Slip.1

a) <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Project Management Form</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: yellow;

        }

        h1 {

            font-size: 6pt;

            color: red;

            text-align: center;

        }

        form {

            background-color: #d3f8ff;

            padding: 20px;

            border: 1px solid #ccc;

            width: 400px;

            margin: 50px auto;

            border-radius: 10px;

        }

        label {

            display: block;

            margin: 10px 0 5px;

        }

        input, select, textarea {

            width: 100%;

            padding: 8px;

            margin-bottom: 10px;

            border: 1px solid #ccc;

            border-radius: 5px;

        }

        .priority {

            display: flex;

            gap: 10px;

        }

        button {

            padding: 10px 20px;

            border: none;

            border-radius: 5px;

            cursor: pointer;

        }

        .submit-btn {

            background-color: green;

            color: white;

        }

        .clear-btn {

            background-color: blue;

            color: white;

        }

    </style>

</head>

<body>

    <h1>Project Management</h1>

    <form>

        <label for="project-name">Project Name</label>

        <input type="text" id="project-name" name="project-name" placeholder="Enter project name">

        <label for="assigned-to">Assigned To</label>

        <input type="text" id="assigned-to" name="assigned-to" placeholder="e.g. Mercy Peterson">

        <label for="start-date">Start Date</label>

      <input type="date" id="start-date" name="start-date">

        <label for="end-date">End Date</label>

        <input type="date" id="end-date" name="end-date">

        <label>Priority</label>

        <div class="priority">

            <label><input type="radio" name="priority" value="High"> High</label>

            <label><input type="radio" name="priority" value="Average"> Average</label>

            <label><input type="radio" name="priority" value="Low"> Low</label>

        </div>

        <label for="description">Description</label>

        <textarea id="description" name="description" placeholder="Enter description"></textarea>

        <button type="submit" class="submit-btn">Submit</button>

        <button type="reset" class="clear-btn">Clear</button>

    </form>

</body>

</html>

b)

// Create the database

use PropertySystem;

// Insert Owner Collection

db.Owner.insertMany([

{"owner\_id": 1, "name": "Mr. Patil", "contact\_details": "9876543210"},

{"owner\_id": 2, "name": "Ms. Sharma", "contact\_details": "8765432109"},

{"owner\_id": 3, "name": "Mr. Khan", "contact\_details": "7654321098"},

{"owner\_id": 4, "name": "Mrs. Desai", "contact\_details": "6543210987"},

{"owner\_id": 5, "name": "Mr. Reddy", "contact\_details": "5432109876"}

]);

// Insert Property Collection

db.Property.insertMany([

{"property\_id": 101, "name": "Green Villa", "area": "500 sqft", "rate": 80000, "location": "Nashik", "owner\_id": 1},

{"property\_id": 102, "name": "Blue Estate", "area": "1200 sqft", "rate": 120000, "location": "Pune", "owner\_id": 2},

{"property\_id": 103, "name": "Red Manor", "area": "700 sqft", "rate": 95000, "location": "Nashik", "owner\_id": 3},

{"property\_id": 104, "name": "Silver Towers", "area": "1000 sqft", "rate": 200000, "location": "Mumbai", "owner\_id": 4},

{"property\_id": 105, "name": "Golden Fields", "area": "800 sqft", "rate": 75000, "location": "Aurangabad", "owner\_id": 5}

]);

a) **Display Area-wise Property Details**

db.Property.aggregate([

{

$group: {

\_id: "$location", // Group by location

properties: { $push: "$$ROOT" } // Add all property details

}

}

]);

b) **Display Property Owned by 'Mr. Patil' Having Minimum Rate**

// Step 1: Get Mr. Patil's owner\_id (if you don't already know it)

db.Owner.find({ name: "Mr. Patil" });

// Step 2: Find properties with owner\_id = 1 and sort by rate

db.Property.find({ owner\_id: 1 }).sort({ rate: 1 }).limit(1);

c) **Give the Details of Owner Whose Property is at 'Nashik'**

db.Property.aggregate([

{

$match: { location: "Nashik" } // Match properties in Nashik

},

{

$lookup: {

from: "Owner", // Join with Owner collection

localField: "owner\_id", // Match owner\_id in Property

foreignField: "owner\_id", // Match owner\_id in Owner

as: "owner\_details" // Output field for joined details

}

}

]);

d) **Display Area of Property Whose Rate is Less Than 100,000**

db.Property.find(

{ rate: { $lt: 100000 } }, // Filter by rate

{ area: 1, \_id: 0 } // Project only the area field

);

Slip 2

1)<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bootstrap Grid</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-4">Column 1</div>

<div class="col-md-4">Column 2</div>

<div class="col-md-4">Column 3</div>

</div>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

2)use NewspaperDB

Publisher Collection:

db.Publisher.insertMany([

{ "publisher\_id": 1, "name": "Lokmat Media", "state": "Maharashtra" },

{ "publisher\_id": 2, "name": "Bennett, Coleman & Co.", "state": "Maharashtra" },

{ "publisher\_id": 3, "name": "Sakal Media Group", "state": "Maharashtra" },

{ "publisher\_id": 4, "name": "DB Corp Ltd.", "state": "Gujarat" },

{ "publisher\_id": 5, "name": "Indian Express Group", "state": "Delhi" }

]);

**Newspaper Collection:**

db.Newspaper.insertMany([

{ "newspaper\_id": 1, "name": "Lokmat Marathi", "language": "Marathi", "publisher\_id": 1, "sales": 90000, "city": "Nashik" },

{ "newspaper\_id": 2, "name": "The Times of India", "language": "English", "publisher\_id": 2, "sales": 120000, "city": "Mumbai" },

{ "newspaper\_id": 3, "name": "Sakal", "language": "Marathi", "publisher\_id": 3, "sales": 85000, "city": "Pune" },

{ "newspaper\_id": 4, "name": "Gujarat Samachar", "language": "Gujarati", "publisher\_id": 4, "sales": 60000, "city": "Ahmedabad" },

{ "newspaper\_id": 5, "name": "Divya Bhaskar", "language": "Gujarati", "publisher\_id": 4, "sales": 70000, "city": "Surat" }

]);

**City Collection:**

db.City.insertMany([

{ "city\_id": 1, "name": "Nashik", "state": "Maharashtra" },

{ "city\_id": 2, "name": "Mumbai", "state": "Maharashtra" },

{ "city\_id": 3, "name": "Pune", "state": "Maharashtra" },

{ "city\_id": 4, "name": "Ahmedabad", "state": "Gujarat" },

{ "city\_id": 5, "name": "Surat", "state": "Gujarat" }

]);

Query a) **List All Newspapers Available in “NASHIK” City**

db.Newspaper.find({ city: "Nashik" });

Query b) **List All Newspapers of “Marathi” Language**

db.Newspaper.find({ language: "Marathi" });

Query c) **Count the Number of Publishers in “Gujarat” State**

db.Publisher.find({ state: "Gujarat" }).count();

Query d) **Cursor to Show Newspapers with Highest Sale in Maharashtra State**

db.Newspaper.aggregate([

{ $match: { city: { $in: ["Mumbai", "Nashik", "Pune"] } } },

{ $sort: { sales: -1 } },

{ $limit: 1 }

]);

Slip 3

1) <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Image Thumbnails</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

    <div class="container mt-5">

        <div class="row">

            <div class="col-md-3">

                <div class="card">

                    <img src="https://via.placeholder.com/150" class="card-img-top" alt="Placeholder Image">

                    <div class="card-body">

                        <p class="card-text">Image 1 Description</p>

                    </div>

                </div>

            </div>

            <div class="col-md-3">

                <div class="card">

                    <img src="images/pic1.jpg" class="card-img-top" alt="Picture 1">

                    <div class="card-body">

                        <p class="card-text">Image 2 Description</p>

                    </div>

                </div>

            </div>

            <div class="col-md-3">

                <div class="card">

                    <img src="https://www.example.com/image.jpg" class="card-img-top" alt="Online Image">

