

**ΕΡΓΑΣΙΑ
ΕΙΔΙΚΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ
ΛΟΓΙΣΜΙΚΟΥ**

ΑΝΑΛΥΣΗ

ΣΥΝΟΠΤΙΚΑ

1. Παραδείγματα Sportswear E-shops
2. Συμπεράσματα
3. Μια πρώτη προσέγγιση – Αρχιτεκτονική
Συστήματος
4. Tech Stack
5. Links

The background features a large white circle in the center, partially overlapping a light blue area on the left and a light pink area on the right. A dark blue shape is at the bottom, also overlapping the white circle. The text is centered within the white circle.

ΜΙΑ ΠΡΩΤΗ ΠΡΟΣΕΓΓΙΣΗ

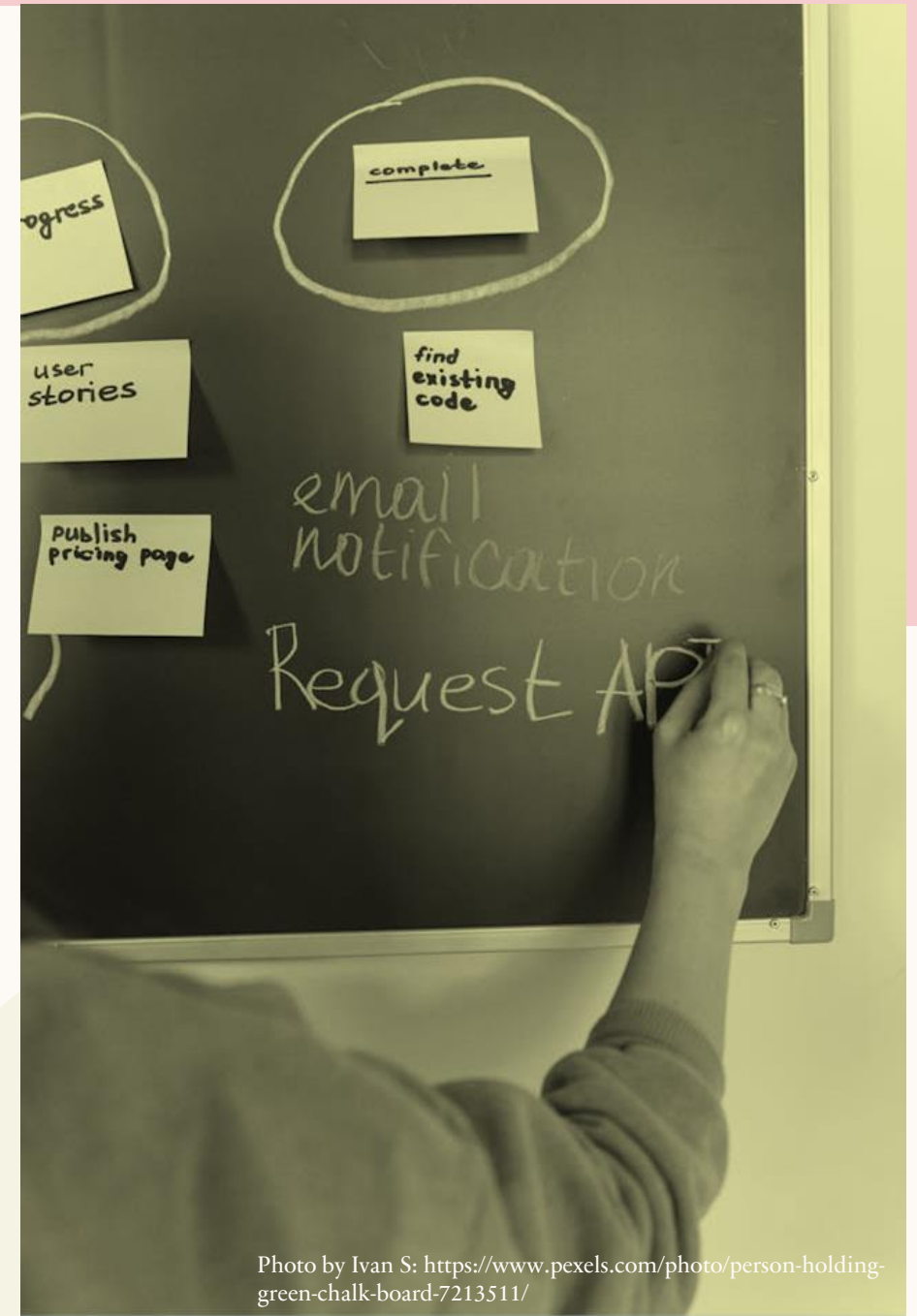
USER STORIES

Μας βοηθούν να:

- δούμε το σύστημα μέσα από τα μάτια του χρήστη,
- να κατανοήσουμε τις απαιτήσεις χωρίς τεχνικές λεπτομέρειες και
- να επικοινωνούμε αποτελεσματικά ως ομάδα.

Περιγράφουν την αξία κάθε λειτουργίας για τον χρήστη και αποτελούν βάση για σχεδιασμό και έλεγχο

Επίσης, είναι απαραίτητα για την εργασία



USER STORIES – (GUEST) USERS



USER STORIES

Guest User

- As a **Guest User** I want to **visit** the website so I can **view** all the products
 - **Details:** This refers to all the product pages being visitable by anyone
 - **Evaluation:** All product pages must be accessible by anyone
- As a **Guest User** I want to be able to **search** so I can **find** what I am looking for
 - **Details:** Search bar must be easily accessible, at any time from all product pages
 - **Evaluation:** User finds what they are looking for

USER STORIES

Guest User

- As a **Guest User** I want to be able to **apply filters** so I can **find** what I am looking for
 - **Details:** Filters are based on item characteristics like color, size, suitability (For Men, Women, Kids), type (shoe, cloth), tags (new season, etc.)
 - **Evaluation:** Filters must work as expected
- As a **Guest User** I want to **register** so I can **have a registered account**
 - **Details:** Credentials must be checked
 - **Evaluation:** User must be visible in the registered users database

USER STORIES – REGISTERED USERS




```
CREATE DATABASE IF NOT EXISTS athleticshop_db;  
USE athleticshop_db;
```

```
CREATE TABLE IF NOT EXISTS users (  
    user_id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL, -- In a real application, you would hash this  
password  
    email VARCHAR(100) NOT NULL UNIQUE,  
    role ENUM('customer', 'admin') NOT NULL DEFAULT 'customer',  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
INSERT INTO users (username, password, email, role) VALUES  
('admin', 'adminpass', 'admin@example.com', 'admin'),  
('customeracc', 'customerpass', 'someone@example.com', 'customer');
```

```
CREATE TABLE IF NOT EXISTS user_profiles (  
    user_profile_id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL UNIQUE,  
    first_name VARCHAR(100),  
    last_name VARCHAR(100),  
    address VARCHAR(255),  
    city VARCHAR(100),  
    postal_code VARCHAR(20),  
    phone_number VARCHAR(20),  
    birthdate DATE,  
    FOREIGN KEY (user_id) REFERENCES users(user_id)  
);
```

```
INSERT INTO user_profiles (user_id, first_name, last_name, address, city, postal_code,  
birthdate, phone_number) VALUES  
(2, 'name', 'surname', '123 E-Commerce Lane', 'Some City', '12345', 01/01/01, '6969696969');
```

```
CREATE TABLE IF NOT EXISTS products (  
    product_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    description TEXT,  
    image_url VARCHAR(255),  
    price DECIMAL(10, 2) NOT NULL,  
    stock INT NOT NULL DEFAULT 0,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE IF NOT EXISTS cart (  
    — cart_id INT AUTO_INCREMENT PRIMARY KEY,  
    — user_id INT NOT NULL,  
    — product_id INT NOT NULL,  
    — quantity INT NOT NULL CHECK (quantity > 0),  
    — FOREIGN KEY (user_id) REFERENCES users(user_id),  
    — FOREIGN KEY (product_id) REFERENCES products(product_id),  
    — -- Ensures a user can only have one entry per product in their cart  
    — UNIQUE KEY user_product (user_id, product_id)  
);  
//
```

```
CREATE TABLE IF NOT EXISTS orders (
  order_id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  total_amount DECIMAL(10, 2) NOT NULL,
  status ENUM('pending', 'completed', 'shipped', 'cancelled') NOT NULL DEFAULT
'pending',
  shipping_address TEXT NOT NULL,
  list_of_products [1,23,5,52, 222, ...] (product Ids),
  quantities [12, 3, 1, 4, 5, ...] (quantities for each product) (update stock)
  payment_method ENUM('cash', 'credit_card', 'paypal') NOT NULL, -- Or other methods
  FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

**** (Πίνακας για το wishlist)**

```
CREATE TABLE IF NOT EXISTS order_details (
  order_detail_id INT AUTO_INCREMENT PRIMARY KEY,
  order_id INT NOT NULL,
  product_id INT NOT NULL,
  quantity INT NOT NULL CHECK (quantity > 0),
  price_per_item DECIMAL(10, 2) NOT NULL,
  FOREIGN KEY (order_id) REFERENCES orders(order_id),
  FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

USER STORIES

Registered User

- As a **Registered User** I want to **Login** so I can **access** registered user content
 - **Details:** Credentials must be verified, **token** will be given.
 - **Evaluation:** User is now logged in and is visible via the GUI.
- As a **Registered User** I want to **add** items in my Cart so I can **order** them.
 - **Details:** Order button inside the cart. Ability to choose which items and quantity to order.
 - **Evaluation:** User must be able to create an order with the Items in their Cart.
- As a **Registered User** I want to **view Orders** so I can **see** all my Orders and their **Status**
 - **Details:** Order statuses: Created, Processing, Out for Delivery, Completed, Delivery Failed, Processing Failed, Return. Some statuses must have a message for the user.
 - **Evaluation:** User can see any orders they have made in the past or now and their status.

