**SAVITRIBAI PHULE PUNE UNIVERSITY**


**A PRELIMINARY PROJECT REPORT ON**

# "E-Gram Panchayat"

**SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE REQUIREMENTS OF**

**BACHELOR OF ENGINEERING (TE COMPUTERENGINEERING)**

**Academic Year: 2020-21**

**By:**

Rohit Naikade TECOC339

Mehul Lokhande TECOC331

Vijay Dabhade TECOC308

Govind Madankar TECOC332

**Under The Guidance of**

**Prof. Anagha N. Chaudhari**

**PIMPRI CHINCHWAD COLLEGE OF ENGINEERING,**

**SECTOR 26, NIGDI, PRADHIKARAN**

**DEPARTMENT OF COMPUTER ENGINEERING**

# CERTIFICATE

certify that the project work entitled **"E-Gram Panchayat**" is a bonafide work carried out by

| | |
|---|---|
| **Rohit Naikade** | **TECOC339** |
| **Mehul Lokhande** | **TECOC331** |
| **Vijay Dabhade** | **TECOC308** |
| **Govind Madankar** | **TECOC332** |

The report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the project.

Mini-Project Coordinator Mini-Project Coordinator

Head of the Department

# *ACKNOWLEDGEMENT*

We feel immense pleasure in presenting this project report on "**E-Gram Panchayat**". We wish to express true sense of gratitude towards **Prof.Anagha Chaudhari**, our project guide who at very discrete step in study of this project, contributed her valuable guidance and helped to solve every problem that arose.

We take this opportunity to thank our Principal **Dr. N.B. Chopade** and **Prof. Dr. K. Rajeswari** (HOD Comp Dept.) for opening the doors of the department towards the realization of our project, for their guidance and encouragement.

Most importantly, we would also like to express our sincere gratitude towards all the staff members of Computer Department. We also express our thanks to all our friends for their support and suggestions shown during the completion of our project. We take this opportunity to express our thanks to all who rendered their valuable help, along with all those unseen people across the internet for maintaining those valuable resources for the successful completion of our project. I owe our success to all of them.

**Rohit Naikade TECOC339,**

**Mehul Lokhande TECOC331,**

**Vijay Dabhade TECOC308,**

**Govind Madankar TECOC332,**

**TE Computer Engineering**

# ABSTRACT

E-Gram Panchayat is aimed at providing government services and secured information to rural citizens. It is an online secure web portal that can be accessed throughout the internet. This System may be used for executing and digitalizing the Gram panchayat activities. The Information about Schemes published by government or any other activities and billing record will be available for users. Documents like Residence Certificate, Land Records, Birth Certificates, Tax Receipts etc. can be issued to the users. Important notifications, notices, village statistical data, Contact details of the important offices and the gram panchayat's committee members will be available on portal, which will ensure easy communication between the government and the public. This system will help to maintain and facilitate easy access to information. With two types of login, namely Admin and user, the security of the portal is maintained. The admin has the authority to maintain and change the data on the portal, approve certificates and user requests through the admin dashboard, thus enabling secure administration. With user friendly interface, the users can enjoy the services of gram panchayat at their ease.

**Keywords:** E-Governance, Digital India, Self Reliant India, E-Government, Web Portal, User Friendly, Secured, Centralized Approach, Rural E-government projects, developing countries.

# INDEX

# List of Tables

# List of Figures

# List of Abbreviations

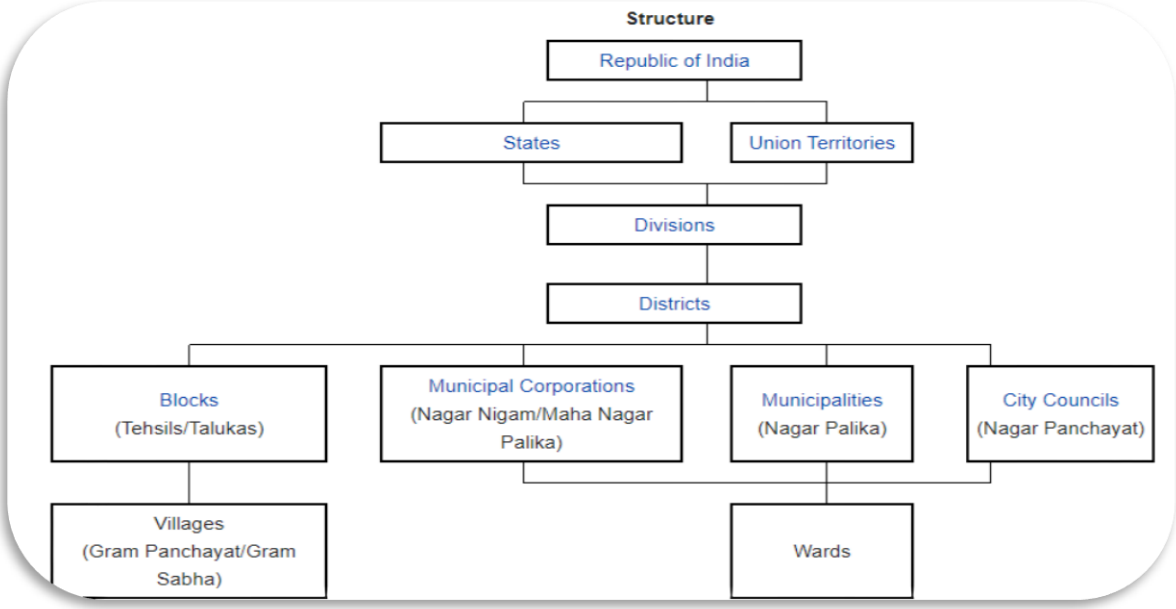
| Abbreviation | Full Form |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

CHAPTER 1

# INTRODUCTION

## 1.1Motivation :

- Panchayati raj originated in 2nd millennium BCE in India during Vedic times. Since Vedic times, the village (gram) in the country is considered as the basic unit for regional self-administration.
- **Gram Panchayat** is a basic village governing institute in Indian villages. It is a democratic structure at the grass-roots level in India. It is a political institute. It acts as cabinet of the village. The gram-sabha work as the general body of Gram Panchayat. The members of the Grampanchayat are elected by the Gram Sabha.
- Gram panchayats, concerned with rural local governments exercise **Administrative, Social and Economic functions in the villages.**
- As cited by our PM, "**Vocal for Local**", it is dire need to focus on local interest, local knowledge and local participation for development of our nation. There are some problems which can be best solved by local authorities only. Such problems need local participation and attention. Hence the formation and reorganization of local bodies have been felt at all times.



1.2 Problem Idea :

- As Mahatma Gandhi pointed that, the village is the functional unit of Indian democracy and if one wants to see development in our country, then our villages need to developed first. He raised the slogans of “Lets go to village”. He Quoted,
  “Let the villages of the future live in our imagination, so that we might one day come to live in them!”
- Citing all the problems faced by the traditional system, we need to develop an integrated system which will cater to needs of villagers and help in making the villages self reliant.
- So we strive to build a system for villagers to be aware of govt. schemes and notify them about revenue taxes of farm, water and home etc.
- The system should be able to narrow down the communication gap between the villagers and the local authorities.
- This project can be scaled to bring the Gram Panchayat at the fingertips of the Villager.
- Our aim is to develop a robust and secure system which will be adopted by govt. for all villages.

CHAPTER 2

# PROBLEM DEFINITION

## 2.1 Problem Definition

Now a day's people in the rural areas have to go to panchayat office in their location to apply and get their certificates provided in that office. It requires a lot of time and may result in work delay. The data in the office has to be maintained manually. There is no security for the data and faults can be encountered during entering the data mainly which require higher calculations. People also face so many problems in their area. They complain to their respective ward members but they may or may not respond quickly. There are many other problems in the present day panchayat raj system.

So our system must tackle these problems and thus below are the solutions:

1. Online and secure payment gateway to pay revenue taxes online.
2. Provide receipt of payment in pdf format.
3. Floating a govt. schemes on website to provide awareness among villagers.
4. Admin dashboard to maintain transaction data of revenue taxes and decide revenue taxes.
5. Villagers can apply for residential certificate online.
6. Villagers will get certificate through email after complete process.

## 2.2 Objectives

- To establish a connection between gram panchayat and villagers for official work, maintaining revenue payment records.
- Villagers can apply for residential certificate online except of visiting gram panchayat office.
- Provide secure and fast payment gateway for villagers to pay their home,farm and water related revenue taxes.
- Floating govt. schemes related to farmers on web application.
- This web application is compatible with all devices.
- Gram Panchayat activities are totally made digital.
- All the Gram Panchayat documents are maintained in online database,

## 2.3 Scope

- **Short-term goals:** Develop a secured web application for revenue payments.
- **Milestones:** Develop a notification floating system and certificate approval module and get approval from govt. to use in gram panchayat.
- **Limitations:** Need of large database, uneducated villagers, security of payments.
- **Top-level requirements:** Large no. of user's document verification for residential certificate, advanced payment gateway.
- **Future Scope:** Develop a Integrated system to function for all gram panchayat offices in disctrict.

CHAPTER 3

# SOFTWARE REQUIREMENT SPECIFICATION

## 3.1 Stakeholders and Actors:

**Stakeholders:** Stakeholders are individuals who either care about or have a vested interest in your project. They are the people who are actively involved with the work of the project or have something to either gain or lose as a result of the project.

- Villagers
- Gram-Panchayat officers.
- Government officials.

**Actors:** An actor is a user or external system with which a system being modeled interacts.

- Villagers as end users.
- Applicants for Residential Certificate.
- Developers as Team members.

## 3.2 Functional Requirements

- **Database:**
  - To store user. credentials(username, password).
  - To store Applicant details.
  - To store feedbacks.

- **Login/Register Page.**
  - To make new account for new users.
  - To give access to users with valid credentials.

- **GUI:**
  - Dashboard.
  - Stastical Charts.
  - Google maps widget.

- **Network:**
  - Server for managing dynamic data.
  - Client-server architecture.

- **Server:**
  - To render user requests.
  - API Routing.

## 3.3 Non-Functional Requirements:

- **Security:**
  - Login  credentials to enter into portal.
  - Provide easy login with oAuth parties like google and facebook.

- **Reliability:**
  - Validations in case of form credentials entered.
  - Must handle load during high traffic.
  - Can run on any device resolution.

- **Maintainability:**
  - Admin Dashboard.
  - Feedback System.

## 4.4 Experimental Details:

### 4.4.1 Technology Stack

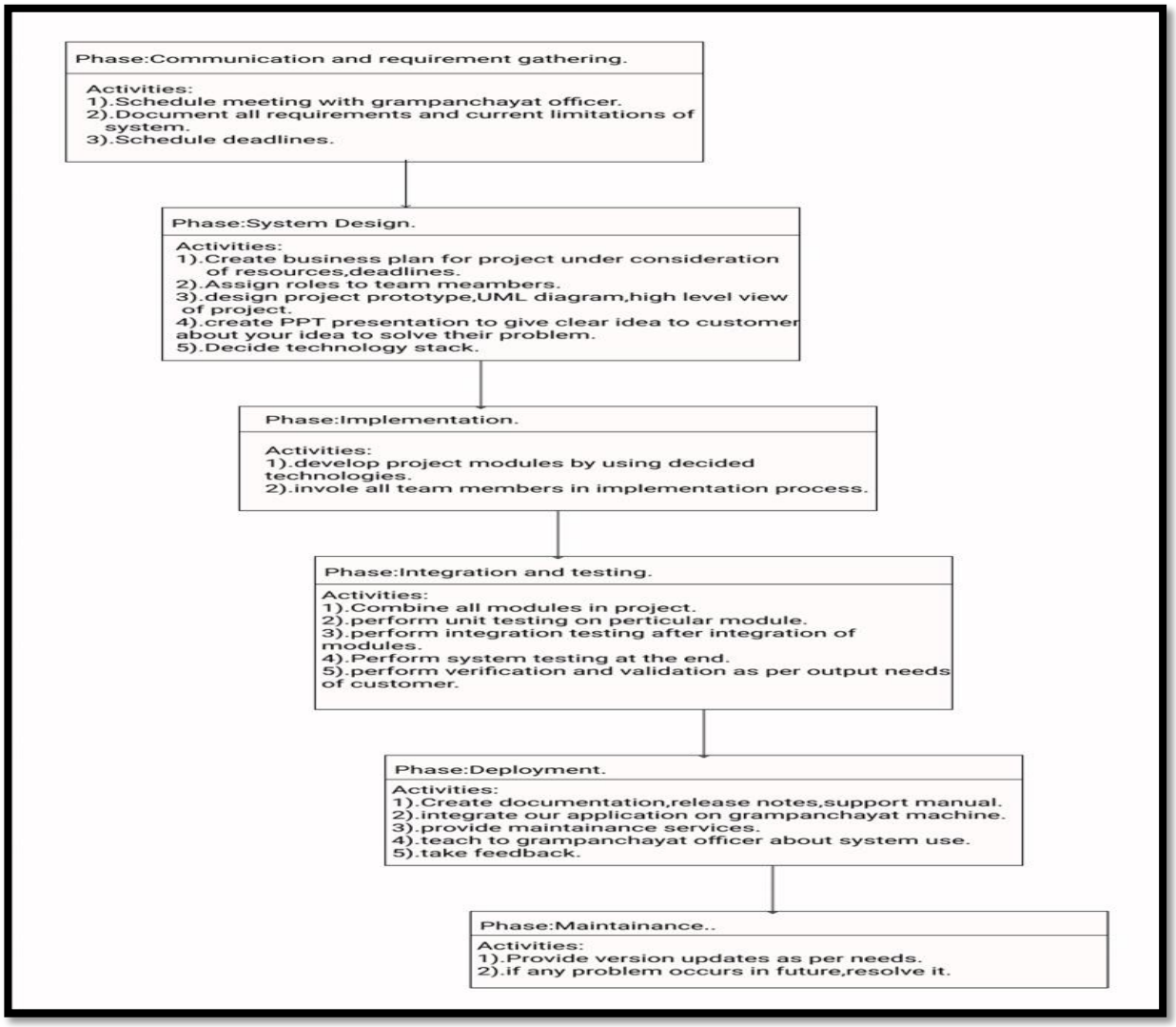
- **Front End**
    - React js: To develop web pages in component form.
    - Bootstrap, JSX, CSS: For responsive design and styling.
- **Database and Hosting**
    - MongoDB: Data will be stored in JSON format.
- **Server side development**
    - Node js: For secure and robust server development.
    - Express js: For API Routing.
    - React Redux: to establish connection with Node js server from client side.
- **Networking**
    - .API Routing:using express js Routing for get/post requests.
    - Client-server Archietecture: developing client using React js and Node js for server development

CHAPTER 4

# PROJECT PLAN

## 5.1 Project Plan

**Software Development Life Cycle**

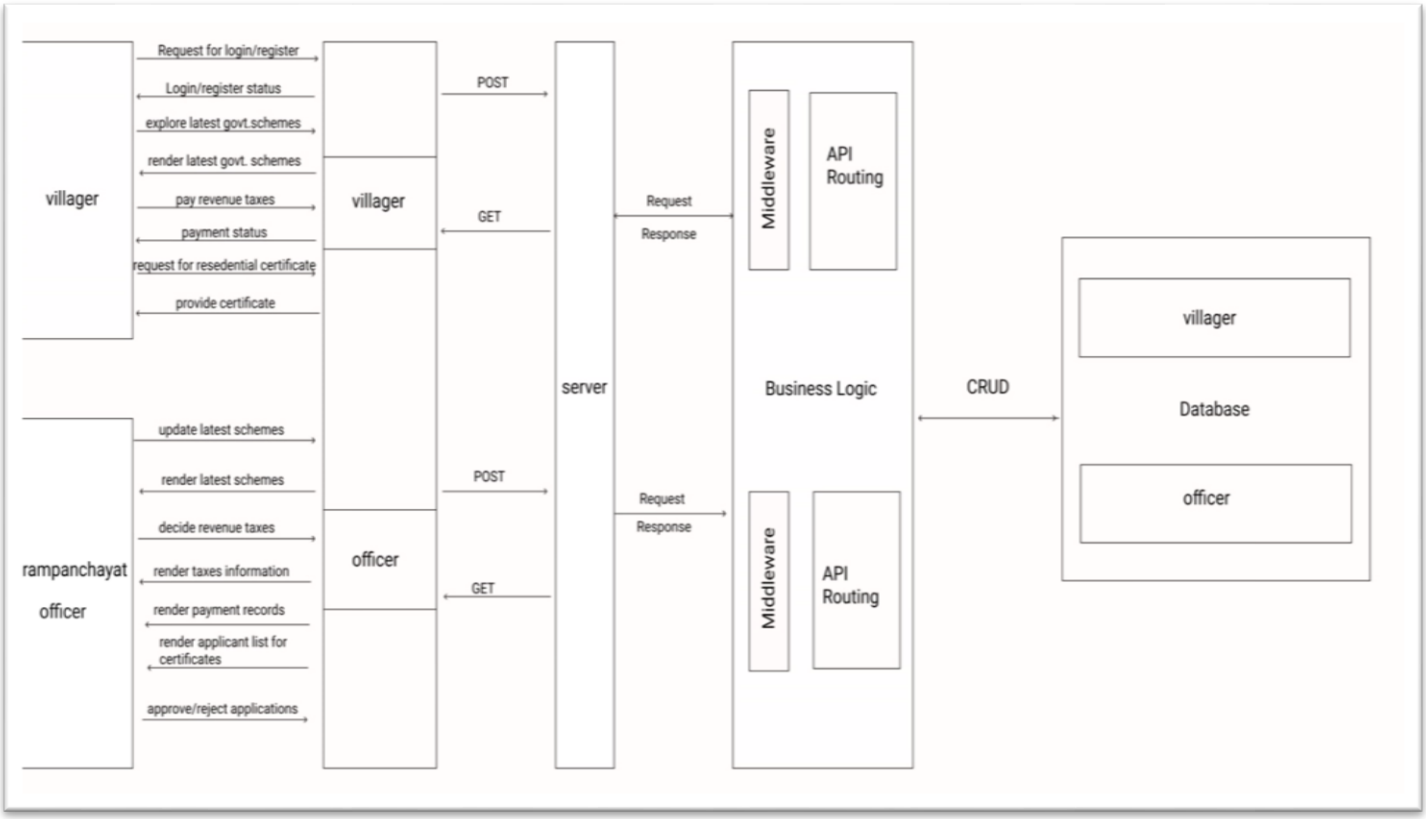


# Project plan of eGram-Panchayat

## Business schedule

| Timeline | technology learning | project research | modelling | construction | testing | version releases |
|---|---|---|---|---|---|---|
| 24 June to 9 july | Rohit:Bootstrap 4 | topic search | | | | |
| | Mehul:html,css,js | topic search | | | | |
| | vijay:Bootstrap | topic search | | | | |
| | govind:html,css | topic search | | | | |
| 10 July to 25 july | Rohit:React JS | information gathering,contact with panchayat officer,decide technology stack. | design of server model, design of database schema, API Routing models. | | | |
| | Mehul:React JS | information gathering, collect information regarding project features. | design of client model, design flow chart of project. | | | |
| | vijay:Bootstrap | collect information about grampanchayat working and data maintainance. | prototype design | | | |
| | govind:html,css | find limitations in current system. | prototype design | | | |
| 26 July to 10 August | Rohit:Node JS,Express JS, Mongodb,SQL | | UML diagrams,business plan and schedule. | Express server with API Routes. | Unit testing of each module. | |
| | Mehul:React JS,MongoDB | | design of functional model of project | CRUD operations on database | | |
| | vijay:React JS | | | Login/Register module. | | |
| | govind:.Javascript,UML models | | PPT for project presentation | | | |
| 11 August to 26 August | Rohit:React Redux | | | connect Login/Register API with redux. | Unit testing of each module. Integration testing | |
| | mehul:MongoDB,Node JS, Express JS | | | develop API Route for Login/ Register. | | |
| | vijay:Node JS,Express JS | | | Database schema for Login/ Register. | | |
| | govind:Bootstrap | | | develop Home,About us page | | |
| 27 August to 11 sept. | Rohit: | | | connect Payment gateway API with redux to client. | Unit testing of each module. Integration testing | |
| | Mehul: | | | develop API Route for Payment Gateway. | | |
| | Vijay:Mongodb | | | Database Schema for Payment Gateway. | | |
| | Govind:React JS | | | Develop Payment gateway page | | |
| 12 sept. to 27 sept. | Rohit: | | | connect schemes API with redux to client. | Unit testing of each module. Integration testing | |
| | Mehul: | | | develop API Route for schemes. | | |
| | Vijay: | | | Database Schema for schemes. | | |
| | Govind: | | | Develop latest schemes page | | |
| 28 sept. to 12 october. | Rohit: | | | connect documents API with redux to client. | Unit testing of each module. Integration testing | |
| | Mehul: | | | develop API Route for documents. | | |
| | Vijay: | | | Database Schema for documents. | | |
| | Govind: | | | Develop documents page | | |
| 13 october to 23 october. | Rohit:SEO | information about best SEO techniques | | Apply SEO on domain. | system testing, verification and validation | Release 1.0 |
| | Mehul:Server Hosting | information about server hosting websites. | | host website on cloud. | | |
| | vijay:server security. | information about server security and algorithms. | | apply security algorithms on server | | |
| | govind:System testing | | | perform system testing,verification and validation | | |

## Higher Level View of project




### 5.2 Effort and Cost Estimation:

**Lines of Code (LOC):** Our Project is basically divided into 7 main modules and each module may be divided into number of sub modules.

**Effort and Development Time**: It took nearly 3 months for planning, designing, implementation, debugging, Testing and Deployment of our project.

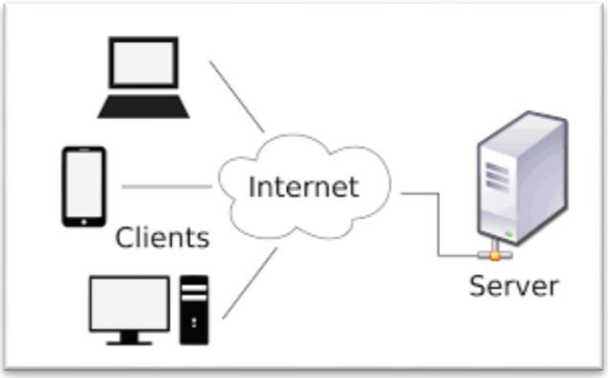
**Number of People:** Our Development team consisted of 4 members, each having proficiency in particular field. But distributed our project work equally such that every one would be able to participate in development and learning of all used technology.

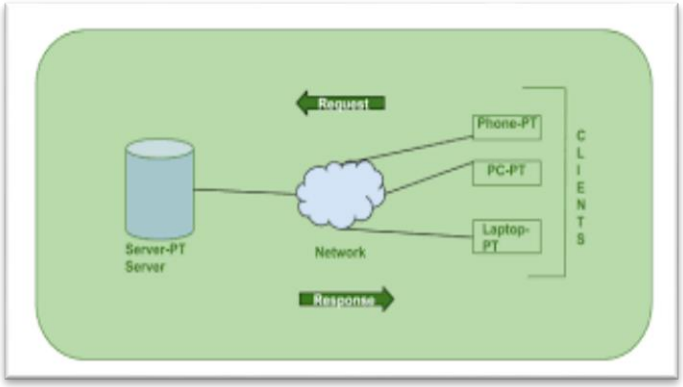
CHAPTER 5

# SYSTEM DESIGN AND IMPLIMENTATION

## .1 System Architecture:

**Client server Architecture**

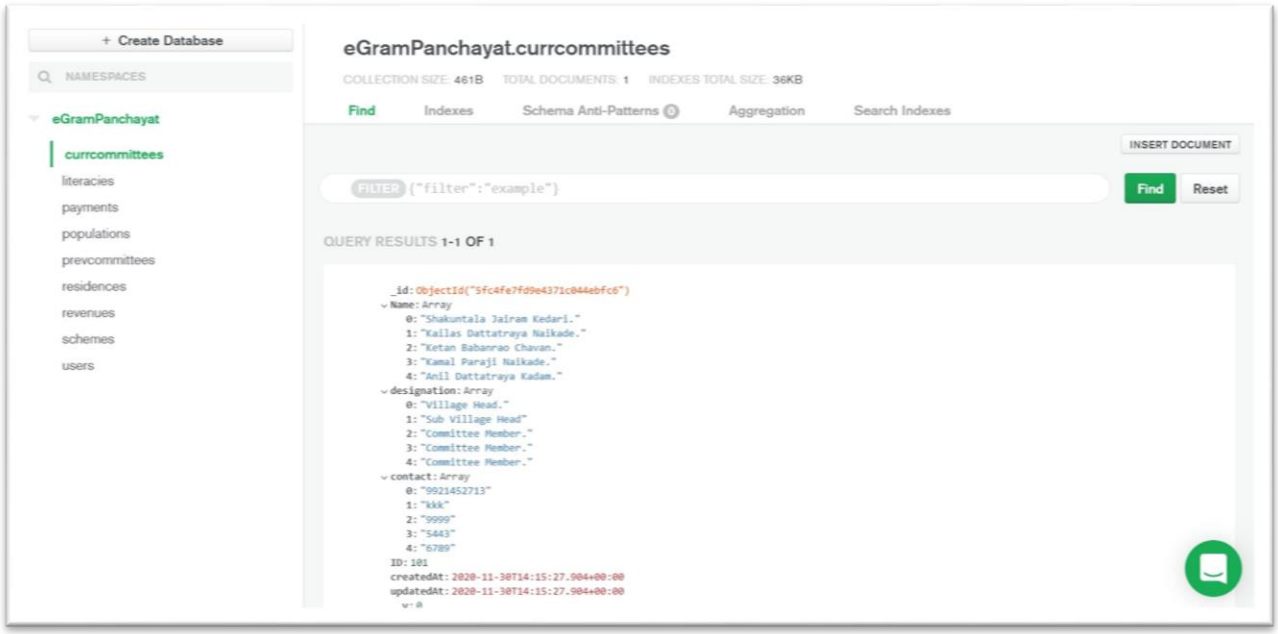


## .2 Database Design:

### Database Schema:

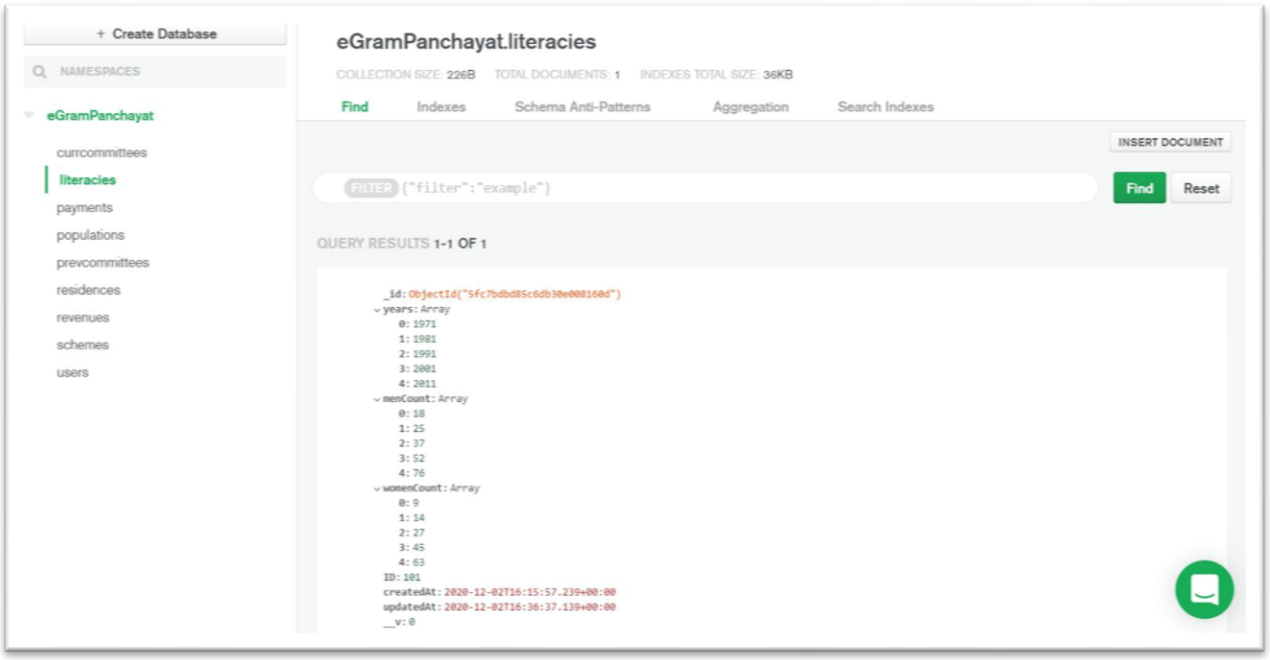
**Current committee Schema**




### Source Code:

```
const currCommitteeSchema = new mongoose.Schema({
   ID:{
      type:Number
   },
   Name: {
      type: [String]
   },
   designation: {
      type: [String]
   },
   contact: {
      type: [String]
   }
}, {
   timestamps: true});
module.exports = mongoose.model('currCommittee', currCommitteeSchema);
```

