

ResolveNow - SmartInternz

=====

ResolveNow *Online Complaint Management System*

"Empowering users to register and resolve complaints seamlessly"

=====

Project Title : ResolveNow – Your Platform for Online Complaints

Category: Full Stack Development with MERN

Team ID: LTVIP2025TMID56856

Team Size: 4

Team Members:

- Bola Hari Kethan
- Battula Dwarakamai
- Boddukuru Lokesh
- Kota Bhargavi

Prepared by: Bola Hari Kethan

Full Stack Developer

Submitted to:

SmartInternz | AICTE | RISE Krishna Sai Prakasam Group Of Institutions



अखिल भारतीय तकनीकी शिक्षा परिषद्
All India Council for Technical Education



Project Overview

Purpose:

ResolveNow is an online complaint registration and management system that simplifies the complaint handling process for users, agents, and admins. It aims to empower organizations to handle customer complaints efficiently, ensuring transparency and quick resolution.

Features:

- User registration and secure login
 - Complaint submission with detailed fields
 - Track complaint status and updates
 - Chat window to communicate with assigned agent
 - Admin dashboard to monitor and assign complaints
 - Agent dashboard to manage assigned complaints
 - Real-time notifications
 - Security and confidentiality with authentication and access controls
-

Architecture

Frontend:

- Built with React.js
- Uses HTML, CSS, JavaScript, Bootstrap, and Material UI
- Axios for API communication
- React Router for routing
- Organized reusable components and pages
- Real-time UI updates

Backend:

- Node.js + Express.js server

- RESTful APIs for CRUD operations
- Authentication and middleware
- Routes: user, agent, admin, complaints

Database:

- MongoDB (NoSQL)
- Collections: user_Schema, complaint_schema, assigned_complaint, message
- Mongoose ODM for schema definitions and interactions
- Secure storage and efficient queries

Additional:

- Chat system uses complaintId and userId references
- Real-time updates and status changes handled via API

Setup Instructions

Prerequisites:

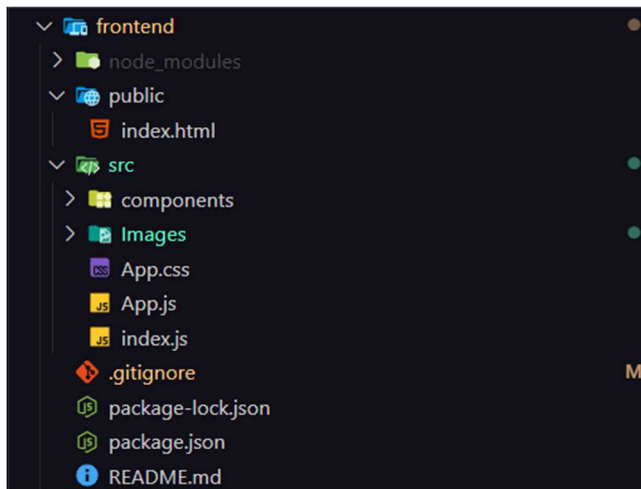
- Node.js (v18+)
- MongoDB (local or Atlas)
- Git

Installation:

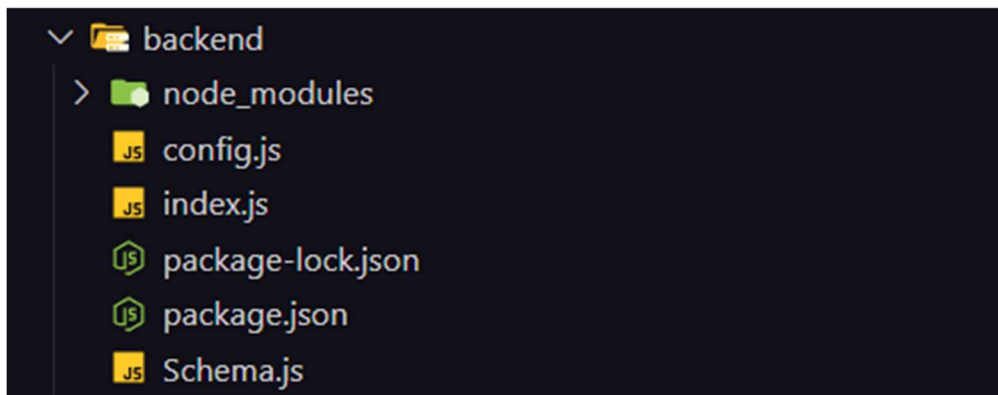
1. Download the project from Google Drive: [Click Here --- Drive Link](#)
2. Extract the ZIP file (if zipped) and navigate into the folder.
3. Install frontend dependencies:
cd frontend
npm install
4. Install backend dependencies:
cd ../backend
npm install
5. Create a .env file inside /backend:
MONGO_URI=your_mongodb_connection_string
JWT_SECRET=your_secret_key

Folder Structure

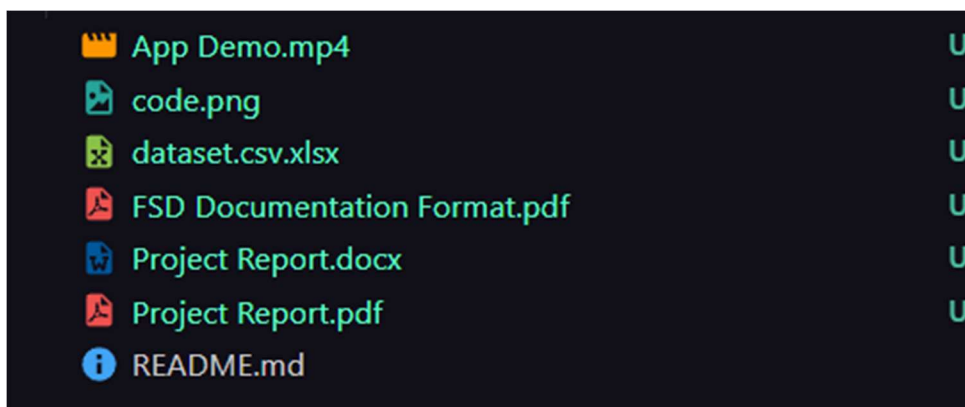
Frontend (React):



Backend (Node.js):



Other:



Running the Application

Frontend:

```
E:\PROJECT\SmartBridge Online Complaints>cd frontend
E:\PROJECT\SmartBridge Online Complaints\frontend>npm start

> task1@0.1.0 start
> react-scripts start

Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
(node:16628) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:16628) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view task1 in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.109.144:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Backend:

```
E:\PROJECT\SmartBridge Online Complaints>cd backend
E:\PROJECT\SmartBridge Online Complaints\backend>npm start

> backend@1.0.0 start
> nodemon index.js

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
server started at 8000
✓ Connected to MongoDB: OnlineComplaint
```

Open browser: <http://localhost:3000>

API Documentation (Example)

POST /api/register

- **Register a new user**

Body:

```
{  
  "name": "Bola Hari Kethan",  
  "email": "ketan@gmail.com",  
  "password": "mysecurepass"  
}
```

POST /api/login

- **Login user**

Body:

```
{  
  "email": "ketan@gmail.com",  
  "password": "mysecurepass"  
}
```

POST /api/complaints

- **Submit new complaint**

Body:

```
{  
  "userId": "123",  
  "description": "Street light not working in Gaddalagunta Palem",  
  "address": "2-1-38, Water Tank degara, Gaddalagunta Palem, Ongole",  
  "status": "pending"  
}
```

GET /api/complaints/123

- **Get complaints submitted by user Bola Hari Kethan**

PUT /api/complaints/456

- **Update status of complaint (e.g., mark as resolved)**

Body:

```
{  
  "status": "resolved"  
}
```

(* Full detailed list available in final version *)

Authentication

- JWT-based authentication
 - Passwords hashed using bcrypt
 - Role-based access: User, Agent, Admin
 - Auth middleware to protect routes
 - Secure token storage
-

User Interface

- Landing page with login/register options
 - Dashboard to submit complaints and track status
 - Admin dashboard to assign/manage complaints
 - Agent dashboard to update and resolve complaints
 - Built-in chat window to communicate with agent
 - Responsive design with Bootstrap and Material UI
-

Testing

- Manual testing of flows
- API testing with Postman

- Validation of database operations with MongoDB Compass
 - (Optional) Plan to add Jest for unit testing
-

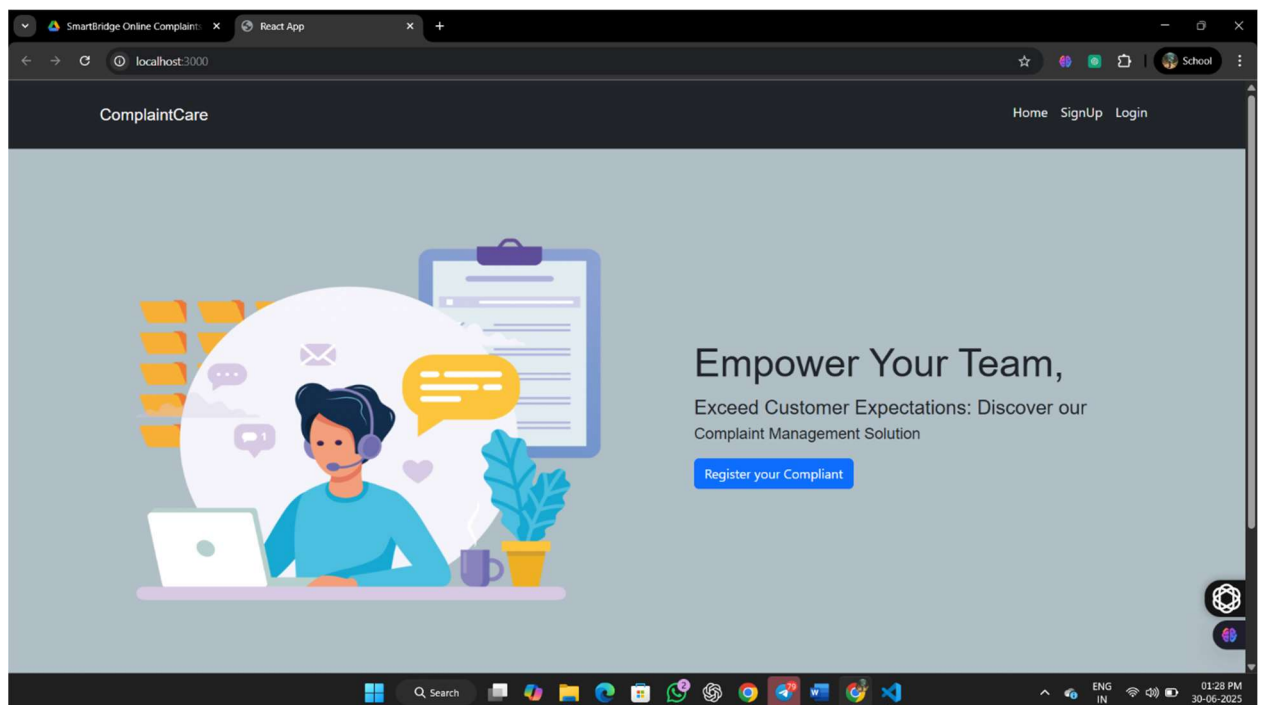
Screenshots or Demo

Project demo video :

<https://drive.google.com/file/d/1RRF34M8DiE24LamNBjGNdvaJyxoITlgx/view?usp=sharing>

OUTPUT SCREENSHOTS

- Home Page



- **Registration For OnlineComplaint**

SignUp For Registering the Complaint

Please enter your details

Full Name

Email

Password

Ordinary
