CUSTOMER CARE REGISTRY
NALAIYA THIRAN PROJECT REPORT 2022

Submitted by

Team ID: PNT2022TMID16183

Gowtham B	**927619BIT4029**
Bharathi Kannan M	**927619BIT4008**
Elankumaran M	**927619BIT4021**
Gowri Shankar M	**927619BIT4028**

Content

TEAMID:PNT2022TMID16183

CUSTOMERCARE REGISTRY

ABSTRACT

ThemainideaistoimplementanonlinesystemformanagingtheinternetcustomersandcomplaintsystemforcustomersforraisingcomplaintsontheissuesrelatedtoISPproviderandprovidebestcustomercareserviceforusersusingthisapplication.ThereAremanyInternetsecurityprovidersinacountrythatwillprovideinternetservicesforusersondifferentpackages.BasicallyISPworksonthreeconnections,DialUpusingtelephoneservice,Broadbandandwireless
connections.Adminmanagesthewholesystembyperformingtasksuchasadding/viewing/editing/deletingemployeedetails.Systemallowsadmintoaddcustomerdetailsandpredicttheinternetplansbasedontheirbusinesstype,regionorage.Internetplanexpirydatesofeachcustomerwillbedisplayedtoadminiftheplanisexpiringin5days.Allthecomplaintdetailswillbedisplayedtotheadminaswellasemployeetoenterthecomplaintresolution.Employeecanloginusingvalididandpasswordwhichisprovidedbytheadmin.
Afterlogin,employeecanviewalltheunsolvedcomplaintsreceivedfromthecustomers.Afterresolvingthecomplaint,employeecanupdatetheprovidedresolutionandclosethecomplaint.Oncethecomplaintisclosed,anemailwillbesenttocustomerwillresolutiondetails.Afterreceivingthemail,customercanviewtheresolutionandcanpostthefeedbacktothisapplication.Admincanviewthereportofcomplaintresolutionprovidedbytheemployee.

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Internetisgainingmoreimportancedayafterdayinalllifeaspects,especiallyinbusinessandmarketingduetotheamazingincreaseininternetusersaroundtheworldwithanestimateof2.4billionusersin2012,whencomparingthisnumbertothenumberofinternetusersin2000,agrowthof566%canbenoticed.Thisis makinginternetthefastestmediaofalltimeinbothgrowthrateandnumberofusers(internetworldstats,2012).Basedonacomprehensivestudyin2011, the numberof advertisements circulatedover thenet was morethan 3.5 milliondaily.

Internetbecameoneofthemostefficientwaystoconductbusiness.Indevelopedandwell-developedcountries,internetprovedtobeofmuchhelpforlocalenterpriseswhereitprovidesgreatpotentialforsuchenterprisestocompeteworldwide.Themainideaistoimplementanonlinesystemformanagingtheinternetcustomersandcomplaintsystemforcustomersforraisingcomplaintsontheissuesrelatedto ISPproviderandprovidebestcustomercareserviceforusersusingthisapplication.TherearemanyInternetsecurityprovidersinacountrythatwillprovide internetservicesforusersondifferentpackages.BasicallyISPworksonthreeconnections,DialUpusingtelephoneservice,Broadbandandwirelessconnections.

1.2 PURPOSE

Themainideaistoimplementanonlinesystemformanagingtheinternetcustomersandcomplaintsystem forcustomersforraisingcomplaintsontheissuesrelatedtoISPproviderandprovidebestcustomercareserviceforusersusingthisapplication.Byaddingmoreentriestothedatabasestore,theapplicationcanrespondto morenumberofqueriesfromthecustomers.Theimportance isgiven ongiving correctreply to theinput queries.

2. LITERATUREREVIEW

2.1 EXISTINGPROBLEM

TITLE:CustomercomplaintsManagementA

UTHOR:MerlinStone

DESCRIPTION:

Consultantsworkingoncommercialprojectsoftenfailtotakeaccountofthedeepand broad academic literature on the topic on which they are working. Because ofhis position as a hybrid academic and consultant, the author is obliged to keepclosely in touch with the different literatures for the areas in which he teaches -broadly marketing, customer relationship management, customer service andbranding.

PUBLISHEDIN: Jun2011

TITLE:SatisfactionFactorsofCustomersA

UTHOR: SruthiSivaprakasham, J

Jayashree**DESCRIPTION:**

Customersatisfactionisanimportantfactorinthephenomenonoftheconstructionproces s and customer relationship. As Construction Companys increases itsantagonism, greater attention continues to be placed on customer relationships andsatisfied customers. Customer satisfaction accredits construction companies todifferentiatethemselvesfromtheircompetitorsandcreatesustainableadvantage. Thosewhobuythegoodsorservicesprovidedbycompaniesarecustomers.
PUBLISHEDIN: Nov2017.

TITLE:CustomerComplaintManagementA
UTHOR:stoneM

DESCRIPTION:

Consultantsworkingoncommercialprojectsoftenfailtotakeaccountofthedeepand broad academic literature on the topic on which they are working. Because ofhis position as a hybrid academic and consultant, the author is obliged to keepclosely in touch with the different literatures for the areas in which he teaches –broadly marketing, customer relationship management, customer service andbranding.

PUBLISHEDIN: June2011

TITLE:TheoryandpracticeofCustomerRelatedimprovementsAU

THOR: DanielGyllenhammar

DESCRIPTION:

Customers are vital to any organization and system, and must therefore beconsidered when seeking to improve. However, how to improve with regard to thecustomer, is not clear, and the knowledge is spread over several research fields,makingitdifficultforresearchersandpractitionerstocomprehend.Thepurposeof this literature review is to show how customer-related improvements are describedintheliteratureandhowtheresearchisperformed.

PUBLISHEDIN: Feb2022

TITLE:**CustomerSatisfactioninnewhampshirestategovermentAU**

THOR: RusellLinden

DESCRIPTION:

A successful performance management program supports and promotes theaccomplishment of an agency’s mission and goals. It does this by aligning teamand individual performance elements and standards with the organizational goals.This will aim everyone’s energies in the same direction: to provide “best-in-business”customerservice.Aquestionstateagenciesmaybeaskingthemselvesis“Why doIneedcustomerservicestandards?”“Weknowwhatcustomerserviceisandwhatwe’resupposedtodo.”

PUBLISHEDIN: Feb 2009

2.2 REFERENCES

1. Customer complaints
   ManagemenAUTHOR:
   MerlinStone

2. Satisfaction
FactorsofCustomersAUTHOR:
SruthiSivaprakasham, JJayashree

3. Customer Complaint
ManagementAUTHOR : Stone M

4. Theory
andpracticeofCustomerRelatedimprovementsAUTHOR:
Daniel Gyllenhammar

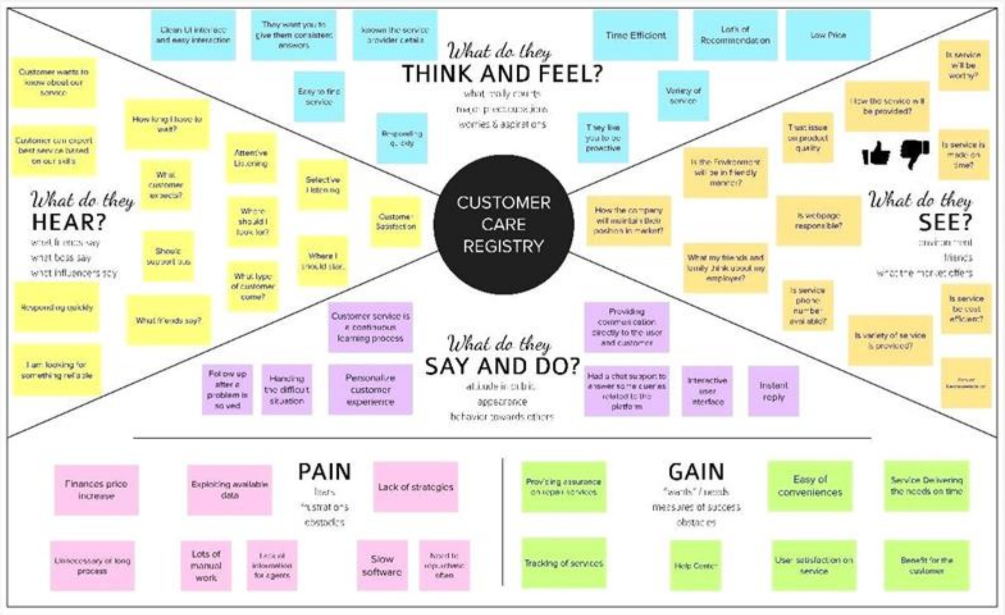
5. CustomerSatisfactioninnewhampshirestategovermentAU
   THOR : Rusell Linden

2.3 PROBLEMSTATEMENTDEFINITION

ProblemStatement(PS)	**I am(Customer)**	**I'mtryingto**	**But**	**Because**	**Which makes mefeel**
PS-1	User	TicketBooking	TimeDelay	Agent NotResponding	Sad
PS-2	User(Agent)	SolveProblem	Customer Not Responding	CustomerUnavailable	Frustrated
PS-3	User(Admin)	BackupData	DataLoss	System Failure	Anxiety

3. IDEATION& PROPOSEDSOLUTION

3.1 EMPATHYMAPCANVAS



CUSTOMER CARE REGISTRY

What do they THINK AND FEEL?

what really counts

major preoccupations

worries & aspirations

Clean UI interface and easy interaction

They want you to give them consistent answers

known the service provider details

Time Efficient

Lack of Recommendation

Low Price

Easy to find service

Responding quickly

Variety of service

They like you to be proactive

What do they HEAR?

what friends say

what boss say

what influencers say

Customer wants to know about our service

Customer can expect best service based on our skills

How long I have to wait?

What customer expects?

Attentive Listening

Selective Listening

Where should I look for?

Customer Satisfaction

Should support bus

Where I should start

What type of customer come?

Responding quickly

What friends say?

I am looking for something reliable

What do they SEE?

environment

friends

what the market offers

Is service will be worthy?

How the service will be provided?

Trust issue on product quality

Is service is made on time?

Is the Environment will be in friendly manner?

How the company will maintain their position in market?

Is webpage responsible?

What my friends and family think about my employer?

Is service phone number available?

Is service be cost efficient?

Is variety of service is provided?

What do they SAY AND DO?

attitude in public

appearance

behavior towards others

Customer service is a continuous learning process

Follow up after a problem is solved

Handling the difficult situation

Personalize customer experience

Providing communication directly to the user and customer

Had a chat support to answer some queries related to the platform

Interactive user interface

Instant reply

PAIN

fears

frustrations

obstacles

Finances price increase

Exploiting available data

Lack of strategies

Unnecessary or long process

Lots of manual work

Lack of information for agents

Slow software

Need to repurchase often

GAIN

"wants" / needs

measures of success

obstacles

Providing assurance on repair services

Easy of conveniences

Service Delivering the needs on time

Tracking of services

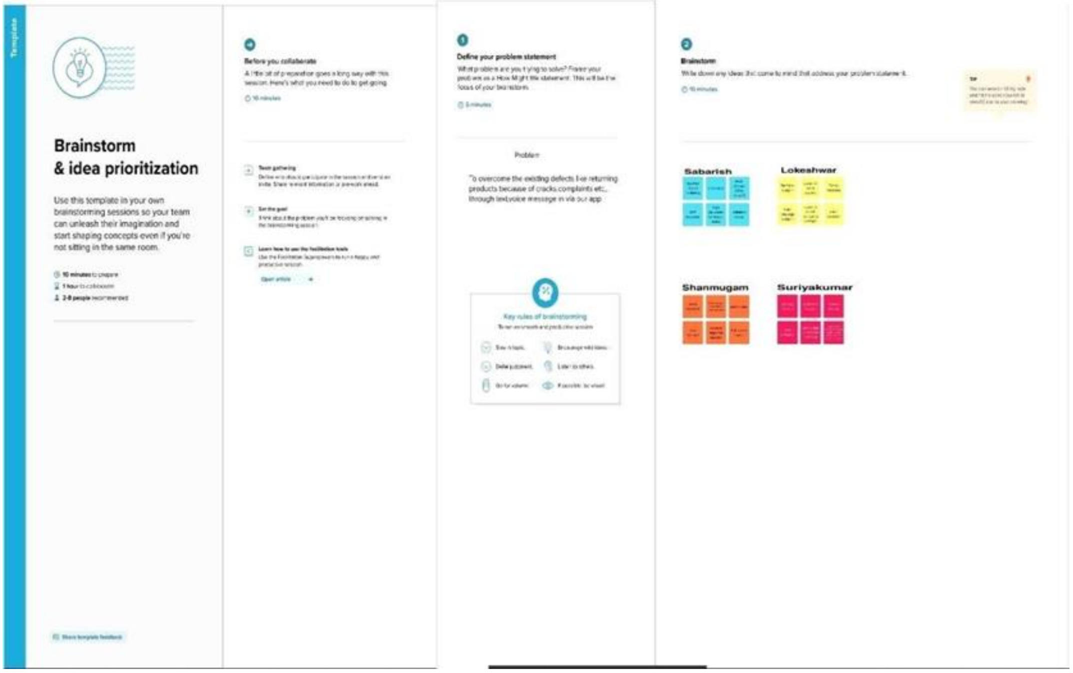
Help Center

User satisfaction on service

Benefit for the customer

3.2 IDEATION&BRAINSTORMING

Step-1:TeamGathering,Collaboration andSelecttheProblemStatement



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

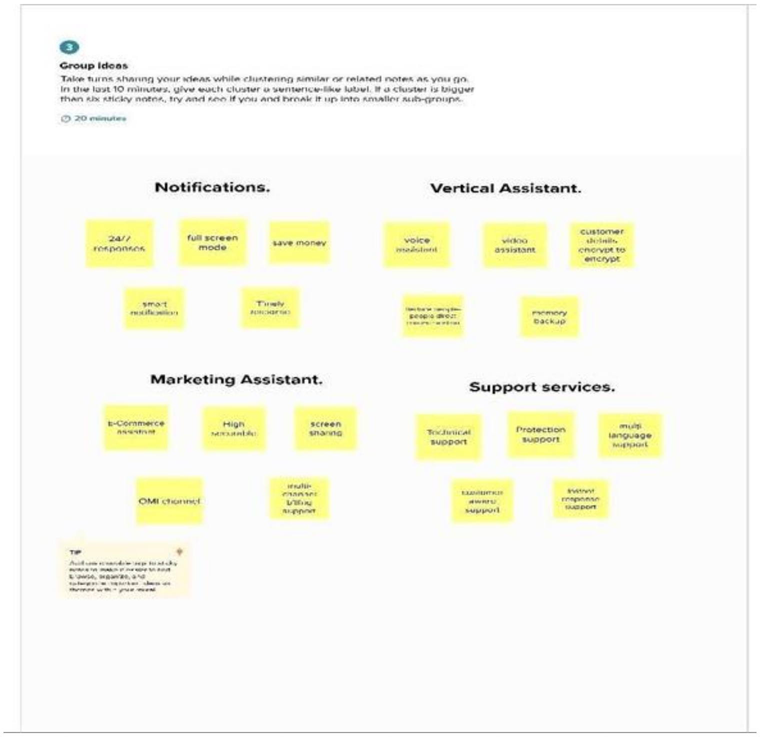
Define your problem statement

Problem

"To overcome the existing defects like returning products because of cracks,complaints etc., through text/voice message in via our app

