**Project Design Phase-II**

**Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| Date | 16 July 2025 |
| Team ID | LTVIP2025TMID59305 |
| Project Name | Resolve Now-Your Platform for Online Complaints |
| Maximum Marks | 4 Marks |

**Functional Requirements:**

**1. User Management**

* **Registration/Login:** Users can sign up and log in as Customer/User, Support Agent, Admin.
* **Authentication & Authorization:** JWT-based authentication to access role-specific features.
* **Profile Management:** Users can update their personal details.

**2. Complaint Handling**

* **Submit Complaint:** Registered users can submit complaints with Title, Description, Category, Address, upload files.
* **Complaint Tracking:** Users can view the status of all complaints (e.g., *Pending*, *In Progress*, *Resolved*, *Closed*).
* **Complaint History:** Each user can access a log of all previously submitted complaints.

**3. Agent Interaction**

* **Assign Complaint:** Admin can assign complaints to agents manually or via logic-based routing
* **Chat System/Inquiry:** Built-in messaging between user and assigned agent for direct communication and clarification.

**4. Complaint Resolution Workflow**

* **Status Management:** Agents can update complaint status.
* **Resolution Notes:** Agents can add comments or resolution notes visible to the user.
* **User Feedback:** After complaint resolution, users can rate their experience and submit feedback.

**5. Document Management**

* **Upload Resolution Documents:** Agents or Admins can upload files related to the resolution (receipts, proof, etc.).
* **PDF Generation:** Users can download complaint summaries or resolution reports as **PDFs**.

6. **Admin Panel**

* **User Management:** Admin can view, block/unblock, or delete users (customers or agents).
* **Complaint Oversight:** Admin can view, filter, and manage all complaints across the system.
* **Agent Assignment:** Assign or reassign agents to unresolved complaints.

**7. Search & Filter**

* **Search Complaints:** Users, agents, and admins can search complaints by ID, status, date, or keywords.
* **Filter Options:** Filter complaints by status, priority, category, or agent.

**8. Security & Compliance**

* **Authentication:** JWT-secured endpoints and password encryption (bcrypt or similar).
* **Role-Based Access Control (RBAC):** Different roles have specific capabilities and limited access scope**.**
* **Data Protection:** Secure handling of sensitive information in compliance with data protection laws (like GDPR if applicable).

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Simple, intuitive UI for all user roles. |
| NFR-2 | **Security** | JWT auth, bcrypt password hashing, secure file uploads |
| NFR-3 | **Reliability** | Responsive UI to support desktop, tablet, and mobile users. |
| NFR-4 | **Performance** | App should handle 100+ concurrent users efficiently. |
| NFR-5 | **Availability** | Built on scalable MERN stack architecture (modular backend + frontend components). |
| NFR-6 | **Scalability** | Modular, well-commented codebase with version control (Git). |

**🧾 Technical Requirements:**

* **Frontend**: React.js, TailwindCSS/Bootstrap
* **Backend**: Node.js, Express.js
* **Database**: MongoDB (with Mongoose ORM)
* **Authentication**: JWT (JSON Web Token)
* **File Handling**: Multer for file uploads
* **Testing**: Postman for API testing, optional Jest/Mocha for unit tests
* **Deployment**: Vercel (Frontend) + Render/Heroku (Backend)
* **Version Control**: Git & GitHub