Angular New Document

```
1)Convert JSON array of objects to string array


 jsonObject = [
    {"name": "Bill Gates"},
    {"name": "Max Payne"},
    {"name": "Trump"},
    {"name": "Obama"}
  ];


arrayList: Array<string> = jsonObject.map(person => person.name);







2)Convert string to array of objects


{
   "status": "success",
   "code": 0,
   "message": "version list",
   "payload" :
"[{\"code\":\"AB\",\"short\":\"AB\",\"name\":\"Alberta\",\"country\":\"CA\
"},{\"code\":\"BC\",\"short\":\"BC\",\"name\":\"British
Columbia\",\"country\":\"CA\"}]"

}



this.values = JSON.parse(data.payload);
```

```
var res = {
"payload" :
"[{\"code\":\"AB\",\"short\":\"AB\",\"name\":\"Alberta\",\"country\":\"CA\
"},{\"code\":\"BC\",\"short\":\"BC\",\"name\":\"British
Columbia\",\"country\":\"CA\"}]"
};

var payloadArray = JSON.parse(res.payload);
console.log( payloadArray )
console.log( Array.isArray(payloadArray) ) // true
```

3)Convert object values into array the Angular way

```
 formData : {
       _id: "550de8956e2d0948080e220f"
       category: "Tag1"
       isFeatured: "Yes"
       likeCount: 557
       title: "Integrating WordPress with Your Website"
    }
arryFormData = ["550de8956e2d0948080e220f", "Tag1", "Yes", 557,
"Integrating WordPress with Your Website"]  //output
```

```
const values = Object.keys(obj).map(it => obj[it])

var values = Object.keys(obj).map(function(it) {
   return obj[it]
})
```

4)convert an array of strings into an array of objects in Angular 2?

```
types = ['Old', 'New', 'Template'];

 [
    {
        id: 1,
        name: 'Old'
    },
    {
        id: 2,
        name: 'New'
    },
    {
        id: 3,
        name: 'Template'
    }
]


let newArr = ['Old', 'New', 'Template'].map((x,i) =>({ id: i + 1, name:
x}))


let objects = types.map((value, index) => {
  return {
    id: index + 1,
    name: value
  };
})
```

5)array of objects duplicte remove

```
const arr=[
  {place: "here",  name: "x", other: "other stuff1" },
  {place: "there", name: "x", other: "other stuff2" },
  {place: "here",  name: "y", other: "other stuff4" },
  {place: "here",  name: "z", other: "other stuff5" }
]


var aa=arr.filter((v,i,a)=>a.findIndex(t=>(t.place === v.place &&
t.name===v.name))===i)

console.log('a',aa);


const a=[...new Map(arr.map(item => [item['place'], item])).values()]

console.log('hi',JSON.stringify(a));
```

6)How to Convert Object to array in Angular 4?

```
const obj = {5.0: 10, 28.0: 14, 3.0: 6};

const mapped = Object.keys(obj).map(key => ({type: key, value:
obj[key]}));

console.log(mapped);
```

7)two array of objects except values filter

```
const arrayOne = [
  { value: "4a55eff3-1e0d-4a81-9105-3ddd7521d642", display: "Jamsheer" },
  { value: "644838b3-604d-4899-8b78-09e4799f586f", display: "Muhammed" },
```

```
  { value: "b6ee537a-375c-45bd-b9d4-4dd84a75041d", display: "Ravi" },
  { value: "e97339e1-939d-47ab-974c-1b68c9cfb536", display: "Ajmal" },
  { value: "a63a6f77-c637-454e-abf2-dfb9b543af6c", display: "Ryan" },
];

const arrayTwo = [
  { value: "4a55eff3-1e0d-4a81-9105-3ddd7521d642", display: "Jamsheer"},
  { value: "644838b3-604d-4899-8b78-09e4799f586f", display: "Muhammed"},
  { value: "b6ee537a-375c-45bd-b9d4-4dd84a75041d", display: "Ravi"},
  { value: "e97339e1-939d-47ab-974c-1b68c9cfb536", display: "Ajmal"},
];

const results = arrayOne.filter(({ value: id1 }) => !arrayTwo.some(({
value: id2 }) => id2 === id1));

console.log(results);



8)string interpolation with condition
{{ value !==1&&2 ? 'Not Yet Paid' : 'paid' }}

{{value == 1 ? 'val1' : (element.source == 2 ? 'val2' : '')}}




9)java stack trace issue

export NODE_OPTIONS="--max-old-space-size=8192"

10)Checkbox

 public filterData: any = [];
  public filterDataId = [];
  public selectedAll: any;
  public selectAllValues = false;
  public productListArray: any;

 this.productListArray = [];
```

```
      if (data && data.length > 0) {
        this.productListArray = data.map(list => {
          return { ...list, selected: false };
        });
      }



selectAll() {
for (let i = 0; i < this.productListArray.length; i++) {
this.productListArray[i].selected = this.selectedAll;
}
this.filterDataList();
}
filterDataList() {
this.filterData = this.productListArray.filter(data => {
if (data.selected === true) {
return data;
}
});
this.filterDataId = this.filterData.map(obj => obj.categoryId);
}
checkIfAllSelected() {
this.selectedAll = this.productListArray.every(function (item: any) {
return item.selected === true;
});
this.filterDataList();
}



<th scope="col" width="5%">
<div class="form-group">
<div class="custom-control custom-checkbox mr-sm-2">
<input *ngIf="categorySandbox.getCategoriesList$ | async" type="checkbox"
class="custom-control-input" id="checkbox1" (change)="selectAll()"
[(ngModel)]="selectedAll" [ngModelOptions]="{standalone: true}" />
<label style="cursor: pointer;" class="custom-control-label"
for="checkbox1"></label>
</div>
</div>
```

```
</th>



<th scope="row">
<div class="form-group">
<div class="custom-control custom-checkbox mr-sm-2">
<input type="checkbox" class="custom-control-input"
(change)="checkIfAllSelected()" id="{{ data.categoryId }}"
[ngModelOptions]="{ standalone: true }" [(ngModel)]="data.selected" />
<label class="custom-control-label" for="{{ data.categoryId }}"></label>
</div>
</div>
</th>

11)ng-select optiongroup

<ng-select [items]="account" bindLabel='attributeName'
bindValue="attributeId" groupBy="attribute"
    [(ngModel)]="selectedProjects">
    <ng-template ng-optgroup-tmp let-item="item">
        {{item.attributeGroupName}}
    </ng-template>
    <ng-template ng-option-tmp let-item="item">
        {{item.attributeName}}
    </ng-template>
</ng-select>

    account= [
        {
            "groupId": 9,
            "attributeGroupName": "Electronics",
            "sortOrder": 0,
            "attribute": [
                {
                    "createdBy": null,
                    "createdDate": "2021-06-05T14:07:14.000Z",
                    "modifiedBy": null,
                    "modifiedDate": "2021-08-27T10:51:17.000Z",
                    "attributeId": 12,
```

```
                "groupId": 9,
                "attributeGroupName": "Electr",
                "attributeName": "Electric Product",
                "sortOrder": 0
            },
            {
                "createdBy": null,
                "createdDate": "2021-08-25T13:49:17.000Z",
                "modifiedBy": null,
                "modifiedDate": "2021-08-25T13:49:17.000Z",
                "attributeId": 16,
                "groupId": 9,
                "attributeName": "fdgf",
                "sortOrder": 0
            }
        ]
    },
    {
        "groupId": 12,
        "attributeGroupName": "dfgfg",
        "sortOrder": 0,
        "attribute": []
    },
    {
        "groupId": 1,
        "attributeGroupName": "Laptop",
        "sortOrder": 1,
        "attribute": [
            {
                "createdBy": null,
                "createdDate": "2021-03-23T03:44:43.000Z",
                "modifiedBy": null,
                "modifiedDate": "2021-03-23T03:44:43.000Z",
                "attributeId": 1,
                "groupId": 1,
                "attributeName": "Generation",
                "sortOrder": 1
            },
            {
                "createdBy": null,
```

```
                    "createdDate": "2021-03-23T03:44:55.000Z",
                    "modifiedBy": null,
                    "modifiedDate": "2021-03-23T03:44:55.000Z",
                    "attributeId": 2,
                    "groupId": 1,
                    "attributeName": "Operating System",
                    "sortOrder": 2
                },
                {
                    "createdBy": null,
                    "createdDate": "2021-03-29T09:39:56.000Z",
                    "modifiedBy": null,
                    "modifiedDate": "2021-03-29T09:39:56.000Z",
                    "attributeId": 4,
                    "groupId": 1,
                    "attributeName": "Disk Space",
                    "sortOrder": 3
                }
            ]
        }    ];



11)cke configq
  public config:any;
 private configq={
   toolbar: [
     [
       'Bold',
       'Italic',
       'BulletedList',
       'Styles'
     ],
     ['Table'],
     ['Image']
   ]
 };

 (or)
   private hyperlinkconfiq = {
```

```
    toolbar: [
      [
        'Bold',
        'Italic',
        'BulletedList',
        'Styles'
      ],
      ['Table'],
      ['Image'],
      ['Format'],
      { name: 'links', items: [ 'Link', 'Unlink']},
    ],
    'format_tags': "p;h1;h2;h3;h4;h5"
    };

<ckeditor [config]="config" formControlName="bannerContent"
[ngClass]="{'Validationcolor':bannerForm.get('bannerContent').hasError('re
quired') && bannerForm.get('bannerContent').touched&&submitted==true}">
</ckeditor>

12)checkbox with search


  Form: FormGroup;
  labels = [
    {id: 1, name: "Test", checked: false},
    {id: 2, name: "Testing", checked: false}
  ];
  labels_display = [];
    selectedArray;
    dataCopy;

  constructor(private _formBuilder: FormBuilder) {

  }

  ngOnInit() {
    this.Form = this._formBuilder.group({
        selected: this._formBuilder.array([])
    });
```

```
    this.labels_display = this.labels.map(x => Object.assign({}, x));
  }

  onSelect(data: number, isChecked: boolean) {
    this.selectedArray = <FormArray>this.Form.controls.selected;

    if(isChecked) {
      this.selectedArray.push(new FormControl(data));
    } else {
      let index = this.selectedArray.controls.findIndex(x => x.value ==
data)
      this.selectedArray.removeAt(index);
    }

    // keep sync between labels and labels_display
    const item = this.labels.find(item => item.id === data);
    item.checked = isChecked;
   }

  search(searchterm): void {
    this.labels_display = this.labels.filter(function(tag) {
      return tag.name.toLowerCase().indexOf(searchterm) >= 0;
    });
  }


  //html



  <input type="text" placeholder="Enter Category" [(ngModel)] =
"searchTerm" (ngModelChange) = "search(searchTerm)">
      <div *ngFor="let data of labels_display">
            <input type="checkbox" (change)="onSelect(data.id,
$event.target.checked)" [checked]="data.checked">
            {{data.name}}
   </div>
```

reference:https://stackblitz.com/edit/angular-rexviu?file=app%2Fapp.component.html

13)Ng class

```
<ul *ngFor="let person of people">
  <li [ngClass]="{
  'text-success':person.country === 'UK',
  'text-primary':person.country === 'USA',
  'text-danger':person.country === 'HK'
  }">
  {{ person.name }} ({{ person.country }})
  </li>
</ul>


  getColor(country) {
    switch (country) {
      case "UK":
        return "green";
      case "USA":
        return "blue";
      case "HK":
        return "red";
    }
  }

  people: any[] = [
    {
      name: "Douglas  Pace",
      country: "UK"
    },
    {
      name: "Mcleod  Mueller",
      country: "USA"
    },
    {
      name: "Day  Meyers",
```

```
      country: "HK"
    },
    {
      name: "Aguirre  Ellis",
      country: "UK"
    },
    {
      name: "Cook  Tyson",
      country: "USA"
    }
  ];


  13.1)ngclass with ternary

  <div [ngClass]="varA === varB ? 'css-class-1' : 'css-class-2'">


  14)ng style

  <ul *ngFor="let person of people">
  <li [class.text-success]="person.country === 'UK'"
      [class.text-primary]="person.country === 'USA'"
      [class.text-danger]="person.country === 'HK'">
    {{ person.name }} ({{ person.country }})
  </li>
</ul>

 [ngStyle]="{'width':dependent.length>0 ? 'calc(100% - 168px)':'100%' }"

   14.1)ngstyle colour

   <div [ngStyle]="{'background-color':'green'}"></<div>

   14.2)ngstyle ternary

   <div [ngStyle]="{'background-color':person.country === 'UK' ? 'green' :
'red' }"></<div>
```

