



Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας

Internet and Cloud Application Development

Καθηγητής: κ. Π. Λουρίδας

Εργασία

Ψηφιακό κοινωνικό δίκτυο προκλήσεων

Μέλη Ομάδας

Κοπανίδης Κωνσταντίνος 3130098

Παπαθεοδώρου Δημήτριος 3130162

2015 - 2016

Χειμερινό Εξάμηνο

Εισαγωγή	1
Περιγραφή - ΦΑΣΗ 1	2
Ιδέα:	2
Τυπικοί χρήστες:	2
Τεχνικοί και Περιβαλλοντικοί παράμετροι:	2
Πλάνο ανάπτυξης:	2
Χρονοδιάγραμμα:	2
Προσωπικό:	2
Προϋπολογισμός:	3
Περιγραφή - ΦΑΣΗ 2	3
Ανάλυση εργασιών	3
Δημιουργία Προφίλ:	3
Δημιουργία Challenge:	3
Throw Gauntlet:	3
Feed:	3
Follow:	3
Like and Comment:	3
Περαίωση Πρόκλησης:	3
Επίλυση τεχνικών προβλημάτων/περιορισμών:	4
Τεχνική Ανάλυση	4
Εγκατάσταση Εφαρμογής	4
Απλή εκκίνηση:	4
Nginx & Docker:	5
Automated Build	6

Εισαγωγή

Για την ανάπτυξη της εφαρμογής μας SecondEyes, χρησιμοποιήσαμε το **μοντέλο ανάπτυξης LUCID**, γιατί είναι ένα αυστηρά ορισμένο μοντέλο ανάπτυξης με σαφώς ορισμένα στάδια, αλλά παράλληλα προσφέρει πολλές δυνατότητες ευελιξίας που πιστεύουμε ότι θα μας καθοδηγήσει στην διεκπεραίωση της εργασίας μας και ταυτόχρονα συνάδει με τον τρόπο σκέψης μας.

Περιγραφή - ΦΑΣΗ 1

Ιδέα:

- Η ιδέα κυμαινόταν γύρω απ' την ανάπτυξη ενός διαφορετικού κοινωνικού δικτύου που αντί να καθηλώνει τους ανθρώπους στην καρέκλα τους, θα τους έθετε σε συνεχή εγρήγορση Έτσι, μέσω μιας απλής αναγωγής σε παιχνίδι, *gamification*, σκεφτήκαμε τη δημιουργία ενός κοινωνικού δικτύου προκλήσεων, όπου κάθε παίκτης θα μπορεί να προκαλέσει έναν άλλον σε μια δοκιμασία με ψηφιακά έπαθλα και φυσική ικανοποίηση.

Τυπικοί χρήστες:

- Οποιαδήποτε άτομα έχουν όρεξη για ζωή, παιχνίδι και προκλήσεις. Δηλαδή κυρίως απευθύνεται σε άτομα νεαρής ηλικίας, χωρίς αυτό να είναι αυστηρό.

Τεχνικοί και Περιβαλλοντικοί παράμετροι:

- Πρόκειται ένα cloud-based web application. Χρησιμοποιήθηκε το PHPStorm IDE ως περιβάλλον ανάπτυξης της εφαρμογής, καθώς και το GitHub ως version control system. Η ανάπτυξη έγινε με χρήση του MEAN Stack, δηλαδή *MongoDB* ως βάση, *NodeJS* ως JavaScript Runtime Environment για τον server, *Express* ως application framework για τη Node και *AngularJS* ως front-end application framework. Επίσης, χρησιμοποιήθηκε το UI component της Angular Material για την αισθητική της σελίδας.
- Το deployment γίνεται με χρήση NGINX για web server και *docker* για το περιβάλλον που εκτελεί τα containers της εφαρμογής
- Επίσης, χρησιμοποιήθηκε *Ansible* για το automation και *mocha* και *chai* για το testing του server.

