

CUSTOMER CARE REGISTRY

Team ID: PNT2022TMID50578

A PROJECT REPORT

Submitted By

- 1. SUGUNA DEVI K**
- 2. SAKTHI PRIYA C**
- 3. RAMA CHITRA T**
- 4. THANGA RAMANA P**

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

**SCAD COLLEGE OF ENGINEERING AND TECHNOLOGY
CHERANMAHADEVI**

ANNA UNIVERSITY: CHENNAI 600 025

NOV / DEC 2022

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	INTRODUCTION	
	1.1 Project Overview	
	1.2 Purpose	
2	LITERATURE SURVEY	
	2.1 Existing Problem	
	2.2 References	
	2.3 Problem Statement Definition	
3	IDEATION & PROPOSED SOLUTION	
	3.1 Empathy Map Canvas	
	3.2 Ideation & Brainstorming	
	3.3 Proposed Solution	
	3.4 Problem Solution fit	
4	REQUIREMENT ANALYSIS	
	4.1 Functional requirement	
	4.2 Non-Functional requirement	
5	PROJECT DESIGN	
	5.1 Data Flow Diagrams	
	5.2 Solution & Technical Architecture	
	5.3 User Stories	
6	PROJECT PLANNING & SCHEDULING	
	6.1 Sprint Planning & Estimation	
	6.2 Sprint Delivery Schedule	
	6.3 Reports from Jira	
7	CODING & SOLUTIONING	
	7.1 Feature 1	
	7.2 Feature 2	
8	TESTING	
	8.1 Test Cases	
	8.2 User Acceptance Testing	
9	RESULTS	

	9.1 Performance Metrics	
10	ADVANTAGES & DISADVANTAGES	
11	CONCLUSION	
12	FUTURE SCOPE	
13	APPENDIX	
	13.1 Source Code	
	13.2 GitHub & Project Demo Link	

1. INTRODUCTION

1.1 Project Overview

Customer Care Registry

Category: Cloud App Development.

Short Description: This Application has been developed to help the customer in processing their complaints. The customers can raise the ticket with a detailed description of the issue. An Agent will be assigned to the Customer to solve the problem. Whenever the agent is assigned to a customer, they will be notified with an email alert. Customers can view the status of the ticket till the service is provided.

Admin: The main role and responsibility of the admin are to take care of the whole process. Starting from Admin login followed by the agent creation and assigning the customer's complaints. Finally, He will be able to track the work assigned to the agent and a notification will be sent to the customer.

User: They can register for an account. After the login, they can create a complaint with a description of the problem they are facing. Each user will be assigned agent. They can view the status of their complaint.

1.2 PURPOSE

- The purpose of the whole project is to:
 - Provide a common platform to the customers to clarify their queries
 - Having expert agents in the platform for better answering

- Customer's tickets (queries) are answered quickly by the agents
- Customers and Agents can chat with one another for better understanding
- While doing so, the former asks questions
- Later, answers those questions as quickly and as legitimately as possible
- Customers can raise as many tickets as they want
- Customers and Agents can also submit their feedback to the Admin, for the betterment of the platform

2. LITERATURE SURVEY

2.1 Existing Problem

- Reviews and rating in the e-commerce websites are not reliable
- Even more so, they are often been given by the manufactures themselves
- Reviews are not from the authentic individuals
- After buying the products, I am left with no option to clear my doubts
- There is no common platform available to us, the customers, to have our doubts cleared
- If it is existing, we are not getting fast replies. By the time, the reply comes, the issue might have been cleared or of not worth of being cleared to the customers

2.2 References

<https://www.helpdesk.com/>

<https://freshdesk.com/helpdesk-software>

<https://freshdesk.com/resources/case-study/hamleys>

<https://pulsedesk.com/>

<https://www.redpoints.com/blog/amazon-fake-reviews/>

2.3 Problem Statement Definition

I am Surya and I am a regular customer in famous e-commerce websites like Amazon, Flipkart. I order regularly. The problem I have is that in most times, I don't have any reliable sources to clear my doubts in some of the products I buy.

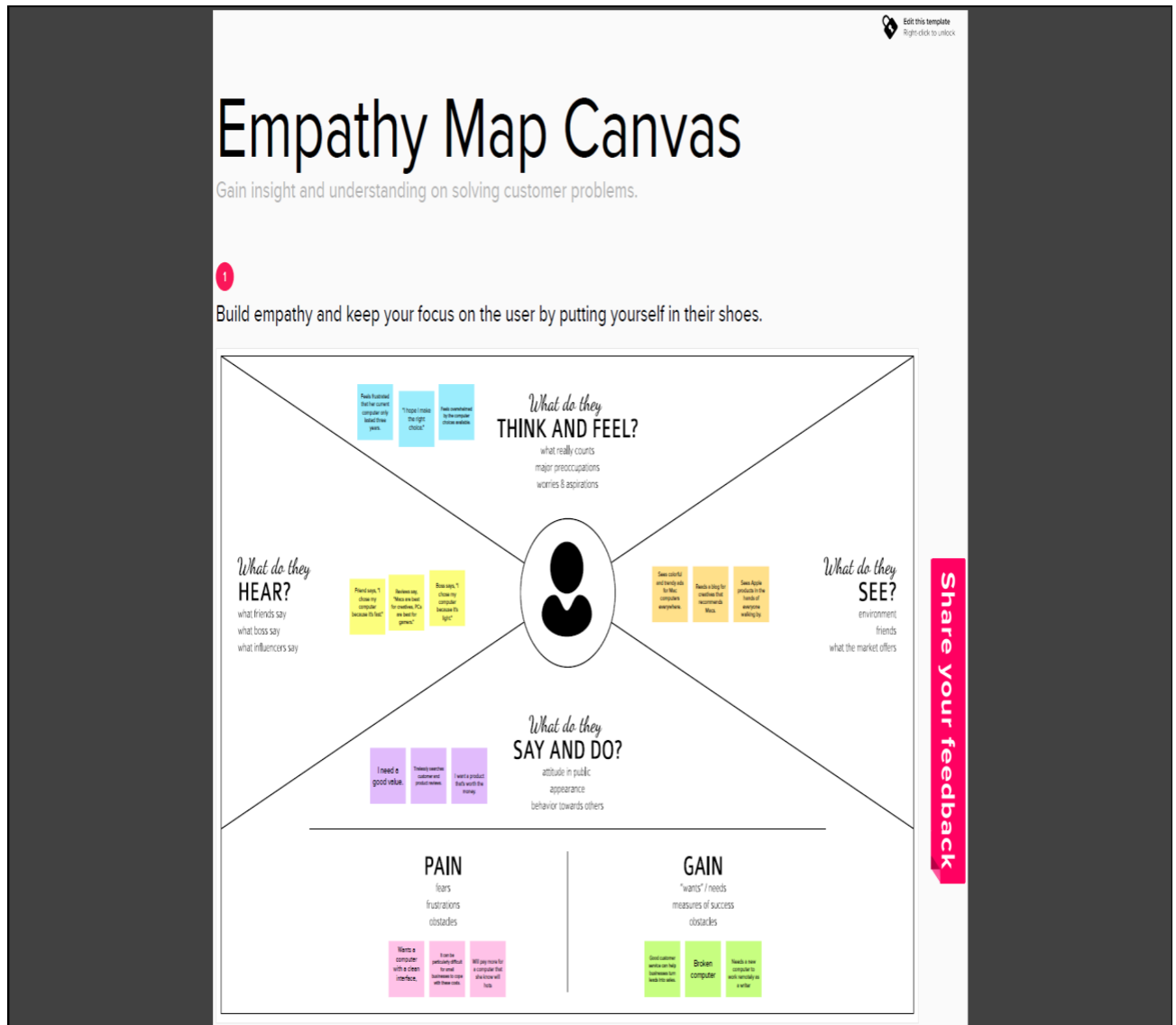
There are reviews and customer ratings in those websites, but somehow, I don't feel they are authentic and real. It would make my world if those replies were from a real expert, and I could clarify all my doubts in a single platform. Of course, I would need instant replies from a real expert who knows about the products I am asking for.

