

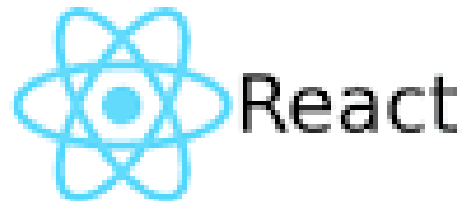
Η ΕΠΙΔΡΑΣΗ ΤΩΝ ΔΙΑΦΟΡΕΤΙΚΩΝ FRONTEND ΤΕΧΝΟΛΟΓΙΩΝ ΣΤΗΝ ΠΟΙΟΤΗΤΑ ΤΟΥ ΚΩΔΙΚΑ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ

Φοιτητής: Αθανάσιος Σίμος

Επιβλέπων Καθηγητής: Θεόδωρος Κασκάλης

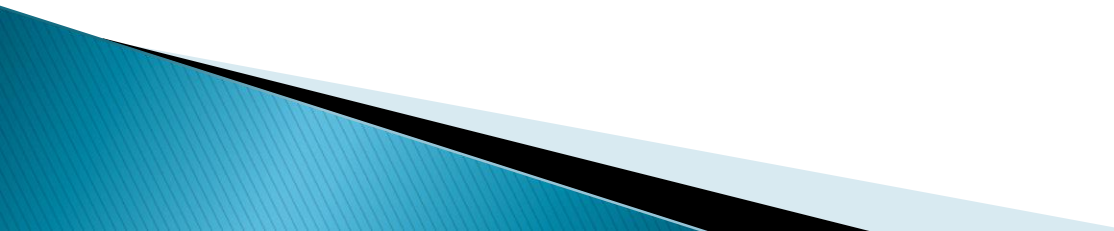
Πρόβλημα

- ▶ Αίσθηση mobile application.
- ▶ Ποιό framework/library να διαλέξω;



Angular vs React vs Vue

Οι προτάσεις του Eric Wohlgethan

- ▶ Typescript -> Angular
 - ▶ Καθοδηγητική δομή κώδικα-> Angular
 - ▶ Αντικειμενοστραφές περιβάλλον ανάπτυξης -> Angular
 - ▶ Ευελιξία -> Vue / React
 - ▶ Εφαρμογή μεγάλου μεγέθους -> All
 - ▶ New user friendly -> Vue
 - ▶ Χρήση καινοτόμων και τελευταίων τεχνολογιών-> Vue
 - ▶ Λογική και στυλ στο ίδιο αρχείο -> Vue
 - ▶ Θέλει να δουλέψει με HTML -> Angular/Vue
 - ▶ Αυστηρή χρήση JavaScript -> React
- 

Ποιότητα Κώδικα

- ▶ Κύριο χαρακτηριστικό για τον προγραμματιστή όσων αφορά την ανάπτυξη και τη συντήρηση ενός προϊόν λογισμικού.
- ▶ Αντικείμενο μελέτης όσων αφορά τη μέτρηση και τη βελτίωσή της.

Τεχνικό Χρέος Ορολογία

Το τεχνικό χρέος είναι μια συλλογή σχεδιασμών ή κατασκευών που μπορεί να λειτουργούν βραχυπρόθεσμα, παρουσιάζουν όμως ένα τεχνικό πλαίσιο που μπορεί να καταστήσει τις μελλοντικές αλλαγές πιο ακριβές στο κόστος ή ακόμη και αδύνατες.

Dagstuhl April 2016



Τεχνικό Χρέος

- ▶ Έλλειψη συμμόρφωσης με τα πρότυπα σχεδίασης
- ▶ Έλλειψη προτύπων σχεδίασης
- ▶ Υψηλές τιμές στις μετρικές κώδικα
- ▶ Κακές οσμές στον κώδικα
- ▶ Έλλειψη εγχειριδίου για τη χρήση του λογισμικού
- ▶ Copy paste

Τεχνικό Χρέος

- ▶ Μετριέται σε ανθρώπινο χρόνο
- ▶ Υπολογίζεται με το πόσοι κανόνες σχεδίασης παραβιάστηκαν και το χρόνο που χρειάζεται για την διόρθωσή τους.

Σκοπός

- ▶ Διαφορετική επιρροή κάθε τεχνολογίας στην ποιότητα κώδικα και στο τεχνικό χρέος.
- ▶ Ποια τεχνολογία παρουσιάζει το χαμηλότερο τεχνικό χρέος.
- ▶ Αναδιάρθρωση των προτάσεων του E. Wohlgethan ανάλογα με τα ευρήματα.

