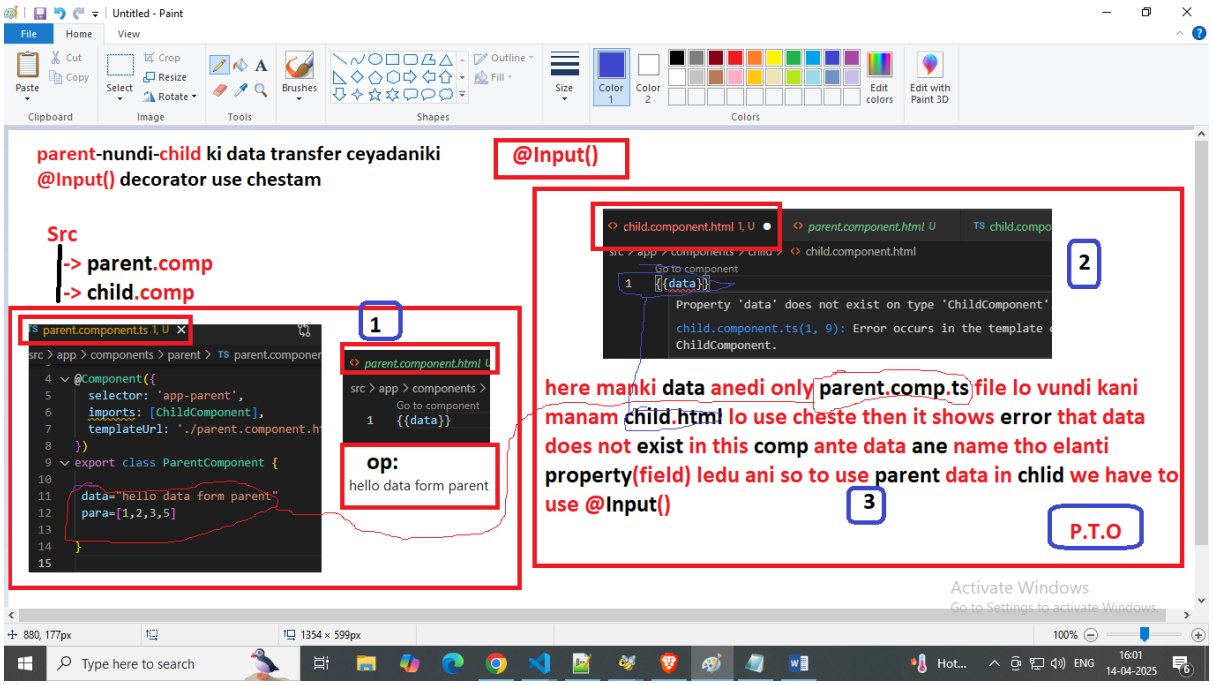
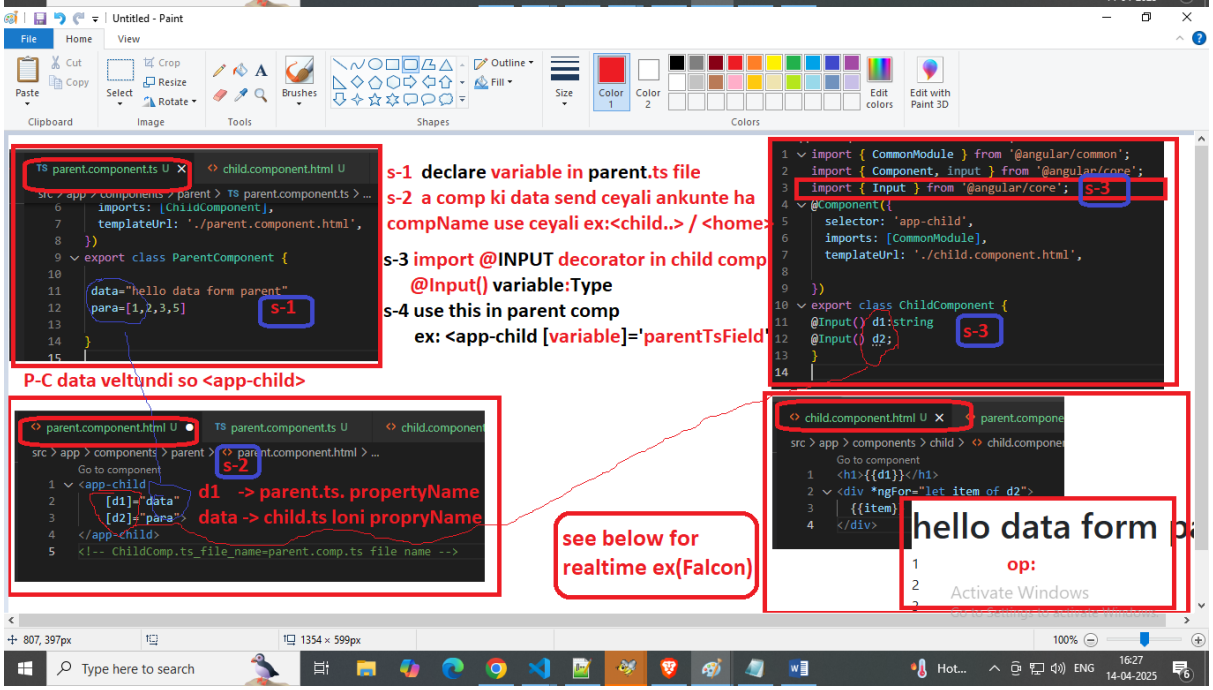
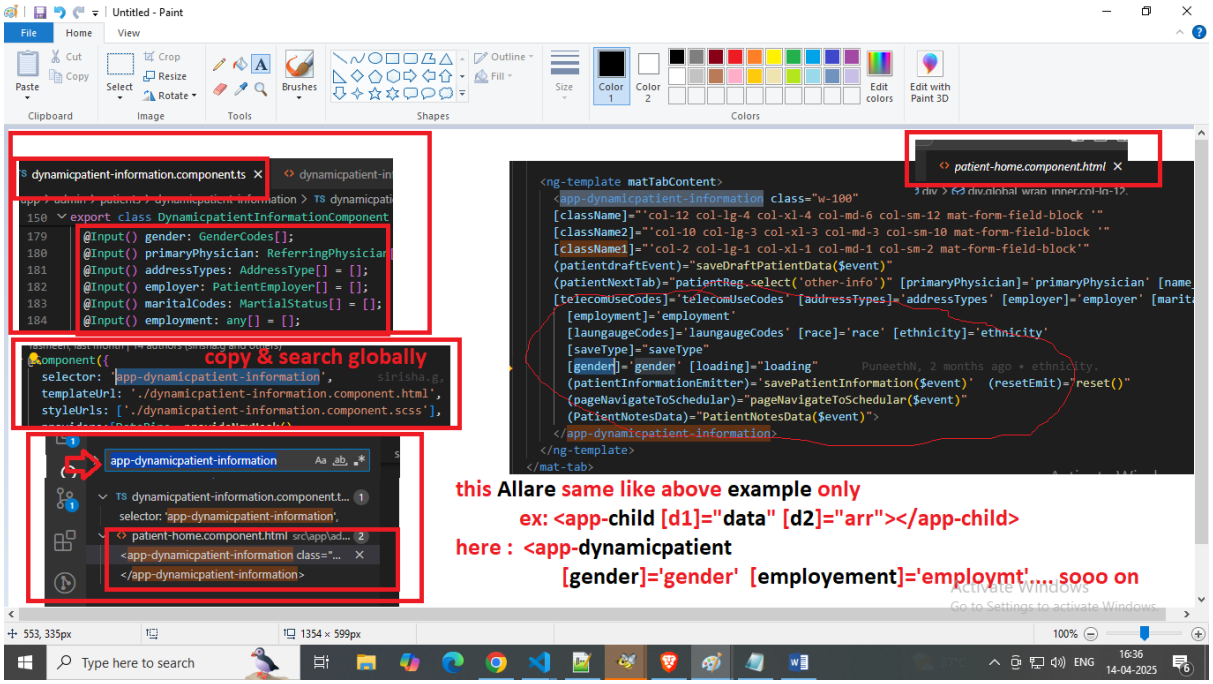
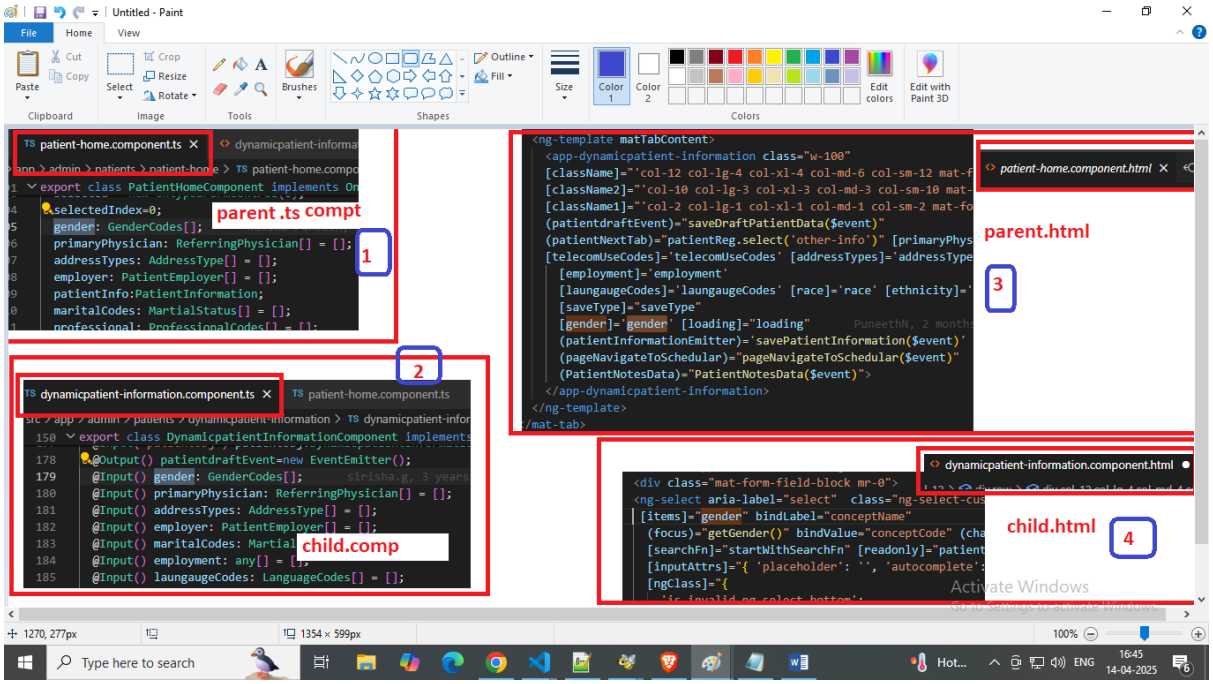
@Input decorator input

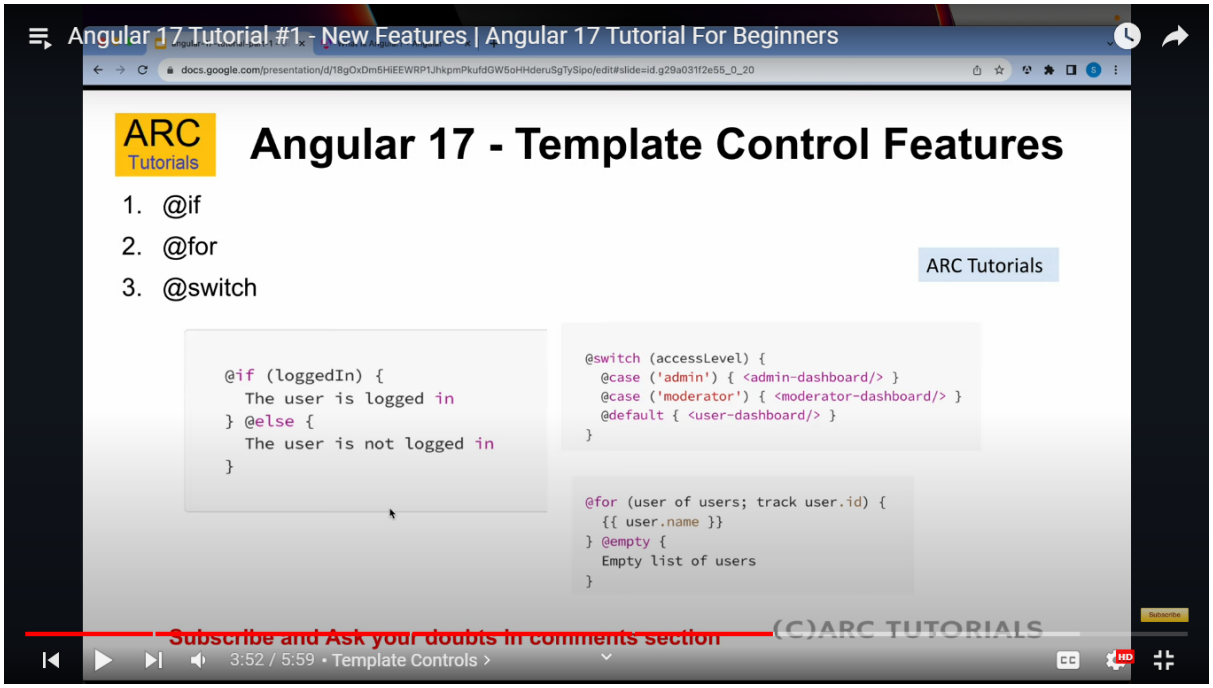


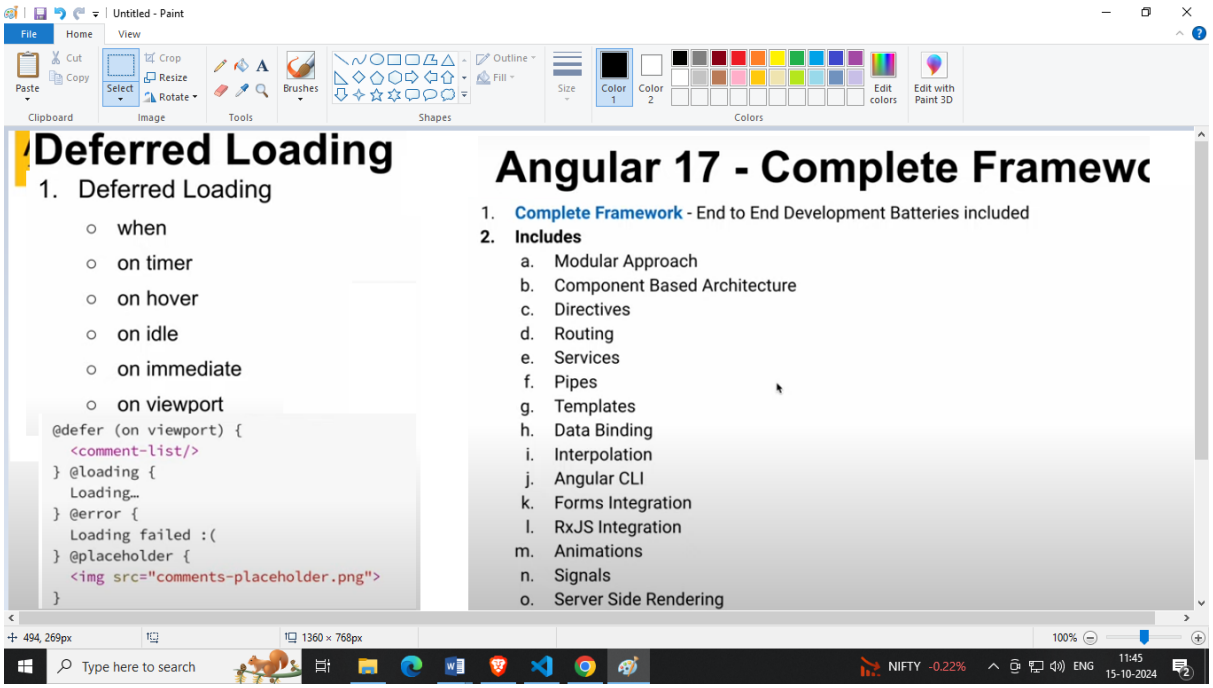


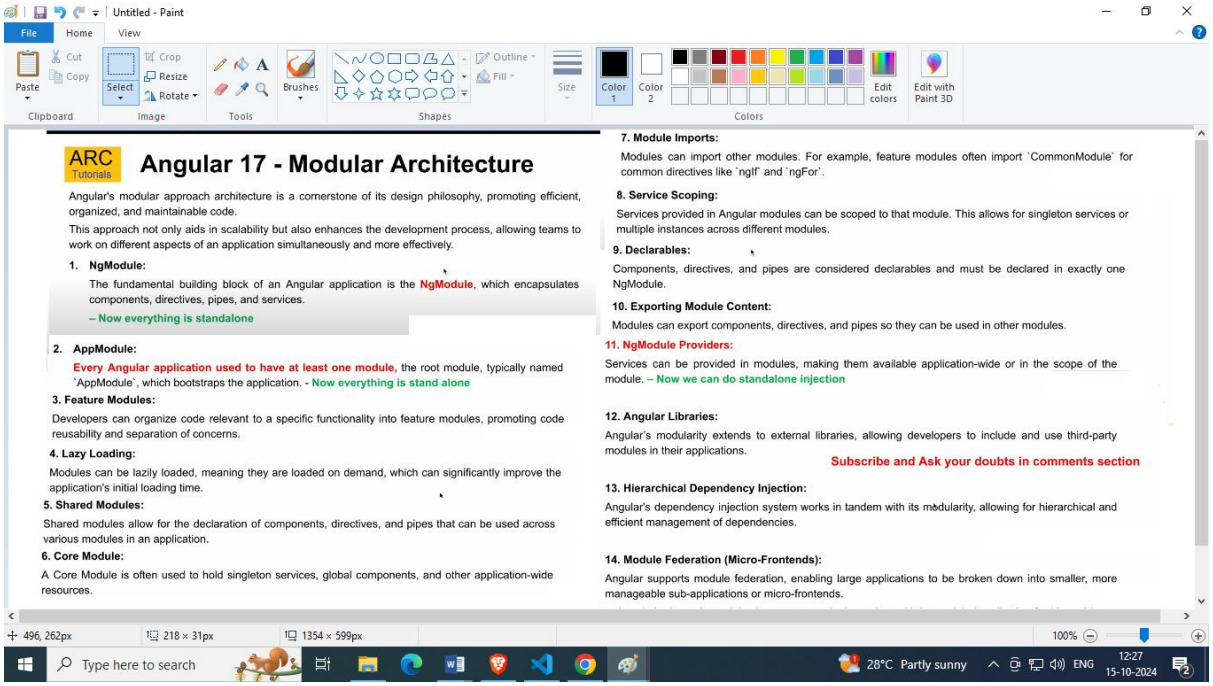

Untitled - Paint
dynamicpatient-information.component.ts
export class DynamicpatientInformationComponent
@Input() gender: GenderCodes[];
@Input() primaryPhysician: ReferringPhysician
@Input() addressTypes: AddressType[] = [];
@Input() employer: PatientEmployer[] = [];
@Input() maritalCodes: MartialStatus[] = [];
@Input() employment: any[] = [];
copy & search globally
selector: 'app-dynamicpatient-information',
templateUrl: './dynamicpatient-information.component.html',
styleUrls: ['./dynamicpatient-information.component.scss'],
app-dynamicpatient-information
patient-home.component.html
<app-dynamicpatient-information class="...
</app-dynamicpatient-information>
<ng-template matTabContent>
<app-dynamicpatient-information class="w-100"
[gender]='gender' [loading]="loading"
(patientInformationEmitter)='savePatientInformation($event)' (resetEmit)="reset()"
(pageNavigateToSchedular)="pageNavigateToSchedular($event)"
(PatientNotesData)="PatientNotesData($event)">
</app-dynamicpatient-information>
</ng-template>
</mat-tab>
this Allare same like above example only
ex: <app-child [d1]="data" [d2]="arr"></app-child>
here : <app-dynamicpatient
[gender]='gender' [employement]='employmt'.... sooo on
Type here to search
ENG
16:36
14-04-2025

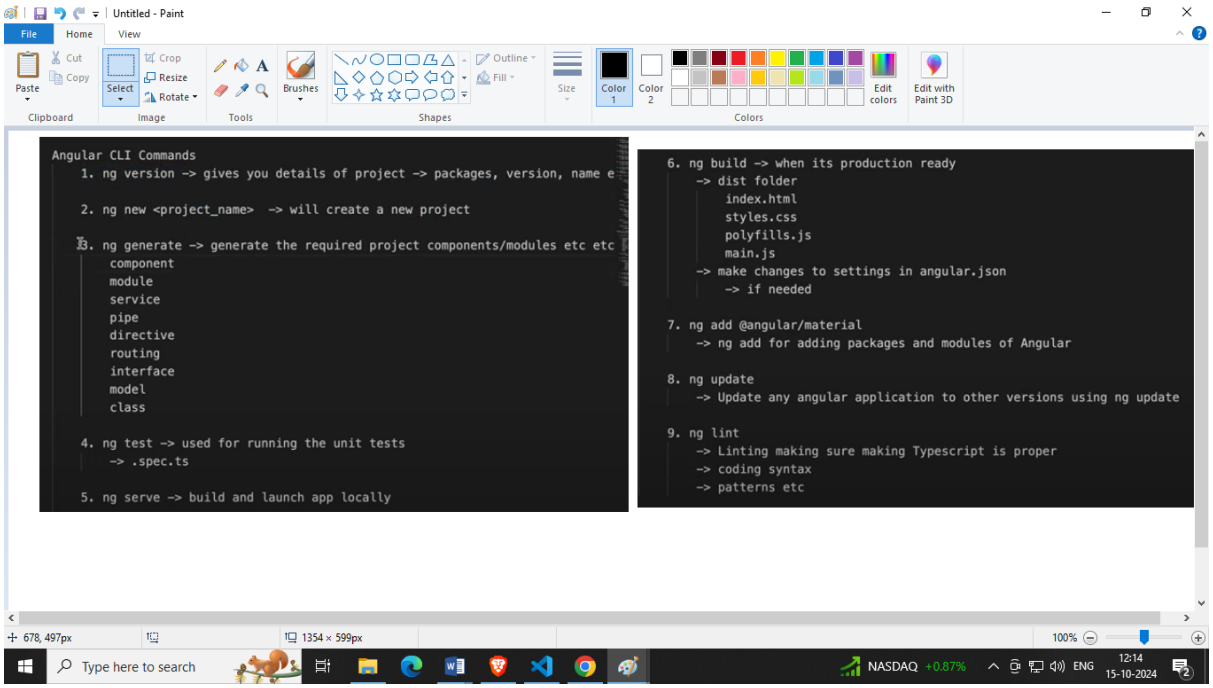

Untitled - Paint
patient-home.component.ts
export class PatientHomeComponent implements On
selectedIndex=0;
gender: GenderCodes[];
primaryPhysician: ReferringPhysician[] = [];
addressTypes: AddressType[] = [];
employer: PatientEmployer[] = [];
patientInfo:PatientInformation;
maritalCodes: MartialStatus[] = [];
parent .ts compt
1
2
dynamicpatient-information.component.ts
patient-home.component.ts
@Output() patientdraftEvent=new EventEmitter();
@Input() gender: GenderCodes[];
@Input() primaryPhysician: ReferringPhysician[] = [];
@Input() addressTypes: AddressType[] = [];
@Input() employer: PatientEmployer[] = [];
@Input() maritalCodes: MartialStatus[] = [];
@Input() employment: any[] = [];
@Input() laungaugeCodes: LanguageCodes[] = [];
child.comp
<ng-template matTabContent>
<app-dynamicpatient-information class="w-100"
(patientdraftEvent)="saveDraftPatientData($event)"
(patientNextTab)="patientReg.select('other-info')" [primaryPhys
[telecomUseCodes]='telecomUseCodes' [addressTypes]='addressType
[employment]='employment'
[laungaugeCodes]='laungaugeCodes' [race]='race' [ethnicity]='
[saveType]="saveType"
[gender]='gender' [loading]="loading"
(patientInformationEmitter)='savePatientInformation($event)'
(pageNavigateToSchedular)="pageNavigateToSchedular($event)"
(PatientNotesData)="PatientNotesData($event)">
</app-dynamicpatient-information>
</ng-template>
</mat-tab>
patient-home.component.html
parent.html
3
dynamicpatient-information.component.html
child.html
4

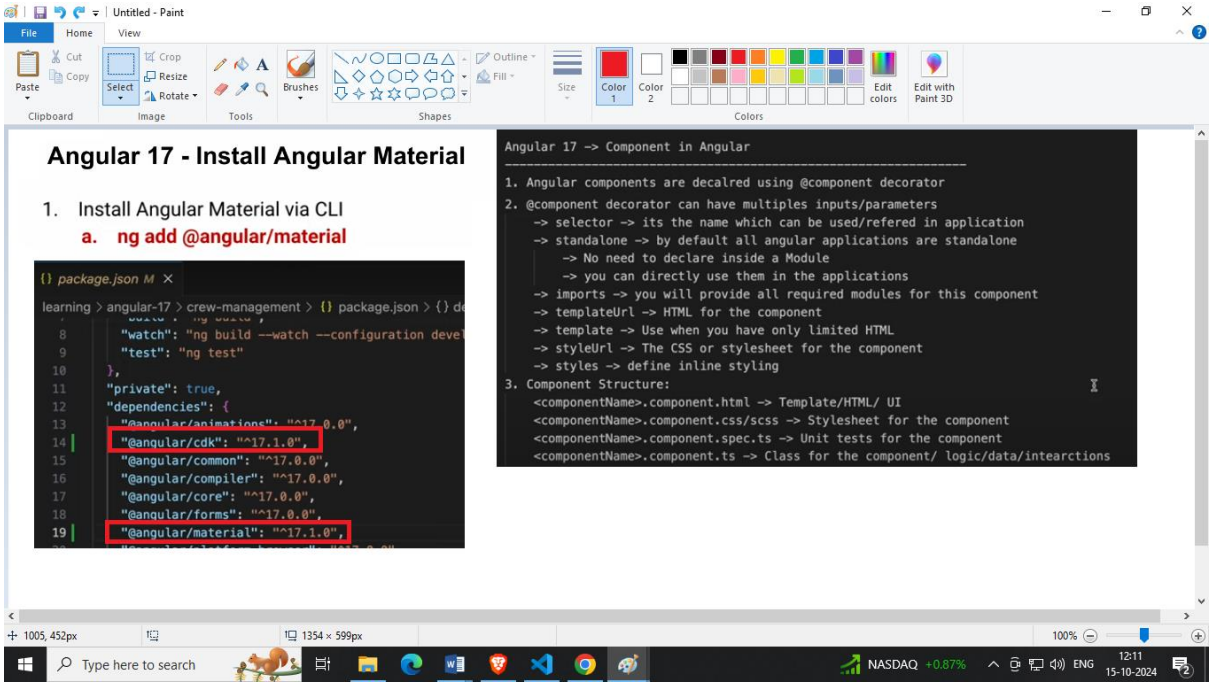
<ng-select aria-label="select" class="ng-select-cus
[items]="gender" bindLabel="conceptName"
(focus)="getGender()" bindValue="conceptCode" (cha
[searchFn]="startWithSearchFn" [readonly]="patient
[inputAttrs]="{ 'placeholder': '', 'autocomplete':
[ngClass]="{
Activate Windows
Type here to search
ENG
16:45
14-04-2025

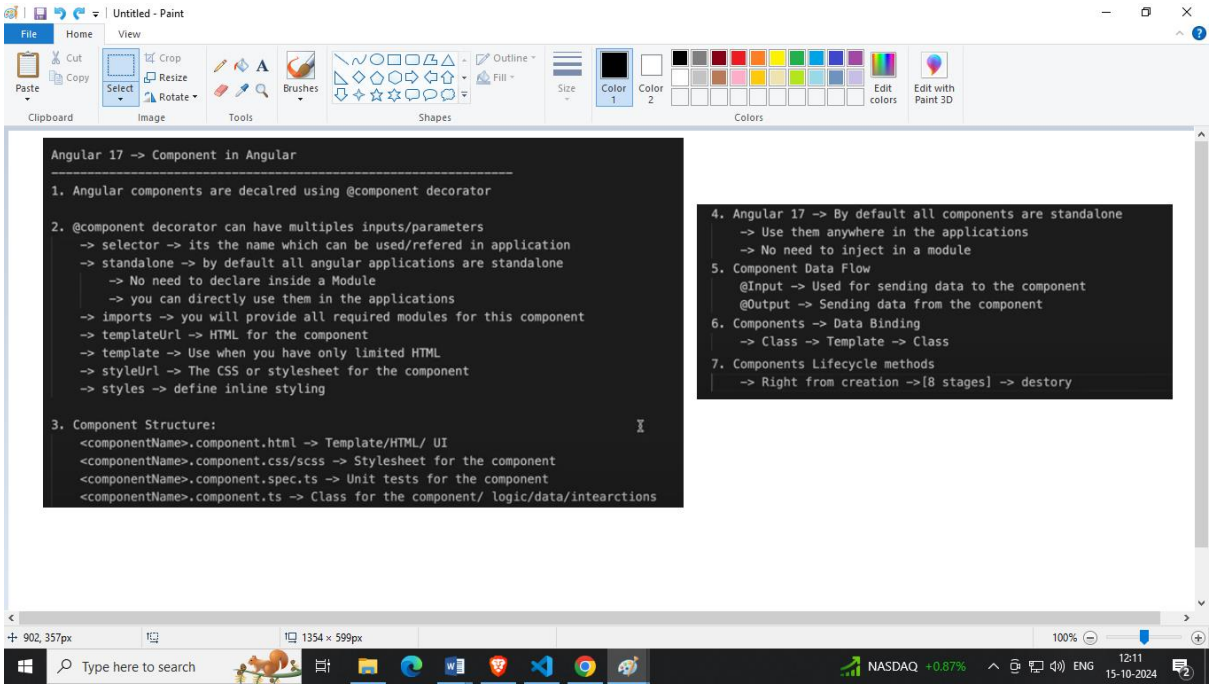

Angular 17 Tutorial #1 - New Features | Angular 17 Tutorial For Beginners
ARC Tutorials
Angular 17 - Template Control Features
1. @if
2. @for
3. @switch
ARC Tutorials
@if (loggedIn) {
The user is logged in
} @else {
The user is not logged in
}
@switch (accessLevel) {
@case ('admin') { <admin-dashboard/> }
@case ('moderator') { <moderator-dashboard/> }
@default { <user-dashboard/> }
}
@for (user of users; track user.id) {
{{ user.name }}
} @empty {
Empty list of users
}
Subscribe and Ask your doubts in comments section
(C)ARC TUTORIALS
3:52 / 5:59 • Template Controls >

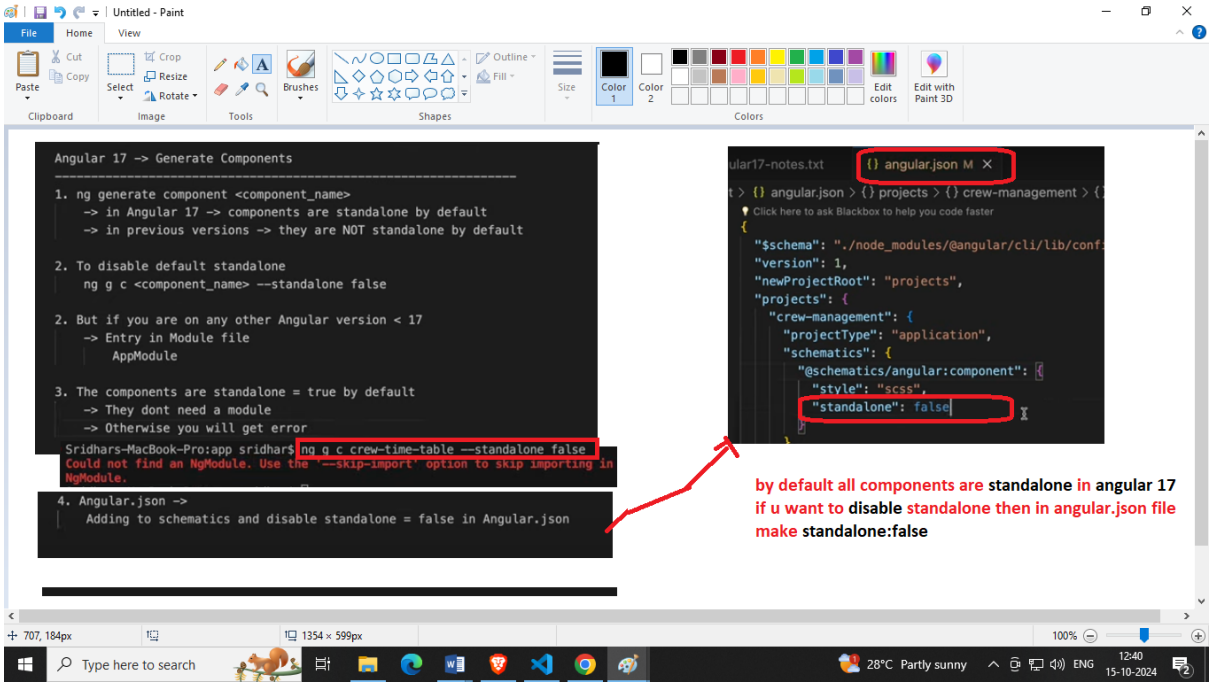

Untitled - Paint
Deferred Loading
1. Deferred Loading
when
on timer
on hover
on idle
on immediate
on viewport
@defer (on viewport) {
<comment-list/>
} @loading {
Loading...
} @error {
Loading failed :(
} @placeholder {
<img src="comments-placeholder.png">
}
Angular 17 - Complete Framewo
1. Complete Framework - End to End Development Batteries included
2. Includes
a. Modular Approach
b. Component Based Architecture
c. Directives
d. Routing
e. Services
f. Pipes
g. Templates
h. Data Binding
i. Interpolation
j. Angular CLI
k. Forms Integration
l. RxJS Integration
m. Animations
n. Signals
o. Server Side Rendering
Type here to search
NIFTY -0.22%
ENG
11:45
15-10-2024

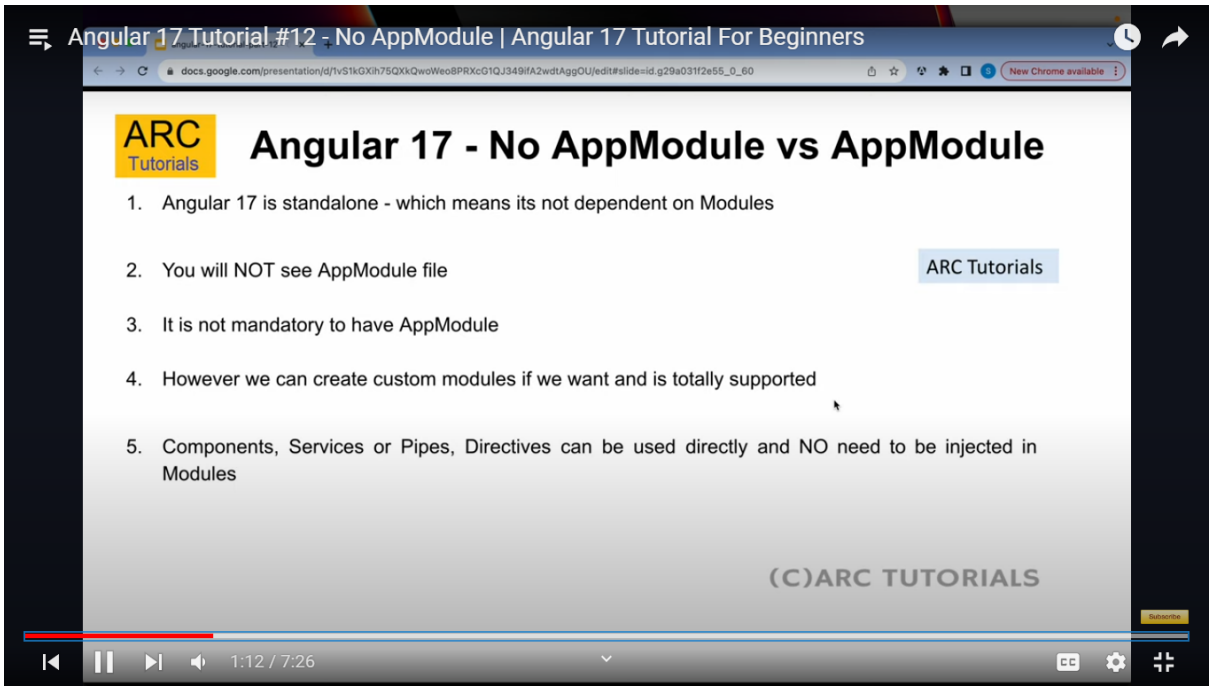

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
ARC Tutorials
Angular 17 - Modular Architecture
Angular's modular approach architecture is a cornerstone of its design philosophy, promoting efficient, organized, and maintainable code.
This approach not only aids in scalability but also enhances the development process, allowing teams to work on different aspects of an application simultaneously and more effectively.
1. NgModule:
The fundamental building block of an Angular application is the NgModule, which encapsulates components, directives, pipes, and services.
– Now everything is standalone
2. AppModule:
Every Angular application used to have at least one module, the root module, typically named `AppModule`, which bootstraps the application. - Now everything is stand alone
3. Feature Modules:
Developers can organize code relevant to a specific functionality into feature modules, promoting code reusability and separation of concerns.
4. Lazy Loading:
Modules can be lazily loaded, meaning they are loaded on demand, which can significantly improve the application's initial loading time.
5. Shared Modules:
Shared modules allow for the declaration of components, directives, and pipes that can be used across various modules in an application.
6. Core Module:
A Core Module is often used to hold singleton services, global components, and other application-wide resources.
7. Module Imports:
Modules can import other modules. For example, feature modules often import `CommonModule` for common directives like `ngIf` and `ngFor`.
8. Service Scoping:
Services provided in Angular modules can be scoped to that module. This allows for singleton services or multiple instances across different modules.
9. Declarables:
Components, directives, and pipes are considered declarables and must be declared in exactly one NgModule.
10. Exporting Module Content:
Modules can export components, directives, and pipes so they can be used in other modules.
11. NgModule Providers:
Services can be provided in modules, making them available application-wide or in the scope of the module. – Now we can do standalone injection
12. Angular Libraries:
Angular's modularity extends to external libraries, allowing developers to include and use third-party modules in their applications.
Subscribe and Ask your doubts in comments section
13. Hierarchical Dependency Injection:
Angular's dependency injection system works in tandem with its modularity, allowing for hierarchical and efficient management of dependencies.
14. Module Federation (Micro-Frontends):
Angular supports module federation, enabling large applications to be broken down into smaller, more manageable sub-applications or micro-frontends.
496, 262px
218 × 31px
1354 × 599px
100%
Type here to search
28°C Partly sunny
ENG
12:27
15-10-2024

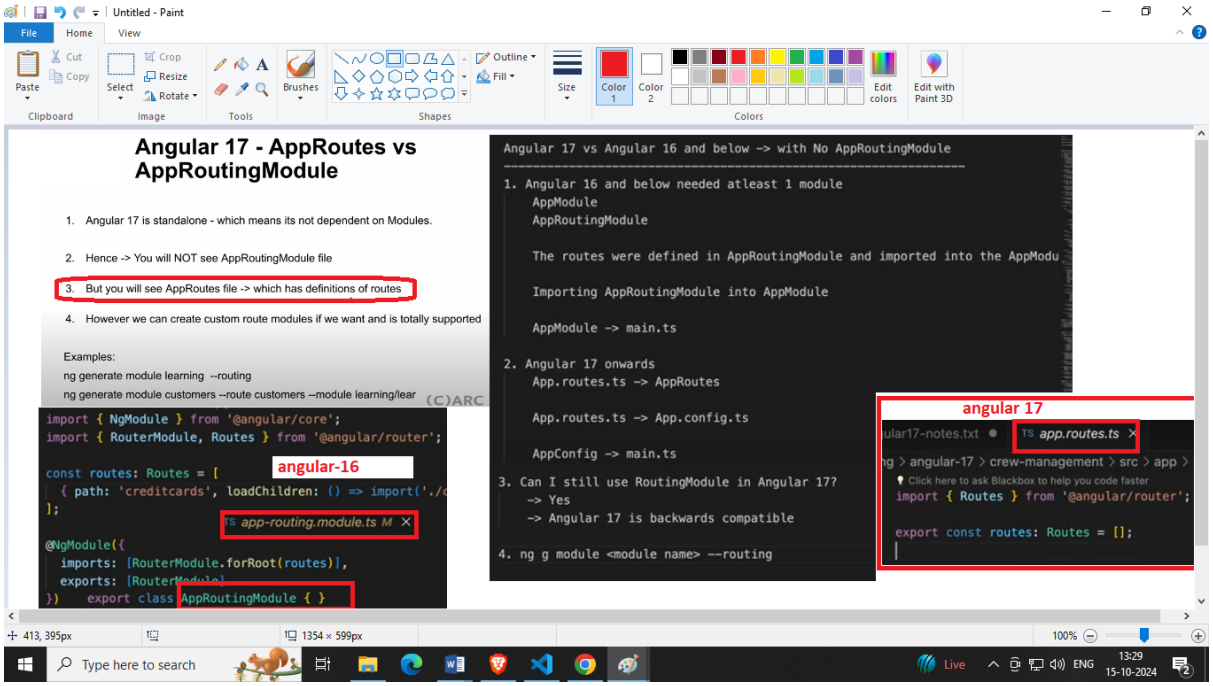

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
Angular CLI Commands
1. ng version -> gives you details of project -> packages, version, name e
2. ng new <project_name> -> will create a new project
3. ng generate -> generate the required project components/modules etc etc
component
module
service
pipe
directive
routing
interface
model
class
4. ng test -> used for running the unit tests
-> .spec.ts
5. ng serve -> build and launch app locally
6. ng build -> when its production ready
-> dist folder
index.html
styles.css
polyfills.js
main.js
-> make changes to settings in angular.json
-> if needed
7. ng add @angular/material
-> ng add for adding packages and modules of Angular
8. ng update
-> Update any angular application to other versions using ng update
9. ng lint
-> Linting making sure making Typescript is proper
-> coding syntax
-> patterns etc
678, 497px
1354 × 599px
100%
Type here to search
NASDAQ +0.87%
ENG
12:14
15-10-2024

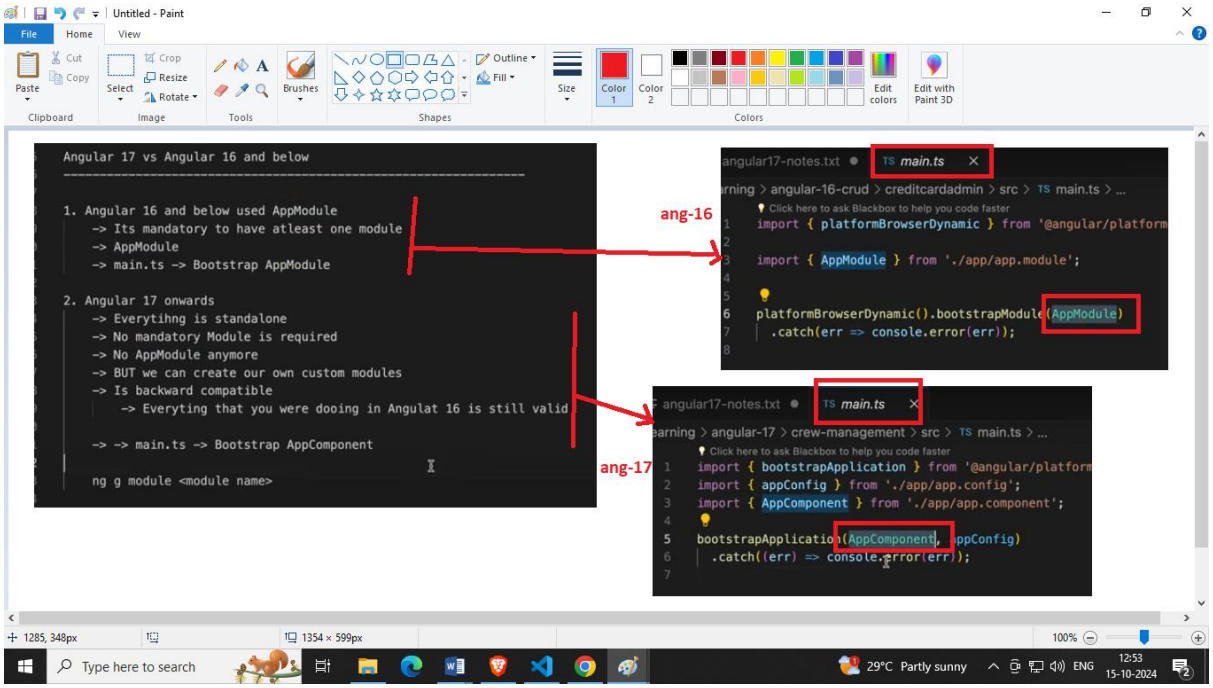

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
Angular 17 - Install Angular Material
1. Install Angular Material via CLI
a. ng add @angular/material
package.json M
learning > angular-17 > crew-management > package.json
"watch": "ng build --watch --configuration deve
"test": "ng test"
"private": true,
"dependencies": {
"@angular/cdk": "^17.1.0",
"@angular/common": "^17.0.0",
"@angular/compiler": "^17.0.0",
"@angular/core": "^17.0.0",
"@angular/forms": "^17.0.0",
"@angular/material": "^17.1.0",
Angular 17 -> Component in Angular
1. Angular components are decalred using @component decorator
2. @component decorator can have multiples inputs/parameters
-> selector -> its the name which can be used/refered in application
-> standalone -> by default all angular applications are standalone
-> No need to declare inside a Module
-> you can directly use them in the applications
-> imports -> you will provide all required modules for this component
-> templateUrl -> HTML for the component
-> template -> Use when you have only limited HTML
-> styleUrl -> The CSS or stylesheet for the component
-> styles -> define inline styling
3. Component Structure:
<componentName>.component.html -> Template/HTML/ UI
<componentName>.component.css/scss -> Stylesheet for the component
<componentName>.component.spec.ts -> Unit tests for the component
<componentName>.component.ts -> Class for the component/ logic/data/intearctions
1005, 452px
1354 × 599px
100%
Type here to search
NASDAQ +0.87%
ENG
12:11
15-10-2024

