

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	TECOC 339
Mehul Lokhande	TECOC 331
Vijay Dabhade	TECOC 308
Govind Madankar	TECOC 332

Under The Guidance of

Prof. R.A. Patil



**DEPARTMENT OF COMPUTER ENGINEERING,
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
SECTOR 26, NIGDI, PRADHIKARAN**



PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that, the project entitled "**E-Gram Panchayat**" successfully submitted by following students of "PCET's Pimpri Chinchwad College of Engineering, Nigdi, Pune-44" as a part of mini project

Under the guidance of Prof. R.A. Patil

In the partial fulfillment of the requirements for the T.E. (Computer Engineering)

Rohit Naikade TECOC 339

Mehul Lokhande TECOC 331

Vijay Dabhade TECOC 308

Govind Madankar TECOC 332

Prof. Rahul A. Patil

Project Guide

ACKNOWLEDGEMENT

We feel immense pleasure in presenting this project report on “**E-Gram Panchayat**”. We wish to express true sense of gratitude towards **Prof. Rahul A. Patil**, our project guide who at very discrete step in study of this project, contributed his 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 TECOC 331,

Vijay Dabhade TECOC 308,

Govind Madankar TECOC 332,

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, and 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	VII
LIST OF FIGURES	VIII
LIST OF ABBREVIATIONS	X

	Topic	Page No.
1.	INTRODUCTION	1
1.1	Problem Statement	1
1.2	Problem Idea	2
1.3	Motivation	3
1.4	Scope	4
1.5	Literature Survey	4
2.	Project Design	5
2.1	H/W,S/W, resources, requirements and their detail explanation	5
2.2	E-R Model	16
2.3	Schema of all tables	16
2.4	Hours Estimation	26
3.	MODULE DESCRIPTION	27
3.1	Explanation of each module	27
4.	Results and discussion	31
4.1	Source Code and Screenshots including GUI	31

4.2	Test Cases	112
5.	CONCLUSION	115
	REFERENCES	116

List of Tables

Name of Table		Page No.
Table 5.1.1	Login Module – Functional Test Cases	112
Table 5.1.2	Login Module – Non - Functional Test Cases	113
Table 5.2	Register Module test cases	113
Table 5.3	Residence Certificate, Payment, revenue tax receipt Module test Cases	114

List of Figures

Name of Figure		Page No.
Figure 1	Panchayati Raj System	3
Figure 2.1	High Level View of Project	9
Figure 2.2	System Architecture	9
Figure 2.3	UML Use Case Diagram	10
Figure 2.4	Class Diagram	11
Figure 2.5	Sequence Diagram – Admin	12
Figure 2.6	Sequence Diagram – User	13
Figure 2.7	State Diagram	14
Figure 2.8	Activity Diagram	15
Figure 2.9	E-R Diagram	16
Figure 2.10	Database Schema	16-26
Figure 3.1	Block Diagram of Modules	27
Figure 4.1	Login Module	31
Figure 4.2	Register Module	37
Figure 4.3	Home Module	44
Figure 4.4	About Village Module	46
Figure 4.5	Schemes Module	53
Figure 4.6	Payment Gateway Module	55
Figure 4.7	Residence Certificate	62

Figure 4.8	Revenue Tax Receipt	66
Figure 4.9	Tax Payment	71
Figure 4.10.1	Admin Center – Home Section	74
Figure 4.10.2	Admin Center – About Village Section	88
Figure 4.10.3	Admin Center – Tax Receipt	99
Figure 4.10.4	Admin Center –Revenue Tax	1.3
Figure 4.11	About Us Module	109

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

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:

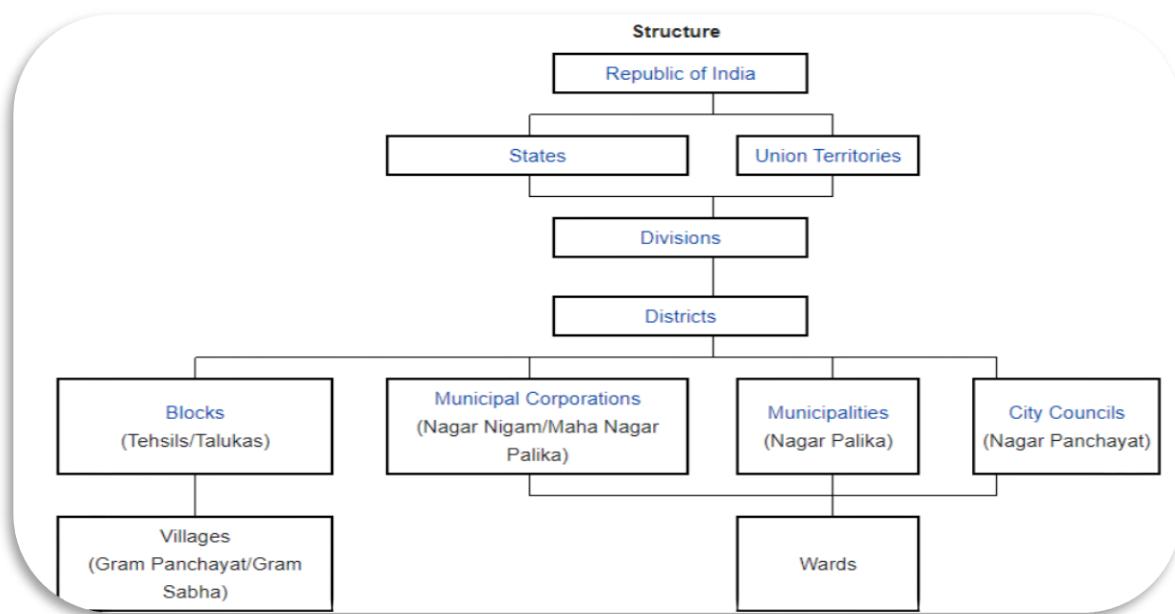
1. Online and secure payment gateway to pay revenue taxes online.
2. Provide receipt of payment in pdf format.
3. Floating govt. schemes on website to provide awareness among villagers.
4. Admin dashboard to maintain transaction data of revenue taxes and decide revenue taxes.
5. Villagers can apply for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

1.2Problem 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 “Let’s 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.

1.3 Motivation :

- 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.



1.4 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 an Integrated system to function for all gram panchayat offices in district.

1.5 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 2

PROJECT DESIGN

2.1 H/W , S/W , resources, requirements & their detail explanation

2.1.1 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.

2.1.2 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.

2.1.3 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.

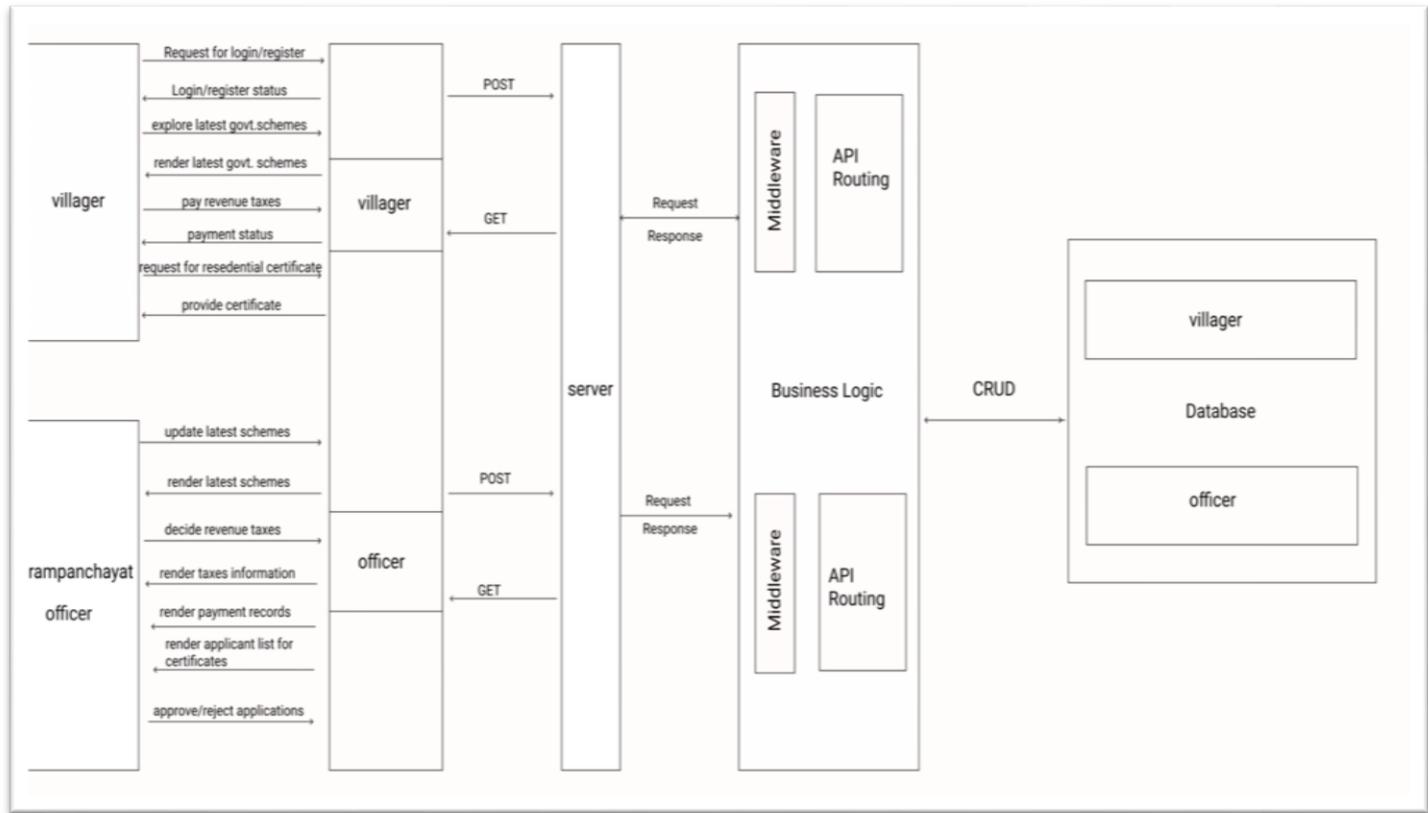
- **Maintainability:**

- Admin Dashboard.
- Feedback System.

