CAPSTONE (Major) Project Report

on

Policy Tracker- Policy Management System

at

Cognizant Technology Solution

Submitted by

Prince

Registration Number – 17MCA10005

Guided by

Mrs. Shilpa Mahajani
Project Manager
Cognizant Technology Solution
(External Guide)

Dr. Pushpinder Singh Patheja
Associate Professor
VIT Bhopal University
(Internal Guide)

Submitted in partial fulfillment of the requirement for
The degree of

"Master of Computer Applications"

Submitted to

School of Computing Science and Engineering

VIT Bhopal University

Bhopal (MP) – 466 114

May 2019

VIT BHOPAL UNIVERSITY, M P - 466114

SCHOOL OF COMPUTING SCIENCE AND 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 VIT University, Bhopal.

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.

Prince

(17MCA10005)

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

Date: _____

Internal Guide Name	External Guide Name
Dr. Pushpinder Singh Patheja	Mrs. Shilpa Mahajani
Designation: - Associate Professor	Project Manager

VIT UNIVERSITY BHOPAL, M P – 466114

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the work embodied in this Capstone Project Report entitled **"Policy Tracker- Policy Management System"** has been satisfactorily completed by **Mr. Prince** Registration No 17MCA10005 in the School Computing Science and Engineering at VIT University, Bhopal. This work is a bonafide piece of work, carried out under my/our guidance in the Organization "**Cognizant Technology Solution**" for the partial fulfilment of the degree of Master of Computer Application.

Mrs. Shilpa Mahajani

Project Manager

Cognizant Technology Solution

Under the Guidance of

Mr. Abhinandan Chivate

Trainee and Mentor

Forwarded by

Dr. Baseera

Program Chair

VIT Bhopal University

Approved by

Dr Raju Shanmugam

Professor & Dean

VIT Bhopal University

VIT UNIVERSITY BHOPAL, M P – 466114

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

ACKNOWLEDGEMENT

The internship opportunity I had with **Cognizant Technology Solutions** was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to the **Ms. Shilpa Mahajani** at CTS, Pune who in spite of being extraordinarily busy with her duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

Prince
(17MCA10005)

TABLE OF CONTENTS

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 to 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.

The purpose of this document is to systematically capture requirements for the project and the 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. PROFILE OF THE PROBLEM. RATIONALE/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 DEFINITION

This product is developed to solve the 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

A feasibility study is an analysis used in measuring the ability and likelihood to complete a project successfully including all relevant factors. It must account for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it.

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.

CHAPTER - 3

SOFTWARE REQUIREMENT ANALYSIS

3.1. TECHNOLOGIES

3.1.1. JAVA

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. Java is guaranteed to be Write Once, Run Anywhere.

Java is −

- Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- Architecture-neutral − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- Portable − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

- Robust − Java tries to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- Multithreaded − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
- Interpreted − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
- High Performance − With the use of Just-In-Time compilers, Java enables high performance.
- Distributed − Java is designed for the distributed environment of the internet.
- Dynamic − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

3.1.2. SPRING

Spring is a lightweight framework. It can be thought of as a framework of frameworks because it provides support to various frameworks such as Struts, Hibernate, Tapestry, EJB, JSF etc. The framework, in broader sense, can be defined as a structure where we find solution of the various technical problems.

The Spring framework comprises several modules such as IOC, AOP, DAO, Context, ORM, WEB MVC etc. We will learn these modules in next page. Let's understand the IOC and Dependency Injection first.

The core features of the Spring Framework can be used in developing any Java application, but there are extensions for building web applications on top of the Java EE platform. Spring framework targets to make J2EE development easier to use and promotes good programming practices by enabling a POJO-based programming model.

Following is the list of few of the great benefits of using Spring Framework −

- Spring enables developers to develop enterprise-class applications using POJOs. The benefit of using only POJOs is that you do not need an EJB container product such as an application server but you have the option of using only a robust servlet container such as Tomcat or some commercial product.

- Spring is organized in a modular fashion. Even though the number of packages and classes are substantial, you have to worry only about the ones you need and ignore the rest.

- Spring does not reinvent the wheel, instead it truly makes use of some of the existing technologies like several ORM frameworks, logging frameworks, JEE, Quartz and JDK timers, and other view technologies.

- Testing an application written with Spring is simple because environment-dependent code is moved into this framework. Furthermore, by using JavaBeanstyle POJOs, it becomes easier to use dependency injection for injecting test data.

- Spring's web framework is a well-designed web MVC framework, which provides a great alternative to web frameworks such as Struts or other over-engineered or less popular web frameworks.

- Spring provides a convenient API to translate technology-specific exceptions (thrown by JDBC, Hibernate, or JDO, for example) into consistent, unchecked exceptions.

- Lightweight IoC containers tend to be lightweight, especially when compared to EJB containers, for example. This is beneficial for developing and deploying applications on computers with limited memory and CPU resources.

- Spring provides a consistent transaction management interface that can scale down to a local transaction (using a single database, for example) and scale up to global transactions (using JTA, for example).

3.1.3. HIBERNATE

Hibernate is an Object-Relational Mapping (ORM) solution for JAVA. It is an open source persistent framework created by Gavin King in 2001. It is a powerful, high performance Object-Relational Persistence and Query service for any Java Application.

Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieves the developer from 95% of common data persistence related programming tasks.

Hibernate sits between traditional Java objects and database server to handle all the works in persisting those objects based on the appropriate O/R mechanisms and patterns.

Figure 1: Hibernate ORM

Hibernate Advantages

- Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.
- Provides simple APIs for storing and retrieving Java objects directly to and from the database.
- If there is change in the database or in any table, then you need to change the XML file properties only.
- Abstracts away the unfamiliar SQL types and provides a way to work around familiar Java Objects.
- Hibernate does not require an application server to operate.
- Manipulates Complex associations of objects of your database.
- Minimizes database access with smart fetching strategies.

3.1.4. JSP

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

JavaServer Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through Webpage forms, present records from a database or another source, and create Webpages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages, and sharing information between requests, pages etc.

Advantages of JSP

Following table lists out the other advantages of using JSP over other technologies −

vs. Active Server Pages (ASP)

The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other MS specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

vs. Pure Servlets

It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML.

vs. Server-Side Includes (SSI)

SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.

vs. JavaScript

JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.

vs. Static HTML

Regular HTML, of course, cannot contain dynamic information.

3.2. TOOLS USED

3.2.1. ECLIPSE

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications.

3.2.2. VISUAL STUDIO CODE

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. The source code is free and open source and released under the permissive MIT License. The compiled binaries are freeware and free for private or commercial use.

3.2.3. MYSQL DATABASE

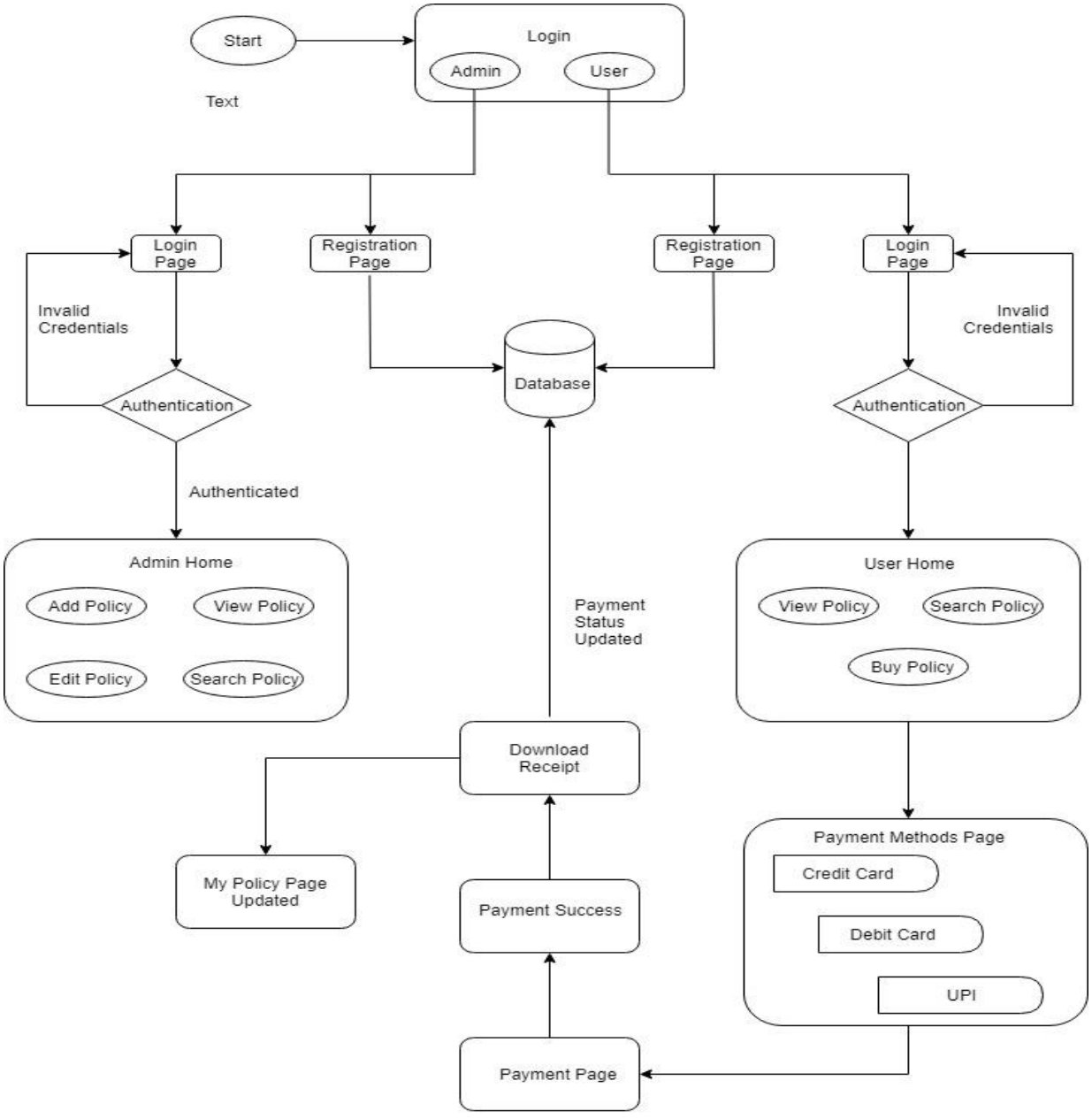
MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

