Lambton College

CLASSIC COMPANY

INSURANCE

2022F-T4 CPL 5559 - WIL Project 24

COMPUTER SOFTWARE AND DATABASE DEVELOPMENT

Submitted By:
Sumandeep Kaur (C0859688)
Bharti Sagar(C0860181)
Jashanpreet Kaur(C0859791)

LAMBTON COLLEGE MISSISSAUGA

CERTIFICATE

This is to certify that the dissertation entitled **"CLASSIC INSURANCE COMPANY" is** submitted by Sumandeep Kaur (C0859688), Bharti Sagar(C0860181), Jashanpreet Kaur(C0859791) in their partial fulfilment of the requirement of the award of the degree of, Lambton College, Mississauga is a record of work carried out by them under my guidance and supervision. The result embodied in this thesis has not been submitted to any other university or institution for the award of any degree or diploma.

INTERNAL GUIDE:
Mr. Robin Singh

HEAD OF DEPARTMENT:
Dr. Olga Shugurova

ABSTRACT

The purpose of the project entitled "CLASSIC INSURANCE COMPANY" is a non-profitable organization and here, develop software which is user friendly simple and fast. It deals with the collection of insurance policies. Traditionally, it was done manually. The main function of the system is to provide a user-friendly ecommerce environment. The user can be entered to the website using a username and password. It is accessible by an administrator. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

INTRODUCTION

Classic insurance company is a one-of-a-kind website that would allow customers to take different insurance plans. This website assist in resolving different challenges and developing new cutting-edge cloud-based technologies to run their 50-year-oldinsurance company. They rely on your capacity to make informed and logical judgements to help them run their business. They understand insurance but aren't very tech savvy. They now have a relatively manual method, but they do employ an old piece of software created with File-maker Pro as well as have a choice to learn new recipes. It plans to connect millions of household customers with distributors. Classic insurance website is an online portal available 24 * 7 to internet friendly customers. It is a virtual online marketplace. People who don't know how to apply and claiming policies online, this company allow customers to get know about new policies via online customer care services.

1.1 OBJECTIVS:

- Admin and Users’ registration
- Entering user’s details
- Selecting policies and Claiming
- Adding to cart
- Payment
- Contact Us

1.2 SCOPE OF THE PROJECT :-

- User can just create the account in the website by entering his/her name, username, email, and password.
- Registered user can login and access his account
- Can choose the Policies, can add to cart and proceed to payment

All this work is done through the website by the admin and user

1.3 MODULES:

The entire project mainly consists of 2 modules, which are

Admin module

1.3.1 User module

1.3.2 Admin module:

- Manage the website, user
- Manage own profile
- Can add policies
- Upload the new Policy plan and announcement that's going on for the upcoming days

1.3.3 user module:

- Can select the Policies
- View the description
- Can purchase

CHAPTER 2

REQUIREMENT STUDY AND ANALYSIS

Requirement analysis is the first stage in the system engineering process and software development process. Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new

or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirement analysis is critical to success and development of project. Requirement must be actionable, measurable, testable, related to identify business opportunities, and different level of detail sufficient for system design.

The primary goal of the system analyst is to improve the efficiency of the existing system and for that specification requirement is very essential. For the development of the new system, a preliminary survey of the existing system will be conducted. Investigation is done whether the upgradation of the system into an application program could solve the problems and eradicate the inefficiency of the existing system.

2.1 EXISTING SYSTEM

Following are the drawbacks of the website that is used by company before our assistance :-

- Manual claim process
- Time consuming
- Less interaction between user and insurance company

They have maximum user traffic due to its intuitive designing and easy to understand and use feature. They have been developed a few years ago and have gained a lot of popularity during a short period of time.

Following are the features that are included in these websites.

- User can view the Policies
- Excellent use of communication technologies, digital technologies
- Online Registration of users
- Mailing option to the user

- Mail to agent for help in using the website

2.2 PROPOSED SYSTEM

Going through the related websites and requirements given according to the project, we are going to design a responsive website that will have appealing interface and easy to understand and use.

It will have the following functionalities.

- Admin and User' Registration
- Entering Policies details
- Easy payment
- User friendly UI
- Homepage
- Simple search of Insurance plans
- Different policies • Insurance Detail page
- Add-to-cart
- Checkout /payment gateway
- Latest Plans

So far, this website is similar to already existing websites

2.2.1 Advantages of proposed system

- Less time-consuming
- Easy to get the Plan details

Title :-Classic Insurance Company

Objective :- Creating a web application

Starting Date :- 15-09-2023

Ending Date :- 04-12-2023

Project Coordinator :- Robin Singh

Project Team:– Sumandeep Kaur, Bharti Sagar, Jashanpreet Kaur

USER STORIES

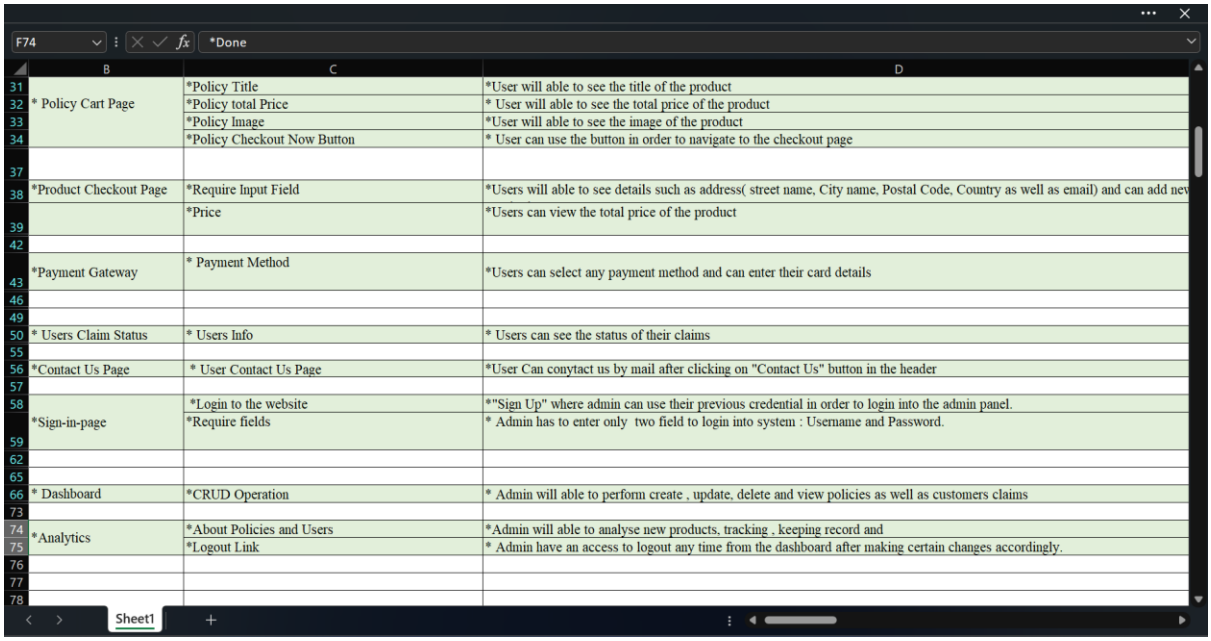
WEBPAGE NAME	FUNCTIONALITY	EXPLANATION
* Registeration Page	* New user or customer can register.	* New User/ Customer having no account can register on website and buy policies
	* Pass through validations.	* In order to make registration process smoother some sort of validation will be applied on Email, Password, Username and Cor
	* This page needs the basic information	* Customer/User can enter their First and Last name, Username , Email, Password and Contact Number into fields/column.
* Sign-In Page	* This will allows users to login into the website	* There is " My Account" section for users who already have an account and can use their old credentials to login
	* New users can create their account.	* User can register as new user using "Register" Link button on Home Page
	* Require Fields.	* In this the role of Admin needs to enter or fill two field in order to login into the system : Username and Password.
	* Policies Section	* Policies section will allow users where they can see new Policies that are displayed on website.
	* About Us Section	* This section includes the introduction part and background of company as well as the type of work that has been carried out ti milestones and acheivements.
	*Footer Section	* Footer will provide direct access links to social media, an organized place in which to include basic information like terms an policy and other legal issues.