S.NO	PAPER	AUTHOR	YEAR	METHOD AND ALGORITHM	ACCURACY/ PRECISION
1	The Role Of Customer Service Through Customer Relationship Management (CRM) To Increase Customer Loyalty And Good Image.	Gede Juanamasta	2019	This study aims to determine the role how customer service through Customer Relationship Management (CRM) to improve customer loyalty and good image . This research method using this qualitative study, researchers used a paradigm Non Positivism / Naturalistic / Interpretative. Interpretative paradigm aims to understand the meaning of behavior, symbols, and phenomena by using sampling purposive sampling. Data Collection Techniques using interviews, documentation, observation. The results of this study that the role of customer service through Customer Relationship Management (CRM) to improve customer loyalty and good image .	85%
2	customer services and their role for industrial small and medium companies.	Lucie Kanovska	2009	The aim of this paper is to present the problems of customer services and their important role for small and medium companies from the theoretical view and also selected results of research held in industrial SME's. Customer services are kind of services being provided by companies to their products. Customer services can be found in all economic spheres, such as in primary sphere, as well as in secondary and tertiary ones	80%

3	The Relationship between Customer Relationship Management and Customer Satisfaction with Services Received.	Mohammad Heydari, Hadiseh Abaszadeh, Habibollah Danai	2015	Today, with the advancement of information technology in the organization of new systems of production that can reduce internal costs, better interaction with the environment and ultimately help to make a profit. Customer satisfaction is one of these tools	85.57%
4	The Customer Service Management Process	A. Michael Knemeyer, Douglas M. Lambert and Sebastián J. GarcíaDastugue	2004	The aim of this paper is to present the problems of customer services and their important role for small and medium companies from the theoretical view and also selected results of research held in industrial SME's. Customer services are kind of services being provided by companies to their products. Customer services can be found in all economic spheres, such as in primary sphere, as well as in secondary and tertiary	82.88%

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step 2: Brainstorm, Idea Listing, and Grouping

1 Prioritize

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

20 minutes

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 actions

- 1 **Shape the mural**
Share a view link to the mural with collaborators to keep them in the loop about the outcomes of the session.
- 2 **Export the mural**
Export a copy of the mural as PNG or PDF to share to email, include in slides, or save to 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, behaviors, and attitudes 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 Proposed Solution

S. No	Parameter	Description
1	Problem Statement (Problem to be solved)	<ul style="list-style-type: none"> I am Surya and I am a regular customer in famous ecommerce websites like Amazon, Flipkart. I order regularly. The problem I have is that in most times, I don't have any reliable sources to clear my doubts in some of the products I buy. There are reviews and

		customer ratings in those websites, but somehow, I don't feel they are authentic and real. It would make my world if those replies are from a real expert and I could clarify all my doubts in a single platform. Of course, I would need instant replies from a real expert who knows about the products I am asking for.
2	Idea / Solution Description	<ul style="list-style-type: none"> • Creating a Customer Care Registry, where the customers can raise their queries in form of tickets. An agent will be assigned to them for replying/clarify their issue.
3	Novelty / Uniqueness	<ul style="list-style-type: none"> • The agents are experts in the product domain and they will communicate well with the customers.
4	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> • Customers will be satisfied with the instant and valid replies. Also, it creates a doubtless society, that boosts sales.
5	Business Model (Revenue Model)	<ul style="list-style-type: none"> • Customers can be charged a minimal amount based on the number of queries (tickets) they can rise in a said period of time.
6	Scalability of the Solution	<ul style="list-style-type: none"> • May be in the future, may be a cross-platform mobile application may be developed, making this customer care registry much more accessible to the users.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

- A functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs.
- It specifies “what should the software system do?”
- Defined at a component level
- Usually easy to define
- Helps you verify the functionality of the software

Following are the functional requirements of the proposed solution.

FR	Functional Requirements (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Signup form (customer)
FR-2	Forgot Password	Resetting the password by sending an OTP to user's mail (customer, agent, admin)
FR-3	User Login	Login through Login form (customer, agent, user)
FR-4	Agent creation (admin)	Create an agent profile with username, email and password
FR-5	Dashboard (customer)	Show all the tickets raised by the customer
FR-6	Dashboard (agent)	Show all the tickets assigned to the agent by admin
FR-7	Dashboard (Admin)	Show all the tickets raised in the entire system
FR-8	Ticket creation (customer)	Customer can raise a new ticket with the detailed description of his/her query
FR-9	Assign agent (admin)	Assigning an agent for the created ticket
FR-10	Ticket details (customer)	1. Showing the actual query, status, assigned agent details 2. Status of the ticket

FR-11	Address Column	Agent clarifies the doubts of the customer
-------	----------------	--

4.3. Non -Functional Requirements:

- A non-functional requirement defines the quality attribute of a software system
- It places constraint on “How should the software system fulfil the functional requirements?”
- It is not mandatory
- Applied to system as a whole
- Usually more difficult to define
- Helps you verify the performance of the software

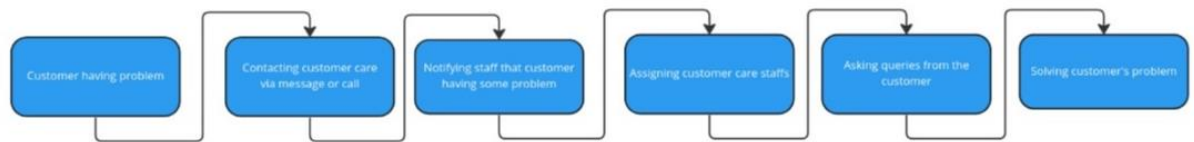
Following are the non-functional requirements of the proposed solution.

NFR	Non-Functional Requirements	Description
NFR-1	Usability	Customers can use the application in almost all the web browsers. Application is with good looking and detailed UI, which makes it more friendly to use.
NFR-2	Security	Customers are asked to create an account for themselves using their email which is protected with an 8 character-long password, making it more secure.
NFR-3	Reliability	Customers can raise their queries and will be replied with a valid reply, as soon as possible, making the application even more reliable and trust-worthy
NFR-4	Performance	Customers will have a smooth experience while using the application, as it is simple and is well optimised.

NFR-5	Availability	Application is available 24/7 as it is hosted on IBM Cloud
NFR-6	Scalability	In future, may be cross-platform mobile applications can be developed as the user base grows.