USER STORIES

Registered User

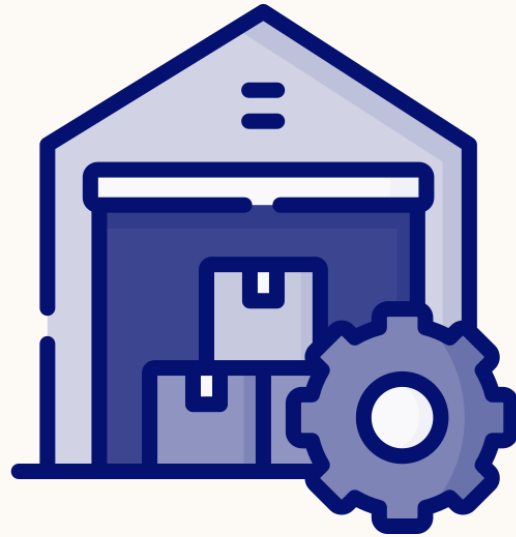
- As a **Registered User** I want to **manage** my User Profile so I can **change** my profile information
 - **Details:** Credentials must be verified, **token** will be given.
 - **Evaluation:**
- As a **Registered User** I want to **edit an Order** so I can **edit** the Order.
 - **Details:** Only if order status is “Ongoing”.
 - **Evaluation:**
- As a **Registered User** I want to **cancel** so I can **cancel** my Order
 - **Details:** When order status is “Ongoing”. When “shipped” only admin
 - **Evaluation:**

USER STORIES

Registered User

- As a **Registered User** I want to **wishlist** an Item so I can **save** Products for later order
 - **Details:** User will be able to select which products to put in cart so they can order them
 - **Evaluation:**
- As a **Registered User** I want to **Log Out** so I can **end** my session securely
 - **Details:**
 - **Evaluation:** User is now logged in and is visible via the GUI.
- *As a **Registered User** I want to create a **Support Ticket** so I can **communicate** with the Admin. (Θα δούμε αν θα το υλοποιήσουμε).

USER STORIES – INVENTORY MANAGER (ADMIN)



USER STORIES

Inventory Manager (ADMIN)

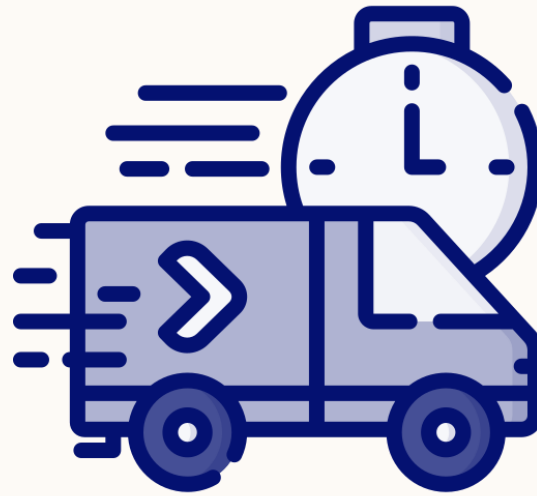
- As an **Inventory Manager** I want to **Login** so I can **access** the Inventory system
 - **Details:** Credentials must be verified, **token** will be given. User can see all the products. Special database user?
 - **Evaluation:** User is now logged in as the Inventory Manager.
- As an **Inventory Manager** I want to **add** items so I can **add** them in the Inventory System.
 - **Details:** Add button. Ability to choose which items and quantity to order.
 - **Evaluation:** User must be able to create an order with the Items in their Cart.
- As an **Inventory Manager** I want to **remove** Items so I can **remove** them from the Inventory System
 - **Details:** Need to ask the user twice before removing the item.
 - **Evaluation:** Item is successfully removed

USER STORIES

Inventory Manager

- As an **Inventory Manager** I want to **edit** Products so I can **edit** their characteristics
 - **Details:** Must have an apply button. Also Verify the edited characteristics are ok. Characteristics include: color, size, tags, category etc.
 - **Evaluation:** Item is edited.

USER STORIES – DELIVERY/ORDER (D/O) MANAGER (ADMIN)



USER STORIES

Delivery/Order (D/O) Manager (ADMIN)

- ~~As a D/O Manager I want to Login so I can access the D/O system~~
 - ~~Details: Credentials must be verified, token will be given. User can see all the orders, and delivery carriers.~~
 - ~~Evaluation: User is now logged in as the Inventory Manager.~~
- ~~As a D/O Manager I want to add delivery carriers so I can add them in the D/O System.~~
 - ~~Details: No additional functionality for now.~~
 - ~~Evaluation: Delivery carrier must created in the database~~
- ~~As a D/O Manager I want to accept Orders so I can change their status to accepted~~
 - ~~Details: Need to ask the user twice before removing the item.~~
 - ~~Evaluation: Item is successfully removed~~

USER STORIES

Delivery/Order (D/O) Manager

- As a **D/O Manager** I want to **edit** an Order so I can **change** the Orders' parameters.
 - **Details:** No additional functionality for now.
 - **Evaluation:** Delivery carrier must created in the database
- As a **D/O Manager** I want to **reject** Orders so I can **change** their status to failed
 - **Details:** Manager must supply message with reason.
 - **Evaluation:** Order is rejected

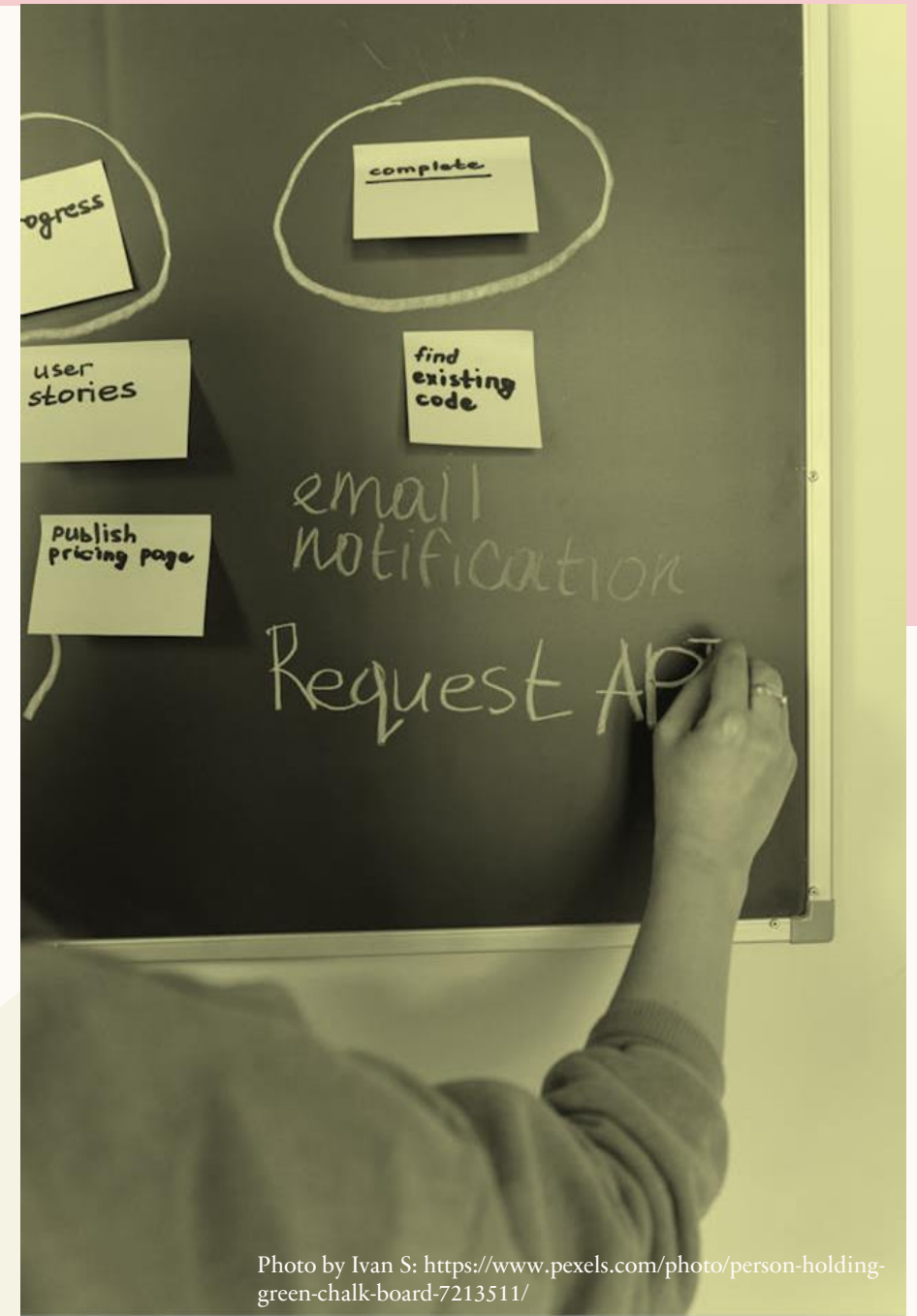
USE CASES

Μας βοηθούν να:

- δούμε το σύστημα με μεγαλύτερη λεπτομέρεια, όχι τόσο τεχνική
- να κατανοήσουμε τις απαιτήσεις

Βοηθούν στην κατανόηση των λειτουργιών που πρόκειται να αναπτυχθούν.

Σε ένα βαθμό έχουν ήδη υλοποιηθεί, οπότε απλά θα προσθέσουμε ό,τι χρειάζεται.