Sheet1

F74 *Done

	B	C	D
19	* Policy Page	*Category Title	*User will able to see the title of the category page that will be displayed on the top.
20		*Policies	* Users will be able to view product image, name and one line desciption.
22			
23	*Policy Detail Page	*Policy Title	* Users will able to view the title of product
24		*Product Image	* User can see the image of the product
25		* Company's Name	* User will able to see the name of the company
26		* Policy total price	* customer can able see the price
29			
30	* Policy Cart Page	* Claim button	* User will able to go directly to the home page with a single click.
31		*Policy Title	*User will able to see the title of the product
32		*Policy total Price	* User will able to see the total price of the product
33		*Policy Image	*User will able to see the image of the product
34		*Policy Checkout Now Button	* User can use the button in order to navigate to the checkout page
37			
38	*Product Checkout Page	*Require Input Field	*Users will able to see details such as address(street name, City name, Postal Code, Country as well as email) and can add nev
39		*Price	*Users can view the total price of the product
42			
43	*Payment Gateway	* Payment Method	*Users can select any payment method and can enter their card details
46			
49			
50	* Users Claim Status	* Users Info	* Users can see the status of their claims
55			
56	*Contact Us Page	* User Contact Us Page	*User Can conytact us by mail after clicking on "Contact Us" button in the header
57			
58		*Login to the website	*"Sign Up" where admin can use their previous credential in order to login into the admin panel.
	*Sign-in-page	*Require fields	* Admin has to enter only two field to login into system : Username and Password

Sheet1



	B	C	D
31	* Policy Cart Page	*Policy Title	*User will able to see the title of the product
32		*Policy total Price	* User will able to see the total price of the product
33		*Policy Image	*User will able to see the image of the product
34		*Policy Checkout Now Button	* User can use the button in order to navigate to the checkout page
37			
38	*Product Checkout Page	*Require Input Field	*Users will able to see details such as address(street name, City name, Postal Code, Country as well as email) and can add nev
39		*Price	*Users can view the total price of the product
42			
43	*Payment Gateway	* Payment Method	*Users can select any payment method and can enter their card details
46			
49			
50	* Users Claim Status	* Users Info	* Users can see the status of their claims
55			
56	*Contact Us Page	* User Contact Us Page	*User Can conytact us by mail after clicking on "Contact Us" button in the header
57			
58	*Sign-in-page	*Login to the website	*"Sign Up" where admin can use their previous credential in order to login into the admin panel.
59		*Require fields	* Admin has to enter only two field to login into system : Username and Password.
62			
65			
66	* Dashboard	*CRUD Operation	* Admin will able to perform create , update, delete and view policies as well as customers claims
73			
74	*Analytics	*About Policies and Users	*Admin will able to analyse new products, tracking , keeping record and
75		*Logout Link	* Admin have an access to logout any time from the dashboard after making certain changes accordingly.
76			
77			
78			

3.1 IDENTIFICATION OF CLASSES

SYSTEM DESIGN AND DEVELOPMENT

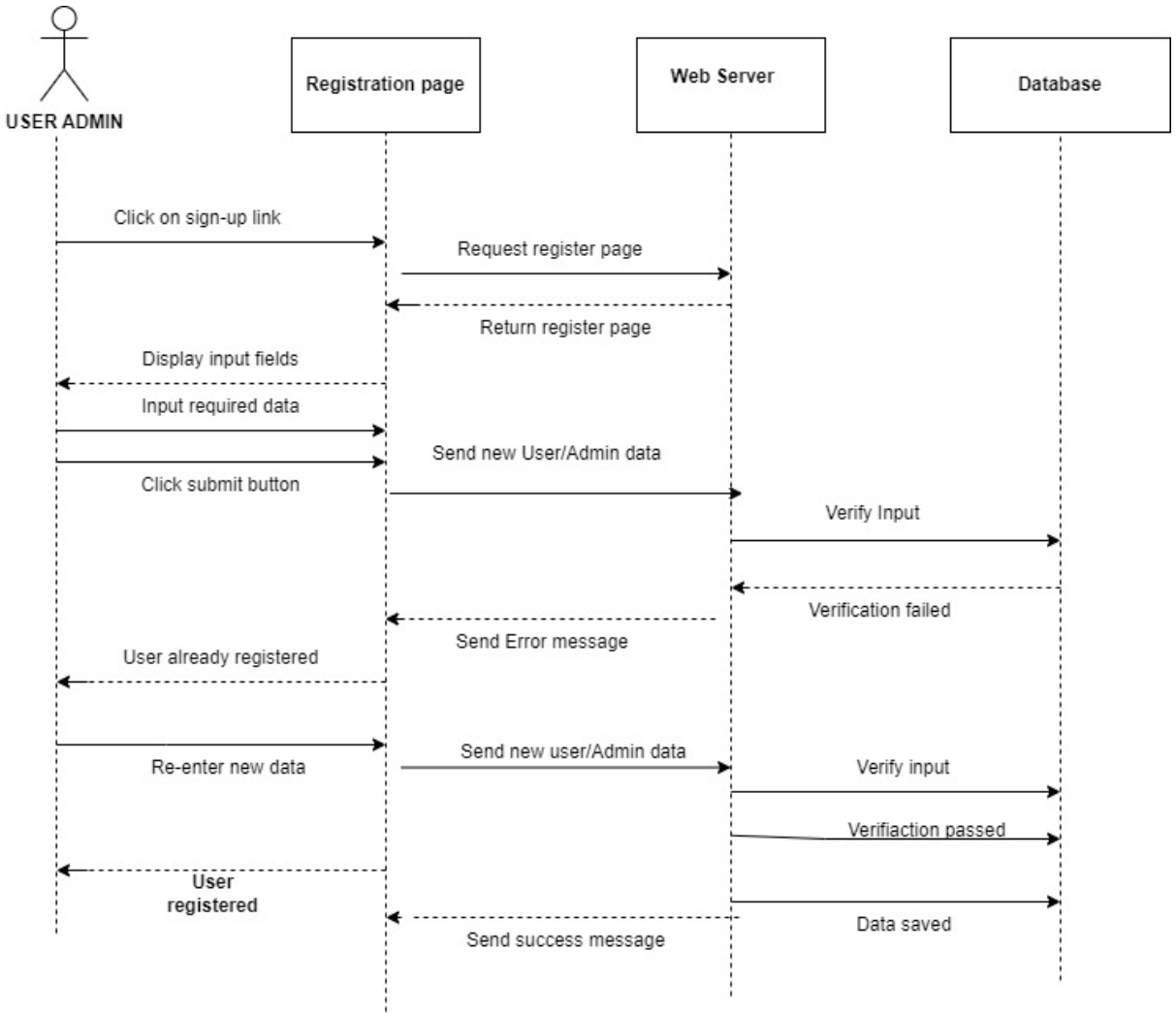
Software design usually involves problem solving and planning a software solution. This includes both low-level component and algorithm design and high-level, architecture design. After the purpose and specification of software are determined, software developers will design or employ designers to develop a plan for a solution.
System designs involve translating information requirements and conceptual design into technical specification and general flow of processing. After the user requirements are identified, related information is gathered to verify the problem and after evaluating the existing system a new system is proposed. The proposed system consists of various tables, their maintenance and report generation.

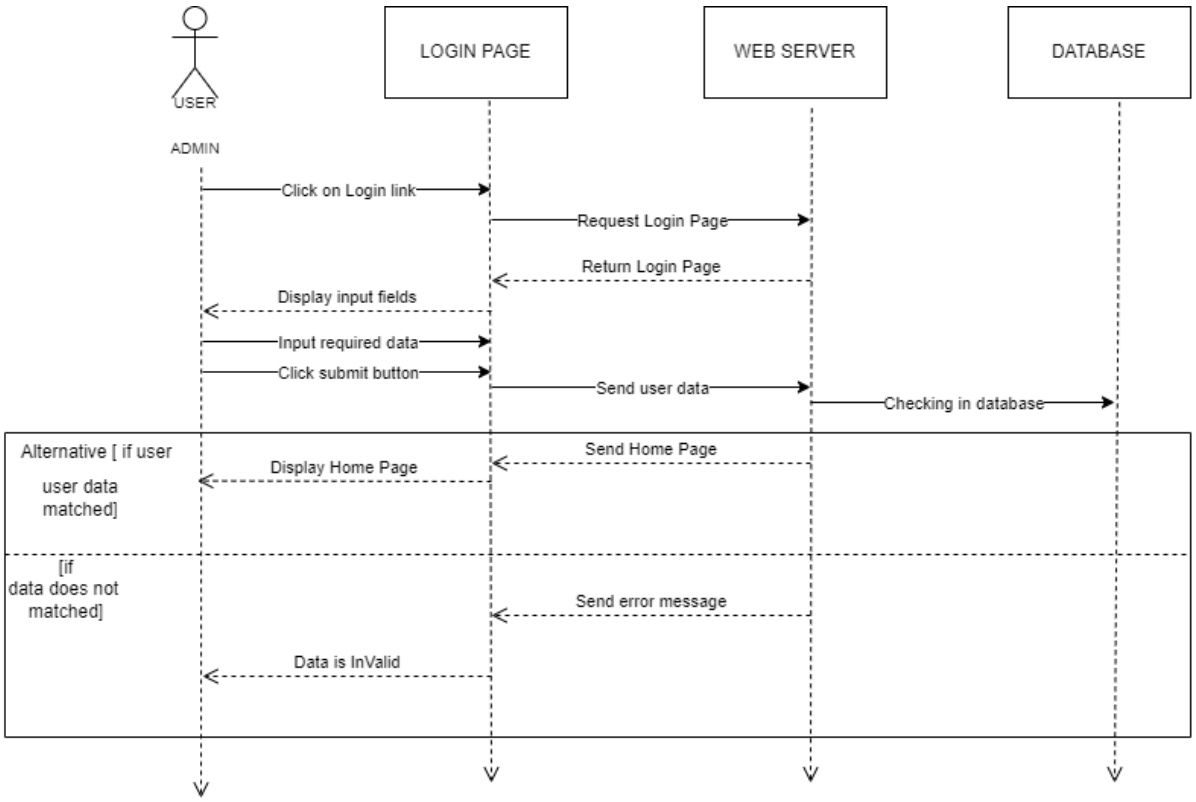
• Admin: The person who take care of the website.

• User: The person who registered in this website consult the patients.