5. PROJECT DESIGN

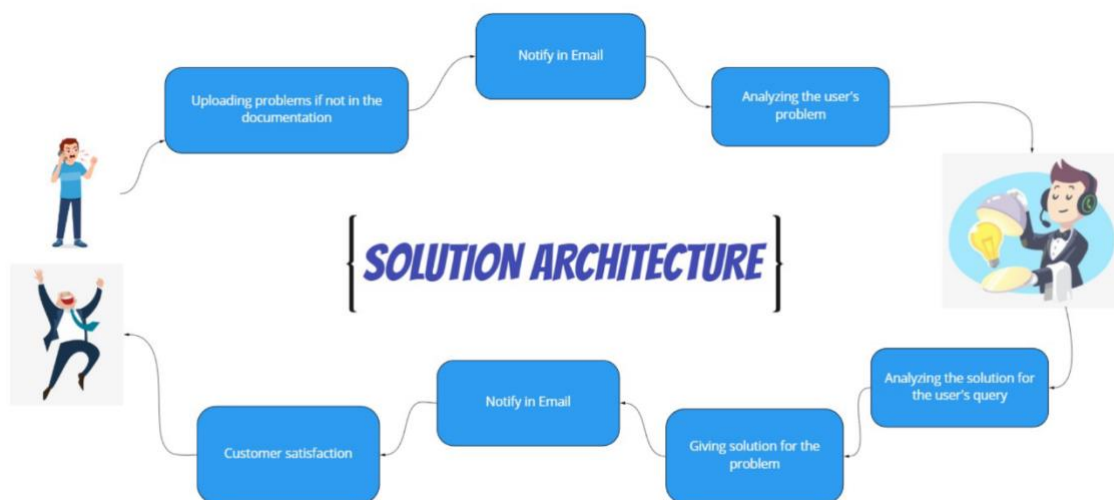
5.1 Data Flow Diagrams



miro

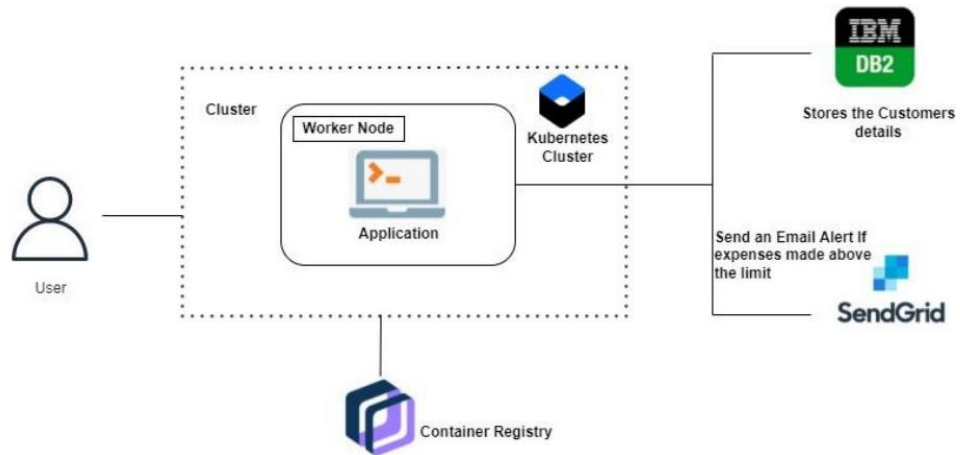
5.2 Solution & Technical Architecture

Solution Architecture:



miro

Technical Architecture:



5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user, if a problem arises, then I post or contact the customer care interface.		High	
		USN-7	As a user, I wait for a solution for my problem from the customer care staff.		High	
		USN-8	As a user, after I get my solution, I will solve the problem.		High	

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Suguna Devi
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Thanga Ramana
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Medium	Sakthi Priya
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Rama Chitra
Sprint-1		USN-5	As a user, I can register for the application through mobile number	2	Medium	Thanga Ramana
Sprint-1		USN-6	As a user, I will receive confirmation code number once I have registered for the application	1	High	Rama Chitra
Sprint-2	Login	USN-7	As a user, I can log into the application by entering email & password	1	High	Sakthi Priya
Sprint-2		USN-8	As a user, I can log into the application by entering mobile number & password	1	High	Suguna Devi
Sprint-3	Dashboard(User)	USN-9	As a user, I can easily able to access the website with the help of guide tour	2	Medium	Suguna Devi

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3		USN-10	As a user, I can view the progress in my dashboard(whether I have been allotted with a live agent or not, whether my complaints have been resolved or In a pending state.	2	High	Thanga Ramana
Sprint-3		USN-11	As a user, I can view my live agent and his activities towards my queries.	2	Medium	Sakthi Priya
Sprint-4	Dashboard(Agent)	USN-12	As a agent, I can make user's queries into actions and notify them with every updates	1	Medium	Thanga Ramana
		USN-13	As a agent, I can able to view the works assigned by admin	1	High	Rama Chitra
Sprint-4	Dashboard(Admin)	USN-14	As a administrator, I can categorize the users based on their needs and assign respective agents to the users	2	High	Sakthi Priya
		USN-15	As a administrator, I can provide security to all the data by blocking different access and maintain the privacy	2	Medium	Rama Chitra

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	4	6 Days	31 Oct 2022	05 Nov 2022	4	05 Nov 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022
Sprint-4	6	6 Days	14 Nov 2022	19 Nov 2022	6	19 Nov 2022