                    <div class="card-body">

                        <p class="card-text">Image 3 Description</p>

                    </div>

                </div>

            </div>

            <div class="col-md-3">

                <div class="card">

                    <img src="image4.jpg" class="card-img-top" alt="Image 4">

                    <div class="card-body">

                        <p class="card-text">Image 4 Description</p>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

2) use companyDB;

Insert Sample Data:

**Insert into Employee Collection**

db.Employee.insertMany([

{ "emp\_id": 1, "name": "John Doe", "department": "Sales", "salary": 60000 },

{ "emp\_id": 2, "name": "Jane Smith", "department": "HR", "salary": 45000 },

{ "emp\_id": 3, "name": "Alice Johnson", "department": "Sales", "salary": 70000 },

{ "emp\_id": 4, "name": "Bob Brown", "department": "Marketing", "salary": 55000 },

{ "emp\_id": 5, "name": "Charlie Davis", "department": "IT", "salary": 80000 }

]);

**Insert into Department Collection**

db.Department.insertMany([

{ "dept\_id": 1, "name": "Sales", "employee\_count": 2 },

{ "dept\_id": 2, "name": "HR", "employee\_count": 1 },

{ "dept\_id": 3, "name": "Marketing", "employee\_count": 1 },

{ "dept\_id": 4, "name": "IT", "employee\_count": 1 }

]);

Query a) Display the name of the employee with the highest salary

db.Employee.find().sort({ salary: -1 }).limit(1);

Query b) Display the biggest department with the maximum number of employees

db.Department.find().sort({ employee\_count: -1 }).limit(1);

**Query c) Write a cursor to show department-wise employee information**

db.Employee.aggregate([

{ $group: { \_id: "$department", employees: { $push: "$$ROOT" } } }

]);

Query d) List all employees who work in the Sales department and have a salary > 50000

db.Employee.find({ department: "Sales", salary: { $gt: 50000 } });

Slip 4

1) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bootstrap Styled Table</title>

<!-- Link to Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container my-4">

<h2 class="text-center">User Information</h2>

<table class="table table-bordered table-striped">

<thead class="table-dark">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Email ID</th>

</tr>

</thead>

<tbody>

<tr>

<td>John</td>

<td>Doe</td>

<td>john.doe@example.com</td>

</tr>

<tr>

<td>Jane</td>

<td>Smith</td>

<td>jane.smith@example.com</td>

</tr>

<tr>

<td>Sam</td>

<td>Wilson</td>

<td>sam.wilson@example.com</td>

</tr>

</tbody>

</table>

</div>

<!-- Link to Bootstrap JS (Optional, for advanced features) -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

2) use documentDB

//Insert into Hospital Collection

db.Hospital.insertMany([

{ "hospital\_id": 1, "name": "Nashik General Hospital", "city": "Nashik", "specializations": ["Pediatric", "Orthopedic"], "rating": 4.5 },

{ "hospital\_id": 2, "name": "Sunrise Hospital", "city": "Nashik", "specializations": ["Gynaec", "Cardiology"], "rating": 3.8 },

{ "hospital\_id": 3, "name": "City Care Hospital", "city": "Pune", "specializations": ["Orthopedic", "Neurology"], "rating": 4.2 },

{ "hospital\_id": 4, "name": "Wellness Hospital", "city": "Mumbai", "specializations": ["Dermatology", "Pediatric"], "rating": 4.0 },

{ "hospital\_id": 5, "name": "Life Care Hospital", "city": "Nashik", "specializations": ["Pediatric", "Neurology"], "rating": 4.7 },

{ "hospital\_id": 6, "name": "Family Hospital", "city": "Nashik", "specializations": ["Gynaec", "Dermatology"], "rating": 4.1 },

{ "hospital\_id": 7, "name": "Central Hospital", "city": "Ahmednagar", "specializations": ["Orthopedic", "Cardiology"], "rating": 3.9 },

{ "hospital\_id": 8, "name": "Healing Touch Hospital", "city": "Mumbai", "specializations": ["Pediatric", "Orthopedic"], "rating": 4.4 },

{ "hospital\_id": 9, "name": "Green Valley Hospital", "city": "Nashik", "specializations": ["Neurology", "Cardiology"], "rating": 3.7 },

{ "hospital\_id": 10, "name": "Global Health Hospital", "city": "Pune", "specializations": ["Orthopedic", "Gynaec"], "rating": 4.3 }

]);

//Insert into Doctor Collection

db.Doctor.insertMany([

{ "doctor\_id": 1, "name": "Dr. Deshmukh", "hospitals": [1, 5, 6] },

{ "doctor\_id": 2, "name": "Dr. Sharma", "hospitals": [2, 4] },

{ "doctor\_id": 3, "name": "Dr. Mehta", "hospitals": [3, 7, 10] },

{ "doctor\_id": 4, "name": "Dr. Joshi", "hospitals": [1, 8] },

{ "doctor\_id": 5, "name": "Dr. Patil", "hospitals": [2, 6] },

{ "doctor\_id": 6, "name": "Dr. Kapoor", "hospitals": [5, 9] },

{ "doctor\_id": 7, "name": "Dr. Reddy", "hospitals": [3, 10] },

{ "doctor\_id": 8, "name": "Dr. Khan", "hospitals": [4, 8] },

{ "doctor\_id": 9, "name": "Dr. Iyer", "hospitals": [7, 2] },

{ "doctor\_id": 10, "name": "Dr. Pillai", "hospitals": [6, 9] }

]);

//Insert into Review Collection

db.Review.insertMany([

{ "review\_id": 1, "hospital\_id": 1, "reviewer\_name": "Alice", "review\_text": "Great pediatric care!", "rating": 5 },

{ "review\_id": 2, "hospital\_id": 2, "reviewer\_name": "Bob", "review\_text": "Helpful staff.", "rating": 4 },

{ "review\_id": 3, "hospital\_id": 3, "reviewer\_name": "Charlie", "review\_text": "Excellent facilities.", "rating": 4.5 },

{ "review\_id": 4, "hospital\_id": 4, "reviewer\_name": "Daisy", "review\_text": "Good dermatology unit.", "rating": 4 },

{ "review\_id": 5, "hospital\_id": 5, "reviewer\_name": "Eve", "review\_text": "Exceptional neurology team.", "rating": 4.7 }

]);

Query a) List the names of hospitals with [specialization]

db.Hospital.find({ specializations: "Pediatric" }, { name: 1, \_id: 0 });

Query b) List the names of all hospitals located in [city]

db.Hospital.find({ city: "Nashik" }, { name: 1, \_id: 0 });

Query c) List the names of hospitals where Dr. Deshmukh visits

db.Doctor.aggregate([

{ $match: { name: "Dr. Deshmukh" } },

{ $lookup: {

from: "Hospital",

localField: "hospitals",

foreignField: "hospital\_id",

as: "hospital\_details"

} },

{ $project: { "hospital\_details.name": 1, \_id: 0 } }

]);

Query d) List the names of hospitals whose rating >= 4

db.Hospital.find({ rating: { $gte: 4 } }, { name: 1, \_id: 0 });

Slip 5

1)<!DOCTYPE html><html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Person List</title>

<style>

table {

border: 2px solid #3498db; /\* Blue border around table \*/

border-radius: 10px; /\* Rounded corners for table \*/

width: 50%; /\* Table width \*/

margin: 20px auto; /\* Center the table horizontally \*/

border-collapse: collapse; /\* Collapse table borders \*/

} th, td {

padding: 12px 15px; /\* Padding for cells \*/

text-align: left; /\* Left align text \*/

border: 1px solid #3498db; /\* Border for each cell \*/

} th {

background-color: #3498db; /\* Background color for header \*/

color: white; /\* Text color for header \*/

} h2 {

text-align: center;

color: #3498db;

