A project report on

Local Charity

Submitted in partial fulfillment of the required for the award of

BCA

Degree of

KANNUR UNIVERSITY

By

RAHUL RAJESH

Register No:SE20BCAR16

LEAD KINDLY LIGHT

1981

S.E.S. COLLEGE

SES COLLEGE SREEKANDAPURAM
KANNUR-670631
KERALA

2023

KANNUR UNIVERSITY

SES COLLEGE

SREEKANDAPURAM,KANNUR-670631

CERTIFICATE

This is to certify that the project entitled “**LOCAL CHARITY**” submitted in partial fulfillment of the requirement for the award of BCA Degree is a result of bonafide work of carried out by **Mr.Rahul Rajesh** (**SE20BCAR16**) during the year 2023.

Internal Guide

Mrs.Krishnendhu

Assistant Professor

SES College, Sreekandapuram

Head of the institution

SES College, Sreekandapuram

External Examiners

1.(Name & Signature):

2.(Name & Signature):

DECLARATION

Rahul Rajesh VI semester BCA student of SES College, Sreekandapuram, under Kannur University do hereby declare that the project entitled "**LOCAL CHARITY**" is the original work carried out by me under the super vision of Mrs.krishnendhu ,Assistant Professor , SES College, Sreekandapuram towards partial fulfillment of the requirement of BCA Degree, and no part there of has been presented of any other degree.

Date:

RAHUL RAJESH
Reg No:SE20BCAR16
Department of BCA
SES College, Sreekandapuram
Kannur-670631

ACKNOWLEDGEMENT

I express my sincere gratitude to Mrs.Deepthi K.K , Head Of the Department of Computer Science and Application SES College Sreekandapuram, for his valuable advices and help throughout this project.

My sincere thanks to Mrs. Krishnendhu , Assistant Professor department of computer science and application , SES College Sreekandapuram, who was my Internal Guide, for his valuable support and helps during the project period.

I convey my thanks to all the staff members of Department of Computer Science and Application, SES College Sreekandapuram, and my friends who helped me during project.

ABSTRACT

Project title: **Local Charity**

The Local Charity is a web-based application that focuses on helping people in need by acting as a mediator between donors and users. This application is designed to support small locality-based charities and enables users to donate items such as books, clothes, and more. The system is divided into two modules: user and administrator. The user module allows users to explore the site, place orders, and create donations. The administrator module is composed of a group of people in a specific locality who are ready to do charity work, and they control the overall system. When a complaint is issued by the user, the administrator resolves it by discussing it with the donor. The process of transaction of items is easy and flexible. Users can directly contact the donor to collect the item, and the donor can also approach the user to handover the item. The donor can courier the item to the user or the person who ordered it. The administrator can also collect the item from the donor and hand it over to the user. The user can collect the item from the administrator. With this web application, donating to a good cause has never been easier.

There are two modules in this web app. They are

1.User module

2.Administrator module

1.User module: The user module in the Local Charity web-based application is designed to enable users to explore the site, place orders, and create donations. Users can register and create an account, which allows them to access the full functionality of the system.

Once logged in, users can browse the available items that are up for donation, add them to their cart, and place an order. Users can also create a donation listing for items that they would like to donate. This can include various items such as books, clothes, toys, or any other useful items that can be donated to people in need.

The user module also allows users to manage their account settings, view their donation history, and track their orders. Overall, the user module is designed to make it easy and convenient for users to donate to charity and help people in need.

2.Administrator module: The administrator module in the Local Charity web-based application is designed to provide a group of people in a specific locality with the ability to manage the overall system. These administrators are responsible for controlling the donation process and ensuring that all transactions go smoothly.

The administrator module allows administrators to manage user accounts and donations. They can view and manage donation listings, approve or reject donations, and handle any complaints that may arise.

One of the primary functions of the administrator module is to ensure that all donations are legitimate and that they reach the intended recipients. This involves verifying the authenticity of donors and ensuring that the donated items are in good condition before passing them on to the users.

Overall, the administrator module is designed to provide a reliable and secure platform for charitable donations. It enables administrators to effectively manage the donation process, ensuring that all parties involved are satisfied and that the charity's goals are achieved.

The process of transaction of items in the Local Charity web-based application involves several options for users and donors to exchange donated items. Here are some of the points related to the process of transaction:

1. Direct Contact: Users can directly contact donors and arrange to collect the donated items from them.
2. Direct Handover: Donors can directly approach users and hand over the items to them.
3. Courier: Donors can also choose to courier the donated items to the user or the person who placed the order.
4. Admin Collection: The administrator can collect the donated items from the donor and hand them over to the user.

5. Admin Handover: Users can also collect the donated items from the administrator.

In all of these options, the administrator acts as a mediator, ensuring that the transaction is completed successfully and that both the donor and the user are satisfied with the exchange. The process provides flexibility and convenience for both parties, allowing them to choose the option that works best for them.

CONTENTS

CHAPTER-1

INTRODUCTION

1.INDRODUCTION

The Local Charity is a web-based application. Which focus on helping people in need. The system act as a mediator between donator and user. The application focus a small locality based charity . This application enable as to donate items such as books, cloths etc . In scenario where we have items no longer need as and if we are willing to donate the items, the web application works as a mediator.

The web-application includes two modules. Such as User module and Administrator module.

CHAPTER-2

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

Requirement analysis allows us to elaborate on basic requirements established during the inception, elicitation , and negotiation tasks are the part of requirement engineering . Requirements are analyzed in order to identify inconsistencies, defects, omissions etc.

Analysis report contains a detailed study of existing system like how is working , what are the drawbacks of the system etc. From that study, we can understand the need of the new system. The analysis of requirements start with requirement of elicitation. This is also known as gathering of requirements. Here the requirements are identified mainly with the help of customer. So, elicitation can succeed only through an effective customer-developer partnership.

For requirement analysis , we choose the method Interview. The first step is to arrange an interview or meeting with the customer. The main objective of conducting an interview is to understands the customer’s expectations from software. It will be impossible to interview every stakeholders. So there are several groups to be considered for conducting interview and representatives

must be selected based on their technical expertise, domain knowledge, credibility and accessibility. The interview may be started with simple questions to set people at ease. The questions will help to start the communication that is essential for understanding the requirements. The reason behind to choose this method is ,we want to collect more information from a customer , worker, and a constructor about to know their needs . So a face to face talking with these people regards to understand what they want actually. If there any requirements which we can’t implement, there is also a facility to express our limitations in implementation at that moment.

2.1 EXISTING SYSTEM

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

There are few system and application that already available for donors to donate. But not all the available system and application are user friendly. So, Local Charity Application comes out with the solution which is user friendly. There are some adding features for this application in order to ease the donors to make a donation.

Existing System(**Donate**InKind)

DonateInKind is partially similar to Local Charity web application. In DonateInKind web application you may donate old clothes, old books ,regular meals, used text books, blood, hair, mother's milk etc to charities directly or you may hand-over the donation to an intermediary organization.

Collection methods vary widely and are dependent on every organization's resources. Some organizations have arrangements in place for clothing donation pickup from homes and offices. Some have set up drop-off points in cities where the used clothing can be left. It is also true that many institutions lack the means to collect used clothing and request that the old clothes may be delivered at the institution's location. In such cases, donors located in the same city as the organization can do the needful.

2.2 PROPOSED SYSTEM

The aim of this application is donators can easily donate their

products like old books ,old cloths or other items . In the case of clothes we have clothes that do not fit us anymore. As children grow up, they outgrow their clothes

every year. So instead of storing the old clothes or throwing them away, donate used clothes because what is waste for you could be a resource for the less fortunate.

Using this app we can easily create our donations. Simply take one or more photo of the product , then open this Local Charity application , then press the create donation button , after that fill the form that includes donation’s details and donator’s details and press the create button. This action may cause to create your donation on this application.

The all donations are visible to all persons entering this app(donators and donors). And the donors can easily order a the items in stock. The order details are visible to administrator and the donator.

Local Charity application is act as a media to connect directly the donators and donors. The donators can easily donate their donations and donors able to order it simply. After that donor can directly communicate with donator or administrator. This is the main advantage of this application. A donator can donate their product to actual persons in need using this application.

Other web application like **Donate**InKind , we can donate our donations but there is no assure to get the donation to actual donators.

Charity is an act of kindness, where a person who has financially more than enough of what he or she needs contributes a part of his or her surplus income for the fulfilment of the needs of those who are less capable. This charity management system will help donators to find donors easily. This web application is developed completely for the donors to donate anything to the people in need. This platform is totally avoiding from any scammers who intended to take the donation's item for their own sake. The system will record all the donor's info. The user can choose which category of their donation.

2.3 FEASIBILITY STUDY

Feasibility is defined as the practical extend to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study.

The following are the feasibility study:

-Technical Feasibility

-Economic Feasibility

-Behavioral Feasibility

-Legal Feasibility

2.3.1 Technical Feasibility

Technical feasibility study deals with the hardware well as software requirements. Here we are used MERN stack for creation of this web application. MERN Stack is a collection of powerful technologies and robust, used to develop scalable master web applications comprising **backend, front-end,** and **database components**. It is JavaScript that is used for the faster and easier development of full-stack web applications. MERN Stack is a technology that is a user-friendly full-stack JavaScript framework for building applications and dynamic websites.

MERN Stack consists of four main components or can say four main technologies:

1. **M** stands for **MongoDB (Database),** mainly used for preparing document database and is a NoSQL (Non-Structured Query Language) Database System.
2. **E** stands for **Express,** mainly used for developing Node.js web framework.
3. **R** stands for **React,** mainly used for developing a client-side JavaScript framework.(**Frontend**)
4. **N** stands for Node.**js,** mainly used for developing the premier JavaScript web server.(**Backend**)

Each of these four technologies plays an important role in providing an end-to-end framework . Even these four technologies play an important role in the development process of web applications.

2.3.2 Economic Feasibility Study

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as Cost/Benefits analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. The Local Charity does not require enormous amount of money to be developed.

2.3.3 Behavioral Feasibility Study

Local Charity web application is designed in user friendly manner and no need to provide training for the people for using. The operating system which we used is windows 11, which is also user friendly.

2.3.4 Legal Feasibility Study

Legal feasibility is to determine whether the proposed system conflicts with the legal requirements. Local Charity will not make violation in the rules, so it is legally feasible.

2.4 System Requirements

2.4.1 Hardware Requirements

System : Intel core i3 or above

Hard Disk : 40GB or above free space

Memory : 4GB RAM or above

2.4.2 Software Requirements

MERN stack is used for creation of the web application.

Frontend : React.js

Backend : Node.js

Database : MongoDB(No SQL)

IDE : Visual Studio Code

Browser Requirements : Google chrome / Mozilla Firefox

2.5 SYSTEM REQUIREMENT SPECIFICATION

System requirement are expressed in a software requirement document. The software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirement definition and the requirement specification. The software requirement document is not a design document.

The software specification document satisfies the following

- It satisfies the external system behaviors.
- It specifies constraints on the implementation.
- It is easy to change.
- It serves as a reference tool for system maintainers.

2.5.1 Actor Identification

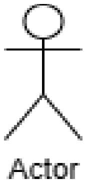
An actor is someone or something that interacts with the system. An actor is he/she who uses the system. An actor exchanges information with the system. Asking certain questions as detailed below can identify the actors of the system.

1	Who are the main users of the System?	Users, Administrator
2	Who is the administration of the system?	Administrator
3	What else devices need to interact with the system?	Database
4	Who are the benefiters of result produced by the system?	Users

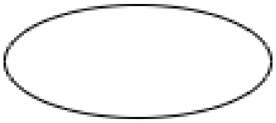
The actors of the system are:

→ Users
→ Administrator

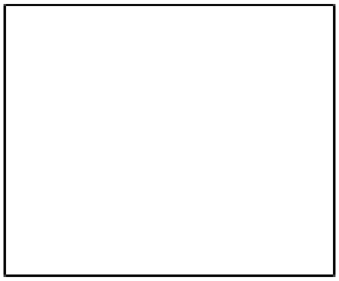
Use Case Notations



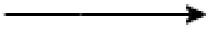
Actor



Use case

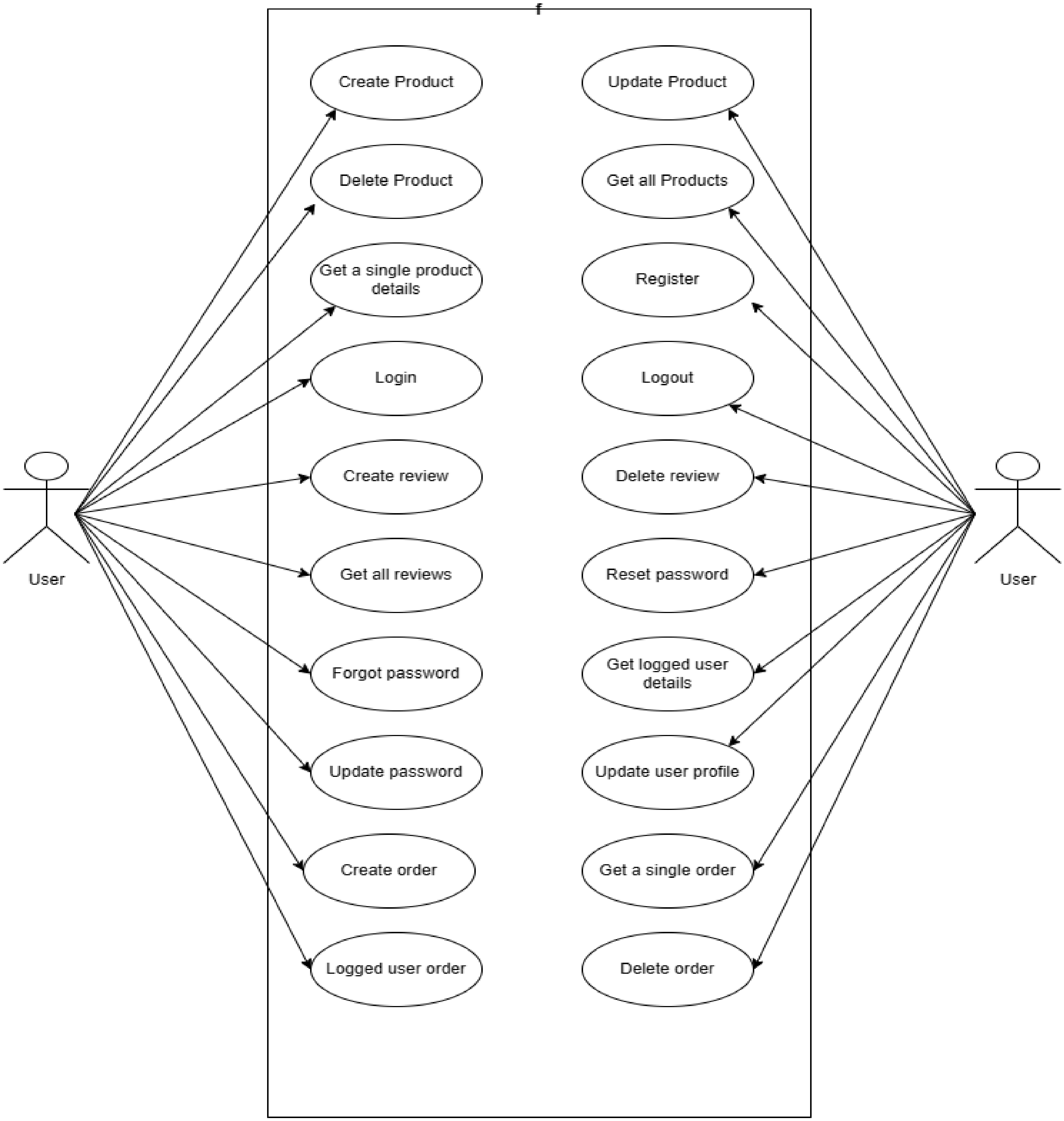


System boundary

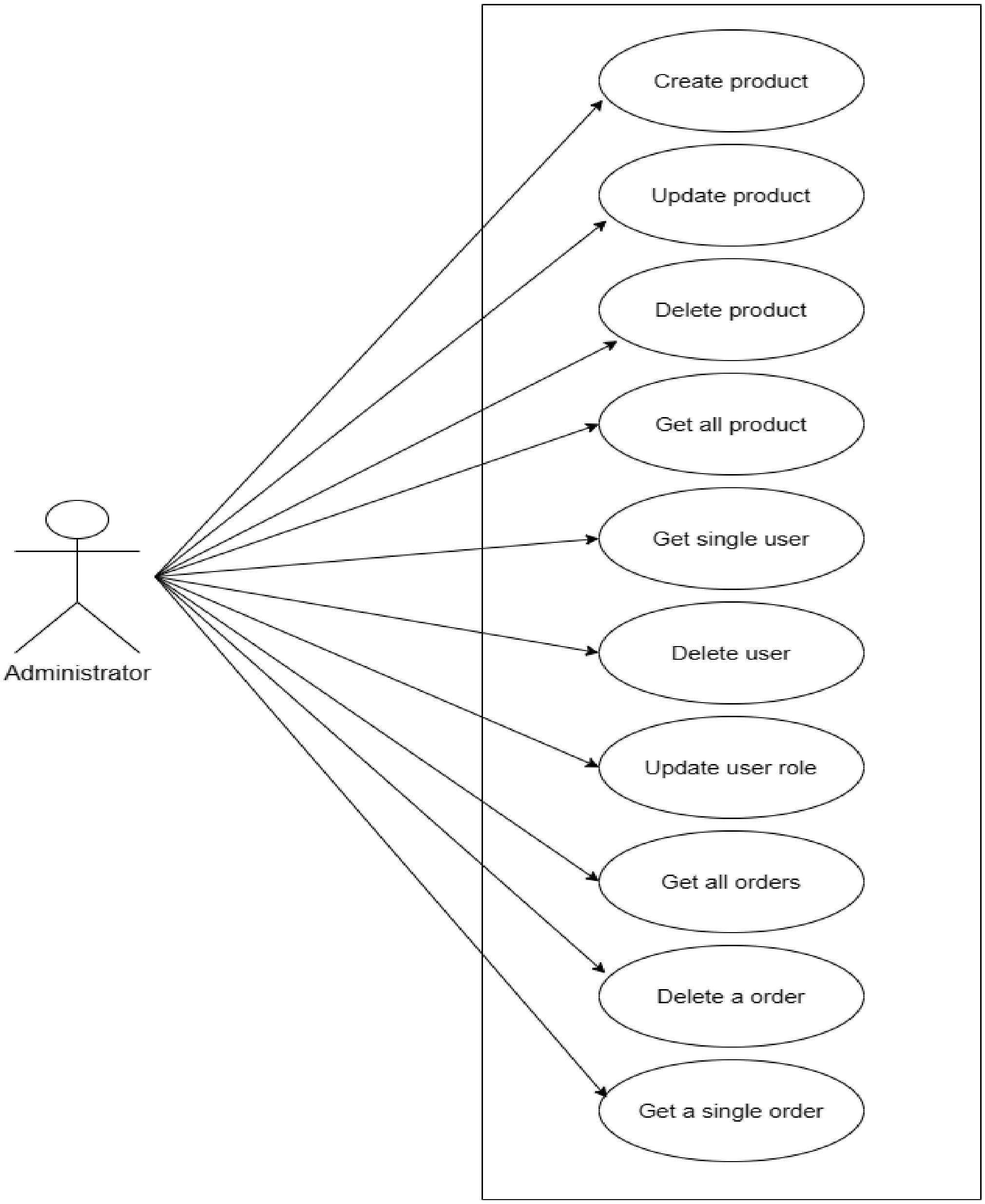


Association relationship

Use case diagram –User



Use case diagram –Admin



CHAPTER-3

SYSTEM DESIGN

3.SYSTEM DESIGN

System design is the process of defining the elements of a system such as the architecture , modules an components, the different interface of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

System design implies a systematic approach to the system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variable of the system that needs to be created from the architecture , to the required hardware and software , right down to the data and how it travels and transforms throughout its travel through the system. Systems design then overlaps with systems analysis, systems engineering and systems architecture.

Major activities during the design phase are:

- Data Base Design
- Architectural Design
- Interface Design
- Modular Design

3.1 Database Design

Designing a database requires an understanding of the business functions you want to model. It also requires an understanding of the database concepts and features that you want to use to represent those business functions. Make sure that you accurately design the database to model the business, because it can be time consuming to significantly change the design of a database after you implement it. A well designed database also performs better.In this project contains NoSQL Database. NoSQL databases(“not only SQL”) are non –tabular databases and store

data differently than relational tables. NoSQL databases come in a variety of types bases on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.

Types of NoSQL Databases

- Document databases store data in documents similar to JSON(JavaScript Object Notation) objects. Each document contains pairs of fields and values. The values can typically be a variety of types including things like strings, numbers, booleans , arrays ,or objects.
- Key-value databases are a simpler type of database where each item contains keys and values.
- Wide-column stores data in tables, tows , and dynamic columns.
- Graphic databases store data in nodes and edges. Nodes typically store information about people , places , and things, while edges information about relationships between the nodes.

Here MongoDB databases program is used to create databases. MongoDB is the most popular NoSQL database, is an open source document-oriented database. The term ‘NoSQL’ mean ‘non-relational’. It means that MongoDB isn’t based on the table mechanism for storage and retrieval of data. This format of storage is called BSON(similar to JSON format).

SQL databases store data in tabular format . This data is stored in predefined data model which is not very much flexible for today’s real-world highly growing applications. Modern applications are more networked, social interactive than ever. Applications are storing more and more data and are accessing it at higher rates.

