CSCI-231 Server-side Development
Quiz 3 Key Name _____

Instructions

In the problems below you will be asked to design schema. For each schema you design, provide only the **keys** and **types** for the properties of the schema that are necessary for the problem.

For example, if you decide to define a schema for a User entity that has a **username** property that holds a String and a **tokens** property that holds an array of Strings you should write the following:

User
name: String
tokens: [String]

1. An app allows a user to search for a reservation by entering a reservation id. If the reservation exists the app displays the name of the user who made the reservation along with the date and time of the reservation and the number of people in the party.

   Redesigned the following schema utilizing the **Extended Reference Pattern**.

User	**Reservation**
name: String	userId: ObjectId (ref User)
password: String	dateTime: Date
...	partySize: Number

Reservation
userId: ObjectId (ref User)
dateTime: Date
partySize: Number
name: String

2. An app allows users to register their vehicles. All vehicles have a Vehicle Identification Number (VIN). Trucks have a towing capacity (measured in tons) whereas passenger cars have a trunk capacity (measured in sq. ft).

   Design schema that utilize the **Inheritance Pattern** that can store truck and passenger car data.

Vehicle
VIN: String
vehicleType: enum { Truck, PassengerCar }
towingCapacity: Number
trunkCapacity: Number

3. An app allows users to rate books as either 1 star, 2 star, or 3 star. When a user rates a book the appropriate rating count field and the numberOfRatings field are incremented.

   The app allows user to search for books that have a specific rating above a specified count that is entered by the user. For example, a user can find all books that have at least 100 3-star ratings.

   Redesign the following Book schema below to utilize the **Attribute Pattern**.

Book
title: String
oneStarCount: Number
twoStarCount: Number
threeStarCount: Number
numberOfRatings: Number

Book
title: String

ratings: [
{
type: enum {oneStar, twoStar, threeStar },
count: Number
}
]

numberOfRatings: Number

4. An app use the following hierarchy of information to classify biological organisms: Kingdom, Phylum, Class, Order, Family, Genus, and Species. Each organism also has a *description*.

   The app allows a user to retrieve data for all organisms that are in a particular Phylum, Class, Order, etc.

   Design, using the **Tree Pattern**, a schema that can be used to store information about the biological organisms.

Organism
description: String
classification: [String]

5. For every request that comes into the API server, the developers want to store the following information:

   a. userId of the user making the request,
   b. the date and time of the request,
   c. the method and endpoint path of the request, and
   d. the response status code.

   Developers would like to be able to find all requests from a particular user that were made during a specified interval of time.

   Design a schema that utilizes the **Bucket Pattern** and that will allow for the storing the request/ response information in buckets.

RequestLog
startTime: Date,
endTime: Date,

logs: [
{
userId: ObjectId,
time: Date,
method: String,
path: String,
responseCode: Number
}
]

6. An app allows users to play various games and saves the game name and score for every game that a user plays. When a user logs into the app, they see their username and a list containing the names of all of the games they have played along with the top score they have received in each of the games.

   Design schema for the app that utilizes the **Subset Pattern**.

User
username: String
highScores: [{ gameName: String, score: Number }]

GameResults
gameName: String
time: Date
score: Number

7. A user can view in an app a count of how many messages they have sent through the app. How would you implement this functionality if utilizing the **Computed Pattern**?

Add a count field to the User document and increment it when the user sends a message.