

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΔΥΝΑΜΙΚΟΥ ΙΣΤΟΧΩΡΟΥ
ΕΞΕΤΑΣΗΣ
ΦΟΙΤΗΤΩΝ**

ΒΑΣΙΛΟΥΔΗΣ ΠΑΝΤΕΛΗΣ

ΕΠΙΒΛΕΠΩΝ: ΔΑΣΥΓΕΝΗΣ ΜΗΝΑΣ

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
<http://arch.icte.uowm.gr>

2/7/2015

Πλάνο παρουσίασης

2

1. Εισαγωγή

- Περιγράφουμε τον ιστοχώρο
- Αναφέρουμε τα πλεονεκτήματα

2. Τεχνολογίες που χρησιμοποιήθηκαν

- PHP - JavaScript
- MySQL - PDO

3. Ανάλυση και Σχεδίαση του ιστοχώρου

- Αναφέρουμε τους ρόλους του συστήματος
- Περιγράφουμε τις απαιτήσεις
- Παρουσιάζουμε την βάση δεδομένων

4. Υλοποίηση του ιστοχώρου

- Αναλύουμε τα βασικά σημεία του προγραμματισμού της σελίδας εξέτασης

5. Βασικές λειτουργίες του ιστοχώρου

6. Συμπεράσματα και μελλοντικές επεκτάσεις

Σκοπός του ιστοχώρου - πλεονεκτήματα


3

- Πρόκειται για ένα δυναμικό ιστοχώρο για την εξέταση φοιτητών ηλεκτρονικά
 - Υπάρχουν ερωτήσεις πολλαπλής επιλογής και ελεύθερου κειμένου
 - Η εξέταση είναι ασύγχρονη
 - Δίνουμε τη δυνατότητα ανώνυμης εξέτασης μέσω token
- Πλεονεκτήματα
 - Ο διδάσκοντας εφαρμόζει ευρύτερη κάλυψη της διδακτέας ύλης
 - Προσαρμόζει τις διαλέξεις σύμφωνα με τα ποσοστά επιτυχίας
 - Οι φοιτητές αφομοιώνουν αποτελεσματικότερα τη διδακτέα ύλη
 - Έτσι, προετοιμάζονται καλύτερα για τις τελικές εξετάσεις

Χρησιμότητα PHP - JavaScript

4

➤ PHP

- Την χρησιμοποιούμε για τον χειρισμό των λειτουργιών που διεκπεραιώνουν οι ιστοσελίδες / είσοδος-έξοδος δεδομένων  \$_POST/ \$_GET
- Εκτελείται στη πλευρά του διακομιστή
- Παρέχει υποστήριξη σε ευρεία γκάμα βάσεων δεδομένων

➤ JavaScript

- Είναι μια γλώσσα σεναρίων
- Την ενσωματώνουμε στον XHTML κώδικα
- Την χρησιμοποιούμε για να αλλάζουμε δυναμικά το περιεχόμενο των XHTML στοιχείων

Τι είναι η MySQL - PDO

5

➤ MySQL

- Σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS)
- Χρησιμοποιεί την SQL ως γλώσσα ερωτημάτων
- Απαιτεί λίγους πόρους και οι χρόνοι απόκρισης είναι πολύ μικροί

➤ PDO

- Παρέχει πρόσβαση σε βάσεις RDBMS
- Χρησιμοποιεί έτοιμες δηλώσεις SQL και δεσμευμένες παραμέτρους
- Προστατεύει από το SQL Injection 1^{ης} τάξης