2.1.4 Technology Stack

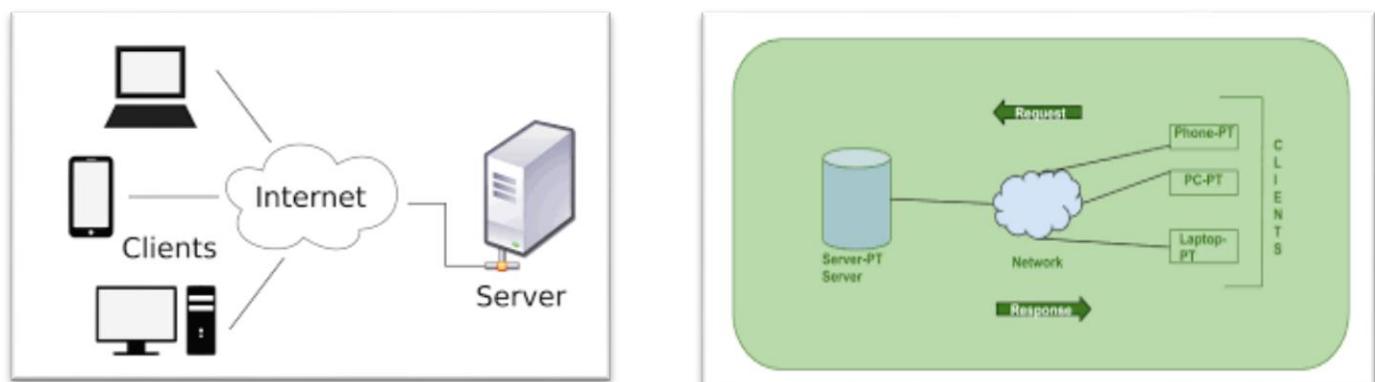
- **Front End**
 - React js: To develop web pages in component form.
 - Bootstrap, JSX, CSS: For responsive design and styling.
- **Database and Hosting**
 - Mongo DB: Data will be stored in JSON format.
- **Server side development**
 - Node js: For secure and robust server development.
 - Express js: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

Higher Level View of project



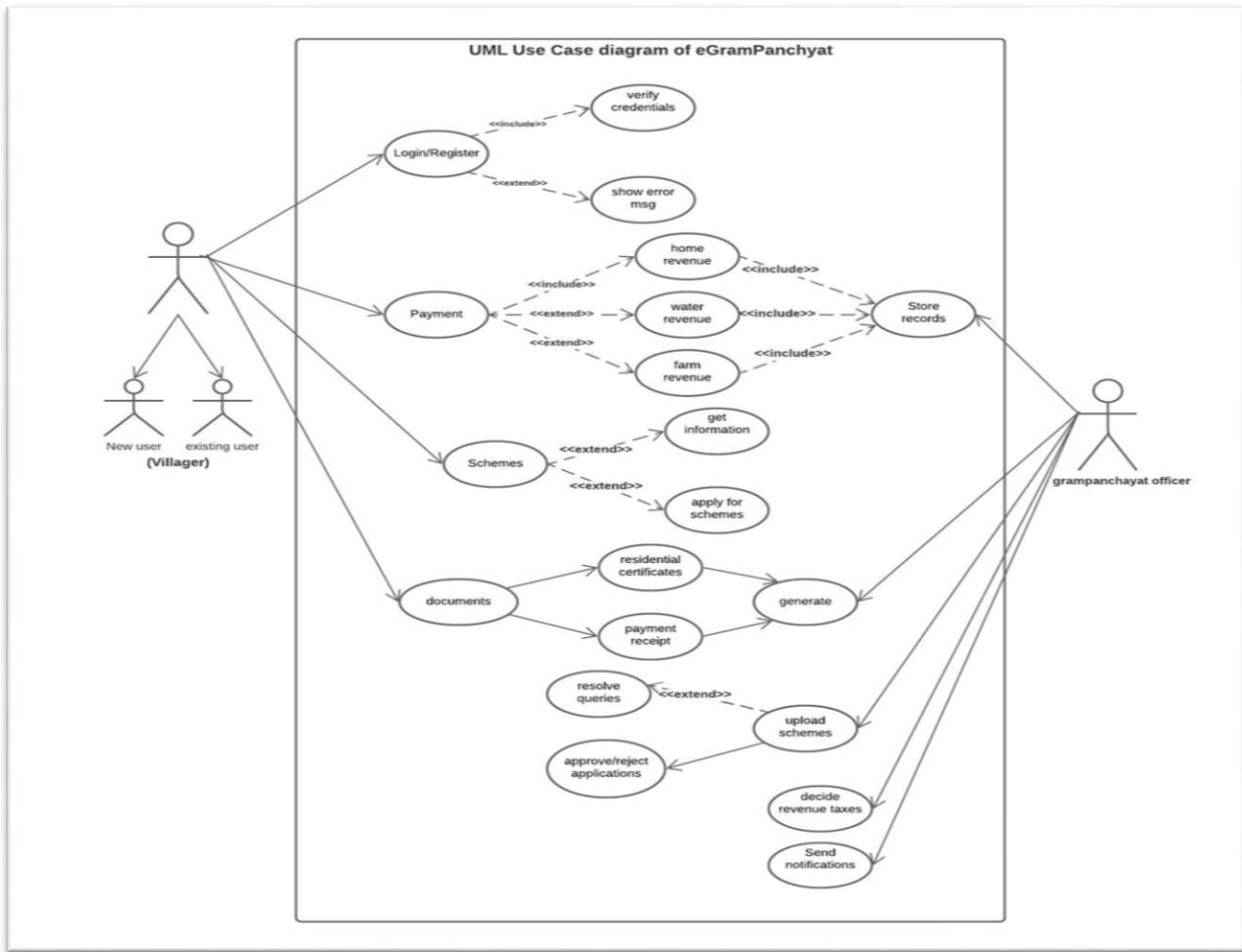
2.1.5 System Architecture:

Client server Architecture

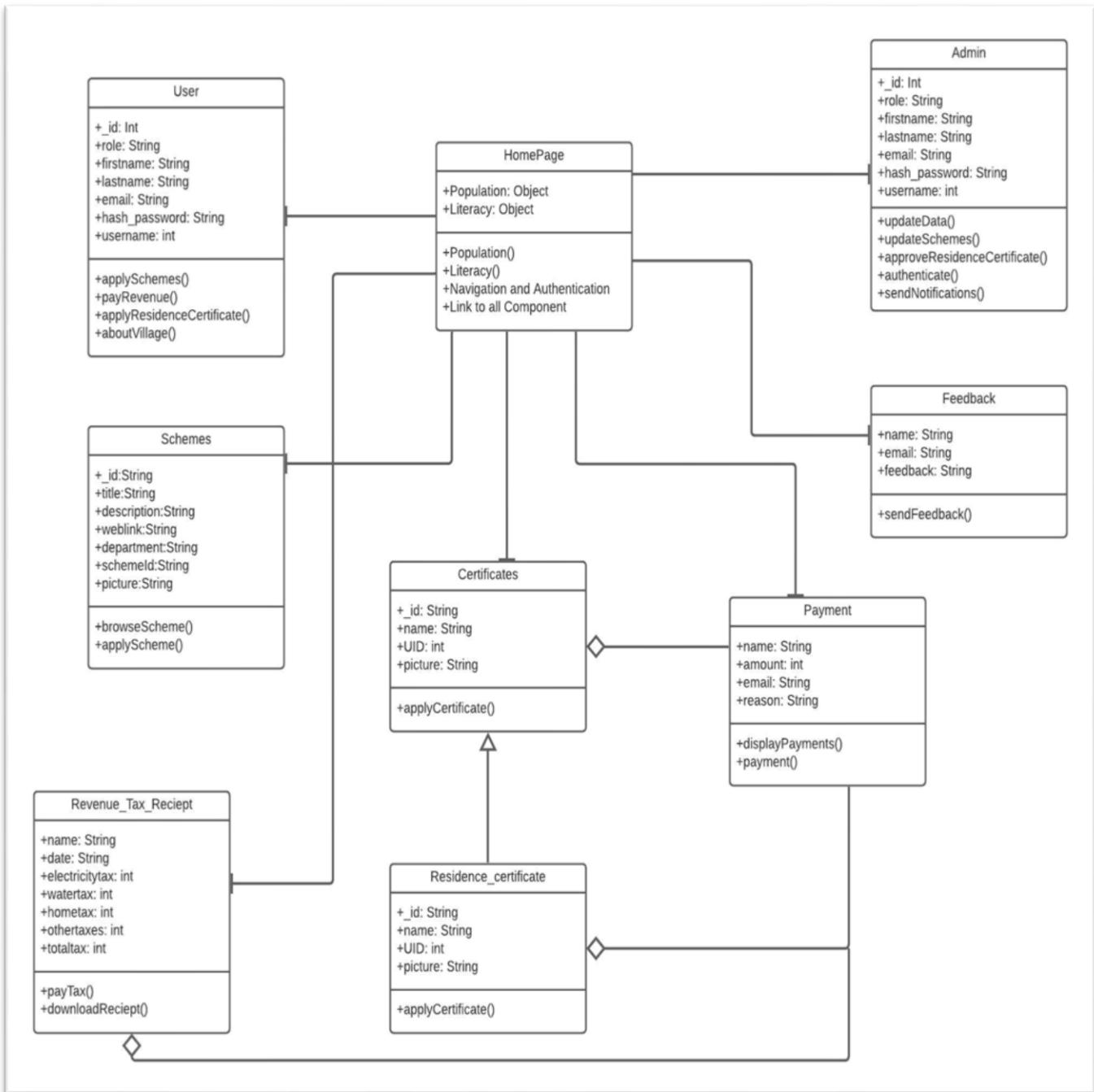


2.1.6 UML Diagrams:

UML Use Case Diagram

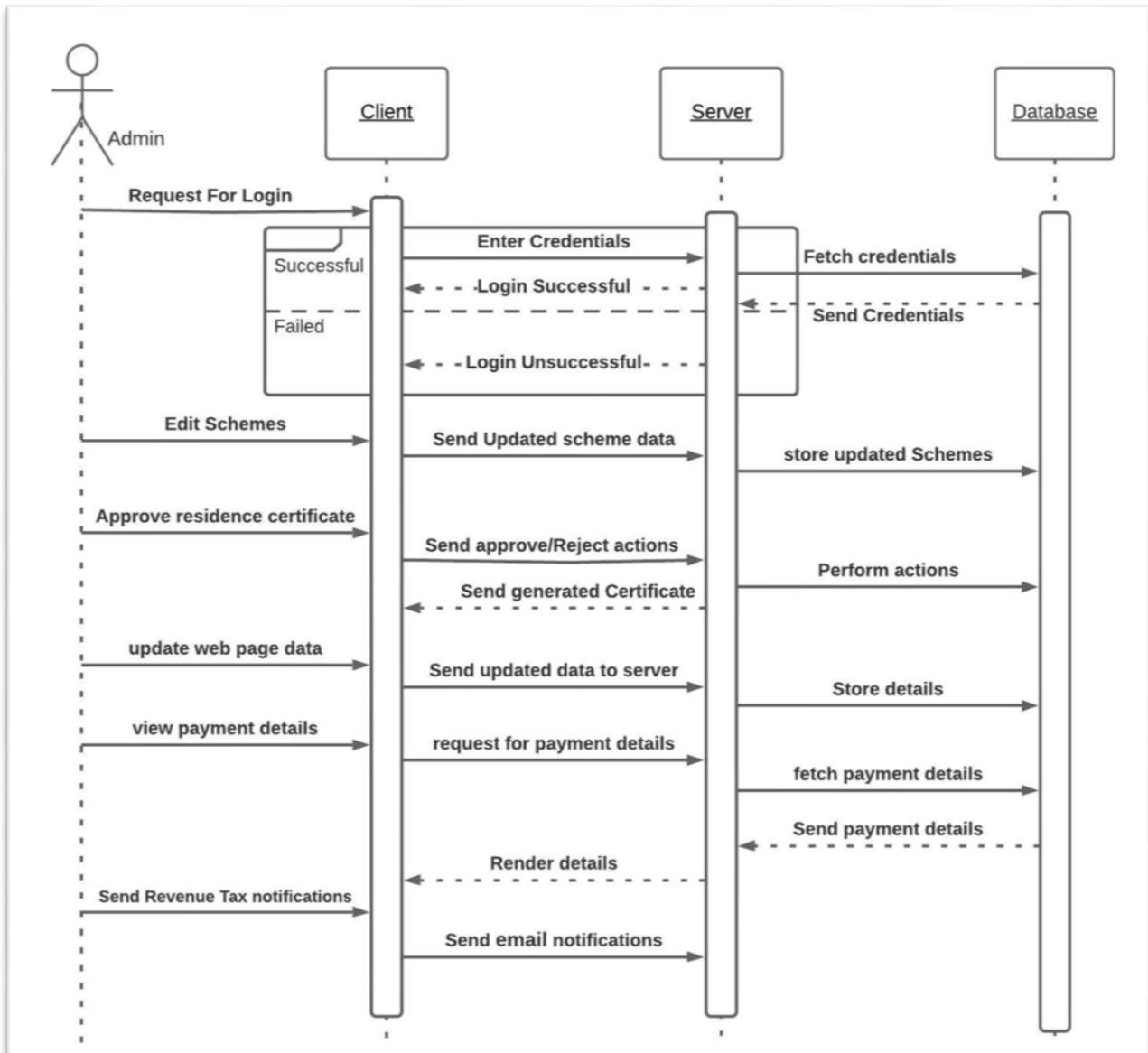


Class Diagram

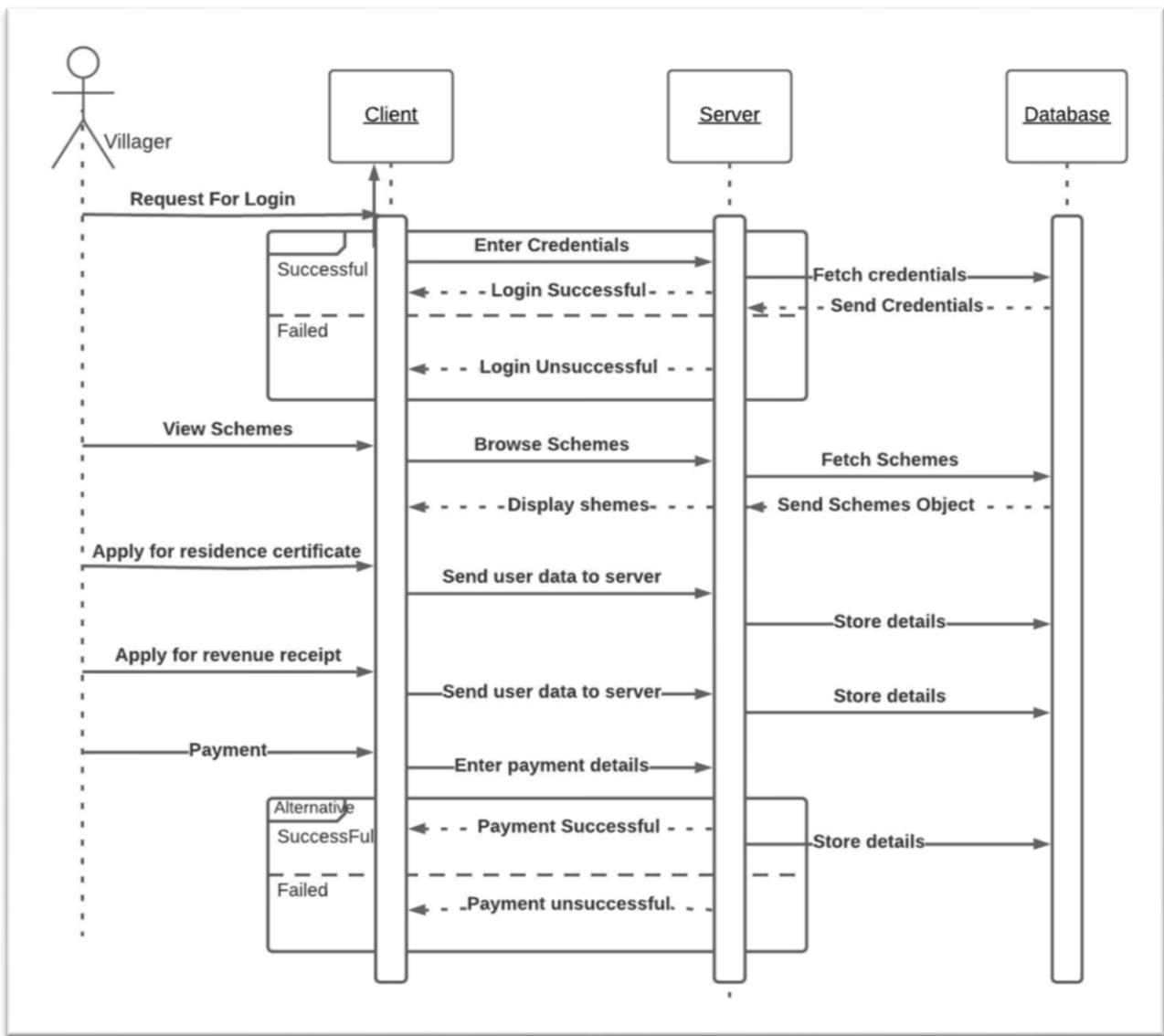


Sequence Diagram

A) Admin:

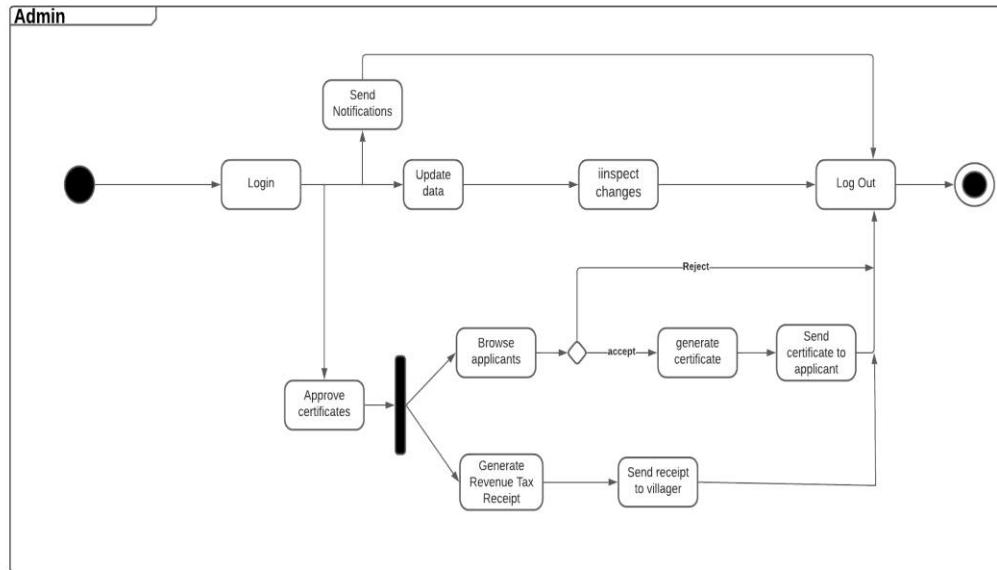


B) User (Villager):

