

(Geliş Tarihi / Received Date: 15.11.2019, Kabul Tarihi/ Accepted Date: 24.12.2020)

## Yaygın Eğitim Kurumları e-Kurs Başvuru Sistemi

Muammer AKÇAY\*<sup>1</sup>, Tolgahan ÇOBANOĞLU<sup>2</sup>

<sup>1</sup>Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kütahya

<sup>2</sup>Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kütahya

### Anahtar Kelimeler:

başvuru,  
e-kurs,  
yaygın eğitim,  
dağıtık sistemler,  
bilgi teknolojileri

**Özet:** e-Kurs Başvuru Sistemi yaygın eğitim merkezlerine gelen kursiyer adaylarının dilediği kursa ön kaydını oluşturabilmesi amacıyla tasarlanmıştır. Yapılan tespitlerde kursiyerlerin kurs başvurusu yapabilmeleri için hazırlanan e-yaygın portalı yeteri kadar kararlı çalışmamaktadır. Başvuruların yönetimi ve denetimi yapılamaması nedeniyle de beklentileri karşılamamaktadır. Bu sebeple de kurum yöneticileri tarafından kurs başvurusu için yazılı form tutulmaktadır. Bu formlar, her bir kurs, kursiyer, grup, zaman dilimi için ayrı ayrı muhafaza edilmektedir. Fakat bu durum zaman kaybına, bu işle uğraşan fazladan en az iki kişilik personel maliyetine ve zaman zaman kaybedilen formlar nedeniyle de çok önemli verilerin kaybına neden olmaktadır.

Geliştirilen bu proje, yöneticilere yönetilebilir bir platform sunmakta ve bu konudaki karmaşıklığı ortadan kaldırarak, %100 oranda zaman-maliyet ve personel tasarrufu imkânı sağlayacaktır.

Projenin başarılı olması durumunda tüm yaygın eğitim yapan kurumların (olgunlaşma enstitüleri, halk eğitimi merkezleri, özel kurslar, sürücü kursları vb.) kursiyer kayıtlarını çevrimiçi oluşturmaları sağlanabilecek, bu kurumlar üzerinde büyük bir yük olan bu iş çözüme kavuşacaktır.

## Life Long Learning Institutions e-Course Application System

### Keywords:

application,  
e-course,  
non-formal education,  
distributed systems,  
information technologies

**Abstract:** The e-Course Application System is designed for the pre-registration of the course candidates who wish to attend the non-formal education centers. According to the findings, the e-widespread portal prepared for the trainees to apply for a course does not work sufficiently stable. It does not meet the expectations due to the lack of management and supervision of the applications. For this reason, a written form is kept for the course managers. These forms are kept separately for each course, trainee, group, time period. However, this leads to loss of time, the cost of at least two additional personnel involved in the business and the loss of very important data due to forms lost from time to time.

This project provides managers with a manageable platform and will eliminate the complexity of this issue and provide 100% time-cost and personnel savings.

If the project becomes successful, all non-formal education institutions (maturation institutes, public education centers, private courses, driving courses, etc.) will be able to create online trainee registrations and this issue will be solved, which is a huge burden on these institutions.

\*İlgili yazar / corresponding author: makcay26@gmail.com

### 1. GİRİŞ

Bu çalışmanın amacı, yaygın eğitim kurumlarına kurs başvurusunda bulunmak için gelen kursiyer adayları için ön başvurularının alındığı ve bu ön başvurulara göre kursiyer adaylarının toplantılara davet edilerek kesin kayıtlarının sağlanmasıdır [1].

Bu çalışma, yaygın eğitim yapan tüm kurumlarda (olgunlaşma enstitüleri, halk eğitimi merkezleri, özel kurslar, sürücü kursları vb.) uygulanabilecek, 7'den 77'ye tüm yaş ve eğitim düzeyindeki vatandaşlar tarafından ki bu kurumlara okuma-yazma bilmeyenler dahi gelmektedir, kolayca kullanılabilir.

Çalışmanın tamamlanması ile birlikte, kurs başvurusu için yaygın eğitim kurumlarına gelen kursiyer adaylarından alınan bilgilerin yönetilebilir bir veri tabanında

saklanarak, sistematik bir şekilde kullanılması sağlanacaktır. Bu durum da, kurum yöneticilerinin hata yapma ihtimallerini ortadan kaldıracaktır. Öyle ki, hali hazırda kullanılan yöntemler ile kurs başvurusu için bu kurumlara gelen kursiyer adaylarının önemli bilgilerinin gizlenmesi konusunda ciddi endişelere imkân vermektedir. Aynı zamanda, zaman zaman başvurusu bulunan adayların bilgileri doğru saklanmadığı (elle alınan başvuru formları, kurumlara özgü olmayan ve kararlı çalışmayan programlar, excel listeleri vb.) için kaybolabilmekte, bu bilgilerin saklanması için ayrıca personel çalıştırılmakta, bu iş için ciddi zaman kaybı olmaktadır. Ayrıca, bazen kursiyerlerin başvuruları gözden kaçırılmak suretiyle kursiyerlerle iletişim kurulamamakta ve bu durumda da kursiyerlerin mağduriyeti ile sonuçlanabilmektedir.

Çalışma, tüm yaygın eğitim yapan kurumların (olgunlaşma enstitüleri, halk eğitimi merkezleri, özel kurslar, sürücü kursları vb.) kursiyer kayıtlarını çevrimiçi oluşturmalarını sağlayacak, kurum yöneticilerin sık sık karşılaştıkları veri ve zaman kaybı önlenecektir.

Aynı zamanda projenin kurum yöneticileri tarafından rahatlıkla yönetilebilir olması ve kursiyer adaylarına kurs başvurusu için rehberlik etme durumunun ortadan kalması nedeniyle personel tasarrufu da sağlanmış olacaktır.

Bu çalışma, veri kaybı, zaman kaybı, ilave personel ihtiyacı gibi ciddi maliyet ve yük getiren sorunlara çözüm getirmekte, kamu kurumlarında uygulanabilecek olması nedeniyle de ülkemiz açısından kaynak tasarrufu sağlayacaktır. Aynı zamanda kursiyer adaylarının mağdur olmalarının önüne geçecektir.

### 1.1. İhtiyaç Analizi

Projeye başlamadan önce ihtiyaçlar belirlenmiştir. Yapılan değerlendirme sonucu, ihtiyaçlar ise şu şekilde sıralanmıştır:

- Proje nasıl bir veri tabanına sahip olmalıdır? Veri tabanının oluşturan tablolar nelerdir?
- Proje hangi programlama dili, mimari ve yazılımlar kullanılarak tasarlanacaktır?
- Projeyi oluşturan kodlama, tanımlama ve teknolojiler nasıl kullanılmalıdır?

