

## Furkan Kazım Çam

21360859036

### 12.Hafta Rapor

Önceki hafta yazdığımız kodlara ek yaparak devam edeceğiz.

Bu hafta önceki hafta yaptığımız işlemlere try catch ve finally blokları ekleyeceğiz. Peki bunlar nedir?

#### Try Catch Finally

Try, catch ve finally blokları JavaScript'te (ve birçok diğer programlama dilinde) hataların yönetilmesi ve belirli kodların her durumda çalıştırılmasını sağlamak için kullanılır.

##### Try Bloğu

try bloğu, çalıştırmak istediğiniz kodu içerir. Eğer bu blokta bir hata meydana gelirse, kontrol catch bloğuna aktarılır.

##### Catch Bloğu

catch bloğu, try bloğunda meydana gelen hataları yakalar ve bu hatalarla nasıl başa çıkacağınızı belirlemenizi sağlar. Hata yakalandığında, catch bloğundaki kod çalıştırılır.

##### Finally Bloğu

finally bloğu, hata olsun veya olmasın her durumda çalıştırılan kodu içerir. Genellikle kaynakların serbest bırakılması veya temizleme işlemleri için kullanılır. Örneğin, bir veritabanı bağlantısının kapatılması gibi işlemler burada yapılabilir.

Evet kodumuza entegre ederek işleme başlayalım. Geçen haftadan farklı olarak kısayol olması için 2 değişken tanımlayarak onlara değerlerini verelim.

```
const usersCollection = db.collection("users");
const tasksCollection = db.collection("tasks");

// Example: Insert a user
try {
  const result = await usersCollection.insertOne({ name: "Sadık", age: 25 });
  console.log('A document was inserted with the _id:', result.insertedId);
} catch (err) {
  console.error('Something went wrong trying to insert the new document:', err);
} finally {
  await client.close();
  console.log('Connection closed');
}
```

Try bloğunda verilen bilgiler veri tabanına eklenir eğer eklenemez ise catch bloğundaki error mesajı yazdırılır ve her halükârda finally bloğuna girip bağlantı kapatılır ve console.log ile kullanıcı bilgilendirilir. Çıktımız aşağıdaki gibi olmalıdır.

```
PS C:\Users\Furka\OneDrive\Masaüstü\Bahar23-24\Node.js\11.Hafta\TaskManager> node md.js
Pinged your deployment. You successfully connected to MongoDB!
A document was inserted with the _id: new ObjectId('664788c971b4c05f55eb3718')
Connection closed
stü\Bahar23-24\Node.js\11.Hafta\TaskManager> █
```



```
_id: ObjectId('664788c971b4c05f55eb3718')
name : "Sadık"
age : 25
```

Bunu diğer kod bloklarımıza da uygulayalım.

```
46 // Example: Insert many user
47 kullanıcılar=[
48   {name: 'Umut', age: 49},
49   {name: 'Ayşe', age: 47}
50 ]
51
52 try {
53   const result = await usersCollection.insertMany(kullanıcılar);
54   console.log(result.insertedCount, ' documents inserted successfully');
55 } catch (err) {
56   console.error('Something went wrong trying to insert the new documents: ',
57     err);
58 } finally {
59   await client.close();
60   console.error('Connection closing');
61 }
62 // Example: Update a user
63 // try {
64 //   const result = await usersCollection.updateOne({ name: "Can" }, { $set: { age: 100
65 //   console.log('Updated', result.modifiedCount, 'documents');
```

SORUNLAR ÇIKIŞ HATA AYIKLAMA KONSOLU TERMINAL BAĞLANTI NOKTALARI SEARCH ERROR AZURE AÇIKLAMALAR POLYGLOT

```
Connection closed
PS C:\Users\Furka\OneDrive\Masaüstü\Bahar23-24\Node.js\11.Hafta\TaskManager> node md.js
Pinged your deployment. You successfully connected to MongoDB!
2 documents inserted successfully
Connection closing
Connection closed
```

Kullanıcılar isminde bir array tanımladık ve onu insertMany ile veri tabanına gönderdik.

Çıktı olarak ise kaç dosya yüklendiği ve veri tabanı bağlantısının kapatıldığı görülmektedir.



```
_id: ObjectId('66478e0d4a1bafced098b213')  
name : "Umut"  
age : 49
```

```
_id: ObjectId('66478e0d4a1bafced098b214')  
name : "Ayşe"  
age : 47
```

Sırada ise var olan bir veriyi güncellemenin nasıl yapıldığı var.

```
63 // Example: Update a user  
64 try {  
65   const result = await usersCollection.updateOne({ name: "Furkan" }, { $set: { age: 100 } });  
66   console.log('Updated', result.modifiedCount, 'documents');  
67 } catch (err) {  
68   console.error('Something went wrong trying to update the document:', err);  
69 }  
70
```

updateOne komutu var olan veriyi set parametresi ile günceller.

```
PS C:\Users\furka\OneDrive\Masaüstü\Bahar23-24\Node.js\11.Hafta\TaskManager> node md.js  
Pinged your deployment. You successfully connected to MongoDB!  
Updated 1 documents  
Connection closed
```



```
_id: ObjectId('663b3148031b4d6bf16f59ce')  
name : "Furkan"  
age : 100
```

