ACKNOWLEDGEMENT

I would like to thank the most benevolent God for his blessing to complete the task at hand successfully.

I am much obliged to Rev. Fr. Johny Kanjirathinkal the Principal. Devagiri CMI Public School for providing facilities to complete this work.

I am greatly indebted to Ms. Vidya's Lectures in computer science for their valuable guidance, inspiring supervision and constant encouragement in carrying out the task in hand.

I would like to extend my gratitude to all faculty members of Devagiri CMI Public School.

I would like to express my gratitude to friends and my parents for their extended incessant support and timely help.

I express my sincere thanks to all those who have contributed directly or indirectly to make this project a success.

INDEX

INTRODUCTION

Due to the fluctuating number of COVID-19 patients, it has become difficult and tiring to keep and file records manually in hospitals.

COVID-19 Patient Analysis System Software(C-PASS) is a Software developed for analysing the data set of COVID-19 patients and displaying the records of patients requiring special attention, classifying and displaying records of patients on the basis of severity of symptoms, displaying records of admitted, discharged and deceased patients and calculating percentage of patients who have recovered and deceased.

The scope of this program is to-

- Provide a suitable database to manage records in the hospital with regard to the Covid-19 patients.
- Provide a user friendly interface to do so.

OVERVIEW

Key features include-

- Python was developed by Guido Van Rossum in the late 80’s and early 90’s at the National Research Institute for Mathematics and Computer Science in the Netherlands.
- Python is a high-level, interpreted, interactive and object-oriented scripting language.
- Python is designed to be highly readable.
- It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

Other features include-

- Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read – Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain – Python's source code is fairly easy-to-maintain.
- A broad standard library – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases – Python provides interfaces to all major commercial databases.
- GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable – Python provides a better structure and support for large programs than shell scripting.

SYNOPSIS

- To add/display records of patients admitted.
- To display the records of patients who require special attention (eg. pregnant women, chronic diseases, etc.)
- To classify and to display the records of patients showing mild, moderate and severe symptoms.
- To transfer/display the records of patients discharged to another file.
- To transfer/display the records of deceased patients to another file.
- To calculate and display percentage of patients who have recovered and deceased

FUNCTIONS USED

- file_exists(file) - used to check if the file exists in the system.
- create_main_csv() - used to create the main csv file containing the info on all the patients.
- create_discharged_csv() - used to create sub csv file to show discharged patients.
- create_deceased_csv() - used to create the sub csv file to show deceased patients.
- append_csv() - used to add records to the main csv file.
- remove_scv() - used to remove records based on their Patient ID's from the main csv file.
- edit_csv() - used to edit records based in their Patient ID's.
- view_csv() - used to view records based on their Patient ID's.
- display_main() - displays all patient records.
- display_discharged() - displays records of discharged patients.
- display_deceased() - displays records of deceased patients.
- discharged() - appends all discharged patients in a new csv file.
- deceased() - appends all deceased patients in a new csv file.
- main() - used to combine all the above functions into a menu driven program for easy use.

MODULES USED

- csv module - The csv module implements classes to read and write tabular data in CSV format.
- time module - This module provides various time-related functions.

