- **Total 100 points. Email the instructor this word file before 11:59 on 12/9/2023 (Saturday). NO late submission!**
- Please select a website and propose your design following the requirements.
- This is a **TEAM project** and each team **MUST** have **3-4** people.
  - If only one person: 10 points off.
  - If only two people: 5 points off.
- Complete all the sections and send this document **in World format** back to instructor before the deadline.
- **Name your file as Team_xxxx.docx where xxxx is the website name (highlight in red below). eg. Team_zillow.docx**
- You must use this word document and format to type your answers and email the word file back to the instructor.
- **Please confirm the final version with your team members, then send the final version of word file to the instructor and CC your team members. The email subject should be** "**Course_ID Team Research xxxx**" where xxxx is the website name and Course_ID is: TECH 3740-01
  - **5 points off if the name not follow the rules.**

Website selections (based on the Google spreadsheet, every team should select different website):

1. ___X__ http://www.zillow.com	2. _____ http://www.hotel.com	3. _____ http://www.walmart.com
4. _____ https://www.netflix.com	5. _____ https://weather.com	6. _____ https://www.united.com
7. _____ https://www.spotify.com	8. _____ https://www.uber.com/	9. _____ https://www.monster.com/

Team members (please list full name and role/tasks):

	Name	role/tasks	total # of Team meeting	total # of meeting attend (individual)
1	**Pedro Romero**	**Data Input/Output Analyst**	**2**	**2**
2	**Erick Chicas**	**Functional Analyst and Programmer**	**2**	**2**
3	**Richard Lopez**	**Database and Schema Designer**	**2**	**2**
4				

A. Identify and describe 5 essential data inputs for regular users to select or enter on the website that the inputs will be stored in the database. You need to list the HTML tag for each input used on the website. You must have at least 3 different HTML tags or input types and you need to give short explanation for each inputs. Your answer should focus on data, not actions. For example, search, click and submit buttons are actions, NOT data inputs. (15 pts).

#	Input data (field)	HTML tag/property	explanation
1	Location Search - Address, City, Zip code	`<input type="search">`	This is a text input where users can enter a specific address, city, ZIP code, or neighborhood for property searches. This data is crucial for filtering search results to a specific area.
2	Property Type - House, Apartment, Condo	<select>	This dropdown menu allows users to select the type of property they are interested in. Options might include houses, apartments, condos, etc. This input is vital for narrowing down search results to match the user's preferences.

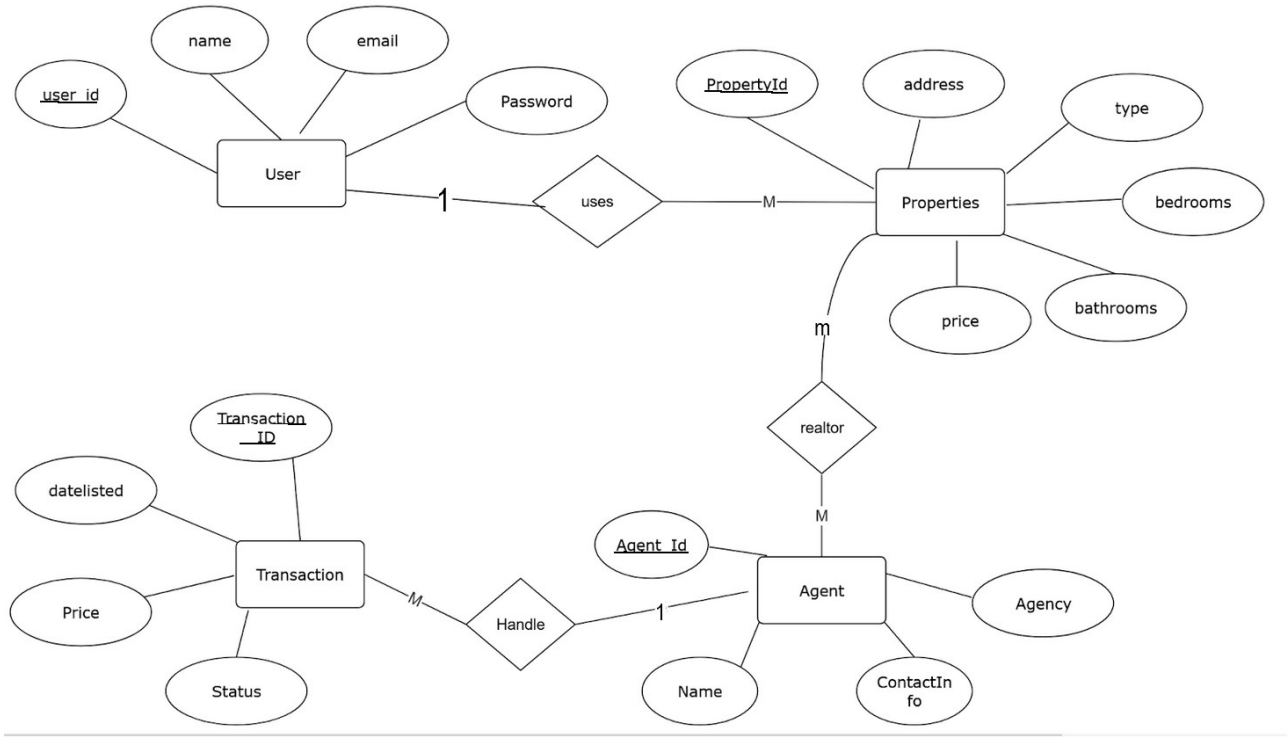
3	Number of Bedrooms/Bathrooms	`<input type="number">	These are numeric inputs where users can specify the minimum number of bedrooms and bathrooms they are looking for in a property. This is a common criterion for filtering property listings.
4	Price Range	<input type="range" name="min-price">, <input type="range" name="max-price">	These are numeric inputs where users can set a minimum and maximum price range for their property search. This helps in filtering the property listings according to the user's budget.
5	Account Registration Information	<input type="email" name="email">, <input type="password" name="password">	For creating an account, users need to input their name, email, and other personal information. This data is stored to create a user profile.

