Problem Statement 1: Building the Game World (Data Modeling & CRUD Operations):

Objective: Design your game's data model in MongoDB and establish CRUD operations for data manipulation.

Task:

• Create a MongoDB database named "adventure_game".

```
dbs> use adventure_games
switched to db adventure_games
```

• Design three collections to represent the core elements of your game:

• Populate each collection with initial data to create your starting game world. This might include a few locations, characters, and items strategically placed.

✓ Locations (name, description, exits - references to other locations)

```
adventure_games> db.createCollection('Locations');
{ ok: 1 }
adventure_games> db.Locations.insertOne({
...     name: "Forest",
...     description: "A dense and dark forest with towering trees.",
...     exits: ['Village','Cave']
... })
{
  acknowledged: true,
  insertedId: ObjectId('6671a223dac462da2490df00')
}
```

✓ Characters (name, description, location - reference to a location)

```
adventure_games> db.createCollection('Characters');
{ ok: 1 }
```

```
adventure_games> db.Characters.insertOne({
...     name: "King",
...     description: "King of the world.",
...     exits:"Cave"
...     })
{
  acknowledged: true,
  insertedId: ObjectId('6671a2a8dac462da2490df01')
}
```

✓ Items (name, description, location - reference to a location)

```
adventure_games> db.createCollection('Items');
{ ok: 1 }
adventure_games> db.Items.insertOne({
...     name: "Sword",
...     description: "Kill the enimies.",
...     exits:"Village"
...     })
{
  acknowledged: true,
  insertedId: ObjectId('6671a31bdac462da2490df02')
}
```

```
adventure_games> db.Locations.insertOne({
...     name: "Forest",
...     description: "A dense and dark forest with towering trees.",
...     exits: ['Village','Cave']
... })
{
  acknowledged: true,
  insertedId: ObjectId('6671a223dac462da2490df00')
}
adventure_games> db.Characters.insertOne({
...     name: "King",
...     description: "King of the world.",
...     exits:"Cave"
...     })
{
  acknowledged: true,
  insertedId: ObjectId('6671a2a8dac462da2490df01')
}
adventure_games> db.Items.insertOne({
...     name: "Sword",
...     description: "Kill the enimies.",
...     exits:"Village"
...     })
{
  acknowledged: true,
  insertedId: ObjectId('6671a31bdac462da2490df02')
}
adventure_games>
```

• Implement functionalities (using a MongoDB client or driver) to perform CRUD operations:

• Create new locations, characters, and items.

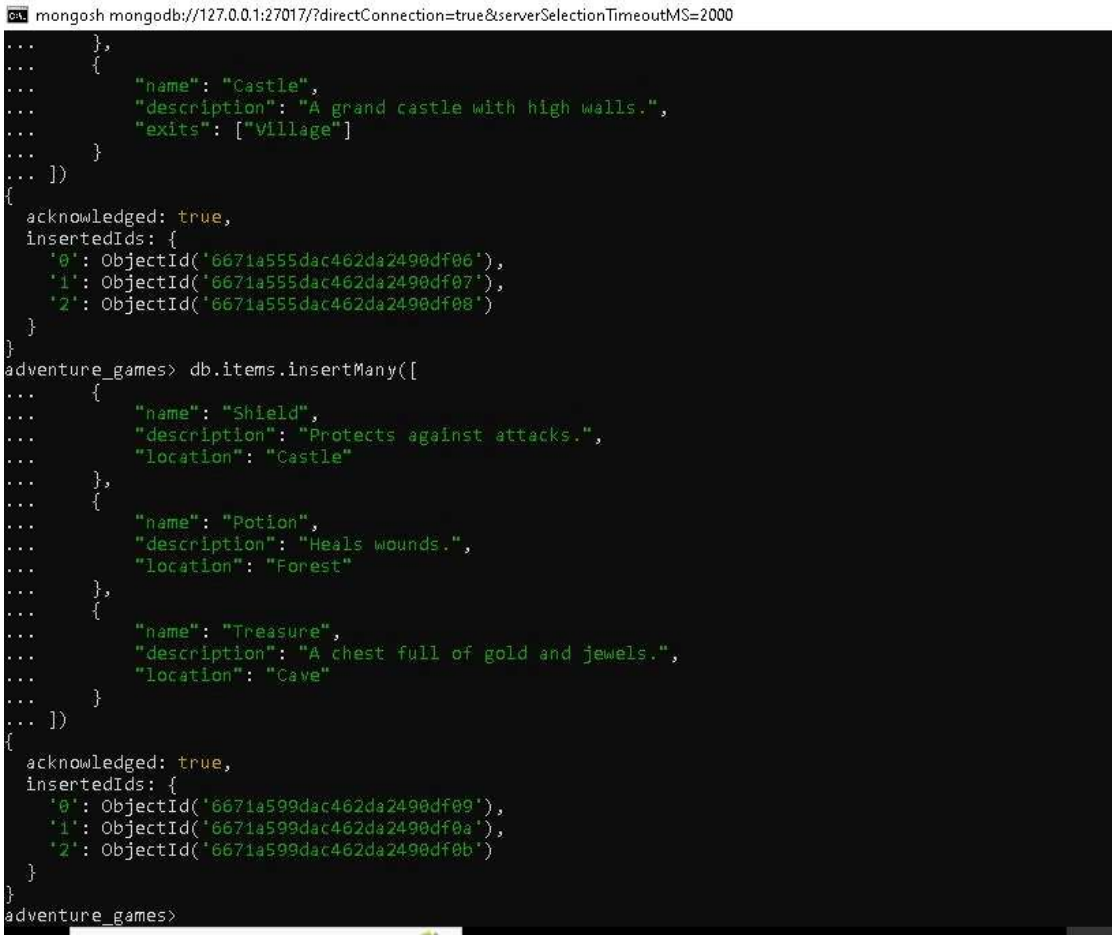
```
db.Characters.insertMany([
  {
    "name": "Hero",
    "description": "The brave protagonist of the story.",
    "location": "Village"
  },
  {
    "name": "Villager",
    "description": "A friendly villager.",
    "location": "Village"
  },
  {
```

```
    "name": "Dragon",
    "description": "A fearsome dragon.",
    "location": "Cave"
  }
])
```

```
db.Locations.insertMany([
  {
    "name": "Village",
    "description": "A small, peaceful village.",
    "exits": ["Forest", "Castle"]
  },
  {
    "name": "Cave",
    "description": "A dark and spooky cave.",
    "exits": ["Forest"]
  },
  {
    "name": "Castle",
    "description": "A grand castle with high walls.",
    "exits": ["Village"]
  }
])
```

```
db.items.insertMany([
  {
    "name": "Shield",
    "description": "Protects against attacks.",
```

```
    "location": "Castle"
  },
  {
    "name": "Potion",
    "description": "Heals wounds.",
    "location": "Forest"
  },
  {
    "name": "Treasure",
    "description": "A chest full of gold and jewels.",
    "location": "Cave"
  }
])
```



