



**ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ  
ΠΜΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΚΤΥΩΝ**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΕΦΑΡΜΟΓΗ INTERNET OF THINGS ΜΕ RASPBERRY PI ΚΑΙ  
PYTHON**

Χρυσούλα Μπαζούκη

Επιβλέπων: Ελευθέριος Στεργίου  
ΔΕΠ, Αναπληρωτής Καθηγητής

Άρτα, Δεκέμβριος, 2021

## **ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

### 1. Επιβλέπων Καθηγητής

Ελευθέριος Στεργίου

ΔΕΠ, Αναπληρωτής Καθηγητής

### 2. Μέλος Επιτροπής

Γρηγόριος Δουμένης

ΔΕΠ, Επίκουρος Καθηγητής

### 3. Μέλος Επιτροπής

Σπυριδούλα Μαργαρίτη

ΕΔΙΠ, Α

## ΠΕΡΙΛΗΨΗ

Η ασφάλεια των τροφίμων στο χώρο αποθήκευσης είναι ένα πολύ σημαντικό ζήτημα για όλους τους εμπλεκόμενους: τους παραγωγούς, τους καταναλωτές, τις κυβερνήσεις, την παγκόσμια κοινότητα. Στο χώρο αποθήκευσης, καίριας σημασίας είναι οι περιβαλλοντικές συνθήκες, όπως για παράδειγμα η θερμοκρασία, υγρασία, βαρομετρική πίεση. Στόχος μας είναι να σχεδιάσουμε και να υλοποιήσουμε ένα έξυπνο σύστημα συνεχούς παρακολούθησης των περιβαλλοντικών συνθηκών στο χώρο αποθήκευσης των τροφίμων για τη βελτίωση της ποιότητας των τροφίμων. Το σύστημα θα πρέπει να ανιχνεύει και να συλλέγει από το χώρο αποθήκευσης πληροφορίες για τη θερμοκρασία, την υγρασία και τη βαρομετρική πίεση, οι οποίες έχουν μεγάλη σημασία για τη διατήρηση της ποιότητας των τροφίμων. Για την υλοποίηση θα πρέπει να καθοριστεί η ροή των δεδομένων, το απαραίτητο υλικό και να αναπτυχθεί το λογισμικό που θα επιτρέπει το χειρισμό και τη λειτουργικότητά του.

Για τον σκοπό αυτό θα χρησιμοποιηθούν συσκευές IoT και εργαλεία λογισμικού. Συγκεκριμένα, η υλοποίηση βασίζεται στο Raspberry Pi 4, σε αισθητήρες συμβατούς με αυτό, όπως το Sense Hat, καθώς και τα κατάλληλα εργαλεία λογισμικού. Στα πλαίσια αυτής της εργασίας, σχεδιάζουμε, αναπτύσσουμε και δοκιμάζουμε σε πραγματικές συνθήκες μια τέτοια εφαρμογή.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ .....	1
1. Εισαγωγή .....	3
1.1 Κίνητρο .....	5
1.2 Αντικείμενο εργασίας και συνεισφορά.....	5
1.3 Δομή Εργασίας .....	7
2. Επισκόπηση Raspberry Pi.....	8
2.1 Εισαγωγή .....	8
2.2 Ανασκόπηση της προόδου του Raspberry Pi.....	9
2.3 Εφαρμογές βασισμένες στο Raspberry Pi.....	12
3. Raspberry Pi: Υλικό, Λογισμικό και Τεχνολογίες.....	15
3.1 Υλικό.....	15
3.2 Λογισμικό .....	20
3.3 Τεχνολογίες.....	21
4. Εφαρμογή Internet of Things με Raspberry Pi και Python.....	23
4.1 Υλοποίηση Εφαρμογής Internet of Things με Raspberry Pi και Python.....	23
4.2 Σχεδίαση Συστήματος.....	23
4.3 Απαιτήσεις Εφαρμογής.....	24
4.4 Συλλογή και διάθεση δεδομένων .....	25
4.5 Ανάπτυξη εφαρμογής.....	26
4.6 Απομακρυσμένη διαχείριση.....	31
5. Λειτουργία και εκτέλεση εφαρμογής.....	39
5.1 Περιγραφή της λειτουργίας και εκτέλεσης εφαρμογής .....	39
5.2 Αποτελέσματα.....	58
5.3 Συζήτηση.....	68
6. Συμπεράσματα .....	71
Βιβλιογραφία .....	72
Παράρτημα Α.....	74
Παράρτημα Β.....	86

Παράρτημα Γ .....	88
Παράρτημα Δ.....	93

## 1. Εισαγωγή

Το Internet of Things(IoT) είναι ένα σύστημα που συγκροτείται από έναν μεγάλο αριθμό έξυπνων αντικειμένων τα οποία είναι αναγνωρίσιμα, ικανά να επικοινωνούν και να αλληλεπιδρούν μεταξύ τους, ώστε με αυτόν τον τρόπο να μπορούν να δημιουργούνται δίκτυα διασυνδεδεμένων αντικειμένων είτε με τελικούς χρήστες ή με άλλες οντότητες στο δίκτυο. Το IoT, εμφανίστηκε το 1999 και μέχρι σήμερα μπορεί να θεωρηθεί ως ένα δυναμικό αλλά και κατανεμημένο δικτυακό σύστημα [7]. Το IoT επιτρέπει τη σύνδεση οικιακών συσκευών, ή κάθε αντικειμένου που ενσωματώνει ηλεκτρονικό υλικό και συνδεσιμότητα στο δίκτυο ούτως ώστε να γίνεται δυνατή η ανταλλαγή δεδομένων [14]. Οι συσκευές αυτές ονομάζονται έξυπνες συσκευές. Ο όρος Things σχετίζεται με συσκευές που ενδέχεται να είναι πολύ διαφορετικές μεταξύ τους, δηλαδή από μια καφετιέρα ή ένα ξυπνητήρι μέχρι πολύπλοκους αισθητήρες αποφυγής σύγκρουσης τοποθετημένους στα αυτοκίνητα [14]. Με την παρουσία έξυπνων συσκευών, έχουμε την δυνατότητα να εξετάζουμε φυσικά φαινόμενα και να τα μεταφράζουμε ή να τα εξηγούμε σε μια ροή δεδομένων πληροφοριών. Επίσης, με την βοήθεια έξυπνων συσκευών έχουμε την ικανότητα να θέτουμε σε λειτουργία διαδικασίες, να μεγιστοποιούμε την ασφάλεια, την άνεση, την ευκολία και την εξοικονόμηση ενέργειας [7]. Αφού τα συστήματα IoT θα μπορούν να σχεδιαστούν, να διαχειρίζονται και να χρησιμοποιούνται από πολλούς ενδιαφερόμενους, με γνώμονα διαφορετικά επιχειρηματικά μοντέλα και διάφορα ενδιαφέροντα, οπότε αυτά τα συστήματα θα πρέπει:

- να καθιστούν δυνατή την κατασκευή νέων εφαρμογών πάνω στα υπάρχοντα συστήματα,
- να καθιστούν δυνατή την ανάπτυξη νέων συστημάτων παράλληλα με τα υπάρχοντα συστήματα,
- να καθιστούν δυνατή ένα επαρκές επίπεδο διαλειτουργικότητας, έτσι ώστε να μπορούν να αναπτυχθούν καινοτόμα και ανταγωνιστικά συστήματα και εφαρμογές μεταξύ τομέων [7].

Οι χρήσεις του IoT είναι απεριόριστες. Μερικά παραδείγματα εφαρμογών του IoT φαίνονται παρακάτω:

**Μετρητές Κατανάλωσης:** Από τις εταιρείες παροχής ρεύματος και νερού, οι οποίες μπορούν να τοποθετούν ειδικούς μετρητές σε κάθε κατοικία ώστε με αυτόν τον τρόπο να καταγράφεται η κατανάλωση. Οι μετρητές αυτοί αποτελούνται από κατάλληλους αισθητήρες και ένα μικροϋπολογιστή ο οποίος με τη σειρά του αναλαμβάνει την επεξεργασία των δεδομένων που διαβάζει από τους αισθητήρες, και στέλνει μέσω διαδικτύου τις πληροφορίες για την κατανάλωση στους αντίστοιχους παρόχους.

**Έξυπνος φωτισμός:** Με τον έξυπνο φωτισμό στις πόλεις, ο οποίος εκτός από τη δυνατότητα εξοικονόμησης ενέργειας όπου είναι εφικτό, μας δίνει επίσης τη δυνατότητα του χαμηλού φωτισμού όταν δεν περνάει κάποιος πεζός. Με αυτόν τον τρόπο, χρησιμοποιώντας ενσωματωμένους αισθητήρες μπορεί να συλλέγει πληροφορίες όπως για τη κίνηση των οχημάτων στους δρόμους και θα λαμβάνει αποφάσεις όπως για παράδειγμα για την αλλαγή των χρονοδοτούμενου προγράμματος των φαναριών κυκλοφορίας με στόχο την καλύτερη δυνατή ροή της κυκλοφορίας.

**Αυτόνομα αυτοκίνητα:** Έρευνες οι οποίες έχουν γίνει από κάποιες εταιρείες σχετικά σε αυτό το τομέα όπου με τη χρήση πολύπλοκων αισθητήρων αλλά και του διαδικτύου δείχνουν ότι το αυτοκίνητο μπορεί να κινηθεί αυτόνομα στο προορισμό που έχει επιλέξει ο οδηγός. Επίσης θα μπορεί ταυτόχρονα να ανταλλάσσει πληροφορίες με άλλα αυτοκίνητα ή ακόμη με τους στύλους του φωτισμού ώστε να επιλέξει την βέλτιστη πορεία προς το προορισμό, αποφεύγοντας τυχόν εμπόδια για τα οποία θα έχει ενημερωθεί εγκαίρως.

**Χρήση στη βιομηχανία:** Άλλη χρήση του IoT είναι ότι μπορεί να εφαρμοστεί σε μεγάλα εργοστάσια. Σε αυτήν την περίπτωση κάθε υποσύστημα της αλυσίδας παραγωγής με χρήση αισθητήρων και μικροελεγκτών, είναι συνδεδεμένο με κάποιο κεντρικό σύστημα το οποίο παρατηρείται από την παραγωγή για τυχόν βλάβες που μπορούν να συμβούν αν προκύψουν και έτσι με αυτόν τον τρόπο ενημερώνεται έγκαιρα ο τεχνικός υπεύθυνος.

**Ιατρική:** Υπάρχουν αρκετές εφαρμογές του IoT και στην Ιατρική. Όπως, για παράδειγμα ότι υπάρχουν ήδη βηματοδότες που συνδέονται άμεσα στο διαδίκτυο. Αυτό έχει ως αποτέλεσμα να μπορεί να ενημερώνεται ο υπεύθυνος γιατρός του ασθενή στην περίπτωση που υπάρξει κάποιο πρόβλημα στην υγεία του. Με άλλα λόγια με αυτήν την τεχνολογία μπορούν να σωθούν αρκετές ζωές.

**Ιχνηλασιμότητα:** Η ιχνηλασιμότητα σχετίζεται με την ικανότητα εντοπισμού, εμπορεύματος, με την ιδιότητα να υπάρχει η δυνατότητα να ακολουθηθεί το ιστορικό του είτε προς τα εμπρός(από την πηγή στον καταναλωτή) ή είτε προς τα πίσω(από τον καταναλωτή προς την πηγή) [15]. Επίσης, αποτελεί πρωταρχικό παράγοντα για τον τομέα των γεωργικών ειδών διατροφής, εξαιτίας της φθαρτής φύσης των τροφίμων και των πιθανών κινδύνων για την υγεία μέσω της μετάδοσης ασθενειών. Ένα εξελιγμένο σύστημα ιχνηλασιμότητας προσκομίζει πολλαπλά οφέλη, όπως αύξηση της ασφάλειας, της εμπιστοσύνης των πελατών και συνεισφέρει στη γρήγορη απόσυρση επικίνδυνων τροφίμων [15]. Στην Ευρωπαϊκή Ένωση η ιχνηλασιμότητα των τροφίμων ελέγχεται αυστηρά και οι επιχειρήσεις στον τομέα των τροφίμων πρέπει να είναι σε θέση να καθορίζουν την προέλευση και τον προορισμό του κάθε προϊόντος διατροφής που παράγουν.

## 1.1 Κίνητρο

Οι συνθήκες αποθήκευσης τροφίμων πρέπει να ελέγχονται και να ρυθμίζονται ανά πάσα στιγμή για να διατηρούνται τα προϊόντα σε ασφαλή και κατάλληλα για κατανάλωση. Η συνεχής και συστηματική παρακολούθηση, δεν είναι απλώς αναγκαία αλλά επιβεβλημένη καθώς:

- αφορά την ασφάλεια των τροφίμων: Εάν οι περιβαλλοντικοί παράγοντες σε έναν χώρο αποθήκευσης γίνουν πολύ ακραίοι, τα τρόφιμα θα τεθούν σε κίνδυνο, γεγονός που μπορεί να βλάψουν σοβαρά την υγεία των καταναλωτών.
- συμβάλει στη μείωση των απορριμμάτων και της σπατάλης των φυσικών πόρων. Σημαντικό ποσοστό των τροφίμων σήμερα καταλήγει στα σκουπίδια λόγω κακών συνθηκών στην κατ' οίκον αποθήκευση [19].
- συνδέεται άμεσα με την κλιματική αλλαγή. Η κλιματική αλλαγή επηρεάζει τη διάρκεια διατήρησης των αποθηκευμένων τροφίμων και αποτελεί σημαντική αιτία καταστροφής τους [20] [21].

Ένας από τους πιο σημαντικούς παράγοντες στην αποθήκευση τροφίμων είναι η θερμοκρασία, η οποία επηρεάζει μεταξύ άλλων την μικροβιακή ανάπτυξη, ενώ σε άλλες περιπτώσεις επηρεάζει την ωρίμανση (π.χ. φρούτων και λαχανικών). Εκτός από τη θερμοκρασία, η σχετική υγρασία είναι ένα εξίσου σημαντικός παράγοντας στην αποθήκευση τροφίμων. Η σχετική υγρασία (RH) επηρεάζει την απώλεια υγρασίας των τροφίμων και τη σχετική απώλεια υδατοδιαλυτών βιταμινών ή μπορεί να οδηγήσει στην ανάπτυξη μούχλας σε αυτά [19]. Επίσης, ένας άλλος παράγοντας ο οποίος είναι σημαντικός για την αποθήκευση τροφίμων είναι η βαρομετρική πίεση η οποία επηρεάζει την επεξεργασία που έχουν υποβληθεί τα τρόφιμα [18]. Σήμερα, η συνεχής και συστηματική παρακολούθηση των συνθηκών ενός χώρου αποθήκευσης τροφίμων, δεν μπορεί να είναι απλά ευθύνη του ανθρώπου και να βασίζεται σε τακτικούς ελέγχους.

Έχοντας ως κίνητρο τις παραπάνω διαπιστώσεις, στα πλαίσια αυτής της εργασίας εξετάζουμε πως η τεχνολογία IoT θα συνδράμει και θα δώσει λύσεις σε τέτοιου είδους προβλήματα.

## 1.2 Αντικείμενο εργασίας και συνεισφορά

Στα πλαίσια αυτής της εργασίας μελετάμε τη δημιουργία συστήματος παρακολούθησης των συνθηκών περιβάλλοντος ενός χώρου αποθήκευσης τροφίμων. Σε έναν τέτοιο χώρο, οι ιδιαίτερες συνθήκες όπως είναι η θερμοκρασία, η υγρασία και η βαρομετρική πίεση είναι βαρύνουσας σημασίας για τη διατήρηση της ποιότητας των τροφίμων. Στόχος μας είναι να σχεδιάσουμε και να υλοποιήσουμε ένα έξυπνο σύστημα συνεχούς παρακολούθησης των συνθηκών του χώρου αποθήκευσης. Ο



χώρος που θα εστιάσουμε την μελέτη μας προορίζεται για την αποθήκευση νωπών φρούτων όπως πορτοκάλια, μανταρίνια και λεμόνια τα οποία απαιτούν ελεγχόμενες συνθήκες αποθήκευσης.

Αξιοποιώντας το Raspberry Pi και μια σειρά από αισθητήρες θα αναπτύξουμε μια εφαρμογή για την παρακολούθηση και καταγραφή των περιβαλλοντικών συνθηκών του χώρου. Το σύστημα θα πρέπει να ανιχνεύει και να συλλέγει δεδομένα από το χώρο αποθήκευσης με τη βοήθεια αισθητήρων και να τροφοδοτεί απομακρυσμένα μια σχετική βάση δεδομένων. Επίσης, το σύστημα θα μπορεί να δημιουργεί προσαρμοσμένες αναφορές, εξαγωγή δεδομένων για περαιτέρω επεξεργασία ώστε να ελέγχονται αποτελεσματικά οι συνθήκες αποθήκευσης των τροφίμων. Ακόμη, το σύστημα θα έχει τη δυνατότητα, υπό ορισμένες προϋποθέσεις, που εμείς θα καθορίσουμε και ανάλογα με τις τιμές για τη θερμοκρασία, την υγρασία και την βαρομετρική πίεση θα στέλνει αυτόματες ειδοποιήσεις στον διαχειριστή του συστήματος. Με άλλα λόγια αν η υγρασία είναι μεγαλύτερη του 70 και η θερμοκρασία είναι μεταξύ των 10 μέχρι 18 βαθμούς Κελσίου, που έχουμε ορίσει εμείς ως κατάλληλες συνθήκες για τον χώρο αποθήκευσής μας, τότε θα εμφανίζει τα κατάλληλα μηνύματα στην ιστοσελίδα και ταυτόχρονα το Raspberry Pi θα ενεργοποιεί μια φωτεινή ένδειξη. Για την υλοποίηση θα πρέπει να καθοριστεί η ροή των δεδομένων, το απαραίτητο υλικό και να αναπτυχθεί το λογισμικό που θα επιτρέπει το χειρισμό και τη λειτουργικότητά του.

Για τον σκοπό αυτό θα χρησιμοποιηθούν συσκευές IoT και εργαλεία λογισμικού, καθώς επίσης και η γλώσσα προγραμματισμού Python. Με τη βοήθεια της γλώσσας προγραμματισμού Python θα αναπτύξουμε και θα υλοποιήσουμε την εφαρμογή.

Στόχος της εργασίας είναι να μετατρέψει έναν χώρο αποθήκευσης νωπών τροφίμων σε έναν “έξυπνο” χώρο αποθήκευσης που θα παρακολουθεί τις περιβαλλοντικές συνθήκες και θα ενημερώνει όταν και όπου χρειαστεί με κατάλληλα μηνύματα με την χρήση συσκευών IoT, Raspberry Pi και αισθητήρων. Η εφαρμογή, που θα την καλούμε “έξυπνος χώρος αποθήκευσης” είναι σε θέση να δέχεται ερεθίσματα από το περιβάλλον και να αυτοματοποιεί τις λειτουργίες της. Συγκεκριμένα, η συνεισφορά της εργασίας μας συνοψίζεται στα ακόλουθα:

- Προτείνουμε ένα έξυπνο σύστημα παρακολούθησης χώρου αποθήκευσης τροφίμων.
- Σχεδιάζουμε και υλοποιούμε το σύστημα έξυπνης αποθήκευσης.
- Προτείνουμε το υλικό και λογισμικό.
- Αξιολογούμε την απόδοση του σε πραγματικές συνθήκες.

### 1.3 Δομή Εργασίας

Η συγκεκριμένη εργασία δομείται σε έξι ενότητες. Η πρώτη ενότητα αναφέρεται στο κίνητρο το οποίο μας εκάνε να αναπτύξουμε την συγκεκριμένη εφαρμογή, το αντικείμενο εργασίας καθώς και στη συνεισφορά εργασίας.

Στη δεύτερη ενότητα γίνεται η επισκόπηση Raspberry Pi. Παρουσιάζουμε τις προηγούμενες γενιές του Raspberry Pi μέχρι και το Raspberry Pi 4 που χρησιμοποιήσαμε για την εφαρμογή μας. Επίσης, παρουσιάζουμε και τις εφαρμογές που χρησιμοποιούνται με το Raspberry Pi δηλαδή που κάνει εφαρμογή η συσκευή αυτή.

Στην τρίτη ενότητα παρουσιάζουμε το υλικό, λογισμικό και τις τεχνολογίες του Raspberry Pi. Στην τέταρτη ενότητα αναπτύσσεται ο τρόπος υλοποίησης της εφαρμογής μας. Σε αυτήν την ενότητα, θα παρουσιάσουμε την σχεδίαση του συστήματος, τις απαιτήσεις της εφαρμογής οι οποίες είναι οι λειτουργικές και μη λειτουργικές, καθώς επίσης αναφερόμαστε και στις απαιτήσεις του συστήματος. Παρουσιάζουμε τη συλλογή και διάθεση των δεδομένων, την ανάπτυξη της εφαρμογής και την απομακρυσμένη διαχείριση όπου ο κάθε χρήστης μπορεί να συνδεθεί ασύρματα στην εφαρμογή μας.

Στην πέμπτη ενότητα περιγράφουμε την λειτουργία και την εκτέλεση της εφαρμογής μας. Αναφερόμαστε επίσης στα αποτελέσματα που έχουμε εξάγει από τα διαγράμματα και γίνεται σύζηση των αποτελεσμάτων αυτών.

Στην έκτη ενότητα έχουμε τα συμπεράσματα σχετικά με την εφαρμογή μας, η οποία υλοποιήθηκε στα πλαίσια αυτής της εργασίας.

## 2. Επισκόπηση Raspberry Pi

### 2.1 Εισαγωγή

Το Raspberry Pi είναι ένας μικρός υπολογιστής του γνωστού ιδρύματος Raspberry Pi, που δημιουργήθηκε στην Αγγλία και έχει κατασκευαστεί γύρω από έναν επεξεργαστή της αρχιτεκτονικής Advanced RISC Machines (ARM). Οι δημιουργοί του είχαν εμπνευστεί από ένα άλλο βρετανικό υπολογιστή τον BBC Micro ο οποίος διατέθηκε στην Αγγλία στις αρχές του 1980 και σχεδιάστηκε με στόχο τη χρήση του στην εκπαίδευση, όπου και πραγματοποιήθηκε. Το BBC Micro είχε διατεθεί σε τρεις βασικές εκδόσεις με τις ακόλουθες ονομασίες, το Model A, το Model B και καταλήγουμε με το τελευταίο μοντέλο το Model B+ , το οποίο είναι το πιο ισχυρό.

Το πρώτο μοντέλο Raspberry Pi, το Model B, δημιουργήθηκε το 2011 και κυκλοφόρησε για πρώτη φορά στην αγορά τον Φεβρουάριο του 2012. Τότε έγινε μεγάλη επιτυχία αφού στα δύο πρώτα χρόνια είχαν πουληθεί πάνω από δυο εκατομμύρια συσκευές. Η ονομασία των συγκεκριμένων εκδόσεων προήλθε από τα ανάλογα μοντέλα του BBC Micro, με εξαίρεση ένα μοντέλο του Raspberry Pi το οποίο είναι αρκετά μικρό σε μέγεθος και ονομάζεται Raspberry Pi Zero.

Αρχικά το Raspberry Pi χρησιμοποιήθηκε για την προώθηση της διδασκαλίας της βασικής επιστήμης των υπολογιστών στα σχολεία κυρίως στις αναπτυσσόμενες χώρες [3]. Το αρχικό μοντέλο έγινε πιο περιζήτητο από το αναμενόμενο, όπου οι πωλήσεις ήταν έξω από τον στόχο της αγοράς και οι οποίες με τον καιρό άρχισαν να αυξάνονται για χρήσεις όπως η ρομποτική. Χρησιμοποιείται ευρέως σε πολλές περιοχές, όπως για την παρακολούθηση του καιρού ή από ανθρώπους που έχουν τον προγραμματισμό ως χόμπι τους, λόγω της υιοθέτησης των θυρών HDMI και USB [3]. Με την κυκλοφορία του Raspberry Pi 2 της γενιάς Model B, το Ίδρυμα Raspberry Pi ίδρυσε μια νέα οντότητα, με την ονομασία Raspberry Pi Trading, και έθεσε τον Eben Upton ως Διευθύνων Σύμβουλος, με την ευθύνη της ανάπτυξης τεχνολογίας. Το Ίδρυμα ορίστηκε ειδικώς ως εκπαιδευτικό φιλανθρωπικό ίδρυμα για την προώθηση της διδασκαλίας της βασικής επιστήμης των υπολογιστών σε σχολεία και αναπτυσσόμενες χώρες [3]. Το Raspberry Pi είναι ένας από τους υπολογιστές που υπάρχουν στη Βρετανία με τις καλύτερες πωλήσεις. Από τον Δεκέμβριο του 2019, έχουν διατεθεί περισσότερα από τριάντα εκατομμύρια συσκευές. Τα περισσότερα Raspberry Pi δημιουργούνται σε ένα εργοστάσιο της Sony στο Pencoed της Ουαλίας, μολονότι άλλα κατασκευάζονται στην Κίνα και στην Ιαπωνία [3].

## 2.2 Ανασκόπηση της προόδου του Raspberry Pi

Στον Πίνακα 1 παρουσιάζονται οι γενιές του Raspberry Pi, αρχίζοντας από την πρώτη γενιά και φτάνοντας στο 2019 και 2020 με την τέταρτη γενιά. Ξεκινάμε με το Raspberry Pi και τελειώνουμε με το Raspberry Pi 4.

Μοντέλο		RAM	Επεξεργαστής	Ενσύρματο Δίκτυο	Ασύρματο Δίκτυο	GPIO	USB ports	1η κυκλοφορία
Raspberry Pi	B	256 ή 512 MiB	1x ARM1176 @ 700Mhz	Ναι	Όχι	26-pin	2	2012
	A	256 MiB		Όχι			1	2013
	B+	512 MiB		Ναι			2	2014
	A+	256 ή 512 MiB		Όχι			1	2014
Raspberry Pi 2	B	1 GiB	4x Cortex-A7 @ 900Mhz	Ναι	Όχι	40-pin	4	2015
Raspberry Pi Zero	Zero	512 MiB	1x ARM1176 @ 1000 Mhz	Όχι	Όχι		1 Micro-USB	2015
	W/WH							Ναι
Raspberry Pi 3	B	1 GiB	4x Cortex-A53 @ 1200Mhz (64/32-bit)	Ναι	Ναι		4	2016
	A+	512 MiB	4x Cortex-A53 @ 1400Mhz (64/32-bit)	Όχι		1	2018	
	B+	1 GiB		Ναι		4	2018	
Raspberry Pi 4	B	2, 4 ή 8 GiB	4x Cortex-A72 @ 1500Mhz (64/32-bit)	Ναι (Gigabit Ethernet)	Ναι	2x USB2.0 2x USB3.0	2019-2020	

Πίνακας 1: Γενιές Raspberry Pi(Πηγή: [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi))

### **Raspberry Pi**

Το Raspberry Pi της γενιάς Model B, κυκλοφόρησε για πρώτη φορά το 2012 και το κόστος του ήταν αρκετά χαμηλό το οποίο έφτανε μέχρι τα 35 δολάρια [2]. Παρόλο που ήταν ένας φθηνός υπολογιστής, μπορούσε να τρέξει βασικά λειτουργικά όπως το Linux. Το Raspberry Pi της γενιάς Model B έχει τα εξής χαρακτηριστικά:

- Επεξεργαστή: 1x ARM1176 @ 700Mhz
- 26 ακροδέκτες GPIO
- 2 θύρες USB
- Μνήμη RAM: 256-512 MB

Διέθετε επίσης ενσύρματη σύνδεση και όχι ασύρματη σύνδεση.

Το Raspberry Pi της γενιάς Model A το οποίο κυκλοφόρησε το 2013 και έχει τα εξής χαρακτηριστικά:

- Επεξεργαστή: 1x ARM1176 @ 700Mhz
- 26 ακροδέκτες GPIO
- 1 θύρα USB
- Μνήμη RAM: 256 MB

Επίσης κυκλοφόρησε το Raspberry Pi της γενιάς Model A+ το οποίο έχει τα εξής χαρακτηριστικά:

- Επεξεργαστή: 1x ARM1176 @ 700Mhz
- 40 ακροδέκτες GPIO
- 2 θύρες USB
- Μνήμη RAM: 256-512 MB

Η πρώτη γενιά του Model A αλλά και του Model A+ του Raspberry Pi δεν διατηρούσαν κάποια σύνδεση στο διαδίκτυο δηλαδή είτε ασύρματη σύνδεση ή ενσύρματη σύνδεση. Το 2014 ενισχύθηκε το I/O στους σαράντα ακροδέκτες από τους είκοσι έξι που είχαν τα πρώτα μοντέλα. Το ίδιο έτος είχε κυκλοφορήσει και το Raspberry Pi, της γενιάς Model B+. Το συγκεκριμένο μοντέλο περιείχε τα παρακάτω χαρακτηριστικά:

- Επεξεργαστή: 1x ARM1176 @ 700Mhz
- 40 ακροδέκτες GPIO
- 2 θύρες USB
- Μνήμη RAM: 512 MB

Λειτουργούσε για ενσύρματο δίκτυο και όχι για ασύρματο δίκτυο.

### **Raspberry Pi 2**

Το 2015 κυκλοφόρησε το Raspberry Pi 2, της γενιάς Model B. Αυτό το μοντέλο περιλάμβανε τα παρακάτω χαρακτηριστικά:

- Επεξεργαστή 4x Cortex-A7 @ 900Mhz
- 40 ακροδέκτες GPIO
- 4 θύρες USB
- Μνήμη RAM: 1 GB

Λειτουργούσε σε ενσύρματο δίκτυο και όχι σε ασύρματο δίκτυο.

### **Raspberry Pi Zero**

Το έτος που κυκλοφόρησε το Raspberry Pi 2 διατέθηκε επίσης και το Zero όπου και αυτό στην πρώτη του εκδοχή δεν είχε σύνδεση στο διαδίκτυο. Όμως ήταν μικρό ώστε να χωράει παντού και αρκετά οικονομικό με κόστος περίπου πέντε ευρώ. Στην ίδια συσκευή τοποθετείται μια στήλη με όνομα GPIO, όπου μπορούμε να παρατηρήσουμε τον αριθμό των ακροδεκτών εισόδων-εξόδων ή ειδαίλλως General Purpose Input-Output. Μετά την γενιά του Zero για το Raspberry Pi Zero, έχουμε και την γενιά του W/WH. Το μοντέλο W/WH σε αυτήν την εκδοχή διαθέτει ασύρματη σύνδεση σε

σχέση με το Zero. Όσον αφορά για τα υπόλοιπα χαρακτηριστικά είναι τα ίδια με το Zero τα οποία είναι τα εξής:

- Επεξεργαστής: 1x ARM1176 @ 1000Mhz
- Μνήμη RAM: 512 MB
- 40 ακροδέκτες GPIO
- 1 θύρα Micro-USB

Επίσης, δεν διαθέτει ενσύρματη σύνδεση και κυκλοφόρησε το 2017.

### **Raspberry Pi 3**

Τα έτη 2016 και 2018 βγήκε σε κυκλοφορία το Raspberry Pi 3 σε τρεις διαφορετικές γενιές, οι οποίες είναι γενιάς Model B, γενιάς Model A+ και γενιάς Model B+. Τα μοντέλα B και B+ του Raspberry Pi 3 έχουν:

- Μνήμη RAM: 1 GB
- 40 ακροδέκτες GPIO
- 4 θύρες USB

Λειτουργούν τόσο σε ενσύρματο δίκτυο όσο και σε ασύρματο δίκτυο. Διαφέρουν μόνο στον επεξεργαστή. Για το μοντέλο B χρησιμοποιείται ο επεξεργαστής 4x Cortex-A53 @ 1200Mhz(64/32-bit). Ενώ το μοντέλο A+ του Raspberry Pi 3 διαθέτει:

- Επεξεργαστής: 4x Cortex-A53 @ 1400Mhz(64/32-bit)
- Μνήμη RAM: 512 MB
- 40 ακροδέκτες GPIO
- 1 θύρα USB

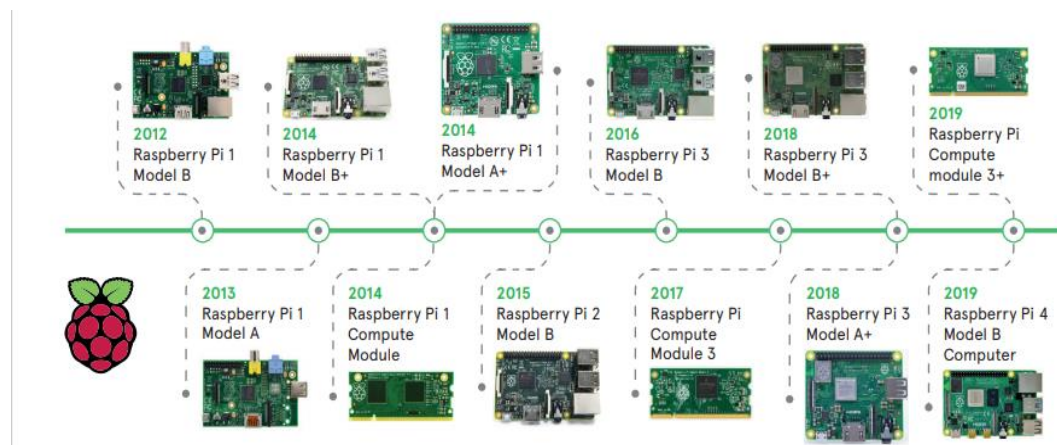
Ωστόσο το Raspberry Pi 3 Model A+ έχει επίσης ενσύρματη σύνδεση, δεν έχει ασύρματη σύνδεση και χρησιμοποιεί τον ίδιο επεξεργαστή του μοντέλου B+. Για την παρούσα εργασία μπορούμε να χρησιμοποιήσουμε οποιοδήποτε μοντέλο δεύτερης ή μεταγενέστερης γενιάς Model B ή Model B+, τα οποία έχουν σύνδεση με το δίκτυο που είναι αναγκαίο για να αναπτύξουμε εφαρμογές IoT, αλλά και σαράντα ακροδέκτες για να είναι τελειώς συμβατό με το Sense Hat όπου θα αναφερθούμε αργότερα. Οπότε αν παραμείνουμε μόνο στα μοντέλα B ή B+ τότε παρατηρούμε ότι οι μεταξύ τους διαφορές είναι κυρίως οι επιδόσεις, δηλαδή ο αριθμός πυρήνων του επεξεργαστή, το μέγεθος της μνήμης, η ταχύτητα της κάρτας δικτύου και άλλα.

### **Raspberry Pi 4**

Το Raspberry Pi 4 της γενιάς Model B έχει:

- Μνήμη RAM: 2 GB
- Επεξεργαστή 4x Cortex-A72 @ 1500Mhz(64/32-bit)
- 40 ακροδέκτες GPIO
- 2 θύρες για USB2.0
- 2 θύρες για USB3.0

Διαθέτει ενσύρματη και ασύρματη σύνδεση και κυκλοφόρησε το 2019-2020. Το Raspberry Pi 4 της γενιάς Model B είναι ένα μοντέλο το οποίο μπορεί να έχει επίσης RAM 4 ή 8 GB. Στην Εικόνα 1 φαίνεται η ιστορική εξέλιξη του Raspberry Pi, δηλαδή από την πρώτη του κυκλοφορία που είναι το Raspberry Pi της γενιάς Model B μέχρι και το τελευταίο μοντέλο του που υπάρχει έως τώρα Raspberry Pi 4 της γενιάς Model B.



