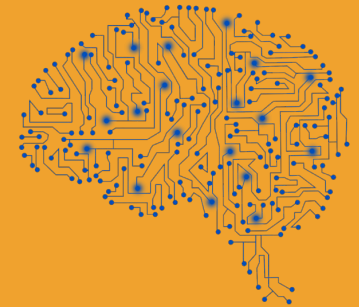


Εθνικό Μετσόβιο Πολυτεχνείο Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Διαδίκτυο και Εφαρμογές

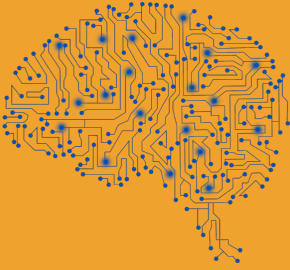


Κατασκευή εφαρμογής για πρόβλεψη της ανάγκης
εφαρμογής μέτρων απέναντι στον SARS-CoV-2 στις
πολιτείες της Αμερικής



Αχλάτης Στέφανος-Σταμάτης

1.Εισαγωγή: Γενική Εισαγωγή

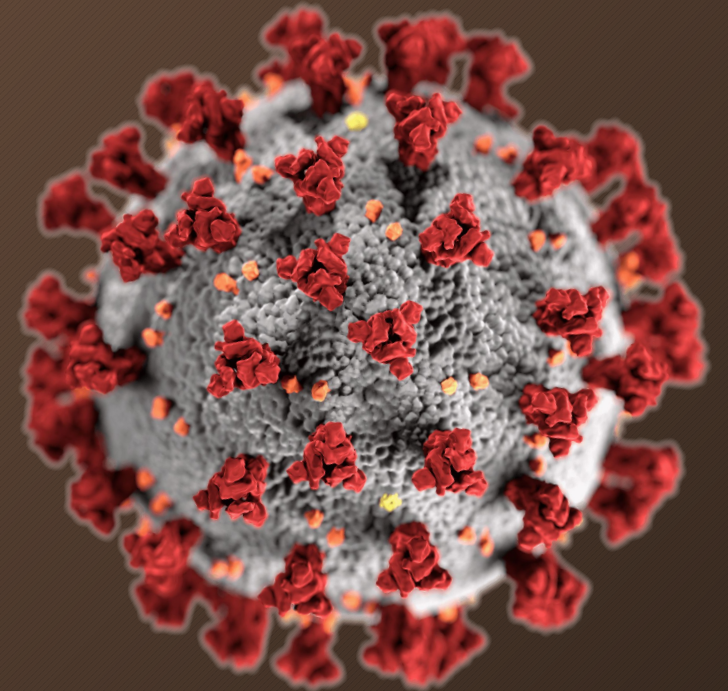


Οι κορονοϊοί που προσβάλλουν τον άνθρωπο προκαλούν συνήθως *ήπιες αναπνευστικές παθήσεις*

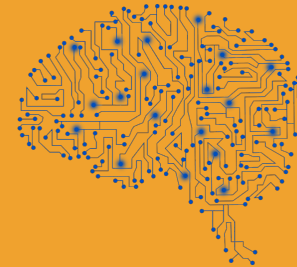
Τις τελευταίες δύο δεκαετίες δύο κορονοϊοί, ο SARS-CoV και ο MERS-CoV, προκάλεσαν *σοβαρή πνευμονία και θανάτους* σε ανθρώπους

Τον Δεκέμβριο του 2019 στην πόλη Wuhan της Κίνας εμφανίστηκε ο νέος κορονοϊός SARS-CoV-2 που προσβάλλει τον άνθρωπο.

Το τελευταίο διάστημα υπάρχουν φόβοι για δεύτερο κύμα της πανδημίας με ανάγκη χρήσης μέτρων κοινωνικής αποστασιοποίησης.

