

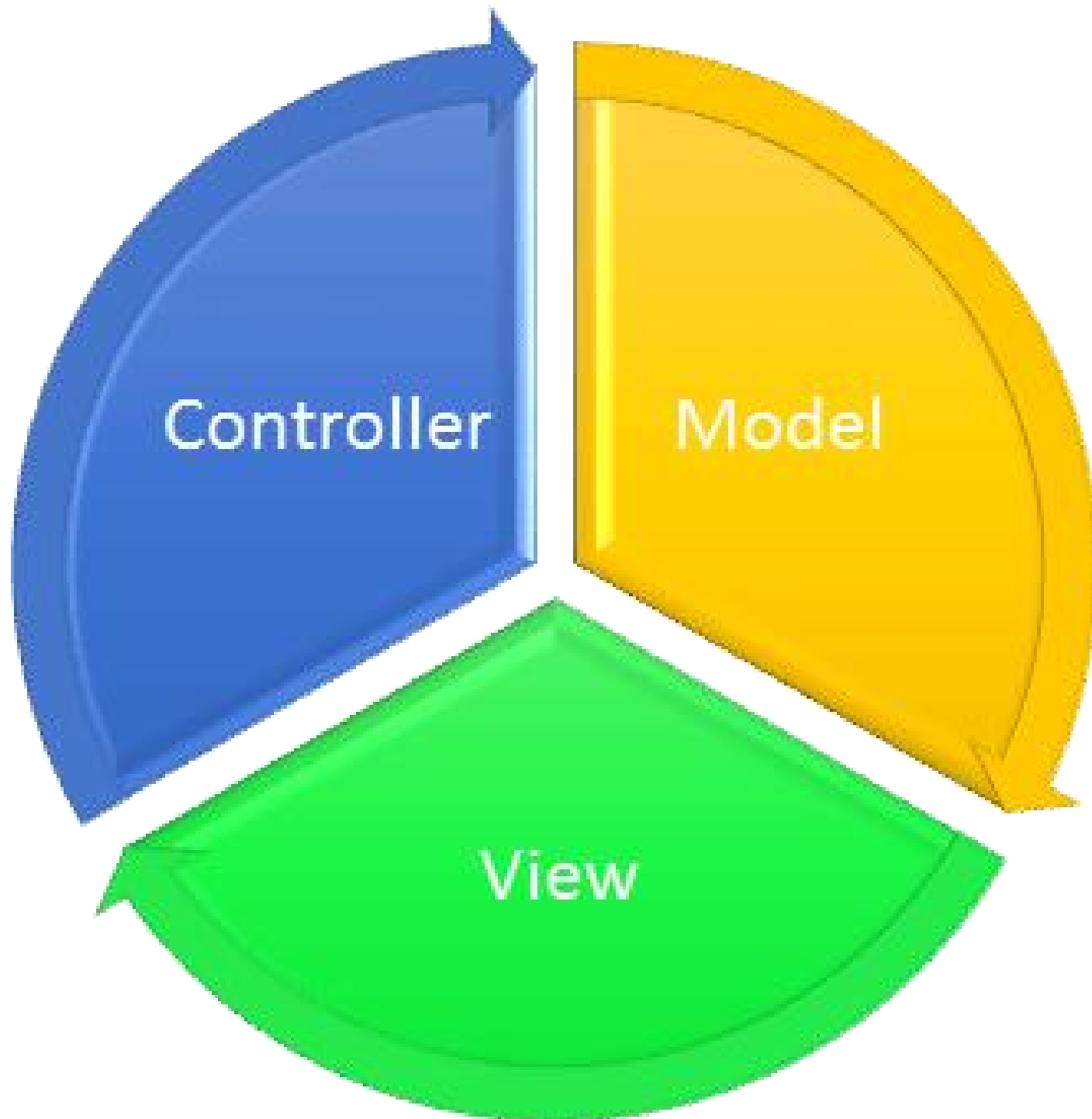


in/poyrazavsever



WEB GELİŞTİRİCİLER İÇİN TEMEL MİMARİLER

# MVC Nedir?





in/poyrazavsever



## Neden Düzen Önemli?

Bir proje büyüdükçe kod karmaşıklaşır.

Farklı fonksiyonlar birbirine karıştığında, hataları bulmak ve yeni özellik eklemek zorlaşır.

İşte tam bu noktada MVC mimarisi devreye girer.



in/poyrazavsever



## MVC Nedir?

MVC (Model - View - Controller), bir uygulamayı 3 ana katmana ayırarak düzen sağlar:

- **Model:** Veriyi ve iş mantığını yönetir.
- **View:** Kullanıcıya gösterilen arayüzdür.
- **Controller:** Kullanıcı ile sistem arasındaki köprüdür.

Bu yapı, “her katman tek bir göreve odaklansın” anlayışıyla tasarlanmıştır.



in/poyrazavsever



## Model Katmanı

Veritabanı işlemleri, doğrulamalar ve iş mantığı burada tanımlanır.

```
// user.model.js
export class User {
  constructor(name, email) {
    this.name = name;
    this.email = email;
  }

  save() {
    // Veritabanına kaydetme işlemi
  }
}
```



in/poyrazavsever



## View Katmanı

HTML, CSS ve bazen JavaScript dosyalarından oluşur.

Görevi: Veriyi görsel olarak sunmak.



```
<h1>Kullanıcı Kaydı</h1>
<form>
  <input type="text" placeholder="İsim">
  <input type="email" placeholder="E-posta">
  <button>Kaydet</button>
</form>
```



in/poyrazavsever



## Controller Katmanı

Controller, kullanıcının yaptığı işlemleri yakalar ve doğru Model & View ile iletişime geçer.

```
// user.controller.js
import { User } from './user.model.js';

export const createUser = (req, res) => {
  const user = new User(req.body.name, req.body.email);
  user.save();
  res.render('userSuccess');
};
```



in/poyrazavsever



## MVC Akış Örneği

1. Kullanıcı formu doldurur (View)
2. Form Controller'a gönderilir
3. Controller, veriyi Model'e iletir
4. Model veriyi işler ve kaydeder
5. Sonuç tekrar View üzerinden kullanıcıya gösterilir

Bu akış, düzenli bir kod mimarisi oluşturur.





in/poyrazavsever



## MVC'nin Avantajları

- Kod daha okunabilir hale gelir
- Ekip çalışmasını kolaylaştırır
- Test etmek ve bakım yapmak daha basittir
- Yeni özellikler eklemek risk oluşturmaz

Kısacası:

MVC, yazılım mimarilerinin temel taşıdır.





in/poyrazavsever



## Gerçek Hayattan Bir Benzetme Yapalım

Bir restoran düşün:

- Model: Mutfak (yemeklerin hazırlanması)
- View: Garsonun müşteriye sunduğu tabak
- Controller: Siparişi mutfağa ileten garson

Her biri kendi işini yapar, sonuçta sistem kusursuz işler.

Senin projelerinde mimari yapı nasıl?

MVC kullanıyor musun, yoksa farklı bir  
yaklaşım mı tercih ediyorsun?



in/poyrazavsever