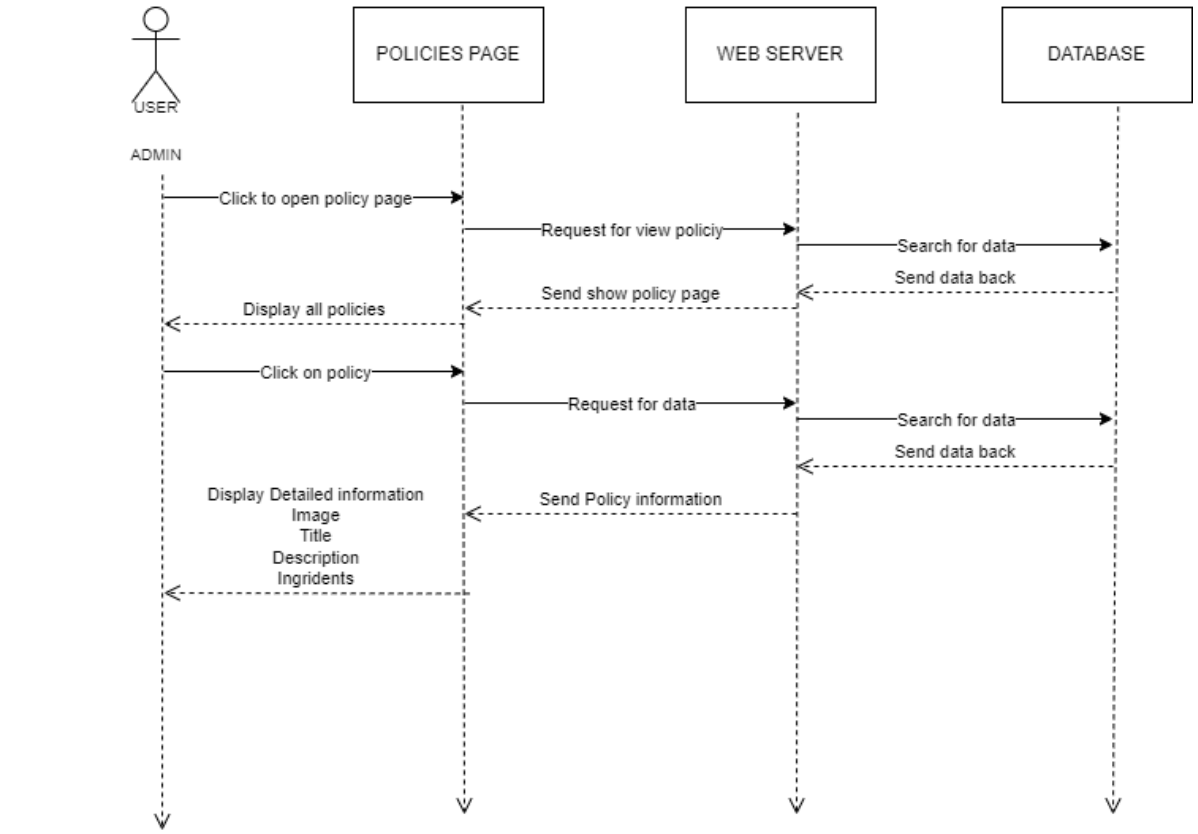
3.2 SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