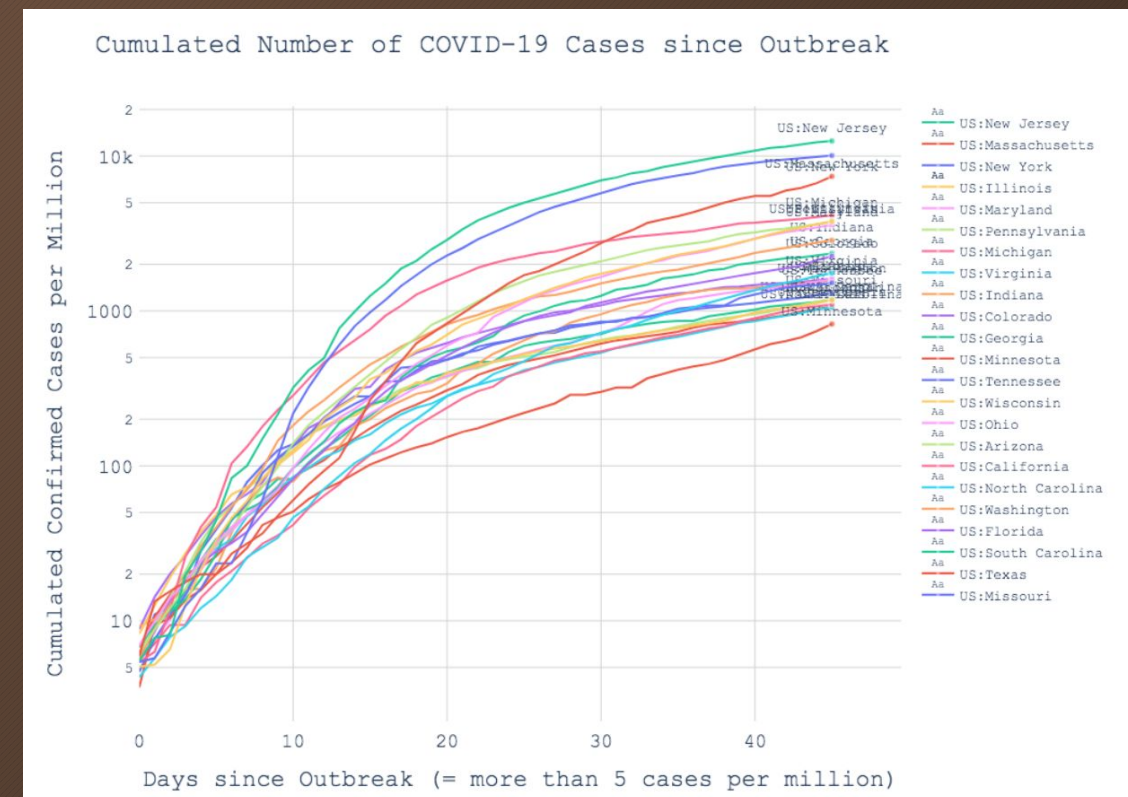


1.Εισαγωγή: Βασική Ιδέα

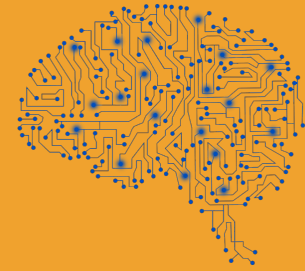


Αρχικά λαμβάνουμε υπόψη την κατηγοριοποίηση των χωρών με βάση την έως τώρα πορεία τους απέναντι στον ιό.

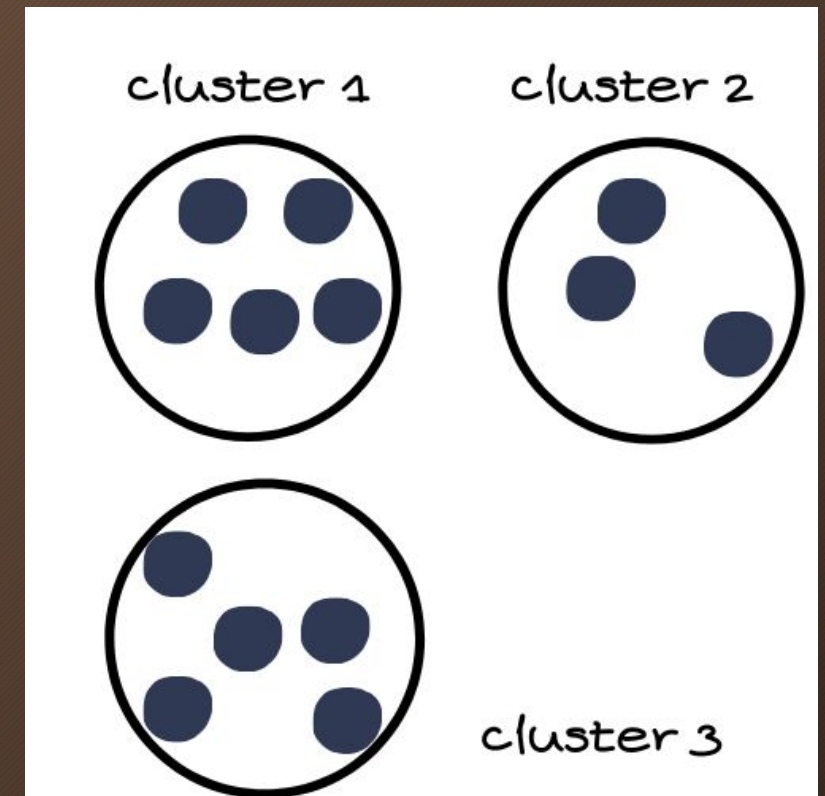
Η κατηγοριοποίηση αυτή λαμβάνεται από την εξής εργασία: “Ανάλυση παραγόντων επιδημιολογικού προφίλ χωρών σχετικά με την πανδημία του SARS-COV-2 με χρήση τεχνικών μηχανικής μάθησης”, Αχλάτης Στέφανος-Σταμάτης, Καπερώνη Φρειδερίκη, Μανιουδάκη Ευαγγελία



