**Requirement Gathering and Analysis Phase**

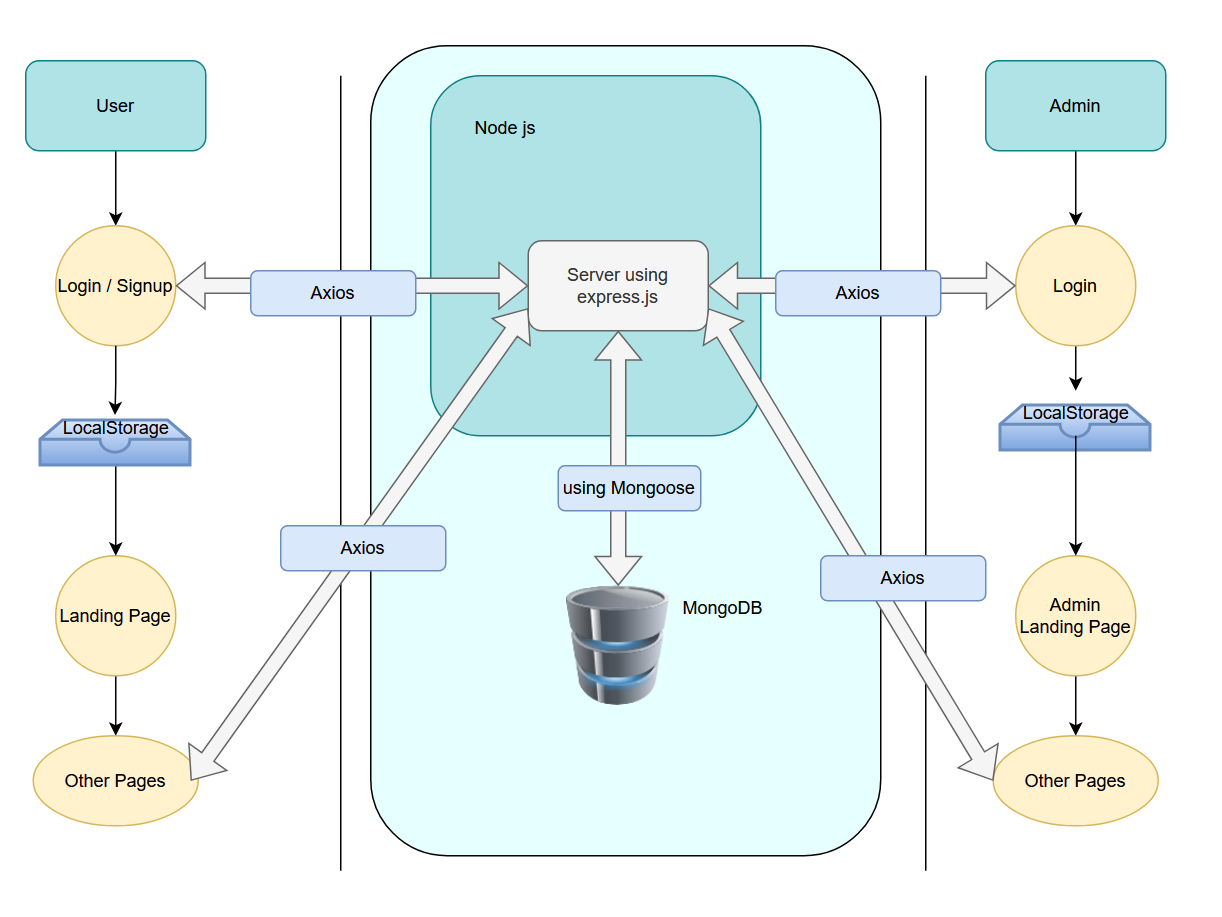
**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 11-07-2024 |
| Team ID | SWTID1720167264c |
| Project Name | Cab Booking App |
| Maximum Marks | 3 |

**Technical Architecture:**

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

**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 i.e Web UI | HTML, CSS, JavaScript, React Js, Bootstrap. |
|  | Application Logic-1 | Backend will be run with these technologies | Node.js, Express.js |
|  | Application Logic-2 | Used to communicate with the backend/server | Axios |
|  | Application Logic-3 | Stores the user details in the browser | LocalStorage |
|  | Database | Database name | MongoDB |
|  | Cloud Database | Database Service on Cloud | MongoDB Atlas |
|  | File Storage | Used to store Files such as images ,videos etc | Multer |
|  | External API-1 | Used to geolocation, maps and places data | Google maps api |
|  | External API-2 | - | - |
|  | Machine Learning Model | - | - |
|  | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud  Local Server Configuration:  Cloud Server Configuration : | Local Server |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | List the open-source frameworks used | React.js, Node.js, Express.js, Bootstrap, MongoDB |
|  | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | Basic authentication and authorization using custom logic in Express.js |
|  | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | MERN stack (MongoDB, Express.js, React.js, Node.js), Modular architecture |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Local server |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Lazy loading in React.js, Efficient MongoDB queries |

**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)