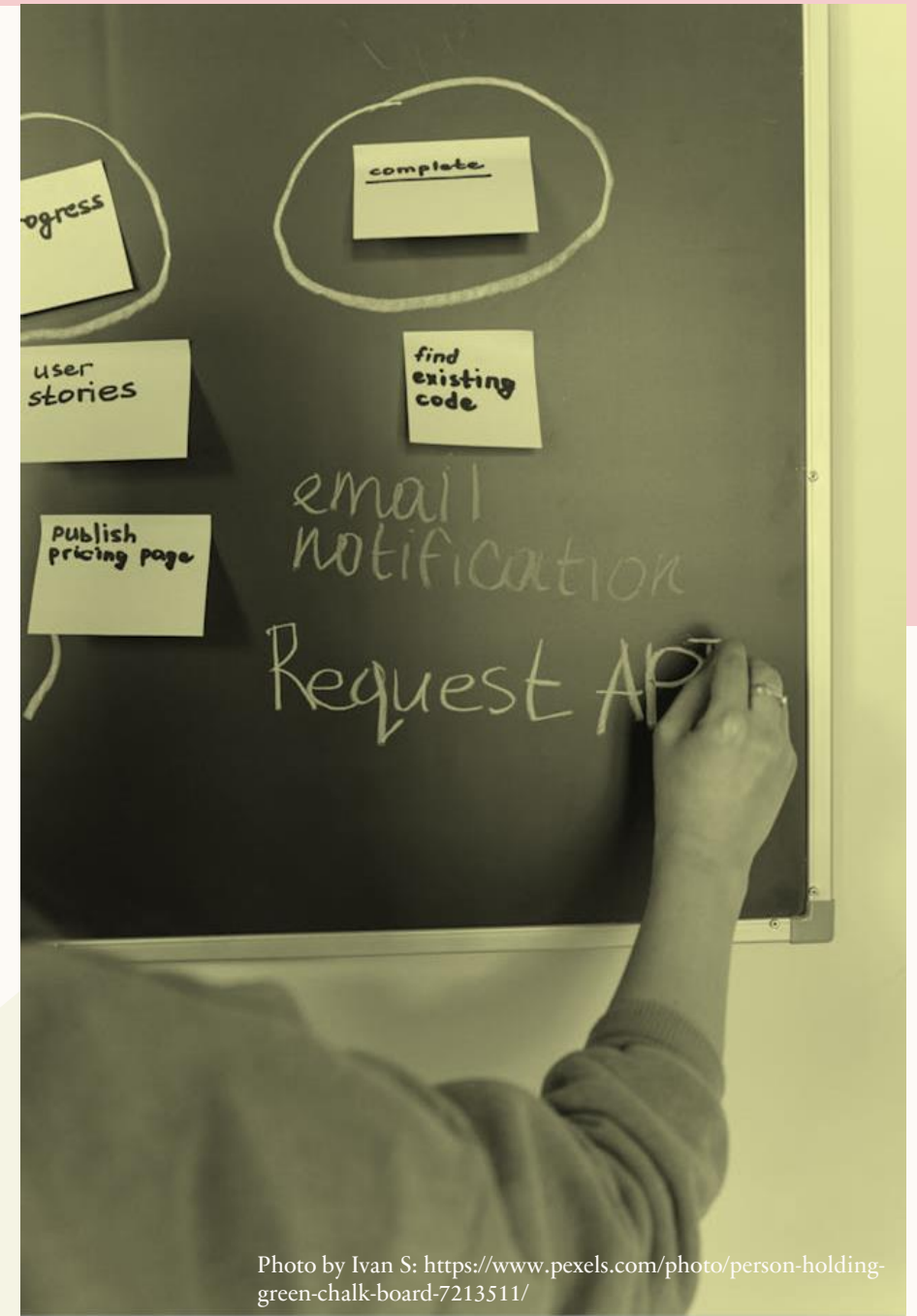


USE CASES

COMPONENT DIAGRAM

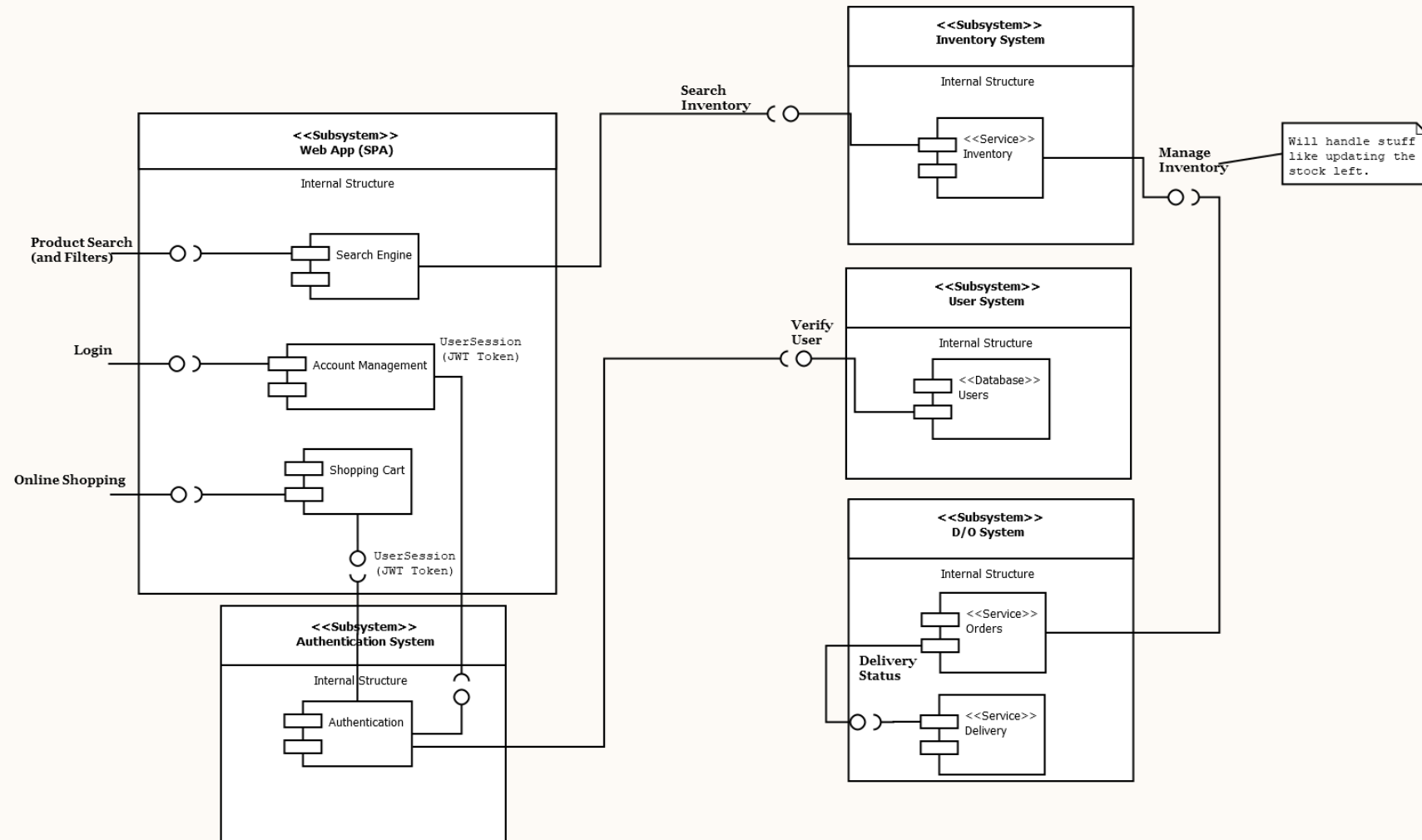
Μας βοηθάει να:

- δούμε το σύστημα πιο αφηρημένα και ολοκληρωμένα,
- να κατανοήσουμε τα διάφορα υποσυστήματα που το απαρτούν και
- να οργανώσουμε πιο αποτελεσματικά την δουλειά της ομάδας.



SPORTSWEAR E-SHOP – COMPONENT DIAGRAM

25





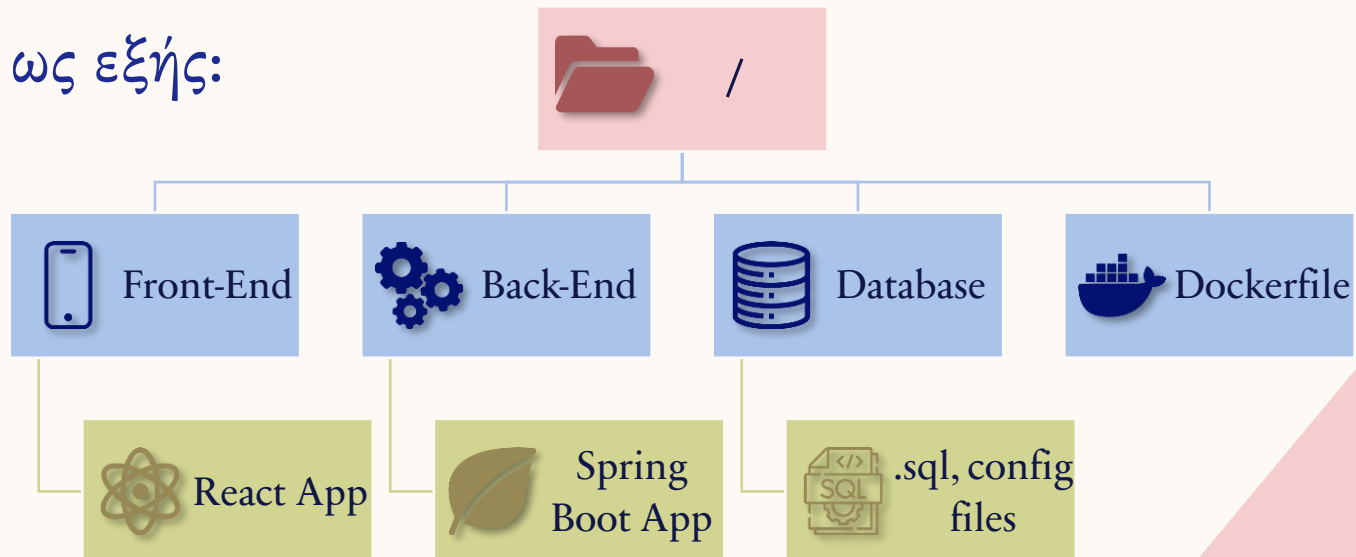
GITHUB & GIT TUTORIAL

GITHUB REPO

Στείλτε μου μόλις φτιάξετε λογαριασμούς ή εάν έχετε ήδη, τα **email** , **username** σας για να σας προσθέσω στο project.

ΠΩΣ ΕΙΝΑΙ ΔΟΜΗΜΕΝΟ ΤΟ REPOSITORY ?

