E-Waste, Plastic Waste Collection System [EPWS]

A PROJECT REPORT

Submitted by

Rahul A B (SHR21CS104)
Theres T L (SHR21CS123)
Joel Jaison (SHR21CS079)
Hanna De Grace (SHR21CS070)

To

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technologyin
Computer Science and Engineering

Department of Computer Science and Engineering
SAHRDAYA COLLEGE OF ENGINEERING AND TECHNOLOGY
KODAKARA, THRISSUR - 680684

JANUARY 2023

CONTENTS

DECLARATION

We, undersigned, hereby declare that the project report "E-Waste, Plastic Waste Collection System [EPWS]", submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under the guidance of Ms. Deepa Devassy, Department of Computer Science Engineering. This submission represents our ideas in our own words and where ideas or words of others have been included; We have adequately and accurately cited and referenced the original sources. We also declare that we haveadhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or fromwhom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma, or similar title of any other University.

Rahul A B
Theres T L
Joel Jaison
Hanna De Grace

Kodakara
04-01-2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SAHRDAYA COLLEGE OF ENGINEERING AND TECHNOLOGY, KODAKARA, THRISSUR

BONAFIDE CERTIFICATE

This is the certify that the project report entitled "**E-Waste, Plastic Waste Collection System [EPWS]**" submitted by **Rahul A B (SHR21CS104), Theres T L (SHR21CS123), Joel Jaison (SHR21CS079), and Hanna De Grace (SHR21CS070)** to APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by him under our guidanceand supervision. This report in any form has not been submitted to any other University or Institute for any other purpose.

GUIDE

Ms. Deepa Devassy
Assistant Professor

PROJECT COORDINATOR

Mr. Wilson Joseph C
Assistant Professor

HEAD OF THE DEPARTMENT

Dr. R Satheesh Kumar
Professor

Kodakara
04-01-2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SAHRDAYA COLLEGE OF ENGINEERING AND TECHNOLOGY, KODAKARA, THRISSUR

ACKNOWLEDGMENT

We would like to express our immense gratitude and profound thanks to all those who helped us to make this project a great success. We express our gratitude to the almighty God for all the blessings endowed on us.

We express our sincere thanks to our Executive Director **Rev.Fr. George Pareman** and Principal **Dr. Nixon Kuruvila** for providing us with such a great opportunity.

We also convey our gratitude to our Head of the Department **Dr. R Satheesh Kumar** for having given us a constant inspiration and suggestion. We extend our deep senseof gratitude to our project coordinator **Mr. Wilson Joseph C**, Assistant Professor ofComputer Science & Engineering Department for providing enlightening guidance through the project. We can hardly find words to express our deep appreciation for the help and warm encouragement that we have received from our project guide **Ms. Deepa Devassy**, Assistant Professor of Computer Science & Engineering Department for his whole-hearted support.

It was their encouragement that helped us to complete the project. We can hardly find words to express our deep appreciation of the help and warm encouragement that we received from our parents. We are extremely thankful and indebted to our friends who supported us in all aspects of the project work.

Rahul A B
Theres T L
Joel Jaison
Hanna De Grace

INSTITUTIONAL VISION

Evolve as a leading technology institute to create high calibre leaders and innovators of global standing with strong ethical values to serve the industry and society.

INSTITUTIONAL MISSION

Provide quality technical education that transforms students to be knowledgeable, skilled, innovative, and entrepreneurial professionals. Collaborate with academia andindustry around the globe, to strengthen the education and research ecosystem. Practice and promote high standards of professional ethics, good discipline, high integrity, and social accountability with a passion for holistic excellence.

QUALITY POLICY

We at Sahrdaya are committed to provide Quality Technical Education through continual improvement and by inculcating Moral & Ethical values to mold into VibrantEngineers with high Professional Standards.

We impart the best education through the support of competent & dedicated faculties, excellent infrastructure, and collaboration with industries to create ambience ofexcellence.

DEPARTMENTAL VISION

To be a nationally recognized center for quality education and research in diverse areas of computer science engineering with a strong social commitment.

