General notes:

- Each member MUST work on at least one of the required functions besides file processing
- كل فرد من أفراد المجموعه لازم يشتغل في فانكشن واحده من المشروع على الأقل بجانب شغل ال files، يعني محدش يقول انا اشتغلت على ال files بس 😀
- Evaluation:

The evaluation will be mainly based on the students' ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used these data structures and why they are better than other data structures in any given case. **You MUST use at least 2 different data structures.**

❖ **Regarding ideas that uses files:**
You should read data once at the beginning of your run then do your operations and accessing datastructures then save in files at the end of your program *this will consume time and lines of code* ;)

TA: Aya Nasser

aya.naser@cis.asu.edu.eg

Title	**Events' Scheduler**
Description	This is an application to schedule the events for multiple users. (each user has an account to deal with his events). ● Each event must have: 1. A name 2. A start date 3. An end date 4. A place 5. Event's start time 6. Reminder time … etc 7. Done (Yes/No) ● The user must be able to: 1. log in (sign up if he/she hasn't an account). 2. Add, 3. Delete an event 4. Update an event. 5. Display all events ● The application should refuse to add an event that intersects with another event. ● The user can display his events sorted by start date and/or reminder time. ● When an event is done it should o disappear from the upcoming events.

	o be added in another data structure holding only the done events. **Store your data in files, no need for database**
Minimum requirements	● You should use at least two appropriate Data structures for each requirement (**selection of DataStructure will be graded**). ● Clean and Efficient Code is a must. ● You can Use the built in DataStructures Libraries (eg.STLs in C++). ● You must fulfill **all** the above requirements.
Bonus Opportunities	● Any extra modules (**Non-Trivial**) designed to give more functionality to your application. ● Good GUI.
Development Tool	● C++ ● Visual Studio

TA: Amira Samir

amira_samir@cis.asu.edu.eg

Title	**Blood Bank Management System**
Description	A system for helping patients especially at emergent cases. Blood donation has a great importance in helping to save other people's lives. Implement a C++ program that allows the user to do the following requirements.
Minimum Requirements	● A user could be donor or recipient: 1. Donor 1. Register with her data a. Validate the donor's request according to her data as a safe donor must be: i. Age from 17 to 60 years old ii. Last donation date must be at least 3 months before her request iii. She mustn't suffer from any disease (blood pressure disorders, thyroid disease, diabetes, cancer, heart disorders, hepatitis) iv. Check that she hasn't any other problem including other diseases or medicine b. If this donor is safe then add her data. i. ID ii. Name iii. Mail iv. Password v. Age vi. Gender vii. Blood type viii. Date of the latest donation 2. Login

3. Donation request
4. Update her data
5. Delete account

2. Recipient
   1. Register with her data
      a. ID
      b. Name
      c. Mail
      d. Password
      e. Age
      f. Gender
      g. Blood type
      h. Hospital
      i. Doctor of the case
   2. Login
   3. Update her data
   4. Delete account
   5. Search for the availability of blood type
   6. Display all blood data
      a. Type
      b. Quantity
      c. Received and Expiry dates
   7. Request the blood type and quantity she wants and confirm about the hospital (place where the patient is available) (Keep track of the available quantity, if the desired quantity is less than the available quantity the customer cannot complete the request).

- You should use the **appropriate Data structures** for each requirement **(selection of Data Structures will be graded)**. **At least 2 different data structures in this project.**
- Clean and Efficient Code is a must.
- You can Use the built in Data Structures Libraries (example STLs in C++)
- You should save the data using files **DO NOT** use Database.
  -Load data from the program (Data Structures) into files only once at the end of your program after finishing.
  -Load data from files into the program (Data Structures) only once at the beginning of the program.

Evaluation:
The evaluation will be mainly based on the students' ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used these data structures and why they are better than other data structures in any given case.

Bonus	- Any extra (**non-trivial**) features to give more functionality to your application. - Amazing GUI.
Development Tool	- Visual Studio (However, you may use any language(s) and IDE(s) you want as long as it implements data structures.)

	- C++ Language

Title	**Online Marketplace Management System**
Description	An online marketplace is an e-commerce platform that links sellers with customers. - The **seller** can offer **products**. - To facilitate the search process, the products are divided into **categories**. - Every **customer** can browse or add products to his cart. - Display the total cost (receipt) for the customer after his confirmation. Implement a C++ program that allows the user to choose either he wants to sell or buy a product and then complete the process as a seller or customer using the following: - **Seller:** 1. ID 2. Name 3. Email - **Product** 1. ID 2. Name 3. Price 4. Category 5. Quantity 6. Seller ID - **Customer** 1. ID 2. Name 3. Address 4. Phone Number 5. Email 6. Customer Cart - **Cart** 1. List of Products 2. Total Price
Minimum Requirements	The program should allow the user to: 1. Choose to sell or buy products **(The user can switch between them In The Same Run - You should handle the adding of more than seller and customer**). 2. Seller: - Adding New Seller (Registration): Enter seller information (name, email, …) - Seller login using Email. - Add product and determine its category and quantity . 3. Customer:

	- Adding New Customer (Registration): Enter customer information (name, email, …) - Customer login using Email. - Browse products by its category. - Browse products by its name. - Add product to his cart. (Keep track of the available quantity, if the desired quantity is less than the available quantity the customer cannot add this product to his cart). - Confirm the buying and display the total price of the ordered products. 4. Add rating system: - The customers can give an overview(rate) out of 5 to the products or the sellers. (Add rate member to the seller and product classes) - The rate for each product or seller is calculated by the average of all the customers rates. - Displaying the products to customers according to its rates. ● You should use the **appropriate Data structures** for each requirement **(selection of Data Structures will be graded)**. **At least 2 different data structures in this project.** ● Clean and Efficient Code is a must. ● You can Use the built in Data Structures Libraries (example STLs in C++) ● You should save the data using files **DO NOT** use Database. -Load data from the program (Data Structures) into files only once at the end of your program after finishing. -Load data from files into the program (Data Structures) only once at the beginning of the program. **Evaluation:** The evaluation will be mainly based on the students' ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used these data structures and why they are better than other data structures in any given case.
Bonus	- Any extra (**non-trivial**) features to give more functionality to your application. - Amazing GUI.
Development Tool	- Visual Studio (However, you may use any language(s) and IDE(s) you want as long as it implements data structures.) - C++ Language
Reference Material	Amazon

TA: Eman Hamdi

emanhamdi@cis.asu.edu.eg

Title	**Mini وصلني**

Description	Your customer wishes to reach his destination in the shortest possible path. Implement a C++ program that allows the user to enter the graph which represents the cities and distances between them, then find the shortest path to reach the destination from any given source city.
Minimum Requirements	-The program should allow the user to: 1. Add graph (Cities and Distances between them) 2. Display graph data 3. Update graph data -Add new city -Add new edge -Delete city (Don't forget the edges that are connected to this city) -Delete edge (The user will enter the names of two cities to delete the edge between them) 4. Traverse the graph (BFS and DFS) 5. Determine two cities as source and destination then find the shortest path between them4 6. You must **-Compute and display total distance of the shortest path** **-Display the shortest path (Cities names in the shortest path)** **<u>-You must implement</u>** 1-Breadth First Search (BFS) 2-Depth First Search (DFS) 3-Dijkstra's Algorithm **-Use the suitable Data Structure for each Algorithm** **-You must use three types or more of data structures in the project** **-Implement the graph using Adjacency list** **-Save the graph data in file so the user can close the program and reopen it to continue working on the saved graph** **-The program must run continuously until the user choose exit** **<u>For saving Data:</u>** -Load data from the program (Data Structures) into files **only onc**e at the end of your program after finishing. -Load data from files into the program (Data Structures) **only once** at the beginning of the program. Evaluation: The evaluation will be mainly based on the students' ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used these data structures and why they are better than other data structures in any given case. Clean code is a must.

	Using ready-made libraries with ready-to-use data structures is allowed.
Bonus	-Additional algorithms to find the shortest path. -GUI
Development Tool	-C++ Language / Visual Studio IDE.

Title	**Vaccine Tracking System**
Description	**Implement an “Egyptian Vaccine Tracking System” in C++. That allows the users to enter their personal and vaccination information and let the administrator calculate basic statistics to draw insights from such data.** **[https://www.washingtonpost.com/graphics/2020/health/covid-vaccine-states-distribution-doses/]** What is meant by Egyptian is that it should be specifically tailored to tracking the vaccination process for Egyptians in Egypt in the sense that the National IDs should be 13 numbers, the user can only enter Egyptian governorates, etc To make things easier, we won’t account for Egyptians living abroad or non-Egyptian residents. Only Egyptian residents.
Minimum Requirements	The program should allow the user to: 1. Users may Add personal record containing: i. Full Name. ii. National ID. [Unique = Shouldn’t accept duplicates] iii. Password. iv. Gender. v. Age. vi. Governorate. vii. Already vaccinated or applying for vaccination? viii. If vaccinated, received only one or both doses? ix. If not, the user should be added to **a waiting list.** 2. Users should be able to display only their own records by entering the National ID and Password. 3. Users may Edit/Update any of their information. 4. Users may Delete their personal records. 5. Can sign in as Admin using Admin password. 6. Admin can: -View or delete (cannot edit/update) **all or any record(s) by entering only Nat. ID.** -View records filtered by the number of doses (one dose and two doses) -View records of vaccinated users ordered by the age of the user -View the record of the user that needs to be vaccinated from the waiting list -View Basic statistics: - Percentage of people registered in the system that have received the first dose. - Percentage of people registered in the system that have received both doses. - Percentage of Females and Males registered in the system.