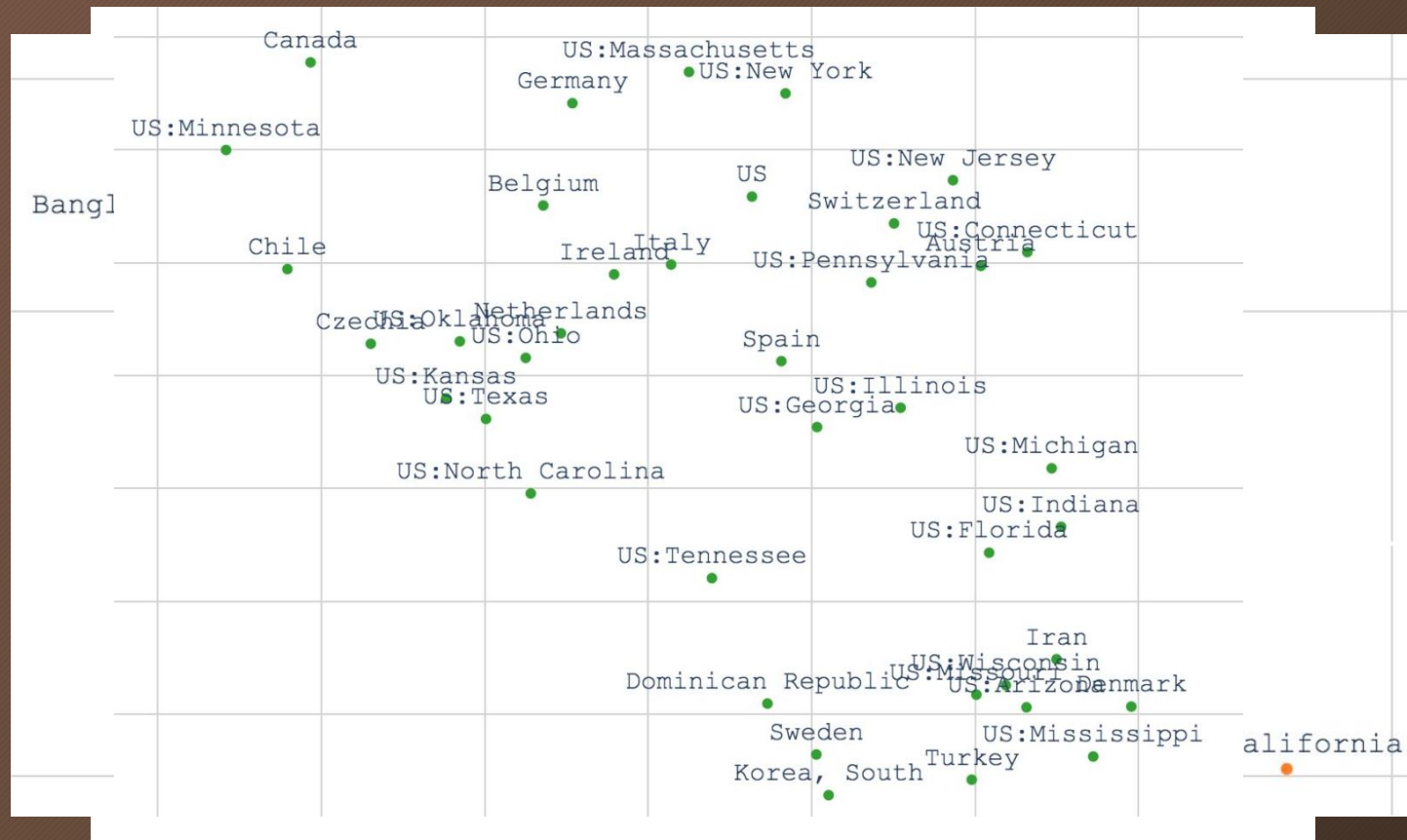
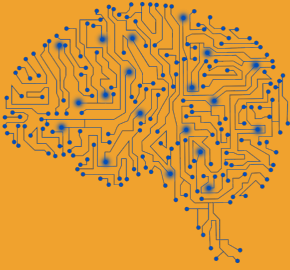
1.Εισαγωγή: Βασική Ιδέα



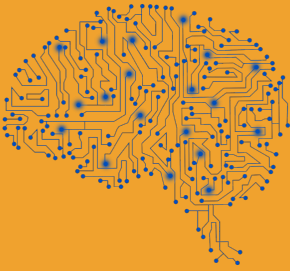
Ορισμένες χώρες και ορισμένες πολιτείες των ΗΠΑ κατηγοριοποιούνται σε τρεις κλάσεις, στατιστικά ανεξάρτητες. Στην κλάση 1 χαμηλού κινδύνου (δεν βρίσκουμε καμία πολιτεία των ΗΠΑ), στην κλάση 2 μετρίου κινδύνου (εδώ βρίσκουμε την Καλιφόρνια) και στην κλάση 3 υψηλού κινδύνου. Η αναλυτική ομαδοποίηση φαίνεται στην επόμενη σελίδα.



