



GAZİ ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

BMT-308

WEB PROGRAMLAMA

PROJE

İŞ SORUN FORMU SİTESİ

Ders Sorumlusu: Doç. Dr. Necaattin BARIŞÇI

Lab Sorumlusu: Arş. Gör. H. Talha OKUR / Arş. Gör. Esra SÖĞÜT

Hazırlayan: Demet MUTLU

171816040 / 2.Şube / Grup 29

ANKARA

2019-2020

İçindekiler

1. GİRİŞ	4
2. PROJE İÇERİĞİ	4
2.1. PROJE TANITIMI.....	4
2.2. PROJE TASARIM VE GELİŞTİRME AŞAMALARI	5
2.3. KULLANILAN TEKNOLOJİLER	10
3. VERİTABANI VE ER DİYAGRAMI	16
4. SONUÇ.....	21
5. EK.....	21
6. KAYNAKÇA	21

Resim 1: Giriş Ekranı	5
Resim 2:Kayıt Ekranı	6
Resim 3:Uygulama Dashboard Ekranı	6
Resim 4:Menü Görünümünü Küçültme	7
Resim 5: Kullanıcı Menüsü.....	7
Resim 6:Proje Listesi	8
Resim 7: Yeni Proje Kaydı.....	8
Resim 8: Sorun Listesi	9
Resim 9:Yeni Sorun Kaydı	9
Resim 10: Sorun Detayları	10
Resim 11:Intelij IDEA	12
Resim 12:Spring Boot	12
Resim 13:Spring Initializr	13
Resim 14:Angular	13
Resim 15: PostgreSql	14
Resim 16:Postman.....	15
Resim 17: Postman Kod Örneği (PUT).....	15
Resim 18:Veritabanı Bağlantısı	16
Resim 19:Veritabanı Tablolar PostgreSql.....	16
Resim 20:Veritabanı Tablolar Intellij	16
Resim 21:ER Diyagramı	20
Resim 22:Kullanıcı Menü İçeriği	21

1. GİRİŞ

İnsanoğlu dünyaya ayak bastığından bu yana çeşitli sorunlar ile karşılaşmış ve sorunları çözmek üzerine aklını kullanarak günümüz teknolojisine ulaşmıştır.

Karşılan sorunların sayısı arttıkça çözmek karmaşıklaşmış ve çözüm süreci de uzamıştır. Buradan doğan ihtiyaçla tüm sorunların bir yerde toplanıp detayları ve durumları ile birlikte saklanması planlanmıştır.

Yapılmış olan ‘İş Sorun Formu Sitesi’ projesinde belli bir iş üzerinde çalışan insanların sorunlarını tek bir yerde toplayarak çözüm sürecinin hızlanması planlanmıştır. Bu proje ile birlikte;

- Yapılan işteki sorunların takibi kolaylaşmıştır.
- Yapılan işin süresinin sorun karmaşası nedeniyle uzaması engellenmiştir.

Proje fikrinin ortaya çıkış hikayesi, öğrencilerin okul tarafından verilmiş olan birçok projenin takibinde zorlanmasıdır. Yapılmakta olan projelerde karşılaşılan sorunlar site üzerinde toplanılarak işleyiş düzene girmektedir.

‘İş Sorun Proje Formu Sitesi’ hedef kitlesi herhangi iş üzerinde çalışan ve sorunlarını düzenli bir şekilde kayıt altına almak isteyen bireylerdir.

2. PROJE İÇERİĞİ

2.1.PROJE TANITIMI

Çözülmesi Planlanan Problem:

Yaşanılan sorunlardaki karmaşıklık, düzensizlik ve bunlardan doğan zaman kaybı.

Proje kapsamında kullanılması planlanan senaryo ise bir şirketin üzerinde çalıştığı projelerdir. Şirketin yaptığı projelerdeki işleyiş süreci, karşılaşılan sorunlar, çözüm süreci gibi faktörler büyük önem arz etmektedir. Bunlarda karışıklıktan doğabilecek her sorun gecikmeye sebep olabilmektedir. Her türlü aksaklığı önlemek için bu web sitesi kullanıma uygundur. Şirketin bu konuda görevli olan çalışanı ilgili isterleri girerek programı kullanacaktır.

Proje ile Probleme Getirilen Çözüm Önerisi:

Her çalışan, kayıt olduktan sonra uygulamaya giriş yapabilmektedir. Şirketin gerçekleştirdiği projeler kayıt edilerek çalışanlar yaşadıkları sorunları tarih, detay, kimin bildirdiği, durum gibi isterler ile sisteme işlerler. Böylece sorunların takibi kolaylaşmış olur.

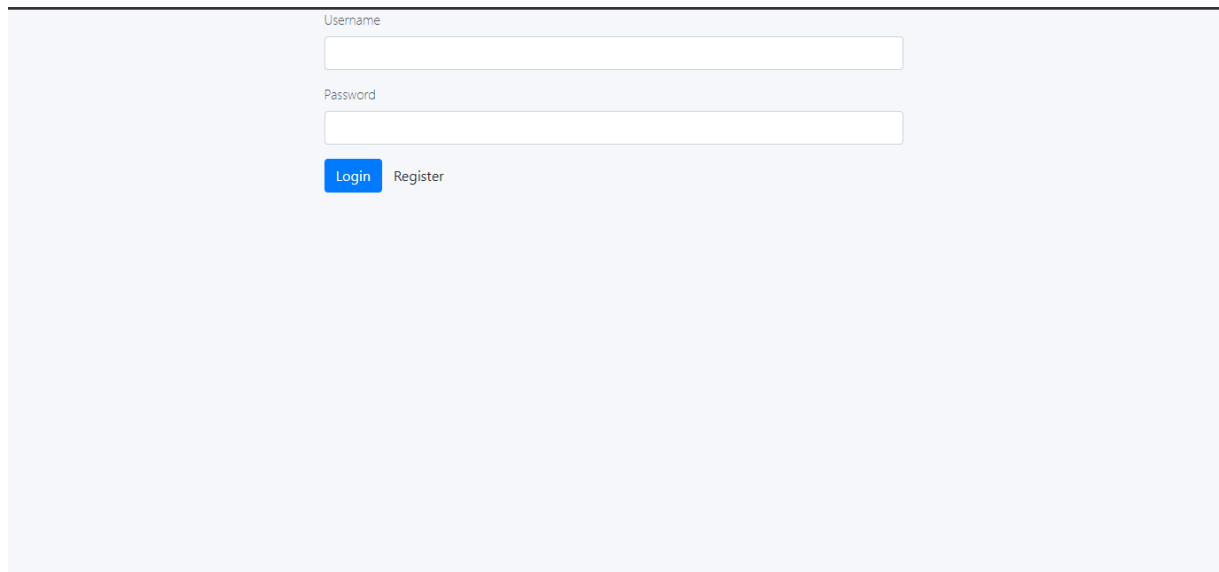
Uygulamanın Yetenekleri:

Kullanıcı, giriş sayfasında ilk olarak kayıt olmalıdır. Kullanıcı adı, parola, isim, soy isim ve e-mail bloklarını doldurarak sisteme kayıt olunmaktadır. Kayıt olunan bilgiler ile giriş yapılır.

Kullanıcıyı Dashboard ekranı karşılar. İlk olarak üzerinde çalışılacak olan proje eklenmelidir. Menüden proje sekmesine tıklayarak proje ekleme sayfasına geçilir. Yeni proje için proje kodu, proje adı ve proje yöneticisi eklenmelidir. Daha sonra menüden sorun sayfasına geçilerek yeni sorun eklenmelidir. Eklenilen bir proje, kendisine bağlı olan sorun olmadığı takdirde silinebilir. Sorun için gerekli açıklama ve ilgili proje seçilir. Bu işlem sorunlar listesine açıklama, tarih, durum, düzenleyen, proje adı, proje yöneticisi, ve detay sayfası olarak eklenir. Eklenmiş olan sorunları düzenlemek için ilgili sorunun detaylarına girilir. Proje durumu ve detay ile birlikte diğer bilgiler de güncellenebilir. Her proje ve sorun kendi içinde listelenebilir, kullanıcıların isteğine göre sıralanarak raporlanabilir.

2.2.PROJE TASARIM VE GELİŞTİRME AŞAMALARI**Proje Arayüz Tasarımı:**

Resim 1: Giriş Ekranı



The image shows a login interface with a light blue background. At the top, there is a header bar with the text 'Giriş Ekranı' in a dark blue font. Below the header, there are two input fields: 'Username' and 'Password'. The 'Username' field is a white rectangle with a light blue border. The 'Password' field is a white rectangle with a light blue border and a small eye icon on the right side. Below the password field, there is a blue button with the text 'Login' in white. To the right of the 'Login' button, there is a link that says 'Register' in a dark blue font. At the bottom of the page, there is a dark blue footer bar with the text 'Giriş Ekranı' in a light blue font.