Tablo 1'de ihtiyaçlar doğrultusunda yapılacak işler sıralanarak, Şekil 1'de Gantt şeması ile gösterilmiştir

### 1.2. Proje İş Tablosu

Proje tasarımı aşamasına geçilmeden önce hangi işlerin yapılacağı Proje İş Tablosunda gösterilmiştir (Tablo 1).

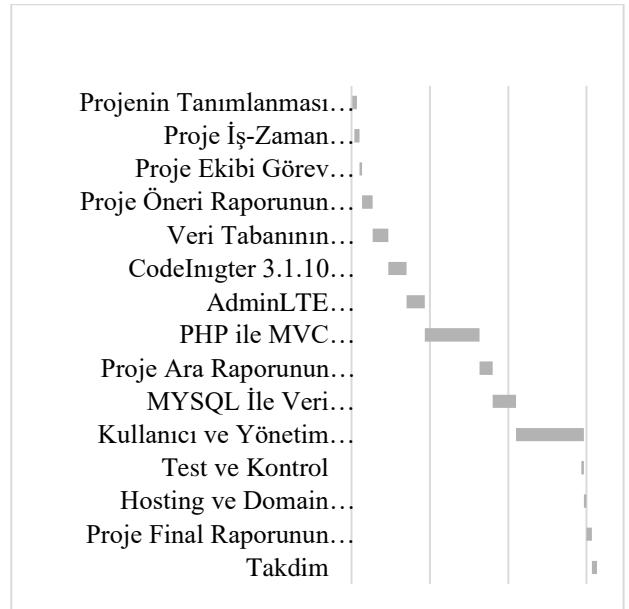
**Tablo 1.** İhtiyaçlar doğrultusunda yapılacak işler

Proje İş Tablosu	
Sıra No	İşlemler
1.	Projenin Tanımlanması ve Hedeflerin Belirlenmesi
2.	Proje İş Tablosunun Oluşturulması
3.	Proje Ekibi Görev Dağılımının Oluşturulması

4.	Proje Öneri Raporunun Oluşturulması
5.	Veri Tabanının Modellemesi
6.	CodeIgniter 3.1.10 Web Framework Kurulumu
7.	AdminLTE OpenSource Template'in projeye uygulanması
8.	PHP ile MVC Yapısının Oluşturulması
9.	MYSQL İle Veri Tabanı Tasarımı
10.	Kullanıcı ve Yönetim Panelinin Tasarım ve Kodlama İşlemlerinin Yapılması
11.	Hosting ve Domain İşlemlerinin Yapılarak, İnternet Ortamında Yayın Yapılması
12.	Test ve Kontrol

### 1.3. Proje Gantt Şeması

Proje iş tablosunda belirlenen aktivitelerin planlanması ve sıralaması Şekil 1'de Gantt Şeması ile grafiksel olarak ifade edilmiştir.



**Şekil 1.** Gantt şeması

Projedeki işlemlerin başlangıç ve bitiş işlemlerini alt alta zaman skalasına yerleştirerek Gantt şemasına aktaran diyagram, hangi işin nerede başlayıp nerede bittiğinin anlaşılması konusunda yardımcı olacaktır (Şekil 1).

## 2. MATERYAL VE METOT

### 2.1. Ana Sayfa Tasarımı

Projenin giriş sayfasını oluşturan bölümdür. Kullanıcı bu sayfada "Kursiyer" ya da "Yönetici" olarak sisteme giriş yapabilir. Bu sayfa oluşturulurken öncelikle /application/view dizini içerisinde *welcome\_message.php* isimli görünüm sayfası hazırlanmıştır.

*welcome\_message.php* isimli görünüm sayfasının çalıştırılması için application/controller dizini içerisinde *Welcome.php* isimli denetçi sayfa oluşturularak *index()* fonksiyonu ile *welcome\_message.php* sayfası yüklenmiştir.

```
//Welcome.php denetçi sayfası
<?php
class Welcome extends CI_Controller {
    public function index()
    {
        $this->load->view('welcome_message');
    }
}>>
```

Burada önemli olan oluşturulan Welcome sınıfının CI\_Controller sınıfındaki metotlara ve değişkenlere sahip olabilmesi için CI\_Controller sınıfına genişletilmesi ve sınıf adının dosya adı ile aynı isimde olmasıdır. Aksi halde sınıf içerisindeki tanımların hiçbiri çalışmayacaktır.

## 2.2. Kursiyer Giriş Sayfası Tasarımı

Projenin kursiyerler tarafından kullanılacak olan bölümüdür. Bu bölümde, kurs başvurusu için “*Kurs Başvuru Ekranı*” bağlantısı, kurslar hakkında bilgi edinmek için “*Kurslar Hakkında*” ve kurum idaresinin görüşlerini alabileceği “*Anketimize Katılın*” bağlantısı bulunacaktır.

## 2.3. Kurs Ön Kayıt (Başvuru) Sayfası Tasarımı

Projenin kursiyerler tarafından kurslara ön başvuru yapmak için kullanacakları bölümüdür. Bu bölümde kursiyerlerden kurs başvurusu için gerekli bilgileri girmeleri ve önceden yöneticiler tarafından hazırlanan tanımlı kurslar arasından seçim yapmaları istenir.

Projenin hedefleri ve kullanım alanları ile ilgili olarak kursiyerlerin başvurularını kurum girişine konulacak dokunmatik ekranlardan yapmalarının daha kullanışlı olacağı tespit edilmiştir.

Bu durum göz önünde bulundurularak Kurs - Ön Kayıt sayfası için bir de ekran klavyesi tasarlandı. Klavye, herhangi bir metin girişi yapılacağında (input işlemi) kullanıcıya gösterilmekte, diğer durumlarda gizlenmektedir. Klavye, jQuery Bootstrap Touch Keyboard plugini ile oluşturularak;

```
//on_kayit.php görünüm sayfası; ekran klavyesi tanımı
$('input').keyboard({placement:'bottom',
initcaps:true});
```

kod bloğu ile sayfaya dahil edilmiştir.

Ayrıca kullanıcının formu doldururken bir standarda göre giriş yapmasını sağlamak amacıyla JQuery teknolojisi ile maskeleyiş işlemi yapılmıştır.

