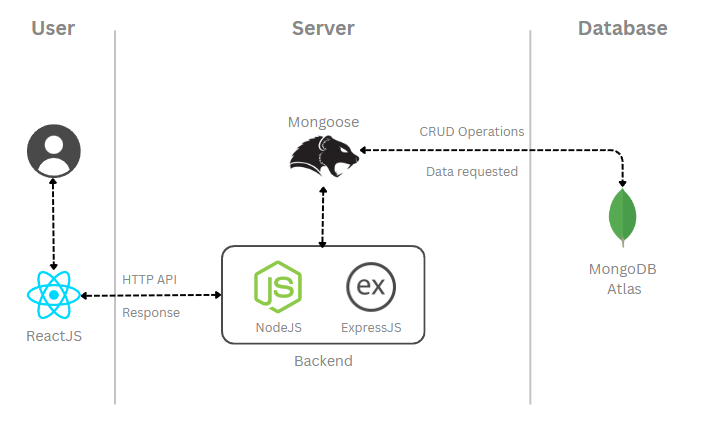
**Requirement Gathering and Analysis Phase**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 05-07-2024 |
| Team ID | SWTID1720166168 |
| Project Name | Project - Journify |
| Maximum Marks |  |

**Technical Architecture:**

****

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | This layer forms the front-end portion of web application. Users directly interact with this layer. | HTML, CSS, Bootstrap, JavaScript, ReactJS |
|  | Application Logic-1 | |  | | --- | | Frontend application logic handling user interactions and UI updates. |  |  | | --- | |  | | Node JS |
|  | Application Logic-2 | Backend application logic managing server-side processes and database interactions. | Node JS |
|  | Application Logic-3 | Web application framework for building RESTful APIs and handling HTTP requests. | Express JS |
|  | Database | |  | | --- | | Manages and stores application data with various data types and configurations. Ensures data  integrity and supports CRUD operations. |  |  | | --- | |  | | MongoDB Database, Mongoose |
|  | Cloud Database | Cloud-hosted database service providing scalable and managed data storage. | MongoDB Atlas |
|  | File Storage | File storage requirements | Local Filesystem |
|  | External API-1 | Integrates external services to fetch data or perform operations required by the application. | RESTful APIs |
|  | External API-2 | Connects to third-party services to enhance application functionality and user experience. | Axios HTTP Client |
|  | Infrastructure (Server / Cloud) | Local server configuration for development and testing on localhost. Supports deployment on various environments. | Local System Server |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | List the open-source frameworks used | NodeJS, ExpressJS, ReactJS, MongoDB |
|  | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | |  | | --- | | JWT, HTTPS, CORS (Cross-origin  resource sharing) |  |  | | --- | |  | |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | MongoDB (sharding and replication), React (component-based architecture), Node.js (asynchronous processing, clustering) |
|  | 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)