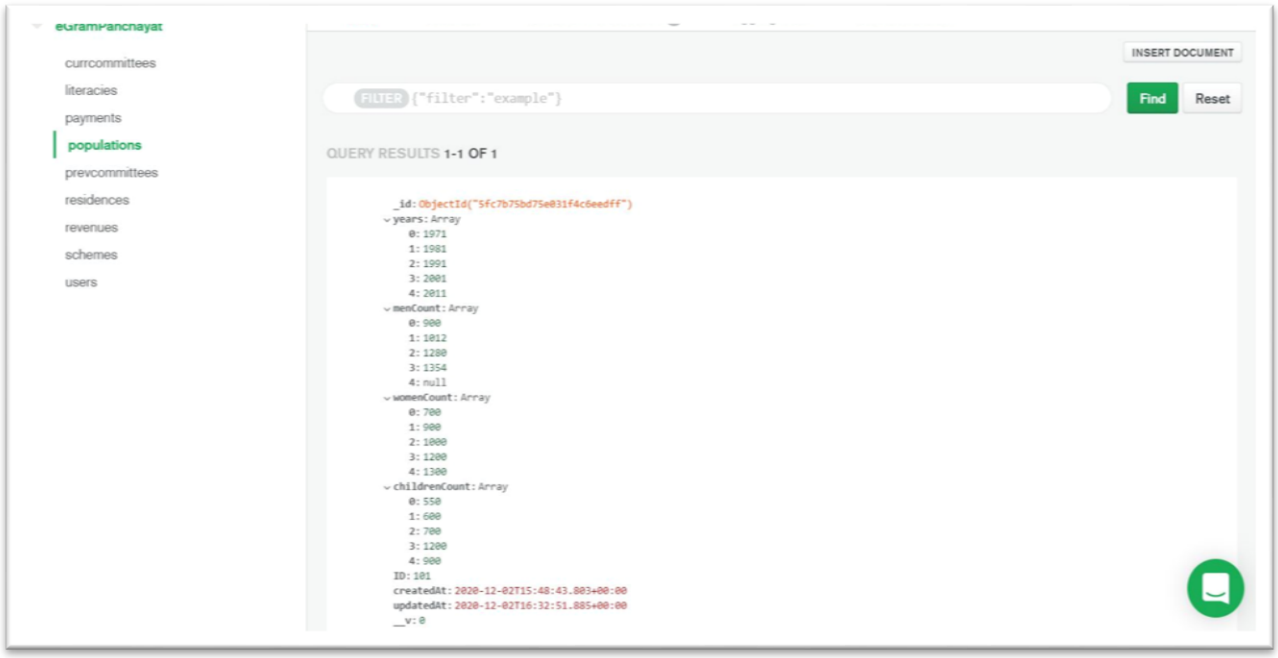
## Literacy Schema




## Source Code:

```
const literacySchema=new mongoose.Schema({
   ID:{
      type:Number
   },
   years:{
      type:[Number],
      required:true,
      max:5,
      min:5
   },
   menCount:{
      type:[Number],
      required:true,
      max:5,
      min:5
   },
   womenCount:{
      type:[Number],
      required:true,
      max:5,
      min:5
   }
},{
   timestamps:true
});
module.exports=mongoose.model('literacy',literacySchema);
```

## Population Schema

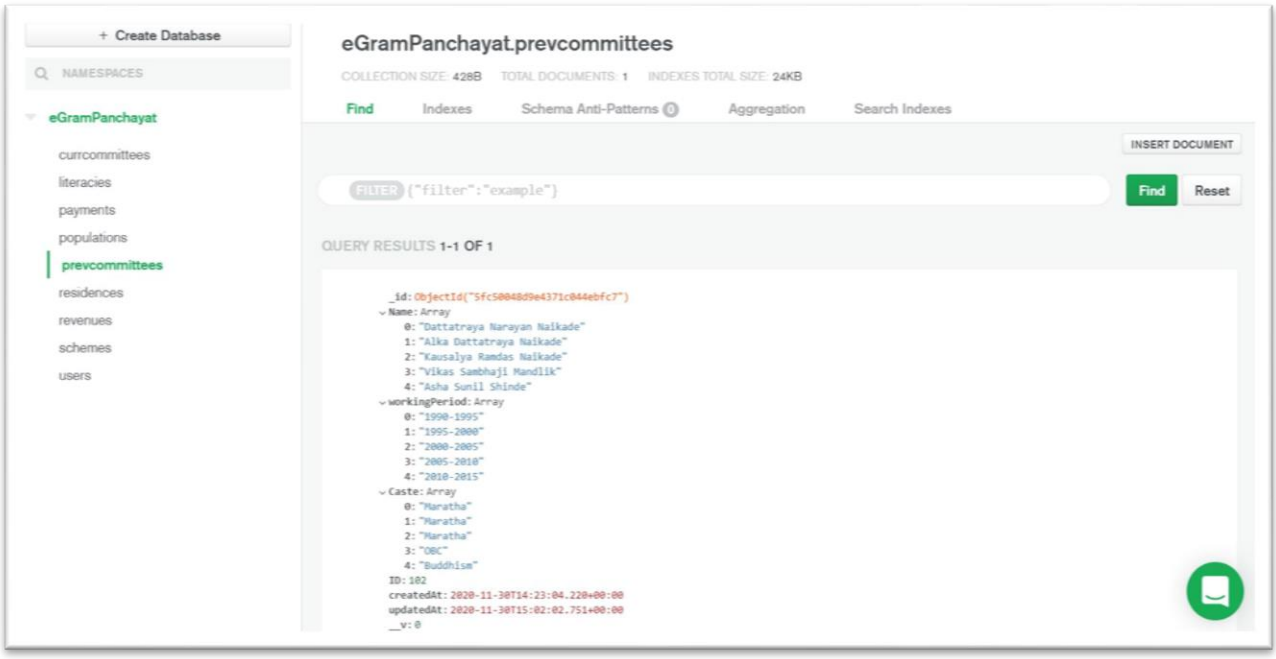



## Source Code:

```
const populationSchema=new mongoose.Schema({
   ID:{
      type:Number
   },
   years:{
      type:[Number],
      required:true,
      max:5,
      min:5
   },
   menCount:{
      type:[Number],
      required:true,
      max:5,
      min:5
   },
   womenCount:{
      type:[Number],
      required:true,
      max:5,
      min:5
   },
```

```
    childrenCount:{
        type:[Number],
        required:true,
        max:5,
        min:5
    },
},{
    timestamps:true
});
module.exports=mongoose.model('population',populationSchema);
```

## Previous committee Schema

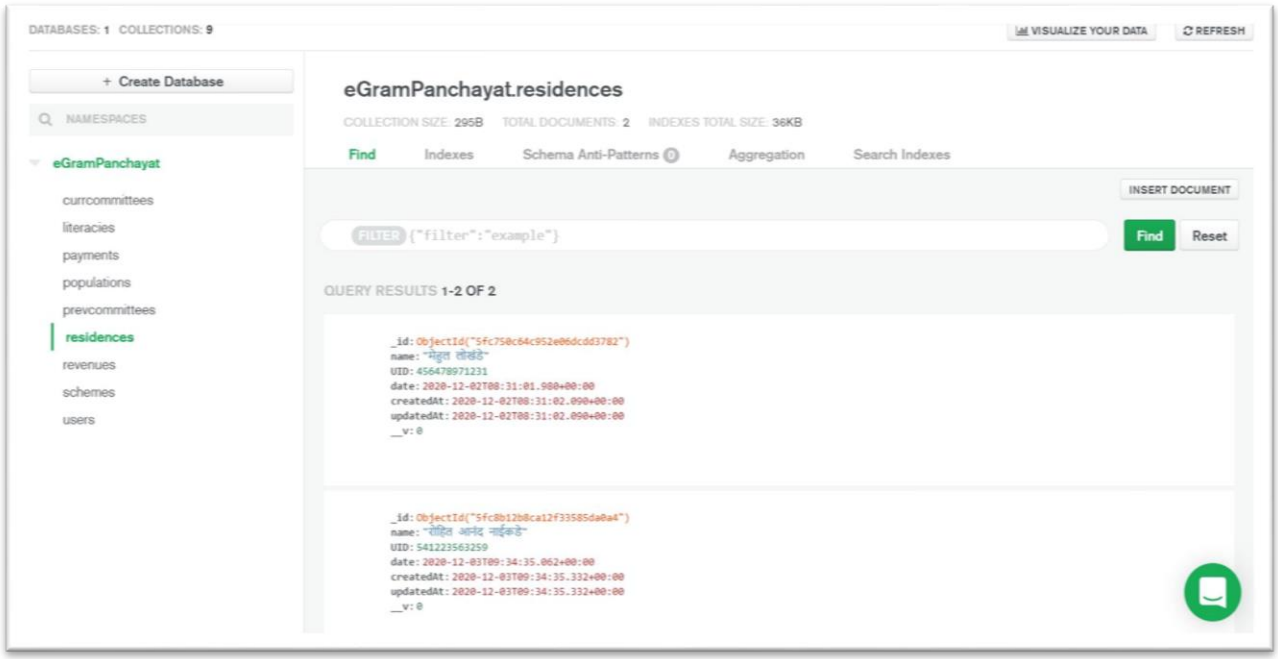



## Source Code:

```
const prevCommitteeSchema = new mongoose.Schema({
    ID:{
        type:Number
    },
    Name: {
        type: [String]
    },
    workingPeriod: {
        type: [String]
    },
    Caste: {
```

```
        type: [String]
    }
}, {
    timestamps: true
});
module.exports = mongoose.model('prevCommittee', prevCommitteeSchema);
```

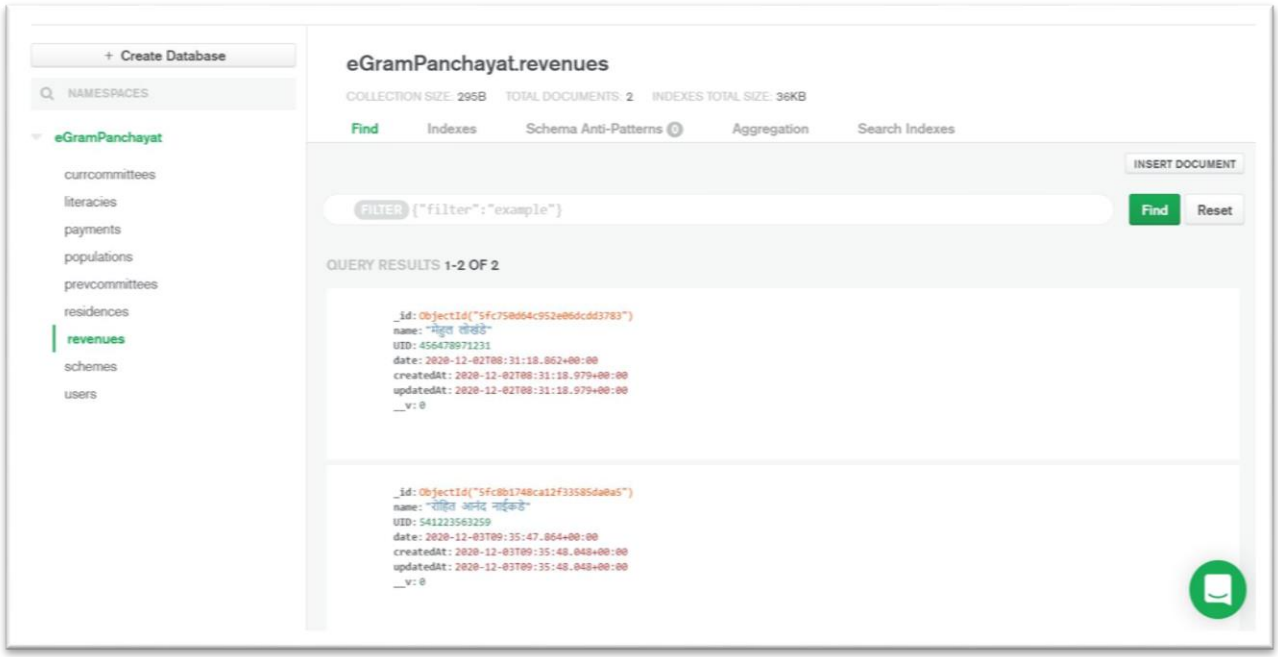
## Residence Schema




## Source Code:

```
const residenceSchema=new mongoose.Schema({
    name:{
        type:String
    },
    date:{
        type:Date
    },
    UID:{
        type:Number
    },
    picture:{
        type:String
    }
},{
    timestamps:true});
module.exports=mongoose.model('residence',residenceSchema);
```

## Revenue Schema

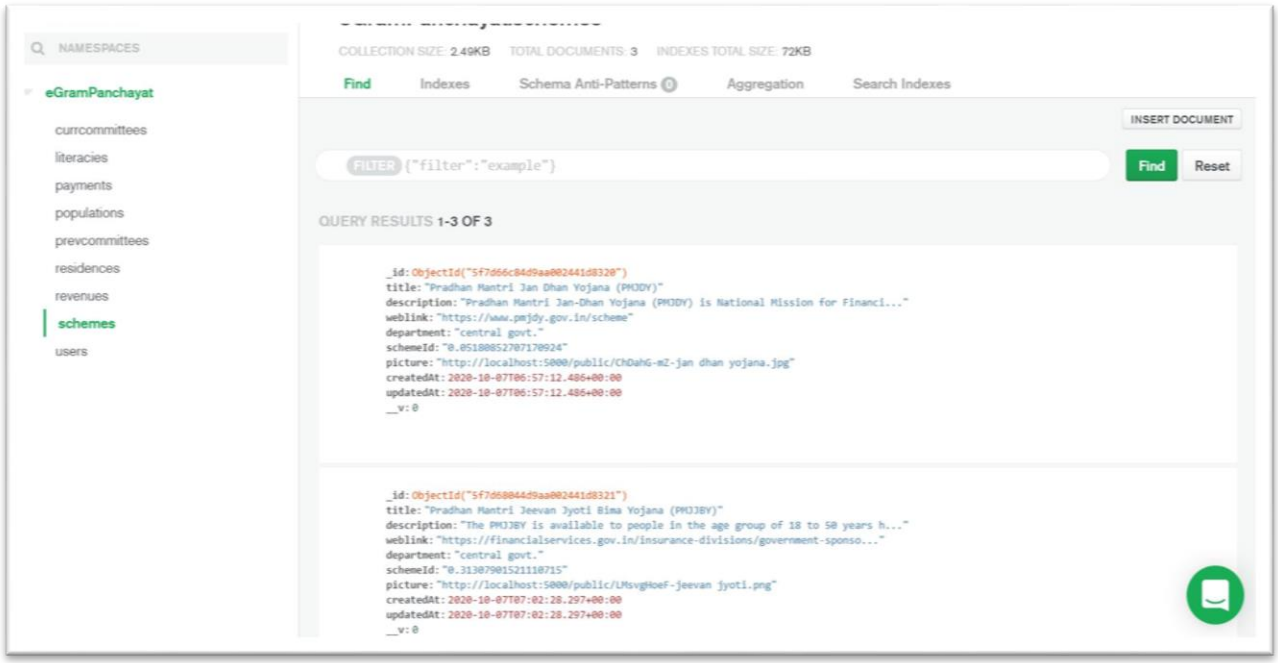



## Source Code:

```
const revenueSchema=new mongoose.Schema({
   name:{
      type:String
   },
   date:{
      type:Date
   },
   UID:{
      type:Number
   },
   home_tax:{
      type:Number
   },
   light_tax:{
      type:Number
   },
   health_tax:{
      type:Number
   },
   water_tax:{
      type:Number
   },
   penalty_tax:{
```

```
    type:Number
  },
  warrant_tax:{
    type:Number
  }
},{
  timestamps:true
});

module.exports=mongoose.model('revenue',revenueSchema);
```

## Schemes Schema

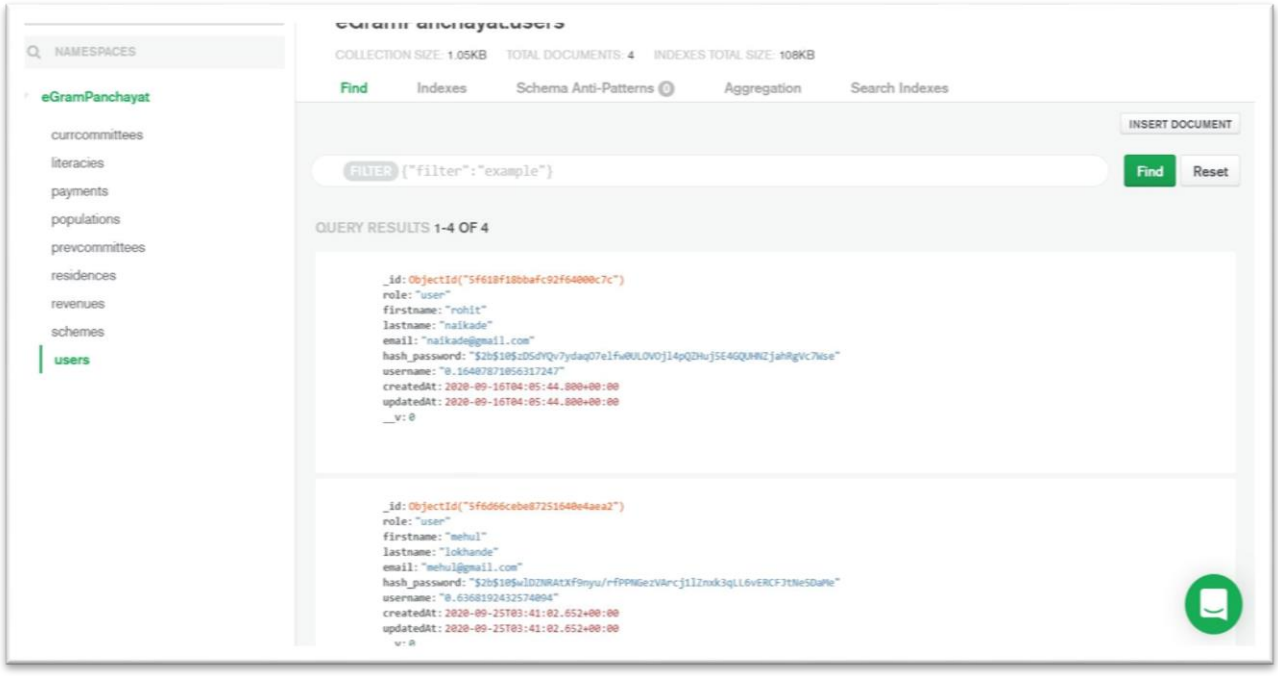



## Source Code:

```
const schemeSchema=new mongoose.Schema({
  schemeId:{
    type:String,
    required:true,
    unique:true
  },
  title:{
    type:String,
    required:true
  },
```

```
  description:{
    type:String,
    required:true
  },
  weblink:{
    type:String,
    required:true,
    lowercase:true
  },
  department:{
    type:String,
    required:true
  },
  picture:{
    type:String
  }
},{
  timestamps:true
});

module.exports=mongoose.model('Scheme',schemeSchema);
```

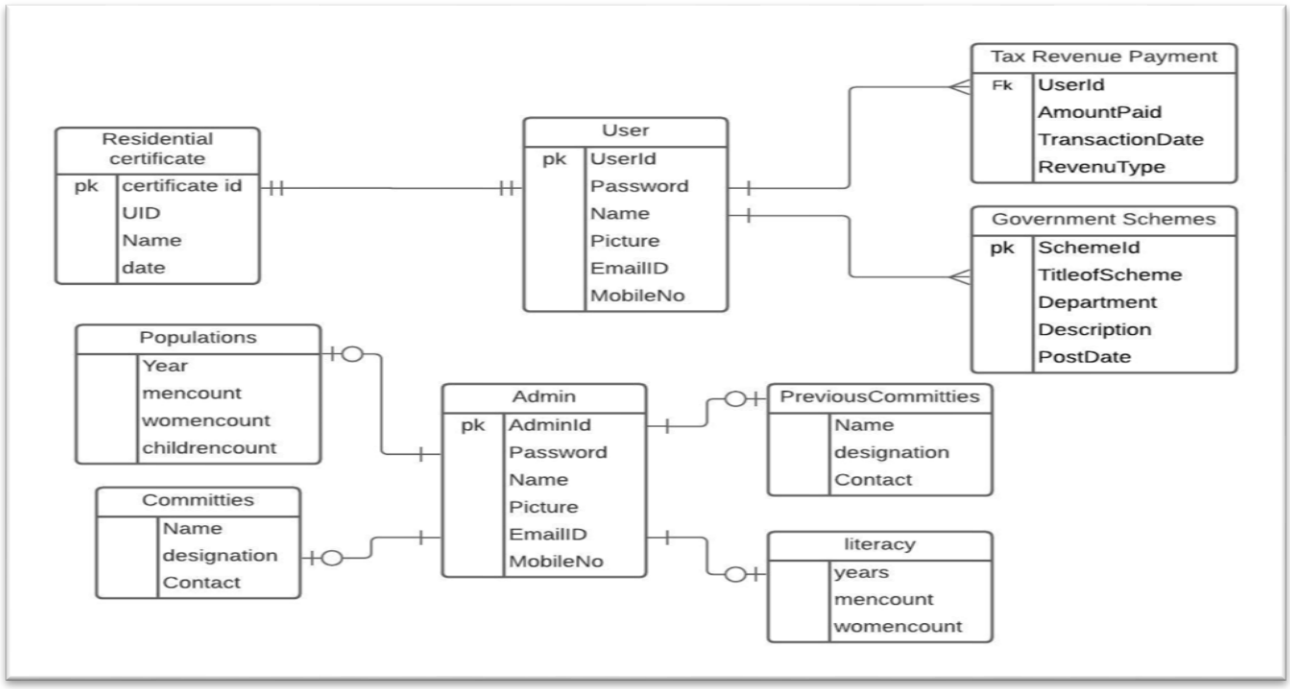
**Users Schema**



**Source Code:**

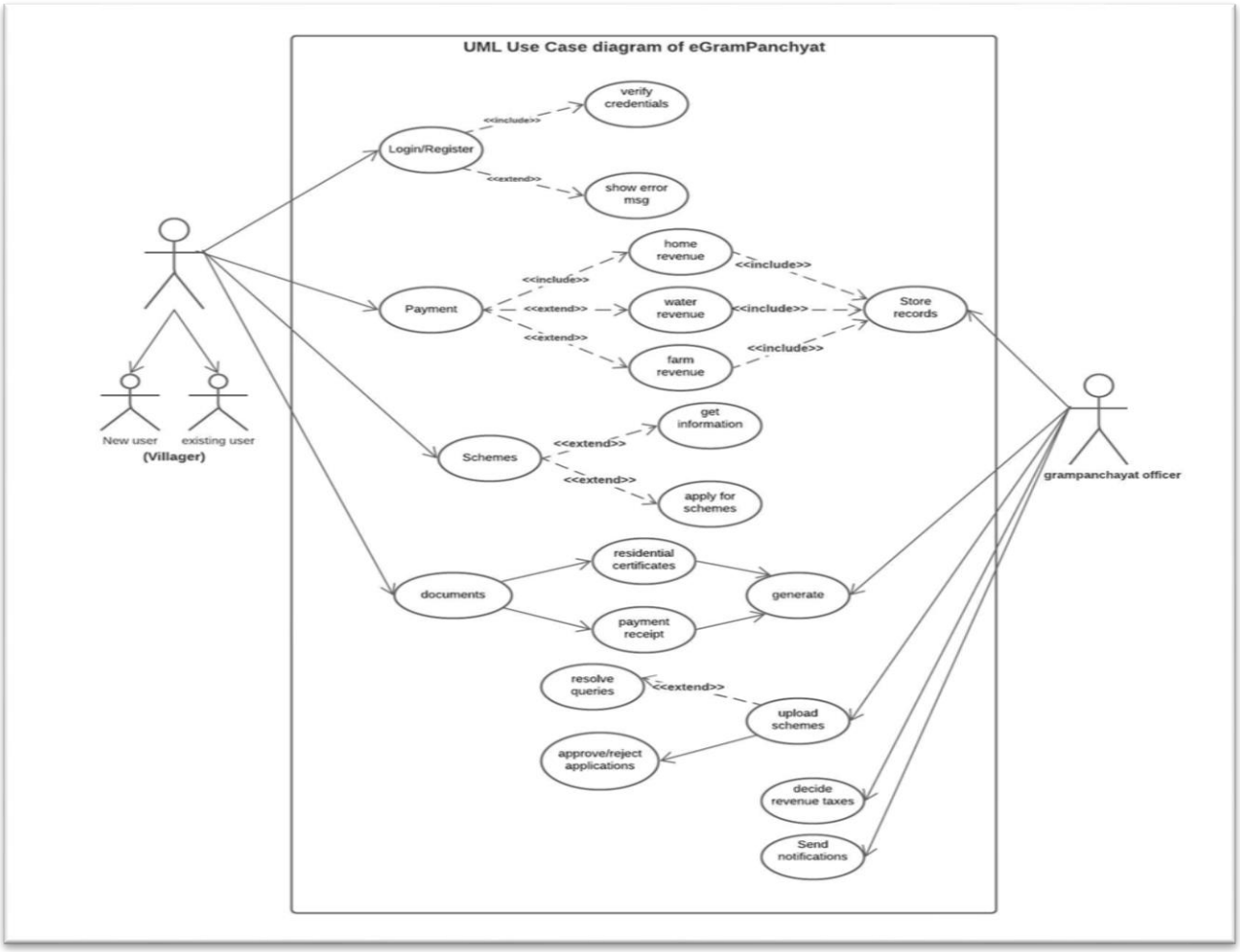
```
const userSchema = new mongoose.Schema(
   {
    email: {
      type: String,
      trim: true,
      required: true,
      // unique: true,
      lowercase: true
    },
    number:{
      type: String,
      trim: true,
      required: true
    },
    name: {
      type: String,
      trim: true,
      required: true
    },
    hashed_password: {
      type: String,
      required: true
    },
    salt: String,
    role: {
      type: String,
      default: 'user'
    },
    resetPasswordLink: {
      data: String,
      default: ''
    }
   },
   {
    timestamps: true }
 );
```

