

Pràctica 2

Front-End de una página d'articles

Sistemes Oberts

KEVIN SÁNCHEZ RAMÍREZ
IULIAN SEBASTIAN OPREA

Índex

Índex.....	2
Introducció.....	3
Estructura de la pràctica.....	3
Controller	3
Model	4
Service	4
Decisions de disseny.....	5
Jocs de proves realitzats	6
Conclusions	30
Manual d'instal·lació.....	31

Introducció

Aquesta pràctica podem dir que es la segona fase de la anterior, es a dir, aquesta part el que fa es millor el que teníem creant un front-end que utilitza la api rest que havíem fet a la practica 1. On per definir breument es una interfície per a l'usuari pugui interactuar i entendre més fàcilment la pagina web que estem dissenyant , en tot cas seria com es veu la nostra pagina web per a l'usuari.

Estructura de la pràctica

La practica segueix el patró de disseny MVC (Model View Controller) una vegada sabem lo que es aquest patró podem entendre millor la estructura. La estructura es la següent:

Package java.deim.urv.cat.homework2 :

Controller

ArticleController : es el controlador que gestiona tot el que te a veure amb els articles i de quina forma interactua el usuari amb les diferents seccions dels articles

ArticleForm : es la classe que valida el paràmetres introduïts per l'usuari però també per a que nosaltres pugéssim a enviar al back-end la informació necessària per a poder crear el article corresponent.

FilterForm : es un classe la qual la fem servir per saber que tenim que enviar per el query param quan el usuari vol filtra el articles que veu per pantalla

LogInController : Es el controlador que gestiona el inici de sessió per part de l'usuari.

LogInForm : conte els diferents valors que en passa el usuari per poder iniciar sessió i que el controlador LogInController pugui validar llavors pugui accedir a totes les parts permeses que hi te.

CustomerController: aquest és el controlador que gestiona les peticions cap a les vistes on interacciona en les dades dels usuaris.

CustomerForm: Bean auxiliar que ens permet tindre les dades que han sigut omplertes per l'usuari a l'hora d'editar el seu perfil, ens permet tindre una gestió automàtica dels camps gràcies a les seves anotacions.

Model

ArticleGQ : es la classe que representa una de les figures que pot tenir la nostra pagina web, la necessitem per a poder interactuar amb el back-end de forma correcta i poguéssim interactuar amb la informació que ens retorna el back-end.

ArticleDTO : segueix la mateixa idea que la classe de dalt però en aquest per a representar una figura diferent en relació a les altres coses es igual no hi canvia res en aquell sentit.

Credentials: és la classe que ens representa les credencials de cada usuari.

Customer: classe que conte tota la informació d'un usuari, aquest és el mateix format que ens retorna el back-end quan volem obtindre dades més detallades d'un usuari.

CustomerDTO: classe que ens permet una representació més ràpida i breu de cada usuari, es fa servir per mostrar el llistat d'usuaris que tenim registrats.

Links: classe auxiliar dels customers on es guarda l'enllaç de l'últim article publicat, si es que es tracta d'un autor.

Service

ArticleService : interfície que defineix els mètodes necessàries que ha tenir la classe que interactua amb el back-end.

ArticleServiceImpl : implementa la anterior interfície i es la classe que ens interactua amb el back-end, es a dir, per a que nosaltres poguéssim fer alguna cosa amb el back-end tenim que demanar a esta classe que es connecta amb la api i fa les pertinents peticions que nosaltres des de el controlador hem demanat.

CustomerService: interfície que defineix els mètodes necessaris que s'han d'implementar per poder comunicar el front-end amb el back-end. Ens permet tindre més flexibilitat i no haver de cridar a una implementació específica, d'aquesta forma podríem tindre altres implementacions i només fer servir la més adequada.

CustomerServiceImpl: implementa la interfície anterior, i és la classe encarregada d'interactuar amb el back-end, això ho aconseguim realitzant peticions i enviant els resultats al front-end, per tal de mantindre una correcta sincronització entre dues bandes.

Decisions de disseny

1. Hem afegit un mètode al back-end el qual ens servirà per fer la validació de si pot iniciar sessió o si li hem de retornar que hi hagut un error.
2. Hem modificat les classes ArticleDTO i ArticleGQ per a poder mostra coses que en la primera fase no s'han observat que es necessiten per a poder fer un bon funcionament per al usuari, hem afegit el id i la privacitat del article corresponent.
3. Hem decidit que el que es mostra primerament sigui tots el articles sense cap filtre afegit per defecte.
4. Per fer el filtre hem decidit que es fa amb un offfcanvas que es obri a la part esquerra de la pantalla, per a una millor comprensió per part del usuari però també un més fàcil desenvolupament per al programador donat que ja es RWD.
5. Tenim un nav, un footer que estan implementats en la carpeta layout.
6. La vista del filtre esta implementada també en layout per si mes endavant ens fa falta en un altre apartat de la pàgina web.
7. Hem utilitzat card per mostrar tots els articles en la pàgina inicial.
8. No es poden crear nous usuaris per a una simplicitat de cara a la pràctica.

Jocs de proves realitzats

Per poder veure que tot funciona segons el esperat hem generat aquest joc de proves el quan prova els extremes de la nostra pagina web.

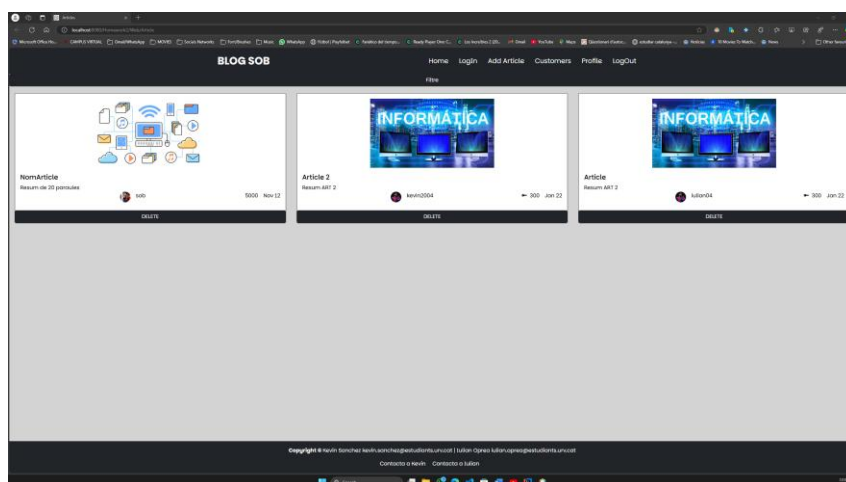
Per assegurar-nos que el nostre projecte funciona correctament en tots els casos possibles, hem realitzat les següents proves:

1. **Proves unitàries:** Inicialment, hem fet proves unitàries de manera individual per a cada component del sistema. Això ens ha permès verificar que cada funció o mòdul del projecte funcioni correctament de manera aïllada, sense dependre d'altres parts del sistema.
2. **Proves d'integració:** Un cop hem provat els components individualment, hem realitzat proves d'integració per comprovar que el conjunt del sistema funcioni de manera correcta. Hem validat que les diferents parts del projecte interaccionin de manera adequada i que no es produeixin errors quan es combinen tots els components.

Cas 1

Validar que ens mostri la informació de la figura numero 1 i que els resultat sigui tots els articles que hi tenim a la base de dades. Amb el seu respectiu disseny que hi te assignat en la vista especificada.

Sortida real:

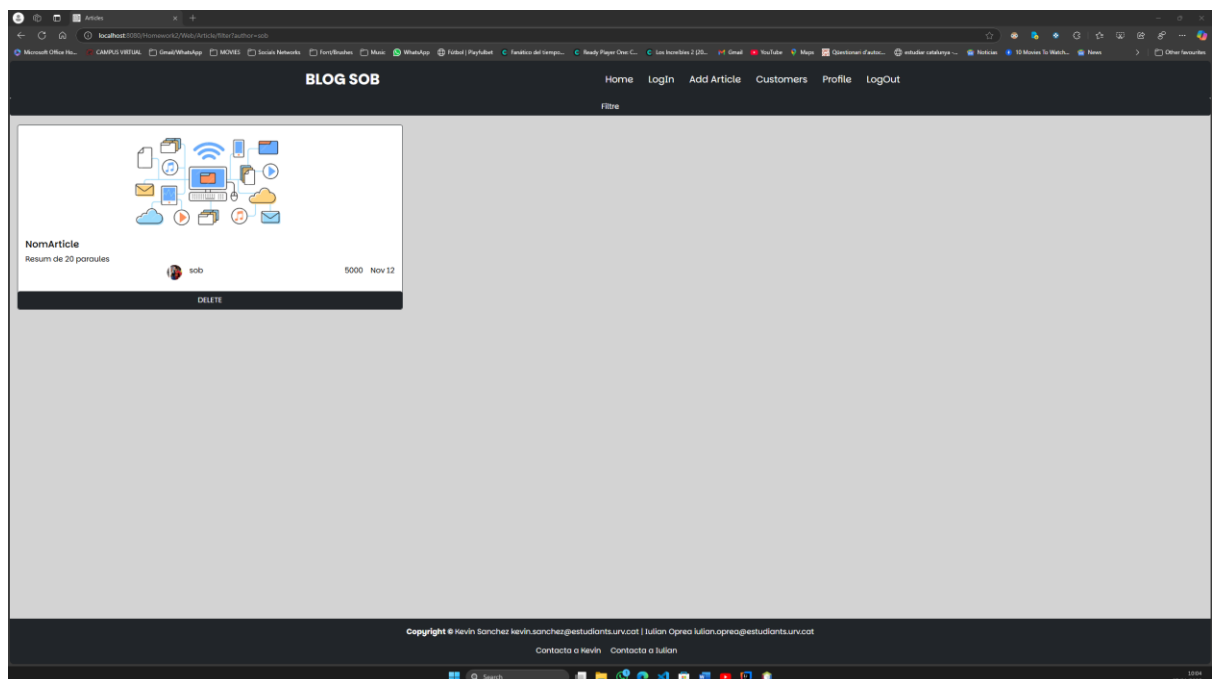
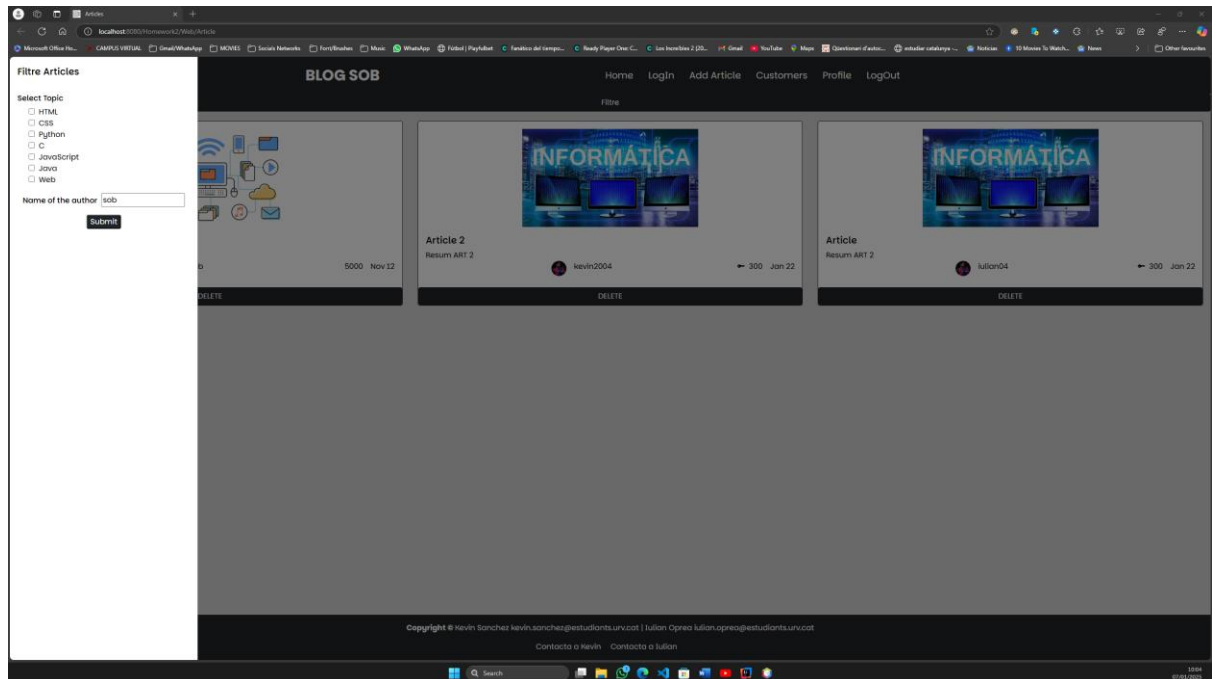


Correcte : SI

Cas 2

Validar que ens mostri la informació de la figura numero 1 i que el resultat només sigui d'aquells articles que tenen aquell autor especificat per QueryParam.

