Front-end:
sell-item/sell-item.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormControl, ReactiveFormsModule, FormGroup, FormBuilder, Validators } from '@angular/forms';
import {AddItemDTO} from '../_models/add-item-dto';
import { ProductService } from '../_services/product.service';
import { Category, CategoryModelResponse } from 'src/app/_models/category-model';
import { CategoriesService } from '../_services/categories.service';
import { AuthenticationService } from '../_services/authentication.service';
@Component({
  selector: 'app-sell-item',
  templateUrl: './sell-item.component.html',
  styleUrls: ['./sell-item.component.css']
})

export class SellItemComponent implements OnInit {

  submittedSuccessfully = false;
  errorMessage = false;
  errorMessageText = 'All Fields are manadatory'
  addItemFormGroup: FormGroup;
  category = {} as CategoryModelResponse;
  selectedCategoryID;
  productName = "";
  productDesc = "";
  productPrice = 0;
  selectedFile: File[] = [];

  constructor(private fb: FormBuilder,
    private _productService: ProductService, private _categoriesService: CategoriesService, private _authService:
AuthenticationService) {

  }

  ngOnInit(): void {
```

```
  // get the list of categories
  this.getCategories();

  // add form validation
  this.addItemFormGroup = this.fb.group({
    productName: ['', Validators.required],
    productDesc: ['', Validators.required],
    productPrice: ['', Validators.required]
  });
}

// HTML Events

selectCategoryOption(id) {
  this.selectedCategoryID = id;
}

processFile(event) {
  const files: File[] = event.target.files;
  this.selectedFile = files;
  console.log(event);
}


// API CALLINGS

getCategories() {
  // call category service to get categories
  this._categoriesService.getAllCategories().subscribe(data=>{
    let p = data.categories;
    this.category = {categories: p} as CategoryModelResponse;
    console.log(data);
  }, error=> {
    console.log(error);
  })
}
```

```
callProductPosting() {

  let loggedInUser = this._authService.currentUser;


  this.errorMessage = false;

  const fd = new FormData();
  fd.append('title', this.addItemFormGroup.get('productName')?.value);
  fd.append('desc', this.addItemFormGroup.get('productDesc')?.value);
  fd.append('price', this.addItemFormGroup.get('productPrice')?.value);
  fd.append('owner', loggedInUser.user.id.toString());
  fd.append('category', this.selectedCategoryID);
  // fd.append('images', JSON.stringify(this.selectedFile));
  for (let file of this.selectedFile) {
    fd.append('images', file);
  }

  console.log("Ready to roll");
  this._productService.addProduct(fd).subscribe(data=>{
    window.scroll(0,0);
    this.submittedSuccessfully = true;
  }, error=> {
    console.log(error);
  });
}

// ACTIONS
submit(){

  console.log('working');
  // console.log(this.ddlCategory);

  this.errorMessage = false;
  if(this.addItemFormGroup.valid)
  {
    if (this.selectedFile == null) {
```

```
      this.errorMessage = true;
      window.scroll(0,0);
      this.errorMessageText = "Please select image(s).";
    } else {
      this.callProductPosting();
    }
  } else {
    this.errorMessage = true;
    window.scroll(0,0);
    this.validateFields();
  }
}

// VALIDATIONS

validateFields() {
  if (this.addItemFormGroup.get('productName')?.value == "") {
    this.errorMessageText = "Please enter product name.";
  } else if (this.addItemFormGroup.get('productDesc')?.value == "") {
    this.errorMessageText = "Please enter product description.";
  } else if (this.addItemFormGroup.get('productPrice')?.value == "") {
    this.errorMessageText = "Please enter product price.";
  } else {
    let priceStr = this.addItemFormGroup.get('productPrice')?.value as string;
    let priceNum = Number(priceStr);
    if (priceNum == null) {
      this.errorMessageText = "Please enter correct price!";
    } else if (priceNum < 0) {
      this.errorMessageText = "Please enter price greater then zero!";
    }
  }
 }
}
```

sell-item/sell-item.component.html