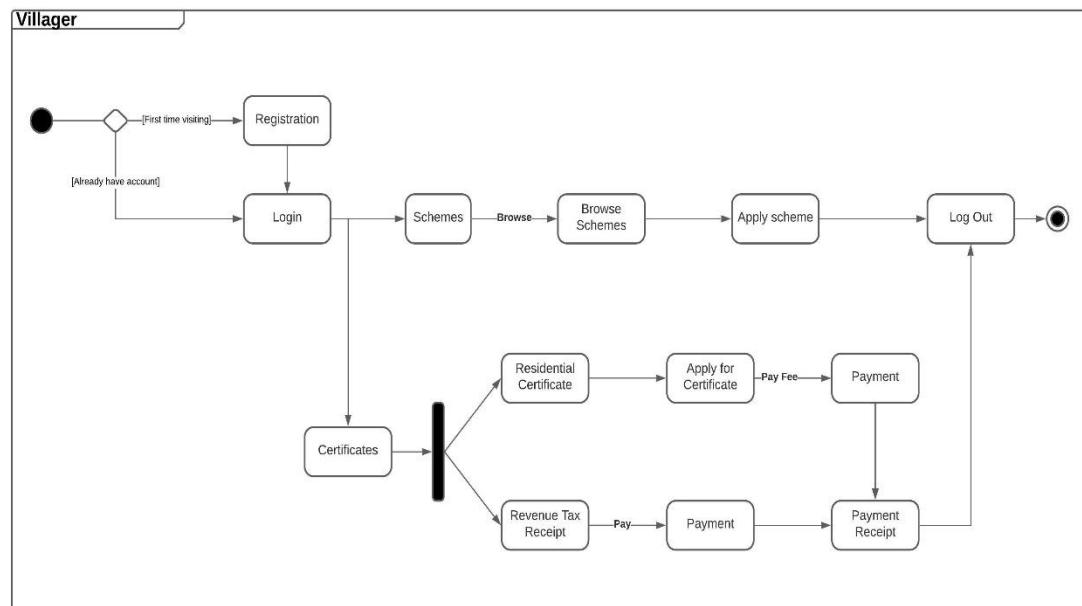


State Diagram

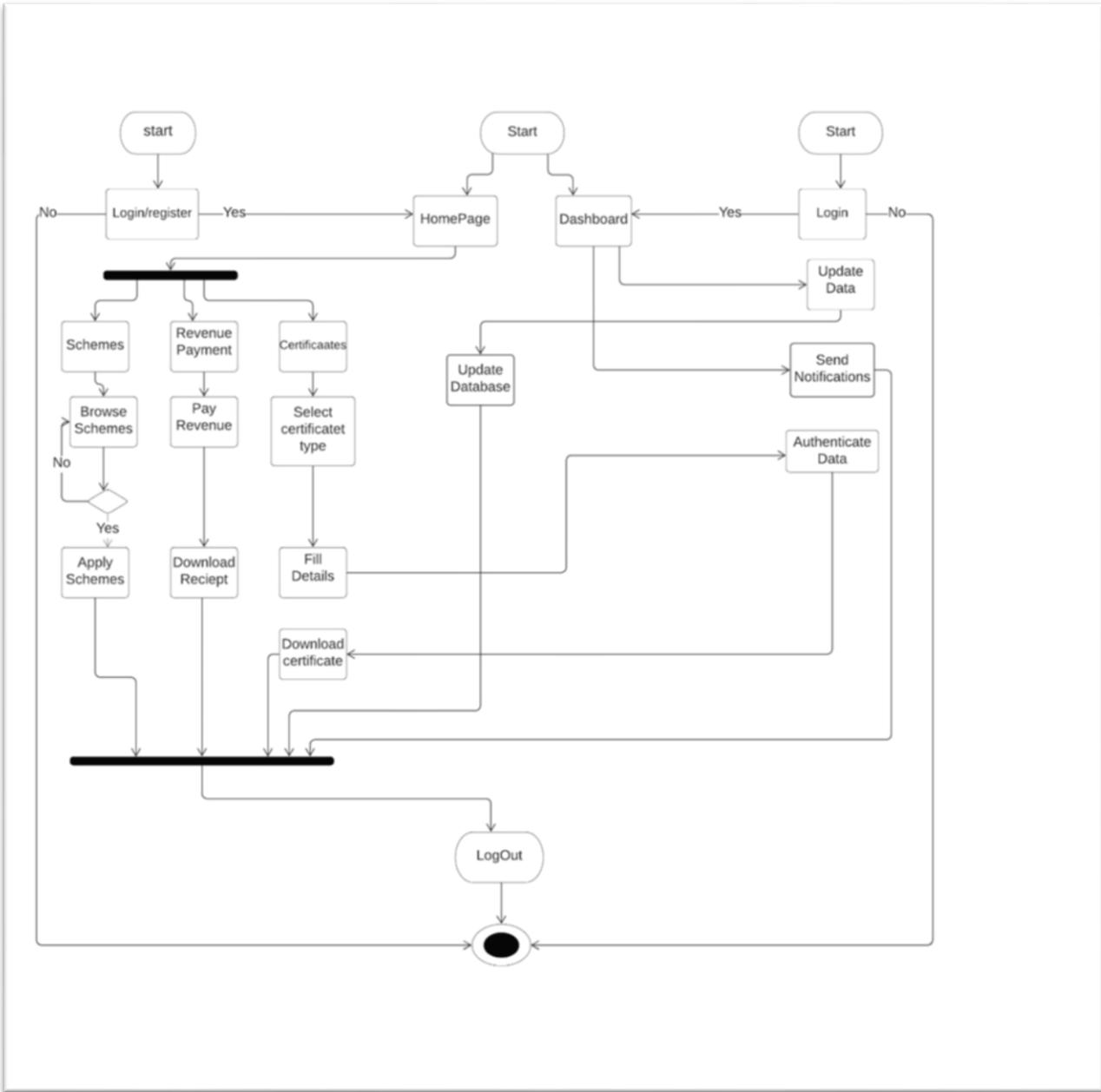
A) State diagram for Admin



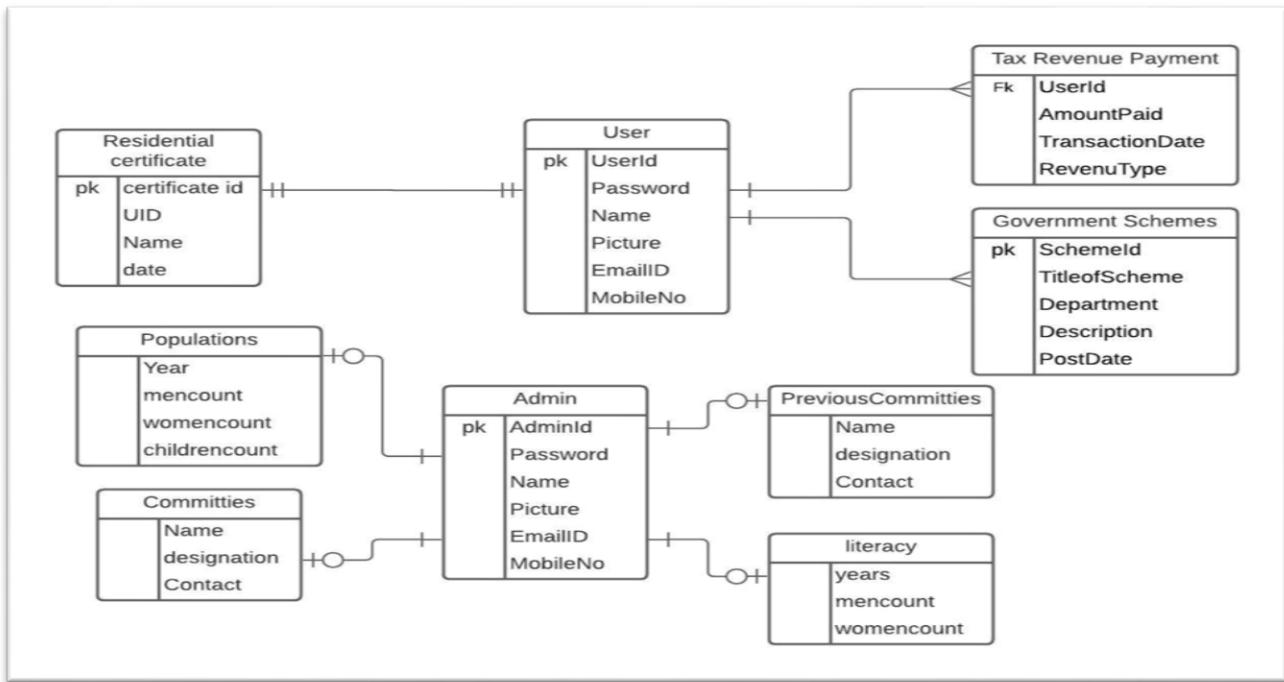
B) State diagram for User(villager):



Activity Diagram



2.2 E-R Model:



2.3 Schema of all the tables:

Current committee Schema

The screenshot shows the MongoDB Compass interface with the following details:

- Collection:** eGramPanchayat.currcommittees
- Document Count:** 1
- Size:** 461B
- Indexes:** None
- Schema Anti-Patterns:** None
- Aggregation:** None
- Search Indexes:** None
- Filter:** {"filter": "example"}
- Query Results:** 1-1 OF 1

```

{
  "_id": "ObjectID("5fc4fe7fd9e4371c044ebfc6")",
  "Name": [
    "Shakuntala Jairam Kedari.",
    "Kallan Dattatreya Naikade.",
    "Ketan Babanrao Chavan.",
    "Kamal Parajji Naikade.",
    "Anil Dattatreya Kadam."
  ],
  "designation": [
    "Village Head.",
    "Sub Village Head",
    "Committee Member.",
    "Committee Member."
  ],
  "contact": [
    "9921452713",
    "kkk",
    "9999",
    "5443",
    "6789"
  ],
  "ID": 181,
  "createdAt": "2020-11-30T14:15:27.904+00:00",
  "updatedAt": "2020-11-30T14:15:27.904+00:00",
  "v": 0
}

```

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

The screenshot shows the MongoDB Compass interface. On the left, the database sidebar lists 'eGramPanchayat' with its collections: 'currcommittees', 'payments', 'populations', 'prevcommittees', 'residences', 'revenues', 'schemes', and 'users'. The 'literacies' collection is selected and highlighted in green. The main pane displays the 'eGramPanchayat.literacies' collection. At the top, it shows 'COLLECTION SIZE: 226B' and 'TOTAL DOCUMENTS: 1'. Below that is a search bar with the placeholder 'FILTER {"filter": "example"}' and buttons for 'Find' and 'Reset'. The bottom section is titled 'QUERY RESULTS 1-1 OF 1' and shows a single document with the following data:

```
_id: ObjectId("5fc7bdbd85c6db30e008160d")
  years: Array
    0: 1971
    1: 1981
    2: 1991
    3: 2001
    4: 2011
  menCount: Array
    0: 18
    1: 25
    2: 37
    3: 52
    4: 76
  womenCount: Array
    0: 9
    1: 14
    2: 27
    3: 45
    4: 63
  ID: 101
  createdAt: 2020-12-02T16:15:57.239+00:00
  updatedAt: 2020-12-02T16:36:37.139+00:00
  __v: 0
```