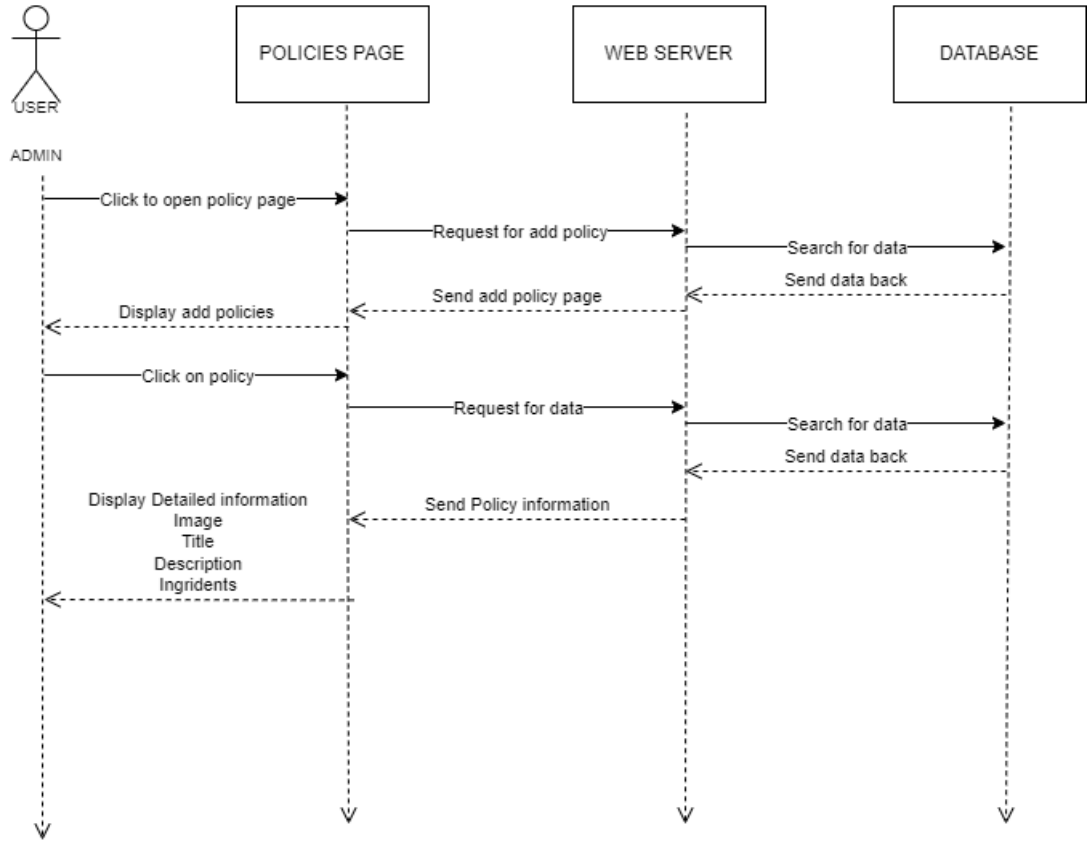


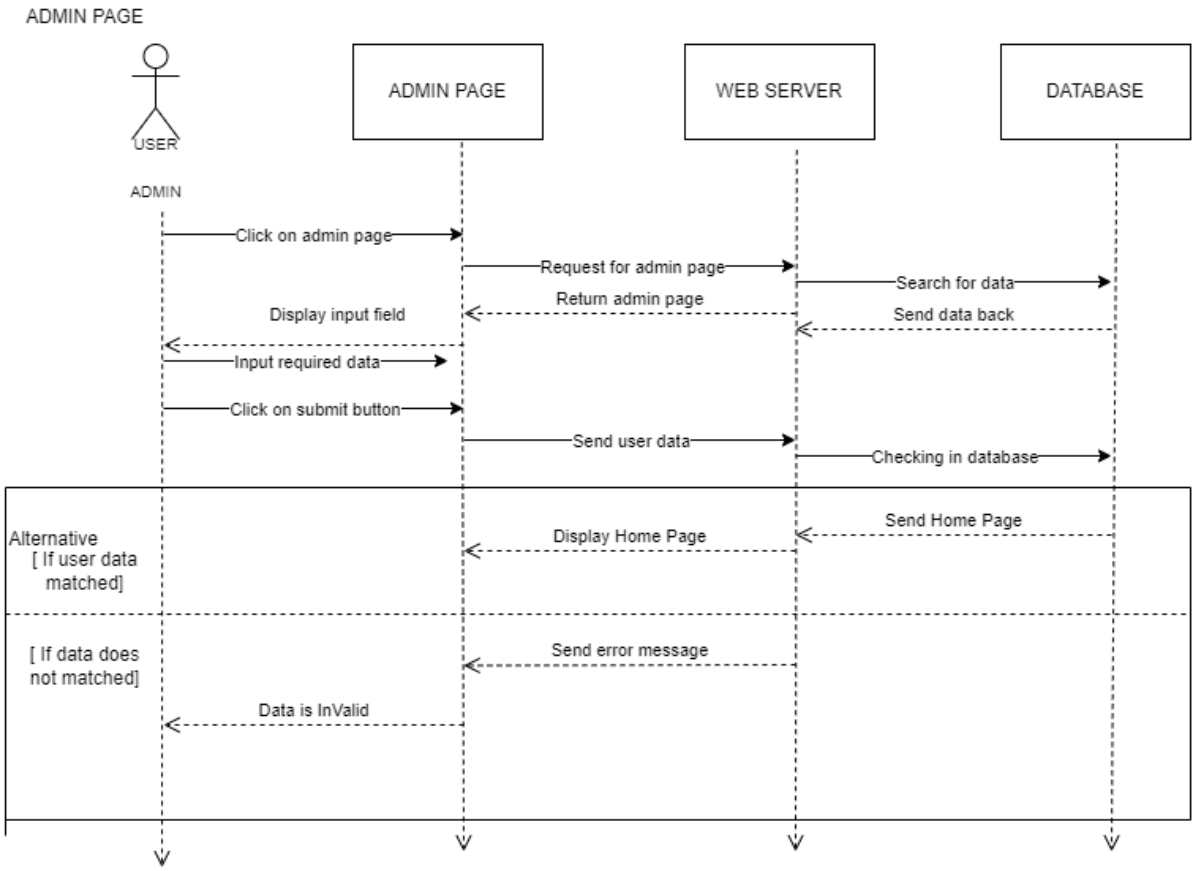


SHOW POLICIES



Add POLICIES





3.3 DATAFLOW DIAGRAM

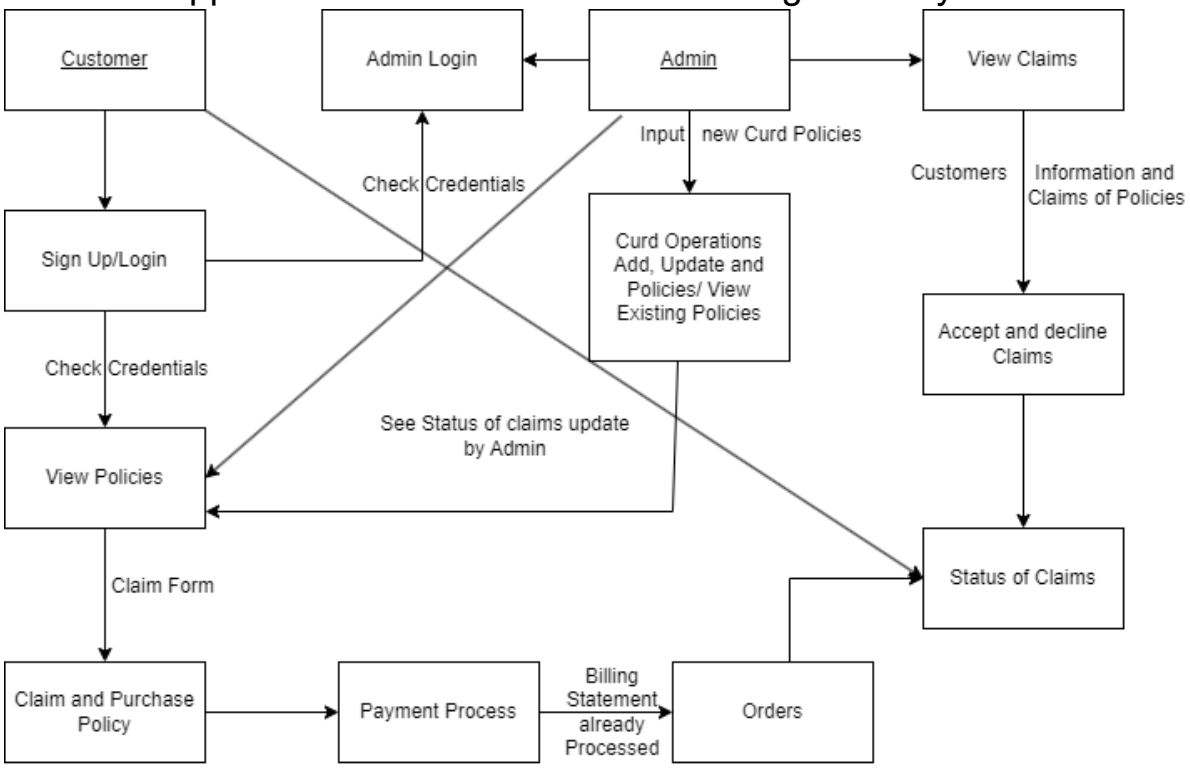
A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modelling its process aspects. Often, they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).
A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).
Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system’s evolution. With a data flow diagram, users can visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system’s data flow diagrams can be drawn up and compared with the new system’s data flow diagrams to draw comparisons to implement a more

efficient system. Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately influences the structure of the whole system from order to dispatch to report.

How any system is developed can be determined through a data flow Model.

3.4 DATABASE DESIGN

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationship map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system.



CHAPTER 4

REQUIREMENT SPECIFICATION

4.1 HARDWARE REQUIREMENTS:

The hardware requirements are the requirements of a hardware device. Most hardware only has operating system requirements or compatibility.

4.1.1 Hardware requirements for present project:

PROCESSOR : Intel Core i3
RAM : 1 GB
HARD DISK : 20 GB

The system requirements or software requirements is a listing of what software programs or hardware devices are required to operate the program or game properly.

4.2 TOOLS AND TECHNOLOGIES

Languages to be used:

- Front End: React
- Back End: Node
- Application Server: Express
- Database: Mongo db.

React:

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies React can be used

as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, react is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Nodejs:

Node.js is an open-source server environment. Node.js is a cross-platform and runs on Windows, Linux, Unix, and macOS. Node.js is a back-end JavaScript runtime environment. Node.js runs on the V8 JavaScript Engine and executes JavaScript code outside a web browser.
Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.
Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games). The Node.js distributed development project was previously governed by the Node.js Foundation and has now merged with the JS Foundation to form the Opens Foundation. Opens Foundation is facilitated by the Linux Foundation's Collaborative Projects program.

Mongo DB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License (SSPL) which is deemed non-free by several distributions.

EXPRESS JS

Express.js, or simply Express, is a back-end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.[3] It has been called the de facto standard server framework for Node.js.

CHAPTER 5

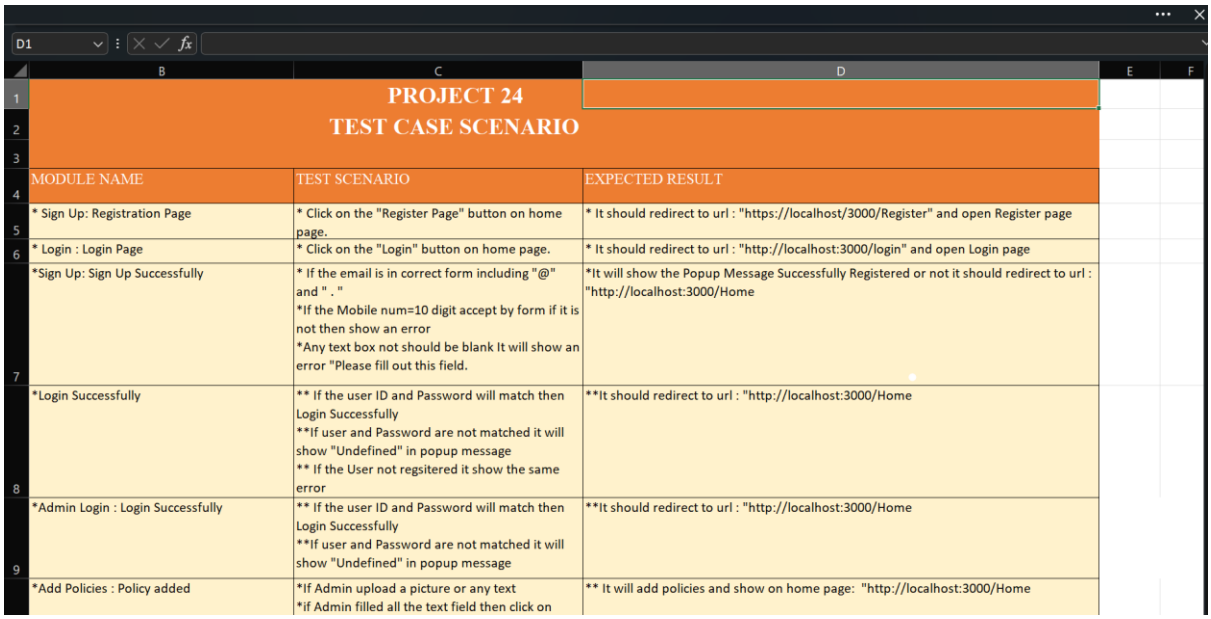
TESTING

The chapter testing describes various testing methodologies which are adapted and detailed view of test data within each database. During testing a program to be tested is executed with a set of test data and the output of the program for data is evaluated. In we-care health Matic virtual clinic, validation and checks have been done to ensure that the developed system is performing up to its expectation. Validation has been done to all the forms. Most of this checking is done at administrator side and in the user side. In all the pages appropriate messages are displayed when an administrator and user provide an invalid response.

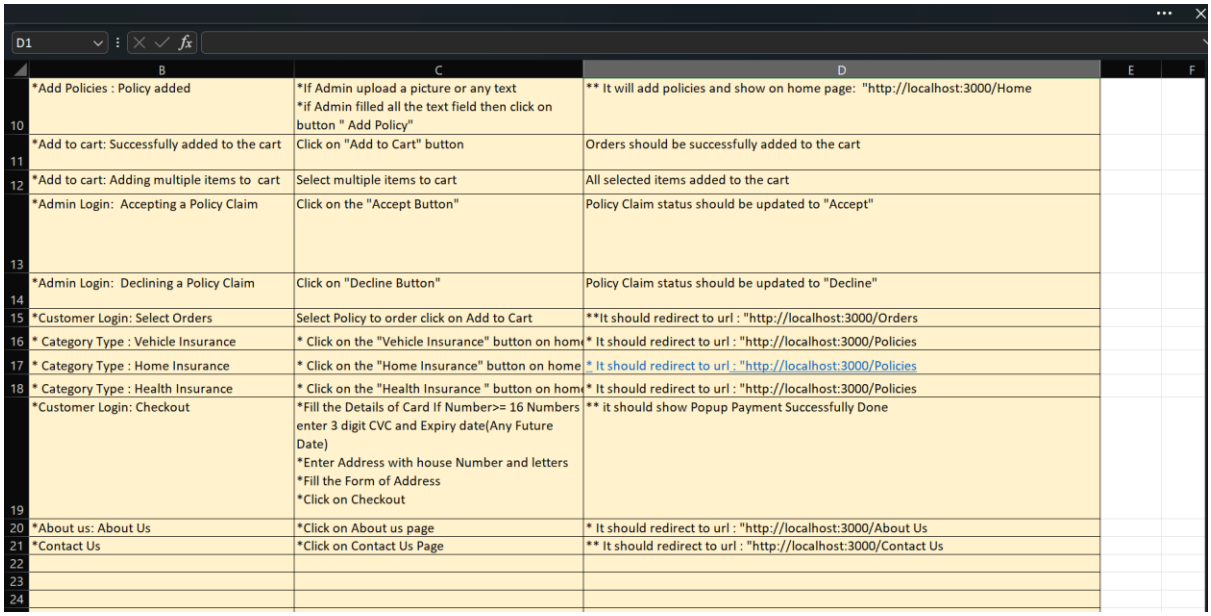
- **UNIT TESTING**

Unit Testing is a procedure used to validate that individual units of source code are working properly. A unit is the smallest testable part of application. In procedural programming a unit may be an individual program, function, procedure etc. While in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class, or derived/child class. Ideally, each test case is independent from others; mock objects and test harnesses can be used to assist testing a module in isolation. Unit testing is typically done by developers and not by software testers or end-users. For example, in our project the unit testing is done in the case of signup forms. We checked whether on entering the values they are being saved in the corresponding databases.

