**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 28 June 2025 |
| Team ID | LTVIP2025TMID57433 |
| Project Name | Online Complaint Registration and Management System |
| Maximum Marks | 4 Marks |

**Technical Architecture – Online Complaint Registration System**

**1. User Interface:**

* **Responsive Web UI**
  + Built using **React.js** components.
  + Provides role-based views for **Users**, **Agents**, and **Admins**.
  + Interfaces include: Complaint submission form, complaint status view, agent dashboard, admin panel.

**2. Application Logic:**

**Frontend:**

* Built with **React.js** (Single Page Application).
* Handles user interactions like:
  + Complaint submission
  + Status tracking
  + Messaging/chat with agents
  + Admin assignment workflows

**Backend:**

* Built using **Node.js** and **Express.js**.
* REST APIs for:
  + User signup/login
  + Complaint creation and update
  + Agent assignment and status management
  + Message/comment handling (chat)

**Business Logic Includes:**

* **Complaint Filtering** by date/status/category.
* **Status Transitions** (Pending → In Progress → Resolved).
* **Role-based Access Control** for different user types.

**3. Database & Storage:**

**Database:**

* **MongoDB** (via Mongoose)
  + Collections: Users, Complaints, Messages, Agents, Assignments

**Authentication:**

* Custom **JWT-based authentication** for session management.
* (Optional) **Firebase Auth** or Google Sign-In for social login.

**Storage:**

* **Cloudinary** or **Firebase Storage** for storing:
  + Uploaded complaint evidence/images
  + Chat attachments (if applicable)

**4. Infrastructure:**

* **Deployment Platforms**:
  + Hosted on **Render**, **Vercel**, **Heroku**, or **Netlify**
* **CI/CD Integration**:
  + Automated deployment using **GitHub Actions** or **Vercel CLI**
* **Scalability**:
  + Future-ready with **Docker** support for containerization.

**5. External Interfaces:**

* **Email Service**:
  + Complaint submission and resolution notifications via **Nodemailer** (using Gmail SMTP or SendGrid)
* **Authentication API**:
  + (Optional) Integration with **Firebase Auth** or **Google OAuth**
* **Notification System**:
  + In-app and email alerts for complaint updates or agent responses

**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web UI for shopping | React JS, HTML, CSS, Bootstrap |
|  | Application Logic-1 | Status , pending ,resolved | JavaScript, React Hooks |
|  | Application Logic-2 | API logic and backend | Node.js, Express.js |
|  | Database | Store users data , complaints | MongoDB |
|  | Infrastructure | Deployment & scaling | Vercel / Heroku / Netlify / Docker |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frameworks and libraries used | React JS, Node.js, Express, MongoDB |
|  | Security Implementations | Data safety and access control | JWT Auth, HTTPS, Password Hashing (bcrypt) |
|  | Scalable Architecture | Designed to support more users | RESTful API, modular components, Docker |
|  | Availability | Accessible across platforms and regions | Cloud hosting with 99.9% uptime |
|  | Performance | Optimized complaint solving | Lazy loading, caching, optimized DB queries |