NALAIYA THIRAN PROJECT – EXPERIENTIAL PROJECT BASED LEARNING

PLASMA DONOR APPLICATION

TEAM ID: PNT2022TMID030002

BATCH: B4-4M6E

TEAM MEMBERS:

SHANMUGARAJ K – 727719EUIT141
TAMIL MARAN R – 727719EUIT168
TILAK YOGESHWAR A – 727719EUIT169
VASANTH S - 727719EUIT171

INDUSTRY MENTOR NAME: Ms. NAVYA

FACULTY MENTOR NAME: Mr. M. ARUNACHALAM

INTRODUCTION

1.1 Project Overview

During COVID-19 crisis the requirement for plasma increased drastically as there were no vaccination found. With plasma therapy the recovery rates were high but the donor count was very low. It was very important to get information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand. It can save time and help the users to track down the necessary information about the donors.

1.2 Purpose

The purpose of the problem is to solve the lack of plasma donors and register them to donate their plasma.They can register them and we can easily get their details.

LITERATURE SURVEY

2.1 Existing problem

As a user We face many difficulties in our daily file. In our daily life money is the most important portion and without it we cannot last one day on earth .Apart from this every single person affect by some of the diseases, and they need blood.In earlier stages blood was collected from donor

2.2 References

1. Yang J.-J., Li J., Mulder J., Wang Y., Chen S., Wu H., Wang Q., Pan H. Emerginginformation technologies for enhanced healthcare. *Comput. Ind.* 2015;69:3–11.doi:10.1016/j.compind.2015.01.012. [CrossRef] [GoogleScholar]
2. Cortada J.W., Gordon D., Lenihan B. *The Value of Analytics in Healthcare.* IBM Institute for Business Value; Armonk, NY, USA: 2012. Report No.: GBE03476-USEN-00. [Google Scholar]
3. Center for Medicare and Medicaid Services. [(accessed on 1 August 2017)]; Availableonline:

https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends- andReports /NationalHealthExpendData/NationalHealthAccountsHistorical.html

4. Berwick D.M., Hackbarth A.D. Eliminating waste in US health care. *J. Am.Med. Assoc.* 2012;307:1513–1516. doi: 10.1001/jama.2012.362. [PubMed] [CrossRef][Google Scholar]
5. Makary M.A., Daniel M. Medical error-the third leading cause of death in theUS. *Br. Med. J.* 2016;353:i2139. doi: 10.1136/bmj.i2139. [PubMed][CrossRef] [Google Scholar]

6. Prokosch H.-U., Ganslandt T. Perspectives for medical informatics. *MethodsInf. Med.* 2009;48:38–44. doi: 10.3414/ME9132. [PubMed] [CrossRef] [Google Scholar]

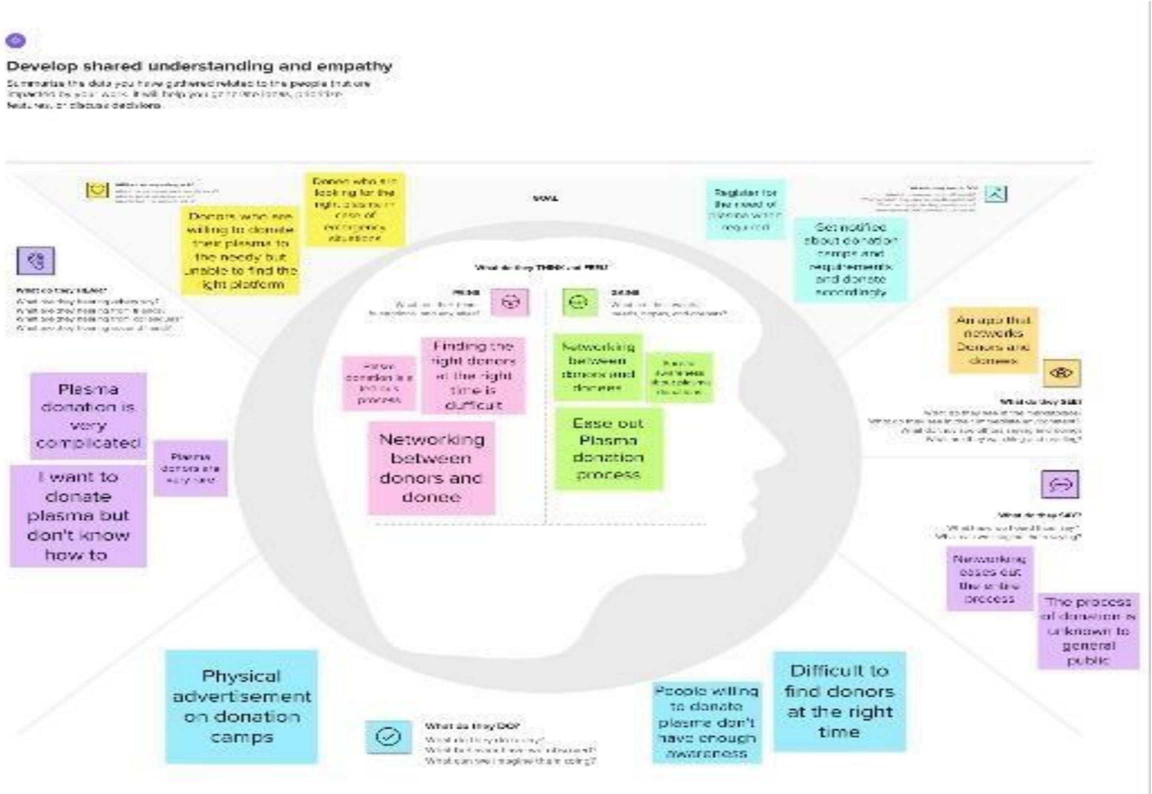
7. Simpao A.F., Ahumada L.M., Gálvez J.A., Rehman M.A. A review ofanalytics andclinical informatics in health care. *J. Med. Syst.* 2014;38:45. doi: 10.1007/s10916-014-0045-x. [PubMed] [CrossRef] [GoogleScholar]

2.3 Problem Statement Definition

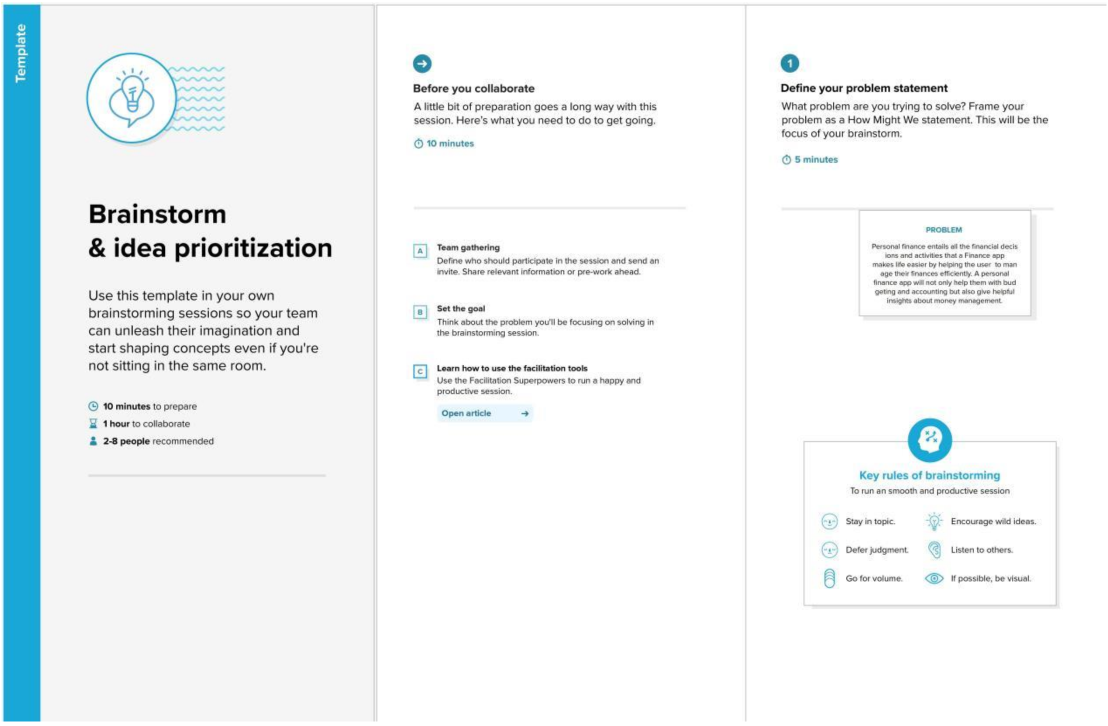
With rapid increase in the usage of social networks sites across the world, there is also a steady increase in blood donation requests as being noticed in the number of posts on these sites such as Facebook and twitter seeking blood donors. Finding blood donor is a challenging issue in almost every country BLOOD donor application provides a reliable platform to connect local blood donors with patients. BLOOD donor creates a communication channel through authenticated clinics whenever a patient needs a donor.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


Develop shared understanding and empathy
Plasma donation is very complicated
I want to donate plasma but don't know how to
Donors who are willing to donate their plasma to the needy but unable to find the right platform
Finding the right donors at the right time is difficult
Networking between donors and donee
Networking between donors and donees
Ease out Plasma donation process
Register for the need of plasma when required
Get notified about donation camps and requirements and donate accordingly
An app that networks Donors and donees
Networking eases out the entire process
The process of donation is unknown to general public
Physical advertisement on donation camps
People willing to donate plasma don't have enough awareness
Difficult to find donors at the right time
What do they DO?

3.2 Ideation & Brainstorming



Template

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.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

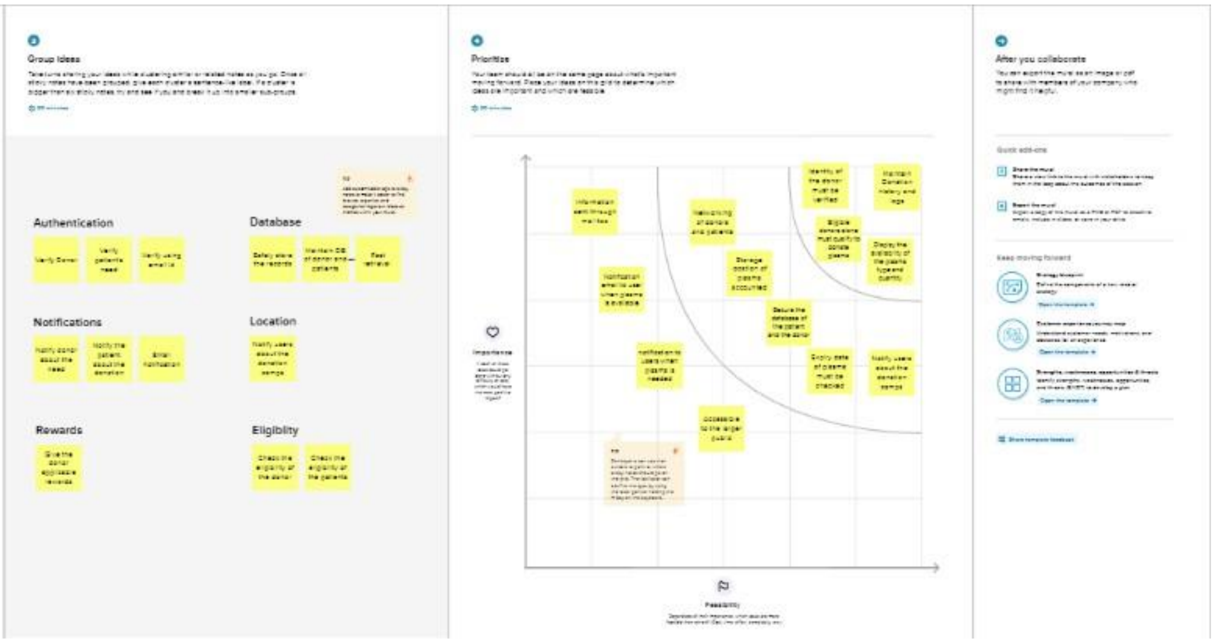
PROBLEM

Personal finance entails all the financial decis ions and activities that a Finance app makes life easier by helping the user to man age their finances efficiently. A personal finance app will not only help them with bud geting and accounting but also give helpful insights about money management.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.


