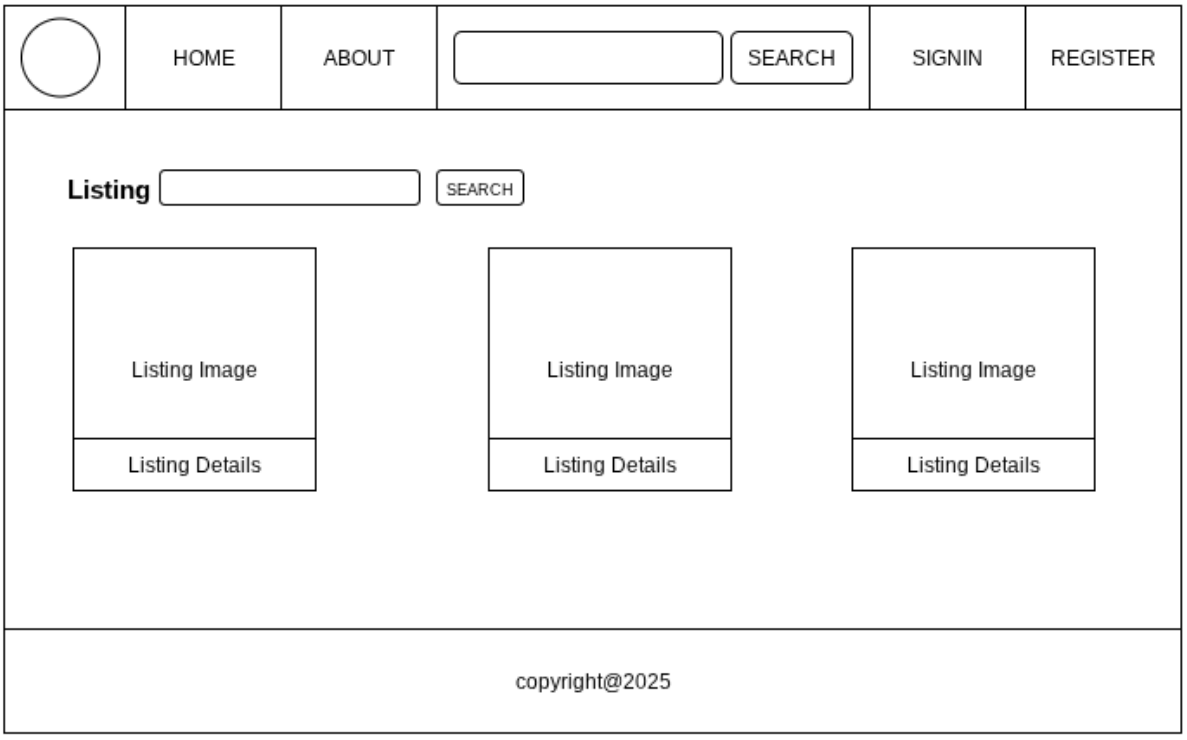
Building Frontend for a Home Listings Admin App using ReactJS