}

</style></head><body>

<h2>List of Persons</h2>

<table> <thead> <tr>

<th>Srno</th>

<th>Person Name</th>

<th>Age</th>

<th>Country</th>

</tr></thead>

<tbody> <tr>

<td>1</td>

<td>John Doe</td>

<td>30</td>

<td>USA</td>

</tr><tr>

<td>2</td>

<td>Jane Smith</td>

<td>25</td>

<td>UK</td>

</tr><tr>

<td>3</td>

<td>Sam Wilson</td>

<td>28</td>

<td>Australia</td>

</tr> </tbody> </table></body></html>

2) use companyDB

//Insert Data into Projects Collection

db.Projects.insertMany([

{ "project\_id": 1, "project\_name": "Project Alpha", "project\_type": "Software", "duration": 6, "employees": [101, 102, 103] },

{ "project\_id": 2, "project\_name": "Project Beta", "project\_type": "Marketing", "duration": 3, "employees": [104, 105] },

{ "project\_id": 3, "project\_name": "Project Gamma", "project\_type": "Software", "duration": 4, "employees": [101, 104] },

{ "project\_id": 4, "project\_name": "Project Delta", "project\_type": "Research", "duration": 2, "employees": [102, 105] },

{ "project\_id": 5, "project\_name": "Project Epsilon", "project\_type": "Software", "duration": 5, "employees": [103, 104, 105] }]);

//Insert Data into Employees Collection

db.Employees.insertMany([

{ "employee\_id": 101, "name": "Mr. Patil", "skills": ["Java", "MongoDB"], "projects": [1, 3] },

{ "employee\_id": 102, "name": "Mr. Shah", "skills": ["Python", "Machine Learning"], "projects": [1, 4] },

{ "employee\_id": 103, "name": "Mr. Desai", "skills": ["JavaScript", "Node.js"], "projects": [1, 5] },

{ "employee\_id": 104, "name": "Mr. Kumar", "skills": ["Digital Marketing", "SEO"], "projects": [2, 3, 5] },

{ "employee\_id": 105, "name": "Mr. Joshi", "skills": ["Marketing", "Sales"], "projects": [2, 4, 5] }]);

Query a) List all names of projects where project\_type = 'Software'

db.Projects.find({ "project\_type": "Software" }, { "project\_name": 1, "\_id": 0 });

Query b) List all the projects with duration greater than 3 months

db.Projects.find({ "duration": { $gt: 3 } }, { "project\_name": 1, "\_id": 0 });

**Query c) Count the number of employees working on a specific project, e.g., Project Alpha**

db.Projects.aggregate([

{ $match: { "project\_name": "Project Alpha" } },

{ $project: { \_id: 0, employee\_count: { $size: "$employees" } } }]);

**Query d) List the names of projects on which Mr. Patil is working**

**db.Employees.aggregate([**

**{ $match: { "name": "Mr. Patil" } },**

**{ $lookup: {**

**from: "Projects",**

**localField: "projects",**

**foreignField: "project\_id",**

**as: "project\_details"**

**} },**

**{ $project: { "project\_details.project\_name": 1, "\_id": 0 } }**

**]);**

**Slip 6**

1) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sample Web Page</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f9;

margin: 0;

padding: 0;

}

header {

background-color: #3498db;

color: white;

padding: 20px;

text-align: center;

} header img {

max-width: 100px;

margin-bottom: 15px;

} nav {

background-color: #2c3e50;

padding: 10px;

} nav a {

color: white;

padding: 10px;

text-decoration: none;

margin: 5px;

border-radius: 5px;

} nav a:hover {

background-color: #2980b9;

} section {

margin: 20px;

padding: 20px;

background-color: white;

border-radius: 8px;

box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

}table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

} table, th, td {

border: 1px solid #ddd;

} th, td {

padding: 12px;

text-align: left;

} th {

background-color: #3498db;

color: white;

} td { background-color: #f9f9f9;

} footer {

background-color: #2c3e50;

color: white;

text-align: center;

padding: 10px;

position: fixed;

width: 100%;

bottom: 0;

} </style></head><body> <header>

<img src="https://via.placeholder.com/100" alt="Logo"> <!-- Placeholder logo image -->

<h1>Welcome to Our Website</h1>

</header> <nav>

<a href="#home">Home</a>

<a href="#about">About Us</a>

<a href="#services">Services</a>

<a href="#contact">Contact</a>

</nav> <section>

<h2>Informative Text</h2>

<p>This is a sample webpage to demonstrate various elements such as logos, text, images, hyperlinks, and tables.</p>

<p>We offer a variety of services that will help you achieve your goals. Stay connected for more updates.</p>

<img src="https://via.placeholder.com/600x300" alt="Sample Image" style="width:100%; border-radius: 8px;">

<p>For more information, visit our <a href="https://www.example.com" target="\_blank">official website</a>.</p>

<h3>Our Services</h3>

<table> <thead> <tr>

<th>Service</th>

<th>Description</th>

<th>Price</th>

</tr> </thead> <tbody>

<tr><td>Web Design</td>

<td>Building responsive and beautiful websites</td>

<td>$500</td>

</tr><tr>

<td>SEO Services</td>

<td>Improving website visibility on search engines</td>

<td>$300</td>

</tr> <tr>

<td>Content Creation</td>

<td>Creating engaging content for your website</td>

<td>$200</td> </tr>

</tbody></table> </section> <!-- Footer section -->

<footer>

<p>&copy; 2024 Company Name. All rights reserved.</p>

</footer></body></html>

2) use InsuranceDB

db.Customers.insertMany([

{ customer\_id: 1, name: "Alice", age: 30, contact: "1234567890" },

{ customer\_id: 2, name: "Bob", age: 45, contact: "2345678901" },

{ customer\_id: 3, name: "Charlie", age: 28, contact: "3456789012" },

{ customer\_id: 4, name: "David", age: 50, contact: "4567890123" },

{ customer\_id: 5, name: "Emma", age: 35, contact: "5678901234" }

]);

db.Policies.insertMany([

{ policy\_id: 101, policy\_name: "Komal Jeevan", policy\_type: "Monthly", premium\_amount: 1000, company\_name: "LIC" },

{ policy\_id: 102, policy\_name: "Jeevan Anand", policy\_type: "Half-Yearly", premium\_amount: 5000, company\_name: "LIC" },

{ policy\_id: 103, policy\_name: "Health Cover", policy\_type: "Yearly", premium\_amount: 8000, company\_name: "Star Health" },

{ policy\_id: 104, policy\_name: "Term Plan", policy\_type: "Monthly", premium\_amount: 1500, company\_name: "HDFC" },

{ policy\_id: 105, policy\_name: "Child Plan", policy\_type: "Half-Yearly", premium\_amount: 3000, company\_name: "ICICI" }

]);

db.CustomerPolicies.insertMany([

{ customer\_policy\_id: 1, customer\_id: 1, policy\_id: 101 },

{ customer\_policy\_id: 2, customer\_id: 2, policy\_id: 102 },

{ customer\_policy\_id: 3, customer\_id: 3, policy\_id: 103 },

{ customer\_policy\_id: 4, customer\_id: 4, policy\_id: 101 },

{ customer\_policy\_id: 5, customer\_id: 5, policy\_id: 105 }

]);

**Query (a): List the details of customers who have taken the "Komal Jeevan" Policy**

db.CustomerPolicies.aggregate([

{

$lookup: {

from: "Policies",

localField: "policy\_id",

foreignField: "policy\_id",

as: "policy\_details"

}

},

{ $unwind: "$policy\_details" },

{ $match: { "policy\_details.policy\_name": "Komal Jeevan" } },

{

$lookup: {

from: "Customers",

localField: "customer\_id",

foreignField: "customer\_id",

as: "customer\_details"

} },

{ $unwind: "$customer\_details" },

{

$project: {

\_id: 0,

"customer\_details.name": 1,

"customer\_details.age": 1,

"customer\_details.contact": 1,

"policy\_details.policy\_name": 1

} } ]);

Query (b): Display the average premium amount

db.Policies.aggregate([

{ $group: { \_id: null, averagePremium: { $avg: "$premium\_amount" } } }

]);

Query (c): Increase the premium amount by 5% for policy\_type = "Monthly"

db.Policies.updateMany(

{ policy\_type: "Monthly" },

{ $mul: { premium\_amount: 1.05 } });

