



LEARNING
MANAGEMENT
SYSTEMS

DADABOYEV SARDOR

JAVASCRIPT

VA

UNING IMKONIYATLARI

JS



**MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI**

INDIVIDUAL LOYIHA

**JAVASCRIPT
VA
UNING IMKONIYATLARI**



DADABOYEV SARDOR



513-19



QAYUMOVA GULSHAN



DEADLINE 10.06.2022

MUNDARIJA

1

BO'LIM JAVASCRIPT HAQIDA NAZARIYA

1.0

JAVASCRIPT o'zi nima 4

1.1

JAVASCRIPT brauzerda nima qila oladi... 6

1.2

JAVASCRIPT brauzerda nima qila olmaydi 6

1.3

Nima uchun JAVASCRIPT ajoyib til..... 8

1.4

JAVASCRIPT ustida qurilgan tillar..... 10

2

BO'LIM JAVASCRIPT AMALIY KO'RSATMALAR

2.0

Javascriptda amaliy ko'rstmalar..... 11

2.1

javascriptda kommentariyalar..... 13

2.2

javascript uchun kerakli intrumentlar..... 14

2.3

javascriptda matematik metodlar... 15

2.4

javascriptda massiv metodlari. 19

2.5

javascriptda funksiyalar 21

2.6

javascriptda destrukturizatsiya..... 22

2.7

NodeJs haqida..... 24

2.8

Server yaratamiza..... 25

2.9

Expressda server yaratamiza..... 26

3

MERN STACK..... 27

3.0

ReactJS..... 28

3.1

MERN da API chiqaramiz... 29

3.2

Ma'limotlar omboriga data joylash..... 30

4

VUEJS 35

5

Xulosa va Foydalangan adabiyotlar..... 40

JAVASCRIPTGA KIRISH

Keling javascriptni nima afzalliklari bor , nima imkoniyatlari bor va boshqa texnologiyalar bilan qanday qilib integratsiya qiladi, shularni ko'rib chiqamiz

JAVASCRIPT o'zi nima?

Boshida Javascript veb sahifalarni jonlantirish uchun yaratilgan edi

Bu dasturlash tilida yozilgan dasturlar scriptlar deb ataladi

Skriptlar oddiy matn sifatida taqsimlanadi va bajariladi. Ularni ishga tushirish uchun maxsus tayyorgarlik yoki kompilyatsiya kerak emas

Bu Javascriptni boshqa dasturlash tillaridan ajratib turadi , misol uchun "Java" dan

Nimaga Javascript?

Boshida Javascript yaratilayotganda , uning nomi <<LiveScript>> edi . Aynan usha paytda "JAVA" dasturlash tili juda mashhur edi , va JavaScript deb shunchaki nomiga qo'shib qo'yishgan , "подход" qilishgan , lekin javascriptni "java" bilan adashtirmaslik lozim.

Vaqt o'tishi bilan JavaScript ECMAScript deb nomlangan spetsifikatsiyasi sababli to'liq mustaqil tilga aylandi va endi Java bilan hech qanday aloqasi yo'q.

Bugungi kunda JavaScript nafaqat brauzerda, balki serverda yoki "JavaScript" deb nomlangan maxsus dasturga ega bo'lgan boshqa qurilmada ham amalga oshirilishi mumkin.

Brauzer o'z dvigateliga ega, ba'zan " virtual JavaScript mashinasi "deb ataladi.

Turli dvigatellar turli "kod nomlari"ga ega. Misol uchun:

V8-Chrome va Opera-da.

SpiderMonkey-Firefox-da.

... IE ning turli versiyalari uchun "Trident" va "Chakra", Microsoft Edge uchun "ChakraCore", Safari uchun "Nitro" va "SquirrelFish" va boshqalar.

Ushbu nomlar bilish uchun foydalidir, chunki ular ishlab chiquvchilar uchun maqolalarda tez-tez ishlatiladi. Biz ularni ham ishlatamiz. Misol uchun, agar "x funksiyasi v8 tomonidan qo'llab-quvvatlansa", "X" Chrome va Opera-da ishlaydi.

Dvijo'klar qanday ishlaydi (движок)

Dvijo'klar murakkab. Lekin asoslarini tushunish oson.

1. Dvijo'k (o'rnatilgan, agar u brauzer bo'lsa) skript matnini o'qiydi, ("tahlil qiladi").
2. Keyin skriptni mashina tiliga aylantiradi ("kompilyatsiya").
3. Shundan so'ng, mashina kodi ishga tushiriladi va juda tez ishlaydi.

Dvigatel har bir bosqichda optimallashtirishni qo'llaydi. U hatto ish vaqtida tuzilgan skriptni ko'zdan kechiradi, u orqali o'tadigan ma'lumotlarni tahlil qiladi va olingan ma'lumotlarga tayanib, mashina kodiga optimallashtirishni qo'llaydi. Natijada, skriptlar juda tez ishlaydi.



Rasm-1 (aktual browserlarda javascript qo'llaniladi)

Javascript nima qila oladi?

Zamonaviy javascript bu xavfsiz dasturlash tilidir Javascriptning imkoniyatlari qaysi muhitda ishlayotganligiga bog'liq. Misol uchun NodeJS , brauzerga fayllarni yuklashga, o'qishga, va tarmoqli so'rovlar yuborishga yordam beradi

Misol uchun Brauzerda javascriptning imkoniyatlari:

1. Web sahifaga yangi HTML kod qo'shish, tayyor holatdagi sahifani o'zgartirishga, va still berishga imkon beradi
2. Web sahifadan foydalanayotgan foydalanuvchini ilg'ash , Bu asosan sichqonchani bosganda , kursorni yurg'izganda , yoki klaviaturani bosganda
3. Serverlarga so'rov yuborish, fayllar yuklab olish , ularni joylashtirish, (asosan AJAX, COMET, AXIOS yordamida qilinadi) ni imkonini beradi
4. Kukilarni (Cookie) olish , o'rnatish , foydalanuvchiga savol berish, xabar yuborish, ko'rsatishni imkonini beradi
5. Klient tarafida ma'lumotlarni saqlash (localStorage) imkonini beradi

Javascript nima qila olmaydi?

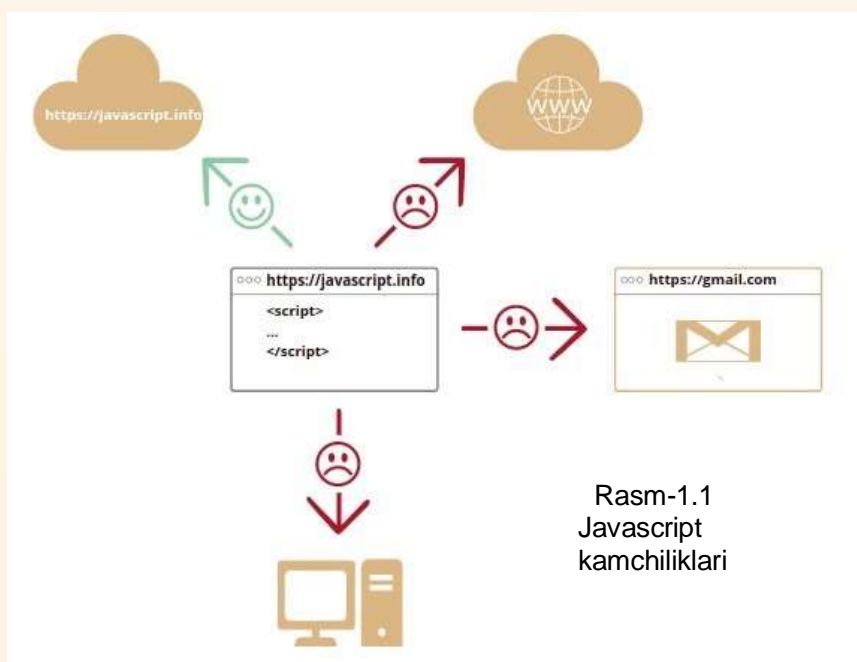
1. Veb-sahifadagi JavaScript-ni qattiq diskdagi o'zboshimchalik bilan fayllarni o'qish/yozish, ularni nusxalash yoki dasturlarni ishga tushirish mumkin emas. OS tizimining xususiyatlariga bevosita kirish imkoni yo'q.
2. Zamonaviy brauzerlar fayllar bilan ishlashga imkon beradi, lekin cheklangan kirish imkoniyati mavjud va foydalanuvchi faqat faylni brauzer oynasiga "sudrab tashlash" yoki <input>tegi bilan tanlash kabi muayyan harakatlar qilsa, uni ta'minlaydi.

3. Kamera/mikrofon va boshqa qurilmalar bilan ishlash usullari mavjud, ammo ular aniq foydalanuvchi ruxsatini talab qiladi. Shunday qilib, JavaScript-ni qo'llab-quvvatlaydigan sahifa veb-kamerani sezilarli darajada yoqmaydi, nima sodir bo'lishini kuzata olmaydi va FSB-ga ma'lumot yubora olmaydi.

4. Turli oyna / yorliqlar bir-birlarini bilishmaydi. Ba'zan JavaScript-ni ishlatib, bitta oyna boshqa Oynani ochadi. Biroq, bu holatda ham, bir sahifadan JavaScript-ni turli saytlardan (boshqa domen, protokol yoki portdan) kelgan bo'lsa, boshqasiga kirish imkoni yo'q. Bunga "bir xil manba siyosati" (bir xil kelib chiqadigan siyosat) deyiladi. Ushbu cheklovni chetlab o'tish uchun ikkala sahifa ham bunga rozi bo'lishi va maxsus ma'lumot almashadigan JavaScript kodini o'z ichiga olishi kerak.

