A

Project Report

On

"Blood Bank Website"

Submitted in partial fulfilment of the requirements for the award of degree

Bachelor of Computer Application

From

Hemchand Yadav Vishwavidyalaya, Durg (C.G.)

Year: 2023 – 2024

RUNGTA EDUCATIONAL FOUNDATION
BHILAI-CHHATTISGARH

Guided By :

Mr. Vinay Kumar Sahu

(Assistant Professor)

Submitted By :

Tushar Rathore

Roll No : 223410130060

Submitted to

GD Rungta College of Science and Technology, Bhilai

Affiliated to Hemchand Yadav Vishwavidyalaya, Durg (C.G.)

RUNGTA EDUCATIONAL FOUNDATION
BHILAI, CHHATTISGARH

GD RUNGTA COLLEGE OF SCIENCE & TECHNOLOGY

(Managed by Rungta Educational Foundation, Bhilai, Chhattisgarh)
Rungta Educational Campus, Kohka-Kurud Road, Bhilai, Dist: Durg, Chhattisgarh - 490024
Ph.: 0788 6666 666, Fax: 0788 2286 480, Email: gdrcst@rungta.ac.in, Website: www.rungta.ac.in

CERTIFICATE OF APPROVAL

This is to certify that the Project work entitled "**Blood Bank Website**" is carried out by **Mr. Tushar Rathore** , A student of BCA-III year at **GD Rungta College of Science and Technology, Bhilai** is hereby approved as a credible work in the discipline of Computer Science and Application for the award of degree of **Bachelor of Computer Application** during the year 2023-2024 from **Hemchand Yadav Vishwavidyalaya,Durg (C.G.).**

HOD

(Dr. Jyoti Upadhyay)

GDRCST,BHILAI

RUNGTA EDUCATIONAL FOUNDATION
BHILAI, CHHATTISGARH

GD RUNGTA COLLEGE OF SCIENCE & TECHNOLOGY

(Managed by Rungta Educational Foundation, Bhilai, Chhattisgarh)
Rungta Educational Campus, Kohka-Kurud Road, Bhilai, Dist: Durg, Chhattisgarh - 490024
Ph.: 0788 6666 666, Fax: 0788 2286 480, Email: gdrcst@rungta.ac.in, Website: www.rungta.ac.in

CERTIFICATE

This is to certify that the Project work entitled "**Blood Bank Website**" Submitted to the **GDRCST** by **Mr. Tushar Rathore** , **Roll No. 223410130060**, of partial fulfilment for the requirements relating to nature and standard of award of Bachelor of Computer Application degree by, **Hemchand Yadav Vishwavidyalaya, Durg (C.G.)** for the academic year 2023-2024.

This project work has been carried out under my guidance.

Mr. Vinay Kumar Sahu

Assistant Professor

RUNGTA EDUCATIONAL FOUNDATION
BHILAI, CHHATTISGARH

GD RUNGTA COLLEGE OF SCIENCE & TECHNOLOGY

(Managed by Rungta Educational Foundation, Bhilai, Chhattisgarh)
Rungta Educational Campus, Kohka-Kurud Road, Bhilai, Dist: Durg, Chhattisgarh - 490024
Ph.: 0788 6666 666, Fax: 0788 2286 480, Email: gdrcst@rungta.ac.in, Website: www.rungta.ac.in

CERTIFICATE OF EVALUATION

This is to certify that the Project work entitled "**Blood Bank Website**" is carried out by **Mr. Tushar Rathore**, a student of BCA-III year at **GD Rungta College of Science and Technology, Bhilai**, after proper evaluation and examination, is hereby approved as a sincere work in the discipline of Computer Science and Application and is done in a satisfactory manner for its acceptance as a requisite for the award of degree of **Bachelor of Computer** Application during the year 2023-2024 from **Hemchand Yadav Vishwavidyalaya, Durg (C.G.)**

Internal Examiner **External Examiner**

DECLARATION

This is to certify that the project work entitled **"Blood Bank Website"** which is submitted by me in partial fulfillment for the award of degree **of "Bachelor of Computer Application, (G.D Rungta college of Science and Technology),** comprises the original work carried out by me.

I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full for the award of any other degree or diploma in this institute or in other university.

Place: Bhilai

Date :

Name – Tushar Rathore

Roll No.-223410130060

ACKNOWLEDGEMENT

With immense pleasure, I would like to present this report on the project assignment of "**Blood Bank Website**". I offer my sincere thanks to Principal **Dr. Neema S Balan Ma'am**, G.D RUNGTA COLLEGE OF SCIENCE AND TECHNOLOGY, Bhilai, for giving us an opportunity to work our college. I would like to thank all faculties for helping us directly or indirectly in the completion of the project.

Next, I would like to thanks **H.O.D**, Department of Computer Science **Dr. Jyoti Upadhyay Mam** for giving us proper environment to work and develop in our laboratory. We would also like to thank the computer staff members and project guide **Mr. Vinay Kumar Sahu Sir** have helped me to collect the required data for our project.

I would like to acknowledge my parents, family members, friends, and all those persons who helped me directly or indirectly in the successful completion the project work.

Name – Tushar Rathore Roll No.- 223410130060

MAIN REPORT

TABLE OF CONTENTS

INTRODUCTION

In a world where finding the right blood type can be a challenge, Life Flow emerges as your friendly guide, simplifying the process of connecting with blood donors. Imagine a platform Life Flow, a Blood Bank project – where discovering the perfect blood match becomes easy, allowing you to engage directly with a willing donor in a seamless and direct connection.

Our goal at Life Flow is to transform the way we manage blood donation records. Tracking donors becomes effortless with our user-friendly system, allowing you to swiftly identify individuals with the precise blood type you require. No more complications or intermediaries – just swift access to the information you need, ensuring a quick and direct connection with a donor when time is of the essence.

Life Flow envisions a world where the search for blood is not only simplified but also personalized. Picture this: discovering the right blood type and engaging in a direct chat with a donor eager to extend a helping hand. We're here to bridge the gap between those in need and those ready to donate, making the entire process easy, friendly, and accessible for everyone.

Life Flow seamlessly integrates HTML, CSS, JavaScript, Python, Django, and SQLite, each playing a distinct role in creating a secure, user-friendly platform for connection, donation, and life-saving initiatives. HTML forms the foundation, CSS provides visual clarity, JavaScript enables interactivity, Python orchestrates operations, Django simplifies development, and SQLite manages data efficiently, forming a collaborative technological ecosystem for positive social impact.

it is not just a platform it's a vision realized, a bridge built between those who need life-saving blood and those with the heart to donate. As we conclude this introduction, we invite you to become a part of our community, where simplicity, ease, and direct connections define the landscape of compassionate healthcare. Together, on Life Flow, let's celebrate the power of connections that save lives and contribute to a world where compassion knows no bounds.

OBJECTIVE

The objective of the application is to facilitate direct communication between blood donors and those in need, providing a user-friendly platform for convenient blood searches and efficient donor responses, thereby enhancing the accessibility and effectiveness of the blood donation process.

Existing System: At present no software is offering direct communication between blood seeker and blood donor. It becomes difficult to get blood of the same type immediately in times of emergency. manually keeping the accounts is also a tedious & risky job & maintaining those accounts in ledgers for a long period is also very difficult. it is hectic to manage and maintain the files there is a chance of damage to files if the data is stored in the files for a long duration of time. It is difficult to keep track of the record of the donor & receiver he has donated or received the blood at the last time.

Proposed System: The proposed system (Life Flow - Blood Bank Website) is designed to help blood seekers fulfill their need for blood by directly communicating with the Donor. The proposed system gives a procedural approach on how to bridge the gap between recipients and donors. This Application will provide a common ground to all two parties and will ensure the fulfillment of the demand for Blood requested by the Recipient and/or Blood Bank. The features of the proposed system are ease of donation system along with providing user-friendly interfaces, no need to maintain any manual register and form as all data of donation will be digitally organized and easily accessible.

Efficient Blood Matching and Distribution: The Blood Bank Project streamlines the blood donation and distribution process, reducing time and effort for both donors and medical institutions.

Modernize Blood Donation Procedures: Shifting from traditional methods, the Blood Bank Project embraces online platforms, minimizing paperwork, and automating blood donation processes.

Mobile Accessibility: Acknowledging the importance of mobile access, the Blood Bank Project is developing a dedicated mobile app, providing users with on-the-go access to blood donation services.

Continuous Improvement: Committed to ongoing enhancements, the Blood Bank Project adapts to evolving healthcare needs for a dynamic and user-friendly experience.

Enhanced User Experience: Prioritizing a positive and user-friendly experience, the Blood Bank Project aims to be a preferred platform for both blood donors and medical institutions.

Regional and Global Reach: The Blood Bank Project extends its reach beyond local boundaries, connecting blood donors with recipients globally, and enabling healthcare institutions to find suitable blood supplies worldwide.

Community Building: Fostering a sense of community, the Blood Bank Project encourages interaction and collaboration among blood donors, recipients, and healthcare professionals.

Transparent Blood Donation Records: Ensuring transparency, the Blood Bank Project maintains clear and detailed records of blood donation transactions, providing comprehensive information for both donors and healthcare institutions.

Accessibility Features: Incorporating accessibility features, the Blood Bank Project ensures inclusivity for users with diverse needs, creating an environment where everyone can actively participate in the blood donation process.

SCOPE OF THE PROJECT

The Life Flow - Blood Bank Website project aims to revolutionize and streamline the process of blood donation by providing a user-friendly platform that facilitates direct communication between blood donors and those in need. The scope of the project encompasses various modules and features to enhance the accessibility, efficiency, and overall experience of the blood donation process.

User Registration and Authentication: Allow users, including donors and blood seekers, to register and authenticate their identities securely. Implement robust authentication mechanisms to ensure the privacy and security of user data.

Donor Profile Management: Enable donors to create and manage comprehensive profiles, including personal information, blood type, and donation history. Provide a user-friendly interface for donors to update and maintain their profiles easily.

Search Filters: Include advanced filters in the blood search module, allowing users to refine their searches based on blood type, location, and other relevant criteria. Enhance the search functionality to provide accurate and tailored results for blood seekers.

Blood Search and Matching: Implement a dynamic search module for blood seekers to find donors with matching blood types in real-time. enable donors to respond promptly to blood requests, creating a seamless connection between those in need and willing donors.

Communication Module: Develop a secure and direct communication channel that facilitates real-time communication between blood donors and seekers. Implement features for chat and coordination to ensure effective communication during the blood donation process.

User-Friendly Interface: Design a simple and intuitive interface for both donors and blood seekers, enhancing the overall user experience. Prioritize user accessibility and inclusivity in the interface design.

Real-Time Blood Search with Blood Type Filters: Create a dynamic search feature that allows blood seekers to find donors in real-time based on blood type, location, and other relevant criteria. ensure that the search functionality is efficient and responsive to meet urgent blood donation needs.

Direct Communication Channel: Establish a built-in communication module that enables direct and secure communication between blood donors and seekers. implement features that foster transparent and reliable communication throughout the donation process.

Email Request Feature: Integrate an email request feature that allows blood seekers to send requests directly to potential donors.

Enhance communication and response time by providing an additional avenue for connecting donors and seekers.

The scope of the Life Flow - Blood Bank Website project is comprehensive, aiming to address the current challenges in the blood donation process by leveraging technology to create a more efficient, user-friendly, and direct communication platform for donors and blood seekers.

THEORETICAL BACKGROUND OF THE PROJECT

Blood Bank Website project is grounded in addressing the challenges and limitations of the current blood donation system while leveraging key principles of user-centric design, information management, and communication technologies. The project draws upon several theoretical concepts to conceptualize and develop a robust platform for facilitating direct communication between blood donors and those in need.

Frontend:

HTML (HyperText Markup Language): HTML provides the structural foundation of the website's pages, defining the content and layout using elements like headers, sections, forms, and lists.

CSS (Cascading Style Sheets): CSS styles the website's appearance, controlling fonts, colors, spacing, layouts, and responsiveness to different screen sizes. It ensures a visually appealing and consistent presentation across various devices.

JavaScript: JavaScript is incorporated to enhance the interactivity of the website. It enables dynamic content updates, form validations, and asynchronous interactions, contributing to a more engaging and responsive user experience.

Bootstrap: Bootstrap, a popular front-end framework, is utilized to streamline the design and development process. It provides a set of pre-designed components and styles, facilitating the creation of a responsive and aesthetically pleasing interface. Bootstrap ensures consistency in design elements and responsiveness across different devices.

Backend:

Python: Python powers the backend logic and database interactions. Its versatile and readable syntax contributes to efficient development, ensuring the functionality of the web application.

Django: Django, a high-level Python web framework, streamlines development by handling URL routing, request processing, database interaction, templating, and security. It follows a structured approach for building web applications.

SQLite: SQLite serves as the backend database, offering a lightweight, file-based solution ideal for offline use. It stores essential data such as login details, signup information, and user numbers.

By integrating HTML, CSS, JavaScript, Bootstrap, Python, Django, and SQLite, the Life Flow - Blood Bank Website combines a visually appealing and interactive frontend with a robust and efficient backend. The inclusion of JavaScript and Bootstrap enhances the user experience, providing dynamic features and responsive design elements to facilitate seamless communication between blood donors and seekers.