Relational Database Management System(RDBMS) is not the correct choice when it comes to handling big data by the virtue of their design since they are not

horizontally scalable. If the databases runs on a single server , then it will reach a scaling limit. NoSQL databases are more scalable and provide superior

performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

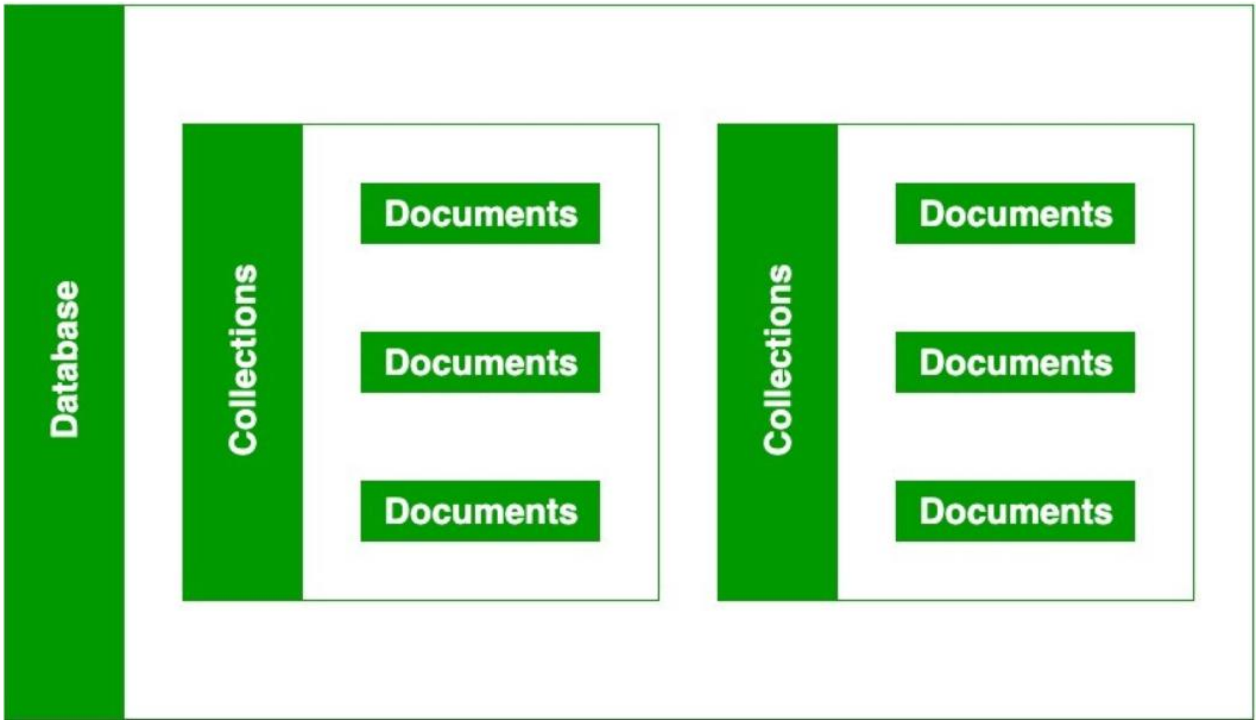
Features of MongoDB:

- **Document Oriented**: MongoDB stores the main subject in the minimal number of document and not by breaking up it up into multiple relational structures like RDBMS. For example, it stores all the information of a computer in a single document called Computer and not in distinct relational structures like CPU, RAM, Hard disk, etc.
- **Indexing**: Without indexing, a database would have to scan every document of a collection to select those that match the query which would be inefficient. So, for efficient searching Indexing is a must and MongoDB uses it to process huge volumes of data in very less time.
- **Scalability**: MongoDB scales horizontally using partitioning data across various servers. Data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. Also, new machines can be added to a running database.
- **Replication and High Availability**: MongoDB increases the data availability with multiple copies of data on different servers. By providing redundancy, it protects the database from hardware failures. If one server goes down, the data can be retrieved easily from other active servers which also had the data stored on them.

- **Aggregation**: Aggregation operations process data records and return the computed results. It is similar to the GROUPBY clause in SQL. A few aggregation expressions are sum, avg, min, max etc.

MongoDB –Database, Collection, and Document

Databases, collections, documents are important parts of MongoDB without them you are not able to store data on the MongoDB server. A Database contains a collection, and a collection contains documents and the documents contain data, they are related to each other.



- **Collection**- Collections are just like tables in relational databases, they also store data, but in the form of documents. A single database is allowed to store multiple collections.

- **Documents**- the data records are stored as BSON documents. Here, BSON stands for binary representation of JSON documents, although BSON contains more data types as compared to JSON. The document is created using field-value pairs or key-value pairs and the value of the field can be of any BSON type.

Syntax:

```
{
                              field1: value1
                              field2: value2
                              ....
                              fieldN: valueN
    }
```

3.1.1 ER Diagram

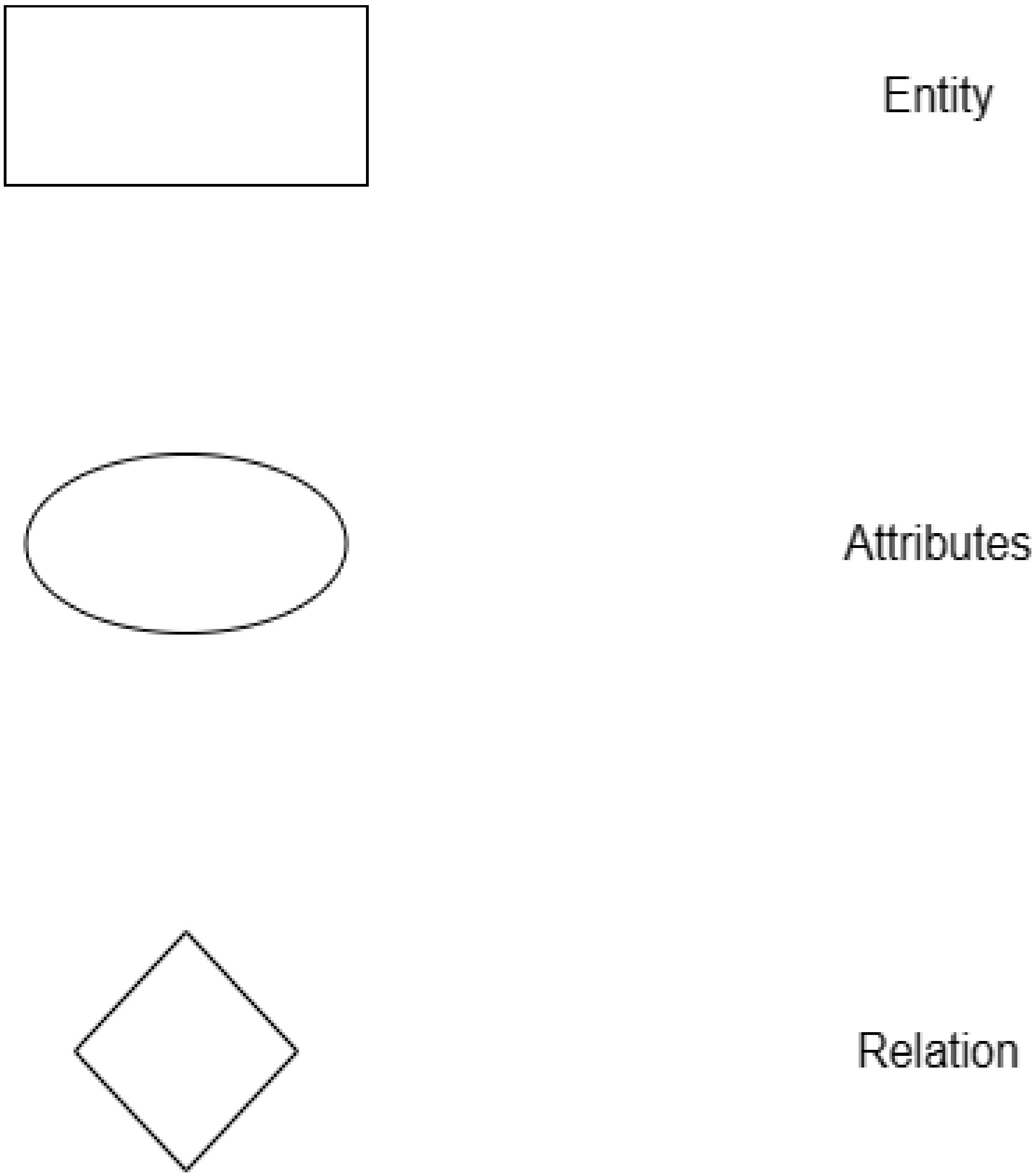
ER-modeling is a data modeling technique used in software engineering to produce a conceptual data model of an information system. Diagrams crated using this ER modeling technique are called Entity-Relationship Diagrams, or ER diagrams. So you can say that Entity Relationship Diagrams illustrate the logical structure of databases.

There are three basic elements in ER-Diagrams:

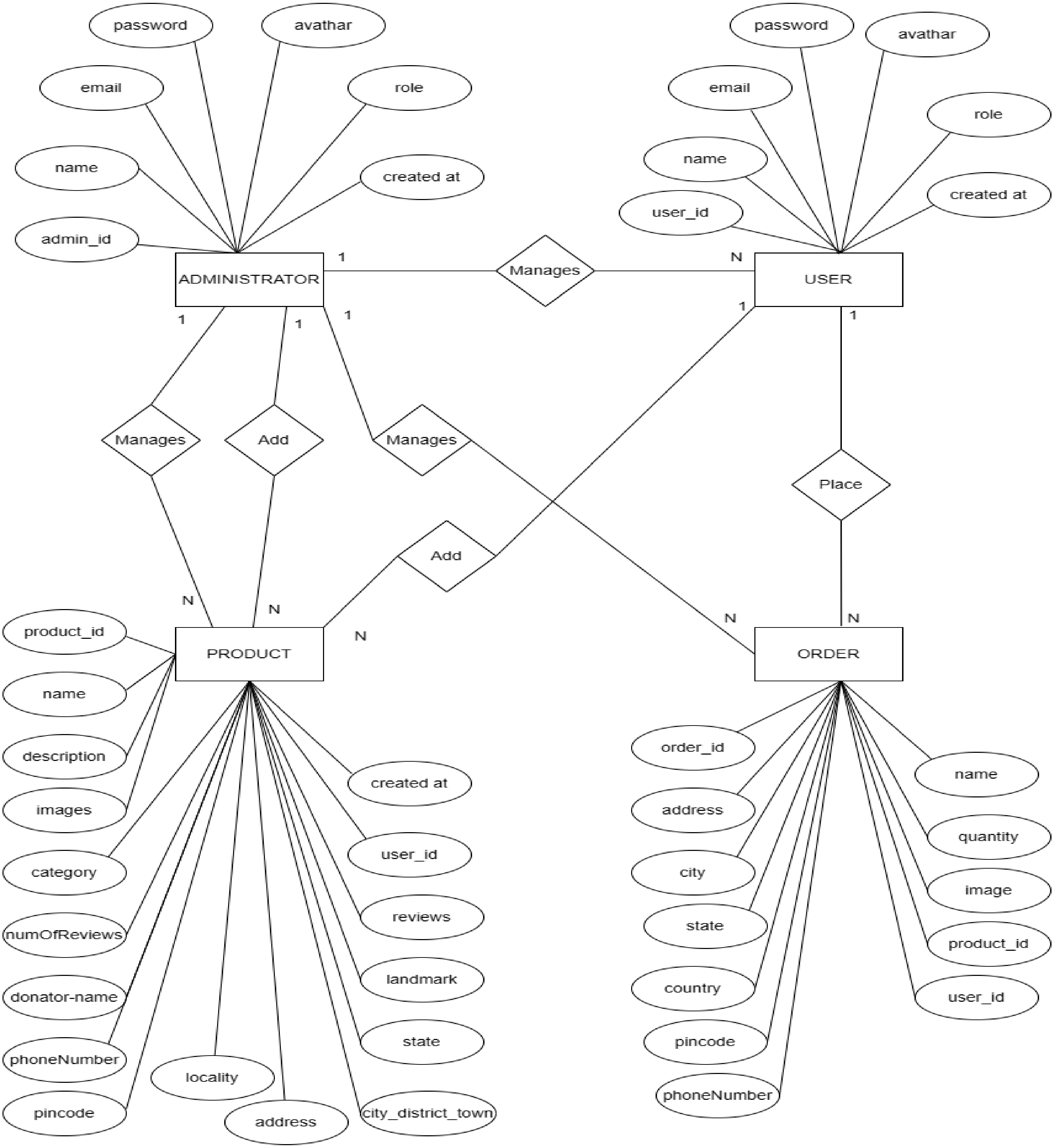
- Entities are the "things" for which we want to store information. An entity is a person, place, thing or event.
- Attributes are the data we want to collect for an entity.
- Relationship describes the relations between the entities.

ER diagrams show entities in a database and relationships between tables within that database. It is essential to have ER-Diagrams if you want to create a good database design. The diagrams help focus on how the database actually works.

ER Diagram Symbols:


Entity
Attributes
Relation

The ER diagram available for the Local Charity are as follows:



3.1.2 Design of Documents Schema

Guidelines:

- Storing together what needs to be accessed together.
- Modeling one-to-one relationships with embedded documents.
- Modeling one-to-few relationships with embedded documents.
- Modeling one-to-many and many-to-many relationships with child references.
- Modeling unbounded one-to-many relationships with parent references.

User Document Schema

```
const userSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
    maxLength: 30,
    minLength: 4,
  },
  email: {
    type: String,
    required: true,
    unique: true,
    validate: validator.isEmail,
  },
  password: {
    type: String,
    required: true,



 minLength: 8,
    select: false,
  },
  avatar: {
    public_id: {
      type: String,
      required: true,
    },
    url: {
      type: String,
      required: true,
    },
  },
  role: {
    type: String,
    default: "user",
  },
  createdAt: {
    type: Date,
    default: Date.now,
  }
});
```

Product Document Schema

```
const productSchema = mongoose.Schema({
  name: {
    type: String,
    required: true,



 trim: true,
  },
  description: {
    type: String,
    required: true,
  },
  ratings: {
    type: Number,
    default: 0,
  },
  images: [
    {
      public_id: {
        type: String,
        required: true,
      },
 url: {
        type: String,
        required: true,
      },
    },
  ],
category: {
    type: String,
    required: true,
  },
  Stock: {
    type: Number,
    required: true,
    maxLength: 4,



 default: 1,
  },
  numOfReviews: {
    type: Number,
    default: 0,
  },

  // Donator details
yourname: {
    type: String,
    required: true,
  },

phoneNumber: {
    type: Number,
    required: true,
  },
 pincode: {
    type: Number,
   required: true,
 },
 locality: {
   type: String,
   required: true,
 },
 address: {
   type: String,
   required: true,
 },



city_district_town: {
   type: String,
   required: true,
 },



state: {
   type: String,
   required: true,
 },
 landmark: {
   type: String,
   required: true,
 },
//for reviews
  reviews: [
    {
      user: {
        type: mongoose.Schema.ObjectId,
        ref: "User",
        required: true,
      },
      name: {
        type: String,
        required: true,
      },



 comment: {
        type: String,
        required: true,
      },
    },
  ],

  user: {
    type: mongoose.Schema.ObjectId,
    ref: "User",
    required: true,
    },

createdAt: {
    type: Date,
    default: Date.now,
  },
});
```

Order Document Schema

```
const orderSchema = new mongoose.Schema({
  shippingInfo: {
    address: {
      type: String,
      required: true,
    },
    city: {
      type: String,
      required: true,
},


state: {
      type: String,
      required: true,
    },
country: {
      type: String,
      required: true,
    },
 pinCode: {
      type: Number,
      required: true,
    },
phoneNo: {
      type: Number,
      required: true,
    },
  },
orderItems: [
    {
name: {
        type: String,
        required: true,
 },
      quantity: {
        type: Number,
        required: true,
      },
      image: {
        type: String,
required: true,
      },
 product: {
        type: mongoose.Schema.ObjectId,
        ref: "Product",
        required: true,
      },
    },
  ],

  user: {
    type: mongoose.Schema.ObjectId,
    ref: "User",
    required: true,
  }
});
```

3.2 Architectural Design

The software architectural design represents the structure of the data and program components that are required to build a computer based system. The architecture highlights early design decisions that will have a profound impact on all software engineering work that follows and , as important , on the ultimate success of the system as an operational entity. So the architecture constitute a relatively small, components work together.

3.2.1 Data Flow Diagram

Data flow diagram (DFD) represents the flows of data between different processes in a business. It is a graphical technique that depicts information flow and the transforms that are applied as data move from input to output. It provides a simple, intuitive method for describing business processes without focusing on the details of computer systems. DFDs are attractive technique because they provide what users do rather than what computers do.

The DFD may be used for any level of data abstraction. DFD can be partitioned into levels. Each level has more information flow and data functional details than the previous level.

Highest level is Context Diagram. Some important points are:

- 1 bubble (process) represents the entire system.
- Data arrows show input and output.
- Data Stores NOT shown. They are within the system.

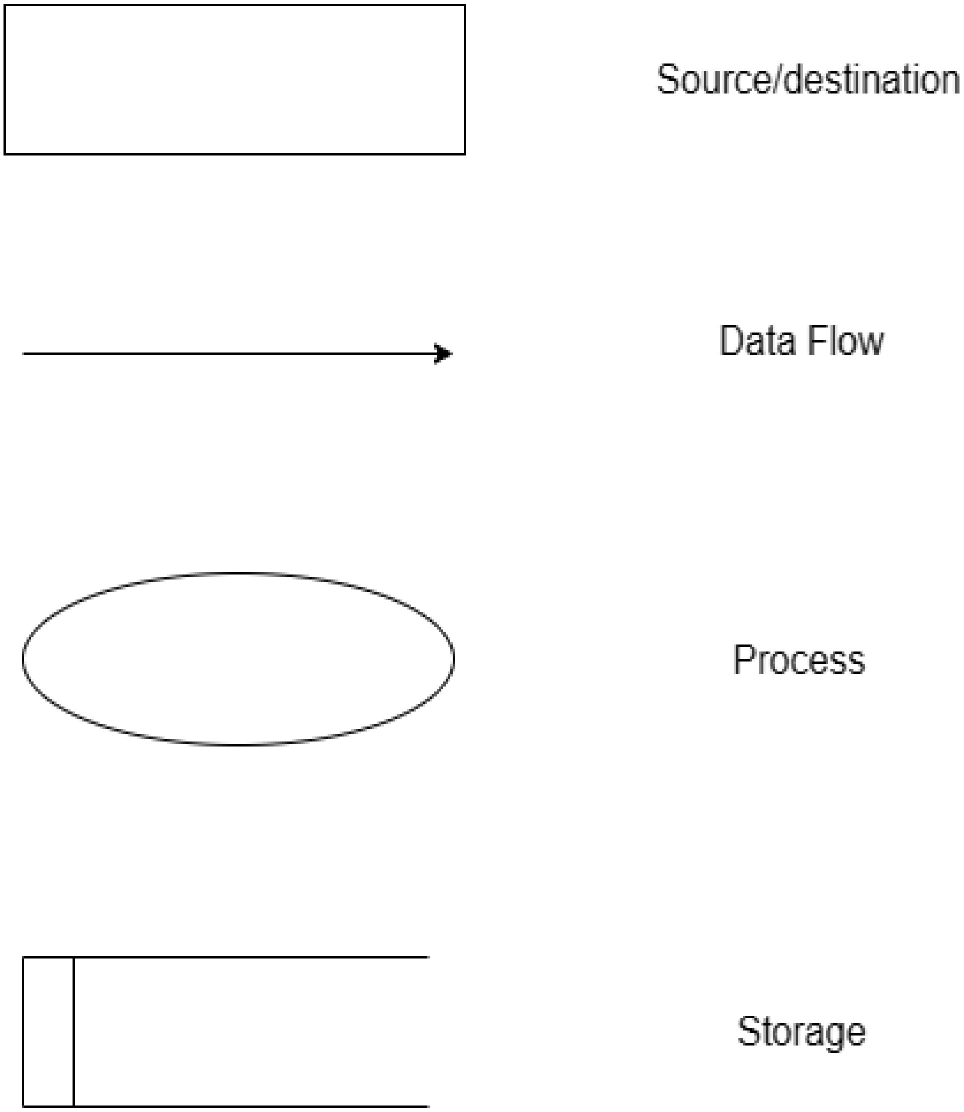
Next Level is Level 0 DFD. Some important points are:

- Level 0 DFD must balance with the context diagram it describes.
- Input going into a process are different from outputs leaving the process.
- Data stores are first shown at this level.

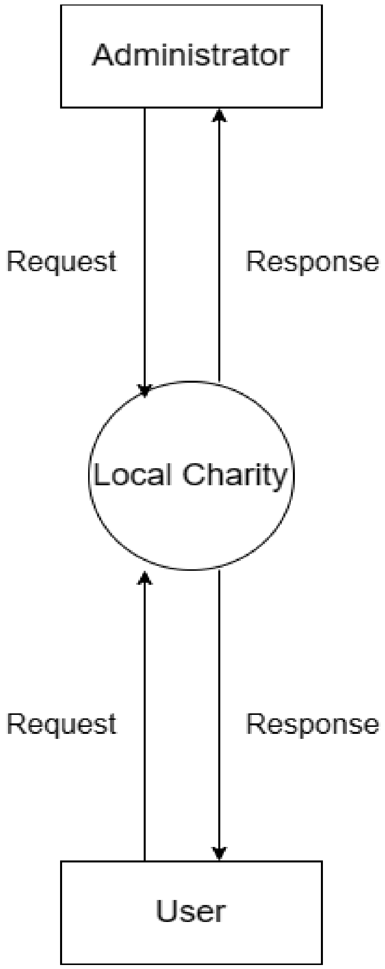
Next level is Level 1 DFD. Some important points are:

- Level 1 DFD must balance with the Level 0 it describes.
- Input going into a process are different from outputs leaving the process.
- Continue to show data stores.

