INTERNSHIP PROJECT REPORT

submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science Engineering

(Project Term January-May, 2019)

POLICY MANAGEMENT SYSTEM

(Policy Tracker)

at

COGNIZANT TECHNOLOGY SOLUTIONS

Submitted by:

Ayushi Dogra

Enrollment Number :151411

Under the Guidance of

Mr. Abhinandan Chivate

to

Department of Computer Science and Engineering

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, (H.P.)

DEPARTMENT OF COMPUTING SCIENCE ENGINEERING

CANDIDATE'S DECLARATION

I hereby declare that the Dissertation entitled "Policy Tracker-Policy Management System" is my own work conducted under the supervision of Guide/Co-Guide Name, Designation, Name of School at **Jaypee University of Information Technology, (H.P.)**.

I further declare that to the best of my knowledge this report does not contain any part of work that has been submitted for the award of any degree either in this university or in other university / Deemed University without proper citation.

Ayushi Dogra

(151411)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:_____

Guide Name

Mr. Abhinandan Chivate

Trainer

Cognizant Technology Solutions

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, (H.P.)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the work done in this Capstone Project Report named **"Policy Tracker- Policy Management System"** has been satisfactorily completed by **Ms. Ayushi Dogra** Registration No 151411. This has been a bonafide piece of work, which was carried under my/our guidance in the Organization "**Cognizant Technology Solutions**" for fulfilment of the degree of Bachelor of Technology.

Mrs. ShilpaMahajani

Project Manager

Cognizant Technology Solution

Under the Guidance of

Mr. Abhinandan Chivate

Trainee and Mentor

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, (H.P.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACKNOWLEDGEMENT

This internship opportunity which I got with **Cognizant Technology Solutions** has been a good chance to learn and professionally develop. So, I consider myself to be a lucky person because I was provided with a chance to be it's part. I am also thankful for having an opportunity to meet many amazing people and professionals who guided me throughout this internship period.

I am using this opportunity to convey my acknowledgement and thankfulness to the **Ms. Shilpa Mahajani** at CTS, Pune who despite being extraordinarily hard-pressed with her duties, took out time to help keep myself on the correct path and allowed me to work on this project in their esteemed organization.

Ayushi
Dogra
(151411)

List of Figures

List of Tables

TABLE OF CONTENTS

PROJECT OBJECTIVES

METHODOLOGY

SYSTEM DESIGN

TEST PLAN

RESULTS AND PERFORMANCE ANALYSIS

CONCLUSION

PROJECT OBJECTIVES

CHAPTER 1

INTRODUCTION

This Project is aimed at:

 This Project is about the admin and user login in which they would manage and access policies as required.

 This project is about Policy Management System where the user/admin end would login to buy, delete and edit policies as per the access rights.

The admin would enter valid credentials in login page t

o enter the admin home page and buy, remove, add and search policies.

Similarly the user would enter valid credentials in login page to enter the user home page and buy, remove and search policies.

This document's aim is to systematically capture prerequisites for this project and system to be developed. The document also captures the Functional requirements and serves as an input for the scope of project.

1.1 Objectives

Below are the objectives that shall be fulfilled post the execution of this project:

Admin

- Access to admin home page
- User registration & credential authentication

- Add new policies
- Edit Policies
- View existing policies
- Search policies

User

- User credential authentication
- Access to user home page
- View purchased policies and policies selected and yet to be purchased in the cart
- Buy and view policies
- Search policies
- Remove policies
- Policy Payment

1.2 Intended Audience

 Interns/Project Team

 Mentors and SME's.

1. PROBLEM'S PROFILE AND SCOPE OF THE STUDY

Through proven experience from successful organizations of all sizes, a strong program of policy and procedure management is much more than a necessary evil to have in place in case something goes wrong. Individual policy and procedure documents are the critical framework upon which an organization's compliance effectiveness and operational success are built. An organization's policies provide the basic rules, direction and definitions that not only protect a company, but also provide formulas for profitability and productivity. If an organization's policies and critical procedures are not managed properly, time is wasted, money is lost and risk exposure is elevated.

The solution developed will address the objective in a holistic manner and will have all the features and functionalities which shall let the portal allow a user to keep a record of his

policies and buy new policies and admin to add new policies along with other features such as edit and search.

CHAPTER - 2

PROBLEM ANALYSIS

2.1. PRODUCT'S DEFINITION

The product is developed to solve problem of policy management by the admin and then helping users to buy the policy. People usually have to stand in queues to get themselves insured, in banks. People plan to buy a policy and then they have to contact a policy agent who will arrange a meeting with the bank officials on their behalf. This would cost them a good amount of money. Using the Policy Tracker users will be able to buy a policy using our simple user interface within minutes.

2.2. FEASIBILITY ANALYSIS

It is the analysis used to measure the capability and probability for completion of a project successfully. This must consider the factors which affect this like economic as well as technological and not only legal but also scheduling considerations. Managers of the project make use of feasibility studies to calculate the potentially negative and positive results of the project prior putting a considerable amount of time and money in the project.

2.2.1. TECHNICAL FEASIBILITY

All the technology's that are required are open source and are freely available to use like JAVA8, Spring, Hibernate and Bootstrap along with all the learning materials. Eclipse is use as an IDE to develop the project along with Maven for dependency injection in the project.

2.2.2. FINANCIAL FEASIBILITY

All the software used to develop this application is freely available so no cost is spent in the development process. Since the software's are open source we will get free update and new features in the future for free.

METHODOLOGY

CHAPTER - 3

ANALYSIS OF SOFTWARE REQUIREMENTS

3.1. TECHNOLOGIES

3.1.1. JAVA

It is a high-level programming language which was developed by Sun Microsystems and was made to release in the year1995. It runs on a wide variety of the platforms, like Mac OS, Windows, and variety of versions of UNIX. It is a Write Once and Run Anywhere language.

It is −

- Everything in this language is an Object and hence is an object oriented language.
- Java is language that is platform independent . It is due to the conversion of source code to byte code.
- It's syntax is derived from C and C++ and hence it is a simple and is therefore easy to learn.
- It is also a secure language and consists of various authentication techniques.
- Portable – It is architecture-neutral and has no aspects which are implementation dependent making it a portable language.
- It is also portable and robust language.
- It also supports multithreading i.e various programs can sun simultaneously.
- It is also an interpreted language unlike C.
- It is also used to enable high performance through Just On Time compilers.
- It's easily adaptable to the distributed internet and web environment.

- C or C++ are lesser dynamic than Java because iteasily adapts to evolving environment.

3.1.2. SPRING

It's a lightweight framework provides supports other frameworks like Hibernate, Struts, etc. Framework is a structure where we get solution to various types of technical complexities.

The Spring framework consists of WEB MVC ,Context, ORM, etc.

It is used to develop Java application of any type, but it requires certain extensions.It aims to make development of J2EE easy and it enables a POJO-based programming model.

Following are the benefits of Spring Framework −

- It is well-organized structure. Although the count of packages and classes is considerable, we need to worry about only those which we need and not the others.
- It uses technologies such as logging, ORM, JEE etc..
- It's structure of web is designed well and it provides a good alternative to web frameworks.
- It just requires a robust container of servlet like Tomcat server.
- Testing of any system with help of Spring is easy since the process is moved inside the framework. Also, making use of JavaBeanstyle persistence objects, it is quite easy to use dependency injection.
- It is beneficial to develop and deploy applications for computers having limited amount of CPU resources and main memory.
- It also gives API suitable to convert exceptions specific to a certain technology to unchecked, consistent exceptions.
- It is used to provide an interface which is of transaction management which can be scaled to global transactions .

3.1.3. HIBERNATE

It is Object-Relational Mapping solution to JAVA. It is open source framework. It is powerful Query service for any type of Application in Java.

It's used to map Java objects to database tables and relieves the developer from complex tasks.

It handles all the works between database server and Java objects.

Figure 1: Hibernate ORM

Advantages

- It is used to provide simple APIs to store and retrieve Java objects to and from database.
- If there are any changes in the database then you only need to change the XML file properties.
- It maps classes to database tables by the help of XML files , without any need to write any code.
- It reduces access to database with the help smart fetching methods.
- It does'nt require any application server in order to operate.
- It manipulates the complicayed associations of objects in database.
- It also abstracts away unknown SQL types and gives a way to work with popularr Java Objects.

3.1.4. JSP

It is server-side programming technology which is used to enable platform-independent method to build Web-based applications. It has access to complete family of Java APIs.

Java Server Pages full form of JSP is a technology to develop Webpages which support dynamism which helps developers to insert code in HTML pages by using special JSP tags, mostly which start with <% .

It is a kind of Java servlet which was designed to fulfill user interface's role for web application in Java. JSPs are written by Web developers in the form of text files which combine XML elements, HTML code and embedded JSP actions as well as commands.

We can collect input from users through Webpage forms or another source using JSP, and can dynamically create Webpages.

JSP tags are used to retrieve information from the database and register user preferences, , share information among requests and pass control among pages etc.

Advantages -

Comparison of JSP with other technologies −

with Active Server Pages (ASP)

There are two advantages of JSP over ASP. Firstly, the dynamic portion of the code is written in Java and not any other language that may be platform dependent and therefore it is easier and more powerful to use. Secondly, it is portable.

with Pure Servlets

It is easier to write regular HTML over plenty of println statements which generate HTML codes.

with Server-Side Includes (SSI)

For "real" programs that use form data, make database connections, SSI is not suitable.It can only be used for simple inclusions.

with JavaScript

JavaScript cannot interact with web server to perform complicated tasks such as database access etc.

with Static HTML

Dynamic information is not contained in regular HTML.

3.2. TOOLS USED

3.2.1. ECLIPSE

Eclipse is used in computer programming. It is integrated development environment (IDE) It is mostly used Java IDE. It consists of a base workspace and an extensible plug-in system to customize the environment. It is written in Java. It is mainly used to develop applications in Java.

3.2.2. VISUAL STUDIO CODE

It is a source-code editor. It was developed by Microsoft for macOS ,Windows and Linux. It consists of support for debugging, snippets, code refactoring etc. It can be easily customized, so users can change the preferences. It's sourse code is free and open source.

3.2.3. MYSQL DATABASE

It is an easy-to-use RDBMS which is used for various purposes of data maintenance. It was developed by MySQL AB. MySQL AB is a Swedish company. It is becoming very much popular now a days because of the following featuress:

- Open-source technology and therefore we need not pay for it.
- It is a powerful program which is used to handle a large subset of the functionality of various databases.
- It is used in standard form of SQL language.
- IWorks on various types of operating systems and also with different programming languages like C++, JAVA, C etc.
- It is fast even with very large data sets.
- It can work well PHP, which is mostly suited for for web development.
- It can support very large data sets, up to 50 million tuples in a table.

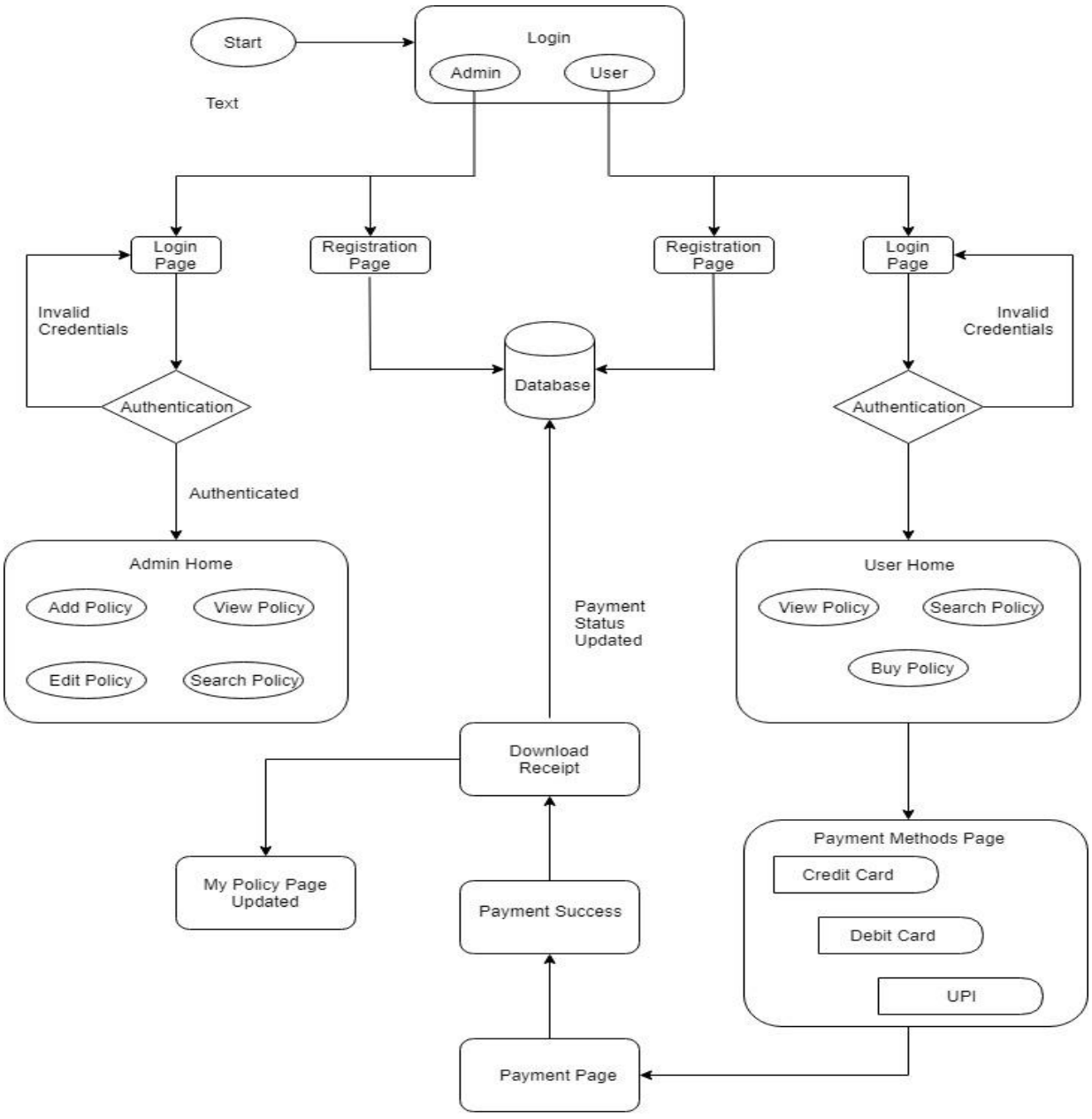
- It can be easily customized.

