**Requirement Gathering and Analysis Phase**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date |  |
| Team ID | SWTID1720026558 |
| Project Name | BOOK A DOCTOR USING MERN |
| 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**](http://S.No) | **Component** | **Description** | **Technology** |
| 1 | User Interface | Web UI for user interaction | HTML, CSS, JavaScript, React.js |
| 2 | Application Logic-1 | Backend logic for user management and authentication | Node.js, Express.js |
| 3 | Application Logic-2 | Integration with IBM Watson Speech to Text service | IBM Watson STT service |
| 4 | Application Logic-3 | Integration with IBM Watson Assistant for chatbot | IBM Watson Assistant |
| 5 | Database | User profiles, appointments storage | MongoDB |
| 6 | Cloud Database | Cloud-based database service | MongoDB Atlas, IBM Cloudant |
| 7 | File Storage | Storage for user documents and medical records | AWS S3, Google Cloud Storage |
| 8 | External API-1 | Integration for weather information | OpenWeather API, etc. |
| 9 | External API-2 | Integration for identity verification (e.g., Aadhar) | Aadhar API, etc. |
| 10 | Machine Learning Model | AI for medical diagnosis or recommendation | TensorFlow, Scikit-learn |
| 11 | Infrastructure | Deployment on cloud servers | AWS EC2, IBM Cloud Foundry |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| [**S.No**](http://S.No) | **Characteristics** | **Description** | **Technology Used** |
| 1 | Open-Source Frameworks | Utilizes open-source frameworks for development and integration | React.js, Node.js, Express.js, MongoDB |
| 2 | Security Implementations | Implements robust security measures including encryption (SHA-256), IAM controls, OWASP guidelines | SHA-256 hashing, TLS encryption, OAuth for API security |
| 3 | Scalable Architecture | Utilizes microservices architecture for scalability and flexibility | Docker, Kubernetes, AWS ECS, Node.js clusters |
| 4 | Availability | Ensures high availability through load balancers, distributed servers, and auto-scaling capabilities | AWS Elastic Load Balancing, Multi-region deployment, Kubernetes |
| 5 | Performance | Optimizes performance with caching strategies, CDN integration, and efficient request handling | Redis caching, CDN (Content Delivery Network), Load testing with JMeter |

**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)