

NAME : VASHANTH TS

EMP NO : 12221

TITLE : **ELECTRICITY BILLING SYSTEM**

DESC :

### **Q17) Electricity Billing System**

A modern take on the traditional electricity billing system, this Java project streamlines collecting data from electricity meters. The primary objective is to automate and simplify the billing process, making it efficient and user-friendly.

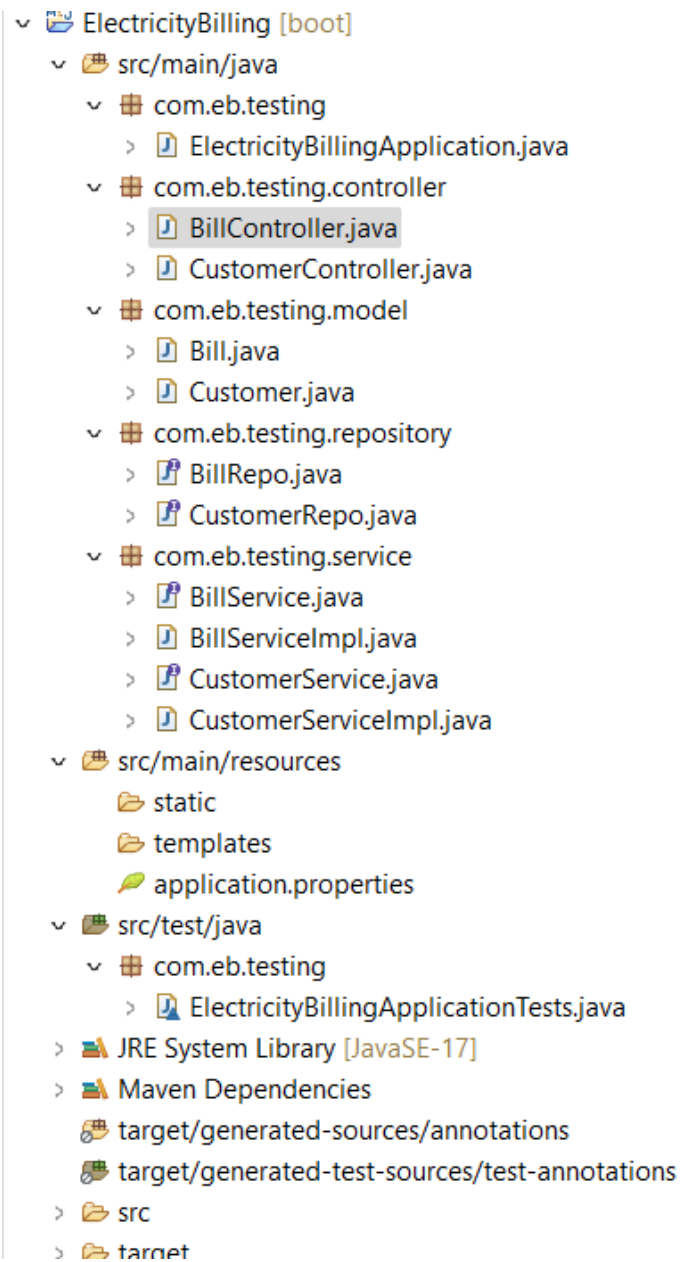
The software calculates the bill amount based on the units of electricity consumed in a month.

Key features of this project include:

User Stories to be Developed:

Perform the CRUD operations between Customer and Bill tables.

Spring Boot Folder structure :



Model :

```
package com.eb.testing.model;
```

```
import jakarta.persistence.CascadeType;
```

```
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.OneToOne;
import jakarta.persistence.Table;
```

```
@Entity
```

```
@Table(name = "Billing_tbl")
```

```
public class Bill {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    @Column(name = "Bill_id")
```

```
    private int id;
```

```
    @Column(name = "Rate_Type")
```

```
    private String type;
```

```
    @Column(name = "EB-Branch")
```

```
    private String city;
```

```
@Column(name = "Contact")
```

```
private long mobile;
```

```
@Column(name = "Bill_Due_Date")
```

```
private String date;
```

```
@Column(name = "UnitsConsumed")
```

```
private int nos;
```

```
@Column(name = "PricePerUnit")
```

```
private int price;
```

```
@Column(name = "Total_Bill")
```

```
private int bills;
```

```
@OneToOne(targetEntity=Customer.class, cascade = CascadeType.MERGE)
```

```
@JoinColumn
```

```
private Customer customer;
```

```
public Bill() {
```

```
super();
```

```
}
```

```
public Bill(int id, String type, String city, long mobile, String date, int nos, int price, int bills,  
Customer customer) {  
    super();  
    this.id = id;  
    this.type = type;  
    this.city = city;  
    this.mobile = mobile;  
    this.date = date;  
    this.nos = nos;  
    this.price = price;  
    this.bills = bills;  
    this.customer = customer;  
}
```

```
public int getId() {  
    return id;  
}
```

```
public void setId(int id) {  
    this.id = id;  
}
```

```
public String getType() {  
    return type;  
}
```

```
public void setType(String type) {  
    this.type = type;  
}
```

```
public String getCity() {  
    return city;  
}
```

```
public void setCity(String city) {  
    this.city = city;  
}
```

```
public long getMobile() {  
    return mobile;  
}
```

```
public void setMobile(long mobile) {  
    this.mobile = mobile;  
}
```

```
public String getDate() {  
    return date;  
}
```

```
public void setDate(String date) {  
    this.date = date;  
}
```

```
public int getNos() {  
    return nos;  
}
```

```
public void setNos(int nos) {  
    this.nos = nos;  
}
```

```
public int getPrice() {  
    return price;  
}
```

```
public void setPrice(int price) {  
    this.price = price;  
}
```

```
public int getBill() {  
    return bills;  
}
```

```
public void setBill(int bills) {  
    this.bills = bills;  
}
```

```
public Customer getCustomer() {  
    return customer;  
}
```

```
public void setCustomer(Customer customer) {
```



```
this.customer = customer;  
}
```

```
@Override
```

```
public String toString() {  
    return "Bill [id=" + id + ", type=" + type + ", city=" + city + ", mobile=" + mobile + ", date=" +  
    date  
    + ", nos=" + nos + ", price=" + price + ", bill=" + bills + ", customer=" + customer + "];"  
}
```

```
}
```

```
package com.eb.testing.model;
```

```
import jakarta.persistence.Column;
```

```
import jakarta.persistence.Entity;
```

```
import jakarta.persistence.GeneratedValue;
```

```
import jakarta.persistence.GenerationType;
```

```
import jakarta.persistence.Id;
```

```
import jakarta.persistence.Table;
```

```
@Entity
```

```
@Table(name = "Customer_tbl")
```

```
public class Customer {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    @Column(name = "Customer_ID")
```

```
    private int cid;
```

```
    @Column(name = "UserName")
```

```
    private String uname;
```

```
    @Column(name = "Address")
```

```
    private String addr;
```

```
    @Column(name = "Mobile")
```

```
    private long cmobile;
```

```
    @Column(name = "Payment_Type")
```

```
    private String pt;
```

```
    @Column(name = "Bill_Amount_Status")
```

```
    private String billsts;
```

```
public Customer() {  
    super();  
  
}
```

```
public Customer(int cid, String uname, String addr, long cmobile, String pt, String billsts) {  
    super();  
    this.cid = cid;  
    this.uname = uname;  
    this.addr = addr;  
    this.cmobile = cmobile;  
    this.pt = pt;  
    this.billsts = billsts;  
}
```

```
public int getCid() {  
    return cid;  
}
```

```
public void setCid(int cid) {  
    this.cid = cid;  
}
```

```
public String getUname() {  
    return uname;
```

```
}
```

```
public void setUsername(String username) {  
    this.username = username;  
}
```

```
public String getAddress() {  
    return address;  
}
```

```
public void setAddress(String address) {  
    this.address = address;  
}
```

```
public long getCellular() {  
    return cellular;  
}
```

```
public void setCellular(long cellular) {  
    this.cellular = cellular;  
}
```

```
public String getCity() {  
    return city;  
}
```

```
public void setPt(String pt) {  
    this.pt = pt;  
}
```

```
public String getBillSts() {  
    return billsts;  
}
```

```
public void setBillSts(String billsts) {  
    this.billsts = billsts;  
}
```

@Override

```
public String toString() {  
    return "Customer [cid=" + cid + ", uname=" + uname + ", addr=" + addr + ", cmobile=" +  
    cmobile + ", pt=" + pt  
    + ", billstatus=" + billsts + "];"  
}  
  
}
```

Repo :

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import com.eb.testing.model.Bill;
```

```
public interface BillRepo extends JpaRepository<Bill, Integer> {  
  
}
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import com.eb.testing.model.Customer;
```

```
public interface CustomerRepo extends JpaRepository<Customer, Integer>{  
  
}
```

Service :

