

# מערכת ניהול שמירות בבסיס צבאי - NestJS

מערכת זו מיועדת לניהול שמירות בבסיס צבאי. היא מאפשרת למפקדים להגדיר שמירות, להקצות חיילים לשמירות, ולחיילים לצפות במשמרותיהם. המערכת כוללת מנגנון הרשאות בסיסי, הרשמה והתחברות, תקשורת מאובטחת עם JWT, ובסיס נתונים SQL עם ORM של Sequelize.

---

## תרשים זרימת המערכת (Flow)

[Client] --> [HTTP Requests: login, register, shifts, assignments] --> [NestJS Server]

|

[Controllers] - מקבלים בקשות

|

[Services]

לוגיקה עסקית -

|

[Middlewares/Guards] - בדיקת הרשאות

|

[Database via Sequelize]

- **Client** שולח בקשות HTTP לשרת (למשל התחברות, יצירת שמירה, צפייה בהקצאות).
- השרת מקבל את הבקשות, עובר דרך **Controllers**, מפעיל את הלוגיקה הרלוונטית ב-**Services**.
- אם נדרש – מופעל **Guard** שבודק הרשאות לפי ה-JWT.
- המידע נשמר/נקרא מבסיס הנתונים SQL באמצעות **Sequelize**.

## שלב 1: הקמת פרויקט NestJS והגדרת API בסיסי

### משימות:

- יצירת פרויקט חדש עם `nest new`
- יצירת מודולים בסיסיים: `auth, users, shifts, assignments`
- הגדרת ראווטים פשוטים שמחזירים תשובה לדוגמה (mocked)
- אימות עם JWT (עדיין ללא DB)

### בסיום שלב 1:

- השרת רץ ומגיב לבקשות
- מוגדר JWT פשוט להעברת טוקן
- קיימת הפרדה ראשונית לפי `roles` (חייל / מפקד)

## שלב 2: הוספת לוגיקה עסקית והגנה על ראווטים

### משימות:

- יצירת Guards / Middlewares להרשאות (role-based)
- יצירת שירותים מדומים (Services עם Array in-memory במקום DB)
- חיבור **bcrypt** להצפנת סיסמאות
- ולידציה של קלטים עם `class-validator`
- טיפול בשגיאות ברמה בסיסית

### בסיום שלב 2:

- אפשר לנהל שמירות והקצאות בצורה לוקאלית (מערכים מקומיים)
- חייל רואה רק את הנתונים שלו, מפקד רואה הכל
- כל הנתונים נשמרים בזיכרון (תשתית מוכנה להחלפה ב-DB)

## שלב 3: חיבור למסד נתונים

### משימות:

- התקנת חבילות לצורך חיבור ל DB.
  - יצירת קונפיגורציית חיבור ל-Cloud PostgreSQL + sequelize
  - יצירת מודלים:
- User (id, name, email, password, role)
  - Shift (id, startTime, endTime, location)
  - Assignment (id, userId, shiftId)

- הגדרת קשרים בין המודלים

### בסיום שלב 3:

- הנתונים עוברים לשמירה במסד נתונים אמיתי
- Sequelize מייצר את הטבלאות אוטומטית
- ניתן לבצע קריאות וכתיבה ל DB

## שלב 4: בדיקות, טיפול בשגיאות והודעות שגיאה מותאמות.

### משימות:

- הודעות שגיאה מותאמות
- בדיקות API ידניות עם Postman או curl

### בסיום שלב 4:

- המערכת עובדת בצורה מלאה
- יש אבטחה בסיסית + הפרדת הרשאות
- המפקד שולט במערכת, החייל משתמש מוגבל

## 1. מערכת התראות והודעות

**תיאור:** כאשר חייל מקבל הקצאה לשמירה – תישלח לו הודעה (לדוגמה: אימייל או הודעה בלוח הודעות פנימי במערכת). מפקד יכול לשלוח הודעות כלליות או אישיות.