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

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible under the collection 'population'. The 'populations' sub-collection is selected. The right side displays a query results page with the following details:

- FILTER**: {"filter": "example"}
- QUERY RESULTS 1-1 OF 1**
- Document Structure (Expended):**
 - _id: ObjectId("5fc7b75bd75e831f4c8eedff")
 - years: Array
 - 0: 1971
 - 1: 1981
 - 2: 1991
 - 3: 2001
 - 4: 2011
 - menCount: Array
 - 0: 900
 - 1: 1012
 - 2: 1288
 - 3: 1354
 - 4: null
 - womenCount: Array
 - 0: 700
 - 1: 900
 - 2: 1000
 - 3: 1200
 - 4: 1300
 - childrenCount: Array
 - 0: 550
 - 1: 600
 - 2: 700
 - 3: 1200
 - 4: 900
 - createdAt: 2020-12-02T15:48:43.803+00:00
 - updatedAt: 2020-12-02T16:32:51.885+00:00
 - __v: 0

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

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible under the 'eGramPanchayat' namespace, including collections like 'currcommittees', 'literacies', 'payments', 'populations', and 'prevcommittees'. The 'prevcommittees' collection is selected. At the top right, there are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. Below these are buttons for 'INSERT DOCUMENT', 'Find', and 'Reset'. A search bar with the filter value '{"filter": "example"}' is present. The results section shows one document with the following data:

```
_id: ObjectId("5fc50048d9e4371c044ebfc7")
  Name: Array
    0: "Dattatraya Narayan Naikade"
    1: "Alka Dattatraya Naikade"
    2: "Kausalya Raamda Naikade"
    3: "Vikas Sambhaji Mandlik"
    4: "Asha Sunil Shinde"
  workingPeriod: Array
    0: "1990-1995"
    1: "1995-2000"
    2: "2000-2005"
    3: "2005-2010"
    4: "2010-2015"
  Caste: Array
    0: "Maratha"
    1: "Maratha"
    2: "Maratha"
    3: "OBC"
    4: "Buddhism"
  ID: 102
  createdAt: 2020-11-30T14:23:04.220+00:00
  updatedAt: 2020-11-30T15:02:02.751+00:00
  __v: 0
```

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

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible with 'eGramPanchayat' expanded, showing collections like 'residences'. In the main pane, the 'residences' collection is selected, displaying two documents. The first document's details are shown in a modal:

```
_id: ObjectId("5fc750c64c952e06dcdd3782")
name: "ମୁଣ୍ଡ ପାଞ୍ଚାୟତ"
UID: 456478971231
date: 2020-12-02T08:31:01.980+00:00
createdAt: 2020-12-02T08:31:02.090+00:00
updatedAt: 2020-12-02T08:31:02.090+00:00
__v: 0
```

The second document has a similar structure. A green circular icon with a white square and a checkmark is located in the bottom right corner of the main pane.

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

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the database structure under the 'eGramPanchayat' namespace, including 'currcommittees', 'literacies', 'payments', 'populations', 'prevcommittees', 'residences', 'revenues' (which is selected), 'schemes', and 'users'. The main panel is titled 'eGramPanchayat.revenues' and shows a collection size of 295B, total documents of 2, and indexes totaling 36KB. It includes tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. A search bar at the top has the filter '["filter": "example"]'. Below it, a green 'Find' button and a 'Reset' button are visible. The results section is titled 'QUERY RESULTS 1-2 OF 2' and lists two documents:

```
_id: ObjectId("5fc750d64c952e06dcdd3783")
name: "मौलि लोग्न"
UID: 456478971231
date: 2020-12-02T08:31:18.862+00:00
createdAt: 2020-12-02T08:31:18.979+00:00
updatedAt: 2020-12-02T08:31:18.979+00:00
__v: 0

_id: ObjectId("5fc80b1748ca12f33585da8a5")
name: "राजित अंदर नार्हुकडे"
UID: 541223563259
date: 2020-12-03T09:35:47.864+00:00
createdAt: 2020-12-03T09:35:48.048+00:00
updatedAt: 2020-12-03T09:35:48.048+00:00
__v: 0
```

A green circular icon with a white document symbol is located in the bottom right corner of the results panel.

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

COLLECTION SIZE: 2.49KB TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 72KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER ("filter": "example")

Find Reset

QUERY RESULTS 1-3 OF 3

_id	ObjectID("5f7d66c84d9ea882441d8320")
title	"Pradhan Mantri Jan Dhan Yojana (PMJDY)"
description	"Pradhan Mantri Jan-Dhan Yojana (PMJDY) is National Mission for Financial Inclusion. It aims to provide a bank account to every Indian citizen."
weblink	https://www.pmdy.gov.in/scheme
department	"central govt."
schemeId	"0.05180852707170924"
picture	http://localhost:5000/public/ChDahG-m2-jan dhan yojana.jpg
createdAt	2020-10-07T06:57:12.486+00:00
updatedAt	2020-10-07T06:57:12.486+00:00
_v	8

_id	ObjectID("5f7d6884d9aa882441d8321")
title	"Pradhan Mantri Jeevan Jyoti Bima Yojana (PMJJBY)"
description	"The PMJJBY is available to people in the age group of 18 to 50 years h..."
weblink	https://financialservices.gov.in/insurance-divisions/government-sponsore...
department	"central govt."
schemeId	"0.31307901521110715"
picture	http://localhost:5000/public/UMsvg@oef-jeevan jyoti.png
createdAt	2020-10-07T07:02:28.297+00:00
updatedAt	2020-10-07T07:02:28.297+00:00
_v	8

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

The screenshot shows the MongoDB Compass interface. On the left, the 'NAMESPACES' sidebar lists the database 'eGramPanchayat' and its collections: currcommittees, literacies, payments, populations, prevcommittees, residences, revenues, schemes, and users. The 'users' collection is selected. At the top, the 'CURRENT COLLECTIONS' header shows 'COLLECTION SIZE: 1.05KB TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 108KB'. Below it are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. A search bar contains the filter '{"filter": "example"}'. To the right are buttons for 'INSERT DOCUMENT', 'Find' (highlighted in green), and 'Reset'. The results section displays two documents under 'QUERY RESULTS 1-4 OF 4':

```
_id: ObjectId("5f618f180baafc92f64000c7c")
role: "user"
firstname: "rohit"
lastname: "naikade"
email: "naikade@gmail.com"
hash_password: "$2b$10$OsdyQv7ydaq07e1fw00L0V0j14pQ3HuJ5E4GQJHN2jahRgVc7lse"
username: "8.16408781056317247"
createdAt: 2020-09-16T04:05:44.800+00:00
updatedAt: 2020-09-16T04:05:44.800+00:00
__v: 0

_id: ObjectId("5f6d66cebe87251640e4aea2")
role: "user"
firstname: "mehul"
lastname: "lokhande"
email: "mehul@gmail.com"
hash_password: "$2b$10$wDZNRAtxF9myu/rPPMGezVARcj1Lznxk3qlL6vERCFJtNe50aPe"
username: "8.6368192432574894"
createdAt: 2020-09-25T03:41:02.652+00:00
updatedAt: 2020-09-25T03:41:02.652+00:00
__v: 0
```

A green circular icon with a white square is visible on the right side of the results area.