Ushbu cheklov yana foydalanuvchi xavfsizligi uchun zarur. Sahifa <https://anysite.com>, foydalanuvchi ochilgan, URL bilan boshqa brauzer yorlig'iga kirish kerak emas <https://gmail.com> va u erdan ma'lumot o'g'irlash.

5. JavaScript-ni joriy sahifa kelgan server bilan osongina muloqot qilishi mumkin. Ammo boshqa saytlardan/domenlardan ma'lumotlarni olish qobiliyati cheklangan. Bu printsiplial jihatdan mumkin bo'lsa-da, uzoq tomon bilan aniq rozilikni (HTTP sarlavhalarida ifodalangan) talab qiladi. Shunga qaramay, bu xavfsizlikning cheklanishidir.



Agar JavaScript brauzerdan tashqarida ishlatilsa, masalan, serverda bunday cheklovlar qo'llanilmaydi. Zamonaviy brauzerlar plaginlar/kengaytmalarni taqdim etadi, ular orqali qo'shimcha ruxsatnomalar talab qilinishi mumkin.

Nima uchun JAVASCRIPT ajoyib til

Javascriptning minimum 3 ta asosiy kuchli taraflari mavjud

1. HTML/CSS bilan to'la integratsiya
2. Oddiy narsalar oddiy amalga oshiriladi
3. Barcha asosiy brauzerlar tomonidan qo'llab-quvvatlanadi va sukut bo'yicha yoqilgan.

JavaScript-bu uchta narsani birlashtiradigan yagona brauzer texnologiyasi.

Bu JavaScript-ni maxsus va ko'rkam qiladi. Shuning uchun brauzerda interfeyslarni yaratish uchun eng keng tarqalgan vosita hisoblanadi.

Albatta, JavaScript-ni nafaqat brauzerlarda, balki serverda, mobil qurilmalarda va hokazolarda ham ilovalar qilish imkonini beradi. (bu haqda keyinroq gaplashamiz)



Rasm-2 (HTML web sahifa skeletei, css web sahifa stili , javascript esa web sahifani jonlantiradi)

JAVASCRIPT ustida qurilgan tillar

JavaScript sintaksisi barcha ehtiyojlar uchun mos emas. Turli odamlar turli imkoniyatlarga ega bo'lishni xohlashadi.

Bu tabiiydir, chunki loyihalar boshqacha va ular uchun talablar ham boshqacha.

Shunday qilib, so'nggi paytlarda brauzerda ishga tushirilgunga qadar JavaScript tiliga konvertatsiya qilinadigan ko'plab yangi tillar paydo bo'ldi.

Zamonaviy vositalar transpilatsiyani juda tez va oshkora qiladi, aslida ishlab chiquvchilarga kodni boshqa tilda yozishga imkon beradi, uni avtomatik ravishda JavaScript-ni "qopqoq ostida" o'zgartiradi.

Bunday tillarning namunalari:

CoffeeScript JavaScript uchun "sintaktik shakar" qo'shadi. U toza va ixcham kodni yozishga imkon beruvchi qisqa sintaksisni taqdim etadi. Odatda Ruby dasturchilar kabi.



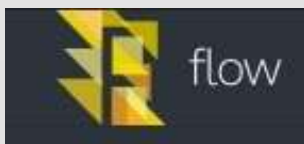
Rasm-3
CoffeScript
logotipi

TypeScript katta va murakkab tizimlarni ishlab chiqish va qo'llab-quvvatlashni soddalashtirish uchun "qattiq yozish" ni qo'shishga qaratilgan. Microsoft Tomonidan Ishlab Chiqilgan.



Rasm-4
TypeScript
logotipi

Flow shuningdek, yozishni qo'shadi, lekin boshqacha. Facebook Tomonidan Ishlab Chiqilgan.



Rasm-5
Flow
logotipi

Dart alohida turadi, chunki u brauzerdan tashqarida ishlaydigan o'z dvigateliga ega (masalan, mobil ilovalarda). Dastlab Google tomonidan JavaScript-ni almashtirish sifatida taklif qilingan, ammo hozirgi vaqtda uning efirga uzatilishi yuqoridagi tillar bilan bir xil bo'lishi kerak.



Rasm-6
Dart
logotipi

Brython Pythonni JavaScript-ni tarjima qiladi, bu esa JavaScript-ni holda sof Pythonda ilovalarni yozishga imkon beradi.



Rasm-7
Brython
logotipi

Boshqalar ham bor. Ammo, agar biz ushbu tillardan birini ishlatsak ham, nima qilayotganimizni tushunish uchun JavaScript-ni bilishimiz kerak.

Shunday qilib:

1. JavaScript dastlab faqat brauzer uchun yaratilgan, ammo hozir ko'plab boshqa platformalarda ishlatilgan.
2. Bugungi kunda JavaScript HTML / CSS bilan to'liq integratsiyaga ega bo'lgan brauzer uchun eng keng tarqalgan til sifatida noyob mavqega ega.
3. Ko'pgina tillar qo'shimcha funktsiyalarni taqdim etish uchun JavaScript-da "tarjima qilingan" bo'lishi mumkin. JavaScript-ni o'zlashtirganidan keyin ularni kamida qisqacha ko'rib chiqish tavsiya etiladi.



Rasm-8
Java ukam , bu
Javascript

2-BO'LIM

JAVASCRIPT

AMALIY KO'RSATMALAR

```
document.getElementById(div).innerHTML = errEmail;
else if (i==2)
{
    var atpos=inputs[i].indexOf("@");
    var dotpos=inputs[i].lastIndexOf(".");
    if (atpos<1 || dotpos<atpos+2 || dotpos>inputs[i].length-1)
    document.getElementById('errEmail').innerHTML = "Invalid email address";
    else
    document.getElementById(div).innerHTML = "Valid email address";
}
```

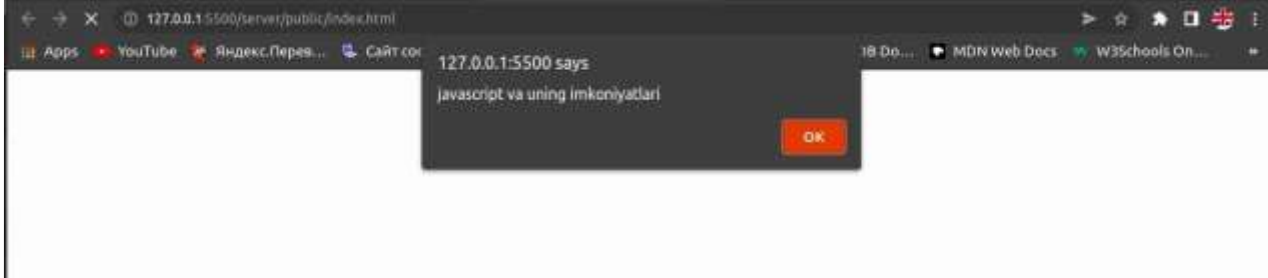
Rasm-9 Javascript kodi

Javascriptni ulash uchun biz script tegidan foydalanamiza

```
server > public > index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>DoskinMarket</title>
8 </head>
9 <body>
10
11   <script>
12     alert("javascript va uning imkoniyatlari")
13   </script>
14
15   <h1>Individual Loyiha </h1>
16   <h2>javascript imkoniyatlari</h2>
17   <!-- <form action="", >
18     <input type="text", placeholder="product name">
19     <input type="text", placeholder="product price">
20     <input type="text", placeholder="product type">
21
22   </form> -->
23
24 </body>
25 </html>
```

Rasm-10 Javascript kodida alert ni bajaryapmiz

Biz hozir script tegini ichida alert komandasi yordamida “Javascript va uning imkoniyatlari “ degan gapni , paydo bo’ladigan oynaga chiqaramiz:



Ok tugmasini bosgandan keyin bizning HTML teglarimiz ishga tushdi



Qattiq rejim (Строгий режим) — "use strict"

Uzoq vaqt davomida JavaScript-ni qayta muvofiqligi bilan muammosiz ishlab chiqdi. Oldingi funkcionallik o'zgarmagan bo'lsa-da, tilga yangi xususiyatlar qo'shildi.

Ushbu yondashuvning afzalligi shundaki, mavjud kod ishlashni davom ettirdi. Va ahvolga tushgan-JavaScript yaratuvchilari tomonidan qabul qilingan har qanday xato yoki nomukammal qaror abadiy tilida qolib ketdi.