1.Εισαγωγή: Βασική Ιδέα



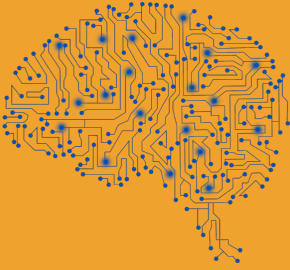
1.Εισαγωγή: Βασική Ιδέα



Στη συνέχεια βλέπουμε τα ημερήσια κρουσματα κορονοϊου κάθε πολιτείας και ανάλογα με το πόσα είναι και σε ποια κλάση ανήκει η πολιτεία προτείνουμε μετρα ή όχι. Αξίζει να σημειωθεί ότι αυτό το στάδιο δεν είχε κάποια επιστημονική ανάλυση από πίσω και χρησιμοποιεί μια ενδεικτική διαδικασία πρόβλεψης, που φαίνεται παρακάτω:

```
if (final_status == 2 && positiveIncrease > 7000)
document.getElementById("print1").innerHTML = "Πρέπει να εφαρμόσετε μέτρα!";
if (final_status == 2 && positiveIncrease < 7000)
document.getElementById("print1").innerHTML = "Δεν χρειάζεται να εφαρμοσετε μέτρα!";
if (final_status == 3 && positiveIncrease > 5500)
document.getElementById("print1").innerHTML = "Πρέπει να εφαρμόσετε μέτρα!";
if (final_status == 3 && positiveIncrease < 5500)
document.getElementById("print1").innerHTML = "Δεν χρειάζεται να εφαρμοσετε μέτρα!";
```


2.Εργαλεία και Τεχνολογίες: html

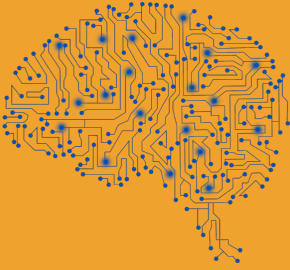


Η Hypertext Markup Language (HTML) είναι markup language για σχεδιαση αρχείων που θα προβληθούν σε έναν web browser.

Χρησιμοποιήθηκε στο Front-end



2.Εργαλεία και Τεχνολογίες: css

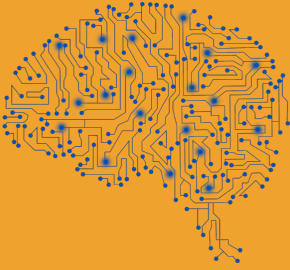


Η Cascading Style Sheets (CSS) είναι μια style sheet language που χρησιμοποιείται για να περιγράψει την παρουσίαση αρχείων γραμμένα σε μια markup language όπως η HTML.

Χρησιμοποιήθηκε στο Front-end



2.Εργαλεία και Τεχνολογίες: Javascript

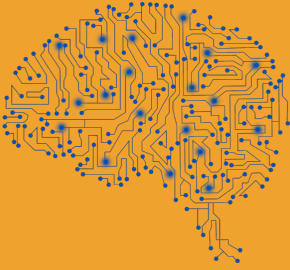


Η Javascript χρησιμοποιείται για να κάνει την σελίδα πιο διαδραστική. Χρησιμεύει στην επικοινωνία του client με το API που χρησιμοποιούμε και την Βάση Δεδομένων που κατασκευάσαμε