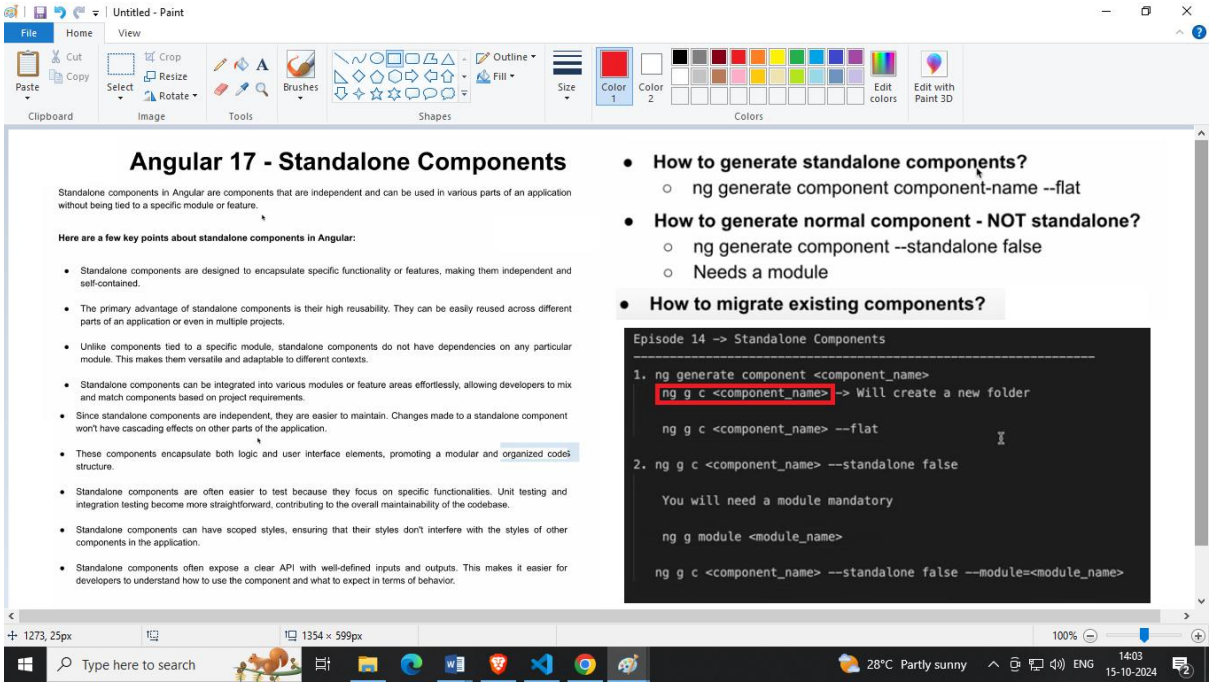

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
Angular 17 -> Component in Angular
1. Angular components are decalred using @component decorator
2. @component decorator can have multiples inputs/parameters
-> selector -> its the name which can be used/refered in application
-> standalone -> by default all angular applications are standalone
-> No need to declare inside a Module
-> you can directly use them in the applications
-> imports -> you will provide all required modules for this component
-> templateUrl -> HTML for the component
-> template -> Use when you have only limited HTML
-> styleUrl -> The CSS or stylesheet for the component
-> styles -> define inline styling
3. Component Structure:
<componentName>.component.html -> Template/HTML/ UI
<componentName>.component.css/scss -> Stylesheet for the component
<componentName>.component.spec.ts -> Unit tests for the component
<componentName>.component.ts -> Class for the component/ logic/data/intearctions
4. Angular 17 -> By default all components are standalone
-> Use them anywhere in the applications
-> No need to inject in a module
5. Component Data Flow
@Input -> Used for sending data to the component
@Output -> Sending data from the component
6. Components -> Data Binding
-> Class -> Template -> Class
7. Components Lifecycle methods
-> Right from creation ->[8 stages] -> destory
902, 357px
1354 × 599px
100%
Type here to search
NASDAQ +0.87%
ENG
12:11
15-10-2024

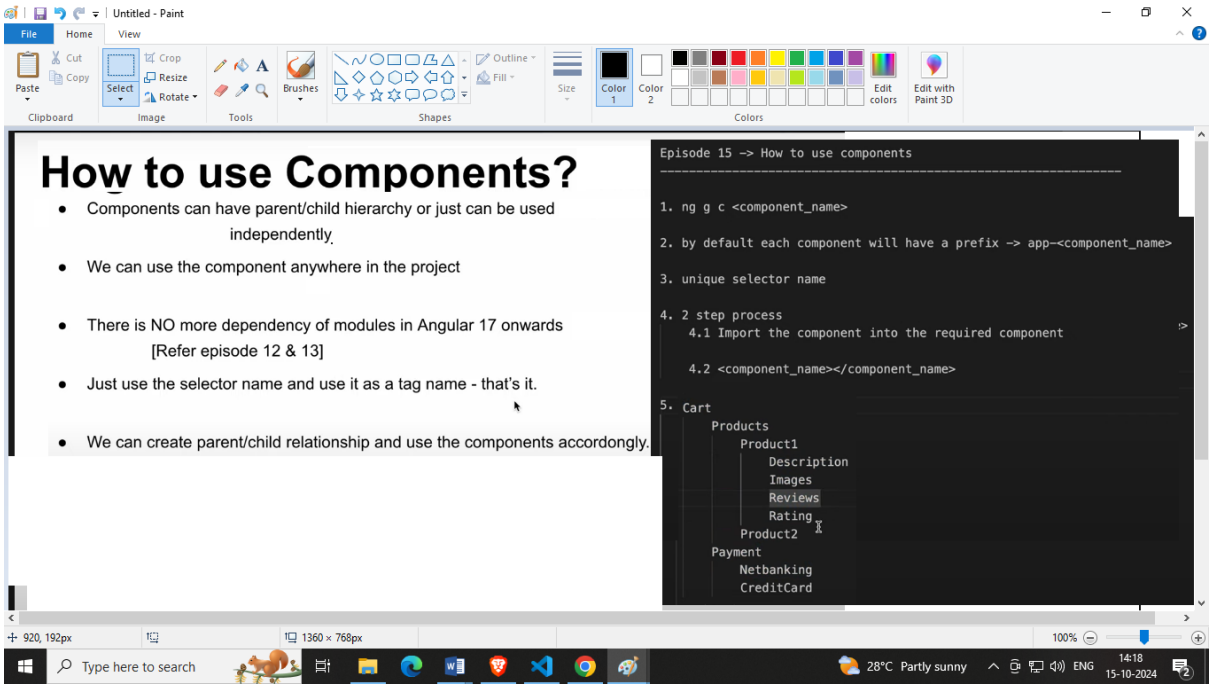

Untitled - Paint
Angular 17 -> Generate Components
1. ng generate component <component_name>
-> in Angular 17 -> components are standalone by default
-> in previous versions -> they are NOT standalone by default
2. To disable default standalone
ng g c <component_name> --standalone false
2. But if you are on any other Angular version < 17
-> Entry in Module file
AppModule
3. The components are standalone = true by default
-> They dont need a module
-> Otherwise you will get error
Sridhars-MacBook-Pro:app sridhar$ ng g c crew-time-table --standalone false
Could not find an NgModule. Use the '--skip-import' option to skip importing in NgModule.
4. Angular.json ->
Adding to schematics and disable standalone = false in Angular.json
{} angular.json M
"$schema": "./node_modules/@angular/cli/lib/conf
"version": 1,
"newProjectRoot": "projects",
"projects": {
"crew-management": {
"projectType": "application",
"schematics": {
"@schematics/angular:component": {
"style": "scss",
"standalone": false
by default all components are standalone in angular 17
if u want to disable standalone then in angular.json file
make standalone:false
Type here to search
28°C Partly sunny
ENG
12:40
15-10-2024

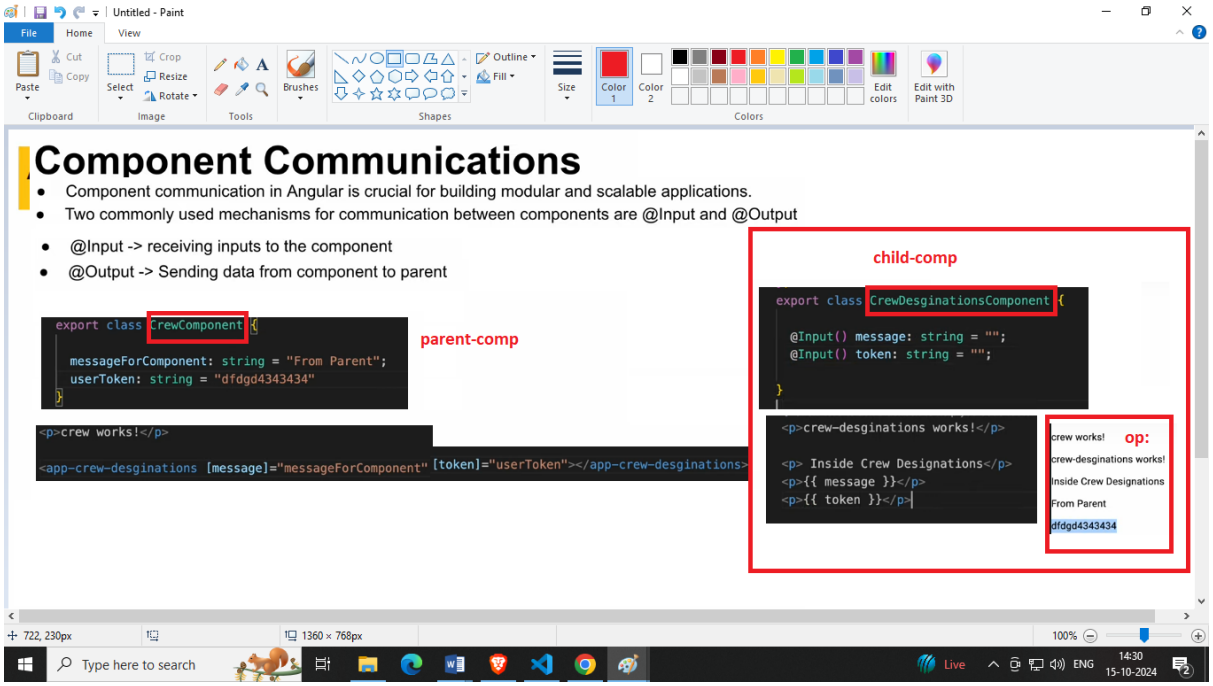

Angular 17 Tutorial #12 - No AppModule | Angular 17 Tutorial For Beginners
ARC Tutorials
Angular 17 - No AppModule vs AppModule
1. Angular 17 is standalone - which means its not dependent on Modules
2. You will NOT see AppModule file
ARC Tutorials
3. It is not mandatory to have AppModule
4. However we can create custom modules if we want and is totally supported
5. Components, Services or Pipes, Directives can be used directly and NO need to be injected in Modules
(C)ARC TUTORIALS
1:12 / 7:26

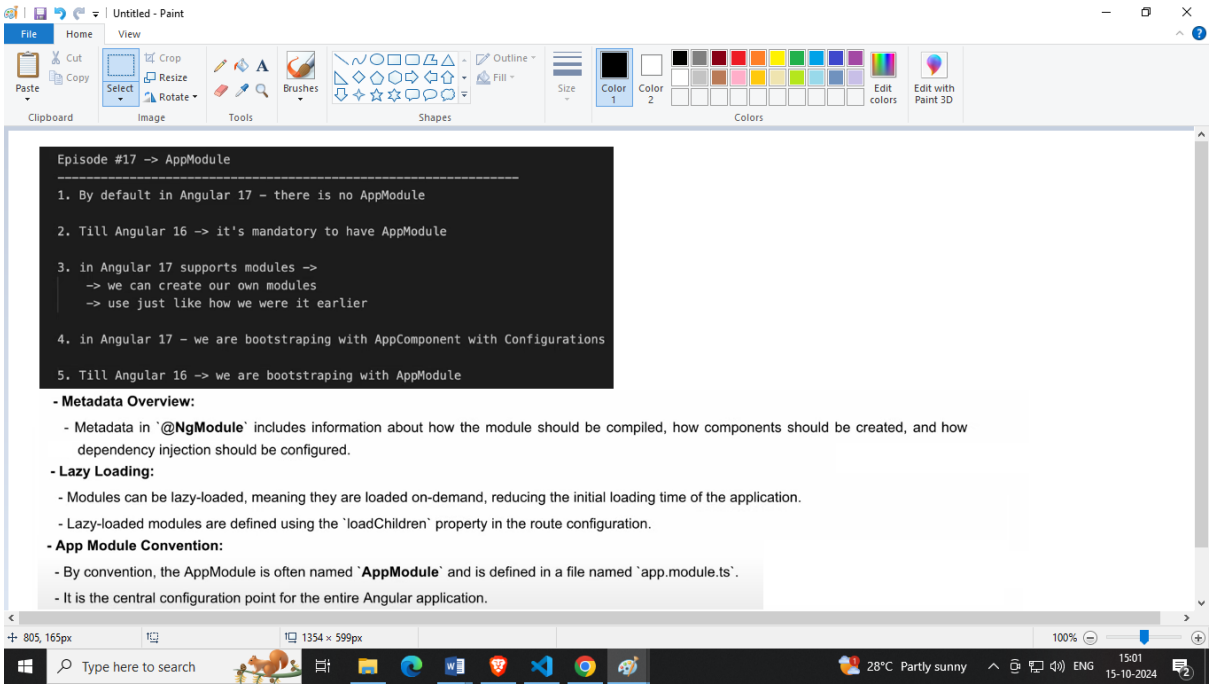

Untitled - Paint
Angular 17 - AppRoutes vs AppRoutingModule
1. Angular 17 is standalone - which means its not dependent on Modules.
2. Hence -> You will NOT see AppRoutingModule file
3. But you will see AppRoutes file -> which has definitions of routes
4. However we can create custom route modules if we want and is totally supported
Examples:
ng generate module learning --routing
ng generate module customers --route customers --module learning/lear
Angular 17 vs Angular 16 and below -> with No AppRoutingModule
1. Angular 16 and below needed atleast 1 module
AppModule
AppRoutingModule
The routes were defined in AppRoutingModule and imported into the AppModu
Importing AppRoutingModule into AppModule
AppModule -> main.ts
2. Angular 17 onwards
App.routes.ts -> AppRoutes
App.routes.ts -> App.config.ts
AppConfig -> main.ts
3. Can I still use RoutingModule in Angular 17?
-> Yes
-> Angular 17 is backwards compatible
4. ng g module <module name> --routing
angular-16
app-routing.module.ts M
angular 17
app.routes.ts
Type here to search

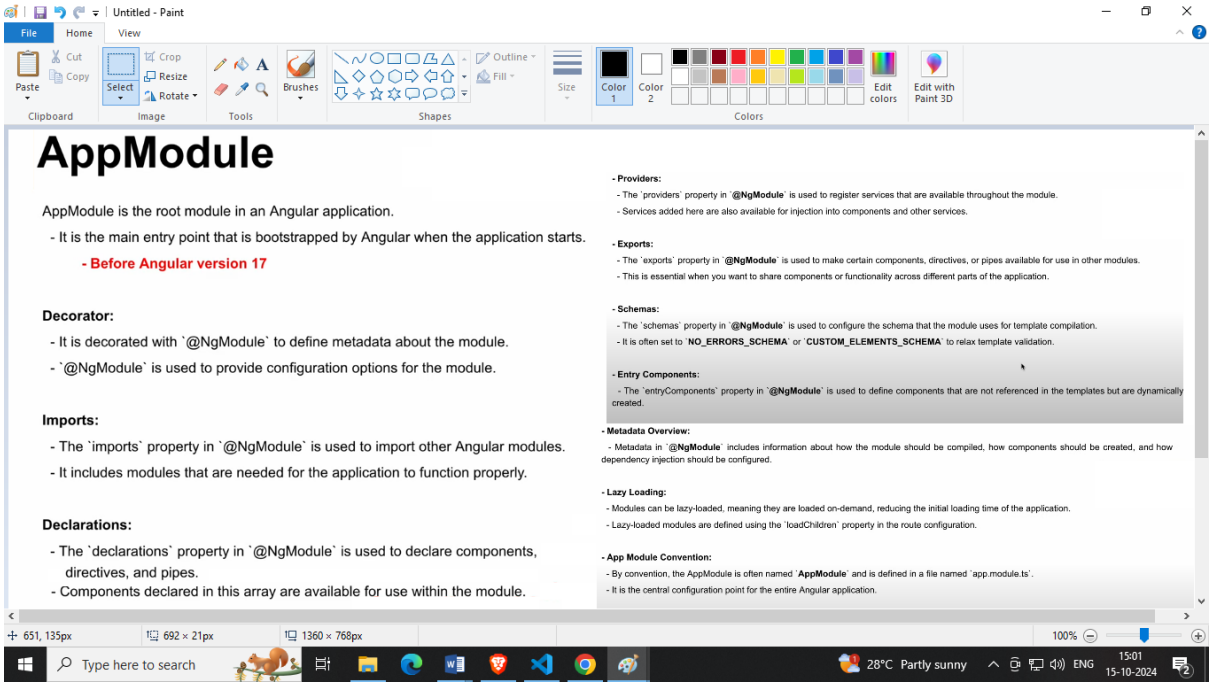

Untitled - Paint
Angular 17 vs Angular 16 and below
1. Angular 16 and below used AppModule
-> Its mandatory to have atleast one module
-> AppModule
-> main.ts -> Bootstrap AppModule
2. Angular 17 onwards
-> Everytihng is standalone
-> No mandatory Module is required
-> No AppModule anymore
-> BUT we can create our own custom modules
-> Is backward compatible
-> Everyting that you were dooing in Angulat 16 is still valid
-> -> main.ts -> Bootstrap AppComponent
ng g module <module name>
ang-16
ang-17
main.ts
Type here to search

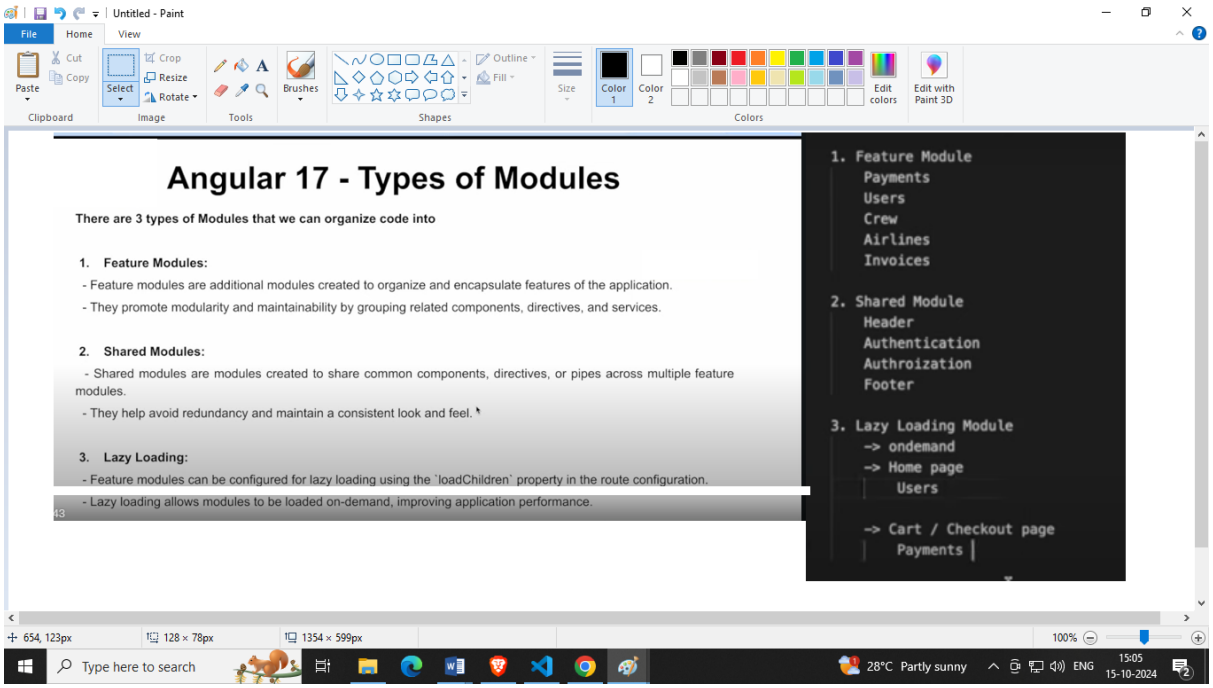

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
Angular 17 - Standalone Components
Standalone components in Angular are components that are independent and can be used in various parts of an application without being tied to a specific module or feature.
Here are a few key points about standalone components in Angular:
Standalone components are designed to encapsulate specific functionality or features, making them independent and self-contained.
The primary advantage of standalone components is their high reusability. They can be easily reused across different parts of an application or even in multiple projects.
Unlike components tied to a specific module, standalone components do not have dependencies on any particular module. This makes them versatile and adaptable to different contexts.
Standalone components can be integrated into various modules or feature areas effortlessly, allowing developers to mix and match components based on project requirements.
Since standalone components are independent, they are easier to maintain. Changes made to a standalone component won't have cascading effects on other parts of the application.
These components encapsulate both logic and user interface elements, promoting a modular and organized codes structure.
Standalone components are often easier to test because they focus on specific functionalities. Unit testing and integration testing become more straightforward, contributing to the overall maintainability of the codebase.
Standalone components can have scoped styles, ensuring that their styles don't interfere with the styles of other components in the application.
Standalone components often expose a clear API with well-defined inputs and outputs. This makes it easier for developers to understand how to use the component and what to expect in terms of behavior.
How to generate standalone components?
ng generate component component-name --flat
How to generate normal component - NOT standalone?
ng generate component --standalone false
Needs a module
How to migrate existing components?
Episode 14 -> Standalone Components
1. ng generate component <component_name>
ng g c <component_name> -> Will create a new folder
ng g c <component_name> --flat
2. ng g c <component_name> --standalone false
You will need a module mandatory
ng g module <module_name>
ng g c <component_name> --standalone false --module=<module_name>
1273, 25px
1354 × 599px
100%
Type here to search
28°C Partly sunny
ENG
14:03
15-10-2024

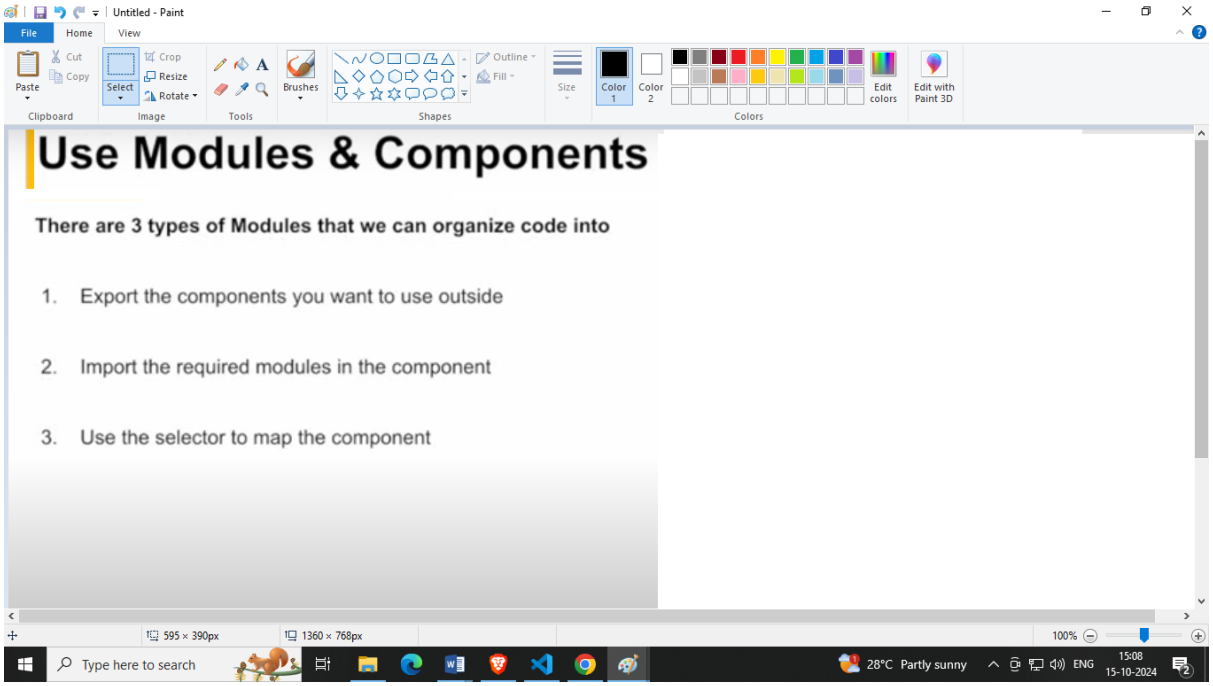

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
How to use Components?
Components can have parent/child hierarchy or just can be used independently.
We can use the component anywhere in the project
There is NO more dependency of modules in Angular 17 onwards [Refer episode 12 & 13]
Just use the selector name and use it as a tag name - that's it.
We can create parent/child relationship and use the components accordongly.
Episode 15 -> How to use components
1. ng g c <component_name>
2. by default each component will have a prefix -> app-<component_name>
3. unique selector name
4. 2 step process
4.1 Import the component into the required component
4.2 <component_name></component_name>
5. Cart
Products
Product1
Description
Images
Reviews
Rating
Product2
Payment
Netbanking
CreditCard
920, 192px
1360 × 768px
100%
Type here to search
28°C Partly sunny
ENG
14:18
15-10-2024

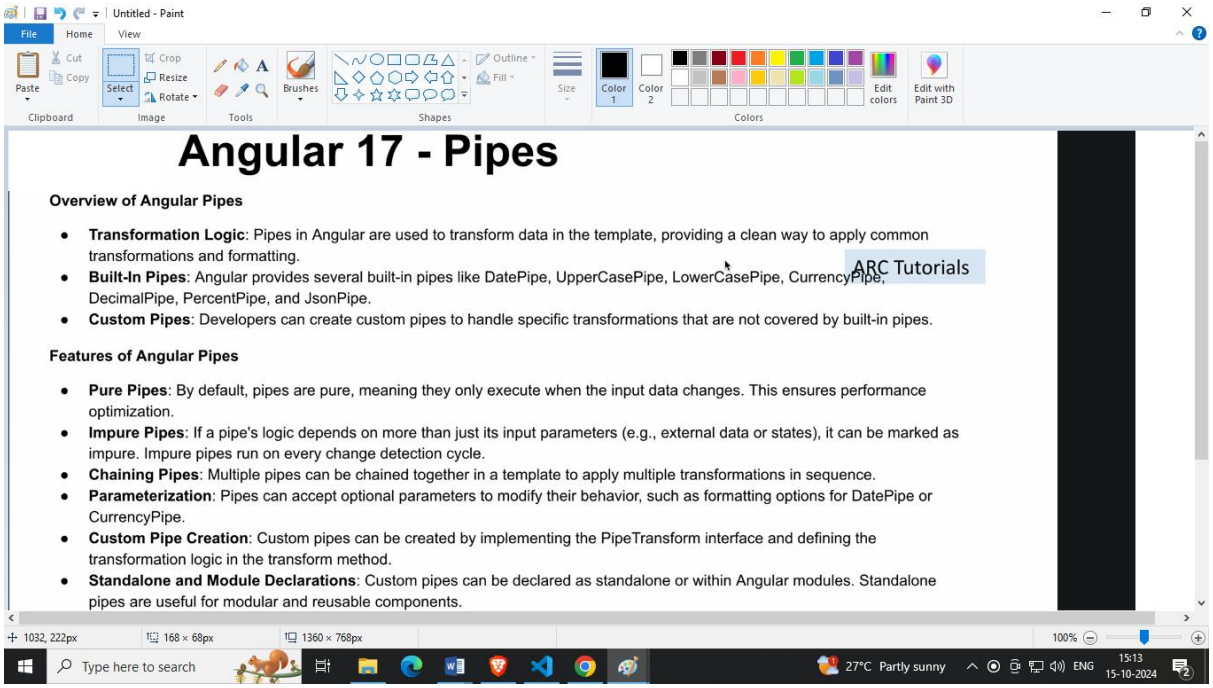

Untitled - Paint
File
Home
View
Component Communications
Component communication in Angular is crucial for building modular and scalable applications.
Two commonly used mechanisms for communication between components are @Input and @Output
@Input -> receiving inputs to the component
@Output -> Sending data from component to parent
parent-comp
export class CrewComponent {
messageForComponent: string = "From Parent";
userToken: string = "dfdgd4343434"
}
crew works!
<app-crew-desginations [message]="messageForComponent" [token]="userToken"></app-crew-desginations>
child-comp
export class CrewDesginationsComponent {
@Input() message: string = "";
@Input() token: string = "";
}
crew-desginations works!
 Inside Crew Designations
{{ message }}
{{ token }}
op:
crew works!
crew-desginations works!
Inside Crew Designations
From Parent
dfdgd4343434
722, 230px
1360 × 768px
100%
Type here to search
Live
ENG
14:30
15-10-2024

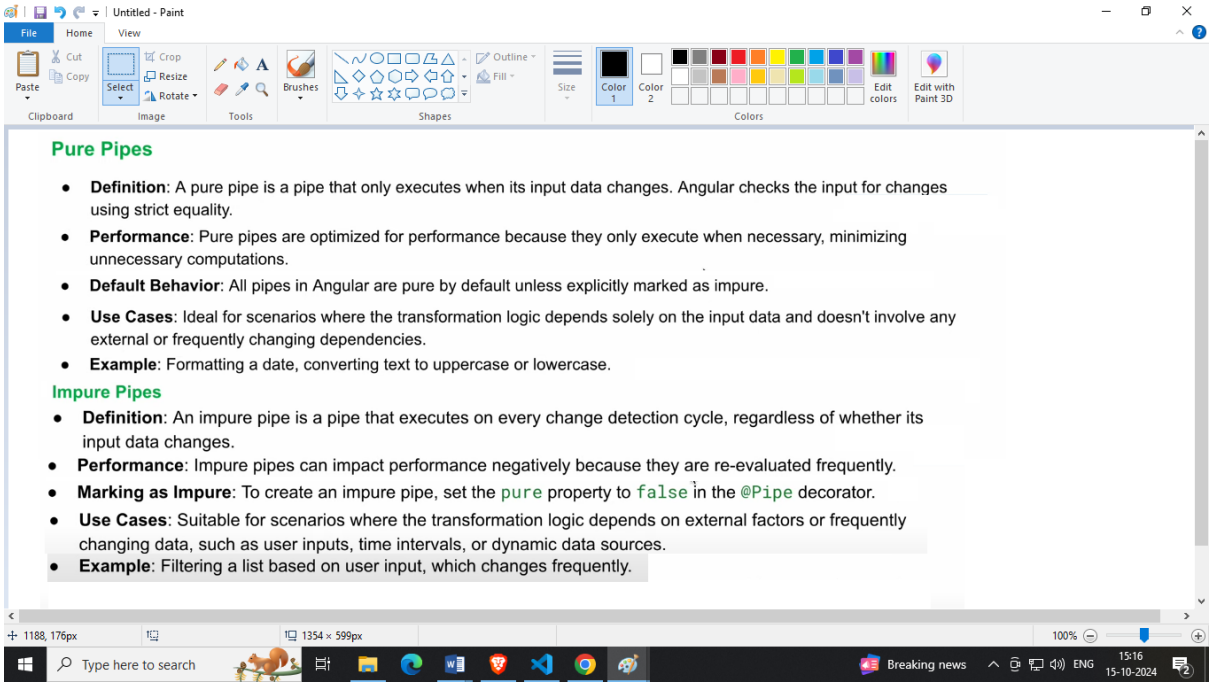

Untitled - Paint
File
Home
View
Episode #17 -> AppModule
1. By default in Angular 17 - there is no AppModule
2. Till Angular 16 -> it's mandatory to have AppModule
3. in Angular 17 supports modules ->
-> we can create our own modules
-> use just like how we were it earlier
4. in Angular 17 - we are bootstraping with AppComponent with Configurations
5. Till Angular 16 -> we are bootstraping with AppModule
- Metadata Overview:
- Metadata in `@NgModule` includes information about how the module should be compiled, how components should be created, and how dependency injection should be configured.
- Lazy Loading:
- Modules can be lazy-loaded, meaning they are loaded on-demand, reducing the initial loading time of the application.
- Lazy-loaded modules are defined using the `loadChildren` property in the route configuration.
- App Module Convention:
- By convention, the AppModule is often named `AppModule` and is defined in a file named `app.module.ts`.
- It is the central configuration point for the entire Angular application.
805, 165px
1354 × 599px
100%
Type here to search
28°C Partly sunny
ENG
15:01
15-10-2024

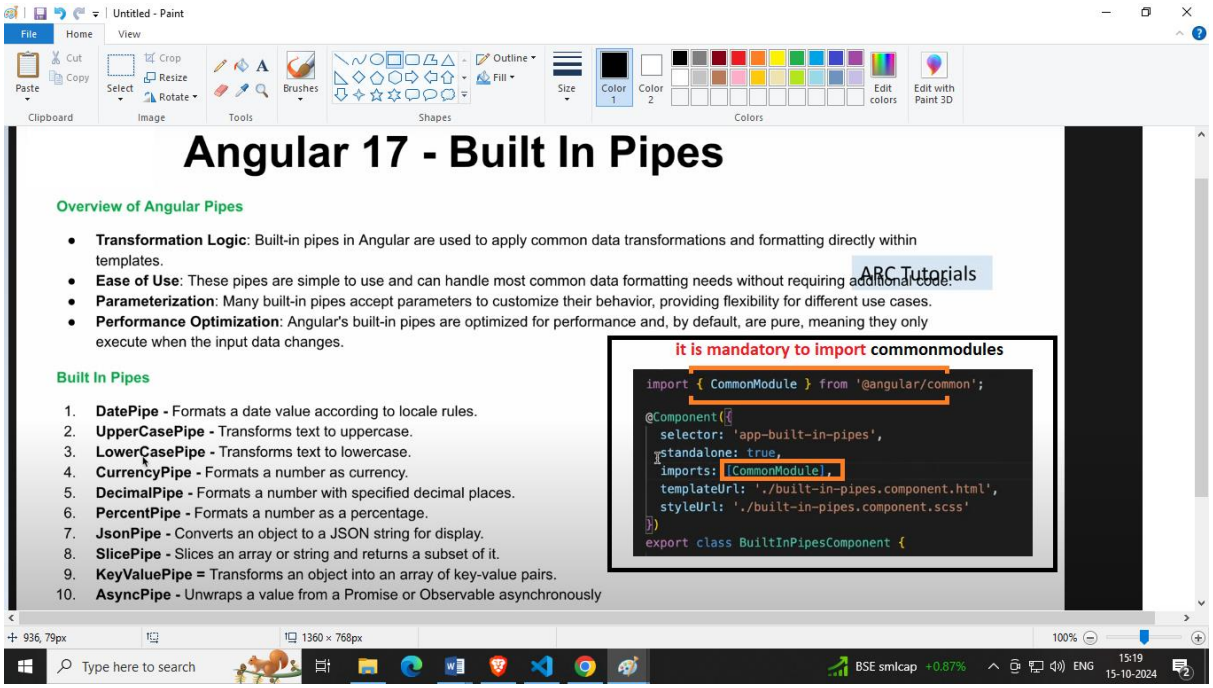

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
AppModule
AppModule is the root module in an Angular application.
- It is the main entry point that is bootstrapped by Angular when the application starts.
- Before Angular version 17
Decorator:
- It is decorated with `@NgModule` to define metadata about the module.
- `@NgModule` is used to provide configuration options for the module.
Imports:
- The `imports` property in `@NgModule` is used to import other Angular modules.
- It includes modules that are needed for the application to function properly.
Declarations:
- The `declarations` property in `@NgModule` is used to declare components, directives, and pipes.
- Components declared in this array are available for use within the module.
- Providers:
- The `providers` property in `@NgModule` is used to register services that are available throughout the module.
- Services added here are also available for injection into components and other services.
- Exports:
- The `exports` property in `@NgModule` is used to make certain components, directives, or pipes available for use in other modules.
- This is essential when you want to share components or functionality across different parts of the application.
- Schemas:
- The `schemas` property in `@NgModule` is used to configure the schema that the module uses for template compilation.
- It is often set to `NO_ERRORS_SCHEMA` or `CUSTOM_ELEMENTS_SCHEMA` to relax template validation.
- Entry Components:
- The `entryComponents` property in `@NgModule` is used to define components that are not referenced in the templates but are dynamically created.
- Metadata Overview:
- Metadata in `@NgModule` includes information about how the module should be compiled, how components should be created, and how dependency injection should be configured.
- Lazy Loading:
- Modules can be lazy-loaded, meaning they are loaded on-demand, reducing the initial loading time of the application.
- Lazy-loaded modules are defined using the `loadChildren` property in the route configuration.
- App Module Convention:
- By convention, the AppModule is often named `AppModule` and is defined in a file named `app.module.ts`.
- It is the central configuration point for the entire Angular application.
651, 135px
692 × 21px
1360 × 768px
100%
Type here to search
28°C Partly sunny
ENG
15:01
15-10-2024

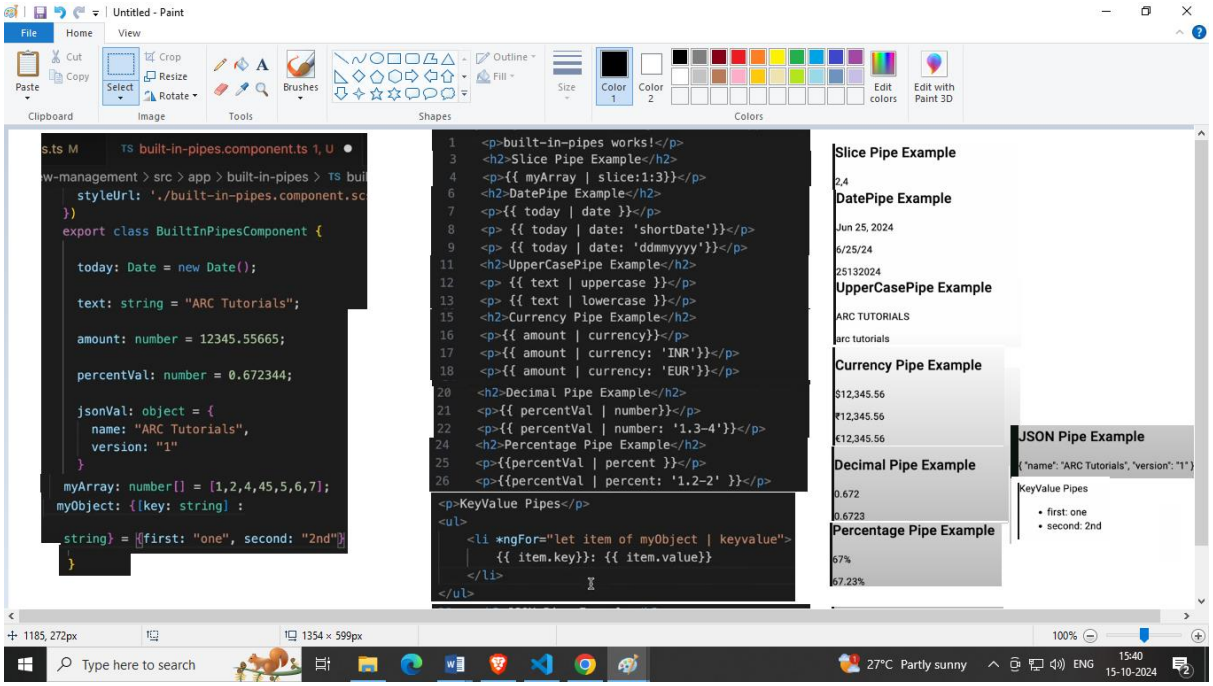

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
Angular 17 - Types of Modules
There are 3 types of Modules that we can organize code into
1. Feature Modules:
- Feature modules are additional modules created to organize and encapsulate features of the application.
- They promote modularity and maintainability by grouping related components, directives, and services.
2. Shared Modules:
- Shared modules are modules created to share common components, directives, or pipes across multiple feature modules.
- They help avoid redundancy and maintain a consistent look and feel.
3. Lazy Loading:
- Feature modules can be configured for lazy loading using the `loadChildren` property in the route configuration.
- Lazy loading allows modules to be loaded on-demand, improving application performance.
1. Feature Module
Payments
Users
Crew
Airlines
Invoices
2. Shared Module
Header
Authentication
Authroization
Footer
3. Lazy Loading Module
-> ondemand
-> Home page
Users
-> Cart / Checkout page
Payments
654, 123px
128 × 78px
1354 × 599px
100%
Type here to search
28°C Partly sunny
ENG
15:05
15-10-2024


Untitled - Paint
File
Home
View
Use Modules & Components
There are 3 types of Modules that we can organize code into
1. Export the components you want to use outside
2. Import the required modules in the component
3. Use the selector to map the component
595 × 390px
1360 × 768px
100%
Type here to search
28°C Partly sunny
ENG
15:08
15-10-2024


Untitled - Paint
File
Home
View
Angular 17 - Pipes
Overview of Angular Pipes
Transformation Logic: Pipes in Angular are used to transform data in the template, providing a clean way to apply common transformations and formatting.
Built-In Pipes: Angular provides several built-in pipes like DatePipe, UpperCasePipe, LowerCasePipe, CurrencyPipe, DecimalPipe, PercentPipe, and JsonPipe.
Custom Pipes: Developers can create custom pipes to handle specific transformations that are not covered by built-in pipes.
ARC Tutorials
Features of Angular Pipes
Pure Pipes: By default, pipes are pure, meaning they only execute when the input data changes. This ensures performance optimization.
Impure Pipes: If a pipe's logic depends on more than just its input parameters (e.g., external data or states), it can be marked as impure. Impure pipes run on every change detection cycle.
Chaining Pipes: Multiple pipes can be chained together in a template to apply multiple transformations in sequence.
Parameterization: Pipes can accept optional parameters to modify their behavior, such as formatting options for DatePipe or CurrencyPipe.
Custom Pipe Creation: Custom pipes can be created by implementing the PipeTransform interface and defining the transformation logic in the transform method.
Standalone and Module Declarations: Custom pipes can be declared as standalone or within Angular modules. Standalone pipes are useful for modular and reusable components.
1032, 222px
168 × 68px
1360 × 768px
100%
Type here to search
27°C Partly sunny
ENG
15:13
15-10-2024

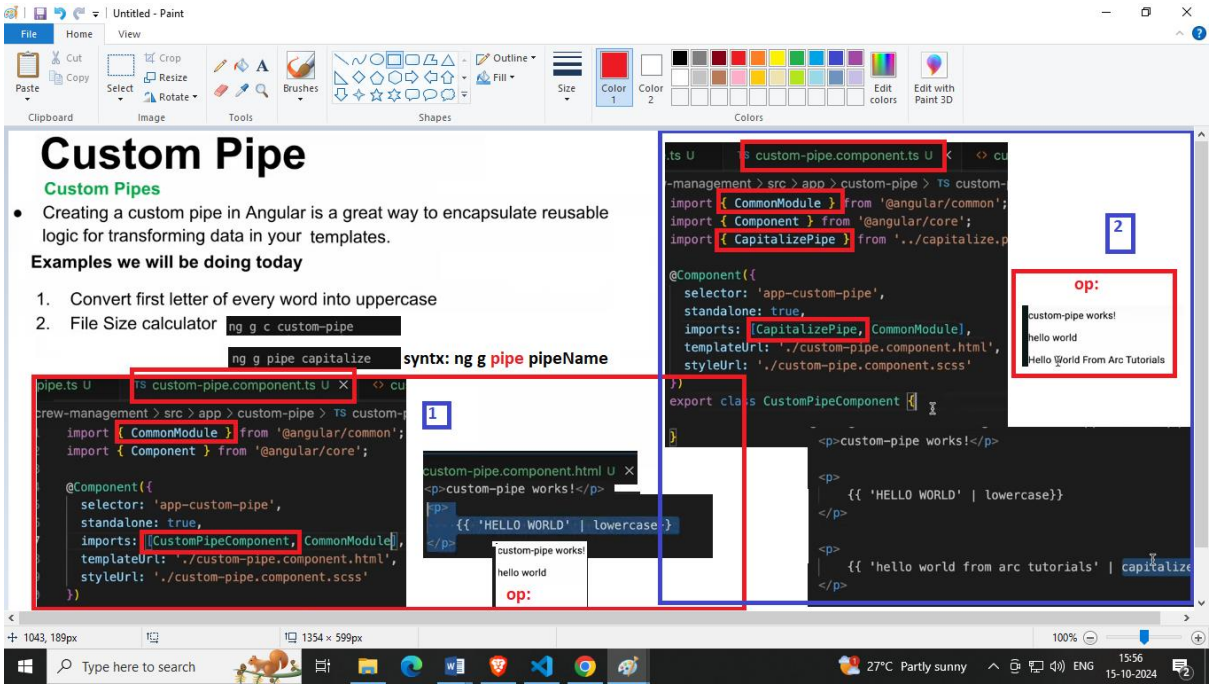

Untitled - Paint
Pure Pipes
Definition: A pure pipe is a pipe that only executes when its input data changes. Angular checks the input for changes using strict equality.
Performance: Pure pipes are optimized for performance because they only execute when necessary, minimizing unnecessary computations.
Default Behavior: All pipes in Angular are pure by default unless explicitly marked as impure.
Use Cases: Ideal for scenarios where the transformation logic depends solely on the input data and doesn't involve any external or frequently changing dependencies.
Example: Formatting a date, converting text to uppercase or lowercase.
Impure Pipes
Definition: An impure pipe is a pipe that executes on every change detection cycle, regardless of whether its input data changes.
Performance: Impure pipes can impact performance negatively because they are re-evaluated frequently.
Marking as Impure: To create an impure pipe, set the pure property to false in the @Pipe decorator.
Use Cases: Suitable for scenarios where the transformation logic depends on external factors or frequently changing data, such as user inputs, time intervals, or dynamic data sources.
Example: Filtering a list based on user input, which changes frequently.
Type here to search
Breaking news
ENG
15:16
15-10-2024


Untitled - Paint
Angular 17 - Built In Pipes
Overview of Angular Pipes
Transformation Logic: Built-in pipes in Angular are used to apply common data transformations and formatting directly within templates.
Ease of Use: These pipes are simple to use and can handle most common data formatting needs without requiring additional code.
Parameterization: Many built-in pipes accept parameters to customize their behavior, providing flexibility for different use cases.
Performance Optimization: Angular's built-in pipes are optimized for performance and, by default, are pure, meaning they only execute when the input data changes.
ARC Tutorials
it is mandatory to import commonmodules
import { CommonModule } from '@angular/common';
@Component({
selector: 'app-built-in-pipes',
standalone: true,
imports: [CommonModule],
templateUrl: './built-in-pipes.component.html',
styleUrl: './built-in-pipes.component.scss'
})
export class BuiltInPipesComponent {
Built In Pipes
DatePipe - Formats a date value according to locale rules.
UpperCasePipe - Transforms text to uppercase.
LowerCasePipe - Transforms text to lowercase.
CurrencyPipe - Formats a number as currency.
DecimalPipe - Formats a number with specified decimal places.
PercentPipe - Formats a number as a percentage.
JsonPipe - Converts an object to a JSON string for display.
SlicePipe - Slices an array or string and returns a subset of it.
KeyValuePipe = Transforms an object into an array of key-value pairs.
AsyncPipe - Unwraps a value from a Promise or Observable asynchronously
Type here to search
BSE smlcap +0.87%
ENG
15:19
15-10-2024


Untitled - Paint
built-in-pipes.component.ts
styleUrl: './built-in-pipes.component.sc
})
export class BuiltInPipesComponent {
today: Date = new Date();
text: string = "ARC Tutorials";
amount: number = 12345.55665;
percentVal: number = 0.672344;
jsonVal: object = {
name: "ARC Tutorials",
version: "1"
}
myArray: number[] = [1,2,4,45,5,6,7];
myObject: {[key: string] :
string} = {first: "one", second: "2nd"}
}
built-in-pipes works!
<h2>Slice Pipe Example</h2>
{{ myArray | slice:1:3}}
<h2>DatePipe Example</h2>
{{ today | date }}
 {{ today | date: 'shortDate'}}
 {{ today | date: 'ddmmyyyy'}}
<h2>UpperCasePipe Example</h2>
 {{ text | uppercase }}
 {{ text | lowercase }}
<h2>Currency Pipe Example</h2>
{{ amount | currency}}
{{ amount | currency: 'INR'}}
{{ amount | currency: 'EUR'}}
<h2>Decimal Pipe Example</h2>
{{ percentVal | number}}
{{ percentVal | number: '1.3-4'}}
<h2>Percentage Pipe Example</h2>
{{percentVal | percent }}
{{percentVal | percent: '1.2-2' }}
KeyValue Pipes

<li *ngFor="let item of myObject | keyvalue">
{{ item.key}}: {{ item.value}}

Slice Pipe Example
2,4
DatePipe Example
Jun 25, 2024
6/25/24
25132024
UpperCasePipe Example
ARC TUTORIALS
arc tutorials
Currency Pipe Example
$12,345.56
₹12,345.56
€12,345.56
Decimal Pipe Example
0.672
0.6723
Percentage Pipe Example
67%
67.23%
JSON Pipe Example
{"name": "ARC Tutorials", "version": "1"}
KeyValue Pipes
first: one
second: 2nd
Type here to search
27°C Partly sunny
ENG
15:40
15-10-2024


Untitled - Paint
Custom Pipe
Custom Pipes
Creating a custom pipe in Angular is a great way to encapsulate reusable logic for transforming data in your templates.
Examples we will be doing today
1. Convert first letter of every word into uppercase
2. File Size calculator
ng g c custom-pipe
ng g pipe capitalize
syntx: ng g pipe pipeName
1
custom-pipe.component.ts
import { CommonModule } from '@angular/common';
import { Component } from '@angular/core';
@Component({
selector: 'app-custom-pipe',
standalone: true,
imports: [CustomPipeComponent, CommonModule],
templateUrl: './custom-pipe.component.html',
styleUrl: './custom-pipe.component.scss'
})
custom-pipe.component.html
custom-pipe works!