DEFINITION OF PROBLEM

In the current blood donation landscape, several challenges hinder the efficiency and accessibility of the process. The absence of a dedicated platform for direct communication between blood donors and seekers creates obstacles during critical situations. Manual record-keeping for blood donors is both cumbersome and prone to errors, leading to delays and potential mismatches.

The lack of a centralized system results in difficulties for blood seekers to promptly find donors with the required blood type. Coordinating and communicating with potential donors in real-time is a complex task, often requiring intermediaries and leading to delays in securing blood when urgently needed.

The existing system relies on traditional, manual methods for maintaining records and connecting donors with recipients. This approach not only poses challenges in data management but also limits the speed and efficiency of the blood donation process. Furthermore, the absence of a direct communication channel between donors and seekers adds layers of complexity, impacting the overall effectiveness of the blood donation system.

The need for a comprehensive solution becomes evident in addressing these issues, streamlining the blood donation process, and fostering direct communication between those in need and willing donors. The Life Flow - Blood Bank Website project seeks to bridge these gaps by leveraging technology to create a user-friendly, efficient, and direct platform that enhances the overall experience for both blood donors and seekers. Through the integration of modern web technologies, the project aims to overcome the limitations of the current system, providing a seamless and responsive solution to the challenges faced in the realm of blood donation.

SYSTEM ANALYSIS & DESIGN

The System Analysis and Design phase of the Life Flow - Blood Bank Website project involves a comprehensive examination of the current blood donation system and the subsequent design of an innovative, user-centric platform to address existing challenges. This phase encompasses various key steps and methodologies to ensure the successful development and implementation of the proposed solution.

1. Requirement Gathering: Conducting thorough interviews and surveys to gather requirements from potential users, including blood donors and seekers. Identifying essential features, functionalities, and user expectations to form the foundation of the system specifications.

2. Feasibility Study: Assessing the technical, operational, and economic feasibility of the proposed system. Evaluating potential risks, benefits, and constraints associated with the implementation of the Life Flow - Blood Bank Website.

3. System Design: Creating a detailed system design that outlines the architecture, components, and interactions of the proposed platform. Designing the database schema to efficiently store and retrieve donor and seeker information using SQLite. Defining the user interface, incorporating HTML, CSS, and Bootstrap for a visually appealing and responsive design.

4. Backend Development: Utilizing Python as the primary backend programming language for its versatility and readability. Integrating Django, a high-level web framework, to handle URL routing, request processing, templating, and security. Implementing robust backend logic to facilitate seamless communication between the frontend and database.

5. Frontend Development: Employing HTML to structure the website's pages and define content. Using CSS to style the interface, ensuring a consistent and visually pleasing presentation. Incorporating JavaScript for dynamic content updates, form validations, and enhanced user interactivity. Leveraging Bootstrap to streamline frontend development and ensure responsiveness across various devices.

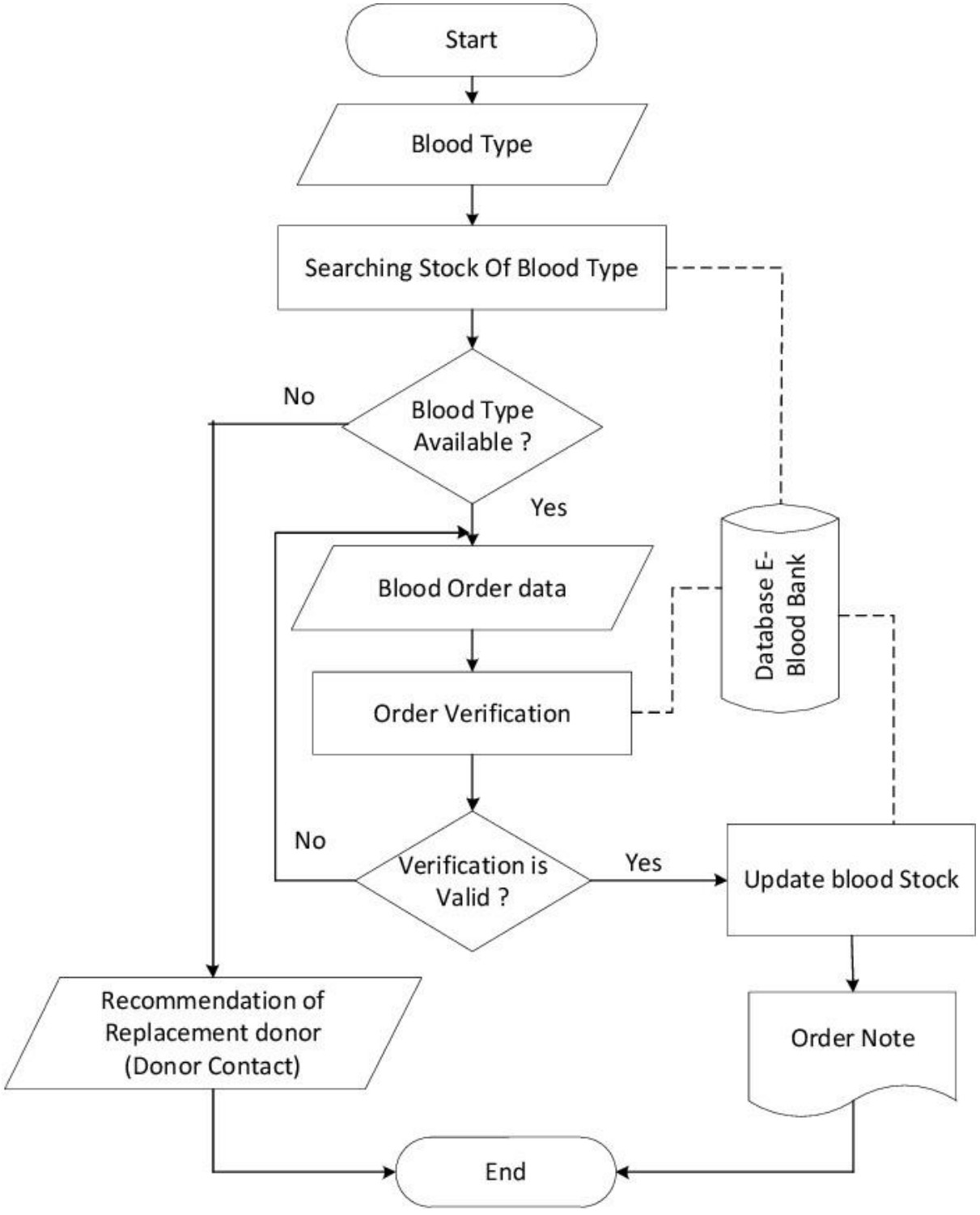
6. Database Integration: Integrating SQLite as the backend database, considering its lightweight and file-based nature suitable for offline use. Implementing database interactions to store and retrieve essential user-related information, such as login details and donation history.

7. Testing: Conducting rigorous testing to identify and rectify any bugs or issues in the system. Ensuring the functionality, security, and responsiveness of the platform across different browsers and devices.

8. Implementation: Deploying the Life Flow - Blood Bank Website for public use after successful testing and refinement. Providing necessary documentation and support for users and administrators.

9. Maintenance and Optimization: Implementing regular maintenance routines to address any emerging issues and ensure continued system efficiency. optimizing the platform based on user feedback and evolving requirements.

The System Analysis and Design phase is crucial for laying the groundwork for the development of the Life Flow - Blood Bank Website. It ensures a systematic and well-planned approach to address the defined problem and create an effective, user-friendly solution for the blood donation process.



System Design Chart of Blood Bank System

SYSTEM PLANNING (PERT CHART)

1 Week: Research -

Initiated project with a comprehensive research phase.

Explored existing blood donation systems and user needs.

Gathered insights to inform subsequent development stages.

2 Week: Analysis & Requirements Gathering -

Conducted detailed analysis of project requirements.

Identified specific functionalities, user roles, and system specifications.

Outcomes informed the foundation for upcoming development.

1 Week: Defining Project Scope -

Clearly outlined project boundaries and objectives.

Ensured a comprehensive understanding of inclusions and exclusions.

2 Weeks: UI/UX Design -

Worked on design considerations for an intuitive interface.

Focused on creating a visually appealing user experience.

2 Weeks: Tech Selection -

Choose the optimal technology stack.

Selected HTML, CSS, and Bootstrap for the frontend and Django for the backend.

Considered dependencies and compatibility with Django.

4 Weeks: Backend Development -

Commenced backend infrastructure development.

Ensured efficient data management and system functionality.

3 Weeks: Frontend Development -

Began front-end development for a user-friendly interface.

Prioritized simplicity and responsiveness.

1 Weeks: Authentication -

Dedicated two weeks to implement a secure user authentication system.

2 Weeks: Security Measures -

Implemented security measures to enhance overall application security.

Safeguarded user data and ensured system integrity.

2 Weeks: Testing -

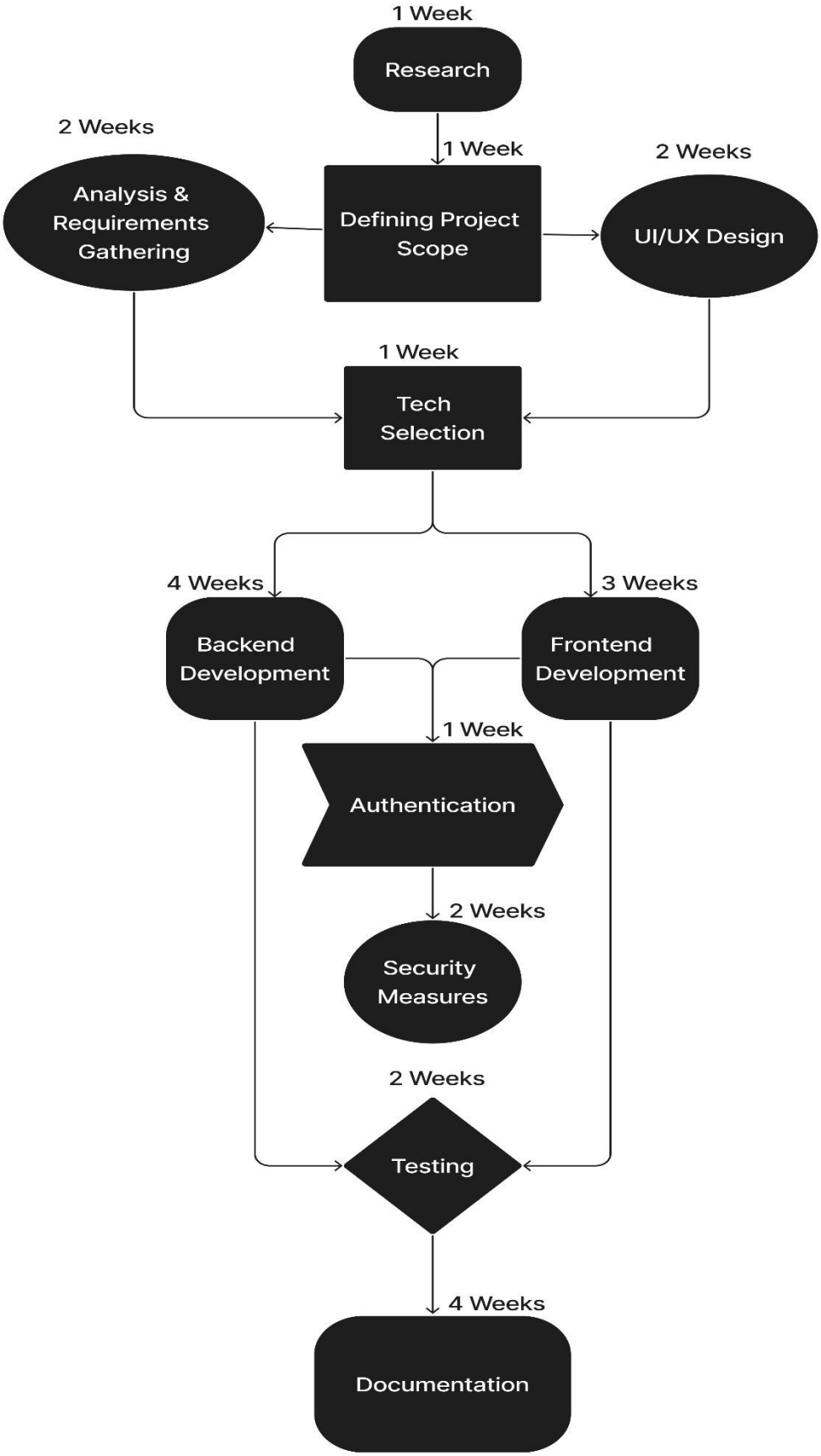
Conducted rigorous testing of all components and functionalities.

Ensured reliability and correctness of the application.

4 Weeks: Documentation -

Spanned four weeks for comprehensive documentation.

