**A**

**Project Report**

**On**

**“Insurance Policy Management”**

This project thesis is submitted to fulfil the requirement for the completion of RLL Project



**Submitted By-**

Shaik Kaleem Arshad-2506335

Shubham Bapusaheb Bhavar-2507162

Shubham Kumar-2503975

Shubham Phulchand Gupta-2506420

SK Nooralam Rahaman-2503765

Srinivas MR-2503972

Subhadip Samanta-2504076

Summi Sinha-2504045

Surajkumar Ramkrushna Yergude-2504314

Surendra Kumar T-2505869

Mphasis: Applied Technology Services and Solution

**Table Of Content**

1. **Introduction**…………………………………………………………………….3
   1. Overview…………………………………………………………….….3
2. **System Overview**…………………………………………………………….3
   1. Authentication and Authorization……………………..…..3
   2. Functional Flow………………………………………………………4
   3. Environment…………………………………………………………..4
3. **Sub-System Details**………………………………………………………….5
   1. Administration……………………………………………………….5
   2. Customer……………………………………………………………….6
4. **Rest APIs used to Build**……………………………………………………6
5. **E-R Diagram**…………………………………………………………………...9
   1. Admin E-R Diagram……………………………………………….9
   2. User E-R Diagram………………………………………………...10
6. **Use-Case Diagram**………………………………………………….……..11
7. **Future Scope**…………………………………………………………………11
8. **Output Screenshot of Project**……………………………………12-18
9. **Testing with DevOps**…………………………………………….…..18-21
10. **Introduction**

Insurance Policy Management System is a web application which is used to tracking the details about the insurance policy, customer details and company details.

In this online process the user enter into the website it will show details about insurance and its types, also it will show the details about different duration schemes to the corresponding insurance type or insurance policy. In this process contains the user registration form which is used to apply for insurance policy through online. It also helps the customer to view their own insurance status information.

**1.1 Overview:**

The system is convenient and flexible to be used, it saves their time, efforts, money and resources .User Friendliness is provided in the application with various controls provided by system Rich user interface  Purchasing insurance online is convenient and fast. With click of a mouse customers can buy any policy from any corner of the world at any point of time.This system maintains profile management of all policy holders.

1. **System Overview**

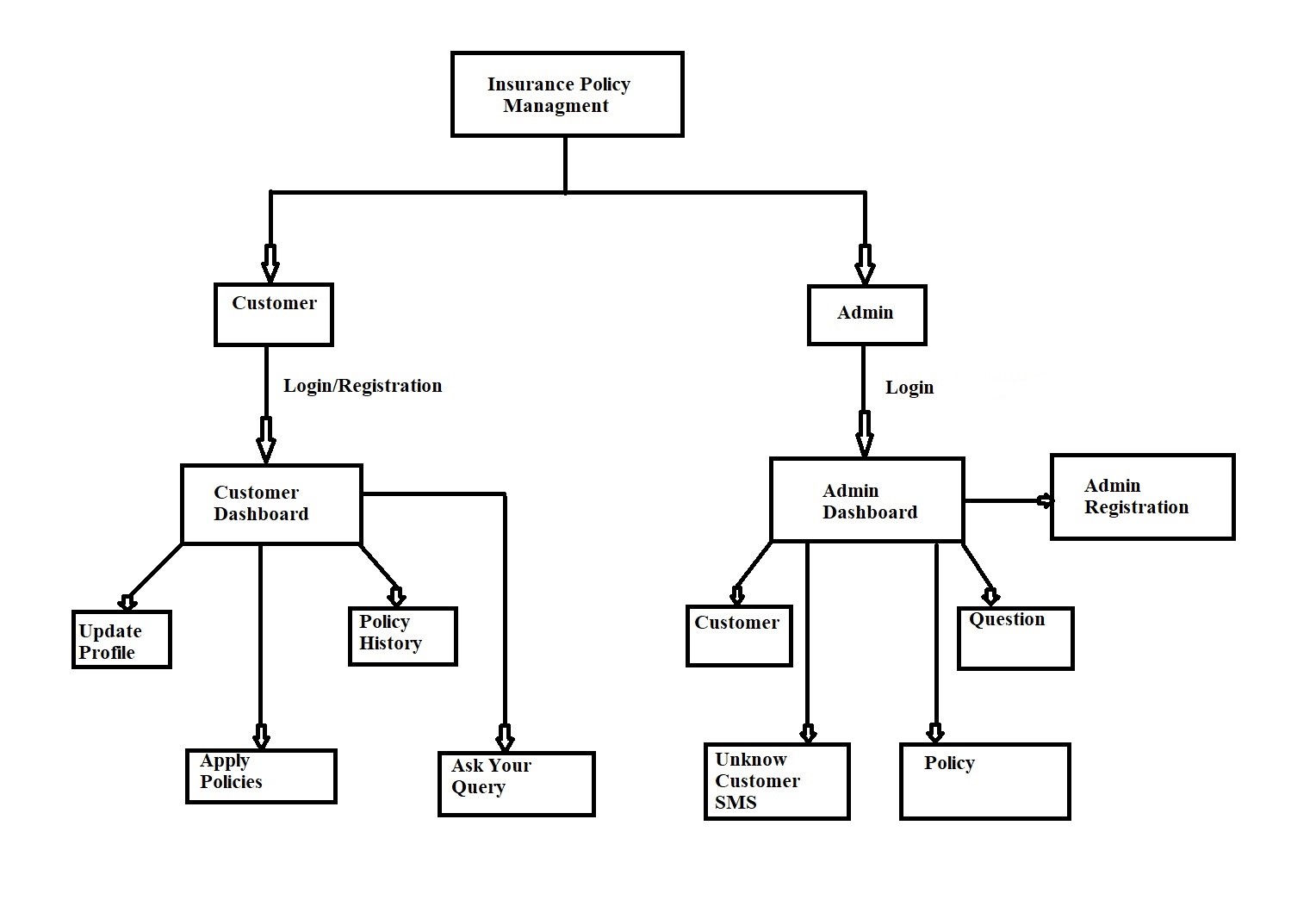
The “Insurance Policy Management” should support basic functionalities for all below listed users.

* Admin
* Customer
  1. **Authentication and Authorization**

Any end-user should be authenticated with a unique user-id and password.

Administrator has the rights to add policy information and view customer details. He can also view policy details and approve or reject policies Whereas Customer has a right to view policy and apply for policies.

* 1. **Functional Flow**



* 1. **Environment**

The system will be developed on any Windows OS machine using J2EE, MySQL, Spring-boot, Angular.

* Server – Apache Tomcat 9.6
* Database – MySQL
* JRE 1.8
* Spring Tool Suite
* JPA
* Windows Platform

1. **Sub-System Details**

The Insurance Policy Management is defined, wherein all users and admin need to login successfully before performing any of their respective operations. Find below (section 3.1 & 3.2) tables that provides functionality descriptions for each type.

* 1. **Admin Table**

The administrator as a user is defined to perform below listed operations after successful login

|  |  |  |
| --- | --- | --- |
| **Objects** | **Operations** | **Data to Insert** |
| Customer | View  Delete | Customer Name  Gender  Mobile No  Address |
| Unknown Customer SMS | View | Customer-SMS |
| Policy | View  Add  Update  Delete | Policy Id  Policy Name  Policy Price  Policy Category |
| Approvals | View  Modify | User Id  Policy Name  Policy Price  Policy Category |

* 1. **Customer Table**

The customer as a user is defined to perform below listed operations after successful login.

|  |  |  |
| --- | --- | --- |
| **Objects** | **Operations** | **Data to include** |
| User Login | Login  Registration  Update | Customer Name  Customer Email |
| Apply Policies | View  Apply | Policy Id  Policy Name  Policy Category  Amount  Apply Policy |
| Policy History | View | Policy Id  Policy Name  Policy Category  Policy Price  Date  Status |
| Queries | View  Add | Question topic  Description |

