

מערכת ניהול שמירות בסיס צבאי - NestJS

מערכת זו מיועדת לניהול שמירות בסיס צבאי. היא מאפשרת למפקדים להגדיר שמירות, להקנות חילים לשמירות, ולהחילם לצפות במשמרותיהם. המערכת כוללת מנגנון הרשות 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] -
בדיקה הרשות

|
[Database via Sequelize]

- Client שולח בקשות HTTP לשרת (למשל התחברות, ייצרת שמירה, צפיה בהקצאות).
- השירות מקבל את הבקשות, עבר דרכ Controllers, מפעיל את הלוגיקה הרלוונטיית ב-Services.
- אם נדרש – מופעל Guard שבודק הרשות לפני ה-JWT.
- המידע נשמר/נקרא מבסיס הנתונים SQL באמצעות Sequelize.

שלב 1: הקמת פרויקט NestJS והגדרת API בסיסי

משימות:

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

בסיום שלב 1:

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

שלב 2: הוספת לוגיקה עסקית והגנה על ראותים

משימות:

- יצרת `Guards / Middlewares` להרשות (role-based)
- יצרת שירותים מדומים (`Services`) עם `in-memory` (DB במקום Array)
- חיבור `bcrypt` להצפנה סיסמאות
- ולידציה של קלטים עם `class-validator`
- טיפול בשגיאות בرمאה בסיסית

בסיום שלב 2:

- אפשר לנויל שמירות והקצאות בצורה לوكאלית (מערכות מקומיים)
- חיל רואה רק את הנתונים שלו, מפקד רואה הכל
- כל הנתונים נשמרים בזיכרון (תשתיית מוכנה להחלפה ב-DB)

שלב 3: חיבור למסד נתונים

משימות:

- התקנת חבילות לצורך חיבור ל-DB.

PostgreSQL + sequelize - Cloud - יצרת קונפיגורציה חיבור ל-

- יצרת 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 לשימרה במסד נתונים אמיתי

Sequelize מיציר 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 curl או Postman

בסיום שלב 4:

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

משימות אקסטרה:

1. מערכת התראות והודעות

תיאור: כאשר חיל מקבל הקenza לשמייה – תישלח לו הודעה (לדוגמה: אימיל 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: חיל יכול לראות את הלוג' האיש שלו בצורה טבלה שבועית. מפקד רואה לוח שמירות של כלל החילות.

- תצוגה חודשית או שבועית.
 - תמיכה ביצוא ליום Google/Outlook.
 - חיפוש לפי תאריכים / מקומות.
-

3. בקשות להחלפת שמירה

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

- ניהול בקשות החלפה.
 - בדיקת זמינות בזמן אמיתי.
 - מערכת חוקים (איך 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 חיל, זמינות,Chrigeot, איחורים וכו'.

- גرافים וסטטיסטיקות (כמה שמירות עשה כל חיל, 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 שמורותות לפי זמינות, עומס, תפקיד, ובקשות מיוחדות.
- חילים עם הכי פחות שמורותות מקבלים עדיפות.
 - התחשבות בבקשות אישיות.
 - מניעת חלוקה לא שוויונית בין חילים.
-

6. תמייה בזמןנות אישית

- תיאור: חילים יכולים להגדיר ימים או שעות בהן אינם זמינים לשמירה (לימודים, הקלות רפואיות וכו').
- המפקד ראה את טווחי הזמינות כשהוא מנסה לשמור.
 - אפשרות להגיש חופשה ולהמתין לאישור.
-

7. לוגים ומעקב אחר פעולות

- תיאור: כל פעולה קրיטית נרשמת – התחברות, שינוי סיסמה, הקצאת שמירה, מחיקת נתונים וכו'.
- מימוש מערכת logs audit לפ' משתמשים.
 - צפיה בפעולות לפי טווחי זמן / סוגים.
 - עוזר בזיהוי התקלות או ניצול לרעה של הרשאות.

8. תמייה בריבוי בסיסים

תיאור: מערכת אחורית שיכולה לשרת כמה בסיסים צבאיים, לכל בסיס יש את המפקדים והחייבים שלו.

- כל חיל שיר לבסיס מסוים (foreign key).
 - ניהול הרשותות לפי בסיס.
 - מפקדים רואים רק נתונים הבסיס שלהם.
-

9. ניהול תפקידים מתקדם

תיאור: תמייהה בתפקידים רבים: חיל, סמל, קצין, קצין אבטחה, מפקד בסיס.

- הרשותות דינמיות לפי role.
 - סמל יכול לאשר בקשות החלפה, אך לא ליצור שמירויות חדשות.
 - קצין יכול לראות סטטיסטיות.
-

10. דוחות חריגות

תיאור: אם חיל אחר לשמירה, לא הופיע, עזב מוקדם – המפקד יכול לבדוק על קר, והמערכת עוקבת אחרי חריגות.

- אפשרות להזין הערות.
 - סטטיסטיות חריגות לכל חיל.
 - דיווח תקופתי לקצין משמעת.
-

11. מנגנון אישורים היררכי

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

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

REST + GraphQL APIs .12

טיואר: תמייה גם ב-GraphQL לצורך אינטגרציה עם אפליקציות מתקדמות (כמו מובייל).

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