SYSTEM DESIGN

CHAPTER - 4

DESIGN

4.1. SYSTEM DESIGN



4.2. PROCESS ARCHITECTURE

Below is the overall functional flow of the project including the components of interaction

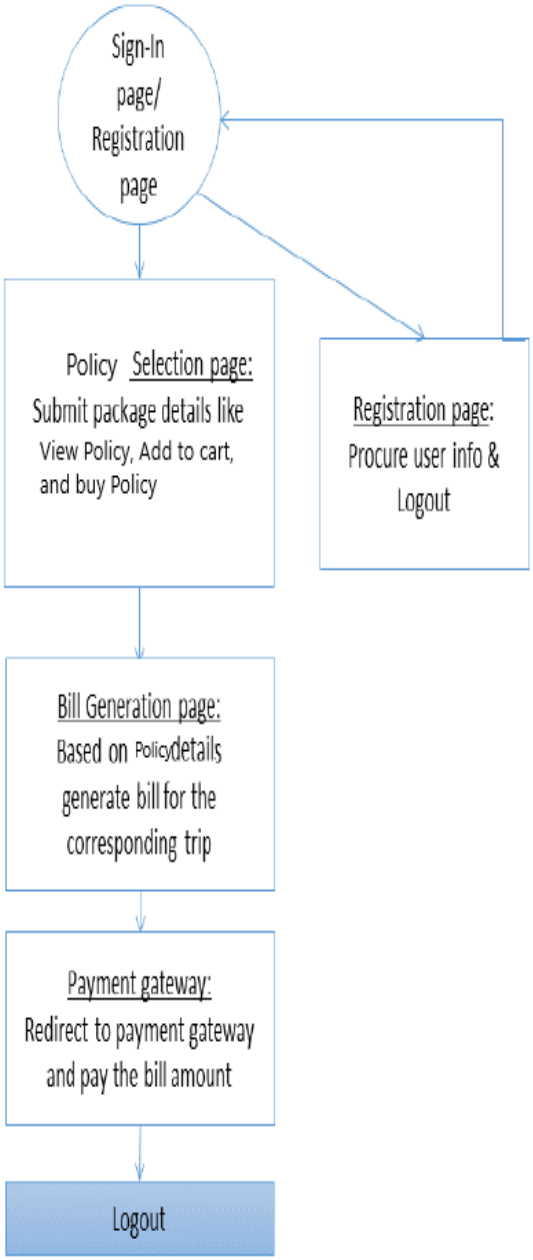


Figure 2:Process Architecture

4.3. HIGH LEVEL BUISNESS REQUIREMENTS

S.No.	Business Requirement ID	Short Description	Description in detail	Interacting Business Processes
1	Req_1	User Registration	Ability of the system to procure the fundamental details of the user	
2	Req_2	User Authentication	Ability of the system to authenticate the user credentials of the registered user	
3	Req_3	Policy Detail	Ability of the system to procure the details of the Policy Detail	
4	Req_4	Bill Generation	Ability of the system to generate the bill of corresponding tour	
5	Req_5	Payment	Ability of the system to redirect to a secure payment gateway for bill payment	

Figure 3:High Level Business Requirements

4.4. FUNCTIONAL REQUIREMENT

Value	Rating	Description
1	Critical	This requirement is critical to the success of the project. The project will not be possible without this requirement.
2	High	This requirement is high priority, but the project can be implemented at a bare minimum without this requirement.
3	Medium	This requirement is somewhat important, as it provides some value but the project can proceed without it.
4	Low	This is a low priority requirement, or a "nice to have" feature, if time and cost allow it.
5	Future	This requirement is out of scope for this project, and has been included here for a possible future release.

Figure 4:Functional Requirement 1

Req. #	Rationale Categorization	Business Requirement	Req. Type	Priority	Originator	BR Traced to Business	Remarks

						Requirement / Use case ID	
Req_1.1	User Registration	Screen should display the option for Admin login / Registration and User login / Registration	UI	Critical	NA	Req_1	Req_1.1
Req_1.2	User Registration	When the user clicks on the registration link, it should re-direct to registration form.	UI	Critical	NA	Req_1	Req_1.2
Req_1.3	User Registration	User needs to fill some of the basic attributes/fields as mentioned below in requirement: First Name, Last Name, Age, Gender, Contact Number, User Id, Password	UI	Critical	NA	Req_1	
Req_1.4	User Registration	Clicking 'Submit' should validate the datatype constraints for each field	F	Critical	NA	Req_1	Req_1.4

Req_1.5	User Registration	User failing to provide information on the mandatory fields be provided with an alert message – 'Please update the highlighted mandatory field(s).' Also, highlight the missed out field in red	E	Medium	NA	Req_1	Req_1.5
Req_1.6	User Registration	Post-successful field level validation, save the information in the database	F	Critical	NA	Req_1	Req_1.6
Req_1.7	User Registration	Upon saving the information in the database, display the message 'Your details are submitted successfully'.	E	Medium	NA	Req_1	Req_1.7
Req_1.8	Credential Authentication	A registered user – is able click 'Login' link, after keying in 'User ID' & 'Password' field and get his credentials authenticated with the existing database entry.	F	Critical	NA	Req_1	Req_1.8

Req_1.9	Admin Registration	When the Vendor clicks on the registration link, it should re-direct to registration form.	UI	Critical	NA	Req_1	Req_1.9
Req_1.10	Admin Registration	Vendor needs to fill some of the basic attributes/fields as mentioned below in requirement: First Name, Last Name, Age, Gender, Contact Number, Vendor Id, Password	UI	Critical	NA	Req_1	Req_1.10
Req_1.11	Admin Registration	Clicking 'Submit' should validate the datatype constraints for each field	F	Critical	NA	Req_1	Req_1.11
Req_1.12	Admin Registration	Admin failing to provide information on the mandatory fields be provided with an alert message – 'Please update the highlighted mandatory field(s).' Also, highlight the missed out field in red	E	Medium	NA	Req_1	Req_1.12
Req_1.13	Admin Registration	Post-successful field level	F	Critical	NA	Req_1	Req_1.13

		validation, save the information in the database					
Req_1.14	Admin Registration	Upon saving the information in the database, display the message 'Your details are submitted successfully'.	E	Medium	NA	Req_1	Req_1.14
Req_1.15	Credential Authentication	A registered user – is able click 'Login' link, after keying in 'Admin ID' & 'Password' field and get his credentials authenticated with the existing database entry.	F	Critical	NA	Req_1	Req_1.15
Req_2.1	Policy Registration	On successful authenticating the Admin,system should allow the admin to create policy	F	Critical	NA	Req_2	Please refer to Table 2.0 under References
Req_2.2	Policy Registration	On valid creation of the policy,system should create a policy ID	F	Medium	NA	Req_2	Req_2.2
Req_2.3	Policy Registration	Admin on clicking the Submit button, all the fields should be validated and the details has to be saved in the database.	E	Medium	NA	Req_2	Req_2.3

Req_2.4	Policy Registration	Admin failing to update the required fields, Message should be thrown as – 'Please update the highlighted mandatory field(s).' Also, highlight the missed out field in red	E	Medium	NA	Req_2	
Req_3.1	Edit Policy Registration	Admin should be able to edit the name of the policy	UI	Medium	NA	Req_3	
Req_3.2	Edit Policy Registration	Admin should be able to edit the Policy type	UI	Medium	NA	Req_3	
Req_3.3	Edit Policy Registration	Admin should be able to edit the duration of the policy	UI	Medium	NA	Req_3	Req_3.3
Req_3.4	Edit Policy Registration	Admin should be able to edit the amount of the policy	UI	Medium	NA	Req_3	Req_3.4
Req_3.5	Edit Policy Registration	Admin on clicking the Submit button, all the fields should be validated and the details has to be saved in the database.	F	Medium	NA	Req_3	Req_3.5
Req_3.6	Edit Policy Registration	Admin failing to update the required fields, Message should be thrown as – 'Please update the highlighted	F	Medium	NA	Req_3	

		mandatory field(s).' Also, highlight the missed out field in red					
Req_3.7	Edit Policy Registration	Upon saving the information in the database, display the message 'Your details are submitted successfully'.	F	Medium	NA	Req_3	
Req_4.1	Search Policies	User Should be able to select policy type, policy id, Number of years, Company Name,Policy name	UI	Critical	NA	Req_4	
Req_4.2	Search Policies	System should display the list of policies as per the search criteria	UI	Medium	NA	Req_4	Req_4.2
Req_4.3	Search Policies	User should select atleast one criteria before selecting the search policy button	F	Medium	NA	Req_4	Req_4.3
Req_4.4	Search Policies	User upon clicking the Search policy button without selecting any fields should be prompted with an error message	F	Medium	NA	Req_4	Req_4.4
Req_5.1	Policy Payment	The user on going to Policy payment page should display	UI	Critical	NA	Req_5	Req_5.1

		the user details – policy Id, Bill Date, Payment amount, Fine, Due date					
Req_5.2	Policy Payment	User upon clicking the Payment button user will pay the corresponding bill amount through the gateaway and he/she will have different options to pay the amount like Credit card,debit card,upi.	F	Critical	NA	Req_5	Req_5.2
Req_5.3	Policy Payment	After successful payment the user will be displayed a success message that 'Payment is successful' and the status of the bill should be changed from 'pending' to 'paid' in the database.	F	Critical	NA	Req_5	Req_5.3

Figure 5:Functional Requirements

4.5. DATABASE DESIGN

4.5.1. USER

Field Name	Field Type	Data Type	Mandatory	Possible Values
First Name	Text(50)	Alphabetic	Yes	First Name
Last Name	Text(50)	Alphabetic	Yes	Last Name

DOB	Text(10)	DD/MM/YYYY	Yes	DOB
Gender	Drop Down	NA	Male, Female	Gender
Contact Number	Text(10)	Numeric	Yes	Contact Number
Address	Text(60)	Alphanumeric	Yes	Address
Email Id	Text(15)	Alphanumeric	Yes	Email Id
Qualification	Text(10)	Alphanumeric	Yes	Qualification
Salary per Month	Numeric(10)	Numeric	Yes	Salary per Month
PAN No	Text(10)	Alphanumeric	Yes	PAN No
Employer type	Text(10)	Alphanumeric	No	Employer type
Employer	Text(10)	Alphanumeric	No	Employer
Hint Question	Text(50)	Alphanumeric	Yes	Hint Question
Hint Answer	Text(50)	Alphanumeric	Yes	Hint Answer
Password	Text(15)	Alphanumeric	Yes	Password

Table 1: User Table

4.5.2. POLICY

Field Name	Field Type	Data Type	Mandatory	Possible Values
Policy Id	Text(50)	Alphanumeric	Yes	Policy Id
Policy Name	Text(50)	Alphanumeric	Yes	Policy
Policy type	Text(10)	Alphanumeric	Yes	Policy type
Duration of years	Text(2)	Numeric	Yes	Duration of years
Company	Text(50)	Alphanumeric		
Initial Deposit	Text(10)	Numeric	Yes	Initial Deposit
User type	Text(10)	Alphanumeric	Yes	User type
Term amount	Text(15)	Numeric	Yes	Term amount
Interest	Text(2)	Numeric	Yes	Interest

Table 2: Policy Table

4.5.3. USERPOLICY

Field Name	Field Type	Data Type	Mandatory	Possible Values
SR No.	Numeric(10)	Numeric	Yes	SR No
User Id	Text(50)	Alphanumeric	Yes	User
Policy Id	Text(10)	Alphanumeric	Yes	Policy Id
Payment Status	Text(10)	Alphanumeric	Yes	Payment status