# ER Diagram

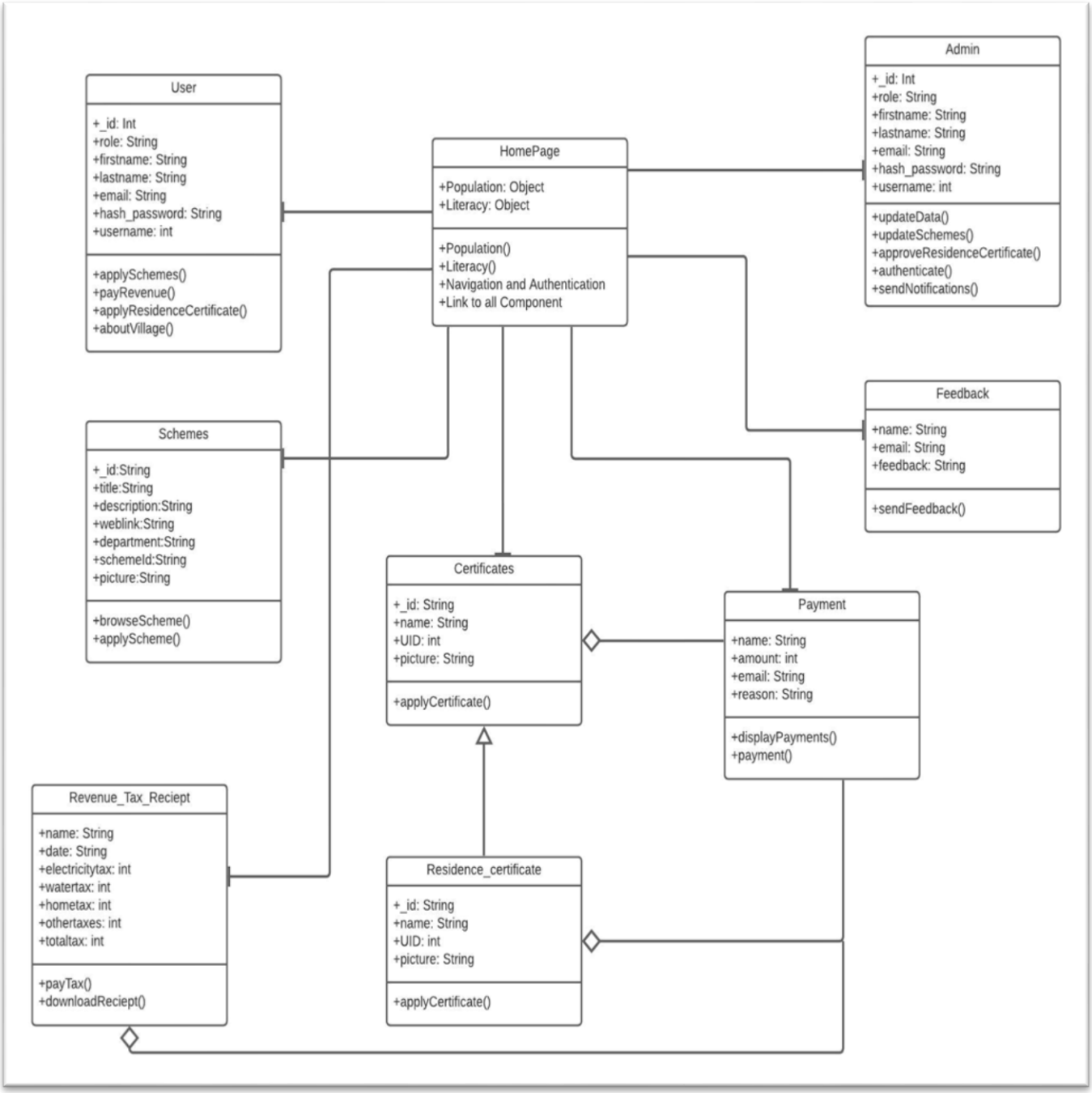

Residential certificate
pk
certificate id
UID
Name
date
User
pk
UserId
Password
Name
Picture
EmailID
MobileNo
Tax Revenue Payment
Fk
UserId
AmountPaid
TransactionDate
RevenuType
Government Schemes
pk
SchemeId
TitleofScheme
Department
Description
PostDate
Populations
Year
mencount
womencount
childrencount
Committies
Name
designation
Contact
Admin
pk
AdminId
Password
Name
Picture
EmailID
MobileNo
PreviousCommitties
Name
designation
Contact
literacy
years
mencount
womencount

## 6.3 UML Diagrams:

### 6.3.1 UML Use Case Diagram

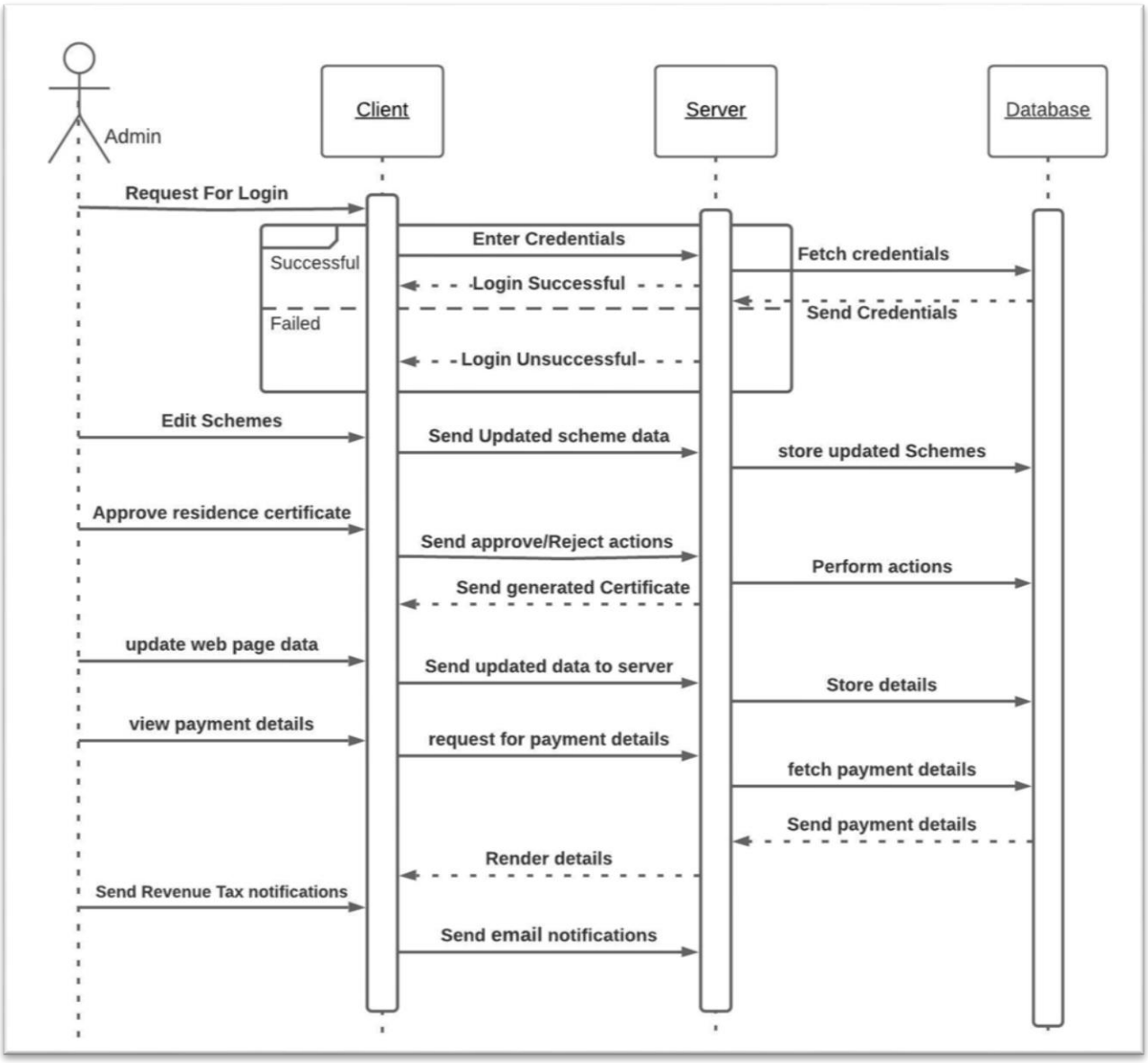


### 6.3.2 Class Diagram

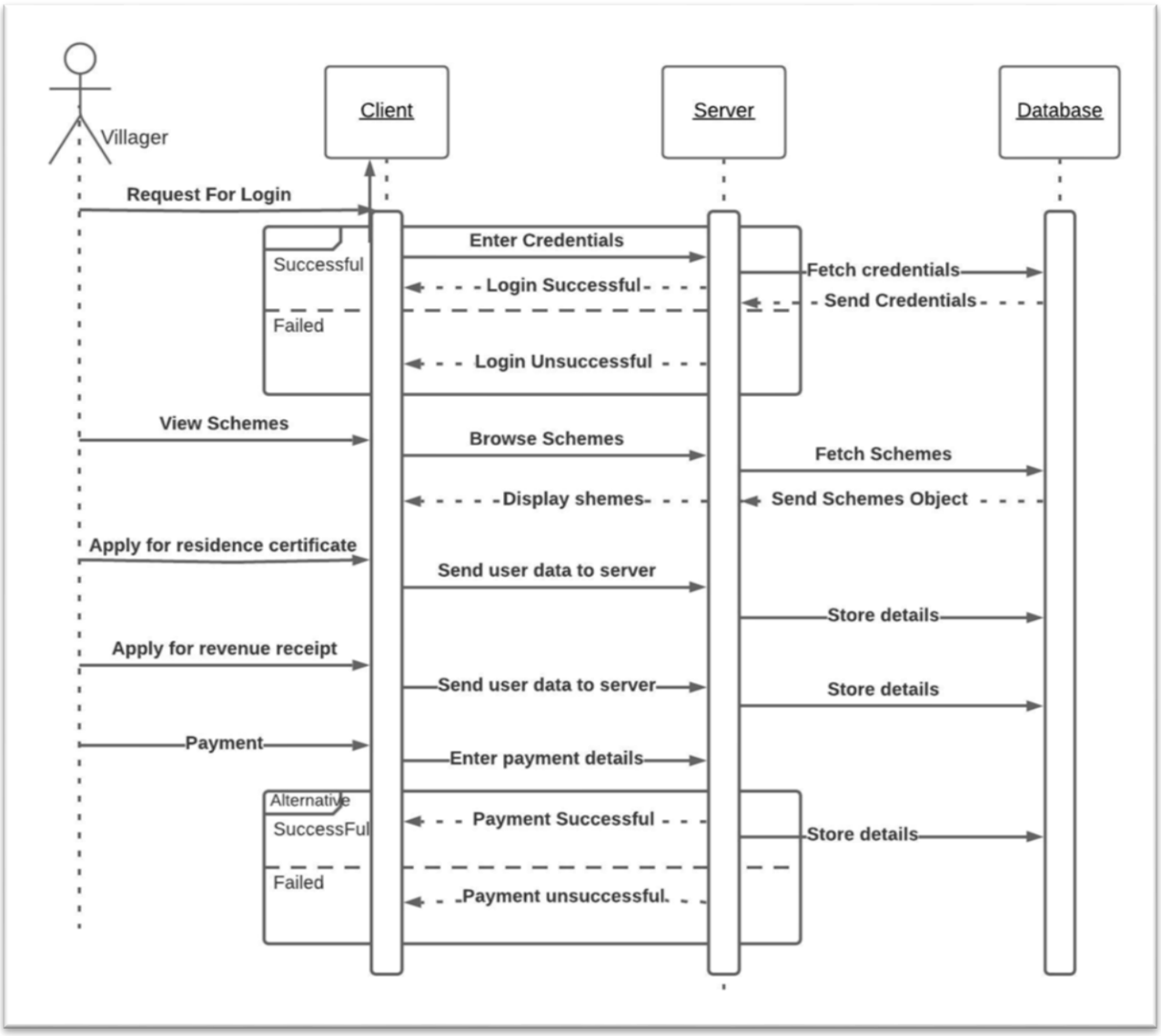


## 6.3.2 Sequence Diagram

## A) Admin:

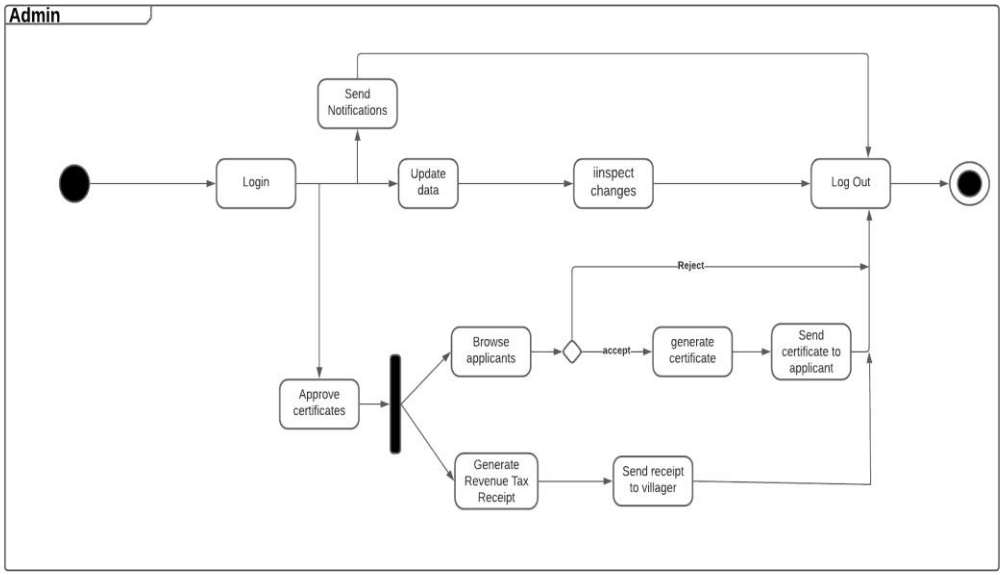


## B) User(Villager):

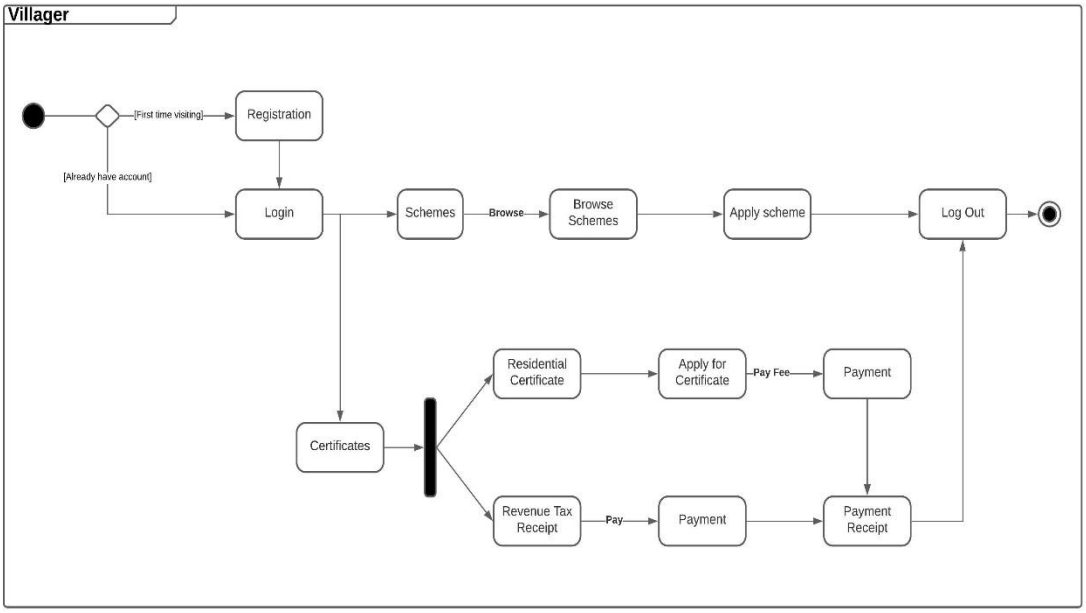


### 6.3.3 State Diagram

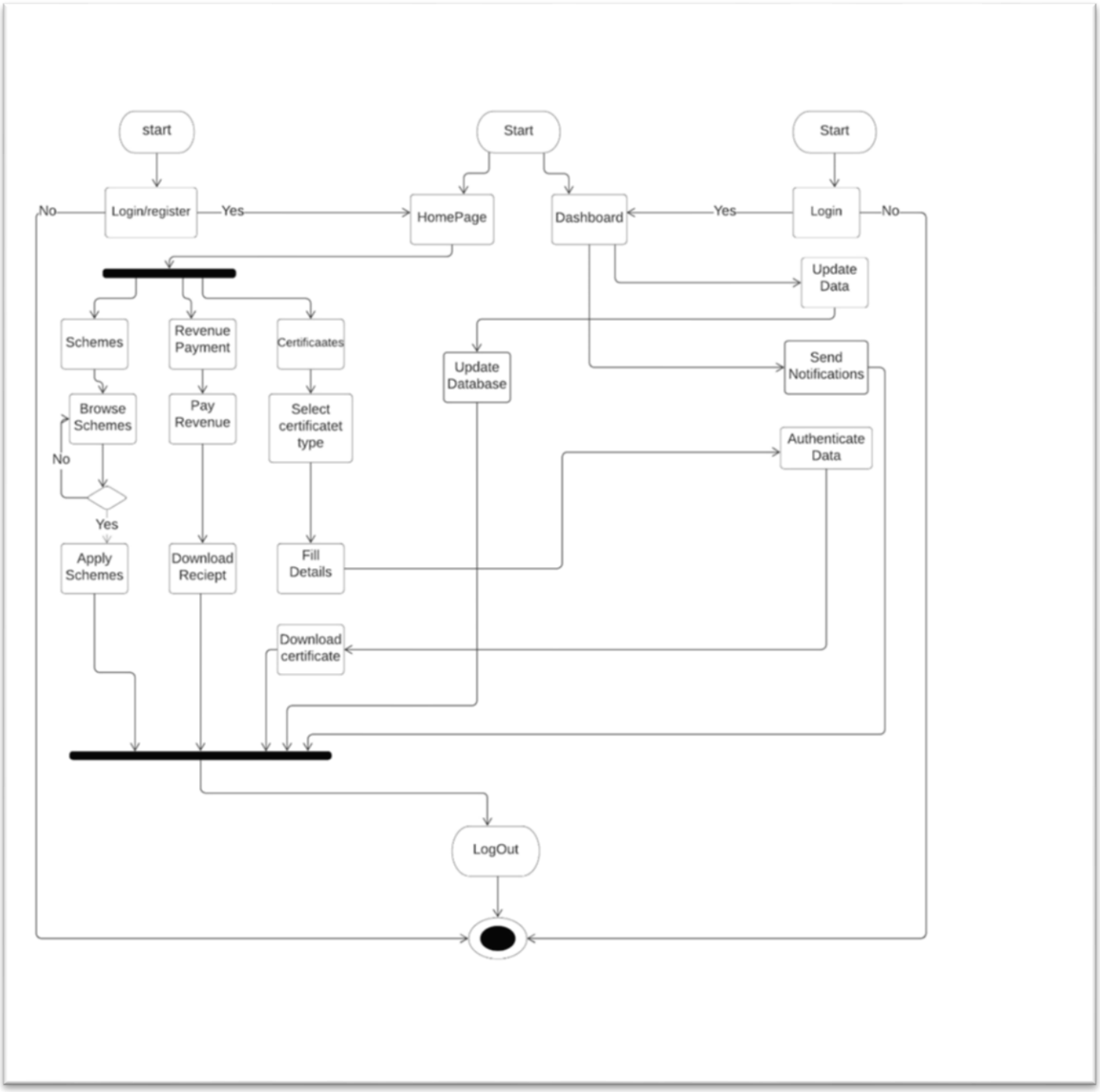
## A) State diagram for Admin




## B) State diagram for User(villager):



## 6.3.5 Activity Diagram



CHAPTER 6

# MODULE DESCRIPTION

## 6.1 Module Description:

- **Login/Register Module:**

For user convenience, we have oAuth login functionality with google and facebook to login into our system. Before login, users have to register him/her into our system with name, mobile number, and email address. After registration, one account activation link is sent on email id. After activation, records will be stored into database.

If you are registered already, you can sign in with email address and password. You can use Google and Facebook oath parties for sign in.

- **Home module:**

We are visualizing population and literacy data of last five surveys using react-charts2 library. Also we have Google maps API with village location. Also with have responsive header for navigation. Footer has quick links for easy navigation through system. Some social media links also there to communicate with development team.

- **About village module:**

In this tab, we have responsive and full-screen carousal with images of tourist places, temples near village. This carousal is showing beauty and attractiveness of village. We are showing information about last 5 village heads, contact details of current gram panchayat committee. Villager can easily contact with village committee with this contact details.

- **Schemes Module:**

On this tab, we are displaying government schemes which are beneficiary for farmers, villagers. From this tab, villager can get information about respective government schemes and by clicking on apply button, it will redirect you to official page of respective scheme where you can apply for that scheme.

- **Payment Module:**

From this tab, you can pay money of residence certificate, revenue tax for gram panchayat using paytm payment gateway. Here you have to fill your name, mobile number, email address, reason to pay amount and payment amount. We are using paytm for secure and reliable payment transactions.

  - **Documents Module:**

    In this tab, we are providing three types of documents as residence certificate, revenue tax receipt and payment receipt respectively. From this module, you can apply for respective document.

  - **Residence certificate:**

    From first dropdown option, you can apply for residence certificate with using name of applicant and Adhar Number. After apply, application is submitted and it is visible for admin in admin center. Admin will approve/reject application after verification of details.

  - **Revenue tax receipt:**

    From second dropdown option, you can apply for revenue tax receipt using name of applicant and Adhar Number. After apply, application is submitted and it is visible for admin in admin center. Admin will approve/reject application after verification of details. Admin will fill tax details of applicant and generate tax receipt and send back to respective villager.

  - **Payment Receipt:**

    From third dropdown option, you can apply for payment receipt using name of applicant and Adhar Number. After apply, application is submitted and it is visible for admin in admin center. Admin will approve/reject application after verification of

details. Admin will fill payment details of applicant and generate tax receipt and send back to respective villager.

- ## Admin Center Module:

From admin tab, admin can modify dynamic data rendered on web pages; approve/reject applications of residence certificate.

- **Home section:**

  In this section, admin can update population and literacy related data which is visualized on home page using react-charts library.

- **About village section:**

  In this section, admin can update contact details of current committee of gram panchayat, details of previous gram panchayat village heads. These details are rendered on about village tab.

- **Schemes section:**

  In this section, admin can upload new schemes, delete previous uploaded schemes. These schemes are rendered on schemes page.

- **Residence certificate section:**

  In this section, admin can approve/reject applications of residence certificate. After approval of application, it will generate residence certificate of applicant in PDF format. Rejected application gets deleted from database.

- **Revenue tax receipt section:**

  In this section, admin can approve/reject applications of revenue tax receipt. Admin should fill revenue tax details of applicant. After approval of application, it will generate revenue tax receipt of applicant in PDF format. Rejected application gets deleted from database.

- **Payment receipt section:**

  In this section, admin can approve/reject applications of payment receipt. After payment from payment gateway, payment record is generated in database and rendered in admin center's payment tab. After approval of application, it will generate

payment receipt of applicant in PDF format. Rejected application get deleted from database.

- **About us Module:**

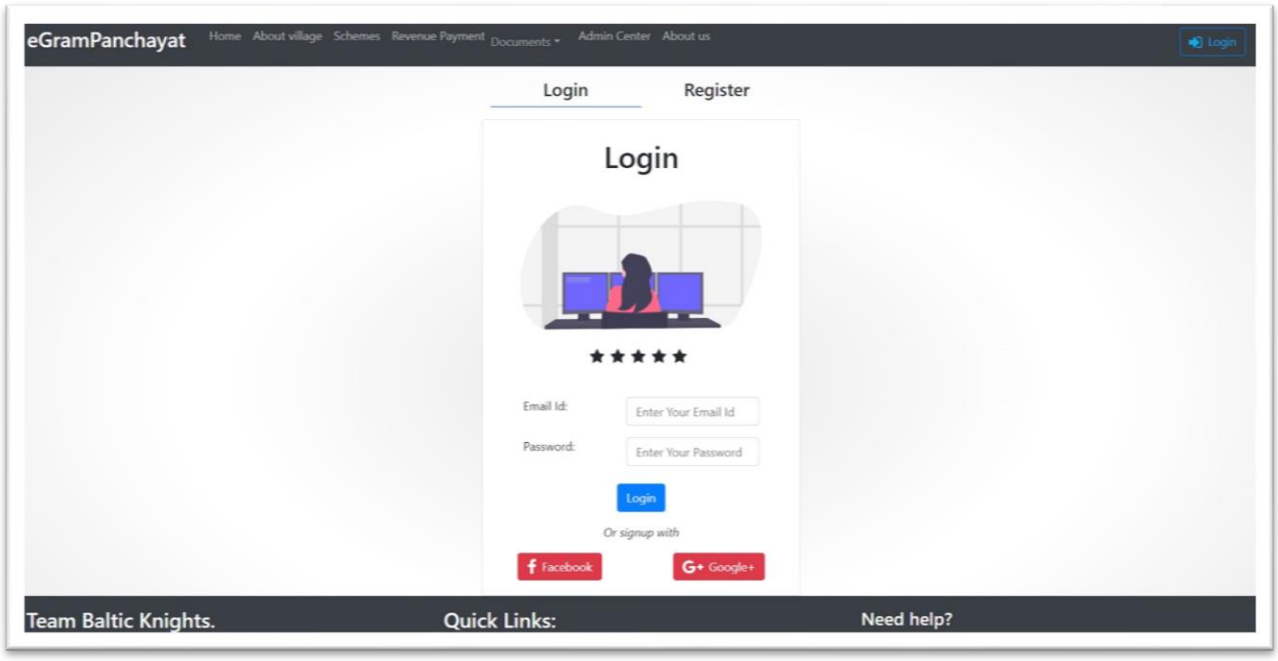
From this tab, villager can contact with development team on social media accounts. We have provided contact details of team members.

