FullStackDevelopmentwithMERN

Project Documentation format

1. Introduction

- **ProjectTitle:**ResolveNow:ComplaintRegistrationandManagement
- **TeamMembers: D Abdulsalam**
  Dudekula Yaseen Basha
  Shaik Mahammad Asif
  D Hasen Basha

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

  - MongoDBinstalled locallyorMongoDBAtlas

- **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. APIDocumentation

- **POST**`/SignUp`–Registerauser
- **POST**`/Login`–Loginuser
- **POST**`/Complaint/:id`–Registeracomplaint byuserID
- **GET**`/status/:id`–GetcomplaintsbyuserID
- **POST**`/messages`–Sendamessage
- **GET**`/messages/:complaintId`–Getmessagesbycomplaint ID
- **GET**`/AgentUsers`–Fetchallagents
- **GET**`/OrdinaryUsers`–Fetchallordinaryusers
- **DELETE**`/OrdinaryUsers/:id`–Deleteauserandtheir complaint
- **PUT**`/user/:id`–Updateuser details
- **PUT**`/complaint/:complaintId`–Updatecomplaintstatus
- **POST**`/assignedComplaints`–Assigncomplaintto agent

8. Authentication

- Basicauthenticationusingemailandpassword.
- Sessiondataisstoredin`localStorage`ontheclientside.
- Notokensusedyet—couldbeenhancedusingJWTforbettersecurity.

9. UserInterface

- ResponsiveBootstrapdesign
- Role-baseddashboards:Admin,Agent,User
- Complaintform,statustracker,andchatinterface
- Cleannavigationandfeedbackmessages
- Dark-themednavbarandcards

10. Testing

FrontendTesting:

- ManualtestingwithReactDeveloper Tools
- Formvalidationusing Bootstrap

Backend Testing:

- APItestingusingPostmanorThunderClient
- MongoDBtestingwithMongoDBCompass

11. ScreenshotsorDemo

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

12. KnownIssues

- Noemail/SMS integrationyetfornotifications
- Nopasswordhashing(authenticationsecuritycouldbeimproved)
- Nofileuploadsforcomplaintproofs

13. FutureEnhancements

- AddJWT-basedauthentication
- Implementemail/SMSnotifications
- Adddocument/imageuploadforcomplaints
- Integratepaginationandsearch
- BuildadashboardanalyticsforAdmin
- Role-basedaccessmiddlewareinbackend