Group Ideas
Authentication
Verify Donor
Verify patients need
Verify using email id
Database
Safely store the records
Maintain DB of donor and patients
Fast retrieval
Notifications
Notify donor about the need
Notify the patient about the donation
Email notification
Location
Notify users about the donation camps
Rewards
Give the donor applicable rewards
Eligiblity
Check the eligibility of the donor
Check the eligibility of the patients
Prioritize
Information sent through mail box
Networking of donors and patients
Identity of the donor must be verified
Maintain Donation history and logs
Eligible donors alone must qualify to donate plasma
Display the availability of the plasma type and quantity
Storage location of plasma accounted
Notification email to user when plasma is available
Secure the database of the patient and the donor
notification to users when plasma is needed
Expiry date of plasma must be checked
Notify users about the donation camps
accessible to the larger public
Importance
Feasibility
After you collaborate

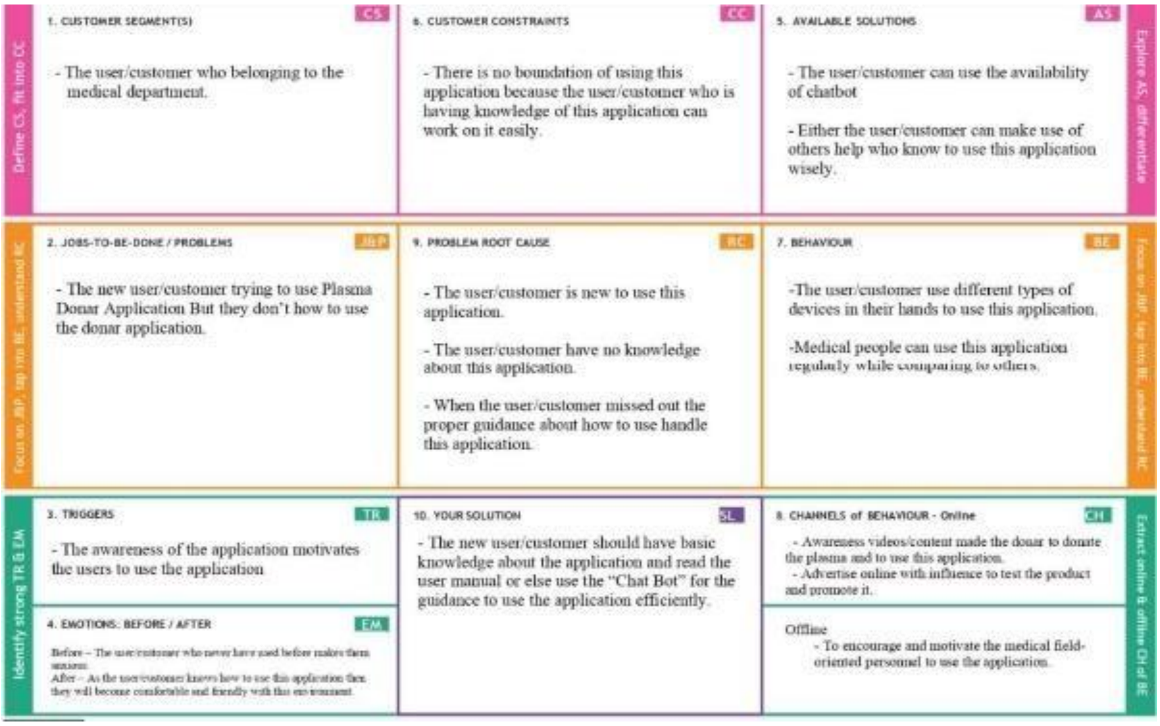
3.3 Proposed Solution

Predict the length of stay of patients.

The length of the stay can be predicted using either Random forest or Decision Tree for more accuracy. Certain parameters like age, stage of the diseases, disease diagnosis, severity of illness, type of admission, facilities allocated, etc., are used for prediction. IBM Cognos will be used for data analytic s. The model will be

trained using colab.It predicts the length of stay (LOS) of the patients with more accuracy. As a result proper resources and therapy can be provided.Patients can get proper treatment and better medical care than before which helps them for their faster recovery. So the prediction minimizes the overflow of patients and helps in resource management and optimize their resource utilization. Hence this leads to faster recovery and lower the expenses for treatment. It improves the trust in hospital management. It avoids the major risk of spreading infection among the hospital staff. This leads to overall safety of hospital staff and patients.Resource consumption is optimized.This model can be used by all government hospitals, private hospitals, and even in The model is trained with the real world hospital survey for better prediction
small clinics.Length of the stay will be predicted with more accuracy.This model predicts the length of the stay for all kinds of patients and predicts with more accuracy.

3.4 Problem Solution fit


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE

1. CUSTOMER SEGMENT(S) CS

- The user/customer who belonging to the medical department.

6. CUSTOMER CONSTRAINTS CC

- There is no boundation of using this application because the user/customer who is having knowledge of this application can work on it easily.

5. AVAILABLE SOLUTIONS AS

- The user/customer can use the availability of chatbot

- Either the user/customer can make use of others help who know to use this application wisely.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- The new user/customer trying to use Plasma Donar Application But they don't how to use the donar application.

9. PROBLEM ROOT CAUSE RC

- The user/customer is new to use this application.

- The user/customer have no knowledge about this application.

- When the user/customer missed out the proper guidance about how to use handle this application.

7. BEHAVIOUR BE

-The user/customer use different types of devices in their hands to use this application.

-Medical people can use this application regularly while comparing to others.

3. TRIGGERS TR

- The awareness of the application motivates the users to use the application

4. EMOTIONS: BEFORE / AFTER EM

Before – The user/customer who never have used before makes them unsure.
After – As the user/customer knows how to use this application then they will become comfortable and friendly with this environment.

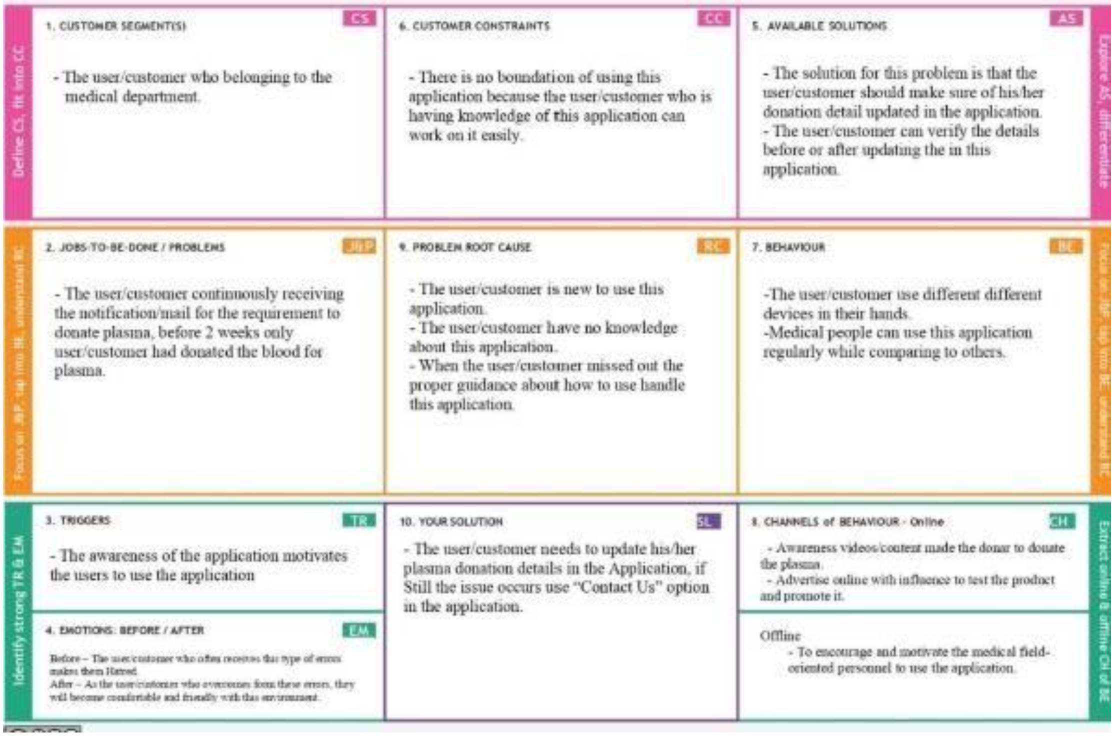
10. YOUR SOLUTION SL

- The new user/customer should have basic knowledge about the application and read the user manual or else use the "Chat Bot" for the guidance to use the application efficiently.

8. CHANNELS of BEHAVIOUR - Online CH

- Awareness videos/content made the donar to donate the plasma and to use this application.
- Advertise online with influence to test the product and promote it.

Offline
- To encourage and motivate the medical field-oriented personnel to use the application.


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE

1. CUSTOMER SEGMENT(S) CS

- The user/customer who belonging to the medical department.

6. CUSTOMER CONSTRAINTS CC

- There is no boundation of using this application because the user/customer who is having knowledge of this application can work on it easily.

5. AVAILABLE SOLUTIONS AS

- The solution for this problem is that the user/customer should make sure of his/her donation detail updated in the application.
- The user/customer can verify the details before or after updating the in this application.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- The user/customer continuously receiving the notification/mail for the requirement to donate plasma, before 2 weeks only user/customer had donated the blood for plasma.

9. PROBLEM ROOT CAUSE RC

- The user/customer is new to use this application.
- The user/customer have no knowledge about this application.
- When the user/customer missed out the proper guidance about how to use handle this application.

7. BEHAVIOUR BE

-The user/customer use different different devices in their hands.
-Medical people can use this application regularly while comparing to others.

3. TRIGGERS TR

- The awareness of the application motivates the users to use the application

4. EMOTIONS: BEFORE / AFTER EM

Before – The user/customer who often receives this type of errors makes them Hatred.
After – As the user/customer who overcomes from these errors, they will become comfortable and friendly with this environment.

10. YOUR SOLUTION SL

- The user/customer needs to update his/her plasma donation details in the Application, if Still the issue occurs use "Contact Us" option in the application.

8. CHANNELS of BEHAVIOUR - Online CH

- Awareness videos/content made the donar to donate the plasma.
- Advertise online with influence to test the product and promote it.

Offline
- To encourage and motivate the medical field-oriented personnel to use the application.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

FUNCTIONALREQUIREMENTS:

Following are the functional requirements of the proposed solution.

FRNo.	FunctionalRequirement(Epic)	SubRequirement(Story/Sub-Task)
FR-1	User Registration	Registration through Form(WebApp)
FR-2	User Confirmation	Confirmation via Email
FR-3	Certification	After the donor donates plasma, we will give them a certificate of appreciation and authentication.

FR-4	Statistical data	The availability of plasma is given in thepage as stats,which will be helpful for the users.
FR-5	User Plasma Request	Users can request to donate plasma byfilling out the request form on the page.
FR-6	Searching/reporting requirements	Users can use the search bar to lookup information about camps and other topics.

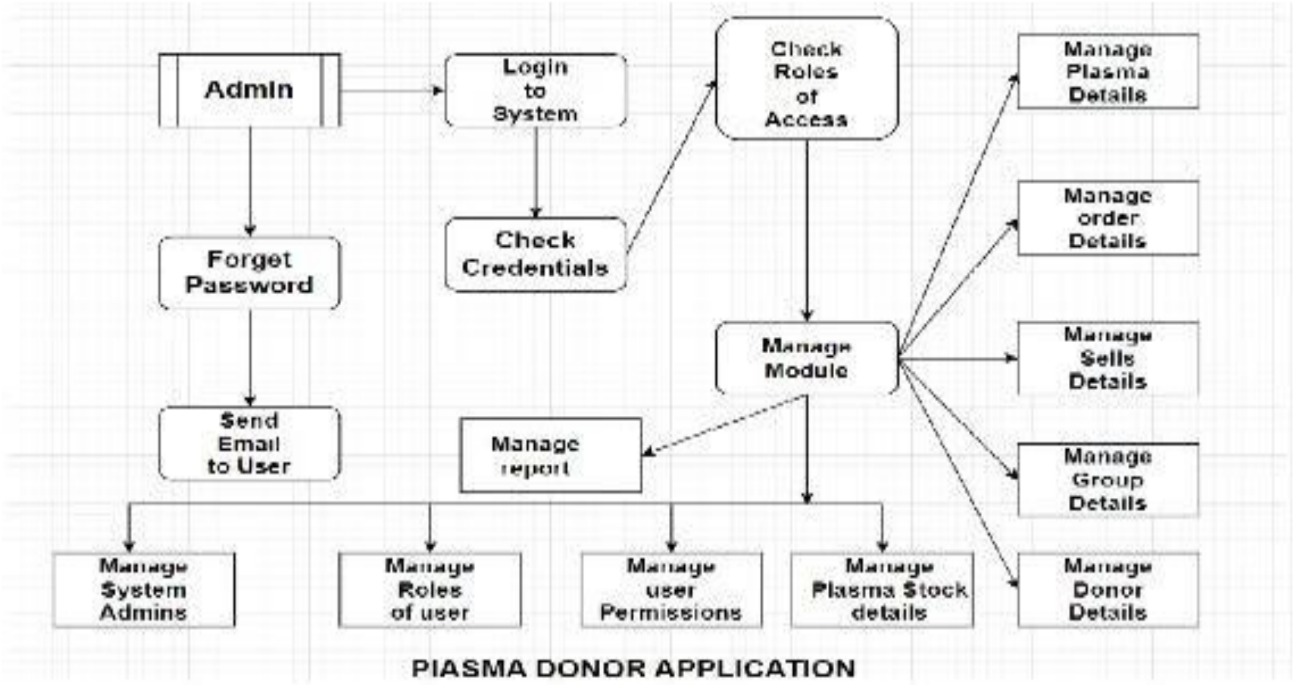
3.1 Non-Functional requirements

NFR-4	**Performance**	Users should have a proper Internet Connection.
NFR-5	**Availability**	The system including the online and offline components should be available 24/7.
NFR-6	**Scalability**	The application has the ability to handle growing number of users and load without compromising on Performance and causing disruptions to user experience.