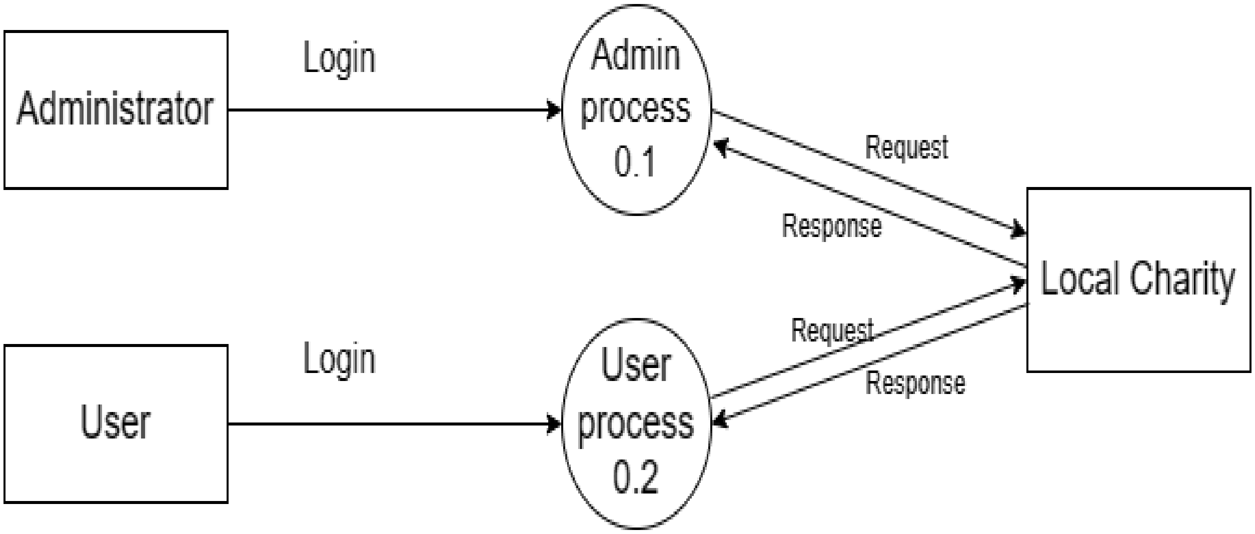
Data Flow Diagram Symbols:



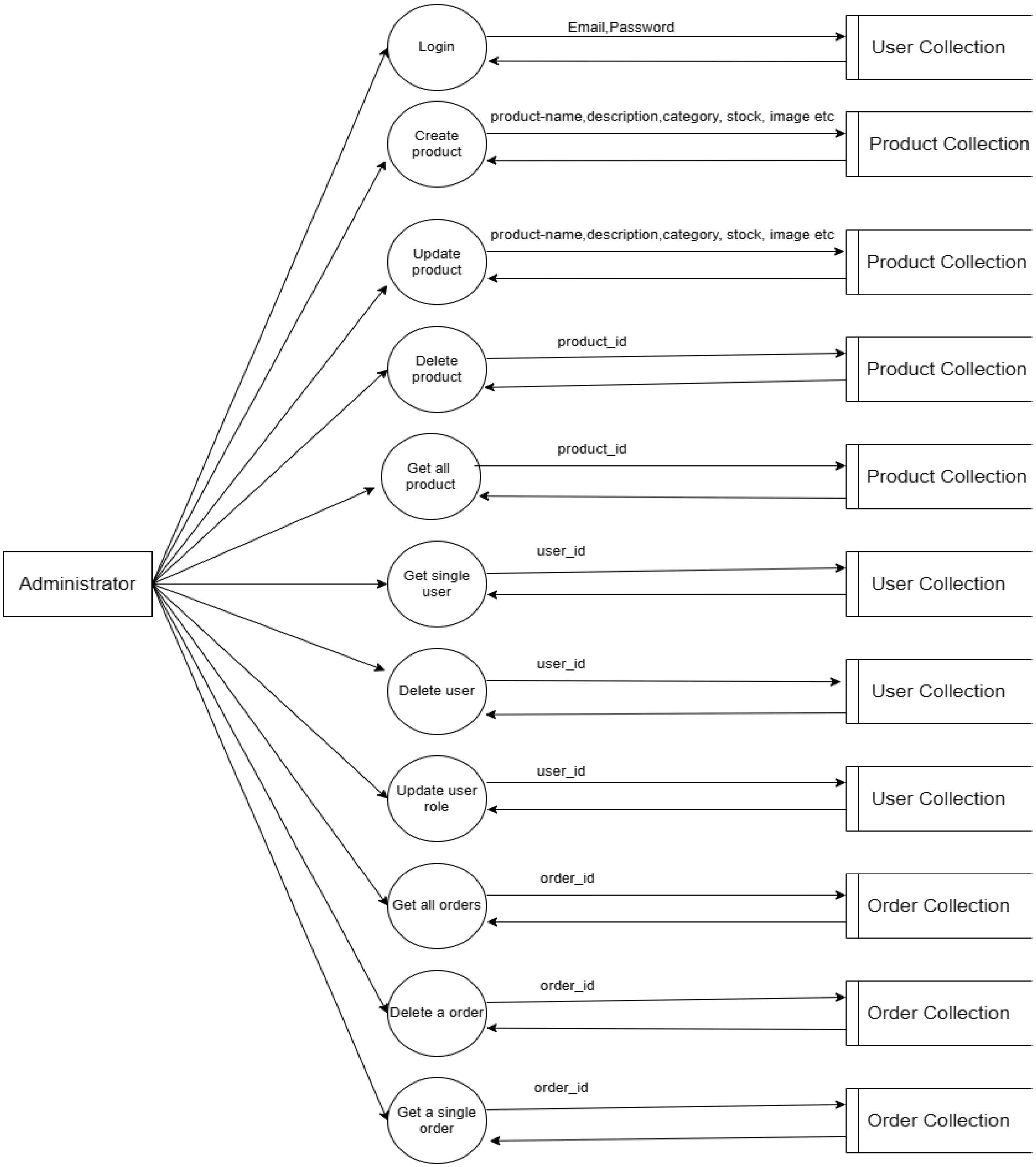
Level 0


Administrator
Request
Response
Local Charity
Request
Response
User

Level 1:

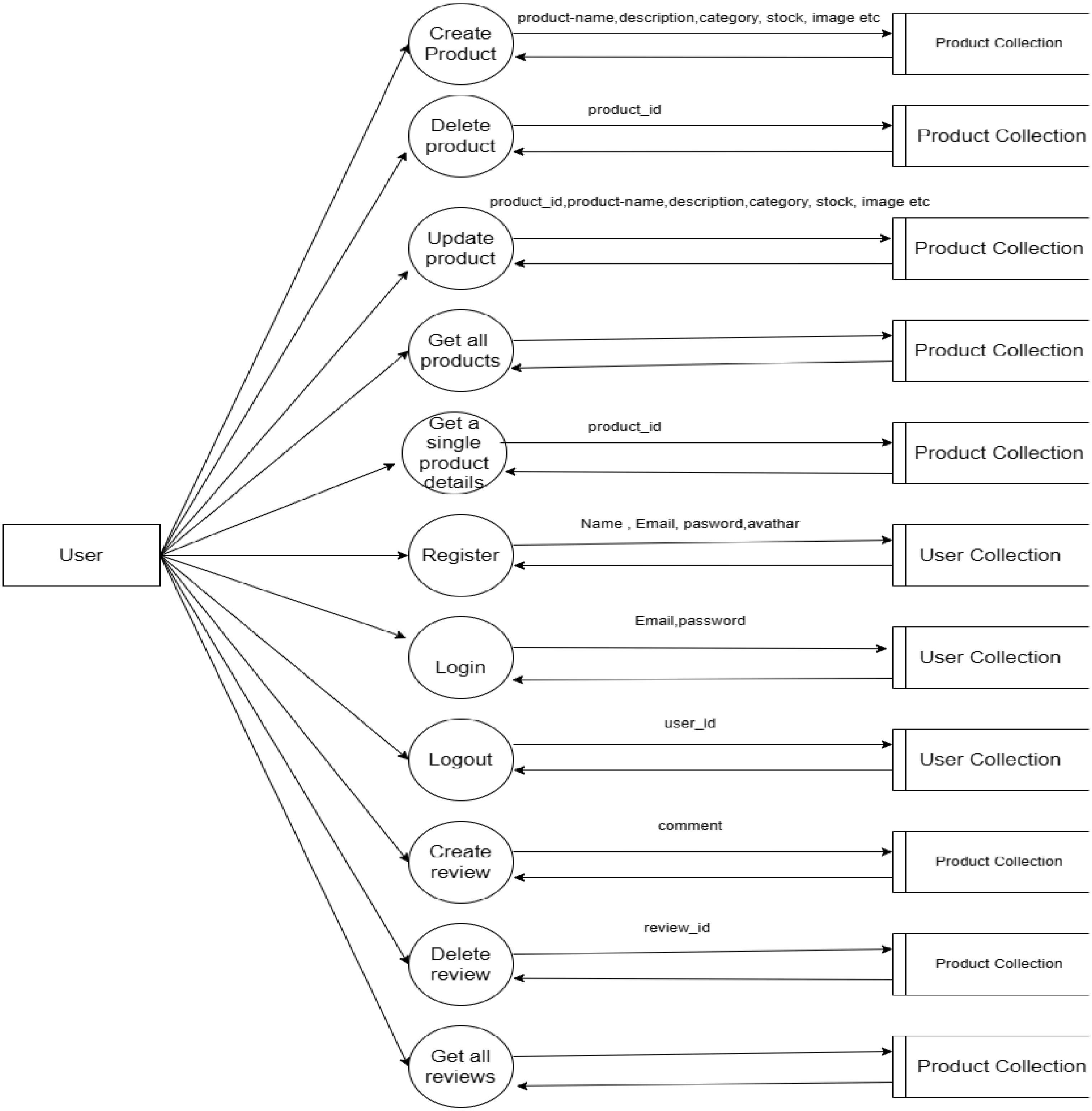

Administrator
Login
Admin process 0.1
Request
Response
Local Charity
Request
Response
User
Login
User process 0.2

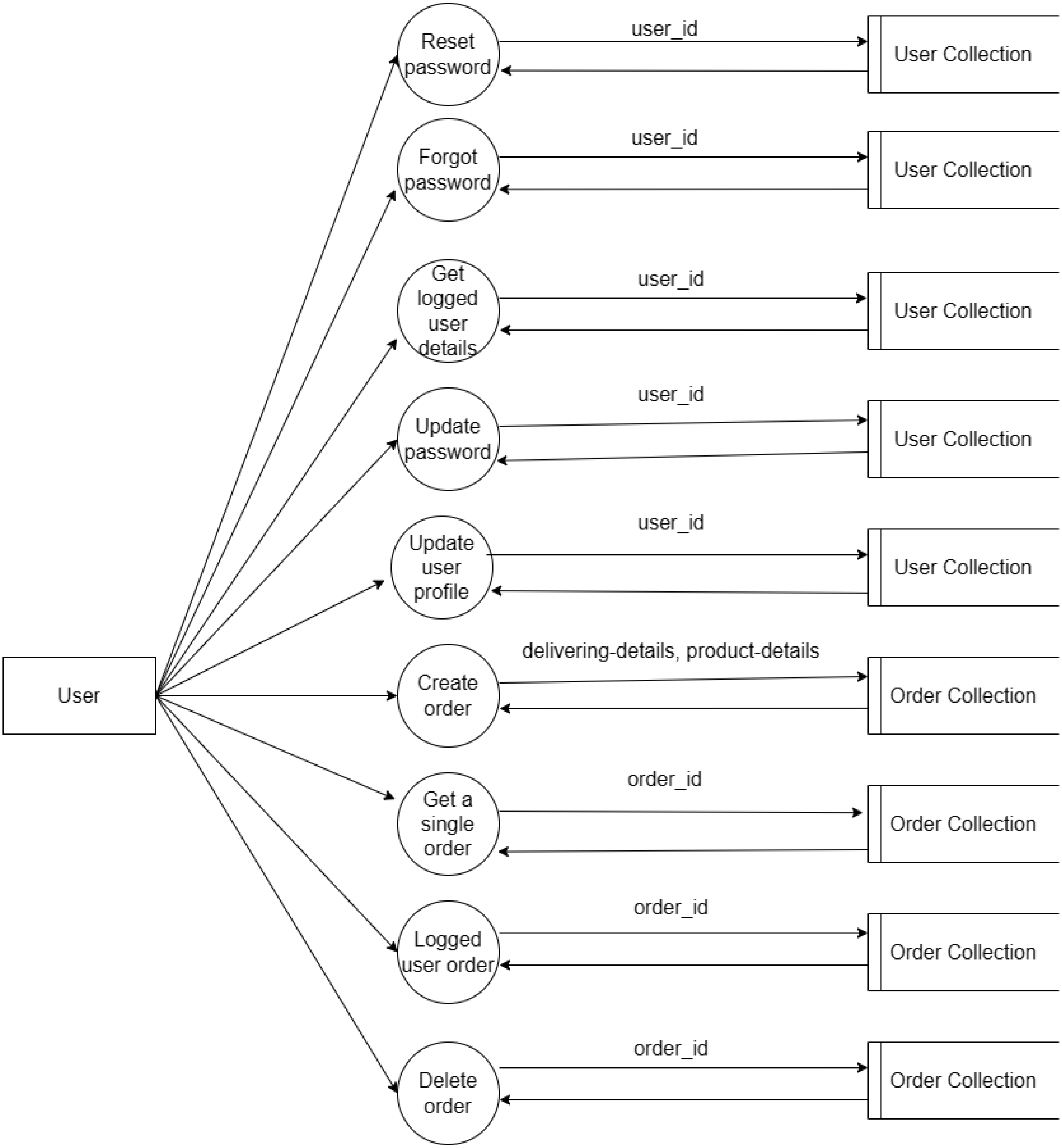
Level 2



Administrator
Login
Email,Password
User Collection
Create product
product-name,description,category, stock, image etc
Product Collection
Update product
product-name,description,category, stock, image etc
Product Collection
Delete product
product_id
Product Collection
Get all product
product_id
Product Collection
Get single user
user_id
User Collection
Delete user
user_id
User Collection
Update user role
user_id
User Collection
Get all orders
order_id
Order Collection
Delete a order
order_id
Order Collection
Get a single order
order_id
Order Collection

Level 3:


User
Create Product
product-name,description,category, stock, image etc
Product Collection
Delete product
product_id
Product Collection
Update product
product_id,product-name,description,category, stock, image etc
Product Collection
Get all products
Product Collection
Get a single product details
product_id
Product Collection
Register
Name , Email, pasword,avathar
User Collection
Login
Email,password
User Collection
Logout
user_id
User Collection
Create review
comment
Product Collection
Delete review
review_id
Product Collection
Get all reviews
Product Collection


User
Reset password
user_id
User Collection
Forgot password
user_id
User Collection
Get logged user details
user_id
User Collection
Update password
user_id
User Collection
Update user profile
user_id
User Collection
Create order
delivering-details, product-details
Order Collection
Get a single order
order_id
Order Collection
Logged user order
order_id
Order Collection
Delete order
order_id
Order Collection

3.3 Interface Design

An interface design elements for the software tell how information flows into and out of the system and hoe it is communicated among the components as part of

The architecture.

3.3.1 Input Design

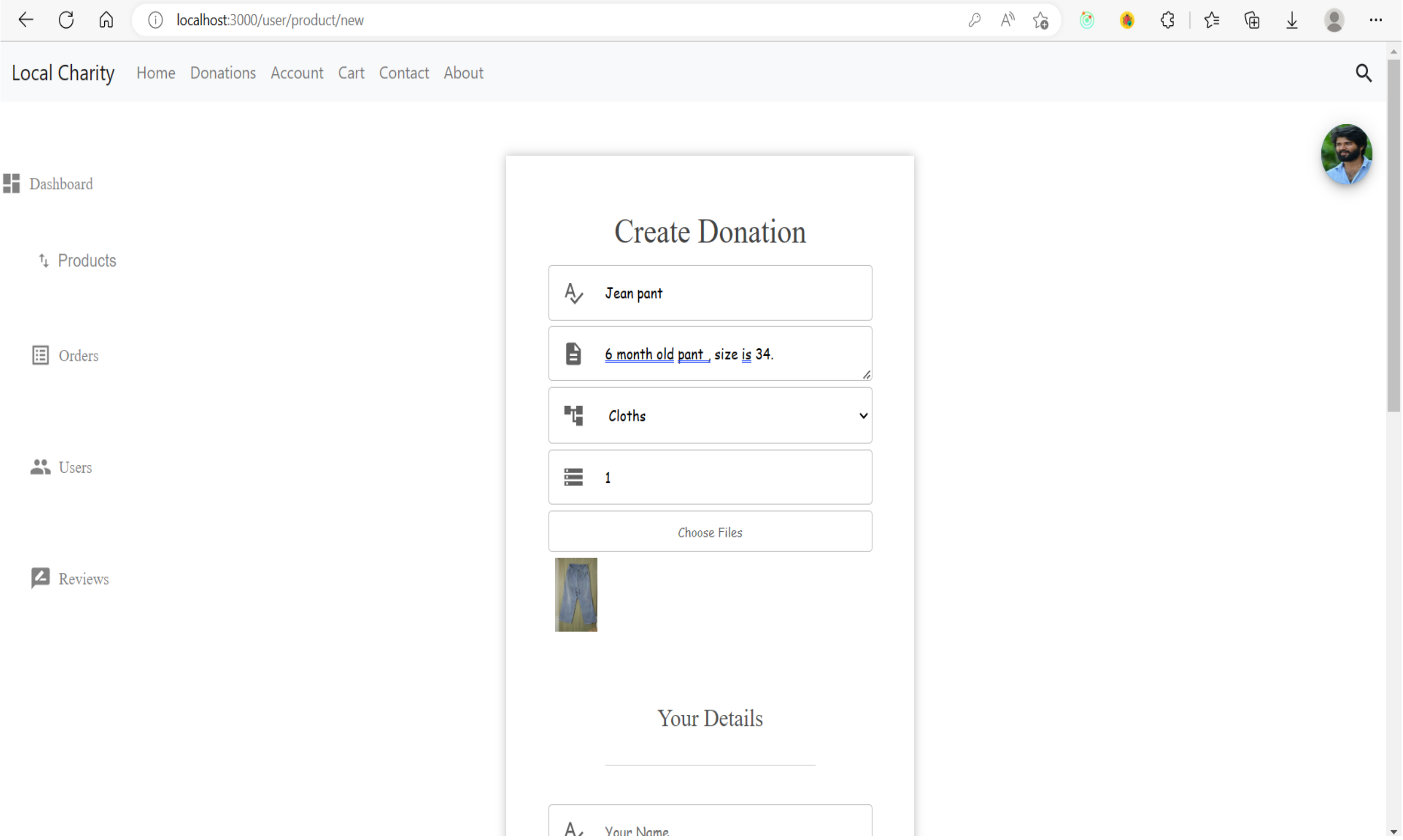
Input design is the link between the information system and users and those steps that are necessary to put transaction that are into a usable from processing data entry. Instructing the computer to read data from a written printed document can active the activity of putting data into the computer for processing or it can occur by errors, avoid delay, and keeping process simply. System analyst decides the following input design details.

- What data to input?
- What medium to use?
- How the data is arranged and code?

The following are the major input forms present in the project:

- Login form
- Registration form for Users.
- Form for creating a donation in the project.
- Profile editing form.
- Form for change password.
- Form for role changing.
- For submit review

Example: Input design for a Adding donation form:



Create Donation

Jean pant

6 month old pant , size is 34.

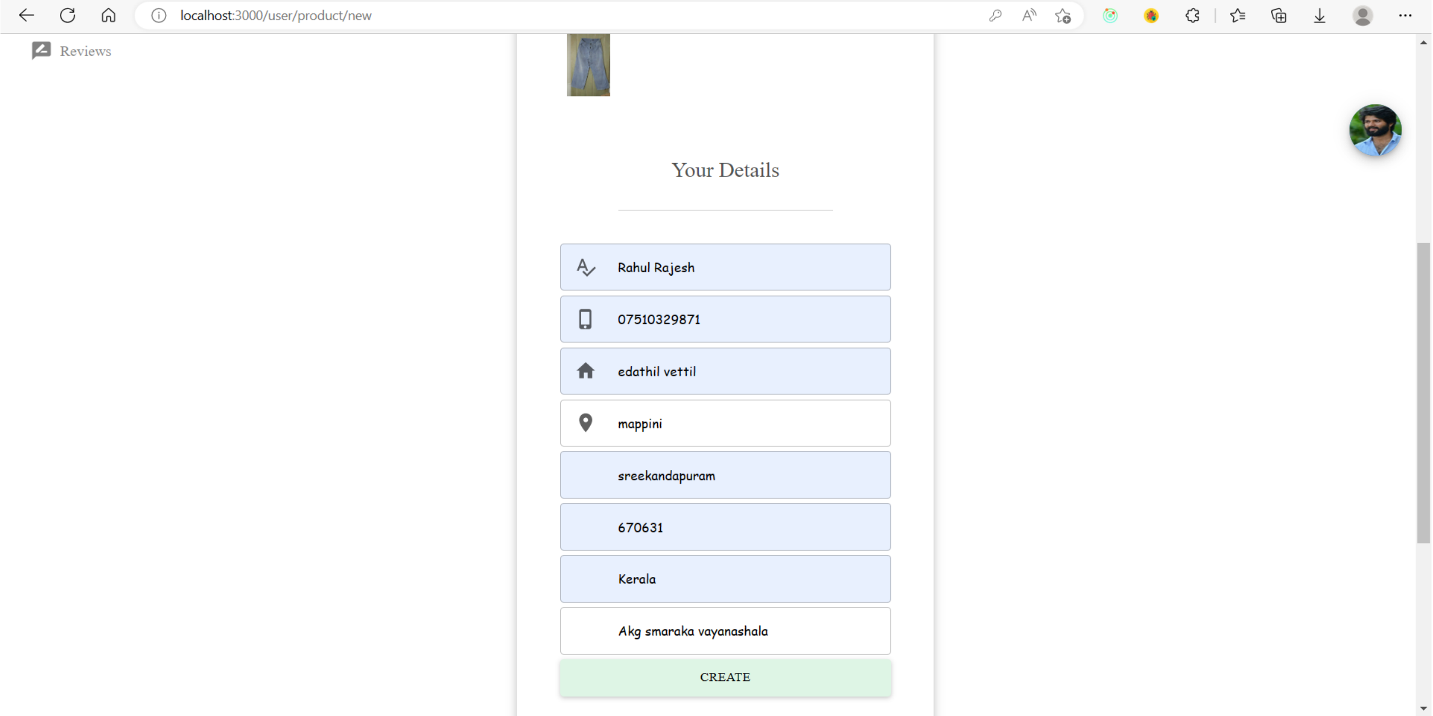
Cloths

1

Choose Files

Your Details

Your Name



Your Details

Rahul Rajesh

07510329871

edathil vettil

mappini

sreekandapuram

670631

Kerala

Akg smaraka vayanashala

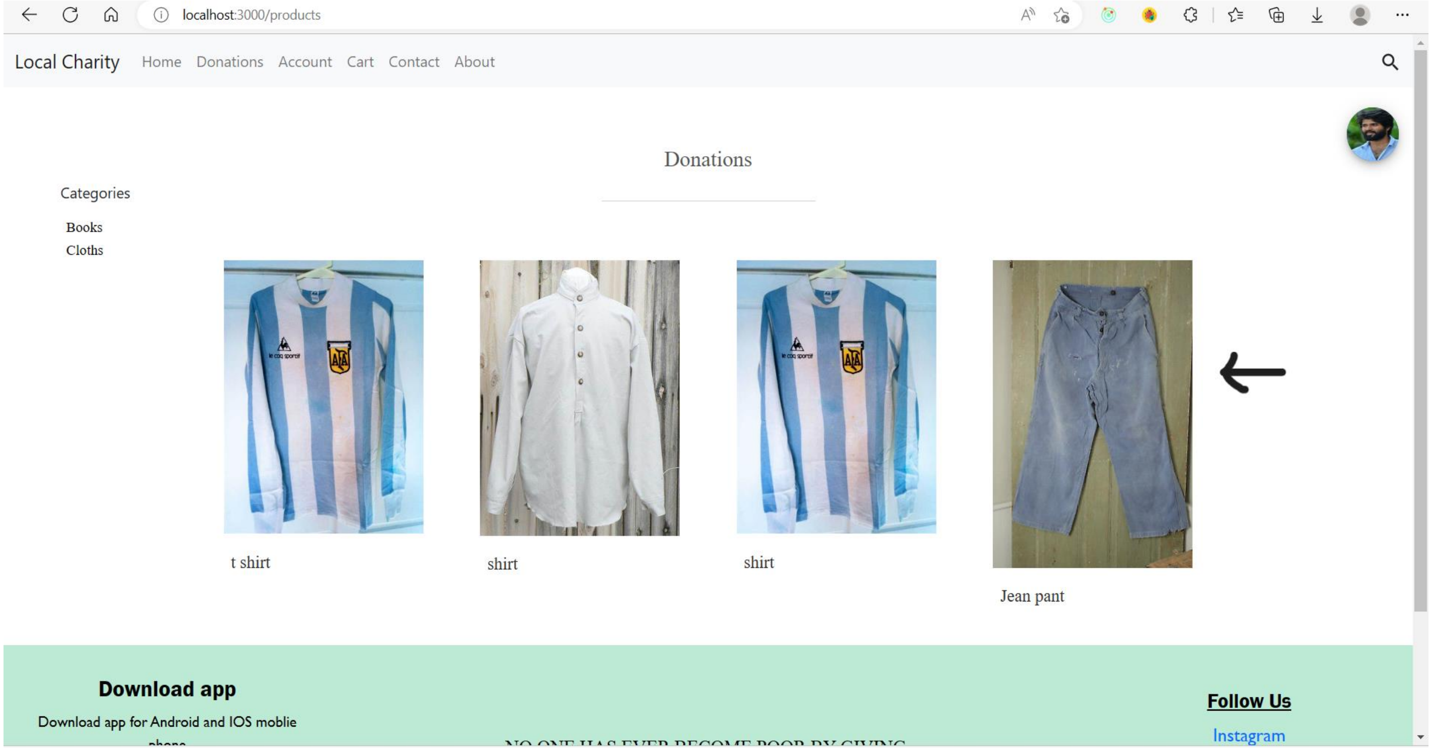
3.3.2 Output Design

Designing computer should processed in well thought out manner. The term output means any information produced by the information system whether printed or displayed. Output design is a process that involves designing necessary output that have to be used by various users according to requirement. The efficient intelligent output design should remove the system relationship with the users and help in decision making.

