INSURE

INSURANCE MANAGEMENT SYSTEM

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

SHONA C
Reg No: 22PMC154

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam (Autonomous)

Peermade, Kerala – 685 531

2023

INSURE

INSURANCE MANAGEMENT SYSTEM

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

SHONA C
Reg No: 22PMC154

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam (Autonomous)

Peermade, Kerala – 685 531

2023

A Project Report on

INSURE

INSURANCE MANAGEMENT SYSTEM

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

SHONA C
Reg No: 22PMC154

Under the guidance of
Sr. Italia Joseph Maria
Assistant Professor
PG Department of Computer Applications
Marian College Kuttikkanam (Autonomous)

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam (Autonomous)

Peermade, Kerala – 685 531

2023

PG DEPARTMENT OF COMPUTER APPLICATIONS

Marian College Kuttikanam (Autonomous)

[MAHATMA GANDHI UNIVERSITY, KOTTAYAM]
KUTTIKKANAM – 685 531, KERALA.

CERTIFICATE

This is to certify that the project work entitled

“INSURE”

is a bonafide record of work done by

SHONA C

Reg. No: -22PMC154

In partial fulfillment of the requirements for the award of Degree of

MASTER OF COMPUTER APPLICATIONS [MCA]

During the academic year 2022 - 2023.

Sr. Italia Joseph Maria

Assistant Professor

PG Department of Computer Applications
Marian College Kuttikkanam (Autonomous)

Mr. Win Mathew John

Head of the Department

PG Department of Computer Applications
Marian College Kuttikkanam (Autonomous)

External Examiner

External Examiner

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. I express my sincere gratitude to Prof Dr. Ajimon George, Principal, Marian College Kuttikkanam (Autonomous), Dr. Mendus Jacob, Director, PG Department of Computer Applications for the support given throughout the project work. I extend my gratitude to Mr. Win Mathew John, HoD, PG Department of Computer Applications, who is a constant source of inspiration and whose advice helped me to complete this project work successfully.

I express my deep sense of gratitude to my project guide, SR. ITALIA JOSEPH MARIA, Associate Professor/Assistant Professor, PG Department of Computer Applications, for her profound guidance for the successful completion of this project work.

With great enthusiasm, I express my gratitude to all the faculty members of the PG Department of Computer Applications for their timely help and support.

Finally, I express my deep appreciation to all my friends and family members for the moral support and encouragement they have given to complete this project work successfully.

SHONA C

ABSTRACT

The Insurance Management System is a comprehensive software solution designed to facilitate the efficient management of insurance policies, user interactions, and administrative tasks. The system aims to simplify the insurance process by providing a user-friendly platform for policy selection, application submission, query handling, and policy history tracking. It empowers both administrators and users by offering an intuitive interface and robust functionality.

The system incorporates various features and functionalities to achieve these objectives. The system provides user registration and authentication mechanisms, allowing users to create accounts and log in securely. This ensures that only authorized individuals can access policy information, submit applications, and ask queries. Administrators have access to a comprehensive dashboard where they can manage insurance categories, create new policies. Once a user selects a policy, they can submit an application for that policy. The system facilitates the submission process and forwards the application to the administrator for review. The administrator can then approve or disapprove the application based on predefined criteria. This workflow ensures proper evaluation and control over policy applications.

By implementing the insurance management system, several benefits can be achieved. The Insurance Management System offers a centralized platform for efficient policy management, user engagement, and streamlined communication between administrators and users. By automating various insurance processes, the system simplifies policy selection, application submission, query handling, and policy history tracking, leading to enhanced user experience and operational effectiveness in the insurance domain.

OBJECTIVE AND SCOPE

- The objective and scope of a Django project for a insurance management system is to effectively and efficiently handle the policy selection, management of insurance policies, policy applications, user interactions, and query handling.
- Develop an efficient and user-friendly web application for managing insurance related operations.
- The system aims to provide users with access to their policy history.
- Implement user authentication mechanisms to ensure secure access to the application.

PROBLEM STATEMENT

In the insurance industry, there is a need for an efficient and streamlined system to manage insurance policies, policy applications, user interactions, and query handling. The current manual processes and lack of a centralized system result in inefficiencies, delays, and a suboptimal user experience. The existing insurance management process lacks a comprehensive and automated system to effectively manage insurance categories, policy administration, user interactions, and query handling. The absence of a centralized platform leads to challenges such as disorganized policy categorization, lengthy application approval processes, limited access to policy history, and ineffective query management. These issues contribute to delays, errors, and a lack of transparency, impacting both administrators and users.

TABLE OF CONTENTS

TABLE INDEX

1. INTRODUCTION

1.1 PROBLEM STATEMENTS

In the insurance industry, there is a need for an efficient and streamlined system to manage insurance policies, policy applications, user interactions, and query handling. The current manual processes and lack of a centralized system result in inefficiencies, delays, and a suboptimal user experience.

1.2 PROPOSED SYSTEM

The proposed Insurance Management System aims to address the challenges and requirements of managing insurance categories, policy administration, user interactions, and query handling. The system will provide a centralized platform that streamlines processes, improves efficiency, and enhances user experience. The system will have an intuitive and user-friendly interface that allows easy navigation, policy selection. Users will have access to their policy history, providing a comprehensive overview of their past policies.

1.3 FEATURES OF THE PROPOSED SYSTEM

The features of this website are:

- Responsive website design.
- User-Friendly navigation.
- Policy Management by Admin.
- Policy Selection by Users.
- Query Submission and Management.

2. FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS

The functional requirements for this website include:

- ***Login and Signup***: The user should be able to register themselves using Django users. The user will be redirected to the user home page after logging in. The admin adds the policy using the admin interface. Both admin and user will be redirected to their respective pages after logging in.
- ***Insurance Category Management***: Admins should be able to create new insurance categories, edit and update existing insurance categories. Admins should be able to remove unnecessary or outdated insurance categories.
- ***Policy Management***: Admins should be able to add new insurance policies to the system, including details such as name, coverage, premiums. Admins should be able to edit and update existing policies
- ***Policy Selection***: Users should be able to select and choose policies
- ***Policy History***: Users should be able to view their policy history, including details of past policies availed.
- ***Query Management***: Users should be able to submit queries. Admins should be able to review incoming queries. Admins should be able to respond to user queries.

3. NON-FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements for this website are:

- ***Usability***: The proposed website is simple, provides enough insight about features and packages, interactive, lets user select packages and schedule pick-ups and all this data is stored in the database.
- ***Reliability***: The system must perform without failure in 95 percent of use cases during a month.
- ***Maintainability***: The mean time to restore the system (MTTRS) following a system failure must not be greater than 10 minutes. MTTRS includes all corrective maintenance timeand delay time.
- ***Availability***: Describes how likely the system is accessible to a user at a given point in time. A user-friendly system with global accessibility should be available around-the clock. In the event that the database is corrupted or the hardware fails, a replacement page will appear. Additionally, a database backup should be kept in case of hardware failure or database corruption.
- ***Security***: Database should be backed up every hour. Under failure, system should be able to come back at normal operation under an hour. All data must be stored, protected, or protectively marked.

4.FEATURES AND HIGHLIGHTS

FEATURES AND HIGHLIGHTS

The features and highlights of this project are:

- ***Login and registration***: The user will be able to register themselves using Django users. The user will be redirected to the user home page after logging in. Both admin and employee will be redirected to their respective pages after logging in from the interface.
- ***Insurance Category Management:*** Administrators can manage insurance categories, including adding new categories, updating existing ones, and deleting categories as needed.
- ***Policy Management:*** Administrators have the ability to add new policies to the system. Policies can be added with details such as policy name, coverage, premium amount, terms, and conditions.
- ***Policy Selection:*** Users can select policies.
- ***Policy Users Application and Approval Process:*** Can apply for policies through the system. Each policy application requires action from the administrator for approval or disapproval.
- ***Policy History***: Users have access to their policy history, which includes details of previously purchased policies.
- ***Query Handling:*** Users can ask queries related to policies. Administrators can manage and respond to user queries.
- ***User-Friendly Interface:*** The system provides a user-friendly interface for both administrators and users.

5. THIRD PARTY LIBRARIES

THIRD-PARTY LIBRARIES

Third-party applications and libraries in Django are pre-built components or packages developed by the community or other companies that you can use to extend the functionality of your Django projects. These libraries provide pre-built solutions for common tasks, saving developers time and effort in implementing certain features from scratch. They are designed to seamlessly integrate with Django and follow its best practices.

Third-party libraries can be installed using package managers like pip, and they usually come with their own documentation and examples to guide developers in their usage. These libraries can cover a wide range of functionalities

The third-party libraries used in this project are:

- ***Django widget tweaks:*** Django Widget Tweaks is a third-party library for Django that provides additional functionalities and utilities to work with form widgets and form rendering. It offers a convenient way to customize and manipulate the appearance and behavior of form fields in Django templates.
- ***Django jazzmin:*** Django Jazzmin is a third-party library for Django that provides an improved admin interface. It is a modern, responsive, and customizable replacement for Django's default admin interface, a drop-in app to jazz up your Django admin site, with plenty of things you can easily customize, including a built-in UI customizer.

6.DATABASE DESIGN

DATABASE DESIGN

The general theme behind a database is to handle information as an integrated whole. It is a collection of interrelated data stored with minimum redundancy to serve many users quicklyand efficiently. The general objective is to make information access easy, quick, inexpensiveand flexible for the user. It is the most widely used relational database. It offers various features and provides users with many niceties. Computer databases can store data in differentforms from simple lines of text to complex data structure that includes pictures, sounds or video images. Data management involves creating, modifying, deleting and adding data in files and using this data to generate reports. The software that allows performing this functionis known as a database management system.

CUSTOMERS TABLE

```
class Customer(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    address = models.CharField(max_length=40)
    mobile = models.CharField(max_length=20,null=False)
```

CATEGORY TABLE

```
class Category(models.Model):
    category_name =models.CharField(max_length=20)
    creation_date =models.DateField(auto_now=True)
    category_details=models.CharField(max_length=200,null=True)
```

POLICY TABLE

```
class Policy(models.Model):
    category= models.ForeignKey('Category', on_delete=models.CASCADE)
    policy_name=models.CharField(max_length=200)
    sum_assurance=models.PositiveIntegerField()
    premium=models.PositiveIntegerField()
    tenure=models.PositiveIntegerField()
    creation_date =models.DateField(auto_now=True)
```

