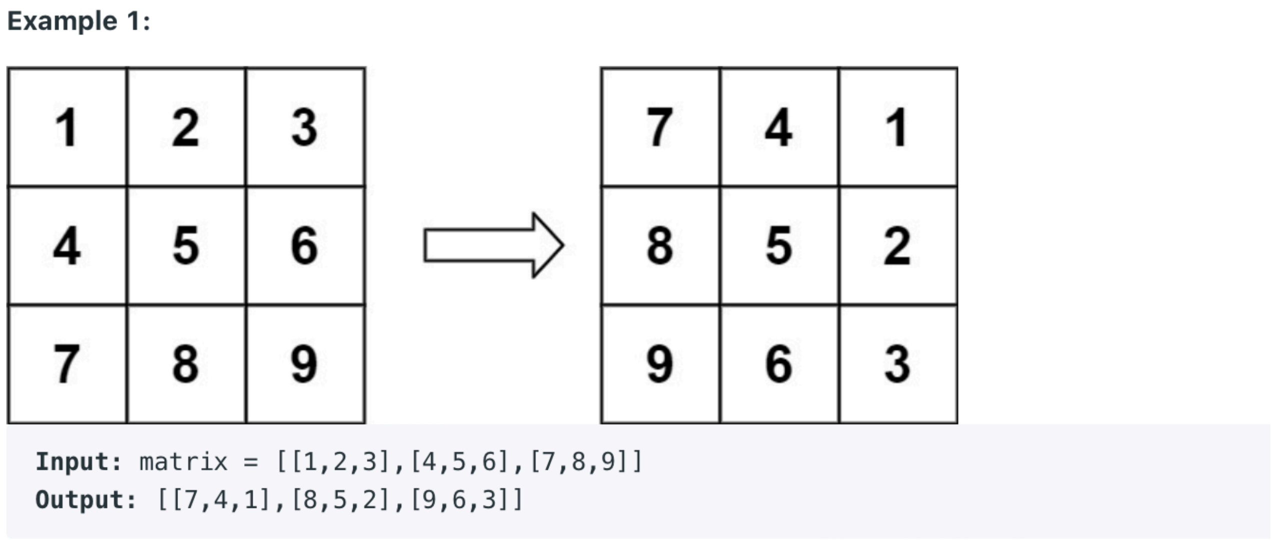
Assessment 1 - Set 1

Instructions : kindly make individual scripts for respective modules and push your code on the GitHub repository by the end of the day. Also please format the code if you want easy marks and cleaner code.

Module 1 - Python

Instructions : make separate '.py' file for each question.

1. Given an array of integers "nums" and an integer target, return indices of the two numbers such that they add up to target. You may assume that each input would have exactly one solution, and you may not use the same element twice. You can return the answer in any order.
2. Given a signed 32-bit integer x, return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range [-231, 231 - 1], then return 0. Assume the environment does not allow you to store 64-bit integers (signed or unsigned).
3. You are given an n x n 2D matrix representing an image, rotate the image by 90 degrees (clockwise).
4. You have to rotate the image in-place, which means you have to modify the input 2D matrix directly.



Example 1:

1	2	3
4	5	6
7	8	9

7	4	1
8	5	2
9	6	3

```
Input: matrix = [[1,2,3],[4,5,6],[7,8,9]]
Output: [[7,4,1],[8,5,2],[9,6,3]]
```

Module 1 : SQL

Q1.

Table: Scores

Column Name	Type
id	int
score	decimal

id is the primary key for this table.
Each row of this table contains the score of a game. Score is a floating point value with two decimal places.

Write an SQL query to rank the scores. The ranking should be calculated according to the following rules:
The scores should be ranked from the highest to the lowest.

If there is a tie between two scores, both should have the same ranking.
After a tie, the next ranking number should be the next consecutive integer value. In other words, there should be no holes between ranks.
Return the result table ordered by score in descending order.
The query result format is in the following example.

Example 1:
Input:
Scores table:

```
+----+-------+
| id | score |
+----+-------+
| 1  | 3.50  |
| 2  | 3.65  |
| 3  | 4.00  |
| 4  | 3.85  |
| 5  | 4.00  |
| 6  | 3.65  |
+----+-------+
```

Output:

```
+-------+------+
| score | rank |
+-------+------+
| 4.00  | 1    |
| 4.00  | 1    |
| 3.85  | 2    |
| 3.65  | 3    |
| 3.65  | 3    |
| 3.50  | 4    |
+———+------+
```

Q2.
Table: Person

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| id          | int     |
| email       | varchar |
+-------------+---------+
```

id is the primary key column for this table.
Each row of this table contains an email. The emails will not contain uppercase letters.

Write an SQL query to delete all the duplicate emails, keeping only one unique email with the smallest id. Note that you are supposed to write a DELETE statement and not a SELECT one.
After running your script, the answer shown is the Person table. The driver will first compile and run your piece of code and then show the Person table. The final order of the Person table does not matter.
The query result format is in the following example.

Example 1:
Input:
Person table:

```
+----+------------------+
| id | email            |
+----+------------------+
| 1  | john@example.com |
| 2  | bob@example.com  |
| 3  | john@example.com |
+----+------------------+
```

Output:

```
+----+------------------+
| id | email            |
+----+------------------+
| 1  | john@example.com |
| 2  | bob@example.com  |
+----+------------------+
```

Explanation: john@example.com is repeated two times. We keep the row with the smallest Id = 1.

Q3.

Table: World

```
+-------------+---------+
| Column Name | Type    |
+-------------+---------+
| name        | varchar |
| continent   | varchar |
| area        | int     |
| population  | int     |
| gdp         | int     |
+-------------+---------+
```

name is the primary key column for this table.
Each row of this table gives information about the name of a country, the continent to which it belongs, its area, the population, and its GDP value.

A country is big if:
it has an area of at least three million (i.e., 3000000 km2), or
it has a population of at least twenty-five million (i.e., 25000000).
Write an SQL query to report the name, population, and area of the big countries.
Return the result table in any order.
The query result format is in the following example.

Example 1:
Input:
World table:

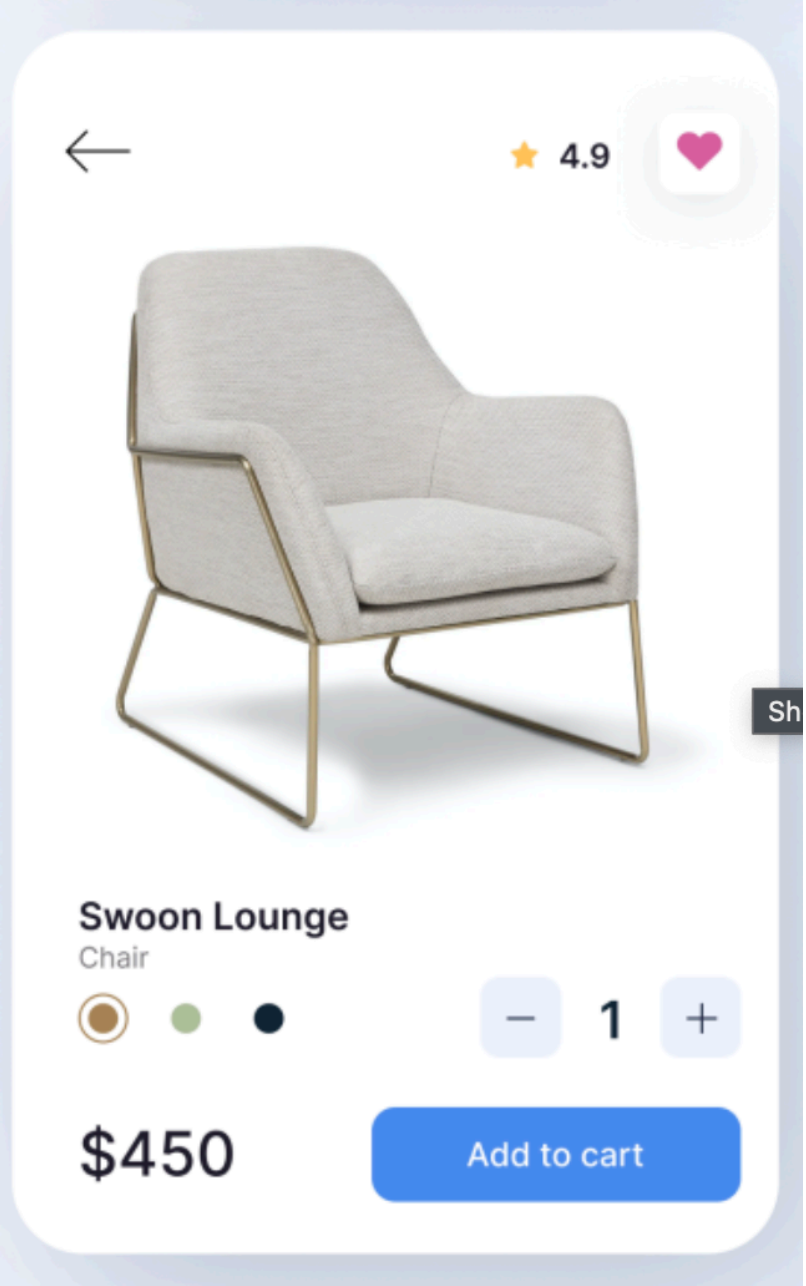
```
+-------------+-----------+---------+------------+--------------+
| name        | continent | area    | population | gdp          |
+-------------+-----------+---------+------------+--------------+
| Afghanistan | Asia      | 652230  | 25500100   | 20343000000  |
| Albania     | Europe    | 28748   | 2831741    | 12960000000  |
| Algeria     | Africa    | 2381741 | 37100000   | 188681000000 |
| Andorra     | Europe    | 468     | 78115      | 3712000000   |
| Angola      | Africa    | 1246700 | 20609294   | 100990000000 |
+-------------+-----------+---------+------------+--------------+
```

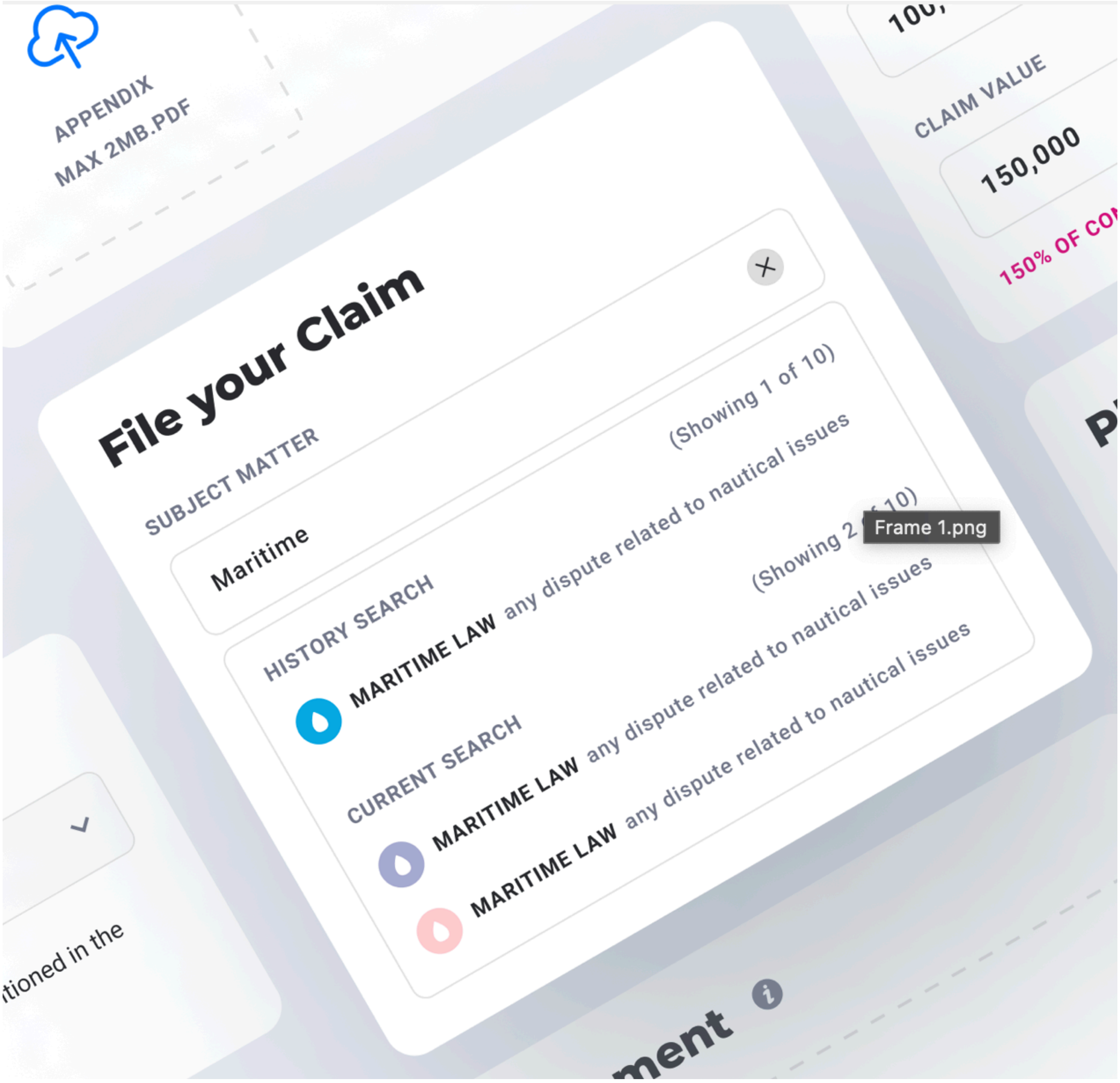
Output:

```
+-------------+------------+---------+
| name        | population | area    |
+-------------+------------+---------+
| Afghanistan | 25500100   | 652230  |
| Algeria     | 37100000   | 2381741 |
+— — — — — —+------------+---------+
```

Module 2: HTML (Fundamentals)

Draw the following layouts using HTML CSS





File your Claim

SUBJECT MATTER

Maritime

HISTORY SEARCH (Showing 1 of 10)

MARITIME LAW any dispute related to nautical issues

CURRENT SEARCH (Showing 2 of 10)

MARITIME LAW any dispute related to nautical issues

MARITIME LAW any dispute related to nautical issues