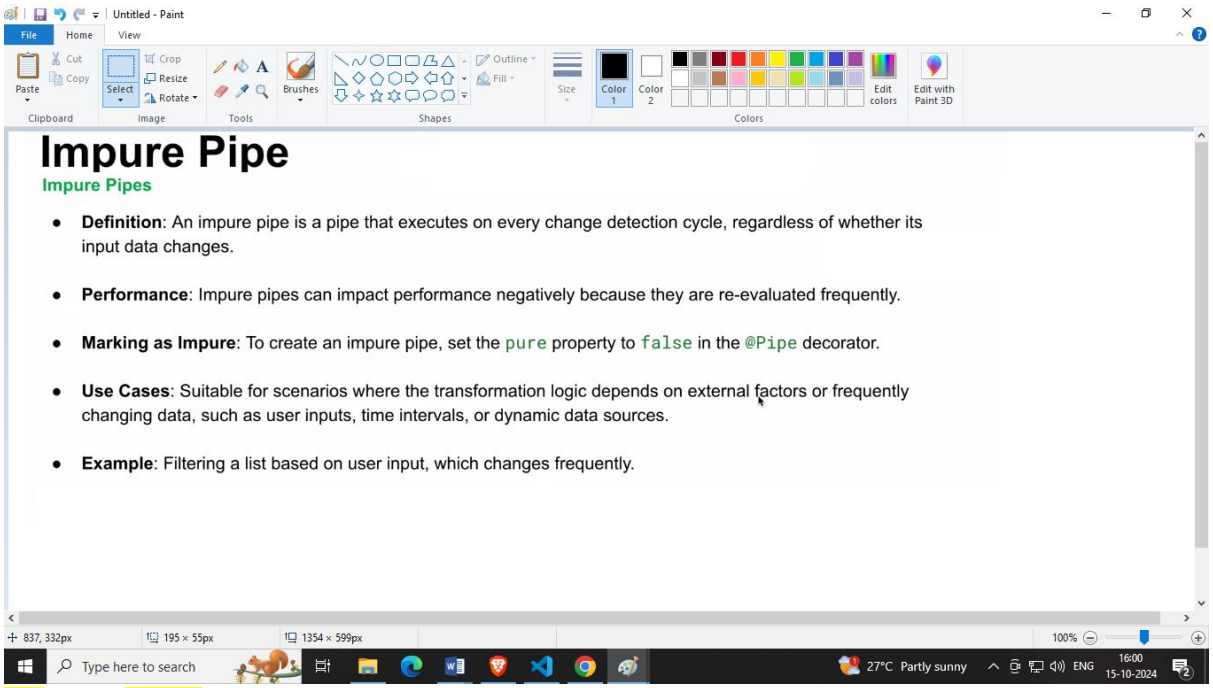
{{ 'HELLO WORLD' | lowercase }}

op:
custom-pipe works!
hello world
2
custom-pipe.component.ts
import { CommonModule } from '@angular/common';
import { Component } from '@angular/core';
import { CapitalizePipe } from '../capitalize.p
@Component({
selector: 'app-custom-pipe',
standalone: true,
imports: [CapitalizePipe, CommonModule],
templateUrl: './custom-pipe.component.html',
styleUrl: './custom-pipe.component.scss'
})
export class CustomPipeComponent {
}
custom-pipe works!

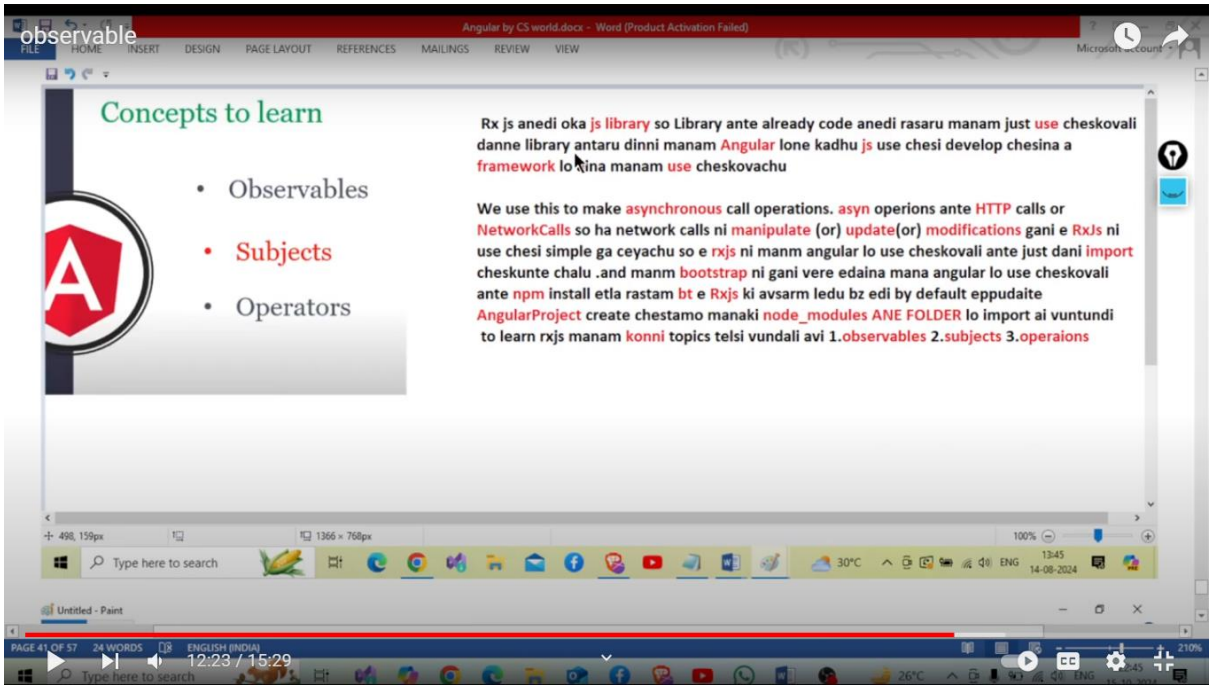
{{ 'HELLO WORLD' | lowercase}}

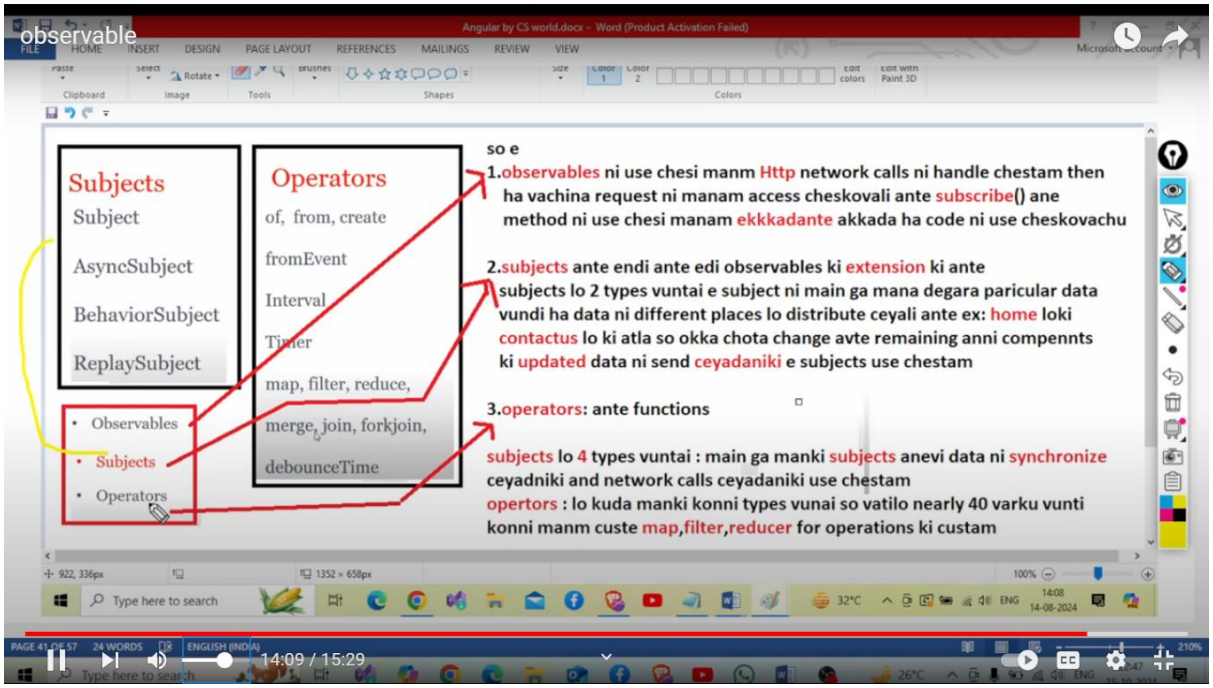
{{ 'hello world from arc tutorials' | capitalize

op:
custom-pipe works!
hello world
Hello World From Arc Tutorials
Type here to search
27°C Partly sunny
ENG
15:56
15-10-2024

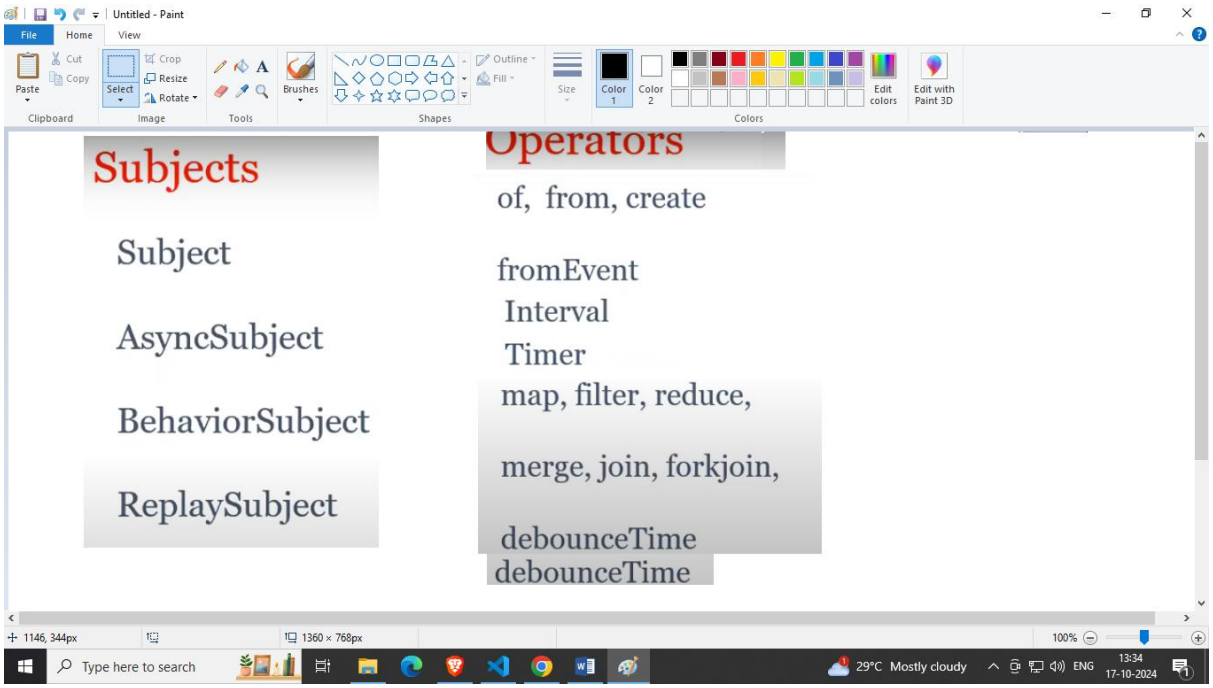


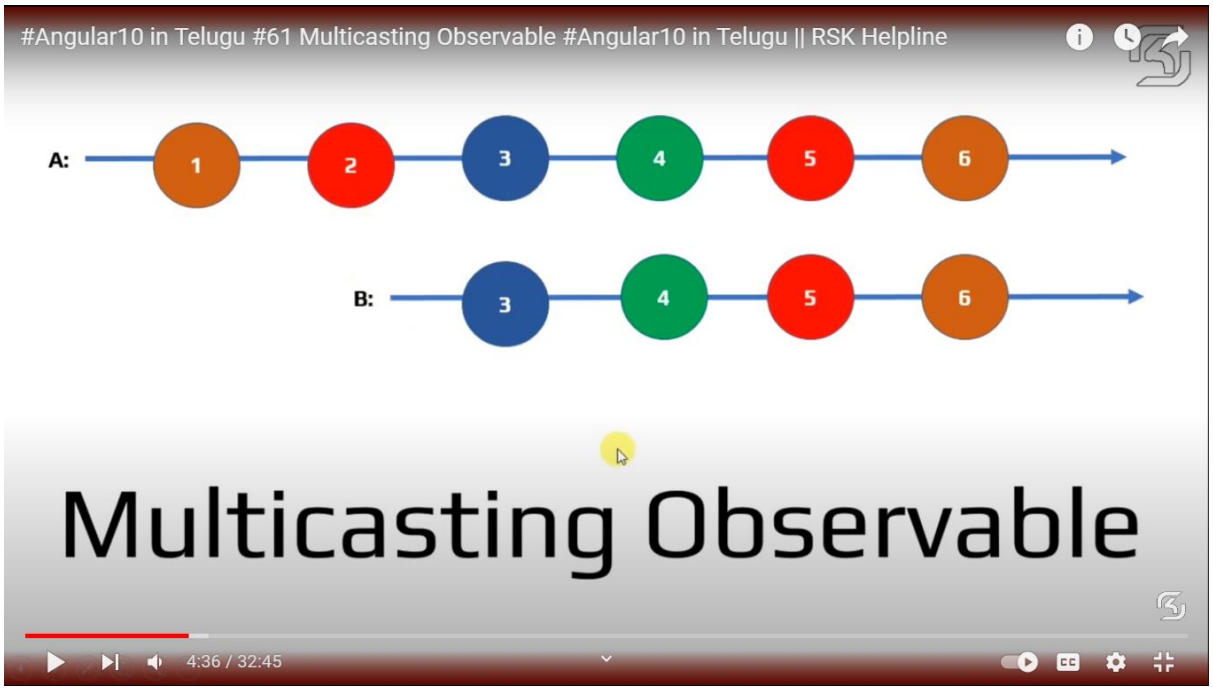
Css world telugu : observables subjects operators[in book about rxjs some clearly defined]

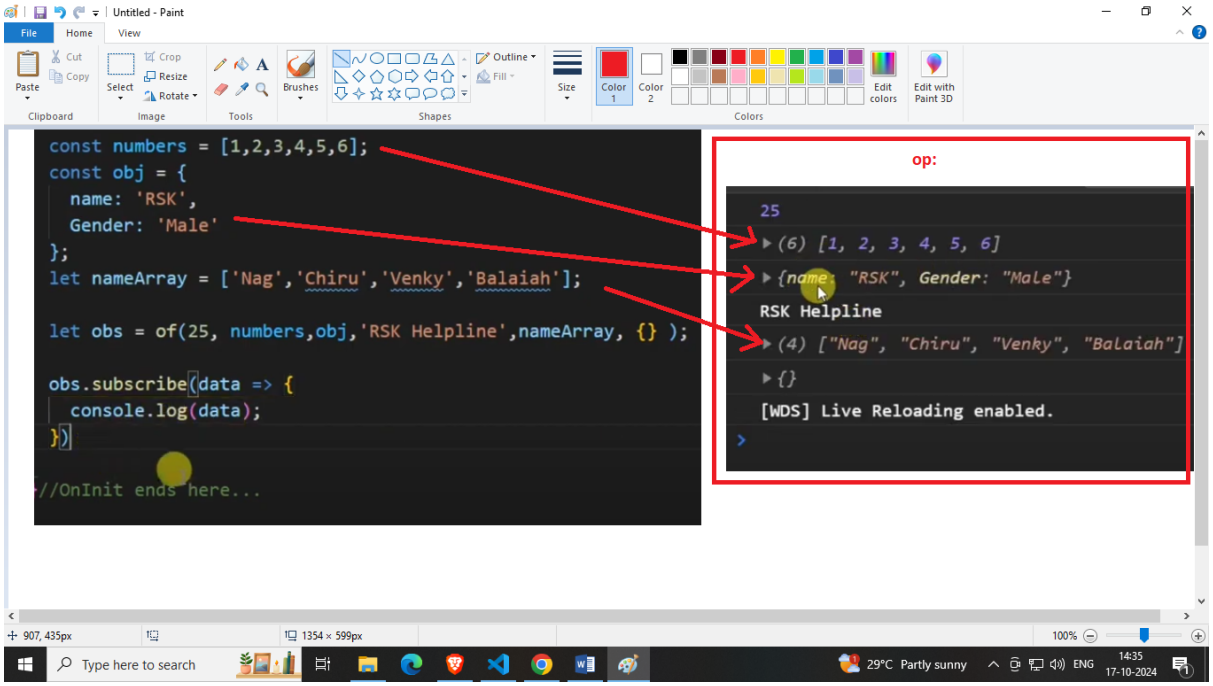


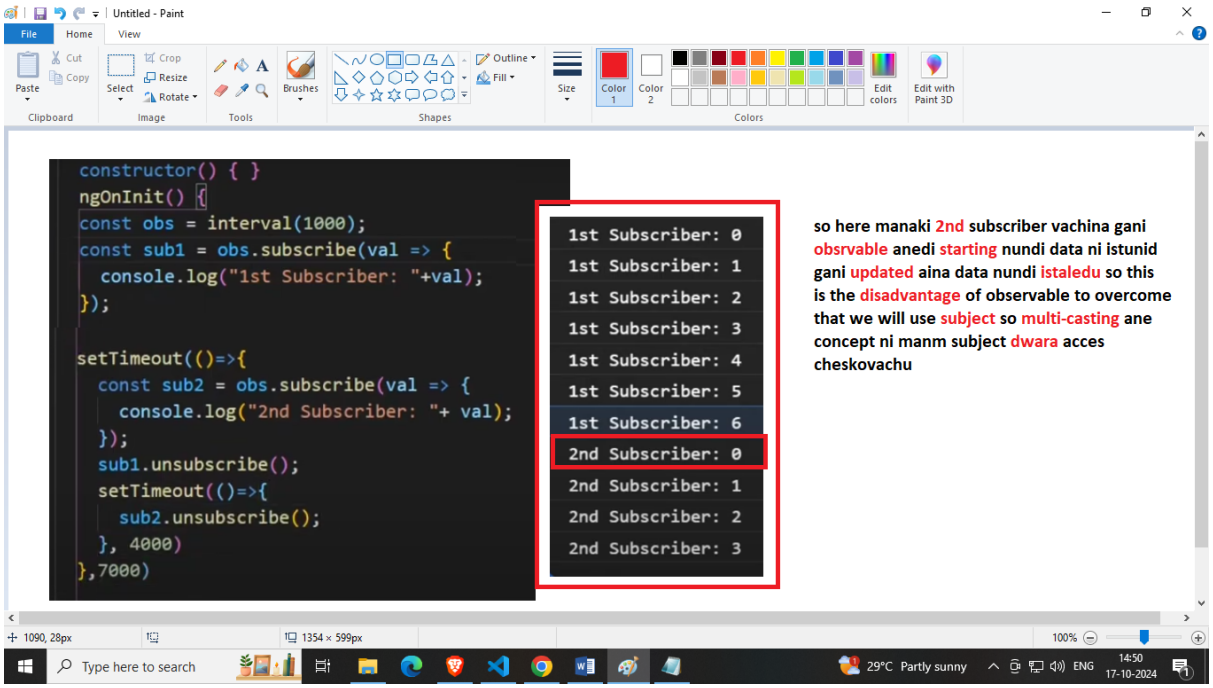

observable
Subjects
Subject
AsyncSubject
BehaviorSubject
ReplaySubject
Operators
of, from, create
fromEvent
Interval
Timer
map, filter, reduce,
merge, join, forkjoin,
debounceTime
Observables
Subjects
Operators
so e
1.observables ni use chesi manm Http network calls ni handle chestam then ha vachina request ni manam access cheskovali ante subscribe() ane method ni use chesi manam ekkkadante akkada ha code ni use cheskovachu
2.subjects ante endi ante edi observables ki extension ki ante subjects lo 2 types vuntai e subject ni main ga mana degara paricular data vundi ha data ni different places lo distribute ceyali ante ex: home loki contactus lo ki atla so okka chota change avte remaining anni compennts ki updated data ni send ceyadaniki e subjects use chestam
3.operators: ante functions
subjects lo 4 types vuntai : main ga manki subjects anevi data ni synchronize ceyadniki and network calls ceyadaniki use chestam
opertors : lo kuda manki konni types vunai so vatilo nearly 40 varku vunti konni manm custe map,filter,reducer for operations ki custam
14:09 / 15:29


rxjs library in angular | what is rxjs library | observables | subjects | rxjs operators | angular
Subjects
Subject
AsyncSubject
BehaviorSubject
ReplaySubject
7:43 / 10:31

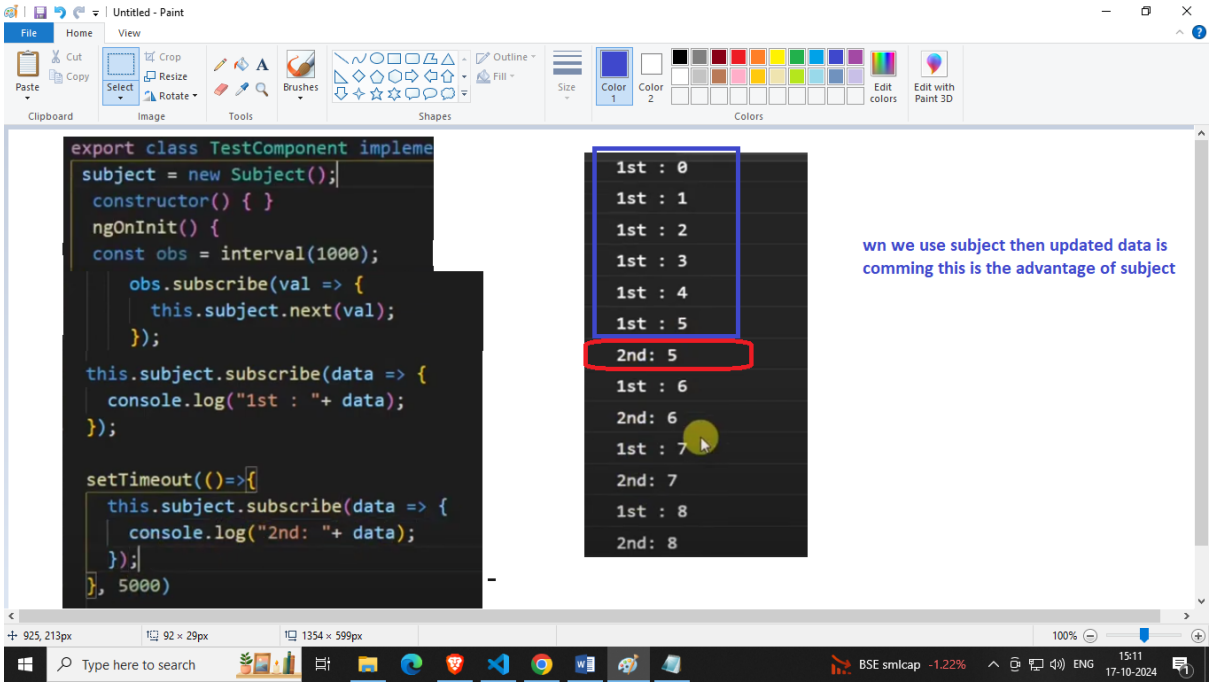

Untitled - Paint
File
Home
View
Subjects
Subject
AsyncSubject
BehaviorSubject
ReplaySubject
Operators
of, from, create
fromEvent
Interval
Timer
map, filter, reduce,
merge, join, forkjoin,
debounceTime
debounceTime
Type here to search
29°C Mostly cloudy
ENG
13:34
17-10-2024

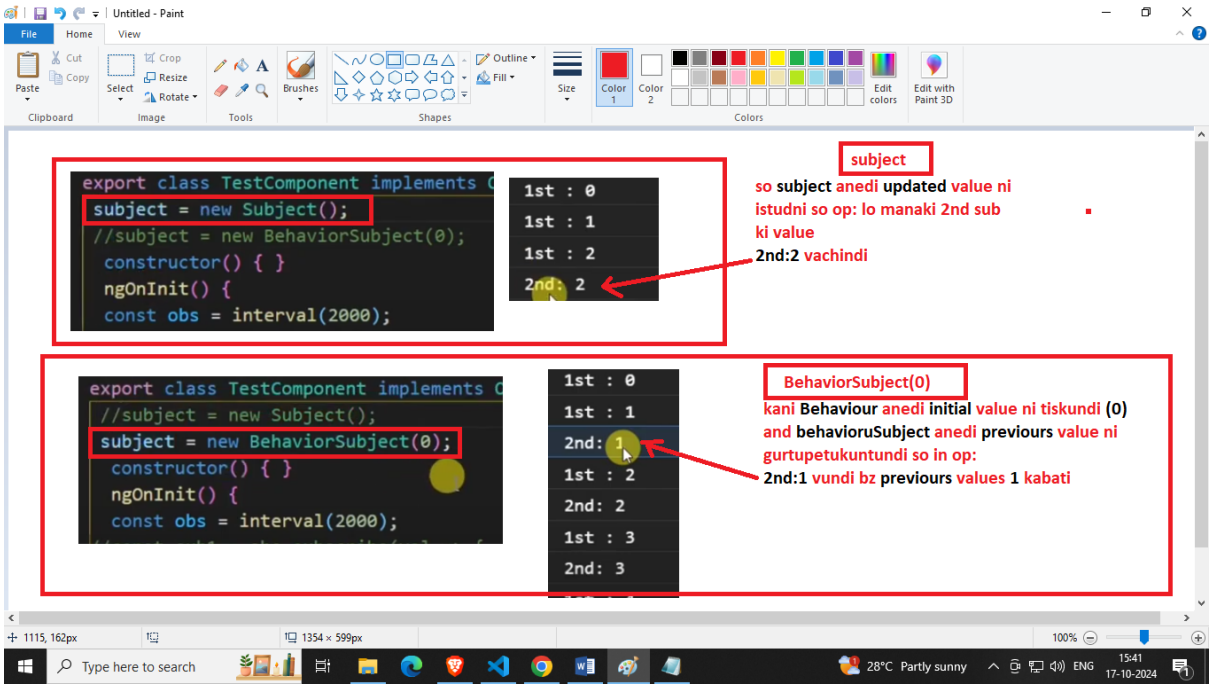

#Angular10 in Telugu #61 Multicasting Observable #Angular10 in Telugu || RSK Helpline
A:
1
2
3
4
5
6
B:
3
4
5
6
Multicasting Observable
4:36 / 32:45

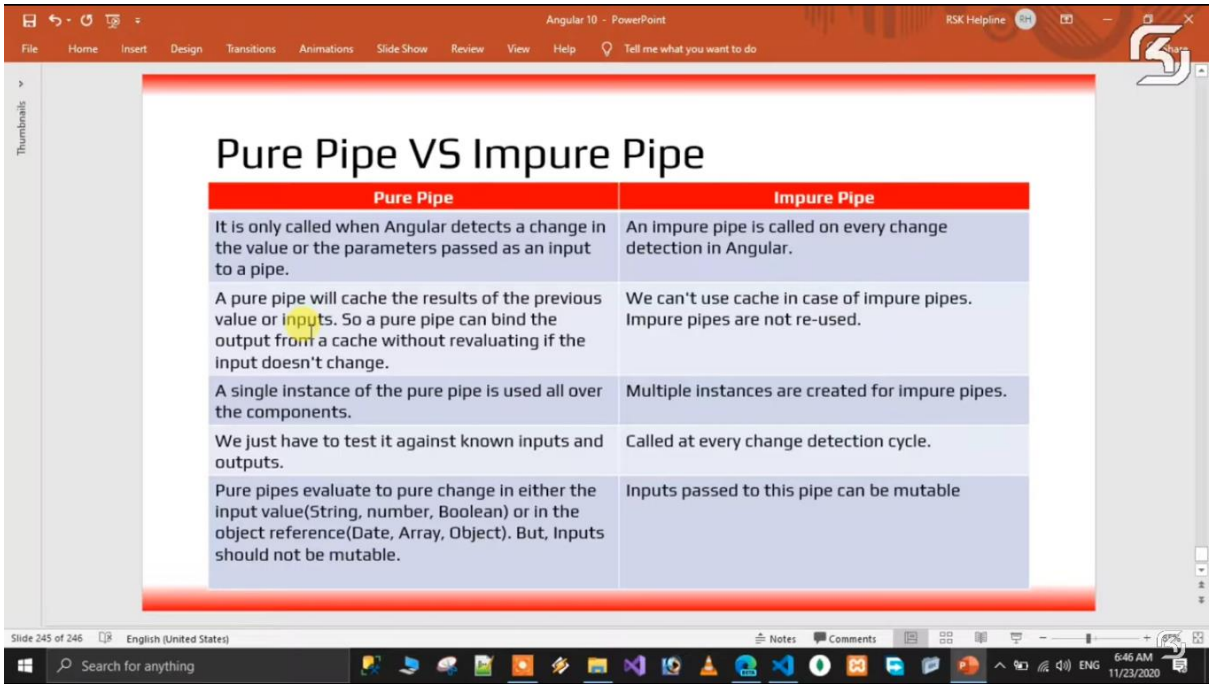


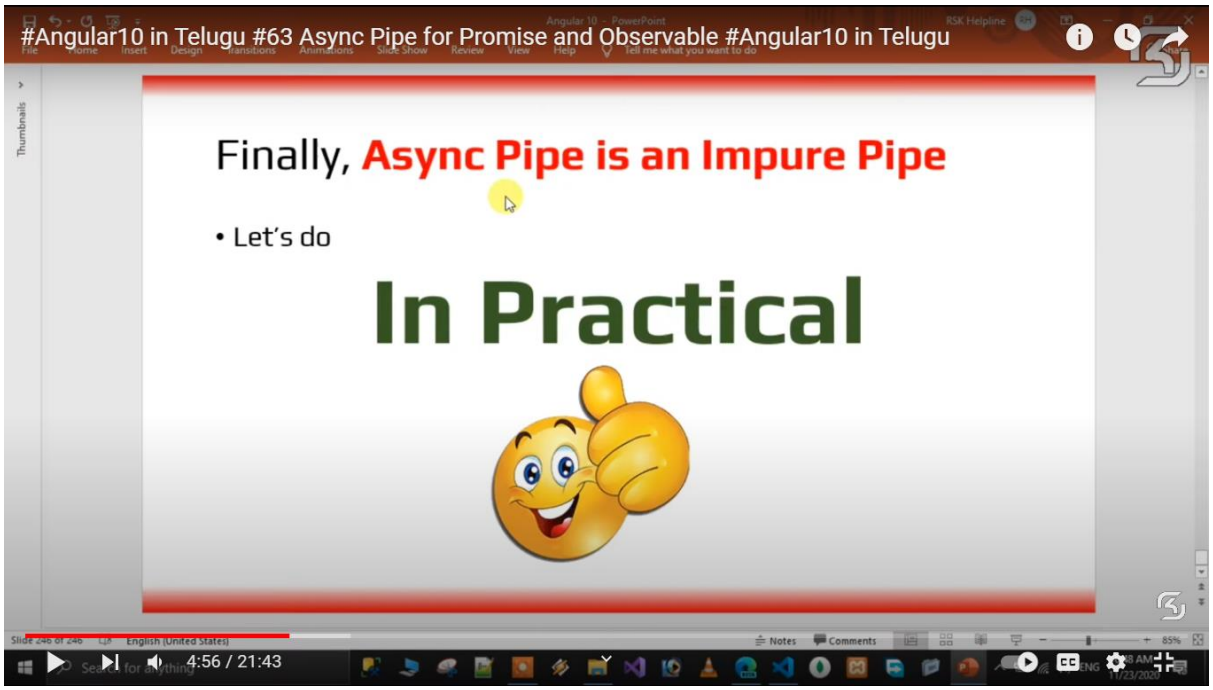


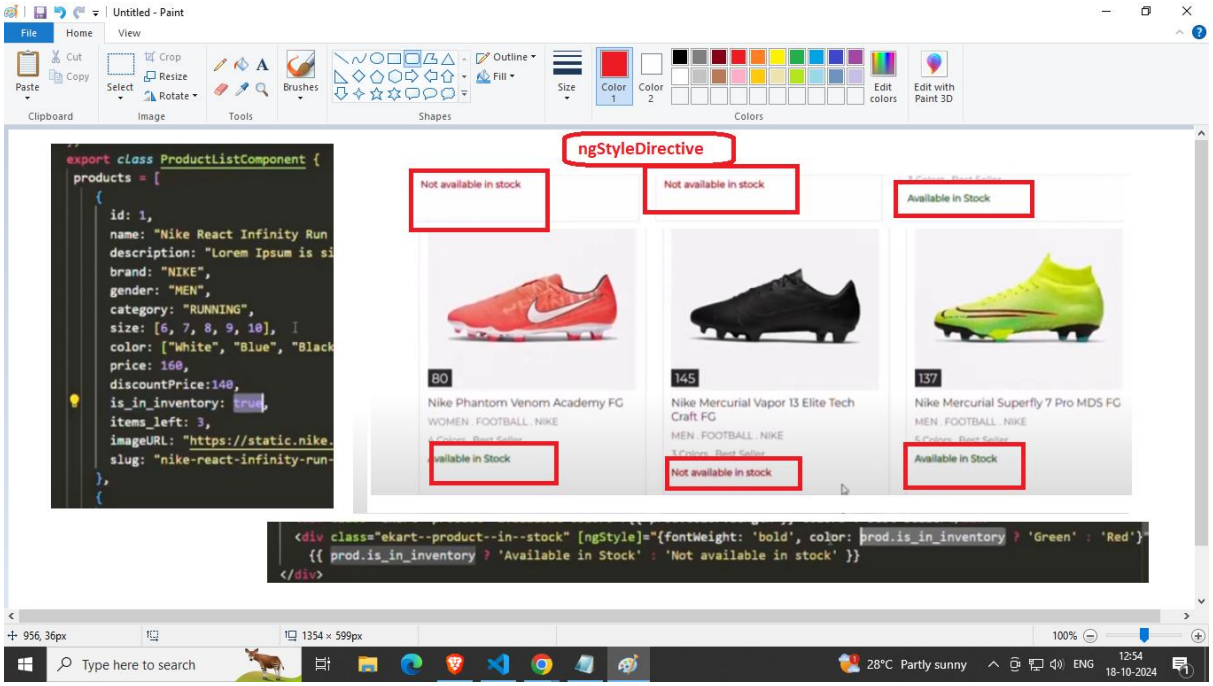
so here manaki 2nd subscriber vachina gani obsrvable anedi starting nundi data ni istunid gani updated aina data nundi istaledu so this is the disadvantage of observable to overcome that we will use subject so multi-casting ane concept ni manm subject dwara acces cheskovachu

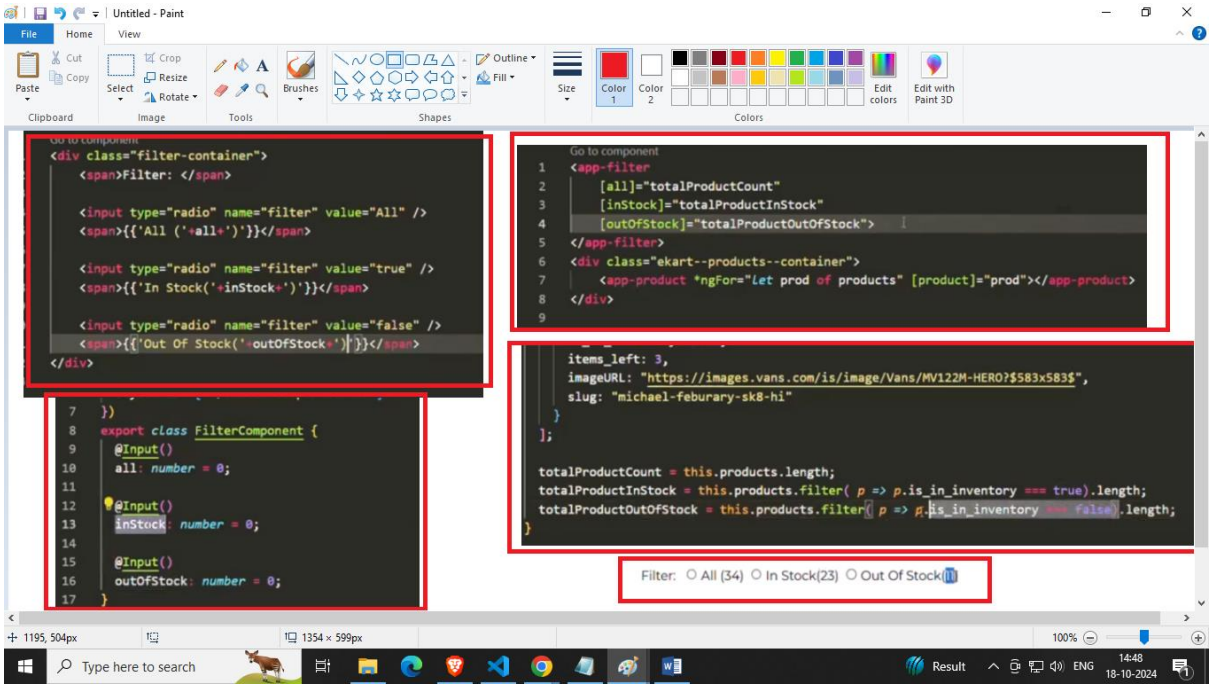

Untitled - Paint
export class TestComponent impleme
subject = new Subject();
constructor() { }
ngOnInit() {
const obs = interval(1000);
obs.subscribe(val => {
this.subject.next(val);
});
this.subject.subscribe(data => {
console.log("1st : "+ data);
});
setTimeout(()=>{
this.subject.subscribe(data => {
console.log("2nd: "+ data);
});
}, 5000)
1st : 0
1st : 1
1st : 2
1st : 3
1st : 4
1st : 5
2nd: 5
1st : 6
2nd: 6
1st : 7
2nd: 7
1st : 8
2nd: 8
wn we use subject then updated data is comming this is the advantage of subject
Type here to search
BSE smlcap -1.22%
ENG
15:11
17-10-2024


Untitled - Paint
export class TestComponent implements
subject = new Subject();
//subject = new BehaviorSubject(0);
constructor() { }
ngOnInit() {
const obs = interval(2000);
1st : 0
1st : 1
1st : 2
2nd: 2
subject
so subject anedi updated value ni istudni so op: lo manaki 2nd sub ki value
2nd:2 vachindi
export class TestComponent implements
//subject = new Subject();
subject = new BehaviorSubject(0);
constructor() { }
ngOnInit() {
const obs = interval(2000);
1st : 0
1st : 1
2nd: 1
1st : 2
2nd: 2
1st : 3
2nd: 3
BehaviorSubject(0)
kani Behaviour anedi initial value ni tiskundi (0) and behavioruSubject anedi previours value ni gurtupetukuntundi so in op:
2nd:1 vundi bz previours values 1 kabati
Type here to search
28°C Partly sunny
ENG
15:41
17-10-2024

