

# SUPERBLOG ORTAK KONU

## Giriş

"Superblog yılın ikinci yarısı boyunca düzenli olarak üzerinde çalışacağımız bir proje. PHP ile başlayacağız ve daha sonra JavaScript'e geçeceğiz.

Adından da anlaşılacağı gibi, ya da öyle umuyorum :), 'Superblog' bloglar oluşturmak için hafif bir içerik yönetim sistemidir (CMS) (büyük kardeşi WordPress gibi).

Bu proje boyunca size rehberlik edeceğim. Ama ellerinizi kirletmeniz gerekecek, aksi takdirde pek eğlenceli olmayacak!

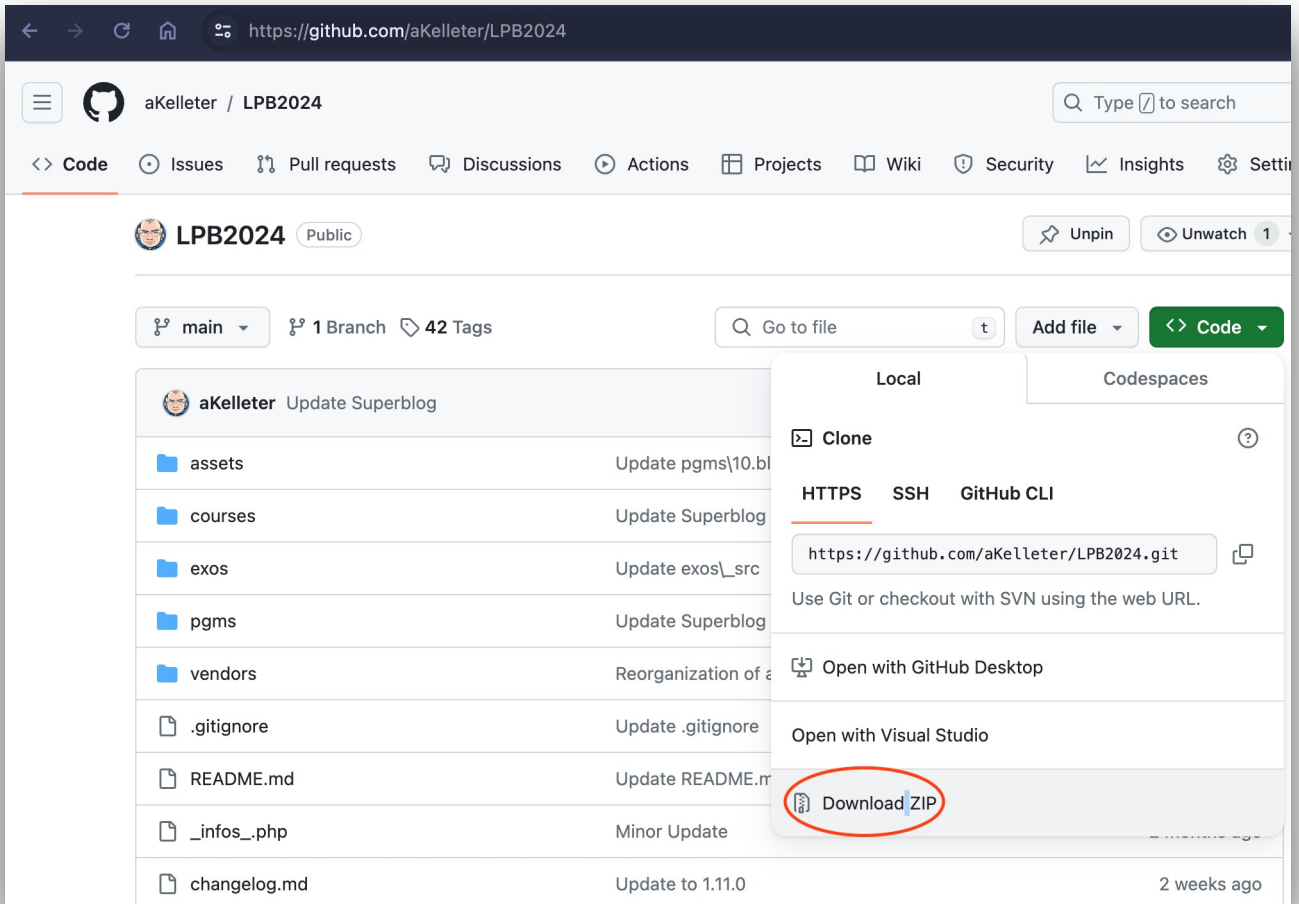
'Süper Blog' projesi için ilk sayfaları ve senaryoları yazmaya başladık, ancak çoğunuz için bu biraz zor oldu.

