PROJECTE UPC (1A PART)

Comenceu a crear la vostra web per a publicitar el joc dels alumnes de la UPC.

Descripció dels elements que han d'intervenir:

(1 punt)
Qui som?: component que mostra una descripció d'aquest projecte conjunt entre el nostre centre i la UPC. Cal també que apareguin les imatges ***imatge1.jpg, logoProven.png, logoUpc.png***

Equip: component que mostra una imatge de vosaltres i els vostres companys de grup de la UPC, i una breu descripció (nom, estudis, interessos,...)
Feu servir, si no teniu imatges, els avatars **avatar1.jpg** i **avatar2.jpg**.

(1 punt)
Registre: component que deixa enregistrar-se al vostre projecte i poder entrar en una zona privada. Cal que reutilitzis la component creada als exemples de classe.
Aquest cop els camps necessaris per fer el registre són:
- *Nom d'usuari*: validareu que sigui obligatori, ha de contenir lletres i/o números, mínim de 5 caràcters i un màxim de 8.
- *Correu electrònic*: validareu que sigui un de vàlid, camp obligatori.
- *Repetir correu electrònic*: camp obligatori i ha de coincidir amb el d'email.
- *Data de naixement:* camp obligatori, que sigui una data correcta, que no sigui del futur i menor a 18 anys enrere.
- *Edat*: camp de només de lectura, es calcularà a partir de la data de naixement. No cal validació.
- T'agrada aquest joc?: de tipus radio, amb 3 possibles respostes (Sí, No, NS/NC, és obligatori de contestar.

Cal que feu servir un formulari reactiu. Caldrà també que valideu els camps i doneu resposta a tots els possibles errors que s'hi puguin produir mitjançant un missatge.

Has de fer servir 2 directives. Una d'elles ha de ser diferent de les utilitzades a classe.

Si falla la validació d'algun camp, no es deixa avançar (botó deshabilitat).
Si tot és correcte, es crea un usuari fent servir una classe, **User**, creada prèviament. A més, a sota del formulari apareix un missatge de salutació cap al nou usuari. Aquest es ficarà es majúscules fent servir una pipe.

(0,5 punts)
Login: component que ens deixar validar fent servir un servei. Es farà que el nom d'usuari i la contrasenya (correu electrònic) passin a un servei, prèviament creat i de nom **UserService**, que comparà aquests dos valors amb els usuaris enregistrats en un array de 10 usuaris fixos de tipus **User**. Si no hi ha coincidència, s'informa a la component i es fica un missatge d'error sota del formulari.
Podeu reutilitzar l'exemple de classe però tingueu en compte que aquí el servei no crea usuaris a l'atzar sinó que els té ja emmagatzemats en un array.

(1 punt)
Menú: component que ha de contenir totes les rutes a les diferents components fent servir el routing d'Angular. A més aquest menú reacciona a si estem o no loguejats a partir d'una localstorage. Aquesta la crearem només quan el login doni coincidència correcta. D'aquesta manera, si fem login i aquest és correcte, no apareixen al menú ni l'opció de login ni la de registre i sí que hi apareixen Merchandising, Compra i Logout.

(0,5 punts)
Logout: component que no conté res. El seu objectiu és el d'esborrar la localstorage

(1 punt)
Merchandising: component que llista 4 productes a la venda relacionades amb el vostre joc. Fes servir les imatges que us passem. Aquests productes estan emmagatzemats a la part .ts de la pròpia component. Cada ítem és un objecte de la classe **Producte**. Les propietats d'aquesta classe són: nom de la imatge, nom del producte, descripció, preu, disponibilitat màxima.
Al costat de cada producte del llistat tindrem un botó per poder **Comprar** i de nom «**Afegir producte**».

(2 punts)
Compra:
Volem que cada cop que fem clic en un dels botons d' **Afegir producte,** aquest producte surti afegit dinàmicament en un div dins d'una nova component, la de Compra. Hem de mostrar de la millor manera les característiques del que comprem: nom, preu, descripció, unitats comprades. També remarcar que si fas clic més d'un cop al mateix «producte» han de **pujar** les unitats comprades.
Per poder fer-ho bé, us recomanem que cada cop que cliquem al botó d'afegir a la cistella, hi hagi un array que capturi l'objecte. Just abans d'aquesta captura cal que comprovis si hi és o no ja el producte. Si no hi és, l'afegeixes i la quantitat es fixa en 1. Si sí que hi és, augmentes la quantitat (+1). Un cop manipulat aquest array cal que l'emmagatzemis en una cookie. Aquesta cookie és la que es mostra a la component de Compra.

(1 punt)
- Caldrà comprovar que si afegim moltes vegades un producte, que d'aquest hi ha hagi prou disponibilitat.
- A sota dels productes comprats es veurà el preu total de la compra.
- A sota del preu total, hi haurà un botó de **Compra final** i un altre de **Cancel·lació**. Si cliquem a Compra final, se'ns dirà que la compra ha tingut èxit. Si cliquem a l'altre, s'eborrarà la cookie sencera i desapareixeran els productes de la cistella.

(0,5 punts)
Documentació de tot el codi

(0,5 punts)
Estructura correcta en carpetes

(1 punt)
Organització del codi (refactorització)

ATENCIÓ: Si vols redireccionar en Angular cal que injectis el servei Router a la component on vols fer el redireccionarment:

```
constructor(private router: Router) {}
```

i al lloc on vulguis fer efectiu el redireccionament escriguis

this.route.navigate('/rutaReconegudaPerRouting');