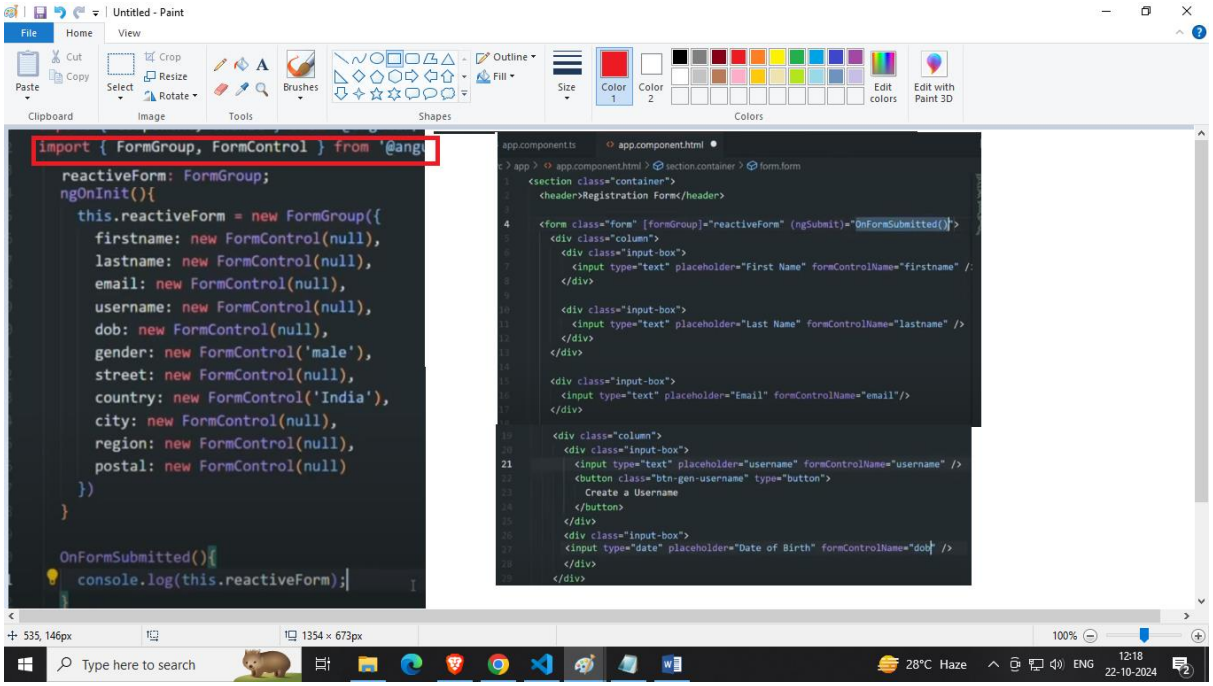

Angular 10 - PowerPoint
Pure Pipe VS Impure Pipe
Pure Pipe
Impure Pipe
It is only called when Angular detects a change in the value or the parameters passed as an input to a pipe.
An impure pipe is called on every change detection in Angular.
A pure pipe will cache the results of the previous value or inputs. So a pure pipe can bind the output from a cache without revaluating if the input doesn't change.
We can't use cache in case of impure pipes. Impure pipes are not re-used.
A single instance of the pure pipe is used all over the components.
Multiple instances are created for impure pipes.
We just have to test it against known inputs and outputs.
Called at every change detection cycle.
Pure pipes evaluate to pure change in either the input value(String, number, Boolean) or in the object reference(Date, Array, Object). But, Inputs should not be mutable.
Inputs passed to this pipe can be mutable

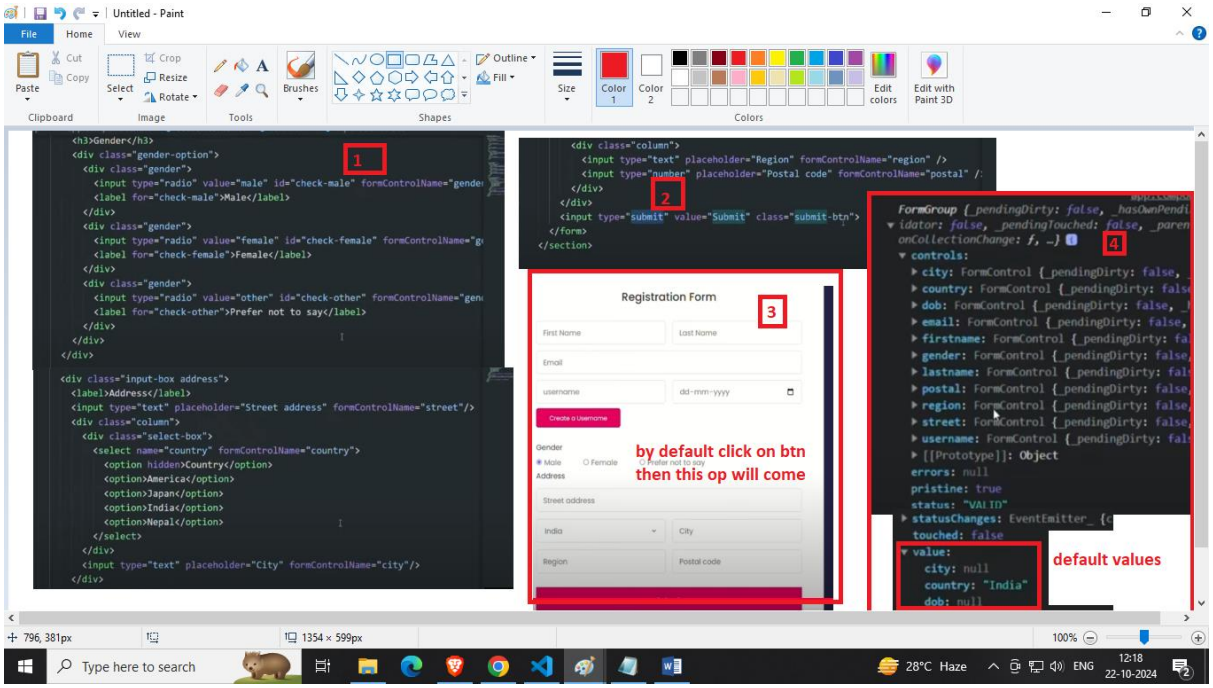

#Angular10 in Telugu #63 Async Pipe for Promise and Observable #Angular10 in Telugu
Finally, Async Pipe is an Impure Pipe
• Let's do
In Practical
4:56 / 21:43



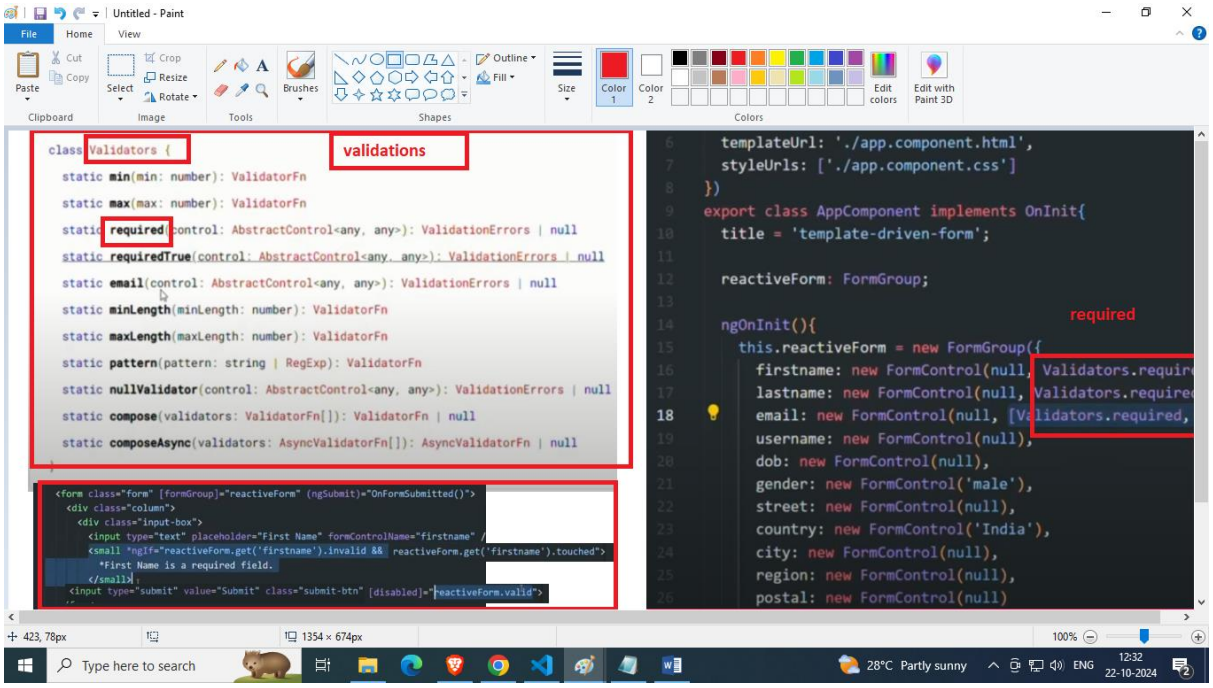


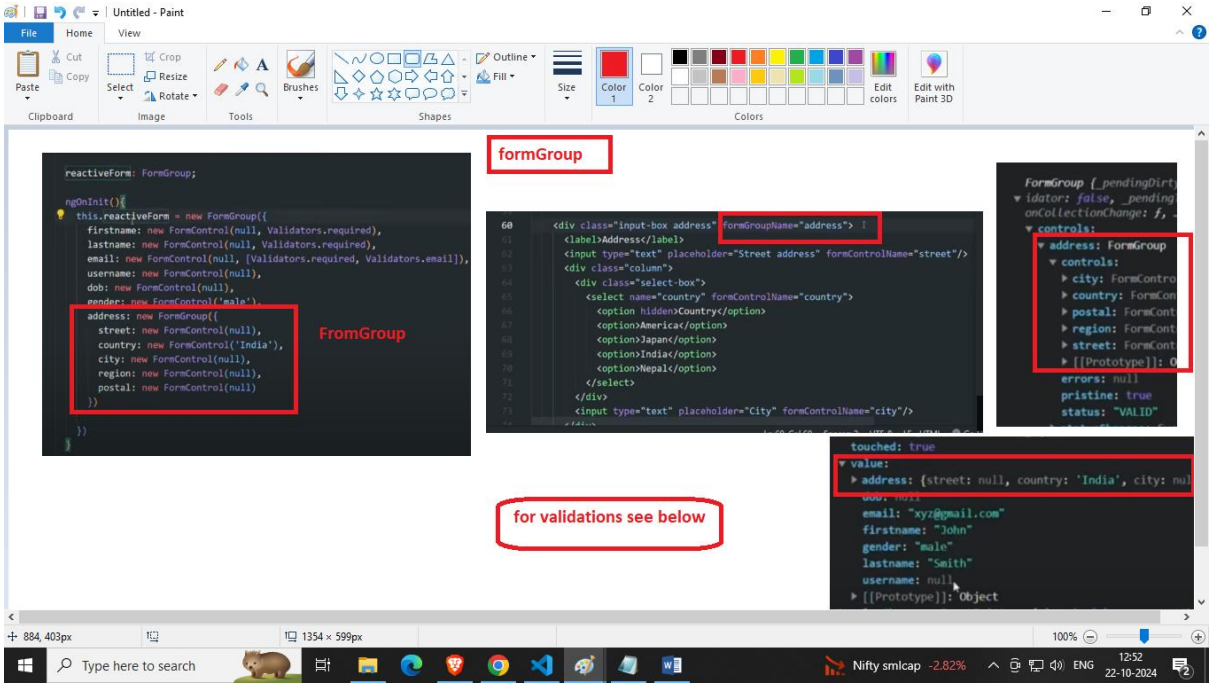
Angular forms by procode

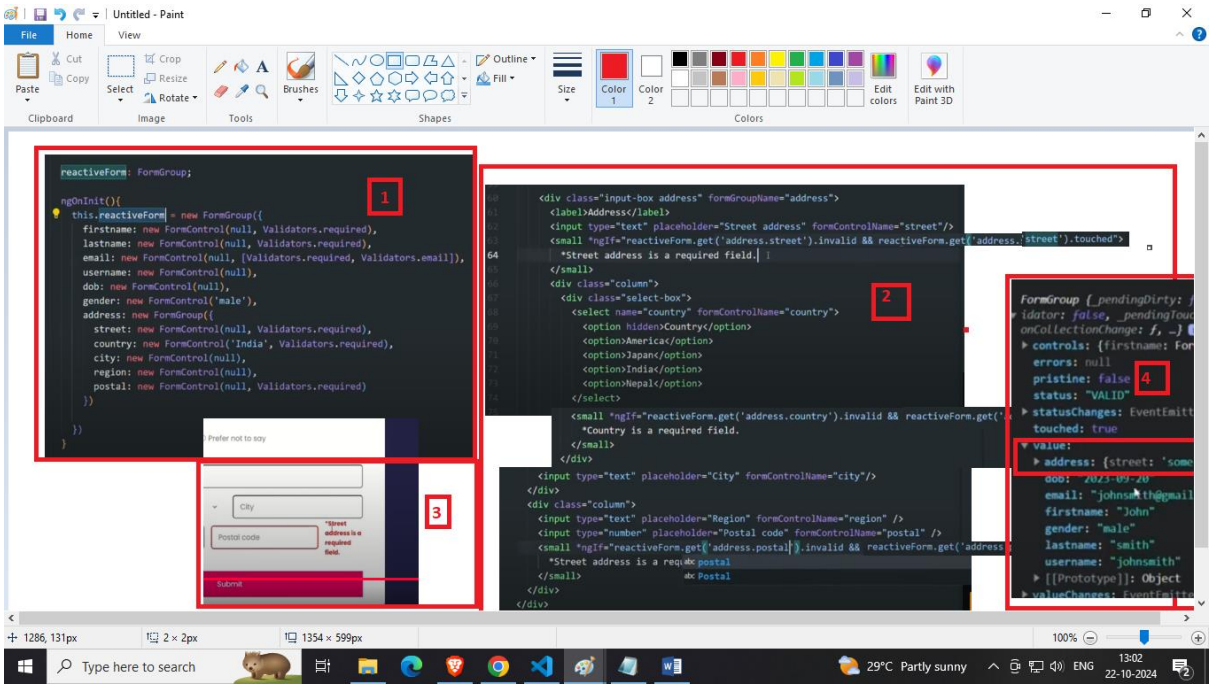

Untitled - Paint
import { FormGroup, FormControl } from '@angu
reactiveForm: FormGroup;
ngOnInit(){
this.reactiveForm = new FormGroup({
firstname: new FormControl(null),
lastname: new FormControl(null),
email: new FormControl(null),
username: new FormControl(null),
dob: new FormControl(null),
gender: new FormControl('male'),
street: new FormControl(null),
country: new FormControl('India'),
city: new FormControl(null),
region: new FormControl(null),
postal: new FormControl(null)
})
}
OnFormSubmitted(){
console.log(this.reactiveForm);
<section class="container">
<header>Registration Form</header>
<form class="form" [formGroup]="reactiveForm" (ngSubmit)="OnFormSubmitted()">
<input type="text" placeholder="First Name" formControlName="firstname" />
<input type="text" placeholder="Last Name" formControlName="lastname" />
<input type="text" placeholder="Email" formControlName="email"/>
<input type="text" placeholder="username" formControlName="username" />
<button class="btn-gen-username" type="button">
Create a Username
</button>
<input type="date" placeholder="Date of Birth" formControlName="dob" />
535, 146px
1354 × 673px
100%
Type here to search
28°C Haze
ENG
12:18
22-10-2024


Untitled - Paint
1
2
3
4
Registration Form
by default click on btn
then this op will come
default values
FormGroup
controls:
city: FormControl
country: FormControl
dob: FormControl
email: FormControl
firstname: FormControl
gender: FormControl
lastname: FormControl
postal: FormControl
region: FormControl
street: FormControl
username: FormControl
errors: null
pristine: true
status: "VALID"
touched: false
value:
city: null
country: "India"
dob: null
796, 381px
1354 × 599px
100%
Type here to search
28°C Haze
ENG
12:18
22-10-2024

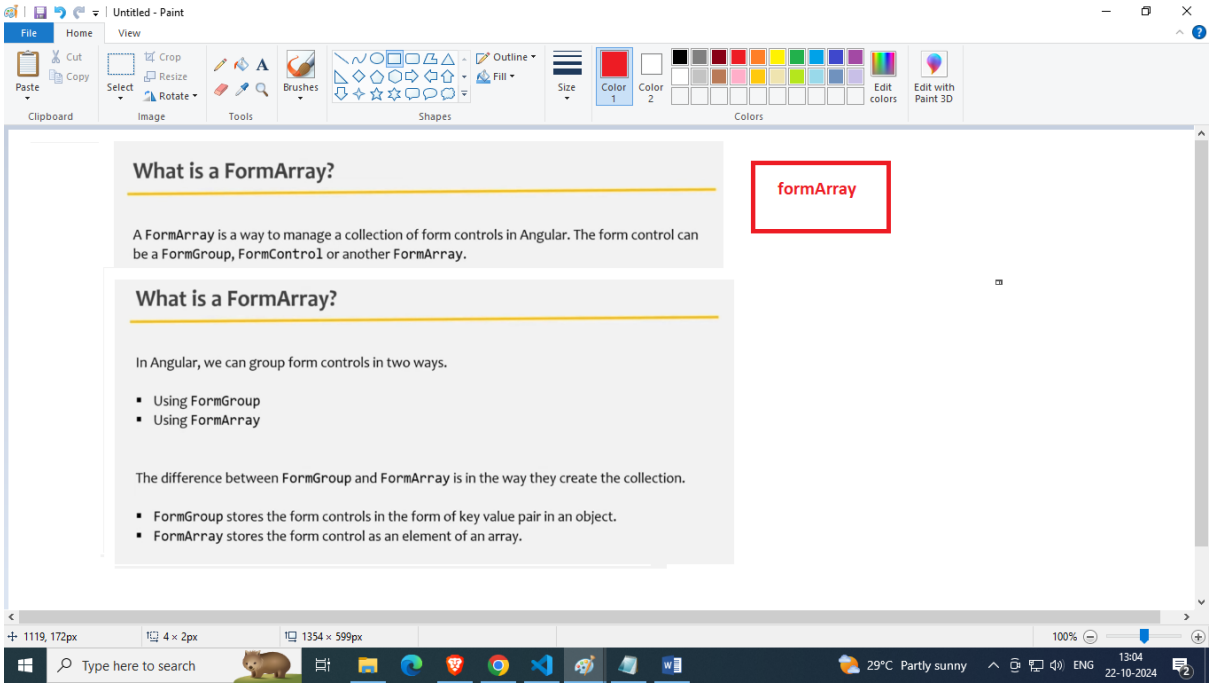
validators


Untitled - Paint
validations
required


Untitled - Paint
formGroup
FromGroup
for validations see below

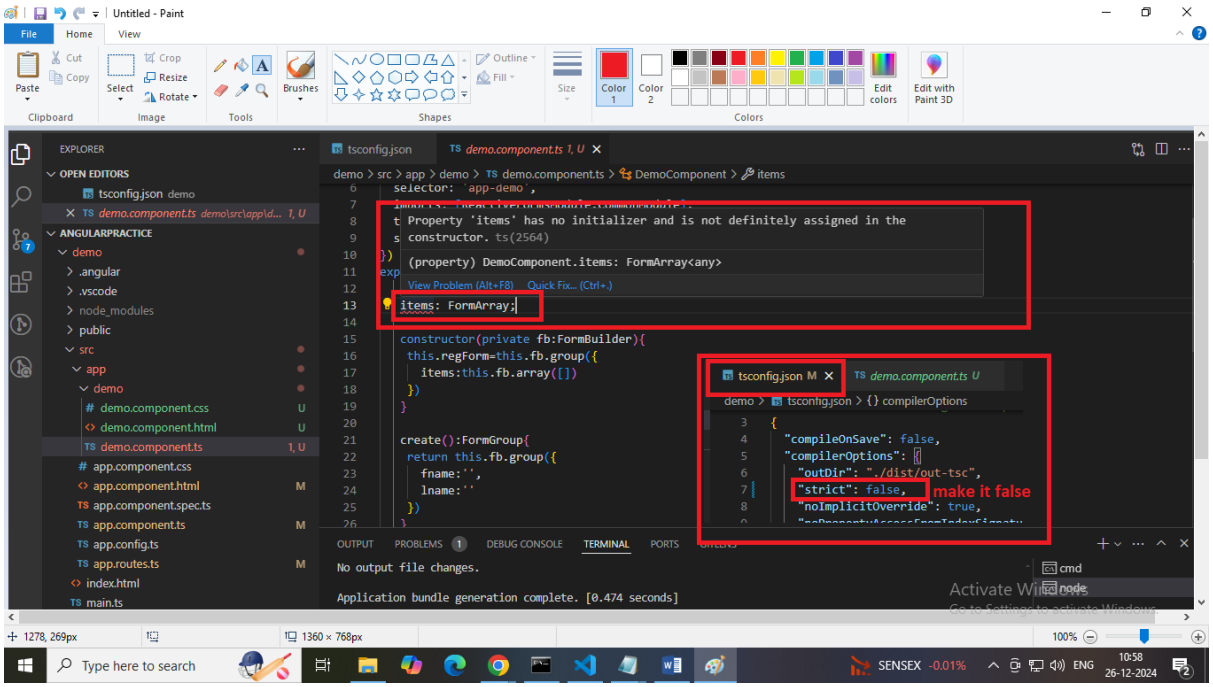

Untitled - Paint
reactiveForm: FormGroup;
ngOnInit(){
this.reactiveForm = new FormGroup({
firstname: new FormControl(null, Validators.required),
lastname: new FormControl(null, Validators.required),
email: new FormControl(null, [Validators.required, Validators.email]),
username: new FormControl(null),
dob: new FormControl(null),
gender: new FormControl('male'),
address: new FormGroup({
street: new FormControl(null, Validators.required),
country: new FormControl('India', Validators.required),
city: new FormControl(null),
region: new FormControl(null),
postal: new FormControl(null, Validators.required)
})
})
}
1
2
3
4
Prefer not to say
City
Postal code
Submit
*Street address is a required field.
*Country is a required field.
Postal
Type here to search
29°C Partly sunny
ENG
13:02
22-10-2024

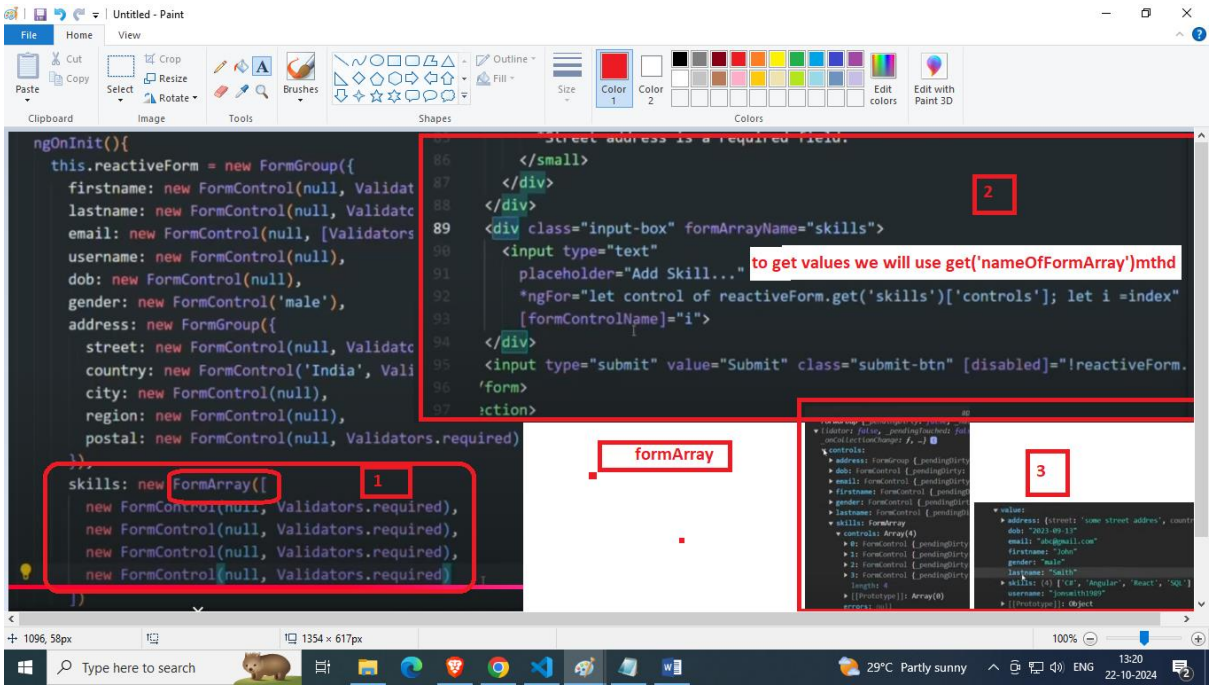
formArray

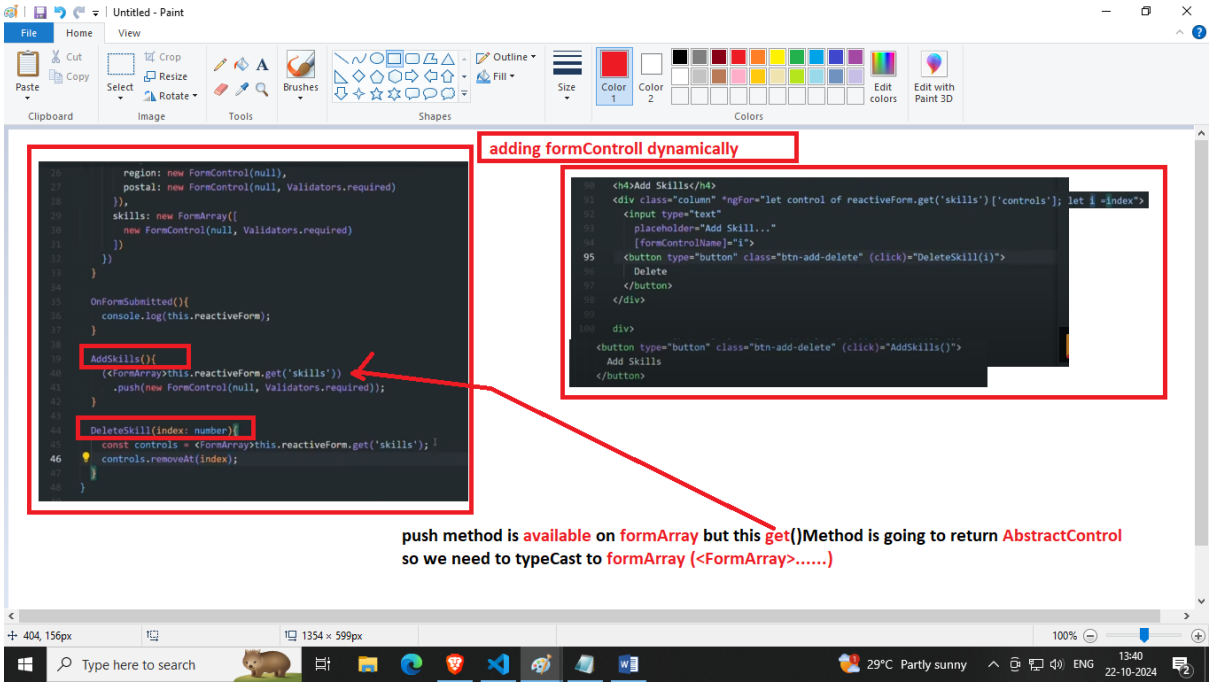

Untitled - Paint
What is a FormArray?
A FormArray is a way to manage a collection of form controls in Angular. The form control can be a FormGroup, FormControl or another FormArray.
formArray
What is a FormArray?
In Angular, we can group form controls in two ways.
Using FormGroup
Using FormArray
The difference between FormGroup and FormArray is in the way they create the collection.
FormGroup stores the form controls in the form of key value pair in an object.
FormArray stores the form control as an element of an array.
Type here to search
29°C Partly sunny
ENG
13:04
22-10-2024

Property has no initializer : if u get this error while using FormArray then follow below dig..

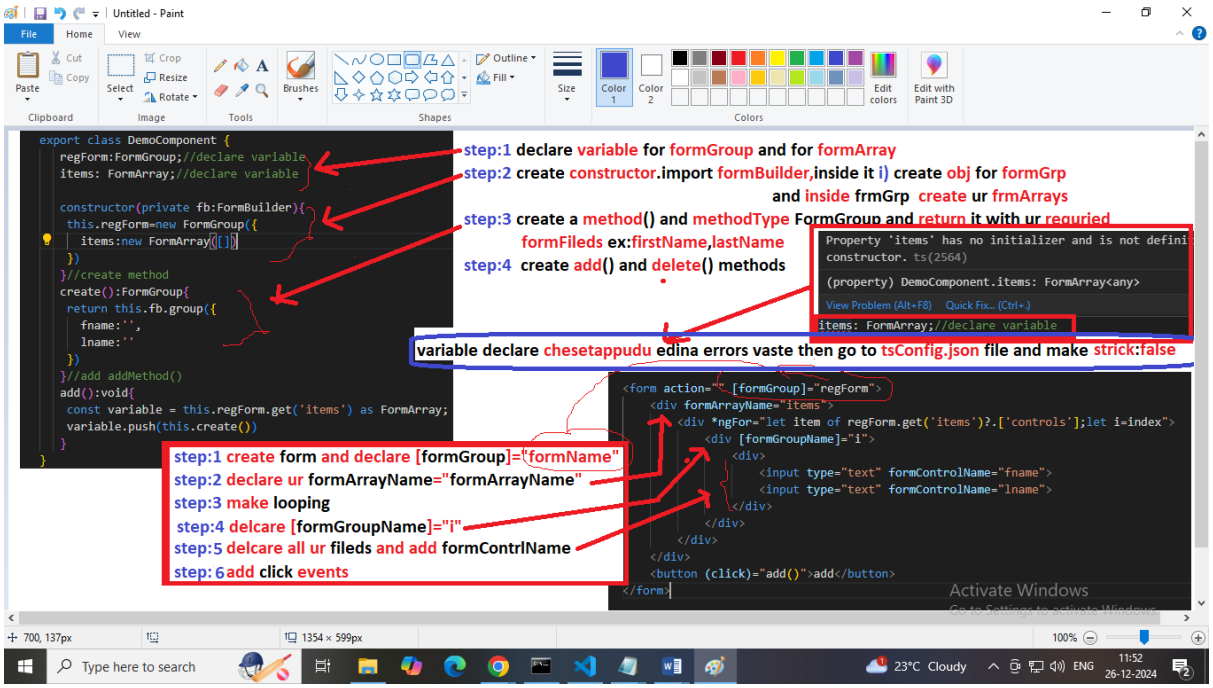
➔ Solution:go to tsconfi.json file and make strick:false






Untitled - Paint
adding formControll dynamically
region: new FormControl(null),
postal: new FormControl(null, Validators.required)
}),
skills: new FormArray([
new FormControl(null, Validators.required)
])
})
}
OnFormSubmitted(){
console.log(this.reactiveForm);
}
AddSkills(){
(<FormArray>this.reactiveForm.get('skills'))
.push(new FormControl(null, Validators.required));
}
DeleteSkill(index: number){
const controls = <FormArray>this.reactiveForm.get('skills');
controls.removeAt(index);
}
<h4>Add Skills</h4>
<div class="column" *ngFor="let control of reactiveForm.get('skills')['controls']; let i =index">
<input type="text"
placeholder="Add Skill..."
[formControlName]="i">
<button type="button" class="btn-add-delete" (click)="DeleteSkill(i)">
Delete
</button>

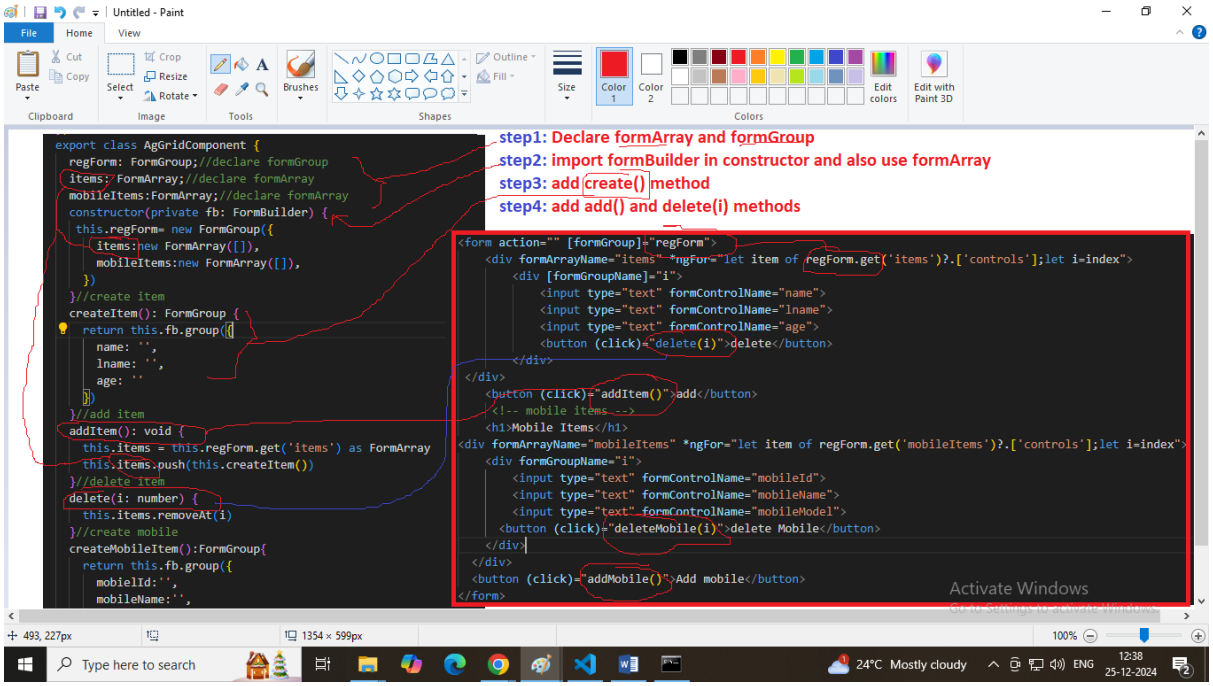
div>
<button type="button" class="btn-add-delete" (click)="AddSkills()">
Add Skills
</button>
push method is available on formArray but this get()Method is going to return AbstractControl
so we need to typeCast to formArray (<FormArray>.....)
404, 156px
1354 × 599px
100%
Type here to search
29°C Partly sunny
ENG
13:40
22-10-2024


Untitled - Paint
export class DemoComponent {
regForm:FormGroup;//declare variable
items: FormArray;//declare variable
constructor(private fb:FormBuilder){
this.regForm=new FormGroup({
items:new FormArray([])
})
}//create method
create():FormGroup{
return this.fb.group({
fname:'',
lname:''
})
}//add addMethod()
add():void{
const variable = this.regForm.get('items') as FormArray;
variable.push(this.create())
}
step:1 declare variable for formGroup and for formArray
step:2 create constructor.import formBuilder,inside it i) create obj for formGrp
and inside frmGrp create ur frmArrays
step:3 create a method() and methodType FormGroup and return it with ur requried
formFileds ex:firstName,lastName
step:4 create add() and delete() methods
Property 'items' has no initializer and is not defini
constructor. ts(2564)
(property) DemoComponent.items: FormArray<any>
View Problem (Alt+F8) Quick Fix... (Ctrl+.)
items: FormArray;//declare variable
variable declare chesetappudu edina errors vaste then go to tsConfig.json file and make strick:false
<form action="" [formGroup]="regForm">

<div *ngFor="let item of regForm.get('items')?.['controls'];let i=index">
<div [formGroupName]="i">

<input type="text" formControlName="fname">
<input type="text" formControlName="lname">

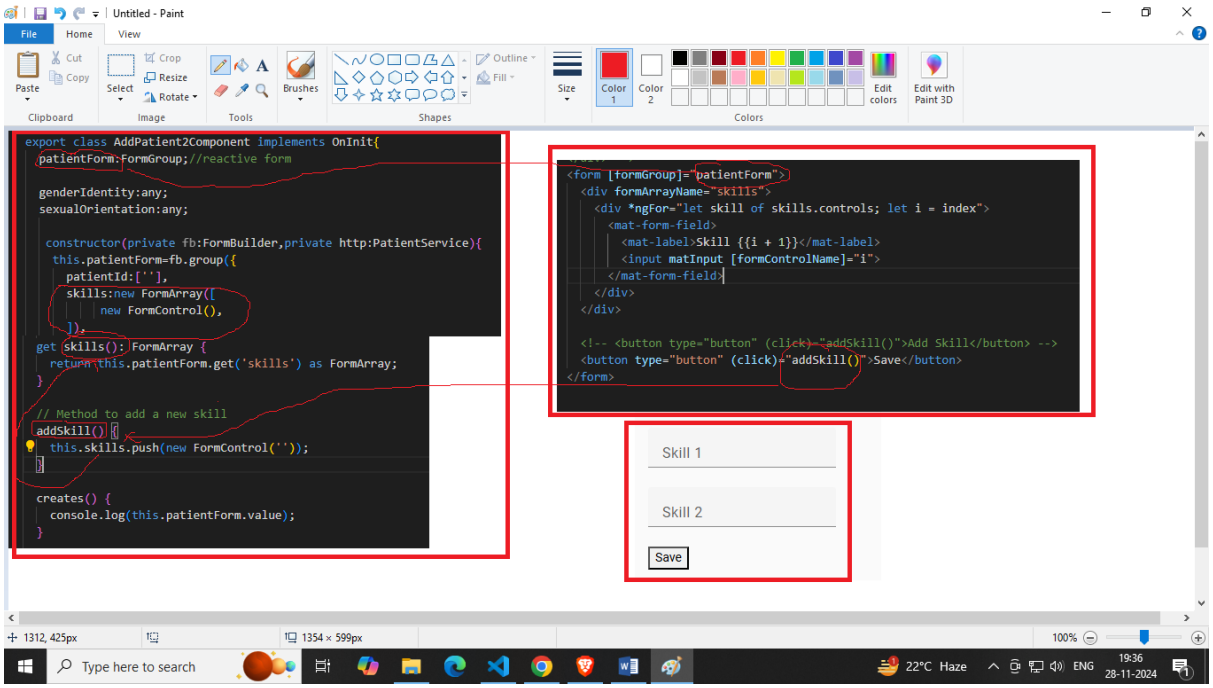
<button (click)="add()">add</button>
</form>
step:1 create form and declare [formGroup]="formName"
step:2 declare ur formArrayName="formArrayName"
step:3 make looping
step:4 delcare [formGroupName]="i"
step:5 delcare all ur fileds and add formContrlName
step: 6 add click events
Activate Windows
700, 137px
1354 × 599px
100%
Type here to search
23°C Cloudy
ENG
11:52
26-12-2024


Untitled - Paint
step1: Declare formArray and formGroup
step2: import formBuilder in constructor and also use formArray
step3: add create() method
step4: add add() and delete(i) methods
export class AgGridComponent {
regForm: FormGroup;//declare formGroup
items: FormArray;//declare formArray
mobileItems:FormArray;//declare formArray
constructor(private fb: FormBuilder) {
this.regForm= new FormGroup({
items:new FormArray([]),
mobileItems:new FormArray([]),
})
}//create item
createItem(): FormGroup {
return this.fb.group({
name: '',
lname: '',
age: ''
})
}//add item
addItem(): void {
this.items = this.regForm.get('items') as FormArray
this.items.push(this.createItem())
}//delete item
delete(i: number) {
this.items.removeAt(i)
}//create mobile
createMobileItem():FormGroup{
return this.fb.group({
mobileId:'',
mobileName:'',
<form action="" [formGroup]="regForm">
<div formArrayName="items" *ngFor="let item of regForm.get('items')?.['controls'];let i=index">
<div [formGroupName]="i">
<input type="text" formControlName="name">
<input type="text" formControlName="lname">
<input type="text" formControlName="age">
<button (click)="delete(i)">delete</button>

<button (click)="addItem()">add</button>
<!-- mobile items -->
<h1>Mobile Items</h1>
<div formArrayName="mobileItems" *ngFor="let item of regForm.get('mobileItems')?.['controls'];let i=index">

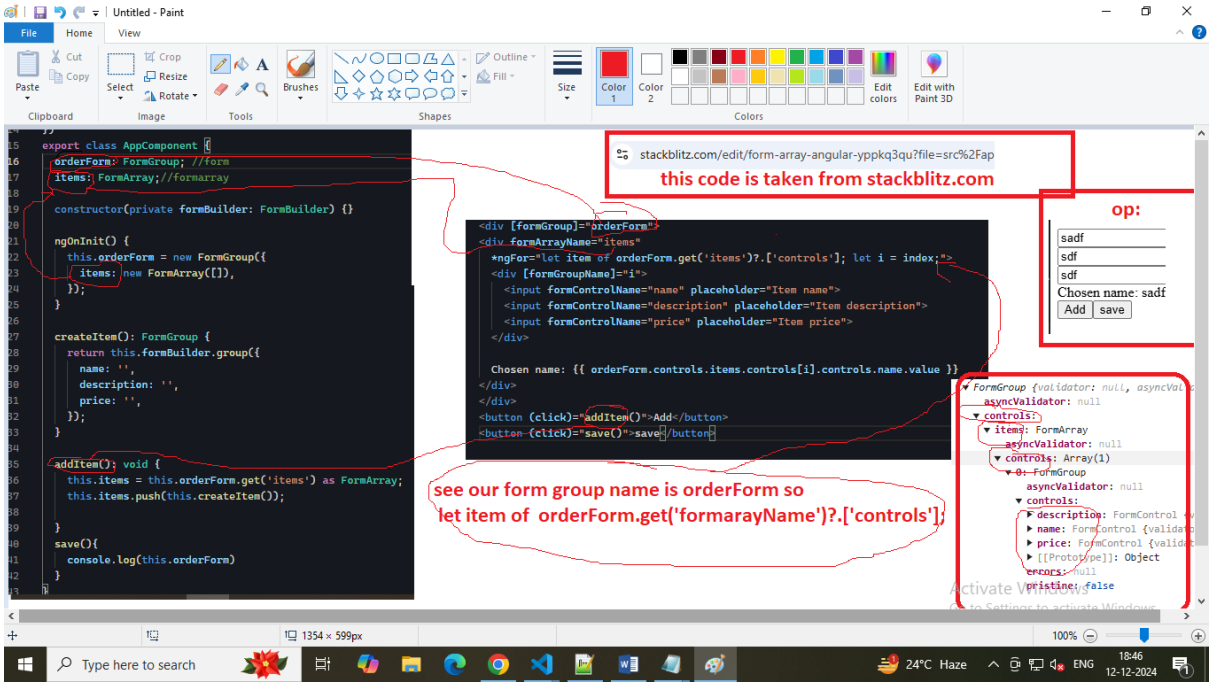
<input type="text" formControlName="mobileId">
<input type="text" formControlName="mobileName">
<input type="text" formControlName="mobileModel">
<button (click)="deleteMobile(i)">delete Mobile</button>

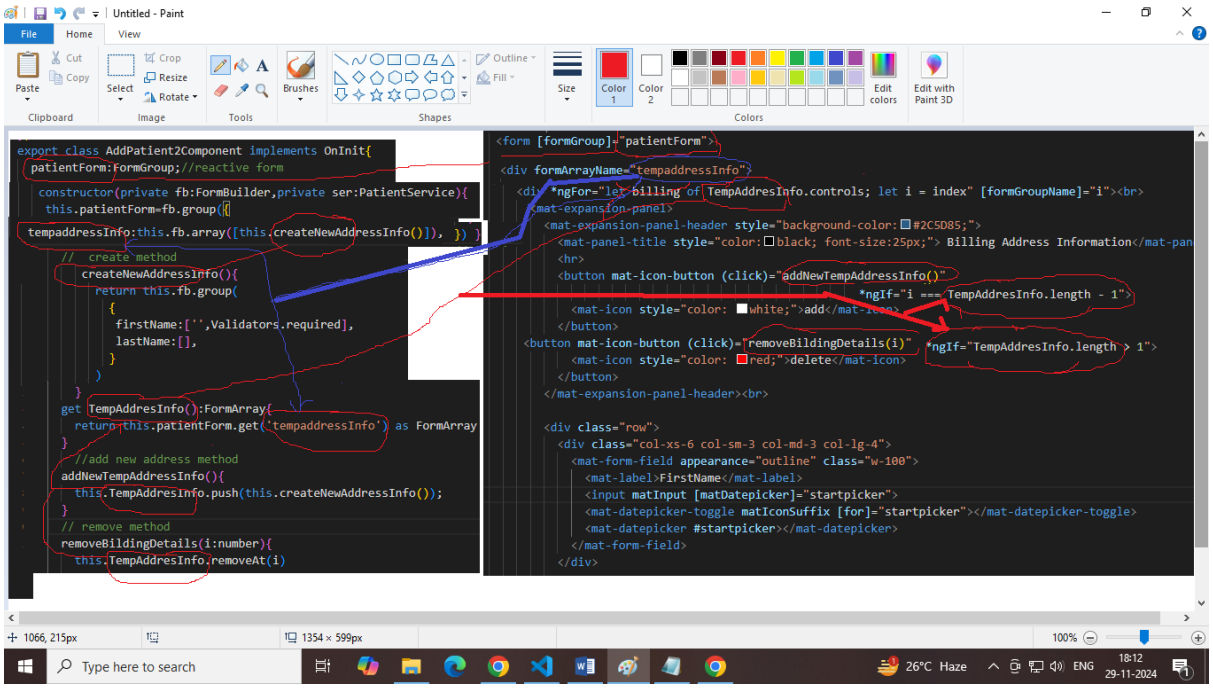
<button (click)="addMobile()">Add mobile</button>
</form>
Activate Windows
493, 227px
1354 × 599px
100%
Type here to search
24°C Mostly cloudy
ENG
12:38
25-12-2024