DEPARTMENT MISSION

- Impart relevant technical knowledge, skills, and attributes along with valuesand ethics.
- Enhance creativity and quality in research through project-based learningenvironment.
- Mold Computer Science Engineering Professionals in synchronization with the dynamic industry requirements.
- Inculcate essential leadership qualities coupled with commitment to the society.

ROGRAMME EDUCATIONAL OBJECTIVES (PEOS)

PEO1	Take up challenging careers in suitable corporate, business, or educational sectors across the world, in multi-cultural work environment.
PEO2	Continuously strive for higher achievements in life keeping moral and ethical values such as honesty, loyalty, good relationship, and best performance, aloft.
PEO3	Be knowledgeable and responsible citizens with good teamwork skills, competent leadership qualities and holistic values.

PSO1	To nurture students with technically inquisitive attitude so that any real- world problem could be tackled with a problem-solving perspective, finding a suitable mathematical model with strong fundamental technological concepts to solve and apply to rapid growing arena of computer technology.
PSO2	To develop professionals with excellent exposure to the latest technologies to design high quality products unique in innovation, technology, software, security, hardware, and usefulness; making high impact on society, business, and technology.
PSO3	To enhance knowledge in practical implementation of technology regarding parallelism, virtualization of networks, scientific analysis and modeling, visualization, natural language processing, digital synthesis of data and its manipulation, wireless and mobile communication, storage, and retrieval of huge amount of data etc.

PROGRAMME OUTCOMES (POS)

PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processesthat meet the specified needs with appropriate consideration for thepublic health and safety, and the cultural, societal, and environmental considerations.

PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues, and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and teamwork: Function effectively as an individual,and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and ap ply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COURSE OBJECTIVES

To develop skills in doing literature survey, technical presentation, and report preparation.

COURSE OUTCOMES

The student will be able to

CO1	Envisage applications for societal needs.
CO2	Develop skills for analysis and synthesis of practical systems.
CO3	Learn to use new tools effectively and creatively.
CO4	Learns to carry out analysis and cost-effective, environmental friendly designs of engineering systems.
CO5	Develops the ability to write Technical / Project reports and oral presentation of the work done to an audience.

ABSTRACT

E-waste, Plastic, and other waste collection is something that Indian citizen and the whole world is in dire need as these wastes take a lot of time to degrade. As Electronic waste (e-waste) is the fastest growing waste on the planet, with an annual growth rate of 3-4%. It is estimated that e-waste generation will reach 52.2 million tons per annum by 2021. Presently, only 15% of e-waste is recycled. One of the most significant constituents of e-waste is plastics, accounting for almost for 20% of it. Despite several technological developments, their recycling is largely hindered due to presence of flame retardants. In this paper, we review some of the notable existing and emerging technologies such as micro factories being employed for e-waste plastics. Furthermore, we present their limitations, advantages, and potential for future development.

e-Waste is a web application which we developed to provide an effortless way to collect the e-waste products online. This application works with two modules admin and user where admin.is responsible of every activity done by user in the application. User can send a request to admin from anywhere and at any time about the garbage details in their apartment. Admin will accept user's request.

1. Introduction

This system is used for the easy collection of e-waste and plastic for households in any area based on the location give on the website. This system is implemented to help the Harithamithram Scheme which is being implemented by the municipalities as well as villages. With the help of this system, collectors can easily collect waste and doesn't need to visit all houses to ask if they have collected the waste or not. And with help of this system the collector can get the idea that the waste is full and can collect on spot to keep the environment clean.

1.1 General Background

To provide easier collection of waste and to help in using 3R's that is Reduce, Reuse and Recycling more efficiently. It also helps in keep the household, neighborhood clean thus leading to clean cities, town, villages etc. which indirectly lead to a clean country. With help o the system we can keep track of our own wastes which can help us to find ways to decrease our own wastes.

1.2 Motivation

To help the worker under the scheme an easier way to finish their work efficiently as well as to clear the piled-up wastes form each household faster. So that it would not cause problems to the owners and their family.