The following are the main output forms present in this project:

- View user details.
- View donations.
- View single donation details.
- View all orders for admin .
- View single order for admin.
- View the review of a donation.

Example: design for viewing created donations:



Donations

Categories

Books

Cloths

t shirt

shirt

shirt

Jean pant

Download app

Download app for Android and IOS moblie phone

NO ONE HAS EVER BECOME POOR BY GIVING

Follow Us

Instagram

CHAPTER 4
CODING

4.CODING

4.1 Description of software

MERN Stack is a Javascript Stack that is used for easier and faster deployment of full-stack web applications. MERN Stack comprises of   4 technologies namely: MongoDB, Express, React and Node.js. It is designed to make the development process smoother and easier.

Each of these 4 powerful technologies provides an end-to-end framework for the developers to work in and each of these technologies play a big part in the development of web applications.

MERN Stack components:

1. **MongoDB: Cross-platform Document-Oriented Database**

MongoDB is a NoSQL database where each record is a document comprising of key-value pairs that are similar to JSON (JavaScript Object Notation) objects. MongoDB is flexible and allows its users to create schema, databases, tables, etc. Documents that are identifiable by a primary key make up the basic unit of MongoDB. Once MongoDB is installed, users can make use of Mongo shell as well. Mongo shell provides a JavaScript interface through which the users can interact and carry out operations (eg: querying, updating records, deleting records).

2. **Express: Back-End Framework:**

Express is a Node.js framework. Rather than writing the code using Node.js and creating loads of Node modules, Express makes it simpler and easier to write the back-end code. Express helps in designing great web applications and APIs. Express supports many middle wares which makes the code shorter and easier to write.

3. **React: Front-End Library**

React is a JavaScript library that is used for building user interfaces. React is used for the development of single-page applications and mobile applications because of its ability to handle rapidly changing data. React allows users to code in JavaScript and create UI components.

4. **Node.js: JS Runtime Environment**

Node.js provides a JavaScript Environment which allows the user to run their code on the server (outside the browser). Node pack manager i.e. npm allows the user to choose from thousands of free packages (node modules) to download.

4.2 Coding Principle

The input to the coding phase is the design document. During coding phase, modules identified in the design document are coded according to the module specification. Objectives of coding phase are, to transform design into code and unit test the code.

Coding guidelines

- Code should be easy to understand.
- Don’t take pride in cryptic code.
- Code should be well documented.
- Comments should be present.
- Functions should be small.

Example code of home page:

```
import React, { Fragment,useEffect } from 'react'
import {CgMouse} from "react-icons/all"
import "./Home.css"
import ProductCard from "./ProductCard"
import MetaData from '../layout/metadata'
import {getProduct} from "../../actions/productAction"
import{useSelector,useDispatch} from "react-redux";
import Loader from '../layout/Loader/Loader';
import Footer from "../layout/footer/Footer"



const Home = () => {
  const dispatch=useDispatch();
  const {loading,products,productsCount}=useSelector(
    (state)=>state.products)
useEffect(()=>{

    dispatch(getProduct())
  },[dispatch]);
return (
    <Fragment>
      {loading?<Loader/>:   <Fragment>
<MetaData title="Local Charity"/>
<div className='banner'>
    <p>Welcome to local charity</p>
    <h1>Find suitable products below</h1>
    <a href="#container">



<button>
        Scroll<CgMouse/>
      </button>
    </a>

</div>
<h4 className="homeheading"><u>Recently added donations</u></h4>
<div className='container' id='container'>

{products && products.map(product=>(<ProductCard product={product}/>))}
</div>
  <Footer/>
 </Fragment>

      }
    </Fragment>

  )
}
export default Home
```

CHAPTER 5

SYSTEM TESTING

5. SYSTEM TESTING

Testing is the penultimate step of software development. An elaborate testing of data is prepared and the system is using test data while doing testing, errors are noted and correction is made. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully.

System testing is aimed at ensuring the system accurately before live operation commences. Testing is vital to the system. A series of testing are performed for the proposed system before the system is ready for user acceptance testing success of the system.

The various steps in testing the system can be listed as below:

1. Running the program to identify any errors that might have occurred while feeding the program into the system.

2. Applying the screen formats to regulate users to extend, so that the screens are comprehensible to the system.

3. Presenting the formats to administration for the purpose of obtaining approval and checking if any modification has to be done. Obtaining feedbacks from customer and analyzing the scope for improvement.

4. Checking the data accessibility from the data server and whether any improvement is needed or not.

The entire testing process can be divided in to four parts

□ Unit testing

□ Integration testing

□ System testing

□ Validation testing

5.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software designs the module. To check whether each module in the software works properly so that it gives input. All validations and conditions are tested in the module level in the unit test. Control path are tested to ensure the information property flows into an out of the program unit an out of the program unit under test. Boundary condition is tested to ensure that the modules operate at boundaries. All independent paths through the control structures ensure that all statements in a module have been executed at least once.

In unit testing

- Module interface is tested to ensure that information properly flows into and out of the program under test.

- Local data structures are examined to ensure that data stored temporarily maintainists. integrity during all steps in algorithm execution.

- Boundary Condition is tested to ensure that the module operates properly at boundaries established to limit or restrict processing.

- All independent paths through the control structures are executed to ensure that all statements in the module have been executed at least once.

- Error handling paths are also tested.

TEST CASES

Login Form

No	Test Scenario	Expected result	Observed result	Result
1	Enter wrong Username and password	Display login form again with a warning message	Message displayed	Pass

2	Enter correct user name and password	Display login form again with a warning message	Message displayed	Pass
3	Enter correct user name and password	The whole registered user can log into the system	Appropriate home page displayed for login	Pass
4	Press login button without filling the username and password	Display message to fill the fields	Warning message displayed	Pass

User’s Registration form

No	Test Scenario	Expected result	Observed result	Result
1	Form displayed	Display the registration form	Form loaded	Pass

2	Enter the mobile number in pass characters	Display an invalid message	Invalid message displayed	Pass
3	Enter the mobile no more than and less than 10 digits	Display an invalid message	Invalid message	Pass
4	Click the register button without filling the details	Display a warning message to fill the fields.	Warning message is displayed	Pass
5	Click on register button with filled fields	Accept the details	Registration successfully done	Pass

a) BLACK BOX TESTING

This testing method focuses on the functional requirements of the software. It attempts to find out the errors of the following categories such as incorrect and missing functions, interface error, error in data structures, performance error an initialization.

b) WHITE BOX TESTING

This testing method is called path testing. It is a test case design method that uses the control structures to the procedural design to drive test case. In this system, unit testing has been successfully handled. The test data was given to each and every module in all respects and got the desired output. Each module has been tested found working properly.

5.2 INTEGRATION TESTING

The major concerns of integration testing are developed an incremental strategy that will limit the complexity of the entire actions among components as they are added to the system. Developing components as they are added to the system, developing an implementation and integration schedules that will make the modules available when needed and designing test cases that will demonstrate the validity of the evolving system though each program works individually, they should work together. This also referred to as interfacing. Each database or table manipulation operation was written as single program was tested again with numerous test data to check for its functionality.

The purpose of integration testing is to verify functional, performance and requirements placed on major design items. These design items or group of units, are exercised

through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystem is exercised through their input interface.

No.	**Input/procedure**	**Expected Result**	**Actual Result**	**Pass/Fail**
1	Check the value pass between different forms	Appropriate operations of different forms	Same as expected	pass

	are appropriate format.			

5.3 DATA VALIDATION TESTING

Data validation is one to see whether the corresponding entries made in the tables are correct. Proper validations are one in case of insertions and updating of tables if any case arises the proper error messages or warning, if any, has to be displayed.

5.4 SYSTEM TESTING

When a system is developed it is hope that it perform properly. The main objectives of system testing are

- To ensure during operation the system will perform as per specification.
- To make sure that the system meets user's requirements user's operations.
- To verify that the controls incorporated in the system function as intended.
- To see when correct input are fed to the system the output are correct.
- To make sure that operation, incorrect input and output will be detected.

System testing is used test the entire system(Integration of the all modules).It also tests to find the discrepancies between the system and the original objective, current specification and system documentation. The entire system is checked to correct deviation to achieve correctness.

No.	**Input/procedure**	**Expected Result**	**Actual Result**	**pass/Fail**

1	Check whether indented output is obtained.	All operations are carried out properly	Same as expected	Pass

CHAPTER 6

SYSTEM IMPLEMENTATION

6. SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system and giving confidence on the new system for the uses that it will work efficiently and effectively. It involves careful planning, investigation of current system and its constraints of Implementation, design of methods to achieve the changeover, an evaluation of, and change over methods.

Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation, on implementation coordinating committee based on policies of individual organization has been appointed.

The implementation process begins with preparing the plan for the implementation for the system. According to this plan, the activities are to be carried out, discussion made regarding the equipment and resources and the additional equipment as to be acquired to implement the new system.

The implementation is the final and important phase. The most critical stage in achieving successful new system and in giving the user confidence that the new system will work and be effective. The system can be implemented only after through testing is done and if it found to working according to the specification.

This method also offers the greatest security. Since the old system can take over if the errors are found or inability to handle certain type of transaction while using new system.

CHAPTER 7

CONCLUSION

7. CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The project “**Local Charity**” has been completed successfully and marks the specification of those who have guided us through this project. The constraints are met and overcame successfully. The system is designed as like it was decided in the design phase.

The website was developed in MERN stack with visual studio. It is expected to live up to the objective for which it was designed. The Local Charity is a web-based application. Which focus on helping people in need. The system act as a mediator between donator and user. The application focus a small locality based charity . This application enable as to donate items such as books, cloths etc . In scenario where we have items no longer need as and if we are willing to donate the items, the web application works as a mediator.

In this project, first an attempt has been made to find the need for the system. To fulfill the needs , a detailed study has been designed in such a way that it is user friendly and easy to use. The developed system is flexible and modification is easy whenever necessary.

The application developed is designed in such a way that any further enhancement can be done with ease. All the suggestions forward in the website proposal have successfully been completed and the final threshold of the application has been crossed. This project has been completed satisfactorily within the specification.

7.2 FUTURE ENHANCEMENT

1.Social Media Integration: One of the future enhancements that can be added to the Local Charity web-based application is social media integration. This would allow users to share their donations on social media platforms, thus increasing awareness and encouraging more people to donate.

2.Online Payment Integration: Another enhancement that can be added is online payment integration. This would enhance users to make donations online using different payment methods, which would increase the convenience and accessibility of the system.

3.Multi-Lingual Support: The Local Charity application can be enhanced with multi-lingual support, which will enable users to access to access the system in their preferred language. This will improve user experience and increase the reach of the application.

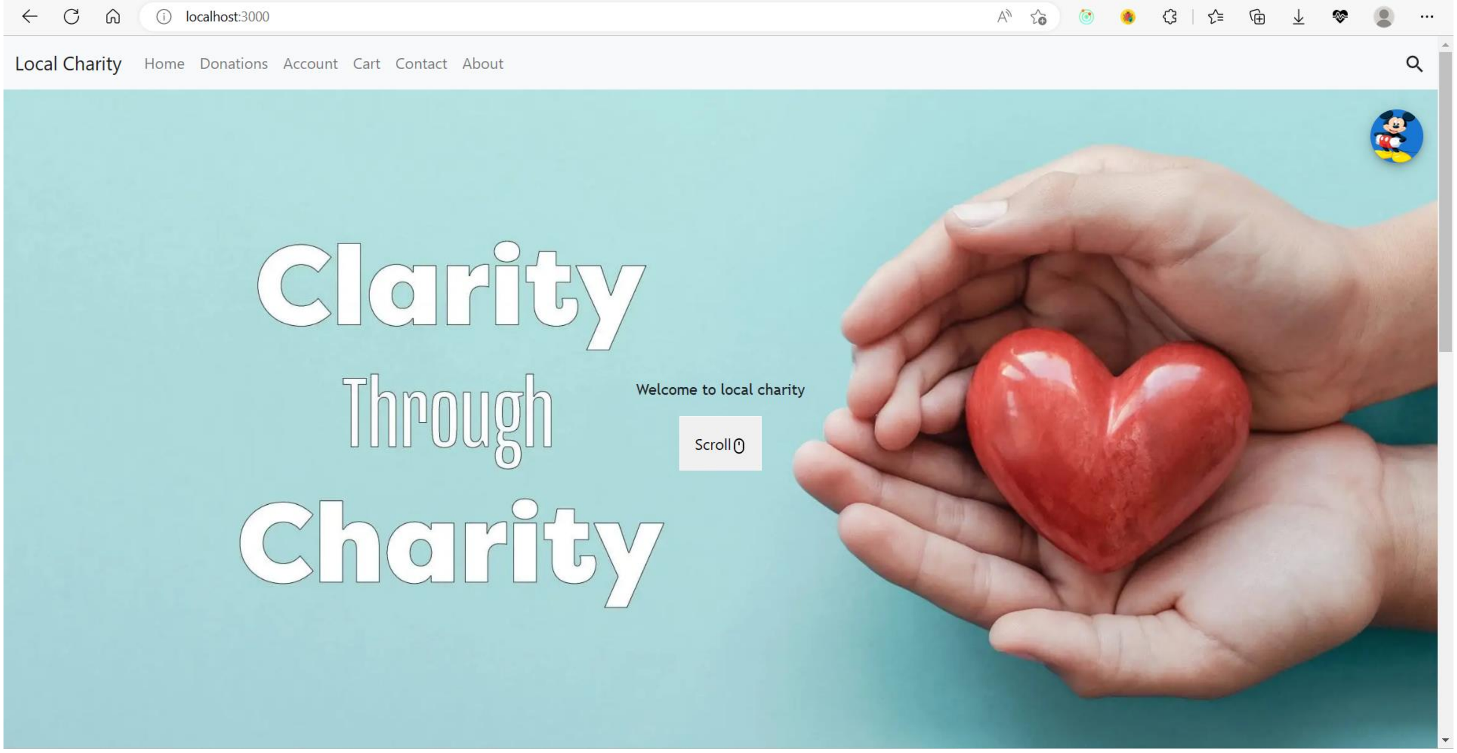
4.Mobile Application Development: Developing a mobile application for the Local Charity web-based application will make it easier for users to donate items on-the-go. This would also increase accessibility and convenience and allow for easier management of the system.

5.Integration with Local Stores: Integration with local stores can be an additional future enhancement. This would enable users to donate items directly at a nearby store and reduce the hassle of delivering the items to the admin or donator. It would also encourage more stores to participate in charity works.

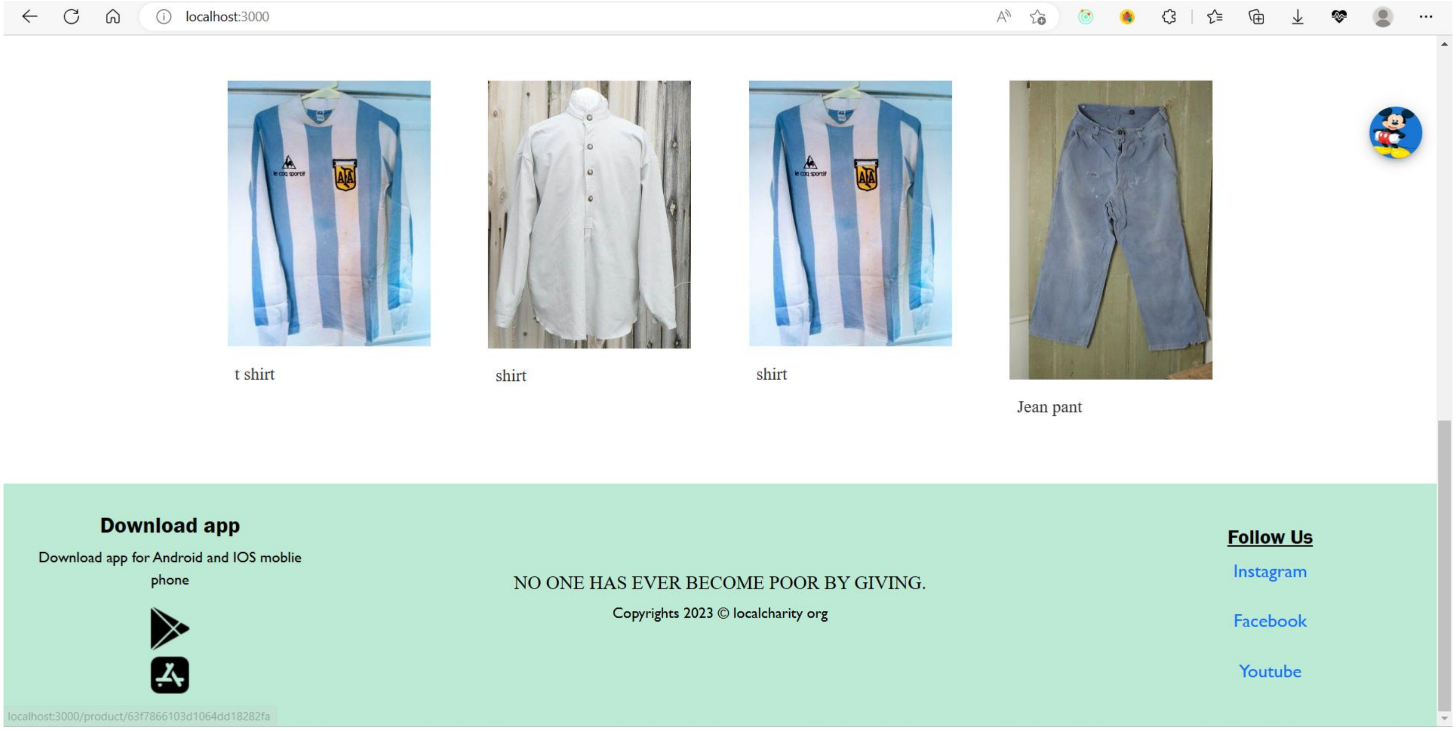
6.Analitics and Reporting: Adding analytics and reporting features can help the administrator to track the performance of the system and identify areas for improvement. It would provide insight into the user’s behavior , trends and preferences, which will be useful in decision making.