Table 3: UserPolicy

4.5.4. BILLDETAIL

Field Name	Field Type	Data Type	Mandatory	Possible Values
SR No	Numeric(10)	Numeric	Yes	SR No.
User Id	Text(50)	Alphabetic	Yes	User Id
User Name	Text(10)	Alphabetic	Yes	User Name
Payment Mode	Text(10)	NA	Male, Female	Payment Mode
Card No.	Text(10)	Numeric	Yes	Card No.
CVV	Numeric	Alphanumeric	Yes	CVV
Expiry	Numeric	Alphanumeric	Yes	Expiry
Password	Password(10)	Alphanumeric	Yes	Password

Table 4: Bill Detail

4.6. DATA FLOW DIAGRAM

4.6.1. CONTEXT LEVEL DFD

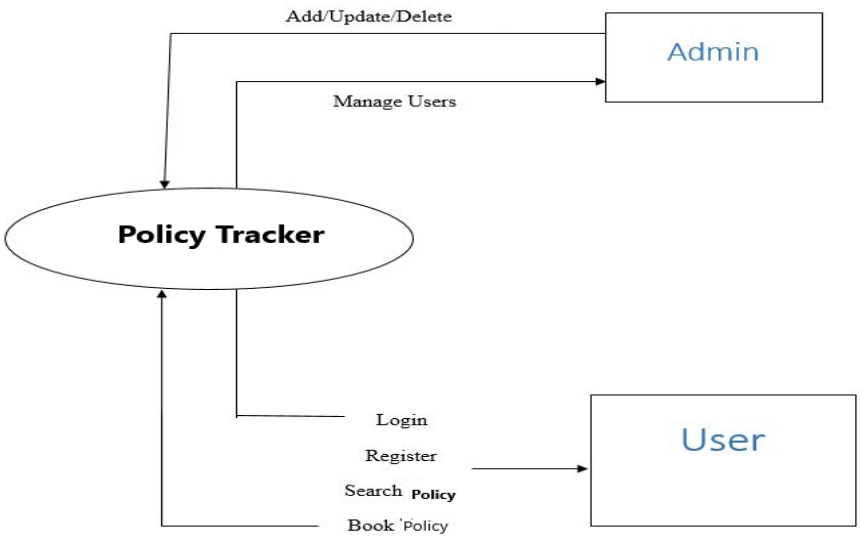


Figure 6: Context Level DFD

4.6. 2. FIRST LEVEL DFD

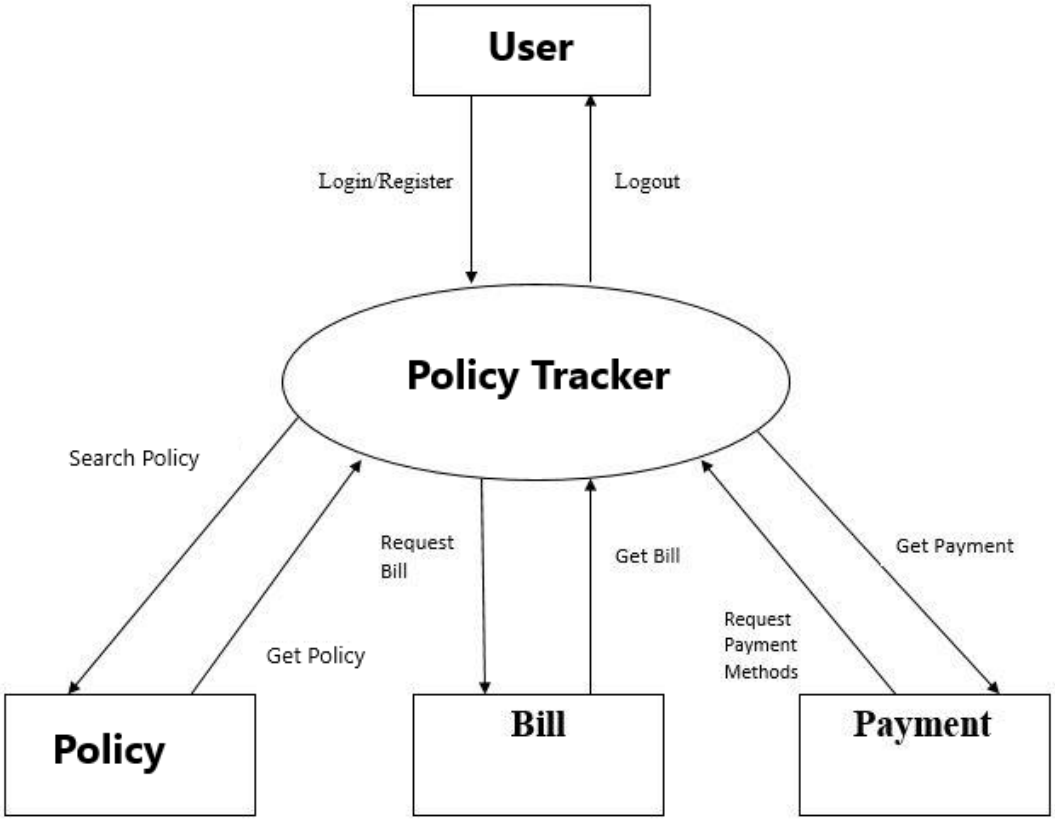


Figure 7:First level DFD

TEST PLAN

CHAPTER – 5

SYSTEM TESTING

It is the process to evaluate the system or its components in order to check if it is satisfying all requirements or not. This outcomes in expected ,actual and difference between their results that is , executing the system with the intent to identify any kind of gaps, errors or missing requirements.

5.1. TESTING STRATEGIES

Different levels of testing strategies are implemented in different phases of software development to ensure that our system does not contain any error.

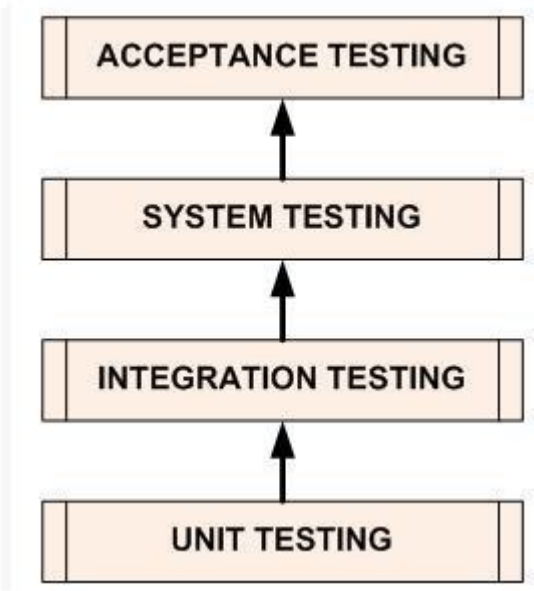


Figure 8: Testing Process

5.1.1. UNIT TESTING

It’s goal is to isolate every part of program and prove that the individual parts are fulfilling all requirements and functionalities.

5.1.2. INTEGRATION TESTING

It is the testing of combined parts of application to check if these parts function correctly with each other or not. It is done using two methods -

5.1.2.1. TOP DOWN TESTING

In this firstly highest-level modules are tested and then the lower-level modules are made to test.

5.1.2.2. BOTTOM-UP TESTING

It is done from smallest and lowest level modules and preceeding one at a time. After the testing of bottom level modules, next lower level ones are tested individually and these are then linked to the previously examined lower level modules. Firstly ,bottom-up testing is done, and then top-down testing.

5.1.3. SYSTEM TESTING

This testing tests the whole system. The whole application is rigorously tested.

5.1.4. ACCEPTANCE TESTING

It is used to check if the application meets the required specifications of the client. It is of two types.

5.1.4.1. ALPHA TESTING

It is the first stage of testing and it is performed amongst the teams. Unit testing, integration testing and system testing together are called alpha testing.

5.1.4.2. BETA TESTING

In beta testing, special customers test the application and they send their feedback to the project team. After getting back the feedback, the project team fixes the problems prior to releasing the software to the actual customers.

5.2 TESTING METHODS

5.2.1 WHITE BOX TESTING

It is a holistic investigation of Code's structure and logic. In order to perform white box testing, the tester needs to have knowledge of the internal working of code.

5.2.2 BLACK BOX TESTING

It is done without having any internal knowledge of the system. The tester is unaware of system architecture and can also not access the source code. The tester interacts with the user interface of the system by providing inputs and without having a knowledge of how and where the inputs are worked on.

5.3 VALIDATION

All the levels in the testing (unit integration, system) and methods (black box, white box) are implemented on our application successfully and the results obtained as expected.

5.4 LIMITATIONS

The execution time for support vector machine is more so that the user may not receive the result fast.

5.5 TEST RESULTS

The testing is done among the team members and by the end users. It satisfies the specified requirements and finally we obtained the results as expected.

TC01	On register page enter valid first name, last name, email, mobile and matching password.	The first name and last name must follow the pattern specified (no alphanumeric string), and email should end with "@something". Mobile number should be exactly 10 characters	After the validation is all successful, the actor would be forwarded to the log in page	Pass
TC02	On login page Enter valid Username and Password. Then submit.	The credentials should match with database. Otherwise it will give error message.	After matching username and password it will forward it to the next page.	Pass
TC03	On the bill page we can view the entire details of	Total amount is calculated based on	If the total amount	Pass

	user and tour details and the total amount that need to be paid	formula which uses number of days, pre estimated price of one place to another place and number of people.	displayed is correct user can click on proceed button to direct to next page of payment.	

Table 3:Test Case

RESULTS AND PERFORMANCE ANALYSIS

CHAPTER – 6

USE CASE

6.1 USE CASE 'ADMIN REGISTRATION'

6.1.1 USE CASE ATTRIBUTES

Use Case Description:

This use case deals with the capture of admin details. The 'admin' here shall be the operator of the system and will be keying in the user details and policy details.

Scope:

Admin registration

Actors:

Admin – the operator

Trigger*:*

Click 'Submit' button in the 'Admin Registration' page

Pre-Condition*:*

Admin being able to access the login page & get redirected to the 'Admin Registration' page upon click of 'New User ?' link on the login page.

Post Condition*:*

Admin is in the registration page & submit details

Flow of Events:

Admin at login page → Click ‘New User ?’ link → Admin at registration page→ Admin is in the registration page → Admin details are submitted and added onto the database

Primary Scenario:
A new admin – is able to click ‘New User ?’ link and able to provide his details and get registered in the system.

6.1.2 BUSINESS RULES

Business rules should be defined using the following attributes: -

- When the admin clicks on the register link, it should re-direct to registration form.
- Admin needs to fill some of the basic attributes/fields as mentioned below in requirement: User Id, First Name, Last Name, DOB, Gender, Contact Number, Address, E-Mail, Qualification, Salary per month, PAN number, Employer Type, Employer, Hint Question, Hint Answer, Password.
- Clicking ‘Register’ should validate the datatype constraints for each field
- Post-successful field level validation, save the information in the database
- Upon saving the information in the database, display the message ‘Your have successfully registered’.

.

