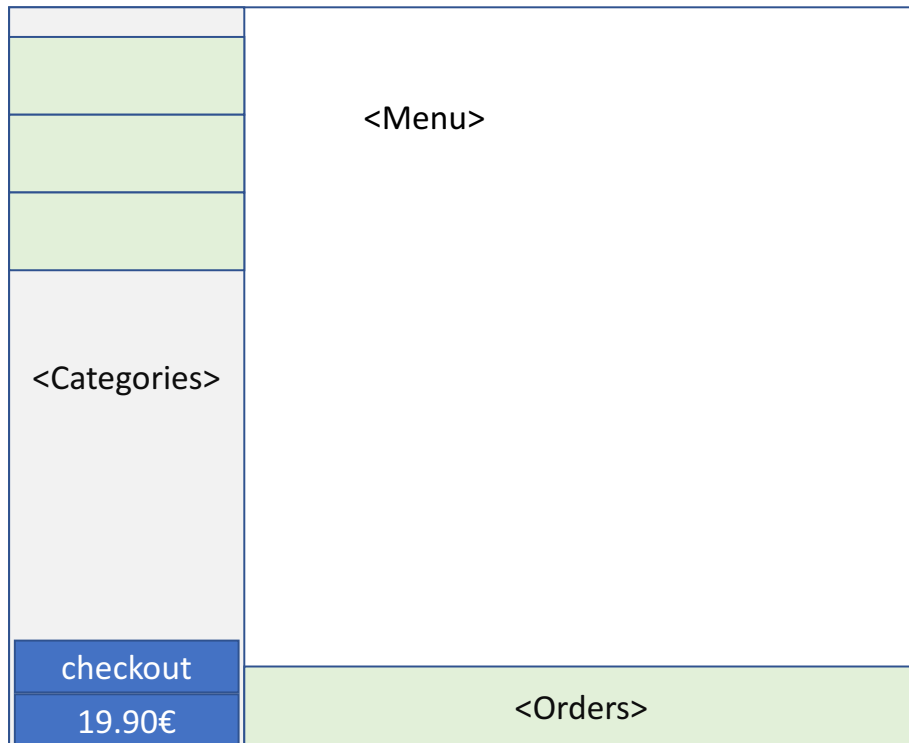
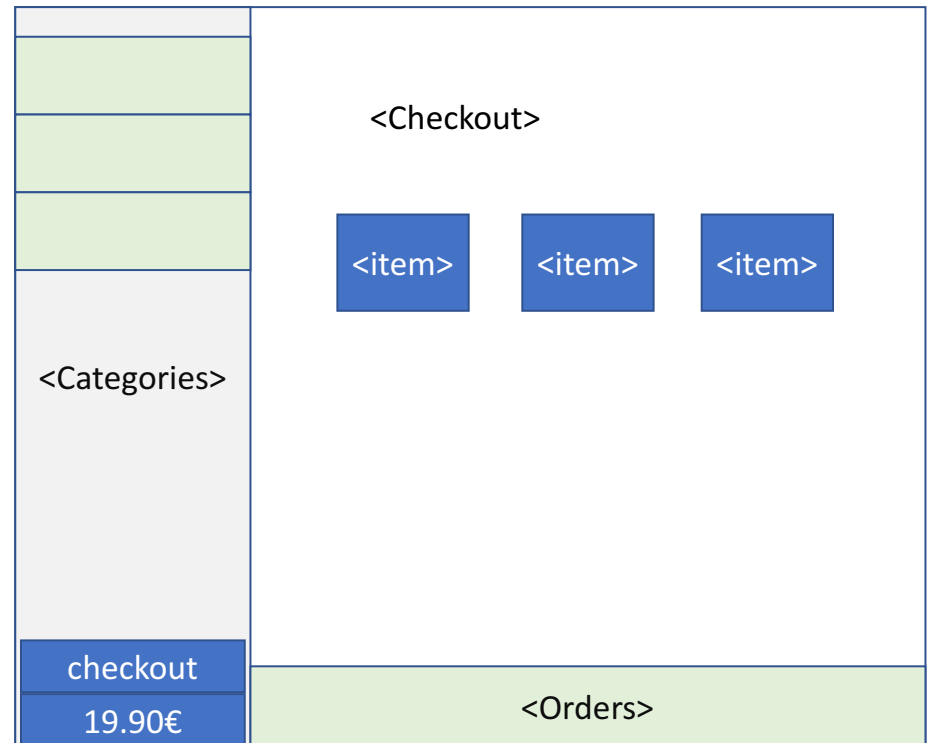