Kullanıcıların formdaki bilgileri eksiksiz olarak doldurmalarının ardından verilerin veri tabanındaki “*basvuru*” tablosuna kaydının gerçekleştirilmesi AJAX teknolojisi kullanılarak model-view-controller (MVC, model-görünüm-denetçi) [2-3] yapısı ile sağlanmıştır.

Yine Ön Kayıt işlemi esnasında forma eksik ya da hatalı bilgi girişini engellemek amacıyla view (görünüm) ve controller (denetçi) sayfalarımızda form doğrulama işlemi ile kısıtlar tanımlanmıştır. Bu işlem forma veri girişi gerçekleştirilen tüm sayfalarda, “*Giriş, Kurs-Kursiyer İşlemleri, Tanımlı Kurslar, Ajanda, Kullanıcı İşlemleri*”, CodeIgniter’in [4-7] sağladığı validation sınıfı ile gerçekleştirilmiştir.

```
<span class="help-block"></span>
<?php echo form_error('Adi'); ?>
```

Yukarıdaki kod bloğu benzer şekilde tüm form elemanları altında ifade edilerek, hata mesajının ekranda gösterilmesi sağlanmıştır.

*on\_kayit.php* denetçi sayfasında tanımlanan save() fonksiyonuna gönderilen \$data dizisi, Basvuru\_model.php sayfasında yapılan tanımlamalar ile veri tabanına kaydedilecektir.

Model sayfasında tanımlanan \$data dizisine ait verilerin veri tabanına eklenmesini sağlayan insert() fonksiyonu CodeIgniter’in basit kodlama yapısına bir örnektir.

Veri tabanı üzerinde yapılacak ekle, sil, listele, güncelle işlemlerine ait tüm tanımlamalar hazır ve hataya imkan verilmeyecek şekilde DB\_query\_builder.php dosyası içerisinde tanımlanmış durumdadır. Burada yapılan temel işlem sadece yapılacak işe ait fonksiyonu model sayfasında çağırma işlemidir. (Örneğin güncelleme işlemi için update(), listeleme işlemi için num\_rows() gibi.)

Not: Bu dosya veri tabanı ile ilgili işlemlerin tümünde kullanılacaktır. Burada bilgi verilmiş olup, diğer bölümlerde bu kısma değinilmeyecektir.

## 2.4. Form Doğrulama (Validation) Sınıfı

CodeIgniter, yazılan kod miktarını en aza indirmeye yardımcı olan kapsamlı bir form doğrulama ve veri hazırlama sınıfı sunar.

Form doğrulamasını uygulamak için üç şeye ihtiyaç vardır:

- Form içeren bir view dosyası.
- Veri başarılı bir şekilde gönderildikten sonra gösterilecek bir “başarı” mesajı içeren bir view dosyası.
- Gönderilen verileri almak ve işlemek için bir denetçi yöntemi.

CodeIgniter, belirli bir alan için ihtiyaç duyulan doğrulama kurallarını belirlemeye, sıraya koymaya ve alan verilerini aynı anda hazırlayarak önceden işlemeyi sağlar.

Doğrulama kurallarını ayarlamak için set\_rules() yöntemi kullanılır.

```
$this->form_validation->set_rules('TcNo','Tc
Kimlik No','trim|required|max_length[11]');
```

CodeIgniter form\_validation kütüphanesini kullanmak için denetçi sayfasına `$this->load->library('form_validation');` kod bloğu mutlaka dâhil edilmelidir [2].

## 2.5. Yönetim (Admin Panel) Sayfası Tasarımı

Kurum yöneticileri tarafından login sayfasında girilecek kullanıcı adı ve şifre ile erişim sağlanacak olan yönetim sayfasında;

- Kurs tanımlama, tanımlı kurs güncelleme, silme,
- Kursiyer başvurularını görüntüleme, yeni kursiyer-kurs ön başvurusu gerçekleştirme, silme,
- Hayat Boyu Öğrenme Genel Müdürlüğü tarafından yayınlanacak yeni kursları sisteme dâhil etme,
- Hayat Boyu Öğrenme Genel Müdürlüğü tarafından kaldırılan kursları sistemden silme,
- Kursların açılacağı sınıflara ilişkin planlama işlemleri,
- Ajanda oluşturma ve yayınlama,
- Sistem kullanıcısı oluşturma, silme ve güncelleme,
- Sisteme kayıtlı kurs ve kursiyerlere ilişkin istatistikî bilgiler görüntüleme, iş ve işlemleri yapılacaktır.

Yönetim sayfasının genel tasarımı CSS, Bootstrap, jQuery teknolojileri ile AdminLTE teması kullanılarak tüm tarayıcılara uyumlu olacak şekilde gerçekleştirilmiştir.

Sayfa, *main*, *kenar\_menu*, *content* ve *footer* olmak üzere üç bölüme ayrılmış, *content* bölümü her bir sayfa için özel olarak tasarlanarak kodlanmıştır. *main*, *kenar\_menu* ve *footer* bölümleri ise ortak kullanım amaçlı tasarlanmıştır. Bu bölümler yönetim paneli içerisinde kullanılan tüm sayfalara;

```
<?php$this->load-
>view('dependencies/kenar_menu');?>
<?php $this->load->view('dependencies/main');?>
<?php$this->load->view('dependencies/footer');?>
```

kodları kullanılarak dahil edilmiş olup, yeni oluşturulacak tüm sayfalarda da aynı işlem yapılacaktır.

*yonetim.php* sayfası projenin “Anasayfa” sını ifade etmekte olup, bu sayfada sisteme kayıtlı kurs, kursiyer ve yöneticilere ilişkin istatistikî bilgiler bulunmaktadır. Ayrıca, kurslarla ilgili istatistikî bilgilerin bulunduğu bölümde 10 kayıttan sonraki kayıtlar “*sayfalama*” yöntemi ile dinamik olarak sayfanmaktadır.

Sayfa içeriği iki bölümden oluşmaktadır. Birinci bölümde yönetici, kursiyer ve sistemde tanımlı olan kurslara ilişkin istatistikî veriler, ikinci bölümde ise başvuru yapılan kurslara ilişkin kursiyer sayılarına ait verileri bulundurmaktadır.

Birinci bölümde, kayıtlara ilişkin veriler, *yonetim.php* denetçi ve *yonetim\_model.php* model sayfaları kullanılarak veri tabanındaki *basvuru* ve *kurstanim* tablolarından çekilecektir.