5. PROJECT DESIGN

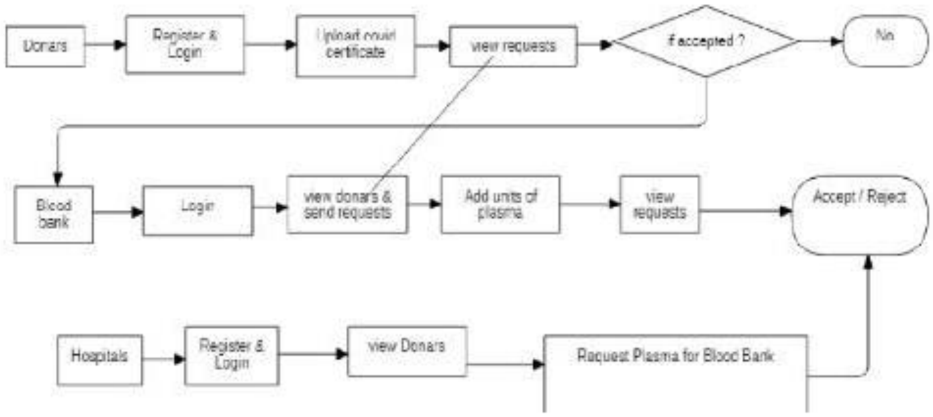
5.1 Data Flow Diagrams

Diagram:

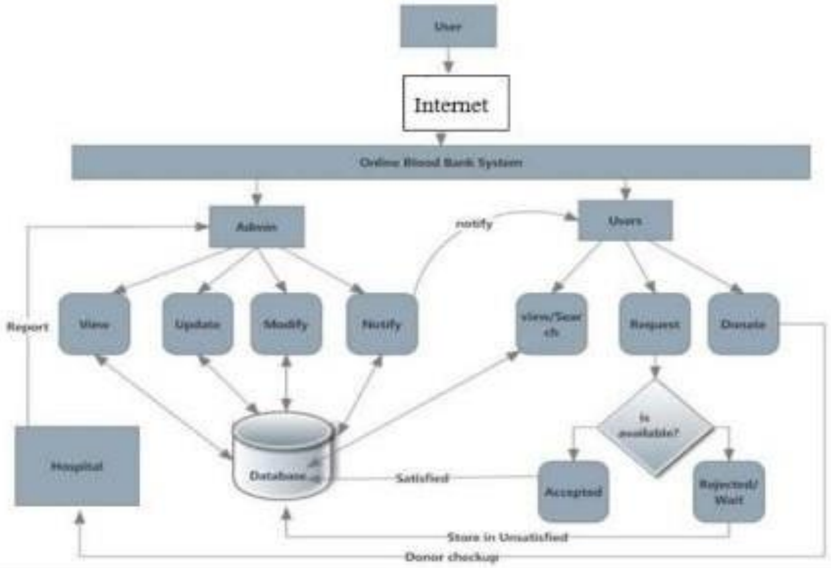


5.2 **Solution & Technical Architecture**

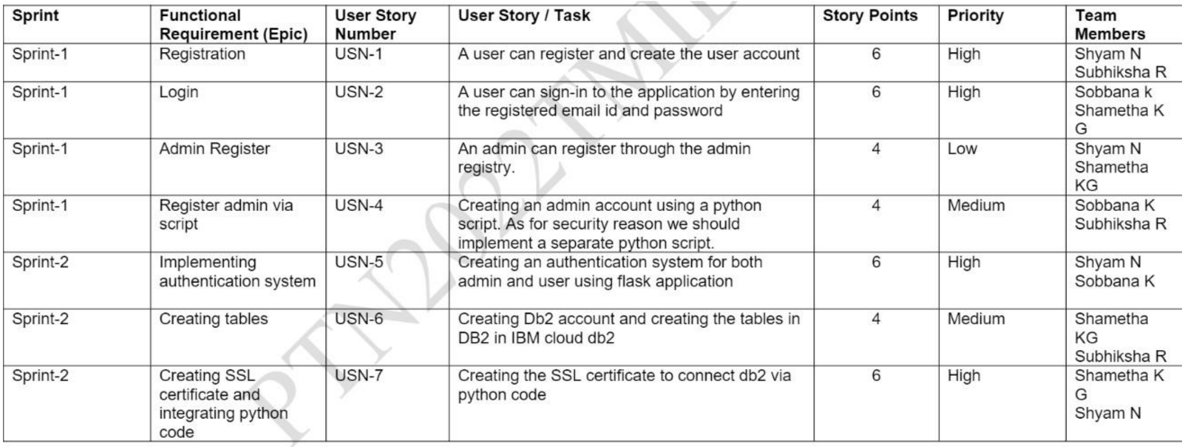
SOLUTION ARCHITECTURE -DONOR



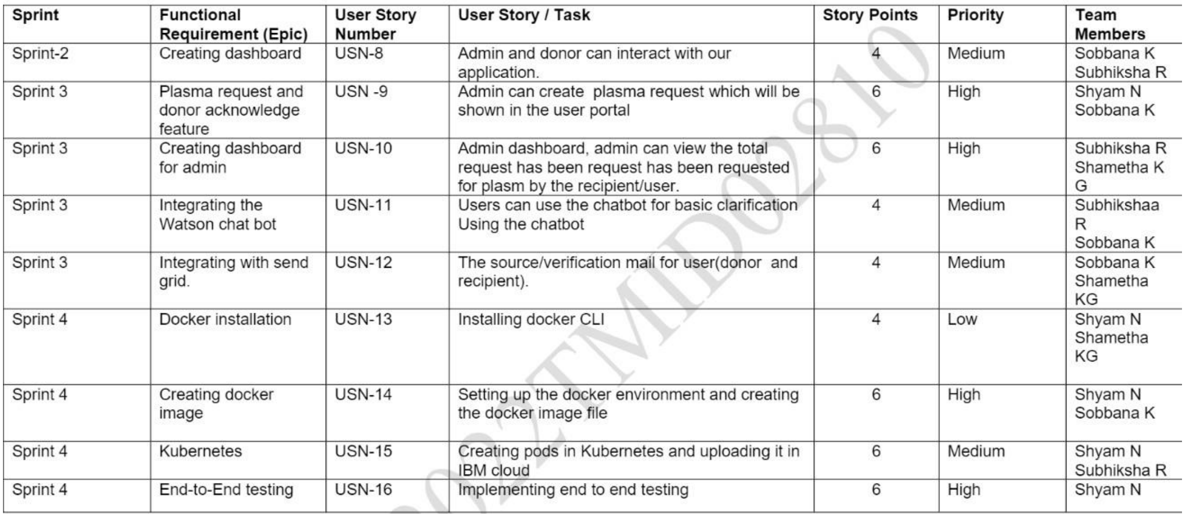
SOLUTION ARCHITECTURE -RECIPIENT



5.3 Table-2: Application Characteristics:



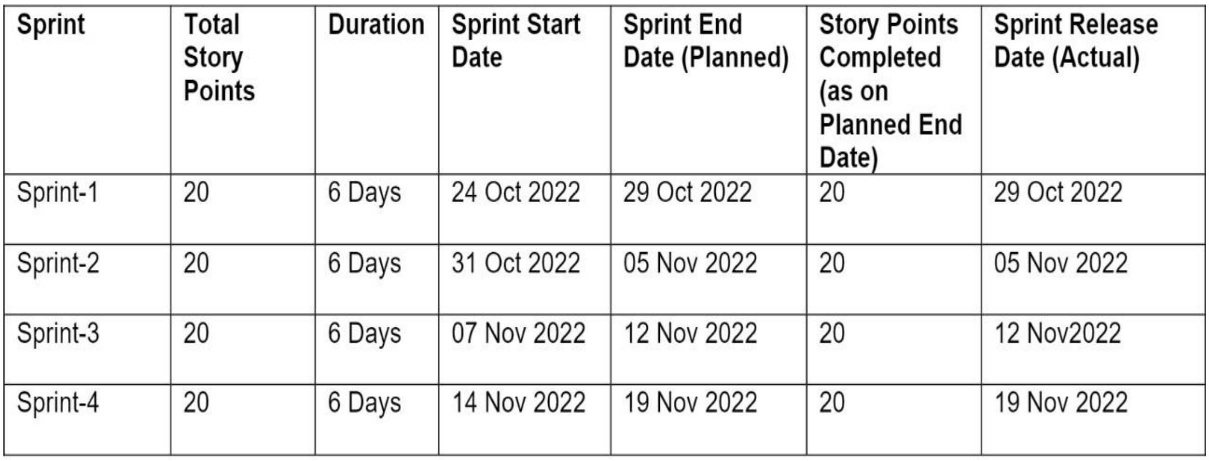
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	A user can register and create the user account	6	High	Shyam N Subhiksha R
Sprint-1	Login	USN-2	A user can sign-in to the application by entering the registered email id and password	6	High	Sobbana k Shametha K G
Sprint-1	Admin Register	USN-3	An admin can register through the admin registry.	4	Low	Shyam N Shametha KG
Sprint-1	Register admin via script	USN-4	Creating an admin account using a python script. As for security reason we should implement a separate python script.	4	Medium	Sobbana K Subhiksha R
Sprint-2	Implementing authentication system	USN-5	Creating an authentication system for both admin and user using flask application	6	High	Shyam N Sobbana K
Sprint-2	Creating tables	USN-6	Creating Db2 account and creating the tables in DB2 in IBM cloud db2	4	Medium	Shametha KG Subhiksha R
Sprint-2	Creating SSL certificate and integrating python code	USN-7	Creating the SSL certificate to connect db2 via python code	6	High	Shametha K G Shyam N



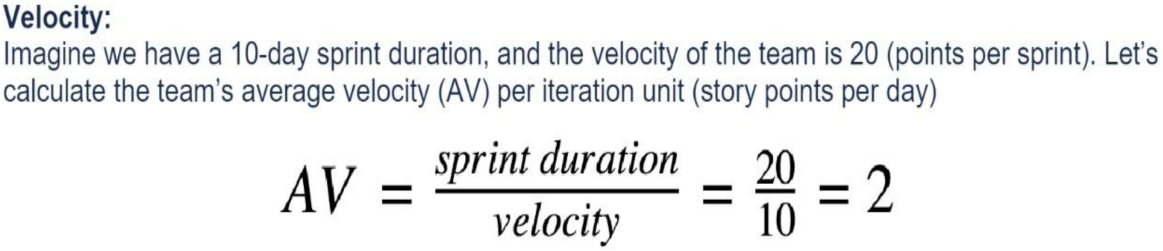
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Creating dashboard	USN-8	Admin and donor can interact with our application.	4	Medium	Sobbana K Subhiksha R
Sprint 3	Plasma request and donor acknowledge feature	USN -9	Admin can create plasma request which will be shown in the user portal	6	High	Shyam N Sobbana K
Sprint 3	Creating dashboard for admin	USN-10	Admin dashboard, admin can view the total request has been request has been requested for plasm by the recipient/user.	6	High	Subhiksha R Shametha K G
Sprint 3	Integrating the Watson chat bot	USN-11	Users can use the chatbot for basic clarification Using the chatbot	4	Medium	Subhikshaa R Sobbana K
Sprint 3	Integrating with send grid.	USN-12	The source/verification mail for user(donor and recipient).	4	Medium	Sobbana K Shametha KG
Sprint 4	Docker installation	USN-13	Installing docker CLI	4	Low	Shyam N Shametha KG
Sprint 4	Creating docker image	USN-14	Setting up the docker environment and creating the docker image file	6	High	Shyam N Sobbana K
Sprint 4	Kubernetes	USN-15	Creating pods in Kubernetes and uploading it in IBM cloud	6	Medium	Shyam N Subhiksha R
Sprint 4	End-to-End testing	USN-16	Implementing end to end testing	6	High	Shyam N