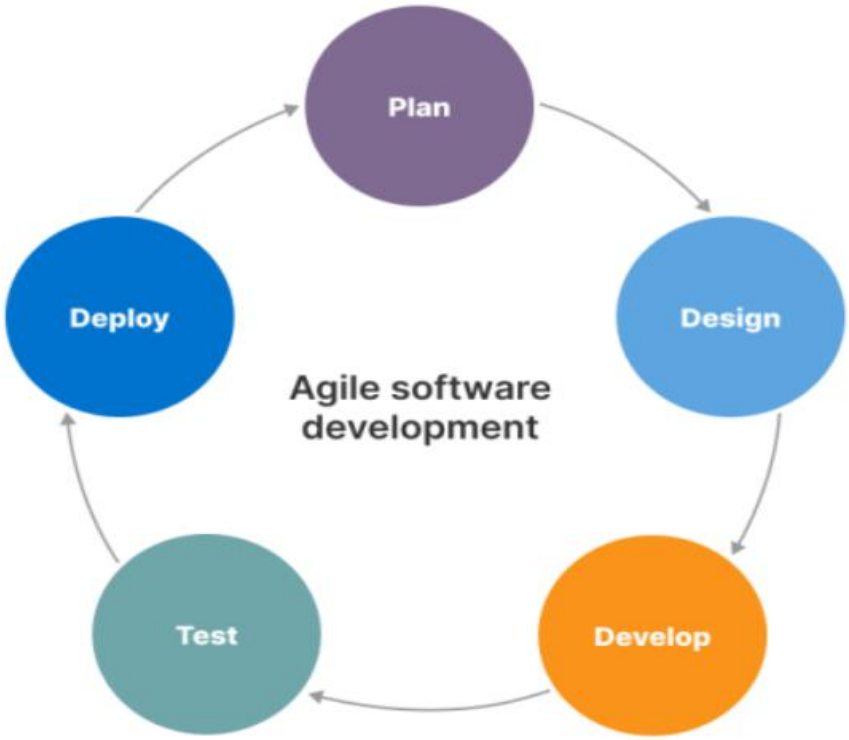
Included technical documentation, user guides, and instructional material.


1 Week
Research
1 Week
2 Weeks
Analysis & Requirements Gathering
Defining Project Scope
2 Weeks
UI/UX Design
1 Week
Tech Selection
4 Weeks
3 Weeks
Backend Development
Frontend Development
1 Week
Authentication
2 Weeks
Security Measures
2 Weeks
Testing
4 Weeks
Documentation

METHODOLOGY ADOPTED

Our project, Life Flow - Blood Bank Website, follows an Agile development approach to ensure flexibility, adaptability, and continuous improvement throughout the software development lifecycle. The Agile methodology is chosen for its iterative and collaborative nature, allowing us to respond effectively to changing requirements and deliver a high-quality product. Here's an overview of the methodology adopted:

Agile Development: Agile development involves breaking down the project into small, manageable iterations. For the blood bank application, each iteration can focus on specific features like user registration, donor profile management, or communication modules.



Collaborative Work Environment: Focused on open communication and collaboration among team members, including developers, designers, and stakeholders. conducted regular meetings to discuss progress, challenges, and adjustments to be made.

User-Centric Design: Prioritized user feedback and involvement throughout the development process. conducted usability testing and incorporated user suggestions to enhance the user experience.

Flexible and Adaptive: Adapted to changing requirements and priorities, allowing for continuous adjustments based on feedback and evolving circumstances. maintained a flexible approach to accommodate unforeseen challenges and opportunities.

Continuous Testing: Integrated testing is an ongoing process within each development cycle. conducted regular testing of new features to identify and address issues promptly.

Documentation Throughout: Maintained comprehensive documentation at each stage of development. documented user stories, technical specifications, and changes made during iterations.

By adopting an Agile approach, we aim to ensure that Life Flow is not only responsive to the needs of blood seekers and donors but also that it can easily adapt to future enhancements and changes in the healthcare landscape. This Agile methodology empowers the team to deliver a resilient and user-focused blood bank application.

SYSTEM IMPLEMENTATION

System implementation marks the pivotal phase in the software development lifecycle where the meticulously planned system design is translated into a functional reality. This stage involves the actual coding and development of both the backend and frontend components, transforming the architectural blueprints into a working software system.

Django Setup:

➢ Install Django using the command pip install django.

➢ Create a Django project using django-admin startproject BloodBank.

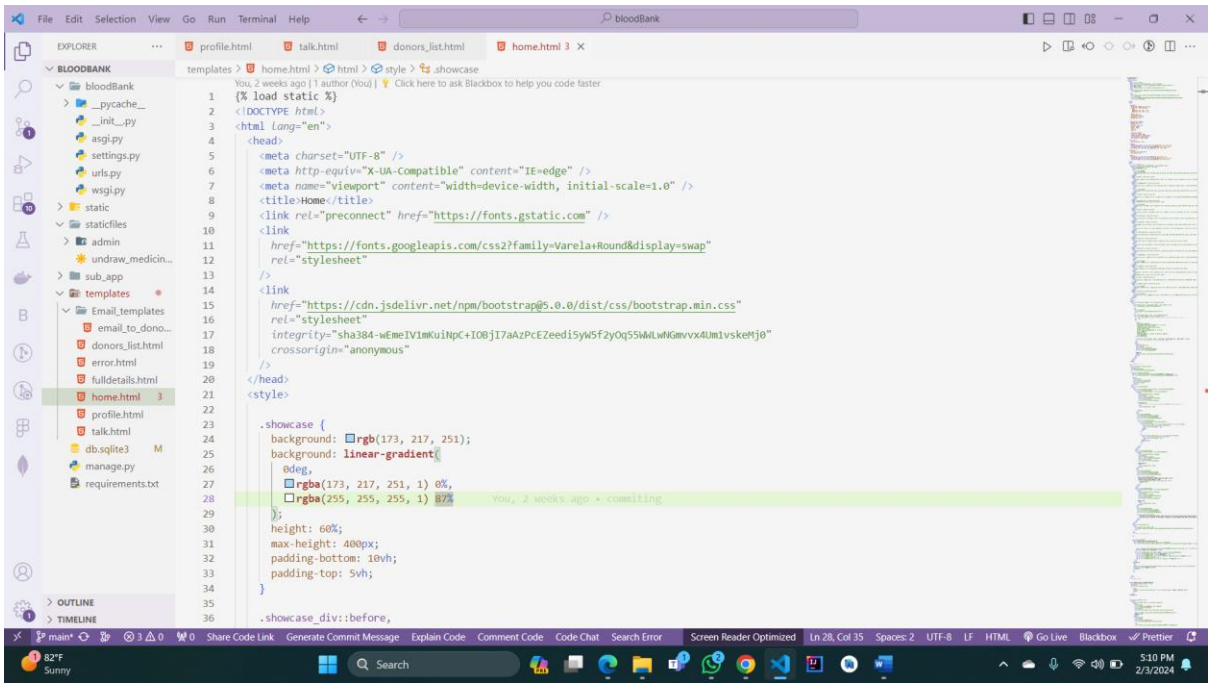
➢ Navigate to the project directory using cd BloodBank.

Front-End Development:

• Design HTML templates for the website's pages (index, search, category, signup, login) and organize them in the 'templates' directory within the Django app.

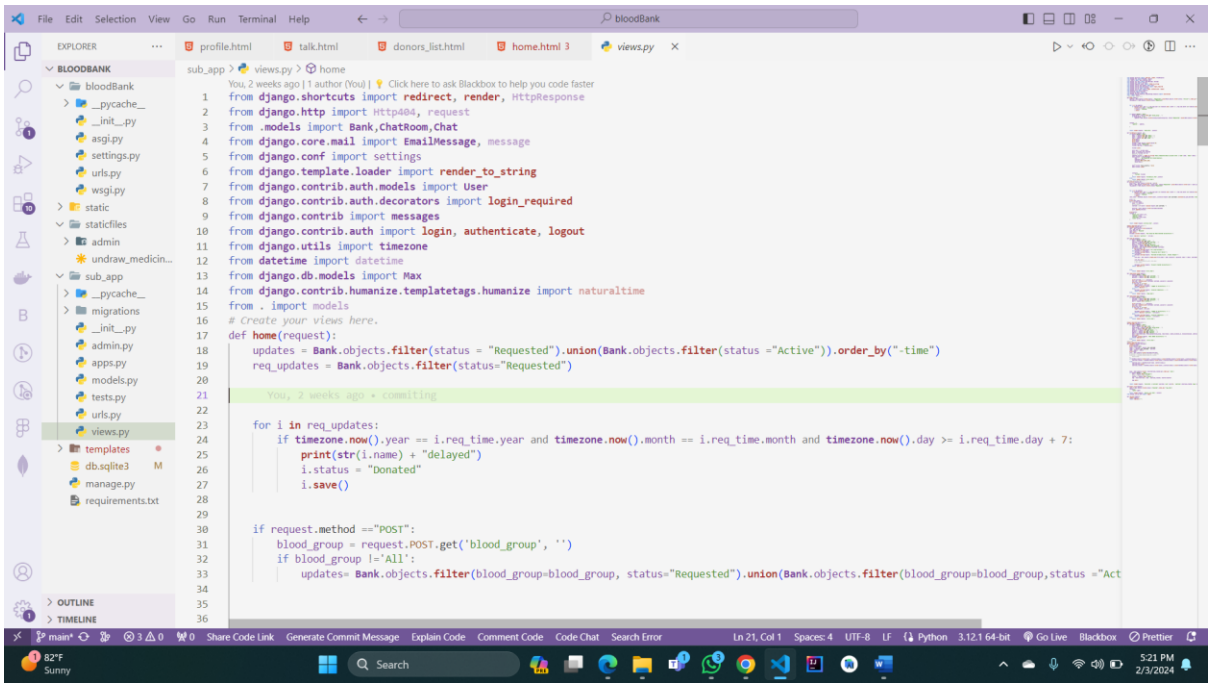
• Using style tag to initiate CSS codes for styling.

• Link CSS tags in HTML templates for a visually appealing layout.



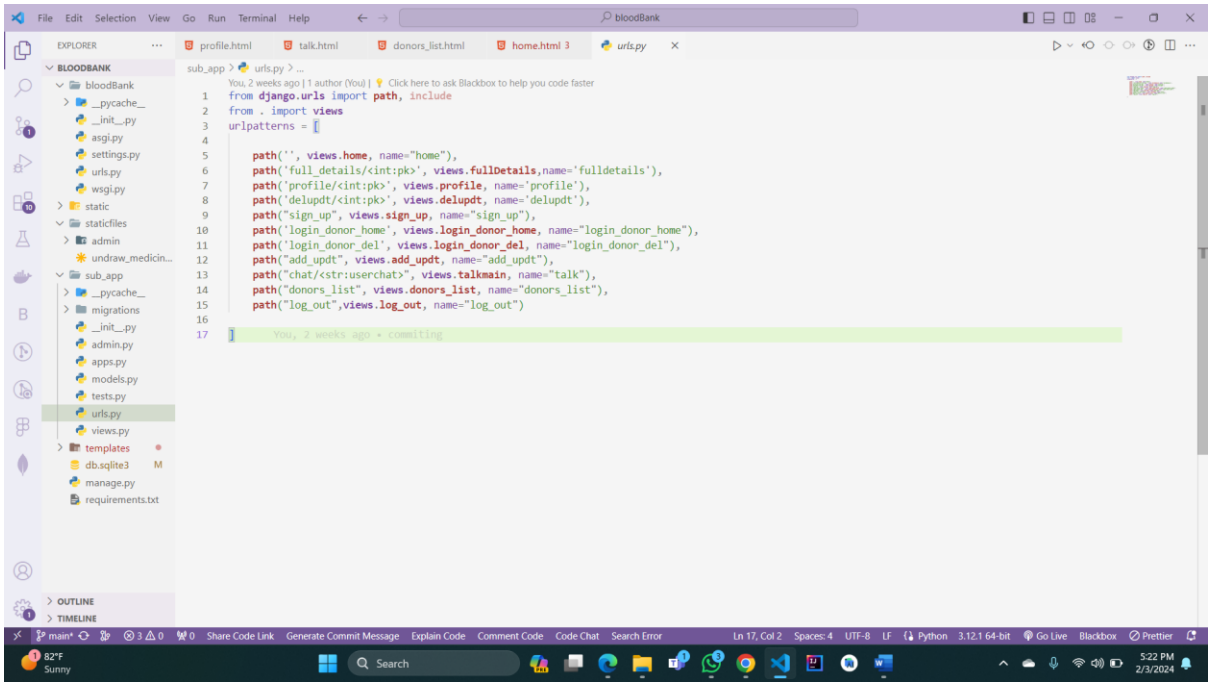
Back-End Development:

➢ Create views in the 'views.py' file for index, admin, signup, login implementing functions to handle HTTP requests and render HTML templates.



```python
from django.shortcuts import redirect, render, HttpResponse
from django.http import Http404, request
from .models import Bank,ChatRoom,Chat
from django.core.mail import EmailMessage, message
from django.conf import settings
from django.template.loader import render_to_string
from django.contrib.auth.models import User
from django.contrib.auth.decorators import login_required
from django.contrib import messages
from django.contrib.auth import login, authenticate, logout
from django.utils import timezone
from datetime import datetime
from django.db.models import Max
from django.contrib.humanize.templatetags.humanize import naturaltime
from . import models
# Create your views here.
def home(request):
    updates = Bank.objects.filter(status = "Requested").union(Bank.objects.filter(status ="Active")).order_by("-time")
    req_updates = Bank.objects.filter(status="Requested")

    for i in req_updates:
        if timezone.now().year == i.req_time.year and timezone.now().month == i.req_time.month and timezone.now().day >= i.req_time.day + 7:
            print(str(i.name) + "delayed")
            i.status = "Donated"
            i.save()


    if request.method =="POST":
        blood_group = request.POST.get('blood_group', '')
        if blood_group !='All':
            updates= Bank.objects.filter(blood_group=blood_group, status="Requested").union(Bank.objects.filter(blood_group=blood_group,status ="Act
```

