

CS3053 - Computer Security

**Continuous assessment - Protection of information
based on sensitivity and privilege levels**

Name : Weerasekara W.M.T.B.

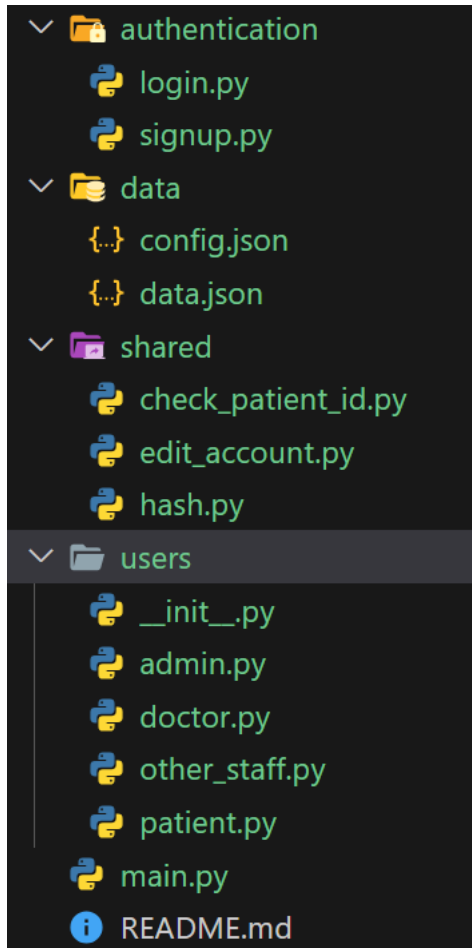
Index : 200698X

1. The program code to carry out the above steps

GitHub : https://github.com/ThejanB/Medical_Processing_System-Security

The medical data processing program was programmed using Python language

File Structure us as following:



Hashing encoding and decoding functions :

```
1  import hashlib
2  import base64
3
4  ### hash_password function to hash password      You, 5 minutes ago
5  def hash_password(user_password):
6      hashed_password = hashlib.md5()
7      hashed_password.update(user_password.encode("utf-8"))
8      return hashed_password.hexdigest()
9
10 ### encode function to encode string to base64
11 def encode(encode_text):
12     base64_bytes = base64.b64encode(encode_text.encode('ascii'))
13     base64_message = base64_bytes.decode('ascii')
14     return base64_message
15
16
17 ### decode function to decode base64 encoded string
18 def decode(decode_text):
19     message_bytes = base64.b64decode(decode_text.encode('ascii'))
20     message = message_bytes.decode('ascii')
21     return message
```

User types are

1. Admin
2. Doctor
3. Other Staff
4. Patient

Created a Class for each to handle its own tasks

Admin Class

```
You, 11 minutes ago | 1 author (You)
1  from shared.hash import *
2  import json
3
4  class Admin:
5      def run(self):
6
7          while True:
8              role_number = input("Press 1 to edit doctor code\nPres
9
10             if role_number == '1':
11                 self.change_doctor_code()
12
13             elif role_number == '2':
14                 self.change_other_staff_code()
15
16             elif role_number == '3':
17                 self.create_admin()
18
19             elif role_number == '-1':
20                 print("Admin logout!\n")
21                 break
22             else:
23                 print("Invalid input")
24
25         @staticmethod
26         def create_admin(): ...
27
28         @staticmethod
29         def change_doctor_code(): ...
30
31         @staticmethod
32         def change_other_staff_code(): ...
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Doctor Class

```
...
1  import json
2  from datetime import datetime
3
4  class Doctor:
5
6      @staticmethod
7      def add_sickness_details(patient_id):
8
9          sickness_details = input("Enter sickness details: \n")
10
11          try:
12              with open("data/data.json", 'r') as json_data_file:
13                  data = json.load(json_data_file)
14
15              # Append sickness details
16              data['sickness_details'].append({
17                  'id': patient_id,
18                  'sickness': sickness_details,
19                  'date': str(datetime.now())[19],
20              })
21
22              with open("data/data.json", 'w') as outfile:
23                  json.dump(data, outfile)
24
25              print("Sickness details added")
26
27          except FileNotFoundError:
28              print("Data file not found. Please ensure the file exists.")
29          except json.JSONDecodeError:
30              print("Error decoding JSON data in the file.")
31
32      @staticmethod
33      def add_drug_prescription(patient_id): ...
34
35      @staticmethod
36      def read_sickness_details(patient_id): ...
37
38      @staticmethod
39      def read_drug_prescription(patient_id): ...
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Other Staff Class

```
1 import json
2 from shared.hash import *
3 from shared.edit_account import *
4
5 class Other_staff:
6
7     def run(self, current_user_id):
8         print("Press 1 to create patient account\nPress 2 to edit personal account")
9         while True:
10             option = input()
11             if option == '1':
12                 Other_staff().create_patient_account()
13                 print("Press next number: ")
14             elif option == '2':
15                 edit_account(current_user_id)
16                 print("Press next number: ")
17             elif option == '3':
18                 view_account(current_user_id)
19                 print("Press next number: ")
20             elif option == '4':
21                 renew_password(current_user_id)
22                 print("Press next number: ")
23             elif option == '-1':
24                 print("Thank you other_staff")
25                 break
26             else:
27                 print("Invalid input. Try again")
28
29     @staticmethod
30     def create_patient_account(): ...
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
```