- תמיכה ב-notification center למשתמש.
  - הודעות יכולות להיות "נקראו / לא נקראו".
  - אפשרות למפקד לשלוח תזכורות אוטומטיות 12 שעות לפני תחילת השמירה.
- 

## 2. יומן אישי ושבועי לשמירות

**תיאור:** חייל יכול לראות את הלוח האישי שלו בצורת טבלה שבועית. מפקד רואה לוח שמירות של כלל החיילים.

- תצוגה חודשית או שבועית.
  - תמיכה בייצוא ליומן Google/Outlook.
  - חיפוש לפי תאריכים / מיקומים.
- 

## 3. בקשות להחלפת שמירה

**תיאור:** חייל יכול לשלוח בקשה להחלפת שמירה עם חייל אחר. המפקד מאשר או דוחה.

- ניהול בקשות החלפה.
  - בדיקת זמינות בזמן אמת.
  - מערכת חוקים (אי אפשר להחליף אם כבר עבר אישור, או פחות מ-X שעות מראש).
- 

## 4. ניתוח עומסים ודוחות

**תיאור:** מפקדים יכולים להפיק דוחות על מספר שמירות לכל חייל, זמינות, חריגות, איחורים וכו'.

- גרפים וסטטיסטיקות (כמה שמירות עשה כל חייל, איזו תקופה הייתה עם הכי הרבה שמירות).
  - ייצוא ל-CSV או PDF.
  - פילוח לפי תפקידים, זמני יום/לילה, ימים עמוסים.
- 

## 5. מערכת חכמת הקצאה (Smart Scheduler)

**תיאור:** אלגוריתם חכם שמחלק שמירות לפי זמינות, עומס, תפקיד, ובקשות מיוחדות.

- חיילים עם הכי פחות שמירות מקבלים עדיפות.
  - התחשבות בבקשות אישיות.
  - מניעת חלוקה לא שוויונית בין חיילים.
- 

## 6. תמיכה בזמינות אישית

**תיאור:** חיילים יכולים להגדיר ימים או שעות בהן אינם זמינים לשמירה (לימודים, הקלות רפואיות וכו').

- המפקד רואה את טווחי הזמינות כשהוא מקצה שמירות.
  - אפשרות להגיש חופשה ולהמתין לאישור.
- 

## 7. לוגים ומעקב אחר פעולות

**תיאור:** כל פעולה קריטית נרשמת – התחברות, שינוי סיסמה, הקצאת שמירה, מחיקת נתונים וכו'.

- מימוש מערכת audit logs לפי משתמשים.
  - צפייה בפעולות לפי טווחי זמן / סוגים.
  - עוזר בדיהוי תקלות או ניצול לרעה של ההרשאות.
- 

## 8. תמיכה בריבוי בסיסים

**תיאור:** מערכת אחת שיכולה לשרת כמה בסיסים צבאיים, לכל בסיס יש את המפקדים והחיילים שלו.

- כל חייל שייך לבסיס מסוים (foreign key).
  - ניהול הרשאות לפי בסיס.
  - מפקדים רואים רק נתוני הבסיס שלהם.
- 

## 9. ניהול תפקידים מתקדם

**תיאור:** תמיכה בתפקידים מרובים: חייל, סמל, קצין, קצין אבטחה, מפקד בסיס.

- הרשאות דינמיות לפי role.
  - סמל יכול לאשר בקשות החלפה, אך לא ליצור שמירות חדשות.
  - קצין יכול לראות סטטיסטיקות.
- 

## 10. דו"חות חריגות

**תיאור:** אם חייל איחר לשמירה, לא הופיע, עזב מוקדם – המפקד יכול לדווח על כך, והמערכת עוקבת אחרי חריגות.

- אפשרות להזין הערות.
  - סטטיסטיקת חריגות לכל חייל.
  - דיווח תקופתי לקצין משמעת.
- 

## 11. מנגנון אישורים היררכי

**תיאור:** פעולות מסוימות דורשות אישור ממספר דרגים (למשל: הקצאה מיוחדת לשמירה מחוץ לבסיס).

- אפשרות לשליחת בקשה לשרשרת אישורים.
- עקבות לכל שלב: ממתין לאישור, אושר, נדחה.
- ניתן לקבוע תנאים לפי סוג הבקשה.

---

## 12. REST + GraphQL APIs

**תיאור:** תמיכה גם ב-GraphQL לצורך אינטגרציה עם אפליקציות מתקדמות (כמו מובייל).

- הגדרת schema מסודר.
- אפשרות לבחירת שיטת גישה.
- דוקומנטציה אוטומטית עם GraphQL Playground / Swagger.