

Laboratuvar: GitHub Kullanarak Çevik Planlama



Tahmini süre: 60 dakika

GitHub Kullanarak Çevik Planlama laboratuvarına hoş geldiniz. Bu laboratuvara, müşteri hesapları mikro hizmet geliştirme projeniz için ilk sprint'e hazırlık olarak bir sprint planı veya Sprint 0 oluşturacaksınız. Plan, hesapları okumak, güncellemek ve silmek için bir hesap hizmeti oluşturmaktır. Hesap hizmeti veritabanı, müşteri isimleri ve adresleri hakkında temel bilgileri içerecektir.

Not: Bu laboratuvar için GitHub'da çalışacaksınız. Laboratuvar ortamını kullanmayacaksınız.

Hedefler

Bu laboratuvar çalışmasında, bir GitHub deposu oluşturacak, bir GitHub Kanban panosu kuracak, bir kullanıcı hikayesi şablonu geliştirecek ve panoya kullanıcı hikayeleri ekleyeceksiniz. Ardından bu hikayeleri sıralayarak geri bildirim için hazırlayacak, ürün geri bildirimini sprint'e hazır hale getirecek ve nihayetinde kapsamlı bir sprint planı oluşturacaksınız.

Teslimatlar

Son projede teslimatınızı sunmak için iki seçenekten olacak.

- **Seçenek 1: AI-Değerlendirmeli Teslimat ve Değerlendirme**
- **Seçenek 2: Eş-Değerlendirmeli Teslimat ve Değerlendirme**

Bu laboratuvar çalışmasında, yanıtlarınızı .jpeg veya .png formatında görüntüler şeklinde ve GitHub depo URL'sini cihazınızda son proje için kaydetmeniz gerekmektedir.

Ekranda görüntüsü almak için, çeşitli ücretsiz ekran yakalama araçlarını kullanabilir veya işletim sisteminize bağlı olarak aşağıdaki kısayol tuşlarını kullanabilirsiniz:

- **Mac:** Tüm ekranınızı yakalamak için klavyenizde Shift + Command + 3 ($\hat{U} + \# + 3$) tuşlarını kullanabilir veya bir pencereyi veya alanın yakalamak için Shift + Command + 4 ($\hat{U} + \# + 4$) tuşlarına basabilirsiniz. Görüntüler, Masaüstü klasörünüzde bir dosya olarak kaydedilecektir.
- **Windows:** Aktif pencerenizi yakalamak için klavyenizde Alt + Print Screen tuşlarını seçebilirsiniz. Bu komut, aktif pencerenizin bir görüntüsünü panoya kopyalar. Ardından, bir resim düzenleyici açın, panodaki görüntüyü resim düzenleyiciye yapıştırın ve resmi kaydedin.

Proje Özeti

Şirketinizdeki müşteri hesap yöneticisi, e-ticaret web sitenizdeki müşterileri takip etmek için bir hesap mikroservisi geliştirmenizi istediler. Bu bir mikroservis olduğu için, diğer mikroservislerin çağrılabileceği iyi yapılandırılmış bir REST API'ye sahip olması bekleniyor. Bu hizmetin başlangıçta müşteri oluşturma, okuma, güncelleme, silme ve listeleme işlemlerini gerçekleştirmesi gerekiyor.

Ayrıca, bu görev üzerinde başka birinin çalışmaya başladığını ve zaten bir veritabanı modeli ile bir müşteri hesabı oluşturmak için bir uç noktaya sahip Python Flask tabanlı bir REST API geliştirdiği belirtildi. Sadece hesapları okumak, güncellemek, silmek ve listelemek için REST API'lerini eklemek üzere bir plan yapmanız gerekiyor. Çevrimiçi bir laboratuvar ortamında çalışacağınız için, o ortamı geliştirme için hazır hale getirmek üzere çalışmanızı planlaması gerekecek.

Alıştırma 1: Bir GitHub deposu oluşturun

Bu alıştırmada, sizin için sağlanacak bir başlangıç şablonunu kullanarak bir GitHub deposu oluşturacaksınız. Yaptığınız değişiklikleri GitHub'a geri gönderebilme ve bunları kaydedebilmek için kendi GitHub deponuzu ihtiyacınız olacak.