6.3 Reports from Jira

Sprint 1 – Burndown Chart

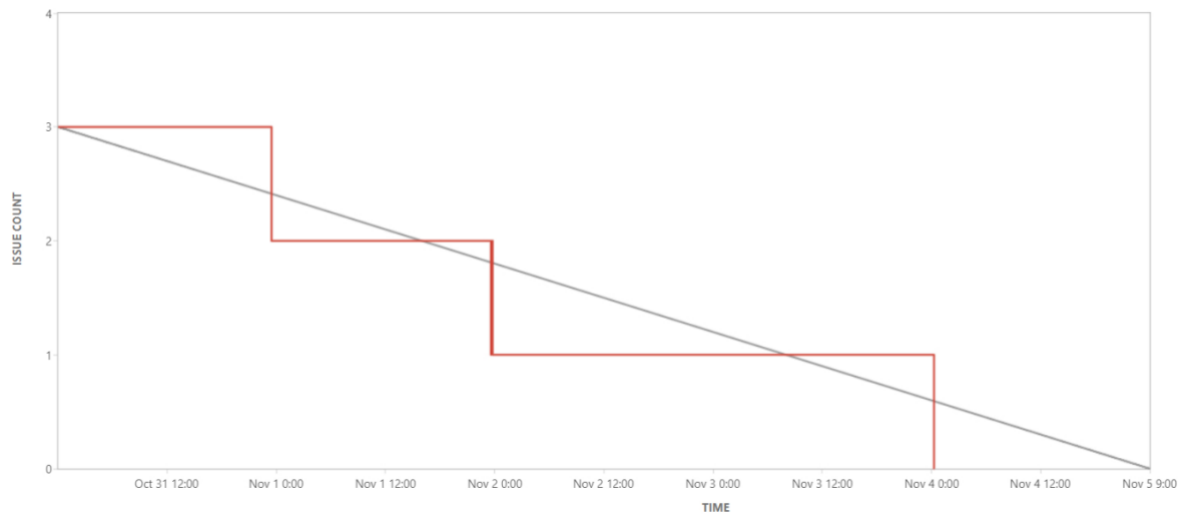
Burndown Chart



Sprint 2 – Burndown Chart

Burndown Chart

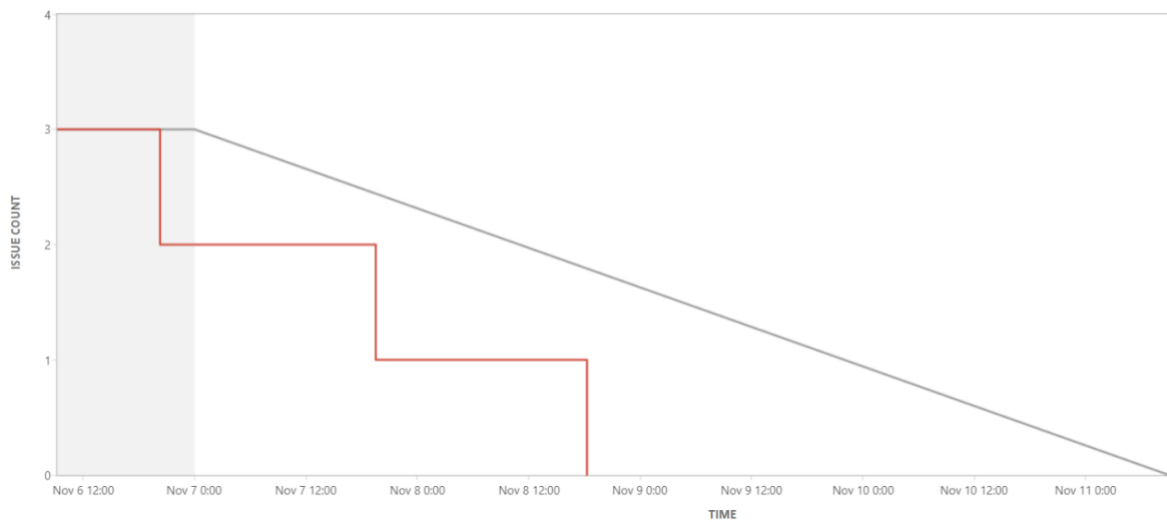
CCR Sprint 2 ▾ Issue Count ▾ ? [How to read this chart](#)



Sprint 3 – Burndown Chart

Burndown Chart

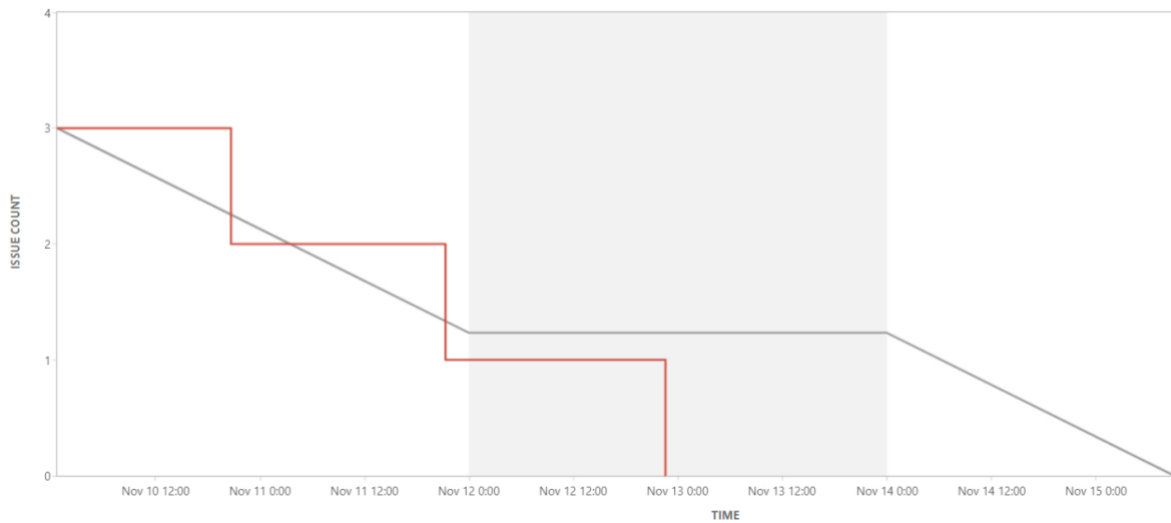
CCR Sprint 3 ▾ Issue Count ▾ ? [How to read this chart](#)



Sprint 4 – Burndown Chart

Burndown Chart

CCR Sprint 4 ▾ Issue Count ▾ ? [How to read this chart](#)



7. CODING & SOLUTIONING

7.1 Feature 1

- **Feature 1:** Having expert agents in the platform for better answering
- **Feature 2:** Customers and Agents can chat with one another for better understanding
- **Feature 3:** Customers can raise as many tickets as they want
- **Feature 4:** Customers and Agents can also submit their feedback to the Admin, for the betterment of the platform

7.2 Feature 2

- Customer can clarify their doubts just by creating a new ticket
- Customer gets replies as soon as possible
- Customers are provided with a unique account, to which the latter can login at any time Very minimal account creation process
- Customers are given clear notifications through email, of all the processes related to login, ticket creation etc.,

8. TESTING

8.1 TEST CASES

The test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case designing includes preconditions, case name, input conditions, and expected result. A test case is a first level action and derived from test scenarios. Test case gives detailed information about testing strategy, testing process, preconditions, and expected output. These are executed during the testing process to check whether the software application is performing the task for that it was developed or not. Test case helps the tester in defect reporting by linking defect with test case ID. Detailed test case documentation works as a full proof guard for the testing team because if developer missed something, then it can be caught during execution of these full-proof test cases. To write the test case, we must have the requirements to derive the inputs, and the test scenarios must be written so that we do not miss out on any features for testing.

Then we should have the test case template to maintain the uniformity, or every test engineer follows the same approach to prepare the test document.

8.2 USER ACCEPTANCE TESTING

1. Purpose of the document

The purpose of this document is to briefly explain the test coverage and open issues of the [Customer Care Registry] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	5	5	24
Duplicate	2	0	2	0	4
External	5	3	2	1	11
Fixed	15	5	5	10	35
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	32	17	17	18	84

3. Test case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10

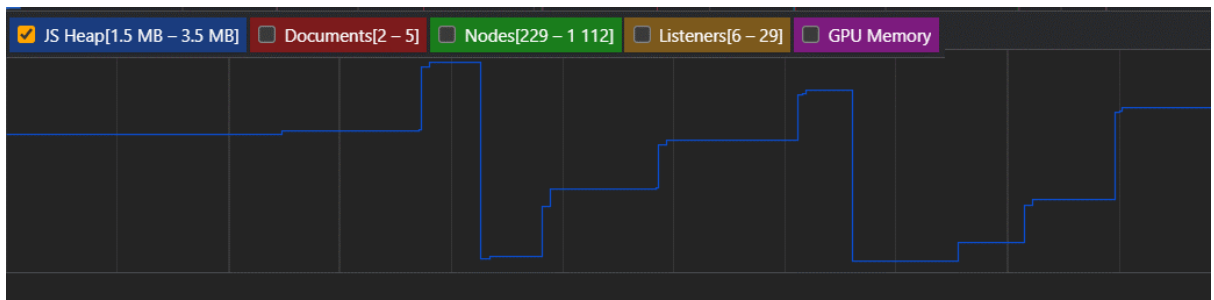
Client Application	40	0	0	40
Security	5	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	10	0	0	10
Final Report Output	4	0	0	4
Version Control	4	0	0	4

9. RESULTS

9.1 Performance Metrics

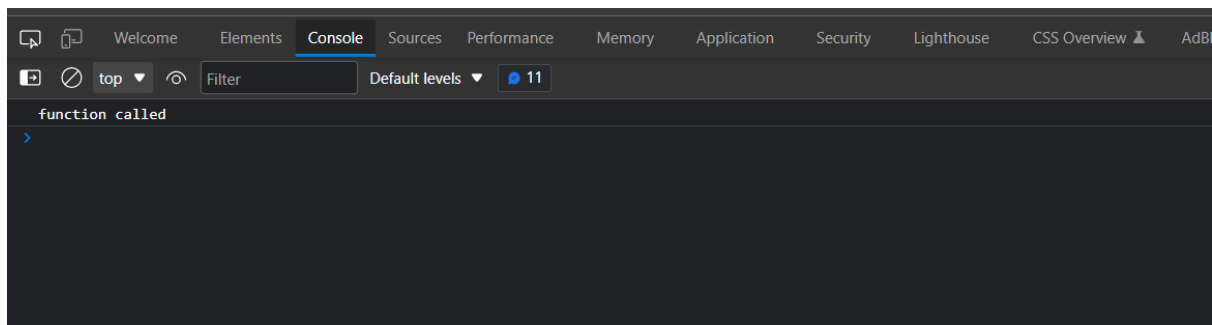
CPU usage:

- Since all the operations run using Flask is in server-side, the client (browser) need not worry about the CPU usage. Just rendering the page, static contents take place in the client-side.
- Memory for client-side functions (Javascript) is allocated using heap. It can be either increased based upon the requirement or removed from the heap.