Sortida real:

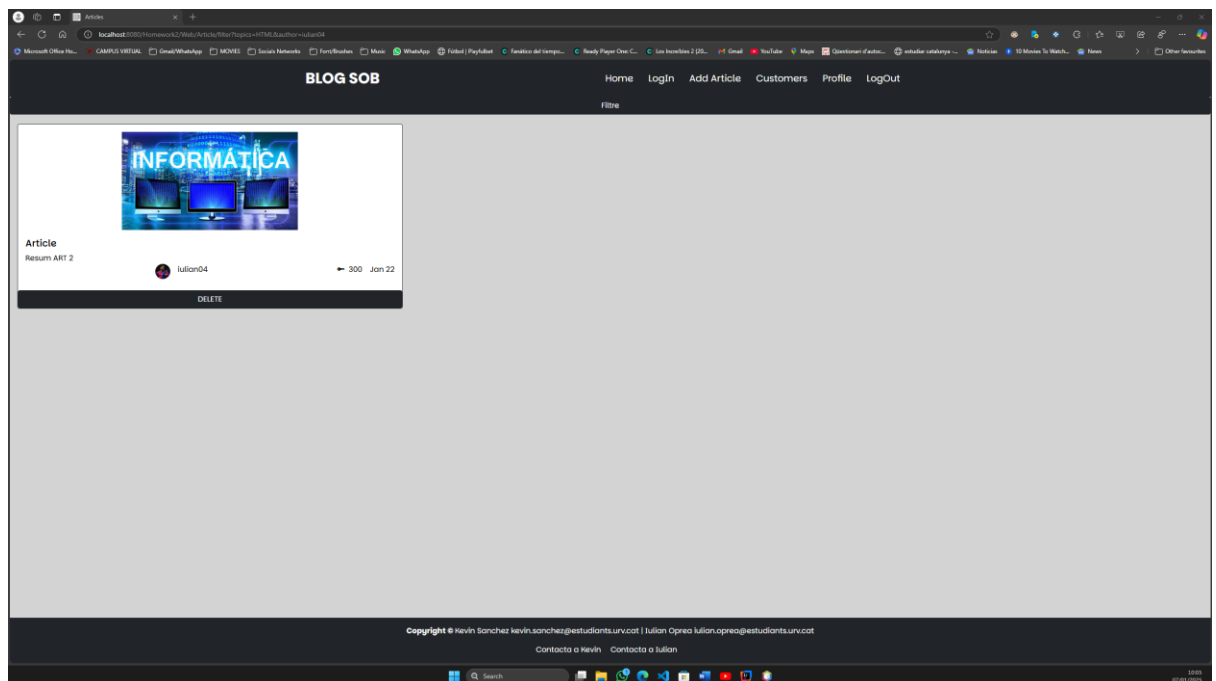
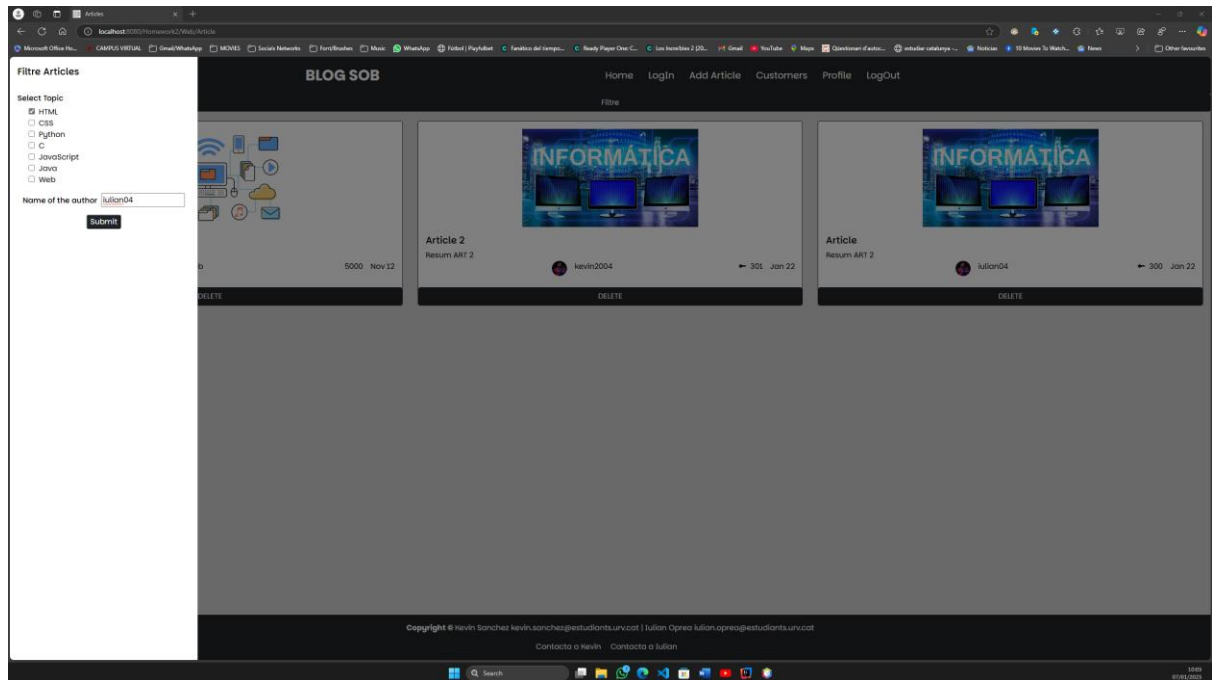


Correcte : SI

Cas 3

Validar que ens mostri la informació de la figura numero 1 i que la nostra api pot admetre per paràmetre un tòpic i un autor.

Sortida esperada:

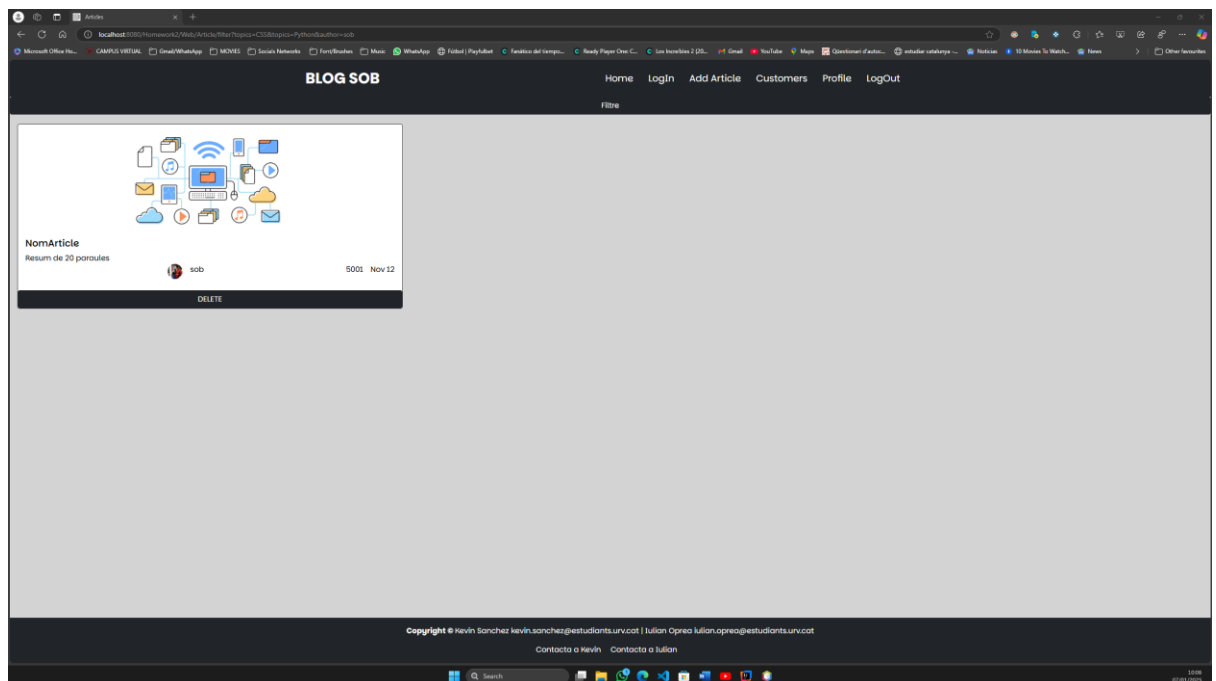
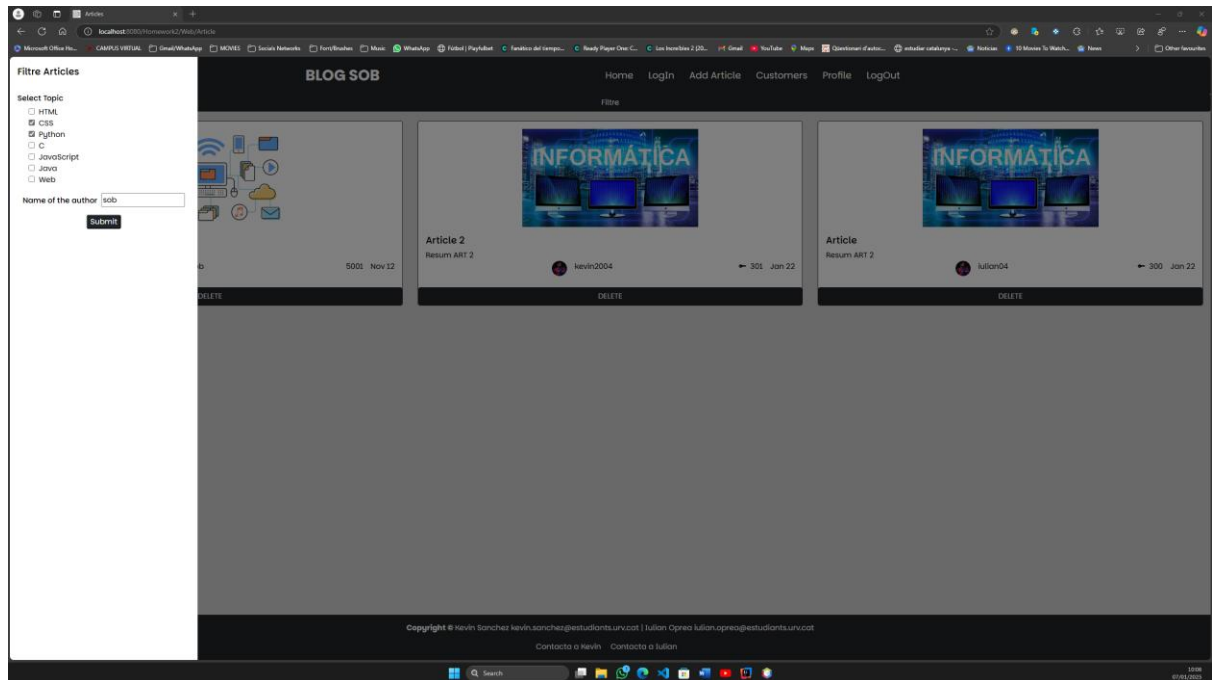


Correcte : SI

Cas 4

Validar que ens mostri la informació de la figura numero 1 i que la nostra api pot admetre per paràmetre dos tòpics i un autor.

