



INSTITUTE FOR ADVANCED

COMPUTING AND

SOFTWARE DEVELOPMENT

AKURDI, PUNE

Documentation On

**“THINGS TO DONATE”**

E-DAC SEP- 2020

*Submitted By:*

**Group No: 64**

**Names & roll numbers**

Pooja Patil: 2121

Geetashri Belle: 2049

**Prashant Karhale Mr. Milind Arjun**

**Centre Coordinator Project Guide**

**Table of Contents**

**1. Introduction 4**

1.1 Document Purpose **4**

1.2 Problem Statement **4**

1.3 Product Scope **5**

1.4 Aim & Objectives **5**

**2. Overall Description 6**

2.1 Existing System Function 6

2.2 Benefits of thins to donate platform 7

2.3 Design and Implementation Constraints 7

**3.Functional Requirements 8**

3.1Admin Module 8

3.2 Donor Module 8

3.3Receiver Module 8

**4. Non-functional Requirements ……………………………………………9**

4.1 Performance Requirements……………………………………………………………………9

4.2 Safety Requirements…………………………………………………………………………..9

4.3 Security Requirements………………………………………………………………………...9

* 1. Software Quality Attributes…………………………………………………………………...9

**5. System Diagram................................................................................................................13**

5.1 Use Case Diagram………………………………………………………………………..13

5.2 Class Diagram................................................................................................................16

5.3 ER Diagram ............................................................................................................... …17

**6. Table Structure 18**

6.1 Receiver 18

6.2 Subcategory 18

6.3 Social work 18

6.4 Product 19

6.5 category 19

6.6 Donor 19

6.7 Product Requirement 20

6.8 Login 20

6.9 Receiver Request 20

**7. Snapshots 21**

**8. Conclusion 29**

**9. References 30**

**List of Figures**

**Use Case Diagrams 13**

Fig 1 Admin 13

Fig 2 Donor 14

Fig 3 Receiver 15

**Class Diagram 16**

Fig 5.2. Class Diagram………………………………………………………………………………16

**ER Diagram 17**

Fig 5.3. Class Diagram……………………………………………………………………………….17

1. **Introduction**

This document communicates the business requirements and scope for developing online donation platform system named as “Things to donate”. The scope of this document is to define the functional and non-functional requirements, business rules and other constraints requirements.

**1.1 Document Purpose**

There is no online donation system for people who willing to donate things/product without wasting their much time. Most of the time People donate their products to NGO’s by visiting them personally or sometimes due to time crunches they just let the products to ragman who tries to sell the product at higher prices, by cutting the commission out.

There is less or NO online platform available for the people who can donate effectively and easily. This proposed system will help people doing same.

**1.2 Problem Statement**

There is no online donation system for people who willing to donate things/product without wasting their much time. Most of the time People donate their products to NGO’s by visiting them personally or sometimes due to time crunches they just let the products to ragman who tries to sell the product at higher prices, by cutting the commission out.

There is less or NO online platform available for the people who can donate effectively and easily. This proposed system will help people doing same.

**1.3 Project Scope**

This project traverses a lot of areas and required to perform several researches to be able to achieve the project objectives. The area covers include:

* Things to donate platform: This includes study on how the things are being donated and received.
* Spring boot REST API at back-end & react JS as a front end used for the development of the application.
* General donor, receiver (NGO’s/Social Serving entities) as well as admin will be able to use the platform effectively.
* Platform means that the system will be available for access 24/7 except when there is a temporary server issue which is expected to be minimal.

**1.4 Aims & Objectives**

1. The main objective of project is to bridge the gap between donor & receiver.
2. This web-based application helps to donate things/products directly to NGO’s and at social serving events. This particular project deals with taking donations and arranging fundraising events so that people could donate effectively and easily.
3. People can choose their receiver from list of NGO’s or social serving events.
4. A non-profit organization that operates independently typically whose purpose is to address a social issue and to achieve reusability of product.
5. **Overall Description:**

**2.1 Existing system function:**

A “things to donate” platform can be used for donating products needed by NGO or Social Serving Entity. The donor who wants to donate products must first register into system. This can be done online. At this point, user has to supply some information such as: name, address, contact information user id password etc. After these details are worked out, the donor can login and donate product to needy ones. The receiver who wants to get products must first register into system. Receiver validation is done by admin if receiver found to be authorized then access is granted to user and can login into system. This can be done online. At this point, user has to information such as: name, address, contact information user\_id password, NGO/ social Serving registration ID (unique identity) etc. After these details are verified by admin, the receiver can login and get product of their need. Also, receiver can make request for product. This is cost free, non-profitable platform made for social serving. Donors are free to choose product of their choice from categories as cloths, electronics, food, medical equipment and books. Receiver can get product by requesting donor & if request is approved then receiver can contact donor and get the product.

The detailed existing system functions are listed as follows-

* Donor and receiver have to register before login into system by providing essential information.
* Receiver validation is done by admin if receiver found to be authorized then access is granted and can login into system.
* Donor can choose product from category, subcategory to be donated.
* Receiver can get product by sending request to donor & if request is approved then receiver can contact donor and get the product.
* Donor has authority to accept or reject request from receiver.
* And report is generated about donated products, their availability etc.

**2.2 Benefits of Online things to donate platform**

* This online platform is fully functional and flexible.
* It is very easy to use.
* This platform helps people to donate things at ease.
* Platform gives opportunity to people who really want to contribute to social activity without investing more time.
* It saves a lot of time.
* The application acts 24/7.
* It provides platform for NGO and social serving entity to send product requirement.

**2.3 Design and Implementation Constraints:**

* The application will use React JS along with Bootstrap and Spring Boot rest as main web technologies.
* Several types of validations make this web application a secured one and SQL Injections can also be prevented.
* Since it is a web-based application, internet connection must be established.
* The platform can be used on PCs, laptop’s and mobile phones and will function via internet or intranet in any web browser.

**3. Functional Requirement**

* 1. **Admin module**
* Admin can login to the system.
* View the list of product donated product list.
* Add new receiver..
* View receiver
* Reject receiver request.
* Add social Work.
* View Social workers working in the society.

### 

**3.2 Donor Module**

* The donor is the user of the system who wants to donate the product.

* He is also able to view the Categories of products to Donate.

**3.3 Receiver Module**

* Receiver is one who will receive the donated product.
* Receiver can be NGO’s or Social Serving entities.
* Receiver is able to view the Categories of products, select & freeze the product.
* Receiver can send requirement for product of their need.

# 

**4. Non-Functional Requirements**

**4.1 Performance Requirements:**

To provide the good interface so Admin can easily get the information about the Donors who want to donate and Receivers who want to receive products ,so that website become user-friendly is beneficial to all. System should store the all-database record of Donors, Receivers And Product to tracking their information. System always connected to the server. Here are some examples of poor performance requirements to avoid:

4.1.1 **Unquantifiable:** Must work faster or must have good performance

4.1.2 **Ambiguous:** The application should load within an adequate timeframe.

4.1.3 **Unrealistic:** The application should load with in a seconds at all times.

4.1.4 **Unverifiable:** Flexible, easy, sufficient, safe, ad hoc, adequate, user-friendly, usable, when required, if required, appropriate, fast, portable, lightweight, small, large, maximize, minimize, robust, quickly, easily, clearly, you get the idea.

**4.2 Safety Requirements:**

4.2.1 Website should keep as user-friendly.

4.2.2 Check the employee details.

* + 1. Secure your computer.

4.2.4 Back up our files.

.

**4.3 Security Requirements**

4.3.1 The functional describes the security aspect of software requirements.

4.3.2 All the passwords of Admin, Donor and Receiver should be protected for privacy whatever constraints provides.

4.3.3 Only Admin have rights to access the data.

4.3.4 Avoid passing along private information online.

4.3.5 The database should be safe from hacker and any cyber-attack. Also, interface should be keep protected.

**4.4 Software Quality Attributes:**

The system should available easily to user that means system should run on various operating system and also hardware. The system not be complicated to user.

**Accessibility:**

The website will be accessible to Admin ,Donor and Receiver so Donor can view request of Receiver . The Admin will give approval to Donate things to Receiver.

**Compatibility:**

The website will be tested for working with multiple user because at same time there are three people get logged in searching the information. So system have robust.

**Effectiveness:**

The Website will good for handled it. there is no complex data to user get confused.

Software should be effective.

**Maintainability:**

Maintainability of a software system is the ease with which the system can be maintained. If the Mean Time between Failures (MTBF) is low or Mean Time to Repair (MTTR) is high for the system being developed, then the maintainability of the system is considered low.

.

**Operating Environment:**

Server Side:

**Processor:** Intel® Xeon® processor 3500 series

**HDD:** Minimum 500GB Disk Space

**RAM:** Minimum 2GB

**OS:** Windows 8.1, Linux 6

**Database:**  MySQL

Client Side (minimum requirement):

**Processor:** Intel Dual Core

**HDD:** Minimum 80GB Disk Space

**RAM:** Minimum 1GB

**OS:** Windows 7, Linux

**Design and Implementation Constraints:**

* The application will use Ajax, JavaScript, jQuery and css as main web technologies.
* HTTP and FTP protocols are used as communication protocols. FTP is used to upload the web application in live domain and the client can access it via HTTP protocol.
* Several types of validations make this web application a secured one and SQL Injections can also be prevented.
* Since Society Management system is a web-based application, internet connection must be established.
* The Society Management System will be used on PCs and will function via internet or intranet in any web browser.