Bu belge işte bu yüzden var!

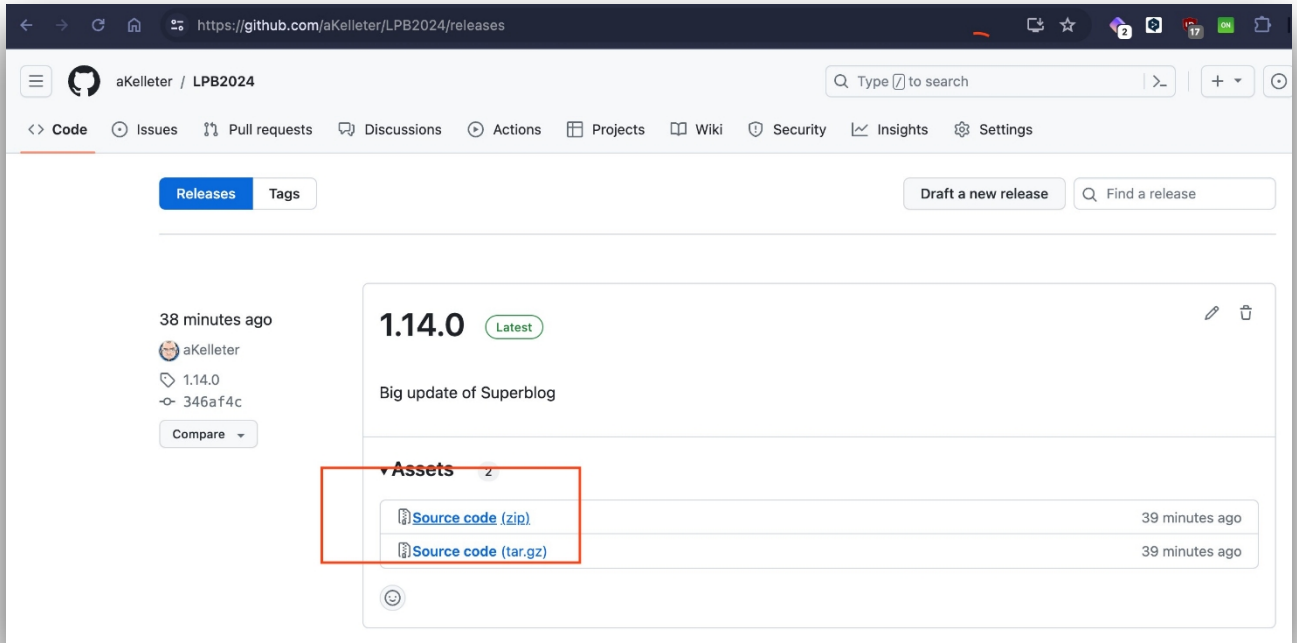
## 1. Öncelikle Superblog kodu LPB2024 uygulamasında, daha doğrusu burada bulunabilir: **LPB2024 / pgms / 10.superblog /**

Muhtemelen şimdiye kadar bildiğiniz gibi, LPB2024'ün kaynak kodu GitHub'da mevcuttur. Başka bir deyişle, projede ilerleme kaydettikçe sürekli güncellenen kaynak koduna erişmek için ya :

- En son güncellemeyi almak için GitHub'ı kullanın.
- GitHub'a gidin ve kaynak kodunu ZIP formatında alın (aşağıdaki resme bakın - "<> Kod" düğmesine (yeşil renkte) tıklayın).



