NITTE EDUCATION TRUST | NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

AN AUTONOMOUS INSTITUTION AFFILIATED TO VTU, BELGAUM, ACCREDIATED by NAAC ('A+' Grade)

YELAHANKA, BANGALORE-560064

A Report on Estate Commerce

Submitted in partial fulfilment of the requirement for the LA-2 of 6th semester
Full Stack Development(21ISE652)

BACHELOR OF ENGINEERING IN INFORMATION SCIENCE AND ENGINEERING

By

Akshay Sinha	1NT21IS023
Veera Goutham	1NT21IS075
Sanya Gupta	1NT21IS142
Seema	1NT21IS145

Under the Guidance of

Mrs. Vani K S
Assistant Professor

Department of Information Science and Engineering

Nitte Meenakshi Institute of Technology, Bengaluru - 560064

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
(Accredited by NBA

Tier-1) 2023-24

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VTU, BELGAUM)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

Accredited by NBA Tier-1

DECLARATION

We,AkshaySinha(1NT21IS023),VeeraGoutham(1NT21IS075),SanyaGupta(1NT21IS142),Seema(1NT21IS145) bonafide students of Nitte Meenakshi Institute of Technology, hereby declare that the project entitled ‘’ Estate Commerce ” submitted in partial fulfillment for the award of Bachelor of Engineering Information Science and Engineering of the Visvesvaraya Technological University, Belgaum during the academic year 2023-2024 is our original work and the project has not formed the basis for the award of any other degree, fellowship or any other similar titles.

INTRODUCTION

Introduction to One-MNC Estate

One-MNC Estate is a sophisticated real estate commerce platform built on the MERN (MongoDB, Express.js, React, Node.js) stack. It provides a seamless experience for users looking to buy, rent, or sell properties. Our application ensures a secure and transparent transaction process, preventing third-party interference and protecting user data.

Using the MERN Stack for One-MNC Estate

The One-MNC Estate application leverages the MERN stack, a powerful combination of technologies that ensures a seamless, efficient, and scalable solution for real estate commerce. Here's a breakdown of how each component of the MERN stack is utilized in this project:

1. MongoDB:

- **Database:** MongoDB is a NoSQL database that stores user credentials, property listings, and transaction details.
- **Scalability:** Its flexible schema allows for easy scalability and quick adaptation to changing data needs.
- **Data Security:** Ensures that sensitive user information and transaction data are securely stored.

2. Express.js:

- **Backend Framework:** Express.js is used to build the backend of the application, handling routing, middleware, and server-side logic.
- **API Development:** Provides robust APIs for interacting with the MongoDB database, allowing for CRUD (Create, Read, Update, Delete) operations on user profiles and property listings.
- **Middleware:** Utilizes middleware for request processing, authentication, and error handling.

3. React:

- **Frontend Library:** React is used to build the user interface, providing a dynamic and responsive user experience.
- **Component-Based Architecture:** Allows for the creation of reusable components, such as forms, buttons, and listing cards, enhancing development efficiency and maintainability.
- **State Management:** Manages application state using React's built-in hooks and context API, ensuring a smooth and interactive user experience.

4. Node.js:

- **Runtime Environment:** Node.js provides the runtime environment for executing JavaScript code on the server side.
- **Performance:** Its event-driven, non-blocking I/O model ensures high performance and scalability, handling multiple simultaneous connections efficiently.
- **Package Management:** Utilizes npm (Node Package Manager) to manage project dependencies, making it easy to install and update libraries and modules.

Implementing One-MNC Estate with MERN:

By leveraging the MERN stack, One-MNC Estate is able to provide a robust, scalable, and user-friendly platform for real estate transactions. The synergy between MongoDB, Express.js, React, and Node.js ensures that both the frontend and backend are seamlessly integrated, delivering a high-quality user experience and reliable performance.

Using the MERN stack, One-MNC Estate ensures that user data and transactions are handled securely, listings are easily searchable and manageable, and the overall experience for buying, renting, or selling properties is efficient and enjoyable.

1. Home Page

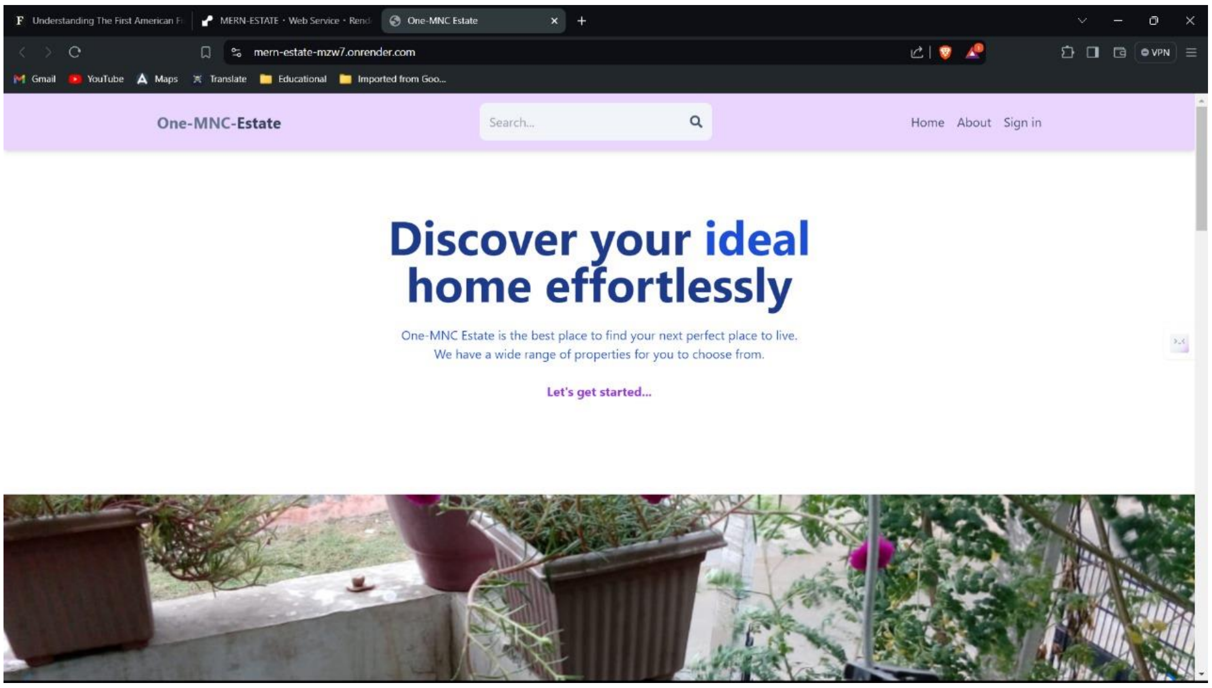
- **Overview:**
  - Introduction to the platform
  - Featured

Code : Home.jsx

```
import { useEffect, useState } from 'react';
import { Link } from 'react-router-dom';
import { Swiper, SwiperSlide } from 'swiper/react';
import { Navigation } from 'swiper/modules';
import SwiperCore from 'swiper';
import 'swiper/css/bundle';
import ListingItem from '../components/ListingItem';

export default function Home() {
  const [offerListings, setOfferListings] = useState([]);
  const [saleListings, setSaleListings] = useState([]);
  const [rentListings, setRentListings] = useState([]);
  SwiperCore.use([Navigation]);

  useEffect(() => {
    const fetchOfferListings = async () => {
      try {
        const res = await fetch('/api/listing/get?offer=true&limit=4');
        const data = await res.json();
        setOfferListings(data);
        fetchRentListings();
      } catch (error) {
        console.log(error);
      }
    };
    const fetchRentListings = async () => {
      try {
        const res = await fetch('/api/listing/get?type=rent&limit=4');
        const data = await res.json();
        setRentListings(data);
        fetchSaleListings();
      } catch (error) {
        console.log(error);
      }
    };

    const fetchSaleListings = async () => {
      try {
        const res = await fetch('/api/listing/get?type=sale&limit=4');
        const data = await res.json();
        setSaleListings(data);
      } catch (error) {
        console.log(error);
      }
    };
    fetchOfferListings();
  }, []);

  return (
    <div className='min-h-screen bg-gradient-to-br from-white-50 to-peach-100'>
      {/* top */}
      <div className='flex flex-col gap-6 py-20 px-5 sm:px-10 lg:px-28 max-w-6xl mx-auto
text-center'>
        <h1 className='text-blue-900 font-bold text-3xl lg:text-6xl'>
          Discover your <span className='text-blue-700'>ideal</span>
          <br />
          home effortlessly
        </h1>
        <div className='text-blue-700 text-sm sm:text-base'>
          One-MNC Estate is the best place to find your next perfect place to
          live.
          <br />
          We have a wide range of properties for you to choose from.
        </div>
        <Link
          to={'/search'}
          className='text-sm sm:text-base text-purple-600 font-bold hover:underline'
        >
          Let's get started...
        </Link>
      </div>

      {/* swiper */}
      <Swiper navigation className='my-10'>
        {offerListings &&
          offerListings.length > 0 &&
          offerListings.map((listing) => (
            <SwiperSlide key={listing._id}>
              <div
                style={{
                  background: url(${listing.imageUrls[0]}) center no-repeat,
                  backgroundSize: 'cover',
                }}
                className='h-[300px] sm:h-[500px]'
              ></div>
            </SwiperSlide>
          ))}
      </Swiper>

      {/* listing results for offer, sale and rent */}
      <div className='max-w-6xl mx-auto p-3 flex flex-col gap-8 my-10'>
        {offerListings && offerListings.length > 0 && (
          <div>
            <div className='my-3 flex justify-between items-center'>
              <h2 className='text-xl sm:text-2xl font-semibold text-blue-900'>Recent
offers</h2>
              <Link className='text-sm text-purple-600 hover:underline'
to={'/search?offer=true'}>Show more offers</Link>
            </div>
            <div className='flex flex-wrap gap-4'>
              {offerListings.map((listing) => (
                <ListingItem listing={listing} key={listing._id} />
              ))}
            </div>
          </div>
        )}
        {rentListings && rentListings.length > 0 && (
          <div>
            <div className='my-3'>
              <h2 className='text-xl sm:text-2xl font-semibold text-blue-900'>Recent places for
rent</h2>
              <Link className='text-sm text-blue-800 hover:underline'
to={'/search?type=rent'}>Show more places for <b>rent</b></Link>
            </div>
            <div className='flex flex-wrap gap-4'>
              {rentListings.map((listing) => (
                <ListingItem listing={listing} key={listing._id} />
              ))}
            </div>
          </div>
```