	-You must use two types or more of data structures in the project **-The program must run continuously until the user choose exit** **-Save data in files so the user can close the program and reopen it to continue working on the saved data.** you can use separate files for different types of users (not vaccinated, vaccinated once, vaccinated twice) **For saving Data:** -Load data from the program (Data Structures) into files **only onc**e at the end of your program after finishing. -Load data from files into the program (Data Structures) **only once** at the beginning of the program. Evaluation: The evaluation will be mainly based on the students' ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used these data structures and why they are better than other data structures in any given case. Clean code is a must. Using ready-made libraries with ready-to-use data structures is allowed.
Bonus	- Calculate more advanced statistics [Check the Washington post article above] [You may need to ask the user for more information to get more statistical insights]. - GUI - Graphs / Data Visualization
Development Tool	- C++ Language / Visual Studio IDE.

TA: Beshoy Victor

beshoyvictor@cis.asu.edu.eg

Title	University Management System
Description	This project requires the use of a suitable data structure to store different courses and their prerequisites and based on this it will allow any student to take any course or not.
Minimum Requirements	● Basic Scenario: The System Admin will enter the courses list and their prerequisites, based on that the students will be allowed or disallowed from taking the desired courses. ● Course Data: 1. Name 2. Code 3. Requirement or elective.

	4. Maximum number of Students. 5. List of pre-required courses. 6. Hours. 7. Instructor. ● <u>Student Data:</u> 1. Name. 2. ID. 3. Email. 4. Password. 5. Maximum hours allowed. 6. Finished Courses. 7. Courses in progress. 8. Academic Year. 9. GPA. ● <u>Student Functionalities:</u> 1. Log in. 2. View List of all available courses. 3. Filter courses. 4. View details of a specific course. 5. Register for a course. 6. View all his/her courses. 7. View his/her courses grades and cumulative GPA. 8. Edit his/her data. ● <u>Admin Data:</u> 1. Name. 2. Password. ● <u>Admin Functionalities:</u> 1. Log in. 2. Add new students (Student email is auto-generated by the system) 3. Add a new course. 4. Enter course prerequisite. 5. View List of all students in a specific course. 6. Add course grade for each student. 7. View List of all courses (Finished - Progressed) of a specific student. 8. Edit all course data.
Important Notes	● You should use the **appropriate Data structures** for each requirement **(selection of Data Structures will be graded)**. **At least 2 different data structures in this project.** ● Store your data in files, no need for database

	○ Load data from files into the program (Data Structures) only once at the beginning of the program. ○ Load data from the program (Data Structures) into files only once at the end of your program after finishing. ● Clean and Efficient Code is a must. ● You can Use the built in DataStructures Libraries (eg.STLs in C++). ● You must fulfill **all** the above requirements.
Bonus	- Good GUI. - Data visualization for courses according to their dependency on one another. - Generate reports and insightful statistics. - Adding extra functionality (discuss and take approval first).
Development Tool	- Visual Studio - C++ Language

TA: Lobna Mady

lobna.mady@cis.asu.edu.eg

Title	**Railway Reservation System**
Description	Railway reservation system automates the process of reserving a railway ticket.
Minimum Requirements	There are two kinds of users: 1. Admin a. Login with the correct password otherwise he is not permitted to use the system b. Add/Delete details of a train i. Train name ii. Train number c. Add/update/Delete details of a trip i. Train number ii. Boarding point iii. Destination point iv. Number of available seats v. Fare per ticket vi. Date of travel d. View list of all trains e. View List of all available trips 2. Passenger(s) a. Register for the first time [email, password] b. Book Ticket with the following information: i. Passenger name

	ii. Boarding point iii. Destination point iv. Date of travel c. The passenger can login to view/update/delete his reservation ● Your system should keep track of all trips (keep the number updated when the user updates or deletes the reservation) going to the same destination at any time so as to offer for the passenger the time of the next train if the requested one is full. ● Use at least two data structures ● Store data in files or DB ● Write clean code is a **Must** ● Apply Object-Oriented programming concepts
Bonus	- GUI (**G**raphical **U**ser **I**nterface).
Development Tool	- Visual Studio - C++ Language

TA: Manar Sultan

ManarSultan@cis.asu.edu.eg

Title	**Modified صراحه**
Description	Modified-Saraha is a website that allows you to send messages in an anonymous way, you don't know any data about the sender except his ID. ● Main class **user** has: ○ ID ○ username ○ password ○ List of Contacts ○ Messages - you can add additional helper classes.
Minimum Requirements	- The website should allow the user to: 1. register/login 2. Add another user in his contacts ONLY if this user sends him a message. 3. Remove a contact and you still can see messages between you. 4. send a message to another user. 5. undo the last sent message. 6. search about contact in my contacts (report NOT FOUND if he doesn't exist). 7. view all contacts of a specific logged user sorted by number of messages between you. 8. view all the sent messages from latest to oldest.