Χρησιμοποιήθηκε τόσο στο Front-end όσο και στο Back-end

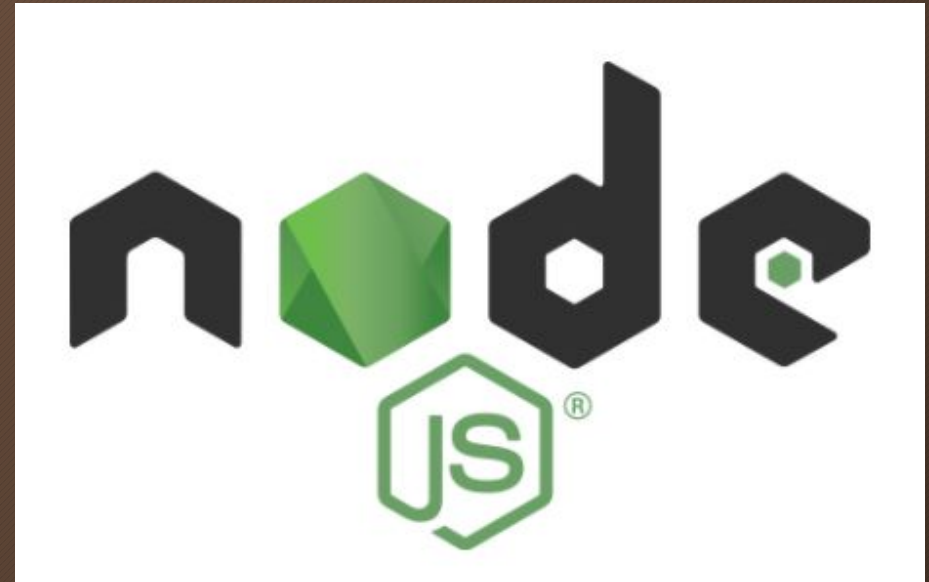
A large yellow square with the letters 'JS' in dark blue, representing the JavaScript logo.

2.Εργαλεία και Τεχνολογίες: Node.js

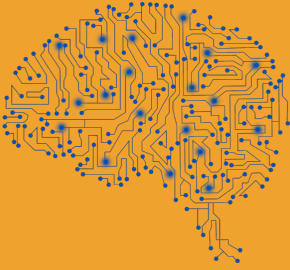


Είναι ένα server side runtime environment (framework) που τρέχει Javascript έξω από έναν web browser. Χτίστηκε πάνω στο chrome's V8 JavaScript engine.

Χρησιμοποιήθηκε στο Back-end.



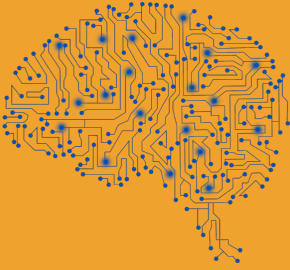
2.Εργαλεία και Τεχνολογίες: Express.js



Η Express.js είναι ένα web application framework για το Node.js και χρησιμοποιείται για την δημιουργία web application και API. Είναι το πιο διαδεδομένο server-side framework για την Node.js. Μπορούμε να πούμε ότι η Express.js είναι για τη Node.js ότι η Ruby on Rails ή η Sinatra για τη Ruby. Σαφώς, δεν συγκρίνεται με client-side frameworks όπως η React, Angular, Vue κλπ. Η Express.js σε βοηθάει να διαχειριστείς πολλές ενέργειες όπως τα routes, να διαχειριστείς requests και views.

The Express.js logo, which is the word "express" in a lowercase, sans-serif font, enclosed within a white rectangular box.

2.Εργαλεία και Τεχνολογίες: Npm

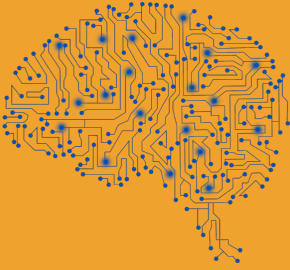


Το npm είναι ένας packet manager για την Javascript.

Χρησιμοποιήθηκε στο Back-end.



2.Εργαλεία και Τεχνολογίες: NeDB



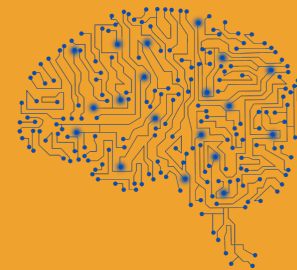
Είναι ένα σχετικά καινούργιο σύστημα διαχείρισης βάσεων δεδομένων. Είναι υποσύνολο της MongoDB και επομένως είναι μια βάση δεδομένων κατάλληλη για JavaScript.

Χρησιμοποιήθηκε στο Back-end.

Το github της είναι [εδώ](#).



3. Δεδομένα: API



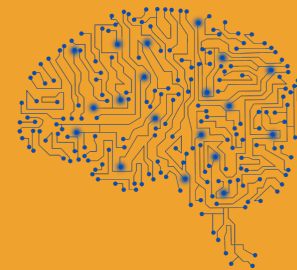
Επιλέξαμε να χρησιμοποιήσουμε το `api.covidtracking.com` που παρέχει δεδομένα για την πορεία του ιού στην κάθε πολιτεία για κάθε μέρα. Πιο συγκεκριμένα αν θέλαμε να μελετήσουμε την πορεία του ιού στην Καλιφόρνια στις 10/08/2020 θα ψάχναμε εδώ:

<https://api.covidtracking.com/v1/state/s/ca/20200810.json>

Είναι αντικείμενα JSON, για διευκόλυνση μας στη JS

```
{  
  "date": 20200810,  
  "state": "CA",  
  "positive": 561911,  
  "negative": 8436442,  
  "pending": null,  
  "hospitalizedCurrently": 6770,  
  "hospitalizedCumulative": null,  
  "inIcuCurrently": 1879,  
  "inIcuCumulative": null,  
  "onVentilatorCurrently": null,  
  "onVentilatorCumulative": null,  
  "recovered": null,  
  "dataQualityGrade": "B",  
}
```

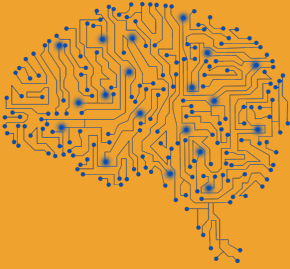

3. Δεδομένα: Βάση Δεδομένων



Δημιουργήθηκε από τα αποτελέσματα της προηγούμενης μελέτης. Ουσιαστικά αποτελείται από αντικείμενα json που περιέχουν πληροφορίες για το state που αναφέρεται το κάθε item και για το status του state, δηλαδή την κλάση στην οποία ανήκει.

```
1 {"state":"ca","status":"2","_id":"kP0JLK58wY5IojpN"}
2 {"state":"or","status":"2","_id":"gRI9KCY1TMYxvS9r"}
3 {"state":"mn","status":"3","_id":"xXI31LcMseakvfWi"}
4 {"state":"ma","status":"3","_id":"CsoIUsz8jFamp48M"}
5 {"state":"ny","status":"3","_id":"IKqyijW0QFVSfG7V"}
6 {"state":"nj","status":"3","_id":"1gphDX1p5yREhR46"}
7 {"state":"ct","status":"3","_id":"iXEEE0FnXuCdFaLI"}
8 {"state":"ok","status":"3","_id":"myeIbt0vg7dVMJq1"}
9 {"state":"ks","status":"3","_id":"HhyusTtldGioAmpM"}
```


4.Ανάλυση: Παράδειγμα Εφαρμογής



Επιλογή Πολιτείας(abbreviation):

ca

Επιλογή Ημερομηνίας(σε μορφή yyyyymmdd):

20200410

SUBMIT

Δεν χρειάζεται να εφαρμοσετε μέτρα!

Επιλογή Πολιτείας(abbreviation):

ca

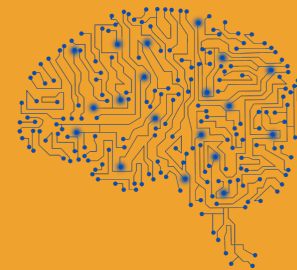
Επιλογή Ημερομηνίας(σε μορφή yyyyymmdd):

20200810

SUBMIT

Πρέπει να εφαρμόσετε μέτρα!

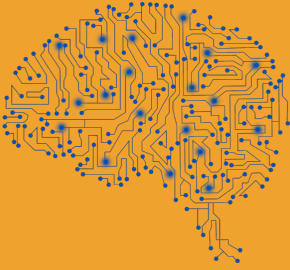
4:Ανάλυση: Η φόρμα σε html



```
<form id="form" >
  Επιλογή Πολιτείας(abbreviation): <input type="text" value="ca" id="country" name="country">
  Επιλογή Ημερομηνίας(σε μοργή yyyyymmdd): <input type="text" value="20200810" id="date" name="date">
    <div class="input-field col s12 m2">
      <button class="btn waves-effect waves-light" type="submit" id="EDW">
        Submit
      </button>
    </div>
</form>
```

Επομένως τη φόρμα την βλέπουμε ως μια οντότητα με αναγνωριστικό “form”.

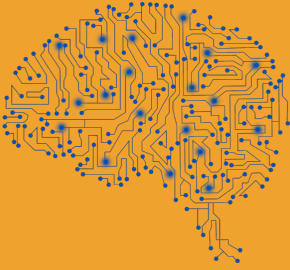
4:Ανάλυση: Event Listener



```
document.querySelector("form").addEventListener("click", function(event) {  
    var data = document.getElementById("form");  
    var api_url = 'https://api.covidtracking.com/v1/states/'+data.elements[0].value+'/'+data.elements[1].value+'.json';  
    event.preventDefault();  
});
```

Με το αναγνωριστικό form, περιμένουμε ο χρήστης να συμπληρώσει την φόρμα, όταν συμπληρωθεί η συνάρτηση γίνεται triggered και παίρνουμε τα δεδομένα από τα δυο κελιά και φτιάχνουμε δυναμικά το api url. Επίσης κανουμε preventDefault, για να μην γίνει reload η σελίδα

