

A Project Report on

# **INSURANCE PLANNER**

Submitted in partial fulfillment of the project for

**Second Year**

in

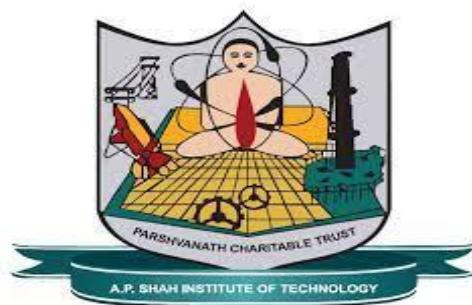
**Computer Engineering**

by

**Somiya Panikar (20102095)**  
**Namrata Narkhede (20102106)**  
**Hrishikesh Rane (20102157)**  
**Omkar Nimbalkar (20102131)**

Guide

**Prof. Merlin Priya Jacob**



**Department of Computer Engineering**  
**A.P. Shah Institute of Technology, Thane**

**Academic Year 2021-2022**



Parshvanath Charitable Trust's  
**A. P. SHAH INSTITUTE OF TECHNOLOGY**  
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)  
(Religious Jain Minority)

## CERTIFICATE

This is to certify that the Mini Project 1B entitled "**Insurance Planner**" is a Bonafede work of "**Somiya Panikar (20102095), Namrata Narkhede (20102106), Hrishikesh Rane (20102157), Omkar Nimbalkar (20102131)**" submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering in Computer Engineering**

---

Guide

Prof. Merlin Priya  
Jacob

Project Coordinator

Prof. Deepak Khachane

Head Of Department

Prof. Sachin Malave

Date:

# **Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---

(Signature)

---

**Somiya Panikar(20102095)**  
**Namrata Narkhede(20102106)**  
**Hrishikesh Rane(20102157)**  
**Omkar Nimbalkar(20102131)**

Date:

# Abstract

There's no doubt that knowing you're financially protected from a variety of unforeseen events can give you more peace of mind. These events can include sudden death, medical emergency, an accident or damage to your vehicle, property etc. Bearing the financial impact of these situations can burn a hole into your pocket. Thus, there is a pressing need for insurance for you and your family for proper coverage and financial support against all risks linked to your life, health and property. Therefore through our project ( Insurance Planner ) we intend to help our users (clients) settle any claims on their insurance, customize insurance programs and ensure that policy requirements are fulfilled by bringing in a system that involves an individual to enter his requirements or needs, then we filter through the various existing policies and provide them with the best policies. One of the major issues faced while buying insurance is whom to reach out to or which policies to buy. In offline mode, on reaching out to a particular company or agent where they insist on buying their own policies, in such a case there would be no room for comparison. One way to resolve this issue would be to bring in a system that allows the ability to draw comparisons and provides less risk of fraud, flexibility and easy online process at **zero percent (0%)** commission. Insurance planner brings an unbiased comparison of financial services from all the major insurance companies to our customers. Through this project, we hope to build an easy-to-use yet effective platform for Insurance planning.

Keywords - **online insurance planner**

# **Contents**

<b>1.</b>	<b>Introduction .....</b>	<b>1</b>
<b>2.</b>	<b>Literature Survey .....</b>	<b>2</b>
<b>3.</b>	<b>Problem Statement.....</b>	<b>4</b>
<b>4.</b>	<b>Objective &amp; scope.....</b>	<b>5</b>
<b>5.</b>	<b>Experimental Setup.....</b>	<b>6</b>
	<b>    5.1Hardware requirement</b>	
	<b>    5.2Software requirement</b>	
<b>6.</b>	<b>System Design.....</b>	<b>7</b>
	<b>    6.1UML Diagram</b>	
	<b>    6.2Algorithm/Process(with inputs and outputs)</b>	
<b>7.</b>	<b>Implementation.....</b>	<b>23</b>
	<b>    7.1Code</b>	
	<b>    7.2Screenshots</b>	
<b>8.</b>	<b>Result.....</b>	<b>27</b>
<b>9.</b>	<b>Conclusion.....</b>	<b>28</b>
<b>10.</b>	<b>References.....</b>	<b>29</b>
<b>11.</b>	<b>Annexure 1- Project Planning (Using Gantt chart) .....</b>	<b>30</b>
<b>12.</b>	<b>List of figures .....</b>	
	<b>    12.1 UML Diagram</b>	
	<b>    12.2 Sequence Diagram</b>	

**12.3 DFD**

**12.4 Gantt Chart**

**13. Acknowledgement ..... 31**

# **Chapter 1**

## **Introduction**

Insurance is a contract, represented by a policy, in which an individual or entity receives financial protection or reimbursement against losses from an insurance company. The company pools clients' risks to make payments more affordable for the insured. Insurance policies are used to hedge against the risk of financial losses, both big and small, that may result from damage to the insured or her property, or from liability for damage or injury caused to a third party. The policy contains crucial information such as the policyholder's name, details of the cover, breakdown of the premium, and other such details. Every insurance policy has a start date and an end date, which is known as the coverage period. The insurance company only covers incidents that occur during this period. The coverage period can be extended by renewing the policy. In simple words, insurance or an insurance policy is a promise. In slightly complex terms, it is a contract. In technical terms, it is a legal document that guarantees financial assistance in challenging situations and is governed by terms and conditions and is offered by the insurer to a policyholder, in exchange for a premium.

# Chapter 2

## Literature Survey

[1]The paper presents a Smart Insurance decision support system that works on the principle of Crowdsourcing. This principle acts like a problem solving mechanism that obtains required ideas, services or content from a large group of people. It is a web based application that supports decision making algorithms. The algorithm performs analysis on the data provided. It gives opinions to the customers based on the details filled by them like personal information, financial details, etc. The proposed web application provides the affordable insurance policies based requirements of the customer. The proposed system provides a four modules division : input data, perform analysis, display policies and registration. Hence, the system has proposed and implemented a crowd sourcing scheme to gather information from people and give opinions to the customers. And the policy maker algorithm based on the decision making system is implemented. And hence provides a common platform for different types of customers

[2] This study suggests a big data-based insurance inquiry and claim system. This method was created using the agile model. Case information management, investigation information management, claim settlement management, intelligent insurance advice, and other features are included. Intelligent customer service and other core functions, relying on Web, mobile applications, and social networks for data collection, filtering, and real-time analysis of the collected large data information, using Hadoop framework to convert structured data into semi-structured or unstructured data, data storage, advanced analysis, prediction model Type, and visual query, mining and analyzing the extracted data, and finally displaying the results afterward. The proposed web application provides the affordable insurance policies based requirements of the customer. The implemented system will assist the client in claiming monthly or quarterly premiums depending on many parameters such as the number of family members to insure, the family's total income, the projected amount of insurance, the family's age group, and

so on. We created a website to make the procedure more user-friendly and efficient. The claim will be confirmed once the client has been properly identified and verified. As a result, it provides a common platform for various types of customers to purchase insurance.

[3] The paper presents a study of potential of Online Insurance Market. This paper use data distribution method to estimate the growth of the Insurance Market. The online insurance is an innovation of the insurance marketing method, according to the diffusion of innovation theory, its diffusion velocity and process are affected by many factors in which market potential is an important factor. This model assumes that consumers have the insurance awareness and have online payment experiences. The reason for the distribution fitting is that the general public data is counted in sections, and different reports released agencies adopt different piecewise point. It is seen that Chinese online insurance market potential is in a relatively low proportion then its calculated in the paper. At the end this paper concludes that Chinese online insurance will grow rapidly in the near future, whether the insurance companies and intermediaries, or the third-party provider of technology platform, they all should seize this opportunity, promote the development of online insurance jointly.