Μεθοδολογία

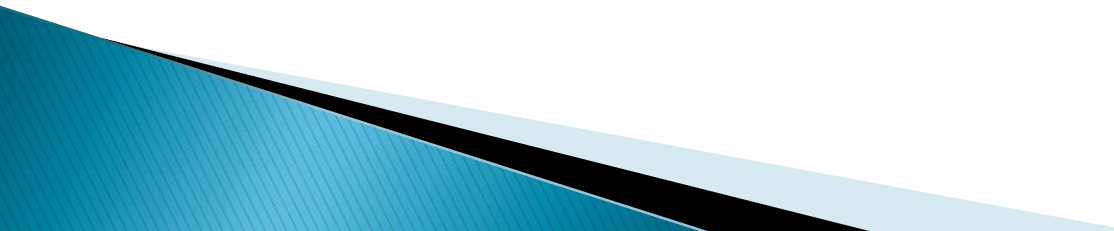
- ▶ Υλοποίηση εφαρμογής
- ▶ Τεχνολογίες υλοποίησης React Vue Angular
- ▶ Number of sprints 5
- ▶ Same milestones σε κάθε υλοποίηση

Εργαλεία

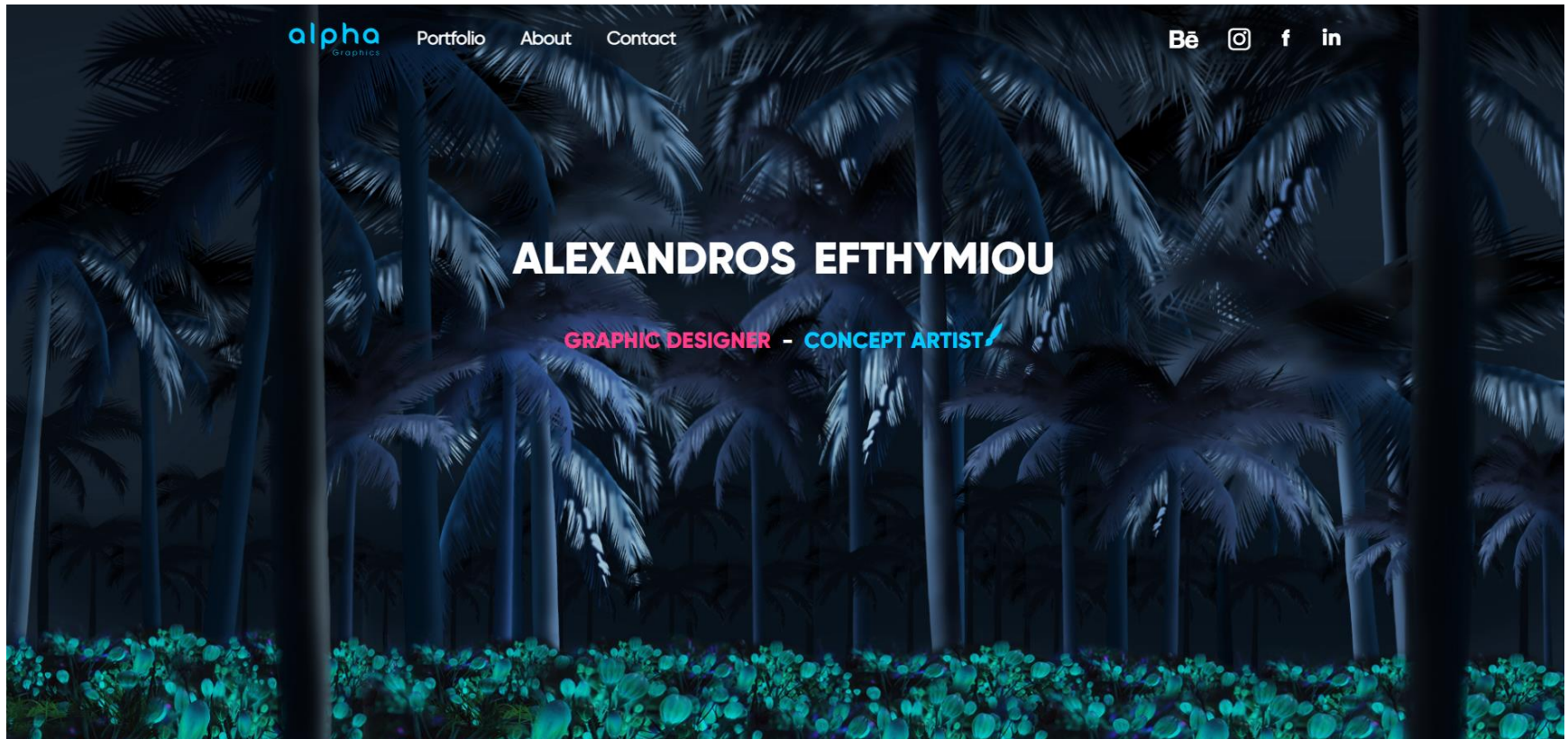
sonarqube 

- Σφάλματα
