

**Μηχανικη Λογισμικου για Διαδικτυακες Εφαρμογες**



**Δημιουργια διαδικτυακης εφαρμογης με χρηση δεδομενων απο εξωτερικες πηγες**

Ομάδα Υλοποίησης του Project

Ηλιάδης Αλέξιος, Κόκκινου Ξανθή, Μήσιος Δημήτριος, Ταουκτσής Βασίλης

Περιεχόμενα

[Περιεχόμενα 1](#_Toc61011885)

[Η διαδικτυακή εφαρμογή 2](#_Toc61011886)

[Περιγραφή API 3](#_Toc61011887)

[Αρχική σελίδα 10](#_Toc61011888)

[Εγγραφή χρήστη(SignUp) 11](#_Toc61011889)

[Σύνδεση/Αποσύνδεση χρήστη(Login/Logout) 12](#_Toc61011890)

[Ανάκτηση κωδικού πρόσβασης 13](#_Toc61011891)

[Προβολή Events 14](#_Toc61011892)

[Βάση δεδομένων 16](#_Toc61011893)

[Αξιολόγηση σελίδας 17](#_Toc61011894)

# Η διαδικτυακή εφαρμογή

Το ***musiCity*** είναι μια διαδικτυακή εφαρμογή η οποία θα υλοποιηθεί στα πλαίσια της ομαδικής εργασίας στο μάθημα «Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές, του μεταπτυχιακού προγράμματος «Ευφυείς Τεχνολογίες Διαδικτύου» του Τμήματος Μηχανικών Πληροφορικής του Διεθνούς Πανεπιστημίου Ελλάδας.

Η εφαρμογής αυτή παρέχει στον χρήστη πληροφόρηση για μουσικά γεγονότα (event) που θα γίνονται σε μια πόλη. Στην υλοποίηση μας, η πόλη είναι προκαθορισμένη από τον προγραμματιστή και είναι η «Θεσσαλονίκη». Ο χρήστης έχει τις παρακάτω δυνατότητες:

* Να ενημερώνεται για μουσικά γεγονότα χωρίς να είναι εγγεγραμμένος.
* Να εγγραφεί στην υπηρεσία.
* Όταν συνδέεται σαν εγγεγραμμένος χρήστης.
* Όταν συνδέεται σαν εγγεγραμμένος χρήστης, έχει τη δυνατότητα να χαρακτηρίζει ή αποχαρακτηρίζει, εκδηλώσεις ως «αγαπημένες».
* Να ζητάει επανέκδοση κωδικού.

Η πορεία εξέλιξης του project φαίνεται στο περιβάλλον του GitHub στον παρακάτω σύνδεσμο: <https://github.com/iee-ihu-gr-coursem102/WebEng20_group9_CSearch>

Η ιστοσελίδα θα εμφανιστεί στην συνέχεια στο παρακάτω URL: <https://nireas.it.teithe.gr/webeng20g9>

Περιγραφή API

Λεπτομέρειες για το API της Songkick είναι διαθέσιμες στην ιστοσελίδα:

<https://www.songkick.com/developer/response-objects>

Για να γίνει χρήση του API πρέπει πρώτα να προηγηθεί αίτημα προς τον πάροχο με αντίστοιχη αιτιολόγηση, ώστε να αποκτηθεί το απαραίτητο «αναγνωριστικό κλειδί». Στην δική μας περίπτωση δόθηκε εκπαιδευτικό κλειδί διάρκειας τριών μηνών.

Το API υποστηρίζει τα παρακάτω αντικείμενα:

* Event object (Περιγραφή της εκδήλωσης)
* Performance object (Περιγραφή της παράστασης)
* Artist object (Περιγραφή καλλιτέχνη)
* Venue object (Περιγραφή του χώρου)
* Location object
* City object

Η κλήση για τον κωδικό μιας πόλης πχ της Θεσσαλονίκης γίνεται ως εξής:

https://api.songkick.com/api/3.0/search/locations.json?query=thessaloniki&apikey={your\_api\_key}

και επιστρέφει το παρακάτω αντικείμενο json, όπου ο κωδικός της πόης είναι το πεδίο *metroArea.id*

*{*

*"resultsPage": {*

*"status": "ok",*

*"results": {*

*"location": [{*

*"metroArea": {*

*"lat": 40.6403,*

*"lng": 22.9439,*

*"country": {*

*"displayName": "Greece"*

*},*

*"displayName": "Thessaloníki",*

*"uri": "http://www.songkick.com/metro\_areas/28999-greece-thessaloniki?utm\_source=59402&utm\_medium=partner",*