6.1.3. UI REQUIREMENTS

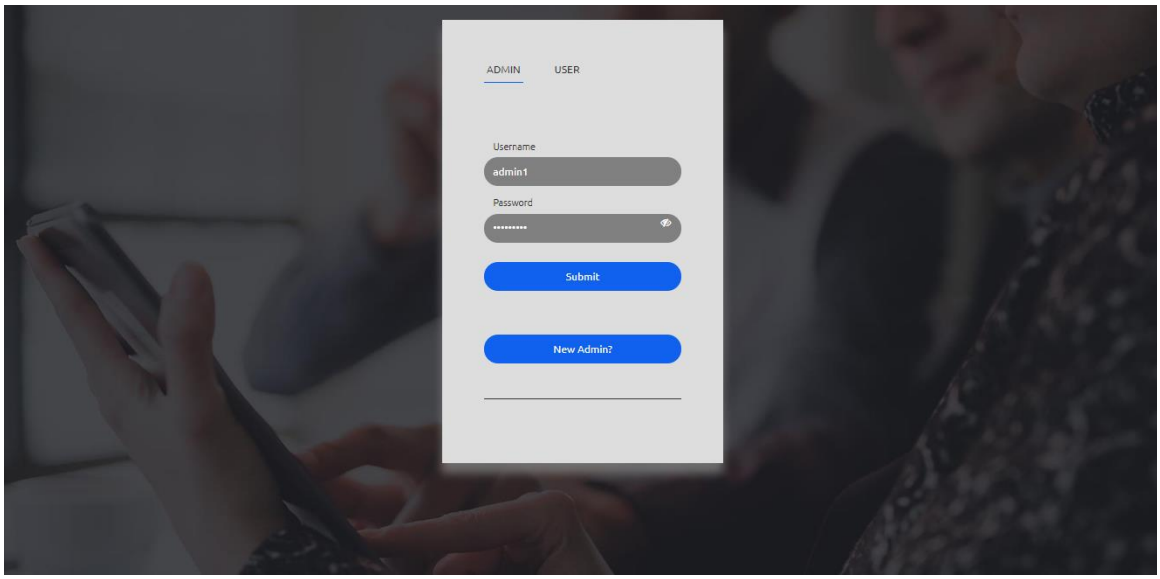


Figure 9: Login Page

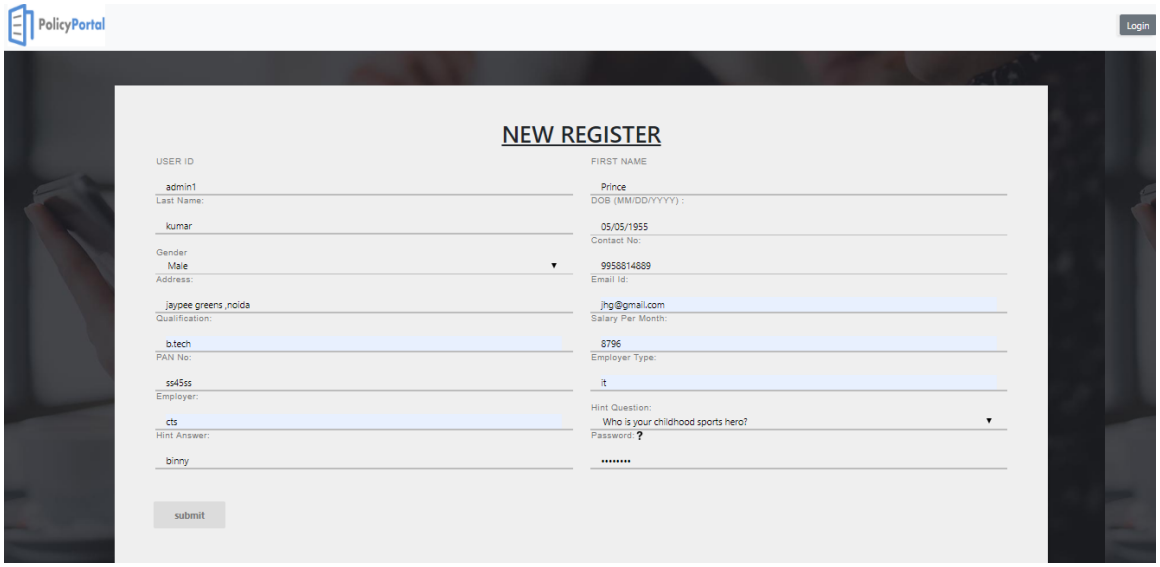


Figure 10: Registration Page

6.1.4. UI FIELD VALIDATIONS

Please refer to the below requirements for field level validations:

- All fields are mandatory.
- Password should have maximum 15 alphanumeric, space & can contain special characters (ex.!,@,#,%,*,& etc.)
- The Email ID format must be checked.
- Phone Number must be of 10 digits' length.
- First Name, Last Name should contain only alphabets.
- Gender should be chosen from the dropdown options.
- PAN, User Id should be alphanumeric maximum.

6.2. USE CASE 'ADMIN CREDENTIAL AUTHENTICATION'

6.2.1. USE CASE ATTRIBUTES

Use Case Description:

This use case deals with the authentication of the admin credentials. The 'admin' here shall be the operator of the system and will be keying in the user information into the system.

Scope:

Admin credentials authentication

Actors:

Admin – the operator

Trigger:

Click ‘Submit’ link, after keying in ‘Username’ & ‘Password’ field.

Pre-Condition:

Admin being able to access the login page

Post Condition:

Admin is in the Admin Home Page.

Flow of Events:

Admin at login page → Key in ‘Username’ & ‘Password’ field → Admin credentials are validated → Admin Home page is displayed.

Primary Scenario:
A registered admin – is able click ‘Submit’ link, after keying in ‘Username’ & ‘Password’ field and get his credentials authenticated with the existing database entry.

Alternative Scenario:
A registered admin – is able click ‘Submit’ link, after keying in ‘Username’ & ‘Password’ field and unable to get his credentials authenticated. The admin is presented with relevant error messages: Invalid username or password and redirected back to the login page.

6.2.2. BUSINESS RULES

Business rules should be defined using the following attributes :-

- A registered admin – is able click ‘Submit’ link, after keying in ‘Username’’ & ‘Password’ field and get his credentials authenticated with the existing database entry.

6.2.3. UI REQUIREMENTS

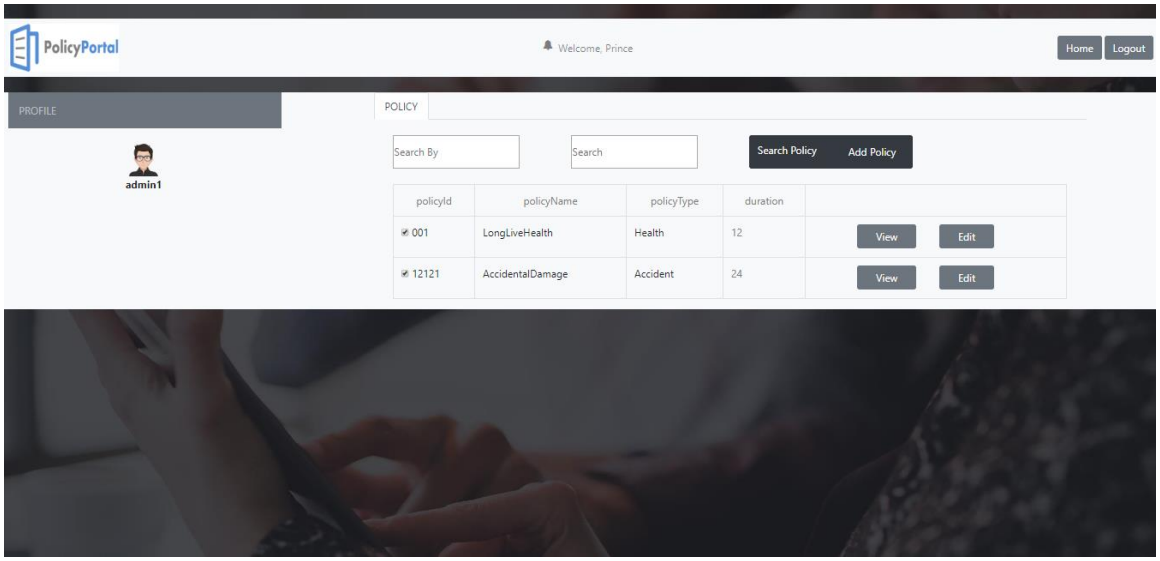


Figure 11 : User Home

6.3. USE CASE 'ADMIN HOME'

6.3.1. USE CASE ATTRIBUTES

Use Case Description:

This use case deals with the management of policies. The 'admin' here shall be the operator of the system and will be keying in the policy details.

Scope:

- Policy Registration

Actors:

Admin – the operator

Trigger:

Admin should get redirected to the 'Admin Home' page when the admin enters valid credentials.

Pre-Condition:

Admin should be able to get redirected to the 'Admin Home' page upon clicking the 'Submit' button on the Login page.

Post Condition:

Admin is in the Admin Home page & can view all the policies listed.

Flow of Events:

Admin at login page → Enter valid credentials → Click 'Submit' button → Admin in the admin home page → All the policies present in the database are displayed.

Primary Scenario:

Admin – is able to click 'Submit' button and is able to land on the admin home page and view all the policies

6.3.2 BUSINESS RULES

Business rules should be defined using the following attributes: -

- When the admin clicks on the submit button after entering the valid credentials, admin lands on the admin home page.
- Admin home page contains the list of all the policies added by the admin in the database.
- Admin home page also consists of Search Policy and Add Policy functionality.
- All the policies listed contains two buttons.
- One button is 'View' and the other button is 'Edit'.

6.3.3. UI REQUIREMENTS

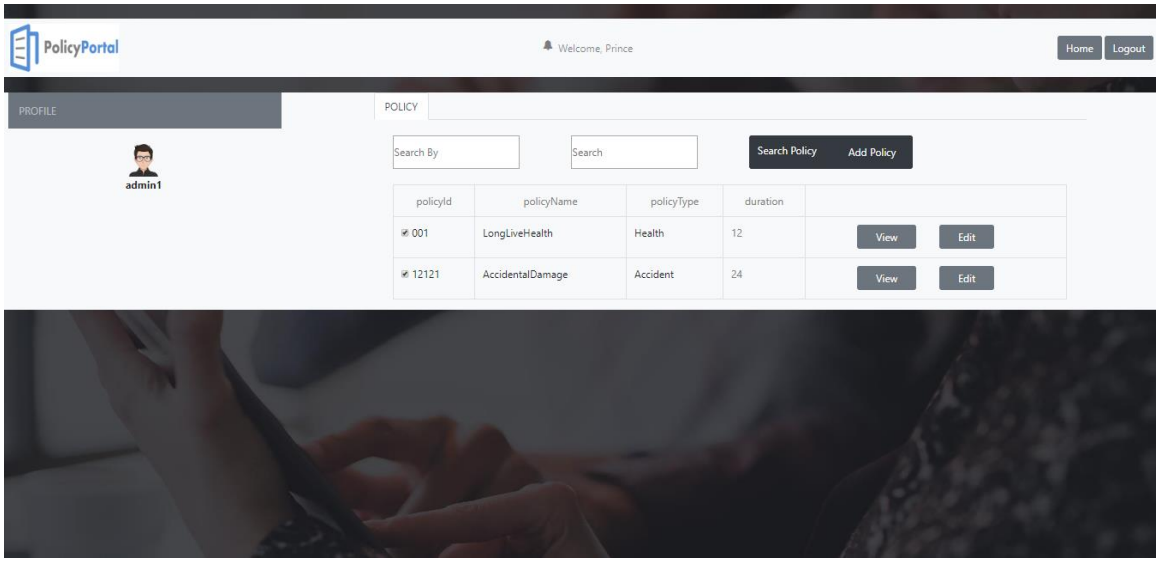


Figure 12: User View Details

6.4.a USE CASE 'Add Policy on Admin Home Page'

6.4.1. USE CASE ATTRIBUTES

Use Case Description:

This use case deals with the ability to add new policies in the database. The ‘admin’ here shall be the operator of the system and will be keying in the policy details.

Scope:

Add Policy

Actors:

Admin – the operator

Trigger:

Click ‘Add Policy’ button in the ‘Admin Home’ page

Pre-Condition:
Admin being able to access the ‘Admin Home’ page by entering the valid credentials. Admin being able to reach ‘Admin Home’ page can register the details of new policies by clicking on the ‘Add Policy’ button. Once the details have been successfully entered, admin is given the option ‘Submit’ to save the entered details in the database and land back to Admin Home Page.

Post Condition:

Admin is successfully redirected to the Admin Home page and the new policy added is reflected on the Admin Home page.

Flow of Events:

Admin at Login Page→Enter Valid credentials → Click on ‘Submit’ button → Admin at ‘Admin Home’ page’ → Click ‘Add Policy’ button → Policy Form details are submitted and added onto the database upon clicking the ‘Submit’ button -> Admin redirected to ’Admin Home’ page.

Primary Scenario:
Admin is able to add new policy and record it in the database, which would then be reflected back to the user.

Business Rules:

Business rules should be defined using the following attributes: -

- When the admin clicks on the ‘View’ button of a particular policy, the admin will be redirected to ‘View Policy’ page, where all the details of that particular policy could be seen on the screen.
- When the admin clicks on the ‘Edit’ button of a particular policy, the admin will be redirected to ‘Edit Policy’ page, where all the details of that particular policy could be seen on the screen and the fields that could be edited are reflected in a light color.

- Other non-edited fields cannot be accessed.
- On clicking the submit button and post-successful field level validation, the information is again updated in the database
- A link for home page is displayed with a message ‘Success’.

