

# Πανεπιστήμιο Πατρών, ΗΜΤΥ

Εργασία του μαθήματος «Εισαγωγή στην επιστήμη του ΗΜΤΥ»

## Ομαδική έκθεση project ΟΕ-01

Δημοσθένης Καραμπάρπας ΑΜ : 1089862  
Νικόλαος Γιάννη ΑΜ : 1092569  
Δημήτριος Βυθούλκας ΑΜ : 1092758  
Ευάγγελος Δίκaros ΑΜ : 1092847  
Απόλλων-Αλέξανδρος Αναστόπουλος ΑΜ : 1092621  
Γεώργιος Κορμάς ΑΜ : 1092764  
Λάμπρος Αβούρης ΑΜ : 1092732

Θέμα Εργασίας :

Δημιουργία flask web application στην οποία ένας χρήστης πραγματοποιεί online test στην Python. Το αποτέλεσμα του αποθηκεύεται σε ιστορικό.

Οργάνωση :

Προτού ξεκινήσουμε τη συγγραφή του κώδικα συναντηθήκαμε όλα τα μέλη της ομάδας προκειμένου να χωρίσουμε αρμοδιότητες. Οι αρμοδιότητες χωρίστηκαν ως εξής:

- Στον Ευάγγελο Δίκaro ανατέθηκε η δημιουργία της διεπαφής του login.html και του register.html.
- Στον Δημοσθένη Καραμπάρπα ανατέθηκε η γραφική διεπαφή του main\_page.html.
- Στον Νικόλαο Γιάννη η δημιουργία του start.html, end.html και η σύνδεση αυτών με το server.py
- Στον Αλέξανδρο Αναστόπουλο ανατέθηκε η υλοποίηση της συνάρτησης loggedquestion(id) αλλά και των συναρτήσεων show\_question() και draw\_question() οι οποίες συσχετίζονται με την εμφάνιση των ερωτήσεων και των απαραίτητων στατιστικών που προκύπτουν σύμφωνα με το πέρας του ερωτηματολογίου , σε συνεργασία με τον Λάμπρο Αβούρη και τον Δημήτρη Βυθούλκα.

- Στον Δημήτριο Βυθούλκα η υλοποίηση των συναρτήσεων που σχετίζονται με την εμφάνιση των ερωτήσεων και υπολογισμό της βαθμολογίας, σε συνεργασία με τους : Λάμπρο, Αλέξανδρο, Γεώργιο
- Στον Γεώργιο Κορμά ανατέθηκε η διασύνδεση του `main_page.html` με το `server.py`.
- Στον Λάμπρο Αβούρη ανατέθηκε η κατασκευή του database και η διαχείριση του deployment της εφαρμογής στο Heroku.

Υλοποίηση:

Η εφαρμογή είναι διαχωρισμένη σε client side server side.

Η client side της εφαρμογής έχει υλοποιηθεί με html και javascript.

Η server side της εφαρμογής έχει υλοποιηθεί με python μέσω της βιβλιοθήκης flask για την δημιουργία του server και της postgresql για την δημιουργία του database.

Τα client side – server side κομμάτια της εφαρμογής διασυνδέονται μέσω gunicorn

Το hosting της εφαρμογής γίνεται μέσω του Heroku.

Μέθοδος επίλυσης :

Για την επίλυση του προβλήματος χρησιμοποιήθηκε η φιλοσοφία bottom up programming. Αρχικά ορίστηκαν τα μέρη του προγράμματος όπως περιεγράφηκε παραπάνω. Μετά σταδιακά άρχισαν να δημιουργούνται λειτουργίες αυξανόμενης πολυπλοκότητας ενώ τελικά συνδέθηκαν για την δημιουργία του τελικού αποτελέσματος.

Οδηγίες χρήσης/εγκατάστασης

Μπορεί κανείς να μπει στο site της εφαρμογής στο Heroku.

Εναλλακτικά, εφόσον ο χρήστης έχει python κατεβασμένη στον υπολογιστή, μπορεί να κατεβάσει τον πηγαίο κώδικα καθώς και τις απαραίτητες βιβλιοθήκες που αναγράφονται στην κορυφή του κάθε υποπρογράμματος με την εντολή `pip install {όνομα βιβλιοθήκης}`. Να τρέξει το πρόγραμμα `server.py` και να επισκεφτεί την θύρα `localhost:5000` στον browser για να δει την σελίδα και offline.

Παραδείγματα χρήσης/screenshots

Ο κώδικας :

Github : <https://github.com/lampros2003/Project-OE01>

To link του site:

<https://project-oe01.herokuapp.com>

Τα αποτελέσματα των ιστοσελίδων της εργασίας φαίνονται παρακάτω:

(όπου οι διπλές αγκύλες συμβολίζουν τις μεταβλητές Jinja και στο τελικό πρόγραμμα αντικαθίστανται από τα στοιχεία του κάθε παίχτη)

# Login

**Username**

**Password**

Continue

Cancel

[Don't have an account? Create account](#)

# Create Account

**Username**

**Password**

Continue

Cancel

[Already have an account? Sign in](#)

# Welcome {{name}}

In previous games...

most recent try: {{try1}}/5

2nd most recent try: {{try2}}/5

3rd most recent try: {{try3}}/5

4th most recent try: {{try4}}/5

Begin



# Player: {{name}}

You scored : {{score}}/5 Congratulations!

In previous games...

most recent try: {{try1}}/5

2nd most recent try: {{try2}}/5

3rd most recent try: {{try3}}/5

4th most recent try: {{try4}}/5



Try again

OR

Study a bit

## Python-Quiz - User: potatas

### Question 20

Τι θα δώσει ο παρακάτω κώδικας:

```
for i in range(3, 12, 4):  
    print(i+5, end=' ')  
else:  
    print('end')
```

Answer (choose one):

1. ☐ 3 7 11 end
2. ☐ 3 7 11
3. ☐ 8 12 16
4. ☐ 8 12 16 end
5. ☐ 3 12 4 end

Submit