	9. view all the received messages from specific contact. 10. put a message in favorites. 11. remove the oldest message from favorites. 12. view all favorite messages .
Notes	- Use suitable data structures for requirements (selection of Data Structure will be graded), **at least 2 different data structures**. - Clean and Efficient Code is a must. - You can Use the built in Data Structures Libraries (STLs in C++) - save the data using files: You should read data once at the beginning of your run then do your operations and accessing data structures then save in files at the end of your program *this will consume time and lines of code* ;)
Bonus	- GUI. - any additional useful feature discussed with me.
Development Tool	- Visual Studio - C++ Language

TA: Mariam Hassanein

mariam.hassanein@cis.asu.edu.eg

Title	**News Management System**

Description

News is important for a number of reasons within a society. Mainly to inform the public about events that are around them and may affect them.

Implement C++ program that allows users to get latest news, filter news by category, rate news and display them.

The News Management System should:

- validates user login data either correct or not.
- determines the user either admin or not.
- allows users to rate and view news.
- ignores trivial or incorrect news depending on news ratings (Rates below 2 should be hidden from users).
- displays news descendingly according to rates.
- allows a user to choose which categories he prefers.

News details are :

1- Title
2- Description.
3- Date.
4- Rate (1 to 5).
5- Category.

<u>Admin Functionalities :</u>

1. Login
2. Post/ Remove News
3. Update Existing News
4. Add New Category (Sport, Health) and assign news to them.
5. Display the average rate for a specific title.
6.

<u>User Functionalities:</u>

1- Register by Username and password.
2- Log in
3- Display latest news
4- Show news depending on specific category
5- Rate News

Minimum Requirements	● Apply all functions mentioned in the description section. ● Store your data in files or database ● The project will be a console application. ● Use suitable Data Structures (**at least 2 different datastructures**) in your project. (You can use STL)
Bonus	● GUI. ● Add user comments to news where each user can see other comments on news. ● Display news based on user's preferences. ● View a timeline for a specific title ● **Spam:** Add spam option for each news , if the user reports the post as spam , the post will disappear from this user's news and this news appears to others users with count of how many people spam it .
Evaluation	● will be based on the students' ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used these data structures and why they are better than other data structures in any given case. ● Clean code is a must.
Development Tool	● Visual Studio ● C++ Language

TA: Engy Abdallah

engy.abdallah@cis.asu.edu.eg

Title	**Fantasy**
Description	An application that implements **Only one** of the 2 below competitions: 1) Home Décor Fantasy **OR** 2) Football Fantasy Implement a program that allows the user to do the following requirements

Minimum requirements	For Home Décor: • Each player to have ID, username, password, budget & Decoration set (Min. 5 players) • Multiple Décor stores, each has ID, Name, List of products and their prices • Each User has ID, username, password & choice • Admin that can control data entry/ removal/ updates of stores & products • **<u>Competition Idea:</u>** - Competition to start as each player to buy products from stores, the target is to decorate a room in the house with a default "startup" budget of 100K - Each player can buy, sell or replace any of the products he owns within his budget (Budget to be updated with each process) - Player can search for any product, giving an output of product name, store it belongs to, points/ratings it has & price - Users will check the stores and give (Add, Update, Remove) ratings/points to products - Players who's their decoration sets get high ratings from users get the top rankings - Products price get increased or decreased per points given from users - By the end of the competition, Top 3 ranked players are displayed and top 1 receives a prize For Football: • Each player to have ID, username, password, budget & his created team of footballers (Min. 5 players) • Multiple Teams, each has ID, Name, List of footballers and their prices • Multiple leagues • Admin that can control data entry/ removal/ updates of teams & footballers ● **<u>Competition Idea:</u>** - Competition to start as each player to buy footballers, the

	target is to set up a team to play the next league match with a default "startup" budget of 100K - Each player can buy, sell or replace any of the footballers he owns within his budget (Budget to be updated with each process) - Player can search for any footballer, giving an output of footballer name, team he belongs to, points he has & price - After each match played in a specific league, Admin to update footballers points per the goals they scored thus footballers price get increased/ decreased per new points - Players who's their teams get higher points after each match played get the top rankings - By the end of the competition, Top 3 ranked players are displayed and top 1 receives a prize
Bonus	• Add violation rules • Lucky Wheel to pick a footballer/ product and add him/it to your team/ decoration-set (i.e., Random choice among footballers/ products) • GUI
Development Tools	• Use suitable Data Structures in your project (You can use STL) • Use <u>at least 2</u> different Data Structures