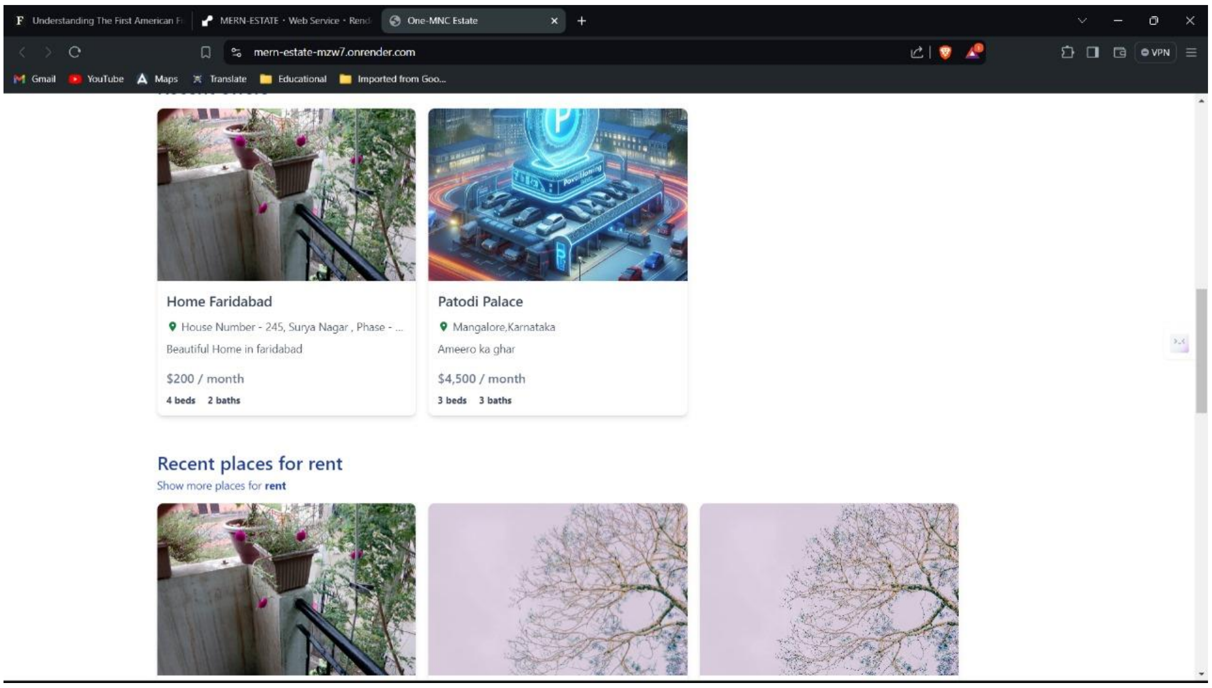
```
    )}
    {saleListings && saleListings.length > 0 && (
     <div>
      <div className='my-3 '>
       <h2 className='text-xl sm:text-2xl font-semibold text-blue-900'>Recent places for
<b>sale</b></h2>
       <Link className='text-sm text-blue-800 hover:underline'
to={'/search?type=sale'}>Show more places for sale</Link>
      </div>
      <div className='flex flex-wrap gap-4'>
       {saleListings.map((listing) => (
        <ListingItem listing={listing} key={listing._id} />
       ))}
      </div>
     </div>
    )}
   </div>
  </div>
 );
}
```

Snapshot of home page:



Discover your ideal home effortlessly

One-MNC Estate is the best place to find your next perfect place to live.
We have a wide range of properties for you to choose from.


Understanding The First American F
MERN-ESTATE · Web Service · Rend
One-MNC Estate
mern-estate-mzw7.onrender.com
Gmail
YouTube
Maps
Translate
Educational
Imported from Goo...
VPN

Home Faridabad
House Number - 245, Surya Nagar , Phase - ...
Beautiful Home in faridabad
$200 / month
4 beds 2 baths

Patodi Palace
Mangalore,Karnataka
Ameero ka ghar
$4,500 / month
3 beds 3 baths

Recent places for rent
Show more places for rent

User Authentication and Profiles:

- Users can sign up and log in using their Google accounts.
- User credentials are securely stored in the database.
- Personalized user profiles to manage listings and view purchase history.

Profile.jsx

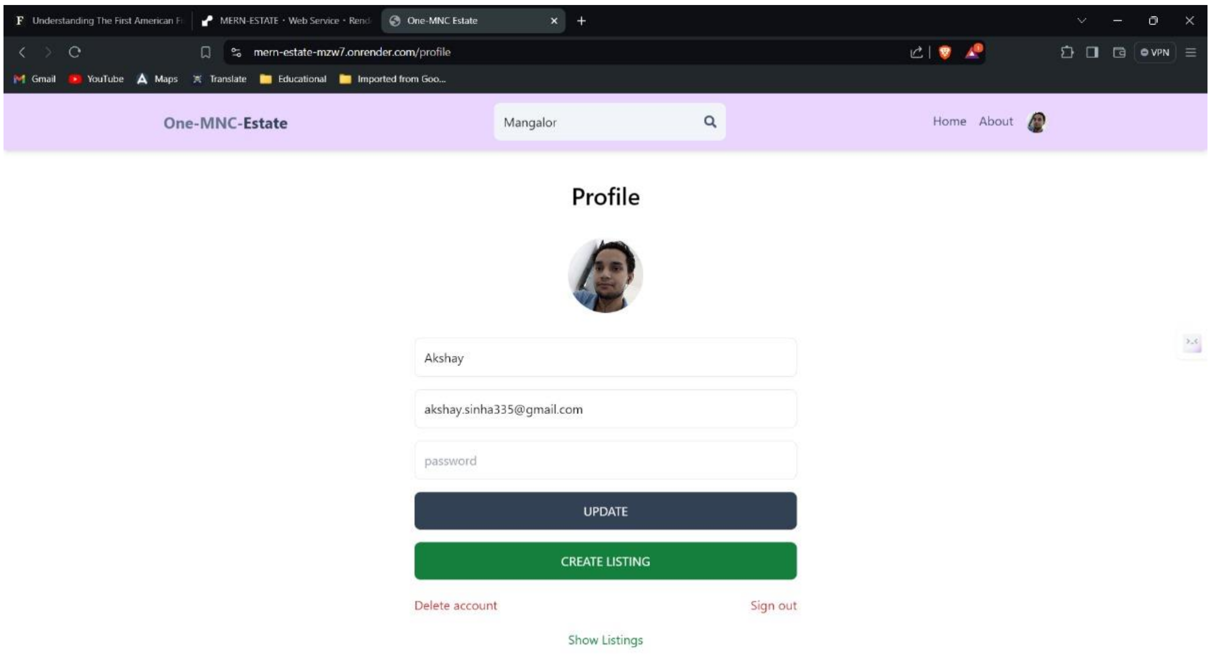
```
import { useSelector } from 'react-redux';
import { useRef, useState, useEffect } from 'react';
import {
  getDownloadURL,
  getStorage,
  ref,
  uploadBytesResumable,
} from 'firebase/storage';
import { app } from '../firebase';
import {
  updateUserStart,
  updateUserSuccess,
  updateUserFailure,
  deleteUserFailure,
  deleteUserStart,
  deleteUserSuccess,
  signOutUserStart,
} from '../redux/user/userSlice';
import { useDispatch } from 'react-redux';
import { Link } from 'react-router-dom';
export default function Profile() {
  const fileRef = useRef(null);
  const { currentUser, loading, error } = useSelector((state) => state.user);
  const [file, setFile] = useState(undefined);
  const [filePerc, setFilePerc] = useState(0);
  const [fileUploadError, setFileUploadError] = useState(false);
  const [formData, setFormData] = useState({});
  const [updateSuccess, setUpdateSuccess] = useState(false);
  const [showListingsError, setShowListingsError] = useState(false);
  const [userListings, setUserListings] = useState([]);
  const dispatch = useDispatch();

  // firebase storage
  // allow read;
  // allow write: if
  // request.resource.size < 2 * 1024 * 1024 &&
  // request.resource.contentType.matches('image/.*')
```

```
useEffect(() => {
  if (file) {
    handleFileUpload(file);
  }
}, [file]);

const handleFileUpload = (file) => {
  const storage = getStorage(app);
  const fileName = new Date().getTime() + file.name;
  const storageRef = ref(storage, fileName);
  const uploadTask = uploadBytesResumable(storageRef, file);

  uploadTask.on(
    'state_changed',
    (snapshot) => {
      const progress =
        (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
      setFilePerc(Math.round(progress));
    },
    (error) => {
      setFileUploadError(true);
    },
    () => {
      getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) =>
        setFormData({ ...formData, avatar: downloadURL })
      );
    }
  );
};

const handleChange = (e) => {
  setFormData({ ...formData, [e.target.id]: e.target.value });
};

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    dispatch(updateUserStart());
    const res = await fetch(/api/user/update/${currentUser._id}, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(formData),
    });
    const data = await res.json();
    if (data.success === false) {
      dispatch(updateUserFailure(data.message));
      return;
    }
    dispatch(updateUserSuccess(data));
    setUpdateSuccess(true);
  } catch (error) {
    dispatch(updateUserFailure(error.message));
  }
};

const handleDeleteUser = async () => {
  try {
    dispatch(deleteUserStart());
    const res = await fetch(/api/user/delete/${currentUser._id}, {
      method: 'DELETE',
    });
    const data = await res.json();
    if (data.success === false) {
      dispatch(deleteUserFailure(data.message));
      return;
    }
    dispatch(deleteUserSuccess(data));
  } catch (error) {
    dispatch(deleteUserFailure(error.message));
  }
};

const handleSignOut = async () => {
  try {
    dispatch(signOutUserStart());
    const res = await fetch('/api/auth/signout');
    const data = await res.json();
    if (data.success === false) {
      dispatch(deleteUserFailure(data.message));
      return;
    }
    dispatch(deleteUserSuccess(data));
  } catch (error) {
    dispatch(deleteUserFailure(data.message));
  }
};

const handleShowListings = async () => {
  try {
    setShowListingsError(false);
    const res = await fetch(/api/user/listings/${currentUser._id});
    const data = await res.json();
    if (data.success === false) {
      setShowListingsError(true);
      return;
    }
    setUserListings(data);
  } catch (error) {
    setShowListingsError(true);
  }
};

const handleListingDelete = async (listingId) => {
  try {
    const res = await fetch(/api/listing/delete/${listingId}, {
      method: 'DELETE',
    });
    const data = await res.json();
    if (data.success === false) {
      console.log(data.message);
      return;
    }

    setUserListings((prev) =>
      prev.filter((listing) => listing._id !== listingId)
    );
  } catch (error) {
    console.log(error.message);
  }
};
return (
  <div className='p-3 max-w-lg mx-auto'>
    <h1 className='text-3xl font-semibold text-center my-7'>Profile</h1>
    <form onSubmit={handleSubmit} className='flex flex-col gap-4'>
      <input
        onChange={(e) => setFile(e.target.files[0])}
        type='file'
        ref={fileRef}
        hidden
        accept='image/*'
      />
      <img
        onClick={() => fileRef.current.click()}
        src={formData.avatar || currentUser.avatar}
        alt='profile'
        className='rounded-full h-24 w-24 object-cover cursor-pointer self-center mt-2'
      />
      <p className='text-sm self-center'>
        {fileUploadError ? (
          <span className='text-red-700'>
            Error Image upload (image must be less than 2 mb)
          </span>
        ) : filePerc > 0 && filePerc < 100 ? (
          <span className='text-slate-700'>{Uploading ${filePerc}%}</span>
        ) : filePerc === 100 ? (
          <span className='text-green-700'>Image successfully uploaded!</span>
          ) : (
            ''
          )}
        </p>
        <input
          type='text'
          placeholder='username'
          defaultValue={currentUser.username}
          id='username'
          className='border p-3 rounded-lg'
          onChange={handleChange}
        />
        <input
          type='email'
          placeholder='email'
          id='email'
          defaultValue={currentUser.email}
          className='border p-3 rounded-lg'
          onChange={handleChange}
        />
        <input
          type='password'
          placeholder='password'
          onChange={handleChange}
          id='password'
          className='border p-3 rounded-lg'
        />
        <button
          disabled={loading}
          className='bg-slate-700 text-white rounded-lg p-3 uppercase hover:opacity-95 
disabled:opacity-80'
        >
          {loading ? 'Loading...' : 'Update'}
        </button>
        <Link
          className='bg-green-700 text-white p-3 rounded-lg uppercase text-center 
hover:opacity-95'
          to={'/create-listing'}
        >
          Create Listing
        </Link>
      </form>
      <div className='flex justify-between mt-5'>
        <span
          onClick={handleDeleteUser}
          className='text-red-700 cursor-pointer'
        >
          Delete account
        </span>
        <span onClick={handleSignOut} className='text-red-700 cursor-pointer'>
```

```
    Sign out
  </span>
</div>

<p className='text-red-700 mt-5'>{error ? error : ''}</p>
<p className='text-green-700 mt-5'>
  {updateSuccess ? 'User is updated successfully!' : ''}
</p>
<button onClick={handleShowListings} className='text-green-700 w-full'>
  Show Listings
</button>
<p className='text-red-700 mt-5'>
  {showListingsError ? 'Error showing listings' : ''}
</p>

{userListings && userListings.length > 0 && (
  <div className='flex flex-col gap-4'>
    <h1 className='text-center mt-7 text-2xl font-semibold'>
      Your Listings
    </h1>
    {userListings.map((listing) => (
      <div
        key={listing._id}
        className='border rounded-lg p-3 flex justify-between items-center gap-4'
      >
        <Link to={/listing/${listing._id}}>
          <img
            src={listing.imageUrls[0]}
            alt='listing cover'
            className='h-16 w-16 object-contain'
          />
        </Link>
        <Link
          className='text-slate-700 font-semibold  hover:underline truncate flex-1'
          to={/listing/${listing._id}}
        >
          <p>{listing.name}</p>
        </Link>

        <div className='flex flex-col item-center'>
          <button
            onClick={() => handleListingDelete(listing._id)}
            className='text-red-700 uppercase'
          >
            Delete
          </button>
          <Link to={/update-listing/${listing._id}}>
            <button className='text-green-700 uppercase'>Edit</button>
          </Link>
        </div>
```

```
            </div>
          ))}
        </div>
      )}
    </div>
  );
}
```