```
import java.util.List;
```

```
import java.util.Optional;
```

```
import com.eb.testing.model.Bill;
```

```
public interface BillService {
```

```
public void addbill(Bill bill);
```

```
public Optional<Bill> getbill(int id);
```

```
public List<Bill> getall();
```

```
public void updatebill(Bill bill);
```

```
public void deletebill(int id);
```

```
}
```

```
import java.util.List;
```

```
import java.util.Optional;
```

```
import org.springframework.stereotype.Service;
```

```
import com.eb.testing.model.Bill;
```

```
import com.eb.testing.repository.BillRepo;
```

```
@Service
```

```
public class BillServiceImpl implements BillService {
```

```
    BillRepo repo;
```

```
    public BillServiceImpl(BillRepo repo) {
```

```
        super();
```

```
        this.repo = repo;
```

```
}
```

```
public void addbill(Bill bill) {
```

```
    repo.save(bill);
```

```
}
```

```
public Optional<Bill> getbill(int id) {
```

```
    return repo.findById(id);
```

```
}
```

```
public List<Bill> getall() {
```

```
    return repo.findAll();
```

```
}
```

```
public void updatebill(Bill bill) {
```

```
    repo.save(bill);
```

```
}
```

```
public void deletebill(int id) {
```

```
    repo.deleteById(id);
```

```
}
```



```
}
```

```
import java.util.List;
```

```
import com.eb.testing.model.Customer;
```

```
public interface CustomerService {
```

```
    public void adduser(Customer customer);
```

```
    public List<Customer> getall();
```

```
    public void updatecustomer(Customer customer);
```

```
    public void deletecustomer(int cid);
```

```
}
```

```
import java.util.List;
```

```
import java.util.Optional;
```

```
import org.springframework.stereotype.Service;
```

```
import com.eb.testing.model.Customer;
import com.eb.testing.repository.CustomerRepo;

@Service

public class CustomerServiceImpl implements CustomerService {

    CustomerRepo repo;

    public CustomerServiceImpl(CustomerRepo repo) {
        super();
        this.repo = repo;
    }

    public void adduser(Customer customer) {

        repo.save(customer);
    }

    public List<Customer> getall() {

        return repo.findAll();
    }

    public Optional<Customer> getuser(int cid) {
```

```
return repo.findById(cid);  
}
```

```
public void updatecustomer(Customer customer) {
```

```
repo.save(customer);  
}
```

```
public void deletecustomer(int cid) {
```

```
repo.deleteById(cid);  
}
```

```
}
```

Controller :

```
import java.util.List;
```

```
import java.util.Optional;
```

```
import org.springframework.web.bind.annotation.CrossOrigin;
```

```
import org.springframework.web.bind.annotation.DeleteMapping;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.PutMapping;
```

```
import org.springframework.web.bind.annotation.RequestBody;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```

import org.springframework.web.bind.annotation.RestController;

import com.eb.testing.model.Bill;
import com.eb.testing.service.BillServiceImpl;

@RestController
@CrossOrigin(origins = "http://localhost:3000")
@RequestMapping("/bill")
public class BillController {

    BillServiceImpl service;

    public BillController(BillServiceImpl service) {
        super();
        this.service = service;
    }

    String success = "Success";
    String failure = "Failure";

    @PostMapping
    public String addbill(@RequestBody Bill bill) {

        String msg = "";
        try {
            service.addbill(bill);
            msg = success;
        } catch (Exception e) {

            msg = failure;
        }
        return msg;
    }

    @GetMapping("/{id}")
    public Optional<Bill> getOrderId(@PathVariable("id") int id) {

        return service.getbill(id);

    }

    @GetMapping("/all")
    public List<Bill> getbills() {

        return service.getall();
    }

```

```
}
```

```
@PutMapping
```

```
public String updateBill(@RequestBody Bill bill) {
```

```
String msg = "";
```

```
try {
```

```
service.updatebill(bill);
```

```
msg = success;
```

```
} catch (Exception e) {
```

```
msg = failure;
```

```
}
```

```
return msg;
```

```
}
```

```
@DeleteMapping("{id}")
```

```
public String deleteBillbyID(@PathVariable("id") int id) {
```

```
String msg = "";
```

```
try {
```

```
service.deletebill(id);
```

```
msg = success;
```

```
} catch (Exception e) {
```

```
msg = failure;
```

```
}
```

```
return msg;
```

```
}
```

```
}
```

```
import java.util.List;
```

```
import java.util.Optional;
```

```
import org.springframework.web.bind.annotation.CrossOrigin;
```

```
import org.springframework.web.bind.annotation.DeleteMapping;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.PutMapping;
```

```
import org.springframework.web.bind.annotation.RequestBody;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```

import org.springframework.web.bind.annotation.RestController;

import com.eb.testing.model.Customer;
import com.eb.testing.service.CustomerServiceImpl;

@RestController
@CrossOrigin(origins = "http://localhost:3000")
@RequestMapping("/customer")
public class CustomerController {

    CustomerServiceImpl service;

    public CustomerController(CustomerServiceImpl service) {
        super();
        this.service = service;
    }

    String success = "Success";
    String failure = "Failure";

    @PostMapping
    public String insertcustomer(@RequestBody Customer customer) {

        String msg = "";
        try {
            service.adduser(customer);
            msg = success;
        } catch (Exception e) {
            msg = failure;
        }
        return msg;

    }

    @GetMapping("/{cid}")
    public Optional<Customer> getbyId(@PathVariable("cid") int id) {

        return service.getuser(id);

    }

    @GetMapping("/all")
    public List<Customer> getusers() {

        return service.getall();
    }

```

```
}
```

```
@PutMapping
```

```
public String updateuser(@RequestBody Customer customer) {
```

```
String msg = "";
```

```
try {
```

```
service.updatecustomer(customer);
```

```
msg = success;
```

```
} catch (Exception e) {
```

```
msg = failure;
```

```
}
```

```
return msg;
```

```
}
```

```
@DeleteMapping("{cid}")
```

```
public String deleteStudentbyID(@PathVariable("cid") int id) {
```

```
String msg = "";
```

```
try {
```

```
service.deletecustomer(id);
```

```
msg = success;
```

```
} catch (Exception e) {
```

```
msg = failure;
```

```
}
```

```
return msg;
```

```
}
```

```
}
```

## MYSQL

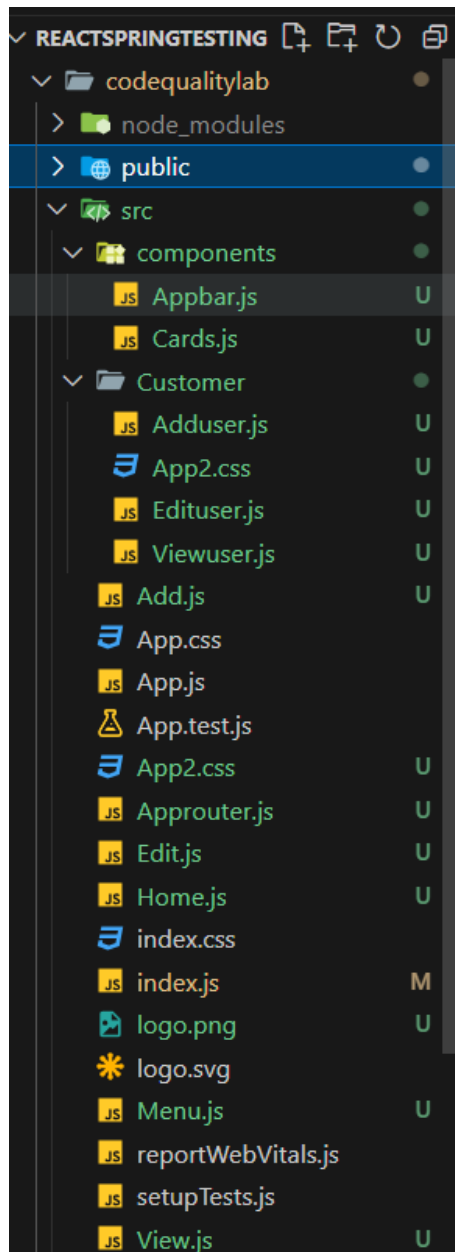
Result Grid									
		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
bill_id	total_bill	eb-branch	bill_due_date	contact	units_consumed	price_per_unit	rate_type	customer_customer_id	
1	2000	Madurai-TNHBColony	19-07-2024	92900149165	500	4	Residential	1	
2	3600	Madurai-Main	9-07-2024	92900149165	900	4	Residential	3	
6	1600	Madurai-Viragnoor	5-07-2024	95432123	400	4	Residential	5	

Result Grid						
		Filter Rows:		Edit:		Export/Import:
customer_id	address	bill_amount_status	mobile	payment_type	user_name	
1	Senthamil-7thSt,Avaniyapuram,Madurai-12	Pending	8668159203	Online	Vashanth	
3	No-2,PRC,Kaalavsal,Madurai-19	Paid	9068132503	Online	Nandha	
5	Chinthamani, 7th st, Madurai-09	Pending	82179180102	EBBoard	KarthiKeyan	

Frontend React :

Folder Structure :





## Components

### Cards:

```
import Button from 'react-bootstrap/Button';
import Card from 'react-bootstrap/Card';

const DispCard=( {imageUrl, title, description, })=>{
  return (

    <Card style={{ width: '18rem' }}>
```

```

    <Card.Img variant="top" src={imageUrl} />
    <Card.Body>
      <Card.Title>{title}</Card.Title>

    </Card.Body>
  </Card>
);
}
export default DispCard;

```

## AppBar

```

import Button from 'react-bootstrap/Button';
import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Nav from 'react-bootstrap/Nav';
import Navbar from 'react-bootstrap/Navbar';
import NavDropdown from 'react-bootstrap/NavDropdown';
import Offcanvas from 'react-bootstrap/Offcanvas';

function AppBar() {
  return (
    <>
      {[ 'lg' ].map((expand) => (
        <Navbar key={expand} expand={expand} className="bg-body-tertiary mb-3"
bg="warning" data-bs-theme="dark">
          <Container fluid>
            <Navbar.Brand href="#">TNEB</Navbar.Brand>
            <Navbar.Toggle aria-controls={`offcanvasNavbar-expand-${expand}`} />
            <Navbar.Offcanvas
              id={`offcanvasNavbar-expand-${expand}`}
              aria-labelledby={`offcanvasNavbarLabel-expand-${expand}`}
              placement="end"
            >
              <Offcanvas.Header closeButton>
                <Offcanvas.Title id={`offcanvasNavbarLabel-expand-${expand}`}>
                  Offcanvas
                </Offcanvas.Title>
              </Offcanvas.Header>
              <Offcanvas.Body>
                <Nav className="justify-content-end flex-grow-1 pe-3">

```

```

        <Nav.Link href="/">Home</Nav.Link>
        <Nav.Link href="/Menus">About</Nav.Link>
        <Nav.Link href="/view">Billing</Nav.Link>

        <Nav.Link href="/payroll">Customers</Nav.Link>
        <Nav.Link href="/signup">Sign up</Nav.Link>
        <NavDropdown
          title="Login"
          id={`offcanvasNavbarDropdown-expand-${expand}`}
        >
          <NavDropdown.Item href="/login">User</NavDropdown.Item>
          <NavDropdown.Item href="#action4">
            Admin
          </NavDropdown.Item>
          <NavDropdown.Divider />
          <NavDropdown.Item href="#action5">

        </NavDropdown.Item>
        </NavDropdown>
      </Nav>
      <Form className="d-flex">
        <Form.Control
          type="search"
          placeholder="Search"
          className="me-2"
          aria-label="Search"
        />
        <Button variant="outline-success">Search</Button>
      </Form>
    </Offcanvas.Body>
  </Navbar.Offcanvas>
</Container>
</Navbar>
  )})
</>
);
}

export default Appbar;

```

Bill Pages

```
import React from 'react'
```

```

import axios from "axios";
import { useState } from 'react';
import { useEffect } from 'react';
import { useNavigate } from 'react-router-dom';

function Add() {
  const[inputdata, setInputdata] = useState({
    id: -1,
    type: "",
    city: "",
    mobile: "",
    date: "",
    nos: "",
    price: "",
    bills:"",
    customer: {
      cid:"",
      uname: "",
      addr: "",
      cmobile: "",
      pt: "",
      billSts: ""
    }
  });
  const [id, setId] = useState([]);
  const [data,setData] = useState([]);
  const naviget = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:1000/customer/all")
      .then((response) => setData(response.data))
      .catch((err) => console.log(err));
  }, []);

  const handleChange = (e) => {
    setId(e.target.value);
    axios
      .get("http://localhost:1000/customer/" + e.target.value)

      .then((response) => {
        setInputdata((inputData) => ({
          ...inputData,

```

```

        customer: response.data
      }));
      console.log(response.data);
    })
    .catch((err) => {
      console.log(err);
    });
  });

let handleSubmit=async (e)=>{
  e.preventDefault();
  let result =validateValues(inputdata);

  if(result === true){
    axios
      .post("http://localhost:1000/bill", inputdata)
      .then((res) => {
        alert("Data added Successfully");
        naviget("/");
        console.log(res.data);
      })
      .catch((err) => console.log(err));
  }else {
    alert("Please Enter the Valid Inputs!!!");
  }

}

const validateValues = (inputdata) => {
  if (inputdata.mobile.length < 5) {
    alert("Mobile Num should be Max 10 Numbers Only !!!");
    return false;
  } else if (inputdata.city.length === 0) {
    alert("Please enter City Name !!!");
    return false;
  } else {

```

```

        return true;
    }
};

return (
    <div
        id="add2"
        className="d-flex w-100 vh-100 justify-content-center align-items-center "
    >
        <div className="w-50 border bg-light p-5">
            <form onSubmit={handleSubmit}>
                <h1>ADD EB BILL DETAILS</h1>
                <div>
                    <label htmlFor="name">EB TYPE</label>
                    <input
                        type="text"
                        name="type"
                        className="form-control"
                        onChange={(e) =>
                            setInputdata({ ...inputdata, type: e.target.value })
                        }
                    />
                </div>
                <div>
                    <label htmlFor="Mobile">Mobile :</label>
                    <input
                        type="number"
                        name="mobile"
                        className="form-control"
                        onChange={(e) =>
                            setInputdata({ ...inputdata, mobile: e.target.value })
                        }
                    />
                </div>
                <div>
                    <label htmlFor="city">Address/City Branch :</label>
                    <input
                        type="text"
                        name="city"
                        className="form-control"
                        onChange={(e) =>
                            setInputdata({ ...inputdata, city: e.target.value })
                        }
                    />
                </div>
            </form>
        </div>
    </div>
);

```

```
    />
  </div>

  <div>
    <label htmlFor="desg">Bill Due Date :</label>
    <input
      type="text"
      name="desg"
      className="form-control"
      onChange={(e) =>
        setInputdata({ ...inputdata, date: e.target.value })
      }
    />
  </div>

  <div>
    <label htmlFor="nos">Consumed Units :</label>
    <input
      type="number"
      name="nos"
      className="form-control"
      onChange={(e) =>
        setInputdata({ ...inputdata, nos: e.target.value })
      }
    />
  </div>

  <div>
    <label htmlFor="price">Price Per Unit :</label>
    <input
      type="number"
      name="price"
      className="form-control"
      onChange={(e) =>
        setInputdata({ ...inputdata, price: e.target.value })
      }
    />
  </div>

  <div>
    <label htmlFor="price">Bill Amount :</label>
```

```
    <input
      type="number"
      name="bills"
      className="form-control"
      value={parseInt(inputdata.price)*parseInt(inputdata.nos)}
      onChange={(e) =>
        setInputdata({ ...inputdata, bills:e.target.value })
      }
    />
  </div>
```

```
    <div>
    <label htmlFor="selectId"> Customer ID:</label>
    <select id="selectId" onChange={handleChange}>
      <option value="" >Select an ID</option>
      {data.map((item) => (
        <option key={item.id} value={item.id}>
          {item.cid}
        </option>
      ))}
    </select>

    </div>
```

```
    <br />
```

```
    <button className="btn btn-info ">Submit</button>
  </form>
```

```
  </div>
</div>
```

```
)
```



```
}  
  
export default Add
```

```
import axios from "axios";  
import React, { useEffect, useState } from "react";  
import { useNavigate, useParams } from "react-router-dom";  
import "./App2.css";  
  
function Edit() {  
  const { id } = useParams();  
  const [data, setData] = useState([]);  
  const [datas, setDatas] = useState([]);  
  const [pid, setPid] = useState([]);  
  // const [inputData,setInputdata]=useState([]);  
  const navigate = useNavigate();  
  useEffect(() => {  
    axios  
      .get("http://localhost:1000/bill/" + id)  
      .then((response) => setData(response.data))  
      .catch((err) => console.log(err));  
  }, []);  
  
  useEffect(() => {  
    axios  
      .get("http://localhost:1000/customer/all")  
      .then((response) => setDatas(response.data))  
      .catch((err) => console.log(err));  
  }, []);  
  
  const handleChange = (e) => {  
    setPid(e.target.value);  
    axios  
      .get("http://localhost:1000/customer/" + e.target.value)  
      .then((response) => {  
        setData((data) => ({  
          ...data,  
          customer: response.data  
        }));  
        console.log(response.data);  
      });  
  };  
}
```