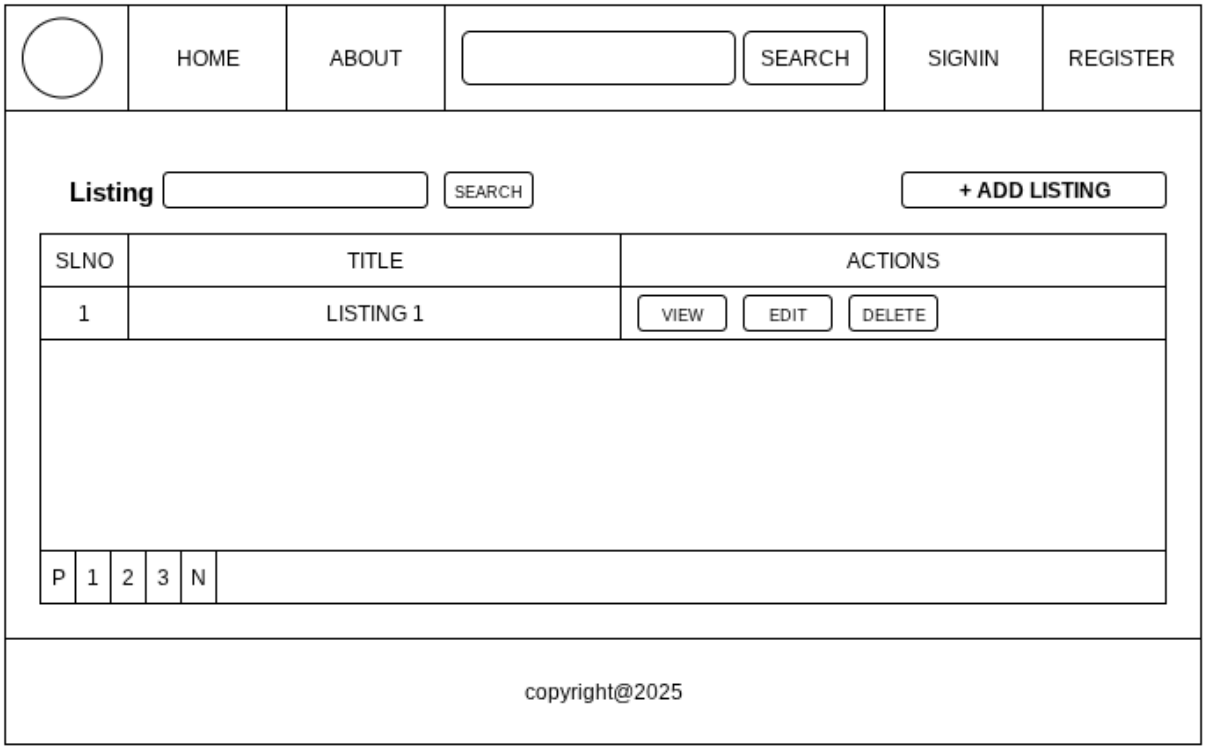
Before building any frontend app from scratch, we need to do a preparatory step of designing the frontend. While we have many design steps that we follow, we will see a brief of those here:

- Starting with an idea: `Home Listings Admin App` For a Home Listings app, we can have a seperate admin interface & than for a normal Customer. Here we will focus on building an Admin Interface.
- Design the Low Level Design: Taking inspiration from other works & deciding on the design, we iterate many time to reach at the desired UI. We decide on how the elements are placed & tentative structure of each element.
  This can be done on pen & paper or any software online. We will consider for our app, as of now 3 pages.
- Home Page
- Dashboard
- Listing Edit