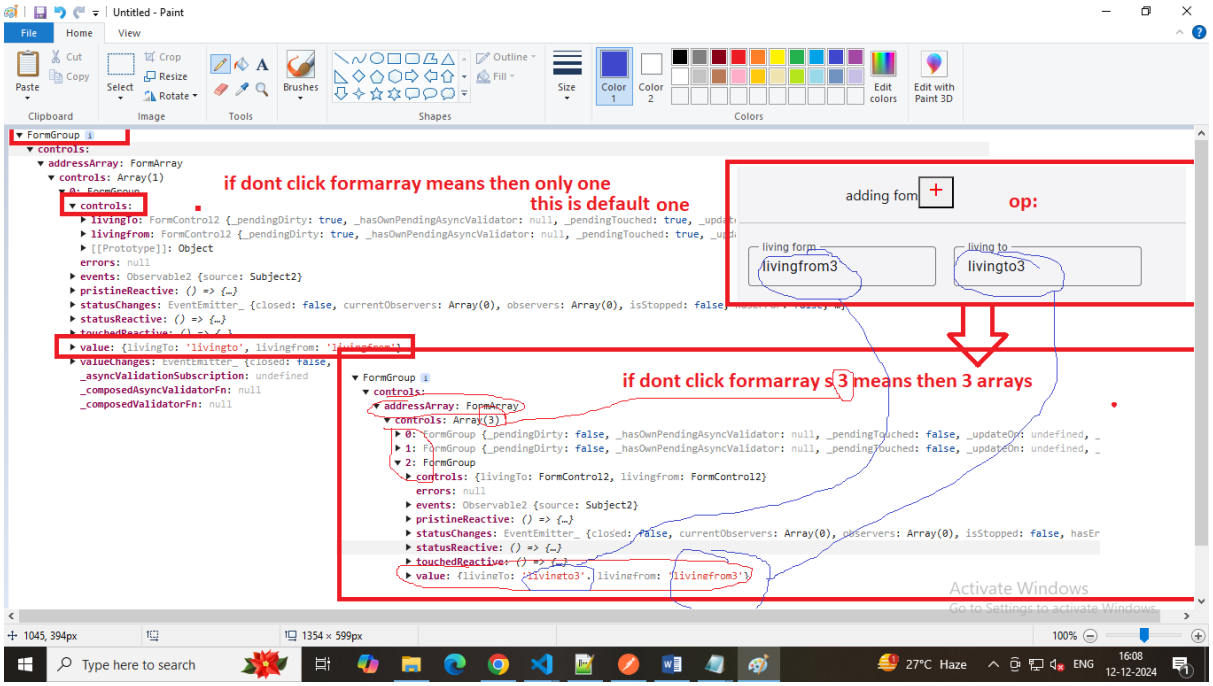

Untitled - Paint
export class AddPatient2Component implements OnInit{
patientForm:FormGroup;//reactive form
genderIdentity:any;
sexualOrientation:any;
constructor(private fb:FormBuilder,private http:PatientService){
this.patientForm=fb.group({
patientId:[''],
skills:new FormArray([
new FormControl(),
]),
get skills(): FormArray {
return this.patientForm.get('skills') as FormArray;
}
// Method to add a new skill
addSkill() {
this.skills.push(new FormControl(''));
}
creates() {
console.log(this.patientForm.value);
}
<form [formGroup]="patientForm">

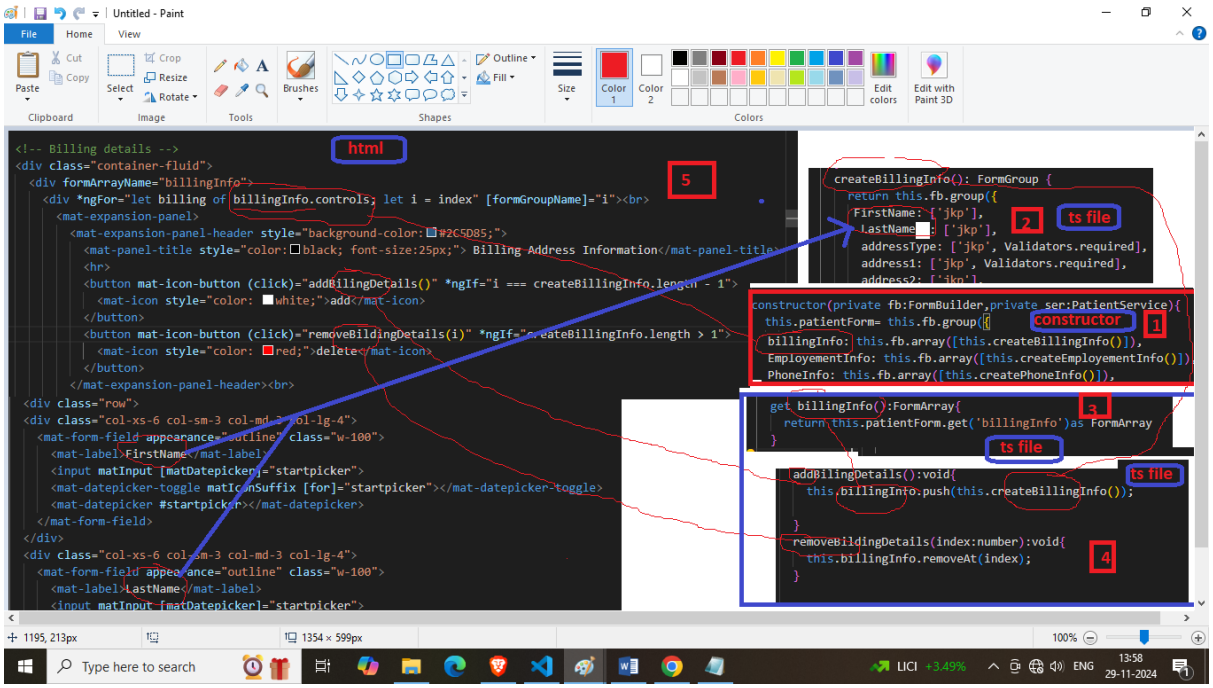
<div *ngFor="let skill of skills.controls; let i = index">
<mat-form-field>
<mat-label>Skill {{i + 1}}</mat-label>
<input matInput [formControlName]="i">
</mat-form-field>

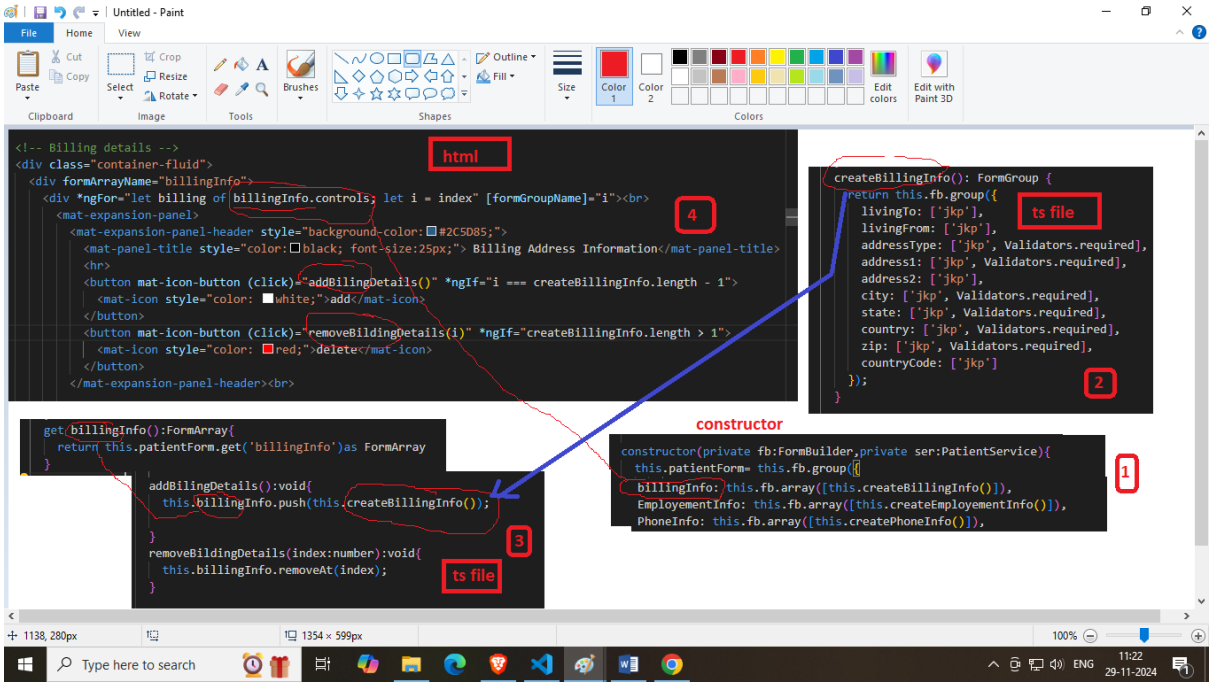
<!-- <button type="button" (click)="addSkill()">Add Skill</button> -->
<button type="button" (click)="addSkill()">Save</button>
</form>
Skill 1
Skill 2
Save
1312, 425px
1354 × 599px
100%
Type here to search
22°C Haze
ENG
19:36
28-11-2024


Untitled - Paint
File
Home
View
stackblitz.com/edit/form-array-angular-yppkq3qu?file=src%2Fap
this code is taken from stackblitz.com
op:
sadf
sdf
sdf
Chosen name: sadf
Add
save
see our form group name is orderForm so
let item of orderForm.get('formarayName')?.['controls'];
Type here to search
24°C Haze
ENG
18:46
12-12-2024

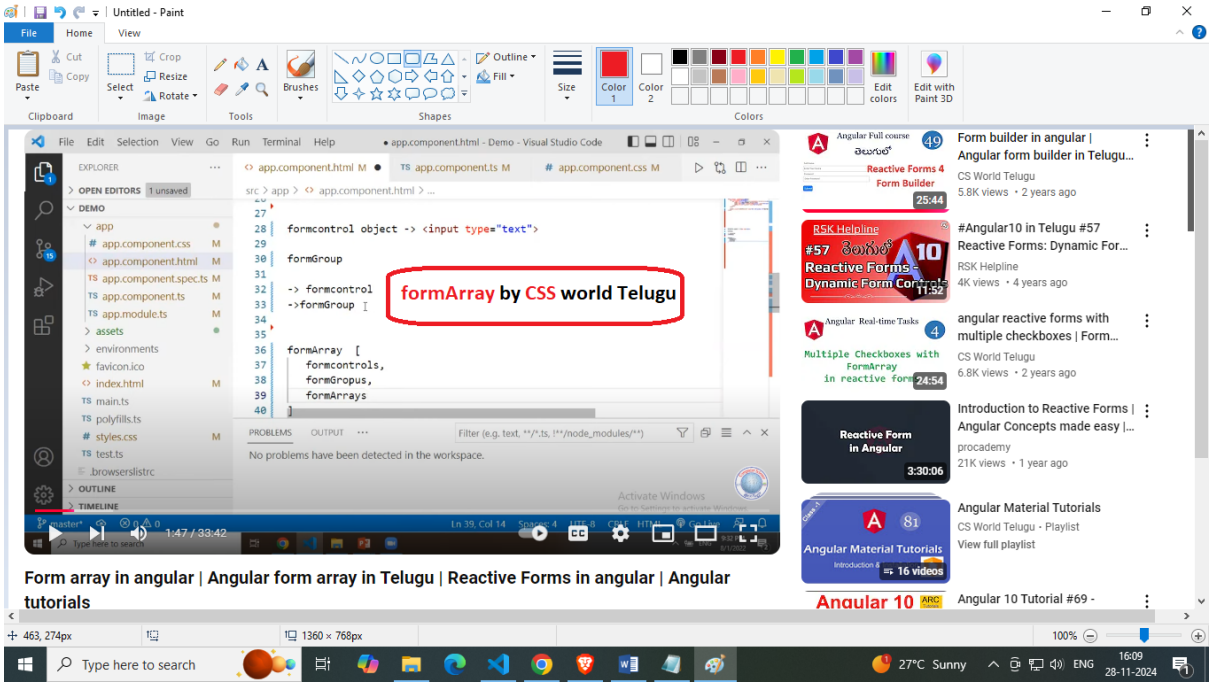

Untitled - Paint
File
Home
View
Type here to search
26°C Haze
ENG
18:12
29-11-2024

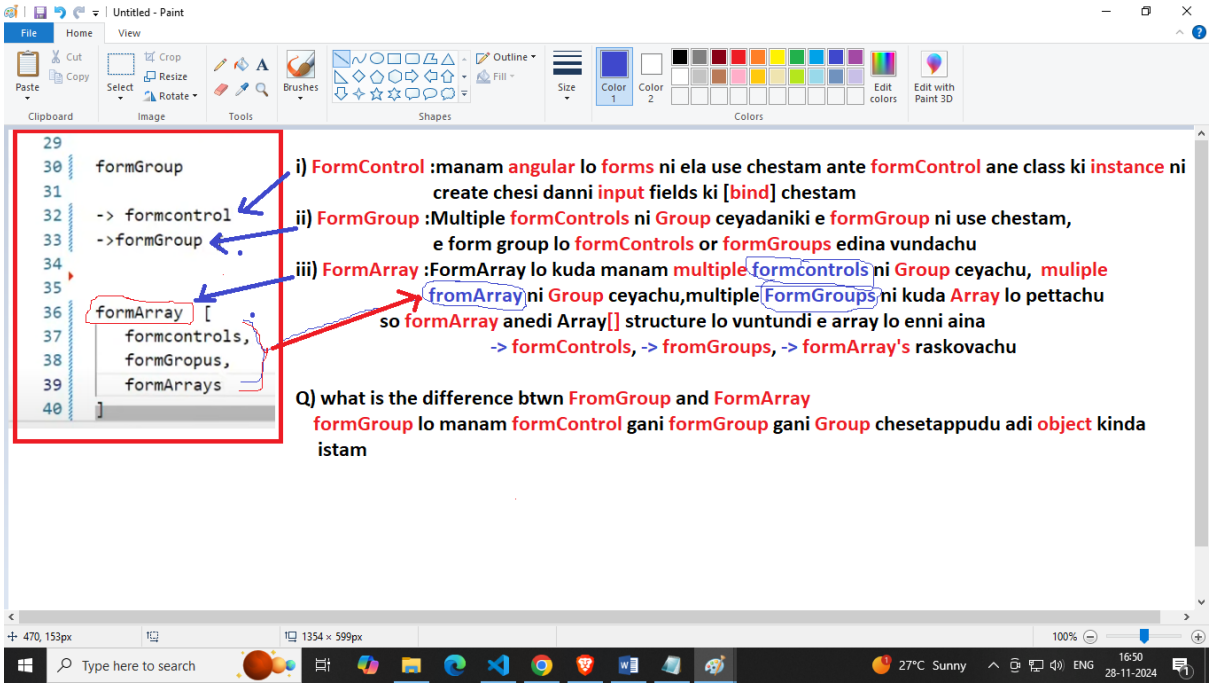

Untitled - Paint
if dont click formarray means then only one
this is default one
adding fom
op:
living form
livingfrom3
living to
livingto3
if dont click formarray s 3 means then 3 arrays
Activate Windows

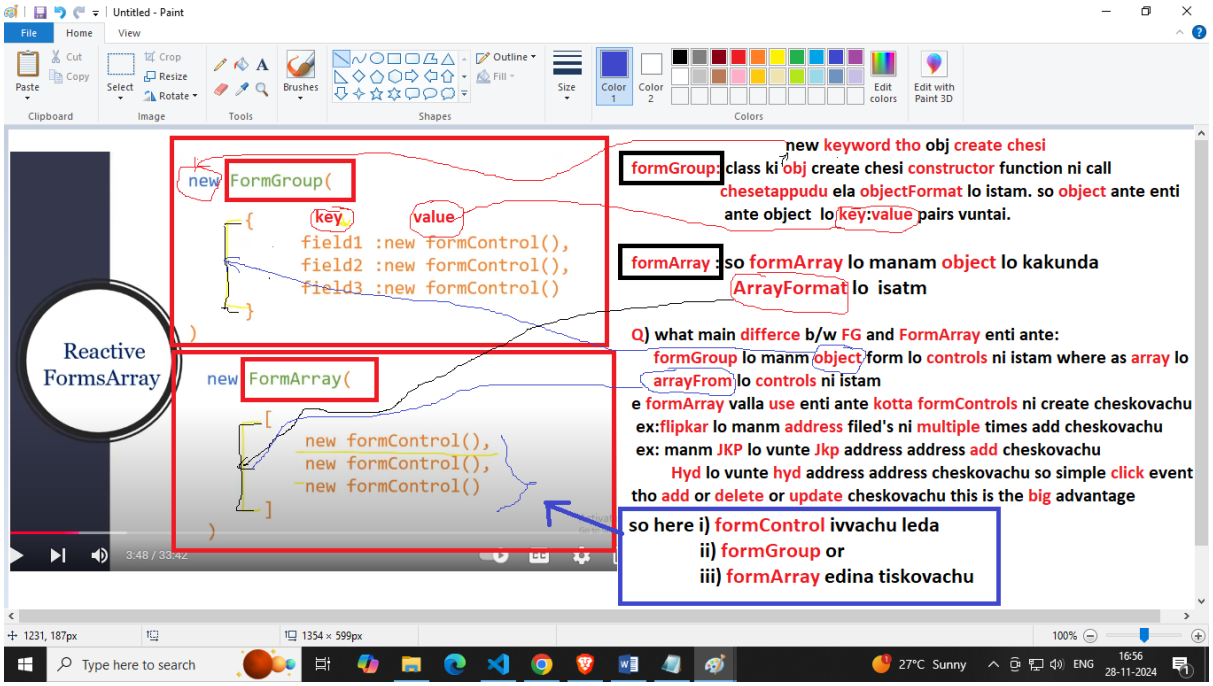

Untitled - Paint
html
5
ts file
constructor
1
2
3
4
ts file
ts file

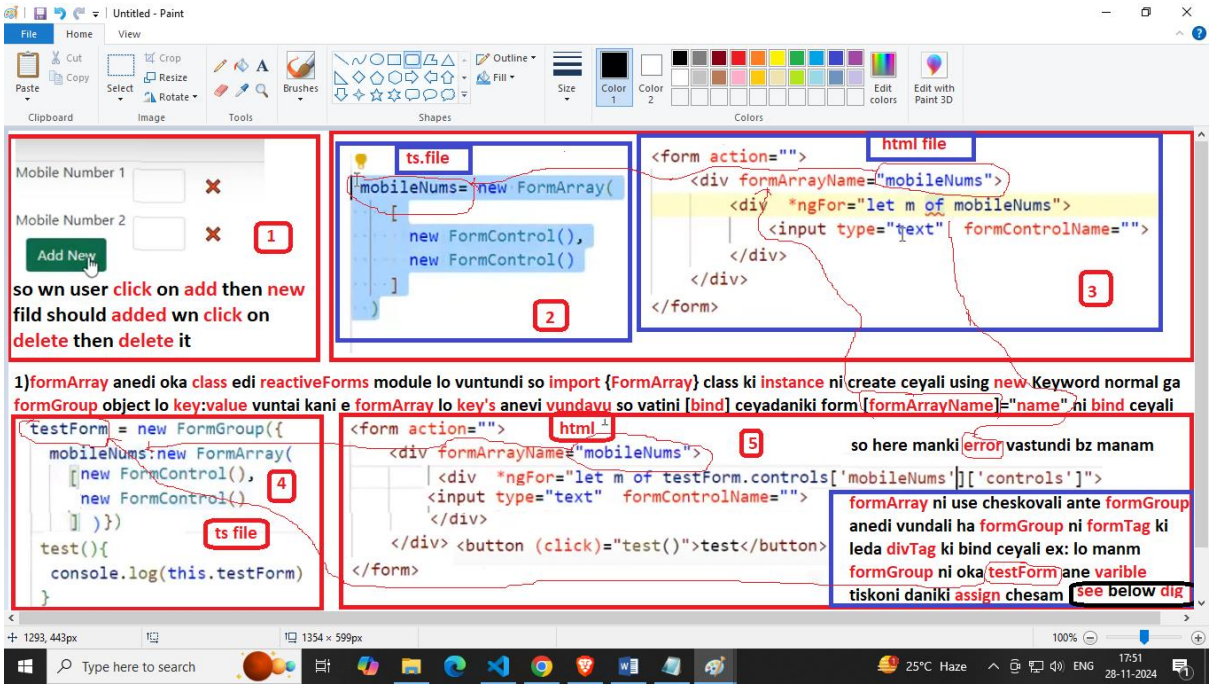

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
html
<!-- Billing details -->

<div *ngFor="let billing of billingInfo.controls; let i = index" [formGroupName]="i">
<mat-expansion-panel>
<mat-expansion-panel-header style="background-color:#2C5D85;">
<mat-panel-title style="color:black; font-size:25px;"> Billing Address Information</mat-panel-title>
<hr>
<button mat-icon-button (click)="addBilingDetails()" *ngIf="i === createBillingInfo.length - 1">
<mat-icon style="color: white;">add</mat-icon>
</button>
<button mat-icon-button (click)="removeBildingDetails(i)" *ngIf="createBillingInfo.length > 1">
<mat-icon style="color: red;">delete</mat-icon>
</button>
</mat-expansion-panel-header>
4
createBillingInfo(): FormGroup {
return this.fb.group({
livingTo: ['jkp'],
livingFrom: ['jkp'],
addressType: ['jkp', Validators.required],
address1: ['jkp', Validators.required],
address2: ['jkp'],
city: ['jkp', Validators.required],
state: ['jkp', Validators.required],
country: ['jkp', Validators.required],
zip: ['jkp', Validators.required],
countryCode: ['jkp']
});
}
ts file
2
get billingInfo():FormArray{
return this.patientForm.get('billingInfo')as FormArray
}
addBilingDetails():void{
this.billingInfo.push(this.createBillingInfo());
}
removeBildingDetails(index:number):void{
this.billingInfo.removeAt(index);
}
3
ts file
constructor
constructor(private fb:FormBuilder,private ser:PatientService){
this.patientForm= this.fb.group({
billingInfo: this.fb.array([this.createBillingInfo()]),
EmployementInfo: this.fb.array([this.createEmployementInfo()]),
PhoneInfo: this.fb.array([this.createPhoneInfo()]),
1
1138, 280px
1354 × 599px
100%
Type here to search
ENG
11:22
29-11-2024


Untitled - Paint
formArray by CSS world Telugu
Form array in angular | Angular form array in Telugu | Reactive Forms in angular | Angular tutorials
Form builder in angular | Angular form builder in Telugu...
CS World Telugu
5.8K views • 2 years ago
#Angular10 in Telugu #57 Reactive Forms: Dynamic For...
RSK Helpline
4K views • 4 years ago
angular reactive forms with multiple checkboxes | Form...
CS World Telugu
6.8K views • 2 years ago
Introduction to Reactive Forms | Angular Concepts made easy |...
procademy
21K views • 1 year ago
Angular Material Tutorials
CS World Telugu • Playlist
View full playlist
Angular 10 Tutorial #69 -
463, 274px
1360 × 768px
100%
Type here to search
27°C Sunny
ENG
16:09
28-11-2024


Untitled - Paint
formGroup
-> formcontrol
->formGroup
formArray [
formcontrols,
formGropus,
formArrays
]
i) FormControl :manam angular lo forms ni ela use chestam ante formControl ane class ki instance ni create chesi danni input fields ki [bind] chestam
ii) FormGroup :Multiple formControls ni Group ceyadaniki e formGroup ni use chestam, e form group lo formControls or formGroups edina vundachu
iii) FormArray :FormArray lo kuda manam multiple formcontrols ni Group ceyachu, muliple fromArray ni Group ceyachu,multiple FormGroups ni kuda Array lo pettachu
so formArray anedi Array[] structure lo vuntundi e array lo enni aina -> formControls, -> fromGroups, -> formArray's raskovachu
Q) what is the difference btwn FromGroup and FormArray
formGroup lo manam formControl gani formGroup gani Group chesetappudu adi object kinda istam
470, 153px
1354 × 599px
100%
Type here to search
27°C Sunny
ENG
16:50
28-11-2024


Untitled - Paint
Reactive FormsArray
new FormGroup(
key
value
field1 :new formControl(),
field2 :new formControl(),
field3 :new formControl()
new FormArray(
new formControl(),
new formControl(),
new formControl()
new keyword tho obj create chesi
formGroup: class ki obj create chesi constructor function ni call chesetappudu ela objectFormat lo istam. so object ante enti ante object lo key:value pairs vuntai.
formArray : so formArray lo manam object lo kakunda ArrayFormat lo isatm
Q) what main difference b/w FG and FormArray enti ante:
formGroup lo manm object form lo controls ni istam where as array lo arrayFrom lo controls ni istam
e formArray valla use enti ante kotta formControls ni create cheskovachu
ex:flipkar lo manm address filed's ni multiple times add cheskovachu
ex: manm JKP lo vunte Jkp address address add cheskovachu
Hyd lo vunte hyd address address cheskovachu so simple click event tho add or delete or update cheskovachu this is the big advantage
so here i) formControl ivvachu leda
ii) formGroup or
iii) formArray edina tiskovachu

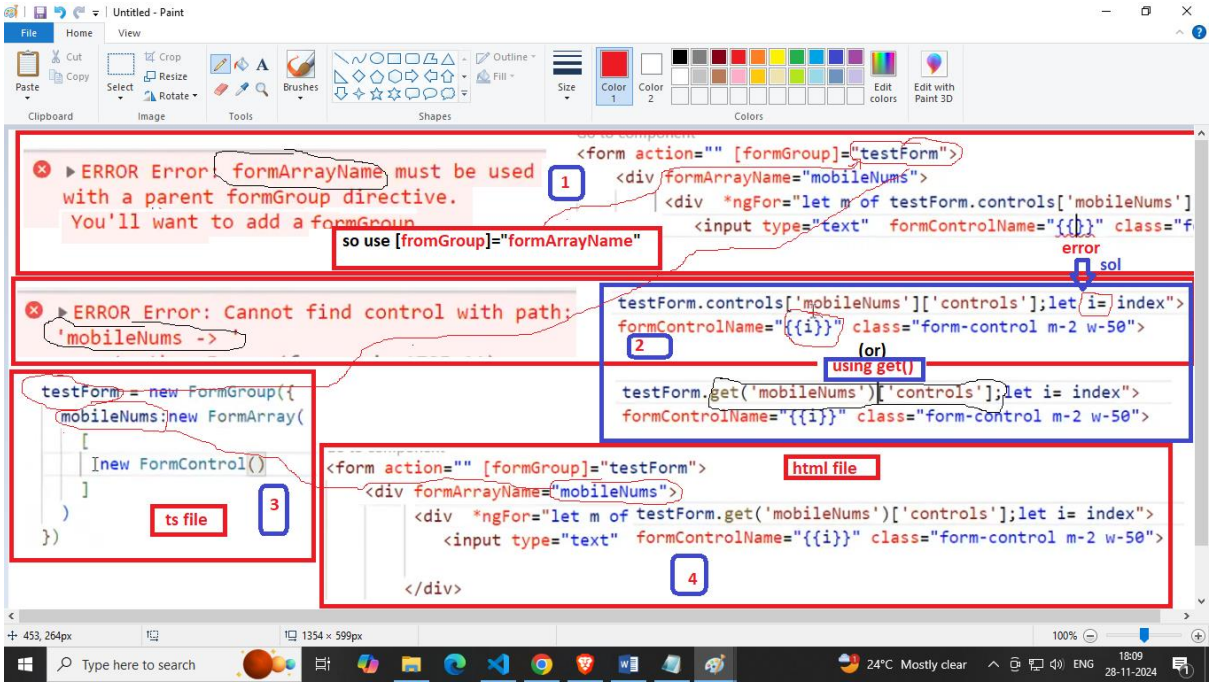

Untitled - Paint
Mobile Number 1
Mobile Number 2
Add New
1
so wn user click on add then new fild should added wn click on delete then delete it
ts.file
mobileNums= new FormArray(
new FormControl(),
new FormControl()
2
html file
<form action="">

<div *ngFor="let m of mobileNums">
<input type="text" formControlName="">

</form>
3
1)formArray anedi oka class edi reactiveForms module lo vuntundi so import {FormArray} class ki instance ni create ceyali using new Keyword normal ga formGroup object lo key:value vuntai kani e formArray lo key's anevi vundavu so vatini [bind] ceyadaniki form [formArrayName]="name" ni bind ceyali
testForm = new FormGroup({
mobileNums:new FormArray(
[new FormControl(),
new FormControl()
])})
test(){
console.log(this.testForm)
}
4
ts file
html
<form action="">

<div *ngFor="let m of testForm.controls['mobileNums']['controls']">
<input type="text" formControlName="">

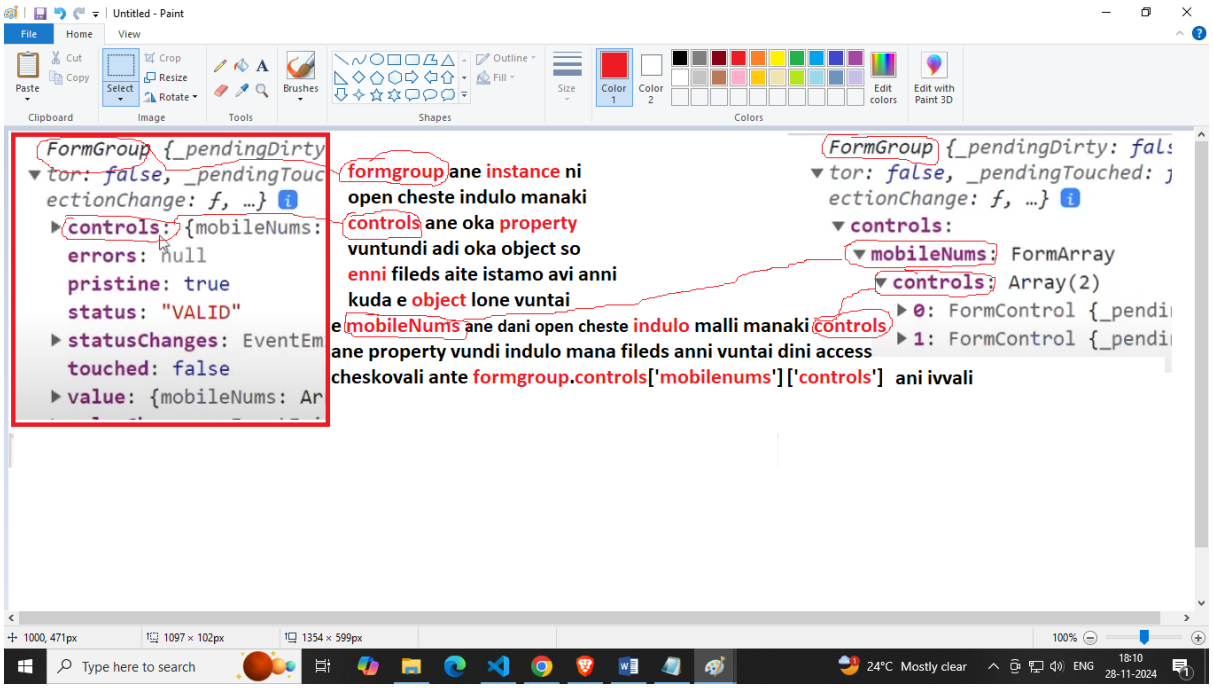
 <button (click)="test()">test</button>
</form>
5
so here manki error vastundi bz manam formArray ni use cheskovali ante formGroup anedi vundali ha formGroup ni formTag ki leda divTag ki bind ceyali ex: lo manm formGroup ni oka testForm ane varible tiskoni daniki assign chesam see below dig

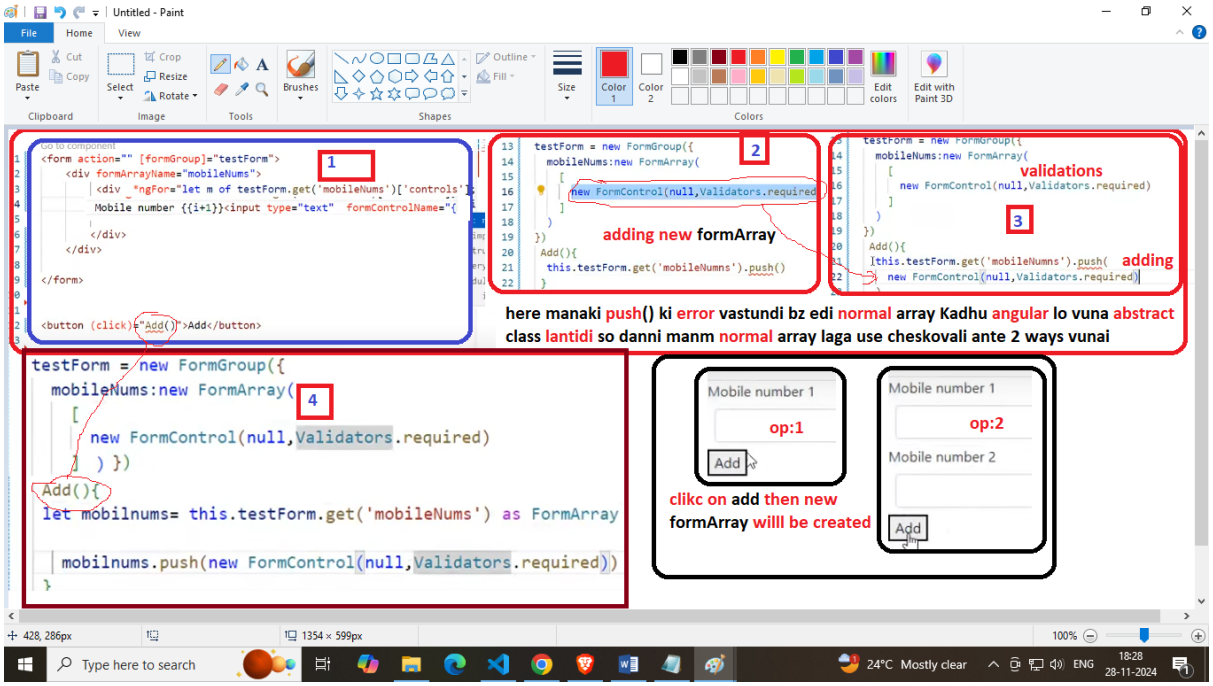

Untitled - Paint
ERROR Error: formArrayName must be used with a parent formGroup directive. You'll want to add a formGroup
so use [fromGroup]="formArrayName"
1
<form action="" [formGroup]="testForm">

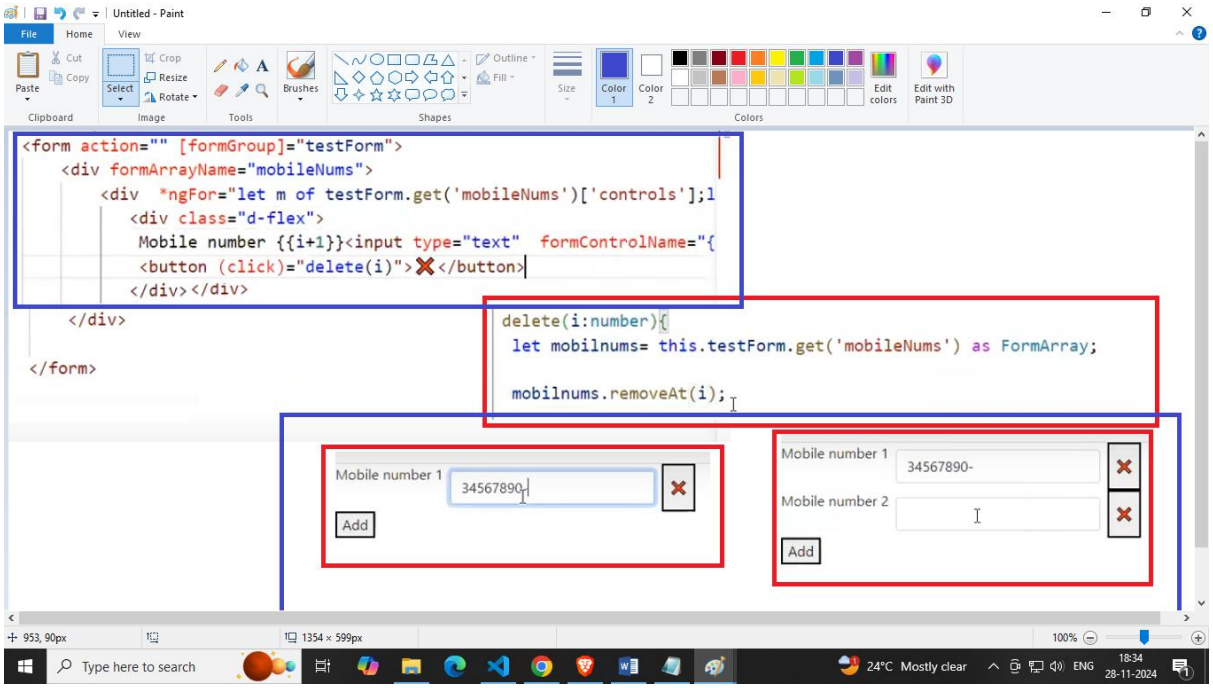
<div *ngFor="let m of testForm.controls['mobileNums']
<input type="text" formControlName="{{i}}" class="f
error
sol
ERROR Error: Cannot find control with path: 'mobileNums' ->
testForm.controls['mobileNums']['controls'];let i= index">
formControlName="{{i}}" class="form-control m-2 w-50">
2
(or)
using get()
testForm.get('mobileNums')['controls'];let i= index">
formControlName="{{i}}" class="form-control m-2 w-50">
testForm = new FormGroup({
mobileNums:new FormArray(
[
new FormControl()
]
)
})
ts file
3
html file
<form action="" [formGroup]="testForm">

<div *ngFor="let m of testForm.get('mobileNums')['controls'];let i= index">
<input type="text" formControlName="{{i}}" class="form-control m-2 w-50">

4
453, 264px
1354 × 599px
100%
Type here to search
24°C Mostly clear
ENG
18:09
28-11-2024


Untitled - Paint
FormGroup {_pendingDirty
tor: false, _pendingTouc
ectionChange: ƒ, …}
controls: {mobileNums:
errors: null
pristine: true
status: "VALID"
statusChanges: EventEm
touched: false
value: {mobileNums: Ar
formgroup ane instance ni
open cheste indulo manaki
controls ane oka property
vuntundi adi oka object so
enni fileds aite istamo avi anni
kuda e object lone vuntai
e mobileNums ane dani open cheste indulo malli manaki controls
ane property vundi indulo mana fileds anni vuntai dini access
cheskovali ante formgroup.controls['mobilenums'] ['controls'] ani ivvali
FormGroup {_pendingDirty: fals
tor: false, _pendingTouched: j
ectionChange: ƒ, …}
controls:
mobileNums: FormArray
controls: Array(2)
0: FormControl {_pendi
1: FormControl {_pendi
1000, 471px
1097 × 102px
1354 × 599px
100%
Type here to search
24°C Mostly clear
ENG
18:10
28-11-2024





Below is : Total text of formArray css world telugu

FormControl:manam angular lo forms ni ela use chestam ante formControl ane class ki instance ni create chesi danni inuput fields ki bind chestam

FormGroup:multiple formControls ni Group ceyadaniki e formGroup ni use chestam,e form group lo formControls or formGroups edina vundachu

FormArray:FormArray lo kuda manam multiple controls ni Group ceyachu,muliple Array ni or multiple FormGroups ni kuda Array pettachu

so formArray anedi Array[] structure lo vuntundi e array lo enni aina formControls,fromGroups,formArray's raskovachu

so difference btwn FromGroup and FormArray

formGroup lo manam formControl gani formGroup gani Group chesetappudu adi object kinda istam

formGroup class ki obj create chesi constructor function call chesetappudu ela objectFormat lo istam so object ante enti object lo key:value pairs vuntai.

formArray : so form array lo manam object lo kakunda arrayFormat lo isatm

main differce enti ante: formGroup lo manm object form lo controls ni istam where array lo arrayFrom lo controls ni istam

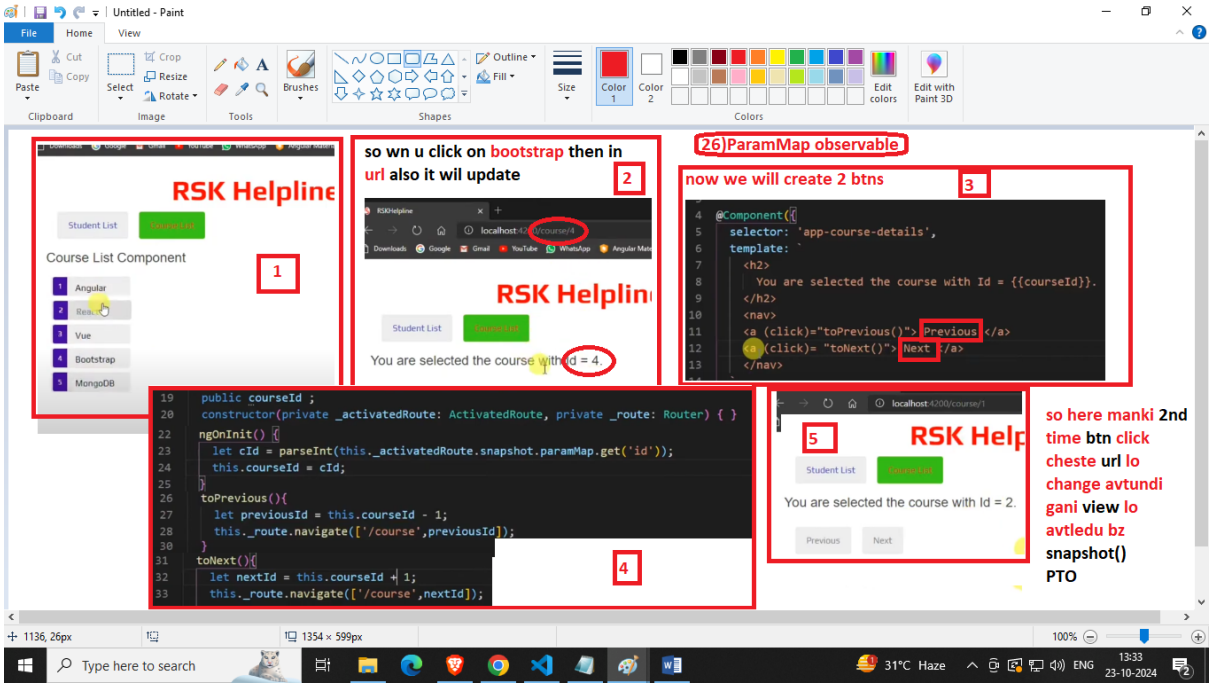
e formArray valla use enti ante kotta formControls ni create cheskovachu ex:flipkar lo manm address filed's ni multiple times add cheskovachu ex: manm JKP lo vunte Jkp address address add cheskovachu Hyd lo vunte hyd address address cheskovachu so simple click event tho add or delet or update cheskovachu

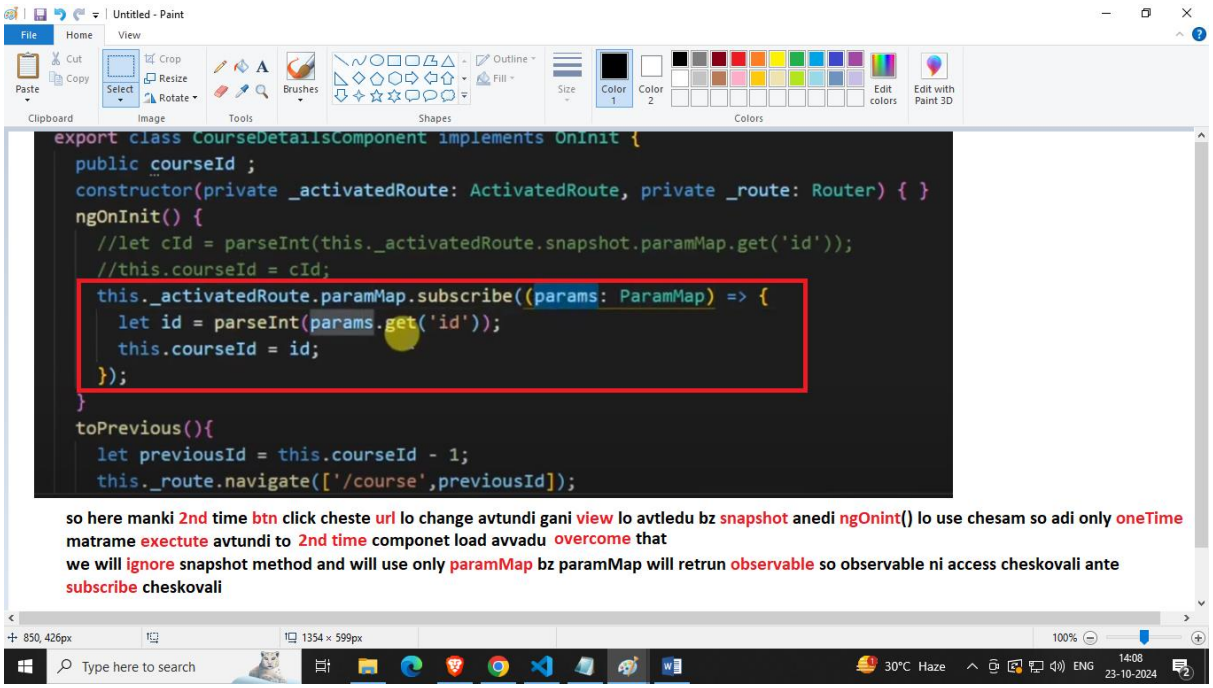
1) form array anedi oka class edi reactiveForms module lo vuntundi so import {FormArray} class ki instance ni create ceyali using new Keyword and normal ga formGroup object lo key:value vuntai kani e formArray lo key's anevi vundavu so vatini [bind] ceyadaniki form [formArrayName]="name" ni bind ceyali

