***BLOOD BANK MANAGEMENT SYSTEM***

Submitted By,

Sangeeth S V - 181CO246

Jayakrishna Sukumaran - 181CO223

**INTRODUCTIOn**

Blood Bank Management System is a web application software which is helpful for sick patients wanting blood units and also for generous donors looking to donate their own. In the current system all the activities are announced via newspaper or other old techniques. It is very time consuming and inefficient. Our Blood Bank Management System deals announces the various activities related to the blood donation system quickly and efficiently.

There are mainly 4 sections in this software

* User section
* Sponsor section
* Donation Camp section
* Hospital section

In the Software we can register as a user and user is of two types, donor and patient with separate tables. Administrator has the power to add new activities such as donation camps and new areas covered can edit and delete a camp or hospital. Anyone can register as user and can setup his profile as a donor or as a patient. All the users can see the waiting list in order to see what blood is urgently needed by the hospitals. The patients can specify their urgency for the blood donation. Based on the date of request and the urgency/priority, the web application places them in the waiting list.

**SYSTEM ANAlysis**

**EXISTING SYSTEM:**

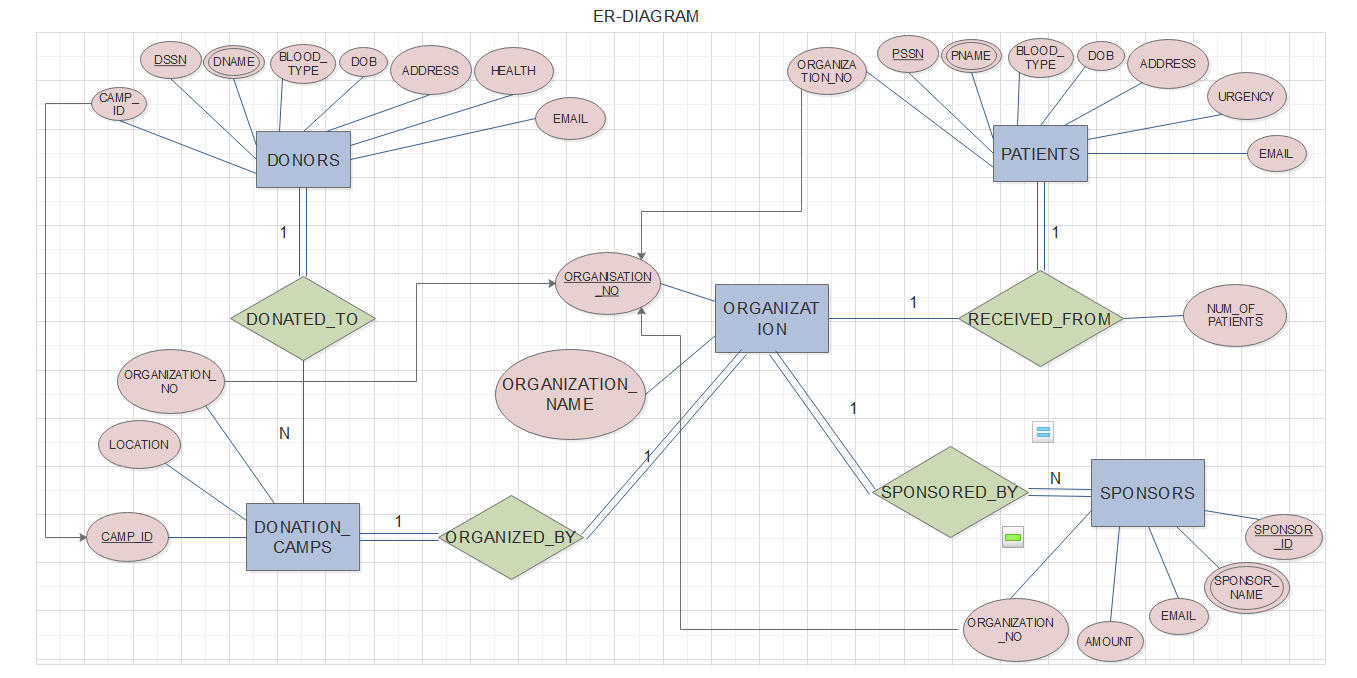
In the current system, most web applications charge the users money to find a donor. The donation camps are not always updated. The waiting list is not sorted according to priority or urgency of the patient. The registration is often not secure and has resulted in cases of fraud.

**PROPOSED SYSTEM:**

In our system, there is no payment/funds involved. Interested parties are allowed to donate money but is not made compulsory.

* A user-friendly interface.
* The donation camps are regularly updated by the admins maintaining the server.
* The waiting list is sorted using both urgency of the patient as well as the date of request/application.
* The registration is very secure and all passwords are stored using hashing technique for added security.
* Search facility for camps is there, which automatically fetches all camps organized at your location.
* Waiting list is also optimized for your location.

**ENTITY RELATIONSHIP (ER) DIAGRAM**

****

The above ER diagram describes the relationships involved in our database.

**Entity Relationships:**

1. DONORS *donated\_to* DONATION\_CAMPS
2. PATIENTS *received\_from* ORGANIZATIONS/HOSPITALS
3. ORGANIZATION sponsored\_by SPONSORS
4. DONATION\_CAMPS organized\_by ORGANIZATIONS

**NORMALIZATION**

The 5 tables/collections are:

1. Donors
2. Patients
3. Donation Camps
4. Sponsors
5. Organizations (Hospitals)

**DONORS:**

EMAIL (key attribute) determines all the other features like NAME, BLOOD\_GROUP, DONATION\_CAMP applied to, etc. It is in BCNF.

LOCATION determines CAMP\_ID. It is in 3NF.

**PATIENTS:**

EMAIL (key attribute) determines all the other features like NAME, URGENCY, BLOOD\_GROUP, etc. It is in BCNF.

LOCATION determines HOSPITAL. It is in 3NF.

**DONATION CAMPS:**

CAMP\_ID (key attribute) determines LOCATION, CAMP\_DATE. It is in BCNF.

**SPONSORS:**

SPONSOR\_NAME determines the other features, BCNF.

**HOSPITALS:**

HOSPITAL\_ID determines all the other features, BCNF.

From the above described normalization, we can say that the database is of good design with less redundancy and optimized data.

**IMPLEMENTATION (CODING)**

The Blood+ web application has been developed using many different softwares for various purposes.

**FRONTEND:**

1. HTML – structuring the web pages.
2. CSS – styling of the web pages.
3. Bootstrap – predefined elements used in HTML
4. Javascript – functionality of the web pages’ features.
5. jQuery – increasing the usability of Javascript.

**BACKEND:**

1. Node.js – for establishing the server and for all server-side coding.

* Express.js – increases usability of node.js
* Passport.js – for the secure authentication maintaining the session.

1. MongoDB – for establishing the database on the server side, for the safe storage of data.
2. Mongoose – makes the coding part to access the mongoDB database easier in node.js.
3. Ejs – this was used as the view engine in node so as to reuse some of the code in html

The above mentioned softwares have been included in the implementation of web application for the Blood Bank Management System.

**INSTRUCTIONS ON EXECUTION:-**

**GITHUB LINK:-** <https://github.com/sangzzz/Bloodbank>.

**To run from scratch**, visit github repository and clone it. Then, download zip version and extract.

Make sure nodejs is installed, if not, install it with all the default settings.

Open terminal, **cd** over to the project folder. Run **npm** **install** to install all the dependencies.

Once, complete, run **‘node app.js’** to set up the server.

Now, open browser and go to <http://localhost:3000/>**.**

**HEROKU LINK:-** <https://lit-temple-61443.herokuapp.com>.

**To check out the working project, visit the above link to view the project hosted on Heroku.**

**Sample Users:**

Admin Access: Username: [sangeeth@mail.com](mailto:sangeeth@mail.com), Password: 1245

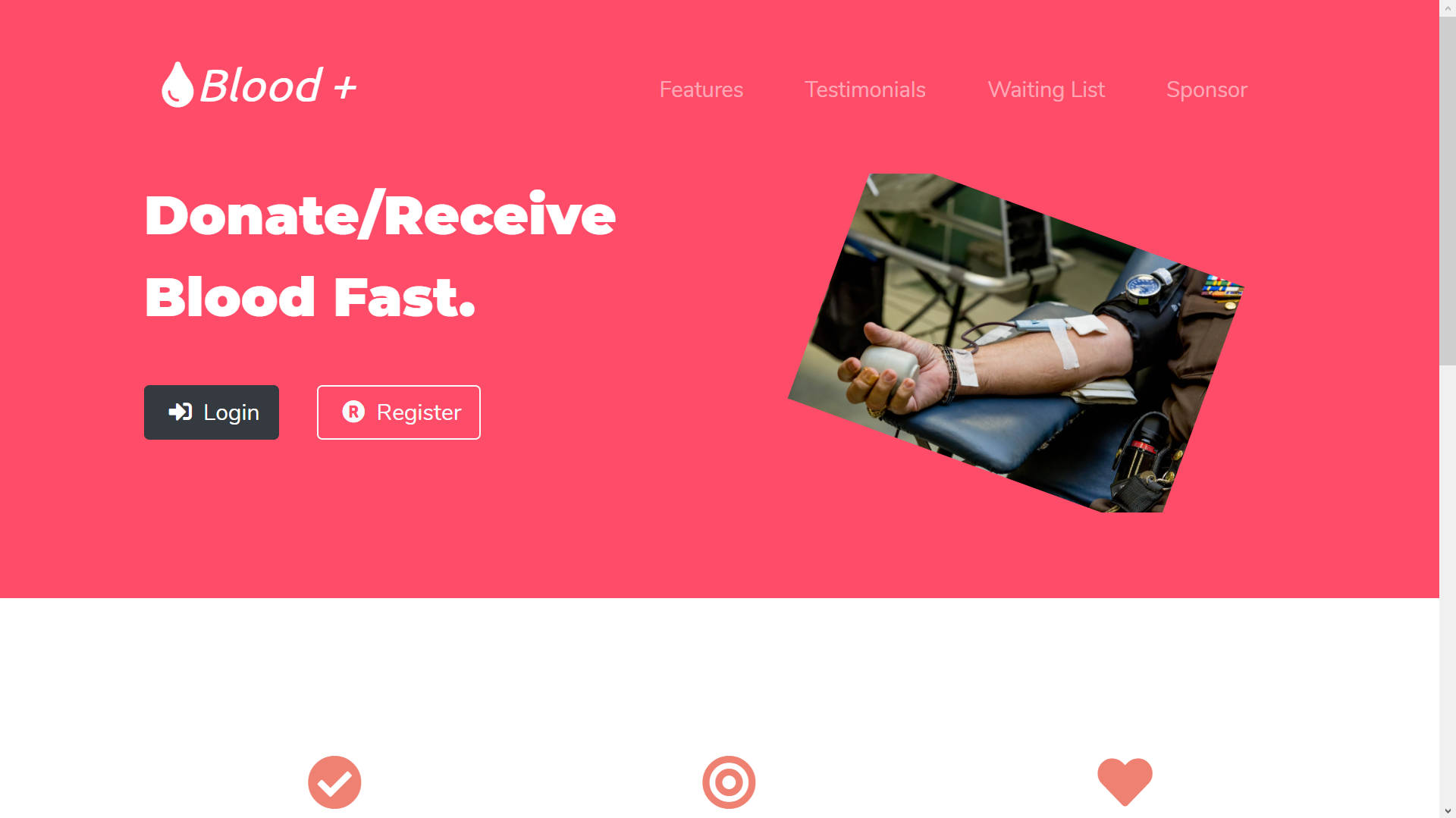
Username: [jaks@mail.com](mailto:jaks@mail.com), Password: 1245