Patient Class

```
1 import json
2
3
4 class Patient:
5
6     @staticmethod
7     def read_sickness_details(patient_id):
8
9         # read sickness details from data file
10         with open("data/data.json", 'r') as json_data_file:
11             data = json.load(json_data_file)
12             details = data['sickness_details']
13
14             for detail in details:
15                 if detail['id'] == patient_id:
16                     print(detail['date'], '-', detail['sickness'])
17
18     @staticmethod
19     def read_drug_prescription(patient_id):
20
21         # read drug prescription from data file
22         with open("data/data.json", 'r') as json_data_file:
23             data = json.load(json_data_file)
24             details = data['drug_presc']
25
26             for detail in details:
27                 if detail['id'] == patient_id:
28                     print(detail['date'], '-', detail['drug_presc'])
29
```

Sign in functions

```
1 import json
2 from shared.hash import *
3
4
5 class SignUp:
6     roles = {'1': "doctor", '2': "other_staff"}
7
8     def create_new_user(self):
9
10         name = input('Username:')
11         password = input('Enter a password including an upper-case letter, lower-case letter, a digit and length not less than 5: \n')
12         while True:
13             if not (any(x.isupper() for x in password) and
14                     any(x.islower() for x in password) and
15                     any(x.isdigit() for x in password) and
16                     len(password) >= 5):
17                 password = input("Password is too weak. Please re-enter a new password: ")
18             else:
19                 break
20         safe_password = hash_password(password)
21
22         # read config file for doctor and other_staff codes
23         with open("data/data/config.json", 'r') as json_data_file:
24             data = json.load(json_data_file)
25             doc_code = data['codes'].get('doctor')
26             rec_code = data['codes'].get('other_staff')
27
28         # hotel staff accounts can only be created by using the relevant code for security
29         while True:
30
31             role_number = input('Press 1 for "doctor account" or 2 for "other_staff account": ')
32
33             if role_number == '1':
34                 code = hash_password(input("Please enter the doctor code to create account: "))
35                 if code == doc_code:
36                     user_type = self.roles.get('1')
37                     privilege_level = '2'
38                     break
39
40             elif role_number == '2':
41                 code = hash_password(input("Please enter the other_staff code to create account: "))
42                 if code == rec_code:
43                     user_type = self.roles.get('2')
44                     privilege_level = '3'
45                     break
46                 else:
47                     print("Invalid code")
48             else:
49                 print("Invalid input. Try again")
50
51         # read and write user to config file
52         with open("data/config.json", 'r') as json_data_file:
53             data = json.load(json_data_file)
54             user_id = len(data['users'])+1
55             data['users'].append({
56                 'id': user_id,
57                 'name': name,
58                 'password': safe_password,
59                 'user_type': user_type,
60                 'privilege_level': privilege_level
61             })
62         with open("data/config.json", 'w') as outfile:
63             json.dump(data, outfile)
64
65         print("Account created successfully")
66
67         # fill account details
68         print("Fill in personal account details")
69         acc_name = input('Your name: ')
70         age = input('Age: ')
71         nic = input('NIC number: ')
72         tel = input('Telephone number: ')
73
74         # write account details to data file
75         with open("data/data.json", 'r') as json_data_file:
76             data = json.load(json_data_file)
77             data['personal_details'].append({
```

```

76     # write account details to data file
77     with open("data/data.json", 'r') as json_data_file:
78         data = json.load(json_data_file)
79         data['personal_details'].append({
80             'id': user_id,
81             'name': acc_name,
82             'age': age,
83             'nic': encode(nic),
84             'tel': encode(tel)
85         })
86     with open("data/data.json", 'w') as outfile:
87         json.dump(data, outfile)
88
89     print("Account completed")
90

