Requirement Gathering and Analysis Phase
Technology Stack (Architecture & Stack)

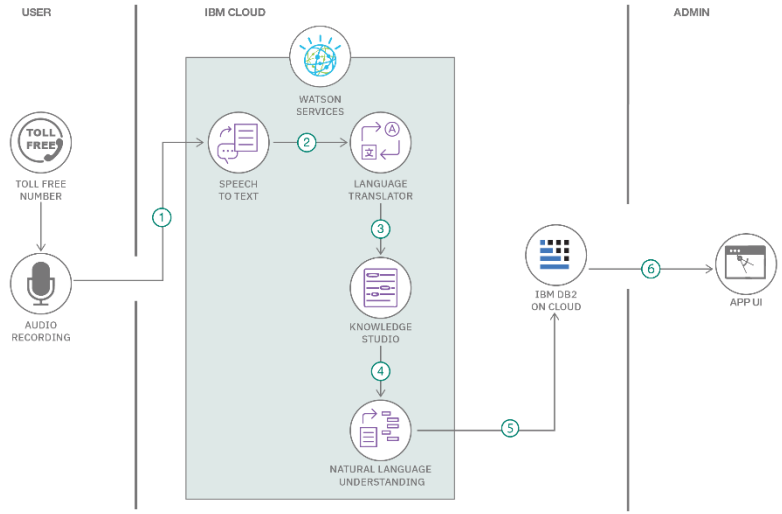
Date	06-07-2024
Team ID	SWTID1720073159
Project Name	TuneTrail
Maximum Marks	

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/



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
1.	User Interface	How users interact with application.	HTML, CSS, JavaScript/ React.js
2.	Application Logic-1	User Registration and Authentication.	Node.js, Express.js, Passport.js, JWT
3.	Application Logic-2	Song Listings, Playlist Creation, Library Management.	Node.js, Express.js
4.	Application Logic-3	Playback Control, Offline Listening	Node.js, Express.js
5.	Application Logic-4	Search Functionality	Node.js, Express.js
6.	Application Logic-5	User Profile Customization	Node.js, Express.js
7.	Application Logic-6	Song Recommendation based on User Trends	Node.js, Express.js
8.	Application Logic-7	Freaky Trail- Streaks based on Pauses & Plays	Node.js, Express.js
9.	Database	Data Storage for user data, songs, playlists, etc.	MongoDB
10.	Cloud Database	Cloud Database Service for scalability & reliability.	MongoDB Atlas
11.	File Storage	Storing song files and other media.	AWS S3, Google Cloud
12.	Infrastructure (Server / Cloud)	Application Deployment	Local Server.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used • MongoDB: NoSQL database for efficient data storage and retrieval. • Express.js: Backend framework for building web applications and APIs. • React.js: Frontend library for building user interfaces. • Node.js: JavaScript runtime for executing server-side code.	MongoDB, Express.js, React.js, Node.js
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc. • JWT (JSON Web Tokens): Secure authentication and authorization. • bcrypt: Password hashing for secure storage. • HTTPS: Ensures secure data transmission.	JWT, bcrypt, HTTPS
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services) Horizontal Scaling: Utilizes Node.js clusters to handle increased traffic by scaling out	Horizontal Scaling with Node.js Clusters.
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.) Distributed Servers: Ensures redundancy and high availability, across different geographical regions.	Distributed Servers, Multi-region Deployment.

S.No	Characteristics	Description	Technology
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. CDN (Cloudflare): Ensures fast content delivery and reduces latency.	Cloudflare CDN

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d