B. Identify and describe 5 output data or fields displayed for the regular users on the website. These output data fields should be the results from the database of the selected website. Please describe the recommend data types to store these data fields in the database. Your output data fields cannot be the same as your input data. (15 pts)

#	output data (field)	Data type in DB	explanation
1	Property Listing Price	Decimal/Numeric	The price of the real estate listing displayed to the user. It's crucial for users to assess affordability.
2	Property Address	Varchar	The address of the property, including street, city, and ZIP code, is essential for users to know the location.
3	Property Images	Blob (Binary Large Object)	Images of the property, showcasing interiors, exteriors, and any notable features.
4	Agent Contact Info	Varchar	Contact details of the property's listing agent or the seller, including name, phone number, and email.
5	Date Listed	Date	The date when the property was listed on the website.

C. Design and draw a meaningful E-R diagram **(not schema diagram)** for this website (20pts).

- Note1. Your design should have at least 4 Entities. Each entity should have at least 4 attributes. (10 pts)
- Note2. Please also identify the relationship and list the necessary attributes for the relationship. (5 pts)
- Note3. Please identify (1:1), (1:N), (N:M) for each relationship. (5 pts)
- Your previous input data in part A and output data fields in part B should be included in your E-R diagram.



D. Please write the SQL statement to create **all** required tables for your E-R diagram. Note: every entity should be a table, and most the relationship should be a table, too. You must define the correct primary key and foreign keys. (25 pts)

```
CREATE TABLE User (
   user_id INT PRIMARY KEY,
   name VARCHAR(255),
   email VARCHAR(255),
   password VARCHAR(255)
);
CREATE TABLE Properties (
   propertyId INT PRIMARY KEY,
   address VARCHAR(255),
   type VARCHAR(50),
   bedrooms INT,
   bathrooms INT,
   price DECIMAL(10, 2)
);
CREATE TABLE Transaction (
   transaction_ID INT PRIMARY KEY,
   dateListed DATE,
   price DECIMAL(10, 2),
   status VARCHAR(50),
   user_id INT,
   propertyId INT,
   FOREIGN KEY (user_id) REFERENCES User(user_id),
   FOREIGN KEY (propertyId) REFERENCES Properties(propertyId)
);
CREATE TABLE Agent (
   agent_Id INT PRIMARY KEY,
   name VARCHAR(255),
   contactInfo VARCHAR(255),
   agency VARCHAR(255)
);
CREATE TABLE Handle (
   transaction_ID INT,
   agent_Id INT,
   PRIMARY KEY (transaction_ID, agent_Id),
   FOREIGN KEY (transaction_ID) REFERENCES Transaction(transaction_ID),
   FOREIGN KEY (agent_Id) REFERENCES Agent(agent_Id)
);
CREATE TABLE Realtor (
   propertyId INT,
   agent_Id INT,
   PRIMARY KEY (propertyId, agent_Id),
   FOREIGN KEY (propertyId) REFERENCES Properties(propertyId),
   FOREIGN KEY (agent_Id) REFERENCES Agent(agent_Id)
);
```

E. Please identify and describe **3 different functions** on the selected website, these functions need to interact with the database. Please describe the possible to-do tasks for each function. For example, handle the inputs, do business related tasks, and generate the outputs. (15 pts, 5 pts for each function)

- Note1. Please write the possible program name and the pseudo codes (not real programs) of the to-do tasks for each function.
- Note2. Your programs should interact with the database. The database interaction can be SELECT, INSERT, UPDATE, DELETE with tables, views, and stored routines.
- Note3. Based on your tables, please write the SQL statements that each function should interact with the database. You can give an example and hardcode the values.

example:

The following USER LOGIN function is for demo only, you cannot include login function in your 3 functions.