Donor Access: Username: sang@ee.th, Password: 1245

Patient Access: Username: jaks@email.in, Password: 1245

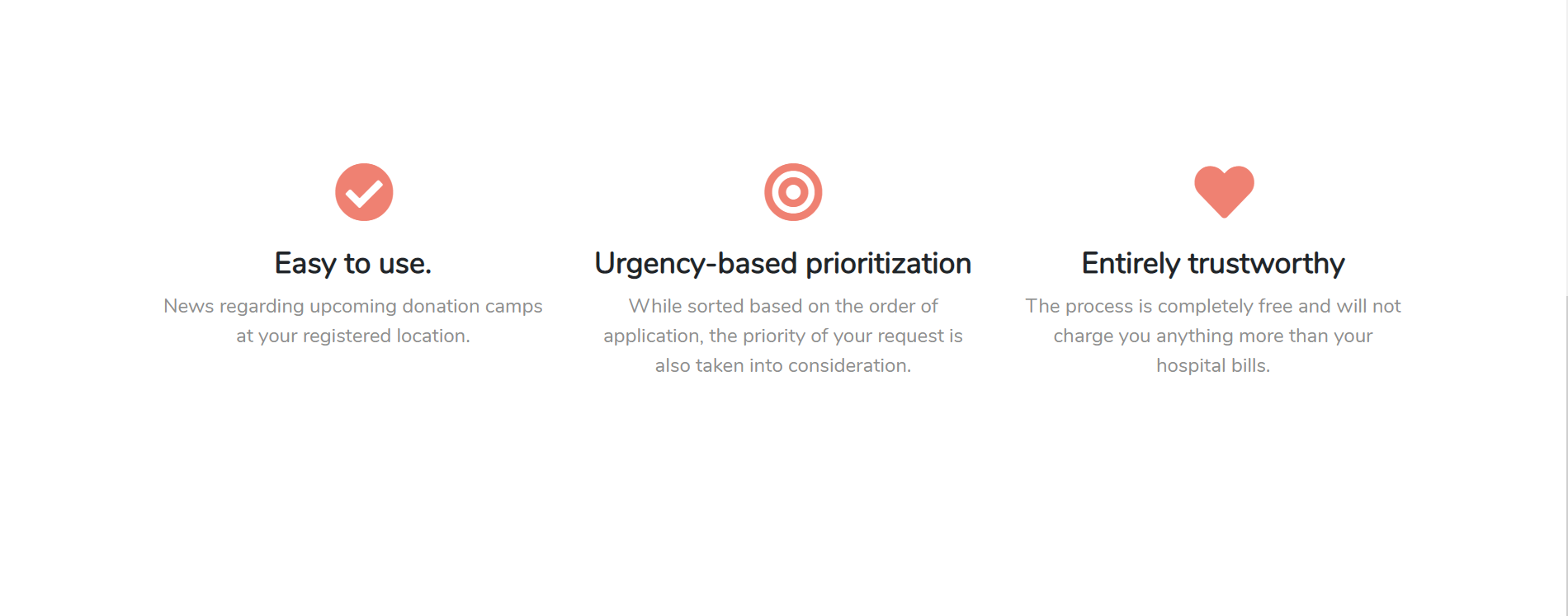
**SNAPSHOTS OF THE OUTPUT**

1. **HOME PAGE**



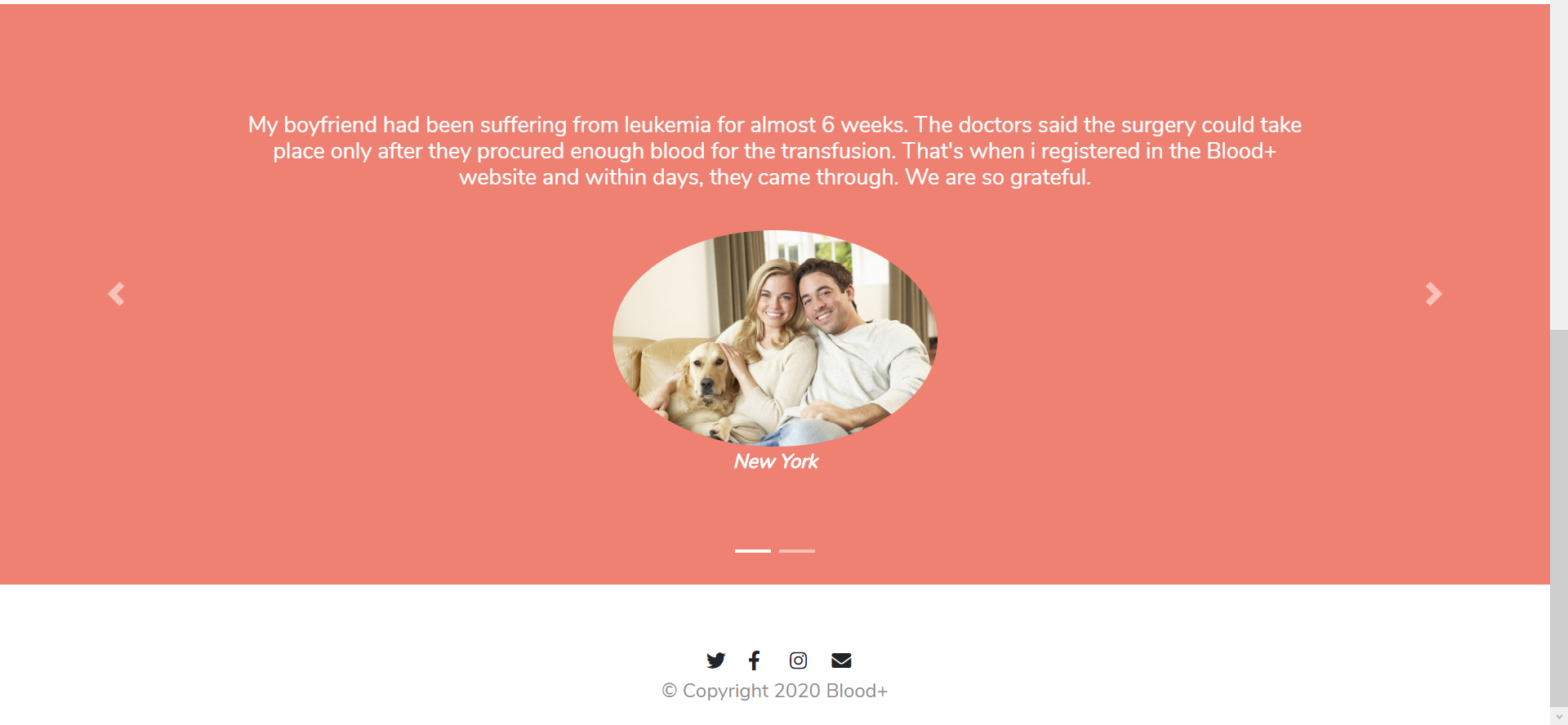
Snapshot 1:- The home page has links for an unauthorized user to view the features and testimonials of previous users. Interested parties are also allowed to donate/sponsor a hospital.

**Features:**



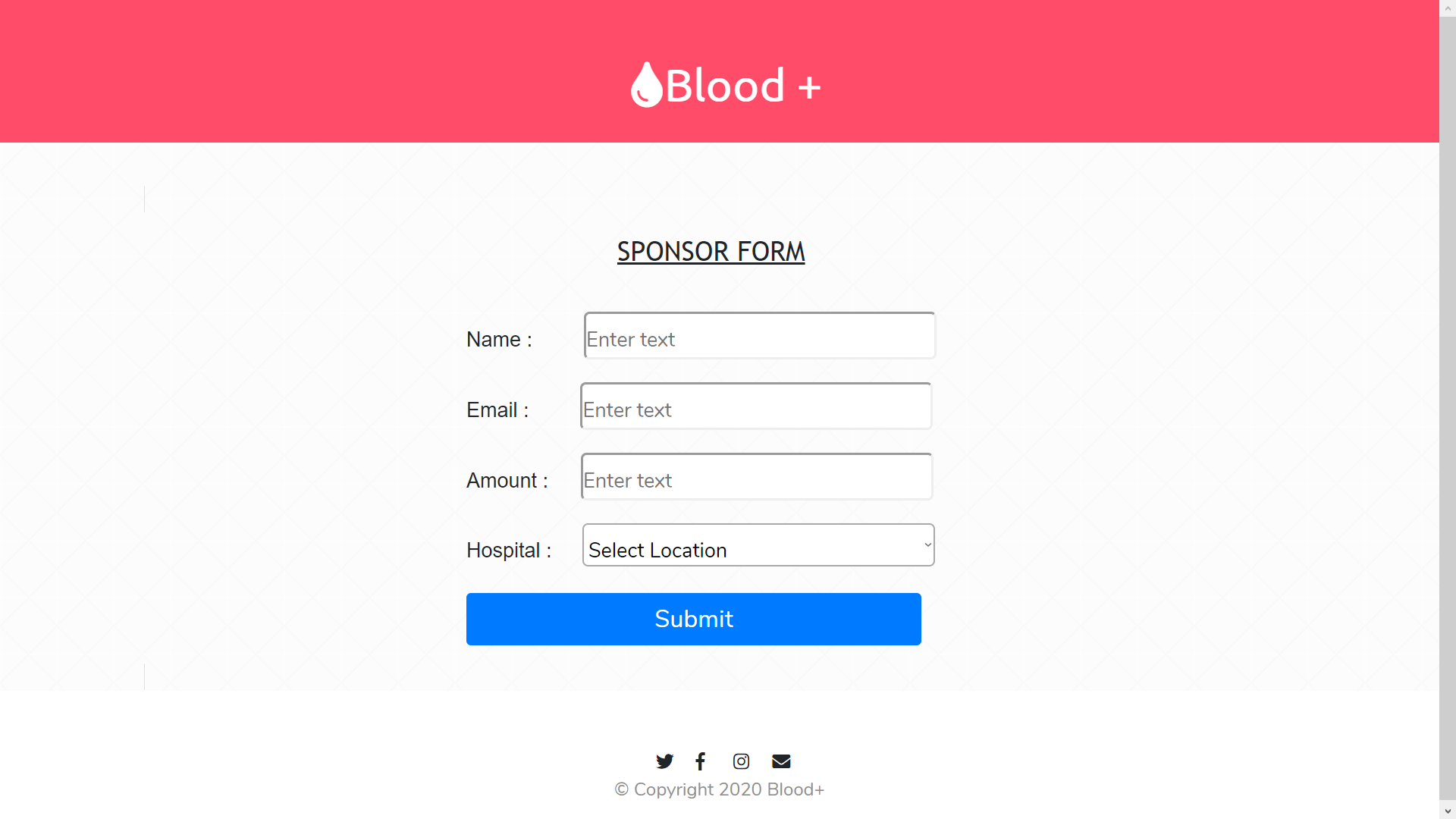
Snapshot 2:- The main features of the web application.

**Testimonials:**

****

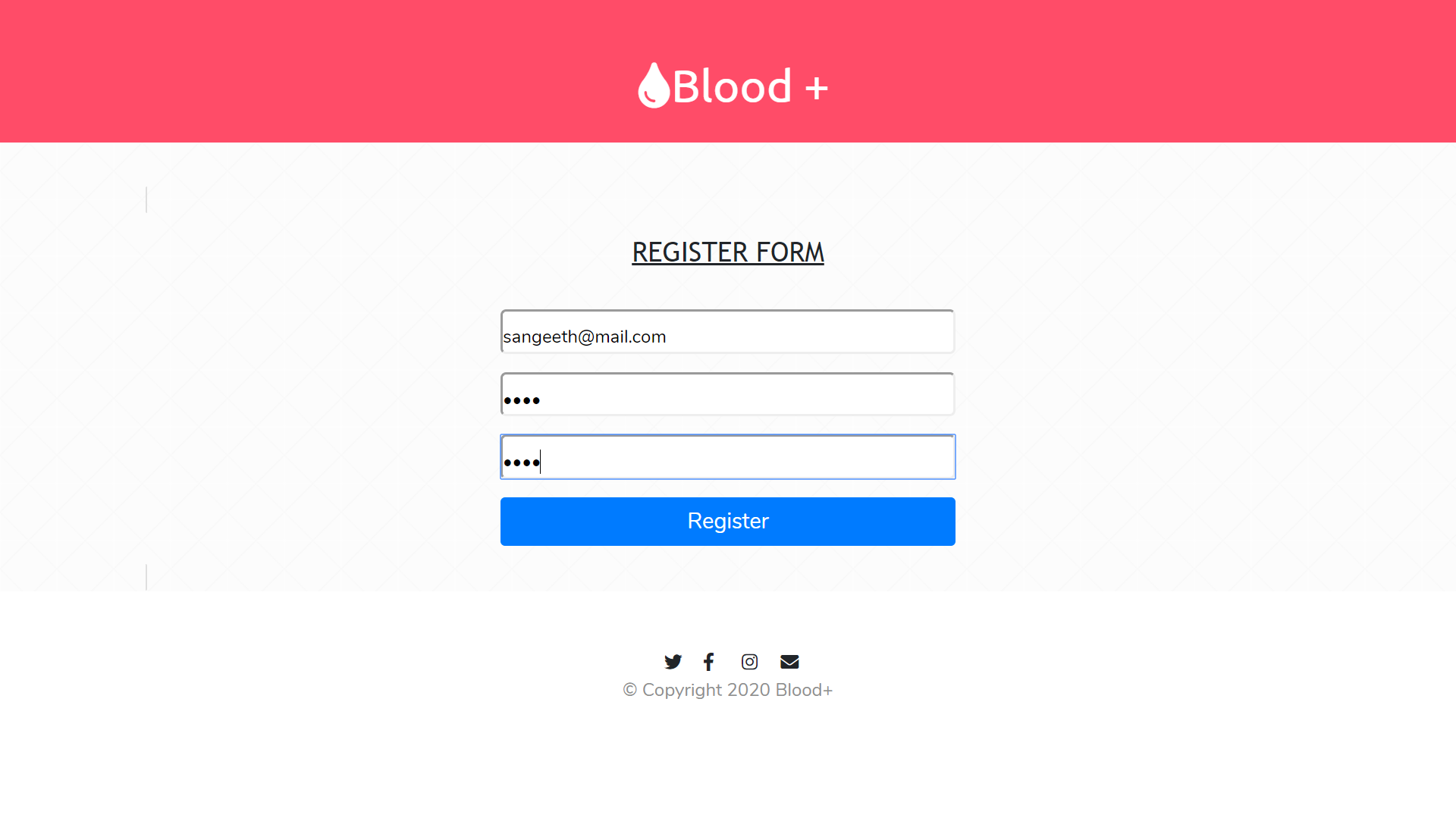
Snapshot 3:- Testimonials of previous users of the web app.

