VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi - 590018

Project Report
on

"INSURANCE POLICY DASHBOARD"

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
by

RUCHITHA N SHETTY	4MT20CS130
SAKSHI SHETTY	4MT20CS136
SOURABH P	4MT20CS165
SABANNA	4MT21CS403
KUMAR MADAR	4MT19CS072

MITE
Invent Solutions

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

(Accredited by NBA)

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution
(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)
Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Badaga Mijar, Moodabidri-574225, Karnataka
2022-23

ABSTRACT

This project report comprehensively analyses an Insurance Policy Dashboard developed using JSP and MySQL. The dashboard caters to two types of users: administrators and regular users. Administrators have the authority to add, update, and delete insurance policies, while users can purchase the available policies. The purchased insurances are accessible in the user dashboard, while the homepage showcases all the unpurchased policies. The admin homepage provides a centralized view of all the added insurances.

The application emphasizes a user-friendly interface, ensuring a smooth and engaging experience for both administrators and users. The login process is streamlined, enabling quick access to the respective dashboards. The dashboard design combines visual appeal with intuitive navigation, enhancing usability. The application integrates APIs, leveraging external data sources to enhance the accuracy and range of insurance policy recommendations. This integration facilitates personalized recommendations based on user preferences, enabling users to explore and select suitable insurance options. Throughout the development process, the project places significant emphasis on security and performance. Industry best practices are followed to protect user data and ensure optimal system performance, adhering to established industry standards. The utilization of version control ensures effective code management, fostering collaboration and enabling traceability of changes.

In conclusion, this JSP and MySQL-based Insurance Policy Dashboard project successfully allows administrators to manage insurance policies while enabling users to purchase and access these policies conveniently. The project's focus on user experience, security, and performance ensures a robust and efficient solution for the insurance industry.

TABLE OF CONTENTS

LIST OF SNAPSHOTS

CHAPTER 1

INTRODUCTION

This report provides an introduction to our project, an Insurance Policy Dashboard built using JSP (JavaServer Pages) and MySQL. This project aims to create a user-friendly and efficient web application that caters to two types of users: administrators and regular users. The primary objective is to enable administrators to manage insurance policies by adding, updating, and deleting them, while users can purchase the available policies.

The Insurance Policy Dashboard is designed to optimize the insurance policy management process and enhance the user experience. It offers administrators a centralized platform where they can easily view, update, and remove insurance policies. On the other hand, users are provided with a convenient interface to explore and purchase insurance policies. The purchased policies are accessible in the user dashboard, while the homepage displays all the available unpurchased policies. To ensure seamless access and compatibility across various devices, the application is built as a responsive web platform. This enables users to access the dashboard and make policy purchases from both desktop computers and mobile devices. Our development approach adheres to industry best practices, focusing on clean and well-documented code, robust security measures, high performance, and effective version control. This ensures that the application meets industry standards and delivers a secure and efficient user experience.

Overall, the Insurance Policy Dashboard project aims to streamline the insurance policy management process, improve user experience, and provide a reliable platform for administrators and users in the insurance industry.

CHAPTER 2

FUNCTIONAL REQUIREMENTS

Objective

The objective of the system is to create a platform that enables insurance company admins to efficiently introduce, update, and manage their policies. It aims to provide users with the ability to easily view and purchase policies from the available options. Additionally, the system will track and maintain a list of purchased policies for each user, ensuring transparency and accessibility of their policy information. By fulfilling these objectives, the system aims to streamline policy management and enhance the user experience in the insurance domain.

Features Supported in Application: -

Functions available for Admins are as follows:

1. Add an Insurance Policy
2. View the list of policies under him/her
3. Update/ Delete Insurance Policy

Functions available for Users are as follows:

1. View his/her purchased policies
2. View other policies
3. Purchase the policy

Features for Admin:

Login > Dashboard > Policy Specific Info with policyholders
OR
Login > Dashboard > Add policy
OR
Login > Dashboard > Update/ Delete Policy

On the policy dashboard page provided for the admins, a list of policies available will be displayed in a tabular form. Each row will contain general information about each policy. Here admins can introduce a new policy and can update the existing policy.

Field Name	Description
Add the Policy	Allows the admins to add a new policy.
Update/ Delete Policy	Allow the admin to update or delete the policy.
Policy Specific Page with Policy Holders	Display all the information about the policy and policyholders' Information.

Admin Interactions:

1. select the Add option in the dashboard to add a policy.