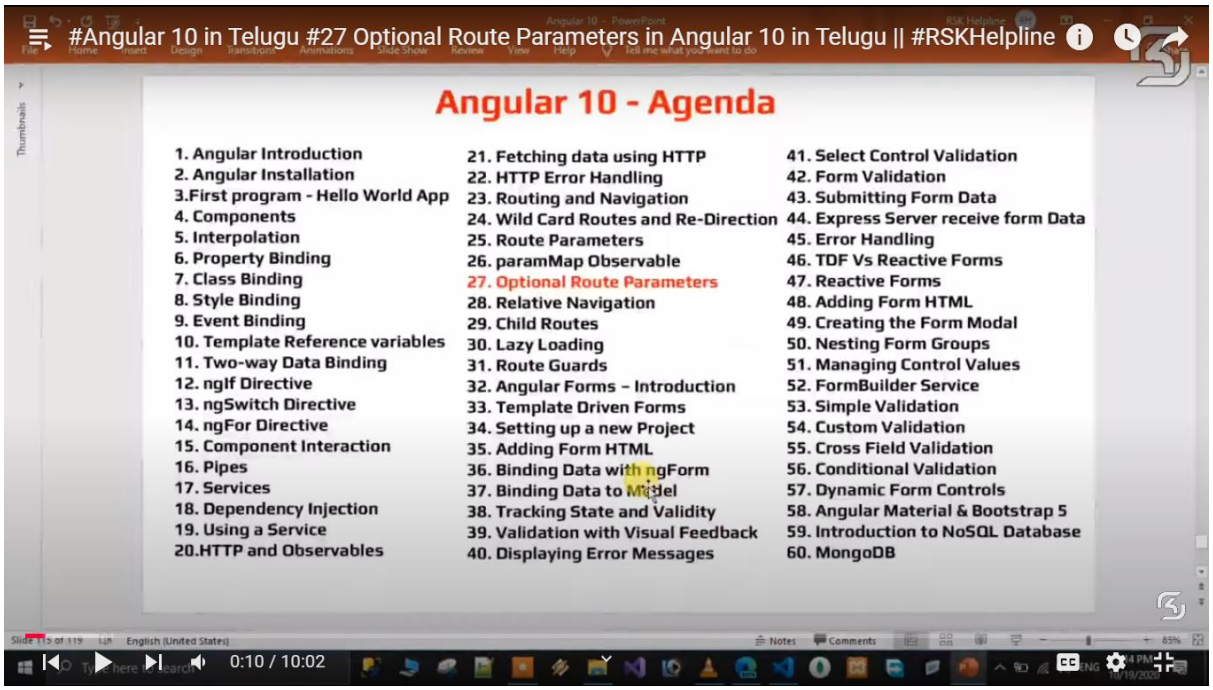
so here manki error vastundi bz manam formArray ni use cheskovali ante formGroup anedi vundali ha formGroup ni formTag ki leda divTag ki bind ceyali ex: lo manm formGroup ni oka testForm ani varible tiskoni daniki assign chesam

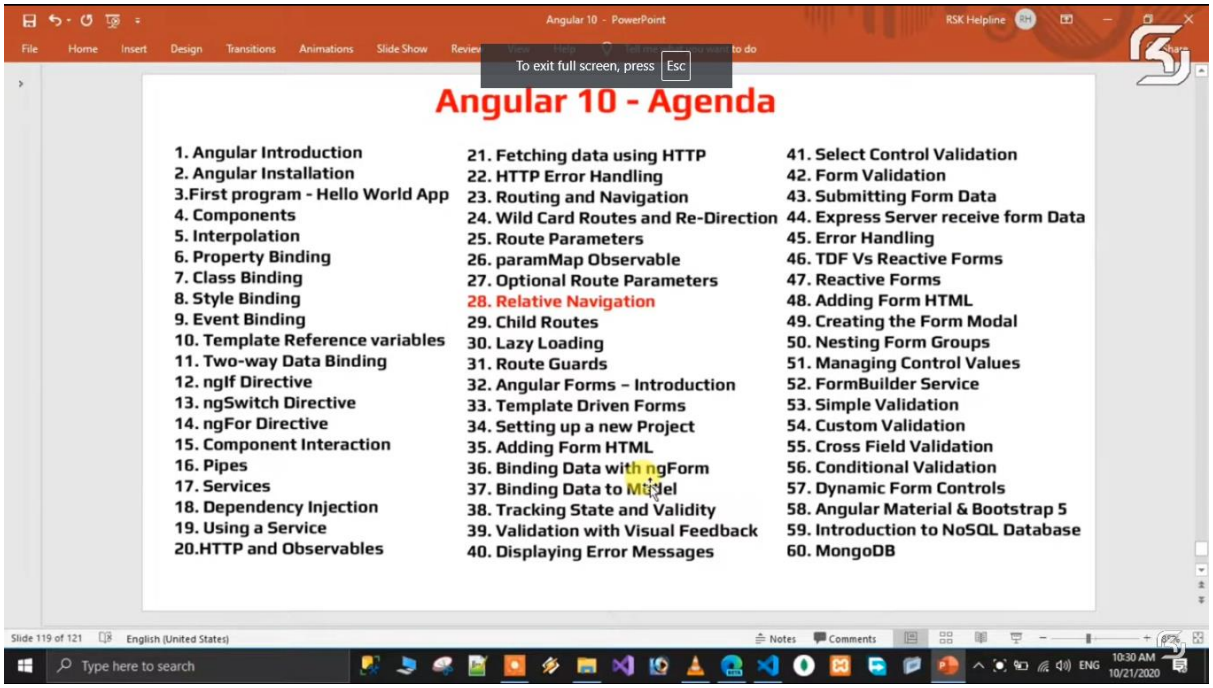
formgroup ane instance ni open cheste indulo manaki controls ane oka property vuntudni adi oka object so enni fileds aite istamo avi anni kuda e object lone vuntai e mobile ane dani open cheste indulo malli manki dini access cheskovali ante formgroup.controls['mobilenums'] ani ivvali then e moblienums lo malli manaki controls ane property indulo mana fileds anni vuntai

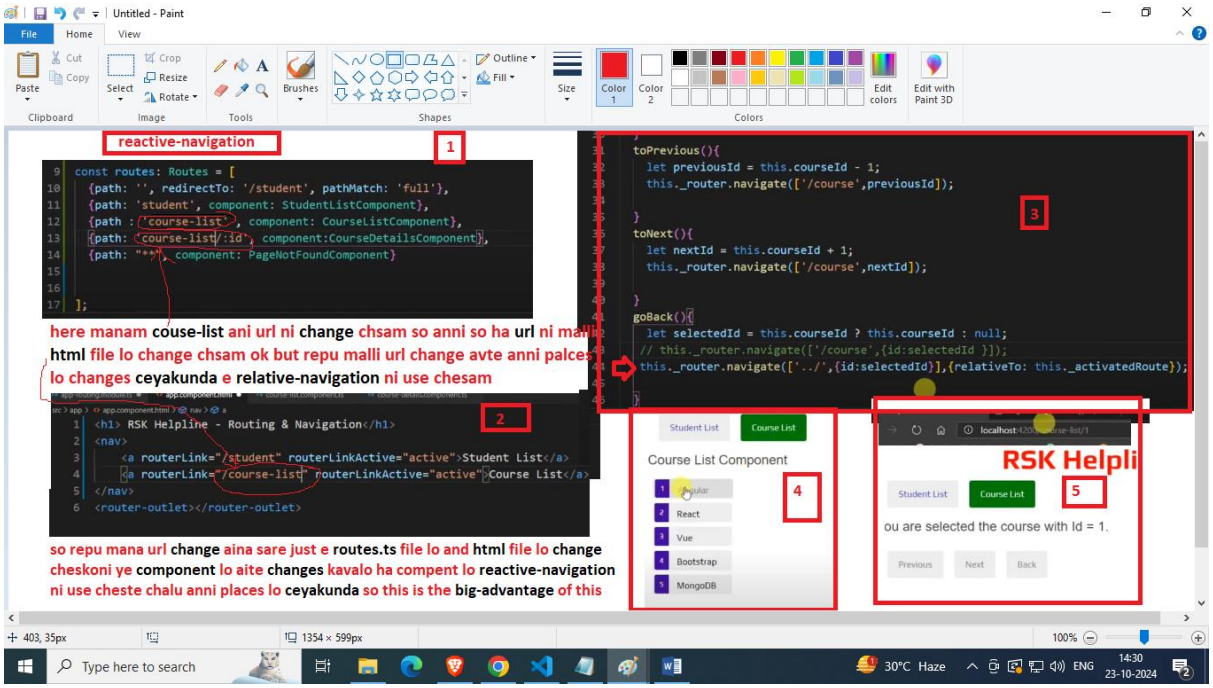
Routing by Rsk HelpLine

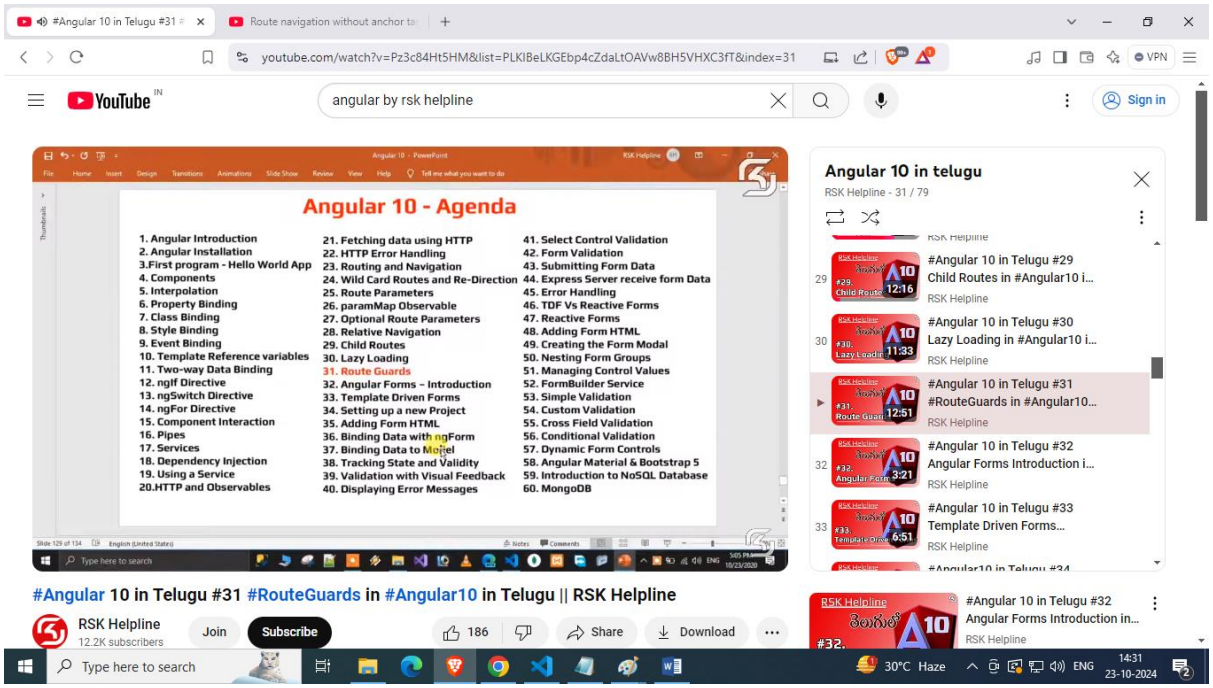


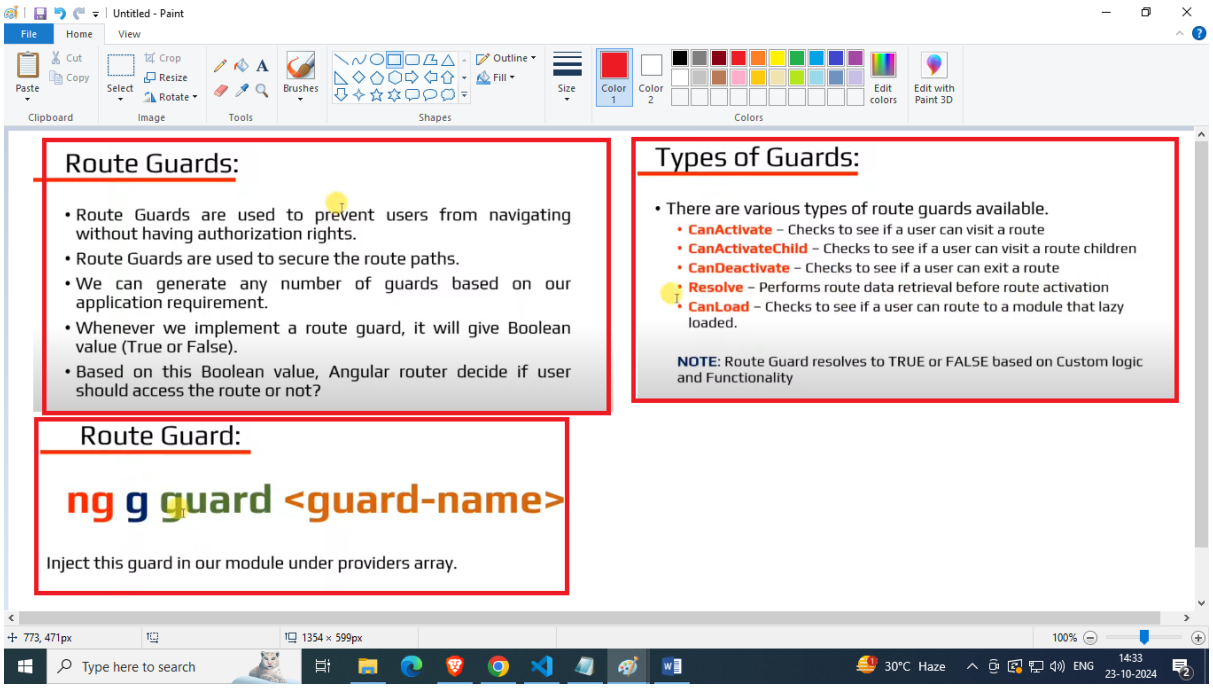

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
export class CourseDetailsComponent implements OnInit {
public courseId ;
constructor(private _activatedRoute: ActivatedRoute, private _route: Router) { }
ngOnInit() {
//let cId = parseInt(this._activatedRoute.snapshot.paramMap.get('id'));
//this.courseId = cId;
this._activatedRoute.paramMap.subscribe((params: ParamMap) => {
let id = parseInt(params.get('id'));
this.courseId = id;
});
}
toPrevious(){
let previousId = this.courseId - 1;
this._route.navigate(['/course',previousId]);
so here manki 2nd time btn click cheste url lo change avtundi gani view lo avtledu bz snapshot anedi ngOnint() lo use chesam so adi only oneTime matrame exectute avtundi to 2nd time componet load avvadu overcome that
we will ignore snapshot method and will use only paramMap bz paramMap will retrun observable so observable ni access cheskovali ante subscribe cheskovali
850, 426px
1354 × 599px
100%
Type here to search
30°C Haze
ENG
14:08
23-10-2024


#Angular 10 in Telugu #27 Optional Route Parameters in Angular 10 in Telugu || #RSKHelpline
Angular 10 - Agenda
1. Angular Introduction
2. Angular Installation
3.First program - Hello World App
4. Components
5. Interpolation
6. Property Binding
7. Class Binding
8. Style Binding
9. Event Binding
10. Template Reference variables
11. Two-way Data Binding
12. ngIf Directive
13. ngSwitch Directive
14. ngFor Directive
15. Component Interaction
16. Pipes
17. Services
18. Dependency Injection
19. Using a Service
20.HTTP and Observables
21. Fetching data using HTTP
22. HTTP Error Handling
23. Routing and Navigation
24. Wild Card Routes and Re-Direction
25. Route Parameters
26. paramMap Observable
27. Optional Route Parameters
28. Relative Navigation
29. Child Routes
30. Lazy Loading
31. Route Guards
32. Angular Forms – Introduction
33. Template Driven Forms
34. Setting up a new Project
35. Adding Form HTML
36. Binding Data with ngForm
37. Binding Data to Model
38. Tracking State and Validity
39. Validation with Visual Feedback
40. Displaying Error Messages
41. Select Control Validation
42. Form Validation
43. Submitting Form Data
44. Express Server receive form Data
45. Error Handling
46. TDF Vs Reactive Forms
47. Reactive Forms
48. Adding Form HTML
49. Creating the Form Modal
50. Nesting Form Groups
51. Managing Control Values
52. FormBuilder Service
53. Simple Validation
54. Custom Validation
55. Cross Field Validation
56. Conditional Validation
57. Dynamic Form Controls
58. Angular Material & Bootstrap 5
59. Introduction to NoSQL Database
60. MongoDB
0:10 / 10:02

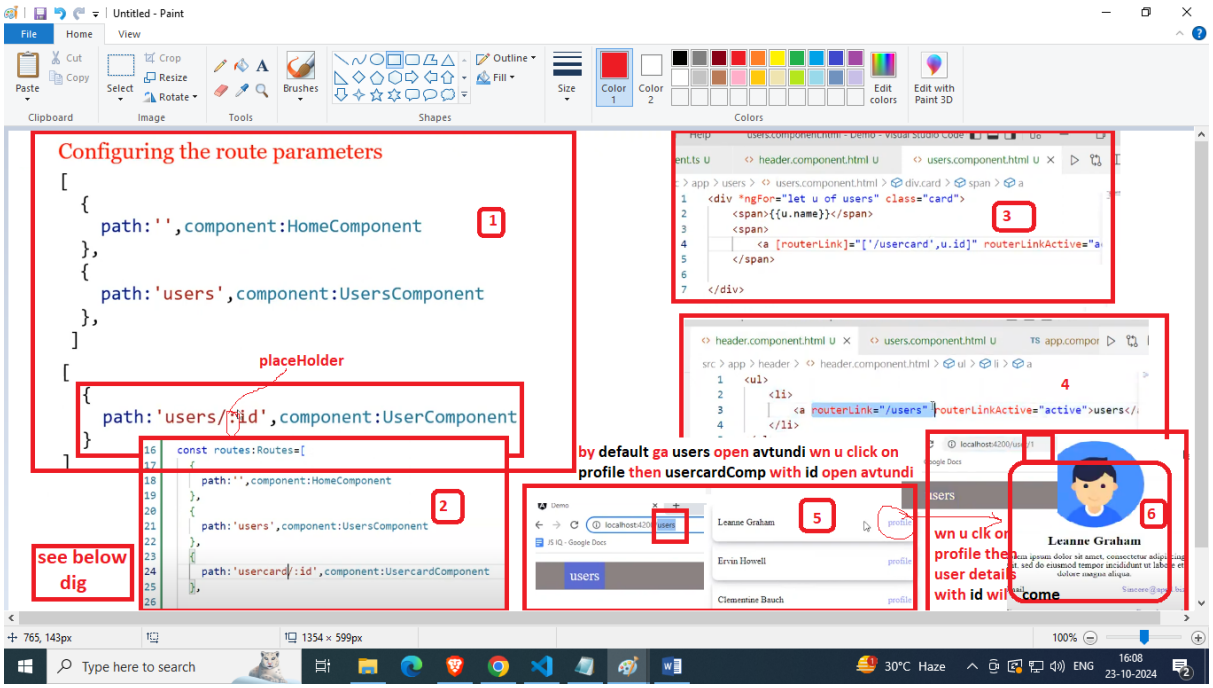

Angular 10 - PowerPoint
To exit full screen, press Esc
Angular 10 - Agenda
1. Angular Introduction
2. Angular Installation
3.First program - Hello World App
4. Components
5. Interpolation
6. Property Binding
7. Class Binding
8. Style Binding
9. Event Binding
10. Template Reference variables
11. Two-way Data Binding
12. ngIf Directive
13. ngSwitch Directive
14. ngFor Directive
15. Component Interaction
16. Pipes
17. Services
18. Dependency Injection
19. Using a Service
20.HTTP and Observables
21. Fetching data using HTTP
22. HTTP Error Handling
23. Routing and Navigation
24. Wild Card Routes and Re-Direction
25. Route Parameters
26. paramMap Observable
27. Optional Route Parameters
28. Relative Navigation
29. Child Routes
30. Lazy Loading
31. Route Guards
32. Angular Forms – Introduction
33. Template Driven Forms
34. Setting up a new Project
35. Adding Form HTML
36. Binding Data with ngForm
37. Binding Data to Model
38. Tracking State and Validity
39. Validation with Visual Feedback
40. Displaying Error Messages
41. Select Control Validation
42. Form Validation
43. Submitting Form Data
44. Express Server receive form Data
45. Error Handling
46. TDF Vs Reactive Forms
47. Reactive Forms
48. Adding Form HTML
49. Creating the Form Modal
50. Nesting Form Groups
51. Managing Control Values
52. FormBuilder Service
53. Simple Validation
54. Custom Validation
55. Cross Field Validation
56. Conditional Validation
57. Dynamic Form Controls
58. Angular Material & Bootstrap 5
59. Introduction to NoSQL Database
60. MongoDB
Slide 119 of 121
English (United States)
Notes
Comments
Type here to search
ENG
10:30 AM
10/21/2020

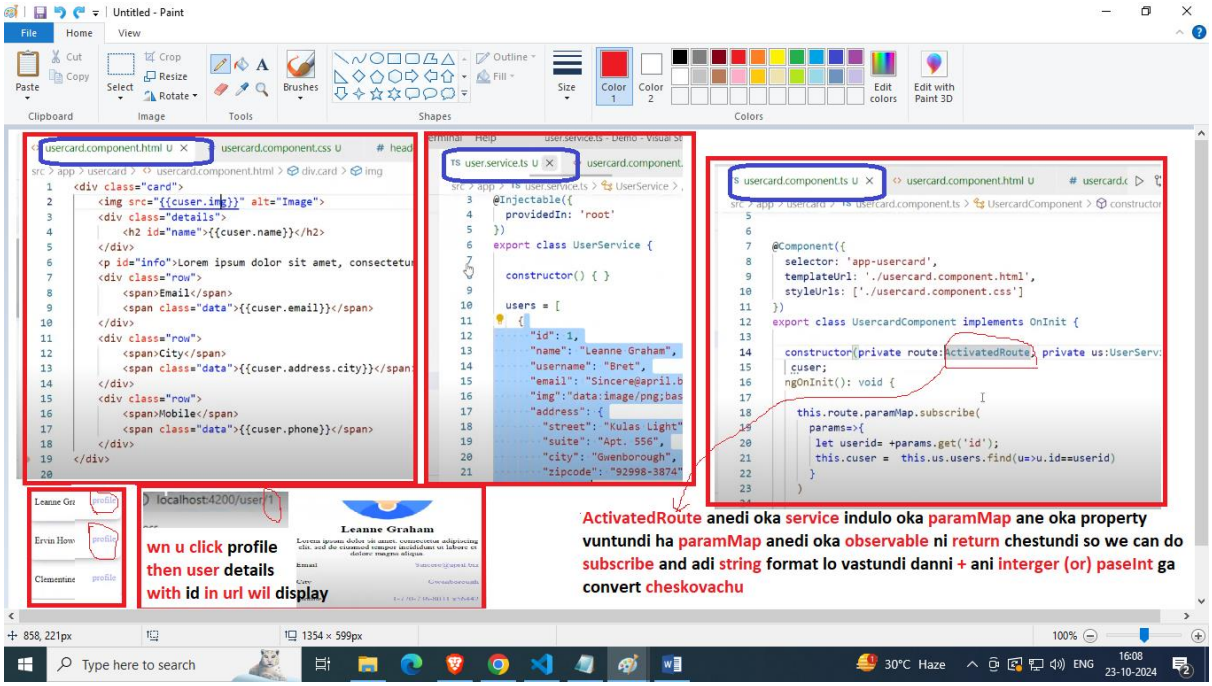

Untitled - Paint
reactive-navigation
1
here manam couse-list ani url ni change chsam so anni so ha url ni malli
html file lo change chsam ok but repu malli url change avte anni palces
lo changes ceyakunda e relative-navigation ni use chesam
2
3
4
5
Course List Component
RSK Helpli
ou are selected the course with Id = 1.
so repu mana url change aina sare just e routes.ts file lo and html file lo change
cheskoni ye component lo aite changes kavalo ha compent lo reactive-navigation
ni use cheste chalu anni places lo ceyakunda so this is the big-advantage of this
Type here to search
30°C Haze
ENG
14:30
23-10-2024

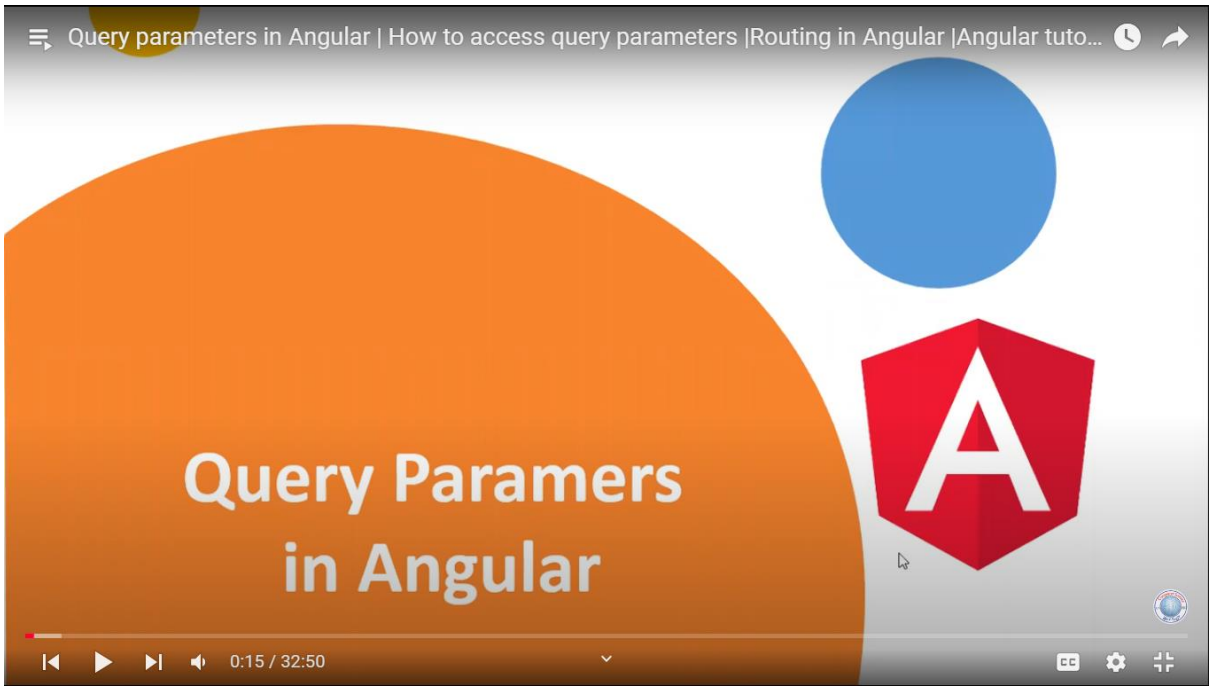

#Angular 10 in Telugu #31
youtube.com/watch?v=Pz3c84Ht5HM&list=PLKIBeLKGEbp4cZdaLtOAVw8BH5VHXC3fT&index=31
angular by rsk helpline
Angular 10 - Agenda
1. Angular Introduction
2. Angular Installation
3.First program - Hello World App
4. Components
5. Interpolation
6. Property Binding
7. Class Binding
8. Style Binding
9. Event Binding
10. Template Reference variables
11. Two-way Data Binding
12. ngIf Directive
13. ngSwitch Directive
14. ngFor Directive
15. Component Interaction
16. Pipes
17. Services
18. Dependency Injection
19. Using a Service
20.HTTP and Observables
21. Fetching data using HTTP
22. HTTP Error Handling
23. Routing and Navigation
24. Wild Card Routes and Re-Direction
25. Route Parameters
26. paramMap Observable
27. Optional Route Parameters
28. Relative Navigation
29. Child Routes
30. Lazy Loading
31. Route Guards
32. Angular Forms – Introduction
33. Template Driven Forms
34. Setting up a new Project
35. Adding Form HTML
36. Binding Data with ngForm
37. Binding Data to Model
38. Tracking State and Validity
39. Validation with Visual Feedback
40. Displaying Error Messages
41. Select Control Validation
42. Form Validation
43. Submitting Form Data
44. Express Server receive form Data
45. Error Handling
46. TDF Vs Reactive Forms
47. Reactive Forms
48. Adding Form HTML
49. Creating the Form Modal
50. Nesting Form Groups
51. Managing Control Values
52. FormBuilder Service
53. Simple Validation
54. Custom Validation
55. Cross Field Validation
56. Conditional Validation
57. Dynamic Form Controls
58. Angular Material & Bootstrap 5
59. Introduction to NoSQL Database
60. MongoDB
Angular 10 in telugu
RSK Helpline - 31 / 79
#Angular 10 in Telugu #29
Child Routes in #Angular10 i...
RSK Helpline
#Angular 10 in Telugu #30
Lazy Loading in #Angular10 i...
RSK Helpline
#Angular 10 in Telugu #31
#RouteGuards in #Angular10...
RSK Helpline
#Angular 10 in Telugu #32
Angular Forms Introduction i...
RSK Helpline
#Angular 10 in Telugu #33
Template Driven Forms...
RSK Helpline
#Angular 10 in Telugu #31 #RouteGuards in #Angular10 in Telugu || RSK Helpline
RSK Helpline
12.2K subscribers
Join
Subscribe
186
Share
Download
#Angular 10 in Telugu #32
Angular Forms Introduction in...
RSK Helpline
Type here to search
30°C Haze
ENG
14:31
23-10-2024

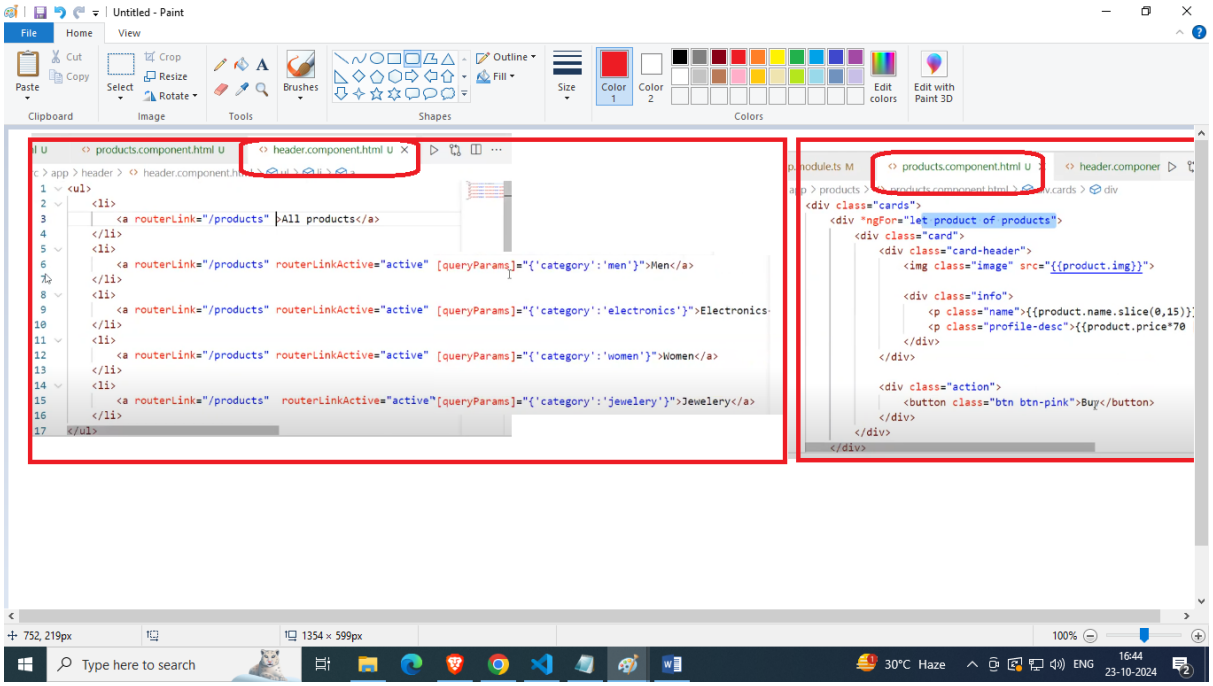

Untitled - Paint
File
Home
View
Route Guards:
• Route Guards are used to prevent users from navigating without having authorization rights.
• Route Guards are used to secure the route paths.
• We can generate any number of guards based on our application requirement.
• Whenever we implement a route guard, it will give Boolean value (True or False).
• Based on this Boolean value, Angular router decide if user should access the route or not?
Types of Guards:
• There are various types of route guards available.
• CanActivate – Checks to see if a user can visit a route
• CanActivateChild – Checks to see if a user can visit a route children
• CanDeactivate – Checks to see if a user can exit a route
• Resolve – Performs route data retrieval before route activation
• CanLoad – Checks to see if a user can route to a module that lazy loaded.
NOTE: Route Guard resolves to TRUE or FALSE based on Custom logic and Functionality
Route Guard:
ng g guard <guard-name>
Inject this guard in our module under providers array.
773, 471px
1354 × 599px
100%
Type here to search
30°C Haze
ENG
14:33
23-10-2024


angular by cs world telugu
Sign in
Access Route Paramers
in Angular
0:02 / 22:43
Angular Course for beginners
CS World Telugu - 30 / 93
Routing in Angular |Angular routing tutorial|router outlet i...
CS World Telugu
Services in Angular |Angular Services in telugu |...
CS World Telugu
Route parameters in Angular | How to access route...
CS World Telugu
Query parameters in Angular | How to access query...
CS World Telugu
Route navigation without anchor tag | query parameter...
CS World Telugu
Route parameters in Angular | How to access route parameters| Routing in Angular |Angular tutorials
Profile card design html css | product card design html css |...
CS World Telugu
Type here to search
30°C Haze
ENG
16:09
23-10-2024


Untitled - Paint
Configuring the route parameters
path:'',component:HomeComponent
path:'users',component:UsersComponent
placeHolder
path:'users/:id',component:UserComponent
see below dig
by default ga users open avtundi wn u click on profile then usercardComp with id open avtundi
wn u clk on profile then user details with id will come
Leanne Graham
Type here to search
30°C Haze
ENG
16:08
23-10-2024


Untitled - Paint
usercard.component.html
usercard.component.css
user.service.ts
usercard.component.ts
@Component({
selector: 'app-usercard',
templateUrl: './usercard.component.html',
styleUrls: ['./usercard.component.css']
})
export class UsercardComponent implements OnInit {
constructor(private route:ActivatedRoute, private us:UserServ
cuser;
ngOnInit(): void {
this.route.paramMap.subscribe(
params=>{
let userid= +params.get('id');
this.cuser = this.us.users.find(u=>u.id==userid)
}
)
localhost:4200/user/1
Leanne Graham
wn u click profile
then user details
with id in url wil display
ActivatedRoute anedi oka service indulo oka paramMap ane oka property
vuntundi ha paramMap anedi oka observable ni return chestundi so we can do
subscribe and adi string format lo vastundi danni + ani interger (or) paseInt ga
convert cheskovachu
Type here to search
30°C Haze
ENG
16:08
23-10-2024


Query parameters in Angular | How to access query parameters |Routing in Angular |Angular tuto...
Query Paramers
in Angular
0:15 / 32:50


Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
products.component.html U
header.component.html U
app > header > header.component.html

<a routerLink="/products" >All products

<a routerLink="/products" routerLinkActive="active" [queryParams]="{'category':'men'}">Men

<a routerLink="/products" routerLinkActive="active" [queryParams]="{'category':'electronics'}">Electronics

<a routerLink="/products" routerLinkActive="active" [queryParams]="{'category':'women'}">Women

<a routerLink="/products" routerLinkActive="active"[queryParams]="{'category':'jewelery'}">Jewelery

products.component.html U
header.componer
app > products > products.component.html > div.cards > div

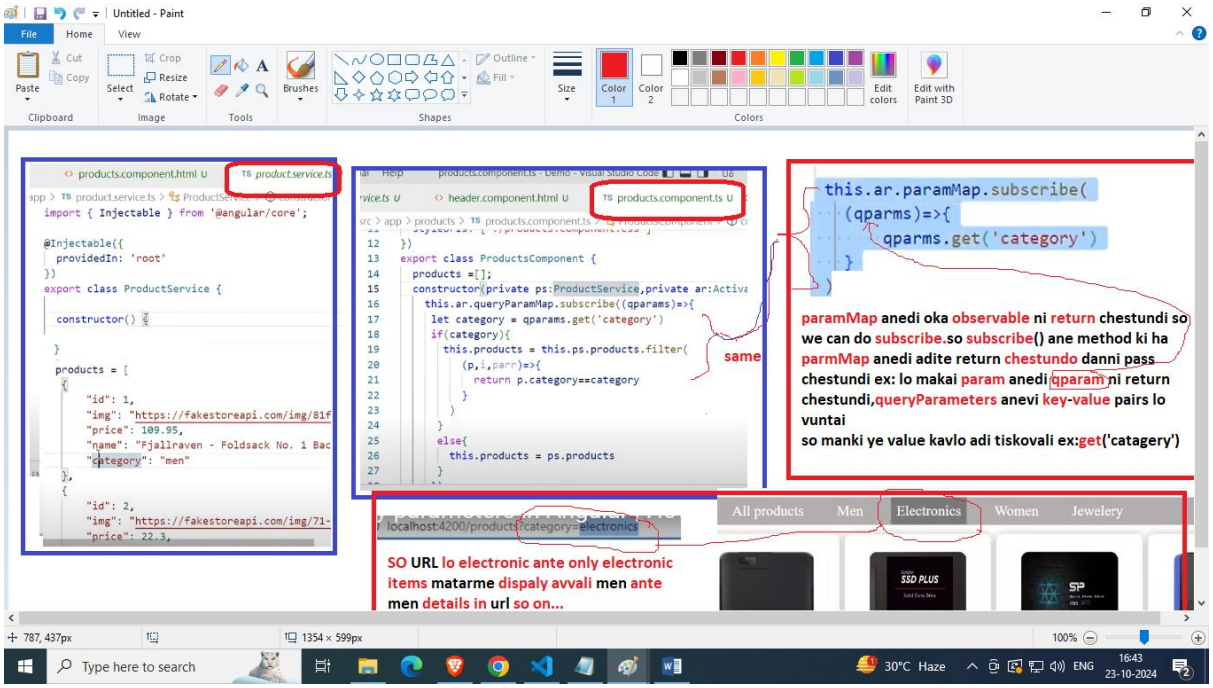
<div *ngFor="let product of products">

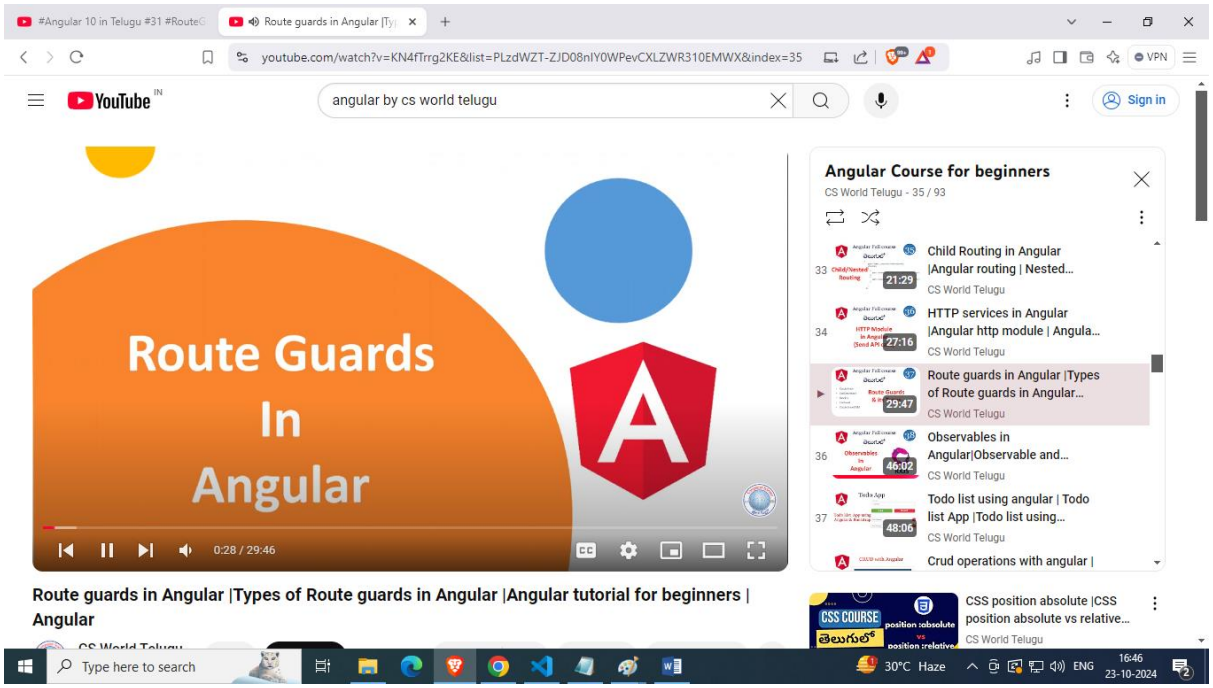
<img class="image" src="{{product.img}}">

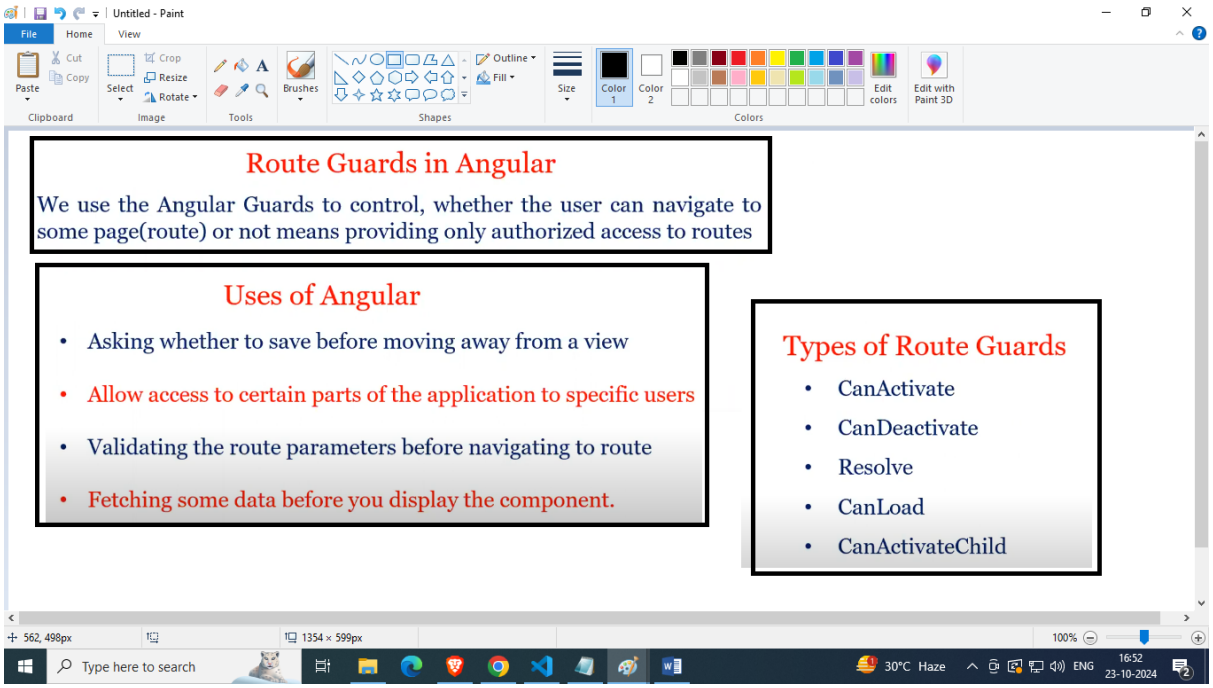
<p class="name">{{product.name.slice(0,15)}}
<p class="profile-desc">{{product.price*70

<button class="btn btn-pink">Buy</button>

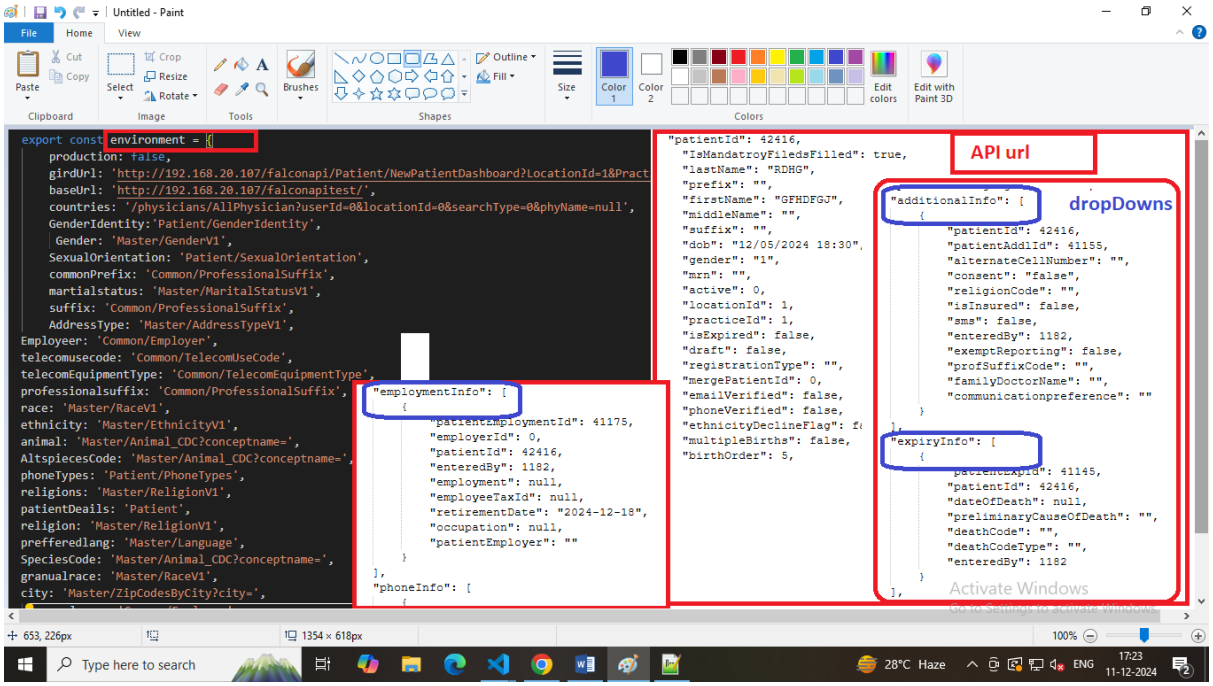
752, 219px
1354 × 599px
100%
Type here to search
30°C Haze
ENG
16:44
23-10-2024