CHAPTER - 4

DESIGN

4.1. SYSTEM DESIGN



4.2. PROCESS ARCHITECTURE

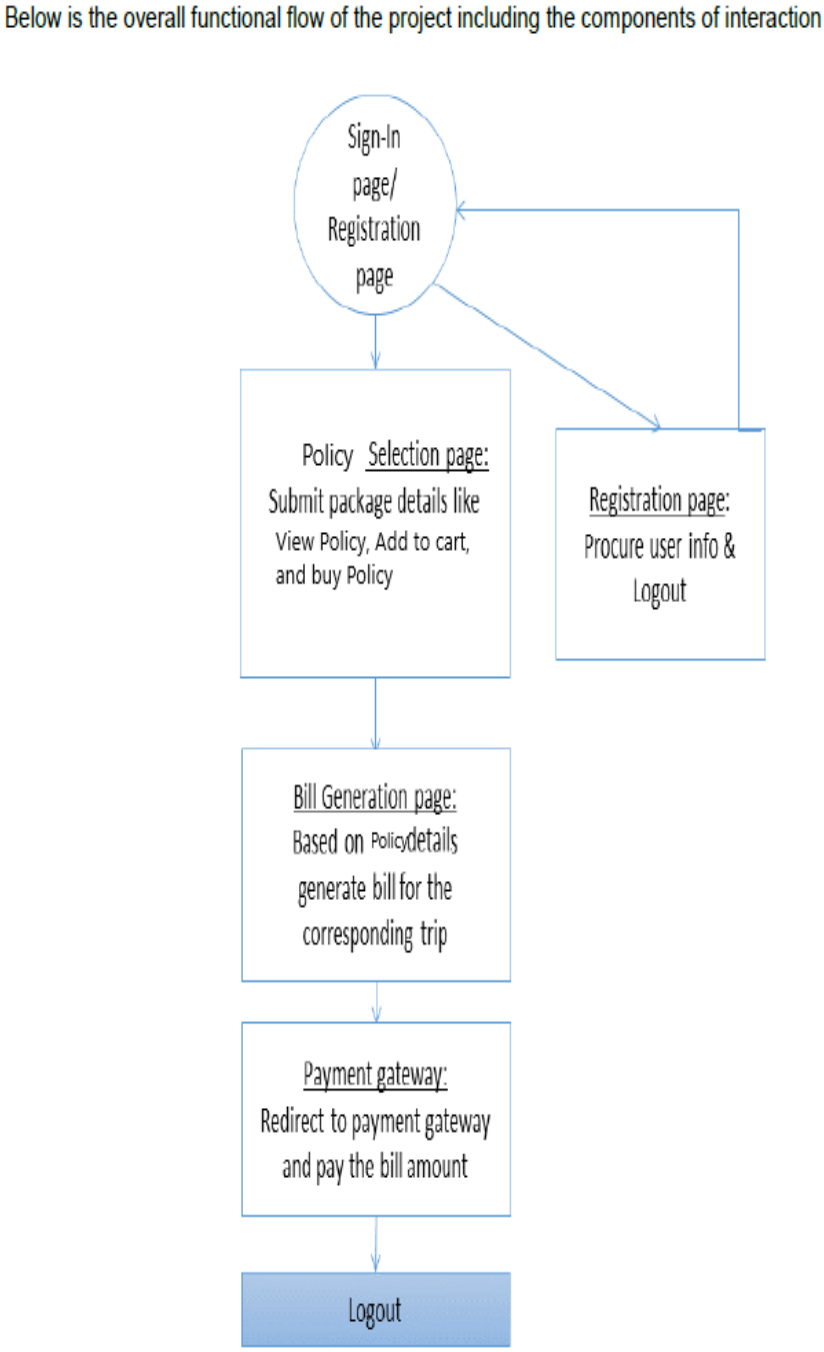


Figure 2:Process Architecture

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 Requirement / Use case ID	Remarks
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	F	Critical	NA	Req_1	Req_1.8

		database entry.					
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	E	Medium	NA	Req_1	Req_1.12

		missed out field in red					
Req_1.13	Admin Registration	Post-successful field level validation, save the information in the database	F	Critical	NA	Req_1	Req_1.13
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,	E	Medium	NA	Req_2	Req_2.3

		all the fields should be validated and the details has to be saved in the database.					
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	F	Medium	NA	Req_3	Req_3.5

		the database.					
Req_3.6	Edit 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	F	Medium	NA	Req_3	
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	F	Medium	NA	Req_4	Req_4.4

		Search policy button without selecting any fields should be prompted with an error message					
Req_5.1	Policy Payment	The user on going to Policy payment page should display the user details – policy Id, Bill Date, Payment amount, Fine, Due date	UI	Critical	NA	Req_5	Req_5.1
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'	F	Critical	NA	Req_5	Req_5.3

		to ‘paid’ in the database.					

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 3: BillDetail

4.6. DATA FLOW DIAGRAM

4.6.1. CONTEXT LEVEL DFD

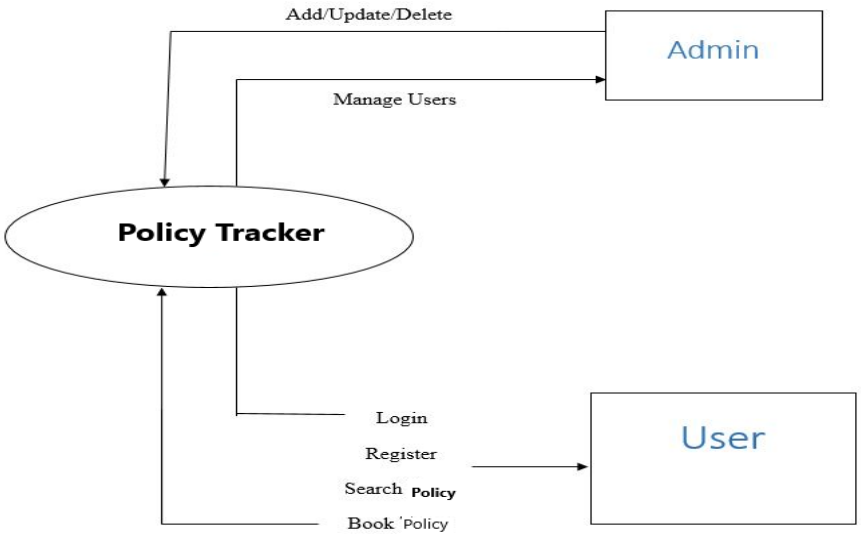


Figure 6: Context Level DFD

4.6. 2. FIRST LEVEL DFD

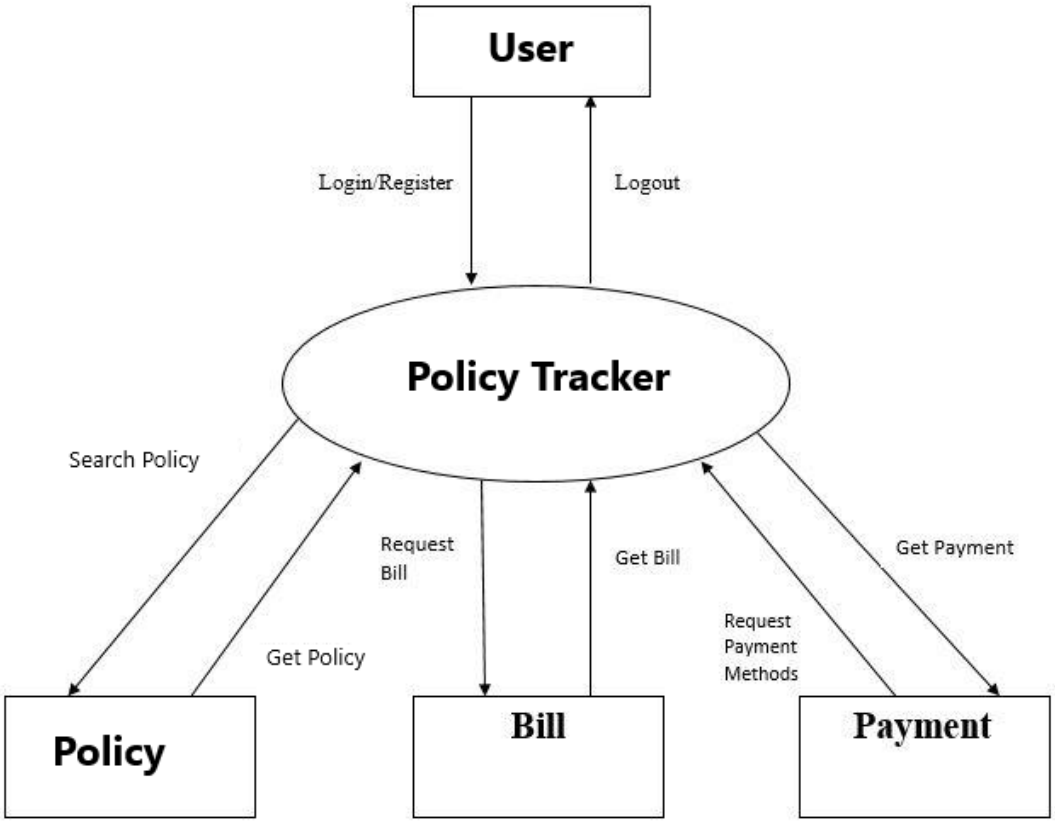


Figure 7:First level DFD

CHAPTER – 5

SYSTEM TESTING

Testing is the process of evaluating a system or its components with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results i.e. testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