5.2 TESTING SCENARIOS

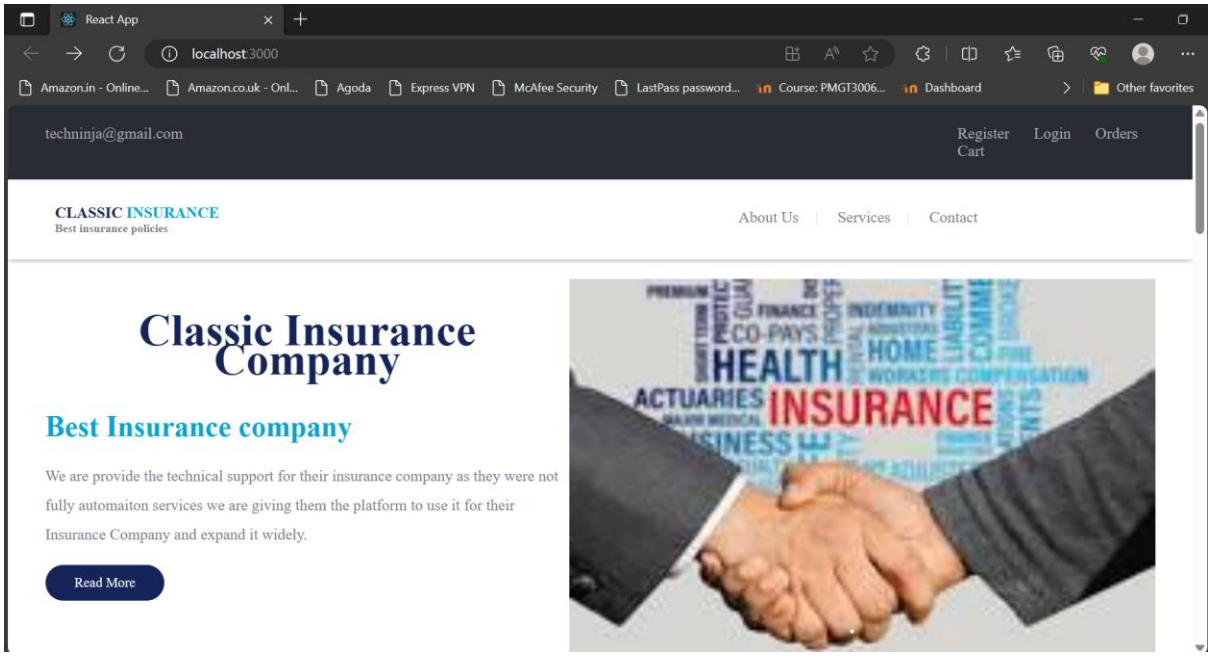


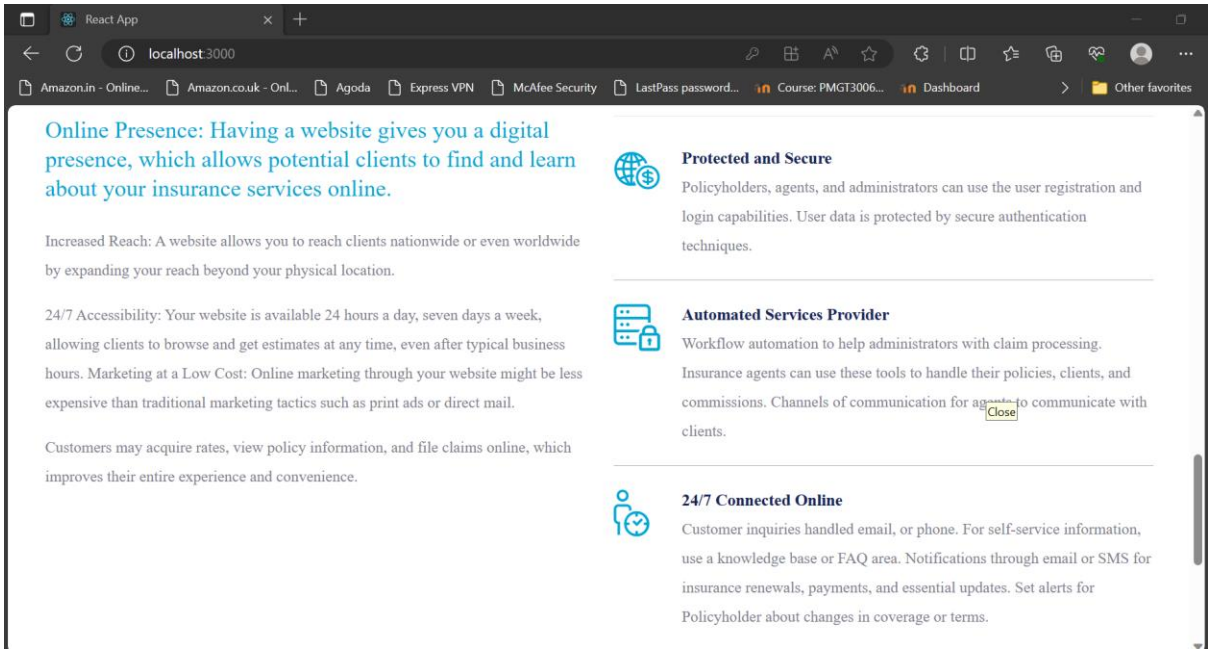
PROJECT 24 TEST CASE SCENARIO		
MODULE NAME	TEST SCENARIO	EXPECTED RESULT
* Sign Up: Registration Page	* Click on the "Register Page" button on home page.	* It should redirect to url : "https://localhost/3000/Register" and open Register page
* Login : Login Page	* Click on the "Login" button on home page.	* It should redirect to url : "http://localhost:3000/login" and open Login page
*Sign Up: Sign Up Successfully	* If the email is in correct form including "@" and " . " *If the Mobile num=10 digit accept by form if it is not then show an error *Any text box not should be blank It will show an error "Please fill out this field.	*It will show the Popup Message Successfully Registered or not it should redirect to url : "http://localhost:3000/Home
*Login Successfully	** If the user ID and Password will match then Login Successfully **If user and Password are not matched it will show "Undefined" in popup message ** If the User not regsitered it show the same error	**It should redirect to url : "http://localhost:3000/Home
*Admin Login : Login Successfully	** If the user ID and Password will match then Login Successfully **If user and Password are not matched it will show "Undefined" in popup message	**It should redirect to url : "http://localhost:3000/Home
*Add Policies : Policy added	*If Admin upload a picture or any text *if Admin filled all the text field then click on	** It will add policies and show on home page: "http://localhost:3000/Home