6.4.2. UI REQUIREMENTS

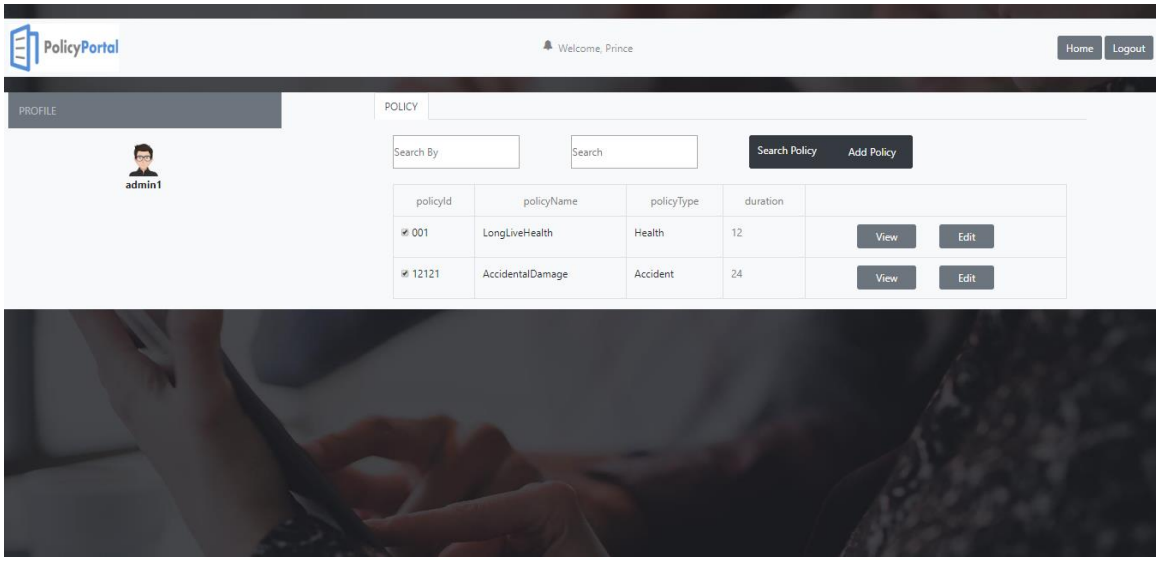


Figure 13: Search Page

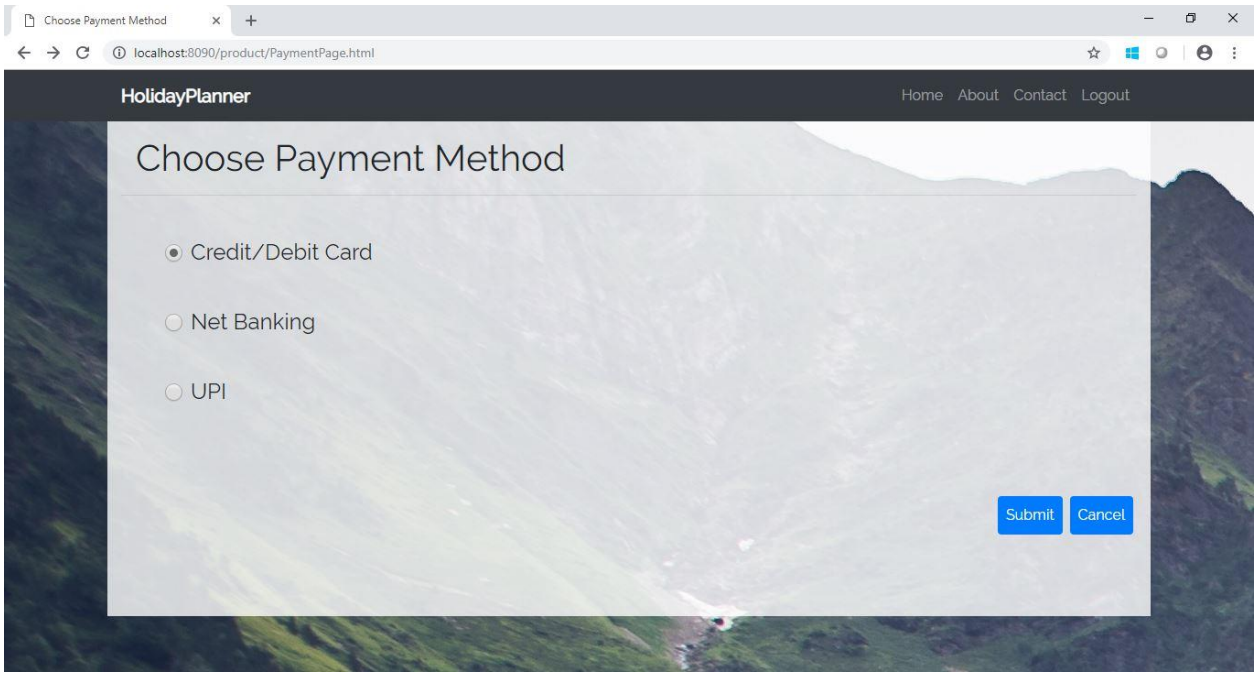


Figure 14: Bill Payment Page

6.4.b USE CASE ‘User Registration’

6.4.1. USE CASE ATTRIBUTES

Use Case Description:

This use case deals with the capture of user details. The ‘user’ here shall be able to view all the policies and buy them.

Scope:

User registration

Actors:

User – the operator

Trigger:

Click ‘Submit’ button in the ‘User Registration’ page

Pre-Condition:

User being able to access the login page & get redirected to the ‘User Registration’ page upon click of ‘New User ?’ link on the login page.

Post Condition:

User is in the registration page & submit details

Flow of Events:

User at login page → Click ‘New User ?’ link → User at registration page→ Admin is in the registration page → User details are submitted and added onto the database

Primary Scenario:

A new User – is able to click ‘New User ?’ link and able to provide his details and get registered in the system.

Business Rules:

Business rules should be defined using the following attributes: -

- When the User clicks on the register link, it should re-direct to registration form.
- User needs to fill some of the basic attributes/fields as mentioned below in requirement: User Id, First Name, Last Name, DOB, Gender, Contact Number, Address, E-Mail, Qualification, Salary per month, PAN number, Employer Type, Employer, Hint Question, Hint Answer, Password.
- Clicking 'Register' should validate the datatype constraints for each field
- Post-successful field level validation, save the information in the database
- Upon saving the information in the database, display the message 'Your have successfully registered'.

6.4.2. UI REQUIREMENTS

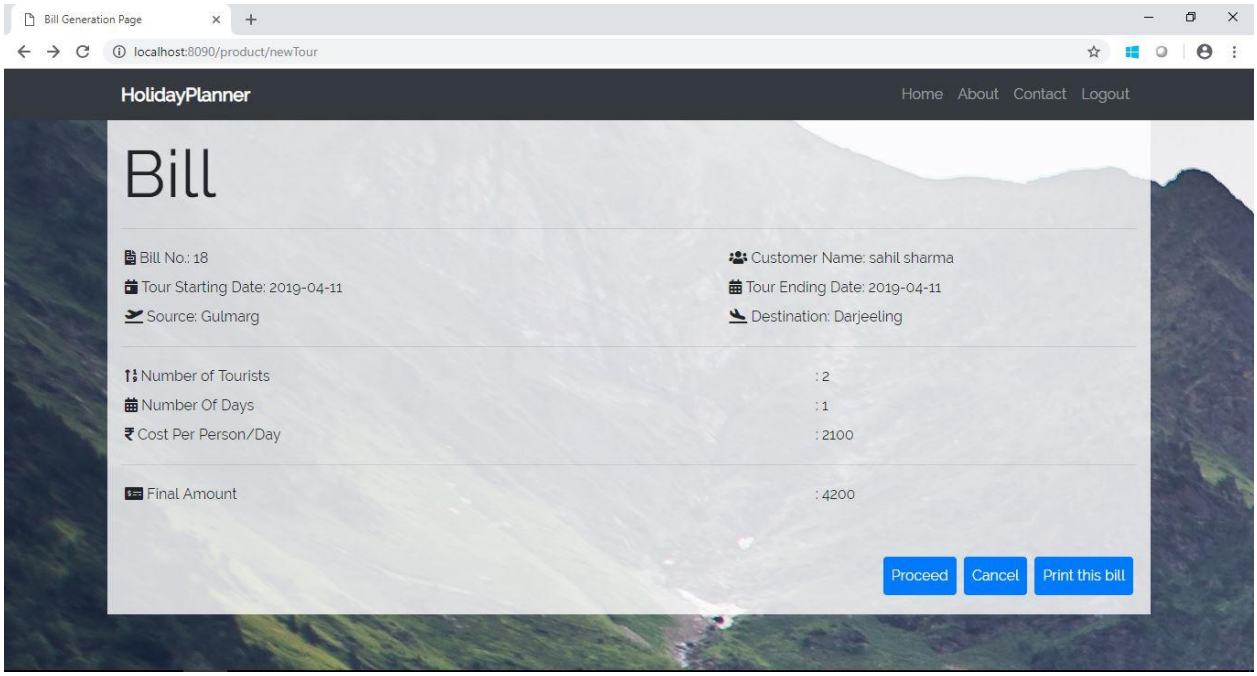


Figure 15: Bill Page

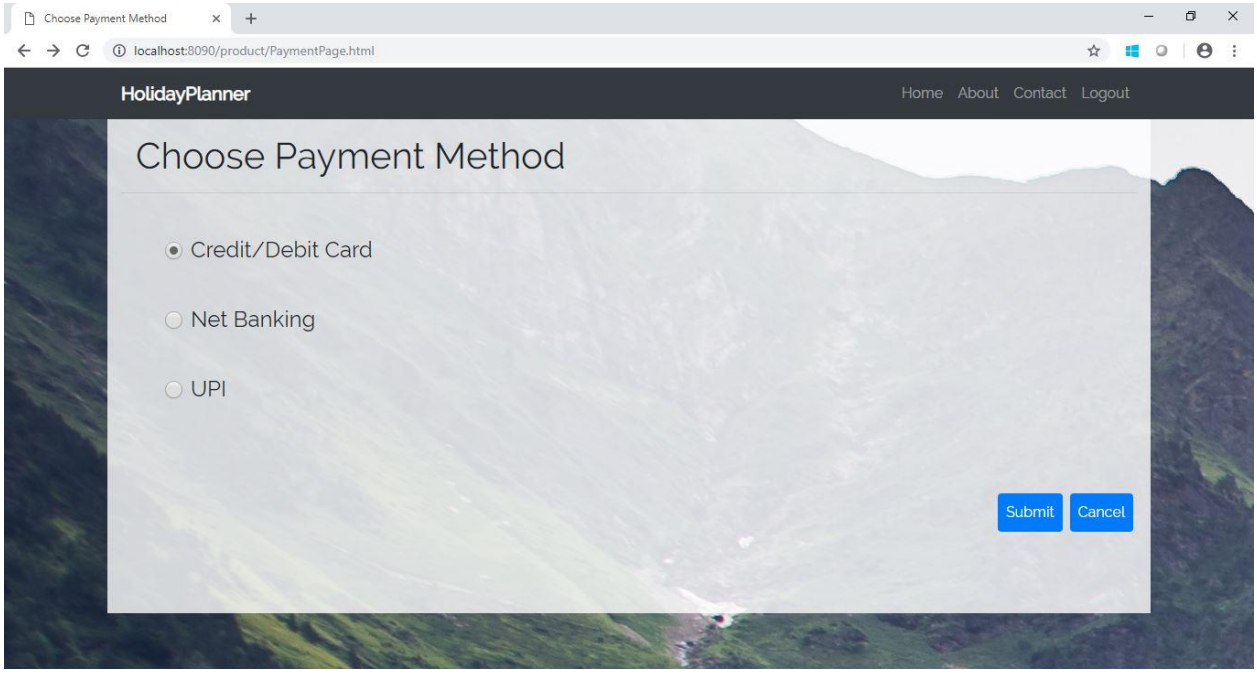


Figure 16: Bill Payment Page

6.1.4. UI FIELD VALIDATIONS

Please refer to the below requirements for field level validations:

- All fields are mandatory.
  Password should have maximum 15 alphanumeric, space & can contain special characters (ex.!,@,#,%,*,& etc.)
- The Email ID format must be checked.
- Phone Number must be of 10 digits' length.
- First Name, Last Name should contain only alphabets.
- Gender should be chosen from the dropdown options.
- PAN, User Id should be alphanumeric maximum.

6.3. USE CASE 'User Credential Authentication'

6.3.1. USE CASE ATTRIBUTES

Post Condition:

User is in the User Home Page.

Flow of Events:

User at login page → Key in 'Username' & 'Password' field → User credentials are validated → User Home page is displayed.

Primary Scenario:

A registered user – is able click ‘Submit’ link, after keying in ‘Username’ & ‘Password’ field and get his credentials authenticated with the existing database entry.

Alternative Scenario:

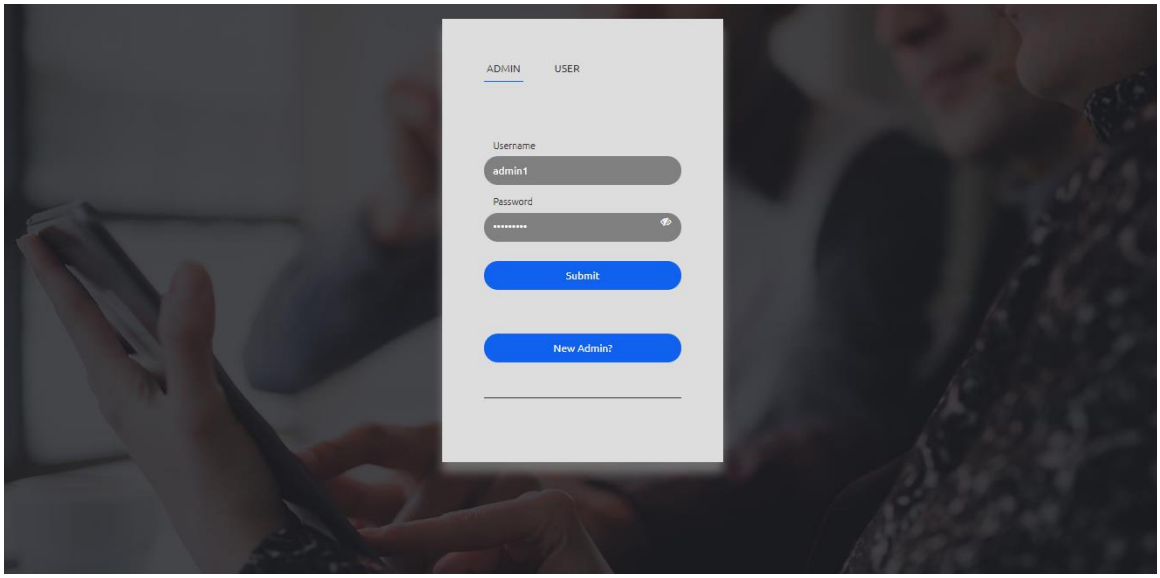
A registered user – is able click ‘Submit’ link, after keying in ‘Username’ & ‘Password’ field and unable to get his credentials authenticated. The user is presented with relevant error messages: Invalid username or password and is redirected back to the login page.