```

    })
    .catch((err) => {
        console.log(err);
    });
});

let handleSubmit = (e) => {
    e.preventDefault();
    axios.put("http://localhost:1000/bill" , data).then((res) => {
        alert("Bill Updated Successfully");
        navigate("/");
    });
};

return (
    <div>
        <div
            id="edit2"
            className="d-flex w-100 vh-100 justify-content-center align-items-center"
"
        >
            <div className="w-50 border bg-light p-5">
                <form onSubmit={handleSubmit}>
                    <h1>UPDATE BILL DETAILS</h1>
                    <div>
                        <label htmlFor="id">ID :</label>
                        <input
                            type="text"
                            disabled
                            name="name"
                            className="form-control"
                            value={data.id}
                        />
                    </div>
                    <div>
                        <label htmlFor="name">EB TYPE</label>
                        <input
                            type="text"
                            name="type"
                            className="form-control"
                            value={data.type}
                            onChange={(e) =>

```

```

        setData({ ...setData, type: e.target.value })
    }
    />
</div>
<div>
    <label htmlFor="mobile">Mobile :</label>
    <input
        type="number"
        name="mobile"
        className="form-control"
        value={data.mobile}
        onChange={(e) => setData({ ...data, mobile: e.target.value })}
    />
</div>

<div>
    <label htmlFor="city">Address/ City Branch :</label>
    <input
        type="text"
        name="city"
        className="form-control"
        value={data.city}
        onChange={(e) => setData({ ...data, city: e.target.value })}
    />
</div>

<div>
    <label htmlFor="desg">Bill Due Date :</label>
    <input
        type="text"
        name="desg"
        className="form-control"
        value={data.date}
        onChange={(e) =>
            setData({ ...data, date: e.target.value })
        }
    />
</div>

<div>
    <label htmlFor="nos">Consumed Units :</label>

```

```

    <input
      type="number"
      name="nos"
      className="form-control"
      value={data.nos}
      onChange={(e) =>
        setData({ ...data, nos: e.target.value })
      }
    />
  </div>

  <div>
    <label htmlFor="price">Price Per Unit :</label>
    <input
      type="number"
      name="price"
      className="form-control"
      value={data.price}
      onChange={(e) =>
        setData({ ...data, price: e.target.value })
      }
    />
  </div>

  <div>
    <label htmlFor="price">Bill Amount :</label>
    <input
      type="number"
      name="bills"
      className="form-control"
      value={parseInt(data.price)*parseInt(data.nos)}
      onChange={(e) =>
        setData({ ...data, bills:
parseInt(data.price)*parseInt(data.nos)})
      }
    />
  </div>

  <div>
    <label htmlFor="selectId">Select Customer ID:</label>
    <select id="selectId" onChange={handleChange}>

```

```

        <option value="" >Select an ID</option>
        {datas.map((item) => (
            <option key={item.cid} value={item.cid}>
                {item.cid}
            </option>
        ))}
    </select>

</div>

    <br />

    <button className="btn btn-info ">Update</button>
</form>
</div>
</div>
</div>
);
}

export default Edit;

```

```

import React from 'react'
import './App2.css'
// import Footer from './Footer'
// import Grid from './Grid'
// import CTA from './CTA'
// import Contact from './Contact'
import Menu from './Menu'
function Home() {
    return (
        <div>
            {/* <CTA/> */}
            <div id='img'>

                </div>
                <Menu/>
            {/* <Grid/>

```

```

        <Contact/>
        <Footer/> */}
    </div>
)
}

export default Home;

import React from 'react'
import DispCard from './components/Cards'
function Menu() {
    return (
        <div id="Menus" >
            <div className="container">

                <h1></h1>
                <div className="row">
                    <div className="col-lg-3 col-md-6 col-sm-12 mt-4">
                        <DispCard imageUrl={"https://www.shutterstock.com/image-
photo/high-voltage-posthigh-tower-sky-260nw-1440470978.jpg"}
                            title={"Connected Everywere Across States , Districts ,
Villages ,Cities"}

                                />
                            </div>
                            <div className="col-lg-3 col-md-6 col-sm-12 mt-4">
                                <DispCard imageUrl={"https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcSWoypeBGrqPVKvp5Qmiy2abc3poPijZE1I1Dbd3hQgalF
Nd0k8CmoqCI11H0Jk0qzjIFw&usqp=CAU"}
                                    title={"Solar Energy - An Alternative Renewable Source"}
                                    description={"Hot nd Spicy "
                                    btn={"Order Now!!"}/>
                                </div>
                                <div className="col-lg-3 col-md-6 col-sm-12 mt-4">
                                    <DispCard
imageUrl={"https://c1.wallpaperflare.com/preview/796/514/362/power-high-pressure-
pylons-transmission-line-thumbnail.jpg"}
                                        title={"Day nd Night Service"}
                                        description={" "}
                                        btn={"Order Now!!"}/>
                                    </div>
                                <div className="col-lg-3 col-md-6 col-sm-12 mt-4">

```

```

        <DispCard imageUrl={"https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQ_Z8e203Sd1vH_JmTAWPltimPhRIy9aqASEmEk-
G10TY6Nfbo3Bh3Fnqd6gB0vu3KtPHs&usqp=CAU"}
        title={"Powerfull YTransformers"}
        description={"Fries Friday Offer "}
        btn={"Order Now!!"}/>
    </div>
</div>
<div className="row">
    <div className="col-lg-3 col-md-6 col-sm-12 mt-4">
        <DispCard imageUrl={"https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcSNuS9LcJv1ByPkkYuBouWmk7F2Qu0BYzvSIA&s"}
        title={"Proper Maintenance"}
        description={" "}
        btn={"Order Now!!"}/>
    </div>
    <div className="col-lg-3 col-md-6 col-sm-12 mt-4">
        <DispCard imageUrl={"https://assets-news.housing.com/news/wp-
content/uploads/2021/07/27201256/How-to-apply-for-new-electricity-connection-in-
Tamil-Nadu-FB-1200x700-compressed.jpg"}
        title={"Digital Meters "}
        description={"More than 20+ varieties"}
        btn={"Order Now!!"} />
    </div>
    <div className="col-lg-3 col-md-6 col-sm-12 mt-4">
        <DispCard imageUrl={"https://digitria.in/wp-
content/uploads/2022/02/TNEB-Reading.jpg"}
        title={"E-Payment"}
        description={" "}
        btn={"Order Now!!"}/>
    </div>
    <div className="col-lg-3 col-md-6 col-sm-12 mt-4">
        <DispCard
imageUrl={"https://www.livechennai.com/businesslistings/News_photo/eb8918.jpg"}
        title={"Monthly Wise EB-Reading"}
        description={" "}
        btn={"Order Now!!"}/>
    </div>
</div>

<br></br>
<br></br>

</div>

```

```

        <br></br>
        <br></br>
    </div>

)
}

export default Menu

import React, { useEffect, useState } from "react";
import axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import "./App2.css";

function View() {
    const [columns, setColumns] = useState([]);
    const [records, setRecords] = useState([]);
    const navigate = useNavigate();
    const [find, setFind] = useState('');
    useEffect(() => {
        axios
            .get("http://localhost:1000/bill/all")
            .then((response) => {
                setColumns(Object.keys(response.data[0]));
                setRecords(response.data);
            })
            .catch((err) => {
                console.log(err);
            });
    }, []);

    let handleSubmit = (id) => {
        const conf = window.confirm("Do you want to delete");
        if (conf) {
            axios

                .delete("http://localhost:1000/bill/" + id)
                .then((res) => {
                    alert("record has deleted");
                    navigate("/");
                    window.location.reload();
                });
        }
    };
}

```



```

    })
    .catch((err) => console.log(err));
  }

};

return (
  <div id="body" >
    <body>

      <div className="container">
        <h1 id="app2" className="text-center text-bg-warning ">
          Electricity Billing Payment System
        </h1>

        <br></br>
        <br></br>
        <div className="relative">
          {/* <svg xmlns="http://www.w3.org/2000/svg" className="absolute
top-0 bottom-0 w-6 h-6 my-auto text-gray-400 left-3" fill="none" viewBox="0 0 24
24" stroke="currentColor">
            <path strokeLinecap="round" strokeLinejoin="round"
strokeWidth={2} d="M21 21l-6-6m2-5a7 7 0 11-14 0 7 7 0 0114 0z" />
          </svg> */}
          <input

            type="text"
            onChange={(e)=>setFind(e.target.value)}
            placeholder="SearchName"
            className="w-60 h-10 py-3 pl-12 pr-4 text-gray-500 border
rounded-md outline-none bg-gray-50 focus:bg-white focus:border-indigo-600"
          />
        </div>

        <div className="text-end">
          <Link to="/create" className="btn btn-primary">
            Add +
          </Link>
        </div>
      <br />
    </body>
  </div>

```

```

<table className="table table-bordered table-striped w-100 border bg-
white shadow px-5 pb-5 rounded ">
  <thead>
    <tr>
      <th key={i}>{c}</th>
    </tr>
    <th>Bill ID</th>
    <th>EB Type</th>