1. **Rest APIs used to Build**

**URL Table's: -**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr  No | URL’S | Method Type | Description | Format |
| Admin Table | | | | |
| 1 | /addadmin | POST | Saves the admin registration data | JSON |
| 2 | /showadmins | GET | Retrieves all the available admins | JSON |
| 3 | /adminshowdata/email/password | GET | Retrieves the data of a particular admin | JSON |
| 4 | /updateadmin/email | PUT | Updates the data of a given admin details | JSON |
| 5 | /deleteadmin/id | DELET | Deletes data of a particular admin | JSON |
| Customer Table | | | | |
| 6 | /addcustomer | POST | Saves the data of a user registration | JSON |
| 7 | /showcustomers | GET | Retrieves data of all the available customers | JSON |
| 8 | /showdata/email/password | GET | Retrieves the data of a particular customer | JSON |
| 9 | /login/email/password | GET | Validates the data of customer at user login | JSON |
| 10 | /updatecustomer/email | PUT | Updates the data of a given customer details | JSON |
| 11 | /deletecustomer/id | DELET | Deletes data of a particular user | JSON |
| 12 | /countcustomer | GET | Retrieves the total count of available customers | JSON |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | URL’s | Method Type | Description | Format |
| Customer Apply Policy Table | | | | |
| 13 | /applypolicy | POST | Saves the data of user applied policies | JSON |
| 14 | /getallaplicationofpolicy | GET | Retrieves the data of all the user applied policies | JSON |
| 15 | /updatestatus/id | PUT | Updates the status of the policies applied by user | JSON |
| 16 | /countApprove | GET | Retrieves the total count of approved policies | JSON |
| 17 | /countPending | GET | Retrieves the total count of pending policies | JSON |
| 18 | /countrejected | GET | Retrieves the total count of rejected policies | JSON |
| 19 | /countapllication | GET | Retrieves the total count of policy applications | JSON |
| Questions Customer Table URL’s | | | | |
| 20 | /savequestion | POST | Saves the data of the questions asked by users | JSON |
| 21 | /customerquestions | GET | Retrieves the data of all the questions asked by users | JSON |
| 22 | /ans/id | PUT | Updates the questions answers for a particular user | JSON |
| 23 | /findqustions/email | GET | Retrieves the data for a particular user's question | JSON |
| 24 | /countCustomerQuestions | GET | Retrieves the data about total number of customer's questions | JSON |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | URL’s | Method Type | Description | Format |
| Policys Table | | | | |
| 25 | /addpolicys | POST | Saves the data of a type | JSON |
| 26 | /getpolicys | GET | Retrieves the data of all the policies available | JSON |
| 27 | /updatepolicy/id | PUT | Updates the details of particular policy | JSON |
| 28 | /deletepolicy/id | DELET | Deletes the data of a particular policy | JSON |
| 29 | /countpolicie | GET | Retrieves data of total number of policies available | JSON |
| Unknown SMS Table URL’s | | | | |
| 30 | /addunknowsms | POST | Saves the data of a question asked by an unregistered user | JSON |
| 31 | /getunknownsms | GET | Retrieves the data of all the questions asked by unregistered users | JSON |
| 32 | /unknownsmscount | GET | Retrieves the data of total number of questions asked by unregistered user | JSON |