Brainstorm

Step-2:Brainstorm,IdeaListingandGrouping



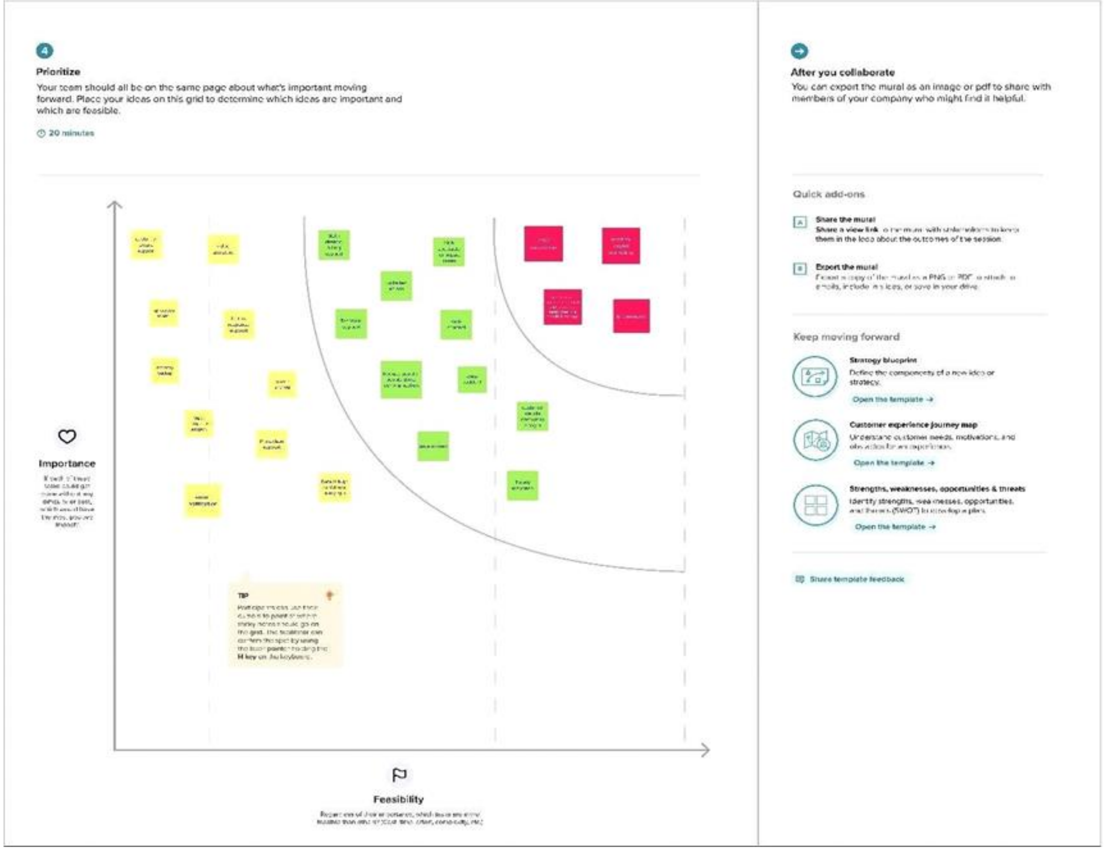
3

Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Step-3:IdeaPrioritization



Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

TIP

Participants can use their cursors to point at where their sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 PROPOSEDSOLUTION

This proposed system provides an online way of solving the problems faced by the public bysaving time and eradicate corruption, and The ability of providing many of the reports on thesystem, and add to Facilitate the process of submitting a complaint. In this project we can designweb application to analyze the complaints and to provide automatic forwarding system of user'scomplaints. User is easily known about status of complaints. If the action can't be taken properlymeans, send to higher authorities. The proposed system is supposed to handle as more number ofcustomersaspossibleinanyparticulartime.Themailserviceisalsoprovidedtohaveacommunicationbetweentheadminandtheusers.Theuserqueriesshouldbeperiodicallyreferred and the solution should be provided quickly.

3.4 PROBLEMSOLUTIONFIT

The existing system is handled manually. The system has a formatted call centre management forcustomersin paperwork like files and document format. The customers are waiting a call totakenbythecallcentreemployeepicktheircalls.So any urgent work we didn't get anyimportant response from the call centre.So the proposed system the manager will look after itandthenhewilltakecareaboutthecustomer'sproblems.Afterthatthemanagerwillenquireand allocate the problem to the specified person in that department. The person will enquire theproblem and then rectifies it.

4. REQUIREMENTANALYSIS

4.1 FUNCTIONAL REQUIREMENTAdmin

- Login
- Add Employees
- Add Customers
- View Details
  - ➢ Employees
  - ➢ Customers
  - ➢ Feed Back

Employee

- Login
- ViewComplaints
- SendNotification(ThroughEmail)
- ViewFeedBack

Customer

- Login
- PostComplaint
- ViewNotification
- FeedBack

MODULEDESCRIPTION

Admin

➢ Login

Inthismodule,theadmincanlogininthesystemusinghis/herusernameandpassword.

➢ AddEmployees

Inthismodule,theadmincanaddtheemployeeinformationlikeemployeename,id,phone number, mail id, location etc.

- AddCustomers

Inthismodule,theadmincanaddthecustomerinformationlikecustomername,id,phonenumber, mail id, location etc.

- ViewDetails

Inthismodule,theadmincanviewtheemployeedetails,customerdetailsandfeedbackdetails.

Employee

- Login

Inthismodule,theemployeecanlogininthesystemusinghis/herusernameandpassword.

- ViewComplaints

Inthismodule,theemployeecanviewthecustomercomplaintusingthisapplication.

- SendNotification(ThroughEmail)

Theemployeecansentthenotificationtotheuserthroughtheemailforupdatestatusofthecomplaint using this system.

- ViewFeedBack

Inthismodule,theemployeecanviewtheuserfeedback.

Customer

- Login

Inthismodule,thecustomercanlogininthesystemusinghis/herusernameandpassword.

- PostComplaint

Inthismodule,thecustomercanposttheinternetservicerelatedcomplainttothissystem.

- ViewNotification

Iftheemployeecanupdatethestatusofthecomplaint,theusercangettheautomaticnotification.

- FeedBack

Inthismodule,theusercanpostthefeedbackofproductsorservice.

4.2 NON

FUNCTIONALREQUIREMENTSNon -

Functional Requirements

Usability

Thesystemshallallowtheuserstoaccessthesystemwithpcusingwebapplication.Thesystemusesawebapplicationasaninterface.Thesystemisuserfriendlywhichmakesthesystemeasy

Availability

Thesystemisavailable100%fortheuserandisused24hrsadayand365daysayear.Thesystemshallbe operational 24 hoursa day and 7days a week.

Scalability

Scalabilityisthemeasureofasystem'sabilitytoincreaseordecreaseinperformanceandcostinresponseto changes inapplication andsystem processing demands.

Security

Asecurityrequirementisastatementofneededsecurityfunctionalitythatensuresoneofmanydifferentsecurity properties of softwareis being satisfied.

Performance

Theinformationisrefresheddependinguponwhethersomeupdateshaveoccurredornotintheapplication. The system shall respond to the member in not less than two seconds from the time of therequest submittal. The system shall be allowed to take more time when doing large processing jobs.Responsestoviewinformationshalltakenolongerthan5secondstoappearonthescreen.

Reliability

The system has to be 100% reliable due to the importance of data and the damages that can be caused byincorrect orincomplete data.Thesystem willrun 7days aweek.24 hoursa day.

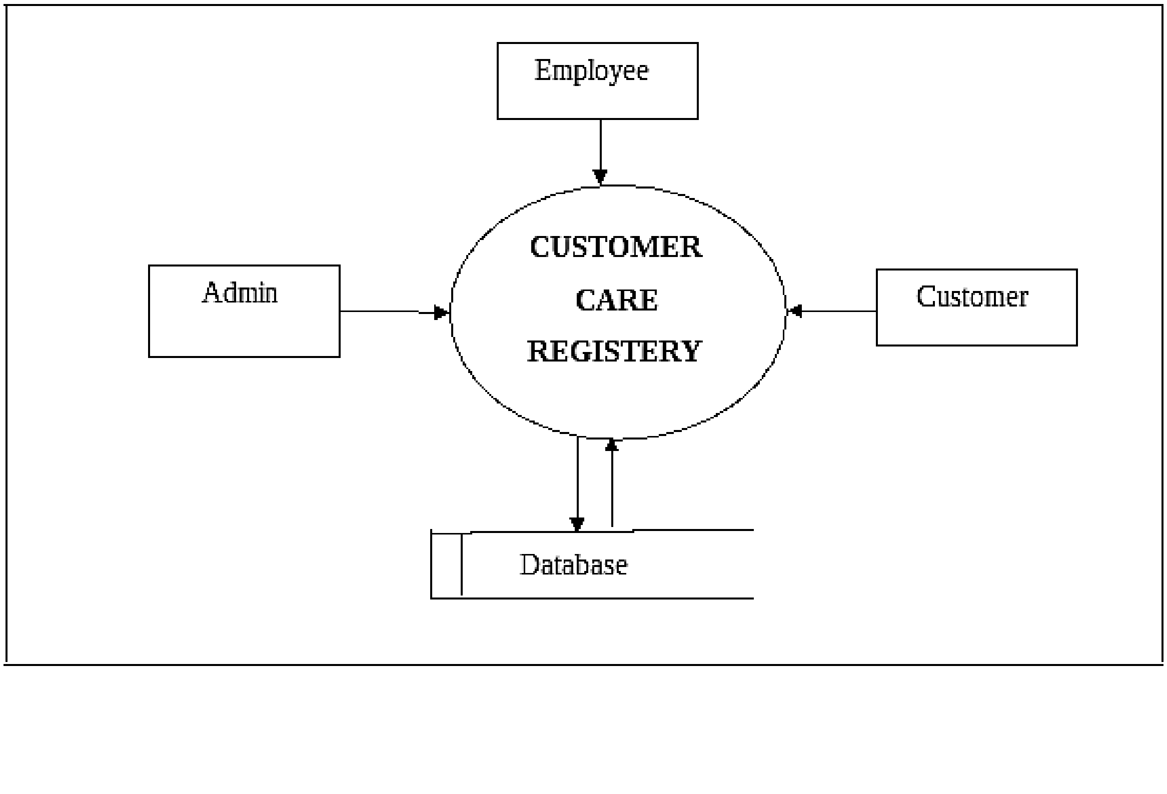
5. PROJECTDESIGN

5.1 DATAFLOWDIAGRAMS

A data-flow diagram is a visual representation of how data moves through a system or a process(usuallyaninformationsystem).TheDFDadditionallygivesdetailsabouteachentity'sinputsand outputs as well as the process itself. A data-flow diagram lacks control flow, loops, anddecision-making processes. Using a flowchart, certain operations depending on the data may bedepicted.

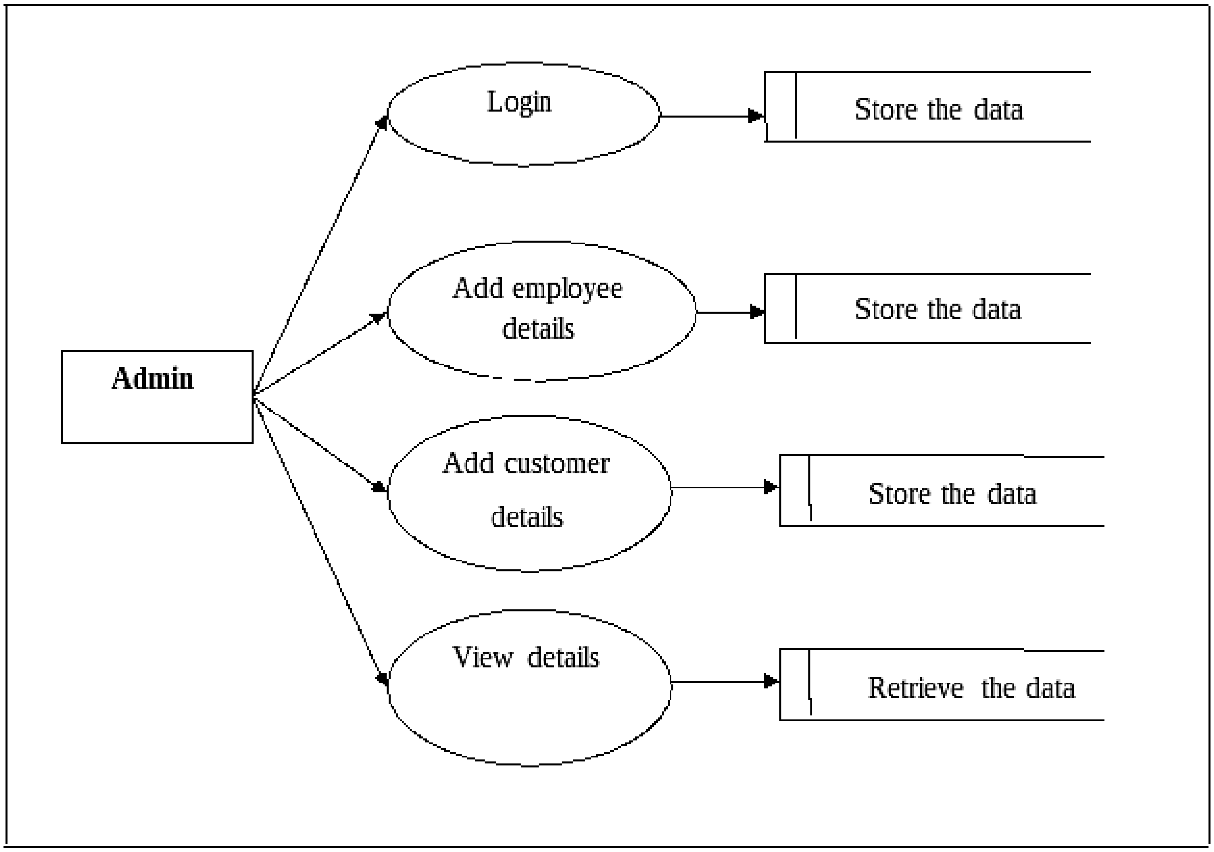
LEVEL0

Itisalsoknownasacontextdiagram.It'sdesignedtobeanabstractionview,showingthesystem as a single process with its relationship to external entities. It represents the entire systemas asinglebubblewithinputandoutputdataindicatedbyincoming/outgoingarrows.