    <th>Contact detail</th>
    <th> Branch</th>
    <th> Due-Date</th>
    <th>Consumed Units</th>
    <th>Price Per Unit</th>
    <th>Bill Amount</th>
    <th>Customer ID</th>
    <th>Customer Name</th>
    <th>Action</th>
  </thead>

  <tbody>
    {records.filter((items)=>{

      return
find.toLowerCase()==='?items :items.customer.uname.toLowerCase().includes(find)
    })

    .map((d, i) => (
      <tr key={i}>
        <td>{d.id}</td>
        <td>{d.type}</td>
        <td>{d.mobile}</td>
        <td>{d.city}</td>
        <td>{d.date}</td>
        <td>{d.nos}</td>
        <td>{d.price}</td>
        <td>{d.bills}</td>
        <td>{d.customer.cid}</td>
        <td>{d.customer.uname}</td>
        <td>

```

```

        <Link
            id="butt1"
            to={` /update/${d.id}`}
            className="btn btn-sm btn-transparent border border-secondary
"
        >
            Update
        </Link>
        <button
            id="butt2"
            onClick={(e) => handleSubmit(d.id)}
            className="btn btn-sm ms-1 btn-danger"
        >
            Delete
        </button>
    </td>
</tr>
    </tbody>
</table>
</div>

</body>
</div>
);
}

export default View;

```

## Customer

```

import React from 'react'
import axios from "axios";
import { useState } from 'react';
import { useEffect } from 'react';
import { useNavigate } from 'react-router-dom';

function Add() {
    const[inputdata, setInputdata] = useState({
        cid: "",
        uname: "",
        addr: "",
        cmobile: "",
    })
    const [error, setError] = useState('');
    const navigate = useNavigate();

    const handleSubmit = (e) => {
        e.preventDefault();
        const { cid, uname, addr, cmobile } = inputdata;
        if (cid === '' || uname === '' || addr === '' || cmobile === '') {
            setError('All fields are required');
            return;
        }
        axios.post('http://localhost:3001/customer', inputdata)
            .then((res) => {
                console.log(res);
                navigate('/');
            })
            .catch((err) => {
                setError(err.message);
            });
    };

    return (
        <div>
            <h3>Add Customer</h3>
            <form>
                <input type="text" value={cid} onChange={handleChange} />
                <input type="text" value={uname} onChange={handleChange} />
                <input type="text" value={addr} onChange={handleChange} />
                <input type="text" value={cmobile} onChange={handleChange} />
                <button type="button" value="Add" onClick={handleSubmit} />
            </form>
            <div>
                {error}
            </div>
        </div>
    );
}

```

```

    pt: "",
    billSts: ""

  });
  const [data, setData] = useState([]);
  const naviget = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:1000/customer/all")
      .then((response) => setData(response.data))
      .catch((err) => console.log(err));
  }, []);

  let handleSubmit=(e)=>{
    e.preventDefault();
    let result =validateValues(inputdata);

    if(result === true){
      axios
        .post("http://localhost:1000/customer", inputdata)
        .then((res) => {
          alert("Data added Successfully");
          naviget("/");
          console.log(res.data);
        })
        .catch((err) => console.log(err));
    }else {
      alert("Please Enter the Valid Inputs!!!");
    }
  }
  const validateValues = (inputdata) => {
    if (inputdata.bill === 0) {
      alert("Bill Cannot be empty value!!! ");
      return false;
    } else {
      return true;
    }
  };

  return (

```

```

<div
  id="add2"
  className="d-flex w-100 vh-100 justify-content-center align-items-center "
>
  <div className="w-50 border bg-light p-5">
    <form onSubmit={handleSubmit}>
      <h1>ADD CUSTOMER DATA</h1>
      <div>
        <label htmlFor="uname">User Name</label>
        <input
          type="text"
          name="uname"
          className="form-control"
          onChange={(e) =>
            setInputdata({ ...inputdata, uname: e.target.value })
          }
        />
      </div>
      <div>
        <label htmlFor="addr">Address :</label>
        <input
          type="text"
          name="addr"
          className="form-control"
          onChange={(e) =>
            setInputdata({ ...inputdata, addr: e.target.value })
          }
        />
      </div>

      <div>
        <label htmlFor="cmobile">Mobile :</label>
        <input
          type="number"
          name="cmobile"
          className="form-control"
          onChange={(e) =>
            setInputdata({ ...inputdata, cmobile: e.target.value })
          }
        />
      </div>

      <div>
        <label htmlFor="pt">Payment Type :</label>

```

```
<input
  type="Text"
  name="pt"
  className="form-control"
  onChange={(e) =>
    setInputdata({ ...inputdata, pt: e.target.value })
  }
/>
</div>
```

```
<div>
  <label htmlFor="bill">Bill Status :</label>
  <input
    type="text"
    name="billSts"
    className="form-control"

    //
value={parseInt(inputdata.bp)+parseInt(inputdata.hr)+parseInt(inputdata.da)+parse
Int(inputdata.ta)}
    onChange={(e) =>
      setInputdata({ ...inputdata, billSts: e.target.value })
    }
  />
</div>
```

```
<br />
```

```
<button className="btn btn-info ">Submit</button>
</form>
```

```

        </div>
    </div>
)
}

export default Add

import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import "./App2.css";

function Edit() {
    const { cid } = useParams();
    // const [data, setData] = useState([]);
    // const [datas, setDatas] = useState([]);
    // const [pid, setPid] = useState([]);
    const [inputData, setInputdata] = useState([]);
    const navigate = useNavigate();
    useEffect(() => {
        axios
            .get("http://localhost:1000/customer/" + cid)
            .then((response) => setInputdata(response.data))

            .catch((err) => console.log(err));
    }, []);

    let handleSubmit = (e) => {
        e.preventDefault();
        axios.put("http://localhost:1000/customer" , inputData).then((res) => {
            alert("User Updated Successfully");
            navigate("/");
        });
    };

    return (
        <div>
            <div>

```

```

    id="edit2"
    className="d-flex w-100 vh-100 justify-content-center align-items-center"
  "
  >
  <div className="w-50 border bg-light p-5">
    <form onSubmit={handleSubmit}>
      <h1>UPDATE CUSTOMER DATA'S</h1>
      <div>
        <label htmlFor="id">ID :</label>
        <input
          type="text"
          readOnly
          name="name"
          className="form-control"
          value={inputData.cid}
          onChange={(e) =>
            setInputdata({ ...inputData, cid: e.target.value })
          }
        />
      </div>
      <div>
        <label htmlFor="uname">User Name</label>
        <input
          type="text"
          name="uname"
          className="form-control"
          value={inputData.uname}
          onChange={(e) =>
            setInputdata({ ...inputData, uname: e.target.value })
          }
        />
      </div>
      <div>
        <label htmlFor="addr">Address :</label>
        <input
          type="text"
          name="addr"
          value={inputData.addr}
          className="form-control"
          onChange={(e) =>
            setInputdata({ ...inputData, addr: e.target.value })
          }
        />
      </div>
    </div>
  </div>

```



```

<div>
  <label htmlFor="cmobile">Mobile :</label>
  <input
    type="number"
    name="cmobile"
    value={inputData.cmobile}
    className="form-control"
    onChange={(e) =>
      setInputdata({ ...inputData, cmobile: e.target.value })
    }
  />
</div>

```

```

<div>
  <label htmlFor="pt">Payment Type :</label>
  <input
    type="Text"
    name="pt"
    value={inputData.pt}
    className="form-control"
    onChange={(e) =>
      setInputdata({ ...inputData, pt: e.target.value })
    }
  />
</div>

```

```

<div>
  <label htmlFor="bill">Bill Status :</label>
  <input
    type="text"
    name="bill"
    value={inputData.billSts}
    className="form-control"
    //
    value={parseInt(inputdata.bp)+parseInt(inputdata.hr)+parseInt(inputdata.da)+parse
    Int(inputdata.ta)}
    onChange={(e) =>
      setInputdata({ ...inputData, billSts: e.target.value })
    }
  />
</div>

```

```

        <br />

        <button className="btn btn-info ">Update</button>
    </form>
</div>
</div>
</div>
);
}

```

```
export default Edit;
```

```

import React, { useEffect, useState } from "react";
import axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import "./App2.css";

```

```

function View() {
    const [columns, setColumns] = useState([]);
    const [records, setRecords] = useState([]);
    const [orders, setOrders] = useState([]);
    const navigate = useNavigate();
    const [find, setFind] = useState('');
    useEffect(() => {
        axios
            .get("http://localhost:1000/customer/all")
            .then((response) => {
                setColumns(Object.keys(response.data[0]));
                setRecords(response.data);
            })
            .catch((err) => {
                console.log(err);
            });
    }, []);

    useEffect(() => {
        axios

```

```

        .get("http://localhost:1000/bill/all")
        .then((response) => setOrders(response.data))
        .catch((err) => console.log(err));
    }, []]);