Sortida Real:

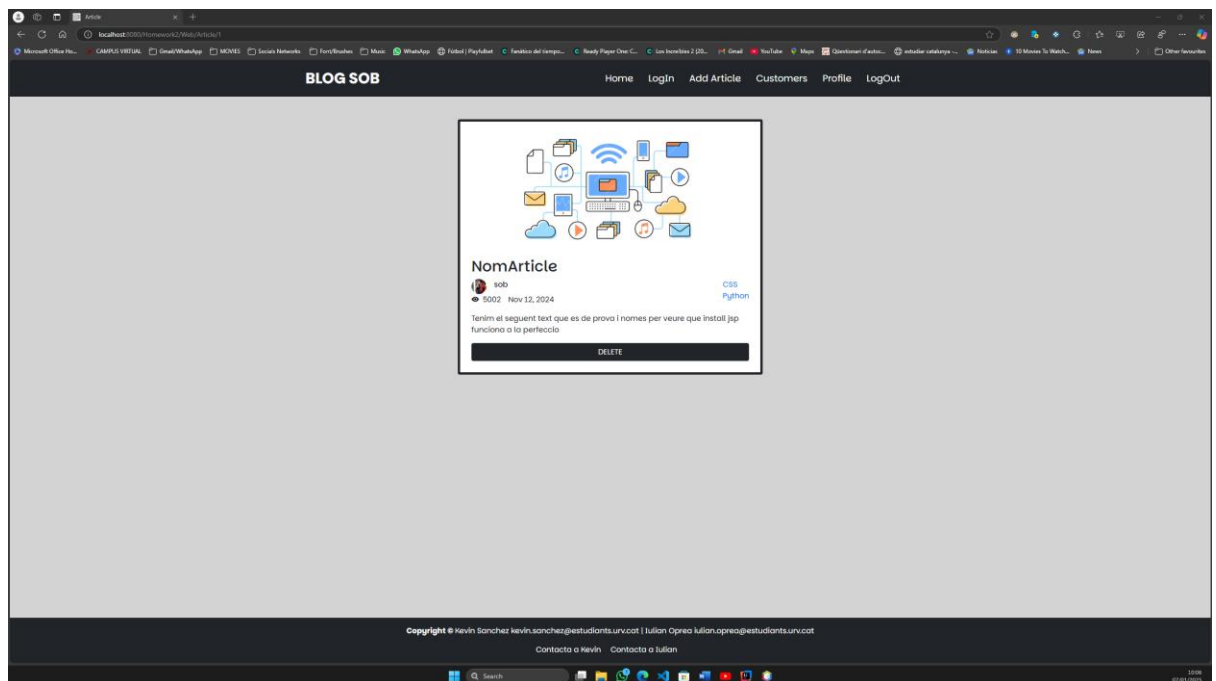
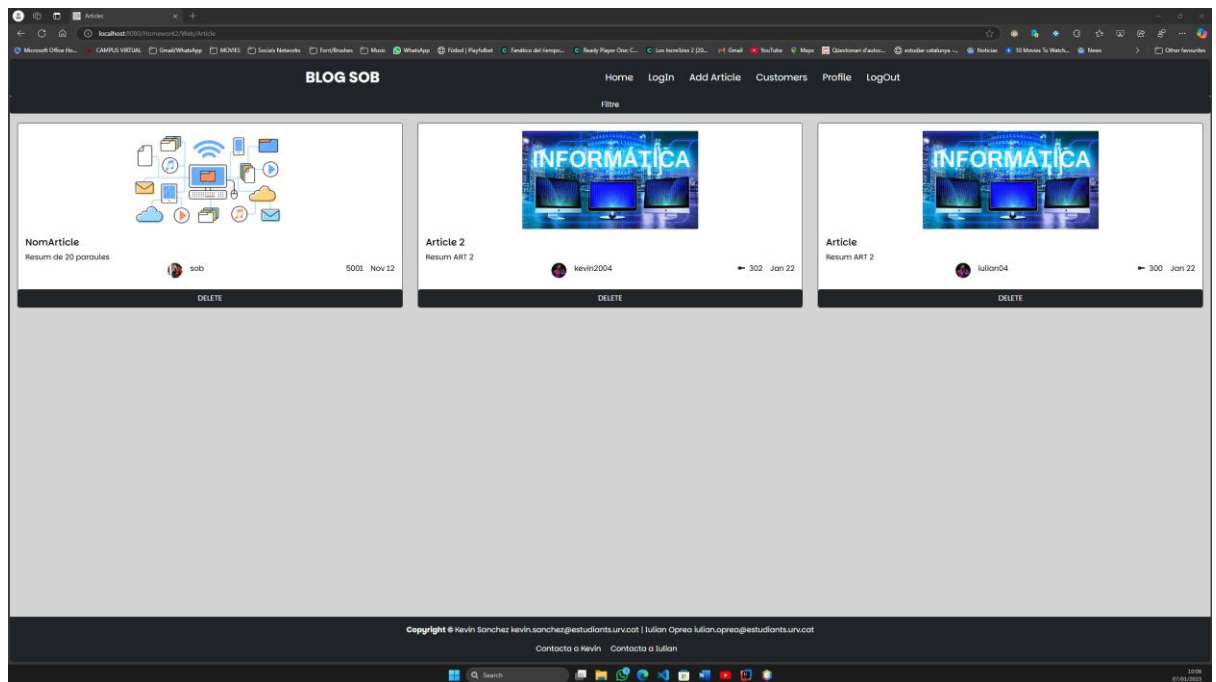


Correcte : SI

Cas 6

Volem veure la informació de la figura 2 i que podem veure el article si es públic encara de no estar registrat i que també augmenta el nombre de visualitzacions

Sortida Real:

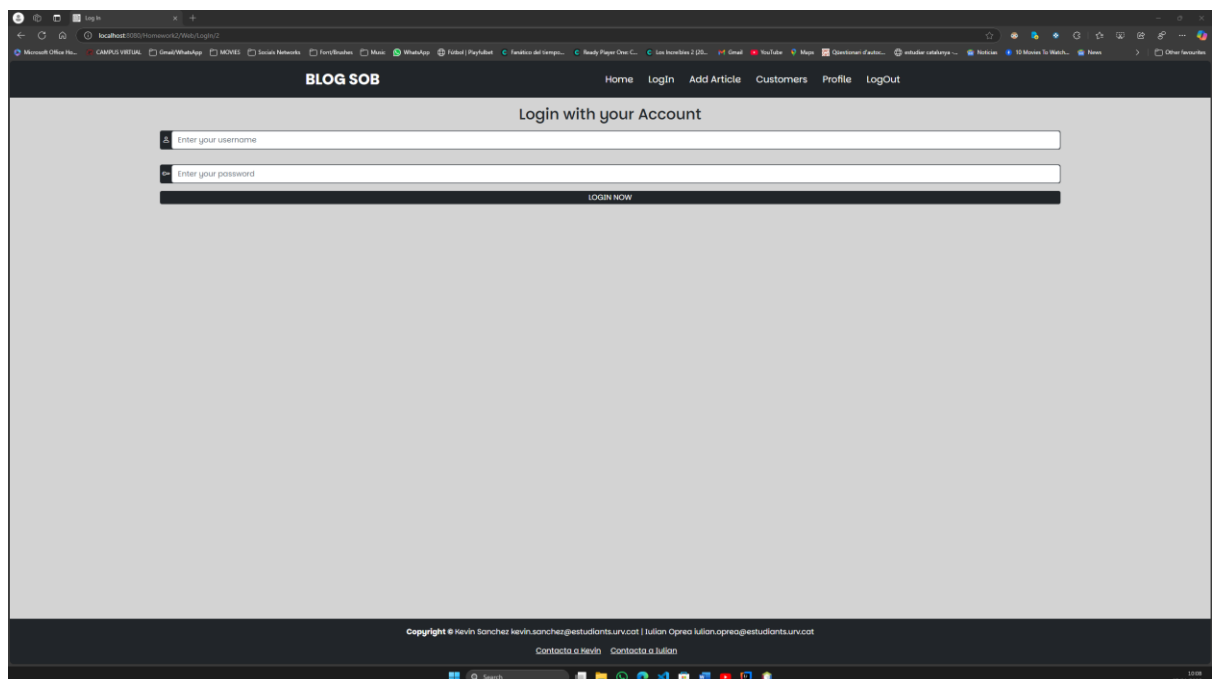
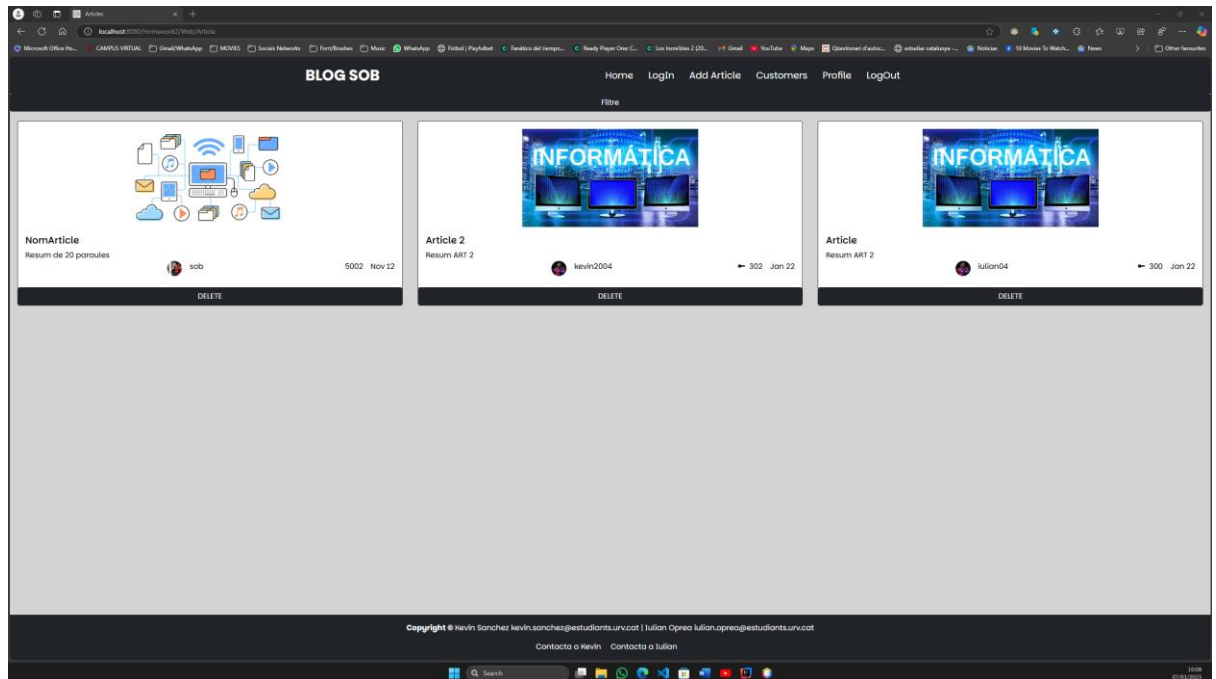


Correcte : SI

Cas 7

Volem veure la informació de la figura 2 i que no podem veure el article si es privat perquè no estem registrat i ens redirigeix a el Login donat que tenim que esta registrat per poder veure el article.

Sortida Real:

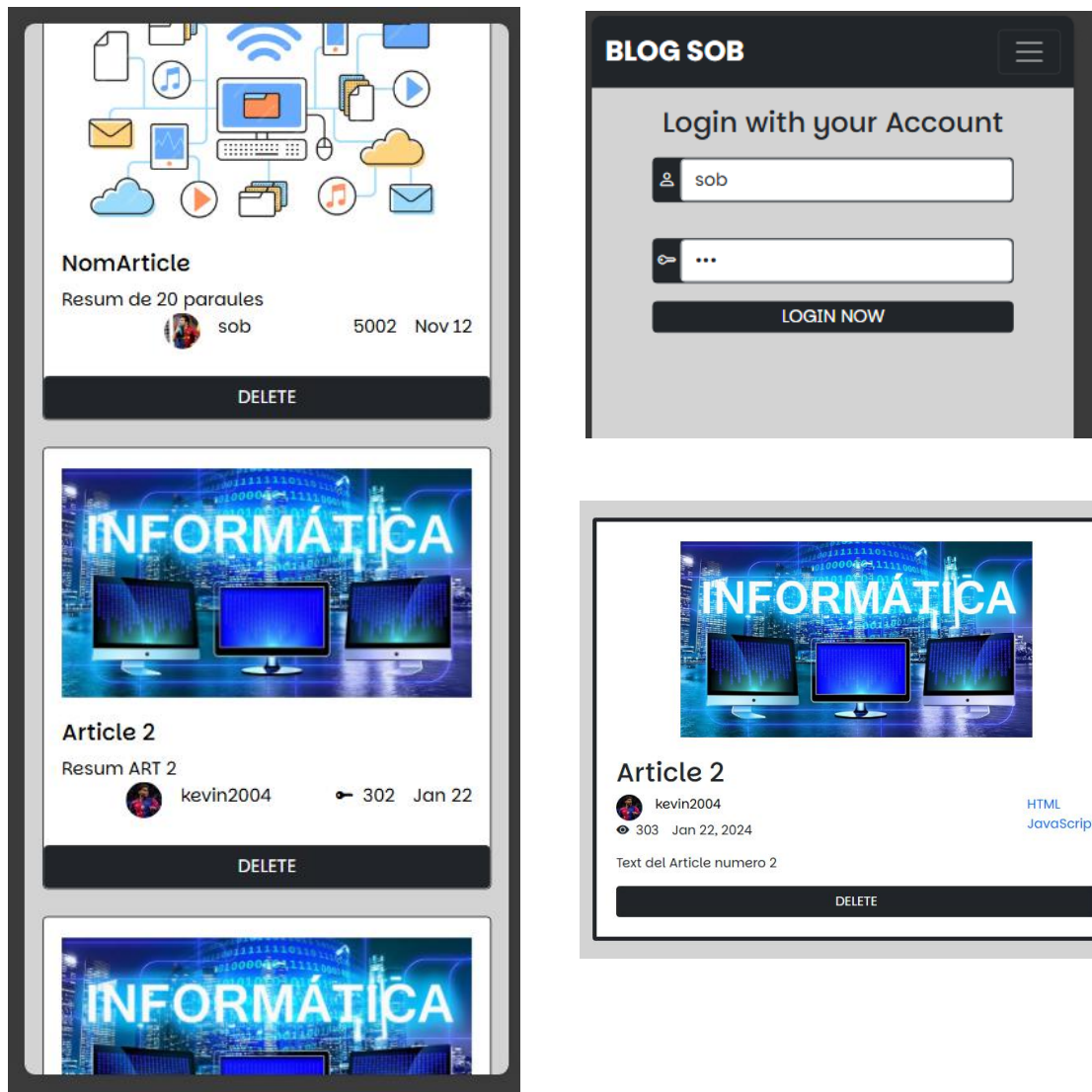


Correcte : Sí

Cas 8

Volem veure la informació de la figura 2 i que podem veure el article si es privat perquè estem registrat i augmenta el numero de visualitzacions

Sortida Real:



Correcte : SI

Cas 9

Volem comprova que al intentar fer el addArticle ens demana primer iniciar sessió si no estem ja iniciats i dins d'aquest apartat introduïm alguna dada malament i ens mostri un missatge de error especificant que es el que hem introduït malament

Sortida Real:

The screenshot shows a web browser window with the URL `localhost:3030/Homeework2/Web/Article/addArticle`. The page title is "BLOG SOB". The navigation bar includes links for Home, Login, Add Article, Customers, Profile, and LogOut. The main content area is titled "Introdueli Un Article" and contains the following form fields:

- Titol:**
- Text:**
- Resum:**
- Imatge:**
- Topics:**
- HTML:**
- Privat:** ☐
- Submit:**

At the bottom of the page, there is a copyright notice: "Copyright © Kevin Sanchez kevin.sanchez@estudiants.urv.cat | Julian Oprea julian.oprea@estudiants.urv.cat" and contact information: "Contacta a Kevin Contacta a Julian".