Πλάνο ανάπτυξης:

- Θα χωρίσουμε την ανάπτυξη σε 3 στάδια:
 - *1ο Στάδιο*: Βασικός κώδικας σε back-end, front-end και σχεδίαση βάσης δεδομένων. Δυνατότητες register και log in.
 - *2ο Στάδιο*: Υλοποίηση επί μέρους λειτουργιών σύμφωνα με τις λειτουργικές απαιτήσεις της εφαρμογής, δηλαδή δημιουργία προκλήσεων, follow χρηστών, like, comment, search, throw gauntlet, κλπ
 - *3ο Στάδιο*: Έλεγχος λειτουργικότητας της εφαρμογής, αποσφαλμάτωση, unit testing, polishing της εφαρμογής και επέκταση/βελτίωση με βάση απαιτήσεις και νέες ιδέες.

Χρονοδιάγραμμα:

- Το χρονοδιάγραμμά μας δεν ήταν σταθερό λόγω του αυξημένου φόρτου εργασίας την περίοδο της ανάπτυξης. Μια κοινή συνιστώσα ενός υποτυπώδους χρονοδιαγράμματος ήταν η εβδομαδιαία συνάντησή μας στη σχολή τις Τετάρτες και τις Παρασκευές και η ενασχόλησή μας τις ενδιάμεσες μέρες.

Προσωπικό:

- Μέλη της ομάδας ανάπτυξης οι: Κοπανίδης Κωνσταντίνος και ο Παπαθεοδώρου Δημήτρης.

Προϋπολογισμός:

- Η “πληρωμή” μας θα είναι η ευχαρίστηση των χρηστών :-)

Περιγραφή - ΦΑΣΗ 2

- Ανάλυση εργασιών
 - Η εφαρμογή έχει τις εξής δυνατότητες:

Δημιουργία Προφίλ:

Δημιουργία ενός απλού προφίλ με μερικά προσωπικά στοιχεία για να μπορείς να συμμετέχεις στις προκλήσεις.

Δημιουργία Challenge:

Κάθε εγγεγραμμένος χρήστης μπορεί να δημιουργήσει μια πρόκληση που αποτελείται από τίτλο, περιγραφή, ημερομηνία περάτωσης και κατηγορία.

Throw Gauntlet:

Όπως οι πολεμιστές στο παρελθόν πετούσαν το “γάντι” τους για να προκαλέσουν κάποιον, έτσι κι εδώ κάποιος μπορεί να πετάξει το ψηφιακό του γάντι σε κάποιον άλλον με βάση ένα Challenge. Έτσι υλοποιείται η πρόκληση ενός παίχτη.

Feed:

Κάθε challenge και gauntlet αποτελεί ένα post στο feed στο οποίο μπορείς να σκρολάρεις.

Follow:

Κάθε χρήστης έχει τη δυνατότητα να ακολουθήσει κάποιον άλλον όπως συμβαίνει σε περισσότερα κοινωνικά δίκτυα. Μόνο στους followers σου μπορείς να ρίξεις gauntlets. Μελλοντικά, μπορεί τα challenges όσων ακολουθείς να φαίνονται πιο πάνω στο feed.

Like and Comment:

Σε κάθε post μπορείς να κάνεις like ή να αφήσεις κάποιο σχόλιο.

Περαίωση Πρόκλησης:

Για να ολοκληρωθεί μια πρόκληση χρειάζονται μερικά βήματα και από τον challenger και τον challengee. Αρχικά ο challengee, πηγαίνει στο προφίλ του, στο tab των gauntlets, και πατάει complete στο σχετικό gauntlet. Εκεί θα χρειαστεί να ανεβάσει μια φωτογραφία, που θα αποδεικνύει την ολοκλήρωση του gauntlet. Έπειτα το gauntlet θα έχει κατάσταση Review, με σκοπό να εξετάσει ο challenger τα αποδεικτικά στοιχεία(φωτογραφία) και να αποφασίσει αν θα δεχτεί ή όχι την περαίωση της πρόκλησης. Για να ελέγξει την φωτογραφία, αρκεί στο βασικό feed να πάει στο tab judge και να πατήσει review. Έπειτα βλέποντας την φωτογραφία έχει 2 επιλογές, accept και reject. Όπου μετά το πάτημα ενός εκ των δύο ο challengee ενημερώνεται για την απόφαση του challenger και 100 δηνάρια προστίθενται ή αφαιρούνται από το σύνολό του ανάλογα με την επιλογή.