    // Calculate total bill for each user
    const calculateTotalBill = (cid) => {
        const userOrders = orders.filter(bill => bill.customer.cid === cid);
        return userOrders.reduce((total, bill) => total + bill.price * bill.nos, 0);
    };

    let handleSubmit = (cid) => {
        const conf = window.confirm("Do you want to delete");
        if (conf) {
            axios

                .delete("http://localhost:1000/customer/" + cid)
                .then((res) => {
                    alert("record has deleted");
                    navigate("/");
                    window.location.reload();
                })
                .catch((err) => console.log(err));
        }
    };

    return (
        <body>
        <div id="body">

            <div className="container  ">
                <h1 id="app2" className="text-center text-bg-success ">
                    EB Customer Data System
                </h1>
                <br></br>
                <br></br>
                <div className="relative">

                    <input
                        type="text"
                        onChange={(e)=>setFind(e.target.value)}
                        placeholder="SearchId"

```

```

        className="w-60 h-10 py-3 pl-12 pr-4 text-gray-500 border
rounded-md outline-none bg-gray-50 focus:bg-white focus:border-indigo-600"
      />
    </div>
    {/* <div>
      <input type="search" onChange={(e)=>setFind(e.target.value)}
placeholder="Search ID"></input>
    </div> */}
    <div className="text-end">
      <Link to="/insertpay" className="btn btn-primary">
        Add +
      </Link>
    </div>
    <br />
    <table className="table table-bordered table-striped w-100 border bg-
white shadow px-5 pb-5 rounded ">
      <thead>
        <tr>
          {/* {columns.map((c, i) => (
            <th key={i}>{c}</th>

          ))} */}
          <th>Customer ID</th>
          <th>Customer Name</th>
          <th>Customer Address</th>
          <th>Contact detail</th>
          <th>Payment Type</th>
          <th>Bill Status</th>
          <th>Bill Amount</th>
          <th>Action</th>
        </tr>
      </thead>

      <tbody>
        {records.filter((items)=>{
          return find === ''?items :items.cid==parseInt(find)
        })

        .map((d, i) => (
          <tr key={i}>
            <td>{d.cid}</td>
            <td>{d.uname}</td>

```

```

        <td>{d.addr}</td>
        <td>{d.cmobile}</td>
        <td>{d.pt}</td>
        <td>{d.billSts}</td>
        <td>{calculateTotalBill(d.cid)}</td>
        <td>
            <Link
                id="butt1"
                to={` /payupdate/${d.cid}`}
                className="btn btn-sm btn-transparent border border-secondary
"
            >
                Update
            </Link>
            <button
                id="butt2"
                onClick={(e) => handleSubmit(d.cid)}
                className="btn btn-sm ms-1 btn-danger"
            >
                Delete
            </button>
        </td>
    </tr>
    </tbody>
    </table>
</div>
</div>
</body>
);
}

export default View;

div #app2{
    margin-top: 40px;
    text-align: center;
}
table{
    border-radius: 10px;
}

```

```
div #add2{
    /* background:
url('https://pearlsol.com/img/breadcrumb/hd%20blog%20pic.jpg'); */
    background:cover;
}
div #edit2{

    /* background:
url('https://pearlsol.com/img/breadcrumb/hd%20blog%20pic.jpg'); */
    background:cover;

}
div{
    font-family: Verdana, Geneva, Tahoma, sans-serif;

}

.nav{
    text-decoration: none;
    padding: 20px;

}

.nav :hover{
    color: cyan;
}
#form{
    align-items: center;
    justify-content: center;
    justify-items: center;
    width: 50%;
    height: 100%;
    margin-left: 20%;
    box-shadow: 5px 5px 5px rgba(67, 67, 67, 0.942) ;

}
#form input {
    align-content: center;
    align-items: center;
    justify-content: center;
    justify-items: center;
    margin-top: 5%;
```

```
margin-left: 35%;
height: 40px;
width: 40%;
border-radius: 5px;
border: none;
background-color: rgba(78, 78, 78, 0.477);
color: rgb(255, 255, 255);
}
#navbar :hover{

    color: cyan;
}

#form button{
    justify-content: center;
    justify-items: center;
    margin-top: 5%;
    margin-left: 45%;
    height: 40px;
    border: none;
    background-color: rgb(49, 171, 35);
    border-radius: 10px;
    margin-bottom: 5%;
}

#form h2{
    text-align: center;
}

h2{
    color: white;
}
#Box{
    padding: 20px;
    gap: 30px;
    justify-content: flex-end;
    align-items: end;
    float: right;
    align-content: end;
```

```
}  
#Box :hover{  
    color: aqua;  
}  
  
body {  
    /* background: url(bg1.png); */  
    /* backdrop-filter: blur(5px); */  
    background-repeat: no-repeat;  
    width: 100%;  
    height: 100%;  
    background-size: cover;  
}  
#form1{  
    background: linear-gradient(rgb(255, 247, 187),rgb(255, 178, 223));  
    border-radius: 20px;  
}  
  