5.1. TESTING STRATEGIES

In order to make sure that system does not have any errors, the different levels of testing strategies that are applied at different phases of software development are

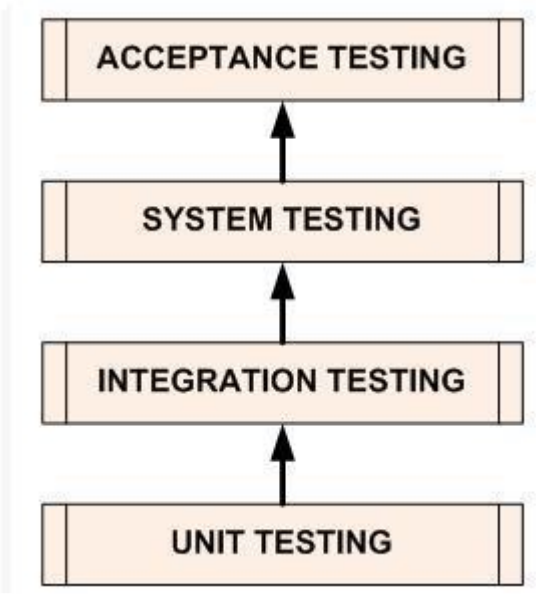


Figure 8: Testing Strategies

5.1.1. UNIT TESTING

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

5.1.2. INTEGRATION TESTING

The testing of combined parts of an application to determine if they function correctly together is Integration testing. This testing can be done by using two different methods

5.1.2.1. TOP DOWN INTEGRATION TESTING

In Top-Down integration testing, the highest-level modules are tested first and then progressively lower-level modules are tested.

5.1.2.2. BOTTOM-UP INTEGRATION TESTING

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

5.1.3. SYSTEM TESTING

This is the next level in the testing and tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards. 30

5.1.4. ACCEPTANCE TESTING

The main purpose of this Testing is to find whether application meets the intended specifications and satisfies the client's requirements. We will follow two different methods in this testing.

5.1.4.1. ALPHA TESTING

This test is the first stage of testing and will be performed amongst the teams. Unit testing, integration testing and system testing when combined are known as alpha testing. During this phase, the following will be tested in the application:

- Broken Links.
- The Application will be tested on machines with the lowest specification to test loading times and any latency problems.

5.1.4.2. BETA TESTING

In beta testing, a sample of the intended audience tests the application and send their feedback to the project team. Getting the feedback, the project team can fix the problems before releasing the software to the actual users.

5.2 TESTING METHODS

5.2.1 WHITE BOX TESTING

White box testing is the detailed investigation of internal logic and structure of the Code. To perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

5.2.2 BLACK BOX TESTING

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

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

Table 3:Test Case

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.

7.1.2 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

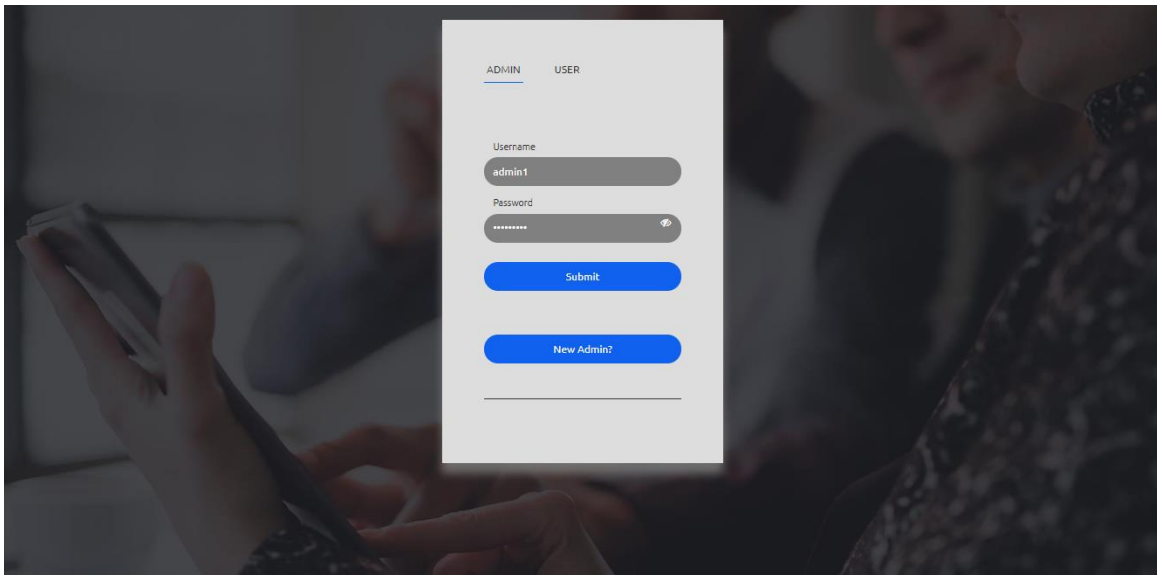


Figure 9: Login Page

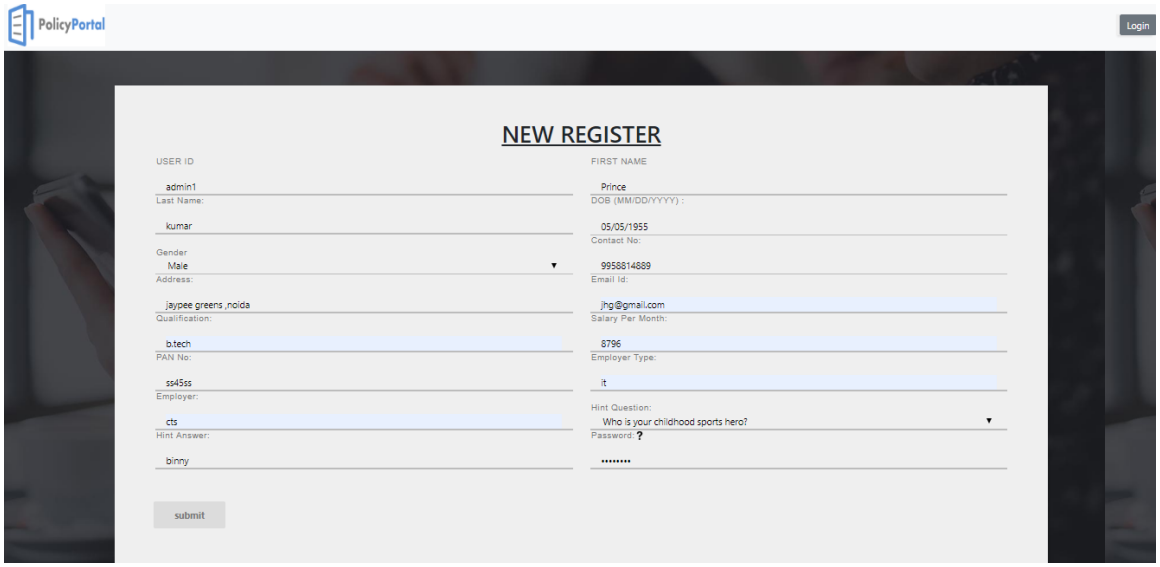


Figure 10: Registration Page

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.