**Sponsor:**



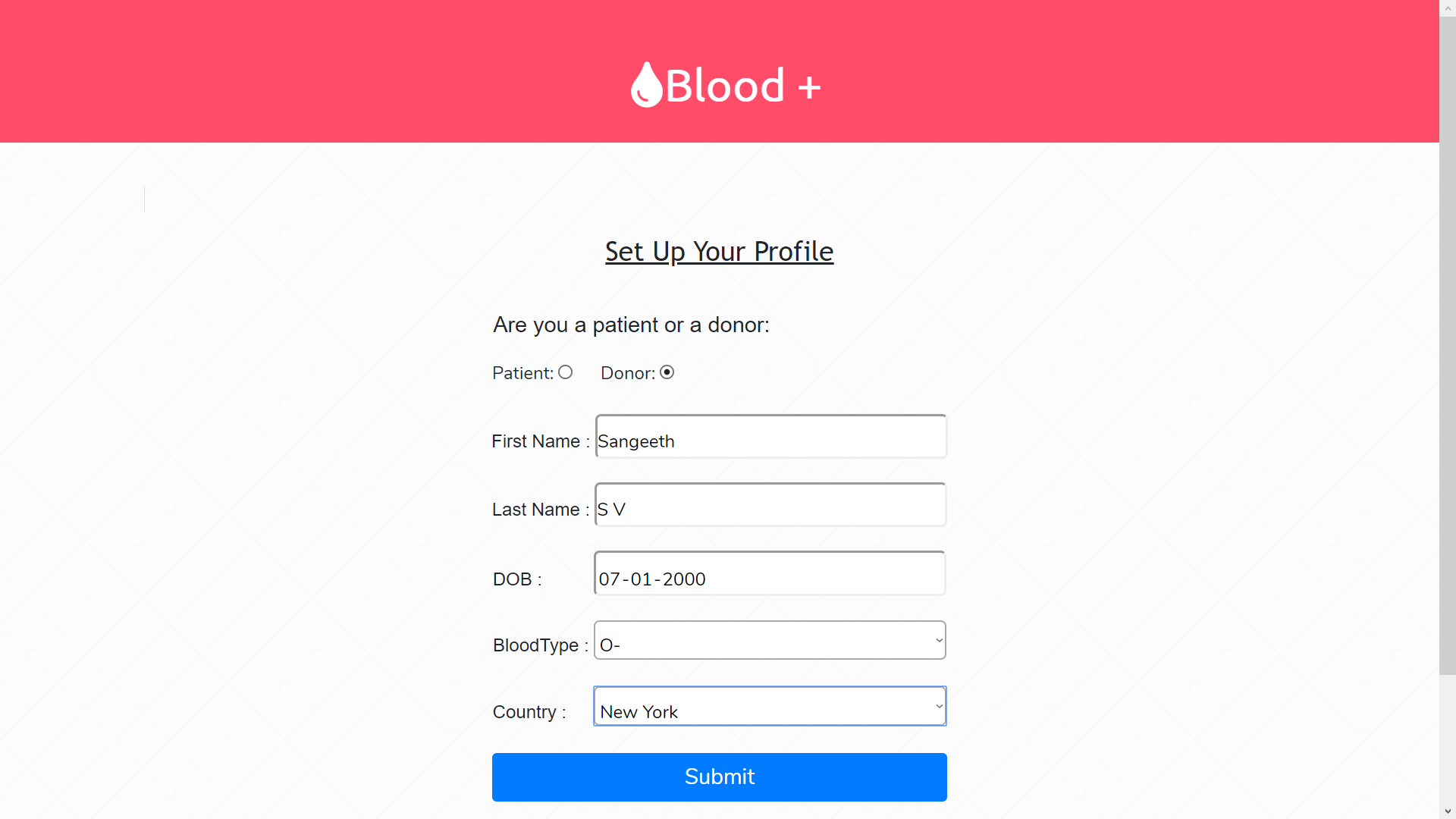
Snapshot 4:- The web app allows us to donate any amount of money to hospitals in the app database. (Payment portal has not been implemented.)

1. **REGISTRATION PAGE:**

****

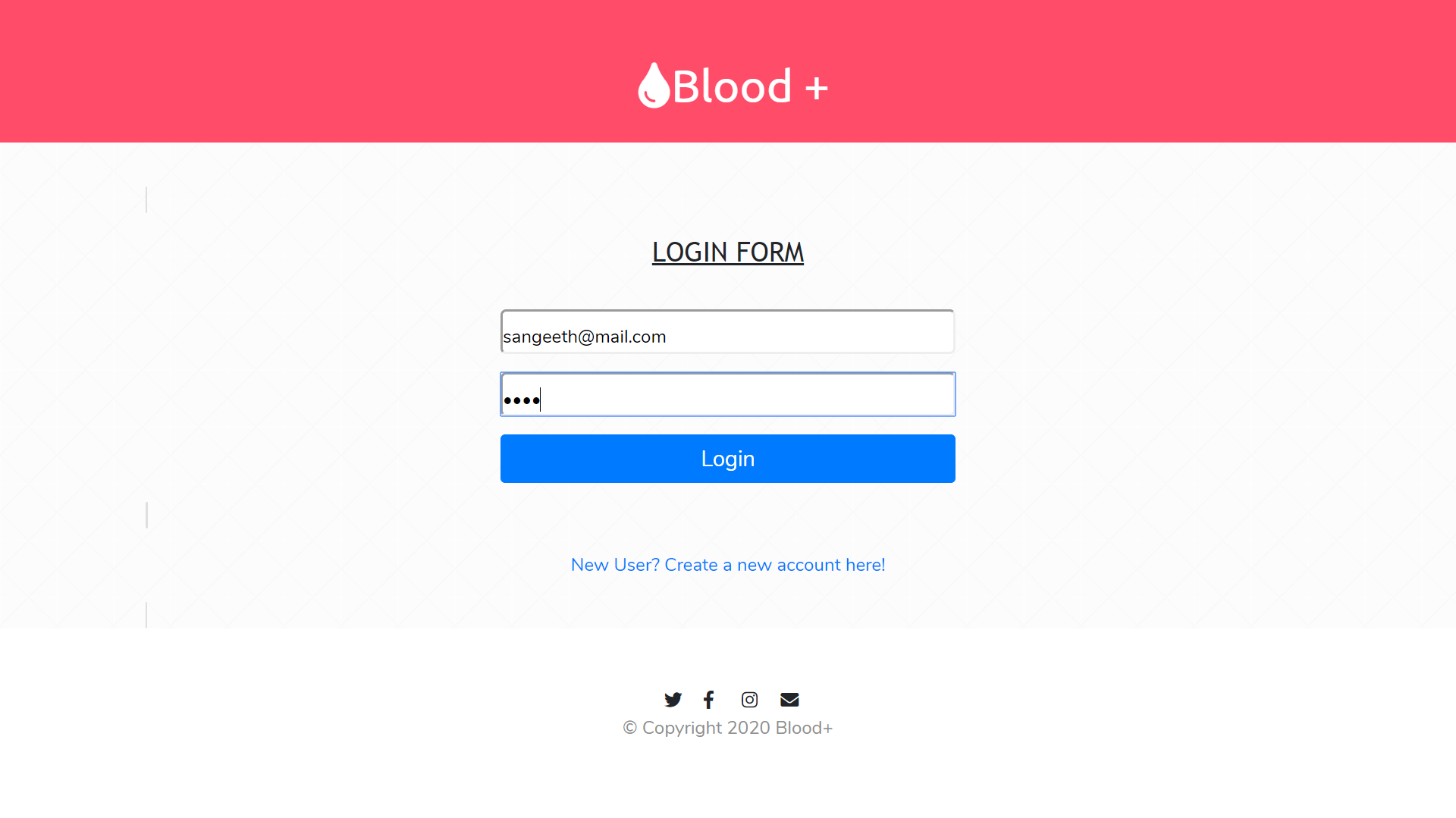
Snapshot 5:- Registration form asks for your email and password entered twice for confirmation. If the passwords do not match or an account with this email already exists, the user is redirected to the register page, once more.

After clicking Register,



Snapshot 6:- After clicking on submit, it goes to the logged-in home page. It doesn’t allow you to leave any fields as empty.