1. **E-R Diagram**
   1. Admin E-R Diagram

login

Dashboard

If Yes

* 1. User E-R Diagram

User

No

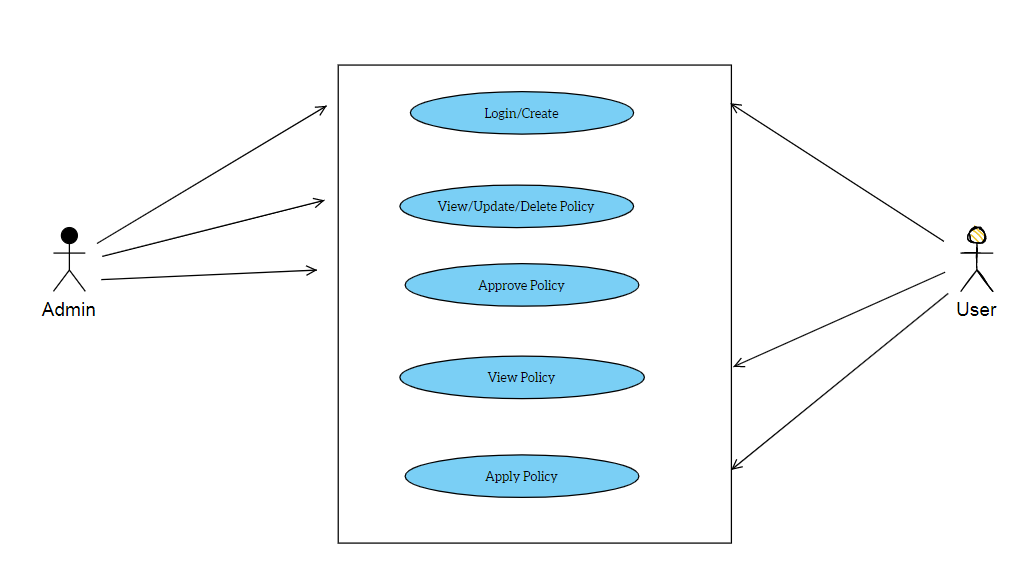
Registration

if

yes

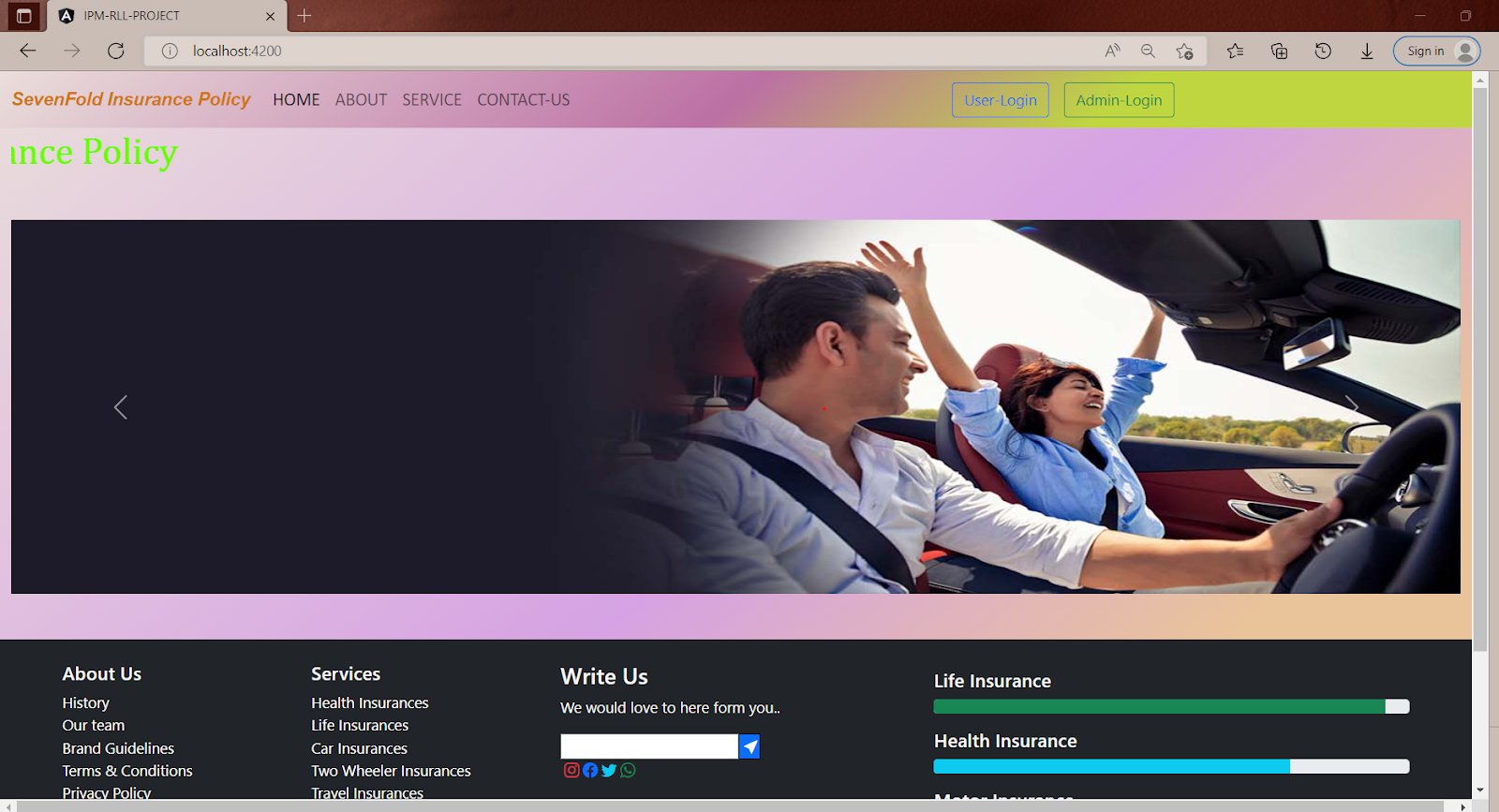
Dashboard

1. **Use-Case Diagram**

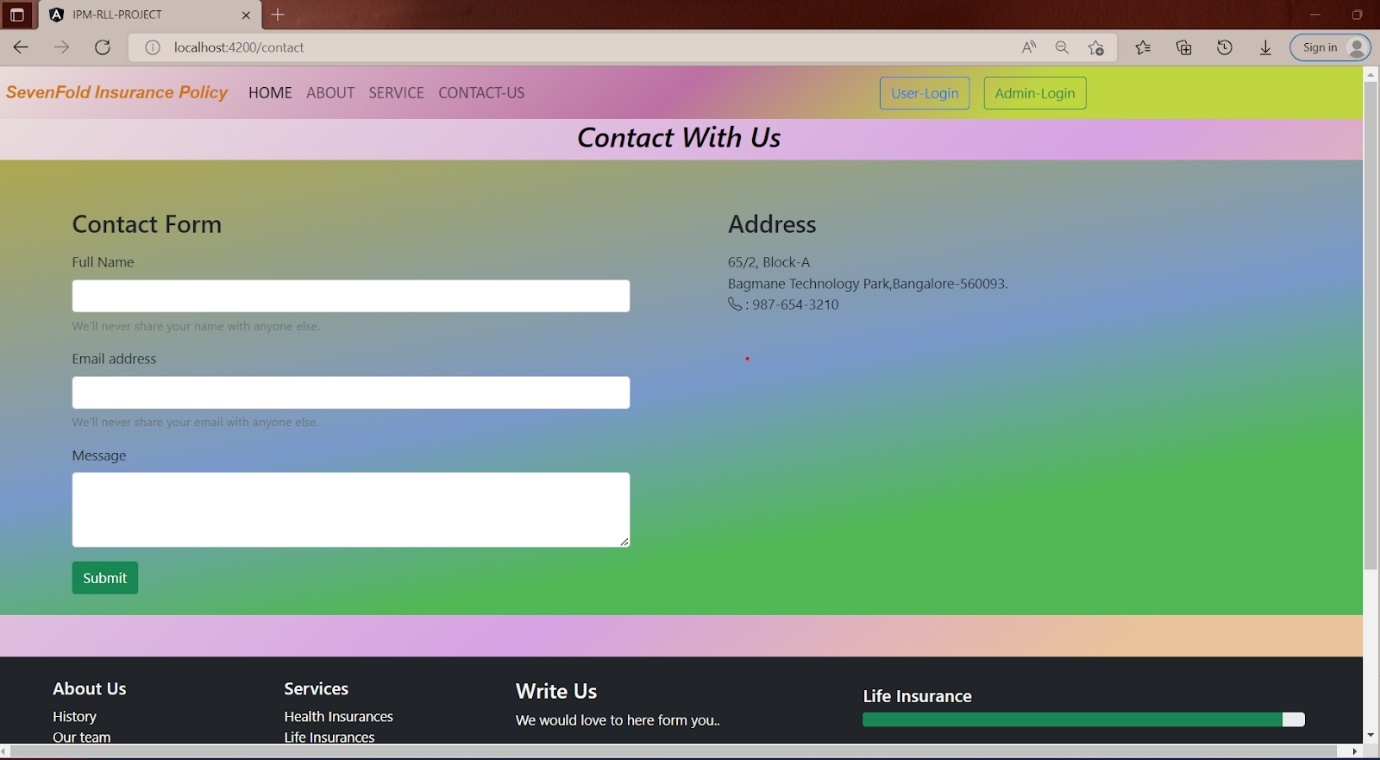


1. **Future Scope**
   * The administrator can add and remove policy into the database on a weekly basis.
   * You must not allow user to add same policy twice.
2. **Output Screenshot of Project**