SOURCE CODE

```
# imports
import csv
import time
main_file_name = "C-Pass.csv"
file_discharged = "C-Pass Discharged.csv"
file_deceased = "C-Pass Deceased.csv"


def file_exists(file):
    """
    the function checks for the presence of the
    csv file to prevent the code from crashing
    if the user tries to append/edit/remove from csv
    """
    try:
        with open(file, "r"):
            return True
    except FileNotFoundError:
        return False


def create_main_csv():
    """
    the function allows the user to create
    the main csv file if it's not created already
    """
    print("Creating file...")
    with open(main_file_name, "w", newline="") as csv_file:
        csv_writer = csv.writer(csv_file)
        csv_writer.writerow(["Patient ID", "Patient Name", "Date of Birth", "Date of Admission",
"Gender", "Symptom Severity", "Status"])
        print("File created!")


def create_discharged_csv():
    """
    the function allows the user to create
    the csv file of the discharged patients
     if it's not created already
    """
    print("Creating file...")
    with open(file_discharged, "w", newline="") as csv_file:
        csv_writer = csv.writer(csv_file)
        csv_writer.writerow(["Patient ID", "Patient Name", "Date of Birth", "Date of Admission",
"Gender", "Symptom Severity", "Status"])
        print("File created!")


def create_deceased_csv():
    """
    the function allows the user to create
    the csv file od deceased patients
    if it's not created already
    """
    print("Creating file...")
    with open(file_deceased, "w", newline="") as csv_file:
        csv_writer = csv.writer(csv_file)
        csv_writer.writerow(["Patient ID", "Patient Name", "Date of Birth", "Date of Admission",
"Gender", "Symptom Severity", "Status"])
        print("File created!")


def append_csv():
    """
    this function allows the user to add new entries
```

```
with the following column headings
Patient ID, Patient Name, Date of Birth, Date of Admission, Gender, Symptom Severity
"""
while file_exists(main_file_name):
    item = []
    patient_id = input("Enter Patient ID: ")
    if patient_id.isdigit():
        item.append(patient_id)
    else:
        print("Invalid input!")
    patient_name = input("Enter Patient Name: ")
    item.append(patient_name)
    dob = input("Enter Date of Birth (in YYYY-MM-DD format): ")
    item.append(dob)
    doa = input("Enter Date of Admission (in YYYY-MM-DD format): ")
    item.append(doa)
    gender = input("Enter Gender (M/F/O): ").upper()
    if gender in ("M", "F", "O"):
        item.append(gender)
    else:
        print("Invalid input!")
    symptom_severity = input("Symptom Severity (Mild, Moderate, Severe): ").capitalize()
    if symptom_severity in ('Mild', 'Moderate', 'Severe'):
        item.append(symptom_severity)
    else:
        print("Invalid input!")
    status = input("Enter Status of Patient (Admitted, Discharged, Deceased): ").capitalize()
    if status in ("Admitted", "Discharged", "Deceased"):
        item.append(status)
    else:
        print("Invalid input!")
    print(item)
    confirm = input("Are you sure you want to add this item? (y/n): ")
    if confirm == "y" or confirm == "Y":
        with open(main_file_name, "a", newline="") as csv_file:
            csv_writer = csv.writer(csv_file)
```

```
                csv_writer.writerow(item)
                print("Item added!")
        elif confirm == "n" or confirm == "N":
            print("Item not added!")
        cont = input("Continue appending?(Y/N): ")
        if cont == "N" or cont == "n":
            break
        elif cont == "Y" or cont == "y":
            continue
        else:
            print("Invalid input!")
    else:
        print("File does not exist, create a new file before continuing.")


def remove_csv():
    """
    used to remove records with respect to Patient ID
    """
    if file_exists(main_file_name):
        with open(main_file_name, "r", newline="") as csv_file:
            csv_writer = csv.reader(csv_file)
            for row in csv_writer:
                print(row[0], row[1])
            p = int(input("Enter Patient ID of item to remove: "))
            l = []
            csv_file.seek(0)
            csv_file.readline()
            for row in csv_writer:
                if p != int(row[0]):
                    l.append(row)
        with open(main_file_name, "w", newline="") as csv_file:
            csv_writer = csv.writer(csv_file)
            csv_writer.writerow(["Patient ID", "Patient Name", "Date of Birth", "Date of Admission", "Gender", "Symptom Severity", "Status"])
            csv_writer.writerows(l)
            print("Item removed!")
    else:
        print("File not found, create a new file to continue\n")



def edit_csv():
    """
    used to edit records i the csv wile with respect to Patient ID
    :return:
    """
    if file_exists(main_file_name):
        s = view_csv(con=1)
        with open(main_file_name, "r", newline="") as csv_file:
            csv_writer = csv.reader(csv_file)
            l = []
            csv_file.seek(0)
            for row in csv_writer:
                if row[0].isdigit():
                    if s == int(row[0]):
                        item_to_edit = int(input("Enter Row to edit\
(1 - Patient Name, 2 - Date of Birth, 3 - Date of Admission, 4 - Gender, 5 - Symptom
Severity, 6 - Status):"))
                        x = []
                        if item_to_edit == 1:
                            new_name = input("Enter new Patient Name: ")
                            x = [row[0], new_name]
                            for i in range(2, len(row)):
                                x.append(row[i])
                            l.append(x)
                        if item_to_edit == 2:
                            new_dob = input("Enter new Date of Birth (in YYYY-MM-DD format): ")
                            for i in range(len(row)-6):
                                x.append(row[i])
                            x.append(new_dob)
                            for i in range(3, len(row)):
                                x.append(row[i])
                            l.append(x)
                        if item_to_edit == 3:
                            new_doa = input("Enter Date of Admission (in YYYY-MM-DD format): ")
                            for i in range(len(row)-5):
                                x.append(row[i])
                            x.append(new_doa)
                            for i in range(2, len(row)):
                                x.append(row[i])
                            l.append(x)
                        if item_to_edit == 4:
                            new_gender = input("Enter Gender: ")
                            for i in range(len(row)-4):
                                x.append(row[i])
                            x.append(new_gender)
                            l.append(x)
                        if item_to_edit == 5:
                            new_severity = input("Enter change in severity of symptoms: ")
                            for i in range(len(row)-3):
                                x.append(row[i])
                            x.append(new_severity)
                            l.append(x)
                        if item_to_edit == 5:
                            new_severity = input("Enter change in severity of symptoms: ")
                            for i in range(len(row)-2):
                                x.append(row[i])
                            x.append(new_severity)
                            l.append(x)
                        if item_to_edit == 6:
                            new_status = input("Enter change in status: ")
                            for i in range(len(row)-1):
                                x.append(row[i])
                            x.append(new_status)
                            l.append(x)
                    else:
                        l.append(row)
                else:
```

```
                    l.append(row)
        with open(main_file_name, "w", newline="") as csv_file:
            csv_writer = csv.writer(csv_file)
            csv_writer.writerows(l)
            print("Item edited!")
    else:
        print("File not found, create a new file to continue\n")


def view_csv(con=0):
    """
    prints the information of an item
    with respect to the Patient ID
    """
    print()
    if file_exists(main_file_name):
        with open(main_file_name, "r", newline="") as csv_file:
            csv_writer = csv.reader(csv_file)
            for row in csv_writer:
                print(row[0], row[1])
            patient_id = int(input("Enter Patient Id of item to View contents of: "))
            csv_file.seek(0)
            csv_file.readline()
            top = {"Patient ID": "", "Patient Name": "", "Date of Birth": "", "Date of Admission":
"", "Gender": "", "Symptom Severity": ""}
            l = []
            for row in csv_writer:
                if int(row[0]) == patient_id:
                    for i in row:
                        l.append(i)
            count = 0
            for i in top:
                top[i] = l[count]
                count += 1
            for i in top:
                print(i, "-", top[i])
            print()
            if con == 1:
                return patient_id
    else:
        print("File not found, create a new file to continue\n")


def display_main():
    """
    displays the entire content of all patients
    """
    if file_exists(main_file_name):
        with open(main_file_name, "r", newline="") as csv_file:
            csv_writer = csv.reader(csv_file)
            length = 0
            for _ in csv_writer:
                length += 1
            csv_file.seek(0)
            for row in csv_writer:
                for J in row:
                    print(J.ljust(20), end=" ")
                print()
    else:
        print("File not found, create a new file to continue\n")


def display_discharged():
    """