7.2.2. 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

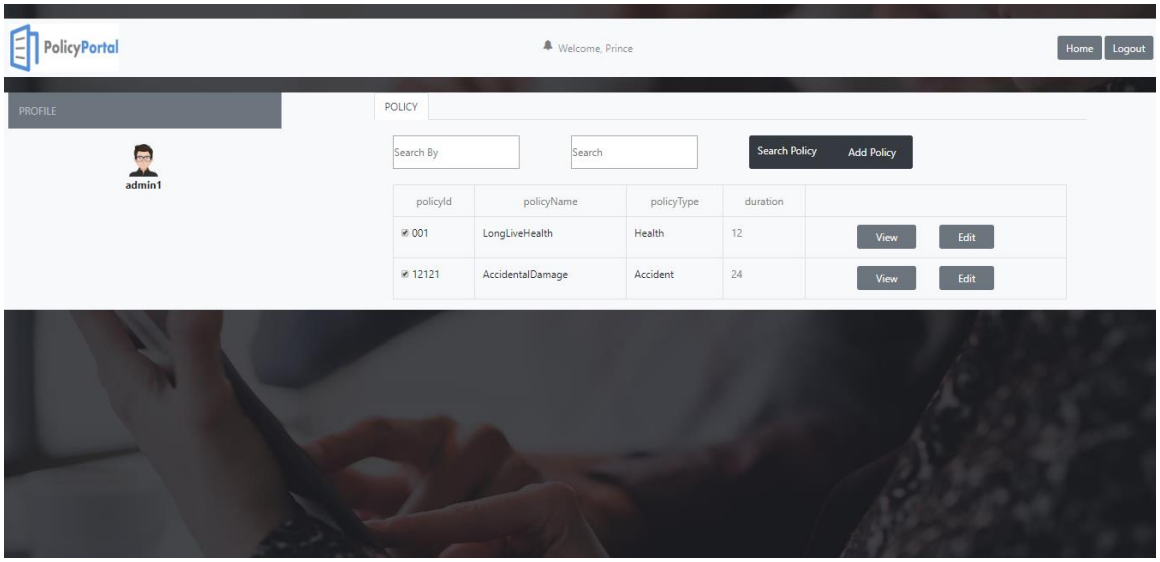


Figure 11 : User Home

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

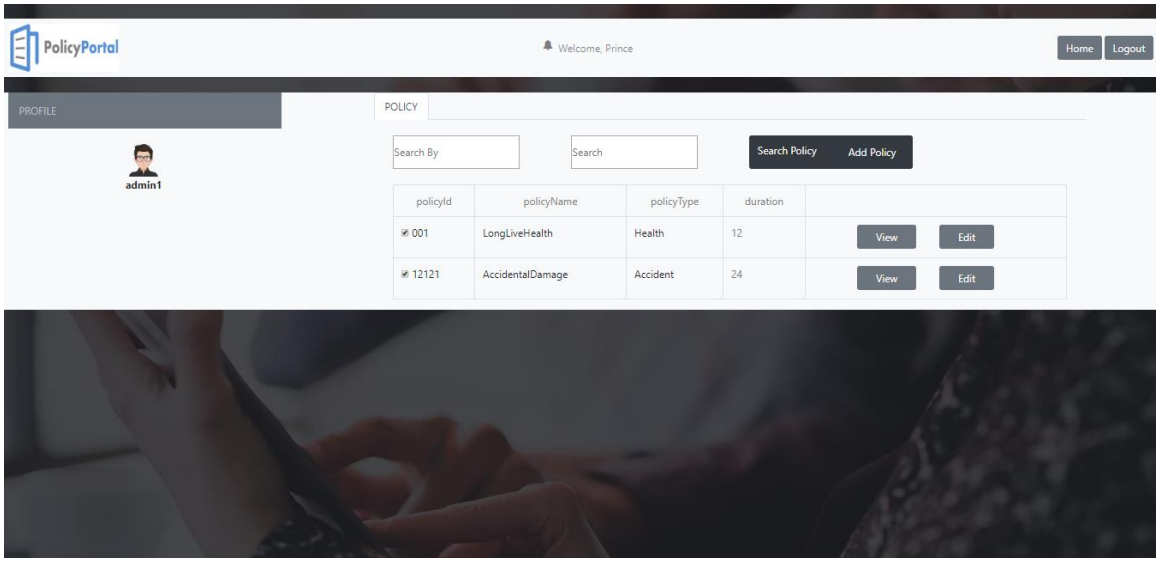


Figure 12: User View Details

6.4. 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

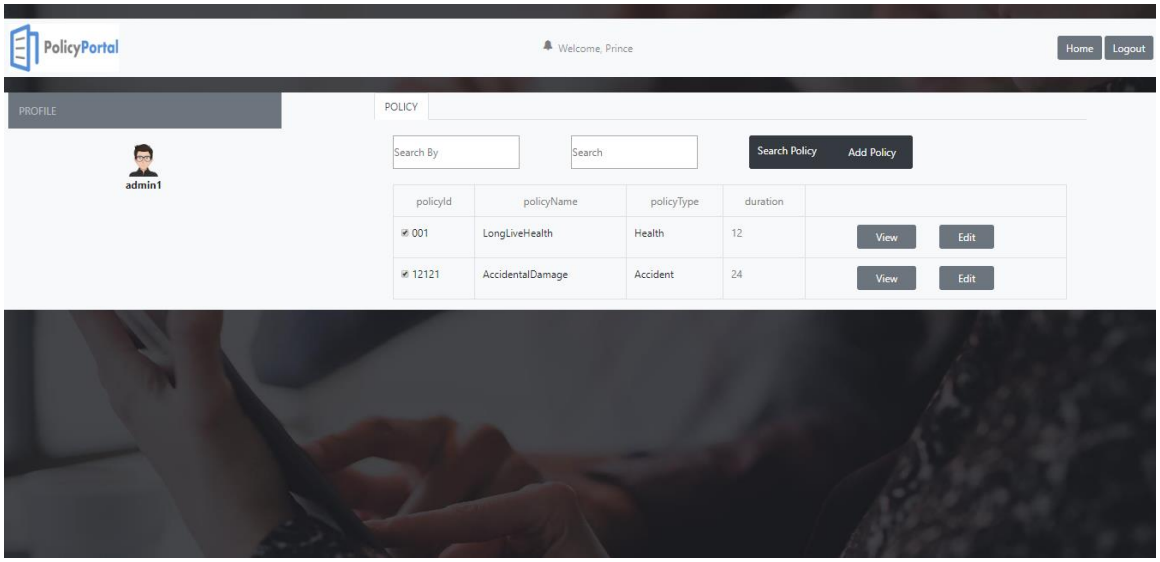


Figure 13: Search Page

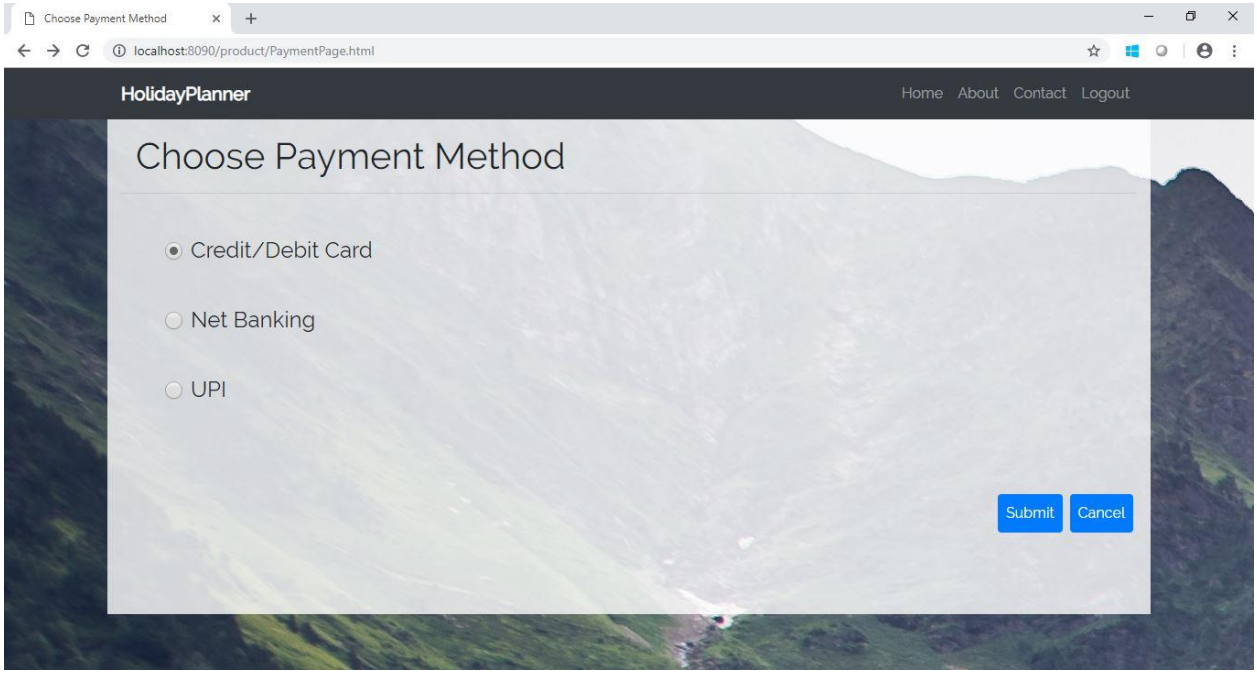


Figure 14: Bill Payment Page

6.4. 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

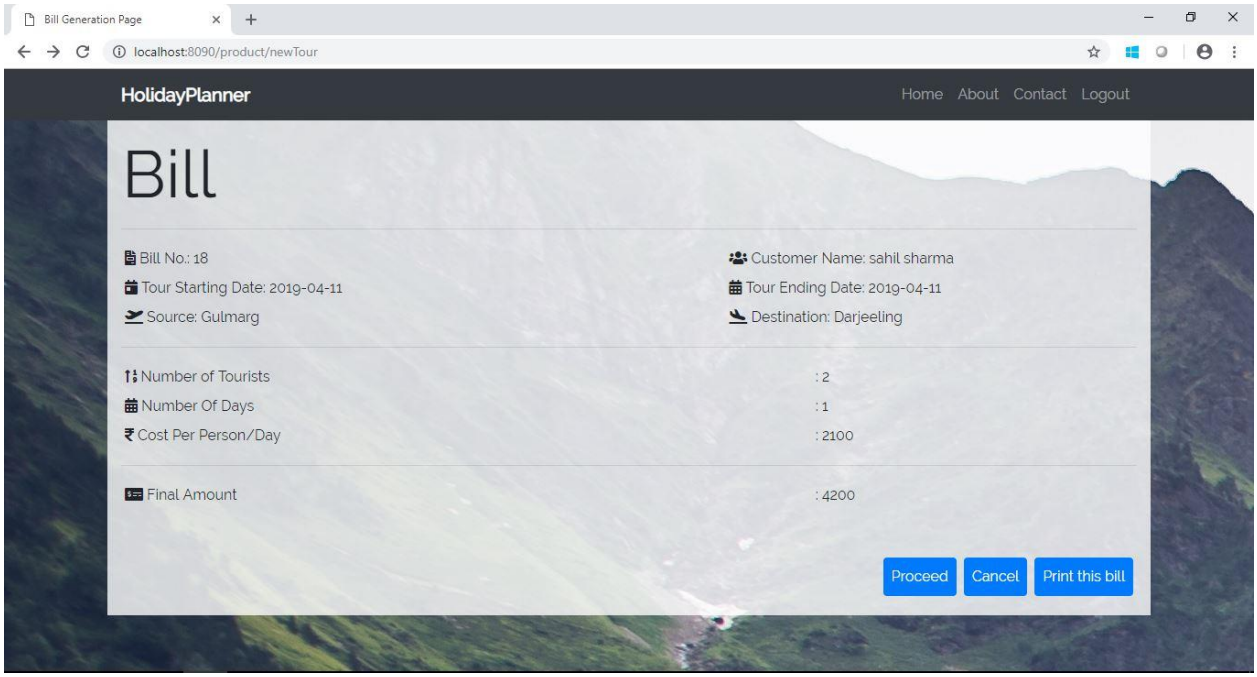


Figure 15: Bill Page

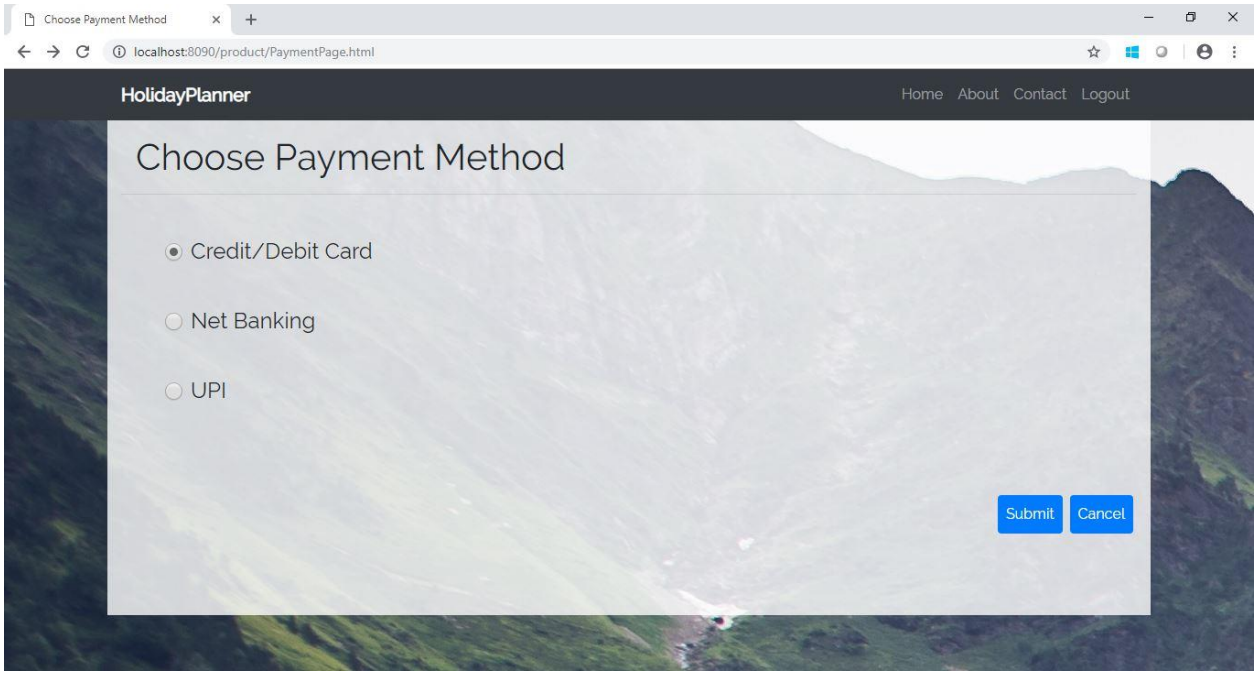


Figure 16: Bill Payment Page

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