Resim 2:Kayıt Ekranı

Username

Password

Name Surname

Email

Register

Login

Resim 3:Uygulama Dashboard Ekranı

IssueManagement

Dashboard

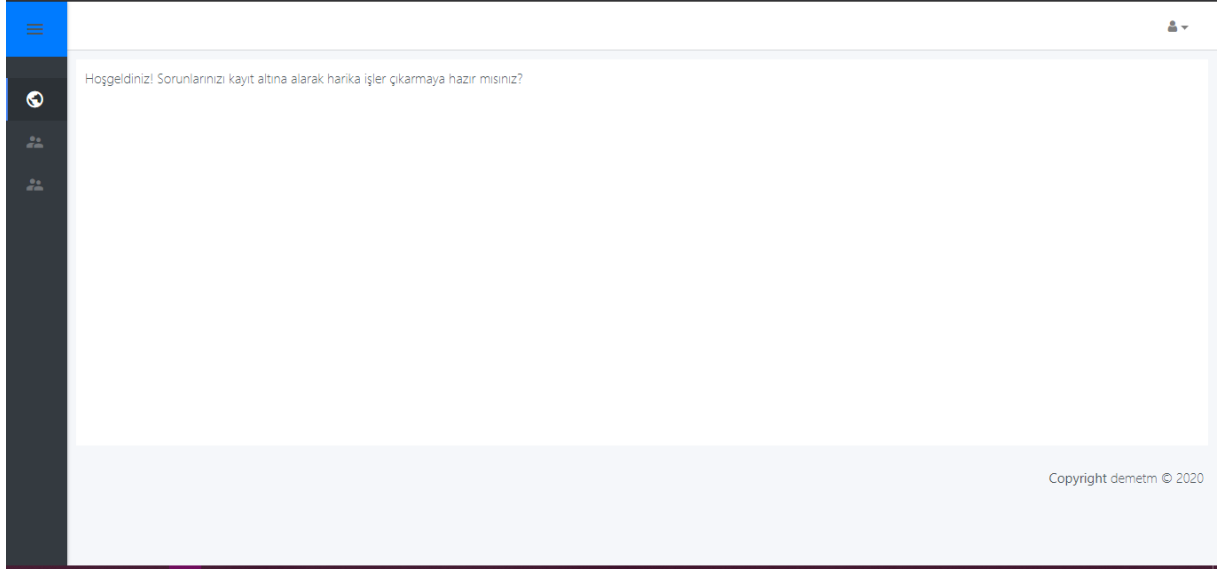
Project

Issue

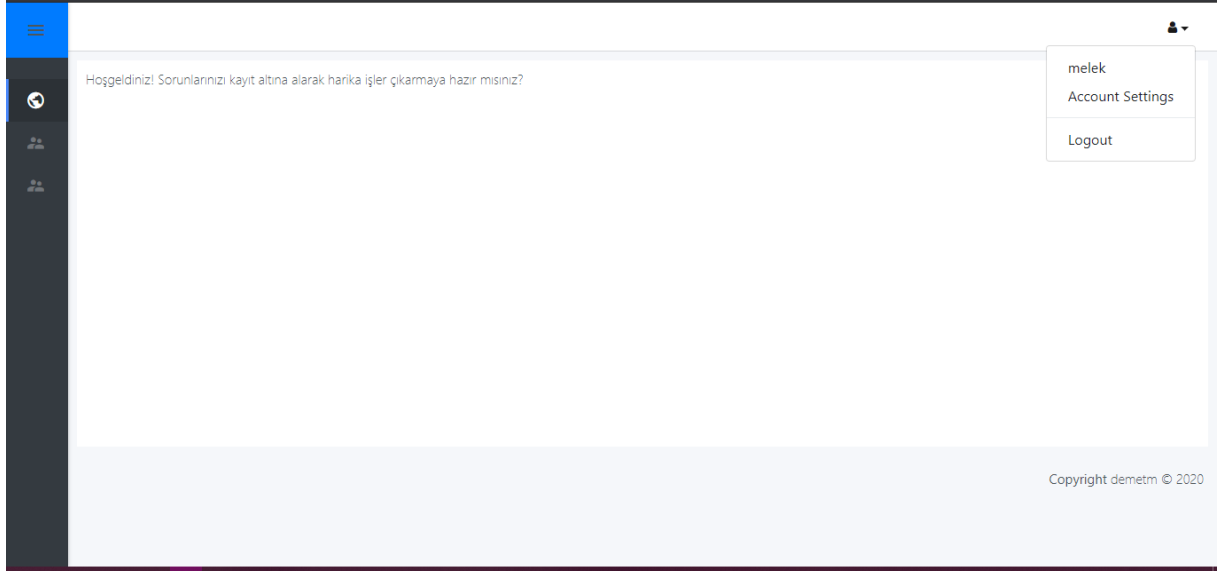
Hosgeldiniz! Sorunlarınızı kayıt altına alarak harika işler çıkarmaya hazır mısınız?

Copyright demetm © 2020

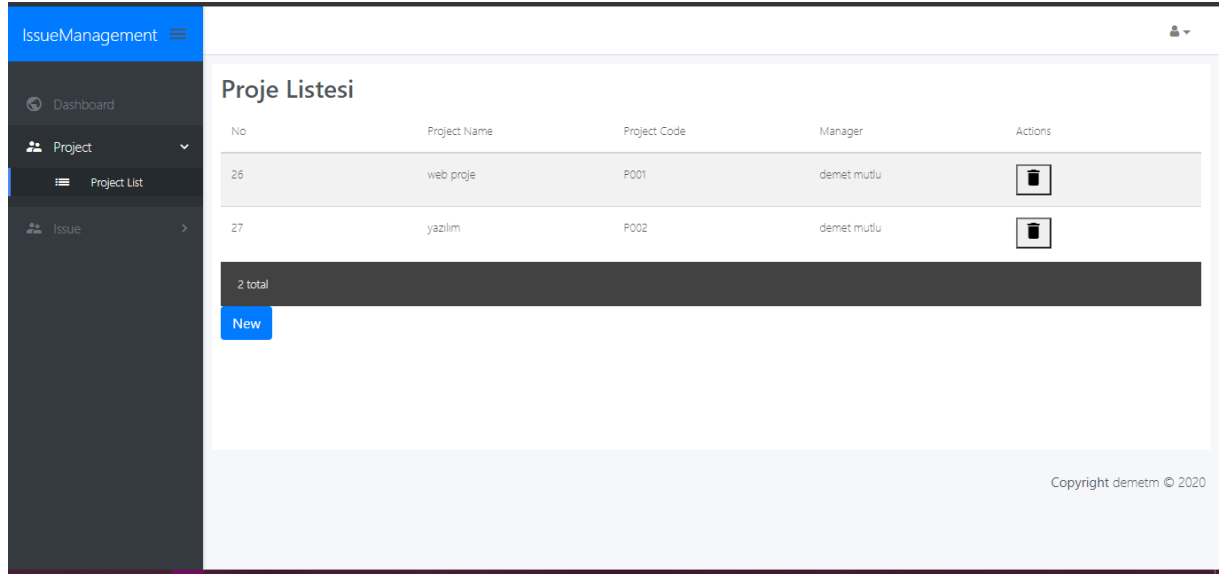
Resim 4:Menü Görünümünü Küçültme



Resim 5: Kullanıcı Menüsü



Resim 6:Proje Listesi



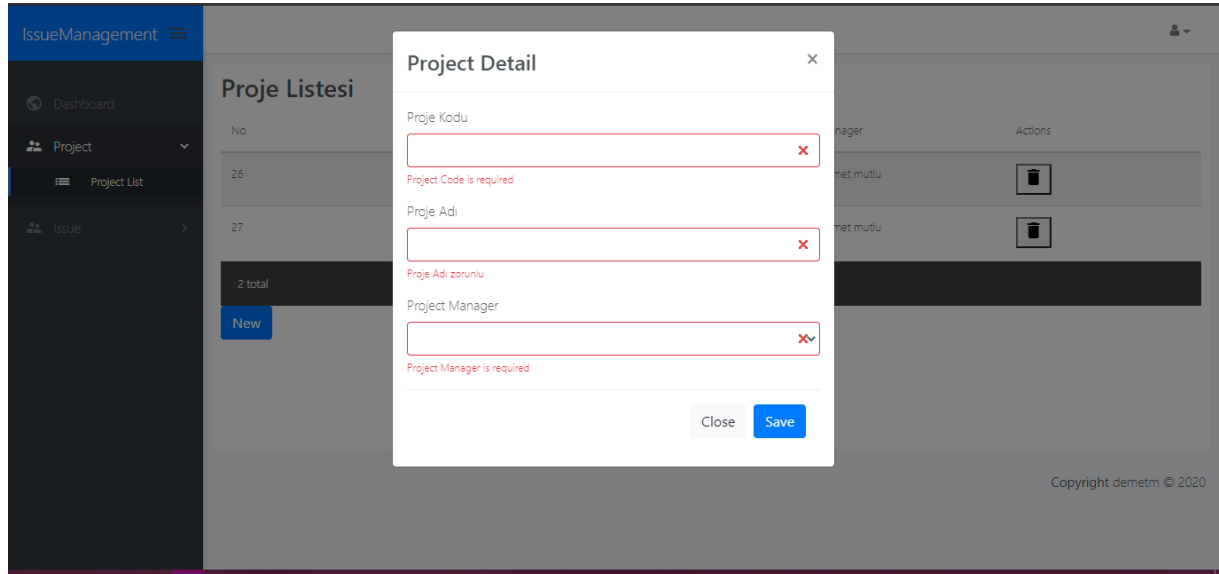
No	Project Name	Project Code	Manager	Actions
26	web proje	P001	demet mutlu	
27	yazılım	P002	demet mutlu	