15)ternary condition

```
<span> {{key === 'walking' ?
'steps':key==='walk'?'new':key==='wa'?'step3':''}} </span>

<span> {{selectedSport.key === 'walking' ? 'steps' : 'km'}} </span>

*ngIf="(sandbox.totalSalesReportLoaded$ | async) &&
(sandbox.totalSalesReport$ | async).length > 0"

this.firstName = this.filterForm.value.firstName ?
this.filterForm.value.firstName : '';

var icon = (area == 0) ? icon0 : (area == 1) ? icon1 : icon2;

this.year = credits < 30 ? "freshman" : credits <= 59 ? "sophomore" :
credits <= 89 ? "junior" : "senior";
```

16)form array

```
<hello name="{{ name }}"></hello>
<p>
Start editing to see some magic happen :)
</p>
<div [formGroup]="orderForm">
<div formArrayName="items"
*ngFor="let item of orderForm.get('items').controls; let i = index;">
<div [formGroupName]="i">
<input formControlName="name" placeholder="Item name">
<input formControlName="description" placeholder="Item description">
<input formControlName="price" placeholder="Item price">
</div>

Chosen name: {{ orderForm.controls.items.controls[i].controls.name.value
}}
</div>
</div>
<button (click)="addItem()">Add</button>
<button (click)="submit()">submit</button>
```

```
orderForm: FormGroup;
items: FormArray;

constructor(private formBuilder: FormBuilder) {}

ngOnInit() {
  this.orderForm = new FormGroup({
    items: new FormArray([])
  });
}

createItem(): FormGroup {
  return this.formBuilder.group({
    name: '',
    description: '',
    price: ''
  });
}

addItem(): void {
  this.items = this.orderForm.get('items') as FormArray;
  this.items.push(this.createItem());
}
submit(){
  console.log(this.orderForm.value.items[0])
}
```

reference:https://stackblitz.com/edit/form-array-angular?file=src%2Fapp%2Fapp.component.ts

refernce:https://www.digitalocean.com/community/tutorials/angular-reactive-forms-formarray-dynamic-fields

17)show password hide password

```
  <div class="input-btn">
                        <input #x type="password" required [ngClass]="{
'is-invalid': submitted && f.oldPSW.errors }" formControlName='oldPSW'>
                        <button
(click)="x.type=x.type=='password'?'text':'password'">
                          <img src="assets/img/view-dark.svg"
*ngIf="x.type=='text'">
                           <img src="assets/img/view-hide.svg"
*ngIf="x.type=='password'" >
                        </button>
                        <div class="text-danger invalid-data"
*ngIf="submitted &&
this.changePassword.controls['oldPSW'].hasError('required')">
                            Current Password is required
                        </div>
   </div>


   18)id value binding


<input type="text" class="form-control"  #myInput
(keyup)="OnInput(myInput.value)">

<br>
<br>
<br>


<input type="text" class="form-control" #myInput1
(keyup)="OnInput(myInput.value)" >
<br>
<br>

{{myInput.value | json}}
<br>
{{myInput.value.length | json}}
```

```
<p *ngIf="(myInput1.value.length > myInput.value.length)">errorrrrr</p>


<br>
<br>
<br>

{{myInput1.value.length | json}}
<br>
{{myInput1.value | json}}

18)ngif hide or show


<hello name="{{ name }}"></hello>

<button (click)="showMyContainer=!showMyContainer">{{showMyContainer ?
'hide' :'show'}}</button>

<div *ngIf="showMyContainer">
     your code
</div>

 showMyContainer: boolean = false;


19)Only Numbers

<input type="text" (keypress)="keyPressNumbers($event)" />

  keyPressNumbers(event) {
    var charCode = (event.which) ? event.which : event.keyCode;
    // Only Numbers 0-9
    if ((charCode < 48 || charCode > 57)) {
      event.preventDefault();
      return false;
    } else {
      return true;
    }
  }
```

```
  (or)


<input type="text" oninput="this.value = this.value.replace(/[^0-9]/g,
'').replace(/(\..*?)\..*/g, '$1');" />

  (or)
<input type="text" oninput="this.value =
this.value.replace(/[^0-9]/g,'');">
20)Only Numbers with Decimals

  keyPressNumbersDecimal(event) {
    var charCode = (event.which) ? event.which : event.keyCode;
    if (charCode != 46 && charCode > 31
      && (charCode < 48 || charCode > 57)) {
      event.preventDefault();
      return false;
    }
    return true;
  }

 <input type="text" (keypress)="keyPressNumbersDecimal($event)" />

21)Only AlphaNumeric [AB99]

<input type="text" (keypress)="keyPressAlphaNumeric($event)" />

// Only AlphaNumeric
  keyPressAlphaNumeric(event) {

    var inp = String.fromCharCode(event.keyCode);

    if (/[a-zA-Z0-9]/.test(inp)) {
      return true;
    } else {
      event.preventDefault();
      return false;
    }
```

```
  }


22)Allow Alphanumeric with Some Characters [19.a_a]

<input type="text" (keypress)="keyPressAlphaNumericWithCharacters($event)"
/>

// Only AlphaNumeric with Some Characters [-_ ]
  keyPressAlphaNumericWithCharacters(event) {

    var inp = String.fromCharCode(event.keyCode);
    // Allow numbers, alpahbets, space, underscore
    if (/[a-zA-Z0-9-_ ]/.test(inp)) {
      return true;
    } else {
      event.preventDefault();
      return false;
    }
  }

 23)single input form

  submitted = false;
 name = new FormControl('',
 [Validators.required, Validators.minLength(6)]);

 submit() {
     // this.name.setValue('Nancy');
   this.submitted = true;
   if(this.name.invalid) {
       return;
    }
 console.log(this.name);

  }



<label>First Name</label><br>
```

```
<input type="text" [formControl]="name" [ngClass]="{ 'is-invalid':
submitted && name.errors }" />
<div *ngIf="submitted && name.errors" class="invalid-feedback">
   <div *ngIf="name.errors.required">First Name is required</div>
   <div *ngIf="name.errors.minlength">name must be at least 6
characters</div>
</div>

<button (click)="submit()">submit</button>


24)disable submit


https://stackblitz.com/edit/angular-7-login-form-reactive-form-q3nffq?file
=src%2Fapp%2Fapp.component.html


<h4 class="text-center">Login Page Example - Reactive Form - Angular</h4>
<div class="container">
<form (keydown.enter)="$event.preventDefault()" [formGroup]="loginForm"
(ngSubmit)="onSubmit()" novalidate>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" formControlName="email"
placeholder="Enter email">
    <small class="form-text text-muted"
      *ngIf="loginForm.controls.email.touched &&
loginForm.controls.email.errors?.required">
        Please enter email address.!
    </small>
    <small class="form-text text-muted"
    *ngIf="loginForm.controls.email.touched &&
loginForm.controls.email.errors?.email">
      Email address not well formed.!
    </small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
```

```
    <input type="password" class="form-control" formControlName="password"
placeholder="Password">
  </div>
  <button *ngIf="!dis" type="submit" class="btn
btn-primary">Submit</button>
</form>
<button class="btn btn-primary" *ngIf="dis"> disabled Submit</button>
</div>

public a:any;
public dis =false;
ngOnInit(){

  this.a=localStorage.getItem('datas');
  if (this.a!= null && this.a.length>0)
{
    this.dis=true;
    console.log('disabled')
  }
}
loginForm: FormGroup;
constructor() {
  this.loginForm = new FormGroup({
    email: new FormControl('', [Validators.required, Validators.email]),
    password: new FormControl('', [Validators.required])
  });
}
onSubmit() {
  if(this.loginForm.valid) {
    console.log(this.loginForm.value);
    localStorage.setItem("datas", JSON.stringify(this.loginForm.value));
    window.location.reload();
  }
}
_v() {
  return this.loginForm.value;
}


25)Prevent submit when user press enter keyy
```

```
 <form (keydown.enter)="$event.preventDefault()"></form>

26)Cannot read property 'length' of null (javascript)
      if (capital != null && capital.length < 1)

27)normal select
<select>
  <option value="" selected disabled hidden>Choose here</option>
  <option value="1">One</option>
  <option value="2">Two</option>
</select>

28)array duplicate remove

const numbers = [2,3,4,4,2,3,3,4,4,5,5,6,6,7,5,32,3,4,5]

console.log([...new Set(numbers)])

29)checkbox single select

<div class="col-md-12 align-items-center">
    <ul class="list-group">
        <li class="list-group-item" *ngFor="let item of checklist">
            <input type="checkbox" value="{{item.id}}"
[checked]="item.isSelected"
 (change)="isAllSelected(item)"/>
                {{item.value}}</li>
    </ul>
</div>

  public checklist: any[];
  constructor() {
    this.checklist = [
      { id: 1, value: "value1", isSelected: false },
      { id: 2, value: "value2", isSelected: false }
    ];
  }

  isAllSelected(item) {
    this.checklist.forEach(val => {
```

```
      if (val.id == item.id) val.isSelected = !val.isSelected;
      else {
        val.isSelected = false;
      }
    });
  }
```

ref:https://stackblitz.com/edit/angular-ivy-tt9yca?file=src%2Fapp%2Fapp.component.ts

30)cursor hide

```
 <input type="text" (keyup)="new($event)" [ngClass]="{'news': a == 3}">

   public a:any;
  new(event){
    console.log('event',event.target.value)
this.a=event.target.value.length

  }

 input {
  color: black;
  text-shadow: 0px 0px 0px black;
}

.news{
color:transparent;
}
```

31)dynamic field add

refernce:https://stackblitz.com/edit/form-array-patch-qslxxb?file=app%2Fapp.component.ts

32)decimal numbers after point with two values

```
| number : '1.2-2'
```

```
createdDate: "2021-10-21T07:45:38.000Z" -- 21/10/2021 1:15 PM

   {{item?.createdDate | date : "dd/MM/yyyy h:mm a"}}

   {{item?.basePrice | currency:currencyCode:''}}


33)filter with input box

                                heroes1 = [
    { id: 11, name: 'Mr. Nice', country: 'India' },
    { id: 12, name: 'Narco', country: 'USA' },
    { id: 13, name: 'Bombasto', country: 'UK' },
    { id: 14, name: 'Celeritas', country: 'Canada' },
    { id: 15, name: 'Magneta', country: 'Russia' },
    { id: 16, name: 'RubberMan', country: 'China' },
    { id: 17, name: 'Dynama', country: 'Germany' },
    { id: 18, name: 'Dr IQ', country: 'Hong Kong' },
    { id: 19, name: 'Magma', country: 'South Africa' },
    { id: 20, name: 'Tornado', country: 'Sri Lanka' },
  ];
  new(event) {
    console.log('value', event);
    this.heroes = this.heroes1.filter((val) => {
      return
val.name.toLocaleLowerCase().match(event.toLocaleLowerCase());
    });


    <div class="container">
  <div class="row">
    <div class="search-hero">
      <input class="form-control" (keyup)="new($event.target.value)"
type="text" name="search" [(ngModel)]="searchText" autocomplete="off"
placeholder="&#61442;  Start searching for a hero by id or name or
country">
    </div>
    <table class="table table-striped">
      <thead>
      <tr>
```

```
        <th>Id</th>
        <th>Hero Name</th>
        <th>Country</th>
      </tr>
      </thead>
      <tbody>
      <tr *ngFor="let hero of heroes">
        <td>{{hero.id}}</td>
        <td>{{hero.name}}</td>
        <td>{{hero.country}}</td>
      </tr>
      </tbody>
    </table>
  </div>
</div>
```