Veri tabanındaki kullanıcılar arasında arama işlemi ile devam edelim.

```
// Example: Find a user
try {
  const user = await usersCollection.findOne({ name: "İbrahim" });
  console.log('Found user:', user);
} catch (err) {
  console.error('Something went wrong trying to find the document:', err);
}
```

findOne komutu ile veriyi bulur ve istediğimiz işlemi yapabiliriz.

```
PS C:\Users\furka\OneDrive\Masaüstü\Bahar23-24\Node.js\11.Hafta\TaskManager> node md.js
Pinged your deployment. You successfully connected to MongoDB!
Found user: {
  _id: new ObjectId('663b3351a1869c80c20f35cc'),
  name: 'İbrahim',
  age: 21
}
Connection closed
PS C:\Users\furka\OneDrive\Masaüstü\Bahar23-24\Node.js\11.Hafta\TaskManager>
```

Verinin tüm bilgileri yazdırılmıştır.

```
▶ _id: ObjectId('663b3351a1869c80c20f35cc')
  name: "İbrahim"
  age: 21
```

```
try {
  //const result = await usersCollection.findOne({name: 'Cana'});
  const result = await usersCollection.findOne({name: 'Umut', age:49});
  if (!result)
    console.log('Could not find user')
  else
    console.log('Found user: ', JSON.stringify(result))
} catch (err) {
  console.error('Something went wrong in find: ', err);
}
finally {
  await client.close();
  console.error('Connection closing');
}
```

Burda ise kullanıcıyı bulup bilgilerini Json formatında konsola yazdırılması istenmiştir. Eğer bulamaz ise error mesajı ile kullanıcı bilgilendirilecektir.

```
PS C:\Users\furka\OneDrive\Masaüstü\Bahar23-24\Node.js\11.Hafta\TaskManager> node md.js
Pinged your deployment. You successfully connected to MongoDB!
Found user: { "_id": "66478e0d4a1bafced098b213", "name": "Umut", "age": 49 }
Connection closing
Connection closed
```

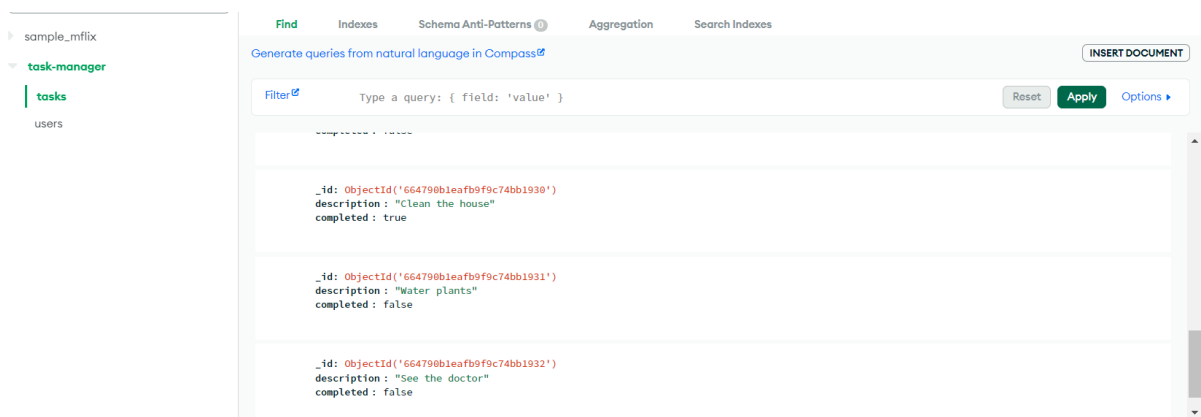
Kullanıcı bilgileri yazdırıldı.

Tasks adlı veri tabanına da veri gönderelim.

```
// Example: Insert multiple tasks
try {
  const tasks = [
    { description: "Clean the house", completed: true },
    { description: "Water plants", completed: false },
    { description: "See the doctor", completed: false },
  ];
  const result = await tasksCollection.insertMany(tasks);
  console.log(result.insertedCount, "documents inserted successfully");
} catch (err) {
  console.error('Something went wrong trying to insert the new documents:', err);
}
```

Tasks dizisi tanımlanır ve insertMany ile bu veriler tasks adlı veri tabanına gönderilir.

```
PS C:\Users\furka\OneDrive\Masaüstü\Bahar23-24\Node.js\11.Hafta\TaskManager> node md.js
Pinged your deployment. You successfully connected to MongoDB!
3 documents inserted successfully
Connection closed
```



Veri tabanındaki tüm bilgileri yazdırmak için ise alttaki kodu kullanabilirsiniz.

```
101 // }
102
103 try {
104   const cursor = tasksCollection.find({completed:false});
105   while (await cursor.hasNext()) {
106     console.log(await cursor.next());
107   }
108 } catch (err) {
109   console.error('Something went wrong in find: ', err);
110 }
111 finally {
112   await client.close();
113   console.error('Connection closing');
114 }
```

```
PS C:\Users\furka\OneDrive\Masaüstü\Bahar23-24\Node.js\11.Hafta\TaskManager> node md.js
Pinged your deployment. You successfully connected to MongoDB!
```

```
{
  _id: new ObjectId('663b33b526741051965592af'),
  description: 'Ödev Yap',
  completed: false
}
{
  _id: new ObjectId('66446c16bd089ca45b7a2794'),
  description: 'Water plants',
  completed: false
}
{
  _id: new ObjectId('66446c16bd089ca45b7a2795'),
  description: 'See the doctor',
  completed: false
}
{
  _id: new ObjectId('6647887898fb983cc59235bb'),
  description: 'Water plants',
  completed: false
}
{
  _id: new ObjectId('6647887898fb983cc59235bc'),
  description: 'See the doctor',
  completed: false
}
{
  _id: new ObjectId('664790b1eafb9f9c74bb1931'),
  description: 'Water plants',
  completed: false
}
{
  _id: new ObjectId('664790b1eafb9f9c74bb1932'),
  description: 'See the doctor',
  completed: false
}
}
Connection closing
Connection closed
```