HomePage

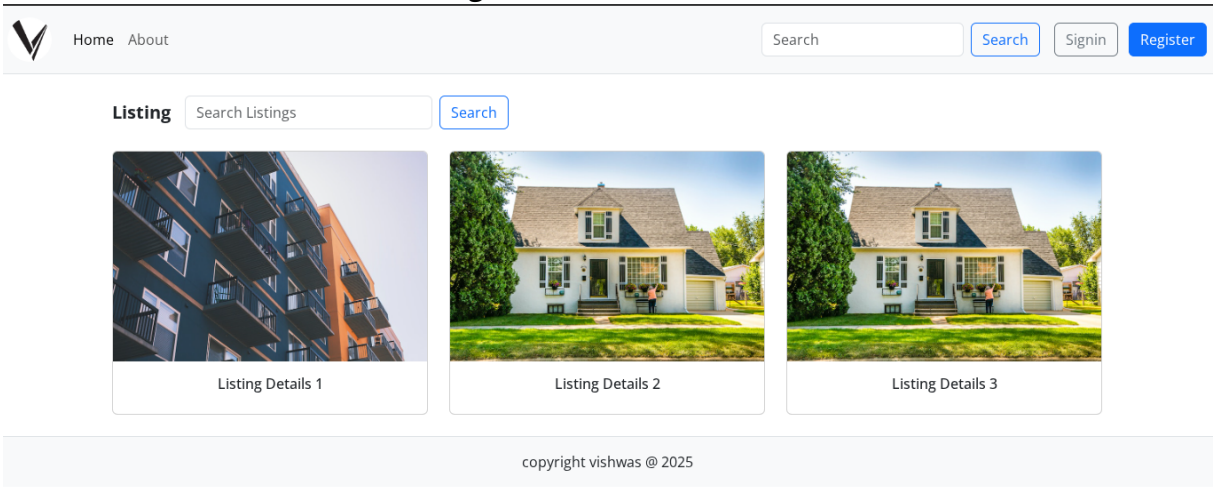


Dashboard

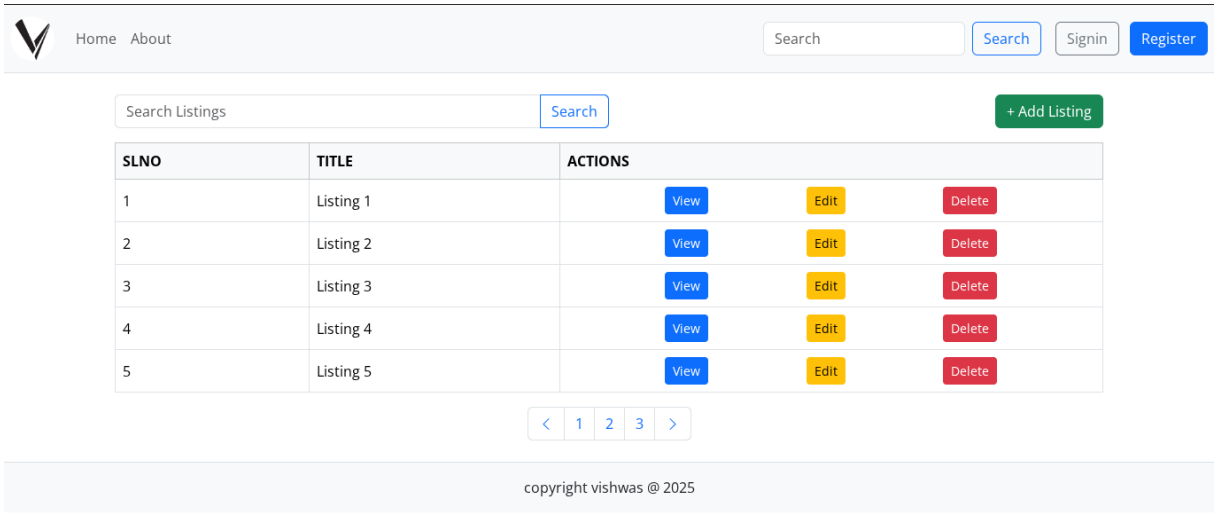


After this we work on typography (fonts) and Color Theory - primary, secondary & tertiary colors for our website, (70-20-10 rule etc..). We come up with a highlevel design