*"id": 28999*

*},*

*"city": {*

*"lat": 40.6403,*

*"lng": 22.9439,*

*"country": {*

*"displayName": "Greece"*

*},*

*"displayName": "Thessaloníki"*

*}*

*}]*

*},*

*"perPage": 50,*

*"page": 1,*

*"totalEntries": 1*

*}*

*}*

Πχ οι κωδικοί πόλεως της Αθήνας, Θεσσαλονίκης και Λονδίνου είναι:

*{*

*"athens": "4120",*

*"thessaloniki": "28999",*

*"london": "24426"*

*}*

Υλοποιήθηκε κλήση στο Event Object, το οποίο έχει τα παρακάτω πεδία:

| **Attribute** | **Type** | **Description** |
| --- | --- | --- |
| *id* | Integer | The Songkick ID of the event |
| *type* | String | The type of the event. 'Concert' or 'Festival' |
| *uri* | String | The URI of the event on Songkick |
| *displayName* | String | A textual representation of the event |
| *start* | Object | An object representing the start time for this event |
| *end* | Object | Festivals only. An object representing the end time for this event |
| *performance* | Array | An array of Performance objects, representing individual artist performances at the event |
| *location* | Object | An object representing the location of the event, in the form of a Songkick Location object |
| *venue* | Object | An object representing the event venue, in the form of a simplified Songkick Venue object |
| *status* | String | The status of the event. 'ok', 'cancelled' or 'postponed' |
| *ageRestriction* | String | The age restriction for the event, if known (nullable) |

Η κλήση στο Event Object γίνεται ως εξής:

https://api.songkick.com/api/3.0/metro\_areas/{city\_code}/calendar.json?apikey={api\_key}&page=1&per\_page =20

και επιστρέφει ένα json αντικείμενο με την παρακάτω δομή:

{

*"id":11129128,*

*"type":"Concert",*

*"uri":"http://www.songkick.com/concerts/11129128-wild-flag-at-fillmore?utm\_source=PARTNER\_ID&utm\_medium=partner",*

*"displayName":"Wild Flag at The Fillmore (April 18, 2012)",*

*"start": {*

*"time":"20:00:00",*

*"date":"2012-04-18",*

*"datetime":"2012-04-18T20:00:00-0800"*

*},*

*"performance": [*

*{*

*"artist":{*

*"uri":"http://www.songkick.com/artists/29835-wild-flag?utm\_source=PARTNER\_ID&utm\_medium=partner",*

*"displayName":"Wild Flag",*

*"id":29835,*

*"identifier":[]*

*},*

*"id":21579303,*

*"displayName":"Wild Flag",*

*"billingIndex":1,*

*"billing":"headline"*

*}*

*],*

*"location": {*

*"city":"San Francisco, CA, US",*

*"lng":-122.4332937,*

*"lat":37.7842398*

*},*

*"venue": {*

*"id":6239,*

*"displayName":"The Fillmore",*

*"uri":"http://www.songkick.com/venues/6239-fillmore?utm\_source=PARTNER\_ID&utm\_medium=partner",*

*"lng":-122.4332937,*

*"lat":37.7842398,*

*"metroArea": {*

*"uri":"http://www.songkick.com/metro-areas/26330-us-sf-bay-area?utm\_source=PARTNER\_ID&utm\_medium=partner",*

*"displayName":"SF Bay Area",*

*"country": { "displayName":"US" },*

*"id":26330,*

*"state": { "displayName":"CA" }*

*}*

*},*

*"status":"ok",*

*"popularity":0.012763*

*}*

Από τα επιστρεφόμενα θα χρησιμοποιήσουμε μόνο τα πεδία:

* Όνομα event με το αντίστοιχο Link
* Όνομα καλλιτέχνη με το αντίστοιχο Link
* Το όνομα του χώρου διεξαγωγής με το αντίστοιχο Link
* Ημερομηνία
* Ώρα
* Τι είδους εκδήλωση είναι (concert/festival)
* Κατάσταση (Αν θα πραγματοποιηθεί ή όχι)