Snapshot :



Akshay

akshay.sinha335@gmail.com

Sign in and Sign up page:

Signin.jsx

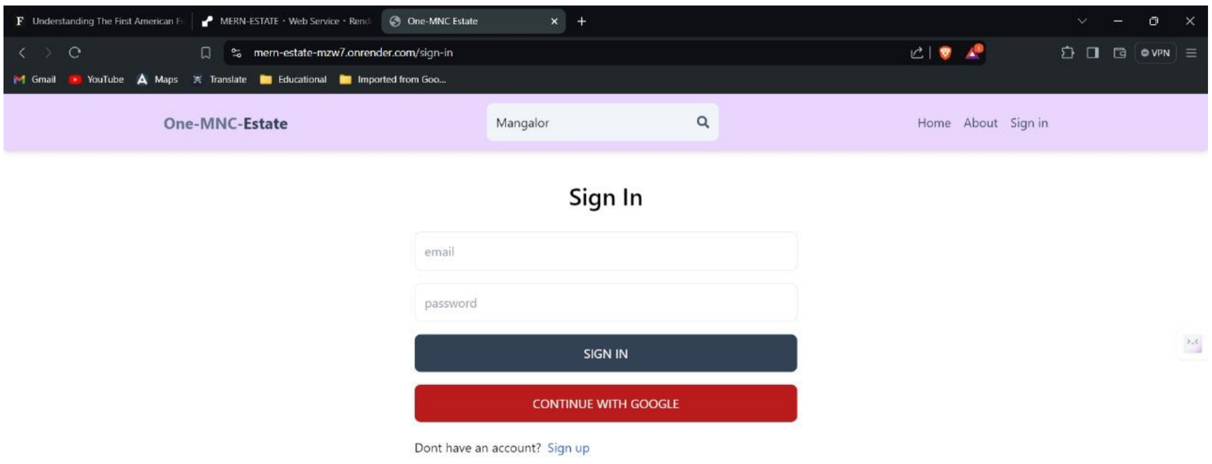
```
import { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import { useDispatch, useSelector } from 'react-redux';
import {
  signInStart,
  signInSuccess,
  signInFailure,
} from '../redux/user/userSlice';
import OAuth from '../components/OAuth';
```

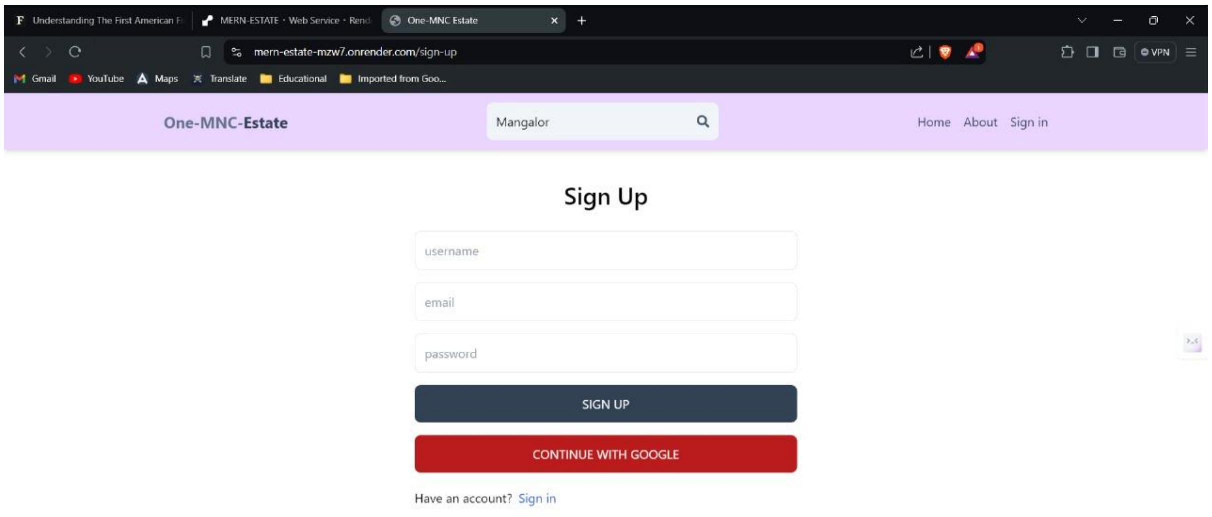
```
export default function SignIn() {
  const [formData, setFormData] = useState({});
  const { loading, error } = useSelector((state) => state.user);
  const navigate = useNavigate();
  const dispatch = useDispatch();
  const handleChange = (e) => {
    setFormData({
      ...formData,
      [e.target.id]: e.target.value,
    });
  };
  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      dispatch(signInStart());
      const res = await fetch('/api/auth/signin', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify(formData),
      });
      const data = await res.json();
      console.log(data);
      if (data.success === false) {
        dispatch(signInFailure(data.message));
        return;
      }
      dispatch(signInSuccess(data));
      navigate('/');
    } catch (error) {
      dispatch(signInFailure(error.message));
    }
  };
  return (
    <div className='p-3 max-w-lg mx-auto'>
      <h1 className='text-3xl text-center font-semibold my-7'>Sign In</h1>
      <form onSubmit={handleSubmit} className='flex flex-col gap-4'>
        <input
          type='email'
          placeholder='email'
          className='border p-3 rounded-lg'
          id='email'
          onChange={handleChange}
        />
        <input
          type='password'
          placeholder='password'
          className='border p-3 rounded-lg'
          id='password'
          onChange={handleChange}
        />

        <button
          disabled={loading}
          className='bg-slate-700 text-white p-3 rounded-lg uppercase hover:opacity-95
disabled:opacity-80'
        >
          {loading ? 'Loading...' : 'Sign In'}
        </button>
        <OAuth/>
      </form>
      <div className='flex gap-2 mt-5'>
        <p>Dont have an account?</p>
        <Link to={'/sign-up'}>
          <span className='text-blue-700'>Sign up</span>
        </Link>
      </div>
      {error && <p className='text-red-500 mt-5'>{error}</p>}
    </div>
  );
}
```

Signup.jsx

```
import { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import OAuth from '../components/OAuth';

export default function SignUp() {
  const [formData, setFormData] = useState({});
  const [error, setError] = useState(null);
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();
  const handleChange = (e) => {
    setFormData({
      ...formData,
      [e.target.id]: e.target.value,
    });
  };
  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      setLoading(true);
      const res = await fetch('/api/auth/signup', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify(formData),
      });
      const data = await res.json();
      console.log(data);
      if (data.success === false) {
        setLoading(false);
        setError(data.message);
        return;
      }
      setLoading(false);
      setError(null);
      navigate('/sign-in');
    } catch (error) {
      setLoading(false);
      setError(error.message);
    }
  };
  return (
    <div className='p-3 max-w-lg mx-auto'>
      <h1 className='text-3xl text-center font-semibold my-7'>Sign Up</h1>
      <form onSubmit={handleSubmit} className='flex flex-col gap-4'>
        <input
          type='text'
          placeholder='username'
          className='border p-3 rounded-lg'
          id='username'
          onChange={handleChange}
        />
        <input
          type='email'
          placeholder='email'
          className='border p-3 rounded-lg'
          id='email'
          onChange={handleChange}
        />
        <input
          type='password'
          placeholder='password'
          className='border p-3 rounded-lg'
          id='password'
          onChange={handleChange}
        />

        <button
          disabled={loading}
          className='bg-slate-700 text-white p-3 rounded-lg uppercase hover:opacity-95
disabled:opacity-80'
        >
          {loading ? 'Loading...' : 'Sign Up'}
        </button>
        <OAuth/>
      </form>
      <div className='flex gap-2 mt-5'>
        <p>Have an account?</p>
        <Link to={'/sign-in'}>
          <span className='text-blue-700'>Sign in</span>
        </Link>
      </div>
      {error && <p className='text-red-500 mt-5'>{error}</p>}
    </div>
  );
}
```

Snapshot :





Property Listings:

- Users can create new property listings by providing necessary details such as property name, description, address, price, and images.
- Listings can be categorized as 'For Sale' or 'For Rent,' with additional options for features like parking spots and furnishing.

