**T.C.**

**TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ**

**ÇORLU MÜHENDİSLİK FAKÜLTESİ**

**STAJ ÖDEVİ**

**Asistan Görev Dağıtıcı CLI Uygulaması**

**Derya Nurhak**

**2180656009**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN: Dr. Öğr. Üyesi AHMET SAYGILI**

**TEKİRDAĞ-2021**

Bu stajda görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, staj içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde eksiksiz biçimde kaynak göstererek belirttiğimi beyan ederim.

STAJ KURULU ÜYELERİ danışmanlığında, Derya Nurhak tarafından hazırlanan “Asistan Sınav Gorevi Dağıtıcı” başlıklı bu çalışma aşağıdaki jüri tarafından gg.aa.2021 tarihinde Bilgisayar Mühendisliği Anabilim Dalı’nda bitirme stajı olarak oy birliği/oy çokluğu ile kabul/red edilmiştir.

Jüri Başkanı : Ünvan Ad SOYAD *İmza:*

Üye : Ünvan Ad SOYAD *İmza:*

Üye : Ünvan Ad SOYAD *İmza:*

Üye : Ünvan Ad SOYAD *İmza:*

Üye : Ünvan Ad SOYAD *İmza:*

# 1. İçindekiler

[1. İçindekiler 1](#_Toc82900436)

[2. PROJE KONUSU 2](#_Toc82900437)

[3. PROJENİN ÇÖZDÜĞÜ SORUNLAR 3](#_Toc82900438)

[4. YARARLANILAN PROGRAMLAR 4](#_Toc82900439)

[5. PROJE TASARIM 5](#_Toc82900440)

# 2. PROJE KONUSU

Üniversitemizde görev yapan akademik personelin görev dağılımını eşit bir şekilde yapacak , kargaşayı önleyip araştırma görevlilerine yardımcı olacak CLI uygulaması.

# 3. PROJENİN ÇÖZDÜĞÜ SORUNLAR

Proje konusu olan terminal uygulaması üniversitelerin sınav dönemlerinde akademik personele görev dağılımını eşit bir şekilde paylaştırmayı ve personel yükünü hafifletmeyi amaçlamaktadır. Sistemin çalışma mantığı çerçevesinde ister tüm görevleri görüntülenmekte istenirse de her personel kendi görevlerini listeleyip görüntüleyebilmektedir. Bu bağlamda görev dağılımı ve bu dağılımın personele bildirimi konusunda karmaşa engellenecek , zaman ve enerji tasarrufu sağlanacaktır.

# 4. YARARLANILAN PROGRAMLAR

Program JavaScript dili ile yazılmıştır.

Programlar: NodeJs , Microsoft,Linux,Mac Terminal , VS Code

# 5. PROJE TASARIM

Görevliler.js dosyasında “görev alacak araştırma görevlileri” array olarak export edilmiştir.

Sinav.json dosyasında “görevler” Json formatında saklanmıştır.

Hazır olarak commander adında NPM modülü kullanılmıştır. Path ve Filesystem(fs), Nodejs ile yerleşik olarak gelen modüllerden yardım alınmıştır.

Commander paketi içindeki command class’ından program objesini türetiyoruz.

const program = new Command();

Ekrana yazdırma komutunu kolay olması amacıyla “log” kelimesine atıyoruz.

const log = console.log

Bu class’ın içindeki option fonksiyonu ile CLI parametrelerini belirliyoruz

program.option('-t, --tumu\_dagit', 'Göreveleri round-robin algoritmasıyla dağıtır')

program.option('-a, --ada\_gore\_dagit', 'adınıza göre düşen görevleri gösterir')

program.option('-i, --indexe\_gore\_dagit', "index'inize düşen görevleri gösterir")

Konsola giren değerleri parse edip program içinde kullanmamıza yarayan fonksiyon

.parse(process.argv)

GorevDagit fonksiyonu görevleri , araştırma görevlilerine “Round Robin” algoritma mantığı ile dağıtıyor.

const gorevDagit = () => {

    let i = 0;

    const gorevliSayisi = gorevliler.length

    const sinavlar = JSON.parse(readFileSync( resolve(process.cwd(), 'bin', 'sinav.json'), 'utf8'))

    return sinavlar.map( sinav => {

        if( i == gorevliSayisi ) i = 0;

        const data =  {

            gorevli : gorevliler[i],

            sinav

        }

        i++

        return data

    })

}

Gelen parametreyi belirlemek için Switch-Case mekanizması kullanılmıştır.

Options {komut\_adi: true } şeklinde bir obje dönmektedir. Objenin key’ini parse edip switch’e parametre olarak verilmektedir.

switch ( Object.keys(options)[0]) {

Eğer “option” olarak “-t” gelir ise tüm görevleri listelemektedir. gorevDagit fonksiyonunu döndüğü değeri doğrudan konsola yazdırmaktadır.

 case "tumu\_dagit":

        log( gorevDagit())

        break;

Eğer “option” olarak “-a” ve “akademisyen\_ismi” olarak gelir ise belirlenen akademisyenin görevlerini isme göre filtrelemektedir. Filtrelenen değeri “isime\_cikti” değişkenine atayıp görüntülemektedir.

  case "ada\_gore\_dagit":

            const isime\_cikti = Array.from(gorevDagit()).filter(gorev => {

               return gorev.gorevli == program.args[0]

               //return gorev.gorevli == gorevliler[Number(program.args)]

           })

            log(isime\_cikti)

        break;

Eğer “option” olarak “-i” ve “index\_numarasi” olarak gelir ise belirlenen akademisyenin görevlerini index numarasına göre filtrelemektedir. Filtrelenen değeri “indexe\_cikti” değişkenine atayıp görüntülemektedir.

 case "indexe\_gore\_dagit":

            const indexe\_cikti = Array.from(gorevDagit()).filter(gorev => {

                return gorev.gorevli == gorevliler[Number(program.args)]

            })

            log(indexe\_cikti)

        break;

Program parametre gönderilmeden çalıştığında basit bir yardım ekranı görüntülenmektedir.

 default:

        log()

        log()

        log("Herhangi bir komut girilmedi, komut listesi için -h")

        log('Görevler round robin algoritması ile dağıtılmıştır')

        log()

        log('Tüm görev listesini görmek için -t yazınız')

        log()

        log( `görevli listesi`, gorevliler.map( (g, i) => { return { isim: g, index: i}}))

        log('Sadece kendi görevinizi görmek için -a "adınız" veya -i index numaranız şeklinde yazınız')

        break;