Çizge Renklendirme ile Ders Programı Hazırlama

Aydın Kasımoğlu   
*Kocaeli Üniversitesi Teknoloji Fak.*  
*Bilişim Sistemleri Mühendisliği*  
211307011

Kerem Işık  
*Kocaeli Üniversitesi Teknoloji Fak.*  
*Bilişim Sistemleri Mühendisliği*  
221307102

Muhammet Ali Kalyas   
*Kocaeli Üniversitesi Teknoloji Fak.*  
*Bilişim Sistemleri Mühendisliği*  
221307106

*Özet*—Bu belge, hazırlanan ders programı hazırlama uygulamasının iç yapısını sunmak için yazılmıştır. Ders planlamasının artan karmaşıklığı, süreci kolaylaştırmak ve kaynak kullanımını artırmak için yenilikçi çözümler gerektirmektedir. Bu rapor, Next.js, PostgreSQL ve Prisma'dan yararlanarak ders planlaması için geliştirilen bir web uygulamasının kapsamlı bir incelemesini sunmaktadır. Sistem, çakışmaları en aza indirmek ve verimli bir kurs programı sağlamak için bir grafik renklendirme algoritması içermektedir.

Anahtar Kelimeler—Çizge renklendirme, ders programı, tablo, çakışma

# Giriş

Etkili ders planlaması, hem öğrenciler hem de eğitmenler için sorunsuz bir akademik deneyim sağlanmasında çok önemli bir rol oynar. Geleneksel olarak, üniversiteler tarafından manuel planlama süreçleri kullanılmakta ve bu da genellikle verimsizliklere ve çatışmalara yol açmaktadır.

Projemizin temel amacı, otomatik bir çözüm sunarak geleneksel ders planlama yöntemlerinin eksikliklerini gidermektir. Modern web teknolojilerini ve grafik renklendirme algoritmalarını kullanarak, web uygulamamız üniversite ortamlarında ders programlarını oluşturmak, yönetmek ve optimize etmek için kullanıcı dostu bir platform sağlamayı amaçlamaktadır.

Bu proje, verimli veri yönetimi için Prisma kullanan bir PostgreSQL veri tabanı ile entegre olan Next.js tabanlı bir web uygulamasının geliştirilmesine odaklanmaktadır. Kapsam, kursların oluşturulmasını, eğitmenlerin eklenmesini ve silinmesini ve bir grafik renklendirme algoritması aracılığıyla programın renklendirilmesini içerir. Çözümümüz tüm olası senaryoları kapsamasa da, kurs planlamasının temel yönlerini otomatikleştirmede önemli bir ilerleme sunuyor.

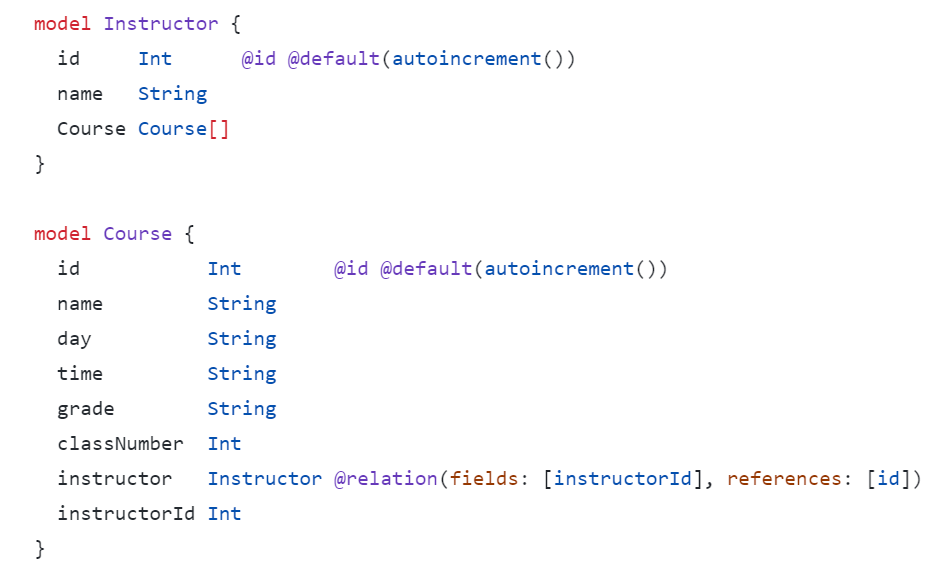
# Yöntem

Ders programı hazırlama uygulamamızın geliştirilmesi, ön uç ve arka uç bileşenlerini sorunsuz bir şekilde entegre eden modüler ve ölçeklenebilir bir mimari içeriyordu. Ön uç, verimli istemci tarafı işleme ve kolay gezinme için React tabanlı bir çerçeve sağlayan Next.js kullanılarak oluşturuldu. Arka uçta, verileri depolamak ve yönetmek için bir PostgreSQL veri tabanı kullandık ve Prisma veri erişim katmanı olarak hizmet vererek uygulama ile veri tabanı arasında sorunsuz iletişimi kolaylaştırdı.

