

תרגיל פרויקט – תיבת תלונות אונימיות בסיסי צבאי (+ Express API)

רקע

זה"ל בוחן מערכת שמאפשרת לחילאים לשלוח תלונות אונימיות באופן דיגיטלי.
המערכת **לא שומרת שם/מזהה משתמש**, רק:

- אוכל / ציוד / פקודות / אחר (category)
- טקסט התלונה (message)
- תאריך יצירה (createdAt)

רק מפקדים שמצוינים **סיסמת אדמין** יכולים לצפות ברשימה התלונות.

מטרות לימוד

צד לקוח (React)

- עבדה עם **React Router** (ניווט בין דפים)
- עבדה עם **טפסים** (controlled components)
- שליחת בקשות לשרת בעזרת **fetch / axios**
- ניהול מצב בסיסי (**useState, useEffect**)
- בניה של נקי ורסתפונסיבי

צד שרת (Express API)

- בניית **REST API** במקום הגשת HTML סטטי
 - שמירה למסד נתונים בענן (**Supabase MongoDB Atlas**) או routes / controllers / services / models
 - הפרדה לארכיטקטורה: services / models
 - בדיקות Unit + Integration Unit לשרת
-

ארQUITקטורה ותיקיות מומלצות

מנוריפו (מומלץ)

```
complaints-system/
  └── client/          # React app
      ├── src/
      │   ├── pages/
      │   │   ├── HomePage.jsx
      │   │   ├── SubmitComplaintPage.jsx
      │   │   ├── AdminLoginPage.jsx
      │   │   ├── AdminComplaintsPage.jsx
      │   │   └── ErrorPage.jsx
      │   ├── components/
      │   │   ├── Layout.jsx
      │   │   ├── ComplaintForm.jsx
      │   │   ├── ComplaintsTable.jsx
      │   │   └── ProtectedRoute.jsx
      │   ├── api/
      │   │   └── complaintsApi.js
      │   ├── styles/
      │   │   └── app.css
      │   └── main.jsx
      └── tests/
          └── (React tests)

  └── server/          # Express API
      ├── src/
      │   ├── routes/
      │   │   └── complaints.routes.js
      │   ├── controllers/
      │   │   └── complaints.controller.js
      │   ├── services/
      │   │   └── complaints.service.js
      │   ├── models/
      │   │   └── complaint.model.js
      │   ├── middlewares/
      │   │   ├── requireAdminAuth.js
      │   │   └── errorHandler.js
      │   ├── db/
      │   │   └── connect.js
      │   └── app.js
      ├── tests/
      │   └── (server tests)
      └── .env
          └── package.json
```

דרישות פונקציונליות – צד לקוח (React)

(1) Home Page (/)

מציג:

- הסבר קצר על מטרת המערכת
- כפתור/לינק: "שליחת תלונה" → מעבר ל-/submit
- אזכור "מפקדים בלבד":
 - שדה סיסמה
 - כפתור "כניסה לאדמין" → מעבר ל-/admin/login או ביצוע התחברות ואז ניתן ל-/admin

dagsh: אין CSS דרך Express כמו בפרויקט הישן; ה-UI מוגש ע"י React.
השרת עכשו אחראי ל-API בלבד.

(2) Submit Complaint Page (/submit)

គורתה: "שליחת תלונה אונזימית"
טופס (Controlled):

- קטגוריה: אוכל / ציוד / פקסיות / אחר
- תוכן תלונה (מינימום תווים מומלץ)
- כפתור שליחה
- אחרי הצלחה:
 - הודעת הצלחה
 - איפוא טופס / ניתן חזרה לדף הבית

(3) Admin Login Page (/admin/login)

- שדה סיסמה
- כפתור "התחברות"
- אם הסיסמה נכונה → הודעת שגיאה + אין גישה לדף התלונות

(4) Admin Complaints Page (/admin)

- מוצג רק אם המשתמש עבר איזמות אדמין
- מציג רשימת תלונות בטבלה/קרטיסים:
 - category
 - message
 - createdAt (פורמט קרייא)
- בניית 필터 לפי category + מונה לכל קטgorיה

(5) הגנה על נתיבים (Client)

- שמונע גישה לה-ProtectedRoute admin אם אין איזמות

- אם אין אימות → ניתוב ל- `/admin/login`
 - חשוב: הגנה בלבד לקו **לא מספקת**. חיברים גם הגנה מצד שרת (ראו בהמשך).
-

דרישות פונקציונליות – צד שרת (Express API)

API Endpoints (מינימום נדרש)