34)search with checkbox

```
  public filterData: any = [];
  public filterDataId = [];
  public selectedAll: any;
  public selectAllValues = false;
  public productListArray: any;
  backup: any;

    ngOnInit() {
    this.productListArray = [
      { id: 1, name: 'Test', checked: false },
      { id: 2, name: 'Testing', checked: false },
      { id: 3, name: 'abc', checked: false },
      { id: 4, name: 'def', checked: false },
    ];
    this.backup = this.productListArray;
  }

  selectAll() {
    for (let i = 0; i < this.productListArray.length; i++) {
      this.productListArray[i].selected = this.selectedAll;
    }
    this.filterDataList();
```

```
  }

  filterDataList() {
    this.filterData = this.backup.filter((data) => {
      if (data.selected === true) {
        return data;
      }
    });
    this.filterDataId = this.filterData.map((obj) => obj.id);
  }

  checkIfAllSelected() {
    this.selectedAll = this.backup.every(function (item: any) {
      return item.selected === true;
    });
        this.selectedAll = this.productListArray.every(function (item:
any) {
      return item.selected === true;
    });
    if (this.productListArray.length === 0) {
      this.selectedAll = false;
    }
    this.filterDataList();
  }

  new(event) {
    if (event.length === 0) {
      this.productListArray = this.backup;
    }
    this.productListArray = this.backup.filter((val) => {
      return
val.name.toLocaleLowerCase().match(event.toLocaleLowerCase());
    });
       this.checkIfAllSelected();
  }


<hello name="{{ name }}"></hello>
<p>
  Start editing to see some magic happen :)
```

```
</p>
<input type="text" (keyup)="new($event.target.value)">
<div class="form-group">
<div class="custom-control custom-checkbox mr-sm-2">
<input type="checkbox" class="custom-control-input" id="checkbox1"
(change)="selectAll()" [(ngModel)]="selectedAll"
[ngModelOptions]="{standalone: true}" />
<label style="cursor: pointer;" class="custom-control-label"
for="checkbox1">Select All</label>
</div>
</div>



<div class="form-group">
<div class="custom-control custom-checkbox mr-sm-2" *ngFor="let data of
productListArray">
<input type="checkbox" class="custom-control-input"
(change)="checkIfAllSelected()" id="{{ data.id }}" [ngModelOptions]="{
standalone: true }" [(ngModel)]="data.selected" />
<label class="custom-control-label" for="{{ data.id
}}">{{data.name}}</label>
</div>
</div>


<span>{{filterDataId | json}}</span>


ref:https://stackblitz.com/edit/angular-jfx8md?file=src%2Fapp%2Fapp.compon
ent.ts

-----delete------

  delete(val){
this.backup.forEach(function (value) {
  if(value==val){
    value.selected=false;
  }
});
```

```
this.checkIfAllSelected();
}


<div *ngFor="let val of filterData">
<span (click)="delete(val)">{{val | json}}</span>
</div>


ref:https://stackblitz.com/edit/angular-93ehnh?file=src%2Fapp%2Fapp.compon
ent.ts

35)array of object to array
    let a:any=[];
    var array = [
      { name:"string 1", value:"1", other: "that" },
      { name:"string 2", value:"5", other: "that" }
  ];
  a = array.map(b => b.value);
  console.log('a',a)


36)object to array of object
    var a={john:"john@gmail.com",venket:"venkat@gmail.com"}
    var obb=Object.entries(a).map(([name,email])=>
      ({name,email})
    )
    console.log('obb',obb)
    ---output---
    [{
  "name": "john",
  "email": "john@gmail.com"
}, {
  "name": "venket",
  "email": "venkat@gmail.com"
}]


37)checkbox only 3 selected
```

```
list: any[];
filterdataid: any = [];
filterdatalist: any = [];

  this.list = [
    {
      id: 1,
      title: 'Администратор',
      checked: false,
    },
    {
      id: 2,
      title: 'Пользователь',
      checked: false,
    },
    {
      id: 3,
      title: 'Директор',
      checked: false,
    },
    {
      id: 4,
      title: 'Начальник',
      checked: false,
    },
    {
      id: 5,
      title: 'Пользователь',
      checked: false,
    },
    {
      id: 6,
      title: 'Директор',
      checked: false,
    },
    {
      id: 7,
      title: 'Начальник',
      checked: false,
```

```
      },
    ];

  result(event, item) {
    if (event.target.checked == true) {
      if (this.filterdatalist.length > 2) {
        this.list.forEach((data) => {
          if (data.id == this.filterdatalist[0].id) {
            console.log('ssssssss')
            data.checked = false;
          }
        });
        this.filterdatalist.push(item)
                this.filterdatalist = this.filterdatalist.filter((item) =>
{
            return item.checked === true;
        });
      } else {
        this.filterdatalist.push(item)
         this.filterdatalist = this.filterdatalist.filter((item) => {
          if (item.checked === true) {
            return item;
          }
        });
                (or)

       this.filterdatalist=this.list.filter(item=>(item.checked))
      }
      this.ids();
    } else if (event.target.checked == false) {
      this.filterdatalist.forEach((data, index) => {
        if (data.id == item.id) {
          this.filterdatalist.splice(index, 1);
        }
      });
      this.ids();
    }
    console.log('filterdatalist', this.filterdatalist);

  }
```

```
ids(){
  this.filterdataid=this.filterdatalist.map(item=>item.id);
}

<!-------Maximum 3 should be selected------>

  result(event, item) {
  if (event.target.checked == true) {
    if (this.filterdatalist.length > 2) {
      this.list.forEach((data) => {
        if (data.id == item.id) {
          item.checked = false;
        }
      });
      return (event.target.checked = false);
    } else {
      this.filterdatalist = this.list.filter((item) => {
        if (item.checked === true) {
          return item;
        }
      });

      (or)

     this.filterdatalist=this.list.filter(item=>(item.checked))


    }
  this.ids();
  } else if (event.target.checked == false) {
    this.filterdatalist.forEach((data, index) => {
      if (data.id == item.id) {
        this.filterdatalist.splice(index, 1);
      }
    });
    this.ids();
  }
  console.log('filterdatalist', this.filterdatalist);
}
ids(){
  this.filterdataid=this.filterdatalist.map(item=>item.id);
```

```
  }


  <h2>Checkbox list example</h2>

<ul>
  <li *ngFor="let item of list; let i=index">
    <input type="checkbox" [(ngModel)]="item.checked"
(change)="result($event,item,i)">{{item.title}}
  </li>
</ul>

<pre>{{this.filterdatalist | json}}</pre>
<pre>{{this.filterdataid | json}}</pre>


38)download file

link--->https://stackblitz.com/edit/angular-blob-file-download


39)download image

      var link = document.createElement("a");

      document.body.appendChild(link); // for Firefox

      link.setAttribute("href", this.base64Image);
      link.setAttribute("download", "mrHankey.jpg");
      link.click();

40)ngb date picker on model(issue)

::ng-deep .cdk-overlay-container {
    z-index: 2000;
}

42)pdf on model
```

```
link--->https://stackblitz.com/edit/angular-modal-pdf?file=src%2Fapp%2Fapp
.component.html


41)form array with ngmodel

  public data: Structure[];


      this.data = [
      {
        name:'Sachin',
        age:27
      },{
        name: "Gopal",
        age: 27
      },{
        name: "Pankaj",
        age: 24,
        someString: 'React',
        someNumber: 99
      }
      ]
  }


    log(){
    console.log(this.data);
  }

  class Structure {
  name: string;
  age: number;
  someString?: string;
  someNumber?: number;
}


<div>
  <div *ngFor='let item of data'>
```

```
    <div>{{item.name}}</div>
    <div>{{item.age}}</div>
    <div>
      <input type='text' [(ngModel)]='item.someString' />
    </div>
    <div>
      <input type='number' [(ngModel)]='item.someNumber' />
      </div>
  </div>
</div>

<input type='button' (click)='log()' value='Log Changes'/>
```

refer:https://stackblitz.com/edit/angular-ngfor-ngmodel-table?file=src%2Fapp%2Fapp.component.ts

41)formarray with ngmodel value

```
  public data:any;
  public data1:any=[];

    this.data = [
      {
        name:'Sachin',
        age:27,
        version:[
          {
            name:"abc",
            value:"2"
        },
        {
          name:"def",
          value:"3"
      }
        ]
      },{
        name: "Gopal",
        age: 27,
        version:[
          {
```

```
            name:"abwec",
            value:"256"
        },
        {
          name:"deefffef",
          value:"353"
      }
        ]
      },{
        name: "Pankaj",
        age: 24,
        someString: 'React',
        someNumber: 99,
        version:[
          {
            name:"abeec",
            value:"24"
        },
        {
          name:"dettttf",
          value:"53"
      }
        ]
      }
      ]
}


  log(){
  console.log(this.data);
  console.log('sd',this.data1)
}


<div>
<div *ngFor='let item of data;index as i'>
  <div>{{item.name}}</div>
  <div>{{item.age}}</div>
  <div>
    <input type='text' [(ngModel)]='item.someString' />
```

```
    </div>
  <div>
      <input type='number' [(ngModel)]='item.someNumber' />
      </div>
      <div>
      <select [(ngModel)]="data1[i]">
      <option *ngFor="let data of item.version" value="{{data.value}}"
>{{data.name}}</option>
      </select>
      </div>
  </div>
</div>

<input type='button' (click)='log()' value='Log Changes'/>
```

42)paste on input box

```
{
  eventt(e){
  console.log('e',e)
  let values = (e.target as HTMLInputElement).value;
  console.log('valuess',values)
  }


  <input type="text" name="abc" (input)="eventt($event)">


  }
```

43)Image type and size validation

```
html

<div *ngIf="imageSizeError" class="upload-error validation-error">{{
'catalog.Error.Imageshouldbelessthan2MB' | translate }}</div>
<div *ngIf="imageTypeError" class="upload-error validation-error">
{{ 'catalog.Error.Pleaseuploadimageonly(.png,.jpg,.jpeg)' | translate }}
</div>
```

```ts
import * as _ from 'lodash';


    public imageTypeError = false;
    public imageSizeError = false;




   const allowed_types = ['image/png', 'image/jpeg', 'image/jpg'];

    if (!_.includes(allowed_types, inputValue.files[0].type)) {
      this.imageTypeError = true;
      this.ImageUrl = '';
      // this.postImageUrl = './assets/upload-banner/upload.png';
      this.filePath.nativeElement.value = '';
      return;
    }
    this.imageTypeError = false;




    const size = Math.round(inputValue.files[0].size / 1024);
    if (size > 2048) {
      this.imageSizeError = true;
      this.imageUrls = '';

      this.filePath.nativeElement.value = '';
       return;
    }
    myReader.readAsDataURL(file);
    this.imageSizeError = false;

44)Object length

Object.keys(val).length
```

45)Array last index value

```
const arr = [1, 3, 6, 2];
console.log(...arr.slice(-1)); // 2

  let items= [{
    "name": "john",
    "email": "john@gmail.com"
  }, {
    "name": "venket",
    "email": "venkat@gmail.com"
  }
]
  let c=items[items.length-1]
  console.log(c)
```

46)price total calculation

```
fruits = [
 {  description: 'orange', Amount: 50},
 {  description: 'orange', Amount: 50},
 {  description: 'apple', Amount: 75},
 {  description: 'kiwi', Amount: 35},
 {  description: 'watermelon', Amount: 25},];
const sumall = fruits.map(item => item.amount).reduce((prev, curr) => prev
+ curr, 0);
console.log(sumall);
```

47)event emitter

https://stackblitz.com/edit/angular-subscribe-to-event-emitter?file=src%2Fapp%2Fapp.component.ts