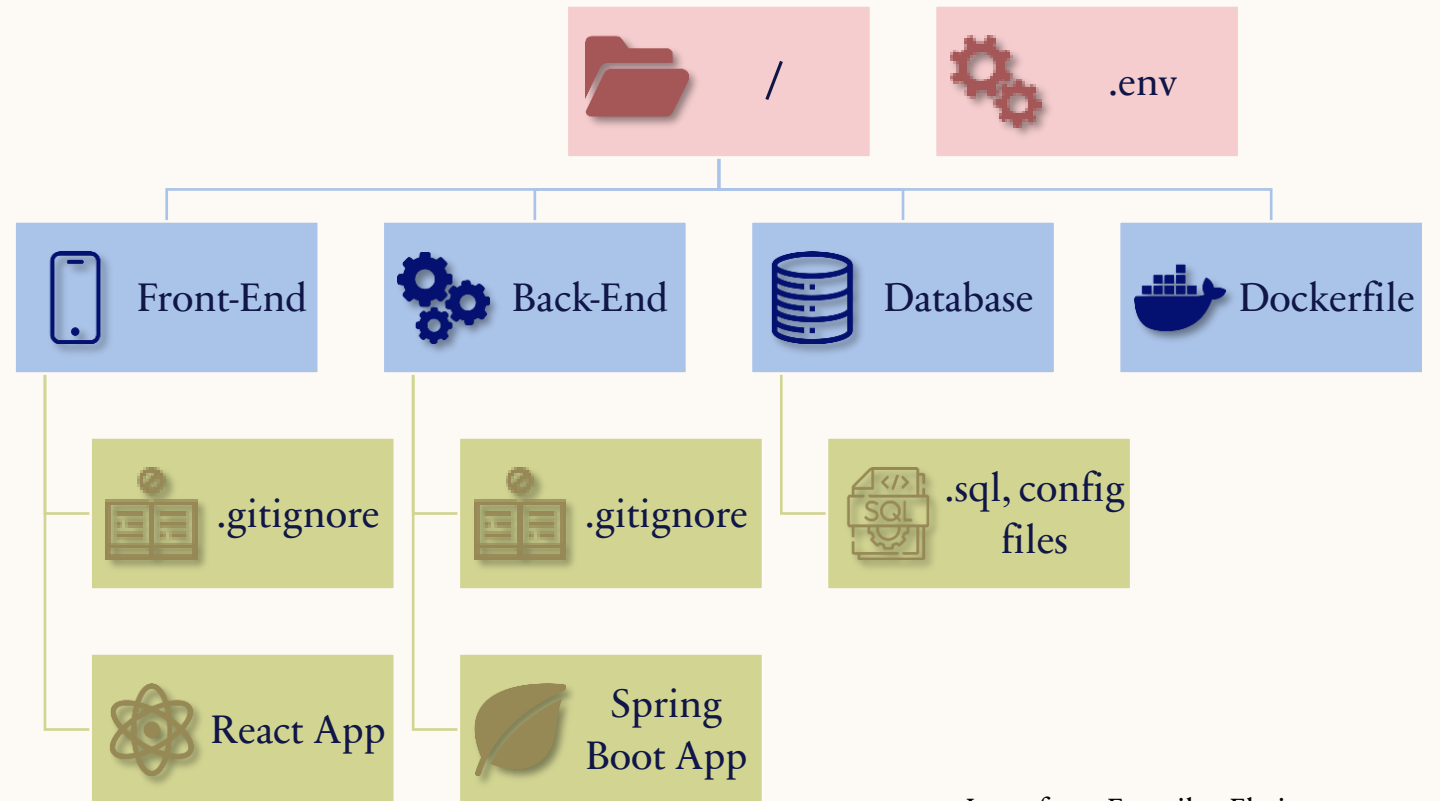
Όπως είχαμε δει και στις πρώτες διαφάνειες της δεύτερης συνάντησης, η δομή του repository έχει ως εξής:



ΠΩΣ ΕΙΝΑΙ ΔΟΜΗΜΕΝΟ ΤΟ REPOSITORY ?

Σε μεγαλύτερη λεπτομέρεια έχουμε τα εξής:

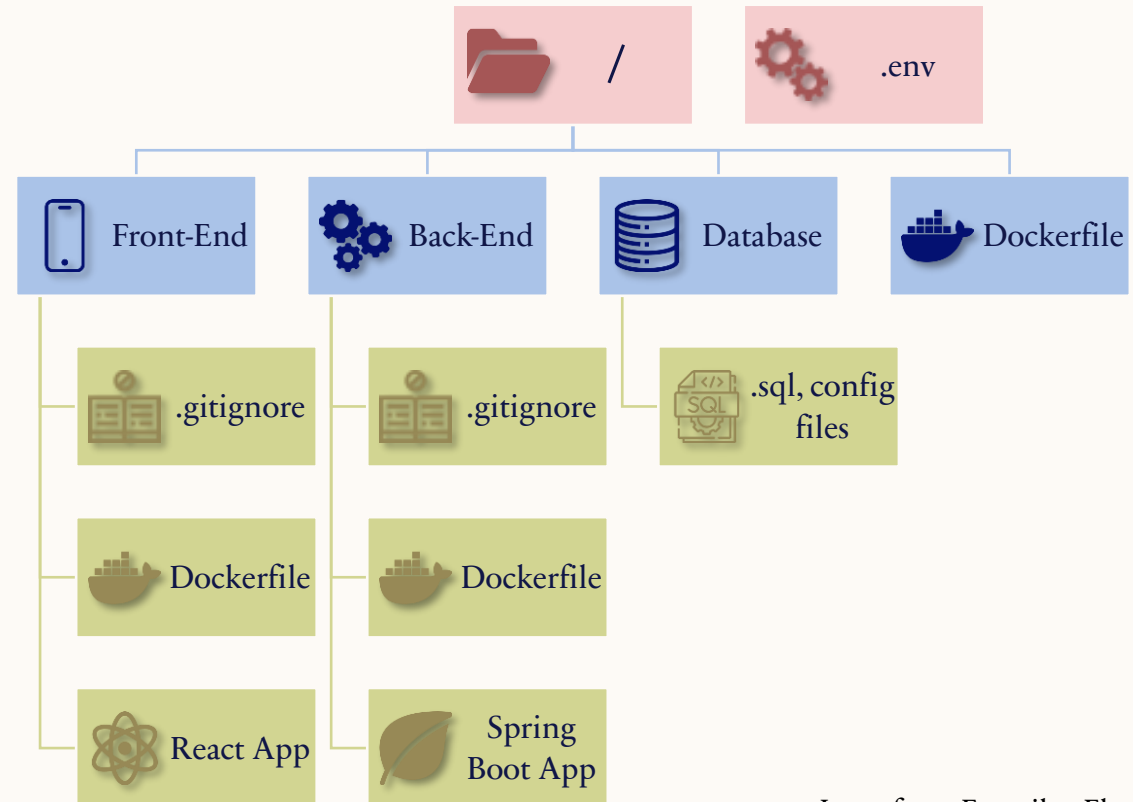
- Πολύ σημαντικό, τα **.gitignore** για κάθε φάκελο ώστε να μην ανεβαίνουν «άχρηστα» αρχεία. (Συνήθως παράγονται αυτόματα από τα IDE αλλά καλό είναι να τα ελέγξετε)



ΠΩΣ ΕΙΝΑΙ ΔΟΜΗΜΕΝΟ ΤΟ REPOSITORY ?

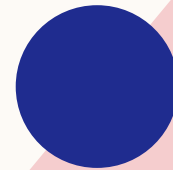
Σε μεγαλύτερη λεπτομέρεια έχουμε τα εξής:

- **.env** – Είναι αρχείο για το **docker** και περιέχει μεταβλητές και τιμές όπως, κωδικούς για την ΒΔ, και άλλες ρυθμίσεις. Το αρχείο αυτό βρίσκεται στον **server** και δεν το χρειάζεται κανείς.
- Θα χρειαστούμε όμως τους κωδικούς πρόσβασης. Αλλά σε ξεχωριστό αρχείο εκτός του repository.



GITHUB DESKTOP

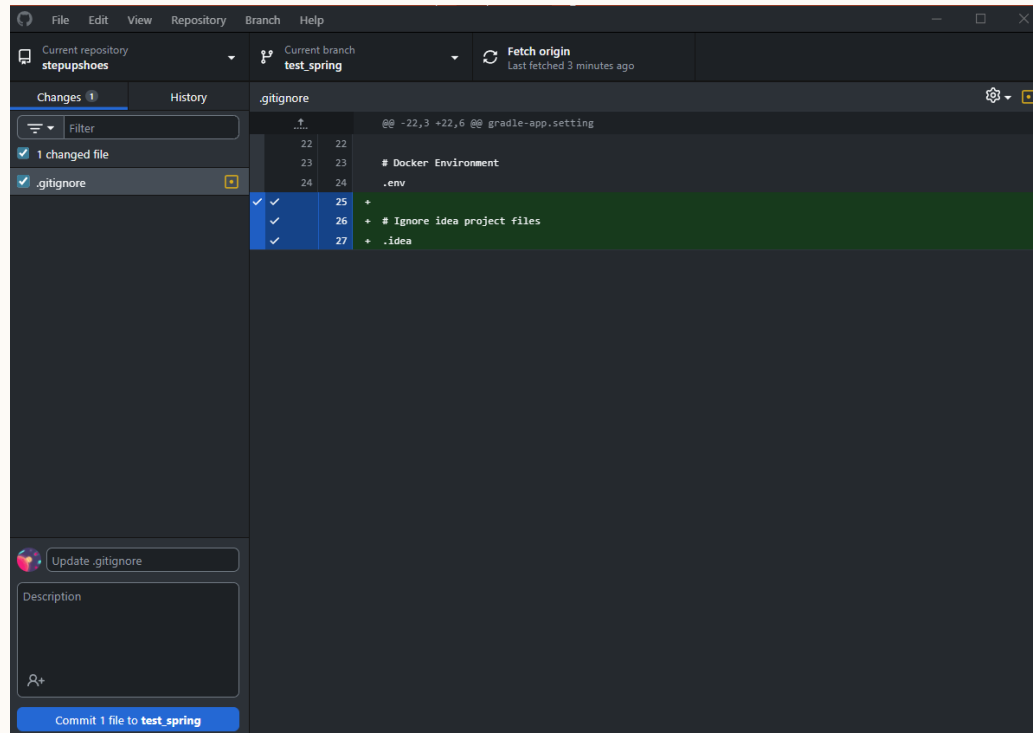
GUI εργαλείο για την διαχείριση του repo.