    displays the entire contents of the CSV of discharged patients
    """
    if file_exists(file_discharged):
        with open(file_discharged, "r", newline="") as csv_file:
            csv_writer = csv.reader(csv_file)
            length = 0
            for _ in csv_writer:
                length += 1
            csv_file.seek(0)
            for row in csv_writer:
                for J in row:
                    print(J.ljust(20), end=" ")
                print()
    else:
        print("File not found, create a new file to continue\n")


def display_deceased():
    """
    displays the entire contents of the CSV of deceased patients
    """
    if file_exists(file_deceased):
        with open(file_deceased, "r", newline="") as csv_file:
            csv_writer = csv.reader(csv_file)
            length = 0
            for _ in csv_writer:
                length += 1
            csv_file.seek(0)
            for row in csv_writer:
                for J in row:
                    print(J.ljust(20), end=" ")
                print()
    else:
        print("File not found, create a new file to continue\n")


def discharged():
    """
    appends records of all discharged patients to new file
    """
    if file_exists(main_file_name):
        csv_main_file = open(main_file_name, "r")
        csv_reader = csv.reader(csv_main_file)
        if file_exists(file_discharged):
            with open(file_discharged, "w", newline="") as csv_file:
                l = []
                for _ in csv_reader:
                    if _[-1] == "Discharged":
                        l.append(_)
                csv_writer = csv.writer(csv_file)
                csv_writer.writerow(["Patient ID", "Patient Name", "Date of Birth", "Date of
Admission", "Gender", "Symptom Severity", "Status"])
                csv_writer.writerows(l)
        else:
            print("File not found, create a new file to continue\n")
    else:
        print("File not found, create a new file to continue\n")



def deceased():
    """
    appends records of all deceased patients to new file
     """
    if file_exists(main_file_name):
        csv_main_file = open(main_file_name, "r")
        csv_reader = csv.reader(csv_main_file)
        if file_exists(file_deceased):
            with open(file_deceased, "w", newline="") as csv_file:
                l = []
                for _ in csv_reader:
                    if _[-1] == "Deceased":
                        l.append(_)
                csv_writer = csv.writer(csv_file)
                csv_writer.writerow(["Patient ID", "Patient Name", "Date of Birth", "Date of
Admission", "Gender", "Symptom Severity", "Status"])
                csv_writer.writerows(l)
        else:
            print("File not found, create a new file to continue\n")
    else:
        print("File not found, create a new file to continue\n")
```

```
def main():
    """
    main function that
    joins all the other functions into
    a menu-driven program
    """
    print("Welcome to C-Pass!”)
    while True:
        print("""1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
4. Edit an item in main file
5. View info of item with respect to Patient ID
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit""")
        if not file_exists(main_file_name):
            print("File not found, create a new file before continuing.")
        else:
            print("File'", main_file_name, " Found")
        if not file_exists(file_deceased):
            print("File not found, create a new file before continuing.")
        else:
            print("File'", file_deceased, " Found")
            if not file_exists(file_discharged):
                print("File not found, create a new file before continuing.")
            else:
                print("File'", file_discharged, " Found")
        choice = input("Enter your choice: ")
        if choice == "1":
            create = input("Enter which file to create (1 - C-Pass.csv. 2 - C-Pass
Discharged.csv, 3 - C-Pass Deceased.csv): ")
            if create == "1":
                create_main_csv()
            if create == "2":
                create_discharged_csv()
            if create == "3":
                create_deceased_csv()
        elif choice == "2":
            append_csv()
        elif choice == "3":
            remove_csv()
        elif choice == "4":
            edit_csv()
        elif choice == "5":
            view_csv()
        elif choice == "6":
            display_main()
        elif choice == "7":
            discharged()
            display_discharged()
        elif choice == "8":
            deceased()
            display_deceased()
        elif choice == "9":
            print("The program will exit in 5 seconds")
            time.sleep(5)
            print("Exiting", end="")
            time.sleep(4)
            print(".", end="")
            time.sleep(3)
            print(".", end="")
            time.sleep(2)
            print(".", end="")
            time.sleep(1)
            break

if __name__ == "__main__":
    main()
```