Not: Gelistireceğiniz laboratuvar ortamının geçici olduğunu anlamak önemlidir. Kısa ömürlüdür ve herhangi bir zamanda silinebilir. Bu nedenle, tüm çalışmalarınızın GitHub'da kaydedilmesi, kolayca geri yüklenmesi için gereklidir.

Yapılması Gereken Adımlar

1. Projeniz için yeni bir GitHub deposu oluşturun ve adını devops-capstone-project olarak belirleyin. Bu, değerlendircilerin çalışmanızı puanlamak için arayacağı ismidir.

Not: Depo için **Public** seçeneğini seçtiğinizden emin olun ve ardından oluşturun.

- Başlangıç kodu projesini açmak için URL'yi seçin: <https://github.com/ibm-developer-skills-network/aolwx-devops-capstone-template>
 - Bu depo klonlamak için yeşil [Use this template] butonunu kullanın. (Fork kullanmayın; Template butonunu kullanın.)
 - Oluşturulan deponun adını devops-capstone-project olarak belirleyin. Bu, değerlendircilerin çalışmanızı puanlamak için arayacağı ismidir.
2. Depo oluşturulduktan sonra, yeni deponuzda README.md dosyasını açın. Ve projeye devops-capstone-project ismiyle güncelleyin. Capstone projesi hakkında kısa bir açıklama ekleyin. Değişiklikleri kaydedin ve taahhüt edin, böylece güncellenmiş bilgiler deponuzun ana sayfasında görünecektir.

Kanıt

- **Seçenek 1: AI-Değerlendirmeli Gönderim ve Değerlendirme** için, proje adı detaylarını içeren README.md dosyasının **kamuya açık GitHub depo URL'sini** kopyalayın ve yapıştırın.
- **Seçenek 2: Eş-Değerlendirmeli Gönderim ve Değerlendirme** için, depo adının devops-capstone-project olduğunu gösteren ve Kamuya Açık olan **screenshot** alın ve bunu planning-repository-done.jpeg veya planning-repository-done.png olarak kaydedin.

Egzersiz 2: Depo içinde bir GitHub Kanban panosu oluşturun

Bu egzersizde, oluşturulan depo için GitHub Projeleri kullanarak bir kanban panosu kuracaksınız.

1. Aşağıda verilen yedi sütunu dahil ederek GitHub kanban panonuzu oluşturun:

- Yeni sorunlar
- Buzdolabı
- Ürün bekleme listesi
- Sprint bekleme listesi
- Devam ediyor
- Gözden geçirme/QA
- Tamamlandı

Aşağıdaki ekran görüntüsüne başvurun:

2. GitHub kanban panosunu laboratuvar gereksinimlerine göre ayarlarken herhangi bir zorlukla karşılaşırsanız, [Agile Geliştirme ve Scrum'a Giriş](#) kursundan [GitHub'da Kurulum Yapın](#) laboratuvarına başvurabilirsiniz.

Alıştırma 3: Bir kullanıcı hikayesi şablonu oluşturun

Bu alıştırmada, GitHub kanban panonuz için iyi formatlanmış kullanıcı hikayeleri yazmanıza yardımcı olacak bir kullanıcı hikayesi şablonu oluşturacaksınız.

Yapılması Gereken Adımlar

1. Projenizin GitHub deposu için bir şablon oluşturun. Şablonun aşağıda listelenen bileşenleri içerdiginden emin olun. Bu metni kopyalayıp yapıştmak ve ardından düzenlemek isteyebilirsiniz çünkü aynı zamanda şablon için ihtiyaç duyacağınız doğru markdown sözdizimini de içermektedir.

```
**As a** [role]
**I need** [function]
**So that** [benefit]

### Details and Assumptions
* [document what you know]
### Acceptance Criteria
gherkin
Given [some context]
When [certain action is taken]
Then [the outcome of action is observed]
```

2. Deponuzda adı `.github/ISSUE_TEMPLATE` olan yeni bir klasör oluşturduğunuzdan emin olun. Bu klasör, `user-story.md` adlı yeni kullanıcı hikayesi şablonunuzu içerecektir.