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



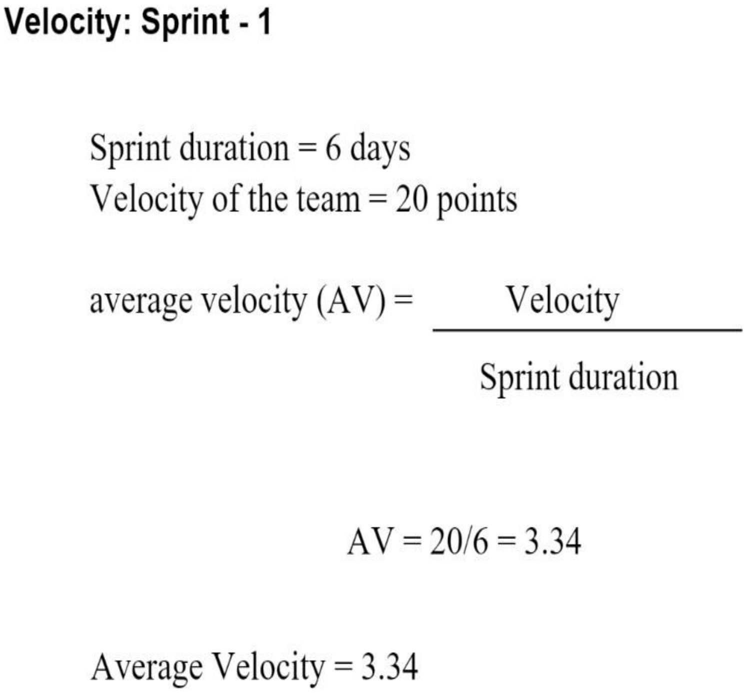
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022



Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$



Velocity: Sprint - 1

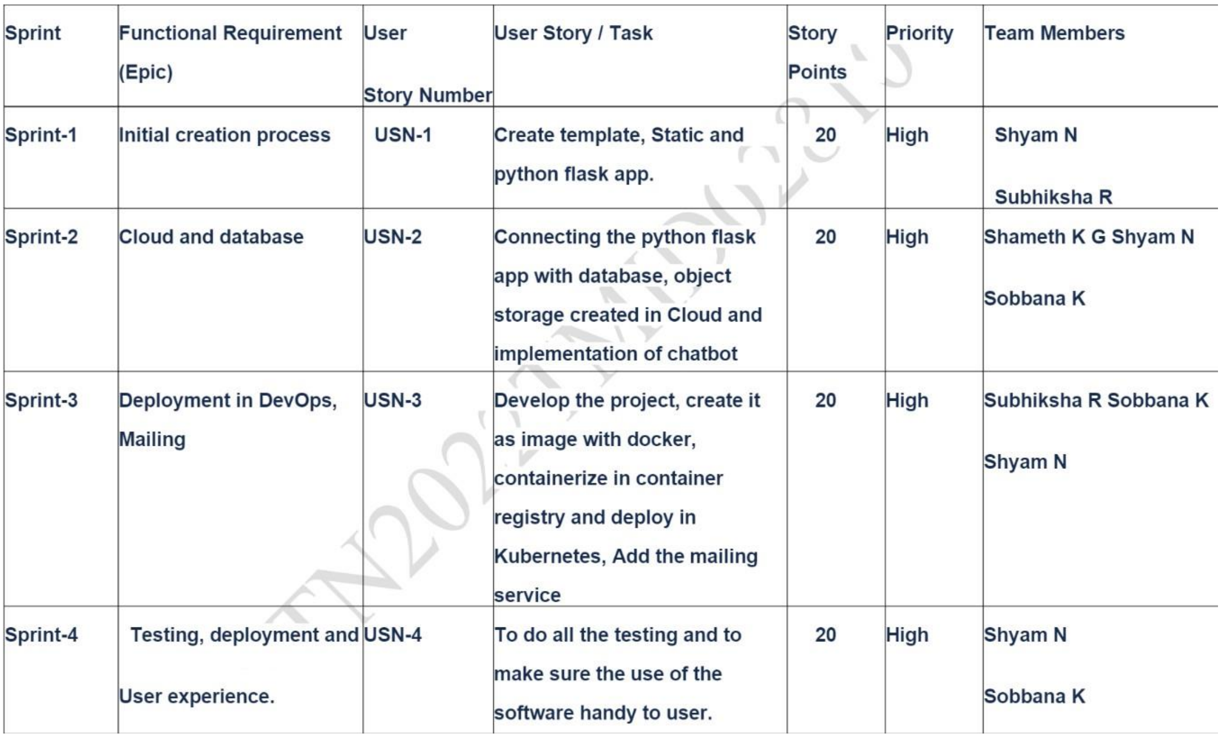
Sprint duration = 6 days
Velocity of the team = 20 points

$$\text{average velocity (AV)} = \frac{\text{Velocity}}{\text{Sprint duration}}$$

AV = 20/6 = 3.34

Average Velocity = 3.34

2.sprint delivery schedule



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Initial creation process	USN-1	Create template, Static and python flask app.	20	High	Shyam N Subhiksha R
Sprint-2	Cloud and database	USN-2	Connecting the python flask app with database, object storage created in Cloud and implementation of chatbot	20	High	Shameth K G Shyam N Sobbana K
Sprint-3	Deployment in DevOps, Mailing	USN-3	Develop the project, create it as image with docker, containerize in container registry and deploy in Kubernetes, Add the mailing service	20	High	Subhiksha R Sobbana K Shyam N
Sprint-4	Testing, deployment and User experience.	USN-4	To do all the testing and to make sure the use of the software handy to user.	20	High	Shyam N Sobbana K

Feature 1

User Login:

The **login page** allows a user to gain access to an application by entering their username and password or by authenticating using a social media login.

Code:

Code

HTML :

```
<!DOCTYPE html>
<!-- This site was created in Webflow. https://www.webflow.com -->
<!-- Last Published: Tue Nov 15 2022 14:21:29 GMT+0000 (CoordinatedUniversal Time) -
->
<html            data-wf-domain="kenkocare.webflow.io"            data-wf-
page="611b46b32597171610edf12d"                                   data-wf-
site="611b46b32597170accedf128" data-wf-status="1">
<head>
<meta charset="utf-8"/>
<title>Plasma donor</title>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<meta content="Webflow" name="generator"/>
<link href="home.css" rel="stylesheet" type="text/css"/>
<script src="https://ajax.googleapis.com/ajax/libs/webfont/1.6.26/webfont.js" type="text/javascript"></script>
<script                type="text/javascript">
WebFont.load({
google: {
```

```
families: ["Open Sans:300,300italic,400,400italic,600,600italic,700,700italic,800,800italic",
"Roboto:300,regular,500"]
}
});
</script>
<!--[if lt IE 9]><script
src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js"
type="text/javascript"></script><![endif]-->
<script type="text/javascript">
!function(o, c) {
var n = c.documentElement
, t = " w-mod-"; n.className +=
t + "js",
("ontouchstart"in o || o.DocumentTouch && c instanceof DocumentTouch) &&(n.className += t + "touch")
}(window, document);
</script>
<link href="https://uploads-ssl.webflow.com/img/favicon.ico" rel="shortcut icon"
type="image/x-icon"/>
<link href="https://uploads-ssl.webflow.com/img/webclip.png" rel="apple-touch-icon"/>
</head>
<body class="body">
<div data-collapse="medium" data-animation="default" data-duration="400" data-
easing="ease" data-easing2="ease" role="banner" class="navigation-barw-nav">
<div class="container-2 w-container">
<a href="/" aria-current="page" class="brand-link w-nav-brand w--current">
<h1 class="brand-text">plasma donor</h1>
</a>
<nav role="navigation" class="navigation-menu w-nav-menu">
<a href="C:\Users\Home\Desktop\Project\Home.html" aria-current="page"
class="navigation-link w-nav-link w--current">Home</a>
<a href="/contact" class="navigation-link w-nav-link">Contact</a>
<a href="C:\Users\Home\Desktop\Project\reg\reg.html" class="button-2 w-button">Register
now</a>
</nav>
</div>
```

```
</div>
<div class="hero-section centered wf-section">
<div data-w-id="e464d218-f801-55d1-1f50-7da00b5bfb8f" style="opacity:0" class="container
w-container">
<h1 data-ix="fade-in-bottom-page-loads" class="hero-heading">plasmadonor</h1>
<div data-ix="fade-in-bottom-page-loads">
<a href="C:\Users\Home\Desktop\Project\reg\reg.html" class="button">signup</a>
<a href="C:\Users\Home\Desktop\Project\login\login.html" class="hollow- button all-
caps">lOGIN</a>
</div>
</div>
</div>
<div class="section wf-section">
<div class="w-container">
<div class="section-title-group">
<h2 class="section-heading centered">we do</h2>
<div class="section-subheading center">ANALYZE YOU IN THESE THREESTEPS</div>
</div>
<div class="w-row">
<div class="w-col w-col-4">
<div data-w-id="270e8437-efa3-df11-d438-de69b23e41e9" style="opacity:0" class="white-
box">
<img src="https://uploads-
ssl.webflow.com/611b46b32597170accedf128/611b779ecb3da267e851af11_im ages.jpeg" alt="AI"
class="grid-image"/>
<h3>Analyzing blood group</h3>
<a href="C:\Users\Home\Downloads\asder\IBM-Project-16293-1659610749-
main\Assignments\Shametha_K_G\Assignment 3\index.html" aria- current="page"
class="brand-link w-nav-brand w--current"></a>
</div>
</div>
<div class="w-col w-col-4">
<div data-w-id="29c25774-570b-ddb2-69b5-f4ddbb194afd" style="opacity:0" class="white-
box">
<img src="https://uploads- ssl.webflow.com/611b46b32597170accedf128/611b46b3259717e5b5edf13c_fe
```

```
ather2-17-white.svg" width="210" alt="Machine learning" class="grid-image"/>
<h3>effective plasma bank managing</h3>
</div>
</div>
<div class="w-col w-col-4">
<div data-w-id="49e69b8a-ef40-4d84-1f92-d2617143b8db" style="opacity:0" class="white-
box">
<img src="https://uploads-
ssl.webflow.com/611b46b32597170accedf128/611b46b325971726d8edf164_fe ather2-22-white.svg"
alt="" class="grid-image"/>
<h3 class="heading">provide experts guidance</h3>
</div>
</div>
</div>
</div>
</div>
<div class="footer wf-section">
<div class="w-container">
<div class="w-row">
<div class="spc w-col w-col-4">
<h5>about plasma donor</h5>
<p>Your health our pride future of healthcare system we are excited to help youat anywhere any
time keep in touch with us...Happy recovery ......................................................</p>
</div>
<div class="spc w-col w-col-4">
<h5>useful links</h5>
<a href="https://www.who.int/philippines/news/feature-stories/detail/20-health- tips-for-2020"
target="_blank" class="footer-link">How to have healthy metabolism</a>
<a href="https://www.healthline.com/health/beauty-skin-care/home-remedies- for-glowing-
skin#coconut-oil" target="_blank" class="footer-link">How to gainnatural skin health</a>
</div>
<div class="w-col w-col-4">
<h5>social</h5>
<div class="footer-link-wrapper w-clearfix">
```

```
<img src="https://uploads-ssl.webflow.com/611b46b32597170accedf128/611b46b3259717cde6edf175_so cial-18.svg" width="20" alt="" class="info-icon"/>
<a href="https://twitter.com/teamCtrlspace" class="footer-link with-icon">Twitter</a>
</div>
<div class="footer-link-wrapper w-clearfix">
<img src="https://uploads-ssl.webflow.com/611b46b32597170accedf128/611b46b32597171d46edf156_so cial-03.svg" width="20" alt="" class="info-icon"/>
<a href="https://www.facebook.com/Team-Ctrl_Space-107663374962795" class="footer-link with-icon">Facebook</a>
</div>
<div class="footer-link-wrapper w-clearfix">
<img src="https://uploads-ssl.webflow.com/611b46b32597170accedf128/611b46b325971735a2edf13d_so cial-11.svg" width="20" alt="" class="info-icon"/>
<a href="https://in.pinterest.com/" class="footer-link with-icon">Pinterest</a>
</div>
<div class="footer-link-wrapper w-clearfix">
<img src="https://uploads-ssl.webflow.com/611b46b32597170accedf128/611b46b3259717714aedf167_so cial-06.svg" width="20" alt="" class="info-icon"/>
<a href="https://www.gmail.com/" class="footer-link with-icon">Google</a>
</div>
</div>
</div>
</div>
</div>
<div class="footer center wf-section">
<div class="w-container">
<div class="footer-text">Copyright ctrl_space Inc. Made in 2021.</div>
</div>
</div>
<script src="https://d3e54v103j8qbb.cloudfront.net/js/jquery-3.5.1.min.dc5e7f18c8.js?site=611b46b32597170accedf128" type="text/javascript" integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous"></script>
```

```
<script src="home.js" type="text/javascript"></script>

<script> window.watsonAssistantChatOptions = {
integrationID: "87ad3502-2685-48d1-bbdd-96ed7b353f93", // The ID of thisintegration.
region: "au-syd", // The region your integration is hosted in. serviceInstanceID: "26b5b847-
d411-43f0-af69-4cd200aed370", // The ID ofyour service instance.
onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
const t=document.createElement('script');
t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/"                    +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
document.head.appendChild(t);
});
</script>
</body>
</html>
```

