24 ชั่วโมงเต็ม



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน



อาจารย์สาเมตร
สถาบันไอโอเก็จเนียส

Vue.js 3 ขั้นสูง

🗓 11-12,18-19,25-26 November 2022

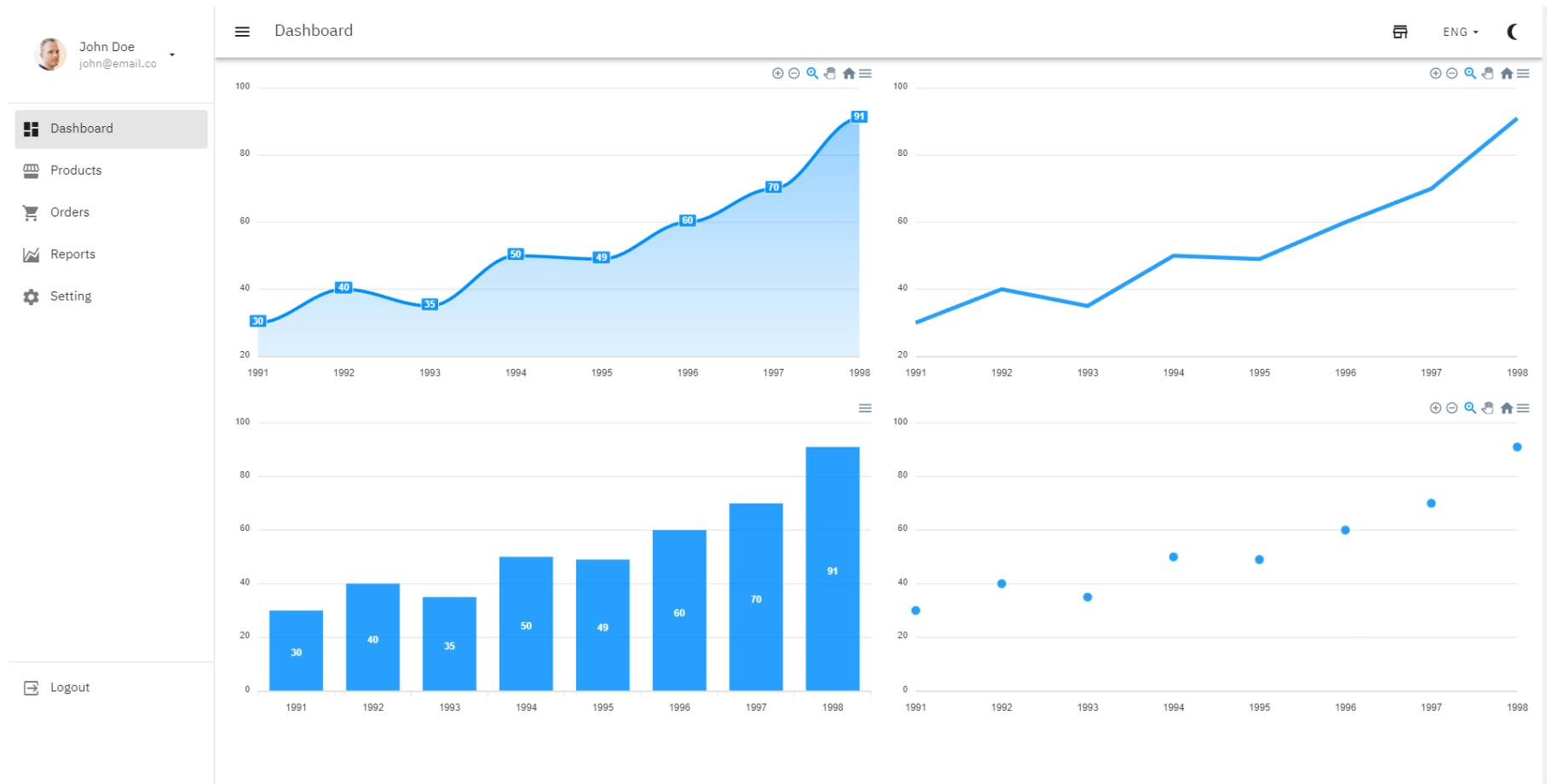
⌚ 20.00-23.30 PM

3

Vue with Apexcharts and Vue datepicker



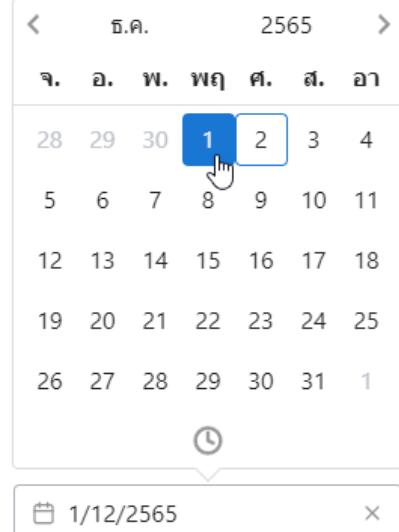
Apexcharts



```
npm install apexcharts  
npm install vue3-apexcharts
```

```
import VueApexCharts from "vue3-apexcharts"  
app.use(VueApexCharts)
```

Vue datepicker



ค่าที่จะนำไปใช้: 2022-12-01 01:28:00



ค่าที่จะนำไปใช้: 2022-12-01 01:28:00

```
npm install @vuepic/vue-datepicker@3.5.3
```

```
import Datepicker from '@vuepic/vue-datepicker';
import '@vuepic/vue-datepicker/dist/main.css'

app.component('Datepicker', Datepicker)
```

เก็บตก

3

6 วัน
24 ชั่วโมงเต็ม



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน



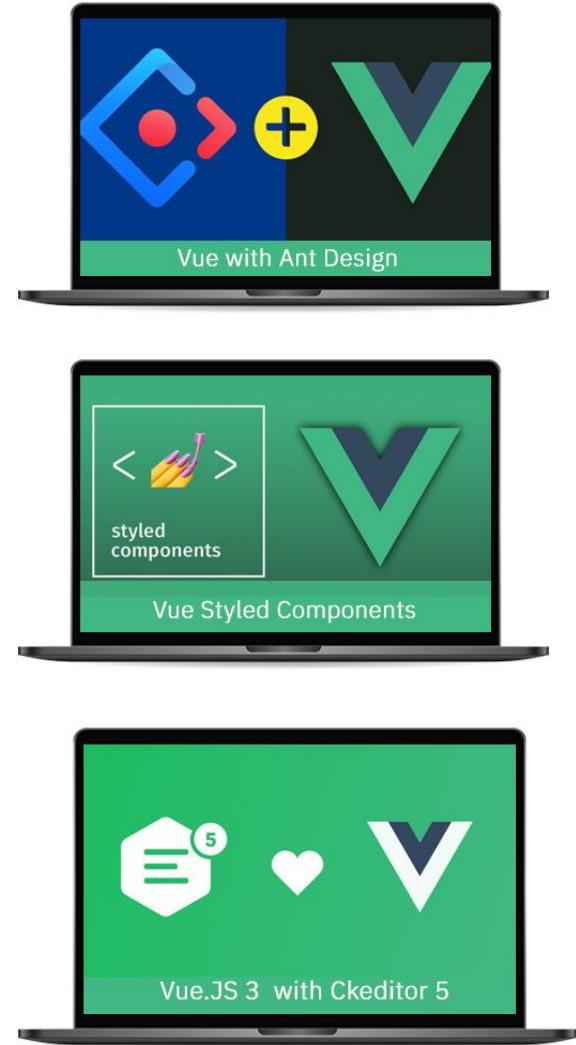
อาจารย์สาเมตร
สถาบันไอโอเก็จเนียส

Vue.js 3 ขั้นสูง

3

11-12,18-19,25-26 November 2022
20.00-23.30 PM

Vue with Ant Design Styled Component and Ckeditor



Ant Design

Default Button Primary Button Dashed Button

Default Button Primary Button Dashed Button

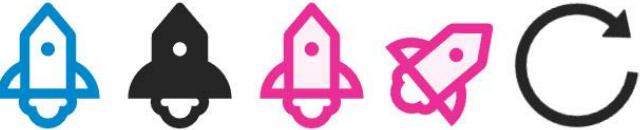
successmessage
lorem ipsum dolor sit amet.

infomessage
lorem ipsum dolor sit amet.

warningmessage
lorem ipsum dolor sit amet.

errormessage
lorem ipsum dolor sit amet.

