İSTANBUL TOPKAPI ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

SUNUCU TARAFLI WEB PROGRAMLAMA

MOBİLYA SATAN E-TİCARET SİTESİ

Eda İpek Sanlı

20040101006

Bahar, 2024

Özellikler	Çalıştığını gösteren ekran görüntülerinin olduğu sayfa	Ön uç kodlarının olduğu sayfa	Arka uç kodlarının olduğu sayfa
Ürünlerin ya da eserlerin bazılarının (en önemli olanlar) veri tabanından çekilerek listelendiği bir ana sayfa (home page).	4	1 2	8
Ürünlerin ya da eserlerin hepsinin listelendiği sayfalama (pagination) uygulanmış bir arama “search” sayfası	5	3	9
Ürün ya da eser ismi ile arama (search) özelliği	6	3	10
Ürün ya da eser kategorisi ile arama (search) özelliği	7	3	11

1.Home.tsx

```
import React, { useEffect, useState } from "react";
import "../styles/home.css";
import { Link } from "react-router-dom";
import Title from "../components/Title/Title";
import { Container, Row, Col } from "reactstrap";
import Banner from "../assets/slider.png";
import Advantages from "../components/Advantages/Advantages";
import Carousel from "react-multi-carousel";
import "react-multi-carousel/lib/styles.css";
import ProductCard from "../components/Product/ProductCard";

import { Furniture } from "./Shop";
const responsive = {
  superLargeDesktop: {
    breakpoint: { max: 4000, min: 3000 },
    items: 5,
  },
  desktop: {
    breakpoint: { max: 3000, min: 1024 },
    items: 3,
  },
  tablet: {
    breakpoint: { max: 1024, min: 464 },
    items: 2,
  },
  mobile: {
    breakpoint: { max: 464, min: 0 },
    items: 1,
  },
};
const Home: React.FC = () => {
  const [editorsPickProducts, setEditorsPickProducts] = useState<Furniture[]>(
    []
  );
  useEffect(() => {
    async function getData() {
      const res = await fetch("http://localhost:8080/api/editors-pick");
      const data = await res.json();
      setEditorsPickProducts(data);
    }
    getData();
  }, []);
  return (
    <Title title={"Home"}>
      <section className="banner-section">
        <Container>
          <Row>
            <Col lg="6" md="6">
              <div className="banner-content">
                <p>Transform Your Space with Timeless Elegance:</p>
                <h2>Discover Our Exquisite Furniture Collection Today!</h2>
                <p>
                  Elevate your home with handcrafted pieces designed to inspire.
                </p>


                <button className="shop-btn">
                  <Link to="/shop">Shop Now!</Link>
                </button>
              </div>
            </Col>
            <Col lg="6" md="6">
              <div className="banner-img">
                <img src={Banner} alt="banner-img" />
              </div>
            </Col>
          </Row>
        </Container>
      </section>

      <Advantages />

      <div className="carousel">
        <h1>Editor's Pick</h1>
        <Carousel responsive={responsive}>
          {editorsPickProducts.map((product) => {
            return (
              <ProductCard
                imageId={product.imageId}
                productName={product.name}
                productPrice={product.price}
                productType={product.category}
                key={product.id}
              />
            );
          })}
        </Carousel>
        ;
      </div>
    </Title>
  );
};
```

```
export default Home;
```