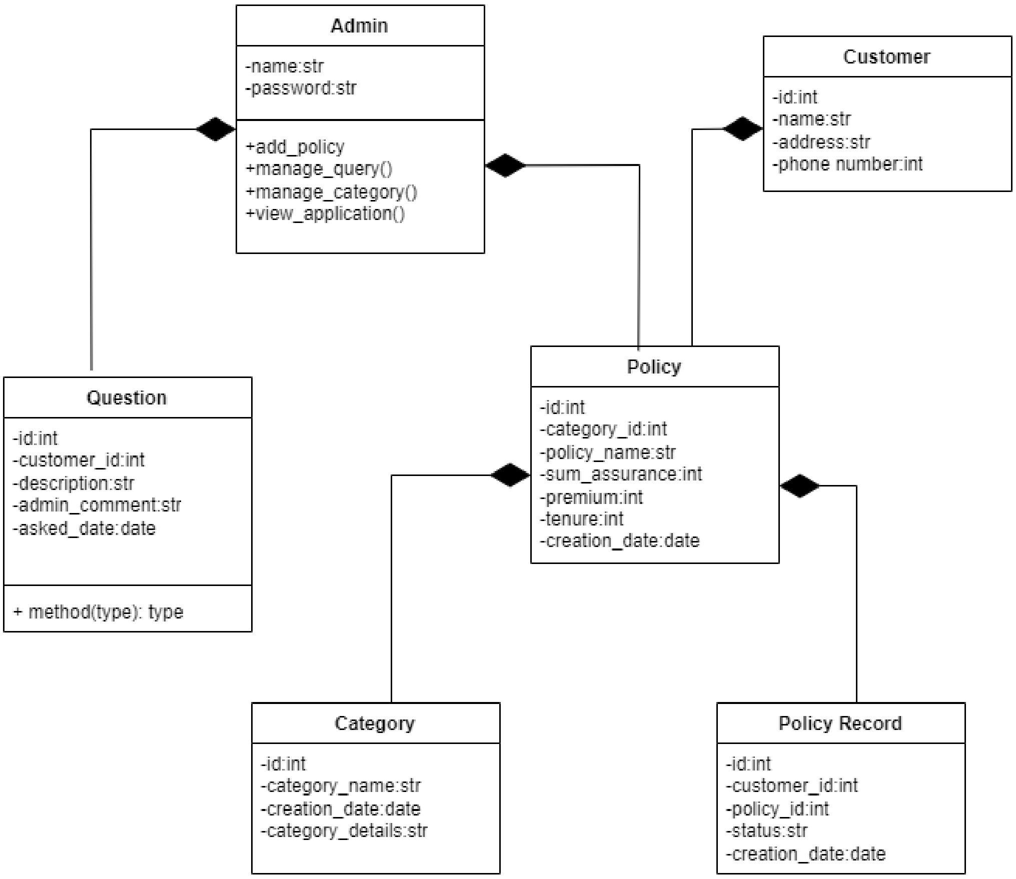
POLICY RECORD TABLE

```
class PolicyRecord(models.Model):
    customer= models.ForeignKey(Customer, on_delete=models.CASCADE)
    Policy= models.ForeignKey(Policy, on_delete=models.CASCADE)
    status = models.CharField(max_length=100,default='Pending')
    creation_date =models.DateField(auto_now=True)
```

QUESTION TABLE

```
class Question(models.Model):
    customer= models.ForeignKey(Customer, on_delete=models.CASCADE)
    description =models.CharField(max_length=500)
    admin_comment=models.CharField(max_length=200,default='Nothing')
    asked_date =models.DateField(auto_now=True)
```

6.1.CLASS DIAGRAM



7. CHALLENGES

CHALLENGES FACED

The challenges faced during the project:

The major challenge faced by me was to plan the process flow of the project. Take time to thoroughly analyze the requirements of the project and create a detailed plan or flowchart that outlines the different steps and components involved. Break down the project into smaller tasks and prioritize them based on dependencies. Refer to the Django documentation and tutorials specifically related to user authentication and registration. Understand the concepts of user models, forms, and views in Django.

8.FUTURE ENHANCEMENTS

FUTURE ENHANCEMENTS

- Policy Renewals and Reminders: Implement a feature that reminds users when their policies are due for renewal. Users can receive notifications and have the option to renew policies directly from the system. This feature can help improve customer retention and streamline the renewal process.
- Policy Document Management: Allow users to access and download policy documents, such as terms and conditions, coverage details, and policy certificates. Implement a secure document repository and provide a user-friendly interface for document management.
- Introduction of a mobile application can also be done so that this system becomes more user friendly.

9.CONCLUSION

CONCLUSION

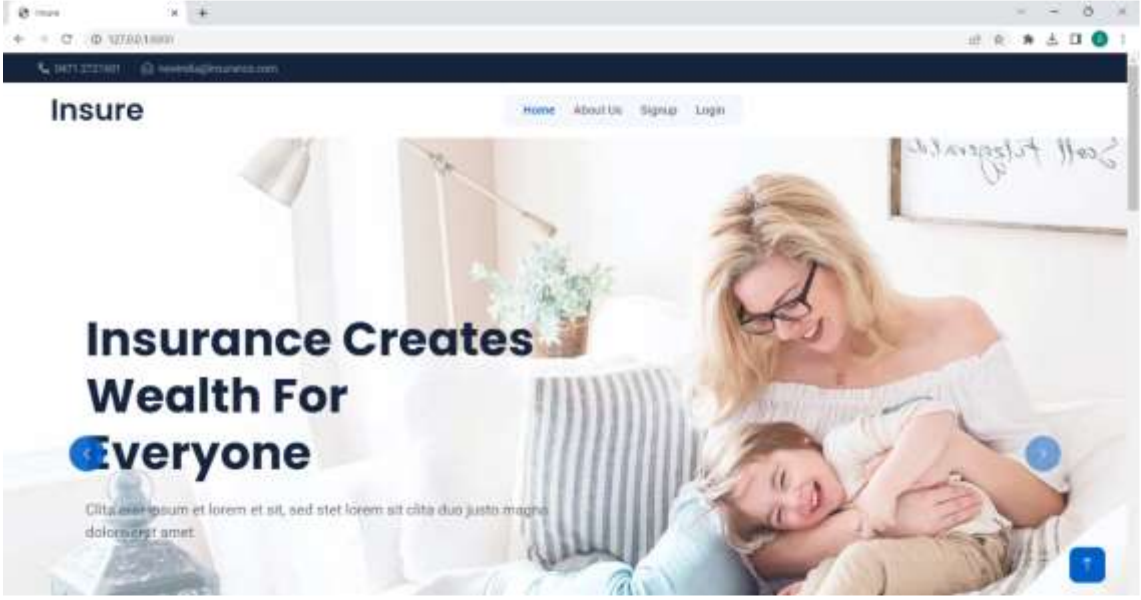
The insurance management system project demonstrated successful implementation and addressed the identified requirements. It showcased effectively managing insurance policies, user interactions, and administrative tasks. The project review highlights the accomplishments achieved, while also acknowledging the potential for future enhancements to further enhance the system's capabilities.