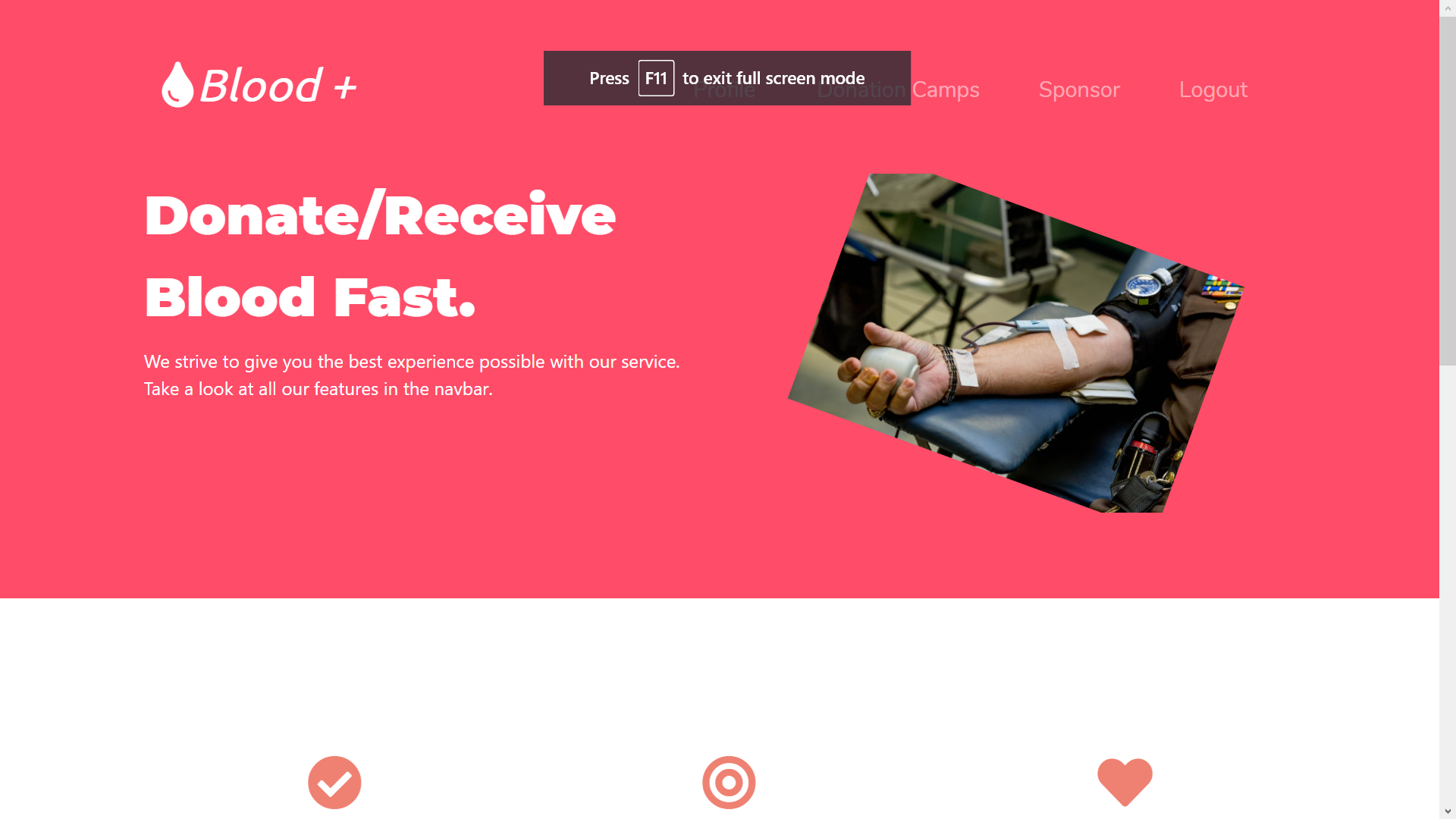
1. **LOGIN PAGE:**



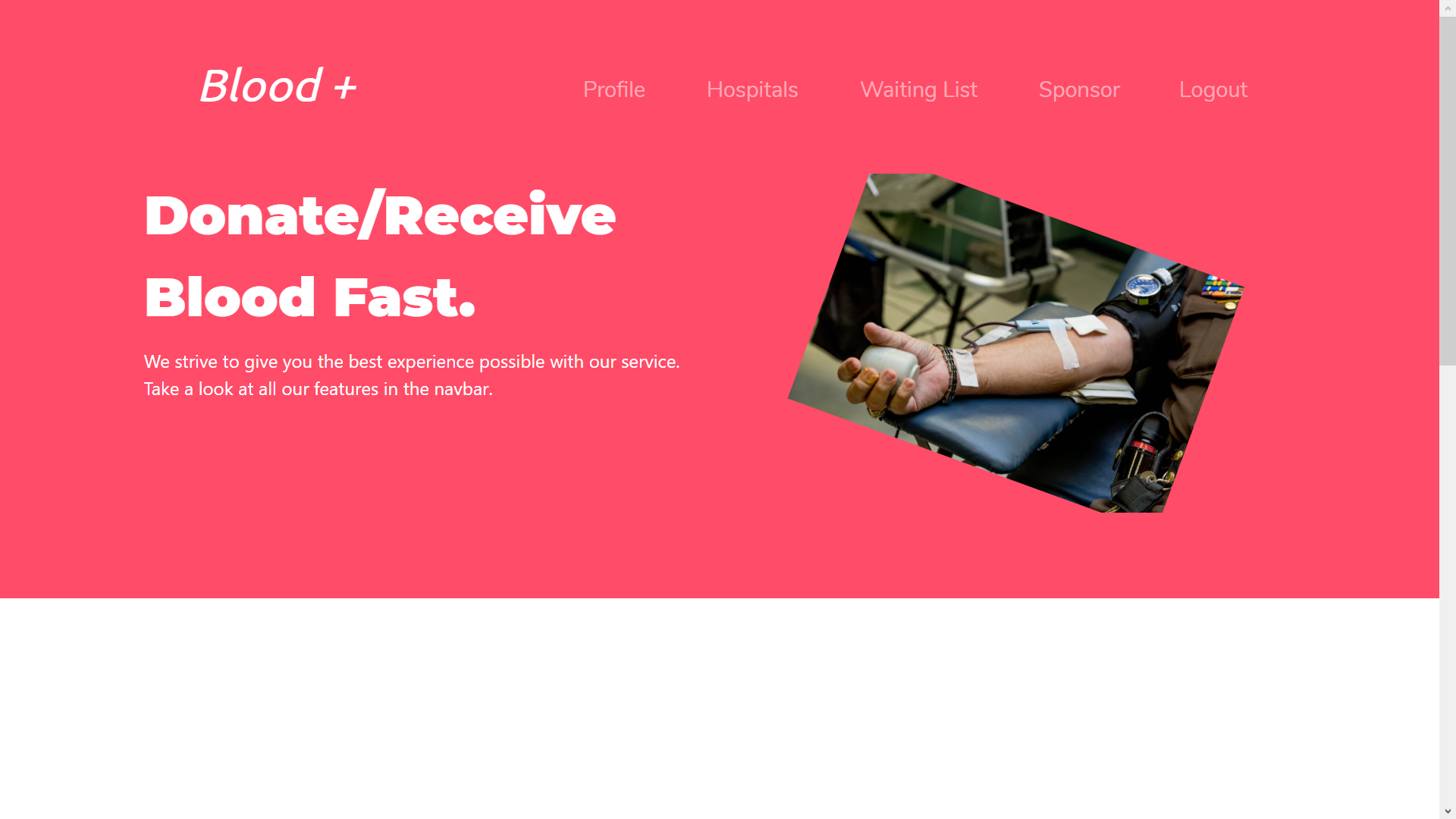
Snapshot 7:- This allows a signed-up user to login to the web application. If you are a new user, there is an option to go to register page.

After clicking Login, you get routed to the Logged-in-homepage of the web app.

**4. LOGGED-IN HOME PAGE:**



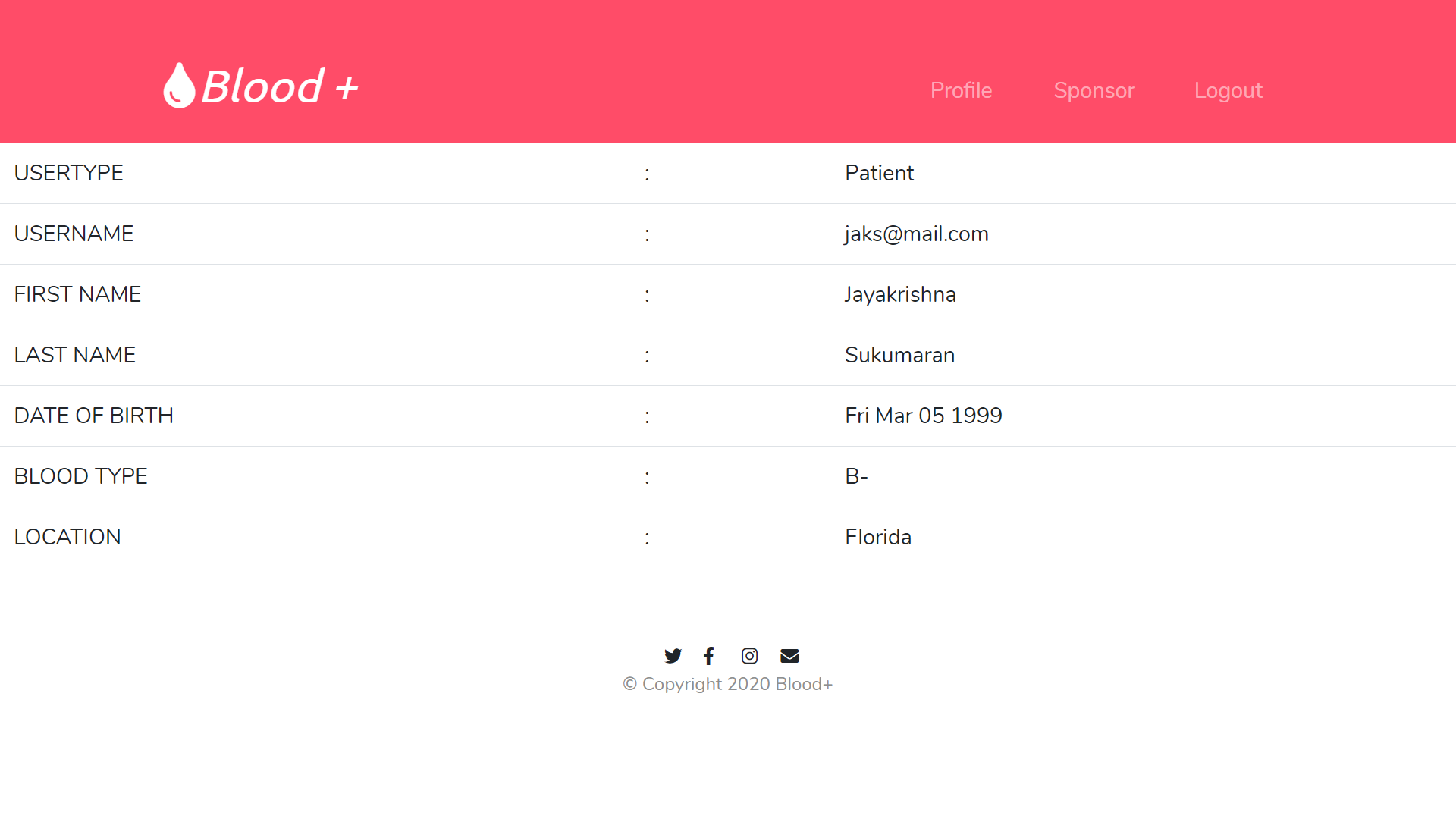
Snapshot :- The Logged In page for a donor. From here, a donor can access donation camps, see the waiting list and apply to donate blood to a camp based on the date, and also, log out of the web app.



Snapshot :- The Logged-In-Homepage of a patient. From here, you can view your profile, apply to a hospital, get the waiting list and also, log out of the web app.