**Query (d): Count the number of customers who have taken policy\_type = "Half-Yearly"**

db.CustomerPolicies.aggregate([

{

$lookup: {

from: "Policies",

localField: "policy\_id",

foreignField: "policy\_id",

as: "policy\_details"

} },

{ $unwind: "$policy\_details" },

{ $match: { "policy\_details.policy\_type": "Half-Yearly" } },

{ $count: "half\_yearly\_customers" } ]);

Slip 7

1) <!DOCTYPE html> <html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>3D Text with Hover Effect</title>

<style>

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background: #f3f4f7;

font-family: Arial, sans-serif;

} .text-container {

font-size: 3rem;

font-weight: bold;

color: #333;

position: relative;

transition: transform 0.3s ease, color 0.3s ease;

} .text-container:hover {

transform: translateZ(20px);

color: #ff5733; /\* Highlighted color on hover \*/

text-shadow:

2px 2px 5px rgba(0, 0, 0, 0.3), /\* Light shadow \*/

-2px -2px 10px rgba(255, 87, 51, 0.5); /\* Glow effect \*/

} .text-container::before {

content: attr(data-text); /\* Mirror the text \*/

position: absolute;

top: 5px;

left: 5px;

color: rgba(0, 0, 0, 0.2);

z-index: -1;

} </style></head> <body>

<div class="text-container" data-text="Hover Me!">Hover Me!</div>

</body></html>

2) db.Customers.insertMany([

{ customer\_id: 1, first\_name: "Sam", last\_name: "Smith", contact: "1234567890", address: "Street 1, City A" },

{ customer\_id: 2, first\_name: "Sara", last\_name: "Connor", contact: "2345678901", address: "Street 2, City B" },

{ customer\_id: 3, first\_name: "John", last\_name: "Doe", contact: "3456789012", address: "Street 3, City C" },

{ customer\_id: 4, first\_name: "Sophia", last\_name: "Williams", contact: "4567890123", address: "Street 4, City D" },

{ customer\_id: 5, first\_name: "Michael", last\_name: "Brown", contact: "5678901234", address: "Street 5, City E" }

]);

db.Accounts.insertMany([

{ account\_id: 101, customer\_id: 1, branch\_name: "Downtown", account\_type: "Saving", open\_date: "2020-01-01" },

{ account\_id: 102, customer\_id: 2, branch\_name: "Uptown", account\_type: "Current", open\_date: "2021-05-10" },

{ account\_id: 103, customer\_id: 3, branch\_name: "Downtown", account\_type: "Loan", open\_date: "2019-03-15" },

{ account\_id: 104, customer\_id: 4, branch\_name: "Midtown", account\_type: "Saving", open\_date: "2020-01-01" },

{ account\_id: 105, customer\_id: 5, branch\_name: "Downtown", account\_type: "Loan", open\_date: "2022-07-20" }

]);

db.Transactions.insertMany([

{ transaction\_id: 1001, account\_id: 101, transaction\_date: "2023-01-15", amount: 500, transaction\_type: "Credit" },

{ transaction\_id: 1002, account\_id: 102, transaction\_date: "2023-02-10", amount: 1000, transaction\_type: "Debit" },

{ transaction\_id: 1003, account\_id: 103, transaction\_date: "2023-03-05", amount: 2000, transaction\_type: "Credit" },

{ transaction\_id: 1004, account\_id: 104, transaction\_date: "2023-03-15", amount: 800, transaction\_type: "Debit" },

{ transaction\_id: 1005, account\_id: 105, transaction\_date: "2023-04-10", amount: 3000, transaction\_type: "Credit" }

]);

**Query (a): List names of all customers whose first name starts with "S"**

db.Customers.find(

{ first\_name: { $regex: "^S" } },

{ \_id: 0, first\_name: 1, last\_name: 1 }

);

**Query (b): List all customers who opened an account on 2020-01-01 in a specific branch**

Assuming the branch is "Downtown

db.Accounts.aggregate([

{ $match: { open\_date: "2020-01-01", branch\_name: "Downtown" } },

{

$lookup: {

from: "Customers",

localField: "customer\_id",

foreignField: "customer\_id",

as: "customer\_details"

}

},

{ $unwind: "$customer\_details" },

{

$project: {

\_id: 0,

"customer\_details.first\_name": 1,

"customer\_details.last\_name": 1

}

}

]);

Query (c): List the names of customers where account\_type = "Saving"

db.Accounts.aggregate([

{ $match: { account\_type: "Saving" } },

{

$lookup: {

from: "Customers",

localField: "customer\_id",

foreignField: "customer\_id",

as: "customer\_details"

}

},

{ $unwind: "$customer\_details" },

{

$project: {

\_id: 0,

"customer\_details.first\_name": 1,

"customer\_details.last\_name": 1

}

}

]);

**Query (d): Count total number of loan account holders in a specific branch**

Assuming the branch is "Downtown":

db.Accounts.countDocuments({ branch\_name: "Downtown", account\_type: "Loan" });

Slip 8

1) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bootstrap Button Styles</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-5">

<h2>Bootstrap Buttons with Different Styles</h2>

<div class="mt-3">

<button type="button" class="btn btn-primary">Primary</button>>

<button type="button" class="btn btn-secondary">Secondary</button>

<button type="button" class="btn btn-success">Success</button>

<button type="button" class="btn btn-danger">Danger</button>

<button type="button" class="btn btn-warning">Warning</button>

<button type="button" class="btn btn-info">Info</button>

<button type="button" class="btn btn-light">Light</button>

<button type="button" class="btn btn-dark">Dark</button> </div> </div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script>

</body></html>

2) db.Items.insertMany([

{ item\_id: 1, item\_name: "Planner", status: "A", height: 10, tags: ["office", "stationery", "planner"] },

{ item\_id: 2, item\_name: "Notebook", status: "B", height: 7, tags: ["stationery", "paper"] },

{ item\_id: 3, item\_name: "Pen", status: "A", height: 5, tags: ["stationery", "writing"] },

{ item\_id: 4, item\_name: "Backpack", status: "C", height: 20, tags: ["bags", "travel"] },

{ item\_id: 5, item\_name: "Lamp", status: "B", height: 15, tags: ["furniture", "lighting"] }

]);

db.Categories.insertMany([

{ category\_id: 1, category\_name: "Stationery", tags: ["office", "stationery", "writing"] },

{ category\_id: 2, category\_name: "Bags", tags: ["bags", "travel"] },

{ category\_id: 3, category\_name: "Furniture", tags: ["furniture", "lighting"] },

{ category\_id: 4, category\_name: "Electronics", tags: ["electronics", "gadgets"] },

{ category\_id: 5, category\_name: "Home Decor", tags: ["decor", "furniture"] }

]);

db.Warehouses.insertMany([

{ warehouse\_id: 1, warehouse\_name: "Warehouse A", location: "New York" },

{ warehouse\_id: 2, warehouse\_name: "Warehouse B", location: "Los Angeles" },

{ warehouse\_id: 3, warehouse\_name: "Warehouse C", location: "Chicago" }

]);

db.Inventory.insertMany([

{ inventory\_id: 1, item\_id: 1, warehouse\_id: 1, quantity: 250 },

{ inventory\_id: 2, item\_id: 2, warehouse\_id: 1, quantity: 500 },

{ inventory\_id: 3, item\_id: 3, warehouse\_id: 2, quantity: 100 },

{ inventory\_id: 4, item\_id: 4, warehouse\_id: 3, quantity: 40 },

{ inventory\_id: 5, item\_id: 5, warehouse\_id: 2, quantity: 150 }

]);

Query (a): List all the items where the quantity is greater than 300

db.Inventory.aggregate([

{ $match: { quantity: { $gt: 300 } } },

{

$lookup: {

from: "Items",

localField: "item\_id",

foreignField: "item\_id",

as: "item\_details"

}

},

{ $unwind: "$item\_details" },

{

$project: {

\_id: 0,

"item\_details.item\_name": 1,

"item\_details.status": 1,

"item\_details.height": 1,

"quantity": 1

}

}

]);