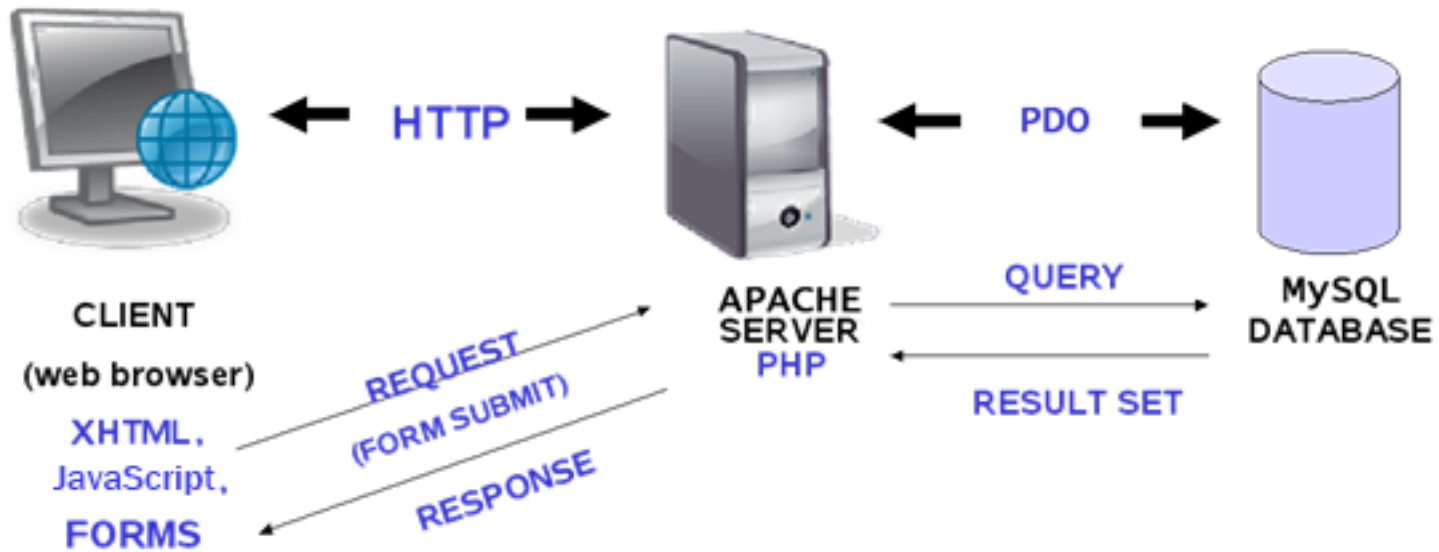
Παράδειγμα δήλωσης PDO

6

1. `$dbconn= new PDO("mysql:host=$dbhost;dbname=$dbbase", $dbuser, $dbpass);`
2. `$test = $_GET['tid'];`
3. `$current_datetime = date('Y-m-d H:i:s');`
4. `$query = $dbconn->prepare("SELECT * FROM tests WHERE id= :tid AND start_date<=:cdate");`
5. `$query->bindParam(':tid', $test);`
6. `$query->bindParam(':cdate', $current_datetime);`
7. `$query->execute();`
8. `$rows = $query->fetch(PDO::FETCH_ASSOC);`

Αρχιτεκτονική Ιστοχώρου

7



<http://i-exams.com/steste.php?tokid=134623792b616>

Ρόλοι του συστήματος

8

➤ Διαχειριστής

- Καθολικός: έχει πρόσβαση σε όλα τα αντικείμενα οποιουδήποτε καθηγητή και μπορεί να τα προσθέτει/ επεξεργαστεί
- Απλός (Καθηγητής): διαχειρίζεται λιγότερα αντικείμενα που αφορούν τα δικά του δεδομένα

➤ Φοιτητής

- Συνδεδεμένος με τα στοιχεία εγγραφής του: εξετάζεται με πλήρη προνόμια και έχει πρόσβαση σε όλους τους βαθμούς ταυτόχρονα
- Συνδεδεμένος με token: έχει επίσης όλα τα προνόμια, αλλά μπορεί να ελέγξει μόνο τον βαθμό της εξέτασης στην οποία αντιστοιχεί το token