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  
});
```

2.4 Hours 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.

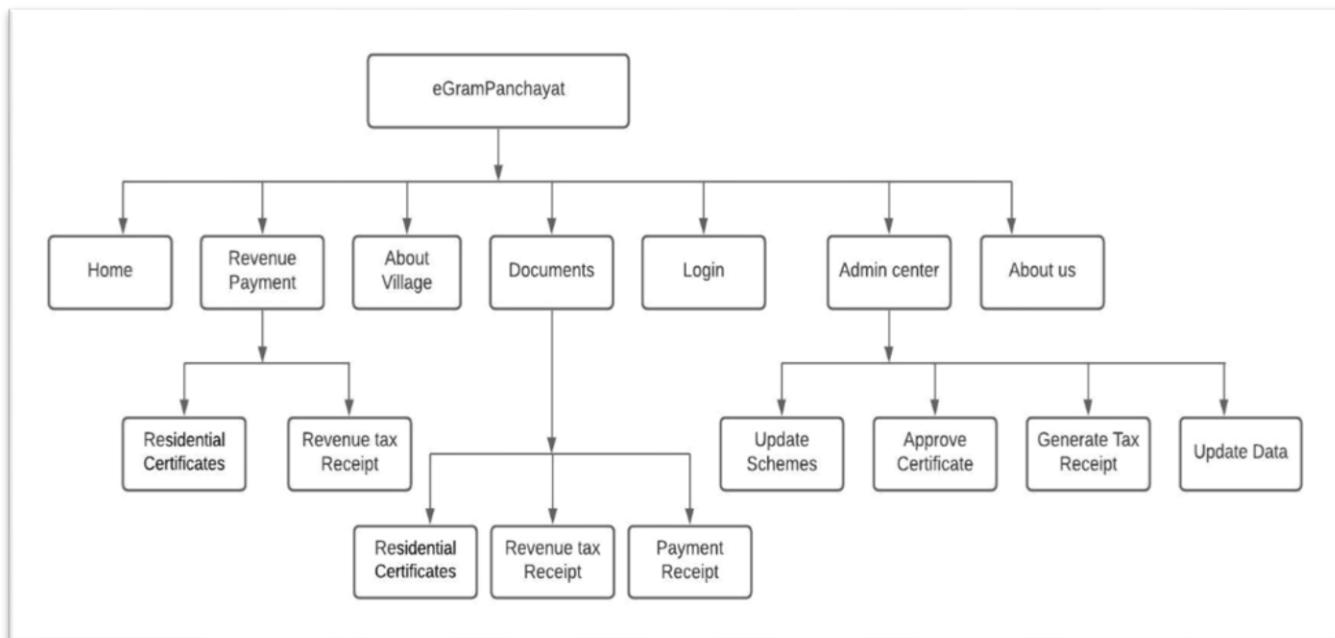
Effort and Development Time: It took nearly 3 months for planning, designing, implementation, debugging, Testing and Deployment of our project.

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 3

MODULE DESCRIPTION

3.1 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 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:**

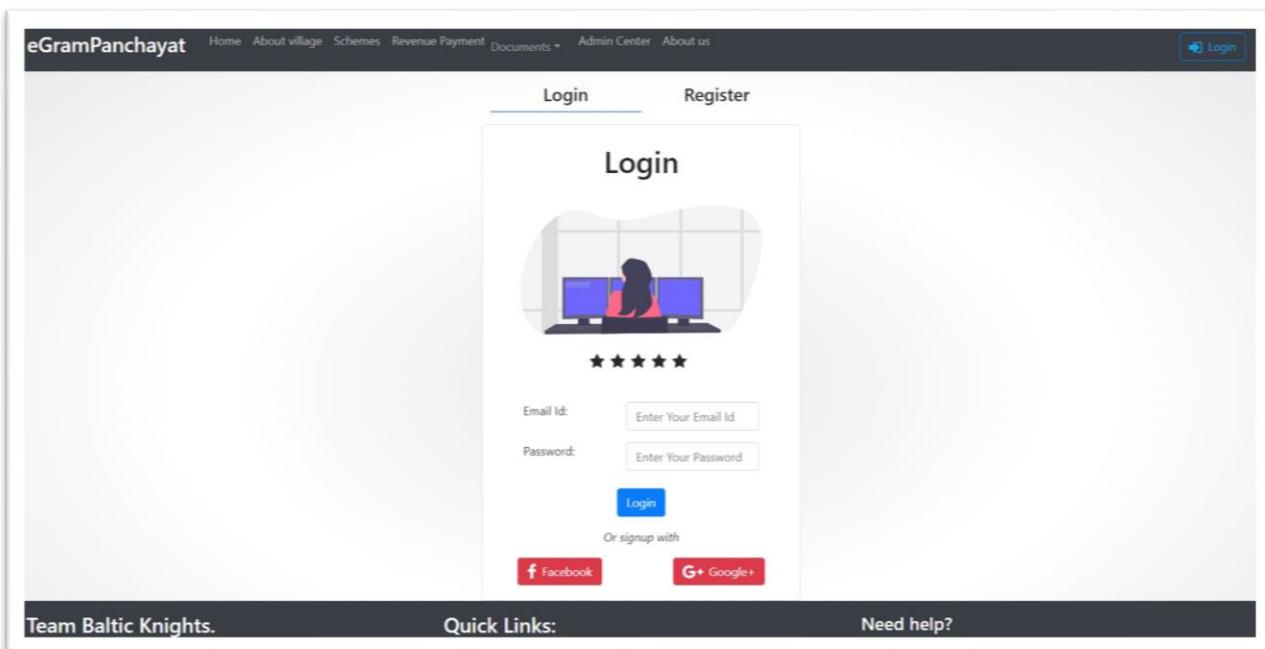
From this tab, villager can contact with development team on social media accounts. We have provided contact details of team members.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Source code and Screen shots including GUI

4.1.1 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}$/.test(val);
const Login = () => {
  const [formData, setFormData] = useState({
    email: '',
    password1: '',
    textChange: 'Sign In'
  });
  const { email, password1, textChange } = formData;
  const handleChange = text => e => {
    setFormData({ ...formData, [text]: e.target.value });
  };
  const handleSubmit = (values) => {
    if (email && password1) {
      setFormData({ ...formData, textChange: 'Submitting' });
      axiosInstance.post("user/login", {
        email, password:password1
      })
      .then(res => {
        authenticate(res, () => {
          setFormData({
            ...formData,
            email: '',
            password1: '',
            textChange: 'Submitted'
          });
          console.log(isAuth())
          isAuth() && isAuth().role === 'admin'
            ? history.push('/admin')
            : history.push('/');
          store.addNotification({
            title: `${res.data.user.name}, welcome back!`,
            message: 'Now you have privileges to explore!',
            type: "success",
            container: 'top-right',
            animationIn: ["animated", "fadeIn"],
            animationOut: ["fadeOut", "animated"]
          });
        })
      .catch(error => {
        console.error(error)
        store.addNotification({
          title: 'An error occurred while logging in',
          message: 'Please try again later',
          type: "error",
          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">

```

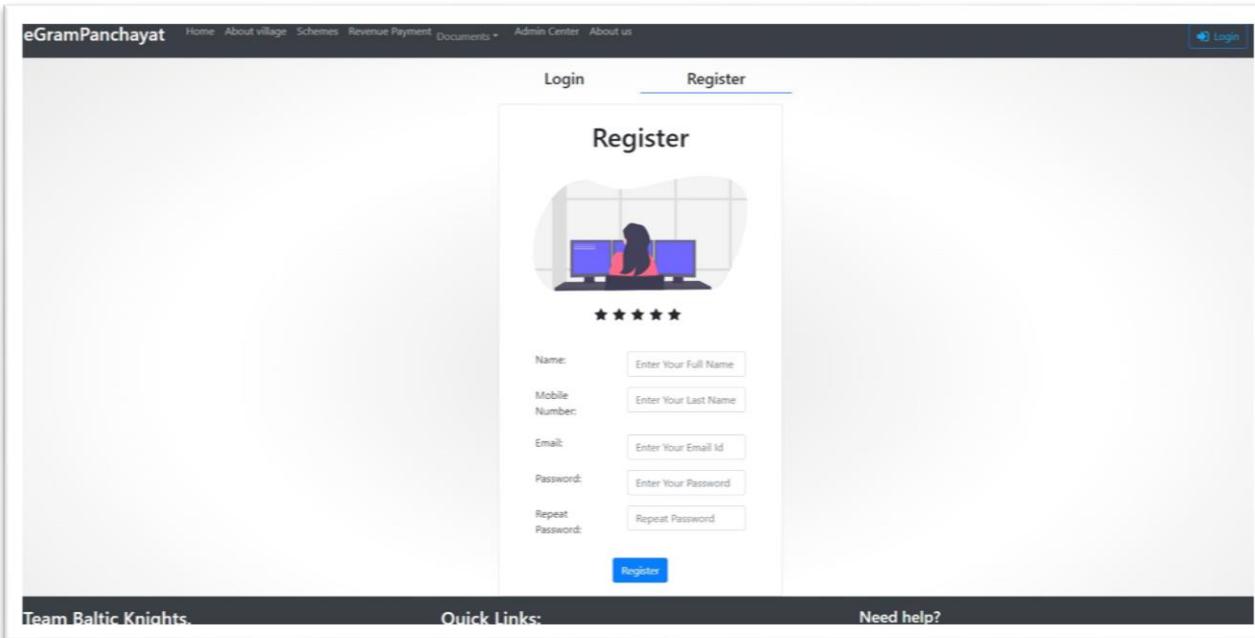
```

<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>

```

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

4.1.2 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}$/.test(val);
const validText = (val) => /^[A-Za-z]+$/i.test(val);
const validMobile=(val)=>/^((\+){1}{91}){1}[1-9]{1}[0-9]{9}$/.test(val);
const validName=(val)=>/^([a-zA-Z0-9]+[a-zA-Z0-9]+[a-zA-Z0-9]{1,})[a-zA-Z0-9]+[a-zA-Z0-9]{3,}[a-zA-Z0-9]{1,}$/.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)} />
    </Col>
  </Row>
</Col></Row>

```

```

    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;

```

4.1.3 Home Module

eGramPanchayat Home About village Schemes Revenue Payment Documents Admin Center About us Login

ग्रामपंचायत तांदूळवाडी आपले सहर्ष स्वागत करत आहे.

Population from last 5 Surveys.

Year	Men	Women	Children
1971	900	650	300
1981	1000	850	450
1991	1250	1000	650
2001	1350	1200	1200
2011	1300	1300	900

Literacy from last 5 Surveys.

Year	Men	Women
1971	20	10
1981	25	15
1991	35	25
2001	45	35
2011	75	65

Locate Us.

Team Baltic Knights.

I dream of a Digital India where knowledge is strength-and empowers the people.
-Narendra Modi.

Quick Links:

- [Home.](#)
- [About village.](#)
- [Schemes.](#)
- [Revenue payment.](#)
- [About us.](#)

Need help?

Email: balticknightsofficial@gmail.com.

Connect with us:

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;

```

4.1.4 About Village Module

eGramPanchayat Home About village Schemes Revenue Payment Documents Admin Center About us [Login](#)

Vindhyaachal Hydro Power PVT. LTD.

Chaskaman Hydro-Power Plant is a 3-MW power station located at Chaskaman Dam on Bhima river near Rajgurunagar in Maharashtra, India. This is the second greenfield Hydel Power Project developed by the Company.

Last 5 village heads of village:

Sr. No:	Name.	Working Period.	Caste.
1	Dattatraya Narayan Naikade	1990-1995	Maratha
2	Alka Dattatraya Naikade	1995-2000	Maratha
3	Kausalya Ramdas Naikade	2000-2005	Maratha
4	Vikas Sambhaji Mandlik	2005-2010	OBC
5	Asha Sunil Shinde	2010-2015	Buddhism

Members of current committee:

Sr. No:	Full name:	Designation:	Contact No:
1	Shakuntala Jairam Kedari.	Village Head.	9921452713
2	Kailas Dattatraya Naikade.	Sub Village Head	9850916901
3	Ketan Babanrao Chavan.	Committee Member.	9623362555
4	Kamal Parajji Naikade.	Committee Member.	9623835281
5	Anil Dattatraya Kadam.	Committee Member.	9270086773

Team Baltic Knights.

I dream of a Digital India where knowledge is strength and empowers the people.

-Narendra Modi.

Quick Links:

- [Home.](#)
- [About village.](#)
- [Schemes.](#)
- [Revenue payment.](#)
- [About us.](#)

Need help?

Email: balticknightsofficial@gmail.com.