10.REFERENCES

REFERENCES

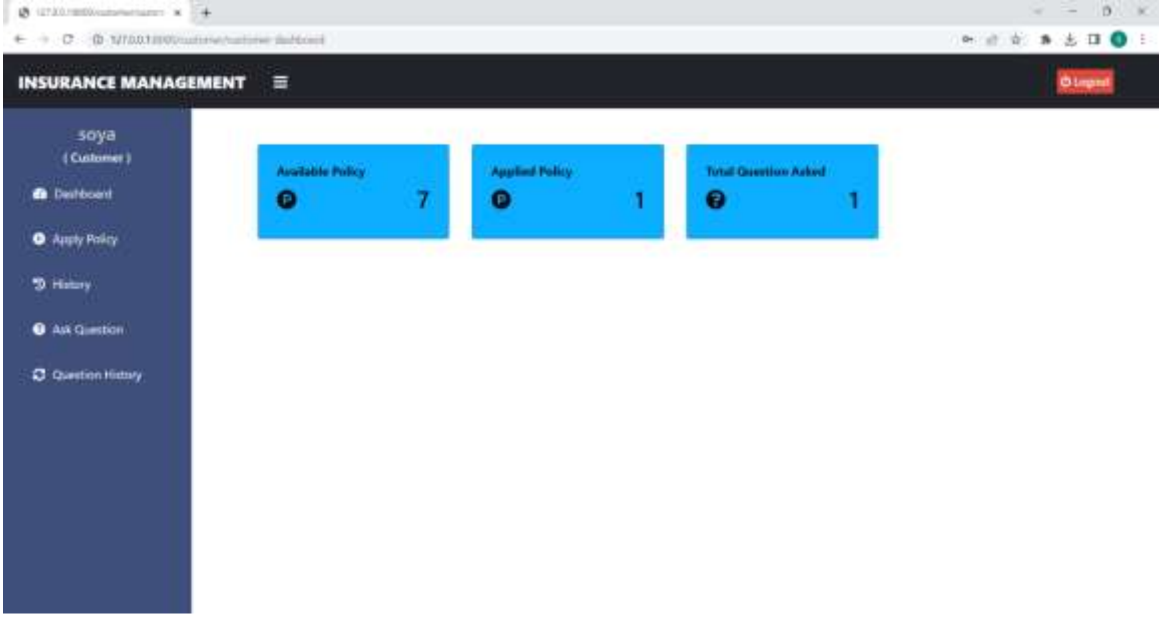
- https://docs.djangoproject.com/
- https://themewagon.github.io/insure/service.html
- https://www.bootstrapdash.com/product/royalui
- https://demos.wrappixel.com/free-admin-templates/bootstrap/ample-bootstrap-free/landingpage/index.html