**Specific Requirement**

**External Interface Requirements:**

User Interfaces:

* All the users will see the same page when they enter in this website. This page asks the users a username and a password.
* After being authenticated by correct username and password, user will be redirect to their corresponding profile where they can do various activities.
* The user interface will be simple and consistence, using terminology commonly understood by intended users of the system. The system will have simple interface, consistence with standard interface, to eliminate need for user training of infrequent users.

Hardware Interfaces:

* No extra hardware interfaces are needed.
* The system will use the standard hardware and data communication resources.
* This includes, but not limited to, general network connection at the server/hosting site, network server and network management tools.

Application Interfaces:

**OS:** Windows 7, Linux

**Web Browser:**

The system is a web-based application; clients need a modern web browser such as Mozilla Firebox, Internet Explorer, Opera, and Chrome. The computer must have an Internet connection in order to be able to access the system.

Communications Interfaces:

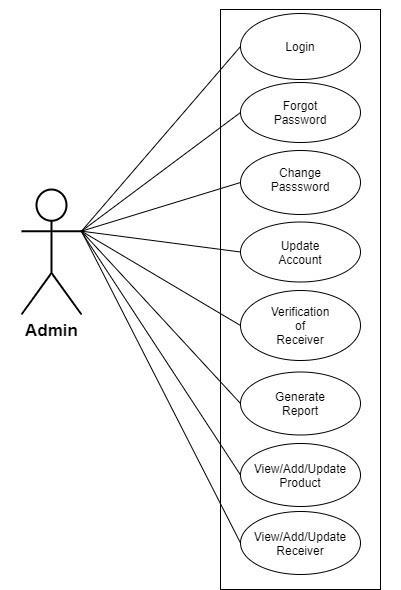
* This system uses communication resources which includes but not limited to, HTTP protocol for communication with the web browser and web server and TCP/IP network protocol with HTTP protocol.

This application will communicate with the database that holds all the booking information. Users can contact with server side through HTTP protocol by means of a function that is called HTTP Service. This function allows the application to use the data

1. **System Design**

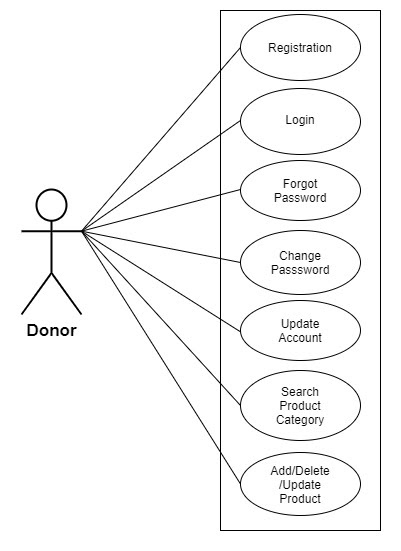
**5.1 Use Case Diagram**

**Admin**

****

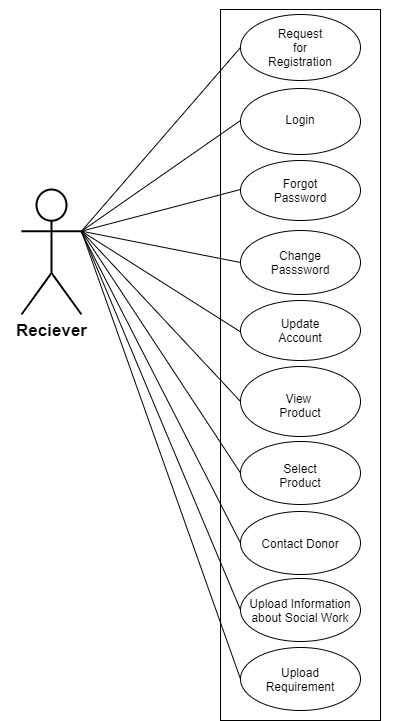
*Fig. 5.1.1 Use case diagram for Admin*

**Donor**

****

*Fig. 5.1.2 Use case diagram for Donor*

**Receiver**

****

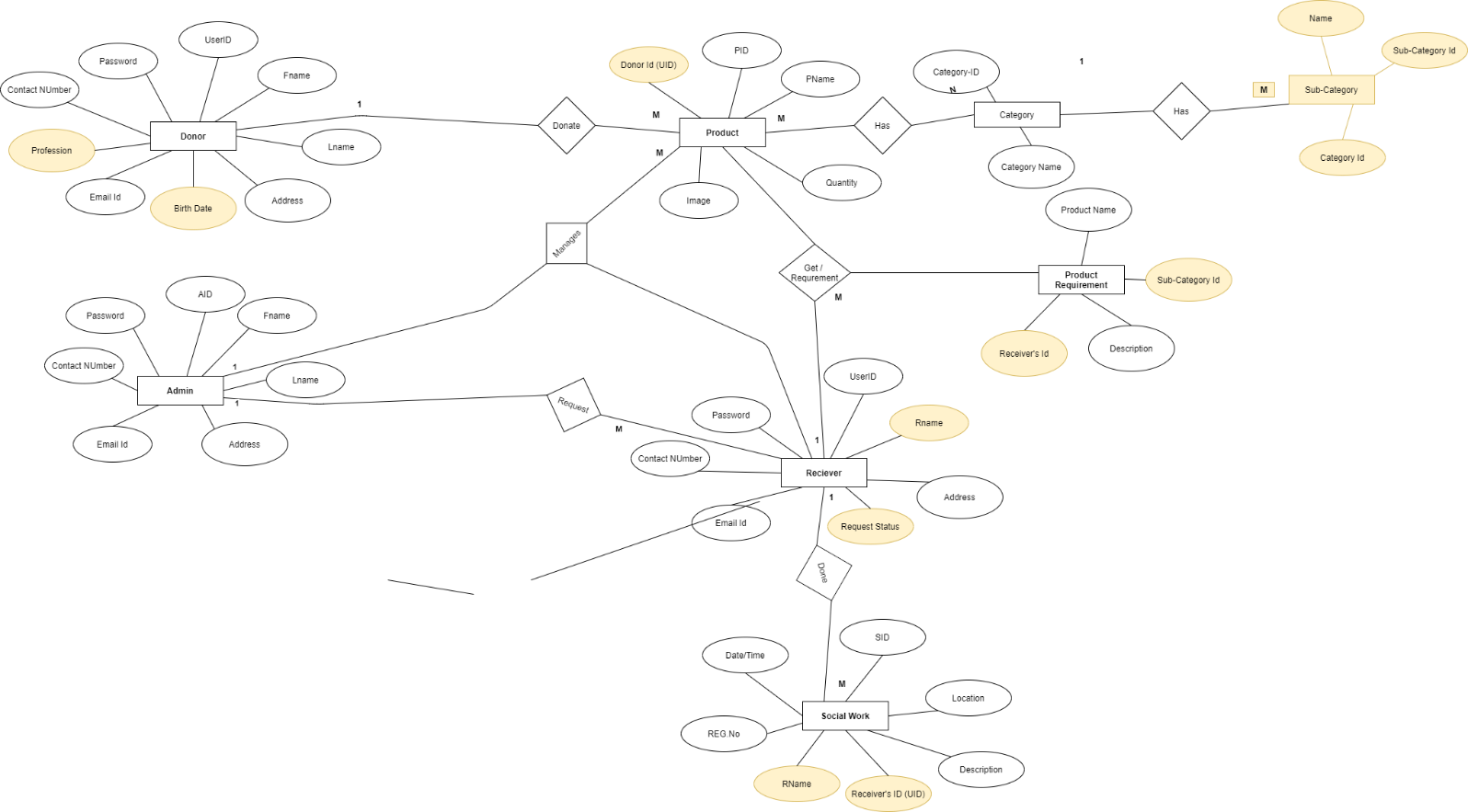
*Fig. 5.1.3 Use case diagram for Receiver*

**5.2 Class Diagram:**



*Fig 5.2. Class Diagram*

**5.3 ER Diagram**



*Fig 5.3 ER Diagram*

1. **Table Structure:**

**6.1] Receiver**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Rid | Integer | No | Primary key | Null | Customer User ID |
| Uno | Integer | No | Foreign Key | Null | Customer username |
| Reqid | Integer | No | Foreign Key | Null | Request ID from Receiver Request |

**6.2] Sub Category**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Sid | Integer | No | Primary  Key | Null | Sub category ID |
| cid | Integer | No | Foreign Key | Null | Category ID from Category Table |
| Sname | Varchar(45) | No |  | Null | Sub Category Name |

**6.3] Social Work**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Wid | Integer | No | Primary  Key | Null | Social work ID |
| Rid | Integer | No | Foreign Key | Null | Receiver ID |
| donateditem | Varchar(45) | No |  | Null | Donated items |
| recipient | Varchar(45) | No |  | Null | Recipient Name |
| location | Varchar(45) | No |  | Null | Recipient Location |
| datetime | Datetime | No |  | Null | Socialwork Datetime |
| description | Varchar(45) | No |  | Null | Description about Social work |

