گزارش فنی پروژه

۱ .معماری کلی سامانه

سامانه یک API RESTful امن است که با استفاده از فریم‌ورک Flask پیاده‌سازی شده و امکان ثبت‌نام، ورود، ذخیره‌سازی و بازیابی امن داده‌های حساس کاربران را فراهم می‌کند. اجزای اصلی سامانه:

- Backend API: پیاده‌سازی شده با Flask و مدیریت نشست کاربران با JWT.
- پایگاه داده :SQLite ذخیره‌سازی اطلاعات کاربران، کلیدهای رمزنگاری، و داده‌های حساس.
- ماژول‌های مستقل برای امنیت :شامل auth.py(برای احراز هویت و ثبت‌نام) و) crypto.pyبرای رمزنگاری/رمزگشایی و مدیریت کلید.(

۲ .الگوریتم‌های رمزنگاری و هش استفاده‌شده

- رمزنگاری داده‌های حساس:
  از کتابخانه‌ی cryptography.fernetبرای رمزنگاری متقارن داده‌ها استفاده شده است. کلید کاربران به‌صورت جداگانه تولید و رمزنگاری می‌شود و سپس در پایگاه داده ذخیره می‌گردد.
- تولید کلید و:KDF
  برای رمزنگاری کلید کاربر، از PBKDF2HMACبا الگوریتم SHA256استفاده شده و کلید نهایی با استفاده از MASTER_KEYو saltمشتق می‌شود.
- هش کردن گذرواژه:
  در فایل auth.py، گذرواژه‌ها با الگوریتم SHA512به‌همراه saltتصادفی و PEPPERاز فایل env.هش می‌شوند. این روش جلوی حملات rainbow table را می‌گیرد.
- توکن‌های احراز هویت:
  از JWTبا کلید JWT_SECRET_KEYبرای ایجاد توکن‌های session استفاده شده که با الگوریتم HMAC امضا می‌شوند.

۳ .ساختار پایگاه داده

پایگاه داده شامل سه جدول اصلی است:

- User:
  - id, username, password_hash, salt
  - اطلاعات مربوط به لاگین و قفل شدن حساب (در صورت ورود ناموفق زیاد)
- EncryptionKey:
  - شامل user_id, key_encrypted, salt
  - کلید رمزنگاری هر کاربر را) که با MASTER_KEY رمز شده (نگه می‌دارد.
- SensitiveData:
  - user_id, data_type, data_encrypted, iv, created_at
  - داده‌های رمزنگاری‌شده کاربر بر اساس نوع داده (مثلاً کارت، یادداشت، ...) ذخیره می‌شوند.

۴ .روش مدیریت کلید

- MASTER_KEY:

کلید اصلی از env.خوانده می‌شود و برای مشتق‌سازی کلید رمزنگاری کلیدهای کاربران استفاده می‌شود.

- کلید هر کاربر :

با Fernet.generate_key()ساخته شده و با کلیدی مشتق‌شده از MASTER_KEYرمز شده و در جدول EncryptionKeyذخیره می‌شود.

- اشتراک‌گذاری کلید:

هر کاربر فقط به کلید خودش دسترسی دارد؛ کلیدها به‌صورت رمزنگاری‌شده نگهداری می‌شوند و هنگام استفاده، به‌صورت موقت رمزگشایی می‌شوند.

۵ .دلایل انتخاب ابزارها و روش‌ها

Flask:

چارچوبی سبک، ساده و مناسب برایAPI های.RESTful

JWT:

برای احراز هویت امن و stateless کاربران.

cryptography.fernet:

پیاده‌سازی امن و ساده از رمزنگاری متقارن، با کنترل کافی بر کلید.

PBKDF2HMAC:

روشی مقاوم در برابر brute-force برای مشتق‌سازی کلید.

SQLite:

ساده و قابل‌حمل برای تست و پیاده‌سازی اولیه.

SHA512 + salt + pepper:

برای هش امن گذرواژه‌ها، مطابق باbest practice های امنیتی.