7.3.2 BUSINESS RULES

Business rules should be defined using the following attributes :-

- A registered user – is able click ‘Submit’ link, after keying in ‘Username’’ & ‘Password’ field and get his credentials authenticated with the existing database entry.

6.3.3. UI REQUIREMENTS



6.3. USE CASE ‘User Home’

6.3.1. USE CASE ATTRIBUTES

Use Case Description:

This use case deals with the ability of the system to display all the policies and let user purchase them. The ‘user’ here shall be the operator of the system and will be keying in the payment details.

Scope:

User Home

Actors:

User – the operator

Trigger:
User should get redirected to the 'User Home' page when the user enters valid credentials.

Pre-Condition:

User should be able to get redirected to the 'User Home' page upon clicking the 'Submit' button on the Login page.

Post Condition:

User is in the User Home page & can view all the policies.

Flow of Events:

User at login page → Enter valid credentials → Click 'Submit' button → User in the user home page → All the policies present in the database are displayed.

Primary Scenario:

User – is able to click 'Submit' button and is able to land on the user home page and view all the policies and buy them

6.3.2 BUSINESS RULES

Business rules should be defined using the following attributes: -

- When the user clicks on the submit button after entering the valid credentials, user lands on the user home page.
- User home page contains the list of all the policies added by the admin in the database.
- User home page also consists of Search Policy and Buy Policy functionality.
- All the policies listed contains two buttons.
- One button is 'View' and the other button is 'Buy'.
- When the user clicks on the 'View' button of a particular policy, the user will be redirected to 'View Policy' page, where all the details of that particular policy could be seen on the screen

6.3.3. UI REQUIREMENTS

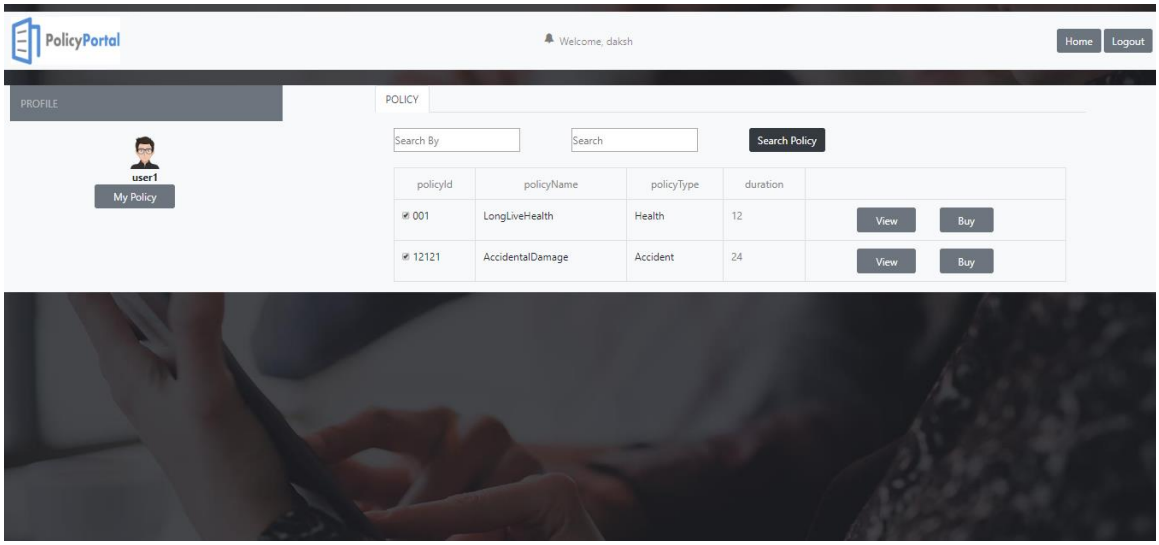


Figure 17: User Details

6.3. USE CASE 'Policy Payment'

6.3.1. USE CASE ATTRIBUTES

Use Case Description:

This use case deals with the ability of the user to buy the policy and pay for it via different payment methods. The 'user' here shall be the operator of the system and will be keying in the policy payment details.

Scope:

- Policy Payment

Actors:

- User – the operator

Trigger:

Click 'Buy' button on the 'User Home' Page for the policy user wants to buy.

Pre-Condition:

User being able to access the 'My Policy' button and check the payment status by clicking on the 'Payment' button. If the pop up occurs with the message 'You have already paid for

the policy' then this shows the user had bought the policy again. Else the user will be redirected to the payment page. As soon as user clicks on 'Buy' in 'User Home' page, the entry will be removed from the page and will be only visible in 'My Policy' page.

Post Condition:

On clicking the 'Payment' button user is redirected to the 'Payment' page which consists of multiple payment methods like Credit/Debit Card, Net Banking, UPI.

Flow of Events:

User at login page → Enter valid credentials → Click 'Submit' → User at User Home page → Click 'Buy' button → User at Payment Page which displays different payment options→ User selects one of the payment options and accordingly buys the policies.

Primary Scenario:

User being able to access the 'My Policy' button and check the payment status by clicking on the 'Payment' button. If the pop up occurs with the message 'You have already paid for the policy' then this shows the user had bought the policy again. Else the user will be redirected to the payment page.

6.3.2 BUSINESS RULES

Business rules should be defined using the following attributes: -

- When the user clicks on the 'Buy' button, it should re-direct to the 'Payment' page.
- Where user can choose the payment method and pay bill.
- Post-successful payment, the corresponding policy status is changed from pending to paid.
- Thereafter we redirect User to 'User Home' page.

6.3.3. UI REQUIREMENTS

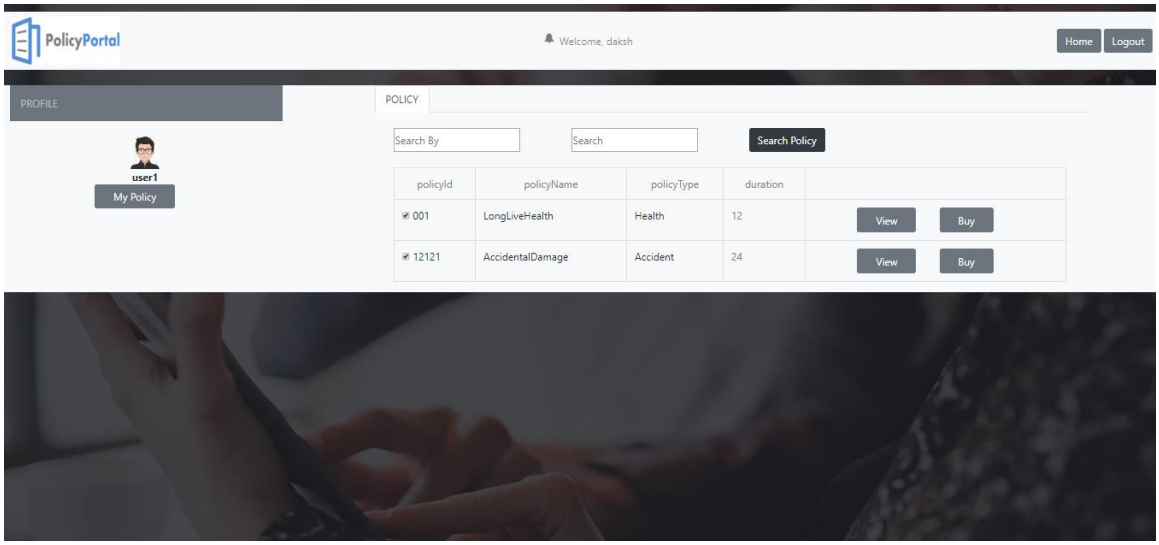


Figure 18: User Details

CHAPTER – 7

IMPLEMENTATION

PROCESS MODEL USED

Spiral model

It is an evolutionary software process model which binds the iterative nature of prototyping to features of the linear sequential model. It also provides fast development of incremental software version.

The development of the software is done in a series of incremental releases.

The framework activities in which spiral model is divided are known as task regions.

It consists of the following 6 task regions:

- **Planning:** It concits of tasks like define resources & timeline & another project related information.
- **Customer Communication:** Used to establish reliable communication between developer and the customer.
- **Engineering:** It is used to build one or more application representation.
- **Customer evaluation:** These are used to obtain customer feedback based on evalution.
- **Construction & release:** It is used to construct, test, install & provide user support
- **Risk analysis:** It required to assess risk factors.

7.1. CONVERSION PLAN

To make this project live, i.e., to build application file for the project followed:

- Installation of Eclipse IDE.
- Select a server at which you will host your web application.
- Once Eclipse is set, you need to create the maven project and select the suitable web-app.
- Select the Database which will serve the purpose of the application accordingly.
- We need to give the right to the admin who could change the database table and details.
- After all the business logic Is successfully written and implemented. The project is live!

7.2 POST IMPLEMENTATION OF PROJECT AND MAINTAINANCE

PIR is the Post Implementation Review that is conducted after the project is completed. It's purpose is to examine how successfully the project requirements have met and how effectively the project management practices were to keep the project on track.

In our project all the objectives have met the requirements and it is more affective as user wants. According to the user requirements the project functionality and objectives are made according to this. It has been observed that the systems which are easy to use, require less manpower, save data and are well received by the customers.

CONCLUSION

CHAPTER – 8

PROJECT LEGACY

8.1. CURRENT STATE OF PROJECT

The current status of our project is that all modules like login, home page, add new candidate page, edit details page, view all candidate page of the project are completed and their design,
coding and testing are done. The application is completely developed and tested.

8.2. REMAINING AREAS OF CONCERN

There are still, after a lot of efforts, the areas of concern in the project. Once the user is registered there is no way to edit the user details. In future release we can add this feature to the project.

8.3. TECHNICAL AND MANAGERIAL LESSONS LEARNT

I learnt many things during the project development

- Working with the Eclipse IDE.
- Working with server-side tasks.
- Style website using Bootstrap Framework.
- Connect Database with the web application.
- Creating and managing databases using MYSQL and using it in Spring MVC.
- Working in a team and co-ordination among them.
- Problem Analysis and problem solving with the team mates.