Listing.jsx

```
import { useEffect, useState } from 'react';
import { useParams } from 'react-router-dom';
import { Swiper, SwiperSlide } from 'swiper/react';
import SwiperCore from 'swiper';
import { useSelector } from 'react-redux';
import { Navigation } from 'swiper/modules';
import 'swiper/css/bundle';
import {
  FaBath,
  FaBed,
  FaChair,
  FaMapMarkedAlt,
  FaMapMarkerAlt,
  FaParking,
  FaShare,
} from 'react-icons/fa';
import Contact from '../components/Contact';

// https://sabe.io/blog/javascript-format-numbers-
commas#:~:text=The%20best%20way%20to%20format,format%20the%20number%20with
%20commas.

export default function Listing() {
  SwiperCore.use([Navigation]);
  const [listing, setListing] = useState(null);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(false);
  const [copied, setCopied] = useState(false);
  const [contact, setContact] = useState(false);
  const params = useParams();
  const { currentUser } = useSelector((state) => state.user);

  useEffect(() => {
    const fetchListing = async () => {
      try {
        setLoading(true);
        const res = await fetch(/api/listing/get/${params.listingId});
        const data = await res.json();
        if (data.success === false) {
          setError(true);
          setLoading(false);
          return;
        }
        setListing(data);
        setLoading(false);
        setError(false);
      } catch (error) {
        setError(true);
        setLoading(false);
      }
    };
    fetchListing();
  }, [params.listingId]);

  return (
    <main>
      {loading && <p className='text-center my-7 text-2xl'>Loading...</p>}
      {error && (
        <p className='text-center my-7 text-2xl'>Something went wrong!</p>
      )}
      {listing && !loading && !error && (
        <div>
          <Swiper navigation>
            {listing.imageUrls.map((url) => (
              <SwiperSlide key={url}>
                <div
                  className='h-[550px]'
                  style={{
                    background: url(${url}) center no-repeat,
                    backgroundSize: 'cover',
                  }}
                ></div>
              </SwiperSlide>
            ))}
          </Swiper>
          <div className='fixed top-[13%] right-[3%] z-10 border rounded-full w-12 h-12 flex
justify-center items-center bg-slate-100 cursor-pointer'>
            <FaShare
              className='text-slate-500'
              onClick={() => {
                navigator.clipboard.writeText(window.location.href);
                setCopied(true);
                setTimeout(() => {
                  setCopied(false);
                }, 2000);
              }}
            />
          </div>
          {copied && (
            <p className='fixed top-[23%] right-[5%] z-10 rounded-md bg-slate-100 p-2'>
              Link copied!
            </p>
          )}
          <div className='flex flex-col max-w-4xl mx-auto p-3 my-7 gap-4'>
            <p className='text-2xl font-semibold'>
              {listing.name} - ${' '}
              {listing.offer
                ? listing.discountPrice.toLocaleString('en-US')
                : listing.regularPrice.toLocaleString('en-US')}
              {listing.type === 'rent' && ' / month'}
            </p>
            <p className='flex items-center mt-6 gap-2 text-slate-600  text-sm'>
              <FaMapMarkerAlt className='text-green-700' />
              {listing.address}
            </p>
            <div className='flex gap-4'>
              <p className='bg-red-900 w-full max-w-[200px] text-white text-center p-1
rounded-md'>
```

```
              {listing.type === 'rent' ? 'For Rent' : 'For Sale'}
            </p>
            {listing.offer && (
              <p className='bg-green-900 w-full max-w-[200px] text-white text-center p-1
rounded-md'>
                ${+listing.regularPrice - +listing.discountPrice} OFF
              </p>
            )}
          </div>
          <p className='text-slate-800'>
            <span className='font-semibold text-black'>Description - </span>
            {listing.description}
          </p>
          <ul className='text-green-900 font-semibold text-sm flex flex-wrap items-center
gap-4 sm:gap-6'>
            <li className='flex items-center gap-1 whitespace-nowrap '>
              <FaBed className='text-lg' />
              {listing.bedrooms > 1
                ? `${listing.bedrooms} beds `
                : `${listing.bedrooms} bed `}
            </li>
            <li className='flex items-center gap-1 whitespace-nowrap '>
              <FaBath className='text-lg' />
              {listing.bathrooms > 1
                ? `${listing.bathrooms} baths `
                : `${listing.bathrooms} bath `}
            </li>
            <li className='flex items-center gap-1 whitespace-nowrap '>
              <FaParking className='text-lg' />
              {listing.parking ? 'Parking spot' : 'No Parking'}
            </li>
            <li className='flex items-center gap-1 whitespace-nowrap '>
              <FaChair className='text-lg' />
              {listing.furnished ? 'Furnished' : 'Unfurnished'}
            </li>
          </ul>
          {currentUser && listing.userRef !== currentUser._id && !contact && (
            <button
              onClick={() => setContact(true)}
              className='bg-slate-700 text-white rounded-lg uppercase hover:opacity-95 p-3'
            >
              Contact landlord
            </button>
          )}
          {contact && <Contact listing={listing} />}
        </div>
      </div>
    )}
  </main>
);
```

```
}
```

Createlisting.jsx

```
import { useState } from 'react';
import {
  getDownloadURL,
  getStorage,
  ref,
  uploadBytesResumable,
} from 'firebase/storage';
import { app } from '../firebase';
import { useSelector } from 'react-redux';
import { useNavigate } from 'react-router-dom';

export default function CreateListing() {
  const { currentUser } = useSelector((state) => state.user);
  const navigate = useNavigate();
  const [files, setFiles] = useState([]);
  const [formData, setFormData] = useState({
    imageUrls: [],
    name: '',
    description: '',
    address: '',
    type: 'rent',
    bedrooms: 1,
    bathrooms: 1,
    regularPrice: 50,
    discountPrice: 0,
    offer: false,
    parking: false,
    furnished: false,
  });
  const [imageUploadError, setImageUploadError] = useState(false);
  const [uploading, setUploading] = useState(false);
  const [error, setError] = useState(false);
  const [loading, setLoading] = useState(false);
  console.log(formData);
  const handleImageSubmit = (e) => {
    if (files.length > 0 && files.length + formData.imageUrls.length < 7) {
      setUploading(true);
      setImageUploadError(false);
      const promises = [];

      for (let i = 0; i < files.length; i++) {
        promises.push(storeImage(files[i]));
      }
      Promise.all(promises)
        .then((urls) => {
```