- Ευπάθειες
- Οσμές κώδικα

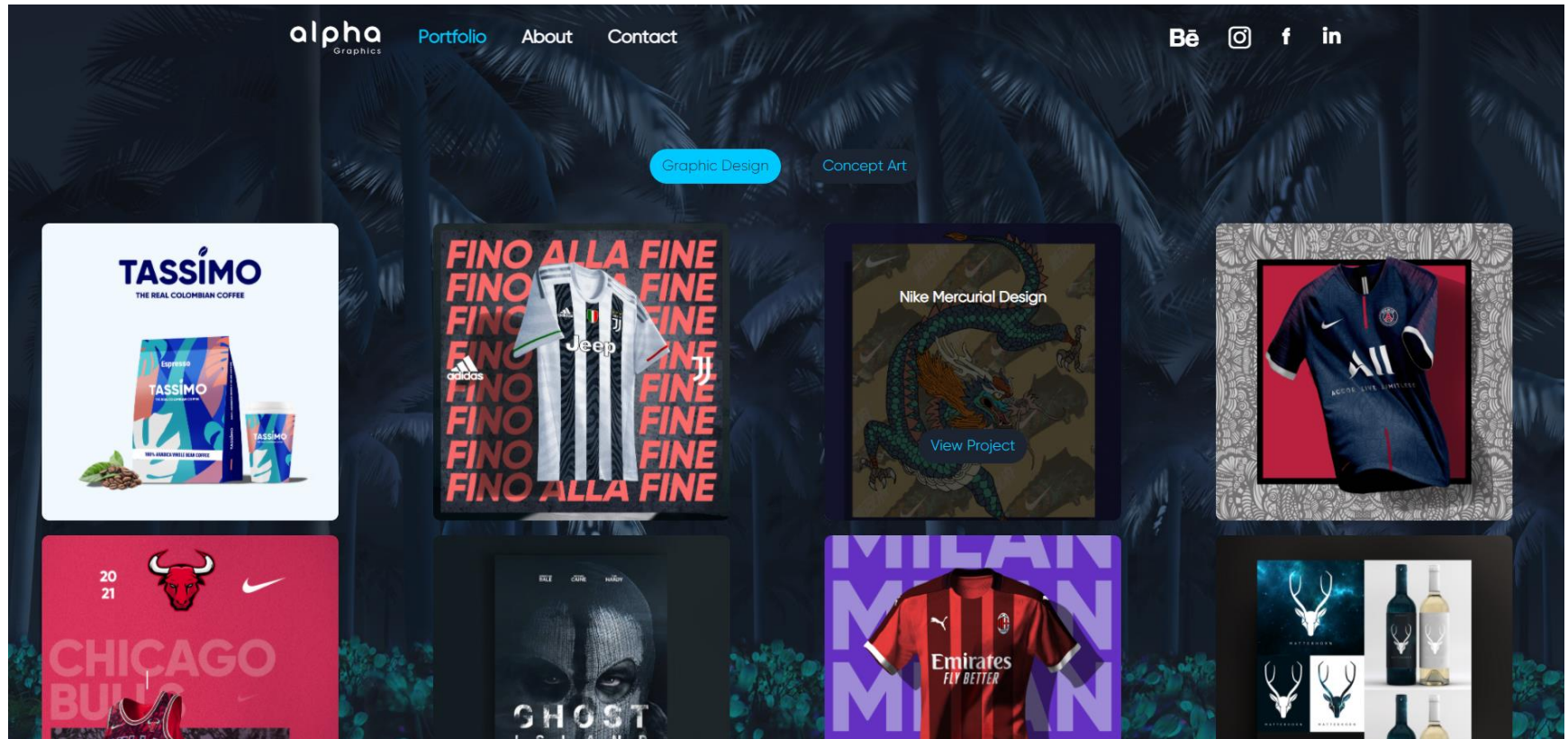
Μετρικές SonarQube

- ▶ Reliability
 - ▶ Security Hotspots
 - ▶ Security
 - ▶ Maintainability
 - ▶ Coverage
 - ▶ Duplications
 - ▶ Size
 - ▶ Complexity (Cyclomatic, Cognitive)
 - ▶ Issues
- 

