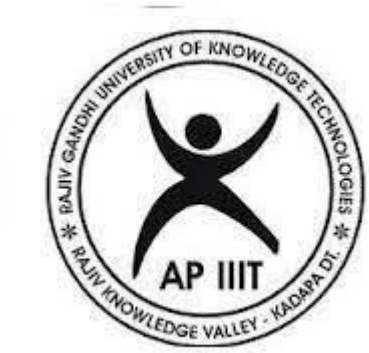
HUNGER FEEDER

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING



Rajiv Gandhi University of Knowledge Technologies
R.K.VALLEY

Submitted by
CH Hemanth – R170174
G Mogileeswar – R170177
N Harish – R170910

Under the guidance of
M Himabindu
Assistant Professor
RGUKT RK Valley.

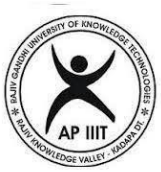
DECLARATION

We hereby declare that the report of the B.Tech Minor Project Work entitled "**HUNGER FEEDER** " which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university/institution for award of any degree.

Date:22-09-2022
Place:RK Valley

CH. Hemanth -- R170174
G. Mogileeswar -- R170177
N. Harish -- R170910

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES



CERTIFICATE FOR PROJECT COMPLETION

This is certify that the project entitled "**HUNGER FEEDER**" submitted by CH .HEMANTH(R170174), G.MOGILEESWAR (R170177), N.HARISH (R170910), under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering-3 during the academic semester-2 2021-2022 at RGUKT, RK VALLEY. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

Project Internal Guide
M Himabindu
Assistant Professor
RGUKT, RK Valley

Head of the Department
P Harinadha
HOD Of CSE
RGUKT, RK Valley

Submitted for the practical examination held on .....…

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and who's constant guidance and encouragement crown all the efforts success. We would like to express my sincere gratitude to M Himabindu ,our project guide for valuable suggestions and keen interest throughout the progress of our project. We are grateful to Mr.Harinadha, HOD CSE, for providing excellent computing facilities and congenial atmosphere for progressing our project. At the outset,We would like to thank Rajiv Gandhi University of Knowledge Technologies(RGUKT),for providing all the necessary resources and support for the successful completion of our course work.

Index

Abstract

A web application which is useful for management of leftover food collection and donations from the College Messes,Faculty and Outside Donors. The leftover food is then donated to the needy . Hunger feeder was designing interventions to reduce hunger among underserved communities in Local.

The system consists of five types of users admin, mess managers,faculty,donors and volunteers. Admins control all the activities and accept/reject donations and select agents. Volunteers are responsible for collecting food from homes of food donors. Donors are the driving users of the application who donate food. Mess Managers are driving users of the application who donate leftover food from their messes. Every user also have a dashboard where they can view several things in short summary. The application provides signup, login and logout functionalities.

Hunger Feeder SRS Document

Introduction:

This document has the requirements of Food donation management .The Hunger feeder is used to manage the donations from the donors .

1.1:Purpose

The purpose of this document is to gather the requirements that are needed for implementing the Hunger Feeder. It also focuses on various key features, the product, product vision and scope,product overview.

1.2:Intended Audience:

The intended audience will be the donors and the volunteers who can access the platform and give donations and the volunteers can collect the food.
Users:
1. Mess managers
2. Faculty
3.Volunteer Students

Product Vision:

Hunger feeder was designing interventions to reduce hunger among underserved communities in Local. Our mission is to feed all the needy people and no one feels that he is going to sleep hungry today.

Technologies:

➢ Node js

➢ Bootstrap

➢ MongoDB

➢ EJS

➢ ExpressJS

Node JS:

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Bootstrap:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

MongoDB:

MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and function which is the equivalent of relational database tables.
Collections → Table
Documents → Rows

EJS:

EJS (Embedded JavaScript Templating) is one of the most popular template engines for JavaScript. As the name suggests, it **lets us embed JavaScript code in a template language that is then used to generate HTML**

ExpressJS:

Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application.
It's a layer built on the top of the Node js that helps manage servers and routes.

Express was created to make APIs and web applications with ease,

• It saves a lot of coding time almost by half and still makes web and mobile applications are efficient.

• Another reason for using express is that it is written in javascript as javascript is an easy language even if you don't have a previous knowledge of any language. Express lets so many new developers enter the field of web development.

The reason behind creating an express framework for node js is:

- Time-efficient
- Fast
- Economical
- Easy to learn
- Asynchronous

SOFTWARE CONFIGURATIONS :

Node.js v16.17.0

Ubuntu 18.04 LTS

BootStrap 5.0

IDE : Visual Studio Code Editor version 1.71

DESIGN

The design phase of software development deals with transforming the requirements as described in the SRS documents into a form implementable using a programming language. The software design process can be divided into the following three levels of phases of design:

1.Interface Design

2.Architectural Design

3.Detailed Design

System-wide Requirements:

Actors:

The system interacts with Five kinds of users. Each user has its own functions to access with the system. The functionalities of users are dependent on each other.

Events:

Hunger Feeder Platform is a multi-user system which provides activities associated with its day to day operations.