Εικόνα 1: Ιστορική εξέλιξη του Raspberry Pi [1]

### 2.3 Εφαρμογές βασισμένες στο Raspberry Pi

Όσον αφορά την χρήση του Raspberry Pi, θα εξετάσουμε με ποιον τρόπο αρχίσε η ιδέα για να κατασκευαστεί αυτή η συσκευή και ποιους ωφέλησε η χρήση του. Ξεκίνησε από τον συγγραφέα της τεχνολογίας Glyn Moody ο οποίος σε μια εργασία του τον Μάιο του 2011 το περιγράφει ως "πιθανό BBC Micro 2.0", όχι αντικαθιστώντας συνηθισμένα μηχανήματα με υπολογιστή αλλά συμπληρώνοντάς τα [3]. Τον Μάρτιο του 2012 ο Stephen Pritchard αναφέρει ότι το Raspberry Pi είναι ο διάδοχος του BBC Micro στο ITPRO. Ο Alex Hope, ευελπιστεί ότι ο υπολογιστής θα προσελκύσει τα παιδιά με τον ενθουσιασμό του προγραμματισμού. Ο Ian Livingstone δήλωσε ότι το BBC θα μπορούσε να συμμετάσχει στη δημιουργία υποστήριξης για τη συσκευή, ενδεχομένως να το κατονομάσει ως BBC Nano [3]. Το Centre for Computing History στηρίζει το έργο Raspberry Pi, φρονώντας ότι θα μπορούσε να "εισάγει μια νέα εποχή".

Ο Harry Fairhead, όμως, προτείνει ότι πρέπει να δοθεί μεγαλύτερη έμφαση στην πρόοδο του εκπαιδευτικού λογισμικού που διατίθεται σε υπάρχον hardware, εφαρμόζοντας εργαλεία όπως το Google App Inventor για να επιστρέψει τον προγραμματισμό στα σχολεία, αντί να επισυνάψει νέες επιλογές hardware [3].

Τον Οκτώβριο του 2012, το Raspberry Pi πήρε το βραβείο T3's Innovation of the Year και ο μελλοντολόγος Mark Pesce ανέφερε ένα (δανειζόμενο)Raspberry Pi ως

επιφοίτηση για τη δουλειά του με το Ambient Device του MooresCloud [3]. Τον Οκτώβριο του 2012, η British Computer Society (BCS) αντέδρασε στην δημοσίευση βελτιωμένων προδιαγραφών αναφέροντας, "it's definitely something we'll want to sink our teeth into (υπάρχει σίγουρα κάτι που θα θέλαμε να βυθίσουμε τα δόντια μας)".

Τον Ιούνιο του 2017, το Raspberry Pi πήρε το Βραβείο Royal Academy of Engineering MacRobert. Η αναφορά για την απονομή του βραβείου στο Raspberry Pi δήλωνε ότι ήταν «για τους φθηνούς μικροϋπολογιστές μεγέθους πιστωτικής κάρτας, οι οποίοι προσδιορίζουν λεπτομερώς τον τρόπο με τον οποίο οι άνθρωποι δουλεύουν με την πληροφορική, εμπνέοντας τους μαθητές να μάθουν την κωδικοποίηση και την επιστήμη των υπολογιστών, όπως επίσης και να διαθέτουν καινοτόμες λύσεις ελέγχου για τη βιομηχανία» [3]. Clusters εκατοντάδων Raspberry Pi έχουν εφαρμοστεί για δοκιμές προγραμμάτων που προορίζονται για υπερυπολογιστές.

Η χρήση της συσκευής του Raspberry Pi καθιερώθηκε στην εκπαίδευση, στον οικιακό αυτοματισμό (Home automation), στον βιομηχανικό αυτοματισμό (Industrial automation) και στα εμπορικά προϊόντα [3].

Τον Ιανουάριο του 2012, υπάρχουν έρευνες σχετικά με την πλατφόρμα του Raspberry Pi στο Ηνωμένο Βασίλειο, από σχολεία τόσο του δημόσιου όσο και του ιδιωτικού τομέα, με ενδιαφέρον περίπου πέντε φορές περισσότερο από τον ιδιωτικό τομέα. Ευελπιστείται ότι οι επιχειρήσεις θα στηρίζουν αγορές για λιγότερο προνομιούχα σχολεία [3]. Το 2014, το Ίδρυμα Raspberry Pi προσέλαβε συγκεκριμένα μέλη της κοινότητάς του, συμπεριλαμβανομένων πρώην εκπαιδευτικών και προγραμματιστών λογισμικού, για να αρχίσει ένα σύνολο δωρεάν πόρων μάθησης για την ιστοσελίδα του. Το Ίδρυμα άρχισε επίσης ένα εκπαιδευτικό σεμινάριο που ονομάζεται Picademy με αποτέλεσμα να βοηθήσει τους εκπαιδευτικούς να προετοιμαστούν για τη διδασκαλία του νέου προγράμματος σπουδών υπολογιστών κάνοντας χρήση του Raspberry Pi στην τάξη [3]. Το 2018, η NASA άρχισε το JPL Open Source Rover Project, το οποίο είναι μια πεπερασμένη έκδοση του Curiosity rover και χρησιμοποιεί ένα Raspberry Pi ως μονάδα ελέγχου, για να στηρίζει τους μαθητές να ασχοληθούν με τη μηχανική, το λογισμικό, την ηλεκτρονική και τη ρομποτική.

Υπάρχουν αρκετοί προγραμματιστές και εφαρμογές που κάνουν χρήση του Raspberry Pi για οικιακό αυτοματισμό. Αυτοί οι προγραμματιστές δοκιμάζουν να τροποποιήσουν το Raspberry Pi σε μια οικονομικά εφικτή λύση για την παρακολούθηση της ενέργειας και στην κατανάλωση ενέργειας. Λόγω του σχετικά χαμηλού κόστους του Raspberry Pi, αυτό έχει γίνει μια περιζήτητη και οικονομική εναλλακτική λύση στις πιο εξεζητημένες εμπορικές λύσεις [3].

Τον βιομηχανικό αυτοματισμό τον Ιούνιο του 2014, ο Πολωνός δημιουργός του βιομηχανικού αυτοματισμού TECHBASE κυκλοφόρησε το ModBerry, έναν υπολογιστή για τις βιομηχανίες που βασίζεται στο Raspberry Pi Compute Module [3]. Η συσκευή έχει διάφορες διεπαφές, κυρίως σειριακές θύρες RS-485/232, ψηφιακές



και αναλογικές εισόδους / εξόδους, CAN και economical 1-Wire buses, τα οποία μεταχειρίζονται ευρέως στη βιομηχανία αυτοματισμού. Ο σχεδιασμός καταλογίζει τη χρήση του Compute Module σε περιβάλλοντα όπου οι συνθήκες δεν είναι τόσο εύκολες για την βιομηχανία, οδηγώντας στο συμπέρασμα ότι το Raspberry Pi δεν δεσμεύεται πλέον σε οικιακά και επιστημονικά έργα, αλλά μπορεί να χρησιμοποιηθεί ευρέως ως λύση Industrial IoT και να επιτύχει στόχους του Industry 4.0 [3].

Τον Μάρτιο του 2018, η SUSE ανακοίνωσε την εμπορική υποστήριξη για το SUSE Linux Enterprise στο Raspberry Pi 3 Model B για να εγγυηθεί έναν αριθμό μη γνωστών πελατών που χρησιμοποιούν την παρακολούθηση των βιομηχανιών με το Raspberry Pi [3].

Τον Ιανουάριο του 2021, η TECHBASE ανακοίνωσε ένα Raspberry Pi Compute Module 4 cluster για χρήση επιταχυντή τεχνητής νοημοσύνης, δρομολόγησης και αρχείων του server. Η συσκευή περιλαμβάνει ένα ή περισσότερα τυπικά Raspberry Pi Compute Module 4s σε ένα βιομηχανικό περίβλημα DIN rail, με ορισμένες εκδόσεις που περιλαμβάνουν μία ή περισσότερες μονάδες επεξεργασίας Coral Edge Tensor [3].

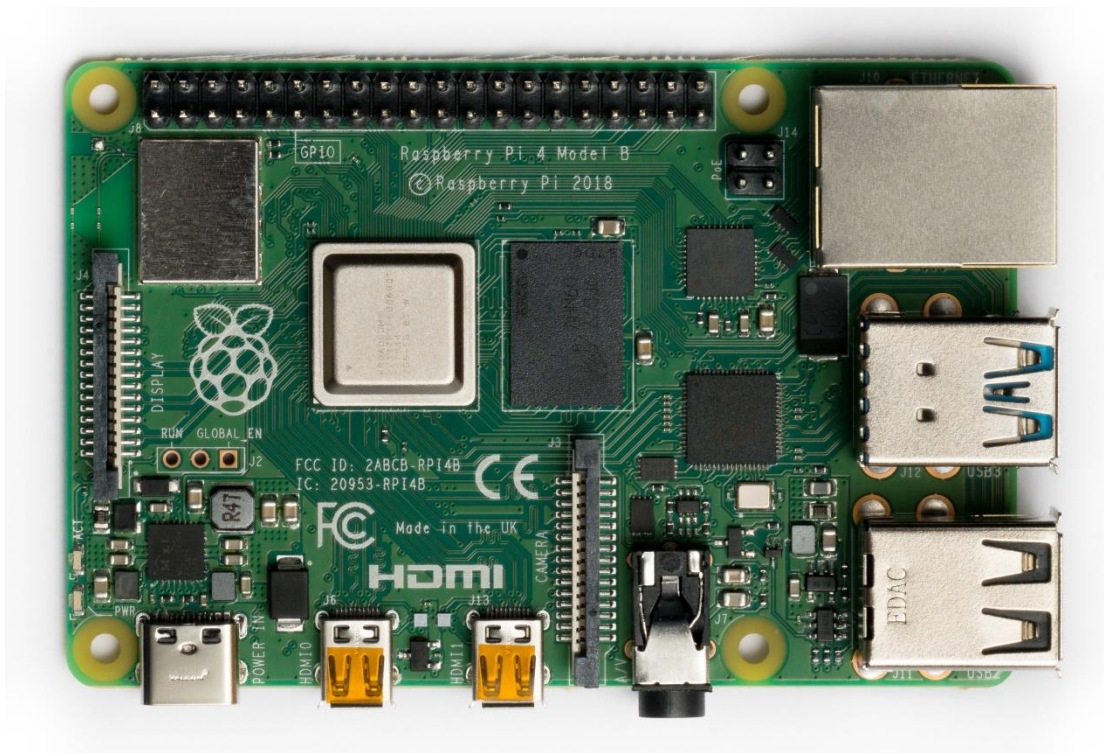
Επίσης, όσον αφορά τα εμπορικά προϊόντα έχουμε το Organelle όπου είναι ένα φορητό συνθεσάιζερ, ένας δειγματολήπτης, ένας sequencer και ένας επεξεργαστής εφέ που σχηματίστηκε και συναρμολογήθηκε από την Critter & Guitari. Συσσωματώνει μια μονάδα υπολογιστή Raspberry Pi με Linux [3]. Το OTTO είναι μια ψηφιακή φωτογραφική μηχανή που αναπτύχθηκε από την Next Thing Co και ενσωματώνει ένα Raspberry Pi Compute Module. Χρηματοδοτήθηκε με αναγνώριση από την εκστρατεία Kickstarter του Μαΐου 2014. Το Slice είναι ένα ψηφιακό πρόγραμμα αναπαραγωγής πολυμέσων το οποίο κάνει χρήση ενός Compute Module ως την καρδιά του. Διακυβερνήθηκε από την εκστρατεία Kickstarter τον Αύγουστο του 2014. Το λογισμικό που επιτελείται στο Slice βασίζεται στο Kodi. Πολλά εμπορικά τερματικά υπολογιστών thin client κάνουν χρήση του Raspberry Pi [3].

### 3. Raspberry Pi: Υλικό, Λογισμικό και Τεχνολογίες

#### 3.1 Υλικό

Το Raspberry Pi είναι ένας μικρός υπολογιστής που το σχήμα του είναι όσο το μέγεθος μιας πιστωτικής κάρτας όπου έχει όλα του τα εξαρτήματα όπως είναι η μνήμη, ο επεξεργαστής καθώς και οι θύρες είναι τοποθετημένες πάνω σε μία μόνο κάρτα. Έτσι αυτού του είδους οι υπολογιστές λέγονται SBC(Single Board Computer).

Στην Εικόνα 2 φαίνεται το Raspberry Pi 4 της γενιάς Model B και περιγράφουμε τι περιέχει πάνω αυτή η συσκευή.



Εικόνα 2: Raspberry Pi 4 γενιάς Model B(Πηγή:

[https://en.wikipedia.org/wiki/Raspberry\\_Pi#/media/File:Raspberry\\_Pi\\_4\\_Model\\_B\\_-\\_Top.jpg](https://en.wikipedia.org/wiki/Raspberry_Pi#/media/File:Raspberry_Pi_4_Model_B_-_Top.jpg))

Το Raspberry Pi διαθέτει τέσσερις USB θύρες για να συνδέσουμε πληκτρολόγιο, ποντίκι και οτιδήποτε άλλο μπορούμε να συνδέσουμε στον σταθερό μας υπολογιστή μέσω USB. Έχει επιπλέον και 2 θύρες οι οποίες είναι τύπου USB 3 με γνώρισμα το μπλε χρώμα. Το Raspberry Pi έχει μια θύρα Ethernet για την συνδεσή μας στο Διαδίκτυο, δυο θύρες Micro Hdmi οι οποίες συνδέονται με την οθόνη ή με την τηλεόραση και μια σύνδεση για να τροφοδοτηθεί η πλατφόρμα όπου είναι τύπου USB-Type C. Οι προγενέστερες εκδόσεις διέθεταν για τροφοδοσία θύρα Micro-USB. Το Raspberry Pi δεν έχει κάποιο κουμπί για να το ενεργοποιήσουμε ή να το

απενεργοποιήσουμε, αλλά μόλις συνδέσουμε το τροφοδοτικό τότε το Raspberry Pi αρχίζει να λειτουργεί. Στη πλατφόρμα αυτή υπάρχει ακόμη μια θύρα για την έξοδο του ήχου η οποία έχει κάποιο ιδιαίτερο χαρακτηριστικό, δηλαδή ότι μπορεί να υποστηρίξει τετραπολικό βύσμα με την έξτρα σύνδεση με ονομασία Composite Video και αναφέρεται στην μετάδοση του αναλογικού σήματος βίντεο, που είναι ένα από τα πιο παλιά standard από την δεκαετία του 60'. Επίσης αυτός ήταν και ο πρωταρχικός τρόπος σύνδεσης του BBC Micro με την τηλεόραση. Μπορεί ακόμη να είναι ένα από τα standard ώσπου μέχρι και σήμερα κάποιες συσκευές διαθέτουν.

Άλλες συνδέσεις οι οποίες δεν είναι τόσο απλές, όπου μπορούμε να τις βρούμε πάνω στην συσκευή του Raspberry Pi είναι η σύνδεση DSI(Display Serial Interface) και CSI(Camera Serial Interface). Για τη σύνδεση DSI μπορούμε να συνδέσουμε ορισμένες lcd οθόνες ενώ για την σύνδεση CSI μπορούμε να συνδέσουμε ορισμένες κάμερες με κατάλληλη έξοδο όπως για παράδειγμα η Raspberry Pi Camera.

Μια σύνδεση που θα χρησιμοποιήσουμε είναι η GPIO η οποία απαρτίζεται από σαράντα ακροδέκτες με τους περισσότερους να είναι γενικής χρήσης εισόδου- εξόδου και με τους υπόλοιπους να αναφέρονται σε άλλες συνδέσεις όπως η τροφοδοσία, τα κανάλια I<sup>2</sup> C. Για τους συγκεκριμένους ακροδέκτες έχουν βγει αρκετά περιφερειακά που συνδέονται σαν καπέλο(HAT) πάνω στο Raspberry Pi και προσκομίζουν λειτουργίες όπως το Sense HAT που έχει διάφορους αισθητήρες. Στην πραγματικότητα το ακρωνύμιο HAT ανταποκρίνεται στο Hardware Attached on Top.

Το Raspberry Pi Sense HAT είναι τοποθετημένο στην κορυφή Raspberry Pi μέσω των 40 ακίδων GPIO (που παρέχουν τα δεδομένα και η διεπαφή ισχύος) για την ανάπτυξη ενός «Astro Pi». Το Sense Hat έχει αρκετούς αισθητήρες με ενσωματωμένο κύκλωμα που ενδέχεται να εφαρμόσουμε για πολλούς διαφορετικούς τύπους πειραμάτων, εφαρμογών , όπως επίσης και για παιχνίδια. Επιπλέον έχει και τις παρακάτω τεχνικές προδιαγραφές [6]:

- *Γυροσκόπιο*: Είναι αισθητήρας γωνιακού ρυθμού και ο ρυθμός αυτός βρίσκεται στα  $\pm 245/500 / 2000$  dfps. Ο αισθητήρας αυτός μετρά την ορμή και την περιστροφή.
- *Επιταχυνσιόμετρο*: Είναι αισθητήρας γραμμικής επιτάχυνσης και η επιτάχυνση βρίσκεται στα  $\pm 2/4/8/16$  g. Ο συγκεκριμένος αισθητήρας μετρά τις δυνάμεις της επιτάχυνσης και ενδέχεται να χρησιμοποιηθεί για να βρει την κατεύθυνση της βαρύτητας [6].
- *Μαγνητόμετρο*: Είναι ένας αισθητήρας ο οποίος μετρά το μαγνητικό πεδίο της Γης, σαν μια πυξίδα. Οι μετρήσεις που παίρνουμε βρίσκονται στα  $\pm 4/8/12/16$  gauss.
- *Βαρόμετρο*: Έχει απόλυτο εύρος από 260 μέχρι 1260 hPa (όπου η ακρίβεια εξαρτάται από τη θερμοκρασία και την πίεση,  $\pm 0,1$  hPa υπό κανονικές συνθήκες).

- *Αισθητήρας θερμοκρασίας:* Ο αισθητήρας αυτός δείχνει με ακρίβεια την θερμοκρασία η οποία κυμαίνεται μέχρι τους  $\pm 2$  ° C στο εύρος 0-65 ° C [6].
- *Αισθητήρας υγρασίας:* Δείχνει με ακρίβεια την υγρασία έως  $\pm 4,5\%$  στην περιοχή 20-80% rH. Σε άλλη περίπτωση δείχνει και στους  $\pm 0,5$  ° C σε εύρος 15-40 ° C.
- *Οθόνη LED Matrix 8x8.*
- *Joystick (μικρό χειριστήριο πέντε κουμπιών):* Έχουμε την ικανότητα να εντοπίσουμε πότε το χειριστήριο του Sense Hat πιέζεται, κρατείται και απελευθερώνεται σε πέντε ανόμοιες κατευθύνσεις: πάνω, κάτω, δεξιά, αριστερά και στο κέντρο. Το χειριστήριο Sense Hat ανταποκρίνεται στα τέσσερα πλήκτρα του δρομέα του πληκτρολογίου και το μεσαίο κλικ του χειριστηρίου ανταποκρίνεται στο πλήκτρο Enter. Αυτό δηλώνει ότι η χρήση του χειριστηρίου διαθέτει ακριβώς το ίδιο αποτέλεσμα με το πάτημα αυτών των πλήκτρων στο πληκτρολόγιο.

Το κεντρικό chip που είναι κατασκευασμένο από την εταιρία Broadcom το οποίο το χαρακτηρίζουμε ως System on Chip, επειδή εκτός από τον τετραπύρρηνο επεξεργαστή τεχνολογίας ARM, περιέχει και κύκλωμα γραφικών αλλά και άλλα εξαρτήματα που είναι αναγκαία για την λειτουργία του Raspberry Pi όπως ο ελεγκτής μνήμης. Το αμέσως επομένο αξιοσημείωτο chip το οποίο το παρατηρούμε είτε από κάτω για παράδειγμα στο Raspberry Pi 3 είτε δίπλα στο κεντρικό chip στο Raspberry Pi 4 είναι η μνήμη RAM που στις τελευταίες γενιές είναι 1GB, μολονότι στο Raspberry Pi 4 έχουμε ως προτίμηση 2,4 ή ακόμη και 8 GB RAM. Ειδικά στο πίσω μέρος της συσκευής υπάρχει και η υποδοχή Micro-Sd όπου θα φορτώσουμε το λειτουργικό μας σύστημα το οποίο είναι το Raspberry Pi Os. Πίσω από τις θύρες USB παρατηρούμε τον ελεγκτή τους. Ο συγκεκριμένος ελεγκτής πριν το Raspberry Pi 4 περιείχε και το Ethernet, του οποίου οι επιδόσεις είχαν μειωθεί από την ταχύτητα του USB. Παρόλα αυτά στο Raspberry Pi 4, η θύρα Ethernet ελέγχεται από ιδιαίτερο chip και μάλιστα Gigabit. Ανάμεσα στο Display DSI και στους ακρόδεκτες GPIO συναντάμε ότι το chip που αναλαμβάνει την ασύρματη σύνδεση του Raspberry και έχει WiFi αλλά και Bluetooth. Υπάρχουν και άλλα εξαρτήματα που υπάρχουν πάνω στην συσκευή Raspberry Pi όπως είναι τα 2 led. Δηλαδή το πρώτο led είναι για να ρυθμίσουμε ότι τροφοδοτείται με ρεύμα, ενώ το δεύτερο led αναβοσβήνει τις στιγμές που πραγματοποιούνται διεργασίες στη συσκευή μας, όπως το λαμπάκι του σκληρού μας δίσκου περίπου. Τέλος αυτό το μικρό chip ρυθμίζει και σταθεροποιεί την τάση του ρεύματος που αποδέχεται το Raspberry.

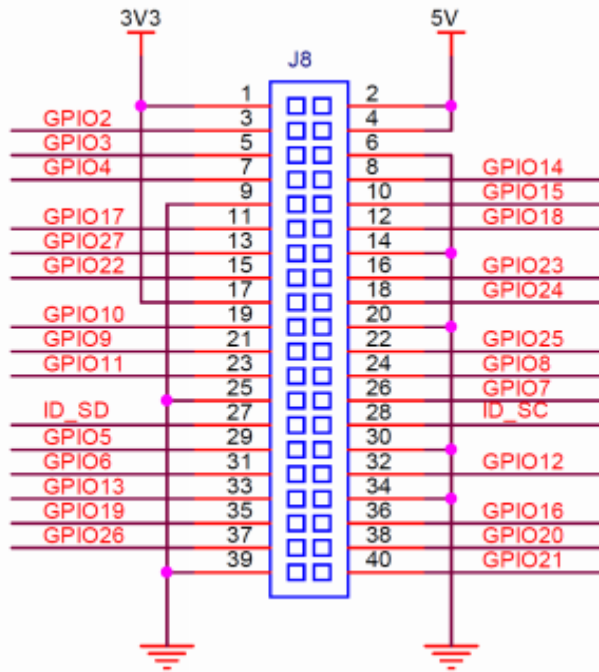
Σε αυτό το σημείο αναλύουμε κάποια από τα εξαρτήματα τα οποία είναι τοποθετημένα πάνω στη συσκευή του Raspberry Pi 4.

BCM2711(Broadcom processor για το Raspberry Pi 4 γενιάς Model B): Όσον αφορά για το Raspberry Pi 4 χρησιμοποιεί το chip της εταιρείας Broadcom και συγκεκριμένα το BCM2711 [4]. Η αρχιτεκτονική του BCM2711 είναι μια αξιοσημείωτη αναβάθμιση σε σχέση με αυτή που χρησιμοποίησαν τα SoCs σε

προγενέστερα ή πρώην μοντέλα Pi. Συνεχίζει τον τετραπύρηννο σχεδιασμό CPU του BCM2837, αλλά κάνει χρήση του πιο δυνατού πυρήνα ο οποίος είναι ο ARM A72 [4]. Έχει ένα πολύ βελτιωμένο σετ χαρακτηριστικών GPU με μια αρκετά πιο γρήγορη είσοδο ή έξοδο, εξαιτίας της ενσωμάτωσης ενός συνδέσμου PCIe που συνδέει τις θύρες USB 2 και USB 3 και έναν εγγενώς συνδεδεμένο Ethernet controller. Είναι επίσης κατάλληλο να διαχειριστεί περισσότερη μνήμη από τα SoCs που χρησιμοποιήθηκαν προηγουμένως. Αξίζει να σημειωθεί ότι οι πυρήνες ARM είναι ικανοί να λειτουργούν στο 1,5 GHz, κάνοντας το Pi 4 περίπου 50% πιο γρήγορο από το Raspberry Pi 3 της γενιάς B+. Η νέα μονάδα 3D VideoCore VI δουλεύει πλέον στα 500 MHz [4]. Οι πυρήνες ARM βρίσκονται στα 64-bit και ενώ το VideoCore βρίσκεται στα 32-bit, υπάρχει μια νέα μονάδα διαχείρισης μνήμης, που δείχνει ότι μπορεί να έχει πρόσβαση σε περισσότερη μνήμη από τις προγενέστερες εκδόσεις. Το BCM2711 chip συνεχίζει να κάνει χρήση της τεχνολογίας για την διάδοση θερμότητας που άρχισε με το BCM2837B0, το οποίο δίνει καλύτερη θερμική διαχείριση [4].

Απόδοση Επεξεργαστή: Το Raspberry Pi πρώτης γενιάς το οποίο λειτουργούσε στα 700 Mhz, αλλά έδινε τελικά πραγματική απόδοση περίπου ισοδύναμη με 0,041 GFLOPS. Σε επίπεδο CPU η απόδοση είναι παραπλήσια με 300 MHz Pentium II της περιόδου 1997–99 [3]. Η GPU δίνει 1 Gpixel/s ή 1,5 Gtexel/s επεξεργασίας γραφικών ή 24 GFLOPS γενικής χρήσης για υπολογιστική απόδοση. Ο επεξεργαστής 4x Cortex-A7 @ 900Mhz του Raspberry Pi χαρακτηρίστηκε ως 4-6 φορές πιο δυνατό από τον προκάτοχό του [3]. Η GPU ήταν αυτή η ίδια με το πρωτότυπο. Σε αυτό το σημείο ο επεξεργαστής 4x Cortex-A53 @ 1200Mhz(64/32-bit) χαρακτηρίζεται ότι έχει δέκα φορές την απόδοση ενός Raspberry Pi 1. Σημειώθηκε επίσης ότι το Raspberry Pi 3 ήταν περίπου 80% γρηγορότερο από το Raspberry Pi 2 για εργασίες που ήταν σχετικές με τον Parallel computing [3]. Ο επεξεργαστής 4x Cortex-A72 @ 1500Mhz(64/32-bit) του Raspberry Pi 4 χαρακτηρίζεται ότι έχει τρεις φορές την απόδοση ενός Raspberry Pi 3.

GPIO: Περιλαμβάνει 40 ακροδέκτες GPIO που βρίσκονται στην κορυφή της πλακέτας. Καθένα από αυτά παριστάνουν διαφορετικούς ακροδέκτες που χρησιμοποιούνται για επικοινωνία με την πλακέτα, για την τροφοδοσία της πλακέτας ή για την γείωση. Στην Εικόνα 3 φαίνονται οι 40 ακροδέκτες GPIO του Raspberry Pi 4 [9].



Εικόνα 3: Ακροδέκτες GPIO [9]

Στην Εικόνα 3, παρατηρούμε ότι τα slots με αριθμό 2 και 4 είναι στα 5V [9]. Οι ακροδέκτες GPIO μπορούν να χρησιμοποιηθούν ως απλοί είσοδοι και έξοδοι ελεγχόμενοι από λογισμικό και μπορούν να μετατραπούν (πολυπλεξία) σε διάφορες άλλες λειτουργίες που υποστηρίζονται από μοναδικά περιφερειακά μπλοκ όπως I2C, UART και SPI. Εκτός από τις χαρακτηριστικές περιφερειακές επιλογές που βρίσκονται στα Pi παλαιού τύπου, έξτρα περιφερειακά I2C, UART και SPI έχουν προστεθεί στο BCM2711 chip και είναι διαθέσιμα ως επιπλέον επιλογές πολυπλέκτη στο Pi 4 [9]. Αυτό παρέχει στους χρήστες πολύ μεγαλύτερη ευελιξία όταν συνδέουν επιπρόσθετο υλικό σε σύγκριση με τα προηγούμενα μοντέλα. Ωστόσο, οι ακροδέκτες ID\_SD και ID\_SC προορίζονται για το HAT ID EEPROM. Κατά την εκκίνηση, θα εξεταστεί η διεπαφή I2C για να αναζητήσει ένα EEPROM που προσδιορίζει την συνδεδεμένη πλακέτα και δέχεται την αυτόματη ρύθμιση των ακροδεκτών GPIO (και προαιρετικά, προγράμματα οδήγησης Linux). Επίσης δεν πρέπει να κάνουμε χρήση των συγκεκριμένων ακροδεκτών για οτιδήποτε άλλο εκτός από την σύνδεση ενός I2C ID EEPROM. Σε αυτήν την περίπτωση δεν πρέπει να συνδέουμε αυτούς τους ακροδέκτες εάν δεν απαιτείται ID EEPROM [9].

Δίκτυο: Τα μοντέλα A, A+ και Pi Zero δεν έχουν κύκλωμα Ethernet και κατά το πλείστον συνδέονται σε δίκτυο χρησιμοποιώντας ένα εξωτερικό προσαρμογέα USB Ethernet ή WiFi που δίνεται από τον χρήστη. Στα μοντέλα B και B+ η θύρα Ethernet δίνεται από έναν ενσωματωμένο προσαρμογέα USB Ethernet χρησιμοποιώντας το SMSC LAN9514 chip [3]. Τα Raspberry Pi 3 και Pi Zero W (ασύρματα) είναι εξοπλισμένα με 2,4 GHz WiFi 802.11n (150

Mbit / s) και Bluetooth 4.1 (24 Mbit / s) τα οποία βασίζονται στο Broadcom BCM43438 FullMAC chip χωρίς αξιόλογη υποστήριξη για την λειτουργία οθόνης και το Pi 3 έχει επίσης θύρα Ethernet 10/100 Mbit / s. Το Raspberry Pi 3 B+ έχει dual-band IEEE 802.11b / g / n / ac WiFi , Bluetooth 4.2 και Gigabit Ethernet (ελαττώνεται περίπου στα 300 Mbit / s από το USB 2.0 bus μεταξύ αυτού και του SoC). Το Raspberry Pi 4, όπως αναφέραμε παραπάνω έχει πλήρη gigabit Ethernet (η απόδοση του δεν περιορίζεται καθώς και δεν διοχετεύεται μέσω του USB chip.) [3]

Raspberry Pi Camera Module: Το Raspberry Pi Camera Module είναι επίσημο προϊόν του ιδρύματος Raspberry Pi. Το πρωτότυπο μοντέλο που ήταν αρχικά στα 5 megapixel κυκλοφόρησε το 2013 και επίσης κυκλοφόρησε ένα που ήταν στα 8-megapixel η Camera Module v2 το 2016. Και για τις δύο επαναλήψεις, υφίστανται εκδόσεις ορατού φωτός και υπέρυθρης ακτινοβολίας. Μια κάμερα υψηλού επιπέδου η οποία είναι στα 12 megapixel κυκλοφόρησε το 2020. Δεν υφίσταται έκδοση υπέρυθρη ακτινοβολίας της φωτογραφικής μηχανής HQ, ωστόσο το IR Filter μπορεί να αφαιρεθεί εάν απαιτείται [5].

### 3.2 Λογισμικό

Το λειτουργικό σύστημα που υποδεικνύεται είναι το Raspberry Pi Os, το οποίο είναι η πιο πρόσφατη έκδοση ενός λειτουργικού συστήματος γνωστού ως Linux. Το Linux αναπτύχθηκε από τον Φιλανδό Linus Torvalds το 1991, την εποχή που ήταν προπτυχιακός φοιτητής και το έδειξε ως πτυχιακή εργασία [12]. Είναι βασισμένο σε ένα άλλο περιζήτητο λειτουργικό, το UNIX, το οποίο αναπτύχθηκε 20 χρόνια πριν την δημιουργία του Linux, δηλαδή το 1971 και εφαρμόζεται ακόμη σε μεγάλα κέντρα δεδομένων. Πήρε το ονομά Linux από το όνομα του δημιουργού του, μολονότι το x στο τέλος δείχνει ότι είναι βασισμένο στο UNIX [12].

Εκτός από το λειτουργικό σύστημα Raspberry Pi Os, υπάρχουν και άλλα λειτουργικά συστήματα που μπορεί να εγκατασταθούν στο Raspberry Pi. Κάποια από αυτά είναι το Ubuntu Mate, Ubuntu Core και το Ubuntu Server δηλαδή τις τρεις διαφορετικές διανομές του Ubuntu και είναι από τις πιο γνωστές διανομές του Linux. Υπάρχουν, επίσης τα λειτουργικά συστήματα OSMC (Open Source Media Center) και LibreElec τα οποία συμπεριλαμβάνουν την εφαρμογή Kodi για εφαρμογή ως media player στην τηλεόραση. Ένα άλλο λειτουργικό σύστημα είναι το Mozilla WebThings το οποίο μας βοηθά να συνδέσουμε όλες τις έξυπνες που μπορεί να έχουμε στο σπίτι μας, ώστε να τις χειριζόμαστε ή να τις ελέγχουμε από μια κοινή πλατφόρμα [13]. Το Pinet είναι ένα σύστημα ιδανικό για αίθουσες διδασκαλίας όπου επιτρέπεται να έχουμε έναν κεντρικό υπολογιστή και όλοι οι σταθμοί εκπαίδευσης να είναι Raspberry Pi , αντί για κανονικούς υπολογιστές. Αυτό πρακτικά δείχνει ότι επιτρέπεται να μειώσουμε πολύ το κόστος μιας αίθουσας εκπαίδευσης, αφού το Raspberry Pi είναι πολύ φθηνότερο από έναν υπολογιστή και ξοδεύει πολύ λιγότερο ρεύμα. Επίσης το Risc Os είναι ένα



διαφορετικό λειτουργικό σύστημα το οποίο έχει δημιουργηθεί από την ομάδα που επινόησε τους επεξεργαστές ARM. Με άλλα λόγια τους επεξεργαστές που συναντάμε στο Raspberry Pi και στα έξυπνα κινητά. Μια άλλη διανομή είναι το weather station το οποίο είχε χρηματοδοτηθεί από την Oracle και περιείχε εξοπλισμό με σκοπό να στήσει κανείς ένα πλήρες μετεωρολογικό σταθμό. Με την χρηματοδότηση της Oracle το Raspberry Pi δώρισε σε χίλια σχολεία σε ολόκληρο τον κόσμο από ένα τέτοιο κιτ. Υπάρχει, επίσης και το Indigo jam ως ένα εργαλείο για την εκμάθηση της γλώσσας Basic με τις διαδραστικές δυνατότητες με τον έξω κόσμο.

Στο Raspberry Pi μπορούμε να χρησιμοποιήσουμε διάφορα περιβάλλοντα προγραμματισμού ή άλλα εργαλεία, για παράδειγμα Python, Scratch, Scratch 3, Sonic Pi, Mu και Thonny. Εμείς για να υλοποιήσουμε την εφαρμογή μας χρησιμοποιήσαμε την εφαρμογή Thonny η οποία είναι η γλώσσα προγραμματισμού Python και την εγκαθιστούμε μέσα στο περιβάλλον του λειτουργικού συστήματος Raspberry Pi Os.