**6.4] Product**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Pid | Integer | No | Primary key | Null | Product id |
| Pname | Varchar(45) | No |  | Null | Product Name |
| Did | Integer | No | Foreign Key | Null | Donor id from Donor table |
| Image | Varchar(45) | No |  | Null | Product Image |
| quantity | Double | No |  | Null | Product Quantity |
| pstatus | Varchar(45) | No |  | Null | Product Status(Available) |

**6.5] Category**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| cid | Integer | No | Primary key | Null | Category id |
| Cname | Varchar(45) | No |  | Null | Category Name |

**6.6] Donor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Did | Integer | No | Primary  Key | Null | Donor id |
| Fname | Varchar(45) | No |  | Null | Donor First Name |
| Lname | Varchar(45) | No |  | Null | Donor Last Name |
| Dob | Varchar(45) | No |  | Null | Donor date of birth |
| Email | Varchar(45) | No |  | Null | Donor Email Id |
| address | Varchar(45) | No |  | Null | Donor Address |
| profession | Varchar(45) | No |  | Null | Donor Profession |
| contactno | Varchar(45) | No |  | Null | Contact no of donor |
| Uno | Integer | No | Foreign key | Null | User no from Login table |

**6.7] Product Requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Prid | Integer | No | Primary  Key | Null | Product Requirement id |
| Pname | Integer | No |  | Null | Product Name |
| Sid | Integer | No | Foreign key | Null | sid from Subcategory  table |
| Rid | Varchar(45) | No | Foreign key | Null | rid from receiver table |
| description | Varchar(45) | No |  | Null | Vehicle id from Vehicle  table |

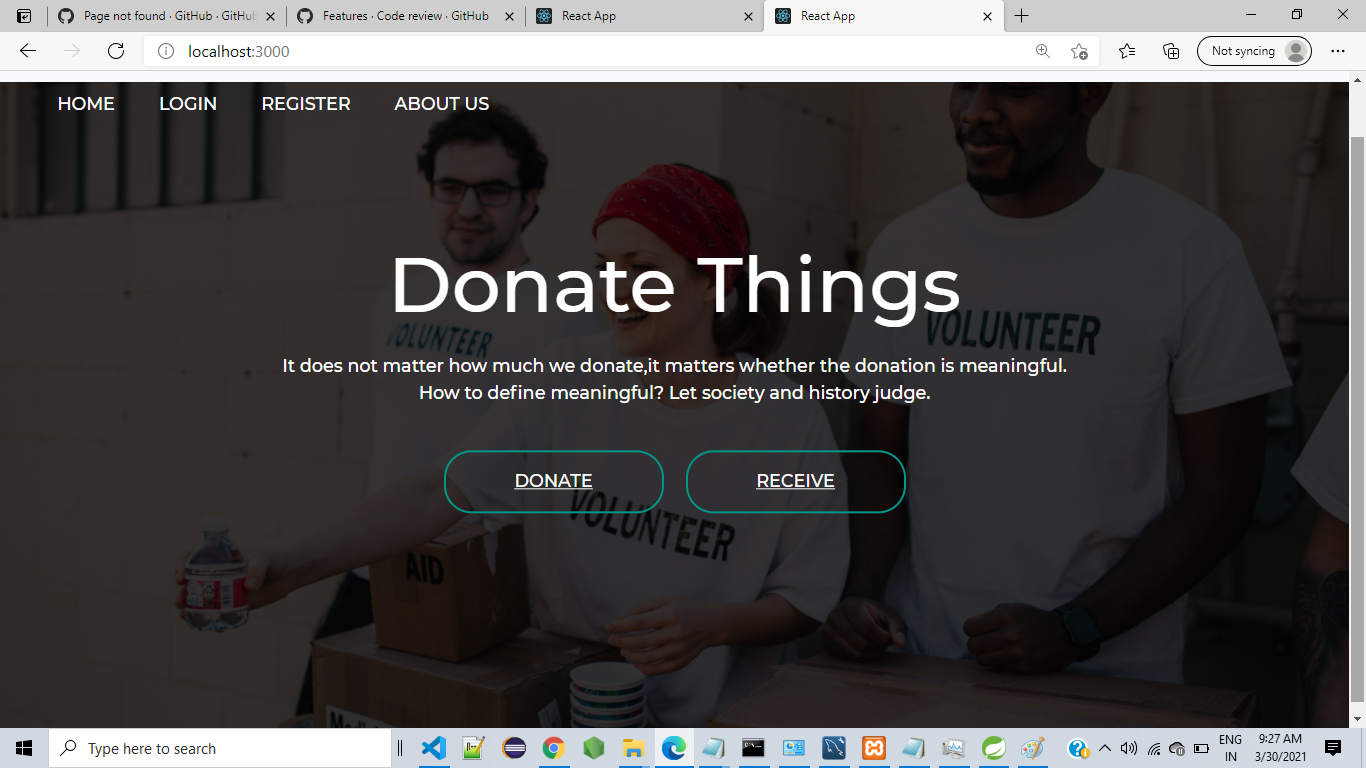
**6.8] Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Uno | Integer | No | Primary  Key | Null | User No |
| Uid | Varchar(45) | No | Unique | Null | User Unique ID |
| Pwd | Varchar(45) | No |  | Null | User Password |
| Role | Varchar(45) | No |  | Null | User Role |

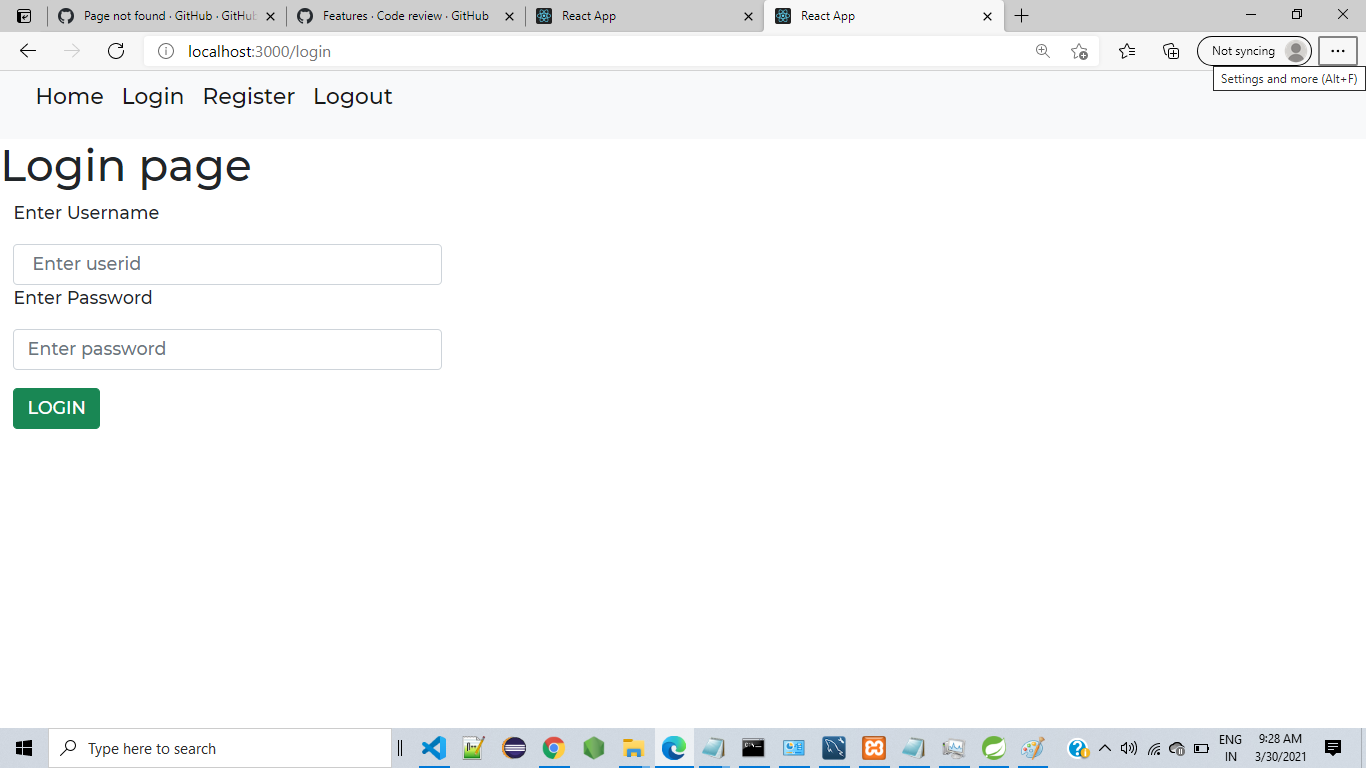
**6.9] Receiver Request**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Reqid | Integer | No | Primary  Key | Null | Receiver Request id |
| Rname | Varchar(45) | No |  | Null | Receiver name |
| uniqueid | Varchar(45) | No |  | Null | Receiver uniqueid |
| Email | Varchar(45) | No |  | Null | Receiver email id |
| Pwd | Varchar(45) | No |  | Null | Receiver password |
| Uid | Varchar(45) | No |  | Null | User id |
| address | Varchar(100) | No |  | Null | Receiver address |
| requeststatus | Integer |  |  |  | Receiver request status |
| contactno | Varchar(45) | No |  | Null | Receiver contact no |