Kanıt

- **Seçenek 1: AI-Değerlendirilen Gönderim ve Değerlendirme** için, kullanıcı hikayesi şablonunu içeren `user-story.md` dosyasının **genel GitHub depo URL'sini** kopyalayın ve bir metin dosyasına kaydedin.
- **Seçenek 2: Eş-Değerlendirilen Gönderim ve Değerlendirme** için, yeni oluşturulan `.github/ISSUE_TEMPLATE` klasörünü ve kullanıcı hikayesi şablonunu gösteren ekran görüntüsünü alın ve `planning-storytemplate-done.jpeg` veya `planning-storytemplate-done.png` olarak kaydedin.

Aliştırma 4: Ürün geri bildirimlerinizi bir araya getirin

Bu alıştırmada, e-ticaret capstone projesi için müşterileri hesapları mikroservisine dayalı kullanıcı hikayeleri oluşturacaksınız. Hesaplar servisi, isimler ve adresler gibi temel müşteri bilgilerine sahip olacaktır.

1. Projenizin aşağıdaki her bir adımı için GitHub kanban panonuzda yedi kullanıcı hikayesi oluşturun:

- Geliştirme ortamını kurun
- Servisten bir hesabı okuyun
- Serviste bir hesabı güncelleyin
- Servisten bir hesabı silin
- Servisteki tüm hesapları listeleyin
- Mikroservisini Docker ile konteynerleştirin
- Docker görüntütüzu Kubernetes'e dağıtan

2. Tüm hikayelerinizin **Yeni Sorunlar** pipeline'ında listelendiğinden emin olun.

Kanıt

Hem Seçenek 1: AI-Değerlendirmeli Gönderim ve Değerlendirme hem de Seçenek 2: Es-Değerlendirmeli Gönderim ve Değerlendirme için Kanban panonuzun 'Yeni Sorunlar' sütununda listelenen tüm kullanıcı hikayelerini gösteren bir screenshot alın ve planning-userstories-done.jpeg veya planning-userstories-done.png olarak kaydedin.

Aliştırma 5: Yeni sorunları önceliklendirme

Bu alıştırmada, Yeni Sorunlar hattındaki sorunları inceleyerek **Backlog İyileştirmesi** yapmaya başlayacaksınız ve bunları ne zaman çalışmayı planladığınıza bağlı olarak Ürün Backlog'u veya Buz Kutusuna taşıyacaksınız. Docker ile konteynerleştirme ve Kubernetes'e dağıtım, birkaç sprint sonra yapacağınız bir şey, bu yüzden hemen önemli değil.

Yapılması Gereken Adımlar

1. Hangi kullanıcı hikayeleri üzerinde hemen çalışacağını belirleyin ve bunları Yeni Sorunlar pipeline'ından Ürün Geri Bildirimi pipeline'ına taşıyın.
2. Kalan hikayeleri Yeni Sorunlardan Buz Kutusuna taşıyın, çünkü bunlar üzerinde daha sonra çalışacaksınız.



İpucu için buraya tıklayın.

Kanıt

Hem Seçenek 1: AI-Değerlendirmeli Gönderim ve Değerlendirme hem de Seçenek 2: Es-Değerlendirmeli Gönderim ve Değerlendirme için Kanban panonuzun bir screenshot'unu alın ve planning-productbacklog-done.jpeg veya planning-productbacklog-done.png olarak kaydedin.

Aliştırma 6: Ürün backlog'unuzu iyileştirin

Bu alıştırmada, bir backlog iyileştirme toplantı yapma adımlarını takip edeceksiniz. Ürün sahibi olarak, ürün backlog'unuzu bir sonraki sprint planlama toplantısına hazırlayacaksınız. Bu hazırlığın amacı, tüm hikayelerinizi sprint için hazır hale getirmektir.

Yapılması Gereken Adımlar

1. Product Backlog içindeki tüm hikayelerin "sprint hazır" olarak kabul edilebilmesi için yeterli detaya sahip olduğundan emin olun. "Tamamlandı" tanımını belirlediğinizden emin olmak için **Kabul Kriterlerine** özel dikkat gösterin.
2. technical debt adında bir etiket oluşturun ve buna yellow renk kodu atayın, ardından bunu deposuna ekleyin.
3. Hikayelerinize etiketler atayın. Müşteriye değer katan her şey bir enhancement olarak kabul edilir ve technical debt, geliştiricilerin ihtiyaç duyduğu ancak görünür müşteri değeri sunmayan şeyler olabilir.
4. Product Backlog içindeki hikayeleri en yüksekten en düşüğe öncelik sırasına göre, sırasıyla pipeline sütununda yukarı veya aşağı sürükleyerek sıralayın. Uygulama sırasını düşünün.