GITHUB DESKTOP – ΕΝΑΡΞΗ

1. Κατεβάζουμε το Github Desktop
2. Κάνουμε σύνδεση με τον λογαριασμό μας
3. Κάνουμε copy/paste το link από την ιστοσελίδα του repo
4. Και θα κατεβεί το αντίγραφο του repo

GITHUB DESKTOP – ЕНАПЭН



GITHUB DESKTOP – ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

Fetching

«Τραβάει» όλες τις αλλαγές που έχουν γίνει στο repository, όμως δεν τις εφαρμόζει στους φακέλους μας άμεσα.

GITHUB DESKTOP – ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

Pulling

«Τραβάει» όλες τις αλλαγές που έχουν γίνει στο repository και τις εφαρμόζει.

GITHUB DESKTOP – ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

Branches

Καλή πρακτική (και όπως συστήθηκε στο μάθημα) είναι η κάθε ομάδα ή καλύτερα το κάθε μέλος να φτιάχνει δικό του **branch** (**developer branches**) και ό,τι αλλαγές κάνει να τις κάνει εκεί και μετά όταν τελειώσει να κάνει **push**.

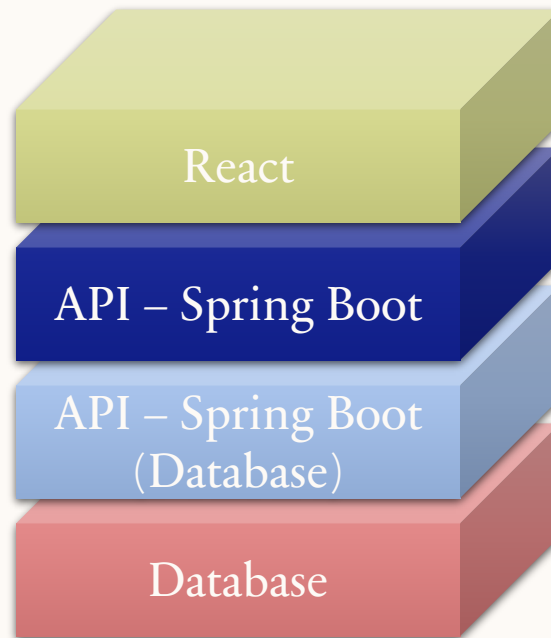
GITHUB DESKTOP – ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

Branches

Δηλαδή π.χ. το μέλος vale θα έχει το branch vale/
και κάτω από αυτό τον «φάκελο» θα φτιάχνει τα
δικά του “feature branches” δηλαδή ένα branch
ανά feature. Παράδειγμα: vale/NewCartFeature

TECH STACK

9. TECH STACK



SERVER

Development

- Ο **Server** θα είναι ρυθμισμένος και θα δημιουργηθούν χρήστες για το κάθε μέλος της ομάδας
- Ο **Server** χρησιμεύει κυρίως για κοινή πρόσβαση στην βάση δεδομένων
 - Παράκληση, να μην εκτελούμε ενέργειες έως ότου συμφωνήσουν όλα τα μέλη της ομάδας

Deployment

- Κατά κύριο λόγο θα χρησιμοποιήσουμε docker για το project
- Ο **server** θα είναι ανοιχτός 24/7
- Θα λάβετε κωδικούς μέσω mail μόλις είναι έτοιμος
- Προσοχή!!! Αλλάξτε τον κωδικό σας το συντομότερο δυνατόν:
\$> **passwd**
Old password: *****
New password: *****

REACT

Development

- Θα χρειαστεί βιβλιοθήκη για το **routing** , όπως π.χ. **React Router**
- Όσο γίνεται πιο απλό, με όσο το λιγότερο βιβλιοθήκες component. **Standard Components**

Για την ανάπτυξη αρκούν **VS Code + nodejs server**

Deployment

- Το deployment θα γίνει στο τέλος σε **server** , με τη βοήθεια του **docker**. (Σχετικά εύκολη διαδικασία)
- Για μεγαλύτερη ευκολία, και σωστή ανάπτυξη , **πρέπει να αποφασίσουμε σε μια έκδοση του nodejs** και ίσως του react.
- Το npm θα χειριστεί τα υπόλοιπα
- Θα πρέπει να δούμε πως θα χειριστούμε τις εικόνες των προϊόντων



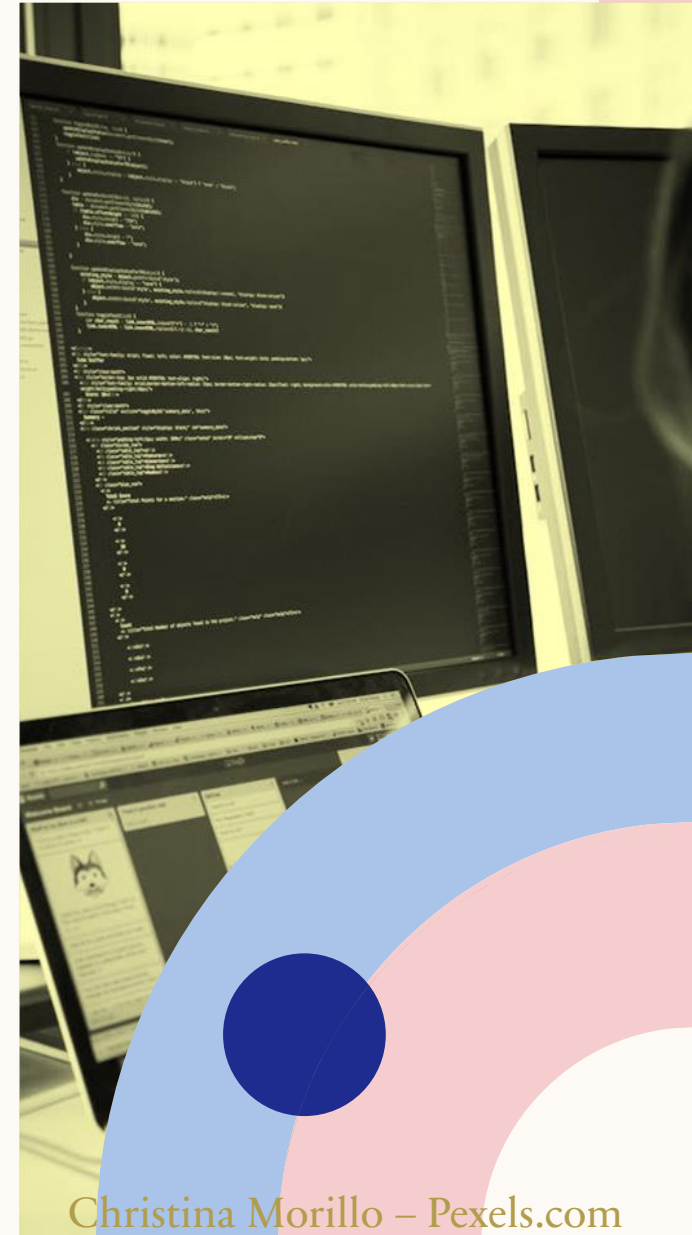
SPRING BOOT

Development

- RESTful Web Services -> Spring Boot Web
 - ORM -> Hibernate
 - Τα **API endpoints** θα προκύψουν κατά βάση από την σχεδίαση του συστήματος, θα πρέπει όμως να είναι **οργανωμένα σε κεντρικό σημείο**.
 - Όπως είπαμε θα χρησιμοποιήσουμε το **IntelliJ IDEA**

Deployment

- Deployment -> Docker
- Καλό θα είναι να χρησιμοποιούμε όλοι την ίδια έκδοση Spring Boot & Java
- Επίσης καλύτερα ένα μέλος της ομάδας να φτιάξει το αρχικό project folder στην ιστοσελίδα <https://start.spring.io/>
- Το Gradle θα χειριστεί ό,τι άλλα πακέτα χρησιμοποιούμε



DATABASE - MYSQL

Development

- Θα κάνω **setup** στον **server** ώστε να έχουμε πρόσβαση όλη η ομάδα
- **Development version**, και συχνά backups*
 - Συνιστώ τη χρήση του **MySQL Workbench** (GUI, εύκολο στη χρήση) ή κάποιο άλλο GUI db admin (π.χ. Dbeaver)

Deployment

- Μάλλον με **docker** -> Θα κάνει την εφαρμογή ακόμα πιο μεταφέρισιμη (όλα σ' ένα repository – docker compose)



GITHUB REPOSITORY

Development

- Πρέπει να αποφασίσουμε στην δομή / ιεραρχία των φακέλων
- Κρατάμε την δομή όπως είναι και δεν την χαλάμε