The screenshot shows the same "BLOG SOB" article form, but with a validation error. The "Resum" field is highlighted with a red border, and a red error message is displayed at the bottom of the form:

resum: Resum must be between 2 and 20 characters

The other fields remain the same as in the previous screenshot.

Correcte : Si

Cas 10

Volem prova que el inserir funciona, llavors fem addArticle de un article que sabem que es vàlid i al torna a la pagina inicial si busquem aquest article especificat en hauria de sortir segons les dades que hem especificat en el formulari que conte aquella vista.

Sortida Real:

Introdueix Un Article

Títol
Bone festes a tothom

Text
Aquest es un possible text sobre el article

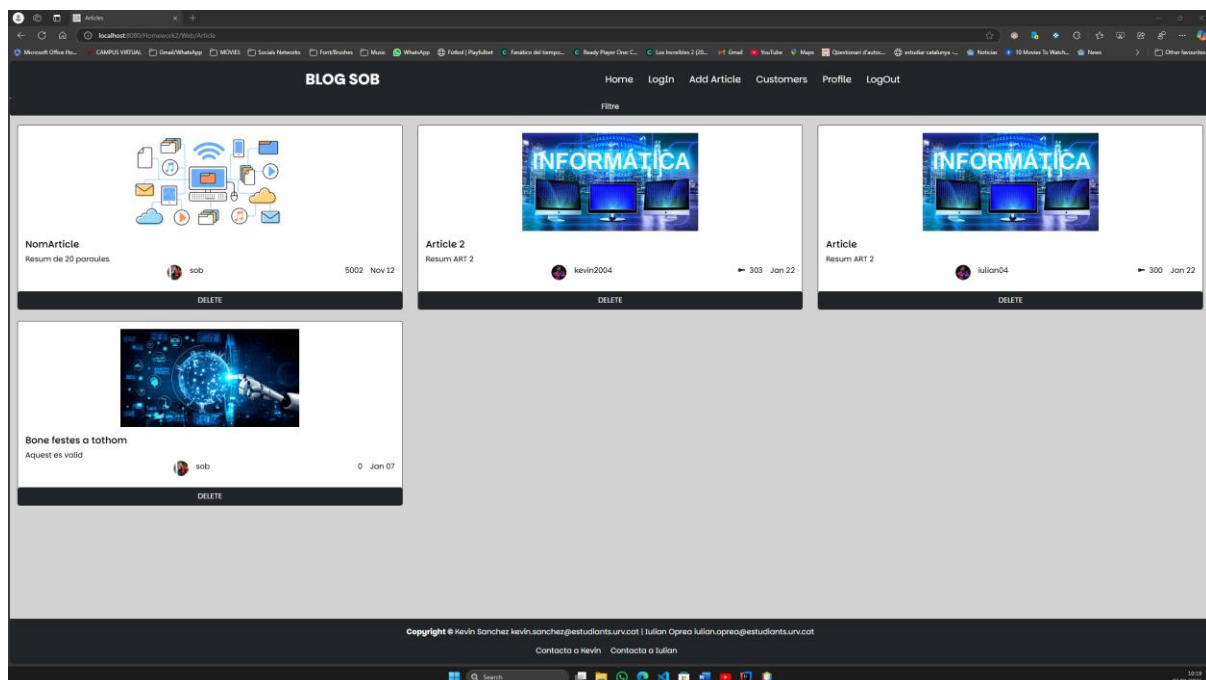
Resum
Aquest es valid

Imatge
https://cdn.pixabay.com/photo/2022/04/04/16/42/technology-7111804_640.jpg

Topics
Java
HTML

Privat ☐

Submit

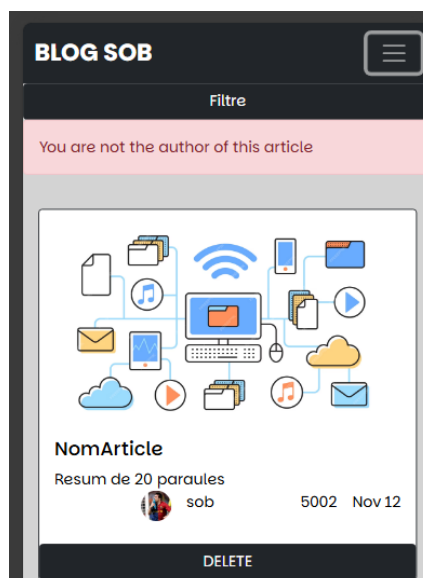
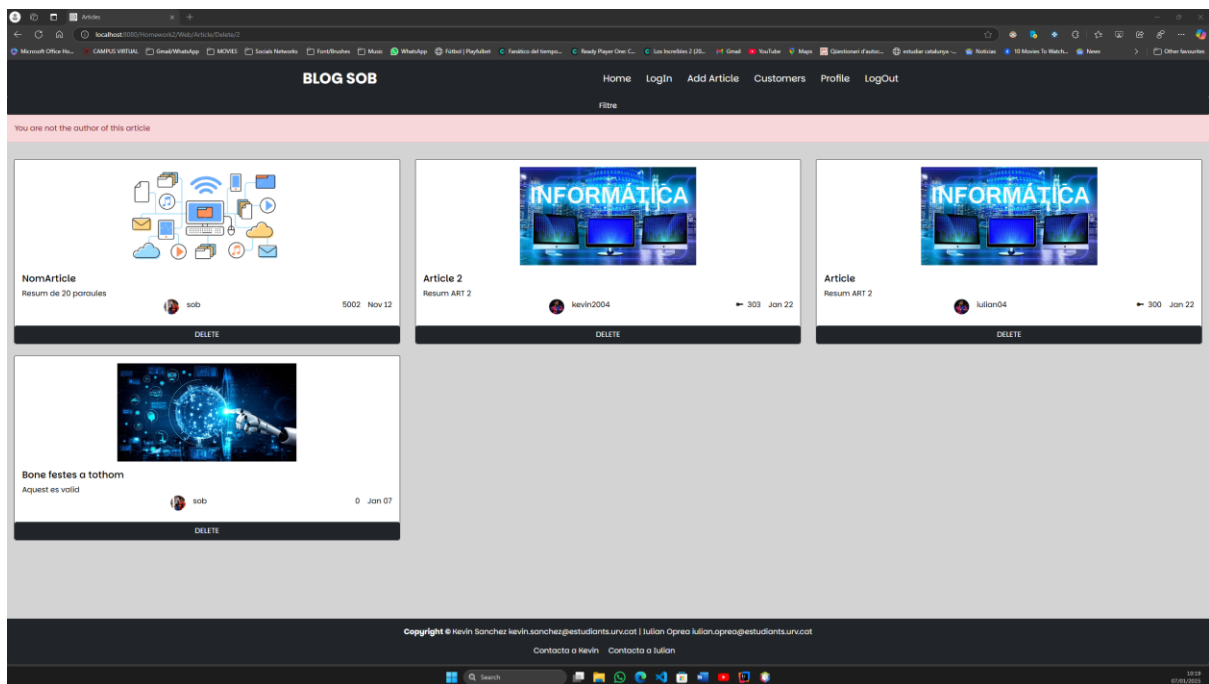


Correcte : SI

Cas 11

Volem fer delete de un article que no es nostre, llavors ens hauria de sortir un missatge de error especificant que el que volem fer no podem perquè no es el nostre article i no hi tenim permisos per fer-ho.

Sortida Real:

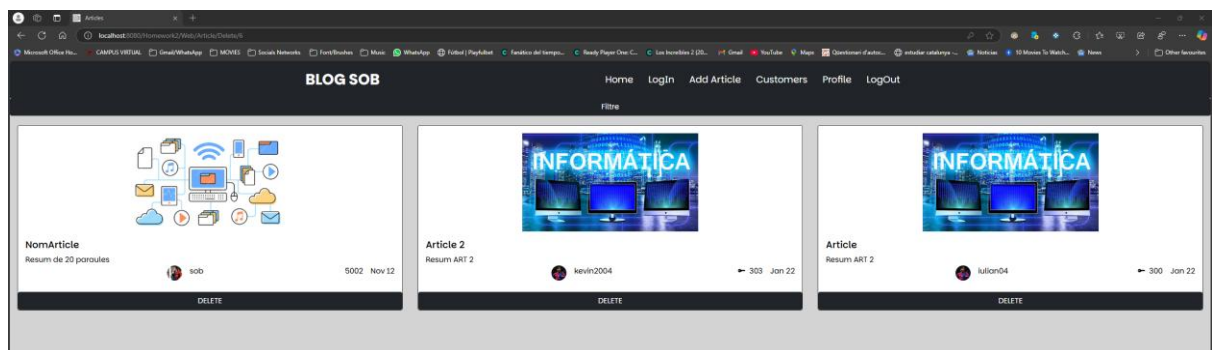
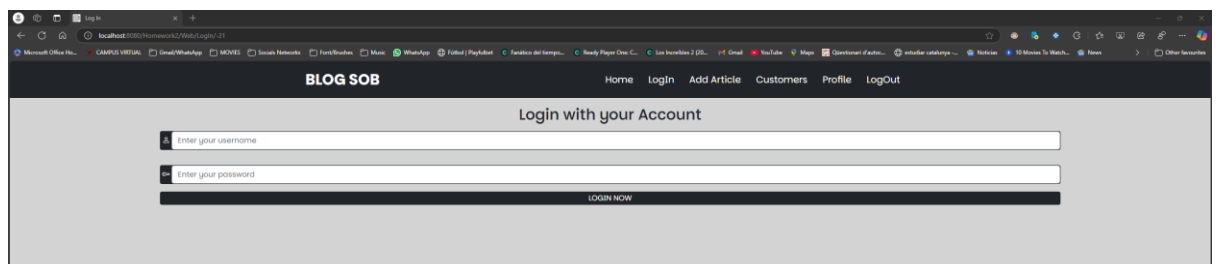
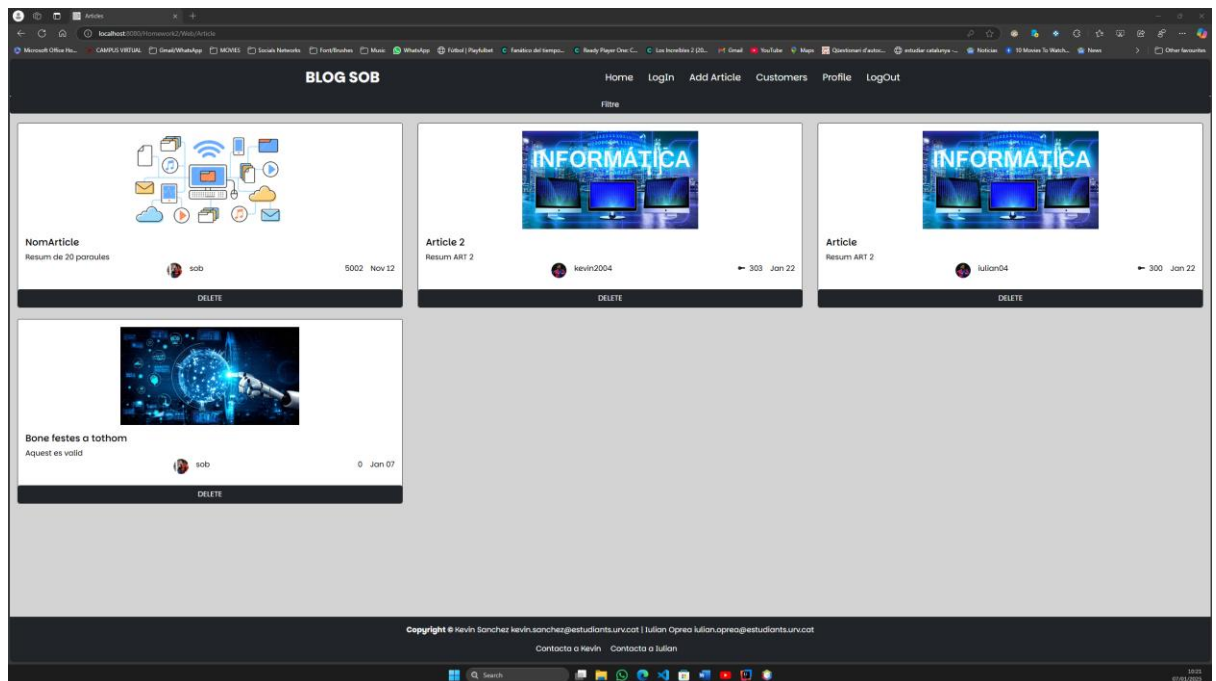


Correcte : SI

Cas 12

Volem fer delete de un article que es nostre llavors hauria de poder fer-lo a la perfecció, en aquest també es prova que si no estem iniciat sessió en demani iniciar sessió però també es faci el delete corresponent i al torna a la pagina inicial aquest article ja no es mostri per pantalla.