CHAPTER 7

# EXPERIMENTS RESULT AND ANALYSIS

## 8.1 GUI Snapshots:

### Login Module




## Source Code:

```
import React,{useState} from "react";
import loginImg from "./login.svg";
import './login.css';
import axiosInstance from '../../helpers/axios';
import { Control, LocalForm, Errors } from 'react-redux-form';
```

```
import { Row, Col, Card, Container, Button, Form } from 'react-bootstrap';
import { FadeTransform } from 'react-animation-components';
import { authenticate, isAuth } from '../../helpers/auth';
import { store } from 'react-notifications-component';
import 'react-notifications-component/dist/theme.css';
import 'animate.css';
import history from '../../helpers/history';
const required = (val) => val && val.length;
const minLength = (len) => (val) => val && (val.length >= len);
const validEmail = (val) => /^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$/i.test(val);
const Login = () => {
  const [formData, setFormData] = useState({
    email: '',
    password1: '',
    textChange: 'Sign In'
  });
  const { email, password1, textChange } = formData;
  const handleChange = text => e => {
    setFormData({ ...formData, [text]: e.target.value });
  };
  const handleSubmit = (values) => {
    if (email && password1) {
      setFormData({ ...formData, textChange: 'Submitting' });
      axiosInstance.post("user/login", {
        email, password:password1
      })
        .then(res => {
          authenticate(res, () => {
            setFormData({
              ...formData,
              email: '',
              password1: '',
              textChange: 'Submitted'
            });
            console.log(isAuth())
            isAuth() && isAuth().role === 'admin'
              ? history.push('/admin')
              : history.push('/');
              store.addNotification({
                title: `${res.data.user.name},welcome back!`,
```

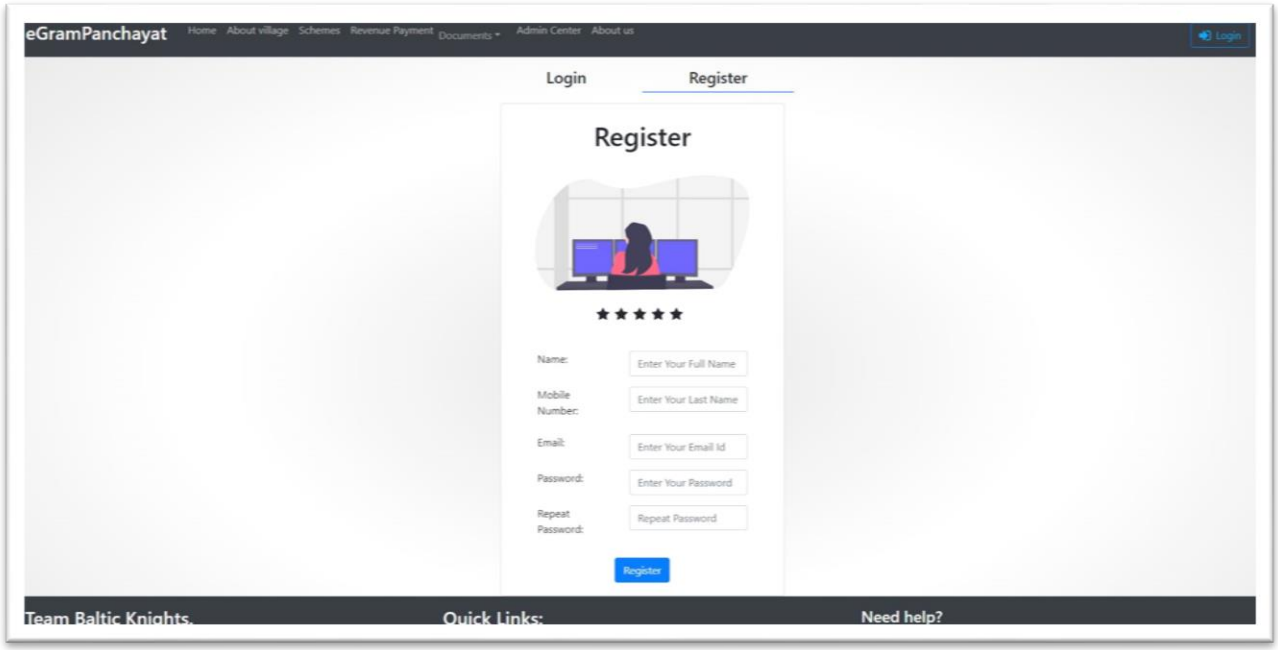
```
            message: 'Now you have privileges to explore!',
            type: "success",
            container: 'top-right',
            animationIn: ["animated", "fadeIn"],
            animationOut: ["animated", "fadeOut"],
            dismiss: {
              duration: 3000,
              showIcon: true
            }
          })
        });
      })
      .catch(err => {
        setFormData({
          ...formData,
          email: '',
          password1: '',
          textChange: 'Sign In'
        });
        console.log(err);
        store.addNotification({
          title: `${err.response.data.errors}`,
          message: 'Try again!',
          type: "danger",
          container: 'top-right',
          animationIn: ["animated", "fadeIn"],
          animationOut: ["animated", "fadeOut"],
          dismiss: {
            duration: 3000,
            showIcon: true
          }
        })
      });
  }
}
return (
  <Container fluid>
    <Row className="justify-content-md-center">
      <Col className='col-md-5 mt-3' >
        <FadeTransform
```

```
          in
          transformProps={{
            exitTransform: 'scale(0.5) translateY(-50%)'
          }}>
          <Card className="frm">
            <div className="text-center mt-4 mb-4"><h1 className="">Login</h1></div>
            <Card.Img varient="top" className="mt-1 col-md-10 col-sm-10 offset-md-
1" src={loginImg}></Card.Img>
            <div className="text-center mt-4 mb-4"><span className="fa fa-star fa-lg mr-
2"></span><span className="fa fa-star fa-lg mr-2"></span><span className="fa fa-star fa-
lg mr-2"></span><span className="fa fa-star fa-lg mr-2"></span><span className="fa fa-
star fa-lg mr-2"></span></div>
            <Card.Body>
              <LocalForm onSubmit={(values) => handleSubmit(values)}>
                <div className="form-group">
                  <Row><Col className="col-md-4 offset-md-1">
                    <Form.Label>Email Id:</Form.Label></Col>
                    <Col className="col-md-6">
                      <Control.text
                        autoComplete="off"
                        model=".email"
                        id="email"
                        className="form-control"
                        placeholder="Enter Your Email Id"
                        value={email}
                        onChange={handleChange('email')}
                        validators={{
                          required, validEmail
                        }}
                      />
                      <Errors
                        className="text-danger"
                        model=".email"
                        show="touched"
                        messages={{
                          required: 'Required ',
                          validEmail: 'Enter a valid email address!'
                        }}
                      />
                    </Col></Row>
```

```
            </div>
            <Row><Col>
              <div className="form-group">
                <Row><Col className="col-md-4 offset-md-
1"><Form.Label>Password:</Form.Label></Col>
                  <Col className="col-md-6">
                   <Control.text
                    autoComplete="off"
                    model=".password"
                    id="password"
                    className="form-control"
                    placeholder="Enter Your Password"
                    value={password1}
                    onChange={handleChange('password1')}
                    validators={{
                     required, minLength: minLength(8)
                    }}
                   />
                   <Errors
                    className="text-danger"
                    model=".password"
                    show="touched"
                    messages={{
                     required: 'Required ',
                     minLength: 'Password should be greater than 8 characters!'
                    }}
                   />
                   <p className="float-right font-italic mt-2">Forgot Password?</p>
                  </Col></Row>
              </div></Col></Row>
            <div className="text-center mt-2">
              <Button variant="primary" type="submit">Login</Button>
            </div>
          </LocalForm>
          <div className="text-center mt-3"><i>Or signup with</i></div>
          <div className="text-center mt-3">
            <Button variant="danger" className="mr-
5" type="submit"><span className="fa fa-facebook fa-lg mr-2"></span>Facebook</Button>
            <a href="/auth/google">
              <Button variant="danger" className="ml-5" type="submit">
```

```
                <span className="fa fa-google-plus fa-lg mr-2"></span>
                  Google+</Button>
              </a>
            </div>
          </Card.Body>
        </Card></FadeTransform>
      </Col>
    </Row>
  </Container>
 );
}
export default Login;
```

## Register Module




## Source Code:

```
import React, { useState } from "react";
import loginImg from "./login.svg";
import './login.css';
import axiosInstance from '../../helpers/axios';
import { Control, LocalForm, Errors } from 'react-redux-form';
import { Row, Col, Card, Container, Form, Button } from 'react-bootstrap';
import { FadeTransform } from 'react-animation-components';
```

```
import { store } from 'react-notifications-component';
import 'react-notifications-component/dist/theme.css';
import 'animate.css';
const required = (val) => val && val.length;
const maxLength = (len) => (val) => !(val) || (val.length <= len);
const minLength = (len) => (val) => val && (val.length >= len);
const validEmail = (val) => /^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$/i.test(val);
const validText = (val) => /^[A-Za-z]+$/i.test(val);
const validMobile=(val)=> /^((\+){1}91){1}[1-9]{1}[0-9]{9}$/i.test(val);
const validName=(val)=> /^([a-zA-Z0-9]+|[a-zA-Z0-9]+\s{1}[a-zA-Z0-9]{1,}|[a-zA-Z0-
9]+\s{1}[a-zA-Z0-9]{3,}\s{1}[a-zA-Z0-9]{1,})$/i.test(val);
const Register = () => {
  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [password1, setPassword1] = useState("");
  const [password2, setPassword2] = useState("");
  const [number, setNumber] = useState("");

  const handleSubmit = (values) => {
    console.log(name, email,number, password1, password2);
    const regData = {
      name,
      email,
      number,
      password:password1,
      number
    }
    if (password1 === password2) {
      axiosInstance.post('user/signup', regData)
        .then(res => {
          if(res.data.error){
            store.addNotification({
              title: `${res.data.error}`,
              message: 'Try again with another account!',
              type: "danger",
              container: 'top-right',
              animationIn: ["animated", "fadeIn"],
              animationOut: ["animated", "fadeOut"],
              dismiss: {
                duration: 3000,
```

```
            showIcon: true
          }
        })
      }else{
        store.addNotification({
          title: `${res.data.message}`,
          message: 'Check your mailbox for account activation!',
          type: "success",
          container: 'top-right',
          animationIn: ["animated", "fadeIn"],
          animationOut: ["animated", "fadeOut"],
          dismiss: {
            duration: 3000,
            showIcon: true
          }
        })
      }

    })
  } else {
    store.addNotification({
      title: 'Passwords are not matching!',
      message: 'Please enter same passwords!',
      type: "danger",
      container: 'top-right',
      animationIn: ["animated", "fadeIn"],
      animationOut: ["animated", "fadeOut"],
      dismiss: {
        duration: 3000,
        showIcon: true
      }
    })
  }
}
return (
  <Container fluid>
    <Row className="justify-content-md-center">
      <Col className='col-md-5 mt-3'>
        <FadeTransform
          in
```

```
          transformProps={{
            exitTransform: 'scale(0.5) translateY(-50%)'
          }}>
          <Card className="frm">
            <div className="text-center mt-4 mb-4"><h1 className="">Register</h1></div>
            <Card.Img varient="top" className="mt-1 col-md-10 col-sm-10 offset-md-
1" src={loginImg}></Card.Img>
            <div className="text-center mt-4 mb-4"><span className="fa fa-star fa-lg mr-
2"></span><span className="fa fa-star fa-lg mr-2"></span><span className="fa fa-star fa-
lg mr-2"></span><span className="fa fa-star fa-lg mr-2"></span><span className="fa fa-
star fa-lg mr-2"></span></div>
            <Card.Body>
              <LocalForm onSubmit={(values) => handleSubmit(values)}>
                <div className="form-group">
                  <Row><Col className="col-md-4 offset-md-1">
                    <Form.Label>Name:</Form.Label>
                  </Col>
                    <Col className="col-md-6">
                      <Control.text
                        model=".fname"
                        autoComplete="off"
                        id="fname"
                        className="form-control"
                        value={name}
                        onChange={e => setName(e.target.value)}
                        placeholder="Enter Your Full Name"
                        validators={{
                          required, validName, maxLength: maxLength(15), minLength: minLength(3)
                        }} />
                      <Errors
                        className="text-danger"
                        model=".fname"
                        show="touched"
                        messages={{
                          required: 'Required ',
                          validName: 'Enter a valid Name!',
                          maxLength: 'Length should be less than 15 characters!',
                          minLength: 'Length should be greater than 3 characters!'        }}
                      />
                    </Col></Row>
```

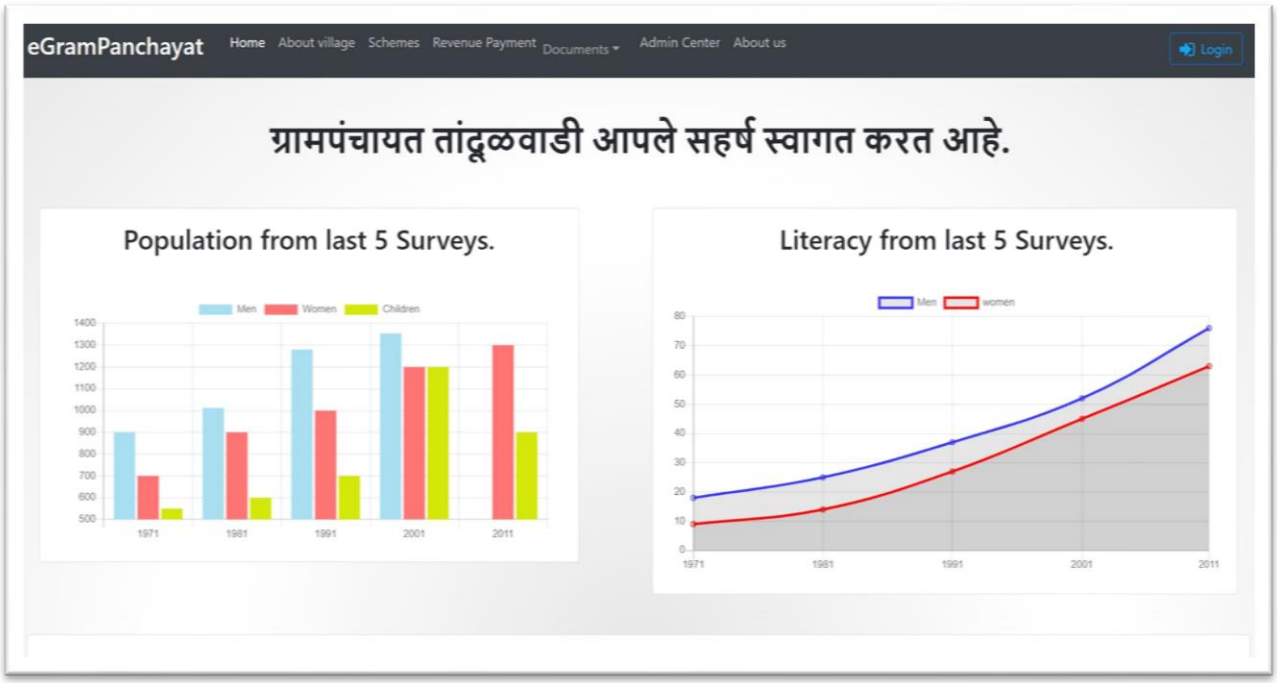
```
          </div>
          <Row><Col>
           <div className="form-group">
            <Row><Col className="col-md-4 offset-md-1">
             <Form.Label>Mobile Number:</Form.Label>
            </Col>
             <Col className="col-md-6">
              <Control.text
               model=".mobno"
               autoComplete="off"
               id="mobno"
               value={number}
               onChange={e => setNumber(e.target.value)}
               className="form-control"
               placeholder="Enter Your Mobile No"
               validators={{
                required, validMobile
               }} />
              <Errors
               className="text-danger"
               model=".mobno"
               show="touched"
               messages={{
                required: 'Required ',
                validMobile: 'Enter a valid Mobile number starting from +91!'}}
              />
            </Col></Row>
           </div></Col></Row>
          <Row><Col>
           <div className="form-group">
            <Row><Col className="col-md-4 offset-md-1">
             <Form.Label>Email:</Form.Label>
            </Col>
             <Col className="col-md-6">
              <Control.text
               autoComplete="off"
               model=".email"
               id="email"
               value={email}
               className="form-control"
```

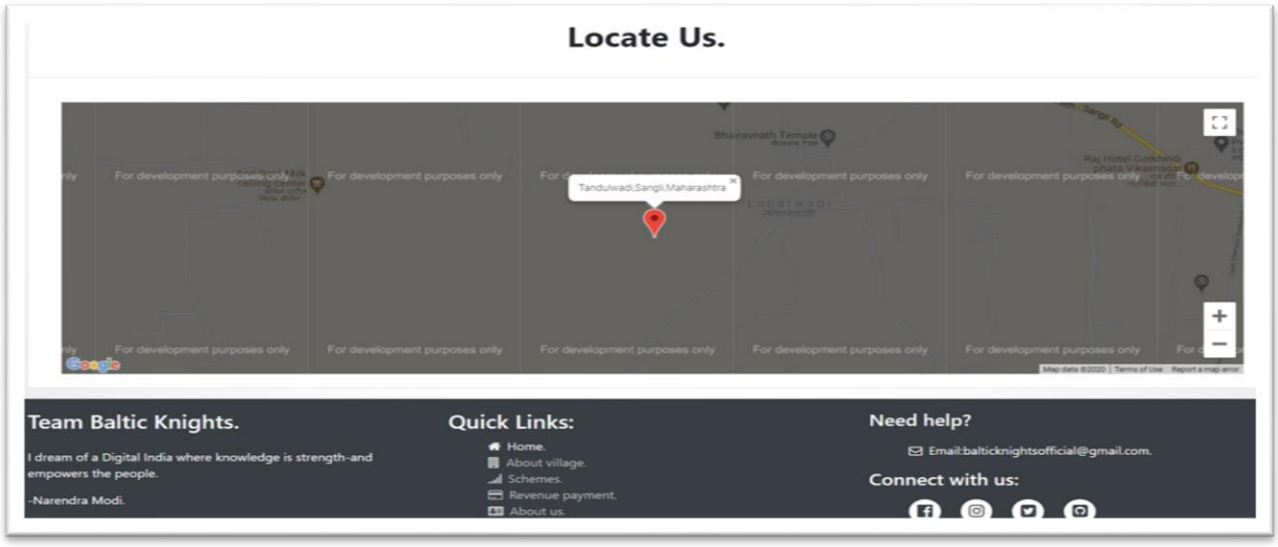
```
            placeholder="Enter Your Email Id"
            onChange={(e) => setEmail(e.target.value)}
            validators={{
              required, validEmail
            }} />
           <Errors
            className="text-danger"
            model=".email"
            show="touched"
            messages={{
              required: 'Required ',
              validEmail: 'Enter a valid email address!'
            }}
           />
          </Col></Row>
        </div></Col></Row>
       <Row><Col>
        <div className="form-group">
         <Row><Col className="col-md-4 offset-md-1">
          <Form.Label>Password:</Form.Label>
         </Col>
          <Col className="col-md-6">
           <Control.text
            model=".password"
            autoComplete="off"
            id="password"
            className="form-control"
            placeholder="Enter Your Password"
            value={password1}
            onChange={(e) => setPassword1(e.target.value)}
            validators={{
              required, minLength: minLength(8)
            }} />
           <Errors
            className="text-danger"
            model=".password"
            show="touched"
            messages={{
              required: 'Required ',
              minLength: 'Password should be greater than 8 characters!'
```

```
                    }}
                  />
                </Col></Row>
              </div></Col></Row>
            <Row><Col>
              <div className="form-group">
                <Row><Col className="col-md-4 offset-md-1">
                  <Form.Label>Confirm Password:</Form.Label>
                </Col>
                  <Col className="col-md-6">
                    <Control.text
                      model=".rpassword"
                      autoComplete="off"
                      id="rpassword"
                      className="form-control"
                      placeholder="Confirm Password"
                      value={password2}
                      onChange={(e) => setPassword2(e.target.value)}
                      validators={{
                        required, minLength: minLength(8)
                      }} />
                    <Errors
                      className="text-danger"
                      model=".rpassword"
                      show="touched"
                      messages={{
                        required: 'Required ',
                        minLength: 'Password should be greater than 8 characters!'
                      }}
                    />
                  </Col></Row>
                </div></Col></Row>
              <div className="text-center mt-
2"><Button variant="primary" type="submit">Register</Button></div>
            </LocalForm>
          </Card.Body>
        </Card></FadeTransform>
      </Col>
    </Row>
  </Container>
```

```
  );}
export default Register;
```

## Home Module



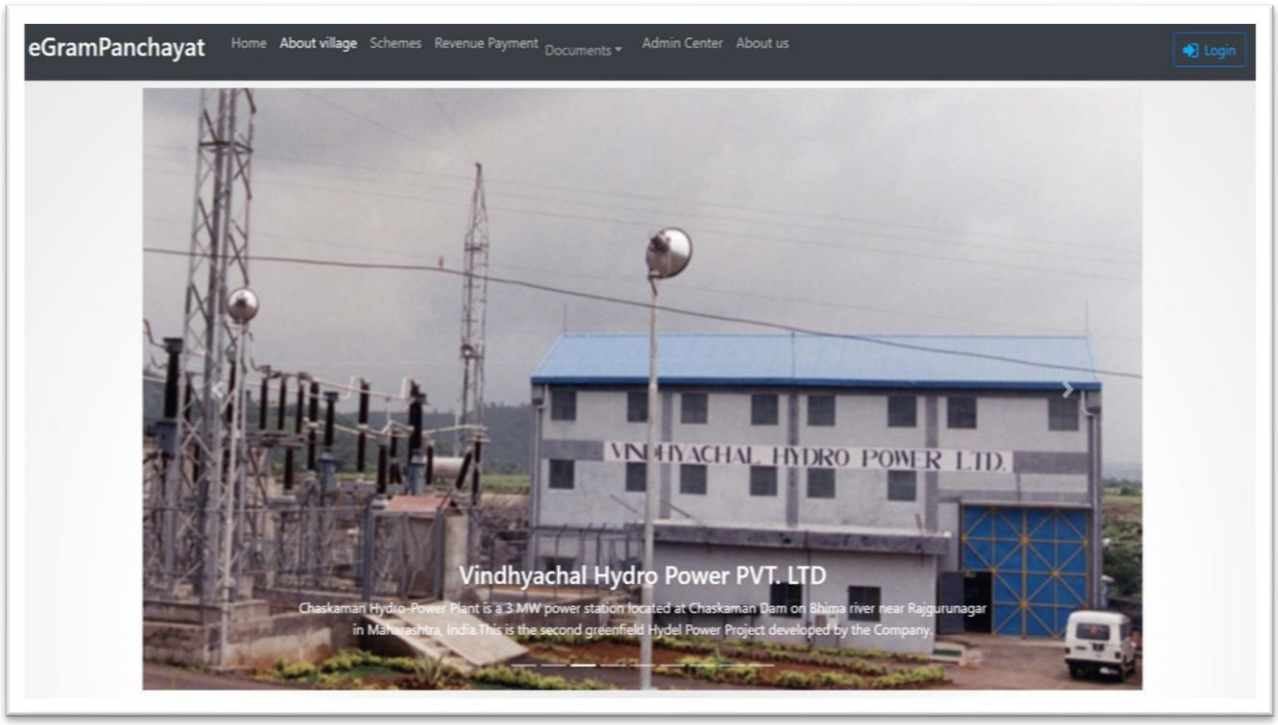

## Source Code:

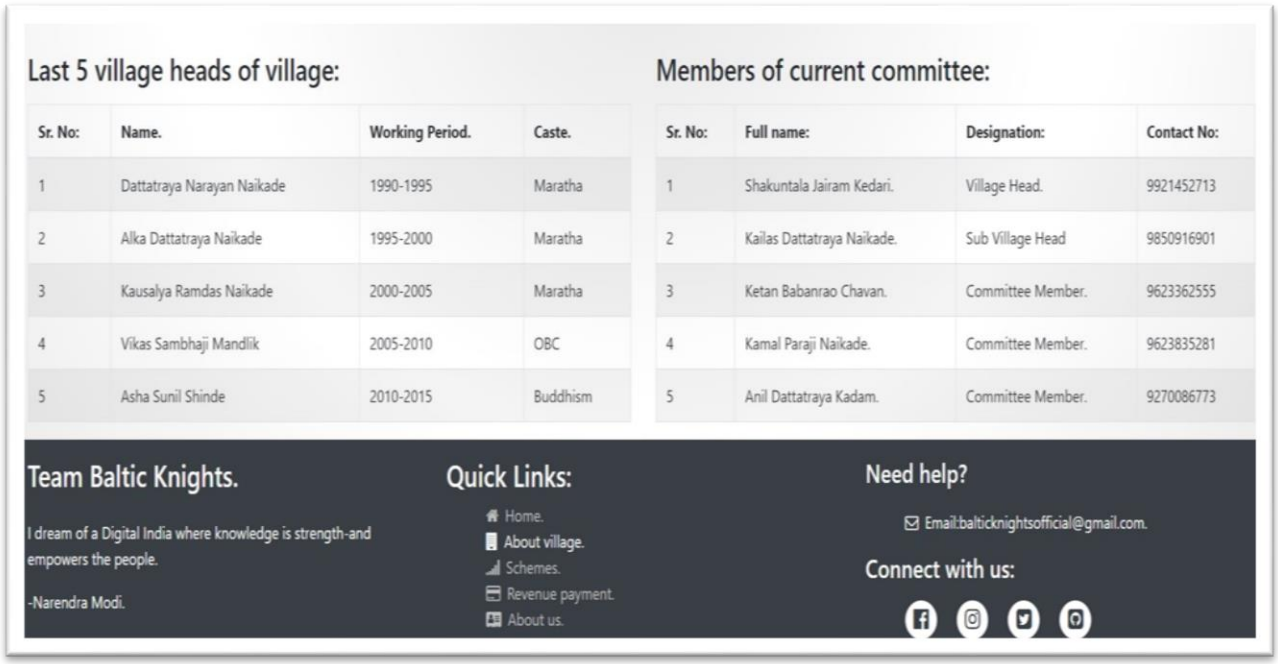
```
import React from "react";
import { Container, Row, Col, Card } from 'react-bootstrap';
```

```
import Population from './population';
import Literacy from './literacy';
import Maps from './maps';
import { FadeTransform } from 'react-animation-components';
const Home = () => {
   return (
      <Container fluid className="mb-3">
         <Row className="text-center mt-5">
            <Col>
               <h1 className="font-weight-
bold">ग्रामपंचायत तांदूळवाडी आपले सहर्ष स्वागत करत आहे.</h1>
            </Col>
         </Row>
         <Container fluid>
            <Row className="d-flex mt-5">
               <Col className="col-md-6">
                  <Population />
               </Col>
               <Col className="col-md-6">
                  <Literacy />
               </Col>
            </Row>
         </Container>
         <Row className="mt-5">
            <Col className="col-md-12">
               <FadeTransform
                  in
                  transformProps={{
                     exitTransform: 'scale(0.5) translateY(-50%)'}}>
                  <Card>
                     <Card.Title className="text-center mt-3">
                        <h1 className="font-weight-bold">Locate Us.</h1></Card.Title>
                     <hr />
                     <Card.Body className="text-center mt-0">
                        <Maps />
                     </Card.Body>
                  </Card>
               </FadeTransform> </Col> </Row>
      </Container>
   );}
```

```
export default Home;
```