```
<div class="seeIItem" *ngIf="!submittedSuccessfully">

   <div class="alert alert-dismissible alert-primary" *ngIf="errorMessage" >
      <button type="button" class="close" data-dismiss="alert">&times;</button>
      <strong>Error:</strong> {{ errorMessageText }}
     </div>

   <h2 class="mb-5">Sell an item</h2>

   <h5 class="mb-5">Please fill the following details</h5>


   <form (ngSubmit)="submit()" [formGroup]="addItemFormGroup"  name="sellItemForm">
      <div class="form-row">
         <div class="form-group col-lg-6 ">
            <label>Category</label>
            <select [(ngModel)]="selectedCategoryID" (change)="selectCategoryOption()" class="form-control form-
control-lg" >
               <!-- <select class="form-control form-control-lg" *ngIf="category"  id="ddlCategory" name="ddlCategory"
[(ngModel)]="ddlCategory" > -->
               <option *ngFor="let cats of category?.categories" value="cats.id" >{{cats.name}}</option>
            </select>

         </div>
      </div>

      <div class="form-row">
         <div class="form-group col-lg-6 ">
            <label>Title of the item</label>
            <input required formControlName="productName" type="text" class="form-control form-control-lg"
id="txtProductName" name="txtProductName" placeholder="Enter Product or Service name"/>
         </div>
      </div>

      <div class="form-row">
         <div class="form-group col-lg-6 ">
            <label>Description</label>
```

```
            <textarea required formControlName="productDesc" class="form-control form-control-lg" id="txtDescription" rows="10" placeholder="Describe your product"></textarea>
          </div>
        </div>

        <div class="form-row">
          <div class="form-group col-lg-6 ">
            <label>Price (€)</label>
            <input required type="number" min="0" pattern="[0-9]*" data-politespace data-grouplength="3" data-delimiter="," formControlName="productPrice" class="form-control form-control-lg" id="txtPrice"  placeholder="Enter Product or Service Price" />
          </div>
        </div>

        <div class="form-row">
          <div class="form-group col-lg-6 ">
            <label>Select Images</label>
            <input required
            formControlName="productImage"
            type="file"
            accept="image/*"
            (change)="processFile($event)"
            class="form-control-file" multiple>
          </div>
        </div>

        <div class="form-row">
          <div class="form-group col-lg-6 mt-5">
            <button type="submit" class="btn btn-primary btn-lg">Submit</button>

          </div>
        </div>


    </form>
</div>
```

```
<div *ngIf="submittedSuccessfully">
    <div class="d-flex justify-content-center">
        <img class="img-fluid" src="assets/images/success.png" />
    </div>
    <div class="d-flex justify-content-center mt-3">
        <h3>Your ad has successfully posted</h3>
    </div>
    <div class="d-flex justify-content-center mt-3">
        <h5>It will appear on the site once Administrator apporved it</h5>
    </div>
</div>
```

Back-end
backend/sql.js

```
function productDetails(id, user_id, cb) {
    // var queryString = "SELECT * FROM dbo.Product p where p.status != 2 AND p.id=" + id;
    var queryString = "SELECT p.id, p.title, p.desc, p.owner, p.category, p.createdAt, p.status, p.price, p.location,
p.thumbnail, u.name as ownerName, u.email as ownerEmail FROM dbo.Product as p inner join dbo.User as u on u.id
= p.owner where p.id=" + id + " GROUP BY p.id ";
    connection.query(queryString,
        function (err, rows) {
            if (err) cb(err);
            else {
                var productImage = "SELECT * FROM dbo.ProductImage p where p.productId=" + id;
                connection.query(productImage,
                    function (err, images) {
                        if (err) cb(err);
                        else {
                            cb(undefined, rows, images);
                        }
                    }
                );
            }
        });
    // var productDetail =connection.query(queryString);
    // console.log(productDetail);
```

```
}
```