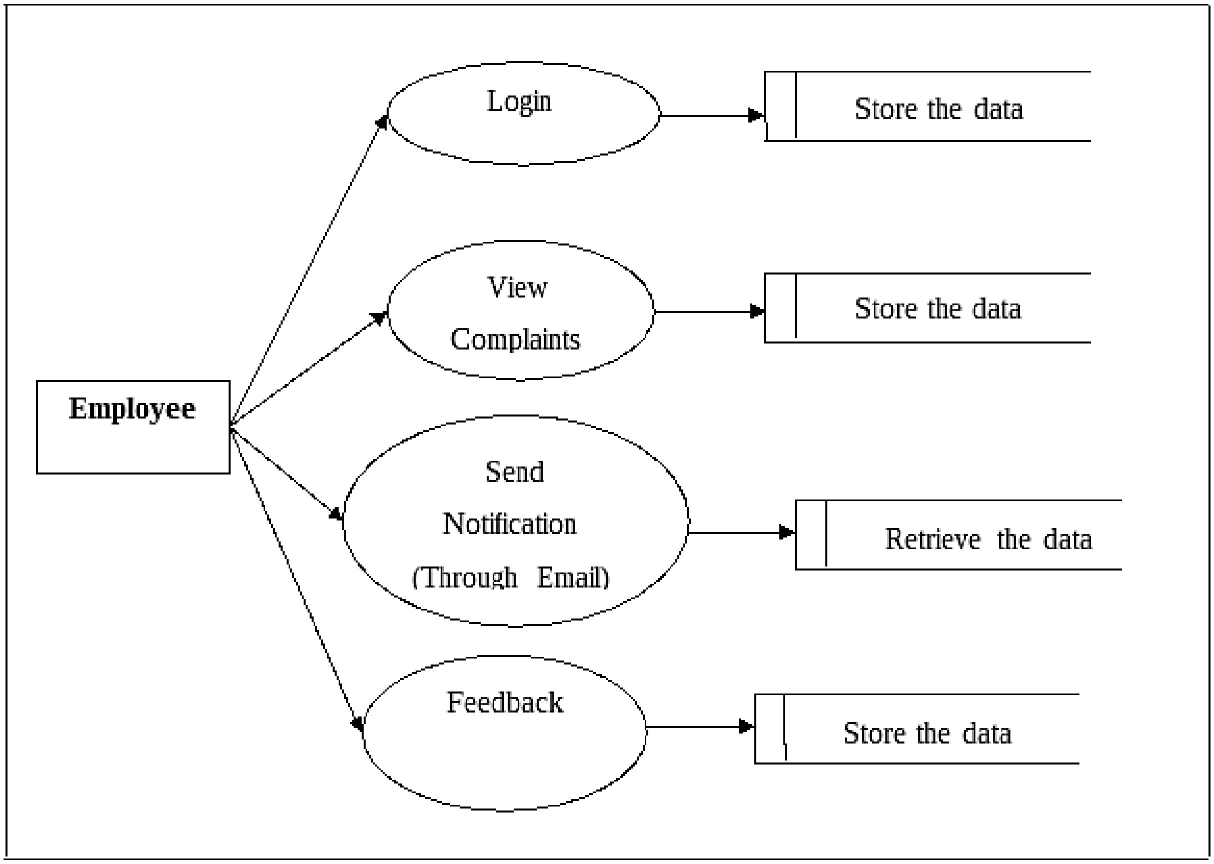
LEVEL1

In1-levelDFD,the context diagram is decomposed into multiple bubbles/processes. In thislevel, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub processes.



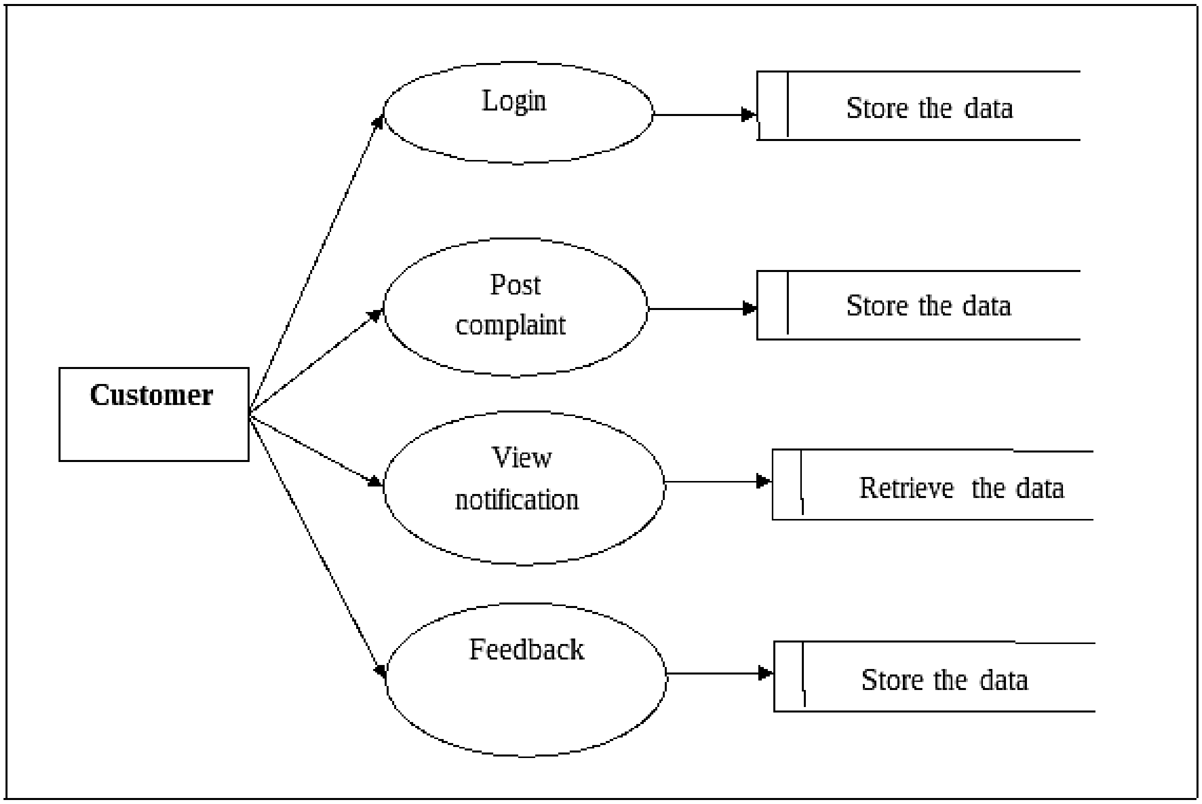
LEVEL2

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record thespecific/necessary detail about the system’s functioning.

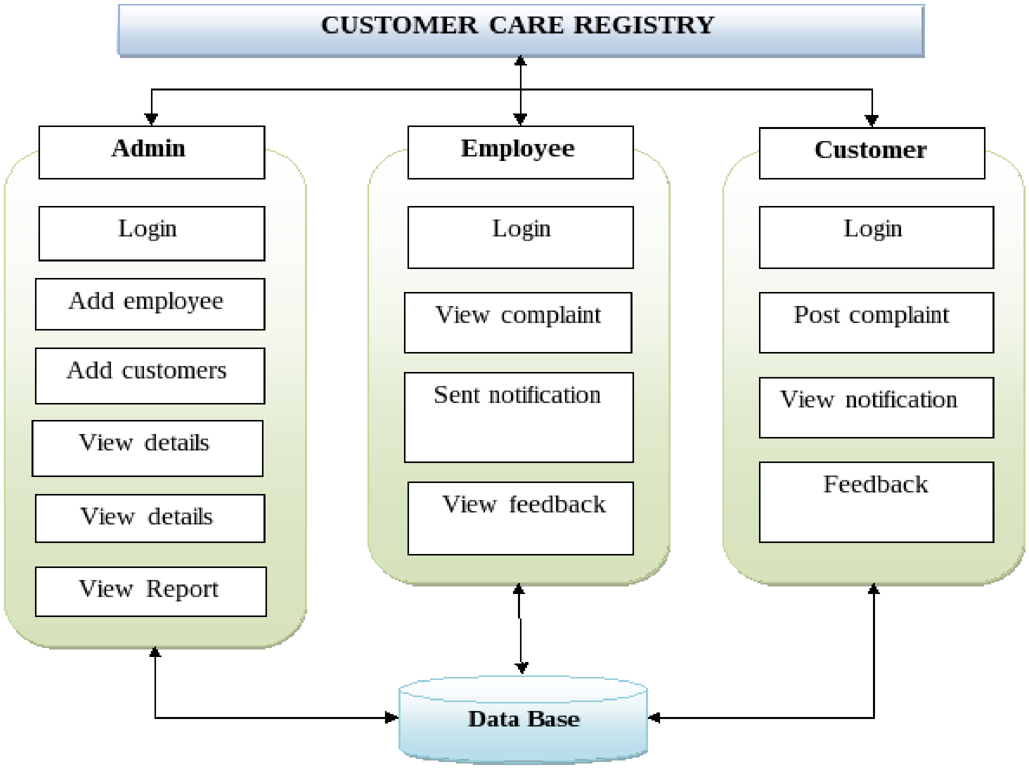


LEVEL3

Adataflowdiagram(DFD)isagraphicalrepresentationoftheflowofdatathroughaninformationsystem. ADFDshowstheflowofdatafromdatasourcesanddatastorestoprocesses, and from processes to data stores and data sinks. DFDs are used for modelling andanalyzing the flow of data in data processing systems, and are usually accompanied by a datadictionary, an entity-relationshipmodel, and anumber of processdescriptions.


Customer
Login
Post complaint
View notification
Feedback
Store the data
Store the data
Retrieve the data
Store the data

5.2 SOLUTION&TECHNICALARCHITECTURE



5.3 USER STORIES

UserType	Functional Requirement (Epic)	User StoryNumber	UserStory/Task	Acceptance criteria	Priority	Release
Customer	Creating newuser	USN-1	Anewuserscanregisterthroughthe portaldetails providedbythem	I can register tothe portal	High	Sprint1
	Login	USN-2	Theexistinguserscanlogin totheportalusingusername and password	I can login to theportal	High	Sprint1
Admin	Login	USN-3	Theadmincanloginto theportal	Successfullogin	High	Sprint2
	Newagent	USN-4	Theadmincancreate newagents	successful creationofagent	High	Sprint2
	ViewingListof Agents	USN-5	Theadmincanviewthe listofagents	Showingthelist ofagents	Medium	Sprint3
	AssigningAgent	USN-6	Theadmincanassignthe agentforthe	Successful assigningofagent	High	Sprint3
	Reviewing theProblem	USN-7	The admin can reviewtheproblemand	Successfulsending of theclarification throughEmail	Medium	Sprint3
Customer	Raisingcompliant	USN-8	A customer can raise thepersonalcompliantthroughthismodule	Successfulpostingofthecomplianttothe admin	High	Sprint4
Agent	Check for theComplaints	USN-9	An agent can view thecompliantsofthecustomerassignedto thembytheadmin	Showing the listofcompliants assigned	High	Sprint4

6. PROJECTPLANNING&SCHEDULING

6.1 SPRINTPLANNING &ESTIMATION

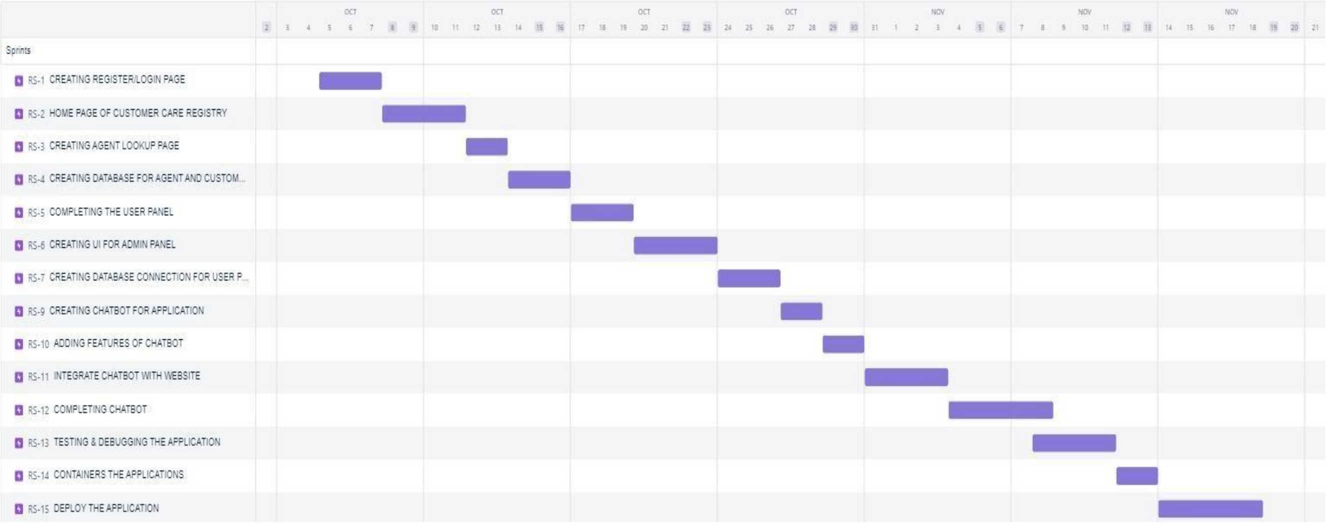
Sprint	Functional Requireme nt(Epic)	User StoryNu mber	UserStory/Task	Story Points	Priority	TeamMembers
Sprint-1	UserPanel	USN-1	Theuserwillloginintothewebsiteandgothroughtheservicesavailableonthewebpage	20	High	Gowtham B Gowri Shankar M Elankumaran M Bharathi Kannan M
Sprint-2	Homepage	USN-2	Homepageactasaninterfacetothisproject.Hereitprovidesguideandbrieftotheenduser	20	High	Gowtham B Gowri Shankar M Elankumaran M Bharathi Kannan M
Sprint-3	Adminpanel	USN-3	.Theroleoftheadministocheck out thedatabaseabouttheavailabilityandhaveatrackofallthethingsthattheusersaregoingtoservice	20	High	Gowtham B Gowri Shankar M Elankumaran M Bharathi Kannan M
Sprint-4	finaldelivery	USN-4	Finally It will store the data and theadmin will allocate the agent to thecorrespondingend user	20	High	Gowtham B Gowri Shankar M Elankumaran M BharathiKannan M

6.2 SPRINTDELIVERYSCHEDULE

Sprint	Total StoryPoints	Duration	SprintStartDate	SprintEndDate(Planned)	StoryPointsCompleted(ason PlannedEndDate)	SprintReleaseDate(Actual)
Sprint-1	20	6Days	24Oct 2022	29Oct 2022	20	29Oct2022
Sprint-2	20	6Days	31Oct 2022	05Nov2022	20	05Nov2022
Sprint-3	20	6Days	07Nov2022	12Nov2022	20	12Nov2022
Sprint-4	20	6Days	14Nov2022	19Nov2022	20	19Nov2022

6.3 REPORTS FORM

JIRABurndown Chart


OCT
OCT
OCT
OCT
NOV
NOV
NOV
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Sprints
RS-1 CREATING REGISTER/LOGIN PAGE
RS-2 HOME PAGE OF CUSTOMER CARE REGISTRY
RS-3 CREATING AGENT LOOKUP PAGE
RS-4 CREATING DATABASE FOR AGENT AND CUSTOM...
RS-5 COMPLETING THE USER PANEL
RS-6 CREATING UI FOR ADMIN PANEL
RS-7 CREATING DATABASE CONNECTION FOR USER P...
RS-9 CREATING CHATBOT FOR APPLICATION
RS-10 ADDING FEATURES OF CHATBOT
RS-11 INTEGRATE CHATBOT WITH WEBSITE
RS-12 COMPLETING CHATBOT
RS-13 TESTING & DEBUGGING THE APPLICATION
RS-14 CONTAINERS THE APPLICATIONS
RS-15 DEPLOY THE APPLICATION

7. CODING&SOLUTIONING

7.1 FEATURE

1NewCompliant

Acustomercomplainthighlightsaproblem,whetherthat'saproblemwithyourproduct,employeesorinternalprocesses,andbyhearingtheseproblemsdirectlyfromyourcustomers,youcaninvestigateandimprovetopreventfurthercomplaintsinthefuture.