## About Village Module




## Source Code:

```
import React, { useState, useEffect } from "react";
import { Container, Col, Row, Carousel, Table } from 'react-bootstrap';
import children from './carousel images/children1.jpg';
import houses from './carousel images/houses1.jpg';
import roads from './carousel images/roads1.jpg';
import women from './carousel images/womens1.jpg';
import dam from './carousel images/dam1.jpg';
import village from './carousel images/village1.jpg';
import river from './carousel images/rivers1.jpg';
import powerhouse from './carousel images/power house1.jpg';
import temple from './carousel images/khandoba.jpg';
import { useDispatch, useSelector } from 'react-redux';
import { committeeFetch } from '../../Redux/actions/committeeActions';
import { FadeTransform, Fade, Stagger } from 'react-animation-components';
const About = () => {
    const dispatch = useDispatch()
    useEffect(() => {
        dispatch(committeeFetch());
    }, [])
    const Record = useSelector((state) => state.committee);
    // console.log(Record)
    let currentCommittee = "";
    let previousCommittee = "";
    if (Record.current.designation) {
        currentCommittee = Record?.current?.Name.map((name, key) => {
            return (
                <tr id={key}>
                    <td>{key + 1}</td>
                    <td>{name}</td>
                    <td>{Record?.current?.designation[key]}</td>
                    <td>{Record?.current?.contact[key]}</td>
                </tr>
            );
        });
        previousCommittee = Record?.previous?.Name.map((name, key) => {
            return (
                <tr id={key}>
                    <td>{key + 1}</td>
                    <td>{name}</td>
                    <td>{Record?.previous?.workingPeriod[key]}</td>
```

```
                <td>{Record?.previous?.Caste[key]}</td>
            </tr>
        );
    });
  }
  return (
    <>
      <Container fluid className="">
        <Row className="d-flex justify-content-md-center ml-0 mr-0">
          <Col className="col-md-10 col-xs-6 mt-2">
            <FadeTransform
              in
              transformProps={{
                exitTransform: 'scale(0.5) translateY(-50%)'
              }}>
              <Carousel>
                <Carousel.Item interval={1000}>
                  <img
                    className="d-block h-50 w-100"
                    src={village}
                    alt="village"
                  />
                  <Carousel.Caption>
                    <h3 style={{ color: "#000000" }}>Kadadhe Village.</h3>
                    <p style={{ color: "#000000" }}>Kadadhe is a Village in Khed Taluka i
n Pune District of Maharashtra State, India. It belongs to Desh or Paschim Maharashtra region . It
belongs to Pune Division . It is located 56 KM towards North from District head quarters Pune. 15
 KM from Khed. 116 KM from State capital Mumbai.</p>
                  </Carousel.Caption>
                </Carousel.Item>
                <Carousel.Item interval={1000}>
                  <img
                    className="d-block w-100"
                    src={dam}
                    alt="dam"
                  />
                  <Carousel.Caption>
                    <h3 style={{ color: "#000000" }}>ChasKaman Water Reservoir</h3>
                    <p style={{ color: "#000000" }}>The Chaskaman Dam is one of the imp
ortant dams of Maharashtra and is built on the Bhima River at Rajgurunagar in Pune district.The c
```

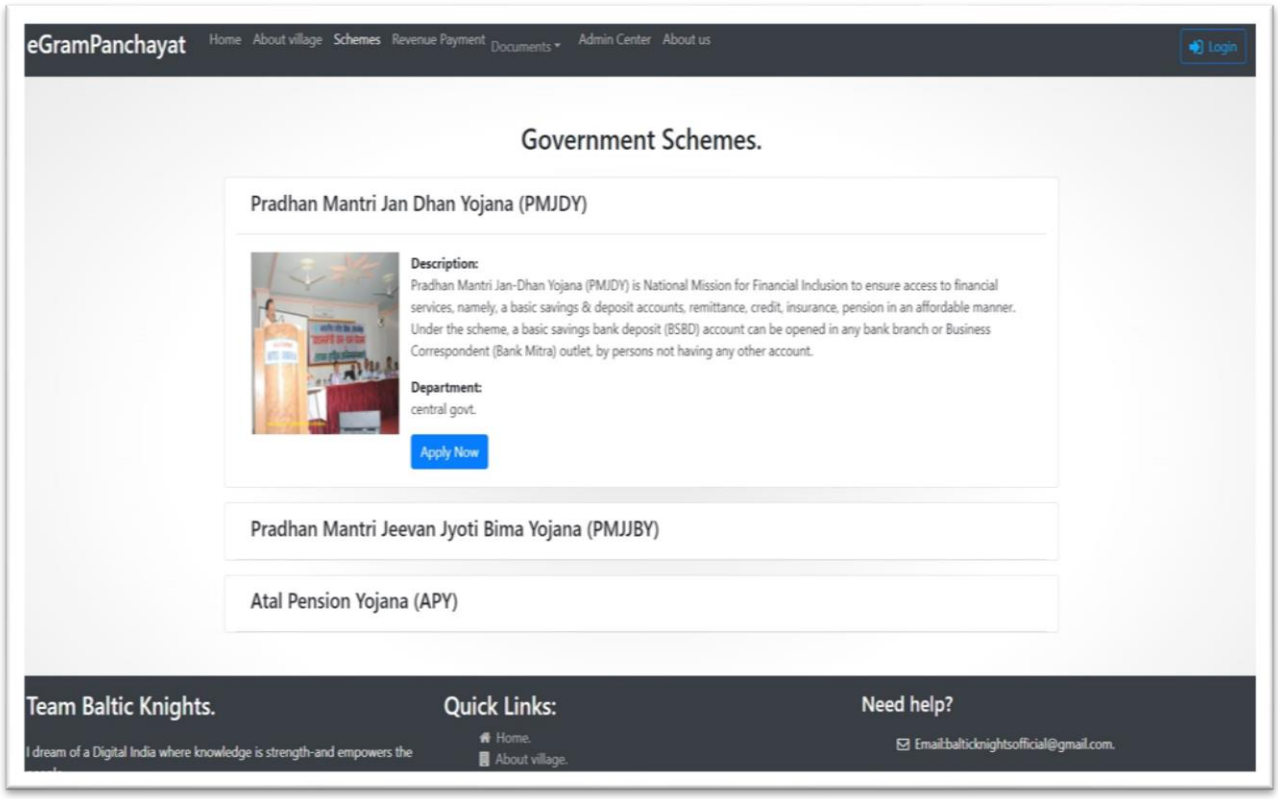
```
apacity of Chas Kaman Dam to irrigate about 32824 ha of land of the villages nearby in Pune distr
ict.</p>
                        </Carousel.Caption>
                    </Carousel.Item>
                    <Carousel.Item interval={1000}>
                        <img
                            className="d-block w-100"
                            src={powerhouse}
                            alt="powerhouse"
                        />
                        <Carousel.Caption>
                            <h3>Vindhyachal Hydro Power PVT. LTD</h3>
                            <p>Chaskaman Hydro-
Power Plant is a 3 MW power station located at Chaskaman Dam on Bhima river near Rajgurunag
ar in Maharashtra, India.This is the second greenfield Hydel Power Project developed by the Com
pany.</p>
                        </Carousel.Caption>
                    </Carousel.Item>
                    <Carousel.Item interval={1000}>
                        <img
                            className="d-block w-100"
                            src={houses}
                            alt="houses"
                        />
                        <Carousel.Caption>
                            <h3>Permenant Agriculture Village.</h3>
                            <p>People from this village are mostly rely on farming over the generati
ons.their houses are reflections of their sinple lifestyle.</p>
                        </Carousel.Caption>
                    </Carousel.Item>
                    <Carousel.Item interval={1000}>
                        <img
                            className="d-block w-100"
                            src={temple}
                            alt="temple"
                        />
                        <Carousel.Caption>
                            <h3>Khandoba temple.</h3>
                            <p>Temple of God Khandoba on north side of village.This is the second
Biggest temple in Pune District with development cost of 3 crores.</p>
```

```
                    </Carousel.Caption>
                </Carousel.Item>
                <Carousel.Item interval={1000}>
                    <img
                        className="d-block w-100"
                        src={river}
                        alt="river"
                    />
                    <Carousel.Caption>
                        <h3 style={{ color: "#000000" }}>Bhima River.</h3>
                        <p>This village on Bank of holy river Bhima.This river is backbone of vi
llagers in terms of water,farming.</p>
                    </Carousel.Caption>
                </Carousel.Item>
                <Carousel.Item interval={1000}>
                    <img
                        className="d-block w-100"
                        src={children}
                        alt="children"
                    />
                    <Carousel.Caption>
                        <h3>Primary School.</h3>
                        <p>There is one primary school for children in village which focuses on
overall growth of students.</p>
                    </Carousel.Caption>
                </Carousel.Item>
                <Carousel.Item interval={1000}>
                    <img
                        className="d-block w-100"
                        src={women}
                        alt="women"
                    />
                    <Carousel.Caption>
                        <h3>Independent Women.</h3>
                        <p>There are lots of work opportunities for women in our village.women
 can live independently and fulfil their family needs.</p>
                    </Carousel.Caption>
                </Carousel.Item>
                <Carousel.Item interval={1000}>
                    <img
```

```
                        className="d-block w-100"
                        src={roads}
                        alt="roads"
                      />
                      <Carousel.Caption>
                        <h3>Concrete roads.</h3>
                        <p>Village has network of concrete roads to connect village with main r
oad.</p>
                      </Carousel.Caption>
                    </Carousel.Item>
                  </Carousel>
                </FadeTransform>
              </Col>
            </Row>
          </Container>
          <Container fluid className="mt-4">
            <Row className="d-flex justify-content-md-center">
              <Col className="col-md-6 col-xs-6 mt-2">
                <h3>Last 5 village heads of village:</h3>
                <FadeTransform
                  in
                  transformProps={{
                    exitTransform: 'scale(0.5) translateY(-50%)'
                  }}>
                  <Table striped bordered hover className="mt-3">
                    <thead>
                      <tr>
                        <th>Sr. No:</th>
                        <th>Name.</th>
                        <th>Working Period.</th>
                        <th>Caste.</th>
                      </tr>
                    </thead>
                    <tbody>
                      {previousCommittee}
                    </tbody>
                  </Table></FadeTransform>
              </Col>
              <Col className="col-md-6 col-xs-6 mt-2">
                <h3>Members of current committee:</h3>
```

```
                <FadeTransform
                    in
                    transformProps={{
                    exitTransform: 'scale(0.5) translateY(-50%)'
                    }}>
                    <Table striped bordered hover className="mt-3">
                        <thead>
                            <tr>
                                <th>Sr. No:</th>
                                <th>Full name:</th>
                                <th>Designation:</th>
                                <th>Contact No:</th>
                            </tr>
                        </thead>
                        <tbody>
                            {currentCommittee}
                        </tbody>
                    </Table></FadeTransform>
                </Col>
            </Row>
        </Container>
    </>
  );
}
export default About;
```

## Schemes Modules



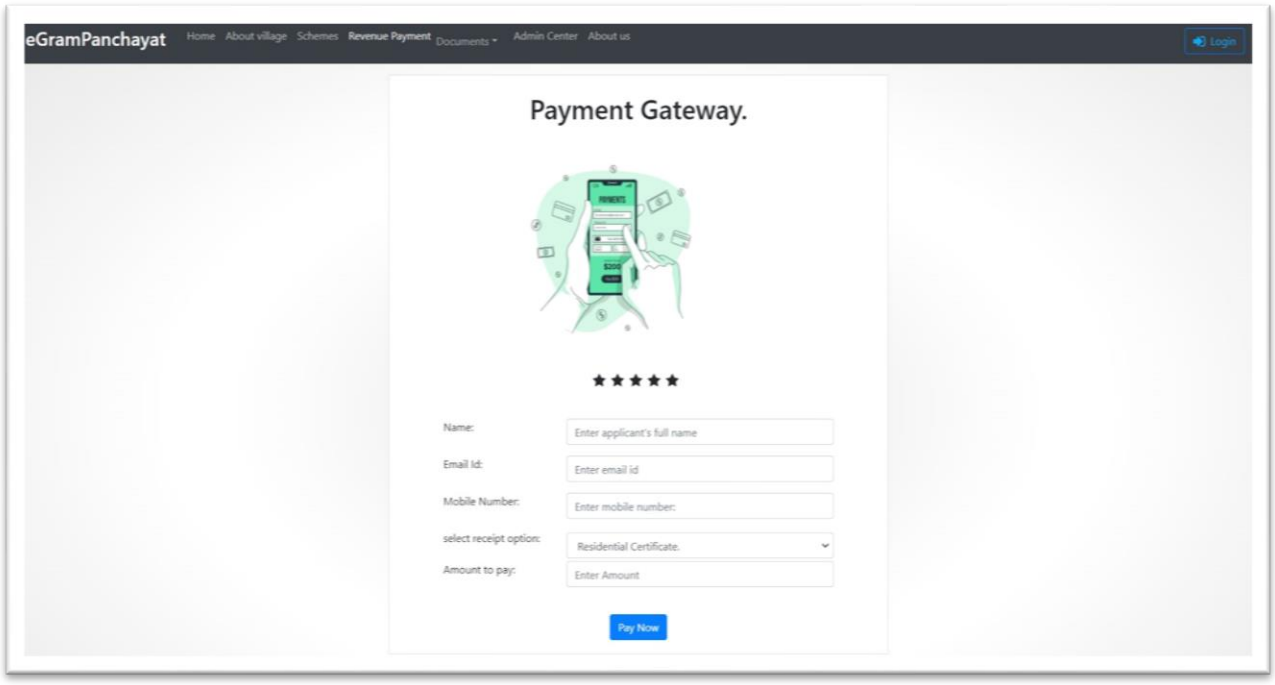

## Source Code:

```
import React, { useEffect } from "react";
import { Container, Media, Row, Card, Accordion, Button, Col } from 'react-bootstrap'
import { useDispatch, useSelector } from 'react-redux';
import { fetchSchemes } from '../../Redux/actions/schemesAction';
import { FadeTransform, Fade, Stagger } from 'react-animation-components';
import { jsx,css } from "@emotion/react";
import "./schemes-style.css"
const override = css`
  display: block;
  margin: 0 auto;
  border-color: red;
`;
const Schemes = () => {
  const dispatch = useDispatch();
  useEffect(() => {
    dispatch(fetchSchemes())
  }, [])
```

```
 const schemes = useSelector(state => state.schemes.data);
 const activeKey = schemes[0]?._id;
 const schemesComponent = schemes.map((scheme) => {
  return (
   <Fade in>
    <Row key={scheme._id} className="mt-3 justify-content-center">
     <Card className="col-md-8 col-sm-8 col-xs-8">
      <Accordion className="myAccordian" defaultActiveKey={activeKey}>
       <Accordion.Toggle as={Card.Header} className="back" eventKey={scheme._id}>
        <h4>{scheme.title}</h4>
       </Accordion.Toggle>
       <Accordion.Collapse eventKey={scheme._id}>
        <Card.Body>
         <Media tag="li">
          <img
           width={200}
           height={200}
           className="mr-3"
           src={scheme.picture}
           alt={scheme.title}
          />
          <Media.Body className="">
           <p><b>Description:</b><br />{scheme.description}</p>
           <p><b>Department:</b><br />{scheme.department}</p>
           <Button href={scheme.weblink} target="blank" className="mr-
auto">Apply Now</Button>
          </Media.Body>
         </Media>
        </Card.Body>
       </Accordion.Collapse>
      </Accordion>
     </Card>
    </Row>
   </Fade>
  );
 });
 return (
  { schemes } ? <Container fluid className="mt-5 mb-5">
   <Row className="mt-3 mb-3">
    <Col>
```

```
        <h2 className="text-center">Government Schemes.</h2>
      </Col>
    </Row>
    <Stagger in>{schemesComponent}</Stagger>
  </Container> : <div>
    {/* <ClipLoader
    css={override}
    size={150}
    color={"#123abc"}
    loading={this.state.loading}
  /> */}
  </div>
 );
}
export default Schemes;
```

## Payment Gateway Module




## Source Code:

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import PaymentImg from './payment.jpg';
```

```
import { Control, LocalForm, Errors } from 'react-redux-form';
import { Container, Card, Form, Button, Row, Col } from 'react-bootstrap';
import { FadeTransform } from 'react-animation-components';
const required = (val) => val && val.length;
const maxLength = (len) => (val) => !(val) || (val.length <= len);
const minLength = (len) => (val) => val && (val.length >= len);
const isNumber = (val) => !isNaN(Number(val));
const validText = (val) => /^[a-zA-Z]+ [a-zA-Z]+$/i.test(val);
const validEmail = (val) => /^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$/i.test(val);
const validMobile=(val)=> /^((\+){1}91){1}[1-9]{1}[0-9]{9}$/i.test(val);
const Payment = () => {
    const [name, setName] = useState("");
    const [amount, setAmount] = useState("");
    const [email, setEmail] = useState("");
    const [reason, setReason] = useState("");
    const [number, setNumber] = useState();
    function isDate(val) {
        // Cross realm comptatible
        return Object.prototype.toString.call(val) === '[object Date]'
    }
    function isObj(val) {
        return typeof val === 'object'
    }
    function stringifyValue(val) {
        if (isObj(val) && !isDate(val)) {
            return JSON.stringify(val)
        } else {
            return val
        }
    }
    function buildForm({ action, params }) {
        const form = document.createElement('form')
        form.setAttribute('method', 'post')
        form.setAttribute('action', action)
        Object.keys(params).forEach(key => {
            const input = document.createElement('input')
            input.setAttribute('type', 'hidden')
            input.setAttribute('name', key)
            input.setAttribute('value', stringifyValue(params[key]))
            form.appendChild(input)
```

```
    })
    return form
}
function post(details) {
    const form = buildForm(details)
    document.body.appendChild(form)
    form.submit()
    form.remove()
}
const handleSubmit = async (e) => {
    const str = name.split(" ").join("");
    const paymentData = {
        name: str,
        forReason: reason,
        amount: amount,
        emailId: email,
        number: number,
        orderId: "ORDER_ID" + (new Date().getTime())
    }
    console.log(paymentData);
    var url = 'http://localhost:5000/pay/paynow';
    var request = {
        url: url,
        params: paymentData,
        method: "get"
    }
    const response = await axios(request);
    const processParams = await response.data;
    console.log(processParams)
    var details = {
        action: "https://securegw-stage.paytm.in/order/process",
        params: processParams
    }
    post(details);
}
return (
    <Container fluid className="mb-3">
        <Row className="justify-content-md-center">
            <Col className='col-md-5 mt-3' >
                <FadeTransform
```

```
                in
                transformProps={{
                   exitTransform: 'scale(0.5) translateY(-50%)'
                }}>
                <Card className='frm'>
                   <div className="text-center mt-4 mb-
4"><h1 className="">Payment Gateway.</h1></div>
                   <Card.Img varient="top" className="pic mt-1 col-md-6 col-sm-10 offset-md-
3" src={PaymentImg}></Card.Img>
                   <div className="text-center mt-4 mb-4"><span className="fa fa-star fa-
lg mr-2"></span><span className="fa fa-star fa-lg mr-2"></span><span className="fa fa-
star fa-lg mr-2"></span><span className="fa fa-star fa-lg mr-
2"></span><span className="fa fa-star fa-lg mr-2"></span></div>
                   <Card.Body>
                      <LocalForm onSubmit={handleSubmit}>
                         <div className="form-group">
                            <Row><Col className="col-md-3 offset-md-1">
                               <Form.Label>Name:</Form.Label></Col>
                               <Col className="col-md-7">
                                  <Control.text
                                     model=".name"
                                     className="form-control"
                                     autocomplete="off"
                                     placeholder="Enter applicant's full name"
                                     name="name"
                                     value={name}
                                     onChange={(e) => setName(e.target.value)}
                                     validators={{
                                        required, validText, maxLength: maxLength(20), minLength:
minLength(3)
                                     }}
                                  />
                                   <Errors
                                     className="text-danger"
                                     model=".name"
                                     show="touched"
                                     messages={{
                                        required: 'Required ',
                                        validText: 'Enter a valid Name!',
                                        maxLength: 'Length should be less than 20 characters!',
```

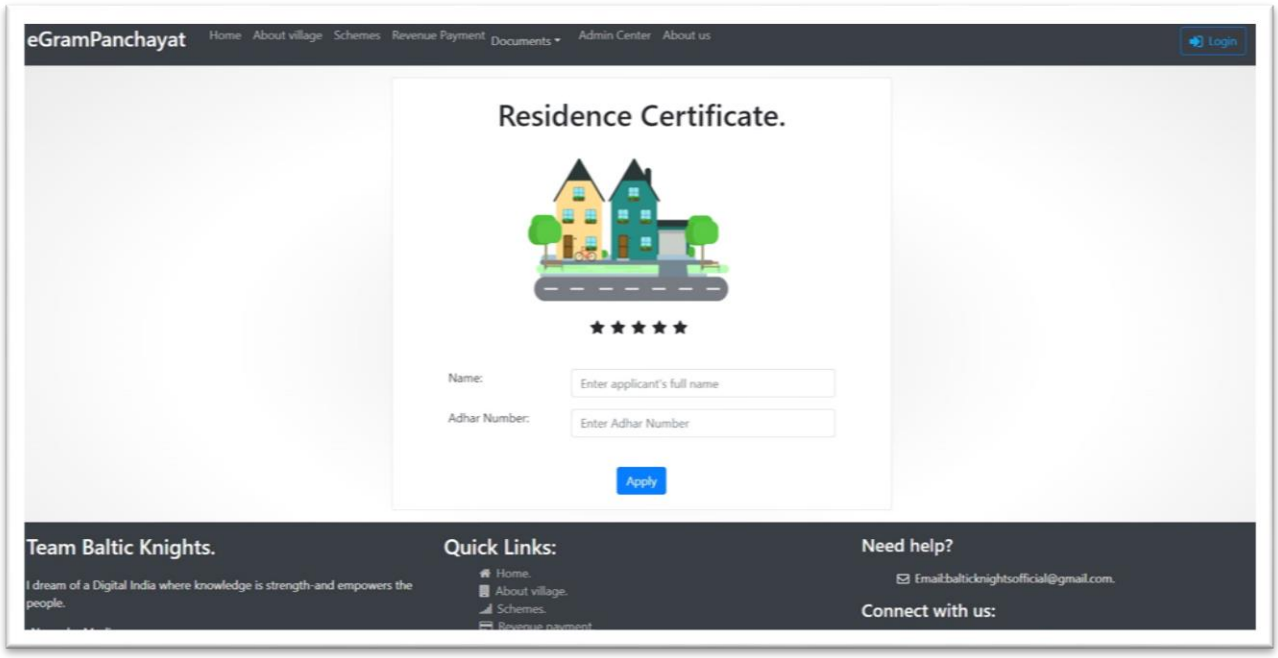
```
                        minLength: 'Length should be greater than 3 characters!'
                    }}
                />
            </Col></Row>
        </div>
        <div className="form-group">
            <Row><Col className="col-md-3 offset-md-1">
                <Form.Label>Email Id:</Form.Label></Col>
                <Col className="col-md-7">
                    <Control.text
                        model=".email"
                        className="form-control"
                        autocomplete="off"
                        placeholder="Enter email id"
                        name="email"
                        value={email}
                        onChange={(e) => setEmail(e.target.value)}
                        validators={{
                            required, validEmail
                        }}
                    />
                    <Errors
                        className="text-danger"
                        model=".email"
                        show="touched"
                        messages={{
                            required: 'Required ',
                            validText: 'Enter a valid Email!'
                        }}
                    />
                </Col></Row>
        </div>
        <div className="form-group">
            <Row><Col className="col-md-3 offset-md-1">
                <Form.Label>Mobile Number:</Form.Label></Col>
                <Col className="col-md-7">
                    <Control.text
                        model=".mobile"
                        className="form-control"
                        autocomplete="off"
```

```
                                placeholder="Enter mobile number:"
                                name="number"
                                value={number}
                                onChange={(e) => setNumber(e.target.value)}
                                validators={{
                                    required, validMobile
                                }}
                            />
                            <Errors
                                className="text-danger"
                                model=".mobile"
                                show="touched"
                                messages={{
                                    required: 'Required ',
                                    validMobile: 'Enter a valid Mobile number starting from +91!'
                                }}
                            />
                        </Col></Row>
                    </div>
                    <Row><Col className="col-md-3 offset-md-
1"><Form.Label>select receipt option:</Form.Label></Col>
                        <Col className="col-md-7">
                            <Control.select
                                model=".reason"
                                as="select"
                                className="my-1 mr-sm-2 form-control"
                                id="inlineFormCustomSelectPref"
                                custom
                                value={reason}
                                onChange={(e) => setReason(e.target.value)}
                            >
                                <option value="Residential Certificate.">Residential Certificate.</o
ption>
                                <option value="Revenue tax receipt.">Revenue tax receipt.</option
>                            </Control.select>
                        </Col></Row>
                    <Row><Col>
                    <div className="form-group">
                        <Row><Col className="col-md-3 offset-md-
1"><Form.Label>Amount to pay:</Form.Label></Col>
```