### 3.3 Τεχνολογίες

Διερευνούμε μερικές από τις τεχνολογίες και χρήσεις όπου μπορεί να έχει μια τέτοια πλατφόρμα, όπως είναι το Raspberry Pi για την ψυχαγωγία μας στο σπίτι πέρα από τον κόσμο του IoT. Μια από τις πιο συνηθισμένες χρήσεις του, είναι να το χρησιμοποιήσουμε ως προσωπικό υπολογιστή. Ας σημειωθεί ότι η τελευταία γενιά του Raspberry δίνει καλύτερες επιδόσεις από τις προηγούμενες και μάλιστα μας προσφέρει την δυνατότητα να συνδέσουμε έως και δυο οθόνες, ανάλυσης μέχρι 4K ταυτόχρονα. Επίσης δεν γίνεται να συγκριθεί με έναν σύγχρονο σταθερό υπολογιστή μιας που ο επεξεργαστής του Raspberry Pi είναι πιο αργός. Οπότε μερικές εφαρμογές θα τρέχουν πιο αργά, αλλά για απλές εργασίες όπως πλοήγηση στο διαδίκτυο και επεξεργασία εγγράφων, τα καταφέρνει παρα πολύ καλά. Ωστόσο ο μικρός επεξεργαστής του Raspberry Pi, χρησιμοποιεί λιγότερη ενέργεια για την εφαρμογή του και δεν διαθέτει υψηλές απαιτήσεις στην ψύξη του. Έτσι με αυτόν τον τρόπο μπορούμε να βάλουμε τη συσκευή να λειτουργεί ολόκληρη τη μέρα χωρίς να έχει θορύβους και χωρίς να κάνει μεγάλη κατανάλωση.

Μια άλλη και ίσως πιο περιζήτητη χρήση του, είναι να δημιουργήσουμε το δικό μας σύστημα tv-box ή αλλιώς media center. Για να το πετύχουμε αυτό, μπορούμε να χρησιμοποιήσουμε κάποιες καλές και έτοιμες εφαρμογές οι οποίες έχουν αρκετές δυνατότητες όπως είναι το Kodi [3]. Το Kodi είναι μια εφαρμογή λογισμικού αναπαραγωγής πολυμέσων ανοιχτού κώδικα, το οποίο μπορούμε να το εγκαταστήσουμε στο λειτουργικό σύστημα ή σε διαφορετική περίπτωση μπορούμε να εγκαταστήσουμε το Librelec σε μια Micro-Sd το οποίο δεν είναι βαρύ λειτουργικό σύστημα και έτσι με αυτόν τον τρόπο έχουμε τα απολύτως αναγκαία για να δουλέψει το Kodi το οποίο και περιλαμβάνει [3]. Επίσης, θα συνδέσουμε το Raspberry Pi στην τηλεόρασή μας και θα είμαστε σε θέση να μπορούμε να παρακολουθήσουμε τις αγαπημένες μας σειρές και ταινίες.



Υπάρχουν HATS που κουμπώνουν πάνω στο Raspberry τα οποία παρέχουν καλής ποιότητας κυκλώματα μετατροπής ψηφιακού ήχου σε αναλογικό(τα αποκαλούμενα DAC) και μερικές φορές περιλαμβάνουν ενισχυτή. Όμως, μαζί με τη χρήση αντίστοιχων εφαρμογών όπως το Volumio, θα μπορούμε να δημιουργήσουμε το δικό μας ηχοσύστημα για να απολαμβάνουμε την αγαπημένη μας μουσική. Άρα, σύμφωνα με την ποιότητα του ψηφιακού αρχείου που κατέχουμε στη συλλογή μας μπορούμε να απολαύσουμε μουσική ακόμα και σε υψηλότερη ανάλυση από ένα μουσικό cd.

Για τα παιχνίδια τύπου Arcade ενδέχεται με την χρήση ειδικών λογισμικών να δημιουργήσουμε μια παιχνιδομηχανή ώστε να μπορούν οι χρήστες να παίξουν με εκατοντάδες κλασικά παιχνίδια. Παραδείγματος χάρη ένα από τα διαθέσιμα συστήματα, το RetroPie περιλαμβάνει πάνω από 50 εξομοιωτές παλαιότερων κονσολών και συστημάτων. Τέλος το Raspberry Pi έχει αρκετές δυνατότητες και εμείς αναφερθήκαμε σε μερικές από τις χρήσεις ή τις τεχνολογίες που έχει.

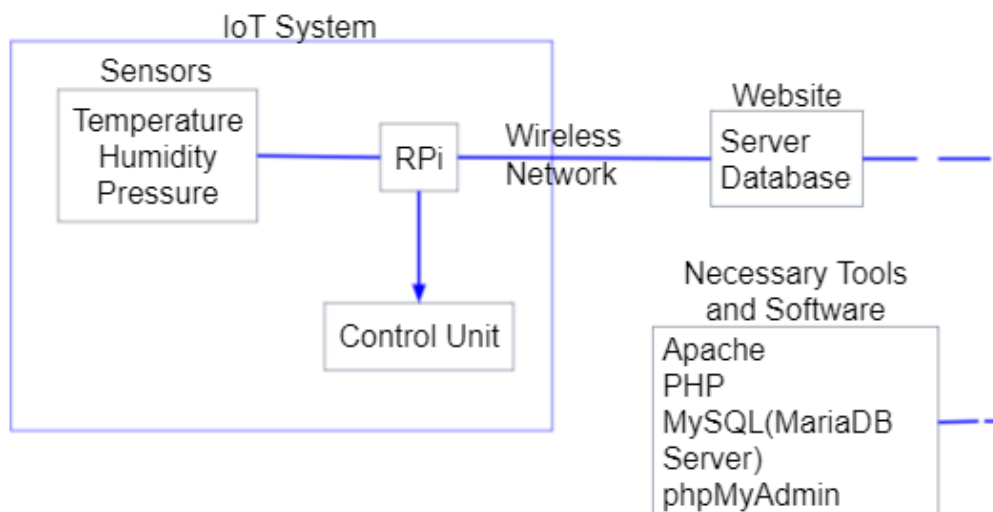
## 4. Εφαρμογή “έξυπνου χώρου αποθήκευσης”

### 4.1 Υλοποίηση Εφαρμογής “έξυπνου χώρου αποθήκευσης”

Η εφαρμογή αφορά τη δημιουργία ενός συστήματος παρακολούθησης των συνθηκών περιβάλλοντος ενός χώρου αποθήκευσης. Σύμφωνα με την νομοθεσία, η αποθήκευση των τροφίμων διέπεται από συγκεκριμένους κανόνες που αφορούν τις συνθήκες αποθήκευσης [17]. Για να το πετύχουμε αυτό στο συγκεκριμένο χώρο τοποθετούνται αισθητήρες οι οποίοι είναι χρήσιμοι για την παρακολούθηση των συνθηκών του περιβάλλοντος. Θα χρησιμοποιήσουμε συσκευές IoT, εργαλεία λογισμικού και τη γλώσσα προγραμματισμού Python. Οι αισθητήρες είναι βασικά μέρη του IoT συστήματος παρακολούθησης των συνθηκών περιβάλλοντος, όπως για παράδειγμα η θερμοκρασία, η υγρασία και η βαρομετρική πίεση. Επίσης, οι συνθήκες του περιβάλλοντος είναι κρίσιμος παράγοντας για οποιονδήποτε χώρο αποθήκευσης τροφίμων.

### 4.2 Σχεδίαση Συστήματος

Για την σχεδίαση του συστήματος, θα χρειαστούμε το IoT σύστημα το οποίο περιλαμβάνει μια μονάδα ελέγχου, τους αισθητήρες, δυνατότητα επικοινωνίας με το δίκτυο και παρουσίαση αποτελεσμάτων μιας Web εφαρμογής. Τα στοιχεία φαίνονται στην Εικόνα 4.



Εικόνα 4: Σύστημα παρακολούθησης IoT

### 4.3 Απαιτήσεις Εφαρμογής

Για την υλοποίηση της εφαρμογής θα λάβουμε υπόψη τις λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος.

Συγκεκριμένα οι **λειτουργικές απαιτήσεις** είναι:

- Συλλογή δεδομένων από το περιβάλλον που αφορούν την θερμοκρασία, την υγρασία και την βαρομετρική πίεση.
- Επικοινωνία δεδομένων
- Παρουσίαση δεδομένων
- Ενημερώσεις/προειδοποιήσεις/λήψη αποφάσεων
- Απομακρυσμένη διαχείριση

**Μη λειτουργικές απαιτήσεις** είναι:

- Διαθεσιμότητα
- Απόδοση
- Ευελιξία ή επεκτασιμότητα
- Ακεραιότητα
- Διαλειτουργικότητα
- Συντηρησιμότητα
- Μεταφερσιμότητα
- Αξιοπιστία
- Επαναχρησιμότητα
- Ευρωστία
- Ελεγχιμότητα
- Ευχρηστία

### Απαιτήσεις συστήματος

Για την υλοποίηση του προτεινόμενου συστήματος στηρίζομαστε στο Raspberry Pi. Για την συγκεκριμένη εργασία χρησιμοποιήσαμε το Raspberry Pi 4 της γενιάς Model B το οποίο έχει:

- Μνήμη RAM: 2 GB
- Επεξεργαστή 4x Cortex-A72 @ 1500Mhz(64/32-bit)
- 40 ακροδέκτες GPIO
- 2 θύρες για USB 2.0
- 2 θύρες για USB 3.0

Για την δική μας εφαρμογή χρησιμοποιήσαμε το μοντέλο Raspberry Pi 4 της γενιάς Model B το οποίο κυκλοφόρησε το 2019-2020 και το λειτουργικό σύστημα που εγκαταστήσαμε είναι το Raspberry Pi Os, το οποίο περιγράφεται στο παράρτημα Α. Οι κυριότεροι λόγοι που μας οδήγησαν στην επιλογή της συγκεκριμένης συσκευής είναι ότι έχει χαμηλή κατανάλωση και χαμηλό κόστος παραγωγής. Εξαιτίας της

χαμηλής κατανάλωσης στο ρεύμα δεν έχει ειδικές προϋποθέσεις στην ψύξη του επεξεργαστή, δηλαδή δεν είναι αναγκαίο να ψύχεται με χρήση ενεργής ψύξης όπως είναι οι ανεμιστήρες.

Στην εργασία μας χρησιμοποιούμε το Raspberry Pi 4 στο οποίο έχουν βελτιωθεί οι επιδόσεις του, οπότε αν κάνουμε αλόγιστη χρήση θα παρατηρήσουμε ότι η θερμοκρασία του επεξεργαστή μπορεί να φτάσει και τους 80 βαθμούς κελσίου. Όταν πλησιάσει επικίνδυνα σε αυτά τα συγκεκριμένα επίπεδα, τότε για την προστασία του επεξεργαστή και ολόκληρης της συσκευής, υπάρχει ένα σύστημα που ονομάζεται *throttling*. Με βάση αυτό το σύστημα θα ελλοιωθεί η ταχύτητα του επεξεργαστή και θα έχει την ικανότητα να επιστρέψει στην κανονική του θερμοκρασία. Η εγκατάσταση του λειτουργικού συστήματος και η διαμόρφωση της συσκευής φαίνονται στο Παράρτημα Α. Η συσκευή αυτή τροφοδοτείται με ρεύμα 5.1V και 15.3W για μέγιστη ισχύ εξόδου. Η πλατφόρμα Raspberry Pi 4 συνδέεται στο Internet μέσω ενός καλωδίου δικτύου, όπου η μια άκρη του καλωδίου την τοποθετούμε στο Router μας και την άλλη άκρη την βάζουμε στην πλατφόρμα Raspberry Pi 4. Χρησιμοποιούμε, επιπλέον το Sense Hat για να παίρνουμε τιμές από το περιβάλλον και το οποίο περιλαμβάνει αρκετούς αισθητήρες, όπως για παράδειγμα τη θερμοκρασία, την υγρασία και την βαρομετρική πίεση.

Επίσης θα χρειαστούμε διασύνδεση με το δίκτυο η οποία μπορεί να είναι ασύρματη ή ενσύρματη και για μια web εφαρμογή για την παρουσίαση και την ανάλυση των αποτελεσμάτων. Για την web εφαρμογή θα χρειαστούμε κατάλληλα εργαλεία και λογισμικό:

- Apache
- PHP
- MySQL(MariaDB Server)
- phpMyAdmin

#### 4.4 Συλλογή και διάθεση δεδομένων

Μέσα στα πλαίσια της εργασίας μας, θα χρειαστεί να συλλέγουμε δεδομένα από το περιβάλλον που αφορούν την θερμοκρασία, την υγρασία και την βαρομετρική πίεση. Για το σκοπό αυτό χρησιμοποιούμε το Sense Hat. Τα δεδομένα συλλέγονται καθόλη τη διάρκεια του εικοσιτετραώρου και για όλο το διάστημα στο οποίο υπάρχουν αποθηκευμένα τρόφιμα. Τα δεδομένα συλλέγονται ανά τακτά χρονικά διαστήματα, μεταφέρονται στο Raspberry Pi και με τη βοήθεια του δικτύου αποθηκεύονται σε μια διαδικτυακή Βάση δεδομένων. Εκεί είναι διαθέσιμα για επεξεργασία και ανάλυση.

## 4.5 Ανάπτυξη εφαρμογής

Η εφαρμογή που θα υλοποιήσουμε χρειάζεται δυο τμήματα και το καθένα υλοποιείται με κώδικα που έχουμε αναπτύξει στην Python. Το πρώτο τμήμα που θα επιτελέσουμε σχετίζεται με την ιστοσελίδα που θα φτιάξουμε και θα διαθέσουμε τα δεδομένα των αισθητήρων του Sense Hat στο τοπικό μας δίκτυο, αλλά και στο διαδίκτυο. Σε αυτόν τον κώδικα θα πρέπει να μετράμε την θερμοκρασία, την υγρασία και την βαρομετρική πίεση. Σύμφωνα με αυτά θα πρέπει να βάλουμε ένα μήνυμα όπου θα λέει ότι όλα πηγαίνουν καλά όταν οι συνθήκες του περιβάλλοντος είναι κατάλληλες αλλιώς σε αντίθετη περίπτωση δεν είναι όλα καλά. Αυτό συμβαίνει γιατί η εφαρμογή που θα υλοποιήσουμε έχει να κάνει με την μελέτη περιβάλλοντος ενός χώρου αποθήκευσης τροφίμων.

Επίσης το δεύτερο τμήμα που θα δημιουργήσουμε έχει σχέση με μια βάση δεδομένων όπου θα κρατάει τα στοιχεία για την θερμοκρασία, την υγρασία, την βαρομετρική πίεση και θα βάλουμε ακόμη μια παραπάνω μεταβλητή την time η οποία θα βγάζει σε πόσο χρόνο θα έχουμε καινούργιες τιμές για την θερμοκρασία, την υγρασία και την βαρομετρική πίεση. Εξηγούμε αναλυτικά τα δύο αυτά τμήματα δηλαδή τι ακριβώς κάνουν καθώς και τι υπάρχει και στα δυο ώστε να λειτουργούν ταυτόχρονα. Παρουσιάζουμε ποια προγράμματα εγκαταστήσαμε ώστε να δουλέψει η εφαρμογή μας καθώς επίσης θα περιγράψουμε την διαδικασία αναλυτικά με την οποία υλοποιήσαμε την εφαρμογή μας. Αναφερόμαστε στην απομακρυσμένη διαχείριση όπου είναι ο τρόπος με τον οποίο θα διαθέσουμε την εφαρμογή μας στο διαδίκτυο και αυτό θα γίνει μέσω του IP Forwarding που θα εξηγήσουμε πιο κάτω πως επιτυγχάνεται. Έτσι με αυτόν τον τρόπο έχουμε τα συγκεκριμένα δύο τμήματα, όπου το καθένα από αυτά τμήματα υλοποιείται σε γλώσσα Python και εγκαταστήσαμε το πρόγραμμα Thonny το οποίο είναι η γλώσσα Python στο περιβάλλον του Raspberry Pi Os.

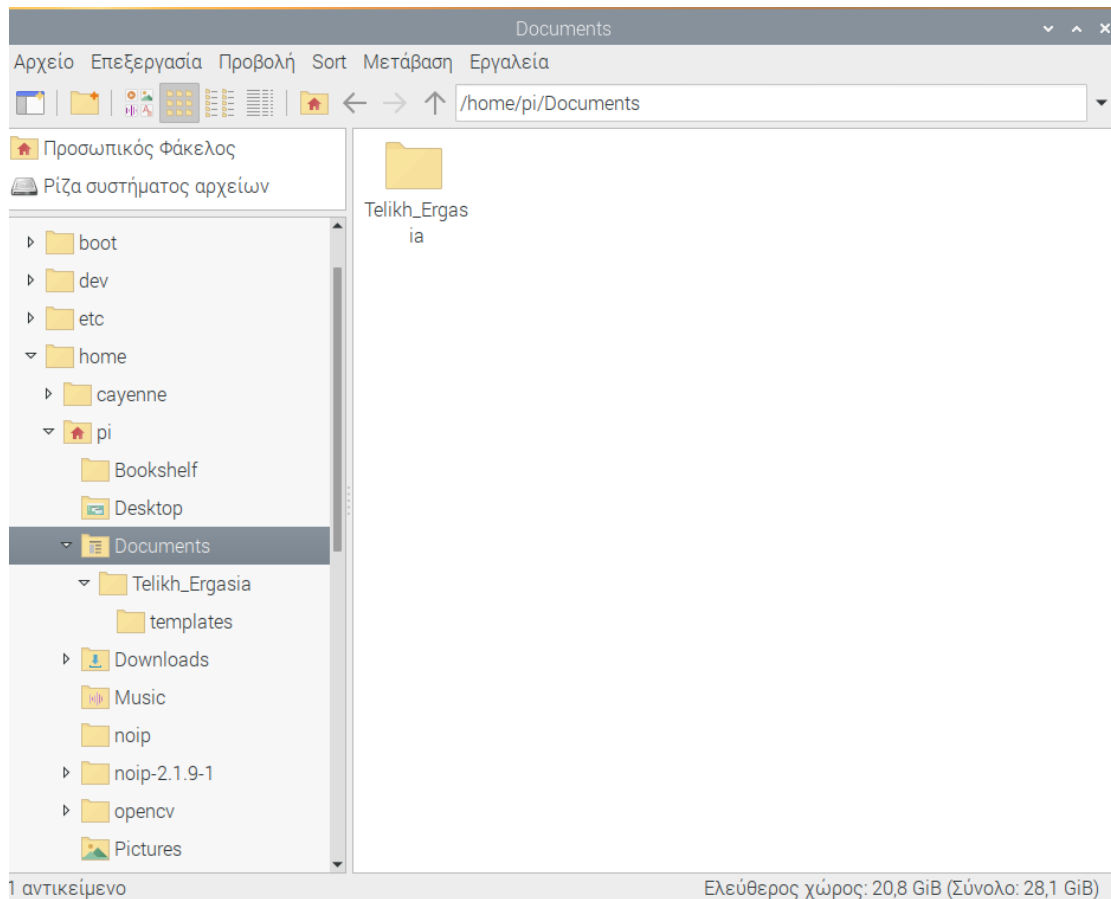
Όσον αφορά για την ανάπτυξη της εφαρμογής θα χρειαστούμε:

- Web εφαρμογή
- Βάση δεδομένων
- Flask, το οποίο περιγράφεται στο παράρτημα Δ.

Συνεχίζουμε με την εγκατάσταση του Apache και της PHP. Η διαδικασία εγκατάστασης περιγράφεται στο παράρτημα Β. Ακόμη θα χρειαστεί η εγκατάσταση και η παραμετροποίηση του MySQL και phpMyAdmin. Η διαδικασία περιγράφεται στο παράρτημα Γ. Περιγράφουμε, αναλυτικά την διαδικασία της υλοποίησης της εφαρμογής μας, όπου πρώτα δημιουργήσαμε έναν φάκελο με το όνομα της εργασίας μας και τον ονομάσαμε Telikh\_Ergasia. Για να φτιαχτεί ο φάκελος που φαίνεται στην Εικόνα 5 πληκτρολογήσαμε στο terminal τις εντολές:

```
cd ~Documents
```

```
mkdir Telikh_Ergasia
```



Εικόνα 5: Φάκελος δημιουργίας της εργασίας μας

Μέσα σε αυτόν τον φάκελο περιέχει τον φάκελο `templates` και τα δυο τμήματα κώδικα που υλοποιούνται σε γλώσσα Python όπου χρειαζόμαστε για την εφαρμογή μας η οποία φαίνεται στην Εικόνα 6. Ο υποφάκελος `templates` δημιουργείται από την παρακάτω εντολή:

```
cd ~/Documents/ Telikh_Ergasia
```

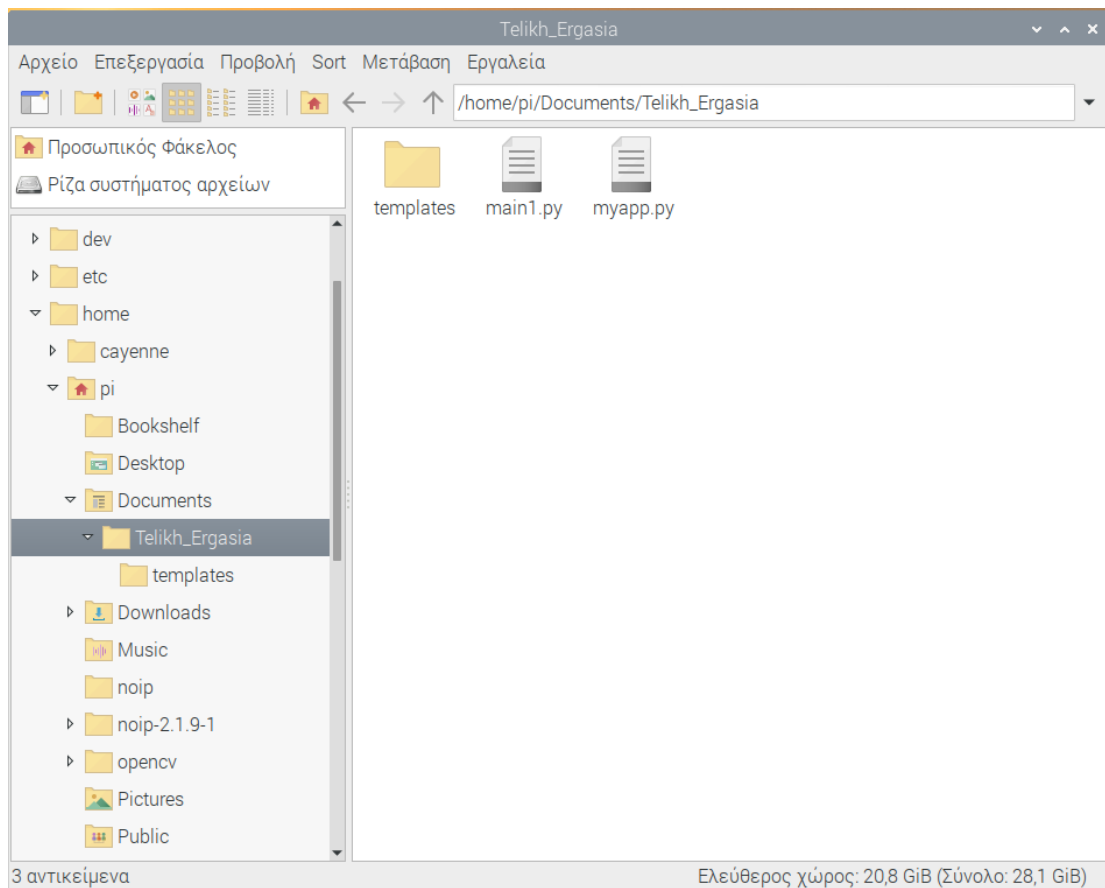
```
mkdir templates
```

Το πρώτο τμήμα κώδικα είναι για να διαθέσουμε την εφαρμογή μας στο διαδίκτυο και ονομάζεται `myapp.py`. Το δεύτερο τμήμα κώδικα αφορά για την βάση δεδομένων που θα χρησιμοποιήσουμε και θα αποθηκεύονται οι τιμές για την θερμοκρασία, την υγρασία και την βαρομετρική πίεση όπου αυτές οι τιμές θα διαθέτονται και αυτές στην ιστοσελίδα μας. Ο συγκεκριμένος κώδικας ονομάζεται `main1.py`. Για να δημιουργήσουμε τα δυο αυτά τμήματα και να μπορέσουμε να γράψουμε μέσα στα αρχεία, τότε τα βάζουμε στον φάκελο `Telikh_Ergasia` με τις εντολές:

```
cd ~/Documents/ Telikh_Ergasia
```

```
nano myapp.py
```

## nano main1.py



Εικόνα 6: Τα αρχεία που περιέχει ο φάκελος Telikh\_Ergasia

Σύμφωνα με την Εικόνα 7, ο φάκελος templates περιέχει μέσα τρία αρχεία HTML. Για να δημιουργήσουμε και να μπορέσουμε να γράψουμε μέσα στα αρχεία HTML, όπου τα αρχεία αυτά βρίσκονται μέσα στον φάκελο templates και γίνονται με τις παρακάτω εντολές:

```
cd ~/Documents/ Telikh_Ergasia/templates
```

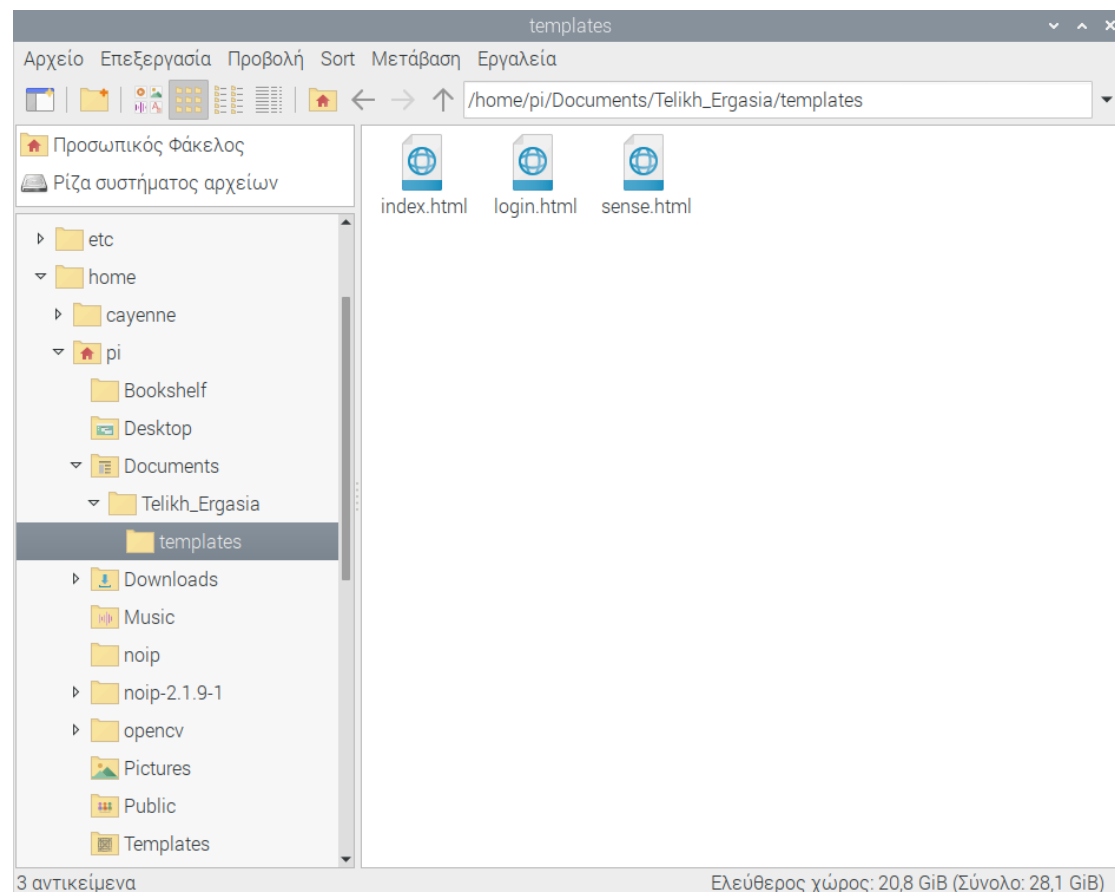
```
nano index.html
```

```
nano login.html
```

```
nano sense.html
```

Τα συγκεκριμένα αρχεία HTML τους έχουμε δώσει διαφορετική ονομασία στο καθένα ξεχωριστά όπου αυτά είναι το index.html, login.html, sense.html. Το αρχείο index.html είναι η σελίδα που επισκέπτεται ο χρήστης και αφού έχει συνδεθεί στη σελίδα μας θα του εμφανίσει επιπλέον το μήνυμα όπου θα καλωσορίζει τον χρήστη, τον σύνδεσμο για τα δεδομένα των αισθητήρων καθώς επίσης και τον σύνδεσμο αποσύνδεση για να αποσυνδεθεί από την εφαρμογή μας. Παρομοίως το αρχείο login.html είναι η σελίδα σύνδεσης όπου ο χρήστης θα μπει στη σελίδα με το όνομα χρήστη και τον κωδικό που έχει βάλει ο καθένας. Ο κωδικός καθώς και το όνομα

χρήστη τα βρίσκουμε στο τμήμα κώδικα της ιστοσελίδας μας, όπου εκεί προσθέτουμε τους χρήστες με την εντολή `append`. Όμως έχουμε και το αρχείο `sense.html` όπου περιέχει τα δεδομένα των αισθητήρων δηλαδή της θερμοκρασίας, της υγρασίας και της βαρομετρικής πίεσης. Σε επόμενη ενότητα θα δούμε τα συγκεκριμένα αρχεία HTML πως ενσωματώνονται στο τμήμα του κώδικα για την δημιουργία της ιστοσελίδας.



Εικόνα 7: Αρχεία HTML του φακέλου templates

Τέλος, έχουμε το κάθε αρχείο HTML να παρατηρήσουμε τι έχουμε γράψει και να εξηγήσουμε τι κάνει το καθένα. Όταν πληκτρολογήσουμε στο terminal την εντολή `nano index.html` αφού είμαστε στον φάκελο `templates` τότε στην οθόνη μας θα εμφανιστεί να επεξεργαστούμε ή να γράψουμε μέσα στο `index.html`. Στην Εικόνα 8 φαίνεται το Γειά σου `{{g.user[1]}}` όπου με το `{g.user[1]}` προβάλλουμε στη θέση αυτή το όνομα χρήστη που έχουμε αποθηκεύσει στο αντικείμενο `g.user`. Προσθέτουμε στην ακριβώς από κάτω γραμμή τον σύνδεσμο με τα δεδομένα των αισθητήρων και στην επόμενη γραμμή θα προσθέσουμε τον σύνδεσμο ΑΠΟΣΥΝΔΕΣΗ όπου όταν ο χρήστης επιλέξει τον συγκεκριμένο σύνδεσμο θα δρομολογηθεί στη σελίδα σύνδεσης. Κυριολεκτικά η σελίδα σύνδεσης θα εμφανιστεί



στον χρήστη όταν θα έχει εκτελεστεί η εντολή `session.pop('user_id', None)` η οποία σβήνει το session cookie και με αυτόν τον τρόπο χρήστης αποσυνδέεται.