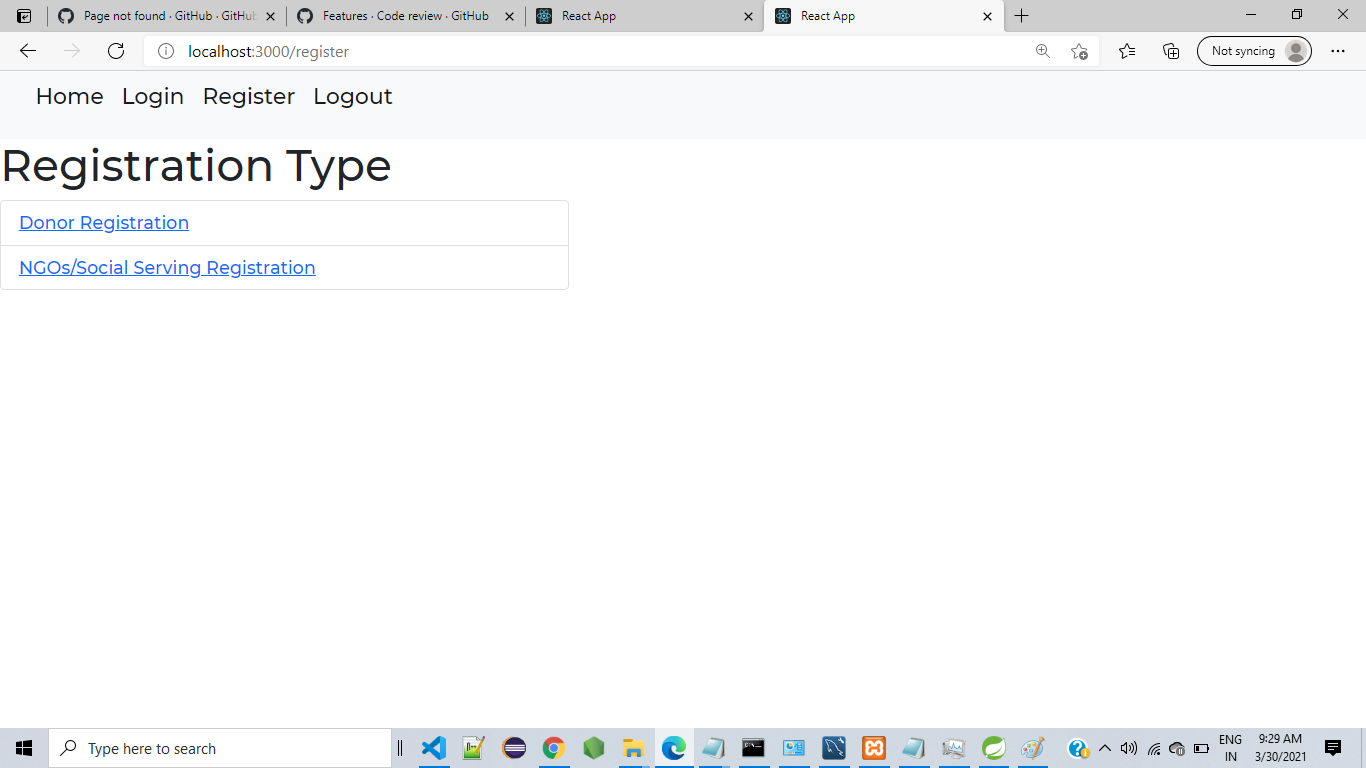
1. **Snapshots**
   1. **Home Page**

****

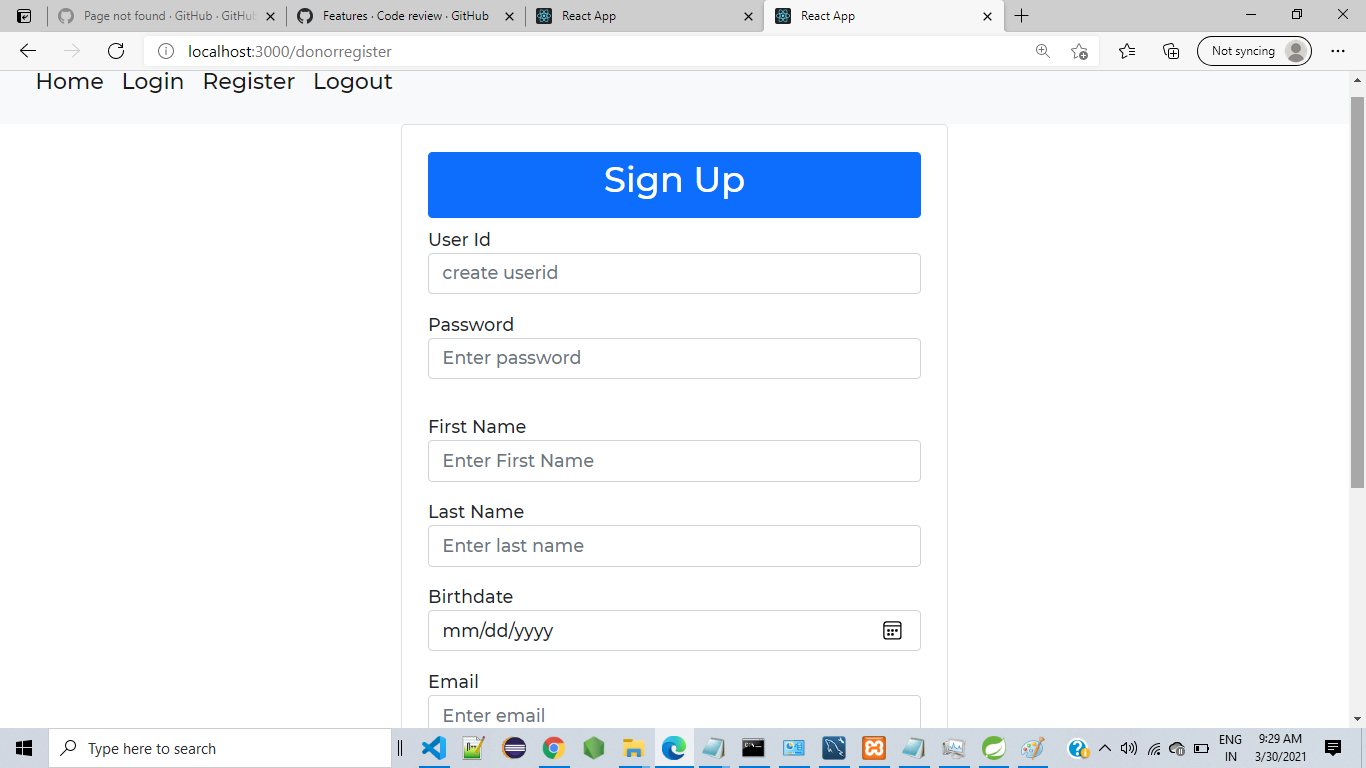
* 1. **Login Page**

****

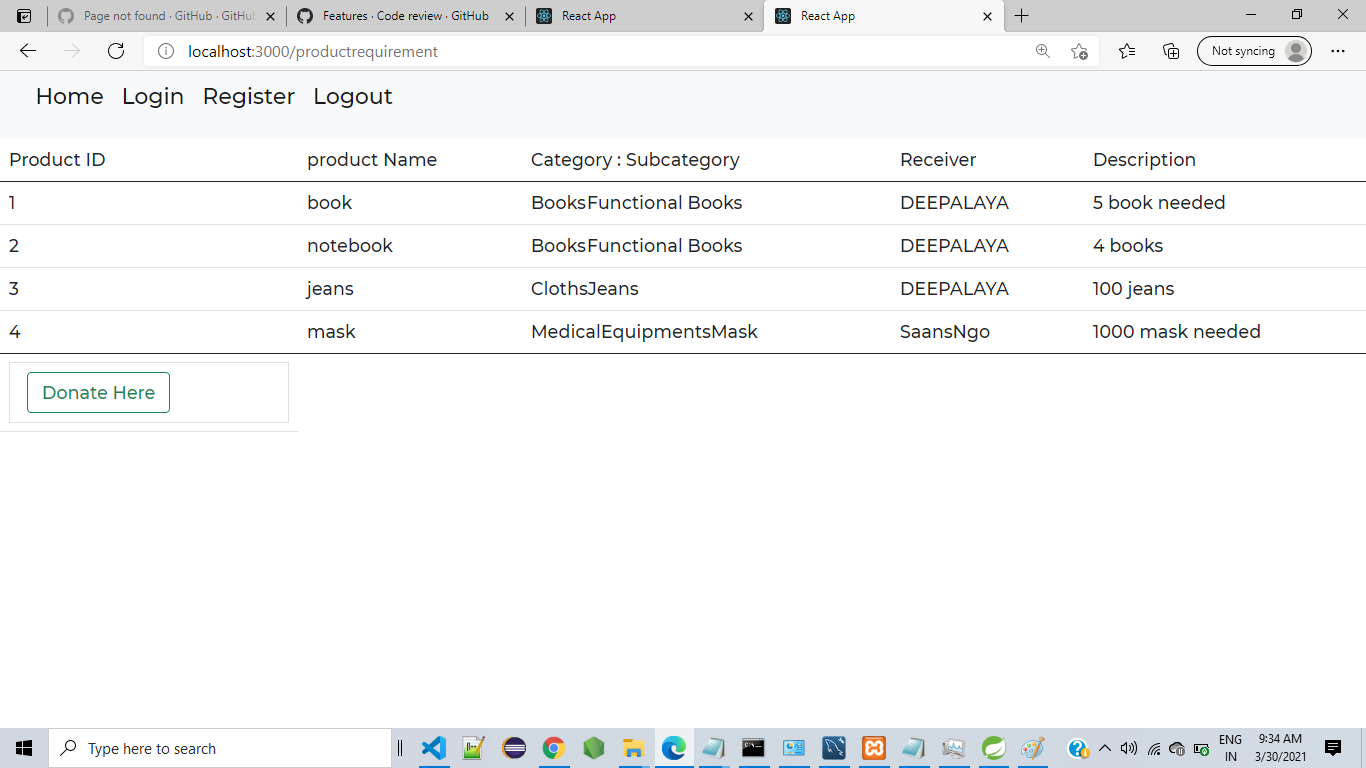
**7.3 Registration Page**

****

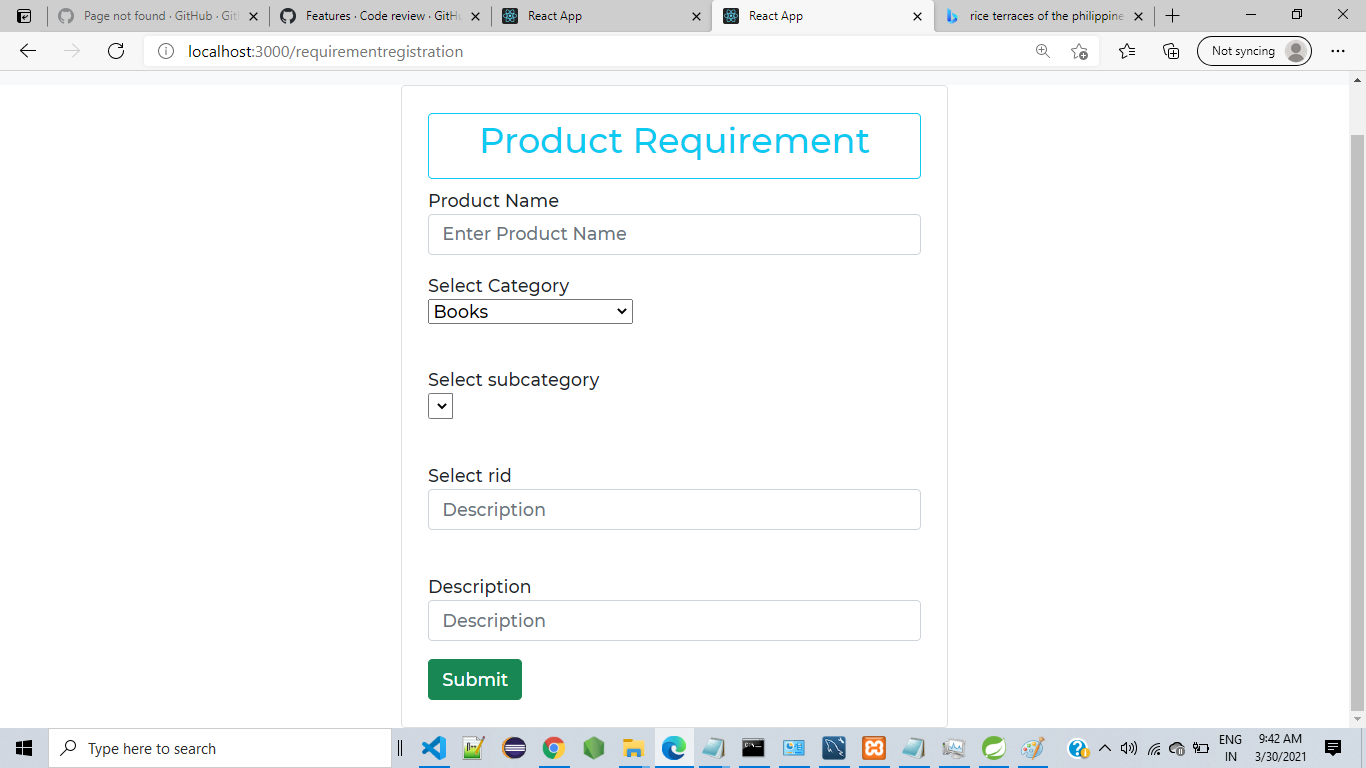
**7.3 Donor Registration**

****

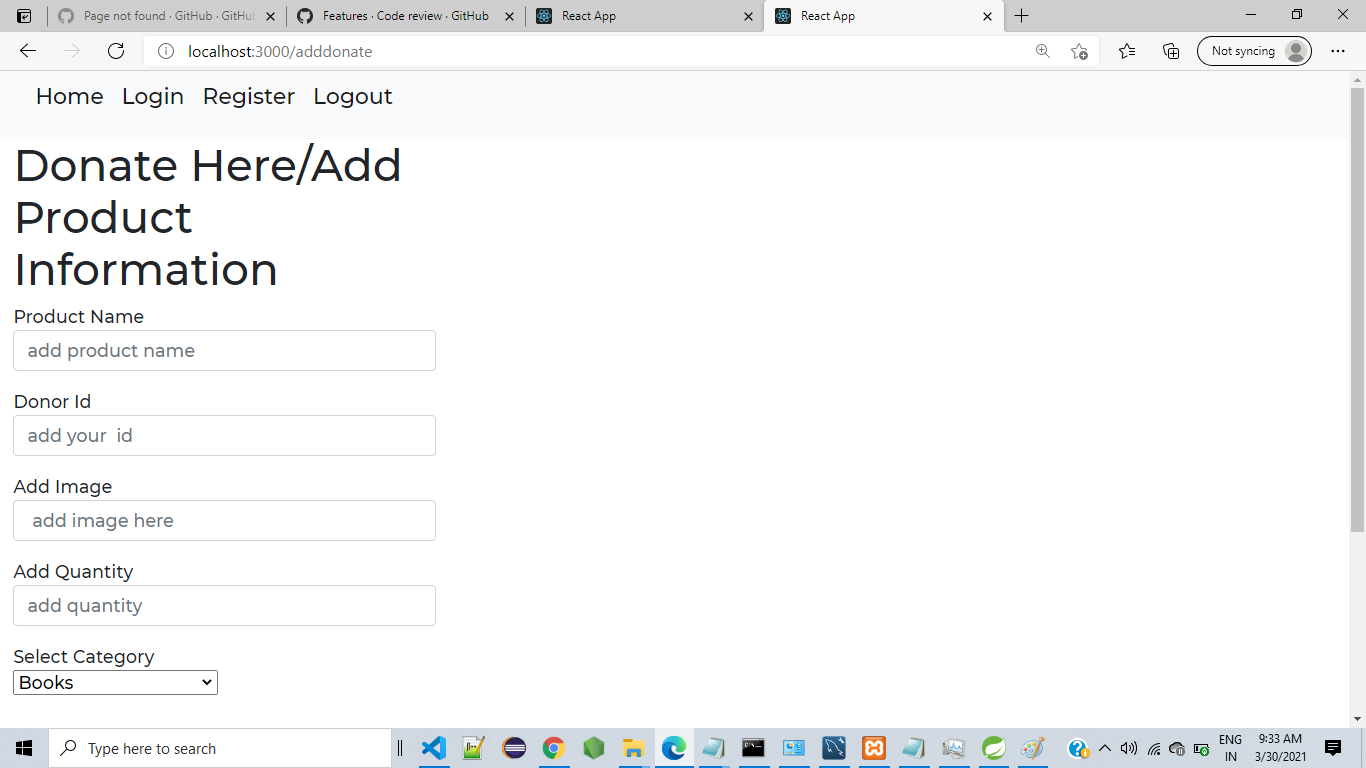
