TP1 : Landing Page Tutorial

Step 1: Setting Up Your React Project

1. Install Node.js: Ensure Node.js is installed. Download it from Node Js website
2. Create Your React App:

```
npx create-react-app react-landing-page
cd react-landing-page
npm start
```

- This will create a basic React app and start the development server.

Step 2: Project Structure

Create a `components` folder inside `src` to keep all your components organized:

css

```
src/
├── components/
│   ├── Header.js
│   ├── Hero.js
│   ├── Features.js
│   ├── Footer.js
├── App.js
└── index.js
```

Step 3: Creating the Header Component

Header.js: This component will display the navigation bar.

javascript

```
// src/components/Header.js
function Header() {
  return (
    <header className="header">
      <nav className="navbar">
        <h1 className="logo">My Landing Page</h1>
        <ul className="nav-links">
          <li><a href="#features">Features</a></li>
          <li><a href="#contact">Contact</a></li>
        </ul>
      </nav>
    </header>
  );
}
```

```
export default Header;
```

Step 4: Creating the Hero Section

Hero.js: This component will serve as the hero section of the landing page.
javascript

```
// src/components/Hero.js
function Hero() {
  return (
    <section className="hero">
      <div className="hero-content">
        <h2>Welcome to Our Product</h2>
        <p>Discover the features and benefits of our amazing
product.</p>
        <button className="cta-button">Get Started</button>
      </div>
    </section>
  );
}

export default Hero;
```

Step 5: Creating the Features Section

Features.js: This section will display the main features of your product.
javascript

```
// src/components/Features.js
function Features() {
  const features = [
    { title: 'Fast Performance', description: 'Our product is
lightning fast and reliable.' },
    { title: 'Easy to Use', description: 'User-friendly design that
anyone can navigate.' },
    { title: 'Responsive Design', description: 'Looks great on any
device, from phones to desktops.' }
  ];

  return (
    <section id="features" className="features">
      <h3>Key Features</h3>
      <div className="features-list">
        {features.map((feature, index) => (
          <div key={index} className="feature-item">
            <h4>{feature.title}</h4>
            <p>{feature.description}</p>
          </div>
        ))}
      </div>
    </section>
  );
}

export default Features;
```

Step 6: Creating the Footer

Footer.js: This will be the footer of your landing page.
javascript

```
// src/components/Footer.js
function Footer() {
  return (
    <footer className="footer">
      <p>&copy; 2024 My Landing Page. All rights reserved.</p>
    </footer>
  );
}

export default Footer;
```

Step 7: Building the Main App Component

App.js: Combine all the components to build the full landing page.
javascript

```
// src/App.js
import Header from './components/Header';
import Hero from './components/Hero';
import Features from './components/Features';
import Footer from './components/Footer';
import './App.css'; // Import your styles

function App() {
  return (
    <div className="App">
      <Header />
      <Hero />
      <Features />
      <Footer />
    </div>
  );
}

export default App;
```

Step 8: Adding CSS Styles

App.css: Create styles for a clean and responsive landing page. Below is a basic CSS setup:

css

```
/* src/App.css */
body {
  margin: 0;
  font-family: Arial, sans-serif;
}

.header {
  background: #282c34;
  padding: 1rem;
  color: white;
}

.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.logo {
  font-size: 1.5rem;
}

.nav-links {
  list-style: none;
  display: flex;
  gap: 1rem;
}

.nav-links a {
  color: white;
  text-decoration: none;
}
```

```css
.hero {
  background: #f0f4f8;
  padding: 5rem 1rem;
  text-align: center;
}

.hero-content h2 {
  font-size: 2.5rem;
  margin-bottom: 1rem;
}

.cta-button {
  background: #007bff;
  color: white;
  padding: 0.75rem 1.5rem;
  border: none;
  cursor: pointer;
  border-radius: 5px;
}

.features {
  padding: 2rem 1rem;
  text-align: center;
}

.features-list {
  display: flex;
  justify-content: center;
  gap: 2rem;
}

.feature-item {
  max-width: 200px;
}


.footer {
  background: #282c34;
  color: white;
  text-align: center;
  padding: 1rem 0;}
```