➤ Ελεύθερος χρήστης

- Περιορισμένα δικαιώματα και έλεγχος βαθμού μόνο μια φορά στο τέλος

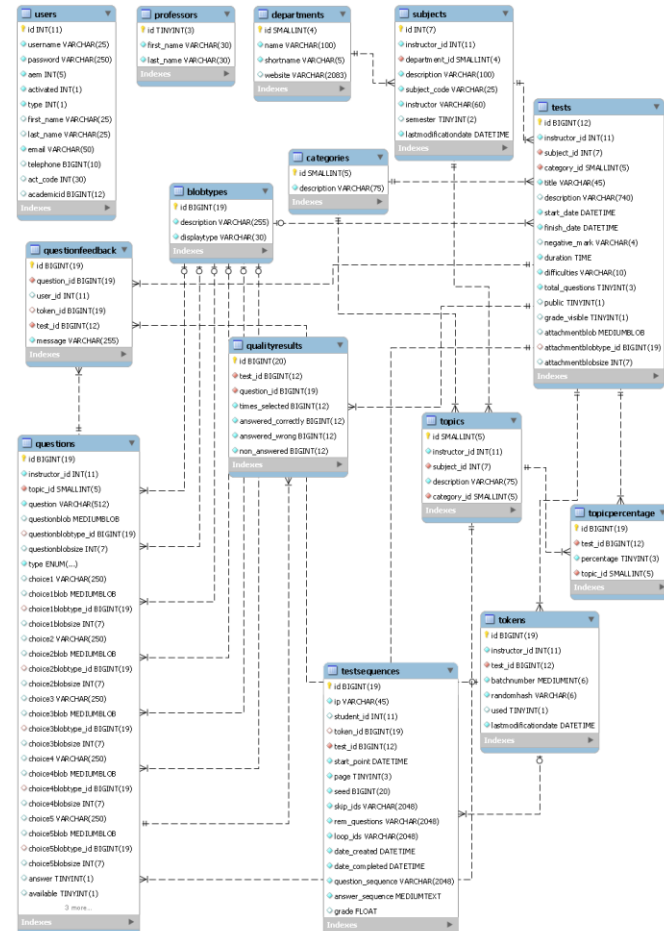
Απαιτήσεις του ιστότοπου

- Ο διαχειριστής - καθηγητής
 - Συνδέεται στο σύστημα
 - Προσθέτει/ επεξεργάζεται/ διαγράφει: (τμήματα), (καθηγητές), μαθήματα, (κατηγορίες), ενότητες, ερωτήσεις (+πολλαπλές μέσω CSV), διαγωνίσματα, tokens
 - Βλέπει/ εκτυπώνει αναλυτικά τις απαντήσεις και τους βαθμούς των φοιτητών
 - Τροποποιεί τον διαθέσιμο χρόνο υποβολής των ενεργών εξετάσεων
 - Ελέγχει γραφήματα που αφορούν τα αποτελέσματα
- Ο φοιτητής
 - Απαντάει/ παραλείπει μια ερώτηση πολλαπλής επιλογής/ ελεύθερου κειμένου ανά σελίδα με τυχαία σειρά εμφάνισης
 - Στέλνει μηνύματα ανά ερώτηση προς τον καθηγητή
 - Ελέγχει τις βαθμολογίες του - μέσο όρο - μέγιστη/ ελάχιστη βαθμολογία

Βάση δεδομένων του συστήματος

10

- Users
- Departments
- Professors
- Subjects
- Categories
- Topics
- Questions
- Topicpercentage
- Tests
- Blobtypes
- Tokens
- Testsequences
- Questionfeedback
- Qualityresults



Χαρακτηριστικά διαγωνίσματος

11

➤ Περιλαμβάνει ερωτήσεις που ανήκουν σε

- Ένα μάθημα
- Μια κατηγορία
- Ένα εύρος επιπέδων δυσκολίας
- Μια μίξη ενοτήτων



* Ποσοστά ενοτήτων(%):

Κεφάλαιο 1: Κεφάλαιο 3: Κεφάλαιο 4:

Συνολικό ποσοστό: 100

➤ Χαρακτηρίζεται από

- Ημερομηνία και ώρα έναρξης/ λήξης
- Διαθέσιμο χρόνο εξέτασης
- Αρνητική βαθμολογία (προαιρετική)
- Ορατότητα τελικού βαθμού στους χρήστες
- Διαθεσιμότητα στους ελεύθερους χρήστες
- Επισυναπτόμενο

Δυνατότητα αλλαγής
υπολογιστή και συνέχιση
της εξέτασης

Αρχικοποίηση εξέτασης

12



```
1. $n = 1;
2. $sql="";
3. $union="";
4. while (isset($_SESSION['topic'][$n]) && $_SESSION['topic'][$n] != "")
5. {
6.     $sql .= $union . "(SELECT * FROM questions WHERE available = 1 AND topic_id= " . $_SESSION['topic'][$n] . " AND
7.     difficulty IN (" . $_SESSION['difficulties'] . ") AND RAND() < (SELECT ((($_SESSION['res'][$n] / COUNT(*)) * 10) FROM
8.     questions) ORDER BY RAND(" . $_SESSION['seed'] . ") LIMIT " . $_SESSION['res'][$n] . ")";
9.     $union=" UNION ";
10.    $n++;
11. }
12. $sql2 = $sql . " ORDER BY RAND(" . $_SESSION['seed'] . ")";
13. //produce a list of the questions
14. $query02 = $dbconn->prepare($sql2);
15. $query02->execute();
16. $rows02 = $query02->fetchAll(PDO::FETCH_ASSOC);
```

