**1-MVC KAVRAMI**

Model - View – Controller olmak üzere 3 bileşenden oluşan kullanıcı arabirimleri geliştirmekte kullanılan yazılım tasarım modelidir. Farklı alanlarda da kullanılan , genel olarak web uygulaması geliştirmekte kullanılan bir tasarım modelidir. Buna ihtiyaç duyulmasındaki sebep ise , birbirinden bağımsız 3 katmandan oluşmaktadır. Bu da büyük projelerde projenin yönetimi ve kontrolünün daha rahat sağlanabilmesi ve kip çalıışması olsun daha sonra gelecek yeni ekipler için olsun projeyi anlamak ve geliştirmek için gereklidir.

**Model**

Veri ile ilgili işlemlerin gerçekleştiği katman. ORM araçlarını , veri tabanına erişim ve veri tabanı ilişkilerini içeren katmandır.

**View**

Kullanıcının ekranda gördüğü katmandır. Arayüz katmanı.

**Controller**

Model ve view arasındaki katman. Kullanıcıdan aldığı verileri model katmanı ile işlenmesini sağlayan katmandır.

MVC OOP ile uyumlu çalışan bir yapıdır. OOP ‘de 4 temel katman vardır.

* **Kapsülleme (Encapsulation)**
* **Soyutlama (Abstraction)**
* **Miras Alma (Inheritance)**
* **Çok Biçimlilik (Polymorphism)**

**Kapsülleme (Encapsulation)**

Hangi verilerin veya metodların dışarıdan erişebileceğini ayarladığımız katman. Public,private ve protected olarak 3 kısımda ilgilenilir.

**Public:** Verilerin herkes tarafından erişilmesini istediğimizde kullanırız.

**Private:** Sadece bulunduğu sınıf üzerinden erişim sağlamasını istersek kullanırız.

**Protected:** Aynı paket ve alt sınıflardan erişimi istendiğinde kullanırız.

**Soyutlama (Abstraction)**

Bu katman kullanıcılara verilerin soyutlandığı , sınıfın özelliklerinin ve metodlarının tanımlandığı yer.

**Miras Alma (Inheritance)**

Alt sınıfların üst sınıfların özelliklerinden yararlanabildiği katmandır.

**Çok Biçimlilik (Polymorphism)**

Alt sınıfın üst sınıfın özelliklerini kullanırken aynı zamanda farklı şekilde de kullanılmasına denir.

**2-BAĞIMSIZ PLATFORM HABERLEŞMESİ**

Birbirinden bağımsız platformların birbiri ile haberleşmeleri api ’ler aracılığı ile sağlanır. Rest api ‘ler client-server arasındaki haberleşmeyi sağlayan HTTP protokolü üzerinden çalışan mimari bir yapıdır. X platformu Rest api’ye json formatında bir istekde bulunup ,Rest api http ile y platformuna istekde bulunur. Y platformu da json formatında Rest api’ye dönüş yapar , Rest api bu cevabı http ile X platformuna cevap verir.

**3- GÜNCELLEMEDEN GÜNCELLEME**

Aklima gelen AJAX olur . Asynchronous JavaScript And XML. Java Script web sayfalarını yenilemeden arka planda veri gönderme ve veri alma işlemlerini yapar aynı zamanda sayfa yenilenmeden sayfayı günceller.

**4-FOR DÖNGÜSÜ**



**5-REMOTE BAĞLANTI**

ssh kullanici@sunucu\_ip\_adresi ile erişim kontrolü yaparım,

ssh -l kullanici\_adi ip\_adresi -p port\_no komutunu kullanarak shell ile bağlantı kurarım . sftp user@server\_ipaddress komutu ile sftp bağlantısı oluştururuz. Get /uzak\_dizin/dosya\_adi server’dan dosya çekmek için kullanırım, get /dosya\_adi/uzak\_dizin server’a dosya atmak için kullanırım.

**7-APACHE SOLR**

[http://solr-server-adress/solr/solr-sermaniz/select?q=date> 2020](http://solr-server-adress/solr/solr-sermaniz/select?q=date%3e%202020) ocak. Şeklinde yazardım.