HomePage

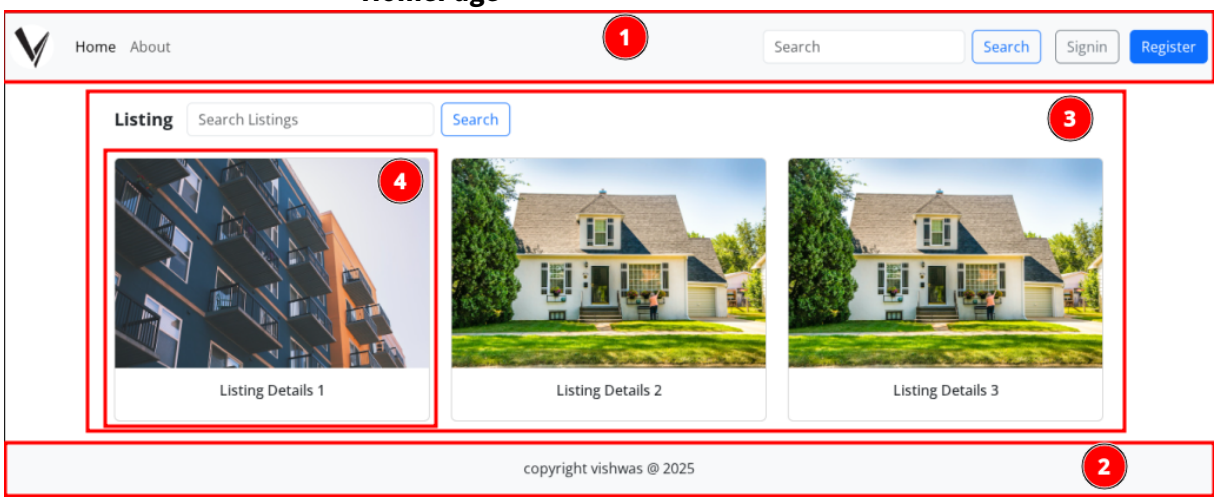


Dashboard



Once we design the pages as above we can break them into components as below

HomePage



Dashboard

We can drilldown into more detail components. With the above pictures we can create

- Page Components
  - HomePage
  - About
  - Dashboard and many more
- Individual Sub-Components
  - Layout
  - Header/AppNavBar (1)
  - ListingCard (3)
  - AppFooter (2)
  - ListingsTable (4)
  - ListingRow (5) and many more

We want our data to be seperate from our UI so we keep a seperate services layer which will be responsible for providing data for us. The project structure will be similar to as below

```
my-app/
├── public/
│   ├── index.html
│   └── favicon.ico
│
├── src/
│   ├── assets/                 # Images, icons, styles, fonts
│   │   ├── images/
│   │   └── styles/
│
```

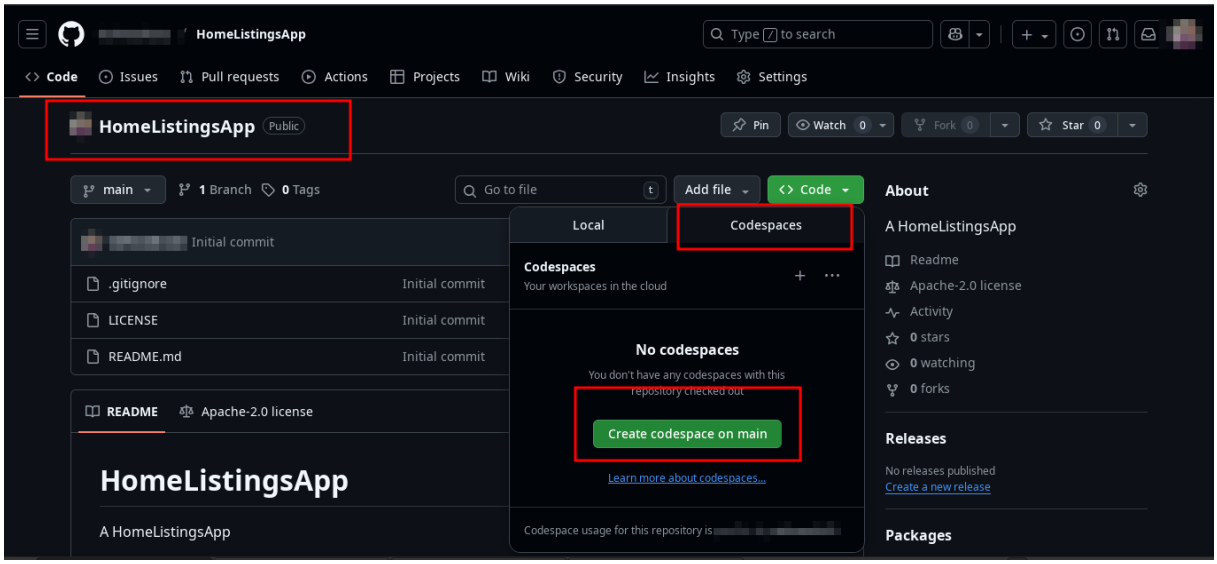
```
│   ├── components/            # Reusable UI components
│   │   ├── layout/
│   │   │   ├── Layout.jsx
│   │   │   ├── Header.jsx      # (AppNavBar)
│   │   │   ├── Header.css
│   │   │   └── Footer.jsx      # (AppFooter)
│   │   │   ├── Footer.css
│   │   │
│   │   ├── listings/
│   │   │   ├── ListingsTable.jsx
│   │   │   ├── ListingsTable.css
│   │   │   ├── ListingRow.jsx
│   │   │   ├── ListingRow.css
│   │   │
│   │   ├── common/             # Buttons, inputs, dialogs, etc.
│   │   │   ├── CustomButton.jsx
│   │   │   └── SearchBox.jsx
│   │   │
│   │   └── index.js            # Optional barrel export
│   │
│   ├── pages/                  # Full page views
│   │   ├── HomePage.jsx
│   │   ├── AboutPage.jsx
│   │   ├── DashboardPage.jsx
│   │
│   ├── services/               # API & business logic
│   │   ├── api/
│   │   │   ├── listingService.js  # CRUD listings
│   │   ├── httpClient.js          # axios/fetch wrapper (Future)
│   │   └── storageService.js      # localStorage/session helpers (Future)
│   │
│   ├── context/                # React Context API (Future)
│   │   ├── AuthContext.jsx
│   │   └── ListingsContext.jsx
│   │
│   ├── hooks/                  # Custom React hooks(Future)
│   │   ├── useAuth.js
│   │   └── useFetch.js
│   │
│   ├── routes/                 # Routing layer (Future)
│   │   └── AppRoutes.jsx
│   │
│   ├── App.jsx                 # Root component with Layout
```

```
│   ├── index.js                # Entry point
│   └── setupTests.js           # Jest/RTL tests
│
└── package.json
```

