به نام خدا

شروع پروژه مرن استک شرکت کارن به اسم karen-app

ابتدا از بک اند شروع میکنیم   
من در این پروژه یه چند بخش مهم دارم یکی معرفی محصولات هستش  
که اسم همه شون تابلو فرمان آسانسور هستش

و در 4 دسته بندی قرار میگیرند : هیدرولیک ، روم لس ، گیرلس و گیر باکس (gearbox)

و همه شون عکس دارند و درایو های متفاوتی با توان های متفاوتی هستند   
  
از طرفی دیگه فرم رجیستر و لاگین داریم و ادمین هم باید بتونه محصولات رو اضافه و حذف کنه و بقیه نمیتونند اما میتونند به صورت یوزر وارد بشن  
ما بخش ارتباط با ما رو داریم که خیلی مهمه و اطلاعات تماس و محل فروش و شرکت اونجا ثبت میشه   
در کل ادمین به عنوان یوزر خیلی مهم به حساب میاد

برای بک اند از اکسپرس و مونگو دیبی استفاده میشه   
برای فرانت اند از ری اکت و برای استایل دهی از تلویند

**گزارش پروژه karen-app تا الان**

**۱. تعریف پروژه**

* **هدف:** ساخت یک وبسایت محصولات تابلو فرمان آسانسور با قابلیت‌های حرفه‌ای برای کاربران و ادمین.
* **ویژگی‌ها:**
  1. محصولات در ۴ دسته‌بندی: هیدرولیک، روم‌لس، گیرلس، گیر باکس
  2. هر محصول شامل:
     + نام محصول
     + تصویر
     + برند درایو (مثل Yaskawa)
     + توان مشخص (مثلاً 7.5 یا 11 وات)
  3. کاربران:
     + **ادمین:** اضافه، ویرایش و حذف محصولات
     + **یوزر:** فقط مشاهده محصولات
  4. فرم رجیستر و لاگین
  5. بخش «ارتباط با ما» با اطلاعات تماس و آدرس شرکت
* **استک فناوری:** MERN (MongoDB, Express, React, Node.js) + TailwindCSS برای استایل

**۲. طراحی دیتابیس (MongoDB)**

**User**

{

username: String,

email: String,

password: String, // hashed

role: String // 'admin' | 'user'

}

**Product**

{

name: String,

category: String, // hydraulic, romless, gearless, gearbox

image: String,

drive: String, // برند درایو

power: Number // توان مشخص

}

**ContactInfo**

{

address: String,

phone: String,

email: String,

company: String

}

**۳. ساختار پروژه بک‌اند**

karen-app-backend/

│-- config/db.js # اتصال به MongoDB

│-- models/ # مدل‌ها (User, Product, ContactInfo)

│-- routes/ # روت‌ها (authRoutes, productRoutes)

│-- controllers/ # منطق روت‌ها (authController, productController)

│-- middleware/ # JWT auth و محدودیت ادمین

│-- server.js # راه‌اندازی Express

│-- .env # متغیرهای محیطی

**۴. اقدامات انجام شده**

1. **راه‌اندازی پروژه و نصب پکیج‌ها**
2. npm init -y
3. npm install express mongoose dotenv bcryptjs jsonwebtoken cors
4. npm install --save-dev nodemon
5. **ایجاد اتصال به MongoDB (لوکال)**
   * فایل config/db.js ایجاد شد
   * متغیر محیطی:
   * MONGO\_URI=mongodb://localhost:27017/karen-app
   * اتصال به MongoDB در server.js انجام شد
6. **ایجاد مدل User**
   * شامل فیلدهای username، email، password، role
   * رمزگذاری با bcrypt
7. **ایجاد روت Auth (Register & Login)**
   * JWT صادر می‌شود
   * role در توکن ذخیره می‌شود
   * آماده برای احراز هویت و مدیریت دسترسی‌ها
8. **راه‌اندازی سرور**
   * فایل server.js ساخته شد
   * Middleware: express.json() و cors
   * روت‌های /api/auth و /api/products آماده اتصال
9. **متغیرهای محیطی نهایی**
10. PORT=5000
11. MONGO\_URI=mongodb://localhost:27017/karen-app
12. JWT\_SECRET=supersecretkey

**۵. وضعیت فعلی**

* سرور Express لوکال آماده اجرا است
* اتصال به MongoDB برقرار شده
* Auth پایه (رجیستر و لاگین) آماده است
* ساختار فولدرها و پروژه مشخص و مرتب است
* JWT و نقش کاربر (user/admin) آماده برای middleware

**۶. مرحله بعدی (پیشنهاد ما)**

**ساخت مدل Product و روت‌های CRUD با محدودیت دسترسی ادمین:**

1. مدل Product با فیلدهای:
   * name, category, image, drive, power
2. روت‌های محصولات:
   * GET /api/products → مشاهده همه محصولات (عمومی)
   * POST /api/products → اضافه کردن محصول (ادمین)
   * PUT /api/products/:id → ویرایش محصول (ادمین)
   * DELETE /api/products/:id → حذف محصول (ادمین)
3. استفاده از **middleware JWT و admin** برای محدود کردن دسترسی
   * اتصال به دیتابیس لوکال karen-app برقرار شد
4. **مدل User**
   * شامل username, email, password, role
   * هش کردن پسورد قبل از ذخیره
   * متد comparePassword برای اعتبارسنجی پسورد
5. **کنترلر Auth**
   * register برای ثبت نام کاربر
   * login برای ورود و تولید JWT
   * حذف پسورد قبل از ارسال اطلاعات به کلاینت
6. **روت‌های Auth**
   * /api/auth/register
   * /api/auth/login
7. **مدل Product**
   * شامل name, category, image, drive, power
   * category محدود به ۴ مقدار معتبر
   * timestamps فعال برای تاریخ ایجاد و بروزرسانی
8. **کنترلر CRUD محصولات**
   * getProducts, getProductById, addProduct, updateProduct, deleteProduct
   * بررسی صحت فیلدها و وضعیت HTTP مناسب

**۵. وضعیت فعلی**

* مدل User، کنترلر Auth و روت‌ها آماده و تست شده ✅
* مدل Product و کنترلر CRUD محصولات آماده ✅
* سرور Express لوکال آماده اجرا ✅
* مرحله بعدی: ساخت **middleware protect و admin** و اعمال روی روت‌های محصولات برای محدود کردن دسترسی ادمین ✅

**۶. مرحله بعدی پیشنهادی**

1. ساخت middleware احراز هویت (protect) و بررسی نقش ادمین (admin)
2. اعمال این middleware روی روت‌های محصولات:
   * فقط ادمین بتواند POST, PUT, DELETE انجام دهد
   * همه کاربران بتوانند GET محصولات را مشاهده کنند

سلام من ری اکت و اکسپرس رو قبلا یاد گرفتم و حالا مدتی گذشته و بخشی از اون ها رو فراموش کردم فکر کردم یه پروژه انجام بدم تا مرور و یادآوری بشه و خودم رو به چالش بذارم   
میخوام یه پروژه طراحی کنیم و سعی بشه همه چی توش باشه  
من میخوام که فرانت این پروژه رو با ری اکت و استایل دهی اون رو با تلویند سی اس اس باشه و برای بک اند از دیتابیس مونگو دیبی استفاده کنیم و همین طور از اکسپرس   
سعی کنیم پروژه طراحی شده کامل باشه   
پیشنهادی داری و میتونی منتور من در این چالش باشی ؟