Bu sayfada *\$kayitsayisi*, *\$basvurular* değişkenleri doğrudan görünüm sayfası içerisinde kullanılmıştır. Klasik yöntemde bu değişkenlerin tüm sayfalarda tanımlanması gerekmektedir. Burada, CodeIgniter’in sağladığı esnek yapı ile denetçi içerisinde tanımlanmış olan bu değişkenlere doğrudan görünüm sayfasından erişilebilir.

Yine `$this->pagination->create_links();` kodu ile de denetçi sayfasında özellikleri belirtilen ve CodeIgniter yapısı içerisinde `/system/libraries` dizininde bulunan `Pagination.php` dosyasını `construct()` metodunda çağırarak “*sayfalama*” olayını gerçekleştirilmiştir.

Ayrıca `$data[]` dizisinin her bir elemanı *yonetim.php* görünüm sayfasında kullanılacak değişkenleri ifade etmektedir. `$this->yonetim->count_all();` gibi sorgular ile *Yonetim\_model.php* model sayfası içerisindeki veri tabanı işlemleri ile ilişkili fonksiyonlar çalıştırılmakta ve model sayfasından geri döndürülen veriler denetçi aracılığı ile görünüm sayfasına taşınmaktadır.

## 2.6. Kurs Tanımlama Paneli Tasarımı

Bu sayfa, kursiyerlerin seçebileceği kursların Kurs-Ön Kayıt sayfasında yönetici tarafından oluşturulmasını sağlayan; “*Kurs Tanımlama Paneli*”, Hayat Boyu Öğrenme Genel Müdürlüğü tarafından yayınlanacak yeni kursların sisteme dâhil edilmesini ve bu kursların Kurs-Ön Kayıt sayfasında tanımlanabilmesini sağlayan; “*Kurs Listesine Yeni Kurs Ekle (Sistemde Yoksa) Paneli*” ile Hayat Boyu Öğrenme Genel Müdürlüğü kaldırılan kursları sistemden silme işlemlerini içeren; “*Kurs Listesinden Kurs Sil Paneli*” olmak üzere üç bölümden oluşmaktadır.

Tüm panellere ilişkin işlemler yine MVC (Model-View-Controller) yapısı ile veri tabanındaki *kurstanim* ve *kurslar* tabloları kullanılarak gerçekleştirilmiştir.

Oluşturulan *kurs\_tanim.php* görünüm sayfasında tanımlanan `OnKosul()`, `kurs_ekle()`, `ajax_add()`, `KursAdi_by_Id()`, `ajax_sil()` fonksiyonlarının kullanacağı verilere ilişkin tanımlar *Kurslar.php* denetçisinde işlenerek *Kurslar\_model* sayfasına gönderilmiştir.

*Kurslar\_model.php* ve *Kurslar.php* sayfaları kurs tanımları ve listeleme işlemlerinin yapıldığı bütün sayfalarda referans olarak tanımlanarak kullanılacaktır.

## 2.7. Tüm Başvurular Paneli Tasarımı

Kurs-Ön Kayıt (Başvuru) ekranından kurs başvurusu yapan kursiyerlere ilişkin verilerin yer aldığı sayfadır.

Bu sayfa bir veri tablosu olmakla birlikte, başvuru yapan kursiyerlere ilişkin silme ve güncelleme işlemleri de yapılabilecektir. Aynı zamanda kurum yöneticileri sayfada bulunan ilgili butonlar aracılığı ile başvuru yapılan kurslara ilişkin .xls, .pdf formatında çıktı alabilecek, verileri yazdırabilecektir. Olası hatalara karşı veri tabanına kursiyer bilgilerinin eklenebilmesi amacıyla da .csv formatında tüm kursiyerlere ilişkin bilgiler indirilebilecektir.

Bu sayfada veri tabanında bulunan *basvuru* tablosu ile kurs bilgilerini alırken *kurslar* tabloları etkin olarak kullanılmıştır.

Çıktı işlemleri jQuery kodları ile gerçekleştirildiği için yalnızca görünüm sayfasında gerçekleştirilen tanımlamalar gösterilmiştir. Bunun için ayrıca /assets/plugins/aktarbtn/ eklentisinde yer alan .js uzantılı javascript dosyaları görünüm sayfasına dahil edilmiştir.

Sayfada hem kurs hem de başvuru yapan kursiyerlere ilişkin verilerin bulunması nedeniyle;

```
$this->load->model('basvuru_model', 'basvuru');
$this->load->model('tanimlikurslar_model', 'tanimlikurslar');
```

kodlarının construct (kurucu) içinde kullanılmasıyla iki model sayfası kullanılarak bu sayfa üzerinde işlemler gerçekleştirilmiştir.

## 2.8. Tanımlı Kurslar Paneli Tasarımı

Kurs tanımlama panelinde kursiyerlerin başvuru yapmaları için oluşturulan kurslara ait veri tablosunun bulunduğu sayfadır.

Bu sayfanın oluşturulmasındaki amaç, kursiyer başvuru ekranında görüntülenen kurslarla ilgili herhangi bir değişiklik (güncelleme), başvuru ekranından kaldırma (silme), tanımlı kurslara ait çıktı alabilme (.pdf, .xls, .csv), yazdırma ve listeleme işlemlerinin yapılmasını sağlamaktır. Bu sayede sistem yöneticisi doğrudan veri tabanına müdahale edebilecek, hata olması ihtimaline karşın önlem alabilecektir.

Sayfada gerçekleştirilen temel işlemler yine MVC yapısı ile bir önceki konularda bahsedildiği gibi gerçekleştirilmiştir. Çıktı alma işlemi ise, 2.7’de anlatıldığı üzere tanımlanmıştır.

## 2.9. Kurs-Kursiyer Paneli Tasarımı

Çalışmanın ana hedefini oluşturan kurs ön başvuru işlemi, herhangi bir kursa ön başvurusunu yapan kursiyer adaylarının o kurs hakkında kesin kayıt görüşmesine çağırılması sonrası kurslara kesin kayıtların yapılmasını sağlamaktadır.

Bu bağlamda, herhangi bir kursa ön başvuru yapan kursiyerlere yapılan kesin kayıt toplantısı sonrasında tekrar görüşme daveti gönderilmesinin engellenmesi için,

bu kursiyerlerin ön kayıtlarının tamamlanmasını sağlayarak sistem üzerinden yönetici tarafından belirlenecek kriterlere göre silinmesi işlemidir. Bu işlemi gerçekleştirecek sayfa ise Kurs-Kursiyer İşlemleri menüsü altında bulunan Kurs-Kursiyer Paneli’dir.