- GitHub'a gidin ve LPB2024'ün en son sürümünü indirin

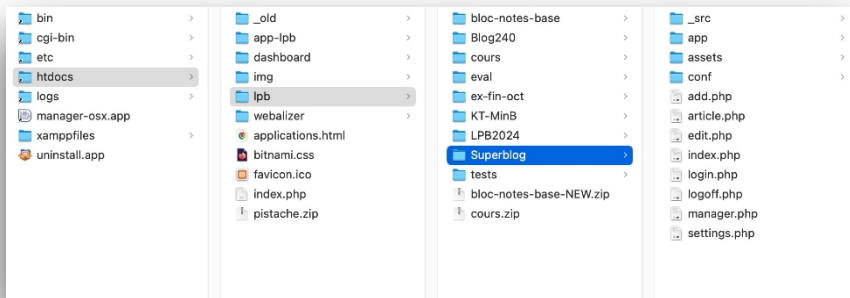


## 2. İkinci olarak, LPB2024'teki Superblog dizininin 10.superlog adında bir kopyasını oluşturmanız gerekir.

Bilgisayarınızda bu tür bir yapı oluşturduysanız, bu kopyayı XAMPP'nin "htdocs" dizinine yerleştirin (bkz. htdocs/lpb).

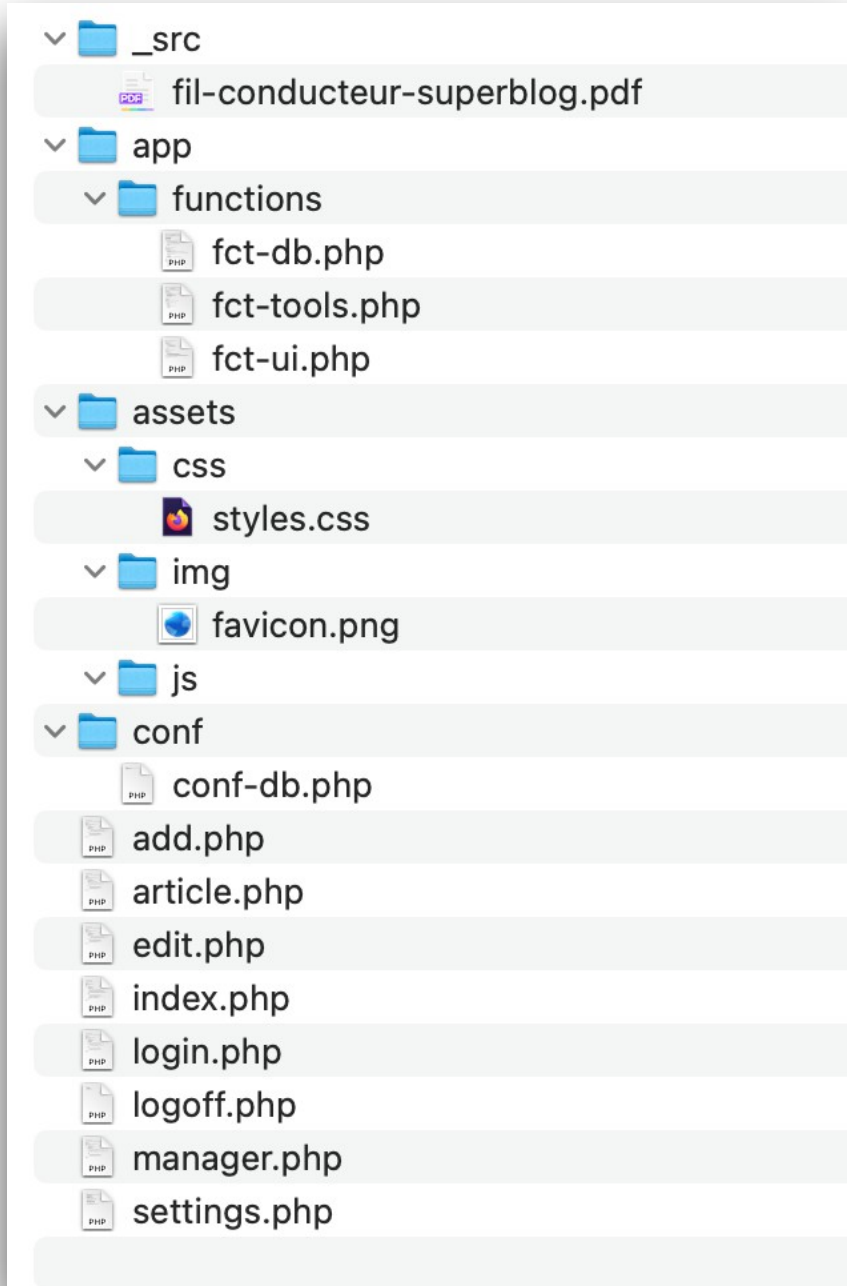
SuperBlog projenizde zaten oldukça ilerlediyseniz, LPB2024 SuperBlog kaynak kodunu kullanmak ve projenize devam etmek zorunda değilsiniz. Ancak ihtiyaç duyarsanız, projedeki belirli öğelerden ve/veya dosyalardan ilham alabilirsiniz.