Shunday qilib, ECMAScript 5 (ES5) paydo bo'lgan 2009 yilgadan keyin o'zgarish bo'ldi. U tilga yangi imkoniyatlar qo'shdi va mavjud bo'lgan funkcionallarni ayrimlarini o'zgartirdi. Eski kodni ishlatish uchun, avvalgidek, bunday o'zgarishlar sukut bo'yicha qo'llanilmaydi. Shuning uchun biz ularni maxsus direktiv bilan aniq faollashtirishimiz kerak:"use strict".

```
1 | "use strict";
2 | alert('Привет, мир!');
```

Rasm-11 (salom dunyo)

```
<script>
    alert("javascript va uning imkoniyatlari")
    // kommentariya yozilishi
</script>
```

ctrl+/*

```
/*
    for(let i=0; i<=15; i++){
        console.log("kommentariyani umumiy qo'llash")
    }
*/
```

```
// Javascriptda o'zgaruvchilar
let a = 5 //oddiy o'zgaruvchilar

const b = 10 //konstantalar

let name = "Sardor" //String turdagi o'zdaruvchilar

let produktalar = ["olma", "nok", "banan", "non", "yog'", "anor","uzum", "Shatoli"] // massivlar

const bbject = { //obyektlar ,obyektlar o'ziga yuqoridagi barcha o'zgaruvchilarni o'z ichiga oladi
    name: "Sardor",
    surname: "Dadaboyev",
    guruh: "513-19",
    loyiha: "Javascript va uning imkoniyatlari",
    JavascriptImkoniyatlariKop = true
}
```

```
// javascriptda ma'lumot tiplari
// Undefined ,
// Boolean,
// Number,
// String,
// BigInt,
// Symbol,
// Null,
// Obyektlar,
// Funksiyalar ,
// Massivlar,
```

JavaScript bilan ishlash uchun bizga nima ke

Albatta buning uchun bizga Redaktor koda kerak Kod muharriiri Kod muharlaridan eng poplarnilari bular VsCode, Atom, WebStrom, Sublime text editor .Bular juda qulay , lekin eng ommaboblari bular: VS Code bilan Web Storm dir.



Rasm-11 VS CODE



Rasm-12 web storm



Rasm-13 atom



Rasm-14 sublime text editor

JAVASCRIPTDA SATRLAR BILAN ISHLASH

```
server >  practice.js > ...
```

```
1 //toUpperCase
2 let satr = "sardor"
3 console.log(satr.toUpperCase())
4
5
```

```
[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
SARDOR
```

```
[Done] exited with code=0 in 0.034 seconds
```

```
// to lowerCase
let satr2 = "MANI KICKINA QILOLMISILA"
console.log("qilolimiza", satr.toLowerCase())
```

```
SARDOR
qilolimiza sardor
```

```
[Done] exited with code=0 in 0.035 seconds
```

```
//substr
```

```
let satr3 = "Individual Loyiha"
let kesamiz = satr3.substr(0, 10)
console.log(kesamiz)
```

```
[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
Individual
```

```
[Done] exited with code=0 in 0.034 seconds
```

```
//substring
let str = 'abcde';
let sub = str.substring(1, 3);
console.log(sub);
```

```
[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
bc
```

```
[Done] exited with code=0 in 0.083 seconds
```

```
//slice
let str = 'abcde';
let sub = str.slice(1);

console.log(sub);
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.083 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
bcde

[Done] exited with code=0 in 0.034 seconds

```
//indexOf
let str = "men javascriptni o'rganyapman";
console.log(str.indexOf('java'));
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.034 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
4

[Done] exited with code=0 in 0.033 seconds

```
//join
let arr = [1, 2, 3];
let str = arr.join('-');

console.log(str);
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.033 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
1-2-3

[Done] exited with code=0 in 0.036 seconds

```
//o'zgartirish orqadan boshlash
let str = '123456789';
let arr1 = str.split('');
let arr2 = arr1.reverse();
let result = arr2.join('');

console.log(result);
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.036 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
987654321

[Done] exited with code=0 in 0.076 seconds

JAVASCRIPT MATEMATIK METODLARI

```
//darajaga oshirish  
console.log("3 ning 4 darajasi",Math.pow(3, 4));
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.036 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
3 ning 4 darajasi 81

[Done] exited with code=0 in 0.063 seconds

```
//ildiz chiqarish  
console.log("256 dan ildiz ",Math.sqrt(256), "chiqadi");
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.063 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
256 dan ildiz 16 chiqadi

[Done] exited with code=0 in 0.035 seconds

```
//sonni yaxlitlash  
console.log(Math.round(6.4));
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.035 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
6

[Done] exited with code=0 in 0.035 seconds

```
//katta songacha aylantirish, okruglyat qilish  
console.log(Math.ceil(3.00001));
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.035 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
4

[Done] exited with code=0 in 0.036 seconds

```
// verguldan keyin nechtagacha ko'rsatich
let num = 1.1111111;
console.log(num.toFixed(3));
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.048 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
1.111

[Done] exited with code=0 in 0.058 seconds

```
// verguldan keyin nechtagacha ko'rsatich va okruglyat qilish
let num = 678.19324;
console.log(num.toPrecision(4));
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.058 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
678.2

[Done] exited with code=0 in 0.035 seconds

```
// kichik songacha yaxlitlash
console.log(Math.floor(2.9999));
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.035 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
2

[Done] exited with code=0 in 0.036 seconds

```
// hozir ko'rasila
let num = 12;
console.log(num.toPrecision(3));
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.036 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
12.0

[Done] exited with code=0 in 0.034 seconds

JAVASCRIPT MASSIV METODLARI

```
// massivni ohiriga element qo'shish
let arr = ['a', 'b', 'c', 'd', 'e'];

arr.push('1', '2');
console.log(arr)
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

```
[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
[
  'a', 'b', 'c',
  'd', 'e', '1',
  '2'
]

[Done] exited with code=0 in 0.038 seconds
```

```
// massivni boshiga element qo'shish
let arr = ['a', 'b', 'c', 'd', 'e'];
arr.unshift('1', '2');

console.log(arr);
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

```
[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
[
  '1', '2', 'a',
  'b', 'c', 'd',
  'e'
]

[Done] exited with code=0 in 0.037 seconds
```

```
// massivni ohirgi elementni o'chirish
let arr = ['a', 'b', 'c', 'd', 'e'];

arr.pop();
console.log(arr);
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

```
]

[Done] exited with code=0 in 0.037 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
[ 'a', 'b', 'c', 'd' ]

[Done] exited with code=0 in 0.035 seconds
```

```
// massivdan qirqib olish
let arr = ['a', 'b', 'c', 'd', 'e'];
let sub = arr.slice(0, 2);
```

```
console.log(sub);
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
['a', 'b']

[Done] exited with code=0 in 0.037 seconds

```
// massivdan aniq elementlarni o'chirish
let arr = ['a', 'b', 'c', 'd', 'e'];
```

```
arr.splice(1, 3);
console.log(arr);
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
['a', 'e']

[Done] exited with code=0 in 0.045 seconds

```
// massiv birinchi elementni o'chirish
let arr = ['a', 'b', 'c', 'd', 'e'];
```

```
arr.shift();
console.log(arr);
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
['b', 'c', 'd', 'e']

[Done] exited with code=0 in 0.036 seconds

```
// massivdan bir nechta metoddan foydalanamiza
```

```
let arr = ['1', '2', '3', '4', '5', '6'];
let result = [];
```

```
while (arr.length > 0) {
  let first = arr.shift();
  let last = arr.pop();

  let str = first + last;
  result.push(str);
}
```

```
result = result.join('-');
console.log(result);
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
16-25-34

[Done] exited with code=0 in 0.034 seconds

JAVASCRIPTDA FUNKSIYALAR

```
// funksiya yaratamiza
function func() {
  console.log("qale nima gap , JS ni o'rganvossanmi?")
}

func();
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
qale nima gap , JS ni o'rganvossanmi?

[Done] exited with code=0 in 0.036 seconds

```
//parametrlari bor funksiya yaratamiza
function salomlashish(ism, familiya) {
  console.log(ism, "qalesiz ", "maniyam familiyam", familiya, " tanishganimdan hursandman")
}
salomlashish("sardor", "Dadaboyev")
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
sardor qalesiz maniyam familiyam Dadaboyev tanishganimdan hursandman

[Done] exited with code=0 in 0.034 seconds

```
//funksiya ichida funksiya
function func(num) {
  return num * num;
}

let result = func(func(2));
console.log(result)
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
16

[Done] exited with code=0 in 0.036 seconds

```
//qidiruv metodlardan : map
let arr = [1, 2, 3, 4, 5];

let result = arr.map(function(elem) {
  return elem * elem;
});

console.log(result)
```

[1, 4, 9, 16, 25]

[Done] exited with code=0 in 0.039 seconds

JAVASCRIPTDA DESTRUKTURIZATSIYA

```
//destrukturizatsiya
let arr = [2025, 12, 31];
let [year, month, day] = arr;
console.log(year);
console.log(month);
console.log(day)
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.039 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
2025
12
31

[Done] exited with code=0 in 0.08 seconds

```
//destrukturizatsiya bilan oddiy holatni faqrlaymiza
let arr = [2025, 12, 31];

let year = arr[0];
let month = arr[1];
let day = arr[2];

console.log(year);
console.log(month);
console.log(day);
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.061 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
2025
12
31

[Done] exited with code=0 in 0.05 seconds

```
//destrukturizatsiya ohirgi elemnt undefined chiqadi chunki u massivda yo'q
let arr = [2025, 12];
let [year, month, day] = arr;

console.log(year);
console.log(month);
console.log(day);
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.05 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"
2025
12
undefined

[Done] exited with code=0 in 0.065 seconds

Javascriptda Date obyekti bilan ishlash

```
//Date obyekti bilan ishlash
let date = new Date();

console.log("yil:",date.getFullYear()); // yil
console.log("oy:", date.getMonth());    // oy
console.log("kun:",date.getDate());      // kun

console.log("soat: ",date.getHours());   // soat
console.log("minut:",date.getMinutes()); // minut
console.log("daqiqqa: ",date.getSeconds()); //daqiqqa
```

OUTPUT DEBUG CONSOLE TERMINAL PROBLEMS

[Done] exited with code=0 in 0.081 seconds

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"

yil: 2022

oy: 0

kun: 10

soat: 15

minut: 31

daqiqqa: 16

```
//Date obyekti bilan ishlash
```

```
let date = new Date();
```

```
console.log([date.getFullYear() + '-' + date.getMonth() + '-' + date.getDate()])
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"