# **Chapter 3**

## **Problem Statement**

Since risk has formed an integral part of our world today, the need arises for insurance in order to combat such risk in our society. 4 major problems arise when trying to create a system to manage such risk, they are risk identification, evaluation, control and financing. The multidisciplinary nature of risk management requires input from all the arrears of the organization. Therefore, the task of identifying and evaluating risks is a problem. Risk control including avoidance and reduction clearly falls within the ambit of corporate safety policy. Protection of property and personnel through effective risk control measures, assumes great significance, particularly in the light of the opportunity costs due to occurrence of a risk. Risk financing has developed into an important technique of effective risk management, whereby costs and benefits of various alternatives are analyzed before arriving at the final decision on what type of insurance to be used. The existing system is the manual system. The manual system is prone to error. It is time consuming. It is very difficult for a person to produce the report. There are chances for changing the scheme report and doing malpractice. This system involves a lot of manual entries with the applications to perform the desired task. Usage of papers in the payment process leads to less efficiency, less accuracy and less productivity.

- Increasing expenditure for papers shuffling and storage.

- Increasing laborers and hence errors.
- Less control of Amounts.
- Time delay between the payment and its receipt.
- Persons who are present in different parts of the world cannot transact efficiently.

# **Chapter 4**

## **Objective and Scope**

To develop a platform that makes the process of obtaining and claiming insurance easier. To create an easy-to-use but powerful insurance planning platform. The main objective of the developed system is to allow admin users to register insured persons with their name, email id,aadhar details and other personal information.This process contains the user registration form which is used to apply for an insurance policy online.

Online insurance is becoming more popular. Customers will be able to easily purchase and claim insurance, as well as compare coverage. Customers save time because they don't have to find an agent for their insurance and don't have to give the papers to the agent because they buy insurance online. Instead, they simply need to upload the papers' pdf or jpg files to the appropriate location, which is simple and handy for the customer. There is no commission in online insurance. There is no such thing as a third party. The insurance business as a whole is growing at a rapid pace.Insurance planner brings an unbiased comparison of financial services from all the major insurance companies to our customers.

# **Chapter 5**

## **Experimental Setup**

### **5.1 Hardware Requirement**

- Processor - INTEL CORE i5
  - RAM – 8GB
  - Hard Disk – 1TB
  - Display Card –GTX 1660 TI
  - Mouse – Logitech Serial Mouse
  - Keyboard – Standard 104 Enhanced Keyboard

### **5.2 Software Requirement**

- Web Server – Heroku
- Browser – Internet Explorer
- Server-side scripting – Python flask Library
- Database – SQLite
- Language –HTML, CSS, JS(BOOTSTRAP)

# **Chapter 6**

## **System Design**

### **6.1.1 UML Diagram**

It is a structural diagram that depicts a static view or structure of a system. It presents an outline for the system. It stresses the elements to be present that are to be modeled.

Through this we can also show the flow between modules, and how the data flows between them.

What can we achieve through this?

- Improved integration between structural models like class diagrams and behavior models like activity diagrams.
- Added the ability to define a hierarchy and decompose a software system into components and subcomponents.