**Αρχιτεκτονική κλήσεων στο API**

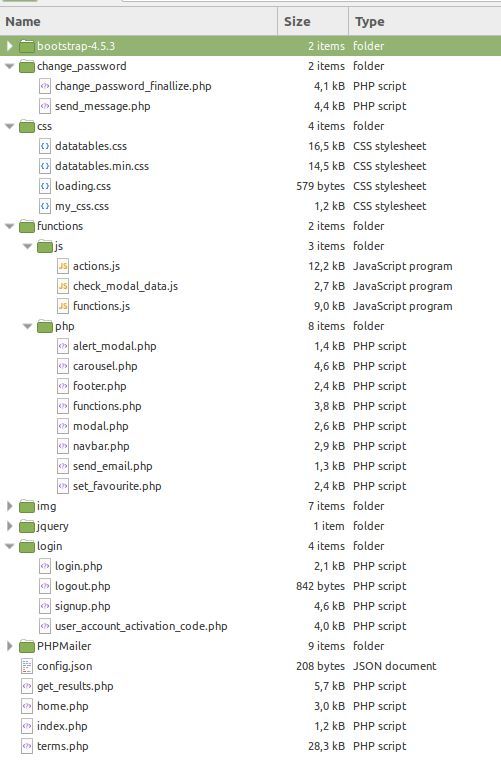
Προκειμένου να αποφευχθούν οι αλλεπάλληλες κλήσεις στο Web Server και τη Βάση, όταν ο χρήστης ζητάει τα events αν δεν είναι συνδεδεμένος αυτά επιστρέφουν χωρίς επιπλέον έλεγχο, ενώ αν έχει συνδεθεί γίνεται έλεγχος των προτιμήσεων του. Δηλαδή για κάθε σετ εκδηλώσεων που επιστρέφει το API, γίνεται έλεγχος της ύπαρξης «αγαπημένων» σε αυτό το σετ. Ο λογικός σχεδιασμός έχει λοιπόν ως εξής:

Εικόνα Κλήση events χωρίς σύνδεση

Εικόνα : Κλήση events από συνδεδεμένο χρήστη

**Δομή Υλοποίησης**

Η εφαρμογή αποτελείται από δύο σελίδες:

* Την κεντρική σελίδα στην οποία με διαχείριση γεγονότων εκτελούνται όλες οι βασικές λειτουργίες
* Τη σελίδα με τους όρους χρήσης της εφαρμογής, τους οποίους πρέπει να αποδεχθεί όποιος επιθυμεί την εγγραφή του.

Εικόνα : Δενδρική Δομή Αρχείων

Οι βιβλιοθήκες και τα αρχεία της εφαρμογής έχουν την παρακάτω δομή:

Χρησιμοποιούνται οι βιβλιοθήκες:

1. Jquery (Javascript)
2. PHPMailer (PHP) για αποστολή email
3. BootStrap 4.5.3 (βασισμένη στο jquery) για διαχείριση της εμφάνισης

Στόχος ήταν ο κώδικας της εφαρμογής να ακολουθήσει τις αρχές της επαναχρησιμοποίησης και του modular σχεδιασμού. Οπότε δόθηκε έμφαση στην ανάπτυξη επαναχρησιμοποιήσιμων moduleς και τη χρήση αυτόνομων τμημάτων κώδικα.

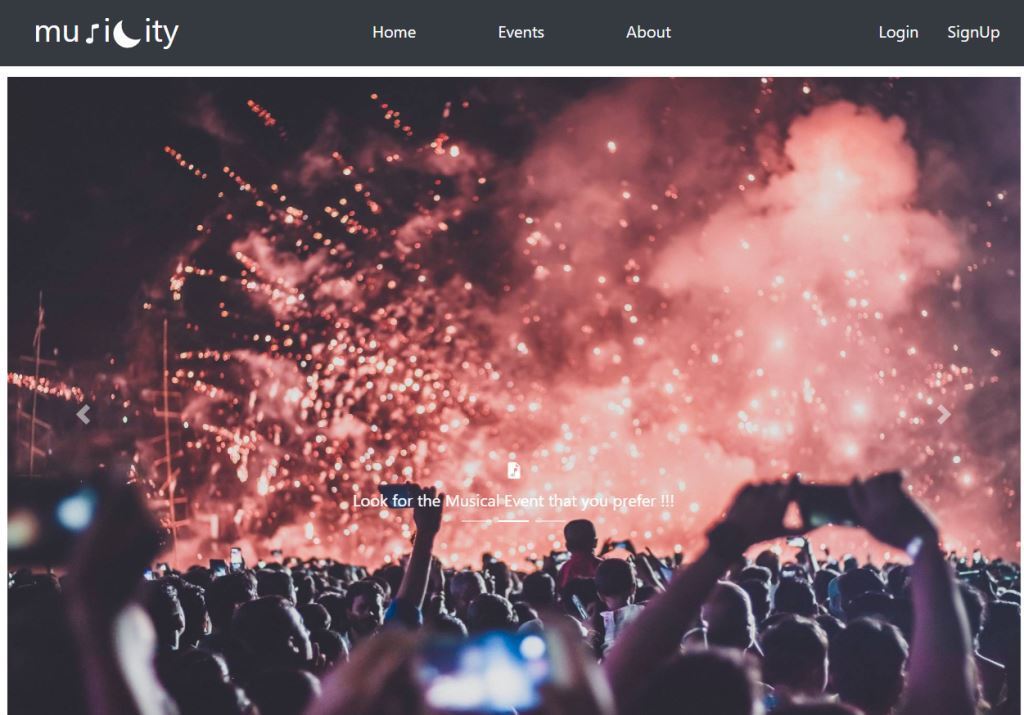
Δομικά συστατικά της εφαρμογής είναι:

1. Βάση δεδομένων με δύο πίνακες όπου καταχωρούνται οι χρήστες και οι προτιμήσεις τους.
2. Κώδικας PHP που αναλαμβάνει την επικοινωνία μεταξύ του client, του API της songkick και της Βάσης MySQL
3. Κώδικας PHP που αναλαμβάνει αποστολή email στους χρήστες κατά τη διαδικασία εγγραφής και αλλαγής κωδικού, προκειμένου να διασφαλιστεί η ταυτοποίηση τους.
4. Κλήσεις Ajax προς το API της songkick προκειμένου να γίνει λήψη πληροφοριών για τις επερχόμενες εκδηλώσεις.
5. Κλήσεις Ajax προς τη βάση προκειμένου να γίνει καταχώριση χρηστών, προτιμήσεων, ελέγχου ύπαρξης χρήστη.
6. Jquery κώδικα που αναλαμβάνει τη διαχείριση των γεγονότων στην πλευρά του client.
7. Χρήση της βιβλιοθήκης Bootstrap, τόσο στο κεντρικό menu (navbar), όσο και στην παραγωγή των popup μηνυμάτων (με modal)
8. CSS του Jquery, Bootstrap, καθώς και δικά μας.

Αρχική σελίδα

Η αρχική σελίδα της διαδικτυακής εφαρμογής περιλαμβάνει μπάρα επιλογών **Home**, **Events**, **About**, **Login**, **SignUp**

Στο index.php γίνεται ενημέρωση με βασικές καθολικές τιμές που είναι απαραίτητες και έχουν σχέση με τη σύνδεση στη βάση, το API, και το λογαριασμό email (configuration.php) και στη συνέχεια καλείται το home.php που αφορά την υλοποίηση της κεντρικής σελίδας.



Εικόνα : Αρχική σελίδα

Η σελίδα είναι χωρισμένη σε τρία μέρη:

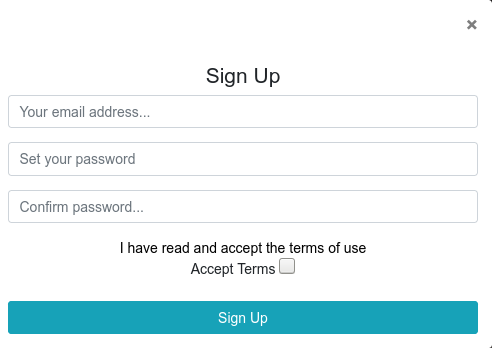
1. Το μενού που υλοποιείται με το στοιχείο navbar του Bootstrap στο αρχείο functions/php/navbar.php.
2. Το κεντρικό θέμα που εμπεριέχεται στο <div> “main\_div” .
3. Το υποσέλιδο που υλοποιείται στο αρχείο functions/php/footer.php.
4. Όταν επιλέγουμε μια ενέργεια (εκτός από την ενημέρωση για τους όρους χρήσης), είτε αντικαθίσταται το περιεχόμενο του “main\_div”, είτε αναδύεται ένα pop-up (σχεδιασμένο με το modal του Bootstrap).