11.ANNEXURE

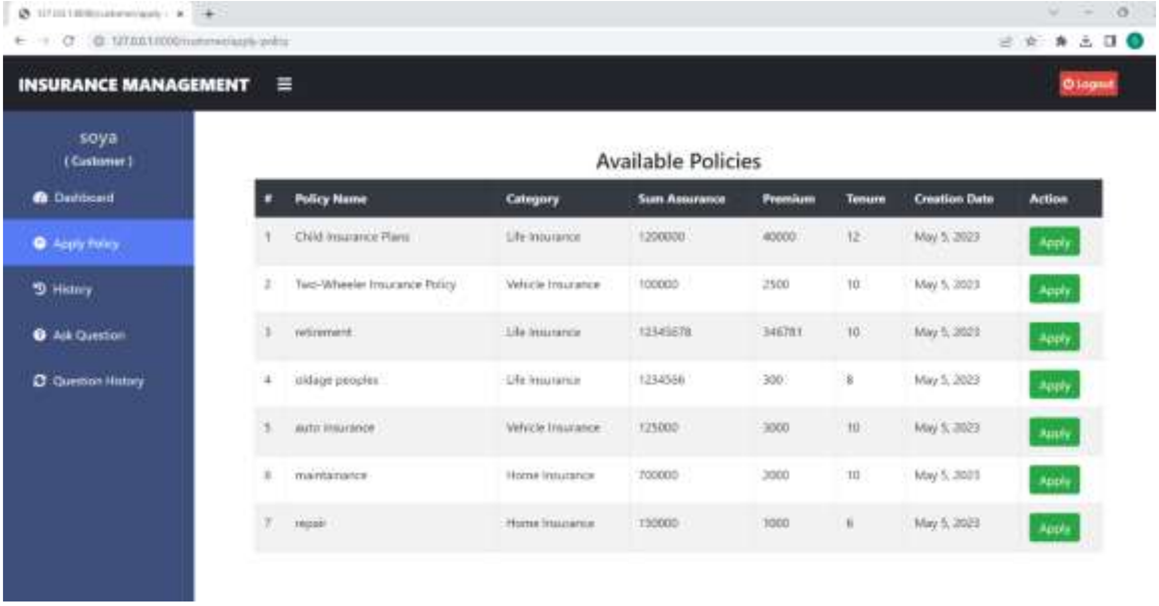


Insurance Creates Wealth For Everyone

Clita erat ipsum et lorem et sit, sed stet lorem sit clita duo justo magna dolore erat amet