2022-0-10

[Done] exited with code=0 in 0.065 seconds

```
//Date obyekti bilan ishlash
```

```
let date = new Date();
```

```
console.log(date.getFullYear() + '-' + (date.getMonth() + 1) + '-' + date.getDate());
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"

2022-1-10

[Done] exited with code=0 in 0.036 seconds

```
//Date obyekti bilan ishlash
```

```
function addZero(num) {
```

```
  if (num >= 0 && num <= 9) {
```

```
    return '0' + num;
```

```
  } else {
```

```
    return num;
```

```
  }
```

```
}
```

```
let date = new Date();
```

```
console.log(addZero(date.getFullYear()) + '-' + addZero(date.getMonth() + 1) + '-' + addZero(date.getDate()));
```

[Running] node "/home/sardor/Desktop/projects/DoskinMarket/server/practice.js"

2022-01-10

[Done] exited with code=0 in 0.078 seconds

NODEJS- haqida

JAVASCRIPT da umuman olganda imkoniyatlar juda ko'p
Misol uchun Browser bilan ishlash , web sahifaga o'zgartirish
kiritish, uni qayta ishlash.

Oldinlari javascript fayllar bilan ishlash, server yaratish
imkoniyati yoq edi , lekin

Node.js — veb-brauzerdan tashqarida JavaScript kodini
bajaradigan ochiq manbali, kross-platformali, JavaScript
runtime muhiti (inglizcha: JavaScript runtime environment).
Node.js 2009-yil 27-mayda Rayan Dal tomonidan yartilingan .
C++, C va JavaScriptda yozilgan bo'lib, asosan server uchun
foydalaniladi.



Rasm-15 Rayan Dal NodeJs asoschisi

1996 yilda Netscape kompaniyasi Server JavaScript-ni
(Server — yon JavaScript-SSJS) yaratishga urinishgan,
ammo texnologiya tarqalmagan.

Ryan dahl, Node yaratuvchisi.js

Node.JS Server veb-komponentlarini yaratish bo'yicha ikki
yillik tajribadan so'ng 2009 yilda Ryan Dal tomonidan ishlab
chiqilgan. Tadqiqot davomida, u an'anaviy parallelizm modeli
o'rniga, oqimlarga asoslangan holda, voqea yo'naltirilgan
tizimlarga murojaat qilish kerak degan xulosaga keldi. Ushbu
model soddaligi, past yuk ("har bir ulanish uchun bitta oqim"
mafkurasi bilan solishtirganda) va tezligi tufayli tanlangan.
Node ning maqsadi "keng ko'lamli tarmoq serverlarini
qurishning oddiy usuli"ni taklif qilishdir.

Javascript da NodeJS yordamida Server yaratish

```
// faqat nodejs o'zidan server yaratish|
import http from "http";

// so'rovlarni qabul qiladigan server tuzamiza
const server = http.createServer((incomingMessage, response) => {
  console.log("so'rovlar boshlandi");
  // javob kodi va http ni yuborramiz
  response.writeHead(200, { "Content-Type": "text/plain; charset=UTF-8" });
  response.end("Hello, world!");
});

// web serverni ihga tushiramiz
server.listen(80, "127.0.0.1", () => {
  const info = server.address();
  console.log(`serverning yoqilgan porti: ${info}`);
});
```

EXPRESS NODEJS NI FREYMWORNI HAQIDA

Express, NodeJS serverining ochilishini sezilarli darajada soddalashtirish, sayt tartibini qismlarga ajratish, shablonlashtirish, yo'l-yo'riq berish imkonini beruvchi freymwork hisoblanadi. Odatda, hozirgi vaqtda Node JS bilan ishlash Express orqali amalga oshiriladi. Keling buni ko'rib chiqamiza



Rasm-16 nodejs va express logotiplari

EXPRESS DA SERVER HOSIL QILAMIZ

O'zimning yaqinda qilayotgan proyektimni ko'rsatib o'taman