Code

```
<html xmlns="">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Customercare</title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="default.css" rel="stylesheet" type="text/css" />
<style>

/*
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License
*/

body
</style>
</head>
<body>
<div id="header">
 <div id="logo">
        <h1><a href="#">Customer Care   </a></h1>
        <h2><a href=""></a></h2>
</div>
 <div id="menu">
        <ul>

        <li><a href="/UserHome">Home</a></li>
                              <li><a href="/NewComplaint" >NewComplaint</a></li>
<li><a href="/UserComplaint" >ComplaintInfo</a></li>
                              <li><a href="/">Logout</a></li>
        </ul>
 </div>
```

```
</div>
<div id="page">
 <div id="content">
            <div style="margin-bottom: 20px;">
                        <h1 class="title">New Complaint Info</h1>

            <form id="form1" name="form1" method="post" action="/newcom">
<table width="35%" border="0" align="center">

<tr>
<td></td>
</tr>
<tr>
<td width="45%" height="48">UserName</td>
<td width="55%"><input name="name" type="text" id="name"  value={{uname}} required />
</td>
</tr>
 <tr>
<td height="50">Complaint Info</td>
<td><textarea name="com" id="com" required></textarea></td>
</tr>
 <tr>
<td height="72">&nbsp;</td>
<td><input name="btn" type="submit" id="btn" value="Submit" />
<input type="reset" name="Submit2" value="Reset" /></td>
</tr>
</table>
</form>
            <p>&nbsp;</p>
                        </blockquote>
 </div>
            <div>&nbsp;</div>
 <div class="twocols"></div>
 </div>
 <!-- end content -->
 <!-- end sidebar -->
 <div style="clear: both;">&nbsp;</div>
</div>
<!-- end page -->
<div id="footer">
 <p id="legal"><a href="#" title="This page validates as CSS"><abbr title="Cascading Style Sheets"></abbr></a></p>
 </div>
```

```
<div align=center><a href='#'></a></div>
</body>
</html>
```

7.2 FEATURE

2Agent Assign

Whyisitimportantforacontactcenteragenttoknowtheneedsofcustomers?Itisessentialthatcustomersupportcontactcentersdevelopacomprehensiveunderstandingofwhatcustomersexpectfromthem,whetherornottheirneedsarebeingmetandhowtheycanimprove their service to meettheir expectations.

Code

```
<html xmlns="">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Customercare</title>
<meta name="keywords" content="" />
<meta name="description" content="" />
<link href="default.css" rel="stylesheet" type="text/css" />
<style>

/*
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License
*/

body
</style>
</head>
<body>
<div id="header">
 <div id="logo">
        <h1><a href="#">Customer Care   </a></h1>
        <h2><a href=""></a></h2>
</div>
 <div id="menu">
        <ul>

                <li><a href="/AdminHome">Home</a></li>
                               <li><a href="/NewAgent">NewAgent</a></li>
                               <li><a href="/AdminComplaintInfo">ComplaintInfo</a></li>
```

```
                                <li><a href="/">Logout</a></li>

        </ul>
 </div>
</div>
<div id="page">
 <div id="content">
        <div style="margin-bottom: 20px;">
                                <h1 class="title">Assign Agent </h1>

        <form id="form1" name="form1" method="post" action="/ass">
<table width="35%" border="0" align="center">

<tr>
<td></td>
</tr>
<tr>
<td width="45%" height="48">AgentId</td>
<td width="55%"><select name="agid" id="agid">
         {%for item1 in data%}
 <option>{{item1[7]}}</option>
                {%endfor%}</select>
</td>
</tr>
 <tr>
<td height="72">&nbsp;</td>
<td><input name="btn" type="submit" id="btn" value="Submit" />

</tr>
</table>
</form>
        <p>&nbsp;</p>
                </blockquote>
 </div>
        <div>&nbsp;</div>
 <div class="twocols"></div>
 </div>
 <!-- end content -->
 <!-- end sidebar -->
 <div style="clear: both;">&nbsp;</div>
```

```
</div>
<!-- end page -->
<div id="footer">
 <p id="legal"><a href="#" title="This page validates as CSS"><abbr title="Cascading Style Sheets"></abbr></a></p>
 </div>
<div align=center><a href='#'></a></div>
</body>
</html>
```

8. TESTING

8.1 TESTCASES

A test case has components that describe input, action and an expected response, in order todetermine if a feature of an application is working correctly. A test case is a set of instructions on“HOW” to validate a particular test objective/target, which when followed will tell us if theexpected behavior of the systemis satisfied or not.

Characteristicsofagoodtestcase:

- Accurate:Exactsthepurpose.
- Economical:Nounnecessarystepsorwords.
- Traceable:Capableofbeingtracedtorequirements.

- Repeatable: Canbeused toperformthe testover andover.

- Reusable:Can be reused ifnecessary.

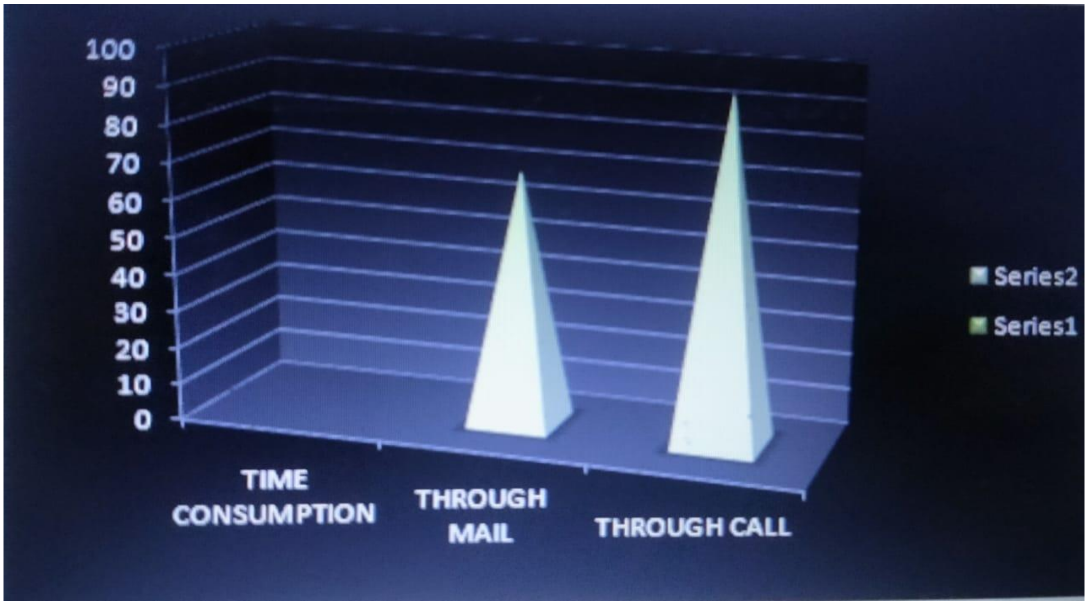
S.NO	Scenario	Input	Exceptedoutput	Actualoutput
1	AdminLoginForm	User and name password	Login	Loginsuccess.
2	EmployeeLoginForm	User name and password	Login	Loginsuccess.
3	UserRegistrationForm	Userbasicdetails	Registereds uccessfully	Userbasicdetailsares toredinthe database.
4	UserLoginForm	User name and password	Login	Loginsuccess.

8.2 USERACCEPTANCETESTING

Thisis a type of testing done by users, customers, or other authorised entities to determineapplication/software needs and business processes.Acceptance testing is the most importantphaseoftestingasthisdecideswhethertheclientapprovestheapplication/softwareornot.Itmay involve functionality, usability, performance, and U.I of the application. It is also known asuseracceptancetesting(UAT),operationalacceptancetesting(OAT),and end-usertesting.

9. RESULTS

9.1 PERFORMANCEMETRICS



10. ADVANTAGES&DISADVANTAGES

ADVANTAGES

- System is easy to understandand user friendly.
- Thesystemispurelybasedonpredictionwhichpredictsaninternetplanforthecustomer.
- Admincaneasilyviewemployeereportbasedontheresolutionprovidedonthecompla int.
- Handlelargenumberofcontextualinformation.
- Userfriendlyandtimeconsumingprocess.
- Usingthisproject,theusercanknowaboutstatusofcomplaintthroughwebsite.
- Keeptrackofdailyinformationexchangeattheserverbytheadministrator.
- Increaseinprocessingandtransferspeedsofinformationoverthenetwork.

DISADVANTAGES

- Requiresanactiveinternetconnection.
- System mayprovideinaccurateresultsifthedataenteredincorrectly.
- Difficulttoprovideproperintimationsystem
- Currentsystemismanualprocess
- Cannotalwaystakingacall
- Towerproblem duringcall conversation

11. CONCLUSION

Application software has been computed successfully and was also tested successfully by taking“test cases”. It is user friendly, and has required option, which can be utilized by the user toperform the desired operations. Application meets the information requirements specified to agreatextent.Thesystemhasbeendesignedkeepinginviewthepresentandfuturerequirementsin mind and made very flexible. The goals that are achieved by the software areInstant access,improvedproductivity,Optimumutilizationofresources,Efficientmanagementofrecords,Simplifications of the operations, Less processing time and getting required information, Userfriendly,Portable and flexible for further enhancement. The system has the benefits of easyaccess because it is be developed as a platform independent web application, so the admin canmaintain a proper contact with their users, which may be access anywhere. All communicationsbetweenthepoliceandadministratorhasdonethroughtheonline,sothiscommunicationcostalso is reduced.

12. FUTURESCOPE

In future we can develop this project in android application with extra features like customercomplaint systemandcollectthefeedbackformfromthecustomer aboutthesystem

13. APPENDIX

Flask:

✓ FlaskisamicrowebframeworkwritteninPython.Itisclassifiedasamicroframeworkbecause it does notrequire particular toolsor libraries

✓ Ithasnodatabaseabstractionlayer,formvalidation,oranyothercomponentswherepre-existingthird-partylibrariesprovidecommonfunctions

JavaScript:

✓ JavaScript,oftenabbreviatedasJS,isaprogramminglanguagethatisoneofthecoretechnologiesofthe WorldWide Web,alongside HTMLand CSS

✓ Asof2022,98%ofwebsitesuseJavaScriptontheclientsideforwebpagebehavior,often incorporatingthird-partylibraries

IBMCloud:

✓ IBM cloudcomputingisasetofcloudcomputingservicesforbusinessofferedbytheinformation technology company IBM

Kubernetes:

✓ Kubernetesisanopen-sourcecontainerorchestrationsystemforautomatingsoftwaredeployment, scaling, and management

Docker:

✓ DockerisasetofplatformsasaserviceproductthatuseOS-levelvirtualizationtodeliversoftware in packages called containers

SOURCECODE

```
from flask import Flask, render_template, flash, request, session,send_file
    from flask import render_template, redirect, url_for, request

    import ibm_db
    import pandas
    import ibm_db_dbi
    from sqlalchemy import create_engine

    engine = create_engine('sqlite://',
                    echo = False)

    dsn_hostname = "125f9f61-9715-46f9-9399-
    c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"
    dsn_uid = "fxc49884"
    dsn_pwd = "2VfAhFMP4u6zY6EB"

    dsn_driver = "{IBM DB2 ODBC DRIVER}"
    dsn_database = "BLUDB"
    dsn_port = "30426"
    dsn_protocol = "TCPIP"
    dsn_security = "SSL"

    dsn = (
       "DRIVER={0};"
       "DATABASE={1};"
       "HOSTNAME={2};"
       "PORT={3};"
       "PROTOCOL={4};"
       "UID={5};"
       "PWD={6};"
       "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol,
    dsn_uid, dsn_pwd,dsn_security)
```

```
try:
    conn = ibm_db.connect(dsn, "", "")
    print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)

except:
    print ("Unable to connect: ", ibm_db.conn_errormsg() )



app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'

@app.route("/")
def homepage():

    return render_template('index.html')

@app.route("/AdminLogin")
defAdminLogin():

    return render_template('AdminLogin.html')



@app.route("/UserLogin")
defUserLogin():
    return render_template('UserLogin.html')



@app.route("/NewUser")
defNewUser():
    return render_template('NewUser.html')
```

```
@app.route("/NewComplaint")
defNewComplaint():
    user = session['uname']
    return render_template('NewComplaint.html',uname=user)

@app.route("/NewAgent")
defNewAgent():

    conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)
selectQuery = "SELECT * FROM agenttb where  "
dataframe = pandas.read_sql(selectQuery, pd_conn)
dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('NewAgent.html',data=data)

@app.route("/AdminHome")
defAdminHome():
    conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)

selectQuery = "SELECT * from regtb "
dataframe = pandas.read_sql(selectQuery, pd_conn)

dataframe.to_sql('Employee_Data',
                    con=engine,
if_exists='append')

    # run a sql query
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()
    return render_template('AdminHome.html',data=data)
```

```
@app.route("/UserHome")
defUserHome():
    user = session['uname']

    conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)
selectQuery = "SELECT * FROM regtb where  UserName= '" + user + "' "
dataframe = pandas.read_sql(selectQuery, pd_conn)
dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()
    return render_template('UserHome.html',data=data)



@app.route("/UserComplaint")
defUserComplaint():
    user = session['uname']

    conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)
selectQuery = "SELECT * FROM booktb where  UserName= '" + user + "' "
dataframe = pandas.read_sql(selectQuery, pd_conn)
dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('UserComplaint.html',data=data)

@app.route("/AdminComplaintInfo")
defAdminComplaintInfo():



    conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)
selectQuery = "SELECT * FROM booktb    "
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('AdminComplaintInfo.html',data=data)

@app.route("/adminlogin", methods=['GET', 'POST'])
defadminlogin():
    error = None
    if request.method == 'POST':
       if request.form['uname'] == 'admin' or request.form['password'] == 'admin':

           conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)

selectQuery = "SELECT * from regtb "
dataframe = pandas.read_sql(selectQuery, pd_conn)

dataframe.to_sql('Employee_Data',
                            con=engine,
if_exists='append')

           # run a sql query
           data = engine.execute("SELECT * FROM Employee_Data").fetchall()
           return render_template('AdminHome.html' , data=data)

       else:
        return render_template('index.html', error=error)
```

```
@app.route("/userlogin", methods=['GET', 'POST'])
defuserlogin():

    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['uname'] = request.form['uname']

        conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)

selectQuery = "SELECT * from regtb where UserName='" + username + "' and password='" + 
password + "'"
dataframe = pandas.read_sql(selectQuery, pd_conn)

        if dataframe.empty:
            data1 = 'Username or Password is wrong'
            return render_template('goback.html', data=data1)
        else:
            print("Login")
selectQuery = "SELECT * from regtb where UserName='" + username + "' and password='" + 
password + "'"
dataframe = pandas.read_sql(selectQuery, pd_conn)

dataframe.to_sql('Employee_Data',
                             con=engine,
if_exists='append')

            # run a sql query
            data = engine.execute("SELECT * FROM Employee_Data").fetchall()

            return render_template('UserHome.html', data=data )
```

```
@app.route("/newuser", methods=['GET', 'POST'])
defnewuser():
    if request.method == 'POST':

        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
pnumber = request.form['phone']
        address = request.form['address']

uname = request.form['uname']
        password = request.form['psw']



        conn = ibm_db.connect(dsn, "", "")

insertQuery = "INSERT INTO regtb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" +
email + "','" + pnumber + "','" + address + "','" + uname + "','" + password + "')"
insert_table = ibm_db.exec_immediate(conn, insertQuery)
        print(insert_table)



    return render_template('UserLogin.html')



@app.route("/newage", methods=['GET', 'POST'])
defnewage():
    if request.method == 'POST':

        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
```

```
pnumber = request.form['phone']
        address = request.form['address']

uname = request.form['uname']

        conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)

insertQuery = "INSERT INTO agenttb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" + 
email + "','" + pnumber + "','" + address + "','" + uname + "')"
insert_table = ibm_db.exec_immediate(conn, insertQuery)
        print(insert_table)

selectQuery = "SELECT * FROM agenttb  "
dataframe = pandas.read_sql(selectQuery, pd_conn)
dataframe.to_sql('booktb1', con=engine, if_exists='append')
        data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('NewAgent.html',data=data)

@app.route("/newcom", methods=['GET', 'POST'])
defnewcom():
    if request.method == 'POST':

        name = request.form['name']
        com = request.form['com']
uname = session['uname']



        conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)

selectQuery = "SELECT  *  FROM  booktb"
dataframe = pandas.read_sql(selectQuery, pd_conn)
```

```
dataframe.to_sql('booktb', con=engine, if_exists='append')
    data2 = engine.execute("SELECT * FROM booktb").fetchall()
    count = 0

    for item in data2:
      count += 1

Bookingid = "COMID00" + str(count)

insertQuery = "INSERT INTO booktb VALUES ('" + Bookingid + "','" + uname + "','" + com +
"','','')"
insert_table = ibm_db.exec_immediate(conn, insertQuery)
    print(insert_table)

selectQuery = "SELECT * FROM booktb where  UserName= '" + uname + "' "
dataframe = pandas.read_sql(selectQuery, pd_conn)

dataframe.to_sql('booktb1', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM booktb1").fetchall()
    return render_template('UserComplaint.html', data=data)



@app.route("/AgentAssign", methods=['GET'])
defAgentAssign():
cid = request.args.get('id')
  session['cid'] = cid

  conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)

selectQuery = "SELECT * FROM agenttb  "
dataframe = pandas.read_sql(selectQuery, pd_conn)

dataframe.to_sql('booktb1', con=engine, if_exists='append')
```