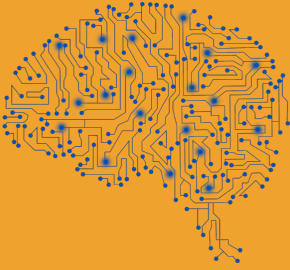
4:Ανάλυση: Δεδομένα από API



```
async function getAPI(){  
    const response = await fetch (api_url);  
    const data = await response.json();  
    const {state,positiveIncrease} = data;  
    getDatabase(positiveIncrease);  
}
```

Όταν ο event listener ακουσει το event, ζητάει από το api να πάρει τα δεδομένα σε μορφή json. Από αυτά τα αντικείμενα κρατάμε μόνο το state και το positiveIncrease. Μετά ζητάμε να μάθουμε σε ποια κλάση ανήκει η πολιτεία που κάνουμε ανάλυση.

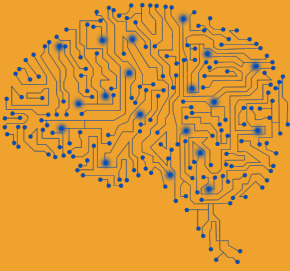
4:Ανάλυση: Δεδομένα από Server



```
async function getDatabase(positiveIncrease){  
    const response = await fetch ('/api');  
    const data = await response.json();  
}
```

Μετά παίρνουμε τα δεδομένα από τον server ακριβώς με ίδια διαδικασία, ωστόσο τώρα την διαδικασία get του server την έχουμε ορίσει αναλυτικά εμείς και επιστρέφει όλο το περιεχόμενο της Βάσης. Τα δεδομένα είναι json αντικείμενα.

4:Ανάλυση: Επεξεργασία δεδομένων



Έχοντας τα θετικά κρούσματα της ημέρας και το περιεχόμενο όλης της βάσης, θέλουμε να δούμε από όλο αυτό το περιεχόμενο ποιο αντιστοιχεί στην πολιτεία που μελετάμε. Αφότου το βρούμε κάνουμε συγκρίσεις με βάση την κλάση που ανήκει και το πληθος των κρουσμάτων και εξάγουμε συμπεράσματα.

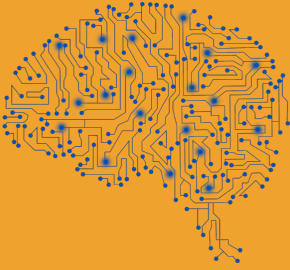
```
for (item of data){
  const root = document.createElement('div');
  const state = document.createElement('div');
  const status = document.createElement('div');

  var a = item.state;
  var lala = document.getElementById("form");
  var b = lala.elements[0].value;

  if (a.localeCompare(b) == 0 ){
    var final_status = item.status;
    console.log('Status is '+ item.status);
    if (final_status == 2 && positiveIncrease > 7000)
      document.getElementById("print1").innerHTML = "Πρέπει να εφαρμόσετε μέτρα!";
    if (final_status == 2 && positiveIncrease < 7000)
      document.getElementById("print1").innerHTML = "Δεν χρειάζεται να εφαρμοσετε μέτρα!";
    if (final_status == 3 && positiveIncrease > 5500)
      document.getElementById("print1").innerHTML = "Πρέπει να εφαρμόσετε μέτρα!";
    if (final_status == 3 && positiveIncrease < 5500)
      document.getElementById("print1").innerHTML = "Δεν χρειάζεται να εφαρμοσετε μέτρα!";

  }
}
```

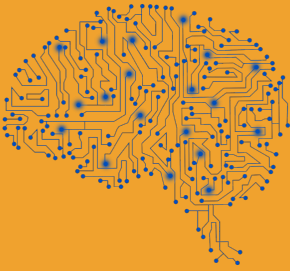

4:Ανάλυση: Back-end Αρχικοποίηση



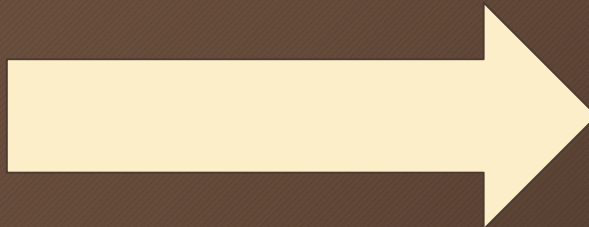
Τρέχουμε node.js και θα χρησιμοποιήσουμε το framework express και το σύστημα βάσης δεδομένων NeDB

```
const express = require('express');  
const Datastore = require('nedb');
```