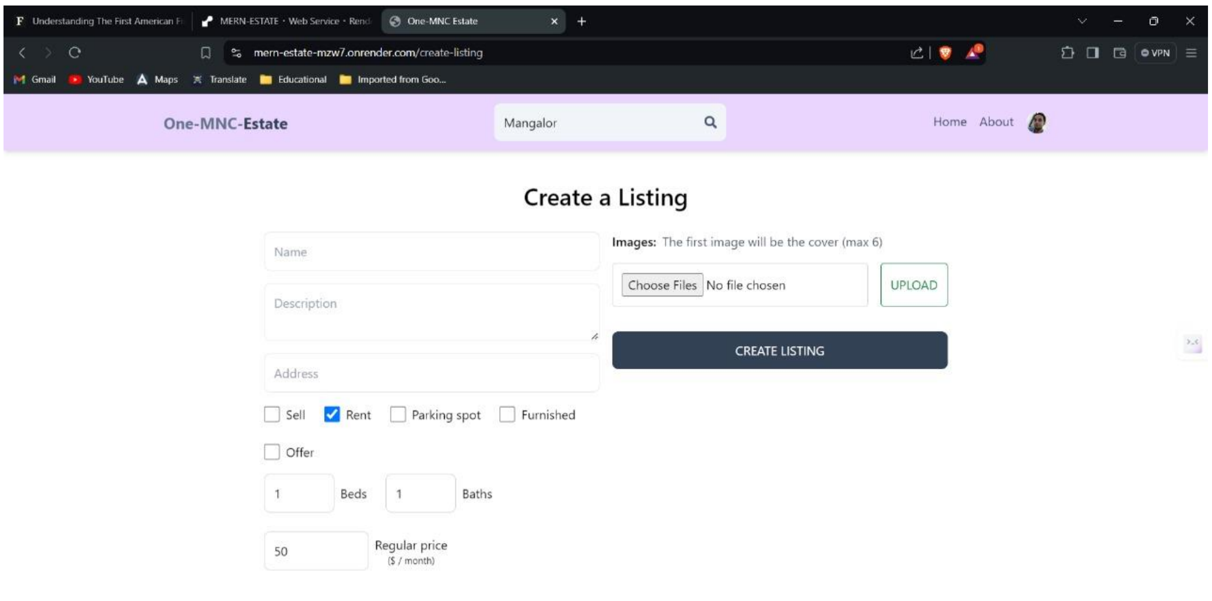
```
        setFormData({
          ...formData,
          imageUrls: formData.imageUrls.concat(urls),
        });
        setImageUploadError(false);
        setUploading(false);
      })
      .catch((err) => {
        setImageUploadError('Image upload failed (2 mb max per image)');
        setUploading(false);
      });
  } else {
    setImageUploadError('You can only upload 6 images per listing');
    setUploading(false);
  }
};

const storeImage = async (file) => {
  return new Promise((resolve, reject) => {
    const storage = getStorage(app);
    const fileName = new Date().getTime() + file.name;
    const storageRef = ref(storage, fileName);
    const uploadTask = uploadBytesResumable(storageRef, file);
    uploadTask.on(
      'state_changed',
      (snapshot) => {
        const progress =
          (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
        console.log(Upload is ${progress}% done);
      },
      (error) => {
        reject(error);
      },
      () => {
        getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
          resolve(downloadURL);
        });
      }
    );
  });
};

const handleRemoveImage = (index) => {
  setFormData({
    ...formData,
    imageUrls: formData.imageUrls.filter((_, i) => i !== index),
  });
};

const handleChange = (e) => {
  if (e.target.id === 'sale' || e.target.id === 'rent') {
    setFormData({
      ...formData,
      type: e.target.id,
    });
  }

  if (
    e.target.id === 'parking' ||
    e.target.id === 'furnished' ||
    e.target.id === 'offer'
  ) {
    setFormData({
      ...formData,
      [e.target.id]: e.target.checked,
    });
  }

  if (
    e.target.type === 'number' ||
    e.target.type === 'text' ||
    e.target.type === 'textarea'
  ) {
    setFormData({
      ...formData,
      [e.target.id]: e.target.value,
    });
  }
};

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    if (formData.imageUrls.length < 1)
      return setError('You must upload at least one image');
    if (+formData.regularPrice < +formData.discountPrice)
      return setError('Discount price must be lower than regular price');
    setLoading(true);
    setError(false);
    const res = await fetch('/api/listing/create', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        ...formData,
        userRef: currentUser._id,
      }),
    });
    const data = await res.json();
```

```
      setLoading(false);
      if (data.success === false) {
        setError(data.message);
      }
      navigate(/listing/${data._id});
    } catch (error) {
      setError(error.message);
      setLoading(false);
    }
  };
  return (
    <main className='p-3 max-w-4xl mx-auto'>
      <h1 className='text-3xl font-semibold text-center my-7'>
        Create a Listing
      </h1>
      <form onSubmit={handleSubmit} className='flex flex-col sm:flex-row gap-4'>
        <div className='flex flex-col gap-4 flex-1'>
          <input
            type='text'
            placeholder='Name'
            className='border p-3 rounded-lg'
            id='name'
            maxLength='62'
            minLength='10'
            required
            onChange={handleChange}
            value={formData.name}
          />
          <textarea
            type='text'
            placeholder='Description'
            className='border p-3 rounded-lg'
            id='description'
            required
            onChange={handleChange}
            value={formData.description}
          />
          <input
            type='text'
            placeholder='Address'
            className='border p-3 rounded-lg'
            id='address'
            required
            onChange={handleChange}
            value={formData.address}
          />
          <div className='flex gap-6 flex-wrap'>
            <div className='flex gap-2'>
              <input
                type='checkbox'
      id='sale'
      className='w-5'
      onChange={handleChange}
      checked={formData.type === 'sale'}
    />
    <span>Sell</span>
  </div>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='rent'
      className='w-5'
      onChange={handleChange}
      checked={formData.type === 'rent'}
    />
    <span>Rent</span>
  </div>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='parking'
      className='w-5'
      onChange={handleChange}
      checked={formData.parking}
    />
    <span>Parking spot</span>
  </div>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='furnished'
      className='w-5'
      onChange={handleChange}
      checked={formData.furnished}
    />
    <span>Furnished</span>
  </div>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='offer'
      className='w-5'
      onChange={handleChange}
      checked={formData.offer}
    />
    <span>Offer</span>
  </div>
</div>
<div className='flex flex-wrap gap-6'>
  <div className='flex items-center gap-2'>
  <input
    type='number'
    id='bedrooms'
    min='1'
    max='10'
    required
    className='p-3 border border-gray-300 rounded-lg'
    onChange={handleChange}
    value={formData.bedrooms}
  />
  <p>Beds</p>
</div>
<div className='flex items-center gap-2'>
  <input
    type='number'
    id='bathrooms'
    min='1'
    max='10'
    required
    className='p-3 border border-gray-300 rounded-lg'
    onChange={handleChange}
    value={formData.bathrooms}
  />
  <p>Baths</p>
</div>
<div className='flex items-center gap-2'>
  <input
    type='number'
    id='regularPrice'
    min='50'
    max='10000000'
    required
    className='p-3 border border-gray-300 rounded-lg'
    onChange={handleChange}
    value={formData.regularPrice}
  />
  <div className='flex flex-col items-center'>
    <p>Regular price</p>
    {formData.type === 'rent' && (
      <span className='text-xs'>($ / month)</span>
    )}
  </div>
</div>
{formData.offer && (
  <div className='flex items-center gap-2'>
    <input
      type='number'
      id='discountPrice'
      min='0'
      max='10000000'
```

```
              required
              className='p-3 border border-gray-300 rounded-lg'
              onChange={handleChange}
              value={formData.discountPrice}
            />
            <div className='flex flex-col items-center'>
              <p>Discounted price</p>

              {formData.type === 'rent' && (
                <span className='text-xs'>($ / month)</span>
              )}
            </div>
          </div>
        )}
      </div>
    </div>
    <div className='flex flex-col flex-1 gap-4'>
      <p className='font-semibold'>
        Images:
        <span className='font-normal text-gray-600 ml-2'>
          The first image will be the cover (max 6)
        </span>
      </p>
      <div className='flex gap-4'>
        <input
          onChange={(e) => setFiles(e.target.files)}
          className='p-3 border border-gray-300 rounded w-full'
          type='file'
          id='images'
          accept='image/*'
          multiple
        />
        <button
          type='button'
          disabled={uploading}
          onClick={handleImageSubmit}
          className='p-3 text-green-700 border border-green-700 rounded uppercase
hover:shadow-lg disabled:opacity-80'
        >
          {uploading ? 'Uploading...' : 'Upload'}
        </button>
      </div>
      <p className='text-red-700 text-sm'>
        {imageUploadError && imageUploadError}
      </p>
      {formData.imageUrls.length > 0 &&
        formData.imageUrls.map((url, index) => (
          <div
            key={url}
            className='flex justify-between p-3 border items-center'
            >
              <img
                src={url}
                alt='listing image'
                className='w-20 h-20 object-contain rounded-lg'
              />
              <button
                type='button'
                onClick={() => handleRemoveImage(index)}
                className='p-3 text-red-700 rounded-lg uppercase hover:opacity-75'
              >
                Delete
              </button>
            </div>
          ))}
        <button
          disabled={loading || uploading}
          className='p-3 bg-slate-700 text-white rounded-lg uppercase hover:opacity-95
disabled:opacity-80'
        >
          {loading ? 'Creating...' : 'Create listing'}
        </button>
        {error && <p className='text-red-700 text-sm'>{error}</p>}
      </div>
    </form>
  </main>
 );
}
```

UpdateListing.jsx

```
import { useEffect, useState } from 'react';
import {
  getDownloadURL,
  getStorage,
  ref,
  uploadBytesResumable,
} from 'firebase/storage';
import { app } from '../firebase';
import { useSelector } from 'react-redux';
import { useNavigate, useParams } from 'react-router-dom';

export default function CreateListing() {
  const { currentUser } = useSelector((state) => state.user);
  const navigate = useNavigate();
  const params = useParams();
  const [files, setFiles] = useState([]);
  const [formData, setFormData] = useState({
    imageUrls: [],
    name: '',
    description: '',
    address: '',
    type: 'rent',
    bedrooms: 1,
    bathrooms: 1,
    regularPrice: 50,
    discountPrice: 0,
    offer: false,
    parking: false,
    furnished: false,
  });
  const [imageUploadError, setImageUploadError] = useState(false);
  const [uploading, setUploading] = useState(false);
  const [error, setError] = useState(false);
  const [loading, setLoading] = useState(false);

  useEffect(() => {
    const fetchListing = async () => {
      const listingId = params.listingId;
      const res = await fetch(/api/listing/get/${listingId});
      const data = await res.json();
      if (data.success === false) {
        console.log(data.message);
        return;
      }
      setFormData(data);
    };

    fetchListing();
  }, []);

  const handleImageSubmit = (e) => {
    if (files.length > 0 && files.length + formData.imageUrls.length < 7) {
      setUploading(true);
      setImageUploadError(false);
      const promises = [];

      for (let i = 0; i < files.length; i++) {
        promises.push(storeImage(files[i]));
      }
      Promise.all(promises)
        .then((urls) => {
          setFormData({
            ...formData,
            imageUrls: formData.imageUrls.concat(urls),
          });
          setImageUploadError(false);
          setUploading(false);
        })
        .catch((err) => {
          setImageUploadError('Image upload failed (2 mb max per image)');
          setUploading(false);
        });
    } else {
      setImageUploadError('You can only upload 6 images per listing');
      setUploading(false);
    }
  };

  const storeImage = async (file) => {
    return new Promise((resolve, reject) => {
      const storage = getStorage(app);
      const fileName = new Date().getTime() + file.name;
      const storageRef = ref(storage, fileName);
      const uploadTask = uploadBytesResumable(storageRef, file);
      uploadTask.on(
        'state_changed',
        (snapshot) => {
          const progress =
            (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
          console.log(Upload is ${progress}% done);
        },
        (error) => {
          reject(error);
        },
        () => {
          getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
            resolve(downloadURL);
          });
        }
      );
    });
  };

  const handleRemoveImage = (index) => {
    setFormData({
      ...formData,
      imageUrls: formData.imageUrls.filter((_, i) => i !== index),
    });
  };

  const handleChange = (e) => {
    if (e.target.id === 'sale' || e.target.id === 'rent') {
      setFormData({
        ...formData,
        type: e.target.id,
      });
    }

    if (
      e.target.id === 'parking' ||
      e.target.id === 'furnished' ||
      e.target.id === 'offer'
    ) {
      setFormData({
        ...formData,
        [e.target.id]: e.target.checked,
      });
    }

    if (
      e.target.type === 'number' ||
      e.target.type === 'text' ||
      e.target.type === 'textarea'
    ) {
      setFormData({
        ...formData,
        [e.target.id]: e.target.value,
      });
    }
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      if (formData.imageUrls.length < 1)
        return setError('You must upload at least one image');
      if (+formData.regularPrice < +formData.discountPrice)
        return setError('Discount price must be lower than regular price');
      setLoading(true);
      setError(false);
      const res = await fetch(/api/listing/update/${params.listingId}, {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          ...formData,
          userRef: currentUser._id,
        }),
      });
      const data = await res.json();
      setLoading(false);
      if (data.success === false) {
        setError(data.message);
      }
      navigate(/listing/${data._id});
    } catch (error) {
      setError(error.message);
      setLoading(false);
    }
  };
  return (
    <main className='p-3 max-w-4xl mx-auto'>
      <h1 className='text-3xl font-semibold text-center my-7'>
        Update a Listing
      </h1>
      <form onSubmit={handleSubmit} className='flex flex-col sm:flex-row gap-4'>
        <div className='flex flex-col gap-4 flex-1'>
          <input
            type='text'
            placeholder='Name'
            className='border p-3 rounded-lg'
            id='name'
            maxLength='62'
            minLength='10'
            required
            onChange={handleChange}
            value={formData.name}
          />
          <textarea
            type='text'
            placeholder='Description'
            className='border p-3 rounded-lg'
            id='description'
            required
            onChange={handleChange}
            value={formData.description}
          />
          <input
            type='text'
            placeholder='Address'
            className='border p-3 rounded-lg'
            id='address'
            required
            onChange={handleChange}
            value={formData.address}
          />
          <div className='flex gap-6 flex-wrap'>
            <div className='flex gap-2'>
              <input
                type='checkbox'
                id='sale'
                className='w-5'
                onChange={handleChange}
                checked={formData.type === 'sale'}
              />
              <span>Sell</span>
            </div>
            <div className='flex gap-2'>
      <input
        type='checkbox'
        id='rent'
        className='w-5'
        onChange={handleChange}
        checked={formData.type === 'rent'}
      />
      <span>Rent</span>
    </div>
    <div className='flex gap-2'>
      <input
        type='checkbox'
        id='parking'
        className='w-5'
        onChange={handleChange}
        checked={formData.parking}
      />
      <span>Parking spot</span>
    </div>
    <div className='flex gap-2'>
      <input
        type='checkbox'
        id='furnished'
        className='w-5'
        onChange={handleChange}
        checked={formData.furnished}
      />
      <span>Furnished</span>
    </div>
    <div className='flex gap-2'>
      <input
        type='checkbox'
        id='offer'
        className='w-5'
        onChange={handleChange}
        checked={formData.offer}
      />
      <span>Offer</span>
    </div>
  </div>
  <div className='flex flex-wrap gap-6'>
    <div className='flex items-center gap-2'>
      <input
        type='number'
        id='bedrooms'
        min='1'
        max='10'
        required
        className='p-3 border border-gray-300 rounded-lg'
        onChange={handleChange}
      value={formData.bedrooms}
    />
    <p>Beds</p>
  </div>
  <div className='flex items-center gap-2'>
    <input
      type='number'
      id='bathrooms'
      min='1'
      max='10'
      required
      className='p-3 border border-gray-300 rounded-lg'
      onChange={handleChange}
      value={formData.bathrooms}
    />
    <p>Baths</p>
  </div>
  <div className='flex items-center gap-2'>
    <input
      type='number'
      id='regularPrice'
      min='50'
      max='10000000'
      required
      className='p-3 border border-gray-300 rounded-lg'
      onChange={handleChange}
      value={formData.regularPrice}
    />
    <div className='flex flex-col items-center'>
      <p>Regular price</p>
      {formData.type === 'rent' && (
        <span className='text-xs'>($ / month)</span>
      )}
    </div>
  </div>
  {formData.offer && (
    <div className='flex items-center gap-2'>
      <input
        type='number'
        id='discountPrice'
        min='0'
        max='10000000'
        required
        className='p-3 border border-gray-300 rounded-lg'
        onChange={handleChange}
        value={formData.discountPrice}
      />
      <div className='flex flex-col items-center'>
        <p>Discounted price</p>
        {formData.type === 'rent' && (
                <span className='text-xs'>($ / month)</span>
              )}
            </div>
          </div>
        ))}
      </div>
    </div>
    <div className='flex flex-col flex-1 gap-4'>
      <p className='font-semibold'>
        Images:
        <span className='font-normal text-gray-600 ml-2'>
          The first image will be the cover (max 6)
        </span>
      </p>
      <div className='flex gap-4'>
        <input
          onChange={(e) => setFiles(e.target.files)}
          className='p-3 border border-gray-300 rounded w-full'
          type='file'
          id='images'
          accept='image/*'
          multiple
        />
        <button
          type='button'
          disabled={uploading}
          onClick={handleImageSubmit}
          className='p-3 text-green-700 border border-green-700 rounded uppercase
hover:shadow-lg disabled:opacity-80'
        >
          {uploading ? 'Uploading...' : 'Upload'}
        </button>
      </div>
      <p className='text-red-700 text-sm'>
        {imageUploadError && imageUploadError}
      </p>
      {formData.imageUrls.length > 0 &&
        formData.imageUrls.map((url, index) => (
          <div
            key={url}
            className='flex justify-between p-3 border items-center'
          >
            <img
              src={url}
              alt='listing image'
              className='w-20 h-20 object-contain rounded-lg'
            />
            <button
              type='button'
              onClick={() => handleRemoveImage(index)}
              className='p-3 text-red-700 rounded-lg uppercase hover:opacity-75'
            >
              Delete
            </button>
          </div>
        ))}
      <button
        disabled={loading || uploading}
        className='p-3 bg-slate-700 text-white rounded-lg uppercase hover:opacity-95
disabled:opacity-80'
      >
        {loading ? 'Updating...' : 'Update listing'}
      </button>
      {error && <p className='text-red-700 text-sm'>{error}</p>}
    </div>
  </form>
</main>
);
}
```



