**Requirement Gathering and Analysis Phase**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 6th July 2024 |
| Team ID | SWTID1720076124 |
| Project Name | Online Complaint Registration and Management System |
| Maximum Marks |  |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



**Guidelines:**

Include all the processes (As an application logic / Technology Block)

Provide infrastructural demarcation (Local / Cloud)

Indicate external interfaces (third party API’s etc.)

Indicate Data Storage components / services

Indicate interface to machine learning models (if applicable)

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | How user interacts with application e.g.  Web UI etc. | HTML, CSS, JavaScript / React Js etc. |
|  | Complain Submission | The user fills the name, address and complaint description. | JavaScript / React Jsx /Express Js etc. |
|  | Complain Assignment | The admin selects from the list of registered agents that has to be assigned to the registered complain. | JavaScript /React Jsx / Express Js MongoDB modules etc. |
|  | Status Update | The agent will update the status of the registered complain from pending to completed once he has resolved the issue. | JavaScript /React Jsx / MongoDB modules / Express Js etc. |
|  | Database | Data Type, Configurations etc. | MongoDb / Mongoose |
|  | Cloud Database | Database Service on Cloud | MongoDb cloud modules |
|  | External API-1 | Purpose of this API is to generate and send an OTP to the user during registration to confirm their email and mobile number. | D7 Verify API |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Axios , Bootstrap , Express.js , Node.js | HTML, CSS, JavaScript |
|  | Security Implementations | **Security/Access Controls Implemented**: Bcrypt provides secure password hashing with automatic salt generation, adjustable hash complexity to counteract brute-force attacks, and protection against rainbow table and timing attacks.  **Use of Firewalls, etc.**: Bcrypt is typically deployed in secure environments with firewalls, intrusion detection/prevention systems (IDS/IPS), and other network security measures to ensure comprehensive protection. | Bcrypt |
|  | Scalable Architecture | The 3-tier architecture (presentation, application, and database layers) can be hosted in a cloud environment, allowing each layer to be independently scaled. Cloud platforms provide the necessary resources to handle increased loads and ensure high availability. | MongoDB Atlas |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Technology used |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Technology used |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)