Mini Project Synopsis

Mini Project Title: **Insurance Management System**

Group member details:

1. Kedar Bhikaji Mane

Domain of The Project:

The domain of an insurance management system project is the insurance industry. The insurance industry is a vast and complex industry that provides financial protection against loss, damage, and death. Insurance companies offer a wide range of products and services, including life insurance, health insurance, property insurance, and liability insurance.

The insurance management system project will be within the insurance industry domain. This means that the project will focus on developing a system that can be used by insurance companies to manage their policies, customers, and claims more efficiently. The system will need to be able to handle a wide range of tasks, including:

- ✓ **Policy management:** This includes tasks such as creating and editing policies, tracking policy status, and managing premium payments.
- ✓ **Customer management:** This includes tasks such as managing customer contact information, policy information, and claims history.
- ✓ **Claims management:** This includes tasks such as tracking and managing claims from start to finish, generating reports on claims data, and processing claims payments.

The insurance management system project will be a complex undertaking, but it has the potential to be a valuable tool for insurance companies. By automating many manual tasks and providing improved insights, the system can help insurance companies to improve their efficiency, customer service, and bottom line.

Objectives of Project :

Improve efficiency and productivity:

An IMS can automate many manual tasks, such as policy administration, claims processing, and customer relationship management. This can free up staff to focus on more strategic initiatives and provide better customer service.

Reduce cost:

An IMS can help to reduce costs by automating tasks, eliminating manual errors, and improving efficiency.

Improve customer service:

An IMS can provide customers with self-service options and make it easier for them to file and track claims. This can lead to increased customer satisfaction and loyalty.

Gain better insights:

An IMS can generate reports on policy data, claims data, and other metrics. This information can help insurance companies to make better decisions about their business, such as pricing policies, developing new products, and improving risk management.

Introduction:

The Insurance Management System (IMS) is a modern and comprehensive platform tailored to the specific needs of insurance companies. It utilizes the power of MongoDB, a highly flexible NoSQL database, to efficiently store, manage, and retrieve data related to insurance policies, policyholders, premium payments, and claims. IMS simplifies the complexity of insurance operations, offers scalability for handling growing data volumes, and ensures secure and compliant data management.

Software and Hardware Platform Requirement:

Operating System: Server: Linux (e.g., Ubuntu,Fedora), Windows Server.

Database Management System: MongoDB for NoSQL database requirements(MongoDB Compass)

Programming Languages: Python , MongoDB

Libraries: pymongo

Implementation and code:

```
import pymongo

client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["insurance_system"]
collection = db["policies"]

def add_policy(policy_id, policy_holder, policy_type, premium_amount):
    policy = {
        "policy_id": policy_id,
        "policy_holder": policy_holder,
        "policy_type": policy_type,
        "premium_amount": premium_amount
    }
    collection.insert_one(policy)
    print("Policy added successfully!")

def get_policies_by_holder(policy_holder):
    policies = collection.find({"policy_holder": policy_holder})
    for policy in policies:
        print(f"Policy ID: {policy['policy_id']}, Type: {policy['policy_type']},
Premium Amount: {policy['premium_amount']}")
```

```
def list_all_policies():
    policies = collection.find()
    for policy in policies:
        print(f"Policy ID: {policy['policy_id']}, Holder:
{policy['policy_holder']}, Type: {policy['policy_type']}, Premium Amount:
{policy['premium_amount']}")

def delete_policy(policy_id):
    result = collection.delete_one({"policy_id": policy_id})
    if result.deleted_count == 1:
        print(f"Policy with ID {policy_id} has been deleted successfully.")
    else:
        print(f"Policy with ID {policy_id} not found.")

while True:
    print("\nInsurance Management System")
    print("1. Add Policy")
    print("2. Get Policies by Holder")
    print("3. List All Policies")
    print("4.Delete the policy")
    print("5. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        policy_id = input("Enter Policy ID: ")
        policy_holder = input("Enter Policy Holder: ")
        policy_type = input("Enter Policy Type: ")
        premium_amount = float(input("Enter Premium Amount: "))
        add_policy(policy_id, policy_holder, policy_type, premium_amount)

    elif choice == "2":
        policy_holder = input("Enter Policy Holder: ")
        get_policies_by_holder(policy_holder)

    elif choice == "3":
        list_all_policies()

    elif choice == "4":
        policy_id = input("Enter Policy ID to delete: ")
        delete_policy(policy_id)

    elif choice == "5":
        print("Exiting the program.")
        break

    else
 print("Invalid choice. Please try again.")
```

Output:

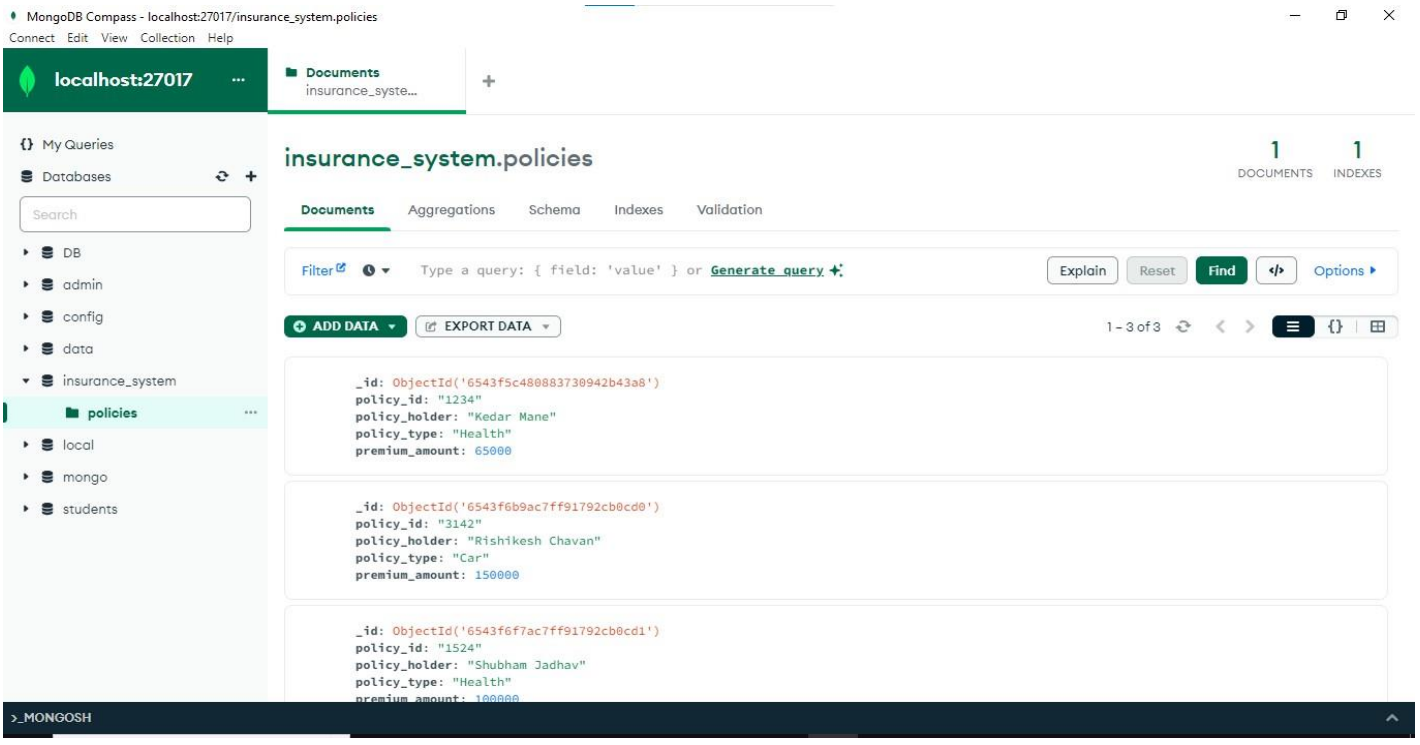
```
Insurance Management System
1. Add Policy
2. Get Policies by Holder
3. List All Policies
4. Delete the policy
5. Exit
Enter your choice: 1
Enter Policy ID: 1234
Enter Policy Holder: Kedar Mane
Enter Policy Type: Health
Enter Premium Amount: 65000.0
Policy added successfully!

Insurance Management System
1. Add Policy
2. Get Policies by Holder
3. List All Policies
4. Delete the policy
5. Exit
Enter your choice: 1
Enter Policy ID: 3142
Enter Policy Holder: Rishikesh Chavan
Enter Policy Type: Car
Enter Premium Amount: 150000
Policy added successfully!

Insurance Management System
1. Add Policy
2. Get Policies by Holder
3. List All Policies
4. Delete the policy
5. Exit
Enter your choice: 1
Enter Policy ID: 1524
Enter Policy Holder: Shubham Jadhav
Enter Policy Type: Health
Enter Premium Amount: 100000
Policy added successfully!

Insurance Management System
1. Add Policy
2. Get Policies by Holder
3. List All Policies
4. Delete the policy
5. Exit
Enter your choice: 3
Policy ID: 1234, Holder: Kedar Mane, Type: Health, Premium Amount: 65000.0
Policy ID: 3142, Holder: Rishikesh Chavan, Type: Car, Premium Amount: 150000.0
Policy ID: 1524, Holder: Shubham Jadhav, Type: Health, Premium Amount: 100000.0
```

Insurance Management System
1. Add Policy
2. Get Policies by Holder
3. List All Policies
4. Delete the policy
5. Exit
Enter your choice: 4
Enter Policy ID to delete: 1524
Policy with ID 1524 has been deleted successfully.

Insurance Management System
1. Add Policy
2. Get Policies by Holder
3. List All Policies
4. Delete the policy
5. Exit
Enter your choice: 5
Exiting the program.

Screenshots:

The Below image is of Database in MongoDB compass after Adding Data



The Below image is of Database in MongoDB compass after deleting Data.