2.productCard.tsx

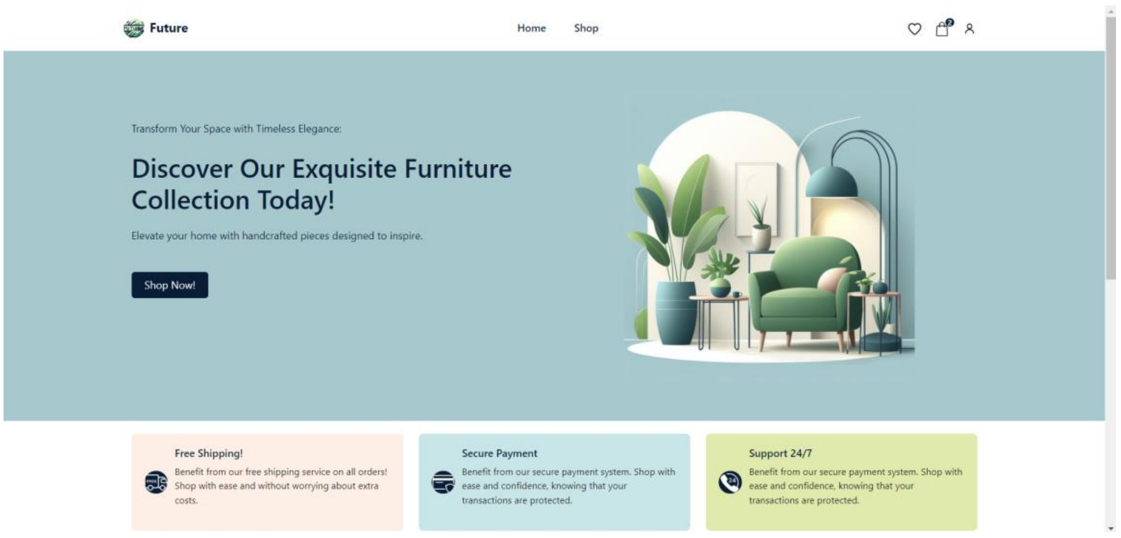
```
import React from "react";
import "../Product/productCard.css";
interface ProductCardProps {
  imageId: string;
  productName: string;
  productPrice: string;
  productType: string;
}
const ProductCard: React.FC<ProductCardProps> = ({
  imageId,
  productName,
  productPrice,
  productType,
}) => {
  return (
    <div className="item">
      <div className="product-img">
        <img
          src={`http://localhost:8080/images/${imageId}.jpg`}
          alt="product"
        />
      </div>
      <div className="product-info">
        <h3 className="product-name">{productName}</h3>
        <p>{productType}</p>
      </div>

      <div className="price-shop">
        <h4>{productPrice}</h4>
        <span>
          <img
            src="https://img.icons8.com/ios/50/ffffff/add--v1.png"
            alt="add--v1"
          />
        </span>
      </div>
    </div>
  );
};

export default ProductCard;
```