Click Me!



Open Modal

Delete

Default Button Primary Button Dashed Button

Default Button Primary Button Dashed Button

successmessage
lorem ipsum dolor sit amet.

infomessage
lorem ipsum dolor sit amet.

warningmessage
lorem ipsum dolor sit amet.

errormessage
lorem ipsum dolor sit amet.

X

Basic Modal

Some contents...

Some contents...

Some contents...

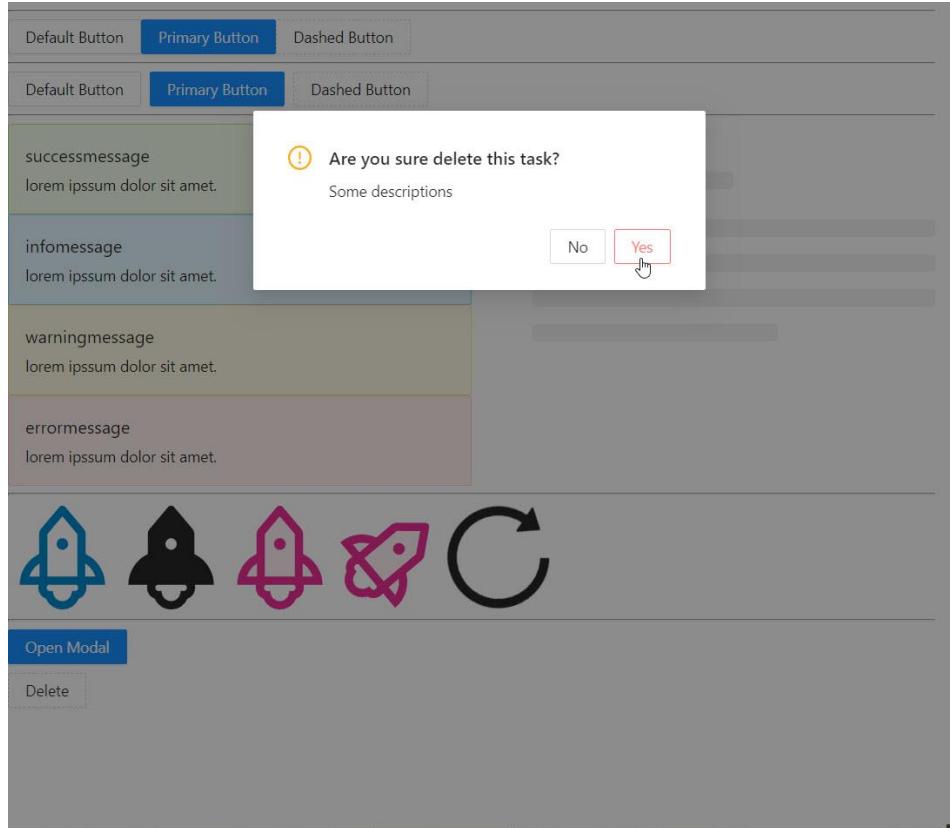
Cancel OK



Open Modal

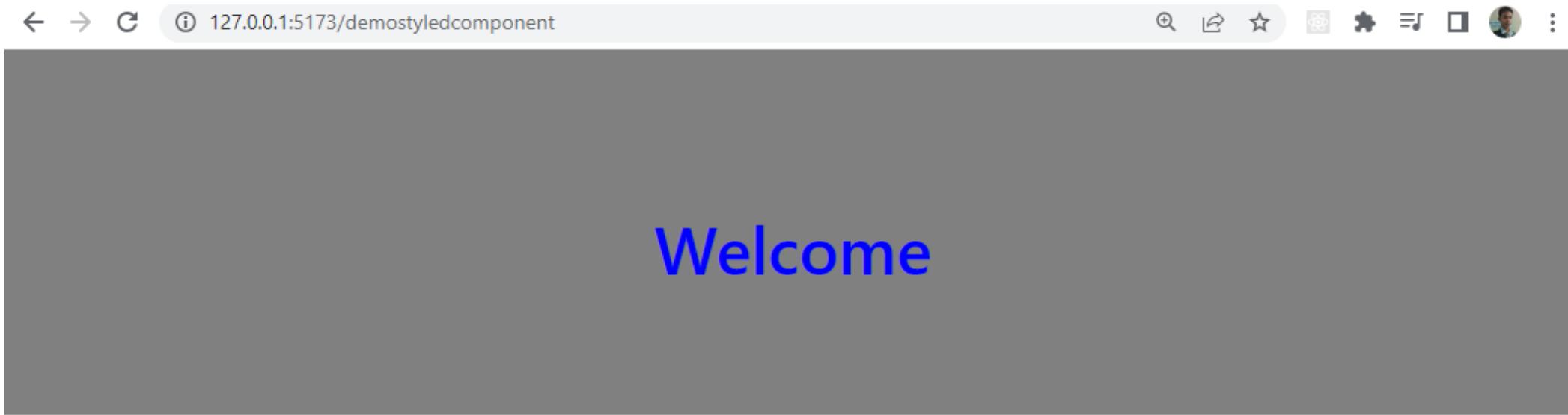
Delete

Ant Design



This screenshot shows a login form interface. The form consists of several input fields and controls. On the left, there is a red asterisk next to the label 'Username:' followed by a red-bordered input field. Below the input field is a red error message: 'Please input your username!'. To the right of the input field is a small circular icon with a password symbol. On the right side of the form, there is a red asterisk next to the label 'Password:' followed by a red-bordered input field. Below the input field is a red error message: 'Please input your password!'. To the right of the input field is a small circular icon with a password symbol. Below these fields is a checked checkbox labeled 'Remember me'. At the bottom right of the form is a large blue 'Submit' button with a white outline and a cursor icon pointing to it.

Styled Component



```
npm install vue3-styled-components  
npm install typescript-styled-plugin
```

Ckeditor 5

Paragraph ▾ | **B** *I* | :| :

Welcome

*No pain to gain
this is my motto*

1. test
2. build
3. go

to my website

- fast
- secure
- easy

this site for every one

[click to view](#)

```
npm install --save @ckeditor/ckeditor5-vue @ckeditor/ckeditor5-build-classic
```

```
<template>
  <a-row type="flex" justify="center" align="middle" style="height: 100vh;">
    <a-col :span="12">
      <ckeditor :editor="editor" v-model="editorData" :config="editorConfig"></ckeditor>
    </a-col>
  </a-row>
</template>

<script setup lang="ts">
  import { ref } from 'vue'
  import ClassicEditor from '@ckeditor/ckeditor5-build-classic'

  const editor = ClassicEditor
  const editorData = '<p>Content of the editor</p>'
  const editorConfig = {}

</script>

<style>
  .ck-editor__editable {
    min-height: 200px;
  }
</style>
```