Sortida Real:



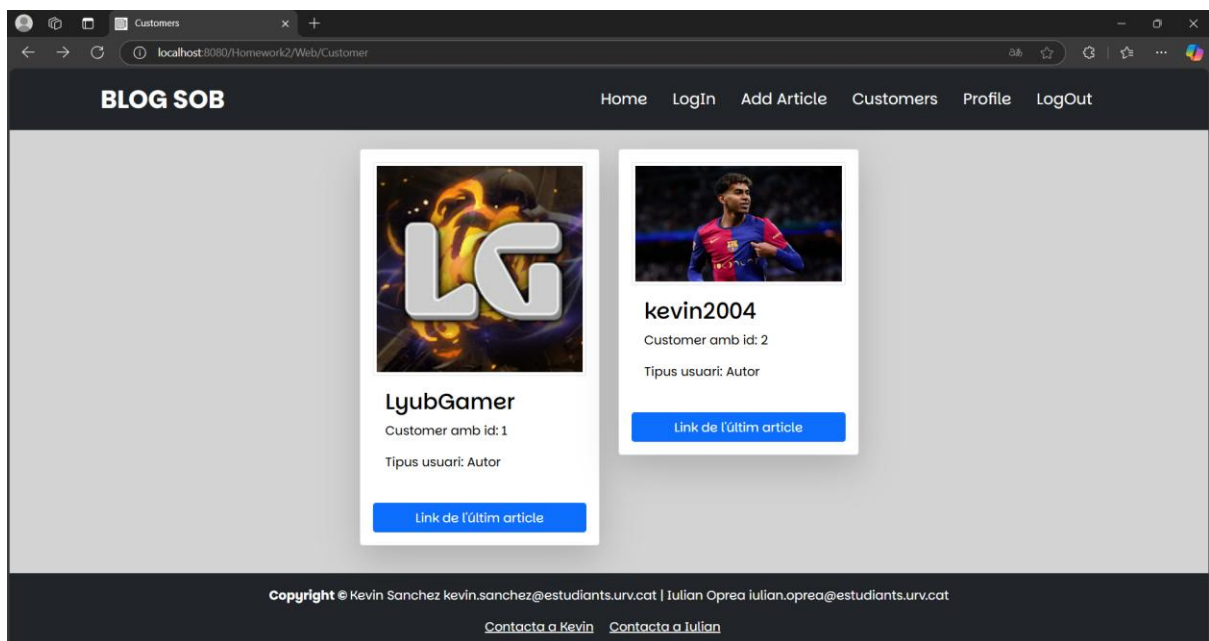
Correcte : SI

Cas 13

GET /rest/api/v1/customer

Volem obtenir una llista de customers amb informació reduïda ha de mostrar el tipus d'usuari (Lector o Autor) i en el cas que sigui autor el link a l'últim article (La modificació de l'usuari LyubGamer esta a posteriori).

Sortida Real:



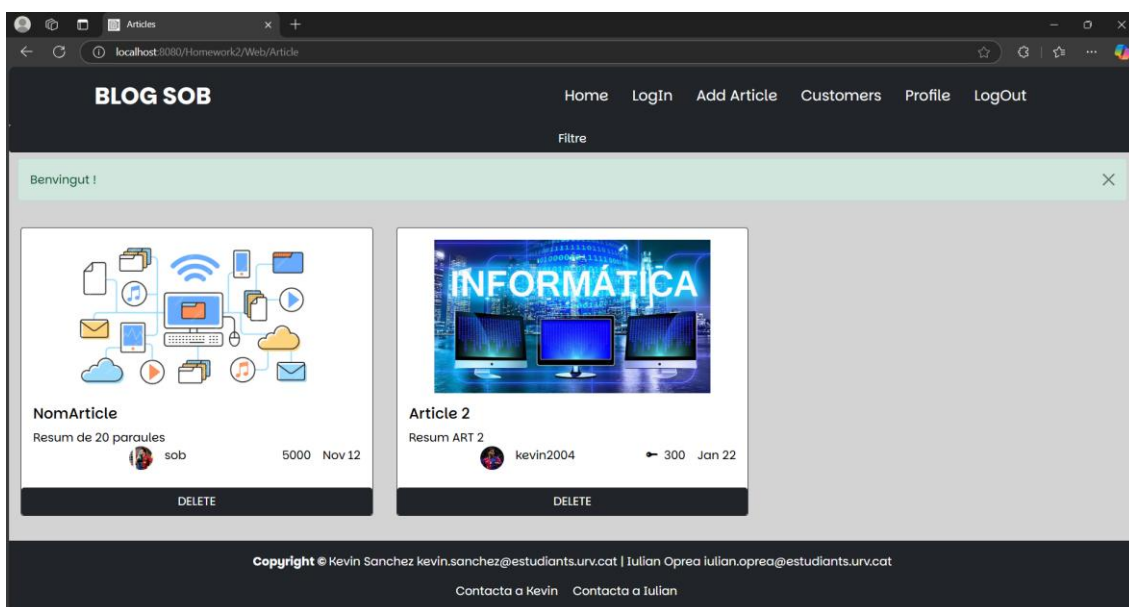
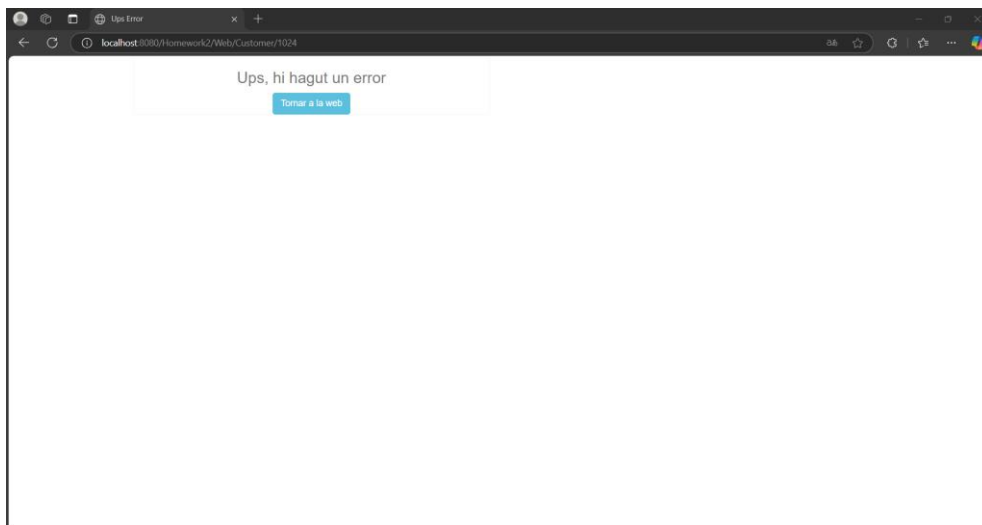
Correcte : SI

Cas 15

GET /rest/api/v1/customer/1024

En aquest cas estem demanant el customer amb identificador 1024, aquest no existeix a la base de dades, per lo que comprovarem que en el cas que no existeixi el recurs sol·licitat pel client, es retorni una web d'error dient que no s'ha trobat el recurs.

Sortida Real:



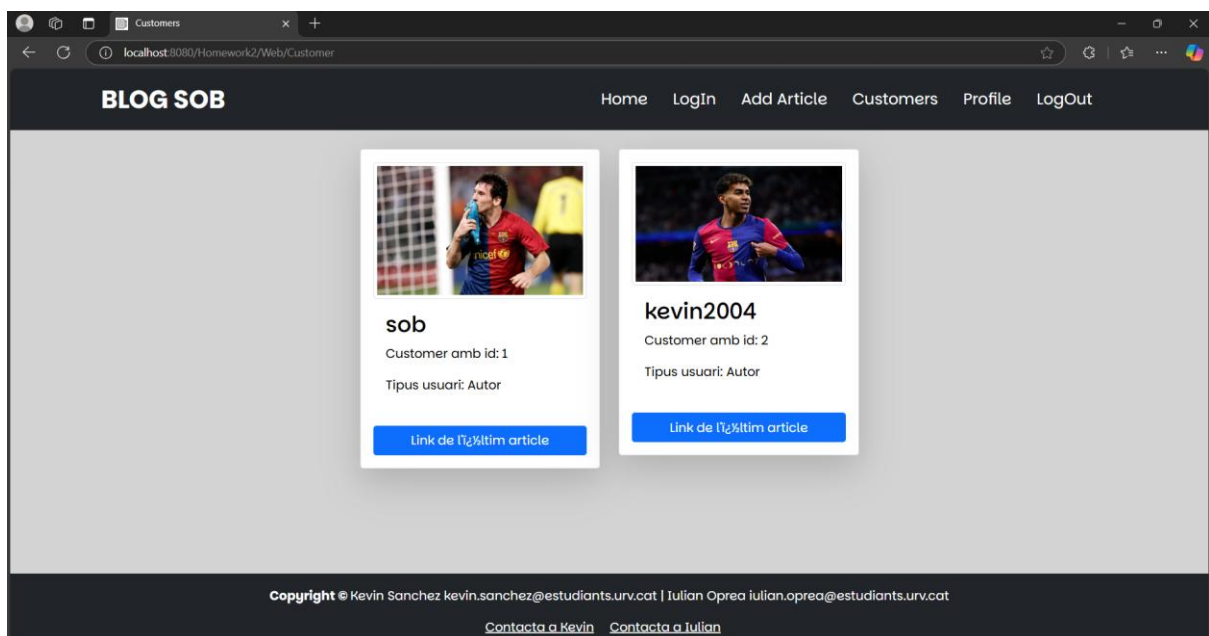
Correcte : SI

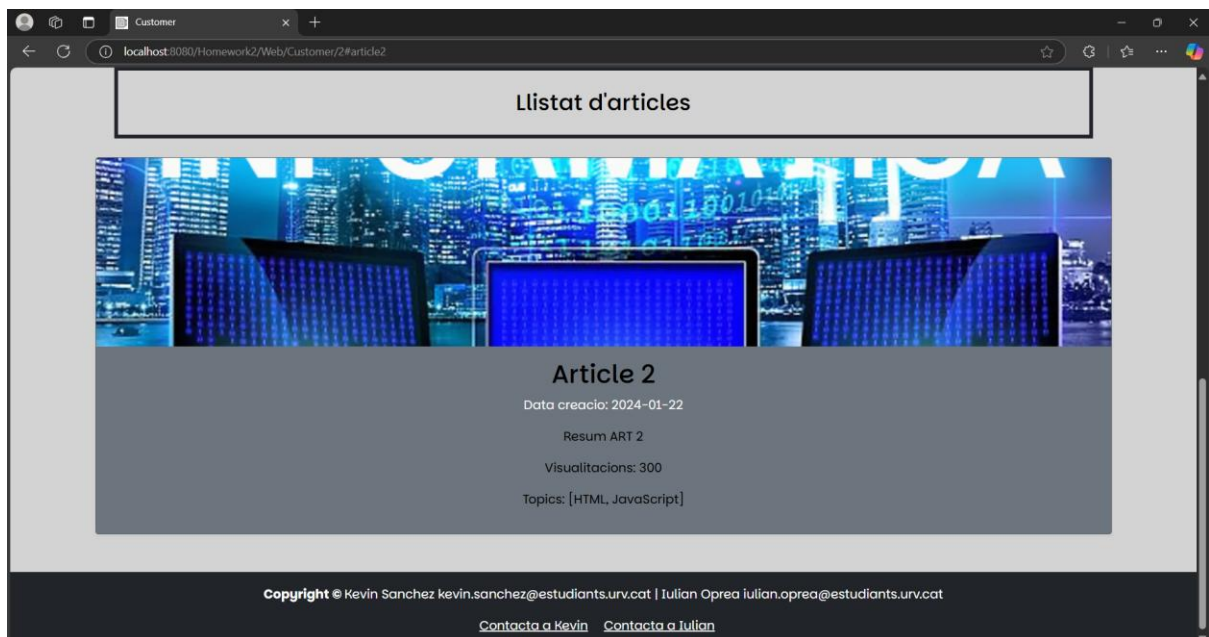
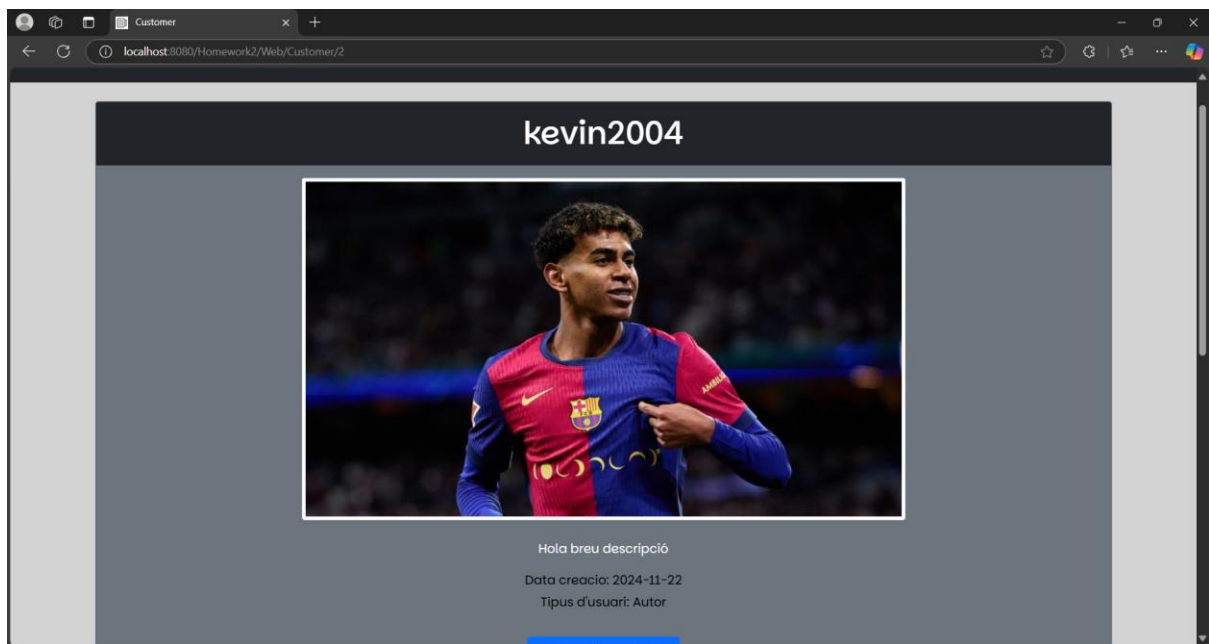
Cas 16

GET /rest/api/v1/customer/2

Volem obtenir la informació específica i detallada del customer amb identificador 2, aquesta informació ha d'estar en format JSON, aquesta petició ens ha de retornar els articles (així com la seva informació), la data de creació, la descripció del perfil, l'últim link al seu últim article, la seva foto de perfil, quin tipus d'usuari és i finalment lo seu username. Es pot accedir fent la petició manualment, o accedint en el llistat de customers.

Sortida Real:





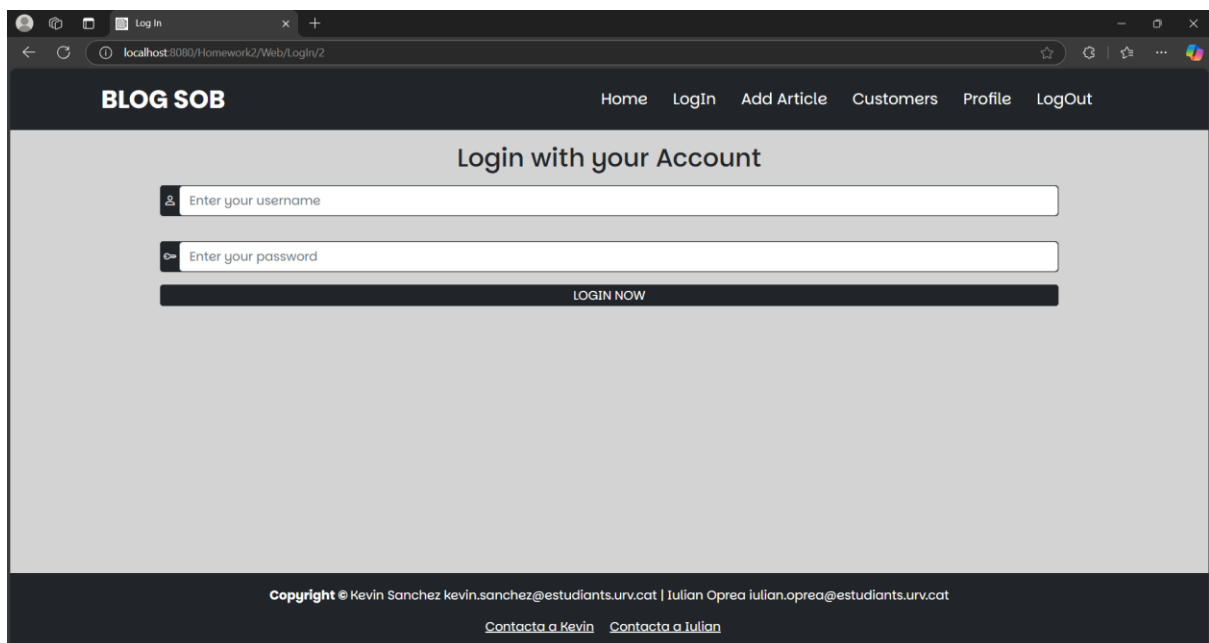
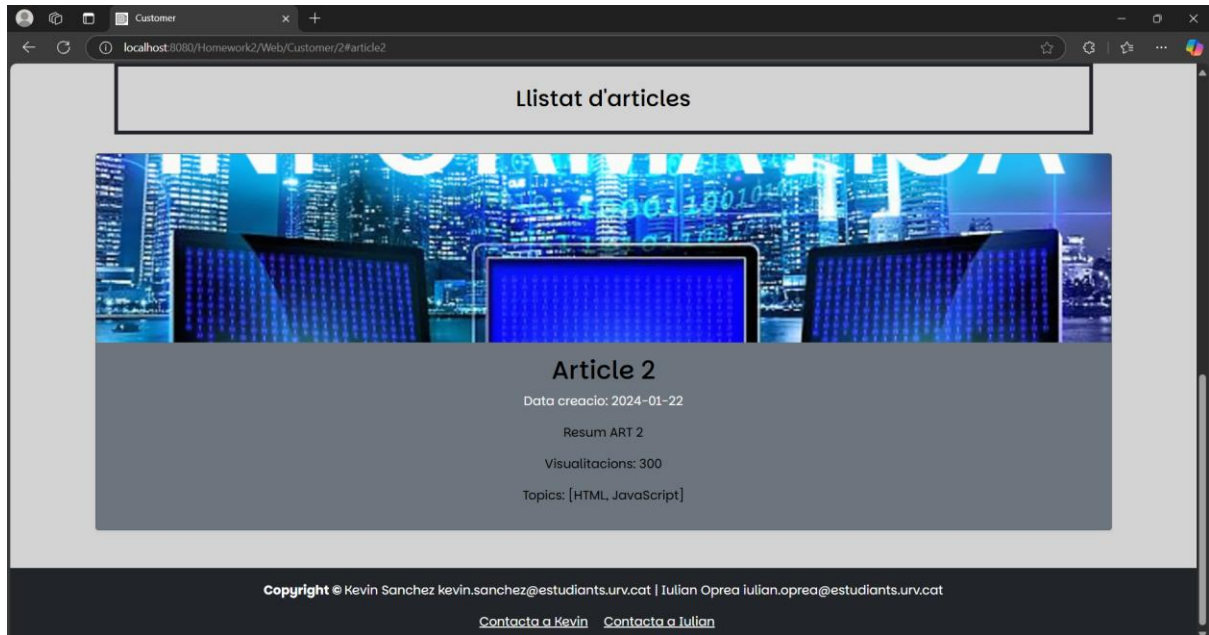
Correcte : SI

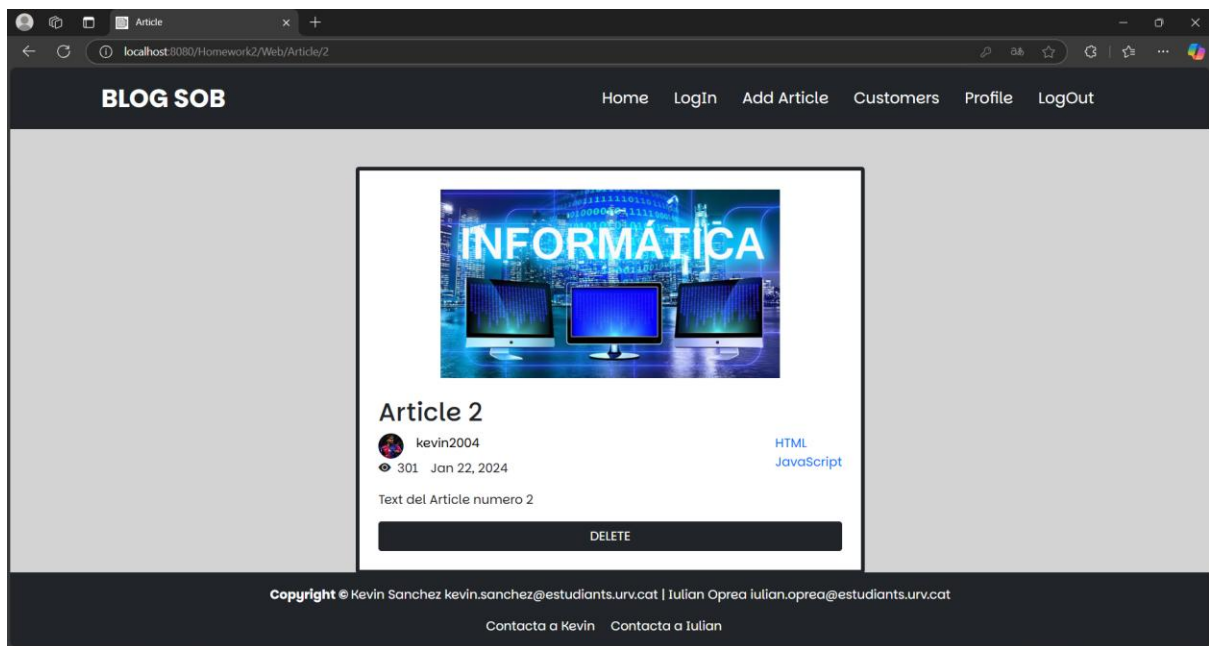
Cas 17

GET /rest/api/v1/customer/2

Volem accedir a l'article 2, que pertany al customer 2 des del seu perfil, ens ha de demanar credencials ja que es un article privat.

Sortida Real:





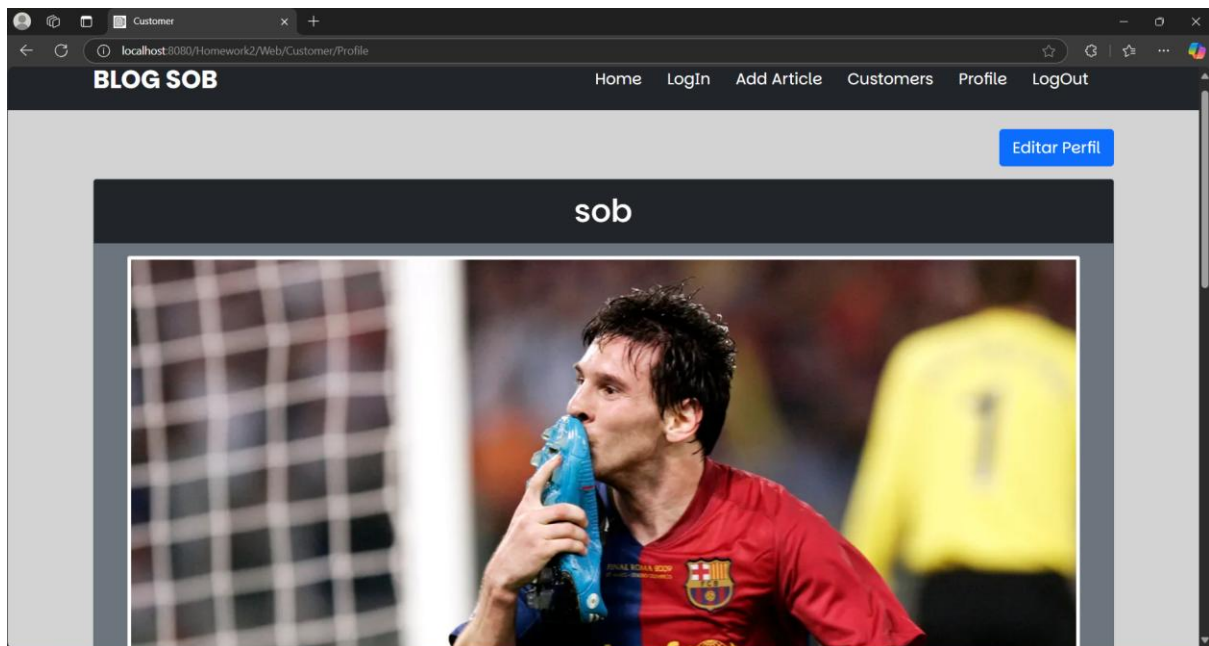
Correcte : SI

Cas 18

PUT /rest/api/v1/customer/1

Ens permet modificar el customer accedint al meu perfil i clickant al boto de editar perfil, es necessari estar logeat en les credencials per poder fer aquesta actualització de dades, per tant si l'usuari no coincideix en lo customer triat (està registrat però intenta modificar altres dades) sortirà un error. D'altra banda si es correcte actualitzarà les dades i les persistirà. A més, ens actualitzarà la nostra sessió per tal de no haver de tornar a iniciar el compte.