Next.js ön ucu, bileşen tabanlı bir mimariyi teşvik ederek yeniden kullanılabilir bileşenler halinde düzenlenmiştir. React'in Context API'si ve Hook'ları aracılığıyla durum yönetimini kullanarak uyumlu ve duyarlı bir kullanıcı arayüzü sağladık. Next.js kullanımı, sunucu tarafı oluşturmaya olanak tanıyarak uygulamanın performansını ve kullanıcı deneyimini artırdı.

Arka uç mimarisi, Prisma'nın ORM'si (Object-Relational Mapping) aracılığıyla PostgreSQL veri tabanı ile etkileşime giren Server Action’lar olarak tasarlanmıştır. Server Action'lar sunucuda yürütülen asenkron fonksiyonlardır. Next.js uygulamalarında form gönderimleri ve veri mutasyonlarını işlemek için Sunucu ve İstemci Bileşenlerinde kullanılabilirler [1]. Prisma, tip güvenli veri tabanı sorgularına ve geçişlerine olanak tanıyarak sistemin genel güvenilirliğine ve sürdürülebilirliğine katkıda bulunur.

Veri modelimiz ders programı hazırlanması için gerekli olan varlıkları doğru bir şekilde temsil edecek şekilde tasarlanmıştır: dersler ve eğitmenler. PostgreSQL ilişkisel veri tabanı olarak kullanılmakta ve Prisma verimli veri tabanı iletişimi için veri erişim katmanı olarak görev yapmaktadır.

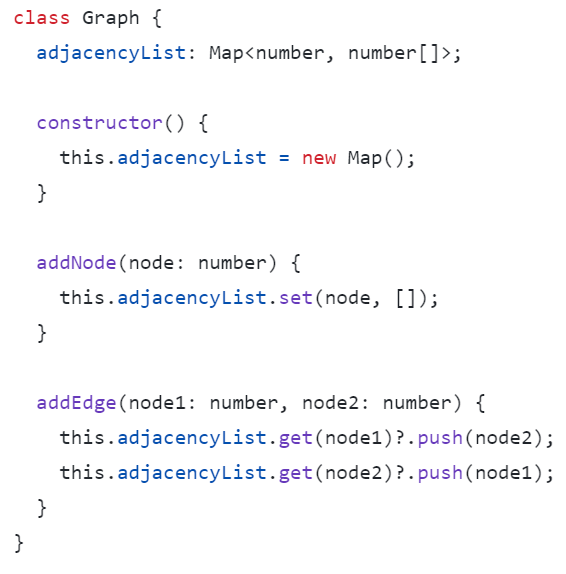


Şekil 1: Uygulamamızda yer alan temel varlıkları temsil eden Prisma veri modeli.

Her ders bir eğitmenle ilişkilendirilir ve birden fazla ders aynı eğitmeni paylaşabilir. Eğitmenler, öğrettikleri kurslarla bağlantılıdır ve bire çok ilişki oluşturur.

Programı renklendirmek için uygulamamız bir grafik renklendirme algoritması içermektedir. Programlama grafiği ders bağımlılıkları, eğitmen uygunluğu ve oda kısıtlamalarına göre oluşturulur. Algoritma, çakışan hiçbir dersin aynı rengi almamasını sağlayarak derslere, zaman dilimlerini temsil eden renkler atar.

İlk adım, Graph sınıfı kullanılarak programlama grafiğinin oluşturulmasını içerir. Her ders için düğümler eklenir ve aynı anda programlanamayan dersler gibi bağımlılıklara dayalı olarak kenarlar oluşturulur.



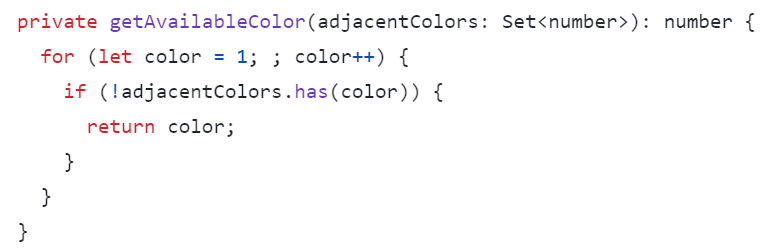
Şekil 2: Graph sınıfı.

