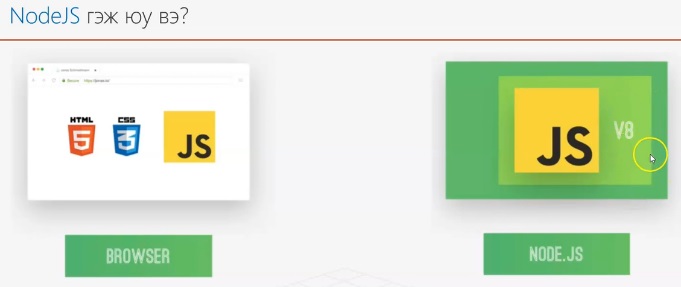
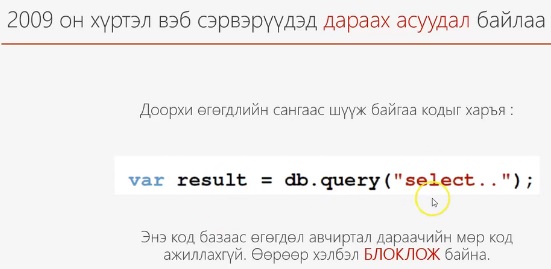
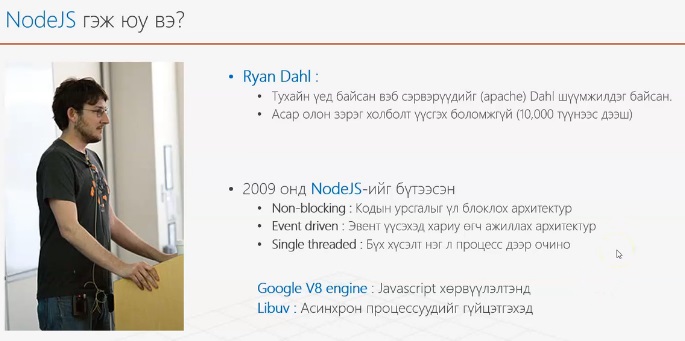
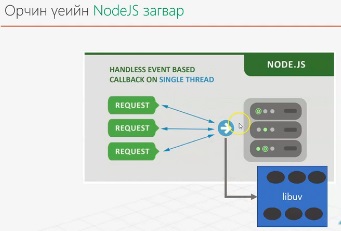
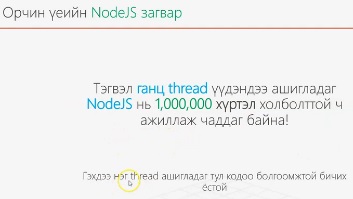
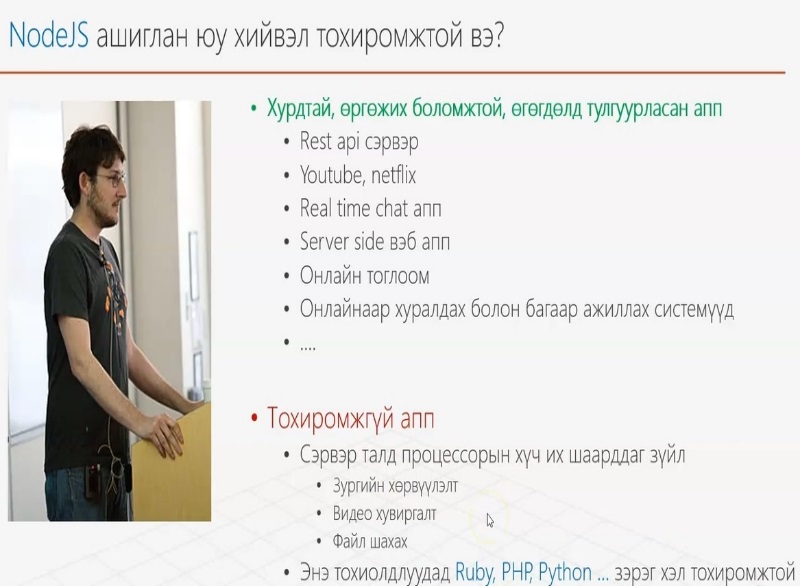
**NodeJS**



NodeJS нь javascript-ыг уншихдаа google гаргасан v8 engine хэрэглэдэг. Deloper-ууд сангууд бичиж эхэлсэн. Тэгээд тэр сангуудаа нэгтээд нэг газарааас авдаг болсон. Тэр нь PACKEGE MANAGER буюу npm ингэж үүссээд ганцхан Node JS labrary биш болоод Javarscrip-ын бүх сан болсон юм.