Now we will start with building our project. We can use local environment or online environment like Github Codespaces.

Let us use codespaces & hence we will

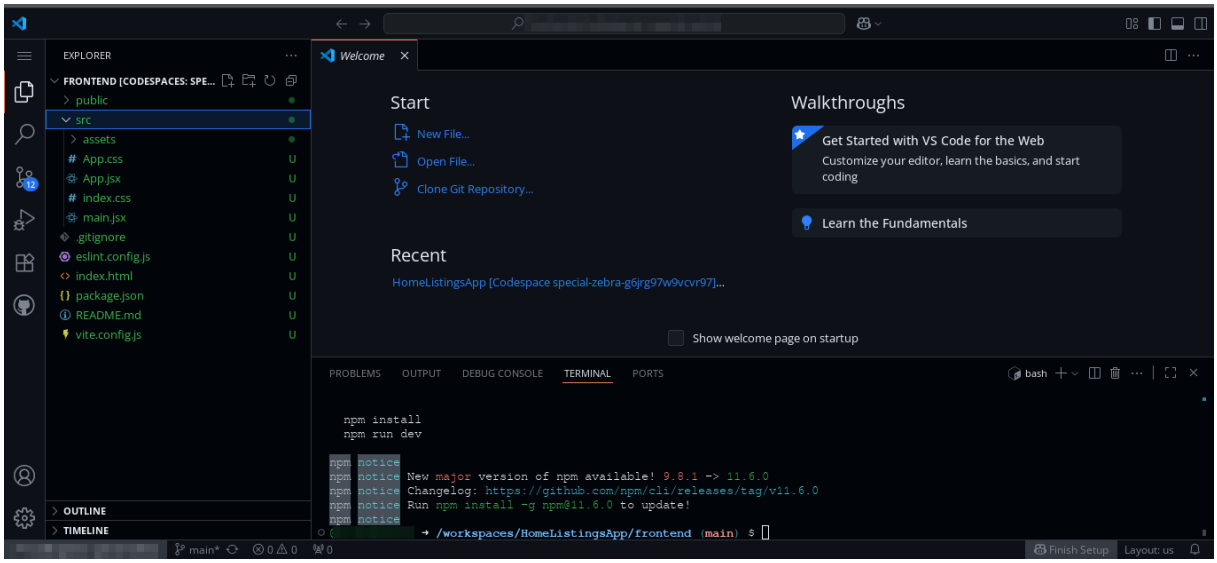
- create a git repo for our listingsapp.