48)array to array of objects

```
var data = [
    ["fruits","frozen","fresh","rotten"],
    ["apples",884,494,494],
    ["oranges",4848,494,4949],
```

```
    ["kiwi",848,33,33]
]


function convertToArrayOfObjects(data) {
    var keys = data.shift(),
        i = 0, k = 0,
        obj = null,
        output = [];

    for (i = 0; i < data.length; i++) {
        obj = {};

        for (k = 0; k < keys.length; k++) {
            obj[keys[k]] = data[i][k];
        }

        output.push(obj);
    }

    return output;
}

output

[
    { fruits: 'apples', fresh: 494, frozen: 884, rotten: 494 },
    { fruits: 'oranges', fresh: 494, frozen: 4848, rotten: 4949 },
    { fruits: 'kiwi', fresh: 33, frozen: 848, rotten: 33 }
]


      (OR)

    const data = [
      ["fruits","frozen","fresh","rotten"],
      ["apples",884,494,494],
      ["oranges",4848,494,4949],
      ["kiwi",848,33,33]
  ]
```

```
  const titles = data[0]
  const obj = data.slice(1).map(([fruits,frozen,fresh,rotten]) => ({
fruits,frozen,fresh,rotten }) )
  console.log(obj)


  [
  {
    "fruits": "apples",
    "frozen": 884,
    "fresh": 494,
    "rotten": 494
  },
  {
    "fruits": "oranges",
    "frozen": 4848,
    "fresh": 494,
    "rotten": 4949
  },
  {
    "fruits": "kiwi",
    "frozen": 848,
    "fresh": 33,
    "rotten": 33
  }
]

49)Find Ids from multi dimensional object array

    var products = [
      { id: 1, groups: [ { id: 1.1, selected: true }, { id: 1.2, selected:
false }]},
      { id: 2, groups: [ { id: 2.1, selected: false }, { id: 2.2,
selected: true } ]},{
        id:3
      }];

      var desiredResult = products.map(product => {
        return product?.groups?.filter(group => group.selected).map(item
=> item.id).toString();
```

```
      });
            desiredResult=desiredResult.filter(val=>val!=undefined)  //
optional if no groups in array of object

      console.log(desiredResult);

      ["1.1", "2.2"]


      (or)

      const products = [
    { id: 1, groups: [{ id: 1.1, selected: true }, { id: 1.2, selected:
false }] },
    { id: 2, groups: [{ id: 2.1, selected: false }, { id: 2.2, selected:
true }] }];

   const allIds = [].concat(...products.map(p => p.groups)).filter(g =>
g.selected).map(x => x.id)

   console.log(allIds)


https://stackoverflow.com/questions/50758324/find-ids-from-multi-dimension
al-object-array

50)tooltip background colour change

html

tooltipClass="my-custom-class"

ts

  .my-custom-class .tooltip-inner {
  background-color: #3259b6 !important;
}

51)two array of object filter method
```

```
const arrayOne = [
  { value: "4a55eff3-1e0d-4a81-9105-3ddd7521d642", display: "Jamsheer" },
  { value: "644838b3-604d-4899-8b78-09e4799f586f", display: "Muhammed" },
  { value: "b6ee537a-375c-45bd-b9d4-4dd84a75041d", display: "Ravi" },
  { value: "e97339e1-939d-47ab-974c-1b68c9cfb536", display: "Ajmal" },
  { value: "a63a6f77-c637-454e-abf2-dfb9b543af6c", display: "Ryan" },
];

const arrayTwo = [
  { value: "4a55eff3-1e0d-4a81-9105-3ddd7521d642", display: "Jamsheer"},
  { value: "644838b3-604d-4899-8b78-09e4799f586f", display: "Muhammed"},
  { value: "b6ee537a-375c-45bd-b9d4-4dd84a75041d", display: "Ravi"},
  { value: "e97339e1-939d-47ab-974c-1b68c9cfb536", display: "Ajmal"},
];

const results = arrayOne.filter(({ value: id1 }) => !arrayTwo.some(({
value: id2 }) => id2 === id1));

console.log(results);

ref:https://stackoverflow.com/questions/21987909/how-to-get-the-difference
-between-two-arrays-of-objects-in-javascript

52)Pagination
ts
  public limit = 10;
  public offset: any = 0;
  public currentPage: any = 1;
  public queryData: any = {};

    constructor(
    public router:Router,
    public route: ActivatedRoute
  ) {

   }

  ngOnInit(): void {
      this.offset = this.route.snapshot.queryParamMap.get('offset') || 0;
    this.currentPage = this.route.snapshot.queryParamMap.get('index');
```

```
    }

      getOrderList() {
    const params: any = {};
    params.limit = this.limit;
    params.offset = this.offset;

        this.queryData.offset = this.offset || 0;
    this.queryData.index = this.currentPage || 1;
    this.router.navigate(
      [],
      {
        relativeTo: this.route,
        queryParams: this.queryData,
        queryParamsHandling: 'merge', // remove to replace all query
params by provided
      });
      this.selectedAll=false;
      this.filterData = [];
      this.filterDataId=[];

    }

      getOrderListcount(){
    const params: any = {};
    params.limit = this.limit;
    params.offset = this.offset;
    }

        // page change event
    pageChange(event) {
      this.currentPage = event;
      this.offset = this.limit * (event - 1);
      this.getOrderList();
    }

html

    <div class="pager" style="right: 0;bottom:30px">
```

```
    <app-pager (pageChange)="pageChange($event)" [pageSize]="limit"
[currentPage]="currentPage" [counts]="(orderSandbox.getAllOrderListcount$
| async)"></app-pager>

</div>
```

53)show hide password

```
html
<input toggle placeholder="Password" type="password" id='test'>
module
import { ToggleDirective } from './toggle.directive';

ToggleDirective
import { Directive, ElementRef } from '@angular/core';

@Directive({
  selector: '[toggle]',
})
export class ToggleDirective {
  private _shown = false;

  constructor(private el: ElementRef) {
    const parent = this.el.nativeElement.parentNode;
    const span = document.createElement('span');
    // span.innerHTML = 'show';
    span.innerHTML = '<i class="fa fa-eye-slash"></i>';
    span.addEventListener('click', () => {
      this.toggle(span);
    });
    parent.appendChild(span);
  }

  toggle(span: HTMLElement) {
    this._shown = !this._shown;
    if (this._shown) {
      this.el.nativeElement.setAttribute('type', 'text');
      // span.innerHTML = 'hide';
      span.innerHTML = '<i class="fa fa-eye"></i>';
    } else {
```

```
      this.el.nativeElement.setAttribute('type', 'password');
      // span.innerHTML = 'show';
      span.innerHTML = '<i class="fa fa-eye-slash"></i>';
    }
  }
}
```

https://stackblitz.com/edit/angular-password-hide-show-pjswes?file=src%2Fstyles.css

54)Scroll to invalid fields in form

```
 if(!form.valid){
        this.scrollToError();
 }

 scrollTo(el: Element): void {
   if (el) {
      el.scrollIntoView({ behavior: 'smooth', block: 'center' });
   }
}
scrollToError(): void {
   const firstElementWithError = document.querySelector('.ng-invalid[formControlName]');
   this.scrollTo(firstElementWithError);
}
```

ref:https://levelup.gitconnected.com/reactive-form-validation-with-smooth-scrolling-to-errors-in-angular-8-57fa63b2689d

55)build once deploy anywhere

ref:https://medium.com/mighty-ghost-hack/configure-angular-environments-build-once-deploy-anywhere-c74ed53f0191

56)ngbdropdown not to close when ngselect or select

```
(click)="$event.stopImmediatePropagation()"
```

ref:https://github.com/basvandenberg/ng-select/issues/204

57)ngOndestroy

```
    private subscriptions: Array<Subscription> = [];

    this.subscriptions.push()

        ngOnDestroy() {
        this.subscriptions.forEach(each => each.unsubscribe());
    }
```

58)Reload or previous prevent modal dialogue

Typescript

```
  @HostListener('window:beforeunload', ['$event'])
  beforeUnloadHandler(event: any) {
  event.preventDefault();
  event.returnValue = '';
}
```

Javascript

```
  window.addEventListener('beforeunload',event=>{
    event.preventDefault();
    event.returnValue = '';
  });
```

59)navigate to different tab or minimize browser

Typescript

```
  @HostListener('document:visibilitychange', ['$event'])
  visibilitychange() {
    this.checkHiddenDocument();
  }

    checkHiddenDocument(){
    if (document.hidden){
    // this.pauseVideo();
    } else {
    // this.playVideo();
```

```
    console.log('moved to new tab')
    }

}
```

```
Javascript
document.addEventListener("visibilitychange", function() {
  if (document.visibilityState === 'visible') {
  } else {
    console.log('Tab Changed')
  }
});

(or)
document.addEventListener("visibilitychange", ()=> {
  if (document.hidden) {
  } else {
    console.log('Tab changed')
  }
});
```

60)route change page scroll to top

https://stackoverflow.com/questions/62324385/angular-9-scroll-to-top-on-route-every-change

61)input field focus cursor in for loop

```
 < div * ngFor= "let x of rows; let i = index" >
 <b>Input {{ x }}</b>:
   < input(keyup.enter)="onEnter(i)" id = "input{{i}}" >
     <button (click)="onClick(i)" > Set Focus < /button>
       < /div>

 public rows: any[] = [0, 1, 2, 3, 4];


onClick(index) {
 const next = index === this.rows.length ? 1 : index;
 document.getElementById(`input${next}`).focus();
}
```

62)input field focus cursor

```
 < input id = "inputs" >
   <button (click)="onClick1()" > Set Focus < /button>

onClick1(index) {
 document.getElementById(`inputs`).focus();
}
```

63)every function with array of object value assign

```
let arr = [
 { 'name': "a", 'value': "4" },
 { 'name': "b", 'value': "2" },
 { 'name': "c", 'value': "5" },
];
arr.every((elem, index, arr) => {
 elem.value = '0'
 return true
})
console.log('arr', arr)
```

64)every function with array value assign

```
let arr = [1, 2, 3, 4];
arr.every((elem, index, arr) => {
 arr[index] = 8;
 return true
})

console.log('arr', arr)
```

65)every function form array validation

```
let arr = [
 { 'name': "", 'value': "4" },
 { 'name': "b", 'value': "2" },
 { 'name': "c", 'value': "5" },
];
```

```
let arr1 = arr.every((elem, index, arr) => {
 return elem.value !== "" && elem.name !== ""
})
if (arr1) {
 alert('success')
 return
} else {
 alert('asd')
}
```

66)get specific data or key value from array of object

```
{ id: 1, val: 3, items: { foo: 4, bar: 2 } },
{ id: 2, val: 2, items: { foo: 3, bar: 2 } },
{ id: 3, val: 1, items: { foo: 1, bar: 2 } }
    ];

let result = objArray.map(({ id, val }) => ({ id, val }))

console.log(result)
```

67)get specific data or key value from array of object inside object.

```
const objArray = [
 { id: 1, items: { foo: 4, bar: 2 } },
 { id: 2, items: { foo: 3, bar: 2 } },
 { id: 3, items: { foo: 1, bar: 2 } }
];

let result = objArray.map(({ id, items: { foo } }) => ({ id, foo }))

console.log(result)
```

68) get differnce between two days

```
const diffDays = (date, otherDate) => Math.ceil(Math.abs(date - otherDate)
/ (1000  60  60 * 24));

// Example
diffDays(new Date('2014-12-19'), new Date('2020-01-01'));   // 1839
```

https://www.codegrepper.com/code-examples/javascript/count+days+between+dates+angular

69) Cannot assign to read only property '0' of object '[object Array]

```
this.devices = devicesLoaded.map(device => { return { ...device }; });
```

70) Cannot assign to read only property '0' of object object object

```
obj = { ...object1 }

const object1 = {};
let obj = {};

Object.defineProperty(object1, 'property1', {
 value: 42,
 ['writable']: false,
});
console.log('object1', object1);
// object1.property1 = 77;
obj = { ...object1 }
// throws an error in strict mode
obj.property1 = 99
console.log(obj);
```

71)checkbox maximum 3 shoulb be select and to disabled

ts

```
maxNo = false;
amt = 0;

onChange(isChecked: boolean) {
 if (isChecked)
   this.amt++
 else
   this.amt--
 this.amt === 5 ? this.maxNo = true : this.maxNo = false;
}
```

```
checkBox = [
 { name: 'Daryl', checked: false },
 { name: 'Maggie', checked: false },
 { name: 'Rick', checked: false },
 { name: 'Carl', checked: false },
 { name: 'Negan', checked: false },
 { name: 'Carol', checked: false },
 { name: 'Michonne', checked: false },
 { name: 'Eugene', checked: false }
 ];

 proceed() {
   if (this.checkBox[0].checked)
     alert('Thanks to Daryl Dixon, you have survived!!');
   else
     alert('You died! Daryl Dixon would have saved you...')
 }

html

<div *ngFor="let chk of checkBox; let i = index">
 <input [(ngModel)]="chk.checked" type="checkbox" [disabled]="!chk.checked
&& maxNo" (change)="onChange($event.target.checked)">{{chk.name}}
</div>

<div *ngIf="maxNo">
 <p>5 people chosen. Are you sure about your choices???</p>
 <button (click)="proceed()">Proceed</button>
</div>

https://stackblitz.com/edit/angular-8ghcjz?file=app%2Fapp.component.ts

72)checkbox maximum 3 should be select disable

<ul>
 <li *ngFor="let item of list">
   <input type="checkbox" [(ngModel)]="item.checked"
[disabled]="this.result.length == 3 && item.checked == false
?true:false">{{item.title}}
```

```
 </li>
</ul>
```

73)same route navigation without reload

html

```
(click)="sameroute('/current/route')"
```

ts

```
sameroute(val){
 console.log('routes',val)
 this.router
 .navigateByUrl('/', { skipLocationChange: true })
 .then(() => this.router.navigate([val]));
}
```

