**مستندات پروژه رصدگر کلیدواژه (Keyword Tracker)**

**۱. معرفی پروژه**

این پروژه یک وب اپلیکیشن برای رصد جایگاه کلیدواژه‌های وب‌سایت‌های مختلف در نتایج جستجوی گوگل است. بک‌اند این پروژه یک API قدرتمند و امن را فراهم می‌کند که قابلیت‌های زیر را ارائه می‌دهد:

* مدیریت کاربران با سطوح دسترسی مختلف (ادمین، مدیر، مشاهده‌گر).
* مدیریت وب‌سایت‌ها، کلیدواژه‌ها و کلیدهای API سرویس‌های خارجی.
* سیستم احراز هویت امن مبتنی بر توکن (JWT).
* قابلیت بررسی دستی و خودکار (روزانه) جایگاه کلیدواژه‌ها.
* ذخیره‌سازی تاریخچه جایگاه‌ها برای تحلیل روند.

**۲. فناوری‌های استفاده شده**

* **محیط اجرا:** Node.js
* **فریم‌ورک وب:** Express.js
* **پایگاه داده:** SQLite (با ORM قدرتمند Sequelize)
* **احراز هویت:** JSON Web Tokens (JWT)
* **هشینگ رمز عبور:** bcrypt
* **زمان‌بندی وظایف:** node-cron
* **ارسال درخواست HTTP:** axios
* **فریم‌ورک تست:** Jest & Supertest

**۳. راه‌اندازی پروژه در محیط توسعه**

**پیش‌نیاز:**

* نصب **Node.js** (نسخه LTS پیشنهاد می‌شود).

**مراحل راه‌اندازی:**

1. **نصب بسته‌ها:** در پوشه backend، دستور زیر را اجرا کنید:

Bash

npm install

1. **ایجاد فایل محیطی:** یک فایل به نام .env در ریشه پوشه backend بسازید و محتوای زیر را در آن قرار دهید:
2. JWT\_SECRET=your\_super\_secret\_and\_long\_key\_here
3. **اجرای سرور توسعه:**

Bash

npm run dev

سرور در آدرس http://localhost:3000 اجرا خواهد شد.

1. **اجرای تست‌ها:** برای اطمینان از صحت عملکرد تمام بخش‌ها، دستور زیر را اجرا کنید:

Bash

npm test

**۴. معماری پروژه**

ساختار فایل‌های پروژه برای تفکیک مسئولیت‌ها و تمیز نگه داشتن کد طراحی شده است:

* **/config**: شامل فایل‌های پیکربندی، مانند اطلاعات سرویس‌های خارجی.
* **/middleware**: شامل میدل‌ورهای Express، مانند authMiddleware برای چک کردن توکن.
* **/models**: تعریف ساختار جداول پایگاه داده (Schema) با استفاده از Sequelize.
* **/routes**: تعریف تمام مسیرهای API (Endpoints) برنامه.
* **/services**: شامل منطق‌های اصلی و قابل استفاده مجدد، مانند سرویس rankChecker.
* **/tests**: شامل تمام فایل‌های تست خودکار پروژه.
* **app.js**: قلب اصلی اپلیکیشن Express که تمام بخش‌ها را به هم متصل می‌کند.
* **server.js**: فایل ورودی برنامه که سرور را اجرا می‌کند.
* **database.js**: مسئول ایجاد و مدیریت اتصال به پایگاه داده.
* **scheduler.js**: مسئول تعریف و اجرای جاب زمان‌بندی شده روزانه.

**۵. ساختار پایگاه داده (Schema)**

* **User**: اطلاعات کاربران (نام کاربری، رمز عبور هش شده، نقش).
* **Website**: اطلاعات وب‌سایت‌ها (نام، آدرس URL).
* **Keyword**: اطلاعات کلیدواژه‌ها (متن کلیدواژه، تاریخ آخرین بررسی).
* **ApiKey**: اطلاعات کلیدهای API سرویس‌های خارجی (نام سرویس، کلید، سهمیه).
* **RankHistory**: تاریخچه نتایج بررسی جایگاه (رتبه، آدرس صفحه، تاریخ).

**ارتباطات:**

* User و Website رابطه **چند-به-چند** دارند (یک کاربر به چند سایت و یک سایت به چند کاربر دسترسی دارد).
* Website و Keyword رابطه **یک-به-چند** دارند (یک سایت کلیدواژه‌های زیادی دارد).
* Keyword و RankHistory رابطه **یک-به-چند** دارند (یک کلیدواژه تاریخچه‌های زیادی دارد).