## 6.1.2 Sequence Diagram

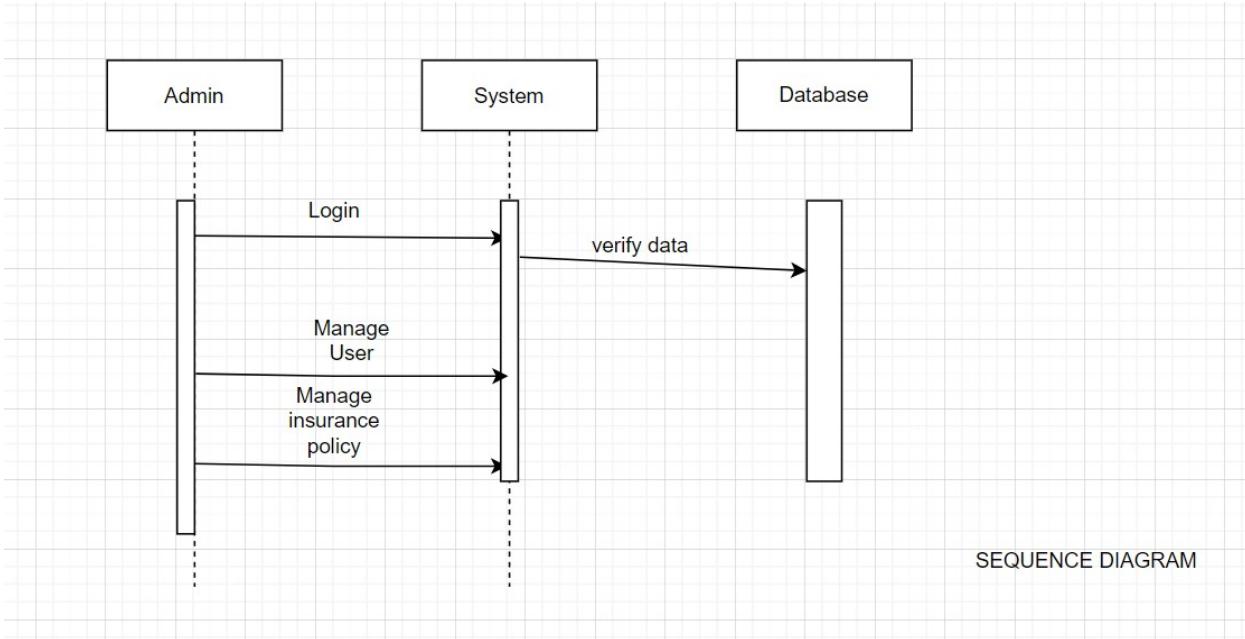


Fig 6.1.2 The sequential flow of our website

## 6.1.3 DFD

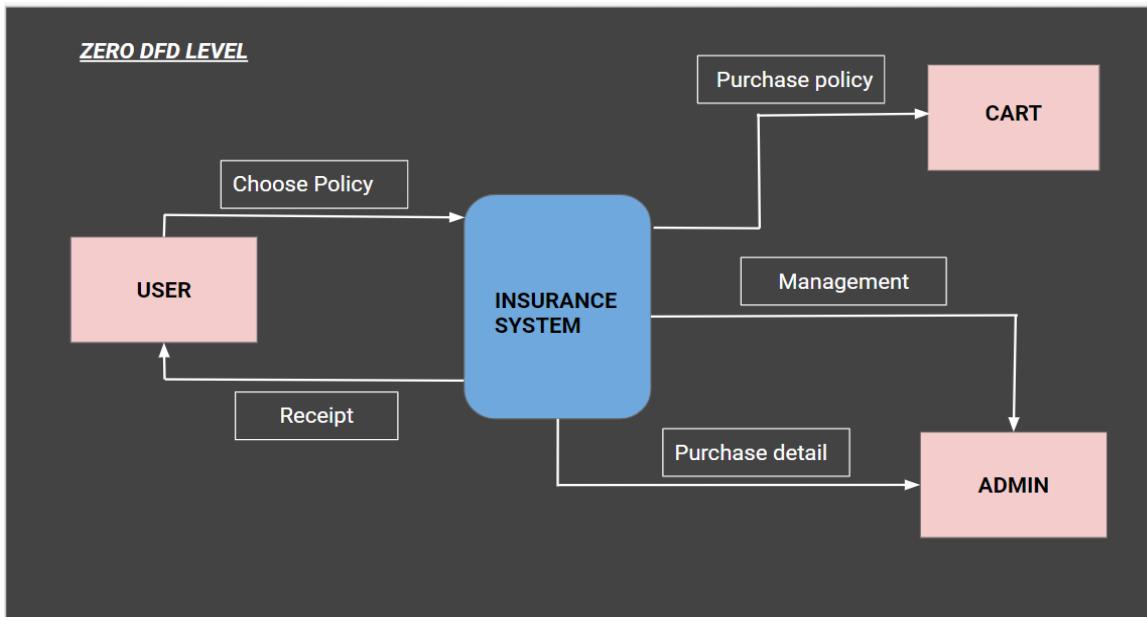
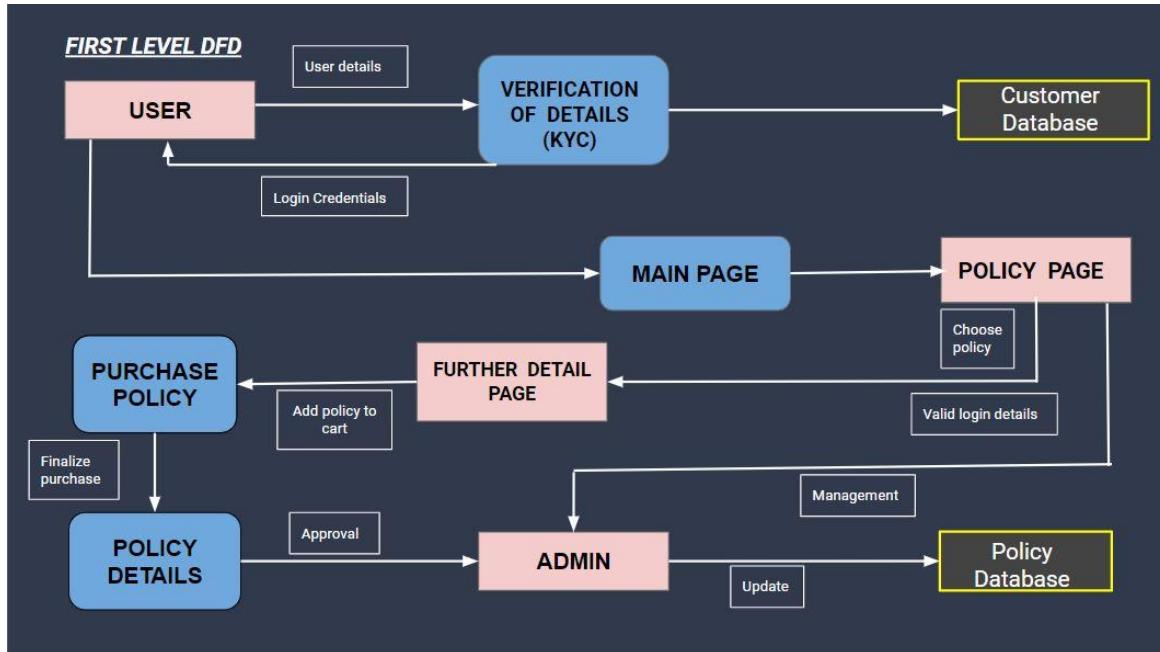


Fig 6.1.3.1 The Zero Level DFD of our website



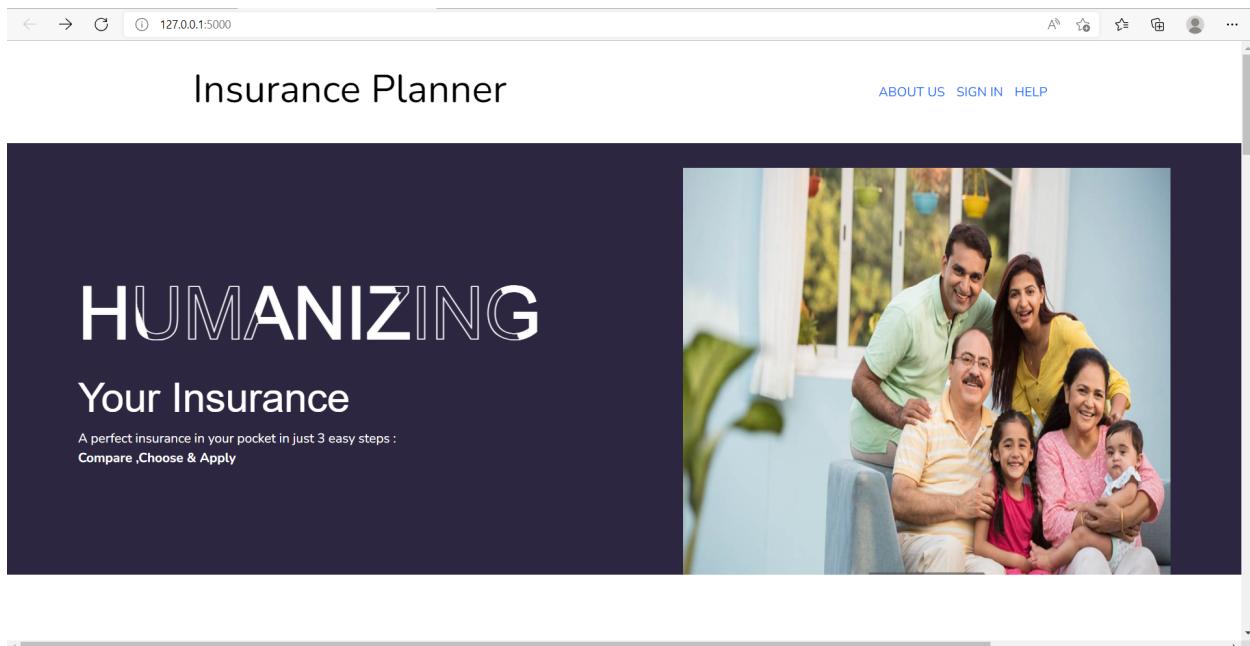
**Fig 6.1.3.2 The First Level DFD our website**