3.Shop.tsx

```
import React, { useEffect, useState } from "react";

import Title from "../components/Title/Title";
import CommonSection from "../components/CommonSection/CommonSection";
import {
  Container,
  Row,
  Col,
  Pagination,
  PaginationItem,
  PaginationLink,
} from "reactstrap";
import "../styles/shop.css";
import ProductCard from "../components/Product/ProductCard";
import { useSearchParams } from "react-router-dom";

export interface Furniture {
  id: number;
  name: string;
  price: string;
  category: string;
  imageId: string;
}

type Paged<T> = {
  content: T;
  totalPages: number;
  pageable: {
    pageNumber: number;
  };
};

const Shop: React.FC = () => {
  const [furnitures, setFurnitures] = useState<Paged<Furniture[]>>({
    content: [],
    totalPages: 0,
    pageable: {
      pageNumber: 0,
    },
  });
  const [searchQuery, setSearchQuery] = useState("");
  const [selectedCategory, setSelectedCategory] = useState<string>("");
  const [params] = useSearchParams();
  const page = params.get("page") ?? 0;
  useEffect(() => {
    const getDatas = async () => {
      const url =
`http://localhost:8080/api?name=${searchQuery}&category=${selectedCategory}&page=${page}`;
      console.log(url);
      const response = await fetch(url);
      const body = await response.json();
      setFurnitures(body);
    };
    getDatas();
  }, [searchQuery, selectedCategory]);
  const [categories, setCategories] = useState<string[]>([]);
  useEffect(() => {
```

```
  const getCategories = async () => {
    const res = await fetch("http://localhost:8080/api/categories");
    const data = await res.json();
    setCategories(data);
  };
  getCategories();
}, [setCategories]);

return (
  <Title title={"Shop"}>
    <CommonSection title="Products" />

    <section>
      <Container>
        <Row>
          <Col lg="3" md="3">
            <div className="filter category">
              <select
                id="categorycal"
                onChange={(e) => {
                  const value = e.currentTarget.value;
                  if (value === "Select category") {
                    return;
                  }
                  setSelectedCategory(value);
                }}
              >
                <option>Select category</option>
                {categories.map((category) => {
                  return (
                    <option value={category} key={category}>
                      {category}
                    </option>
                  );
                })}
              </select>
            </div>
          </Col>
          <Col lg="6" md="6">
            <div className="search">
              <input
                type="text"
                placeholder="Search... "
                id="search-input"
                onChange={(e) => {
                  console.log(e.currentTarget.value);
                  setSearchQuery(e.currentTarget.value ?? "");
                }}
              />
              <span>
                <img
                  width="50"
                  height="50"
                  src="https://img.icons8.com/ios/50/0a1d37/search--v1.png"
                  alt="search--v1"
                />
              </span>
            </div>
          </Col>
          <Col lg="3" md="3">
            <div className="filter sort">
              <select id="sorting-filter">
                <option>Sort By</option>
                <option value="ascending">Ascending</option>
                <option value="descending">Descending</option>
              </select>
            </div>
          </Col>
        </Row>
      </Container>

      {furnitures.content.map((furniture) => {
        return (
          <div key={furniture.id}>
            <ProductCard
              imageId={furniture.imageId}
              productName={furniture.name}
              productPrice={furniture.price}
              productType={furniture.category}
            />
          </div>
        );
      })}
    </section>
    <Pagination>
      <PaginationItem>
        <PaginationLink href="?page=0" first></PaginationLink>
      </PaginationItem>
      {Array.from({ length: furnitures.totalPages }, (_, i) => {
        return (
          <PaginationItem
            key={i}
            active={furnitures.pageable.pageNumber === i}
          >
            <PaginationLink href={`?page=${i}`}>{i + 1}</PaginationLink>
          </PaginationItem>
        );
      })}
      <PaginationItem>
        <PaginationLink
          href={`?page=${furnitures.totalPages - 1}`}
          last
        ></PaginationLink>
      </PaginationItem>
    </Pagination>
  </Title>
);
};

export default Shop;
```

4.Home Page



Transform Your Space with Timeless Elegance:

Discover Our Exquisite Furniture Collection Today!

Elevate your home with handcrafted pieces designed to inspire.

Free Shipping!

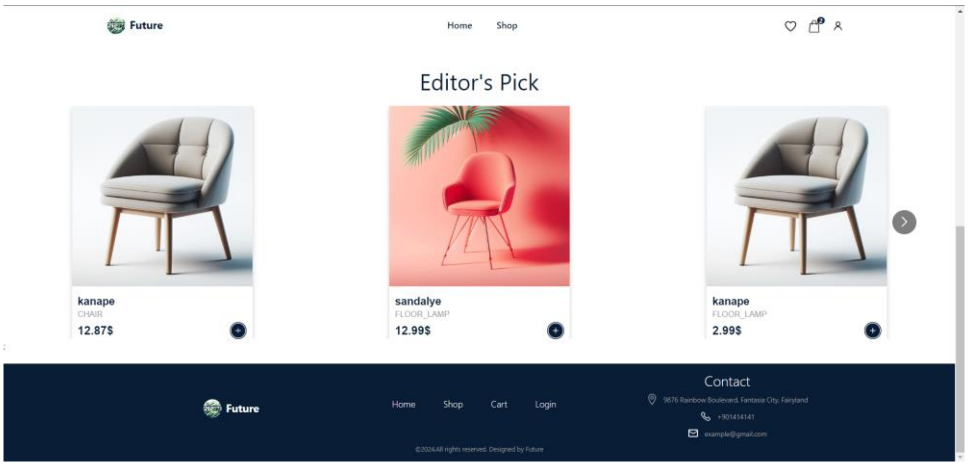
Benefit from our free shipping service on all orders! Shop with ease and without worrying about extra costs.