Query (b): List all items which have tags less than 5

db.Items.aggregate([

{ $match: { "tags": { $size: { $lt: 5 } } } },

{

$project: {

\_id: 0,

"item\_name": 1,

"tags": 1

}

}

]);

**Query (c): List all items having status equal to "B" or having quantity less than 50 and height greater than 8**

db.Inventory.aggregate([

{

$lookup: {

from: "Items",

localField: "item\_id",

foreignField: "item\_id",

as: "item\_details"

}

},

{ $unwind: "$item\_details" },

{

$match: {

$or: [

{ "item\_details.status": "B" },

{ $and: [{ quantity: { $lt: 50 } }, { "item\_details.height": { $gt: 8 } }] }

]

}

},

{

$project: {

\_id: 0,

"item\_details.item\_name": 1,

"item\_details.status": 1,

"item\_details.height": 1,

"quantity": 1

}

}

]);

Query (d): Find all warehouses that keep the item "Planner" and have stock quantity less than 20

db.Inventory.aggregate([

{

$lookup: {

from: "Items",

localField: "item\_id",

foreignField: "item\_id",

as: "item\_details"

}

},

{ $unwind: "$item\_details" },

{

$match: {

"item\_details.item\_name": "Planner",

quantity: { $lt: 20 }

}

},

{

$lookup: {

from: "Warehouses",

localField: "warehouse\_id",

foreignField: "warehouse\_id",

as: "warehouse\_details"

}

},

{ $unwind: "$warehouse\_details" },

{

$project: {

\_id: 0,

"warehouse\_details.warehouse\_name": 1,

"warehouse\_details.location": 1

}

}]);

Slip 9

1) <!DOCTYPE html> <html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>College Admission - Student Registration</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f9;

margin: 0;

padding: 20px;

} .container {

width: 60%;

margin: 0 auto;

background-color: #fff;

padding: 20px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

border-radius: 8px;

} h2 {

text-align: center;

color: #333;

} form {

display: flex;

flex-direction: column;

} label {

font-size: 14px;

margin: 8px 0 4px;

color: #333;

} input, select, textarea {

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 5px;

} button {

padding: 12px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

} button:hover {

background-color: #0056b3;

}

</style></head>

<body>

<div class="container">

<h2>College Admission - Student Registration</h2>

<form action="#" method="post">

<label for="name">Full Name</label>

<input type="text" id="name" name="name" placeholder="Enter your full name" required>

<label for="email">Email Address</label>

<input type="email" id="email" name="email" placeholder="Enter your email" required>

<label for="phone">Phone Number</label>

<input type="tel" id="phone" name="phone" placeholder="Enter your phone number" required pattern="[0-9]{10}">

<label for="dob">Date of Birth</label>

<input type="date" id="dob" name="dob" required>

<label for="gender">Gender</label>

<select id="gender" name="gender" required>

<option value="" disabled selected>Select your gender</option>

<option value="Male">Male</option>

<option value="Female">Female</option>

<option value="Other">Other</option>

</select> <label for="course">Select Course</label>

<select id="course" name="course" required>

<option value="" disabled selected>Select course</option>

<option value="BSc Computer Science">BSc Computer Science</option>

<option value="BSc Mathematics">BSc Mathematics</option>

<option value="BA Economics">BA Economics</option>

<option value="BCom">BCom</option>

<option value="BBA">BBA</option>

</select>

<label for="address">Permanent Address</label>

<textarea id="address" name="address" rows="4" placeholder="Enter your permanent address" required></textarea>

<label for="search">Search College Preferences</label>

<input type="search" id="search" name="search" placeholder="Search for your college preference">

<label for="enrollment\_date">Preferred Enrollment Date</label>

<input type="date" id="enrollment\_date" name="enrollment\_date" required>

<label for="documents">Upload Documents</label>

<input type="file" id="documents" name="documents" accept=".jpg,.jpeg,.png,.pdf,.docx" required>

<button type="submit">Register</button> </form></div></body></html>

2) db.Customers.insertMany([

{ customer\_id: 1, customer\_name: "Mr. Patil", contact\_number: "9876543210", address: "123, ABC Street, Pimpri" },

{ customer\_id: 2, customer\_name: "Mrs. Deshmukh", contact\_number: "9988776655", address: "456, DEF Lane, Pune" },

{ customer\_id: 3, customer\_name: "Mr. Desai", contact\_number: "1234567890", address: "789, GHI Road, Mumbai" },

{ customer\_id: 4, customer\_name: "Mr. Dhillon", contact\_number: "1122334455", address: "101, JKL Square, Pune" },

{ customer\_id: 5, customer\_name: "Mr. D'souza", contact\_number: "2233445566", address: "102, MNO Circle, Pimpri" },

{ customer\_id: 6, customer\_name: "Ms. Gupta", contact\_number: "3344556677", address: "201, PQR Street, Mumbai" },

{ customer\_id: 7, customer\_name: "Mr. Patel", contact\_number: "4455667788", address: "202, STU Block, Pune" },

{ customer\_id: 8, customer\_name: "Mrs. Pandit", contact\_number: "5566778899", address: "303, VYZ Lane, Pimpri" },

{ customer\_id: 9, customer\_name: "Mr. Rane", contact\_number: "6677889900", address: "404, XYZ Avenue, Mumbai" },

{ customer\_id: 10, customer\_name: "Mr. Dixit", contact\_number: "7788990011", address: "505, ABC Street, Pune" }

]);

db.Loans.insertMany([

{ loan\_id: 1, customer\_id: 1, loan\_type: "Home Loan", loan\_amt: 500000, loan\_status: "Approved", loan\_start\_date: new Date("2023-01-15"), city: "Pimpri" },

{ loan\_id: 2, customer\_id: 2, loan\_type: "Personal Loan", loan\_amt: 200000, loan\_status: "Pending", loan\_start\_date: new Date("2023-02-20"), city: "Pune" },

{ loan\_id: 3, customer\_id: 3, loan\_type: "Car Loan", loan\_amt: 300000, loan\_status: "Approved", loan\_start\_date: new Date("2023-03-10"), city: "Mumbai" },

{ loan\_id: 4, customer\_id: 4, loan\_type: "Home Loan", loan\_amt: 450000, loan\_status: "Approved", loan\_start\_date: new Date("2023-04-05"), city: "Pune" },

{ loan\_id: 5, customer\_id: 5, loan\_type: "Personal Loan", loan\_amt: 150000, loan\_status: "Pending", loan\_start\_date: new Date("2023-05-12"), city: "Pimpri" },

{ loan\_id: 6, customer\_id: 6, loan\_type: "Education Loan", loan\_amt: 250000, loan\_status: "Approved", loan\_start\_date: new Date("2023-06-10"), city: "Mumbai" },

{ loan\_id: 7, customer\_id: 7, loan\_type: "Home Loan", loan\_amt: 700000, loan\_status: "Approved", loan\_start\_date: new Date("2023-07-18"), city: "Pune" },

{ loan\_id: 8, customer\_id: 8, loan\_type: "Personal Loan", loan\_amt: 100000, loan\_status: "Approved", loan\_start\_date: new Date("2023-08-25"), city: "Pimpri" },

{ loan\_id: 9, customer\_id: 9, loan\_type: "Car Loan", loan\_amt: 350000, loan\_status: "Approved", loan\_start\_date: new Date("2023-09-12"), city: "Mumbai" },

{ loan\_id: 10, customer\_id: 10, loan\_type: "Home Loan", loan\_amt: 600000, loan\_status: "Approved", loan\_start\_date: new Date("2023-10-05"), city: "Pune" }

]);

Query a: List all customers whose name starts with 'D'

db.Customers.find({ "customer\_name": { $regex: "^D", $options: 'i' } })

**Query b: List the names of customers in descending order who have taken a loan from Pimpri city**

db.Loans.aggregate([

{ $match: { "city": "Pimpri" } },

{ $lookup: {

from: "Customers",

localField: "customer\_id",

foreignField: "customer\_id",

as: "customer\_details"