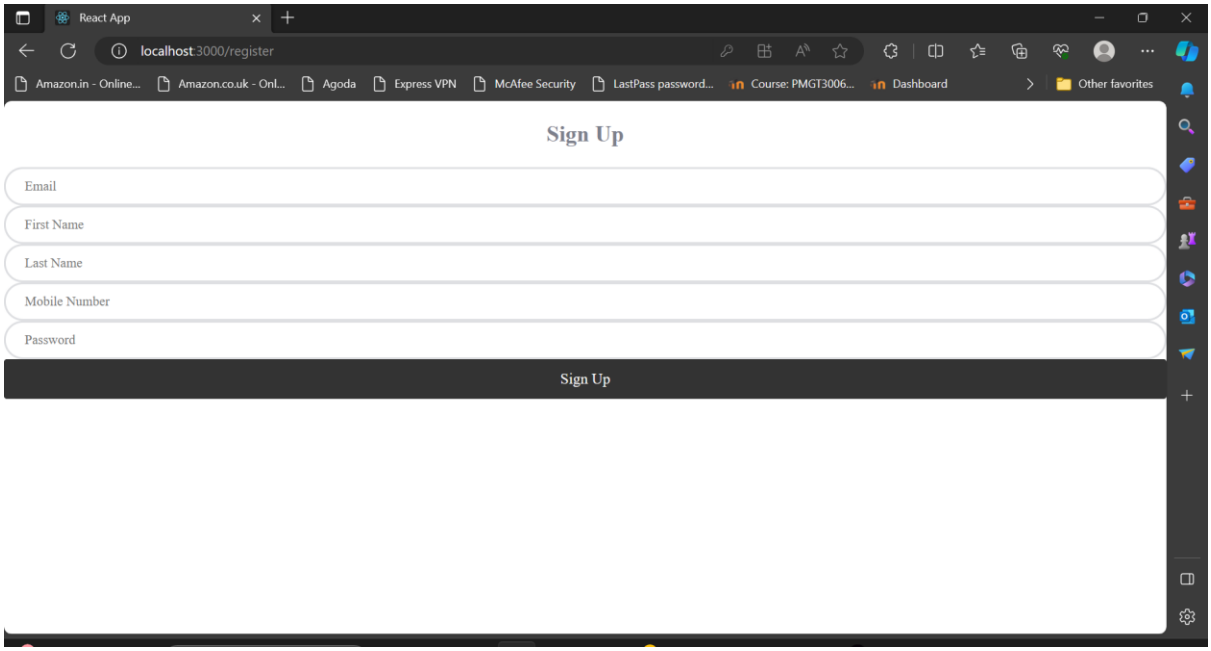


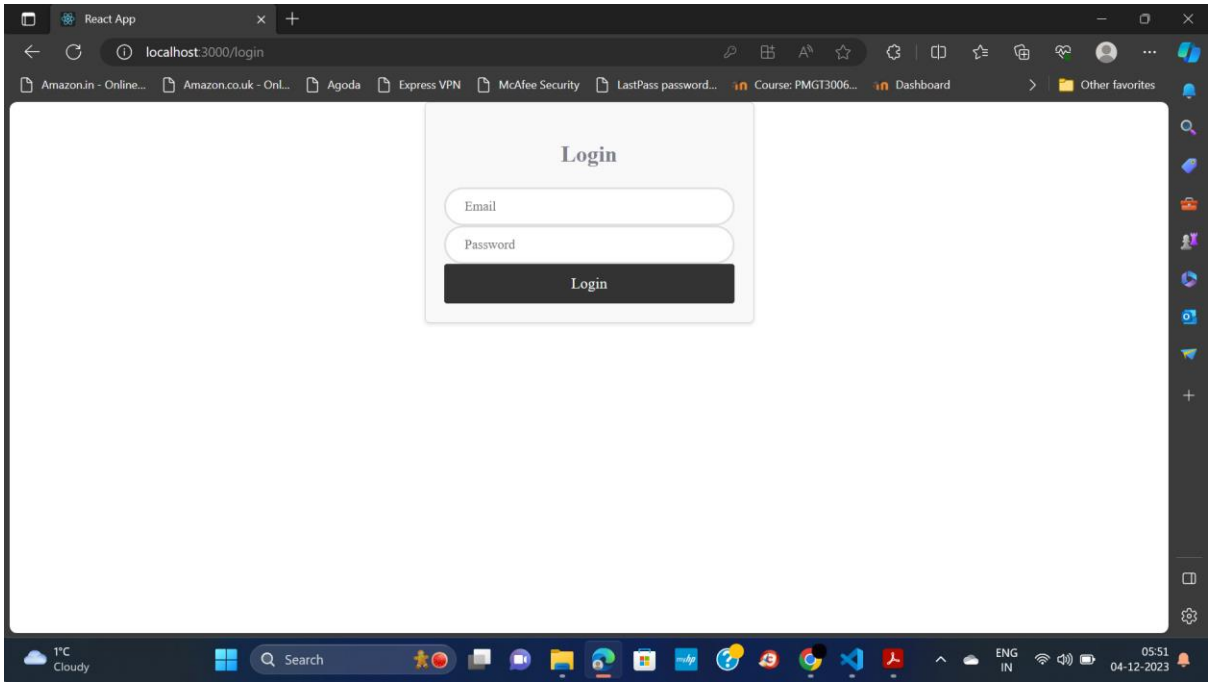
*Add Policies : Policy added	*If Admin upload a picture or any text *if Admin filled all the text field then click on button " Add Policy"	** It will add policies and show on home page: "http://localhost:3000/Home
*Add to cart: Successfully added to the cart	Click on "Add to Cart" button	Orders should be successfully added to the cart
*Add to cart: Adding multiple items to cart	Select multiple items to cart	All selected items added to the cart
*Admin Login: Accepting a Policy Claim	Click on the "Accept Button"	Policy Claim status should be updated to "Accept"
*Admin Login: Declining a Policy Claim	Click on "Decline Button"	Policy Claim status should be updated to "Decline"
*Customer Login: Select Orders	Select Policy to order click on Add to Cart	**It should redirect to url : "http://localhost:3000/Orders
* Category Type : Vehicle Insurance	* Click on the "Vehicle Insurance" button on hom	* It should redirect to url : "http://localhost:3000/Policies
* Category Type : Home Insurance	* Click on the "Home Insurance" button on home	* It should redirect to url : "http://localhost:3000/Policies
* Category Type : Health Insurance	* Click on the "Health Insurance " button on hom	* It should redirect to url : "http://localhost:3000/Policies
*Customer Login: Checkout	*Fill the Details of Card If Number>= 16 Numbers enter 3 digit CVC and Expiry date(Any Future Date) *Enter Address with house Number and letters *Fill the Form of Address *Click on Checkout	** it should show Popup Payment Successfully Done
*About us: About Us	*Click on About us page	* It should redirect to url : "http://localhost:3000/About Us
*Contact Us	*Click on Contact Us Page	** It should redirect to url : "http://localhost:3000/Contact Us

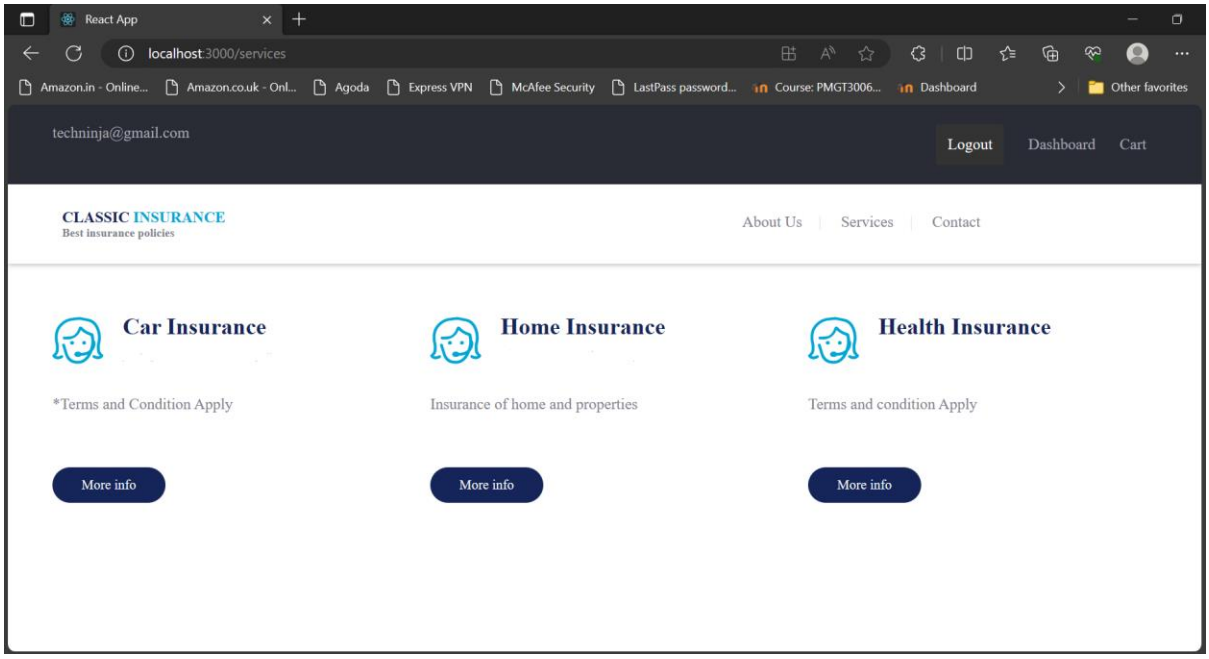
Screenshots:

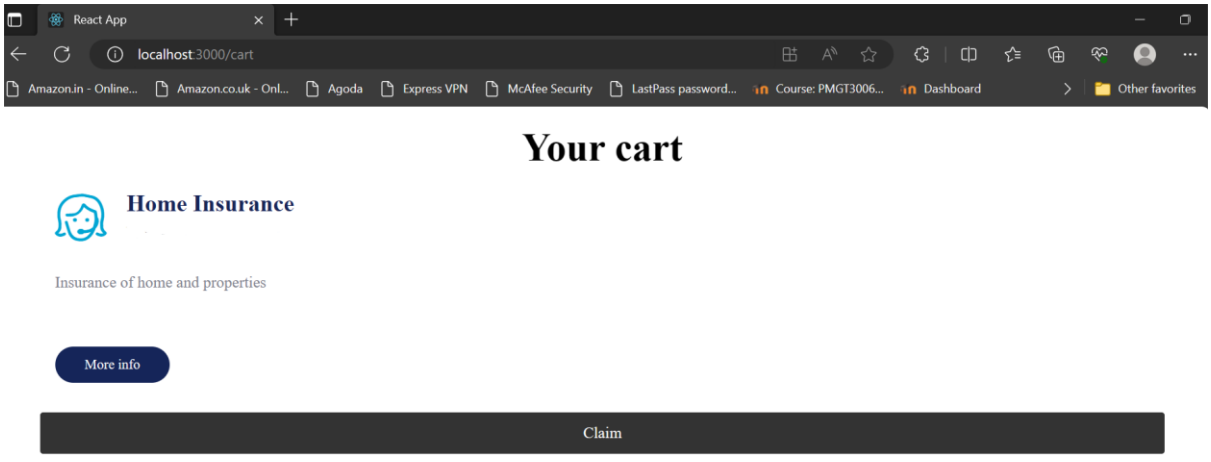


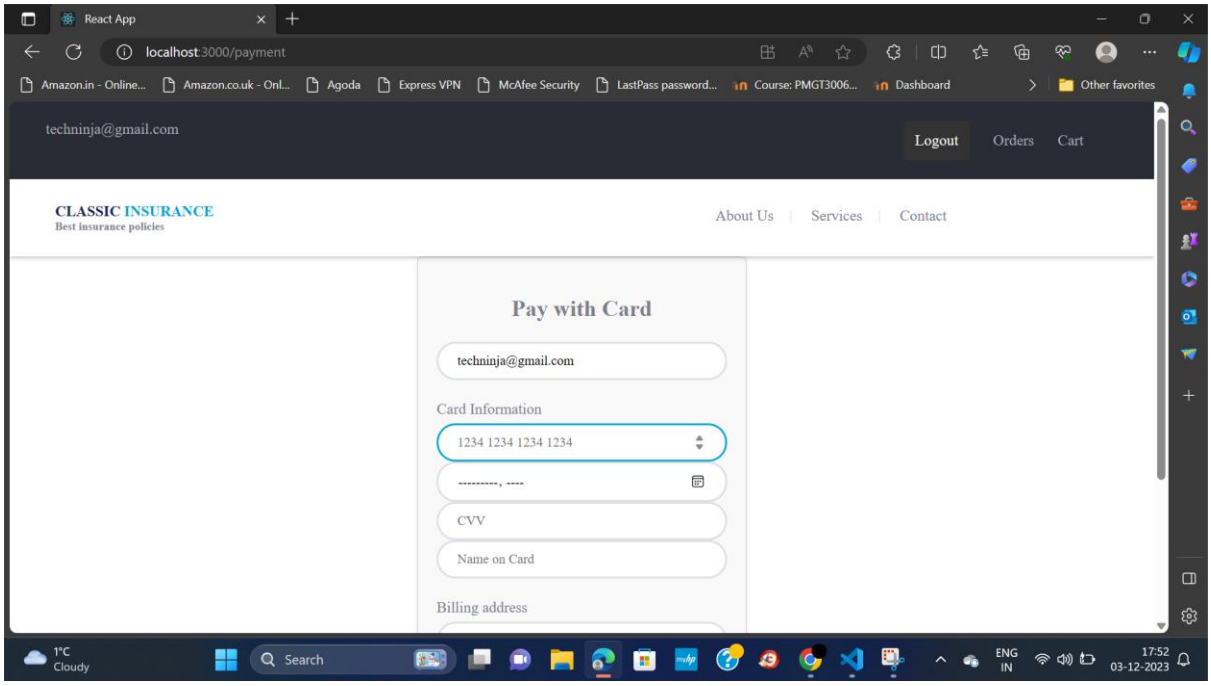


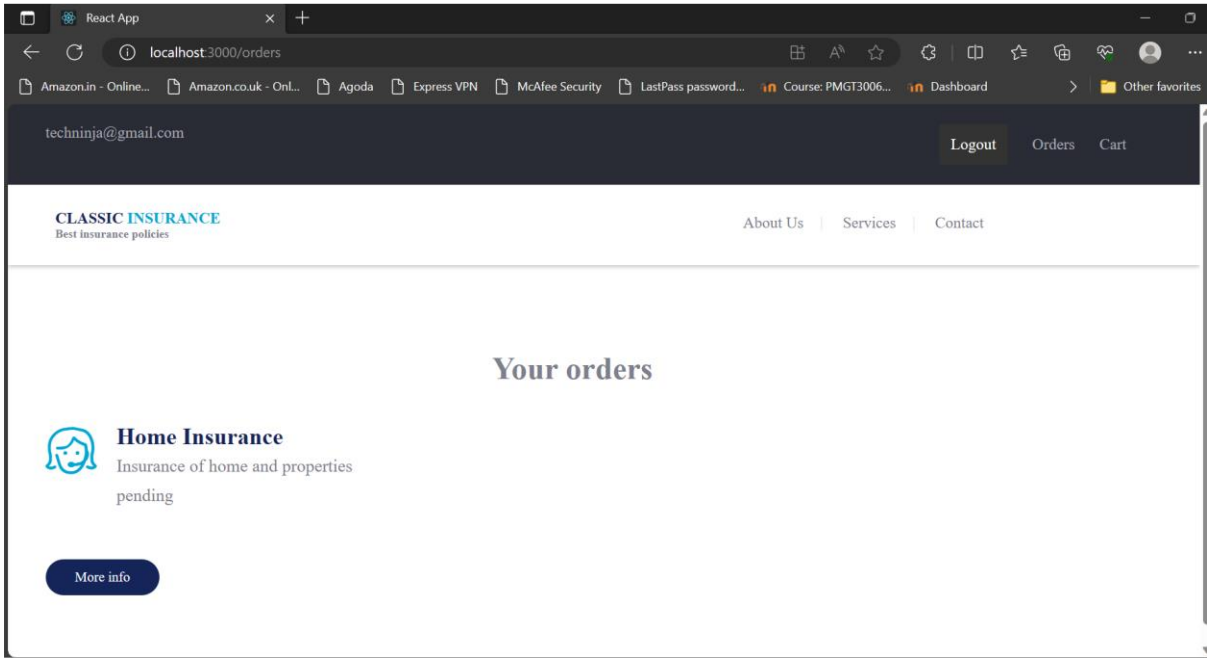

React App
localhost:3000/register
Amazon.in - Online...
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
Course: PMGT3006...
Dashboard
Other favorites
Sign Up
Email
First Name
Last Name
Mobile Number
Password
Sign Up

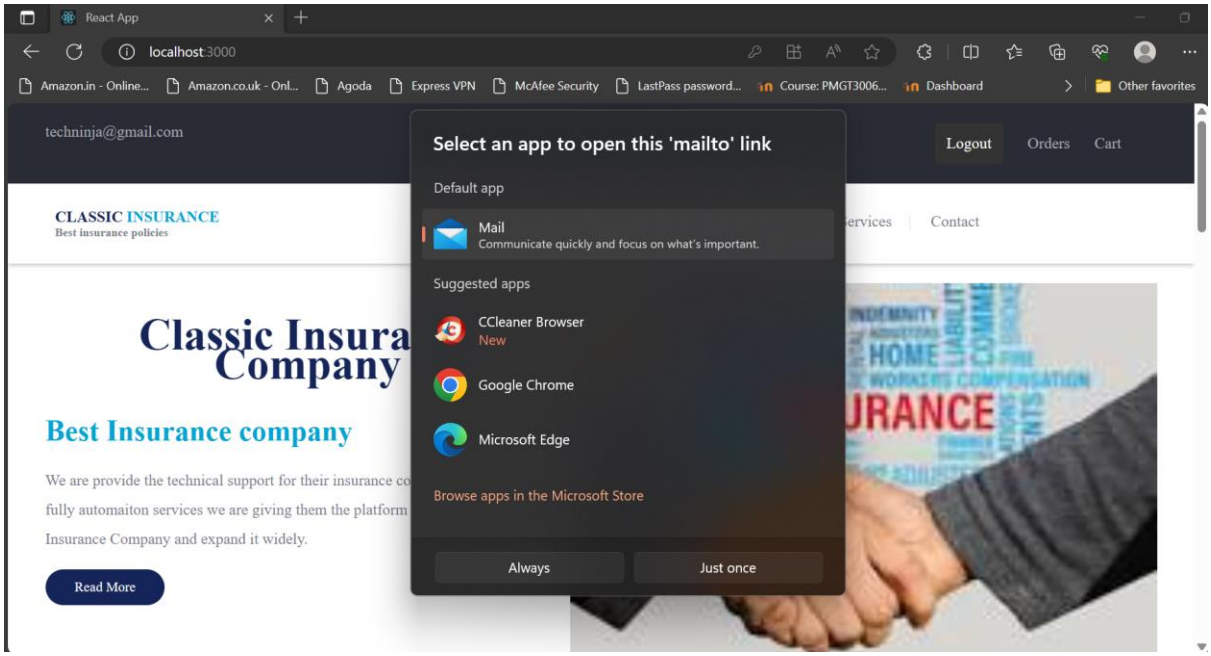

React App
localhost:3000/login
Amazon.in - Online...
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
Course: PMGT3006...
Dashboard
Other favorites
Login
Email
Password
Login
1°C
Cloudy
Search
ENG
IN
05:51
04-12-2023