Errors:

- Since all the backend functions are done using flask, any exceptions / errors rising are well-handled. Though they appear, user's interaction with the site is not affected in any way



Latency and Response time:

It takes less than a second to load a page in the client. From this it is evident that there is low latency

11 requests 238 kB transferred 285 kB resources Finish: 892 ms DOMContentLoaded: 810 ms Load: 905 ms

10. ADVANTAGES AND DISADVANTAGES

Advantages:

Customers can clarify their doubts just by creating a new ticket

✓ Customer gets replies as soon as possible

- ✓ Not only the replies are faster, the replies are more authentic and practical
- ✓ Customers are provided with a unique account, to which the latter can login at any time
- ✓ Very minimal account creation process
- ✓ Customers can raise as many tickets as they want
- ✓ Application is very simple to use, with well-known UI elements
- ✓ Customers are given clear notifications through email, of all the processes related to login, ticket creation etc.,
- ✓ Customers' feedbacks are always listened
- ✓ Free of cost.

Disadvantages:

- × Only web application is available right now (as of writing)
- × UI is not so attractive, it's just simple looking
- × No automated replies
- × No SMS alerts
- × Supports only text messages while chatting with the Agent
- × No tap to reply feature
- × No login alerts
- × Cannot update the mobile number
- × Account cannot be deleted, once created
- × Customers cannot give feedback to the agent for clarifying the queries

11. CONCLUSION

Thus, there are many customer service applications available on the internet. Noting down the structural components of those applications and we built a customer care registry application. It will be a web application build with Flask (Python micro-web framework), HTML, JavaScript. It will be a ticket-based customer service registry.

Customers can register into the application using their email, password, first name and last name. Then, they can login to the system, and raise as tickets as they want in the form of their tickets.

These tickets will be sent to the admin, for which an agent is assigned. Then, the assigned agent will have a one-to-one chat with the customer and the latter's queries will be clarified. It is also the responsibility of the admin, to create an agent.

12. FUTURE SCOPE

- ✓ Attracting and much more responsive UI throughout the application
- ✓ Releasing cross-platform mobile applications
- ✓ Incorporating automatic replies in the chat columns
- ✓ Deleting the account whenever customer wishes to
- ✓ Supporting multi-media in the chat columns
- ✓ Creating a community for our customers to interact with one another
- ✓ Call support
- ✓ Instant SMS alerts

13. APPENDIX

Flask:

- ✓ Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries
- ✓ It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions

JavaScript:

- ✓ JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS
- ✓ As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries

IBM Cloud:

- ✓ IBM cloud computing is a set of cloud computing services for business offered by the information technology company IBM

Kubernetes:

- ✓ Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management

Docker:

- ✓ Docker is a set of platforms as a service product that use OS-level virtualization to deliver software in packages called containers

13.1 Source Code from flask

```
import Flask, render_template, url_for
from markupsafe import escape

app=Flask(__name__,template_folder='templates')

@app.route("/")
def index():
    return render_template('index.html')

@app.route("/index")
def home():
```

```
return render_template('index.html')
```

```
@app.route('/blog')
```

```
def blog():
```

```
    return render_template("blog.html")
```

```
@app.route('/about')
```

