

SuperBlog projesinin değerlendirilmesi

Hafif bir CMS (blog) motoru

Bu belgede, birlikte başlattığımız projeyi sonuçlandırmak için size mümkün olan en iyi temeli sağlamayı amaçlayan proje hakkında bilgi bulacaksınız.

Yapılması gereken ilk şey [LPB2024](#)'ün en son sürümünü indirmektir.

En son sürüm 1.17.0'dır. PGMS bölümüne gidin. Orada projenin temelini 10.superblog adı altında bulacaksınız.

Bu belgenin tamamını bir kez okumanızı ve gerektiğinde tekrar dönmenizi tavsiye ederim.

A. SuperBlog'un şu anki durumu nedir?

10.superblog dizini, uygulamanın düzgün çalışması için gereken tüm dosyalarla birlikte projenin tüm yapısını içerir (belgeye bakın: [fil-conducteur-superblog.pdf](#)). Bu yapıda, verilerimizi yönettiğimiz veritabanını (blog) oluşturmak için gereken iki SQL dosyasını içeren bir `_src/` dizini bulacaksınız:

- `blog.sql`: blog DB'sini kullanıcılar için açık metin parolalarıyla oluşturmak veya sıfırlamak için bu dosyayı içe aktarın
- `blog-pwd-hash.sql`: karma kullanıcı parolaları ile blog oluşturmak veya DB blog oluşturmak için bu dosyayı içe aktarın.

Bu belgenin yanı sıra "fil conducteur" adlı belgeyi de bulacaksınız.

B. SuperBlog projesinin değerlendirme sözleşmesini tamamlamak için ne yapmam gerekiyor?

1. Makale başlıklarını görüntüleyen bir ana sayfaya ([index.php](#)) ihtiyacınız var **YAYINLANDI**. DB blogumuzun "makaleler" tablosunda kayıtlı makaleler.

2. Ana sayfadaki makalenin başlığına tıkladığınızda, makaleyi tam olarak görüntüleyen yeni bir sayfa ([article.php](#)) açılır, yani d. Başlığı ve içeriği.
3. Bir tanımlama formuna ([login.php](#)) ihtiyacınız var. Bu form uygulamaya giriş yapmak için kullanılır. İçeriği yönetmek ([manager.php](#)), içerik eklemek ([add.php](#)), içeriği düzenlemek ([edit.php](#)) ve içeriği silmek ([manager.php](#)) için tanımlanmanız gerekir. Formun eylem etiketi login.php dosyasının kendisini çağırır ve tanımlama işlemi sayfanın üzerindeki denetleyicide gerçekleştirilir. Kullanıcı tanımlandıktan sonra, header() PHP fonksiyonu kullanılarak site yönetim sayfasına ([manager.php](#)) yönlendirilir
4. Tüm makalelerin (yayınlanmış ve yayınlanmamış) başlıklarını görüntüleyecek bir yönetim sayfasına ([manager.php](#)) ihtiyacınız var. Her yazının başlığının altına 3 bağlantı / düğme eklemeniz gerekir. Bir "Düzenle" düğmesi, bir Ayrıca bir "Oku" düğmesi ve bir "Sil" düğmesi de bulunmaktadır.
5. "Değiştir" düğmesi düzenleme sayfasına ([edit.php](#)) bir bağlantıdır ve bağlantıda değiştirilecek makalenin kimliğini iletiriz. Düzenleme sayfasındaki denetleyiciden kimliği test edip alıyoruz ve kimliğe karşılık gelen makalenin verilerini yüklüyoruz. Alınan verileri değişiklik sayfasındaki forma yerleştirin ve sayfanın altındaki değiştir düğmesine tıklayarak değişikliği doğrulayın. Güncelleme fonksiyonundan gelen dönüşü bağlı olarak bir mesaj görüntülenir.
6. "Sil" düğmesi, şu anda bulunduğunuz sayfa olan yönetim sayfasına ([manager.php](#)) bir bağlantıdır. Bağlantı ayrıca silinecek ögenin kimliğini de içerir. Kimliği test edip alırız ve makaleyi silme işlevini başlatırız. Silme işlevinden geri dönüşü bağlı olarak bir mesaj görüntülenir.
7. "Oku" düğmesi, ana sayfamızdakine ([index.php](#)) benzer bir bağlantıdır. Söz konusu makalenin ID'si ile birlikte makalenin tamamını gösteren sayfaya ([article.php](#)) bir bağlantıdır.
8. Bir makaleyi değiştirmek için kullanılan sayfa ([edit.php](#)). "[manager.php](#)" sayfasına doğru bir şekilde bağlantı verdiyseniz, URL'deki kimliği almalı ve \$_GET['id'] kullanarak makale verilerini