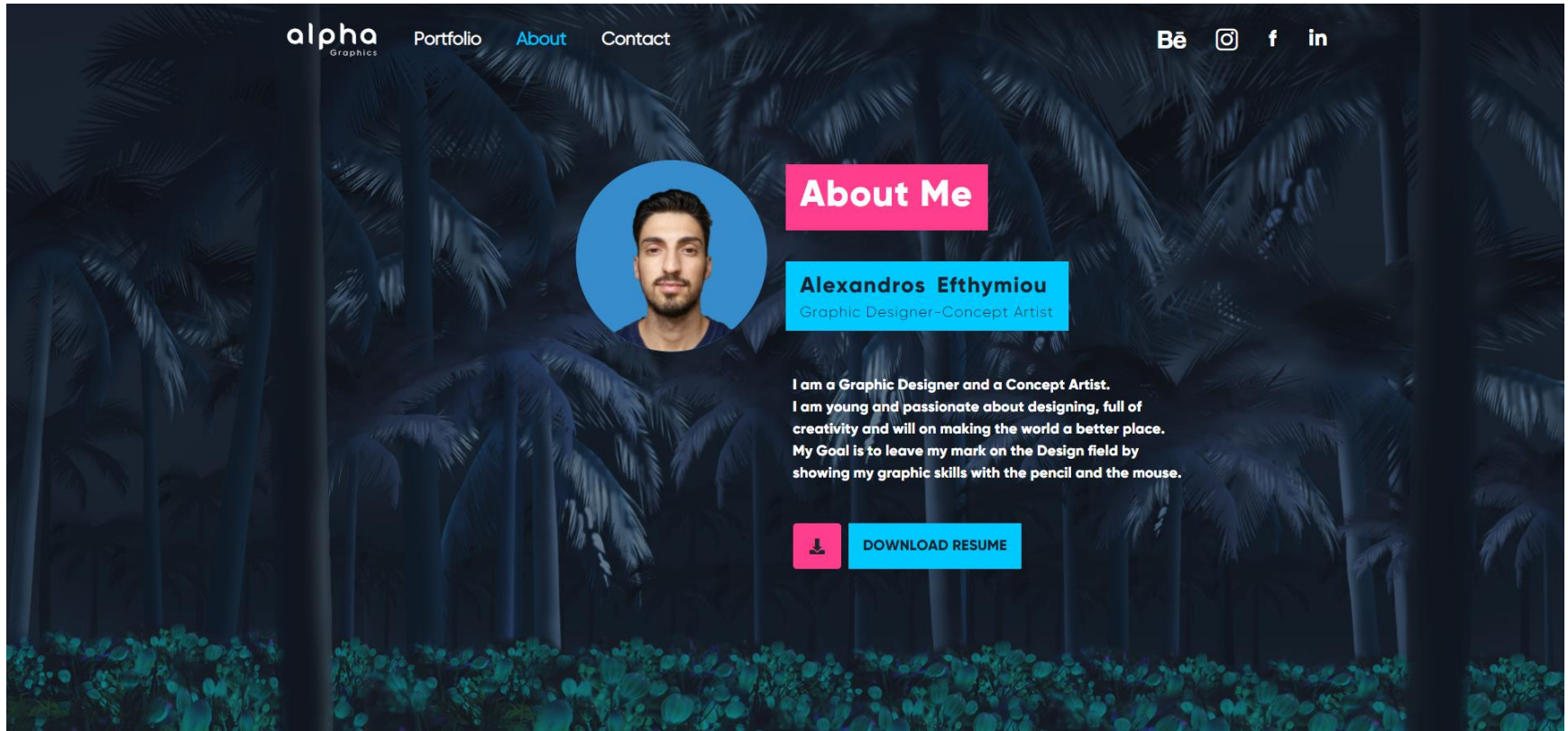
Η Εφαρμογή (Home Page)



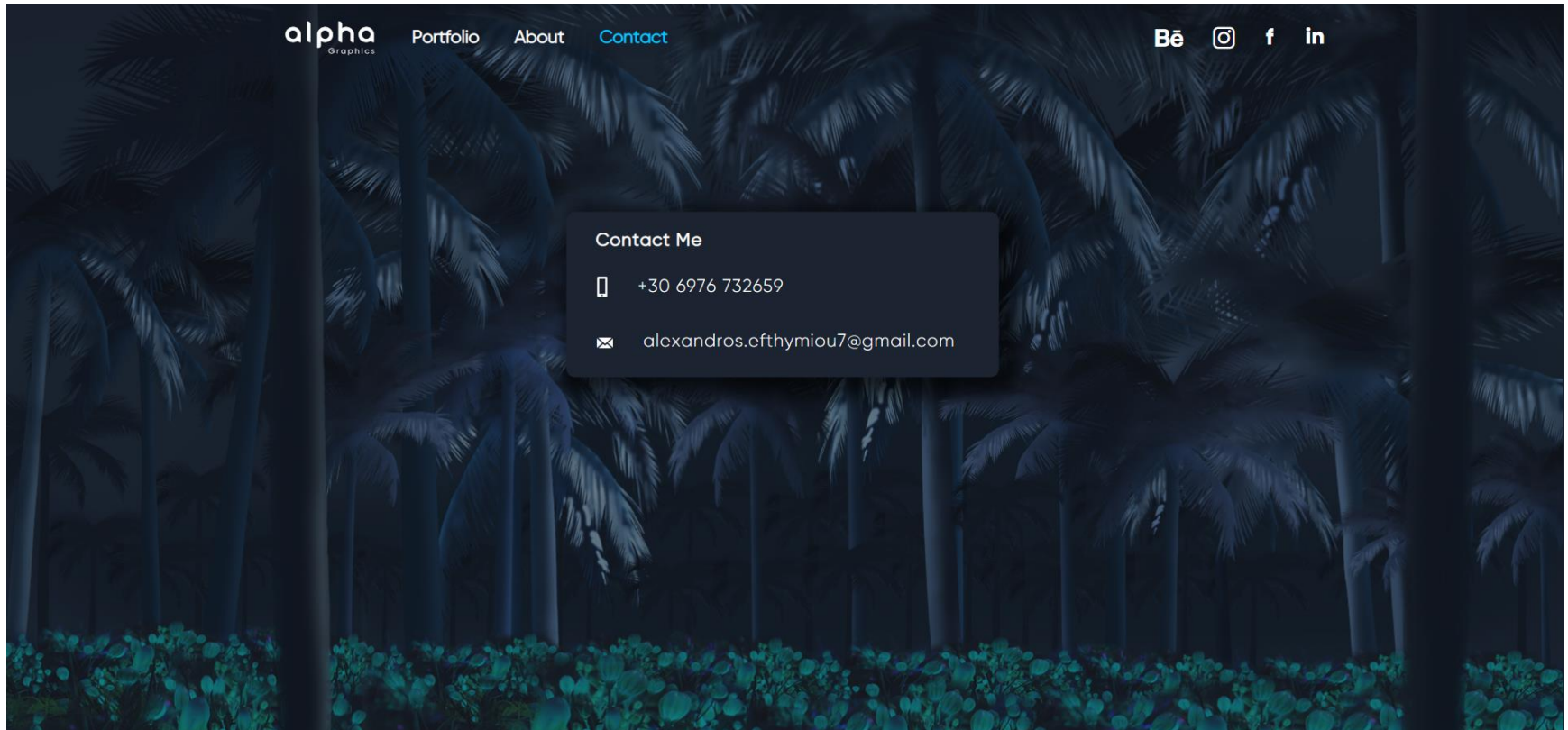
Η Εφαρμογή (Portfolio Page)