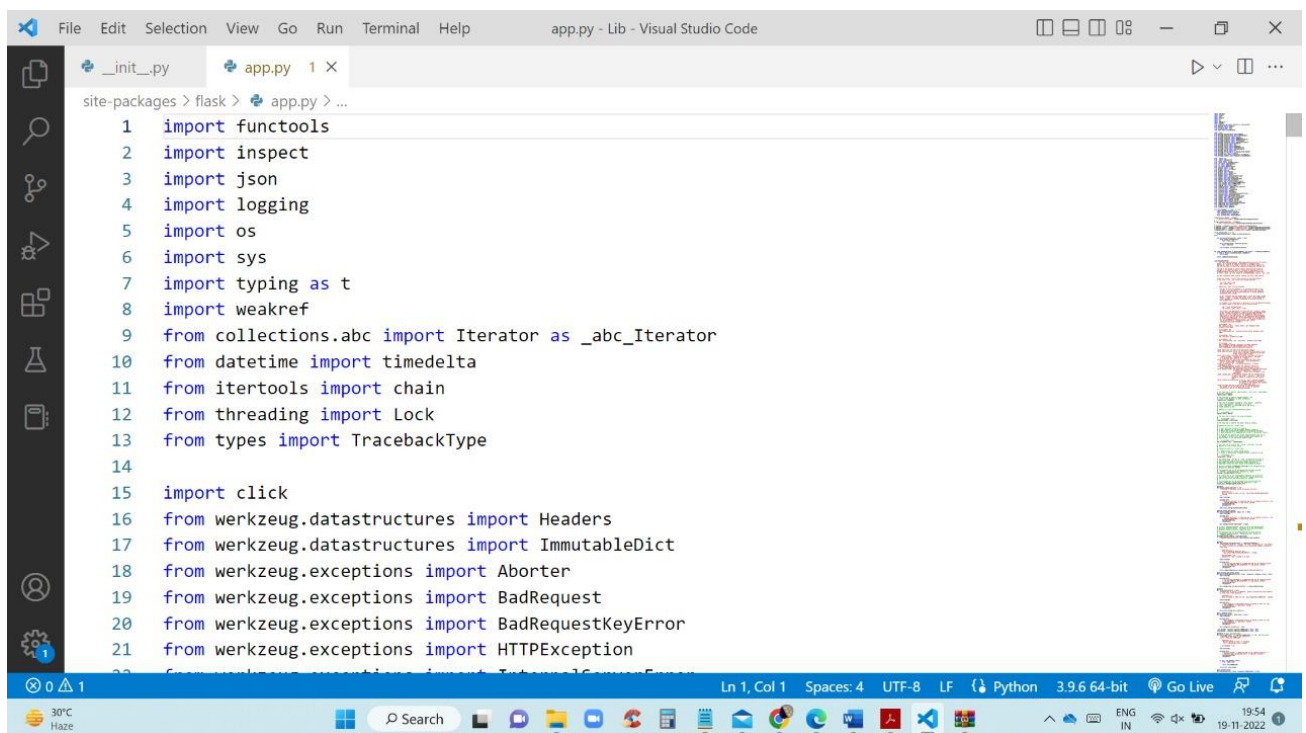
```
def about():
```

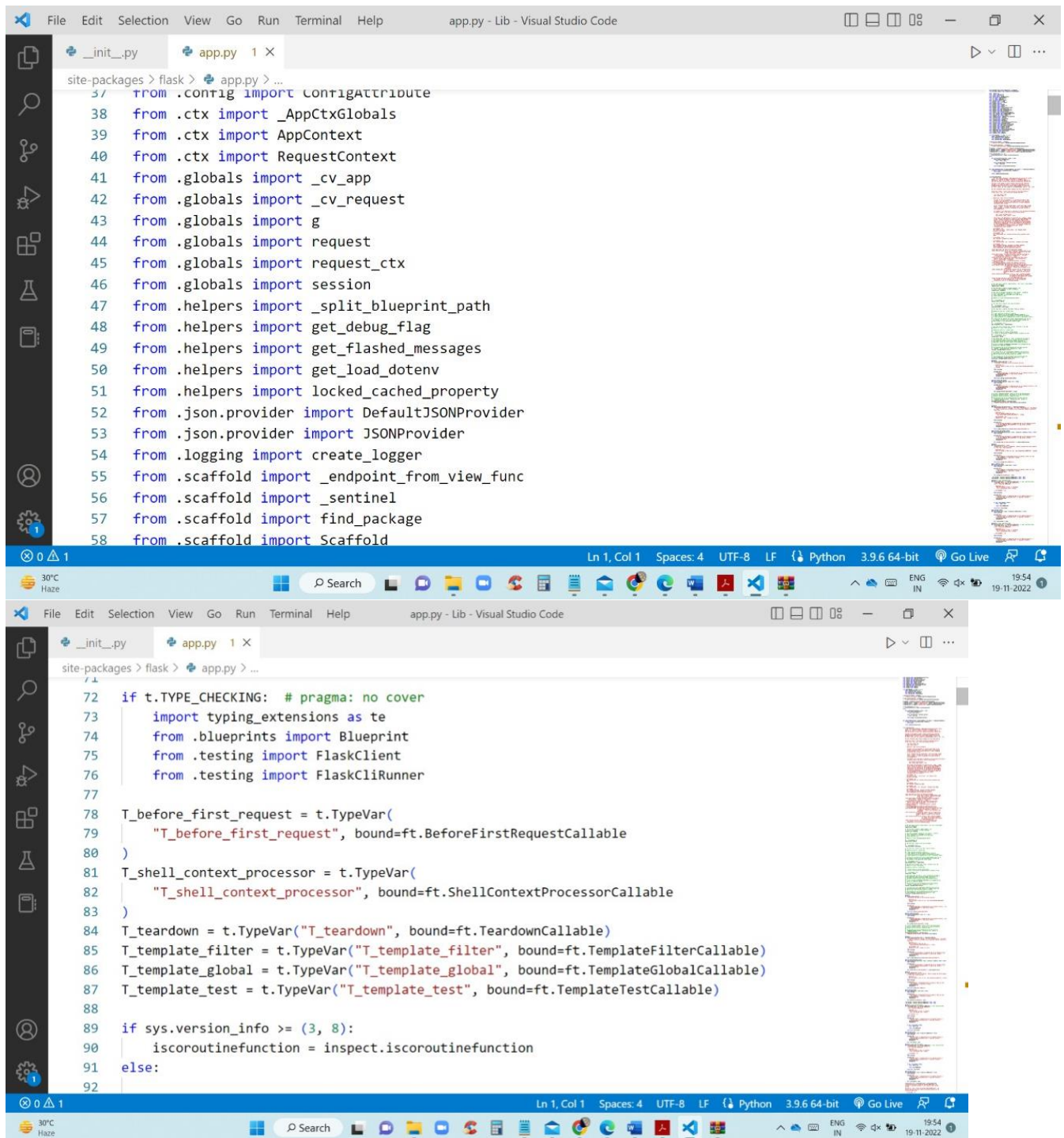
```
    return render_template("Aboutus.html")
```

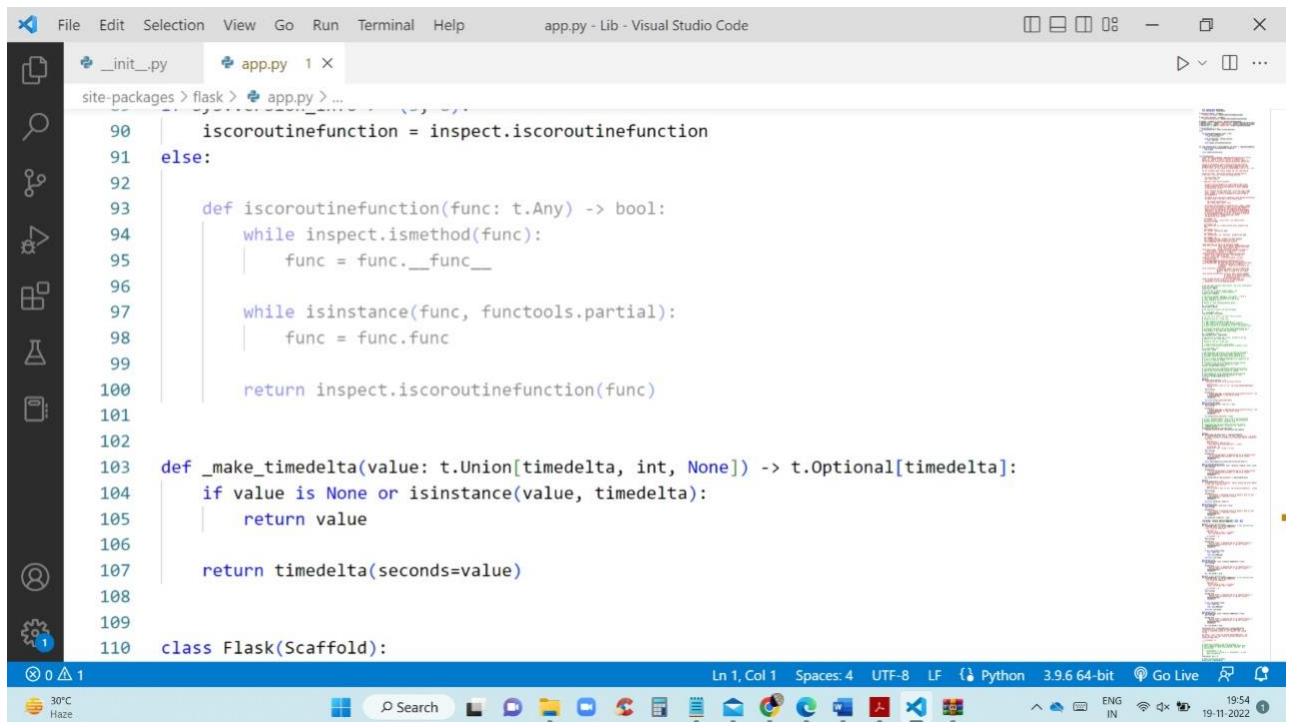
```
@app.route('/sign-In')
```

```
def sign():
```

