Feature 1

* Users can change the background color for each Todo item.

Here, a palette paint icon was placed on the far right of the row. By pressing this icon, the colour of the line can be changed.

(Burada satırda en sağ kısıma bir palet boya ikonu koyuldu. Bu ikona basınca kullanıcı satırın rengi değiştirilebiliyor)

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Feature 2

* Users can add and remove tags to the Todo items.

‘Tag’ column has been added to todo item class. It was enabled for the user to add tag to the related todo item and remove it if desired.

(Todo item sınıfına “Tag” kolonu eklendi. Kullanıcının ilgili todo item’a tag eklenmesi , isterse kaldırması sağlandı.)

metin, ekran görüntüsü, sayı, numara, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, yazı tipi, beyaz, grafik içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* Users can filter Todo items by tags.

Filtering input was made, parameter dto was created to the relevant api controller in the back-end and filtering was done with linq in ‘GetTodosQuery’.

(Filtreleme inputu yapıldı back-end’de ilgili api controller’a parametre dto’su oluşturuldu ve “GetTodosQuery” içerisinde linq ile filtreleme yapıldı.)

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, çizgi, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* Add shortcuts on UI for the most used tags by the user. (nice to have)

Previously made searches and how many times they were made on the filtering input were added as shortcuts and automatic filtering was enabled when clicked.

(Filtreleme inputunun üstüne önceden yapılan aramalar ve kaç defa yapıldığı shortcuts olarak eklendi ve tıklandığı durumda otomatik filtreleme yapılması sağlandı)

metin, ekran görüntüsü, yazı tipi, logo içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

The values of the filters tried to be made in the back-end and how many times they want to be filtered are kept, so that when you change the page, the previously made tag filters are brought as a shortcut.

(Back-end’de yapılmaya çalışılan filtreler değerleri ve kaç defa filtrelenmek istendikleri tutuldu , böylelikle sayfa değiştirince de önceden yapılmış tag filtreleri shortcut olarak getirilmekte.)

 metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* Add text search. (nice to have)

This item was seen as open-ended and ‘Title’ filtering enhancement in ‘TodoList’ was preferred. By using ‘KeyUp’ event in the input, connection from the front side to the back side was provided. A Title was added to the ‘TodoFilterDto’ class created for Tag and filtering was provided with linq in the backend.

(Bu madde açık uçlu olarak görüldü ve “TodoList” içerisinde “Title” filtreleme geliştirmesi tercih edildi. İnput’da “KeyUp” event’i kullanılarak ön yüzden arka yüze bağlantısı sağlandı. Tag için oluşturulan “TodoFilterDto” sınıfına bir de Title eklendi ve backend’de linq ile filtreleme sağlandı.)

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.



Feature 3

* Add soft delete for Todo lists and items.
* Deleted items should not be deleted from the database
* Deleted items should not be shown on the UI or included in any query

The **"IsDeleted"** columns were added to the **"TodoItems"** and **"TodoLists"** tables via migration, and a boolean **"IsDeleted"** variable was added to the corresponding entity classes. In the **"ApplicationDbContext"** class, the **"SaveChangesAsync"** function was modified to automatically set the **"IsDeleted"** column to **true** for delete operations. Instead of completely removing the record from the database, its **"State"** was changed to **"Modified"**. After this development, it was confirmed that records with **"IsDeleted" = true** were not displayed in the UI but could still be retrieved from the database using a query.

(“TodoItems” ve “TodoLists” tablolarına “IsDeleted” kolonları migration ile tablolara eklendi ve ilgili entity sınıflarına bool tipinde “IsDeleted” değişkeni eklendi. “ApplicationDbContext” sınıfında “SaveChangesAsync” fonksiyonuna “Delete” işlemlerinde “IsDeleted” kolonuna otomatik “true” atandı ve veri tabanından kaydı tamamen silmek yerine “State” durumu “Modified” olarak değiştirildi. Yapılan geliştirme sonrası “IsDeleted” kolonunun 1 olduğu indexlerin ui’da görülmediği , veri tabanından sorgu ile select edilebildiğine ulaşıldı.)

metin, ekran görüntüsü, sayı, numara, ekran, görüntüleme içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, yazı tipi, yazılım, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.