2 total

New

Copyright demetm © 2020

Resim 7: Yeni Proje Kaydı



Project Detail

Proje Kodu

Project Code is required

Proje Adı

Proje Adı zorunlu

Project Manager

Project Manager is required

Close Save

Copyright demetm © 2020

Resim 8: Sorun Listesi

IssueManagement

Dashboard

Project

Issue

Issue List

Issue List

No	Description	Issue Date	Status	Assignee	Project Name	Project Manager	Actions
28	kodlarda eksiklik	2020-06-15T05:44:30.958+0000	OPEN		yazılım	demet mutlu	Detail
30	donanımsal eksiklik	2020-06-11T00:00:00.000+0000	IN_PROGRESS	melek mutlu	yazılım	demet mutlu	Detail

2 total

New

Copyright demetm © 2020

Resim 9: Yeni Sorun Kaydı

IssueManagement

Dashboard

Project

Issue

Issue List

Issue List

No	Description	Issue Date	Status	Assignee	Project Name	Project Manager	Actions
28	kodlarda				yazılım	demet mutlu	Detail
30	donanımsal				yazılım	demet mutlu	Detail

2 total

New

New Issue

Issue Description

Description is required

Project

Project is required

Close

Save

Copyright demetm © 2020

Resim 10: Sorun Detayları

The screenshot shows the 'Issue Management' application interface. On the left is a sidebar with a menu containing 'Dashboard', 'Project', 'Issue', and 'Issue List'. The main area displays the 'Issue Details' form for an issue with ID 31. The form fields are as follows:

- Description:** donanımsal eksiklik
- Issue Date:** 11.06.2020
- Assignee:** melek mutlu
- Issue Status:** IN_PROGRESS
- Project:** yazılım
- Project Manager:** demet mutlu
- Details:** donanım kontrolü yapıldı

A blue 'Save' button is located at the bottom right of the form. Below the form is a table with the following data:

No	Description	Issue Date	Issue Status	Assignee	Project Name
31	donanımsal eksiklik	15/06/2020	OPEN		yazılım

Proje her haliyle kullanıcıların isterleri düşünülerek tasarlanmış ve geliştirilmiştir. Menü kısmında projenin asıl amacı olan projeler ve sorunlar listesi bulunmaktadır. Bu iki sayfa içeriği de proje ve sorunların gerekli olan özelliklerini içermektedir.

Projeler veritabanında id ile tutularak veritabanı ilişkisi güçlendirilmiştir.

Yapılan her kayıt sonrası sayfa altında girişler otomatik listelenmektedir. Bu işlem ile listeler, kullanıcının istediği şekilde sıralanabilmektedir.

Menü sekmesinin tam olarak görünür olması site ekranını küçültmektedir. Bunu önlemek adına menü ekranı sadece sembollerin görülebileceği şekilde küçültülmüştür.

2.3.KULLANILAN TEKNOLOJİLER

Kullandığınız teknolojileri (yazılımlar, yazılım dilleri, yardımcı kaynaklar vb.) açıklayınız. Hangi teknolojilere neden ihtiyaç duydunuz? Size ne gibi kolaylıklar sağladı?

- Java Development Kit 8 (JDK8)
- IntelliJ
- IntelliJ Lombok Plugini
- PostgreSQL Server
- NodeJS
- Angular CLI