**Өөрийнхөө файлын системрүү хандах**

Computer дээрээ test гэсэн folder үүсгээд VS code-оор ороод test.js файл үүсгэ.

Console.log(“Hello..”)

Terminal гаргаж ирээд bash тerminal-аараа node test.js гэж өгөхөд Hello.. хэвлэгдэнэ.

Файлын системрүү хандах

const fs = require("fs");

fs.writeFileSync("./config.txt",

"файлын системрүү хандах."

);

bash тerminal-аараа node test.js гэж өгөхөд файлын системрүү хандах.

хэвлэгдэнэ.

const fs- = require("fs"); file sytem-ын labrary хэрэглэхийн тулд эхлээд заавалчгүй require Хийнэ. Reqiure гэдэг бол javascript-ын гаднаас labrary-ын module –аа оруулж ирэх syntax нь байж байна.

ES6 дээр import … from “…” гэж бичпэг. Харин node дээр хуучнаараа байгаа.

const fs = require("fs") Энэ болохоороо fs сангаас fs нэртэйгээр оруулж ирий гэж байна. Ер нь devolper-ууд нэрийг адилхан өгчихдөг. Энэд ямар зам гэх мэт зүйл заагаагүй бол core labrary-гаас эсвэл npm-ээр суулгасан labrary-гаас авч байна. FS – File system гэсэн үг.

fs.writeFileSync("./config.txt", "файлын системрүү хандаж ");

fs.writeFileSync - Ямар ситем ашиглах гэж байгаа бичнэ.

("./config.txt", - Замаа зааж өгөөд нэрийн бичиж өгнө.

"файлын системрүү хандаж " - Юу бичих гэж байгаагаа зааж өгнө.

**Интернетэд байгаа url уншаад доторх content-ыг өгдөг бэлэн labrary болох request ашиглан файл татах авах.**

const fs = require("fs");

const request = require("request");

request(“URL”, (error, response, body) => {

    fs.writeFileSync("./folder-ын нэр хаана хадгалах ", body);

})

Бүх html файлыг json гэж нэрэлбэл Json файлаар гарч ирнэ. VS code гэх мэт editor nodejs доторх document-уудыг өөр дотороо нэгтгэхэд шууд уншиж татаж а авдаг.

Жишээ нь ямар нэг doc хуудсыг нь татаж авчираад file болгоод хадгалаад авчихий.

<https://nodejs.org/dist/latest-v14.x/docs/api/cluster.json>

Интернетэд байгаа URL уншаад доторх content-ыг өгдөг бэлэн labrary байж таараа. Зөндөө байдаг. Broser дотор

Fetch api

Axios api

Request гэх мэт гуравдагч labrary

Гуравдагч labrary VS code дээрээ depenses ашиглахын тулд npm init –y гэдэг

Npm init гээд бичих юм бол төрөл бүрийн асуулт асуугаад байдаг. –y бүх асуултын алгасаад defualt-аар энэ коммандыг ажилуулдаг. Энгээд өгөнгүүт package.json гэсэн folder VS code дотор сууна.

Npm install гэдэг нь гуравдагч labrary – гаас суулгана.

npm install request –-save request сууллаа.

**REQUEST АШИГЛААД DOCUMENT ТАТАХ:**

const fs = require("fs");

const request = require("request");

request('https://nodejs.org/dist/latest-v14.x/docs/api/cluster.json', (error, response, body) => {

    fs.writeFileSync("./cluster.json", body);

})

----------------------------------------------------------------------

const fs- = require("fs"); file sytem-ын labrary хэрэглэхийн тулд эхлээд заавалчгүй require Хийнэ. Reqiure гэдэг бол javascript-ын гаднаас labrary-ын module –аа оруулж ирэх syntax нь байж байна.

Энэ болохоороо fs сангаас fs нэртэйгээр оруулж ирий гэж байна. Ер нь devolper-ууд нэрийг адилхан өгчихдөг. Энэд ямар зам гэх мэт зүйл заагаагүй бол core labrary-гаас эсвэл npm-ээр суулгасан labrary-гаас авч байна. FS – File system гэсэн үг.

const request = require("request"); - Request-ын объект зарлах

fs.writeFileSync - Ямар ситем ашиглах гэж байгаа бичнэ.