Deployment

- Ένα μέλος της ομάδας φτιάχνει το repository και κάνει invite τα υπόλοιπα

**ΠΑΡΑΔΕΙΓΜΑΤΑ
SPORTSWEAR
E-SHOP**

ADIDAS.COM

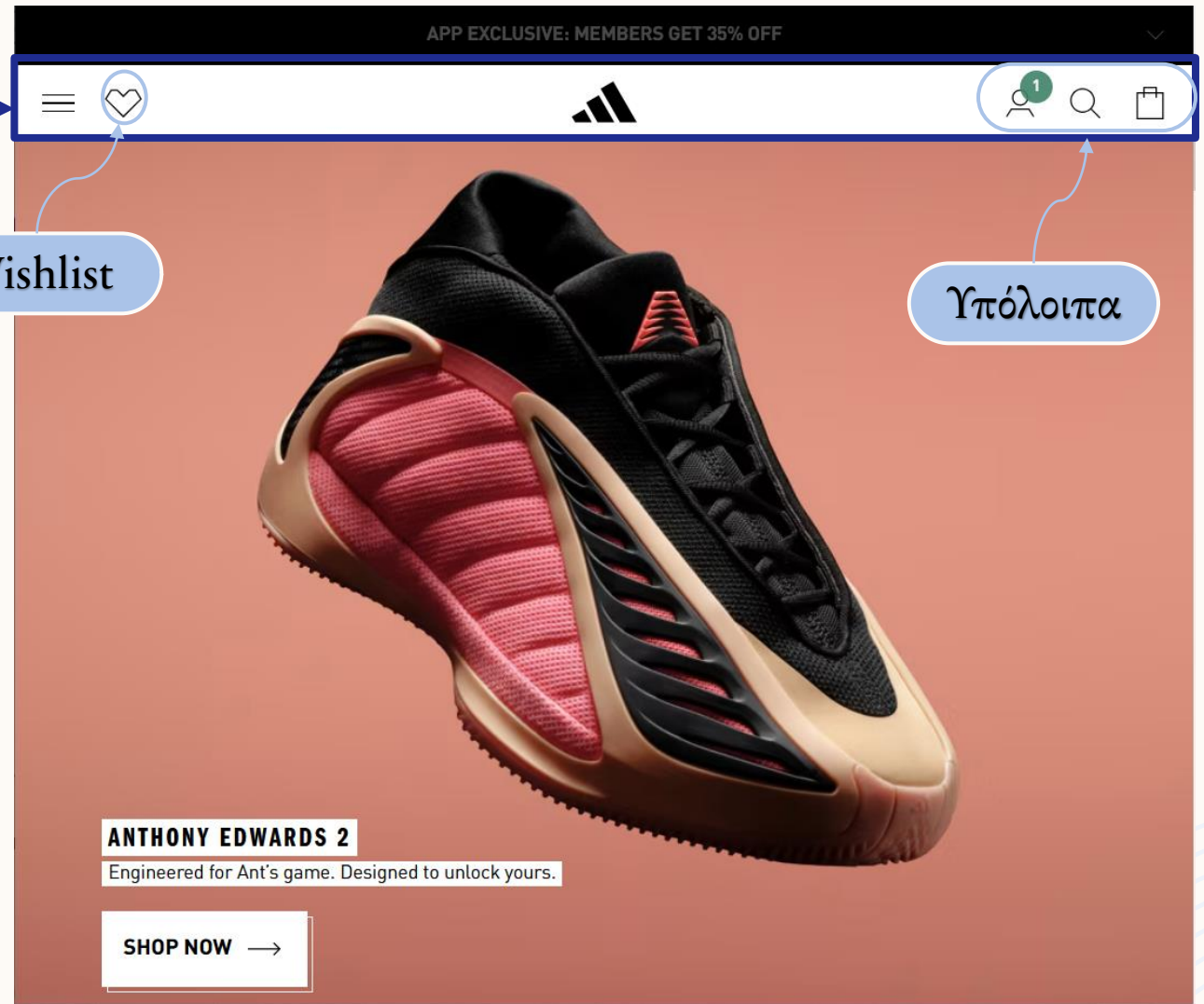
Landing Page:

- Νέα & Διάσημα προϊόντα στην αρχική
- **Navbar:**
 - Μενού
 - Wishlist
 - Χρήστης (Επισκέπτης – Μη-συνδεδεμένος)
 - Αναζήτηση
 - Καλάθι
- **Design:**
 - Απλό & Μοντέρνο, χωρίς να υπερφορτώνει τον ελεύθερο χώρο

Navbar

Wishlist

Υπόλοιπα

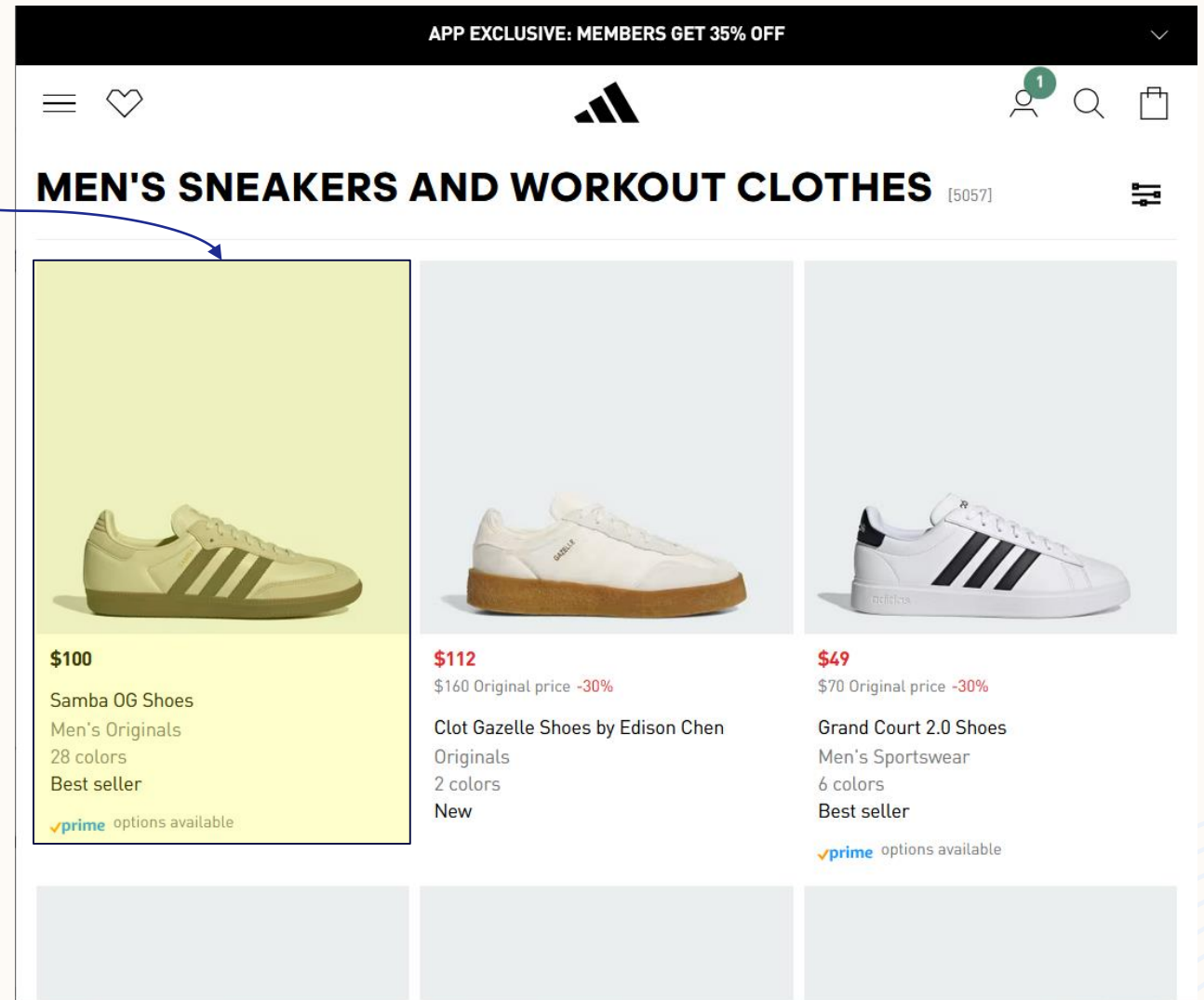


ADIDAS.COM

Products Page:

- **Product card:**
 - Εικόνα προϊόντος
 - Τιμή (ειδική τιμή σε περίπτωση προσφοράς, εδώ έχω μπει ως μέλος και εμφανίζονται οι ειδικές εκπτώσεις μέλους)
 - Όνομα προϊόντος
 - Κατηγορία ?
 - Διαθέσιμα χρώματα
 - Ετικέτα (Tag : Best Seller, New, etc.) ?
 - Ειδικό κείμενο που αφορά μέλη prime

Product Card



ADIDAS.COM

Product Page / Details:

- Product Image
- Product Details:
 - Κατηγορία Προϊόντος
 - Κρητικές
 - Όνομα Προϊόντος
 - Τιμή Προϊόντος
 - Ετικέτα
 - Περιγραφή
 - Διαθέσιμα χρώματα
 - Μεγέθη

BEST SELLING GIFT

Women's • Originals

★★★★★ [205](#)

SL 72 OG SHOES

\$110

prime options available

★ **Popular** over 73 people have bought this product in the last 24 hours.

Log in to use available adiClub discount vouchers on this item. Other promo codes will not apply.

Dark Brown / Off White / Semi Impact Orange

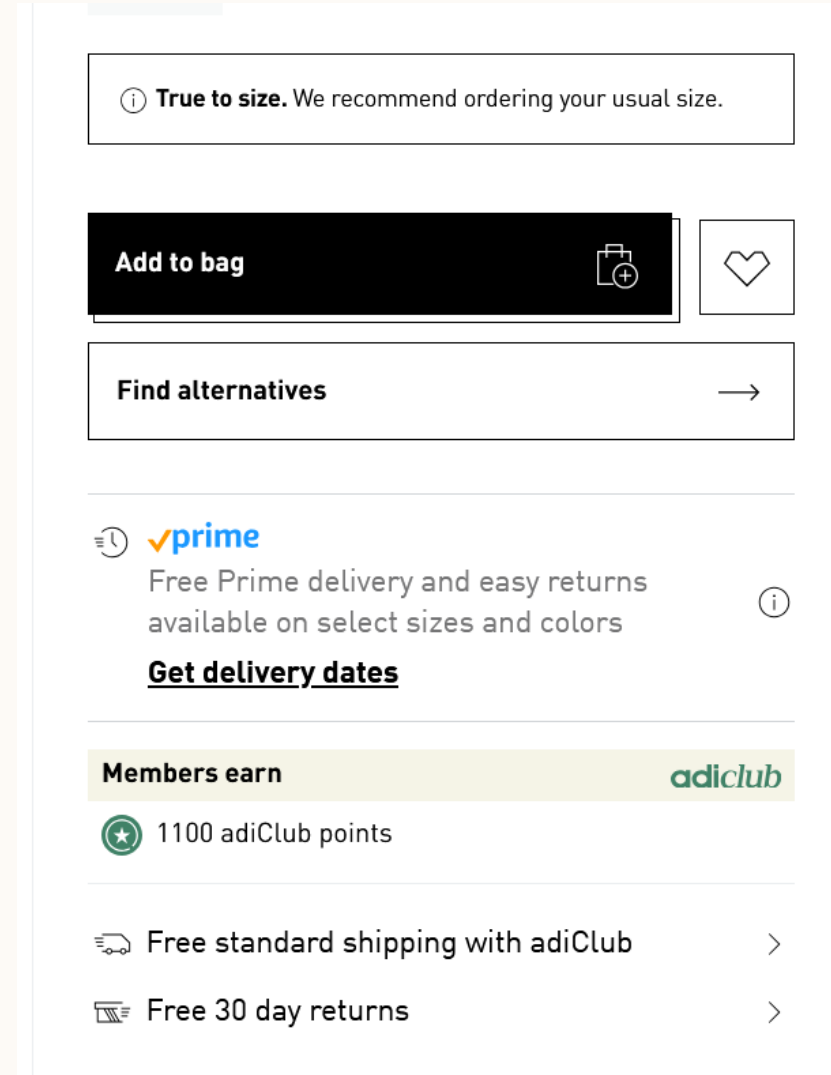
Sizes [Size guide](#)