#form2{  
    background: linear-gradient(rgb(116, 183, 254),rgb(106, 252, 203));  
    border-radius: 20px;  
}  
Link #butt1 :hover{  
    background-color: rgb(1, 254, 13);  
    color: rgb(255, 255, 255);  
}
```



TNEB

HomeAboutBillingCustomersSign upLoginSearch

Search

# Welcome to TNEB Electric Billing Portal !!

Electricity Billing Payment System

SearchName

Add +

Bill ID	EB Type	Contact detail	Branch	Due-Date	Consumed Units	Price Per Unit	Bill Amount	Customer ID	Customer Name	Action
1	Residential	92900149165	Madurai-TNHBColony	19-07-2024	500	4		1	Vashanth	<div>UpdateDelete</div>
2	Residential	92900149165	Madurai-Main	9-07-2024	900	4		3	Nandha	<div>UpdateDelete</div>
6	Residential	95432123	Madurai-Viragnoor	5-07-2024	400	4		5	KarthiKeyan	<div>UpdateDelete</div>

localhost:3000/view

Search

ENG IN16:2903-07-2024

localhost:3000/view

Search

ENG IN16:3003-07-2024

TNEB

HomeAboutBillingCustomersSign upLoginSearch

Electricity Billing Payment System

k

Add +

Bill ID	EB Type	Contact detail	Branch	Due-Date	Consumed Units	Price Per Unit	Bill Amount	Customer ID	Customer Name	Action
6	Residential	95432123	Madurai-Viragnoor	5-07-2024	400	4	1600	5	KarthiKeyan	<div>Update</div> <div>Delete</div>

16:3203-07-2024

TNEB

localhost:3000 says  
Data added Successfully

Sign upLoginSearch

ADD CUSTOMER DATA

User Name

Nagarjune

Address :

KamrajSalai,6thMainSt

Mobile :

928014365

Payment Type :

UPI

Bill Status :

Done

Submit

16:4203-07-2024

localhost:3000/create

Sign up Login Search

localhost:3000 says  
Data added Successfully

## ADD EB BILL DETAILS

EB TYPE  
Residential

Mobile :  
95432123

Address/City Branch :  
Madurai-Munchalai

Bill Due Date :  
17-07-2024

Consumed Units :  
1200

Price Per Unit :  
4

Bill Amount :  
4800

Customer ID: 6

Submit

localhost:3000/payroll

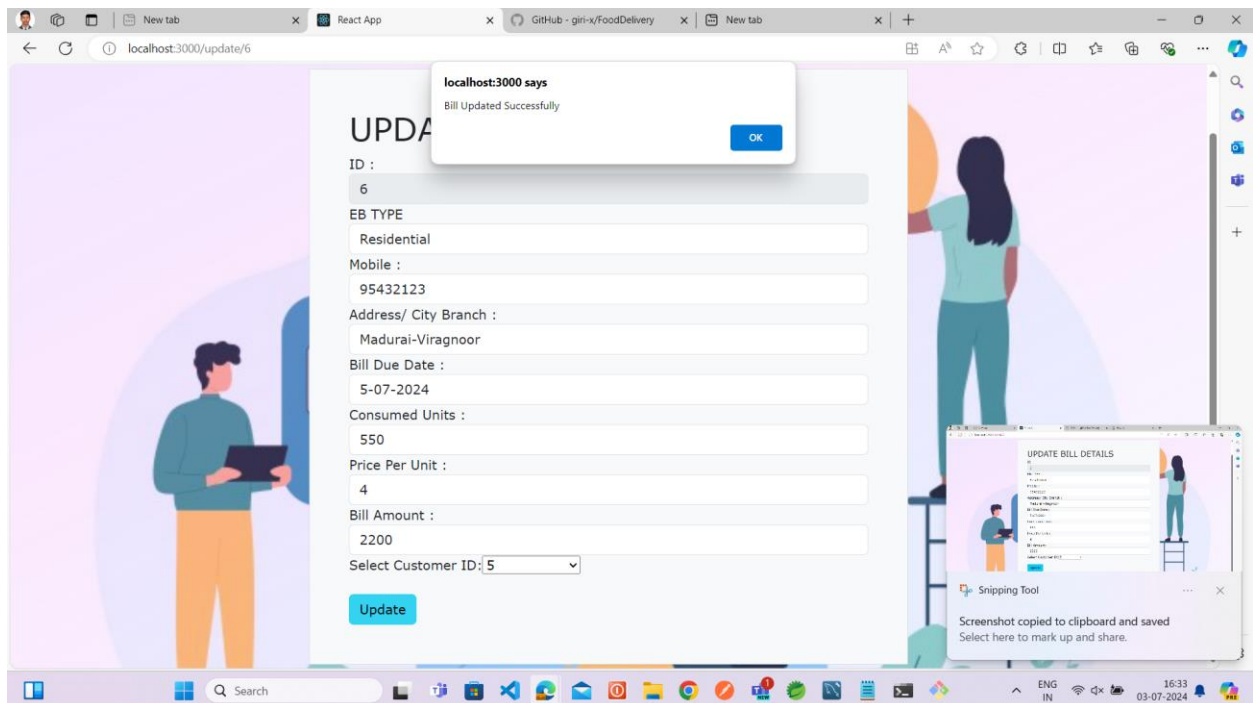
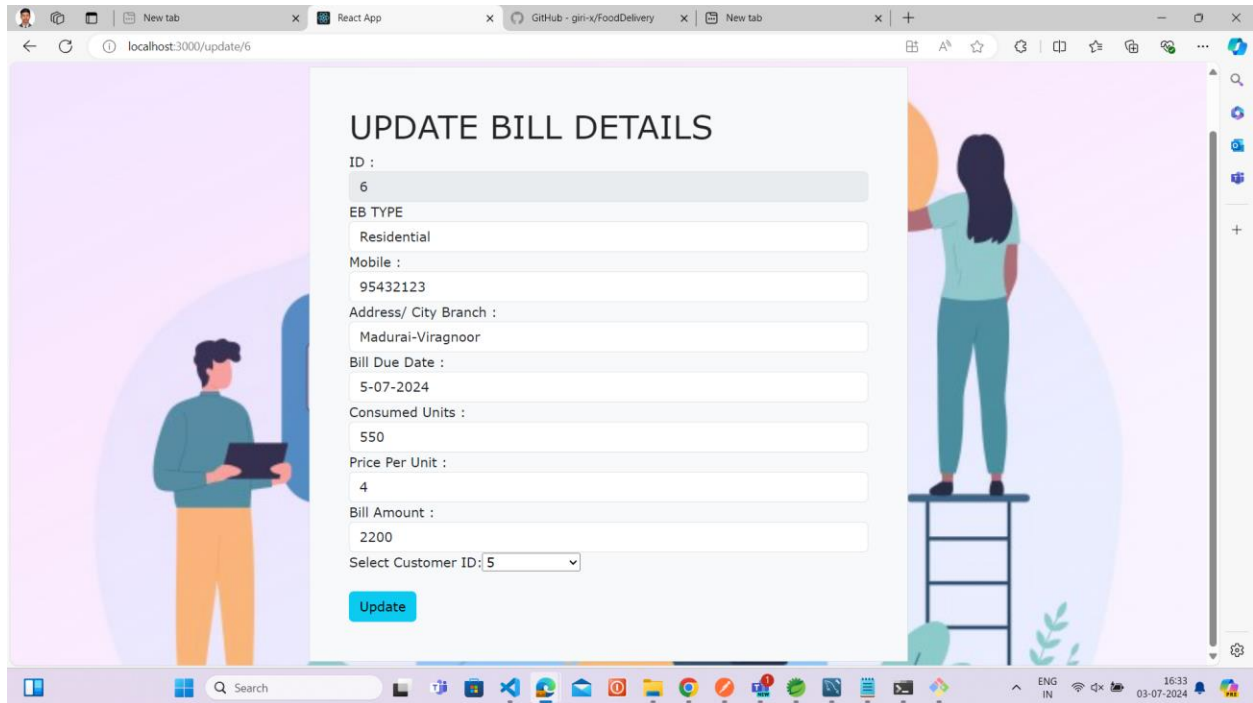
Home About Billing Customers Sign up Login Search

## EB Customer Data System

SearchId

Add +

Customer ID	Customer Name	Customer Address	Contact detail	Payment Type	Bill Status	Bill Amount	Action
1	Vashanth	Senthamil-7thSt,Avaniyapuram,Madurai-12	8668159203	Online	Pending	2000	<button>Update</button> <button>Delete</button>
3	Nandha	No-2,PRC,Kaalavsal,Madurai-19	9068132503	Online	Paid	3600	<button>Update</button> <button>Delete</button>
5	KarthiKeyan	Chinthamani,7th st, Madurai-09	82179180102	EBBoard	Pending	2200	<button>Update</button> <button>Delete</button>
6	Nagarjune	KamrajSalai,6thMainSt	928014365	UPI	Done	4800	<button>Update</button> <button>Delete</button>



TNEB

HomeAboutBillingCustomersSign upLoginSearch

EB Customer Data System

SearchId

BILL

+

Customer ID	Customer Name	Customer Address	Contact detail	Payment Type	Bill Status	Bill Amount	Action
1	Vashanth	Senthamil-7thSt,Avaniyapuram,Madurai-12	8668159203	Online	Pending	2000	<div>Update</div> <div>Delete</div>
3	Nandha	No-2,PRC,Kaalavsal,Madurai-19	9068132503	Online	Paid	3600	<div>Update</div> <div>Delete</div>
5	KarthiKeyan	Chinthamani,7th st, Madurai-09	82179180102	EBBoard	Pending	2200	<div>Update</div> <div>Delete</div>

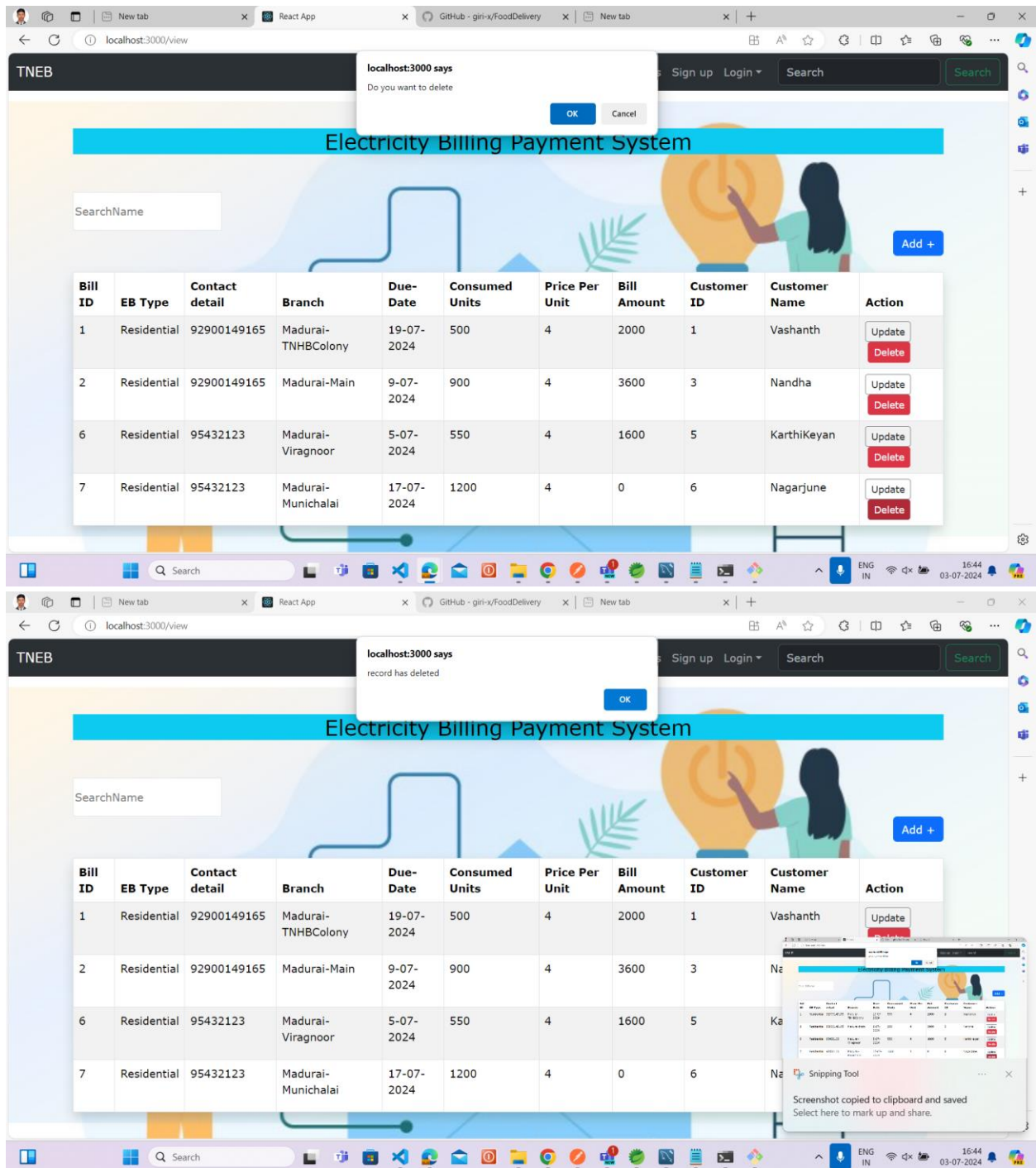
+

ENG IN

16:41

03-07-2024





SonarCube Report Before Correcting :

There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes. [Learn More](#)

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

ElectricityBilling main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

**Passed**  
All conditions passed.

New Code Overall Code

0 Bugs Reliability A

0 Vulnerabilities Security A

0 Security Hotspots Security Review A

20min Debt 3 Code Smells Maintainability A

0.0% 1 0.0% 0

There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes. [Learn More](#)

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

ElectricityBilling main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

My Issues All

Filters

Issues in new code

Type CODE SMELL

Severity

Scope

Resolution

Status

Security Category

Bulk Change

1 / 3 issues 20min effort

src/main/java/com/eb/testing/model/Bill.java

Rename field "bill" Code Smell Major Open Not assigned 10min effort Comment 1 minute ago L46 brain-overload

Complete the task associated to this TODO comment. Code Smell Info Open Not assigned 0min effort Comment 1 minute ago L56 cwe

src/test/java/com/eb/testing/ElectricityBillingApplicationTests.java

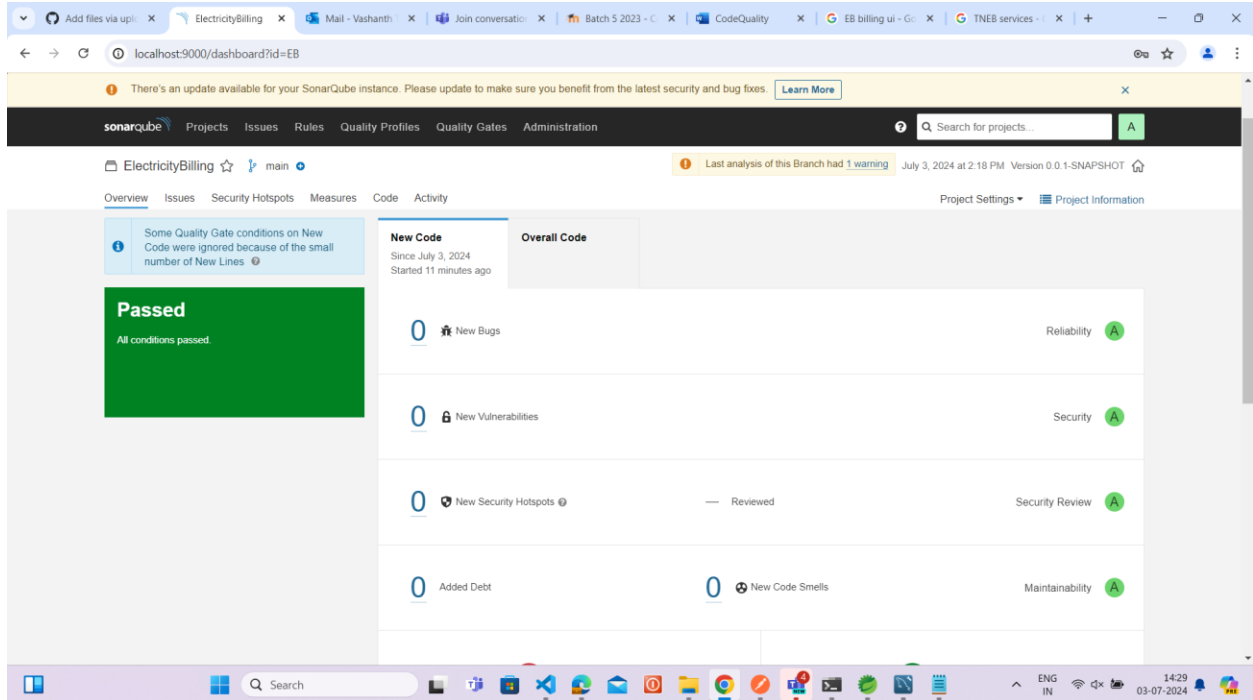
Add at least one assertion to this test case. Code Smell Blocker Open Not assigned 10min effort Comment 1 minute ago L10 junit, tests

3 of 3 shown

Embedded database should be used for evaluation purposes only  
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA  
Community Edition - Version 9.9.4 (build 87374) - LGPLv3 - Community - Documentation - Plugins - Web API

SonarCube Report After Correction :



Review Process :

MINGW64:/d/CodeQualityReview