```
server > main.js > ...
1  const express = require('express')
2  const mongoose = require('mongoose')
3  const PORT = process.env.PORT || 3000
4  const product = require('./routers/prodRoute/product')
5  const productImage = require('./routers/prodRoute/uploadImage')
6  const path = require('path')
7  const router = require('./routers/jwt/jwtRoute')
8  const cors = require('cors')
9  const cookieParser = require('cookie-parser')
10 const app = express()
11
12 const statistics = require('./routers/statistics/statistics')
13
14 app.use(express.json({extended: true}))
15
16 app.use('/images', express.static(path.join(__dirname, 'images')))
17
18 app.use('/market', product) // jwt registratsiya login va hk va product crud uchun
19 app.use('/product', productImage) // rasm uchun
20 app.use('/api', router) // Ulbi Tv ni JWT si uchun
21 app.use('/statistics', statistics)
22 app.use(cookieParser())
23 app.use(cors())
24 const start = async() => {
25   try{
26     await mongoose.connect('mongodb://localhost:27017/DoskinMarket')
27     console.log("database is ready for work")
28     app.listen(PORT, () => console.log(`server is ready on port ${PORT}`))
29   }catch(e){
30     console.log({
31       message: "произошла ошибка ",
32       reason: e,
33       from1: "main.js",
34       from2: "start func"
35     })
36   }
37 }
38
39 start()
```

Bu asosiy server fayli hisoblanadi , hamma routlar funksiyalar , Middleware lar shunga ulanadi , va biz shu faylni ishga tushiramiz

Umuman olganda NodeJS dan foydalanish uchun biz NODEJS Internetdan yuklab olib o'rnatishimiz kerak, va NPM (NODE paketlar menedjeri) ni ham yuklab olamimza , npm bizga Kerakli intrumentlarni yuklab olishimizni imkonini beradi Hammasini o'tqazib olgandan keyhin biz terminalni ochamiza va papka yaratamiza va papkani ichida biz server faylini asosiy faylni yaratamiza , undan keyin biz shu komandalarni kiritishimiz kerak bo'lad, NODEJS o'rnatilgan yoki yo'q buni bilish uchun Biz node --version deb komanda kiritamiza , agar nodejs ni versiya nomdelari chiqsa demak biz nodejsni o'rnatgan bo'lamiza Undan keyin biz npm init deb nodejs proyektimizni initsializatsiya Qilamiz, undak keyin bizning proyektimizda node modules va package json degan papka fayl paydo bo'ladi Tabrikliman proyektin tayyorlash ishi boshlandi :)

MERN- STACK

Umuman hozirgi kunda shunday terminlarni uchratishimiz mumkin , MERN STACK, MEAN STACK bu nima degani MERN stack bu quyidagi texnologiyalarning boshlang'ich nomlaridan olingan stek hisoblanadi, ya'ni

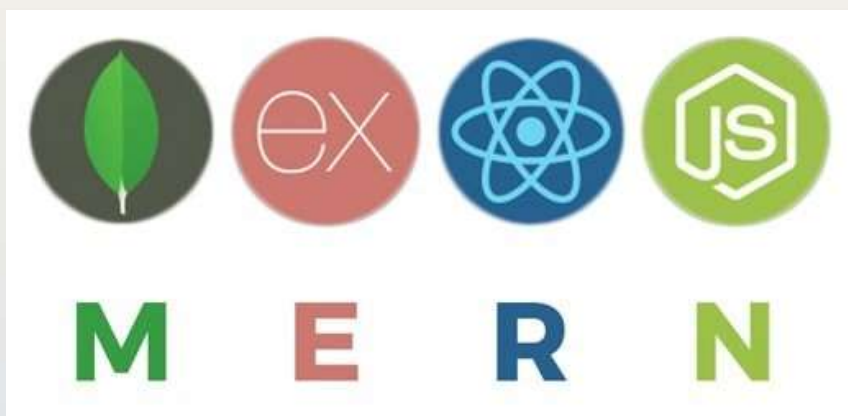


M- **MongoDb** (ma'lumotlar ombori NoSQL)

E- **Expressjs** (nodejs kutubxona)

R- **Reactjs** (Javascript kutubxona)

N - **NodeJS** (Javascript uchun platforma)



Rasm-17 MERN STACK

Hozirgi kunda mern stack yordamida biz katta saytlarni, web Platformalarni , CRM sistemalarni, va hk larni yaratishimiz mumkin Keling endi bular haqida batafsil gaplashamiza



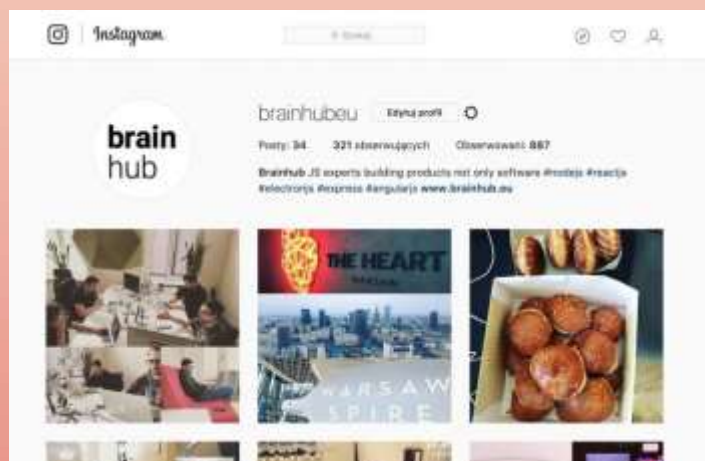
Rasm-18 tavsiya qiladigan kitoblarimdan biri

REACT JS

React js nima - JavaScript-maxsus interfeyslarni yaratish uchun kutubxona , NodeJs bizga javascript yordamiza saytning backend qismini yani server qismini yaratish uchun yordam bergan bo'lsa React bizga saytning klient qismini yaratish uchun juda katta imkoniyatlarni beradi

Nega aynan React , hozirgi kunda reactdan tashqari Angular, Vue degan kutubxonalar ham mavjud , reakt bularning eng mashxuri hisoblanadi

Chunki quyidagi proyektlar aynan reaktga qilingan Facebook, Instagram , Netflix, New York Times va boshqalar, bular web saytlar hisoblanadi ,



Rams – 19 instagram web sayti



Rams – 20 netflix web sayti



Rams – 21 New York Times web sayti

MERN STACK DA API CHIQARAMIZ

Buning uchun bizga kerakli narsalar :

1. NODEJS
2. NPM
3. EXPRESSJS
4. POSTMAN,
5. MONGODB,
6. KERAKLI INSTRUMENTLAR

POSTMAN BU so'rovlarni amalga oshirish uchun kerak bo'ladigan dastur hisoblanadi , juda qulay tarzda so'rovlar yuborishimiz mumkin,

MongoDB – bu esa NOSQL turidagi ma'lumotlar ombori hisoblanadi,

Hozirgi proyektda men mongoDB ga ulanishni o'rgatmayman , balki chunki manda o'rnatilib bo'lingan , internetdan qanday qilib o'rnatish kerakligi va qanday ishlatish kerakligi haqida ma'lumotlar juda ko'p , usha yerdan o'rganib olasiz degan umiddaman.

Hozirgi projekt nima haqida bo'ladi:

Bu projekt internet magazin hisoblanib unda biz ma'lumotlar omboriga (MB) yangi turdagi mahsulotlarni qo'shamiza

MB ni mahsulotlarga to'ldiramiza , va aynan shu mahsulotlarimiz bilan ishlash uchun har xil API larni yozamiz ,

Bular asosan REST API deb nomlanadi , va biz bu dasturimizda CRUD amallarini bajaramiza

CRUD bu:

C: CREATE (MB ga yangi narsa qo'shish)

R: READ (MB dan yaratilgan narsani o'qish ko'rish)

U: UPDATE (Mbdagi yaratilgan narsani o'zgartirish, uni yangilash)

D: DELETE (MB dagi narsani o'chirish, olib tashlash)

Uchun hizmat qiladigan amallar to'plami hisoblanadi

Hozir shulardan ba'zilarini ko'rib o'tamiza



Productni MB ga joylash funksiyasi

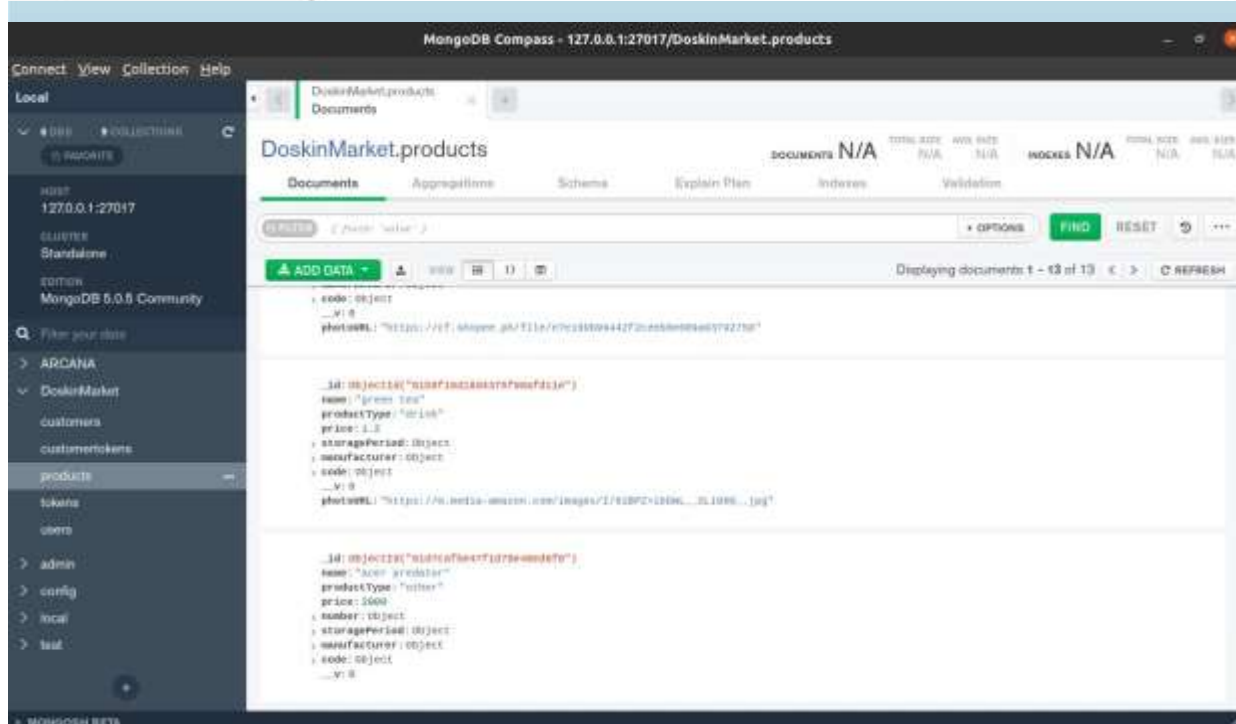
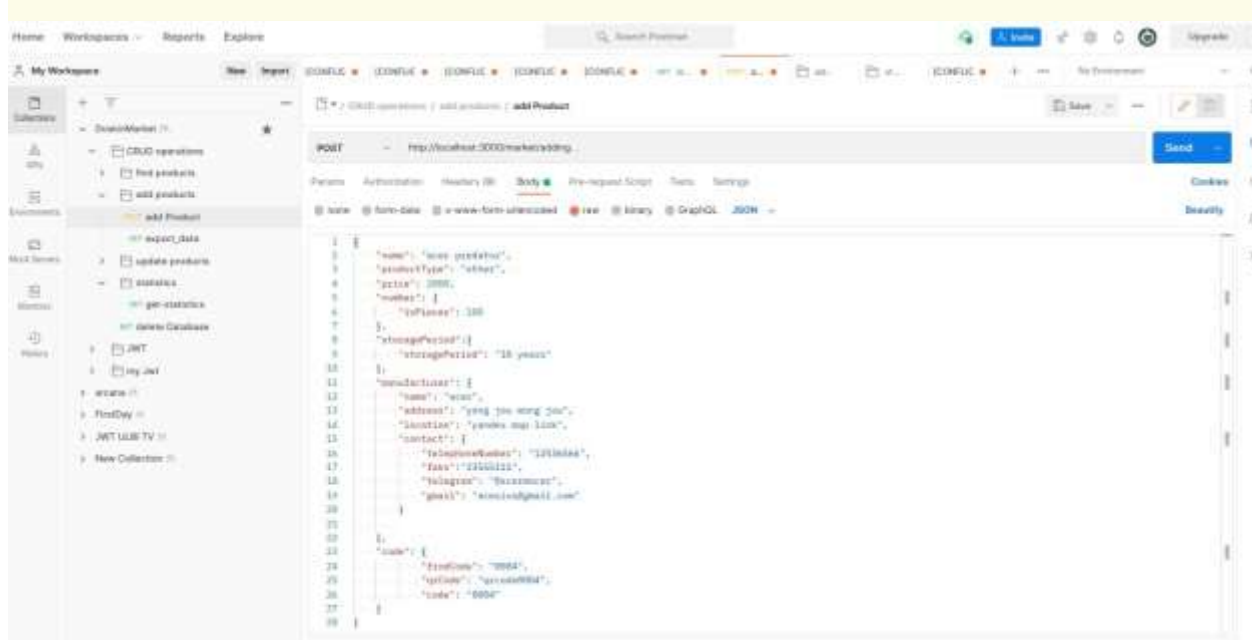
```
async addProduct ( req , res ) {  
  try {  
    const product = {  
      name: req.body.name,  
      price: req.body.price,  
      number: req.body.number,  
      productType: req.body.productType,  
      storagePeriod: req.body.storagePeriod,  
      manufacturer: req.body.manufacturer,  
      code: req.body.code  
    }  
  
    try {  
      // const type = {  
      //   value: req.body.type  
      // }  
      // await Types.create(type)  
  
      const productlar = await Products.create(product)  
      console.log(productlar)  
      res.send("added")  
    } catch (e) {  
      console.log({  
        message: "произошла ошибка ",  
        reason: e,  
        reason: e.message,  
        from1: "2",  
        from2: "addProduct"  
      })  
      res.send({  
        message: "произошла ошибка ",  
        reason: e,  
        reason: e.message,  
        from1: "2",  
        from2: "addProduct"  
      })  
      console.log(e)  
    }  
  }  
}
```