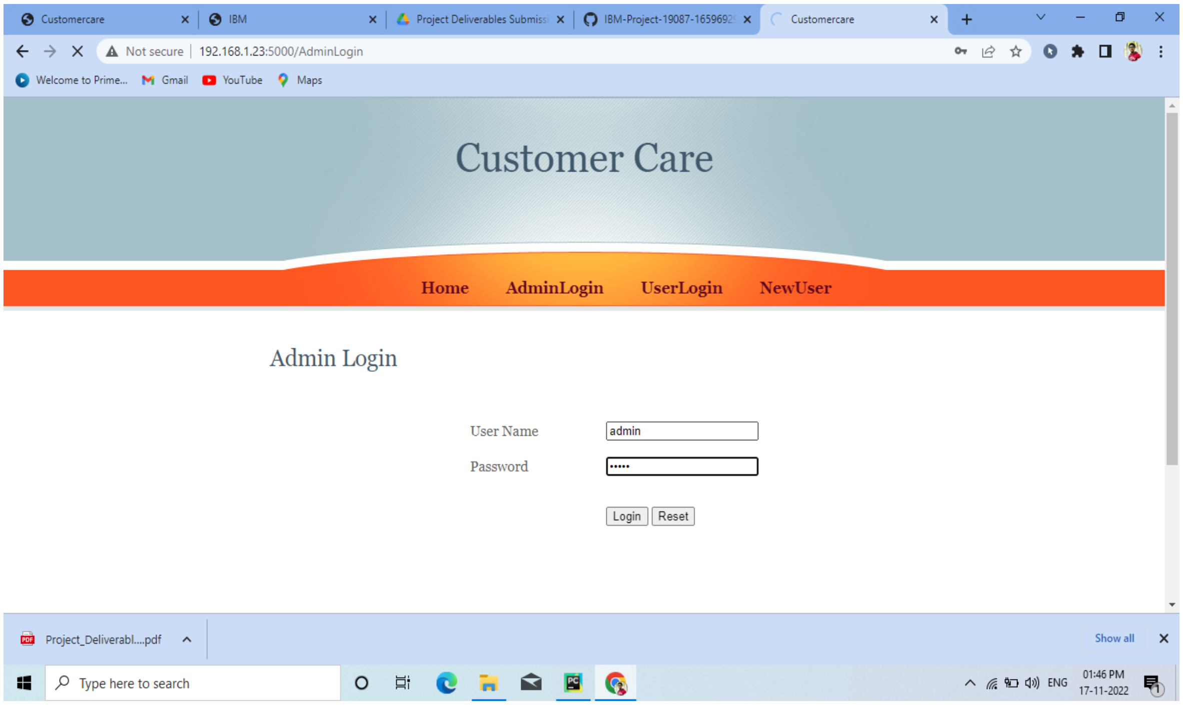
```
    data = engine.execute("SELECT * FROM booktb1").fetchall()

    return render_template('AgentAssign.html',data=data)



@app.route("/Action", methods=['GET'])
def Action():
cid = request.args.get('id')
    session['cid'] = cid

    return render_template('Action.html')



@app.route("/ass", methods=['GET', 'POST'])
def ass():

agid = request.form['agid']

cid = session['cid']

uname = session['uname']

    conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery1 = "SELECT  *  FROM  regtb where  UserName='" + uname + "'"
dataframe = pandas.read_sql(selectQuery1, pd_conn)

dataframe.to_sql('regtb', con=engine, if_exists='append')
    data1 = engine.execute("SELECT * FROM regtb").fetchall()

    for item1 in data1:
        Mobile = item1[5]
        Email = item1[4]
```

```
sendmsg(Email,"Assign Agent id"+agid)

insertQuery = "update booktb set AgentName='"+ agid +"' where ComplaintId='"+ cid +"' "
insert_table = ibm_db.exec_immediate(conn, insertQuery)

    alert = 'Agent Assign Send Notication'

    return render_template('goback.html', data=alert)



@app.route("/acc", methods=['GET', 'POST'])
defacc():

    com = request.form['com']

cid = session['cid']

uname = session['uname']

    conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery1 = "SELECT  *  FROM  regtb where  UserName='" + uname + "'"
dataframe = pandas.read_sql(selectQuery1, pd_conn)

dataframe.to_sql('regtb', con=engine, if_exists='append')
    data1 = engine.execute("SELECT * FROM regtb").fetchall()

    for item1 in data1:
        Mobile = item1[5]
        Email = item1[4]
sendmsg(Email,"Action Information "+com)

insertQuery = "update booktb set ACTIONINFO='"+ com +"' where ComplaintId='"+ cid +"' "
```

```
insert_table = ibm_db.exec_immediate(conn, insertQuery)

    alert = 'Action Info Saved Send Notication'

    return render_template('goback.html', data=alert)

defsendmsg(Mailid,message):
    import smtplib
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    from email.mime.base import MIMEBase
    from email import encoders

fromaddr = "gowrishankarit2001@gmail.com"
toaddr = Mailid

    # instance of MIMEMultipart
msg = MIMEMultipart()

    # storing the senders email address
msg['From'] = fromaddr

    # storing the receivers email address
msg['To'] = toaddr

    # storing the subject
msg['Subject'] = "Alert"

    # string to store the body of the mail
    body = message

    # attach the body with the msg instance
msg.attach(MIMEText(body, 'plain'))
```

```
    # creates SMTP session
    s = smtplib.SMTP('smtp.gmail.com', 587)

    # start TLS for security
s.starttls()

    # Authentication
s.login(fromaddr, "yijcqhwewodudjoy")

    # Converts the Multipart msg into a string
    text = msg.as_string()

    # sending the mail
s.sendmail(fromaddr, toaddr, text)

    # terminating the session
s.quit()



if __name__ == '__main__':
app.run(host='0.0.0.0', debug='TRUE')
```