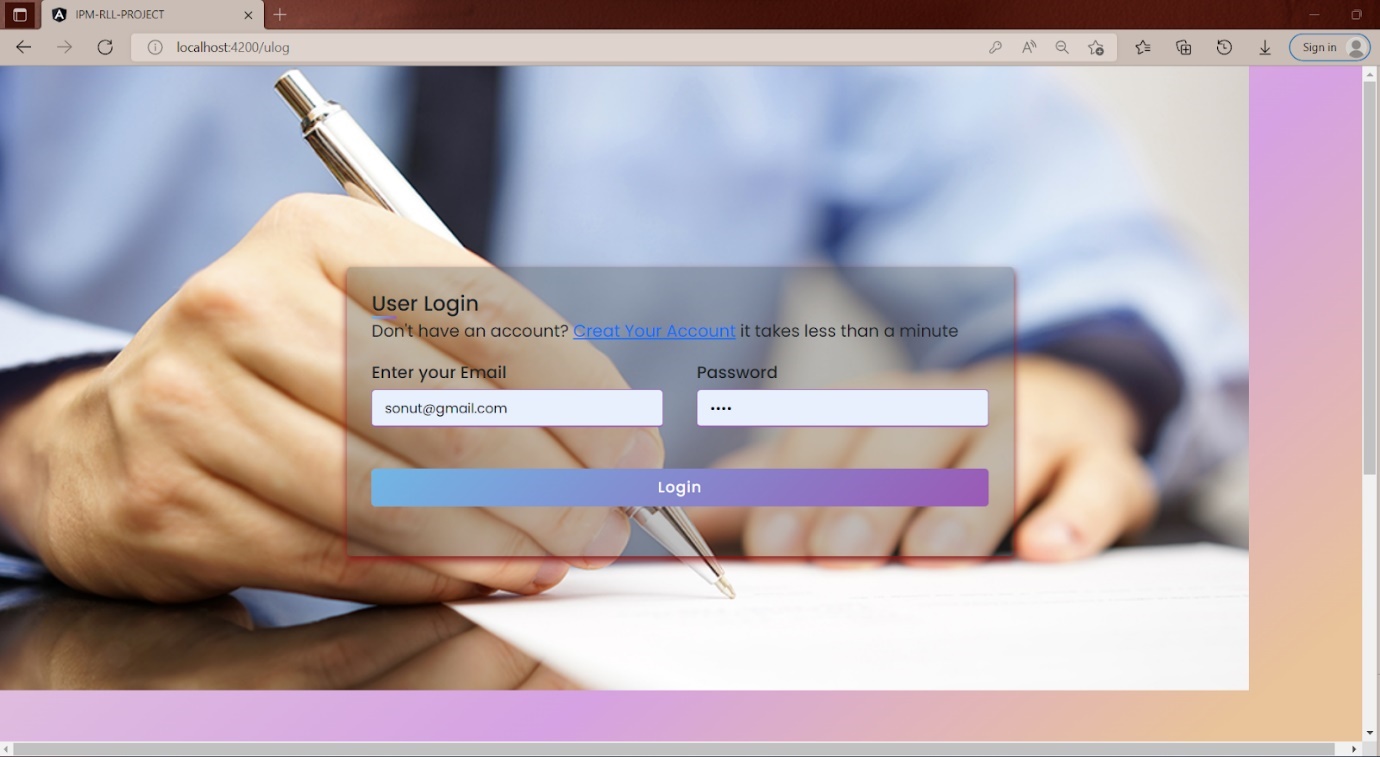
Home Page

****

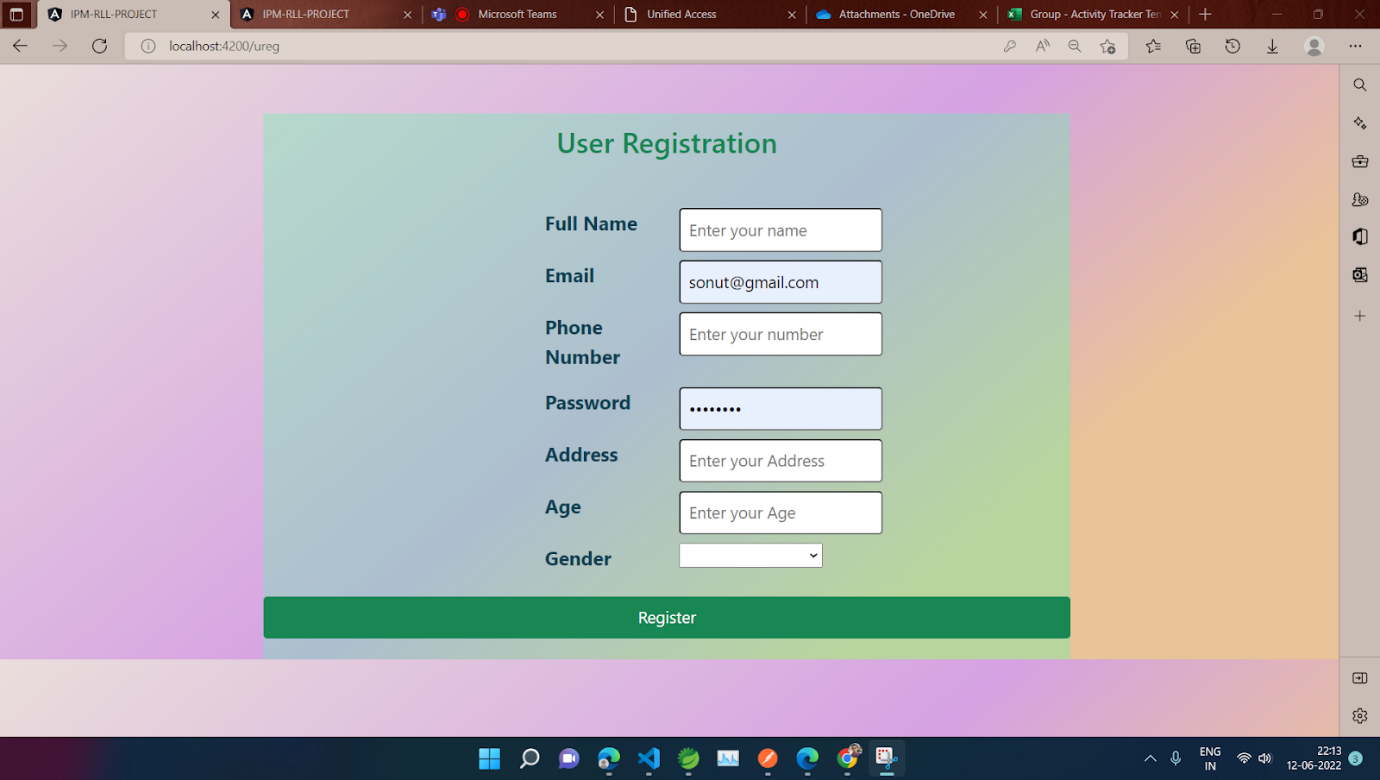
Contact Us Page

****

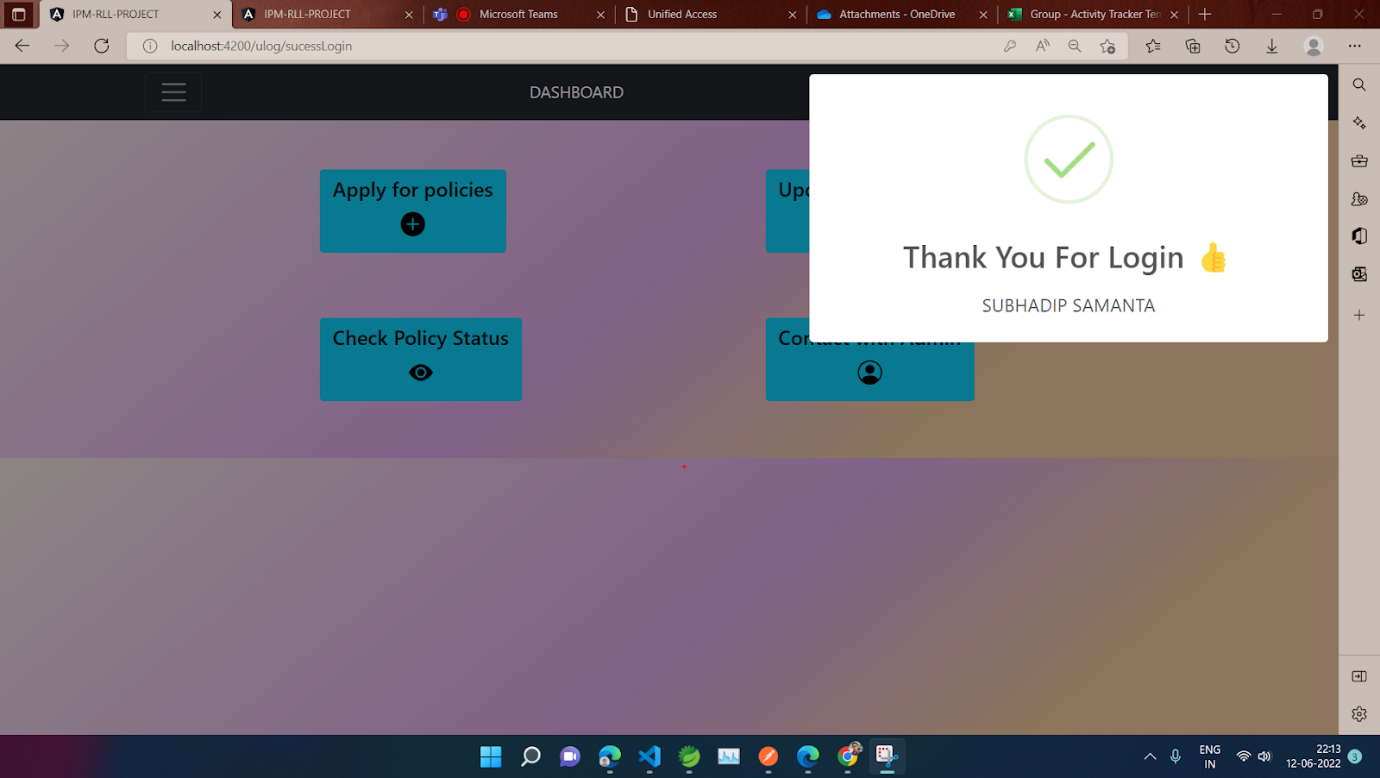
User Login Page

