

**CHITKARA**  
UNIVERSITY



**Batch- 2020**

**Department: Computer Science & Engineering**

Project Title:

**BloodLink**

**An Online Blood Bank**

**Name: Ujjwal**

**Roll No.: 2010991754**

**Group: G-23**

**Subject: Back-end Engineering [CS187]**

**Submitted To: Prof. Anupinder Singh**

# Introduction

- **Purpose of the project:** The Online Blood Bank project aims to create a platform for managing and tracking blood donations, connecting donors with recipients, and providing real-time information on blood shortages and needs.
- **Objectives:** The primary objectives of the project are to increase the efficiency and accessibility of the blood donation process, improve the matching of donors and recipients, and provide real-time information on blood shortages and needs.
- **Scope of the project:** The scope of the project includes the development of a web-based platform for managing and tracking blood donations, connecting donors with recipients, and providing real-time information on blood shortages and needs. The platform will include both a user-facing interface and an admin interface for managing the data.

## Project Description

- BloodLink will work as a platform for users to register as blood donors and blood banks/recipients to request blood. The system will also have an administrative side where authorized personnel can manage and monitor the system's operations.
- The milestones for the project include completing the database design, completing the user interface design, completing the back-end functionality, and completing testing.
- **Features:**
  - **User registration:** Users can register as blood donors and update their information, including their blood type and contact details.

- **User Authentication:** The system will use express-session and passport.js for user authentication. Users will be required to register and login to the system. Passwords will be stored in an encrypted format (such as bcrypt) to ensure the security of user's information. Express-session will be used to create sessions for logged-in users where session ID will be stored in a cookie on the user's device. Users will also have the ability to update their account information, including their password.
- **Blood request:** Hospitals, blood banks and recipients can request blood and specify the required blood type.
- **Blood donation:** Users can view blood requests and offer to donate blood.
- **Blood bank management:** Authorized personnel can manage blood requests, blood donations, and registered users.
- **Search functionality:** Users and hospitals/blood banks can search for blood donors based on blood type and location.
- **UI design:** The system will be designed in a way to provide a visually pleasing and user-friendly interface.

## **Technical Specification**

- **Front-end:** React.js
- **Back-end:** Node.js, Express.js
- **Database:** MongoDB

### **Database Design & Schema**

The database for the BloodLink project will be implemented using MongoDB, a NoSQL database. The following collections will be used to store and organize the data for the project: Users, Blood Banks, Blood Donations, and Blood Requests.

➤ **Users:** This collection will store information about registered users, including their name, contact details, blood type, and other relevant information. The schema for this collection can include fields such as:

- `_id` (ObjectId): unique identifier for each user
- `name` (string): name of the user
- `email` (string): email of the user
- `password` (string): hashed password of the user
- `bloodType` (string): blood type of the user
- `contact` (object): contact details of the user, including phone number and address

➤ **BloodRequests:** This collection will store information about blood requests made by hospitals and blood banks. The schema for this collection can include fields such as:

- `_id` (ObjectId): unique identifier for each request
- `requesterId` (ObjectId): reference to the user making the request
- `bloodType` (string): type of blood requested
- `quantity` (number): quantity of blood requested
- `status` (string): status of the request (e.g. open, closed, fulfilled)

➤ **BloodDonations:** This collection will store information about blood donations made by users. The schema for this collection can include fields such as:

- `_id` (ObjectId): unique identifier for each donation
- `donorId` (ObjectId): reference to the user making the donation
- `requestId` (ObjectId): reference to the corresponding blood request
- `bloodType` (string): type of blood donated

- quantity (number): quantity of blood donated
- date (date): date of donation

➤ **Admin:** This collection will store information about authorized personnel who can manage and monitor the system's operations. The schema for this collection can include fields such as:

- \_id (ObjectId): unique identifier for each admin
- name (string): name of the admin
- email (string): email of the admin
- password (string): hashed password of the admin

----- Thank You