```
                                <Col className="col-md-7">
                                  <Control.text
                                    model=".amount"
                                    className="form-control"
                                    autoComplete="off"
                                    placeholder="Enter Amount"
                                    name="Amount"
                                    value={amount}
                                    onChange={(e) => setAmount(e.target.value)}
                                  />
                                </Col></Row>
                            </div></Col></Row>
                        <div className="text-center mt-4">
                          <Button variant="primary" type="submit">Pay Now</Button>
                        </div>
                    </LocalForm>
                </Card.Body>
            </Card></FadeTransform></Col>
        </Row>
    </Container>
  )
}
export default Payment;
```

## Documents Module:

### Residence certificate




## Source Code:

```
import React, { useState } from 'react';
import './residence.css';
import axiosInstance from '../../helpers/axios';
import ResidenceImg from './residence.png';
import { Control, LocalForm, Errors } from 'react-redux-form';
import { Container, Card, Form, Button, Row, Col } from 'react-bootstrap';
import { FadeTransform } from 'react-animation-components';
import { store } from 'react-notifications-component';
import 'react-notifications-component/dist/theme.css';
import 'animate.css';
const required = (val) => val && val.length;
const maxLength = (len) => (val) => !(val) || (val.length <= len);
const minLength = (len) => (val) => val && (val.length >= len);
const isNumber = (val) => !isNaN(Number(val));
const validText = (val) => /^[a-zA-Z]+ [a-zA-Z]+$/i.test(val);
const Residence = () => {
    const [name, setName] = useState("");
    const [UID, setUID] = useState();
```

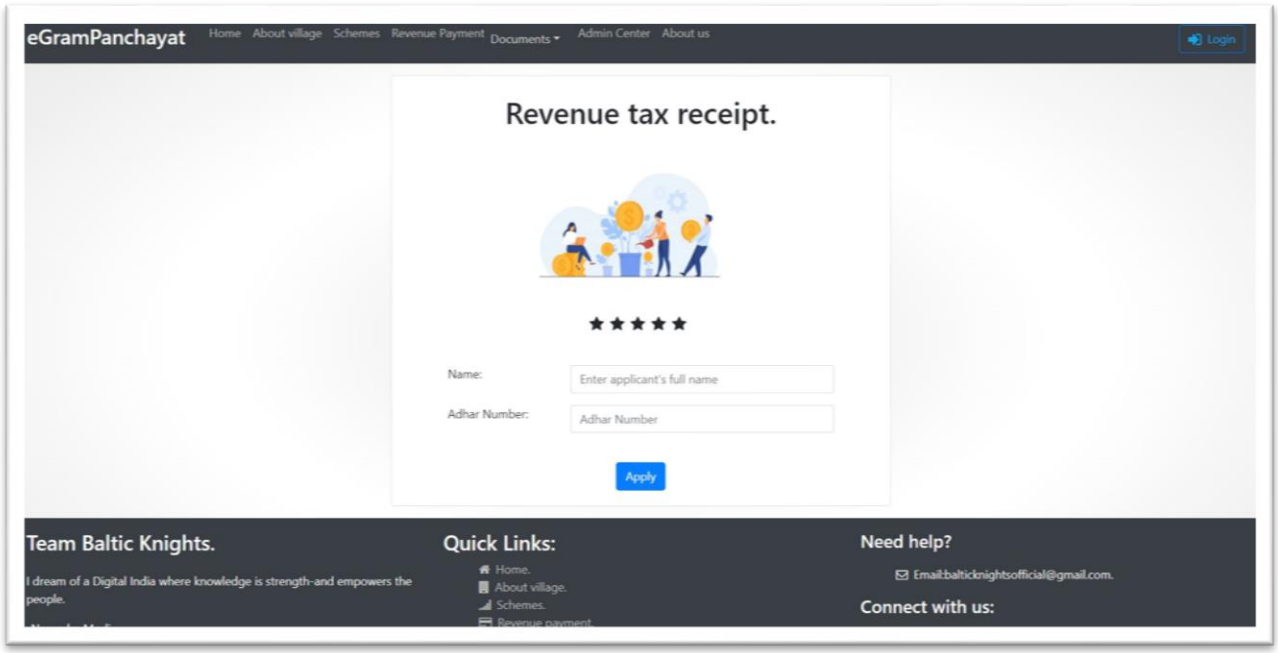
```
const generatePDF = e => {

    const residenceData = {
        name: name,
        UID: Number(UID),
        date: new Date()
    }
    console.log(residenceData);
    axiosInstance.post('residence/create', residenceData)
    store.addNotification({
        title: 'Registration Successful!!!',
        message: 'Your Application is Submitted Successfully.',
        type: "success",
        container: 'top-right',
        animationIn: ["animated", "fadeIn"],
        animationOut: ["animated", "fadeOut"],
        dismiss: {
            duration: 3000,
            showIcon: true
        }
    })
}
const downloadPDF = e => {
    axiosInstance.get('residence/download', { responseType: 'arraybuffer' })
        .then(res => {
            const url = window.URL.createObjectURL(new Blob([res.data]
                , { type: "application/pdf" }))
            var link = document.createElement('a');
            link.href = url;
            link.setAttribute('download', 'residence.pdf');
            document.body.appendChild(link);
            link.click();
        })
}
return (
    <Container fluid className="mb-3">
        <Row className="justify-content-md-center">
            <Col className='col-md-5 mt-3' >
                <FadeTransform
```

```
                in
                transformProps={{
                    exitTransform: 'scale(0.5) translateY(-50%)'
                }}>
                <Card className='frm'>
                    <div className="text-center mt-4 mb-
4"><h1 className="">Residence Certificate.</h1></div>
                    <Card.Img varient="top" className="pic mt-1 col-md-6 col-sm-10 offset-md-
3" src={ResidenceImg}></Card.Img>
                    <div className="text-center mt-4 mb-4"><span className="fa fa-star fa-
lg mr-2"></span><span className="fa fa-star fa-lg mr-2"></span><span className="fa fa-
star fa-lg mr-2"></span><span className="fa fa-star fa-lg mr-
2"></span><span className="fa fa-star fa-lg mr-2"></span></div>
                    <Card.Body>
                        <LocalForm>
                            <div className="form-group">
                                <Row><Col className="col-md-3 offset-md-1">
                                    <Form.Label>Name:</Form.Label></Col>
                                    <Col className="col-md-7">
                                        <Control.text
                                            model=".name"
                                            className="form-control"
                                            autoComplete="off"
                                            placeholder="Enter applicant's full name"
                                            name="name"
                                            value={name}
                                            onChange={(e) => setName(e.target.value)}
                                            validators={{
                                                required, maxLength: maxLength(20), minLength: minLength(
3)
                                            }}
                                        />
                                        <Errors
                                            className="text-danger"
                                            model=".name"
                                            show="touched"
                                            messages={{
                                                required: 'Required ',
maxLength: 'Length should be less than 15 characters!',
                                                minLength: 'Length should be greater than 3 characters!'
```

```
                                    }}
                                />
                            </Col></Row>
                        </div>
                        <Row><Col>
                            <div className="form-group">
                                <Row><Col className="col-md-3 offset-md-
1"><Form.Label>Adhar Number:</Form.Label></Col>
                                    <Col className="col-md-7">
                                        <Control.text
                                            model=".adhar"
                                            className="form-control"
                                            autoComplete="off"
                                            placeholder="Enter Adhar Number"
                                            name="UID"
                                            value={UID}
                                            onChange={(e) => setUID(e.target.value)}
                                            validators={{
                                                required, isNumber, maxLength: maxLength(12), minLength
: minLength(12)
                                            }}
                                        />
                                        <Errors
                                        className="text-danger"
                                        model=".adhar"
                                        show="touched"
                                        messages={{
                                            required: 'Required ',
                                            isNumber: 'Enter a valid Number!',
                                            maxLength: 'Length should be less than 12 characters!',
                                            minLength: 'Length should be exact 12 characters!'
                                        }}
                                    />
                                    </Col></Row>
                            </div></Col></Row>
                        <div className="text-center mt-4">
                            <Button variant="primary" type="submit" onClick={generatePDF}>App
ly</Button>
                        </div>
                    </LocalForm>
```

```
                    </Card.Body>
                </Card></FadeTransform></Col>
            </Row>
        </Container>
    )
}
export default Residence;
```

## Revenue tax receipt




**Source code:**

```
import React, { useState } from 'react';
import './revenue.css';
import axiosInstance from '../../helpers/axios';
import RevenueImg from './revenue.jpg';
import { Control, LocalForm, Errors } from 'react-redux-form';
import { Container, Card, Form, Button, Row, Col } from 'react-bootstrap';
import { FadeTransform } from 'react-animation-components';
import { store } from 'react-notifications-component';
import 'react-notifications-component/dist/theme.css';
import 'animate.css';
```

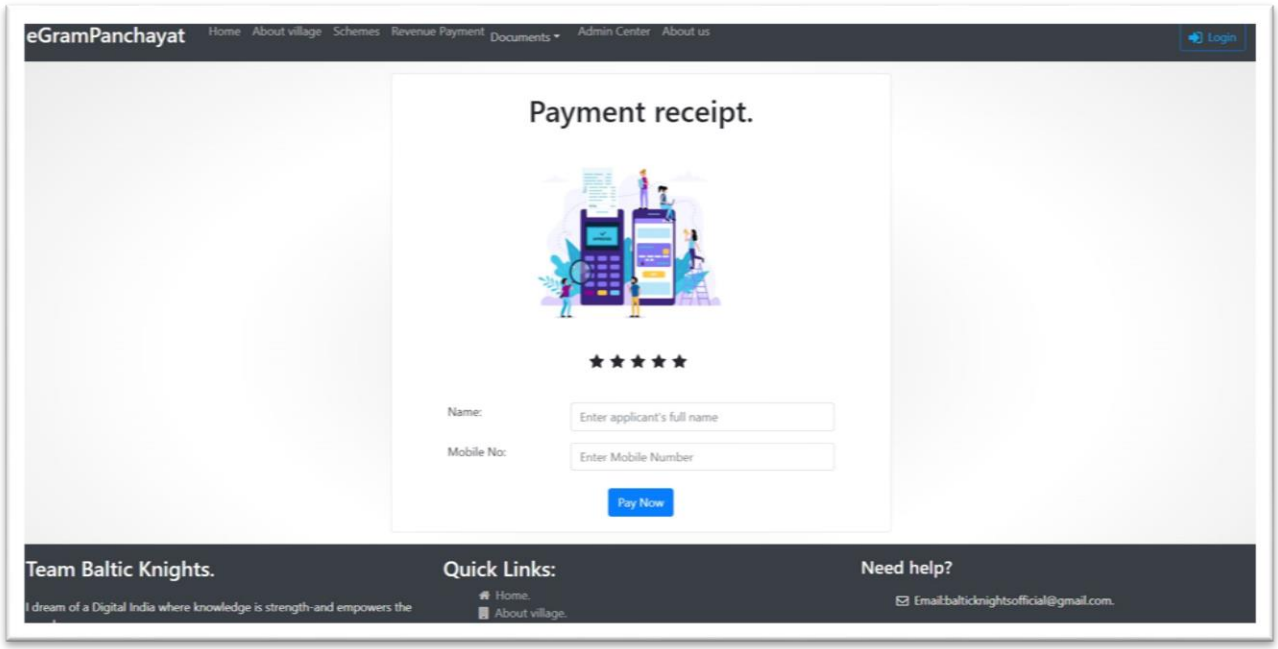
```
const required = (val) => val && val.length;
const maxLength = (len) => (val) => !(val) || (val.length <= len);
const minLength = (len) => (val) => val && (val.length >= len);
const isNumber = (val) => !isNaN(Number(val));
const validText = (val) => /^[a-zA-Z]+ [a-zA-Z]+$/i.test(val);
const Residence = () => {
   const [name, setName] = useState("");
   const [UID, setUID] = useState();
   const generatePDF = e => {
      const revenueData = {
         name:name,
         UID:Number(UID)
      }
      console.log(revenueData);
      axiosInstance.post('revenue/create', revenueData)
      store.addNotification({
         title: 'Registration Successful!!',
         message: 'Your Application is Submitted Successfully!',
         type: "success",
         container: 'top-right',
         animationIn: ["animated", "fadeIn"],
         animationOut: ["animated", "fadeOut"],
         dismiss: {
            duration: 3000,
            showIcon: true
         }
      })
  }
   return (
      <Container fluid className="mb-3">
         <Row className="justify-content-md-center">
            <Col className='col-md-5 mt-3' >
               <FadeTransform
                  in
                  transformProps={{
```

```
                    exitTransform: 'scale(0.5) translateY(-50%)'
                }}>
                <Card className='frm'>
                    <div className="text-center mt-4 mb-
4"><h1 className="">Revenue tax receipt.</h1></div>
                    <Card.Img varient="top" className="pic mt-1 col-md-6 col-sm-
10 offset-md-3" src={RevenueImg}></Card.Img>
                    <div className="text-center mt-4 mb-
4"><span className="fa fa-star fa-lg mr-2"></span><span className="fa fa-
star fa-lg mr-2"></span><span className="fa fa-star fa-lg mr-
2"></span><span className="fa fa-star fa-lg mr-
2"></span><span className="fa fa-star fa-lg mr-2"></span></div>
                    <Card.Body>
                        <LocalForm onSubmit={generatePDF}>
                            <div className="form-group">
                                <Row><Col className="col-md-3 offset-md-1">
                                    <Form.Label>Name:</Form.Label></Col>
                                    <Col className="col-md-7">
                                        <Control.text
                                            model=".name"
                                            className="form-control"
                                            autoComplete="off"
                                            placeholder="Enter applicant's full name"
                                            name="name"
                                            value={name}
                                            onChange={(e) => setName(e.target.value)}
                                            validators={{
                                                required, validText, maxLength: maxLength(20),
 minLength: minLength(3)
                                            }}
                                        />
                                        <Errors
                                            className="text-danger"
                                            model=".name"
                                            show="touched"
```

```
                                    messages={{
                                        required: 'Required ',
                                        validText: 'Enter a valid Name!',
                                        maxLength: 'Length should be less than 15 chara
cters!',
                                        minLength: 'Length should be greater than 3 char
acters!'
                                    }}
                                />
                            </Col></Row>
                        </div>
                        <Row><Col>
                            <div className="form-group">
                                <Row><Col className="col-md-3 offset-md-
1"><Form.Label>Adhar Number:</Form.Label></Col>
                                    <Col className="col-md-7">
                                        <Control.text
                                            model=".adhar"
                                            autoComplete="off"
                                            className="form-control"
                                            placeholder="Adhar Number"
                                            name="Adhar Number"
                                            value={UID}
                                            onChange={(e) => setUID(e.target.value)}
                                            validators={{
                                                required, isNumber, maxLength: maxLength(1
2), minLength: minLength(12)
                                            }}
                                        />
                                        <Errors
                                        className="text-danger"
                                        model=".adhar"
                                        show="touched"
                                        messages={{
                                            required: 'Required ',
```

```
                                    isNumber: 'Enter a valid Number!',
                                    maxLength: 'Length should be less than 12 chara
cters!',
                                    minLength: 'Length should be exact 12 character
s!'
                                }}
                            />
                            </Col></Row>
                        </div></Col></Row>
                    <div className="text-center mt-4">
                        <Button variant="primary" type="submit">Apply</Button
>
                    </div>
                </LocalForm>
            </Card.Body>
        </Card></FadeTransform></Col>
    </Row>
</Container>
  )
}
export default Residence;
```

## Payment Receipt




## Source Code:

```
import React, { useState } from 'react';
import axiosInstance from '../../helpers/axios';
import { Control, LocalForm, Errors } from 'react-redux-form';
import PaymentImg from './payment receipt.jpg'
import { Container, Card, Form, Button, Row, Col } from 'react-bootstrap';
import { FadeTransform } from 'react-animation-components';
const required = (val) => val && val.length;
const maxLength = (len) => (val) => !(val) || (val.length <= len);
const minLength = (len) => (val) => val && (val.length >= len);
const isNumber = (val) => !isNaN(Number(val));
const validText = (val) => /^[a-zA-Z]+ [a-zA-Z]+$/i.test(val);
const PaymentReceipt = () => {
   const [name, setName] = useState("");
   const [email, setEmail] = useState("");

   const handleSubmit = e => {
      e.preventDefault();
      const paymentData = {
         name: name,
         email: email
```

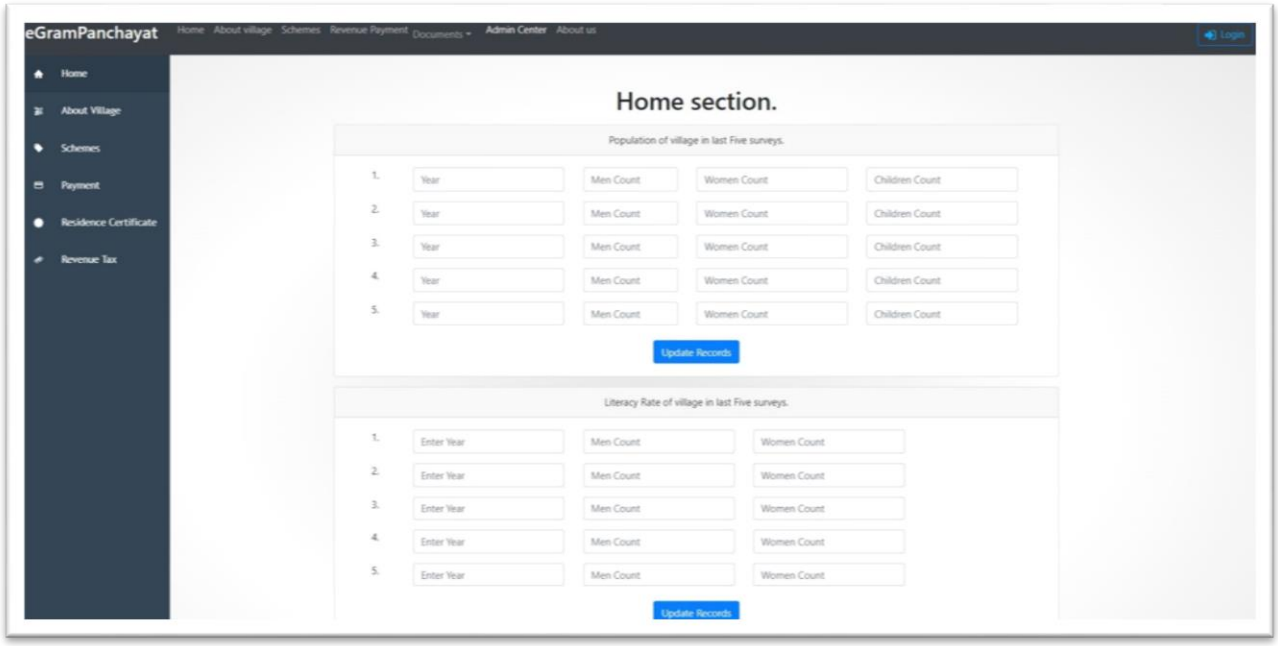
```
    }
  }
  return (
    <Container fluid className="mb-3">
      <Row className="justify-content-md-center">
        <Col className='col-md-5 mt-3' >
          <FadeTransform
            in
            transformProps={{
              exitTransform: 'scale(0.5) translateY(-50%)'
            }}>
            <Card className='frm'>
              <div className="text-center mt-4 mb-
4"><h1 className="">Payment receipt.</h1></div>
              <Card.Img varient="top" className="pic mt-1 col-md-6 col-sm-10 offset-md-
3" src={PaymentImg}></Card.Img>
              <div className="text-center mt-4 mb-4"><span className="fa fa-star fa-
lg mr-2"></span><span className="fa fa-star fa-lg mr-2"></span><span className="fa fa-
star fa-lg mr-2"></span><span className="fa fa-star fa-lg mr-
2"></span><span className="fa fa-star fa-lg mr-2"></span></div>
              <Card.Body>
                <LocalForm onSubmit={handleSubmit}>
                  <div className="form-group">
                    <Row><Col className="col-md-3 offset-md-1">
                      <Form.Label>Name:</Form.Label></Col>
                      <Col className="col-md-7">
                        <Control.text
                          model=".name"
                          className="form-control"
                          autoComplete="off"
                          placeholder="Enter applicant's full name"
                          name="name"
                          value={name}
                          onChange={(e) => setName(e.target.value)}
                          validators={{
                            required, validText, maxLength: maxLength(20), minLength:
minLength(3)
                          }}
                        />
                        <Errors
```

```
                        className="text-danger"
                        model=".name"
                        show="touched"
                        messages={{
                            required: 'Required ',
                            validText: 'Enter a valid Name!',
                            maxLength: 'Length should be less than 15 characters!',
                            minLength: 'Length should be greater than 3 characters!'
                        }}
                    />
                </Col></Row>
            </div>
            <div className="form-group">
                <Row><Col className="col-md-3 offset-md-1">
                    <Form.Label>Mobile No:</Form.Label></Col>
                    <Col className="col-md-7">
                        <Control.text  model=".mobNo"
                            className="form-control"
                            placeholder="Enter Mobile Number"
                            name="mobNo"
                            value={email}
                            onChange={(e) => setEmail(e.target.value)}
                            validators={{
                                required, isNumber, maxLength: maxLength(10), minLength:
minLength(10)
                            }}
                        />
                        <Errors
                            className="text-danger"
                            model=".mobNo"
                            show="touched"
                            messages={{
                                required: 'Required ',
                                isNumber: 'Enter a valid Number!',
                                maxLength: 'Length should be less than 10 characters!',
                                minLength: 'Length should be exact 10 characters!'
                            }}
                        />
                    </Col></Row>
            </div>
```

```
                        <div className="text-center mt-4">
                            <Button variant="primary" type="submit">Pay Now</Button>
                        </div>
                    </LocalForm>
                </Card.Body>
            </Card></FadeTransform></Col>
        </Row>
    </Container>
  )
}
export default PaymentReceipt;
```

## Admin center Module:

### Home section




## Source Code:

```
import React, { useState } from 'react';
import { Container, Row, Col,Card,Form,Button } from 'react-bootstrap';
import axiosInstance from '../../../helpers/axios';
import { store } from 'react-notifications-component';
import 'react-notifications-component/dist/theme.css';
import 'animate.css';
```

```
import { FadeTransform } from 'react-animation-components';
import Sidebar from '../Sidebar';
function AdHome() {
   const [yearOne, setyearOne] = useState();
   const [yearTwo, setyearTwo] = useState();
   const [yearThree, setyearThree] = useState();
   const [yearFour, setyearFour] = useState();
   const [yearFive, setyearFive] = useState();
   const [oneMen, setoneMen] = useState();
   const [twoMen, settwoMen] = useState();
   const [threeMen, setthreeMen] = useState();
   const [fourMen, setfourMen] = useState();
   const [fiveMen, setfiveMen] = useState();
   const [oneWomen, setoneWomen] = useState();
   const [twoWomen, settwoWomen] = useState();
   const [threeWomen, setthreeWomen] = useState();
   const [fourWomen, setfourWomen] = useState();
   const [fiveWomen, setfiveWomen] = useState();
   const [litYearOne, setlitYearOne] = useState();
   const [litYearTwo, setlitYearTwo] = useState();
   const [litYearThree, setlitYearThree] = useState();
   const [litYearFour, setlitYearFour] = useState();
   const [litYearFive, setlitYearFive] = useState();
   const [litOneMen, setlitOneMen] = useState();
   const [litTwoMen, setlitTwoMen] = useState();
   const [litThreeMen, setlitThreeMen] = useState();
   const [litFourMen, setlitFourMen] = useState();
   const [litFiveMen, setlitFiveMen] = useState();
   const [litOneWomen, setlitOneWomen] = useState();
   const [litTwoWomen, setlitTwoWomen] = useState();
   const [litThreeWomen, setlitThreeWomen] = useState();
   const [litFourWomen, setlitFourWomen] = useState();
   const [litFiveWomen, setlitFiveWomen] = useState();
   const [oneChildren, setoneChildren] = useState();
   const [twoChildren, settwoChildren] = useState();
   const [threeChildren, setthreeChildren] = useState();
   const [fourChildren, setfourChildren] = useState();
   const [fiveChildren, setfiveChildren] = useState();
   const  submitPopulation=e=>{
      const population={
```

```
        ID:101,
        years:[Number(yearOne),Number(yearTwo),Number(yearThree),Number(yearFour),Number(yearFive)],
        menCount:[Number(oneMen),Number(twoMen),Number(threeMen),Number(fourMen),Number(litFiveMen)],
        womenCount:[Number(oneWomen),Number(twoWomen),Number(threeWomen),Number(fourWomen),Number(fiveWomen)],
        childrenCount:[Number(oneChildren),Number(twoChildren),Number(threeChildren),Number(fourWomen),Number(fiveChildren)]
      }
      console.log(population)
      axiosInstance.post('populate/add', population);
      store.addNotification({
        title: 'Records updated successfully!!',
        message: 'Population related data updated!!',
        type: "success",
        container: 'top-right',
        animationIn: ["animated", "fadeIn"],
        animationOut: ["animated", "fadeOut"],
        dismiss: {
          duration: 3000,
          showIcon:true
        }
       })
  }
  const  submitLiteracy=e=>{
     const literacy={
        ID:101,
        years:[Number(litYearOne),Number(litYearTwo),Number(litYearThree),Number(litYearFour),Number(litYearFive)],
        menCount:[Number(litOneMen),Number(litTwoMen),Number(litThreeMen),Number(litFourMen),Number(litFiveMen)],
        womenCount:[Number(litOneWomen),Number(litTwoWomen),Number(litThreeWomen),Number(litFourWomen),Number(litFiveWomen)]
      }
      console.log(literacy)
      axiosInstance.post('literate/add', literacy);
      store.addNotification({
        title: 'Records updated successfully!!',
        message: 'Literacy related data updated!!',
```

```
        type: "success",
        container: 'top-right',
        animationIn: ["animated", "fadeIn"],
        animationOut: ["animated", "fadeOut"],
        dismiss: {
          duration: 3000,
          showIcon:true
        }
      })
  }
  return (
    <Container fluid className="m-0 p-0">
      <Row className="d-flex">
        <Col className="col-md-3">
          <Sidebar />
        </Col>
        <Col className="col-md-7 mt-5 mb-3 text-center">
          <h1>Home section.</h1>
          <FadeTransform
            in
            transformProps={{
              exitTransform: 'scale(0.5) translateY(-50%)'
            }}>
          <Card className="mt-3">
            <Card.Header>Population of village in last Five surveys.</Card.Header>
            <Card.Body>
              <Form>
                <Row className="col-md-12">
                  <Col className="col-md-1">
                    <Form.Label>1.</Form.Label>
                  </Col>
                  <Col className="col-md-3">
                    <Form.Control
                      type="text"
                      placeholder="Year"
                      autoComplete="off"
                      value={yearOne}
                      onChange={e => setyearOne(e.target.value)}
                    />
                  </Col>
```