Bu panelde kurum yöneticileri herhangi bir kurs için kesin kaydı tamamlanan kursiyerleri başvuru tarihi ve grubuna göre listeleyerek o kurstan kalıcı olarak silebilecektir. Aynı zamanda yönetici bu sayfa üzerinde listelediği verileri rapor alabilecektir.

Sayfanın temel yapısı arama, süzme işlemleri üzerine inşa edilmiş olup, bu işlemler MVC yapısı ile AJAX ve PHP kodları kullanılarak gerçekleştirilmiştir. JQuery komutları *kurs\_kursiyer.php* görünüm sayfası üzerinde tanımlanmış ve form elemanlarından gelen verileri işleyerek denetçiye göndermektedir.

*kurs-kursiyer.php* isimli görünüm sayfasında bulunan

```
"url": "<?php
Echo
site_url('Basvuru/kurs_kursiyer_liste')?>"
```

komut satırı yazılan AJAX kodlarının hangi denetçi üzerinde hangi fonksiyonun çalıştığını göstermektedir.

*Basvuru.php* denetçisi kursiyer işlemlerine ait denetimlerin bulunduğu sayfa olması nedeniyle kursiyerle ilgili tüm denetimlere ait işlemler burada tanımlanmıştır. Kurs-Kursiyer Paneli için de *Basvuru.php* denetçi sayfasındaki *kurs\_kursiyer\_liste()* metodu kullanılmıştır.

Veri tabanı işlemleri için *basvuru* ve *kurstanim* tabloları oluşturulmuştur. Bu tablolarla ilgili işlemlerin yapıldığı model sayfaları ise *basvuru\_model.php* ve *tanimlikurslar\_model.php* sayfaları olup, *Basvuru.php* denetçisine ait kurucuda tanımlanmıştır.

## 2.10. Kurs-Kursiyer Paneli Tasarımı

Kullanıcı İşlemleri menüsü sistem yöneticilerinin sistem kullanıcı bilgilerini görüntüleyebildiği ve yönetebildiği bölümdür. Bu menü *Profil Bilgileri* ve *Yeni Kullanıcı* isminde iki alt menüden oluşmaktadır.

Profil Bilgileri panelinden sisteme giriş yapan kişiye ait bilgiler bulunmakta ve bazı bilgiler üzerinde güncelleme işlemleri yapılabilecektir.

Yeni Kullanıcı panelinden ise yeni sistem kullanıcısı oluşturularak, yönetici olarak giriş yapabilme imkânı sağlanacaktır. Bu panel üzerinde “*Yeni Kullanıcı Oluşturma*” ve “*Kullanıcı Silme*” işlemleri gerçekleştirilebilmektedir. Bu kısımda yapılan veri tabanı işlemleri ve kodlama yapısı Javascript mod eklentisi kullanılarak gerçekleştirilmiş olup, kullanılan AJAX ve PHP kodları Tüm Başvurular Paneli’ndeki komut yapısı ile benzerlik göstermektedir. Fakat bu bölümde diğer bölümlerde bahsedilen ve AJAX ile gerçekleştirilen veri tabanı işlemlerine ek olarak, resim dosyası yükleme

işlemine sağlayabilmek amacıyla `FormData` nesnesi kullanılmıştır.

Resim yükleme işlemi için kullanılan denetçi yapısı ise `CodeIgniter`'a ait dosya yükleme sınıfı ile gerçekleştirilmiştir.

### 2.10.1. AJAX İle Veri Tabanına Nesne Gönderme

HTML5'ten önce, jQuery için AJAX dosya yüklemelerini uygulamak için bir çok yöntem ve eklenti bulunmaktadır. HTML5 ile birlikte dosya yüklemelerini basitleştirebilen `FormData` sınıfı gelmiştir.

Form elemanları arasında nesne niteliğinde olanlar var ise, (dosya, resim vb.) bu tür değişkenlerin veri tabanına gönderilebilmesi için `FormData` nesnesi kullanılmalıdır. `FormData` nesnesi ile herhangi bir eklentiye ihtiyaç olmadan form elemanları üzerinde işlemler yapılabilmektedir. Bunun için aşağıdaki üç durum mutlaka kurulmalıdır:

- `FormData` örneği oluşturulduğunda, form yerine `form[0]` formunu iletir. Bu ifade, formdaki elemanların form nesnesi olduğunu jQuery ifade olmadığını göstermektedir.
- `contentType: false` olarak seçilmelidir. Öyle ki, bu ifade jQuery kodlarına başlık eklenmemesi gerektiğini bildirmektedir.
- `processData: false` olarak seçilmelidir. Bu ifade formdan gelen nesne tipindeki elemanların (resim, dosya vb.) string tipe dönüştürülmesini engelleyecektir [3].

Bu durumda örnek doğru kullanım şekli aşağıdaki gibi olacaktır:

```
var fd = new FormData();
fd.append( 'file', input.files[0] );
$.ajax({
  url: 'http://example.com/script.php',
  data: fd,
  processData: false,
  contentType: false,
  type: 'POST',
  success: function(data){
    alert(data);
  }
});
```

### 2.10.2. Dosya Yükleme Sınıfı

Dosya yönetimi çoğu web uygulaması için önemlidir. Bir içerik yönetim sistemi geliştirilecekse, görüntüleri, pdf raporlarını vb. yükleyebilmek gerekmektedir. `CodeIgniter` Dosya Yükleme sınıfı, bunların tümünün yapılmasını kolaylaştırır.

`CodeIgniter`'da dosya yüklemede iki ana bölüm vardır. Ön uç ve arka uç. Ön uç, form giriş türü dosyasını kullanan HTML formu tarafından işlenir. Arka uçta, dosya yükleme kitaplığı gönderilen girişi formdan işler ve yükleme dizinine yazar [4].

Diğer kütüphanelere benzer şekilde, tercihlere göre nelerin yüklenmesine izin verildiğini kontrol edebilmektedir. Oluşturulan denetleyicide aşağıdaki tercihler belirlenmiştir [2]:

```
$ config [ 'upload_path' ] = './uploads/' ;
$ config [ 'allow_types' ] = 'gif|jpg|png' ;
$ config [ 'max_size' ] = '100' ;
$ config [ 'max_width' ] = '1024' ;
$ config [ 'max_height' ] = '768' ;
$ this -> load -> library ( 'upload',$config );
$ this -> upload -> initialize ( $ config );
```

Yine bu bölümde veri tabanındaki *kullanici* tablosu kullanılmıştır. Bu tablo ve görünüm sayfası olan kullanıcı profil bilgilerinin görüntülediği ve güncellendiği *profilbilgi.php* sayfası arasındaki veri tabanı işlemleri *Kullanici\_model.php* modelinde tanımlanmıştır.