- We will use the below command to create a react app using vite in current folder

`npm create vite@latest .`

follow the oncreeen instructions to select `react`, `javascript`



Next we have to install the dependencies

```
npm install
```

We will start the development server

```
npm run dev
```

We will start our development by

- delete `react.svg` in assets
- clear contents of `App.css` & `App.jsx`, `index.css`
- We will create the necessary folders of `components`, `pages`, `services`

```
mkdir -p src/assets/images src/assets/styles
mkdir -p src/services/api
mkdir -p src/components/layout src/components/listings src/components/common
mkdir -p src/pages
```

- next we will add the bootstrap to our project

```
npm install bootstrap
```

import once in `src/index.jsx`

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

just use Bootstrap classes directly in JSX. Also instead of using raw divs everywhere, we can use react-bootstrap which gives us sanitized components.

```
npm install react-bootstrap
```

Instead of writing raw Bootstrap classes, you can use React components.

- create AppNavbar, Footer, Layout Components.
  In General to build a component we do

1. Export the component
2. Define the function
3. Add markup

Example

```
export default function Profile() {
  return (
    <img
      src="https://i.imgur.com/MK3eW3Am.jpg"
      alt="Katherine Johnson"
    />
  )
}
```

4. Using a component we can nest it inside other components

AppNavBar

```
import { Navbar, Nav, Container, Button, Form, FormControl } from
 ↪  'react-bootstrap';

function AppNavBar() {
  return (
    <Navbar bg="light" expand="lg" className="border-bottom">
      <Container>
        <Navbar.Brand href="#">MyApp</Navbar.Brand>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav">
          <Nav className="me-auto">
            <Nav.Link href="/">Home</Nav.Link>
            <Nav.Link href="/about">About</Nav.Link>
          </Nav>
          <Form className="d-flex me-2">
            <FormControl type="search" placeholder="Search" className="me-2"
             ↪  />
```

```
            <Button variant="outline-primary">Search</Button>
          </Form>
          <Button variant="outline-secondary"
           ↪  className="me-2">Signin</Button>
          <Button variant="primary">Register</Button>
        </Navbar.Collapse>
      </Container>
    </Navbar>
  );
}

export default AppNavBar;
```

Footer.jsx

```
import { Container } from "react-bootstrap";

export default function Footer() {
  return (
    <footer className="bg-light border-top py-3 mt-auto">
      <Container className="text-center">
        <small className="text-muted">
          © vishwas @ 2025
        </small>
      </Container>
    </footer>
  );
}
```

Layout.jsx

```
import AppNavBar from "./AppNavBar";
import Footer from "./Footer";
import { Container } from "react-bootstrap";

export default function Layout({ children }) {
  return (
    <div className="d-flex flex-column min-vh-100">
      <AppNavBar />
      <Container className="flex-grow-1 my-4">
        {children}
      </Container>
      <Footer />
    </div>
```

```
  );
}
```

