User signs in using (Button: "**Sign in")** email or username + password, the password should be stored in our database (hashed and salted) you can use JWT to help you in keep memorizing the *session id* while our user is still signed in (*same session*).

If the user signs in, then they have access to **the store + mailbox** + **list of orders**

The store shows two different sections for types of items, one product and packs, each product holds a **type** variable to decide to which section it belongs, plus the **image path, description and title, and the price.**

When clicking **'Buy now'**, the page should load the specific chosen **product (title, description, image, price)**, with a **quantity (input_type_number)**, under these details, list of products should be shown as suggestions.

When clicking **"Order",** the page should save **the product id, the user's name, his phone number and his email, and the quantity** to the orders table in our date base. It should also send an **email** to his mailbox to let him know that he has ordered a product and will be contacted soon.

The mail should require a button to **cancel** this order, when clicked, the user is sent to his list of orders, where he can click a **'Cancel'** button that allows us to make the order as **'canceled'**

Admin can log in to his dashboard (using his email+password)

He can CRUD **users** (*id/username, Role, name, email, phone number, sector, date of adhesion*), **events** (*id, title, description, date, link*), **news** (*id, title, path of image, description*), **products** (*id, title, description, path of image, type [item, pack], inStore [true, false]*), **orders** (*id, #id_username, #id_product, date, status [New, Confirmed, Pending, shipped, Cancelled]*)

The database should contain the tables suggested above;