CHAPTER - 9

DEMONSTRATION/SCREENSHOTS

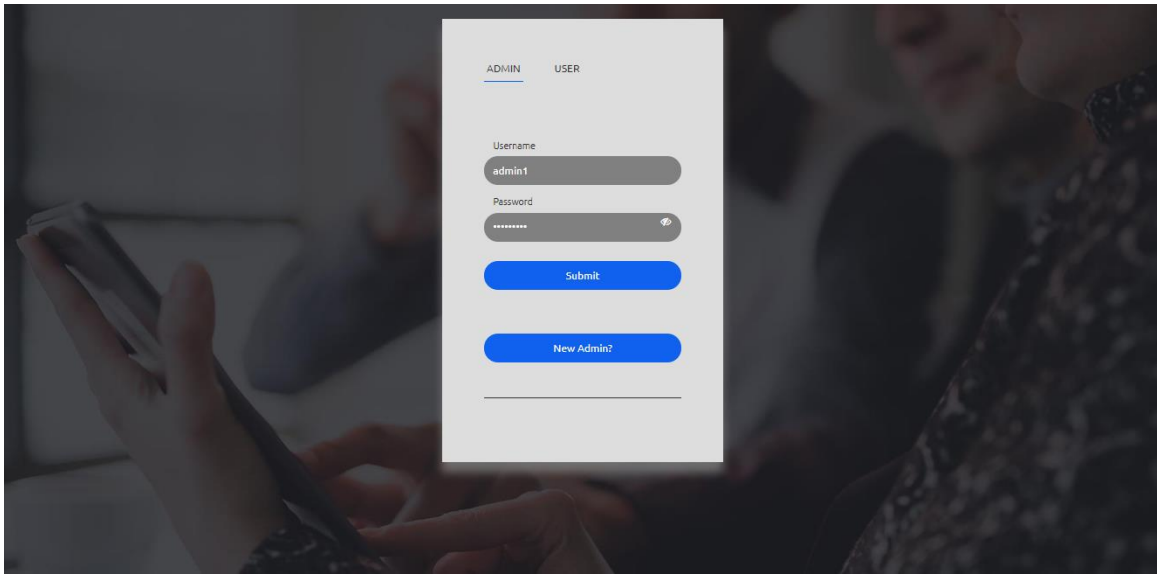


Figure 19: Admin Login Page

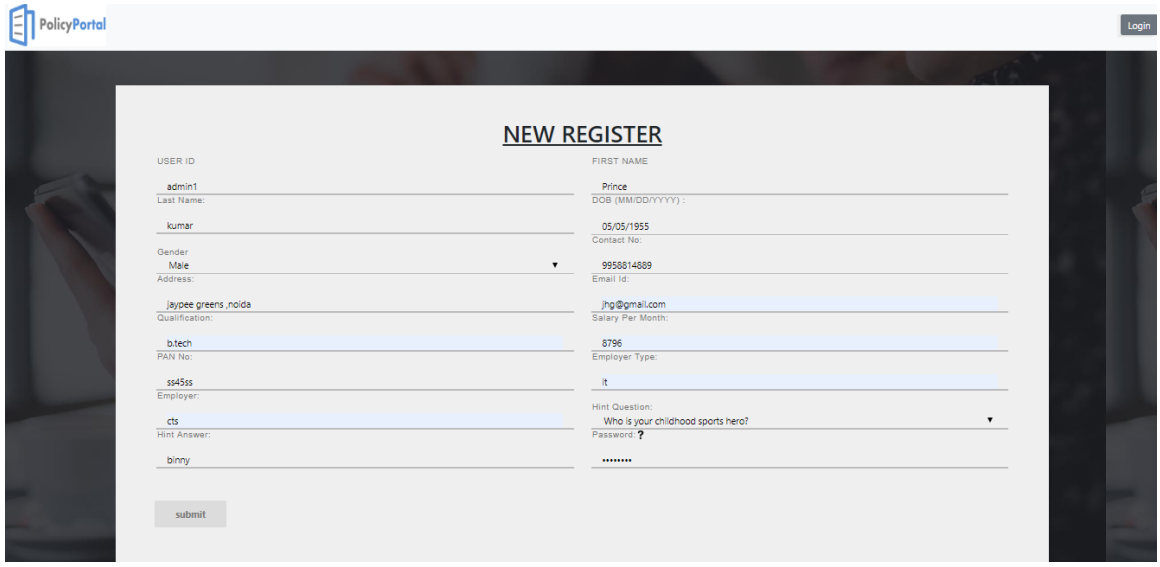


Figure 20 : Registration Page

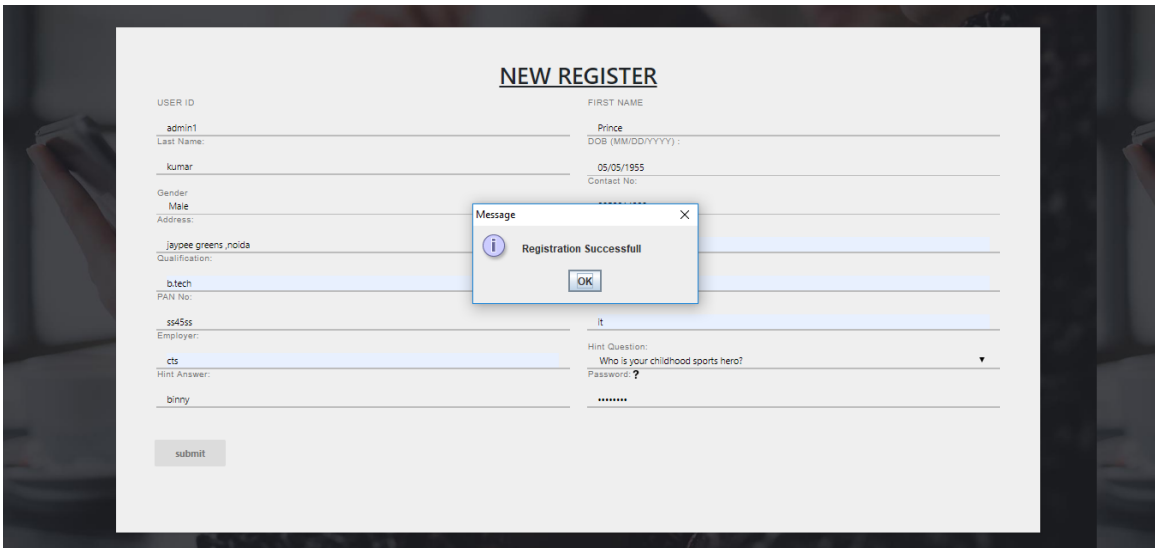


Figure 21 : Registration Successful Pop-up

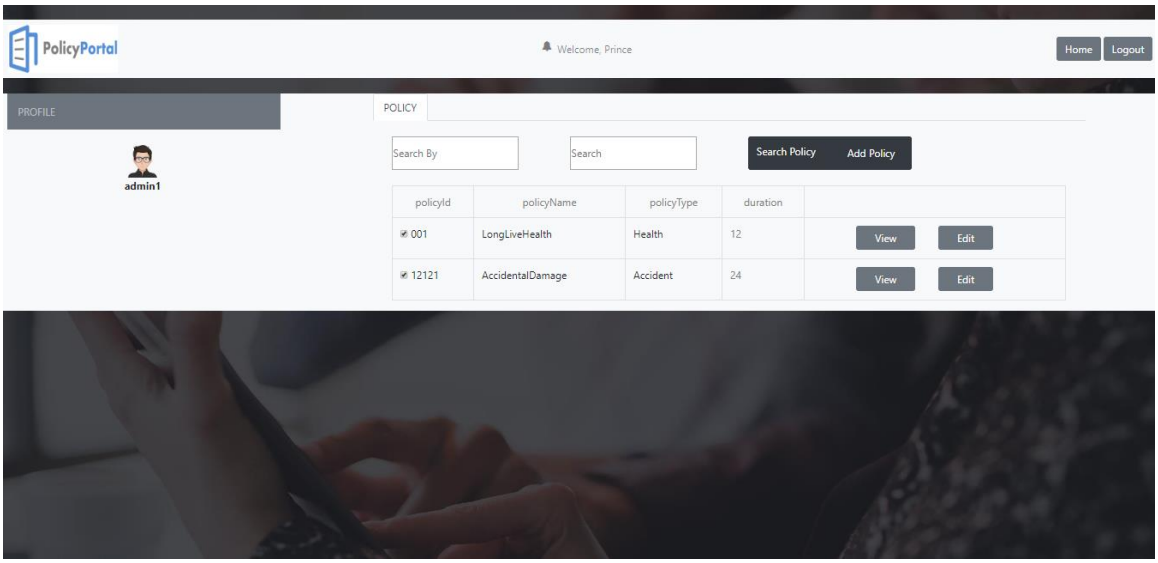


Figure 22 : Admin Home Page

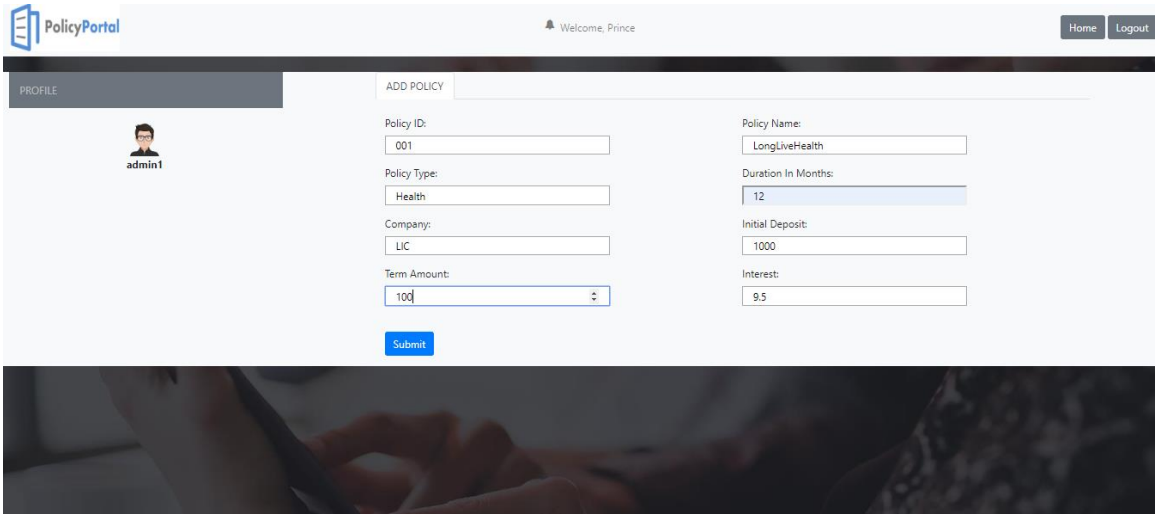


Figure 23 : Add Policy Page

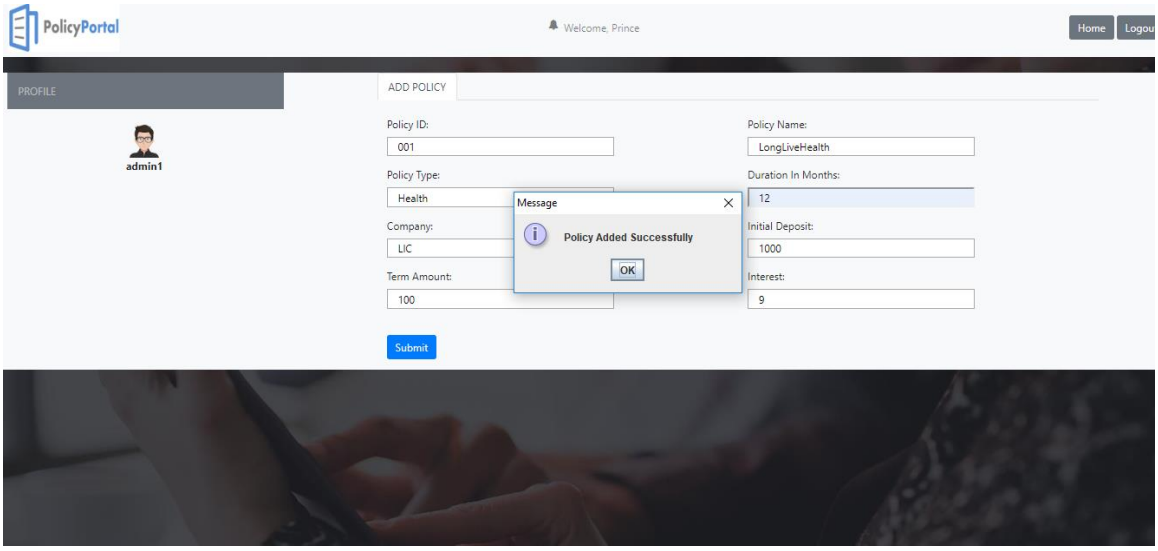


Figure 24 : Policy Addition Successful

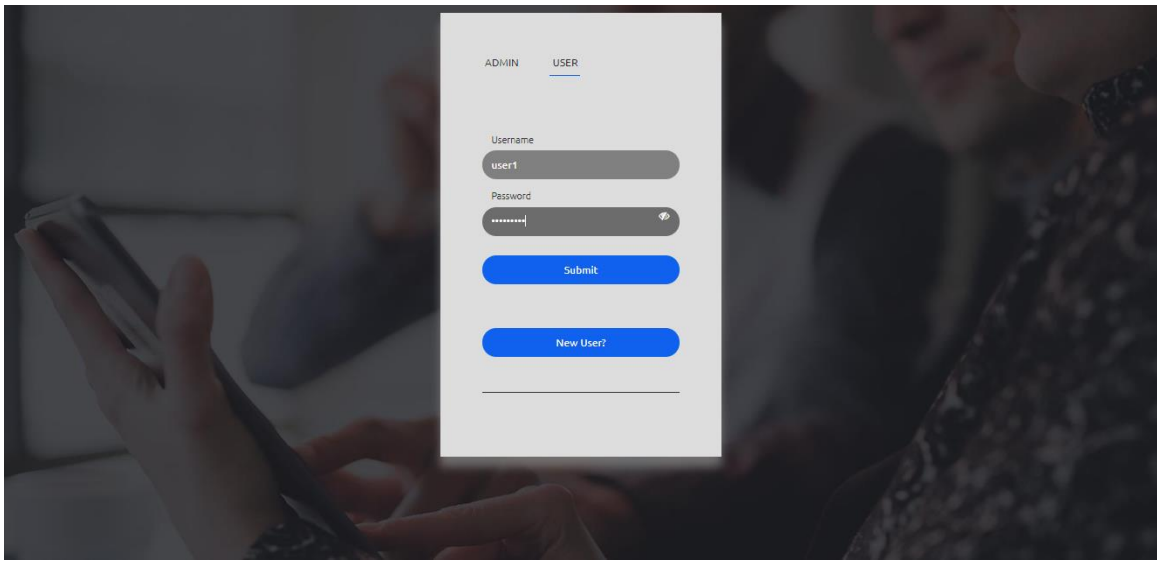


Figure 25 : User Login

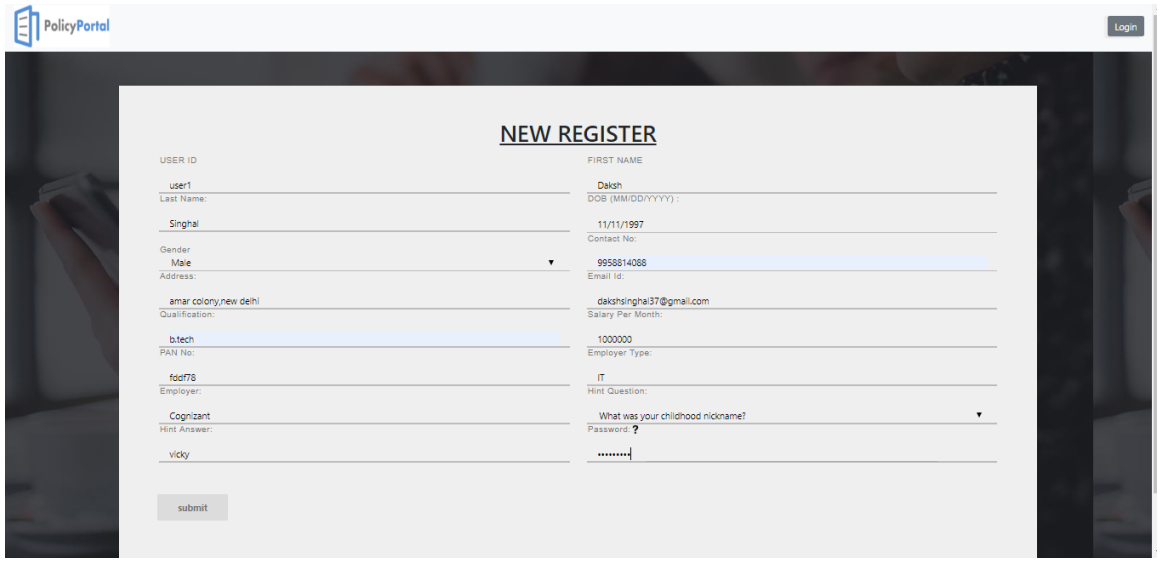