15.06.2020

- PLATFORM

JDK 8

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

- IDE

IntelliJ IDE

<https://www.jetbrains.com/idea/download/>

IntelliJ Lombok Plugin

<https://plugins.jetbrains.com/plugin/6317-lombok-plugin>

Eclipse

<https://www.eclipse.org/downloads/packages/>

Eclipse - Lombok ve Spring Tool Suite Plugin

<https://projectlombok.org/setup/eclipse>

POSTMAN

<https://www.getpostman.com/>

Notepad++

<https://notepad-plus-plus.org/download/v7.6.3.html>

- DATABASE

PostgreSQL

<https://www.postgresql.org/>

PostgreSQL Download >10

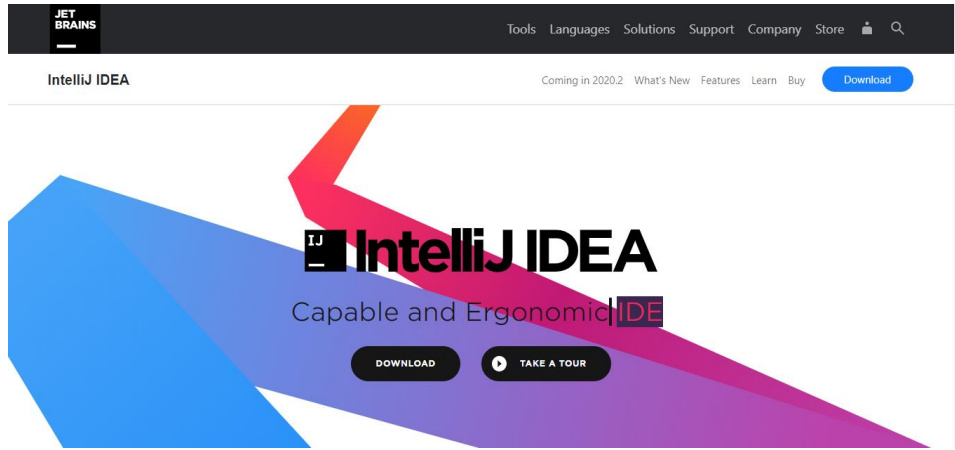
<https://www.postgresql.org/download/>

- NODEJS

NodeJS >10

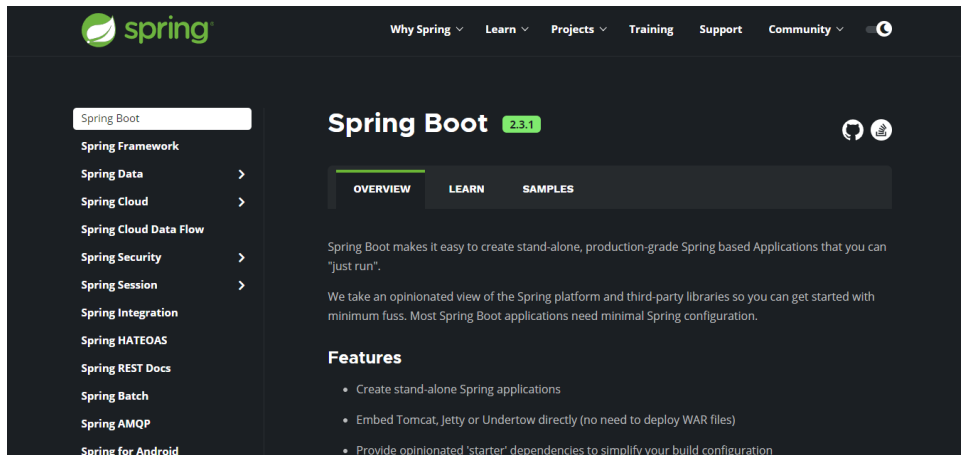
<https://nodejs.org/en/>

Resim 11: IntelliJ IDEA



IntelliJ Idea ile java yazılım dili kullanılarak projenin backend tarafı kodlanmıştır. Kapsamlı projeler için nesne yönelimli yazılım dili olan java en uygun olarak görülmüştür. Java için güncel, kullanımı kolay ve gelişmiş olan IntelliJ Idea kullanılmıştır.

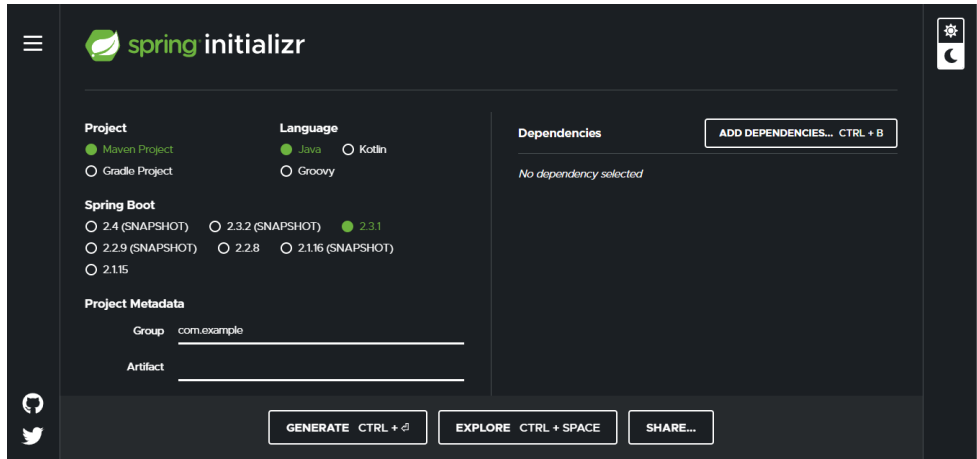
Resim 12: Spring Boot



Spring Boot, Spring tabanlı uygulama geliştirmenin en hızlı ve kolay yolu olması amacıyla geliştirilmiş bir framework'tür. Spring Boot web sunucusu olarak Tomcat ve diğer birçok ek özellikle beraber gelmektedir. Spring Boot'un sağladığı en büyük avantajlardan biri ise sizi herhangi bir XML konfigürasyonu ile uğraşmak zorunda bırakmamasıdır.

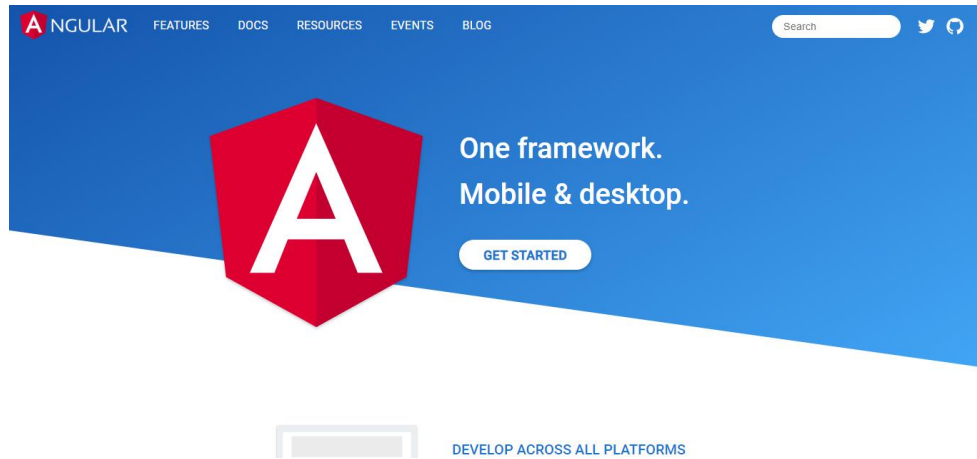
- Bağımsız (stand-alone) Spring uygulamaları oluşturulur.
- Gömülü bir web sunucusu (Tomcat, Jetty, Undertow) ile gelmektedir.
- Build konfigürasyonunu kolaylaştırmak için sağladığı starter'lar vardır.
- Otomatik konfigürasyon içerir.
- Kod üretimi ve XML konfigürasyona ihtiyaç duymaz.

Resim 13:Spring Initializr



Projenin ilk oluşturma aşaması Spring Initializr üzerinden yapılmıştır. Gerekli olan teknolojiler ‘ADD DEPENDENCES’ butonundan eklenmiştir.

Resim 14:Angular

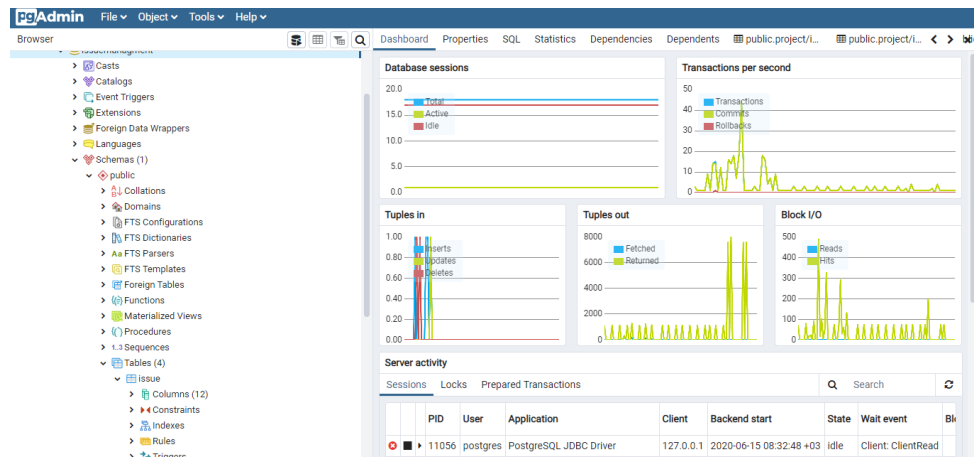


Angular, Google tarafından geliştirilen ve desteklenen SPA (Single Page Application) uygulama yapılmasına olanak sağlayan javascript kütüphanesidir.