url: /  
<HomePage>

url: /shop/  
<Shop> <Menu>



url: /shop/checkout  
<Shop><Checkout>



state --- orders, menu, category,active,

url: /shop/userinfo  
<Shop><UserInfo>

back

19.90€

Name

Address

Tel. nr

Order

<Orders>

Confirmation ?

url: /admin/orders  
<Admin>

orders	
menu	
history	

url: /admin/menu  
<Admin>

	drinks	meals	snacks
orders			
menu	photo	name	price
history			
	<div></div> <div>name price category</div>	<div></div> <div>name price category</div>	

1. node serverio failas
2. sukuriam client papke ir i ja instaliuojam react
3. react projekte mum reikia node-sass ir sass-loader biblioteku
4. isskleidiziam webpac kompilieri ( npm run eject )
5. I webpac.config.dev ir webpac.config.prod failus sudedam sass nustatymus
6. paleisti node ir react kartu

concurrently leidzia sujungti dvi terminalo komandas i viena

```
"scripts": {  
  "server": "nodemon app.js",  
  "client": "npm start --prefix client",  
  "dev": "concurrently \"npm run server\" \"npm run client\" ",  
  "test": "echo \"Error: no test specified\" && exit 1"  
},
```

paleidimas **npm run dev**

**proxy client dalyje, package.json failas**

```
"proxy":{  
  "/api":{  
    "target":"http://localhost:9000"  
  },  
  "/images":{  
    "target":"http://localhost:9000"  
  }  
},
```

7. Sukuriam React komponentus reikalingus aplikacijai paleisti
8. Sass failai
9. redux integracija
10. asinchroninis redux    redux-thunk biblioteka  
    sinchroninis – viskas paeiliui  
    asinhroninis – prisingai (kai kurios funkcijos vykdymas gali buti atidetas)
11. i index.js (client) importuoju applyMiddleware, compose

```
{  
  name:'Chicken burger',  
  price: 12,  
  img: ' /images/menu/...'  
  category:'burgers',  
  instocl:true  
}
```

## paths

myweb.com/admin/

+

img/one.png

myweb.com/admin/img/one.png

myweb.com/admin/

+

/img/one.png

myweb.com/img/one.png



## SQL & NoSQL DB's

SQL

table(users)