เก็บตก

4

6 วัน
24 ชั่วโมงเต็ม



สอนสดผ่าน Zoom
รับจำนวนจำกัด



เมวีดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน



อาจารย์สาเมตร
สถาบันไอทีจีเนียส

Vue.js 3 ขั้นสูง

3

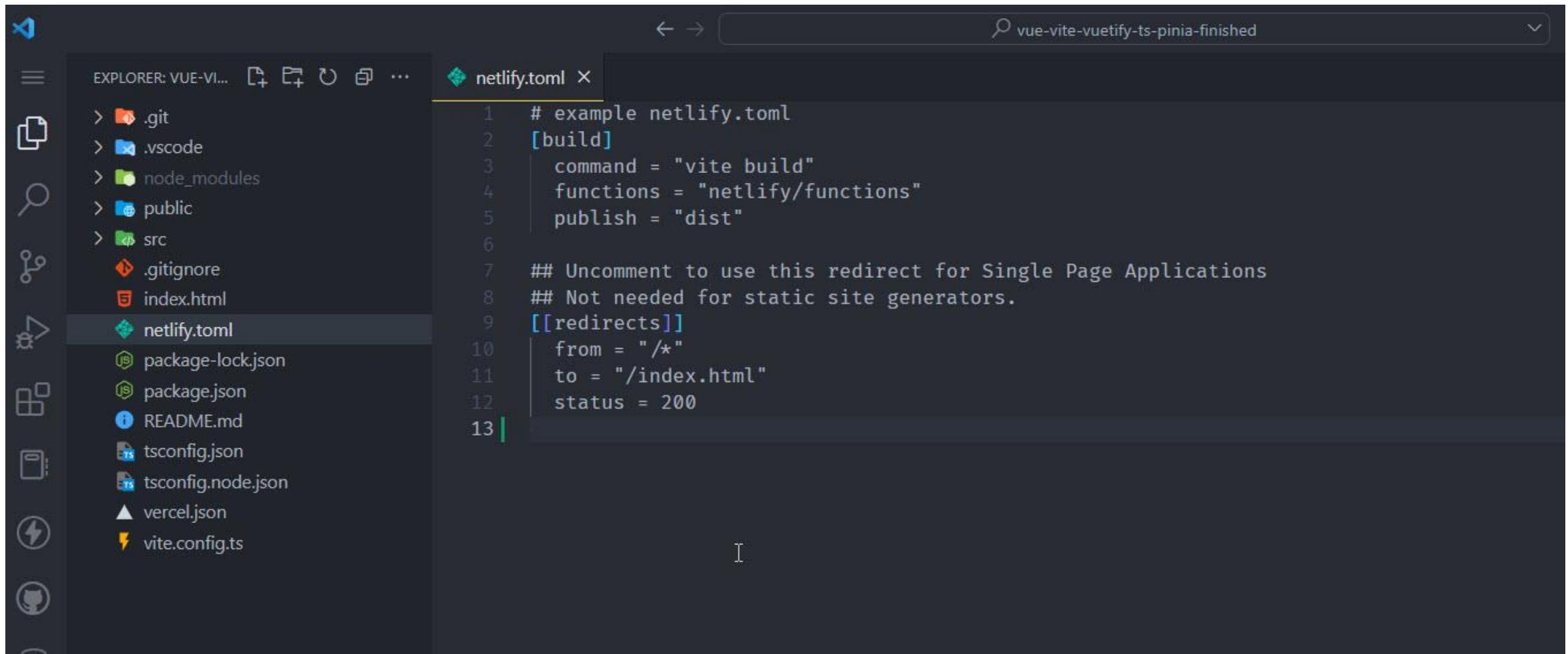
🗓 11-12,18-19,25-26 November 2022

⌚ 20.00-23.30 PM

Vue deployed to Netlify and Vercel



Deployed to Netlify



A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with various project files: .git, .vscode, node_modules, public, src, .gitignore, index.html, netlify.toml (which is selected), package-lock.json, package.json, README.md, tsconfig.json, tsconfig.node.json, vercel.json, and vite.config.ts. The main editor area displays the contents of the netlify.toml file:

```
# example netlify.toml
[build]
  command = "vite build"
  functions = "netlify/functions"
  publish = "dist"

## Uncomment to use this redirect for Single Page Applications
## Not needed for static site generators.
[[redirects]]
  from = "*"
  to = "/index.html"
  status = 200
```

Deployed to Netlify

main ▾ 1 branch 0 tags

Go to file Add file ▾ Code ▾

About

No description, website, or topics provided.

Readme 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

File	Message	Time	Commits
iamsamit initial commit	2d99192	3 minutes ago	1 commit
.vscode	initial commit	3 minutes ago	
public	initial commit	3 minutes ago	
src	initial commit	3 minutes ago	
.gitignore	initial commit	3 minutes ago	
README.md	initial commit	3 minutes ago	
index.html	initial commit	3 minutes ago	
netlify.toml	initial commit	3 minutes ago	
package-lock.json	initial commit	3 minutes ago	
package.json	initial commit	3 minutes ago	
tsconfig.json	initial commit	3 minutes ago	
tsconfig.node.json	initial commit	3 minutes ago	
vercel.json	initial commit	3 minutes ago	
vite.config.ts	initial commit	3 minutes ago	

Deployed to Netlify

app.netlify.com/teams/iamsamitdev/sites

Samit Koyom's team ▾

Upgrade Search anything

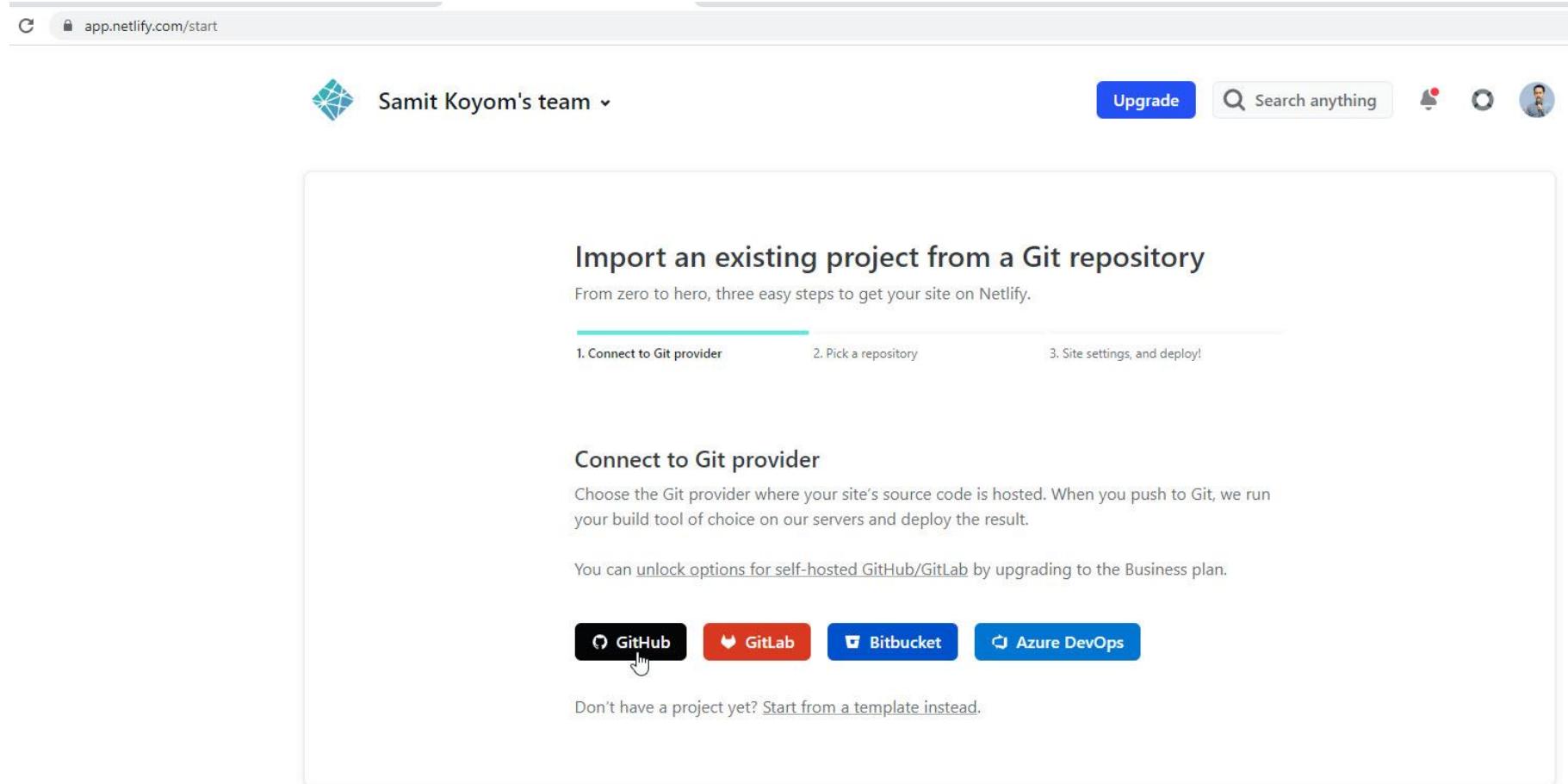
Team overview Sites Builds Integrations Domains Members Audit log Billing Team settings