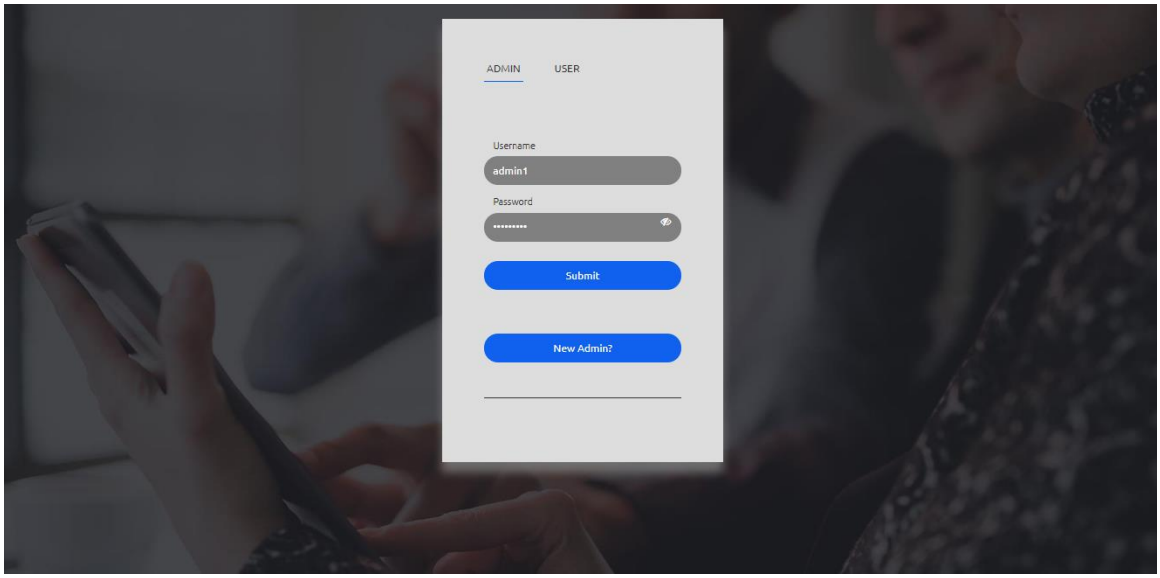


Figure 17: Tour Details

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

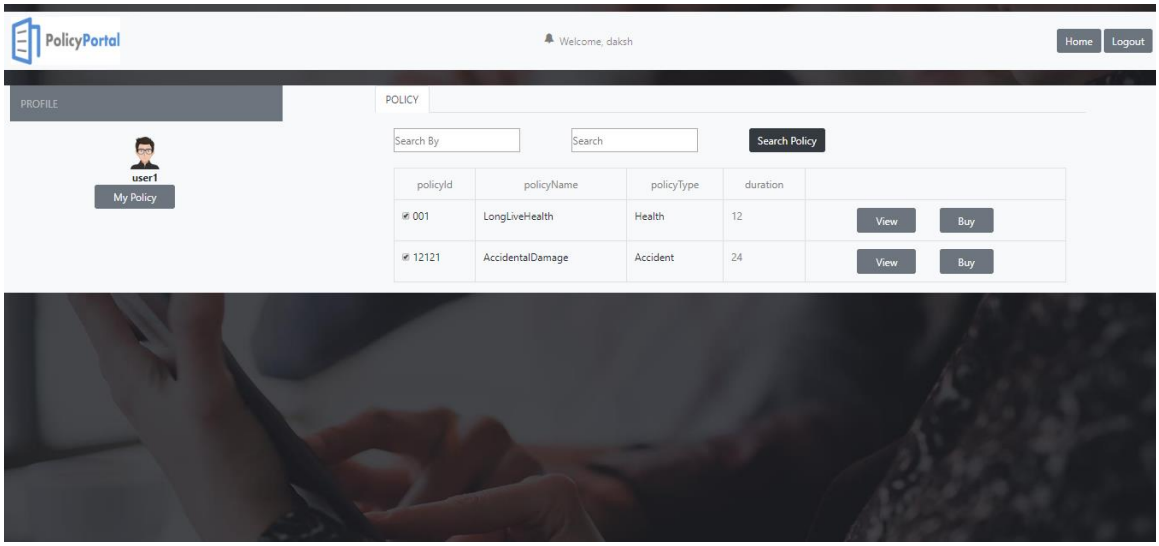


Figure 18: User Details

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

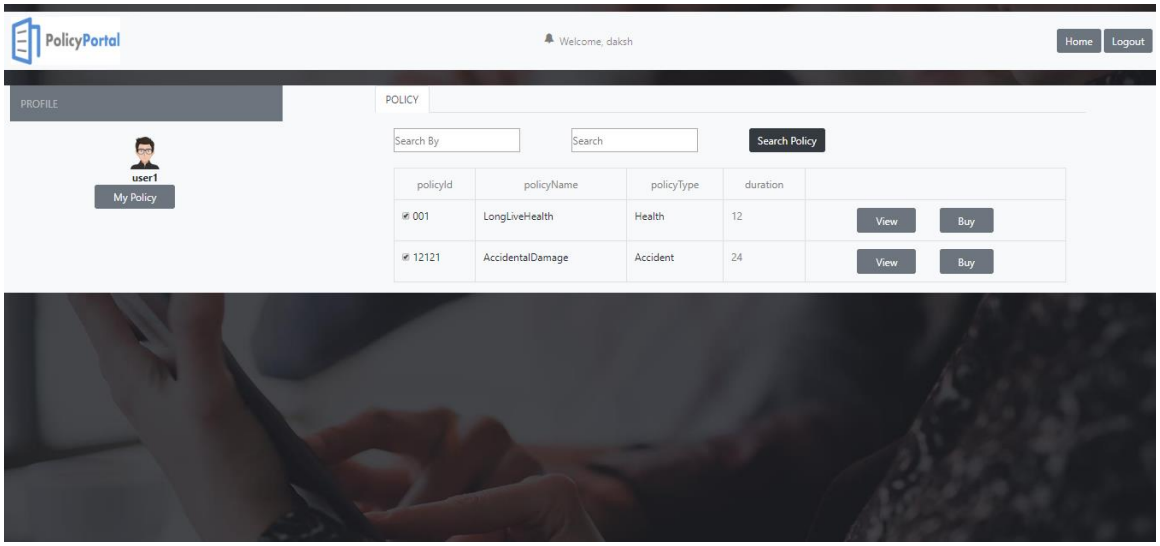


Figure 19: User Details

CHAPTER – 7

IMPLEMENTATION

PROCESS MODEL USED

Spiral model

The Spiral model originally proposed, is an evolutionary software process model that couples the iterative nature of prototyping with the controlled & systematic aspects of the linear sequential model. It provides the potential for rapid development of incremental version of the software.