```

Login function

```

You, 15 minutes ago | 1 author (You)
1  import json
2  from shared.hash import hash_password
3
You, 15 minutes ago | 1 author (You)
4  class Login:
5
6      @staticmethod
7      def auth_user():
8          result = 'Login failed'
9
10         name = input("Enter username: ")
11         hashed_password = hash_password(input("Enter password: "))
12
13         with open("data/config.json", 'r') as json_data_file:
14             data = json.load(json_data_file)
15             for user in data['users']:
16                 if name == user['name'] and hashed_password == user['password']:
17                     current_user = user
18                     result = "Login successful"
19             print(result)
20             if result == "Login successful":
21                 return current_user
22             else:
23                 return False
24
25
26

```

2. A description of how I decided the access to data records based on sensitivity of data

The different data records have different levels of sensitivity and reading and writing rights should only be given to certain user roles.

I took four types of user roles with different privilege levels

1. **Admin** - privilege level 1
 - set up secret codes for Doctor and Other staff accounts, this will be saved as hashed codes in the config file.
 - Add new admin functionality
2. **Doctor** - privilege level 2
 - Can create patient accounts. Before being allowed to create patient accounts, the doctor user will be prompted to enter the doctor code.
 - Doctor can add and view patients details and illness data of patients
 - Can view and edit their personal details.
3. **Other staff** - privilege level 3
 - Can create patient accounts. Before being allowed to create patient accounts, the other staff user will be prompted to enter the other_staff code.
 - Other staff can view patients details and illness data of patients. They cannot add patients details or illness data.
 - Can view and edit their personal details.
4. **Patient** - privilege level 4
 - Can only read their own sickness details
 - Can view and edit their personal details.

Security Measures

This command line, role-based access control system ensures that users only have access to data and operations that are appropriate for their role and privilege level.

All passwords are hashed using MD5 before being written to the config file, enhancing the security of stored passwords.

Sensitive data, such as NIC numbers and telephone numbers, are encoded in base64 before being written to the data file, providing an additional layer of data security.

3. Configuration and data files

Config file -> save user authentication data, doctor_code and other_staff_code

```
1      {
2          "users": [
3              {
4                  "id": 1,
5                  "name": "200698X",
6                  "password": "daadcead1a5bf23ef97f53f6bcb3bc43",
7                  "user_type": "admin",
8                  "privilege_level": "1"
9              },
10             {
11                 "id": 2,
12                 "name": "doctor1",
13                 "password": "daadcead1a5bf23ef97f53f6bcb3bc43",
14                 "user_type": "doctor",
15                 "privilege_level": "2"
16             },
17             {
18                 "id": 3,
19                 "name": "staff1",
20                 "password": "daadcead1a5bf23ef97f53f6bcb3bc43",
21                 "user_type": "other_staff",
22                 "privilege_level": "3"
23             },
24             {
25                 "id": 4,
26                 "name": "patient1",
27                 "password": "daadcead1a5bf23ef97f53f6bcb3bc43",
28                 "user_type": "patient",
29                 "privilege_level": "4"
30             }
31         ],
32         "codes": {
33             "doctor": "563e2741736500e8d0eac3edb54f67e1",
34             "other_staff": "a722c63db8ec8625af6cf71cb8c2d939"
35         }
36     }
37
```

Data File -> Save patients data as lists

```

100, 10 minutes ago | 1 commit (new)
1  {
2    "personal_details": [
3      {
4        "id": 1,
5        "name": "Name 1",
6        "age": "23",
7        "nic": "OTc2NDUzMzc5dg==",
8        "tel": "MDcxMjIyNjQ2OQ=="
9      },
10     {
11       "id": 2,
12       "name": "Name 2",
13       "age": "22",
14       "nic": "OTg2NDUzMzc5dg==",
15       "tel": "MDC4MjQ2NTk2Mw=="
16     },
17     {
18       "id": 3,
19       "name": "Name 3",
20       "age": "22",
21       "nic": "OTc2NTMxMTk2dg==",
22       "tel": "MDC5MjM1MTg4Mw=="
23     },
24     { ...
30     },
31     { ...
37     },
38     { ...
44     }
45   ],
46   "sickness_details": [ ...
72   ],
73
74   "drug_presc": [ ...
100   ]
101 }

```

--- All the Codes are available in above GitHub Link ---