```
product.post('/adding', addProduct, (req,res) => {  
  })
```

```
const {addProduct} = require("../controllers/product/prod-control")
```

Hozir biza yaratilgan funksiyani eksport qilib asosiy faylda import qilamiza va import qilishdan oldin routerimizni ulab olamiza

Endi biz postmanni ochamiza va yangi mahsulotni bazaga joylaymiza

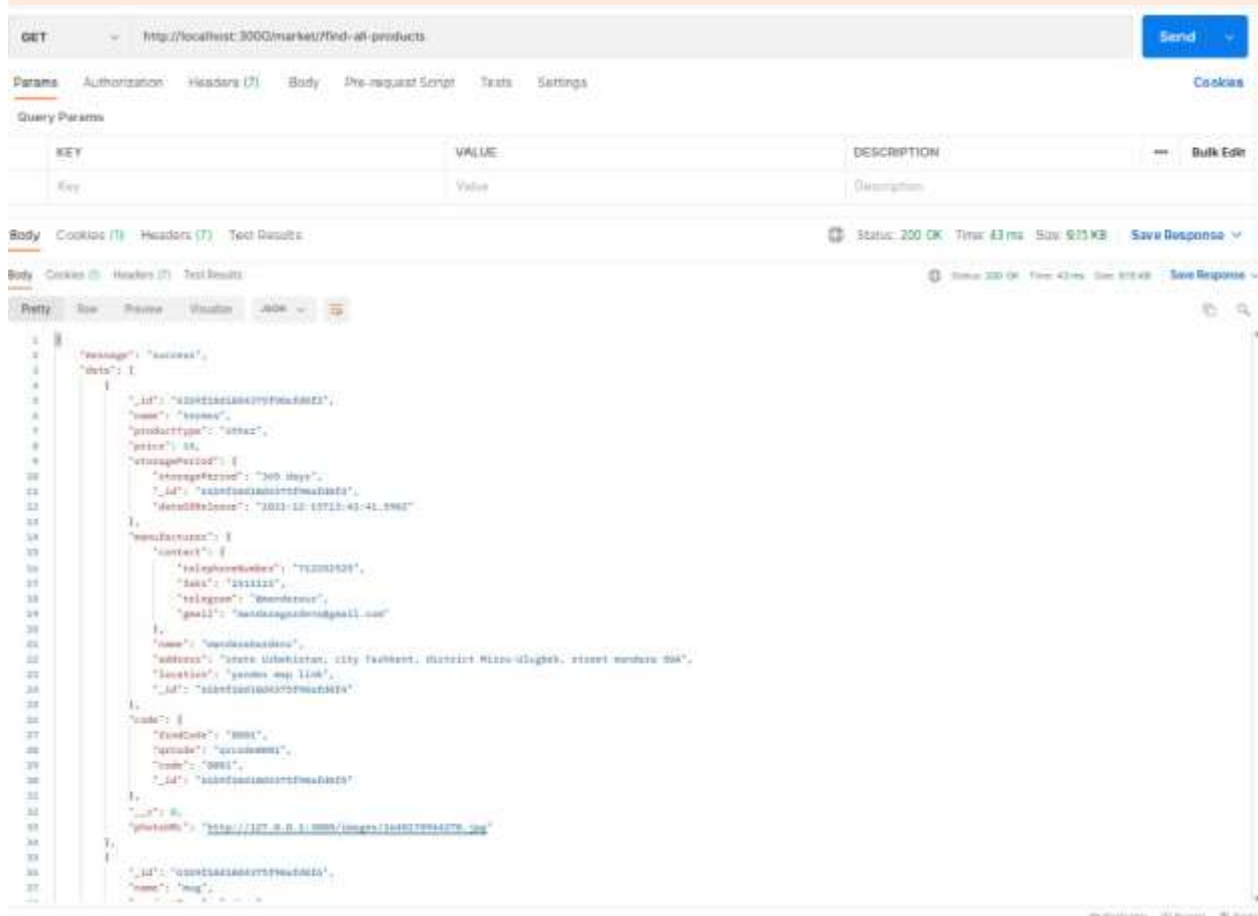


Mana Ma'lumotlar omboriga mahsulot joylandi. shunday qilib bir nechta turdagi mahsulotlarni bazaga kiritamiza , mahsulotimizning bazada: nomi, qanday turdagi mahsulot ekanligi uning ishlab chiqaruvchisi, ishlab chiqaruvchining manzili, kontakt nomerlari , email adreslari, telegram adresi undan tashqari mahsulotning qancha ekanligi , misol uchun 100 ta mahsulot , mahsulotning narxi , uning bazaga joylangan sanasi, kodlari(qr kodi, qidiruv kodi, va hk lar bo'lishi kerak, endi biz bazaga ko'p mahsulotlarni joylaganimizdan so'ng, ularni: o'qish, qidirish, sortirovka qilish , tartiblash, bazadan o'chirish, va hk larni bajaruvchi routlarni yozib chiqamiz biz bazamizdan foydalanish uchun avvalam bor foydalanuvchini autentifikatsiya qilishimiz lozim , chunki ma'lum foydalanuvchilar faqat shu bazadan foydalana olishi kerak , bu bizning dasturimizni havfsiz qilinishiga yordam beradi , hohlagan odam kirib bizning baza mizdan foydalanishini oldini olishimiz kerak hozir endi biz bazadan ma'lumotlarni ko'radigan skript yozamiz, va uni ko'ramiz.

```

async findProduct (req , res ) {
  try {
    let data = await Products.find()
    res.send({
      message: "success",
      data: data
    })
  } catch (e) { //3
    console.log({
      message: "произошла ошибка ",
      reason: e,
      from1: "3",
      from2: "findProduct"
    })
  }
}
}

```



Ko'rib turganingizdek biz hozir routimiz bo'yicha serverimizga postman orqali so'rov yubordik , va server bizning so'rovimizni qabul qildi va ma'lumotlar omboridan bizga kerakli narsalarni berdi ya'ni bizning bazamizdagi barcha mahsulotlarni chiqarib berdi , natija success bo'ldi va javob sifatida 200 status kodi keldi , bu degani hammasi yaxshi , hech qanday hatolik kuzatilmadi,

Endi biz bazadan o'zimizga kerakli bo'lgan mahsulotlarni chiqaramiz va narxi bo'yicha chiqaramiz, misol uchun bizning mahsulotimizning tipi mevalar bo'lsin , va bizga narxi 4\$ dan kam bo'lgan mevalarni chiqarib berish kerakligi talab qilindi, bunday vaziyatda biz script yozamiz , bizga kerakli so'rovni bajaradigan funksionallikni tuzamiz , va u quyidagichadir:


```

357 async sortByPrice (req, res) {
358   try {
359     const {price, productCategory} = req.body
360     // const price = req.body.price
361     // const productCategory = req.body.productCategory
362     if(price==0){
363       res.send({
364         message: 'oooo nu oldman, ${price} is an impossible price , please correct your price',
365         code: 1
366       })
367     }else if(price<0) {
368
369       try {
370         if(productCategory=='all'){
371           const allProducts = await Products.find({price: {$lte: price}}).select({name: 1, _id:0, price:1, "manufacturer.name": 1})
372
373           const mapsData = allProducts.map((product) => ({
374             name: product.name,
375             price: product.price,
376             manufacturerName: product.manufacturer.name
377           }))
378
379           res.send({
380             status: "success",
381             data: mapsData
382           })
383         }else{
384           const productsFromDB = await Products.find({ productType: productCategory, price: { $lte: price } }).select({'manufacturer.name':1, name:1,
385             if(productsFromDB.length!=0){
386
387               // const mapsData = productsFromDB.map((product) => {
388               //   const data = {
389               //     name: product.name,
390               //     manufacturerName: product.manufacturer.name
391               //   }
392               //   return data
393             // }
394
395             const mapsData = productsFromDB.map((product) => ({
396               name: product.name,
397               price: product.price,
398               manufacturerName: product.manufacturer.name
399             })))
400
401             res.send({
402               status: "success",
403               data: mapsData
404             })
405           } else{
406             res.send({
407               message: " there is not --${productCategory}-- category in our Market",
408               code: 2
409             })
410           }
411         }
412       } catch(e){
413         console.log(e)
414       }
415     }else{
416       res.send({
417         message: "price must be only a number {}".
418         code: 3
419       })
420     }
421   }
422   catch (e) {
423     console.log({
424       error: e,
425       message: e.message,
426       place: "sortByPrice"
427     })
428   }
429 }

```

`const {sortByPrice} = require('../controllers/product/prod-control')`

```

product.post('/sort-by-price', sortByPrice, (req,res) =>{
})

```

Buni yanayam yaxshiroq varianti query parametr orqali

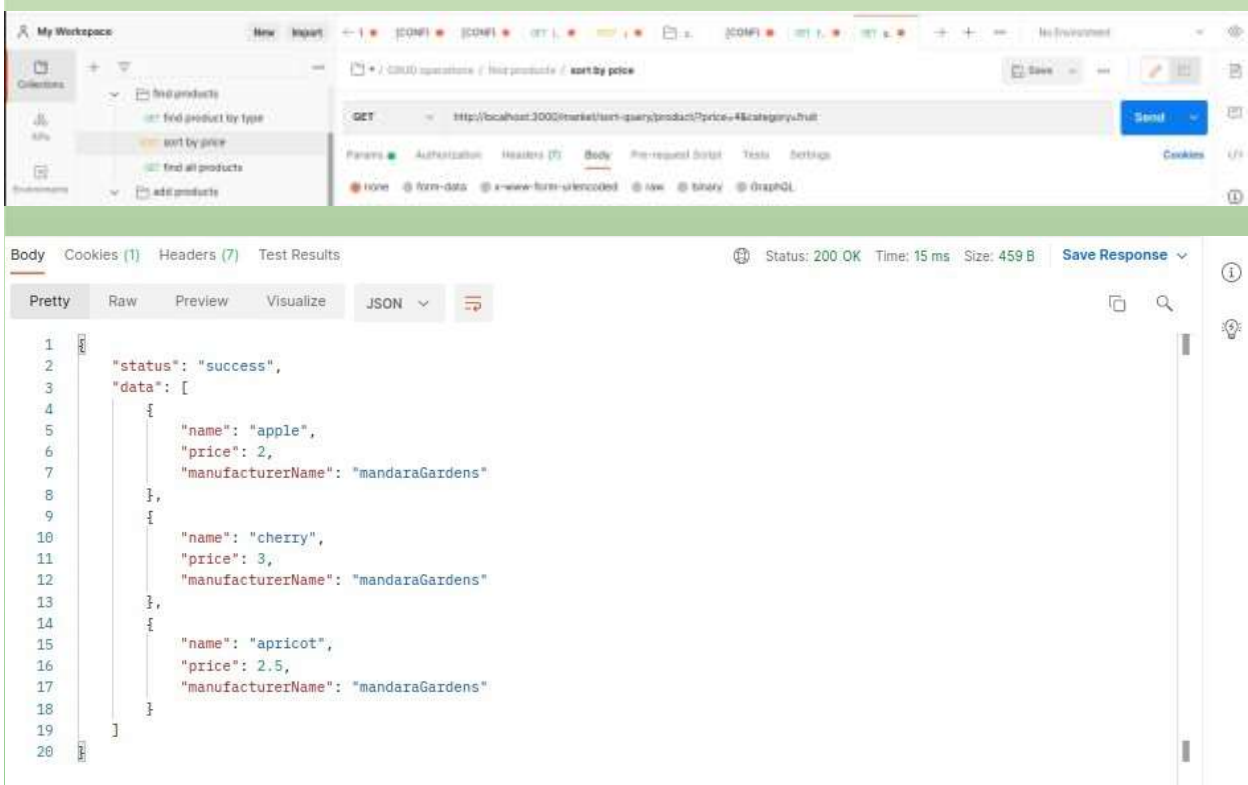
```

