

Algorithm for Blood Bank Management System

1. USER LOGIN ALGORITHM

Input: username, password

Output: Redirect to respective dashboard or show error

1. START

2. Receive username and password from login form

3. Authenticate user using Django's auth system

4. IF authentication is successful THEN

a. Get user_type from CustomUser

b. IF user_type = 'admin' THEN redirect to Admin Dashboard

c. ELSE IF user_type = 'donor' THEN redirect to Donor Profile

d. ELSE IF user_type = 'requester' THEN redirect to User Index

5. ELSE

a. Show error message: "Invalid Credentials"

6. END

2. DONOR REGISTRATION ALGORITHM

Input: Donor registration form data

Output: Save donor details

1. START

2. Collect user details: name, email, password, blood group, etc.

3. Create a CustomUser instance with user_type = 'donor'
4. Create DonorReg instance and link it to the CustomUser
5. Save profile picture, mobile number, address, etc.
6. Redirect to donor profile with confirmation message
7. END

3. BLOOD REQUEST BY USER

Input: Form with patient details and donor selection

Output: Save blood request and notify admin

1. START
2. Get donor ID and requester details from the form
3. Validate fields (name, email, message, etc.)
4. Create a new BloodRequest instance
5. Set status as "Pending" by default
6. Save to database
7. Notify admin or log request for review
8. Redirect to confirmation page
9. END

4. ADMIN - MANAGE BLOOD GROUP

Input: Admin commands to add, update, or delete blood groups

Output: Reflect changes in the blood group table

1. START

2. IF Admin selects "Add Blood Group"
 - a. Get group name from form
 - b. Create and save new Bloodgroup instance
3. IF Admin selects "Update"
 - a. Fetch Bloodgroup by ID
 - b. Update name, save changes
4. IF Admin selects "Delete"
 - a. Delete Bloodgroup by ID
5. Refresh blood group management page with updated list
6. END

5. USER - CONTACT US FORM

Input: Contact form with name, mobile, email, message

Output: Store message and mark as 'Pending'

1. START
2. Get form data
3. Validate input fields
4. Create new Contact instance
5. Set status as "Pending"
6. Save to database
7. Show success message
8. END