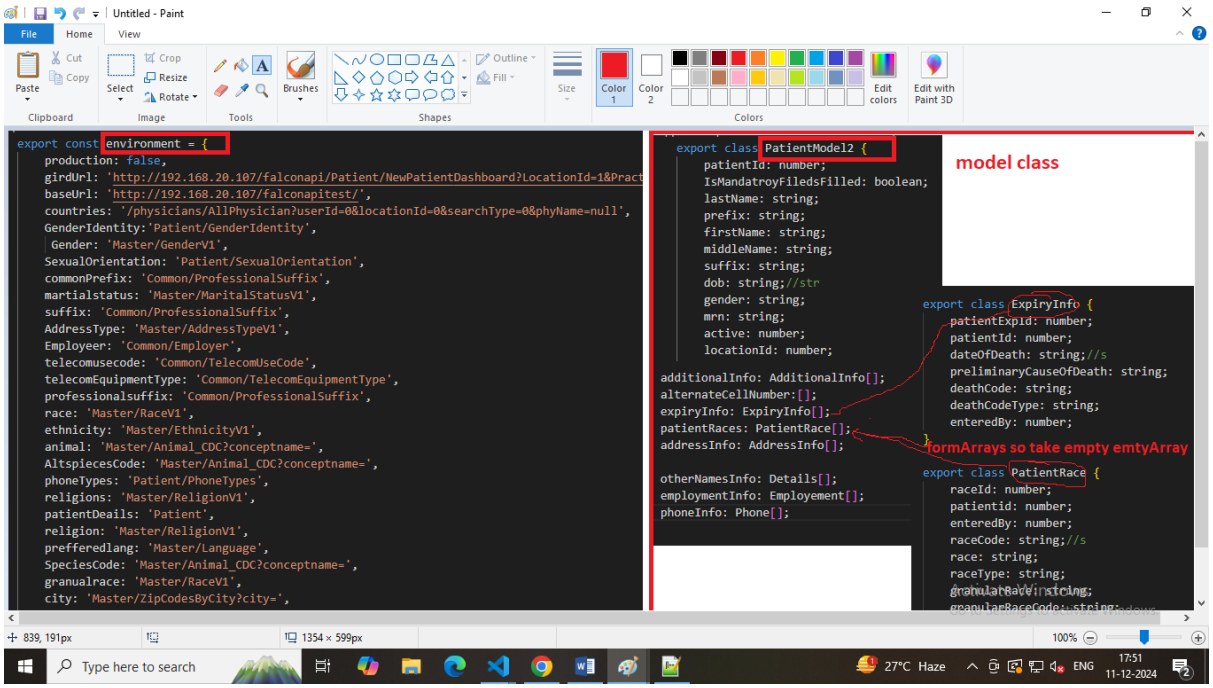

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
products.component.html U
product.service.ts
import { Injectable } from '@angular/core';
@Injectable({
providedIn: 'root'
})
export class ProductService {
constructor() {
}
products = [
{
"id": 1,
"img": "https://fakestoreapi.com/img/81f
"price": 109.95,
"name": "Fjallraven - Foldsack No. 1 Bac
"category": "men"
},
{
"id": 2,
"img": "https://fakestoreapi.com/img/71-
"price": 22.3,
header.component.html U
products.component.ts U
products.component.ts - Demo - Visual Studio Code
export class ProductsComponent {
products =[];
constructor(private ps:ProductService,private ar:Activa
this.ar.queryParamMap.subscribe((qparams)=>{
let category = qparams.get('category')
if(category){
this.products = this.ps.products.filter(
(p,i,parr)=>{
return p.category==category
}
)
}
else{
this.products = ps.products
}
same
this.ar.paramMap.subscribe(
(qparms)=>{
qparms.get('category')
}
)
paramMap anedi oka observable ni return chestundi so we can do subscribe.so subscribe() ane method ki ha parmMap anedi adite return chestundo danni pass chestundi ex: lo makai param anedi qparam ni return chestundi,queryParameters anevi key-value pairs lo vuntai
so manki ye value kavlo adi tiskovali ex:get('catagery')
localhost:4200/products?category=electronics
All products
Men
Electronics
Women
Jewelery
SSD PLUS
SP
SO URL lo electronic ante only electronic items matarme dispaly avvali men ante men details in url so on...
787, 437px
1354 × 599px
100%
Type here to search
30°C Haze
ENG
16:43
23-10-2024

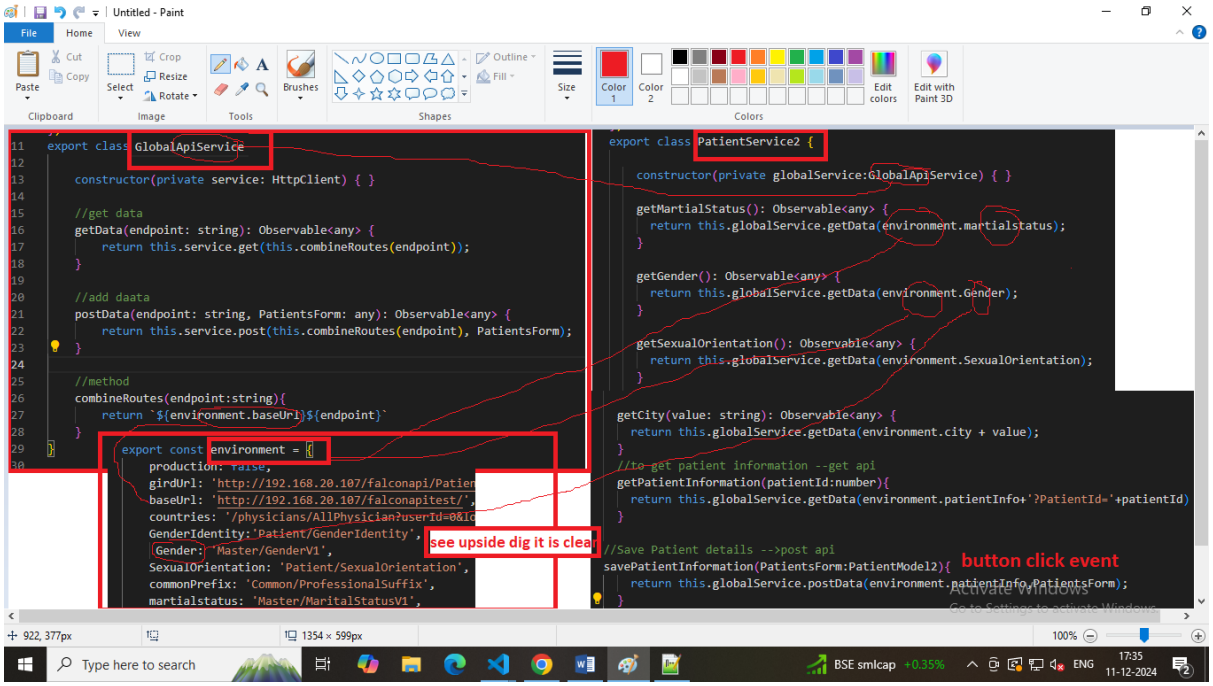


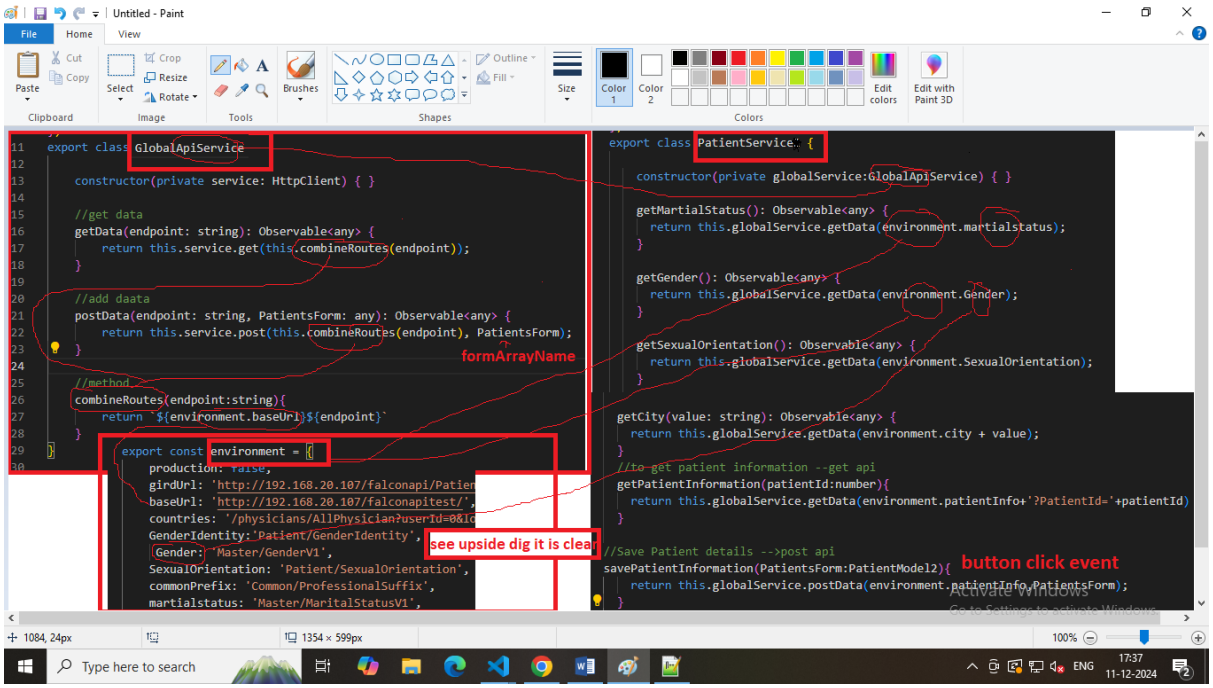


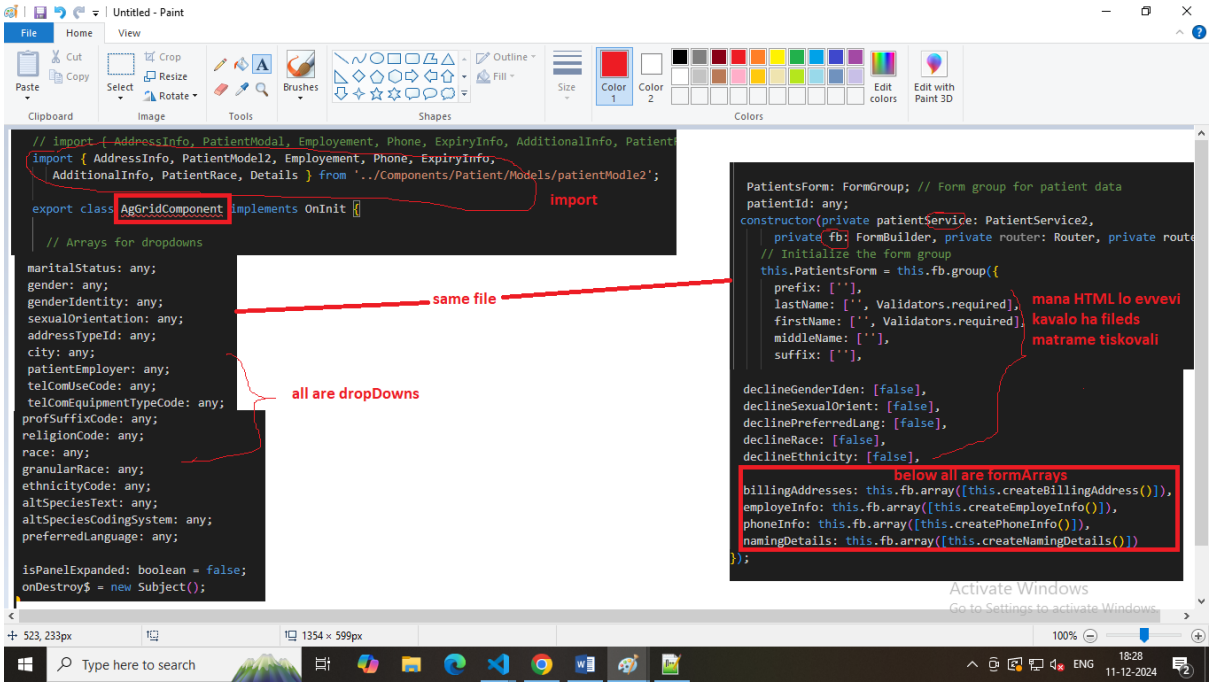
Falkon project

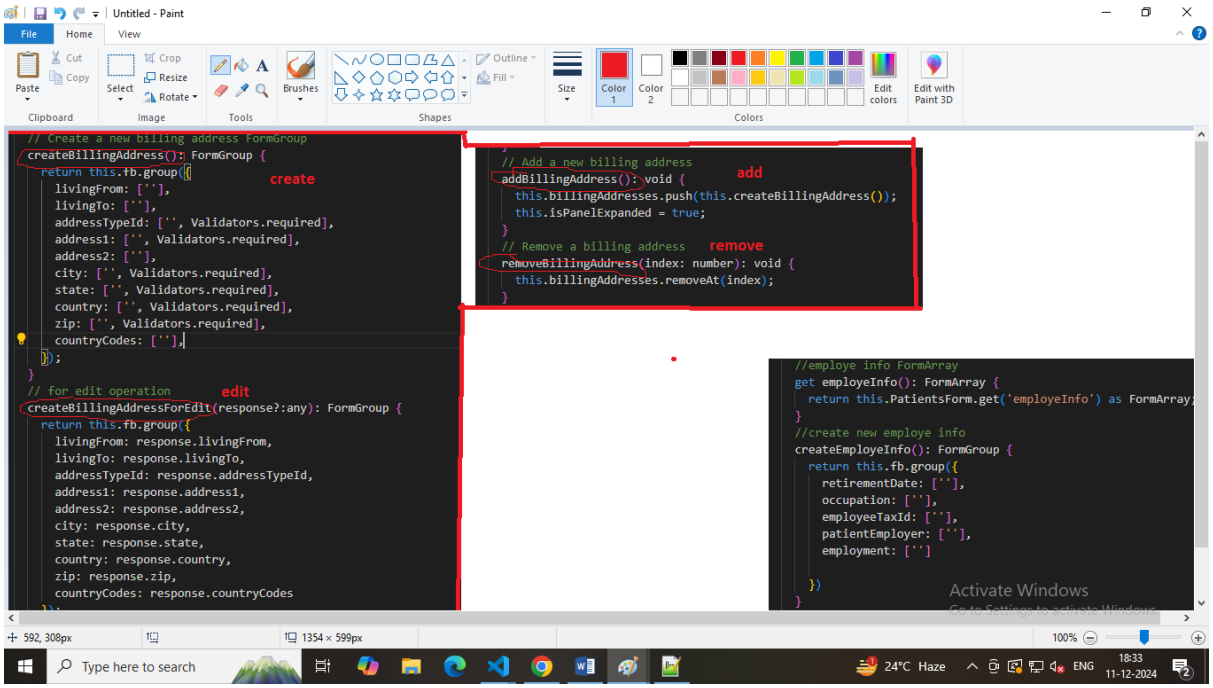

Untitled - Paint
export const environment = {
production: false,
girdUrl: 'http://192.168.20.107/falconapi/Patient/NewPatientDashboard?LocationId=1&Pract
baseUrl: 'http://192.168.20.107/falconapitest/',
countries: '/physicians/AllPhysician?userId=0&locationId=0&searchType=0&phyName=null',
GenderIdentity:'Patient/GenderIdentity',
Gender: 'Master/GenderV1',
SexualOrientation: 'Patient/SexualOrientation',
commonPrefix: 'Common/ProfessionalSuffix',
martialstatus: 'Master/MaritalStatusV1',
suffix: 'Common/ProfessionalSuffix',
AddressType: 'Master/AddressTypeV1',
Employeer: 'Common/Employer',
telecomusecode: 'Common/TelecomUseCode',
telecomEquipmentType: 'Common/TelecomEquipmentType',
professionalsuffix: 'Common/ProfessionalSuffix',
race: 'Master/RaceV1',
ethnicity: 'Master/EthnicityV1',
animal: 'Master/Animal_CDC?conceptname=',
AltspiecesCode: 'Master/Animal_CDC?conceptname=',
phoneTypes: 'Patient/PhoneTypes',
religions: 'Master/ReligionV1',
patientDeails: 'Patient',
religion: 'Master/ReligionV1',
prefferedlang: 'Master/Language',
SpeciesCode: 'Master/Animal_CDC?conceptname=',
granualrace: 'Master/RaceV1',
city: 'Master/ZipCodesByCity?city=',
"employmentInfo": [
"patientEmploymentId": 41175,
"employerId": 0,
"patientId": 42416,
"enteredBy": 1182,
"employment": null,
"employeeTaxId": null,
"retirementDate": "2024-12-18",
"occupation": null,
"patientEmployer": ""
"phoneInfo": [
"patientId": 42416,
"IsMandatroyFiledsFilled": true,
"lastName": "RDHG",
"prefix": "",
"firstName": "GFHDFGJ",
"middleName": "",
"suffix": "",
"dob": "12/05/2024 18:30",
"gender": "1",
"mrn": "",
"active": 0,
"locationId": 1,
"practiceId": 1,
"isExpired": false,
"draft": false,
"registrationType": "",
"mergePatientId": 0,
"emailVerified": false,
"phoneVerified": false,
"multipleBirths": false,
"birthOrder": 5,
API url
"additionalInfo": [
dropDowns
"patientId": 42416,
"patientAddlId": 41155,
"alternateCellNumber": "",
"consent": "false",
"religionCode": "",
"isInsured": false,
"sms": false,
"enteredBy": 1182,
"exemptReporting": false,
"profSuffixCode": "",
"familyDoctorName": "",
"communicationpreference": ""
"expiryInfo": [
"patientExpId": 41145,
"patientId": 42416,
"dateOfDeath": null,
"preliminaryCauseOfDeath": "",
"deathCode": "",
"deathCodeType": "",
"enteredBy": 1182
Activate Windows
653, 226px
1354 × 618px
100%
Type here to search
28°C Haze
ENG
17:23
11-12-2024

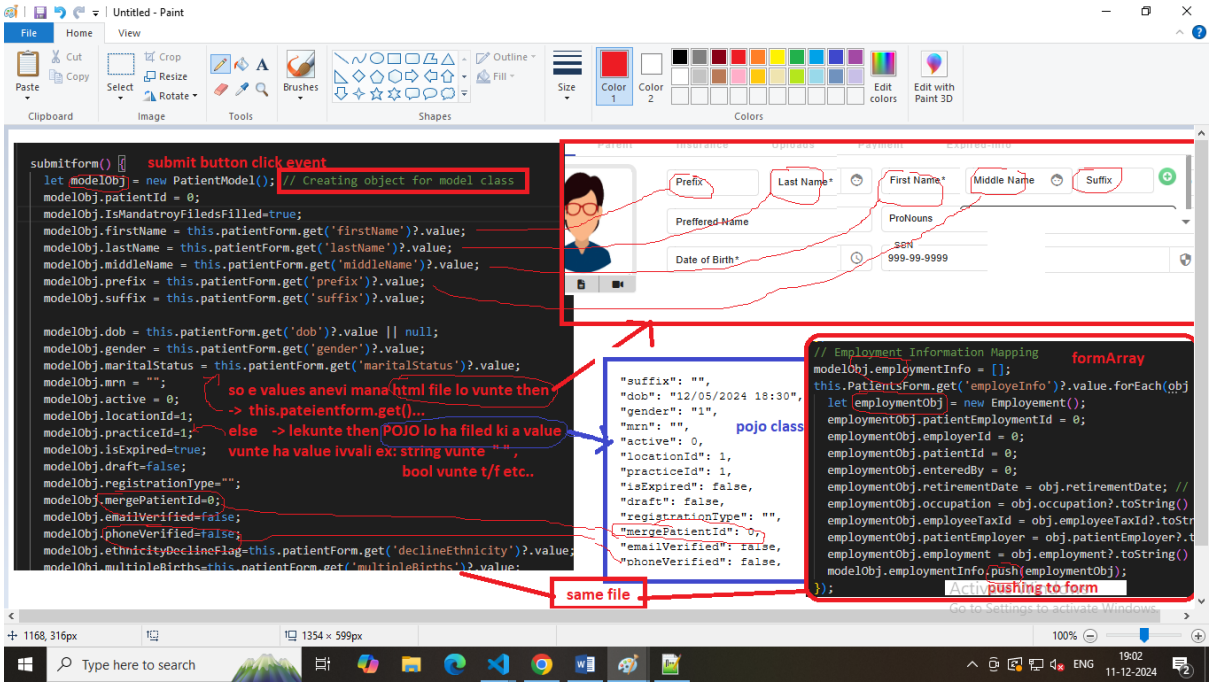

Untitled - Paint
export const environment = {
production: false,
girdUrl: 'http://192.168.20.107/falconapi/Patient/NewPatientDashboard?LocationId=1&Pract
baseUrl: 'http://192.168.20.107/falconapitest/',
countries: '/physicians/AllPhysician?userId=0&locationId=0&searchType=0&phyName=null',
GenderIdentity:'Patient/GenderIdentity',
Gender: 'Master/GenderV1',
SexualOrientation: 'Patient/SexualOrientation',
commonPrefix: 'Common/ProfessionalSuffix',
martialstatus: 'Master/MaritalStatusV1',
suffix: 'Common/ProfessionalSuffix',
AddressType: 'Master/AddressTypeV1',
Employeer: 'Common/Employer',
telecomusecode: 'Common/TelecomUseCode',
telecomEquipmentType: 'Common/TelecomEquipmentType',
professionalsuffix: 'Common/ProfessionalSuffix',
race: 'Master/RaceV1',
ethnicity: 'Master/EthnicityV1',
animal: 'Master/Animal_CDC?conceptname=',
AltspiecesCode: 'Master/Animal_CDC?conceptname=',
phoneTypes: 'Patient/PhoneTypes',
religions: 'Master/ReligionV1',
patientDeails: 'Patient',
religion: 'Master/ReligionV1',
prefferedlang: 'Master/Language',
SpeciesCode: 'Master/Animal_CDC?conceptname=',
granualrace: 'Master/RaceV1',
city: 'Master/ZipCodesByCity?city=',
export class PatientModel2 {
patientId: number;
IsMandatroyFiledsFilled: boolean;
lastName: string;
prefix: string;
firstName: string;
middleName: string;
suffix: string;
dob: string;//str
gender: string;
mrn: string;
active: number;
locationId: number;
additionalInfo: AdditionalInfo[];
alternateCellNumber:[];
expiryInfo: ExpiryInfo[];
patientRaces: PatientRace[];
addressInfo: AddressInfo[];
otherNamesInfo: Details[];
employmentInfo: Employement[];
phoneInfo: Phone[];
model class
export class ExpiryInfo {
patientExpId: number;
patientId: number;
dateOfDeath: string;//s
preliminaryCauseOfDeath: string;
deathCode: string;
deathCodeType: string;
enteredBy: number;
formArrays so take empty emtyArray
export class PatientRace {
raceId: number;
patientid: number;
enteredBy: number;
raceCode: string;//s
race: string;
raceType: string;
granulaRace: string;
granulaRaceCode: string;
839, 191px
1354 × 599px
100%
Type here to search
27°C Haze
ENG
17:51
11-12-2024


Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
export class GlobalApiService
constructor(private service: HttpClient) { }
//get data
getData(endpoint: string): Observable<any> {
return this.service.get(this.combineRoutes(endpoint));
}
//add daata
postData(endpoint: string, PatientsForm: any): Observable<any> {
return this.service.post(this.combineRoutes(endpoint), PatientsForm);
}
//method
combineRoutes(endpoint:string){
return `${environment.baseUrl}${endpoint}`
}
export const environment = {
production: false,
girdUrl: 'http://192.168.20.107/falconapi/Patien
baseUrl: 'http://192.168.20.107/falconapitest/',
countries: '/physicians/AllPhysician?userId=0&IG
GenderIdentity:'Patient/GenderIdentity',
Gender: 'Master/GenderV1',
SexualOrientation: 'Patient/SexualOrientation',
commonPrefix: 'Common/ProfessionalSuffix',
martialstatus: 'Master/MaritalStatusV1',
see upside dig it is clear
export class PatientService2 {
constructor(private globalService:GlobalApiService) { }
getMartialStatus(): Observable<any> {
return this.globalService.getData(environment.martialstatus);
}
getGender(): Observable<any> {
return this.globalService.getData(environment.Gender);
}
getSexualOrientation(): Observable<any> {
return this.globalService.getData(environment.SexualOrientation);
}
getCity(value: string): Observable<any> {
return this.globalService.getData(environment.city + value);
}
//to get patient information --get api
getPatientInformation(patientId:number){
return this.globalService.getData(environment.patientInfo+'?PatientId='+patientId)
}
//Save Patient details -->post api
savePatientInformation(PatientsForm:PatientModel2){
return this.globalService.postData(environment.patientInfo,PatientsForm);
}
button click event
922, 377px
1354 × 599px
100%
Type here to search
BSE smlcap +0.35%
ENG
17:35
11-12-2024

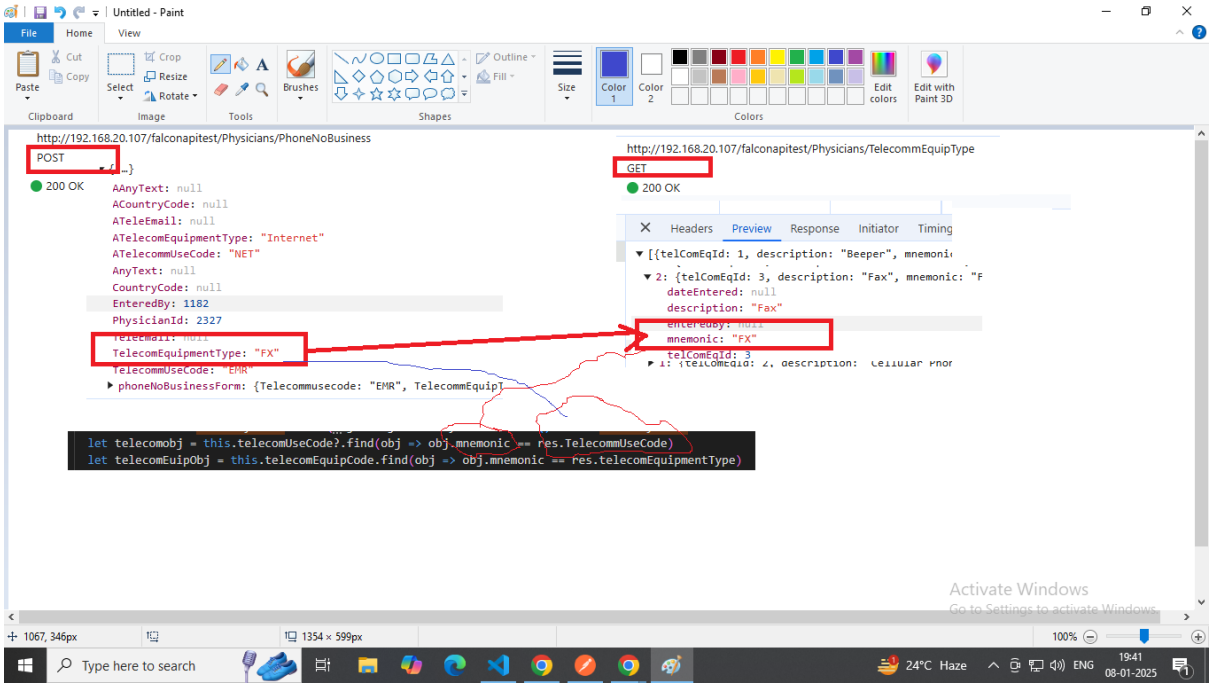

Untitled - Paint
File
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Edit with Paint 3D
Clipboard
Image
Tools
Shapes
Colors
export class GlobalApiService
constructor(private service: HttpClient) { }
//get data
getData(endpoint: string): Observable<any> {
return this.service.get(this.combineRoutes(endpoint));
}
//add daata
postData(endpoint: string, PatientsForm: any): Observable<any> {
return this.service.post(this.combineRoutes(endpoint), PatientsForm);
}
formArrayName
//method
combineRoutes(endpoint:string){
return `${environment.baseUrl}${endpoint}`
}
export const environment = {
production: false,
girdUrl: 'http://192.168.20.107/falconapi/Patien
baseUrl: 'http://192.168.20.107/falconapitest/',
countries: '/physicians/AllPhysician?userId=0&IG
GenderIdentity:'Patient/GenderIdentity',
Gender: 'Master/GenderV1',
SexualOrientation: 'Patient/SexualOrientation',
commonPrefix: 'Common/ProfessionalSuffix',
martialstatus: 'Master/MaritalStatusV1',
see upside dig it is clear
export class PatientService {
constructor(private globalService:GlobalApiService) { }
getMartialStatus(): Observable<any> {
return this.globalService.getData(environment.martialstatus);
}
getGender(): Observable<any> {
return this.globalService.getData(environment.Gender);
}
getSexualOrientation(): Observable<any> {
return this.globalService.getData(environment.SexualOrientation);
}
getCity(value: string): Observable<any> {
return this.globalService.getData(environment.city + value);
}
//to get patient information --get api
getPatientInformation(patientId:number){
return this.globalService.getData(environment.patientInfo+'?PatientId='+patientId)
}
//Save Patient details -->post api
savePatientInformation(PatientsForm:PatientModel2){
return this.globalService.postData(environment.patientInfo,PatientsForm);
}
button click event
1084, 24px
1354 × 599px
100%
Type here to search
ENG
17:37
11-12-2024

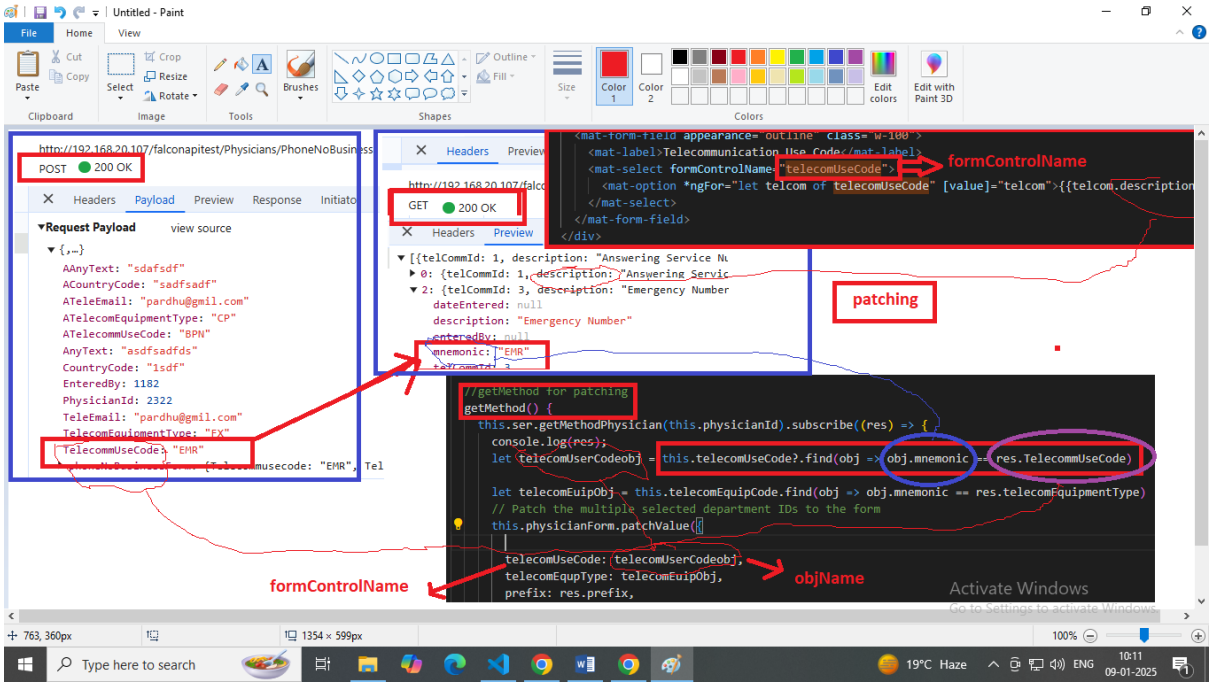

Untitled - Paint
// import { AddressInfo, PatientModal, Employement, Phone, ExpiryInfo, AdditionalInfo, Patient
import { AddressInfo, PatientModel2, Employement, Phone, ExpiryInfo,
AdditionalInfo, PatientRace, Details } from '../Components/Patient/Models/patientModle2';
import
export class AgGridComponent implements OnInit {
// Arrays for dropdowns
maritalStatus: any;
gender: any;
genderIdentity: any;
sexualOrientation: any;
addressTypeId: any;
city: any;
patientEmployer: any;
telComUseCode: any;
telComEquipmentTypeCode: any;
profSuffixCode: any;
religionCode: any;
race: any;
granularRace: any;
ethnicityCode: any;
altSpeciesText: any;
altSpeciesCodingSystem: any;
preferredLanguage: any;
isPanelExpanded: boolean = false;
onDestroy$ = new Subject();
same file
all are dropDowns
PatientsForm: FormGroup; // Form group for patient data
patientId: any;
constructor(private patientService: PatientService2,
private fb: FormBuilder, private router: Router, private route
// Initialize the form group
this.PatientsForm = this.fb.group({
prefix: [''],
lastName: ['', Validators.required],
firstName: ['', Validators.required],
middleName: [''],
suffix: [''],
mana HTML lo evvevi
kavalo ha fileds
matrame tiskovali
declineGenderIden: [false],
declineSexualOrient: [false],
declinePreferredLang: [false],
declineRace: [false],
declineEthnicity: [false],
below all are formArrays
billingAddresses: this.fb.array([this.createBillingAddress()]),
employeInfo: this.fb.array([this.createEmployeInfo()]),
phoneInfo: this.fb.array([this.createPhoneInfo()]),
namingDetails: this.fb.array([this.createNamingDetails()])
});
Activate Windows
Type here to search
ENG
18:28
11-12-2024

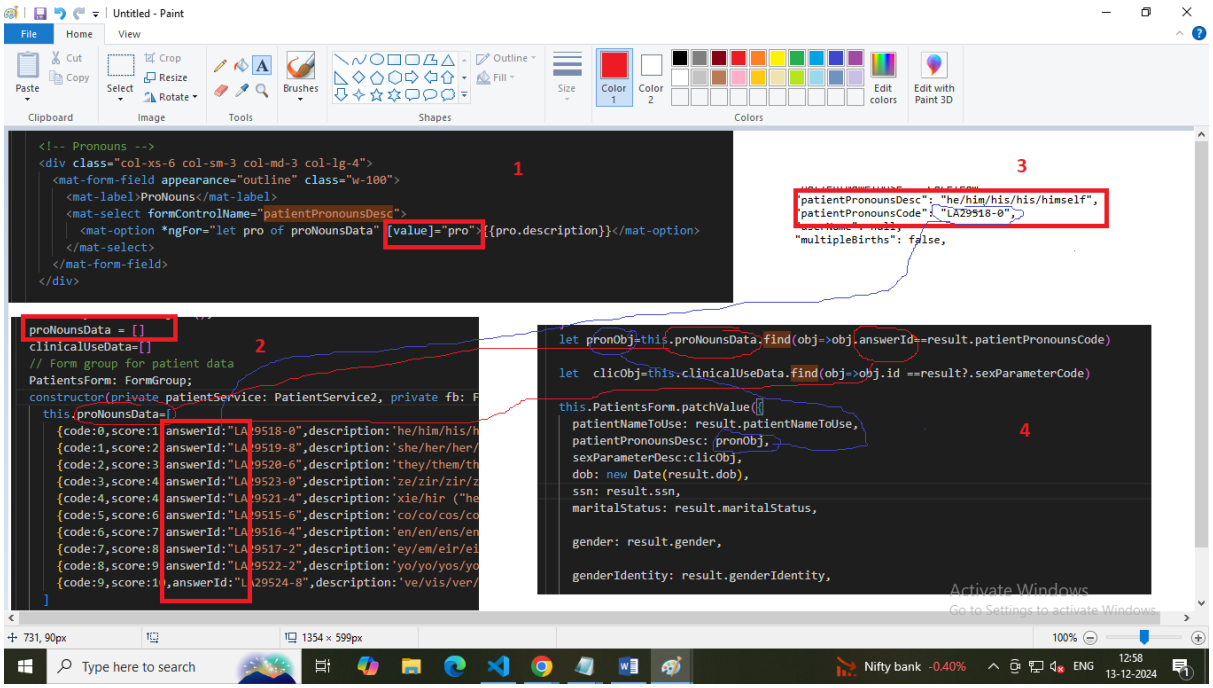

Untitled - Paint
// Create a new billing address FormGroup
createBillingAddress(): FormGroup {
return this.fb.group({
create
livingFrom: [''],
livingTo: [''],
addressTypeId: ['', Validators.required],
address1: ['', Validators.required],
address2: [''],
city: ['', Validators.required],
state: ['', Validators.required],
country: ['', Validators.required],
zip: ['', Validators.required],
countryCodes: [''],
});
}
// for edit operation
edit
createBillingAddressForEdit(response?:any): FormGroup {
return this.fb.group({
livingFrom: response.livingFrom,
livingTo: response.livingTo,
addressTypeId: response.addressTypeId,
address1: response.address1,
address2: response.address2,
city: response.city,
state: response.state,
country: response.country,
zip: response.zip,
countryCodes: response.countryCodes
// Add a new billing address
addBillingAddress(): void {
this.billingAddresses.push(this.createBillingAddress());
this.isPanelExpanded = true;
add
// Remove a billing address
remove
removeBillingAddress(index: number): void {
this.billingAddresses.removeAt(index);
//employe info FormArray
get employeInfo(): FormArray {
return this.PatientsForm.get('employeInfo') as FormArray;
//create new employe info
createEmployeInfo(): FormGroup {
return this.fb.group({
retirementDate: [''],
occupation: [''],
employeeTaxId: [''],
patientEmployer: [''],
employment: ['']
Activate Windows
Type here to search
24°C Haze
ENG
18:33
11-12-2024

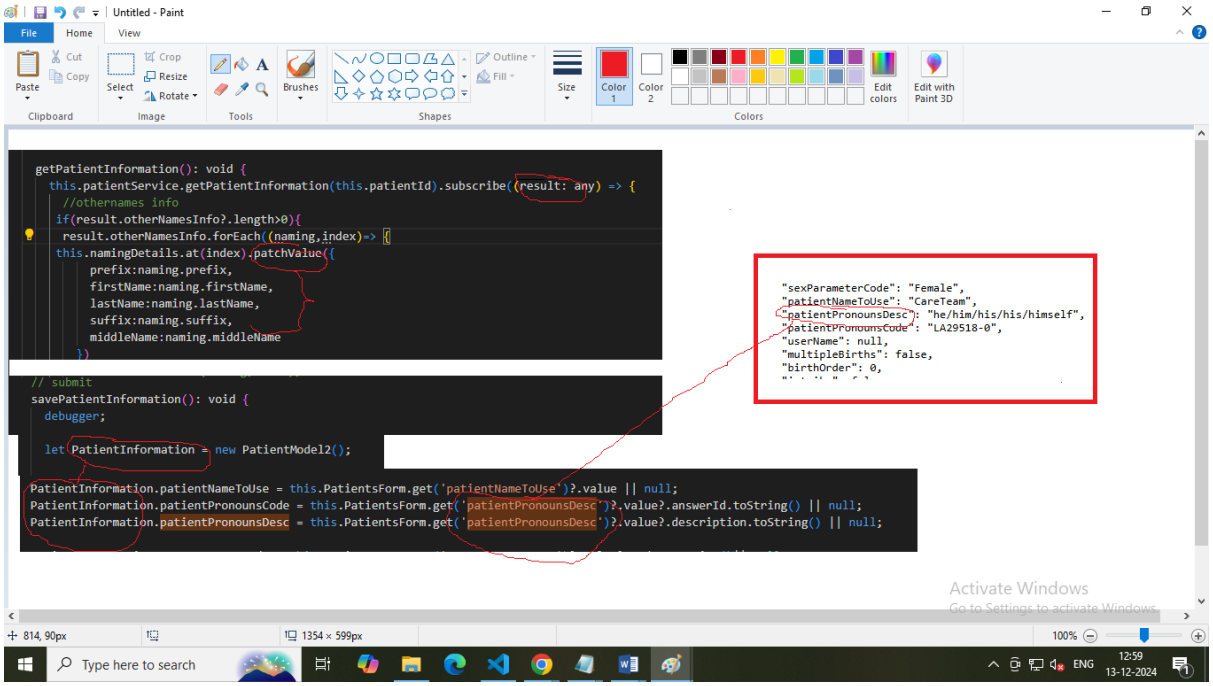

Untitled - Paint
submitform() {
submit button click event
let modelObj = new PatientModel(); // Creating object for model class
modelObj.patientId = 0;
modelObj.IsMandatroyFiledsFilled=true;
modelObj.firstName = this.patientForm.get('firstName')?.value;
modelObj.lastName = this.patientForm.get('lastName')?.value;
modelObj.middleName = this.patientForm.get('middleName')?.value;
modelObj.prefix = this.patientForm.get('prefix')?.value;
modelObj.suffix = this.patientForm.get('suffix')?.value;
modelObj.dob = this.patientForm.get('dob')?.value || null;
modelObj.gender = this.patientForm.get('gender')?.value;
modelObj.maritalStatus = this.patientForm.get('maritalStatus')?.value;
modelObj.mrn = "";
modelObj.active = 0;
modelObj.locationId=1;
modelObj.practiceId=1;
modelObj.isExpired=true;
modelObj.draft=false;
modelObj.registrationType="";
modelObj.mergePatientId=0;
modelObj.emailVerified=false;
modelObj.phoneVerified=false;
modelObj.ethnicityDeclineFlag=this.patientForm.get('declineEthnicity')?.value;
so e values anevi mana html file lo vunte then -> this.pateientform.get()...
else -> lekunte then POJO lo ha filed ki a value vunte ha value ivvali ex: string vunte " ", bool vunte t/f etc..
"suffix": "",
"dob": "12/05/2024 18:30",
"gender": "1",
"mrn": "",
"active": 0,
"locationId": 1,
"practiceId": 1,
"isExpired": false,
"draft": false,
"registrationType": "",
"mergePatientId": 0,
"emailVerified": false,
"phoneVerified": false,
pojo class
same file
Prefix
Last Name*
First Name*
Middle Name
Suffix
Preffered Name
ProNouns
Date of Birth*
SSN
999-99-9999
// Employment Information Mapping
formArray
modelObj.employmentInfo = [];
this.PatientsForm.get('employeInfo')?.value.forEach(obj
let employmentObj = new Employement();
employmentObj.patientEmploymentId = 0;
employmentObj.employerId = 0;
employmentObj.patientId = 0;
employmentObj.enteredBy = 0;
employmentObj.retirementDate = obj.retirementDate; //
employmentObj.occupation = obj.occupation?.toString()
employmentObj.employeeTaxId = obj.employeeTaxId?.toStr
employmentObj.patientEmployer = obj.patientEmployer?.t
employmentObj.employment = obj.employment?.toString()
modelObj.employmentInfo.push(employmentObj);
});
pushing to form
1168, 316px
1354 × 599px
100%
Type here to search
ENG
19:02
11-12-2024

patching


Untitled - Paint
http://192.168.20.107/falconapitest/Physicians/PhoneNoBusiness
POST
200 OK
AAnyText: null
ACountryCode: null
ATeleEmail: null
ATelecomEquipmentType: "Internet"
ATelecommUseCode: "NET"
AnyText: null
CountryCode: null
EnteredBy: 1182
PhysicianId: 2327
TeleEmail: null
TelecomEquipmentType: "FX"
TelecommUseCode: "EMR"
phoneNoBusinessForm: {Telecommusecode: "EMR", TelecommEquipT
http://192.168.20.107/falconapitest/Physicians/TelecommEquipType
GET
200 OK
Headers
Preview
Response
Initiator
Timing
[{telComEqId: 1, description: "Beeper", mnemonic
2: {telComEqId: 3, description: "Fax", mnemonic: "F
dateEntered: null
description: "Fax"
enteredBy: null
mnemonic: "FX"
telComEqId: 3
let telecomobj = this.telecomUseCode?.find(obj => obj.mnemonic == res.TelecommUseCode)
let telecomEuipObj = this.telecomEquipCode.find(obj => obj.mnemonic == res.telecomEquipmentType)
1067, 346px
1354 × 599px
100%
Type here to search
24°C Haze
ENG
19:41
08-01-2025