74)Compare a nested array with another array and assign if values are same

```
const
   arrayA1 = [{ 13928: false, 13932: true, 13935: true, 13995: false }, {
13927: true, 14024: false }, { 13906: false, 13935: true, 13928: true,
14029: false }],
   arrayA2 = [{ id: 13995, nestedArray: [{ id: 13928, isVerificationMet:
false }, { id: 13932, isVerificationMet: false }, { id: 13935,
isVerificationMet: false }] }, { id: 14024, nestedArray: [{ id: 13927,
isVerificationMet: false }] }, { id: 14029, nestedArray: [{ id: 13906,
isVerificationMet: false }, { id: 13935, isVerificationMet: false }, { id:
13928, isVerificationMet: false }] }],
   result = arrayA2.map(({ id, nestedArray }) => ({
       id,
       nestedArray: nestedArray.map(o => ({
           ...o, isVerificationMet: arrayA1.find(q => id in q)?.[o.id]
       }))
   }));

console.log(result);
```

75)angular wit firebase notification.

https://github.com/firebase/quickstart-js/blob/e66ca9175db42053dc7edeeca3b1117ab6e41e19/messaging/index.html

76)firebase
https://medium.com/@a.adendrata/push-notifications-with-angular-6-firebase-cloud-massaging-dbfb5fbc0eeb
part 1 == https://youtu.be/ODE9l1c3ujY
part 2 == https://youtu.be/sLYm0cxXwhI

77)

```
   var data = [{
     "data": {
       "list": [{
         "name": "A",
         "key": "name",
       }, {
         "name": "B",
         "key": "title",
       }, {
         "name": "C",
         "key": "note",
       }, {
         "name": "D",
         "key": "desc",
       }],
       "show": [
         "title",
         "desc"
       ]
     }
   }]

   let d=data[0].data;
   let arr= d.list.filter(x=> !d.show.includes(x.key));

   console.log(arr);

   //output
```

```
[
 {
   "name": "A",
   "key": "name"
 },
 {
   "name": "C",
   "key": "note"
 }
]
```

ref:https://stackoverflow.com/questions/56835873/how-to-traverse-push-arrays-if-are-not-equal-to-values

78)nested array object comparison with another array of elements and create new array using Javascript or ES6

```
   const sectionDetail = [
       { id: 1, name: 'ma' },
       { id: 2, name: 'na' },
       { id: 3, name: 'ra' },
       { id: 4, name: 'ka' },
       { id: 5, name: 'pa' }];

   const abc = [
       { id: '1', name: 'zam', sections: ['1', 4] },
       { id: '2', name: 'dam', sections: ['3'] },
       { id: '3', name: 'nam', sections: ['2', '4'] }
   ];

   const desired = abc.map(({id, name, sections}) => {
       return {id, name, sections : sectionDetail.filter(f => {
           return sections.map(s => +s).includes(f.id)
       })};

   })

   console.log(desired);
```

```
output:
[
 {
   "id": "1",
   "name": "zam",
   "sections": [
     {
       "id": 1,
       "name": "ma"
     },
     {
       /**id:5**/
       "id": 4,
       "name": "ka"
     }
   ]
 },
 {
   "id": "2",
   "name": "dam",
   "sections": [
     {
       "id": 3,
       "name": "ra"
     }
   ]
 },
 {
   "id": "3",
   "name": "nam",
   "sections": [
     {
       "id": 2,
       "name": "na"
     },
     /**ref:5**/
   ]
 }
]
```

```
ref:https://stackoverflow.com/questions/58310192/nested-array-object-compa
rison-with-another-array-of-elements-and-create-new-arr

79)how to loop through nested objects and push fields after comparing them
in javascript

const data = [
 {group: {id: 1, name:"name1", accountNumber: 1234}, plan:{},
plan_account:{}, plan_benefits:{}},
 {group: {id: 2, name:"name2", accountNumber: 1235}, plan:{},
plan_account:{}, plan_benefits:{}},
 {group: {id: 3, name:"name3", accountNumber: 1234}, plan:{},
plan_account:{}, plan_benefits:{}},
 {group: {id: 4, name:"name4", accountNumber: 1236}, plan:{},
plan_account:{}, plan_benefits:{}},
 {group: {id: 5, name:"name5", accountNumber: 1237}, plan:{},
plan_account:{}, plan_benefits:{}},
]

const exists = (result, obj) => !!result[obj.group.accountNumber];
const result = data.reduce((result, obj) => {
 result[obj.group.accountNumber] = !exists(result, obj) ? {id:
obj.group.id, accountNumber: obj.group.accountNumber} :
result[obj.group.accountNumber];
 return result;
}, {});
console.log(Object.values(result))

//output

[
 {
   "id": 1,
   "accountNumber": 1234
 },
 {
   "id": 2,
   "accountNumber": 1235
 },
 {
```

```
   "id": 4,
   "accountNumber": 1236
 },
 {
   "id": 5,
   "accountNumber": 1237
 }
]
```

reference:https://stackoverflow.com/questions/62886563/how-to-loop-through-nested-objects-and-push-fields-after-comparing-them-in-javas

80)group objects inside nested array

```
const parentArray = [[
  {
     key: 123,
     value: 'India'
  }, {
     key: 124,
     value: 'USA'
  }, {
     key: 125,
     value: 'Japan'
  }, {
     key: 126,
     value: 'Denmark'
  }, {
     key: 127,
     value: 'Austria'
  },
], [
  {
     key: 124,
     value: 'Kenya'
  }, {
     key: 126,
     value: 'UK'
  }, {
```

```
    key: 123,
    value: 'Germany'
  }, {
    key: 127,
    value: 'Spain'
  }, {
    key: 125,
    value: 'Portugal'
  },
]];
const map = {};
parentArray.forEach(arr => {
  arr.forEach(obj => {
    const { key, value } = obj;
    if(map[key]){
       map[key].push(value);
    }else{
       map[key] = [value]
    }
  })
});
console.log(map);


//output

{
  '123': [ 'India', 'Germany' ],
  '124': [ 'USA', 'Kenya' ],
  '125': [ 'Japan', 'Portugal' ],
  '126': [ 'Denmark', 'UK' ],
  '127': [ 'Austria', 'Spain' ]
}


reference:https://www.tutorialspoint.com/group-objects-inside-the-nested-array-javascript

81)Comparing arrays of objects with nested array of object
```

```
const
   arrayA = [[{ value: "#0767b9", id: 162, productId: 1 }, { value:
"#f4b7d4", id: 164, productId: 1 }], [{ value: "#44acd8", id: 102,
productId: 2 }], [{ value: "#609923", id: 106, productId: 3 }, { value:
"#ee3b70", id: 107, productId: 3 }]],
   arrayB = [{ id: 1, optionValue: [{ value: "#002e63", id: 161,
productId: 1 }, { value: "#0767b9", id: 162, productId: 1 }, { value:
"#010b1d", id: 163, productId: 1 }, { value: "#f4b7d4", id: 164,
productId: 1 }] }, { id: 2, optionValue: [{ value: "#EC7063", id: 93,
productId: 2 }, { value: "#bf0000", id: 94, productId: 2 }, { value:
"#44acd8", id: 102, productId: 2 }, { value: "#ffdbdb", id: 103,
productId: 2 }] }, { id: 3, optionValue: [{ value: "#d861bd", id: 105,
productId: 3 }, { value: "#609923", id: 106, productId: 3 }, { value:
"#ee3b70", id: 107, productId: 3 }] }, { id: 4, optionValue: [{ value:
"#44acd8", id: 165, productId: 4 }] }],
   identifiers = arrayA.reduce((r, a) => {
       a.forEach(({ id, productId }) => (r[productId] = r[productId] ||
{})[id] = true);
       return r;
   }, {}),
   result = arrayB.map(o => identifiers[o.id]
       ? { ...o, optionValue: o.optionValue.filter(({ id, productId }) =>
identifiers[productId][id]) }
       : o
   );

  console.log(result);


//output

[
 {
   "id": 1,
   "optionValue": [
     {
       "value": "#0767b9",
       "id": 162,
       "productId": 1
     },
```

```
    {
      "value": "#f4b7d4",
      "id": 164,
      "productId": 1
    }
  ]
},
{
  "id": 2,
  "optionValue": [
    {
      "value": "#44acd8",
      "id": 102,
      "productId": 2
    }
  ]
},
{
  "id": 3,
  "optionValue": [
    {
      "value": "#609923",
      "id": 106,
      "productId": 3
    },
    {
      "value": "#ee3b70",
      "id": 107,
      "productId": 3
    }
  ]
},
{
  "id": 4,
  "optionValue": [
    {
      "value": "#44acd8",
      "id": 165,
      "productId": 4
    }
```

```
   ]
 }
]
```

reference:https://stackoverflow.com/questions/65679664/javascript-comparing-arrays-of-objects-with-nested-array-of-object

82)two div with search

```
 public originalData = [
   {
     productId: 290,
     id: 1067,
     taxType: 0,
     taxValue: 0,
     skuId: 2600,
     price: '1000.00',
     name: 'chocolate',
     isSimplified: 0,
     description: 'chocolate',
     quantity: 50,
     rating: null,
     productSlug: 'bb123',
     hasStock: 0,
     outOfStockThreshold: null,
     containerName: 'Product/Electronics/Laptops/',
     image: 'Img_1557483575649.jpeg',
     defaultImage: 1,
     skuName: 'choco1231',
     productDiscount: null,
     productSpecial: null,
     pricerefer: '',
     flag: '',
     stockStatus: 'inStock',
     wishListStatus: 0,
```

```
  },
  {
    productId: 291,
    id: 1067,
    taxType: 0,
    taxValue: 0,
    skuId: 2600,
    price: '1000.00',
    name: 'chocolate 2',
    isSimplified: 0,
    description: 'chocolate 1',
    quantity: 50,
    rating: null,
    productSlug: 'bb123',
    hasStock: 0,
    outOfStockThreshold: null,
    containerName: 'Product/Electronics/Laptops/',
    image: 'Img_1557483575649.jpeg',
    defaultImage: 1,
    skuName: 'choco1231',
    productDiscount: null,
    productSpecial: null,
    pricerefer: '',
    flag: '',
    stockStatus: 'inStock',
    wishListStatus: 0,
  },
  {
    productId: 1384,
    id: 1068,
    taxType: 1,
    taxValue: 0,
    skuId: 2159,
    price: '500.00',
    name: 'Hopscotch Girl Empire Knee Length Dress',
    isSimplified: 1,
    description:
      '<ul>\n\t<li>Care Instructions: Gentle Wash</li>\n\t<li>Fit Type:
Regular</li>\n\t<li>Material: 95%Cotton+ 5% Polyester</li>\n\t<li>Care
```

```
Instruction: Gentle Wash</li>\n\t<li>All Dimensions are in cms :- Casual
Dresses: Length: 46 cm , Chest: 58 cm</li>\n</ul>\n',
    quantity: 100,
    rating: '5.00',
    productSlug: 'hopscotch-girl-empire-knee-length-dress',
    hasStock: 0,
    outOfStockThreshold: null,
    containerName: 'Ven0191/',
    image: '610egeH1iOL_1634018940718.jpeg',
    defaultImage: 1,
    skuName: 'SKU74564867',
    productDiscount: null,
    productSpecial: null,
    pricerefer: '',
    flag: '',
    stockStatus: 'inStock',
    wishListStatus: 0,
  },
];

public selectedDatas = [];
filval = '';
filval1 = '';

selectData(data) {
  this.originalData.forEach((val, i) => {
    if (val.productId == data.productId) {
      this.selectedDatas.push(val);
      this.originalData.splice(i, 1);
    }
  });
}

unselectData(value) {
  console.log('value', value);
  this.selectedDatas.forEach((val, i) => {
    if (val.productId == value.productId) {
      this.originalData.push(val);
      this.selectedDatas.splice(i, 1);
    }
```

```
   });
 }



//html

<div style="display: flex;
justify-content: space-evenly;">
<div style="color:blue;">
<h1>Unselected Data</h1>
<ul>
 <li *ngFor="let data of originalData | search: [filval]: ['name'] ;">
   <a (click)="selectData(data)">{{data.name}}</a>
 </li>
 // <tr *ngFor="let item of array | search: [name.value, category.value]:
['name', 'catName'] ;">
</ul>

</div>

<div style="color:grey;">
<h1>Selected Data</h1>
<ul>
<li *ngFor="let datas of selectedDatas | search: [filval1]: ['name'] ;">
 <a (click)="unselectData(datas)">{{datas.name}}</a>
</li>
</ul>
</div>
</div>



//search.pipe.ts

import { Pipe, PipeTransform } from "@angular/core";
@Pipe({
 name: "search",
 pure:false
})
export class SearchPipe implements PipeTransform {
```

```
 transform(list: any[], value: [], key: []): any {
   console.log('value',value)
   value.forEach((name, index) => {
     if (name) {
       console.log('name',name)
       list = list.filter((item) => {
         return (item[key[index]]
           .toString()
           .toLowerCase()
           .indexOf(name.toString().toLowerCase()) !== -1)
       });
     }
   });
   return list;
 }
}


import { SearchPipe} from './search.pipe';

 declarations: [SearchPipe ],

ref:https://stackblitz.com/edit/angular-multiple-search-filters-zutrsb?file=src%2Fapp%2Fapp.component.html,src%2Fapp%2Fapp.component.ts,src%2Fapp%2Fhello.component.ts,src%2Fapp%2Fsearch.pipe.ts,src%2Fapp%2Fapp.module.ts


83)search pipe multiple column with single input

html

<label for="name">Name </label>
<input type="text" id="name" [formControl]="name"/>
<table>
 <tr>
   <th>Name</th>
   <th>Category</th>
 </tr>
 <tr *ngFor="let item of array | search: name.value;">
   <td>{{item.name}}</td>
```

```
    <td>{{item.catName}}</td>
  </tr>
</table>

ts

name = new FormControl('');

array = [
  {
    name: 'Ali',
    catName: 'Human',
  },
  {
    name: 'Ahmed',
    catName: 'Human',
  },
  {
    name: 'Alexa',
    catName: 'Robot',
  },
  {
    name: 'Tom',
    catName: 'Robot',
  },
  {
    name: 'Thompson',
    catName: 'Human',
  },
  {
    name: 'Peter',
    catName: 'Human',
  },
  {
    name: 'Peshawar',
    catName: 'City',
  },
]
/*search.pipe.ts */
import { Pipe, PipeTransform } from "@angular/core";
```

```
@Pipe({
 name: "search"
})

export class SearchPipe implements PipeTransform {

 transform(value:any[],searchString:string ){

    if(!searchString){
      console.log('no search')
      return value
    }

    return value.filter(it=>{
        const name = it.name.toLowerCase().includes(searchString)
        const catName =
it.catName.toLowerCase().includes(searchString.toLowerCase())
        return (name + catName);
    })
 }
}
```