}},

{ $unwind: "$customer\_details" },

{ $sort: { "customer\_details.customer\_name": -1 } },

{ $project: { "\_id": 0, "customer\_details.customer\_name": 1 } }

])

Query c: Display customer details having maximum loan amount

db.Loans.aggregate([

{ $sort: { "loan\_amt": -1 } },

{ $limit: 1 },

{ $lookup: {

from: "Customers",

localField: "customer\_id",

foreignField: "customer\_id",

as: "customer\_details"

}},

{ $unwind: "$customer\_details" },

{ $project: { "\_id": 0, "customer\_details": 1, "loan\_amt": 1 } }

])

Query d: Update the address of customer whose name is "Mr. Patil" and loan amount is greater than 100000

db.Customers.updateOne(

{ "customer\_name": "Mr. Patil" },

{ $set: { "address": "789, XYZ Lane, Pimpri" } },

{ $in: [

{ "loan\_amt": { $gt: 100000 } }

]})

Slip 10

1) <!DOCTYPE html> <html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Transition Effects Example</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f4f4f9;

margin: 0;

}

.transition-button {

padding: 15px 30px;

background-color: #3498db;

color: white;

font-size: 18px;

font-weight: bold;

border: none;

border-radius: 5px;

cursor: pointer;

transition: all 0.6s ease-in-out; /\* Transition effect on all properties \*/

}

.transition-button:hover {

background-color: #2ecc71; /\* Change background color \*/

width: 300px;

height: 70px; }

.transition-button:hover {

transition-delay: 0.3s; /\* Wait for 0.3s before starting transition \*/

}

</style></head>

<body> <button class="transition-button">Hover over me</button>

</body></html>

2) db.Customers.insertMany([

{ customer\_id: 1, customer\_name: "Alice", email: "alice@example.com", address: "123 Main St", city: "Pune", purchase\_history: [1, 2] },

{ customer\_id: 2, customer\_name: "Bob", email: "bob@example.com", address: "456 Park Ave", city: "Mumbai", purchase\_history: [3] },

{ customer\_id: 3, customer\_name: "Charlie", email: "charlie@example.com", address: "789 Oak Rd", city: "Pune", purchase\_history: [4, 5] },

{ customer\_id: 4, customer\_name: "David", email: "david@example.com", address: "321 Pine Ln", city: "Mumbai", purchase\_history: [6] },

{ customer\_id: 5, customer\_name: "Eva", email: "eva@example.com", address: "654 Maple Dr", city: "Bangalore", purchase\_history: [7] }

]);

db.Products.insertMany([

{ product\_id: 1, product\_name: "Laptop", brand\_id: 1, warranty\_period: 1, price: 50000, rating: 4.5 },

{ product\_id: 2, product\_name: "Smartphone", brand\_id: 2, warranty\_period: 2, price: 30000, rating: 4.2 },

{ product\_id: 3, product\_name: "Headphones", brand\_id: 1, warranty\_period: 1, price: 5000, rating: 4.7 },

{ product\_id: 4, product\_name: "Tablet", brand\_id: 3, warranty\_period: 1, price: 15000, rating: 3.8 },

{ product\_id: 5, product\_name: "Smartwatch", brand\_id: 2, warranty\_period: 1, price: 8000, rating: 4.3 },

{ product\_id: 6, product\_name: "Camera", brand\_id: 4, warranty\_period: 2, price: 25000, rating: 4.6 },

{ product\_id: 7, product\_name: "Gaming Console", brand\_id: 5, warranty\_period: 1, price: 30000, rating: 4.0 }

]);

db.Purchases.insertMany([

{ purchase\_id: 1, customer\_id: 1, product\_id: 1, purchase\_date: new Date("2023-07-01"), bill\_amt: 50000, quantity: 1 },

{ purchase\_id: 2, customer\_id: 1, product\_id: 3, purchase\_date: new Date("2023-07-15"), bill\_amt: 5000, quantity: 2 },

{ purchase\_id: 3, customer\_id: 2, product\_id: 2, purchase\_date: new Date("2023-08-01"), bill\_amt: 30000, quantity: 1 },

{ purchase\_id: 4, customer\_id: 3, product\_id: 4, purchase\_date: new Date("2023-08-10"), bill\_amt: 15000, quantity: 1 },

{ purchase\_id: 5, customer\_id: 3, product\_id: 5, purchase\_date: new Date("2023-08-12"), bill\_amt: 8000, quantity: 1 },

{ purchase\_id: 6, customer\_id: 4, product\_id: 6, purchase\_date: new Date("2023-08-15"), bill\_amt: 25000, quantity: 1 },

{ purchase\_id: 7, customer\_id: 5, product\_id: 7, purchase\_date: new Date("2023-08-20"), bill\_amt: 30000, quantity: 1 }

]);

db.Brands.insertMany([

{ brand\_id: 1, brand\_name: "Brand A", rating: 4.5 },

{ brand\_id: 2, brand\_name: "Brand B", rating: 4.2 },

{ brand\_id: 3, brand\_name: "Brand C", rating: 3.8 },

{ brand\_id: 4, brand\_name: "Brand D", rating: 4.6 },

{ brand\_id: 5, brand\_name: "Brand E", rating: 4.0 }

]);

Query a: List the names of products whose warranty period is one year

db.Products.find({ "warranty\_period": 1 }, { "product\_name": 1 })

Query b: List the customers who made purchases on "15/08/2023"

db.Purchases.aggregate([

{ $match: { "purchase\_date": new Date("2023-08-15") } },

{ $lookup: {

from: "Customers",

localField: "customer\_id",

foreignField: "customer\_id",

as: "customer\_details"

}},

{ $unwind: "$customer\_details" },

{ $project: { "\_id": 0, "customer\_details.customer\_name": 1 } } ])

Query c: Display the names of products with brand which have the highest rating

db.Products.aggregate([

{ $lookup: {

from: "Brands",

localField: "brand\_id",

foreignField: "brand\_id",

as: "brand\_details"

}},

{ $unwind: "$brand\_details" },

{ $sort: { "brand\_details.rating": -1 } },

{ $limit: 1 },

{ $project: { "\_id": 0, "product\_name": 1, "brand\_details.brand\_name": 1 } }

])

**Query d: Display customers who stay in a specific city and have a bill amount greater than 50000**

db.Purchases.aggregate([

{ $match: { "bill\_amt": { $gt: 50000 } } },

{ $lookup: {

from: "Customers",

localField: "customer\_id",

foreignField: "customer\_id",

as: "customer\_details"

}},

{ $unwind: "$customer\_details" },

{ $match: { "customer\_details.city": "Pune" } },

{ $project: { "\_id": 0, "customer\_details.customer\_name": 1, "customer\_details.city": 1 } }

])

Slip 11

1) <!DOCTYPE html>

<html>

<head>

    <title>Company XYZ</title>

    <style>

        body {

            margin: 0;

            font-family: Arial, sans-serif;

        }

        #header {

            background-color: #4CAF50;

            color: white;

            text-align: center;

            padding: 20px;

            font-size: 24px;

        }

        #menu {

            width: 20%;

            height: 100%;

            float: left;

            background-color: #f4f4f4;

            padding: 10px;

            box-shadow: 2px 0 5px rgba(0,0,0,0.1);

        }

        #content {

            width: 80%;

            height: 100%;

            float: left;

            padding: 10px;

        }

        a {

            display: block;

            margin: 10px 0;

            text-decoration: none;

            color: #4CAF50;

        }

        a:hover {

            text-decoration: underline;

        }

    </style>

</head>

<body>

<div id="header">

    Company XYZ

</div>

<div id="menu">

    <h3>Departments</h3>

    <a href="#" onclick="loadContent('hr')">Human Resources</a>

    <a href="#" onclick="loadContent('it')">Information Technology</a>

    <a href="#" onclick="loadContent('finance')">Finance</a>

</div>

<div id="content">

    <h2>Welcome to Company XYZ</h2>

    <p>Please select a department to view more information.</p>

</div>

<script>

    function loadContent(department) {

        const content = {

            hr: `

                <h2>Human Resources</h2>

                <p>The HR department is responsible for employee welfare, recruitment, and training.</p>