**7.4 Donor Side Product Req. List**

****

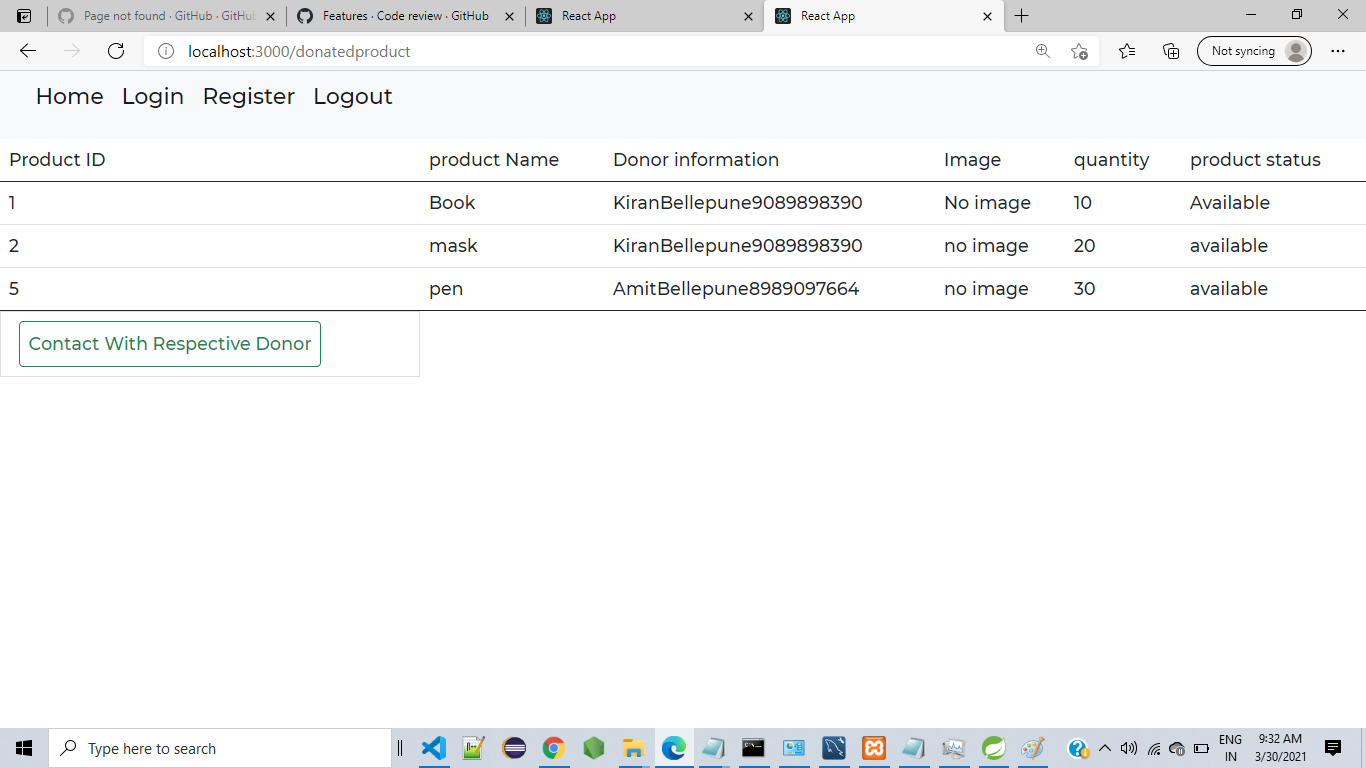
**7.5 Product Requirement**

****

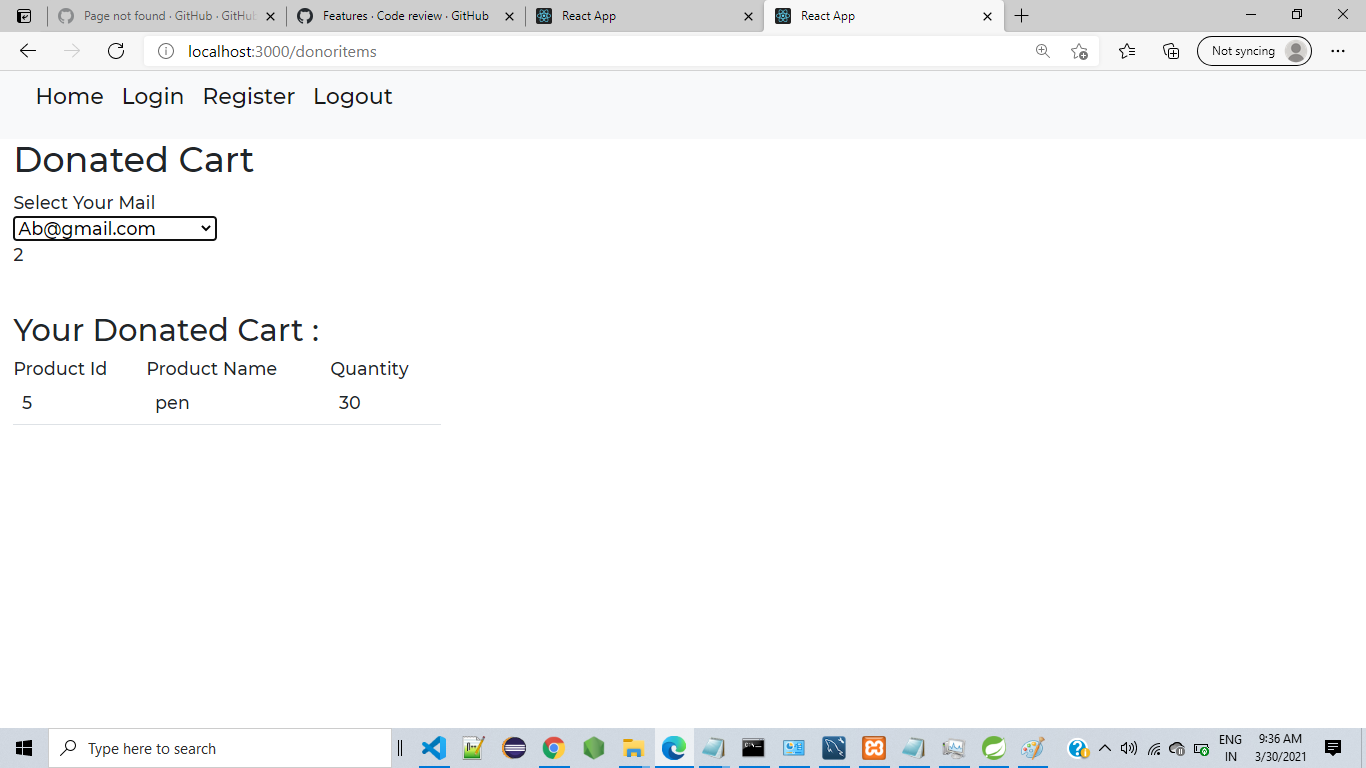
**7.6 Donated Product Form**

****

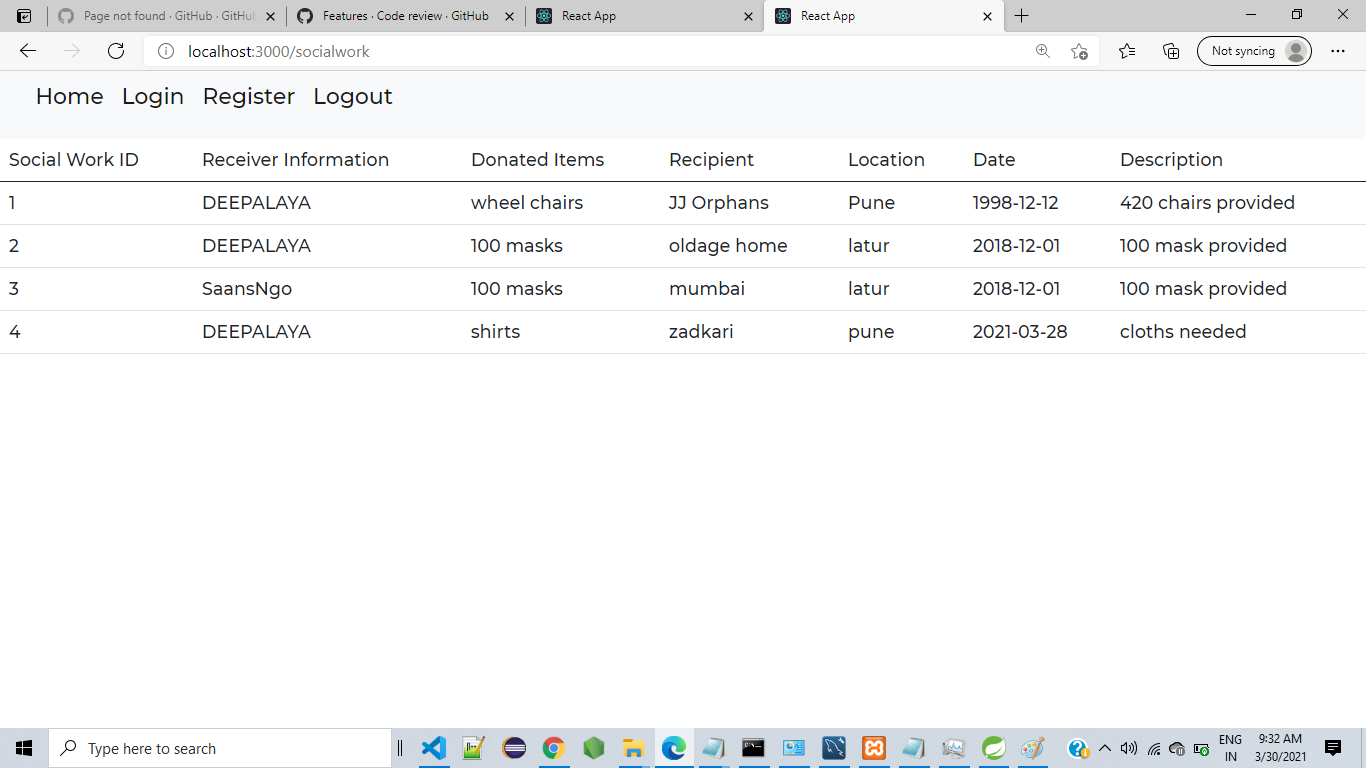
**7.7 Donated Item List**

****

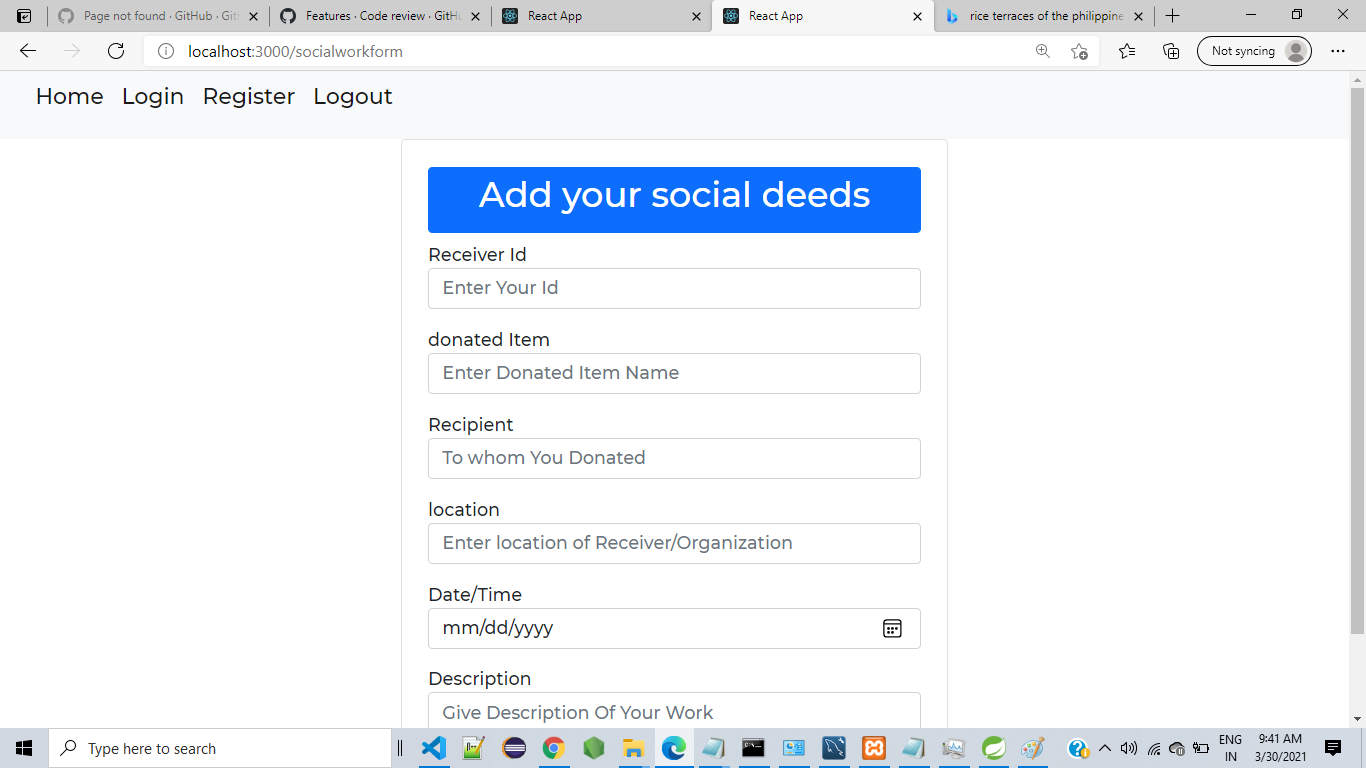