1. POST `/api/complaints`
 - קולט: { category, message }
 - שומר במסד: { category, message, createdAt }
 - מחזקת תשובה של הצלחה (אפשר גם את האובייקט שנשמר)
 2. POST `/api/admin/login`
 - קולט: { password }
 - משווה ל- `ADMIN_PASSWORD` מתוך `env`.
 - אם נכון:
 - מחזקת "אימות הצליח"
 - ונותן מנגן אימות (אחד מה הבאים):
- Option A (פשוט): טוקן קצר חיים (JWT) שמורזר ללקוט
- Option B: Session cookie
- אם לא נכון: **Unauthorized 401**
 - 3. GET `/api/complaints`
 - דורש אימות אדמין (middleware)
 - מחזקת רישימת תלונות (ממיינית מהחדש לשן)

כללי אבטחה בסיסיים

- הסיסמה נשמרת רק ב- `env`
 - אין לחשוף URI למסד נתונים בקוד
 - אסור לשמור שם מזהה משתמש של החיל (לא IP, לא `userId`, לא "name")
-

דרישות עיצוב (React UI)

המערכת צריכה להיראות נקייה, אחידה ורספונסיבית:

- רקע עדין (למשל `#f0f0f0`)
- Container מרכזי לבן עם `padding + border-radius + shadow` עדין
- כותרות ממורכזות
- הטופס בטור אחד, עם רווחים ברורים
- כפתורים ברורים (Primary לפעולה הראשית)
- התאמה למוביל: שימוש ביחסות יחסיות + `max-width`

דרישות בדיקות

בדיקות צד שרת

Unit Tests

- בדיקה לוגיקה של יצירת תלונה (validation בoS: category חוק, message לא ריק)
- בדיקה לשירות שמויף תלונה למסד (עם mocking למודל/DB)

Integration Tests

- DB test / in-memory → מבחן הצלחה ושומר במסד (אפשר / שומר במסד (mock
(mock
401/403 GET /api/complaints •
401 POST /api/admin/login •
401 POST /api/complaints עם סיסמה נכונה ואז POST /api/admin/login •
רשימה

בדיקות צד לקוח (React)

- בדיקה שטופה "שליחת תלונה" שלוח בקשה ומציג הודעה הודעת הצלחה (עם mock ל-*I*-API)
- בדיקה ש-*ProtectedRoute* חוסם `admin` ללא אימות
- בדיקה ש-*page Admin* מציג תלונות אחרות (fetch) (mock)

שלבי עבודה מוצעים (פאות)

פאה 1 — הקמה ותשתיות

- ייצרת `server` (Express-*I*) (`client` (React))
- הגדרת `.env` לשרת
- חיבור למסד (MongoDB Atlas / Supabase)

פאה 2 — API בסיסי

- מודל `Complaint`
- `POST /api/complaints`
- בדיקות בסיס ליצירה

פאה 3 — UI בסיסי

- `Home + Submit page`
- טופס מלא + קרייה ל-*I*-API

פaza 4 — Admin Auth + הגנה

- POST /api/admin/login
- שמירת auth בלקות (token / session)
- מוגן GET /api/complaints
- בלקות ProtectedRoute

פaza 5 — בדיקות מקיפות + ניקיון ארכיטקטוני

- בדיקות Unit + Integration לשרת
- בדיקות React
- טיפול שגיאות מסודר (ErrorPage / toast / messages)

אתגרים למתקדמים (לא חובה)

- הוספה DELETE /api/complaints/:id (מוגן אדמין)
- גրף תלונות לפי קטגוריה (Client: chart)
- "נשלח לפני X ימים" (human readable time)
- api/complaints?category=food/ פילטרים + API JSON

screenshots:

Home Page

תיריבת תלונות אוננימיות בבסיס צבאי

שלחו תלצתה באורה אונונימית ונשווה.

שליחת תלצתה

תפקידים בלבה

סימון: **כיסית לאג'ין**

Submit Complaint Page

שליחת תלצתה אונונימית

חומר תחילתי:

* ויק חלונה:

שליחת

תלצתה נשלחה בעיתה

Admin Login Page

כניסה לאהו האדמוני

סיסמה:

התחברות

סיסמת שפיה, גנט שוב

Admin Complaints Page

רשימת תלונות

כלוחות ציד ציה סדר ובלי

חומר	שם חלונה	מושג תאורטי
אוכלי	האכל בהור חור אוכלי לא מספקו רדי.	2024, נוב, 1: 4
ציוויל	חסר שוד שוד בשוחחת רישם.	2024, מרץ, 3: 3
פקודות	פקודות לא ברוחם מכבלו.	2024, מרץ, 3: 3
אחר.	או שכל החלטתו מוקה.	2024, מרץ, 09:20