➤ `$_SESSION['seed'] = rand();`
➤ `$_SESSION['res'][$n] = round(($_rows01['percentage'] / 100) * $_rows['total_questions']);`



Ανακάτεμα επιλογών ερώτησης

13

```
1. $choices = array
2. (
3.     0=>array ('correct' => 1, 'answer' => $question['choice1'], 'id' => 1, 'blob' => $question['choice1blob'], 'blobid' =>
        $question['choice1blobtype_id']),
4.     1=>array ('correct' => 0, 'answer' => $question['choice2'], 'id' => 2, 'blob' => $question['choice2blob'], 'blobid' =>
        $question['choice2blobtype_id']),
5.     2=>array ('correct' => 0, 'answer' => $question['choice3'], 'id' => 3, 'blob' => $question['choice3blob'], 'blobid' =>
        $question['choice3blobtype_id']),
6.     3=>array ('correct' => 0, 'answer' => $question['choice4'], 'id' => 4, 'blob' => $question['choice4blob'], 'blobid' =>
        $question['choice4blobtype_id']),
7.     4=>array ('correct' => 0, 'answer' => $question['choice5'], 'id' => 5, 'blob' => $question['choice5blob'], 'blobid' =>
        $question['choice5blobtype_id'])
8. );
9. shuffle($choices);
10. $_SESSION['choices'][$_SESSION['page']]=$choices;  $output  $j.(".$corr.")'.$id."> ".$choice
```

- `<button type="submit" name="add" id="add">Υποβολή</button>`
- `<button type="submit" name="skip" id="skip">Παράληψη</button>`
- `<button type="submit" name="end" id="end">Υποβολή τεστ</button>`

Επεξεργασία αποτελέσματος

14

```
1.     if (isset($_POST['end']))
2.     {
3.         session_write_close();
4.         $dbconn = null;
5.         header("location: stesee.php?tid=".$_SESSION['test']."&msg=1");
6.         exit();
7.     }
8.     if (isset($_POST['add']))
9.     {
10.        $_SESSION['answer'] .= $_POST['question_id'].>'. $_POST['choice']. '|';
11.    }
12.    if (isset($_POST['skip']))
13.    {
14.        $_SESSION['skip_ids'] .= $_POST['question_id']. ',';
15.    }
```

- ▣ `$_SESSION['rem_questions'] = str_replace($_POST['question_id'].",", "", $_SESSION['rem_questions']);`
- ▣ `$_SESSION['page'] = $_SESSION['page'] + 1;`
- ▣ `$_SESSION['total_question_loop'] = $_SESSION['total_questions'];`


Αρχικός και δευτερεύον βρόγχος ερωτήσεων

15

```
1. if ($_SESSION['page'] <
   $_SESSION['total_question_loop'])
2. {
3.     $_SESSION['page'] = $_SESSION['page'] + 1 ;
4. }
```

- ▣ \$remq_ids = rtrim(\$_SESSION['rem_questions'], ",");
- ▣ \$sql = "(SELECT * FROM questions WHERE id IN ('.\$remq_ids.') LIMIT 0,1)";
- ▣ \$query2 = \$dbconn->prepare(\$sql);
- ▣ \$query2->execute();
- ▣ \$rows2 = \$query2->fetchAll(PDO::FETCH_ASSOC);

```
$SESSION['total_question_loop'] =
count($loop_ids);
```



```
1. if ($_SESSION['page'] == $_SESSION['total_question_loop'])
2. {
3.     if (isset($_SESSION['skip_ids']) &&
   $_SESSION['skip_ids'] != "")
4.     {
5.         $_SESSION['loop_ids'] =
   $_SESSION['skip_ids'];
6.         $_SESSION['skip_ids'] = "";
7.         $_SESSION['page'] = 1;
8.     }
9. }
```