HomePage

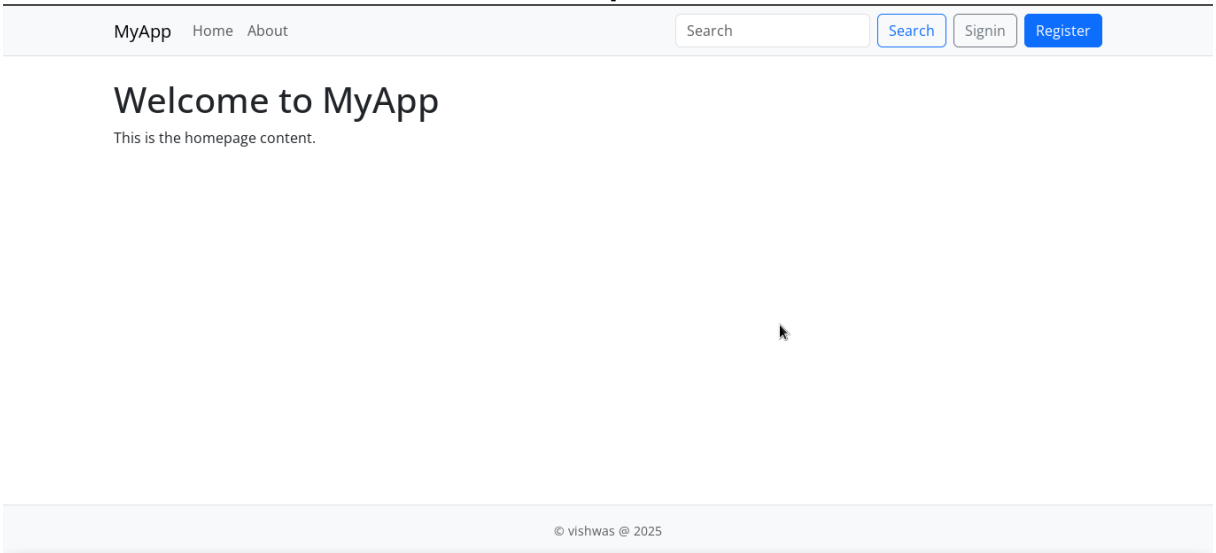
```
export default function HomePage() {
  return (
    <div>
      <h1>Welcome to MyApp</h1>
      <p>This is the homepage content.</p>
    </div>
  );
}
```

We will use these in App.jsx

```
import Layout from "./components/layout/Layout";
import HomePage from "./pages/HomePage";
function App() {
  return (
    <Layout>
      <HomePage />
    </Layout>
  );
}

export default App;
```

Output



For keeping the data seperate we will populate a sample data and simulate api in services.

We will start with a list of homeListings in HomePage.

```
let listings = [
  { id: 1, title: "Cozy Apartment in City Center", image:
 ↪  "public/images/house.jpg" },
  { id: 2, title: "Spacious Villa with Garden", image:
 ↪  "public/images/house.jpg" },
  { id: 3, title: "Modern Studio near Metro", image:
 ↪  "public/images/house.jpg" },
  { id: 4, title: "Luxury Penthouse with Pool", image:
 ↪  "public/images/house.jpg" },
  { id: 5, title: "Family Home in Suburbs", image: "public/images/house.jpg"
 ↪  },
];
```

We will create a card component to display on home page **ListingCard.jsx**

```
import { Card, Button } from "react-bootstrap";

export default function ListingCard({ listing }) {
  return (
    <Card className="shadow-sm h-100">
      <Card.Img
        variant="top"
        src={listing.image}
        alt={listing.title}
        style={{ height: "200px", objectFit: "cover" }}
      />
      <Card.Body className="d-flex flex-column">
        <Card.Title className="fs-6">{listing.title}</Card.Title>
        Listing Details
      </Card.Body>
    </Card>
  );
}
```

Use that in HomePage

```
import { Container, Row, Col } from "react-bootstrap";
import ListingCard from "../components/listings/ListingCard";

let listings = [
  { id: 1, title: "Cozy Apartment in City Center", image:
 ↪  "public/images/house.jpg" },
```

```
  { id: 2, title: "Spacious Villa with Garden", image:
 ↪  "public/images/house.jpg" },
  { id: 3, title: "Modern Studio near Metro", image:
 ↪  "public/images/house.jpg" },
  { id: 4, title: "Luxury Penthouse with Pool", image:
 ↪  "public/images/house.jpg" },
];

export default function HomePage() {
  return (
    <Container className="my-4">
      <h2 className="mb-4">Home Listings</h2>
      <Row xs={1} sm={2} md={3} lg={4} className="g-4">
        {listings.map((listing) => (
          <Col key={listing.id}>
            <ListingCard
              listing={listing}
            />
          </Col>
        ))}
      </Row>
    </Container>
  );
}
```

Output