CSS

```
        body {
          background-color: #edeff2;
          font-family:  'Open  Sans',  sans-serif; color:
          #6a859c;
          font-size: 16px; line-
          height: 20px;
        }

        h1 {
          margin-top: 0px; margin-
          bottom:   10px;  font-size:
          38px;
          line-height:      44px;
          font-weight: 700;
        }
```

```
h2 {
  margin-top: 0px; margin-
  bottom: 10px; color:
  #676770;
  font-size: 32px; line-
  height: 36px; font-
  weight: 300; text-
  align: center;
}

h3 {
  margin-top: 0px;
  margin-bottom: 0px;
  color: #676770;
  font-size: 20px; line-
  height: 30px; font-
  weight: 300; letter-
  spacing: 7px;
  text-transform: uppercase;
}

h4 {
  margin-top: 0px; margin-
  bottom: 10px; font-size:
  18px;
  line-height: 24px;
  font-weight: 700;
}

h5 {
  margin-top: 20px;
  margin-bottom: 0px;
  color: #676770;
  font-size: 18px; line-
  height: 20px; font-
  weight: 300; letter-
  spacing: 4px;
  text-transform: uppercase;
}

h6 {
  margin-top: 0px; margin-
  bottom: 10px; font-size:
  12px;
  line-height: 18px;
  font-weight: 700;
}

p {
  margin-top: 10px;
  margin-bottom: 10px;
  font-size: 14px;
  line-height: 25px;
  font-weight: 300;
}

.button {
  display: inline-block;
  margin-right: 10px;
  margin-left: 10px;
  padding: 12px 30px;
  border-radius: 4px;
  background-color: #192024;
  -webkit-transition: background-color 300ms ease; transition:
  background-color 300ms ease;
  color: #edeff2; font-
  size: 16px; line-
  height: 21px; font-
  weight: 300; text-
  align: center;
  letter-spacing: 2px; text-
  decoration: none;
  text-transform: uppercase;
}
```

```
.button:hover {
  background-color: #fc7d64;
}

.button.w--current { background-
  color: #2e80b6;
}

.button.full-width
  { display: block;
  width: 100%;
  margin-right: 0px;
  margin-left: 0px;
}

.button.tab { margin-
  right: 8px; margin-
  left: 8px;
  background-color: #92a0ad;
}

.button.tab:hover { background-
  color: #2e80b6;
}

.button.tab.w--current
  { background-color: #2e80b6;
}

.navigation-link
  {margin-top: 3px;
  -webkit-transition: all 300ms ease-in-out;
  transition: all 300ms ease-in-out;
  color: #676770;
}

.navigation-link:hover
  {color: #2e9dff;
```

```
        }

        .navigation-link.w--current {color:
          #192024;
          text-decoration: underline;
        }
```

JAVA SCRIPT:

```
var interopRequireWildcard =___________webpack_require_(18);

Object.defineProperty(exports, "___________esModule",
{value: true
});
var          _exportNames          =
{     IX2EngineActionTypes:     true,
IX2EngineConstants: true
};
exports.IX2EngineConstants = exports.IX2EngineActionTypes = void 0;

var triggerEvents =__________webpack_require_(188);

Object.keys(_triggerEvents).forEach(function  (key)  {  if  (key  ===
"default" || key === "_____________________esModule") return;
if      (Object.prototype.hasOwnProperty.call(_exportNames,      key))      return;
Object.defineProperty(exports, key, {
enumerable:     true,     get:
function get() {
return _triggerEvents[key];
}
});
});

var animationActions =__________webpack_require_(94);

Object.keys(_animationActions).forEach(function  (key)  { if  (key  ===
"default" || key === "_____________________esModule") return;
if (Object.prototype.hasOwnProperty.call(_exportNames, key)) return;Object.defineProperty(exports, key,
{
```

```
enumerable:      true,      get:
function get() {
return _animationActions[key];
}
});
});

var triggerInteractions =__________webpack_require_(189);

Object.keys(_triggerInteractions).forEach(function  (key)  { if  (key  ===
"default" || key === "____________________esModule") return;
if (Object.prototype.hasOwnProperty.call(_exportNames, key)) return;Object.defineProperty(exports, key,
{
enumerable:      true,      get:
function get() {
return _triggerInteractions[key];
}
});
});

var reducedMotion =__________webpack_require_(190);

Object.keys(_reducedMotion).forEach(function  (key)  { if  (key  ===
"default" || key === "____________________esModule") return;
if (Object.prototype.hasOwnProperty.call(_exportNames, key)) return;Object.defineProperty(exports, key,
{
enumerable:      true,      get:
function get() {
return _reducedMotion[key];
}
});
});

var IX2EngineActionTypes = interopRequireWildcard(webpack_require_(191));

exports.IX2EngineActionTypes = IX2EngineActionTypes;
```

```
var IX2EngineConstants = interopRequireWildcard(webpack_require_(192));

exports.IX2EngineConstants = IX2EngineConstants;

/*/ }),
/* 5 */
/*/ (function(module, exports) {

var FunctionPrototype =
Function.prototype; var bind =
FunctionPrototype.bind;
var call = FunctionPrototype.call;
var callBind = bind && bind.bind(call);

module.exports = bind ? function
(fn) {return fn && callBind(call,
fn);
} : function (fn) {
return fn && function
() { return
call.apply(fn,
arguments);
};
};


/*/ }),
/* 6 */
/*/ (function(module, exports,

_webpack_require_) { var freeGlobal =

_webpack_require_(99);

/** Detect free variable `self`. */
var freeSelf = typeof self == 'object' && self && self.Object === Object &&
self;

/** Used as a reference to the global object. */
var root = freeGlobal || freeSelf || Function('return this')();

module.exports = root
```

3. ADVANTAGES & DISADVANTAGES

Advantages

- Analysing clinical data to improve medical research
- Using patient data to improve health outcomes
- Gaining operational insights from healthcare provider data
- Improved staffing through health business management analytics
- Research and prediction of disease.
- Automation of hospital administrative processes.
- Early detection of disease.
- Prevention of unnecessary doctor's visits.
- Discovery of new drugs.
- More accurate calculation of health insurance rates.
- More effective sharing of patient data.

Disadvantages

Replacing Medical Personnel

Application of technology in every sphere of human life is improving the way things are done. These technologies are are also posing some threat to world of works. Robotics are replacing human labour.

Data Safety

Data security is another challenge in applying big data in healthcare. Big data storage is usually targets of hackers. This endangers the safety of medical data. Healthcare organisations are very much concerned about the safety of patients' sensitive personal data. For this, all healthcare applications must meet the requirement for data security and be HIPAA compliant before they can be deployed for healthcare services

Privacy

One of the major drawbacks in the application of big data in healthcare industry is the issue of lack of privacy. Application of big data technologies involves monitoring of patient's data, tracking of medical inventory and assets, organizing collected data, and visualization of data on the dashboard and the reports. So visualization of sensitive medical data especially that of the patients creates negative impression of big data as it violets privacy

Man Power

`Applying big data solutions in healthcare requires special skills, and such kills are scarce. Handling of big data requires the combination of medical, technological and statistical knowledge.

3. CONCLUSION

Data analytics is the science of analysing raw datasets in order to derive a conclusion regarding the information they hold. It enables us to discover patterns in the raw data and draw valuable information from them.To some, the domain of healthcare data analytics may look new, but it has a lot of potential, especially if you wish to engage in challenging job roles and build a strong data analytics profile in the upcoming years. In this blog, we have covered some of the major topics such as what is healthcare data analytics, its applications, scope, and benefits, etc. We hope it helps you in your decision-making as a healthcare data analytics professional.

4. FUTURE SCOPE

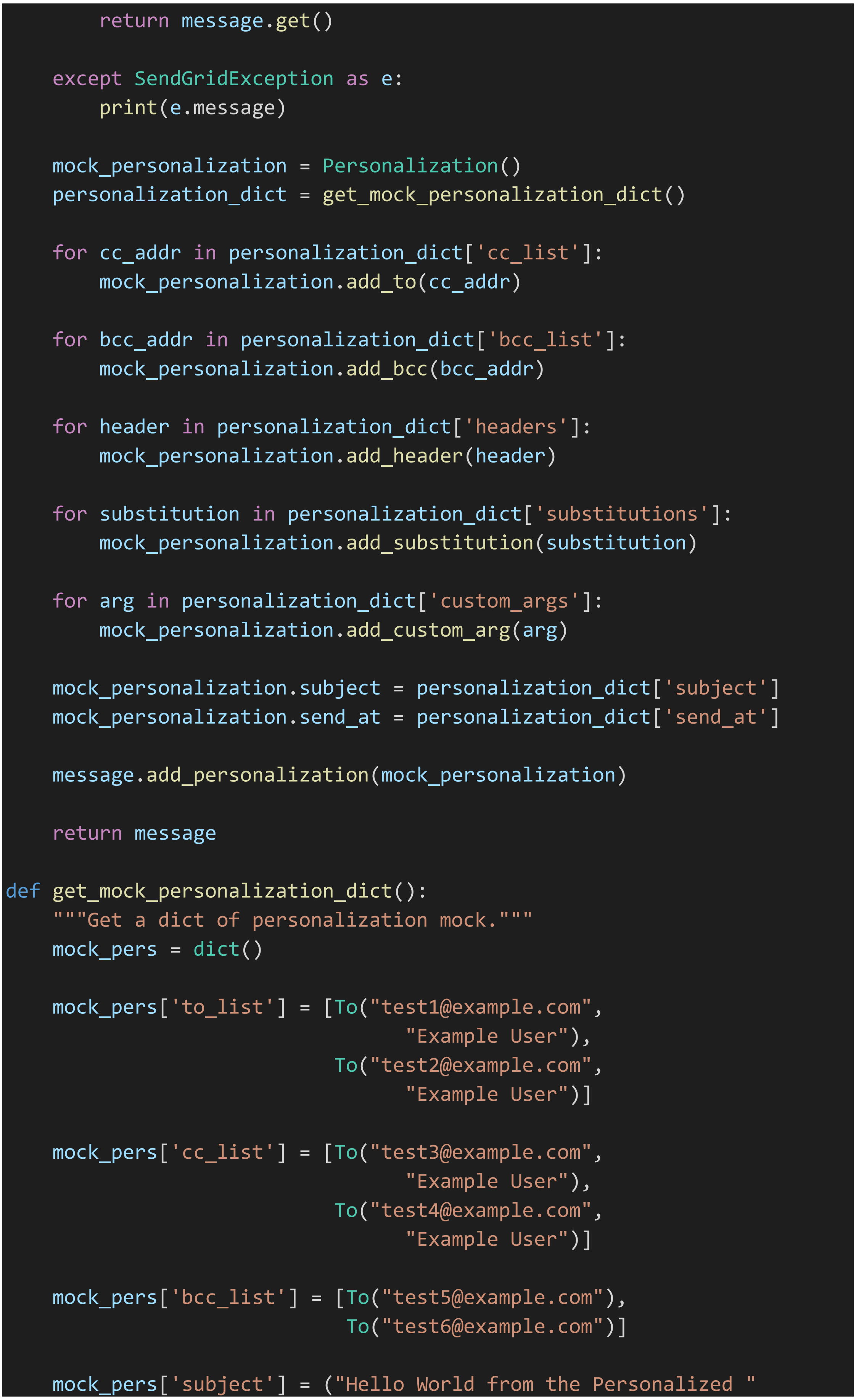
The Future of Healthcare, Intel provides a foundation for big data platforms and AI to advance health analytics. Predictive data analytics is helping health organizations enhance patient care, improve outcomes, and reduce costs by anticipating when, where, and how care should be provided.The future of big data in healthcare will be determined by technological breakthroughs from 2022 to 2030. Complete patient care and cost-effective prescription procedures are required for population health management. To assess clinical and claims data, they must be combined on the same platform.