Secure Payment

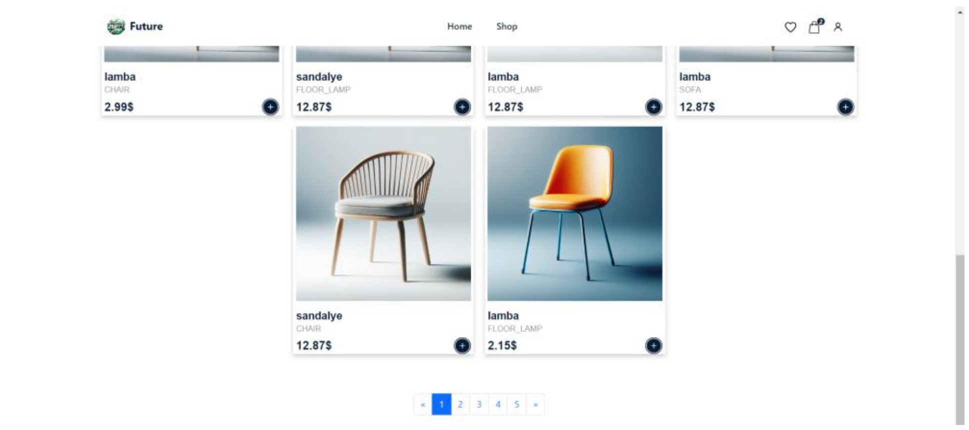
Benefit from our secure payment system. Shop with ease and confidence, knowing that your transactions are protected.

Support 24/7

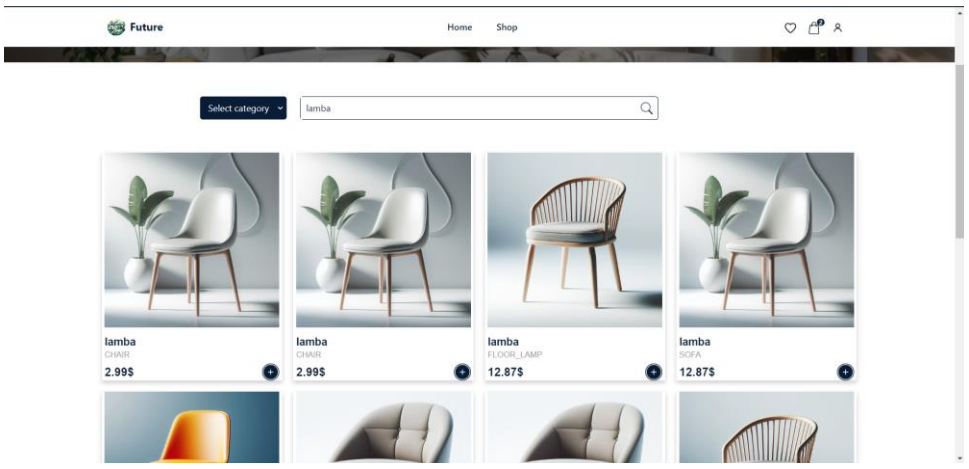
Benefit from our secure payment system. Shop with ease and confidence, knowing that your transactions are protected.