Algoritmanın çekirdeği CourseScheduler sınıfında, özellikle de scheduleCourses yönteminde bulunur. Bu yöntem, grafikteki her ders düğümünü yineleyerek bitişik derslerin (kenarlarla bağlı) farklı renkler almasını sağlamak için renkler atar.



Şekil 3: scheduleCourses metodu.

scheduleCourses yöntemi her dersi yineleyerek komşu derslere atanan renkleri belirler. Daha sonra ders için uygun bir renk belirlemek üzere getAvailableColor adlı bir yardımcı yöntem kullanır ve bitişik iki dersin aynı rengi paylaşmamasını sağlar.



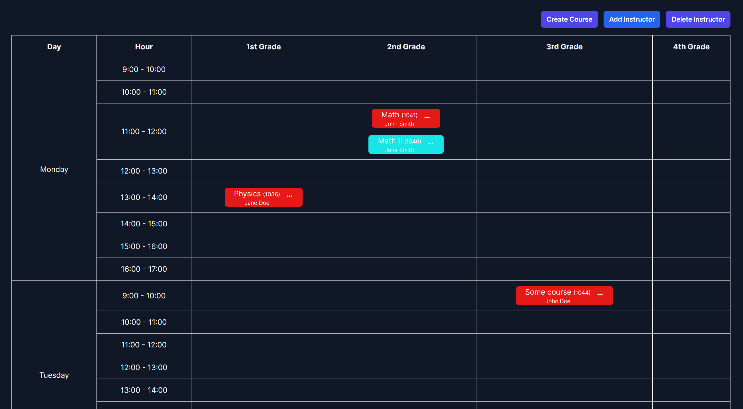
Şekil 4: getAvailableColor metodu.

Bu algoritma, farklı zaman dilimlerini veya kaynakları temsil eden renkleri derslere en iyi şekilde atar, böylece verimli ve çakışmasız bir ders programı oluşturur.

# Örnek Sonuçlar

Bu bölümde, ders oluşturma, eğitmen yönetimi ve çakışmasız çizelgeleme için çizge renklendirme algoritmasının uygulanması dahil olmak üzere uygulanan özelliklerin etkinliğini vurgulayarak, ders planlama web uygulamamız tarafından üretilen örnek sonuçları sergiliyoruz.

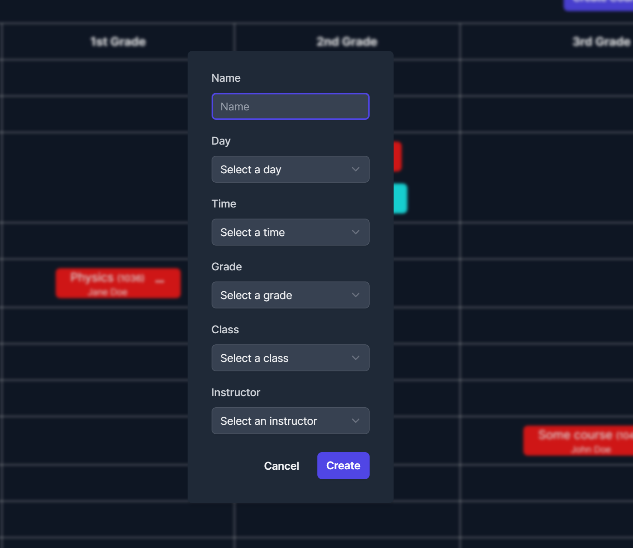
Kullanıcı arayüzü, ders planlama uygulaması ile etkileşim için sezgisel bir platform sağlar. Kullanıcılar kurs oluşturma, eğitmen ekleme ve program görselleştirme gibi işlevler arasında sorunsuz bir şekilde gezinebilir. Şekil 5, uygulamanın ana arayüzünün bir anlık görüntüsünü göstermektedir:



Şekil 5: Ders Planlama Web Uygulaması Arayüzü

Ekran görüntüsü, ders yönetimi için organize bir düzen ve mevcut programın görsel bir temsilini içeren uygulamanın ana kontrol panelini göstermektedir.

Uygulamanın dinamik olarak yeni dersler oluşturma becerisini değerlendirmek için testler gerçekleştirdik. Kullanıcılar kurs adı, günü, saati, sınıf seviyesi ve sınıf numarası dahil olmak üzere kurs ayrıntılarını girebilmektedir. Uygulama daha sonra bağımlılıkları ve kısıtlamaları göz önünde bulundurarak bu dersleri ders programına dâhil ediyor.



Şekil 6: Ders ekleme formu.

##### Kaynakça

1. “Data Fetching: Server Actions and Mutations | Next.js,” nextjs.org. https://nextjs.org/docs/app/building-your-application/data-fetching/server-actions-and-mutations (accessed Jan. 07, 2024).