Connect with us:



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>
        </tr>
      );
    });
  }
}
```

```

        <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 important dams of Maharashtra and is built on the Bhima River at Rajgurunagar in Pune district. The capacity of Chas Kaman Dam to irrigate about 32824 ha of land of the villages nearby in Pune district.</p>

```
</Carousel.Caption>
</Carousel.Item>
<Carousel.Item interval={1000}>
  <img
    className="d-block w-100"
    src={powerhouse}
    alt="powerhouse"
  />
  <Carousel.Caption>
    <h3>Vindhyaachal Hydro Power PVT. LTD</h3>
    <p>Chaskaman Hydro-
```

Power Plant is a 3 MW power station located at Chaskaman Dam on Bhima river near Rajgurunagar in Maharashtra, India. This is the second greenfield Hydel Power Project developed by the Company.</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 generations.their houses are reflections of their simple 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 road.</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;

```

4.1.5 Schemes Modules

The screenshot shows the eGramPanchayat website interface. At the top, there is a navigation bar with links: Home, About village, Schemes, Revenue Payment, Documents, Admin Center, About us, and a Login button. The main content area is titled "Government Schemes." It features three cards for different government schemes:

- Pradhan Mantri Jan Dhan Yojana (PMJDY)**: Description: Pradhan Mantri Jan-Dhan Yojana (PMJDY) is National Mission for Financial Inclusion to ensure access to financial services, namely, a basic savings & deposit accounts, remittance, credit, insurance, pension in an affordable manner. Under the scheme, a basic savings bank deposit (BSBD) account can be opened in any bank branch or Business Correspondent (Bank Mitra) outlet, by persons not having any other account. Department: central govt. [Apply Now](#)
- Pradhan Mantri Jeevan Jyoti Bima Yojana (PMJJBY)**
- Atal Pension Yojana (APY)**

The footer contains the following information:

- Team Baltic Knights.**
- Quick Links:**
 - [Home](#)
 - [About village](#)
- Need help?**
 - [Email](#) balticknightsofficial@gmail.com

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="myAccordion" 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>

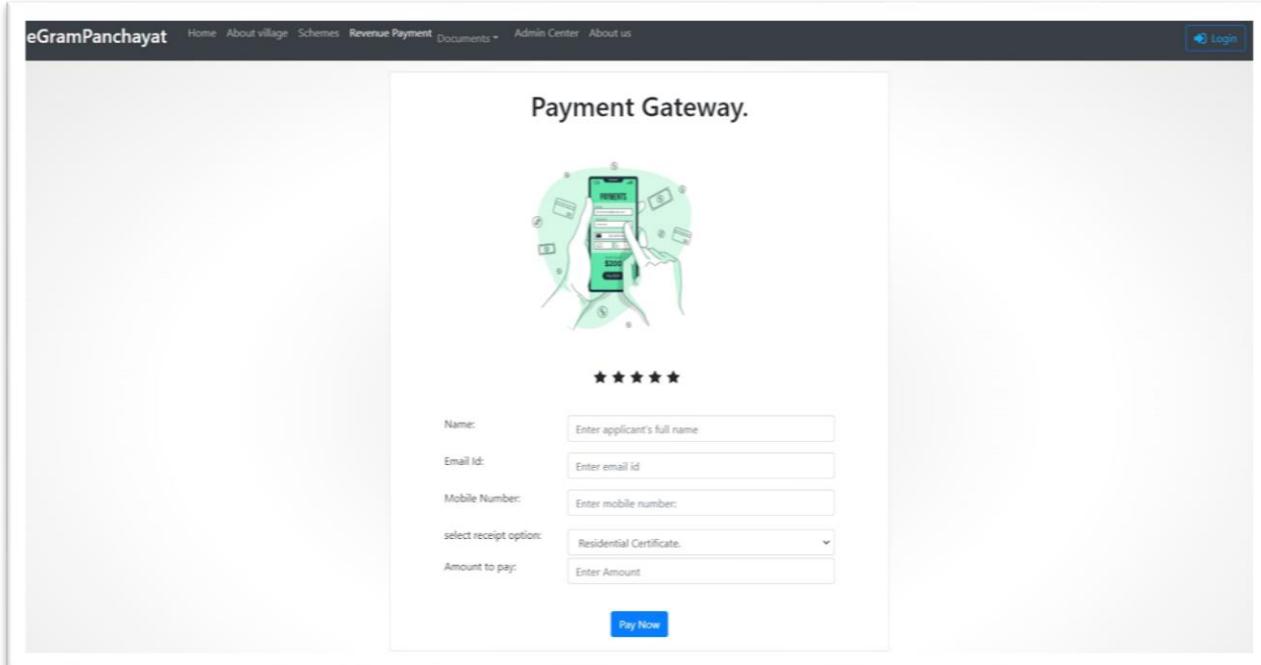
```

```

<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;

```

4.1.6 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 compatible
    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"
                    >
                  </Col>
                </Row>
              </div>
            </LocalForm>
          </Card.Body>
        </Card>
      </FadeTransform>
    </Col>
  </Row>
</Container>

```

```

    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.</option>
    <option value="Revenue tax receipt.">Revenue tax receipt.</option>
</Control.select>
</Col></Row>

```

```

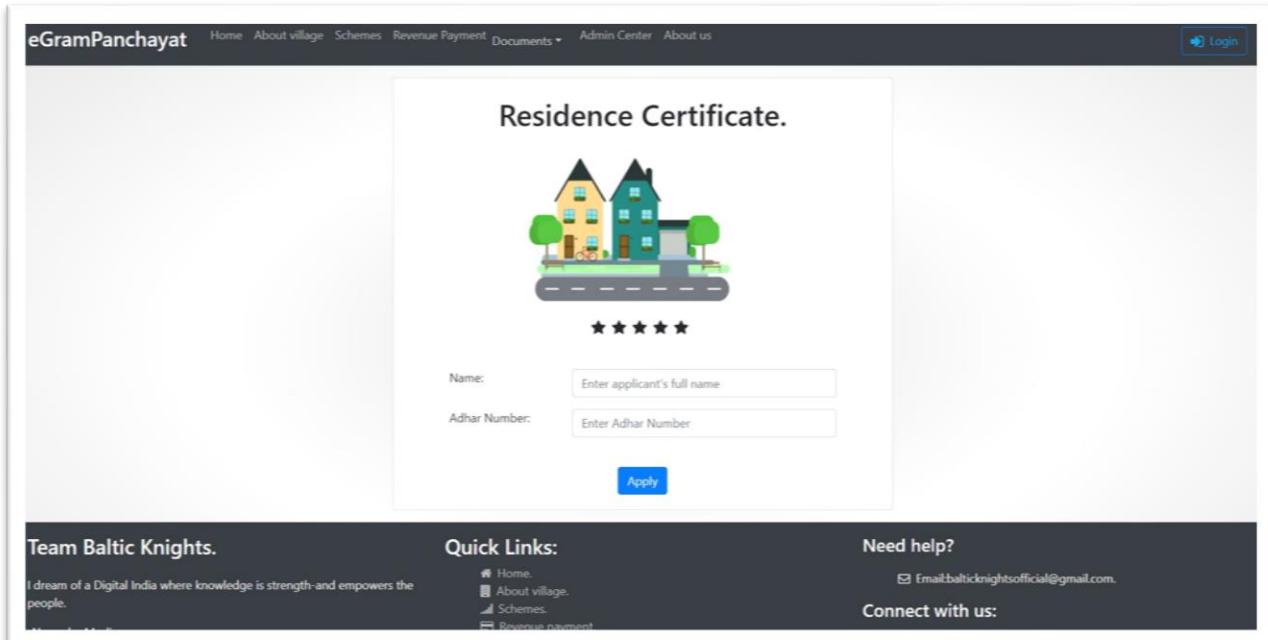
<Row><Col>
  <div className="form-group">
    <Row><Col className="col-md-3 offset-md-1"><Form.Label>Amount to pay:</Form.Label></Col>
      <Col className="col-md-7">
        <Control.text
          model=".amount"
          className="form-control"
          autoComplete="off"
          placeholder="Enter 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;

```

4.1.7 Documents Module:

4.1.7.1 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>
      </LocalForm>
    </Card.Body>
  </Card>

```

```

        }
      />
    </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;

```

4.1.7.2 Revenue tax receipt

The screenshot shows the eGramPanchayat website interface. At the top, there is a navigation bar with links for Home, About village, Schemes, Revenue Payment, Documents, Admin Center, and About us. A 'Login' button is also present. The main content area has a title 'Revenue tax receipt.' and a decorative illustration of three stylized figures with balloons. Below the illustration is a five-star rating icon. There are two input fields: 'Name:' and 'Adhar Number:', both with placeholder text. A blue 'Apply' button is located below these fields. At the bottom of the page, there is a footer section with the text 'Team Baltic Knights.', a 'Quick Links' menu with items like Home, About village, Schemes, and Revenue payment, and a 'Need help?' section with an email address: Emailbalticknightsofficial@gmail.com. The footer also includes a 'Connect with us:' section.

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,
        transition: "out-fade"
      }
    })
  }
}
```

```

        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 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"
        autoComplete="off"
        className="form-control"
        placeholder="Adhar Number"
        name="Adhar Number"
        value={UID}|

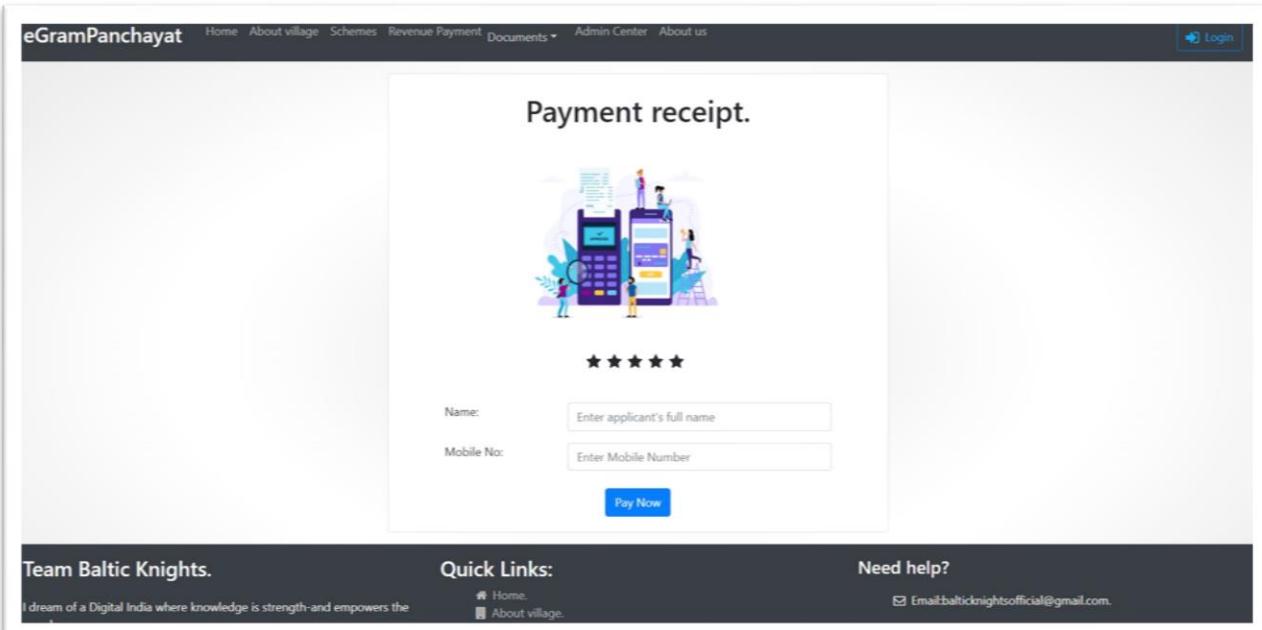
```

```

        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">Apply</Button>
</div>
</LocalForm>
</Card.Body>
</Card></FadeTransform></Col>
</Row>
</Container>
)
}
export default Residence;

```

4.1.7.3 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)
                                            )}>
                                    </Col>
                                </Row>
                            </div>
                            <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;