```
vashanth.thoppey@RLP2186 MINGW64 /d/CodeQualityReview
$ git clone https://github.com/giri-x/FoodDelivery.git
Cloning into 'FoodDelivery'...
remote: Enumerating objects: 55, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 55 (delta 5), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (55/55), 21.88 KiB | 4.38 MiB/s, done.
Resolving deltas: 100% (5/5), done.

vashanth.thoppey@RLP2186 MINGW64 /d/CodeQualityReview
$
```



github.com/Vip-Vashi/CodeQualityLab/pull/1#issuecomment-2205575864

Back to notifications

Done Unsubscribe Mark as unread Save

Open Add files via upload #1  
Vip-Vashi wants to merge 2 commits into main from Vashanth-12221

Vip-Vashi requested a review from giri-x 29 minutes ago

Update README.md Verified 856454e

giri-x commented 1 minute ago

Report.pdf  
Code Quality is maintained well

Add more commits by pushing to the Vashanth-12221 branch on Vip-Vashi/CodeQualityLab.

Review requested  
Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#) Show all reviews

1 pending reviewer

Require approval from specific reviewers before merging  
[Rulesets](#) ensure specific people approve pull requests before they're merged. Add rule

Continuous integration has not been set up

No one—assign yourself

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

Development  
Successfully merging this pull request may close these issues.  
None yet

Notifications  
Customize  
Unsubscribe  
You're receiving notifications because you're watching this repository.

2 participants

Product Solutions Resources Open Source Enterprise Pricing

Search or jump to... Sign in Sign up

giri-x / FoodDelivery Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags

giri-x Add files via upload

FoodDelivery Add files via upload

README.md Initial commit

README

FoodDelivery

Clone

HTTPS GitHub CLI

https://github.com/giri-x/FoodDelivery.git Copy url to clipboard

Clone using the web URL

Open with GitHub Desktop

Download ZIP

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Report repository

Releases

No releases published

Packages

No packages published

Languages

Java 100.0%

Snipping Tool

Screenshot copied to clipboard and saved. Select here to mark up and share.

giri-x requested your review on this pull request. [Add your review](#)

## Add files via upload #1

giri-x wants to merge 1 commit into `main` from `Girivasan-12213`

Conversation 0 Commits 1 Checks 0 Files changed 28 +1,055 -0

giri-x commented 1 minute ago

No description provided.

Add files via upload Verified bae74e9

giri-x requested a review from Vip-Vashi 1 minute ago

Add more commits by pushing to the `Girivasan-12213` branch on `giri-x/DeliverySystem`.

**Review requested**

Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#)

1 pending reviewer

**This branch has no conflicts with the base branch**

Merging can be performed automatically.

Reviewers: Vip-Vashi

Still in progress? [Convert to draft](#)

Assignees: No one—[assign yourself](#)

Labels: None yet

Projects: None yet

Milestone: No milestone

Development

Add files via upload #1

Vip-Vashi wants to merge 1 commit into `main` from `Vashanth-12221`

Vip-Vashi commented now

Review Request

Add files via upload Verified 2936543

Vip-Vashi requested a review from giri-x now

Add more commits by pushing to the `Vashanth-12221` branch on `Vip-Vashi/CodeQualityLab`.

**Review requested**

Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#)

1 pending reviewer

giri-x was requested for review

**Require approval from specific reviewers before merging**

[Rulesets](#) ensure specific people approve pull requests before they're merged. [Add rule](#)

**Continuous integration has not been set up**

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

**This branch has no conflicts with the base branch**

Merging can be performed automatically.

Reviewers: giri-x

Still in progress? [Learn about draft PRs](#)

Assignees: No one—[assign yourself](#)

Labels: None yet

Projects: None yet

Milestone: No milestone

Development

Successfully merging this pull request may close these issues.

Notifications: [Customize](#)

[Unsubscribe](#)

You're receiving notifications because you're watching this repository.

Vip-Vashi reviewed now [View reviewed changes](#)

Vip-Vashi left a comment Collaborator

Well Structured Code , Followed coding standards and code quality