• Read existing data from each collection based on specific criteria (e.g., find a character by name).

1.Find All Documents in a Collection

adventure_games> db.Locations.find()

2.Find Documents that Match Query Criteria

adventure_games> db.Characters.find({name:'King'})

3.using $in

db.Locations.find({exits :{$in:['Castle','Forest']}})

4.using $regex

db.Items.find({name : {$regex :/^S/}})

• Update information about locations, characters, or items (e.g., move an item to a new location).

db.Items.updateOne({name:"Shield"},{$set:{location:"Village"}})

```
adventure_games> db.Items.updateOne({name:"Shield"},{$set:{location:"Village"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
adventure_games> db.Items.find()
[
  {
    _id: ObjectId('6671a31bdac462da2490df02'),
    name: 'Sword',
    description: 'Kill the enimies.',
    exits: 'Village'
  },
  {
    _id: ObjectId('6671ac9edac462da2490df0f'),
    name: 'Shield',
    description: 'Protects against attacks.',
    location: 'Village'
  },
  {
    _id: ObjectId('6671ac9edac462da2490df10'),
    name: 'Potion',
    description: 'Heals wounds.',
    location: 'Forest'
  },
  {
    _id: ObjectId('6671ac9edac462da2490df11'),
    name: 'Treasure',
    description: 'A chest full of gold and jewels.',
    location: 'Cave'
  }
]
```

• Delete unnecessary data from the collections (be mindful of maintaining game world consistency).

db.Items.deleteOne({name:"Sword"})

```
adventure_games> db.Items.deleteOne({name:"Sword"})
{ acknowledged: true, deletedCount: 1 }
adventure_games> db.Items.find()
[
  {
    _id: ObjectId('6671ac9edac462da2490df0f'),
    name: 'Shield',
    description: 'Protects against attacks.',
    location: 'Village'
  },
  {
    _id: ObjectId('6671ac9edac462da2490df10'),
    name: 'Potion',
    description: 'Heals wounds.',
    location: 'Forest'
  },
  {
    _id: ObjectId('6671ac9edac462da2490df11'),
    name: 'Treasure',
    description: 'A chest full of gold and jewels.',
    location: 'Cave'
  }
]
adventure_games>
```

Problem Statement 2. Exploring the Game World (MongoDB Queries):

Develop MongoDB queries to retrieve information relevant to the player's exploration:

✓ Describe the current location based on its name or ID.

```
adventure_games> db.Characters.find({name:"Dragon"},{location:1})
[ { _id: ObjectId('6671a4d0dac462da2490df05'), location: 'Cave' } ]
```

✓ List available exits from a specific location using the references stored in the collection.

```
adventure_games> db.Locations.find({name:"Village"},{exits:1})
[
  {
    _id: ObjectId('6671a555dac462da2490df06'),
    exits: [ 'Forest', 'Castle' ]
  }
]
```

✓ Find characters or items based on their properties (e.g., find a weapon in the current location).

```
adventure_games> db.Items.find({location:'Village'},{name:1})
[ { _id: ObjectId('6671ac9edac462da2490df0f'), name: 'Shield' } ]
adventure_games>
```

✓ Utilize logical operators (AND, OR) to construct more advanced queries (e.g., find a character named "Mage" located in the "Forest").

find location named forest or village which is having exit in cave

```
adventure_games> db.Locations.find({$and:[{$or:[{ name:"Forest"},{ name:"Village"}]},{ exits: "Cave" }]})
[
  {
    _id: ObjectId('6671a223dac462da2490df00'),
    name: 'Forest',
    description: 'A dense and dark forest with towering trees.',
    exits: [ 'Village', 'Cave' ]
  }
]
```