File 1: Products search component (Typescript)

```typescript
export class ProductsSearchComponent implements OnInit {

  @Input() categoryId: number;
  @Input() searchString: string;
  @Input() isTopRecord: boolean;

  products = {} as ProductModelResponse;
  totalRecords = 0;
  isLoaded = false;
  sortT: string;
  sortV: string;
  pMin: number;
  pMax: number;
  sortIndex: number = 0;

  baseUrl = environment.apiUrl;

  constructor(private _http: HttpClient,
    private _categoryService: CategoriesService,
    private _productService: ProductService,
    private _authService: AuthenticationService) {
  }

  ngOnInit(): void {
    if(this.categoryId != undefined && this.searchString != undefined)
    {
      this.getProducts();
    }
  }

  getProducts(){
    this.isLoaded = false;
      console.log('calling product-search.component');
      //console.log(this.searchString);

      this._productService.getProducts().subscribe(data=> {
        this.products = {products: data.products} as ProductModelResponse;
        this.totalRecords = this.products.products.length;
        this.isLoaded = true;
      }, error=>{console.log(error)});
  }

  sortChanged() {
    this.sortT = null;
    this.sortV = null;

    if(this.sortIndex == 0 || this.sortIndex == 1) {
      this.sortT = 'dt';
      this.sortV = this.sortIndex == 1 ? 'asc' : 'desc';
    }
    else if(this.sortIndex == 2 || this.sortIndex == 3) {
      this.sortT = 'p';
      this.sortV = this.sortIndex == 3 ? 'asc' : 'desc';
    }

    if(this.sortT && this.sortV) {
      this._productService.filter.sortT = this.sortT;
      this._productService.filter.sortV = this.sortV;
    }
    this.getProducts();
  }

  priceChanged() {
    this._productService.filter.pMin = this.pMin;
    this._productService.filter.pMax = this.pMax;
    this.getProducts();
    return false;
  }

}
```

Server File(Products.js)

```
router.get('/', function(req, res) {

    id = req.query.id;

    if(id) {
        sqlManager.productDetails(id, '1', function(err, result, images) {
            if (err) {
                res.status(500).json({status:'Failed', message: err.message});
                return
            }
            if(result.length == 0) {
                res.status(404).json({status: 'Success', message: 'Product not
found.'});
                return
            }
            res.status(200).json({status: 'Success', products: result,
productImages:images});
        });
    }
    else {
        sqlManager.searchProducts(req.query.sq, req.query.cat, req.query.pMin,
req.query.pMax, req.query.sortT, req.query.sortV , function(err, result) {
            if (err) {
                res.status(500).json({status:'Failed', message: err.message});
                return
            }
            if(result.length == 0) {
                res.status(200).json({status: 'Success', message: 'No Products
Found.', products: []});
                return
            }
            res.status(200).json({status: 'Success', products: result});
        })
    }

});
```

Server File SQL.

```
//Product Status,
function searchProducts(searchQuery, cat, pMin, pMax, sortT, sortV, cb) {
    var queryString = "SELECT p.id, p.title, p.location, p.status, p.category,
p.price, p.thumbnail, p.createdAt FROM Product p WHERE p.status = 1";
    var paramsCount = 0
    if (searchQuery) {
        paramsCount++;
    }
    if (cat) {
        paramsCount++;
    }
    if (pMin) {
        paramsCount++;
    }
    if (pMax) {
        paramsCount++;
    }

    if(paramsCount > 0) {
        queryString += ' AND'
    }

    if (searchQuery) {
        queryString = queryString+" p.title LIKE '%"+searchQuery+"%'";
        paramsCount--;
        queryString += paramsCount > 0 ? ' AND' : '';
    }
    if (cat) {
        queryString = queryString+" p.category = "+cat;
        paramsCount--;
        queryString += paramsCount > 0 ? ' AND' : '';
    }
    if (pMin) {
        queryString = queryString+ " p.price >= "+pMin;
        paramsCount--;
        queryString += paramsCount > 0 ? ' AND' : '';
    }
    if (pMax) {
        queryString = queryString+ " p.price <= "+pMax;
    }
    if (sortT && sortT === 'dt' && sortV && (sortV === 'asc' || sortV === 'desc'))
{
        queryString = queryString+" ORDER BY  p.createdAt "+(sortV === "asc" ?
'ASC' : 'DESC');
    }
    else if (sortT && sortT === 'p' && sortV && (sortV === 'asc' || sortV ===
'desc')) {
        queryString = queryString+" ORDER BY  p.price "+(sortV === "asc" ? 'ASC' :
'DESC');
    }
    else {
        queryString = queryString+" ORDER BY p.id DESC";
    }

    connection.query(queryString,
        function (err, rows) {
            if (err) cb(err);
            else cb(undefined, rows);
        });
}
```

HTML File for Front-end of Searching

```
<app-loader-animation [isShow]="!isLoaded"></app-loader-animation>

<div *ngIf="isLoaded">

<div class="alert alert-dismissible alert-success" *ngIf="isTopRecord == false">
  <strong>Found {{totalRecords}} records...</strong>
</div>


<div clas="row" *ngIf="isTopRecord == false">

</div>

<div class="mt-2 mb-1" *ngIf="isTopRecord == false">
  <div class="form-row">
    <div class="col-lg-3  mb-3">
      <input type="text" class="form-control form-control-lg" id="boxMin"
name="boxMin" type="text" #boxMin (keyup.enter)="(priceChanged())"
[(ngModel)]="pMin" placeholder="Minimum Price">
    </div>
    <div class="col-lg-3 mb-3">
```

```html
      <input type="text" class="form-control form-control-lg" id="boxMax"
name="boxMax" type="text" #boxMax (keyup.enter)="(priceChanged())"
[(ngModel)]="pMax" placeholder="Maximum Price">
    </div>
    <div class="col-lg-6 mb-3">
      <select class="form-control form-control-lg" id="sortingOption"
(change)="sortChanged()" [(ngModel)]="sortIndex">
        <option value="0">Date(Recent First)</option>
        <option value="1">Date(Oldest First)</option>
        <option value="2">Price(Highest First)</option>
        <option value="3">Price(Lowest First)</option>
    </select>
    </div>
  </div>
</div>


<div class="container">
    <div class="row">
      <div class="col-md-4" *ngFor="let prod of products?.products">
        <div class="card mb-4 box-shadow">
          <a [routerLink]="['/home/detail', {id: prod.id}]">
            <img
              class="card-img-top"
              alt="Thumbnail [100%x225]"
              style="height: 225px; width: 100%; display: block"
              src="{{baseUrl+'/'+prod.thumbnail}}"
              data-holder-rendered="true"
            />
          </a>
          <div class="card-body">
            <div
              class="d-flex justify-content-between align-items-center"
            >
              <h4 style="height: 44px;">{{prod.title}}</h4>
            </div>
            <div>
              <button type="button" class="btn btn-primary"
[routerLink]="['/home/detail', {id: prod.id}]">
                {{prod.price}} €
              </button>
            </div>
          </div>
        </div>
      </div>

    </div>
  </div>
<div>
```