Frontend Development with React.js

Project Documentation format

1. **Introduction**

- **Project Title:** [Online Complaint Registration and Management System]
- **Pratham Pawar** –Frontend Developer, Backend Developer,Database Management,Project Documentation

2. **Project Overview**

- **Purpose:** The purpose of this project is to make it easy for people to register complaints online and for administrators to manage them. It aims to save time, reduce paperwork, and provide a transparent process for handling complaints.
- Features:
- User registration and login
- Submit complaints online
- View complaint status
- Admin dashboard to manage complaints
- Search and filter complaints
- Update complaint status

3. Architecture

Component Structure:

- Header & Footer – For navigation and branding
- Home Page – Shows project introduction and features
- Login & Signup – For authentication
- Complaint Form – For submitting complaints
- Complaint List – Shows submitted complaints
- Admin Dashboard – Manage all complaints

State Management:

- Context API for global state (user authentication, complaint data)
- Local state with React's useState and useEffect for individual components

Routing:

- React Router DOM for page navigation
- Routes for /, /login, /signup, /complaints, /admin

4. Setup Instructions

Prerequisites:

- Node.js (latest version)
- npm or yarn installed
- MongoDB running locally or on cloud (e.g., MongoDB Atlas)

Installation:

1. Clone the repository

```
git clone [repository-link]
```

2. Go to the project folder

```
cd client
```

3. Install dependencies

```
npm install
```

4. Create a .env file and add environment variables (API URLs, DB connections)
5. Start the application

```
npm start
```

5. Folder Structure

Client:

- components/ – Header, Footer, ComplaintForm, etc.
- pages/ – Home, Login, Signup, ComplaintList, AdminDashboard

- assets/ – Images, icons, and styles
- App.js – Main entry point
- App.css – Global styling

Utilities:

- api.js – API request functions
- auth.js – Authentication helper functions

6. Running the Application

```
cd client
npm start
```

7. Component Documentation

Key Components:

- Header – Shows navigation menu
- ComplaintForm – Accepts complaint details from user
- ComplaintList – Displays complaints for users or admin
- AdminDashboard – Allows admin to update complaint status

Reusable Components:

- Button – Custom styled button used across the app
- InputField – Reusable input field for forms

8. State Management

Global State:

- User authentication and complaint data handled using Context API

Local State:

- Form inputs and temporary UI states handled with useState

9. User Interface

- Simple, clean, and responsive design
- Easy-to-use forms for complaint registration
- Dashboard for complaint management

10. Styling

- CSS and Bootstrap for responsive design
- Simple and modern look with a focus on usability

11.Testing

- Manual testing for all features
- Jest for unit testing of components.

12.Known Issues

- Email notifications not yet implemented
- Search filter may be slow with large data

13.Future Enhancements

- Add complaint tracking via SMS/Email
- Implement user roles and permissions
- Improve analytics for admin