   All the related fields will appear on the screen as a form that has to be filled.
2. To **Update/ Delete a Policy**, select the options filed concerning the policy in the dashboard.

   OR

   Click the policy to **Update/Delete** then click the options filled in that page.
3. Click and then click **View** the policy.

   Detailed information about the policy and a list of policyholders as a table having a name, email, starting and ending date of the policy, premiums paid/ remaining, and date of next premium will be displayed.

Click **Back** to Dashboard to navigate back to the dashboard

Features for Users:

Register/Login > Home > View Purchased Policies

OR

Register/Login >Home > View Policies

OR

Register/Login >Home > View Policies > Compare policy

OR

Register/Login >Home > Purchase Policy

When the user logs in to the application, the user will be displayed on the home page. From here, users can find different policies available and purchase a specific policy by navigating to it or viewing and comparing similar policies available.

From the home page user can go to his/her dashboard to view purchased policies. Where different fields related to it will be displayed.

Field Name	**Description**
View Dashboard	Display a list of purchased policies by that user.
Available policies to buy	Displays the list of policies that can be purchased.
Policy Specific Page.	Display all the information about the policy.
Compare Policy in the Viewed Policy Page	Display similar category policies to compare the pros and cons.
Purchase Policy	Allow users to purchase the policy.

User's Interactions:

1. To **View the Dashboard,** click the **Dashboard** option. This will take the user to the dashboard, where a list of purchased policies and their respective fields like name of the policy, next premium date & amount, start date, end date, etc will be displayed.
2. To view the **Available policies to buy, the** user can find a policy best suited for him/her.
3. Click the specific policy to **View Detailed information** about the policy.
4. User can **Compare related Policies** in one of the interested policy-specific pages.
5. To **Purchase the Policy**, the user needs to click the **Purchase** button, which will ask for confirmation, and click the **Confirm** button to confirm the purchase.

Policy specific page will contain all the necessary information as entered by the admin will be displayed.

Click **Back to Home** to navigate back to the Home.

Click **Logout** to log out.

CHAPTER 3

TECHNICAL DETAILS

3.1 JSP

JSP (JavaServer Pages) is a technology that allows the creation of dynamic web pages by combining HTML or XML with Java code. It provides a way to separate the presentation logic from the business logic in web applications. Here are some key points about JSP:

Dynamic Content Generation:

- JSP allows the inclusion of Java code within HTML or XML markup.

- The Java code can be used to generate dynamic content, retrieve data from databases, perform calculations, or interact with other components of the web application.

Server-Side Processing:

- JSP files are processed on the server side and converted into servlets before being executed.

- The servlet container compiles the JSP files into Java servlets, which are then handled by the servlet engine to generate the final HTML output.

Tag-Based Syntax:

- JSP uses both standard HTML/XML tags and JSP-specific tags.

- Standard tags are used for creating HTML structure, while JSP tags provide additional functionality and control over dynamic content generation.

Directives:

- JSP includes directives that provide instructions to the container for processing the JSP files.

- Common directives include the `page` directive for setting page-level attributes, the `include` directive for including other files, and the `taglib` directive for importing custom tag libraries.

3.2 HTML

HTML (Hypertext Markup Language) is the standard markup language for creating web pages and applications. It defines the structure and content of a webpage, allowing browsers to interpret and display the information to users. Here are some key points about HTML:

Document Structure:

HTML documents have a hierarchical structure defined by tags. The basic structure consists of an opening <html> tag and a closing </html> tag, with the content contained within the <body> tags.

Tags and Elements:

HTML uses tags to define elements and their characteristics. Tags are enclosed in angle brackets, such as <tagname>content</tagname>. Examples of common tags include <p> for paragraphs, <h1> to <h6> for headings, <a> for links, and <img> for images.

Attributes:

Tags can have attributes that provide additional information or modify their behavior. Attributes are specified within the opening tag, using the format attributeName="value". For example, the <a> tag has an href attribute that specifies the URL of the link.

Semantic Elements:

HTML5 introduced semantic elements that convey the meaning of the content. Semantic elements, such as <header>, <nav>, <main>, <section>, and <footer>, provide a clearer structure and improve accessibility.