Eğer çok ileri düzeyde değilseniz ya da bu size güneşli kırsalda bir gezintiden çok haç yolu gibi görünüyorsa, LPB2024'ün Superblog'undan kaynak kodu almanızı tavsiye ederim.



### 3. Yapı ve içerik

Aşağıda projenin temel yapısı yer almaktadır. Elbette zaman içinde gelişecektir.



13/12/2023 tarihindeki yapı

#### \_src/ dizini

Proje için tamamlayıcı dosyalar içerir, ancak bunlar "Superblog" uygulamasının bir parçası değildir.

## **app/ dizini**

Uygulamanın temel dosyalarını içerir. Örneğin, fonksiyon dosyaları ("functions" alt dizininde) ve yakında "classes" alt dizinindeki sınıf dosyaları.

## **assets/ dizini**

"Varlıklar" dediğimiz şeyi içerir. Web sitesi tasarımı ve geliştirme açısından "varlıklar" genellikle uygulama tarafından oluşturulan verileri değil, uygulamanın bir parçası olan metin içeriğini, CSS kurallarını, grafikleri, fotoğrafları, videoları ve ses dosyalarını ifade eder.

## **conf/ dizini**

Superblog kullanıcılarının uygulamalarını çalıştırmak için güncellemeleri gereken yapılandırma dosyasını içerir. Şu anda, Superblog'un bir veritabanına bağlanmasını sağlayan verileri içerir.

## **Projenin kökü /**

Tüm temel dosyaları ve daha doğrusu sayfaları ve belirli uygulama komut dosyalarını içerir.

**index.php** : Bu, URL'de hiçbir şey belirtilmediğinde web sunucusunun aradığı dosya/sayfadır, örneğin www.240am.be. Bu, Blogumuzun ana sayfasıdır ve ziyaretçilere **yayınlanan** makalelerimizin listesini gösterecektir.

**add.php** : Bu, Blogumuza bir makale eklememize izin verecek dosya / sayfadır. Bunu yapmak için, bir form görüntüleyeceğiz ve onun tarafından gönderilen verileri işleyeceğiz.

**article.php** : Bu, bir makalenin tamamını, yani başlığını, içeriğini ve prensip olarak bir veya daha fazla resim gibi diğer öğeleri görüntülemek için kullanacağımız dosya / sayfadır. Bu sayfa, URL'sinde görüntülenecek makalenin kimliğini alacaktır.