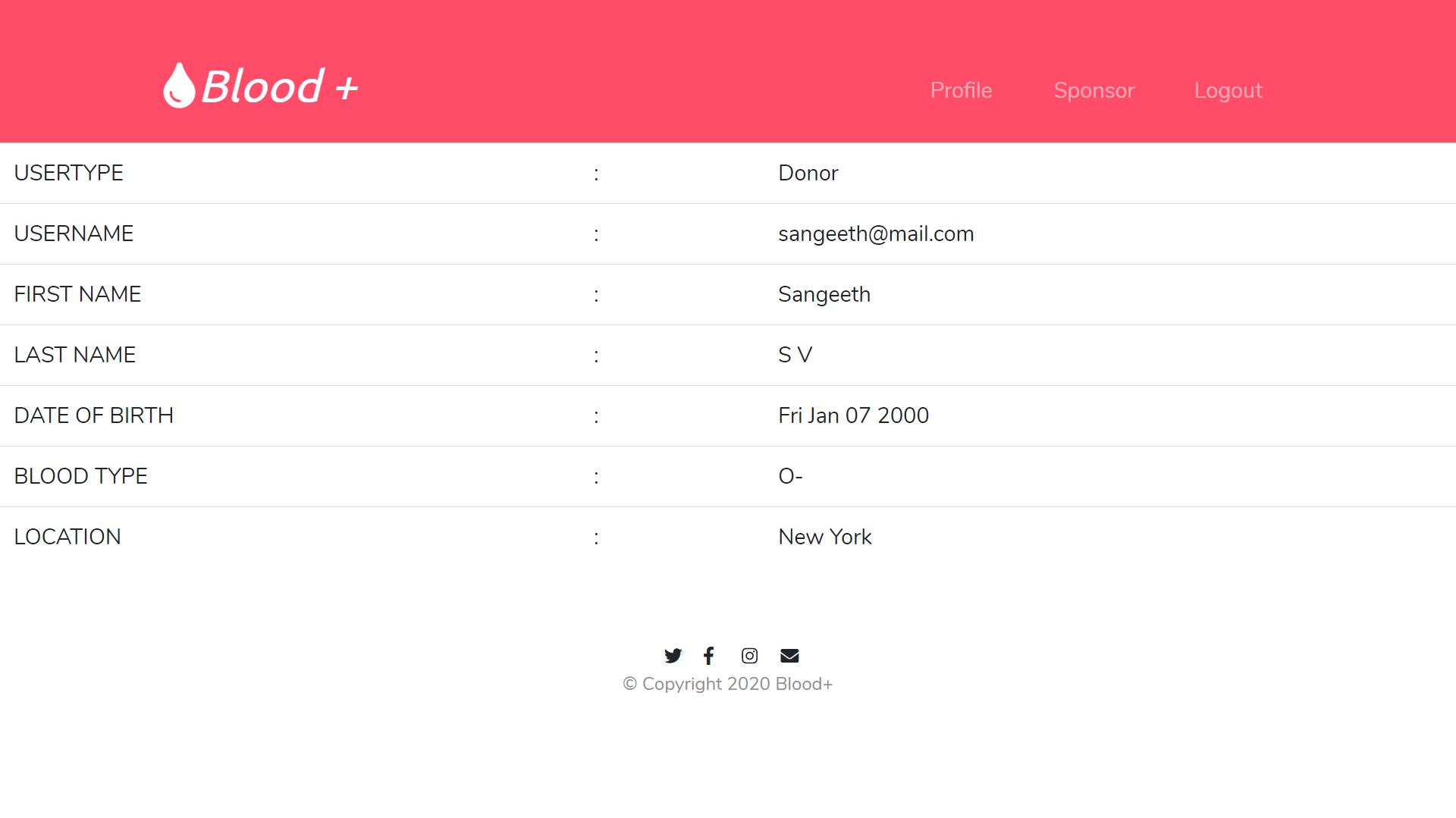
**5. PROFILE:-**

**Patient Profile:**

****

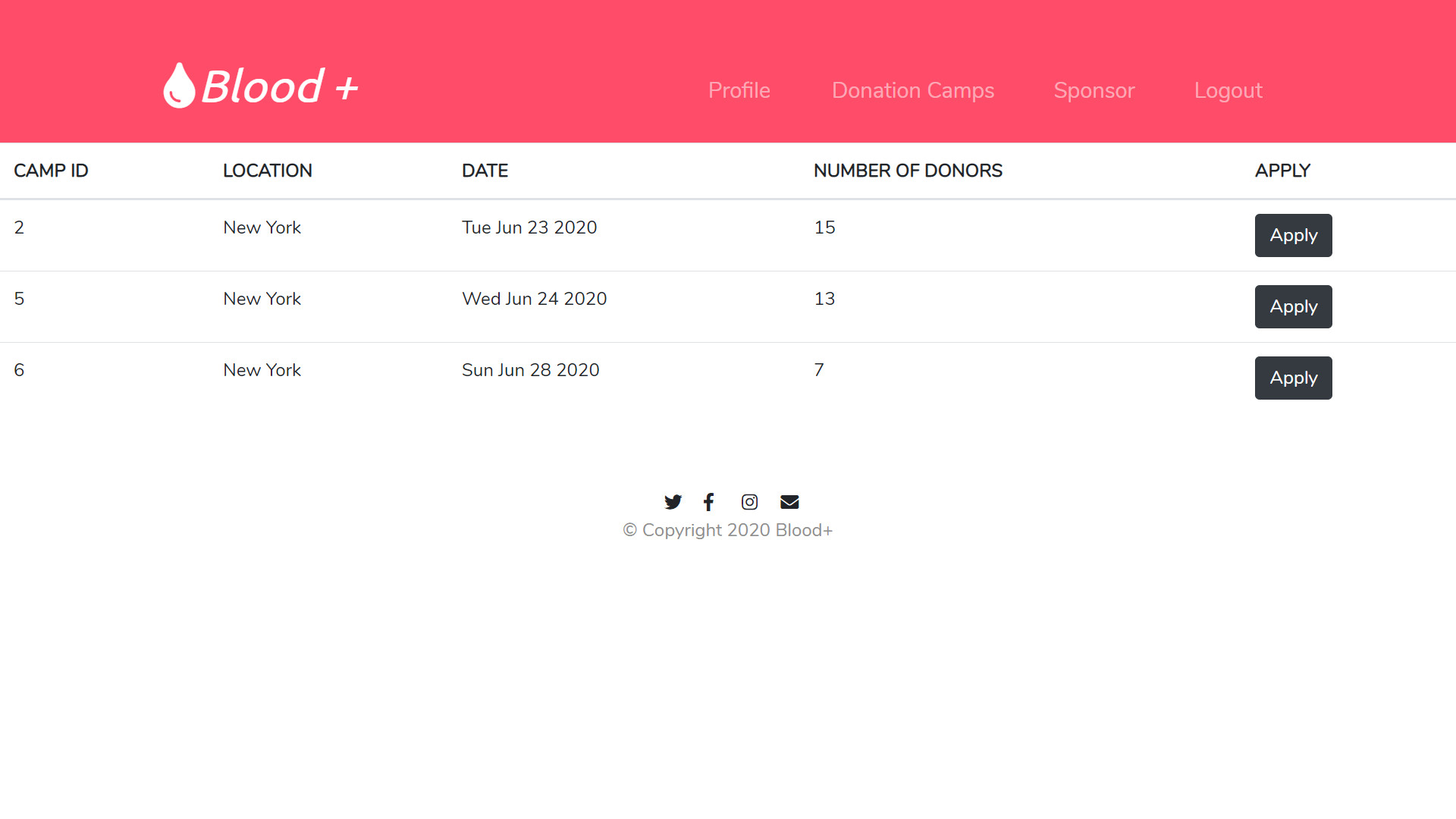
Snapshot :- Profile Page of a patient.

**Donor Profile:**



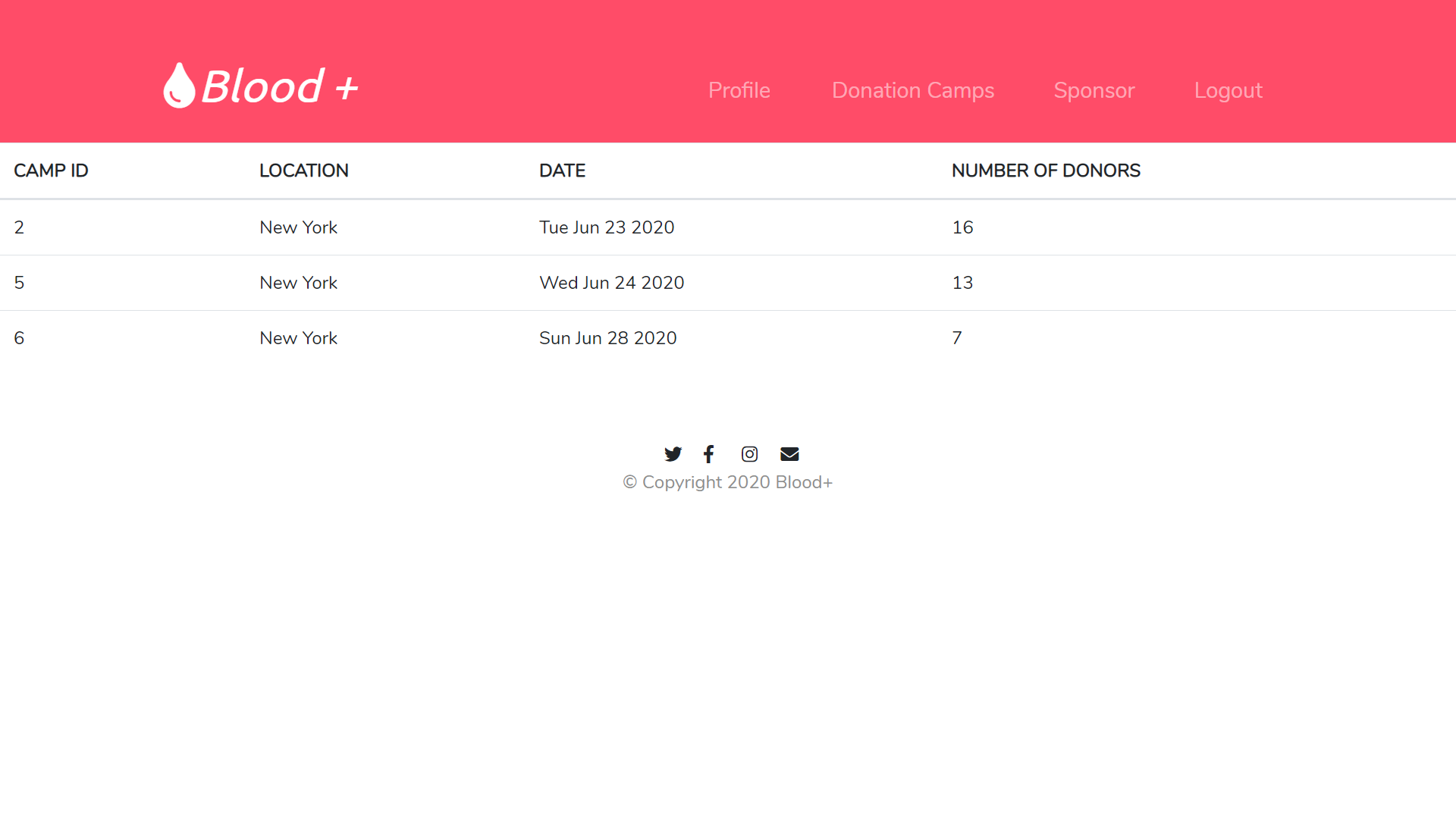
Snapshot :- Profile Page of a donor

**6. DONATION CAMP (Accessible to donors only.):-**



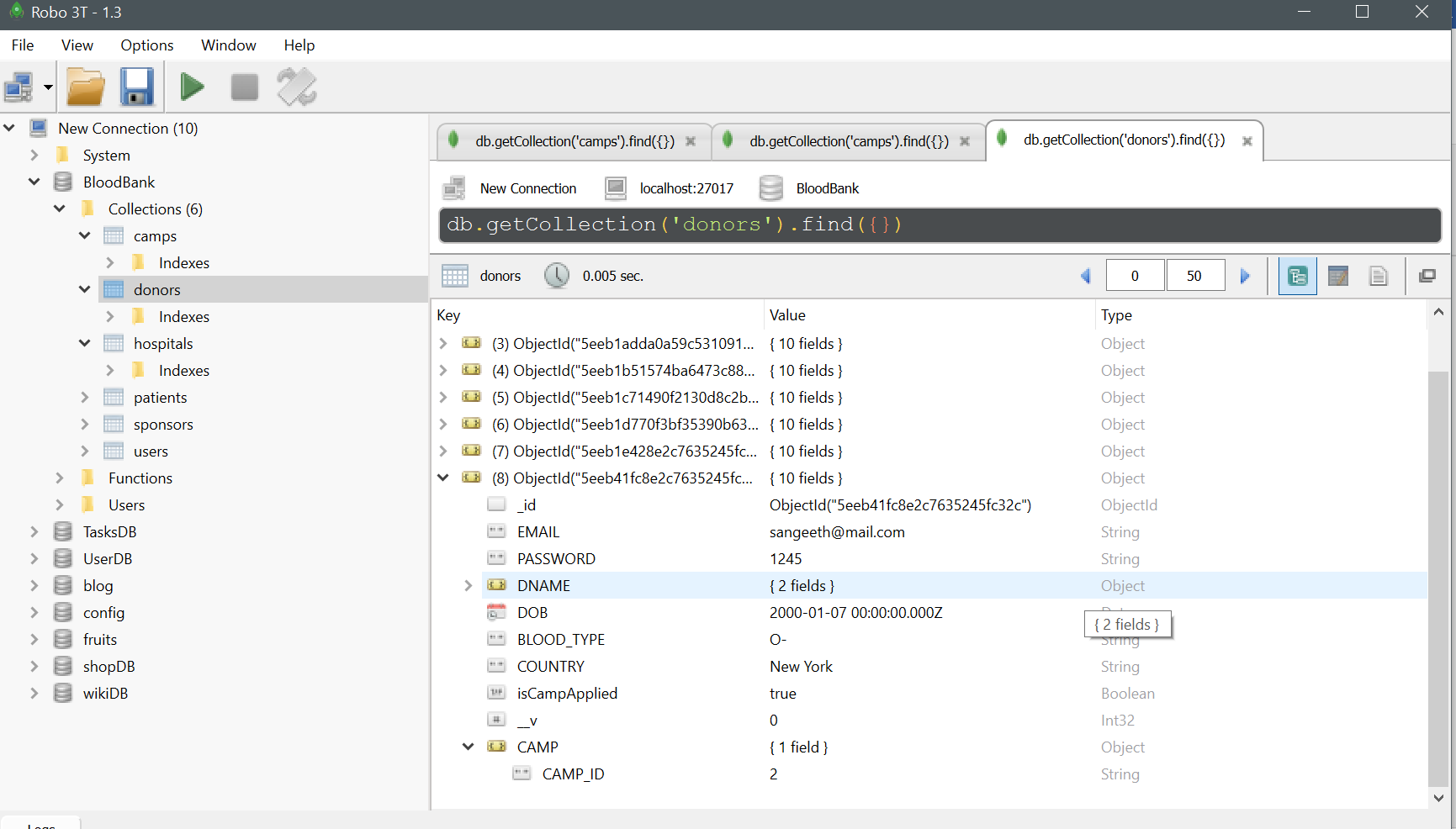
Snapshot :- This lists all the donation camps to be held in the donor’s location in the coming days. It lists the dates, camp\_ids and also an option to apply.

Here, we are applying to donate blood at camp\_id, 2 with a number of donors of 15.



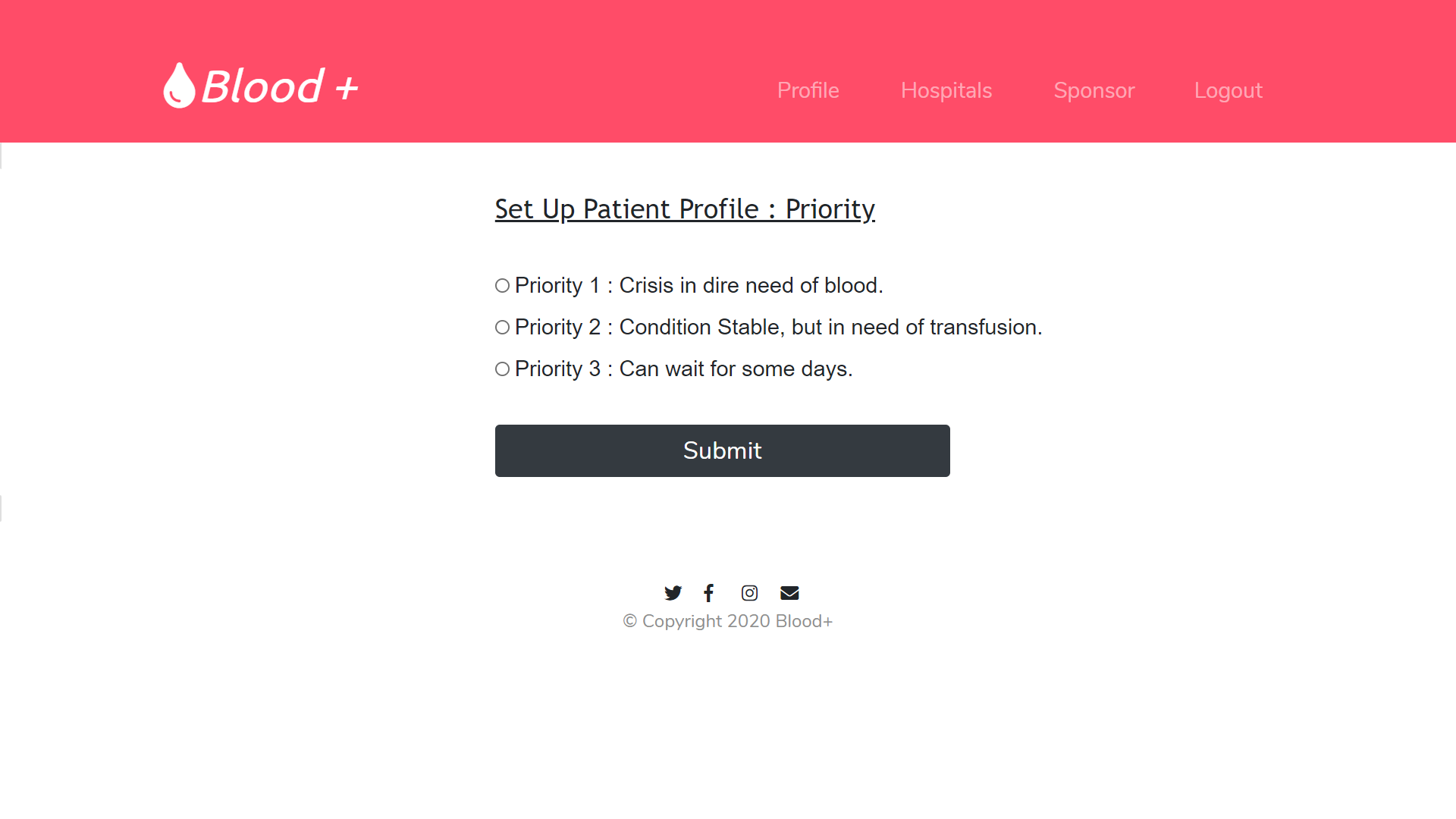
Snapshot :- Now, after applying to the donation camp, the number of donors is 16. And also, the apply option is now disabled.