Εγγραφή χρήστη(SignUp)

Εάν ο χρήστης επιθυμεί την **εγγραφή** του, επιλέγει από το SignUp.

Τα ζητούμενα πεδία στη φόρμα εγγραφής είναι:

* Η διεύθυνση ηλεκτρονικού ταχυδρομείου
* Ο κωδικός πρόσβασης με τον οποίο θα πραγματοποιείται η είσοδος στο σύστημα
* Επιβεβαίωση του κωδικού πρόσβασης.
* Επιβεβαίωση αποδοχής των όρων χρήσης της ιστοσελίδας.



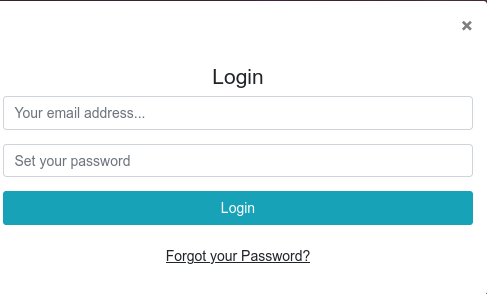
Εικόνα 5: Εγγραφή χρήστη

Κατά τη διαδικασία εγγραφής:

* Γίνεται έλεγχος ύπαρξης χρήστη με την ίδια διεύθυνση ηλεκτρονικού ταχυδρομείου
* Αν υφίσταται, εμφανίζεται το αντίστοιχο μήνυμα και τελειώνει η διαδικασία
* Αν δεν είναι καταχωρημένο το email τότε αποστέλλεται μήνυμα για επιβεβαίωση της πρόθεσης εγγραφής, ενώ ταυτόχρονα καταχωρείται στη βάση σαν μη ενεργός
* Εφόσον επιβεβαιωθεί η επιθυμία του χρήστη για εγγραφή, αλλάζει η κατάσταση του σε «ενεργός», του αποστέλλεται μήνυμα και μπορεί πλέον να συνδεθεί.

Σύνδεση/Αποσύνδεση χρήστη(Login/Logout)

Οι εγγεγραμμένοι χρήστες μπορούν επιλέξουν να συνδεθούν. Απαραίτητα στοιχεία είναι το email και ο κωδικός που έχουν καταχωρήσει κατά την εγγραφή.



Εικόνα : Οθόνη σύνδεσης χρήστη

Όταν ο χρήστης επιθυμεί την αποσύνδεση του από την υπηρεσία, θα μπορεί να κάνει κλικ στο σύνδεσμο «Exit» που βρίσκεται στη μπάρα επιλογών. Η επιλογή της ενέργειας «Exit» θα εμφανίζεται μόνο όταν έχει προηγηθεί η ενέργεια της σύνδεσης. Μόλις γίνει κλικ για «Exit», ο χρήστης θα αποσυνδέεται και θα οδηγείται στην αρχική σελίδα.

# Ανάκτηση κωδικού πρόσβασης

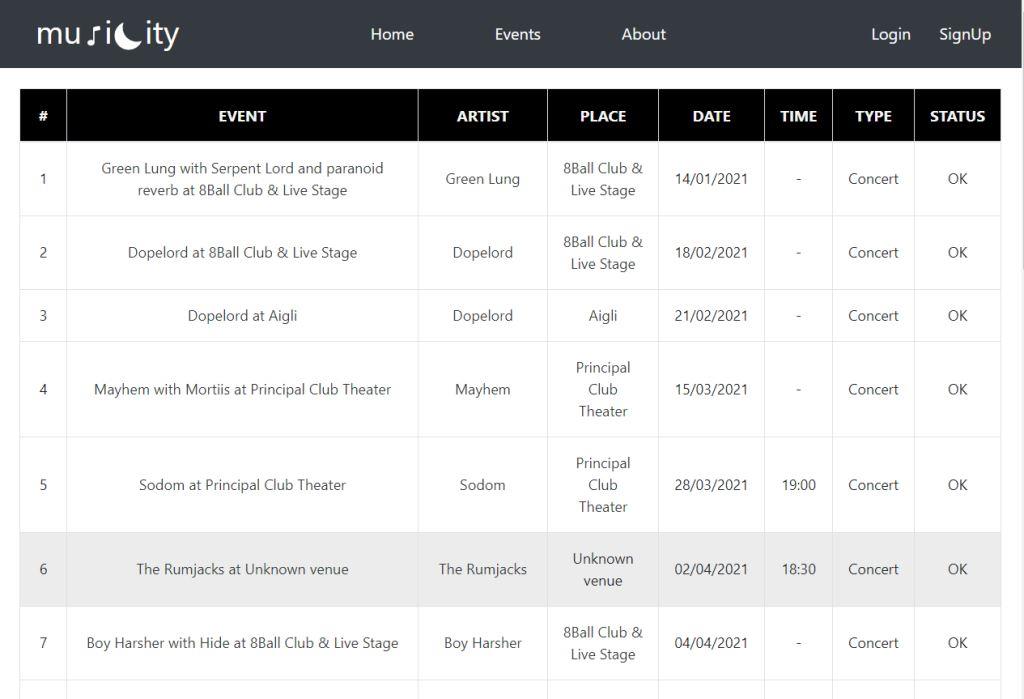
Σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό με τον οποίο πραγματοποιεί σύνδεση, υπάρχει αντίστοιχος σύνδεσμος «**Forget your password**?» που του επιτρέπει να καθορίσει εκ νέου κωδικό πρόσβασης.