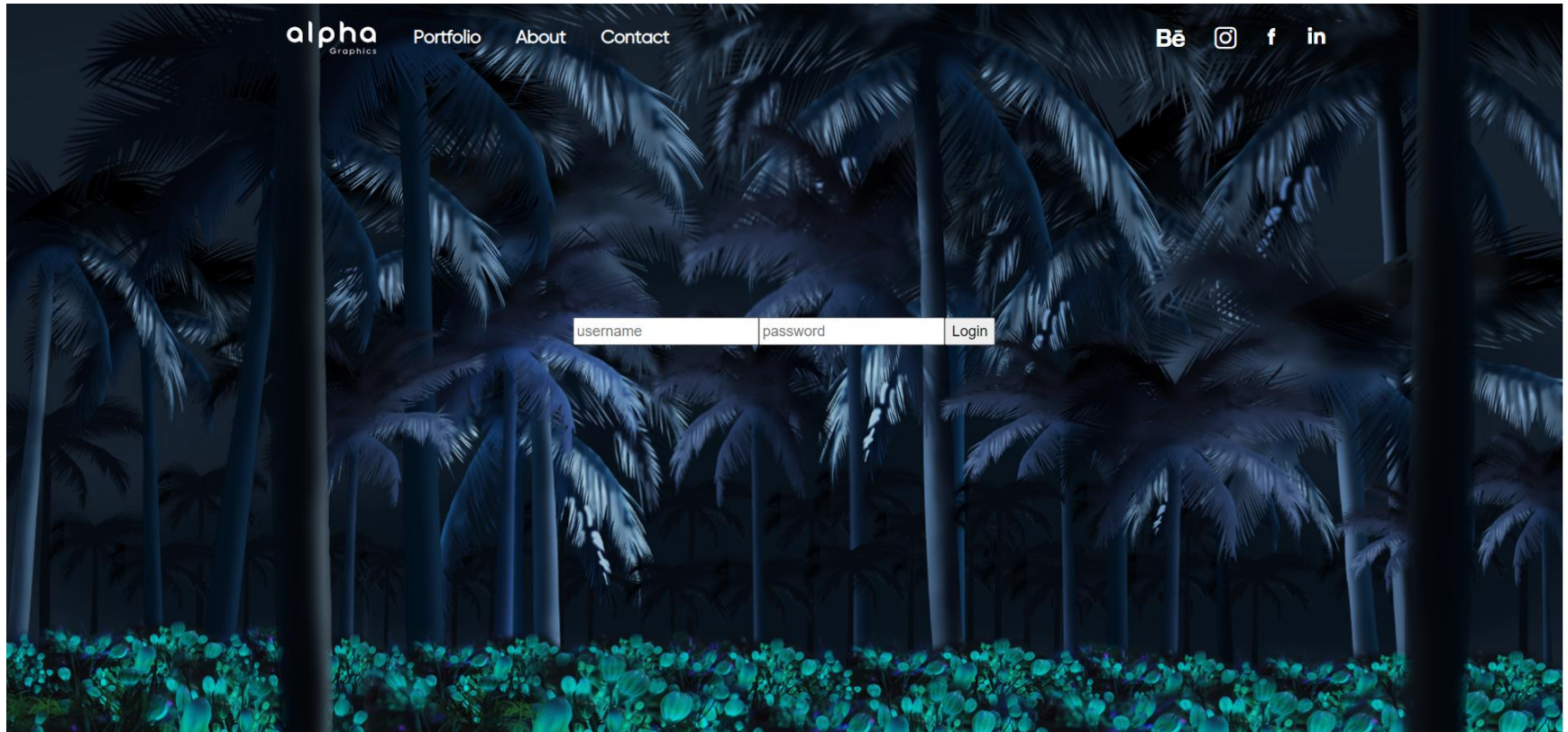
Η Εφαρμογή (About Page)