getArticleByIDDB() fonksiyonunu kullanın ve verileri forma yerleştirin. Form eylem etiketi form verilerini **manager.php** sayfasına gönderir ve yönetim sayfası denetleyicisinde aldığınız formu kontrol eder ve güncelleme işlevini başlatırsınız. Fonksiyondan geri dönüşle bağlı olarak bir mesaj görüntülenir.

9. Bir öge ekleme sayfası (**add.php**) kullanıcının doldurduğu boş bir formla başlar ve ardından yeni ögeyi (formdaki verileri) yönetim sayfamıza (**manager.php**) gönderir. Şimdiye kadar anlamaya başlamışsınızdır, yönetim sayfası denetleyicisinde aldığımız formu kontrol ediyoruz (formun altındaki *gizli* giriş etiketini hatırlayın, *formun* adıyla birlikte bir *değer* niteliği vardır (**edit.php** sayfasıyla aynı)) ve ardından veritabanına yeni bir öge ekleyecek işlevi başlatıyoruz. Kullanıcıya yeni bir makale ekleme sonucunu bildiren küçük bir mesaj görüntülemek için fonksiyonun geri dönüşünü kontrol ediyoruz - prensipte iyi haber!
10. Giriş sayfası (**login.php**) bir *e-posta* alanı ve bir *şifre* alanı içeren basit bir formdur. Formun eylem niteliği **login.php** olarak ayarlanmıştır; boş bırakabiliriz, çünkü boş = bulunduğunuz sayfa, başka bir deyişle **login.php**. Sayfanın üst kısmındaki denetleyicide, şimdiye kadar alıştığımız gibi, kullanıcının e-posta ve parola alanlarını doldurup doldurmadığını kontrol etmek de dahil olmak üzere bir dizi test çalıştırıyoruz ve eğer öyleyse, tanımlama işlevini başlatıyoruz. Kullandığınız veritabanına bağlı olarak (düz metin parola veya karma parola), *userIdentificationDB()* veya *userIdentificationWithHashPwdDB()* işlevini kullanacağız. Fonksiyonun geri dönüşünü kontrol edin. Eğer fonksiyon kullanıcının verilerini içeren bir dizi döndürürse, kullanıcı sistem tarafından tanınmıştır ve kullanıcı yönetim sayfasına (**manager.php**) yönlendirilir.
11. Menüdeki logout düğmesi (kırmızı renkte) **logout.php** betiğini çağırır ve mevcut oturumu yok eder, ancak hemen ardından yeni bir oturum oluşturur ve betiğin sonunda ana sayfaya yönlendirir.

Sanırım başvurumuzun tüm sayfalarını ele aldık.

Yıl sonu hediyeleri :)

Son güncellemenin kodunda zaten olduğu için bunu **yapmak zorunda değilsiniz:**

1. **index.php** sayfası: sitenin ana sayfası, 18/12/2023 Pazartesi günü birlikte yaptık
2. **article.php** sayfası: makalenin t a m a m ı n ı görüntüleyen sayfa, bunu da birlikte oluşturduk.
3. **login.php** sayfası: sayfa ve tüm tanımlama s i s t e m i çalışır durumda d ır ve uygulamaya bağlanabilirsiniz (**login.php** veritabanındaki kullanıcılar tablosunda iki e-posta adresi blog: **john@mail.com** veya **alain@mail.com** aynı parolalarla: her ikisi için de 1234).
4. **Add.php** sayfası: Form (HTML) mevcut ve işlevseldir. Form **manager.php** sayfasına ve PHP denetleyicisine gönderilir de çalışır durumdadır (zaten bağlıysa yeniden yönlendirme), birkaç başlatılmış değişken ve basit bir DB bağlantı kontrolü).

Aslında ne yapman gerekiyor?