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>HOME</title>
7
8 </head>
9 <body>
10     <p>Γεια σου {{g.user[1]}}</p>
11     <p><a href="/sense">Δεδομένα αισθητήρων</a></p>
12     <p><a href="/logme">ΑΠΟΣΥΝΔΕΣΗ</a></p>
13 </body>
14 </html>
15
16
```

Εικόνα 8: Αρχείο index.html

Το login.html όπως φαίνεται στην Εικόνα 9 είναι η σελίδα σύνδεσης όπου θα βάλουμε το όνομα χρήστη και τον κωδικό για να μπούμε στη σελίδα. Ωστόσο ανάμεσα στις ετικέτες body θα βρούμε μια φόρμα(form) η οποία περιέχει ένα πεδίο κειμένου με όνομα username για τη συμπλήρωση του ονόματος χρήστη καθώς και ένα πεδίο password. Στο πεδίο password είναι η παραμέτρος `type="password"` με την οποία καθορίζουμε ότι θα εμφανίζονται τελείες στη θέση των χαρακτήρων που πληκτρολογεί ο χρήστης. Τα ορίσματα placeholder εμφανίζουν τα κείμενα που έχουμε θέσει μέσα στα πεδία της φόρμας. Το τελικό αντικείμενο της φόρμας είναι το κουμπί ΕΙΣΟΔΟΣ το οποίο αναφέρεται με την ετικέτα button.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Σελίδα Σύνδεσης</title>
7 </head>
8 <body>
9     <form method="POST">
10         <input type="text" name="username" placeholder="Όνομα Χρήστη"><br/>
11         <input type="password" name="password" placeholder="Κωδικός"><br/>
12         <button type="submit">ΕΙΣΟΔΟΣ</button>
13     </form>
14 </body>
15 </html>
```

Εικόνα 9: Αρχείο login.html

Ενώ το αρχείο sense.html στην Εικόνα 10 είναι η σελίδα μετρήσεων που περιέχει τα δεδομένα των αισθητήρων, το Γεια σου {{g.user[1]}} όπου είπαμε τι κάνει ήδη στο αρχείο index.html, το μήνυμα εάν πάνε όλα καλά ή όχι και αυτό εξαρτάται από τις σωστές μετρήσεις θερμοκρασίας και υγρασίας που έχουμε ορίσει στον κωδικά μας. Όπως επίσης περιέχει έναν υπερσύνδεσμο τον οποίο αν τον πατήσουμε τότε θα μεταβούμε στην αρχική σελίδα στην οποία θα πρέπει να δώσουμε ξανά αν θέλουμε να μπούμε μέσα στη σελίδα το όνομα χρήστη καθώς και τον κωδικό. Ακόμη το αρχείο sense.html κάνει την αυτόματη ανανέωση της σελίδας ανά 5 δευτερόλεπτα.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <meta http-equiv="refresh" content="5">
7   <title>Σελίδα Μετρήσεων</title>
8 </head>
9 <body>
10  <p>Γεια σου {{g.user[1]}}</p>
11  <p>Θερμοκρασία: {{ roundv(g.s.temperature,2) }} βαθμοί Κελσίου</p>
12  <p>Υγρασία: {{ roundv(g.s.humidity,1) }} %</p>
13  <p>Βαρομετρική πίεση: {{ roundv(g.s.pressure,3) }} mbar</p>
14  <p>Μήνυμα: {{g.message}} </p>
15
16
17  <p><a href="/">Αρχική σελίδα</a></p>
18 </body>
19 </html>
20
```

Εικόνα 10: Αρχείο sense.html

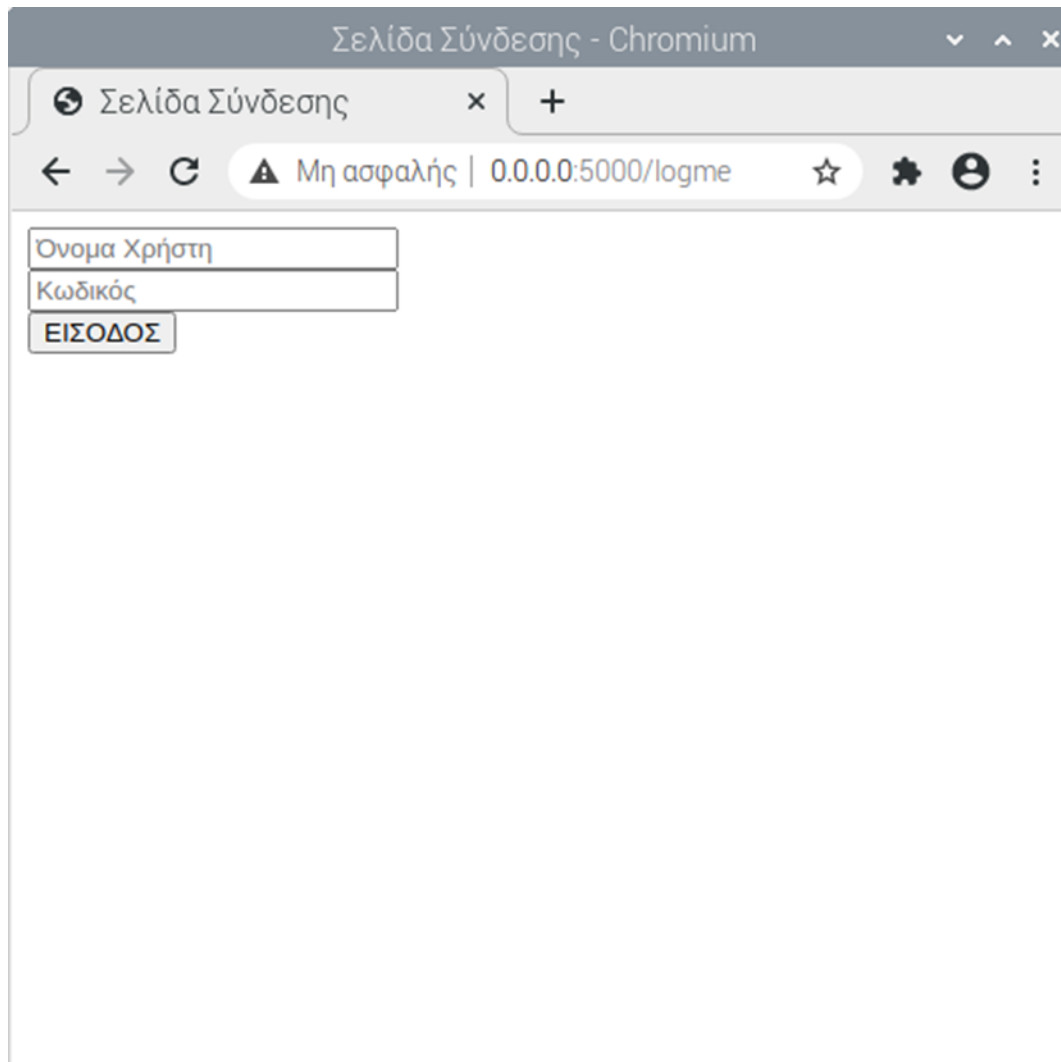
#### 4.6 Απομακρυσμένη διαχείριση

Μέχρι τώρα επισκεπτόμασταν την σελίδα μας μόνο μέσα από το περιβάλλον του Raspberry Pi Os. Δηλαδή ο χρήστης μπορούσε να συνδεθεί στην εφαρμογή μας εφόσον βρισκεται κόντα στην οικία μας. Κατόπιν όμως έχουμε προστατέψει την σελίδα με χρήση κωδικού, την οποία την περιγράφουμε στην επόμενη ενότητα που είναι η λειτουργία και η εκτέλεση της εφαρμογής μας και τώρα είμαστε έτοιμοι σε θέση να την διαθέσουμε στο Internet. Αρχικά θα εξετάσουμε πως μας δίνεται η δυνατότητα να επισκεφτούμε τη σελίδα από άλλους υπολογιστές του δικτύου της οικίας μας. Για να γίνει αυτό θα πρέπει να απενεργοποιήσουμε την εντολή `app.run(debug=True)` και θα γράψουμε την παρακάτω εντολή:

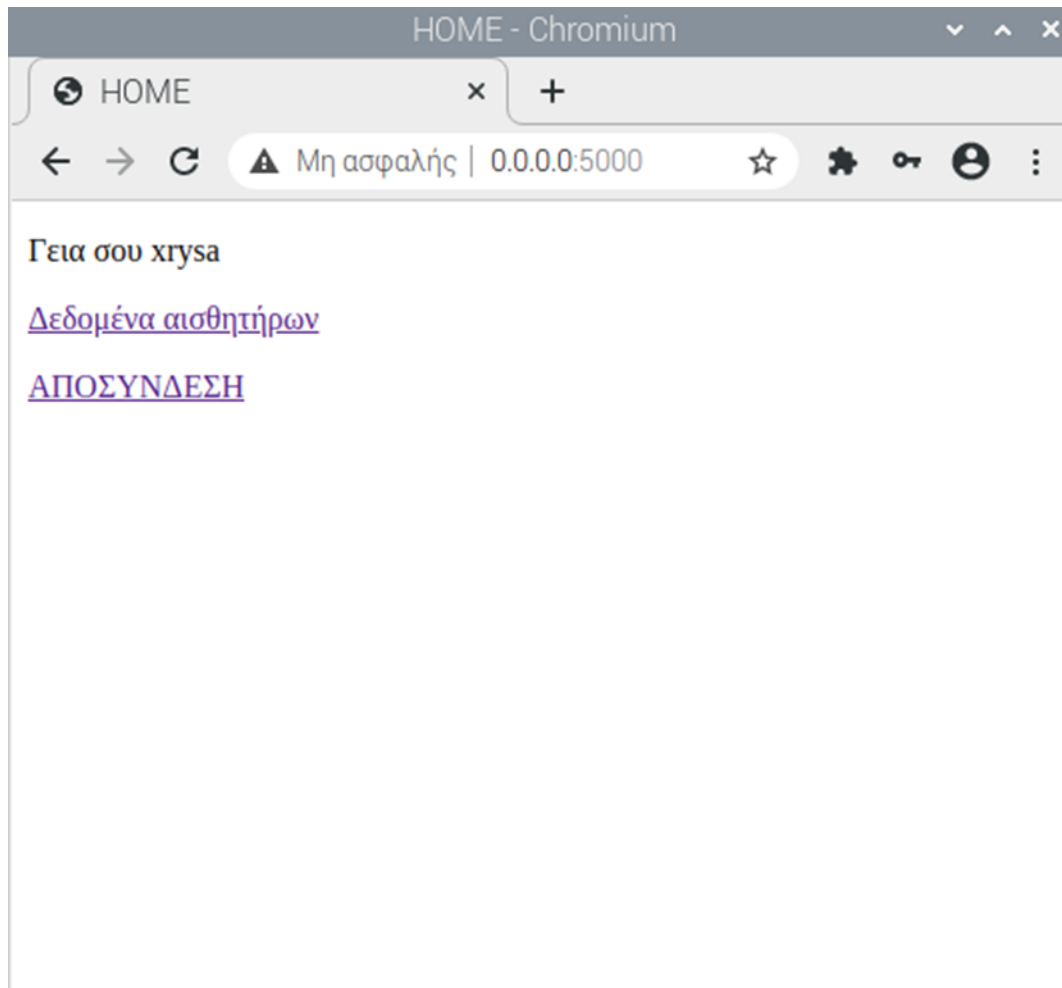
```
83
84 ▶ if __name__ == '__main__':
85     #app.run(debug=True)
86     app.run(debug=False,host='0.0.0.0')
87
```

Με την εντολή που φαίνεται στην παραπάνω εικόνα έχουμε πρωτίστως απενεργοποιήσει για λόγους ασφαλείας την αποσφαλμάτωση(Debug mode:off) και στο όρισμα έχουμε προσθέσει το '0.0.0.0' που πρακτικά σημαίνει ότι το Flask ενδέχεται να δεχτεί αιτήματα από οποιαδήποτε διεύθυνση. Η αλλαγή της συγκεκριμένης εντολής γίνεται στο αρχείο myapp.py όπου και αποθηκεύουμε το αρχείο. Αφού αποθηκεύσαμε το αρχείο θα κάνουμε εκτέλεση του κώδικα μας και θα πληκτρολογήσουμε πάλι τις παρακάτω εντολές και θα μας οδηγήσουν στο αποτέλεσμα που παρατηρούμε μπροστά μας:

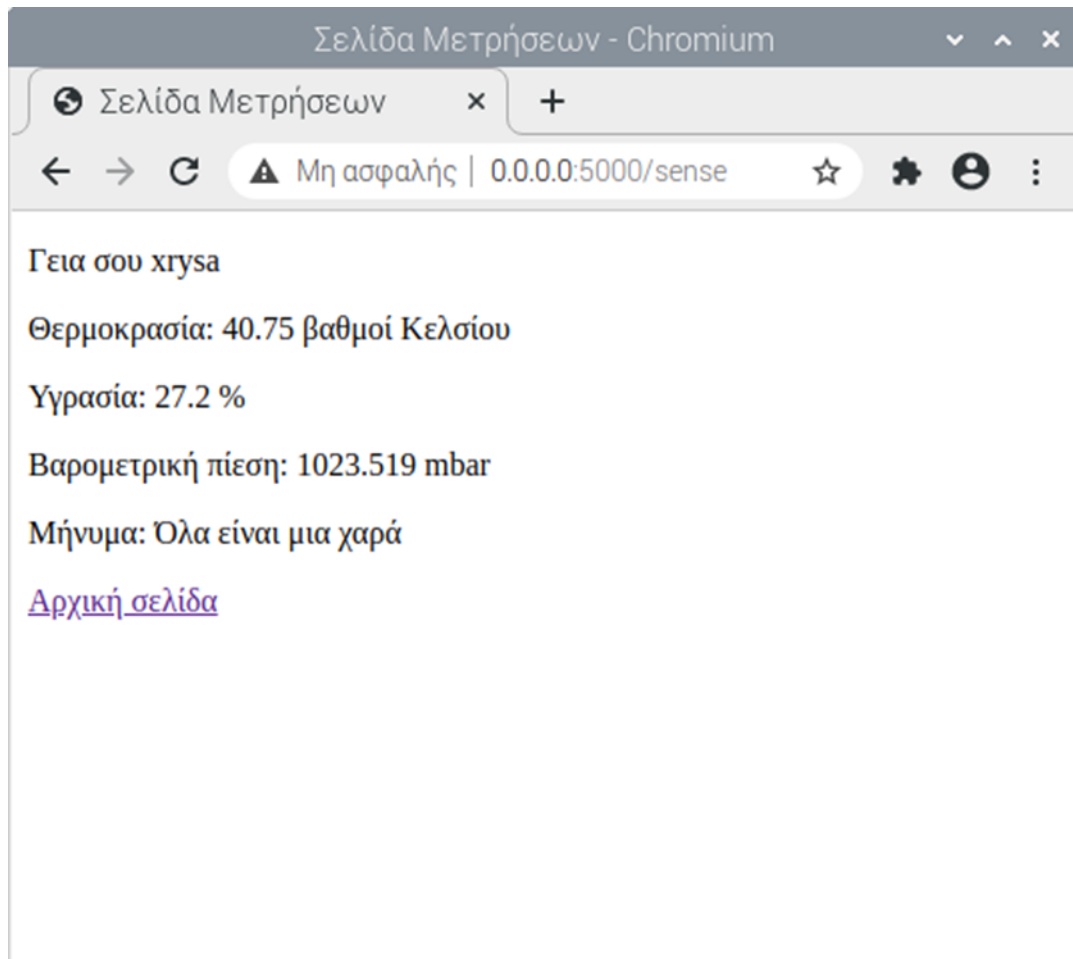
```
πi@raspberrypi4: ~/Documents/Telikh_Ergasia
Αρχείο Επεξεργασία Καρτέλες Βοήθεια
πi@raspberrypi4:~ $ cd ~/Documents/Telikh_Ergasia
πi@raspberrypi4:~/Documents/Telikh_Ergasia $ python3 myapp.py
* Serving Flask app "myapp" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [21/May/2021 15:20:05] "GET / HTTP/1.1" 302 -
127.0.0.1 - - [21/May/2021 15:20:05] "GET /logme HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:20:05] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [21/May/2021 15:21:00] "POST /logme HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:21:30] "POST /logme HTTP/1.1" 302 -
127.0.0.1 - - [21/May/2021 15:21:30] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:21:33] "GET /sense HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:21:35] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:21:36] "GET /logme HTTP/1.1" 200 -
^Cπi@raspberrypi4:~/Documents/Telikh_Ergasia $ scrot -d 10
```



Ωστόσο αφού είμαστε στη σελίδα σύνδεσης όπως δείχνει παραπάνω, τώρα ό χρήστης θα συμπληρώσει τα στοιχεία του και θα πατήσει το κουμπί ΕΙΣΟΔΟΣ για να μπει στη σελίδα μας. Αφού μπει στη σελίδα μας θα βρει δύο συνδέσμους και το μήνυμα για τον χρήστη που έχουμε αποθηκεύσει στο αντικείμενο `g.user` που φαίνονται παρακάτω:



Επίσης αν κλικάρουμε πάνω στον σύνδεσμο για τα Δεδομένα αισθητήρων θα εμφανιστεί:



Για να σταματήσουμε να τρέχει το Flask πρέπει να πατήσουμε ταυτόχρονα τα πλήκτρα **Ctrl-C**.

Για να μάθουμε ποια διεύθυνση έχει λάβει το Raspberry Pi Os από τον δρομολογητή (router) οικίας μας, εκτελούμε την παρακάτω εντολή στο terminal:

### **ifconfig**

Αυτή η εντολή μας επιστρέφει πληροφορίες για κάθε συσκευή που έχει το Raspberry Pi. Οι πληροφορίες που μας δίνει όταν εκτελέσουμε την εντολή `ifconfig` είναι ότι έχουμε τρεις παραγράφους το `eth0`, το `lo` και το `wlan0`, όπου εμείς χρειαζόμαστε την παράγραφο `wlan0` για να εξετάσουμε αν θα συνδέσουμε τη συσκευή μας είτε στο ασύρματο δίκτυο είτε στο ενσύρματο δίκτυο. Εμείς έχουμε συνδέσει το Raspberry Pi στο ενσύρματο δίκτυο με το καλώδιο δικτύου και θα αναζητήσουμε τη MAC Address της συσκευής μας στο πεδίο `eth0` που θα βρούμε στην παράγραφο `wlan0`. Επίσης στην παράγραφο `wlan0` θα βρούμε την διεύθυνση της συσκευής μας στο πεδίο `inet` στην περίπτωση που έχουμε σύνδεση τη συσκευή μας με το ασύρματο δίκτυο. Αφού τσεκάρουμε τη διεύθυνση IP της συσκευής μας τότε εκτελούμε την εφαρμογή μας. Οπότε είμαστε έτοιμοι να καλέσουμε τη σελίδα μας από οποιονδήποτε υπολογιστή ο οποίος βρίσκεται στο ίδιο δίκτυο με την συσκευή μας. Ανοίγουμε τον browser στο κινητό μας ή στον προσωπικό μας υπολογιστή και στη διεύθυνση γράφουμε:

**`http://η_διεύθυνση_της_συσκευής_μας:5000/logme`**

Σε διαφορετική περίπτωση μπορούμε να κάνουμε χρήση αντί της διεύθυνσης το όνομα της συσκευής μας, παραδείγματος χάρι:

**`http://raspberrypi4:5000/logme`**

Για να έχουμε πρόσβαση στην εφαρμογή μας, εκτός της οικίας και του δικτύου μας, θα πρέπει να ληφθεί μια σειρά από ρυθμίσεις που προϋποθέτουν να ξέρουμε τα στοιχεία πρόσβασης του δρομολογητή μας (router). Το πρώτο βήμα είναι να θέσουμε μια στατική διεύθυνση στο Raspberry Pi. Αυτό μπορεί να γίνει με πολλούς τρόπους, αλλά είναι προτιμότερο να κάνουμε χρήση για το σκοπό αυτό το δρομολογητή της οικίας μας. Ο δρομολογητής μέσω ενός εξυπηρετητή που διαθέτει (DHCP Server) παρέχει αυτόματα από μια διεύθυνση IP σε κάθε συσκευή που συνδέεται στο δίκτυό μας. Αυτές οι διευθύνσεις έχουν χρόνο ζωής που έχει καθοριστεί στο δρομολογητή μας, συνήθως από 24 έως 48 ώρες. Αυτό πρακτικά δηλώνει, πως αν έχουμε την συσκευή Raspberry Pi σβηστή για περισσότερο από το παραπάνω διάστημα, την επόμενη φορά που θα την ενεργοποιήσουμε θα έχει πάρει διαφορετική διεύθυνση. Υπάρχει, επιπλέον η δυνατότητα να δώσουμε εντολή στον δρομολογητή να παρέχει πάντοτε την ίδια διεύθυνση στη συσκευή μας. Για να το πετύχουμε αυτό θα χρειαστούμε και τη διεύθυνση MAC address η οποία είναι μοναδική για κάθε κάρτα δικτύου. Βρίσκουμε τη διεύθυνση με την εκτέλεση της εντολής `ifconfig` στα πεδία `inet` και `ether` της παραγράφου `wlan0`. Όταν μάθουμε την IP και MAC Address, συνδεόμαστε με τη βοήθεια του browser μας στον δρομολογητή μας. Για να μπούμε στο περιβάλλον διαχείρισης του δρομολογητή μας γίνεται όταν πληκτρολογήσουμε την διεύθυνση “192.168.1.254”. Αφού πλοηγηθούμε στη διεύθυνση του δρομολογητή μας, μας ζητά να συμπληρώσουμε τα δικά μας στοιχεία πρόσβασης. Άμα δώσουμε τα στοιχεία προσβάσης μας και συνδεθούμε με το router μας, θα αναζητήσουμε ένα εργαλείο με το όνομα DHCP binding, το οποίο συνήθως υπάρχει κάτω από το μενού LAN(Local Area Network).

Εκεί συμπληρώνουμε ένα όνομα που θέλουμε, καθώς επίσης την IP και MAC Address του Raspberry Pi Os που βρήκαμε με την βοήθεια του ifconfig. Αφού συμπληρώσουμε τα στοιχεία πατάμε “Apply” για να τα αποθηκεύσουμε. Πηγαίνουμε στο μενού “Internet”, στην ενότητα “Security”, όπου θα βρούμε την επιλογή “Port-Forwarding”. Αξίζει να σημειωθεί ότι το Port-Forwarding(προώθηση θύρας) επιτρέπει σε απομακρυσμένους υπολογιστές(παραδείγματος χάρι, υπολογιστές στο διαδίκτυο) οι οποίοι μπορούν να συνδεθούν με έναν συγκεκριμένο υπολογιστή μέσα σε ένα ιδιωτικό τοπικό δίκτυο.

Στα στοιχεία που πρέπει να συμπληρώσουμε συμπεριλαμβάνονται συνήθως ένα όνομα για να μη ξεχάμε τη ρύθμιση αυτή, το πρωτόκολλο επικοινωνίας (συνήθως αρκεί το TCP), η IP διεύθυνση της συσκευής, η δημόσια πόρτα με το όνομα WAN και η πόρτα συσκευής στην οποία τρέχει η υπηρεσία όπου θέλουμε να έχουμε πρόσβαση από το δημόσιο δίκτυο. Στη διεύθυνση IP(LAN Host) συμπληρώνουμε τη διεύθυνση της συσκευής μας, μολονότι στη θύρα της συσκευής(LAN Host Port) συμπληρώνουμε τον αριθμό της θύρας που τρέχει στο Flask, η οποία είναι το 5000. Στη δημόσια θύρα έχουμε την δυνατότητα να συμπληρώσουμε μια θύρα από το 1 έως το 65535, αν και είναι προτιμότερο να κάνουμε χρήση μιας θύρας η οποία βρίσκεται στο διάστημα από το 1024 έως το 65535. Στη παρακάτω εικόνα χρησιμοποιήθηκε η θύρα 14000.



Αποθηκεύουμε τις ρυθμίσεις που κάναμε και τώρα είμαστε σε θέση να δοκιμάσουμε να συνδεθούμε στην ιστοσελίδα μας από υπολογιστή εκτός δικτύου. Όμως πρώτα θα πρέπει να γνωρίζουμε ποια είναι η δημόσια IP διεύθυνσή μας. Αυτή μπορούμε να την βρούμε μέσα στο περιβάλλον του router, καθώς επίσης και σε κάποιες διαδικτυακές υπηρεσίες για παράδειγμα στη σελίδα <http://whatismyip.host/>

Σε αυτό το σημείο μπορούμε να πλοηγηθούμε στη σελίδα χρησιμοποιώντας ένα υπολογιστή εκτός του δικτύου της οικίας μας ή μέσω των δεδομένων του κινητού μας τηλεφώνου, διαμορφώνοντας τη διεύθυνση σύμφωνα με τη δημόσια IP που γνωρίσαμε από την παραπάνω ιστοσελίδα και τη θύρα που καθορίσαμε στο Port-Forwarding:

**<http://xxx.xxx.xxx.xxx:14000/logme>**

Η παραπάνω διεύθυνση μπορεί να αλλάζει κάθε 2-3 μέρες ή κάθε φορά που κάνουμε επανεκκίνηση του δρομολογητή μας. Στην περίπτωση αυτή θα πρέπει να αλλάξουμε την IP διεύθυνση και την αντίστοιχη MAC ώστε να κάνουμε τις απαραίτητες αλλαγές στο router μας όπως αναφέραμε ήδη προηγουμένως. Μόλις το κάνουμε αυτό, τότε η εφαρμογή μας θα είναι έτοιμη να λειτουργήσει.

## 5. Λειτουργία και εκτέλεση εφαρμογής

Για να μπορέσουμε να μπούμε μέσα στο περιβάλλον του Raspberry Pi Os θα χρειαστούμε να συνδέσουμε κάποια εξαρτήματα στην συσκευή μας και μια κάρτα microSD για την εγγραφή του λειτουργικού συστήματος. Επομένως, τα εξαρτήματα που θα συνδέσουμε στο Raspberry Pi είναι:

- καλώδιο δικτύου, για ενσύρματο δίκτυο
- τροφοδοτικό
- καλώδιο HDMI
- Sense Hat, το οποίο το τοποθετούμε πάνω στο Raspberry Pi

Αφού συνδέσουμε τα εξαρτηματά στη συσκευή τότε έχουμε μπει στο περιβάλλον του Raspberry Pi Os. Επίσης, θα δημιουργήσουμε δυο κώδικες οι οποίοι ο ένας είναι για την ιστοσελίδα και ο άλλος είναι για την βάση μας, οι οποίοι περιγράφονται αναλυτικά στην επόμενη ενότητα. Σύμφωνα με αυτούς τους κώδικες θα εξετάσουμε πως εκτελείται και λειτουργεί η εφαρμογή μας.

### 5.1 Περιγραφή της λειτουργίας και εκτέλεσης εφαρμογής

Σε αυτό το σημείο περιγράφουμε αναλυτικά τα δυο τμήματα κώδικα που χρησιμοποιήσαμε καθώς και τη λειτουργία της εφαρμογή μας.

Πρώτα θα ξεκινήσουμε με το **πρώτο τμήμα του κώδικα για την ιστοσελίδα** το οποίο φαίνεται παρακάτω και θα αρχίσουμε να εξηγούμε τι ακριβώς κάνει.

```

1 import mysql.connector
2 from flask import Flask,render_template,request,session,g,url_for,redirect
3 from sense_hat import SenseHat
4 s=SenseHat()
5 s.clear()
6
7 mydb = mysql.connector.connect(
8     host="localhost",
9     user="root",
10    password="12345",
11    database = "ergasia1Database",
12    autocommit=True
13 )
14
15 users= []
16 users.append([1, 'Giannis','kodikos_Giannis'])
17 users.append([2, 'Maria','kodikos_Marias'])
18 users.append([3, 'xrysa','kodikos_xrysas'])
19
20 #Δίνουμε το όνομα app στο flask
21 app = Flask(__name__)
22 app.secret_key = "Μυστικό_Κλειδί_Κρυπτογράφησης"
23
24 def getLastData():
25     mycursor = mydb.cursor()
26     mycursor.execute("SELECT * FROM values3 order by date desc LIMIT 0, 1")
27     myresult = mycursor.fetchone()
28     print(myresult)
29     print(myresult[0])
30     print(myresult[1])
31     print(myresult[2])
32     print(myresult[3])
33     g.temperature=myresult[0]
34     g.humidity = myresult[1]
35     g.pressure = myresult[2]
36

```

```

37 @app.before_request
38 def before_request():
39     g.user = None
40     if 'user_id' in session:
41         for user in users:
42             if user[0] == session['user_id']:
43                 g.user = user
44                 g.s=s
45
46 @app.route('/')
47 def index():
48     if not g.user:
49         return redirect(url_for('logme'))
50     return render_template('index.html')
51
52 @app.route('/logme', methods=['POST', 'GET'])
53 def logme():
54     session.pop('user_id', None)
55     if request.method == 'POST':
56         username = request.form['username']
57         password = request.form['password']
58         for user in users:
59             if user[1] == username and user[2] == password:
60                 session['user_id'] = user[0]
61                 return redirect(url_for('index'))
62     return render_template('login.html')
63
64 @app.route('/sense')
65 def sense():
66     if not g.user:
67         return redirect(url_for('logme'))
68     if (10<=s.temperature<=18) and s.humidity>30:
69         g.message = 'Όλα είναι μια χαρά'
70         s.show_letter("p")
71     else:
72         g.message = 'Οι συνθήκες δεν είναι κατάλληλες'
73         s.show_letter("m")
74     return render_template('sense.html')

```

```

76 @app.context_processor
77 def a_processor():
78     def roundv(value, digits):
79         return round(value, digits)
80
81     return {'roundv': roundv}
82
83
84 if __name__ == '__main__':
85     #app.run(debug=True)
86     app.run(debug=False,host='0.0.0.0')
87

```

Στην αρχή πρέπει να εγκαταστήσουμε το mysql connector γιατί σε περίπτωση που δεν το κάνουμε όταν θα τρέξουμε τον κώδικα θα μας βγάλει σφάλμα ότι δεν έχουμε κάνει την συγκεκριμένη εγκατάσταση. Για να εγκαταστήσουμε το mysql connector πρέπει να ανοίξουμε το terminal και να γράψουμε την εξής εντολή:

**sudo apt-get -y install python3-mysql.connector**

Εκτός από την εγκατάσταση του mysql connector, στο αρχείο του κώδικα που το ονομάσαμε myapp.py, έχουμε να βάλουμε το Flask framework μαζί με μερικά από τα εργαλεία του τα οποία θα τα εξηγήσουμε και γίνεται με την εντολή:

```
from flask import Flask,render_template,request,session,g,url_for,redirect
```

Όπου για να διαθέσουμε τα παραπάνω αρχεία HTML μέσα στο προγράμμα μας τότε θέλουμε μία μέθοδο του Flask την render\_template. Επίσης στο Flask υπάρχει ένα άλλο αντικείμενο το οποίο ονομάζεται request και με αυτό ενδέχεται να διαβάσουμε δεδομένα που στέλνει ο επισκέπτης της σελίδας. Δηλαδή μας ενδιαφέρει να λάβουμε τα στοιχεία που θα βάλει ο χρήστης στη φόρμα εισόδου. Με τα άλλα εργαλεία που έχει το Flask δηλαδή τα αντικείμενα session, g, url\_for, redirect είναι για να διαθέσουμε στο Internet τα δεδομένα της συσκευής μας και πως μας επιτρέπεται να ενισχύσουμε την ασφάλεια με χρήση κωδικού. Έτσι, θα μπορούμε να καθορίσουμε τους χρήστες, οι οποίοι πληκτρολογώντας τον κωδικό τους, θα επιτρέπεται να έχουν πρόσβαση στις πληροφορίες που θέλουμε. Τα υπόλοιπα εργαλεία που έχει το Flask είναι τα εξής:

1. Το αντικείμενο session (συνεδρία) μας αφήνει να αποθηκεύσουμε πληροφορίες που έχουν σχέση με τον επισκέπτη, τις οποίες εκείνος φέρει σε κάθε κινήσή του μέσα στην ιστοσελίδα μας. Λειτουργεί με τον ίδιο τρόπο όπως τα cookies, με την διαφορά ότι είναι πιο ασφαλές γιατί χρησιμοποιεί κρυπτογράφηση.
2. Μέσα στο αντικείμενο g μπορούμε να βάλουμε μεταβλητές και άλλα αντικείμενα, τα οποία θα είναι προσπελάσιμα σε όλες τις σελίδες μας.
3. Η url\_for() παίρνει ως όρισμα το όνομα μιας συνάρτησης και επιστρέφει τη διεύθυνση που αυτή η συνάρτηση παριστάνει.
4. Τέλος, την redirect() η οποία δρομολογεί τον χρήστη στη διεύθυνση που θα καθορίσουμε ως παράμετρο.

Βάζουμε την βιβλιοθήκη για το Sense Hat και αυτό γίνεται με την εντολή:

```
from sense_hat import SenseHat
```

Εδώ να τονίσουμε ότι θα κάνουμε χρήση των κεφαλαίων γράμματων μόνο όπου χρειάζονται, γιατί η python όπως και το περιβάλλον του linux είναι case-sensitive. Στην επόμενη εντολή θα μπορούμε να δώσουμε παραδείγματος χάρη το όνομα s στη συσκευή μας:

```
s=SenseHat()
```

Οι δυο παραπάνω εντολές είναι αναγκαίο να περιέχονται στα προγράμματά μας αν κάνουμε οποιαδήποτε χρήση του Sense Hat.

Για να μπορούμε να διαγράψουμε τυχόν παλιές τιμές που έχει δεχθεί η συστοιχία LED με την εντολή **s.clear()**.

Βάζουμε τα στοιχεία δημιουργίας της βάσης μας τα οποία έχουμε πρώτα στο δεύτερο τμήμα του κώδικα της βάσης μας και υπάρχουν στο αρχείο main1.py. Στο δεύτερο τμήμα για τον κώδικα της βάσης έχουμε ολόκληρη την δημιουργία βάσης. Απλώς εδώ στο πρώτο τμήμα για τον κώδικα της ιστοσελίδας μεταφέρουμε τα συγκεκριμένα στοιχεία της βάσης μας όπως φαίνονται παρακάτω μαζί με την συνάρτηση την getLastData() για να ενώσουμε τα δυο τμήματα του κωδικά μας. Οπότε ο κώδικας για τα στοιχεία δημιουργίας της βάσης μας γίνεται ως εξής:

```
import mysql.connector  
  
mydb=mysql.connector.connect(  
  
    host="localhost",  
  
    user="root",  
  
    password="12345",  
  
    database="ergasia1Database",  
  
    autocommit=True  
  
)
```

Δημιουργούμε μια λίστα η οποία θα περιλαμβάνει μικρότερες λίστες με τα στοιχεία του κάθε χρήστη. Πρώτα θα βάλουμε την μεταβλητή users σε τύπο λίστας και αυτό γίνεται όταν χρησιμοποιούμε τα συμβολα []. Με την εντολή append προσθέτουμε κάθε χρήστη ως μια μικρότερη λίστα η οποία απαρτίζεται από τρία πεδία. Το πρώτο πεδίο έχει σχέση με τον μοναδικό αριθμό(user id), το δεύτερο πεδίο έχει σχέση με το όνομα του χρήστη και το τρίτο πεδίο έχει σχέση με τον κωδικό του χρήστη. Αυτά γίνονται με τις εντολές:

```
users=[]  
  
users.append([1,'Giannis','kodikos_Giannis'])  
  
users.append([2,'Maria','kodikos_Marias'])  
  
users.append([3,'xrysa','kodikos_xrysas'])
```

Όπου τα στοιχεία αυτά τα έχουμε αποθηκεύσει στη βάση μας που είπαμε παραπάνω τι γίνεται. Όπως παρατηρήσαμε προηγουμένως, για να λειτουργήσουν οι συνεδρίες, θα χρειαστούμε ένα μυστικό κλειδί κρυπτογράφησης. Θέτουμε ένα δικό μας με την εντολή:

```
app.secret_key="Μυστικό_Κλειδί_Κρυπτογράφησης"
```

Πληκτρολογούμε την παραπάνω εντολή αμέσως μετά αφού έχουμε δώσει το όνομα app στο Flask, δηλαδή κάτω από την γραμμή:

**app=Flask(\_name\_)**

Η συνάρτηση getLastData() διαβάζει την πιο πρόσφατη εγγραφή από τον πίνακα values3. Επίσης ταξινομεί τις εγγραφές του πίνακα κατά φθίνουσα σειρά, άρα η πιο πρόσφατη ημερομηνία θα είναι η πρώτη και στη συνέχεια με τη Limit να διαβάσει την πρώτη τιμή μόνο.

```
24 def getLastData():
25     mycursor = mydb.cursor()
26     mycursor.execute("SELECT * FROM values3 order by date desc LIMIT 0, 1")
27     myresult = mycursor.fetchone()
28     print(myresult)
29     print(myresult[0])
30     print(myresult[1])
31     print(myresult[2])
32     print(myresult[3])
33     g.temperature=myresult[0]
34     g.humidity = myresult[1]
35     g.pressure = myresult[2]
```

Εικόνα 11:Συνάρτηση getLastData()

Εφόσον πρόκειται, να φτιάξουμε συνεδρίες για τους χρήστες που συνδέονται στην σελίδα μας, θα χρησιμοποιήσουμε την συνάρτηση logme(). Στην Εικόνα 11 φαίνεται η συνάρτηση getLastData() την οποία θα εξηγήσουμε πως λειτουργεί. Όταν ένας χρήστης επισκεφτεί τη σελίδα login τότε βγάζουμε το τυχόν session cookie που έχει αποθηκευτεί στον υπολογιστή του χρήστη. Αυτό το πετυχαίνουμε με την εντολή session.pop('user\_id',None). Τα περιεχόμενα της πρώτης if θα εκτελεστούν όταν ο χρήστης πατήσει το κουμπί ΕΙΣΟΔΟΣ. Αποθηκεύουμε το username και το password από τον χρήστη που έβαλε τα στοιχεία του στην φόρμα. Διατρέχουμε τη λίστα των χρηστών για να τσεκάρουμε αν υπάρχει χρήστης σύμφωνα με τα στοιχεία που λάβαμε από τη φόρμα. Σε περίπτωση που ο χρήστης έχει συμπληρώσει σωστά τα στοιχεία του στη φόρμα, τότε φτιάχνουμε ένα session cookie με όνομα user\_id και αποθηκεύουμε εκεί το μοναδικό αριθμό που αντιστοιχεί στον χρήστη. Έτσι, με αυτόν τον τρόπο δρομολογούμε το χρήστη στην αρχική σελίδα όπου θα είναι διαθέσιμοι οι σύνδεσμοι της αποσύνδεσης του χρήστη και των δεδομένων των αισθητήρων, καθώς επίσης θα περιλαμβάνει και το αντικείμενο g.user. Στην Εικόνα 12 φαίνεται η συνάρτηση logme().

```

51
52 @app.route('/logme', methods=['POST', 'GET'])
53 def logme():
54     session.pop('user_id', None)
55     if request.method == 'POST':
56         username = request.form['username']
57         password = request.form['password']
58         for user in users:
59             if user[1] == username and user[2] == password:
60                 session['user_id'] = user[0]
61                 return redirect(url_for('index'))
62     return render_template('login.html')
63

```

Εικόνα 12: Συνάρτηση logme()

Για να λειτουργήσουν σωστά οι συνεδρίες θα χρειαστεί να πραγματοποιήσουμε μια συνάρτηση που θα εκτελείται πριν την οποιαδήποτε αλλαγή σελίδας του χρήστη. Το Flask μας δίνει την δυνατότητα αυτή με την εντολή `@app.before_request` που εισάγεται πριν τη συνάρτηση. Αυτό φαίνεται στην Εικόνα 13, όπου είναι το κομμάτι κώδικα που έχουμε γράψει για την συνάρτηση `before_request()`.