**edit.php** : Bu, bir makaleyi güncellememize veya değiştirmemize izin verecek dosya / sayfadır. Add.php sayfasına benzer bir formdan oluşacaktır ve URL'si değiştirilecek makalenin ID'sini içermelidir.

**login.php** : Bu, uygulamanın kimlik dosyası/sayfasıdır. Giriş ve şifre içeren mini bir formdan oluşur.

**logoff.php** : Bu, uygulama oturum kapatma betiğidir.

**manager.php** : Bu, makale yönetim dosyası/sayfasıdır. Index.php sayfasında olduğu gibi, bu sayfada da **henüz yayınlanmamış olanlar da** dahil olmak üzere yazıların listesi görüntülenecektir. Her yazı için, söz konusu yazıyı düzenlemenizi, görüntülemenizi ve silmenizi sağlayacak bağlantıları görüntülemelisiniz.

**settings.php** : Bu, uygulama yapılandırma betiğinin kendisidir. Genel olarak uygulamanın doğru çalışması için sabitleri ve mantıksal öğeleri içerir.

## 4. Dosya yapısı

```
1  <?php
2      require_once('settings.php');
3
4      /**
5       * ICI VOUS ECRIVEZ LE CODE PHP QUI GERE LA LOGIQUE ET LES DONNEES DE L'APPLICATION
6       */
7
8  ?>
9  <!DOCTYPE html>
10 <html lang="en">
11 <head>
12     <?php displayHeadSection('Ajouter un article'); ?>
13 </head>
14 <body>
15     <div class="container">
16         <div id="header-logo">
17             <h1><?php echo APP_NAME; ?></h1>
18         </div>
19         <div id="main-menu">
20             <?php displayNavigation(); ?>
21         </div>
22         <h1>Ajouter un article</h1>
23         <div id="content">
24             <!--
25              Créez ici un formulaire HTML pour ajouter un nouvel article
26              * Astuces :
27                  - L'attribut "action" de votre balise form devra contenir "manager.php"
28                  - C'est dans le fichier manager.php que l'on va traiter les données du formulaire
29                  - L'attribut "method" devra contenir "post"
30              -->
31
32         </div>
33         <footer>
34             <!--
35              Ouvrez une balise php pour lancer la fonction d'affichage
36              du footer. Fonction que vous allez écrire dans fct-ui.php
37              Affichez le nom de l'app sa version sa date de mise à jour
38              et d'autres choses si vous le souhaitez
39              -->
40         </footer>
41     </div>
42 </div>
43 </body>
44 </html>
```

.php dosyaları (sayfaları) her şeyden önce tam bir HTML yapısına **<html></html>** sahip dosyalardır.

**<?php ?>** etiketleri HTML yapısının üzerine yerleştirilir.  
PHP kodunu buraya yazacağımızı gösterir.

PHP kodu için ayrılan sayfaların üzerindeki bu alana, sayfa **denetleyicisi olarak** adlandıracağımız şeyi, yani sayfanın ve genel olarak uygulamanın doğru çalışması için gereken mantığı, verileri ve diğer işlemleri yönetecek kodu yazacağız.

HTML yapısında, mümkün olduğunca yalnızca **<?php displayNavigation(); ?>** gibi HTML kodu döndüren işlevleri çağırarak görüntülemeye çalışıyoruz.

HTML yapısı, yukarıdaki gibi PHP kodunun kısa bölümlerini içerecek ve daha fazlasını içermeyecektir. Tüm iş kodu, mantık ve diğer iş süreçleri sayfanın üst kısmındaki kontrolcüde olmalıdır.

Ekran fonksiyonlarınızı **fct-ui.php** dosyasında oluşturun. Benim kullandığıma benzer veya aynı bir adlandırma kuralını benimseyin, örneğin :

```
function displayArticlesPublies() {}  
function displayArticlesAll(){}
```

Ve böyle devam eder.