**۶. مستندات مسیرهای API (Endpoints)**

**احراز هویت و کاربران (/api/users)**

* **POST /register**
  + **توضیح:** ثبت نام یک کاربر جدید. اولین کاربر به صورت خودکار admin و بقیه viewer خواهند بود.
  + **دسترسی:** عمومی
  + **بدنه درخواست:** { "username": "string", "password": "string" }
  + **پاسخ موفق:** 201 Created
* **POST /login**
  + **توضیح:** ورود کاربر و دریافت توکن JWT.
  + **دسترسی:** عمومی
  + **بدنه درخواست:** { "username": "string", "password": "string" }
  + **پاسخ موفق:** 200 OK به همراه token.
* **GET /profile**
  + **توضیح:** دریافت اطلاعات کاربر لاگین کرده.
  + **دسترسی:** کاربر احراز هویت شده (هر نقشی).
  + **نیاز به توکن:** بله
* **GET /**
  + **توضیح:** دریافت لیست تمام کاربران سیستم.
  + **دسترسی:** فقط ادمین.
  + **نیاز به توکن:** بله
* **GET /my-websites**
  + **توضیح:** دریافت لیست وب‌سایت‌هایی که کاربر فعلی به آن‌ها دسترسی دارد.
  + **دسترسی:** کاربر احراز هویت شده (هر نقشی).
  + **نیاز به توکن:** بله

**مدیریت وب‌سایت‌ها (/api/websites)**

* **POST /**
  + **توضیح:** افزودن یک وب‌سایت جدید به سیستم.
  + **دسترسی:** فقط ادمین.
  + **نیاز به توکن:** بله
  + **بدنه درخواست:** { "name": "string", "url": "string (valid URL)" }
* **GET /**
  + **توضیح:** دریافت لیست تمام وب‌سایت‌های موجود در سیستم.
  + **دسترسی:** فقط ادمین.
  + **نیاز به توکن:** بله
* **POST /:websiteId/assign**
  + **توضیح:** اختصاص دسترسی یک وب‌سایت به یک کاربر.
  + **دسترسی:** فقط ادمین.
  + **نیاز به توکن:** بله
  + **پارامتر آدرس:** websiteId
  + **بدنه درخواست:** { "userId": integer }

**مدیریت کلیدواژه‌ها (/api/websites/:websiteId/keywords)**

* **POST /:websiteId/keywords**
  + **توضیح:** افزودن یک کلیدواژه جدید به یک وب‌سایت مشخص.
  + **دسترسی:** ادمین یا مدیر سایت.
  + **نیاز به توکن:** بله
  + **پارامتر آدرس:** websiteId
  + **بدنه درخواست:** { "text": "string" }
* **GET /:websiteId/keywords**
  + **توضیح:** دریافت لیست تمام کلیدواژه‌های یک وب‌سایت مشخص.
  + **دسترسی:** ادمین، مدیر یا مشاهده‌گر سایت.
  + **نیاز به توکن:** بله
  + **پارامتر آدرس:** websiteId

**بررسی جایگاه (/api/websites/:websiteId/keywords/:keywordId/check-rank)**

* **POST /:websiteId/keywords/:keywordId/check-rank**
  + **توضیح:** اجرای دستی فرآیند بررسی جایگاه برای یک کلیدواژه مشخص.
  + **دسترسی:** ادمین یا مدیر سایت.
  + **نیاز به توکن:** بله
  + **پارامتر آدرس:** websiteId, keywordId

**مدیریت کلیدهای API (/api/keys)**

* **POST /**
  + **توضیح:** افزودن یک کلید API جدید از سرویس‌های خارجی.
  + **دسترسی:** فقط ادمین.
  + **نیاز به توکن:** بله
  + **بدنه درخواست:** { "serviceName": "string", "keyValue": "string", "monthlyLimit": integer }
* **GET /**
  + **توضیح:** دریافت لیست تمام کلیدهای API ذخیره شده.
  + **دسترسی:** فقط ادمین.
  + **نیاز به توکن:** بله
* **DELETE /:id**
  + **توضیح:** حذف یک کلید API.
  + **دسترسی:** فقط ادمین.
  + **نیاز به توکن:** بله
  + **پارامتر آدرس:** id (شناسه کلید API)