Sortida Real:



Editar Perfil

localhost:8080/Homework2/Web/Customer/EditProfile

BLOG SOB

[Home](#) [LogIn](#) [Add Article](#) [Customers](#) [Profile](#) [LogOut](#)

Editar Perfil (les caselles en blanc mantindran les dades actuals)

Username

NouSob

Password

Descripcio

Foto de perfil

https://example.com

Enviar

Copyright © Kevin Sanchez kevin.sanchez@estudiants.urv.cat | Iulian Oprea iulian.oprea@estudiants.urv.cat

[Contacta a Kevin](#) [Contacta a Iulian](#)

Editar Perfil

localhost:8080/Homework2/Web/Customer/EditProfile

BLOG SOB

[Home](#) [LogIn](#) [Add Article](#) [Customers](#) [Profile](#) [LogOut](#)

S'han modificat les dades correctament

Editar Perfil (les caselles en blanc mantindran les dades actuals)

Username

Password

Descripcio

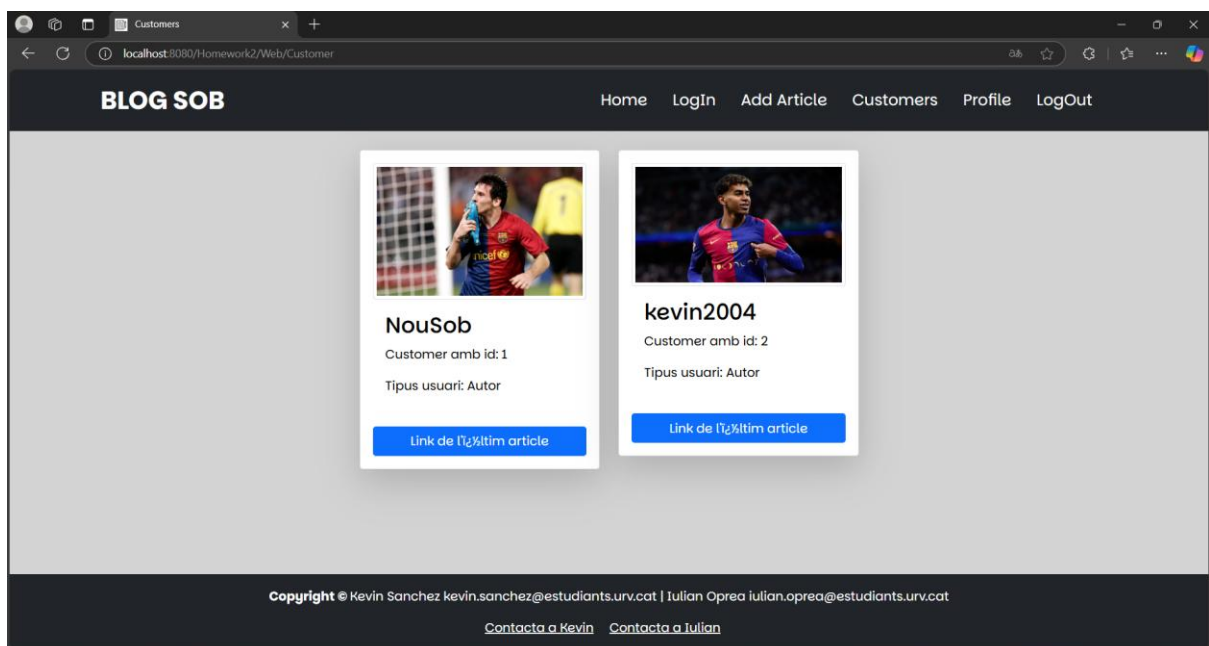
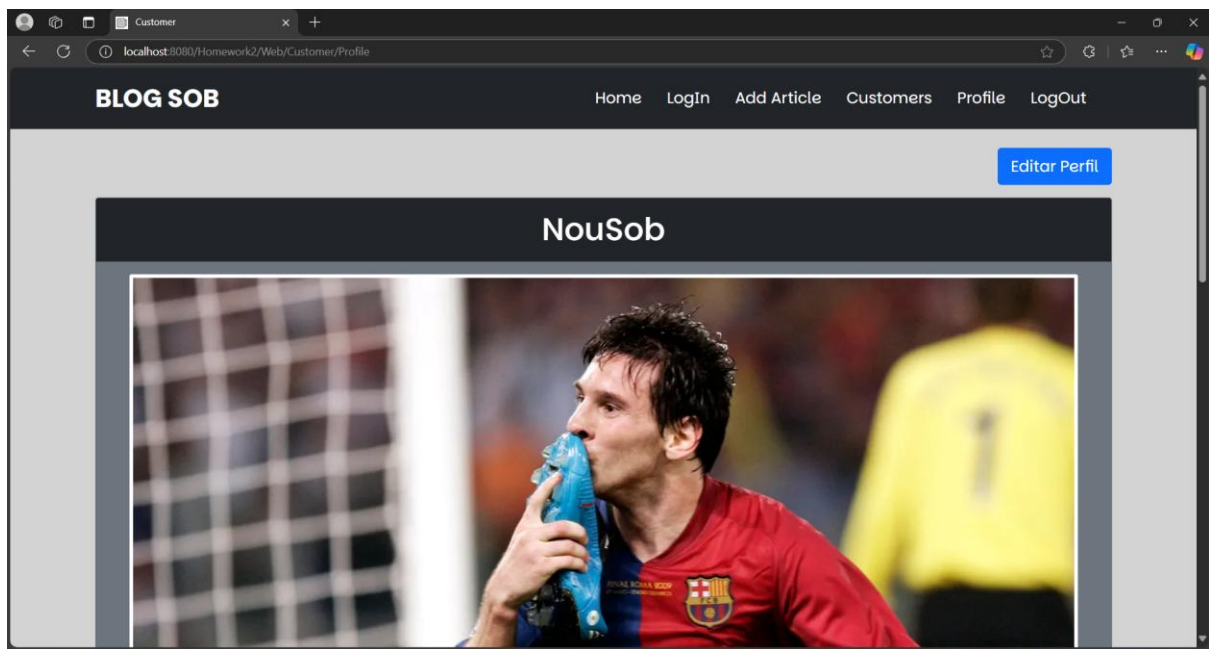
Foto de perfil

https://example.com

Enviar

Copyright © Kevin Sanchez kevin.sanchez@estudiants.urv.cat | Iulian Oprea iulian.oprea@estudiants.urv.cat

[Contacta a Kevin](#) [Contacta a Iulian](#)



Correcte : SI

Cas 19

PUT /rest/api/v1/customer/1

Modificarem tots els camp del customer, per comprovar si fa una actualització completa correcta també.

Editar Perfil (les caselles en blanc mantindran les dades actuals)

Username
LyubGamer

Password

Descripció
Aquest es un nou perfil, videos de brawl stars en youtube. Lyubgamer9859

Foto de perfil
https://yt3.googleusercontent.com/ytc/Aldro_lrjQZIC7nMgMneIasAZPAYzBQuq

Enviar

Copyright © Kevin Sanchez kevin.sanchez@estudiants.urv.cat | Iulian Oprea iulian.oprea@estudiants.urv.cat
[Contacta a Kevin](#) [Contacta a Iulian](#)

S'han modificat les dades correctament

Editar Perfil (les caselles en blanc mantindran les dades actuals)

Username

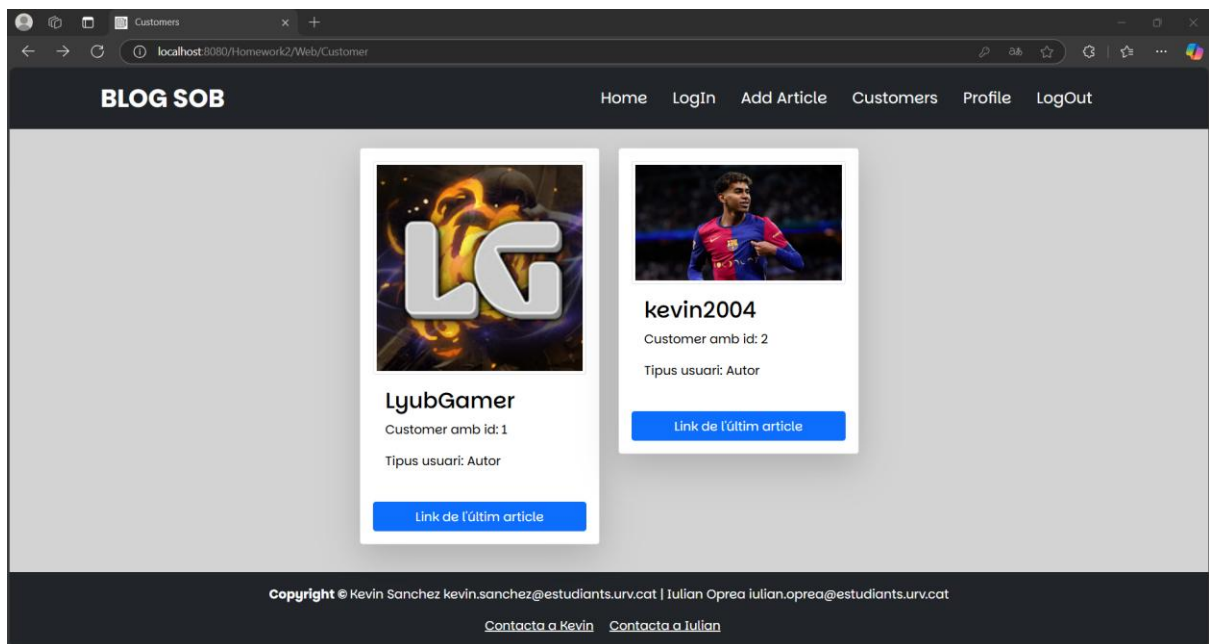
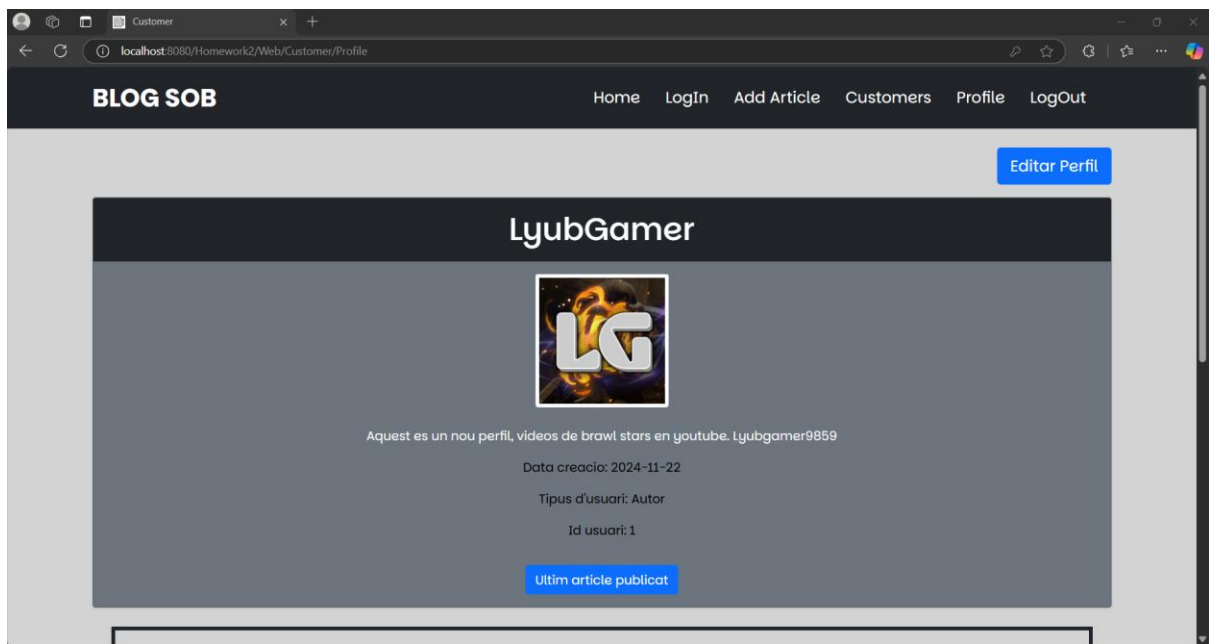
Password