            `,

            it: `

                <h2>Information Technology</h2>

                <p>The IT department manages the company's technology infrastructure.</p>

            `,

            finance: `

                <h2>Finance</h2>

                <p>The Finance department handles budgeting, accounts, and financial planning.</p>

            `

        };

        // Update content area with the selected department information

        document.getElementById('content').innerHTML = content[department] || '<h2>Department Not Found</h2>';

    }

</script></body></html>

2) db.products.insertMany([

{ "\_id": 1, "name": "Laptop", "price": 50000, "stock": 10 },

{ "\_id": 2, "name": "Smartphone", "price": 20000, "stock": 50 },

{ "\_id": 3, "name": "Tablet", "price": 15000, "stock": 30 },

{ "\_id": 4, "name": "Smartwatch", "price": 10000, "stock": 20 },

{ "\_id": 5, "name": "Headphones", "price": 5000, "stock": 100 }

]);

db.customers.insertMany([

{ "\_id": 1, "name": "Mr. Rajiv", "email": "rajiv@example.com", "phone": "1234567890" },

{ "\_id": 2, "name": "Ms. Priya", "email": "priya@example.com", "phone": "0987654321" },

{ "\_id": 3, "name": "Mr. Anil", "email": "anil@example.com", "phone": "1112223334" },

{ "\_id": 4, "name": "Ms. Sara", "email": "sara@example.com", "phone": "2223334445" },

{ "\_id": 5, "name": "Mr. Arjun", "email": "arjun@example.com", "phone": "3334445556" }

]);

db.orders.insertMany([

{ "\_id": 1, "customer\_id": 1, "products": [{ "product\_id": 1, "quantity": 1 }, { "product\_id": 2, "quantity": 1 }], "total\_value": 70000, "processed": true },

{ "\_id": 2, "customer\_id": 2, "products": [{ "product\_id": 3, "quantity": 2 }], "total\_value": 30000, "processed": true },

{ "\_id": 3, "customer\_id": 3, "products": [{ "product\_id": 4, "quantity": 1 }], "total\_value": 10000, "processed": false },

{ "\_id": 4, "customer\_id": 4, "products": [{ "product\_id": 5, "quantity": 4 }], "total\_value": 20000, "processed": false },

{ "\_id": 5, "customer\_id": 1, "products": [{ "product\_id": 2, "quantity": 3 }], "total\_value": 60000, "processed": true }

]);

db.invoices.insertMany([

{ "\_id": 1, "order\_id": 1, "date": "2024-11-25", "amount": 70000 },

{ "\_id": 2, "order\_id": 2, "date": "2024-11-24", "amount": 30000 },

{ "\_id": 3, "order\_id": 5, "date": "2024-11-23", "amount": 60000 }

]);

Query: List All Products in the Inventory

db.products.find({}, { \_id: 0, name: 1, price: 1, stock: 1 });

Query:List the Details of Orders with a Value > 20000

db.orders.find({ total\_value: { $gt: 20000 } });

Query: List All Orders Which Have Not Been Processed

db.orders.find({ processed: false });

Query: List All Orders Along with Their Invoice for “Mr. Rajiv”

db.orders.aggregate([

{ $match: { customer\_id: 1 } }, // Match orders for Mr. Rajiv

{

$lookup: {

from: "invoices", // Join with the invoices collection

localField: "\_id", // Order ID in orders

foreignField: "order\_id",// Order ID in invoices

as: "invoice\_details" // Resulting field for invoices

}

}

]);

Slip 12

1) <!DOCTYPE html>

<html>

<head>

<title>Customer Registration Form</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 20px;

background-color: #f4f4f4;

}

.form-container {

width: 100%;

max-width: 500px;

margin: auto;

background: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.form-container h2 {

text-align: center;

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 5px;

font-weight: bold;

}

input[type="text"], input[type="tel"], textarea, select {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

}

input[type="radio"], input[type="checkbox"] {

margin-right: 10px;

} .form-buttons {

display: flex;

justify-content: space-between;

}

input[type="submit"], input[type="reset"] {

padding: 10px 20px;

border: none;

border-radius: 5px;

background-color: #4CAF50;

color: white;

cursor: pointer;

}

input[type="submit"]:hover, input[type="reset"]:hover {

background-color: #45a049;

}

input[type="reset"] {

background-color: #f44336;

}

</style></head><body>

<div class="form-container">

<h2>Customer Registration Form</h2>

<form action="/submit" method="POST">

<label for="name">Name:</label>

<input type="text" id="name" name="name" placeholder="Enter your full name" required>

<label for="contact">Contact Number:</label>

<input type="tel" id="contact" name="contact" placeholder="Enter your contact number" required>

<label>Gender:</label>

<input type="radio" id="male" name="gender" value="Male" required>

<label for="male" style="display: inline;">Male</label>

<input type="radio" id="female" name="gender" value="Female">

<label for="female" style="display: inline;">Female</label>

<input type="radio" id="other" name="gender" value="Other">

<label for="other" style="display: inline;">Other</label>

<label>Preferred Days of Purchasing:</label>

<input type="checkbox" id="monday" name="days[]" value="Monday">

<label for="monday" style="display: inline;">Monday</label>

<input type="checkbox" id="tuesday" name="days[]" value="Tuesday">

<label for="tuesday" style="display: inline;">Tuesday</label>

<input type="checkbox" id="weekend" name="days[]" value="Weekend">

<label for="weekend" style="display: inline;">Weekend</label>

<label for="favorite">Favorite Item:</label>

<select id="favorite" name="favorite" required>

<option value="" disabled selected>Select your favorite item</option>

<option value="Groceries">Groceries</option>

<option value="Electronics">Electronics</option>

<option value="Clothing">Clothing</option>

<option value="Home Essentials">Home Essentials</option>

<option value="Toys">Toys</option>

</select>

<label for="suggestions">Suggestions:</label>

<textarea id="suggestions" name="suggestions" rows="4" placeholder="Share your suggestions here..."></textarea>

<div class="form-buttons">

<input type="submit" value="Submit">

<input type="reset" value="Reset">

</div> </form>

</div></body></html>

2) db.movies.insertMany([

{

"\_id": 1,

"title": "Movie A",

"budget": 10000000,

"actors": [{ "actor\_id": 1, "role": "Hero" }, { "actor\_id": 2, "role": "Villain" }],

"producers": [1, 2]

},

{

"\_id": 2,

"title": "Movie B",

"budget": 20000000,

"actors": [{ "actor\_id": 3, "role": "Hero" }, { "actor\_id": 1, "role": "Support" }],

"producers": [2]

},

{

"\_id": 3,

"title": "Movie C",

"budget": 15000000,

"actors": [{ "actor\_id": 2, "role": "Hero" }, { "actor\_id": 4, "role": "Heroine" }],

"producers": [3]

},

{

"\_id": 4,

"title": "Movie D",

"budget": 30000000,

"actors": [{ "actor\_id": 5, "role": "Hero" }, { "actor\_id": 1, "role": "Villain" }],

"producers": [3, 4]

},

{

"\_id": 5,

"title": "Movie E",

"budget": 5000000,

"actors": [{ "actor\_id": 1, "role": "Hero" }],

"producers": [1]

}

]);

db.actors.insertMany([

{ "\_id": 1, "name": "Akshay", "movies": [1, 2, 4, 5] },

{ "\_id": 2, "name": "Salman", "movies": [1, 3] },

{ "\_id": 3, "name": "Aamir", "movies": [2] },

{ "\_id": 4, "name": "Katrina", "movies": [3] },

{ "\_id": 5, "name": "Hrithik", "movies": [4] }

]);

db.producers.insertMany([

{ "\_id": 1, "name": "Producer A", "movies": [1, 5], "years\_active": [2023, 2024] },

{ "\_id": 2, "name": "Producer B", "movies": [1, 2], "years\_active": [2023, 2023] },

{ "\_id": 3, "name": "Producer C", "movies": [3, 4], "years\_active": [2024, 2024] },