Mangalor

Create a Listing

Name

Description

Address

Images: The first image will be the cover (max 6)

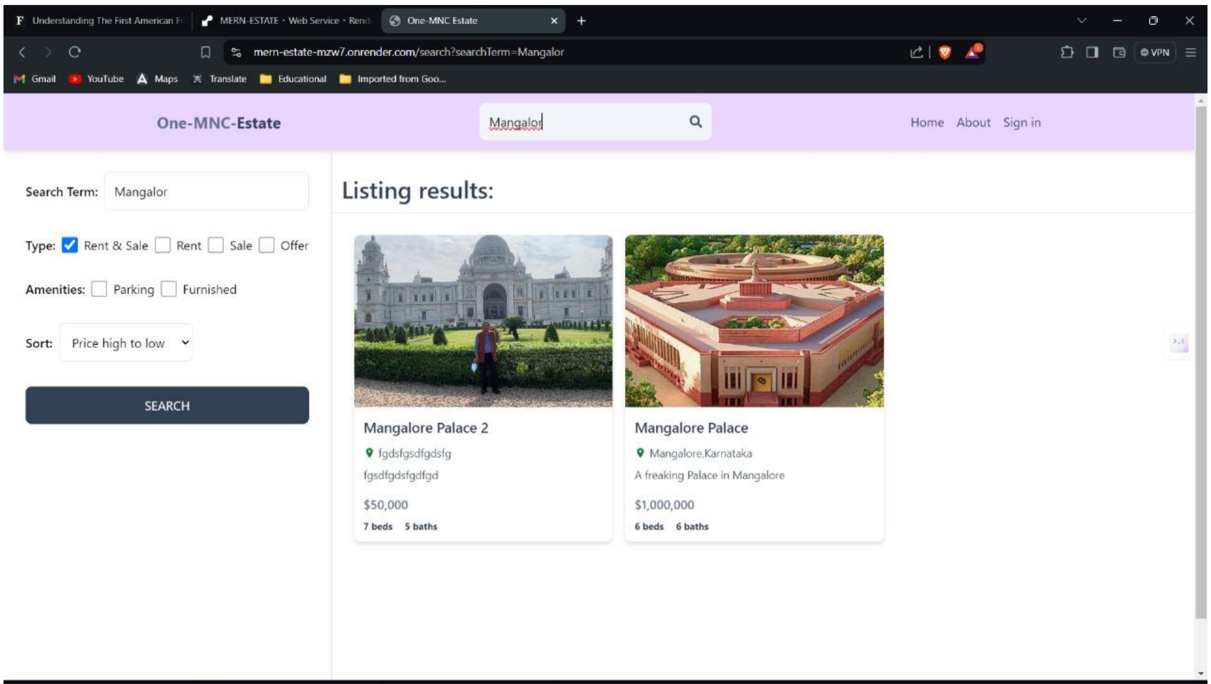
Choose Files No file chosen

[ ] Sell [✓ Rent] [ ] Parking spot [ ] Furnished

[ ] Offer

1 Beds 1 Baths

50 Regular price ($ / month)



Understanding The First American F | MERN-ESTATE · Web Service · Rend | One-MNC Estate

mern-estate-mzw7.onrender.com/search?searchTerm=Mangalor

Gmail YouTube Maps Translate Educational Imported from Goo...

One-MNC-Estate | Mangalor | Home About Sign in

Search Term: Mangalor

Type: [✓ Rent & Sale] Rent Sale Offer

Amenities: Parking Furnished

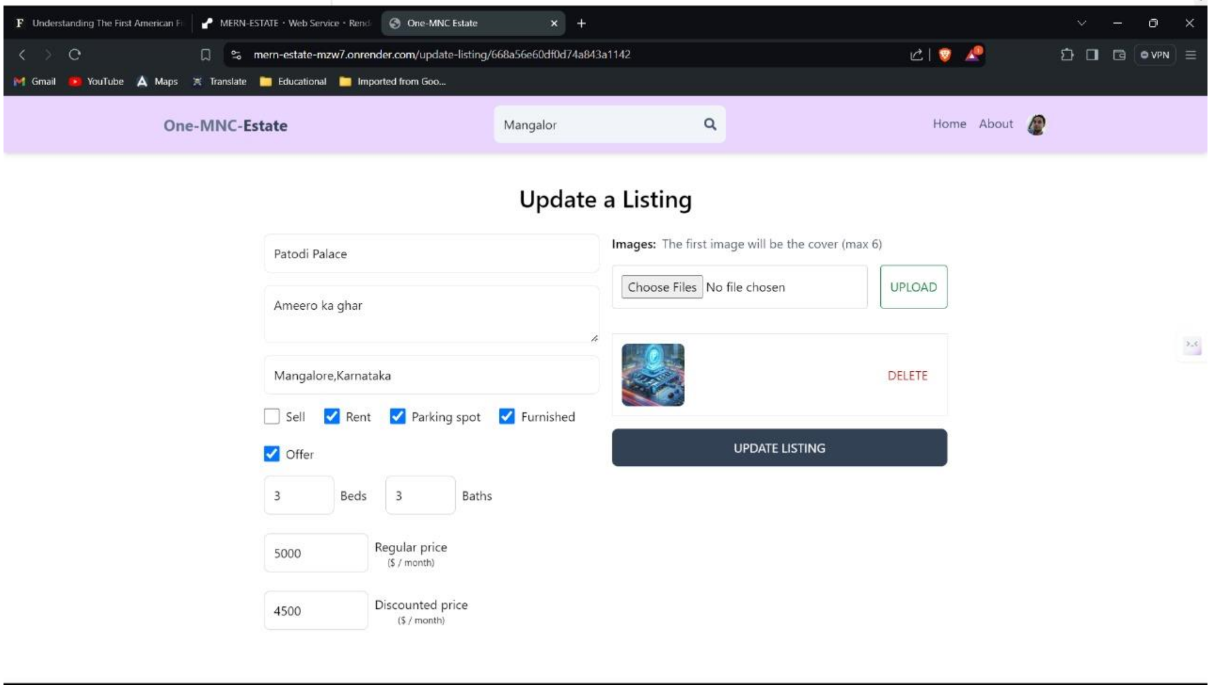
Sort: Price high to low

SEARCH

Listing results:

Mangalore Palace 2
fgdsfgsdfgdsfg
fgsdfgdsfgdfgd
$50,000
7 beds 5 baths

Mangalore Palace
Mangalore,Karnataka
A freaking Palace in Mangalore
$1,000,000
6 beds 6 baths



Understanding The First American F | MERN-ESTATE · Web Service · Rend | One-MNC Estate

mern-estate-mzw7.onrender.com/update-listing/668a56e60df0d74a843a1142

Gmail YouTube Maps Translate Educational Imported from Goo...