➢ Configure URL patterns in 'urls.py' for each view

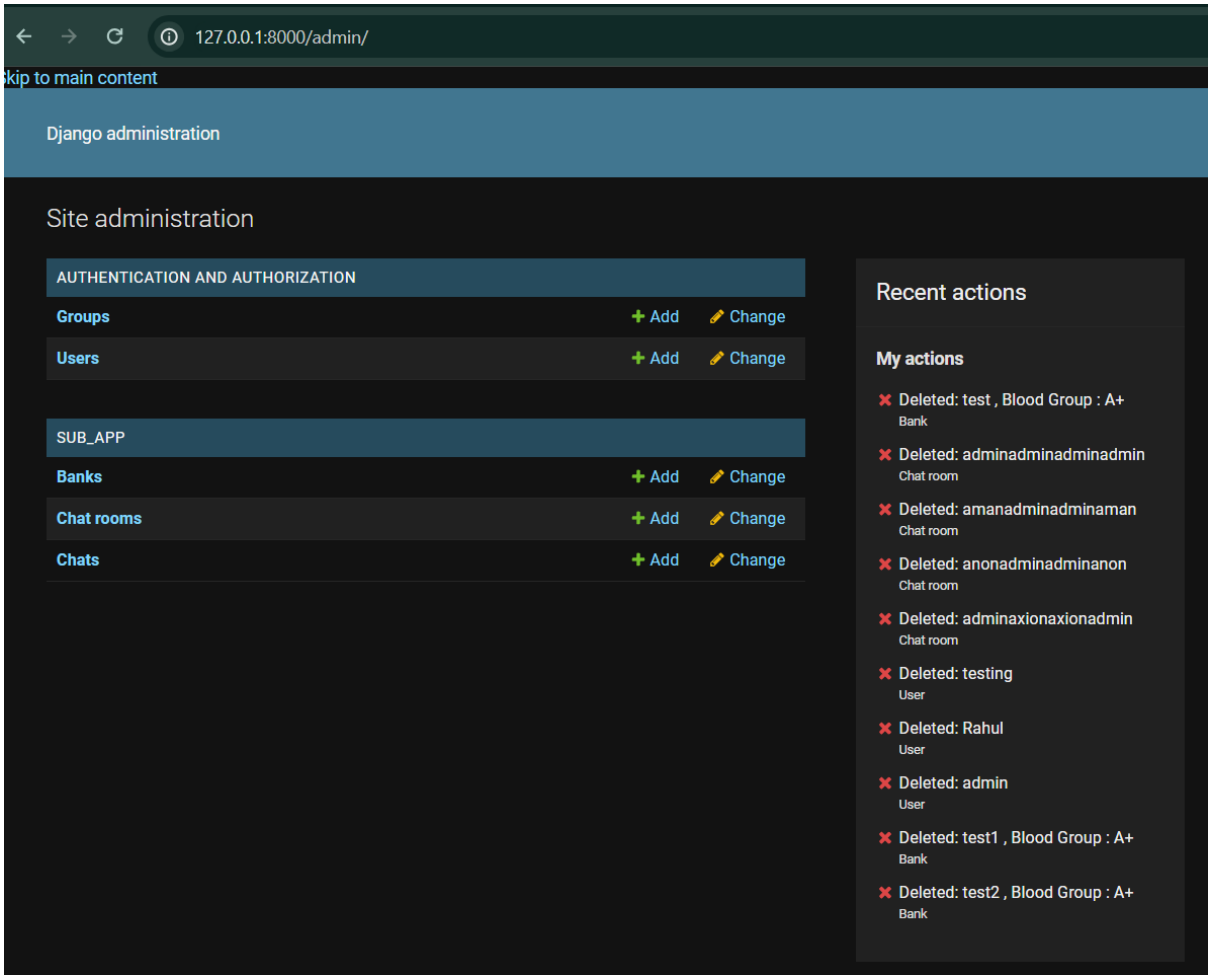


```python
from django.urls import path, include
from . import views
urlpatterns = [

    path('', views.home, name="home"),
    path('full_details/<int:pk>', views.fullDetails,name='fulldetails'),
    path('profile/<int:pk>', views.profile, name='profile'),
    path('delupdt/<int:pk>', views.delupdt, name='delupdt'),
    path("sign_up", views.sign_up, name="sign_up"),
    path('login_donor_home', views.login_donor_home, name="login_donor_home"),
    path('login_donor_del', views.login_donor_del, name="login_donor_del"),
    path("add_updt", views.add_updt, name="add_updt"),
    path("chat/<str:userchat>", views.talkmain, name="talk"),
    path("donors_list", views.donors_list, name="donors_list"),
    path("log_out",views.log_out, name="log_out")

]
```

Integration:

• Connect the front-end and back-end using Django templates to render dynamic content.

• Implement user authentication for secure interactions.



➢ Write unit tests for views and models to ensure the functionality is correct.

➢ Run tests using the command python manage.py test to identify and rectify any bugs or issues.

Deployment:

➢ Deploy the website following the hosting provider's instructions. ➢ Configure servers, set up databases, and ensure the platform operates smoothly.

❖ Hardware and Software Requirements

Hardware:

➢ Processor: 11th Gen Intel(R) Core (TM) i3/ AMD Ryzen 3 and above

➢ RAM: 4 GB and above Page | 20

Software:

➢ Operating System: 64-bit operating system

➢ Front-End**: HTML, CSS**

➢ Front-End Framework: **Bootstrap**

➢ Back-End: **Python (Django)**

➢ Database: **SQLite**

IDE Used:

Utilize **Visual Studio Code** or any preferred integrated development environment for coding and project management.

User Requirement:

Ensure that end-users have a web browser and internet connection for accessing the news website.

Python Packages Required:

- asgiref == 3.7.2 Django == 5.0.1
- django-widget-tweaks == 1.4.8
- Pillow == 10.1.0
- sqlparse == 0.3.1

To install python packages use this command in terminal: **pip install package-name**

for example: **pip install Django== 5.0.1**

After installing all the packages successfully in your environment, Now you have to migrate files to establish your local database use this command –

- python manage.py makemigrations & python manage.py migrate

SYSTEM MAINTENANCE AND EVALUATION

After crafting our blood bank system, it's crucial to enter the next phase – maintaining and evaluating the system. This ongoing process is vital for keeping the platform functional, secure, and aligned with the ever-changing needs of users in the realm of blood donation.

Regular Updates and Patch Management:

We consistently roll out updates to introduce new features, implement security patches, and enhance overall performance. This proactive approach keeps the blood bank system up-to-date and resilient in the face of emerging challenges.

Security Measures:

Security is a top priority during maintenance. Regular audits, vulnerability assessments, and strict adherence to industry best practices are in place to protect user data and thwart potential threats to the blood bank system.

Performance Optimization:

Continuous monitoring and optimization efforts are focused on improving the system's performance. This involves refining code, optimizing database queries, and ensuring a responsive and efficient experience for users involved in blood donation.

User Feedback Integration:

We maintain an open channel for user feedback, incorporating suggestions and addressing reported issues. This collaborative engagement ensures that the blood bank system stays in tune with user expectations and preferences.

Blood Inventory Management:

Regular assessments of blood inventory and diversity are conducted. Analyzing user engagement patterns, identifying popular blood types, and adjusting the inventory strategy to meet diverse needs in blood donation.

Scalability and Adaptability:

Adapting to changing user traffic and data volume is a continuous process. We assess scalability, anticipate growth trends, and implement measures to seamlessly scale the blood bank system as needed.

Technology Stack Updates:

Periodic evaluations of the technology stack are performed, with updates implemented to leverage the latest advancements. This ensures compatibility with new devices, browsers, and emerging technologies in the field of blood donation.

Performance Analytics:

Robust analytics tools are employed to assess the system's performance. Monitoring metrics such as response times, user engagement, and traffic patterns provides insights for ongoing optimization and improvement.

Backup and Disaster Recovery:

A robust backup and disaster recovery strategy is maintained to safeguard against data loss or system disruptions. Regular testing of recovery procedures ensures the system's resilience in unforeseen circumstances.

User Education and Support: Continuous efforts are invested in user education, providing resources, tutorials, and support channels. This empowers users to make the most of the blood bank system, contributing to a positive and user-friendly experience in the realm of blood donation.

COST AND BENEFIT ANALYSIS

The development of a Blood-Bank website involves a comprehensive cost analysis, encompassing various aspects of the project lifecycle. The following breakdown provides insights into the key components contributing to the overall cost:

Personnel Costs:

- Development Team: Includes front-end and back-end developers, UI/UX designers, and quality assurance professionals, contributing significantly to personnel costs.

Technology and Infrastructure:

- Software Licenses: Expenses related to acquiring licenses for development tools, frameworks, and any proprietary software.

Design and User Experience:

- Graphic Designers: Costs for hiring graphic designers to create visually appealing and user-friendly interfaces.

External Services:

- Blood Database Services: Costs associated with external services for accessing and maintaining a comprehensive blood donor database.

Documentation:

- Documentation: Expenses related to creating comprehensive documentation for internal and external stakeholders.

Maintenance and Support:

- Technical Support: Costs associated with providing ongoing technical support to users.

- System Maintenance: Budget for routine maintenance, updates, and addressing issues post-launch.

Miscellaneous:

- Miscellaneous Expenses: Any other incidental costs not explicitly covered by the categories above.

Benefit Analysis:

Improved Blood Donation Engagement:

Enhance user engagement by providing a user-friendly platform, streamlined donation processes, and personalized donor experiences.

Wider Donor Base:

Attract a broader donor base through effective outreach, awareness campaigns, and a dynamic blood donation strategy catering to various community needs.

Monetization Opportunities:

Explore revenue streams through partnerships, sponsored blood donation drives, and community engagement initiatives.

Brand Visibility and Credibility:

Establish a trusted brand by providing accurate, timely, and relevant information on blood donation events, campaigns, and urgent needs.

Data-Driven Decision Making:

Empower decision-making with analytics insights into donor behavior, preferences, and blood donation patterns.

Competitive Edge:

Gain a competitive edge in the blood donation domain with feature-rich offerings and staying abreast of technological trends.

User Satisfaction and Retention:

Ensure high levels of donor satisfaction through an excellent user experience, personalization, and responsive communication.

Adaptability to Market Changes:

Quickly adapt to changes in blood donation regulations, emerging technologies, and donor preferences with Agile development practices.

Strategic Partnerships:

Establish strategic partnerships with healthcare providers, community organizations, or technology partners to enhance blood donation diversity and availability.

Cost-Benefit Ratio:

- Positive: If the benefits outweigh the costs, indicating a potentially successful and impactful blood bank system.

- Negative: If the costs exceed the benefits, necessitating a reevaluation of the project's viability and potential adjustments to maximize impact.

DETAILED LIFE CYCLE OF PROJECT

ER DIAGRAM

An Entity-Relationship (ER) diagram is a visual representation of the relationships among entities in a database. It serves as a blueprint for database design, depicting how different entities are connected and the nature of those connections. In the Life Flow Blood Bank System, the Entity-Relationship (ER) Diagram visually represents the key entities, relationships, and attributes within the database.

Entities:

a. User Entity:

Attributes: User_ID (Primary Key), Username, Password, Email, Contact Number, Account Status.

b. Donor Entity (Subtype of User):

Attributes: Donor_ID (Primary Key), Blood Type, Donation History, Eligibility Status.

c. Recipient Entity (Subtype of User):

Attributes: Recipient_ID (Primary Key), Blood Type, Medical History, Urgency Level.

d. Administrator Entity:

Attributes: Admin_ID (Primary Key), Admin Username, Admin Password, Admin Privileges.

Relationships:

a. Donation Relationship:

Relates Donor Entity to Recipient Entity.

Cardinality: Many-to-Many.

Attributes: Donation_ID (Primary Key), Donation Date, Quantity.

b. User-Administrator Relationship:

Establishes the administrative control over user accounts.

Cardinality: One-to-Many.

Attributes:

a. User Attributes:

User_ID as a primary key uniquely identifies each user. Username, Password, Email, Contact Number, Account Status.

b. Donor Attributes:

Donor_ID as a primary key uniquely identifies each donor. Blood Type, Donation History, Eligibility Status.

c. Recipient Attributes:

Recipient_ID as a primary key uniquely identifies each recipient. Blood Type, Medical History, Urgency Level.

d. Administrator Attributes:

Admin_ID as a primary key uniquely identifies each administrator. Admin Username, Admin Password, Admin Privileges.