```
                    <Col className="col-md-2">
                      <Form.Control
                        type="text"
                        placeholder="Men Count"
                        autoComplete="off"
                        value={oneMen}
                        onChange={e => setoneMen(e.target.value)}
                      />
                    </Col>
                    <Col className="col-md-3">
                      <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Women Count"
                        value={oneWomen}
                        onChange={e => setoneWomen(e.target.value)}
                      />
                    </Col>
                    <Col className="col-md-3">
                      <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Children Count"
                        value={oneChildren}
                        onChange={e => setoneChildren(e.target.value)}
                      />
                    </Col>
                  </Row>
                  <Row className="col-md-12 mt-3">
                    <Col className="col-md-1">
                      <Form.Label>2.</Form.Label>
                    </Col>
                    <Col className="col-md-3">
                      <Form.Control
                        type="text"
                        placeholder="Year"
                        autoComplete="off"
                        value={yearTwo}
                        onChange={e => setyearTwo(e.target.value)}
                      />
```

```
                    </Col>
                    <Col className="col-md-2">
                      <Form.Control
                        type="text"
                        placeholder="Men Count"
                        autoComplete="off"
                        value={twoMen}
                        onChange={e => settwoMen(e.target.value)}
                      />
                    </Col>
                    <Col className="col-md-3">
                      <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Women Count"
                        value={twoWomen}
                        onChange={e => settwoWomen(e.target.value)}
                      />
                    </Col>
                    <Col className="col-md-3">
                      <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Children Count"
                        value={twoChildren}
                        onChange={e => settwoChildren(e.target.value)}
                      />
                    </Col>
                  </Row>
                  <Row className="col-md-12 mt-3">
                    <Col className="col-md-1">
                      <Form.Label>3.</Form.Label>
                    </Col>
                    <Col className="col-md-3">
                      <Form.Control
                        type="text"
                        placeholder="Year"
                        autoComplete="off"
                        value={yearThree}
                        onChange={e => setyearThree(e.target.value)}
```

```
                />
            </Col>
            <Col className="col-md-2">
                <Form.Control
                    type="text"
                    placeholder="Men Count"
                    autoComplete="off"
                    value={threeMen}
                    onChange={e => setthreeMen(e.target.value)}
                />
            </Col>
            <Col className="col-md-3">
                <Form.Control
                    type="text"
                    autoComplete="off"
                    placeholder="Women Count"
                    value={threeWomen}
                    onChange={e => setthreeWomen(e.target.value)}
                />
            </Col>
            <Col className="col-md-3">
                <Form.Control
                    type="text"
                    autoComplete="off"
                    placeholder="Children Count"
                    value={threeChildren}
                    onChange={e => setthreeChildren(e.target.value)}
                />
            </Col>
        </Row>
        <Row className="col-md-12 mt-3">
            <Col className="col-md-1">
                <Form.Label>4.</Form.Label>
            </Col>
            <Col className="col-md-3">
                <Form.Control
                    type="text"
                    placeholder="Year"
                    autoComplete="off"
                    value={yearFour}
```

```
                onChange={e => setyearFour(e.target.value)}
              />
            </Col>
            <Col className="col-md-2">
              <Form.Control
                type="text"
                placeholder="Men Count"
                autoComplete="off"
                value={fourMen}
                onChange={e => setfourMen(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                autoComplete="off"
                placeholder="Women Count"
                value={fourWomen}
                onChange={e => setfourWomen(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                autoComplete="off"
                placeholder="Children Count"
                value={fourChildren}
                onChange={e => setfourChildren(e.target.value)}
              />
            </Col>
          </Row>
          <Row className="col-md-12 mt-3">
            <Col className="col-md-1">
              <Form.Label>5.</Form.Label>
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                placeholder="Year"
                autoComplete="off"
```

```
                                value={yearFive}
                                onChange={e => setyearFive(e.target.value)}
                            />
                        </Col>
                        <Col className="col-md-2">
                            <Form.Control
                                type="text"
                                placeholder="Men Count"
                                autoComplete="off"
                                value={fiveMen}
                                onChange={e => setfiveMen(e.target.value)}
                            />
                        </Col>
                        <Col className="col-md-3">
                            <Form.Control
                                type="text"
                                autoComplete="off"
                                placeholder="Women Count"
                                value={fiveWomen}
                                onChange={e => setfiveWomen(e.target.value)}
                            />
                        </Col>
                        <Col className="col-md-3">
                            <Form.Control
                                type="text"
                                autoComplete="off"
                                placeholder="Children Count"
                                value={fiveChildren}
                                onChange={e => setfiveChildren(e.target.value)}
                            />
                        </Col>
                    </Row>
                    <Button className="mt-
4" type="submit" onClick={submitPopulation}>Update Records</Button>
                </Form>
            </Card.Body>
        </Card></FadeTransform>
        <FadeTransform
            in
            transformProps={{
```

```
                exitTransform: 'scale(0.5) translateY(-50%)'
            }}>
          <Card className="mt-3">
            <Card.Header>Literacy Rate of village in last Five surveys.</Card.Header>
            <Card.Body>
              <Form>
                <Row className="col-md-12">
                  <Col className="col-md-1">
                    <Form.Label>1.</Form.Label>
                  </Col>
                  <Col className="col-md-3">
                    <Form.Control
                      type="text"
                      placeholder="Enter Year"
                      autoComplete="off"
                      value={litYearOne}
                      onChange={e => setlitYearOne(e.target.value)}
                    />
                  </Col>
                  <Col className="col-md-3">
                    <Form.Control
                      type="text"
                      placeholder="Men Count"
                      autoComplete="off"
                      value={litOneMen}
                      onChange={e => setlitOneMen(e.target.value)}
                    />
                  </Col>
                  <Col className="col-md-3">
                    <Form.Control
                      type="text"
                      autoComplete="off"
                      placeholder="Women Count"
                      value={litOneWomen}
                      onChange={e => setlitOneWomen(e.target.value)}
                    />
                  </Col>
                </Row>
                <Row className="col-md-12 mt-3">
                  <Col className="col-md-1">
```

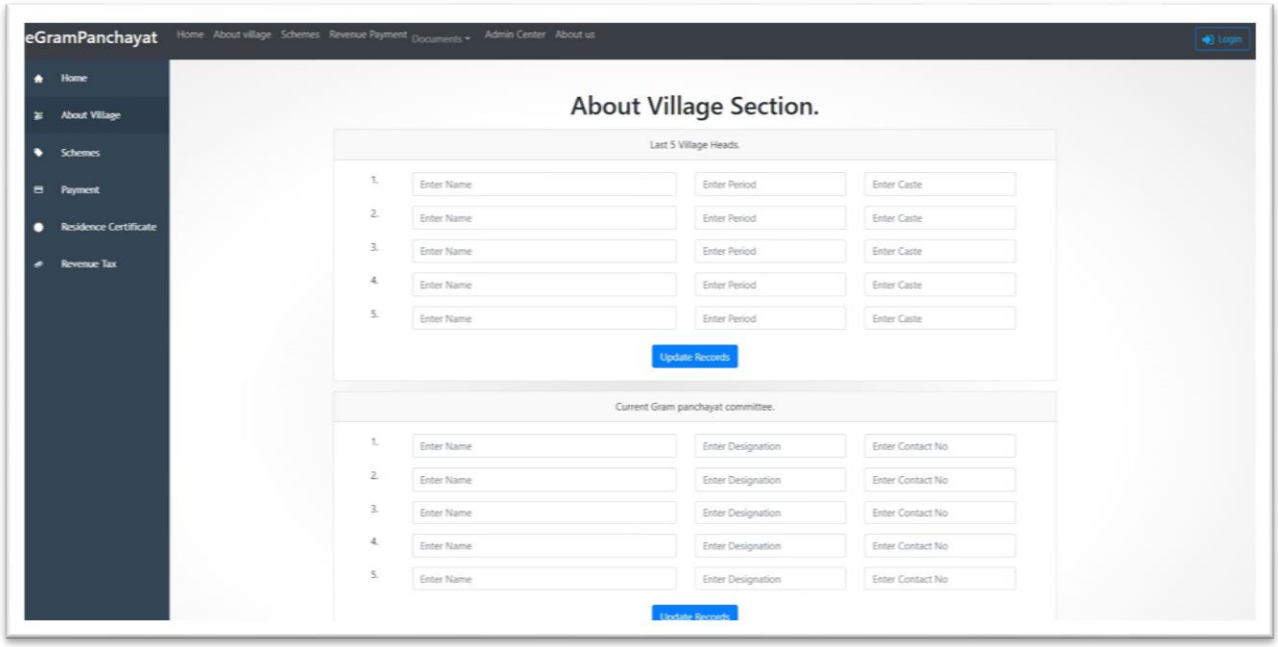
```
                    <Form.Label>2.</Form.Label>
                </Col>
                <Col className="col-md-3">
                    <Form.Control
                        type="text"
                        placeholder="Enter Year"
                        autoComplete="off"
                        value={litYearTwo}
                        onChange={e => setlitYearTwo(e.target.value)}
                    />
                </Col>
                <Col className="col-md-3">
                    <Form.Control
                        type="text"
                        placeholder="Men Count"
                        autoComplete="off"
                        value={litTwoMen}
                        onChange={e => setlitTwoMen(e.target.value)}
                    />
                </Col>
                <Col className="col-md-3">
                    <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Women Count"
                        value={litTwoWomen}
                        onChange={e => setlitTwoWomen(e.target.value)}
                    />
                </Col>
            </Row>
            <Row className="col-md-12 mt-3">
                <Col className="col-md-1">
                    <Form.Label>3.</Form.Label>
                </Col>
                <Col className="col-md-3">
                    <Form.Control
                        type="text"
                        placeholder="Enter Year"
                        autoComplete="off"
                        value={litYearThree}
```

```
                onChange={e => setlitYearThree(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                placeholder="Men Count"
                autoComplete="off"
                value={litThreeMen}
                onChange={e => setlitThreeMen(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                autoComplete="off"
                placeholder="Women Count"
                value={litThreeWomen}
                onChange={e => setlitThreeWomen(e.target.value)}
              />
            </Col>
          </Row>
          <Row className="col-md-12 mt-3">
            <Col className="col-md-1">
              <Form.Label>4.</Form.Label>
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                placeholder="Enter Year"
                autoComplete="off"
                value={litYearFour}
                onChange={e => setlitYearFour(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                placeholder="Men Count"
                autoComplete="off"
```

```
                value={litFourMen}
                onChange={e => setlitFourMen(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                autoComplete="off"
                placeholder="Women Count"
                value={litFourWomen}
                onChange={e => setlitFourWomen(e.target.value)}
              />
            </Col>
          </Row>
          <Row className="col-md-12 mt-3">
            <Col className="col-md-1">
              <Form.Label>5.</Form.Label>
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                placeholder="Enter Year"
                autoComplete="off"
                value={litYearFive}
                onChange={e => setlitYearFive(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                placeholder="Men Count"
                autoComplete="off"
                value={litFiveMen}
                onChange={e => setlitFiveMen(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                autoComplete="off"
```

```
                                placeholder="Women Count"
                                value={litFiveWomen}
                                onChange={e => setlitFiveWomen(e.target.value)}
                            />
                        </Col>
                    </Row>
                    <Button className="mt-
4" type="submit" onClick={submitLiteracy}>Update Records</Button>
                </Form>
            </Card.Body>
        </Card></FadeTransform>
    </Col>
</Row>
</Container>
  )
}
export default AdHome;
```

## About village section




## Source Code:

```
import React, { useState } from 'react'
import Sidebar from '../Sidebar';
import axiosInstance from '../../../helpers/axios';
import { Container, Row, Col, Card, Button, Form } from 'react-bootstrap';
import { store } from 'react-notifications-component';
import 'react-notifications-component/dist/theme.css';
import 'animate.css';
import { FadeTransform } from 'react-animation-components';
function AdVillage() {
    const [PrevFirstname, setPrevFirstname] = useState("");
    const [PrevSecondname, setPrevSecondname] = useState("");
    const [PrevThirdname, setPrevThirdname] = useState("");
    const [PrevFourthname, setPrevFourthname] = useState("");
    const [PrevFifthname, setPrevFifthname] = useState("");
    const [PrevFirstPeriod, setPrevFirstPeriod] = useState("");
    const [PrevSecondPeriod, setPrevSecondPeriod] = useState("");
    const [PrevThirdPeriod, setPrevThirdPeriod] = useState("");
    const [PrevFourthPeriod, setPrevFourthPeriod] = useState("");
    const [PrevFifthPeriod, setPrevFifthPeriod] = useState("");
    const [PrevFirstCaste, setPrevFirstCaste] = useState("");
    const [PrevSecondCaste, setPrevSecondCaste] = useState("");
    const [PrevThirdCaste, setPrevThirdCaste] = useState("");
    const [PrevFourthCaste, setPrevFourthCaste] = useState("");
    const [PrevFifthCaste, setPrevFifthCaste] = useState("");
    const [CurrFirstname, setCurrFirstname] = useState("");
    const [CurrSecondname, setCurrSecondname] = useState("");
    const [CurrThirdname, setCurrThirdname] = useState("");
    const [CurrFourthname, setCurrFourthname] = useState("");
    const [CurrFifthname, setCurrFifthname] = useState("");
    const [CurrFirstPeriod, setCurrFirstPeriod] = useState("");
    const [CurrSecondPeriod, setCurrSecondPeriod] = useState("");
    const [CurrThirdPeriod, setCurrThirdPeriod] = useState("");
    const [CurrFourthPeriod, setCurrFourthPeriod] = useState("");
    const [CurrFifthPeriod, setCurrFifthPeriod] = useState("");
    const [CurrFirstCaste, setCurrFirstCaste] = useState("");
    const [CurrSecondCaste, setCurrSecondCaste] = useState("");
    const [CurrThirdCaste, setCurrThirdCaste] = useState("");
    const [CurrFourthCaste, setCurrFourthCaste] = useState("");
    const [CurrFifthCaste, setCurrFifthCaste] = useState("");
    const submitCurrRecord = (e) => {
```

```
      const current={
         ID:101,
         Name:[CurrFirstname,CurrSecondname,CurrThirdname,CurrFourthname,CurrFifthname],
         designation:[CurrFirstPeriod,CurrSecondPeriod,CurrThirdPeriod,CurrFourthPeriod,CurrFi
fthPeriod],
         contact:[CurrFirstCaste,CurrSecondCaste,CurrThirdCaste,CurrFourthCaste,CurrFifthCaste
]
      }
      console.log(current)
      axiosInstance.post('currCommittee/addData', current);
      store.addNotification({
         title: 'Records updated successfully!!',
         message: 'current committee data is updated!!',
         type: "success",
         container: 'top-right',
         animationIn: ["animated", "fadeIn"],
         animationOut: ["animated", "fadeOut"],
         dismiss: {
           duration: 3000,
           showIcon:true
         }
        })
   }
   const submitPrevRecord = (e) => {
      // e.preventDefault();
      const previous={
         ID:102,
         Name:[PrevFirstname,PrevSecondname,PrevThirdname,PrevFourthname,PrevFifthname],
         workingPeriod:[PrevFirstPeriod,PrevSecondPeriod,PrevThirdPeriod,PrevFourthPeriod,Pre
vFifthPeriod],
         Caste:[PrevFirstCaste,PrevSecondCaste,PrevThirdCaste,PrevFourthCaste,PrevFifthCaste]
      }
      console.log(previous)
      axiosInstance.post('prevCommittee/addData', previous);
      store.addNotification({
         title: 'Records updated successfully!!',
         message: 'Previous committee data is updated!!',
         type: "success",
         container: 'top-right',
         animationIn: ["animated", "fadeIn"],
```

```
        animationOut: ["animated", "fadeOut"],
        dismiss: {
          duration: 3000,
          showIcon:true
        }
      })
  }
  return (
    <Container fluid className="m-0 p-0">
      <Row className="d-flex">
        <Col className="col-md-3">
          <Sidebar />
        </Col>
        <Col className="col-md-7 mt-5 mb-3 text-center">
          <h1 className="">About Village Section.</h1>
          <FadeTransform
            in
            transformProps={{
              exitTransform: 'scale(0.5) translateY(-50%)'
            }}>
          <Card className="mt-3">
            <Card.Header>Last 5 Village Heads.</Card.Header>
            <Card.Body>
              <Form>
                <Row className="col-md-12">
                  <Col className="col-md-1">
                    <Form.Label>1.</Form.Label>
                  </Col>
                  <Col className="col-md-5">
                    <Form.Control
                      type="text"
                      placeholder="Enter Name"
                      autoComplete="off"
                      value={PrevFirstname}
                      onChange={e=>setPrevFirstname(e.target.value)}
                    />
                  </Col>
                  <Col className="col-md-3">
                    <Form.Control
                      type="text"
```

```
                placeholder="Enter Period"
                autoComplete="off"
                value={PrevFirstPeriod}
                onChange={e=>setPrevFirstPeriod(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                autoComplete="off"
                placeholder="Enter Caste"
                value={PrevFirstCaste}
                onChange={e=>setPrevFirstCaste(e.target.value)}
                />
            </Col>
          </Row>
          <Row className="col-md-12 mt-3">
            <Col className="col-md-1">
              <Form.Label>2.</Form.Label>
            </Col>
            <Col className="col-md-5">
              <Form.Control
                type="text"
                placeholder="Enter Name"
                autoComplete="off"
                value={PrevSecondname}
                onChange={e=>setPrevSecondname(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                placeholder="Enter Period"
                autoComplete="off"
                value={PrevSecondPeriod}
                onChange={e=>setPrevSecondPeriod(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
```

```
                type="text"
                autoComplete="off"
                placeholder="Enter Caste"
                value={PrevSecondCaste}
                onChange={e=>setPrevSecondCaste(e.target.value)}
              />
            </Col>
          </Row>
          <Row className="col-md-12 mt-3">
            <Col className="col-md-1">
              <Form.Label>3.</Form.Label>
            </Col>
            <Col className="col-md-5">
              <Form.Control
                type="text"
                placeholder="Enter Name"
                autoComplete="off"
                value={PrevThirdname}
                onChange={e=>setPrevThirdname(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                autoComplete="off"
                placeholder="Enter Period"
                value={PrevThirdPeriod}
                onChange={e=>setPrevThirdPeriod(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control type="text"
              autoComplete="off"
                placeholder="Enter Caste"
                value={PrevThirdCaste}
                onChange={e=>setPrevThirdCaste(e.target.value)}
              />
            </Col>
          </Row>
          <Row className="col-md-12 mt-3">
```

```
                    <Col className="col-md-1">
                      <Form.Label>4.</Form.Label>
                    </Col>
                    <Col className="col-md-5">
                      <Form.Control
                        type="text"
                        placeholder="Enter Name"
                        autoComplete="off"
                        value={PrevFourthname}
                        onChange={e=>setPrevFourthname(e.target.value)}
                      />
                    </Col>
                    <Col className="col-md-3">
                      <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Enter Period"
                        value={PrevFourthPeriod}
                        onChange={e=>setPrevFourthPeriod(e.target.value)}
                      />
                    </Col>
                    <Col className="col-md-3">
                      <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Enter Caste"
                        value={PrevFourthCaste}
                        onChange={e=>setPrevFourthCaste(e.target.value)}
                      /></Col>
                  </Row>
                  <Row className="col-md-12 mt-3">
                    <Col className="col-md-1"><Form.Label>5.</Form.Label></Col>
                    <Col className="col-md-5">
                      <Form.Control
                        type="text"
                        placeholder="Enter Name"
                        value={PrevFifthname}
                        autoComplete="off"
                        onChange={e=>setPrevFifthname(e.target.value)}
                      />
```

```
                        </Col>
                        <Col className="col-md-3">
                          <Form.Control
                            type="text"
                            autoComplete="off"
                            placeholder="Enter Period"
                            value={PrevFifthPeriod}
                            onChange={e=>setPrevFifthPeriod(e.target.value)}
                          />
                        </Col>
                        <Col className="col-md-3">
                          <Form.Control type="text"
                          autoComplete="off"
                            placeholder="Enter Caste"
                            value={PrevFifthCaste}
                            onChange={e=>setPrevFifthCaste(e.target.value)}
                          />
                        </Col>
                      </Row>
                      <Button className="mt-
4" type="submit" onClick={submitPrevRecord}>Update Records</Button>
                    </Form>
                  </Card.Body>
                </Card></FadeTransform>
                <FadeTransform
                  in
                  transformProps={{
                    exitTransform: 'scale(0.5) translateY(-50%)'
                  }}>
                <Card className="mt-3">
                  <Card.Header>Current Gram panchayat committee.</Card.Header>
                  <Card.Body>
                  <Form>
                      <Row className="col-md-12">
                        <Col className="col-md-1">
                          <Form.Label>1.</Form.Label>
                        </Col>
                        <Col className="col-md-5">
                          <Form.Control
                            type="text"
```

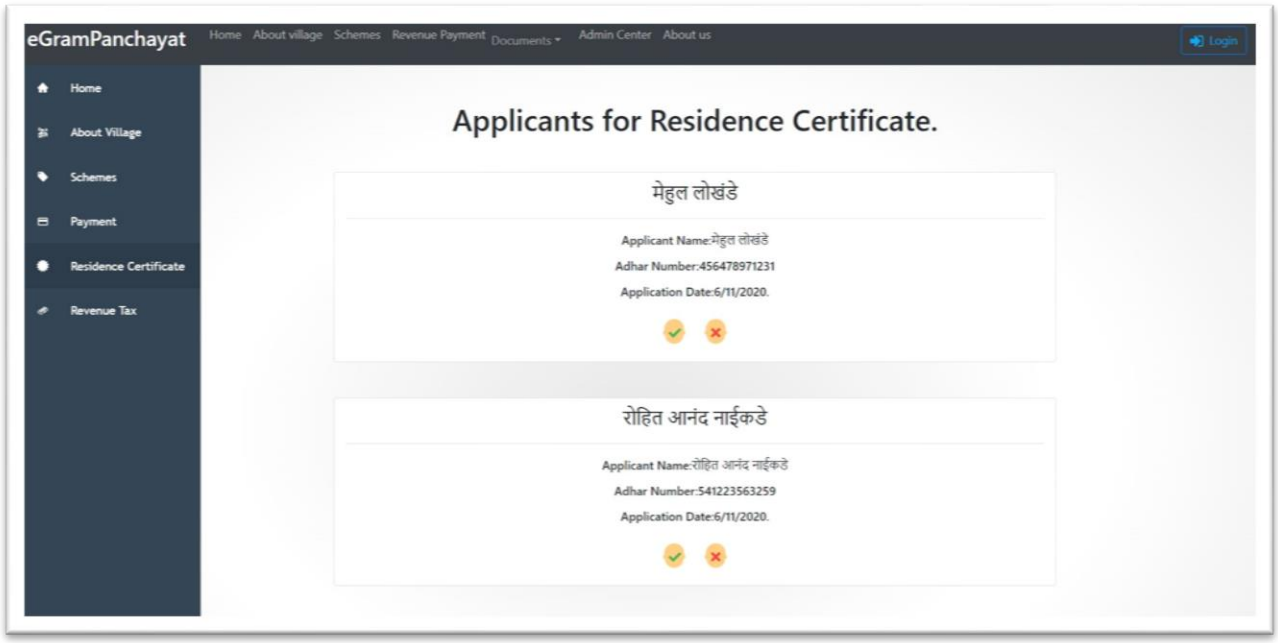
```
                placeholder="Enter Name"
                autoComplete="off"
                value={CurrFirstname}
                onChange={e=>setCurrFirstname(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                placeholder="Enter Designation"
                autoComplete="off"
                value={CurrFirstPeriod}
                onChange={e=>setCurrFirstPeriod(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
                type="text"
                autoComplete="off"
                placeholder="Enter Contact No"
                value={CurrFirstCaste}
                onChange={e=>setCurrFirstCaste(e.target.value)}
                />
            </Col>
          </Row>
          <Row className="col-md-12 mt-3">
            <Col className="col-md-1">
              <Form.Label>2.</Form.Label>
            </Col>
            <Col className="col-md-5">
              <Form.Control
                type="text"
                placeholder="Enter Name"
                autoComplete="off"
                value={CurrSecondname}
                onChange={e=>setCurrSecondname(e.target.value)}
              />
            </Col>
            <Col className="col-md-3">
              <Form.Control
```

```
                        type="text"
                        placeholder="Enter Designation"
                        autoComplete="off"
                        value={CurrSecondPeriod}
                        onChange={e=>setCurrSecondPeriod(e.target.value)}
                    />
                </Col>
                <Col className="col-md-3">
                    <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Enter Contact No"
                        value={CurrSecondCaste}
                        onChange={e=>setCurrSecondCaste(e.target.value)}
                    />
                </Col>
            </Row>
            <Row className="col-md-12 mt-3">
                <Col className="col-md-1">
                    <Form.Label>3.</Form.Label>
                </Col>
                <Col className="col-md-5">
                    <Form.Control
                        type="text"
                        placeholder="Enter Name"
                        autoComplete="off"
                        value={CurrThirdname}
                        onChange={e=>setCurrThirdname(e.target.value)}
                    />
                </Col>
                <Col className="col-md-3">
                    <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Enter Designation"
                        value={CurrThirdPeriod}
                        onChange={e=>setCurrThirdPeriod(e.target.value)}
                    />
                </Col>
                <Col className="col-md-3">
```

```
                    <Form.Control type="text"
                    autoComplete="off"
                        placeholder="Enter Contact No"
                        value={CurrThirdCaste}
                        onChange={e=>setCurrThirdCaste(e.target.value)}
                    />
                </Col>
            </Row>
            <Row className="col-md-12 mt-3">
                <Col className="col-md-1">
                    <Form.Label>4.</Form.Label>
                </Col>
                <Col className="col-md-5">
                    <Form.Control
                        type="text"
                        placeholder="Enter Name"
                        autoComplete="off"
                        value={CurrFourthname}
                        onChange={e=>setCurrFourthname(e.target.value)}
                    />
                </Col>
                <Col className="col-md-3">
                    <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Enter Designation"
                        value={CurrFourthPeriod}
                        onChange={e=>setCurrFourthPeriod(e.target.value)}
                    />
                </Col>
                <Col className="col-md-3">
                    <Form.Control
                        type="text"
                        autoComplete="off"
                        placeholder="Enter Contact No"
                        value={CurrFourthCaste}
                        onChange={e=>setCurrFourthCaste(e.target.value)}
                    /></Col>
            </Row>
            <Row className="col-md-12 mt-3">
```

