Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	03 October 2022
Team ID	PNT2022TMID39311
Project Name	Project – Plasma Donor Application
Maximum Marks	4 Marks

Technical Architecture:

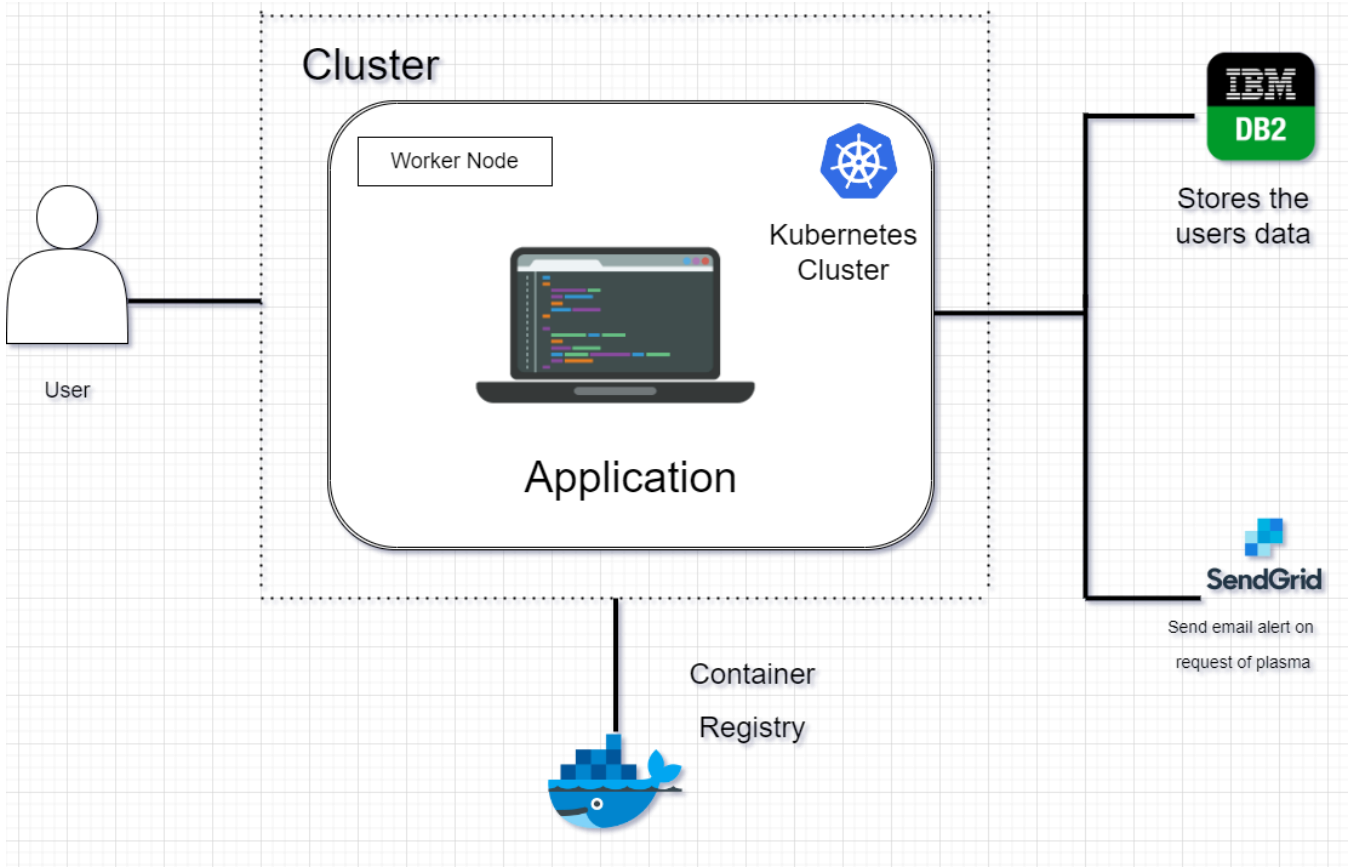

Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	The User will interact with the application in Web UI in their mobile or pc browser.	HTML, CSS, Bootstrap, JavaScript, Vue.js
2.	User Registration	User data will be collected using the form. The data will be Validated and Verified before pushing it to database.	JavaScript, Python – Flask, ibm_db
3.	Making the Plasma request	The recipient will enter their details in the request form including the required plasma type, location etc. The request will be verified and added to request queue.	Google Maps- Web API, Python – Flask, ibm_db
4.	Donors accepting the request and making donation appointment	Once the donor notified about the request, and if the donor wiling to donate their plasma, they can easily accept the request and make a appointment with the recipient.	SendGrid – Email API, Python – Flask, ibm_db, REST API
5.	Database	User profiles along with password will be maintained in separate table with in single database. The separate database will be used to store the donor's information and maintaining the current request and track the transactions.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	**External API-1** Google Maps – Web API	To Embed the google map in the web for location the recipient and donor's location	Google Maps – Web API
8.	**External API-2** Send Grid – Email API	To send an email to registered donors upon a new plasma request of their plasma type	Send Grid – Email API
9.	Containerization	Containerizing the entire application with the help of docker and register in the container registry	Docker, Docker hub, Docker Container Registry
10.	Infrastructure (Server / Cloud)	Application Deployment will be on Cloud Cloud Server Configuration: Separate Database Server Model	Kubernetes

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	• For Implementing the Web interface – Vue.js. • For backend of the application and WSGI – Flask Python. • For designing the user interface - Bootstrap • For Containerization – Docker • For testing purpose – Cypress	• Vue.js, Pinia, Vue Router. • Flask – Jinja, itsDangerous, Werkzeug, Click, Markupsafe. • Bootstrap, Popper.js • Cypress
2.	Security Implementations	• reCAPTCHA by google will be used to prevent the system from bots. • All the user provided details will be encrypted using an Encryption algorithm.	• reCAPTCHA by google. • SHA-256, AES, DSA, edDSA. • Cloud Fare.
3.	Scalable Architecture	The Entire application developed on basis of the **Microservices** architecture which will provides easy scaling, prevent single point failure, fault tolerance and CI/CD etc.	• Docker. • Docker Hub. • Docker Registry. • Kubernetes.
4.	Availability	Just to increase the availability of the application at any time load balancer will be introduced into the server architecture and to maintain the distributed server across the many regions to make sure the peoples across the region may get the service availability.	• Load balancers – Haproxy for TCP and NGINX for http. • IBM Cloud.
5.	Performance	To improve the performance of the application, caching mechanism will be implemented and the contents for the application will provided by the CDN will helps to improve loading time of the web	• Caching mechanism. • CDN – Content Delivery Network.