1.3 Problem Definition

The idea to this problem came to us when we were trying to find problem from our own surroundings. As the project should be a sustainable one we got the idea to create a website which help these working in under Harithamithram Scheme. As these people walk to all the houses in a village to collect waste. With help of the system, they only need to get to those houses whose waste have filled the required amount. Thus, help them to chart the way they must travel to retrieve the wastes from those houses.

2. Proposed System

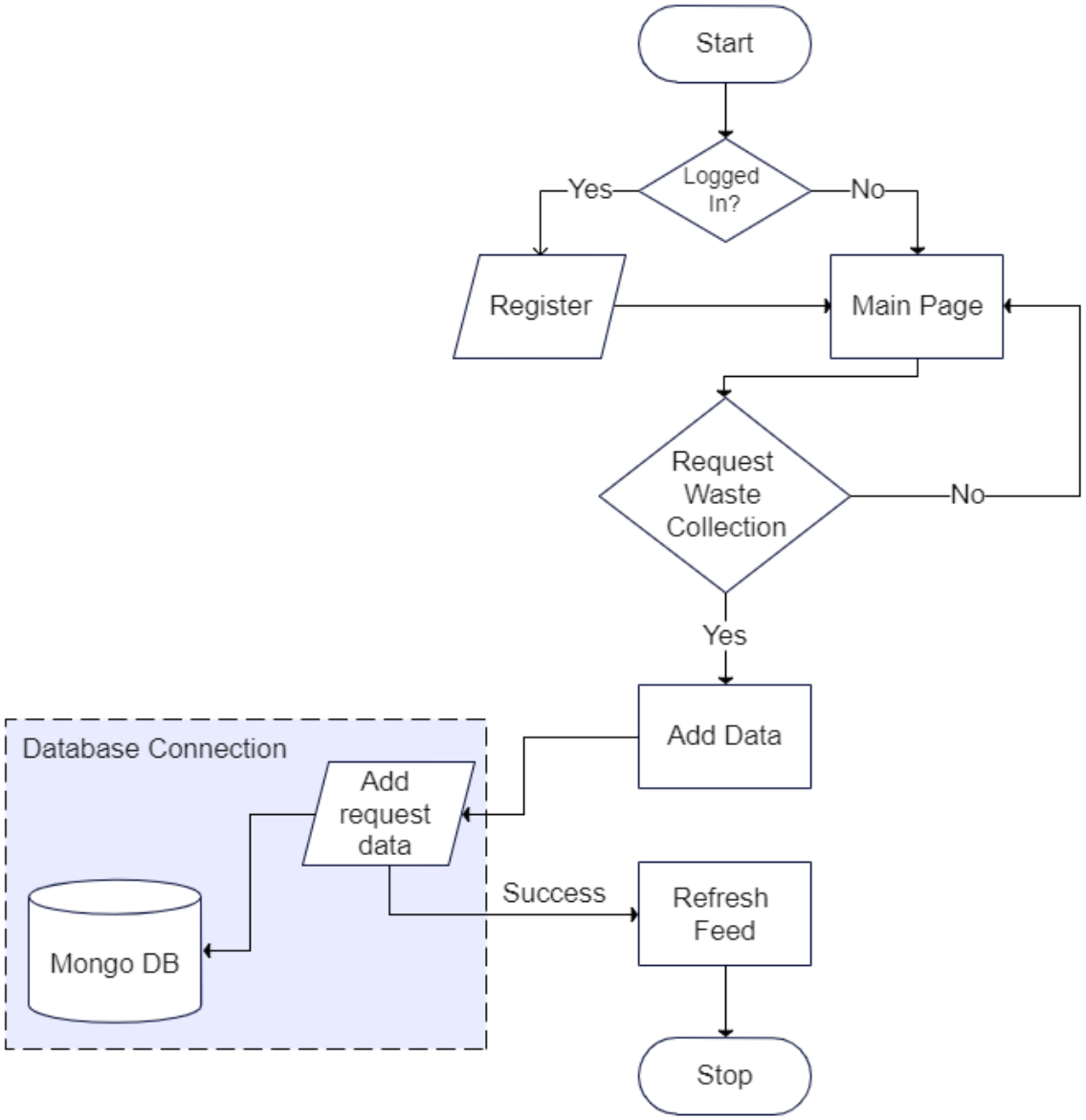
- The main purpose of online plastic and e-waste collection system is to provide another way for customers to give plastic and e-waste materials. The plastic and e-waste collection system are an Internet based application that can be accessed throughout the net and can be accessed by anyone who has a net connection. It is an automatic system, where we will automate the selling the waste material and enquiries about waste equipment and plastic are collection. After inserting the data to the database, staff need not to worry about the orders received through the system and hence reduces the manual labor. One of the best features of the system is to deploy or recycling the plastic, electrical and electronic from customer house and the city.