Countries around the world have started to invest more capital in medical infrastructure, pharmaceuticals, and healthcare smart analytics solutions. The market is growing and will continue to expand, given the benefits of healthcare data analytics. It has also risen as a good career option for fresh data science and data analytics graduates or professionals who wish to build their career in the healthcare sector. Due to the sensitivity of the profession, the salary offers for healthcare data analysts are lucrative around the world. Apart from the remuneration, the opportunities to work with some of the biggest names in the healthcare sector is also worth mentioning. Hence, healthcare data analytics is growing to be one of the most rewarding branches of data analytics in the coming future.

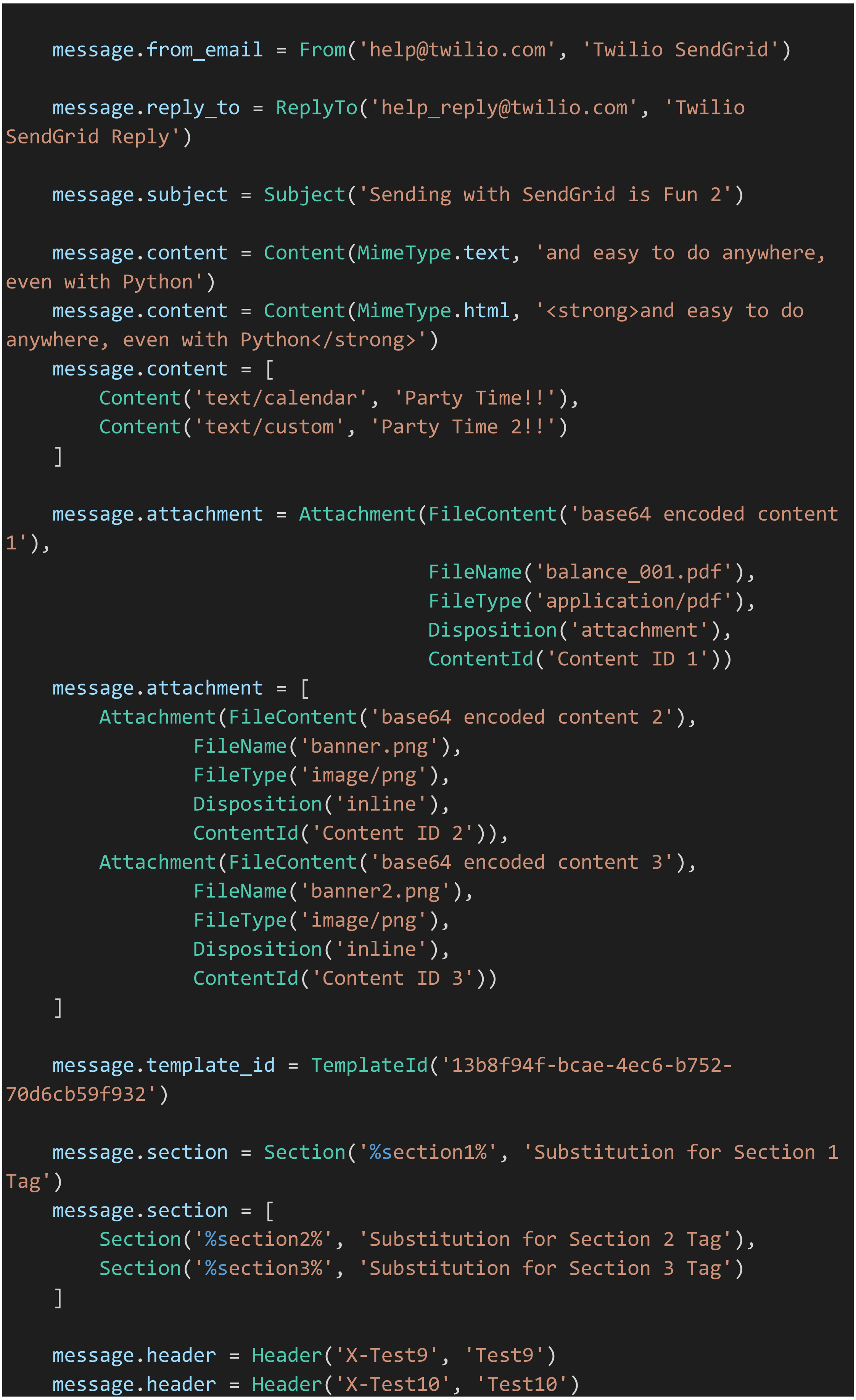
13. APPENDIX

Source Code

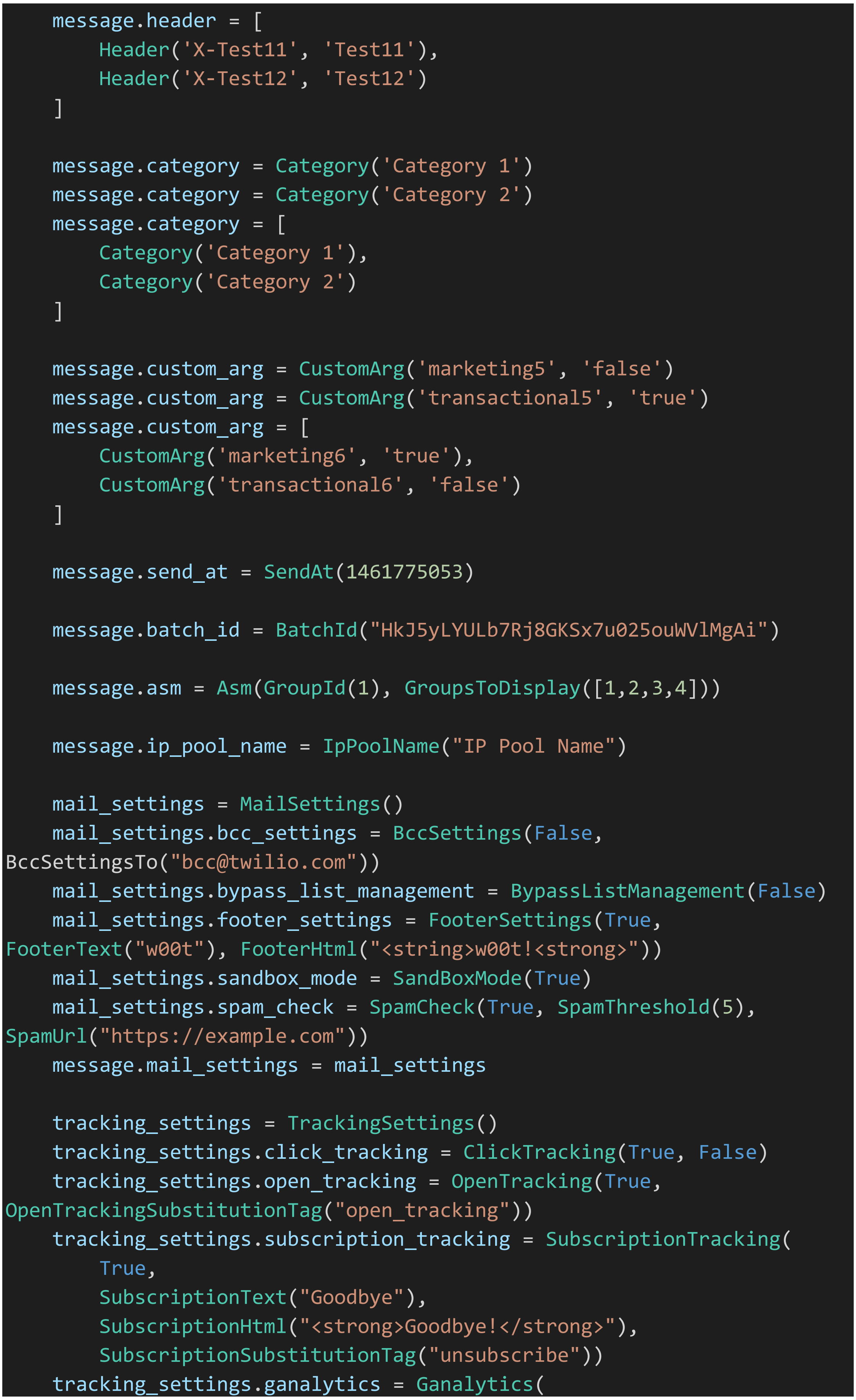
```
import os
import json

from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import *


# NOTE: you will need move this file to the root
# directory of this project to execute properly.


def build_hello_email():
    ## Send a Single Email to a Single Recipient

    message = Mail(from_email=From('from@example.com', 'Example From
Name'),
                   to_emails=To('to@example.com', 'Example To Name'),
                   subject=Subject('Sending with SendGrid is Fun'),
                   plain_text_content=PlainTextContent('and easy to do
anywhere, even with Python'),
                   html_content=HtmlContent('<strong>and easy to do
anywhere, even with Python</strong>'))

    try:
        print(json.dumps(message.get(), sort_keys=True, indent=4))
```



```
        return message.get()

    except SendGridException as e:
        print(e.message)

    mock_personalization = Personalization()
    personalization_dict = get_mock_personalization_dict()

    for cc_addr in personalization_dict['cc_list']:
        mock_personalization.add_to(cc_addr)

    for bcc_addr in personalization_dict['bcc_list']:
        mock_personalization.add_bcc(bcc_addr)

    for header in personalization_dict['headers']:
        mock_personalization.add_header(header)

    for substitution in personalization_dict['substitutions']:
        mock_personalization.add_substitution(substitution)

    for arg in personalization_dict['custom_args']:
        mock_personalization.add_custom_arg(arg)

    mock_personalization.subject = personalization_dict['subject']
    mock_personalization.send_at = personalization_dict['send_at']

    message.add_personalization(mock_personalization)

    return message

def get_mock_personalization_dict():
    """Get a dict of personalization mock."""
    mock_pers = dict()

    mock_pers['to_list'] = [To("test1@example.com",
                                  "Example User"),
                            To("test2@example.com",
                                  "Example User")]

    mock_pers['cc_list'] = [To("test3@example.com",
                                  "Example User"),
                            To("test4@example.com",
                                  "Example User")]

    mock_pers['bcc_list'] = [To("test5@example.com"),
                             To("test6@example.com")]

    mock_pers['subject'] = ("Hello World from the Personalized "
                            "SendGrid Python Library")

    mock_pers['headers'] = [Header("X-Test", "test"),
                            Header("X-Mock", "true")]

    mock_pers['substitutions'] = [Substitution("%name%", "Example
User"),
                                  Substitution("%city%", "Denver")]

    mock_pers['custom_args'] = [CustomArg("user_id", "343"),
                                CustomArg("type", "marketing")]

    mock_pers['send_at'] = 1443636843
    return mock_pers

def build_multiple_emails_personalized():
    # Note that the domain for all From email addresses must match

    message = Mail(from_email=From('from@example.com', 'Example From
Name'),
                subject=Subject('Sending with SendGrid is Fun'),
                plain_text_content=PlainTextContent('and easy to do
anywhere, even with Python'),
                html_content=HtmlContent('<strong>and easy to do
anywhere, even with Python</strong>'))

    mock_personalization = Personalization()
    mock_personalization.add_to(To('test@example.com', 'Example User
1'))
    mock_personalization.add_cc(Cc('test1@example.com', 'Example User
2'))
    message.add_personalization(mock_personalization)

    mock_personalization_2 = Personalization()
    mock_personalization_2.add_to(To('test2@example.com', 'Example User
3'))
    mock_personalization_2.set_from(From('from@example.com', 'Example
From Name 2'))
    mock_personalization_2.add_bcc(Bcc('test3@example.com', 'Example
User 4'))
    message.add_personalization(mock_personalization_2)

    try:
        print(json.dumps(message.get(), sort_keys=True, indent=4))
        return message.get()

    except SendGridException as e:
        print(e.message)
    return message

def build_attachment1():
    """Build attachment mock. Make sure your content is base64 encoded
before passing into attachment.content.
    Another example: https://github.com/sendgrid/sendgrid-
python/blob/HEAD/use_cases/attachment.md"""

    attachment = Attachment()
    attachment.file_content =
("TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbnNl"
                          "Y3RldHVyIGFkaXBpc2NpbmcgZWxpdC4gQ3JhcyBwdW12
")
    attachment.file_type = "application/pdf"
    attachment.file_name = "balance_001.pdf"
    attachment.disposition = "attachment"
    attachment.content_id = "Balance Sheet"
    return attachment


def build_attachment2():
    """Build attachment mock."""
    attachment = Attachment()
    attachment.file_content = "BwdW"
    attachment.file_type = "image/png"
    attachment.file_name = "banner.png"
    attachment.disposition = "inline"
    attachment.content_id = "Banner"
    return attachment

def build_kitchen_sink():
    """All settings set"""
    from sendgrid.helpers.mail import (
        Mail, From, To, Cc, Bcc, Subject, PlainTextContent,
        HtmlContent, SendGridException, Substitution,
        Header, CustomArg, SendAt, Content, MimeType, Attachment,
        FileName, FileContent, FileType, Disposition, ContentId,
        TemplateId, Section, ReplyTo, Category, BatchId, Asm,
        GroupId, GroupsToDisplay, IpPoolName, MailSettings,
        BccSettings, BccSettingsEmail, BypassListManagement,
        FooterSettings, FooterText, FooterHtml, SandBoxMode,
        SpamCheck, SpamThreshold, SpamUrl, TrackingSettings,
        ClickTracking, SubscriptionTracking, SubscriptionText,
        SubscriptionHtml, SubscriptionSubstitutionTag,
        OpenTracking, OpenTrackingSubstitutionTag, Ganalytics,
        UtmSource, UtmMedium, UtmTerm, UtmContent, UtmCampaign)
    import time
    import datetime
    message = Mail()

    # Define Personalizations

    message.to = To('test1@sendgrid.com', 'Example User1', p=0)
    message.to = [
        To('test2@sendgrid.com', 'Example User2', p=0),
        To('test3@sendgrid.com', 'Example User3', p=0)
    ]

    message.cc = Cc('test4@example.com', 'Example User4', p=0)
    message.cc = [
        Cc('test5@example.com', 'Example User5', p=0),
        Cc('test6@example.com', 'Example User6', p=0)
    ]

    message.bcc = Bcc('test7@example.com', 'Example User7', p=0)
    message.bcc = [
        Bcc('test8@example.com', 'Example User8', p=0),
        Bcc('test9@example.com', 'Example User9', p=0)
    ]

    message.subject = Subject('Sending with SendGrid is Fun 0', p=0)

    message.header = Header('X-Test1', 'Test1', p=0)
    message.header = Header('X-Test2', 'Test2', p=0)
    message.header = [
        Header('X-Test3', 'Test3', p=0),
        Header('X-Test4', 'Test4', p=0)
    ]

    message.substitution = Substitution('%name1%', 'Example Name 1',
p=0)
    message.substitution = Substitution('%city1%', 'Example City 1',
p=0)
    message.substitution = [
        Substitution('%name2%', 'Example Name 2', p=0),
        Substitution('%city2%', 'Example City 2', p=0)
    ]

    message.custom_arg = CustomArg('marketing1', 'true', p=0)
    message.custom_arg = CustomArg('transactional1', 'false', p=0)
    message.custom_arg = [
        CustomArg('marketing2', 'false', p=0),
        CustomArg('transactional2', 'true', p=0)
    ]
    message.send_at = SendAt(1461775051, p=0)

    message.to = To('test10@example.com', 'Example User10', p=1)
    message.to = [
        To('test11@example.com', 'Example User11', p=1),
        To('test12@example.com', 'Example User12', p=1)
    ]

    message.cc = Cc('test13@example.com', 'Example User13', p=1)
    message.cc = [
        Cc('test14@example.com', 'Example User14', p=1),
        Cc('test15@example.com', 'Example User15', p=1)
    ]

    message.bcc = Bcc('test16@example.com', 'Example User16', p=1)
    message.bcc = [
        Bcc('test17@example.com', 'Example User17', p=1),
        Bcc('test18@example.com', 'Example User18', p=1)
    ]

    message.header = Header('X-Test5', 'Test5', p=1)
    message.header = Header('X-Test6', 'Test6', p=1)
    message.header = [
        Header('X-Test7', 'Test7', p=1),
        Header('X-Test8', 'Test8', p=1)
    ]

    message.substitution = Substitution('%name3%', 'Example Name 3',
p=1)
    message.substitution = Substitution('%city3%', 'Example City 3',
p=1)
    message.substitution = [
        Substitution('%name4%', 'Example Name 4', p=1),
        Substitution('%city4%', 'Example City 4', p=1)
    ]

    message.custom_arg = CustomArg('marketing3', 'true', p=1)
    message.custom_arg = CustomArg('transactional3', 'false', p=1)
    message.custom_arg = [
        CustomArg('marketing4', 'false', p=1),
        CustomArg('transactional4', 'true', p=1)
    ]

    message.send_at = SendAt(1461775052, p=1)

    message.subject = Subject('Sending with SendGrid is Fun 1', p=1)

    # The values below this comment are global to entire message
```