Figure 26 : User Registration Page

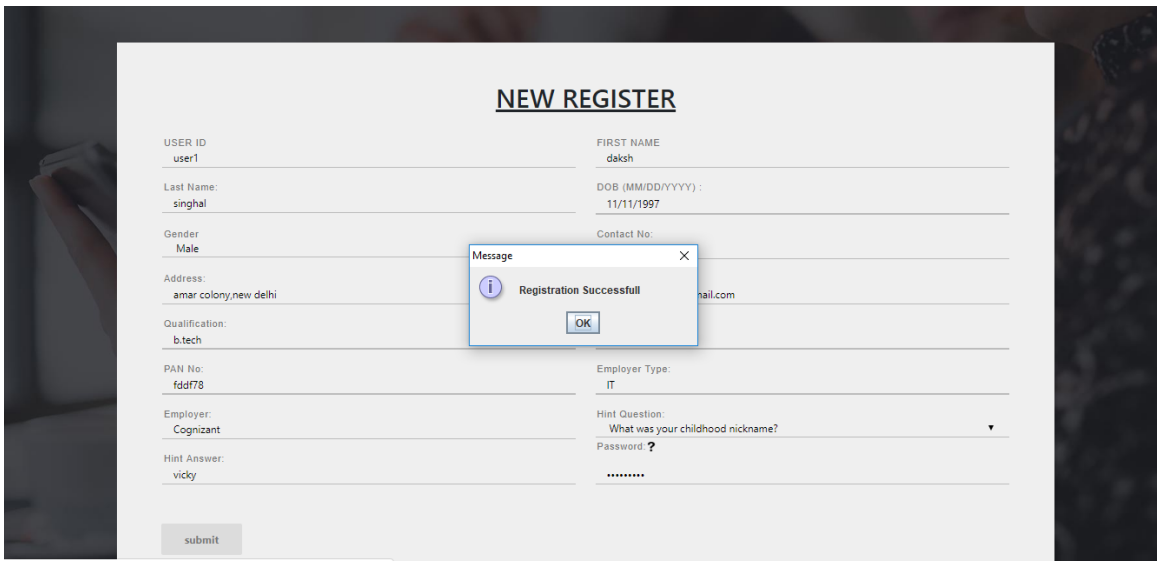


Figure 27 : User Registration Successful Pop-up

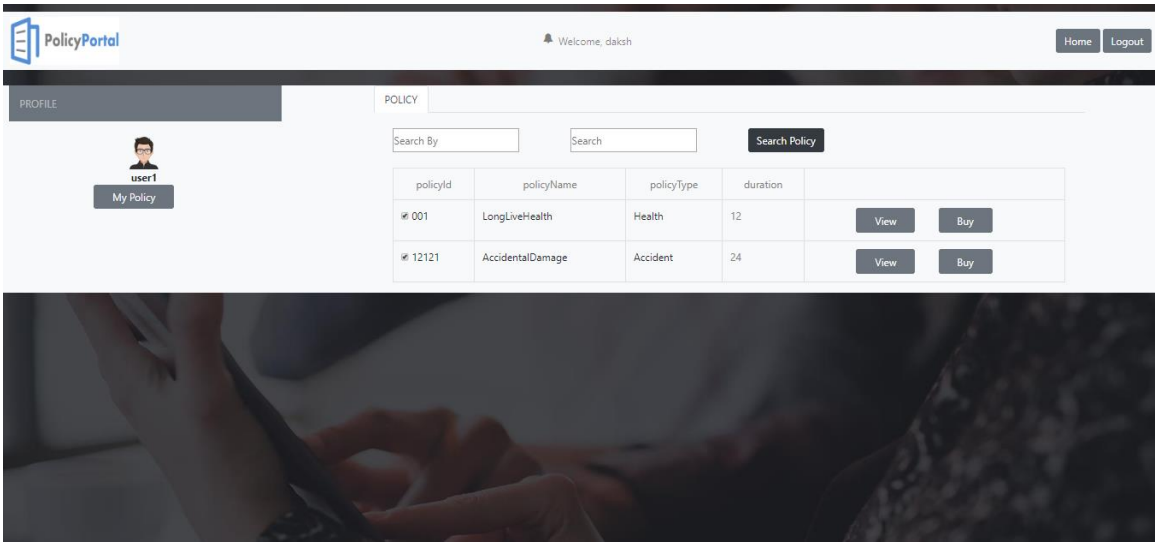


Figure 28 : User Home Page

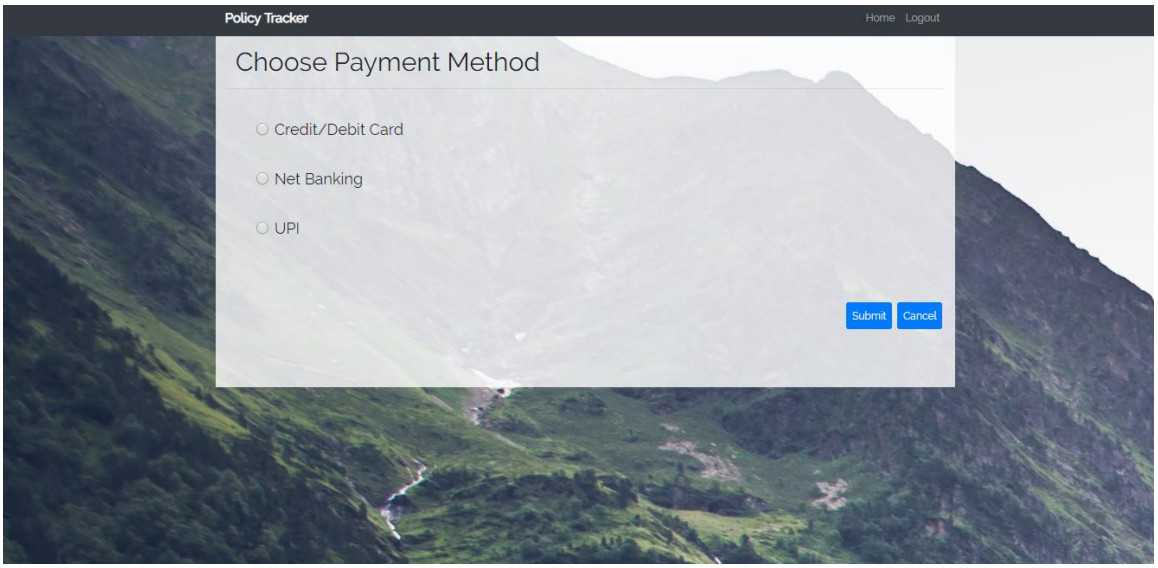


Figure 29 : Payment Methods Page

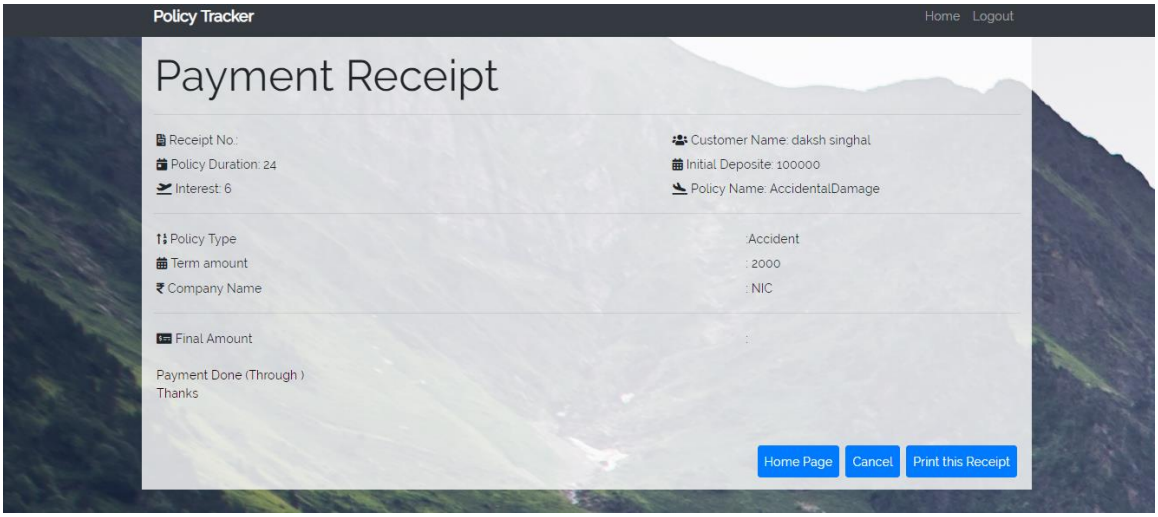


Figure 30 : Payment Successful Page

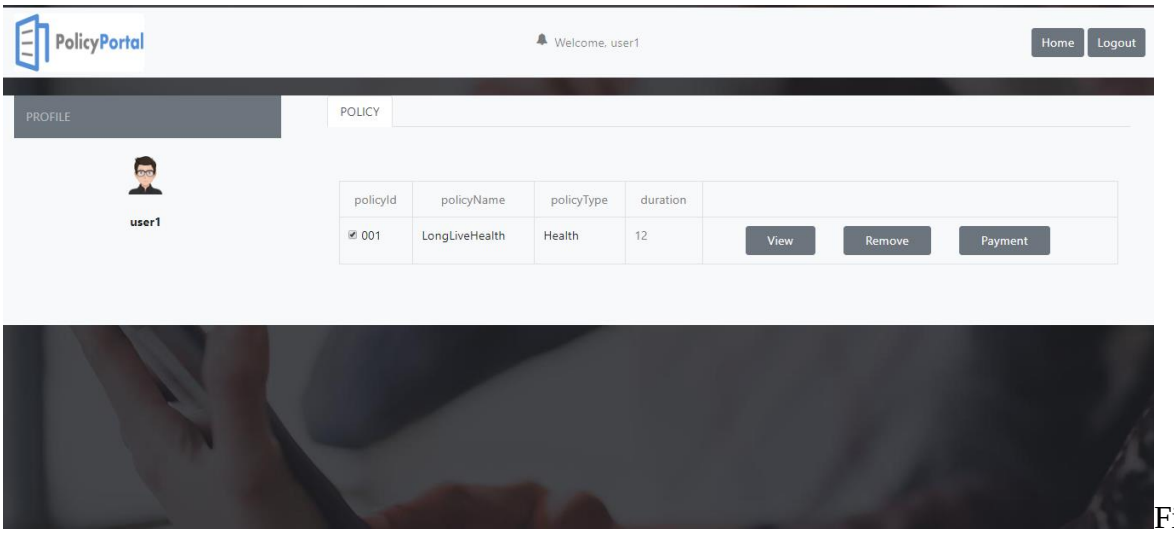


Figure 31 : My Policy Page

BIBLIOGRAPHY

[1] https://www.tutorialspoint.com

[2] https://getbootstrap.com/docs/4.1/getting-started/introduction/

[3] https://www.javatpoint.com/

[4] https://www.w3schools.com/bootstrap4/