## 6.2 Algorithm/Process (with Expected input and outputs)

### Modules of System

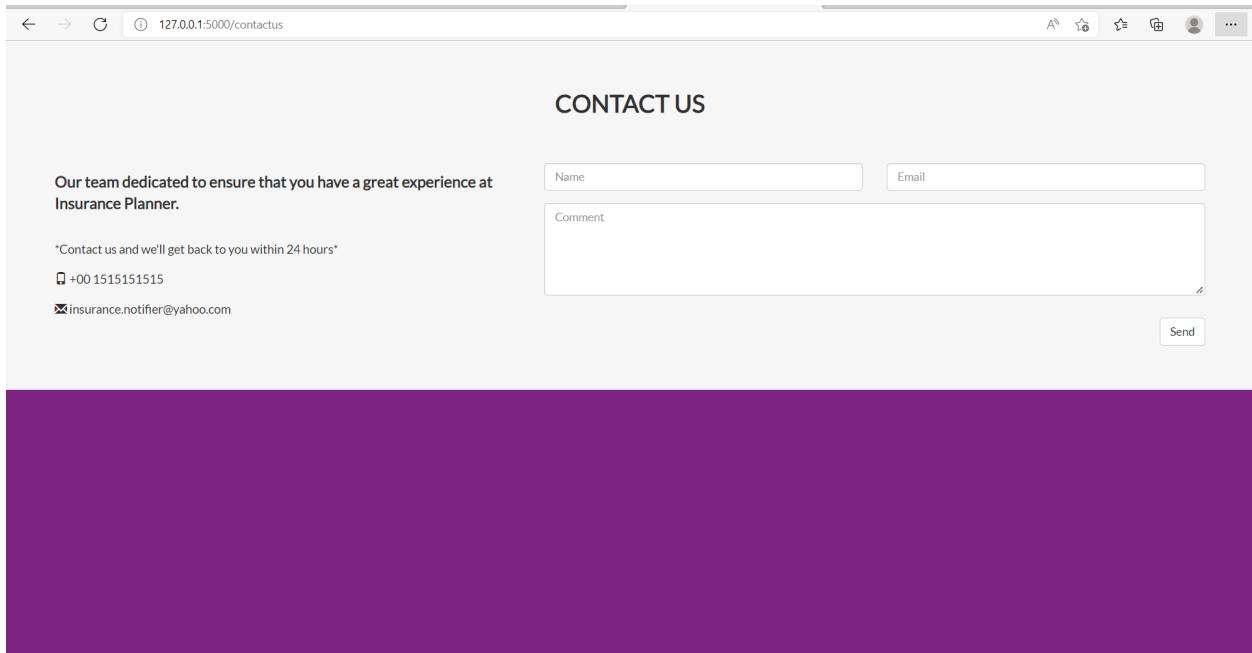
The project is divided into various modules to help the work divide into small yet efficient modules. Each module's output acts as an input to the next module. Each module requires a specific set of knowledge and skill which takes time hence they have been divided accordingly.

### 5.0 Home Page



**Fig 5.0 The Main or Home page of our website,i.e, the page that will be visible to users as soon as they enter our site**

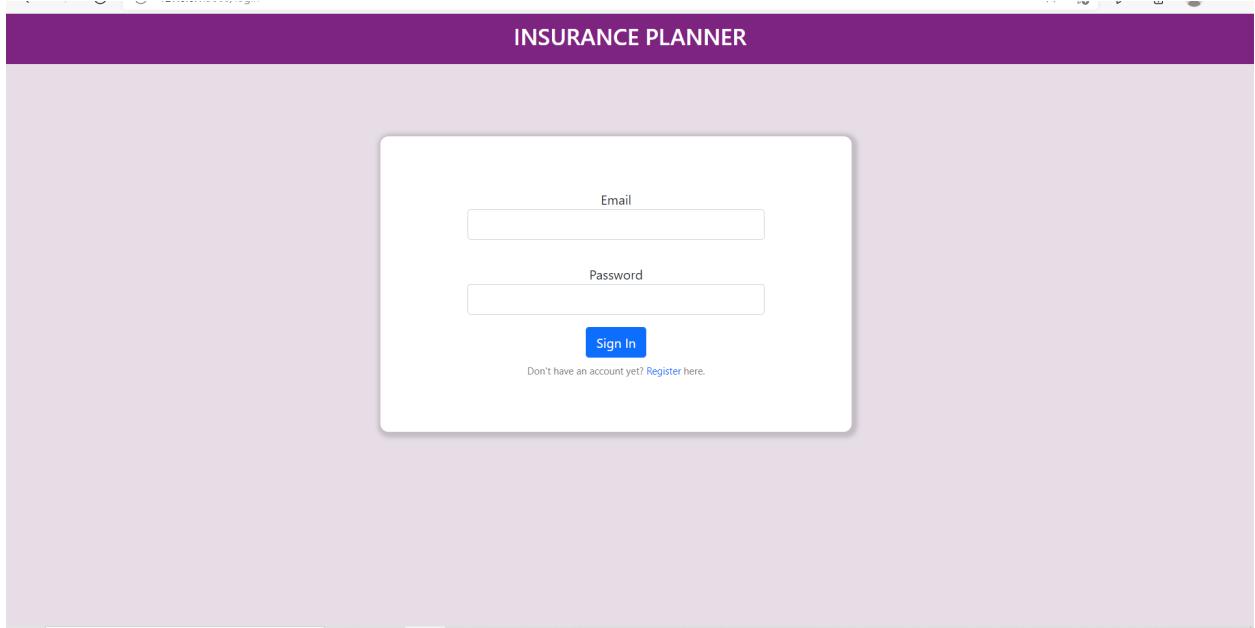
## 5.1 Contact Module



**Fig 5.1 The Contact us page of our website**

In this module, it will contain all the contact information related to our website. So it will include phone number, email id, social networking site details, developers name and details, etc.

## 5.2 Login /Register Module



**Fig 5.2 The Login and Register page of our website**

Through this module, the user will be able to sign in and create an account. When users access the system through Portal Direct Entry, they're considered guests until they log in. The Login Module is a portal module that allows users to type a user name and password to log in. You can add this module on any module tab to allow users to log in to the system.

### 5.3 Kyc Module

INSURANCE PLANNER

KYC for Admin

Email

Moodle Id

INSURANCE PLANNER

KYC for Admin

First Name

Last Name

Mobile No

Upload your photo  
 No file chosen

The figure consists of four screenshots of a web application interface, likely a dashboard or form page. The top two screenshots show a 'KYC for Admin' section with an 'Aadhar No.' input field and a file upload area for an Aadhar PDF. The bottom two screenshots show a photo upload section with a 'Take Photo' button and a 'Submit' button.

KYC for Admin

Aadhar No.