```
                        <Col className="col-md-1"><Form.Label>5.</Form.Label></Col>
                        <Col className="col-md-5">
                          <Form.Control
                            type="text"
                            placeholder="Enter Name"
                            value={CurrFifthname}
                            autoComplete="off"
                            onChange={e=>setCurrFifthname(e.target.value)}
                          />
                        </Col>
                        <Col className="col-md-3">
                          <Form.Control
                            type="text"
                            autoComplete="off"
                            placeholder="Enter Designation"
                            value={CurrFifthPeriod}
                            onChange={e=>setCurrFifthPeriod(e.target.value)}
                          />
                        </Col>
                        <Col className="col-md-3">
                          <Form.Control type="text"
                          autoComplete="off"
                            placeholder="Enter Contact No"
                            value={CurrFifthCaste}
                            onChange={e=>setCurrFifthCaste(e.target.value)}
                          />
                        </Col>
                      </Row>
                      <Button className="mt-
4" type="submit" onClick={submitCurrRecord}>Update Records</Button>
                    </Form>
                  </Card.Body>
                </Card></FadeTransform>
              </Col>
            </Row>
          </Container>
    )
}
export default AdVillage;
```

## Residence certificate




## Source Code:

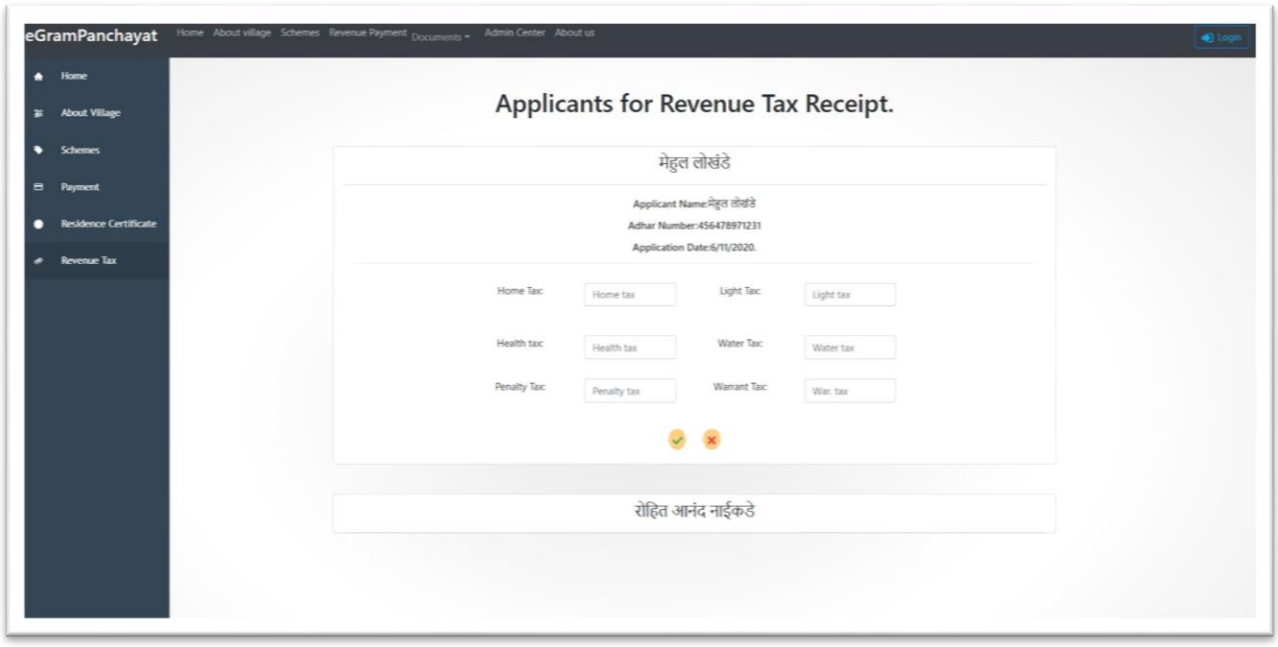
```
import React, { useEffect } from 'react';
import { useDispatch, useSelector } from 'react-redux';
import Sidebar from '../Sidebar';
import { Container, Row, Col, Card, Accordion } from 'react-bootstrap';
import { readApplicants } from '../../../Redux/actions/residenceActions';
import { Stagger } from 'react-animation-components';
import * as FcIcons from "react-icons/fc";
import axiosInstance from '../../../helpers/axios';
import { store } from 'react-notifications-component';
import 'react-notifications-component/dist/theme.css';
import 'animate.css';
import { BeatLoader } from 'react-spinner';
import { jsx, css } from "@emotion/react";
import './pages.css';
function AdResidence() {
   const dispatch = useDispatch();
   useEffect(() => {
      dispatch(readApplicants())
   }, [])
   const override = css`
```

```
    display: block;
    margin: 0 auto;
    border-color: red;
`;
const applicants = useSelector(state => state.residence);
let cards = "";
const Approve = (name, UID) => {
    const residenceData = {
        name: name,
        UID: Number(UID),
    }
    axiosInstance.post('residence/download', residenceData)
    store.addNotification({
        title: 'Application Approved!',
        message: 'Residence certificate generated sucessfully!',
        type: "info",
        container: 'top-right',
        animationIn: ["animated", "fadeIn"],
        animationOut: ["animated", "fadeOut"],
        dismiss: {
            duration: 3000,
            showIcon: true
        }
        // .then(() => axiosInstance.get('residence/download', { responseType: 'blob' }))
        // .then((res) => {
        //     const pdfBlob = new Blob([res.data], { type: 'application/pdf' });
        //     saveAs(pdfBlob, 'generatedDocument.pdf')
        //   })
    })
    window.location.reload(false);
}
const Reject = (UID) => {
    const residenceData = {
        UID: Number(UID),
    }
    console.log(residenceData);
    axiosInstance.post('residence/reject', residenceData)
    store.addNotification({
        title: 'Application Rejected!',
        message: 'Send Notification to villager.',
```

```
            type: "danger",
            container: 'top-right',
            animationIn: ["animated", "fadeIn"],
            animationOut: ["animated", "fadeOut"],
            dismiss: {
              duration: 3000,
              showIcon: true
            }
          })
          window.location.reload(false);
        }
        if (applicants?.applicants?.data) {
          const activeKey = applicants?.applicants?.data[0]._id;
          cards = applicants?.applicants.data.map((data, id) => {
            return (
              <Card className="col-md-12 col-sm-12 mt-5">
                <Accordion className="myAccordian" defaultActiveKey={activeKey}>
                  <Accordion.Toggle as={Card.Header} className="back" eventKey={data._id}>
                    <h4>{data.name}</h4>
                  </Accordion.Toggle>
                  <Accordion.Collapse eventKey={data._id}>
                    <Card.Body>
                      <h6 className="">Applicant Name:{data.name}</h6>
                      <h6 className="mt-3">Adhar Number:{data.UID}</h6>
                      <h6 className="mt-
3">Application Date:{new Date().getDate()}/{new Date().getMonth()}/{new Date().getFullYear(
)}.</h6>
                      <div className="mt-4">
                        <FcIcons.FcApprove className="icons" size={40} onClick={((e) => Appr
ove(data.name, data.UID))} />
                        <FcIcons.FcDisapprove className="ml-
3 icons" size={40} onClick={((e) => Reject(data.UID))} />
                        {/* <MdIcons.MdDelete className="ml-3 icons" size={30} /> */}
                      </div>
                    </Card.Body>

                  </Accordion.Collapse>
                </Accordion>
              </Card>
            )
```

```
      })
    }
    if(applicants){
      return (
        <Container fluid className="m-0 p-0">
          <Row className="d-flex">
            <Col className="col-md-3">
              <Sidebar />
            </Col>
            <Col className="col-md-7 mt-5 mb-3 text-center">
              <h1>Applicants for Residence Certificate.</h1>
              <Stagger in><div>{cards}</div></Stagger>
            </Col>
          </Row>
        </Container>
      )
    }else{
      return(
        <BeatLoader
        css={override}
        size={150}
        color={"#123abc"}
        loading
      />
      )
    }
}
export default AdResidence;
```

## Revenue tax




## Source Code:

```
import React, { useEffect,useState } from 'react';
import { useDispatch, useSelector } from 'react-redux';
import Sidebar from '../Sidebar';
import * as FcIcons from "react-icons/fc";
import { readRevenue } from '../../../Redux/actions/revenueActions';
import { Container, Row, Card, Col, Accordion,Form } from 'react-bootstrap';
import { FadeTransform, Fade, Stagger } from 'react-animation-components';
import axiosInstance from '../../../helpers/axios';
import { store } from 'react-notifications-component';
import 'react-notifications-component/dist/theme.css';
import 'animate.css';
function AdRevenue() {
    const [home, setHome] = useState();
    const [water, setWater] = useState();
    const [health, setHealth] = useState();
    const [light, setLight] = useState();
    const [penalty, setPenalty] = useState();
    const [warrant, setWarrant] = useState();
    const dispatch = useDispatch();
    useEffect(() => {
```

```
    dispatch(readRevenue())
}, [])
const revenue = useSelector(state => state.revenue);
let cards = "";
const generatePDF=(name,uid)=>{
    const revenueData = {
        name:name,
        UID:uid,
        home_tax: Number(home),
        water_tax: Number(water),
        health_tax: Number(health),
        light_tax: Number(light),
        penalty_tax: Number(penalty),
        warrant_tax: Number(warrant),
        date:new Date()
    }
    axiosInstance.post('revenue/download', revenueData);
    store.addNotification({
        title: 'Application Approved!!',
        message: 'Residence certificate generated sucessfully!',
        type: "success",
        container: 'top-right',
        animationIn: ["animated", "fadeIn"],
        animationOut: ["animated", "fadeOut"],
        dismiss: {
          duration: 3000,
          showIcon:true
        }
      })
      window.location.reload(false);
}
const Reject = (UID) => {
    const revenueData = {
        UID: Number(UID),
    }
    axiosInstance.post('revenue/reject', revenueData)
    store.addNotification({
        title: 'Application Rejected!',
        message: 'Send Notification to villager.',
        type: "danger",
```

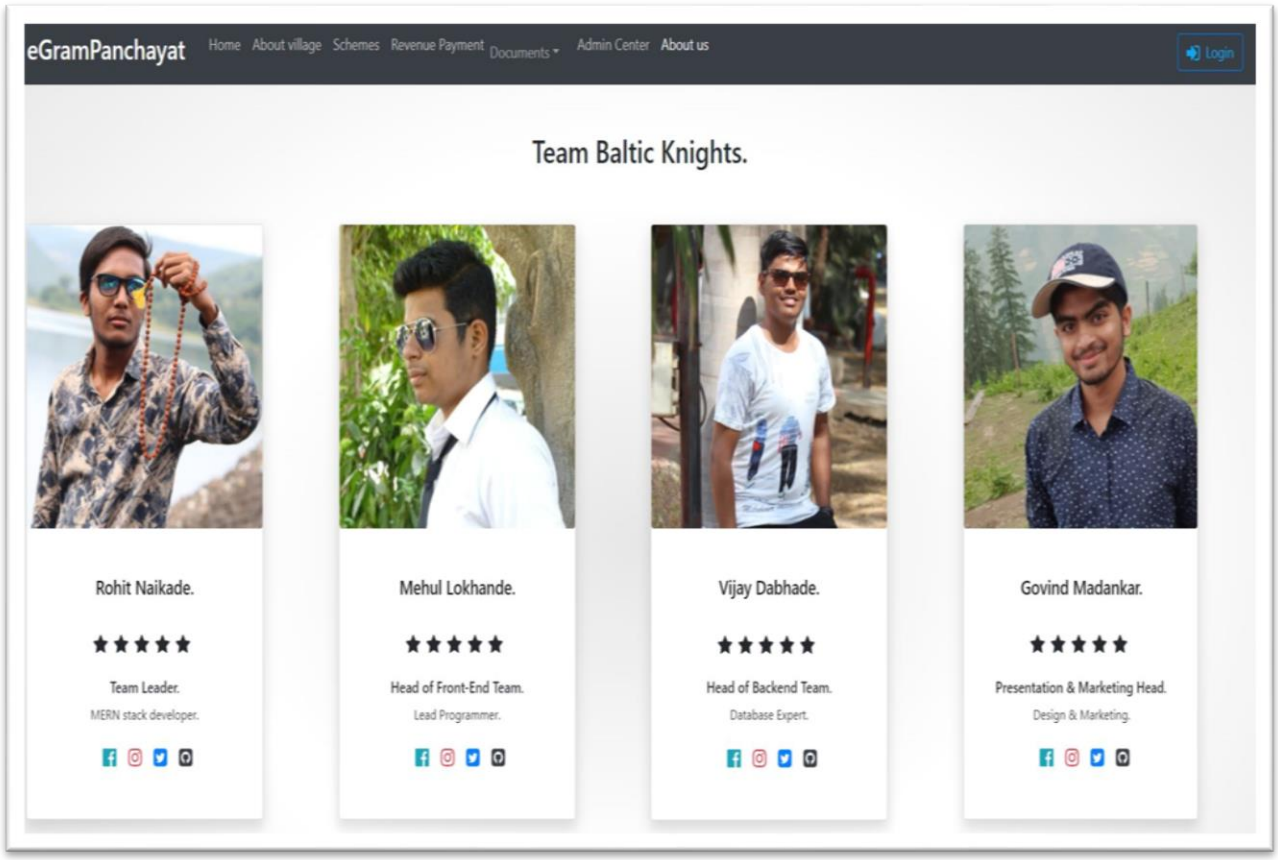
```
        container: 'top-right',
        animationIn: ["animated", "fadeIn"],
        animationOut: ["animated", "fadeOut"],
        dismiss: {
          duration: 3000,
          showIcon: true
        }
      })
      window.location.reload(false);
    }
    if (revenue?.revenueData?.data) {
      const activeKey = revenue?.revenueData?.data[0]._id;
      cards = revenue?.revenueData?.data.map((data, id) => {
        return (
          <Card className="col-md-12 col-sm-12 mt-5">
            <Accordion className="myAccordian" defaultActiveKey={activeKey}>
              <Accordion.Toggle as={Card.Header} className="back" eventKey={data._id}>
                <h4>{data.name}</h4>
              </Accordion.Toggle>
              <Accordion.Collapse eventKey={data._id}>
                <Card.Body>
                  <h6 className="">Applicant Name:{data.name}</h6>
                  <h6 className="mt-3">Adhar Number:{data.UID}</h6>
                  <h6 className="mt-
3">Application Date:{new Date().getDate()}/{new Date().getMonth()}/{new Date().getFullYear(
)}.</h6>
                  <hr />
                  <Form className="">
                    <Row><Col>
                      <Form.Group controlId="formGroupPassword">
                        <Row className="col-md-12 mt-3 d-flex justify-content-md-center">
                          <Col className="col-md-
2"><Form.Label>Home Tax:</Form.Label></Col>
                          <Col className="col-md-2">
                            <Form.Control type="tax"
                              autoComplete="off"
                              placeholder="Home tax"
                              name="home tax"
                              value={home}
                              onChange={(e) => setHome(e.target.value)}
```

```
                                    />
                                </Col>
                                <Col className="col-md-
2"><Form.Label>Light Tax:</Form.Label></Col>
                                <Col className="col-md-2">
                                    <Form.Control type="tax"
                                        autoComplete="off"
                                        placeholder="Light tax"
                                        name="Light tax"
                                        value={light}
                                        onChange={(e) => setLight(e.target.value)}
                                    />
                                </Col></Row>
                            </Form.Group></Col></Row>
                        <Row><Col>
                            <Form.Group controlId="formGroupPassword">
                                <Row></Row>
                            </Form.Group></Col></Row>
                        <Row><Col>
                            <Form.Group controlId="formGroupFile">
                                <Row className="col-md-12 mt-3 d-flex justify-content-md-center">
                                    <Col className="col-md-
2"><Form.Label>Health tax:</Form.Label></Col>
                                    <Col className="col-md-2">
                                        <Form.Control type="tax"
                                            autoComplete="off"
                                            placeholder="Health tax"
                                            name="Health tax"

                                            value={health}
                                            onChange={(e) => setHealth(e.target.value)}
                                        />
                                    </Col>
                                    <Col className="col-md-
2"><Form.Label>Water Tax:</Form.Label></Col>
                                    <Col className="col-md-2">
                                        <Form.Control type="tax"
                                            autoComplete="off"
                                            placeholder="Water tax"
                                            name="Water tax"
```

```
                                    value={water}
                                    onChange={(e) => setWater(e.target.value)}
                                />
                            </Col></Row>
                        </Form.Group></Col></Row>
                    <Row><Col>
                        <Form.Group controlId="formGroupFile">
                            <Row className="col-md-12 mt-3 d-flex justify-content-md-center">
                                <Col className="col-md-
2"><Form.Label>Penalty Tax:</Form.Label></Col>
                                <Col className="col-md-2">
                                    <Form.Control type="tax"
                                    autoComplete="off"
                                    placeholder="Penalty tax"
                                    name="Pen. tax"

                                    value={penalty}
                                    onChange={(e) => setPenalty(e.target.value)}
                                />
                                </Col>
                                <Col className="col-md-2 float-
right"><Form.Label>Warrant Tax:</Form.Label></Col>
                                <Col className="col-md-2">
                                    <Form.Control type="tax"
                                    autoComplete="off"
                                    placeholder="War. tax"
                                    name="Warrant tax"

                                    value={warrant}
                                    onChange={(e) => setWarrant(e.target.value)}
                                />
                                </Col></Row>
                        </Form.Group></Col></Row>
                    <div className="mt-4">
                    <FcIcons.FcApprove className="icons" size={40} onClick={((e) => gene
ratePDF(data.name, data.UID))} />
                    <FcIcons.FcDisapprove className="ml-
3 icons" size={40}  onClick={((e) => Reject(data.UID))} />                    </div>
                </Form>
```

```
                    </Card.Body>
                  </Accordion.Collapse>
                </Accordion>
              </Card>
            )
          })
    }
    return (
      <Container fluid className="m-0 p-0">
        <Row className="d-flex">
          <Col className="col-md-3">
            <Sidebar />
          </Col>
          <Col className="col-md-7 mt-5 mb-3 text-center">
            <h1>Applicants for Revenue Tax Receipt.</h1>
            <Stagger in><div>{cards}</div></Stagger>
          </Col>
        </Row>
      </Container>
    )
}
export default AdRevenue;
```

## About us Module




## Source Code:

```
import React from "react";
import mehul from './images/mehul.jpg';
import rohit from './images/rohit.jpg';
import vijay from './images/vijay.jpg';
import govind from './images/govind.jpg';
import Cards from './card';
import './card-style.css';
import { Row, Col, Container } from 'react-bootstrap';
class Village extends React.Component {
  render() {
    return (
      <Container fluid className="mb-5">
        <Row className="d-flex justify-content-center text-center">
          <Col className="mt-5">
            <h2 className="head">Team Baltic Knights.</h2>
```

```
        </Col>
      </Row>
      <Row className="d-flex justify-content-center text-center mt-5">
        <Col className="col-md-3 col-sm-10 col-xs-10">
          <Cards imgsrc={rohit}
            name="Rohit Naikade."
            designation="Team Leader."
            profile="MERN stack developer."
            github="https://github.com/RohitNaikade264"
            instagram="https://www.instagram.com/bhole_sarkar._/"
            twitter="https://twitter.com/BholeSarkar3"
            facebook="https://www.facebook.com/rohit.naikade.37"
          />
        </Col>
        <Col className="col-md-3 col-sm-10 col-xs-10">
          <Cards imgsrc={mehul}
            name="Mehul Lokhande."
            designation="Head of Front-End Team."
            profile="Lead Programmer."
            github="https://github.com/MEHUL25"
            instagram="https://www.instagram.com/mehullokhande.js/"
            twitter=""
            facebook="https://m.facebook.com/mehul.lokhande.39?ref=bookmarks"
          />
        </Col>
        <Col className="col-md-3 col-sm-10 col-xs-10">
          <Cards imgsrc={vijay}
          name="Vijay Dabhade."
          designation="Head of Backend Team."
          profile="Database Expert."
          github=""
          instagram=""
          twitter=""
          facebook="" />
        </Col>
        <Col className="col-md-3 col-sm-10 col-xs-10">
          <Cards imgsrc={govind}
          name="Govind Madankar."
          designation="Presentation & Marketing Head."
          profile="Design & Marketing."
```

```
          github=""
          instagram=""
          twitter=""
          facebook=""/>
        </Col>
      </Row>
    </Container>
  );
 }
}
export default Village;
```

# Test Cases:

# Login Module:

## Functional test cases:

| Sr.No. | Functional Test Cases. | Positive/Negative. |
|---|---|---|
| 1. | Verify if a user will be able to login with a valid username and valid password. | Positive |
| 2. | Verify if a user cannot login with a valid username and an invalid password. | Negative |
| 3. | Verify the login page for both, when the field is blank and Submit button is clicked. | Negative |
| 4. | Verify the 'Forgot Password' functionality. | Positive |
| 5. | Verify the messages for invalid login. | Positive |
| 6. | Verify if the data in password field is either visible as asterisk or bullet signs. | Positive |
| 7. | Check whether user can login with google or facebook third parties. | Positive |
| 8. | Verify if a user is able to login with a new password only after he/she has changed the password. | Positive |

## Non-functional test cases:

| Sr.No. | Non-functional test cases. | Positive/negative |
|---|---|---|
| 1 | Verify if a user cannot enter the characters more than the specified range in each field (email id and Password). | Negative |
| 2 | Verify the login page by pressing 'Back button' of the browser. It should not allow you to enter into the system once you log out. | Negative |
| 3 | Verify the timeout functionality of the login session. | Positive |
| 4 | Verify if a user should not be allowed to log in with different credentials from the same browser at the same time. | Negative |
| 5 | Verify if a user should be able to login with the same credentials in different browsers at the same time. | Positive |
| 6 | Verify the Login page against SQL injection attack. | Negative |
| 7 | Verify the implementation of SSL certificate. | Positive |

# Register Module:

| Sr.No. | Test Cases. | Positive/Negative |
|---|---|---|
| 1 | Verify that all the specified fields are present on the registration page. | Positive |
| 2 | Verify that clicking submits button after entering all the required fields, submits the data to the server. | Positive |
| 3 | Verify that whenever possible validation should take place at the client-side. | Positive |
| 4 | Verify that not filling the mandatory fields and clicking the submit button will lead to a validation error. | Negative |
| 5 | Check the upper limit of the textboxes. | Positive |
| 6 | Check validation on numeric fields by entering alphabets and special characters. | Positive |
| 7 | Verify that leading and trailing spaces are trimmed. | Positive |

## Residence Certificate,Payment,revenue tax receipt Module:

| Sr.No. | Test cases. | Positive/Negative |
|---|---|---|
| 1 | Check whether it displays error if any field in empty. | positive |
| 2 | Check whether UID field is accepting less than 12 digits or not. | negative |
| 3 | Verify whether UID field is accepting characters. | positive |
| 4 | Check whether name field is accepting numbers. | positive |
| 5 | Form should not be submitted if any field in empty. | negative |

CHAPTER 8

# ADVANTAGES AND DISADVANTAGES

## 8.1Advantages

- **Time saving**
  - No need to visit again and again to panchayat office for residential certificate.

- **Easy Administration**
  - It will become easy for the higher bodies and government to monitor the functioning and development of the village.

- **Development of Village**
  - Due to competition, every village will strive for betterment of its image.

- **Awareness**
  - Villagers will be aware about benefits of govt. schemes.

- **Less data misplacement**
  - Because of less paper work,misplacment of data will reduce.

- **Secured and easy payment**
  - Villagers can pay revenue taxes from anywhere and anytime.

## Disadvantages/Limitations :

- The most common hurdles in implementing e-government projects are the poor connectivity in rural areas due to lack of telecom network and Internet Facility,
- Frequent power failures
- Insufficient funds to equip the village with the latest infrastructure
- Difficulty of illiterate people to cope with the new technology
- Rural sector data: Since there is little information available on rural sector, it makes it difficult to propose appropriate services in the system.
- Operational problems associated with Inadequate Infrastructure, dusty environment.
- Local languages: Since most rural population is English illiterate, user interfaces designed for local languages make adoption easier.
- Technology adoption: The rural people, at times, may not accept the project until they are convinced of benefits.

CHAPTER 9

# FUTURE SCOPE

- The Needy will be benefited with the schemes, as there will be no middlemen and hence no corruption.
- Villagers can concentrate on farming and their duties, as time will be saved of visiting the Gram Panchayat.
- The Central Government can directly keep in touch with the lowest level of democracy.
- This project can be scaled to bring the Gram Panchayat at the fingertips of the Villager.
- To digitalize and save the database online of Birth register, Death register, Property Records,Tax collection Account for Gram panchayat etc.
- Connecting the different Gram panchayats and their upper bodies.
- Every villager is able to get the all information about gram panchayats

CHAPTER 10

# CONCLUSION

- In the e-government projects described above (Table 1), most projects were designed adequately until the last phase (customisation) of design, while only a few reached the last phase of implementation (i.e. transact phase).

- This suggests that future projects should be designed and implemented with incremental approach in order to realise full benefits. Furthermore, there is little evidence of involvement of citizens in government decision-making (Marchionini et al., 2003).

- Employment of new technologies will enable governments to provide the above services more meaningfully. However, governments should carefully devise their regulatory procedures so that competition is encouraged and innovation is stimulated .

- E-Governance for panchayat provides online services to the people living in that panchayat. It helps for the people in that area to easily complete their work which involves the action of authority of the panchayat people.

- As everything is made online people can request their applications from anywhere at any time.

- Thus, eGram Panchayat will be an integrated platform, providing vibrant services for the villagers and making the services transparent. This initiative will surely help in bringing the Gram Panchayat at the fingertips of villagers.

# LITERATURE SURVEY

## 11.1 Literature Survey

1. Panchayati Raj System
2. **E-Gram Panchayat** : https://www.ijresm.com/Vol.3_2020/Vol3_Iss4_April20/IJRESM_V3_I4_85.pdf
3. **E-Gram Panchayat Management System** : https://www.ijresm.com/Vol.3_2020/Vol3_Iss5_May20/IJRESM_V3_I5_24.pdf
4. **Building an alternative e-governance model : lessons from e-Gram in Gujarat :** https://idl-bnc-idrc.dspacedirect.org/handle/10625/41775
5. **E-GRAM PANCHAYAT MANAGEMENT SYSTEM :** http://oaijse.com/
6. **Hurdles in rural e-government projects in India: lessons for developing countries** : http://citeseerx.ist.psu.edu

CHAPTER 11

# REFERENCES


- Government portal Aatmanirbhar Bharat.
- PMO portal.
- **nodejs** : https://nodejs.org/en/docs/
- **expressjs** : https://expressjs.com/
- **Reactjs** : https://reactjs.org/docs/getting-started.html
- **MongoDB** : https://docs.mongodb.com/