Now, a foreign key is formed in the donor database for this particular user, as shown below.

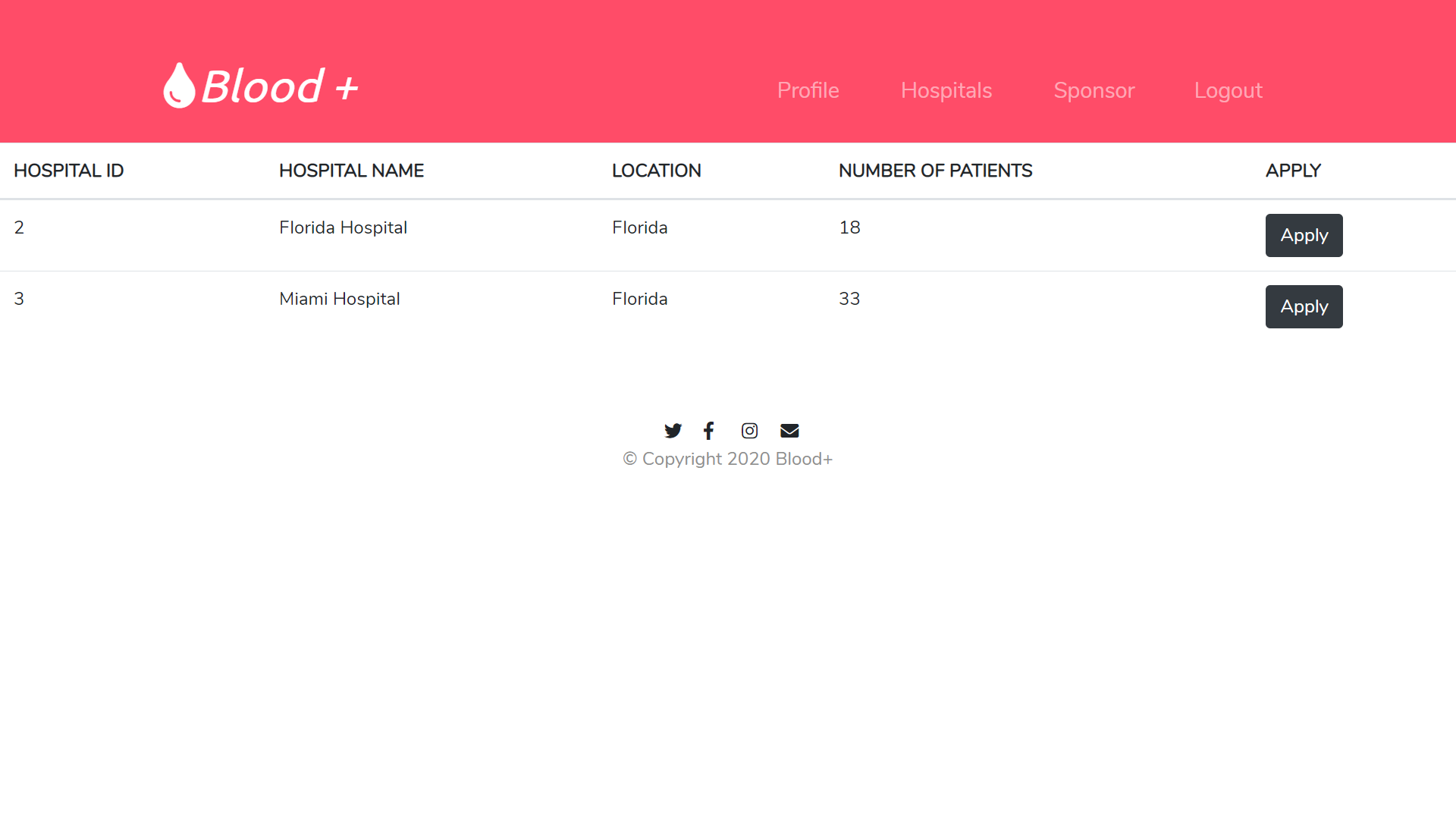


Snapshot :- The CAMP\_ID is now set to 2, which is the camp we applied for.

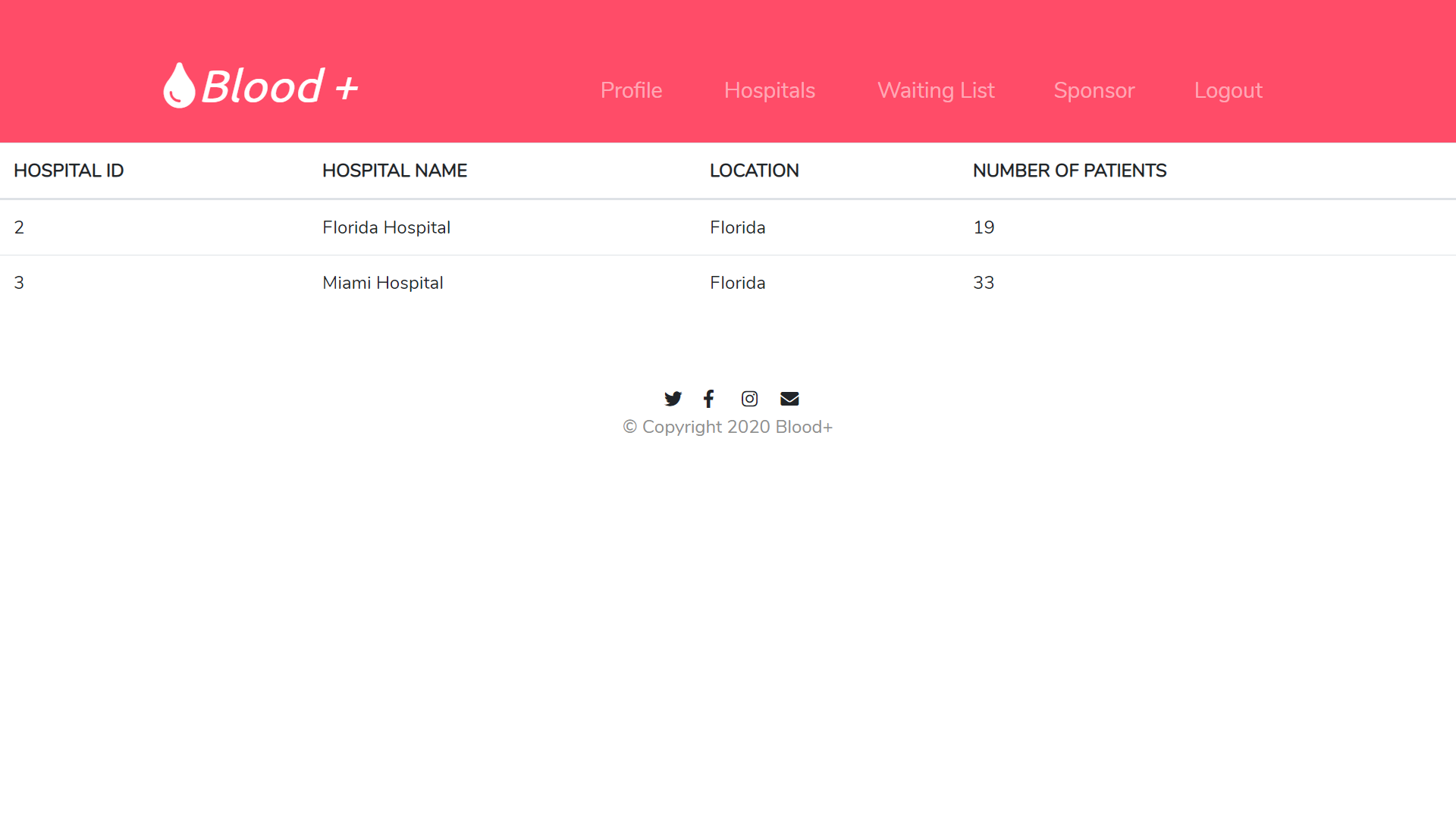
**7. HOSPITALS (Accessible to patients only.):-**

****

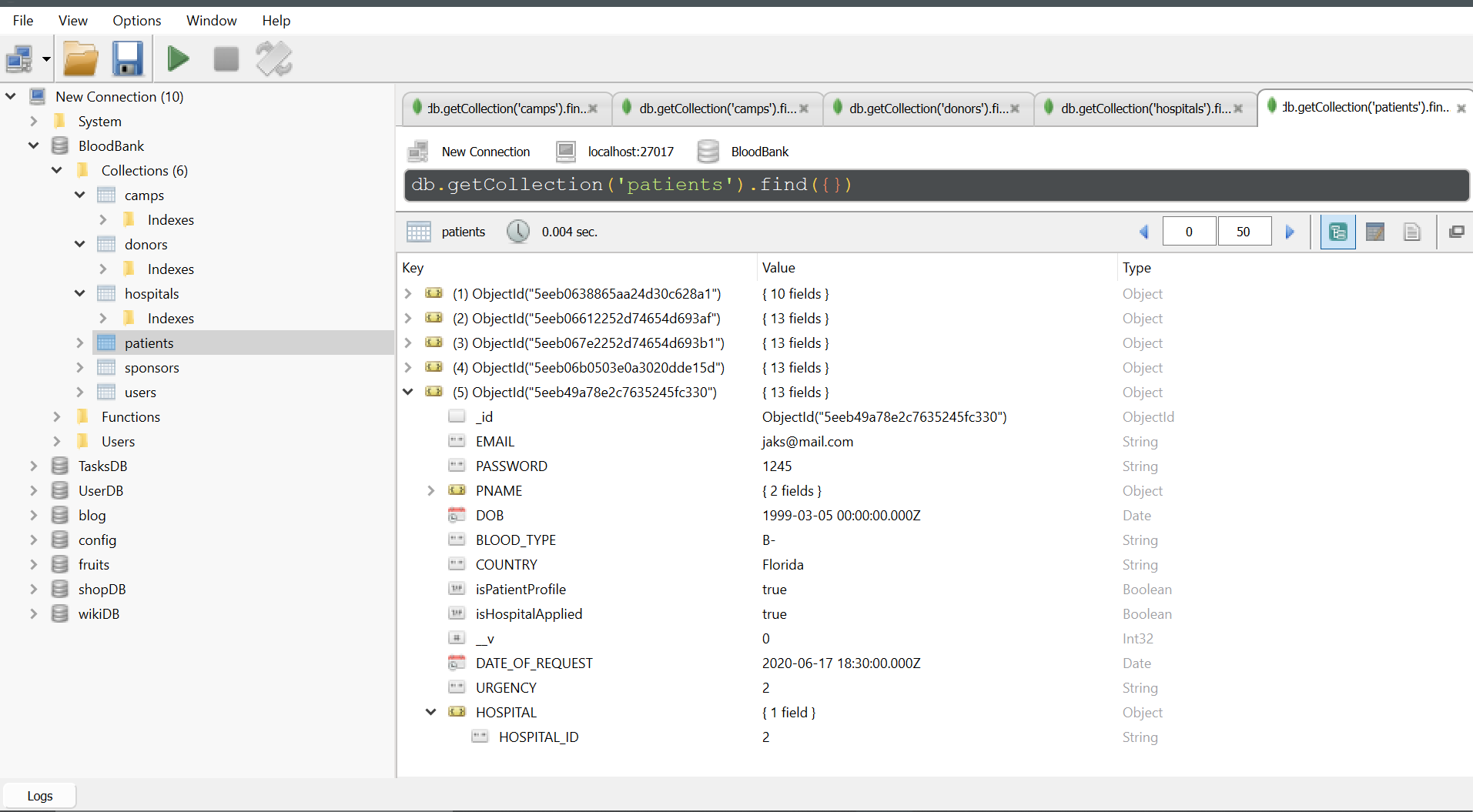
Snapshot :- When hospitals tab is clicked in the navbar, we are routed to set up the patient profile, ie, the urgency of the issue the patient is facing. Here, priority 2 is chosen.



