

پروژه ی قفل هوشمند

استاد محترم: سرکار خانم زهرا سادات عصایی معمم

دانشجویان: محمد امیر نوروزی (40016341054298) – کیمیا هاشم پور (40016341054144)

نیم سال دوم – 1402

تعداد صفحات: 4

شبه کد پروژه ی هوش مصنوعی:

طراحی این شبه کد بر اساس طرح ایده ی نو مطرح شده در متن پروژه می باشد و برای خوانایی بهتر کد در فایل ورد قرار گرفته است.

گجتی که بر اساس ویژگی انحصاری (مغز) انسان پالس هایی را به دیتا بیس ارسال و در صورت تایید در باز میشود .

```
class BrainPatternCollector:
```

```
    def init(self):
```

```
        # لیستی برای ذخیره الگوهای مغزی
```

```
        self.brain_patterns = []
```

```
    def collect_pattern(self, pattern):
```

```
        # افزودن الگوی مغزی به لیست
```

```
        self.brain_patterns.append(pattern)
```

```
class Scanner:
```

```
    def init(self, pattern_collector):
```

```
        # متغیری برای نگهداری گردآورنده الگوهای مغزی
```

```
        self.pattern_collector = pattern_collector
```

```
def scan_brain_pattern(self, user):
```

```
    الگوی مغزی از گجت کاربر دریافت می‌شود #
```

```
    brain_pattern = user.gadget.get_brain_pattern()
```

```
    الگوی مغزی در گردآورنده ذخیره می‌شود #
```

```
    self.pattern_collector.collect_pattern(brain_pattern)
```

```
    return brain_pattern
```

```
class Gadget:
```

```
    def init(self, user_id):
```

```
        شناسه کاربر و لیستی برای ذخیره الگوهای مغزی #
```

```
        self.user_id = user_id
```

```
        self.brain_patterns = []
```

```
        ویژگی اثر انگشت #
```

```
        self.fingerprint_enabled = False
```

```
    def record_brain_pattern(self, pattern):
```

```
        افزودن الگوی مغزی به لیست #
```

```
        self.brain_patterns.append(pattern)
```

```
    def get_brain_pattern(self):
```

```
        آخرین الگوی مغزی را باز می‌گرداند #
```

```
        return self.brain_patterns[-1]
```

```
    def enable_fingerprint(self):
```

```
        فعال کردن ویژگی اثر انگشت #
```

```
        self.fingerprint_enabled = True
```

```
        print("Fingerprint authentication is now enabled.")
```

```
    def authenticate_fingerprint(self):
```

```
        اعتبارسنجی اثر انگشت #
```

```
        if self.fingerprint_enabled:
```

```
            print("Fingerprint authentication successful.")
```

```
            return True
```

```
        else:
```

```
            print("Fingerprint authentication is not enabled.")
```

```
            return False
```

```

class User:
    def init(self, user_id, gadget):

        self.user_id = user_id
        self.gadget = gadget

class Environment:
    def init(self):

        self.registered_users = {}

        self.pattern_collector = BrainPatternCollector()

        self.scanner = Scanner(self.pattern_collector)

    def register_user(self, user):

        self.registered_users[user.user_id] = user

    def access_control(self, user_id):

        if user_id in self.registered_users:
            user = self.registered_users[user_id]

            scanned_pattern = self.scanner.scan_brain_pattern(user)

            print(f"Access granted for user {user.user_id}.")
        else:
            print("User not registered.")

smart_lock_environment = Environment()

user1_gadget = Gadget(user_id="001")

```

شناسه کاربر و گجت مرتبط با او #
 دیکشنری برای ثبت نام کاربران #
 گردآورنده الگوهای مغزی #
 اسکنر برای اسکن الگوهای مغزی #
 ثبت نام کاربر در محیط #
 بررسی دسترسی کاربر #
 اسکن الگوی مغزی کاربر #
 عملیات دسترسی #
 مثال استفاده: #
 ایجاد گجت و فعال کردن اثر انگشت #

```
user1_gadget.enable_fingerprint()
```

ایجاد کاربر و ثبت نام در محیط

```
user1 = User(user_id="001", gadget=user1_gadget)  
smart_lock_environment.register_user(user1)
```

دسترسی کاربر

```
smart_lock_environment.access_control("001")
```

با تشکر از حسن توجه شما