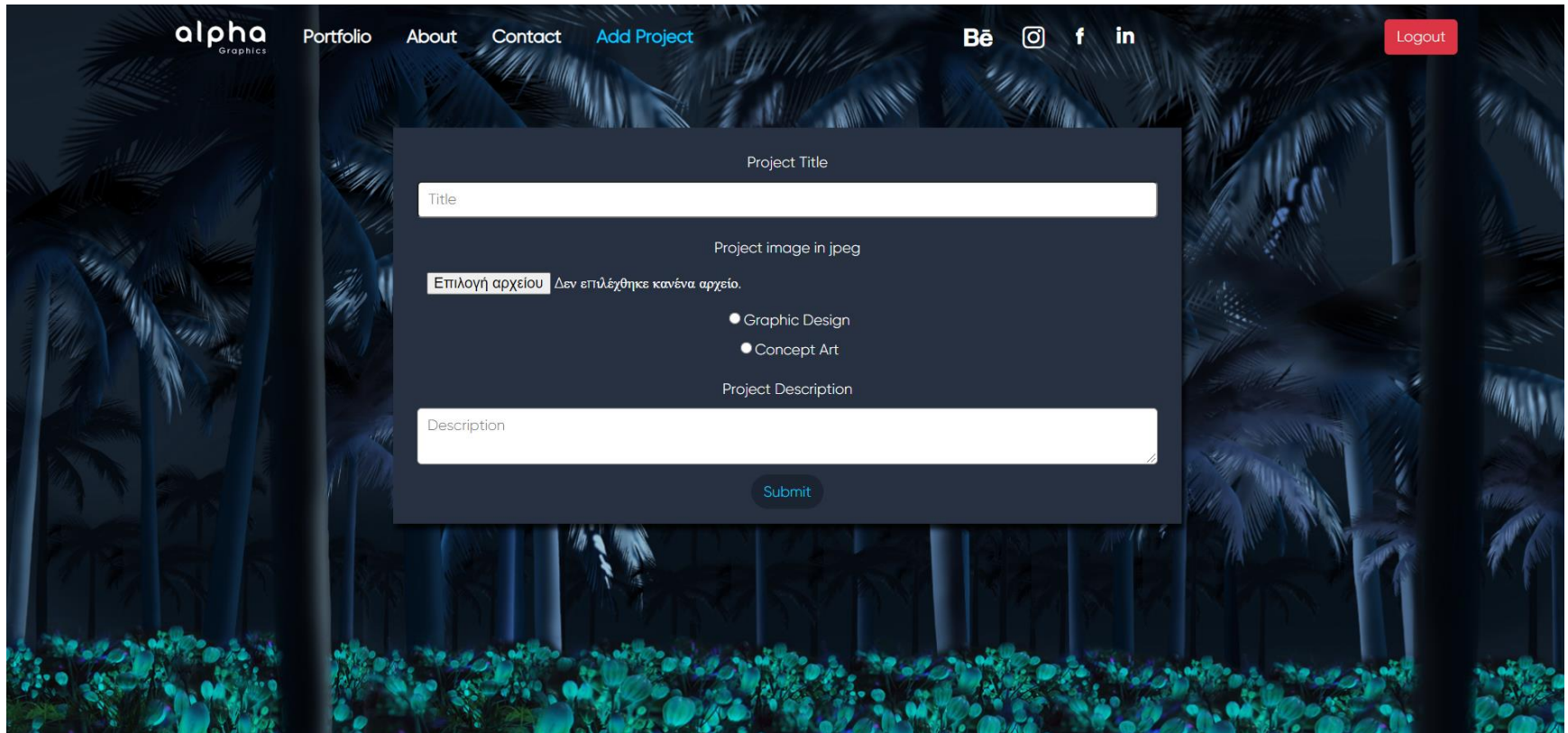
Η Εφαρμογή (Contact Page)



Η Εφαρμογή (Login Page)






Η Εφαρμογή (logged in Add Project Page)



alpha Graphics

Portfolio About Contact [Add Project](#)

Bē   

[Logout](#)

Project Title

Title

Project image in jpeg

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο.