- Hızlı bir biçimde SPA uygulama geliştirilebilir.
- Modüler bir yapıya sahiptir. Uygulama parçalara bölünebilmektedir. Tekrar kullanılabilirlik konusunda oldukça başarılıdır.
- İhtiyaç olan bütün paketler Angular içinde mevcut gelmektedir.
- Cross Platformlar üzerinde çalışılabilir. Uygulama web, desktop ve mobile olarak geliştirilebilir.
- Kullanıcı deneyimi olarak oldukça başarılıdır.

- Directive adı verilen yapı sayesinde HTML'e yeni etiketler veya attribute'lar eklemek mümkündür.
- Aynı değişken üzerindeki herhangi bir değişiklik two-way data binding sayesinde bütün sayfa üzerindeki ilgili alanların değişmesini tetikler.
- Test Driven Development kavramına uygundur. Unit test yazmaya olanak sağlar.

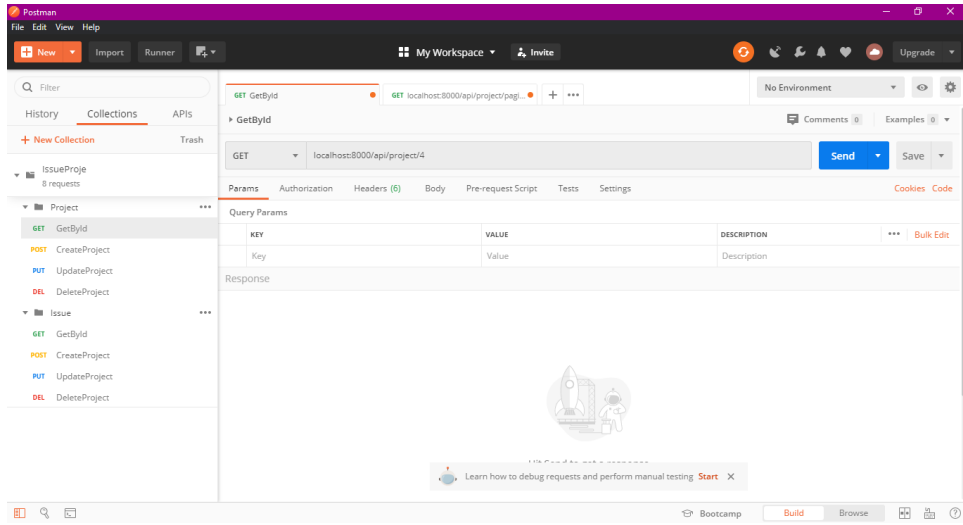
Resim 15: PostgreSQL



PostgreSQL, güçlü özellikler ve avantajlara sahip, açık kaynaklı ve tamamen ücretsiz nesne ilişkisel veri tabanı sistemidir. SQL dilinin güvenlik, depolanabilirlik ve ölçeklendirilebilirlik özelliklerinden faydalanan PostgreSQL, birçok alanda veri tabanı yöneticisi olarak da kullanılmaktadır.

- PostgreSQL kaynak kodu, açık kaynak lisansı altında serbestçe kullanılabilir. Bu da iş gereksinimlerine göre kullanma, değiştirme ve uygulama özgürlüğü sağlar.
- Coğrafi nesneleri destekler, böylece konum tabanlı servisler ve coğrafi bilgi sistemleri için kullanılabilir.
- Coğrafi nesneleri destekler, böylece konum tabanlı hizmetler ve coğrafi bilgi sistemleri için coğrafi bir veri deposu olarak kullanılabilir.
- PostgreSQL'i öğrenmek için kullanımı kolay olduğu kadar fazla eğitime ihtiyacınız yoktur.
- Hem gömülü hem de kurumsal kullanım için düşük bakım yönetimine sahiptir.

Resim 16: Postman

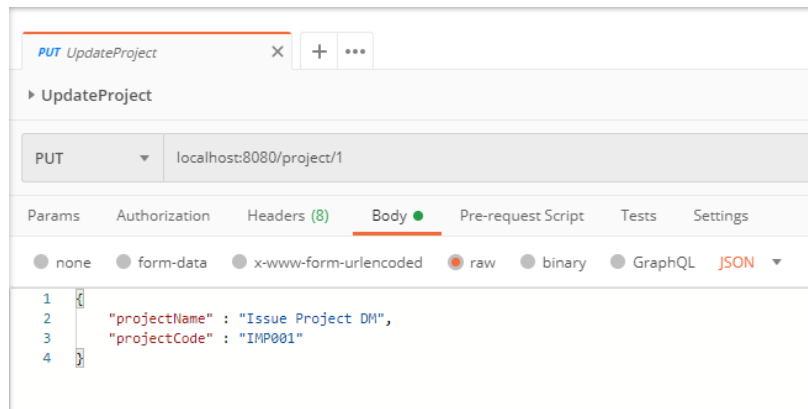


Back-end yazılım geliřtirmek için üretilmiř bir Rest Client'tir. Postman Chrome uzantısı olarak çalıřmaktadır. Bu uzantının yetenekleri API testleri için oldukça başarılı ve yeteneklidir.

API'nın yeteneklerini görmek çalıřma hızı ve durumunu test etmek veya veri içerisinde gelen deęerleri sorugulamak için Postman kullanılmıřtır.