OUTPUT

```
1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
4. Edit an item in main file
5. View info of item with respect to Patient ID
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit
File' C-Pass.csv  Found
File' C-Pass Deceased.csv  Found
File' C-Pass Discharged.csv  Found
Enter your choice: 1
Enter which file to create (1 - C-Pass.csv. 2 - C-Pass Discharged.csv, 3 - C-Pass
Deceased.csv): 1
Creating file...
File created!
1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
4. Edit an item in main file
5. View info of item with respect to Patient ID
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit
File' C-Pass.csv  Found
File' C-Pass Deceased.csv  Found
File' C-Pass Discharged.csv  Found
Enter your choice: 1
Enter which file to create (1 - C-Pass.csv. 2 - C-Pass Discharged.csv, 3 - C-Pass
Deceased.csv): 2
Creating file...
File created!
1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
4. Edit an item in main file
5. View info of item with respect to Patient ID
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit
File' C-Pass.csv  Found
File' C-Pass Deceased.csv  Found
File' C-Pass Discharged.csv  Found
Enter your choice: 1
Enter which file to create (1 - C-Pass.csv. 2 - C-Pass Discharged.csv, 3 - C-Pass
Deceased.csv): 3
Creating file...
File created!
1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
4. Edit an item in main file
5. View info of item with respect to Patient ID
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit
File' C-Pass.csv  Found
File' C-Pass Deceased.csv  Found
File' C-Pass Discharged.csv  Found
Enter your choice: 2
Enter Patient ID: 101
Enter Patient Name: Carolin James
Enter Date of Birth (in YYYY-MM-DD format): 1999-09-09
Enter Date of Admission (in YYYY-MM-DD format): 2022-02-02
Enter Gender (M/F/O): F
Symptom Severity (Mild, Moderate, Severe): Mild
Enter Status of Patient (Admitted, Discharged, Deceased): Admitted
['101', 'Carolin James', '1999-09-09', '2022-02-02', 'F', 'Mild', 'Admitted']
Are you sure you want to add this item? (y/n): y
Item added!
Continue appending?(Y/N): y
Enter Patient ID: 102
Enter Patient Name: Derry Sanders
Enter Date of Birth (in YYYY-MM-DD format): 1989-08-09
Enter Date of Admission (in YYYY-MM-DD format): 2021-09-08
Enter Gender (M/F/O): m
Symptom Severity (Mild, Moderate, Severe): Moderate
Enter Status of Patient (Admitted, Discharged, Deceased): Discharged
['102', 'Derry Sanders', '1989-08-09', '2021-09-08', 'M', 'Moderate', 'Discharged']
Are you sure you want to add this item? (y/n): y
Item added!
Continue appending?(Y/N): y
Enter Patient ID: 103
Enter Patient Name: Gary Shimmers
Enter Date of Birth (in YYYY-MM-DD format): 2001-01-02
Enter Date of Admission (in YYYY-MM-DD format): 2022-04-03
Enter Gender (M/F/O): m
Symptom Severity (Mild, Moderate, Severe): Severe
Enter Status of Patient (Admitted, Discharged, Deceased): Deceased
['103', 'Gary Shimmers', '2001-01-02', '2022-04-03', 'M', 'Severe', 'Deceased']
Are you sure you want to add this item? (y/n): y
Item added!
Continue appending?(Y/N): n
1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
4. Edit an item in main file
5. View info of item with respect to Patient ID
```

```
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit
File' C-Pass.csv  Found
File' C-Pass Deceased.csv  Found
File' C-Pass Discharged.csv  Found
Enter your choice: 6
Patient ID          Patient Name        Date of Birth       Date of Admission    Gender
Symptom Severity     Status
101                 Carolin James       1999-09-09          2022-02-02          F                   Mild
Admitted
102                 Derry Sanders       1989-08-09          2021-09-08          M
Moderate            Discharged
103                 Gary Shimmers        2001-01-02          2022-04-03          M
Severe              Deceased
1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
4. Edit an item in main file
5. View info of item with respect to Patient ID
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit
File' C-Pass.csv  Found
File' C-Pass Deceased.csv  Found
File' C-Pass Discharged.csv  Found
Enter your choice: 7
Patient ID          Patient Name        Date of Birth       Date of Admission    Gender
Symptom Severity     Status
102                 Derry Sanders       1989-08-09          2021-09-08          M
Moderate            Discharged
1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
```

```
4. Edit an item in main file
5. View info of item with respect to Patient ID
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit
File' C-Pass.csv  Found
File' C-Pass Deceased.csv  Found
File' C-Pass Discharged.csv  Found
Enter your choice: 8
Patient ID          Patient Name        Date of Birth        Date of Admission    Gender
Symptom Severity     Status
103                 Gary Shimmers       2001-01-02           2022-04-03           M
Severe              Deceased
1. Create a new file
2. Append to existing file
3. Remove an item from file with respect to Patient ID
4. Edit an item in main file
5. View info of item with respect to Patient ID
6. Display the entire content of all patients
7. Show discharged patients in a new file
8. Show deceased patients in a new file
9. Exit
File' C-Pass.csv  Found
File' C-Pass Deceased.csv  Found
File' C-Pass Discharged.csv  Found
Enter your choice: 9
The program will exit in 5 seconds
Exiting...
```

Note:

Code Tables shown irregular due to page size constraints.

CONCLUSION

- The software C-PASS was successfully designed, implemented, tested and deployed.
- The software makes it easier to keep records of covid patients in hospitals.
- It can successfully transfer records to another file when necessary.
- It is able to efficiently store and retrieve data from the disk files when requested

BIBLIOGRAPHY

- Computer Science with Python for Class XI - Preethi Arora
- Computer Science with Python for Class XII - Preethi Arora
- www.ww3school.com
- www.realpython.com
- www.docsforpython.org