Upload your aadhar pdf

Choose File No file chosen

Submit

← → ⌂ 127.0.0.1:5000/upload

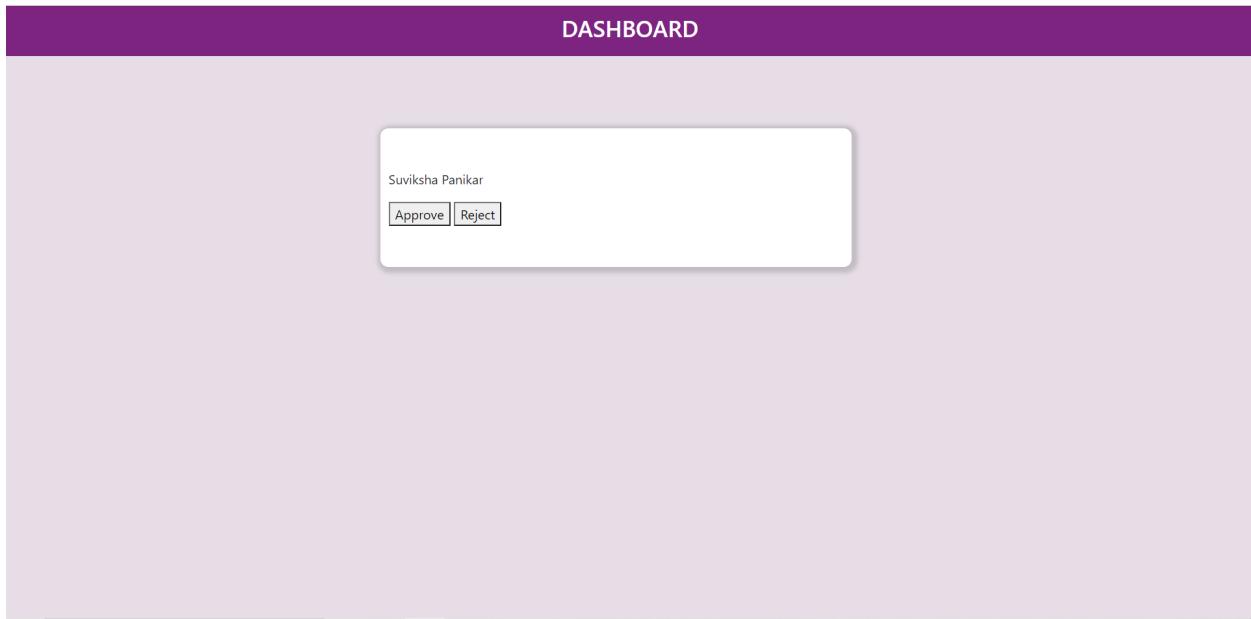
Take Photo

Submit

**Fig 5.3 The Four KYC Pages of our website**

Through this module, the user will be able to do their kyc. It includes four pages: email verification, personal detail check, aadhar detail check and photo upload. Only if the status of kyc is active ( after getting approval from admin) only then the user will be able to continue with policy purchase.

#### **5.4 Admin Module (Dashboard)**



**Fig 5.4 Dashboard Page of our website (only be visible to the admin)**

This module will be used from the admin's side for approving and rejecting the kyc of users. So from this module the admin will be able to keep track of users activity.

## 5.5 Insurance Module

Health Insurance

Life Insurance

Car Insurance

Bike Insurance

### GENERAL INSURANCE

Surprises, both good and bad, are inevitable. Everyone loves the good ones and prays that they are spared the unpleasant ones. Yet, things happen. When the going gets tough, you need general insurance that truly cares.

#### WHAT WOULD YOU LIKE US TO INSURE?

**HEALTH INSURANCE**

Health Insurance is a medical coverage that helps you meet your medical expenses by offering financial assistance. Due to the high cost of hospitalization expenses, it is important to have a health insurance plan in place.

**LIFE INSURANCE**

Life insurance is a contract that offers financial compensation in case of death or disability. Some life insurance policies even offer financial compensation after retirement or a certain period of time.

**CAR INSURANCE**

It's precious—your car. You paid lakhs of rupees to buy that beauty. Even a single scratch can be painful, forget about bigger damages. Car insurance can reduce this pain for a few thousand rupees..

**BIKE INSURANCE**

This is your bike's guardian angel. It's similar to Car insurance. You cannot ride a bike or scooter in India without insurance.

**HOME INSURANCE**

Home insurance is a cover that pays or compensates you for damage to your home due to natural calamities, man-made disasters or other threats. It covers liabilities due to fire, burglary, theft, flood, earthquakes, and sabotage.

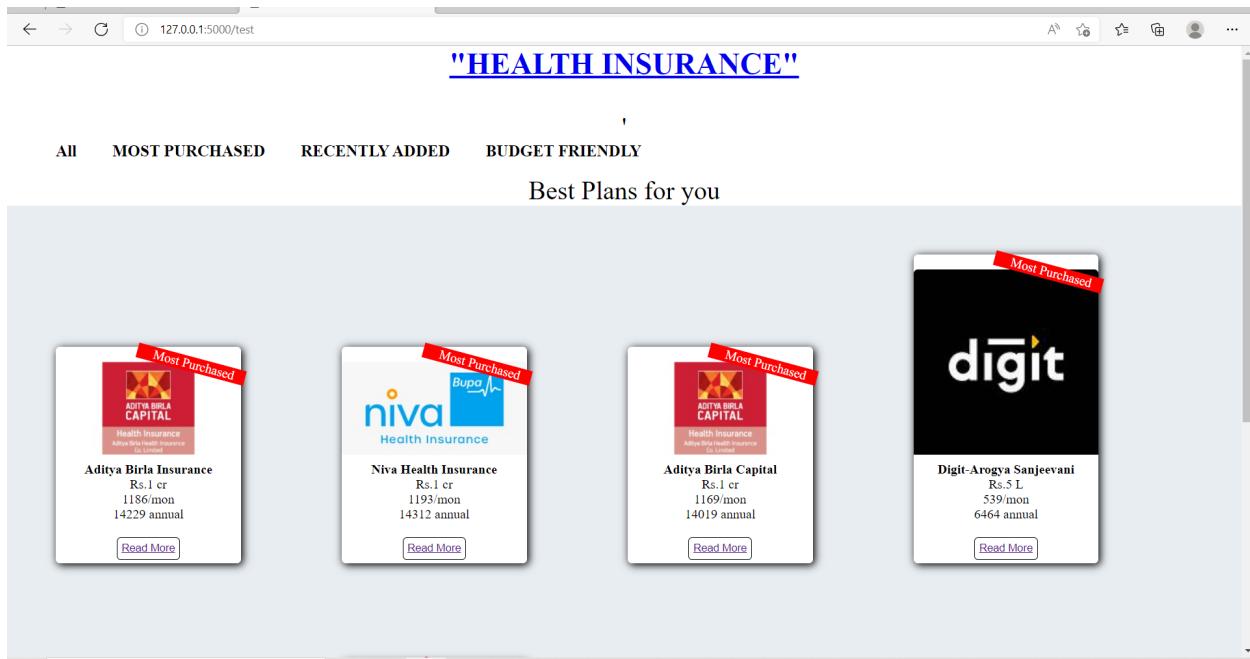
**TRAVEL INSURANCE**

A travel insurance compensates you or pays for any financial liabilities arising out of medical and non-medical emergencies during your travel abroad or within the country.

**Fig 5.5 Insurance Pages of our website ( info about all insurance )**

In this module, all the different types of insurances like life insurance, health insurance, home insurance, car insurance, bike insurance, travel insurance and their information will be displayed. After the selection of the type of insurance, the user will be directed to the policy module.

## 5.6 Policy Module



**Fig 5.6 The Policy Page of our website ( policies of all the insurances )**

This module will comprise of all the policies that are sorted according to the information and requirements given by the user. It will provide the user with details like premium , per year/month and to know more about the policy there is a read more option that will direct the user to the policy-info module.

## 5.7 Policy info Module

The screenshot shows a web browser window with the URL `127.0.0.1:5000/test` in the address bar. The page contains a form for policy information. The form fields include:

- "Whom To Insure" dropdown menu set to "Me".
- "Does any member have illness?"
  - Yes
  - No
- "If yes, Specify here" input field.
- "Covid Positive?"
  - Yes
  - No
- "Any member went through a surgery?"
  - Yes
  - No
- "Enter your Income(Per Annum)" input field.
- "Submit" button.

**Fig 5.7 The Policy Info page of our website**

This module will ask the user basic info like income, who to insure and other related questions to filter out the policies.