431 async sortByQuery (req, res) {
432   try {
433     const {price, category} = req.query
434     // const price = req.body.price
435     // const productCategory = req.body.productCategory
436     if(price==0){
437       res.send({
438         message: 'oooo nu oldman, ${price} is an impossible price , please correct your price',
439         code: 1
440       })
441     }else if(price<0) {
442
443       try {
444         if(category=='all'){
445           const allProducts = await Products.find({price: {$lte: price}}).select({name: 1, _id:0, price:1, "manufacturer.name": 1})
446
447           const mapsData = allProducts.map((product) => ({
448             name: product.name,
449             price: product.price,
450             manufacturerName: product.manufacturer.name
451           }))
452
453           res.send({
454             status: "success",
455             data: mapsData
456           })
457         }else{
458           const productsFromDB = await Products.find({ productType: category, price: { $lte: price } }).select({'manufacturer.name':1, name:1, _id:0, price: 1})
459           if(productsFromDB.length!=0){
460
461             // const mapsData = productsFromDB.map((product) => {
462             //   const data = {
463             //     name: product.name,
464             //     manufacturerName: product.manufacturer.name
465             //   }
466             //   return data
467             // }
468
469             const mapsData = productsFromDB.map((product) => ({
470               name: product.name,
471               price: product.price,
472               manufacturerName: product.manufacturer.name
473             })))
474           }
475         }
476       } catch(e){
477         console.log(e)
478       }
479     }else{
480       res.send({
481         message: "price must be only a number {}".
482         code: 3
483       })
484     }
485   }
486   catch (e) {
487     console.log({
488       error: e,
489       message: e.message,
490       place: "sortByQuery"
491     })
492   }
493 }

```

```
1871         status: "success",
1872         data: responseData
1873       }
1874     }
1875     if (err) {
1876       res.send({
1877         message: "there is not --<category>-- category in our Market",
1878         code: 2
1879       })
1880     }
1881   } catch (err) {
1882     console.log(err)
1883   }
1884 }
1885
1886 res.send({
1887   message: "price must be only a number (!)",
1888   code: 3
1889 })
1890 }
1891 } catch (e) {
1892   console.log(e)
1893   error: e,
1894   message: e.message,
1895   place: "errorHydria"
1896 }
1897 }
1898
1899 const objectToJson = new productController()
1900 module.exports = objectToJson
```

Endi postmandan so'rov yuboramiz



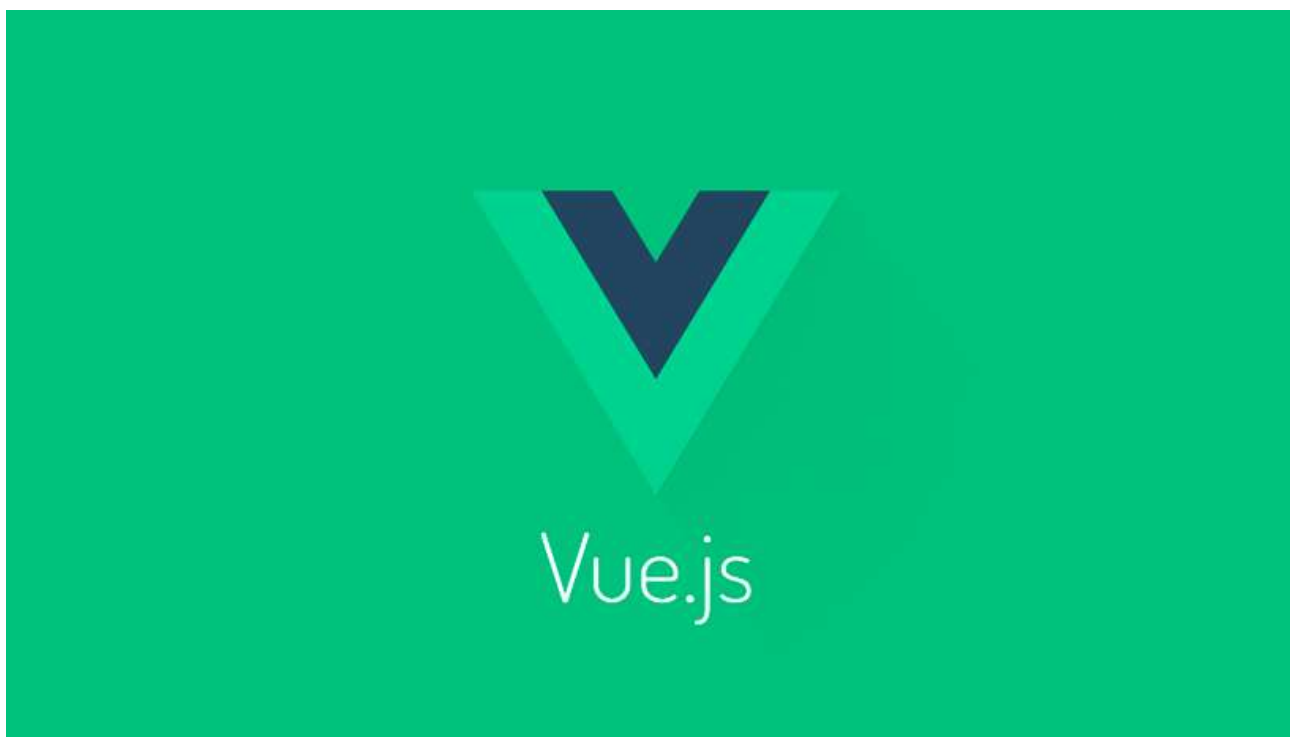
Ko'rib turganingizdek biz so'rovimizga query parametr orqali narxi 4\$ va categoriyasi fruit (mevalar) ni so'rov qildik va bizga 3 ta mahsulot keldi ular

Apple – 2\$, olma Cherry – 3\$, olcha

Aprikot- 2.5\$ o'rik

Biz bu routimizga frontendda turib narxi va kategoriyasi bo'yicha qidirish degan tugmamizga link(shu routni linki) ni berib qo'yamiz , shunda foydalanuvchi shu tugmani bosganda unga bazadan mahsulotlar keladi , va u ularni ko'ra oladi, Hulas shunaqa do'stlar , har hil turdagi routlarni yozishs mumkin , aytib o'tganimizdek CRUD operatsiyalarini bajaradigan va h.k.

VUEJS FRAMEWORKKI



Yuqorida biz ReactJS haqida gaplashgan edik , hozir esa biz VUEJS haqida uning imkoniyatlari nima ekanligini bilib o'tamiz

Evan You tomonidan yaratilgan Vue.js - foydalanuvchi interfeyslari (interfeyslari) va bitta sahifali dasturlarni yaratish uchun ochiq manbali progressiv JavaScript ramkasi; u odatda Vue deb nomlanadi. Ushbu framework ishlab chiquvchilarga foydalanuvchi interfeyslarini (UI) bosqichma-bosqich yaratishga imkon beradigan "yuqori ajratish" dan foydalanadi.

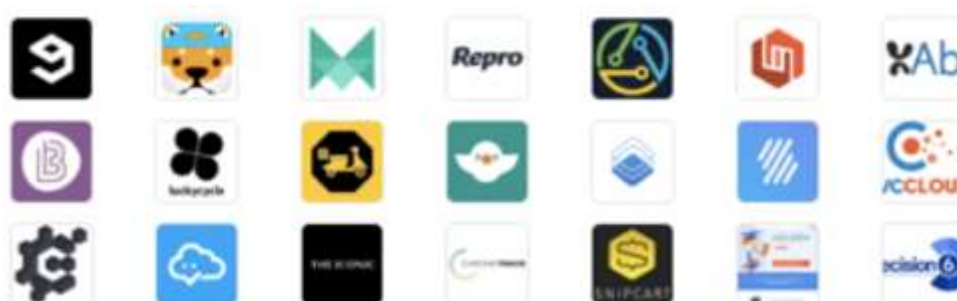
Frameworkdan foydalangan holda kutubxonani modullashtirish frontend rivojlanishida keng tarqalgan. React va Angular modulizatsiyaga ega. Ammo Vue.js-ni boshqa alternatalardan ajratib turadigan narsa - bu "yuqori ajratish", funktsional imkoniyatlarni kengaytirish qanchalik oson va modullar kiritilgandan so'ng barcha qismlarning ishlash darajasi. Masalan, biz kichik vizual komponentlarni tashkil qilishni va ko'rsatishni istasak, bizga Vue.js-ning "asosiy" kutubxonasi kerak; qo'shimcha kutubxonalarni kiritish kerak emas. Ilova o'sib borishi bilan bizda 'vue-router' kabi marshrutlarni boshqarish uchun kutubxonalar, 'vuex' kabi global holatni boshqarish uchun kutubxonalar yoki 'bootstrap-vue' kabi javob beradigan veb-illovalarni yaratish uchun kutubxonalar mavjud. Bundan tashqari, agar bizning dasturimiz optimallashtirilishi kerak yoki yaxshi SEO kerak bo'lsa, biz "vue-server-rendering" kutubxonasini o'z ichiga olamiz.

Vue.js komponentlari tizimi reaktiv, ya'ni Vue.js asenkron hodisalar orqali qanday aloqa qilishni biladi; masalan, bola komponenti ota-ona komponenti bilan voqealar orqali bog'lanishi mumkin. Vue.js-da boshqa kutubxonalar yoki manbalar bilan ishqalanish bo'lmaydi, boshqacha qilib aytganda, biz eng qulay bo'lgan vositadan foydalanishimiz mumkin. Masalan, biz faqat HTML va JavaScript-ni yozishimiz mumkin yoki agar xohlasak CSS, JSX yoki TypeScript-ni qo'shishimiz mumkin.

Vue.js-da JS tugunida yaratilgan maxsus buyruq satri (CLI) mavjud. Ushbu vosita bizga qozon plitasi (yoki asosiy shablon) yordamida loyihani boshlashga imkon beradi. Bundan tashqari, Vue.js ishlab chiquvchi guruhi bizning kengaytmamiz daraxtining qanday ko'rsatilishini, voqealar qanday boshlanishi va yozilishini, har bir komponentning ichki holati qanday saqlanishini va komponentning global holatini ko'rishimizga imkon beradigan Chrome kengaytmasini saqlaydi. o'zini tutmoqda.

VUEJS dan kimlar foydalanadi

700 dan ortiq kompaniyalar Vue.js.-dan foydalanmoqda. Ulardan ba'zilari: Xiaomi, Alibaba va Gitlab.



VUEJS da proyektni tuzamiz

Buning uchun biz rasmiy vuejs.org rasmiy web saytiga kirib bir necha usuldan o'zimizga qulayini tanlab o'tqazib olamiz :

Terminaga o'tib shu komandani yozamiz :



