Developing with Git and GitHub Eğitimi

Süre: **3 Gün**

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com

Developing with Git and GitHub, modern yazılım geliştirme süreçlerinde kullanılan en popüler **Versiyon Kontrol Sistemi (Version Control System)** olan **Git** ve dünyanın en yaygın kod iş birliği platformlarından biri olan **GitHub** konusunda uzmanlaşmak isteyen profesyoneller için hazırlanmış kapsamlı bir eğitim programıdır. Eğitim, teorik bilgi ile uygulamalı laboratuvar çalışmalarını bir araya getirerek katılımcıların gerçek projelerde kullanabilecekleri beceriler geliştirmesini sağlar.

Program boyunca katılımcılar, Git mimarisini anlamaktan başlayarak kod versiyonlama, dallanma (**Branching**), birleştirme (**Merging**), iş birliği süreçleri, güvenlik uygulamaları ve**CI/CD Pipeline** tasarımı gibi kritik konuları öğrenirler. Ayrıca **GitHub Actions**, **GitHub Codespaces**, **GitHub Copilot** ve proje yönetim araçları gibi gelişmiş GitHub özelliklerini kullanarak ekip verimliliğini artırma yöntemlerini keşfederler.

Bu eğitim, yazılım geliştirme ekiplerinde daha etkili iş birliği kurmak, kod kalitesini artırmak ve modern DevOps süreçlerine uyum sağlamak isteyen profesyoneller için ideal bir çözümdür.

Önkoşullar

Bu eğitim orta seviyede (**Intermediate Level**) tasarlanmıştır.

Katılımcıların aşağıdaki bilgi ve deneyimlere sahip olması tavsiye edilir:

- 6 ila 12 ay yazılım geliştirme deneyimi
- Temel komut satırı (**Command Line**) kullanımı
- Dosya ve klasör yönetimi bilgisi
- Temel programlama bilgisi
- Kod versiyonlama kavramlarına aşinalık
- Takım tabanlı geliştirme süreçleri hakkında temel bilgi

Kimler Katılmalı

Bu eğitim aşağıdaki profesyoneller için uygundur:

- **Software Developers**
- **DevOps Engineers**
- **Cyber Security Specialists**
- **Technical Project Managers**
- **Data Scientists**
- Yazılım mimarları
- Uygulama geliştiricileri
- Yazılım ekip liderleri
- Modern geliştirme süreçlerini öğrenmek isteyen teknik profesyoneller

Developing with Git and GitHub Eğitimi

Süre: **3 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Neler Öğreneceksiniz

Eğitim sonunda katılımcılar:

- **Git Architecture** bileşenlerini açıklayabilir.
- Git ile versiyon kontrol süreçlerini yönetebilir.
- Yerel ve uzak (**Remote**) repository işlemlerini gerçekleştirebilir.
- Branch oluşturabilir ve merge süreçlerini yönetebilir.
- Merge çatışmalarını çözebilir.
- GitHub üzerinde ekip iş birliği süreçleri oluşturabilir.
- **Pull Request** ve **Code Review** süreçlerini uygulayabilir.
- GitHub güvenlik mekanizmalarını kullanabilir.
- **GitHub Actions** ile CI/CD süreçleri geliştirebilir.
- **GitHub Codespaces** kullanarak geliştirme ortamları oluşturabilir.
- **GitHub Copilot** ile yapay zeka destekli geliştirme süreçlerinden faydalanabilir.

Eğitim İçeriği

Git Temelleri

Versiyon Kontrolüne Giriş

- **Version Control System (VCS)** kavramı
- Git nedir?
- Git'in diğer versiyon kontrol sistemlerinden farkları
- Git kullanım senaryoları

Git Mimarisi

- **Working Directory**
- **Staging Area**
- **Repository**
- Git iş akışının temel yapısı

Git Kurulumu ve Yapılandırması

- Git kurulumu
- **git config**
- Kullanıcı ve proje ayarları
- Ortam yapılandırmaları

Temel Git Operasyonları

Repository Yönetimi

- **git init**
- Repository oluşturma
- Proje başlatma süreçleri

Developing with Git and GitHub Eğitimi

Süre: **3 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Dosya ve Commit İşlemleri

- **git add**
- **git status**
- **git commit**
- Anlamlı commit oluşturma teknikleri
- **git log**
- Commit geçmişini inceleme

HEAD ve Versiyon Takibi

- HEAD kavramı
- Versiyon geçmişinin yönetimi
- En iyi commit uygulamaları
- **.gitignore** kullanımı

Geri Alma ve Kurtarma İşlemleri

- **git reset**
- **git checkout**
- Değişiklikleri geri alma
- **git diff**
- **git stash**
- Geçici çalışma alanları
- Kurtarma stratejileri
- **git reflog**

Branching ve Merging

Branch Yönetimi

- Branch kavramı
- **git branch**
- Branch oluşturma ve silme
- İsimlendirme standartları

Merge İşlemleri

- **git merge**
- Branch birleştirme stratejileri
- Merge conflict yönetimi
- Çatışma çözme teknikleri

GitHub Temelleri ve İş Birliği

GitHub'a Giriş

- GitHub nedir?
- Git ve GitHub farkları
- Hesap oluşturma ve yapılandırma
- Repository yönetimi

Uzak Repository Operasyonları

- **git remote**
- **git clone**
- **git fetch**
- **git pull**
- **git push**
- Tracking branch yapıları

Takım Çalışması ve İş Birliği Süreçleri

Pull Request ve Code Review

- **Pull Request (PR)** süreçleri
- Kod inceleme yöntemleri
- Branch koruma kuralları
- Merge stratejileri

Fork ve Clone Yaklaşımları

- Fork kullanımı
- Açık kaynak katkı süreçleri
- Takım geliştirme iş akışları

GitHub Proje Yönetimi

- **GitHub Issues**
- Milestone yönetimi
- **Project Boards**
- Markdown dokümantasyonu
- Wiki kullanımı
- README en iyi uygulamaları

GitHub Güvenliği

- **Personal Access Tokens**
- SSH anahtarları
- **Two-Factor Authentication (2FA)**
- Repository güvenlik ayarları
- Erişim yönetimi
- Güvenlik en iyi uygulamaları

GitHub Actions ve CI/CD

CI/CD Temelleri

- **Continuous Integration**
- **Continuous Delivery**
- Workflow yapıları

GitHub Actions

Developing with Git and GitHub Eğitimi

Süre: **3 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Workflow oluşturma
- Trigger mekanizmaları
- Environment Variables
- Secrets yönetimi
- Test ve deployment süreçleri

GitHub Codespaces

- Bulut tabanlı geliştirme ortamları
- Codespaces yapılandırması
- Özelleştirme seçenekleri
- Performans ve maliyet yönetimi

GitHub Copilot

Yapay Zeka Destekli Yazılım Geliştirme

- **GitHub Copilot** kurulumu
- Prompt oluşturma teknikleri
- Kod önerileri ve tamamlama
- Verimlilik artırma yöntemleri
- Güvenlik ve kullanım sınırları

Gelişmiş Git Komutları

- **git rebase**
- **git show**
- Geçmiş yönetimi
- Repository optimizasyonu
- İleri seviye Git teknikleri