6. Sensitive information storage

This task has educational value only. Implementing cryptographically secure storage requires skills and knowledge way beyond the scope of this course. DO NOT use results of this task in production applications without consulting with security and cryptography specialists.

Your task is to modify your app previously created in lab 5 to store sensitive/private information of each user. This may be home address, phone number,personal photos or files etc.

The easiest way to store this info is to use the same db that is used to store user credentials. Though you are free to use any kind of storage that you may be familiar with: db, files, buckets etc.

1. Let the user store information that he or anyone else can view through your app. E.g. think of OLX: anyone may get access to someone's phone number. But if their db gets stolen - no phone number may be retrieved (at least that's how it is supposed to work).

   You may achieve this by storing AEAD-encrypted data and key in separate locations. E.g. data is stored in db, while the key rests in a config file with restricted access (possibly also encrypted, read about envelope encryption). For key storage you are free to utilise any available options, although I recommend using KMS provided by any major cloud provider. But again, this task is more about educational value than real security. Thus you may want to utilise simpler options, but include in your report that you are aware of security implications.

2. Write a short report which includes:
   a. How did you implement your storage?
   b. Why did you choose particular storage options/algorithms/libs etc?
   c. What are the possible attack vectors on your system (i.e. how the stored information may be stolen)?

Read up on best practices of cryptographic storage: OWASP crypto storage cheat sheet.

Upload all your results to a public github repository. **This task will be graded based on your code and your report.**