Snapshot :-Now, the list of all the hospitals in the patients’ location is listed along with the number of patients the hospital already has. Now, let’s apply to Florida Hospital.

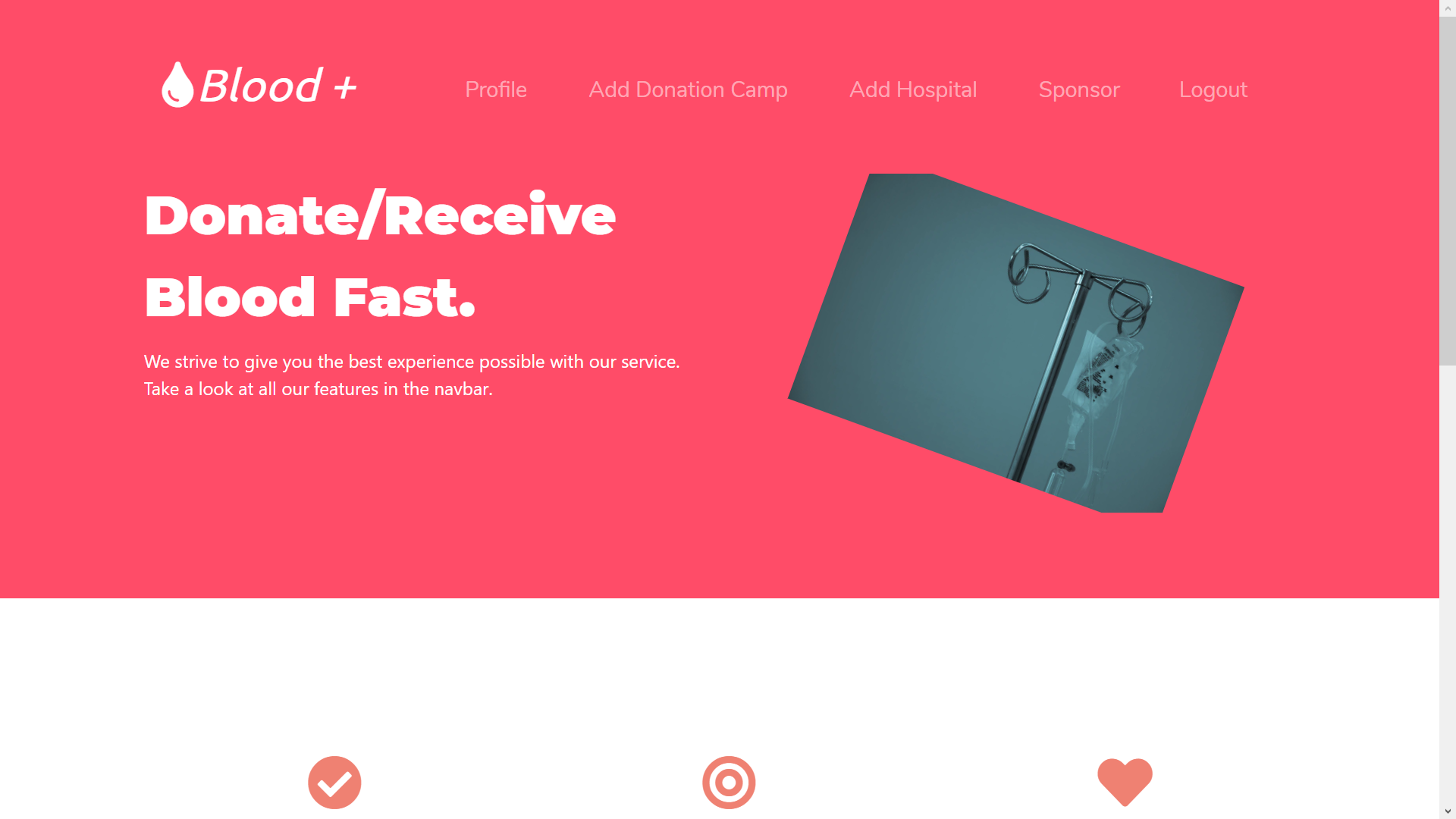


Snapshot :- Now, the apply option is removed and the count of the number of patients has increased to 19. Also, a foreign key is established for the patient in question, with the hospital ID.

****

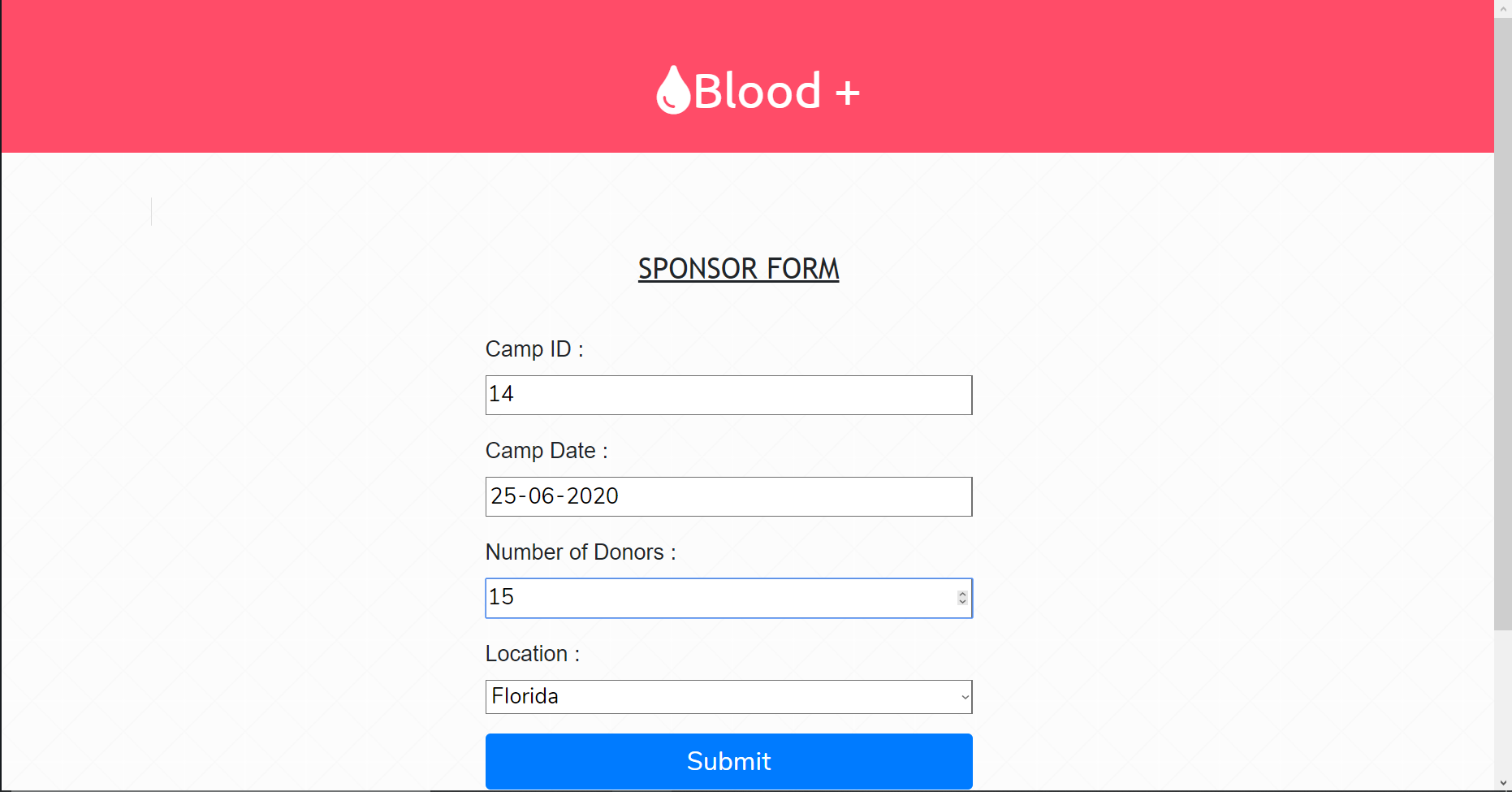
Snapshot :- The hospital id of 2 has been added to the patient in question.

**8. ADMIN ACCESS:**

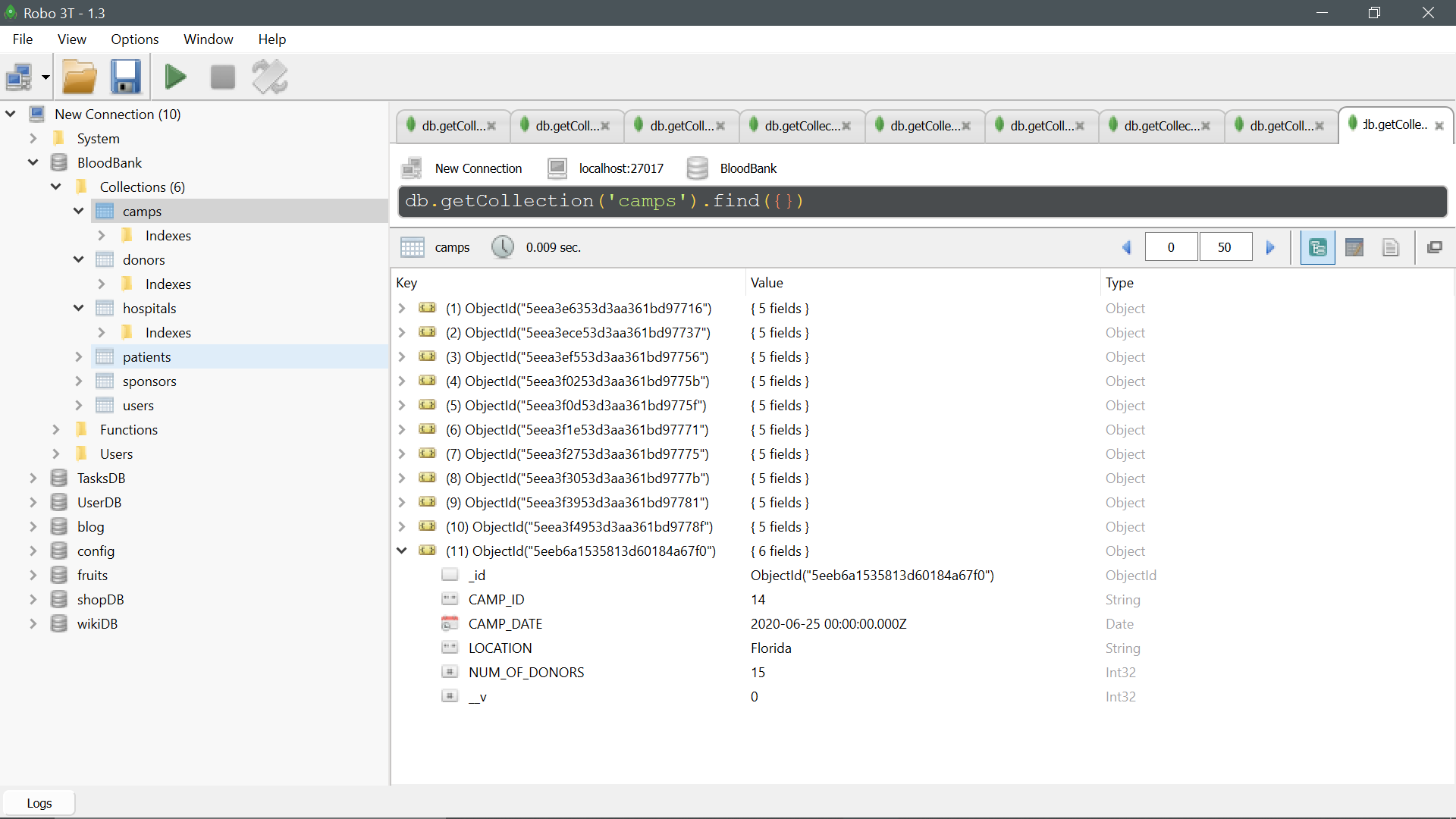
****

Snapshot :- Admin Access gives two new options: Add Donation Camp and Add Hospital.