Modules

- **Admin**
- **Volunteer**
- **Donors**
  - **Mess Manager**
  - **Faculty**
  - **Outside Donor**

Admin Module

Admin is the super user of the website who can manage everything on the website. Admin can log in through the login page.

◆ Dashboard:In this section, the admin can see all detail in brief like the total donations, assigned and the Volunteers/Agents donations collected .

◆ Pending Donations :All the Pending Donations that are not collected yet and shown here.

◆ Previous Donations: All collected Donations are shown here.

◆Volunteers :Manage the Volunteers details.

Volunteer Module:

Volunteer is the person who collects the Provided Donations.

◆ Dashboard:In this section, the admin can volunteer all detail in brief like the total assigned donations, Collected Donations.

◆ Pending Donations :All the Pending Donations that are not collected yet and shown here .

◆ Previous Donations: All collected Donations are shown here.

◆Profile :Manage the Profile details and can update them .

Donor Module:

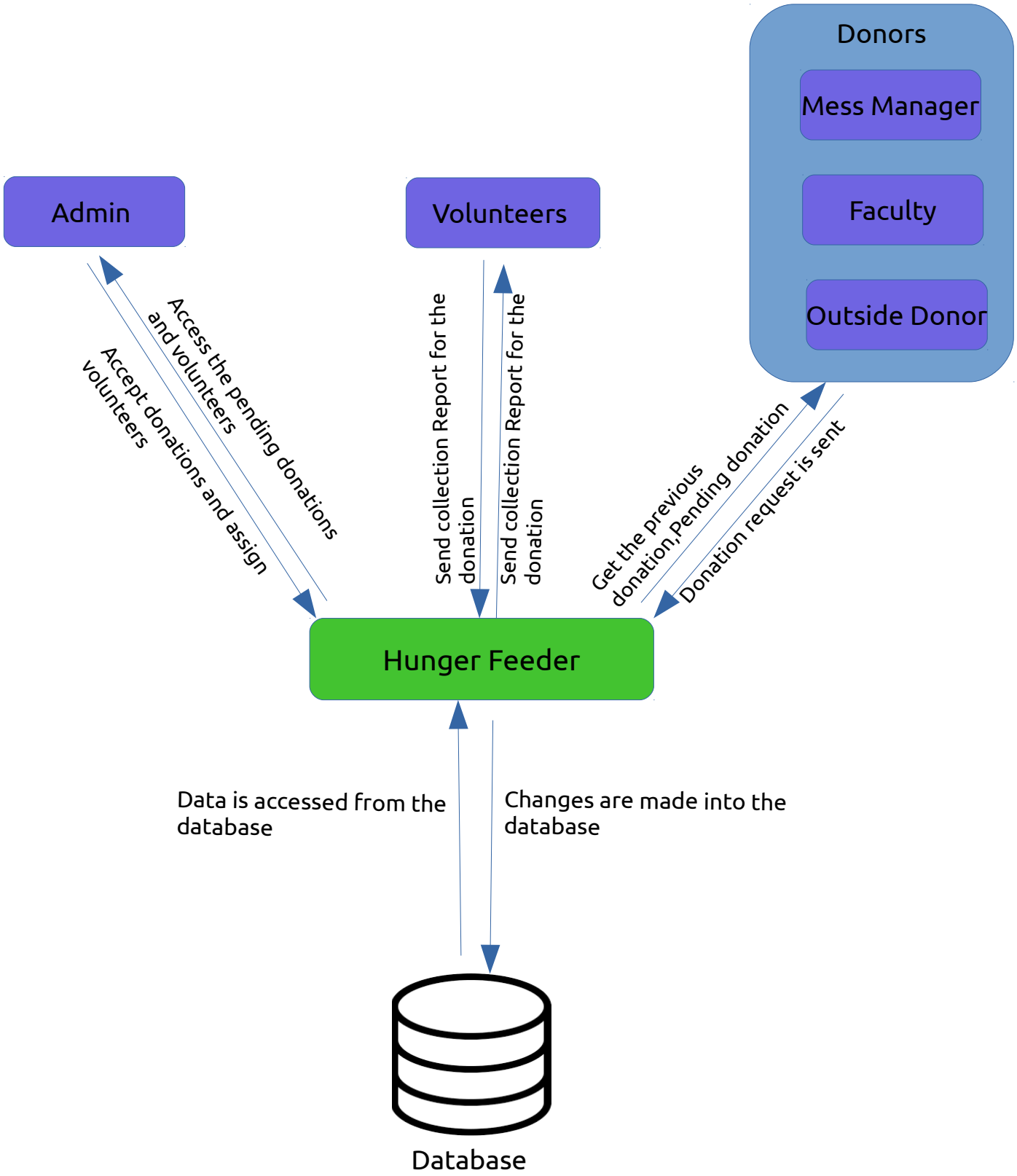
◆ Dashboard:In this section, the Donor can view all detail in brief like the total Previous Donations, Collected Donations.

◆ Pending Donations :All the Pending Donations that are not collected yet and shown here .