8. APPENDIX

Home page



localhost:3000
Local Charity
Home
Donations
Account
Cart
Contact
About
Clarity
Through
Charity
Welcome to local charity
Scroll



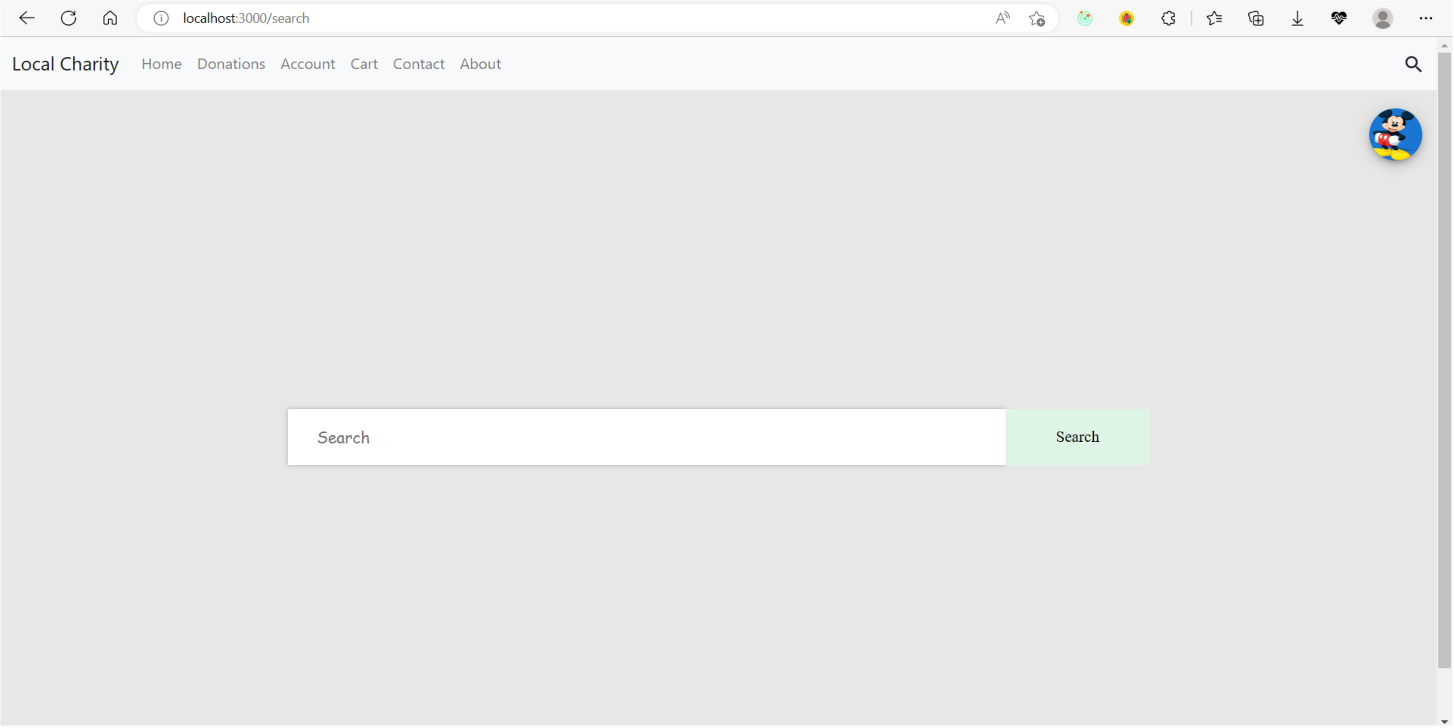
Download app

Download app for Android and IOS moblie phone

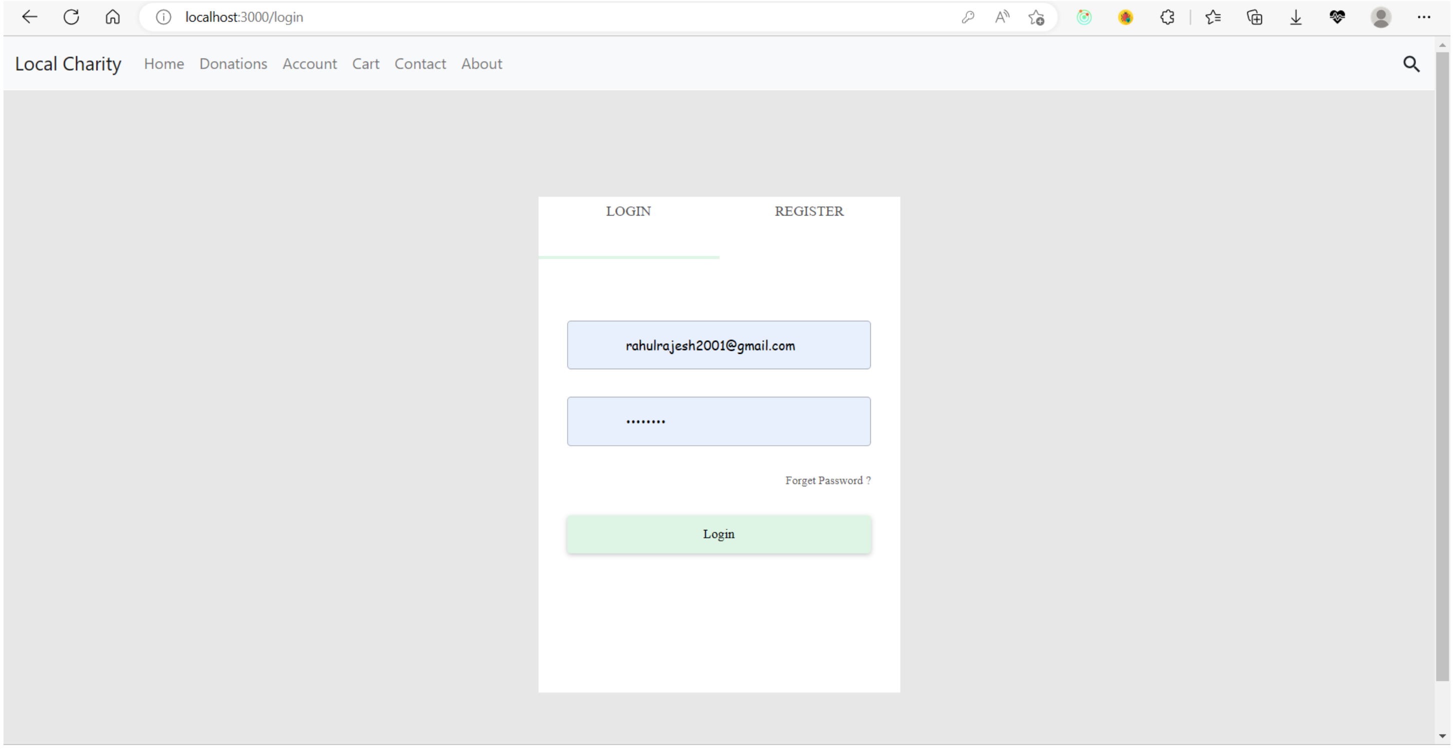
NO ONE HAS EVER BECOME POOR BY GIVING.

Copyrights 2023 © localcharity org

Search page:



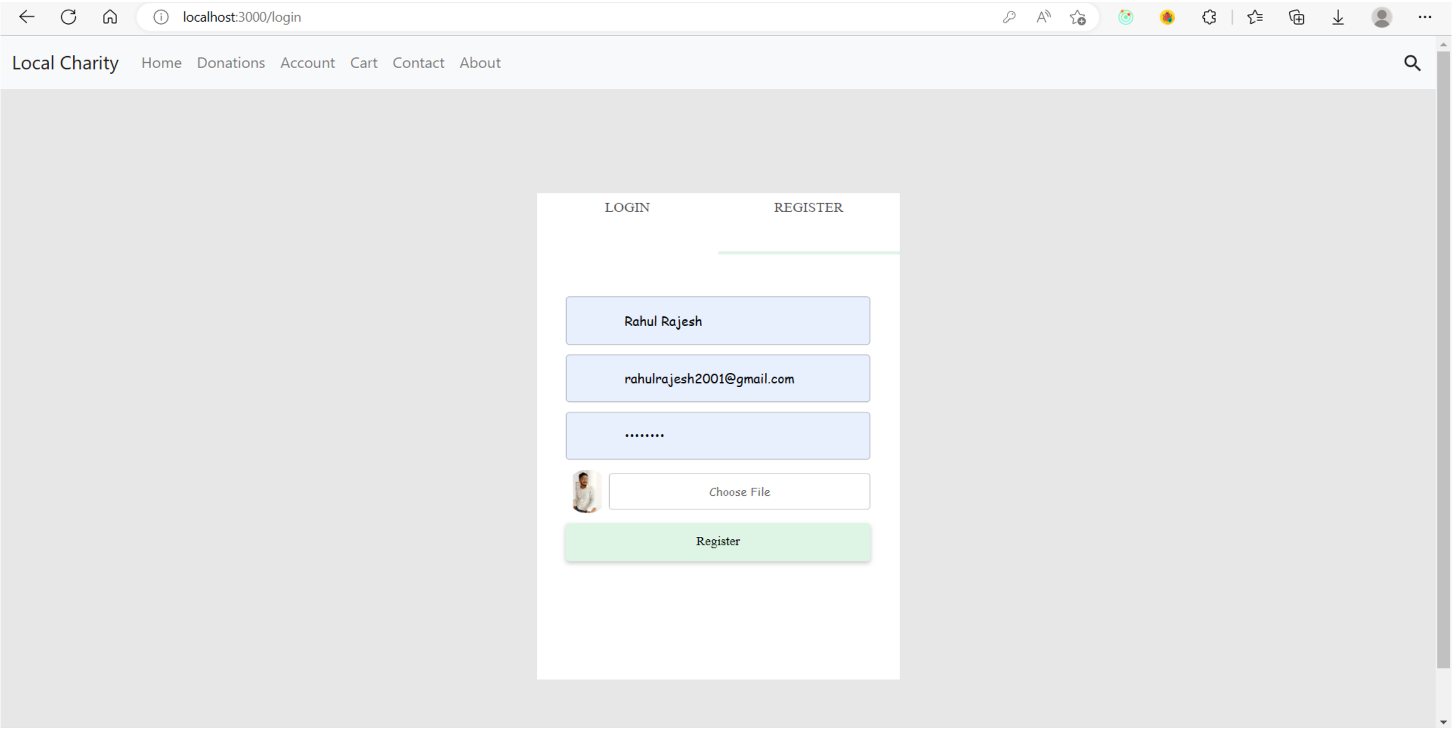
Login page:



rahulrajesh2001@gmail.com

••••••••

Register page:

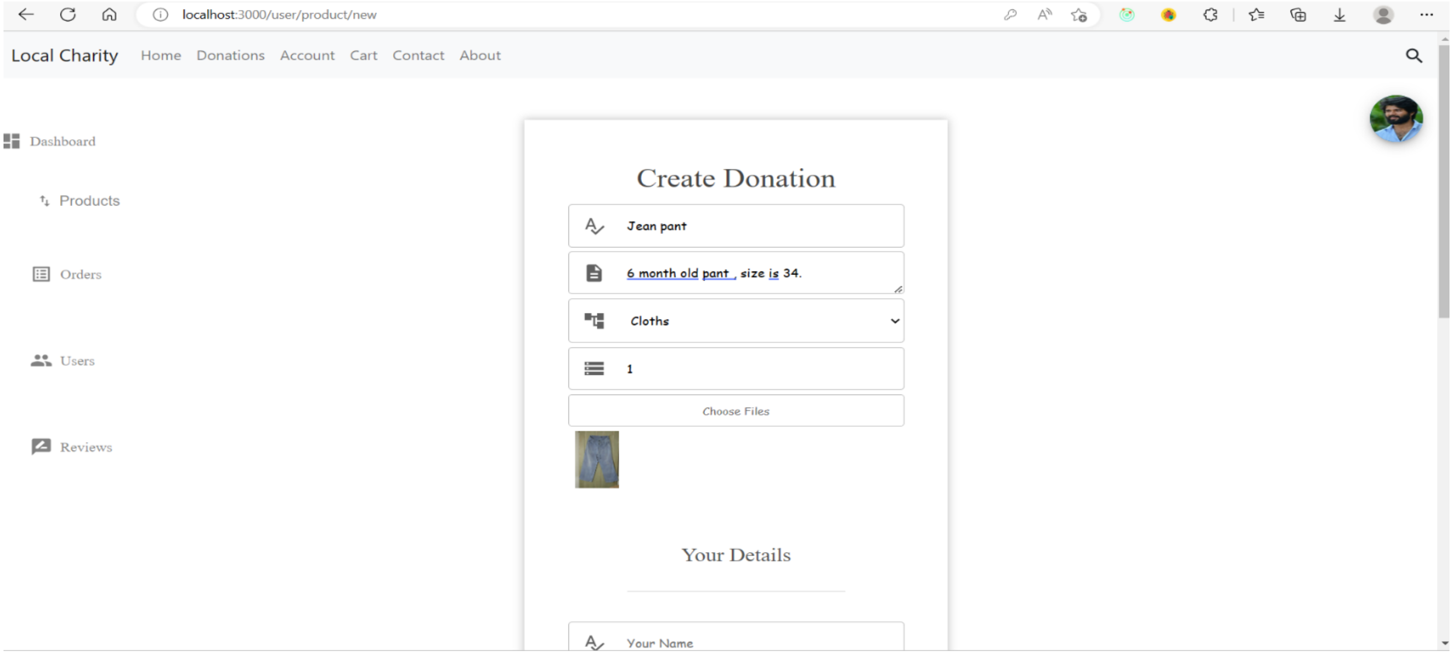


Rahul Rajesh

rahulrajesh2001@gmail.com

••••••••

Create donation page:



Create Donation

Jean pant

6 month old pant , size is 34.

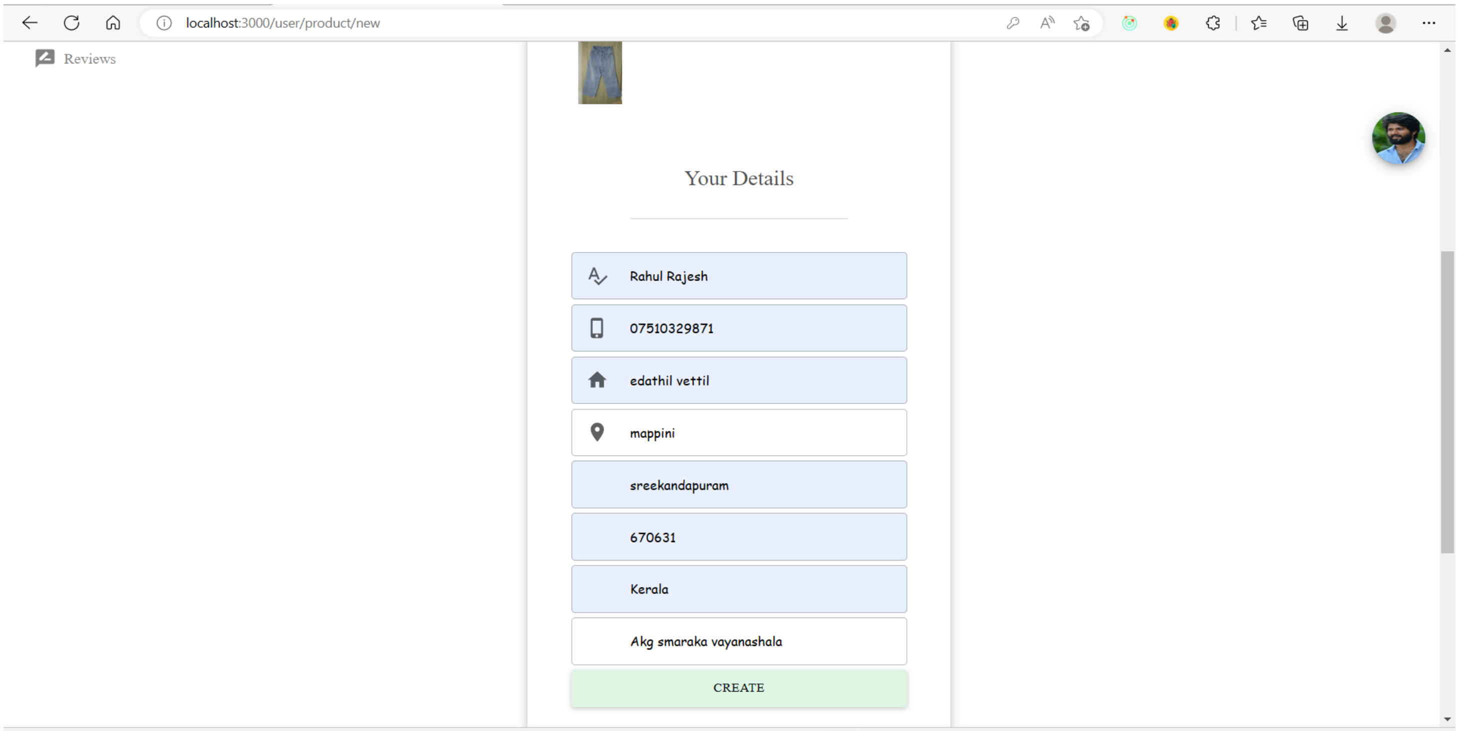
Cloths

1

Choose Files

Your Details

Your Name



Your Details

Rahul Rajesh

07510329871

edathil vettil

mappini

sreekandapuram

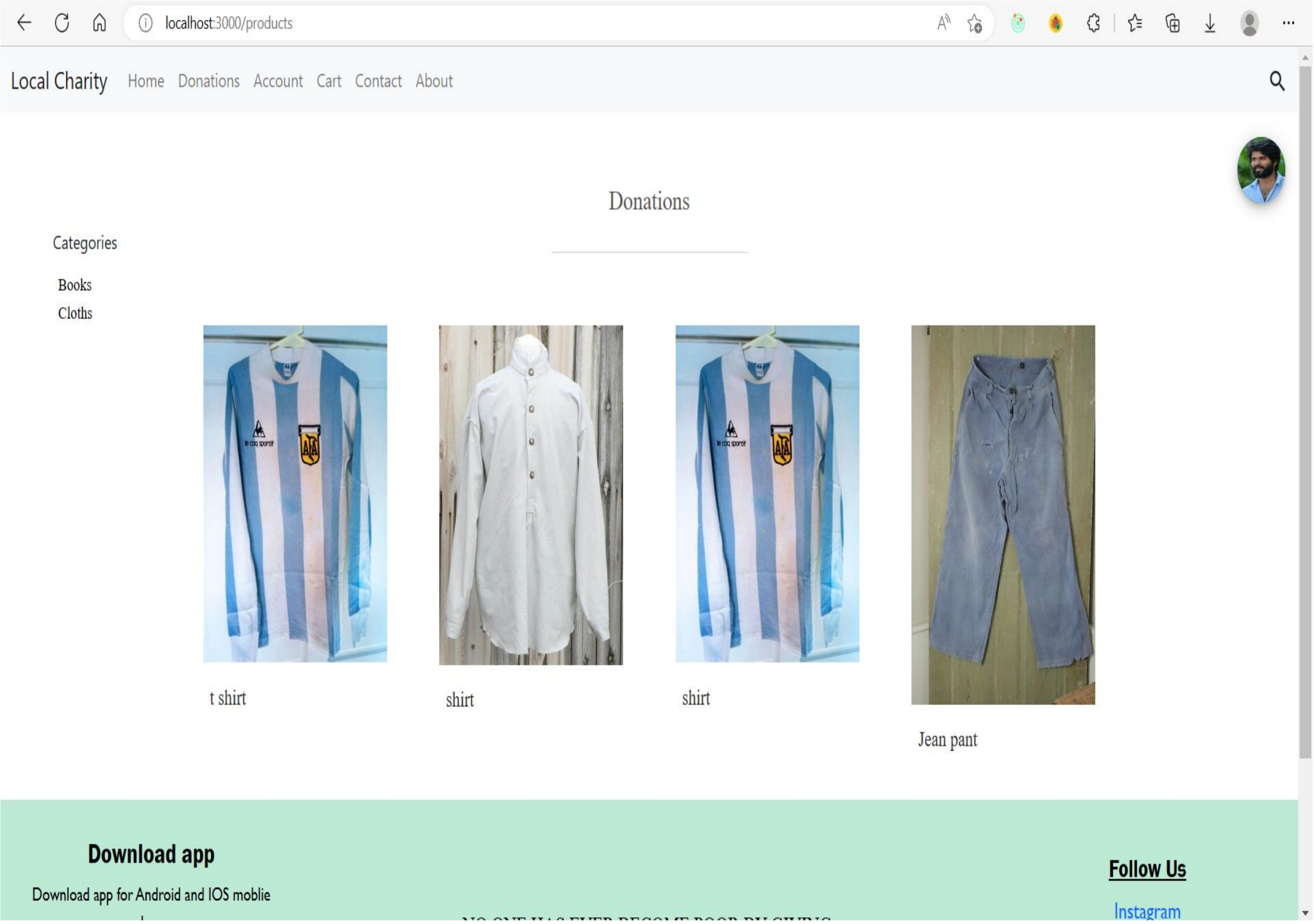
670631

Kerala

Akg smaraka vayanashala

CREATE

Donations page:



Donations

Categories

Books

Cloths

t shirt

shirt

shirt

Jean pant

Download app

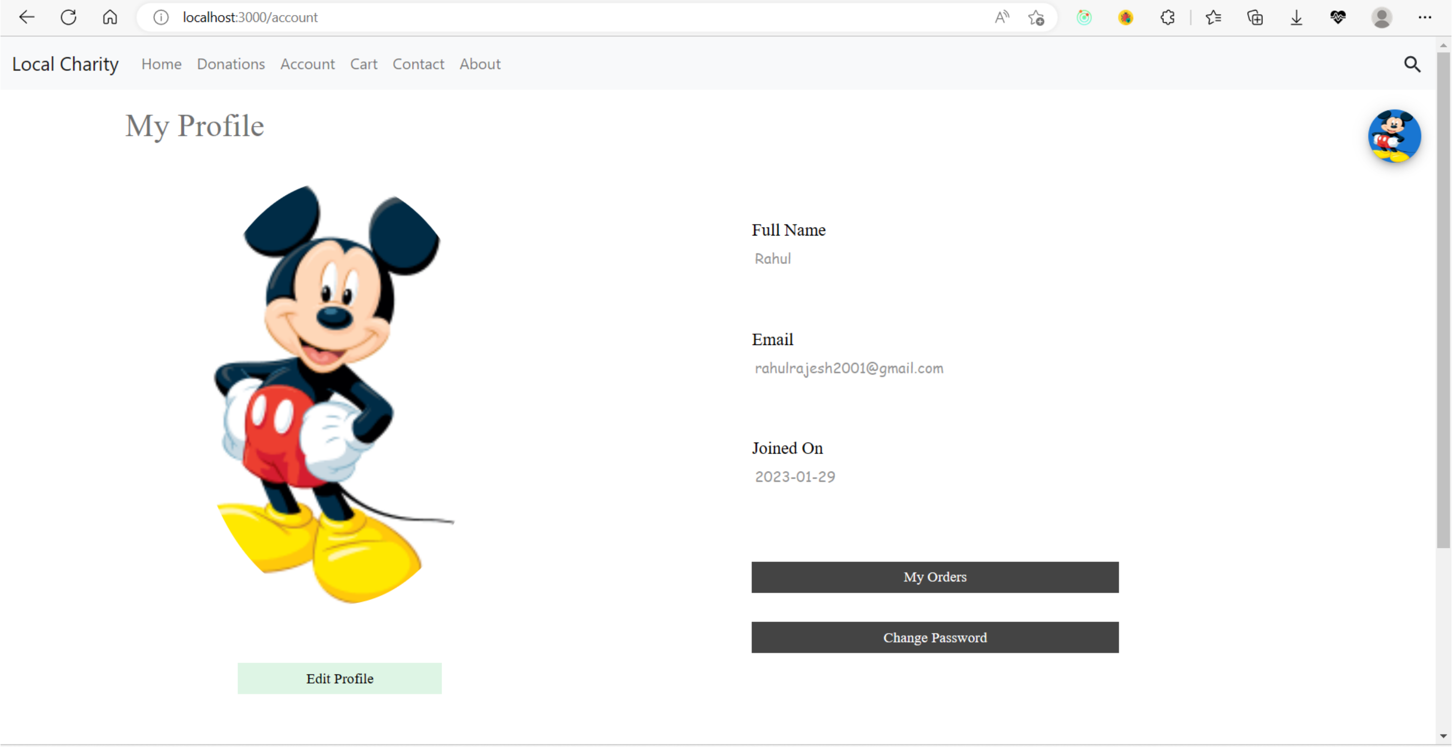
Download app for Android and IOS moblie

NO ONE HAS EVER BECOME POOR BY GIVING

Follow Us

Instagram

Account page:



My Profile

Full Name

Rahul

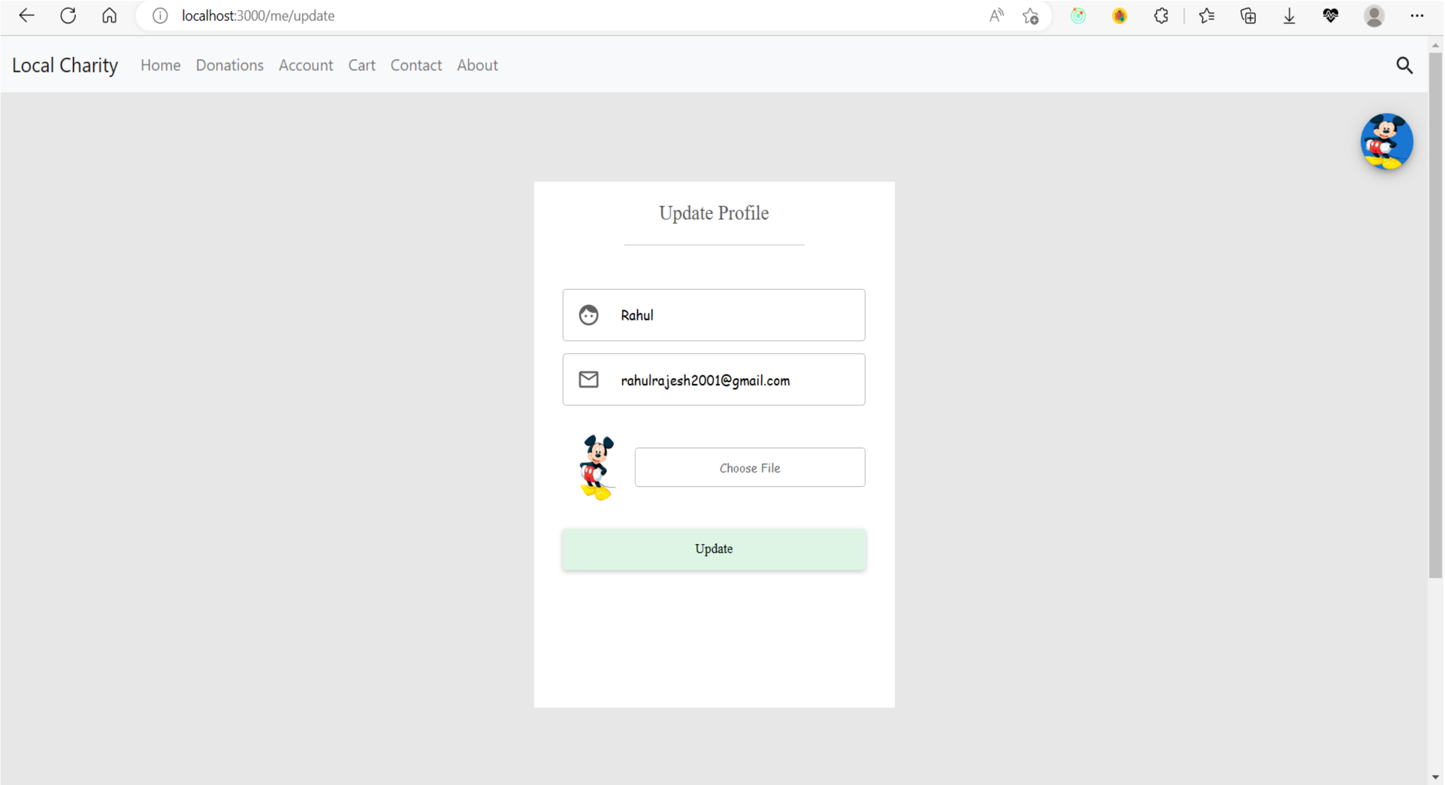
Email

rahulrajesh2001@gmail.com

Joined On

2023-01-29

Update profile page:

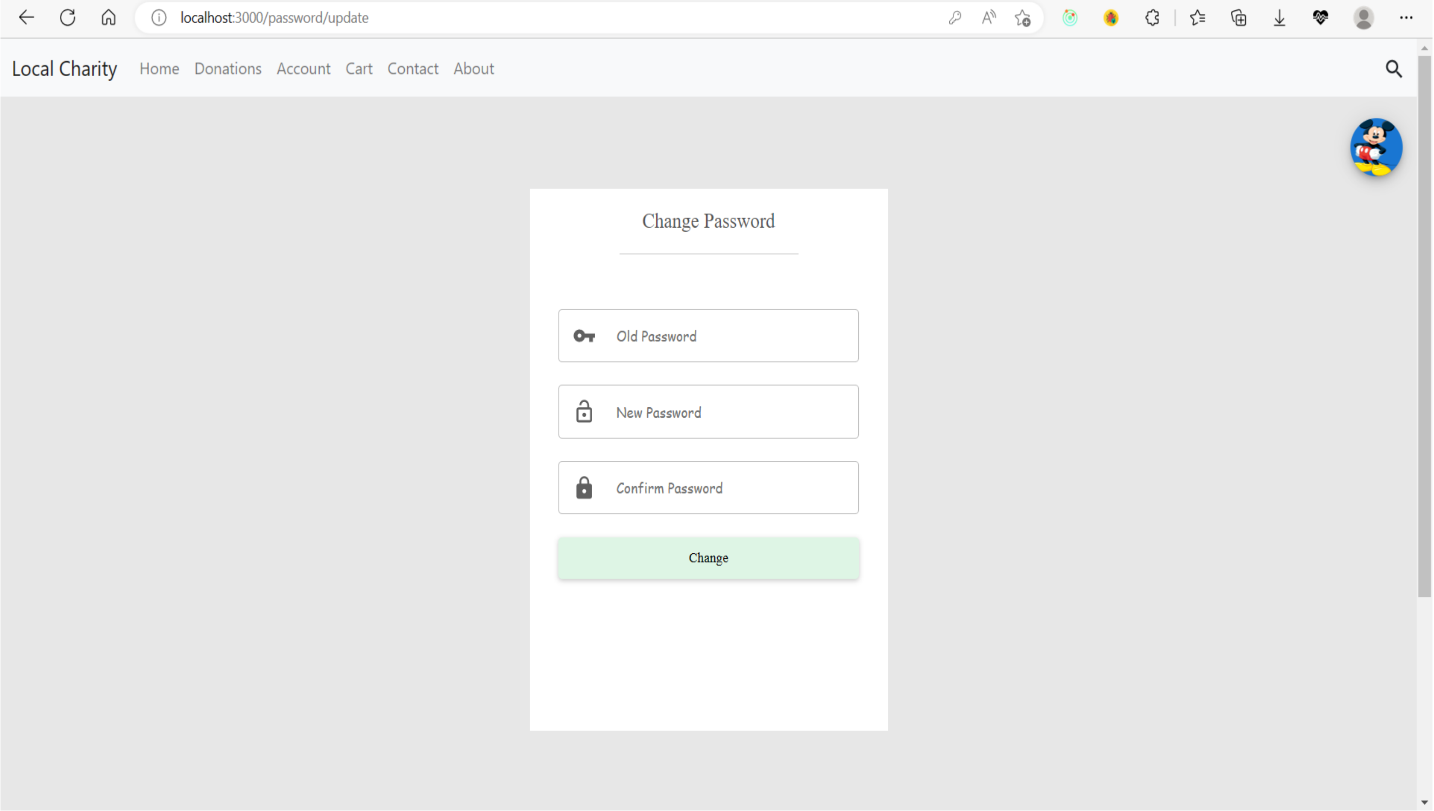


Update Profile

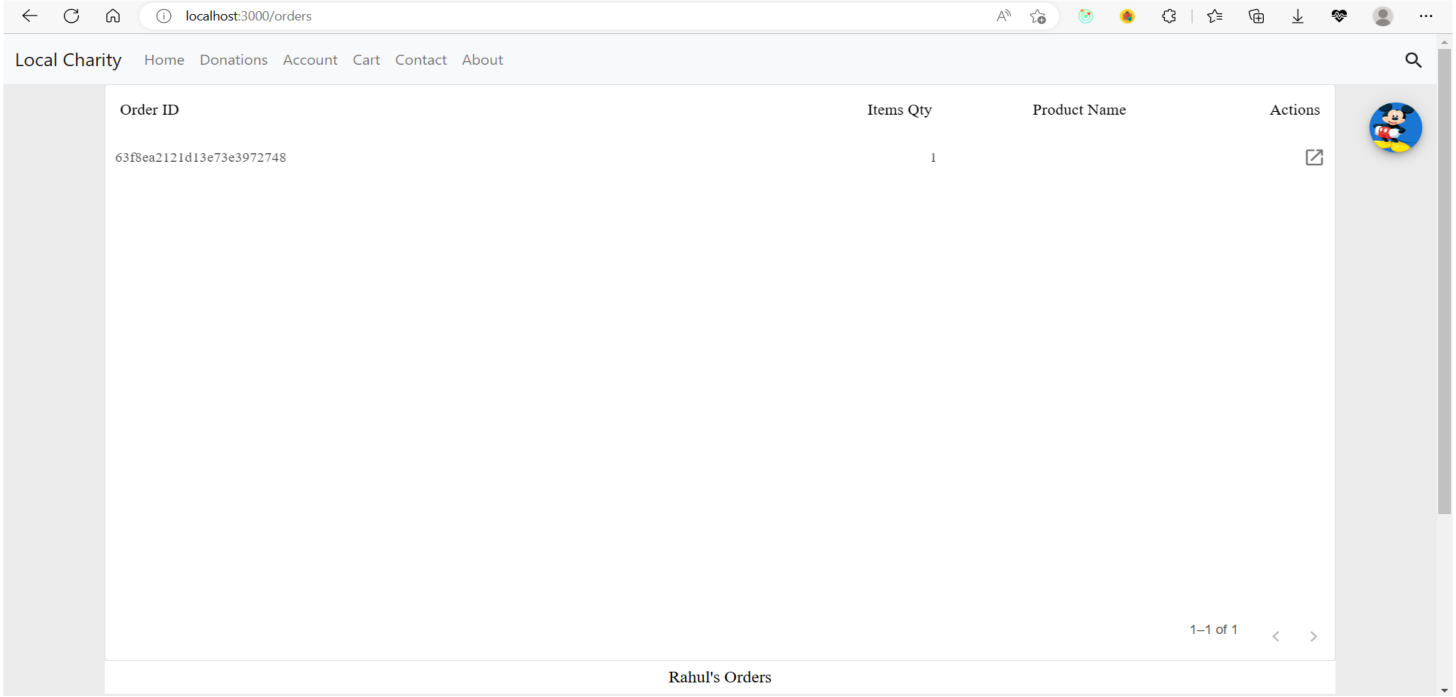
Rahul

rahulrajesh2001@gmail.com

Change password page:

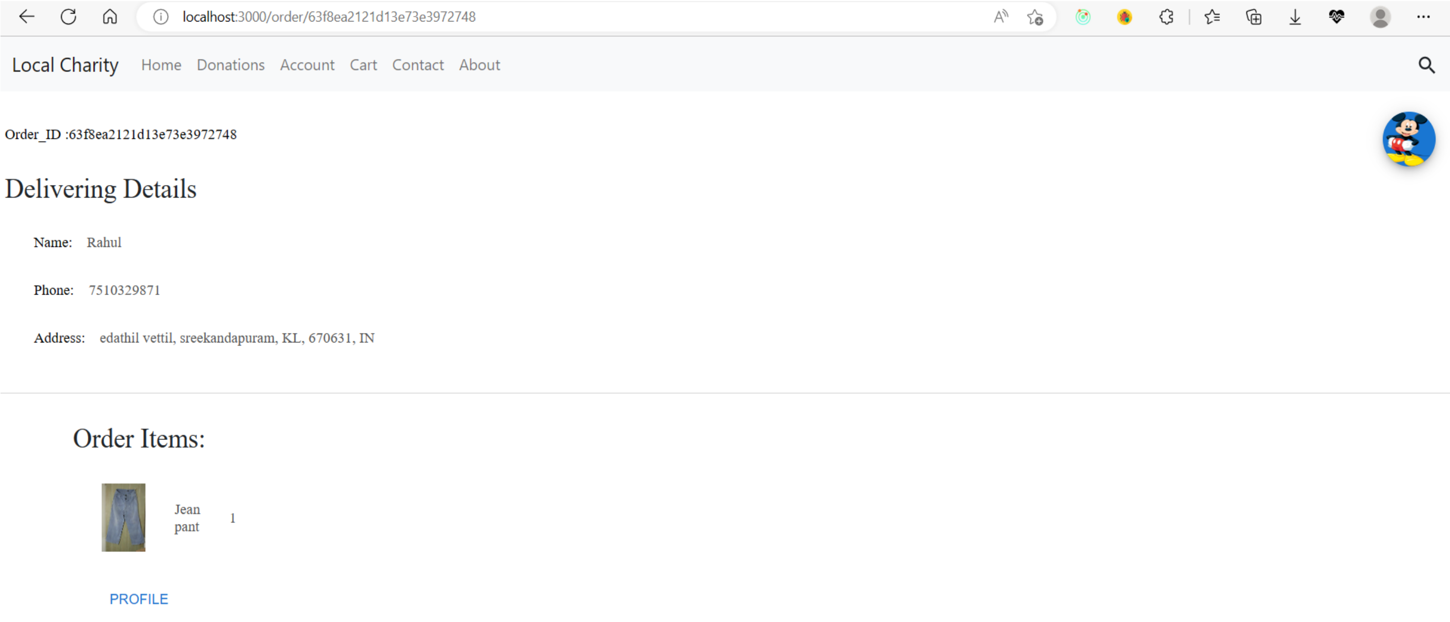


Order and Order details page:



Order ID	Items Qty	Product Name	Actions
63f8ea2121d13e73e3972748	1		

Rahul's Orders



Order_ID :63f8ea2121d13e73e3972748

Delivering Details

Name: Rahul

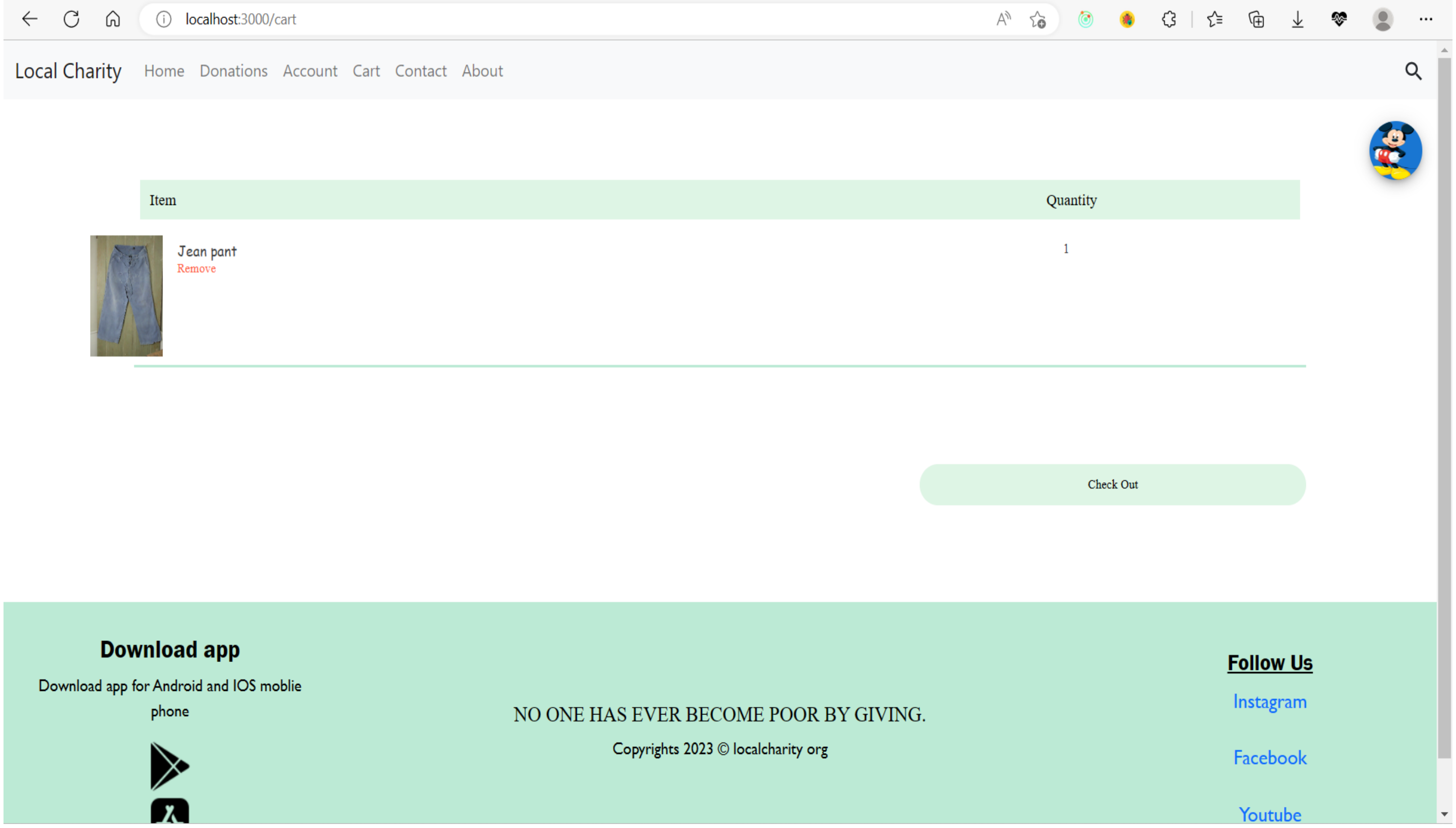
Phone: 7510329871

Address: edathil vettil, sreekandapuram, KL, 670631, IN

Order Items:

Jean pant 1

Cart page:



Item	Quantity
Jean pant Remove	1

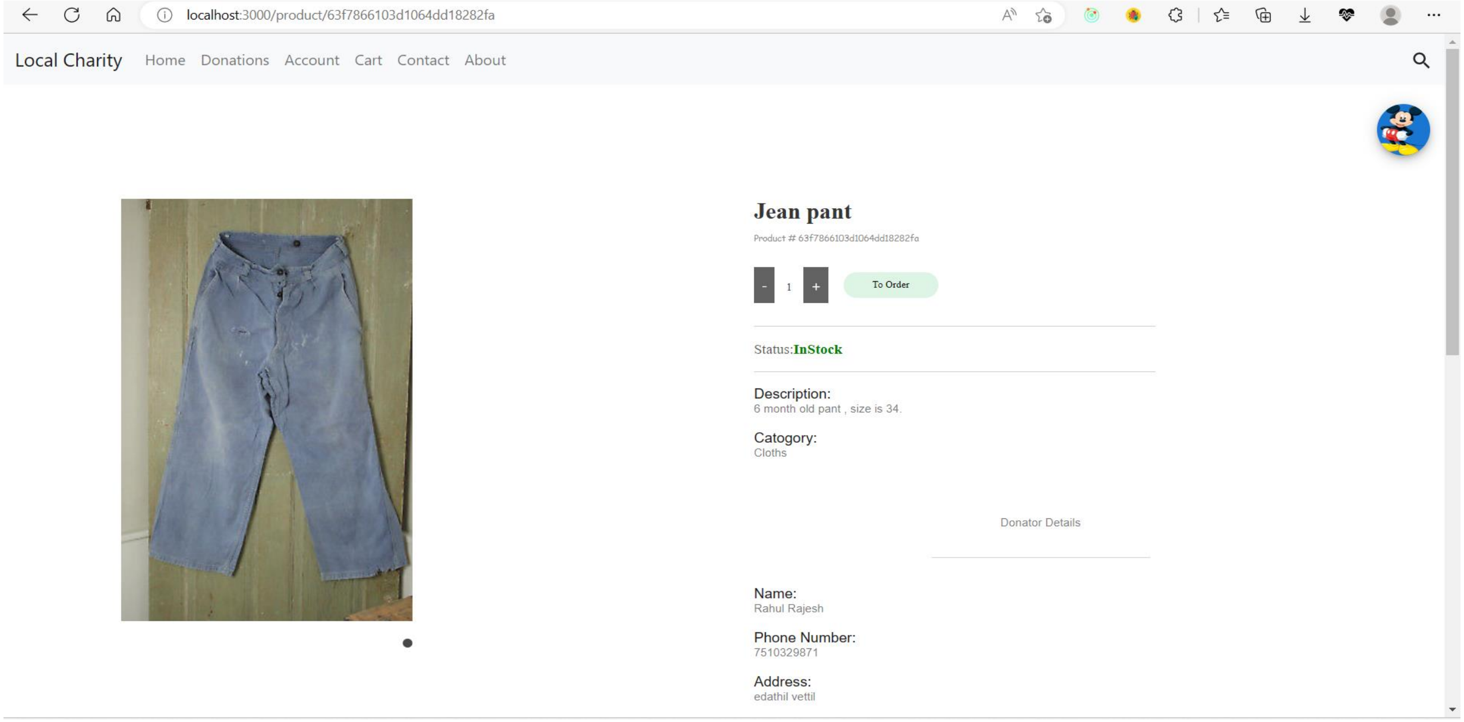
Download app

Download app for Android and IOS moblie phone

NO ONE HAS EVER BECOME POOR BY GIVING.

Copyrights 2023 © localcharity org

Product details page:



localhost:3000/product/63f7866103d1064dd18282fa

Local Charity Home Donations Account Cart Contact About

Jean pant

Product # 63f7866103d1064dd18282fa

- 1 + To Order

Status:**InStock**

Description:
6 month old pant , size is 34.