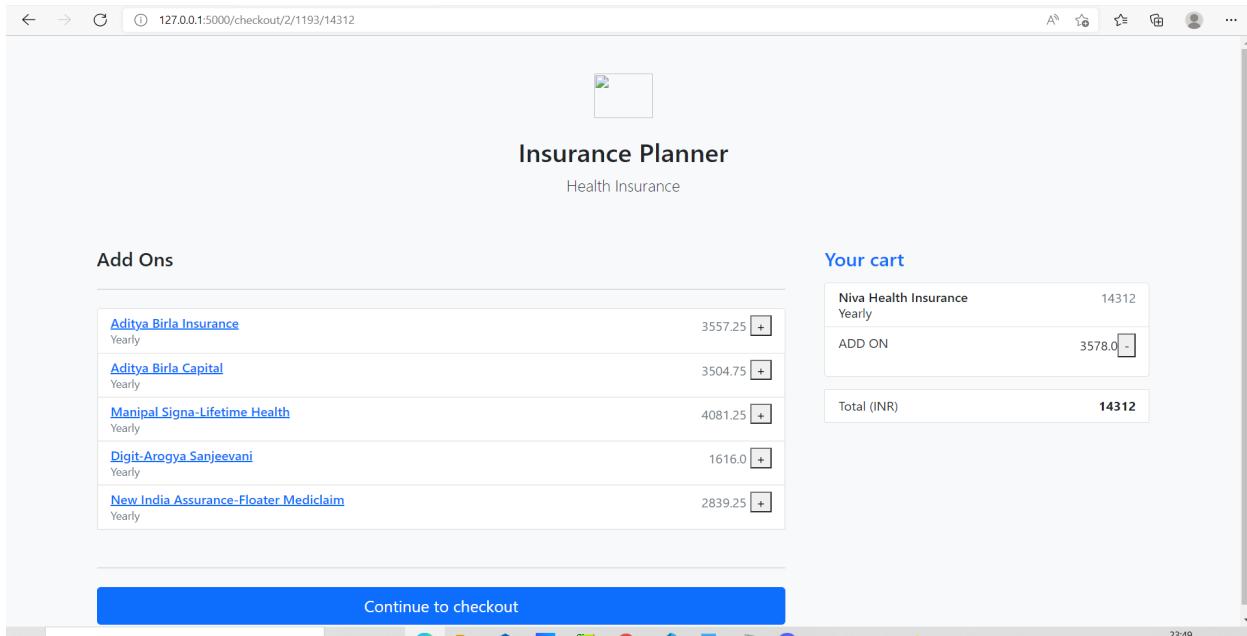
## 5.8 Read More Module

The screenshot shows a web browser window with the URL 127.0.0.1:5000/read-more/2/1193/14312. The page has a header with navigation icons and a search bar. Below the header is a navigation menu with 'Features', 'Claims', 'More Info' (which is underlined), and a blue 'Purchase Policy' button. The main content area has a dark background. At the top left is the Niva Health Insurance logo. To its right, 'Rs.1 Cr Sum Insured' is displayed. Below this, there are two columns: 'Cover' (Rs.1 Cr) and 'Premium' (1193). Further down, the policy number 14312 is shown. On the right side of the page, there is a detailed explanation of the 96% Claim Settlement Ratio, mentioning that it is explained as (Number Of Claims Settled / Number Of Claims) by the insurance company. It also notes that the source is data provided by the insurer and that the feature covers diabetes, with a 4-year waiting period for existing illnesses.

**Fig 5.8 The Read More Page of our website**

This module will display all the further details regarding the policy that has been selected. It will provide the users with details such as cover, premium, and various other features depending upon the type of insurance, for eg, cashless hospitals for health and life insurance, cashless garages for bike and car insurance and also add-on feature. The users will be given an option of comparison module.

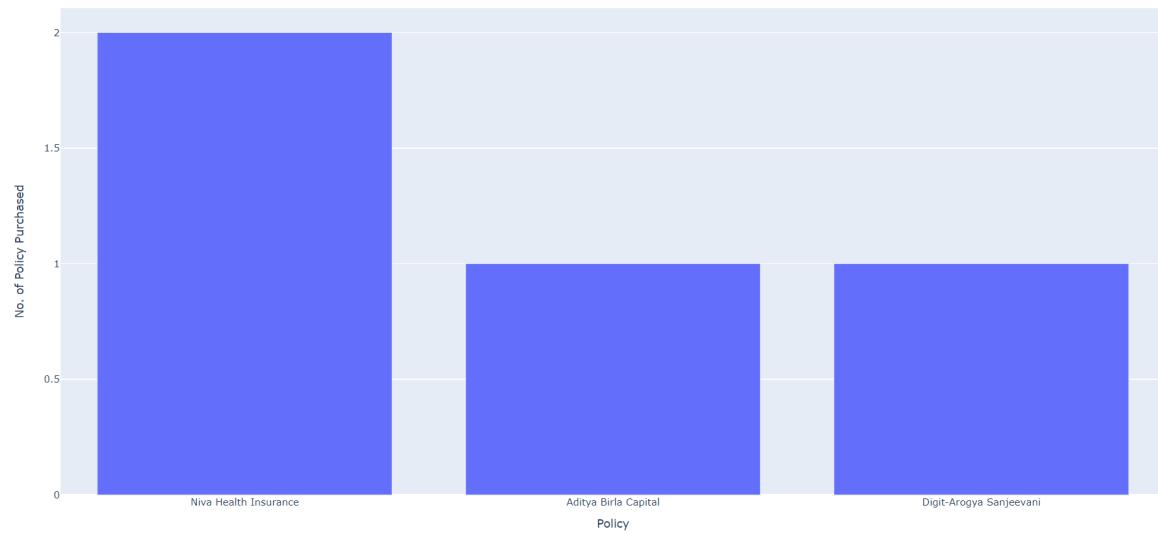
## 5.9 Purchase Module



**Fig 5.9 The Purchase Page of our website**

In this module, the user will be able to view all the final details regarding the policy and also finalize the purchase of the policy, also providing the user with a safe and reliable payment gateway. Once the policy is purchased the user will be provided with a receipt.