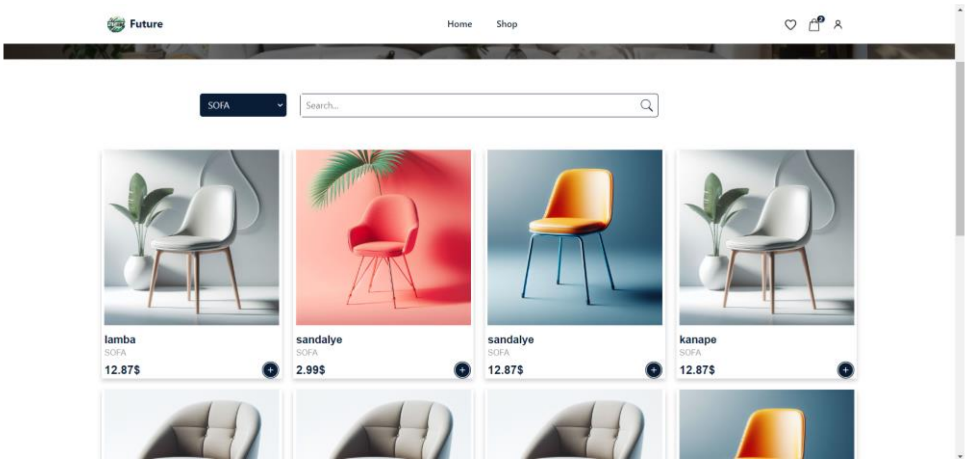
5.Pagination



6.Search



7.Kategorikal Arama



8. Önemli Ürünlerin Veri Tabanından Çekilmesi

```
@GetMapping
@RequestMapping(path = "/editors-pick")
public List<Furniture> getEditorsPick() {
    return furnitureService.getEditorsPick();
}
```

```
public List<Furniture> getEditorsPick() {
    return furnitureRepository.getEditorsPick();
}
```

```
@Query("SELECT f FROM Furniture f WHERE f.isEditorsPick = true")
List<Furniture> getEditorsPick();
```

9.Pagination

```
public Page<Furniture> getAllFurnitures(Pageable pageable) {
    return furnitureRepository.findAll(pageable);
}
```

```
@GetMapping
public Page<Furniture> getAllFurnitures(
        @RequestParam(required = false, defaultValue = "0") Integer page,
        @RequestParam(required = false, defaultValue = "9") Integer size
) {
    Pageable pageable = Pageable.ofSize(size).withPage(page);
    return furnitureService.getAllFurnitures(pageable);
}
```

10.Search

```java
@GetMapping(params = {"name", "category"})
public Page<Furniture> searchFurnitureByName(
        @RequestParam String name,
        @RequestParam(required = false) Optional<FurnitureCategory> category,
        @RequestParam(required = false, defaultValue = "0") Integer page,
        @RequestParam(required = false, defaultValue = "10") Integer size
) {
    Pageable pageable = Pageable.ofSize(size).withPage(page);
    if (name.isBlank() && category.isEmpty()) {
        return furnitureService.getAllFurnitures(pageable);
    }
    if (name.isBlank()) {
        return furnitureService.getFurnitureByCategory(category.get(), pageable);
    }
    if (category.isEmpty()) {
        return furnitureService.searchFurnitureByName(name, pageable);
    }
    return furnitureService.searchFurnitureByNameAndCategory(name, category.get(), pageable);
}
```

```java
public Page<Furniture> searchFurnitureByName(String name, Pageable pageable) {
    return furnitureRepository.searchFurnitureByName(name, pageable);
}
```

```java
public Page<Furniture> searchFurnitureByNameAndCategory(String name, Furniture.FurnitureCategory
category, Pageable pageable) {
    return furnitureRepository.searchFurnitureByNameAndCategory(name, category, pageable);
}
```

```java
@Query("SELECT f FROM Furniture f WHERE f.name LIKE %:name%")
Page<Furniture> searchFurnitureByName(String name, Pageable pageable);
```

```java
@Query("SELECT f FROM Furniture f WHERE f.name LIKE %:name% AND f.category = %:category%")
Page<Furniture> searchFurnitureByNameAndCategory(String name, Furniture.FurnitureCategory
category, Pageable pageable);
```

11. Kategori ile Arama

```java
@GetMapping
@RequestMapping(path = "/categories")
public List<FurnitureCategory> getCategories() {
    return Arrays.asList(FurnitureCategory.values());
}
```

```java
Page<Furniture> getFurnitureByCategory(Furniture.FurnitureCategory category, Pageable pageable);
```

```java
public Page<Furniture> getFurnitureByCategory(Furniture.FurnitureCategory category, Pageable
pageable) {
    return furnitureRepository.getFurnitureByCategory(category, pageable);
}
```