- **The goals of the system are:**
    - To provide anytime anyplace service for the customer.
    - To reuse electronic waste material by recycling or deploy
    - To decrease the electronic waste material from household
    - To obtain statistic information about the problems effect by the e-waste material.
    - To provide awareness about plastic, electrical and electronic material, or household.

3. Design Diagram

3.1. Architecture Diagram

3.2.Flowchart



4. Requirements

4.1. Software Requirements

1) Operation System : Windows or Ubuntu or Mac OS
2) Language : HTML, SCSS and JavaScript
3) Tools and Framework : NodeJS
4) Database : MongoDB

5. Conclusion

Even though the site was a success it wasn’t fully developed as when we add data and confirm the entry, we were to manually refresh the page to make the Harithamithram Scheme. Even though we have some short coming I hope that in the upcoming future we would improve and perfect this design. Electronic and electrical equipment's cannot be avoided in today's world. So also, is the case of waste electronic and electrical equipment. If this is a necessary evil, it must be best managed to minimize its adverse impacts on environment. Electronic waste piles are growing, as is their pollution potential. Most of these problems have their source in the development and design of the products concerned. Using this type of system, we can conclude that using the methodology of Reduce, Reuse and Recycle (3R) decrease the piles of electronic and electrical equipment, and make environment to be cleaned and healthy.

6. Code

a) App

```
require("dotenv").config();
const express = require("express");
const mongoose = require("mongoose");
const cookieParser = require("cookie-parser");
const path =  require('path');
const authRoute = require("./routes/auth");
const userRoute = require("./routes/user");
const adminRoute = require("./routes/admin");
const app=express();

app.use(express.json());
app.use(express.urlencoded());
app.use(cookieParser());
app.get("/api",(req,res)=> {
   res.send("This is react with backend");
});
app.post("/name", (req,res) =>{
   if(req.body.name) {
      return res.json({
         name: req.body.name
      });
   } else {
      return res.status(400).json({
         error: "No name provided"
      });
   }
})
app.use("/api/auth", authRoute)
app.use("/api/user",userRoute)
app.use("/api/admin",adminRoute)
app.use(express.static(path.resolve(__dirname, "./client/build")));
app.get("*", (req, res) => {
   res.sendFile(path.resolve(__dirname, "./client/build", "index.html"));
  });
   mongoose.connect(process.env.MONGO_URI).then(()=>{
      console.log("connected to database")
      app.listen(process.env.PORT,() =>{
         console.log(`Server running on port ${process.env.PORT}`)
      })
```

```
        }).catch((error)=>{
            console.log(error);
        });
```

b) Admin

```
const express = require("express");
const router = express.Router();
const requiresAuth = require("../middleware/permissions");
const UserInput = require("../models/UserInput");
const validateUserInput = require("../validation/userinputValidation");
router.get("/testuser", (req, res) => {
   res.send("Auth route working");
 });
 router.get("/collection"  , requiresAuth , async (req,res)=>{
   if (req.user.role!=='Collector') {
      return res.status(401).send("Unauthorized");
    }
    const incompletedCollection = await UserInput.find({complete:
false}).sort({completedAt: -1});
    return res.json({incompleted: incompletedCollection});
 })
 router.put("/:collectionId/complete" , requiresAuth , async (req,res)=>{
   try {
      const incompleted = UserInput.findOne({_id: req.params.collectionId});
      if (!incompleted) {
        return res.status(404).json({ error: "Could not find ToDo" });
       }
       if (incompleted.complete) {
        return res.status(400).json({ error: "ToDo is already complete" });
       }
       const updatedcollection = await UserInput.findOneAndUpdate(
        {
          _id: req.params.collectionId,
        },
        {
          complete: true
        },
        {
          new: true,
        }
       );
       return res.json(updatedcollection );
```

```
    } catch (error) {
        return res.status(400).send(error.message);
    }
})
  module.exports = router;
```