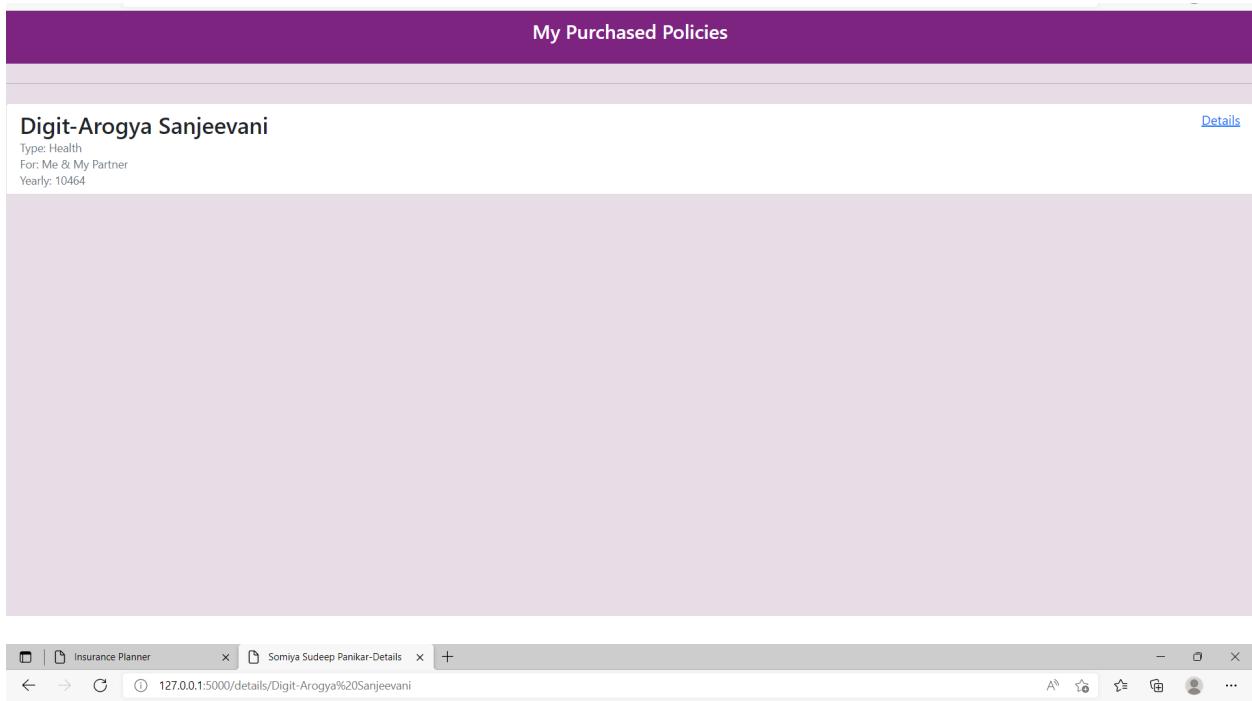
## 6.0 Data analytics Module



**Fig 6.0 The Data Analytics Page of our website**

In this module, the user will be able to see the data analysis graph , i.e., the most purchased policy based on all the data provided by the users.

## 6.1 My Purchase Module



Somiya Sudeep Panikar - Digit-Arogya Sanjeevani Details

Policy for : Me & My PartnerPartner first name: Partner last name: xyz Partner mobile no: 9820101475 Partner aadhar no: 231234354554 Partner aadhar photo:Screenshot (12).png

[Home](#)

**Fig 6.1 The My Purchase Page of our website**

Through this module, the users will be able to view all the policies that they have purchased. And also the details of the dependencies if any.

# Chapter 7

## Implementation

### 7.1 Code

The image shows a code editor with two tabs open, both titled "main.py". The top tab contains code for a form class named "HealthForm" which includes fields for user choice, radio buttons for illness, text input for COVID status, radio buttons for surgery, a float field for income, and a submit button. The bottom tab contains code for an "admin\_only" decorator and a "User" database model. The "User" model has attributes for id, email, password, name, policies (a relationship), kyc (a column), mobileno (a column), aadhar (an integer column), image (a string column), aadharphoto (a string column), and selfie (a string column).

```
1 import smtplib
2 import pandas as pd
3 import sqlite3
4 import plotly.express as px
5 from functools import wraps
6 from flask import abort
7 from flask import Flask, render_template, redirect, url_for, flash, request
8 from werkzeug.security import generate_password_hash, check_password_hash
9 from flask_bootstrap import Bootstrap
10 from flask_sqlalchemy import SQLAlchemy
11 from flask_login import UserMixin, login_user, LoginManager, login_required, current_user, logout_user
12 from sqlalchemy.orm import relationship
13 from flask_wtf import FlaskForm
14 from wtforms import StringField, SubmitField, PasswordField, FileField, IntegerField, SelectField, RadioField, \
15     FloatField
16 from wtforms.validators import DataRequired, Email, Length
17 from werkzeug.utils import secure_filename
18 from email.mime.text import MIMEText
19 from email.mime.multipart import MIMEMultipart
20
21
22 class HealthForm(FlaskForm):
23     choices = ["Me", "Me & My Partner", "Me, My Partner and Child", "My Parents"]
24     user_choice = SelectField("Whom To Insure", choices=choices, validators=[DataRequired()])
25     radio = RadioField("Does any member have illness?", coerce=int, choices=[(1, 'Yes'), (0, 'No')])
26     desc = StringField("If yes, Specify here")
27     covid = RadioField("Covid Positive?", coerce=int, choices=[(1, 'Yes'), (0, 'No')])
28     surgery = RadioField("Any member went through a surgery?", coerce=int, choices=[(1, 'Yes'), (0, 'No')])
29     income = FloatField("Enter your Income(Per Annum)", validators=[DataRequired()])
30     submit = SubmitField("Submit")
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56 def admin_only(f):
57     @wraps(f)
58     def decorated_function(*args, **kwargs):
59         # If id is not 1 then return abort with 403 error
60         if current_user.id != 1:
61             return abort(403)
62         # Otherwise continue with the route function
63         return f(*args, **kwargs)
64
65     return decorated_function
66
67
68 class User(UserMixin, db.Model):
69     __tablename__ = "Users"
70     id = db.Column(db.Integer, primary_key=True)
71     email = db.Column(db.String(100), unique=True)
72     password = db.Column(db.String(100))
73     name = db.Column(db.String(100))
74     policies = relationship("Insurance", back_populates="user_name")
75     kyc = db.Column(db.String(100), default="Pending")
76     mobileno = db.Column(db.String(15))
77     aadhar = db.Column(db.Integer)
78     image = db.Column(db.String(150))
79     aadharphoto = db.Column(db.String(150))
80     selfie = db.Column(db.String(150))
```

```
Travel Policy1.html × main.py × dashboard.html × admin.html × InsurancePlanner.html × my_policy.html × Register.html ×
253     return render_template('InsurancePlanner.html')
254
255     @app.route("/contactus", methods=["GET", "POST"])
256     def contactus():
257         return render_template('contactus.html')
258
259
260     @app.route('/register', methods=["GET", "POST"])
261     def register():
262         form = RegisterForm()
263         # user1 = Insurance.query.filter_by(id=1).first()
264         # db.session.add(policy)
265         # db.session.delete(user1)
266         # db.session.commit()
267         if form.validate_on_submit():
268             if User.query.filter_by(email=form.email.data).first():
269                 flash("You've already signed up with that email, log in instead!")
270                 return redirect(url_for('login'))
271             hash_and_salted_password = generate_password_hash(
272                 form.password.data,
273                 method='pbkdf2:sha256',
274                 salt_length=8
275             )
276             new_user = User(
277                 email=form.email.data,
278                 name=form.name.data,
279                 password=hash_and_salted_password,
280             )
281             db.session.add(new_user)
282             db.session.commit()
283             login_user(new_user)
```

```
Travel Policy1.html × main.py × dashboard.html × admin.html × InsurancePlanner.html × my_policy.html × Register.html ×
308
309     @app.route("/healthpol")
310     def healthpol():
311         return render_template("Health Policy1.html")
312
313
314     @app.route('/login', methods=["GET", "POST"])
315     def login():
316         form = LoginForm()
317         if form.validate_on_submit():
318             email = form.email.data
319             password = form.password.data
320
321             user = User.query.filter_by(email=email).first()
322             # Email doesn't exist or password incorrect.
323             if not user:
324                 flash("That email does not exist, please register.")
325                 return redirect(url_for('login'))
326             elif not check_password_hash(user.password, password):
327                 flash('Password incorrect, please try again.')
328                 return redirect(url_for('login'))
329             else:
330                 login_user(user)
331                 return redirect(url_for('home'))
332
333
334     @app.route('/logout')
335     @login_required
336     def logout():
337         return render_template("admin.html", form=form, current_user=current_user)
```

```
Travel Policy1.html x main.py x dashboard.html x admin.html x InsurancePlanner.html x my_policy.html x Register.html x
367     user.mobileno = mobileno
368     db.session.commit()
369     return redirect(url_for("kyc3"))
370
371
372
373     @app.route("/kyc3", methods=["GET", "POST"])
374     def kyc3():
375         form = KycForm3()
376         if request.method == "POST":
377             aadharno = form.aadhar.no.data
378             f = form.aadarphoto.data
379             file = f'static/userimages/{current_user.name}/{secure_filename(f.filename)}'
380             f.save(file)
381             user = User.query.filter_by(id=current_user.id).first()
382             user.aadharphoto = file
383             user.aadhar.no = aadharno
384             db.session.commit()
385             return redirect(url_for('upload'))
386
387
388
389     @app.route("/read-more/<int:id>/<int:newmonthly>/<int:new_amt>")
390     def read_more(id, newmonthly, new_amt):
391         policy = Policy.query.filter_by(id=id).first()
392         return render_template("adityabirla.html", policy=policy, newmonthly=newmonthly, new_amt=new_amt)
393
394
395     @admin_only
396     @app.route("/dashboard", methods=["GET", "POST"])
397     def dashboard():
```

```
Travel Policy1.html x main.py x dashboard.html x admin.html x InsurancePlanner.html x my_policy.html x Register.html x
544     <br>Regards,<br><a href="/">Insurance Planner</a></p>
545     </div>
546     </body>
547   </html>
548
549   """
550   with smtplib.SMTP("smtp.mail.yahoo.com") as smtp:
551       msg = MIMEText(message, 'html')
552       msg['Subject'] = 'KYC Validation Email'
553       msg['From'] = my_email
554       msg['To'] = user.email
555       smtp.starttls()
556       smtp.login(user=my_email, password=password)
557       smtp.sendmail(
558           from_addr=my_email,
559           to_addrs=user.email,
560           msg=msg.as_string()
561       )
562       flash("email sent", "success")
563   return redirect(url_for("dashboard"))
564
565
566
567     @app.route('/upload', methods=['GET', 'POST'])
568     def upload():
569         file = ""
570         form = SubmitForm()
571         if request.method == 'POST':
572             fs = request.files.get('snap')
573             if fs:
574                 file = f'static/userimages/{current_user.name}/{secure_filename("selfie.jpg")}'
```

```

745     msg['From'] = my_email
746     msg['To'] = user.email
747     smtp.starttls()
748     smtp.login(user=my_email, password=password)
749     smtp.sendmail(
750         from_addr=my_email,
751         to_addrs=user.email,
752         msg=msg.as_string()
753     )
754     flash("email sent", "success")
755     return redirect(url_for("dashboard"))
756 return render_template("reject.html", form=form)

758
759 @app.route("/analysis")
760 def analysis():
761     policy = []
762     cnx = sqlite3.connect('usersfinalfinal.db')
763     df = pd.read_sql_query("SELECT * FROM insurance", cnx)
764     count = df['policy_id'].value_counts()
765     for idx in df['policy_id'].unique():
766         policy.append(str(Policy.query.filter_by(id=int(idx)).first().name))
767     if policy:
768         fig = px.bar(df, x=policy, y=count, labels={'x': 'Policy', 'y': 'No. of Policy Purchased'})
769         fig.write_html("templates/analysis.html")
770         return render_template("analysis.html", fig=fig)
771     else:
772         return "No Data"
773
774
775 @app.route("/test/<type>", methods=["GET", "POST"])