```

37 @app.before_request
38 def before_request():
39     g.user = None
40     if 'user_id' in session:
41         for user in users:
42             if user[0] == session['user_id']:
43                 g.user = user
44                 g.s=s
45

```

Εικόνα 13: Συνάρτηση before\_request()

Στη συνάρτηση `before_request()`, βάζουμε πρώτα στο αντικείμενο `g.user` εκχωρούμε το `None`. Τσεκάρουμε αν είναι κατελιημένο το `session` cookie, με άλλα λόγια αν έχει συνδεθεί ο χρήστης στη σελίδα μας. Διατρέχουμε την λίστα των χρηστών, για να βρούμε σε ποιον ανήκει ο μοναδικός αριθμός που είναι αποθηκευμένος στο `session` cookie. Αποθηκεύουμε τη λίστα που έχει σχέση με τα στοιχεία του χρήστη στο `g.user`, ώστε να είναι προσπελάσιμα από όλες τις σελίδες. Με τον ίδιο τρόπο που κάνουμε γνωστά τα στοιχεία του χρήστη σε όλες τις σελίδες καθώς κάνουμε γνωστό και το αντικείμενο `s` ώστε να έχουμε πρόσβαση στα δεδομένα του Sense Hat.

Για να περιορίσουμε την πρόσβαση στην αρχική σελίδα η οποία δηλώνεται με το σύμβολο `/` ως όρισμα της `@app.route` τότε θα χρειαστεί να βάλουμε επιπλέον έναν έλεγχο στην αντίστοιχη συνάρτηση. Οπότε η συνάρτηση `index()` στην Εικόνα 14, θα γίνει:

```

45
46 @app.route('/')
47 def index():
48     if not g.user:
49         return redirect(url_for('logme'))
50     return render_template('index.html')
51

```



Εικόνα 14: Συνάρτηση index()

Με την if που προσθήσαμε στη συνάρτηση index(), τσεκάρουμε αν δεν έχει συνδεθεί κάποιος χρήστης. Με άλλα λόγια το g.user είναι άδειο. Για την συγκεκριμένη κατάσταση εκτελείται η εντολή redirect(), η οποία τον μεταφέρει στη φόρμα εγγραφής.

Στη συνάρτηση sense(), στην πρώτη if που έχουμε προσθέσει, τσεκάρουμε αν δεν έχει συνδεθεί κάποιος χρήστης. Με άλλα λόγια το g.user είναι άδειο. Για αυτή την περίπτωση θα πρέπει να κάνουμε ξανά έλεγχο αν η θερμοκρασία και η υγρασία που έχουμε βάλει είναι μέσα στις επιτρεπόμενες τιμές που τοποθετήσαμε στον κώδικα. Αν είναι μέσα στις επιτρεπόμενες που έχουν οριστεί στον κώδικα τότε θα βγάλει το μήνυμα ότι «Όλα είναι μια χαρά». Αυτή η πληροφορία θα εμφανιστεί στο Sense Hat με το γράμμα p μέσω της s.show\_letter("p"). Αλλιώς σε διαφορετική περίπτωση μιας που δεν είναι μέσα στις επιτρεπόμενες τιμές της θερμοκρασίας και της υγρασίας που έχουμε ορίσει τότε βγάζει το μήνυμα «Οι συνθήκες δεν είναι κατάλληλες». Η συγκεκριμένη πληροφορία θα εμφανιστεί επίσης και αυτή στο Sense Hat με άλλο γράμμα το οποίο είναι το m, όπου αυτό γίνεται μέσω της s.show\_letter("m"). Με την συνάρτηση s.show\_letter() δίνουμε οδηγία στη βιβλιοθήκη του Sense Hat να προβληθεί ένας χαρακτήρας στην συστοιχία LED. Το συγκεκριμένο κομμάτι κώδικα φαίνεται στην Εικόνα 15.

```
64     @app.route('/sense')
65     def sense():
66         if not g.user:
67             return redirect(url_for('logme'))
68         if (10<=s.temperature<=18) and s.humidity>30:
69             g.message = 'Όλα είναι μια χαρά'
70             s.show_letter("p")
71         else:
72             g.message = 'Οι συνθήκες δεν είναι κατάλληλες'
73             s.show_letter("m")
74         return render_template('sense.html')
```

Εικόνα 15: Συνάρτηση sense()

Συνεπώς αν θέλησουμε να στρογγυλοποιήσουμε τις τιμές που εμφανίζει η σελίδα /sense, μπορούμε να δημιουργήσουμε μια συνάρτηση που θα διαθέτει τη λειτουργία αυτή και με το εργαλείο @app.context\_processor να την κάνουμε διαθέσιμη μέσα στα templates. Όποτε η συγκεκριμένη συνάρτηση είναι:

```
76     @app.context_processor
77     def a_processor():
78         def roundv(value, digits):
79             return round(value, digits)
80
81         return {'roundv': roundv}
82
83
```

Εικόνα 16: Συνάρτηση για στρογγυλοποίηση αριθμών

Για την συνάρτηση στην Εικόνα 16, αρχικά δημιουργούμε μια βασική συνάρτηση η οποία είναι η `a_processor`. Μέσα στη βασική συνάρτηση πραγματοποιούμε άλλη μια συνάρτηση με όνομα `roundn()` η οποία δέχεται δυο ορίσματα όπου το ένα όρισμα είναι η τιμή και το άλλο όρισμα είναι το πλήθος των δεκαδικών ψηφίων. Με τα συγκεκριμένα ορίσματα τα περνάμε στη συνάρτηση `round()` της `python`, η οποία μας εκτελεί την απαιτούμενη εργασία. Τέλος, η βασική συνάρτηση επιστρέφει υπό μορφή λεξικού το όνομα της συνάρτησης `roundn()` που πραγματοποιήσαμε, για να είναι διαθέσιμη στα `templates`. Για να κάνουμε χρήση της συνάρτησης `roundn()` που έχουμε φτιάξει το συναντάμε μέσα στο αρχείο `sense.html` και είναι οι παρακάτω γραμμές:

```
10 <p>Θερμοκρασία: {{ roundn(g.s.temperature,2) }} βαθμοί Κελσίου</p>
11 <p>Υγρασία: {{ roundn(g.s.humidity,1) }} %</p>
12 <p>Βαρομετρική πίεση: {{ roundn(g.s.pressure,3) }} mbar</p>
```

Για τις παραπάνω γραμμές θα εξασφαλίσουμε ακρίβεια δυο δεκαδικών ψηφίων για την θερμοκρασία, ένα δεκαδικό ψηφίο για την υγρασία και τρία δεκαδικά ψηφία για την βαρομετρική πίεση.

Για το τέλος του συγκεκριμένου κώδικα προκειμένου να εκτελεστεί η εφαρμογή μας ως διαδικτυακή εφαρμογή με χρήση του `Flask`, γίνεται όταν γράψουμε στην `python` την εξής εντολή:

```
88 ▶ if __name__ == '__main__':
89     app.run(debug=True)
90
```

Η παραπάνω `if` τσεκάρει με την μεταβλητή `_name_` αν εμείς εκτελέσαμε την εφαρμογή και όχι κάποιο πρόγραμμα της `python`, εφόσον πρόκειται να ενεργοποιήσει την μέθοδο `app.run` που περιλαμβάνει. Η `app.run` ενεργοποιεί το `Flask` και τον εξυπηρετητή `web`, που περιλαμβάνει το `Flask framework`. Επί τούτου έχουμε θέσει την παράμετρο εντοπισμού σφαλμάτων `debug=True` σαν όρισμα στην `app.run()`. Έτσι με αυτόν τον τρόπο θα μας εμφανίσει πιθανά σφάλματα στην ιστοσελίδα, βοηθώντας μας να τα περιορίσουμε. Όμως, σε ένα περιβάλλον παραγωγής, θα θέλαμε να το προσαρμόσουμε σε `False` για να αποφύγουμε τυχόν προβλήματα ασφάλειας.

Συνεχίζουμε με **το δεύτερο τμήμα για τον κώδικα της βάσης** μας το οποίο θα το εξηγήσουμε και θα το αναλύσουμε λεπτομερώς.

```

2 import mysql.connector
3 import time
4 from sense_hat import SenseHat
5 s = SenseHat()
6 s.clear()
7
8 mydb = mysql.connector.connect(
9     host="localhost",
10    user="root",
11    password="12345",
12    database="ergasia1Database"
13 )
14
15
16 def print_hi(name):
17     # Use a breakpoint in the code line below to debug your script.
18     print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
19
20
21 def createDatabase():
22     mycursor = mydb.cursor()
23     mycursor.execute("CREATE DATABASE ergasia1Database")
24     print_hi('Created')
25
26
27 def createTable():
28     mycursor = mydb.cursor()
29     mycursor.execute("CREATE TABLE values3 (temperature double(40,2), humidity double(40,2), pressure double(40, 2), date double(40, 7)) ")
30     print_hi('Table')
31
32
33 def insertValueToDatabase(temperature, humidity, pressure, date):
34     mycursor = mydb.cursor()
35     sql = "INSERT INTO values3 (temperature, humidity, pressure, date) VALUES (% s, % s, % s, % s)"
36     val = (temperature, humidity, pressure, date)
37     mycursor.execute(sql, val)
38     mydb.commit()
39     print("1 record inserted, ID:", mycursor.lastrowid)
40
41

```

```

42 def getLastData():
43     mycursor = mydb.cursor()
44     mycursor.execute("SELECT * FROM values3")
45     myresult = mycursor.fetchone()
46     print(myresult)
47     print(myresult[0])
48     print(myresult[1])
49     print(myresult[2])
50     print(myresult[3])
51
52
53 def writeValuesLoop():
54     lastUpdateTime = time.time()
55
56     while True:
57         print(lastUpdateTime)
58         if (time.time() > lastUpdateTime + 300):
59             print(time.time() - lastUpdateTime)
60             lastUpdateTime = time.time()
61             temperature = s.temperature
62             humidity = s.humidity
63             pressure = s.pressure
64             print(lastUpdateTime)
65             insertValueToDatabase(temperature, humidity, pressure, lastUpdateTime)
66
67
68 if __name__ == '__main__':
69     print_hi('PyCharm')
70
71 # createDatabase()
72 # insertValueToDatabase(13.5, 14.9, 9.8, 3.1)
73 # getLastData()
74 # createTable()
75 writeValuesLoop()
76

```

Ξεκινάμε πρώτα με την δημιουργία της βάσης μας η οποία γίνεται ως εξής:

```
import mysql.connector  
  
mydb=mysql.connector.connect  
  
    host="localhost",  
  
    user="root",  
  
    password="12345",  
  
    database="ergasia1Database"  
  
)
```

Σε αυτό το σημείο, θα χρειαστεί να εγκαταστήσουμε ξανά το `mysql.connector` που ανήκει στη βιβλιοθήκη `import mysql.connector` και γίνεται με την εξής εντολή γράφοντας στο terminal:

```
sudo apt-get -y install python3-mysql.connector
```

Υπάρχει, επίσης η συνάρτηση `print_hi(name)` η οποία χρησιμοποιεί ένα breakpoint στην γραμμή κώδικα που το τοποθετούμε και το παρατηρούμε στον εντοπισμό σφαλμάτων ώστε να μπορέσουμε να διορθώσουμε το script μας. Αυτό φαίνεται παρακάτω:

```
def print_hi(name):  
  
    print(f 'Hi, {name}')
```

Επίσης υπάρχει η βιβλιοθήκη `import time` η οποία χρησιμοποιείται για τον χρόνο τον όποιο χρειάζεται για να γίνει η καταγραφή των τιμών της θερμοκρασίας, της ατμοσφαιρικής πίεσης και της υγρασίας. Η συγκεκριμένη καταγραφή γίνεται στην συνάρτηση `writeValuesLoop` την οποία θα εξηγήσουμε παρακάτω.

Η βάση μας αποτελείται από τον host τον οποίο τον εκχωρούμε σε `localhost`, τον user τον εκχωρούμε σε `root`, τον κωδικό τον εκχωρούμε με το `12345` γιατί αυτό τον κωδικό χρησιμοποιήσαμε για να εγκαταστήσουμε την `MySql(MariaDB Server)`. Η βάση μας περιέχει και το όνομα που της δίνουμε το οποίο είναι «`ergasia1Database`», αλλά για να γίνει αυτό θα πρέπει να δημιουργήσουμε τη συνάρτηση `createDatabase()` η οποία φαίνεται παρακάτω:

```
def createDatabase():  
  
    mycursor = mydb.cursor  
  
    mycursor.execute("CREATE DATABASE ergasia1Database")
```

```
print_hi("Created")
```

Για να μπορέσουμε να εκτελέσουμε τη συνάρτηση `createDatabase()` με την οποία δημιουργούμε το όνομα της βάσης μας θα πρέπει από τη βάση μας να μην έχουμε βάλει το όνομα της το οποίο είναι `database="ergasia1Database"` δηλαδή να έχουμε μόνο τα πεδία `host`, `user` και `password`. Επίσης για να την εκτελέσουμε πρέπει να καλέσουμε την συνάρτηση με το όνομα της και χωρίς ορίσματα. Οπότε καλούμε την `createDatabase()` αλλά καλώντας την θα μας βγάλει σφάλμα γιατί δεν θα έχει δημιουργηθεί ακόμη η βάση μας, επειδή τώρα θα πρέπει να βάλουμε το όνομα της μέσα στη βάση μας το οποίο είναι `database="ergasia1Database"`. Την συνάρτηση `createDatabase()` την εκτελούμε μόνο μια φορά γιατί θέλουμε να δημιουργήσουμε μια βάση δεδομένων. Δεν θέλουμε να εκτελείται συνέχεια και να μας βγάζει πολλές βάσεις δεδομένων. Για να μην γίνει αυτό και έχουμε πολλές βάσεις δεδομένων τότε βάζουμε την δίσηση μπροστά από την συνάρτηση `createDatabase()` και την απενεργοποιούμε. Πριν απενεργοποιήσουμε την βάση πρέπει να τονίσουμε ότι με την `print_hi("Created")` θα μας βγάλει στο κέλυφος της `python` ότι δημιουργήθηκε η βάση μας. Απενεργοποιούμε την συνάρτηση ως εξής:

```
#createDatabase()
```

Έτσι με αυτόν τον τρόπο δημιουργήσαμε την βάση μας. Το ίδιο θα γίνει και στις άλλες συναρτήσεις που θα τις εκτελούμε μόνο μια φορά και μετά θα τις απενεργοποιούμε με την δίσηση μπροστά από την συνάρτηση. Συνεχίζουμε με την βιβλιοθήκη `Sense Hat`, η οποία είναι:

```
from sense_hat import SenseHat
```

Με την επόμενη εντολή θα μπορέσουμε να δώσουμε παραδείγματος χάρη το όνομα `s` στη συσκευή μας:

```
s=SenseHat()
```

Οι δυο ακριβώς από πάνω εντολές είναι αναγκαίο να περιέχονται στα προγράμματά μας αν κάνουμε οποιαδήποτε χρήση του `Sense Hat`.

Με την εντολή `s.clear()` θα μπορούμε να διαγράψουμε τυχόν παλιές τιμές που έχει δεχθεί η συστοιχία `LED`.

Επίσης, δημιουργούμε τη συνάρτηση `createTable()` όπου αυτή η συνάρτηση δημιουργεί έναν πίνακα με τις μεταβλητές `θερμοκρασία`, `υγρασία`, `βαρομετρική πίεση` και `χρόνου`. Η κάθε μια από τις μεταβλητές αυτές είναι τύπου `double(size, d)` όπου το `size` είναι ο συνολικός αριθμός των ψηφίων και το `d` είναι το μέρος των αριθμών μετά την υποδιαστολή όπου είναι ο αριθμός των δεκαδικών ψηφίων που θα έχει η μεταβλητή. Δηλαδή πόσα δεκαδικά ψηφία θα υπάρχουν μετά την υποδιαστολή. Το κομμάτι κώδικα που το περιγράφει είναι:

```

27 def createTable():
28     mycursor = mydb.cursor()
29     mycursor.execute("CREATE TABLE values3 (temperature double(40,2), humidity double(40,2), pressure double(40, 2), date double(40, 7))")
30     print_hi('Table')
31

```

**def createTable():**

**mycursor = mydb.cursor()**

**mycursor.execute("CREATE TABLE values3(temperature double (40,2), humidity double(40,2), pressure double(40,2), date double(40,7))")**

**print\_hi('Table')**

Για να δημιουργηθεί ο πίνακας πρέπει να τον καλέσουμε με το όνομα που έχει οριστεί από εμάς και η συνάρτηση αυτή δεν έχει ορίσματα. Την καλούμε με το όνομα createTable() και δημιουργούμε τον πίνακα της βάσης μας. Αξίζει να σημειωθεί ότι την καλούμε μια φορά. Με την εντολή print\_hi('Table') μας βγάζει μήνυμα στο κέλυφος της rython ότι ο πίνακας έχει δημιουργηθεί. Για να την απενεργοποιήσουμε, ώστε να μην εκτελείται συνέχεια βάζουμε δίεση μπροστά από την createTable() δηλαδή #createTable().

Με την συναρτήση insertValueToDatabase() εισάγουμε τις τιμές στις μεταβλητές θερμοκρασία, υγρασία, βαρομετρική πίεση και χρόνο. Η συνάρτηση φαίνεται στην Εικόνα 17.

```

32
33
34 def insertValueToDatabase(temperature, humidity, pressure, date):
35     mycursor = mydb.cursor()
36     sql = "INSERT INTO values3 (temperature, humidity, pressure, date) VALUES (% s, % s, % s, % s)"
37     val = (temperature, humidity, pressure, date)
38     mycursor.execute(sql, val)
39     mydb.commit()
40     print("1 record inserted, ID:", mycursor.lastrowid)

```

Εικόνα 17: Συνάρτηση insertValueToDatabase

Για να εκτελέσουμε την συνάρτηση που είναι στην Εικόνα 17, θα πρέπει να την καλέσουμε με το ονομά της που έχει οριστεί από εμάς και στα ορίσματα που έχει η συνάρτηση πρέπει να βάλουμε τιμές. Όπως θα παρατηρήσουμε στην Εικόνα 17 οι τιμές που θα εισάγουμε στη συνάρτηση είναι τέσσερις, μια τιμή για κάθε μεταβλητή που έχουμε ορίσει. Δηλαδή δίνουμε τιμές για τις μεταβλητές θερμοκρασία, υγρασία, βαρομετρική πίεση και χρόνου. Αφού βάλουμε τις τιμές στις μεταβλητές που ορίσαμε τότε καλούμε την συναρτηση insertValueToDatabase η οποία προσθέτει τις τιμές στον πίνακα values3. Για να απενεργοποιήσουμε την συνάρτηση αυτή γίνεται με την εντολή:

**# insertValueToDatabase**

Η συνάρτηση getLastData() η οποία φαίνεται στην Εικόνα 18, διαβάζει την τελευταία εγγραφή του πίνακα values3 και επιστρέφει έναν πίνακα ο οποίος περιέχει τις τιμές των αισθητήρων μας οι οποίες είναι η θερμοκρασία, η υγρασία και η βαρομετρική

πίεση. Για να καλέσουμε την συνάρτηση `getLastData()`, τότε την καλούμε με το ονομά της με και χωρίς ορίσματα. Την εκτελούμε μόνο μια φορά. Για να απενεργοποιήσουμε την συνάρτηση αυτή γίνεται με την εντολή `#getLastData()`.

```
42 def getLastData():
43     mycursor = mydb.cursor()
44     mycursor.execute("SELECT * FROM values3")
45     myresult = mycursor.fetchone()
46     print(myresult)
47     print(myresult[0])
48     print(myresult[1])
49     print(myresult[2])
50     print(myresult[3])
51
```

Εικόνα 18: Συνάρτηση `getLastData()`

Ωστόσο, έχουμε την συνάρτηση `writeValuesLoop()` η οποία ελέγχει εάν πέρασε ο χρόνος που έχουμε ορίσει ως διάστημα μεταξύ δυο συνεχόμενων εγγραφών στη βάση και αν ναι καλεί την συνάρτηση `insertValueDatabase()` για να προστεθούν οι νέες τιμές που λάβαμε από τους αισθητήρες στη βάση.

Η συνάρτηση `writeValuesLoop()` στην αρχή αποθηκεύει τον χρόνο του συστήματος στη μεταβλητή `LastUpdateTime` και στη συνέχεια δημιουργεί ένα Loop το οποίο δεν σταματάει ποτέ μέσα στο οποίο ελέγχει συνεχώς αν η ώρα του συστήματος είναι μεγαλύτερη από τη `LastUpdateTime` κατά 300 δευτερόλεπτα και αν ναι προσθέτει τις τιμές των αισθητήρων στη βάση και κάνει ίση τη `LastUpdateTime` με την τρέχουσα ούτως, ώστε η επόμενη εγγραφή να γίνει 300 δευτερόλεπτα μετά. Ο αριθμός για τα 300 δευτερόλεπτα προκύπτει ότι θέλαμε η βάση μας να καταγράφει τις τιμές της βάσης μας ανά 5 λεπτά. Οι τιμές τις οποίες έχουμε στη βάση μας είναι η θερμοκρασία, η υγρασία και η βαρομετρική πίεση. Πιο αναλυτικά τα 5 λεπτά προκύπτουν ως εξής:  $5 \text{ min} = 5 * 60 \text{ sec} = 300 \text{ sec}$ .

Για να την καλέσουμε τη συγκεκριμένη συνάρτηση την καλούμε με το όνομα `writeValuesLoop()` και την αφήνουμε να εκτελείται συνέχεια ώστε να γράφονται οι τιμές των αισθητήρων καθώς και ο χρόνος που γίνεται η κάθε εγγραφή στη βάση δεδομένων μας. Η συνάρτηση `writeValuesLoop()` φαίνεται στην Εικόνα 19.

```
53 def writeValuesLoop():
54     lastUpdateTime = time.time()
55
56     while True:
57         print(lastUpdateTime)
58         if (time.time() > lastUpdateTime + 300):
59             print(time.time() - lastUpdateTime)
60             lastUpdateTime = time.time()
61             temperature = s.temperature
62             humidity = s.humidity
63             pressure = s.pressure
64             print(lastUpdateTime)
65             insertValueToDatabase(temperature, humidity, pressure, lastUpdateTime)
```

Εικόνα 19: Συνάρτηση `writeValuesLoop()`

Επίσης, πρέπει να προσθέσουμε στο δεύτερο τμήμα του κώδικα της βάσης μας τις παρακάτω δύο γραμμές για να ολοκληρωθεί ο κώδικας της βάσης μας:

```
if __name__ == '__main__':
```

```
print_hi('PyCharm')
```

Ύστερα αφού εξηγήσαμε τα δυο τμήματα του κώδικα, όπου το κάθενα από τα τμήματα αυτά υλοποιείται με κώδικα που έχουμε αναπτύξει σε γλώσσα Python, τώρα είμαστε σε θέση να εξηγήσουμε την λειτουργία και την εκτέλεση της εφαρμογής μας. Ανοίγουμε το terminal για να πληκτρολογήσουμε και να εκτελέσουμε τις παρακάτω εντολές:

```
cd ~/Documents/Telikh_Ergasia
```

```
python3 myapp.py
```

Η εντολή `cd ~/Documents/Telikh_Ergasia` έχει την κάθετο / μετά το σύμβολο (~) δηλώνει ότι προσπαθούμε να πλοηγηθούμε σε υποφάκελο ενώ όλο το μονοπάτι που ακολουθεί είναι ότι θέλουμε να μπούμε στον υποφάκελο Documents για να δούμε τι περιέχει ο φάκελος Telikh\_Ergasia. Η επόμενη εντολή `python3 myapp.py` όταν την εκτελέσουμε μας βγάζει τα παρακάτω αποτελέσματα που φαίνονται στην Εικόνα 20.

```
* Serving Flask app "myapp" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 155-181-660
```

Εικόνα 20: Ενημερωτικά μηνύματα

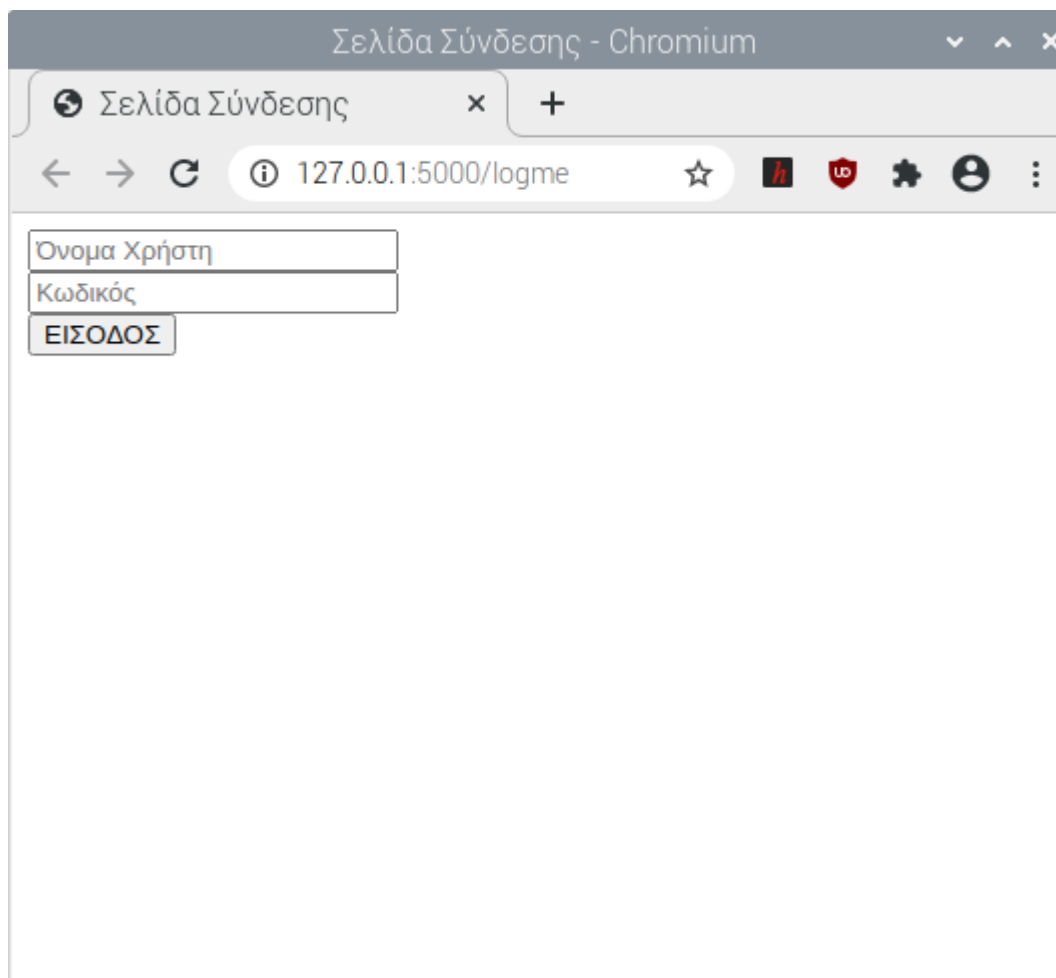
Η Εικόνα 20 με την εκτέλεση της εντολής `python3 myapp.py` μας βγάζει τα σχετικά ενημερωτικά μηνύματα. Πρωτίστως μας δηλώνει ότι εκτελεί την εφαρμογή μας(myapp) σε ένα περιβάλλον παραγωγής, μαζί με ένα μήνυμα το οποίο μας προτείνει να μην εφαρμόσουμε το development server που είναι ενσωματωμένο στο Flask, σε περιβάλλον παραγωγής. Επίσης, μας ενημερώνει για τον εντοπισμό των σφαλμάτων ο οποίος είναι ενεργός(Debug mode:on) και συντρέχει το Flask στη διεύθυνση :

<http://127.0.0.1:5000/>

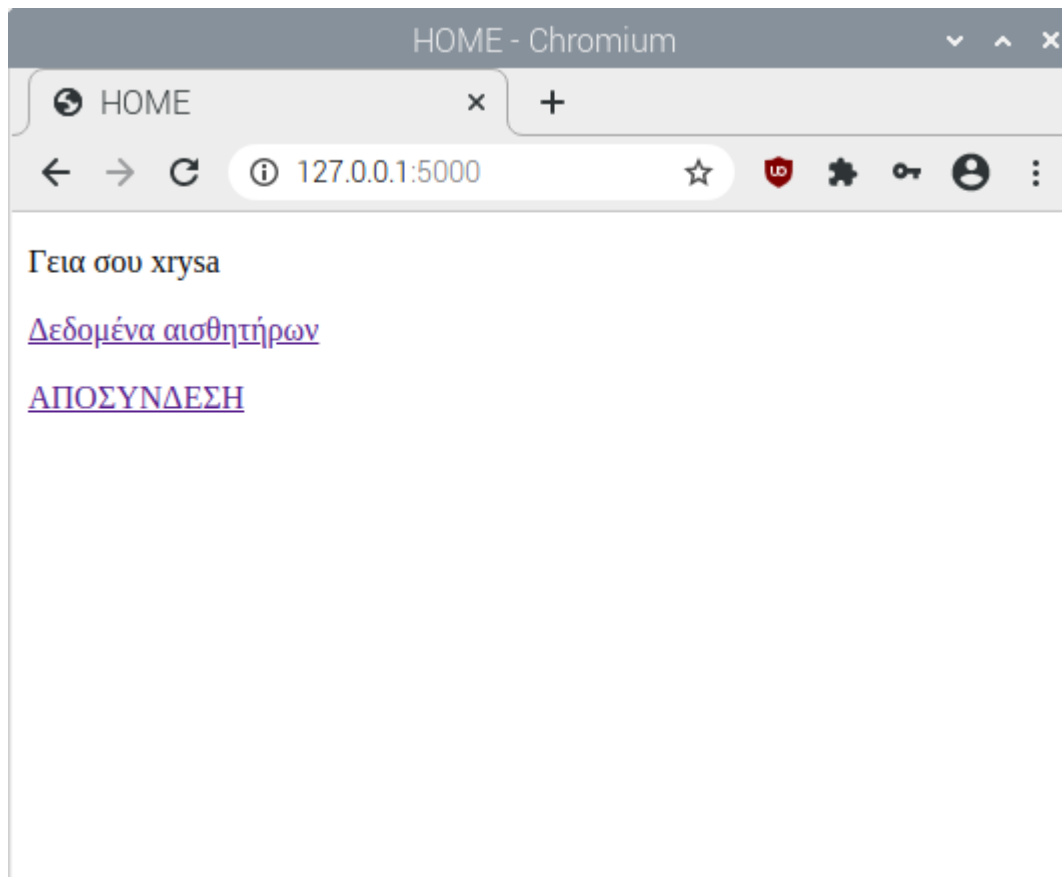
Αμέσως μετά επισκεπτόμαστε την παραπάνω διεύθυνση μέσα από τον περιηγητή του Raspberry Pi και έχουμε το εξής αποτέλεσμα:



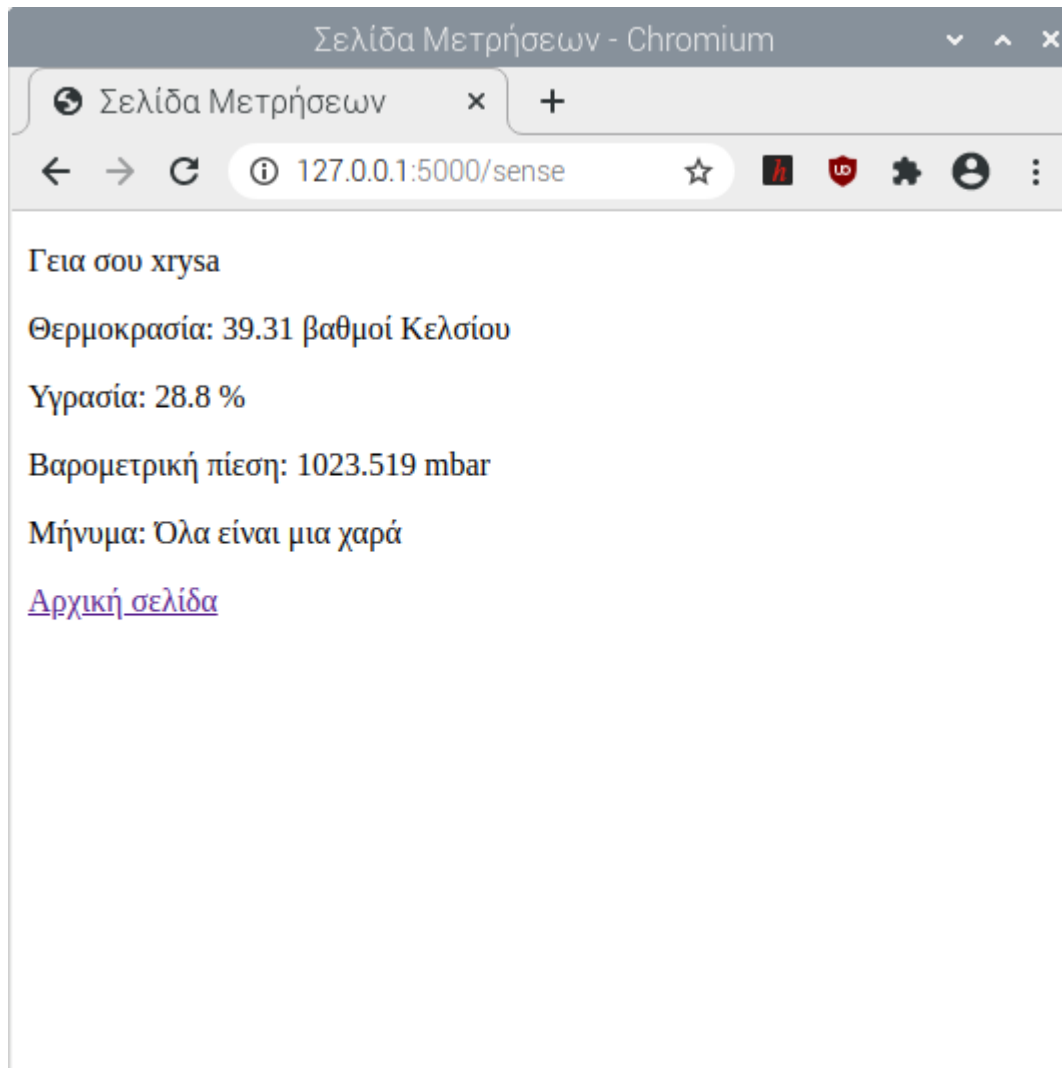
```
pi@raspberrypi4: ~/Documents/Telikh_Ergasia
Αρχείο Επεξεργασία Καρτέλες Βοήθεια
pi@raspberrypi4:~ $ cd ~/Documents/Telikh_Ergasia
pi@raspberrypi4:~/Documents/Telikh_Ergasia $ python3 myapp.py
* Serving Flask app "myapp" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 155-181-660
127.0.0.1 - - [21/May/2021 15:06:28] "POST /logme HTTP/1.1" 302 -
127.0.0.1 - - [21/May/2021 15:06:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:07:36] "GET /sense HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:07:38] "GET /sense HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:07:39] "GET /sense HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:07:40] "GET /sense HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:07:40] "GET /sense HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:07:42] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/May/2021 15:07:45] "GET /logme HTTP/1.1" 200 -
^Cpi@raspberrypi4:~/Documents/Telikh_Ergasia $ scrot -d 10
```



Το παραπάνω αποτέλεσμα μας οδηγεί στην σελίδα σύνδεσης και εκεί ο χρήστης συμπληρώνει τα στοιχεία του και πατάει το κουμπί είσοδος για να μπει στη σελίδα. Αφού μπούμε στη σελίδα σύνδεσης εκεί θα παρατηρήσουμε ότι το `{{g.user[1]}}` που υπάρχει στο μήνυμα Γεια σου `{{g.user[1]}}` και βρίσκεται στο αρχείο `index.html`, τότε προβάλλουμε στη θέση αυτή το όνομα του χρήστη που έχουμε αποθηκεύσει στο αντικείμενο `g.user` καθώς επίσης και τους συνδέσμους Δεδομένα αισθητήρων και ΑΠΟΣΥΝΔΕΣΗ.



Αν επιλέξουμε τον σύνδεσμο Δεδομένα Αισθητήρων, τότε θα παρατηρήσουμε ότι θα έχει τις τιμές της θερμοκρασίας, της βαρομετρικής πίεσης και της υγρασίας. Περιλαμβάνει, επίσης και το μήνυμα αν όλα πάνε καλά ή όχι και αυτό ορίζεται αν η τιμή που θέλουμε να βάλουμε είναι αρκετά καλή στις επιτρεπόμενες τιμές που έχουμε βάλει στον κωδικά μας. Παρατηρούμε και σε αυτό ότι με το `{{g.user[1]}}` στο μήνυμα «Γεια σου `{{g.user[1]}}`» όπου υπάρχει στο `index.html`, τότε προβάλλουμε στη θέση αυτή το όνομα του χρήστη που έχουμε αποθηκεύσει στο αντικείμενο `g.user`.



Για να σταματήσουμε το Flask εφαρμόζουμε το συνδυασμό των πλήκτρων **Ctrl-C**. Αν δεν πατήσουμε ταυτόχρονα τα πλήκτρα **Ctrl-C** τότε είναι πιθανό να αντιμετωπίσουμε λάθος κατά την επανεκτέλεση της εφαρμογής μας, όπως αυτό απεικονίζεται παρακάτω.

**Σε περίπτωση που μας εμφανιστεί το λάθος:**

**OSError:[Errno 98] Address already in use**

Τότε σε αυτή την περίπτωση, αν κατά την εκτέλεση της εφαρμογής εμφανιστεί λάθος με περιγραφή "Address already in use" αυτό δηλώνει ότι εκτελείται ακόμη το flask. Συνεπώς θα πρέπει να περιορίσουμε την εκτέλεση της εφαρμογής αυτής και να την τερματίσουμε.

Άλλη μια εντολή του bash στο Linux που μπορούμε να γνωρίζουμε είναι η ps, η οποία μας αφήνει να δούμε τις διεργασίες που εκτελούνται στο λειτουργικό σύστημα.

Για να εξετάσουμε αν εκτελείται κάποια διεργασία θα πρέπει στο terminal να πληκτρολογήσουμε την εντολή:

### **ps -fA | grep python3**

Η παραπάνω εντολή απαρτίζεται από δύο εντολές χωρισμένες με το σύμβολο της κατακόρυφης μπάρας (|), ή αλλιώς pipe. Αυτό μας δείχνει πως το αποτέλεσμα της πρώτης εντολής περνάει στη δεύτερη εντολή, δηλαδή στην εντολή grep στο παράδειγμά μας.

Στην εντολή ps η παράμετρος -f σημαίνει ότι η ps θα μας εμφανίσει περισσότερες πληροφορίες, ενώ η παράμετρος -A σημαίνει ότι θα εμφανιστούν όλες οι διεργασίες. Η εντολή grep αναλαμβάνει να αναζητήσει στο αποτέλεσμα της ps τις γραμμές που περιέχουν τη λέξη python3.

Έτσι η παραπάνω εντολή θα εμφανίσει κάτι που να είναι ανάλογο με το παρακάτω:

```
pi@raspberrypi4:~/Documents/Telikh_Ergasia $ ps -fA | grep python3
root      447      1   0 17:29 ?        00:00:00 /usr/bin/python3 -m sensehat.sta
rtservice
cayenne   796      1   0 17:29 ?        00:00:03 /usr/bin/python3 -m myDevices -P
/var/run/myDevices/cayenne.pid
pi        1057     1   0 17:29 ?        00:00:00 /usr/bin/python3 /usr/share/syst
em-config-printer/applet.py
pi        2062    1036   5 17:31 ?        00:00:17 /usr/bin/python3 /usr/bin/thonny
/home/pi/Documents/Telikh_Ergasia/myapp.py
pi        2070    2062   1 17:31 ?        00:00:04 /usr/bin/python3 -u -B -m thonny
.plugins.cpython /home/pi/Documents/Telikh_Ergasia
pi        2249    2194   0 17:32 pts/0    00:00:00 python3 myapp.py
pi        3261    2194   0 17:37 pts/0    00:00:00 grep --color=auto python3
pi@raspberrypi4:~/Documents/Telikh_Ergasia $ kill 2249
pi@raspberrypi4:~/Documents/Telikh_Ergasia $
```

Μας ενδιαφέρει πρωτίστως η τελευταία στήλη, η οποία μας εμφανίζει (όπως φαίνεται στην προτελευταία γραμμή της παραπάνω εικόνας), ότι εκτελείται ακόμη το python3 myapp.py. Σημειώνουμε τον αριθμό που θα βρούμε στη δεύτερη στήλη ο οποίος είναι το 2249 και διακόπτουμε την διεργασία με την εντολή:

### **kill 2249**

Θα παρατηρήσουμε ότι το 2249 είναι ο αριθμός της διεργασίας που έχουμε εντοπίσει στο δικό μας σύστημα και φαίνεται στην παραπάνω εικόνα. Τώρα είμαστε έτοιμοι να εκτελέσουμε το πρόγραμμά μας κανονικά.

Τέλος για να γίνει η εκτέλεση της εφαρμογής μας που περιγράψαμε ήδη θα πρέπει στον κώδικα της ιστοσελίδας μας που είναι το αρχείο myapp.py θα πρέπει να περιλαμβάνονται στην main οι παρακάτω δύο γραμμές:

```
88 ▶ if __name__ == '__main__':
89     app.run(debug=True)
90
```

## 5.2 Αποτελέσματα

Μέσω της διαδικτυακής εφαρμογής <https://timestamp.online/> όπου βάζουμε στο Convert date to timestamp την ημερομηνία που θέλουμε όπως επίσης και την ώρα για να πάρουμε τα δεδομένα που θέλουμε και να πατήσουμε το κουμπί Convert και μας δείχνει τον αριθμό στο χρόνο που έγινε η καταγραφή για την ημερομηνία που έχουμε ορίσει. Αυτό το κάνουμε και για όλες τις ημερομηνίες που θέλουμε. Επίσης η καταγραφή έγινε για 4 συνεχόμενες ημέρες, όπου είναι για τις 14 Ιουλίου, τις 15 Ιουλίου, τις 16 Ιουλίου και 17 Ιουλίου. Στην παρακάτω εικόνα θα παρατηρήσουμε την ημερομηνία μέσω της διαδικτυακής εφαρμογής για την ημερομηνία στις 14 Ιουλίου και ώρα 00:00.

### Convert date to timestamp

The screenshot shows a web application interface with the following elements:

- An input field containing the text "2021-07-14 00:00".
- A button labeled "Convert" below the input field.
- An output field containing the text "1626210000".
- Below the output field, the text "1626210000" is displayed in blue, followed by "Countdown to: 1627200000" in black.

Το ίδιο κάνουμε και για τις υπόλοιπες ημερομηνίες. Ακολουθούν οι εικόνες αντίστοιχα για τις ημερομηνίες 15 Ιουλίου, 16 Ιουλίου και 17 Ιουλίου όπου γι'αυτές τις ημερομηνίες έχουμε ορίσει την ώρα 00:00.

### Convert date to timestamp

The screenshot shows a web application interface with the following elements:

- An input field containing the text "2021-07-15 00:00".
- A button labeled "Convert" below the input field.
- An output field containing the text "1626296400".
- Below the output field, the text "1626296400" is displayed in blue, followed by "Countdown to: 1627200000" in black.

## Convert date to timestamp

  
  
  
[1626382800](#)  
Countdown to: [1627200000](#)

## Convert date to timestamp

  
  
  
[1626469200](#)  
Countdown to: [1627200000](#)

Με τη μορφή ερωτημάτων που γίνεται στη βάση μας, μας βγάζει τα αποτελέσματα της καταγραφής από τις 14 Ιουλίου την ώρα 00:00 μέχρι τις 15 Ιουλίου την ώρα 00:00. Το ερώτημα είναι:

```
SELECT * FROM 'values3' WHERE date>=1626210000 and date<1626296400
```

το οποίο φαίνεται στην επόμενη εικόνα:

192.168.1.12 / localhost / ergasia1Database / values3 | phpMyAdmin 4.6.6deb5 - Chromium

192.168.1.12 / localhost x +

Μη ασφαλής | 192.168.1.12/phpmyadmin/tbl\_sql.php?db=ergasia1Database&table=value...

phpMyAdmin

Πρόσφατα | Αγαπημένοι

Νέα  
 ergasia1Database  
 Νέο  
 values3  
 information\_schema  
 mysql  
 performance\_schema  
 phpmyadmin

Περιήγηση | Δομή | Κώδικας SQL | Αναζήτηση | Προσθήκη | Περισσότερα

Προβολή παραθύρου ερωτήματος

⚠ Η τρέχουσα επιλογή δεν περιέχει μια μοναδική στήλη. Τα χαρακτηριστικά επεξεργασίας πλέγματος, πλαίσια ελέγχου, Επεξεργασίας, Αντιγραφής και Διαγραφής δεν είναι διαθέσιμα.

✔ Εμφάνιση εγγραφών 0 - 6 (7 συνολικά. Το ερώτημα χρειάστηκε 0.0002 δευτερόλεπτα.)

```
SELECT * FROM `values3` WHERE date>=1626210000 and date<1626296400
```

Δημιουργία προφίλ [ Επεξεργασία εσωτερικά ] [ Επεξεργασία ] [ Ανάλυση SQL ] [ Δημιουργία κώδικα PHP ] [ Ανανέωση ]

Εμφάνιση όλων | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτηση σε αυτόν τον πίνα

+ Επιλογές

temperature	humidity	pressure	date
36.58	48.50	1015.86	1626294321.5418358
37.81	45.76	1015.85	1626294621.6924780
38.28	44.25	1015.88	1626294921.8057630
42.96	37.20	1015.88	1626295221.9899924
45.82	33.21	1015.96	1626295522.2334878
47.07	31.14	1016.05	1626295822.2519042
42.38	35.50	1016.05	1626296122.2702003

Εμφάνιση όλων | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτηση σε αυτόν τον πίνα

Λειτουργίες αποτελεσμάτων ερωτήματος

Εκτύπωση | Αντιγραφή στο πρόχειρο | Εξαγωγή | Εμφάνιση διαγράμματος  
 Δημιουργία προβολής

Αποθήκευση αυτού του ερωτήματος SQL

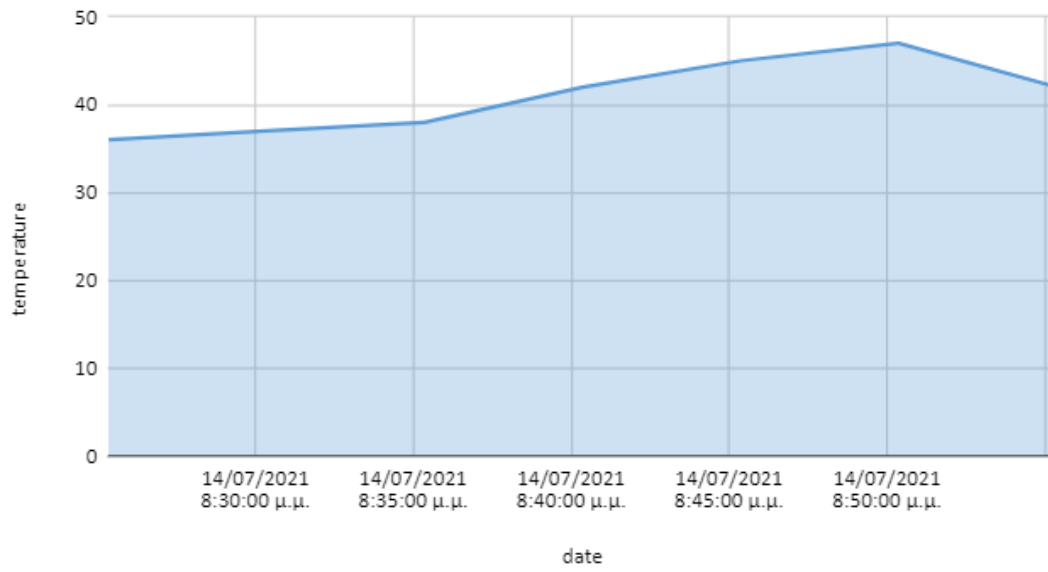
Ετικέτα:   Δικαίωμα πρόσβασης στο σελιδοδείκτη σε κάθε χρήστη

Αποθήκευση αυτού του ερωτήματος SQL

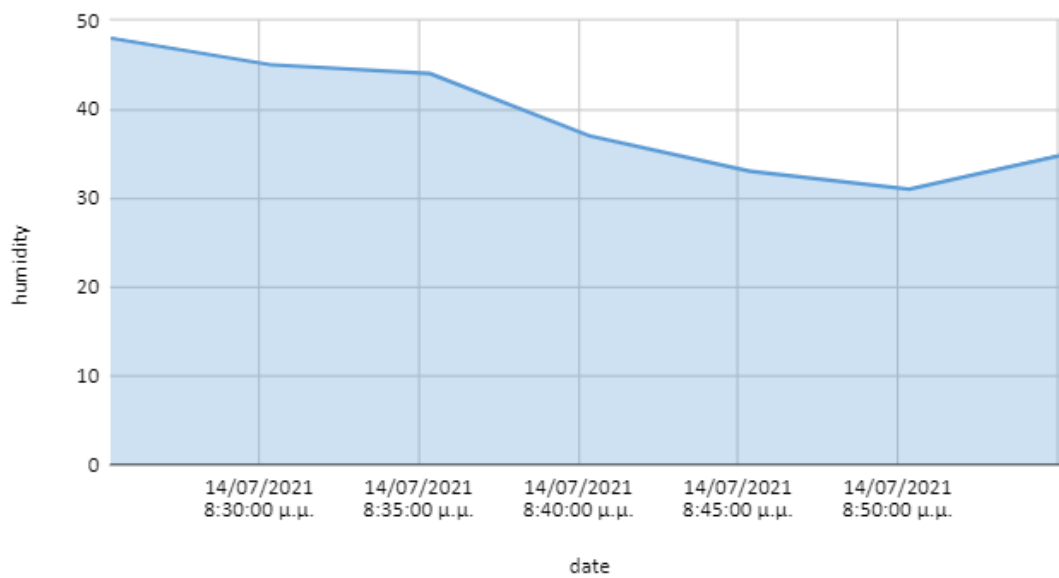
Παράθυρο

Τα αντίστοιχα διαγράμματα για την παραπάνω καταγραφή είναι:

Διάγραμμα θερμοκρασίας-χρόνου

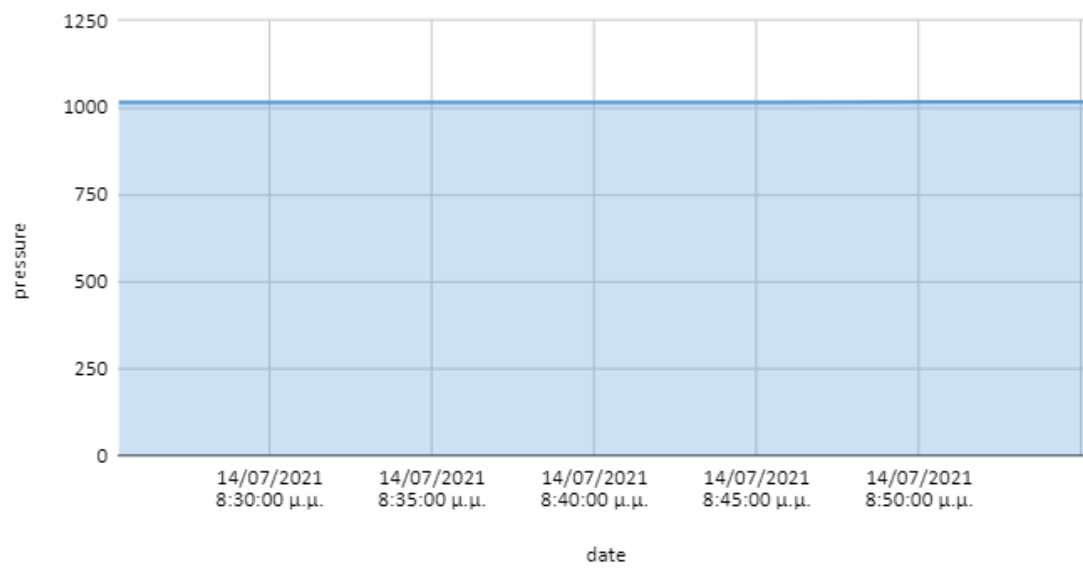


Διάγραμμα υγρασίας-χρόνου





## Διάγραμμα πίεσης-χρόνου



Επίσης, έχουμε την καταγραφή για τις ημερομηνίες 15 Ιουλίου μέχρι 16 Ιουλίου και ώρα 00:00. Όπου και αυτή για να πάρουμε τα αποτελέσματα αυτής της καταγραφής γίνεται με τη μορφή ερώτηματος το οποίο είναι:

```
SELECT * FROM 'values3' WHERE date>=1626296400 and date<1626382800 και φαίνονται στην επόμενη εικόνα:
```

192.168.1.12 / localhost / ergasia1Database / values3 | phpMyAdmin 4.6.6deb5 - Chromium

192.168.1.12 / localhost x +

Μη ασφαλής | 192.168.1.12/phpmyadmin/tbl\_sql.php?db=ergasia1Database&table=value...

phpMyAdmin

Πρόσφατα / Αγαπημένοι

- Nέα
- ergasia1Database
  - Nέο
  - values3
- information\_schema
- mysql
- performance\_schema
- phpmyadmin

Περιήγηση | Δομή | Κώδικας SQL | Αναζήτηση | Προσθήκη | Περισσότερα

Προβολή παραθύρου ερωτήματος

⚠ Η τρέχουσα επιλογή δεν περιέχει μια μοναδική στήλη. Τα χαρακτηριστικά επεξεργασίας πλέγματος, πλαίσια ελέγχου, Επεξεργασίας, Αντιγραφής και Διαγραφής δεν είναι διαθέσιμα.

✔ Εμφάνιση εγγραφών 0 - 16 (17 συνολικά, Το ερώτημα χρειάστηκε 0.0016 δευτερόλεπτα.)

```
SELECT * FROM `values3` WHERE date>=1626296400 and date<1626382800
```

Δημιουργία προφίλ [ Επεξεργασία εσωτερικά ] [ Επεξεργασία ] [ Ανάλυση SQL ] [ Δημιουργία κώδικα PHP ] [ Ανανέωση ]

Εμφάνιση όλων | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτηση σε αυτόν τον πίνακα

+ Επιλογές

temperature	humidity	pressure	date
39.87	39.16	1016.09	1626296422.4144707
39.83	39.05	1016.04	1626296722.5893292
40.03	38.39	1015.98	1626297022.6852267
40.23	38.40	1016.00	1626297323.0307900
40.23	38.31	1016.01	1626297623.1261215
39.16	39.71	1015.97	1626297923.4147115
38.95	40.75	1016.00	1626298223.7550642
39.16	40.35	1015.95	1626298523.9386368
39.31	40.30	1015.91	1626298824.0367267
39.34	40.42	1015.90	1626299124.4126450
39.45	40.16	1015.90	1626299424.8741424
39.76	39.68	1015.89	1626299725.2761898
39.56	39.71	1015.88	1626300025.4904394
39.87	39.80	1015.91	1626300325.6332817
39.47	39.98	1015.80	1626300625.9733727
39.33	40.04	1015.86	1626300925.9736116
39.62	39.98	1015.91	1626301226.3871565

Εμφάνιση όλων | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτηση σε αυτόν τον πίνακα

Λειτουργίες αποτελεσμάτων ερωτήματος

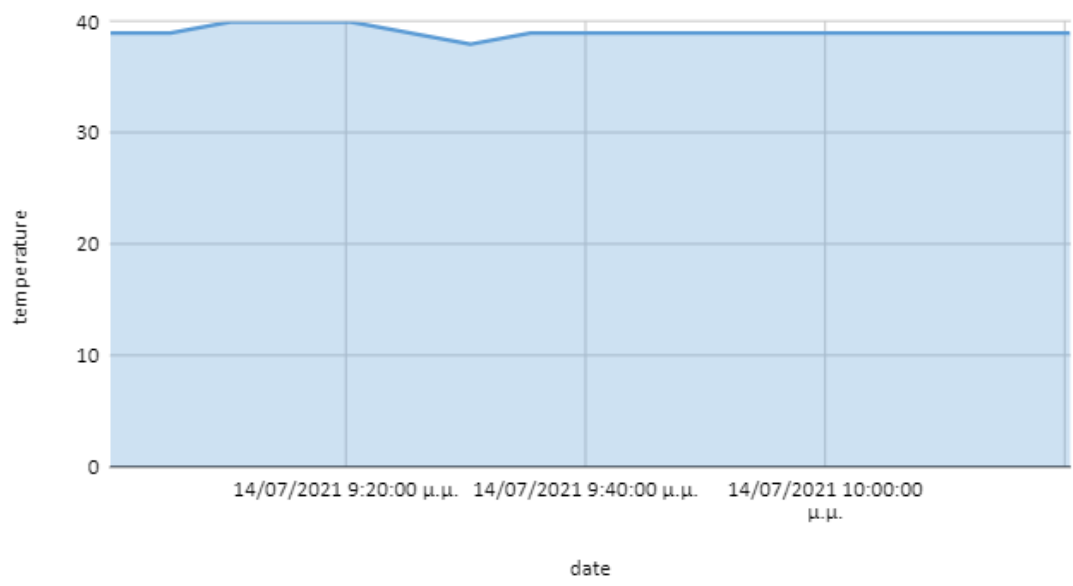
Εκτύπωση | Αντιγραφή στο πρόχειρο | Εξαγωγή | Εμφάνιση διαγράμματος

Δημιουργία προβολής

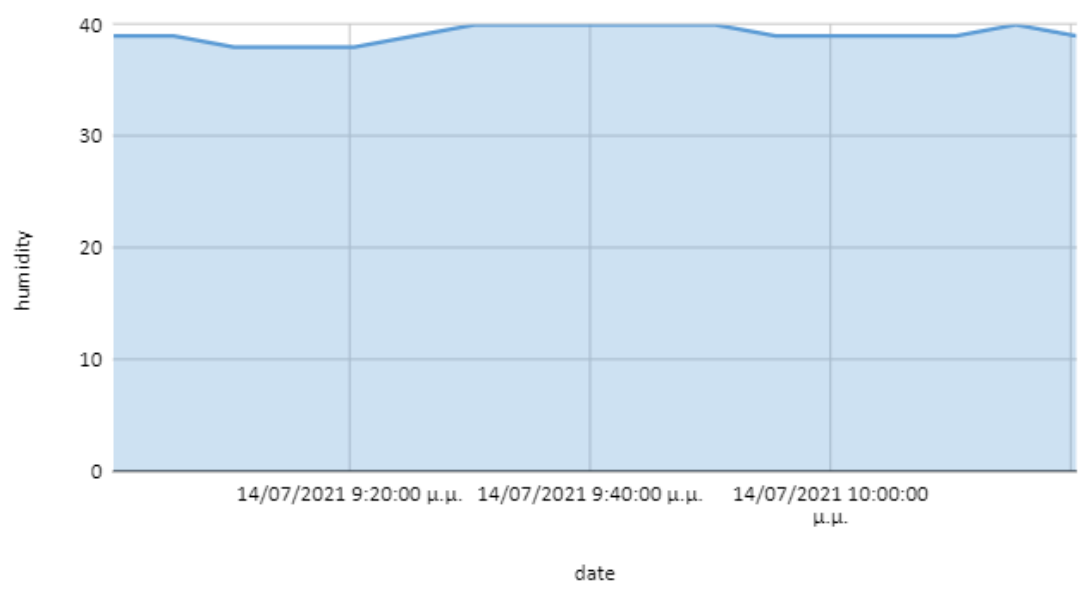
Παράθυρο

Εδώ σε αυτή την καταγραφή έχουμε τα αντίστοιχα διαγράμματα:

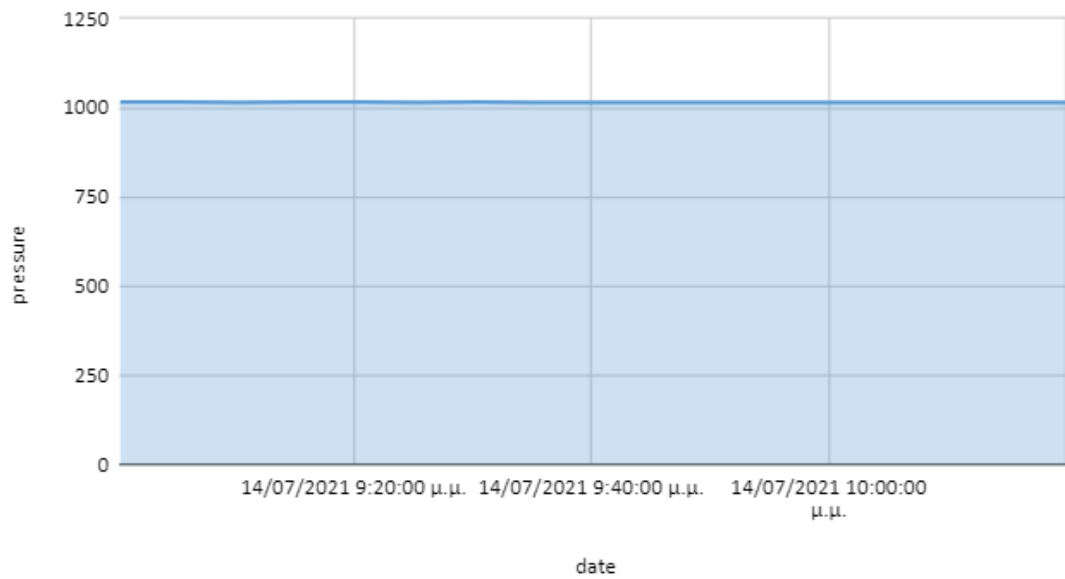
### Διάγραμμα θερμοκρασίας-χρόνου



### Διάγραμμα υγρασίας-χρόνου



### Διάγραμμα πίεσης-χρόνου



Τέλος, έχουμε την τελευταία καταγραφή η οποία είναι για τις ημερομηνίες 16 Ιουλίου μέχρι 17 Ιουλίου και ώρα 00:00, όπου τα αποτελέσματα της συγκεκριμένης καταγραφής γίνεται με το εξής ερώτημα:

```
SELECT * FROM 'values3' WHERE date>=1626382800 and date<1626469200
```

και τα αποτελέσματα αυτά φαίνονται στις παρακάτω εικόνες:

192.168.1.12 / localhost x +

Μη ασφαλής | 192.168.1.12/phpmyadmin/tbl\_sql.php?db=ergasia1Database&table=value...

phpMyAdmin

Πρόσφατα | Αγαπημένοι

Νέα  
ergasia1Database  
  Νέο  
  values3  
information\_schema  
mysql  
performance\_schema  
phpmyadmin

Περιήγηση | Δομή | Κώδικας SQL | Αναζήτηση | Προσθήκη | Περισσότερα

Εμφάνιση εγγραφών 0 - 24 (45 συνολικά, Το ερώτημα χρειάστηκε 0.0017 δευτερόλεπτα.)