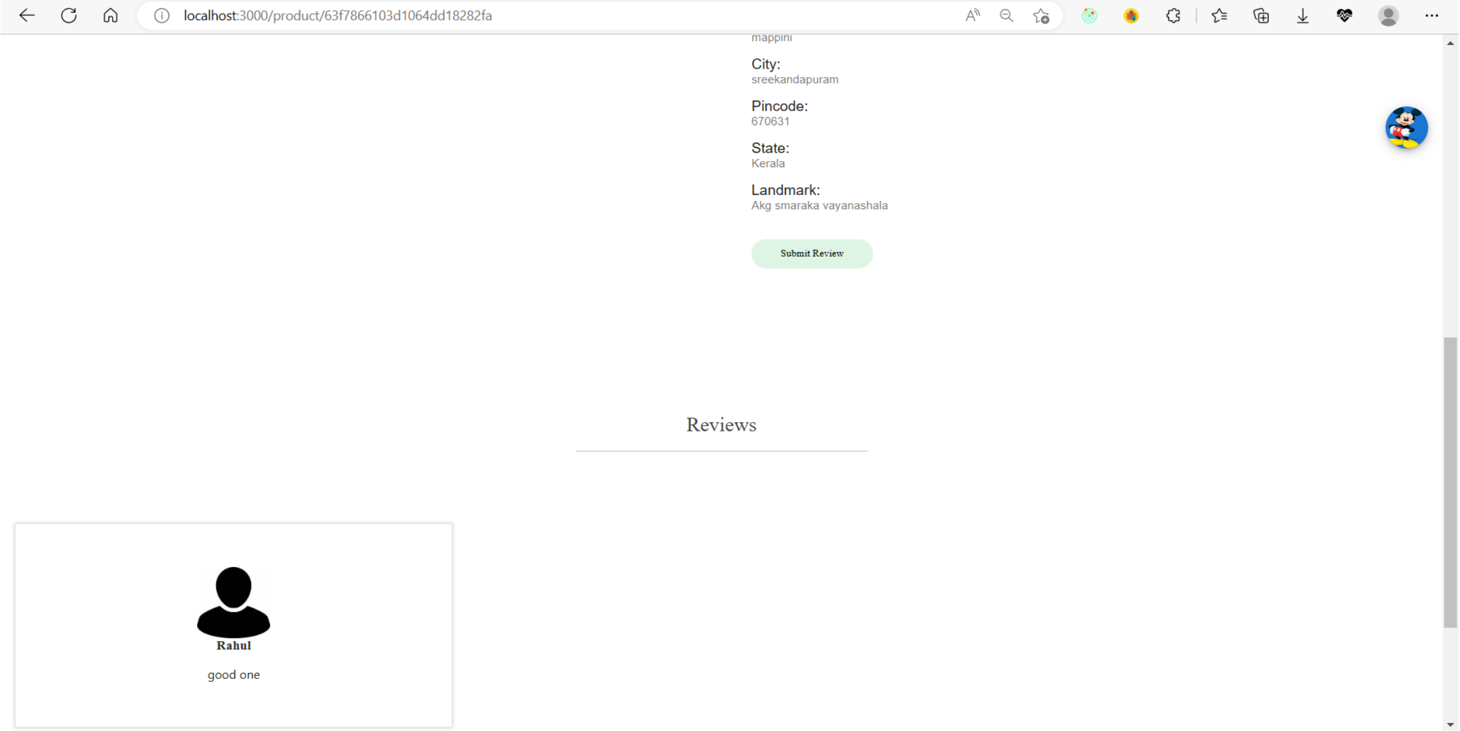
Catogory:
Cloths

Donator Details

Name:
Rahul Rajesh

Phone Number:
7510329871

Address:
edathil vettil



localhost:3000/product/63f7866103d1064dd18282fa

mappini

City:
sreekandapuram

Pincode:
670631

State:
Kerala

Landmark:
Akg smaraka vayanashala

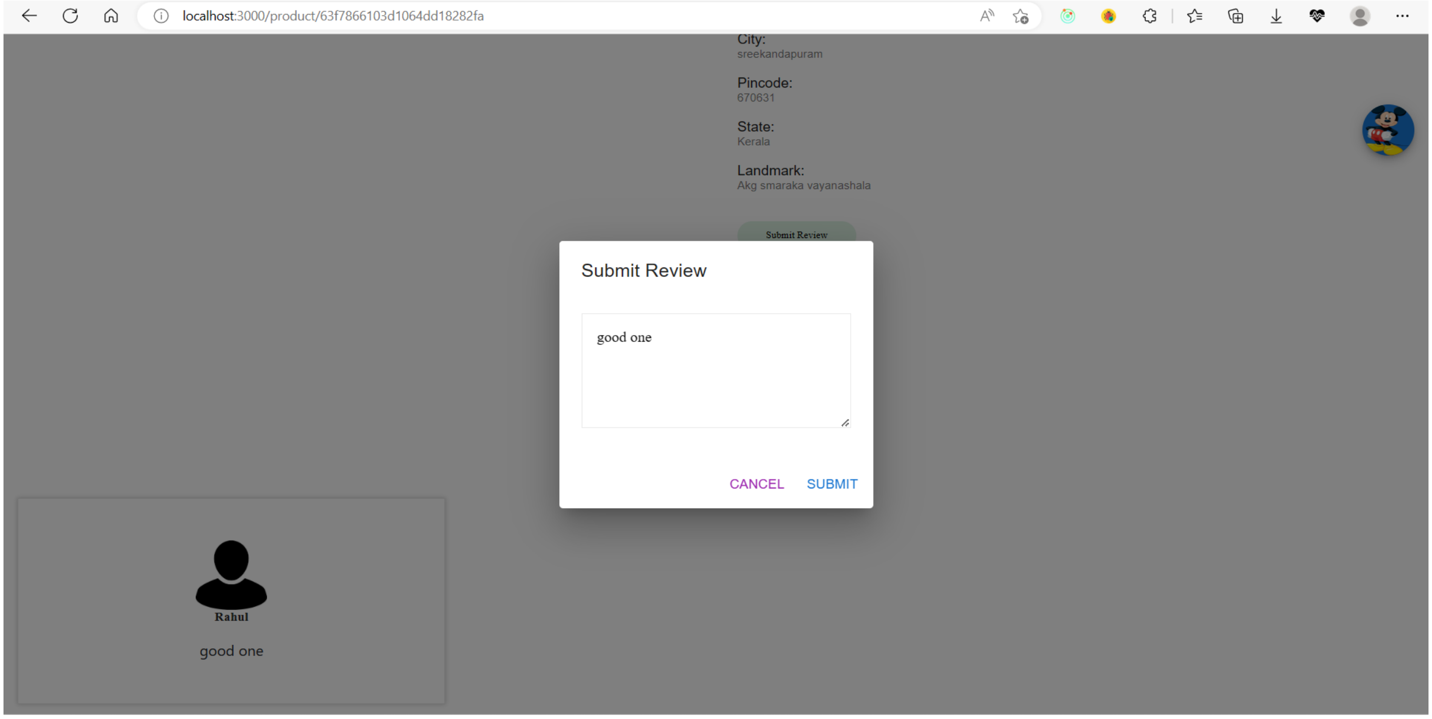
Submit Review

Reviews

Rahul

good one

Review submitting page:



localhost:3000/product/63f7866103d1064dd18282fa

City:
sreekandapuram

Pincode:
670631

State:
Kerala

Landmark:
Akg smaraka vayanashala

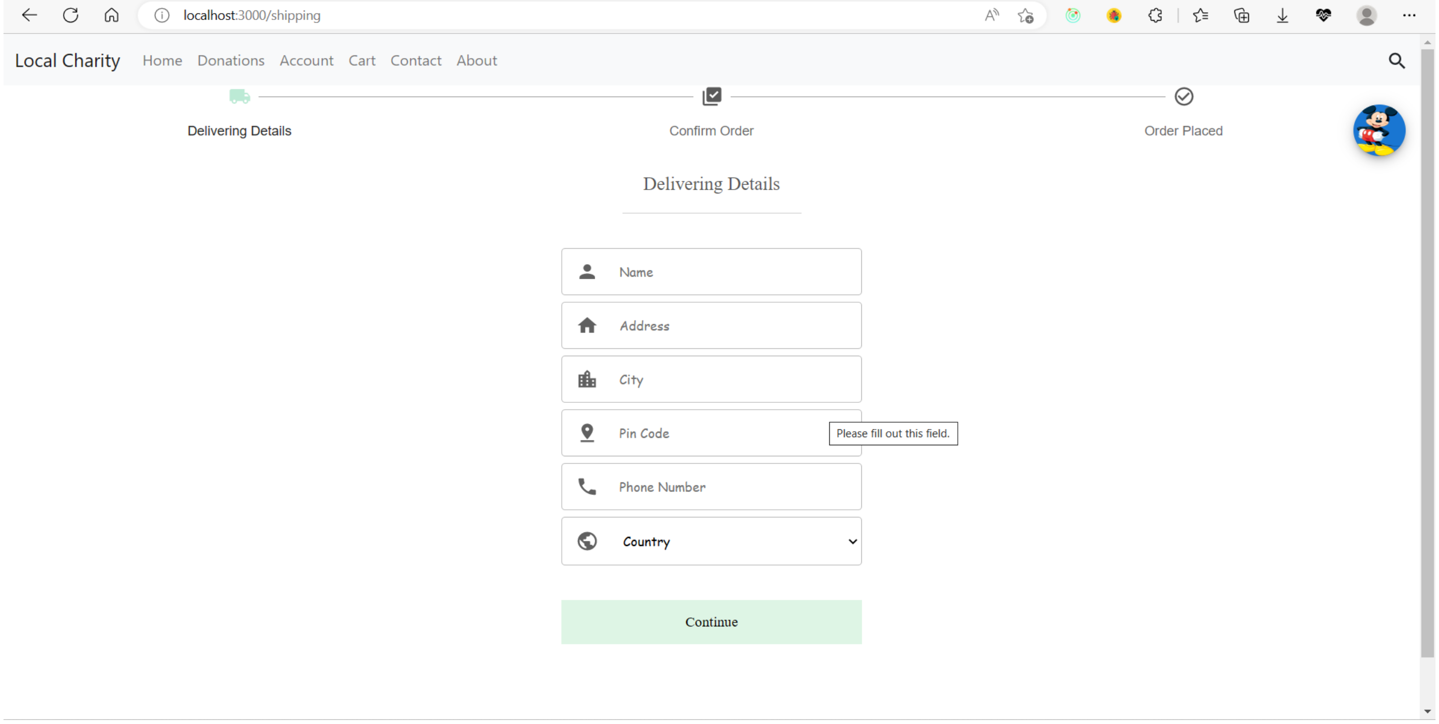
Submit Review

good one

Rahul

good one

Delivering details page:



localhost:3000/shipping

Delivering Details

Name

Address

City

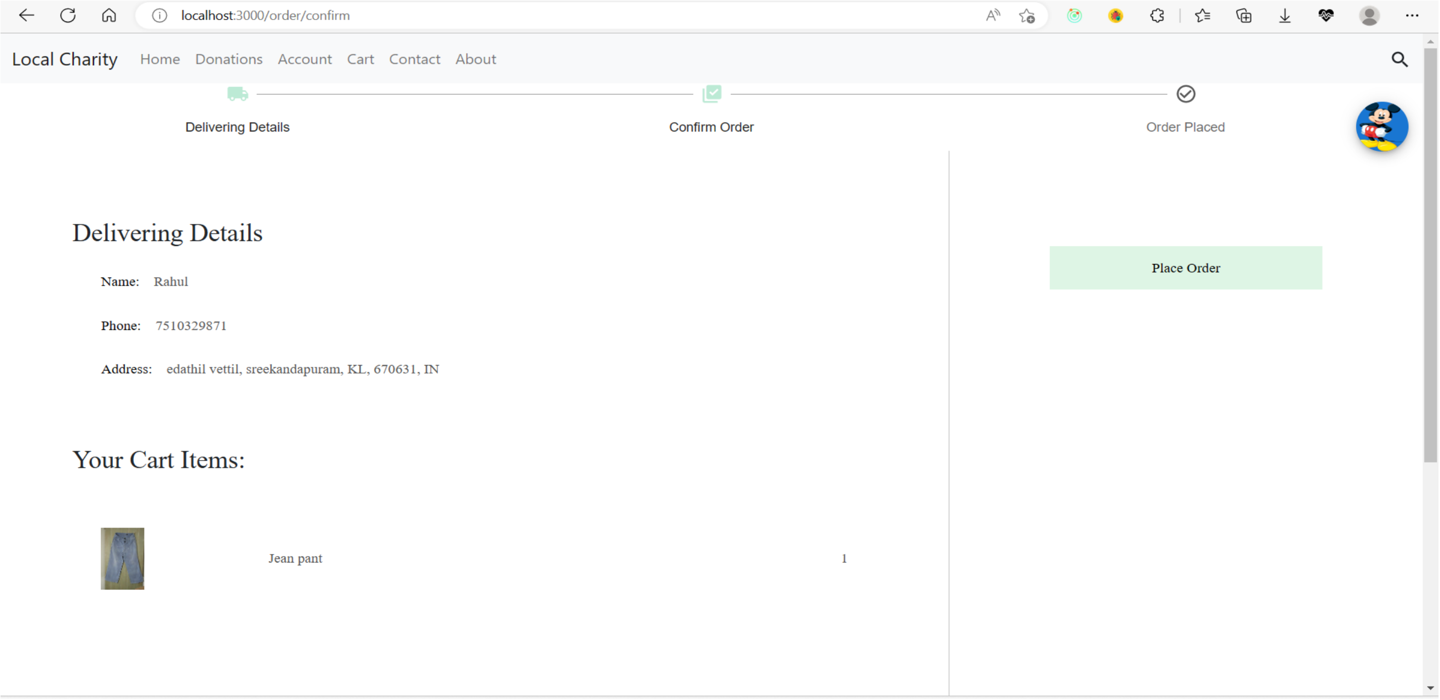
Pin Code

Please fill out this field.

Phone Number

Country

Confirm order page:



Delivering Details

Name: Rahul

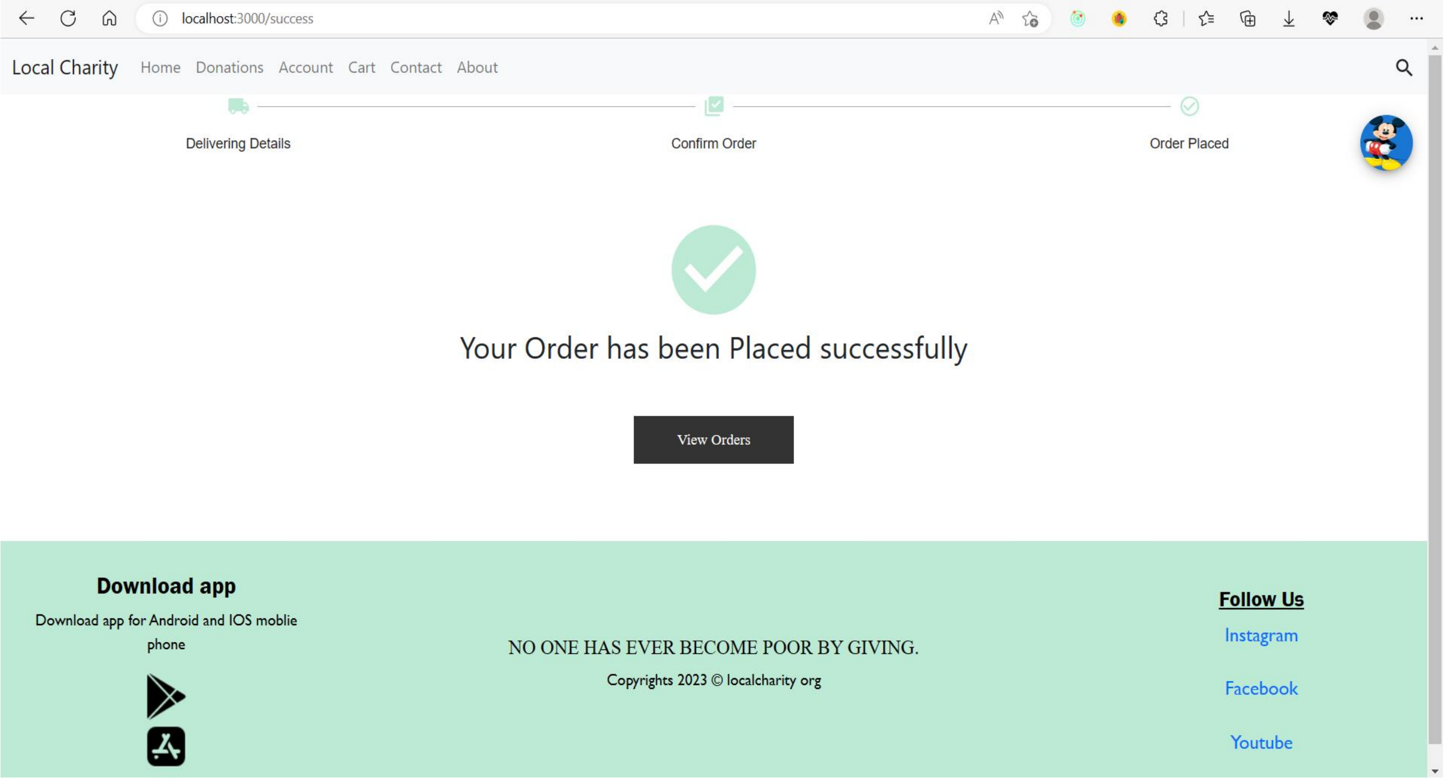
Phone: 7510329871

Address: edathil vettil, sreekandapuram, KL, 670631, IN

Your Cart Items:

Jean pant 1

Order placed page:



Your Order has been Placed successfully

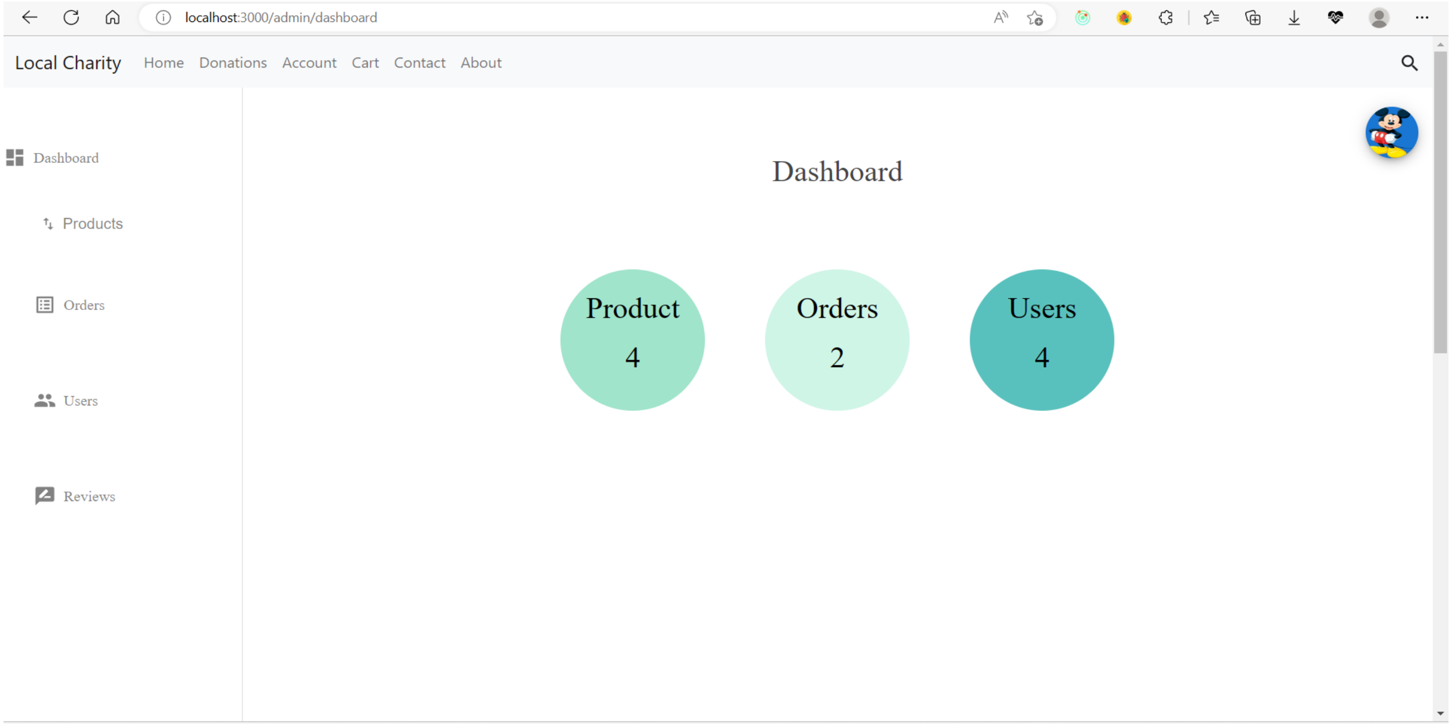
Download app

Download app for Android and IOS moblie phone

NO ONE HAS EVER BECOME POOR BY GIVING.

Copyrights 2023 © localcharity org

Dashboard page:



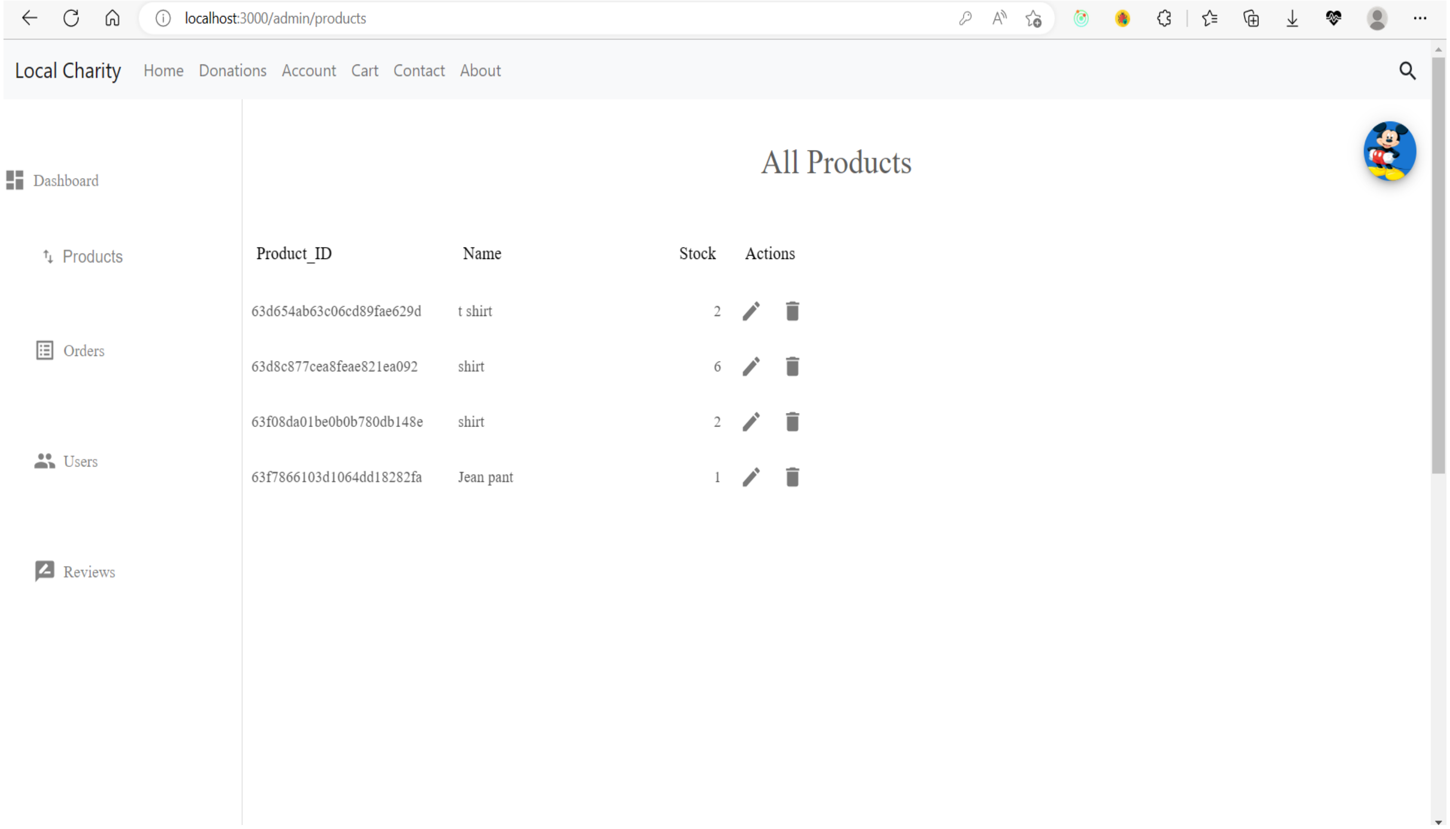
Dashboard

Product 4

Orders 2

Users 4

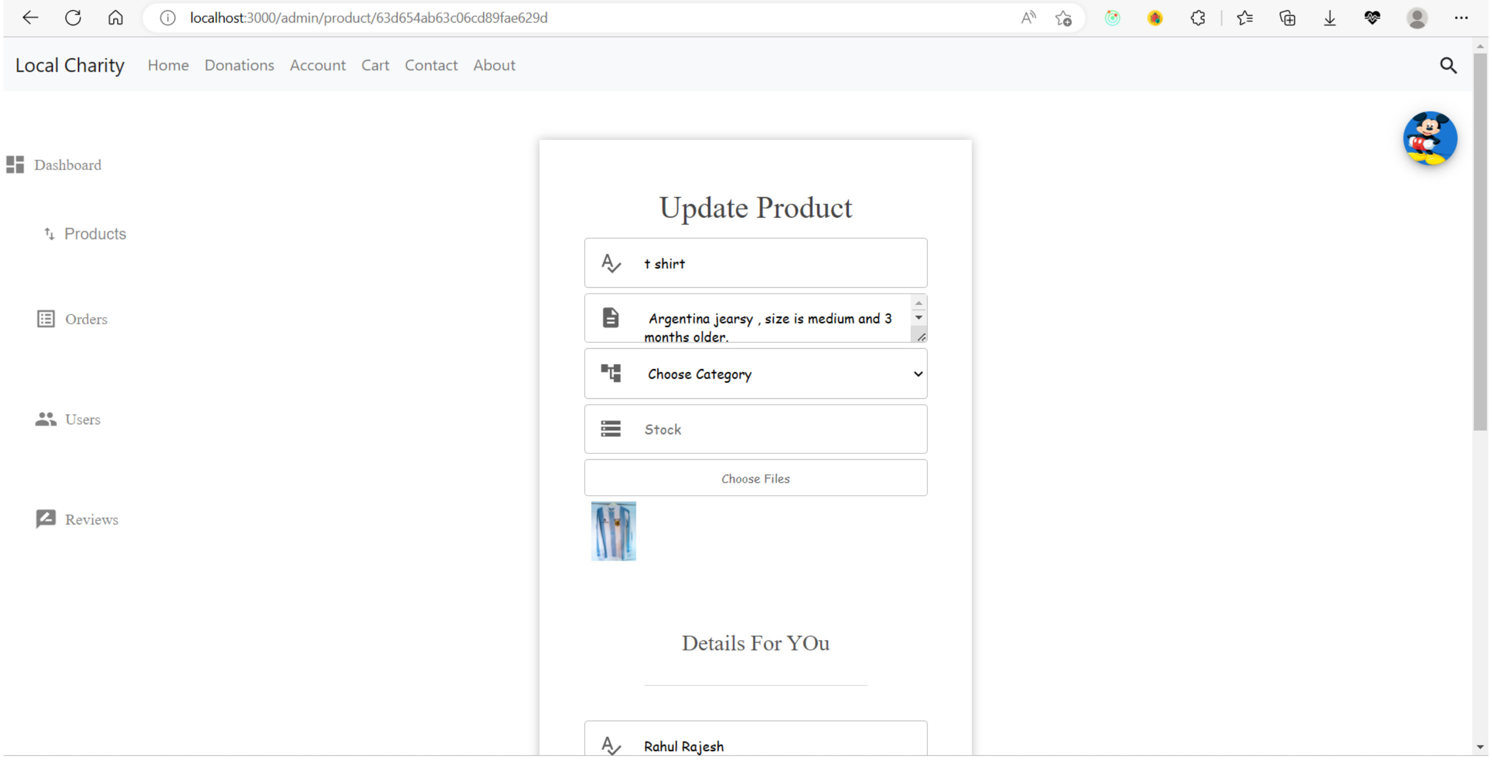
All products:



All Products

Product_ID	Name	Stock	Actions
63d654ab63c06cd89fae629d	t shirt	2	✎ 🗑
63d8c877cea8feae821ea092	shirt	6	✎ 🗑
63f08da01be0b0b780db148e	shirt	2	✎ 🗑
63f7866103d1064dd18282fa	Jean pant	1	✎ 🗑

Update product:



Update Product

t shirt

Argentina jearsy , size is medium and 3 months older.

Choose Category

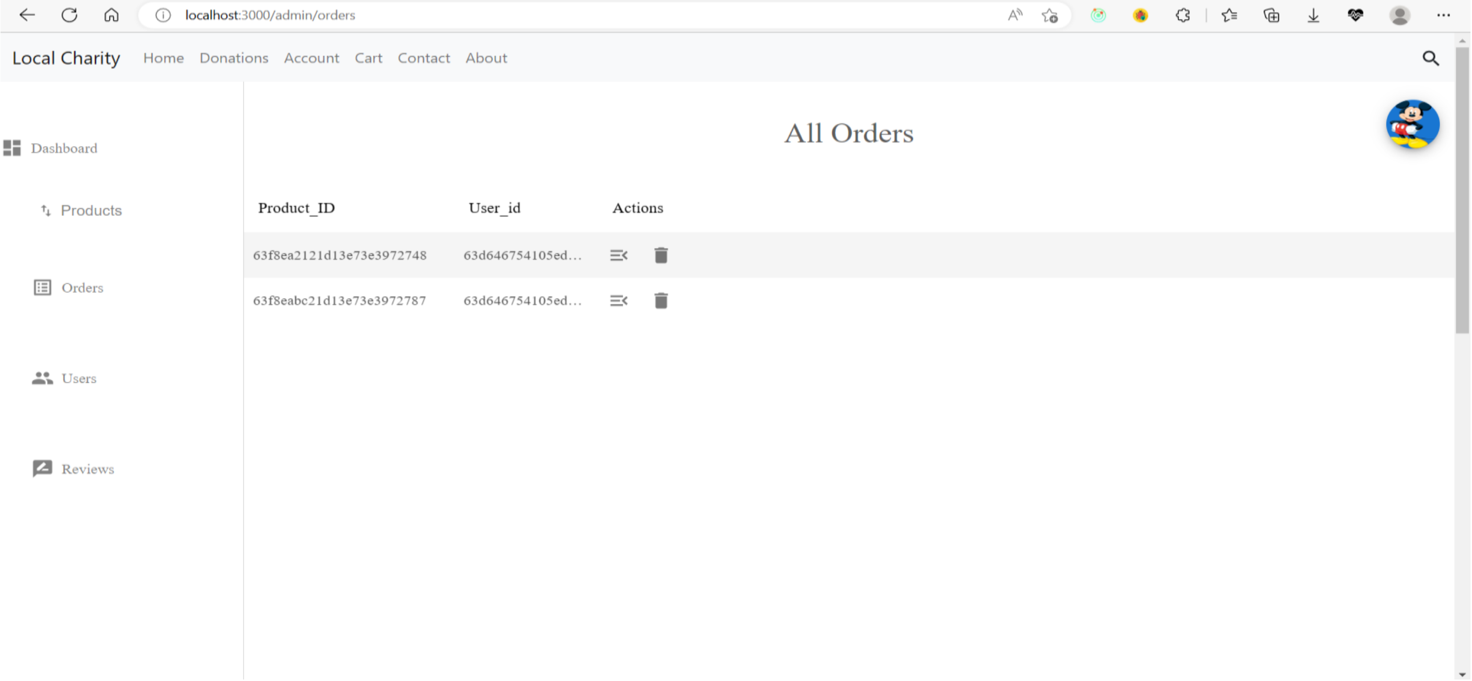
Stock

Choose Files

Details For YOu

Rahul Rajesh

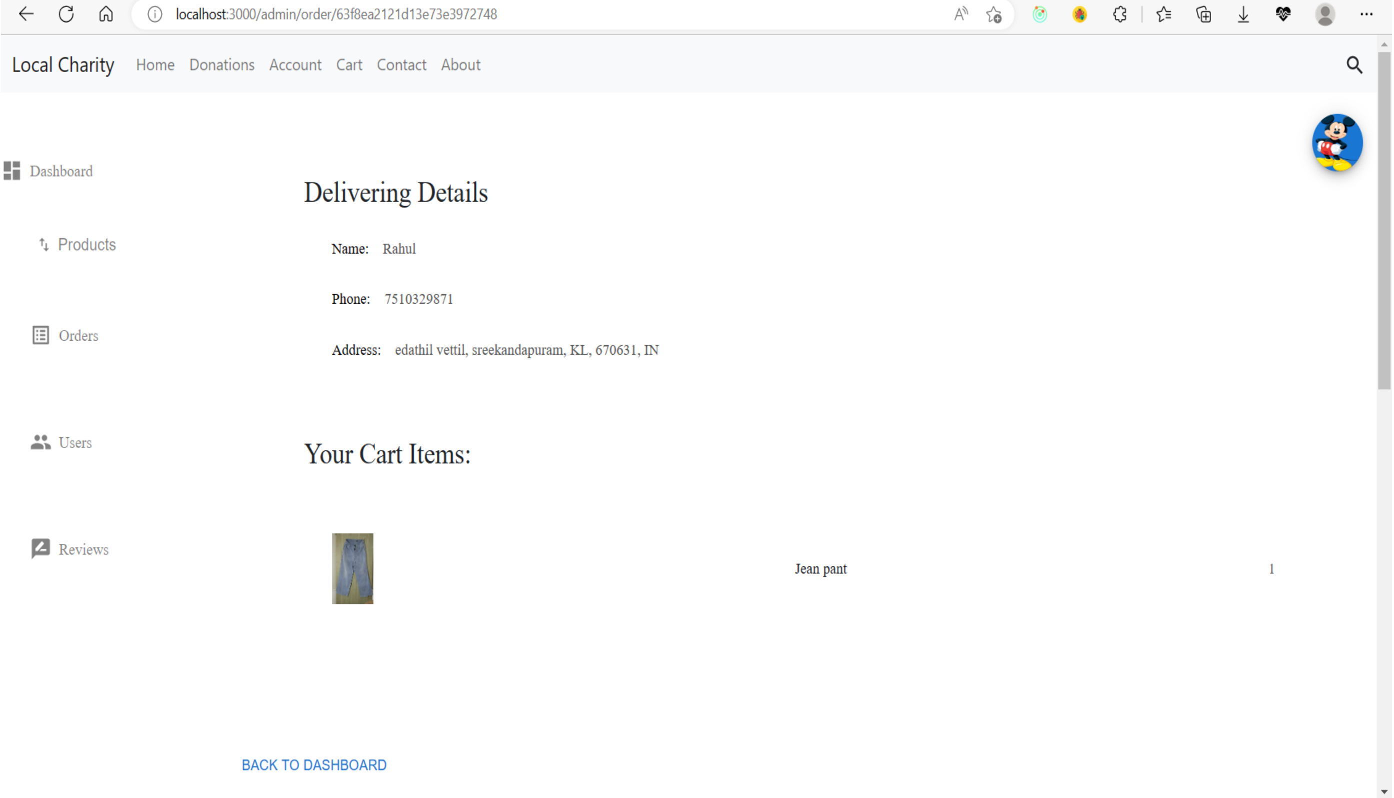
All orders page:



All Orders

Product_ID	User_id	Actions
63f8ea2121d13e73e3972748	63d646754105ed…	
63f8eabc21d13e73e3972787	63d646754105ed…	

Order details page:



Delivering Details

Name: Rahul

Phone: 7510329871

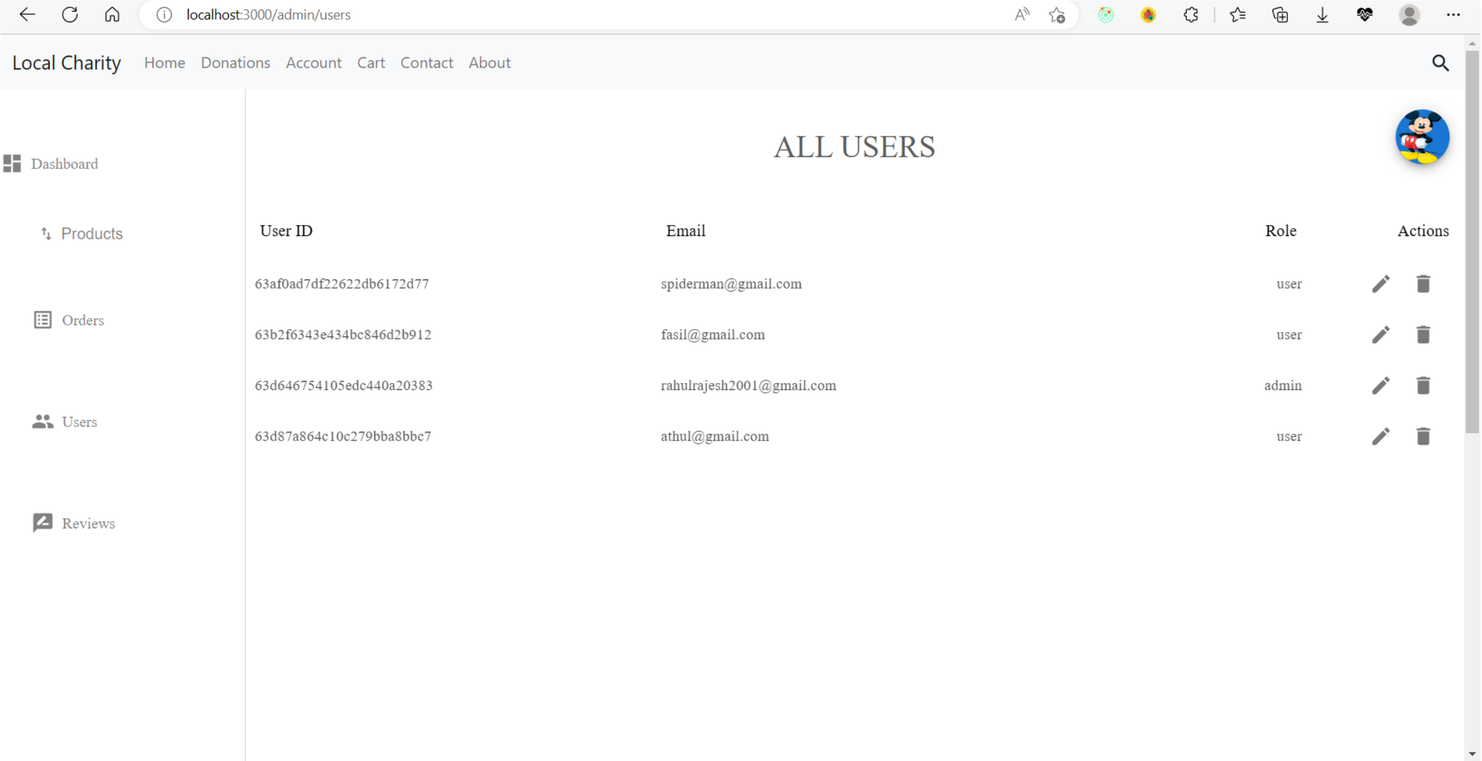
Address: edathil vettil, sreekandapuram, KL, 670631, IN

Your Cart Items:

	Jean pant	1

BACK TO DASHBOARD

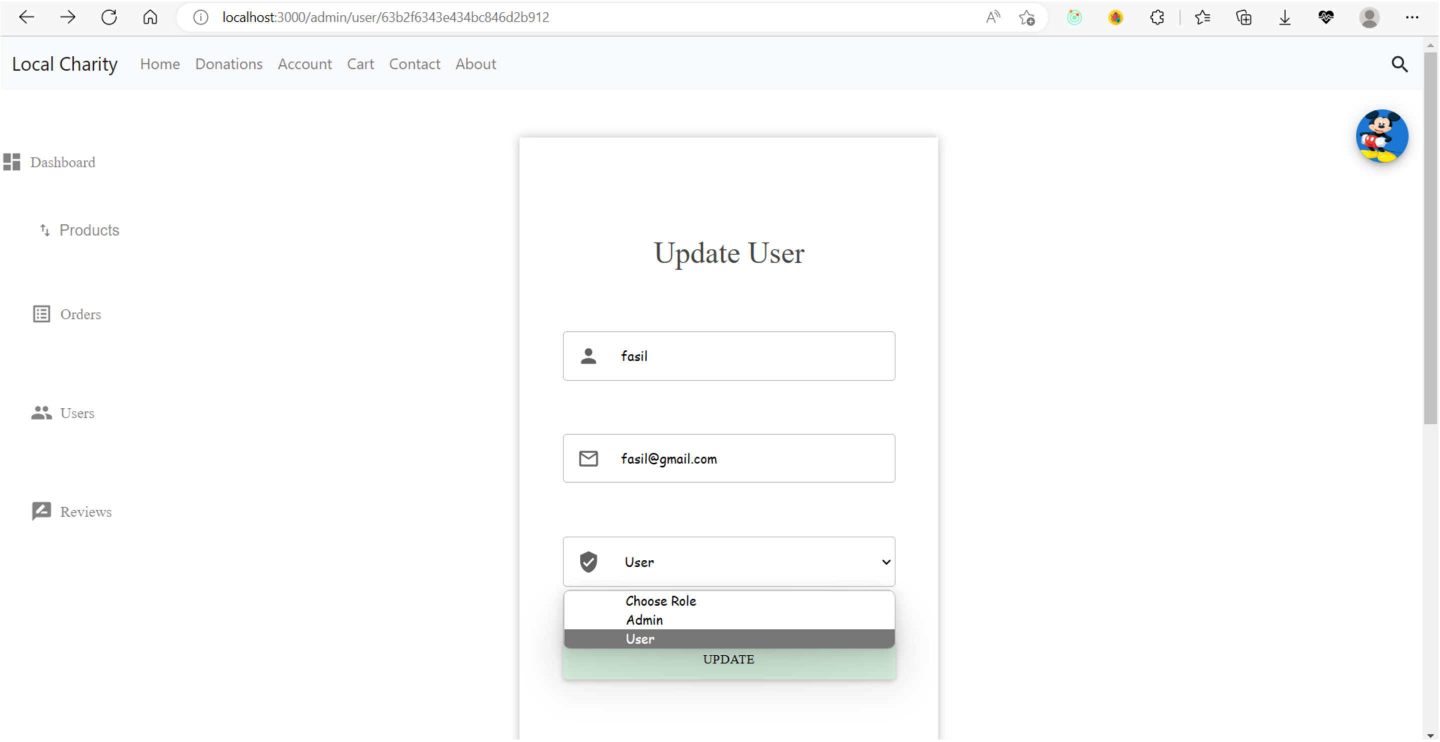
All users page:



ALL USERS

User ID	Email	Role	Actions
63af0ad7df22622db6172d77	spiderman@gmail.com	user	✎ 🗑
63b2f6343e434bc846d2b912	fasil@gmail.com	user	✎ 🗑
63d646754105edc440a20383	rahulrajesh2001@gmail.com	admin	✎ 🗑
63d87a864c10c279bba8bbc7	athul@gmail.com	user	✎ 🗑

Update user page:



localhost:3000/admin/user/63b2f6343e434bc846d2b912

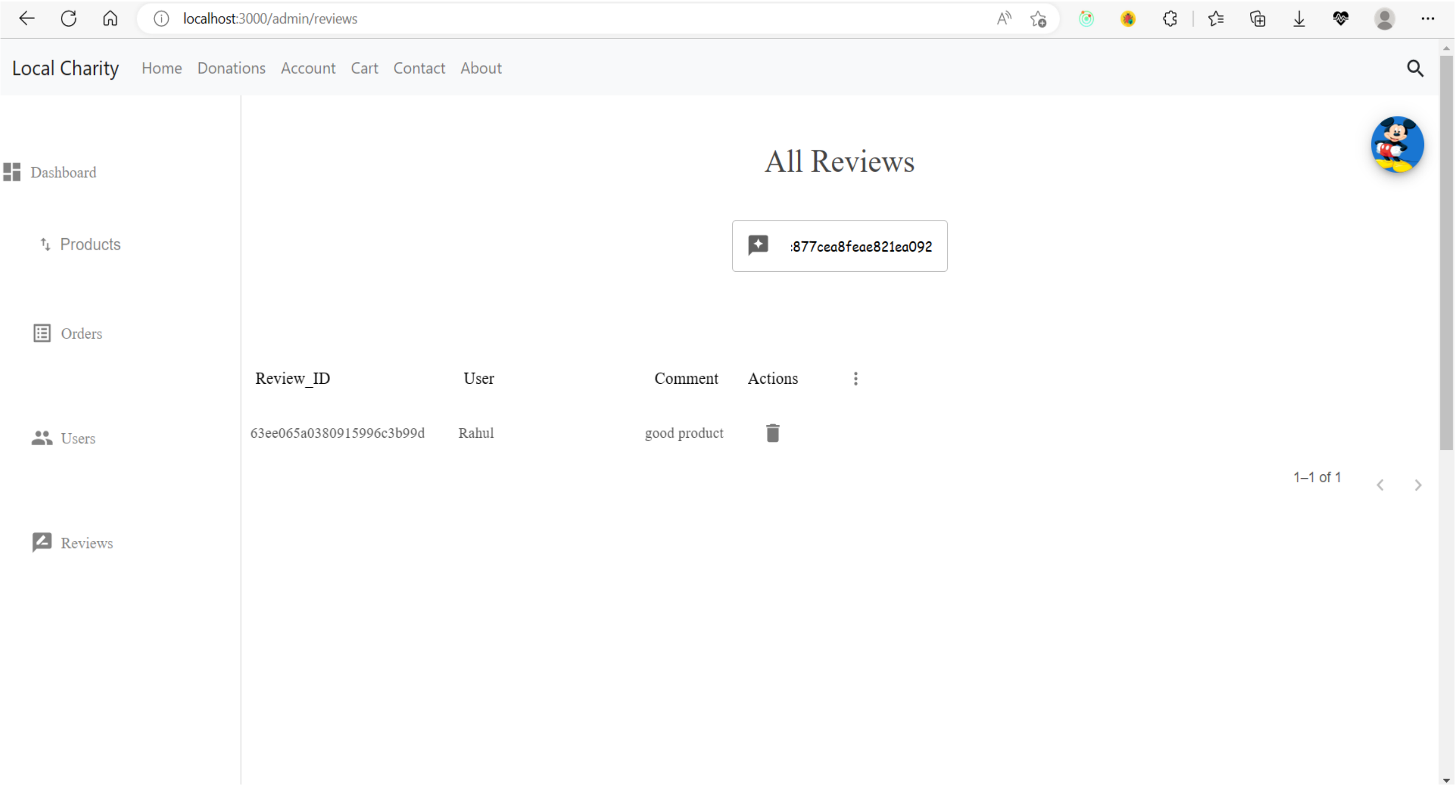
Update User

fasil

fasil@gmail.com

User

Review page:



localhost:3000/admin/reviews

All Reviews

:877cea8feae821ea092

Review_ID	User	Comment	Actions
63ee065a0380915996c3b99d	Rahul	good product	🗑

1–1 of 1