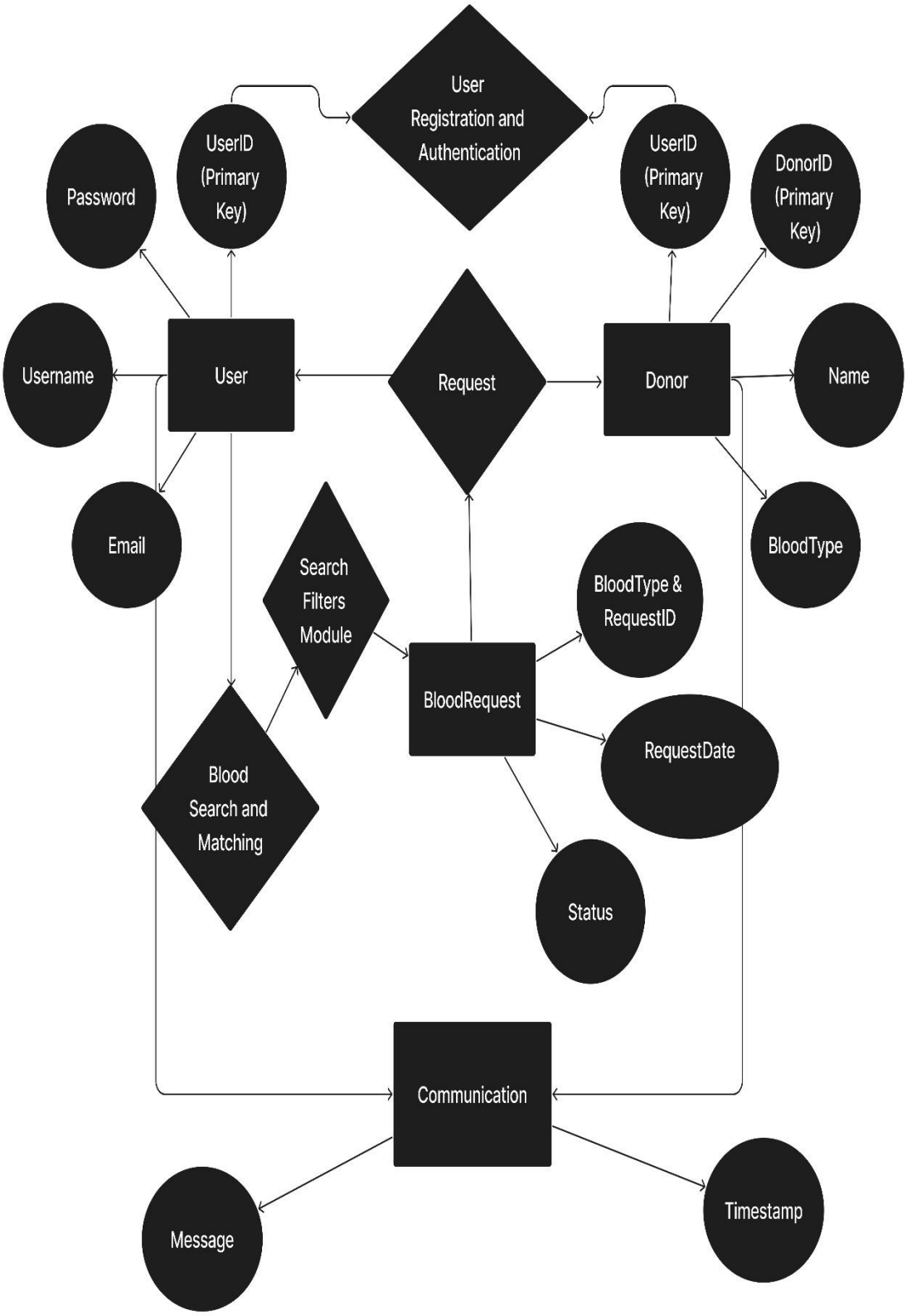
e. Donation Attributes:

Donation_ID as a primary key uniquely identifies each donation. Donation Date, Quantity.

Primary and Foreign Keys:

User_ID, Donor_ID, Recipient_ID, and Admin_ID serve as primary keys.

Donation_ID acts as a primary key in the Donation Entity and a foreign key linking to the User Entity.


User Registration and Authentication
UserID (Primary Key)
Password
UserID (Primary Key)
DonorID (Primary Key)
Username
User
Request
Donor
Name
Email
Search Filters Module
BloodType & RequestID
BloodType
BloodRequest
Blood Search and Matching
RequestDate
Status
Communication
Message
Timestamp

DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of how data flows within a system, illustrating the processes, data sources, data destinations, and data storage. It provides a clear and concise view of the system's data movement and processing. In the context of the Life Flow - Blood Bank Website project, we can create a DFD to showcase the flow of information within the application.

Level 0 DFD:

The Level 0 DFD represents the highest level of abstraction and illustrates the system as a single process, showcasing the external entities interacting with the system.

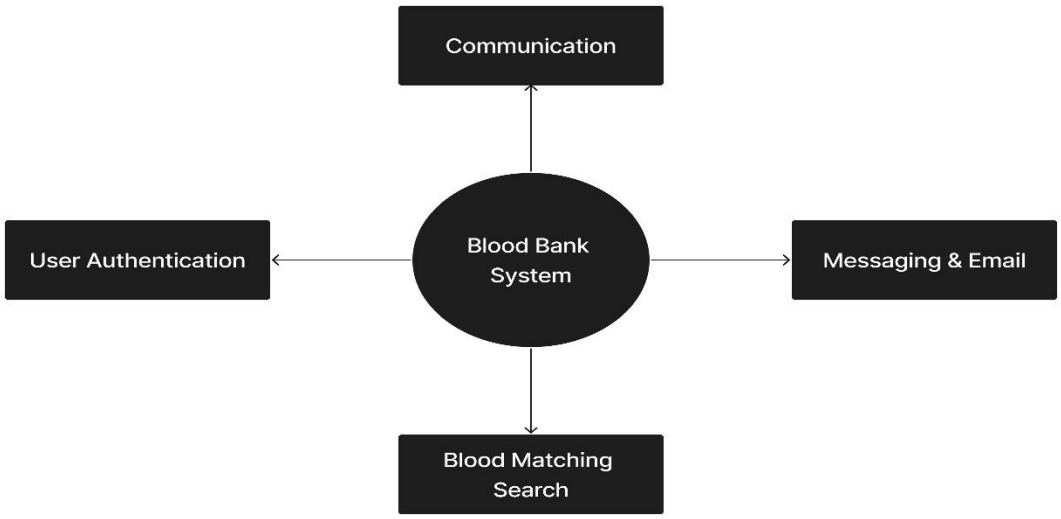
Processes:

Blood Bank System: Represents the overall system that includes various modules and functionalities.

External Entities:

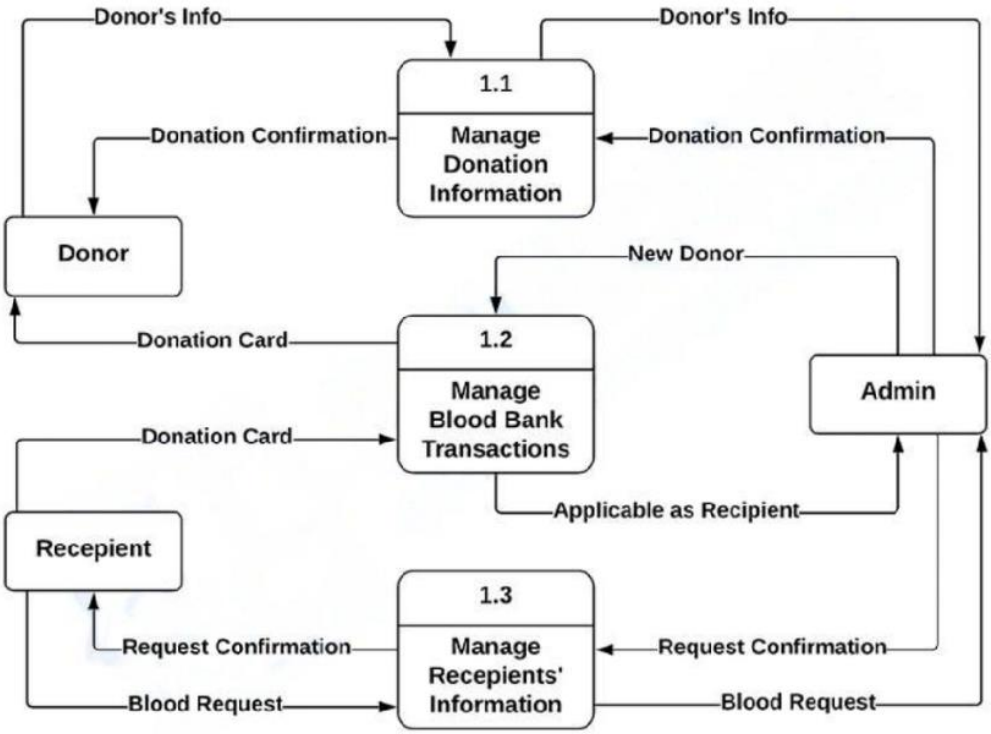
User: Represents individuals interacting with the system, including donors and blood seekers.

Data Flows: User Input and Output: Represents the flow of information between the User and the Blood Bank System.



Level 1 DFD:

At Level 1, the DFD breaks down the system into major processes:



DATA FLOW DIAGRAM LEVEL 1

A Level 2 Data Flow Diagram (DFD) provides a more detailed view of specific processes within the Blood Bank System. Each process identified in the Level 1 DFD is decomposed into subprocesses, allowing for a more granular understanding of the data flow within the system.

Processes:

User Registration and Authentication Process:

Manages user registration and authentication.

Subprocesses include validation and account creation.

Donor Profile Management Process:

Handles creation and modification of donor profiles.

Subprocesses involve user profile management.

Blood Search and Matching Process:

Implements search module for blood seekers.

Subprocesses include filtering and matching algorithms.

Communication Module Process:

Facilitates direct communication between donors and seekers.

Subprocesses involve message handling and coordination.

External Entity:

User (Donor/Blood Seeker):

Represents external entities interacting with specific processes.

Data Stores:

User Database, Donor Database, Blood Request Database, Communication Database:

Store user information, donor profiles, blood request details, and communication records.

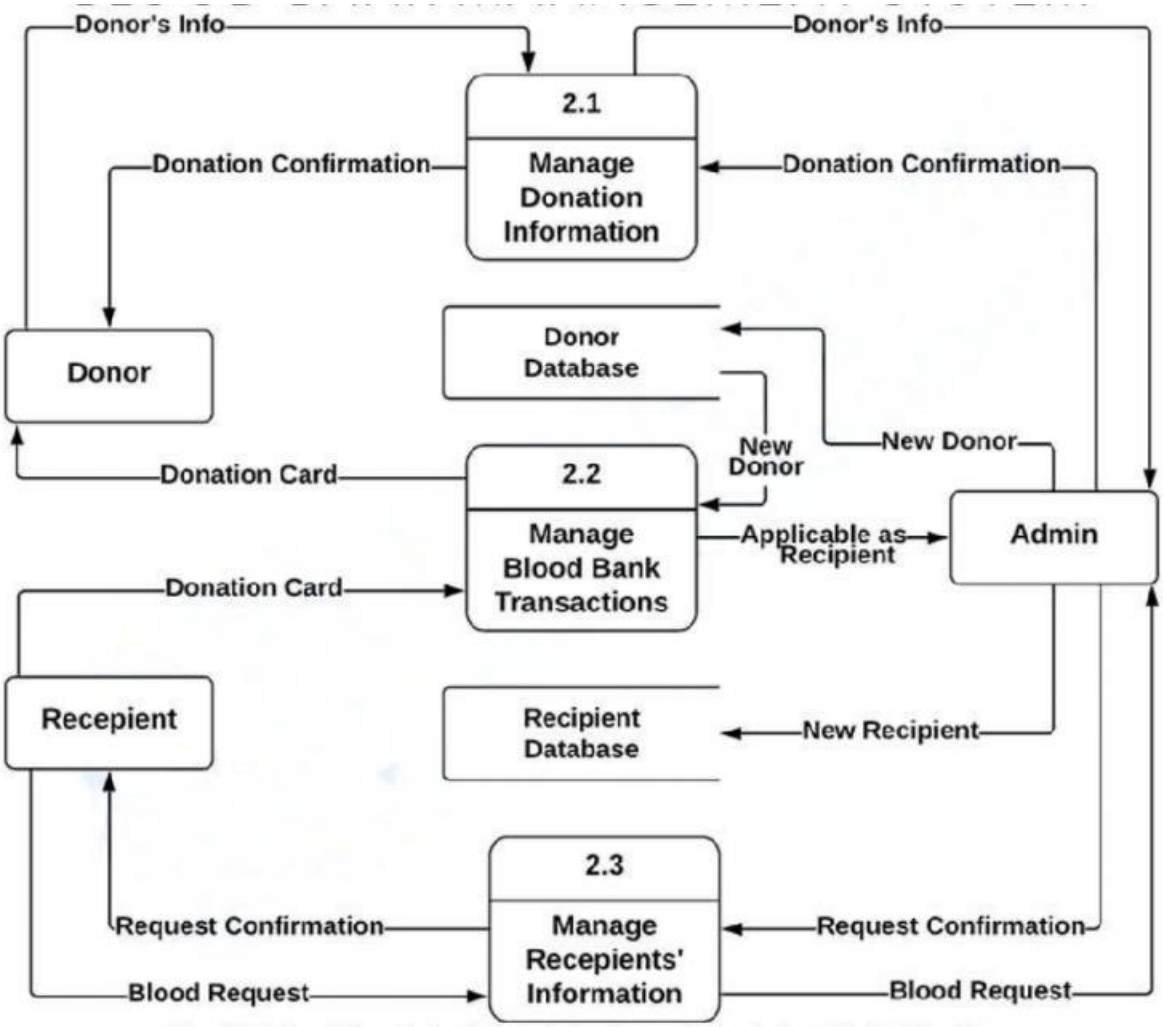
Data Flows:

User Input and Output:

Represents information exchange between users and system processes.

