

FullStackDevelopmentwithMERN

Project Documentation format

1. Introduction

- **ProjectTitle:**ResolveNow:ComplaintRegistrationandManagement

TeamMembers: Shaik Sameer Ahamed

P Manjunath

Pavan Kumar

Rajesh Medapuram

•

2. ProjectOverview

- **Purpose:**ResolveNowaimstostreamlinethecomplaint managementprocessbyallowing userstoregister issues, tracktheir status,andinteractwithagents. Itprovidesa centralized, secure, and user-friendly web application for efficient complaint resolution.
- **Features:**
 - UserRegistration/Login
 - ComplaintSubmission
 - StatusTracking
 - Real-timeMessaging
 - Role-BasedAccessforAdmins, Agents,andUsers
 - Complaint AssignmentbyAdmin
 - AgentDashboardforComplaintHandling

3. Architecture

- **Frontend:**BuiltusingReact.jswithReactRouterfornavigation.StyledusingBootstrap and MDB React UI Kit. Axios is used for HTTP communication.
- **Backend:**BuiltusingNode.jsandExpress.js.ContainsRESTfulAPIendpointsforusers, complaints, messages, and assignments.
- **Database:**MongoDBwithMongooseODMisusedtostoreuser details, complaints, assigned complaints, and messages. Schemas are defined for each collection.

4. SetupInstructions

- **Prerequisites:**
 - Node.js(v14+)

- npm
- MongoDB installed locally or MongoDB Atlas

- **Installation:**

```
#Clonetheproject
gitclonehttps://github.com/your-username/ResolveNow.git cd
ResolveNow
```

```
#SetupBackend
cd backend
npminstall
nodeindex.js#oruse nodemon
```

```
#SetupFrontend
cd ../frontend
npm install
npmstart
```

5. FolderStructure

- **Client:**DescribethestructureoftheReactfrontend.

```
frontend/
|
|—public/
|
|—src/
|
| |— components/
| |
| | |—common/
| | |
| | |—admin/
| | |
| | |—agent/
| |
| |└─user/
|
| |— App.js
|
|   └─ index.js
```

- **Server:**ExplaintheorganizationoftheNode.jsbackend.

```
backend/
|— index.js
|— schema.js
|— config.js
```

6. RunningtheApplication

- Providecommandstostartthefrontendandbackendserverslocally.
 - **Frontend:**cd


```
frontend
npm start
```

- **Backend:**

```
cd backend
nodeindex.js
```

7. API Documentation

- **POST**/SignUp—Register a user
- **POST**/Login—Login user
- **POST**/Complaint/:id—Register a complaint by user ID
- **GET**/status/:id—Get complaints by user ID
- **POST**/messages—Send a message
- **GET**/messages/:complaintId—Get messages by complaint ID
- **GET**/AgentUsers—Fetch all agents
- **GET**/OrdinaryUsers—Fetch all ordinary users
- **DELETE**/OrdinaryUsers/:id—Delete a user and their complaint
- **PUT**/user/:id—Update user details
- **PUT**/complaint/:complaintId—Update complaint status
- **POST**/assignedComplaints—Assign complaint to agent

8. Authentication

- Basic authentication using email and password.
- Session data is stored in `localStorage` on the client side.
- No tokens used yet—could be enhanced using JWT for better security.

9. User Interface

- Responsive Bootstrap design
- Role-based dashboards: Admin, Agent, User
- Complaint form, status tracker, and chat interface
- Clean navigation and feedback messages
- Dark-themed navbar and cards

10. Testing

Frontend Testing:

- Manual testing with React Developer Tools
- Form validation using Bootstrap

Backend Testing:

- API testing using Postman or Thunder Client
- MongoDB testing with MongoDB Compass

11. Screenshots or Demo

- <https://drive.google.com/file/d/1EbZcS3YRYe0345D6UeuFjGZNG4rVRG-Nla/view?usp=drivesdk>

12. Known Issues

- No email/SMS integration yet for notifications
- No password hashing (authentication security could be improved)
- No file uploads for complaint proofs

13. Future Enhancements

- Add JWT-based authentication
- Implement email/SMS notifications
- Add document/image upload for complaints
- Integrate pagination and search
- Build a dashboard analytics for Admin
- Role-based access middleware in backend