*Kullanici\_model.php* sayfası, *profilbilgi.php* görünümü üzerindeki işlemleri gerçekleştirmek üzere hem *ProfilBilgi.php* hem de *Yenikullanici.php* denetçi sayfaları için ortak kullanılmıştır.

### 2.10.3. CodeIgniter Şifreleme İşlemleri

Sistem yöneticilerine ait şifreler veritabanındaki kullanıcı tablosunda saklanmaktadır. Projede, şifrelerin saklanması, BCRIPT işlemi ile daha güvenli olarak yapılmaktadır. BYCRPT ile günümüzün en güvenli şifre saklama yöntemlerinden biri olan tuzlama (salting) işlemi yapılmıştır. Bunun için *Yenikullanici.php* denetçi sayfasında aşağıdaki komut satırı kullanılmıştır.

```
'kullaniciSifre' => password_hash($this->
input->post('kullaniciSifre'),PASSWORD_BCRYPT
```

#### 2.10.3.1. BYCRIPT Tanımı

`PASSWORD_BCRYPT`, `CRYPT_BLOWFISH` algoritmasını kullanır ve 60 karakterlik bir string döndürür. Her zaman 60 karakter genişliğindeki "\$2y\$" şifreleme formatını kullanan karma bir sonuç verir.

Desteklenen Seçenekler:

- Tuz (Salt): Her şifre için manuel olabildiği gibi, default olarak bir tuz (salt) üretir.
- Maliyet (Tamsayı): Donanımına bağlı olarak arttırabilir, kullanılması gereken algoritmik maliyeti ifade eder, varsayılan değer 10'dur [5].

## 2.11. Yönetici Giriş (Login) Sayfası Tasarımı

Yönetici Giriş (Login) sayfası sistem yönetici olarak tanımlanan kullanıcıların yönetim paneline giriş işlemi için kullanılacaktır.

Bu sayfada öne çıkan özellikler;

- Google CAPTCHA doğrulama ile yönetim paneline güvenli giriş sağlanmıştır.
- Session işlemleridir.

## 2.12. CodeIgniter – Google reCAPTCHA İşlemleri

Öncelikle, uygulama ile Google’ın hizmeti arasındaki iletişime yardımcı olacak güvenlik anahtarlarına sahip olunması gerekir. Bunun için aşağıdaki adresten;

<https://www.google.com/recaptcha/intro/index.html>

site kaydedilerek, Google Site Anahtarı ve Google Gizli Anahtar alınmıştır.

MVC yapısındaki CodeIgniter projesinde görünüm sayfasına captcha bölgesini alacak olan api.js dosyası eklenmiştir.

Ayrıca, formun içinde, captcha bölgesi için yer tutucusu olan div, alınan Google Site Anahtarına eşit “data-sitekey” e sahip olarak sunulmalıdır.

```
<div class="g-recaptcha" id="g-recaptcha-response" name="g-recaptcha-response" data-sitekey="*****"></div>
```

Denetçi sayfada ise gidip ziyaretçinin form doğrulama kurallarına uyup uymadığı kontrol edilmelidir [6].

## 2.13. CodeIgniter Session İşlemleri

Oturumlar, kullanıcının etkinliklerini sayfadan sayfaya bilmek istediğinizde genellikle kullanışlıdır. Örneğin, web sitesinde korumalı bir alana sahip olduğunuz varsayılırsa, kullanıcıların her sayfada oturum açmasına gerek yoktur. Kullanıcının bir kez oturum açmasına ve ayrıntılarını bir oturum değişkeninde saklamasına izin verdikten sonra aynı veriler başka isteklerde yeniden kullanılabilmektedir.

Alternatif olarak, CodeIgniter, verileri bir sonraki istekte yalnızca bir kez kullanılabilir duruma getirmek için de oturumlar kullanır. Bu, bir veritabanı kaydının düzenlediği ve güncellediği, başka bir sayfaya yönlendirildiğinde kullanıcıya geri bildirim verilmek istendiğinde faydalıdır. CodeIgniter oturum yönetimi için session kütüphanesini kullanır [7].

Bu çalışmada kullanılan session kütüphanesine ait durumları ve kullanımları şöyle olacaktır:

- a) Oturum Başlatma: Oturumlar genellikle her sayfa yüklemesinde genel olarak çalışır, bu nedenle Session sınıfı denetleyici kurucularınızda başlatılmalıdır veya sistem tarafından otomatik olarak yüklenebilir. Çoğunlukla, oturum sınıfı arka planda katılsız çalışır, bu nedenle yalnızca sınıfı başlatmak, gerektiğinde oturumları okumasına, oluşturmasına ve güncellemesine neden olur.

Session sınıfını denetleyicide bulunan kurucuda manuel olarak başlatmak için aşağıdaki yöntemin kullanılması gerekir.

```
$this->load->library('session');
```

- b) Oturum Verisi Alma: Oturum dizisinden elde edilen her türlü bilgiye \$\_SESSION süper küreseli üzerinden ulaşılabilir. Bunun için CodeIgniter birkaç farklı yöntem sunar. Bu çalışmada oturum verilerini almak için aşağıdaki komut satırı kullanılmıştır.

```
$ this -> session -> userdata ( 'item' );
```

- c) Oturum Verileri Ekleme: Herhangi bir kullanıcı sisteme giriş yaparak kimlik doğrulaması yapıldıktan sonra, kullanıcı adı ve e-posta adresi oturuma eklenebilir, böylece ihtiyaç olduğunda bir veritabanı sorgusu çalıştırılmadan bu veriler genel olarak erişilebilir hale getirilmektedir. \$\_SESSION komutu ile herhangi bir diğer değişkende olduğu gibi diziye de veri atanabilmektedir. Alternatif olarak \$this->session yöntemi de kullanılabilir. Ancak, yeni bir yöntem olan ve bu çalışmada da kullanılan, veri içeren bir diziyi set\_userdata() yöntemiyle aktarmak;

```
$ this -> session -> set_userdata ( 'some_name' , 'some_value' );
```

komut satırı ile gerçekleştirilmiştir.