Τεχνική Ανάλυση

Η τεχνική οργάνωση της εφαρμογής είναι αρκετά απλή και ακολουθεί το file structure των εφαρμογών MEAN.

Τα routes της εφαρμογής αναλαμβάνουν την διαχείριση των requests των χρηστών, και περιέχουν το μεγαλύτερο κομμάτι της λογικής της εφαρμογής. Αν το codebase ήταν μεγαλύτερο θα “σπάγαμε” το κάθε route σε επιμέρους modules για καλύτερη διαχείριση και συντήρηση του κώδικα. Ο φάκελος logic περιέχει functions που βοηθούν στην συνολική λειτουργία της πλατφόρμας, και ορισμένα σημεία αποτελούν κομμάτι και κάποιων διαδικασιών των routes. Στα models βρίσκονται όλα τα αντικείμενα που η εφαρμογή αποθηκεύει στην βάση, και κάποιοι έλεγχοι εγκυρότητας σε κάποια από αυτά. Στον φάκελο db γίνεται το αρχικό connection με την βάση, όπου τα connection details λαμβάνονται από τα enviroment variables.

Το authentication και session management της εφαρμογής, γίνεται με την χρήση του πρωτοκόλλου OAuth2. Ο λόγος για την χρήση αυτή είναι επειδή όντας κοινωνικό δίκτυο και σε μια πιθανή του επιτυχία, θα πρέπει ο χρήστης να μπορεί να χρησιμοποιήσει τον λογαριασμό του για login σε άλλες πλατφόρμες/εφαρμογές. Επίσης αυτό ανοίγει και την δυνατότητα σε πιθανά integrations με άλλες εφαρμογές

Το deployment γίνεται με την χρήση του σχετικού dockefile. Ο λόγος επιλογής docker είναι γιατί θέλαμε η εφαρμογή να μπορεί εύκολα και γρήγορα να εγκατασταθεί σε οποιοδήποτε μηχάνημα, και να έχει πλήρες isolation από το filesystem το οποίο δεν χρησιμοποιεί.

Επίσης, μιας και η εφαρμογή είναι stateless τα containers αποτελούν ιδανικό solution παρέχοντάς μας πολλά benefits στην συνολική αρχιτεκτονική και κανένα drawback στην μέχρι στιγμής ανάπτυξη.

Εγκατάσταση Εφαρμογής

Η εφαρμογή μπορεί να εγκατασταθεί ακολουθώντας τα παρακάτω σαφώς ορισμένα βήματα. Θα παραθέσουμε 3 διαδικασίες για την εγκατάσταση. Η πρώτη αφορά την εκτέλεση της εφαρμογής χωρίς την χρήση nginx ή docker, η δεύτερη την χειροκίνητη εγκατάσταση αλλά με τις προηγούμενες τεχνολογίες, και η τρίτη με τον αυτοματοποιημένο τρόπο με την χρήση του εργαλείου ansible.

Απλή εκκίνηση:

Βήμα 1: Κάνουμε εγκατάσταση του nodejs έκδοση 6.1

```
sudo apt-get install python-software-properties
```

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```

```
apt-get update & apt-get install nodejs
```

Βήμα 2: Εκτελούμε git clone για να κατεβάσουμε το project
git clone <https://github.com/kkopanidis/Desafio.git>

Τώρα πρέπει στον φάκελο Desafio να έχουμε το Project

Βήμα 3: Μπαίνουμε στον φάκελο Desafio και στο τερματικό εκτελούμε το εξής

```
npm install
```

Το οποίο θα εγκαταστήσει τα απαραίτητα πακέτα για την εκτέλεση της εφαρμογής

Επεξεργαζόμαστε το αρχείο mongoose του φακέλου db, ώστε το link να δείχνει στην σωστή διεύθυνση της mongo βάσης.