3.3 Cascading Style Sheets (CSS):

CSS (Cascading Style Sheets) is a styling language used to describe the presentation and appearance of HTML (or XML) documents. It provides a way to define the visual aspects of web pages, including layout, colors, fonts, and other stylistic elements. Here are some key points about CSS:

Selectors and Declarations:

- CSS uses selectors to target HTML elements that need to be styled. Selectors can be based on element types, class names, IDs, attributes, or their relationships within the document structure.

- Declarations are used to define the styles applied to selected elements, such as colors, fonts, margins, and padding.

Styling Techniques:

- CSS offers various styling techniques, including:

- Box Model: Controlling the dimensions and spacing of elements using properties like width, height, margin, padding, and border.

- Typography: Modifying font properties such as size, family, weight, style, and text alignment.

- Colors and Backgrounds: Setting background colors, images, gradients, and text colors.

- Layout: Positioning and arranging elements with properties like display, float, positioning, and flexbox/grid layouts.

- Transitions and Animations: Applying smooth transitions and animations to elements using CSS properties.

CSS Selectors:

CSS provides a wide range of selectors for targeting specific elements or groups of elements. Selectors include element selectors, class selectors, ID selectors, attribute selectors, pseudoclasses, and pseudo-elements.

3.4 JAVA

Java is a popular, general-purpose programming language that was first released by Sun Microsystems in 1995. It was designed to be platform-independent, meaning that Java programs can run on any device or operating system that has a Java Virtual Machine (JVM) installed. This characteristic, along with its object-oriented nature, has contributed to Java's widespread adoption and versatility.

Key Features of Java:

Object-Oriented: Java follows the principles of object-oriented programming (OOP), which organizes code into reusable objects that interact with each other. This approach promotes modularity, code reusability, and easier maintenance.

Platform-Independent: Java programs are compiled into bytecode, which can be executed on any system that has a JVM. This "write once, run anywhere" capability allows Java applications to run on different platforms without modification.

Strong Memory Management: Java manages memory automatically through a process called garbage collection. Developers do not have to manually allocate or deallocate memory, reducing the risk of memory leaks and improving overall program stability.

Robust Standard Library: Java comes with a comprehensive standard library, known as the Java Development Kit (JDK), which provides a wide range of classes and methods for various purposes. This extensive library simplifies common programming tasks, such as input/output operations, networking, and multithreading.

Exception Handling: Java includes built-in exception handling mechanisms, which allow developers to catch and handle errors or exceptional events during program execution. This feature enhances the robustness and reliability of Java applications.

CHAPTER 4

IMPLEMENTATION

4.1 Admin Dashboard

In the project built using JSP and MySQL for the Insurance Policy Dashboard, the admin operations play a crucial role in managing and maintaining the insurance policies within the system. The admin, being one of the two types of users, has specific privileges and responsibilities.

Adding Insurance Policies:

The admin has the authority to add new insurance policies to the system. This includes providing all the necessary details such as policy type, coverage details, premium rates, and additional benefits. By adding policies, the admin expands the range of options available for users to purchase.

Updating Insurance Policies:

The admin can update existing insurance policies to reflect any changes, such as modified coverage terms, premium adjustments, or revised benefits. This ensures that the policies remain up-to-date and accurately reflect the offerings provided by the insurance company.

Deleting Insurance Policies:

If required, the admin can remove insurance policies from the system. This may be necessary for policies that are no longer relevant or have been discontinued. Removing such policies ensures that users are presented with accurate and current options when browsing through available policies.

Overall, the admin operations in the Insurance Policy Dashboard project give the admin control over adding, updating, and deleting insurance policies. By monitoring purchased and unpurchased policies, the admin can effectively manage the offerings available to users, ensure accuracy, and make data-driven decisions to optimize the user experience and sales performance.

4.2 User Dashboard

The user portal within the Insurance Policy Dashboard project, developed using JSP and MySQL, serves as a central hub for customers, providing a seamless and personalized experience.

User Registration and Login:

The user portal offers a user-friendly registration process, allowing individuals to create an account effortlessly. By providing essential information such as name, email, and password, users can quickly register and gain access to the portal's extensive functionalities. Once registered, users can securely log in using their credentials and access their personalized dashboard.

Personalized Product Recommendations:

Leveraging advanced algorithms and user-provided information, the user portal provides personalized product recommendations tailored to each user's specific needs. By analyzing factors like existing insurance coverage, user preferences, and risk profiles, the portal offers relevant insurance product suggestions, simplifying the decision-making process for users.

