

lesson - 4

npm i react-router-dom

App.js

<BrowserRouter>

<Routes>

<Route path='/' element={<HomePage>}>

<Route path='/about' element={<AboutPage>}>

<Route path='/contact' element={<ContactPage>}>

<Route path='/services' element={<ServicesPage>}>

</Routes>

</BrowserRouter>

✓ components

✓ footer

footer.module.css

index.jsx

```
export class Footer extends Component {
```

```
    render() {
```

```
        return <div> Footer </div>
```

```
}
```

```
export default Footer;
```

✓ header

header.module.css

index.jsx

rec yoxib inleme, \$header - ga özgərtirəm.

✓ Layout

Layout.module.css

[index.jsx]

```
import Header from './header';
import Footer from './footer';

export class Layout extends Component {
  render() {
    return (
      <Fragment>
        <Header />
        {this.props.children}
        <Footer />
      </Fragment>
    );
  }
}

export default Layout
```

✓ pages

HomePage.jsx

```
import Layout from '../components/Layout';
export class HomePage extends Component {
  render() {
    return(
      <Layout>
        <Section> HomePage </Section>
      </Layout>
    );
  }
}

export default HomePage;
```

✓ header → ichiga yozamis

header.module.min.css

index.jsx

< Header >

< NavLink className="container d-flex justify-content-between align-items-center" >

< Link to="/" > Home </Link >

< Link to="/about" > About </Link >

< Link to="/services" > Services </Link >

< Link to="/contact" > Contact </Link >

sayt yang plan may state.

< NavLink ga organiz gysak actio

className actio boladdi

✓ pages

PortfolioPage.jsx

Dynamic page

< Link className="me-3" to="/portfolios/1" >

Portfolio 1

</Link >

< Link className="me-3" to="/portfolios/2" >...

< Link className="me-3" to="/portfolios/3" >...

< Link className="me-3" to="/portfolios/4" >...

PortfolioPage.jsx

yatagi page ochis kee gilib yozib olamis

App.js

< Route path="/portfolios/:portfolioId" >

element = {< Portfolio >} />

PortfolioPage.jsx

Let portfolioId = window.location.pathname.
split('/').at(-1);

Bizansini ushlab beradi.

App.js

Nested routes

```
<Route path='/'>
  <Route path='' element={<HomePage/>} />
  <Route path='about' element={<AboutPage/>} />
  ...
  <Route path='portfolios'>
    Route path='' element={<PortfoliosPage />} />
    Route path=':portfolioId' element={<PortfolioPage />} />
  </Route>
  </Route>
  <Route path='*' element={<NotFoundPage />} />
  ...
  </div>
```

~Pages

NotFound.jsx

```
<div>
  <h1> NotFound </h1>
  <Link to='/'> Home </Link>
</div>
```

Bizansini hozi kodimizda <Layout /> takrorlangan, shu larho öchiwb fashlab App.js ga

route path = '/' element = { < Layout > } , ,

gilib olib . < Layout > olasige { this.props
children } - ni ochildib

< Outlet >

yozamiz

▷ Code splitting -

initial load - Bolgan paytma da hamma
kodlar byuklari yotwadiki code splitting da
aynan usha saxfaga tegishli bolganlar yub Paradise.

App.js

■ import HomePage from '~~pages/HomePage~~'
. /pages / HomePage ' ;

gilib kommentar yage olib

const HomePage = lazy(() => import('./pages/
HomePage'));

hammasi yozib chigulash.

< Suspense fallback = { < Loading > } >

< BrowserRouter >

...

< BrowserRouter >

< Suspense >

▷ Lifecycle method

▷ Initial

- constructor

▷ Mounting

- ~~componentWillMount~~

- ~~render~~

- ~~componentDidMount~~

- render

- componentWillMount

- componentDidMount

lifecyclePage.jsx

```
constructor(props) {
  super(props);
  this.state = {
    counter: 0,
  };
  log('constructor')
}
```

```
componentWillMount() {
  log('componentWillMount')
}
```

```
componentDidMount() {
  log('componentDidMount')
}
```

```
render() {
  log('render')
}
```

```
}
```

natija:

constructor

componentWillMount

componentDidMount

▷ Updating

- shouldComponentUpdate
- componentDidUpdate

▷ Unmounting

- componentWillUnmount - remove AddListener, aborting network request
- componentWillUnmount, componentDidMount dan key di display

natija; componentDidMount

componentWillUnmount

npm i axios

server

index.js

import axios from 'axios';

const request = axios.create({

baseURL: 'https://jsonplaceholder.typicode.com/

timeout: 1000,

})

export default request;

~ pages

Post 5 page

```
state = {
  posts: [],
  loading: false,
},
async getPosts() {
  try {
    this.setState({ loading: true })
    let { data } = await request('posts');
    this.setState({ posts: data });
    log(data)
  } catch (err) {
    log(err)
  } finally {
    this.setState({ loading: false })
  }
}
componentDidMount() {
  this.getPosts();
}
render() {
  const { posts, loading } = this.state;
  if (loading) (<ServiceLoader />);
}
```

```
(posts.map((post, i) => PostCards.key = {i})  
  { ... post } ,>)  
  ) } .
```

| vcards
| PostCards.jsx

```
const { id, title, body } = this.props;
```

```
<h3> {id}. {title} </h3>  
<p> {body} </p>
```

| v loading
| ServerLoading.jsx

```
<div> Server Loading ... </div>
```