84)search multiple column with single input(easy)

```
 array = [
   {
     name: 'Ali',
     catName: 'Human22',
   },
   {
     name: 'Ahmed',
     catName: 'Human2',
   },
   {
     name: 'Alexa',
     catName: 'Robot',
   },
   {
     name: 'Tom',
     catName: 'Robot',
```

```
   },
   {
     name: 'Thompson',
     catName: 'Human',
   },
   {
     name: 'Peter',
     catName: 'Human',
   },
   {
     name: 'Peshawar',
     catName: 'City',
   },
 ]
   name = new FormControl('');

//html

<label for="name">Name </label>
<input type="text" id="name" [formControl]="name"/>
<table>
 <tr>
   <th>Name</th>
   <th>Category</th>
 </tr>
 <tr *ngFor="let item of array | search:['name','catName']:name.value">
  <!-- <tr *ngFor="let item of array | search:'name':name.value">-->

   <td>{{item.name}}</td>
   <td>{{item.catName}}</td>
 </tr>
</table>

//searchpipe.ts

@Pipe({
   name: 'search',
   pure: false
})
```

```
export class SearchPipe implements PipeTransform {
   transform(items: any[], field : any, value : string): any[] {
       if (!items) return [];
       if(typeof field == 'string'){
           let rtItems:any = items;
           try{
               rtItems = items.filter(it =>
it[field].toLowerCase().indexOf(value.toLowerCase()) > -1 );
           }finally{
               return rtItems;
           }
       }else{
           let rtItems:any = items;
           try{
               rtItems = items.filter(it => {
                   for(let f of field){
                       if(it[f].toLowerCase().indexOf(value.toLowerCase())
> -1){
                           return true;
                       }
                   }
               });
           }finally{
               return rtItems;
           }
       }


   }
}
```

85)file download progress ts file

```
     let url = this.baseUrl +
'/download/item-data-download-key?customerCode=' + this.customerDetails +
'&key=' + this.downloardKey;
     this.http.get(url, { responseType: 'blob', reportProgress: true,
observe: 'events' }).subscribe(event => {
```

```
      if (event.type === HttpEventType.DownloadProgress) {
        this.percentDone = Math.round(100 * event.loaded / event.total);
      }
      if (event.type === HttpEventType.Response) {
        const filename = this.downloardKey + Date.now() + '.xlsx';
        const blob = new Blob([(event['body'])], { type: 'text/xlsx' });
        saveAs(blob, filename);
      }
    }

    )


85)queryParams (html)
<button [routerLink]="['routerlink']"
    queryParamsHandling="preserve"> Back </button>


ts
 this.router.navigate([routerlink], { queryParamsHandling: 'preserve' })



Ts

   this.router.navigate(['routerlink'], { queryParams: { page:
this.currenPage, itemType: this.itemType }, queryParamsHandling: "merge"
});

85)outside div to close model or div

ts

  isMenuOpen = false;


  @ViewChild('toggleButton') toggleButton: ElementRef;
  @ViewChild('menu') menu: ElementRef;

  constructor(private renderer: Renderer2) {
```

```
    this.renderer.listen('window', 'click',(e:Event)=>{
        if(e.target !== this.toggleButton.nativeElement &&
e.target!==this.menu.nativeElement){
            this.isMenuOpen=false;
        }
    });
  }


  toggleMenu() {
    this.isMenuOpen = !this.isMenuOpen;
  }


  html


 <button #toggleButton (click)="toggleMenu()"> Toggle Menu</button>

<div class="menu" *ngIf="isMenuOpen" #menu>
I'm the menu. Click outside to close me
</div>


ref:https://www.angularfix.com/2022/01/how-to-detect-click-outside-of-elem
ent.html


86)Ngb date picker maxdate today

html

[maxDate]="todaysDate"

ts

  todaysDate: any;

  this.todaysDate = {
    year: new Date().getFullYear(),
```

```
    month: new Date().getMonth() + 1,
    day: new Date().getDate()
  };
```

87)Ngb date picker set to previous months

```
         public datePipe: DatePipe

         const date2 = new Date();
         let b;
         b = new Date(date2.getFullYear(), date2.getMonth() - 2, 1);
         this.todayDate = {
           year: Number(this.datePipe.transform(b,"yyyy")),
           month: Number(this.datePipe.transform(b,"MM")),
           day: 1
         };
```

88)form value changes

```
      this.form.valueChanges.subscribe(x => {
        console.log(x)
     })
```

89)form single field value changes

```
      this.form.get("firstName").valueChanges.subscribe(x => {
        console.log('firstname value changed')
        console.log(x)
        console.log(this.reactiveForm.get("firstname").value) //updates
latest value
        console.log(this.form.value)   //still shows the old first name

     })


        this.form.get("firstName").valueChanges.subscribe(x => {
        console.log('firstname value changed')
        console.log(x)
        console.log(this.form.value)   //still shows the old first name
```

```
        setTimeout(() => {
          console.log('after',this.form.value)   //shows the new first
name
        })

     })
```

90)ValueChanges of FormGroup

```
   this.reactiveForm.get("address").valueChanges.subscribe(selectedValue
=> {
  console.log('address changed')
  console.log(selectedValue)
})
```

ref:https://www.tektutorialshub.com/angular/valuechanges-in-angular-forms/

91)filter methods

https://danielk.tech/home/angular-how-to-apply-filters-to-ngfor#custom-fil
ter-pipe

92)scroll to last element in div

https://stackblitz.com/edit/angular-dam13i?file=app%2Fapp.component.ts

93)scroll to last element (easy)

html

```
<button (click)="buttons()">scroll</button>
<button (click)="getData()">Get Data</button>

<ul *ngIf="users">
  <li #user *ngFor="let userDetail of users">
    <h1>{{ userDetail.name }}</h1>
    <h2>{{ userDetail.username }}</h2>
    <h3>{{ userDetail.email }}</h3>
  </li>
```

```
</ul>

ts

import {
  Component,
  ChangeDetectorRef,
  ViewChildren,
  ElementRef,
  QueryList,
  NgZone,
} from '@angular/core';

  users: any = [];
  @ViewChildren('user', { read: ElementRef })
  renderedUsers: QueryList<ElementRef>;
  indexToScrollTo = 4;

  buttons() {
    this.indexToScrollTo = this.users.length - 1;
    const userToScrollOn = this.renderedUsers.toArray();
    userToScrollOn[this.indexToScrollTo]?.nativeElement.scrollIntoView({
      behavior: 'smooth',
    });
  }


ref:https://stackblitz.com/edit/angular-scroll-to-item-in-list-after-data-is-fetched-from-api?file=src%2Fapp%2Fapp.component.ts,src%2Fapp%2Fapp.component.html


 94)scroll to last element

ref:https://stackblitz.com/edit/angular-scroll-bottom?file=app%2Ftodos.service.ts

 95)scroll to last

   setTimeout(() => {
```

```
      document.getElementById('scroll').scrollTop =
document.getElementById('scroll').scrollHeight
      console.log("document.getElementById('scroll')",
document.getElementById('scroll'));
    }, 500)
```

96)error subscribe

```
 getUser(id){
  return this.http.get('http:..../' + id)
    .map(res => res.json())
    .catch(this.handleError);
}

private handleError(error: any) {
  let errMsg = (error.message) ? error.message : error.status ?
`${error.status} - ${error.statusText}` : 'Server error';
  return Observable.throw(error);
}
```

97)difference between two days

```
let firstDate = moment('2019/03/31');
let secondDate = moment('2019/04/01');
const diffInDays = Math.abs(this.firstDate.diff(this.secondDate, 'days'));
const daysDiff = Math.floor(Math.abs(<any>firstDate - <any>secondDate) /
(1000*60*60*24));

console.log('daysDiff',daysDiff,)
```