İpucu için buraya tıklayın.

Not: Etiketleri eklediyseniz ancak kanban tahtasında göremiyorsanız, lütfen şu adımları izleyin:

- Aşağıdaki ekran görüntüsünde 1 olarak vurgulanın backlog yanındaki açılır menüyü açın.
- 2 olarak vurgulanın Fields seçeneğinin yanındaki ok simgesini seçin.
- Hidden fields listesinden Visible fields listesine Etiketleri sürükleyip bırakın (3 olarak vurgulanan).
- Değişiklikleri kaydetmek için Save seçeneğini seçin.

The screenshot shows the GitHub interface for a project named 'devops-capstone-project'. At the top, there are tabs for 'Backlog', 'Priority board', 'Team items', 'Roadmap', 'In review', 'My items', and '+ New view'. A red box highlights the 'Layout' icon. Below the tabs, there's a 'Filter by keyword' input field. On the left, a sidebar shows 'Backlog' status with a note 'This item hasn't been assigned yet'. The main area is a Kanban board with columns: 'Backlog' (green), 'In progress' (yellow), and 'Done' (grey). A red box highlights the 'Board' tab in the navigation bar. A context menu is open over the 'Backlog' column, with a red box highlighting the 'Fields' section. The 'Visible fields' section includes 'Title', 'Assignees', 'Status', 'Priority', 'Estimate', and 'Labels' (which is also highlighted with a red box). The 'Hidden fields' section includes 'Linked pull requests', 'Milestone', 'Repository', 'Reviewers', 'Start date', and 'End date'. At the bottom of the menu, there are 'Save' and 'Discard' buttons, with 'Save' highlighted with a red box.

Kanıt

Hem Seçenek 1: AI Değerlendirmeli Gönderim ve Değerlendirme hem de Seçenek 2: Eş Değerlendirmeli Gönderim ve Değerlendirme için Kanban panonuzun bir screenshot'unu alın ve planning-labels-done.jpeg veya planning-labels-done.png olarak kaydedin.

Aliştırma 7: Ürün geri yüklemenizden ilk sprint'inizi oluşturun

Bu alıştırmada, Ürün Geri Yüklemenizden ilk Sprint Geri Yüklemenizi dolduracaksınız. Genellikle, sprint planını tüm ekibinizle sprint planlama toplantısı sırasında oluşturursunuz. Ancak bu projeyi tek başımıza tamamladığınız için o toplantıyı simülle etmeniz gerekecek.

Yapılması Gereken Adımlar

1. GitHub'da üç sprint (**Sprint 1**, **Sprint 2** ve **Sprint 3**) oluşturun ve sprintlerin süresini bir hafta olarak ayarlayın.
2. Product Backlog listesinin en üstündeki ilk hikayeyi açın ve ona tahmini hikaye puanları atayın. Simdilik, **3, 5, 8, 13 = S, M, L, XL** ölçüğünü kullanın.
3. Mevcut hikayeyi **Sprint 1**'e atayın.
4. Mevcut hikayeyi kapatın ve Product Backlog'dan Sprint Backlog'a taşıyın, sıralı düzenini korumaya dikkat edin. Örneğin, Product Backlog'un en üstündeki hikaye, Sprint Backlog'ta da en üstte kalmalıdır.
5. Product Backlog'daki kalan dört hikaye için adım 3-5'i tekrarlayın, sıralı düzenlerini koruyarak.



Kontrol listesi için buraya tıklayın.

Kanıt

Hem Seçenek 1: AI-Değerlendirmeli Gönderim ve Değerlendirme hem de Seçenek 2: Akran-Değerlendirmeli Gönderim ve Değerlendirme için Kanban panonuzun bir screenshot'unu alın ve planning-kanban-done.jpeg veya planning-kanban-done.png olarak kaydedin.

Özet

GitHub Kullanarak Çevik Planlama Laboratuvarı’ni tamamladığınız için tebrikler! Bu laboratuvar kapsamında bu capstone projesi için ilk sprint planınızı oluşturduınız. Artık hesap mikroservisinizi uygulamaya başlamaya hazırlısanız.

Yazar(lar)

Tapas Mandal
[John J. Rofrano](#)



Skills Network