Add new site ▾

- Import an existing project 
- Start from a template
- Deploy manually

Search sites	
 adv-vuejs3-presentation Manual deploys	Owned by Samit Koyom's team Last published on Nov 25 (6 days ago)
 hotel-design Deploys from GitHub	Owned by Samit Koyom's team Last published on Nov 8 (24 days ago)
 vue3-cmcomplaint Deploys from GitHub	Owned by Samit Koyom's team Last published on Sep 8 (3 months ago)
 samitdev.com Deploys from GitHub with Next.js	Owned by Samit Koyom's team Last published on Apr 22 (7 months ago)

Deployed to Netlify



The screenshot shows the Netlify start page at app.netlify.com/start. The top navigation bar includes the Netlify logo, the team name "Samit Koyom's team", an "Upgrade" button, a search bar, and user profile icons. The main content area is titled "Import an existing project from a Git repository" with the sub-instruction "From zero to hero, three easy steps to get your site on Netlify." Below this, a three-step process is outlined: "1. Connect to Git provider", "2. Pick a repository", and "3. Site settings, and deploy!". The "Connect to Git provider" section features four buttons: GitHub (highlighted with a cursor icon), GitLab, Bitbucket, and Azure DevOps. A note below the GitHub button states: "You can unlock options for self-hosted GitHub/GitLab by upgrading to the Business plan." At the bottom of this section, it says "Don't have a project yet? Start from a template instead."

Deployed to Netlify

The screenshot shows the Netlify interface for importing a project. At the top, there's a navigation bar with a team icon, the text "Samit Koyom's team", an "Upgrade" button, a search bar ("Search anything"), and user profile icons. Below this, a large card titled "Import an existing project from a Git repository" guides the user through three steps: "1. Connect to Git provider", "2. Pick a repository", and "3. Site settings, and deploy!". Step 2 is currently active. The main content area is titled "Pick a repository from GitHub" and asks to choose a repository to link to the site. A search bar contains the query "-vite-vuetify-demo-netlify". Below it, a list shows a single repository: "iamsamitdev/vue-vite-vuetify-demo-netlify". A cursor icon is positioned over the repository name. At the bottom of the card, there's a link: "Can't see your repo here? [Configure the Netlify app on GitHub](#)".

Deployed to Netlify

The screenshot shows the Netlify 'Site settings' page for a repository named 'iamsamitdev/vue-vite-vuetify-demo-netlify'. The top navigation bar indicates the URL as app.netlify.com/start/repos/iamsamitdev%2Fvue-vite-vuetify-demo-netlify. The main title is 'Import an existing project from a Git repository' with a subtitle 'From zero to hero, three easy steps to get your site on Netlify.' Below this, a progress bar shows three steps: '1. Connect to Git provider', '2. Pick a repository', and '3. Site settings, and deploy!' (the current step). The 'Site settings for iamsamitdev/vue-vite-vuetify-demo-netlify' section includes fields for 'Owner' (set to 'Samit Koyom's team') and 'Branch to deploy' (set to 'main'). Under 'Basic build settings', there are fields for 'Base directory' (empty), 'Build command' (set to 'vite build'), and 'Publish directory' (set to 'dist'). A 'Show advanced' button is visible, and at the bottom is a green 'Deploy site' button with a hand cursor icon.

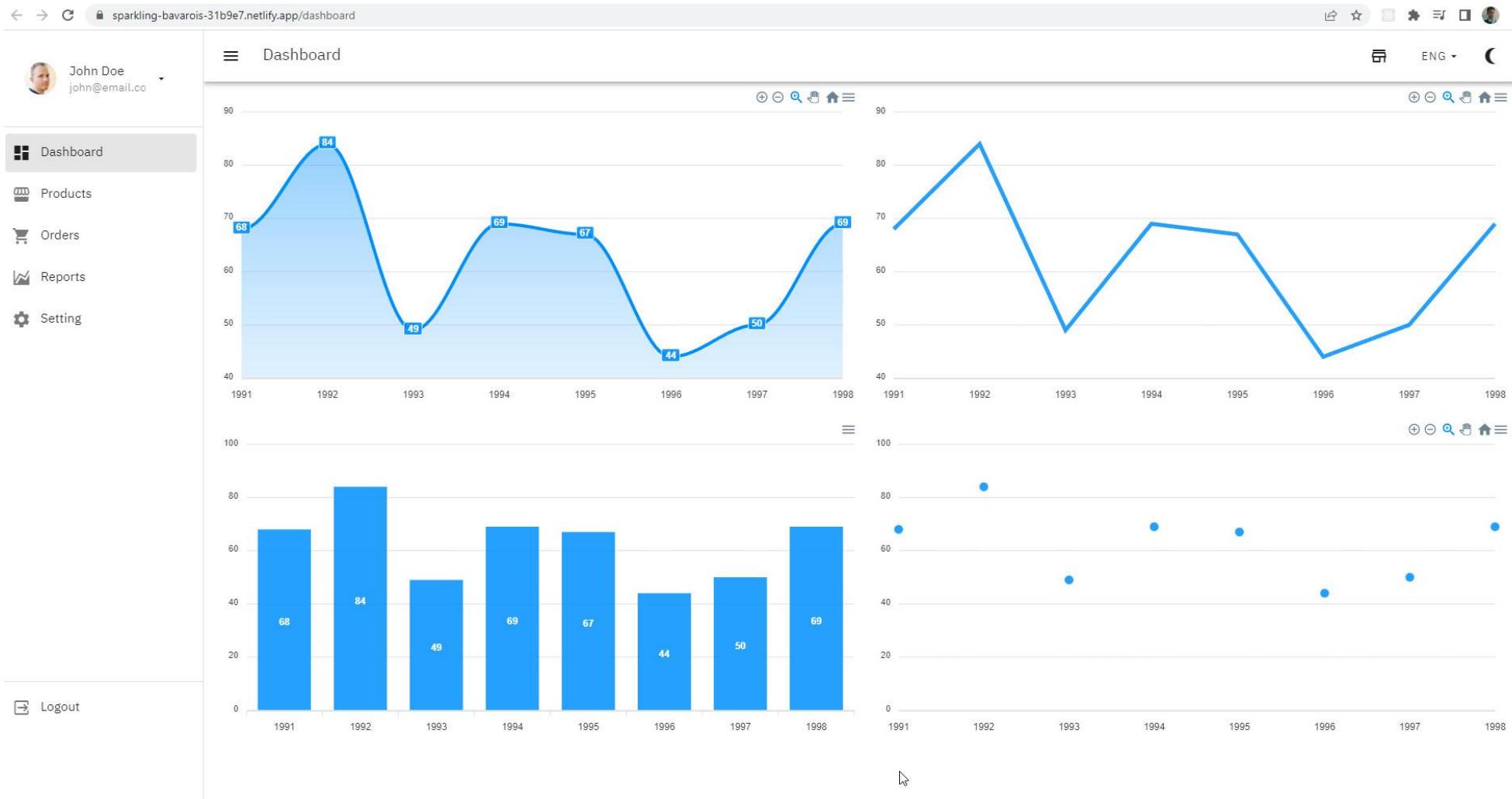
Deployed to Netlify

The screenshot shows the Netlify site overview page for the site 'sparkling-bavarois-31b9e7'. The top navigation bar includes links for Site overview, Deployments, Functions, Edge Functions, Identity, Forms, Large Media, Split Testing, Analytics, Graph, and Site settings. A prominent 'Upgrade' button is visible. The main content area displays the site's URL (<https://sparkling-bavarois-31b9e7.netlify.app>), deployment information from GitHub, and a small placeholder image for the site icon. To the right, a box titled 'Netlify GitHub app permissions' explains the requirement for GitHub org admin permission, with a 'Update permissions' button. Below this, a 'Set up your site' section provides three numbered steps: 1. Your site is deployed ✓ (with a note to try a test build), 2. Set up a custom domain → (with a note to buy a new domain or use an existing one), and 3. Secure your site with HTTPS (noting automatic Let's Encrypt certificate). At the bottom, sections for Production deploys and Deploy Previews are partially visible.