```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/powershell)

PS D:\VS CODE\arcabase\arcabase> npm install vue@next
```

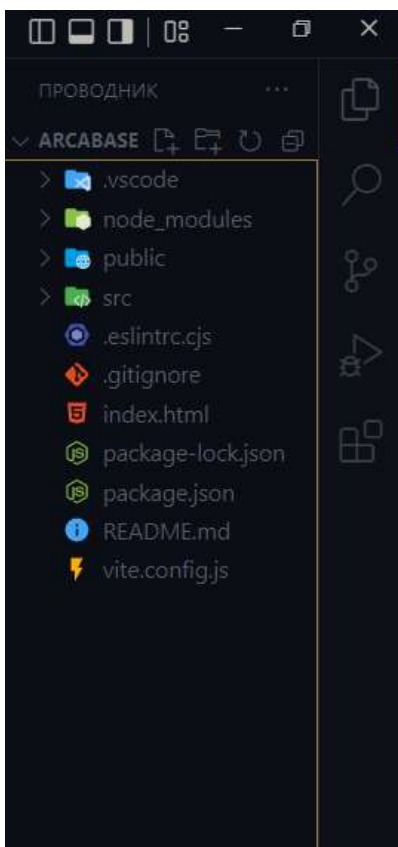
npm run dev orqali ishga tushiramiz



```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/powershell)

PS D:\VS CODE\arcabase\arcabase> npm run dev
```



Shunday qilib bizda projekt shakillanadi

```
App.vue src index.html x App.vue ...style
src > style > index.html > html > body > img
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Document</title>
8   </head>
9   <body>
10    
11  </body>
12 </html>
13
```

```
App.vue src x index.html App.vue ...style
src > App.vue > {} "App.vue" > template > div.header > div.wrapper > div.block > div.wrap
1 <template>
2   <div class="header">
3     <div class="head"><h2>INDIVIDUAL LOYIHA</h2></div>
4     <div class="wrapper">
5       <div class="block">
6         <div class="wrap">
7           <div class="img img1"></div>
8         </div>
9         <h2>POSITANO, ITALY</h2>
10      </div>
11      <div class="block">
12        <div class="wrap">
13          <div class="img img2"></div>
14        </div>
15        <h2>VENICE, ITALY</h2>
16      </div>
17      <div class="block">
18        <div class="wrap">
19          <div class="img img3"></div>
20        </div>
21        <h2>TASHKENT, UZBEKISTAN</h2>
22      </div>
23      <div class="block">
24        <div class="wrap">
25          <div class="img img4"></div>
26        </div>
27        <h2>MEKKA</h2>
28      </div>
29    </div>
30  </div>
31 </template>
32 <script>
33 export default {
34   data() {
35     return {}
36   },
37   methods: {},
38 }
39 import './style/App.vue'
40 </script>
```



```

src > style > App.vue > .img
1 <style>
2 * {
3   padding: 0;
4   margin: 0;
5 }
6
7 .head {
8   width: 100%;
9   background-color: aquamarine;
10  display: flex;
11  justify-content: center;
12  align-items: center;
13  padding: 20px 0;
14  letter-spacing: 2px;
15  font-weight: 700;
16 }
17
18 .wrapper {
19   margin-top: 50px;
20   display: flex;
21   justify-content: space-around;
22
23   padding: 30px 0;
24 }
25 .block {
26   display: flex;
27   flex-direction: column;
28   justify-content: center;
29   padding: 25px 20px;
30   align-items: center;
31   font-weight: 600;
32   font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande',
33     'Lucida Sans', Arial, sans-serif;
34   box-shadow: 10px 10px 29px 2px aquamarine;
35 }
36 .wrap {
37   display: inline-block;
38   overflow: hidden;
39   width: 350px;
40   height: 250px;

```

```

41 }
42 .img {
43   width: 350px;
44   height: 250px;
45   transition: transform 0.8s;
46   margin-bottom: 10px;
47   display: block;
48 }
49 .img1 {
50   background: url('../img/positano.jpg') no-repeat center / cover;
51 }
52 .img2 {
53   background: url('../img/venetsia.jpg') no-repeat center / cover;
54 }
55 .img3 {
56   background: url('../img/tashkent.jpg') no-repeat center / cover;
57 }
58 .img4 {
59   background: url('../img/mekka2.jpg') no-repeat center / cover;
60 }
61
62 .img:hover {
63   transform: scale(1.3);
64 }
65 </style>
66

```

Proyekt tuzizm



Shunday qilib aziz do'stlar biz bugun javascript va uning imkoniyatlari degan loyihani ko'rib chiqdik , bunda biz ko'p ma'lumotlarga ega bo'ldik , lekin bu ma'lumotlar , dengizning bir qismi holos , hali ma'lumotlar juda ko'p , shu bilan birga bizni ishimizni osonlashtiradigan texnologiyalar ham juda ko'p, Javascript yilla davomida rivojlanib keldi , hozirgi kunda biron-bir Web saytni javascriptsiz tasavvur ham qilib bo'lmaydi, hozirgi kunda javascript ko'plab sohalarni qamrab oldi, Shu jumladan server hosil qilish, (NodeJS ExpressJS) , bu haqda ozgina gaplashdik, web saytning frontend qismi uchun kutubxonalar (React, Angular, Vue) , Javascriptda hozirgi kunda Kompyuterlarga dasturlar tuzish imkoniyati ham bor u ElectronJS orqali amalga oshadi (siz biladigan VS Code kod muharriri , aynan ElectronJS da yozilgan , undan tashqari Javascript mobil dasturlar tuzish sohasida ham sobitqadamlardan hisoblanadi, mobil telefonlarga dastur yozish uchun javascriptda ReactNative freymworkidan foydalaniladi, qo'lingizdagi telefondagi instagram Ijtimoiy tarmoqi ham aynan ReactNative yordamida yozilgan , Undan tashqari Javascript yordamida telegram uchun botlar tuzish mumkin bu uchun esa TelegrafJS yordamga keladi, Zamon rivojlangani sayin Javascript Machine Learning sohasiga ham kirib keldi bu uchun esa TensorFlow.Js yordamga keladi, Ko'rib turibsiz Javascript dasturchining yo'llari juda ko'p , asosiysi esa ularni yaxshilab o'rganish zarur. Katta raxmat.

FOYDALANILGAN RESURLAR RO'YXATI

1. <http://code.mu>
2. <https://www.tensorflow.org/js>
3. <https://developer.mozilla.org/en-US/>
4. <https://expressjs.com/ru/>
5. <https://translate.yandex.ru/>
6. <https://ru.reactjs.org/>
7. <https://learn.javascript.ru/>
8. <https://www.pngrepo.com/>
9. <https://docs.mongodb.com/>
10. <https://www.google.com/>