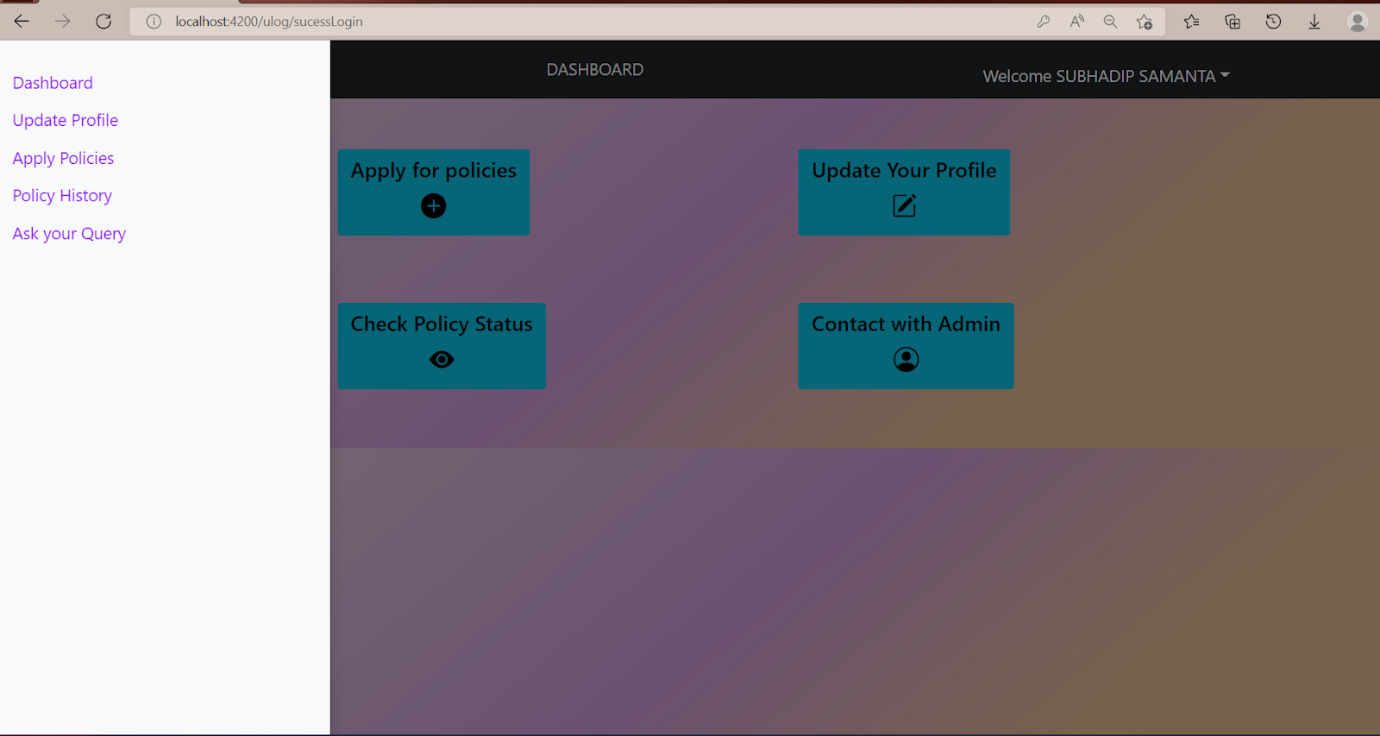
****

User Registration

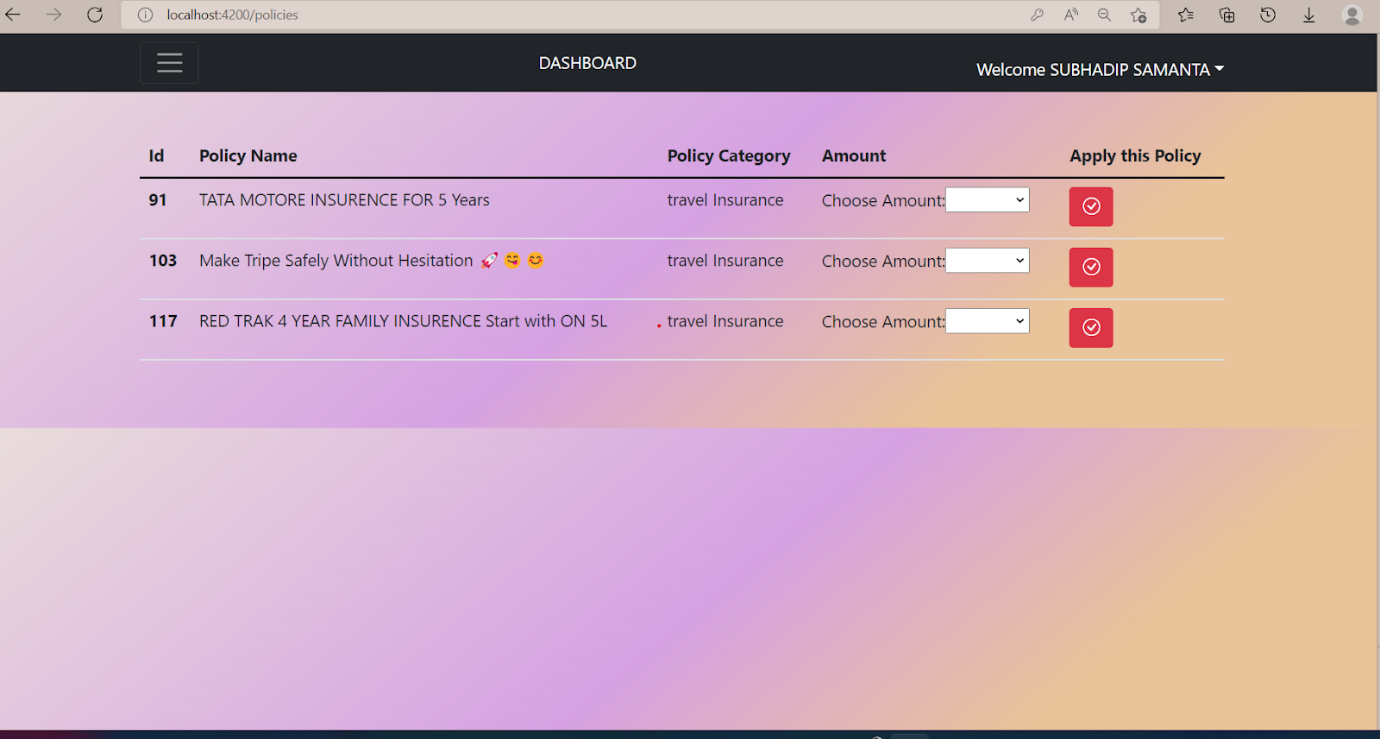
****

User Dashboard

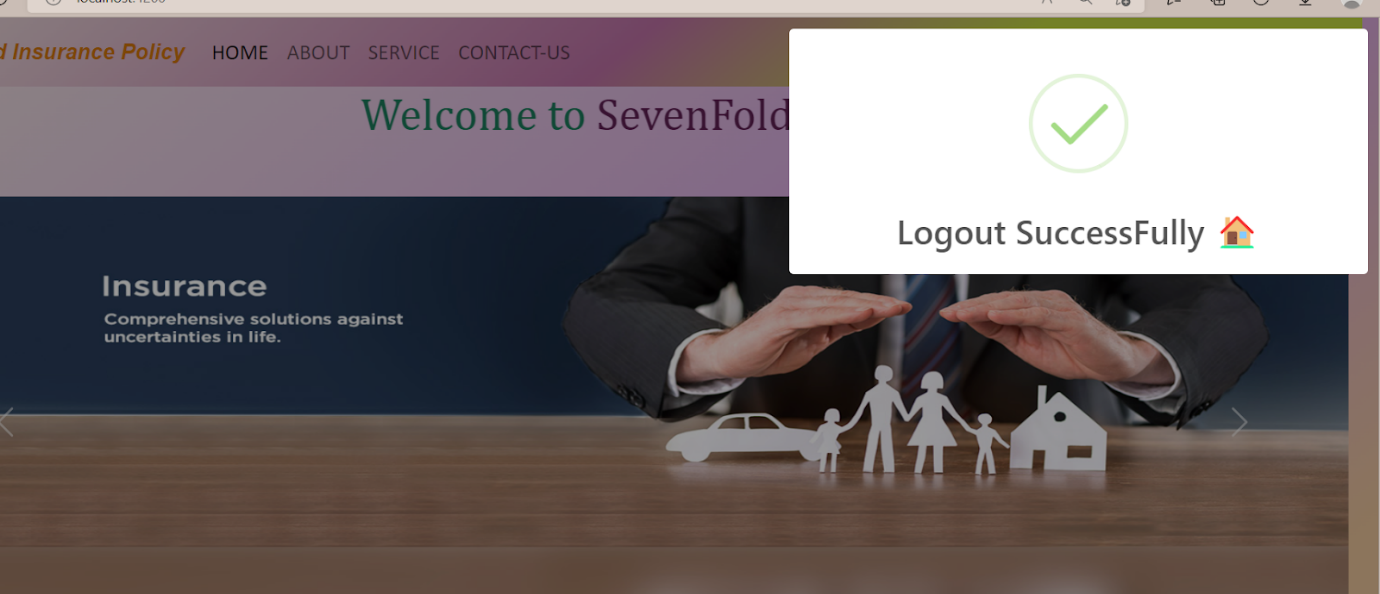