Deployed to Netlify

The screenshot shows the Netlify site overview page for the site 'sparkling-bavarois-31b9e7'. The top navigation bar includes links for Site overview, Deployments, Functions, Edge Functions, Identity, Forms, Large Media, Split Testing, Analytics, Graph, and Site settings. A prominent 'Upgrade' button is visible. The main content area displays the site's URL (<https://sparkling-bavarois-31b9e7.netlify.app>), deployment information (Deployed from GitHub, Created at 1:43 AM), and a 'Site settings' button. To the right, a 'Netlify GitHub app permissions' section is shown, stating that GitHub org admins must grant permission for real-time updates. Below this, a 'Set up your site' section provides three steps: 1) Your site is deployed (with a green checkmark), 2) Set up a custom domain, and 3) Secure your site with HTTPS. The bottom navigation bar includes Production deploys and Deploy Previews.

Deployed to Netlify



Deployed to Vercel

The screenshot shows the VS Code interface with the Explorer sidebar on the left and the Editor pane on the right.

Explorer: Shows the project structure with the following files and folders:

- .git
- .vscode
- node_modules
- public
- src
- .gitignore
- index.html
- package-lock.json
- package.json
- README.md
- tsconfig.json
- tsconfig.node.json
- vercel.json** (selected)
- vite.config.ts

Editor: The `vercel.json` file is open, displaying the following JSON configuration:

```
1  {
2    "rewrites": [
3      { "source": "/(.*)", "destination": "/index.html" }
4    ],
5    "buildCommand": "npm run build",
6    "outputDirectory": "dist"
7  }
```

Deployed to Vercel

The screenshot shows a GitHub repository page for `iamsamitdev/vue-vite-vuetify-demo-vercel`. The repository is public and contains one branch and no tags. The commit history shows an initial commit by `iamsamit` made one minute ago. The repository has 0 stars, 1 watching, and 0 forks. It uses Vue.js (65.5%) and TypeScript (21.2%) as primary languages.

Code | **Issues** | **Pull requests** | **Actions** | **Projects** | **Wiki** | **Security** | **Insights** | **Settings**

About

No description, website, or topics provided.

Readme

Releases

No releases published
Create a new release

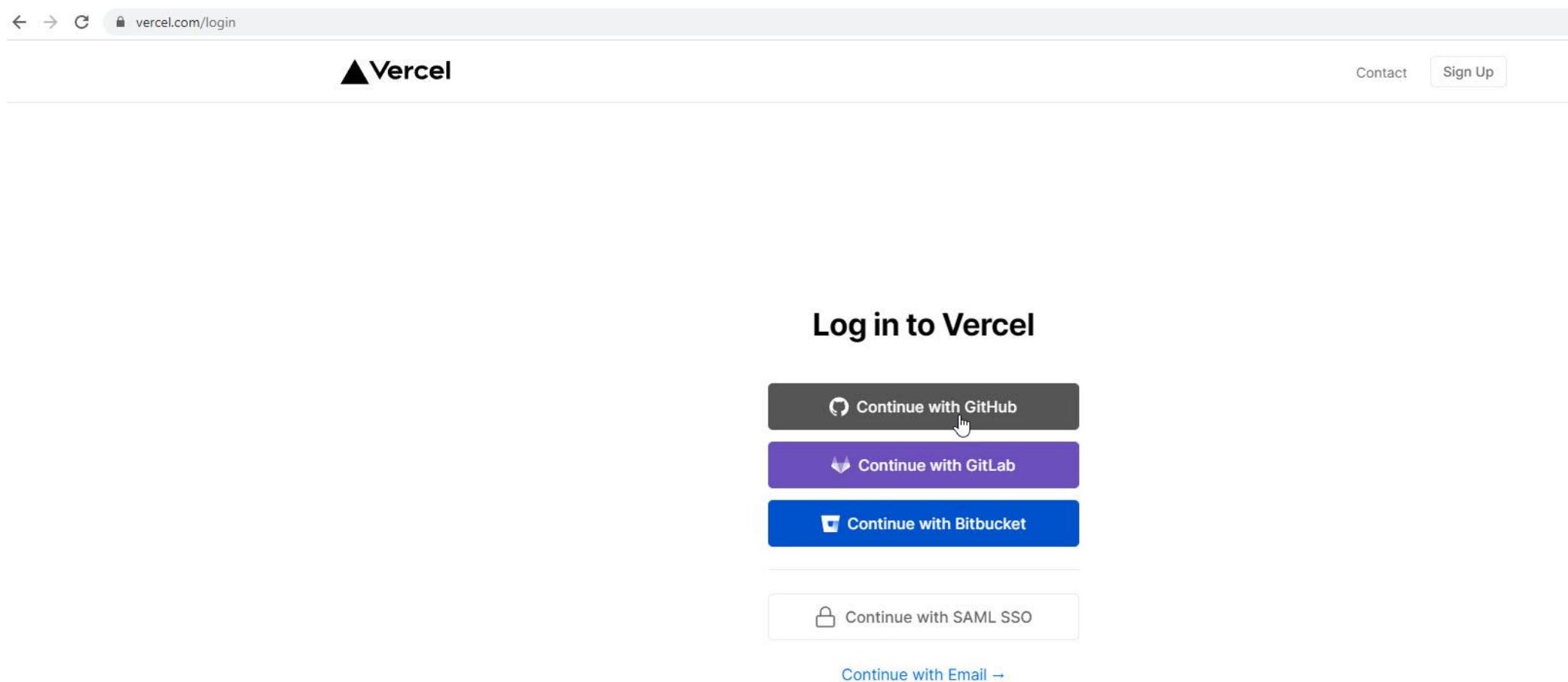
Packages

No packages published
Publish your first package

Languages

Vue 65.5% | TypeScript 21.2%

Deployed to Vercel



Deployed to Vercel

The screenshot shows the Vercel dashboard interface. At the top, there's a header with navigation icons (back, forward, refresh), a lock icon, and the URL "vercel.com/dashboard". Below the header, the user profile "Samit Koyom" is shown with a "Hobby" badge. The main navigation bar includes "Overview" (which is underlined), "Integrations", "Activity", "Domains", "Usage", and "Settings". A search bar with the placeholder "Search..." is positioned above the project cards. On the right side, there are three buttons: "Add New...", "Project" (with a cursor icon), "Domain", and "Team". The dashboard displays three project cards:

- nextjs-tailwind-thaiparliament**
nextjs-tailwind-thaiparliament.vercel...
first commit
From main
33d ago via 🎨
- react-vercel**
react-vercel-wine.vercel.app
No Git Repository connected.
640d ago
- react-adminkit-vercel**
react-adminkit-vercel.vercel.app
No Git Repository connected.
642d ago

Deployed to Vercel

The screenshot shows the Vercel deployment interface. At the top, there's a header with navigation icons and the URL "vercel.com/new". Below the header, a user profile for "Samit Koyom" is shown with a "Hobby" tag. The main content area has a heading "Let's build something new." and a sub-instruction "To deploy a new Project, import an existing Git Repository or get started with one of our Templates." On the left, a card titled "Import Git Repository" offers three options: "Continue with GitHub" (selected), "Continue with GitLab", and "Continue with Bitbucket". It also includes a "Manage Login Connections" link and a "Import Third-Party Git Repository" button. On the right, a card titled "Clone Template" displays five template cards: "Welcome to Next.js", "Next.js", "Next.js Commerce", "SvelteKit", and "Nuxt.js". A "Collaborate with a Team" button is located above the template cards. At the bottom of the template card, there's a "Browse All Templates" link.