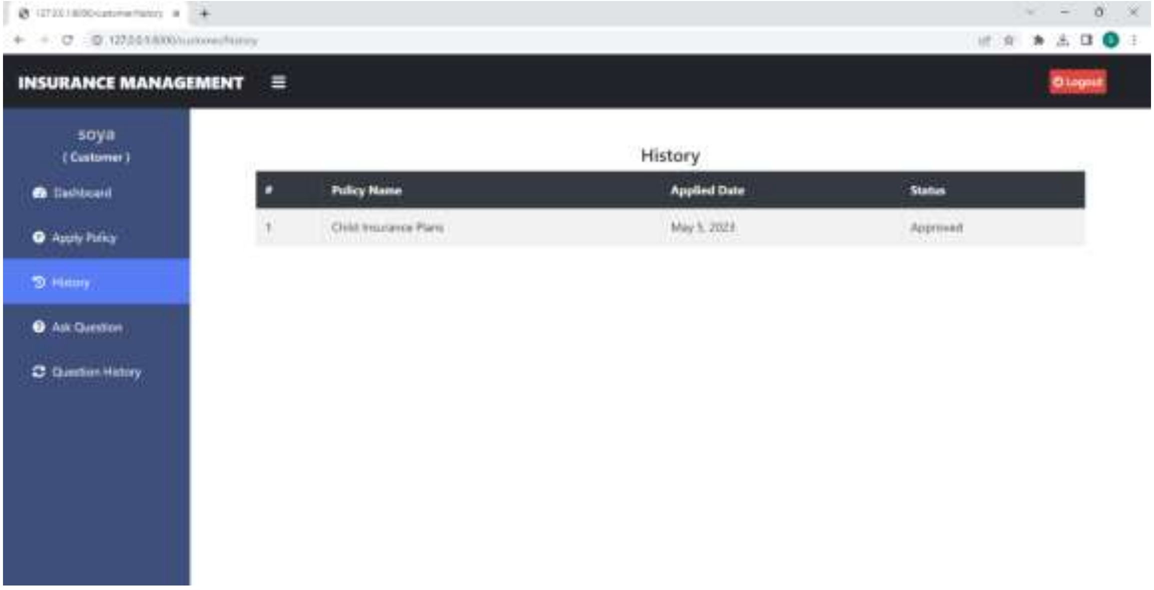


Available Policy	Applied Policy	Total Question Asked
7	1	1



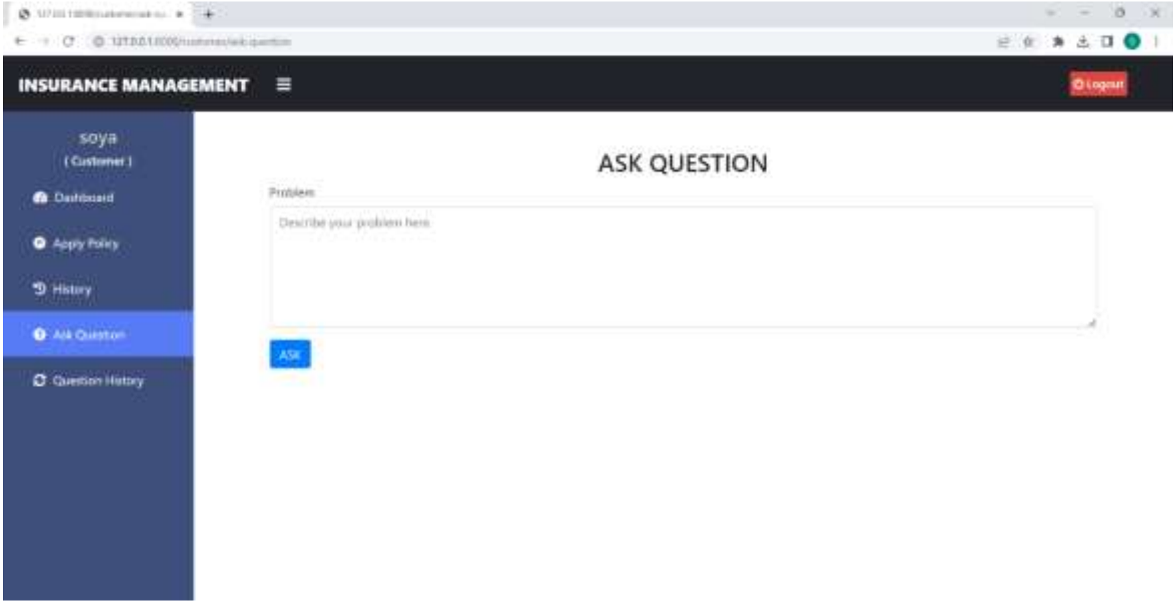
Available Policies

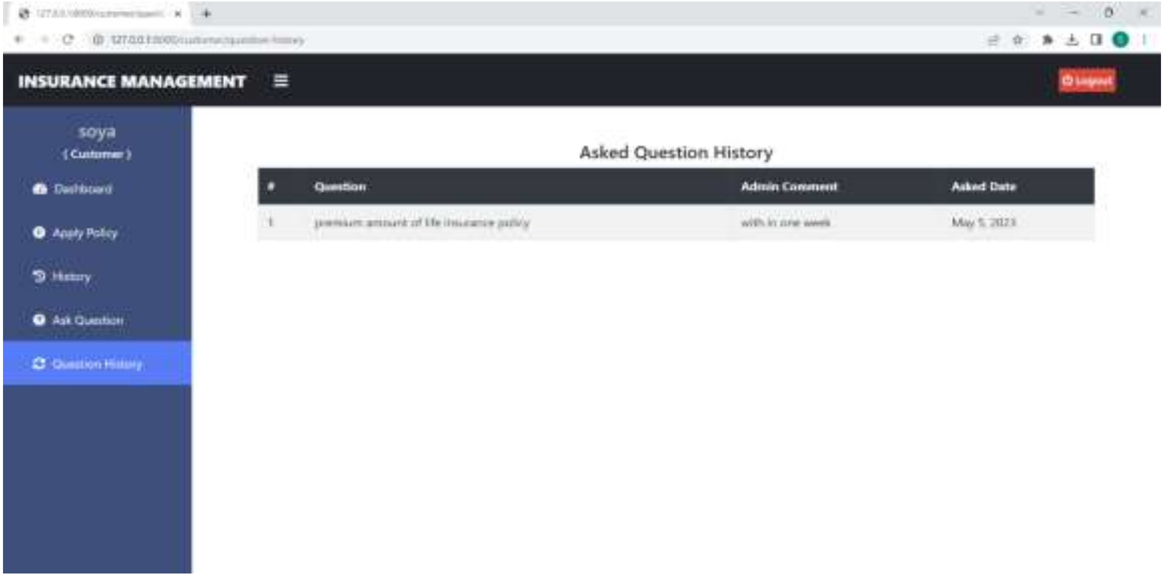
#	Policy Name	Category	Sum Assurance	Premium	Tenure	Creation Date	Action
1	Child Insurance Plans	Life Insurance	1200000	40000	12	May 5, 2023	Apply
2	Two-Wheeler Insurance Policy	Vehicle Insurance	100000	2500	10	May 5, 2023	Apply
3	retirement	Life Insurance	12345678	346781	10	May 5, 2023	Apply
4	oldage peoples	Life Insurance	1234566	300	8	May 5, 2023	Apply
5	auto insurance	Vehicle Insurance	125000	3000	10	May 5, 2023	Apply
6	maintanance	Home Insurance	700000	2000	10	May 5, 2023	Apply
7	repair	Home Insurance	130000	1000	6	May 5, 2023	Apply



History

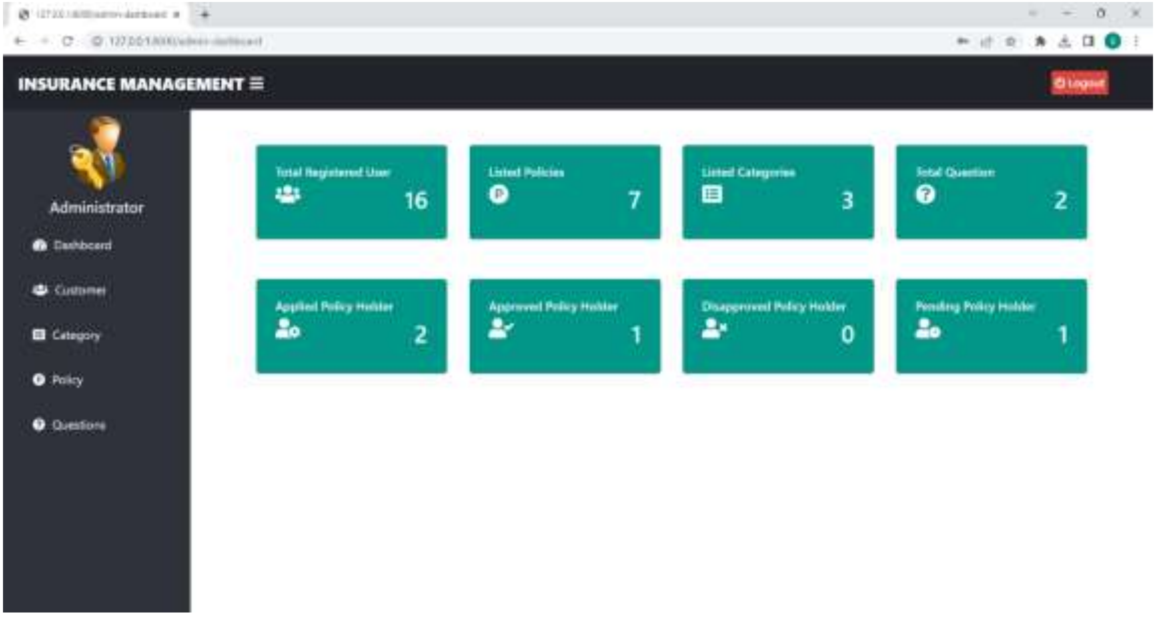
#	Policy Name	Applied Date	Status
1	Child Insurance Plans	May 5, 2023	Approved