("./config.txt", - Замаа зааж өгөөд нэрийн бичиж өгнө.

"файлын системрүү хандаж " - Юу бичих гэж байгаагаа зааж өгнө.

Server үүсгэх

Http гэсэн folder үүсгээд VS code editor нээ. Дотор нь server.js файл үүсгэ.

Terminal bash ажиллуулаад npm init –y Ингэснээр depencese файл суулгах боломжтой болно.

const http = require('http');

const hostname = '127.0.0.1';

const port = 3000;

const server = http.createServer((req, res) => {

res.statusCode = 200;

res.setHeader('Content-Type', 'text/plain');

res.end('Hello, World!\n');

});

server.listen(port, hostname, () => {

console.log(`Server running at http://${hostname}:${port}/`);

});

const server = http.createServer(() => {}); энэ бол сэрвэр үүсгэхүү гээд үүсгэж байна. Функц дамжуулах ёстой. Цаанаас бидний вэб сэрвэрүү хандах юм бол энэ функцыг дууддаг

Хүсэлтэнд хариулахдаа 2 аргумент орж ирнэ.

Request = req = Broseroo нээж байгаад манай вэб сэрвэрүү хандах юм бол миний broser-оос ирсэн http request message байна.

Responce = res = Request-эд юу гэж хариулахыгаа бичнэ. Энэд бичсэн миний brosert ирнэ.

server.listen(port, hostname, () => {

console.log(`Server running at http://${hostname}:${port}/`);

}); Сэрвэрээ ямар портон дээр ажиллуулахаа бичиж өгдөг. 0 –> 64000 хооронд ажилладаг байж байгаа. Стандарт вэб бол 80 –Р порт дээр ажилладаг. Development хийж байхад ямар порон дээр ажилуулж өолно. Гэхдээ 1024 оос доош нь ихэвчлэн үйлдлийн системийн стандарт болчихсон байдаг. Тусгай зөвшөөрөл шаарддаг.

listen() – Ямар портон дээр сонгох функц

Listen(5000, () = {}) - Дотор нь байгаа функц сэрвэр асах болгонд ажилдаг функц.

const http = require('http');

const server = http.createServer((req, res) => {

  const {headers, url, method} = req;

  console.log(`header =====>`, headers);

  console.log(`url =====> ${url}`);

  console.log(`method =====> ${method}`);

res.setHeader('Content-Type', 'text/plain');

res.write("<h1> welcome to NODEJS hello!!</h1>");

res.end();

server.listen(5000,  () => {

  console.log("http 5000 aslaa");

});

Хүсэлт явуулангуут 2 хариулт ирнэ. Нэг нь favorite icon хүсэлт юм. Favirate icon явуул гэсэн.

URL бичсэн нь GET

Хүсэлтээр явуулсан POST

const http = require("http");

const server = http.createServer((req, res) => {

const { headers, url, method } = req;

res.setHeader("content-type", "text/html");

if (url === "/") {

res.statusCode = 200;

res.write("<h1>Manai delguurt tavtai moril</h1>");

res.write(`<br><br>Login hiihiin tuld <a href="/login">END DARNA UU</a> `);

res.end();

} else if (url === "/login") {

// Login form html butsaana

res.statusCode = 200;

res.write("<h1>Login hiih</h1>");

res.write(`<form action="/logincheck" method="POST"> `);

res.write(`<br><input type="text" name="email" /> `);

res.write(`<br><input type="password" name="password" /> `);

res.write(`<br><input type="submit" value="Login" /> `);

res.write(`<form /> `);

res.end();

} else if (url === "/logincheck" && method === "POST") {

// login hiisnii daraa usreh heseg

res.statusCode = 200;

res.write("<h1>Login hiij uzlee...</h1>");

res.write("<br><h1>" + method + "</h1>");

res.end();

} else if (url === "/home") {

// login hiisnii daraa usreh heseg

} else {

res.statusCode = 404;

res.write("<h1>404 not found</h1>");

res.end();

}

});

server.listen(5000, () => {

console.log("http сэрвэр 5000 порт дээр аслаа...");

});

1. express

2. nodemon

3. mongoose

4. dotenv

5. cors