Deployed to Vercel

The screenshot shows the Vercel web interface for creating a new project. At the top, there's a header with navigation icons (back, forward, refresh), a URL bar showing 'vercel.com/new', and a user profile for 'Samit Koyom Hobby'. Below the header, the main title 'Let's build something new.' is displayed, followed by a subtitle: 'To deploy a new Project, import an existing Git Repository or get started with one of our Templates.' A 'Collaborate with a Team' button is also present.

Import Git Repository

A dropdown menu shows the user 'iamsamitdev'. A search bar contains 'Search...'. Below it, a list of repositories is shown:

- vue-vite-vuetify-demo-vercel · 4m ago Import
- vue-vite-vuetify-demo-netlify · 17m ago Import
- vue3-vite3-vuetify3-ttypescript-pinia-calendar · 2d ago Import
- vue-vite-vuetify-ts-pinia-finished · 1d ago Import
- vue-vite-vuetify-ts-pinia-day7 · 4d ago Import

[Import Third-Party Git Repository →](#)

Clone Template

Four template cards are shown:

- Next.js**: Welcome to Next.js!
- Next.js Commerce**: A commerce application built with Next.js.
- SvelteKit**: A SvelteKit application.
- Nuxt.js**: A Nuxt.js application.

[Browse All Templates →](#)

Deployed to Vercel

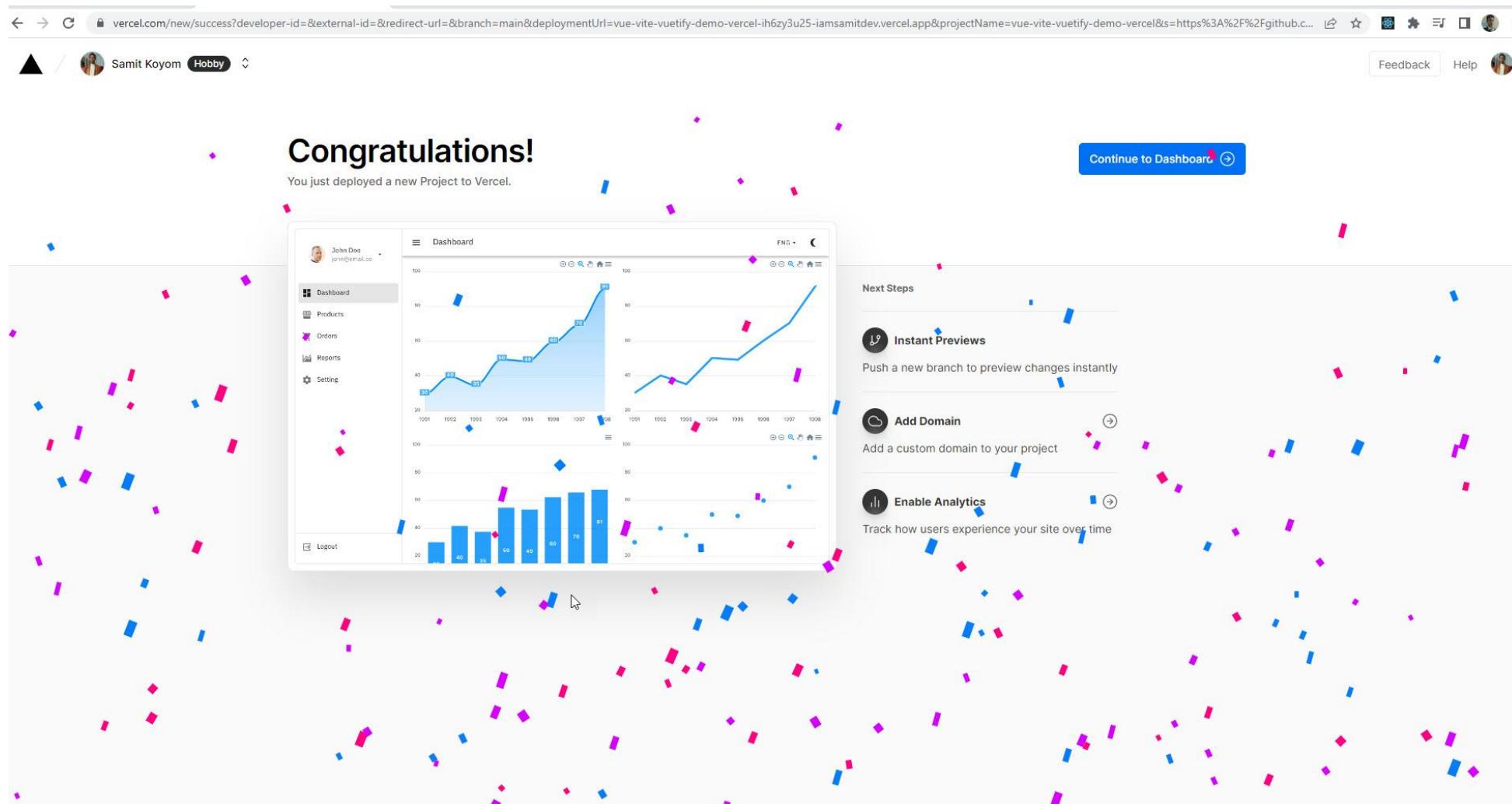
The screenshot shows the Vercel project configuration interface. At the top, there's a header with navigation icons and a URL bar containing the import URL. Below the header, a user profile for Samit Koyom is shown, with a 'Hobby' button. A back arrow and a 'Back' link are also present.

The main area features a large success message: "You're almost done." followed by the instruction: "Please follow the steps to configure your Project and deploy it."

On the left, a sidebar displays the project name "vue-vite-vuetify-demo..." and two main sections: "Configure Project" and "Deploy". Under "GIT REPOSITORY", it shows the repository "iamsamitdev/vue-vite-vuetify-demo..." with a "main" branch selected. There are also links to "Import a different Git Repository" and "Browse Templates".

The right side is the "Configure Project" panel, which includes fields for "Project Name" (set to "vue-vite-vuetify-demo-vercel"), "Framework Preset" (set to "Vite"), "Root Directory" (set to "./"), and "Build and Output Settings" and "Environment Variables" sections. A prominent blue "Deploy" button at the bottom is highlighted with a cursor icon.

Deployed to Vercel



Deployed to Vercel

← → 🔍 vercel.com/iamsamitdev/vue-vite-vuetify-demo-vercel

Samit Koyom Hobby / vue-vite-vuetify-demo-vercel

Project Deployments Analytics Settings Feedback

vue-vite-vuetify-demo-vercel

[View Git Repository](#) [Visit](#)

Production Deployment

The deployment that is available to your visitors.