1. **Edit.php** sayfası: Formun %95'i tamamlandı, sadece yapmanız gerekenler Verileri başlık ve içerik etiketlerine yerleştirin, onay kutusu adlı bir değişkeni başlatmanız koşuluyla, zaten yapılandırılmış veriyi DB'den alan ve PHP denetleyicisini sayfanın en üstünde sonlandıran işlev tarafından alınan verilerle birlikte makale.
2. **Yönetici.php** sayfası: Bu sayfa oluşturulur, yapısına sahiptir HTML ve size yardımcı olacak temel talimatlar v e çok sayıda yorum içeren bir denetleyici. Bitirmeniz ve iade etmeniz gerekiyor tamamen işlevsel.
3. Tüm uygulamanın mükemmel bir ş e k i l d e çalışıp çalışmadığını **kontrol** etmeniz **gerekir**, örn:

- Yalnızca yayınlanmış makaleleri gösteren bir ana s a y f a
- Seçilen makaleyi tam olarak görüntüleyen bir makale sayfası
- Bağlantı kurma yeteneği
- Giriş yaptıysanız farklı bir menüye sahip olun (yönetin ve ekleyin)
- Tüm makaleleri görüntüleyen v e her birini değiştirmenize, okumanıza ve silmenize olanak tanıyan işlevsel bir yönetim sayfası.
- Yeni bir makale ekleyebilmelisiniz
- Bir öğeyi değiştirebilmeniz gerekir
- Bir makaleyi silebilmeniz gerekir
- Kullanıcıya her şeyin yolunda gidip gitmediğini bildirmek i ç i n mesajlar görüntülemeniz gerekir.

Size elimden geldiğince yardımcı oldum. Bu değerlendirme, yıl boyunca yapılan çalışmaların genel değerlendirmesinde (yıl sonu sınavı hariç) %25 civarında bir ağırlığa sahip olacaktır. Umarım hepiniz eksiksiz bir proje sunmayı başarırsınız. Bu belgeyi ve konu başlığını dikkatlice okumanızı ve her şeyden önce dosyalardaki kodu ve yorumları okumanızı tavsiye ederim. Yazdıklarımı ve tabii ki yazdığınız kodu anlamak için elinizden geldiğince çok çalışın.

LPB2024'teki koddan başlarsanız, yazmanız gereken binlerce satır kod olmayacaktır. Kendi kodunuzdan bir projeye başlıyorsanız, LPB2024 SuperBlog'daki kodu ilham kaynağı olarak kullanabilir veya neye rahat olduğunuza veya neye ihtiyacınız olduğuna bağlı olarak kullanmayabilirsiniz.

SuperBlog projenizi en geç 7 Pazar gece yarısına kadar teslim etmelisiniz. Projenizin zip veya rar formatında bir arşivi mükemmel olacaktır.

Veritabanı söz konusu olduğunda, **blog** veritabanını, DB'ye ve login.php sayfa denetleyicisinde seçtiğiniz kimlik türüne bağlı olarak açık metin veya karma parola ile kullanacağım.

İlişkisel veritabanları hakkında temel bir anlayışa sahip değilseniz, şimdiye kadar kullandığımız blog veritabanından başka bir veritabanı, örneğin kişisel bir veritabanı kullanmanızı tavsiye etmem. Bu durumda bana veri tabanınızın SQL formatında bir dışa aktarımını sağlamanız gerekeceğini ve bunun biraz daha zor olabileceğini unutmayın.

LPB2024'teki SuperBlog kodunu kullanmayı önceden bitirdiyseniz, uygulamayı sıfırdan kendiniz yeniden yazmayı deneyin. Temel bilgilerle, ana sayfa ve makale sayfasıyla başlayın ve uygulamanızı azar azar oluşturun, kademeli olarak geliştirin ve ekleyin. Programlama çıraklığınızda bu şekilde öğrenir ve ilerlersiniz.

BANA İHTİYACINIZ OLURSA MÜSAITIM :)

Kişisel e-posta adresim: alain@kelleter.be

Hepinize iyi bayramlar diler, aileniz, arkadaşlarınız ve sevdiklerinizle iyi vakit geçirmenizi temenni ederim. 08 Ocak'ta SuperBlog projesinin ve yeni bir dil olan SuperBlog'un bilgilendirme toplantısında görüşmek üzere.

JavaScript :)