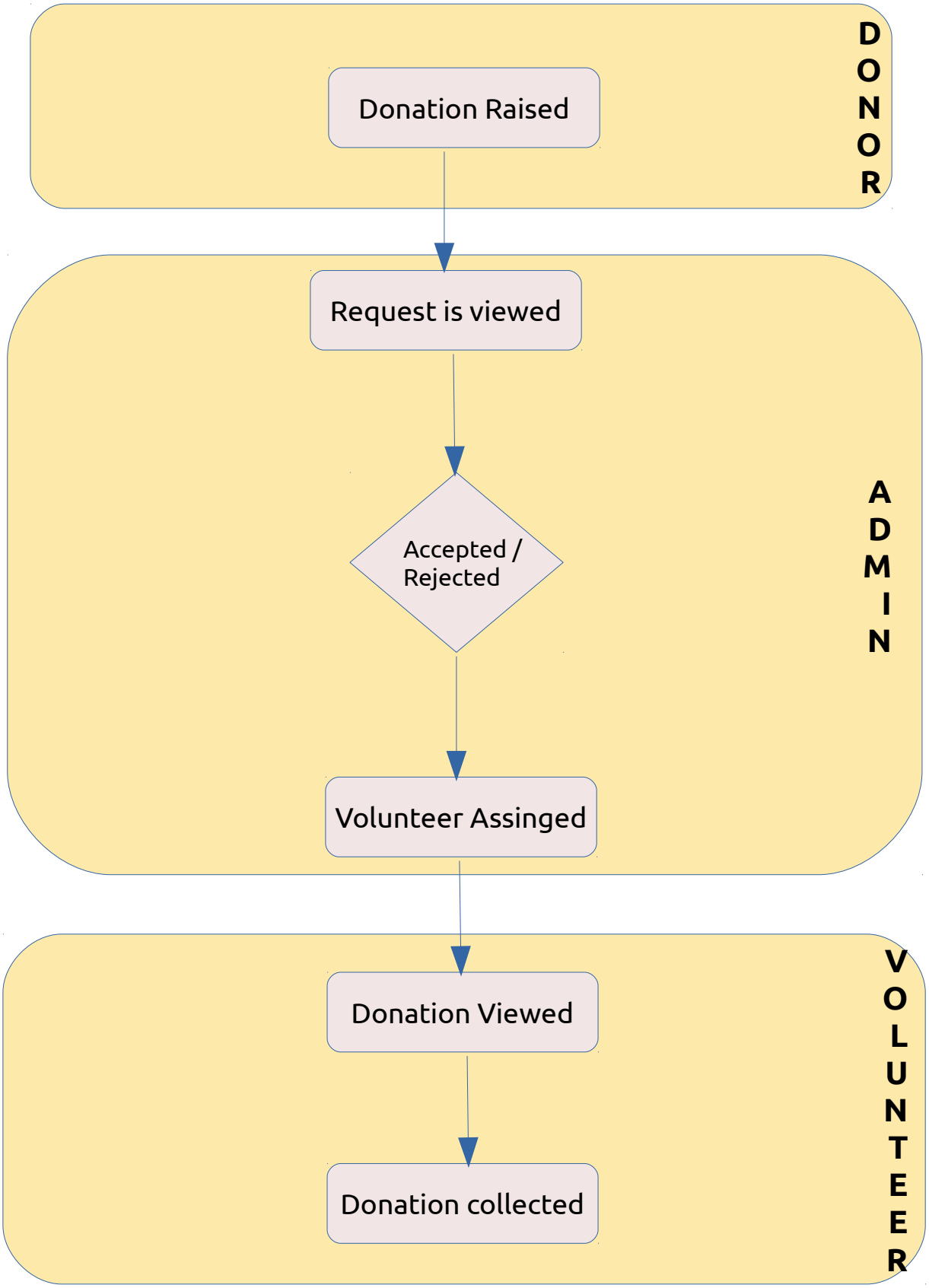
◆ Previous Donations: All collected Donations are shown here.

◆Profile :Manage the Profile details and can update them.