****

****

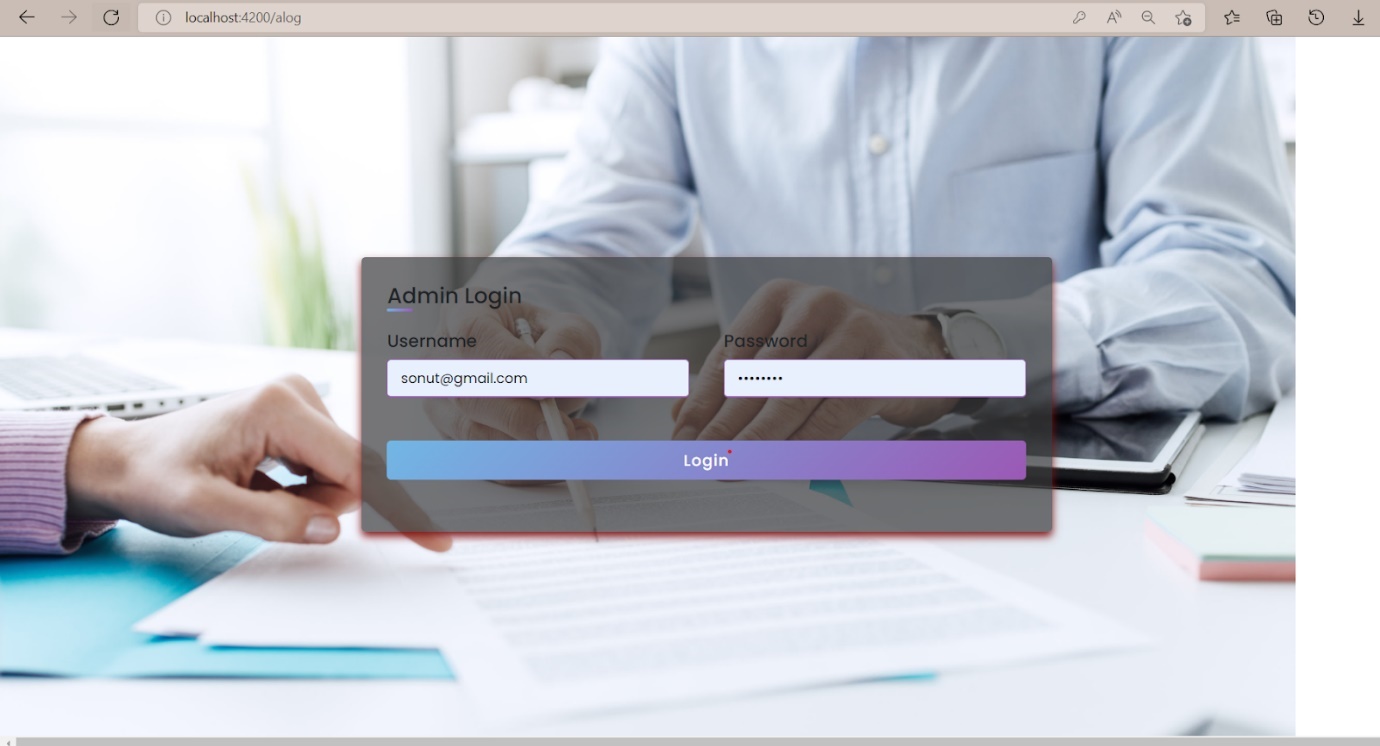
Apply Policy

****

User Logout

****

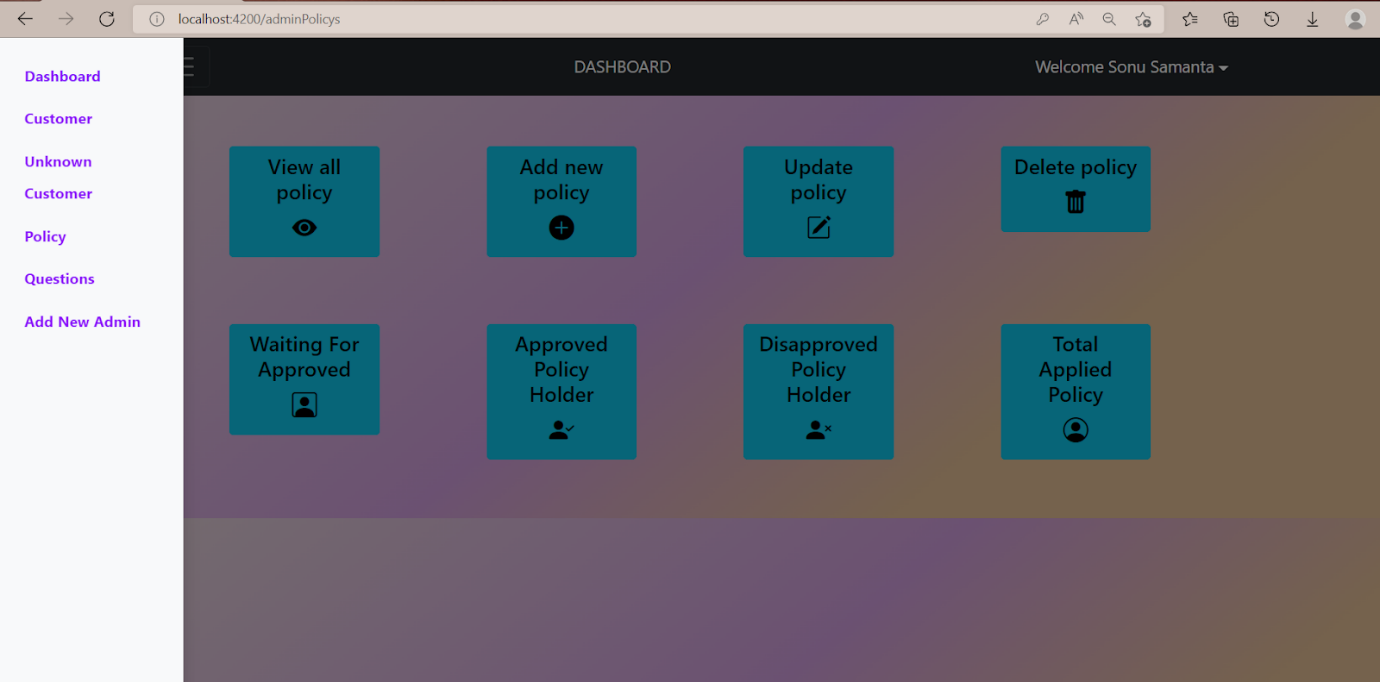
Admin Login

****

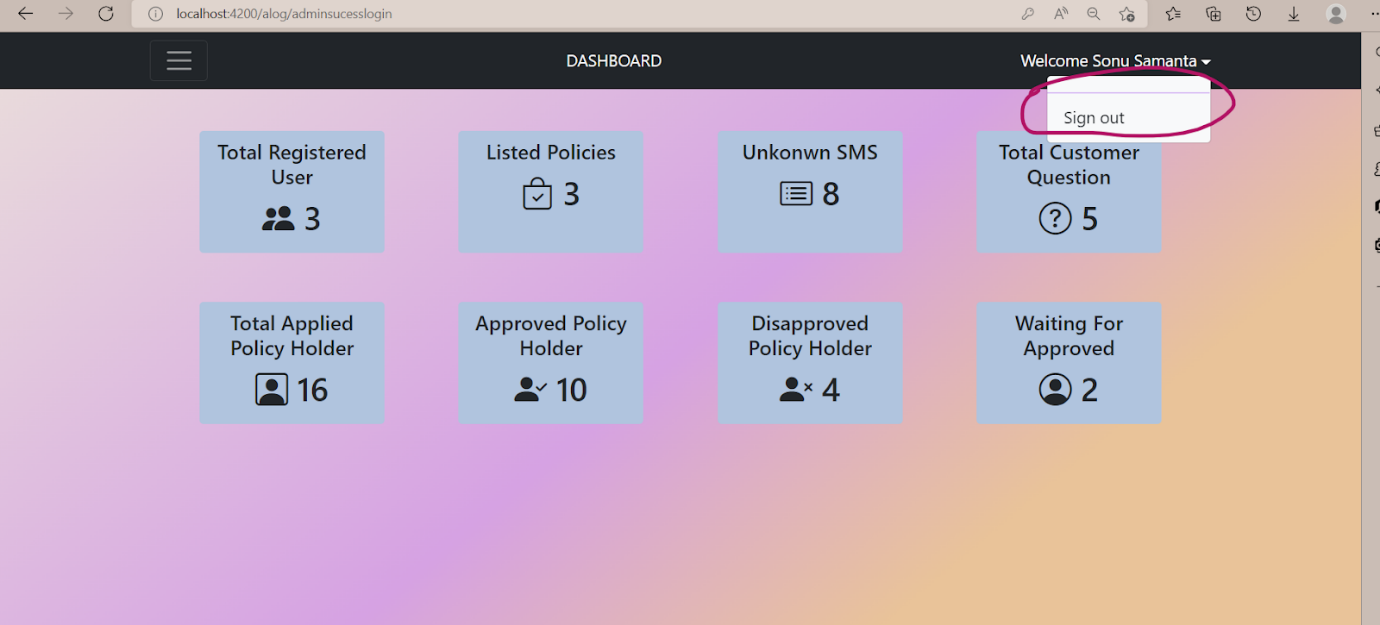
Admin Dashboard