5	5.5	6	6.5	7
7.5	8	8.5	9	9.5
10	10.5	11	11.5	12
12.5	13	13.5	14	14.5

ADIDAS.COM

Product Page / Details:



- **Add to bag:**
 - Προσθέτει το προϊόν με τις επιλογές (μέγεθος, χρώμα) και τα χαρ/κά του, στο καλάθι του «συνδεδεμένου» χρήστη
- **Wishlist**
 - Προσθήκη του προϊόντος στο **wishlist**
- **Shipping costs**
 - (+ **UX points**) -> Επιτρέπει στον χρήστη να υπολογίσει τα μεταφορικά του με βάση την διεύθυνσή του.
- **Return policy**
 - (+ **UX points**, + **Design points**) Πολύ απλό, θα ισχύει για όλα τα προϊόντα και θα είναι ένα απλό κείμενο



ADIDAS.COM

Παραγγελία:

- Στοιχεία επικοινωνίας:
 - Email
 - Κιν. Τηλ.
- Διεύθυνση:
 - Διεύθυνση παράδοσης
 - Όνομα
 - Επώνυμο
 - Διεύθυνση
 - Διεύθυνση Χρέωσης
- Η Παραγγελία σου:
 - Σύνολο προϊόντων
 - Έξοδα Αποστολής
 - Σύνολικό κόστος (+ Εκτίμηση ΦΠΑ).
 - Λίστα προϊόντων

ΓΕΙΑ ΣΟΥ, GIANNIS!
 adiclub

ΟΛΟΚΛΗΡΩΣΗ ΠΑΡΑΓΓΕΛΙΑΣ


(1 προϊόν) € 70,00


ΕΠΙΚΟΙΝΩΝΙΑ


babarashka2+adiclub2@gmail.com

ΔΙΕΥΘΥΝΣΗ

Διεύθυνση παράδοσης

Όνομα *
John
 

Επώνυμο *
Papaz
 


Πρώτη γραμμή διεύθυνσης *
Αγίου Σπυρίδωνος 17
 

π.χ. Οδός Κολοκοτρώνη 123

Η ΠΑΡΑΓΓΕΛΙΑ ΣΟΥ

1 προϊόν	€ 70,00
Αποστολή	Δωρεάν
Σύνολο	€ 70,00
(Περιλαμβάνει φόρο € 13,55)	

ΕΠΕΞΕΡΓΑΣΙΑ



adidas Adicolor Oversize F...
€ 70

Μέγεθος: XL / Ποσότητα: 1
Χρώμα: Beige / Black

ADIDAS.COM

Λοιπές Πληροφορίες:

- Διεύθυνση παράδοσης (Συνέχεια):
 - Επιπρόσθετες πληροφορίες
 - T.K. (Για Ελλάδα 5 ψηφία)
 - Πόλη
 - Χώρα (Μόνο Ελλάδα προς το παρόν?)
 - Τηλέφωνο [προαιρετικά?]
 - (+ UX Points) Αποθήκευση διεύθυνσης και στοιχείων επικοινωνίας
 - Στοιχεία χρεώσης και αποστολής είναι ίδια
- Στοιχεία Χρεώσης
 - Ον/μο
 - Διεύθυνση
 - Email
 - Κιν. Τηλ.

Πρώτη γραμμή διεύθυνσης *

Agiou Spiridonos 17

✓

π.χ. Οδός Κολοκοτρώνη 123

Επιπρόσθετες πληροφορίες

π.χ. Εταιρία, Διαμέρισμα, Κτήριο, κτλ.

Ταχυδρομικός Κώδικας *

10451

✓

π.χ. 55555

Πόλη *

Athens

✓

π.χ. Πάτρα

Χώρα: Ελλάδα

Τηλεφωνο (προαιρετικά)

Θα σε καλέσουμε μόνο σε περίπτωση ερωτήσεων σχετικά με την παραγγελία σου.

☒ Αποθήκευση διεύθυνσης και στοιχείων επικοινωνίας για επόμενες παραγγελίες

☒ Τα στοιχεία χρέωσης και αποστολής είναι ίδια.

A.Φ.Μ.

ADIDAS.COM

Επιλογές Παράδοσης:

- Λίστα με διαθέσιμες επιλογές (για το συγκ. Προϊόν)

ΕΠΙΛΟΓΕΣ ΠΑΡΑΔΟΣΗΣ

Απλή Αποστολή

Δωρεάν

Δευτέρα, 10 Νοε.
08:00 – 18:00

ACS

ή παράλαβε την παραγγελία σου από σημείο παραλαβής

Κατάστημα adidas

Δωρεάν

Παραλαβή από τις Σάββατο, 1 Νοε.

adidas Outlet Store Larissa, Larissa
4km K. Karamanlis Avenue 1
(79,6 km)

ΕΠΟΜΕΝΟ



ADIDAS.COM

Λοιπές επιλογές & Δυνατότητες:

- **Product card:**
 - Εικόνα προϊόντος
 - Τιμή (ειδική τιμή σε περίπτωση προσφοράς, εδώ έχω μπει ως μέλος και εμφανίζονται οι ειδικές εκπτώσεις μέλους)
 - Όνομα προϊόντος
 - Κατηγορία ?
 - Διαθέσιμα χρώματα
 - Ετικέτα (Tag : Best Seller, New, etc.) ?
 - Ειδικό κείμενο που αφορά μέλη prime

FEED
ΠΑΡΑΓΓΕΛΙΕΣ
ΛΟΓΑΡΙΑΣΜΟΣ

ΣΥΝΟΨΗ ΛΟΓΑΡΙΑΣΜΟΥ

Προσωπικές Πληροφορίες

Βιβλίο Διευθύνσεων

Προτιμήσεις

adiClub pass

Αποσύνδεση

ΤΑ ΣΤΟΙΧΕΙΑ ΜΟΥ

Μπορείς να επεξεργαστείς τα προσωπικά σου στοιχεία παρακάτω, για να κρατάς ενημερωμένο τον λογαριασμό σου.

ΣΤΟΙΧΕΙΑ

GIANNIS PAPAS

2002-02-02

ΑΛΛΟ

ΕΠΕΞΕΡΓΑΣΙΑ

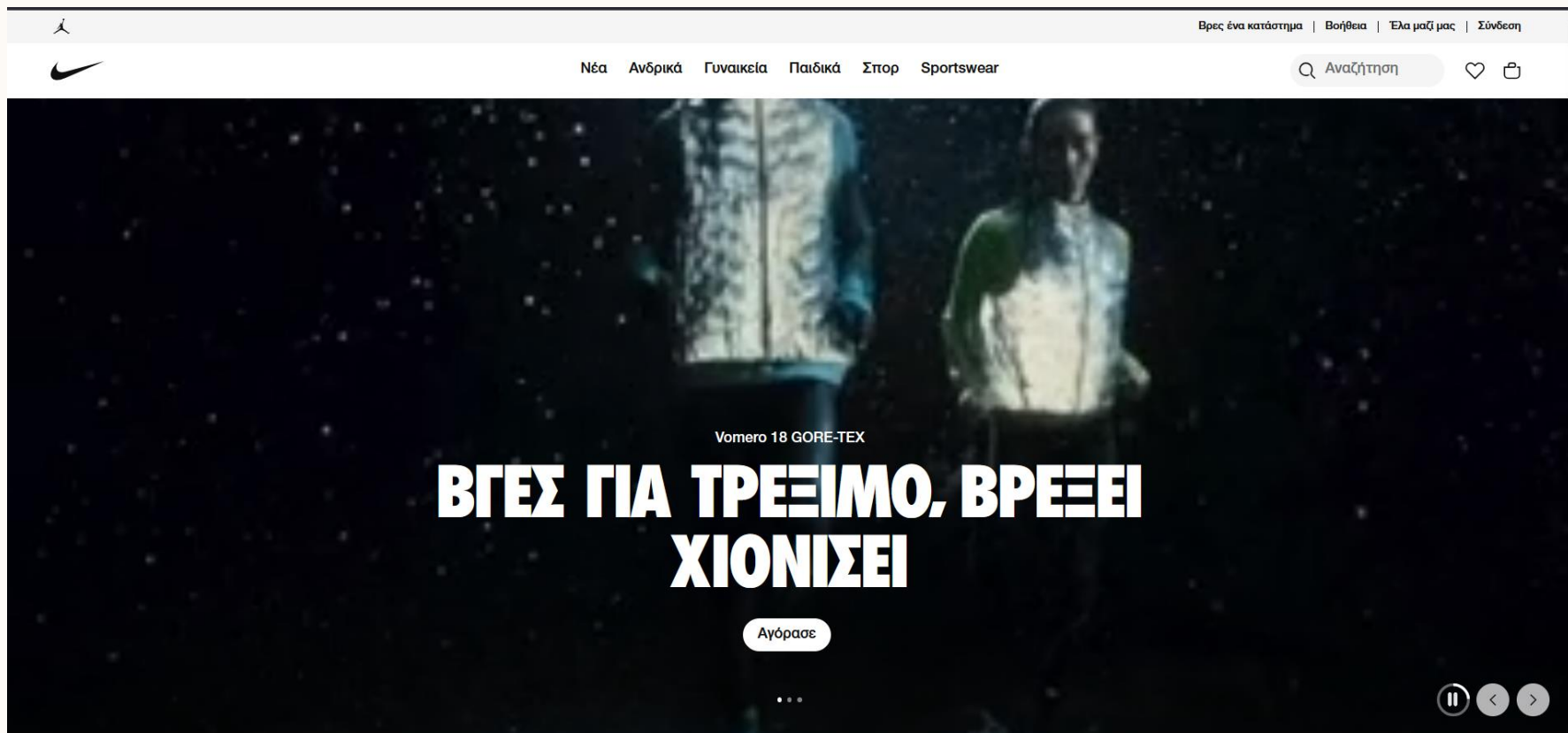
ΣΤΟΙΧΕΙΑ ΣΥΝΔΕΣΗΣ

EMAIL

BABARASHKA2+ADICLUB2@GMAIL.COM

ΕΠΕΞΕΡΓΑΣΙΑ

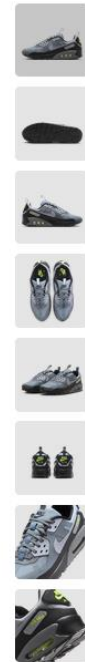
NIKE.COM



NIKE.COM - ΠΑΠΟΥΤΣΙΑ

Όπως και στο adidas, τα κοινά χαρ/κά είναι:

- Όνομα
- Τύπος / Κατηγορία (Ανδρικό παπούτσι)
- Τιμή
- Χρώμα / Σχέδιο
- Μέγεθος
- Περιγραφή



Nike Air Max 90

Ανδρικά παπούτσια

149,99 €



Επιλογή μεγέθους

Οδηγός μεγεθών

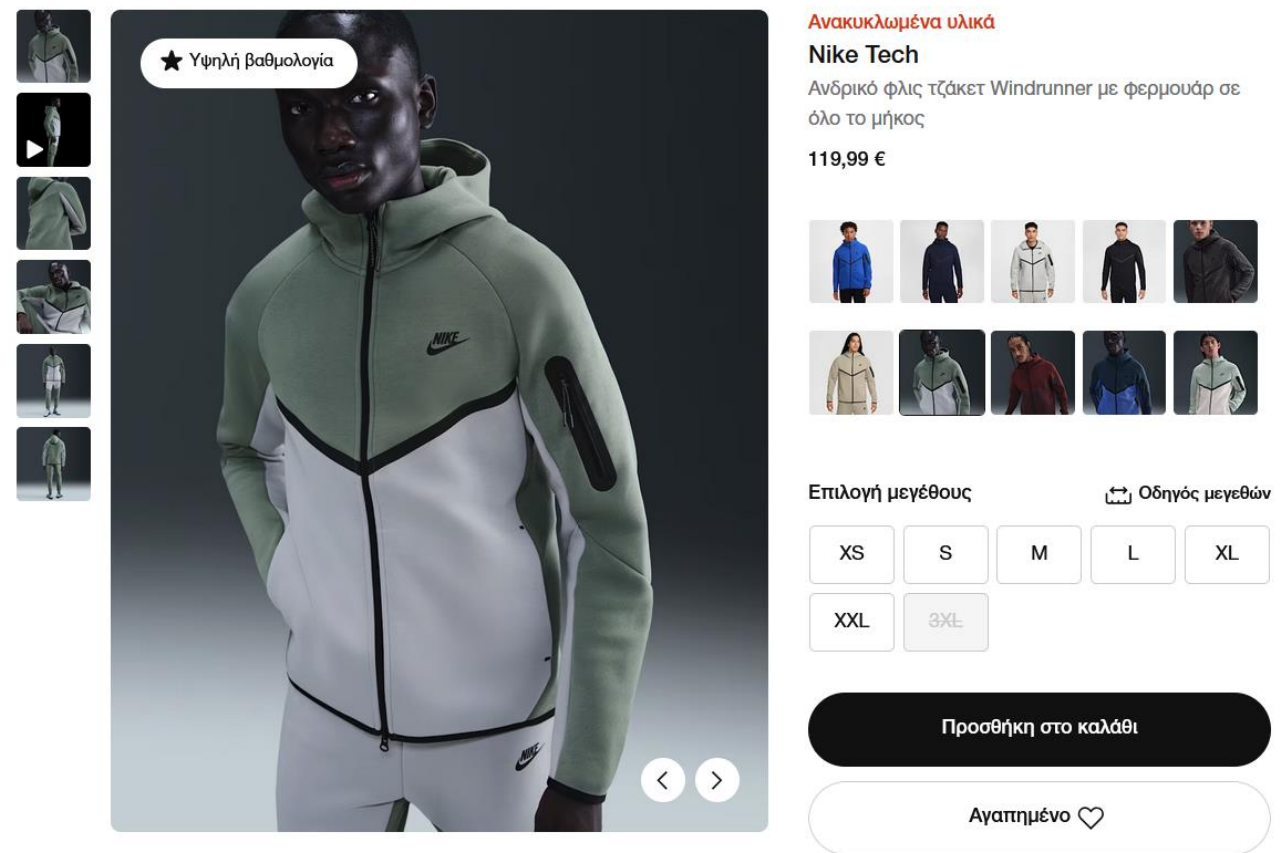
EU 38.5	EU 39	EU 40
EU 40.5	EU 41	EU 42
EU 42.5	EU 43	EU 44
EU 44.5	EU 45	EU 45.5
EU 46	EU 47	EU 47.5
EU 48.5	EU 49.5	

Προσθήκη στο καλάθι

NIKE.COM - ΡΟΥΧΑ

Όπως και στο adidas, τα κοινά χαρ/κά είναι:

- Όνομα
- Τύπος / Κατηγορία (Ανδρικό φλις τζάκετ ...)
- Τιμή
- Χρώμα / Σχέδιο
- Μέγεθος (άλλη μονάδα για τα ρούχα)
- Περιγραφή



★ Υψηλή βαθμολογία

Ανακυκλωμένα υλικά

Nike Tech

Ανδρικό φλις τζάκετ Windrunner με φερμουάρ σε όλο το μήκος

119,99 €

Επιλογή μεγέθους

XS S M L XL

XXL 3XL

Προσθήκη στο καλάθι

Αγαπημένο

ΣΥΜΠΕΡΑΣΜΑΤΑ

Γενικά μπορούμε να συμπεράσουμε τα εξής:

Για το στυλ και την πλατφόρμα:

- Γενικά ακολουθείται ένα απλό και μοντέρνο “flat” στυλ.
- Καρτέλες προϊόντων με τα στοιχεία τους σε **grid layout**
- Υποστηρίζουν λογαριασμούς και προτιμήσεις

Για τα χαρ/κά των προϊόντων και τις οντότητες:

- Τα προϊόντα κάθε τύπου έχουν τα εξής χαρ/κά:
 - **Εμπορικό όνομα**
 - **Τύπος** (για Άνδρα / Γυναίκα / Παιδί)
 - **Κατηγορία** (Ρούχο, Παπούτσι, εξαρτήματα, κτλπ)
 - **Τιμή**
 - **Κωδ. Προϊόντος**
 - **Περιγραφή**
 - **Απόθεμα**

ΣΥΜΠΕΡΑΣΜΑΤΑ

Γενικά μπορούμε να συμπεράσουμε τα εξής:

Για το στυλ και την πλατφόρμα:

- Γενικά ακολουθείται ένα απλό και μοντέρνο “flat” στυλ.
- Καρτέλες προϊόντων με τα στοιχεία τους σε **grid layout**
- Υποστηρίζουν λογαριασμούς και προτιμήσεις

Για τα χαρ/κά των προϊόντων και τις οντότητες:

- Ο Επισκέπτων Χρήστης
 - Έχει πρόσβαση σε όλα τα προϊόντα
 - Μπορεί να τα προσθέσει στο «καλάθι αγορών»
 - Μπορεί να δημιουργήσει λογ/σμό
- Ο εγγεγραμμένος Χρήστης
 - Έχει πρόσβαση σε όλα τα προϊόντα
 - Μπορεί να τα προσθέσει στο «καλάθι αγορών»
 - Μπορεί να παραγγείλει όσα είναι στο καλάθι του.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Γενικά μπορούμε να συμπεράσουμε τα εξής:

Για το στυλ και την πλατφόρμα:

- Γενικά ακολουθείται ένα απλό και μοντέρνο “flat” στυλ.
- Καρτέλες προϊόντων με τα στοιχεία τους σε **grid layout**
- Υποστηρίζουν λογαριασμούς και προτιμήσεις

Για τα χαρ/κά των προϊόντων και τις οντότητες:

- Ο εγγεγραμμένος Χρήστης
 - Έχει πρόσβαση σε όλα τα προϊόντα
 - Μπορεί να τα προσθέσει στο «καλάθι αγορών»
 - Μπορεί να παραγγείλει όσα είναι στο καλάθι του.
 - Μπορεί να διαχειριστεί τον λογαριασμό του
 - Μπορεί να προσθέσει προϊόντα στα αγαπημένα του ή στο wishlist

ΣΥΜΠΕΡΑΣΜΑ - ER DIAGRAM

Άρα , μπορούμε ορίσουμε τις βασικές οντότητες της πλατφόρμας

LINKS

- **Spring Boot Tutorials**
 - **Spring Boot E-Commerce App:** <https://www.youtube.com/watch?v=m559BxR30ls>
 - Spring Boot (General – Freecodeamp) : [Spring Boot & Spring Data JPA – Complete Course](#)
 - **Spring Boot (GeeksforGeeks):** <https://www.geeksforgeeks.org/advance-java/spring-boot/>
 - **Spring Initializer (For project initial setup):** <https://start.spring.io/>
 - Authentication / Authorization in Spring Boot: <https://www.geeksforgeeks.org/advance-java/....>
 - JWT Auth Spring Boot: <https://www.geeksforgeeks.org/springboot/...>

LINKS

- **React Tutorials**
 - **GeeksforGeeks (Self-Paced):** <https://www.geeksforgeeks.org/reactjs/react/>
- **Docker Tutorials**
 - **GeeksforGeeks :** <https://www.geeksforgeeks.org/devops/docker-tutorial/>
 - **Spring boot with docker:** <https://spring.io/guides/gs/spring-boot-docker>
 - **Docker Tutorial:** <https://youtu.be/3c-iBn73dDE>
- **General (Code, etc.)**
 - **Programming Principles:** <https://learn.microsoft.com/en-us/.../architectural-principles>



THANK YOU

Valentino Velchev

[Your phone]

[Your email]