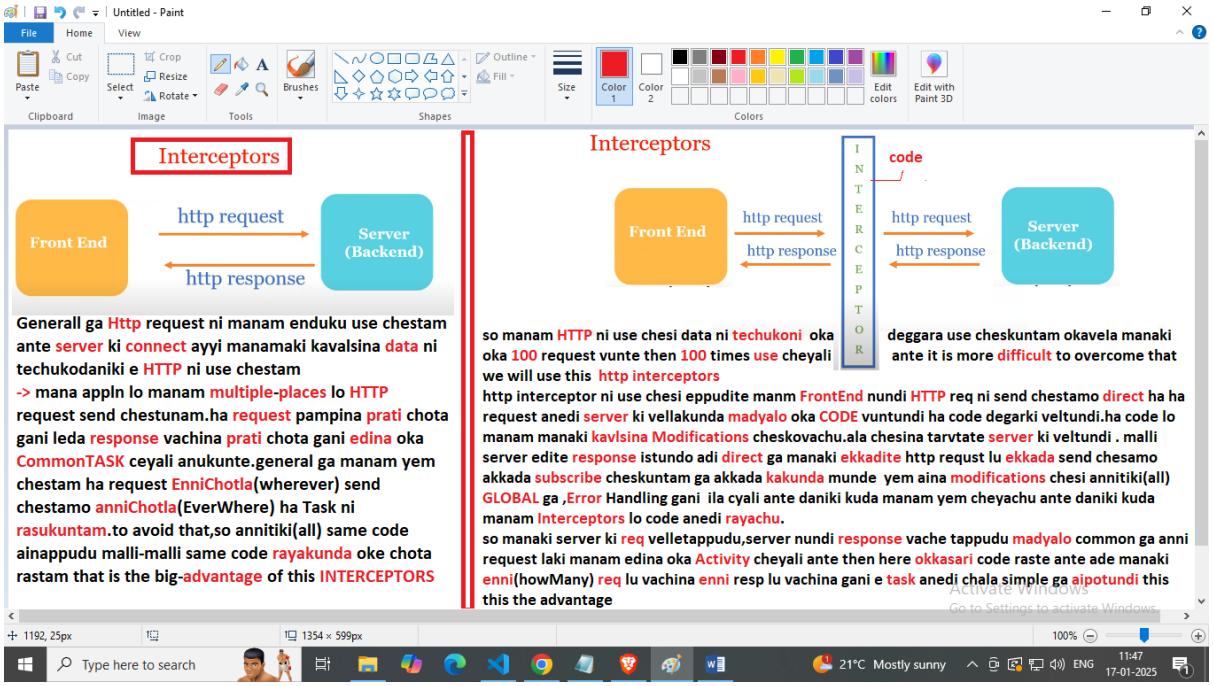

Untitled - Paint
POST 200 OK
Request Payload
TelecommUseCode: "EMR"
GET 200 OK
mnemonic: "EMR"
formControlName
patching
//getMethod for patching
getMethod() {
this.ser.getMethodPhysician(this.physicianId).subscribe((res) => {
let telecomUserCodeobj = this.telecomUseCode?.find(obj => obj.mnemonic == res.TelecommUseCode)
this.physicianForm.patchValue({
telecomUseCode: telecomUserCodeobj,
telecomEqupType: telecomEuipObj,
prefix: res.prefix,
formControlName
objName
Activate Windows
763, 360px
1354 × 599px
100%
Type here to search
19°C Haze
ENG
10:11
09-01-2025


Untitled - Paint
<!-- Pronouns -->
<mat-label>ProNouns</mat-label>
<mat-select formControlName="patientPronounsDesc">
<mat-option *ngFor="let pro of proNounsData" [value]="pro">{{pro.description}}</mat-option>
1
2
3
4
proNounsData = []
clinicalUseData=[]
// Form group for patient data
PatientsForm: FormGroup;
"patientPronounsDesc": "he/him/his/his/himself",
"patientPronounsCode": "LA29518-0",
"multipleBirths": false,
let pronObj=this.proNounsData.find(obj=>obj.answerId==result.patientPronounsCode)
let clicObj=this.clinicalUseData.find(obj=>obj.id ==result?.sexParameterCode)
this.PatientsForm.patchValue({
patientNameToUse: result.patientNameToUse,
patientPronounsDesc: pronObj,
sexParameterDesc:clicObj,
dob: new Date(result.dob),
ssn: result.ssn,
maritalStatus: result.maritalStatus,
gender: result.gender,
genderIdentity: result.genderIdentity,
Activate Windows
731, 90px
1354 × 599px
100%
Type here to search
Nifty bank -0.40%
ENG
12:58
13-12-2024



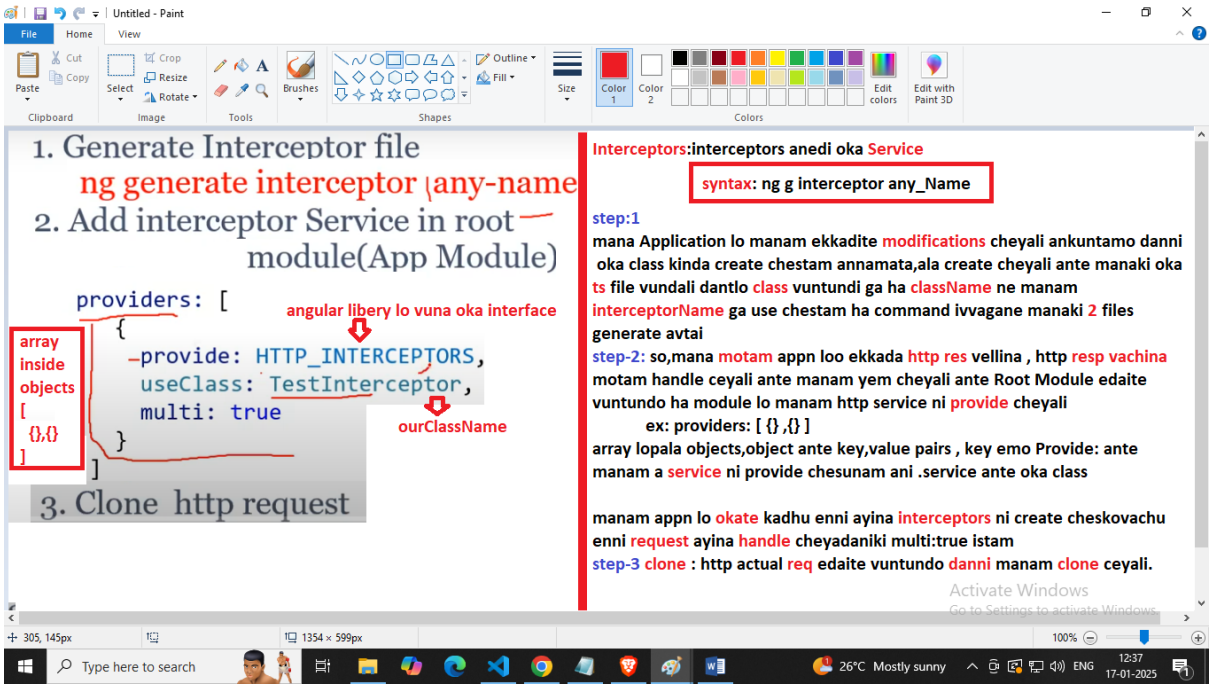
interceptors by procodemy

Interceptors by css world



Generall ga Http request ni manam enduku use chestam ante server ki connect ayyi manamaki kavalsina data ni techukodaniki e HTTP ni use chestam
-> mana appln lo manam multiple-places lo HTTP request send chestunam.ha request pampina prati chota gani leda response vachina prati chota gani edina oka CommonTASK ceyali anukunte.general ga manam yem chestam ha request EnniChotla(wherever) send chestamo anniChotla(EverWhere) ha Task ni rasukuntam.to avoid that,so annitiki(all) same code ainappudu malli-malli same code rayakunda oke chota rastam that is the big-advantage of this INTERCEPTORS

so manam HTTP ni use chesi data ni techukoni oka oka 100 request vunte then 100 times use cheyali we will use this http interceptors deggara use cheskuntam okavela manaki ante it is more difficult to overcome that
http interceptor ni use chesi eppudite manm FrontEnd nundi HTTP req ni send chestamo direct ha ha request anedi server ki vellakunda madyalo oka CODE vuntundi ha code degarki veltundi.ha code lo manam manaki kavlsina Modifications cheskovachu.ala chesina tarvtate server ki veltundi . malli server edite response istundo adi direct ga manaki ekkadite http requst lu ekkada send chesamo akkada subscribe cheskuntam ga akkada kakunda munde yem aina modifications chesi annitiki(all) GLOBAL ga ,Error Handling gani ila cyali ante daniki kuda manam yem cheyachu ante daniki kuda manam Interceptors lo code anedi rayachu.
so manaki server ki req velletappudu,server nundi response vache tappudu madyalo common ga anni request laki manam edina oka Activity cheyali ante then here okkasari code raste ante ade manaki enni(howMany) req lu vachina enni resp lu vachina gani e task anedi chala simple ga aipotundi this this the advantage

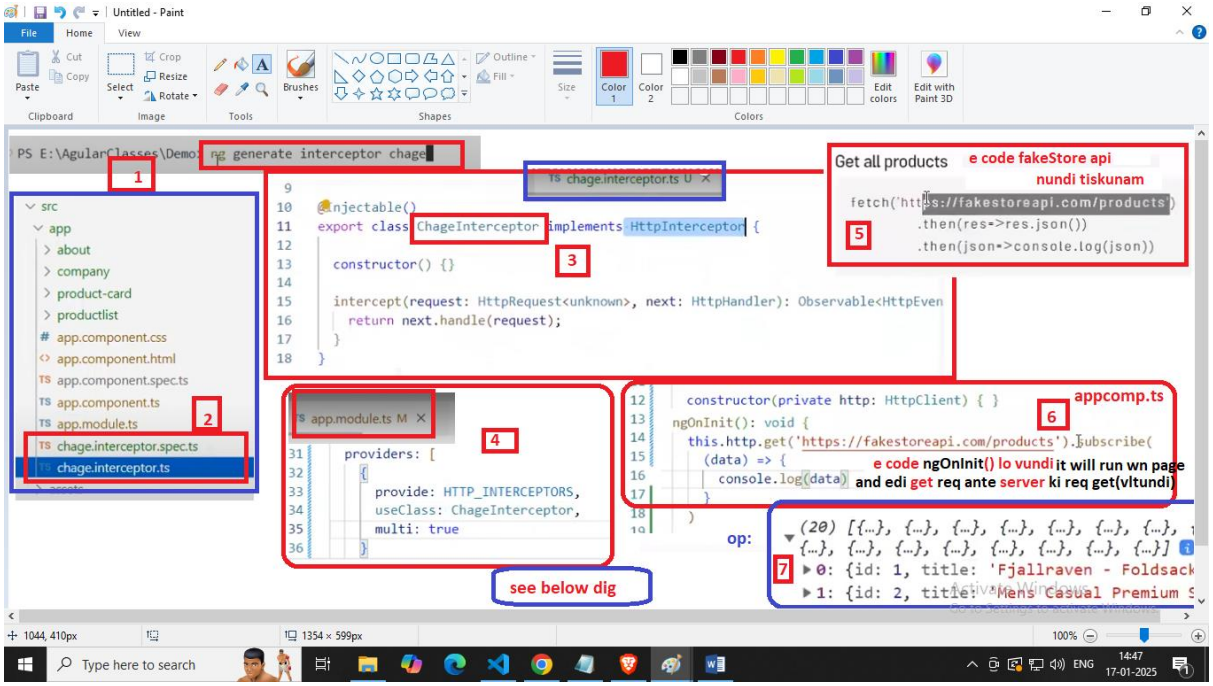


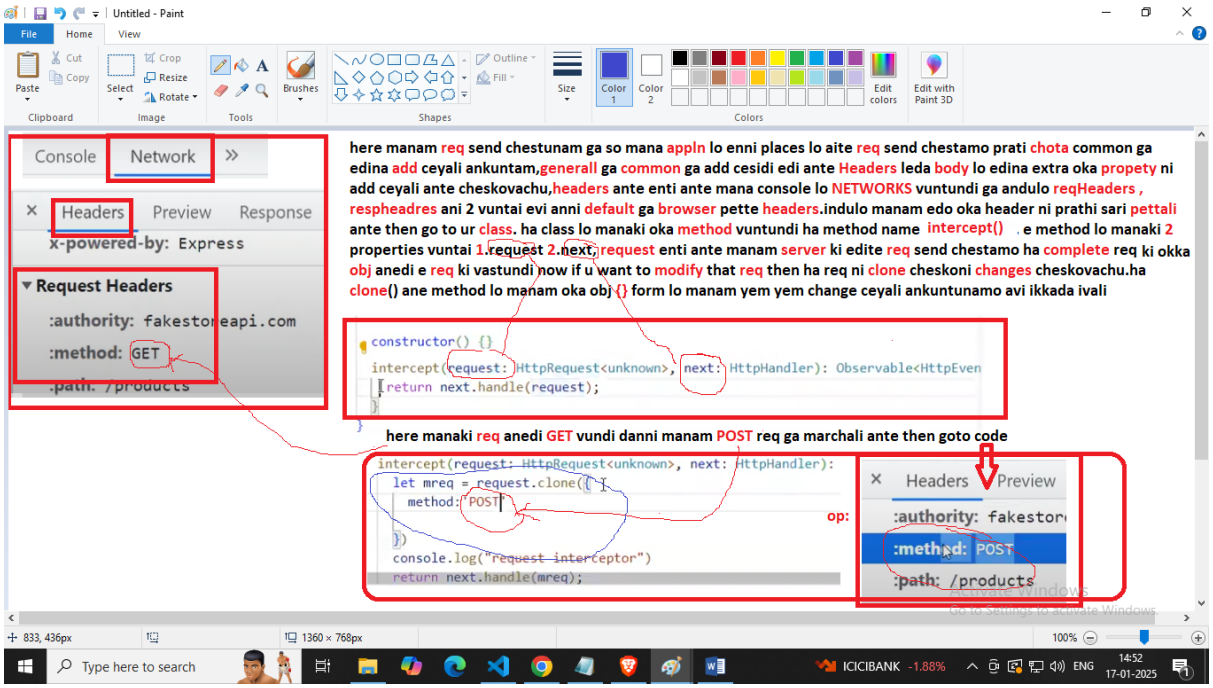
Interceptors:interceptors anedi oka Service

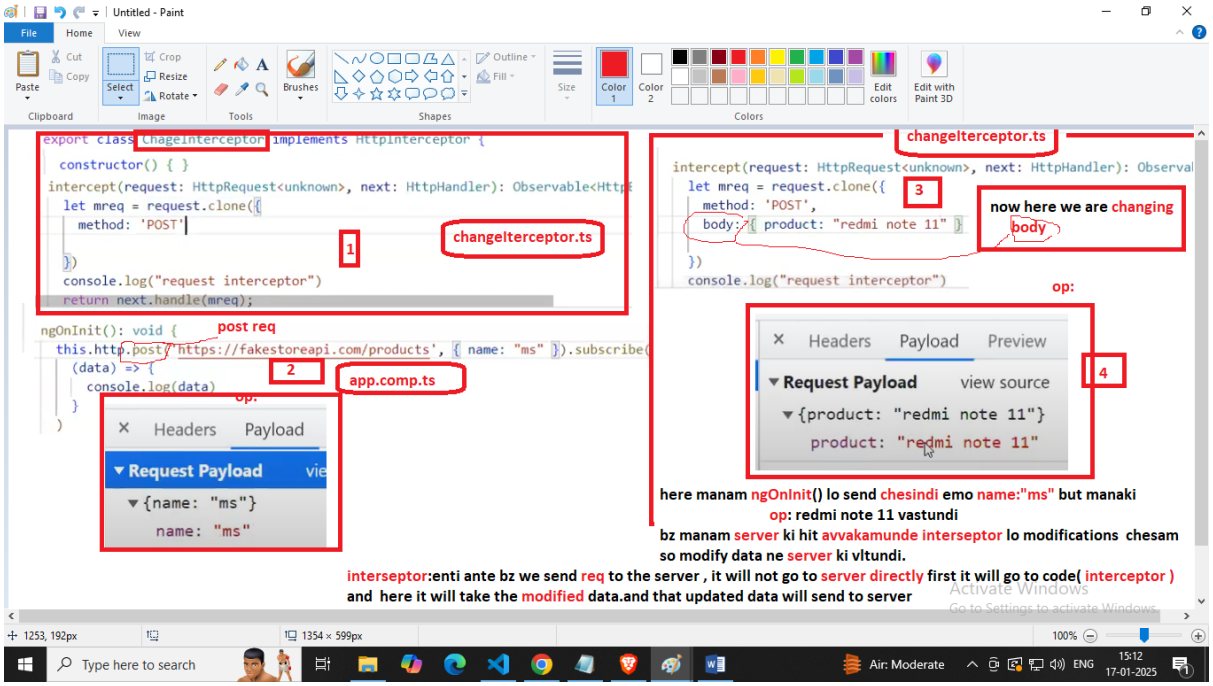
syntax: ng g interceptor any_Name

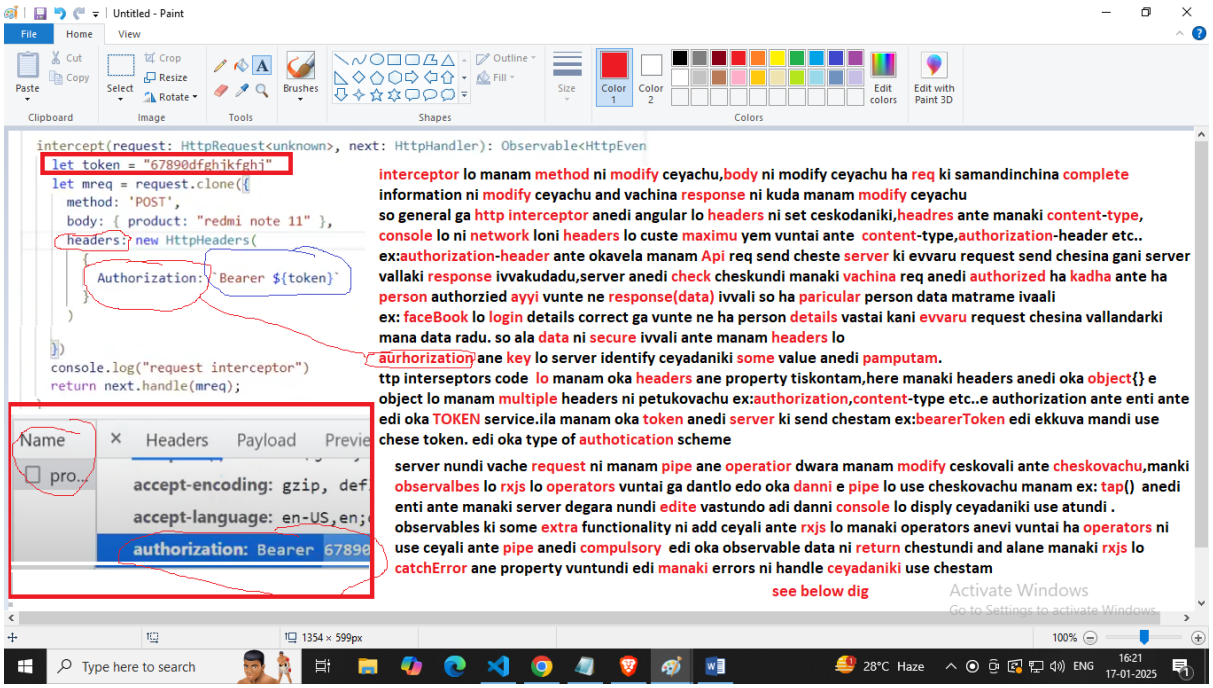
step:1
mana Application lo manam ekkadite modifications cheyali ankuntamo danni oka class kinda create chestam annamata,ala create cheyali ante manaki oka ts file vundali dantlo class vuntundi ga ha className ne manam interceptorName ga use chestam ha command ivvagane manaki 2 files generate avtai
step-2: so,mana motam appn loo ekkada http res vellina , http resp vachina motam handle ceyali ante manam yem cheyali ante Root Module edaite vuntundo ha module lo manam http service ni provide cheyali
ex: providers: [{} ,{}]
array lopala objects,object ante key,value pairs , key emo Provide: ante manam a service ni provide chesunam ani .service ante oka class

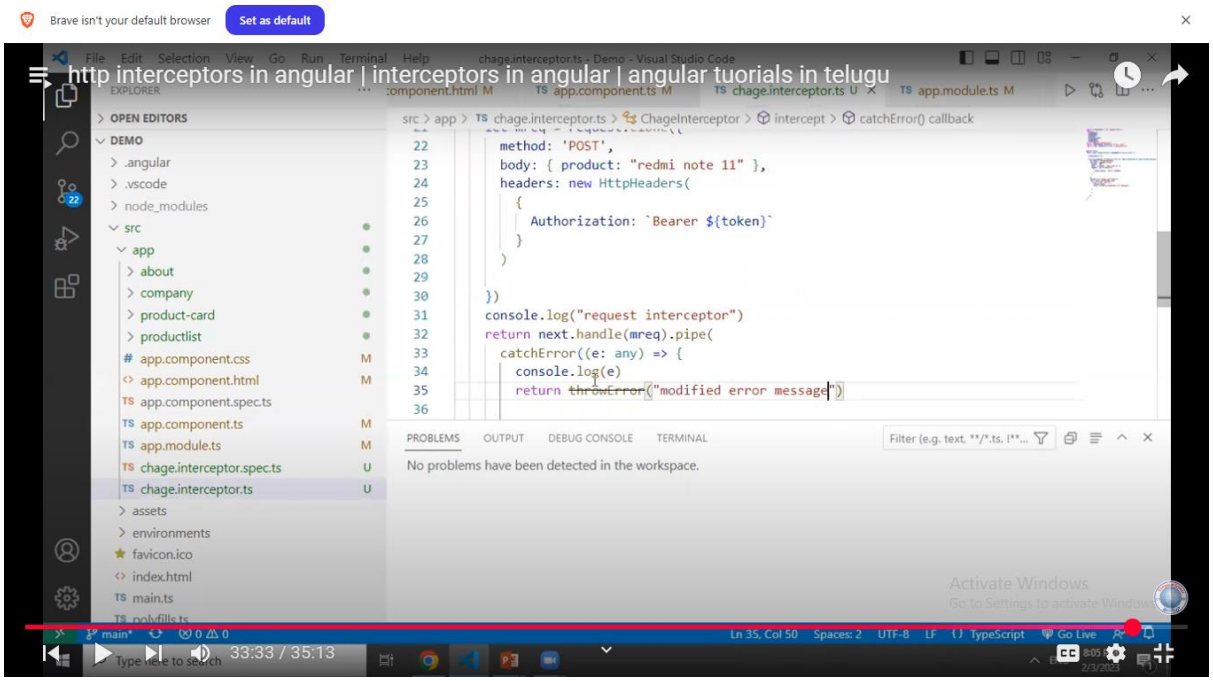
manam appn lo okate kadhu enni ayina interceptors ni create cheskovachu enni request ayina handle cheyadaniki multi:true istam
step-3 clone : http actual req edaite vuntundo danni manam clone ceyali.


Untitled - Paint
PS E:\AgularClasses\Demo> ng generate interceptor chage
1
src
app
about
company
product-card
productlist
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.module.ts
chage.interceptor.spec.ts
chage.interceptor.ts
2
chage.interceptor.ts U
@Injectable()
export class ChageInterceptor implements HttpInterceptor {
constructor() {}
intercept(request: HttpRequest<unknown>, next: HttpHandler): Observable<HttpEven
return next.handle(request);
3
app.module.ts M
providers: [
{
provide: HTTP_INTERCEPTORS,
useClass: ChageInterceptor,
multi: true
}
4
see below dig
Get all products
e code fakeStore api nundi tiskunam
fetch('https://fakestoreapi.com/products')
.then(res=>res.json())
.then(json=>console.log(json))
5
constructor(private http: HttpClient) { }
ngOnInit(): void {
this.http.get('https://fakestoreapi.com/products').subscribe(
(data) => {
console.log(data)
}
)
6
appcomp.ts
e code ngOnInit() lo vundi it will run wn page and edi get req ante server ki req get(vltundi)
op:
(20) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
0: {id: 1, title: 'Fjallraven - Foldsack
1: {id: 2, title: 'Mens Casual Premium S
7
1044, 410px
1354 × 599px
100%
Type here to search
ENG
14:47
17-01-2025


Untitled - Paint
Console
Network
Headers
Preview
Response
x-powered-by: Express
Request Headers
:authority: fakestoreapi.com
:method: GET
:path: /products
here manam req send chestunam ga so mana appln lo enni places lo aite req send chestamo prati chota common ga edina add ceyali ankuntam,generall ga common ga add cesidi edi ante Headers leda body lo edina extra oka propety ni add ceyali ante cheskovachu,headers ante enti ante mana console lo NETWORKS vuntundi ga andulo reqHeaders , respheadres ani 2 vuntai evi anni default ga browser pette headers.indulo manam edo oka header ni prathi sari pettali ante then go to ur class. ha class lo manaki oka method vuntundi ha method name intercept() . e method lo manaki 2 properties vuntai 1.request 2.next. request enti ante manam server ki edite req send chestamo ha complete req ki okka obj anedi e req ki vastundi now if u want to modify that req then ha req ni clone cheskoni changes cheskovachu.ha clone() ane method lo manam oka obj {} form lo manam yem yem change ceyali ankuntunamo avi ikkada ivali
constructor() {}
intercept(request: HttpRequest<unknown>, next: HttpHandler): Observable<HttpEven
return next.handle(request);
here manaki req anedi GET vundi danni manam POST req ga marchali ante then goto code
intercept(request: HttpRequest<unknown>, next: HttpHandler):
let mreq = request.clone({
method: "POST"
})
console.log("request interceptor")
return next.handle(mreq);
op:
Headers
Preview
:authority: fakestor
:method: POST
:path: /products
833, 436px
1360 × 768px
100%
Type here to search
ICICIBANK -1.88%
ENG
14:52
17-01-2025


Untitled - Paint
changeInterceptor.ts
1
changeInterceptor.ts
post req
2
app.comp.ts
op:
Headers
Payload
Request Payload
{name: "ms"}
name: "ms"
changeInterceptor.ts
3
now here we are changing body
op:
Headers
Payload
Preview
Request Payload
view source
{product: "redmi note 11"}
product: "redmi note 11"
4
here manam ngOnInit() lo send chesindi emo name:"ms" but manaki op: redmi note 11 vastundi
bz manam server ki hit avvakamunde interseptor lo modifications chesam so modify data ne server ki vltundi.
interseptor:enti ante bz we send req to the server , it will not go to server directly first it will go to code(interceptor) and here it will take the modified data.and that updated data will send to server


Untitled - Paint
let token = "67890dfghjkfghj"
Authorization: `Bearer ${token}`
Name
Headers
Payload
accept-encoding: gzip, def
accept-language: en-US,en;
authorization: Bearer 67890
interceptor lo manam method ni modify ceyachu,body ni modify ceyachu ha req ki samandinchina complete information ni modify ceyachu and vachina response ni kuda manam modify ceyachu
so general ga http interceptor anedi angular lo headers ni set ceskodaniki,headres ante manaki content-type, console lo ni network loni headers lo custe maximu yem vuntai ante content-type,authorization-header etc..
ex:authorization-header ante okavela manam Api req send cheste server ki evvaru request send chesina gani server vallaki response ivvakudadu,server anedi check cheskundi manaki vachina req anedi authorized ha kadha ante ha person authorzied ayyi vunte ne response(data) ivvali so ha paricular person data matrame ivaali
ex: faceBook lo login details correct ga vunte ne ha person details vastai kani evvaru request chesina vallandarki mana data radu. so ala data ni secure ivvali ante manam headers lo
aurhorization ane key lo server identify ceyadaniki some value anedi pamputam.
ttp interseptors code lo manam oka headers ane property tiskontam,here manaki headers anedi oka object{} e object lo manam multiple headers ni petukovachu ex:authorization,content-type etc..e authorization ante enti ante edi oka TOKEN service.ila manam oka token anedi server ki send chestam ex:bearerToken edi ekkuva mandi use chese token. edi oka type of authotication scheme
server nundi vache request ni manam pipe ane operatior dwara manam modify ceskovali ante cheskovachu,manki observalbes lo rxjs lo operators vuntai ga dantlo edo oka danni e pipe lo use cheskovachu manam ex: tap() anedi enti ante manaki server degara nundi edite vastundo adi danni console lo disply ceyadaniki use atundi .
observables ki some extra functionality ni add ceyali ante rxjs lo manaki operators anevi vuntai ha operators ni use ceyali ante pipe anedi compulsory edi oka observable data ni return chestundi and alane manaki rxjs lo catchError ane property vuntundi edi manaki errors ni handle ceyadaniki use chestam
see below dig



Interceptors theory css world

Generall ga Http request ni manam enduku use chestam ante server ki connect ayyi manamaki kavalsina data ni techukodaniki e HTTP ni use chestam

-> mana appln lo manam multiple-places lo HTTP request send chestunam.ha request pampina prati chota gani leda response vachina prati chota gani edina oka CommonTASK ceyali anukunte.general ga manam yem chestam ha request EnniChotla(wherever) send chestamo anniChotla(EverWhere) ha Task ni rasukuntam.so annitiki(all) same code ainappudu malli malli same code rayakunda oke chota rastam that is the big-advantage of this INTERCEPTORS

ex:see diagram

so manam HTTP ni use chesi data ni techukoni oka deggara use cheskuntam okavela manaki oka 100 request vunte then 100 times use cheyali ante it is more difficult to overcome that we will use this http interceptors

http interceptor ni use chesi eppudite manm FrontEnd nundi HTTP req ni send chestamo direct ha ha request anedi server ki vellakunda madyalo oka CODE vuntundi ha code degarki veltundi.ha code lo manam manaki kavlsina Modifications cheskovachu.ala chesina tarvta server ki veltundi . malli server edite response istundo adi direct ga manaki ekkadite http requst lu ekkada send chesamo akkada subscribe cheskuntam ga akkada kakunda munde yem aina modifications chesi annitiki(all) GLOBAL ga ,Error Handling gani ila cyali ante daniki kuda manam yem cheyachu ante daniki kuda manam Interceptors lo code anedi rayachu.

so manaki server ki req velletappudu,server nundi response vache tappudu madyalo common ga anni request laki manam edina oka Activity cheyali ante then here okkasari code raste adi manaki enni(howMany) req lu vachina enni resp lu vachina gani e task anedi chala simple ga aipotundi this this the advantage

Interceptors:interceptors anedi oka Service

syntax: ng g interceptor any_Name

mana Application lo manam ekkadite modifications cheyali ankuntamo

step:1 oka class kinda create chestam annamata,ala create cheyali ante manaki oka ts file vundali dantlo class vuntundi ga ha className ne manam interceptorName ga use chestam ha command ivvagane manaki 2 files generate avtai

step-3: so,mana motam appn loo ekkada http res vellina , htpp resp vachina motam handle ceyali ante manam yem cheyali Root Module edaite vuntundo ha module lo manam http service ni provide cheyali ex: providers: [{}] array lopala objects,object ante key,value key emo Provide ante manam a service ni provide chesunam.service ante oka class

ex:app confi file

manam appn lo okate kadhu enni ayina interceptors ni create cheskovachu enni request ayina handle cheyadaniki multi:true istam

step-3 clone : http actual req edaite vuntundo danni manam clone ceyali.

here manam req send chestunam ga so mana appln lo enni places lo aite req send chestamo prati chota common ga edina add ceyali ankuntam,generall ga common ga add cesidi edi ante Headers leda body lo edina extra oka propety ni add ceyali ante cheskovachu,headers ante enti ante mana console lo NETWORKS vuntundi ga andulo reqHeaders , respheadres ani 2 vuntai evi anni default ga browser pette headers.indulo manam prathi sare ha header pettali ante then go to ur class. ha class lo manaki oka method vuntundi ha method name interceptor(),indulo manaki 2 properties vuntai 1.request 2.next, request enti ante manam server ki edite req send chestamo ha complete req okka obj anedi e req ki vastundi now if u want to modify that req then ha req ni clone cheskoni change cheskovachu.ha clone() ane method lo manam oka obj {} form lo manam yem yem change ceyali ankuntunamo avi ikkada ivali

interceptor lo manam method ni modify ceyachu,body ni modify ceyachu ha req ki samandinchina complete information ni modify ceyachu vachina response ni kuda manam modify ceyachu

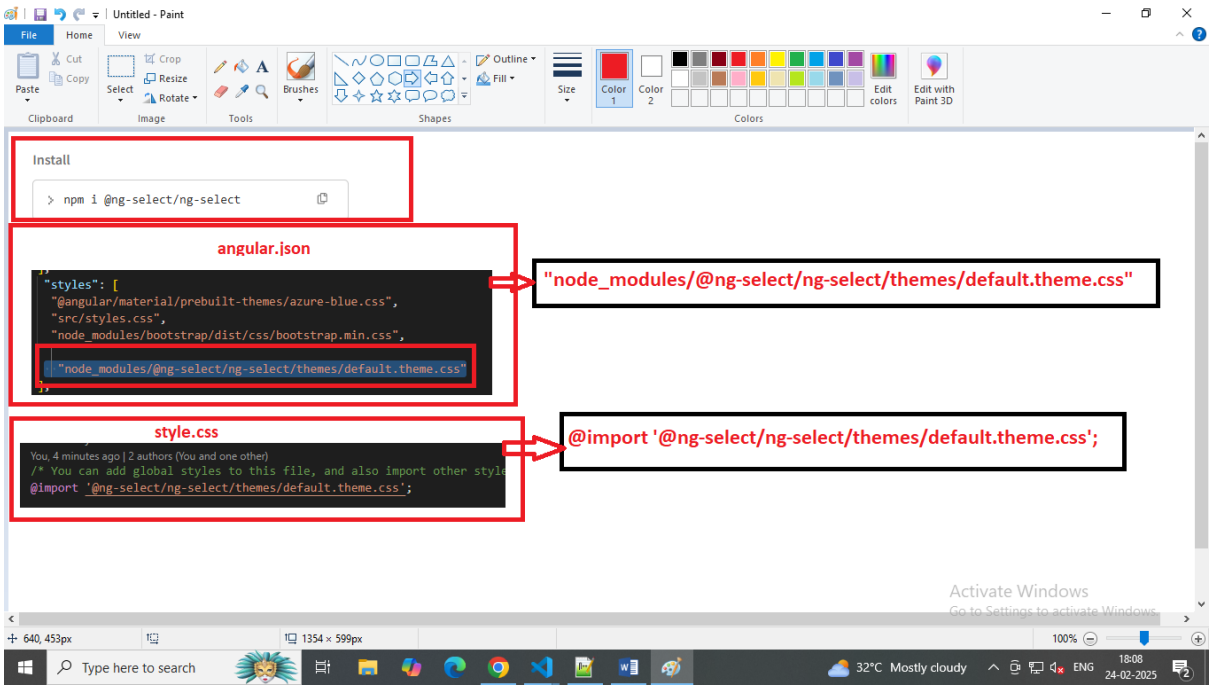
so general ga http interceptor anedi angular lo headers ni set ceskodaniki headres ante manaki content-type,console lo ni network loni headers lo custe maximu yem vuntai content-type,authorization-header vuntundi etc..

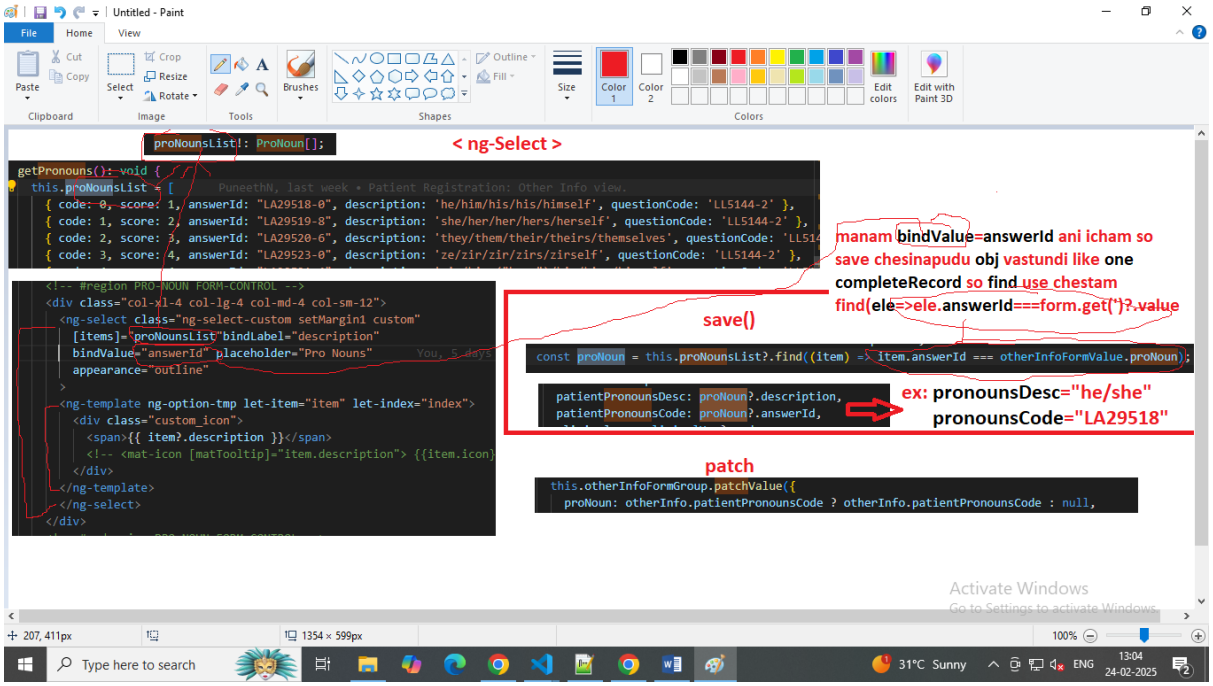
ex:auth-header ante okavela manam api req send cheste server ki evvaru request send chesina gani server vallaki response ivvakudadu,server anedi check chekundi manaki vachina req anedi authorizaion ante ha person authorzied ayyi vunte ne response ivvali so ha paricular person data ne ivaali ex: fb login lo login details correct vunte ne ha person details vastai kani evvru request cheste vallaki radu. so ala data ni ivvali ante manam headers lo aurhorization ane key lo server identify ceyadaniki some value anedi pamputam.

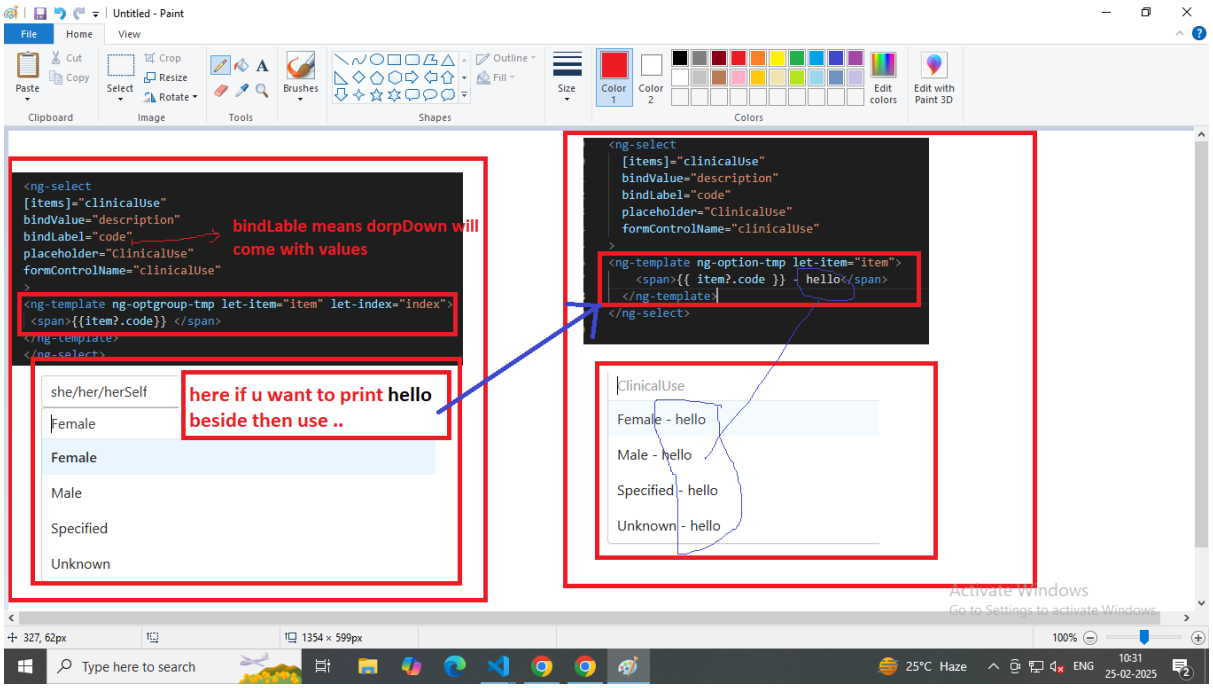
http interseptors code lo manam oka headers ane property tiskontam,here manaki headers anedi oka object{} e object lo manam multiple headers ni petukovachu ex:authorization,content-type etc..e authorization ante enti ante edi oka TOKEN service.ila manam oka token anedi server ki send chestam ex:bearerToken edi ekkuva mandi use chese token. edi oka type of authotication scheme

server nundi vache request ni manam pipe ane operatior dwara manam modify ceskovali ante cheskovachu,manki observalbes lo rxjs lo operators vuntai ga dantlo edo oka danni e pipe lo use cheskovachu manam ex: tap() anedi enti ante manaki server degara nundi edite vastundo adi danni console lo disply ceyadaniki use avtundi.observables ki some extra functionality ni add ceyali ante rxjs lo manaki operators anevi vuntai ha operators ni use ceyali ante pipe anedi compulsory edi oka observable data ni return chestundi and alane manaki rxjs lo catchError ane property vuntudni edi manaki errors ni handle ceyadaniki use chestam

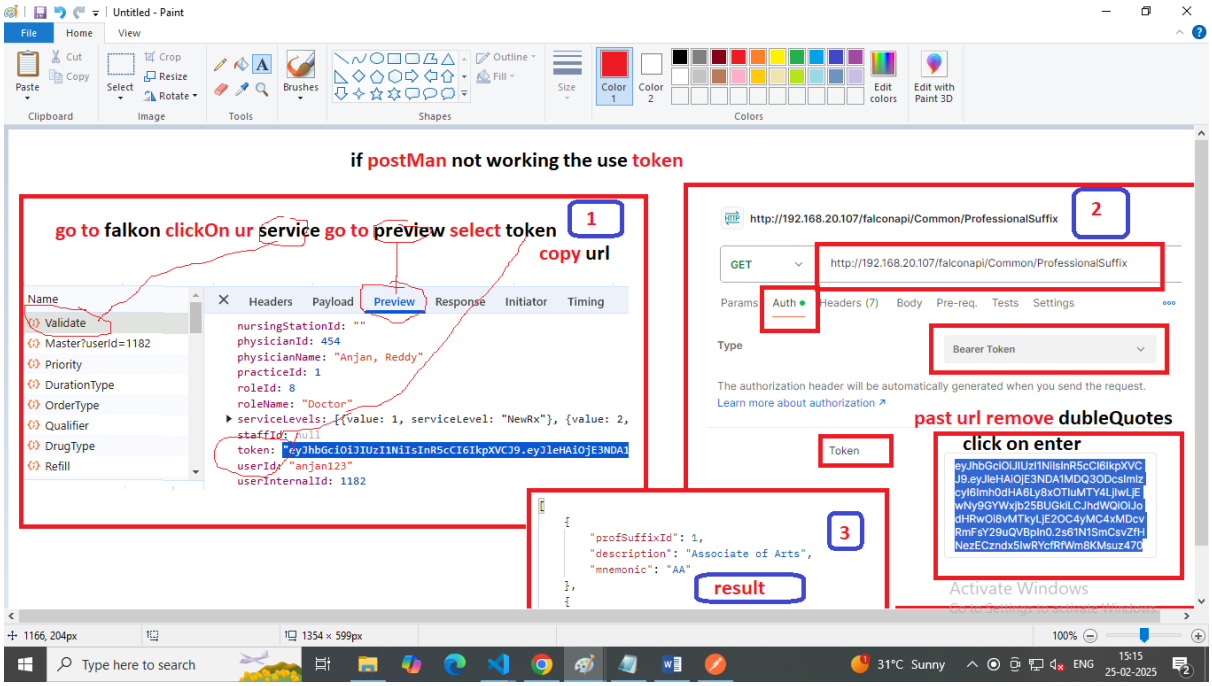
ng-Select ng select ngselect







Postman postman postman


Untitled - Paint
if postMan not working the use token
go to falkon clickOn ur service go to preview select token
copy url
1
Name
Validate
Master?userId=1182
Priority
DurationType
OrderType
Qualifier
DrugType
Refill
Headers
Payload
Preview
Response
Initiator
Timing
nursingStationId: ""
physicianId: 454
physicianName: "Anjan, Reddy"
practiceId: 1
roleId: 8
roleName: "Doctor"
serviceLevels: [{value: 1, serviceLevel: "NewRx"}, {value: 2,
staffId: null
token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3NDA1
userId: "anjan123"
userInternalId: 1182
http://192.168.20.107/falconapi/Common/ProfessionalSuffix
2
GET
http://192.168.20.107/falconapi/Common/ProfessionalSuffix
Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Type
Bearer Token
The authorization header will be automatically generated when you send the request.
Learn more about authorization
past url remove dubleQuotes
click on enter
Token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC
J9.eyJleHAiOjE3NDA1MDQ3ODcsImlz
cyI6Imh0dHA6Ly8xOTIuMTY4LjIwLjE
wNy9GYWxjb25BUGkiLCJhdWQiOiJo
dHRwOi8vMTkyLjE2OC4yMC4xMDcv
RmFsY29uQVBpIn0.2s61N1SmCsvZfH
NezECzndx5IwRYcfRfWm8KMsuz470
[
{
"profSuffixId": 1,
"description": "Associate of Arts",
"mnemonic": "AA"
},
{
3
result
Activate Windows
1166, 204px
1354 × 599px
100%
Type here to search
31°C Sunny
ENG
15:15
25-02-2025