SCREENSHOTS



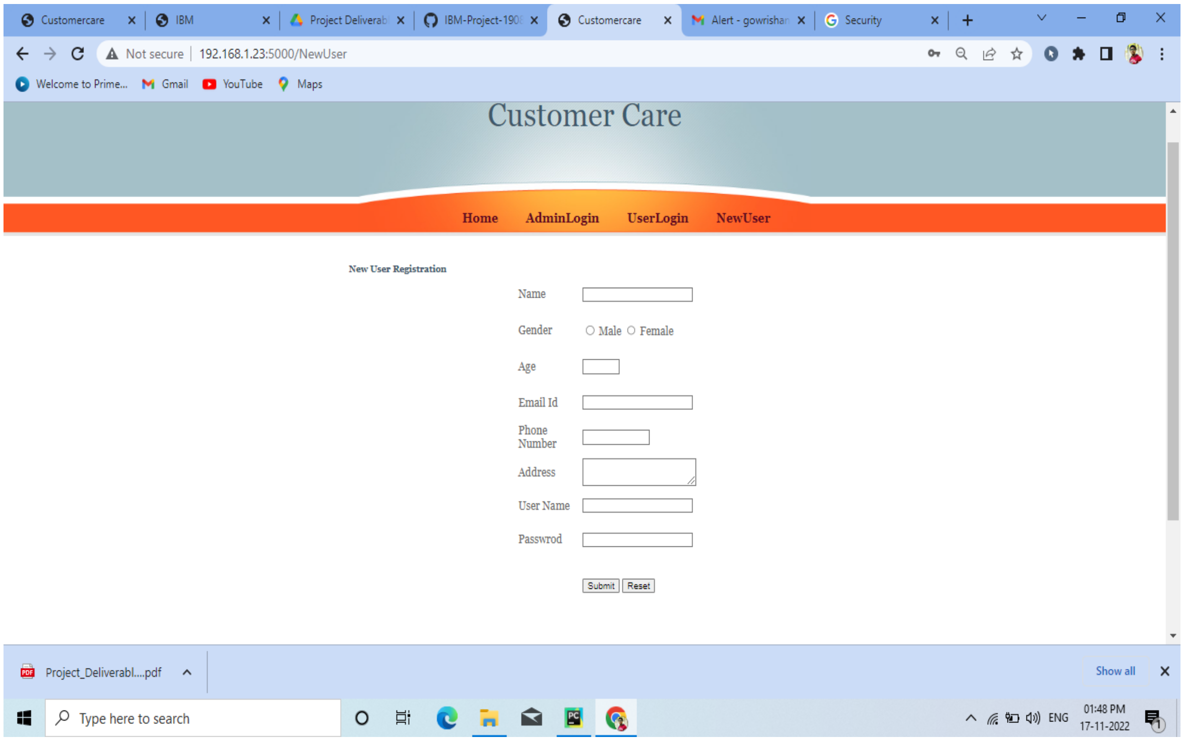
Customer Care

Home AdminLogin UserLogin NewUser

Admin Login

User Name	admin
Password	•••••

Login Reset



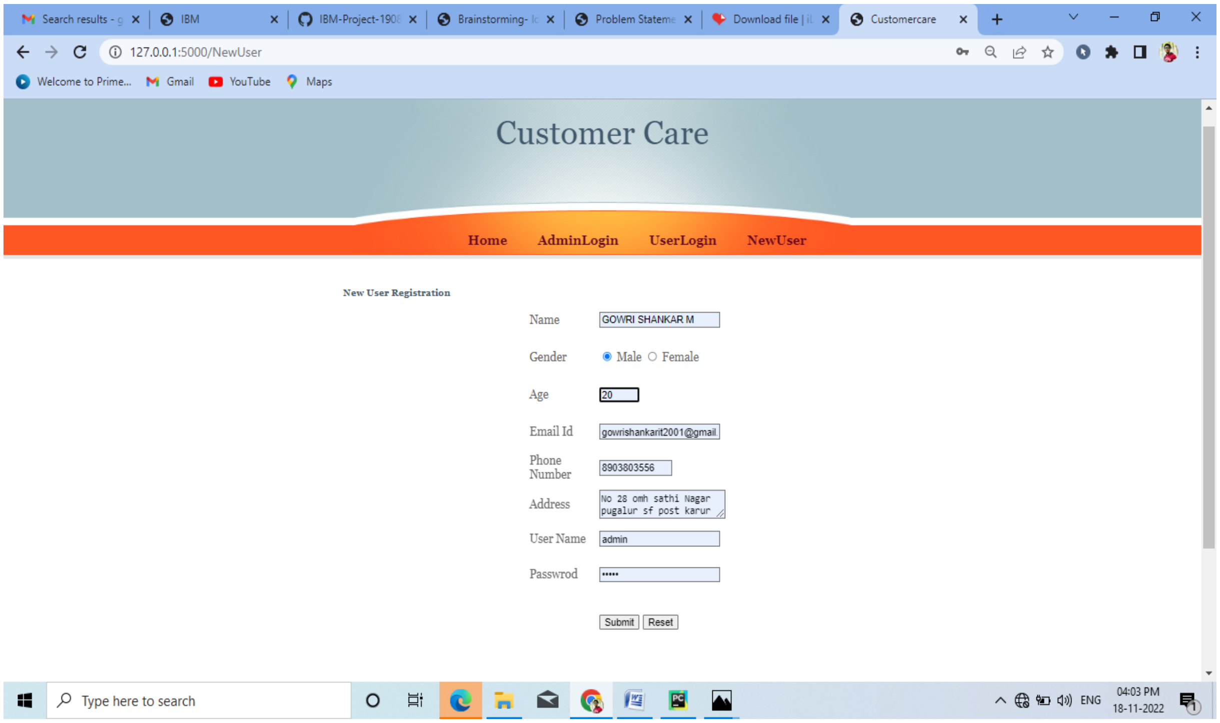
Customer Care

Home AdminLogin UserLogin NewUser

New User Registration

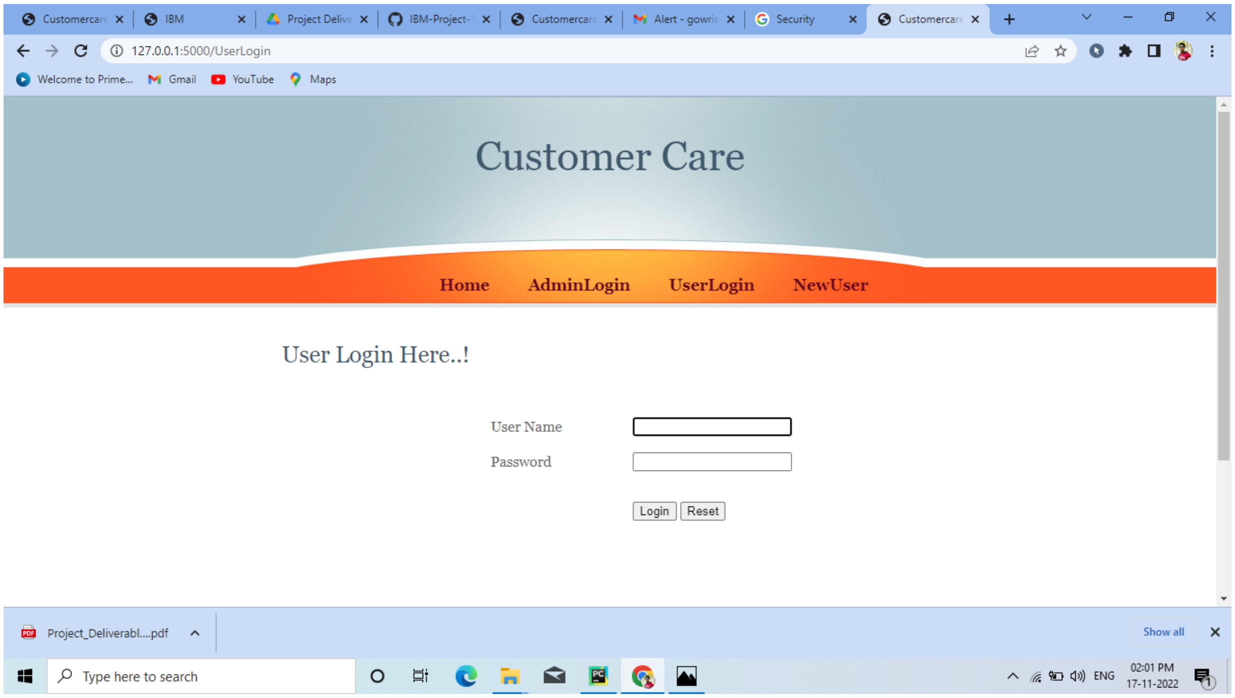
Name	
Gender	○ Male ○ Female
Age	
Email Id	
Phone Number	
Address	
User Name	
Passwrod	

Submit Reset



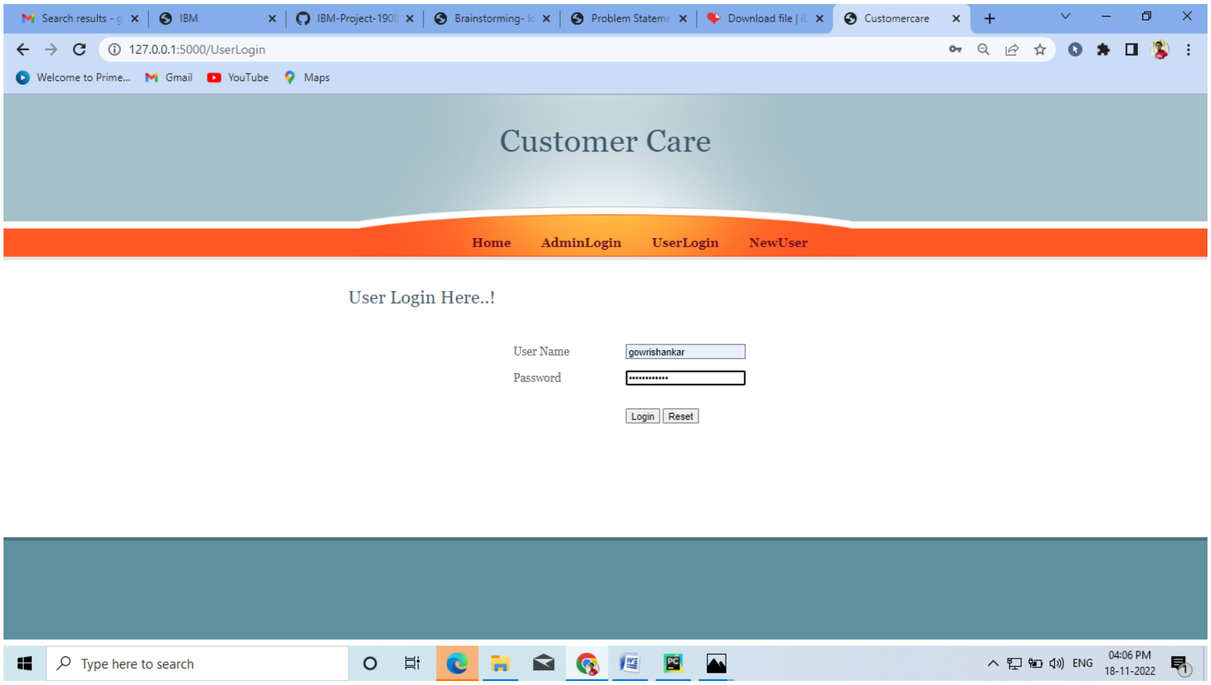
New User Registration

Name	GOWRI SHANKAR M
Gender	● Male ○ Female
Age	20
Email Id	gowrishankarit2001@gmail.
Phone Number	8903803556
Address	No 28 omh sathi Nagar pugalur sf post karur
User Name	admin
Passwrod	•••••



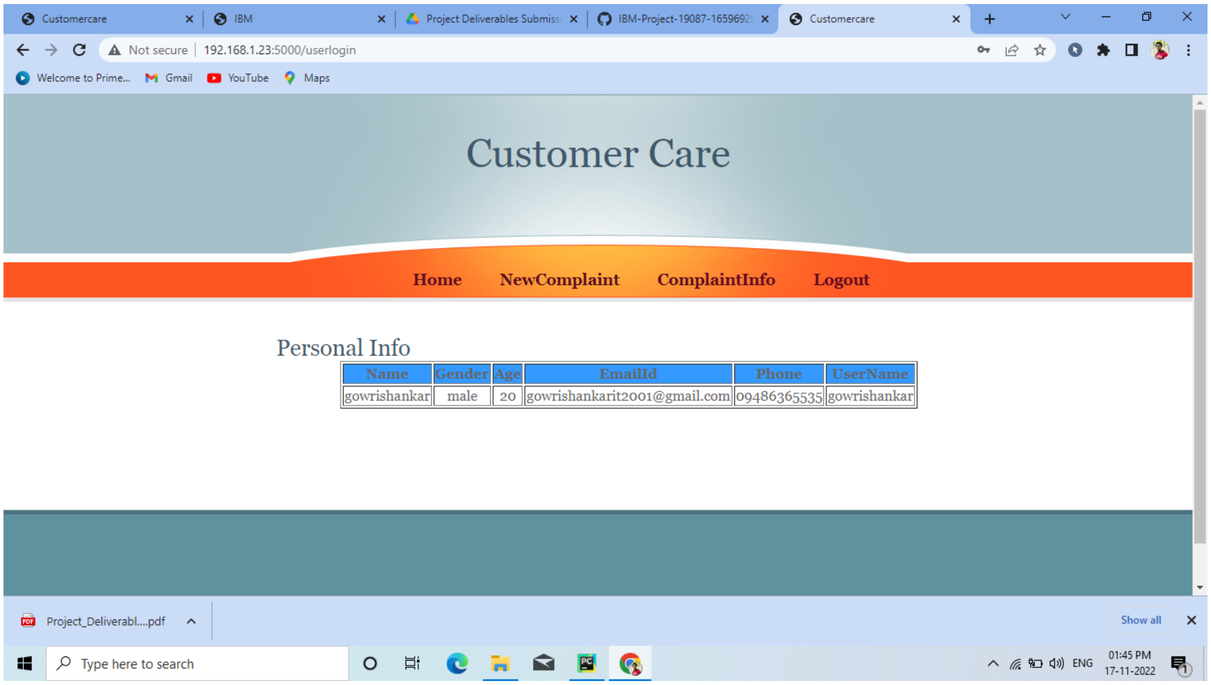
User Login Here..!

User Name	
Password	



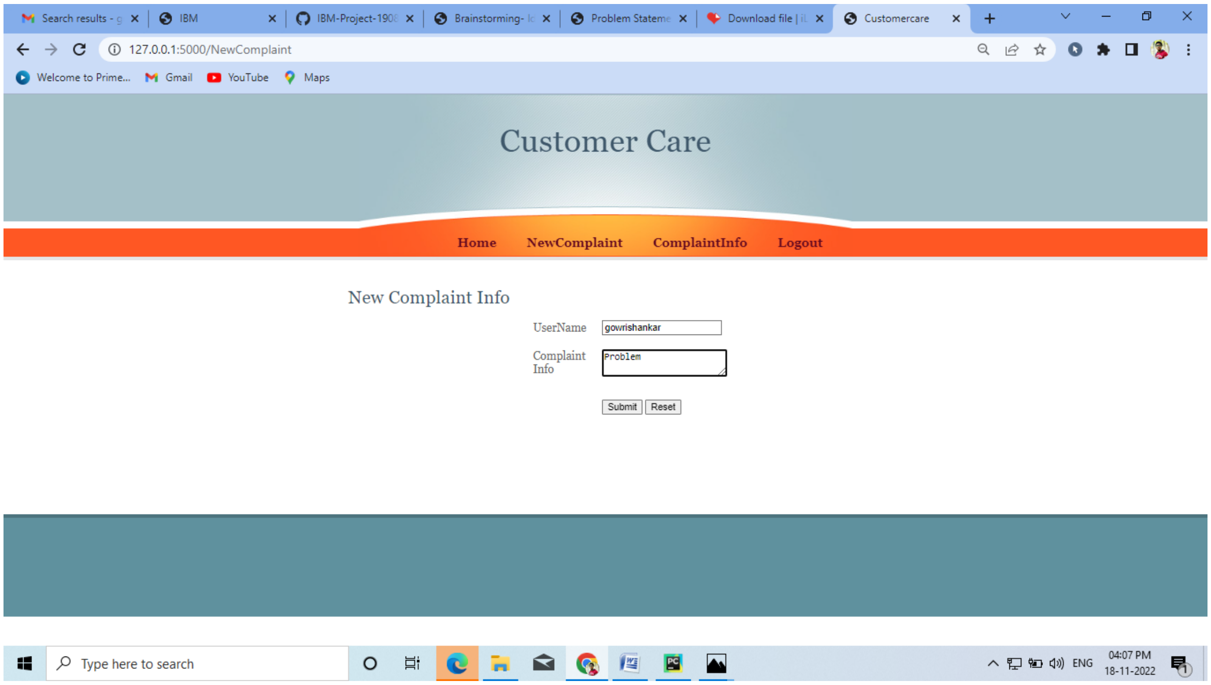
User Login Here..!

User Name	gowrishankar
Password	••••••••••••



Personal Info

Name	Gender	Age	EmailId	Phone	UserName
gowrishankar	male	20	gowrishankarit2001@gmail.com	09486365535	gowrishankar

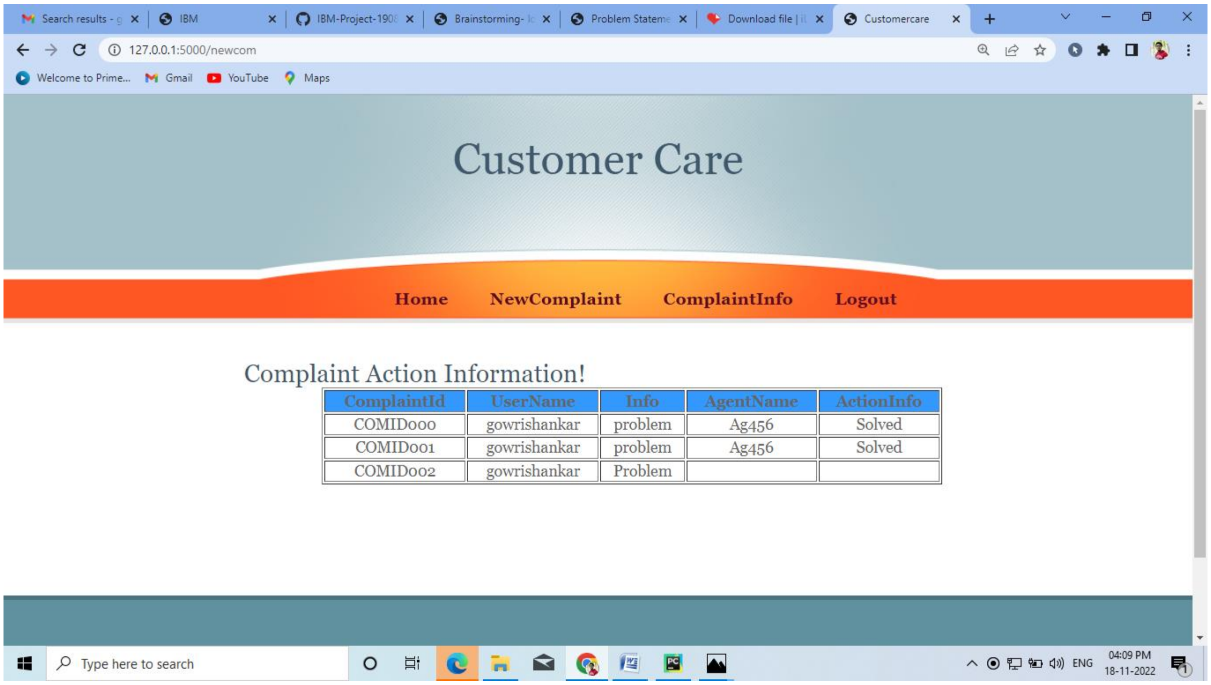


Customer Care

New Complaint Info

UserName gowrishankar

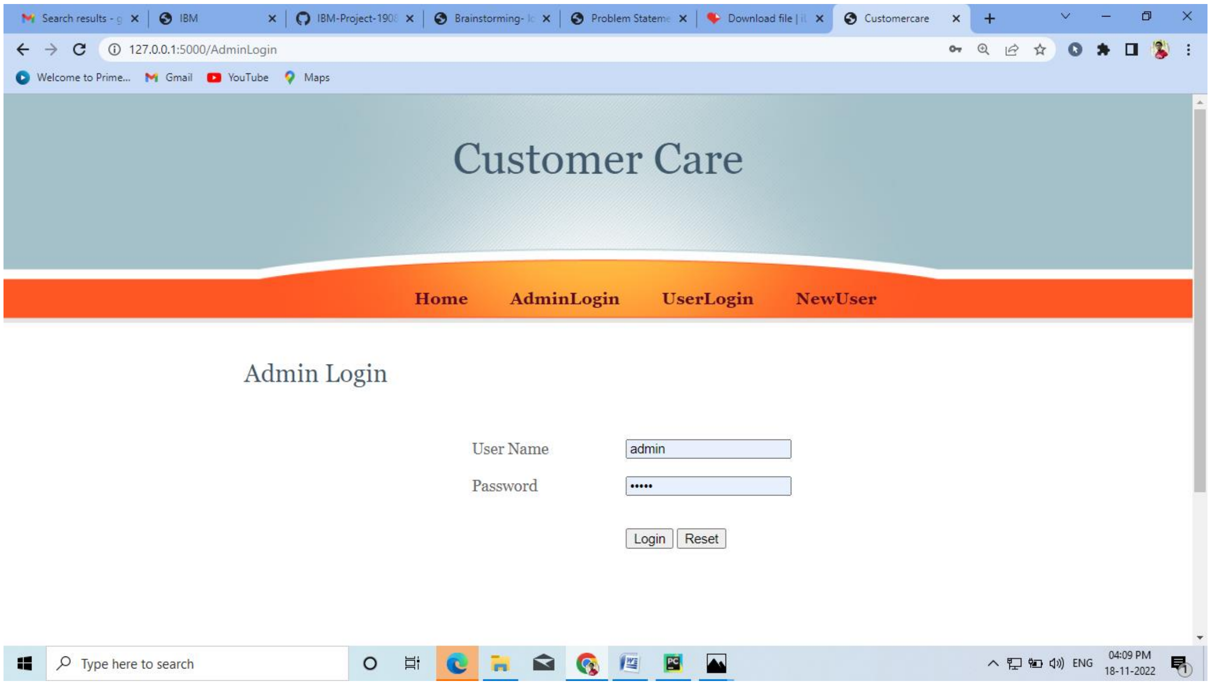
Complaint Info Problem



Customer Care

Complaint Action Information!