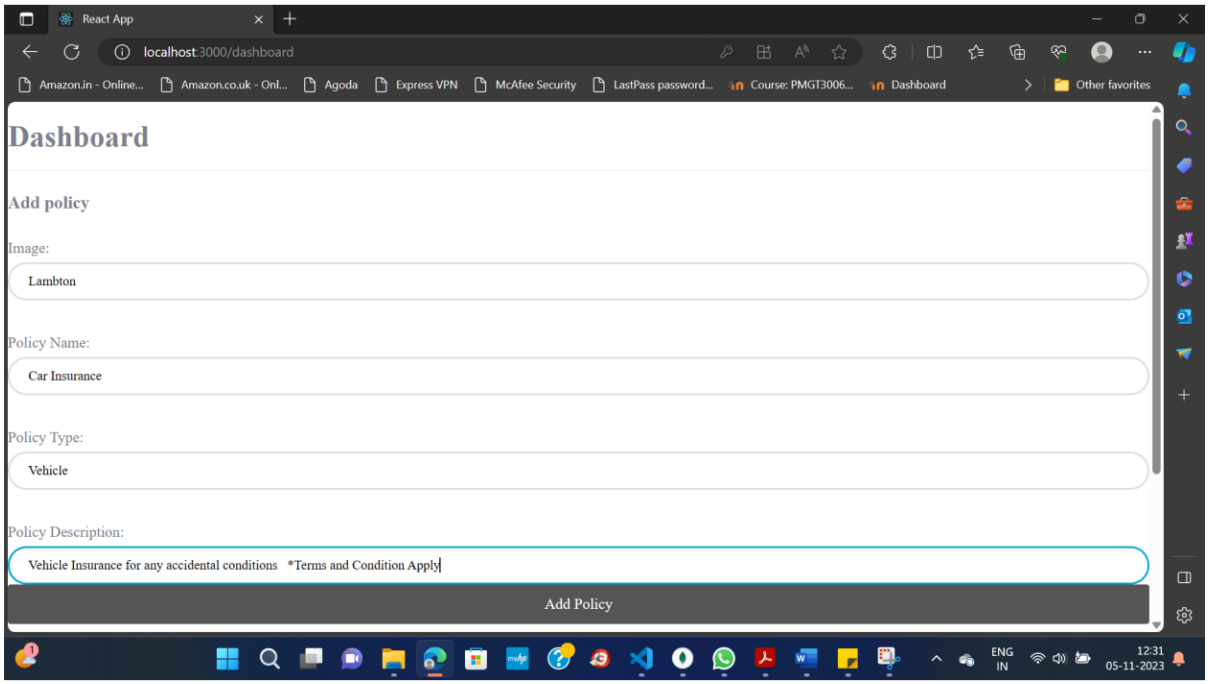

React App
localhost:3000/services
techninja@gmail.com
Logout
Dashboard
Cart
CLASSIC INSURANCE
Best insurance policies
About Us
Services
Contact
Car Insurance
*Terms and Condition Apply
More info
Home Insurance
Insurance of home and properties
More info
Health Insurance
Terms and condition Apply
More info

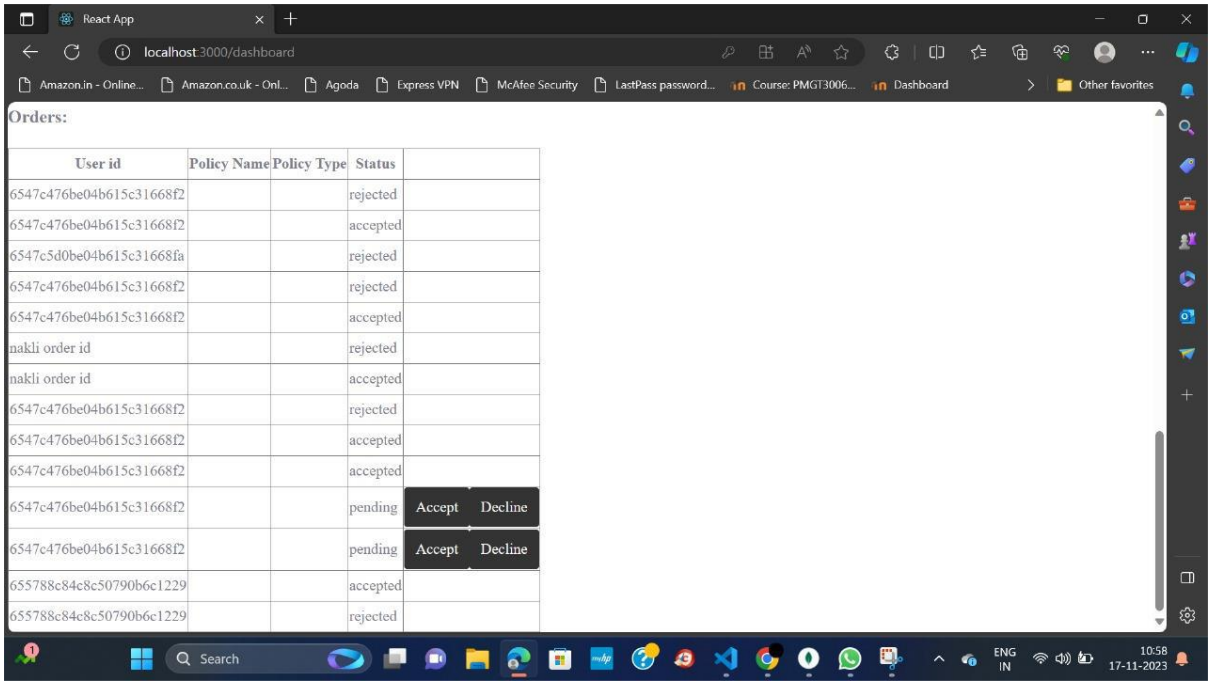

React App
localhost:3000/cart
Your cart
Home Insurance
Insurance of home and properties
More info
Claim

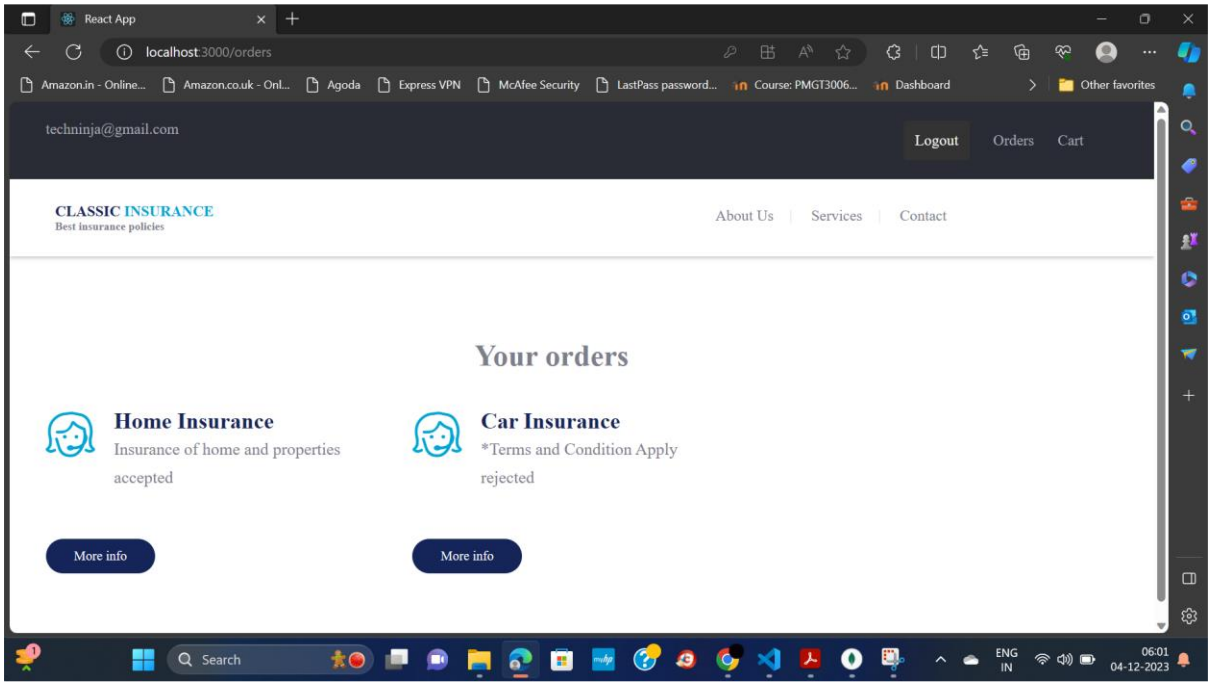

React App
localhost:3000/payment
techninja@gmail.com
Logout
Orders
Cart
CLASSIC INSURANCE
Best insurance policies
About Us
Services
Contact
Pay with Card
techninja@gmail.com
Card Information
1234 1234 1234 1234
CVV
Name on Card
Billing address


React App
localhost:3000/orders
techninja@gmail.com
Logout
Orders
Cart
CLASSIC INSURANCE
Best insurance policies
About Us
Services
Contact
Your orders
Home Insurance
Insurance of home and properties
pending
More info


React App
localhost:3000
techninja@gmail.com
Logout
Orders
Cart
CLASSIC INSURANCE
Best insurance policies
Contact
Classic Insura
Company
Best Insurance company
We are provide the technical support for their insurance co
fully automaiton services we are giving them the platform
Insurance Company and expand it widely.
Read More
Select an app to open this 'mailto' link
Default app
Mail
Communicate quickly and focus on what's important.
Suggested apps
CCleaner Browser
New
Google Chrome
Microsoft Edge
Browse apps in the Microsoft Store
Always
Just once


React App
localhost:3000/dashboard
Dashboard
Add policy
Image:
Lambton
Policy Name:
Car Insurance
Policy Type:
Vehicle
Policy Description:
Vehicle Insurance for any accidental conditions *Terms and Condition Apply
Add Policy


React App
localhost:3000/dashboard
Orders:
User id
Policy Name
Policy Type
Status
6547c476be04b615c31668f2 rejected
6547c476be04b615c31668f2 accepted
6547c5d0be04b615c31668fa rejected
6547c476be04b615c31668f2 rejected
6547c476be04b615c31668f2 accepted
nakli order id rejected
nakli order id accepted
6547c476be04b615c31668f2 rejected
6547c476be04b615c31668f2 accepted
6547c476be04b615c31668f2 accepted
6547c476be04b615c31668f2 pending Accept Decline
6547c476be04b615c31668f2 pending Accept Decline
655788c84c8c50790b6c1229 accepted
655788c84c8c50790b6c1229 rejected
Search
ENG IN
10:58 17-11-2023


React App
localhost:3000/orders
techninja@gmail.com
Logout
Orders
Cart
CLASSIC INSURANCE
Best insurance policies
About Us
Services
Contact
Your orders
Home Insurance
Insurance of home and properties
accepted
More info
Car Insurance
*Terms and Condition Apply
rejected
More info
Search
ENG IN
06:01 04-12-2023