98)ngb date compare

```
  startdate: {year: number, month: number,day:number};
    todaydate:any;
  public errors:any=false;
  constructor(private calendar: NgbCalendar) {
    this.todaydate = {
      year: new Date().getFullYear(),
      month: new Date().getMonth() + 1,
      day: new Date().getDate()
```

```
    };
  }

    submit(){
    const date: NgbDate = new NgbDate(this.startdate.year,
this.startdate.month, this.startdate.day);
    let condition1=date.after(this.todaydate);
    console.log('conditions',condition1,date.before(this.todaydate))
    let condition= date.equals(NgbDate.from({ year: this.todaydate.year,
month: this.todaydate.month, day: this.todaydate.day }));
    if(condition1||condition){
      this.errors=false;
    }else{
      this.errors=true;
    }
    console.log('condition',condition)
    console.log('if',condition1||condition)



      (or)

         const date: NgbDate = new NgbDate(this.startdate.year,
this.startdate.month, this.startdate.day);
    let condition = (date.after(this.todaydate)) ||
(date.equals(NgbDate.from({ year: this.todaydate.year, month:
this.todaydate.month, day: this.todaydate.day })))
     if(condition==true){
      this.errors=false
     }else{
       this.errors=true;
     }

  }



  <ngb-datepicker #dp [(ngModel)]="startdate"></ngb-datepicker>
<p *ngIf="errors">error</p>

<button (click)="submit()">click</button>
```

```
ref:https://stackblitz.com/edit/angular-drdelm?file=src%2Fapp%2Fdatepicker
-basic.html,src%2Fapp%2Fdatepicker-basic.ts,src%2Fapp%2Fdatepicker-basic.m
odule.ts


(or)

  if
(moment(this.todayDate).isSameOrAfter(this.taskForm.value.reviewDate)) {
      this.ShowErrorForReview = false;
    } else {
      this.ShowErrorForReview = true;
            return;
    }

99)ngb date picker compare

  startdate: {year: number, month: number,day:number};
  enddate: {year: number, month: number,day:number};
  todaydate:any;
  public errors:any=false;
  public enderror:any=false;
  constructor(private calendar: NgbCalendar) {
    this.todaydate = {
      year: new Date().getFullYear(),
      month: new Date().getMonth() + 1,
      day: new Date().getDate()
    };
  }

  isValid(){

    const Startdate: NgbDate = new NgbDate(this.startdate?.year,
this.startdate?.month, this.startdate?.day);
    let isValidStartdate = (Startdate.after(this.todaydate)) ||
(Startdate.equals(NgbDate.from({ year: this.todaydate.year, month:
this.todaydate.month, day: this.todaydate.day })))
```

```
     const Enddate: NgbDate = new NgbDate(this.enddate?.year,
this.enddate?.month, this.enddate?.day);
     let isValidenddate = (Enddate.after(this.todaydate)) ||
(Enddate.equals(NgbDate.from({ year: this.todaydate.year, month:
this.todaydate.month, day: this.todaydate.day })))

     if((isValidStartdate==true)){
      this.errors=false;
     }else{
       this.errors=true;
     }

      if((isValidenddate==true)){
       this.enderror=false;
      }else{
        this.enderror=true;
      }
      return [this.errors,this.enderror];

  }


    submit(){

      // console.log('asd',this.isValid())
      let dateValid=this.isValid();
      if(dateValid[0]==true || dateValid[1]==true){
        return
      }
      alert('hiiii')

  }

  html

  <ngb-datepicker #dp [(ngModel)]="startdate"></ngb-datepicker>
<p *ngIf="errors">start date error</p>
<hr />
<ngb-datepicker #dp [(ngModel)]="enddate"></ngb-datepicker>
<p *ngIf="enderror">end date error</p>
```

```
 <button (click)="submit()">click</button>

100) array of object inside object based on condition

    this.list = [
      {
          "moduleName": "Job Introduction",
            "moduleAliasName": "Job Introduction",
        "moduleSlugName": "job_introduction",
            "moduleRoutes": "job-intro",
        "sequenceId": 1
      },
      {
        "moduleName": "About us",
            "moduleAliasName": "About Us",
        "moduleSlugName": "about_us",
        "moduleRoutes": "about",
            "sequenceId": 2,
        "childModule":
                    [
                      {
              "moduleName": "Who we are",
                  "moduleAliasName": "Who We Are",
              "moduleSlugName": "who_we_are",
              "moduleRoutes": "whoweare",
                  "sequenceId": 3
                      },
                      {
                      "moduleName": "TimeLine",
                  "moduleAliasName": "TimeLine",
              "moduleSlugName": "timeline",
              "moduleRoutes": "timeline",
                  "sequenceId": 4
                      },
                       {
                      "moduleName": "Core Team",
                  "moduleAliasName": "Core Team",
              "moduleSlugName": "core_team",
```

```
"moduleRoutes": "core-team",
    "sequenceId": 5
        },
         {
        "moduleName": "Hear it from them",
    "moduleAliasName": "Hear it from them",
"moduleSlugName": "hear_it_from_them",
"moduleRoutes": "hear-from",
    "sequenceId": 6
        },
         {
        "moduleName": "CEO story",
    "moduleAliasName": "CEO story",
"moduleSlugName": "ceo_story",
"moduleRoutes": "ceo-story",
    "sequenceId": 7
        },
         {
        "moduleName": "Funding",
    "moduleAliasName": "Funding",
"moduleSlugName": "funding",
"moduleRoutes": "funding",
    "sequenceId": 8
        },
         {
        "moduleName": "Your peer group",
    "moduleAliasName": "Your peer group",
"moduleSlugName": "your_peer_group",
"moduleRoutes": "peer-group",
    "sequenceId": 9
        },
         {
        "moduleName": "Talent acquisition",
    "moduleAliasName": "Talent acquisition",
"moduleSlugName": "talent_acquisition",
"moduleRoutes": "talent",
    "sequenceId": 9
        },
        {
        "moduleName": "Product service",
```

```
            "moduleAliasName": "Product service",
          "moduleSlugName": "product_service",
          "moduleRoutes": "product-service",
              "sequenceId": 10
                  }

                ]

    },
      {
          "moduleName": "More in detail",
          "moduleAliasName": "More in detail",
      "moduleSlugName": "more_in_detail",
      "moduleRoutes": "more-in-detail",
          "sequenceId": 11,
      "childModule":
                  [
                     {
                    "moduleName": "Vision",
                "moduleAliasName": "Vision",
          "moduleSlugName": "vision",
          "moduleRoutes": "vision",
              "sequenceId": 12
                   },
                    {
                    "moduleName": "Mission",
                "moduleAliasName": "Mission",
          "moduleSlugName": "mission",
          "moduleRoutes": "mission",
              "sequenceId": 13
                   },
                    {
                    "moduleName": "Culture",
                "moduleAliasName": "Culture",
          "moduleSlugName": "culture",
          "moduleRoutes": "culture",
              "sequenceId": 14
                   },
                    {
                    "moduleName": "Clientele",
```

```
            "moduleAliasName": "clientele",
        "moduleSlugName": "clientele",
        "moduleRoutes": "clientele",
            "sequenceId": 15
                }
             ]

      },
      {
        "moduleName": "Benefits",
        "moduleAliasName": "Benefits",
    "moduleSlugName": "benefits",
    "moduleRoutes": "benefit",
        "sequenceId": 16
      },
      {
        "moduleName": "Job description",
        "moduleAliasName": "Job description",
    "moduleSlugName": "Job description",
    "moduleRoutes": "job-desc",
        "sequenceId": 17
      },
      {
        "moduleName": "Gallery",
        "moduleAliasName": "Gallery",
    "moduleSlugName": "gallery",
    "moduleRoutes": "gallery",
        "sequenceId": 18,
    "childModule":
              [
                 {
                "moduleName": "Work",
            "moduleAliasName": "work",
        "moduleSlugName": "work",
        "moduleRoutes": "work",
            "sequenceId": 19
                 },
                  {
                "moduleName": "Outdoor",
            "moduleAliasName": "Outdoor",
```

```
            "moduleSlugName": "outdoor",
            "moduleRoutes": "outdoor",
                "sequenceId": 20
                    },
                     {
                   "moduleName": "Fun & Culture",
                "moduleAliasName": "Fun & Culture",
            "moduleSlugName": "fun_culture",
            "moduleRoutes": "fun-culture",
                "sequenceId": 21
                    }
                 ]
          },
          {
            "moduleName": "Online Feed",
            "moduleAliasName": "Online Feed",
        "moduleSlugName": "online_feed",
        "moduleRoutes": "online-feed",
            "sequenceId": 22
          },
          {
            "moduleName": "Q&A",
            "moduleAliasName": "Q&A",
        "moduleSlugName": "question_answer",
        "moduleRoutes": "question-answer",
            "sequenceId": 23
          },
          {
            "moduleName": "Other Opening",
            "moduleAliasName": "Other Opening",
        "moduleSlugName": "other_opening",
        "moduleRoutes": "other-opening",
            "sequenceId": 24
          },
          {
            "moduleName": "Reasons for Decling",
            "moduleAliasName": "Reasons for Decling",
        "moduleSlugName": "reasons_for_decling",
        "moduleRoutes": "reasons",
            "sequenceId": 25
```

```
            },
            {
              "moduleName": "Interested",
              "moduleAliasName": "Interested",
          "moduleSlugName": "interested",
          "moduleRoutes": "interested",
              "sequenceId": 27
            },
          {
              "moduleName": "Show case panel",
              "moduleAliasName": "Show case panel",
          "moduleSlugName": "show_case_panel",
          "moduleRoutes": "showcase-panel",
              "sequenceId": 28
            }

    ];
    this.list.map(val => {
      val?.childModule?.forEach((data: any) => {
        data.selected = false;
      })
      // console.log('val at onint', this.list)
    })

  let a=this.list?.map((element) => {
    console.log('e',element)
      return {...element, childModule:
element?.childModule?.filter((subElement) => subElement.selected == true)}
    })

  let final=  a.filter(val=>{
      console.log('val',val.childModule!=undefined)
      return val.childModule!=undefined && val.childModule.length>0;
    })
    console.log('a',a,final)


ref:https://stackoverflow.com/questions/38375646/filtering-array-of-object
s-with-arrays-based-on-nested-value
```

101) parent and child checkbox

html

```
<div class="form-group">
<div class="custom-control custom-checkbox mr-sm-2" *ngFor="let data of
list">
<input type="checkbox" class="custom-control-input"
(change)="checkparent(data,0)" id="{{ data.moduleAliasName }}"
[ngModelOptions]="{ standalone: true }" [(ngModel)]="data.selected" />
<label class="custom-control-label" for="{{ data.moduleAliasName
}}">{{data.moduleName}}-{{data.selected}}</label>
<br>
<div class="custom-control custom-checkbox a ml-9" *ngFor="let datas of
data.childModule">
<input type="checkbox" class="custom-control-input"
(change)="checkparent(data,1)" id="{{ datas.moduleSlugName }}"
[ngModelOptions]="{ standalone: true }" [(ngModel)]="datas.checked" />
<label class="custom-control-label" id="{{ datas.moduleSlugName }}"
>{{datas.moduleName}}-{{datas.checked}}</label>
</div>
</div>
</div>
```

ts

```
  public list = [];
  public checkedList = [];
  public templist = [
    {
      moduleName: 'Perks & Benefits',
      moduleAliasName: 'Perks & Benefits',
      moduleSlugName: 'benefits',
      moduleRoutes: 'benefit',
      sequenceId: 4,
    },
    {
      moduleName: 'Tech Stack',
      moduleAliasName: 'Tech Stack',
```

```
    moduleSlugName: 'tech_stack',
    moduleRoutes: 'tech-stack',
    sequenceId: 5,
  },
  {
    moduleName: 'Online Feed',
    moduleAliasName: 'Online Feed',
    moduleSlugName: 'online_feed',
    moduleRoutes: 'online-feed',
    sequenceId: 12,
  },
  {
    moduleName: 'FAQs',
    moduleAliasName: 'FAQs',
    moduleSlugName: 'question_answer',
    moduleRoutes: 'question-answer',
    sequenceId: 13,
  },
  {
    moduleName: 'Company Culture',
    moduleAliasName: 'Company Culture',
    moduleSlugName: 'company_culture',
    moduleRoutes: 'company-culture',
    sequenceId: 14,
  },
  {
    moduleName: 'Testimonials',
    moduleAliasName: 'Testimonials',
    moduleSlugName: 'testimonials',
    moduleRoutes: 'testimonials',
    sequenceId: 15,
  },
  {
    moduleName: 'About',
    moduleAliasName: 'About',
    moduleSlugName: 'about',
    moduleRoutes: 'about',
    sequenceId: 1,
    childModule: [
      {
```

```
        moduleName: 'About Company',
        moduleAliasName: 'About Company',
        moduleSlugName: 'adout_company',
        moduleRoutes: 'about-company',
        sequenceId: 2,
      },
      {
        moduleName: 'Founder story',
        moduleAliasName: 'Founder story',
        moduleSlugName: 'founder_story',
        moduleRoutes: 'founder-story',
        sequenceId: 3,
      },
    ],
  },
  {
    moduleName: 'Team',
    moduleAliasName: 'Team',
    moduleSlugName: 'team',
    moduleRoutes: 'team',
    sequenceId: 6,
    childModule: [
      {
        moduleName: 'Core Team',
        moduleAliasName: 'Core Team',
        moduleSlugName: 'core_team',
        moduleRoutes: 'core-team',
        sequenceId: 7,
      },
      {
        moduleName: 'Other Team',
        moduleAliasName: 'Other Team',
        moduleSlugName: 'other_team',
        moduleRoutes: 'other-team',
        sequenceId: 8,
      },
    ],
  },
  {
    moduleName: 'Gallery',
```

```
    moduleAliasName: 'Gallery',
    moduleSlugName: 'gallery',
    moduleRoutes: 'gallery',
    sequenceId: 9,
    childModule: [
      {
        moduleName: 'Image',
        moduleAliasName: 'Image',
        moduleSlugName: 'image',
        moduleRoutes: 'image',
        sequenceId: 10,
      },
      {
        moduleName: 'Video',
        moduleAliasName: 'Video',
        moduleSlugName: 'video',
        moduleRoutes: 'video',
        sequenceId: 11,
      },
    ],
  },
];
ngOnInit() {
  this.list = JSON.parse(JSON.stringify(this.templist));
  this.list.forEach((val) => {
    if (val) {
      val.selected = false;
    }
    if (val.childModule) {
      val.childModule.forEach((datass) => {
        datass.checked = false;
      });
    }
  });
}
name = 'Angular ' + VERSION.major;

checkparent(val,type) {
   console.log('this.list',val)
   if(type==0){
```

```
      if (val.selected == true) {
        this.list.forEach((parent) => {
          if (val.moduleSlugName == parent.moduleSlugName) {
            parent?.childModule?.forEach((child) => {
              child.checked = true;
            });
          }
        });
      } else {
        this.list.forEach((parent) => {
          if (val.moduleSlugName == parent.moduleSlugName) {
            parent.childModule.forEach((child) => {
              child.checked = false;
            });
          }
        });
      }



    }

    if(type==1){
      this.list.forEach((parent) => {
        if (val.moduleSlugName == parent.moduleSlugName) {
          let selectedAll = parent?.childModule?.every(function (item:
any) {
            return item.checked === false;
          });
          parent.selected = !selectedAll
          console.log('seeeeeee',selectedAll)
        }
      });



    }



   let a = this.list?.map((element) => {
     // console.log('e', element)
```

```
      return {
        ...element,
        childModule: element?.childModule?.filter(
          (subElement: any) => subElement?.checked == true
        ),
      };
    });

    let final = a.filter((val) => {
      // console.log('val', val.childModule != undefined)
      return (
        val.selected == true ||
        (val.childModule != undefined && val.childModule.length > 0)
      );
    });

    let result = final.map(val=>val.childModule == undefined ? (delete
val.childModule, val):val);

    console.log('final', result);
  }


(or)

  checkparent(val,type) {
     console.log('this.list',val)
     if(type==0){
        this.list.forEach((parent) => {
          if (val.moduleSlugName == parent.moduleSlugName) {
            parent?.childModule?.forEach((child) => {
              child.checked = val.selected == true?true:false
            });
          }
        });
     }

    else if(type==1){
      this.list.forEach((parent) => {
        if (val.moduleSlugName == parent.moduleSlugName) {
```

```
          let selectedAll = parent?.childModule?.every(function (item:
any) {
            return item.checked === false;
          });
          parent.selected = !selectedAll
          console.log('seeeeeee',selectedAll)
        }
      });


     }



    let a = this.list?.map((element) => {
      return {
        ...element,
        childModule: element?.childModule?.filter(
          (subElement: any) => subElement?.checked == true
        ),
      };
    });

    let final = a.filter((val) => {
      return (
        val.selected == true ||
        (val.childModule != undefined && val.childModule.length > 0)
      );
    });

    let result = final.map(val=>val.childModule == undefined ? (delete
val.childModule, val):val);

  }


102)remove object based on condition
```

```
var obj = [{ id: 1, field: "finance"}, { id: null, field: "service"}, {
id: 2, field: "information"}]
result = obj.map(val=>val.id ? val : (delete val.id, val));
console.log(result);

[
  {
    "id": 1,
    "field": "finance"
  },
  {
    "field": "service"
  },
  {
    "id": 2,
    "field": "information"
  }
]
```