Database Updates:

Shows flow of updated information between processes and data stores.

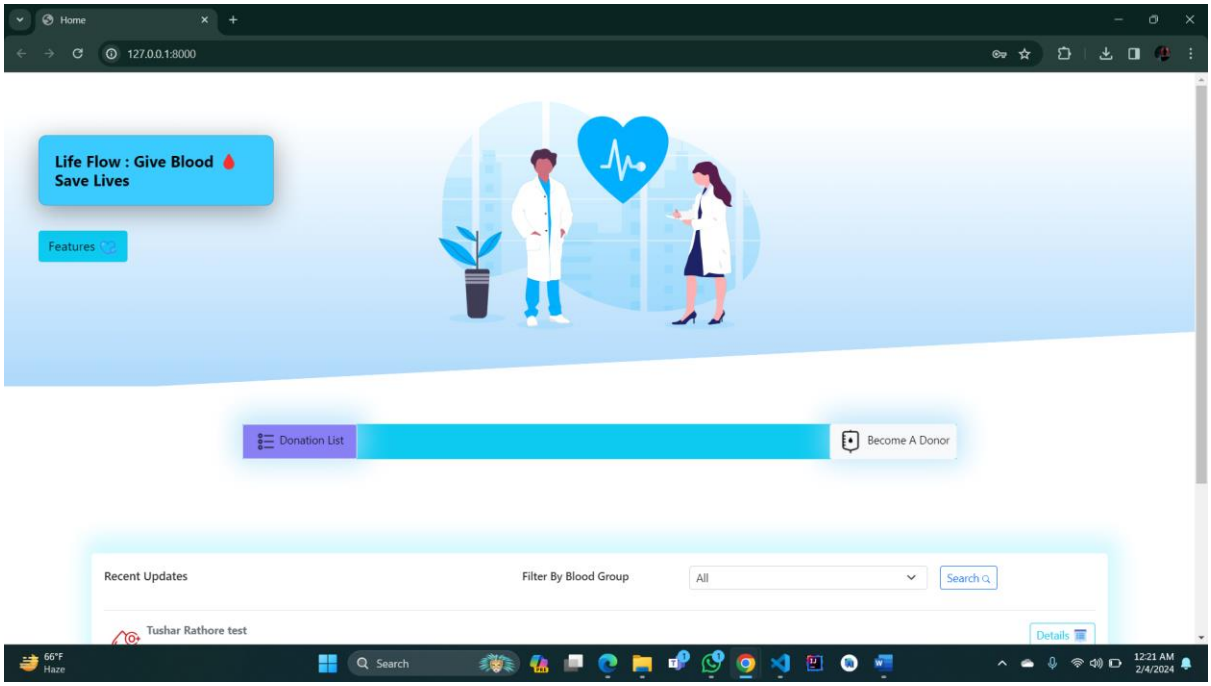


DATA FLOW DIAGRAM LEVEL 2

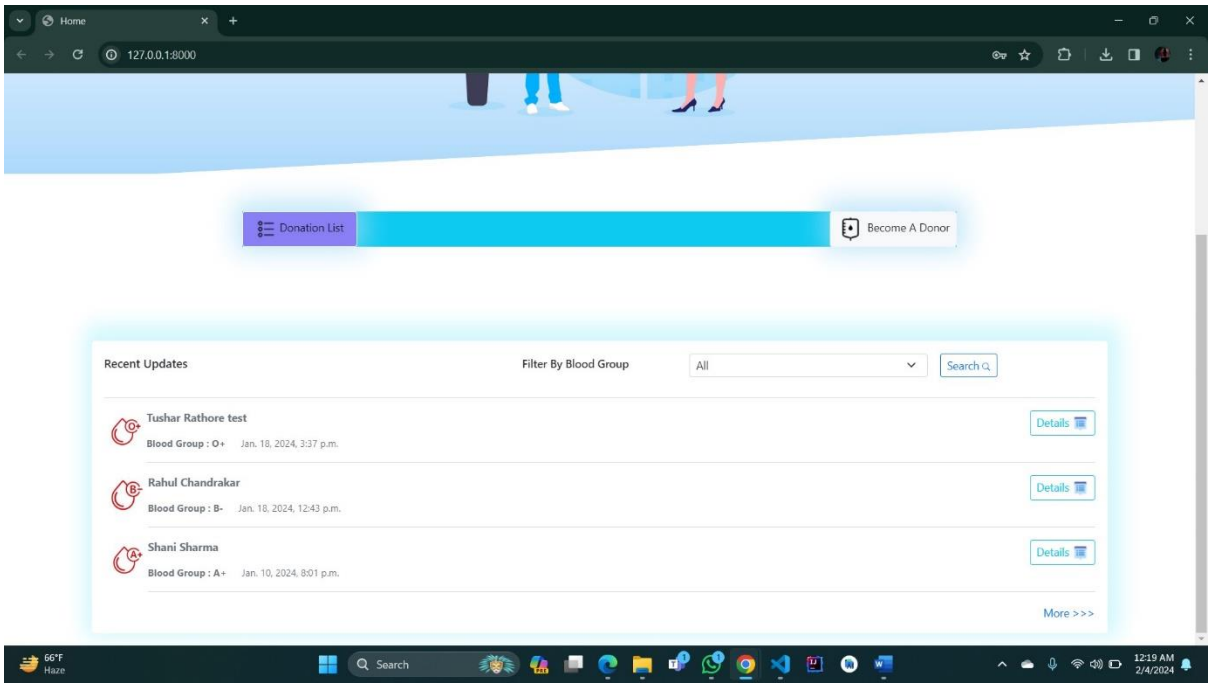
INPUT AND OUTPUT SCREEN DESIGN

This project is in the form of multi-page web application:

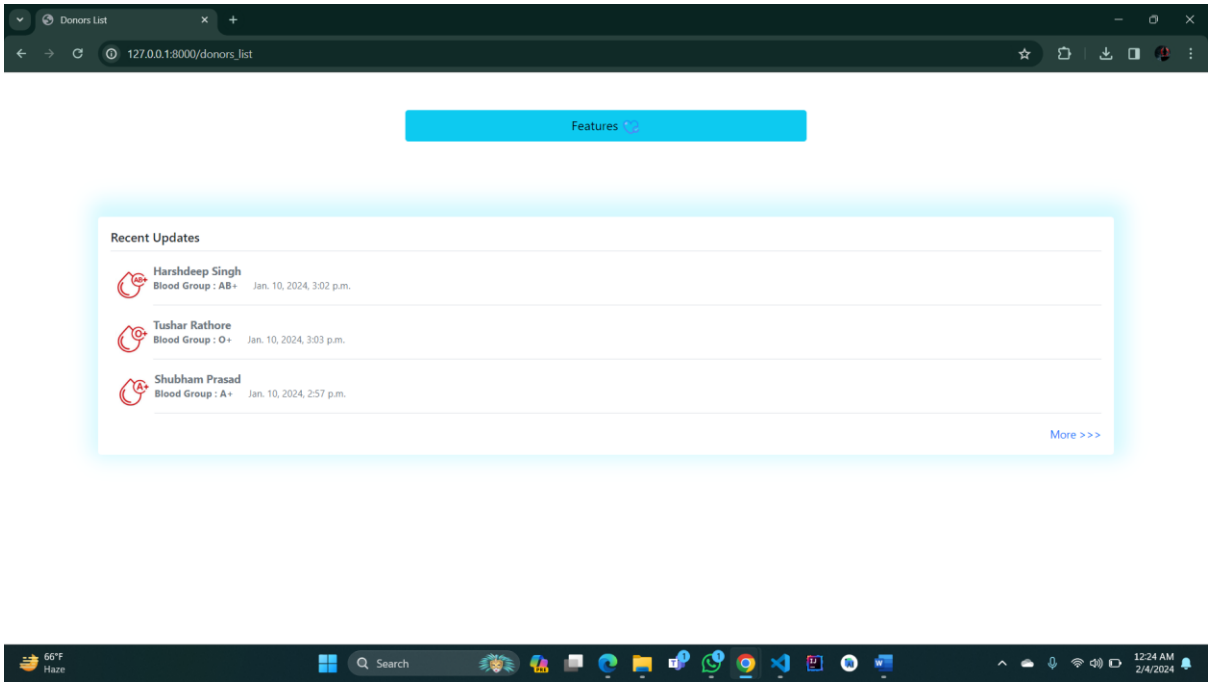
1) Home Page:



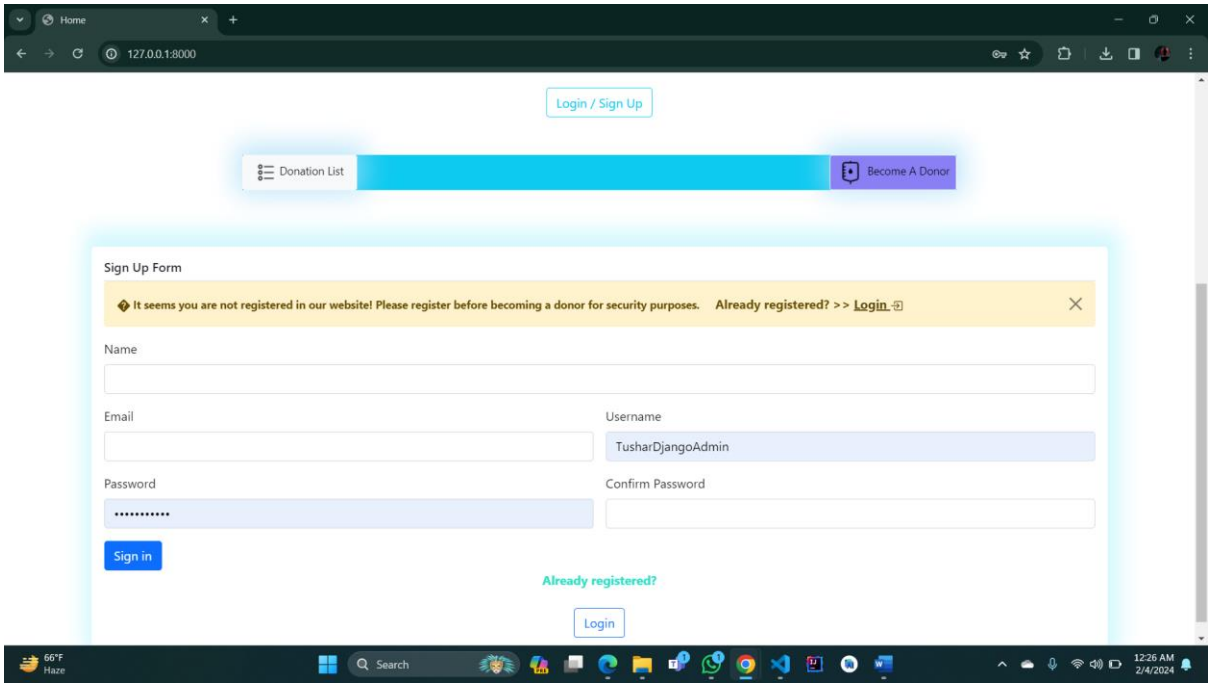
2) Search Fliter:



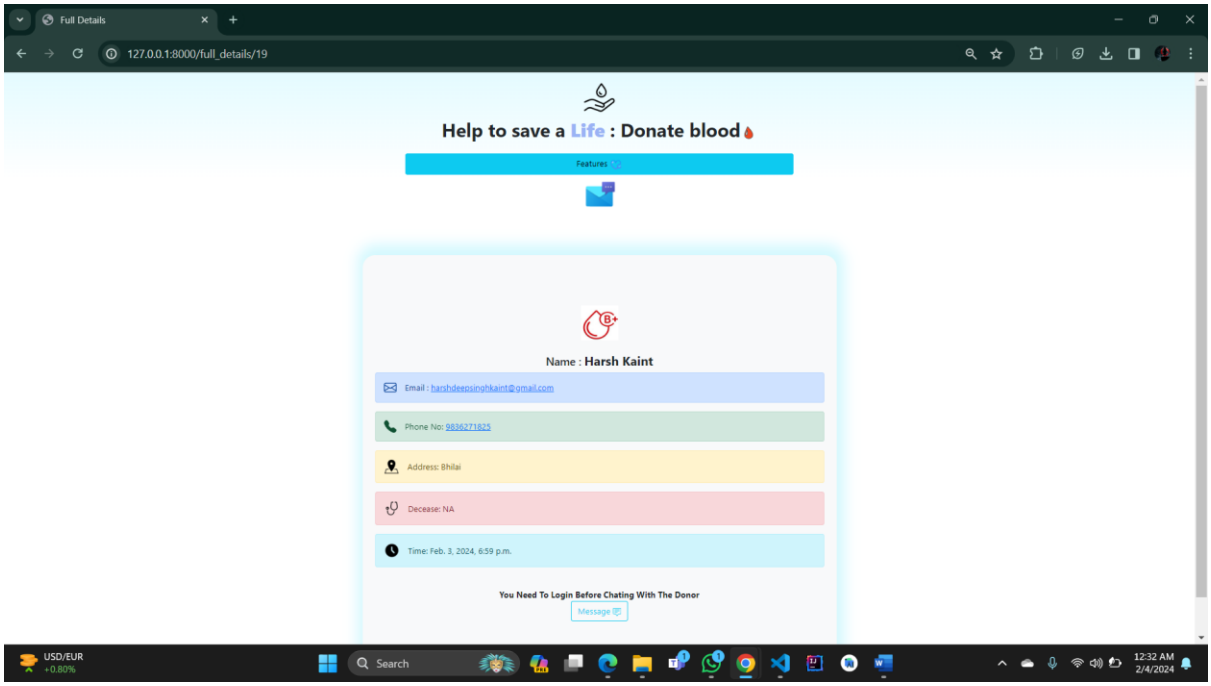
3) Donor List:



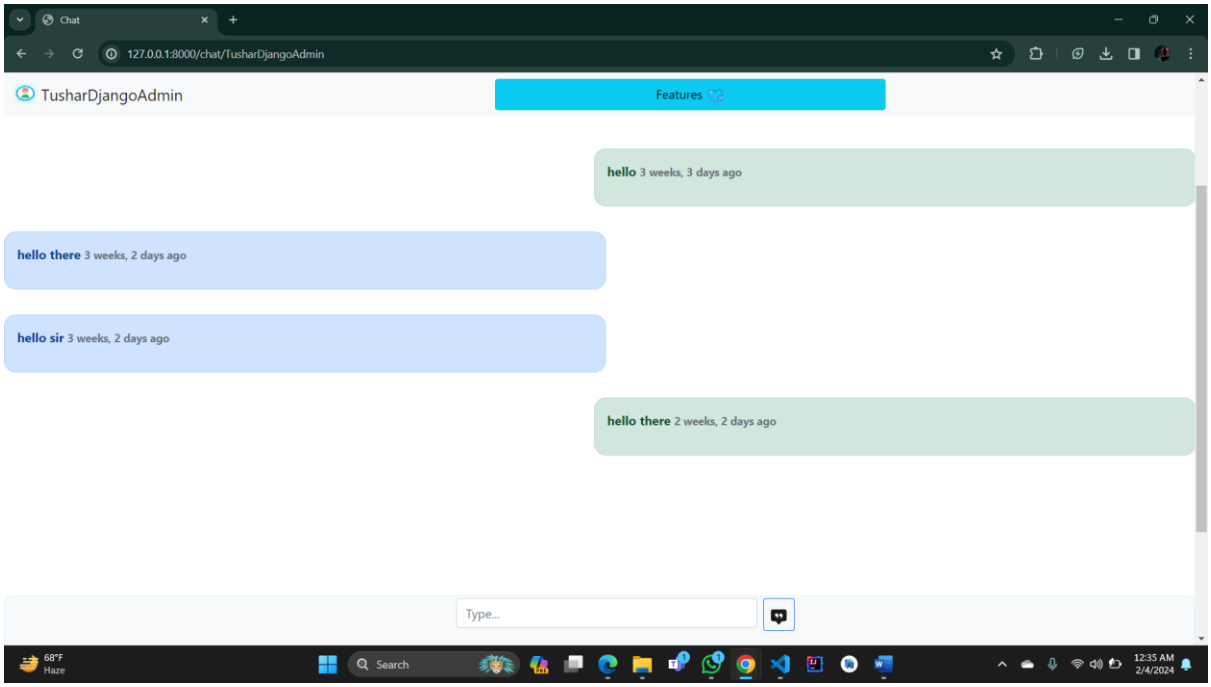
4) Login/Signup



5) Communication Page

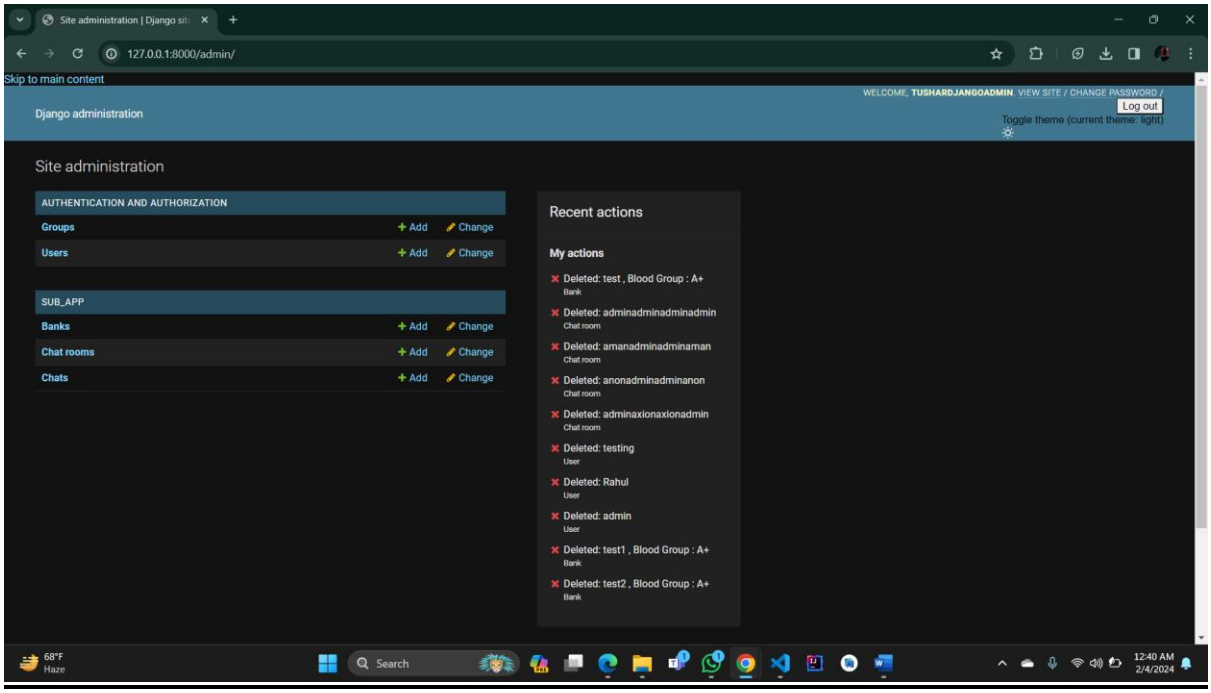


6) Communication channel

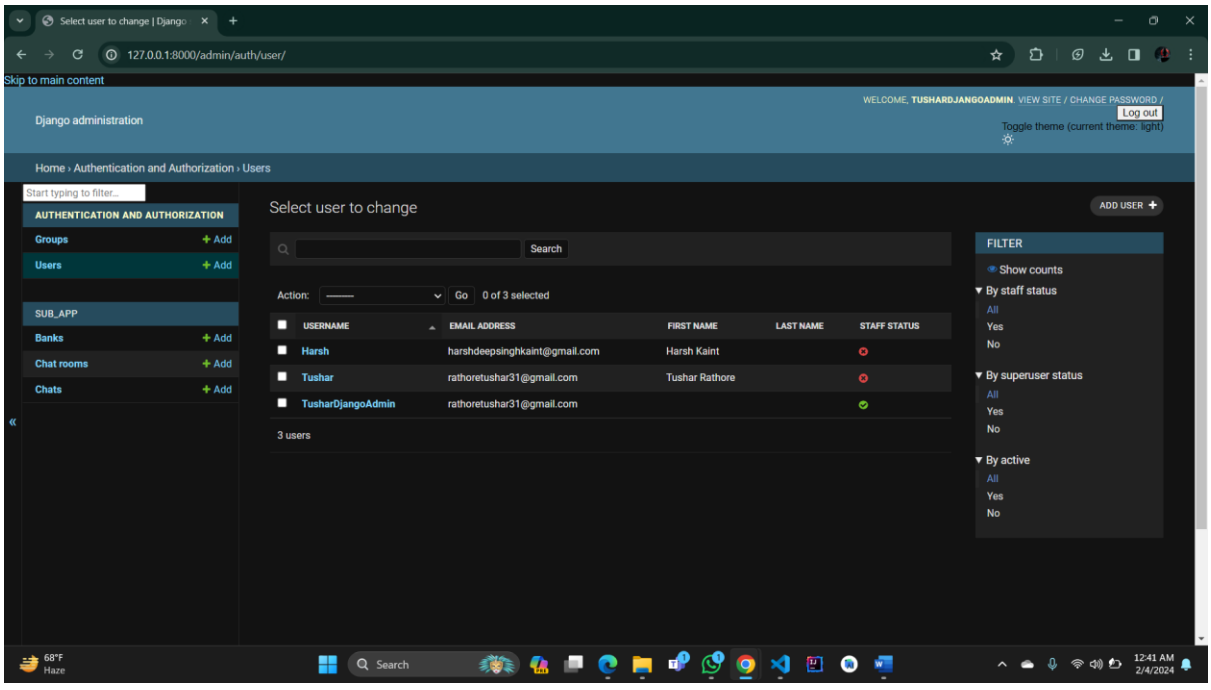


DJANGO ADMINISTRATION

1) Admin User

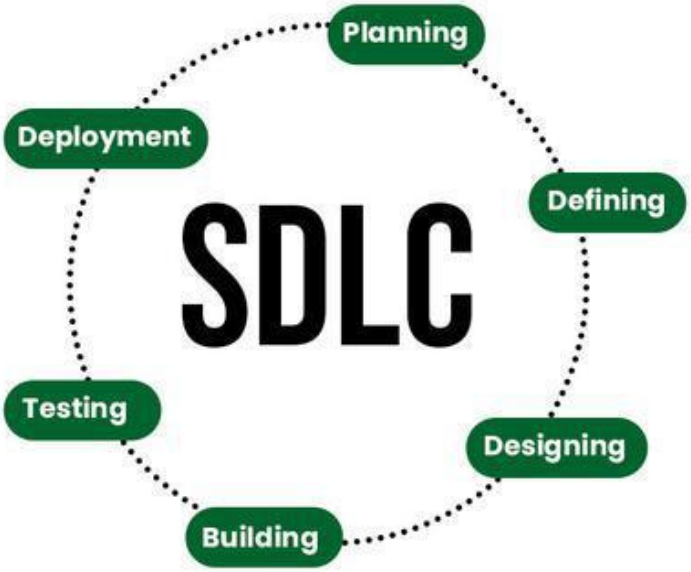


2) User Details



PROCESS INVOLVED

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.



A typical Software Development Life Cycle consists of the following stages −

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Python, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

The Life Flow - Blood Bank Website involves several essential processes that contribute to the seamless functioning of the platform. Each process plays a crucial role in achieving the system's objectives and providing a user-friendly experience. Here are the key processes involved:

User Registration and Authentication Process:

Objective: Allow users, including donors and blood seekers, to register and authenticate their identities securely.

Activities:

User registration with necessary details.

Authentication processes to verify user identity.

Secure handling of login credentials.

Donor Profile Management Process:

Objective: Enable donors to create and manage their profiles, including personal information, blood type, and donation history.

Activities:

Donor profile creation with relevant details.

Management of personal information.

Recording and updating donation history.

Blood Search and Matching Process:

Objective: Implement a search module for blood seekers to find donors with matching blood types and enable donors to respond to blood requests.

Activities:

Integration of search filters for blood type and other criteria.

Matching algorithms to connect blood seekers with suitable donors.

Facilitation of donor responses to blood requests.

Communication Module Process:

Objective: Facilitate direct communication between blood donors and seekers, allowing them to chat and coordinate blood donation details.

Activities:

Real-time communication features within the platform.

Secure and direct messaging between donors and seekers.

Coordination of blood donation logistics.

METHODOLOGY USED FOR TESTING

White-box Testing:

White-box testing for the Life Flow - Blood Bank Website involves a thorough examination of the internal code structure, emphasizing key aspects such as code coverage, security, performance, database interactions, error handling, and usability. This method entails activities like code coverage analysis, employing techniques such as statement, branch, and path coverage to assess the thoroughness of testing. security testing is a crucial component, aiming to identify and address vulnerabilities in the system. Simultaneously, performance testing evaluates the efficiency of algorithms and database queries related to blood donation processes, facilitating optimization efforts.

Database testing ensures the accurate storage and retrieval of donor information, contributing to the overall reliability of the blood bank platform. the comprehensive approach extends to usability testing, evaluating the user interface and overall experience, with user feedback solicited for continuous improvement. The analysis of specific paths and decision points within the codebase, along with robust database interaction security measures, rounds out the white-box testing strategy. this meticulous testing process is documented comprehensively, providing insights into code coverage, security measures, performance metrics, and usability assessments, contributing to the delivery of a secure, efficient, and user-friendly blood donation platform.

Fig: White Box Testing

TEST REPORT

This testing report pertains to a Job Portal website constructed using the Django framework. The aim of this report is to detail the testing conducted, identify any problems or glitches discovered, and assess the general functionality of the website. Additionally, the report provides details about the testing environment, including the Django version employed and any other essential dependencies or libraries.

Test Environment:

Operating System: Windows 11

Database: SQLite 3.40

Django Version: 5.0.1

Python Version: 3.10.5

Other Dependencies: HTML 5, CSS 3, Bootstrap , JavaScript

Python Packages : Django, Pillows, asgiref, sqlparse, Django-tweak-widgets

Testing Methods:

1. Functional Testing: All the website's functions were tested to ensure they are working correctly, such as adding items to the cart, making purchases, and managing account information.

2. Usability Testing: User surveys, user interviews, and user testing were performed to evaluate how easy the website is to use and navigate.

3. Performance Testing: The website's performance, such as its loading speed, response time, and scalability were evaluated.

4. Security Testing: The website's security, such as its ability to protect against hacking and data breaches, was evaluated.