Admin
Agent

Select User ▼

Select User Type

SignUp For Registering the Complaint

Please enter your details

Full Name

Email Please fill out this field.

Password

Mobile No.

Select User ▼

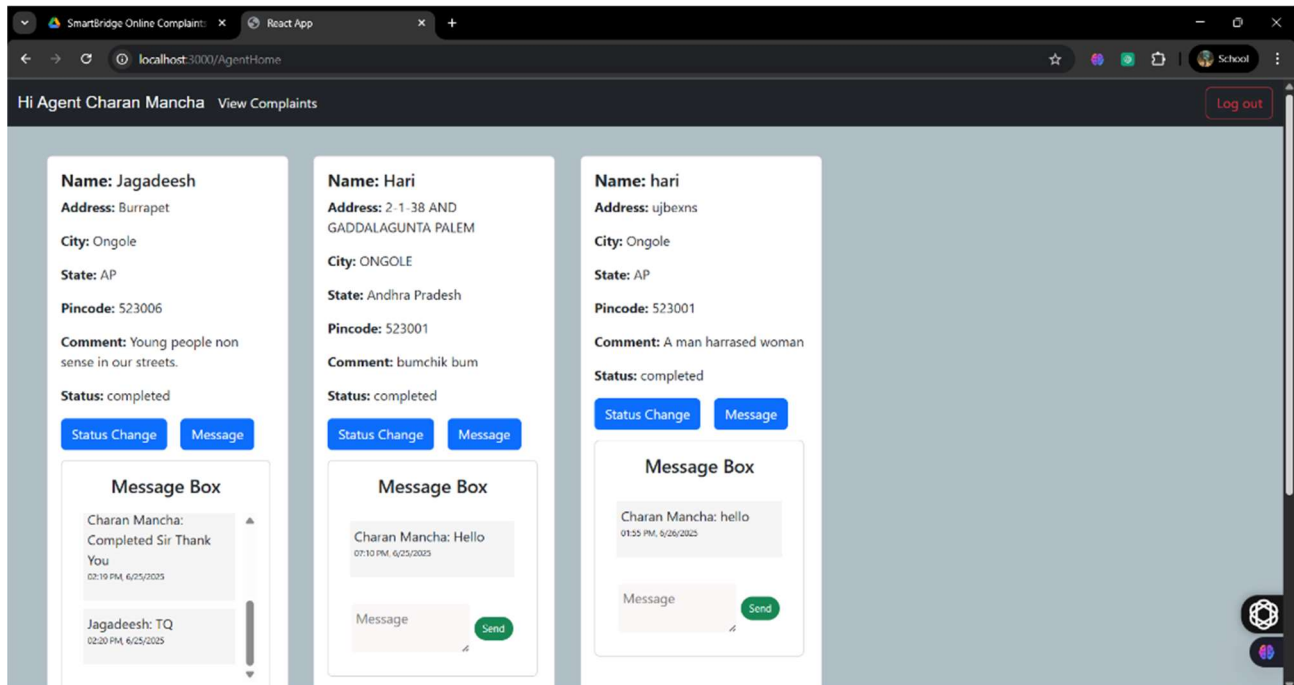
Select User Type

- **User Login : To raise a Complaint**

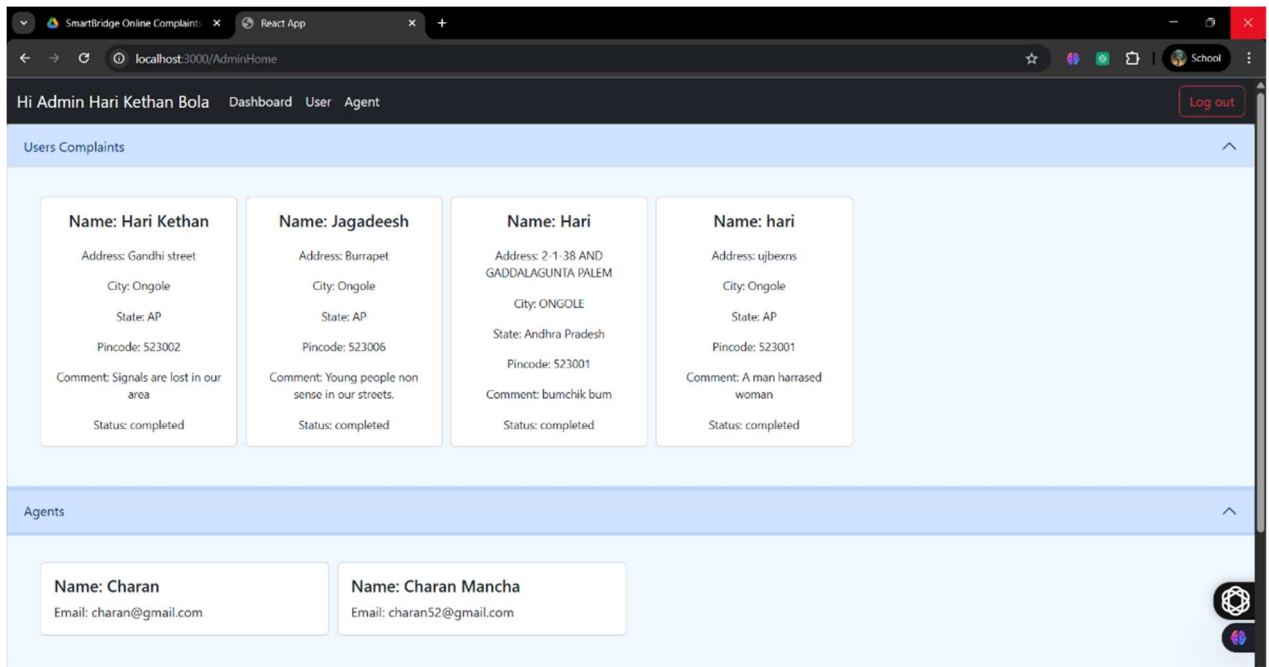
The screenshot shows a web browser window with the URL `localhost:3000/HomePage`. The page has a dark header with the text "Hi, Hari" and navigation links "Complaint Register" and "Status". A red "LogOut" button is in the top right. The main content area features a dark gray form box with the following fields: "Name" (text input), "Address" (text input), "City" (text input), "State" (text input), "Pincode" (text input), "Status" (text input with the value "type pending"), and "Description" (text area). A green "Register" button is at the bottom of the form. The footer of the page says "ComplaintCare".

The screenshot shows the same web browser window, but now displaying the details of a registered complaint. The header and footer are identical to the previous screenshot. The main content area shows a white box with the following information: "Name: hari", "Address: ujbexns", "City: Ongole", "State: AP", "Pincode: 523001", "Comment: A man harrassed woman", and "Status: completed". A blue "Message" button is located at the bottom right of this box.

- **Agent Login**



- **Admin Login**



Known Issues

- No email/SMS notifications yet
- Basic validation on some forms
- Password reset feature missing

Future Enhancements

- Add real-time notifications via socket.io
- Email/SMS alerts on status change
- Analytics dashboard for admin
- Mobile app version using React Native
- Improve chat experience with file attachments

Technical Details

- Frontend: React.js, Bootstrap, Material UI, Axios
- Backend: Node.js, Express.js, REST API
- Database: MongoDB, Mongoose
- Version control: Git & GitHub
- Deployment ready for cloud (Heroku / Vercel)

Reference Links

- Project Drive link : https://drive.google.com/drive/folders/1kP6ZsEKbWHyV73YR7WpO4gnO-yfyy_sL?usp=drive_link
- Source code GitHub : <https://github.com/HariKethanBola/RESOLVENOW>

END