name	age	position

NoSQL

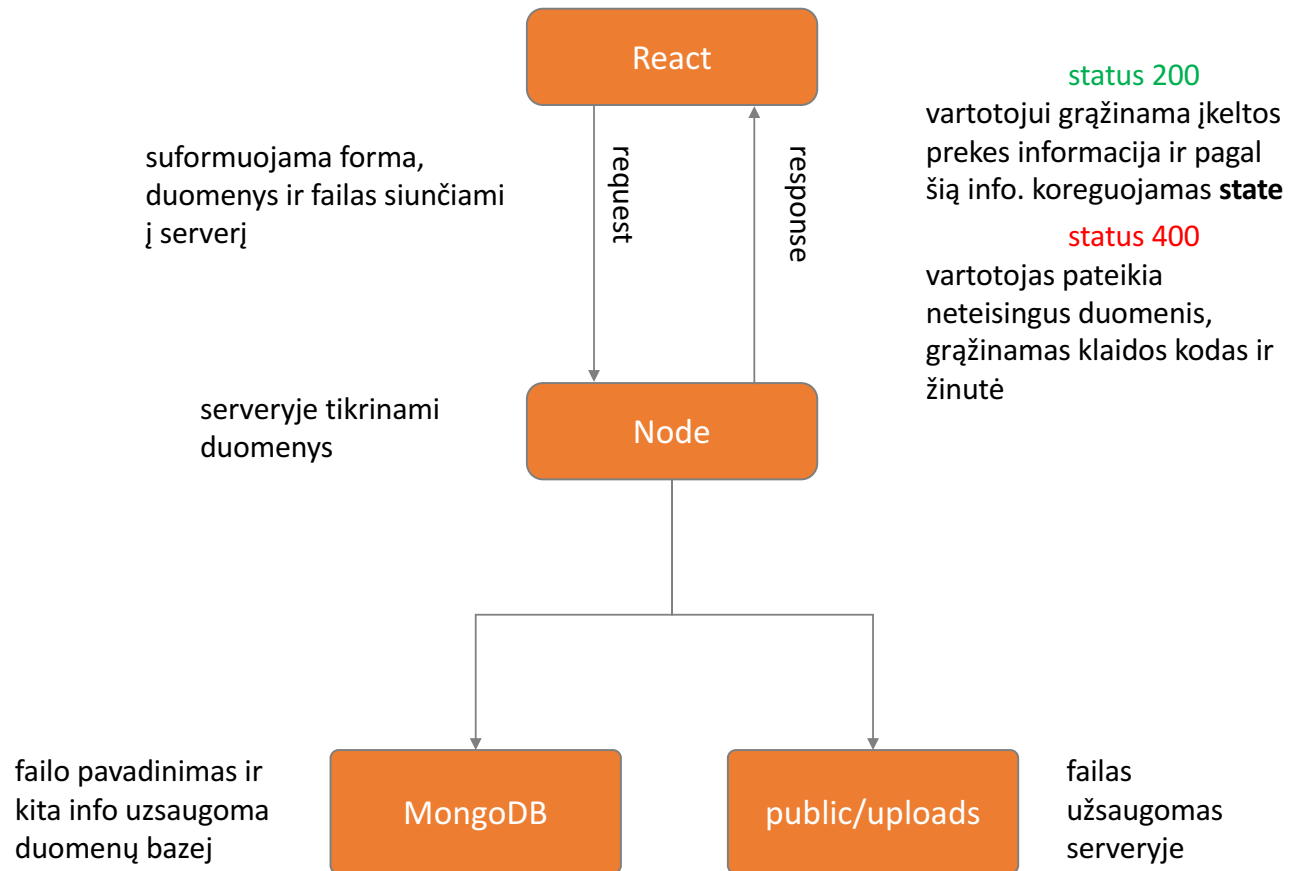
collection (users)

document  
property  
value

```
{  
  name:'john',  
  age: 26,  
  position:'admin'  
}
```

```
{  
  name:bill',  
  age: 36,  
  salary:'1000'  
}
```

## File uploads



## Backend

### #1.

Serverinėje dalyje parašom route kuriuo galėsime įkelti failą.

### #2.

Failų įkėlimui ir užsaugojimui naudojame npm modulį **multer**.

### #3.

Suformuojame klaidų gaudymo bloką.

### #4.

Naudojant Postman testuojam failų įkėlimą.

### #5.

Papildome užklausą su papildoma informacija iš formos. Suformuojame objektą ir padarome įrašą į DB.

### #6.

Grąžiname į frontend žinutę apie padarytą įrašą arba gautą klaidą.

## Frontend

### #7.

Integruojame **react-dropzone** modulį patogiam failų įkėlimui

### #8.

Sukuriame metodą, kurio pagalba bus siunčiami duomenys į serverį

### #9.

sukuriam objektą pagal FormData klasę. Jį užpildome per formą gautais duomenimis

### #10.

Naudojant axios modulį, siunčiame duomenis į serverį (request)

### #11.

Gautą atsakymą (response) atvaizduojame vartotojui papildydami redux ar react state