```
SELECT * FROM `values3` WHERE `date`>=1626382800 and `date`<1626469200
```

Δημιουργία προφίλ [ Επεξεργασία εσωτερικά ] [ Επεξεργασία ] [ Ανάλυση SQL ] [ Δημιουργία κώδικα PHP ] [ Ανανέωση ]

1 > >>  Εμφάνιση όλων | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτ

+ Επιλογές

temperature	humidity	pressure	date
37.39	53.21	1014.16	1626386294.5675454
38.96	48.81	1014.22	1626386594.6237454
39.76	47.81	1014.31	1626386894.6543972
39.85	46.20	1014.30	1626387194.8115873
40.00	45.96	1014.28	1626387495.0381122
39.54	46.66	1014.30	1626387795.1598213
39.00	47.81	1014.22	1626388095.2854960
39.09	47.71	1014.25	1626388395.5038247
38.49	48.26	1014.18	1626388695.7003298
38.42	48.45	1014.18	1626388995.8919854
38.35	48.24	1014.18	1626389296.0152838
38.42	48.24	1014.14	1626389596.0712466
38.44	47.65	1014.16	1626389896.1379790
38.51	47.07	1014.12	1626390196.2333887
38.29	47.33	1014.05	1626390496.4476020
38.02	46.79	1014.03	1626390796.4789855
38.31	46.39	1014.02	1626391096.5239835
38.38	45.84	1014.00	1626391396.6000648
38.33	45.86	1013.93	1626391696.7951360
37.90	46.40	1013.87	1626391997.0306623
37.95	46.05	1013.82	1626392297.1466180
38.19	45.61	1013.88	1626392597.6095405
37.81	45.89	1013.84	1626392897.6619823
37.79	45.74	1013.87	1626393197.9073240
36.97	45.81	1011.91	1626463193.1532950

1 > >>  Εμφάνιση όλων | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτ

Παράθυρο

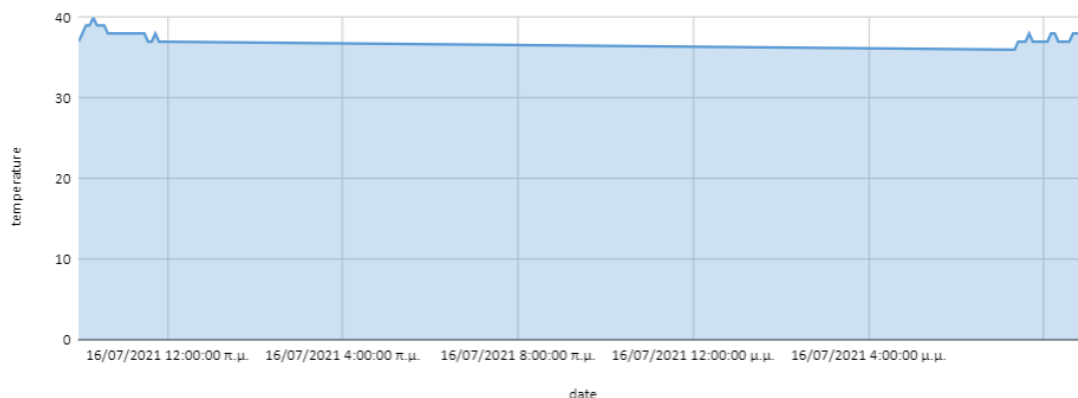
The screenshot shows the phpMyAdmin interface for a database named 'ergasia1Database'. The table 'values3' is selected, displaying a list of weather records. The columns are 'temperature', 'humidity', 'pressure', and 'date'. The data shows temperature values ranging from 36.97 to 38.71, humidity from 42.38 to 45.81, and pressure from 1011.91 to 1012.45. The dates are in the format YYYYMMDD.

temperature	humidity	pressure	date
36.97	45.81	1011.91	1626463193.1532950
37.50	44.79	1012.02	1626463493.2117527
37.46	45.34	1012.03	1626463793.2129114
37.73	44.35	1012.10	1626464093.3433046
38.24	43.73	1012.10	1626464393.3874261
37.95	43.91	1012.12	1626464693.4117840
37.59	44.60	1012.16	1626464993.4394596
37.86	44.04	1012.23	1626465293.6190060
37.82	44.35	1012.19	1626465593.7238567
37.82	44.30	1012.21	1626465893.8090281
38.42	43.23	1012.20	1626466193.9456756
38.29	43.22	1012.21	1626466493.9740236
37.90	43.88	1012.15	1626466794.1103177
37.37	44.98	1012.21	1626467094.3566513
37.50	44.42	1012.31	1626467394.5362320
37.82	43.86	1012.38	1626467694.5659764
38.04	43.57	1012.46	1626467994.6305542
38.44	42.31	1012.45	1626468294.7168480
38.64	42.75	1012.42	1626468594.8106058
38.71	42.38	1012.45	1626468894.8480017
38.71	42.51	1012.43	1626469195.2380414

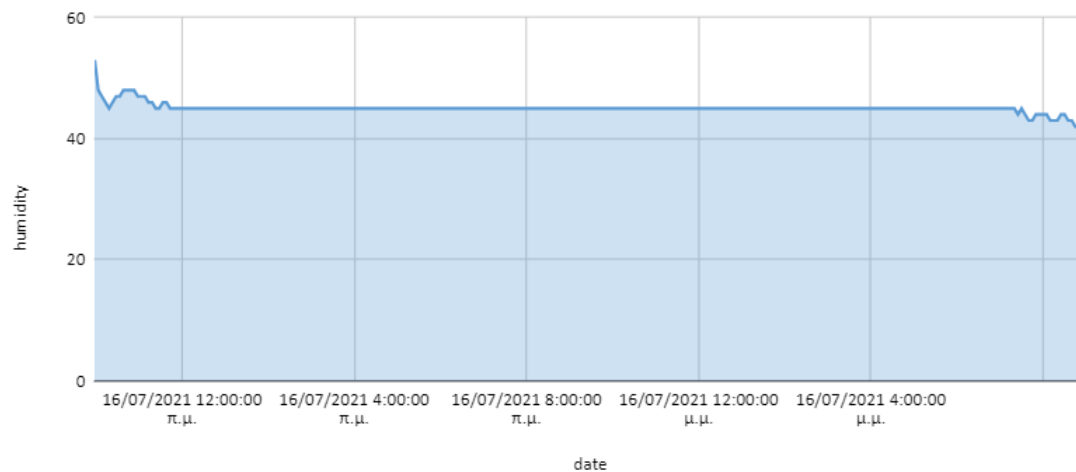
Below the table, there are options for 'Εμφάνιση όλων' (Show all), 'Αριθμός εγγραφών' (Number of records) set to 50, and 'Φιλτράρισμα εγγραφών' (Filter records) set to 'Αναζήτηση σε αυτόν τον πίνακα' (Search in this table). There are also buttons for 'Λειτουργίες αποτελεσμάτων ερωτήματος' (Query result operations) and 'Αποθήκευση αυτού του ερωτήματος SQL' (Save this SQL query).

Για την τελευταία καταγραφή έχουμε τα τελευταία διαγράμματα:

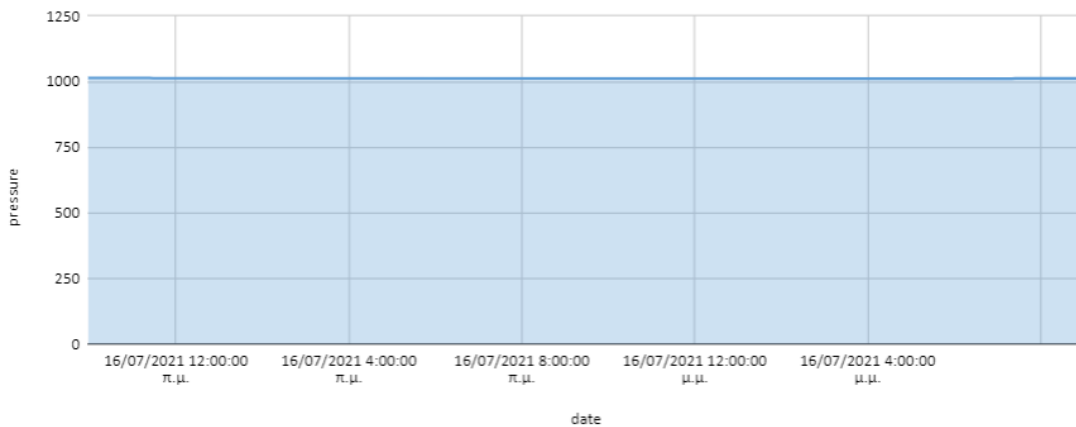
Διάγραμμα θερμοκρασίας-χρόνου



Διάγραμμα υγρασίας-χρόνου



Διάγραμμα πίεσης-χρόνου



Για τα αποτελέσματα των καταγραφών που είδαμε παραπάνω, δηλαδή για τις ημερομηνίες από 14-17 Ιουλίου, μπορούμε να παρατηρήσουμε ότι η πρώτη στήλη αντιστοιχεί για την θερμοκρασία, η δεύτερη στήλη είναι για την υγρασία, η τρίτη στήλη για την βαρομετρική πίεση και η τελευταία στήλη αντιστοιχεί στον χρόνο στον οποίο γίνεται η κάθε εγγραφή ανά 5 λεπτά.

### 5.3 Συζήτηση

Σε αυτή την ενότητα, θα εξηγήσουμε ποιες θα πρέπει να είναι οι συνθήκες του περιβάλλοντος όπου είναι η θερμοκρασία, η υγρασία και η βαρομετρική πίεση ώστε να μπορέσουν να συντηρηθούν τα τρόφιμα και αναλόγως με τις συνθήκες του περιβάλλοντος θα μπορέσουμε να ελέγξουμε για πόσο χρονικό διάστημα θα μπορέσουν να διατηρηθούν τα τρόφιμα σε έναν χώρο αποθήκευσης με βάση τα διαγράμματα που έχουμε εξάγει.

Για να μπορέσουν να συντηρηθούν σωστά τα τρόφιμα όπως είναι οι καρποί θα πρέπει να εξετάσουμε ποιες είναι οι επιτρεπτές συνθήκες για τους καρπούς αυτούς, ώστε να μην αλλοιωθεί η ποιότητά τους λόγω της γήρανσης και της ανάπτυξης παθογόνων μικροοργανισμών που πιθανόν να βρίσκονται σε έναν χώρο αποθήκευσης [16]. Για να υπάρξει επιτυχής συντήρηση των καρπών θα πρέπει οι συνθήκες του περιβάλλοντος όπως η θερμοκρασία, η υγρασία και η βαρομετρική πίεση να είναι κατάλληλες για τον χώρο αποθήκευσης όπου αποθηκεύονται οι καρποί αυτοί. Επίσης οι συνθήκες αυτές παίζουν σημαντικό ρόλο για την διάρκεια ζωής των καρπών και για την ποιότητά τους. Οι καρποί αυτοί είναι τα μανταρίνια, τα πορτοκάλια, τα λεμόνια και άλλα είδη εσπεριδοειδών.

Οι συνθήκες οι οποίες είναι επιτρεπτές για να συντηρηθούν τα μανταρίνια, είναι ότι πρέπει να βρίσκονται σε θερμοκρασία 4-6 °C, υγρασία 85-95% και συνθήκες ελεγχόμενης ατμόσφαιρας δηλαδή βαρομετρικής πίεσης με 2.5%-5% CO<sub>2</sub> και 10% O<sub>2</sub> [16]. Το χρονικό διάστημα που μπορούν να συντηρηθούν τα μανταρίνια είναι 2-3 μήνες. Το O<sub>2</sub> (οξυγόνο) και CO<sub>2</sub> (διοξείδιο του άνθρακα) επιδρούν στην ένταση της αναπνοής των καρπών. Ο γρήγορος ρυθμός αναπνοής οδηγεί σε μικρή διάρκεια ζωής του προϊόντος. Οπότε είναι χρήσιμο να γίνεται εξέταση των αερίων αυτών που βρίσκονται στην ατμόσφαιρα του χώρου αποθήκευσης. Με την ενίσχυση της περιεκτικότητας της ατμόσφαιρας σε CO<sub>2</sub> ή με την μείωση του O<sub>2</sub> συμβάλλουν στην επιβράδυνση του ρυθμού της αναπνοής και συνεισφέρουν στην αύξηση του χρόνου ζωής του προϊόντος [16].

Επίσης, οι συνθήκες στις οποίες πρέπει να βρίσκονται τα πορτοκάλια ώστε να μπορέσουν να συντηρηθούν και να αποθηκευτούν είναι οι εξής:

Θερμοκρασίες από 4-7°C, υγρασία 85% και ελεγχόμενη ατμόσφαιρα δηλαδή βαρομετρική πίεση με 6% CO<sub>2</sub> και 12% O<sub>2</sub>. Τα πορτοκάλια έχουν διάρκεια ζωής 3-5 μήνες και ενδέχεται να συντηρηθούν για μεγάλο χρονικό διάστημα.

Τέλος, έχουμε τα λεμόνια τα οποία για να συντηρηθούν πρέπει να βρίσκονται σε θερμοκρασίες 10-14 °C, υγρασία 85% και για μεγάλο χρονικό διάστημα σε συνθήκες ελεγχόμενης ατμόσφαιρας δηλαδή βαρομετρικής πίεσης με 5% O<sub>2</sub> και 0-5% CO<sub>2</sub> [16].

Επειδή όμως η εργασία μας, την οποία έχουμε υλοποιήσει πρόκειται για ένα σύστημα παρακολούθησης των συνθηκών περιβάλλοντος ενός χώρου αποθήκευσης τροφίμων, τότε τα τρόφιμα που θα υπάρχουν στον χώρο αποθήκευσής μας θα πρέπει να συντηρούνται για αρκετό μεγάλο χρονικό διάστημα. Για να γίνει αυτό θα πρέπει να συντηρήσουμε τους καρπούς μας με πλαστικό φιλμ. Με άλλα λόγια αφού οι καρποί έχουν υποστεί τις αντίστοιχες μετασυστασιακές επεμβάσεις με μυκητοκτόνα, τα καλύπτουμε με πλαστικό φιλμ ώστε με αυτόν τον τρόπο να γίνεται ο έλεγχος της απώλειας νερού και να αποφεύγεται η φθορά του προϊόντος, μολονότι παράλληλα αναπτύσσεται και ένα είδος τροποποιημένης ατμόσφαιρας γύρω από το προϊόν [16].



Σύμφωνα, όμως με τα αποτελέσματα από τα διαγράμματα που βγάλαμε, συμπεραίνουμε ότι λόγω των συνθηκών του περιβάλλοντος που έγιναν στις καταγραφές από 14 Ιουλίου-17 Ιουλίου, τα τρόφιμα θα πρέπει να διατηρηθούν για κάποιο χρονικό διάστημα στον χώρο αποθήκευσης και έπειτα να τα δώσουν σε καταστήματα για να τεθούν προς πώληση.

## 6. Συμπεράσματα

Το Raspberry Pi είναι ένα πανίσχυρο εργαλείο για τη δημιουργία IoT εφαρμογών. Μπορεί να διαχειριστεί πλήθος αισθητήρων και να προγραμματιστεί εύκολα με διάφορες γλώσσες προγραμματισμού.

Το Raspberry Pi μπορεί να λειτουργήσει ως server και να επιστρέφει ιστοσελίδες οι οποίες θα παρέχουν πληροφορίες για τιμές αισθητήρων αλλά και θα επιτρέπουν τον έλεγχο συσκευών μέσα από αυτές.

Η γλώσσα Python αποτελεί ένα πανίσχυρο εργαλείο για την ανάπτυξη εφαρμογών IoT. Στην εργασία αυτή χρησιμοποιήθηκε για τον έλεγχο αισθητήρων και συσκευών, για τη διαχείριση βάσεων δεδομένων αλλά και για την ανάπτυξη ιστοσελίδων. Μάλιστα, παρατηρήσαμε ότι με πολύ εύκολο τρόπο μπορέσαμε μέσα από την γλώσσα προγραμματισμού Python να δημιουργήσουμε ένα ολοκληρωμένο έργο IoT.

Στα πλαίσια της παρούσας διπλωματικής εργασίας σχεδιάστηκε και υλοποιήθηκε ένα σύστημα παρακολούθησης των συνθηκών ενός χώρου αποθήκευσης τροφίμων. Εξετάσαμε πως το συγκεκριμένο σύστημα παρακολούθησης ανιχνεύει και συλλέγει τα δεδομένα από το χώρο αποθήκευσης με τη βοήθεια των αισθητήρων και τα τροφοδοτεί απομακρυσμένα σε μια σχετική βάση δεδομένων. Ο αισθητήρας που χρησιμοποιήσαμε για να συλλέξουμε τα δεδομένα είναι το Sense Hat το οποίο έχει αρκετούς αισθητήρες για να μπορούμε να ελέγχουμε τις περιβαλλοντικές συνθήκες του χώρου αποθήκευσης.

Με τη χρήση αυτού του αυτοματοποιημένου συστήματος, μπορούμε να αξιολογήσουμε και να παρακολουθήσουμε τα διάφορα περιβαλλοντικά δεδομένα μέσω διαδικτύου ή από το κινητό μας τηλέφωνο. Εάν υπάρξει κάποια αποτυχία, τότε μπορεί επίσης ενημέρωση να γίνει αυτοματοποιημένα.

Παρουσιάσαμε, ποια προγράμματα εγκαταστήσαμε επιπλέον για να λειτουργήσει ο υπολογιστή μας ως server. Εξετάσαμε, επίσης πως δουλεύουν τα δυο τμήματα του κωδικά μας, όπου το πρώτο τμήμα είναι ο κώδικας της ιστοσελίδας μας και το δεύτερο τμήμα που είναι ο κώδικας της βάσης μας, έτσι ώστε με αυτό τον τρόπο να λειτουργήσει σωστά η εφαρμογή μας.

Τέλος το παρών έργο πέρα από τη διαχείριση ενός χώρου αποθήκευσης, μπορεί να χρησιμοποιηθεί με μικρές αλλαγές σε ένα πλήθος εφαρμογών IoT και αυτό γιατί η δομή που ακολουθούμε σε ένα έργο IoT είναι παρόμοια και σε άλλες εφαρμογές.

## Βιβλιογραφία

- [1] Raspberry Pi - The Historical Journey - ELEMENT14. (n.d.). Retrieved May 20, 2021, from <https://www.element14.com/news/download/727060/raspberrypilineinfographic-107499.pdf>.
- [2] The history of the Raspberry Pi with RPI zero and RPI3. (n.d.). Retrieved May 20, 2021, from <https://hitaltech.co.uk/the-history-of-the-raspberry-pi/>.
- [3] Wikimedia Foundation. (2021, October 20). Raspberry Pi. Wikipedia. Retrieved May 20, 2021, from [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi).
- [4] Raspberry pi documentation. Processors. (n.d.). Retrieved May 21, 2021, from <https://www.raspberrypi.com/documentation/hardware/raspberrypi/bcm2711/README.md>.
- [5] Raspberry pi documentation. Camera. (n.d.). Retrieved May 21, 2021, from <https://www.raspberrypi.com/documentation/hardware/camera/>.
- [6] Raspberry pi sense hat - WECL.COM.HK. (n.d.). Retrieved May 14, 2021, from <http://www.wecl.com.hk/distribution/PDF/RaspberryPi/58-32-5300.pdf>.
- [7] Maksimović, M., Vujović, V., Davidović, N., Milošević, V. and Perišić, B., 2014. Raspberry Pi as Internet of things hardware: performances and constraints. *design issues*, 3(8), pp.1-6.
- [8] Raspberry pi documentation. Sense HAT. (n.d.). Retrieved May 21, 2021, from <https://www.raspberrypi.com/documentation/hardware/sense-hat/>.
- [9] Datasheet - Arrow. (n.d.). Retrieved May 21, 2021, from <http://static6.arrow.com/aropdfconversion/ca1b2b50fe1fc1862deb22f28e95bc7645e4116b/RPI4B-original.pdf>.
- [10] Longdon, A., Ed, BeeGee, Conzelmann, N., Chin, O. K., Mo, Crawford, J., Rodrigues, R., Wright, O. L., Fourie, J., Steve, Banerjee, S. B., Maple, J., Matsumunyane, B.-B. B., McKendree, C., Chinkey, & Peck, K. (2021, September 7). Raspberry pi: Install apache + mysql + PHP (lamp server). Random Nerd Tutorials. Retrieved May 20, 2021, from <https://randomnerdtutorials.com/raspberry-pi-apache-mysql-php-lamp-server/>.
- [11] Wallen, J. (2019, April 26). How to set, change, and recover a mysql root password. TechRepublic. Retrieved May 21, 2021, from <https://www.techrepublic.com/article/how-to-set-change-and-recover-a-mysql-root-password/>.
- [12] Wikimedia Foundation. (2021, October 20). Linux. Wikipedia. Retrieved May 21, 2021, from <https://en.wikipedia.org/wiki/Linux>.
- [13] Mozilla WebThings documentation. (n.d.). Retrieved May 21, 2021, from <https://iot.mozilla.org/docs/>.
- [14] Wikimedia Foundation. (2018, April 29). Internet of things. Wikipedia. Retrieved May 21, 2021, from [https://en.wikipedia.org/wiki/Internet\\_of\\_Things](https://en.wikipedia.org/wiki/Internet_of_Things).

- [15] Τσοτάκης, Α., 2018. Σύγχρονες τεχνολογίες στην εφοδιαστική αλυσίδα τροφίμων (Master's thesis, Πανεπιστήμιο Πειραιώς).
- [16] Εσπεριδοειδή: αποθήκευση – συντήρηση - farmacon - blog - H... (n.d.). Retrieved May 20, 2021, from <https://blog.farmacon.gr/katigories/tehniki-arthrografia/syntirisi-apothikefsi/item/1301-esperidoeidi-apothikefsi-syntirisi>.
- [17] EFET. (n.d.). Retrieved May 19, 2021, from [https://www.efet.gr/files/F5242\\_odhgap.pdf](https://www.efet.gr/files/F5242_odhgap.pdf).
- [18] De Ancos B;Rodrigo MJ;Sánchez-Moreno C;Pilar Cano M;Zacarías L; (n.d.). Effect of high-pressure processing applied as pretreatment on carotenoids, flavonoids and vitamin C in juice of the sweet oranges 'navel' and the red-fleshed 'Cara Cara'. Food research international (Ottawa, Ont.). Retrieved June 30, 2021, from <https://pubmed.ncbi.nlm.nih.gov/32331674/>.
- [19] van Holsteijn, F. and Kemna, R., 2018. Minimizing food waste by improving storage conditions in household refrigeration. Resources, Conservation and Recycling, 128, pp.25-31.
- [20] Sutton, A.O., Strickland, D. and Norris, D.R., 2016. Food storage in a changing world: implications of climate change for food-caching species. Climate Change Responses, 3(1), pp.1-25.
- [21] Abeliotis, K., Lasaridi, K., Costarelli, V. and Chroni, C., 2015. The implications of food waste generation on climate change: The case of Greece. Sustainable production and consumption, 3, pp.8-14.

## Παράρτημα Α

### Εγκατάσταση του λειτουργικού συστήματος Raspberry Pi Os

Ο πιο απλός τρόπος για να εγκαταστήσουμε το λειτουργικό σύστημα Raspberry Pi Os είναι με τη χρήση ενός νέου εργαλείου που αποκαλείται Raspberry Pi Imager. Είναι μια εφαρμογή που μέσα σε μερικά βήματα κατεβάζει το λειτουργικό σύστημα που θα διαλέξουμε, σε μορφή αρχείου εικονικού δίσκου και αργότερα το εγκαθιστά μέσα στην κάρτα Micro-Sd.

Για να το κάνουμε αυτό πρέπει να πάρουμε ένα card reader και μια κάρτα Micro-Sd. Το card reader είναι ένας αναγνώστης μνήμης ο οποίος δέχεται μνήμες τύπου Sd ή τύπου Micro-Sd. Βρίσκουμε τέτοιους αναγνώστες μνήμης και σε συγκεκριμένους υπολογιστές. Επίσης χρησιμοποιήσαμε την Micro-Sd η οποία έχει μνήμη 32 GB. Οι κάρτες μαζί με τον αντάπτορα είναι αρκετά οικονομικές και αφού οι προδιαγραφές είναι σωστές δεν αναμένονται προβλήματα συμβατότητας. Στις εικόνες 21 και 22 παρατηρούμε την Micro-Sd και το card reader που χρησιμοποιήσαμε για να κατεβάσουμε το λειτουργικό σύστημα Raspberry Pi Os.



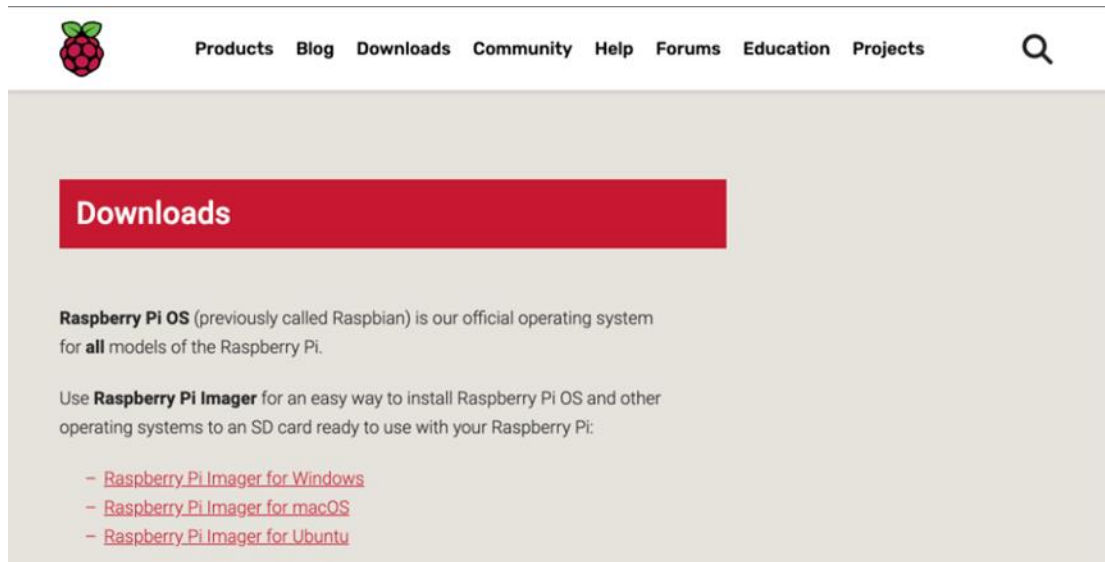
Εικόνα 21: Micro-Sd 32 GB και Αντάπτορας (Πηγή:

<https://www.skroutz.gr/s/20701680/Kingston-Canvas-Select-Plus-microSDHC-32GB-U1-V10-A1-with-Adapter.html>)



Εικόνα 22: Card Reader(Πηγή: <https://www.e-shop.gr/esperanza-ea134o-micro-sd-usb-20-card-reader-orange-p-PER.572836> )

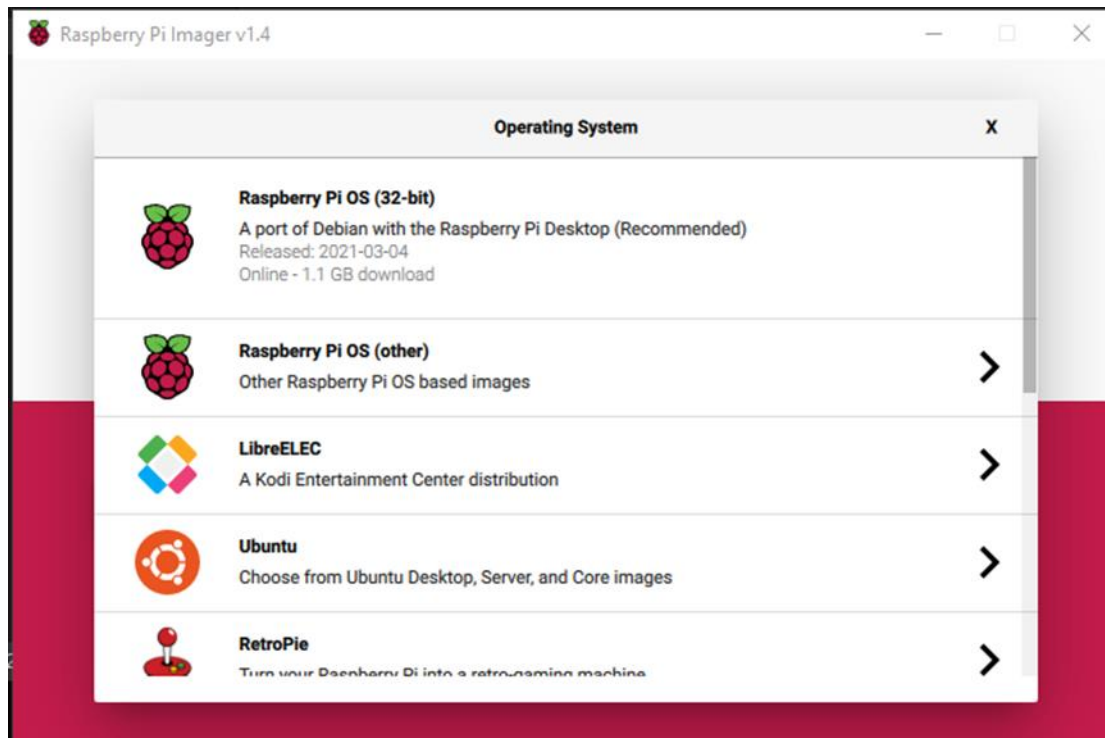
Πρώτα τοποθετούμε την Micro-Sd στο card reader και είμαστε έτοιμοι να βάλουμε το card reader στη θύρα USB του υπολογιστή μας όπου θα κατεβάσουμε το λειτουργικό μας σύστημα. Αφού τοποθετήσουμε το card reader στον υπολογιστή μας, θα πρέπει να επισκεφτούμε τον ιστότοπο [raspberrypi.org](http://raspberrypi.org) και θα βρούμε την εφαρμογή Raspberry Pi Imager στην αρχή της σελίδας στα “Downloads” όπως φαίνεται στην Εικόνα 23.



Εικόνα 23: Εγκατάσταση Raspberry Pi Imager

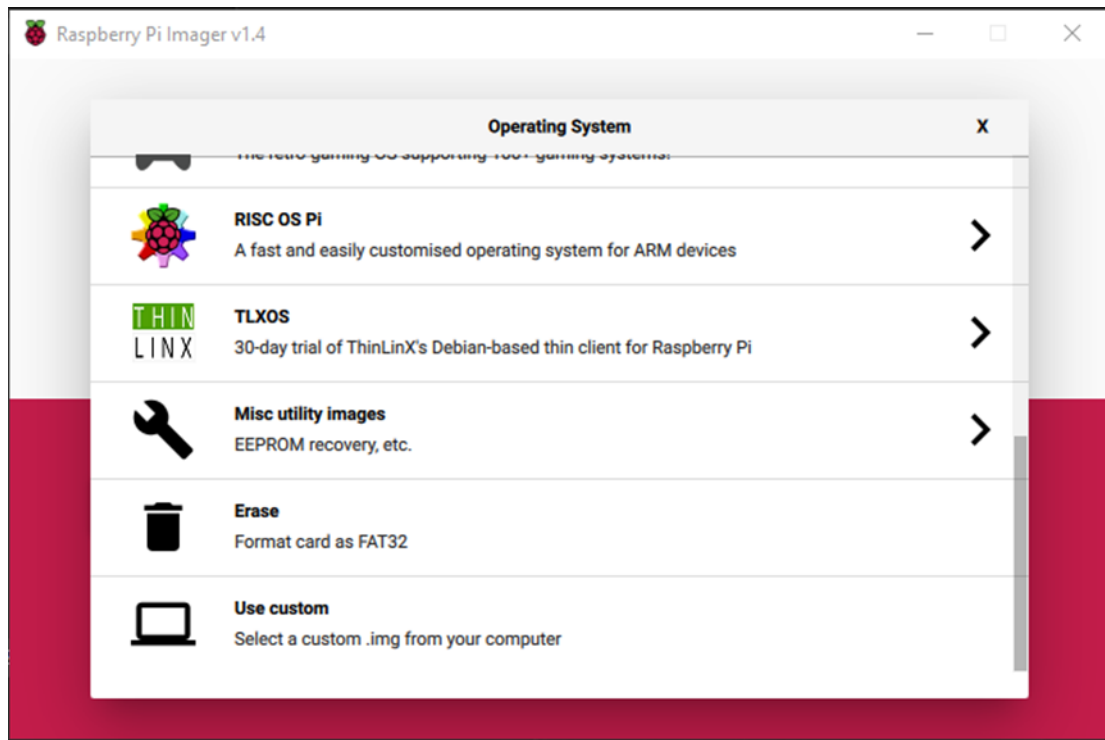
Ανάλογα τον υπολογιστή που θα χρησιμοποιήσουμε για να κάνουμε εγγραφή στην Micro-Sd, θα εγκαταστήσουμε και την αντίστοιχη έκδοση. Το εργαλείο αυτό υπάρχει ως διαθέσιμο για Windows, Mac-Os και Ubuntu Linux. Όταν το κατεβάσουμε, κάνουμε την εγκατάσταση του και ενδέχεται πλέον να τρέξουμε την εγκατεστημένη εφαρμογή, αναζητώντας την ανάμεσα στις εφαρμογές μας.

Στο πρώτο βήμα διαλέγοντας το “CHOOSE OS”, η εφαρμογή μας συνιστά να επιλέξουμε το λειτουργικό σύστημα. Εκεί ενδέχεται να παρατηρήσουμε ότι ως πρώτο μας παρουσιάζεται το προτεινόμενο Raspberry Pi Os(32-bit) με μέγεθος 1GB περίπου.



Εικόνα 24: Λειτουργικό Σύστημα Raspberry Pi Os

Ακριβώς απο κάτω θα κοιτάζουμε το Raspberry Pi Os με την λέξη other που αν το διαλέξουμε θα μας εμφανίσει δυο ακόμη εκδόσεις, την μικρότερη σε μέγεθος με το όνομα Lite η οποία δεν έχει γραφικό περιβάλλον αλλά και άλλη μια την Full όπου το μέγεθος του αρχείου εγκατάστασης φτάνει τα 2.5 GB και περιέχει περισσότερες εφαρμογές από την προτεινόμενη έκδοση. Μεταξύ των επιπρόσθετων εφαρμογών είναι η Wolfram Mathematica, ικανές εφαρμογές για την εισαγωγή στον προγραμματισμό όπως το scratch, μια σουίτα προγραμμάτων αντίστοιχη του Microsoft Office όπου ονοματίστηκε ως LibreOffice, μερικά παιχνίδια και κάποιες άλλες εφαρμογές.



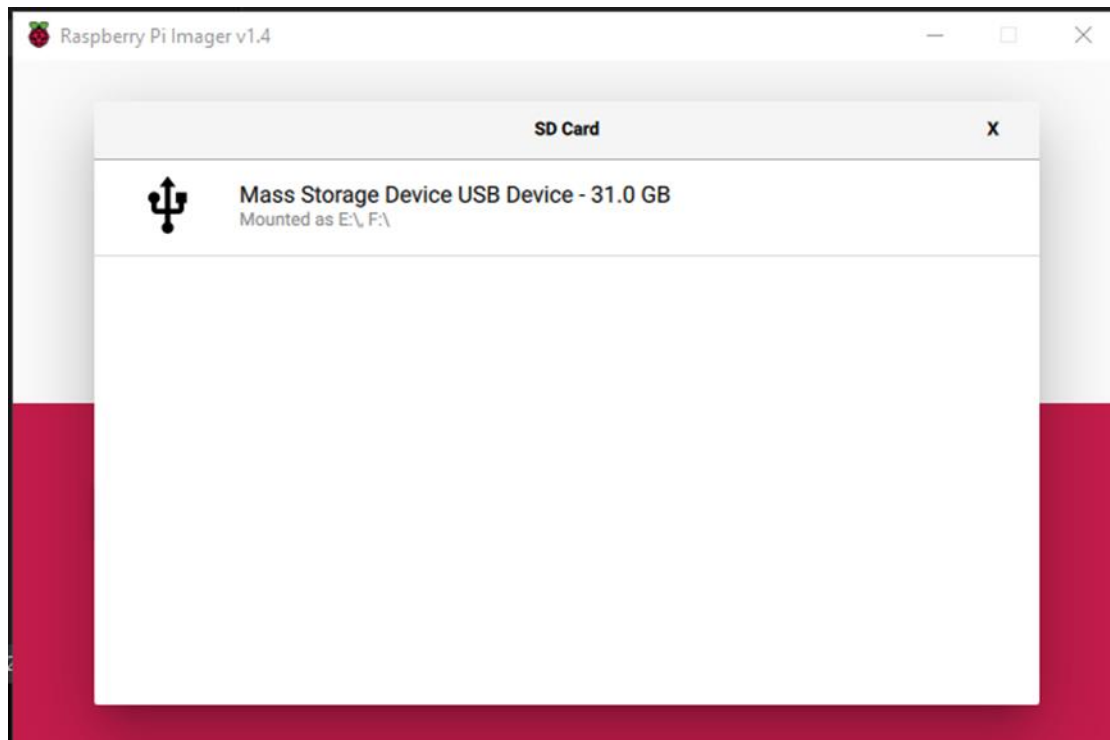
Εικόνα 25: Erase και Use custom

Στην Εικόνα 25, παρατηρούμε το τέλος της λίστας με τα διαθέσιμα και πολύ βοηθητικά εργαλεία όπως για παράδειγμα το εργαλείο Erase με το οποίο επιτρέπεται να σβήσουμε την Micro-Sd ενώ με το εργαλείο Use custom έχουμε την δυνατότητα να εγκαταστήσουμε ένα άλλο λειτουργικό σύστημα που δεν υπάρχει στην παραπάνω λίστα.

Για να προχωρήσουμε με την εγκατάσταση θα διαλέξουμε την προτεινόμενη διανομή του Raspberry Pi Os με μέγεθος 1.1 GB, το οποίο φαίνεται στην Εικόνα 24. Διαλέγουμε αυτή την εκδόση γιατί μπορούμε αρκετά εύκολα να εγκαταστήσουμε και τις υπόλοιπες εφαρμογές που περιέχει η πλήρης έκδοση, την οποία αναφέραμε νωρίτερα.

Προσθέτουμε ή επισυνάπτουμε την Micro-Sd στον υπολογιστή μας και την διαλέγουμε από το επόμενο βήμα που είναι το CHOOSE SD CARD όπου φαίνεται στην Εικόνα 26.





Εικόνα 26: Επιλογή κάρτας Micro-Sd

Στο τελευταίο βήμα διαλέγουμε το WRITE και το πρόγραμμα αναλαμβάνει πρωτίστως να κατεβάσει από το διαδίκτυο το λειτουργικό σύστημα που διαλέξαμε στο πρώτο βήμα και έπειτα να το εγκαταστήσει στην Micro-Sd, αφού διαγράψει πρώτα τα περιεχόμενά της.

Μόλις τελειώσει η διαδικασία εμφανίζεται ένα μήνυμα που μας ζητά να βγάλουμε την Micro-Sd από τον υπολογιστή μας.

Αφαιρούμε την Micro-Sd με ασφάλη κατάργηση και είμαστε έτοιμοι να την βάλουμε στο Raspberry Pi. Αφού τοποθετήσουμε την Micro-Sd στη συσκευή, θα συνεχίσουμε με την σύνδεση των εξαρτημάτων με το Raspberry Pi.

### **Σύνδεση των εξαρτημάτων με το Raspberry Pi**

Παρουσιάζουμε τα εξαρτήματα που χρειαστήκαμε και χρησιμοποιήσαμε για να κάνουμε τη σύνδεση στο Raspberry Pi. Εκτός από το card reader και την κάρτα Micro-Sd χρησιμοποιήσαμε και κάποια άλλα εξαρτήματα. Τα εξαρτήματα αυτά είναι:

Ένα τροφοδοτικό: Το τροφοδοτικό αυτό είναι το γνήσιο με καλώδιο type-c, γιατί δεν υπάρχουν αρκετά τροφοδοτικά τα οποία δουλεύουν σωστά με το Raspberry Pi 4 και φαίνεται στην Εικόνα 27. Επίσης έχει τα εξής χαρακτηριστικά:

- AC 100-240V 50/60Hz για είσοδο.
- DC 5.1V 3A για έξοδο.
- 15.3W για μέγιστη ισχύ εξόδου.
- 1.5m 18 AWG δεσμευμένο καλώδιο.
- USB-C βύσμα εξόδου.



Εικόνα 27: Τροφοδοτικό(Πηγή: <https://www.elektor.com/official-eu-power-supply-for-raspberry-pi-4-black>)

Ένα καλώδιο HDMI για σύνδεση στην οθόνη: Για τη συσκευή μας το Raspberry Pi 4 χρειαστήκαμε να αποκτήσουμε ένα καλώδιο από Micro-HDMI σε HDMI μιας που οι διαθέσιμες θύρες του είναι τύπου Micro-HDMI. Το καλώδιο αυτό ονομάζεται Raspberry Pi Micro HDMI cable και φαίνεται στην Εικόνα 28.



Εικόνα 28: Καλώδιο HDMI(Πηγή: <https://www.elektor.com/official-hdmi-cable-for-raspberry-pi-4-white-1-m>)

Πληκτρόλογιο και ποντίκι: Αυτά είναι δυο εξαρτήματα τα οποία τα έχουμε ήδη. Επίσης για οποιαδήποτε συσκευή θα μπορούσαν να λειτουργήσουν αρκεί να συνδέονται με θύρα USB.

Όσον αφορά για τη σύνδεση στο δίκτυο χρησιμοποιήσαμε ένα καλώδιο δικτύου για ενσύρματο δίκτυο. Αλλά υπάρχει και η δυνατότητα του ασυρμάτου δικτύου αν έχουμε τις τελευταίες γενεές Raspberry Pi. Το καλώδιο δικτύου που χρησιμοποιήσαμε έχει μήκος περίπου 1 ή 1.5 μέτρο και φαίνεται στην Εικόνα 29.



Εικόνα 29: Καλώδιο δικτύου

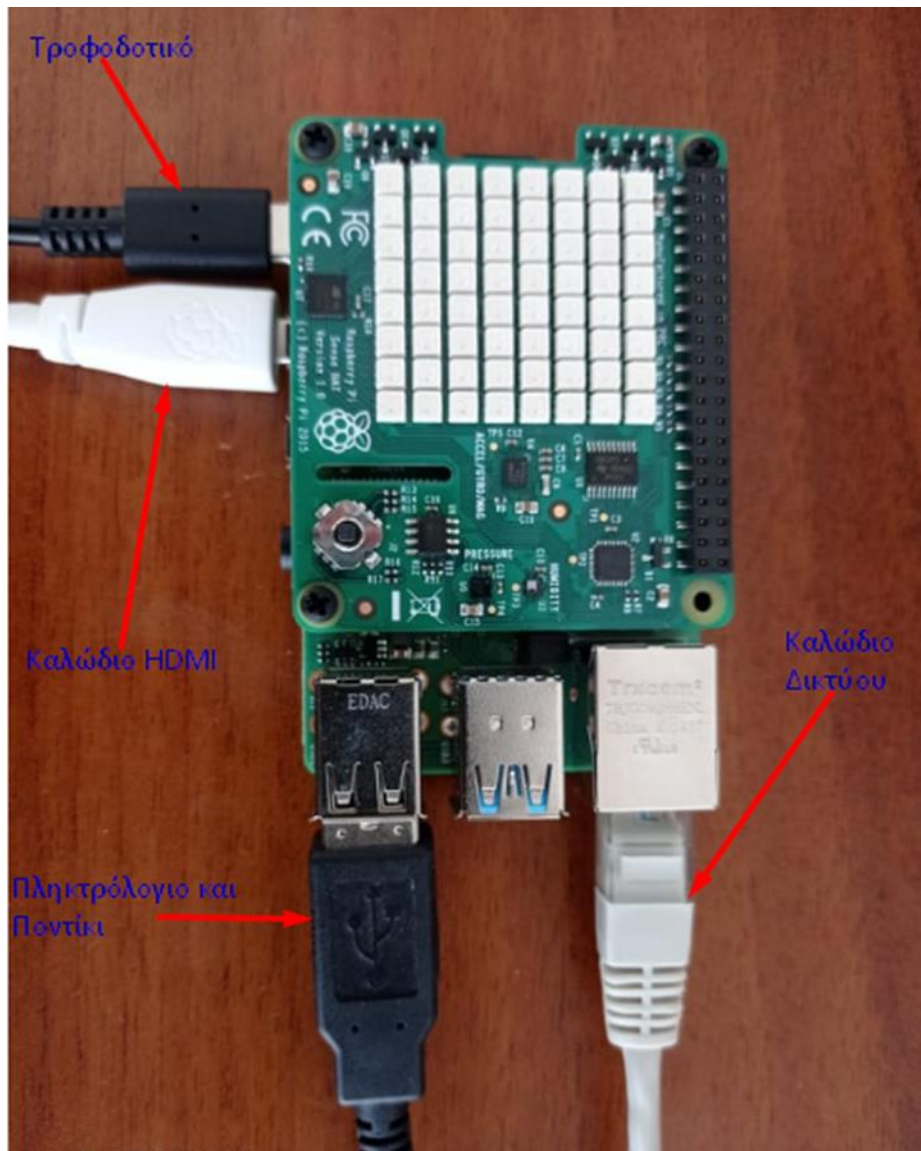
Ένα Sense Hat: Το Sense Hat όπως φαίνεται στην Εικόνα 30 τοποθετείται πάνω στους 40 ακροδέκτες GPIO που έχει το Raspberry Pi 4.





Εικόνα 30: Sense Hat

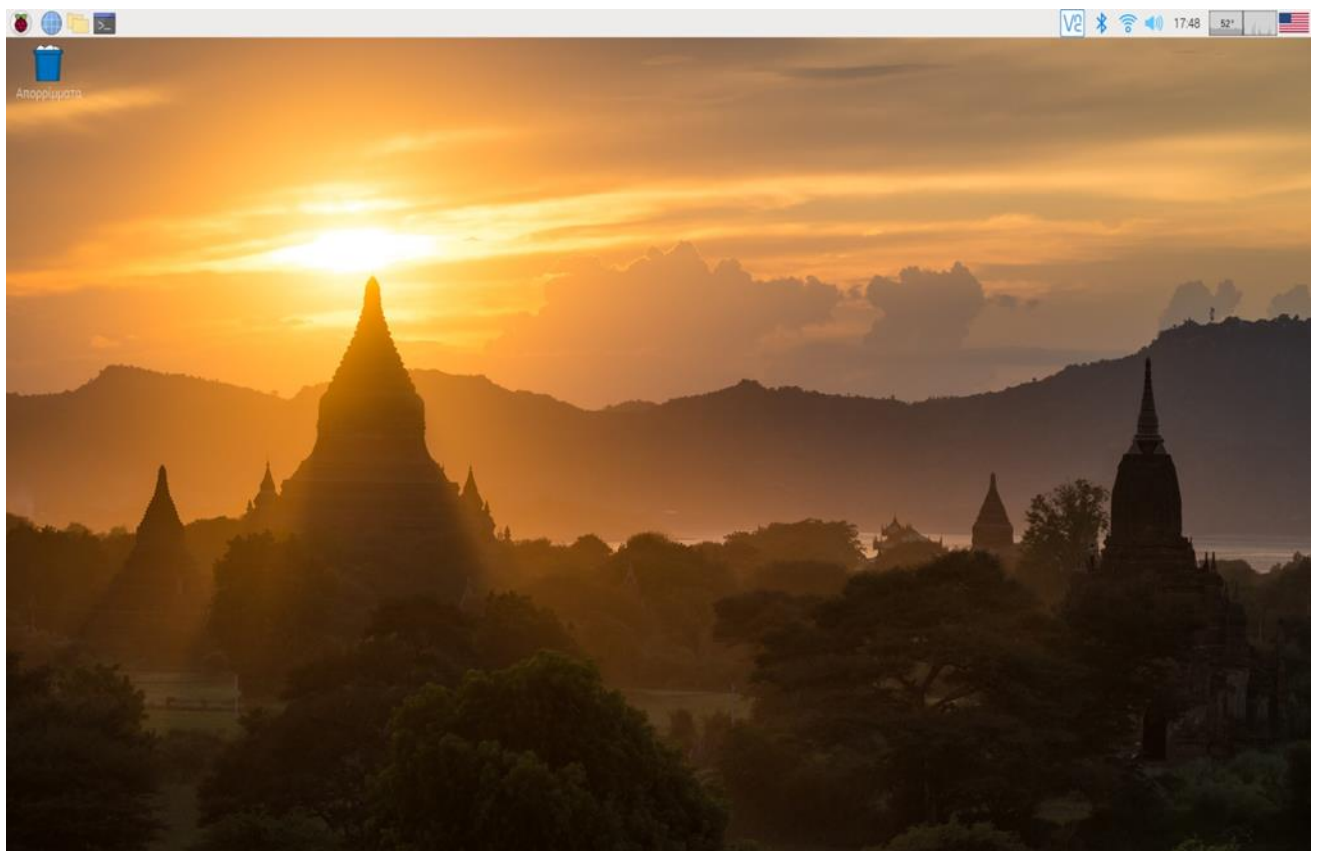
Με τα εξαρτήματα αυτά που είπαμε τώρα, είμαστε σε θέση να τα συνδέσουμε στη συσκευή Raspberry Pi και να μπούμε στο περιβάλλον του Raspberry Pi Os. Η συσκευή Raspberry Pi που θα χρησιμοποιήσουμε είναι το Raspberry Pi 4 Model B με 2 GB RAM. Χρησιμοποιούμε την πιο οικονομική έκδοχή αυτή με τα 2 GB RAM, όπου το κόστος της φτάνει γύρω στα 40 ευρώ. Επίσης θα αναφερθούμε στην σύνδεση των εξαρτημάτων με την συσκευή Raspberry Pi. Πριν ξεκινήσουμε όμως, θα παρατηρήσουμε στην Εικόνα 31, την σύνδεση των εξαρτημάτων χρησιμοποίησης με το Raspberry Pi και θα εξετάσουμε πως επιτυγχάνεται η συγκεκριμένη σύνδεση. Στην Εικόνα 31 λόγω επειδή οι θύρες USB είναι μαζί δηλαδή η μια πάνω και η άλλη κάτω, τότε στην κάτω θύρα έχουμε συνδέσει το πληκτρολόγιο και στην πάνω θύρα έχουμε συνδέσει το ποντίκι. Επίσης για την συγκεκριμένη εικόνα φαίνεται και το Sense Hat που το έχουμε τοποθετήσει πάνω στο Raspberry Pi.



Εικόνα 31: Εξαρτήματα Χρησιμοποίησης

Αρχικά, ξεκινάμε με την θέση της κάρτας Micro-Sd που έχουμε προετοιμάσει, στην ειδική υποδοχή της συσκευής μας. Προτιμότερο είναι να κάνουμε τις συνδέσεις με ορισμένη σειρά, αν και το πιο βασικό που πρέπει να προσέξουμε είναι να βάλουμε πρώτα την κάρτα Micro-Sd και τελευταίο το τροφοδοτικό. Βάζουμε λοιπόν την κάρτα Micro-Sd έχοντας αναποδογυρίσει το Raspberry Pi και προσέχουμε να βάλουμε την κάρτα με τις επαφές της να βλέπουν προς την πλακέτα του Raspberry Pi. Τοποθετούμε το πληκτρολόγιο και το ποντίκι στις θύρες USB. Μπορούμε να προτιμήσουμε για τις συγκεκριμένες συσκευές τις θύρες με μαύρο χρώμα, αφού οι συσκευές αυτές δεν κάνουν εφαρμογή του πιο γρήγορου USB3 που έχει το Raspberry Pi 4 με το χαρακτηριστικό μπλε χρώμα. Επίσης συνδέουμε το καλώδιο HDMI με το Raspberry Pi στην πρώτη θύρα, πιο συγκεκριμένα την θύρα με ετικέτα HDMI0 και κατά προτίμηση πρώτα πάνω στο Raspberry Pi και ύστερα στην οθόνη ή στην

τηλεόρασή μας. Εκτός από την εικόνα μεταφέρεται και ο ήχος ψηφιακά διαμέσου του καλωδίου HDMI στην τηλεόρασή μας ή στην οθόνη, εφόσον αυτή έχει ηχεία. Στην περίπτωση που έχουμε οθόνη χωρίς ηχεία, τότε ενδέχεται να κάνουμε χρήση της διπλανής υποδοχής τύπου Mini Jack για να συνδέσουμε ακουστικά ή ηχεία. Για να συνδέσουμε το Raspberry Pi στο ενσύρματο δίκτυο, χρησιμοποιούμε ένα καλώδιο δικτύου στην ανάλογη υποδοχή. Πρέπει να συνδέσουμε ακόμη το τροφοδοτικό στη θύρα USB Type C, όπου είναι και το τελικό βήμα για τη σύνδεση μας με το Raspberry Pi. Εφόσον η τηλεόρασή μας ή η οθόνη μας είναι αναμμένη, συνδέουμε το τροφοδοτικό σε μια πρίζα. Τέλος αν έχουν πάει όλα καλά, τότε θα παρατηρήσουμε να φορτώνεται το λειτουργικό σύστημα. Όταν τελειώσει να φορτώνει το λειτουργικό τότε μπαίνουμε κατευθείαν στο περιβάλλον του Raspberry Pi Os, το οποίο φαίνεται στην Εικόνα 32.



Εικόνα 32: Περιβάλλον του Raspberry Pi Os

## Εγκατάσταση Sense Hat

Παρουσιάζουμε τα βήματα της εγκατάστασης του Sense Hat. Για να δουλέψει σωστά, το Sense HAT απαιτεί έναν ενημερωμένο πυρήνα, το I2C να είναι ενεργοποιημένο και κάποιες βιβλιοθήκες για να ξεκινήσετε [8]. Τα βήματα της εγκατάστασης είναι τα εξής:

- Να κάνουμε ενημέρωση στις λίστες με τις διαθέσιμες εφαρμογές με την apt. Η εντολή είναι η `sudo apt update`.
- Να κάνουμε εγκατάσταση του πακέτου `sense-hat` για να σιγουρευτούμε ότι ο πυρήνας θα είναι ενημερωμένος, θα ενεργοποιήσει το I2C και θα εγκαταστήσει τις αναγκαίες βιβλιοθήκες και προγράμματα με την εντολή `sudo apt install sense-hat` [8].
- Σε περίπτωση που μπορεί να απαιτείται επανεκκίνηση είναι όταν το I2C απενεργοποιηθεί ή ο πυρήνας δεν είναι ενημερωμένος πριν από την εγκατάσταση και γίνεται με την εντολή `sudo reboot`.



## Παράρτημα Β

### Εγκατάσταση Apache

Τα προγράμματα που εγκαταστήσαμε για να λειτουργήσει η εφαρμογή μας είναι ο Apache, PHP και MySQL [10]. Περιγράφουμε αναλυτικά τα βήματα με τα οποία εγκαταστάθηκαν οι συγκεκριμένες εφαρμογές. Πριν ξεκινήσουμε την εγκατάσταση του Apache πρώτα πρέπει να ανοίξουμε το terminal και να κάνουμε ενημέρωση το Raspberry Pi. Αυτό γίνεται πληκτρολογώντας στο terminal την εξής εντολή:

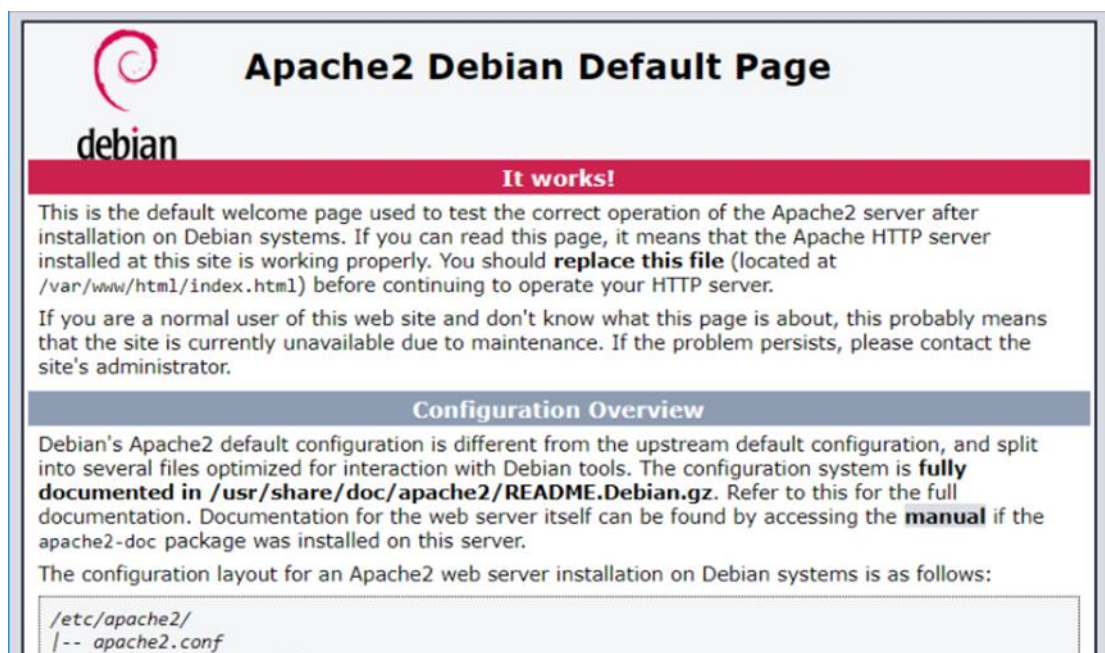
```
sudo apt update && sudo apt upgrade -y
```

Αφού κάναμε την ενημέρωση στο Raspberry Pi, τώρα θα ξεκινήσουμε με την εγκατάσταση του Apache [10]. Πρώτα θα κάνουμε εγκατάσταση της βιβλιοθήκης apache2 και ύστερα θα πρέπει να εκτελέσουμε την παρακάτω εντολή:

```
sudo apt install apache2 -y
```

Έτσι με αυτόν τον τρόπο εγκαταστήσαμε τον Apache. Για να τσεκάρουμε ότι η εγκατάσταση πήγε καλά αρκεί να πληκτρολογήσουμε την IP του Raspberry Pi 4.

Σε έναν περιηγητή διαδικτύου και από προεπιλογή ο Apache βάζει ένα HTML αρχείο στον φάκελο του διαδικτύου. Οπότε στην οθόνη μας θα εμφανιστεί ότι δουλεύει ο Apache και φαίνεται στην Εικόνα 33.



Εικόνα 33: Έλεγχος σωστής λειτουργίας του Apache web Server [10]

## Εγκατάσταση PHP

Περιγράφουμε την εγκατάσταση της PHP στο Raspberry Pi 4 [10]. Για να κάνουμε την εγκατάσταση και αφού είμαστε στον φάκελο του διαδικτύου ο οποίος είναι ο `/var/www/html` πληκτρολογούμε στο terminal την εξής εντολή:

```
sudo apt install php -y
```

Επίσης, μπορούμε να καταργήσουμε το αρχείο HTML και να φτιάξουμε ένα σενάριο PHP για να τσεκάρουμε την εγκατάσταση πληκτρολογώντας τις εξής εντολές:

```
sudo rm index.html
```

```
sudo nano index.php
```

Όταν πληκτρολόγουμε την εντολή `sudo nano index.php` τότε μας επιτρέπει να διαμορφώσουμε το αρχείο `index.php` και θα μπορούμε να γράψουμε μέσα στο αρχείο αυτό τον ακόλουθο κώδικα ώστε να δούμε στην οθόνη μας το μήνυμα “hello world” [10]. Το κομμάτι κώδικα που θα προσθέσουμε στο αρχείο `index.php` είναι:

```
<?php echo "hello world"; ?>
```

Αποθηκεύουμε το αρχείο πατώντας τα πλήκτρα `Ctrl+X`, ακολουθούμενο από το πλήκτρο `y` και το πλήκτρο `Enter` για να βγούμε από το αρχείο `index.php`. Κάνουμε επανεκκίνηση του Apache με την εντολή:

```
sudo service apache2 restart
```

Για να τσεκάρουμε εάν το Apache εξυπηρετεί αρχεία.php, πρέπει να πληκτρολογήσουμε την IP του Rpi 4 σε έναν περιηγητή διαδικτύου και θα εμφανίζει το μήνυμα " hello world " από το αρχείο `index.php` που ήδη φτιάξαμε νωρίτερα [10].

Τέλος αν όλα πάνε καλά τότε μπορούμε να καταργήσουμε το αρχείο `index.php` από τον φάκελο του διαδικτύου ο οποίος είναι ο `/var/www/html` τότε αυτό γίνεται με την εντολή:

```
sudo rm index.php
```

## Παράρτημα Γ

### Εγκατάσταση MySQL (MariaDB Server)

Κάνουμε την εγκατάσταση της MySQL, η οποία είναι η πιο περιζήτητη σχεσιακή βάση δεδομένων ανοιχτού κώδικα. Για να γίνει αυτό θα πρέπει να εγκαταστήσουμε τα πακέτα MySQL Server (MariaDB Server) και PHP-MYSQL αφού εκτελέσουμε τις παρακάτω εντολές:

```
sudo apt install mariadb-server php-mysql -y
```

```
sudo service apache2 restart
```

Με την εγκατάσταση της MySQL Server (MariaDB Server), προτείνεται για την ασφάλεια της εγκατάστασης να εκτελέσουμε την εντολή:

```
sudo mysql_secure_installation
```

Με την εκτέλεση της παραπάνω εντολής θα μας ερωτηθεί να εισάγουμε τον προσωρινό κωδικό για το root (τύπος ενός ασφαλούς password) και πατάμε το πλήκτρο Enter [10].

Πληκτρολογούμε Y και πατάμε το πλήκτρο Enter για να καθορίσουμε τον κωδικό πρόσβασης για το root.

Πληκτρολογούμε τον κωδικό στον καινούριο κωδικό όπου αυτό γίνεται με την προτροπή και με το πάτημα του πλήκτρου Enter.

Υπάρχει κάτι το οποίο είναι πολύ σημαντικό και δεν πρέπει να ξεχάσουμε είναι ο κωδικός για το root τον οποίο θα τον χρειαστούμε αργότερα.

Πληκτρολογούμε Y και το πλήκτρο Enter για να βγάλουμε τους ανώνυμους χρήστες.

Πληκτρολογούμε Y και το πλήκτρο Enter ώστε να μην είναι δυνατή η πρόσβαση για την σύνδεση του root από απόσταση [10].

Πληκτρολογούμε Y για να ακυρώσουμε ή να αναιρέσουμε τη δοκιμαστική βάση δεδομένων και να μην επιτρέπεται η πρόσβαση σε αυτήν.

Πληκτρολογούμε Y για να διαβάσει ξανά τους πίνακες προνομιών.

Όταν τελειώσει η ολοκλήρωση της εγκατάστασης θα εμφανιστεί στην οθόνη μας το μήνυμα “Thanks for using MariaDB!” όπως φαίνεται στην Εικόνα 34:

```
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Εικόνα 34: Ολοκλήρωση εγκατάστασης του Server MariaDB [10]

Τέλος χρειάστηκαν και κάποιες επιπλέον εντολές για την εγκατάσταση της MySQL οι οποίες είναι:

```
$ sudo mysql -u root # I had to use "sudo" since is new installation

mysql> USE mysql;
mysql> UPDATE user SET plugin='mysql_native_password' WHERE User='root';
mysql> FLUSH PRIVILEGES;
mysql> exit;

$ sudo service mysql restart
```

Σύμφωνα με την παραπάνω εικόνα εξηγούμε τις παραπάνω εντολές. Μπαίνουμε μέσα στο περιβάλλον της MySQL ως χρήστης root [11]. Η δήλωση **USE** λέει στη MySQL να κάνει χρήση της ονομαζόμενης βάσης δεδομένων ως την προεπιλεγμένη(τρέχουσα) βάση δεδομένων για τις επόμενες δηλώσεις. Αυτή η δήλωση χρειάζεται κάποιο προνόμιο για τη βάση δεδομένων ή κάποιο αντικείμενο μέσα σε αυτήν [11]. Για να μπορέσουμε να συνδεθούμε με τον κωδικό πρόσβασης, πρέπει να αλλάξουμε την δήλωση από `auth_socket` `mysql_native_password`. Διακόπτουμε τη χρήση της επιλεγμένης βάσης και κάνουμε επανεκκίνηση την MySQL για να πάρει τις καινούριες ρυθμίσεις.

Υπάρχει, ακόμη ένα τελευταίο πρόγραμμα το οποίο είναι το phpMyAdmin [10]. Για να το εγκαταστήσουμε, πρέπει να πληκτρολογήσουμε στο terminal του Raspberry Pi την παρακάτω εντολή:

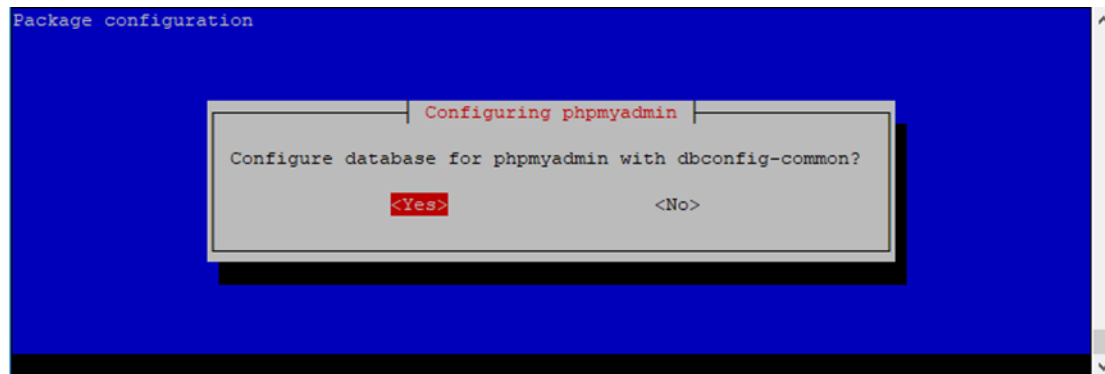
**sudo apt install phpmyadmin -y**

Το πρόγραμμα εγκατάστασης του PHPMyAdmin θα μας θέσει μερικές ερωτήσεις. Θα κάνουμε χρήση του `dbconfig-common`.

Θα διαλέξουμε τον Apache2 όταν μας ζητηθεί και πατάμε το πλήκτρο Enter

Στη ρύθμιση του phpmyadmin πατάμε OK [10].

Όταν εμφανιστεί στην οθόνη μας το ερώτημα ότι η διαμόρφωση της βάσης δεδομένων για το phpmyadmin με το dbconfig-common; Τότε σε αυτή την περίπτωση θα πατήσουμε Yes το οποίο φαίνεται στην παρακάτω εικόνα.



(Πηγή: <https://randomnerdtutorials.com/raspberry-pi-apache-mysql-php-lamp-server/>)

Πληκτρολογούμε τον κωδικό πρόσβασης μας και πατάμε OK.

Ενεργοποιούμε την επέκταση PHP MySQLi και επανεκκινούμε τον Apache2 για να γίνουν οι αλλαγές [10].

```
sudo phpenmod mysql
```

```
sudo service apache2 restart
```

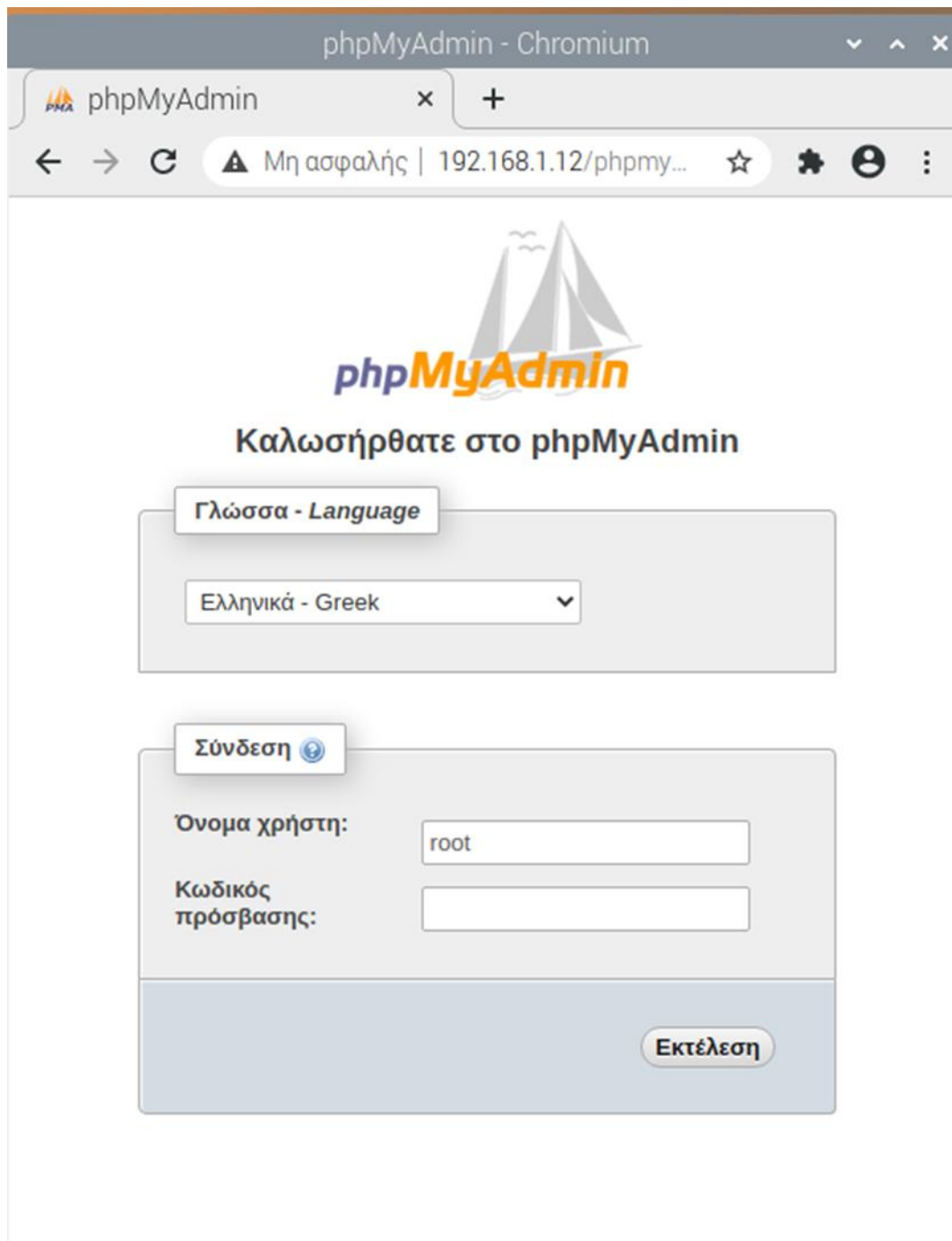
Ανοίγουμε, μετά τον browser του προσωπικού μας υπολογιστή και πληκτρολογούμε την διεύθυνση IP του Raspberry Pi της συσκευής μας ακολουθούμενη από το /phpmyadmin(στην δική μου περίπτωση είναι <http://192.168.1.12>), όπου θα καταλάβουμε πιθανότατα τη σελίδα του σφάλματος "Not Found" [10].

Σε αυτό εδώ το σημείο, θα πρέπει να μετακινήσουμε τον φάκελο phpmyadmin στο /var/www/html, όπου αυτό γίνεται με την εκτέλεση της ακόλουθης εντολής:

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

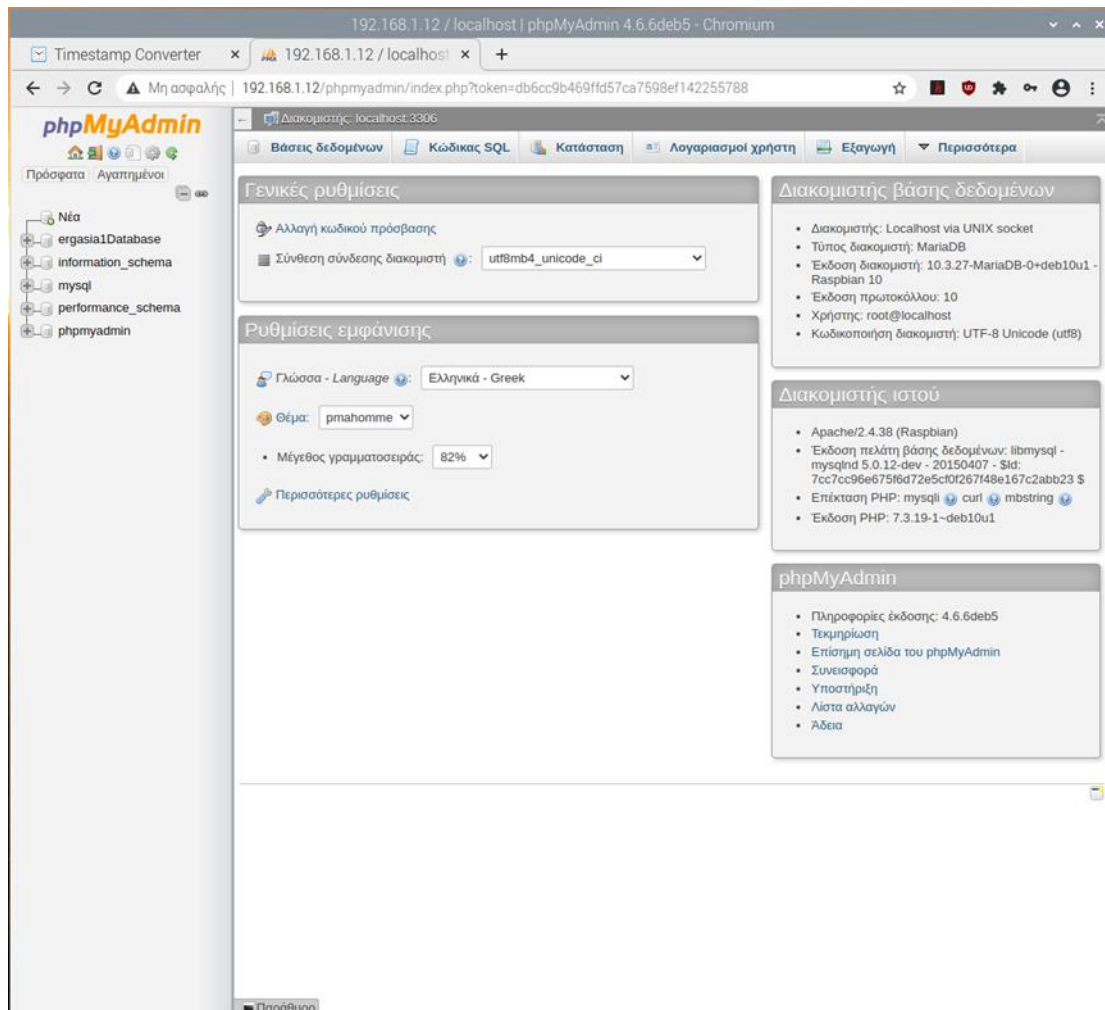
Αν βάλουμε στη σειρά το ένα μετά το άλλο τα αρχεία, τότε θα πρέπει να επιστρέψουμε στο φάκελο phpmyadmin. Αυτό γίνεται όταν πληκτρολογήσουμε την εντολή ls θα μας εμμανίσει τον φάκελο phpmyadmin [10].

Φορτώνουμε ξανά την ιστοσελίδα μας(<http://192.168.1.12>), όπου θα πλοηγηθούμε στην σελίδα σύνδεσης για το διαδικτυακό περιβάλλον phpMyAdmin:



Για να μεταβούμε μέσα στο διαδικτυακό περιβάλλον του phpMyAdmin πρέπει να δώσουμε το καθορισμένο όνομα χρήστη το οποίο είναι Username = root και τον κωδικό πρόσβασης τον οποίο τον ορίσαμε κατά την εγκατάσταση. Ο κωδικός αυτός είναι 12345 όπου τον βρίσκουμε στο τμήμα του κώδικα της βάσης και της ιστοσελίδας μας. Είναι απαραίτητος ώστε να λειτουργήσει σωστά η εφαρμογή, γιατί διαφορετικά αν κάναμε κάτι λάθος στην εγκατάσταση για το phpMyAdmin ειδικά για τον κωδικό πρόσβασης, τότε δε θα λειτουργούσε η εφαρμογή και θα έπρεπε να κάνουμε πάλι ξανά την εγκατάσταση για το phpMyAdmin.

Στο τέλος πρέπει να πατήσουμε το κουμπί “Go” για να συνδεθούμε στο phpMyAdmin και θα μας φορτώσει την σελίδα που φαίνεται στην παρακάτω εικόνα [10]:



## Παράρτημα Δ

### **Flask**

Με τη χρήση του εργαλείου Flask θα διερευνήσουμε πως μπορούμε να αναπτύξουμε και να φιλοξενήσουμε μια ιστοσελίδα στο λειτουργικό σύστημα Raspberry Pi Os, η οποία θα δίνει πληροφορίες που θα συλλέγει από το Sense Hat.

Για να ξεκινήσουμε να δημιουργήσουμε μια “έξυπνη ιστοσελίδα” θα μας χρησιμεύσει ένα web-framework, το οποίο είναι ένα ειδικό λογισμικό, σχεδιασμένο να υποστηρίζει διαδικτυακές εφαρμογές. Συνήθως αυτά τα web-frameworks περιέχουν πολλές βιβλιοθήκες που δίνουν πολλές δυνατότητες όπως σύνδεση με βάσεις, διαχείριση χρηστών, έτοιμες φόρμες και άλλα. Ένα τέτοιο web-framework είναι το django, ενώ εμείς θα εξετάσουμε ένα ελαφρύτερο, με την ονομασία Flask το οποίο θα έχουμε ήδη εγκατεστημένο στην python3. Το Flask, αν και δεν διαθέτει πολλές “έτοιμες” λειτουργίες όπως το django, δεν σημαίνει πως δεν είναι πολύ δυνατό.

Επεκτείνεται εύκολα και δίνει μεγαλύτερη ελευθερία στους προγραμματιστές να κάνουν χρήση τα εργαλεία που προτιμούν. Έτσι με αυτόν τον τρόπο έχουμε μάθει πως μπορούμε να διαθέσουμε τα στοιχεία των αισθητήρων του Sense Hat στο τοπικό μας δίκτυο, αλλά και στο ίντερνετ. Επίσης θα είμαστε σε θέση να κάνουμε το πρώτο και πιο σπουδαίο βήμα για την ασφάλεια των δεδομένων, που δεν είναι άλλο από το να εμποδίσουμε την πρόσβαση σε τρίτους στη σελίδα μας με χρήση κωδικού. Αυτά τα είπαμε ήδη προηγουμένως στην περιγραφή της λειτουργίας και εκτέλεσης της εφαρμογής μας.