c) Customer

```
const express = require("express");
const router = express.Router();
const requiresAuth = require("../middleware/permissions");
const UserInput = require("../models/UserInput");
const validateUserInput = require("../validation/userinputValidation");
router.get("/testuser", (req, res) => {
    res.send("Auth route working");
  });
  router.post("/collect", requiresAuth , async (req,res)=> {
    try {
        const { errors , isValid } = validateUserInput(req.body);
        if (!isValid) {
            return res.status(400).json(errors);
          }
          const userInput = new UserInput({
            user: req.user._id,
            materials: req.body.materials,
            recyclable:req.body.recyclable,
            reuseable: req.body.reuseable,
            phonenumber: req.body.phonenumber,
            address: req.body.address
          });
          const saveduserInput = await userInput.save();
          return res.json(saveduserInput);
    } catch (error) {
        console.log(error)
        res.status(500).send(error)
    }
  })
  router.get("/current", requiresAuth, async (req , res)=> {
    try {
        const finishedCollection = await UserInput.find({user:
req.user._id,complete: true}).sort({completedAt: -1});
        const unfinishedCollection = await UserInput.find({user:
req.user._id,complete: false}).sort({createdat: -1});
```

```
      return res.json({incomplete: unfinishedCollection, complete:
finishedCollection})
    } catch (error) {
      return res.send(error.message).status(400);
    }
})
  module.exports = router;
```

d) Authentication

```
const express = require("express");
const router = express.Router();
const User = require("../models/User")
const validateRegisterInput = require("../validation/registerValidation")
const bcrypt = require("bcryptjs");
const jwt = require("jsonwebtoken");
const requiresAuth = require("../middleware/permissions");
//
router.get("/test", (req, res) => {
  res.send("Auth route working");
});
// @route   POST /api/auth/register
// @desc    Create a new user
// @access  Public
router.post("/register", async (req, res) => {
  try {
    const { errors, isValid } = validateRegisterInput(req.body);
    if (!isValid) {
      return res.status(400).json(errors);
    }
    // check for existing user
    const existingEmail = await User.findOne({
      email: new RegExp("^" + req.body.email + "$", "i"),
    });
    if (existingEmail) {
      return res
        .status(400)
        .json({ error: "There is already a user with this email" });
    }
    // hash the password
    const hashedPassword = await bcrypt.hash(req.body.password, 12);
    // create a new user
    const newUser = new User({
```

```
      email: req.body.email,
      password: hashedPassword,
      name: req.body.name,
      role:req.body.role,
      location:req.body.location
    });
    // save the user to the database
    const savedUser = await newUser.save();
    const payload = { userId: savedUser._id };
    const token = jwt.sign(payload, process.env.JWT_SECRET, {
      expiresIn: "7d",
    });
    res.cookie("access-token", token, {
      expires: new Date(Date.now() + 7 * 24 * 60 * 60 * 1000),
      httpOnly: true,
      secure: process.env.NODE_ENV === "production",
    });
    const userToReturn = { ...savedUser._doc };
    delete userToReturn.password;
    // return the new user
    return res.json(userToReturn);
  } catch (err) {
    // error here
    console.log(err);
    res.status(500).send(err.message);
  }
});
// @route   POST /api/auth/login
// @desc    Login user and return a access token
// @access  Public
router.post("/login", async (req, res) => {
  try {
    // check for the user
    const user = await User.findOne({
      email: new RegExp("^" + req.body.email + "$", "i"),
    });
    if (!user) {
      return res
        .status(400)
        .json({ error: "There was a problem with your login credentials" });
    }
    const passwordMatch = await bcrypt.compare(
      req.body.password,
```

```
      user.password
    );
    if (!passwordMatch) {
      return res
        .status(400)
        .json({ error: "There was a problem with your login credentials" });
    }
    const payload = { userId: user._id };
    const token = jwt.sign(payload, process.env.JWT_SECRET, {
      expiresIn: "7d",
    });
    res.cookie("access-token", token, {
      expires: new Date(Date.now() + 7 * 24 * 60 * 60 * 1000),
      httpOnly: true,
      secure: process.env.NODE_ENV === "production",
    });
    const userToReturn = { ...user._doc };
    delete userToReturn.password;
    return res.json({
      token: token,
      user: userToReturn,
    });
  } catch (err) {
    console.log(err);
    return res.status(500).send(err.message);
  }
});
// @route   GET /api/auth/current
// @desc    Return the currently authed user
// @access  Private
router.get("/current", requiresAuth, (req, res) => {
  if (!req.user) {
    return res.status(401).send("Unauthorized");
  }
  return res.json(req.user);
});
// @route   GET /api/auth/role
// @desc    Return the currently authed user
// @access  Private
router.get("/role", (req, res) => {
  if (!req.user) {
    return res.status(401).send("Unauthorized");
  }
```

```
  return res.json(req.user);
});
// @route   PUT /api/auth/logout
// @desc    Logout user a clear the cookie
// @access  Private
router.put("/logout", requiresAuth, async (req, res) => {
  try {
    res.clearCookie("access-token");
    return res.json({ success: true });
  } catch (err) {
    console.log(err);
    return res.status(500).send(err.message);
  }
});
module.exports = router;
```