Describe the function (demo)
USER LOGIN function: users need to enter their login and password to login to the system **Note: the functions must be based on your selected website.**

	Pseudo codes for function (demo)
HTML	**You need to describe the correct HTML tag/property on the web page for user to enter the login and password** **Note: You need to write the correct HTML tag/property with a short description**
PHP	**You need to describe the correct PHP statements how your PHP program receive the login & Password from the browser** **Note: You must specify the method using PHP statements with a short description**
SQL in PHP	**SELECT * FROM xxxdb.xxxtable where xxxcolumn=$xxxvariables;** **Note: your SQL statement must have PHP variables with a short description**

1. function 1:

Describe the function 1
New Account Function: users create new account by typing their email and password to the system.

	Pseudo codes for function 1
HTML	**<form id="newAccountForm">** ** <label for="email">Email:</label>** ** <input type="email" id="email" name="email" required> ** ** <label for="password">Password:</label>** ** <input type="password" id="password" name="password" required> ** ** <button type="button" onclick="createAccount()">Create Account</button>** **</form>** **post method form** **input email “email”** **input password “password”** **submit button**
PHP in PHP	**if ($_SERVER['REQUEST_METHOD'] === 'POST') {**

	// Retrieve form data $firstName = $_POST['firstName']; $lastName = $_POST['lastName']; $email = $_POST['email']; $password = $_POST['password']; check the input for the required fields and then match them with the existing entries in the database The PHP script checks if the form has been submitted using $_SERVER["REQUEST_METHOD"] == "POST". It retrieves the user input (email and password) from the $_POST array. It hashes the password using password_hash for secure storage. The user data is then inserted into a hypothetical database table named "users."
SQL	hashed_password = hash_function(user_input_password) INSERT INTO users (username, email, password) VALUES (user_input_username, user_input_email, hashed_password);

2. function 2:

Describe the function2
Saved Home Function: This function allows users to save the Homes they looked at

	Pseudo codes for function 2

HTML	<script>
	
	var savedHomes = [
	{ id: 1, address: '123 Main St' },
	{ id: 2, address: '456 Elm St' },
	// Add more saved homes as needed
	];
	
	// Function to display saved homes
	function displaySavedHomes() {
	var savedHomesList = document.getElementById('savedHomesList');

PHP	<?php
	
	function isHomeSaved($homeId, $userId) {
	}
	
	function saveHome($homeId, $userId) {
	if (!isHomeSaved($homeId, $userId)) {
	}
	}

	function removeSavedHome($homeId, $userId) {
	  if (isHomeSaved($homeId, $userId)) {
	  }
	}
	
	The PHP script checks if the form is submitted using $_SERVER["REQUEST_METHOD"] == "POST".
	It retrieves the users email address and the home ID from the form.
	It then inserts the saved home into the saved_homes table in the database.
	After saving, it retrieves and displays the list of saved homes for the user.

SQL in PHP	CREATE FUNCTION IsHomeSaved(userId INT, homeId INT) RETURNS BOOLEAN
	BEGIN
	  DECLARE isSaved BOOLEAN;
	  SELECT EXISTS(
	  SELECT 1 FROM saved_homes
	  WHERE user_id = userId AND home_id = homeId
	  ) INTO isSaved;
	
	  RETURN isSaved;
	END;
	
	INSERT INTO saved_homes (user_id, home_id) VALUES (1, 123);
	SELECT homes.address FROM saved_homes
	JOIN homes ON saved_homes.home_id = homes.id
	WHERE saved_homes.user_id = 1;

3. function 3:

Describe the function 3
Saved Searches Function: This function allows users to save where they have searched such as towns, types of houses, and prices

	Pseudo codes for function 3
HTML	<form id="savedSearchForm"> <label for="town">Town:</label> <input type="text" id="town" name="town" required> <label for="houseType">Type of House:</label> <input type="text" id="houseType" name="houseType" required> <label for="price">Price Range:</label> <input type="text" id="price" name="price" placeholder="e.g., $100,000 - $200,000" required>

	<button type="button" onclick="saveSearch()">Save Search</button> </form>

PHP	function saveSearch($userId, $searchName, $searchCriteria) {
	if (!isSearchSaved($userId, $searchName)) {
	}
	}
	
	function removeSavedSearch($userId, $searchName) {
	if (isSearchSaved($userId, $searchName)) {
	}
	}
	
	The PHP script checks if the form is submitted using $_SERVER["REQUEST_METHOD"] == "POST".
	It retrieves the users email address and the search data from the form.
	It then inserts the saved search into the saved_searches table in the database.
	After saving, it retrieves and displays the list of saved searches for the user.

SQL in PHP	CREATE FUNCTION isSearchSaved(userId INT, searchName VARCHAR(255))
	RETURNS BOOLEAN
	BEGIN
	DECLARE searchCount INT;
	SELECT COUNT(*) INTO searchCount
	FROM saved_searches
	WHERE user_id = userId AND search_name = searchName;
	
	RETURN (searchCount > 0);
	END;
	
	The getUserId function simulates user authentication and returns the user email.
	
	The INSERT query inserts the saved search into the saved_searches table.
	The SELECT query retrieves saved searches for the user and displays them.

G. Team presentation. (10 pts).

- Each team has 10 minutes.
- Present the draft report (MS-Word). No slides.
- Every team member needs to present.