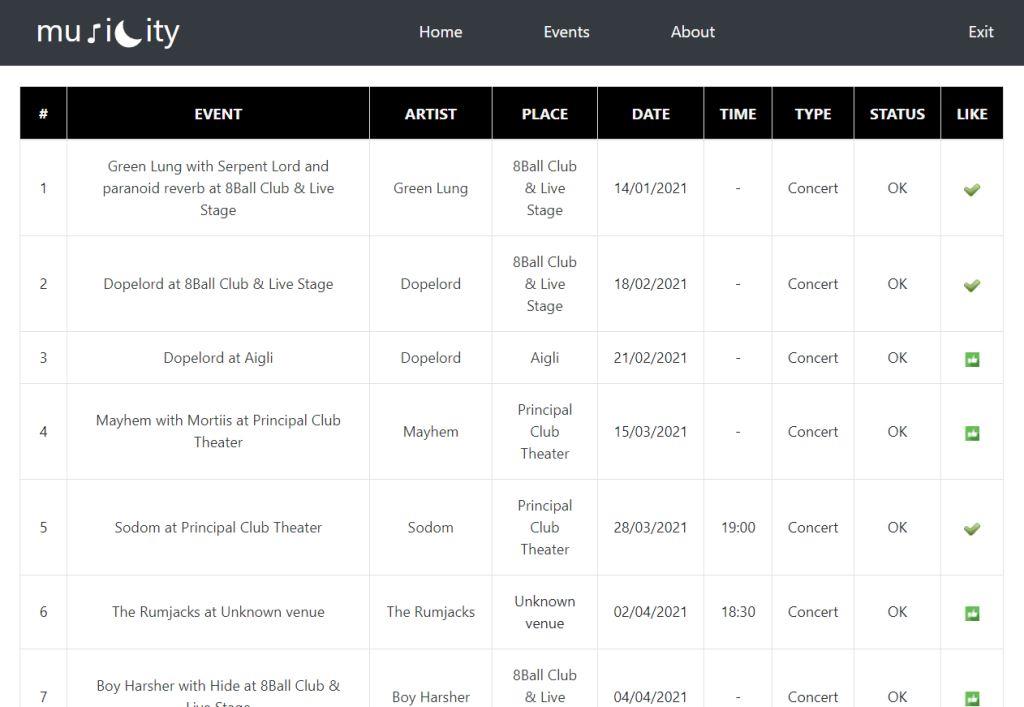
Λειτουργεί με την ίδια φιλοσοφία της διαδικασίας εγγραφής:

1. Ο χρήστης καταχωρεί το email και τον νέο κωδικό πρόσβασης
2. Γίνεται έλεγχος ύπαρξης χρήστη
3. Αν ο χρήστης υφίσταται τότε ο νέος κωδικός αποθηκεύεται σε προσωρινό πεδίο
4. Αποστέλλεται μήνυμα στο χρήστη για επιβεβαίωση της πρόθεσης αλλαγής κωδικού
5. Εφόσον επιβεβαιωθεί η αλλαγή κωδικού, ο προσωρινός κωδικός αντικαθιστά τον τρέχοντα κωδικό και ο χρήστης μπορεί αν συνδεθεί