```

4.1.8 Admin center Module:

4.1.8.1 Home section

The screenshot shows the 'Home' section of the eGramPanchayat Admin Center. On the left, there is a sidebar with navigation links: Home, About Village, Schemes, Payment, Residence Certificate, and Revenue Tax. The 'Residence Certificate' link is currently selected. The main content area has a title 'Home section.' and two tables.

Population of village in last Five surveys.

1.	Year	Men Count	Women Count	Children Count
2.	Year	Men Count	Women Count	Children Count
3.	Year	Men Count	Women Count	Children Count
4.	Year	Men Count	Women Count	Children Count
5.	Year	Men Count	Women Count	Children Count

Literacy Rate of village in last Five surveys.

1.	Enter Year	Men Count	Women Count
2.	Enter Year	Men Count	Women Count
3.	Enter Year	Men Count	Women Count
4.	Enter Year	Men Count	Women Count
5.	Enter Year	Men Count	Women Count

Both tables have a blue 'Update Records' button at the bottom right.

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, settthreeChildren] = 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 => settreeMen(e.target.value)}
    />
</Col>
<Col className="col-md-3">
    <Form.Control
        type="text"
        autoComplete="off"
        placeholder="Women Count"
        value={threeWomen}
        onChange={e => settreeWomen(e.target.value)}
    />
</Col>
<Col className="col-md-3">
    <Form.Control
        type="text"
        autoComplete="off"
        placeholder="Children Count"
        value={threeChildren}
        onChange={e => settreeChildren(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}
    />
  </Col>
</Row>

```

```

        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;

```

4.1.8.2 About village section

The screenshot shows the 'About Village Section' page of the eGramPanchayat application. The left sidebar includes links for Home, About Village, Schemes, Payment, Residence Certificate, and Revenue Tax. The main content area has a title 'About Village Section.' and two data entry forms.

Last 5 Village Heads:

1.	Enter Name	Enter Period	Enter Caste
2.	Enter Name	Enter Period	Enter Caste
3.	Enter Name	Enter Period	Enter Caste
4.	Enter Name	Enter Period	Enter Caste
5.	Enter Name	Enter Period	Enter Caste

Update Records

Current Gram panchayat committee:

1.	Enter Name	Enter Designation	Enter Contact No
2.	Enter Name	Enter Designation	Enter Contact No
3.	Enter Name	Enter Designation	Enter Contact No
4.	Enter Name	Enter Designation	Enter Contact No
5.	Enter Name	Enter Designation	Enter Contact No

Update Records

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, CurrFifthPeriod],
    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,PrevFifthPeriod],
        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>
      </Row>
    </Form>
  </Card.Body>
</Card>

```

```

<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>
<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>
</Row>

```

```

        />
      </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>
      </Row>
    
```

```

<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;

```

4.1.8.4 Residence certificate

The screenshot shows the eGramPanchayat website interface. On the left is a sidebar with navigation links: Home, About Village, Schemes, Payment, Residence Certificate (which is highlighted), and Revenue Tax. The main content area has a header "Applicants for Residence Certificate." Below it are two card-like boxes, each representing an application.

Top Application:

- Name:** મેહુલ લોહંડે
- Applicant Name:** મેહુલ લોહંડે
- Adhar Number:** 456478971231
- Application Date:** 6/11/2020
- Status:** ✓ ✗

Bottom Application:

- Name:** રોહિત આનંદ નાઈકડે
- Applicant Name:** રોહિત આનંદ નાઈકડે
- Adhar Number:** 541223563259
- Application Date:** 6/11/2020
- Status:** ✓ ✗

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="myAccordion" 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={()=>Approve(data.name, data.UID)} />
                    <FcIcons.FcDisapprove className="ml-3 icons" size={40} onClick={()=>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

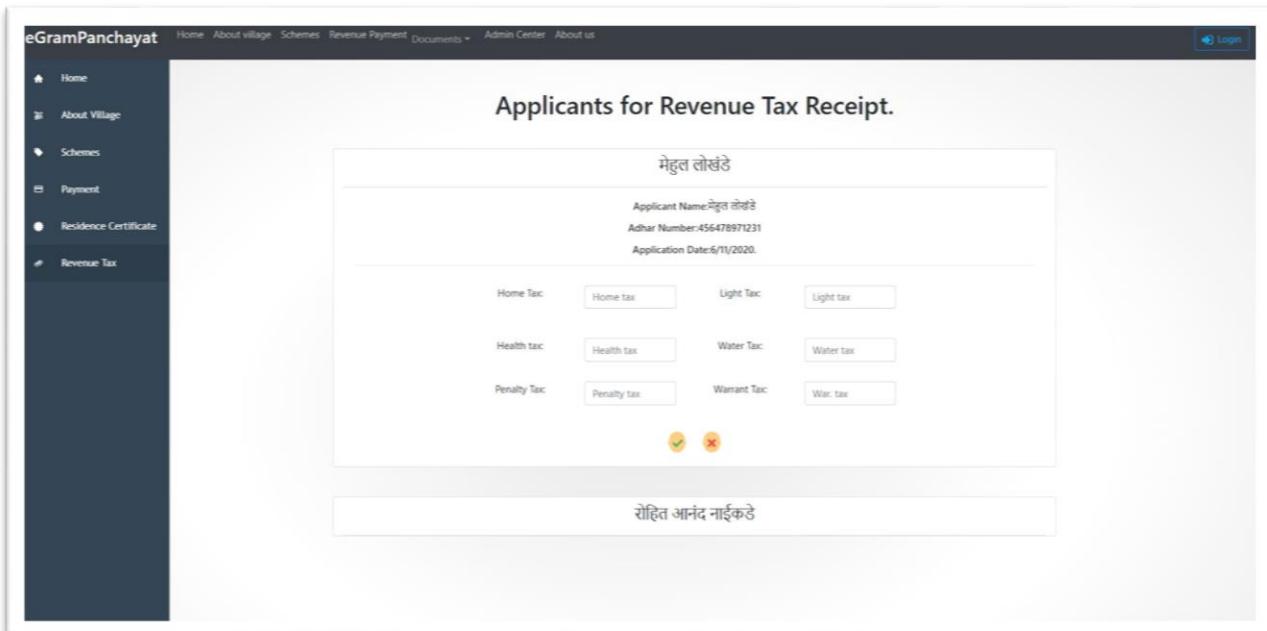
```

```

css={override}
size={150}
color={"#123abc"}
loading
/>
)
}
}
export default AdResidence;

```

4.1.8.5 Revenue tax



The screenshot shows the eGramPanchayat application interface. The left sidebar has a dark blue background with white icons and text for Home, About Village, Schemes, Payment, Residence Certificate, and Revenue Tax. The main content area has a light gray background. At the top, it says "Applicants for Revenue Tax Receipt." Below this, there is a section titled "મેહુલ લોખંડે" (Memul Lohund) containing applicant details: Name: મેહુલ લોખંડે, Adhar Number: 456478971231, Application Date: 6/11/2020. There are four groups of tax checkboxes: Home Tax (Home tax, Light Tax), Health tax (Health tax, Water Tax), Penalty Tax (Penalty tax, Warrant Tax). Below these groups are two small circular icons: a green checkmark and a red X. At the bottom, there is a button labeled "રોઝિત આપના નાઈકાર્ય" (Rojita Apna Naikary).

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="myAccordion" defaultActiveKey={activeKey}>
          <Accordion.Toggle as={Card.Header} className="back" eventKey={data._id}>
            <h4>{data.name}</h4>
          </Accordion.Toggle>
        </Accordion>
      </Card>
    )
  })
}

```

```

<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>
          </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>
          </Row>
        </Form.Group>
      </Col>
    </Row>
  
```

```

        />
      </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={()=> generatePDF(data.name, data.UID)} />
      <FcIcons.FcDisapprove className="ml-3 icons" size={40} onClick={()=> 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;
```

4.1.9 About us Module

The screenshot shows a web page titled "eGramPanchayat" with a navigation bar including "Home", "About village", "Schemes", "Revenue Payment", "Documents", "Admin Center", "About us", and a "Login" button. The main content area is titled "Team Baltic Knights." and displays four team members in separate cards:

- Rohit Naikade.** Team Leader. MERN stack developer. Rating: ★★★★☆. Social media icons: Facebook, Instagram, Twitter, LinkedIn.
- Mehul Lokhande.** Head of Front-End Team. Lead Programmer. Rating: ★★★★★. Social media icons: Facebook, Instagram, Twitter, LinkedIn.
- Vijay Dabhade.** Head of Backend Team. Database Expert. Rating: ★★★★☆. Social media icons: Facebook, Instagram, Twitter, LinkedIn.
- Govind Madankar.** Presentation & Marketing Head. Design & Marketing. Rating: ★★★★★. Social media icons: Facebook, Instagram, Twitter, LinkedIn.

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 imgs={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 imgs={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>
        </Row>
      </Container>
    );
  }
}
```

```
</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;
```

4.2 Test cases:

1) 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

2) 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

3) 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 5

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.
- Every villager is able to get all information about gram panchayats
- 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.

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/>