Using the spiral model software is developed in a series of incremental releases. A spiral model is divided into a number of framework activities also called task regions.

A spiral model contains six task regions:

- **Customer Communication:** Tasks required to establish effective communication between developer & customer.
- **Planning:** Tasks required to define resources, timeline & timeline & another project related information.
- **Risk analysis:** Task required to access.
- **Engineering:** Tasks required to build one or more representation of the application.
- **Construction & release:** Task required to construct, test, install & provide user support (e.g., documentation & Training)
- **Customer evaluation:** Tasks required to obtain customer feedback based on evolution of the software representation created during the engineering stage & implemented during the installation stage.

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

The Post Implementation Review (PIR) is conducted after a project has been completed. The purpose of the PIR is to evaluate how successfully the project objectives have been met and how effective the project management practices were in keeping the project on track.

In our project the all objectives met to 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 is generally found that systems that are easy to use, require less manpower, saves the data entry and well received by people. But still the following points have to consider.

1. How have systems changed the way in which operations were performed?
2. How have systems changed the timeliness of information and reports user received?

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

We have learnt a lot of things while developing the project.

- 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.

2. DEMONSTRATION/SCREENSHOTS

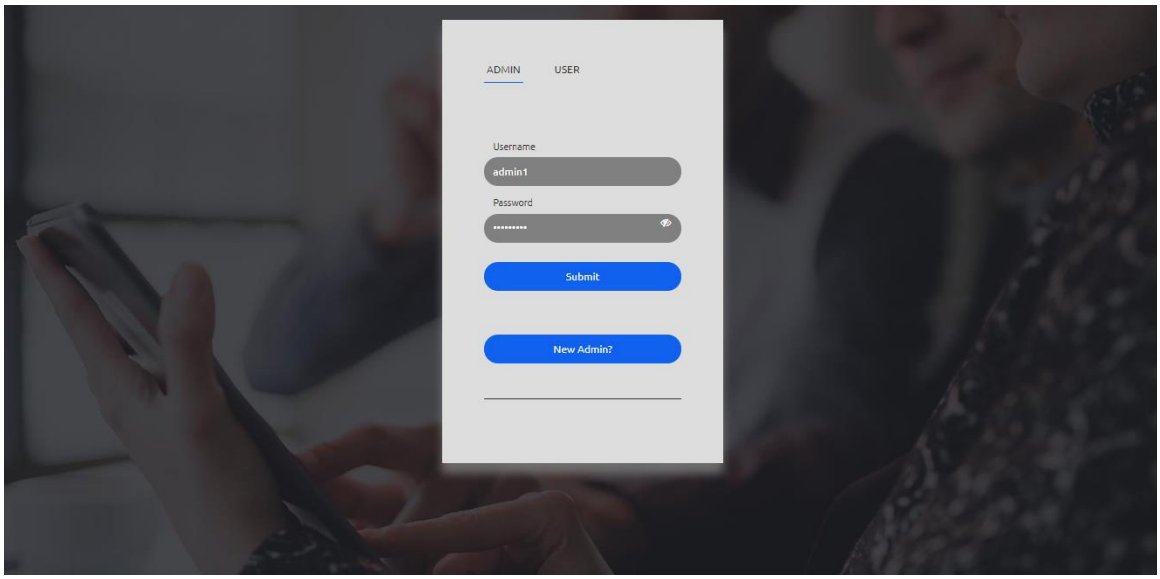


Figure 20: Admin Login Page

