**Aşağıdaki task yapmanızı rica edeceğim.** [**hamitmizrak@gmail.com**](mailto:hamitmizrak@gmail.com) **mail adresine proje kodları ve Readme.md olarak göndermenizi isteyeceğim.**

**Ödev Başlığı**

**TaskBoard API — Spring Boot Secure REST (Basic)**

**1) Senaryo (Kısa)**

“TaskBoard” adında küçük bir görev yönetim servisi yazacaksınız. Kullanıcılar projeler açar, projelere görev (task) ekler. **Rol bazlı erişim** şarttır:

* **ADMIN**: Proje/Görev üzerinde tam yetki.
* **USER**: Sadece üye olduğu projelerde okuma/yazma; diğer projelere erişim yok.

**Güvenlik seviyesi:** “Basic” pistte **HTTP Basic + InMemory kullanıcılar** (zorunlu).  
**Bonus pist:** **JWT** ile kimlik doğrulama + refresh (opsiyonel).

**2) Teknik Gereksinimler**

* **Java 21**, **Spring Boot 3.3.x**
* Bağımlılıklar: spring-boot-starter-web, spring-boot-starter-security, spring-boot-starter-validation (+ Bonus için jjwt veya spring-security-oauth2-jose)
* **Stateless** API (session yok), **CSRF kapalı** (REST için), **CORS** gerekli ise kontrollü açın.
* **Versiyonlu endpoint**: /api/v1/\*\*
* **DTO ve Validation** zorunlu (örn. @NotBlank, @Size, @Email…).
* **Hata modeli**: Tutarlı JSON hata yanıtı (örn. RFC-7807 ProblemDetail veya kendi ErrorResponse nesneniz).
* Veri saklama: **Basic pistte** memory repository serbest (Map/Service katmanı ile). İsterseniz H2/JPA da kullanabilirsiniz.

**3) Domain ve İzinler**

**3.1. Varlıklar (öneri)**

* **User**: id, username, password, roles [ADMIN|USER] *(Basic pistte inMemory yeter)*
* **Project**: id, name, description, ownerUsername, members[]
* **Task**: id, projectId, title, status [TODO|IN\_PROGRESS|DONE], assigneeUsername, dueDate

**3.2. Erişim Kuralları**

* **ADMIN**: tüm Project/Task üzerinde **CRUD**.
* **USER**:
  + Üye olduğu projeleri **listeleyebilir/görebilir**.
  + Üye olduğu projeye **task ekleyebilir/düzenleyebilir/silebilir**.
  + Üye olmadığı projeye erişmeye çalışırsa **403**.

**4) İstenen Endpoint’ler (Basic Pist – HTTP Basic)**

Not: Bonus pistte /auth/register, /auth/login (JWT) ekleyebilirsiniz.

**Project**

* GET /api/v1/projects → (ADMIN: tümü, USER: üyesi oldukları) liste
* GET /api/v1/projects/{id} → gör
* POST /api/v1/projects → **ADMIN** veya **USER (sadece kendini owner yapabilir)**
* PUT /api/v1/projects/{id} → owner veya ADMIN
* DELETE /api/v1/projects/{id} → owner veya ADMIN
* POST /api/v1/projects/{id}/members → projeye üye ekle (owner/ADMIN)

**Task**

* GET /api/v1/projects/{id}/tasks → proje görevleri (erişim kuralına tabi)
* POST /api/v1/projects/{id}/tasks → üye veya ADMIN
* PUT /api/v1/tasks/{taskId} → görevin bağlı olduğu projeye üye veya ADMIN
* DELETE /api/v1/tasks/{taskId} → üye (assignee ise) veya ADMIN

**Sağlık/Kimlik**

* GET /api/v1/health → public: "ok" döner (öğrenci testleri kolay başlatır)
* **Basic kullanıcıları**:
  + admin / admin123 → ROLE\_ADMIN
  + alice / alice123 → ROLE\_USER
  + bob / bob123 → ROLE\_USER

**5) Güvenlik Gereksinimleri (Basic Pist)**

* SecurityFilterChain bean ile:
  + /api/v1/health **permitAll()**
  + Diğer tüm istekler **authenticated()**
  + Role eşlemesi: hasRole("ADMIN"), hasAnyRole("ADMIN","USER")
  + **httpBasic(Customizer.withDefaults())**
  + csrf(AbstractHttpConfigurer::disable)
  + (Opsiyonel) @EnableMethodSecurity + @PreAuthorize kullanımı

**Bonus Pist (JWT)**

* /api/v1/auth/register (public)
* /api/v1/auth/login (public) → accessToken, refreshToken
* Bearer token ile koruma; expiration, refresh, PasswordEncoder zorunlu
* Token’a roles claim’i ekleyin; hasAuthority("ROLE\_ADMIN") haritalayın.

**6) Doğrulama ve Hata Yanıtları**

* Ör. ProjectCreateRequest { name: @NotBlank @Size(max=80), description: @Size(max=500) }
* Ör. TaskCreateRequest { title: @NotBlank, status: optional default TODO }
* Validation hatası → **400** + alan/mesaj listesi
* Yetkisiz → **401**, Yetki yok → **403**, Bulunamadı → **404**
* Tutarlı bir errorCode, message, path, timestamp alanları tavsiye.

**7) Örnek İstekler (curl)**

# Health (public)

curl -i http://localhost:8080/api/v1/health

# Basic Auth ile proje listeleme (ADMIN tümünü görür)

curl -u admin:admin123 http://localhost:8080/api/v1/projects

# USER sadece üyesi olduğu projeleri görür

curl -u alice:alice123 http://localhost:8080/api/v1/projects

# Proje oluşturma (owner: istek yapan kullanıcı)

curl -u alice:alice123 -H "Content-Type: application/json" \

-d '{"name":"AI Study","description":"weekly work"}' \

http://localhost:8080/api/v1/projects

# Üye ekleme (owner/ADMIN)

curl -u alice:alice123 -H "Content-Type: application/json" \

-d '{"username":"bob"}' \

http://localhost:8080/api/v1/projects/1/members

# Görev ekleme (üyeler veya ADMIN)

curl -u bob:bob123 -H "Content-Type: application/json" \

-d '{"title":"Read paper","assigneeUsername":"bob"}' \

http://localhost:8080/api/v1/projects/1/tasks

**8) Teslim Şartları**

* **GitHub repo**: README.md mutlaka olsun (çalıştırma adımları, örnek istekler, rol tablosu, mimari notlar).
* **Postman/Insomnia koleksiyonu** veya \*.http dosyaları.
* (Opsiyonel) **H2** veya bellek içi repo; JPA isterseniz migration gerekmez.
* **Unit/Integration test**: En az 5 test (2 güvenlik, 2 validasyon, 1 servis).
* Kod kalitesi: paketleme, servis/DTO ayrımı, logger kullanımı.