ComplaintId	UserName	Info	AgentName	ActionInfo
COMID000	gowrishankar	problem	Ag456	Solved
COMID001	gowrishankar	problem	Ag456	Solved
COMID002	gowrishankar	Problem		



Customer Care

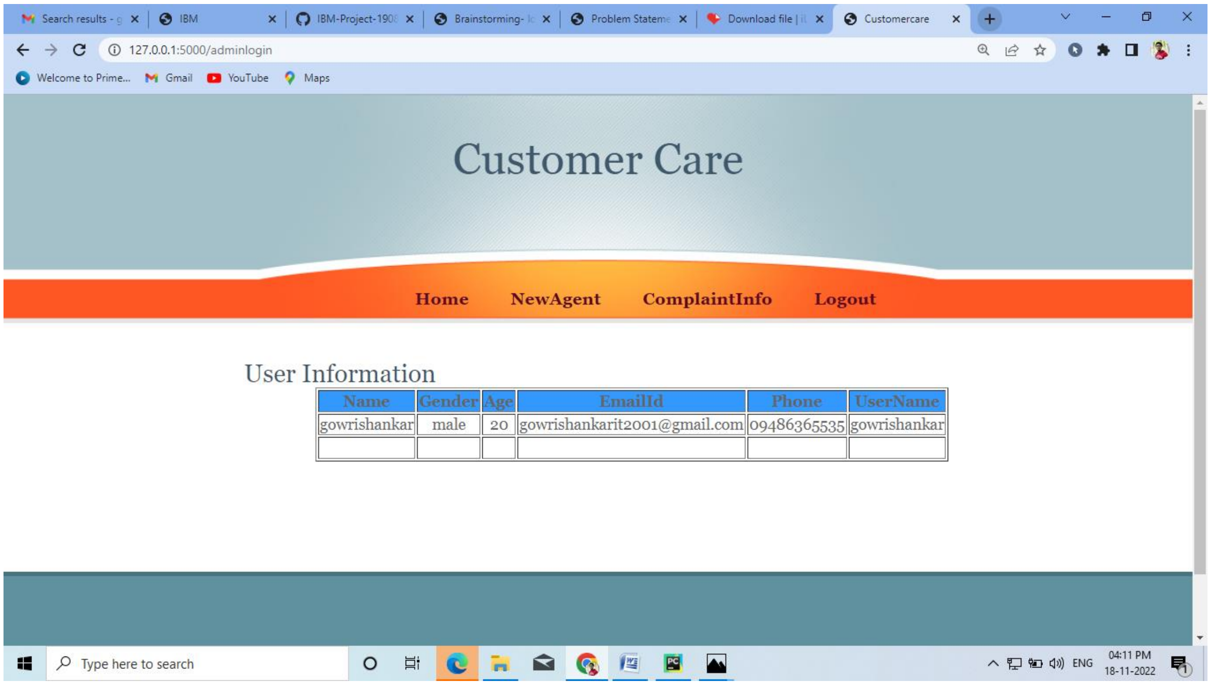
Home AdminLogin UserLogin NewUser

Admin Login

User Name: admin

Password: •••••

Login Reset

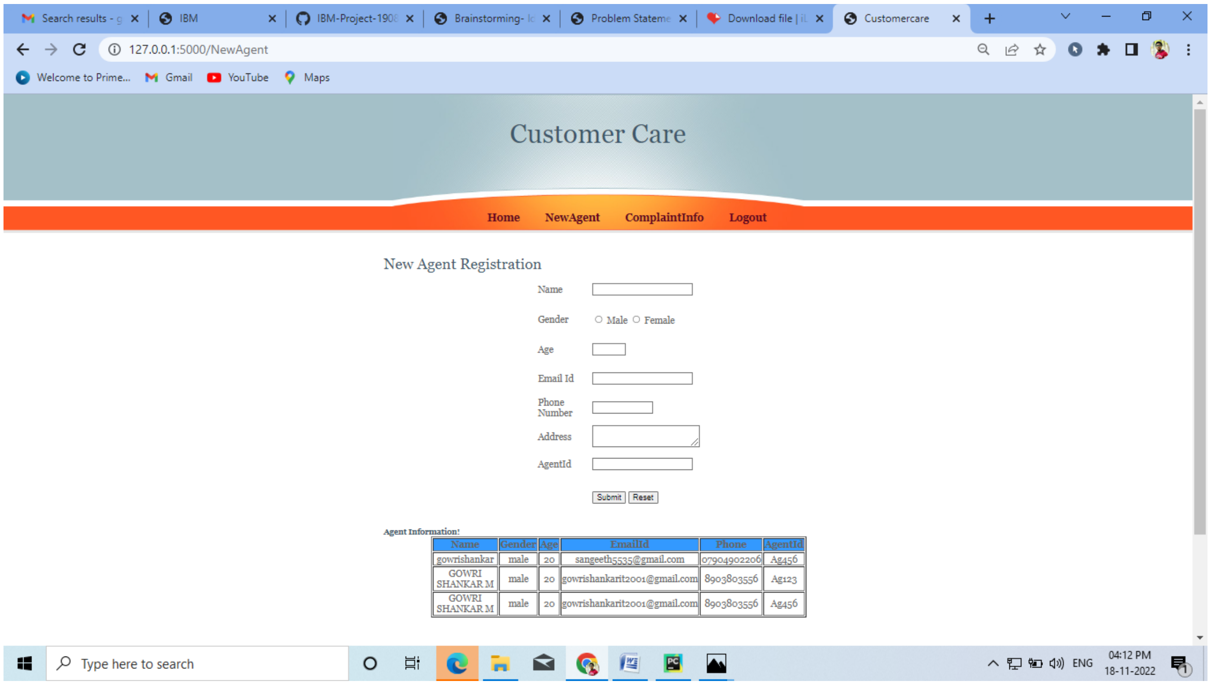


Customer Care

Home NewAgent ComplaintInfo Logout

User Information

Name	Gender	Age	EmailId	Phone	UserName
gowrishankar	male	20	gowrishankarit2001@gmail.com	09486365535	gowrishankar



Customer Care

Home NewAgent ComplaintInfo Logout

New Agent Registration

Name

Gender ○ Male ○ Female

Age

Email Id

Phone Number

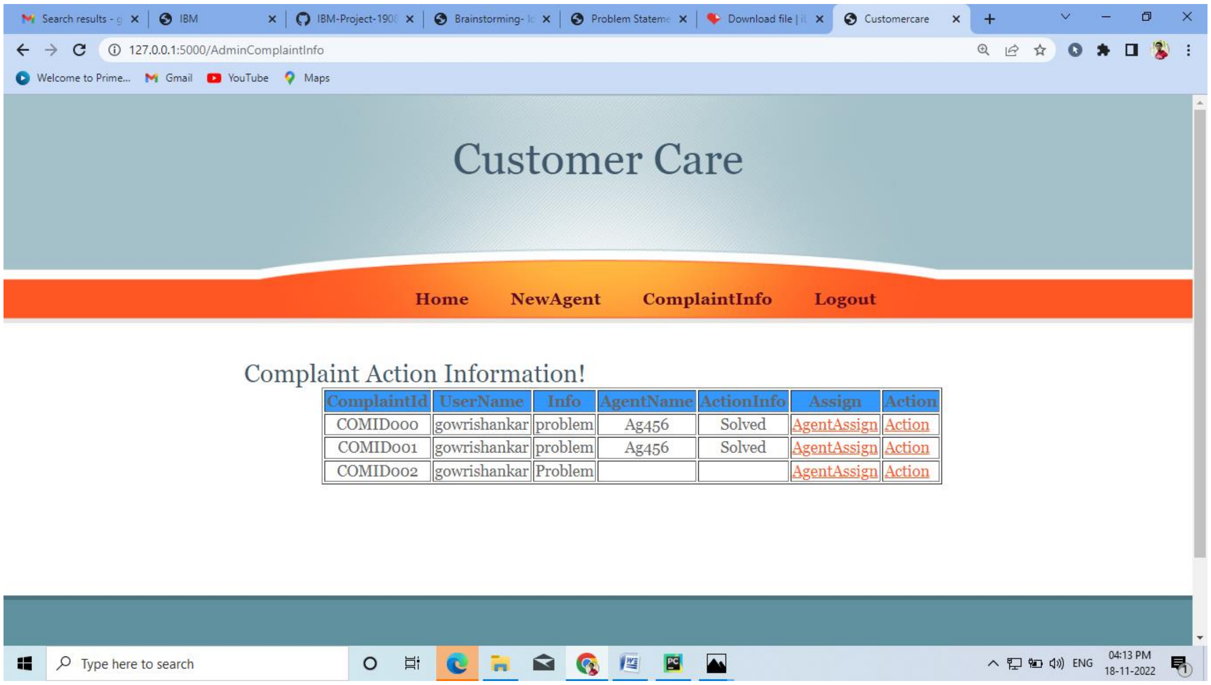
Address

AgentId

Submit Reset

Agent Information!

Name	Gender	Age	EmailId	Phone	AgentId
gowrishankar	male	20	sangeeth5535@gmail.com	07904902206	Ag456
GOWRI SHANKAR M	male	20	gowrishankarit2001@gmail.com	8903803556	Ag123
GOWRI SHANKAR M	male	20	gowrishankarit2001@gmail.com	8903803556	Ag456

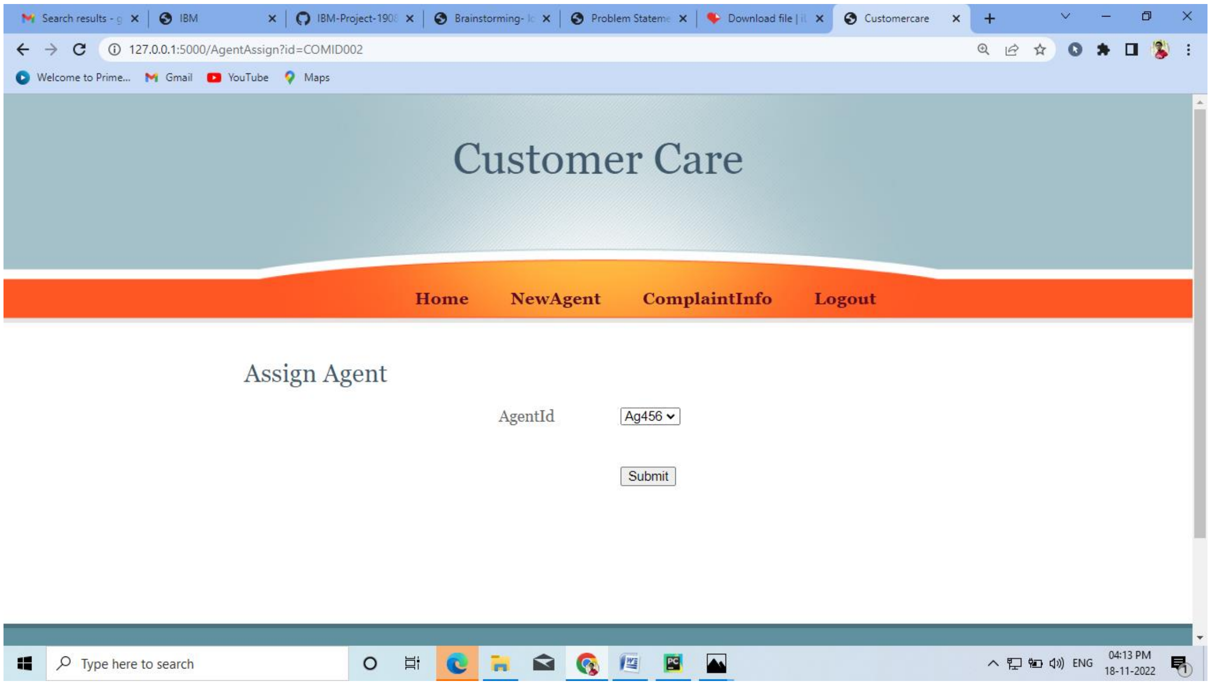


Customer Care

Home NewAgent ComplaintInfo Logout

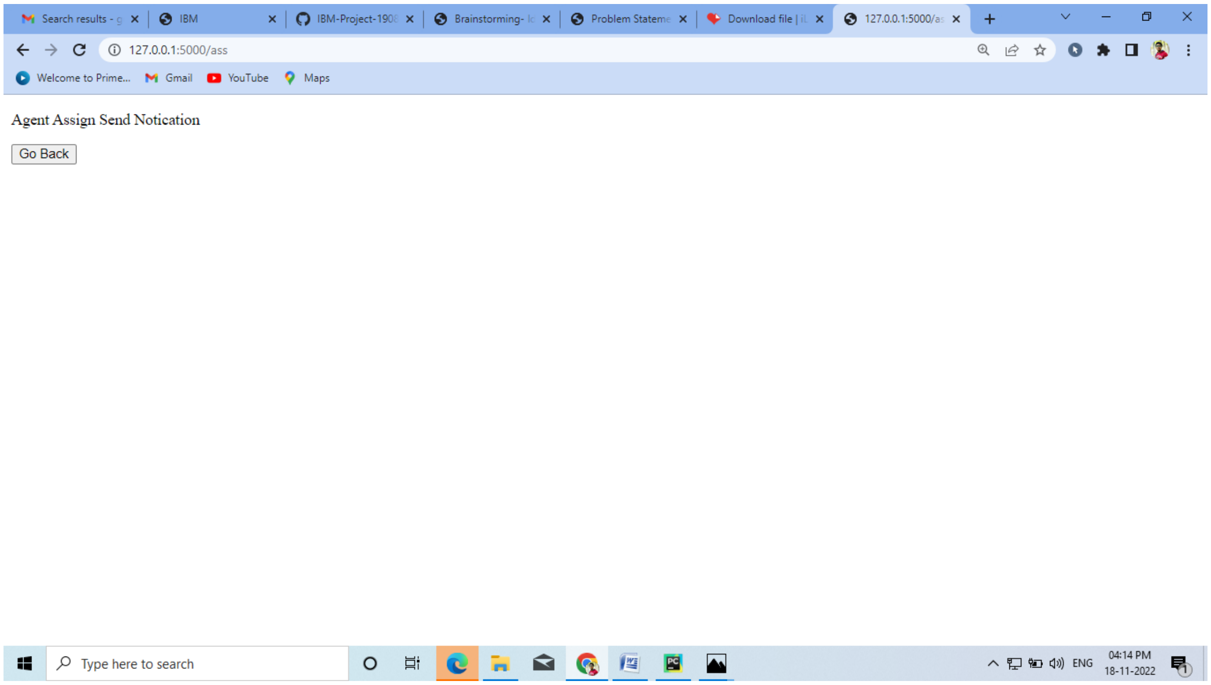
Complaint Action Information!