☐ Graphic Design

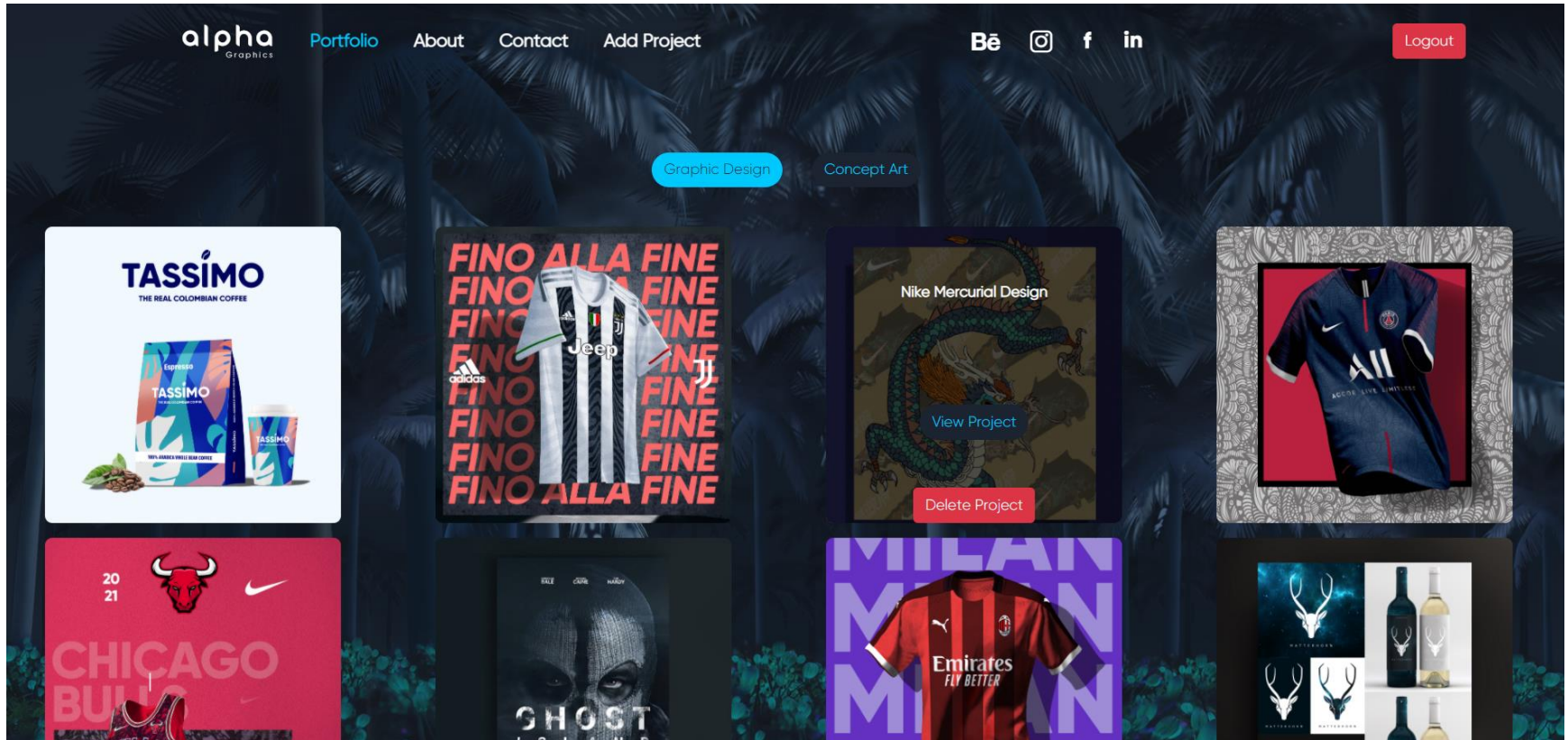
☐ Concept Art

Project Description

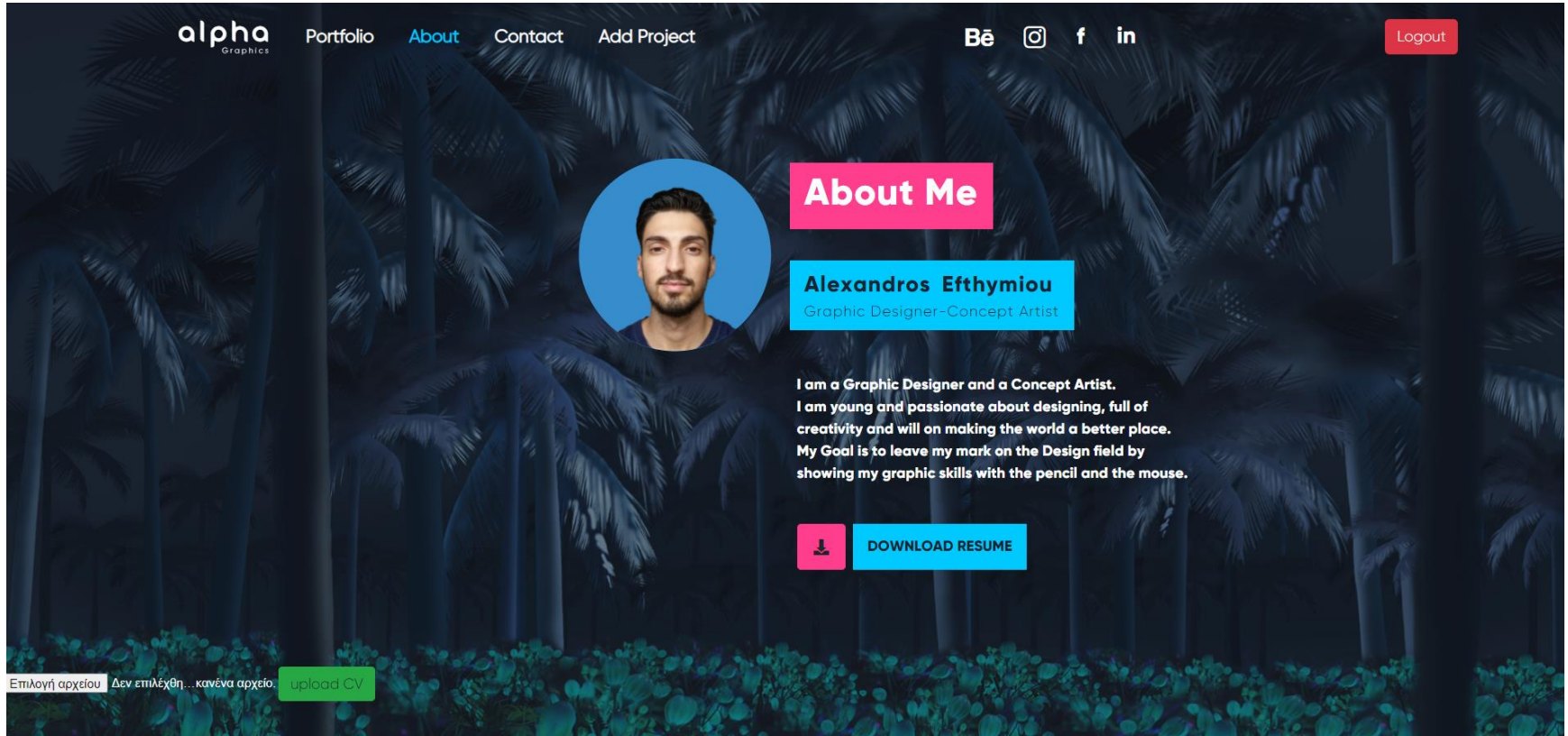
Description

Submit

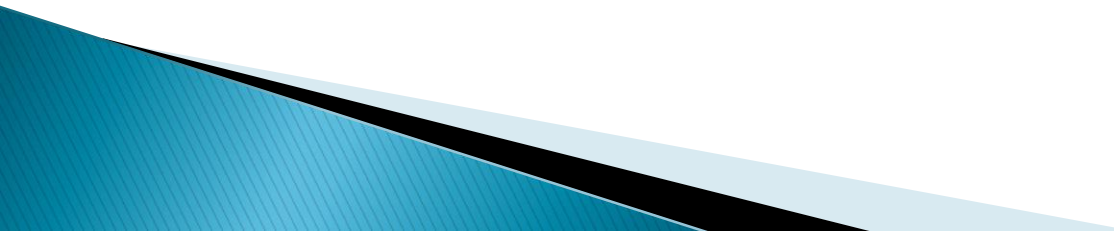
Η Εφαρμογή (logged in Portfolio Page)




Η Εφαρμογή (logged in About page)





Ta Sprints

1. Μενού περιήγησης, routes.
 2. Home Page, social media menu στο Μενού περιήγησης
 3. Portfolio Page, Login Page, authentication/authorization
 4. About Page, Contact Page, AddProject Page, delete project functionality.
 5. Responsive design, Loaders on async requests.
- 

Μετρήσεις


 Angular	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Debt	5mins	4mins	15mins	20mins	23mins
Complexity (cognitive)	0	4	39	56	58

 React	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Debt	3mins	7mins	27mins	45mins	65mins
Complexity (cognitive)	2	7	65	82	87