Descripció

Foto de perfil
<https://example.com>

Enviar

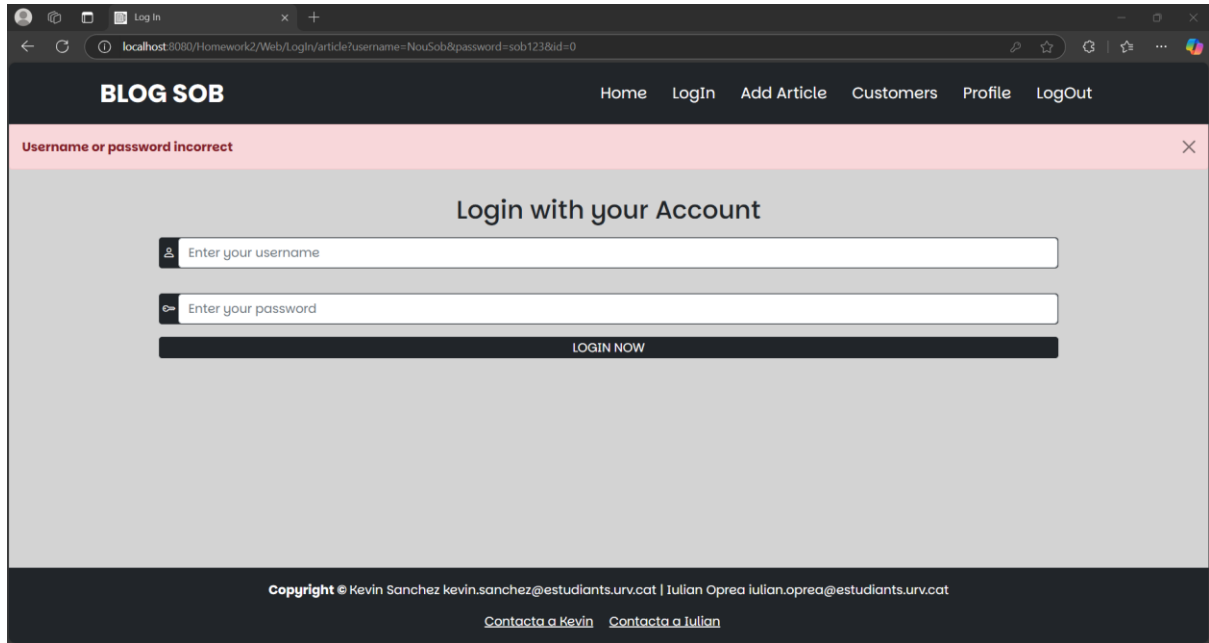
Copyright © Kevin Sanchez kevin.sanchez@estudiants.urv.cat | Iulian Oprea iulian.oprea@estudiants.urv.cat
[Contacta a Kevin](#) [Contacta a Iulian](#)



Correcte: SI

Cas 20

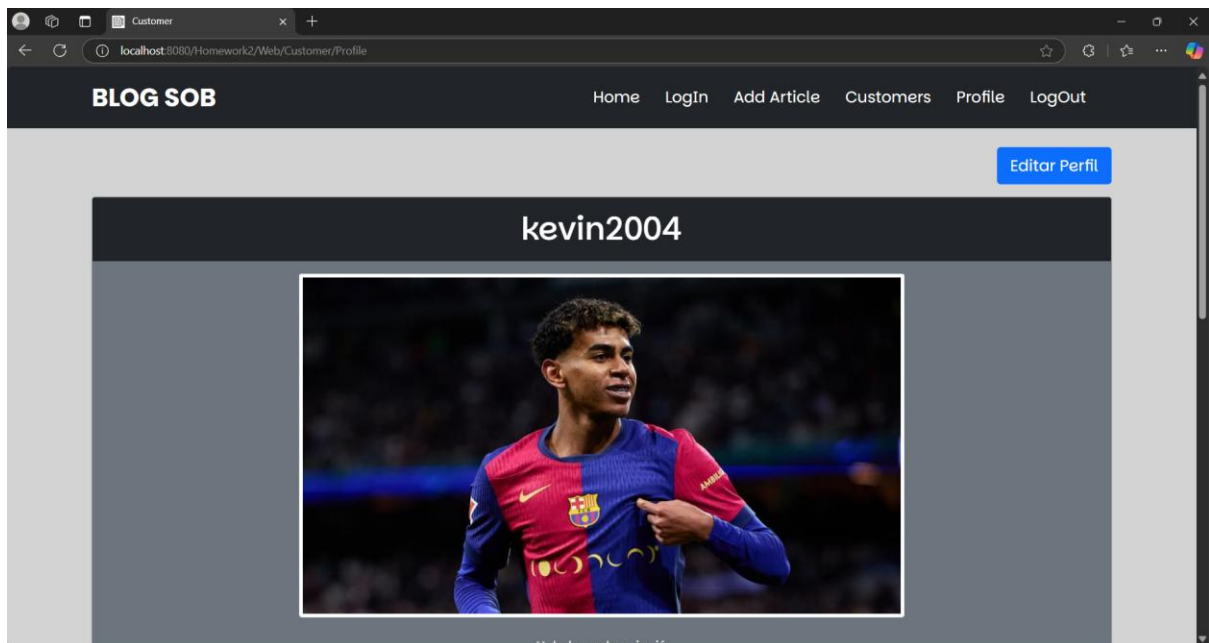
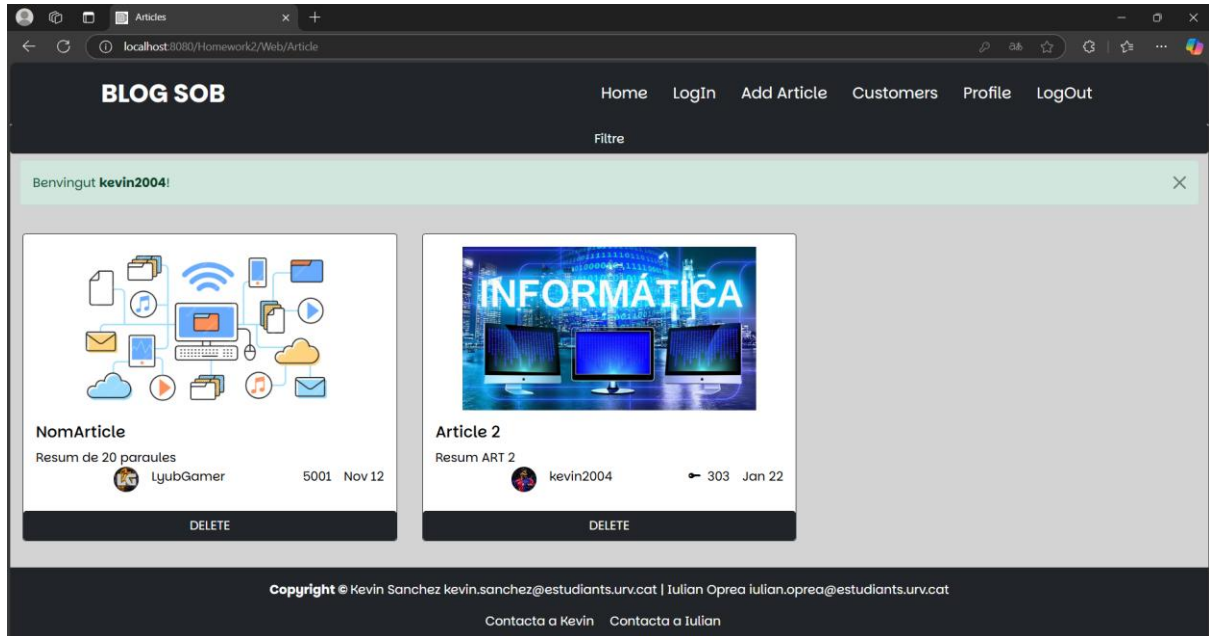
Al fer el login ficar credencials incorrectes, ens ha de mostrar gràficament per pantalla aquest error.



Correcte: SI

Cas 21

Al tindre una sessió iniciada, cada cop que anem al Home que ens digui “Benvingut, nom!” i si pressiones el nom et redirigeix cap al teu perfil.



Correcte: SI

Conclusions

Conclusions:

En finalitzar aquest projecte, hem arribat a diverses conclusions que ens han permès aprendre i comprendre millor com funciona el desenvolupament web. Hem après a gestionar tant el backend com el frontend d'una pàgina web, i entendre la importància de les interaccions entre ambdós components. Aquest coneixement ens serà molt útil en el futur, especialment si decidim treballar o fer pràctiques en empreses on busquen professionals amb aquestes habilitats. El fet de dominar les eines i tecnologies relacionades amb el desenvolupament web em permetrà afrontar projectes més complets.

Elements que més ens han agradat (ordenats per prioritat):

1. **Bootstrap (RWD):** Facilita la creació de pàgines web adaptatives de manera molt eficient.
2. **HTML:** És fonamental i relativament senzill, permetent crear pàgines de manera clara i estructurada.
3. **MVC:** El patró ajuda a organitzar el codi i mantenir l'arquitectura neta i escalable.

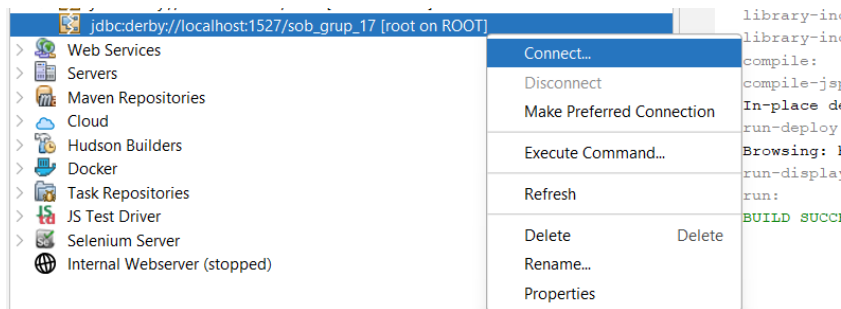
Elements que menys ens han agradat (ordenats per prioritat):

1. **JavaScript:** La manipulació de DOM i la interactivitat dinàmica poden ser difícils i confuses en alguns casos.
2. **CSS:** La creació de dissenys complexos i adaptatius em sembla sovint una tasca complicada.
3. **JSP:** No és tan intuïtiu com altres tecnologies i pot resultar difícil d'entendre al principi.

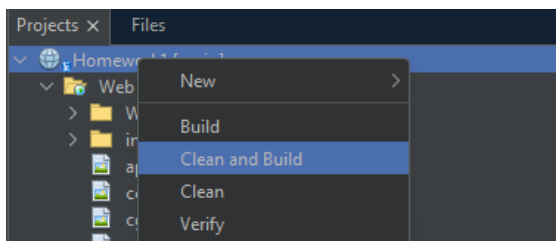
Aquest projecte ens ha donat una comprensió més profunda sobre el desenvolupament web i ens ha ajudat a identificar àrees en què podem millorar.

Manual d'instal·lació

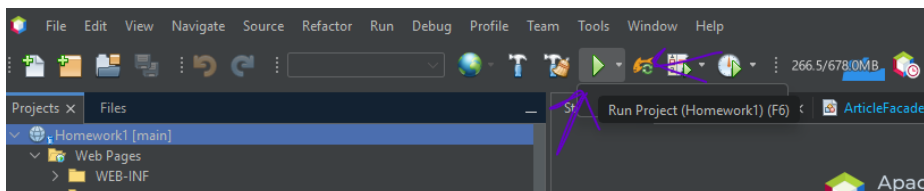
1. Obrim **Apache Netbeans**
2. Obrim el projecte **Homework1**
3. Ens connectem a la base de dades **sob_grup_17** (usuari i contrasenya: root)



4. Fem un “clean and build de Homework1”



5. Fem un “deploy and run Homework1”



6. Donem “click” al boto de insert



7. Obrim el projecte **Homework2**
8. Fem un “clean and build de Homework2”
9. Fem un “deploy and run Homework2” i esperem que es carregui la pàgina web i se'ns obrigui el navegador.