4:Ανάλυση: Back-end κατασκευή βάσης

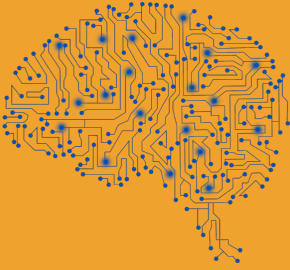


```
const database = new Datastore('database.db');
database.loadDatabase();
database.insert({state:'ca',status:'2'});
database.insert({state:'or',status:'2'});
database.insert({state:'mn',status:'3'});
database.insert({state:'ma',status:'3'});
database.insert({state:'ny',status:'3'});
database.insert({state:'nj',status:'3'});
database.insert({state:'ct',status:'3'});
database.insert({state:'ok',status:'3'});
database.insert({state:'ks',status:'3'});
database.insert({state:'tx',status:'3'});
database.insert({state:'nc',status:'3'});
database.insert({state:'il',status:'3'});
database.insert({state:'tx',status:'3'});
database.insert({state:'mi',status:'3'});
database.insert({state:'in',status:'3'});
database.insert({state:'fl',status:'3'});
database.insert({state:'az',status:'3'});
database.insert({state:'wi',status:'3'});
database.insert({state:'pa',status:'3'});
database.insert({state:'ga',status:'3'});
```



```
{"state":"ca","status":"2","_id":"kP0JLK58wY5IojpN"}
{"state":"or","status":"2","_id":"gRI9KCY1TMYxvS9r"}
{"state":"mn","status":"3","_id":"xXI31LcMseakvfWi"}
{"state":"ma","status":"3","_id":"CsoIUsz8jFamp48M"}
{"state":"ny","status":"3","_id":"IKqyijW0QFVSfG7V"}
{"state":"nj","status":"3","_id":"1gphDX1p5yREhR46"}
{"state":"ct","status":"3","_id":"iXEEE0FnXuCdfaLI"}
{"state":"ok","status":"3","_id":"myeIbt0vg7dVMJq1"}
{"state":"ks","status":"3","_id":"HhyusTtldGioAmpM"}
{"state":"tx","status":"3","_id":"JkteI3Aof0LaayA"}
{"state":"nc","status":"3","_id":"SJ5aFn5o05dgYFMo"}
{"state":"il","status":"3","_id":"JivjX8a26uF0fmAS"}
{"state":"tx","status":"3","_id":"YVWygIAAj7z0ga1P"}
{"state":"mi","status":"3","_id":"FwdAnvHntQEpQr6B"}
{"state":"in","status":"3","_id":"ksra5Gxzn4bwCSnq"}
{"state":"fl","status":"3","_id":"mhWyg3ncvkdGoC9L"}
{"state":"az","status":"3","_id":"du27MfzdJA5npz3w"}
{"state":"wi","status":"3","_id":"YkKTFdwfXjRShlMt"}
{"state":"pa","status":"3","_id":"YTyCdaHYstdioEQ1"}
{"state":"ga","status":"3","_id":"0UIuCHDV5Co6v6b0"}
```


4:Ανάλυση: Back-end Express



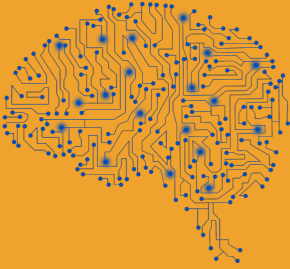
```
const app = express();  
app.listen(3000,()=>console.log('listening at 3000'));  
app.use(express.static('public'));
```

Επικοινωνία στην πόρτα 3000 και
αρχικοποιήσεις

Απάντηση του fetch, φέρνει όλα τα
δεδομένα από την βάση μέσω του
query που εξηγείται και στο
documentation του NeDB

```
app.get('/api', (request, response) => {  
  database.find({}, (err, data) => {  
    if(err){  
      response.end();  
      return;  
    }  
    response.json(data);  
  });  
});
```

5: Η εφαρμογή τρέχει!



```
(base) MacBook-Pro-tou-chreste-Stephanos:Internet-and-Applications-NTUA StefanosAchlati$ node index.js  
listening at 3000  
█
```

MetraforAll

Ανάλυση Σχετικά με εμάς Επικοινωνία

Μέτρα απέναντι στον SARS-CoV-2

Χρησιμοποιώντας ένα μοντέλο που χτίστηκε με Μηχανική Μάθηση, προβλέπουμε το κατά πόσον μια πολιτεία της Αμερικής πρέπει να λάβει μέτρα απέναντι στον SARS-CoV-2

ΜΑΘΕ ΠΕΡΙΣΣΟΤΕΡΑ ΓΙΑ ΤΟ ΜΟΝΤΕΛΟ ΜΑΣ!

Επιλογή Πολιτείας προς ανάλυση

Επιλογή Πολιτείας(abbreviation):
ca

Επιλογή Ημερομηνίας(σε μορφή yyyy-mm-dd):
20200810

Σας ευχαριστούμε πολύ!