```

```

904     # f"{adnop2}\n"
905     # f"{adtp2}\n"
906     else:
907         return redirect(url_for("policies", id=current_user.id))
908     return render_template("checkout.html", form=form, current_policy=current_policy, policies=policies, newmonthly=newmonthly, newyearly=newyearly)

909
910
911 @app.route("/mypolicies/<int:id>")
912 def policies(id):
913     insurance = Insurance.query.filter_by(user_id=id).all()
914     return render_template("my_policy.html", insurances=insurance)

915
916
917 @app.route("/details/<current_policy>")
918 def details(current_policy):
919     # with open(f"static/userdata/{current_user.name}-{current_policy}-details.txt") as f:
920     #     lines = f.readlines()
921     policy = Policy.query.filter_by(name=current_policy).first()
922     dependency = Dependencies.query.filter_by(user=current_user.id, policy=policy.id).first()
923     return render_template("details.html", current_user=current_user, lines=dependency, current_policy=current_policy)

924
925
926 if __name__ == "__main__":
927     app.run(debug=True)
928

```

# **Chapter 8**

## **Result**

Mortgage Redemption- The system acts as an effective tool to cover mortgages and loans taken by the policy holders so that in case of any unforeseen event, the burden of repayment does not fall on the bereaved family.

Risk cover -life is full of uncertainties; in this scenario the use of the system ensures that one enjoys a good quality of life against any unforeseen event.

Protection against rising health expenses-with the system available, customers can choose from a wide range of plans that would offer the benefits of protection against critical diseases and hospitalization expenses.

Asset Protection: Certain types of insurance policies are designed to protect assets such as homes, cars, boats and other valuable tangible items.

Income Protection-There are insurance policies available that will protect the ability to earn a living. If, for some reason, a worker is no longer able to continue with gainful employment, these policies can be structured to replace a majority of lost income.

# **Chapter 9**

## **Conclusion**

Effective insurance management lies in balancing the costs and benefits of insurance, through optimizing the coverage at economical cost and further through conscious and careful retention of risks within the world. This is precisely where the Online Insurance supports and improves many of the core functionality of the insurance organization i.e. insurance project helps in quick easy monitoring of the reports that have been automatically generated as and when the admin and policy agent performs transactions in the system. Using such a system helps the organization in minimizing the time consumed in fulfilling the day-to-day functionality and cutting down the expenses incurred on the same. It must also be remembered that insurance is a method of transferring the financial impact of risk and the risk itself. Hence the basic responsibility of the system is to act as a guild in risk management.

# **References**

- [1]Yachna Gupta, Anmol Sarwana, Meenal Gogia, M.K. Nallakaruppan “ A Software for Insurance Consultancy “ (2013).
- [2]Jihong Wu Dalian Neusoft University of Information, Dalian, China, International Conference on Virtual Reality and Intelligent Systems (ICVRIS)(2019)
- [3]The Analysis of Online Insurance Market Potential Based on the Distribution Fitting-Minghong Zhang, Zhiyuan Zhang, Xinman Zhang Department of Public Economics The School of Economics, Xiamen University

# Annexure 1- Project Planning (Using Gantt chart)

## Acknowledgement

We have great pleasure in presenting the report on Insurance Planner. We take this opportunity to express our sincere thanks towards our guide Prof. Merlin Jacob Priya, Department of Computer Engineering, APSIT, Thane and Prof. Sachin Malave, the HOD of Computer Engineering department for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards her constant encouragement, support and guidance through the development of the project. We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Somiya Panikar  
20102095

Namrata Narkhede  
20102106

Omkar Nimbalkar  
20102131  
Hrishikesh Rane  
20102157