5. Compatibility Testing: The website's compatibility with different devices, browsers, and screen resolutions was evaluated.

6. User Acceptance Testing: The website was tested to determine whether it meets the needs and expectations of the users.

7. Regression Testing: The impact of changes and updates on the website were evaluated and existing functionality was tested to ensure it was not broken.

Test Results:

➢ The website's functions were verified to be operating correctly. ➢ The website demonstrated ease of use and navigation.

➢ The website exhibited good performance, capable of handling high traffic and concurrent users effectively.

➢ The website's security measures were effective in safeguarding against hacking and data breaches.

➢ Compatibility was observed across various devices, browsers, and screen resolutions.

➢ The website effectively met the needs and expectations of users.

➢ No significant issues or bugs were identified during regression testing.

PRINTOUT OF THE CODE SHEET

Templates:

Home.html

{ % load static %}

< !DOCTYPE html > <html lang = "en" > <head > <meta charset = "UTF-8" / ><meta http - equiv = "X-UA-Compatible"content = "IE=edge" / ><meta name = "viewport"content = "width=device-width, initial-scale=1.0" / ><title > Home < /title>

<link rel="preconnect" href="https:/ / fonts.gstatic.com " />

<link href="https:/fonts.googleapis.com/css2?family=Varela+Round&display=swap" rel = "stylesheet" / ><link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/css/bootstrap.min.css"rel = "stylesheet"integrity = "sha384-wEmeIV1mKuiNpC+IOBjI7aAzPcEZeedi5yW5f2yOq55WWLwNGmvvx4Um1vskeMj0"crossorigin = "anonymous" / ></head> <header>

<div style ="margin-left: -1rem; margin-right: -1rem; border: none"

class="card bg-dark text-white showcase_div"

>img ="showcase" ="{% static 'undraw_medicine_b1ol.svg' %}"

alt="..." > <div style = "margin-left: 2rem"class = "card-img-overlay" > <p style = "color: black"class = "card-text" > </p>

<!-- color: #FF2E63;font-family: 'Varela Round', sans-serif;font-size: 1.4rem; -->

<br / > <br / ><h5>style = "

background: #3bcbff;

box-shadow: 0 8px 32px 0 rgba(0, 0, 0, 0.37);

backdrop-filter: blur(7.5px);

-webkit-backdrop-filter: blur(7.5px); border-radius: 10px;

border: 1px solid rgba(130, 0, 0, 0.18); color: #00f7ff; padding: 20px;

max-width: 300px; /* Adjust the width as needed */ color:#000000; "class = "card-title" > Life Flow: Give Blood 🩸 < br > Save Lives < /strong> :
 Your Single Act,
Their Second Chance. -->

</h5 > <br / ><button class = "btn btn-info"type = "button"data - bs - toggle = "offcanvas"data - bs - target = "#offcanvasExample"aria - controls = "offcanvasExample" > Features < img src = "https://img.icons8.com/nolan/24/heart-health.png" / ></button> <br / > </header > <!--Sidebar start--><div class = "offcanvas offcanvas-start"tabindex = "-1"id = "offcanvasExample"aria - labelledby = "offcanvasExampleLabel" > <img src = "https://img.icons8.com/cotton/64/000000/--bloodbag.png" / ><h5 class = "offcanvas-title"id = "offcanvasExampleLabel" > Blood Bank 🩸 < /h5>

<button type ="button" class="btn-close text-reset" data-bs-dismiss="offcanvas" aria-label="Close" ></button >

<h2 class="accordion-header" id="flush-headingOne">

<button class ="accordion-button collapsed"

type="button" -bs-toggle="collapse"

data-bs-target="#flush-collapseOne"

aria-expanded="false"

aria-controls="flush-collapseOne" >

Navigation 🧭 < src="https:/ / img.icons8.com / doodle / 32 / 000000 / drop - of - blood.png " -->

</button>

```
</h2><div
  id="flush - collapseOne "
  class="accordion - collapse collapse "
  aria-labelledby="flush - headingOne "
  data-bs-parent="#accordionFlushExample "  >
  <div class="accordion - body ">
   <ul class="nav nav - pills flex - column mb - auto ">
    <li class="nav - item ">
     <a href=" { % url 'home' % }
" class="nav - link active " aria-current="page ">
      <svg class="bi me - 2 " width="16 " height="16 ">
       <use xlink:href="#home "></use>
```

Welcome to Life Flow Blood Bank, where the gift of life flows through every drop of blood. Our mission is to connect donors with those in need, creating a lifeline that supports and sustains communities.

At Life Flow, we believe in the power of humanity to come together and make a difference. Our online blood bank platform provides a seamless experience for donors to contribute and for recipients to find the life-saving support they require.

```
</div> </div> </div> </div> </div> </div>
<div class="container mt - 5 ">
 <img
  style="width: 95 % ;
height: 50 % " src ="https: //github.com/Tushar-rathore/bloodBank/blob/main/static/marginalia-679.png?raw=true"
alt = "" / ></div> function first_btn() {
<script>
```

```
var popoverTriggerList = [].slice.call(
  document.querySelectorAll('[data-bs-toggle="popover"]')
);
var popoverList = popoverTriggerList.map(function (popoverTriggerEl) {
  return new bootstrap.Popover(popoverTriggerEl);
});
var popover = new bootstrap.Popover(
  document.querySelector(".popover-dismiss"),
  {
    trigger: "focus",
  }
);
// function for changing the button
function req_change(pk) {
  var pk = pk;
  console.log(pk);
  document.getElementById("fade_btn" + pk).style.width = 0;
  document.getElementById("fade_btn" + pk).style.height = 0;
  document.getElementById("fade_btn" + pk).style.background = "white";
  document.getElementById("fade_btn" + pk).style.border = "none";
  document.getElementById("fade_btn" + pk).style.cursor = "none";
  document.getElementById("fade_btn" + pk).innerHTML = " ";
```

```
        document.getElementById("request" + pk).style.display = "block";


        document.getElementById("request" + pk).innerHTML =
         "<button type='button' class='btn btn-outline-success btn-sm' data-bs-toggle='modal'
data-bs-target='#staticBackdrop" +
         pk +
         " '" +
         ">Request" +
         "<img src='https://img.icons8.com/fluent/18/000000/invite.png'/></button>";
      }
        document.getElementById("firstBtn").style.background = "#8980F5";
        document.getElementById("secondBtn").style.background = "#f8f9fa";
        document.getElementById("updt").style.display = "block";
        document.getElementById("frm").style.display = "none"; }
      function second_btn() {.getElementById("secondBtn").style.background =
"#8980F5";document.getElementById("firstBtn").style.background = "#f8f9fa";
        document.getElementById("frm").style.display = "block";
        document.getElementById("updt").style.display = "none";
      }  function show_(){
        document.getElementById("secondBtn").style.background = "#8980F5";
        document.getElementById("firstBtn").style.background = "#f8f9fa";
        document.getElementById("frm").style.display = "block";
        document.getElementById("updt").style.display = "none";      }
 </script > </body>
</html >
```

Python Codes

Views.py

```
from django.shortcuts import redirect, render, HttpResponse

from django.http import Http404, request

from .models import Bank,ChatRoom,Chat

from django.core.mail import EmailMessage, message

from django.conf import settings

from django.template.loader import render_to_string

from django.contrib.auth.models import User

from django.contrib.auth.decorators import login_required

from django.contrib import messages

from django.contrib.auth import login, authenticate, logout

from django.utils import timezone

from datetime import datetime

from django.db.models import Max

from django.contrib.humanize.templatetags.humanize import naturaltime

from . import models

# Create your views here.

def home(request):

    updates = Bank.objects.filter(status = "Requested").union(Bank.objects.filter(status
="Active")).order_by("-time")

    req_updates = Bank.objects.filter(status="Requested")
```

URLS.PY

```
from django.urls import path, include

from . import views

urlpatterns = [

    path('', views.home, name="home"),

    path('full_details/<int:pk>', views.fullDetails,name='fulldetails'),

    path('profile/<int:pk>', views.profile, name='profile'),

    path('delupdt/<int:pk>', views.delupdt, name='delupdt'),

    path("sign_up", views.sign_up, name="sign_up"),

    path('login_donor_home', views.login_donor_home, name="login_donor_home"),

    path('login_donor_del', views.login_donor_del, name="login_donor_del"),

    path("add_updt", views.add_updt, name="add_updt"),

    path("chat/<str:userchat>", views.talkmain, name="talk"),

    path("donors_list", views.donors_list, name="donors_list"),

    path("log_out",views.log_out, name="log_out") ]
```

Model.Py

```
from django.db import models

from django.contrib.auth.models import User

# Create your models here. class Bank(models.Model):    name =
models.CharField(max_length = 200)

    blood_group = models.CharField(max_length =15)

   user = models.ForeignKey(User, on_delete=models.CASCADE)

 email = models.CharField(max_length = 50)    phone_no = models.CharField(max_length =
10)    decease = models.CharField(max_length =300)
```

address =models.CharField(max_length = 300) time = models.DateTimeField(blank = True,null=True,auto_now_add=True)

req_time =models.DateTimeField(blank = True,null=True)

status = models.CharField(max_length=10,default="Active")

def __str__(self): return str(self.name) + " , Blood Group : " + str(self.blood_group)

#chat system class ChatRoom(models.Model):

#user_requested = models.ForeignKey(User, on_delete=models.CASCADE, related_name='user_reqed')

#user_requesting = models.ForeignKey

(User, on_delete=models.CASCADE, related_name='user_reqing')

owner = models.CharField(max_length=900)

chatter1 = models.CharField(max_length=200)

def __str__(self): str(self.owner)

class Chat(models.Model): room = models.ForeignKey(ChatRoom, on_delete=models.CASCADE) text = models.TextField()

time = models.DateTimeField(auto_now_add=True) chatter = models.CharField(max_length=900) __str__(self):

return str(self.text)

Admin.Py

from django.contrib import admin

from . models import Bank,Chat,ChatRoom # Register your models here.

admin.site.register(Bank) admin.site.register(Chat)

admin.site.register(ChatRoom)

User/Operational Manual

Welcome to Life Flow, your user-friendly guide in blood donation. This manual provides essential guidance on using the platform efficiently and securely.

Table of Contents:

User Registration and Authentication:

To access Life Flow, start by registering your account.

Provide accurate information during registration to ensure efficient communication.

Authentication ensures the security of your account. Keep your login credentials confidential.

Donor Profile Management:

Donors can create and manage profiles, including personal details, blood type, and donation history.

Maintain an up-to-date profile for accurate matching and effective communication.

Donors' eligibility status is tracked, enhancing the safety of the blood donation process.

Search Filters:

Refine your search for blood donors using filters such as blood type and location.

Streamline the search process for quicker and more accurate results.

Blood Search and Matching:

Blood seekers can find donors with matching blood types and send requests.

Donors receive requests and can respond promptly to fulfill blood donation needs.

Communication Module:

Utilize the built-in communication module for direct and secure communication. chat with donors to coordinate blood donation details efficiently.

Real-Time Blood Search:

Benefit from a dynamic search feature for real-time results based on blood type, location, and other criteria. stay updated on available donors to connect quickly.

Email Request Feature:

Use the integrated email request feature to send direct requests to potential donors. improve communication and response time for urgent blood needs.

Security Aspects:

Confidentiality:

Keep your login credentials confidential. do not share sensitive information with unauthorized individuals.

Data Encryption:

Life Flow employs encryption techniques to secure data during transmission. your personal information is protected from unauthorized access.

Access Rights:

Users have access rights based on their roles (Donor, Blood Seeker). administrators have additional access rights for platform management.

Backup Control:

Regular data backups are conducted to prevent data loss. in the event of system issues, data recovery is ensured.

CONCLUSION

Life Flow is like a helpful friend for finding blood when you need it. The goal is to make it super easy to connect with blood donors, ensuring you can quickly find the right blood type and talk directly to the donor who wants to help.

The existing way of keeping track of blood donors is a bit tricky, so Life Flow simplifies things. It keeps records of donors, making it hassle-free to find someone with the right blood type. The aim is to create a system where you can easily connect with a donor without any middleman involved.

The proposed Life Flow system is designed to make it easy for blood seekers to communicate directly with donors. It provides a common ground for both parties, ensuring that blood requests are fulfilled efficiently. The features, like a user-friendly interface and real-time blood search, contribute to making the blood donation process accessible and effective.

The technology used, including HTML, CSS, Bootstrap for the look and feel, and Django with SQLite for the backend, forms a strong foundation. Python libraries like Django, Pillow, and Django-tweak-widgets ensure everything runs smoothly.

Looking ahead, Life Flow aims to implement smart donor matching using AI, provide real-time updates for hospitals, form partnerships, and engage the community in blood donation efforts.

In conclusion, Life Flow is not just a website; it's a friendly tool that makes finding and donating blood easy and friendly for everyone involved. It's designed to be user-friendly and continuously improve to make the process as smooth as possible.

REFERENCES

- **Python :-**

  https://www.python.org/downloads/

- **Bootstrap :-**

  https://getbootstrap.com/

- **Javascript :-**
  https://www.javascript.com/resources

- **Django :-**

  https://www.djangoproject.com/

- **Sqlite :-**
  https://www.sqlite.org/index.html

- **JavaScript: The Definitive Guide**
  Author: David Flanagan

- **Django for Professionals: Production Websites with Python & Django**
  William S. Vincent, 2019