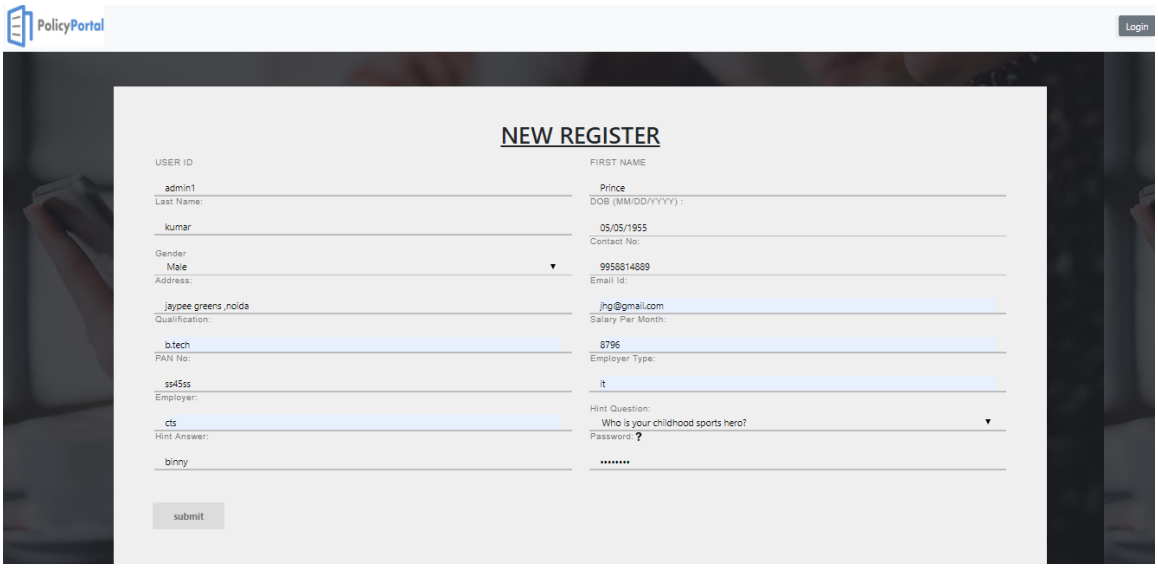


Figure 21 : Registration Page

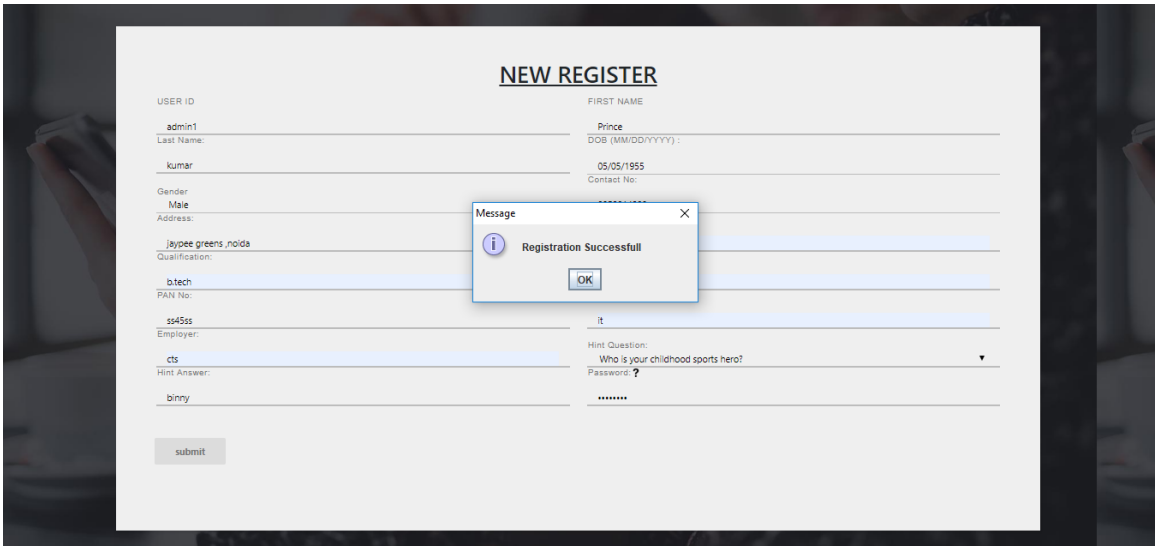


Figure 22 : Registration Successful Pop-up

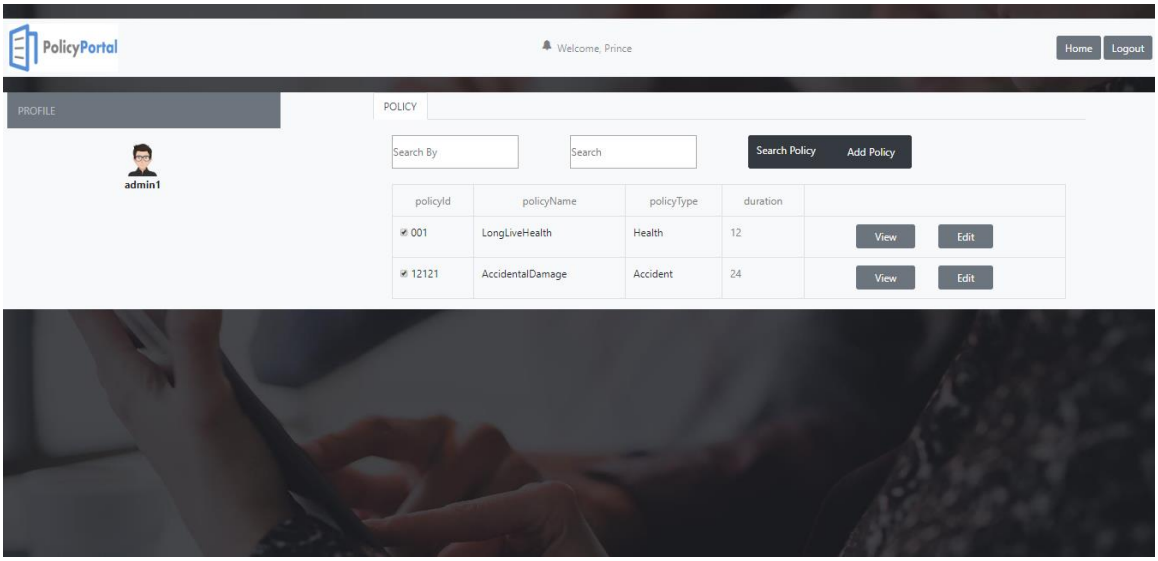


Figure 23 : Admin Home Page

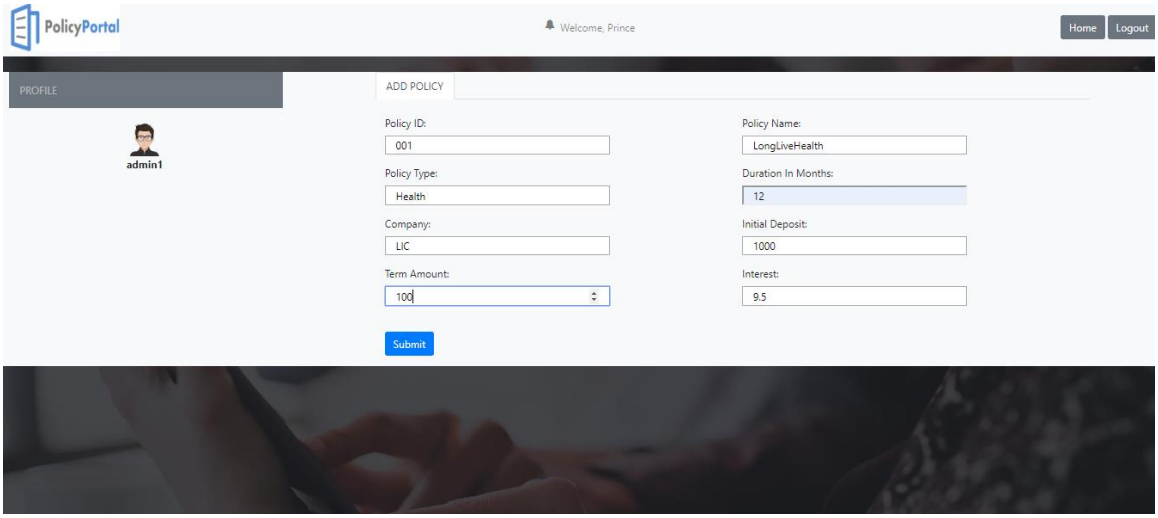


Figure 24 : Add Policy Page

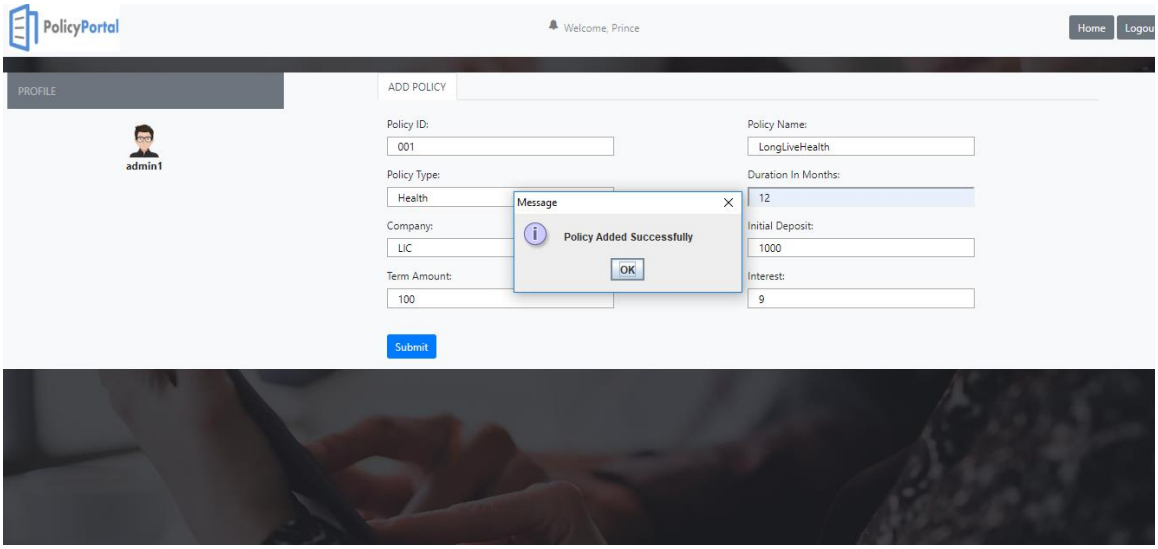


Figure 25 : Policy Addition Successful

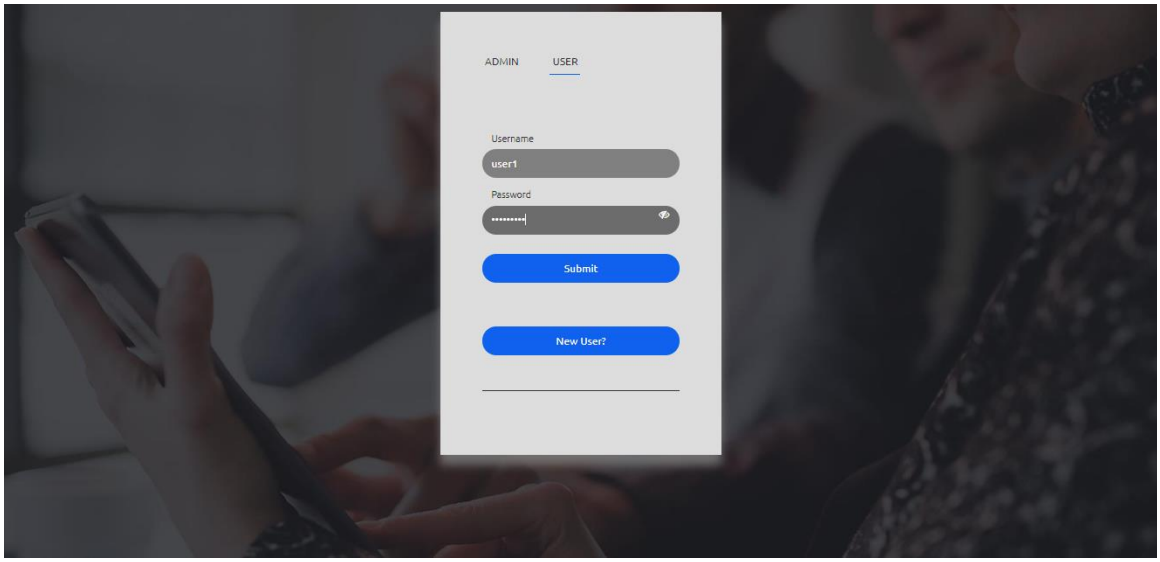


Figure 26 : User Login

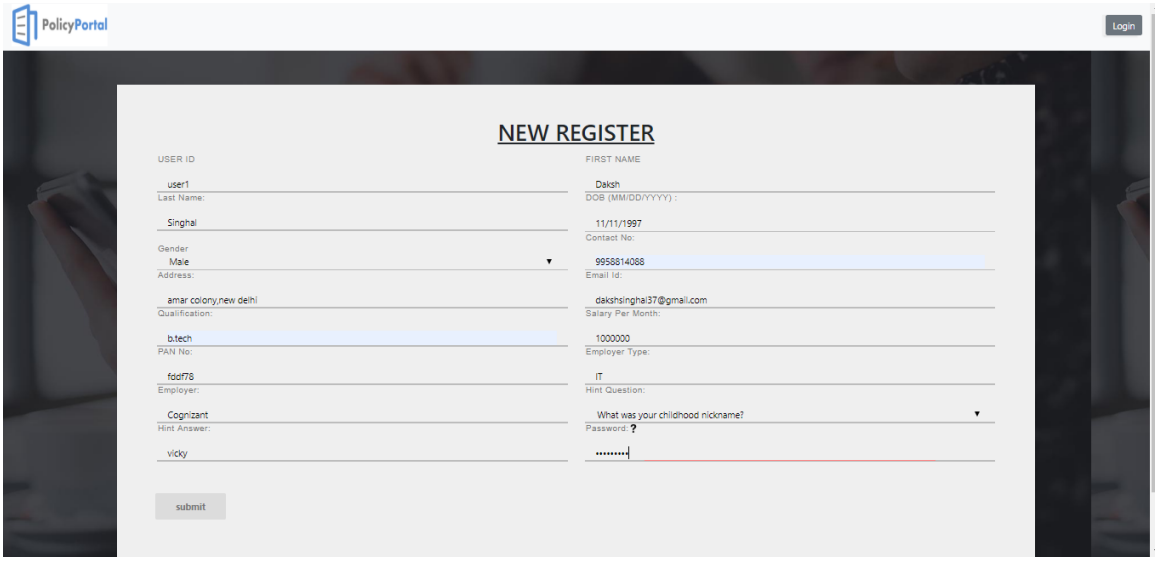


Figure 27 : User Registration Page

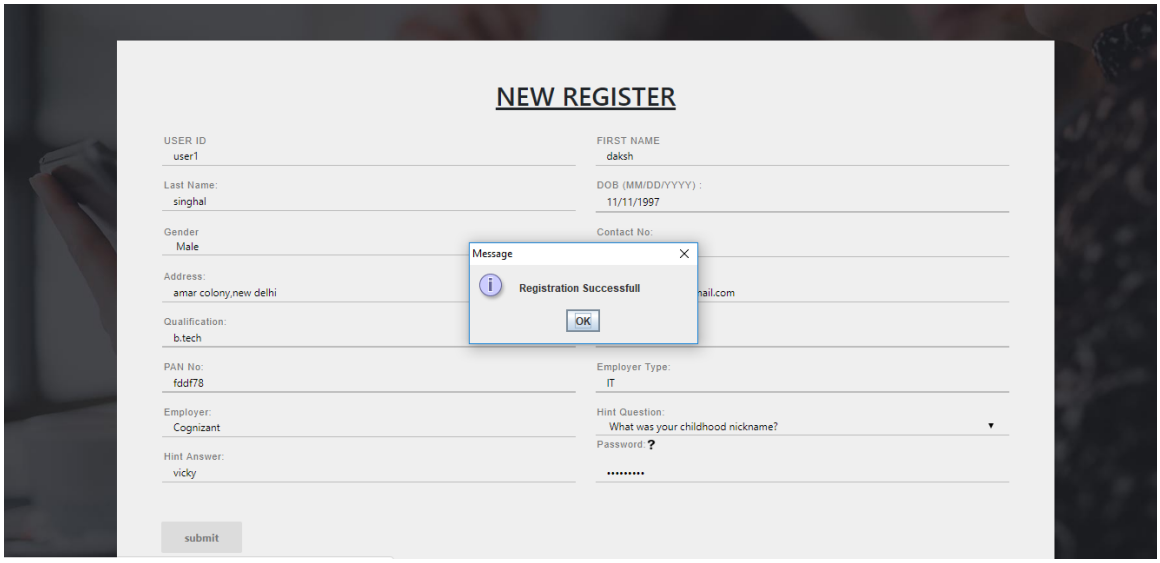


Figure 28 : User Registration Successful Pop-up

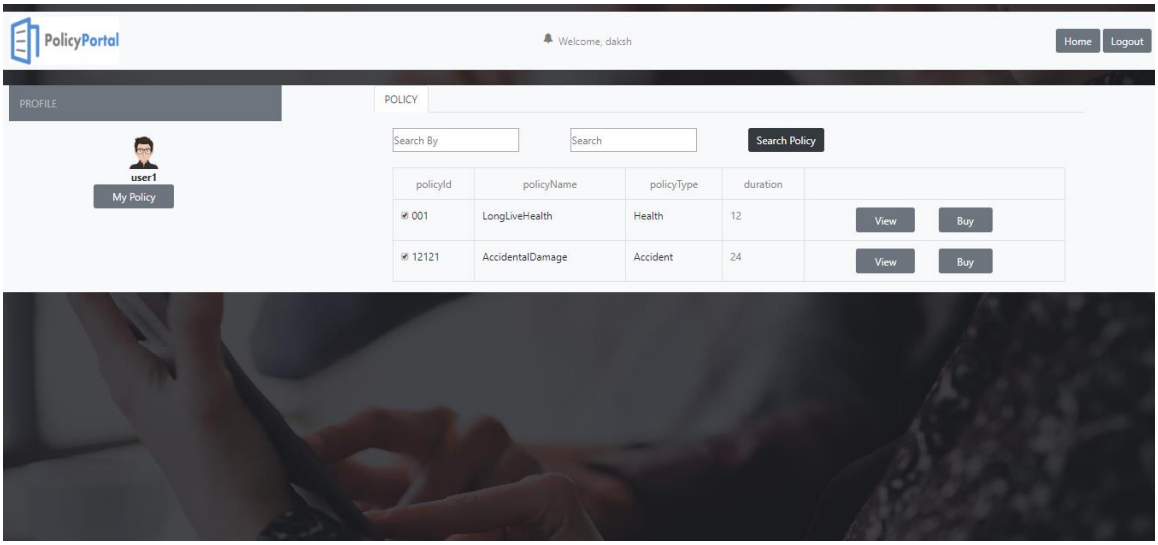


Figure 29 : User Home Page

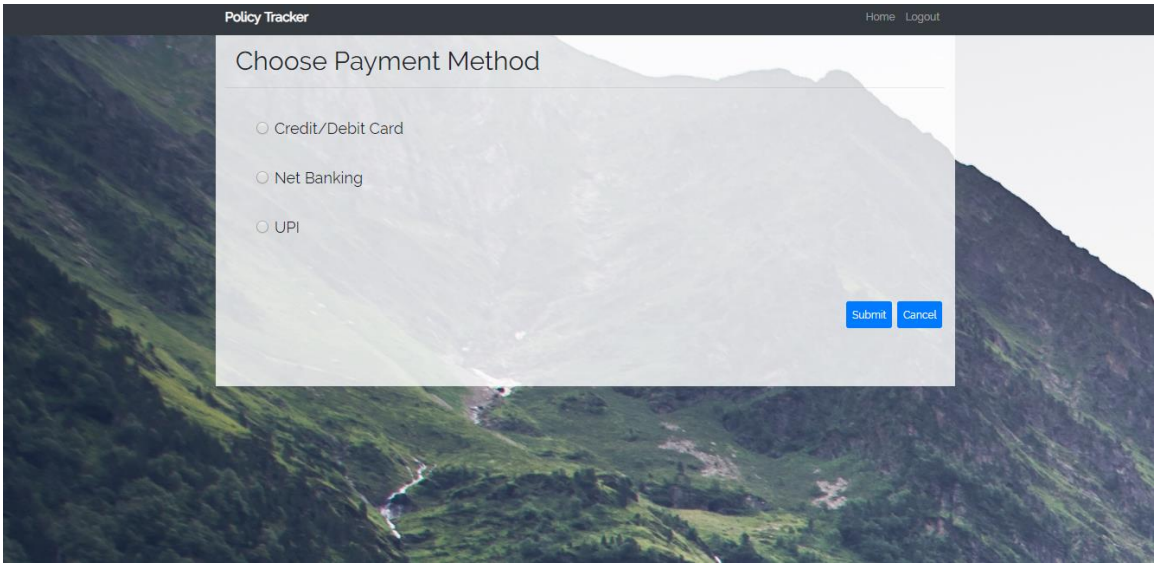


Figure 30 : Payment Methods Page

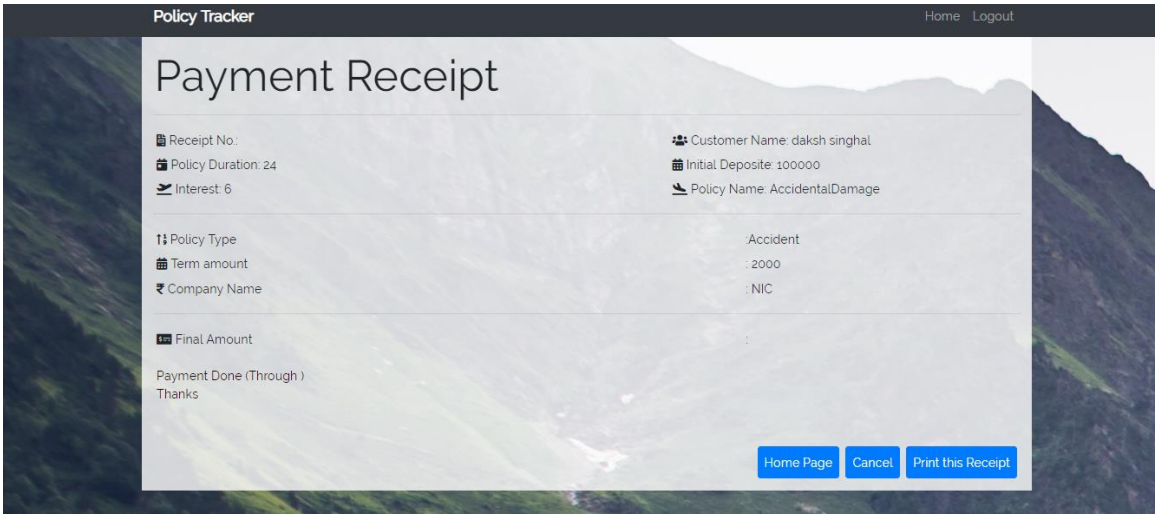


Figure 31 : Payment Successful Page

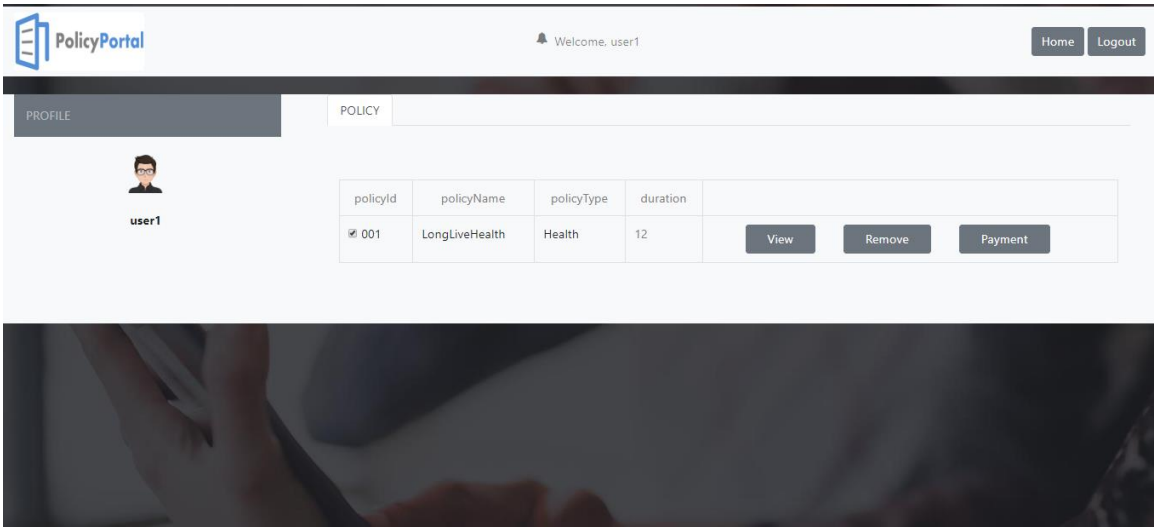


Figure 32 : My Policy Page

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/