e) Fetching Posts

```
import React from "react";
import { Link, useNavigate } from "react-router-dom";
import axios from "axios";
import { useGlobalContext } from "../context/GlobalContext";
const AuthBox = ({ register }) => {
  const { getCurrentUser, user } = useGlobalContext();
  const navigate = useNavigate();
  const [role,setRole]=React.useState("")
  const [email, setEmail] = React.useState("");
  const [password, setPassword] = React.useState("");
  const [confirmPassword, setConfirmPassword] = React.useState("");
  const [name, setName] = React.useState("");
  const [errors, setErrors] = React.useState({});
  React.useEffect(() => {
    if (user && navigate) {
      if(user.role === 'Collector'){
        navigate("/admin");
      }else if(user.role === 'User'){
        navigate("/users");
      }else {
        navigate('/')
      }
    }
  }, [user, navigate]);
  console.log(user);
```

```
  const onSubmit = (e) => {
    e.preventDefault();
    let data = {};
    if (register) {
      data = { name, role, email, password, confirmPassword, };
    } else {
      data = { email, password, };
    }
    axios
      .post(register ? "/api/auth/register" : "/api/auth/login", data)
      .then(() => {
        getCurrentUser();
      })
      .catch((err) => {
        if (err?.response?.data) {
          setErrors(err.response.data);
        }
      });
  };
  return (
    <div className="auth">
      <div className="auth__box">
        <div className="auth__header">
          <h1>{register ? "Register" : "Login"}</h1>
        </div>
        <form onSubmit={onSubmit}>
          {register && (
            <div className="auth__field">
              <label>Name</label>
              <input type="text" value={name} onChange={(e) =>
setName(e.target.value)} />
              {errors.name && <p className="auth__error">{errors.name}</p>}
            </div>
          )}
          {register && (
            <div className="auth__field">
              <label>Role</label>
              <input type="text" value={role} onChange={(e) =>
setRole(e.target.value)} />
              {errors.name && <p className="auth__error">{errors.role}</p>}
            </div>
          )}
          {register && (
```

```
          <div className="auth__field">
            <label>Location</label>
            <input type="text" value={role} onChange={(e) =>
setRole(e.target.value)} />
            {errors.name && <p className="auth__error">{errors.location}</p>}
          </div>
        )}
        <div className="auth__field">
          <label>Email</label>
          <input type="text" value={email} onChange={(e) =>
setEmail(e.target.value)} />
          {errors.email && <p className="auth__error">{errors.email}</p>}
        </div>
        <div className="auth__field">
          <label>Password</label>
          <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} />
          {errors.password && (
            <p className="auth__error">{errors.password}</p>
          )}
        </div>
        {register && (
          <div className="auth__field">
            <label>Confirm Password</label>
            <input type="password" value={confirmPassword} onChange={(e) =>
setConfirmPassword(e.target.value)} />
            {errors.confirmPassword && (
              <p className="auth__error">{errors.confirmPassword}</p>
            )}
          </div>
        )}
        <div className="auth__footer">
          {Object.keys(errors).length > 0 && (
            <p className="auth__error">{register ? "You have some validation
errors" : errors.error} </p>
          )}
          <button className="btn" type="submit" >{register ? "Register" :
"Login"}</button>
          {!register ? (
            <div className="auth__register">
              <p>Not a member? <Link to="/register">Register now</Link></p>
            </div>
          ) : (
```

```
            <div className="auth__register">
              <p>Already a member? <Link to="/">Login now</Link></p>
            </div>
          )}
        </div>
      </form>
    </div>
  </div>
 );
};
export default AuthBox;
```