Product Display and Details:

The user portal presents a comprehensive display of various available insurance products, allowing users to explore and compare offerings effortlessly. Each product is accompanied by detailed information, including coverage details, premium rates, and additional benefits, enabling users to make well-informed decisions.

Streamlined Application Process:

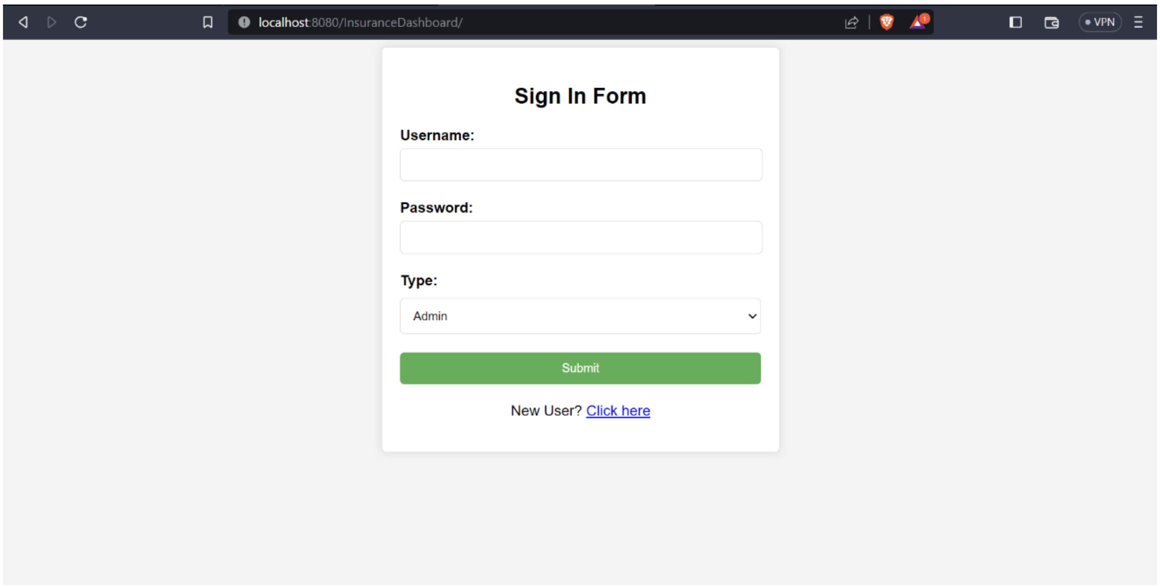
The user portal streamlines the insurance application process, providing users with a seamless and intuitive interface to submit their applications. By guiding users through a step-by-step process and automating form-filling where possible, the portal aims to minimize friction and enhance the efficiency of applying for insurance coverage.

In summary, the JSP and MySQL-based Insurance Policy Dashboard project's user portal serves as a centralized platform for users, offering a personalized and streamlined experience.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Sign-in Page



localhost:8080/InsuranceDashboard/

Sign In Form

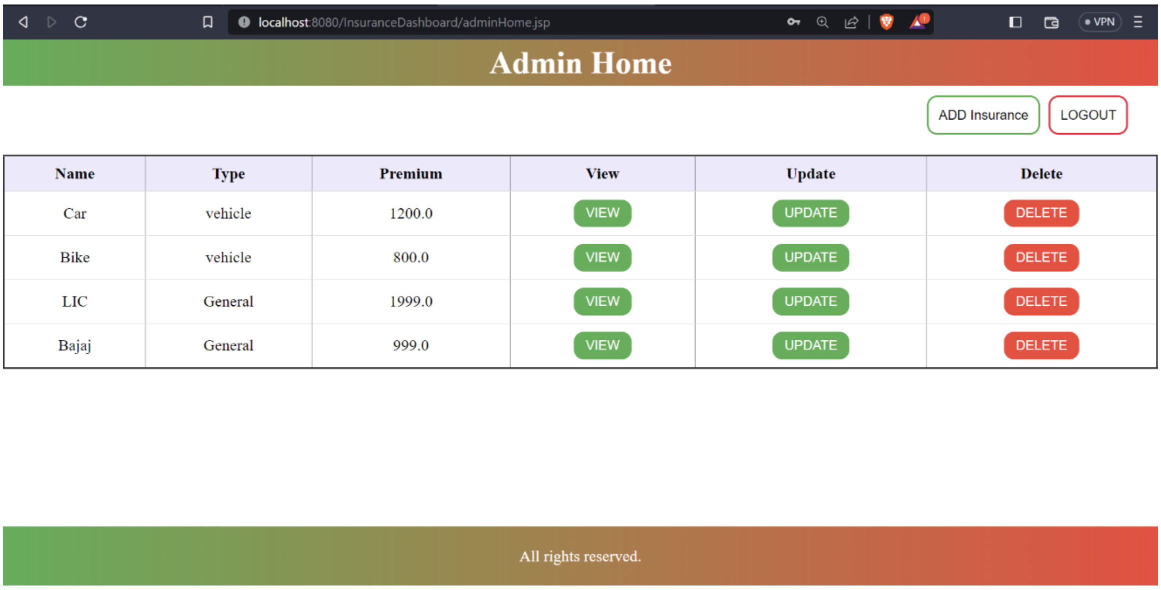
Username:

Password:

Type: Admin

Fig 5.1 Sign-in Page

5.2 Admin Home Page



localhost:8080/InsuranceDashboard/adminHome.jsp

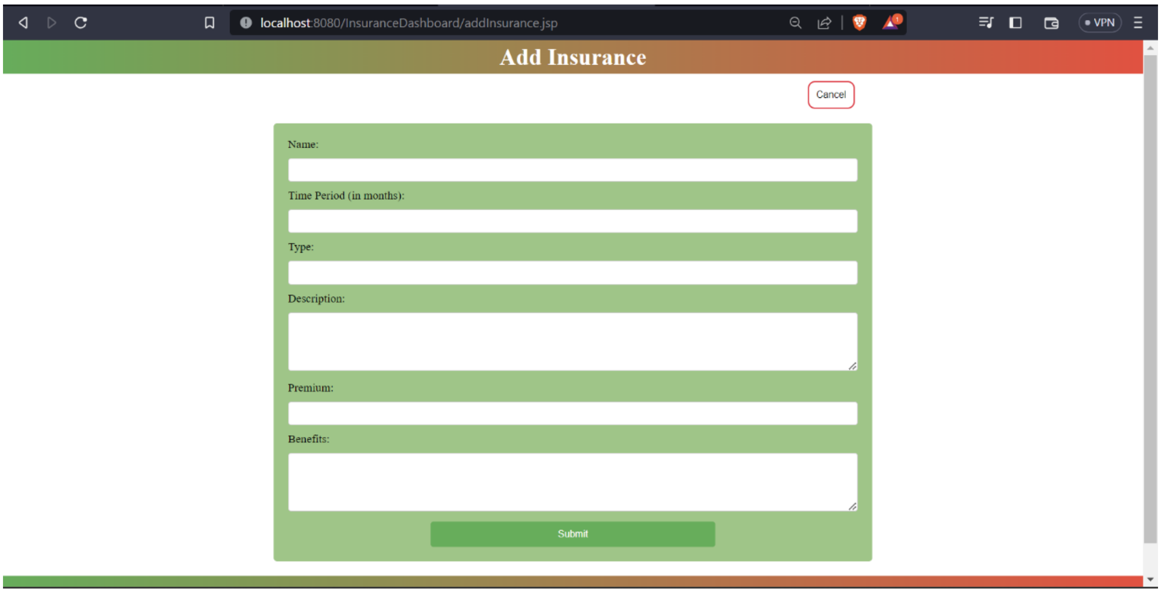
Admin Home

Name	Type	Premium	View	Update	Delete
Car	vehicle	1200.0	VIEW	UPDATE	DELETE
Bike	vehicle	800.0	VIEW	UPDATE	DELETE
LIC	General	1999.0	VIEW	UPDATE	DELETE
Bajaj	General	999.0	VIEW	UPDATE	DELETE

All rights reserved.

Fig 5.2 Admin Home Page

5.3 Add Insurance Page



Add Insurance

Name:

Time Period (in months):

Type:

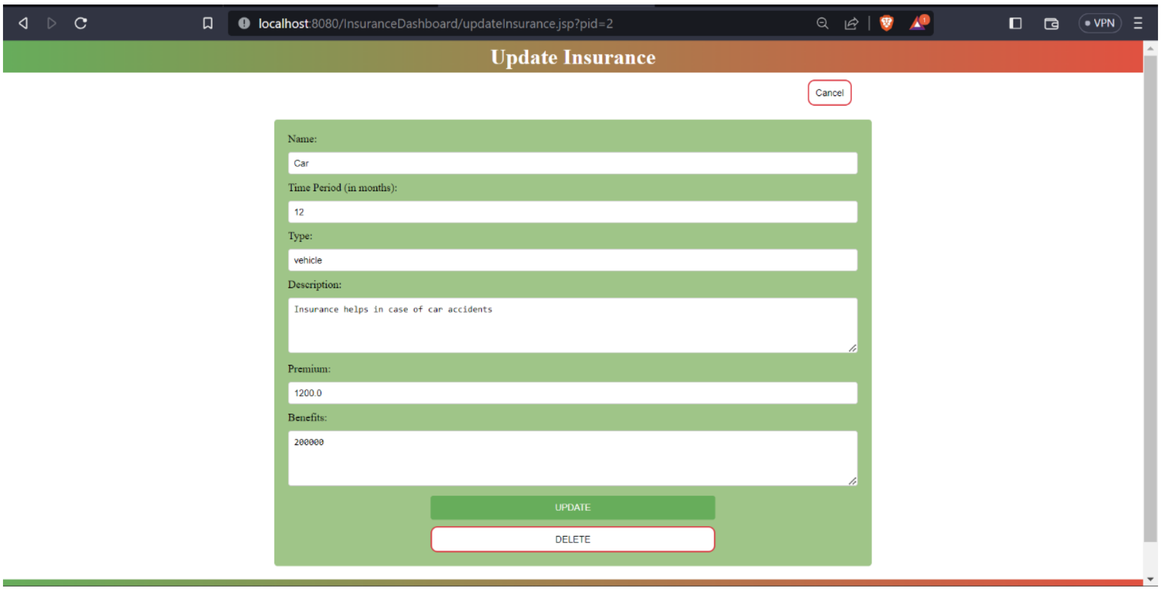
Description:

Premium:

Benefits:

Fig 5.3 Add Insurance Page

5.4 Update Insurance Page



Update Insurance

Name:
Car

Time Period (in months):
12

Type:
vehicle

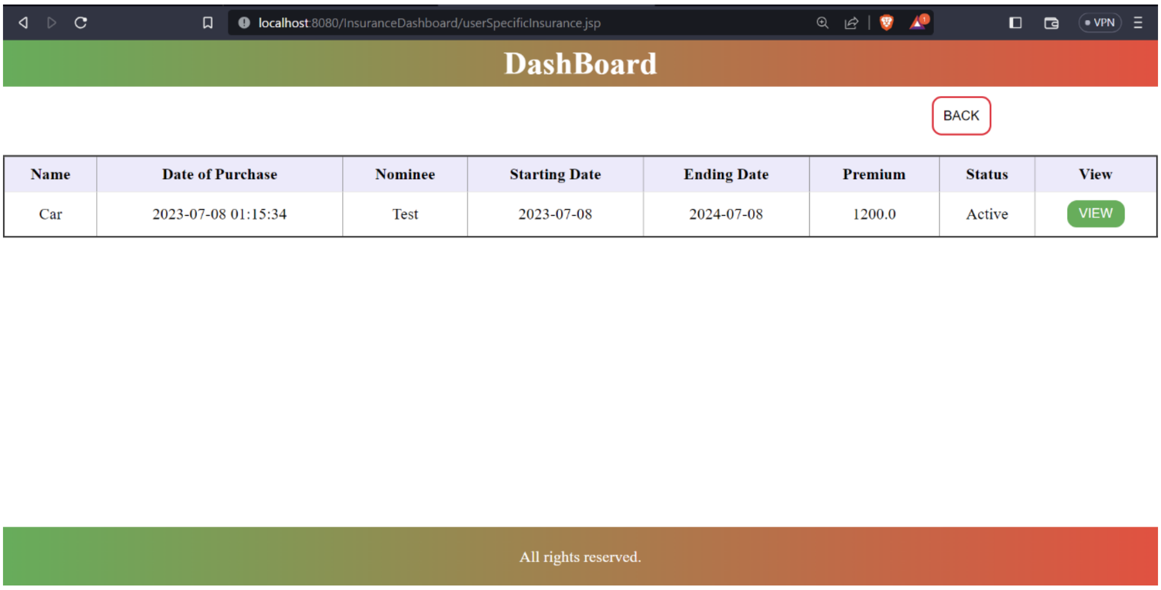
Description:
Insurance helps in case of car accidents

