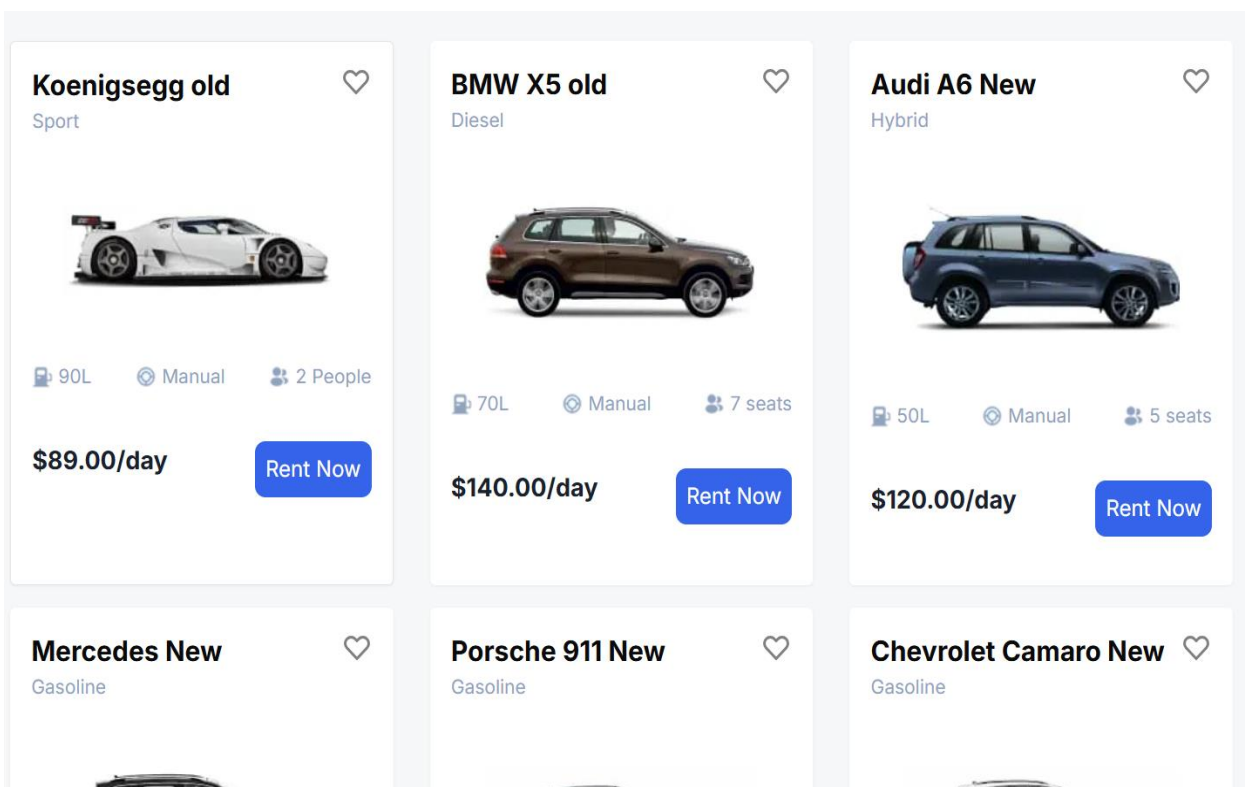


# Day 4 - Dynamic Frontend Components - MORENT

## 1. Functional Deliverables:


- The product listing page with dynamic data






- Individual product detail pages

### Koenigsegg old - finest design and unmatched acceleration

combining safety, comfort, and a futuristic, elegant look for a truly thrilling driving experience.





## Koenigsegg old

☆☆☆☆☆ 440+ Reviewers

Discover the Koenigsegg old, where comfort, reliability, and style seamlessly blend, offering the perfect ride for every journey at Morent — your ultimate driving experience.