DEPLOYMENT
vue-vite-vuetify-demo-vercel-ih6zy3u25-iamsamitdev.vercel.app

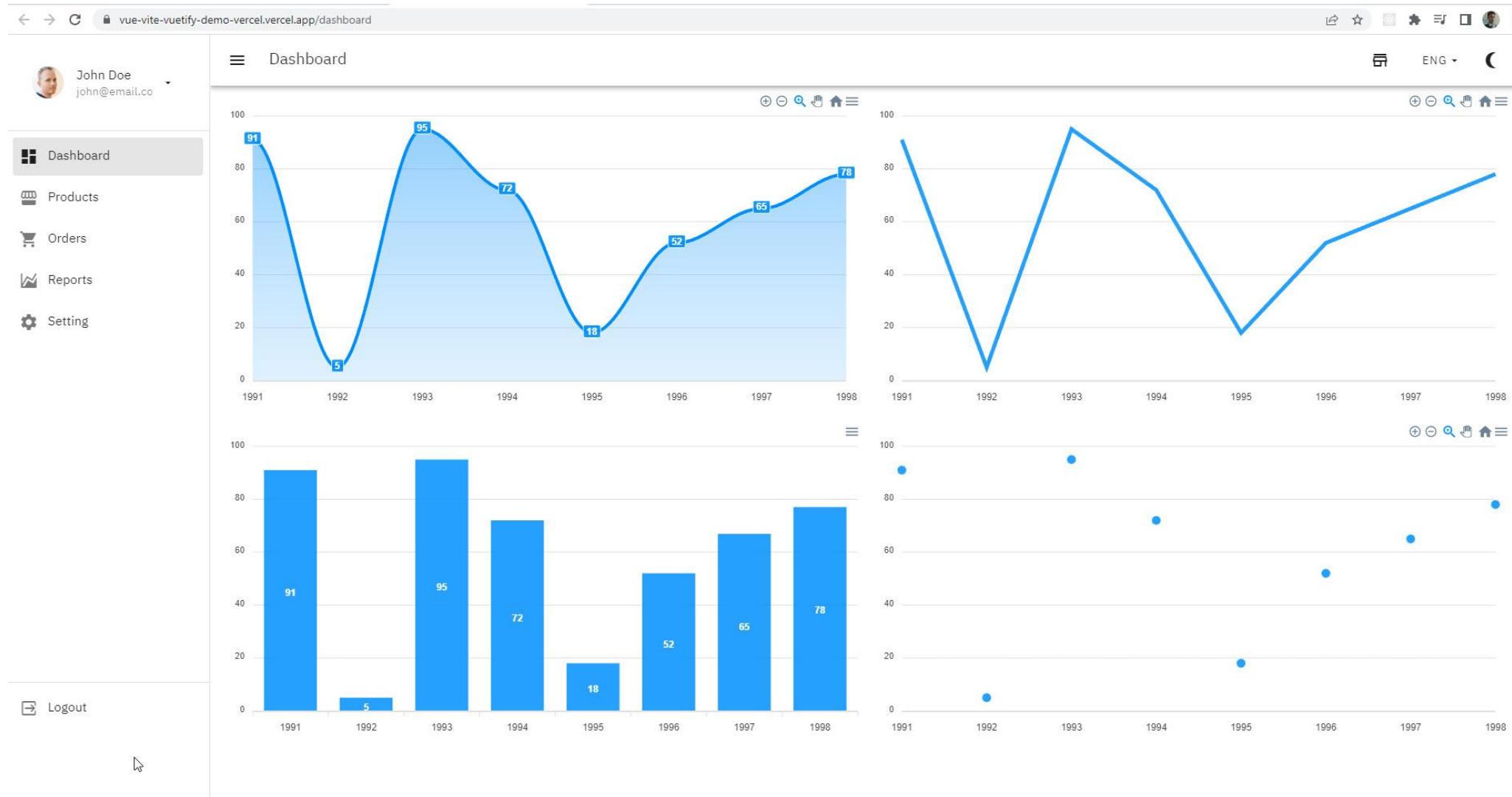
DOMAINS
vue-vite-vuetify-demo-vercel.vercel.app [+2](#)

STATUS CREATED [ⓘ](#)
Ready 1m ago by iamsamit 

BRANCH
main
initial commit

To update your Production Deployment, push to the "main" branch. [Learn More](#)

Deployed to Vercel



Deployed to Vercel

The screenshot shows a web application interface for managing products. The URL in the browser is `vue-vite-vuetify-demo-vercel.vercel.app/product`. The interface is in Korean.

Left Sidebar:

- User profile: John Doe, john@email.co.kr
- Navigation menu:
 - 계기판
 - 제품** (selected)
 - 명령
 - 보고서
 - 설정
- Logout link: 로그 아웃

Top Bar:

- Search icon
- Language: 한국어
- Light/Dark mode switch

Main Content Area:

제품

Product List:

#	이름	가격	관리하다
8	Samsung Galaxy	\$ 489.00	edit trash
7	Airpod 2022	\$ 779.00	edit trash
6	Macbook Pro M1	\$ 899.00	edit trash
5	Lenovo Notebook	\$ 300.00	edit trash
4	Huawei P 30	\$ 255.00	edit trash
3	Samsung Note 20	\$ 459.00	edit trash
2	Apple Watch	\$ 129.00	edit trash
1	iPad 2022	\$ 399.00	edit trash