- ▣ \$loop_ids = rtrim(\$_SESSION['loop_ids'], ",");
- ▣ \$sql = "(SELECT * FROM questions WHERE id IN ('.\$loop_ids.') ORDER BY RAND('.\$SESSION['seed'].') LIMIT ".\$start_from.",1)";
- ▣ \$query2 = \$dbconn->prepare(\$sql);
- ▣ \$query2->execute();
- ▣ \$rows2 = \$query2->fetchAll(PDO::FETCH_ASSOC);

Παράδειγμα ακολουθίας ερωταπαντήσεων

16

page	rem_questions	skip_ids	loop_ids	answer_sequence
1	<u>16</u> ,17,14,13,23,			16>2(3)4
2	<u>17</u> ,14,13,23,	17,		16>2(3)4
3	<u>14</u> ,13,23,			16>2(3)4 14>Hello
4	<u>13</u> ,23,	17,13		16>2(3)4 14>Hello
5	<u>23</u> ,		17,13	16>2(3)4 14>Hello 23>Hi
1			17, <u>13</u>	16>2(3)4 14>Hello 23>Hi 13>2(2)1
2			<u>17</u> ,13	16>2(3)4 14>Hello 23>Hi 13>2(2)1 17>3(5)3

\$_SESSION['mark'] = 100 / \$rows['total_questions'];
 \$_SESSION['negative_mark'] = π.χ. 0.8 %



20 %

grade

-0.8 + 20 - 0.8 = 18.4 % (αν οι ελ. κειμένου είναι σωστές +20 + 20 = 58.4 %)

Δημιουργία χρονομέτρου εξέτασης

17

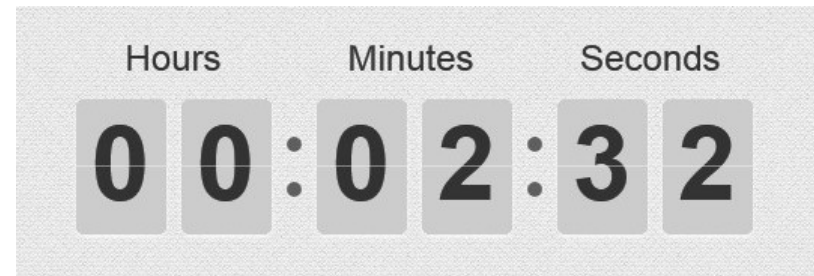
```
1. <script type="text/javascript">
2.     $(document).ready(function(){
3.         setInterval(function(){
4.             $.ajax({
5.                 type:"GET",
6.                 url:"get_time.php",
7.                 data:{action:"showresults"},
8.                 dataType: 'json',
9.                 success:function(data){
10.                    $('#remHours').val(data.remHours);
11.                    $('#remMinutes').val(data.remMinutes);
12.                    $('#remSeconds').val(data.remSeconds);
13.                }
14.            });
15.        }, 100);
16.    });
17. </script>
```

➤ Χρόνος που απομένει = χρόνος λήξης – τρέχον
χρόνος



start_point + \$_SESSION['duration']

```
echo json_encode(array('remHours' =>
    $remainingHours,'remMinutes' =>
    $remainingMinutes,'remSeconds' =>
    $remainingSeconds));
```



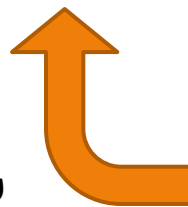
Προβολή/ βαθμολόγηση απαντήσεων

18

➤ Έλεγχος απαντήσεων

Όνοματεπ... φοιτητή	AEM	IP χρήστη	Σειρά έκδοσης	Αρ. Token	Σωστές	Λάθος	Αναπάντη...	Αβαθμολό...	Βαθμός %	Μεταφορά σε PDF
Ξενοδόκης Ηλίας	118	127.0.0.1			3	0	0	2	60	
Αρ. ερώτησης		ID ερώτησης			Αποτέλεσμα					
1		16			✓					
2		17			✓					
3		14			Αβαθμολόγητη					
4		13			✓					
5		23			Αβαθμολόγητη					

➤ Βαθμολόγηση ελ. κειμένου



Βαθμολόγηση απαντήσεων

Τι ονομάζουμε αντικείμενα στον αντικειμενοστραφή προγραμματισμό?