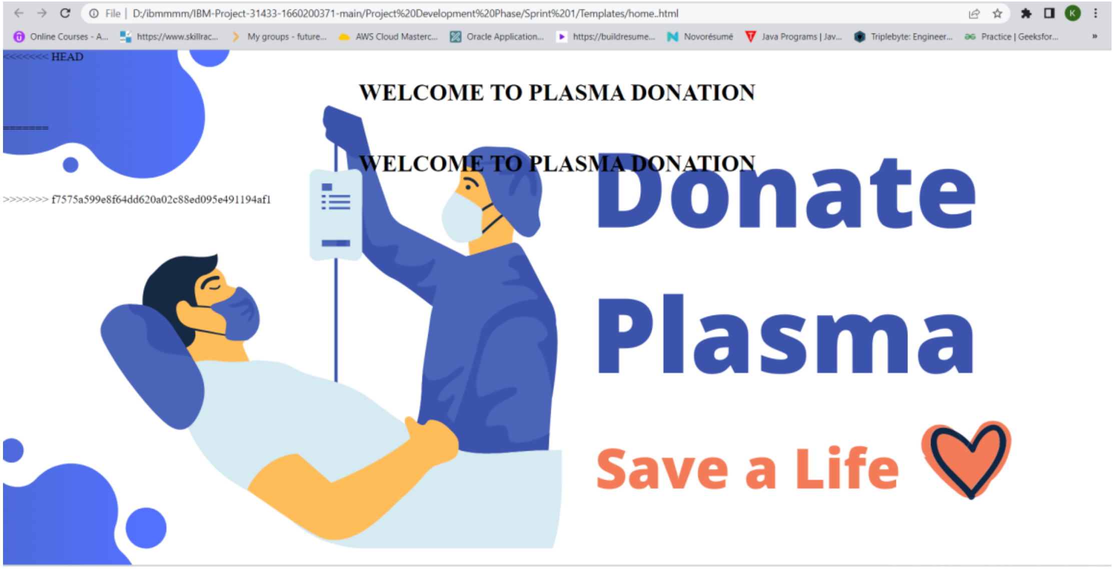
```
    message.from_email = From('help@twilio.com', 'Twilio SendGrid')

    message.reply_to = ReplyTo('help_reply@twilio.com', 'Twilio
SendGrid Reply')

    message.subject = Subject('Sending with SendGrid is Fun 2')

    message.content = Content(MimeType.text, 'and easy to do anywhere,
even with Python')
    message.content = Content(MimeType.html, '<strong>and easy to do
anywhere, even with Python</strong>')
    message.content = [
        Content('text/calendar', 'Party Time!!'),
        Content('text/custom', 'Party Time 2!!')
    ]

    message.attachment = Attachment(FileContent('base64 encoded content
1'),
                                    FileName('balance_001.pdf'),
                                    FileType('application/pdf'),
                                    Disposition('attachment'),
                                    ContentId('Content ID 1'))
    message.attachment = [
        Attachment(FileContent('base64 encoded content 2'),
                FileName('banner.png'),
                FileType('image/png'),
                Disposition('inline'),
                ContentId('Content ID 2')),
        Attachment(FileContent('base64 encoded content 3'),
                FileName('banner2.png'),
                FileType('image/png'),
                Disposition('inline'),
                ContentId('Content ID 3'))
    ]

    message.template_id = TemplateId('13b8f94f-bcae-4ec6-b752-
70d6cb59f932')

    message.section = Section('%section1%', 'Substitution for Section 1
Tag')
    message.section = [
        Section('%section2%', 'Substitution for Section 2 Tag'),
        Section('%section3%', 'Substitution for Section 3 Tag')
    ]

    message.header = Header('X-Test9', 'Test9')
    message.header = Header('X-Test10', 'Test10')
```



```
    message.header = [
        Header('X-Test11', 'Test11'),
        Header('X-Test12', 'Test12')
    ]

    message.category = Category('Category 1')
    message.category = Category('Category 2')
    message.category = [
        Category('Category 1'),
        Category('Category 2')
    ]

    message.custom_arg = CustomArg('marketing5', 'false')
    message.custom_arg = CustomArg('transactional5', 'true')
    message.custom_arg = [
        CustomArg('marketing6', 'true'),
        CustomArg('transactional6', 'false')
    ]

    message.send_at = SendAt(1461775053)

    message.batch_id = BatchId("HkJ5yLYULb7Rj8GKSx7u025ouWVlMgAi")

    message.asm = Asm(GroupId(1), GroupsToDisplay([1,2,3,4]))

    message.ip_pool_name = IpPoolName("IP Pool Name")

    mail_settings = MailSettings()
    mail_settings.bcc_settings = BccSettings(False,
BccSettingsTo("bcc@twilio.com"))
    mail_settings.bypass_list_management = BypassListManagement(False)
    mail_settings.footer_settings = FooterSettings(True,
FooterText("w00t"), FooterHtml("<string>w00t!<strong>"))
    mail_settings.sandbox_mode = SandBoxMode(True)
    mail_settings.spam_check = SpamCheck(True, SpamThreshold(5),
SpamUrl("https://example.com"))
    message.mail_settings = mail_settings

    tracking_settings = TrackingSettings()
    tracking_settings.click_tracking = ClickTracking(True, False)
    tracking_settings.open_tracking = OpenTracking(True,
OpenTrackingSubstitutionTag("open_tracking"))
    tracking_settings.subscription_tracking = SubscriptionTracking(
        True,
        SubscriptionText("Goodbye"),
        SubscriptionHtml("<strong>Goodbye!</strong>"),
        SubscriptionSubstitutionTag("unsubscribe"))
    tracking_settings.ganalytics = Ganalytics(
        True,
        UtmSource("utm_source"),
        UtmMedium("utm_medium"),
        UtmTerm("utm_term"),
        UtmContent("utm_content"),
        UtmCampaign("utm_campaign"))
    message.tracking_settings = tracking_settings

    return message

def send_multiple_emails_personalized():
    # Assumes you set your environment variable:
    # https://github.com/sendgrid/sendgrid-python/blob/HEAD/TROUBLESHOOTING.md#environment-variables-and-your-sendgrid-api-key
    message = build_multiple_emails_personalized()
    sendgrid_client = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
    response = sendgrid_client.send(message=message)
    print(response.status_code)
    print(response.body)
    print(response.headers)

def send_hello_email():
    # Assumes you set your environment variable:
    # https://github.com/sendgrid/sendgrid-python/blob/HEAD/TROUBLESHOOTING.md#environment-variables-and-your-sendgrid-api-key
    message = build_hello_email()
    sendgrid_client = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
    response = sendgrid_client.send(message=message)
    print(response.status_code)
    print(response.body)
    print(response.headers)


def send_kitchen_sink():
    # Assumes you set your environment variable:
    # https://github.com/sendgrid/sendgrid-python/blob/HEAD/TROUBLESHOOTING.md#environment-variables-and-your-sendgrid-api-key
    message = build_kitchen_sink()
    sendgrid_client = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
    response = sendgrid_client.send(message=message)
    print(response.status_code)
    print(response.body)
    print(response.headers)
```

```
## this will actually send an email
# send_hello_email()

## this will send multiple emails
# send_multiple_emails_personalized()

## this will only send an email if you set SandBox Mode to False
# send_kitchen_sink()
```

Screen Shots:



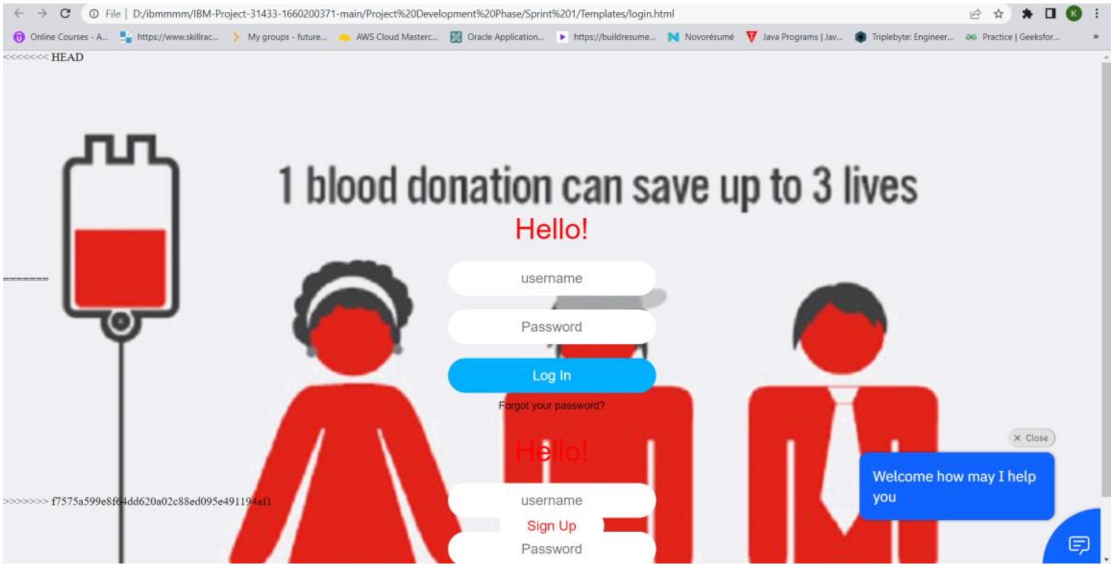
<<<<<<< HEAD

WELCOME TO PLASMA DONATION

=======

WELCOME TO PLASMA DONATION

>>>>>>> f7575a599e8f64dd620a02c88ed095e491194af1



<<<<<<< HEAD

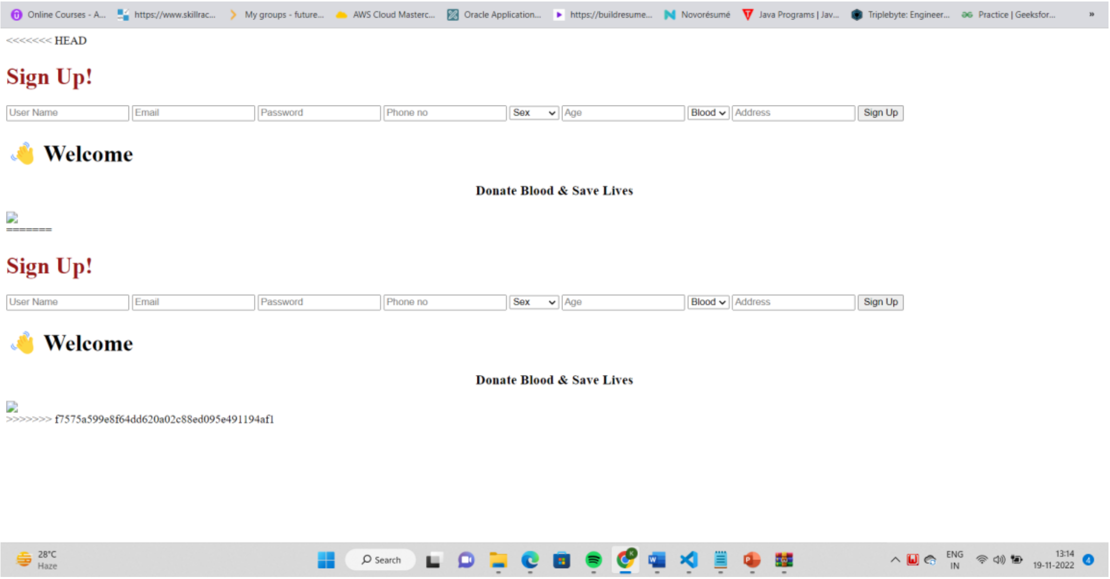
Hello!

=======

Hello!

>>>>>>> f7575a599e8f64dd620a02c88ed095e491194af1

Welcome how may I help you



<<<<<<< HEAD

Sign Up!

👋 Welcome

Donate Blood & Save Lives

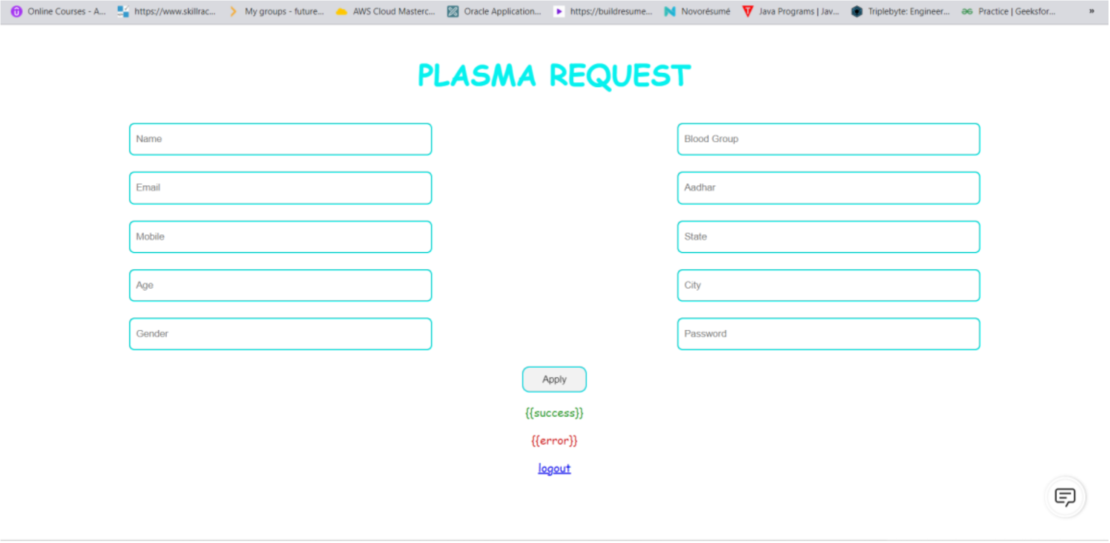
=======

Sign Up!

👋 Welcome

Donate Blood & Save Lives

>>>>>>> f7575a599e8f64dd620a02c88ed095e491194af1

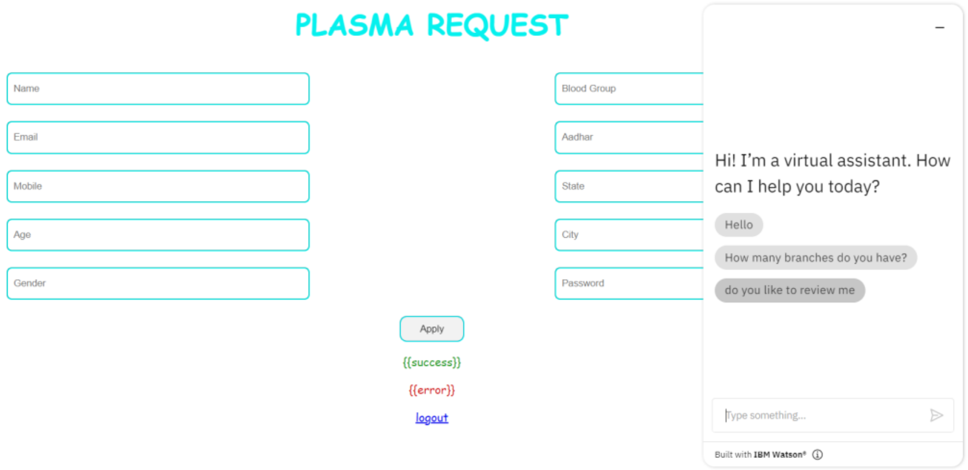


PLASMA REQUEST

{{success}}

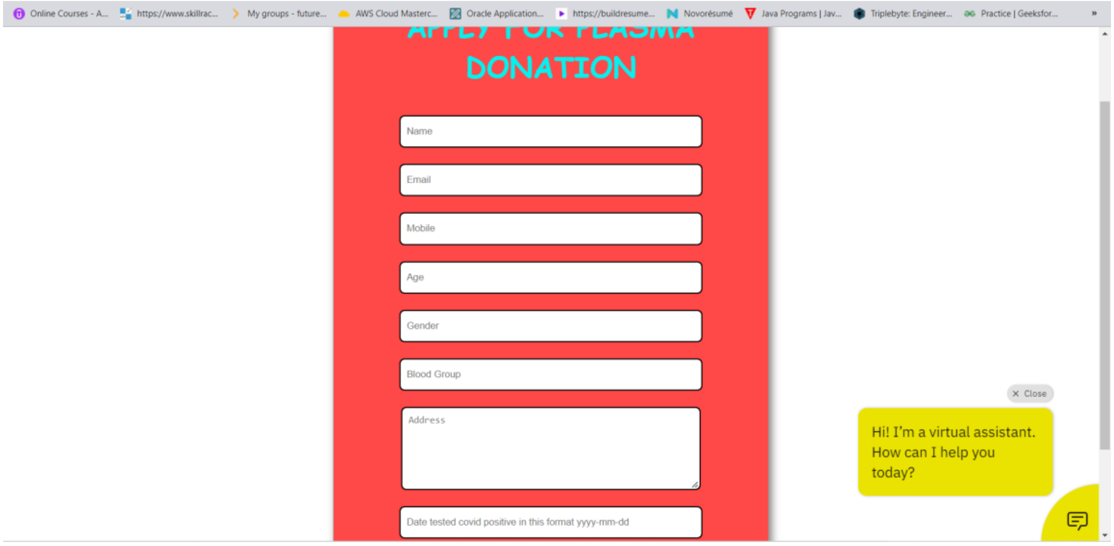
{{error}}

logout



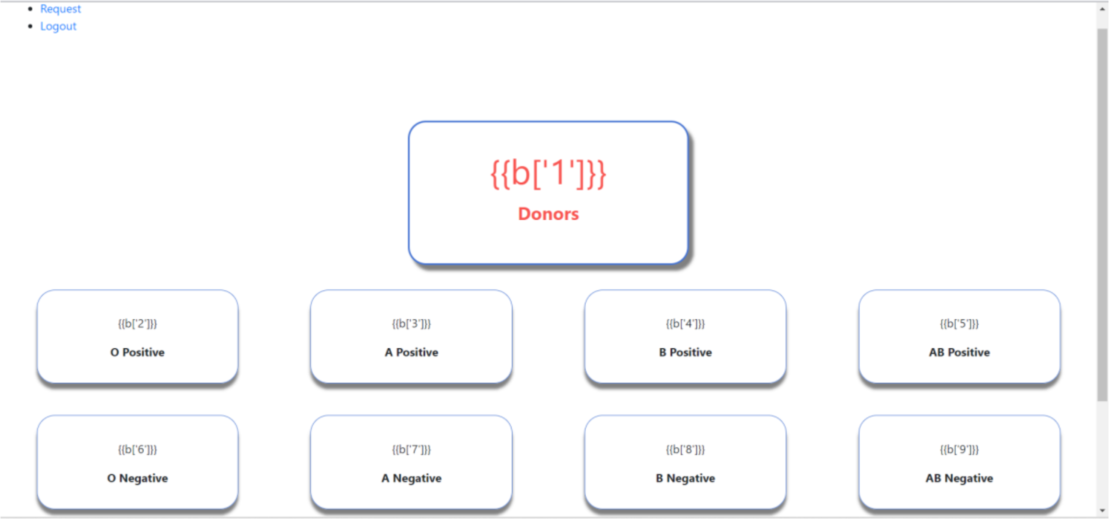
PLASMA REQUEST

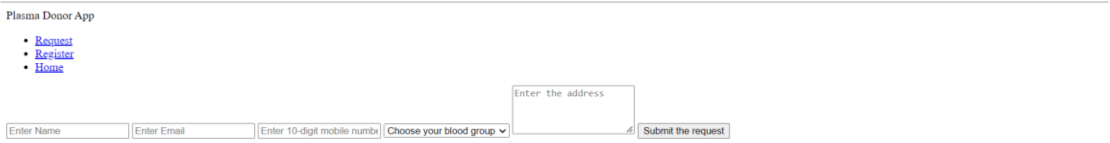
Hi! I’m a virtual assistant. How can I help you today?



APPLY FOR PLASMA DONATION

Hi! I’m a virtual assistant. How can I help you today?


Request
Logout
{{b['1']}}
Donors
{{b['2']}}
O Positive
{{b['3']}}
A Positive
{{b['4']}}
B Positive
{{b['5']}}
AB Positive
{{b['6']}}
O Negative
{{b['7']}}
A Negative
{{b['8']}}
B Negative
{{b['9']}}
AB Negative


Plasma Donor App
Request
Register
Home
Enter the address
Enter Name
Enter Email
Enter 10-digit mobile numb
Choose your blood group
Submit the request

GitHub Link:
https://github.com/IBM-EPBL/IBM-Project-24133-1659938472