- d) Oturum Verilerini Kaldırma: Sistem yöneticilerin sistemden güvenli bir şekilde çıkış yapabilmelerini sağlamak için oturum verilerinin çıkış işleminden sonra silinmesi gerekir. Bunun için aşağıdaki komut satırı kullanılmıştır.

```
$ this -> session -> unset_userdata ( $ array_items );
```

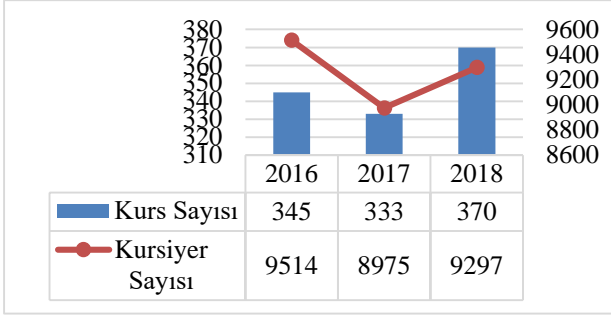
- e) Oturum İle İlgili Mesaj Görüntüleme: CodeIgniter “flashdata” veya yalnızca bir sonraki istek için kullanılacak oturum verilerini destekler ve ardından otomatik olarak temizlenir. Bu, özellikle bir defaya mahsus bilgilendirme, hata veya durum mesajları için çok yararlı olacaktır. Bu çalışmada, oturum ile ilgili mesajları görüntülemek için aşağıda bir örneği gösterilen komut satırı kullanılmıştır.

```
$this->session->set_flashdata('msj','Geçersiz Kullanıcı Adı/Şifre');
```

Buna ek olarak TempData sınıfı da yine bilgilendirme işlemleri için kullanılan bir başka bileşendir [8].

## 2.14. Ajanda Paneli Tasarımı

Çalışmanın yapıldığı kurumun son üç yıldaki faaliyetleri incelenmiş istatistik ve değerlendirmelere göre yılda ortalama 1.100 kurs açıldığı tespit edilmiş ve açılan bu kursların ortalama 350 tanesi 24 derslikli merkez binasında yapılmaktadır. Bina içerisinde yapılan bu kurslara, yıllık ortalama 9262 vatandaş katılım sağlamaktadır. Yıllara göre bina içerisinde açılan kurs ve kurslara katılan kursiyer sayısı Şekil 2’de gösterilmiştir.



Şekil 2. Kurs-Kursiyer Verileri (Son 3 Yıl)

Bu verilere göre bina içerisindeki kurs ve kursiyer sirkülasyonunun çok fazla olduğu gözlemlenmiştir (Şekil 2). Buna göre kurumun sınıf/derslik ve etkinliklerinin düzenli olarak planlanması, herhangi bir aksaklık ve çakışma olmadan dersliklerin kullanılabilmesi amacıyla ajanda paneli geliştirilerek, kurum yöneticilerinin kuruma ait her türlü planlamayı bu ajanda üzerinde yapmaları imkânı sağlanmıştır.

Bu panel, ajanda.php view sayfasında jQuery ve denetçi ile model sayfalarında ise PHP ve CodeIgniter komut yapısı kullanılarak geliştirilmiştir. Panelin geliştirilmesinde fullcalendar.io isimli Javascript Event Calendar uygulamasından faydalanılmıştır.

### 2.15. CodeIgniter İle Log Kaydı Tutma

Bilişim sistemlerinde tehditlerin belirlenmesi ve kötücül aktivitelerin tespit edilebilmesi için kullanılacak veri kaynaklarından birisi de log dosyalarıdır. Bilişim sistemlerinde kullanılan birçok cihaz ve sistem üzerinde meydana gelen olayları kayıt altına alırlar. Bu log kaydı, zamana bağlı olarak hangi kullanıcının hangi aktiviteyi yaptığını gösterir. Log kayıtları bir uçağın kara kutusuna benzer bir şekilde işlev gören text dosyalarıdır. Farklı formatlardaki log kayıtları, üzerinde buldukları sistem ile ilgili birçok bilgiyi tutmaktadır. Bu bilgiler ile saldırı tespiti, adli bilişim süreç analizi, performans analizleri, adli olayların aydınlatılması vb. mümkün olabilmektedir [10].

CodeIgniter ile log kaydı tutmak için yapılması gereken, application/config/config.php dizininde bulunan log bölümünde konfigürasyon ayarlarının yapılmasıdır. Konfigürasyon ayarları yapılırken, CodeIgniter kullanıcısına hangi tür verilere ilişkin log tutulması konusunda 5 seçenek sunmaktadır. Bunlar;

- | 0 = Disables logging, Error logging TURNED OFF
- | 1 = Error Messages (including PHP errors)
- | 2 = Debug Messages
- | 3 = Informational Messages
- | 4 = All Messages

olarak ifade edilmektedir.

Hangi tür verilere ilişkin log kaydı tutulacağı belirlendikten sonra Log kaydı için konfigürasyon ayarları şöyle olacaktır:

```
$config['log_threshold'] = 2;
$config['log_path'] = '';
$config['log_date_format'] = 'Y-m-d H:i:s';
```

Burada, log\_path boş bırakıldığı takdirde default olarak application dizini içerisinde bulunan logs klasöründe kayıtlar tutulacaktır.

### 3. BULGULAR

Şu anda kullanılan başka bir kurs ön kayıt programı farklı platformlarda ve online çalışacak şekilde tasarlanmamıştır. Bu da, kursiyer adaylarının başvuru için kurum binasında gelmelerini zorunlu kılmaktadır. Projenin tüm tarayıcılarda çalışabilir tasarımı ve web üzerinde çalışabilecek şekilde programlanması ile pc, cep telefonu veya başka herhangi bir platformdan, evden, işten kısacası internet erişimi olan her yerden kurs başvurusu yapılabilmektedir. Kursiyer adaylarının sağlıklı bir şekilde başvuru yapmalarını sağlamak amacıyla, Kurs-Ön Kayıt sayfasında geliştirilen dokunmatik ekran klavyesi ile başvuru işlemleri dokunmatik ekranlarda da gerçekleştirilebilecek durumdadır. Kurs-Ön Kayıt sayfasından kursiyer adayı başvuru yapabilmektedir, buradaki veriler sağlıklı bir şekilde veri tabanına kaydedilebilmektedir.