เก็บตก

5

6 วัน
24 ชั่วโมงเต็ม



สอนสดผ่าน Zoom
รับจำนวนจำกัด



มีวิดีโอบันทึกอบรม
ย้อนหลังให้ทุกวัน



อาจารย์สาเมตร
สถาบันไอโอเก็จเนียส

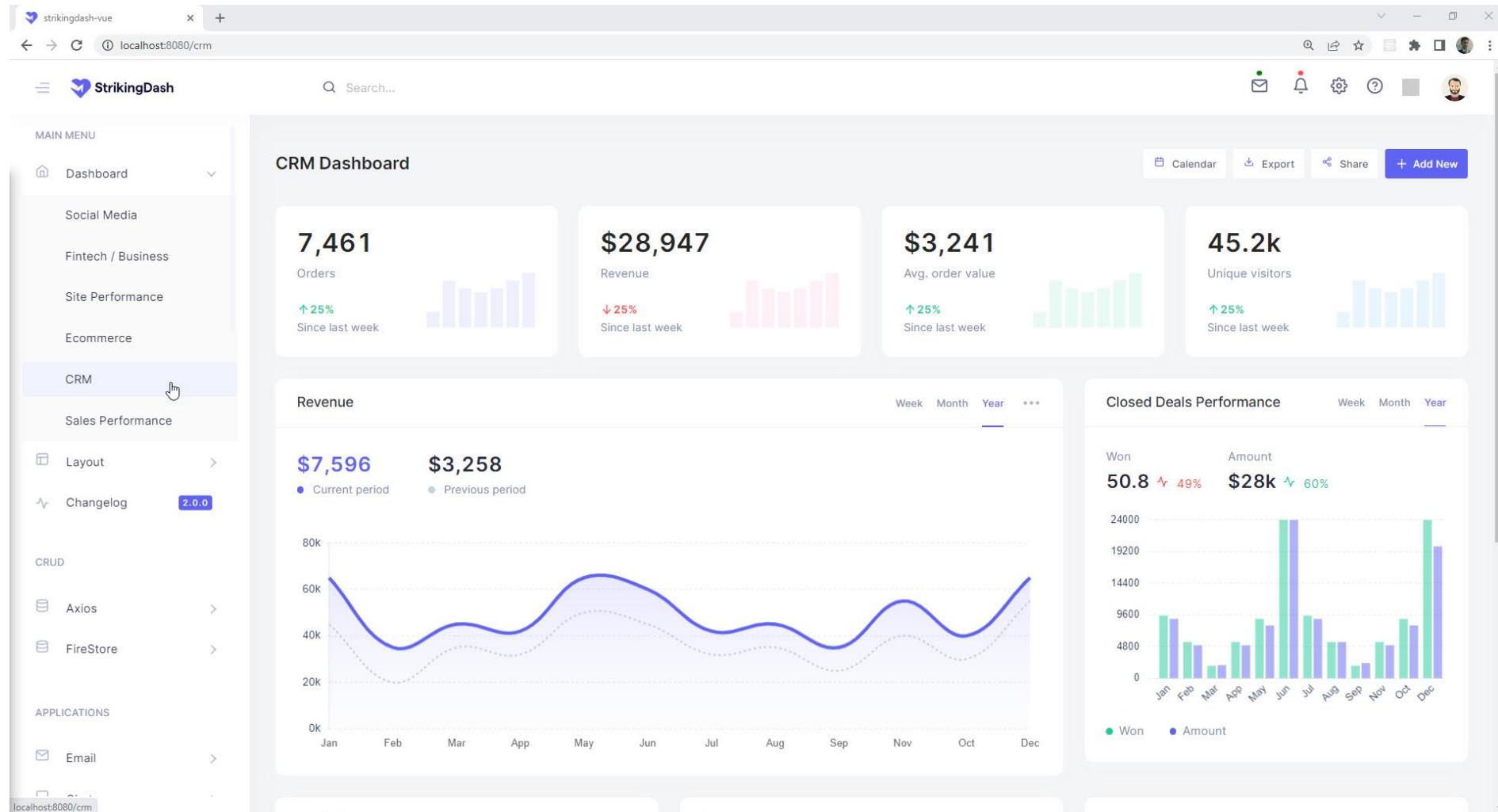
Vue.js 3 ขั้นสูง

3

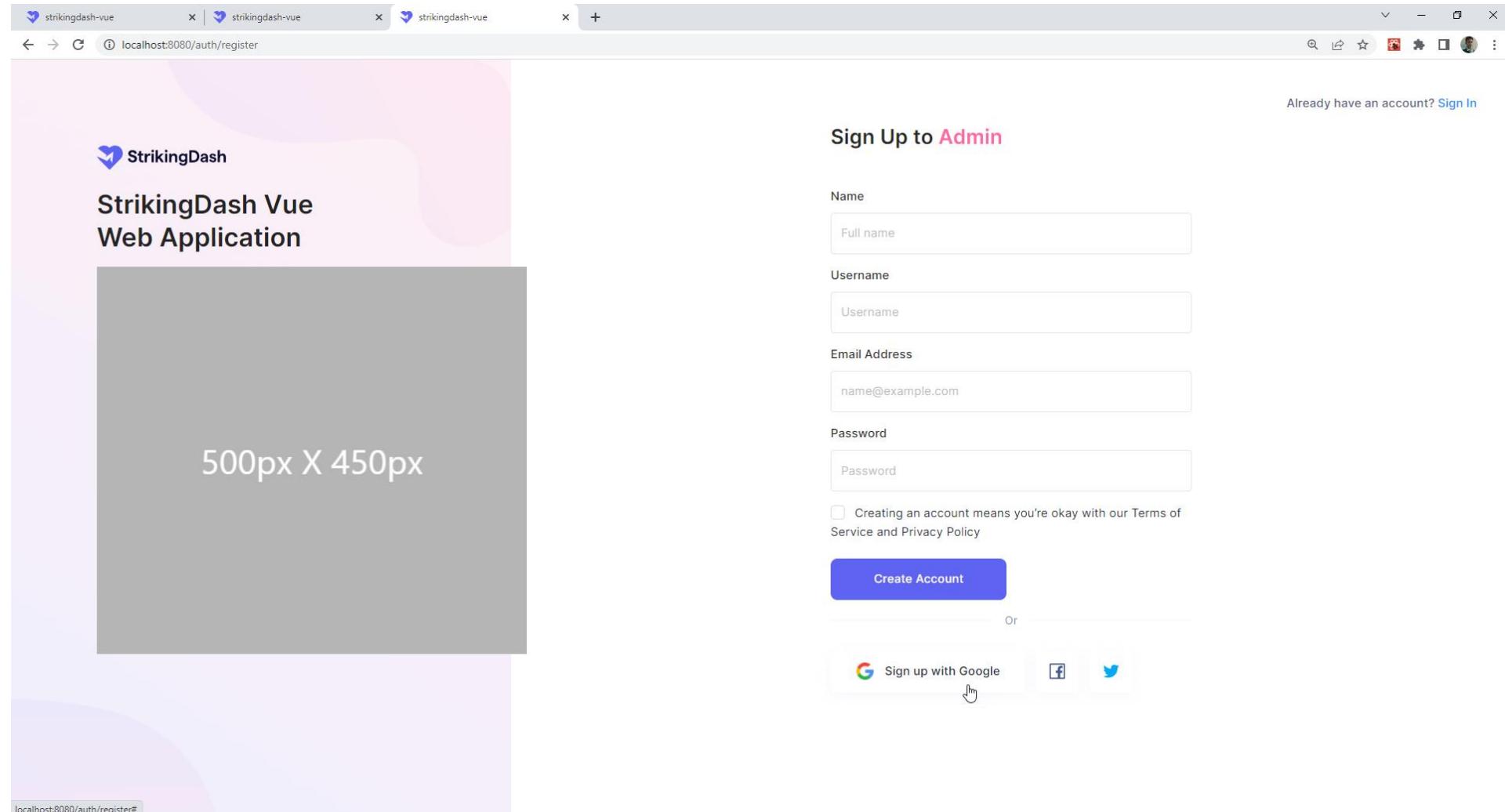
🗓 11-12,18-19,25-26 November 2022

⌚ 20.00-23.30 PM

Free Template Vue 3



Free Template Vue 3



Free Template Vue 3

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer (Left):** Shows the project structure. Key folders include `node_modules`, `public`, `src` (containing `components`, `config`, `database` with `firebase.js`, `dataService`, `icon`, `map`, `theme` with `auth0.js`, `config.js`, `configApp.js`, `core`, `demoData`, `layout`, `routes`, `static`, `utility`, `view`, `vuex`, `modules` (with `authentication` containing `axios` with `actionCreator.js` and `mutations.js`, and `firebase` with `actionCreator.js` and `mutations.js`), `calendar`, `chartContent`, `chat`, `contact`, and `crud`).
- Editor (Center):** Displays the `package.json` file content. The `dependencies` section lists various packages and their versions. The `vue` dependency is highlighted with a blue selection bar.
- Bottom Status Bar:** Shows file statistics: 3 selections (79 characters selected), 2 spaces, CRLF encoding, JSON format, Go Live button, Prettier button, and a refresh icon.

```
26 "auth0-lock": "^11.29.1",
27 "axios": "^0.24.0",
28 "chart.js": "2.9",
29 "core-js": "^3.6.5",
30 "countdown-js": "^1.0.2",
31 "emoji-picker-element": "^1.4.0",
32 "feather-icons": "^4.28.0",
33 "file-saver": "^2.0.5",
34 "firebase": "^8.10.0",
35 "glightbox": "^3.0.9",
36 "jest-worker": "^27.4.6",
37 "js-cookie": "^3.0.1",
38 "leaflet": "^1.7.1",
39 "mitt": "^3.0.0",
40 "moment": "^2.29.1",
41 "svgmap": "^2.7.1",
42 "v-calendar": "^3.0.0-alpha.3",
43 "vue": "^3.2.26",
44 "vue-draggable-next": "^2.0.1",
45 "vue-inline-svg": "^3.0.0-beta.2",
46 "vue-json-csv": "^1.2.12",
47 "vue-masonry": "^0.14.0",
48 "vue-router": "^4.0.4",
49 "vue-types": "^4.1.1",
50 "vue2-daterange-picker": "^0.6.3",
51 "vue3-apexcharts": "^1.2.1",
52 "vue3-google-chart": "^0.0.1",
53 "vue3-google-map": "^0.8.3",
54 "vue3-perfect-scrollbar": "^1.5.5",
55 "vue3-styled-components": "^1.2.1",
56 "vuedraggable": "^4.0.1",
57 "vuex": "^4.0.0",
58 "xlsx": "^0.17.4"
59 },
60 "devDependencies": {
```

Free Template Vue 3

การติดตั้ง lib และ dependencies

npm install --legacy-peer-deps

การสั่ง run โปรเจกต์

npm run serve



سامิตต์ โกยม

ปริญญาโทคณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยี
พระจอมเกล้าพระนครเหนือ

▶ Frontend

Angular, React, NextJS, VueJS,
Bootstrap, TailwindCSS

▶ Backend

PHP, Python, Java, NodeJS, ASP.net

▶ Database

MySQL, PostgreSQL, MS SQL Server,
MongoDB

▶ Mobile

Java, Kotlin, Objective C, Swift, Cordova,
ionic, Flutter

▶ DevOps

Git, Github, Gitlab, Docker,
Kubernetes



อาจารย์سامิต โกยม

สถาบันไอทีจีเนียส



samitkoyom@gmail.com



[iamsamit](#)



[faeacebook.com/iamsamit](#)



[twitter.com/iamsamit](#)



[github.com/iamsamitdev](#)



[linkedin.com/in/samit-koyom-72173348](#)

