Central Blood Management System

GROUP - 14

Mentor : Ashutosh
Instructor - Dr. Indranil Saha

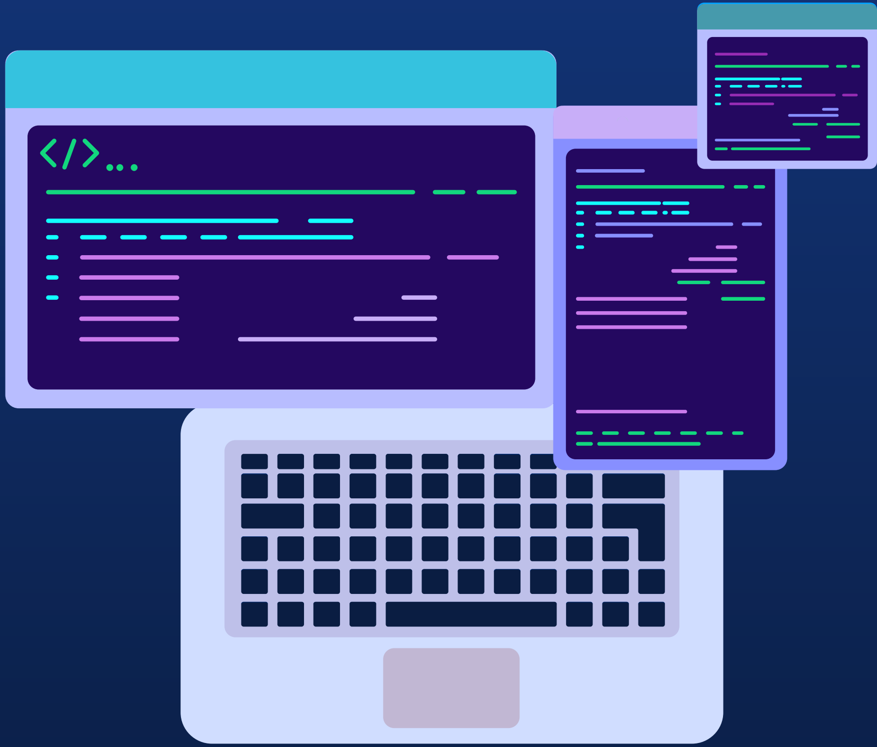

TABLE OF CONTENTS

01

NEED FOR SOFTWARE

02

MAJOR REQUIREMENTS

03

KEY DESIGN

04

DEMONSTRATION

05

FUTURE PLAN

06

LESSON LEARNT

01

NEED FOR THIS SOFTWARE / POTENTIAL USER

Need for this software

- The purpose of our app is to provide a centralized blood database online.
- It allows blood banks from various parts of the country to register themselves and be part of a centralized blood database.
- Helps the user to view the available blood units of different blood groups and blood components of the blood bank.
- App supports adding blood camp events and these can be viewed by all the users.
- It can be used to issue blood donation certificates.
- Interaction app between donors, blood banks and patients.

Potential User

- Patients- Anyone who is in need of a particular blood component of a blood group can look up if there are available blood units in any of the blood banks nearby her/his area.

- Donors - Anyone who is willing to donate blood can look up in the app if there are any blood camp events happening around his/her neighborhood or can directly go to nearby blood bank to donate blood.

- Blood Banks - To update blood component amount and to look for volunteer donors. It can also issue donation certificates for these donors.

02

MAJOR REQUIREMENTS

EXTERNAL INTERFACE REQUIREMENTS

- HARDWARE INTERFACES
  - SERVER SIDE - The web application will be hosted on a web server which is listening on the web standard port, port 8080.
  - CLIENT SIDE - There are no specific requirements on the client side.

- SOFTWARE INTERFACES
  - SERVER SIDE - Django, being a web framework, needs a web server in order to operate.
  - CLIENT SIDE - A client should have a modern browser which supports JAVAScript and HTML5.

FUNCTIONAL REQUIREMENTS

- BLOOD BANK
  - Can register themselves on the website and can be part of a centralized blood donation system.
  - Can register blood donation camps events on the website so that register donor can get notified.
  - Can issue appreciation certificate for the donor who have donated blood.
  - Get a list of volunteer donors that can be contacted by the Blood Banks.

- PATIENTS
  - Can look up the website and keep track of available units of the required type and components.

FUNCTIONAL REQUIREMENTS

DONOR

- Can look up the website to see if there are any blood donation camps happening.
- Can register on the website so that they can get updates about blood donation events.
- Can select blood banks which can contact the user.
- Can download the appreciation certificate issued by blood bank.

NON-FUNCTIONAL REQUIREMENTS

- PERFORMANCE REQUIREMENTS
  - Regular virus protection must be done.
  - Database must be archived regularly.

- SAFETY AND SECURITY REQUIREMENTS
  - Access to data in the database is restricted and the data is secured through multiple layers of protection.
  - If there is extensive damage to a wide portion then the recovery method restores a past copy of the database.

03

Key design and implementation decisions

- Architecture : Model-View-Controller (MVC)
    - Used for developing user-interface applications
- Backend : Django
    - Is complete in itself
    - Easy to learn and use
    - Uses MVC like architecture
- Database : SQLite
    - LightWeight
    - Does not require installation
    - Easy to use with Django
    - Can be easily shifted to other database like PostgreSQL.

04

Demonstration

05

Future Development Plans

- Our vision for this application is to expand the interface to include additional features and become a one-stop shop for blood donors, blood organisations, and consumers.
- We would also like to add an interface to schedule or request blood units from a blood bank. Thus, individuals can be assured of receiving the necessary blood units when they visit a blood bank.
- It could be possible to register for the delivery of blood units to different locations. The user must initially request blood units. Upon acceptance, the recipient could choose to have the blood units delivered to his residence.

06

Lessons Learnt

- We utilised HTML, CSS, and JavaScript for the project's frontend. These languages have the benefit of being supported by all browsers and being compatible with any underlying framework. In addition, their featherweight weight makes the system quicker. We utilised Bootstrap and Google Fonts for styling. Utilising these frameworks on the frontend has the advantage of facilitating the creation of responsive designs much more quickly than without them. Additionally, they offer cross-browser compatibility.
- Learnt to use Github, Django, SQLite. Reasons for using this database language in place of others like MySQL and PostgreSQL are that SQLite is lightweight and thus provides better performance, does not require installation, is reliable and portable. Also, it works very well with our chosen Django Framework.
- Learn various phases during a software development like various documentation, implementation, testing and fixing if some bug appear etc. Learn how to complete a project on or before deadline with a team.

Q/A

Thank You