f) Creating Posts

```
import React from "react";
import axios from "axios";
import { useGlobalContext } from "../context/GlobalContext";
import { useNavigate } from "react-router-dom";
import ToCompleteCard from "./ToCompleteCard";
const Users= () => {
 const { user , completeCollection,incompleteCollection  } =
useGlobalContext();
 console.log(incompleteCollection);
 console.log(completeCollection);
 const [materials, setMaterials] = React.useState("");
 const[recyclable,setRecyclable] = React.useState("")
 const[reuseable,setReuseable] = React.useState("")
 const[phonenumber,setPhonenumber] = React.useState("")
 const[address,setAddress] = React.useState("")
 const navigate = useNavigate();
 React.useEffect(() => {
  if (!user && navigate) {
   navigate("/");
  }
 }, [user, navigate]);
 let data={}
 const onSubmit = (e) => {
  data = { materials, recyclable, reuseable, phonenumber, address, }
   // getdata();
   e.preventDefault();
   axios.post("/api/user/collect",data).then((res) => {
    setMaterials("");
    setRecyclable("");
```

```
      setReuseable("");
      setPhonenumber("");
      setAddress("");
    });
  };
 return (
  <div className="User">
   <div className="hero__section">
    <div className="left__section">
     <div className="user_field">
      <form onSubmit={onSubmit}>
      <label>Enter the type of wastes</label>
      <input type="text" value={materials} onChange={(e) =>
setMaterials(e.target.value)} />
        <label>Enter the recyclable material</label>
      <input type="text" value={recyclable} onChange={(e) =>
setRecyclable(e.target.value)} />
      <label>Enter the reuseable  material</label>
      <input type="text" value={reuseable} onChange={(e) =>
setReuseable(e.target.value)} />
      <label>Enter the PhoneNumber</label>
      <input type="number" value={phonenumber} onChange={(e) =>
setPhonenumber(e.target.value)} />
      <label>Enter the address</label>
      <input type="text" value={address} onChange={(e)
=>setAddress(e.target.value)} />
        <button className="btn" type="submit">submit</button>
      </form>
     </div>
    </div>
    <div className="right__section">
     <div className="profile">
      <h2>{user.name}</h2>
      <h2>{user.email}</h2>
      <h2>{user.location}</h2>
     </div>
    </div>
   </div>
    <div className="toComplete">
         {/* <ToDoCard /> */}
         {incompleteCollection.map((incomplete, i) =>{
          return <ToCompleteCard toComplete={incomplete} key={i}/>
         })}
```

```
        </div>
      </div>
    )
}
export default Users;
```

Full Code Available In GitHub: klrab3490/PEWS: Plastic And E-Waste Collection System (github.com)