

## **Εργαστηριακή Άσκηση 2**

### **Υλοποίηση Ερωτημάτων.**

#### **Ερώτημα 1**

Για την υλοποίηση του πρώτου ερωτήματος χρησιμοποιήθηκαν τα αρχεία slider.js , slider.css.Επίσης έγιναν τροποποιήσεις στον ελεκτη users controller και στο αρχείο html της αρχικής σελίδας για να προσαρμόσουμε τη λύση στην πρώτη ασκήση.Ο κώδικας που υλοποιεί την προβολή παρουσίασης είναι δική μας κατασκευής και λειτουργεί ως εξής:

Στην αρχική σελίδα εμφανίζονται : η πιο πρόσφατη φωτογραφία του χρήστη καθώς και η πιο πρόσφατη φωτογραφία κάθε άλλου χρήστη που ο χρήστης ακολουθεί.Περνοντας το ποντικί πάνω από μια φωτογραφία εμφανίζεται το παράθυρο της προβολής.Επειτα γίνεται η κλήση στον photos controller με παραμέτρους το id του χρήστη που έχει ανεβάσει τη φωτογραφία και μια μεταβλητή index.Ο controller με βάση το index θα επιστρέψει το πολυ τις 10 επόμενες φωτογραφίες Στο σημείο αυτό μέχρι να ολοκληρωθεί η κλήση εμφανίζουμε ένα loader στο παράθυρο προβολής και αφαιρούμε κατά την επιτυχία κλησης.Τελος γίνεται προσθήκη των φωτογραφιών στο παράθυρο.

Ο χρήστης έχει την επιλογή να μετακινήσει τις φωτογραφίες αριστερά η δεξιά η να ενεργοποιήσει το live show οπου οι φωτογραφίες εναλλάσσονται ανά 2 δευτερόλεπτα μέχρι και την τελευταία από οπου γίνεται επιστροφή στην αρχή και το live show τερματίζει.Οσο ο χρηστης προχωράει προς τα δεξια γίνονται κλήσεις για να φέρουμε τις επόμενες φωτογραφίες στον client. Οι φωτογραφίες εμφανίζονται από την πιο πρόσφατη στην πιο παλιά καθώς επίσης και ο τίτλος.

Οι φωτογραφίες εναλλάσσονται με την εξής λογική:

Κάθε φορά οι νέες φωτογραφίες προστίθενται δεξιά από την τελευτα.Αν το παραθυρο προβολής έχει θέση ίση με 0% τότε η δεξιά φωτογραφία είναι σε απόσταση 110% . Η ανάθεση των τοποθεσιών έγινε με τον κανονα transform = "translateX( position% )"; Για παράδειγμα ένα στιγμιότυπο της λειτουργιάς είναι το εξής: -110% , 0% , 110 % . Ο χρήστης βλέπει την δεύτερη φωτογραφία και αν κινιδει προς τα δεξιά 8α γίνει η εναλλαγή : -220% , -110% , 0% . Τέλος αν ξαναπατήσει δεξιά η προβολή θα επανέλθει στην αρχική κατάσταση 0% , 110% , 220%.Το παράθυρο της προβολής έχει overflow: hidden για να μην υπάρχει υπερχείλιση . Επίσης προσθέσαμε animations κατά την εναλλαγή για την καλύτερη οπτική αναπαράσταση.

Αν ο χρήστης απομακρύνει τον δείκτη του ποντικιού έξω από την φωτογραφία, η προβολή παρουσίασης σταματά και οι φωτογραφίες διαγραφονται.

## **Ερώτημα 2**

Κάνοντας (μονο)click πάνω στη φωτογραφία εμφανίζεται το comment pop up για τη αντίστοιχη φωτογραφία όπου ο χρήστης μπορεί να δει όλα τα σχόλια της φωτογραφίας ή να κάνει σχόλια . Τα σχολια φορτώνονται απο τον server ο οποίος επιστρέφει ένα πινάκα από αντικείμενα που περιέχουν πληροφορία για κάθε σχόλιο. Όταν ο χρήστης γράψει σχόλιο γίνεται αποθήκευση του σχόλιου με ασύγχρονη κλήση στον controller και επίσης δημιουργούμε ένα αντικείμενο (html element) σχόλιου με βάση το response της κλήσης και το προσθέτουμε στο pop up.

Αν ο χρήστης κάνει click στην περίπτωση που το live show είναι ενεργό τότε θα γίνει διακοπή και θα συνεχίσει όταν ο χρήστης κλείσει το pop up. Το styling του comment pop up ορίζεται στο slider.css.

Τελος μετα το κλεισιμο του pop up τα σχολια διαγραφονται απο το DOM.

## **Ερώτημα 3**

Ο εντοπισμός του διπλού click γίνεται ως εξής:

Αρχικά εντοπίζουμε το πρώτο click και ορίζουμε ένα time out το οποίο θα εκτελεστεί μέσα σε ένα πολύ μικρό χρόνο σε περίπτωση που δεν εντοπιστεί δεύτερο click. Όταν το δεύτερο click καταφθάσει μέσα στο time out τότε το ακυρώνουμε και εκτελούμε την λειτουργιά του δεύτερο click. Αν όμως περάσει το time out τότε το εκλαβανουμε ως μονό click.

Επειτα από το DOM περνούμε το id της φωτογραφίας και καλούμε τον controller ασύγχρονα για την διαγραφή από την βάση. Στη συνέχεια διαγραφούμε την φωτογραφία και από την παρουσίαση και κάνουμε ανατοποθέτηση των φωτογραφιών τροποποιώντας κατάλληλα της θέσεις με τον κήνονα transform = "translateX( position% )";

Εδώ στην περίπτωση που ο χρήστης διαγραφεί συνεχεία της φωτογραφίες χωρίς να έχουν ληφθεί όλες από τον server η παρουσίαση θα αδειάσει παραλού που υπάρχουν και άλλες φωτογραφίες που ανήκουν στον χρήστη.

Τέλος ο χρήστης μπορεί να διαγράψει μονό δικές του φωτογραφίες. Αυτό έγινε με την πρόσθεση:

```
- if @user.id == photo.user_id  
  .js-belongs-to-user  
- else  
  .js-belongs-to-followed
```

σε κάθε παραθυρο προβολής κατά την δημιουργία της σελίδας.

Όλες οι ασύγχρονες κλήσεις έγιναν με τη χρήση του fetch API και ο κώδικας για τα ερωτήματα βρίσκεται στα αρχεία slider.js , slider.css . Επειδή είχαμε κάποιο πρόβλημα με τα events στην jquery στην πρώτη άσκηση αποφασίσαμε να χρησιμοποιήσουμε vanilla js.

Τέλος για τον καλύτερο έλεγχο της άσκησης σας στέλνουμε όλο το project.