ComplaintId	UserName	Info	AgentName	ActionInfo	Assign	Action
COMID000	gowrishankar	problem	Ag456	Solved	AgentAssign	Action
COMID001	gowrishankar	problem	Ag456	Solved	AgentAssign	Action
COMID002	gowrishankar	Problem			AgentAssign	Action

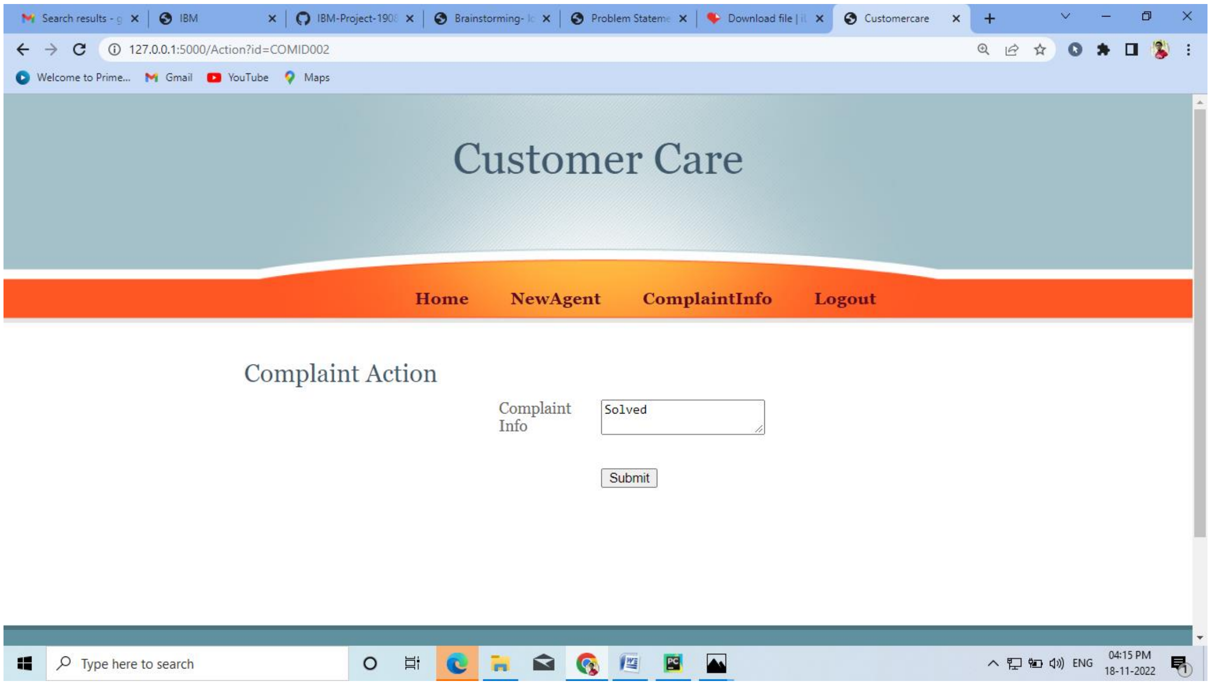


Assign Agent

AgentId Ag456



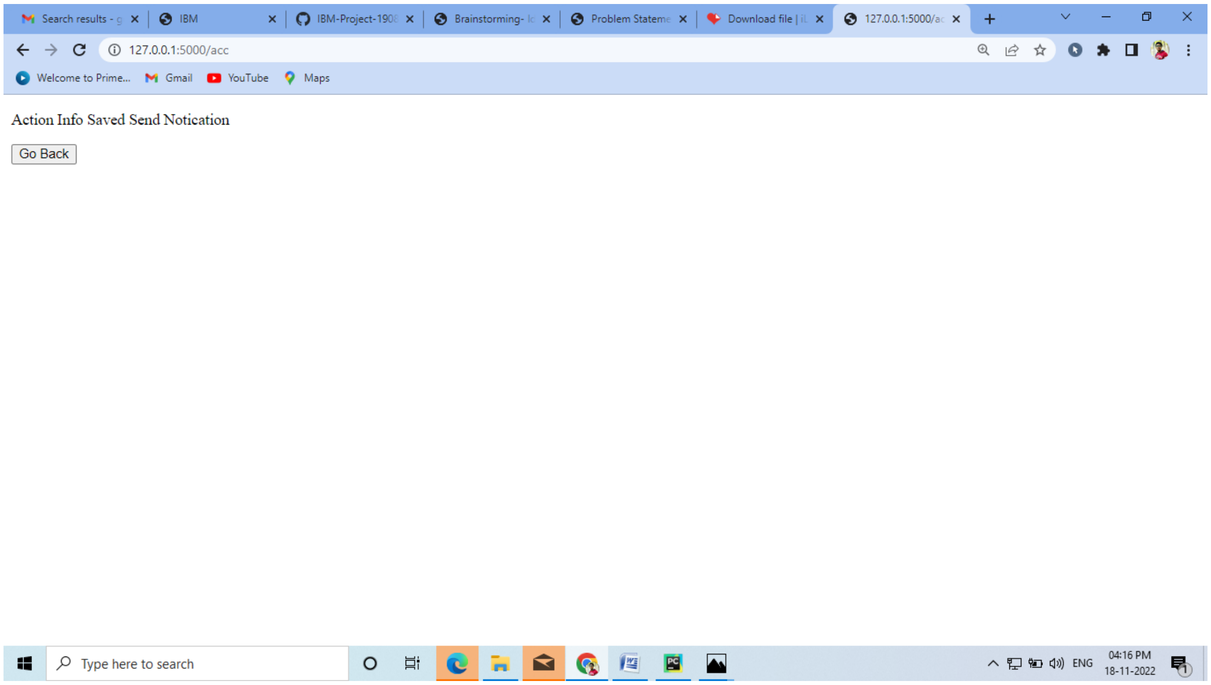
Agent Assign Send Notication



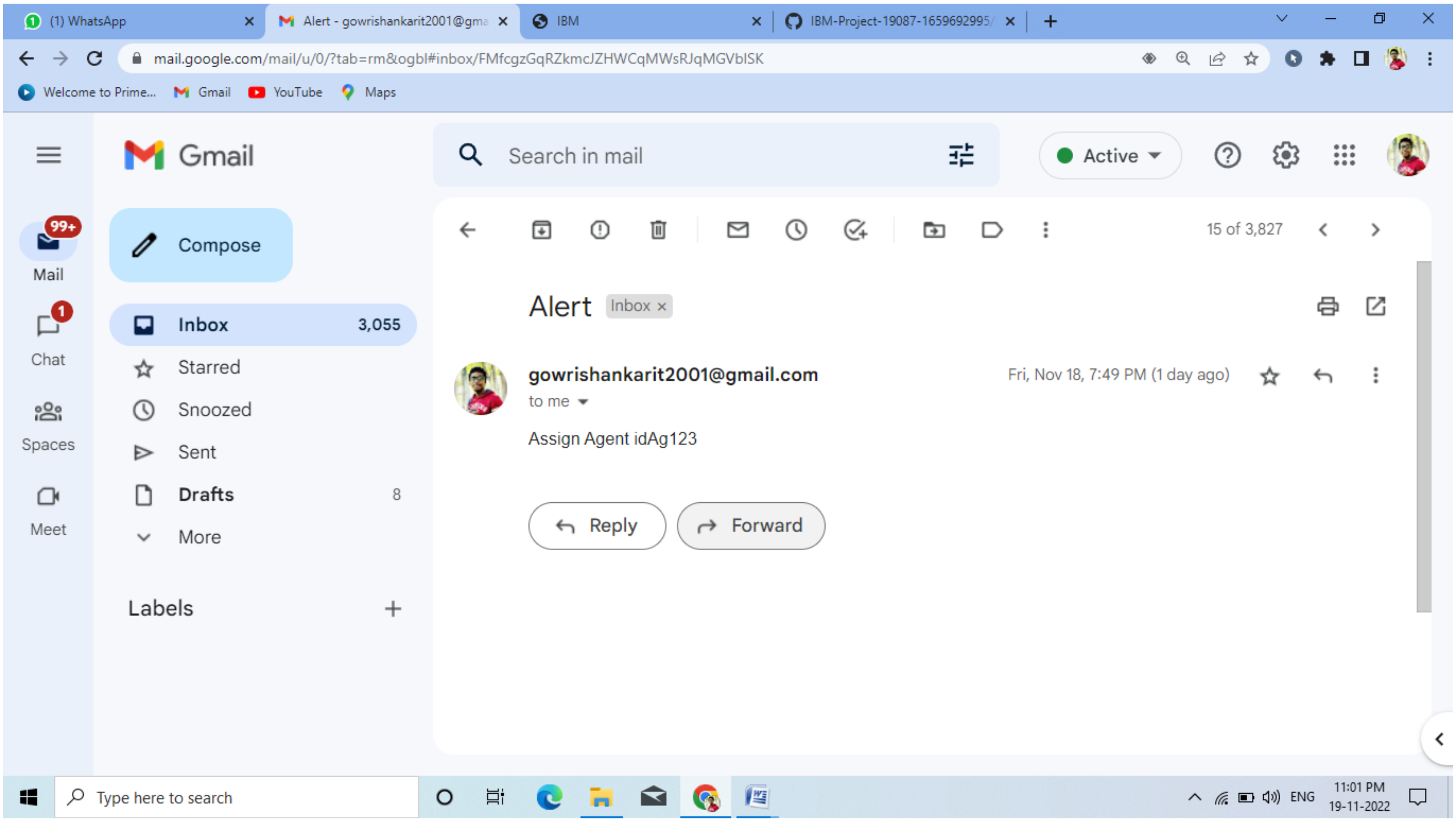
Customer Care

Complaint Action

Complaint Info: Solved



Action Info Saved Send Notication



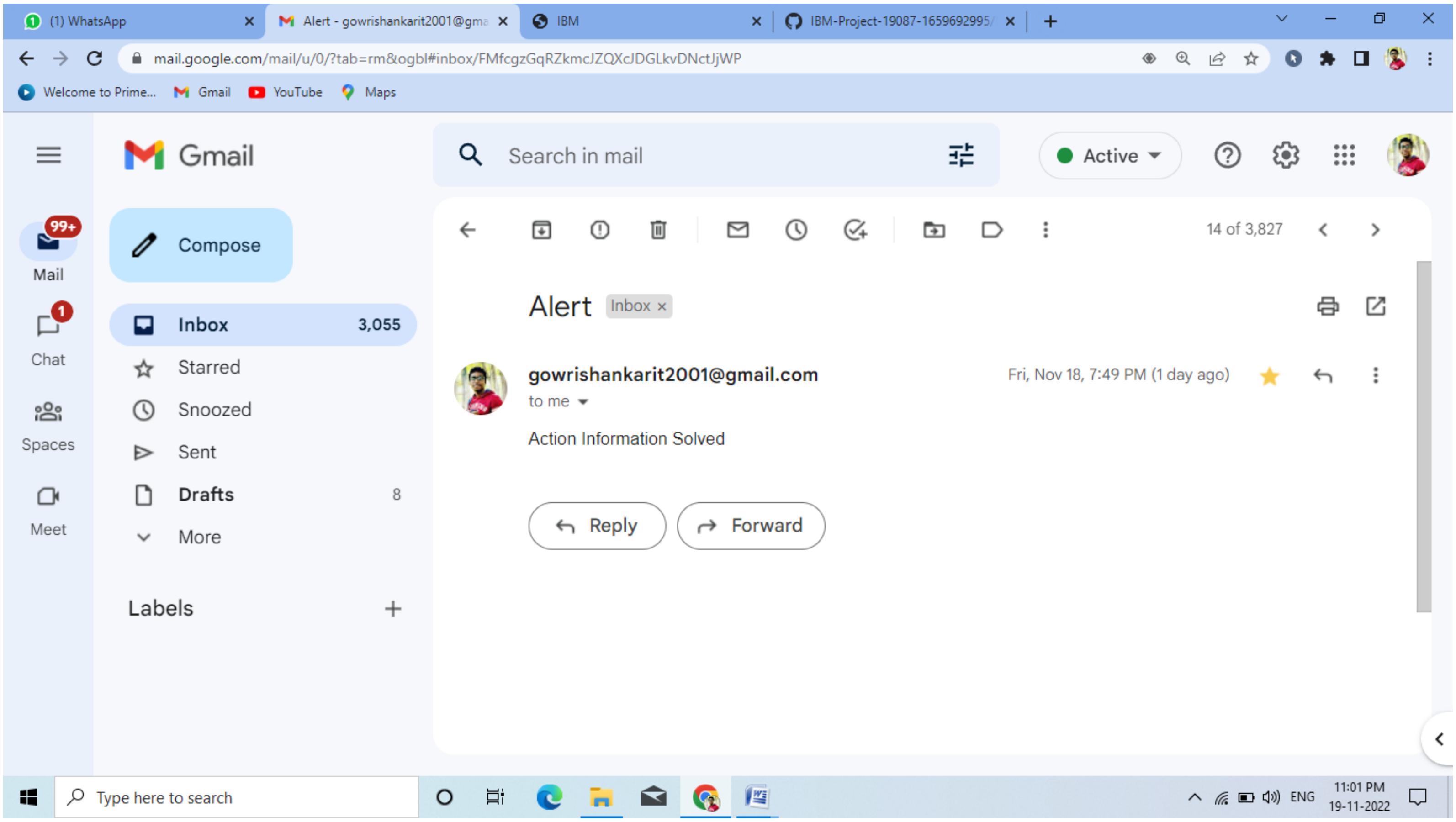
Alert Inbox ×

gowrishankarit2001@gmail.com

to me

Fri, Nov 18, 7:49 PM (1 day ago)

Assign Agent idAg123



Alert Inbox ×

gowrishankarit2001@gmail.com

to me

Fri, Nov 18, 7:49 PM (1 day ago)

Action Information Solved

GITHUB&PROJECTDEMOLINK

GITHUBID=https://github.com/IBM-EPBL/IBM-Project-19087-1659692995

VideoDemoLink=https://drive.google.com/file/d/1xsysUKE0pdskXyaYLJ7Z_1Mal_mml6f/view?usp=drivesdk