**7.8 Donated Cart Item**

****

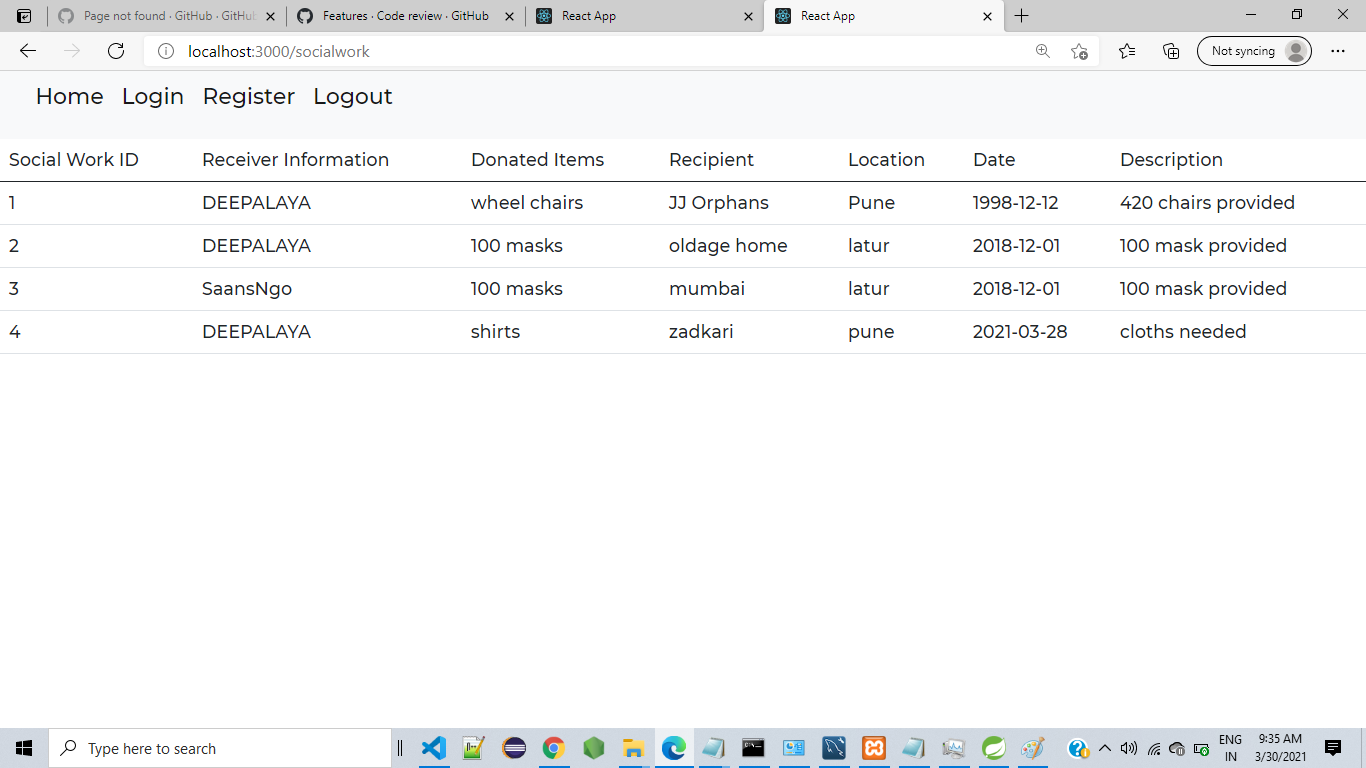
**7.9 Social Deeds**

****

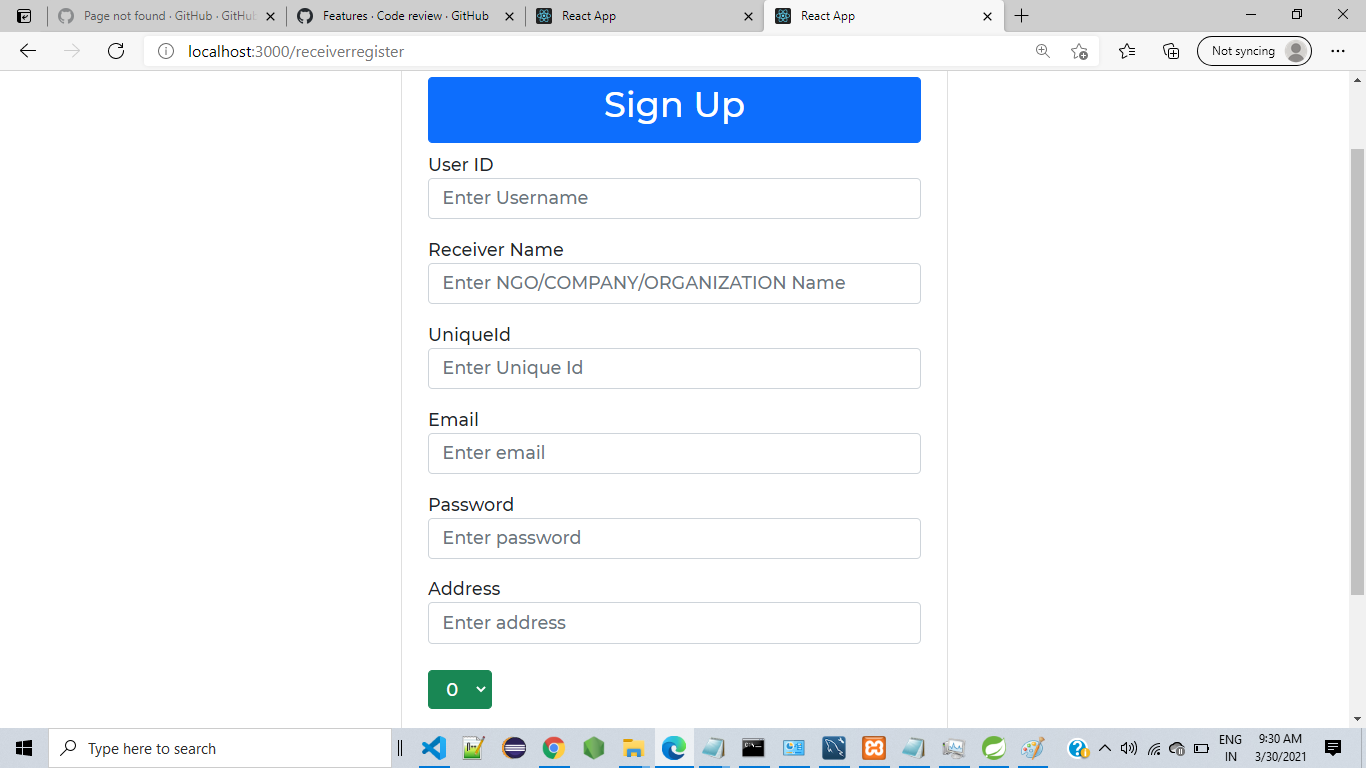
**7.10 Add Social Deeds**

****

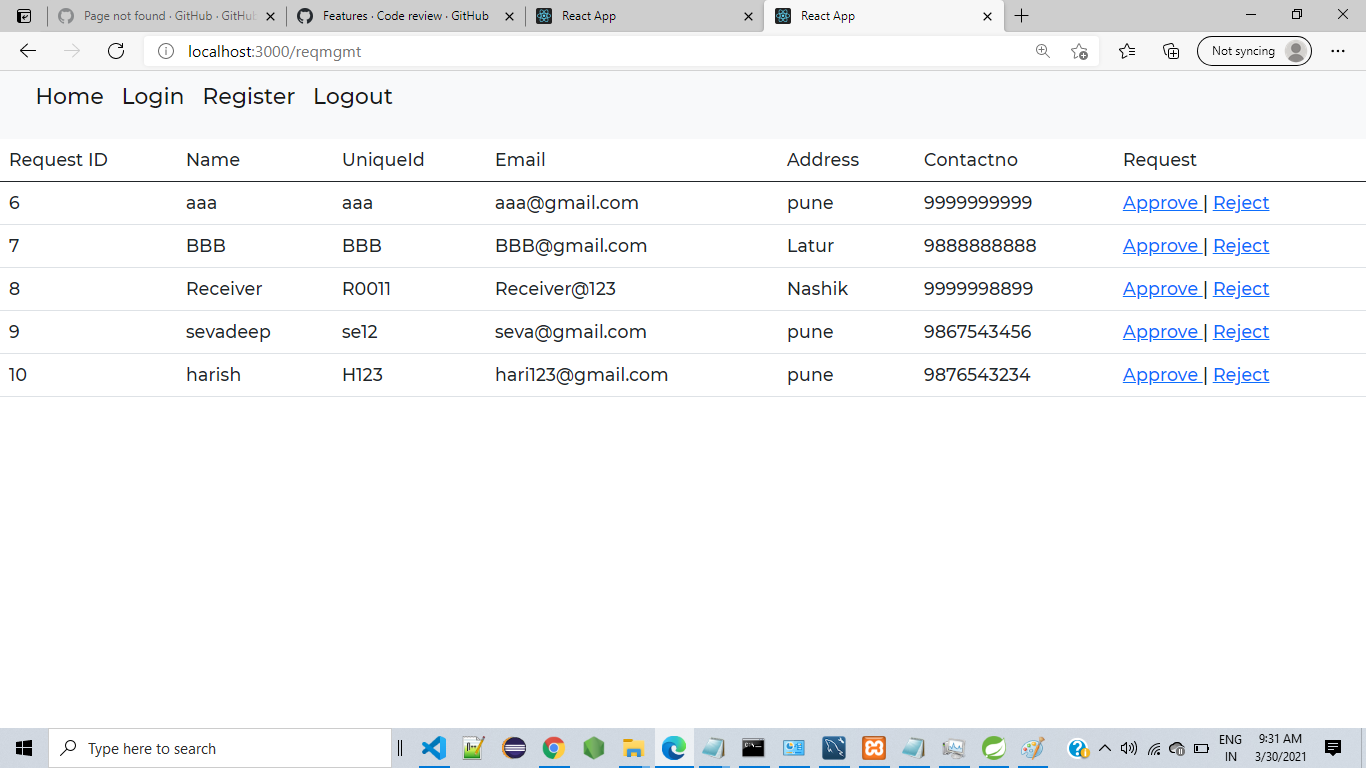
**7.11 Donor Side social Deeds List**

****

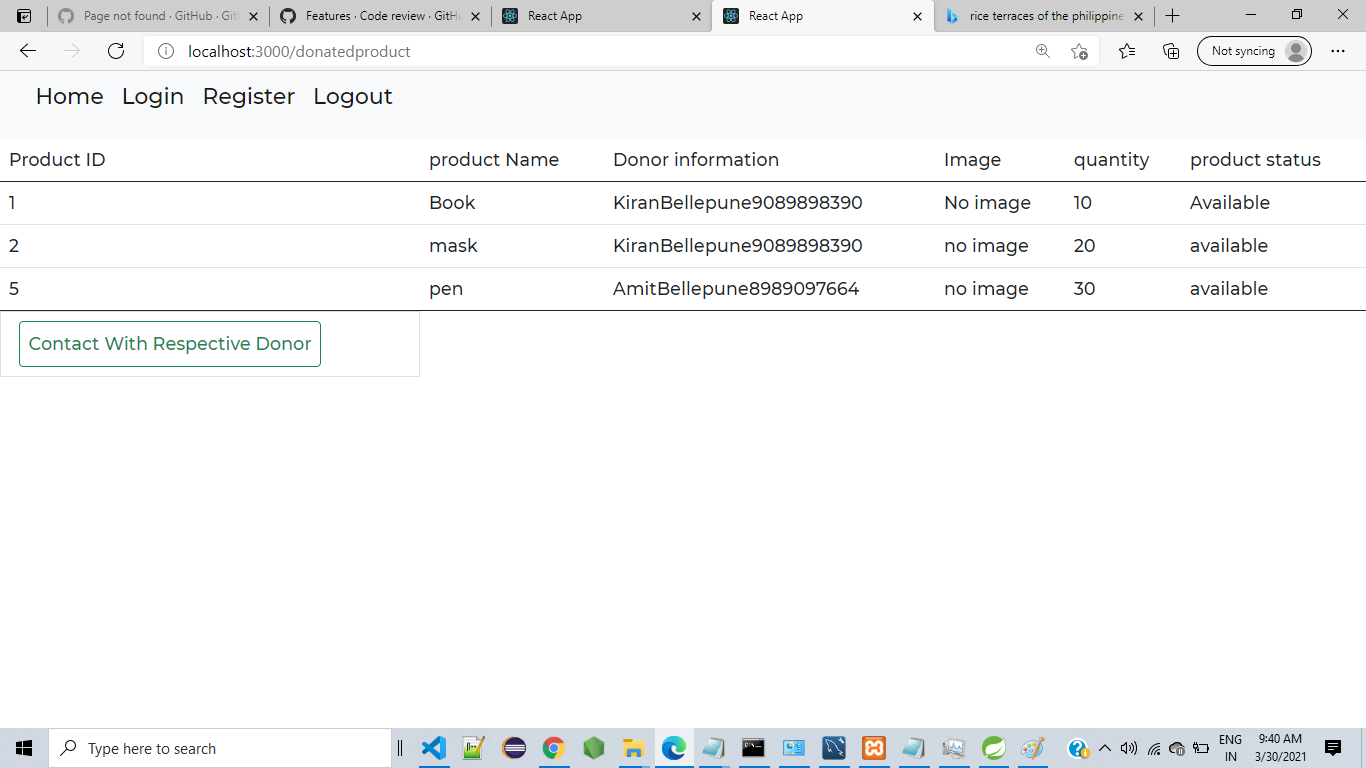
**7.12 Receiver Registration**

****

**7.13 Request Registration**

****

**7.14 Receiver Side Donor List**

****

**8.Conclusion**

Online donation platform puts forth the helping hands for social deeds. Donor can Donate services and functionalities from the society anywhere and anytime for their own comfort.

**Future Scope**

The software is flexible enough to be modified and implemented as per future requirements. We have tried our best to present this free and user–friendly website to Donate Things . Message and Email alerts for various happenings in the society can be added to the system so that Donor do not miss the updates

**9. References**

<https://stackoverflow.com>

<https://github.com>