{ "\_id": 4, "name": "Producer D", "movies": [4], "years\_active": [2024] },

{ "\_id": 5, "name": "Producer E", "movies": [], "years\_active": [] }

]);

1. List the Names of Movies with the Highest Budget

db.movies.find({ budget: { $eq: db.movies.find().sort({ budget: -1 }).limit(1)[0].budget } }, { \_id: 0, title: 1 });

2. Display the Details of Producers Who Have Produced More Than One Movie in a Year

db.producers.find({ $expr: { $gt: [ { $size: "$movies" }, 1 ] } });

3. List the Names of Actors Who Have Acted in At Least One Movie in Which "Akshay" Has Acted

db.actors.find({

movies: { $in: db.actors.findOne({ name: "Akshay" }).movies }

}, { \_id: 0, name: 1 });

4. List the Names of Movies Produced by More Than One Producer

db.movies.find({ "producers.1": { $exists: true } }, { \_id: 0, title: 1 });

Slip 13

1) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sample Web Page</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f9f9f9;

}

header {

background-color: #333;

color: #fff;

padding: 15px 20px;

text-align: center;

}

nav {

background-color: #555;

color: #fff;

padding: 10px;

text-align: center;

}

nav a {

color: #fff;

text-decoration: none;

margin: 0 15px;

}

nav a:hover {

text-decoration: underline;

}

main {

display: flex;

margin: 20px;

}

aside {

width: 25%;

background-color: #e4e4e4;

padding: 15px;

margin-right: 20px;

border-radius: 5px;

}

section {

width: 75%;

background-color: #fff;

padding: 15px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

footer {

background-color: #333;

color: #fff;

text-align: center;

padding: 10px 20px;

position: relative;

bottom: 0;

width: 100%;

}

</style>

</head>

<body>

<header>

<h1>Welcome to My Website</h1>

<p>Your one-stop destination for news, updates, and resources.</p>

</header>

<nav>

<a href="#home">Home</a>

<a href="#about">About</a>

<a href="#services">Services</a>

<a href="#contact">Contact</a>

</nav>

<main>

<aside>

<h3>Quick Links</h3>

<ul>

<li><a href="#news">Latest News</a></li>

<li><a href="#blog">Blog Posts</a></li>

<li><a href="#resources">Resources</a></li>

</ul>

<h3>Contact Us</h3>

<p>Email: support@example.com</p>

<p>Phone: +123-456-7890</p>

</aside>

<section>

<h2>About Us</h2>

<p>We are a team of dedicated professionals providing the latest news, insightful articles, and valuable resources to our users.</p>

<h2>Services</h2>

<ul>

<li>News and Updates</li>

<li>Blog Articles</li>

<li>Community Forums</li>

<li>Exclusive Resources</li>

</ul>

<h2>Latest News</h2>

<p>Stay updated with the latest happenings around the world. Visit our <a href="#news">news section</a> for more information.</p>

</section>

</main>

<footer>

<p>&copy; 2024 My Website. All Rights Reserved.</p>

<p>Follow us on

<a href="https://facebook.com" target="\_blank">Facebook</a>,

<a href="https://twitter.com" target="\_blank">Twitter</a>, and

<a href="https://instagram.com" target="\_blank">Instagram</a>.

</p>

</footer></body>

</html>

2) db.students.insertMany([

{ "\_id": 1, "name": "Alice", "class": "FY", "competitions": [1, 3] },

{ "\_id": 2, "name": "Bob", "class": "FY", "competitions": [2, 3] },

{ "\_id": 3, "name": "Charlie", "class": "SY", "competitions": [1, 2] },

{ "\_id": 4, "name": "David", "class": "TY", "competitions": [1] },

{ "\_id": 5, "name": "Eva", "class": "FY", "competitions": [3, 4] },

{ "\_id": 6, "name": "Frank", "class": "SY", "competitions": [4] },

{ "\_id": 7, "name": "Grace", "class": "FY", "competitions": [1, 2, 4] },

{ "\_id": 8, "name": "Hannah", "class": "SY", "competitions": [3, 4] },

{ "\_id": 9, "name": "Ian", "class": "TY", "competitions": [2, 3] },

{ "\_id": 10, "name": "Jack", "class": "FY", "competitions": [1] }

]);

db.competitions.insertMany([

{ "\_id": 1, "name": "Programming", "participants": [1, 3, 4, 7, 10], "winners": [7, 3, 1] },

{ "\_id": 2, "name": "Quiz", "participants": [2, 3, 7, 9], "winners": [3, 9, 7] },

{ "\_id": 3, "name": "E-Rangoli", "participants": [1, 2, 5, 8], "winners": [8, 5, 1] },

{ "\_id": 4, "name": "Debate", "participants": [5, 6, 7, 8], "winners": [6, 8, 5] },

{ "\_id": 5, "name": "Coding Hackathon", "participants": [3, 4, 6, 9], "winners": [9, 6, 4] }

]);

1 Display the Average Number of Students Participating in Each Competition

db.competitions.aggregate([

{

$project: {

name: 1,

avgParticipants: { $avg: { $size: "$participants" } }

}

}

]);

2 Find the Number of Students Participating in the Programming Competition

db.competitions.findOne({ name: "Programming" }, { participants: 1, \_id: 0 });

Count the participants:

db.competitions.aggregate([

{ $match: { name: "Programming" } },

{ $project: { numParticipants: { $size: "$participants" } } }

]);

3 Display the Names of the First Three Winners of Each Competition

db.competitions.aggregate([

{

$project: {

name: 1,

winners: 1,

firstThreeWinners: { $slice: ["$winners", 3] }

}

}

]);

4 Display Students from Class 'FY' Who Participated in 'E-Rangoli'

db.students.find({

class: "FY",

competitions: { $in: [3] }

}, { name: 1, \_id: 0 });

Slip 14

1) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Travel Plan Booking Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.container {

width: 50%;

margin: 50px auto;

background-color: #fff;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h2 {

text-align: center;

color: #333;

}

label {

font-size: 16px;

margin-bottom: 10px;

display: block;

}

input, select, textarea {

width: 100%;

padding: 8px;

margin-bottom: 20px;

border: 1px solid #ccc;

border-radius: 5px;

}

.checkbox-group {

display: flex;

gap: 15px;

}

.buttons {

display: flex;

justify-content: space-between;

}

button {

padding: 10px 15px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

button[type="reset"] {

background-color: #f44336;

}

button:hover {

opacity: 0.8;

}

</style>

</head>

<body>

<div class="container">

<h2>Travel Plan Booking Form</h2>

<form action="#" method="post">

<label for="name">Full Name:</label>

<input type="text" id="name" name="name" required placeholder="Enter your full name">

<label for="address">Address:</label>

<textarea id="address" name="address" rows="4" required placeholder="Enter your address"></textarea>

<label for="contact">Contact Number:</label>

<input type="tel" id="contact" name="contact" required placeholder="Enter your contact number">

<label>Gender:</label>

<input type="radio" id="male" name="gender" value="Male" required>

<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="Female" required>

<label for="female">Female</label>

<input type="radio" id="other" name="gender" value="Other" required>

<label for="other">Other</label>

<label>Preferred Season (select all that apply):</label>

<div class="checkbox-group">

<input type="checkbox" id="winter" name="season" value="Winter">

<label for="winter">Winter</label>

<input type="checkbox" id="summer" name="season" value="Summer">

<label for="summer">Summer</label>

<input type="checkbox" id="spring" name="season" value="Spring">

<label for="spring">Spring</label>

<input type="checkbox" id="autumn" name="season" value="Autumn">

<label for="autumn">Autumn</label>

</div>

<label for="location">Preferred Location Type:</label>

<select id="location" name="location" required>

<option value="">--Select Location Type--</option>

<option value="Beach">Beach</option>

<option value="Mountain">Mountain</option>

<option value="City">City</option>

<option value="Countryside">Countryside</option>

</select>

<div class="buttons">

<button type="submit">Submit</button>

<button type="reset">Reset</button>

</div> </form>

</div></body></html>