One-MNC-Estate | Mangalor | Home About

Update a Listing

Patodi Palace

Ameero ka ghar

Mangalore,Karnataka

Sell [✓ Rent] [✓ Parking spot] [✓ Furnished]

[✓ Offer]

3 Beds 3 Baths

5000 Regular price ($ / month)

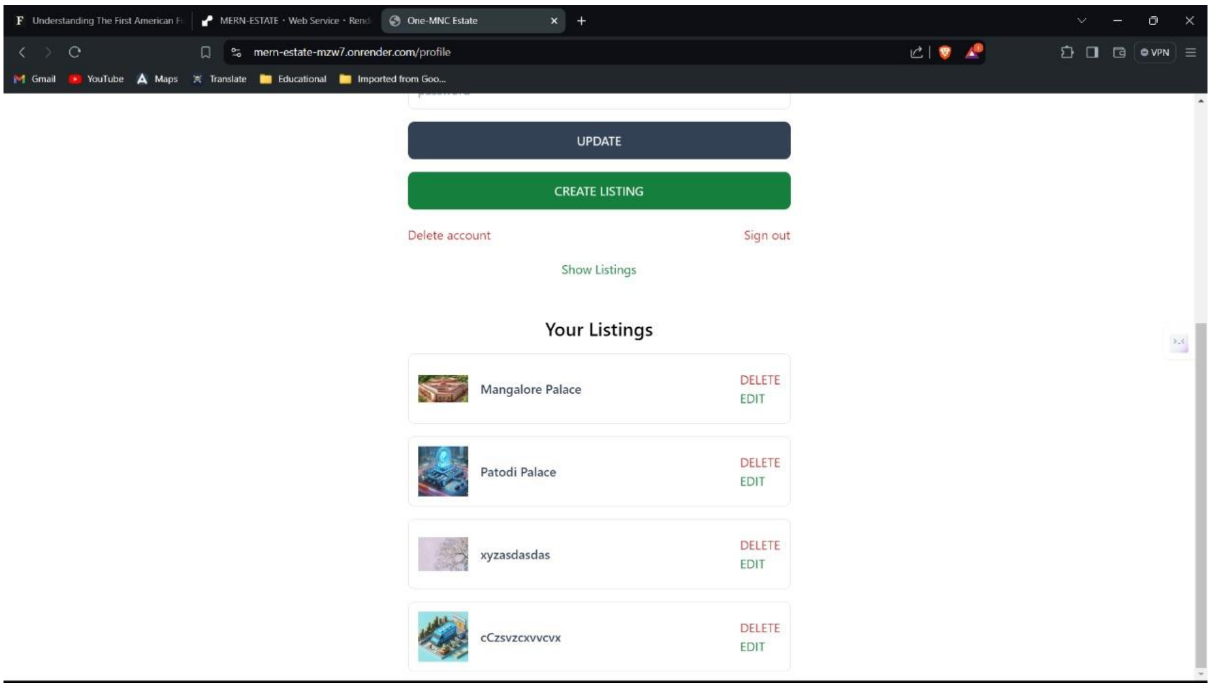
4500 Discounted price ($ / month)

Images: The first image will be the cover (max 6)

Choose Files No file chosen UPLOAD

DELETE

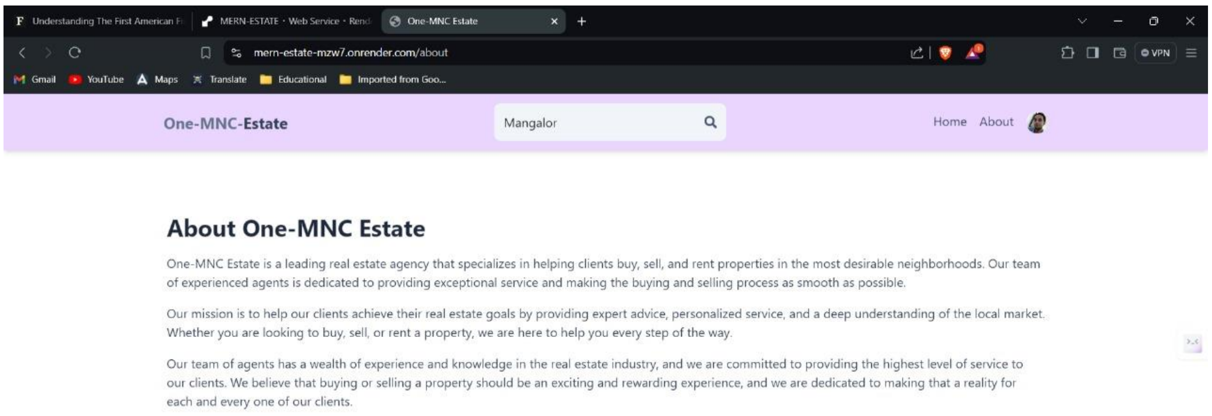
UPDATE LISTING



About Page:

- Information about One-MNC Estate, including its mission, team, and the services offered.
- Provides insights into the company's commitment to helping clients achieve their real estate goals.

```
import React from 'react'

export default function About() {
  return (
    <div className='py-20 px-4 max-w-6xl mx-auto'>
      <h1 className='text-3xl font-bold mb-4 text-slate-800'>About One-MNC Estate</h1>
      <p className='mb-4 text-slate-700'>One-MNC Estate is a leading real estate agency that specializes in helping clients buy, sell, and rent properties in the most desirable neighborhoods. Our team of experienced agents is dedicated to providing exceptional service and making the buying and selling process as smooth as possible.</p>
      <p className='mb-4 text-slate-700'>
      Our mission is to help our clients achieve their real estate goals by providing expert advice, personalized service, and a deep understanding of the local market. Whether you are looking to buy, sell, or rent a property, we are here to help you every step of the way.
      </p>
      <p className='mb-4 text-slate-700'>Our team of agents has a wealth of experience and knowledge in the real estate industry, and we are committed to providing the highest level of service to our clients. We believe that buying or selling a property should be an exciting and
rewarding experience, and we are dedicated to making that a reality for each and every one of our clients.</p>
    </div>
  )
}
```



Understanding The First American F | MERN-ESTATE · Web Service · Rend | One-MNC Estate

mern-estate-mzw7.onrender.com/about

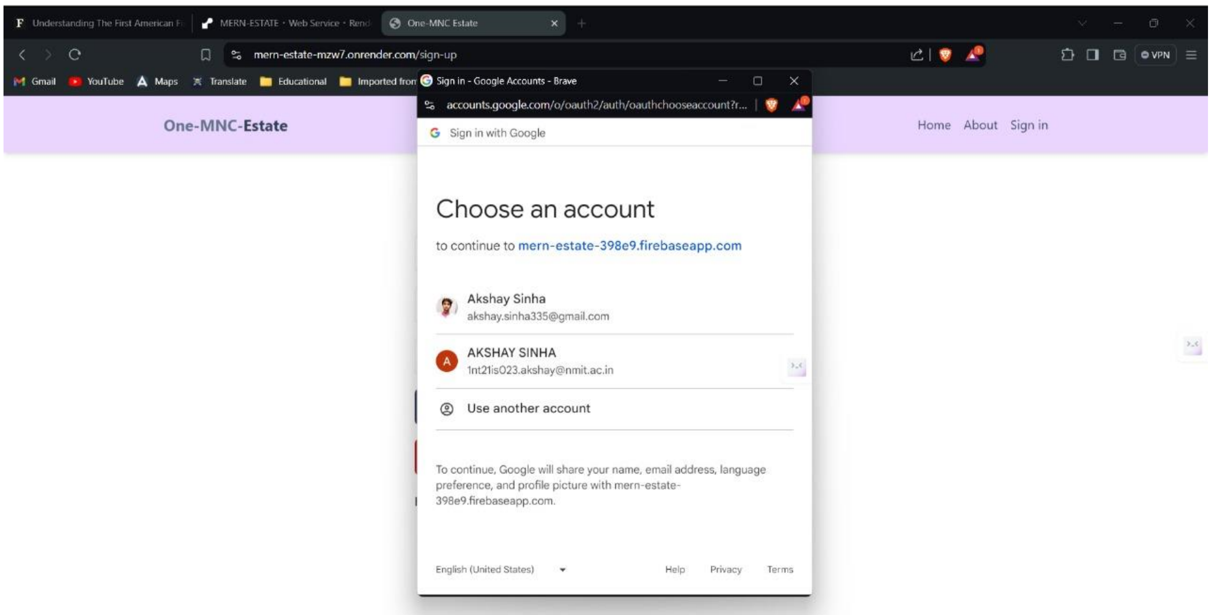
One-MNC-Estate | Mangalor | Home | About

About One-MNC Estate

One-MNC Estate is a leading real estate agency that specializes in helping clients buy, sell, and rent properties in the most desirable neighborhoods. Our team of experienced agents is dedicated to providing exceptional service and making the buying and selling process as smooth as possible.

Our mission is to help our clients achieve their real estate goals by providing expert advice, personalized service, and a deep understanding of the local market. Whether you are looking to buy, sell, or rent a property, we are here to help you every step of the way.

Our team of agents has a wealth of experience and knowledge in the real estate industry, and we are committed to providing the highest level of service to our clients. We believe that buying or selling a property should be an exciting and rewarding experience, and we are dedicated to making that a reality for each and every one of our clients.



mern-estate-mzw7.onrender.com/sign-up

One-MNC-Estate | Home | About | Sign in

Sign in - Google Accounts - Brave

accounts.google.com/o/oauth2/auth/oauthchooseaccount?r...

Sign in with Google

Choose an account

to continue to mern-estate-398e9.firebaseapp.com

Akshay Sinha
akshay.sinha335@gmail.com

AKSHAY SINHA
1nt21is023.akshay@nmit.ac.in

Use another account

To continue, Google will share your name, email address, language preference, and profile picture with mern-estate-398e9.firebaseapp.com.

English (United States) | Help | Privacy | Terms

Rent Page

The Rent Page of One-MNC Estate is designed to provide users with an efficient and intuitive interface to browse and find rental properties that suit their needs. Here’s a detailed look at the features and functionality of the Rent Page:

Key Features:

1. **Search Functionality:**
   - **Search Bar:** Users can enter specific locations or property names to quickly find relevant rental listings.
   - **Filters:** Advanced filtering options allow users to narrow down results based on criteria such as price range, number of bedrooms, number of bathrooms, availability of parking spots, and whether the property is furnished.
2. **Listings Display:**
   - **Grid/List View:** Users can switch between grid and list views to browse property listings in their preferred format.
   - **Property Cards:** Each property is displayed as a card featuring key information such as the property name, location, price, and a thumbnail image. Clicking on a card provides more detailed information.
3. **Detailed Property View:**
   - **Property Details:** Users can click on a listing to view comprehensive details, including a full description, amenities, high-quality images, and contact information for the landlord or property manager.
   - **Interactive Map:** An integrated map view shows the exact location of the property, along with nearby points of interest such as schools, hospitals, and public transport.
4. **User Interaction:**
   - **Save to Favorites:** Users can save listings to their favorites for easy access later.
   - **Contact Landlord:** Direct messaging or contact form to inquire about the property, schedule viewings, or ask questions.
   - **Reviews and Ratings:** Users can read reviews and ratings from previous tenants to get an idea of the property and landlord's reliability.
5. **Responsive Design:**
   - **Mobile-Friendly:** The page is fully responsive, ensuring a seamless experience on both desktop and mobile devices.