Premium:
1200.0

Benefits:
200000

Fig 5.4 Update Insurance Page

5.5 User Dashboard Page



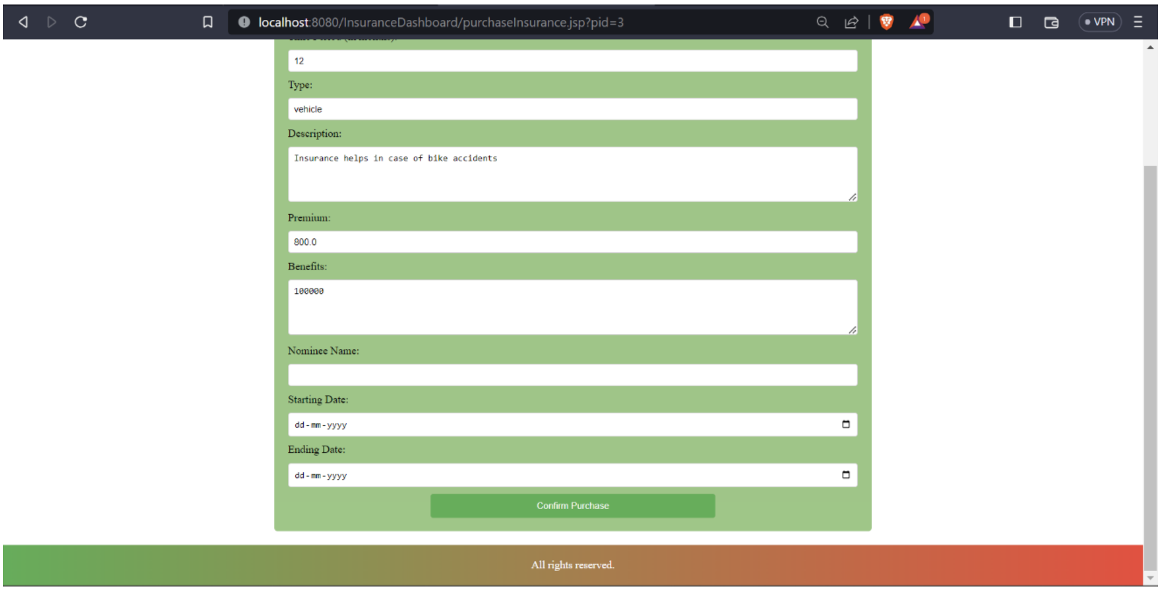
DashBoard

Name	Date of Purchase	Nominee	Starting Date	Ending Date	Premium	Status	View
Car	2023-07-08 01:15:34	Test	2023-07-08	2024-07-08	1200.0	Active	VIEW

All rights reserved.

Fig 5.5 User Dashboard Page

5.6 Purchase Insurance Page


localhost:8080/InsuranceDashboard/purchaseInsurance.jsp?pid=3
VPN

	12
Type:	vehicle
Description:	Insurance helps in case of bike accidents
Premium:	800.0
Benefits:	100000
Nominee Name:	
Starting Date:	dd-mm-yyyy
Ending Date:	dd-mm-yyyy

All rights reserved.

Fig 5.6 Purchase Insurance Page

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion

In conclusion, the Insurance Policy Dashboard project built using JSP and MySQL offers significant benefits for insurance companies. By centralizing lead capture, tracking, and policy management within a comprehensive dashboard, the system enhances sales efficiency, streamlines operations, and improves the overall customer experience. Administrators can effectively manage insurance policies, while users can easily purchase policies and access them through their personalized dashboard. The inclusion of customer reviews and feedback fosters transparency and trust, enabling informed decision-making. Overall, this project optimizes lead management, increases customer acquisition, and facilitates growth and success in the competitive insurance industry.

Future Enhancement

In future enhancements to the Insurance Policy Dashboard project, several features could be considered. Firstly, implementing an advanced search functionality would allow users to easily find specific insurance policies based on their criteria, such as coverage type, premium range, or specific benefits. Additionally, incorporating an interactive chatbot or live support feature could provide users with immediate assistance and answers to their queries. This would enhance the user experience and improve customer satisfaction. Another valuable enhancement could be the integration of a claims management system, allowing users to file and track insurance claims directly through the user portal. This would streamline the claims process, provide transparency, and further enhance the overall functionality and efficiency of the system.

REFERENCES

- Wikipedia contributors. (2023, July 9). Insurance. Wikipedia, the Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Insurance
- Asia Insurance Review- “Milliman Asia Papers”