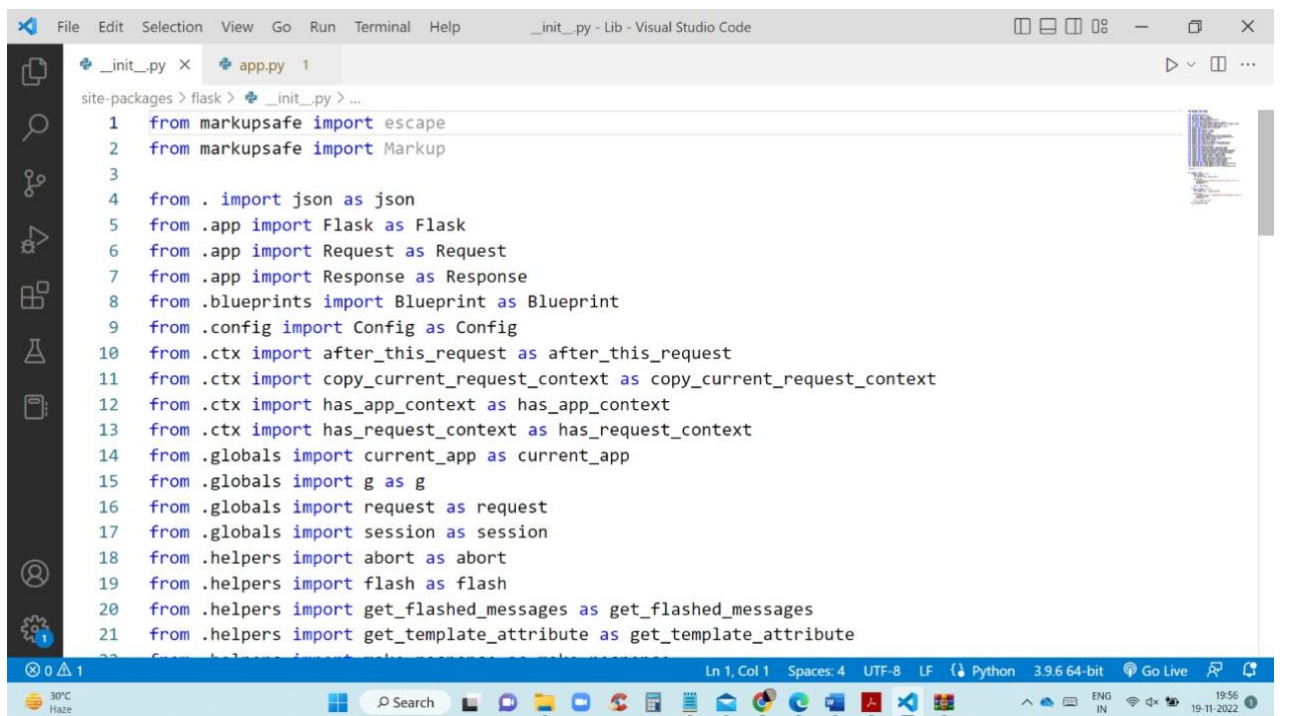
```
    return render_template("Register.html")
```