****

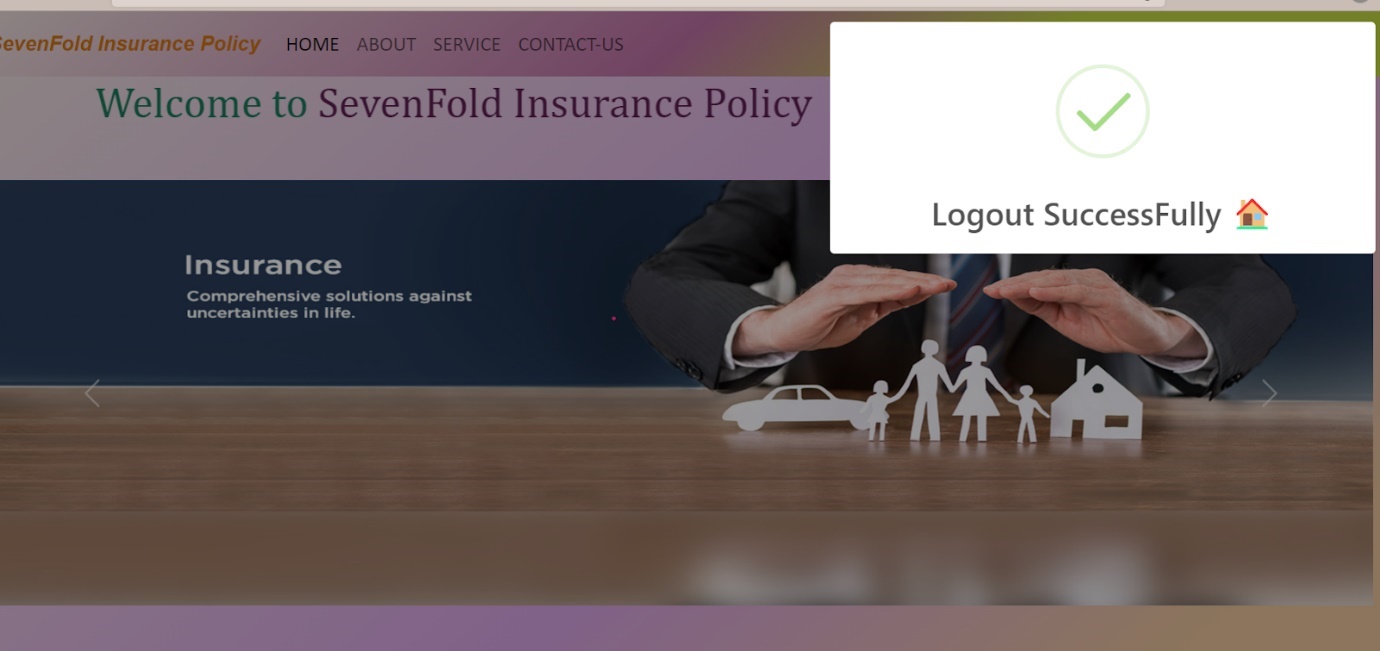
Admin Tasks

****

Admin Logout

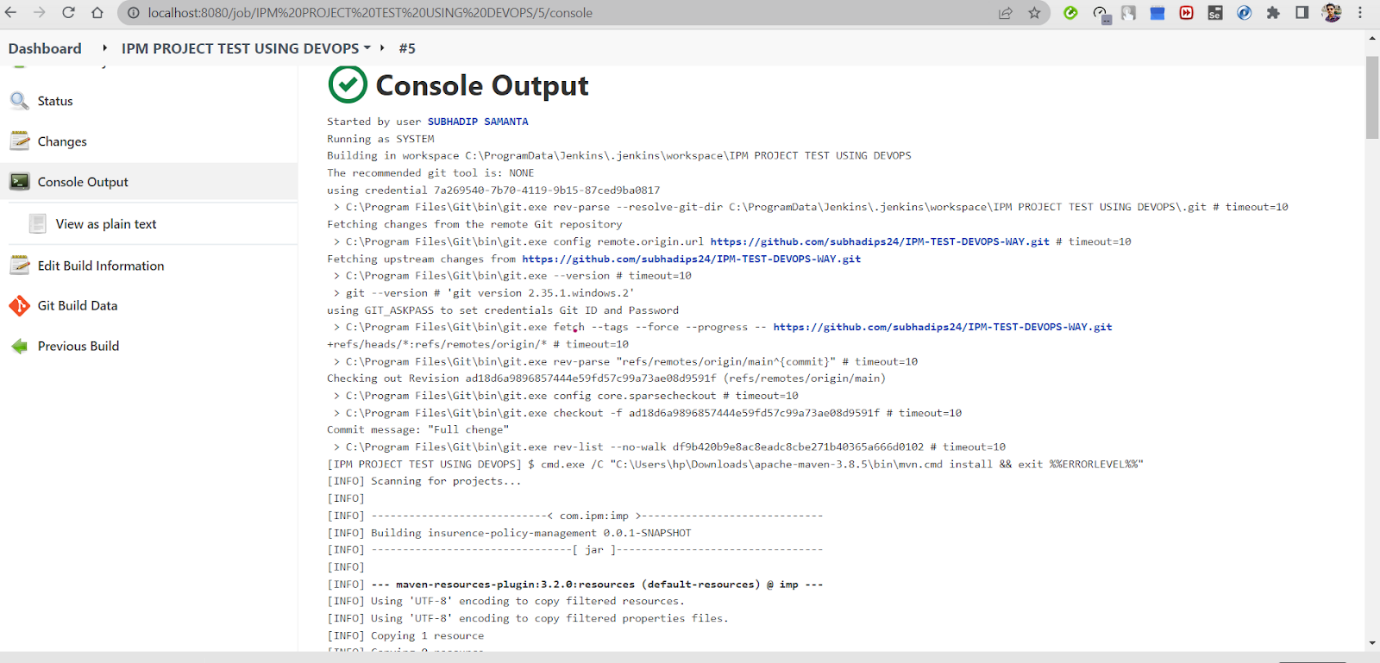
****

Admin Logout Successfully

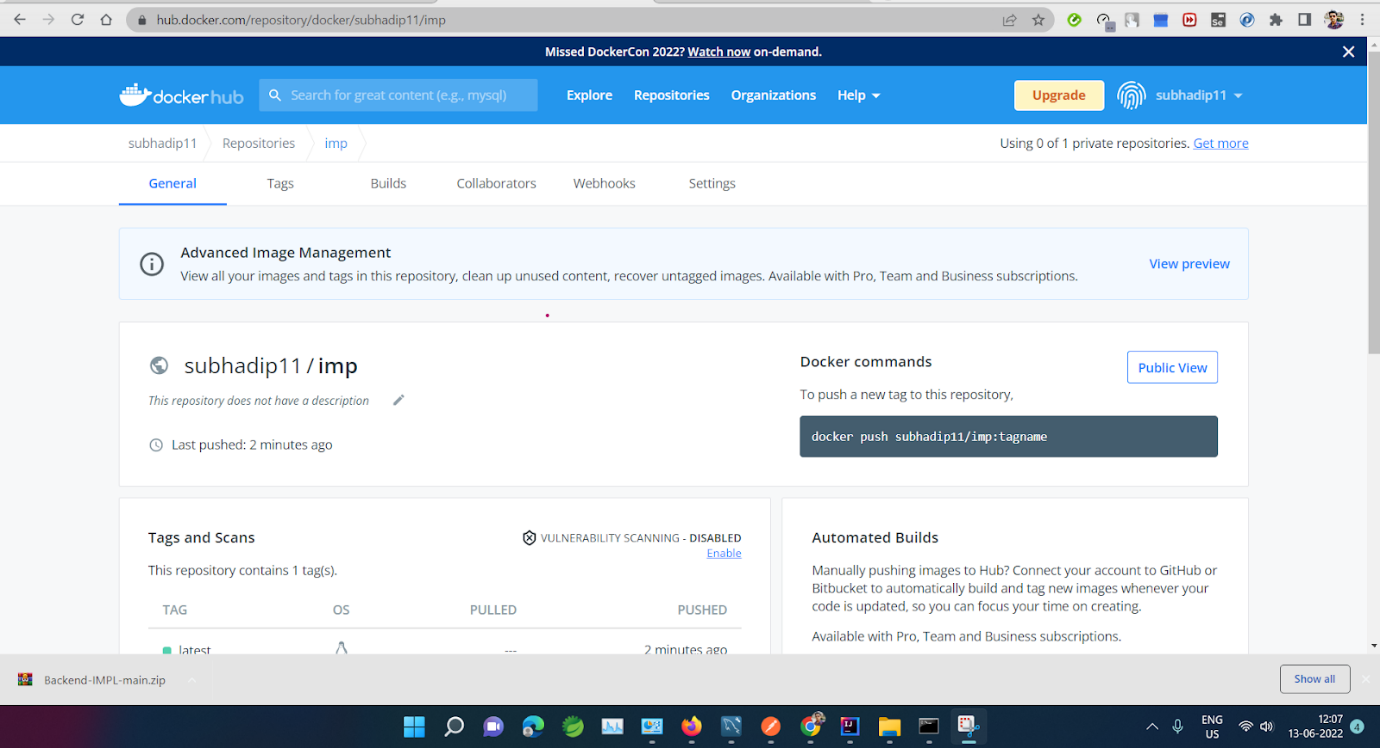