## 5. Noel hediyeleri!

Fonksiyon dosyaları, **fct-db.php** ve **fct-tools.php** zaten fonksiyonlarına sahiptir.
























































































































**fct-db.php** dosyası söz konusu olduğunda, tüm işlevler oluşturulmuş ve çalışır durumdadır, bu nedenle bunları yazmanıza gerek yoktur. Dilerseniz, bunlar üzerinde daha önce çalışmış olanlarınız kendi fonksiyonlarını yazmaya devam edebilirler. Bu fonksiyonlara daha sonra geri döneceğiz. Tek yapmanız gereken onları çağırarak kullanmak.

**logoff.php** dosyası da tamamlanmıştır.
























































































**styles.css** dosyası zaten birkaç CSS kuralı içermektedir.

**Bootstrap** CSS çerçevesini projeye entegre etmedim, bu yüzden kullanmak isteyip istemediğinizi size bırakacağım - gerekli değil.

## 6. DB

Objets	articles@blog (localhost)	users@blog (localhost)
  	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 

**Kullanıcılar** tablosundaki alanlar ve veriler

Objets	articles@blog (localhost)	users@blog (localhost)
  	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 
	 	 

**Öğeler** tablosundaki alanlar ve veriler

Öğeler tablosunda tam olarak aynı içeriğe sahip olmayabilirsiniz.

Veritabanı ile ilgili bir sorun yaşarsanız, projenin \_src dizinine bir **blog.sql** dosyası yerleştirdim. XAMPP'nizin MySQL DB motorunda blog DB'sini oluşturmak veya yeniden oluşturmak için kullanabilirsiniz.

Bunu yapmak için [phpMyAdmin](#) web uygulamasına gidin ve sayfanın üst kısmındaki **İçerik Aktar**'a tıklayın, ardından dizinde bulunan dosyayı seçin Proje src: blog.sql. Bitirmek için sayfanın en altına gidin ve **İçerik Aktar** düğmesine tıklayın.



## 7. Peki şimdi ne yapacağım?

Ne soru ama..., tabii ki tüm uygulamaları!

Ama daha ciddiysi, **18/12/2023** Pazartesi günü için:

### Henüz yapmamış olanlar için...

Kursa çalışan bir **index.php** sayfası ile gelmenizi istiyorum, yani bilgisayarınızda oluşturulmuş Blog veritabanındaki yazıları düzgün bir şekilde gösteren bir sayfa (`var_dump()` ile bir dizi değil).

Ham verileri **index.php** sayfasında görüntülemeyi bitirmiştik, ancak **index.php** sayfasında `fetch data` fonksiyonu zaten çağırılmıştı, bu nedenle sayfadaki makale başlıklarını listeleyecek `display` fonksiyonunu yazmanız gerekiyor. Bir DB'nin çıktısı bir dizidir, bu yüzden...

Burada durmayın, eğer yapabilirsiniz, görüntü açısından **index.php** sayfasına benzeyen, ancak bazı ekstra öğeler içeren **manager.php** sayfasını başlatın.

Ve **z a m a n ı n ı z** varsa, sayfanın HTML yapısını zaten oluşturabilirsiniz **login.php**, Zaten bir giriş formu oluşturduk.

Daha fazla zaman :) **add.php** ve **edit.php** sayfalarını başlatın, HTML terimleriyle bunlar formlardır

### Diğerleri için.

**index.php** sayfasını veya diğer sayfaları oluşturmuş olanlarınız, lütfen bu yönde devam edin ve projede ilerleyin.

**HERKESE CESARET**

**MÜMKÜN OLDUĞUNCA MÜSAIT OLMAYA ÇALIŞTIĞIMI UNUTMAYIN, BU YÜZDEN LÜTFEN BENİMLE İLETİŞİME GEÇMEKTEN ÇEKİNMEYİN.**

**AK**