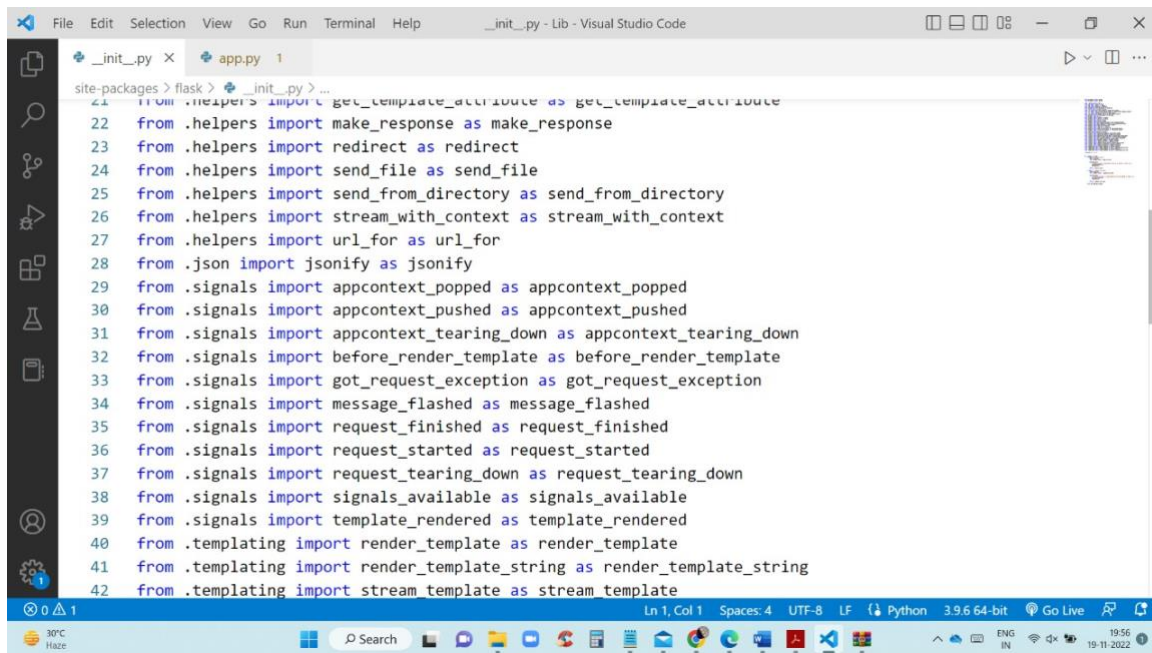




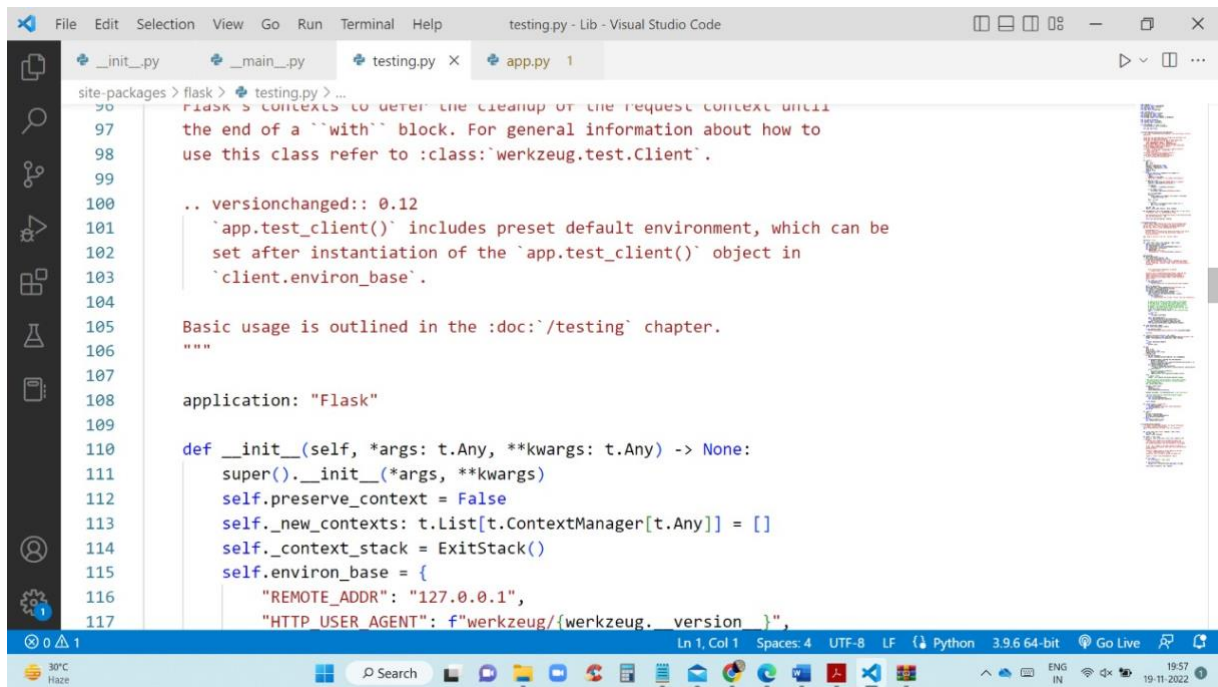
```
90     iscoroutinefunction = inspect.iscoroutinefunction
91 else:
92
93     def iscoroutinefunction(func: t.Any) -> bool:
94         while inspect.ismethod(func):
95             func = func.__func__
96
97         while isinstance(func, functools.partial):
98             func = func.func
99
100        return inspect.iscoroutinefunction(func)
101
102
103 def _make_timedelta(value: t.Union[timedelta, int, None]) -> t.Optional[timedelta]:
104     if value is None or isinstance(value, timedelta):
105         return value
106
107     return timedelta(seconds=value)
108
109
110 class Flask(Scaffold):
```



```
1 from markupsafe import escape
2 from markupsafe import Markup
3
4 from . import json as json
5 from .app import Flask as Flask
6 from .app import Request as Request
7 from .app import Response as Response
8 from .blueprints import Blueprint as Blueprint
9 from .config import Config as Config
10 from .ctx import after_this_request as after_this_request
11 from .ctx import copy_current_request_context as copy_current_request_context
12 from .ctx import has_app_context as has_app_context
13 from .ctx import has_request_context as has_request_context
14 from .globals import current_app as current_app
15 from .globals import g as g
16 from .globals import request as request
17 from .globals import session as session
18 from .helpers import abort as abort
19 from .helpers import flash as flash
20 from .helpers import get_flashed_messages as get_flashed_messages
21 from .helpers import get_template_attribute as get_template_attribute
```

```
site-packages > flask > _init_.py > ...
22 from .helpers import get_template_attribute as get_template_attribute
23 from .helpers import make_response as make_response
24 from .helpers import redirect as redirect
25 from .helpers import send_file as send_file
26 from .helpers import send_from_directory as send_from_directory
27 from .helpers import stream_with_context as stream_with_context
28 from .helpers import url_for as url_for
29 from .json import jsonify as jsonify
30 from .signals import appcontext_popped as appcontext_popped
31 from .signals import appcontext_pushed as appcontext_pushed
32 from .signals import appcontext_tearing_down as appcontext_tearing_down
33 from .signals import before_render_template as before_render_template
34 from .signals import got_request_exception as got_request_exception
35 from .signals import message_flashed as message_flashed
36 from .signals import request_finished as request_finished
37 from .signals import request_started as request_started
38 from .signals import request_tearing_down as request_tearing_down
39 from .signals import signals_available as signals_available
40 from .templating import render_template as render_template
41 from .templating import render_template_string as render_template_string
42 from .templating import stream_template as stream_template
```



```
site-packages > flask > testing.py > ...
97 flask's contexts to defer the cleanup of the request context until
98 the end of a ``with`` block. For general information about how to
99 use this class refer to :class:`werkzeug.test.Client`.
100
101 .. versionchanged:: 0.12
102     `app.test_client()` includes preset default environment, which can be
103     set after instantiation of the `app.test_client()` object in
104     `client.environ_base`.
105
106 Basic usage is outlined in the :doc:`testing` chapter.
107 """
108
109 application: "Flask"
110
111 def __init__(self, *args: t.Any, **kwargs: t.Any) -> None:
112     super().__init__(*args, **kwargs)
113     self.preserve_context = False
114     self.new_contexts: t.List[t.ContextManager[t.Any]] = []
115     self._context_stack = ExitStack()
116     self.environ_base = {
117         "REMOTE_ADDR": "127.0.0.1",
118         "HTTP_USER_AGENT": f"werkzeug/{werkzeug.__version__}",
119     }
```

The screenshot shows the Visual Studio Code editor with the file `testing.py` open. The editor is displaying the `test_request_context` method of the `Flask` class. The code is as follows:

```
117         "HTTP_USER_AGENT": f"werkzeug/{werkzeug.__version__}",
118     }
119
120     @contextmanager
121     def session_transaction(
122         self, *args: t.Any, **kwargs: t.Any
123     ) -> t.Generator[SessionMixin, None, None]:
124         """When used in combination with a ``with`` statement this opens a
125         session transaction. This can be used to modify the session that
126         the test client uses. Once the ``with`` block is left the session is
127         stored back.
128
129         ::
130
131             with client.session_transaction() as session:
132                 session['value'] = 42
133
134         Internally this is implemented by going through a temporary test
135         request context and since session handling could depend on
136         request variables this function accepts the same arguments as
137         :meth:`~flask.Flask.test_request_context` which are directly
```

The status bar at the bottom indicates the file is at line 1, column 1, with 4 spaces, using UTF-8 encoding and LF line endings. The Python version is 3.9.6 64-bit.

The screenshot shows the Visual Studio Code editor with the file `testing.py` open. The editor is displaying the `open` method of the `Flask` class. The code is as follows:

```
215         # request is None
216         request = self._request_from_builder_args(args, kwargs)
217
218         # Pop any previously preserved contexts. This prevents contexts
219         # from being preserved across redirects or multiple requests
220         # within a single block.
221         self._context_stack.close()
222
223         response = super().open(
224             request,
225             buffered=buffered,
226             follow_redirects=follow_redirects,
227         )
228         response.json_module = self.application.json # type: ignore[misc]
229
230         # Re-push contexts that were preserved during the request.
231         while self._new_contexts:
232             cm = self._new_contexts.pop()
233             self._context_stack.enter_context(cm)
234
235         return response
```

The status bar at the bottom indicates the file is at line 1, column 1, with 4 spaces, using UTF-8 encoding and LF line endings. The Python version is 3.9.6 64-bit.

```
site-packages > flask > testing.py > ...
235         return response
236
237     def __enter__(self) -> "FlaskClient":
238         if self.preserve_context:
239             raise RuntimeError("Cannot nest client invocations")
240         self.preserve_context = True
241         return self
242
243     def __exit__(
244         self,
245         exc_type: t.Optional[type],
246         exc_value: t.Optional[BaseException],
247         tb: t.Optional[TracebackType],
248     ) -> None:
249         self.preserve_context = False
250         self._context_stack.close()
251
252
253 class FlaskCliRunner(CliRunner):
254     """A :class:`~click.testing.CliRunner` for testing a Flask app's
255     CLI commands. Typically created using
256     :meth:`~flask.Flask.test_cli_runner`. See :ref:`testing-cli`
```

```
269
270     If the ``obj`` argument is not given, passes an instance of
271     :class:`~flask.cli.ScriptInfo` that knows how to load the Flask
272     app being tested.
273
274     :param cli: Command object to invoke. Default is the app's
275                 :attr:`~flask.app.Flask.cli` group.
276     :param args: List of strings to invoke the command with.
277
278     :return: a :class:`~click.testing.Result` object.
279     """
280     if cli is None:
281         cli = self.app.cli # type: ignore
282
283     if "obj" not in kwargs:
284         kwargs["obj"] = ScriptInfo(create_app=lambda: self.app)
285
286     return super().invoke(cli, args, **kwargs)
287
```

13.2 GitHub & Project Demo Link

Complete Source: [IBM-Project-47942-1660803473/PROJECT DEVELOPMENT PHASE at main · IBM-EPBL/IBM-Project-47942-1660803473 \(github.com\)](https://github.com/IBM-EPBL/IBM-Project-47942-1660803473)

Repo Link: [IBM-EPBL/IBM-Project-47942-1660803473: Customer Care Registry \(github.com\)](https://github.com/IBM-EPBL/IBM-Project-47942-1660803473)