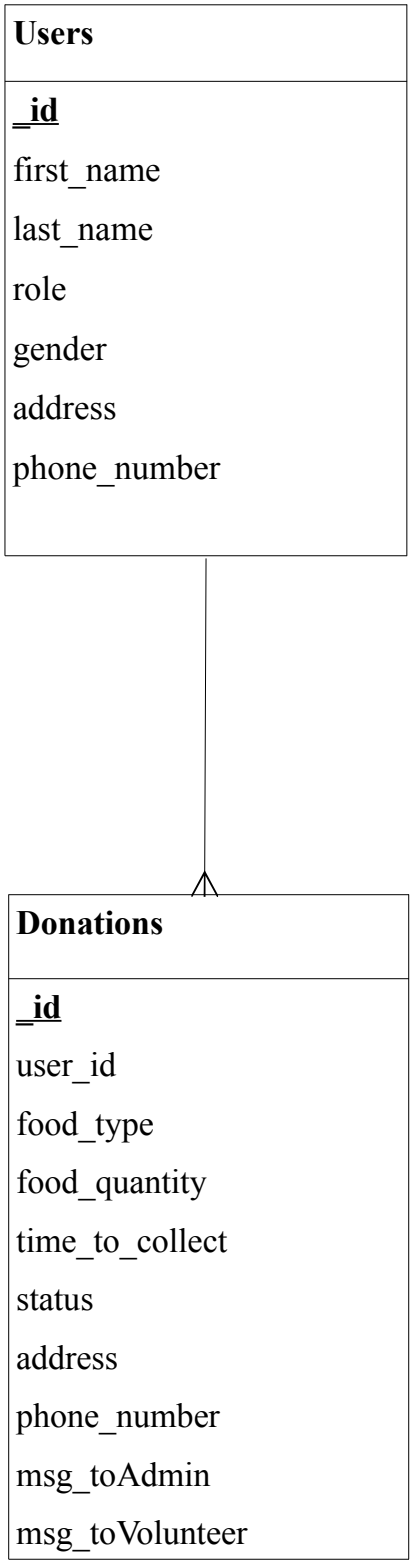
CONTEXT DIAGRAM:



DONATION FLOW DIAGRAM:



ER Diagrams:


Users
_id
first_name
last_name
role
gender
address
phone_number
Donations
_id
user_id
food_type
food_quantity
time_to_collect
status
address
phone_number
msg_toAdmin
msg_toVolunteer

Coding

app.js:

```
const express = require("express");
const app = express();
const passport = require("passport");
const flash = require("connect-flash");
const session = require("express-session");
const expressLayouts = require("express-ejs-layouts");
const methodOverride = require("method-override");
const homeRoutes = require("./routes/home.js");
const authRoutes = require("./routes/auth.js");
const adminRoutes = require("./routes/admin.js");
const donorRoutes = require("./routes/donor.js");
const volunteerRoutes = require("./routes/volunteer.js");
const messManagerRoutes = require("./routes/messmanager")
const facultyRoutes = require("./routes/faculty")
require("dotenv").config();
require("./config/dbConnection.js")();
require("./config/passport.js")(passport);


app.set("view engine", "ejs");
app.use(expressLayouts);
app.use("/assets", express.static(__dirname + "/assets"));
app.use(express.urlencoded({ extended: true }));
app.use(express.json());
app.use(session({
secret: "secret",
resave: true,
saveUninitialized: true

}));
app.use(passport.initialize());
app.use(passport.session());
app.use(flash());
app.use(methodOverride("_method"));
app.use((req, res, next) => {
res.locals.currentUser = req.user;
res.locals.error = req.flash("error");
```

```
res.locals.success = req.flash("success");
res.locals.warning = req.flash("warning");
next();
});



// Routes
app.use(homeRoutes);
app.use(authRoutes);
app.use(donorRoutes);
app.use(adminRoutes);
app.use(volunteerRoutes);
app.use(messManagerRoutes);
app.use(facultyRoutes)
app.use((req,res) => {
res.status(404).render("404page", { title: "Page not found" });
});


const port = process.env.PORT || 5000;
app.listen(port, console.log(`Server is running at http://localhost:${port}`));
```

Installation

1. Open the code in your code editor.
2. To install all the dependencies (listed in package.json file) in your project, go to terminal and type the following command and hit enter:

   npm install
3. Create a file named ".env" and enter the following credentials:

   MONGO_URI=your-mongo-uri
4. Go to terminal and type the following command and hit enter:

   npm run dev
5. Open browser and go to url
6. You need to first signup and then login to run the application.

Dependencies:

```
"author": "mogileeswar",
"license": "ISC",
"dependencies": {
  "bcryptjs": "^2.4.3",
  "connect-flash": "^0.1.1",
  "dotenv": "^10.0.0",
  "ejs": "^3.1.6",
  "env-cmd": "^10.1.0",
  "express": "^4.17.2",
  "express-ejs-layouts": "^2.5.1",
  "express-session": "^1.17.2",
  "method-override": "^3.0.0",
  "mongoose": "^6.1.2",
  "passport": "^0.5.2",
  "passport-local": "^1.0.0"
},
"devDependencies": {
  "nodemon": "^2.0.19"
}
```

Scripts:

```
▷ Debug
"scripts": {
  "start": "node app.js",
  "dev": "env-cmd -f ./config/dev.env nodemon src/index.js"
},
```

TESTING

Here we performed two types of testing to the software for finding bugs

1.Functional Testing:

we tested main features like testing each and every module like login,signup,Donate,View Previous Donations,View Pending Donations,Profile and status.

a)Integration Testing:
Here,the data flow is tested.For example,if we take login module by entering valid credentials it redirects to the respected user's Dashboard.It is Done to the all created modules.

b)System Testing:
Here,the end to end Testing is done on application from entering credentials, navigating to the all modules such as Donations,Details of the Donor,Status etc. and atlast to the logout page.