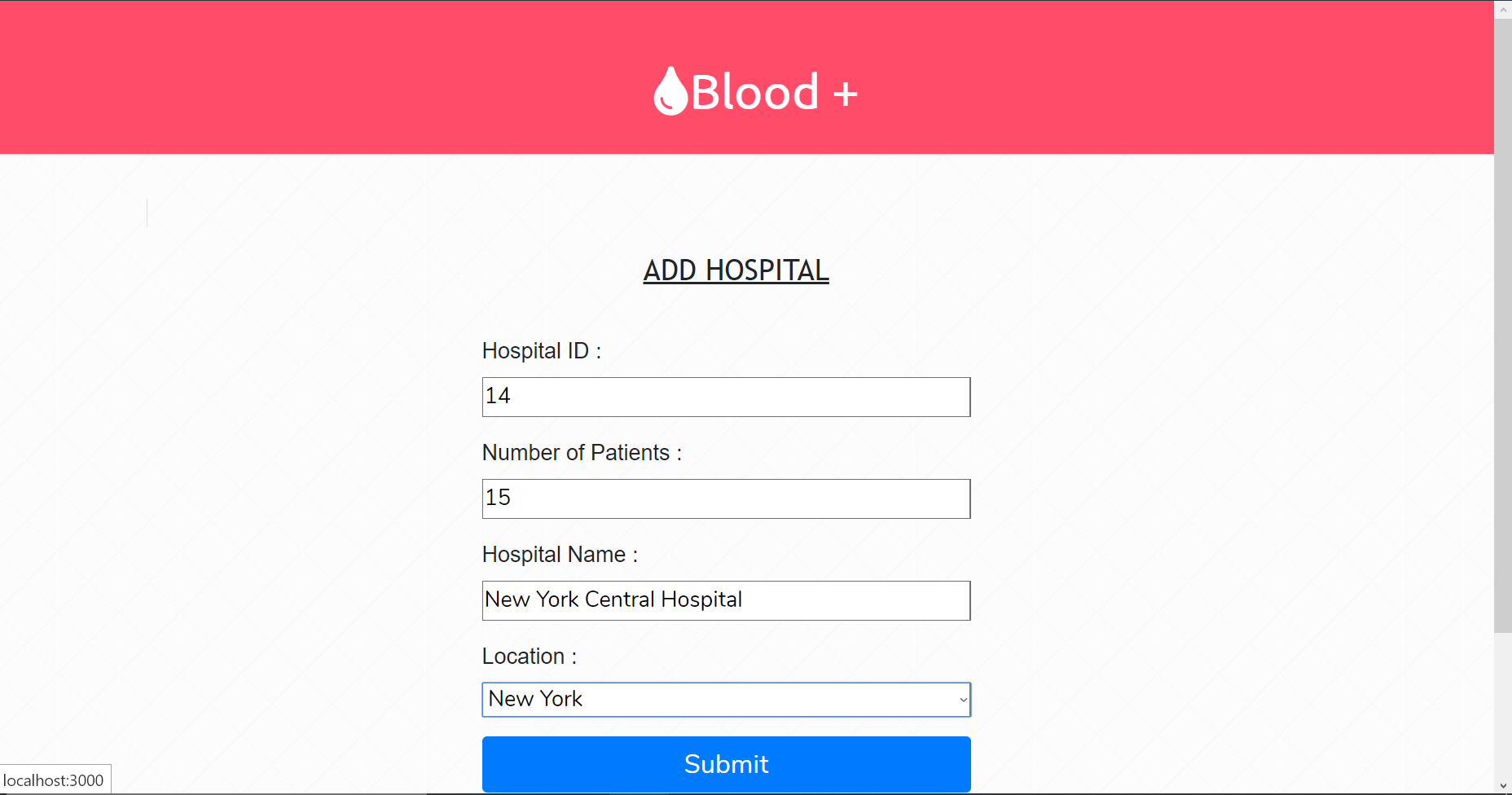
**ADD DONATION CAMP:**

****

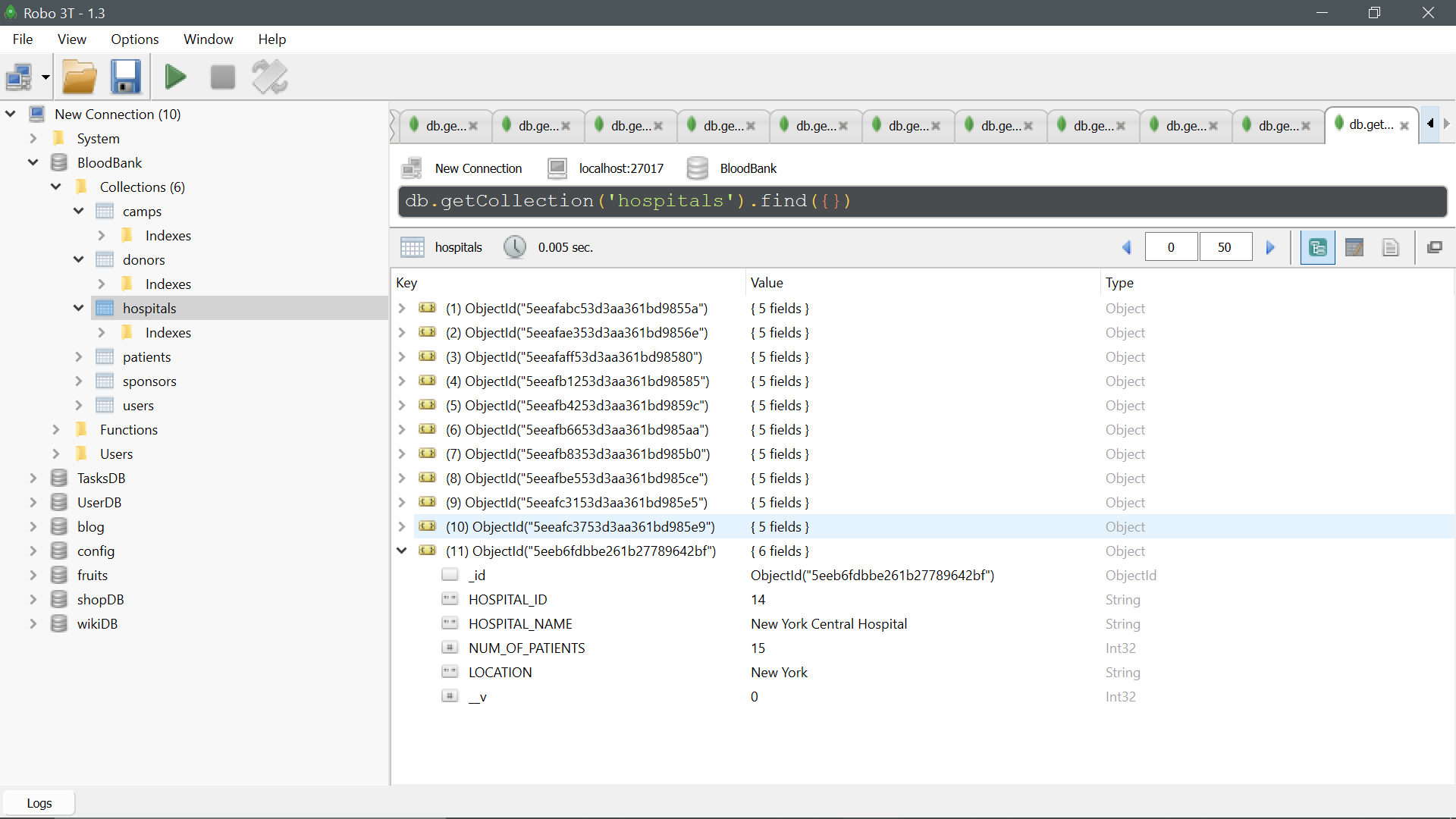
Snapshot :- Admin can add any donation camp that comes up.



Snapshot :- The Camp has been added to the database.

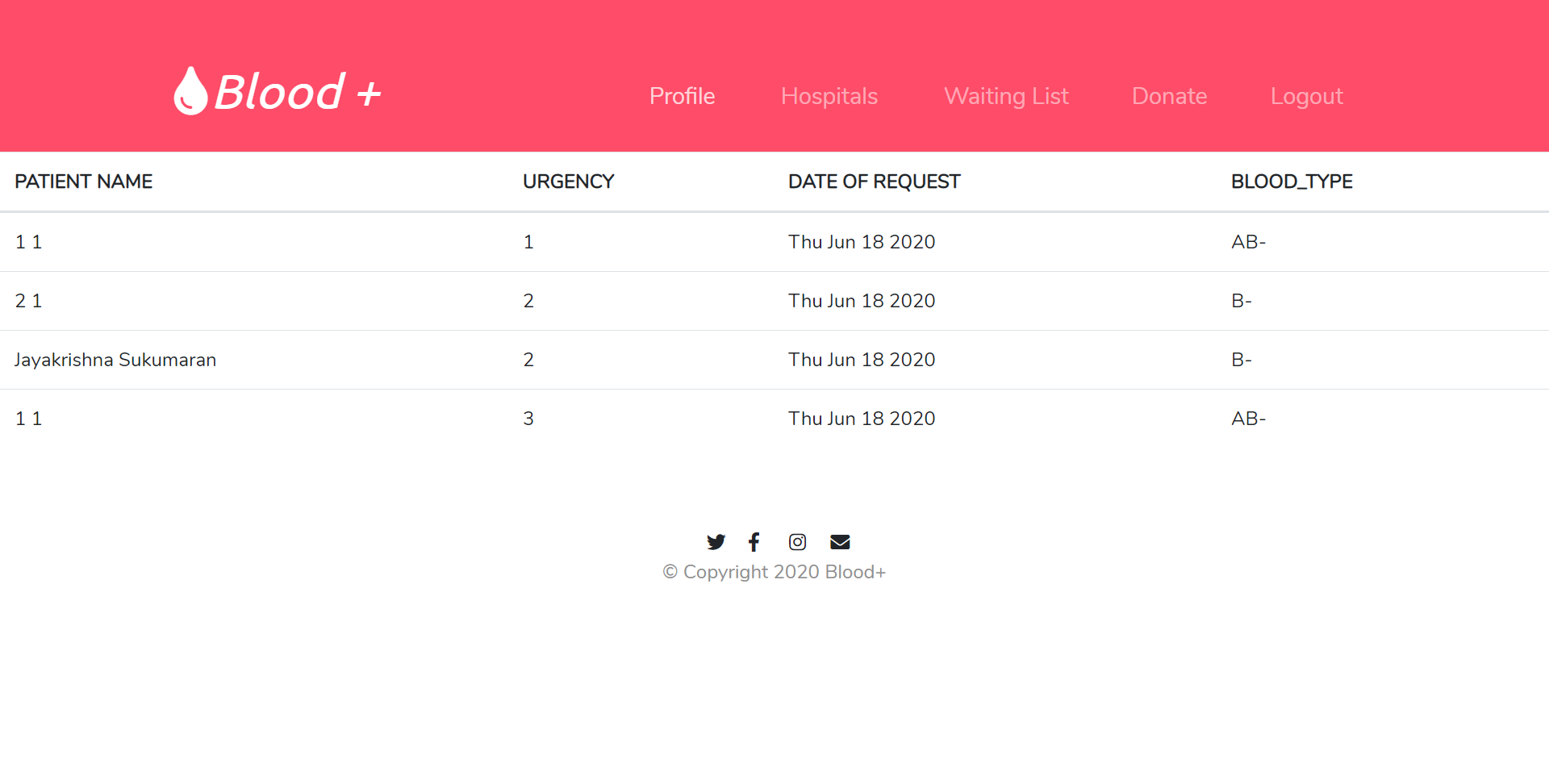


Snapshot :- Admin can add newly covered hospital to the database.



Snapshot :- The new hospital has been added.

**9. WAITING LIST:-**



Snapshot :- Now, the waiting list will also be optimized to show only the patients waiting at the applied Hospital, in this case, the Florida Hospital. After each donation camp, the blood bank admin will relieve some of these issues and the patient will move up the list. (for patients).

For donors and for unregistered users, the whole patients waiting list will be listed irrespective of the location.

**CONCLUSION**

This project is a venture to look into the existing system of Blood Bank Management and improve upon it. Several user-friendly software has been utilized for this project. This project shall prove very useful in future ventures where setup of a large and continuously running server is possible.

With more time more camps and hospitals and new locations can be added. This is a very limited setup now, with only 5 locations supported.

Now, we would like to take this opportunity to thank **Dr.M.Venkatesan**, for giving us the means and platform to undertake such a project. We also thank **Aishwarya Ma’am**, for the continued support and guidance provided to us throughout the semester.

**REFERENCES**

1. <https://devdocs.io/html/>
2. <https://devdocs.io/css/>
3. <https://nodejs.org/dist/latest-v13.x/docs/api/>
4. <https://mongoosejs.com/docs/documents.html#>
5. <https://www.w3schools.com/jquery/jquery_ref_html.asp>
6. <https://getbootstrap.com/docs/4.4/getting-started/introduction/>
7. <https://robomongo.org>
8. <http://www.passportjs.org/docs/>