 Vue	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Debt	4mins	9mins	7mins	21mins	27mins
Complexity (cognitive)	0	0	16	35	35

Συμπεράσματα

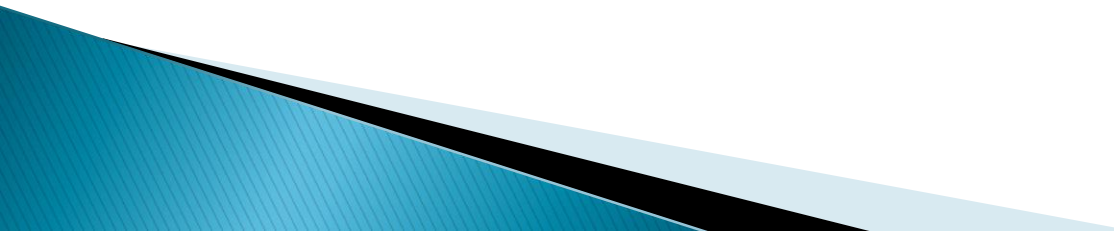
Angular vs React vs Vue (Vol 2)

- ▶ Typescript -> Angular
 - ▶ Καθοδηγητική δομή κώδικα-> Angular
 - ▶ Αντικειμενοστραφές περιβάλλον ανάπτυξης -> Angular
 - ▶ Ευελιξία -> Vue
 - ▶ Εφαρμογή μεγάλου μεγέθους -> Angular
 - ▶ New user friendly -> Vue
 - ▶ Χρήση καινοτόμων και τελευταίων τεχνολογιών-> Vue
 - ▶ Λογική και στυλ στο ίδιο αρχείο -> Vue
 - ▶ Θέλει να δουλέψει με HTML -> Angular
 - ▶ Αυστηρή χρήση JavaScript -> React
- 

Όρια και περιορισμοί

- ▶ Το μέγεθος της εφαρμογής
- ▶ Η ανάλυση των επιμέρους υλοποιήσεων

Μελλοντικές επεκτάσεις

- ▶ Ανάλυση περισσότερων τεχνολογιών (Ember)
 - ▶ Typescript
 - ▶ Μεγαλύτερου μεγέθους εφαρμογή
 - ▶ State management
- 

Ευχαριστώ πολύ!
Ερωτήσεις;