2.Non Functional Testing:

Here we tested the Non-functional features like Compatability,Performance and Adhoc Testing

1.Compatability Testing :- Here We tested this software on Various Operating System Such as Linux,windows etc…

2.Performance Testing :- Here we tested the speed,efficiency.The software is given accurate results when the user enters the data

3.Adhoc Testing :- Basically,Adhoc Testing means Testing the software Randomly because every user enters the data differently. So We perform Random testing to our software in order to achieve users perspective.

Advantages:

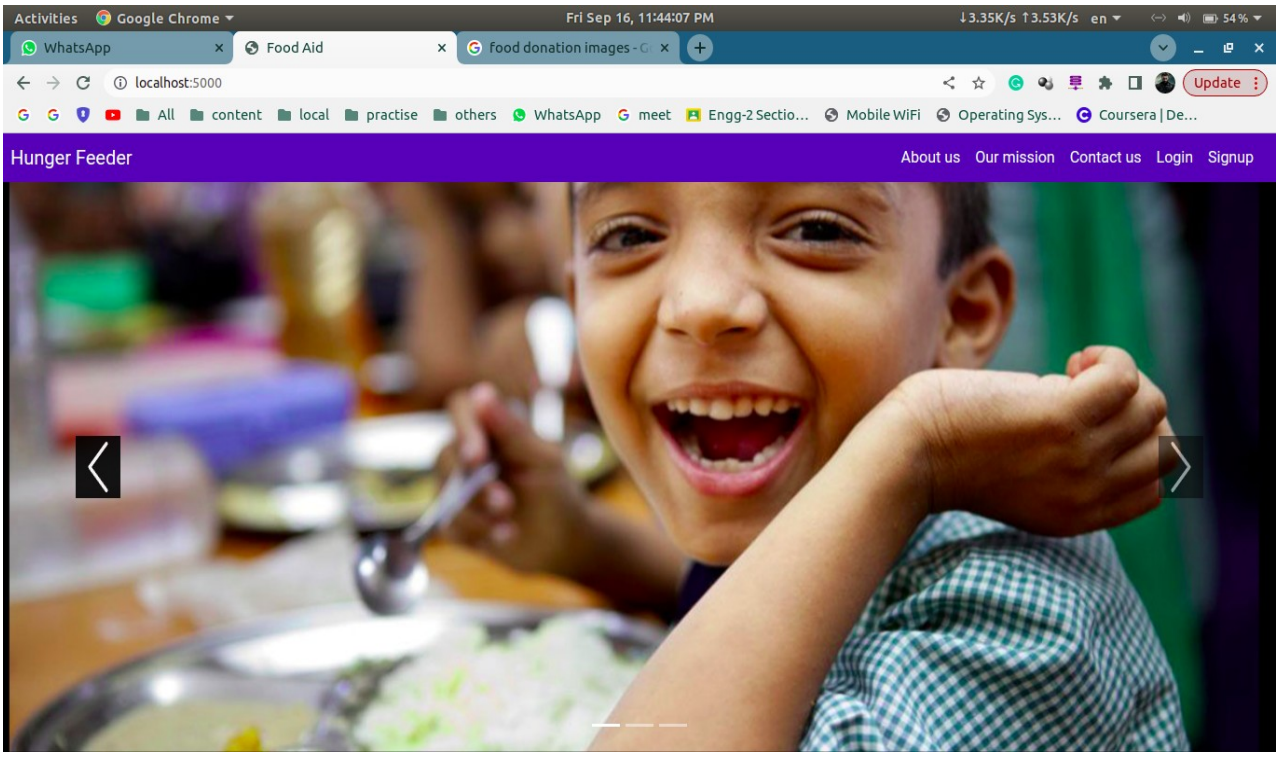
- This software mainly designed for the people who are actually concern about Donating food to the Poor ones who are actually Thriving for Hunger
- It will reduce the poverty upto some extent
- Donating to the Needy is a Great Way to improve Conditions in our Community
- For many donors, donating is a platform for knowing about the issues surrounding that particular need. This Software encourages the donors in an ease

Future Improvements:

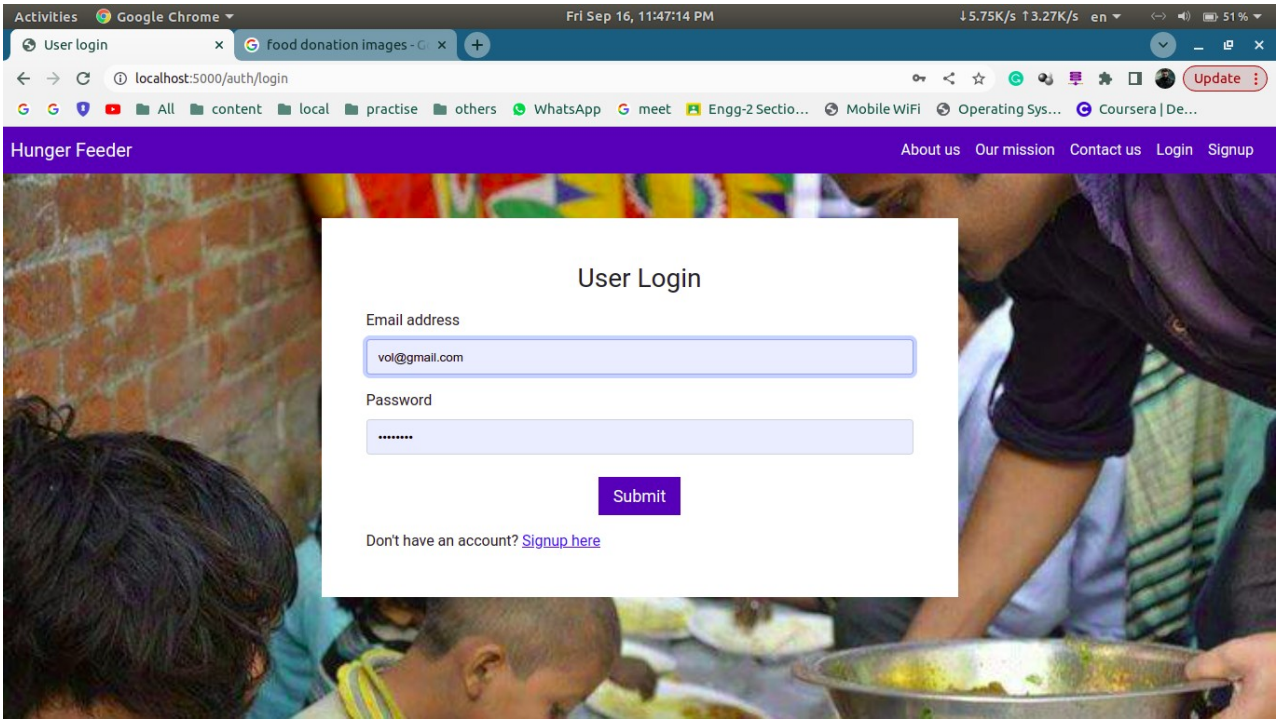
1. Automating the assigning task and eliminate the burden on the admin.
2. An unavailable tag for the volunteer so that no Donation is assigned to that volunteer.
3. Update donor where their food is donated.
4. Perks for the active donors

Snippets

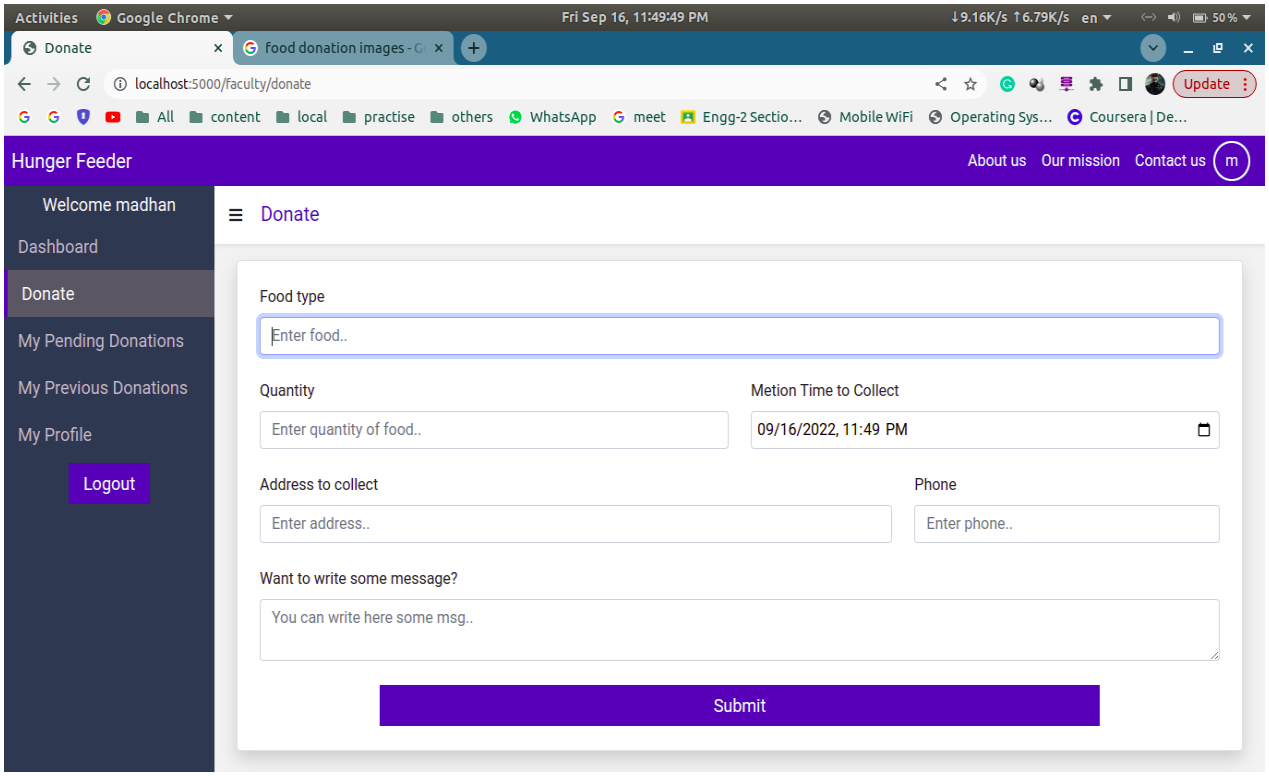
Home pages:



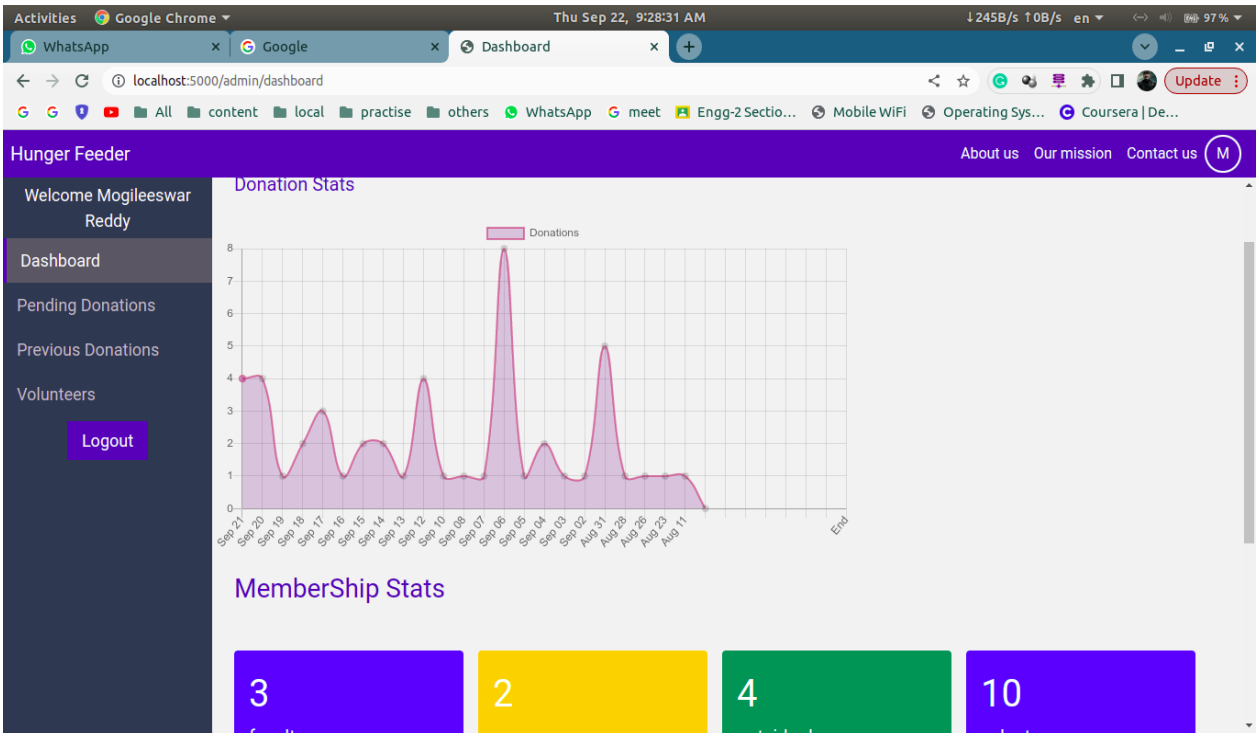
Login page:



Donate Page:

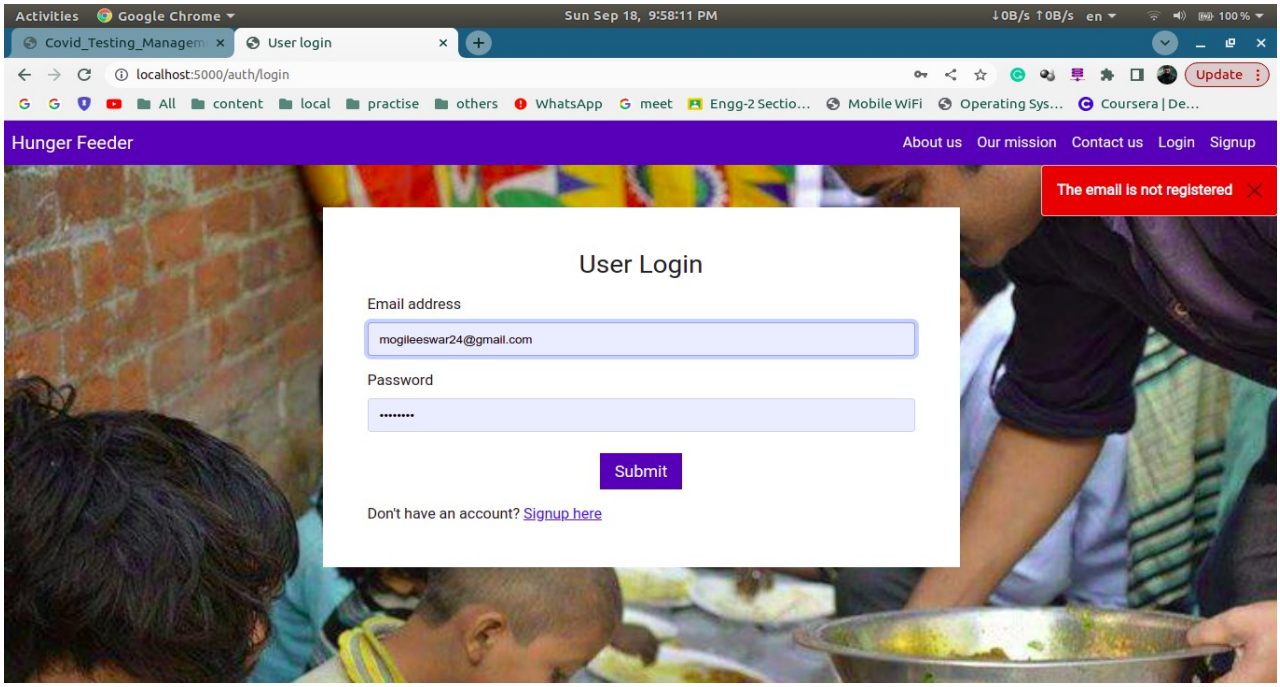


Admin Dashboard:

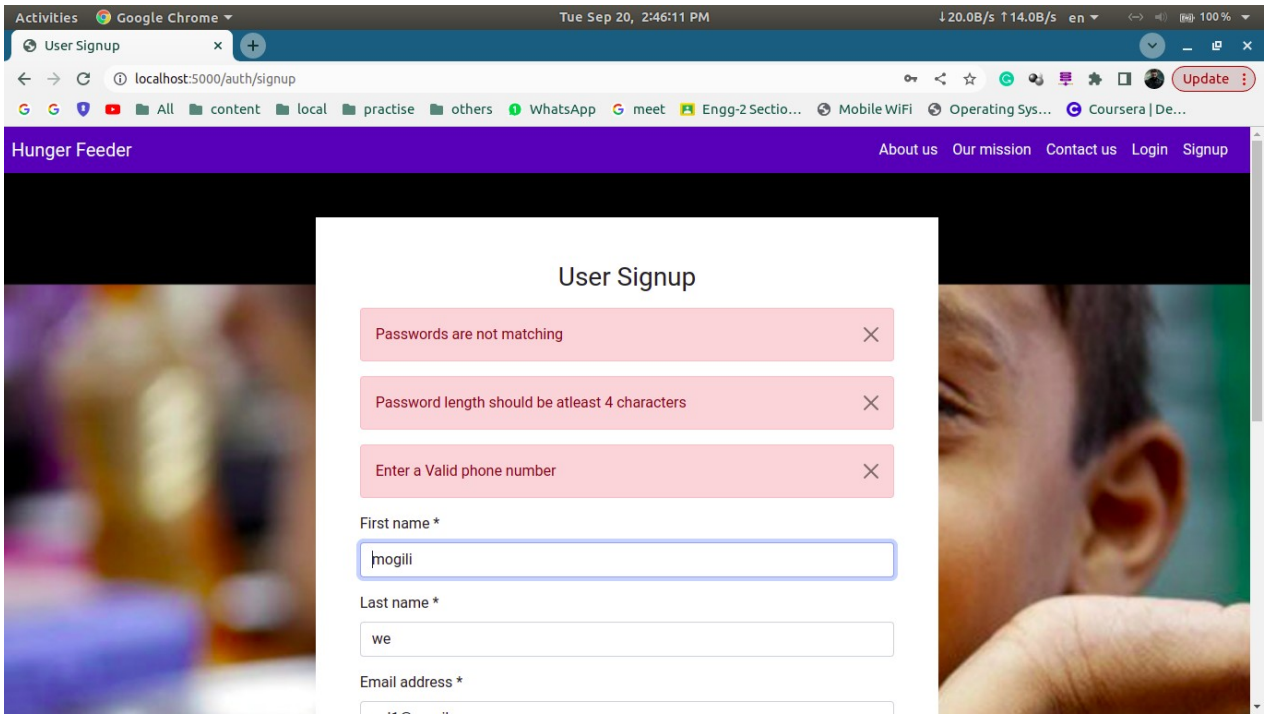


Testing Site:

Not Registered member:



Form Validation:



Conclusion:

We would like to conclude that our project shall aim at helping the needy by connecting them with the donors by using the NGOs as an intermediary who shall do their job aided by the application that we shall provide them. Our application shall aim to mitigate issues like lack of awareness among donors, lack of transparency in the donation process and thus acts as a bridge between the people in need. The surplus food from the mess and gatherings can be donated easily. The visualization impact of the donation can create a positive impact on the users. Minimizing food wastage and feeding the hunger is the main goal of the Hunger Feeder. The application is targeted in two ways, the user who is collecting the food and the person that is claiming the food.

References:

https://www.youtube.com/watch?v=-RCnNyD0L-s&t=1274s Login System

https://nodejs.org/en/docs/ NodeJS docs

https://getbootstrap.com/docs/4.1/ Bootstrap Docs

https://www.mongodb.com/docs/ MongoDB Docs

https://heynode.com/tutorial/authenticate-users-node-expressjs-and-passportjs/