Asked Question History

#	Question	Admin Comment	Asked Date
1	premium amount of life insurance policy	with in one week	May 5, 2023



Total Registered User 16

Listed Policies 7

Listed Categories 3

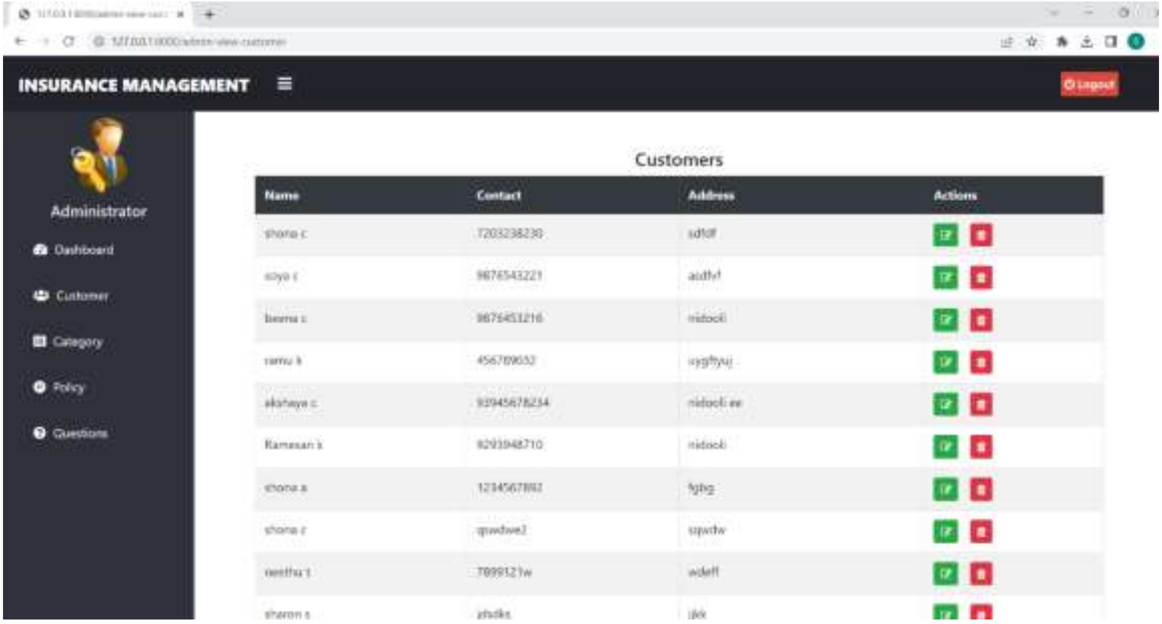
Total Question 2

Applied Policy Holder 2

Approved Policy Holder 1

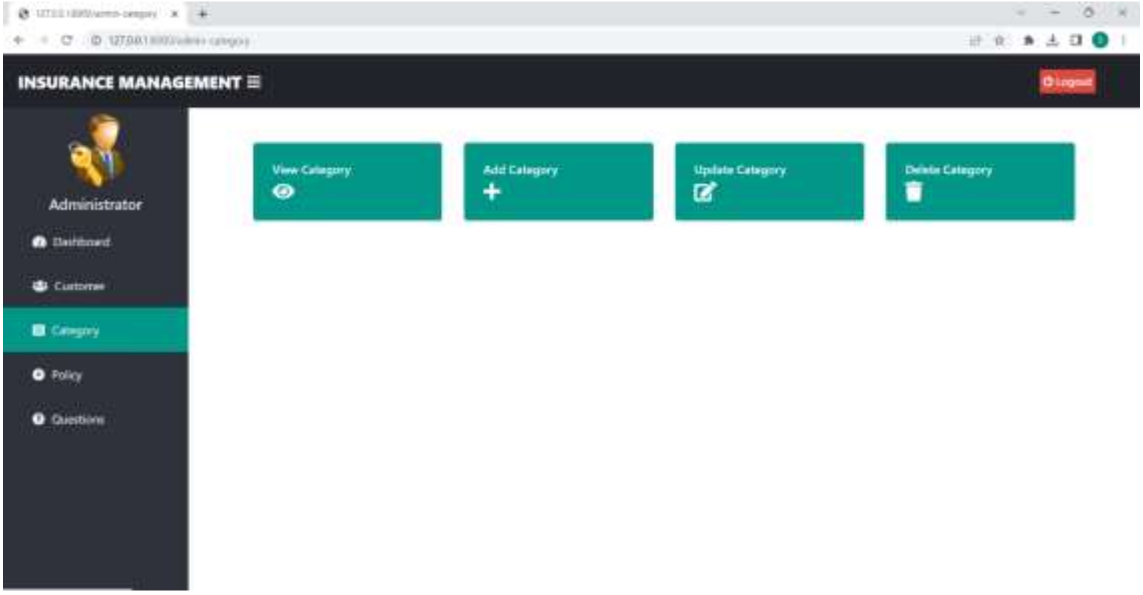
Disapproved Policy Holder 0

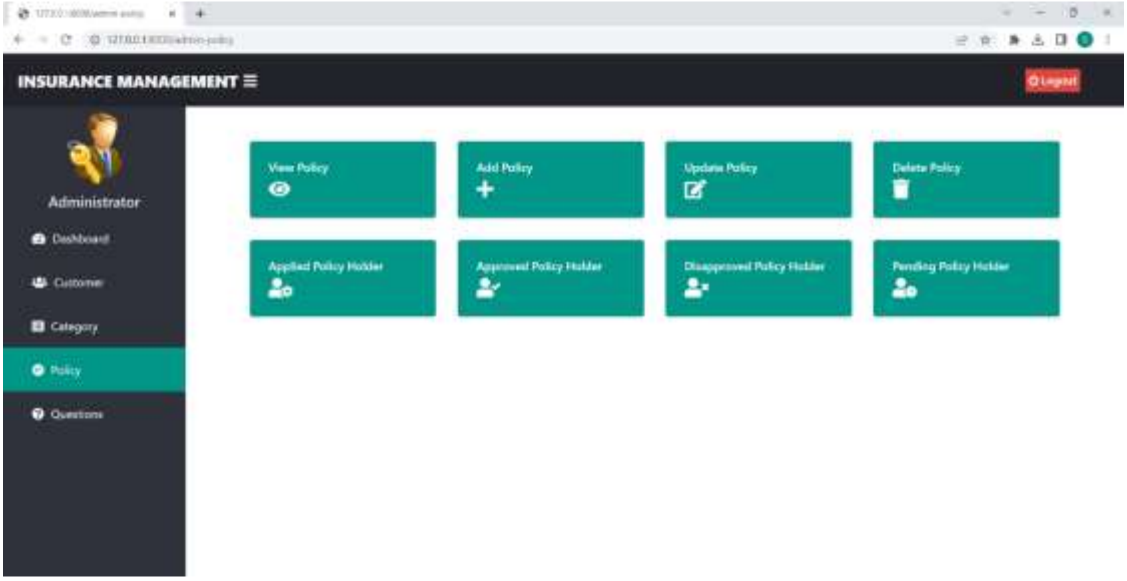
Pending Policy Holder 1

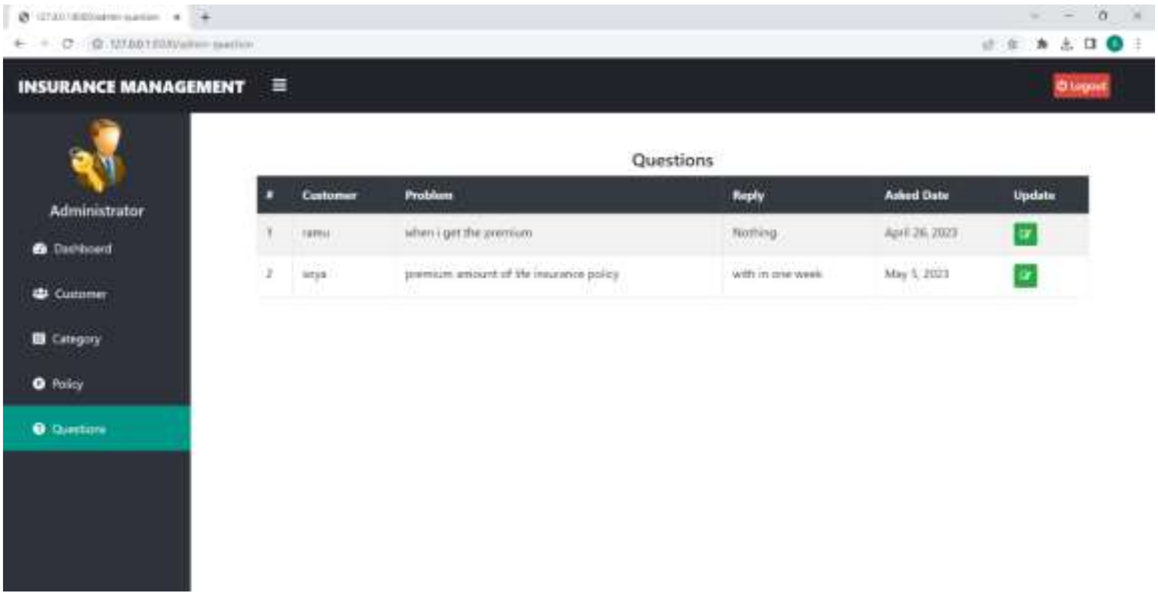


Customers

Name	Contact	Address	Actions
shona c	7203238230	sdfdf	🟩 🟥
soya c	9876543221	asdfvf	🟩 🟥
beena c	9876453216	nidooli	🟩 🟥
ramu k	456789652	uygftyuj	🟩 🟥
akshaya c	93945678234	nidooli ee	🟩 🟥
Ramesan k	8293948710	nidooli	🟩 🟥
shona a	1234567892	fghg	🟩 🟥
shona c	qwdwe2	sqwdw	🟩 🟥
neethu t	789912 1w	wdeff	🟩 🟥
sharon s	jdsdks	jjkk	🟩 🟥


127.0.0.1:8000/admin-category
INSURANCE MANAGEMENT ≡
Logout
Administrator
Dashboard
Customer
Category
Policy
Questions
View Category
Add Category
Update Category
Delete Category


127.0.0.1:8000/admin-policy
INSURANCE MANAGEMENT ≡
Logout
Administrator
Dashboard
Customer
Category
Policy
Questions
View Policy
Add Policy
Update Policy
Delete Policy
Applied Policy Holder
Approved Policy Holder
Disapproved Policy Holder
Pending Policy Holder



Questions

#	Customer	Problem	Reply	Asked Date	Update
1	ramu	when i get the premium	Nothing	April 26, 2023	✎
2	sriya	premium amount of life insurance policy	with in one week	May 5, 2023	✎