****

**9.Testing with DevOps**

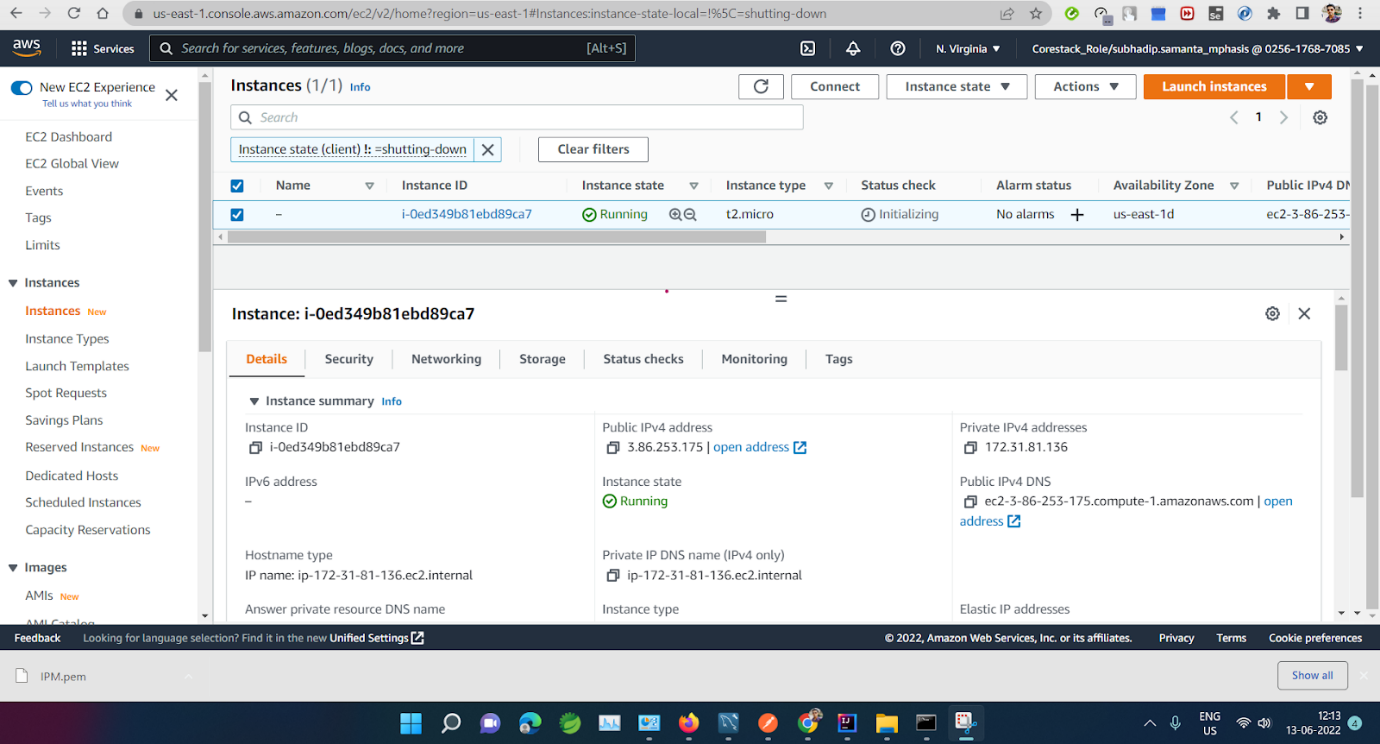
Jenkins(CI/CD)

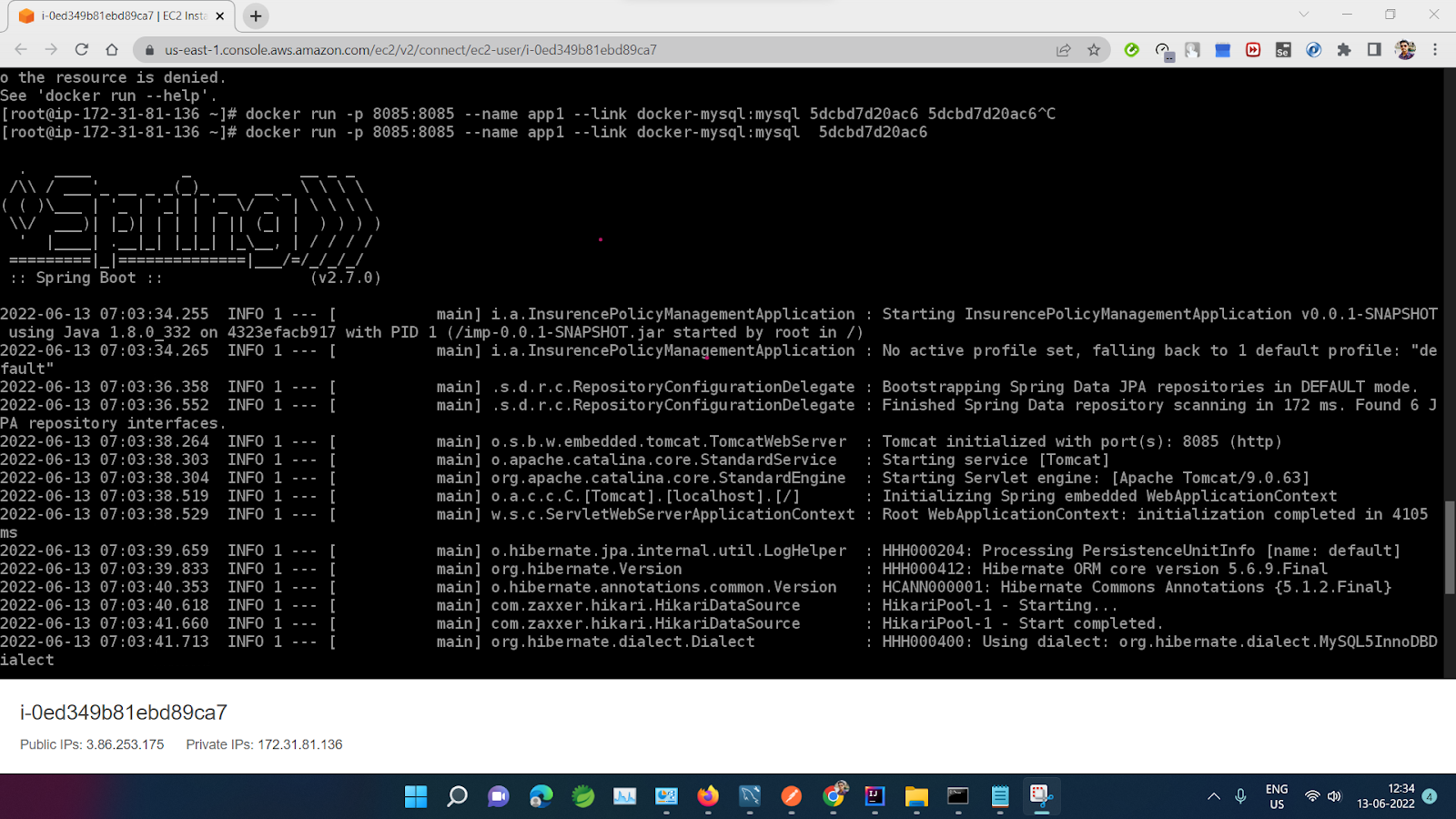


DockerHub



AWS





Testing with aws live link