- GET
- POST
- PUT
- DELETE

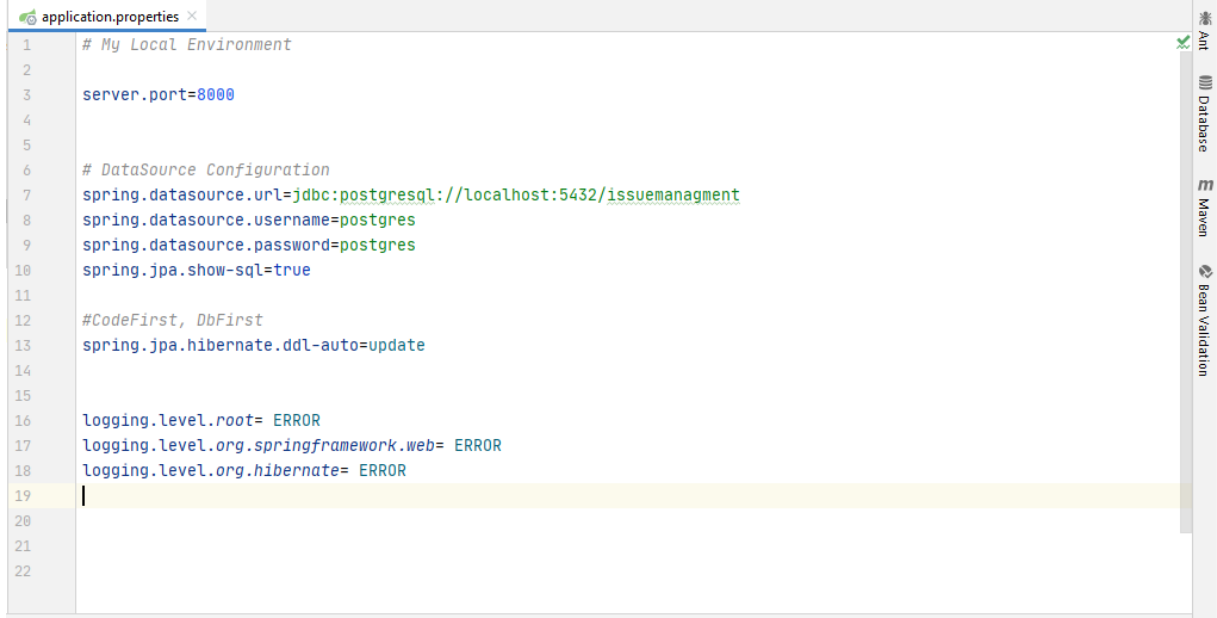
Resim 17: Postman Kod Örneęi (PUT)



3. VERİTABANI VE ER DİYAGRAMI

Önceki bölümlerde veritabanı için PostgreSQL kullanıldığı belirtilmişti. Proje ve veritabanı bağlantısı ‘application.properties’ dosyası içinde yapılmıştır.

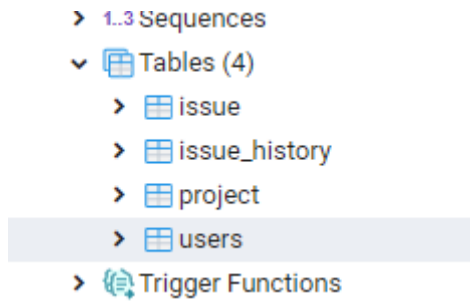
Resim 18:Veritabanı Bağlantısı



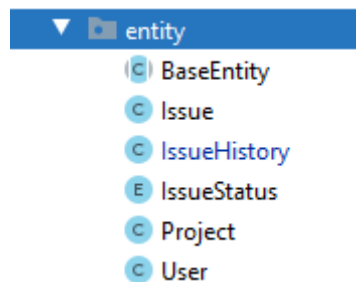
```

1  # My Local Environment
2
3  server.port=8000
4
5
6  # DataSource Configuration
7  spring.datasource.url=jdbc:postgresql://localhost:5432/issuemanagment
8  spring.datasource.username=postgres
9  spring.datasource.password=postgres
10 spring.jpa.show-sql=true
11
12 #CodeFirst, DbFirst
13 spring.jpa.hibernate.ddl-auto=update
14
15
16 logging.level.root= ERROR
17 logging.level.org.springframework.web= ERROR
18 logging.level.org.hibernate= ERROR
19
20
21
22
  
```

Resim 19:Veritabanı Tablolar PostgreSQL



Resim 20:Veritabanı Tablolar IntelliJ



- Issue Tablosu

```

@Entity
@Table(name = "issue")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Issue extends BaseEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @Column(name = "description", length = 1000)
    private String description;

    @Column(name = "details", length = 4000)
    private String details;

    @Column(name = "date")
    @Temporal(TemporalType.TIMESTAMP)
    private Date date;

    @Column(name = "issue_status")
    @Enumerated(EnumType.STRING)
    private IssueStatus issueStatus;

    @JoinColumn(name = "assignee_user_id")
    @ManyToOne(optional = true, fetch = FetchType.LAZY)
    private User assignee;

    @JoinColumn(name = "project_id")
    @ManyToOne(optional = true, fetch = FetchType.LAZY)
    private Project project;
}

```

- issue_history Tablosu

```

@Entity
@Table(name = "issue_history")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class IssueHistory extends BaseEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @JoinColumn(name = "issue_id")
    @ManyToOne(optional = true, fetch = FetchType.LAZY)
    private Issue issue;

    @Column(name = "description", length = 1000)
    private String description;
}

```

```

@Temporal(TemporalType.TIMESTAMP)
@Column(name = "date")
private Date date;

@Column(name = "issue_status")
@Enumerated(EnumType.STRING)
private IssueStatus issueStatus;

@Column(name = "details", length = 4000)
private String details;

@JoinColumn(name = "assignee_user_id")
@ManyToOne(optional = true, fetch = FetchType.LAZY)
private User assignee;
}

```

- Project Tablosu

```

@Entity
@Table(name = "project")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Project extends BaseEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @Column(name = "project_name", length = 300)
    private String projectName;

    @Column(name = "project_code", length = 30)
    private String projectCode;

    @JoinColumn(name = "manager_user_id")
    @ManyToOne(optional = true, fetch = FetchType.LAZY)
    private User manager;
}

```

- users Tablosu

```

@Entity
@Table(name = "users", indexes = {@Index(name = "idx_username", columnList = "uname")})
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class User extends BaseEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @Column(name = "uname", length = 100, unique = true)
    private String username;

    @Column(name = "pwd", length = 200)
    private String password;
}