Type Car Sport	Capacity 2 People
Steering Manual	Gasoline 90L

**\$89.00/day**


[Rent Now](#)




## • Category Filters

[ALL](#) [SPORT](#) [DIESEL](#) [HYBRID](#) [GASOLINE](#) [SUV](#) [SEDAN](#) [HATCHBACK](#) [ELECTRIC](#)


**Tesla Model 3 old** 

Electric







 100kWh  Manual  5 seats

**\$100.00/day** [Rent Now](#)

**Tesla Model 3 New** 


Electric




 100kWh  Manual  5 seats




**\$120.00/day** [Rent Now](#)

[ALL](#) [SPORT](#) [DIESEL](#) [HYBRID](#) [GASOLINE](#) [SUV](#) [SEDAN](#) [HATCHBACK](#) [ELECTRIC](#)


**MG ZX Exclusive Old** 

Hatchback







 90L  Manual  2 People

**\$89.00** [Rent Now](#)

**MG ZX Exclusive New** 

Hatchback



 90L  Manual  2 People


**\$99.00** [Rent Now](#)

- Wishlist Component

### Your Wishlist

Koenigsegg old

Sport



90L

Manual


2 People

\$89.00/day

Rent Now

BMW X5 old

Diesel



70L

Manual

7 seats

\$140.00/day

Rent Now

- SearchBar Component

MORENT

or rob room

ro

Chevrolet Camaro New

Rolls-Royce Latest Model

Rolls-Royce old

Rolls-Royce New


Chevrolet Camaro old

Rolls-Royce Black

Your Wishli

Koenigsegg ol


Sport



90L

Manual

2 People



70L

Manual

7 seats

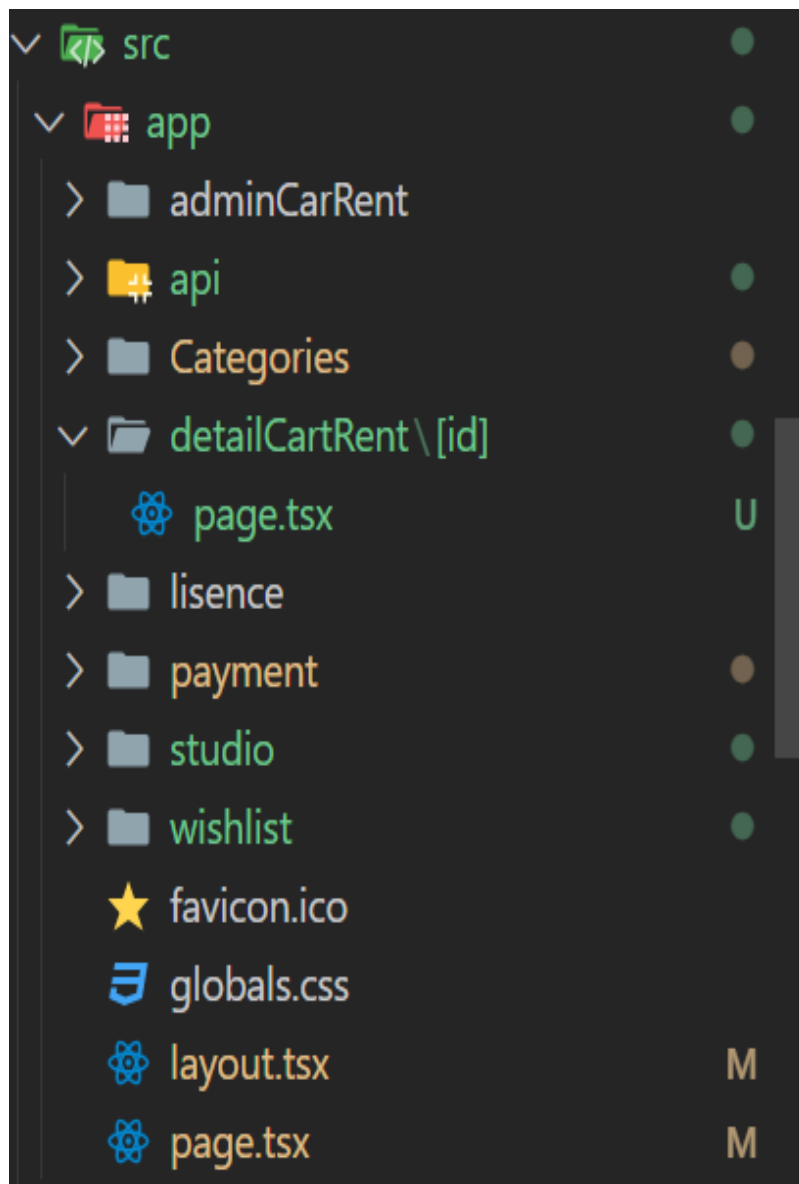
# Code Deliverables:

- SearchBar

```
    /* Search Bar */  
    <div className="relative">  
      <input  
        type="text"  
        className="h-[48px] mt-4 w-full rounded-[10px] border placeholder:text-[#596780] px-4"  
        placeholder="Search something here"  
        value={searchQuery}  
        onChange={(e) => setSearchQuery(e.target.value)} // Update search query  
        onKeyDown={(e) => e.key === "Enter" && handleSearchSubmit()} // Trigger search on Enter key  
      </input>  
      {searchQuery && filteredProducts.length > 0 && (  
        <div className="absolute bg-white w-full mt-1 border border-gray-300 rounded-md shadow-lg z-10">  
          <ul>  
            Pieces: Comment | Pieces: Explain  
            {filteredProducts.map((product: any) => (  
              <li  
                key={product._id}  
                className="px-4 py-2 text-black hover:bg-gray-200 cursor-pointer"  
              >  
                <Link href={` /detailCarRent/${product._id}`} passHref>  
                  {product.name}  
                </Link>  
              </li>  
            ))}  
          </ul>  
        </div>  
      )}</div>
```

- Product Detail (Dynamic Routing - detailCarRent)

## File Structure:



# Code

```
interface Params {
  params: {
    id: string;
  };
}

const CarDetailPage = async ({ params }: {params: {id:string}}) => {

  const carDetail = await client.fetch(
    `*[_type == "car" && _id == "${params.id}"]{
      _id,
      name,
      type,
      seatingCapacity,
      fuelCapacity,
      pricePerDay,
      originalPrice,
      tags,
      "imageUrl": image.asset->url
    }[0]`
  );

  if (!carDetail) {
    return <div>Car with ID {params.id} not found</div>;
  }

  return (
    <main className="h-full w-full bg-[#F6F7F9]">
      <DetailHero carDetail={carDetail} />
      <Reviews />
      <PopularCar />
    </main>
  );
};
```

## • Product Listing

```
const CategoriesMain: React.FC = () => {
  const router = useRouter();
  const searchParams = useSearchParams(); // Get URL params
  const [cars, setCars] = useState<Car[]>([]); // Array of cars with type definition
  const [carTypes, setCarTypes] = useState<string[]>([]); // Array of car types
  const [selectedType, setSelectedType] = useState<string>("all"); // Selected car type
  const [searchQuery, setSearchQuery] = useState<string>(""); // Search query

  // Fetch car data from Sanity
  Pieces: Comment | Pieces: Explain
  useEffect(() => {
    const fetchCars = async () => {
      try {
        const query = `*[_type == "car"]{
          _id,
          name,
          type,
          "image": image.asset->url,
          fuelCapacity,
          transmission,
          seatingCapacity,
          pricePerDay,
          originalPrice,
          tags
        }`;

        const result: Car[] = await client.fetch(query);
        Pieces: Comment | Pieces: Explain
        const carsWithImageUrls = result.map((car) => ({
          ...car,
          imageUrl: urlFor(car.image), // Adding imageUrl as a string
        }));
        setCars(carsWithImageUrls);

        // Extract unique car types
        const uniqueTypes = Array.from(
```



# API Integration:

```
// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
```

```
// Create Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2021-08-31'
});
```

Pieces: Comment | Pieces: Explain

```
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop()
    });
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}
```

Pieces: Comment | Pieces: Explain

```
async function importData() {
  try {
    console.log('Fetching car data from API...');

    // API endpoint containing car data
    const response = await axios.get('https://sanity-nextjs-application.vercel.app/api/hackathon/template7');
    const cars = response.data;

    console.log(`Fetched ${cars.length} cars`);

    for (const car of cars) {
      console.log(`Processing car: ${car.name}`);
    }
  }
}
```

# Dynamic Routing:

```
interface Params {
  params: {
    id: string;
  };
}

const CarDetailPage = async ({ params }: {params: {id:string}}) => {

  const carDetail = await client.fetch(
    `*[_type == "car" && _id == "${params.id}"]{
      _id,
      name,
      type,
      seatingCapacity,
      fuelCapacity,
      pricePerDay,
      originalPrice,
      tags,
      "imageUrl": image.asset->url
    }[0]`
  );

  if (!carDetail) {
    return <div>Car with ID {params.id} not found</div>;
  }

  return (
    <main className="h-full w-full bg-[#F6F7F9]">
      <DetailHero carDetail={carDetail} />
      <Reviews />
      <PopularCar />
    </main>
  );
};

export default CarDetailPage;
```