Βήμα 4: Τέλος εκτελούμε “npm start” και η εφαρμογή πλέον τρέχει στην θύρα 3000 του υπολογιστή, και μπορούμε να την δούμε πηγαίνοντας μέσω browser στην διεύθυνση “<http://localhost:3000>”

Για να μπορεί να λειτουργήσει σωστά, την πρώτη φορά που θα τρέξει, πηγαίνουμε στην διεύθυνση “<http://localhost:3000/init>” ώστε να κάνει η εφαρμογή μια αρχικοποίηση.

Από εκεί και πέρα είναι πλέον έτοιμη

Nginx & Docker:

Για να εγκαταστήσουμε το προηγμένο deployment της εφαρμογής κάνουμε διαφορετικά βήματα.

Αρχικά τρέχουμε το βήμα 2 της προηγούμενης διαδικασίας για να κατεβάσουμε τον κώδικα της εφαρμογής.

Βήμα 1: Εγκαθιστούμε το nginx, τρέχοντας τις ακόλουθες εντολές

```
sudo apt-get update
sudo apt-get install nginx
sudo systemctl enable nginx
```

```
sudo vi /etc/nginx/sites-enabled/desafio
```

Στην τελευταία εντολή, εντός του vi βάζουμε το κείμενο που υπάρχει εντός του φακέλου ansible/files με όνομα nginx.j2

Βήμα 2: Εγκαθιστούμε το docker engine τρέχοντας τις ακόλουθες εντολές

```
sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
sudo add-apt-repository deb https://apt.dockerproject.org/repo ubuntu-xenial main
```

```
sudo apt-get update
```

```
sudo apt-get install docker-engine
```

```
sudo systemctl enable docker
```

```
sudo reboot now
```

Βήμα 3: Δημιουργούμε ένα container για την mongoDB βάση, εκτελώντας τα ακόλουθα

```
docker run --name mongo -v /data:/data/db --restart=always -d mongo:latest
```

Βήμα 4: Μπαίνουμε στον φάκελο Desafio και εντός εκτελούμε τα εξής:

```
docker build -t desafio:latest .
```

```
docker run --name desafio -d -p 3000:3000 --link mongo:mongo --restart=always  
desafio:latest
```

Βήμα 5: Have fun

Automated Build

Το αυτόματο deployment έχει πιο απλά βήματα για να καταλήξουμε σε μια λειτουργική εφαρμογή.

Βήμα 1: Αρχικά δημιουργούμε στο μηχανήμα μας ένα private key, και αποθηκεύουμε το Public στο μηχανήμα που θέλουμε να γίνει το deployment. Όστε το ansible να μπορεί να συνδεθεί χωρίς κωδικό.

Βήμα 2: Όντας εντός του φακέλου ansible του project directory εκτελούμε

```
ansible-playbook -i hosts provision.yml
```

ΠΡΙΝ την εκτέλεση ελέγχουμε το αρχείο hosts και επαληθεύουμε ότι τα στοιχεία είναι σωστά

Βήμα 3: Για να κάνουμε τώρα deploy την εφαρμογή εκτελούμε

```
ansible-playbook -i hosts deploy.yml
```

Βήμα 4: Πληκτρολογούμε στον browser την ip του μηχανήματος και πρέπει να βρεθούμε στην αρχική οθόνη της εφαρμογής

ΠΡΟΣΟΧΗ: Πρέπει και στις 2 παραπάνω διαδικασίες, όπως και στην πρώτη να πάμε στο link /init για να αρχικοποιήσουμε την εφαρμογή.