Hello

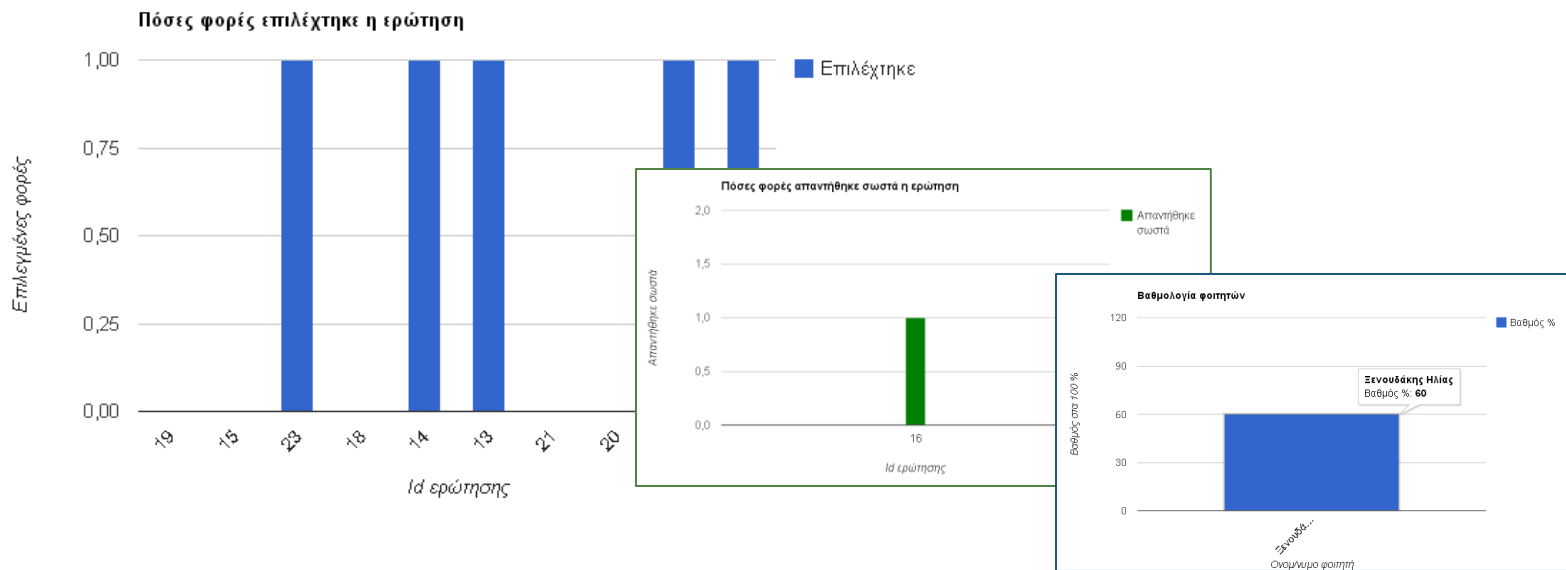
Σωστή: Λάθος:

Βαθμολόγηση

Γραφήματα και εκτύπωση token

19

- Γραφήματα διαγωνίσματος/ ερωτήσεων/ βαθμολογιών ανά τεστ και ανά φοιτητή



- Εκτύπωση token



Διδάσκων: Παντελής Βασιλούδης
Μάθημα: Αντικειμενοστραφής Προγραμματισμός
Κατηγορία: Εργαστήριο 1
Διαγώνισμα: τεστ 1
Ημ/νία λήξης: 2015-06-10 15:40:00
localhost/i-exams/steste.php?tokid=13462375bed26

Συμπεράσματα και μελλοντικές επεκτάσεις

20

- Συμπεράσματα
 - Καινοτόμο σύστημα σε σχέση με τα ήδη υπάρχοντα
 - Φιλικό περιβάλλον και ασφαλές σε ικανοποιητικό βαθμό
 - Γρήγοροι χρόνοι ανταπόκρισης
- Μετρικές
 - Δημιουργήσαμε 143 αρχεία.
 - Γράψαμε 23,755 γραμμές κώδικα (με σχόλια)
 - Χρησιμοποιήσαμε 26 συναρτήσεις JavaScript
- Μελλοντικές επεκτάσεις
 - Υποστήριξη πολλαπλών σωστών απαντήσεων
 - Εισαγωγή κώδικα φοιτητή και ξεχωριστή βαθμολόγηση
 - Δημιουργία εφαρμογής για Android

Τέλος

21

- Ακολουθούν
 - Επίδειξη
 - Ερωτήσεις



EVERYTHING THAT HAS A BEGINNING HAS AN END