# Προβολή Events

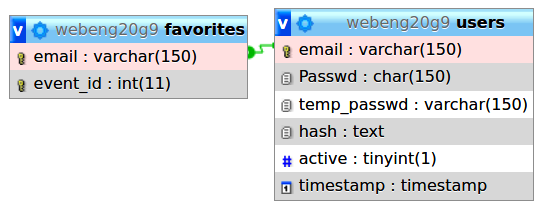
Επιλέγοντας Events προβάλλονται τα μουσικά γεγονότα της Θεσσαλονίκης. Όταν ο χρήστης κάνει μια αναζήτηση δεν επιστρέφουν σαν αποτέλεσμα όλες τις πληροφορίες που μπορεί να στείλει το API. Φιλτράρεται το αποτέλεσμα και στον χρήστη εμφανίζονται:

* Όνομα event με το αντίστοιχο Link
* Όνομα καλλιτέχνη με το αντίστοιχο Link
* Το όνομα του χώρου διεξαγωγής με το αντίστοιχο Link
* Ημερομηνία
* Ώρα
* Τι είδους εκδήλωση είναι (concert/festival)
* Κατάσταση (Αν θα πραγματοποιηθεί ή όχι)

Όταν αυτή η επιλογή γίνει από συνδεδεμένο χρήστη τότε εκτός από την προβολή των μουσικών γεγονότων της πόλης δίνεται η δυνατότητα στο χρήστη να επιλέξει τα event που τον ενδιαφέρουν.

# Βάση δεδομένων

Χρησιμοποιούμε δύο πίνακες για τις ανάγκες της εφαρμογής μας



Ο πίνακας users έχει τα παρακάτω δομή κατασκευής:

CREATE TABLE `users` (

`email` varchar(150) NOT NULL,

`Passwd` char(150) CHARACTER SET latin1 COLLATE latin1\_bin NOT NULL,

`temp\_passwd` varchar(150) NOT NULL DEFAULT '',

`hash` text NOT NULL,

`active` tinyint(1) DEFAULT '0',

`timestamp` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

ALTER TABLE `users`

ADD PRIMARY KEY (`email`);

Αναλυτικότερα:

* email: αποθηκεύεται το email του χρήστη και είναι το κύριο κλειδί
* Passwd: αποθηκεύεται κρυπτογραφημένος (md5) ο κωδικός του χρήστη
* temp\_passwd: αποθηκεύεται προσωρινά ο κρυπτογραφημένος νέος κωδικός, όταν ο χρήστης αιτηθεί αντικατάσταση κωδικού.
* hash: χρησιμοποιείται σαν τμήμα του σύνδεσμου επιβεβαίωσης που αποστέλεται στο χρήστη όταν αιτείται τη δημιουργία λογαριασμού ή την επανέκδοση κωδικού.
* active: Έχει την τιμή 0 όταν ο χρήστης δεν έχει επιβεβαιώσει την επιθυμία εγγραφής στην υπηρεσία και 1 όταν προβεί σε επιβεβαίωση.
* timestamp: Χρησιμοποιείται για τη διαχείριση των αιτημάτων λογαριασμού ή αλλαγής κωδικού που δεν επιβεβαιώνονται. Κάθε 30 λεπτά εκτελείται ένα php.script (μέσω crontab) το οποίο διαγράφει όσα αιτήματα λογαριασμού δεν έχουν επιβεβαιωθεί για 30 λεπτά και μηδενίζει αντίστοιχα τις τιμές στο `temp\_passwd` και `hash` για τις περιπτώσεις που έχει γίνει αίτημα αλλαγής κωδικού και δεν έχει επιβεβαιωθεί.

Ο πίνακας favorites έχει τα παρακάτω δομή κατασκευής:

CREATE TABLE `favorites` (

`email` varchar(150) NOT NULL,

`event\_id` int(11) NOT NULL DEFAULT '2'

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

ALTER TABLE `favorites`

ADD PRIMARY KEY (`email`,`event\_id`);

ALTER TABLE `favorites`

ADD CONSTRAINT `email\_ibfk\_1` FOREIGN KEY (`email`) REFERENCES `users` (`email`) ON UPDATE CASCADE;

COMMIT;

Αναλυτικότερα:

* email: αποθηκεύεται το email του χρήστη
* event\_id: αποθηκεύει το id του γεγονότος

# Αξιολόγηση σελίδας

Δημιουργήθηκε ένα **Ερωτηματολόγιο αξιολόγησης σελίδας**

<https://forms.gle/MPfiqt7ApPsk1Cpm6>