Yöneticiler, Kurs Yönetim Paneli'ni kullanarak kurs ve kursiyerler ile ilgili istatistiklere ulaşabilmektedir. Yöneticiler kursiyerlerin başvurabileceği kursları sisteme tanımlayabilecektir. Tanımlanan kurslar üzerinde güncelleme ya da silme işlemleri yapabilecektir. Aynı zamanda .xls, .csv, pdf formatında bu kurs bilgileri kaydedilebilmekte olup, yazıcıdan çıktı alınabilmektedir. Hayat Boyu Öğrenme Genel Müdürlüğü'nün tanımladığı yeni kurslar olduğu takdirde bu kurslar yönetici tarafından veri tabanına eklenebilmektedir. Başvuru yapan kursiyerlere ilişkin bilgiler yönetici tarafından güncellenebilmekte, kursiyer başvurusu ihtiyaca göre silinebilmekte ve yeni kursiyer ön başvurusu yönetici ekranından da gerçekleştirilebilmektedir. Kurum yöneticilerinin talepleri doğrultusunda başvuru yapan kursiyerlere ait başvuru bilgilerine göre filtreleme yapılabilmekte ve kesin kayıt işlemleri yapıldıktan sonra sistemden silinmeleri sağlanmaktadır. Yine bu ekranda .xls, .csv ve pdf formatında raporlama yapılabilmektedir. Başvuru yapan kursiyerlere ilişkin bilgiler .xls, .csv, pdf formatında kaydedilebilmekte ve raporlama yapılabilmektedir.

Ana giriş sayfasındaki Kursiyer Giriş Ekranı'na tıklayınca gelen Anketimize Katılın sayfasında kurum tarafından belirli aralıklarla memnuniyet anketleri yayınlanmaktadır.

Projenin kullanılacağı kurumdaki derslik ve etkinliklerinin düzenli olarak planlanması, herhangi bir aksaklık ve çakışma olmadan dersliklerin kullanılabilmesi amacıyla ajanda paneli geliştirilerek, kurum yöneticilerinin kuruma ait her türlü planlamayı bu ajanda üzerinde yapmaları imkânı sağlanmıştır. Yönetim paneline sistem kullanıcıları kendi fotoğrafları ile



tanımlanabilmekte ve yönetim panelini şifre sahibi yöneticilerin kullanması sağlanmaktadır.

#### 4. TARTIŞMA VE SONUÇ

Proje, yaygın eğitim kurumlarının özellikle kurs ön kayıt aşamasındaki ihtiyaçlarını gidermek, ön kayıt yapan kursiyerlerin sistem üzerinden listesini alarak bilgilendirme toplantısına davet etmek ve yapılan bilgilendirme sonrası kesin kayıtlarını gerçekleştirmek amacıyla oluşturulmuştur. Bu haliyle proje tüm yaygın eğitim kurumlarına uygulanabilir. Ancak, ilgili servis sağlayıcılar ile anlaşarak projeye SMS modülünün de eklenmesi kesin kayıt işlemi için yapılan toplantılara kursiyerlerin davet edilmesinde bir yöntem olarak kullanılan SMS gönderiminden kaynaklanan maliyeti düşürecektir. Özellikle, bu modül kurumları ek maliyetten de kurtarabilir. Tabi burada kursiyerlerin toplantılara davet için neden SMS kullanıldığı, e-posta kullanılmadığı akla gelebilir. 3438 kurs çeşidi ve bu kurslara her yaş ve eğitim grubundan vatandaşların katıldığı düşünüldüğünde e-posta hesabı olmayan çok fazla kişi bulunmaktadır. Bu sebeple SMS bu hususta önemli bir araç olarak kullanılmaktadır.

Bir başka husus ise, proje yaygın olarak kullanılmaya başladığında sistem yöneticileri için rol tanımlarının (Müdür, Müdür Yardımcısı, Memur gibi) da yapılması uygun olabilir. Ancak mevcut durumda ihtiyaç olmadığına karar verilmiştir.

#### TEŞEKKÜR

Çalışmanın geliştirilmesi aşamasında, sahadaki ihtiyaçların tespiti ve çözüm önerileri konusunda bizlere her zaman yardımcı olan, tavsiyelerde bulunan Kütahya Halk Eğitimi Merkezi Müdür Yardımcısı Ahmet ÖZÇELİK'e ve projenin hazırlanması esnasında bizlere sabır gösteren değerli eşlerimiz ve çocuklarımıza teşekkür ederiz.

#### KAYNAKLAR

- [1] İnternet: Millî Eğitim Bakanlığı Hayat Boyu Öğrenme Kurumları Yönetmeliği, [http://hbogm.meb.gov.tr/meb\\_iys\\_dosyalar/2018\\_04/11093946\\_MEB\\_HBO\\_KURUM\\_LARI\\_YYNETMELYYY.pdf](http://hbogm.meb.gov.tr/meb_iys_dosyalar/2018_04/11093946_MEB_HBO_KURUM_LARI_YYNETMELYYY.pdf), 11 Nisan 2018.
- [2] Kandemir Gökhan, (2017). PHP CodeIgniter, Dikey Eksen Yayınları (1. baskı), Ankara, Türkiye.
- [3] İnternet: [https://developer.mozilla.org/en-US/docs/Web/API/FormData/Using\\_FormData\\_Objects](https://developer.mozilla.org/en-US/docs/Web/API/FormData/Using_FormData_Objects), 05 Mayıs 2019
- [4] İnternet: File Uploading, [https://codeigniter.com/user\\_guide/libraries/file\\_uploading.html](https://codeigniter.com/user_guide/libraries/file_uploading.html), 05 Mayıs 2019
- [5] İnternet: Öntanımlı Sabitler, <https://www.php.net/manual/tr/password.constants.php>, 05 Mayıs 2019

- [6] İnternet: <https://developers.google.com/recaptcha/intro>, 05 Mayıs 2019
- [7] Griffiths Adam, (2010). CodeIgniter 1.7 Professional Development, Birmingham, B27 6PA, UK.
- [8] İnternet: Session Library, [https://www.codeigniter.com/user\\_guide/libraries/sessions.html#adding-session-data](https://www.codeigniter.com/user_guide/libraries/sessions.html#adding-session-data), 05 Mayıs 2019
- [9] İnternet: Millî Eğitim Bakanlığı Yaygın Eğitim Kurs Programları, <http://hbogm.meb.gov.tr/modulerprogramlar/?q=>, 05 Mayıs 2019
- [10] Baykara M , Daş R , Tuna G . Web Sunucu Erişim Kütüklerinden Web Ataklarının Tespitine Yönelik Web Tabanlı Log Analiz Platformu. Fırat Üniversitesi Mühendislik Bilimleri Dergisi. 2016; 28(2): 291-302.