```

```

@Column(name = "name_surname", length = 200)
private String nameSurname;

@Column(name = "email", length = 100)
private String email;

@JoinColumn(name = "assignee_user_id")
@OneToMany(fetch = FetchType.LAZY)
private List<Issue> issues;
}

```

- Sorunların bulunduğu durumlar için: IssueStatus Tablosu

```

public enum IssueStatus {

    OPEN,
    CLOSED,
    IN_REVIEW,
    IN_PROGRESS,
    RESOLVED
}

```

- Her tabloda ortak olacak: BaseEntity

```

@Getter
@Setter
@MappedSuperclass
public abstract class BaseEntity implements Serializable {

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name = "created_at")
    private Date createdAt;

    @Column(name = "created_by", length = 50)
    private String createdBy;

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name = "updated_at")
    private Date updatedAt;

    @Column(name = "updated_by", length = 50)
    private String updatedBy;

    @Column(name = "status")
    private Boolean status;
}

```

Bazı Önemli Kullanımlar:

`@Column(name = "created_at")` //kolonların özellikler

`@Temporal(TemporalType.TIMESTAMP)` //tarih alanının veri tipi (saniye dakika saat gün ay yıl)

`@MappedSuperclass` //tüm tablolarda kullanılmasını sağlıyor

`@Entity` //tabloya karşılık gelir

`@Table(name = "issue")` //tablo özelliklerini belirtme

`public class Issue extends BaseEntity` { //kalıtım yoluyla 5 kolon eklenmiş oldu

`@Id`

`@GeneratedValue(strategy = GenerationType.AUTO)` //id nin otomatik oluşturulması

`@ManyToOne(optional = true, fetch = FetchType.LAZY)` //tablolar arası ilişkilendirme

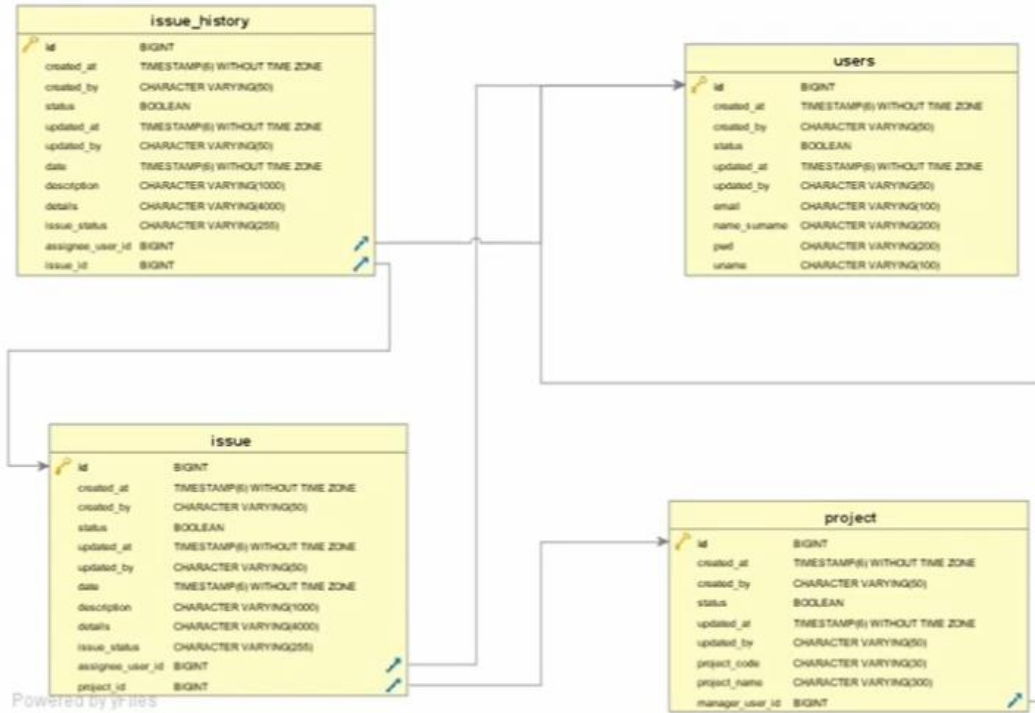
`private User assignee;` //birçok issue classı bir tane user ile ilişkilendirilebilir

`@Column(name = "uname", length = 100, unique = true)` //aynı isimden birden çok kullanıcı olamaz

`@JoinColumn(name = "assignee_user_id")`

`@OneToMany(fetch = FetchType.LAZY)` //bir kullanıcının birden fazla issuesi olabilir

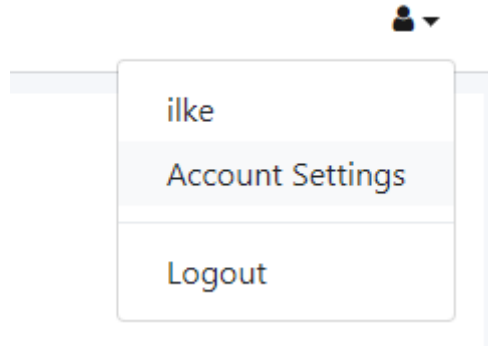
Resim 21:ER Diyagramı



4. SONUÇ

Proje planlanan problem çözümünü karşılamaktadır. Olduğu haliyle gelişmelere açıktır.

Resim 22:Kullanıcı Menü İçeriği



Kullanıcı menü içeriğine kullanıcı, ayarlar ve çıkış butonları eklendi. Geliştirme kısmında kullanıcı ve ayarlar sayfası oluşturularak uygulama ilerletilebilir.

Dashboar sayfasına farklı içerikler eklenebilir.

Kullanıcı mantığında sınırlamalara gidilerek yönetici / çalışan bölümleri ayrılabilir.

5. EK

Proje Github adresi: <https://github.com/demetmtl/issuemanagement>

6. KAYNAKÇA

<https://thinkster.io/tutorials/building-real-world-angular-2-apps/page-layout-and-routing>

<https://angular.io/start>

<https://www.techiediaries.com/angular/angular-9-tutorial-and-example/>

<https://spring.io/guides/gs/spring-boot/>

<https://medium.com/@mrpehlivan/springboot-ile-proje-olu%C5%9Fturma-2c16ce806a4f>

<https://swagger.io/solutions/api-documentation/>