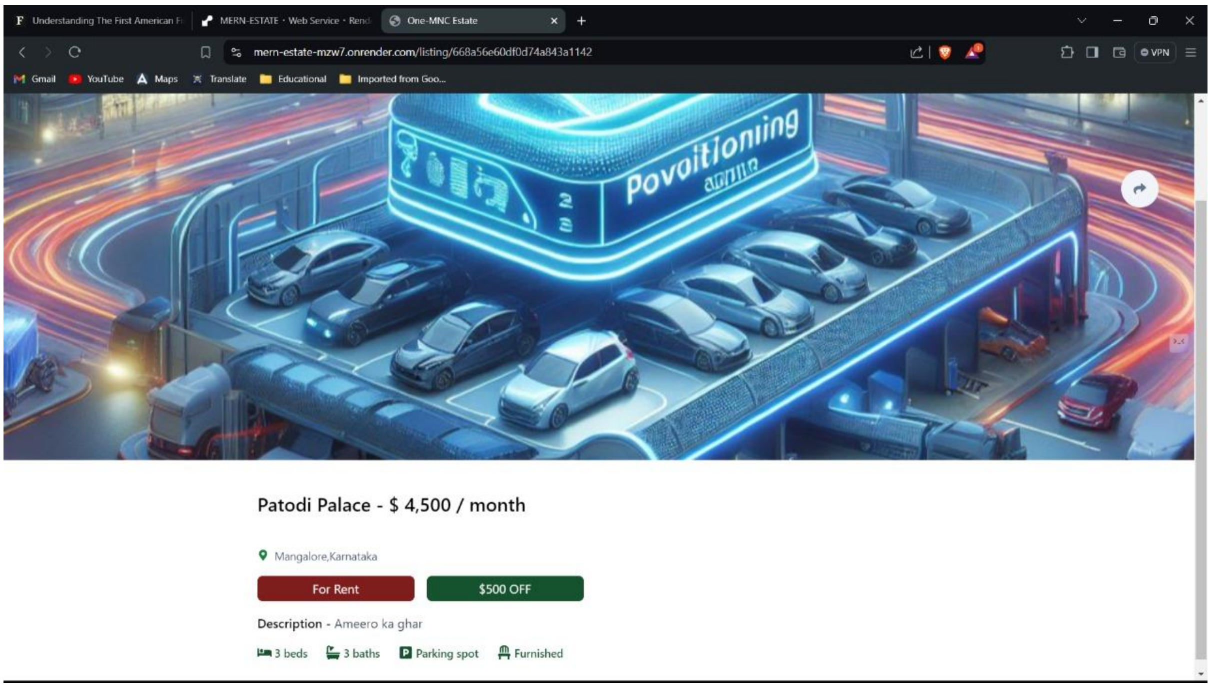
Search.jsx

```
import { useEffect, useState } from 'react';
import { useNavigate } from 'react-router-dom';
import ListingItem from '../components/ListingItem';

export default function Search() {
  const navigate = useNavigate();
  const [sidebardata, setSidebardata] = useState({
    searchTerm: '',
    type: 'all',
    parking: false,
    furnished: false,
    offer: false,
    sort: 'created_at',
    order: 'desc',
  });

  const [loading, setLoading] = useState(false);
  const [listings, setListings] = useState([]);
  const [showMore, setShowMore] = useState(false);

  useEffect(() => {
    const urlParams = new URLSearchParams(location.search);
    const searchTermFromUrl = urlParams.get('searchTerm');
    const typeFromUrl = urlParams.get('type');
    const parkingFromUrl = urlParams.get('parking');
    const furnishedFromUrl = urlParams.get('furnished');
    const offerFromUrl = urlParams.get('offer');
    const sortFromUrl = urlParams.get('sort');
    const orderFromUrl = urlParams.get('order');

    if (
      searchTermFromUrl ||
      typeFromUrl ||
      parkingFromUrl ||
      furnishedFromUrl ||
      offerFromUrl ||
      sortFromUrl ||
      orderFromUrl
    ) {
      setSidebardata({
        searchTerm: searchTermFromUrl || '',
        type: typeFromUrl || 'all',
        parking: parkingFromUrl === 'true' ? true : false,
        furnished: furnishedFromUrl === 'true' ? true : false,
        offer: offerFromUrl === 'true' ? true : false,
      sort: sortFromUrl || 'created_at',
      order: orderFromUrl || 'desc',
    });
  }

  const fetchListings = async () => {
    setLoading(true);
    setShowMore(false);
    const searchQuery = urlParams.toString();
    const res = await fetch(/api/listing/get?${searchQuery});
    const data = await res.json();
    if (data.length > 8) {
      setShowMore(true);
    } else {
      setShowMore(false);
    }
    setListings(data);
    setLoading(false);
  };

  fetchListings();
}, [location.search]);

const handleChange = (e) => {
  if (
    e.target.id === 'all' ||
    e.target.id === 'rent' ||
    e.target.id === 'sale'
  ) {
    setSidebardata({ ...sidebardata, type: e.target.id });
  }

  if (e.target.id === 'searchTerm') {
    setSidebardata({ ...sidebardata, searchTerm: e.target.value });
  }

  if (
    e.target.id === 'parking' ||
    e.target.id === 'furnished' ||
    e.target.id === 'offer'
  ) {
    setSidebardata({
      ...sidebardata,
      [e.target.id]:
        e.target.checked || e.target.checked === 'true' ? true : false,
    });
  }

  if (e.target.id === 'sort_order') {
    const sort = e.target.value.split('_')[0] || 'created_at';
    const order = e.target.value.split('_')[1] || 'desc';

    setSidebardata({ ...sidebardata, sort, order });
  }
};

const handleSubmit = (e) => {
  e.preventDefault();
  const urlParams = new URLSearchParams();
  urlParams.set('searchTerm', sidebardata.searchTerm);
  urlParams.set('type', sidebardata.type);
  urlParams.set('parking', sidebardata.parking);
  urlParams.set('furnished', sidebardata.furnished);
  urlParams.set('offer', sidebardata.offer);
  urlParams.set('sort', sidebardata.sort);
  urlParams.set('order', sidebardata.order);
  const searchQuery = urlParams.toString();
  navigate(/search?${searchQuery});
};

const onShowMoreClick = async () => {
  const numberOfListings = listings.length;
  const startIndex = numberOfListings;
  const urlParams = new URLSearchParams(location.search);
  urlParams.set('startIndex', startIndex);
  const searchQuery = urlParams.toString();
  const res = await fetch(/api/listing/get?${searchQuery});
  const data = await res.json();
  if (data.length < 9) {
    setShowMore(false);
  }
  setListings([...listings, ...data]);
};
return (
  <div className='flex flex-col md:flex-row'>
    <div className='p-7  border-b-2 md:border-r-2 md:min-h-screen'>
      <form onSubmit={handleSubmit} className='flex flex-col gap-8'>
        <div className='flex items-center gap-2'>
          <label className='whitespace-nowrap font-semibold'>
            Search Term:
          </label>
          <input
            type='text'
            id='searchTerm'
            placeholder='Search...'
            className='border rounded-lg p-3 w-full'
            value={sidebardata.searchTerm}
            onChange={handleChange}
          />
```

```
</div>
<div className='flex gap-2 flex-wrap items-center'>
  <label className='font-semibold'>Type:</label>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='all'
      className='w-5'
      onChange={handleChange}
      checked={sidebardata.type === 'all'}
    />
    <span>Rent & Sale</span>
  </div>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='rent'
      className='w-5'
      onChange={handleChange}
      checked={sidebardata.type === 'rent'}
    />
    <span>Rent</span>
  </div>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='sale'
      className='w-5'
      onChange={handleChange}
      checked={sidebardata.type === 'sale'}
    />
    <span>Sale</span>
  </div>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='offer'
      className='w-5'
      onChange={handleChange}
      checked={sidebardata.offer}
    />
    <span>Offer</span>
  </div>
</div>
<div className='flex gap-2 flex-wrap items-center'>
  <label className='font-semibold'>Amenities:</label>
  <div className='flex gap-2'>
    <input
      type='checkbox'
      id='parking'
```

```
                className='w-5'
                onChange={handleChange}
                checked={sidebardata.parking}
              />
              <span>Parking</span>
            </div>
            <div className='flex gap-2'>
              <input
                type='checkbox'
                id='furnished'
                className='w-5'
                onChange={handleChange}
                checked={sidebardata.furnished}
              />
              <span>Furnished</span>
            </div>
          </div>
          <div className='flex items-center gap-2'>
            <label className='font-semibold'>Sort:</label>
            <select
              onChange={handleChange}
              defaultValue={'created_at_desc'}
              id='sort_order'
              className='border rounded-lg p-3'
            >
              <option value='regularPrice_desc'>Price high to low</option>
              <option value='regularPrice_asc'>Price low to hight</option>
              <option value='createdAt_desc'>Latest</option>
              <option value='createdAt_asc'>Oldest</option>
            </select>
          </div>
          <button className='bg-slate-700 text-white p-3 rounded-lg uppercase
hover:opacity-95'>
            Search
          </button>
        </form>
      </div>
      <div className='flex-1'>
        <h1 className='text-3xl font-semibold border-b p-3 text-slate-700 mt-5'>
          Listing results:
        </h1>
        <div className='p-7 flex flex-wrap gap-4'>
          {!loading && listings.length === 0 && (
            <p className='text-xl text-slate-700'>No listing found!</p>
          )}
          {loading && (
            <p className='text-xl text-slate-700 text-center w-full'>
              Loading...
            </p>
          )}
          {!loading &&
            listings &&
            listings.map((listing) => (
              <ListingItem key={listing._id} listing={listing} />
            ))}

          {showMore && (
            <button
              onClick={onShowMoreClick}
              className='text-green-700 hover:underline p-7 text-center w-full'
            >
              Show more
            </button>
          )}
        </div>
      </div>
    </div>
  );
}
```

Snapshot of Rentpage:


Understanding The First American F
MERN-ESTATE · Web Service · Rend
One-MNC Estate
mern-estate-mzw7.onrender.com/listing/668a56e60df0d74a843a1142
Gmail
YouTube
Maps
Translate
Educational
Imported from Goo...
VPN

Patodi Palace - $ 4,500 / month

Mangalore,Karnataka

For Rent

$500 OFF

Description - Ameero ka ghar

3 beds 3 baths Parking spot Furnished

Conclusion

One-MNC Estate is a sophisticated real estate commerce platform that exemplifies the strengths of the MERN stack. This application offers a user-friendly and secure environment for buying, renting, and selling properties, effectively preventing third-party interference in transactions. With a robust backend powered by Node.js and Express.js, a scalable and flexible database using MongoDB, and a dynamic, responsive frontend built with React, One-MNC Estate delivers an exceptional user experience. The seamless integration of these technologies ensures efficient performance, secure data management, and a cohesive development process. As a result, users can confidently manage their real estate needs, while developers benefit from the streamlined, full-stack JavaScript development environment. One-MNC Estate not only meets current market demands but is also well-positioned for future growth and enhancements, making it a standout solution in the realm of real estate commerce.