103)Ellipsis directive

declaration:[EllipsifyMeDirective]

ellipsify-me.directive.ts

```
import { Directive, HostListener, ElementRef, AfterViewInit, Renderer2 }
from '@angular/core';

@Directive({
  selector: '[appEllipsifyMe]'
})
export class EllipsifyMeDirective implements AfterViewInit {
  domElement: any;
  constructor(private renderer: Renderer2, private elementRef: ElementRef)
{
    this.domElement = this.elementRef.nativeElement; // to get DOM element
and store it in global variable
    // setting compulsory required styles to the DOM element
```

```
    const elipsifyme = {
      'text-overflow': 'ellipsis',
      'overflow': 'hidden',
      'white-space': 'nowrap',
    };
    Object.keys(elipsifyme).forEach(newStyle => {
      this.renderer.setStyle(
        this.domElement, `${newStyle}`, elipsifyme[newStyle]
      );
    });
  }

  // to check and add title attribute on the element at the time when
application renders first time.
  ngAfterViewInit(): void {
    // to see effect try removing below two lines and check if the title
is added at the first time rendering.
    this.renderer.setProperty(this.domElement, 'scrollTop', 1);
    this.isTitleAttribute();
  }

  @HostListener("window:resize", ["$event.target"])
  isTitleAttribute() {
    // to add or remove title attribute on the element when it is changing
width.
    (this.domElement.offsetWidth < this.domElement.scrollWidth) ?
      this.renderer.setAttribute(this.domElement, 'title',
this.domElement.textContent) :
      this.renderer.removeAttribute(this.domElement, 'title');
  }

}

html
<div appEllipsifyMe>{{longText}}</div>
<div appEllipsifyMe class="z-index">{{longText}}</div>


https://stackblitz.com/edit/ng-ellipsiswithtooltip?embed=1&file=src/app/ap
p.component.html
```

```
104) Array

1. indexOf()-find an item's index
  const myPets=['Dog','Cat','Hamster'];
  myPets.indexOf('Cat')→//1
2. join()-createastring from array items
  const myPets=['Dog','Cat','Hamster'];
  myPets.join('and');//'Dog and Cat and Hauster
3. slice()-split an array at given index(es)
  const myPets=['Dog','Cat','Hamster'];
  myPets.slice(1);//['Cat','Hamster']
  myPets.slice(1,2);//['Cat']

4. splice()-split an array and/or insert new items
  const myPets=['Dog','Cat','Hamster'];
 myPets.splice(1,2,'Lizard')//['Dog','Lizard']
5. concat()-concatenate one or more arrays
 const myPets=['Dog','Cat','Hamster'];
 const myFlying Pets=['Bird'];
 const myWaterPets=['Fish'];
 const allPets=myPets.concat(myFlying Pets,myWaterPets);
 // allPets→['Dog','Cat','Hamster','Bird','Fish']
6. forEach()-loop over an array and access
each item
  const myPets=['Dog','Cat','Hamster'];
  myPets.forEach(pet console.log(pet));
7. filter()-createanew array based onafilter
 const myPets=['Dog','Cat','Hamster'];
 const threeLetterPets=myPets.filter(petpet.length=3)
 // threeLetterPets→['Dog','Cat']
8. map()-loop over an array and run some
operation on each item without mutating
the original array
 const myPets=['Dog','Cat','Hamster'];
  const lovedPets=myPets.map(pet'${pet}')
  // LovedPets→['Dog'Cat','Hamster']

9. flat()-flatten an array toasingle dimension
```

```
 const values=[1,2,[7],3,[1,2],4]
 console.log(values.flat())
 //[1,2,7,3,1,2,4]

const new_array = values.flat([depth]);

10. reduce()-runacallback on each item and
reduce the array toasingle value
 const values=[1,6,7,1,3,4];
 const total=values.reduce((total,currentVal)total+currentNum);
 // total→22
11. findIndex()-finds the index of an item
based onacondition
 const people=[{name:'David'},{name:'Peter'),{name:'Alex'}]
 const peterIndex=people.findIndex(person person.name='Peter');
 // peterIndex→1

12. every()-check if every item meetsacondition
  const values=[1,6,7,1,3,4];
  console.log(values.every(val val<8))
  // true
13. find()-find first value that meetsacondition
 const values=[4,1,7,2,5,7,9,25];
 const firstValueOverSeven=values.find(val⇒val>7);
 // firstValueOverSeven→9
14. some()-check if some values meetacondition
  const values=[1,6,7,1,3,4];
  console.log(values.some(val⇒val>7))
  // false
15. sort()-sorts an array
  const values=[4,1,7,2,5];
  const names=['David','Alexander','Peter'];
  values.sort((a,b)a-b);//[1,2,4,5,7]
  names.sort();//['Alexander','David','Peter']

16)push,pop,shift,unshift

const month=['Jan','April','June","Nov','Dec'1;
const nunArr=[602,3820,2003,622,42,589,841;
//*-*-*-*-*This Methods Returns New length of Array*-*-*-*-*
```

```
console.log(month.push('October'));
// Expected Output:['Jan',"april","June','Nov","Doc","October1
console.log(month.unshift('Feb"));
   Expected Output'Feb','Jan','April,June","Now","Dec','October1
   *-*-*-*-*This Methods Returns Deleted Element*-*-*-*-*
console.log(numArr.pop());
// Expected Output:84
console.log(numArr.shift());
// Expected Output:602

17) sort
const months=['Jan','April','June','Nov','Dec'];
const numArray=[602,3020,1003,622,42,589,84];
console.log(months.sort());
// Expected output['April','Dec','Jan','June','Nov)
console.log(numArray.sort());
// Expected output[1983,3820,42,589,42,589,602,622,84]
//*-*-*-*-*For properly sort integer element*-*-*-*-*
console.log(numArray.sort((a,b)⇒a-b));
// Expected Output[42,84,589,602,622,1003,3020]

18)indexof ,lastindex of ,includes
const months=['Jan','April','June','Nov','Dec','April');
console.log(months.indexOf('April'));
// Expected output:1
console.log(months.lastIndexOf('April'));
7/Expected output:5
console.log(months.includes('June'));
// Expected output:true
19)find,find index,filter

const prices=[100,200,300,400,500,600,7001;
console.log(prices.find((price)price<400));
// Expected output:100
console.log(prices.findIndex((price)price>400));
// Expected output:4
console.log(prices.filter((price)price 400));
// Expected output:[100,200,300,400]

20)slice,splice
```

```
const month=['Jan','April','June','Nov','Dec'l;
// Update from index0to2step.
console.log(month.splice(0,2,'Feb','March'));
// Expected output:['Jan','April']
console.log(month.slice(1,3));
// Expected output:['March','June']
var arr = ["orange", "mango", "banana", "sugar", "tea"]; var removed =
arr.splice(2, 0, "water"); console.log("After adding 1: " + arr );
console.log("removed is: " + removed);


21)concat,string,join
const girls=['Cecilie','Lone'];
const boys=['Emil','Tobias','Linus'];
console.log(girls.concat(boys));
// Expected Output:['Cecilie','Lone','Emil','Tobias','Linus']
console.log(boys.toString());
// Expected Output:Emil,Tobias,Linus
console.log(boys.join(''));
// Expected Output:Emil Tobias*Linus

21)Fill

let arr = new Array(5).fill(true); // [true, true, true, true, true]

let arr = Array.from({length: 5}, (value, index) => index); // [0, 1, 2,
3, 4]

let arr = Array.from({length: 5}, (value, index) => true); // [true, true,
true, true, true]


105)array of object filter based on conditions.

const array = [{ "Item": "A", "Quantity": 2 }, { "Item": "B", "Quantity": 7
}, { "Item": "C", "Quantity": 7 },{ "Item": "D", "Quantity": 7 },{ "Item":
"E", "Quantity": 7 },{ "Item": "F", "Quantity": 1 }]; const filteredValues =
array.filter((el) => el.Quantity === 7).map(elem => elem.Item);
console.log(filteredValues)
```

```
["B","C","D","E"]
```

references link:

1)Number only Directive

https://stackblitz.com/edit/angular-numbers-only-directive?file=app%2Fnumbers-only.directive.ts

2)Regex pattern

https://www.freakyjolly.com/angular-allow-only-numbers-or-alphanumeric-in-input-restrict-other-characters-using-keypress-event/

3)Swiper

https://swiperjs.com/demos

4)file download progress

https://stackblitz.com/edit/angular-image-download-with-progressbar-7ckjst?file=src%2Fapp%2Fapp.component.ts

https://stackblitz.com/edit/angular-upload-file-with-progress-bar-7cum2v?file=src%2Fapp%2Fapp.component.ts

https://stackblitz.com/edit/angular-file-upload-nakooi?file=app%2Fapp.component.ts

https://stackblitz.com/edit/angular-file-download-progress-mtvy96?file=src%2Fapp%2Fapp.component.ts,src%2Fapp%2Fdownload.service.ts

https://nils-mehlhorn.de/posts/angular-file-download-progress

Ref

https://stackblitz.com/edit/angular-ty36de?file=src%2Fapp%2Fapp.component.ts,src%2Fapp%2Fapp.component.html,src%2Fapp%2Fapp.component.css

Pagination

https://stackblitz.com/edit/angular-gw2bd9?file=src%2Fapp%2Fpagination-customization.ts,src%2Fapp%2Fpagination-customization.html,src%2Fapp%2Fpagination-customization.module.ts

1)Number only Directive

https://stackblitz.com/edit/angular-numbers-only-directive?file=app%2Fnumbers-only.directive.ts

2)Regex pattern

https://www.freakyjolly.com/angular-allow-only-numbers-or-alphanumeric-in-input-restrict-other-characters-using-keypress-event/

3)Swiper

https://swiperjs.com/demos

4)file download progress

https://stackblitz.com/edit/angular-image-download-with-progressbar-7ckjst?file=src%2Fapp%2Fapp.component.ts

https://stackblitz.com/edit/angular-upload-file-with-progress-bar-7cum2v?file=src%2Fapp%2Fapp.component.ts

https://stackblitz.com/edit/angular-file-upload-nakooi?file=app%2Fapp.component.ts

https://stackblitz.com/edit/angular-file-download-progress-mtvy96?file=src%2Fapp%2Fapp.component.ts,src%2Fapp%2Fdownload.service.ts

https://nils-mehlhorn.de/posts/angular-file-download-progress

5)nvm install
https://tecadmin.net/how-to-install-nvm-on-ubuntu-20-04/

6)how to uninstall angular/update cli

https://stackoverflow.com/questions/39566257/how-to-uninstall-upgrade-angular-cli

[redacted]